

Coupling Transport Equations to Mechanics in the Material Point Method Using an Approximate Full Capacity Matrix Inverse[☆]

John A. Nairn*

Wood Science and Engineering, Oregon State University, Corvallis, OR 97330, USA

Abstract

Coupling solutions of one or more transport equations to mechanics modeling in the material point method (MPM), extends MPM to a platform for solving multi-physics problems. Unfortunately, the standard MPM approach to transport equations using a lumped transport capacity matrix is prone to large oscillations in particle values especially near interfaces, cracks, or rapid changes in transport value. Switching from lumped capacity matrix methods to an approximate full capacity matrix approach can eliminate those oscillations. Although this switch can cause unwanted numerical diffusion, that problem can be suppressed using periodic or blending options. Several examples illustrate the new methods for heat conduction, solvent transport, poroelasticity, and phase field fracture simulations. In thermal calculations, the new methods improve accuracy for tracking thermodynamics state functions.

Keywords: Material Point Method, Transport Analysis, Computational Mechanics

1. Introduction

The material point method (MPM) is a computational mechanics tool [1, 2] with potential advantages for modeling large deformations, explicit cracks [3], contact [4, 5], and history-dependent materials. MPM extends to full-physics simulations by coupling solutions of mechanics and transport equations. This extension by standard MPM methods is straightforward [6, 7]. Some applications are to model plastic heating, crack tip heating, and frictional heating at interfaces [4].

In standard MPM transport analysis, some transport value (*e.g.*, temperature for heat conduction) is tracked on each particle. Each MPM time step extrapolates transport content (*e.g.*, heat energy for heat conduction) to the grid, solves the transport equation, and then extrapolates a transport “rate” back to the particles [6] using lumped transport-capacity matrix calculations. The particle state is then updated using the so-called FLIP (FLuid Implicit Particle) method [8, 9] that increments particle transport value using extrapolated rate. This mapping back and forth between particles and the grid with FLIP particle updates is standard practice in MPM, but is prone to noise caused by null-space in the particle to grid mappings. Because the number particles is normally greater than the number of nodes, particles can develop values that extrapolate to zero on the grid (*e.g.*, balanced oscillations within a grid cell). Such null-space values are transparent to the grid where equations are solved and thus act as noise that misrepresents the desired solution on the particles.

Methods for managing null-space noise and stabilizing MPM mechanics modeling are available [10, 11]. Null-space effects in transport modeling, however, can be more severe than in mechanics modeling. A typical observation is that FLIP methods get excellent grid values, but particle values develop very large oscillations near edges, interfaces, or other discontinuities. This behavior means modeling cannot implement features that depend on a particle’s transport value (*e.g.*, temperature-dependent material properties). One approach is to use particle values for solving transport equations, but when particle value is needed for some other feature, extrapolate it from grid values [6]. A better approach is to replace FLIP with a method that eliminates oscillations such that particle values and grid values match and both are accurate. This option

[☆] *Computer Methods in Applied Mechanics and Engineering*, **420**, 116757 (2024). DOI: 10.1016/j.cma.2024.116757

*Corresponding author: john.nairn@oregonstate.edu, Tel: +1-541-737-4265, Fax: +1-541-737-3385

restores MPM's advantage for modeling history-dependent materials because the particle transport history becomes a reliable value.

This paper applies recent full mass matrix methods for MPM mechanics, known as FMPM(k) [11], to derive a full transport-capacity matrix method for solving transport equations. The "Numerical Methods for Heat Transport" section first solves one specific transport equation for modeling thermal conduction. The "Other Transport, Boundary Conditions, and Coupling" next generalizes transport methods to include solvent diffusion, poroelasticity, and phase field diffusion. This section also describes boundary conditions and methods to handle transport-mechanics coupling terms (*e.g.*, stresses causing temperature changes), residual stresses caused by transport (*e.g.*, thermal expansion), proper thermodynamics calculations (*e.g.*, heat energy and entropy), and the disparate time steps needed when modeling both mechanics and transport. The "Results and Discussion" section shows that FMPM(k) for transport eliminates all oscillations and when done correctly is more accurate than FLIP.

2. Numerical Methods for Heat Transport

The transport equation for flow of heat energy, τ (in J), can be cast as:

$$c_\theta \frac{\partial \theta}{\partial t} = -\nabla \cdot \mathbf{q}(\mathbf{x}) + \dot{q}(\mathbf{x}) \quad (1)$$

where θ is temperature, c_θ is constant-volume heat capacity (in J/(K·m³)), $\mathbf{q}(\mathbf{x}) = -\mathbf{k}\nabla\theta$ is heat flux (in J/(sec·m²)), and $\dot{q}(\mathbf{x})$ is a source term (in J/(sec·m³)). In the flux term, $\mathbf{k}$ is thermal conductivity tensor (in W/(m·K)).

Solving Eq. (1) in the weak form starts with [2]:

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

where $w(\mathbf{x})$ is an arbitrary weighting function. This form reduces to MPM equations by using generalized interpolation material point analysis or GIMP [1, 6, 7]. In outline form, the steps are: 1. Use divergence theorem to eliminate divergence term in Eq. (2) [2]. 2. Expand transport terms in a particle basis:

$$c_\theta \frac{\partial \theta}{\partial t} = \sum_p C_{\Theta,p} \frac{\partial \Theta_p}{\partial t} \chi_p(\mathbf{x}), \quad \mathbf{q}(\mathbf{x}) = \sum_p \mathbf{Q}_p \chi_p(\mathbf{x}), \quad \text{and} \quad \dot{q}(\mathbf{x}) = \sum_p \dot{Q}_p^{(g)} \chi_p(\mathbf{x})$$

where $\chi_p(\mathbf{x})$ is a particle basis function [1]. Note that particle quantities are denoted with uppercase letters while transport equation and grid quantities (below) use lowercase letters (*e.g.*, θ and Θ for temperature, c_θ and C_Θ for heat capacity, $\mathbf{q}$ and $\mathbf{Q}$ for flow, *etc.*) and subscripts p indicate a quantity for particle p .[†] Particle source terms are divided into $\dot{Q}_p = \dot{Q}_p^{(g)} + \dot{Q}_p^{(p)}$ where $\dot{Q}_p^{(g)}$ is source extrapolated to the grid (done above) and $\dot{Q}_p^{(p)}$ is source added below during particle update. Although $\dot{Q}_p^{(g)}$ is used in prior MPM heat condition [6, 7] and included here for generality, a new recommendation below is that all particle sources should be in $\dot{Q}_p^{(p)}$ leaving $\dot{Q}_p^{(g)} = 0$. 3. Expand weight function and its gradient using grid shape functions (typically linear or spline shape functions [11]):

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

where subscripts i indicate a quantity for node i . 4. Substitute into the weak form and use arbitrary nature of $w(\mathbf{x})$ to derive an equation for each node:

$$\sum_p V_p C_{\Theta,p} \frac{\partial \Theta_p}{\partial t} S_{pi} = \sum_p V_p \mathbf{Q}_p \cdot \mathbf{G}_{pi} + \sum_p V_p \dot{Q}_p^{(g)} S_{pi} - \int_{\delta V} (N_i(\mathbf{x}) \mathbf{q}(\mathbf{x})) \cdot \hat{n} dS \quad (3)$$

[†]MPM equations refer to timestep j , but for simplicity, superscripts (j) (*e.g.*, $C_{\Theta,p}^{(j)}$) have been omitted

where V_p is particle volume and S_{pi} and $\mathbf{G}_{pi}$ are MPM shape functions and gradients [1]:

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

These equations are for Cartesian coordinates; extension to axisymmetry is straight forward [6].

2.1. MPM Grid Equations

For compactness, this section introduces vectors or matrices of quantities on particles (upper case letters) or on nodes (lower case letters) and for shape functions. We define a vector for nodal heat energy extrapolated from the particles using:

$$\boldsymbol{\tau} = \mathbf{S}^T \mathbf{C} \boldsymbol{\Theta} \quad (4)$$

where $\mathbf{C} = \text{diag}(\mathbf{C})$ is a diagonal matrix with $C_p = V_p C_{\Theta,p}$ on the p^{th} diagonal and $\boldsymbol{\Theta}$ is a vector of particle temperatures (Θ_p). Extrapolating $\boldsymbol{\tau}$ to the grid is analogous to extrapolating momentum in MPM mechanics modeling. Nodal temperatures, $\boldsymbol{\theta}$, are related to heat energy by:

$$\boldsymbol{\theta} = \tilde{\mathbf{c}}^{-1} \boldsymbol{\tau} \quad \text{where} \quad \tilde{\mathbf{c}} = \mathbf{S}^T \mathbf{C} \mathbf{S} \quad \left(i.e., (\tilde{\mathbf{c}})_{ij} = \sum_p V_p C_{\Theta,p} S_{ip}^T S_{pj} \right)$$

is a full heat-capacity matrix analogous to a full mass matrix in MPM mechanics [2, 11].

The nodal heat equation in Eq. (3) becomes:

$$\frac{d\boldsymbol{\tau}}{dt} = \mathbf{f} + \mathbf{s} \quad (5)$$

where

$$\mathbf{f} = \mathbf{G}^T \mathbf{V} \mathbf{Q} + \mathbf{S}^T \mathbf{V} \dot{\mathbf{Q}} \quad \text{and} \quad s_i = - \int_{\delta V} (N_i(\mathbf{x}) \mathbf{q}(\mathbf{x})) \cdot \hat{\mathbf{n}} dS \quad (6)$$

are heat forces (in J/sec or watts). Here $\mathbf{V}$ is a diagonal matrix with V_p on the p^{th} diagonal, while $\mathbf{Q}$ and $\dot{\mathbf{Q}}$ are vectors of particle values $\mathbf{Q}_p$ and $\dot{\mathbf{Q}}_p^{(g)}$, respectively. The gradient shape function matrix has $(\mathbf{G}^T)_{ip} = \mathbf{G}_{pi}$ as row vectors. Nodal $\mathbf{f}$ are internal forces while $\mathbf{s}$ are forces at object surfaces due to flux boundary conditions.

Particle fluxes, $\mathbf{Q}_p$ (column vectors for each element of $\mathbf{Q}$), are found from temperature gradient on each particle:

$$\mathbf{Q}_p = -\mathbf{k}_p \nabla \Theta_p$$

where $\mathbf{k}_p$ is the particle's thermal conductivity tensor. Rather than track particle fluxes, they are calculated each time step by extrapolating grid temperature back to the particles using:

$$\nabla \boldsymbol{\Theta} = \mathbf{G} \boldsymbol{\theta} = \mathbf{G} \tilde{\mathbf{c}}^{-1} \boldsymbol{\tau} \quad (7)$$

Giving these fluxes, Eq. (5) is solved by explicit Euler's method to give updated nodal energy:

$$\boldsymbol{\tau}^+ = \boldsymbol{\tau} + (\mathbf{f} + \mathbf{s}) \Delta t \quad \text{and} \quad \boldsymbol{\theta}^+ = \tilde{\mathbf{c}}^{-1} \boldsymbol{\tau}^+$$

where Δt is time step and superscript “+” indicates an updated grid energy in the current time step.

2.2. Particle Updates

Because inverting the $n \times n$ $\tilde{\mathbf{c}}$ matrix is impractical (where n is number of active nodes), all prior MPM transport methods simplified this analysis by assuming a diagonal lumped heat capacity matrix, $\mathbf{c} = \text{diag}(\mathbf{c}^L)$ (without a tilde), where diagonal elements are given by the vector $\mathbf{c}^L$:

$$c_i^L = \sum_j \tilde{\mathbf{c}}_{ij} \quad \implies \quad \mathbf{c}^L = \mathbf{S}^T \mathbf{C} \quad \left(i.e., (c^L)_i = \sum_p V_p C_{\Theta,p} S_{ip}^T \right) \quad (8)$$

When using a lumped matrix, lumped nodal temperatures and their updates become:

$$\boldsymbol{\theta}^L = \mathbf{c}^{-1} \boldsymbol{\tau} \text{ or } \theta_i^L = \frac{\tau_i}{c_i^L} \quad \text{and} \quad \boldsymbol{\theta}^{L+} = \mathbf{c}^{-1} \boldsymbol{\tau}^+ \text{ or } \theta_i^{L+} = \theta_i^L + \frac{(f_i + s_i) \Delta t}{c_i^L}$$

Given these nodal temperatures, all prior MPM for transport [6, 7] updated particle values using the FLIP method [8, 9] with a transport velocity, $\mathbf{v}^{(\theta)}$, extrapolated from the grid or:

$$\boldsymbol{\Theta}^+ = \boldsymbol{\Theta} + \mathbf{S} \mathbf{v}^{(\theta)} \Delta t + d\boldsymbol{\Theta}^{(ad)} \quad \text{where} \quad v_i^{(\theta)} = \frac{\Delta \theta_i^L}{\Delta t} = \frac{f_i + s_i}{c_i^L} \quad (\text{FLIP}) \quad (9)$$

and $d\boldsymbol{\Theta}^{(ad)}$ is vector of adiabatic source terms arising from $\dot{Q}_p^{(p)}$ on the particles (see below). This FLIP approach gets good grid temperatures, $\boldsymbol{\theta}^{L+}$, but particle temperatures, $\boldsymbol{\Theta}^+$, can develop problematic oscillations.

Oscillations can be eliminated by switching to a new method. Replacing the lumped heat capacity matrix with the full heat capacity matrix, updated particle temperatures can be found from:

$$\boldsymbol{\theta}^+ = \tilde{\mathbf{c}}^{-1} \boldsymbol{\tau}^+ \quad \implies \quad \boldsymbol{\Theta}^+ = \mathbf{S} \tilde{\mathbf{c}}^{-1} \boldsymbol{\tau}^+ + d\boldsymbol{\Theta}^{(ad)}$$

Because particle temperatures are replaced by temperatures extrapolated from the grid, this approach removes null-space noise. Because inverting $\tilde{\mathbf{c}}$ remains infeasible, the approach here is to approximate $\tilde{\mathbf{c}}^{-1}$ by expanding it in a series and then keeping enough terms. Following full-mass matrix methods for MPM mechanics [11], $\tilde{\mathbf{c}}$ can be cast as:

$$\tilde{\mathbf{c}} = \mathbf{c} \mathbf{S}^\Theta \mathbf{S} = \mathbf{c} (\mathbf{I}_n - (\mathbf{I}_n - \mathbf{S}^\Theta \mathbf{S})) = \mathbf{c} (\mathbf{I}_n - \mathbf{A})$$

where $\mathbf{S}^\Theta = \mathbf{c}^{-1} \mathbf{S}^T \mathbf{C}$, $\mathbf{I}_n$ is an $n \times n$ identity matrix, and $\mathbf{A} = \mathbf{I}_n - \mathbf{S}^\Theta \mathbf{S}$. Note that $\mathbf{S}$ is a matrix of shape functions that maps grid temperatures to particles while $\mathbf{S}^\Theta$ is a matrix that maps particle temperatures to the grid as lumped nodal temperatures $\boldsymbol{\theta}^L = \mathbf{S}^\Theta \boldsymbol{\Theta}$. With perfect mapping, the product $\mathbf{S}^\Theta \mathbf{S}$ would be $\mathbf{I}_n$ and $\mathbf{A}$ would be zero. With suboptimal mapping caused by both non-square mapping matrices and lumped heat capacity matrix approximations in $\mathbf{S}^\Theta$, a non-zero $\mathbf{A}$ characterizes lumping error. When $\mathbf{A}$ is sufficiently small, the full heat capacity matrix inverse can be usefully expanded in a Taylor series that should converge fast:

$$\tilde{\mathbf{c}}^{-1} = (\mathbf{I}_n - \mathbf{A})^{-1} \mathbf{c}^{-1} = (\mathbf{I}_n + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \dots) \mathbf{c}^{-1}$$

We define $\tilde{\mathbf{c}}_k^{-1}$ as the full heat capacity matrix inverse expanded to k terms (last term is $\mathbf{A}^{k-1}$) and FMPM(k) as a new approach to MPM transport using $\tilde{\mathbf{c}}_k^{-1}$ in place of $\tilde{\mathbf{c}}^{-1}$. Expanding powers of $\mathbf{A}$ leads to [11]:

$$\tilde{\mathbf{c}}_k^{-1} = \left(k \mathbf{I}_n + \sum_{\ell=1}^{k-1} \sum_{j=\ell}^{k-1} (-1)^\ell \binom{j}{\ell} (\mathbf{S}^\Theta \mathbf{S})^\ell \right) \mathbf{c}^{-1} = \left(\sum_{\ell=1}^k (-1)^{\ell+1} \binom{k}{\ell} (\mathbf{S}^\Theta \mathbf{S})^{\ell-1} \right) \mathbf{c}^{-1} \quad (10)$$

Next define FMPM(k) grid temperatures using

$$\boldsymbol{\theta}(k) = \tilde{\mathbf{c}}_k^{-1} \boldsymbol{\tau}^+ = \left(\sum_{\ell=1}^k (-1)^{\ell+1} \binom{k}{\ell} (\mathbf{S}^\Theta \mathbf{S})^{\ell-1} \right) \boldsymbol{\theta}^{L+} \quad (11)$$

Particle update equations for FLIP and FMPM(k) become:

$$\boldsymbol{\Theta}^+ = \begin{cases} \boldsymbol{\Theta} + \mathbf{S} \mathbf{v}^{(\theta)} \Delta t + d\boldsymbol{\Theta}^{(ad)} & \text{FLIP} \\ \mathbf{S} \boldsymbol{\theta}(k) + d\boldsymbol{\Theta}^{(ad)} & \text{FMPM}(k) \end{cases} \quad (12)$$

FMPM(1) reduces to $\tilde{\mathbf{c}}_1^{-1} = \mathbf{c}^{-1}$ and $\boldsymbol{\theta}(k) = \boldsymbol{\theta}^{L+}$, which is equivalent to the PIC (for Particle In Cell) method [12] that replaces particle temperatures on each time step with new temperatures extrapolated from the grid using a lumped heat capacity matrix. Although PIC eliminates oscillations, it causes undesirable numerical

Table 1: Physical terms for transport equation when modeling transport of temperature (T), concentration potential ($\bar{c}$), pore pressure (σ), or a damage phase field (ξ). The notation used is: ρ is density, C_v is constant-volume heat capacity per unit mass, Q_B is Biot poroelasticity modulus [13], $\mathbf{k}$ is thermal conductivity tensor, $\mathbf{D}$ is diffusion tensor, $\mathbf{K}$ is Darcy's law permittivity, η is pore fluid or phase field viscosity, G_c is toughness, and ℓ is phase field length scale.

Property	Temp.	Conc. Pot.	Pore Press.	Phase Field
Content τ	Energy (J)	Volume (m^3)	Volume (m^3)	Energy Time (J·sec)
Value θ	T (K)	$\bar{c}$ (none)	σ (Pa)	ξ (none)
Capacity c_θ	ρC_v ($\frac{\text{J}}{\text{K}\cdot\text{m}^3}$)	1 (none)	$\frac{1}{Q_B}$ (Pa^{-1})	η (Pa·sec)
Diffusion κ	$\mathbf{k}$ ($\frac{\text{W}}{\text{m}\cdot\text{K}}$)	$\mathbf{D}$ ($\frac{\text{m}^2}{\text{sec}}$)	$\frac{\mathbf{K}}{\eta}$ ($\frac{\text{m}^2}{\text{Pa}\cdot\text{sec}}$)	$G_c \ell \mathbf{I}$ (N)

diffusion and must be rejected. Examples below show that FMPM(k) with $k > 1$ provides a method that eliminates problems in FLIP and, when done correctly, causes no significant numerical diffusion.

Equation (7) finds gradients on the particles using the full heat capacity matrix. In FLIP, this equation is replaced by $\nabla \Theta = \mathbf{G} \theta^L$. In FMPM(k), one is tempted to replace $\tilde{\mathbf{c}}^{-1}$ with $\tilde{\mathbf{c}}_k^{-1}$, but that approach does not converge unless the time step is reduced as order k is increased. Fortunately, a hybrid method using lumped temperatures to get particle gradients ($\nabla \Theta = \mathbf{G} \theta^L$) while using FMPM(k) to update particle temperatures both converges well and eliminates oscillations seen in FLIP.

This implementation is a large-deformation method provided particle heat capacity, volume, and thermal conductivity all adjust for the current particle deformation. The τ extrapolation conveniently reduces to $C_p = V_p C_{\Theta,p} = V_p \rho_p C_V = m_p C_V$ where ρ_p is current particle density, m_p is particle mass, and C_V is constant volume heat capacity in J/(kg·K). The $m_p C_V$ form is independent of particle deformation. Particle thermal conductivity should use current conductivity if it changes with deformation. Finally, source terms, $\dot{Q}_p$, and surface fluxes, s_i , should reflect current particle deformation. All examples in this paper are small deformation examples.

3. Other Transport, Boundary Conditions, and Coupling

The heat equation (Eq. (1)) can be generalized to modeling various other transport properties by generalizing the physical terms in the equation. Each transport process involves a conserved quantity, τ , connected to a transport value θ . The heat capacity changes to a transport capacity, c_θ , defining τ per unit change transport value per unit volume. The flux term is flow of τ in units m -(c_θ units)-(θ units)/sec and that flow is controlled by replacing thermal conductivity tensor, $\mathbf{k}$, with a transport “diffusion” tensor, κ , with units of m^2 -(c_θ units)/sec. The $\dot{q}(\mathbf{x})$ source term changes to units τ per unit volume per sec or units (c_θ units)-(θ units)/sec. For specific examples, Table 1 lists the changes from heat energy equation needed to model solvent diffusion, poroelasticity [13, 14], or a diffusive fracture phase field [15, 16]. These examples replace modeling of temperature with methods to track concentration potential ($\bar{c}$), pore pressure (σ), and a fracture phase field (ξ). Some elaboration on each one follows:

In solvent diffusion, the flow term, $\mathbf{q}(\mathbf{x})$, is found from the gradient in chemical potential, μ :

$$\mathbf{q}(\mathbf{x}) = -\frac{c}{RT} \mathbf{D} \nabla \mu$$

The chemical potential is:

$$\mu = \mu^\ominus + RT \ln a = \mu^\ominus + RT \ln \frac{\gamma c}{c_{ref}}$$

where $\mu^\ominus$ is chemical potential in a reference state where activity is $a = 1$ ($c = c_{ref}/\gamma$), and γ is an activity coefficient. Taking gradient of chemical potential, the diffusion equation transforms to

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(\frac{c_{ref}}{\gamma} \mathbf{D} \nabla \frac{\gamma c}{c_{ref}} \right) + \dot{q}(\mathbf{x})$$

If concentration is in mass (or moles) per unit volume, this equation is for flow of τ corresponding to conserved mass (or moles) of solvent. It is preferable, however, to recast this equation in terms of a dimensionless

concentration potential. If c_{ref} and γ are independent of position, diffusion analysis reduces to Eq. (1) with $\theta = \bar{c} = c/c_{ref}$ and $\mathbf{q}(\mathbf{x}) = -\mathbf{D}\nabla\bar{c}$ (keeping diffusion tensor $\mathbf{D}$ inside the divergence allows it to depend on position or on particle state, such as its deformation). The term $\bar{c}$ is referred to here as “concentration potential.” The conserved quantity τ becomes volume but it refers to reference volume related to conserved amount of solvent and to not material point volume that may expand or contract due to stresses or the presence of solvent. Modeled source terms must be converted to concentration potential per sec.

Additional changes are needed in problems where c_{ref}/γ within a single material model depends on $\mathbf{x}$ (note that if γ depends on concentration, it implicitly depends on position whenever concentration depends on position). Allowing c_{ref}/γ to depend on $\mathbf{x}$ and normalizing by a constant, baseline reference concentration, $c_{ref}^{(0)}$, gives:

$$\frac{\partial\bar{c}}{\partial t} = \nabla \cdot \left(\frac{c_{ref}(\mathbf{x})}{\gamma(\mathbf{x})c_{ref}^{(0)}} \mathbf{D} \nabla \frac{\gamma(\mathbf{x})c}{c_{ref}(\mathbf{x})} \right) + \dot{q}(\mathbf{x}) = \nabla \cdot \left(c_{rel} \mathbf{D} \nabla \frac{\bar{c}}{c_{rel}} \right) + \dot{q}(\mathbf{x}) \quad (13)$$

where $\bar{c} = c/c_{ref}^{(0)}$ is a revised concentration potential, $c_{rel} = c_{ref}(\mathbf{x})/(\gamma(\mathbf{x})c_{ref}^{(0)})$, $\mathbf{q}(\mathbf{x}) = -c_{rel}\mathbf{D}\nabla(\bar{c}/c_{rel})$ (*i.e.*, gradient of activity with a scaled diffusion tensor), and source terms must be converted to revised concentration potential per sec. Physically, $\bar{c}/c_{rel} = a(\mathbf{x})/c_{ref}(\mathbf{x}) = \bar{a}$ is an “activity potential.” Note that MPM implementation of Eq. (13) must extrapolate both $\bar{c}$ and $\bar{a}$ to the grid. The first is needed to track concentration potential while the later is only needed to find $\mathbf{q}(\mathbf{x})$.

The reason for using both γ and c_{ref} are to keep $\bar{c}$ dimensionless and to handle a common problem when modeling composites where $\gamma = 1$ in each phase. For example, when modeling moisture transport into composites, each material type could set c_{ref} to its saturation concentration. Equation (13) within each material type then reduces to the standard concentration potential equation based on each material’s c_{ref} and $\mathbf{D}$. Equilibrium would correspond to constant $\bar{c}$ when all phases are at the same fraction of their saturation concentration and flow stops because $\nabla\bar{c} \rightarrow 0$.

Early poroelasticity modeling noted an equivalence of the poroelasticity equation to the heat energy equation by redefining terms as in Table 1 [17]. This equivalence even extends to an “adiabatic” source term defined below. Like solvent diffusion, the flow quantity, τ , as volume is a stand-in for conserved amount of water in the material.

The phase field equation referenced in Table 1 is specific to variational fracture mechanics modeling [18, 19]. Prior finite element modeling [15, 16, 20] and MPM modeling [21–23] has implemented variational fracture mechanics modeling by coupling a diffusive fracture phase field to energy flow in the mechanics modeling. In the absence of energy driving fracture propagation, the phase field around any pre-existing crack should evolve to a narrow band with characteristic phase field length scale, ℓ . The full phase field diffusion equation has ubiquitous source terms in $\dot{q}(\mathbf{x})$ that control both a diffusive crack field’s shape and its propagation (see below). Phase field methods are also used for non-fracture methods, such as electrodeposition [24, 25], by changing properties and altering the source term to model different physics [25].

3.1. Transport Boundary Conditions

Transport analysis boundary conditions either impose transport values on the grid or set transport flux on particle surfaces. Grid boundary conditions are straight-forward, but require attention to detail to handle both FLIP and FMPM(k). The recommended steps are:

1. After extrapolating content to the grid and finding θ^L , copy θ_i^L for any node with a boundary condition and replace it with that node’s transport-value boundary condition $\theta_i^{(BC)}$. Find transport gradients with $\nabla\Theta = \mathbf{G}\theta^L$ using the modified grid values. The point of this step is to make sure gradients respond to grid boundary conditions. When modeling solvent diffusion with activity coefficients that depend on position this step changes to finding gradients of $\bar{a}$ — see the appendix for details. After getting gradients, restore copied θ_i^L to return θ^L to its initially extrapolated values.
2. After updating grid values to θ^{L+} , replace θ_i^{L+} for any node with a boundary condition with $\theta_i^{(BC)}$ and calculate the rate change this replacement implies from:

$$\Delta v_i^{(\theta)} = \frac{\theta_i^{(BC)} - \theta_i^{L+}}{\Delta t}$$

The net rate on each boundary condition node becomes:

$$v_i^{(BC)} = v_i^{(\theta)} + \Delta v_i^{(\theta)} = \frac{\theta_i^{(BC)} - \theta_i^L}{\Delta t}$$

which is the rate needed to move the initially extrapolated value to the boundary condition value. These corrected rates are needed because a FLIP update extrapolates grid rates to the particles.

3. Grid boundary conditions cause transport flow into or out of the object to maintain grid values. The net flow at each boundary condition is potentially useful information and given by:

$$Q_1^{(BC)} = c_i^L \Delta v_i^{(\theta)}$$

or the net “force” needed to convert θ_i^{L+} to $\theta_i^{(BC)}$.

4. When using FMPPM(k) for $k > 1$, $\theta(k)$ is calculated from θ^{L+} after step 2, but these $\theta(k)$ may causes a change at nodes with boundary conditions. To fix this issue, replace $\theta_i(k)$ for any node with a boundary condition with $\theta_i^{(BC)}$ and calculate a new net flow:

$$Q_2^{(BC)} = c_i^L \frac{\theta_i^{(BC)} - \theta_i(k)}{\Delta t}$$

No rate adjustments are needed because subsequent FMPPM(k) updates do not use them.

5. Net flow at a collection of boundary conditions (such as one edge of a sample set to constant value) is found by summing $Q_1^{(BC)}$ for FLIP or FMPPM(1) or $Q_2^{(BC)}$ for FMPPM($k \geq 2$).

Transport flux boundary conditions are applied on particle surfaces by the same methods used for traction boundary conditions in MPM mechanics [6] except traction $\mathbf{T}$ is replaced by a scalar flux $\Phi = \mathbf{q}(\mathbf{x}) \cdot \hat{\mathbf{n}}$ where $\hat{\mathbf{n}}$ is normal to the particle surface. $\Phi > 0$ or $\Phi < 0$ is for inward or outward flux. Surface fluxes translate into nodal fluxes by:

$$s_i = \sum_{\partial p} \Phi_{p,s} \int_{A_s} N_i(\mathbf{x}) dS \quad (14)$$

where $\partial p = \{p, s : \text{particle } p \text{ has flux } \Phi_{p,s} \text{ on surface } s\}$ and A_s is the deformed area for surface s of particle p ($s = 1$ to 4 surfaces for 2D particles or 1 to 6 surfaces for 3D). Reference [6] explains how to integrate shape functions over particle surfaces. A large deformation implementation should integrate over deformed surfaces.

3.2. Coupling to Mechanics

The motivation for solving transport equations in MPM is not to solve transport equations alone, which can be done by many methods, but rather to couple transport quantities to MPM computational mechanics. Three coupling mechanisms are:

1. Mechanisms where particle state changes transport value. These terms are best modeling as particle source terms in $d\Theta^{(ad)}$.
2. Mechanisms where transport value affects particle mechanics (*e.g.*, strains induced by temperature changes) or affects other coupled transport phenomena.
3. Modeling materials whose properties depend on transport value (*e.g.*, moisture dependent wood properties).

Type #1 coupling occurs whenever particle deformation state induces a change in transport value. Such coupling terms are handled by adding to particle source term $\dot{Q}_p^{(p)}$. For example, volume changes in elastic materials cause temperature to change. Poroelastic materials have a corresponding phenomenon where volume changes cause pore pressure to change. For isotropic thermoelastic and poroelastic materials, these coupling terms are:

$$\dot{Q}_p^{(p)} \Delta t = -\alpha_V K_{bulk} T_p \frac{\Delta V}{V} \quad \text{and} \quad \dot{Q}_p^{(p)} \Delta t = -Q_B \alpha_B \frac{\Delta V}{V} \quad (15)$$

where α_V is volumetric thermal expansion coefficient, K_{bulk} is bulk modulus, T_p is current particle temperature, α_B is Biot–Willis fluid/solid coupling coefficient, and $\Delta V/V$ is volumetric strain increment. Besides, these reversible terms in coupled thermoelasticity or poroelasticity, modeled materials might generate transport property changes by other mechanisms. Some examples are temperature changes due to dissipated energy by plasticity, frictional contact [4], or crack tip propagation.

In fracture phase field modeling, Eq. (1) captures the divergence term while the rest of the phase-field evolution equation is interpreted as a particle source term given by:

$$\dot{Q}_p^{(p)} \Delta t = -\frac{G_c}{\ell} \xi - g'(\xi) H_p \quad (16)$$

where G_c is toughness, ℓ is phase-field length scale, and H_p is an energetic history variable tracked on each particle that promotes fracture [15, 16]. The transport property is the phase field ξ that varies from 0 for undamaged material to 1 for a crack surface. In the absence of input H_p energy, the steady-state fracture phase equation reduces to:

$$\frac{G_c}{\ell} (\ell^2 \nabla^2 \xi - \xi) = 0$$

Solving this equation, ξ decays exponentially from 1 on crack planes to zero with exponential decay rate of $1/\ell$ and cracks remain static. With loading, non-zero H_p provides an energy source that promotes crack propagation while $g(\xi)$ is a “softening law” that models degradation in mechanical properties. $g(\xi)$ is typically set to $g(\xi) = (1 - \xi)^2$ [15, 16] because linear $g'(\xi) = -2(1 - \xi)$ facilitates linear finite element methods. Allowing $g(\xi)$ to have nonlinear forms enables modeling of materials with new softening behaviors. Fortunately, the transport approach to phase field modeling neither requires nor benefits from a linear $g'(\xi)$.

Particle source terms $\dot{Q}_p^{(p)}$ could be treated instead as $\dot{Q}_p^{(g)}$ and extrapolated to the grid in Eq. (6). Such extrapolation, however, would cause some non-physical diffusion to nearby particles. A better approach is to store them for addition to particle values during particle updates. The increments needed by updates (see Eq. (12)) can be cast as:

$$d\Theta_p^{(ad)} = \sum_{\text{source terms}} \frac{\Delta t}{c_\theta} \left(\dot{Q}_p^{(p)}(\text{reversible}) + \dot{Q}_p^{(p)}(\text{irreversible}) \right) \quad (17)$$

where (ad) standards for “adiabatic” source terms. In general transport analysis, “adiabatic” means change in transport value in the absence of flux. In heat conduction, it means temperature change in the absence of heat flow. In poroelasticity, it means a pressure change without fluid transport, which is called an “undrained” response. Separation into “reversible” and “irreversible” source terms is only relevant for heat conduction and used below for entropy calculations. Other transport modeling would not use this separation.

Type #2 coupling is when a change in transport value causes expansion or shrinkage leading to residual stresses. For heat conduction, solvent diffusion, and poroelasticity, linear residual strain increments are:

$$\Delta \varepsilon_{\text{thermal}} = \alpha \Delta T, \quad \Delta \varepsilon_{\text{solvent}} = \beta c_{ref} \Delta \bar{c} \quad \text{and} \quad \Delta \varepsilon_{\text{pore pressure}} = \frac{\alpha_B}{3K_{bulk}} \Delta \sigma \quad (18)$$

where α and β are linear thermal and solvent expansion coefficients and c_{ref} accounts for diffusion modeling using a concentration potential $\bar{c}$. The transport property increments needed for these strains (*e.g.*, ΔT , $\Delta \bar{c}$, and $\Delta \sigma$) can be determined in the particle update. But, in FLIP methods, particle value oscillations would make such increments inaccurate. A method required for FLIP is for each particle to track two transport values — one derived in the FLIP update and a “smoothed” value derived by extrapolating lumped grid values (adjusted for boundary conditions) to the particles:

$$\bar{\Theta}^+ = \mathbf{s}\theta^{L+} \quad (19)$$

Whenever particle transport value is needed, calculations should use the smoothed transport value [6]. For example, the transport value increment between two time steps should be found from:

$$\Delta \Theta = \bar{\Theta}^+ - \bar{\Theta}$$

instead of from particle transport values. Finally, to model residual stresses caused by initial particle values differing from a reference value Θ_{ref} (such as a stress-free temperature in conduction modeling), $\bar{\Theta}$ should

be initialized to Θ_{ref} on each particle rather than the particle's initial value. By this approach, $\Delta\Theta$ for first time step is the transport value change between reference conditions and the initial particle value. In explicit, dynamic simulations, applying such a potentially large change in the first time step might cause numerical artifacts. A better approach is to start all particles at Θ_{ref} and then slowly increment each particle's value to desired initial values in a separate task.

Although smoothed transport value tracking is essential for FLIP, FMPM(k) does not need it. In FMPM(k), smoothed values change to $\bar{\Theta}^+ = \mathbf{S}\theta(k) = \Theta^+$; in other words “smoothed” values and particle values are equal. Similarly, transport increments can be found from particle transport values: $\Delta\Theta = \bar{\Theta}^+ - \bar{\Theta} = \Theta^+ - \Theta$.

Transport increments are found during the particle update, but they are not used until the next particle constitutive law calculations. They can be stored on each particle until then. For small-strain, linear elastic materials, an incremental stress-strain, constitutive law is

$$d\boldsymbol{\sigma} = \mathbf{C}(d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}_{res}) = \mathbf{C}d\boldsymbol{\varepsilon}_{eff}$$

where $\mathbf{C}$ is the stiffness tensor. In other words, incremental strains due to transport properties are subtracted from the current step's incremental strain to find an *effective* strain increment (an effective strain approach easily extends to small-strain, plasticity materials as well). Large-deformation analysis can divide deformation gradient, $\mathbf{F}$, into two steps — deform by transport-induced strains and then by mechanical loading.

Type #3 coupling is when modeling materials whose properties depend on transport value (*e.g.*, temperature-, solvent-, or damage-dependent properties) or its increment for other calculations (*e.g.*, energy dissipation in phase-field fracture materials). Modeling such materials simply requires finding those properties as a function of the current particle transport value and its increment. To avoid numerical artifacts caused by transport value oscillations, transport-dependent mechanical properties for FLIP must use smoothed particle value $\bar{\Theta}$ (see Eq. (19)). FMPM(k) can use particle values.

3.3. Thermodynamics (Heat Energy Only)

Good heat energy modeling can track thermodynamic state functions. Each particle should track work energy, W_p , heat energy, Q_p (not to be confused with $\mathbf{Q}$ fluxes or $\dot{Q}_p$ sources), and entropy S_p (all per unit mass to best handle large deformation). Work energy increment is $dW_p = (1/\rho)\boldsymbol{\sigma} \cdot \nabla\mathbf{v}\Delta t$ where $\nabla\mathbf{v}$ is velocity gradient in each time step. Heat energy and entropy are updated as follows:

1. Particle sources are divided into isoentropic (no heat flow or $q = 0$) reversible temperature changes:

$$dT_{p,q=0} = \frac{\dot{Q}_p^{(p)}(\text{reversible})\Delta t}{c_\theta}$$

and irreversible energy dissipation $d\Phi_p = (1/\rho)dQ_p^{(n)}$ (irreversible). When these changes occur, $dT_{p,q=0} + d\Phi_p/C_v$ is added to $d\Theta_p^{(ad)}$ in Eq. (17), and the particle entropy change is due only to the irreversible temperature increase:

$$\Delta S_p = C_v \ln \frac{\bar{T}_p + \frac{d\Phi_p}{C_v}}{\bar{T}_p}$$

This calculation uses smoothed $\bar{T}_p$ for FLIP or particle value for FMPM(k). These local, adiabatic changes correspond to zero change in particle heat energy ($dQ_p = 0$). A particle experiences no heat energy change until it exchanges heat with neighboring particles by conduction.

2. During particle updates, particle heat energy increment due to conduction is found from $dQ_p = C_v dT_p^{(cond)}$ where $dT_p^{(cond)} = \bar{T}_p^+ - \bar{T}_p$ is “smoothed” temperature change due to heat exchange between particles by conduction only (or just particle temperature change when using FMPM(k)). The entropy increment due to conduction alone is:

$$\Delta S_p = C_v \ln \frac{\bar{T}_p^+}{\bar{T}_p} \quad (20)$$

Note that $dT_p^{(cond)}$ on the first time step should be relative to the particle's initial temperature ($T_p^{(i)}$) and *not* the stress-free temperature (T_0) resulting in heat and entropy change from initial particle state. To convert to heat and entropy change relative to all particles at T_0 , each particle should be initialized with heat and entropy needed to change them from T_0 to $T_p^{(i)}$.

This approach relies on all adiabatic heating being in $d\Theta_p^{(ad)}$ (*i.e.*, all source terms in MPM heat conduction modeling must be in $\dot{Q}_p^{(p)}$ and must have $\dot{Q}_p^{(g)} = 0$). When true, particle temperature change $dT_p^{(cond)}$ in step #2 above is temperature change resulting only from heat exchange between particles. This temperature change corresponds to changes in heat energy. Once that calculation is done the adiabatic change, dT_p^{ad} , is added to particle temperature and by definition that change has no effect on heat energy.

MPM simulations coupled to conduction using the above tracking steps provides full-physics modeling of coupled momentum and heat equations. If the modeled structure is adiabatic (*i.e.* exchanges no heat with the exterior, which in MPM means no thermal boundary conditions), total heat energy will be zero and all entropy will be irreversible entropy. If a simulation has thermal boundary conditions, total heat energy may increase or decrease due heat exchange with the exterior. Following Kondepudi and Prigogine [26], an entropy increment can be written as $dS = d_e S + d_i S$ where $d_e S$ is entropy change due to heat exchange with the exterior (it is reversible and can be positive or negative) and $d_i S$ is entropy change in the interior (it is irreversible and can only be positive). MPM can track and spatially resolve both reversible and irreversible entropy. Total reversible entropy may increase or decrease, but irreversible entropy can only increase.

Isothermal calculations are done by turning off conduction calculations, setting all particles to constant temperature T_0 , revising step #1 in thermodynamics tracking, ignoring step #2 (which never occurs when conduction is off), and calculating $dT_{p,q=0}$ and $d\Phi_p$ but never adding them to particle temperature. When a material model finds $dT_{p,q=0}$ and $d\Phi$, change particle heat energy by using $dQ_p = -C_v dT_{p,q=0} - d\Phi_p$. In effect, this tracking assumes temperature increases are expelled to the exterior as heat and decreases are compensated by heat input from the exterior. Because dQ_p is exchanged with the exterior, it contributes to reversible entropy, but $d\Phi_p$ causes irreversible entropy. Total particle entropy change becomes:

$$dS_p = d_e S + d_i S = \frac{dQ_p}{T_p} + \frac{d\Phi_p}{T_p} = -\frac{C_v dT_{p,q=0}}{T_p}$$

Isothermal modeling in MPM, or in other methods, implies that heat energy instantaneously exchanges with the exterior. This assumption is unrealistic for any real material modeled on typical dynamic-mechanics simulation time scales. Nevertheless, isothermal methods provide simplified modeling (*i.e.*, no need to solve the heat equation) and allow for comparison to analytical solutions (which usually ignore temperature changes).

By tracking W_p , Q_p , and S_p , a simulation can track and spatially resolve other thermodynamics functions. For example, internal energy would be $U_p = W_p + Q_p$ and Helmholtz free energy would be $A_p = U_p - T_p S_p = W_p + Q_p - T_p S_p$ (note that for isothermal, $dA_p = dW_p - d\Phi_p$). In general, these tracked thermodynamics values have no influence on MPM simulations, but they may have uses. Two examples are the Mie-Grüneisen equation of state [27], where the constitutive law depends on internal energy, and phase field fracture, where Helmholtz free energy tracks dissipated fracture energy well.

3.4. Transport-Specific Details

Each transport process corresponds to a physical process that might impose some restrictions on the transport solution. This section considers three transport-specific details.

Concentration potential should never go negative and updates should therefore disallow negative values. When modeling a composite with c_{ref} set to each phase's saturation concentration, concentration potential becomes a relative concentration. It should never exceed 1 and updates should enforce that limit. When c_{ref} is an arbitrary reference concentration, concentration potential may have any non-negative value.

Poroelasticity derived by Biot [13] models materials saturated in fluid (usually water) where changes in volume induce changes in pore pressure and pore pressure is always positive. The above transport modeling, however, does not restrict transport values to positive numbers. Pore pressure could, for example, turn negative in problems where outward-directed flux boundary conditions are given enough time to drain all fluid. Because partially-saturated materials should have zero pore pressure (and not negative pressure), special handling is needed. These changes appear to work:

1. Solve the pore pressure transport equation, including the particle source term in Eq. (15), for positive and negative σ .
2. Whenever $\sigma > 0$, it corresponds to pore pressure and changes in pore pressure ($\Delta\sigma$) cause residual strains by Eq. (18).
3. Whenever $\sigma < 0$, the pore pressure is zero and $\Delta\sigma$ is set to zero to eliminate residual strain coupling. Drained space volume while $\sigma < 0$ can be approximated by $-\sigma/Q_B$.

A fracture phase field tracks damage state that can only increase and cannot exceed fully cracked state of $\xi = 1$. If an update causes phase field to decrease, that change is ignored or phase field is left unchanged. If an update exceeds 1, it is truncated to 1.

3.5. Disparate Time Steps

When simultaneously modeling mechanics and transport by explicit updates, the time step must be set to minimum of the mechanics (Δt_M) and any transport time steps (Δt_Θ):

$$\Delta t = \min(\Delta t_M, \Delta t_\Theta) \quad \text{where} \quad \Delta t_M = f_M \frac{\Delta x}{v_{max}} \quad \text{and} \quad \Delta t_\Theta = f_\Theta \frac{\Delta x^2}{2} \frac{C_\Theta}{\kappa_{max}}$$

Here Δx is cell size, v_{max} and κ_{max} are the maximum wave speed and diffusion tensor in the problem, f_M is a CFL time-step factor for mechanics equations, and f_Θ is a time-factor applied to von Neumann stability criterion for the transport equation (f_M and f_Θ must be less than 1 for convergence). The time step ratio is:

$$R_{\Delta t} = \frac{\Delta t_\Theta}{\Delta t_M} = \frac{f_\Theta}{f_M} \frac{v_{max} C_\Theta \Delta x}{2 \kappa_{max}} \quad (21)$$

Unless Δx is very small, typical modeling will have $v_{max} \gg 2\kappa_{max}/(C_\Theta \Delta x)$, $R_{\Delta t} > 1$, and $\Delta t = \Delta t_M$. Because transport will be using the same time step, its effective f_Θ becomes:

$$f_\Theta^{(eff)} = \Delta t_M \frac{2\kappa_{max}}{\Delta x^2 C_\Theta} \quad (22)$$

3.6. Computational Cost

A task-oriented transport analysis process for MPM is in the appendix. Each task loops over particles and thus computational cost for FLIP updates is NC_0 where N is number of particles and C_0 is the complete time step cost per particle. For FMPM(k) updates, each time step must additionally evaluate Eq. (10), which requires an additional $k - 1$ particle loops for a computational cost of $N(k - 1)C_x$ where C_x is the time per particle for one FMPM(k) loop. The total FMPM(k) calculation time relative to FLIP is thus:

$$\text{Relative Time} = 1 + (k - 1)F \quad (23)$$

where $F = C_x/C_0$ is the fraction of a time step spent within one FMPM(k) loop.

4. Results and Discussion

4.1. Heat Conduction

Large oscillations and their elimination are demonstrated by this simple problem in heat conduction: consider a 1 m^3 cube with material left of the centerline starting at $T_L = 250\text{K}$ and material to the right starting at $T_R = 350\text{K}$. Conduction within this cube was modeled by 2D MPM using 50 mm cells, four particles per cell, and transport time step factor $f_\Theta = 0.5$. This analysis used a “transport-only” mode that solves only the heat conduction equation. The only isotropic material properties needed were $k = 1000 \text{ W/(m-K)}$, $\rho = 1 \text{ g/cm}^3$, and $C_v = 1 \text{ J/(kg-K)}$. Figure 1 shows the temperature profile after 200 ms. The solid, square symbols show severe oscillations in particle values near the initial interface when using FLIP. Despite these poor particle values, FLIP grid values are smooth (see solid “Grid” line in Fig. 1).

FLIP oscillations in particle temperatures are caused by a disconnect between grid temperatures and rates and particle temperatures and rates. The reason oscillations can persist while maintaining a good grid solution is null-space noise made possible by the non-square mapping from particles to the grid (see Eq. (4)).

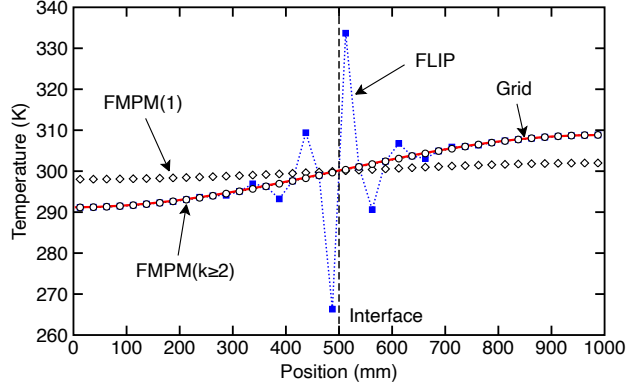


Figure 1: Temperature profile after 200 ms for a 1000 mm×1000 mm box with material left of the centerline starting at $T = 250\text{K}$ and material to the right starting at $T = 350\text{K}$. The symbols are particle temperatures for FLIP (filled squares), FPM(1) (diamonds), or FPM($k \geq 2$) (circles) simulations. The solid line labeled “Grid” plots smoothed or grid temperature by FLIP.

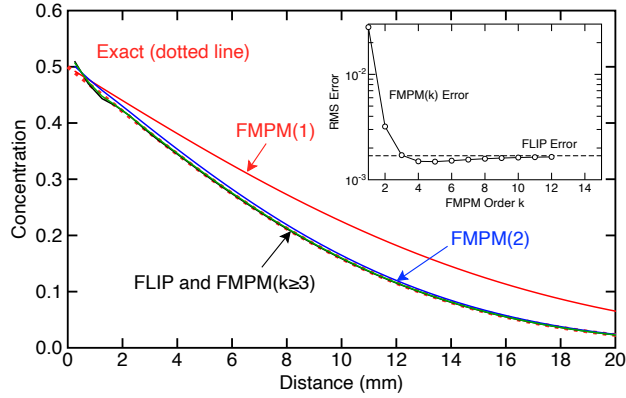


Figure 2: Diffusion after 5 seconds into an 80×10 mm strip with $c = 0.5$ at $x = 0$ and $D = 10 \text{ mm}^2/\text{sec}$. The curves are MPM simulations done using FPM(1), FPM(2), FPM($k > 3$), and FLIP. The dotted line is the analytical solution. The inset plot shows RMS error for FPM(k) as a function of k (circles) compared to FLIP error (horizontal dashed line).

FPM(k) was derived to approximate the full capacity matrix. An alternate interpretation is that it filters out null-space noise by replacing particle values on each time step with new values extrapolated from the grid (see Eq. (12)). Indeed, FPM(1) (open, diamond symbols in Fig. 1) eliminated all oscillations, but unfortunately caused numerical diffusion leading to results with too much thermal conduction. Fortunately, FPM(k) for any $k \geq 2$ (open circles in Fig. 1) eliminated oscillations while avoiding unwanted numerical diffusion.

This example provides a useful check on heat and entropy tracking. By running long enough, all particles approached $T_f = 300 \text{ K}$. This irreversible heat flow has zero net heat energy (heat is transferred from one side to the other, but none is exchanged with the exterior). Total entropy, however, should increase due to heat flow from the right to the left:

$$\Delta S = \Delta S_L + \Delta S_R = \rho C_v \frac{V}{2} \ln \left(\frac{T_f}{T_L} \right) - \rho C_v \frac{V}{2} \ln \left(\frac{T_f}{T_R} \right) = \frac{1}{2} \rho C_v V \ln \left(\frac{T_f^2}{T_L T_R} \right) = 14.0854 \text{ J/K}$$

When using FPM($k \geq 2$), the above entropy tracking methods returned this value to 6 significant figures. In contrast, FLIP underestimated entropy change by 5.3%.

4.2. Diffusion

The following diffusion example evaluates accuracy, numerical diffusion, and convergence by comparing simulations to an analytical solution. Consider “transport-only” analysis of diffusion into an 80×10 mm strip

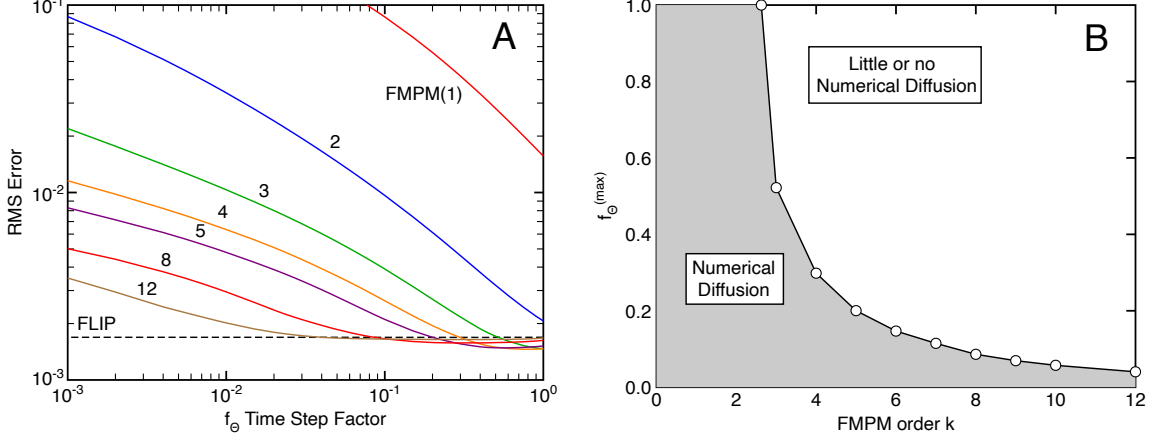


Figure 3: A. RMS error for diffusion calculations as a function of f_Θ and FMPM(k) for k from 1 to 12. The horizontal line is the FLIP RMS error, which was independent of f_Θ . B. Plot of $f_\Theta^{(max)}(k)$ defined as f_Θ when FMPM(k) and FLIP RMS errors are equal in A.

using 1 mm cells, four particles per cell, and transport time step factor $f_\Theta = 0.5$ for an isotropic material with $D = 10 \text{ mm}^2/\text{sec}$ and $c_{ref} = \gamma = 1$. All particles started at zero concentration. Grid boundary conditions set $\bar{c} = 0.5$ on the left edge ($x = 0$) and $\bar{c} = 0$ on the right edge ($x = 80$). The numerical diffusion solution was compared to an analytical solution:

$$c(x, t) = 0.5 \left(\xi + 2 \sum_{n=1}^{\infty} \frac{(-1)^n}{n\pi} e^{-\lambda_n^2 t} \sin(n\pi\xi) \right) \quad (24)$$

where $\xi = (80 - x)/L$ and $\lambda_n = (n\pi/L)\sqrt{D}$.

Figure 2 shows concentration as a function x after 5 seconds for FLIP, FMPM(1), FMPM(2), and FMPM($k \geq 3$). The dotted line plots the analytical solution summed to 20 terms. FMPM(1) (or PIC) caused significant numerical diffusion and is unacceptable. Both FLIP and FMPM($k > 1$) were close to the analytical solution. The inset plot shows RMS error

$$\text{RMS error} = \sqrt{\frac{1}{N_p} \sum_{p=1}^{N_p} (c_p - c(x_p, 5))^2}$$

for FMPM(k) as a function of order k . Here N_p is number of particles, c_p is particle concentration, and $c(x_p, 5)$ is theoretical result from Eq. (24) evaluated at particle location for $t = 5$ sec. The horizontal line is RMS error using FLIP. Using just FMPM(2) reduced the error compared to FMPM(1) by an order of magnitude. All FMPM($k \geq 3$) were more accurate than FLIP

The above results using $f_\Theta = 0.5$ and FMPM($k \geq 3$) eliminated numerical diffusion. But when MPM calculations are coupled to mechanics calculations, typical properties lead to $\Delta t_M \ll \Delta t_\Theta$ resulting in small $f_\Theta^{(eff)}$. To check for numerical diffusion at small f_Θ , Fig. 3A plots RMS error as a function of f_Θ for FLIP and FMPM(k) for k from 1 to 12. A decrease in f_Θ means that a 5 second simulation has more time steps. Even if FMPM(k) causes only a small amount numerical diffusion on each time step, with enough time steps, FMPM(k) simulations can degrade compared to non-diffusive (albeit oscillation-prone) FLIP simulations. Figure 3A shows that FMPM(1) (or PIC) and FMPM(2) are more diffusive than FLIP for $f_\Theta \leq 1$. FMPM(3) to FMPM(12) improve on FLIP, but cross over FLIP if f_Θ is too small. Figure 3B plots $f_\Theta^{(max)}(k)$ defined as f_Θ when FMPM(k) and FLIP RMS errors are equal. This plot provides a map for acceptable f_Θ values. The region left of the curve will show numerical diffusion while the region to the right will show negligible diffusion (assuming FLIP represents negligible numerical diffusion).

The question remains — how should FMPM(k) be used for transport modeling when $\Delta t_M \ll \Delta t_\Theta$? The answer is determined by calculating $f_\Theta^{(eff)}$ in Eq. (22). If $f_\Theta^{(eff)} > f_\Theta^{(max)}(k)$, then FMPM(k) calculations can

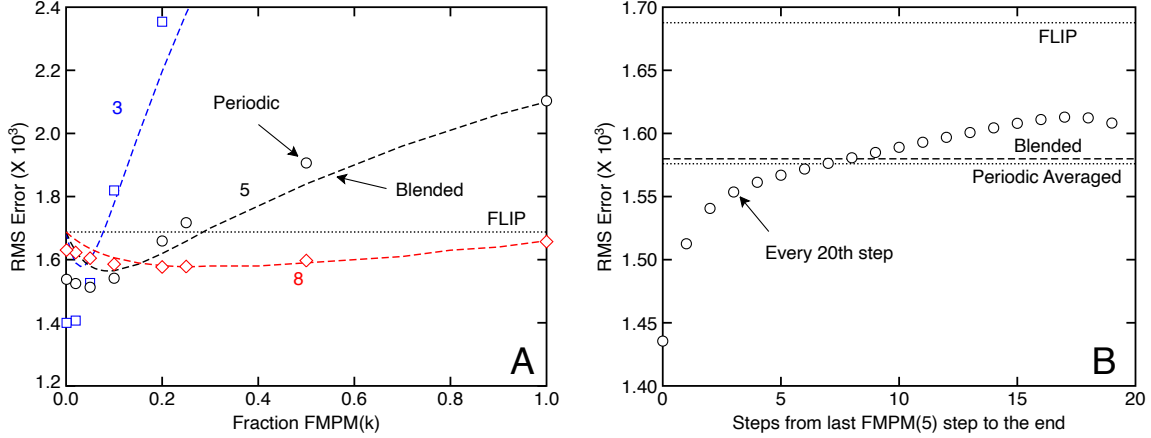


Figure 4: A. RMS error for diffusion calculations as a function of fraction FMPM(k), ψ , for $k = 3, 5$, or 8 . The symbols are for periodic FMPM(k) every $m = 1/\psi$. The dashed lines used blended FLIP/FMPM(k). The horizontal line is the FLIP RMS error. B. RMS error for periodic FMPM(5) using $m = 20$ as a function of number of time steps between the last FMPM(5) time step and the end of the simulation. The horizontal lines gives RMS errors for FLIP, blended FLIP/FMPM(5) with $\psi = 0.05$, and averaged for all periodic FMPM(5) simulations.

be done every time step, but if $f_{\Theta}^{(eff)} < f_{\Theta}^{(max)}(k)$, FMPM(k) calculations should change. Two approaches are to use FMPM(k) calculations periodically (use them every m^{th} time step while all other time steps use FLIP) or to blend FLIP and FMPM(k) (using a fraction ψ of FMPM(k) on every time step). Both methods define a third time-step factor, $f_X \geq f_{\Theta}^{(max)}(k)$. For periodic FMPM(k), the step interval would be $m \gtrsim \text{int}(f_X/f_{\Theta}^{(eff)})$; for blending, the fraction FMPM(k) would be $\psi \lesssim f_{\Theta}^{(eff)}/f_X$. Blending FLIP and FMPM(k) requires some modifications. First, particle updates and the separately-tracked “smoothed” values become:

$$\begin{aligned}\Theta^{(+*)} &= \psi \Theta^+ + (1 - \psi)(\Theta + \mathbf{S} \mathbf{v}^{(\theta)} \Delta t) + d\Theta^{(ad)} \\ \bar{\Theta}^{(+*)} &= \psi \bar{\Theta}^+ + (1 - \psi) \bar{\Theta}^+\end{aligned}$$

Second, transport value increments are found from the revised smoothed values ($\Delta \Theta = \bar{\Theta}^{(+*)} - \bar{\Theta}^{(*)}$). Third, implement grid boundary conditions as explained above being sure to only adjust rates when modifying θ^L . If net flow at boundary conditions is needed, the above flow terms are blended:

$$Q_{tot}^{(BC)} = (1 - \psi)Q_1^{(BC)} + \psi Q_2^{(BC)}$$

Figure 4A compares RMS error for periodic FMPM(k) (symbols) and blended FLIP/FMPM(k) (dashed lines) when using $f_{\Theta}^{(eff)} = 0.1$ and $k = 3, 5$, or 8 . These simulations varied f_X from 0.1 to 100 ; the results are plotted as a function of $\psi = 1/m = 0.1/f_X$. For $k = 3$ or 5 the errors were higher than FLIP when fraction FMPM(k) is too high, but became lower for $\psi < 0.08$ ($f_X > 1.25$) or $\psi < 0.25$ ($f_X > 0.4$), respectively. The transitions to lower errors were when f_X was higher than $f_{\Theta}^{(max)}(3)$ and $f_{\Theta}^{(max)}(5)$ in Fig. 3B. For $k = 8$ the errors were always lower than FLIP because $f_{\Theta}^{(max)}(8) \sim f_{\Theta}^{(eff)} = 0.1$ in these calculations. Although using high k works, it is computationally expensive. The best practice is to use low $k \geq 2$ and pick f_X to eliminate unwanted diffusion.

Periodic and blended methods were similar, but not identical. One reason is that these simulations used exactly 1001 time steps and periodic FMPM(k) calculations choose m to always be divisor of 1000. Thus, all periodic results had an FMPM(k) step on the penultimate time step. Each FMPM(k) time step replaces all particle transport values with filtered values that can remove oscillations even when done infrequently. When RMS error is calculated soon after this replacement, it is low but the intervening FLIP steps may cause it to increase. Figure 4B compares RMS error for FMPM(5) done every 20 times steps, but with the FMPM(5) time steps shifted to vary the number of time steps between the last FMPM(5) time step and the end of the simulation. The error was smallest for shift of zero and increased as the shift increased. The

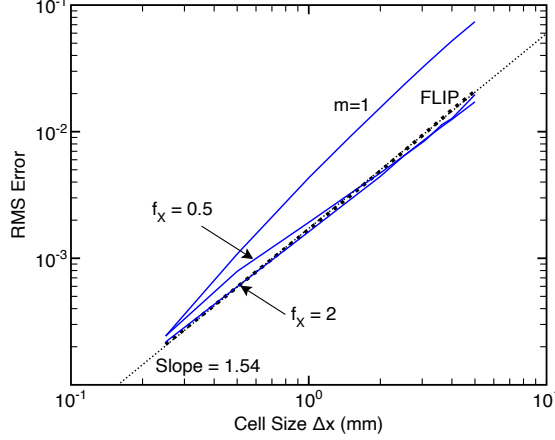


Figure 5: Spatial convergence of diffusion calculations for FLIP (dots), FMPM(4) done every time step ($m = 1$) or periodic FMPM(4) using $f_X = 0.5$ or $f_X = 2$. All calculations used a constant time step $\Delta t = 1.5625$ ms. The dotted line is power-law slope of 1.54.

“Periodic Averaged” line is average error for all possible shifts and it was nearly identical to the blended FLIP/FMPM(5) error using $\psi = 0.05$. Both periodic or blended methods had lower errors than using FLIP alone.

Based on Fig. 4, periodic FMPM(k) and blended FLIP/FMPM(k) appear equally effective when $f_{\Theta}^{(eff)} < f_{\Theta}^{(max)}(k)$ providing one uses $f_X > f_{\Theta}^{(max)}(k)$. An advantage of periodic FMPM(k) is increased efficiency because the computational cost in Eq. (23) changes to $1 + (k - 1)F/m$ (*i.e.* computational cost of FMPM only occurs every m^{th} time step). In contrast, blended FLIP/FMPM(k) incurs full computation cost of FMPM regardless of whether $\psi = 1$ or $\psi = 0.01$. A disadvantage of periodic FMPM(k) is that error varies with a large decrease happening after each FMPM(k) time step. Simulations that are sensitive to changes in transport value ($\Delta\Theta$) could be adversely affected by periodic FMPM(k). Using blended FLIP/FMPM(k) is a potential alternative to eliminate those issues.

Neither FLIP nor FMPM(k) benefit from time steps below shorter than needed for convergence. Both methods, however, converge further by reducing cell size as shown in Fig. 5. These calculations repeated the above diffusion calculations using a constant time step $\Delta t = 1.5625$ ms based on $f_{\Theta} = 0.5$ for the smallest cell size of $\Delta x = 0.25$ mm. For larger cells, the time step was kept constant by setting $f_{\Theta}^{(eff)} = 0.5 * (0.25/\Delta x)^2$, which decreased to $f_{\Theta}^{(eff)} = 0.00125$ for 5 mm cells. FLIP (dotted line) converged in proportional to $\Delta x^{1.54}$. The three solid lines correspond to FMPM(4) calculations every time step ($m = 1$) or to periodic FMPM(4) using $f_X = 0.5$ or $f_X = 2$. When done every time step, calculations for large cell size have too small a time step and therefore RMS error increased. Using periodic FMPM(k) resolved this issue resulting in convergence similar to FLIP. These calculations were repeated using blending with $\psi = f_{\Theta}^{(eff)}/f_X$ and the results were virtually identical to the periodic method.

4.3. Poroelasticity

Figure 6 shows a Biot column containing a porous material saturated with fluid and surrounded on two sides by impenetrable walls and on the bottom by a porous wall. This column was compressed with an impenetrable piston to constant displacement or constant load and then held. If the bottom wall is kept closed, this compression causes pore pressure to increase. If the bottom wall is opened after compressing, fluid can escape. At constant displacement, axial compression stress should decrease while at constant load, plunger displacement should increase. This section models Biot column experiments on Ruhr sandstone filled with water. Unlike previous “transport-only” examples, poroelasticity modeling is always coupled to mechanics modeling.

Ruhr sandstone modeled as an isotropic, poroelastic material has three mechanical properties — bulk modulus $K = 13$ GPa, Poisson’s ratio $\nu = 0.12$, and density $\rho = 2.5$ g/cm³ — and four poroelasticity properties — undrained bulk modulus $K_u = 30$ GPa, Biot-Willis coefficient $\alpha_B = 0.65$, Darcy’s law permittivity

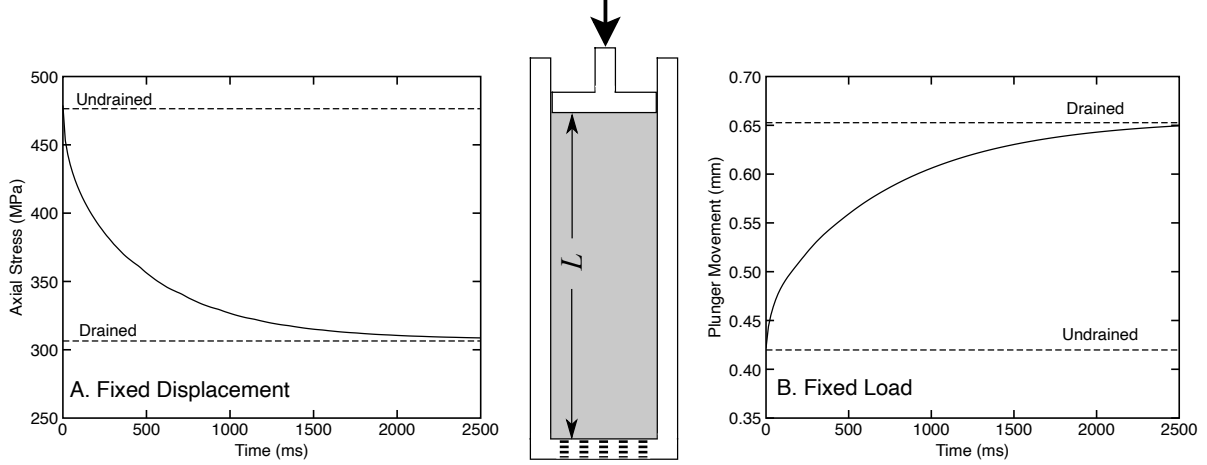


Figure 6: The middle shows a Biot column that compresses a porous material with a plunger while a porous bottom allows fluid to escape. A. Average axial compression stress in the Ruhr sandstone for 100×40 mm column compressed by 1 mm and then held. B. Plunger displacement for the same column compressed with constant compression stress of 200 MPa.

$k = 0.2$ millidarcys, and pore water viscosity $\eta = 1$ cP [14]. The Biot modulus used in transport equations is $Q_B = (K_u - K)/\alpha_B^2 = 40.2$ GPa. The column was $L = 100$ mm long and 40 mm wide and modeled by 2D, plane strain methods using 5×5 mm cells with four particles per cell. The column was initially compressed to a constant displacement of $d_z = 1$ mm or to a constant compressive stress of $\sigma_z = 200$ MPa over the time for stress waves to travel column length 50 times (about 1.45 ms for tensile wave speed of 3400 m/s) and then held for twice the load time. The goal was to quasi-statically apply initial displacement or load. After the hold phase, the bottom was opened by setting grid boundary conditions along the column's bottom to zero pore pressure.

Fixed displacement and fixed load simulation results using FMPM(2) with $f_X = 2$ are in Fig. 6. At fixed displacement, the stress initially increased to about 475 MPa. Over 2500 ms of holding, the average stress gradually decreased to about 300 MPa. At fixed load, the initial plunger movement was about 0.42 mm. As fluid leaked out the bottom, the plunger movement increased toward 0.65 mm. The plotted “drained” limits are the *final compressibility* results derived in Biot [13] as:

$$\sigma_z = \frac{d_z}{aL} \quad \text{and} \quad d_z = a\sigma_z L \quad \text{where} \quad a = \frac{(1 + \nu)}{3K(1 - \nu)}$$

The plotted “undrained” limits are the *initial compressibility* results derived in Biot [13] as:

$$\sigma_z = \frac{d_z}{a_i L} \quad \text{and} \quad d_z = a_i \sigma_z L \quad \text{where} \quad a_i = \frac{a}{1 + \alpha_B^2 a Q}$$

Both simulations correctly reached “undrained” results after initial quasi-static loading and then approached “drained” limits over time.

FMPM(k) can be used for mechanics analysis (see Ref. [11]), transport analysis (as derived here), both, or neither. Repeating these simulations using neither (*i.e.*, FLIP for both mechanics and transport analysis) was always unstable. These calculations required FMPM(k) for mechanical stability. For simulations using FMPM(2) for mechanics, the transport analysis method affected the results. Using FLIP gave reasonable global results, but local pore pressures developed significant oscillations. Using FMPM(2) on every time step (with $m = 1$) approached the drained limit too fast. This numerical diffusion was because the mechanics time step was much shorter than the transport time step. Numerical diffusion was eliminated by using either periodic or blended FMPM(2) with $f_X = 2$ for transport analysis.

Another challenge in poroelasticity modeling is the disparity in time scales between stress waves and pore pressure evolution, especially in low permeability materials. For Ruhr sandstone, the stress wave speed is 3400 m/s while pore fluid velocity in the constant displacement simulation was about $v_f = (k/\eta)dp/dy \sim$

0.01 m/s. On the time scale of pore fluid transport, all mechanics is quasi static. An option to speed up calculations, therefore, is to scale time by multiplying both sides of Eq. (5) by t_{ref} . The net effect is a new equation in reduced time ($t_{red} = t/t_{ref}$) for a material with permittivity and flux boundary conditions both increased by factor of t_{ref} . The simulation output then multiplies simulation time by t_{ref} to get low-permittivity results for actual time. The simulations in Fig. 6 used $t_{ref} = 10$. The displacement control simulations were identical in the range of $t_{ref} = 10$ to 1000. A real-time simulation ($t_{ref} = 1$) eventually developed numerical artifacts near the bottom drain that degraded the results. The load control simulations were identical for $t_{ref} = 1$ to 100. Simulations with $t_{ref} = 1000$ gave similar displacement change rate but the displacement oscillated around the lower t_{ref} results.

4.4. Fracture Phase Field Diffusion

An example modeling crack propagation by coupled phase-field diffusion analysis, as termed a “viscous regularization” method [16], was a $1 \times 1 \times 1$ mm specimen with a 0.5 mm single-edge notched loaded in tension on top and bottom (SENT) at a constant displacement rate slow enough to model quasi-static loading [15, 16]. The MPM model used 0.01 mm cells with 4 particles per cell. The initial 0.5 mm crack was modeled as an explicit MPM crack [3]. The material properties were modulus $E = 210$ GPa, Poisson’s ratio $\nu = 0.3$, and density $\rho = 8$ g/cm³. The material fracture properties were $G_c = 2700$ J/m², $\ell = 0.02$ mm, and $g(\xi) = (1 - \xi)^2$. Miehe *et. al* [16] recommend ℓ to be at least 4 finite element lengths to resolve phase field shape near a crack. MPM calculations similarly need ℓ to be at least 4 particle diameters (as set in this example). The phase field viscosity, η , is ideally small. This example first picked η such that the mechanics and transport time steps matched, which by Eq. (21) is:

$$\eta_{min} = \frac{f_M}{f_\Theta} \frac{2G_c \ell}{v_{max} \Delta x}$$

In this example, using $f_M = 0.25$ and $f_\Theta = 0.5$ gave $\eta_{min} = 0.3134$ Pa · sec.

Figure 7 shows phase field calculations when the crack has propagated half way to the specimen’s edge. Using standard FLIP methods for transport gave poor results and extensive oscillations (seen as checkerboard patterns in Fig. 7A). Modeling crack planes, which are always near interfaces, is an example that shows MPM transport methods that rely on FLIP are unable to model phase-field fracture by coupled transport calculations. In contrast, using periodic FMPM(2) for transport with $f_X \geq 2$ to avoid unwanted diffusion completely removes all oscillations and provides a valid phase-field fracture method (see Fig. 7B).

Fracture mechanics calculations for an SENT specimen under displacement control show that energy release rate increases with crack length implying that crack growth, once initiated, should be rapid and unstable. Simulation results should reach a peak and then phase field should propagate close to wave speed of the material. Simulated force-displacement results in Miehe *et al.* [15, 16] all have that behavior. FMPM(2) (with $f_X \geq 2$) also has that behavior and results are virtually identical to Miehe *et al.* [15, 16] (hence judged as sufficiently converged for purposes of this paper). The checkerboard phase field distribution when using FLIP caused force to rise too high before failure and crack propagation to be too slow. Although FLIP phase field on the grid was reasonable, the oscillations in particle values made FLIP unacceptable for fracture phase field calculations.

Alternate phase-field fracture implementations in MPM directly solve the phase field equation (*i.e.*, equation with $\eta = 0$) on each time step using linear finite element methods [21, 22] or conjugate gradient methods [23]. While these approaches work and avoid transport modeling oscillation problems, they have two drawbacks. First, they are inefficient. Repeating the above SENT specimen calculations with a grid-based finite element analysis on each time step on the same computer was 36 times slower than the coupled transport analysis method. Second, a reliance on linear finite element methods to solve for phase field limits the modeling to linear $g'(\xi) = c_1 + c_2\xi$. More general material modeling benefits from modeling methods that can handle nonlinear $g'(\xi)$.

When using “viscous regularization,” η must be small enough for reliable results. For example, Fig. 8 plots displacement at the midpoint of the post-peak rapid drop to zero force as a function of η . This geometry had consistent results for $\eta < 1$ Pa · sec. For $\eta > 1$ Pa · sec, the slowed phase-field diffusion delays the failure initiation thereby shifting the displacement when the load drops. The dotted line shows η_{min} used above. While this value is acceptable, it was selected simply by matching time steps. More complete calculations should always verify η is acceptable.

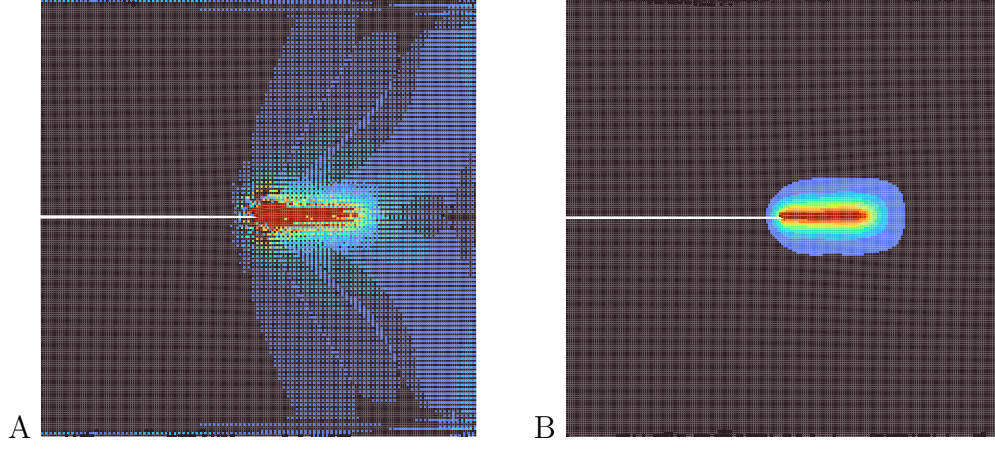


Figure 7: Plots of phase field when the crack has propagated about half way from initial crack tip to the specimen's edge. The 10 contours are for $\xi = 0$ (remote from the cracks in black) to $\xi = 1$ (near the propagating crack path in red) in increments of 0.1. A. Phase field diffusion solved using FLIP methods. B. Phase field diffusion solved using periodic FMPM(2) with $f_X = 2$.

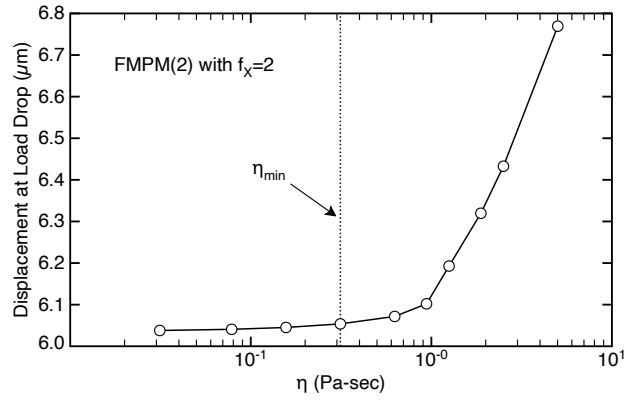


Figure 8: Displacement when the load has dropped to half the peak value as a function the η using the phase-field transport equation. The vertical dotted line is value of η such that transport time step matches the mechanics time step.

Finally, the point of including fracture phase field diffusion in this paper was neither to recommend nor denigrate phase field methods for fracture modeling. Rather, this example suggests that a viscous regularization method [16] the couples mechanics to a diffusive phase field and uses FMPM(k) for the phase field diffusion is the preferred method for implementing phase field fracture methods in MPM. The diffusion approach is much faster than prior MPM phase field fracture methods [21, 22] and permits modeling with nonlinear $g'(\xi)$. A successful diffusion approach, however, requires inclusion of FMPM($k > 2$) for transport to eliminate the phase field oscillations that corrupt any modeling based on FLIP transport methods.

4.5. Thermodynamics

The “Thermodynamics” in section 3.3 provides methods for tracking and spatially resolving thermodynamic state variables. Validating this tracking in solid mechanics is difficult because the effects are small and few exact solutions are available. One validation option is to simulate an ideal gas. For an ideal gas, the current pressure, volume, and temperature follow:

$$\frac{PV}{T} = \frac{P_0 V_0}{T_0}$$

where P_0 , V_0 , and T_0 are for a reference state. MPM codes typically work with specific Cauchy stress, which for an ideal gas reduces to:

$$\frac{\sigma}{\rho} = -\frac{P}{\rho} \mathbf{I} \quad \text{where} \quad \frac{P}{\rho} = \frac{P_0}{\rho_0 T_0} T$$

Here ρ and ρ_0 are current and reference state gas density. The internal energy of an ideal gas depends on only temperature. This response can be modeled by converting all work to a reversible adiabatic temperature rise $dT_{p,q=0} = dW_p/C_v$. This material implemented in MPM is stable provided the gas is confined to a box (*i.e.*, free gas expansion into a vacuum is unstable).

Imagine a gas confined to a box with a moving piston that slowly (*i.e.*, reversibly) applies an axial strain ε_{xx} such that volume has changed to $V = V_0(1 + \varepsilon_{xx})$. For adiabatic compression (or expansion), the expected results [26] averaged over all particles are:

$$P = \frac{P_{init}}{(1 + \varepsilon_{xx})^\gamma}, \quad T = \frac{T_{init}}{(1 + \varepsilon_{xx})^{\gamma-1}}, \quad W = U = C_v(T - T_i), \quad \text{and} \quad Q = S = 0$$

and $\gamma = 5/3$ and $C_v = 3P_0/(2\rho_0 T_0)$ for monoatomic gas or $\gamma = 7/5$ and $C_v = 5P_0/(2\rho_0 T_0)$ for diatomic gas. For isothermal compression, the expected results [26] are:

$$P = \frac{P_{init}}{1 + \varepsilon_{xx}}, \quad T = T_{init}, \quad -Q = W = -P_{init} V_0 \ln(1 + \varepsilon_{xx}), \quad U = 0, \quad \text{and} \quad S = \frac{P_0 V_0}{T_0} \ln(1 + \varepsilon_{xx})$$

Coupled heat conduction and mechanics modeling were run in adiabatic and isothermal modes and thermodynamics values were tracked following section 3.3. All expected results accurately matched above analytical results (the “uninteresting” plots are not provided). Although modeling an ideal gas in MPM is uncommon, it is a good material model for validating thermodynamics calculations.

5. Conclusions

One motivation for MPM is to model materials with history-dependent response. Because the entire solution and material history is tracked on the particles, MPM handles such materials well. By extending MPM to couple transport equations, MPM becomes a good platform for multi-physics problems. The problem with straight-forward FLIP transport methods, however, is that particle values often develop oscillations. The oscillations are usually near interfaces or rapid changes in value and can be large. Although the grid solution remains accurate, particle values are unreliable. In short, standard MPM methods to couple transport with mechanics lose the advantage of tracking accurate information on the particles.

The FMPM(k) method derived here completely resolves these issues. It removes all oscillations and returns particle values to accurate transport values. Constitutive laws that depend on these values can then use them in calculations. When the characteristic transport time step is much longer than the mechanics

time step, FMPM(k) can cause unwanted numerical diffusion. This issue can be resolved by using higher k , by using periodic FMPM(k), or by blending FLIP with FMPM(k). The most efficient method is to use low $k \geq 2$ and periodic FMPM(k) with $f_X \geq f_{\Theta}^{(max)}(k)$. If the periodic method causes bothersome periodic variations, the next best method is to use blended FLIP/FMPM(k) with low k and fraction FMPM(k) $\psi \leq f_{\Theta}^{(eff)}/f_{\Theta}^{(max)}(k)$.

The first two examples showed how FMPM(k) can eliminate large oscillations in temperature values and can improve accuracy. These examples were not coupled to mechanics. The examples on poroelasticity and phase field fracture are always coupled to mechanics. These examples were unstable or inaccurate using standard FLIP methods. Both became stable and accurate when doing transport by FMPM(k) methods.

Acknowledgements

I thank Murata Manufacturing Co., Ltd., for sponsoring a project that depended on coupling three transport phenomena in MPM and Grant Smith for finding issues in the project needing improvement for MPM to be a robust, multi-physics platform.

Appendix

These steps outline tasks for implementing FMPM(k) transport analysis in an MPM time step:

1. Extrapolate Particle Properties to the Grid: Loop over particles calculating nodal content, τ_i , lumped transport capacity, c_i^L , and a relative content, $\tau_{i,rel}$, using Eqs. (4) and (8) or:

$$\tau_i = \sum_p S_{pi} V_p C_{\Theta,p} \Theta_p, \quad c_i^L = \sum_p S_{pi} V_p C_{\Theta,p}, \quad \text{and} \quad \tau_{i,rel} = \sum_p S_{pi} V_p C_{\Theta,p} \frac{\Theta_p}{c_{rel,p}^L}$$

When done, lumped nodal values are $\theta_i^L = \tau_i/c_i^L$; save a copy in θ_i^{L0} for use below in boundary conditions. $\tau_{i,rel}$ is only needed for materials with position-dependent activity; if needed, find activity potential from $\bar{a}_i^L = \tau_{i,rel}/c_i^L$.

2. Particle Gradients: Change θ_i^L on any node with a boundary condition to $\theta_i^{(BC)}$. Use these revised, lumped nodal values and Eq. (7) to get transport gradients on the particles by:

$$\nabla \Theta_p = \sum_i \mathbf{G}_{pi} \theta_i^L$$

When done, restore θ_i^L to θ_i^{L0} . Note that when concentration is modeling activity, this calculation needs to get gradient of activities by changing $\bar{a}_i^L$ on any node with a boundary condition to $\bar{a}_i^L \bar{c}_i^{(BC)}/\theta_i^L$ (i.e., scale the concentration potential BC by extrapolated activity coefficient $\gamma_i = \bar{a}_i^L/\theta_i^L$). Once relative activities are revised, find the needed gradients instead from:

$$\nabla \Theta_p = \sum_i \mathbf{G}_{pi} \bar{a}_i^L$$

3. Grid Forces: Loop over particles and extrapolate particle flows and grid source terms to the grid using Eq. (6) or:

$$f_i = \sum_p \left(-\mathbf{G}_{pi} V_p \boldsymbol{\kappa}_p \nabla \Theta_p + S_{pi} V_p \dot{Q}_p^{(g)} \right)$$

Add s_i due to any flux boundary conditions to f_i using Eq. (14). Note that grid source terms, $\dot{Q}_p^{(g)}$, are normally not used.

4. Update Grid Values: Find transport rate on each node by lumped methods in Eq. (9) and then update lumped transport value using

$$\theta_i^{L+} = \theta_i^L + v_i^{(\theta)} \Delta t$$

Change θ_i^{L+} on any node with a boundary condition to $\theta_i^{(BC)}$ and increment $v_i^{(\theta)}$ on that node by $(\theta_i^{(BC)} - \theta_i^{L+})/\Delta t$ such that $v_i^{(\theta)}$ is the *consistent* transport rate that updates from θ_i^L to $\theta_i^{(BC)}$.

5. FMPM(k) Calculations: When using FMPM(k) on the current time step, calculate new transport values on the grid from updated lumped values using Eq. (11), which for implementation is recast as

$$\boldsymbol{\theta}(k) = \sum_{\ell=1}^k (-1)^{\ell+1} \boldsymbol{\theta}^{(\ell)}, \quad \boldsymbol{\theta}^{(1)} = k\boldsymbol{\theta}^{L+}, \quad \text{and} \quad \boldsymbol{\theta}^{(\ell)} = \frac{k-\ell+1}{\ell} \mathbf{S}^{\ominus} \mathbf{S} \boldsymbol{\theta}^{(\ell-1)}$$

A single subroutine to calculate $\boldsymbol{\theta}^{(\ell)}$ can be called recursively to implement FMPM(k). When done, change $\theta_i(k)$ on any node with a boundary condition to $\theta_i^{(BC)}$, but do not increment $v_i^{(\theta)}$ on those nodes. This task is the only extra task needed to implement FMPM(k); more FMPM(k) implementation information is given in Refs. [10] and [11].

6. Particle Updates: Particle updates for FLIP and FMPM(k) using Eq. (12) are:

$$\Theta_p^+ = \begin{cases} \Theta_p + \sum_i S_{pi} v_i^{(\theta)} \Delta t + d\Theta_p^{(ad)} & \text{FLIP} \\ \sum_i S_{pi} \theta_i(k) + d\Theta_p^{(ad)} & \text{FMPM}(k) \end{cases}$$

For FLIP, evaluate $\bar{\Theta}_p^+$ and find value increment by $\Delta\Theta_p = \bar{\Theta}_p^+ - \bar{\Theta}_p$. For FMPM(k) find value increment by $\Delta\Theta_p = \bar{\Theta}_p^+ - \Theta_p$ (but change Θ_p to Θ_{ref} on the first time step). When using blended FLIP/FMPM(k) combine these calculations by the chosen fraction FMPM(k). When doing heat conduction modeling, increment heat energy and entropy as explained in section 3.3 by the temperature change caused only by conduction.

7. Particle Stresses, Strains, and Thermodynamics: In MPM mechanics using FMPM(k), the recommended place to implement constitutive laws is after the particle updates [11]. When coupled to transport modeling, this task takes $\Delta\Theta_p$ as input for modeling residual stresses caused by changes in transport value. The constitutive law calculations should also calculate and buffer any modeled adiabatic changes to transport value, $d\Theta_p^{(ad)}$, to be added in the next particle update. When doing heat conduction modeling, these adiabatic changes should be divided into reversible and irreversible effects and added to heat energy and entropy as explained in section 3.3.

References

- [1] S. G. Bardenhagen, E. M. Kober, The generalized interpolation material point method, *Computer Modeling in Engineering & Sciences* 5 (2004) 477–496.
- [2] D. Sulsky, Z. Chen, H. Schreyer, A particle method for history-dependent materials, *Computer Methods in Applied Mechanics and Engineering* 118 (1) (1994) 179 – 196.
- [3] J. A. Nairn, Material point method calculations with explicit cracks, *Computer Modeling in Engineering and Sciences* 4 (6) (2003) 649–664.
- [4] J. A. Nairn, S. G. Bardenhagen, G. S. Smith, Generalized contact and improved frictional heating in the material point method, *Computational Particle Mechanics* 5 (3) (2018) 285–296. doi:10.1007/s40571-017-0168-1.
- [5] J. A. Nairn, C. C. Hammerquist, G. Smith, New material point method contact algorithms for improved accuracy, large-deformation problems, and proper null-space filtering, *Computer Methods in Applied Mechanics and Engineering* 362 (2020) 112859.
- [6] J. A. Nairn, J. E. Guilkey, Axisymmetric form of the generalized interpolation material point method, *Int. J. for Numerical Methods in Engineering* 101 (2015) 127–147.
- [7] J. A. Nairn, Modeling heat flow across material interfaces and cracks using the material point method, *Computational Particle Mechanics* 6 (1) (2019) 133–144.
- [8] J. U. Brackbill, H. M. Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, *Journal of Computational Physics* 65 (2) (1986) 314 – 343.

- [9] J. Brackbill, D. Kothe, H. Ruppel, FLIP: A low-dissipation, particle-in-cell method for fluid flow, *Computer Physics Communications* 48 (1) (1988) 25 – 38.
- [10] C. C. Hammerquist, J. A. Nairn, A new method for material point method particle updates that reduces noise and enhances stability, *Computer Methods in Applied Mechanics and Engineering* 318 (2017) 724–738.
- [11] J. A. Nairn, C. C. Hammerquist, Material point method simulations using an approximate full mass matrix inverse, *Computer Methods in Applied Mechanics and Engineering* 337 (2021) 113667.
- [12] F. Harlow, The particle in cell computing method for fluid dynamics, *Methods in Computational Physics* 3 (1964) 319.
- [13] M. A. Biot, General theory of three dimensional consolidation, *J. Appl. Phys.* 12 (1941) 155–164.
- [14] E. Detournay, A.-D. Cheng, Fundamentals of poroelasticity, in: C. Fairhurst (Ed.), *Comprehensive Rock Engineering: Principles, Practice and Projects*, Vol. II, Analysis and Design Method, Pergamon Press, 1993, pp. 113–171.
- [15] C. Miehe, F. Welschinger, M. Hofacker, Thermodynamically-consistent phase field models of fracture: Variational principles and multi-field FE implementations, *International Journal for Numerical Methods in Engineering* 83 (10) (2010) 1273–1311.
- [16] C. Miehe, M. Hofacker, F. Welschinger, A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits, *Computer Methods in Applied Mechanics and Engineering* 199 (2010) 2765–2778.
- [17] J. R. Rice, M. P. Cleary, Some basic stress diffusion solutions for fluid-saturated elastic porous media with compressible constituents, *Reviews of Geophysics and Space Physics* 14 (227–241).
- [18] B. Bourdin, G. A. Francfort, J.-J. Marigo, The variational approach to fracture, *J. Elasticity* 91 (2008) 5–148.
- [19] G. Francfort, J. Marigo, Revisiting brittle fracture as an energy minimization problem,, *ournal of the Mechanics and Physics of Solids* 46 (1998) 1319–1342.
- [20] M. Hofacker, C. Miehe, Continuum phase field modeling of dynamic fracture: variational principles and staggered fe implementation, *Int J. Fract.* 178 (2012) 113–129.
- [21] E. G. Kakouris, S. P. Triantafyllou, Material point method for crack propagation in anisotropic media: a phase field approach, *Arch Appl Mech* 88 (2018) 287–316.
- [22] E. G. Kakouris, S. P. Triantafyllou, Phase-field material point method for brittle fracture, *Int J Numer Meth Engng.* 112 (2017) 1750–1776.
- [23] J. Wolper, Y. Fang, M. Li, J. Lu, M. Gao, C. Jiang, Cd-mpm: Continuum damage material point methods for dynamic fracture animation, *ACM Trans. Graph.* 38 (4) (2019) 119.
- [24] L. Chen, H. W. Zhang, L. Y. Liang, Z. Liu, Y. Qi, P. Lu, J. Chen, L.-Q. Chen, Modulation of dendritic patterns during electrodeposition: A nonlinear phase-field model, *Journal of Power Sources* 300 (2015) 376–385.
- [25] G. S. Smith, J. A. Nairn, An MPM-based phase-field simulation of plating and stripping of lithium with a solid electrolyte interphase, *Journal of the Electrochemical Society* (2023) Submitted.
- [26] D. Kondepudi, I. Prigogine, *Modern Thermodynamics*, John Wiley & Sons, 1998.
- [27] M. L. Wilkens, *Computer Simulation of Dynamic Phenomena*, Springer-Verlag, New York, 1999.