LA-UR-16-24765

Approved for public release; distribution is unlimited.

Title: Multiscale Modeling using Molecular Dynamics and Dual Domain Material Point Method

Author(s): Dhakal, Tilak Raj

Intended for: Interview

Issued: 2016-07-08

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Multiscale Modeling using Molecular Dynamics and Dual Domain Material Point Method

Tilak R. Dhakal

Fluid Dynamics and Solid Mechanics Group, T-3
Theoretical Division
Los Alamos National Laboratory

PhD Candidate
Rice University
Houston, TX

July 7, 2016

Introduction

- For problems involving large material deformation rate, the material deformation time scale can be shorter than the material takes to reach a thermodynamical equilibrium
- For such problems, it is difficult to obtain a constitutive relation
- History dependency become important because of thermodynamic non-equilibrium

Our goal is to build a multi-scale numerical method which can bypass the need for a constitutive relation

Liouville Equation and Transport Equations

In principle, a system can be completely and uniquely determined by a point $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n; \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, in the position-velocity phase space.

The Liouville Equation describes the evolution of probability density function $P(\mathcal{C}, t)$ as,

$$\frac{\partial}{\partial t} P + \sum_{j=1}^N \nabla_{\mathbf{x}^j} \cdot (\mathbf{v}_j P) + \sum_{j=1}^N \nabla_{\mathbf{v}^j} \cdot (\dot{\mathbf{v}}_j P) = 0$$

Taking moment of this equation yields continuum equations, e.g. for any quantity g_j pertaining to a particle j ,

$$\int \sum_j \delta(\mathbf{x}_j - \mathbf{x}) g_j \left\{ \frac{\partial}{\partial t} P + \sum_{j=1}^N \nabla_{\mathbf{x}^j} \cdot (\mathbf{v}_j P) + \sum_{j=1}^N \nabla_{\mathbf{v}^j} \cdot (\dot{\mathbf{v}}_j P) \right\} d\mathbf{x}_1 \dots d\mathbf{x}_N d\mathbf{v}_1 \dots d\mathbf{v}_N = 0$$

After some manipulations, one can obtain the transport equation for g as

$$\frac{\partial}{\partial t} (n\bar{g}) + \nabla \cdot (n\bar{g}\mathbf{v}) = n \frac{d\bar{g}}{dt}$$

Here n is the number density, and

$$n\bar{g} = \int \sum_j \delta(\mathbf{x} - \mathbf{x}_j) g_j P(\mathcal{C}, t) d\mathcal{C}$$

Momentum Equation

$$\frac{\partial}{\partial t}(n\bar{g}) + \nabla \cdot (n\bar{g}\mathbf{v}) = n\frac{dg}{dt}$$

- $g = m$ gives the Continuity equation
- $g = m\mathbf{v}$ gives the Momentum equation
- $g = \frac{1}{2}m\mathbf{v}^2$ gives the Energy equation

$$g = m\mathbf{v} \Rightarrow \frac{\partial n\overline{m\mathbf{v}}}{\partial t} + \nabla \cdot (n\overline{m\mathbf{v}\mathbf{v}}) = n\overline{m\mathbf{b}} + n\sum_{\beta} \overline{\mathbf{f}_{\alpha\beta}}$$

Noting $\rho = n\bar{m}$, $\tilde{\mathbf{v}} = \frac{n\overline{m\mathbf{v}}}{\rho}$, $\mathbf{f}_{\alpha\beta} = -\mathbf{f}_{\beta\alpha}$, after some manipulations, we get the momentum equation as

$$\frac{\partial \rho \tilde{\mathbf{v}}}{\partial t} + \nabla \cdot (\rho \tilde{\mathbf{v}} \tilde{\mathbf{v}}) = \rho \mathbf{b} + \nabla \cdot \boldsymbol{\sigma}$$

$$\boldsymbol{\sigma} = -\rho \overline{\mathbf{v}'\mathbf{v}'} + \frac{1}{2} \int \sum_{\alpha} \delta(\mathbf{x} - \mathbf{y}_{\alpha}) \sum_{\beta} (\mathbf{y}_{\beta} - \mathbf{y}_{\alpha}) \mathbf{f}_{\alpha\beta} P d\mathbf{y}_1 \dots d\mathbf{y}_N d\mathbf{v}_1 \dots d\mathbf{v}_N$$

No continuum assumption is necessary for these equations.

$$\boldsymbol{\sigma} = \frac{1}{2} \int \sum_{\alpha=1}^N \sum_{\beta=1}^N \delta(\mathbf{x} - \mathbf{y}_{\alpha}) \mathbf{r}_{\alpha\beta} \mathbf{f}_{\alpha\beta} P(\mathcal{C}, t) d\mathcal{C} - \overline{(\rho \mathbf{v}' \mathbf{v}')}.$$

Method 1: Use constitutive relation to calculate $\boldsymbol{\sigma}$. Most closure models are for thermodynamically equilibrium states – small strain rates.

Method 2: Use Molecular Dynamics (MD) to calculate $\boldsymbol{\sigma}$ directly. (This talk.)

- It is impractical to simulate an entire macroscopic domain using MD
- For a macroscopic continuum calculation, we propose to perform MD **only** around stress points in numerical methods.

Numerical requirements:

- Capable of handling extreme deformation.
- Lagrangian capability to track history of deformation.
- No or little communication among stress points.
- Small number of stress points.

Material Point Method

We want to solve the momentum equation,

$$\rho \frac{dv}{dt} - \nabla \cdot \sigma = 0$$

Similar to FEM, Material Point method (MPM) seeks weak solutions of PDEs.

For an arbitrary continuous function δv

$$\int (\rho \frac{dv}{dt} - \nabla \cdot \sigma) \delta v dV = 0$$

Discretizing v and δv as, $v = \sum v_i S_i(x)$ and $\delta v = \sum \delta v_j S_j(x)$, and using integration by parts, we get

$$\sum_i M_{ji} \frac{dv_i}{dt} - \int \sigma \cdot n S_j dA + f_j^{int} = 0$$

Material Point Method (contd.)

FEM

$$\int \sigma \cdot \nabla S_j dv = \sum_{g=1}^N J_g W_g \sigma_g \nabla S_j(\mathbf{x}_g),$$

- Elements are Lagrangian. They can be distorted for large material deformations.
- MPM uses Lagrangian material points to follow the history of the material and Eulerian cells to calculate the spatial derivatives in continuum level.
- Avoids numerical diffusion issues of Eulerian Methods and mesh distortion issues of Lagrangian methods as in FEM.
- Material points only communicate with cell nodes. No communication among material points is necessary.

Material Point Method (MPM)

$$\int \sigma \cdot \nabla S_j dv = \sum_{p=1}^{N_p} v_p \sigma_p \nabla S_j(\mathbf{x}_p)$$

- Mesh is fixed. **No mesh distortion.**
- Particles move (= **advection**).

Mesh distortion in FEM (Lagrangian Method)
Cline and Raugh, J. De. Physique IV, 1991.

Eulerian Method

Improvements in Material Point Method

- The original MPM suffers numerical noise when the material points cross the cell boundaries because of the discontinuity in the gradient of shape function.

Shape Function

Gradient of Shape Function

- Past solutions:
 - Using higher order shape functions
 - Requires enlarged support
- Generalized Interpolation Material Point (GIMP) Method:
 - Works for moderate deformations
 - Need to track particle size and shape for large deformation
- Convected Particle Domain Interpolation (CPDI) Method:
 - Introduces smoothing effects using particle domains
 - For large deformation, the particle domains can overlap causing instability
- Dual Domain Material Point (DDMP) Method:
 - Modifies the gradient of shape function.
 - The shape function is the same as original MPM.
 - Can be used for very large deformations.

Dual Domain Material Point Method

The gradient of shape function is modified as,

$$\overline{\nabla} S_i(\mathbf{x}) = \alpha(\mathbf{x}) \nabla S_i(\mathbf{x}) + [1 - \alpha(\mathbf{x})] \widetilde{\nabla} S_i(\mathbf{x})$$

where,

$$\alpha(\mathbf{x}) = 0.5 \left\{ \prod_{k=1}^{n_c} [n_c S_k(\mathbf{x})] \right\}^{\frac{3}{2(n_c-1)d}}$$

$$\widetilde{\nabla} S_i(\mathbf{x}) = \sum_{j=1}^N \frac{S_j(\mathbf{x})}{V_j} \int_{\Omega} S_j \nabla S_i dv$$

$\sum_i S_i(\mathbf{x}) = 1 \Rightarrow$ Mass Conservation

$\sum_i \overline{\nabla} S_i(\mathbf{x}) = 0 \Rightarrow$ Momentum Conservation

$\sum_i \mathbf{x}_i \overline{\nabla} S_i(\mathbf{x}) = \mathbf{I} \Rightarrow$ Energy conservation (2nd order)

1D shock wave in ideal gas

- The GIMP and CPDI methods give reasonable results even for small (8) number of mat'l pt. per cell
- The pressure spikes increase as the number of mat'l pt. per cell increases
- The DDMP method is inaccurate when a small number of particles is used, but improves as the number of particles increases.
- However, the large number of particles required to achieve accurate results renders this method very expensive, especially for similar problems in two- or three- dimensional cases.

GIMP 8 Mat'l pt. per cell

GIMP 64 Mat'l pt. per cell

CPDI 8 Mat'l pt. per cell

CPDI 64 Mat'l pt. per cell

DDMP 8 Mat'l pt. per cell

DDMP 64 Mat'l pt. per cell

DDMP with Sub-points

The need for many material points in DDMP arises not from the stress variation but from the variation of the shape functions over the cell size Δx .

Sub-point Method:

$$\mathbf{f}_i^{int} = - \int_{\Omega} \sigma \cdot \nabla S_i dv$$

$$DDMP : \mathbf{f}_i^{int} = - \sum_{p=1}^{N_p} \sigma_p v_p \cdot \nabla \overline{S_i}(\mathbf{x}_p)$$

$$Sub - Point : \mathbf{f}_i^{int} = - \sum_{p=1}^{N_p} \sigma_p \cdot \sum_{s=1}^{n_p} v_{ps} \nabla \overline{S_i}(\mathbf{x}_{ps})$$

- Sub-points are created in every time step solely for integration purpose.
- The Sub-point method preserves the conservation properties of DDMP

DDMP 2 mat'l pt. per cell, 16 sub-points

Strong shock DDMP

Molecular Dynamics: Cerium

Embedded Atom Method (EAM):

The total energy of the system

$$E = \sum_i F_i(\rho_{h,i}) + \frac{1}{2} \sum_{\substack{i,j \\ i \neq j}} \phi_{ij}(r_{ij})$$

ϕ_{ij} = pair potential

r_{ij} = distance between atoms i and j

F_i = embedding energy

$\rho_{h,i}$ = host electron density.

$$\rho_{h,i} = \sum_{j(\neq i)} \rho_{ij}(r_{ij})$$

Momentum Equation:

$$\frac{\partial \rho \tilde{\mathbf{v}}}{\partial t} + \nabla \cdot (\rho \tilde{\mathbf{v}} \tilde{\mathbf{v}}) = \rho \mathbf{b} + \nabla \cdot \boldsymbol{\sigma}$$

Stress calculation:

$$\boldsymbol{\sigma} = -\frac{1}{V} \left[\sum_i m_i \mathbf{v}_i' \mathbf{v}_i' + \sum_i \sum_{j>i} \mathbf{r}_{ij} \mathbf{f}_{ij} \right]$$

Force Calculation:

$$\mathbf{f}_i = -\nabla_i E = \sum_j \mathbf{f}_{ij}$$

$$\mathbf{f}_{ij} = -[\phi'_{ij} + F'_i \rho'_{ij} + F'_j \rho'_{ji}] \hat{\mathbf{r}}$$

$$\mathbf{f}_{ij} = -\mathbf{f}_{ji}$$

Periodic Boundary Condition is used

Tabulated potential generated by H.W. Sheng (hsheng@gmu.edu) has been used.

Molecular Dynamics in GPU

- Spatial decomposition using uniform grids of size cutoff distance (r_c).
- Calculate each atoms cell ID based on its position
- Sort the atoms based on their Cell ID (`thrust::sort_by_key`)
- Find the start and end address of any given cell in sorted list
- Reorder position and velocity arrays into sorted order
- For each atom, search for neighbors in the neighbor cells
- $120 \times$ speed up in compared to single core CPU for 1M atoms

Memory	Scope	Lifetime	Performance
Register	Thread	Thread	Fast
Local	Thread	Thread	Slow
Shared	Block	Block	Fast
Global	Grid	Application	slow
Constant	Grid	Application	Fast

CUDA memory architecture

Index	Unsorted list (cell id, particle id)	List sorted by cell id	Cell start
0	(9, 0)	(4, 3)	
1	(6, 1)	(4, 5)	
2	(6, 2)	(6, 1)	
3	(4, 3)	(6, 2)	
4	(6, 4)	(6, 4)	
5	(4, 5)	(9, 0)	
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

Sorting atoms in uniform grids

GPU specification:

- NVIDIA Tesla K20
- 2496 total cores
- 5 GB memory

Multi-scale calculation using GPUs

- Initialize atomic position, velocity data in CPU and copy to GPU
- Continuum level (DDMP) calculation is performed in CPU
- GPU takes strain rate as an input from CPU and perform MD calculation to calculate stress
- The size of each MD domain changes according to the macroscopic strain rate at the material point.
- One GPU thread is assigned for one atom

1D shock wave propagation in Cerium

Weak Shock at 0K

Strong Shock at 0K

Strong Shock at 300K

- MD: Full MD simulation using about 1.1 M molecules
- Continuum DDMP: 400 equal cells with 1 material point and 8 sub-points per material point, Young's Modulus, $Y = 31.5$ GPa for 0K and 42 GPa for 300K
- Multiscale DDMP-MD: DDMP calculation with stress calculated using MD simulation with 256 atoms
- MD time step: 1 fs
- DDMP time step: 50 fs
- Left half is held at finite pressure, right half at zero pressure
- At $t = 0$, the hold is released
- Computation time:
 - MD: 6 hrs (In GPU)
 - DDMP-MD: 15 mins
 - DDMP: 5 secs (In CPU)

1D shock wave propagation in Cerium (contd.)

(A) Different mat'l pt. per cell and sub-point effect

(B) Convergence of results with refined mesh

(C) Average vs. one step stress calculation

Figure (B)

- The results converge to the correct solution as mesh is refined

Figure (C)

- In general, stress is calculated at the last MD time step
- Another option would be to use the average stress calculated at each MD time step
- The results are similar but computation time is $1.8 \times$ higher for average stress method

Figure (A)

- The result improves with more mat'l pt per cell
- The most computationally expensive part is to calculate stress in each mat'l pt.
- Sub-points method improves result without the expense of computation time.
- Computation time:
 - 8 mat'l pt. per cell: 110 mins
 - 1 mat'l pt. per cell and 8 sub-points: 15 mins

MD calculation of Jet formation in Copper

- EAM potential developed by Mishin *et al.* is used.
- Dimensions: $x = 144$ nm, $y = 72$ nm
- Groove depth, $h = 36$ nm
- Piston size: 14.4 nm
- Total # of Molecules: 2.8 M
- At $t = 0$, the sample is shocked with an impact velocity $u_p = 2.5$ km/s
- Maximum stress developed: 162 GPa (Agrees to previous work by Cherne *et al.*)

Mishin *et al.* , Phys. Rev. B 63, 224106 (2001)
Cherne *et al.* , J. Appl. Phys. 118, 185901 (2015)

Conclusion

- Multi-scale simulation method is developed based on the dual domain material point (DDMP)
- Molecular dynamics (MD) simulation is performed to calculate stress
- Since the communication among material points is not necessary, the computation can be done embarrassingly parallel in CPU-GPU platform

Future Works

- Move on to 2D (currently working) and 3D problems
- Use multiple GPUs to speed up the calculation

Momentum equation derivation

$$\begin{aligned}\overline{n \sum_{\beta} \mathbf{f}_{\alpha\beta}} &= \int \sum_{\alpha} n \delta(\mathbf{x} - \mathbf{y}_{\alpha}) \sum_{\beta} \mathbf{f}_{\alpha\beta} P(\mathcal{C}) d\mathcal{C} \\&= \int \sum_{\alpha} \sum_{\beta} n \delta(\mathbf{x} - \mathbf{y}_{\alpha}) \mathbf{f}_{\alpha\beta} P(\mathcal{C}) d\mathcal{C} \\&= \int \sum_{\beta} \sum_{\alpha} n \delta(\mathbf{x} - \mathbf{y}_{\beta}) \mathbf{f}_{\beta\alpha} P(\mathcal{C}) d\mathcal{C} \\&= \frac{1}{2} \int \sum_{\alpha} \sum_{\beta} n [\delta(\mathbf{x} - \mathbf{y}_{\alpha}) \mathbf{f}_{\alpha\beta} + \delta(\mathbf{x} - \mathbf{y}_{\beta}) \mathbf{f}_{\beta\alpha}] P(\mathcal{C}) d\mathcal{C} \\&= \frac{1}{2} \int \sum_{\alpha} \sum_{\beta} n [\delta(\mathbf{x} - \mathbf{y}_{\alpha}) - \delta(\mathbf{x} - \mathbf{y}_{\beta})] \mathbf{f}_{\alpha\beta} P(\mathcal{C}) d\mathcal{C}\end{aligned}$$

Here,

$$\delta(\mathbf{x} - \mathbf{y}_{\beta}) = \delta(\mathbf{x} - \mathbf{y}_{\alpha} - (\mathbf{y}_{\beta} - \mathbf{y}_{\alpha})) \nabla_{\mathbf{x}} \delta(\mathbf{x} - \mathbf{y}_{\alpha})$$

Taylor series expansion yields,

$$\delta(\mathbf{x} - \mathbf{y}_{\beta}) = \delta(\mathbf{x} - \mathbf{y}_{\alpha}) - (\mathbf{y}_{\beta} - \mathbf{y}_{\alpha})$$

Pluggin in,

$$\begin{aligned}\overline{n \sum_{\beta} \mathbf{f}_{\alpha\beta}} &= \frac{1}{2} \int \sum_{\alpha} \sum_{\beta} n (\mathbf{y}_{\beta} - \mathbf{y}_{\alpha}) \nabla_{\mathbf{x}} \delta(\mathbf{x} - \mathbf{y}_{\alpha}) \mathbf{f}_{\alpha\beta} P(\mathcal{C}) d\mathcal{C} \\&= \nabla_{\mathbf{x}} \cdot \left[\frac{1}{2} \int \sum_{\alpha} \sum_{\beta} n (\mathbf{y}_{\beta} - \mathbf{y}_{\alpha}) \delta(\mathbf{x} - \mathbf{y}_{\alpha}) \mathbf{f}_{\alpha\beta} P(\mathcal{C}) d\mathcal{C} \right]\end{aligned}$$