OLE POLYTECHNIQUE
DÉRALE DE LAUSANNE

Application of Material Point Method

Enseignant I: Professor Jean-François Molinari
Enseignant II: Enrico Milanese

Selimcan Ozden

What is MPM

- MPM is a recent particle-in-cell method that discretizes material domain Ω with a set of material points in the initial configuration throughout deformation.

- Each material points has internal state variables such as mass, volume, density, Cauchy stress tensor, velocity etc.

Why are we interested in MPM ?

- To reproduce the wear mechanism observed in atomistic simulations at a larger scale where material can be described as a continuum.

Eulerian vs Lagrangian description

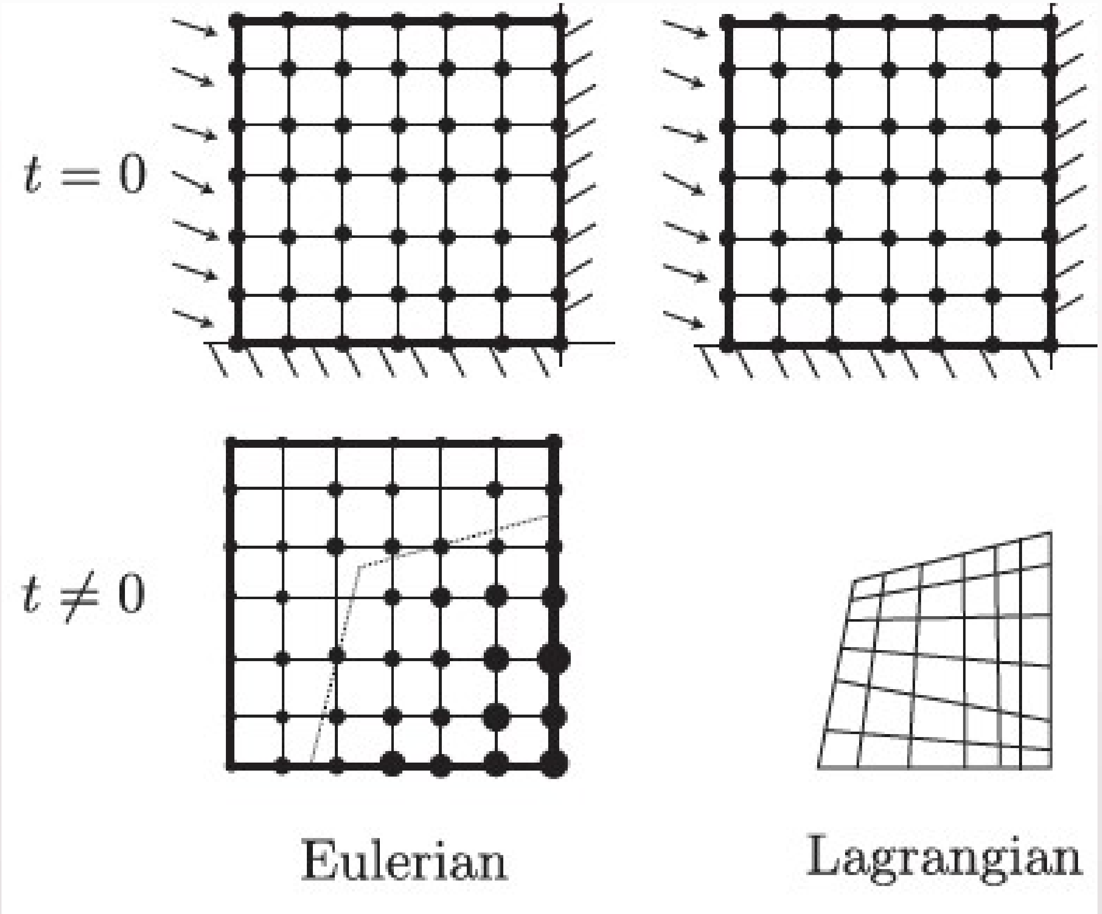


Figure 1. Eulerian vs Lagrangian description
(Vinh Phu Nguyen, Material Point Method: basics and applications)

Governing Equations

- Balance Laws (conservation of mass, conservation of momentum and conservation of energy)
- Constitutive equation
- Kinematics equation
- Boundary conditions (displacement and traction boundary)

Weak Formulation of Updated Lagrangian Formulation

- Weak formulation is the way to solve partial differential equations numerically with initial and boundary conditions.

$$\int_{\Omega} \rho \ddot{u}_i \delta u_i dV + \int_{\Omega} \rho \frac{\sigma_{ij}}{\rho} \delta u_{i,j} dV - \int_{\Omega} \rho b_i \delta u_i dV - \int_{\Gamma_t} \rho \frac{\bar{t}_i}{\rho} \delta u_i dA = 0$$

- This weak form equation can be written in terms of virtual work of the internal force, external force and inertial force.

$$\delta w = \delta w^{int} - \delta w^{ext} + \delta w^{kin} = 0 \qquad \delta w^{int} = \int_{\Omega} \rho \frac{\sigma_{ij}}{\rho} \delta u_{i,j} dV \qquad \delta w^{kin} = \int_{\Omega} \rho \ddot{u}_i \delta u_i dV \qquad \delta w^{ext} = \int_{\Omega} \rho b_i \delta u_i dV + \int_{\Gamma_t} \rho \frac{\bar{t}_i}{\rho} \delta u_i dA$$

δu_i	virtual displacement	ρ	density
$\ddot{u}_i$	acceleration	dA	area element
dV	volume element	δw^{int}	virtual work of internal force
b_i	body force per unit mass	δw^{ext}	virtual work of external force
$\delta u_{i,j}$	derivative of virtual displacement	δw^{kin}	virtual work of inertial force
$\frac{\sigma_{ij}}{\rho}$	specific stress	δw	virtual work
		$\frac{\bar{t}_i}{\rho}$	specific traction

Particle Quadrature

- For MPM, weak formulation is rewritten in terms of $\frac{\sigma_{ij}}{\rho}$ in order to integrate on all material points.

$$\int_{\Omega_p} \rho \cdot g(x,y,z)dV = \sum_{p=1}^{n_p} m_p \cdot g(x_p, y_p, z_p)$$

ρ	density
$\sigma_{ij}^s = \frac{\sigma_{ij}}{\rho}$	specific stress
m_p	mass of particle
n_p	number of particles
$g(x_p, y_p, z_p)$	coordinate of particle

ALGORITHM

- **Algorithm divided into 4 main parts:**
- 1) Mapping from particles to nodes
- 2) Imposing Boundary condition
- 3) Update momenta
- 4) Mapping from nodes to particles

1) Mapping from particles to nodes		
Description	**Theory**	**Application**
Calculate grid nodal mass	$m_I^k = \sum_{p=1}^{n_p} m_p N_{Ip}^k$	$m_1^k = mN_1^k$ $m_2^k = mN_2^k$
Calculate grid nodal momentum	$p_{iI}^k = \sum_{p=1}^{n_p} m_p v_{ip}^k N_{Ip}^k$	$p_{x1}^k = mv_x^k N_1^k$ $p_{x2}^k = mv_x^k N_2^k$
Impose boundary condition	$p_{iI}^k = 0$	$p_{x1}^k = 0$
Derivatives of shape functions	$N_{Ip,j}^k$	$N_{1,x}^k$ $N_{2,x}^k$
Calculate grid nodal internal forces	$f_{iI}^{int,k} = -\sum_{p=1}^{n_p} N_{Ip,j}^k \sigma_{ijp} \frac{m_p}{\rho_p}$	$f_{x1}^{int,k} = -N_{1,x}^k \sigma_{xx} \frac{m}{\rho}$ $f_{x2}^{int,k} = -N_{2,x}^k \sigma_{xx} \frac{m}{\rho}$

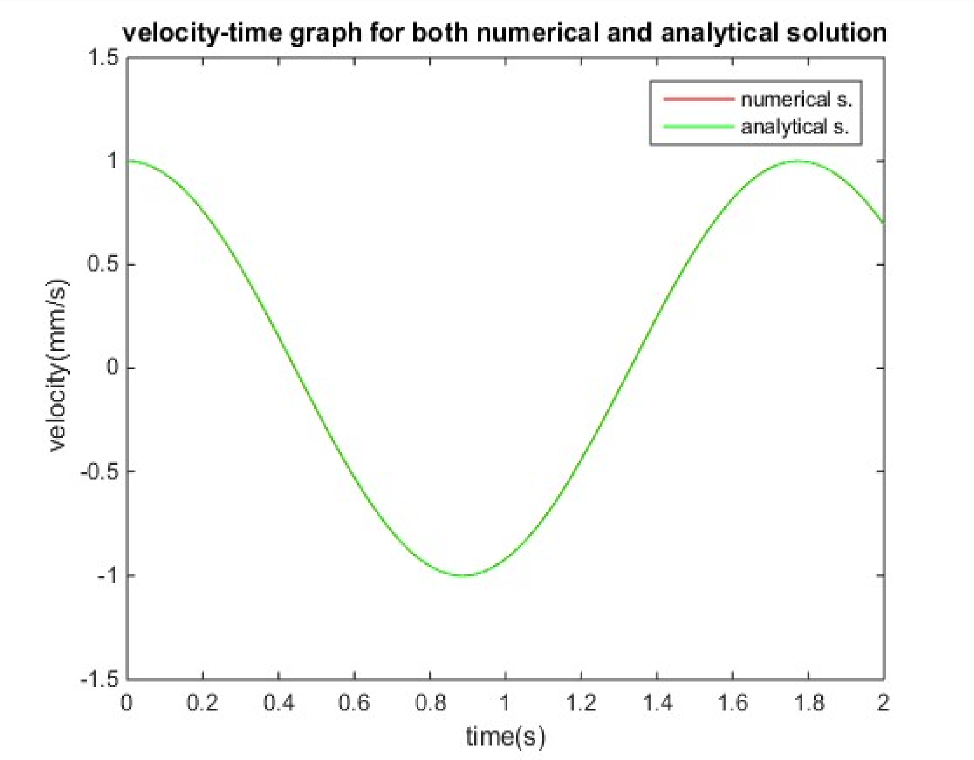
2) Imposing boundary conditions		
Description	**Theory**	**Application**
Impose boundary condition	$f_{iI}^{int,k} = 0$	$f_{x1}^{int,k} = 0$
Impose boundary condition	$p_{iI}^{k} = 0$	$p_{x1}^{k} = 0$

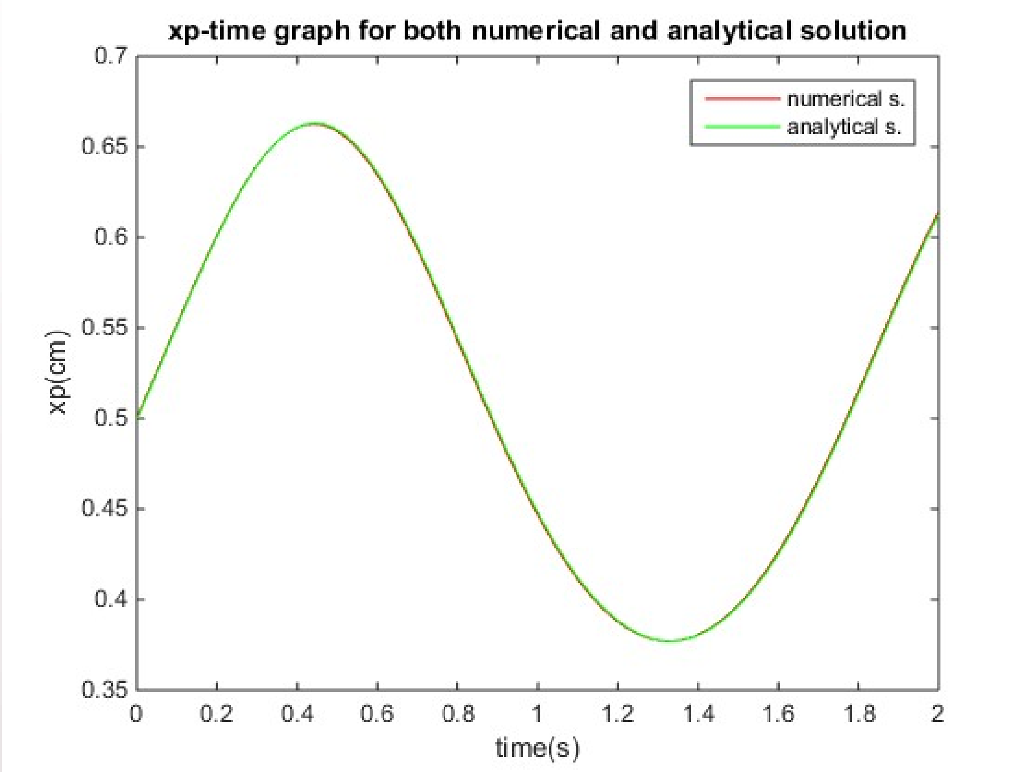
3) Update the momenta		
Description	**Theory**	**Application**
Update grid nodal momentum	$p_{iI}^{k+1} = p_{iI}^{k} + f_{iI}^{k}\Delta t$	$p_{x1}^{k+1} = p_{x1}^{k} + f_{x1}^{k}\Delta t$ $p_{x2}^{k+1} = p_{x2}^{k} + f_{x2}^{k}\Delta t$

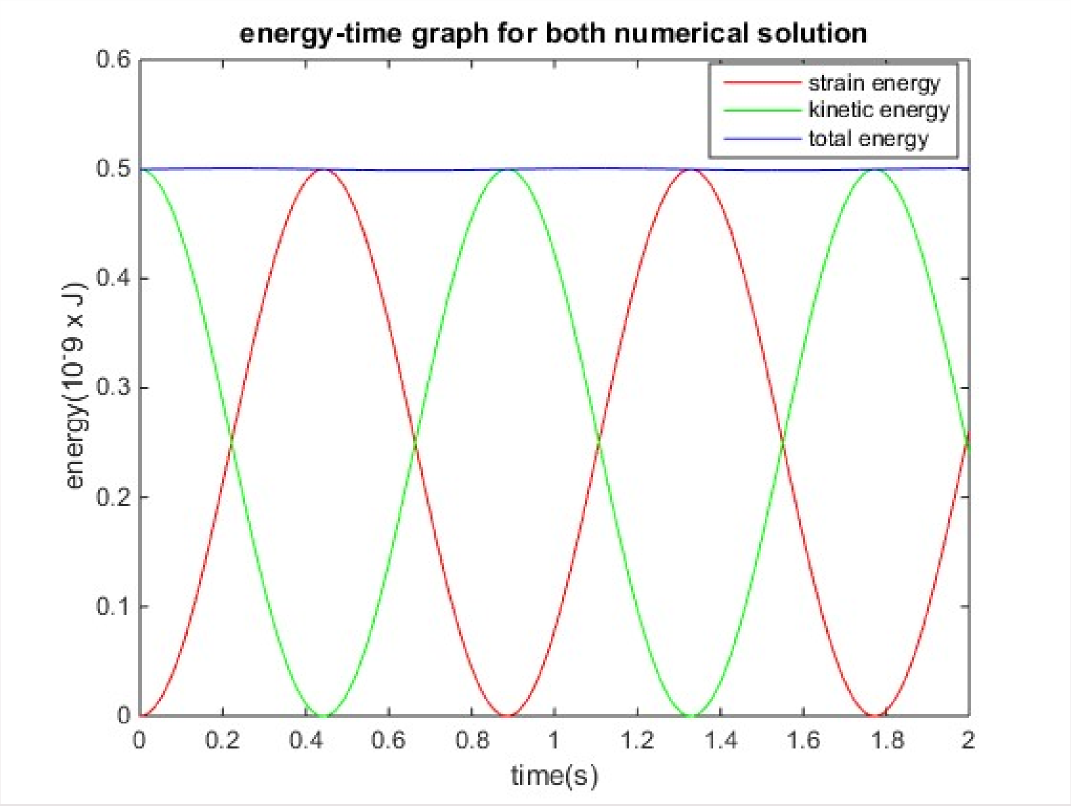
4) Mapping from nodes to particles		
Description	**Theory**	**Application**
Update particle velocity based on grid nodal acceleration	$v_{ip}^{k+1} = v_{ip}^{k} + \sum_{I=1}^{n_I} \frac{f_{iI}^{k} N_{Ip}^{k}}{m_I^k} \Delta t$	$v_x^{k+1} = v_x^k + \frac{f_{x1}^k N_1^k}{m_1^k} \Delta t + \frac{f_{x2}^k N_2^k}{m_2^k} \Delta t$
Update particle position based on grid nodal velocity	$x_{ip}^{k+1} = x_{ip}^{k} + \sum_{I=1}^{n_I} \frac{p_{iI}^{k} N_{Ip}^{k}}{m_I^k} \Delta t$	$x_x^{k+1} = x_x^k + \frac{p_{x1}^k N_1^k}{m_1^k} \Delta t + \frac{p_{x2}^k N_2^k}{m_2^k} \Delta t$
Recalculate grid nodal momentum based on particle updated particle momentum	$p_{iI}^{k+1} = \sum_{p=1}^{n_p} m_p v_{ip}^{k} N_{Ip}^{k}$	$p_{x1}^{k+1} = m v_x^{k+1} N_1^k$ $p_{x2}^{k+1} = m v_x^{k+1} N_2^k$
Impose boundary condition	$p_{iI}^{k+1} = 0$	$p_{x1}^{k+1} = 0$
Calculate grid nodal velocity	$v_{iI}^{k+1} = \frac{p_{iI}^{k+1}}{m_I^k}$	$v_{x1}^{k} = \frac{p_{x1}^{k+1}}{m_1^k}, \quad v_{x2}^{k} = \frac{p_{x2}^{k+1}}{m_2^k}$
Impose boundary condition	$v_{iI}^{k+1} = 0$	$v_{x1}^{k+1} = 0$
Calculate particle strain increment	$\Delta\varepsilon_{ijp}^{k+1} = \frac{1}{2}\left(N_{Ip,j}^{k} v_{iI}^{k+1} + N_{Ip,i}^{k} v_{jI}^{k+1}\right)\Delta t$	$\Delta\varepsilon_{xx}^{k+1}$ $= \frac{1}{2}\left(N_{1,x}^{k} v_{x1}^{k+1}\right.$ $\left. + N_{1,x}^{k} v_{x1}^{k+1} + N_{2,x}^{k} v_{x2}^{k+1} + N_{2,x}^{k} v_{x2}^{k+1}\right)\Delta t$
Calculate particle stress	$\sigma_{ijp}^{k+1} = \sigma_{ijp}^{k} + C_{ijke}\Delta\varepsilon_{ijp}^{k+1}$	$\sigma_{xx}^{k+1} = \sigma_{xx}^{k} + E\Delta\varepsilon_{xx}^{k+1}$ (for 1D example)

First Application: Vibration of a single material on a 1D bar

Property	Value
Young's modulus, E	4π MPa
Length, h	1 cm
Density, ρ	1 g/cm^3
Mass, m_p	1 g
Volume, V_p	1 cm^3
Initial velocity, v_o	1 mm/s

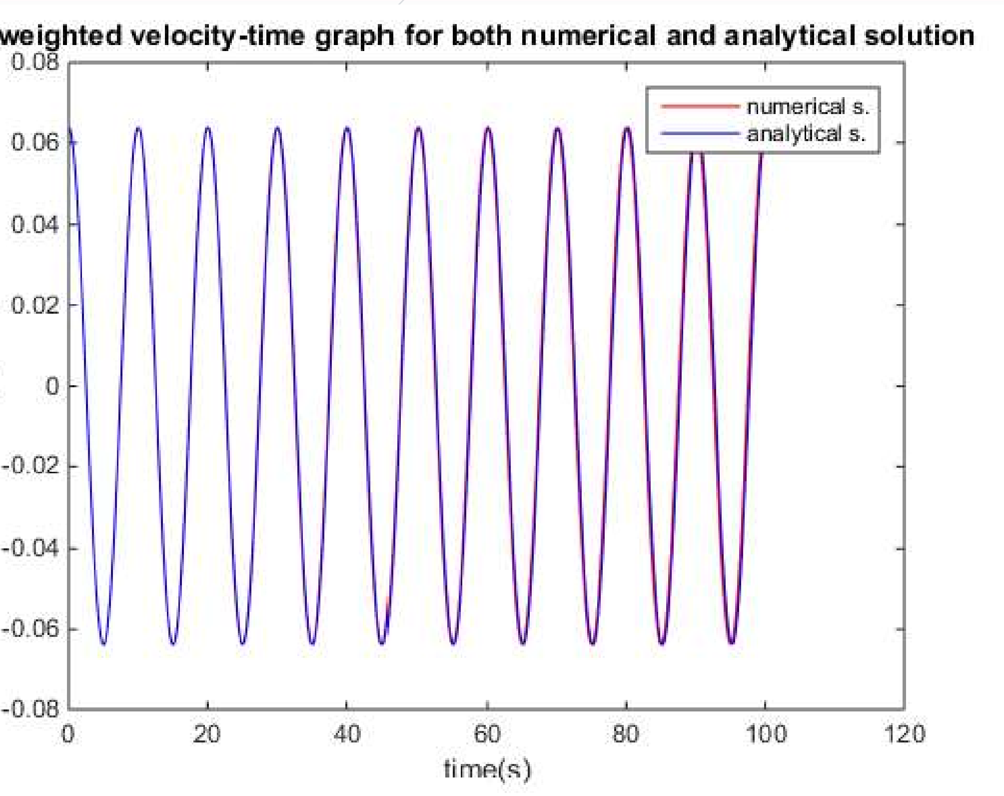

velocity-time graph for both numerical and analytical solution
numerical s.
analytical s.
velocity(mm/s)
time(s)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

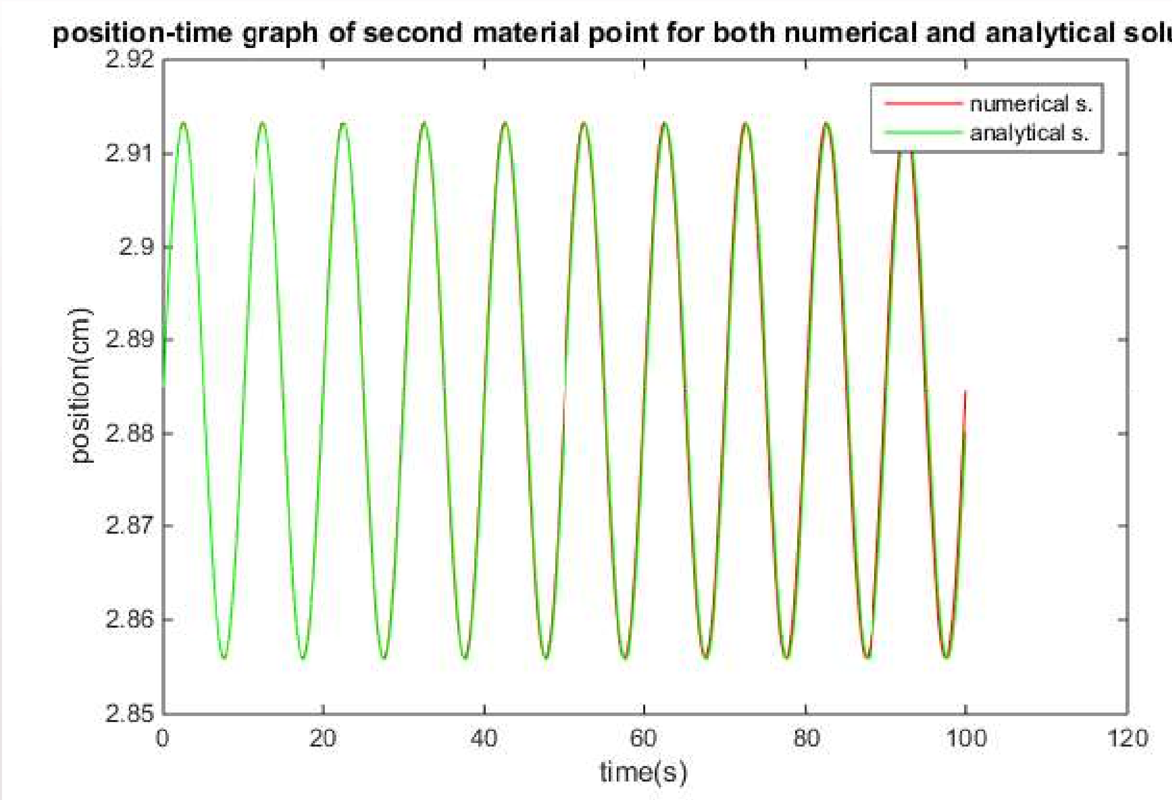

xp-time graph for both numerical and analytical solution
numerical s.
analytical s.
xp(cm)
time(s)
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

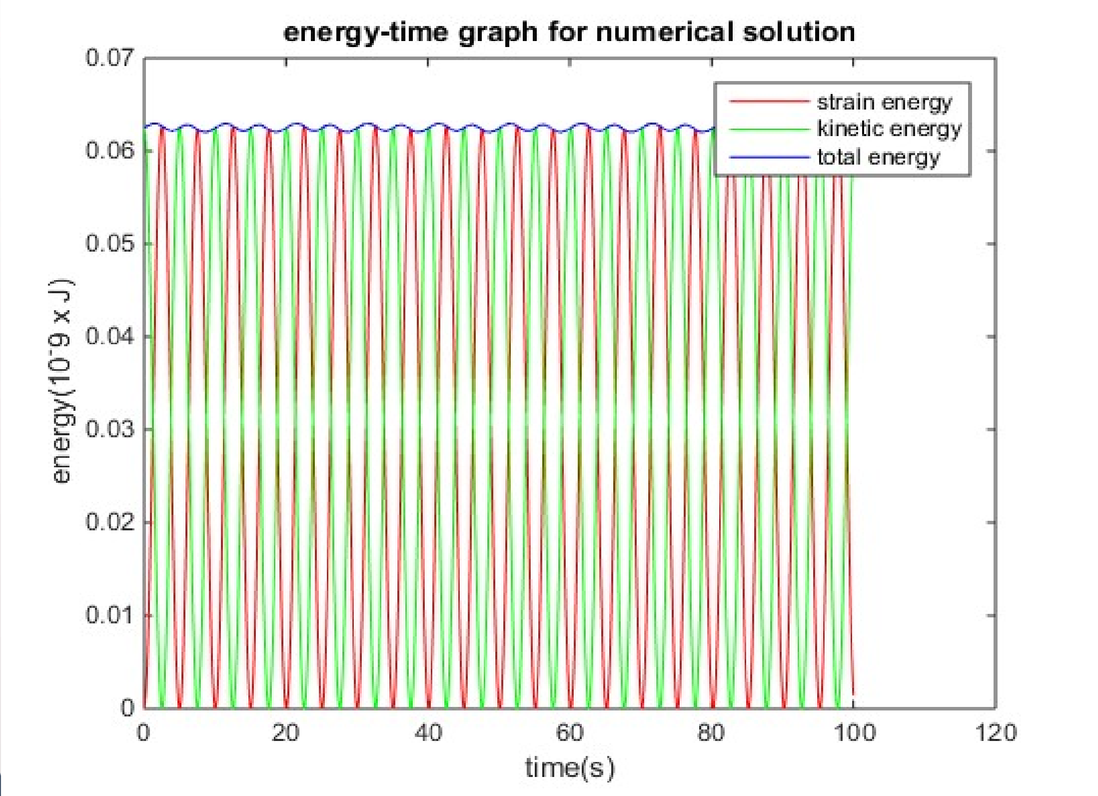

energy-time graph for both numerical solution
strain energy
kinetic energy
total energy
energy(10^-9 x J)
0
0.1
0.2
0.3
0.4
0.5
0.6
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
time(s)

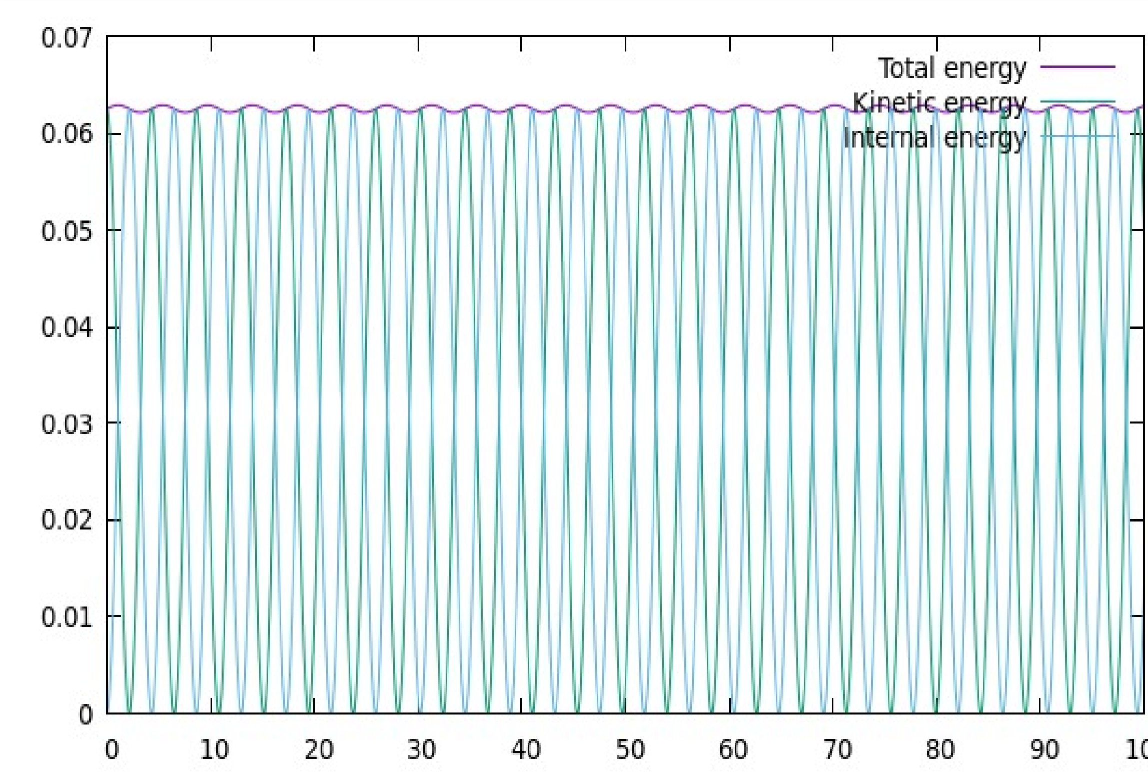
Second Application: Vibration of a continuum bar modeled by multiple material points

Property	Value
Young's modulus, E	100 MPa
Length, h	25 cm
Density, ρ	1 g/cm^3
Mass, m_p	1 g
Volume, V_p	1 cm^3
Initial velocity, v_o	$v(x,0) = v_o \sin(\beta x)$ mm/s ($\beta = \pi/(2L)$ and $v_o = 1$ mm/s)


weighted velocity-time graph for both numerical and analytical solution
numerical s.
analytical s.
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
0
20
40
60
80
100
120
time(s)

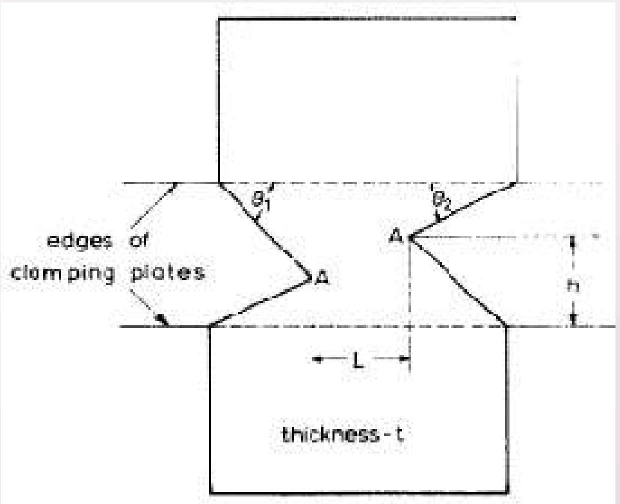

position-time graph of second material point for both numerical and analytical solu
numerical s.
analytical s.
position(cm)
2.92
2.91
2.9
2.89
2.88
2.87
2.86
2.85
0
20
40
60
80
100
120
time(s)


energy-time graph for numerical solution
strain energy
kinetic energy
total energy
energy(10^-9 x J)
time(s)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0
20
40
60
80
100
120


Total energy
Kinetic energy
Internal energy
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0
10
20
30
40
50
60
70
80
90

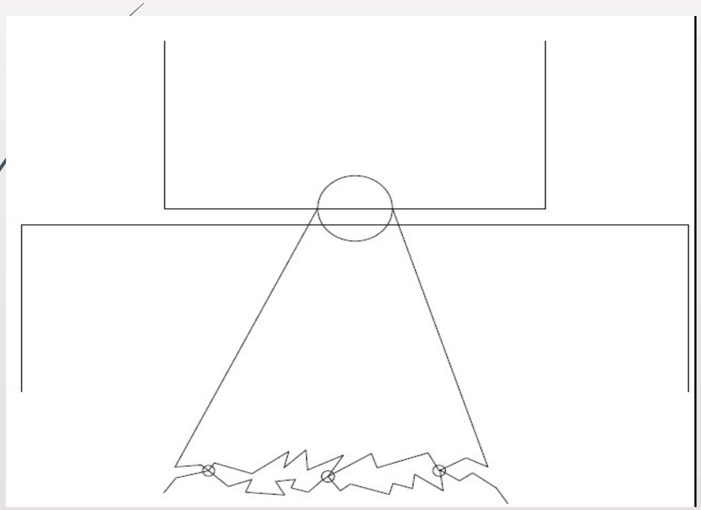
Wear Mechanism

- The experiment was performed by Brockley and Fleming in 1965 based on model copper junction experiments to observe metallic wear. Experiment reveals that a linear relationship exists on wear rate and simulated load area for series of models plotting.

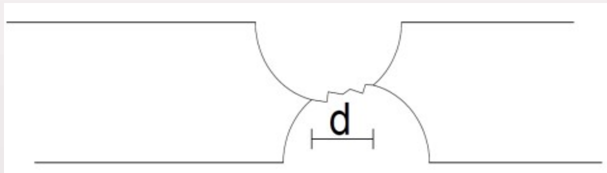


Geometry of the model junction

- Contacting surfaces lead to adhesive wear due to adhesive force between them. At the end of experiment, it is concluded that all junctions that are used lead to a wear particle.



Demonstration of contact surfaces



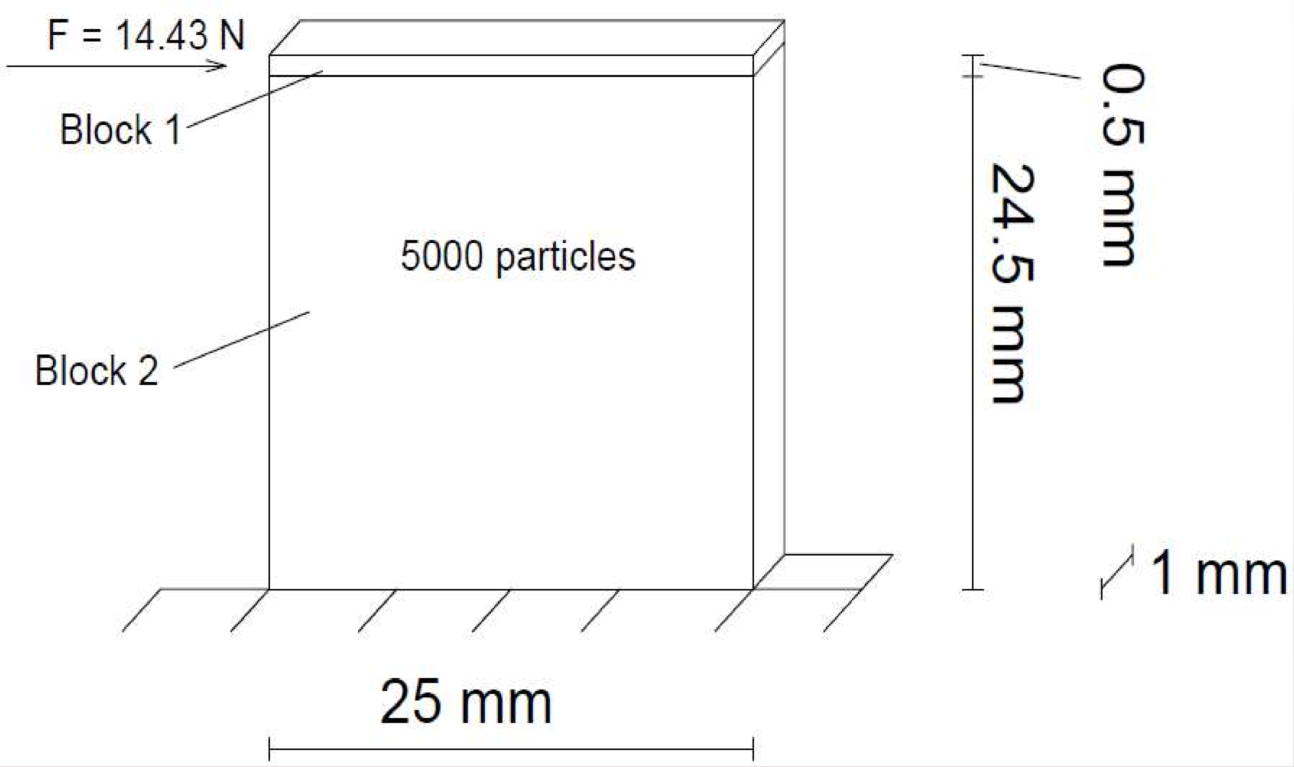
$$d^* = \lambda \frac{\Delta w}{\sigma_j^2} G$$

Critical length scale to generate debris particles

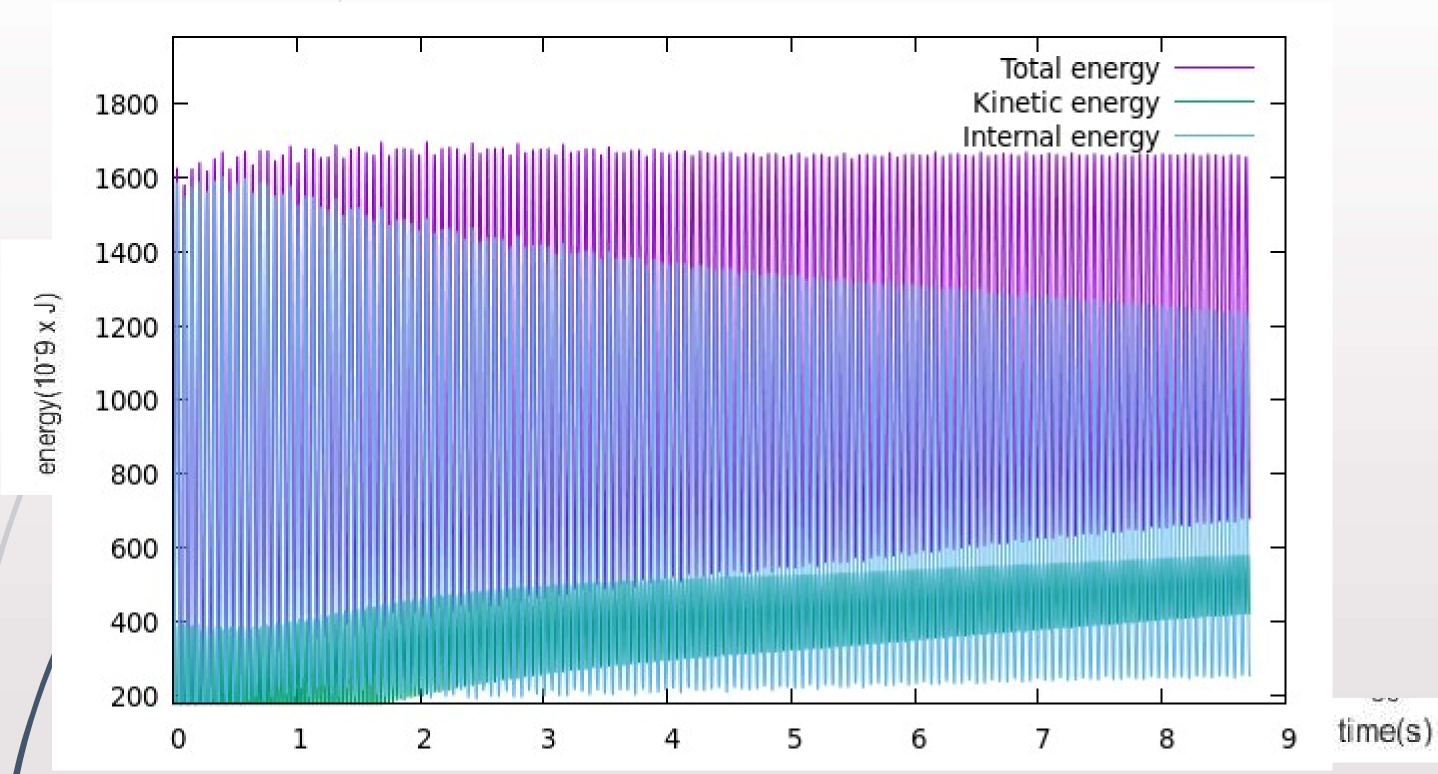
Third Application: Applied shear on a box

Property	**Value**
Young's modulus, E	128000 MPa
Length of x, y, z direction respectively	25 mm, 25 mm, 1 mm
Density, ρ	8.96×10^{-3} g/cm^3
Mass, m_p	9.21 g
Poisson ratio, v	0.3
Applied force	14.43 N

Figure



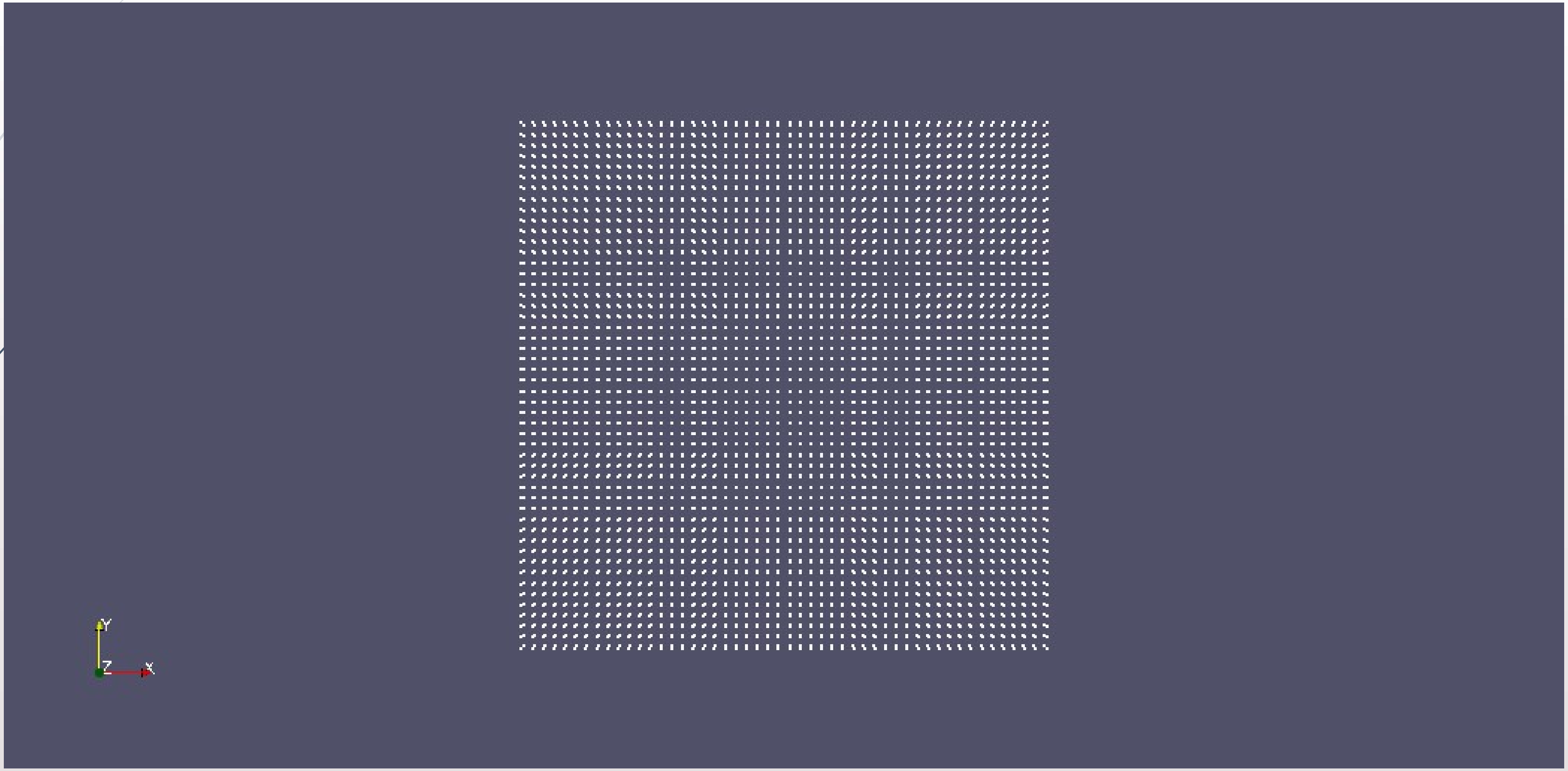
Energy plot



- Transient regime is the case

- Total energy cycles show oscillations that they are continuous until boundary conditions will make it stable if longer loop time is applied as assumed.

Simulation



Conclusion

- No advection term appears in MPM method so it allows easier treatment large deformation.
- Easy to implement 1D and with respect to FEM has advantages tested for complex problems as seen in Tsinghua study.
- Total energy is constant in the system.
- There is no damping in MPM. Thus, it oscillates but boundary conditions make it stable after a while.

Further Study

- To reach precise constant total energy from third application, code could be run for longer time.
- From the different trials for third example, it was seen that separation of particles include formation of new surfaces which is fracture at the end. Implementations are required to obtain undistorted behaviour such as enlarging background grid.
- To get better capture of fracture behavior, implementing different constitutive laws (plasticity, hardening behavior) would be helpful to understand fracture behaviour where wear formation occurs.

- Conservation of mass $\rho J = J_o$
- Conservation momentum $\sigma \cdot \nabla + \rho b = \rho \dot{v}$
- Conservation of energy $\rho \dot{e} = D : \sigma$
- Constitutive equation $\sigma^{\nabla} = \sigma^{\nabla}(D, \sigma, \ldots)$
- Rate of deformation $D = \frac{1}{2}(L + L^T)$
- Boundary conditions $v|\Gamma u = \bar{v}$, $(n \cdot \sigma)|\Gamma t = \bar{t}$
- Initial conditions $v(X,0) = v_o(X)$, $u(X,0) = u_o(X)$

ρ	density
σ	Cauchy stress
∇	gradient operator
$\dot{v}$	acceleration
σ^{∇}	Jaumann rate
L	velocity gradient tensor

J	determinant of deformation gradient
b	body force per unit mass acting onthe continuum
$\dot{e}$	change of internal energy per unit mass
D	rate of deformation tensor
n	unit normal of traction boundary
Γu	Neumann boundary conditions (displacement boundary)
Γt	Dirichlet boundary conditions (traction boundary)