Discrete/Distinct Element Method

Sierra Melton
Boning Zhang
Jiayi Fan
Yibo Song
Salih Koc
Zihan Ren

PENNSTATE

1855

Outline

1. Introduction to DEM
2. Historical Perspective
3. General Principles
4. Governing Equations
5. Hand-calculation Example
6. Numerical example
7. Applications of DEM

Introduction

What is the Discrete Element Method (DEM)?

- Model of discrete matter
- Material is composed of several discrete particles
  - Particles can have different shapes and sizes

- Materials that can be simulated with DEM
  - Granular matter (e.g. sediment)
  - Powder
  - Granular flows and blocky masses (e.g. rockslides)
  - Bulk materials in storage
  - Liquids and solutions

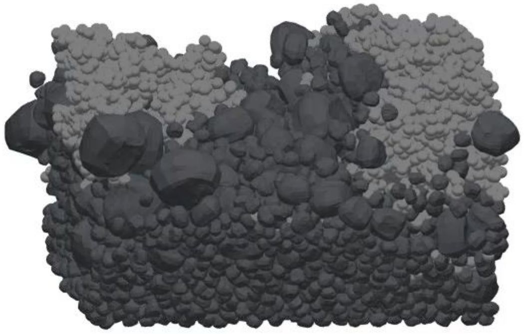


Image source: EDEM Webinar

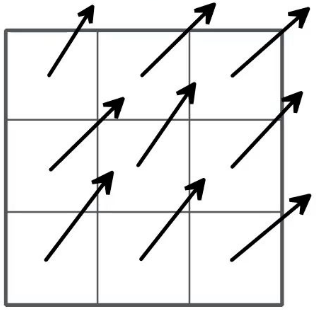
Introduction

When it comes to simulating particulate systems, two main modeling approaches can be identified –continuum (Eulerian) and discrete (Lagrangian).

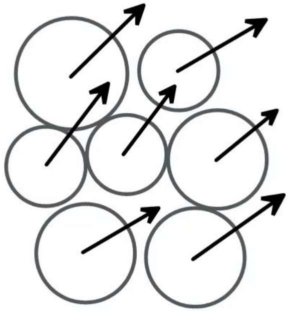
Continuum	*Discrete*
Continuous systems	Discontinuous, granular media
Assumes granular substance fills the space it occupies	Models behavior of individual particles
Relates stresses and strains through constitutive equations	Overall system behavior results from individual particle interactions
Suitable when length scale of importance is higher	Good for investigating phenomena occurring at particle length scale

Introduction

FEM (continuous) vs. DEM (discrete)



- Occupies entire space
- Cannot calculate relative particle movements and rotations



- Discontinuous
- Can calculate particle displacements and rotations

Image source: EDEM Webinar

Introduction

Advantages

- Performs well with **granular** and **discontinuous materials**
  - Good method for rock mechanics
- Enables visualization of position, velocity, force, stress, and strain networks
- Accounts for...
  - **Particle rotation**
  - **Time steps**
  - **Progressive failure**

Disadvantages

- Computationally intensive
- Limits number of particles and/or simulation length
- Difficult to capture...
  - Complex particle arrangements
  - Particle roughness
  - Particle breakage

PENNSTATE

Introduction

Industrial Applications

- Agriculture & food transport
- Oil and gas
- Mining
- Geomechanics
- Chemical engineering
- Civil engineering
- Pharmaceutical
- Powder metallurgy

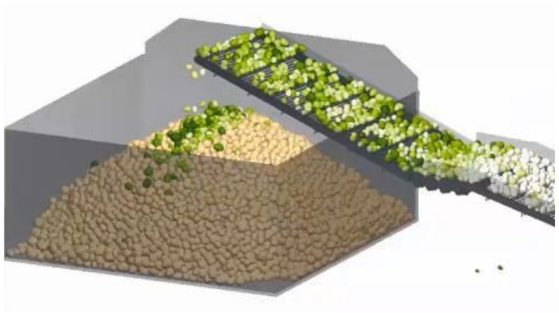


Image source: EDEM Webinar

PENNSTATE 1855

Historical Perspective

Newton, 1697
Theoretical principles

Alder & Wainwright, 1956
Molecular dynamics

Peter Cundall, 1971
Discrete Element Method
"A computer model for simulating progressive large-scale movements in blocky rock systems"

Williams, Hocking, & Mustoe, 1985
DEM as generalized FEM

Williams, Pande, & Beer, 1990
Application to geomechanics
Numerical Methods in Rock Mechanics

Shi, 1992
Discontinuous Deformation Analysis

General Principles

Newton mechanics

Conservation of momentum

Particle motion

- **Translational**
- **Rotational**

Numerical integration with time steps

Forces between particles

Force displacement laws

- Friction
- Stiffness

Image source: EDEM Webinar

General Principles

Hard-sphere vs. Soft-sphere Methods

Hard-sphere

Impulsive forces

Exchange of momentum

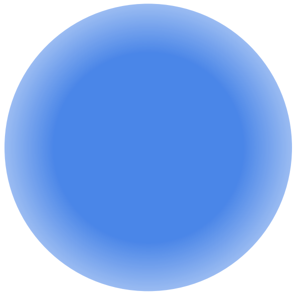
One collision at a time

Soft-sphere

Rigid particles but small overlaps allowed

Evaluates forces accurately

Simultaneous contacts possible



General Principles

Hard-sphere vs. Soft-sphere Methods

Soft-sphere

Rigid particles but small overlaps allowed

Evaluates forces accurately

Simultaneous contacts possible

Most common and accurate

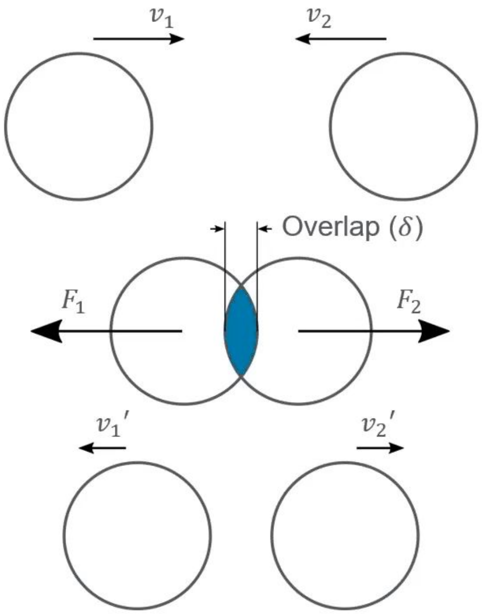


Image source: EDEM Webinar

General Principles

Forces and Contact Force Models

- **Contact force models**
  The contact models relate the amount of overlap (tangential and normal) between two objects to determine the magnitudes of forces.

- **Other forces**
  Forces resulting from particle collisions are not the only ones present in DEM.
  The effects of particle body forces like gravity or noncontact forces like electrostatics or Van der Waals can also be simulated.

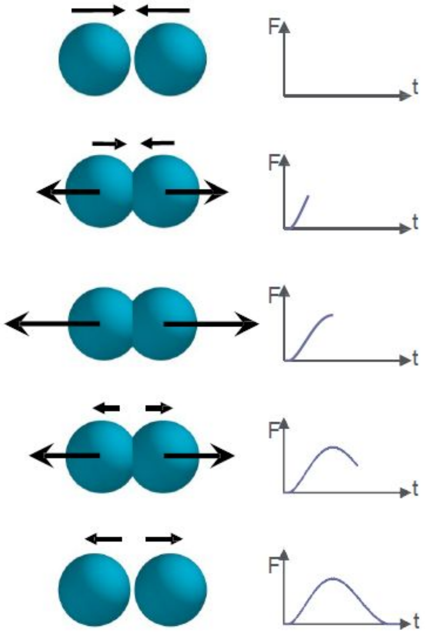


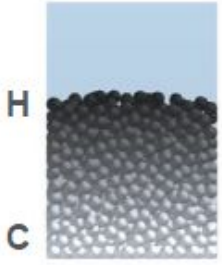
Image source: EDEM Webinar

General Principles

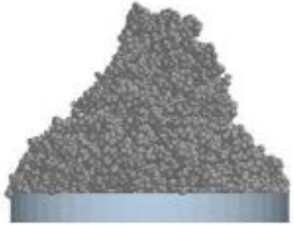
Models for Various Material Behavior

- Different types of material behavior can be simulated by a range of well-established models with DEM.
  - Dry granular material

  - Heat transfer



  - Cohesion

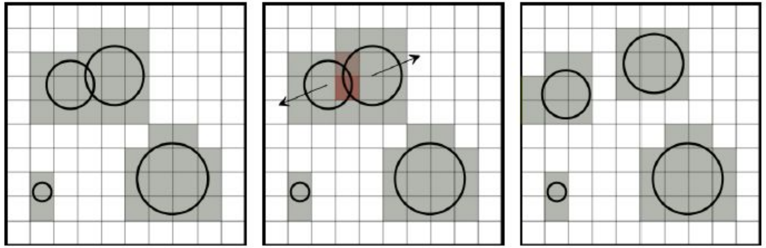


  - Electrostatics

Image source: EDEM Webinar

Contact Detection Algorithms

- Importance
- Steps of a typical contact detection algorithm



- Particle shape Representations
- Three main irregular particle shapes used in DEM simulations

Governing Equations: Particle Motion

Translational motion $ma = \sum F$

$$m\frac{dv}{dt} = F_g + F_c + F_{nc}$$

m: mass of particle

a: translational acceleration

v: translational velocity

Fg: gravitational force

Fc: contact forces between particles

Fnc: non-contact forces

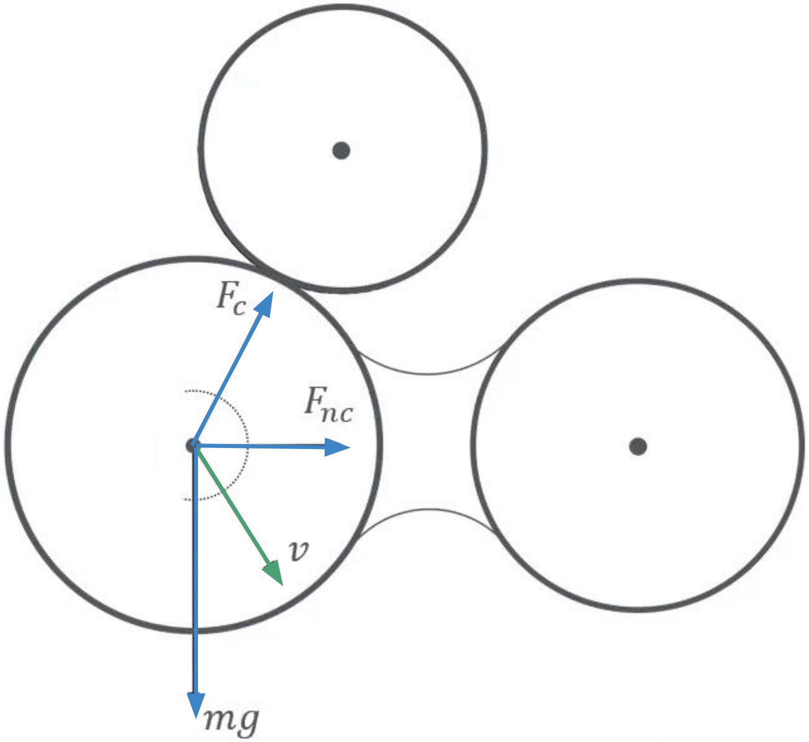


Image source: EDEM Webinar

Governing Equations: Particle Motion

Rotational motion

$$I\alpha = M$$

$$I\frac{d\omega}{dt} = M$$

I: moment of inertia

α: angular acceleration

ω: angular velocity

M: torque acting on particle

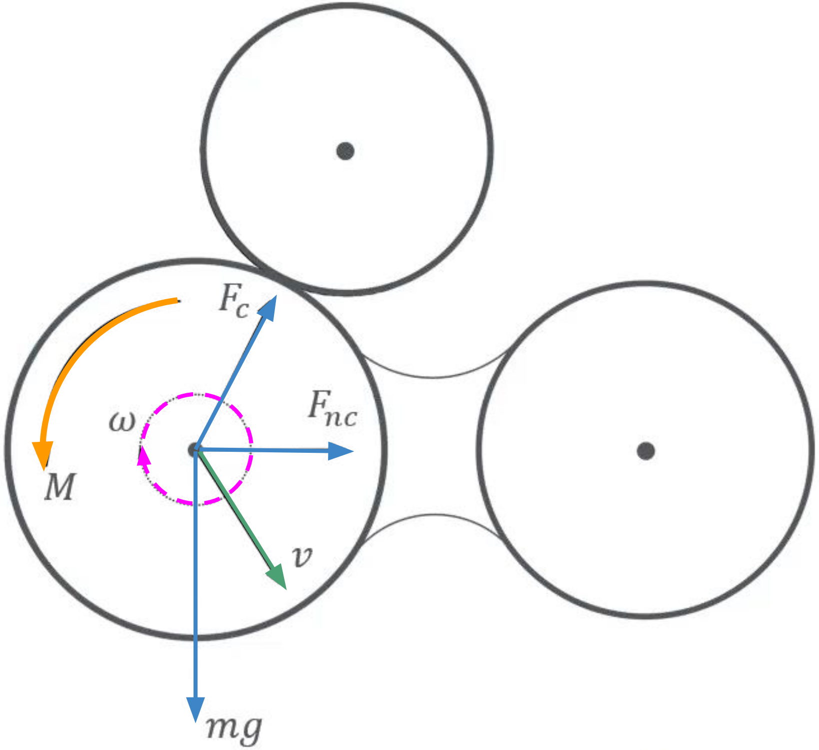


Image source: EDEM Webinar

Governing Equations: Other Forces

Contact/non-contact forces to consider

- Friction between particles
- Contact plasticity (recoil)
- Gravity between particles
- Cohesion/adhesion
- Molecular forces
  - e.g. electrostatic, Coulomb

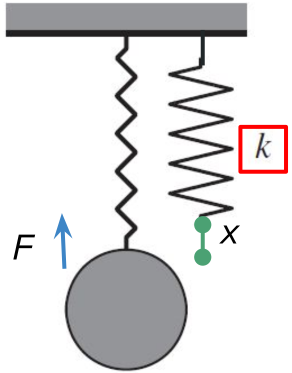
Hooke's Law

$$\boldsymbol{F} = -kx$$

F: restoring force

k: spring constant

x: extension length



Governing Equations: Time-stepping

Numerical Integration

After computing accelerations, integrate over ***time step (Δt)*** *to calculate particle* ***velocities*** *and* ***positions*** *(update at every time step)*

$$\boldsymbol{x}(t+\Delta t)=\boldsymbol{x}(t)+\boldsymbol{v}(t)\Delta t$$
$$\boldsymbol{v}(t+\Delta t)=\boldsymbol{v}(t)+\boldsymbol{a}(t)\Delta t$$

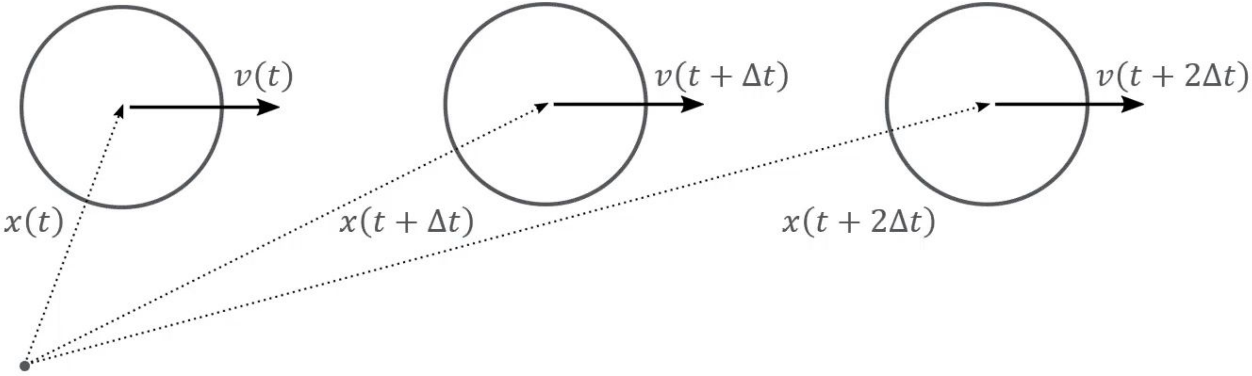
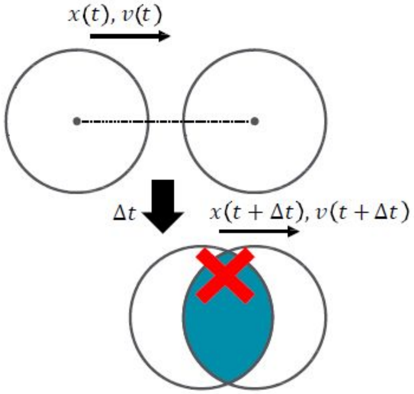


Image source: EDEM Webinar

Importance of Time Steps

Time steps chosen

Time step (Δt) has to be chosen sufficiently small (1e-4 to1e-6 s) for two main reasons: prevent excessive overlaps which result in unrealistically high forces and avoid effects of disturbance waves (Rayleigh waves).



Rayleigh surface waves

$$T_R = \frac{\pi R(\rho/G)^{1\ 2}}{0.1631\nu + 0.8766}$$

T_R is the Rayleigh time step
R is the particle radius
ρ is the density
G is the shear modulus
ν is the Poisson's ratio of the particle

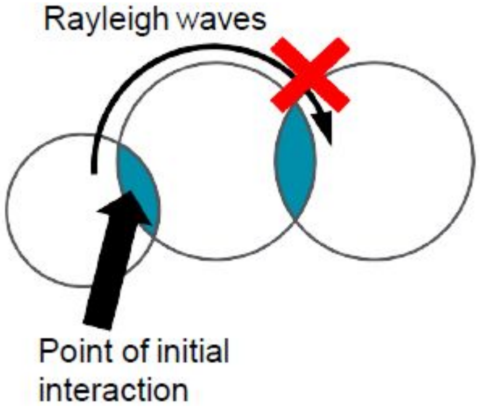
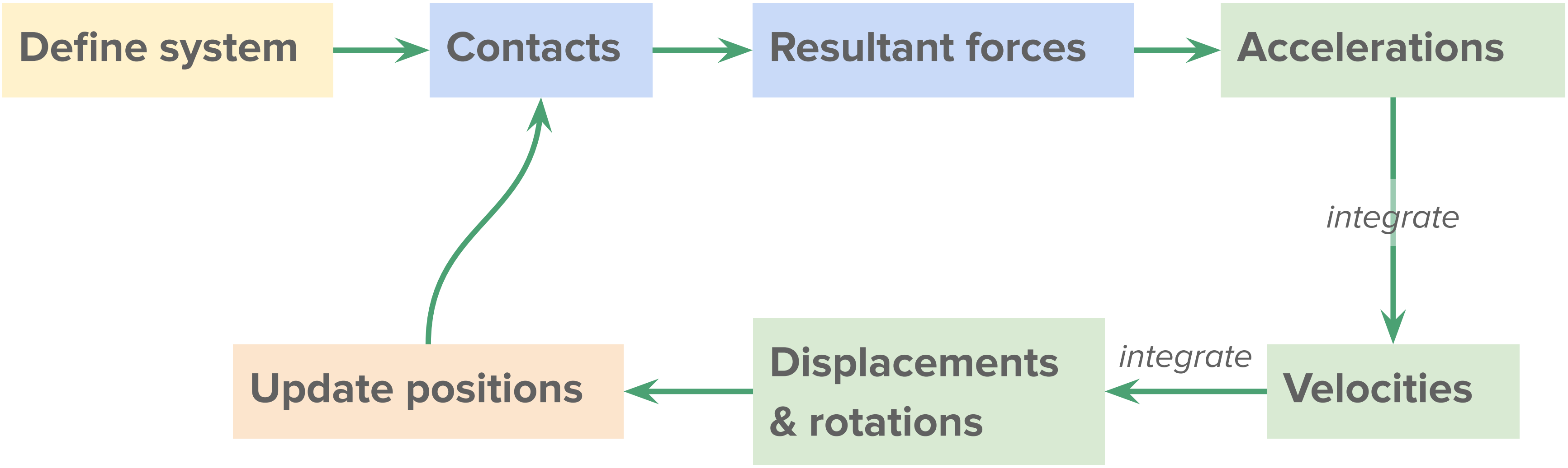


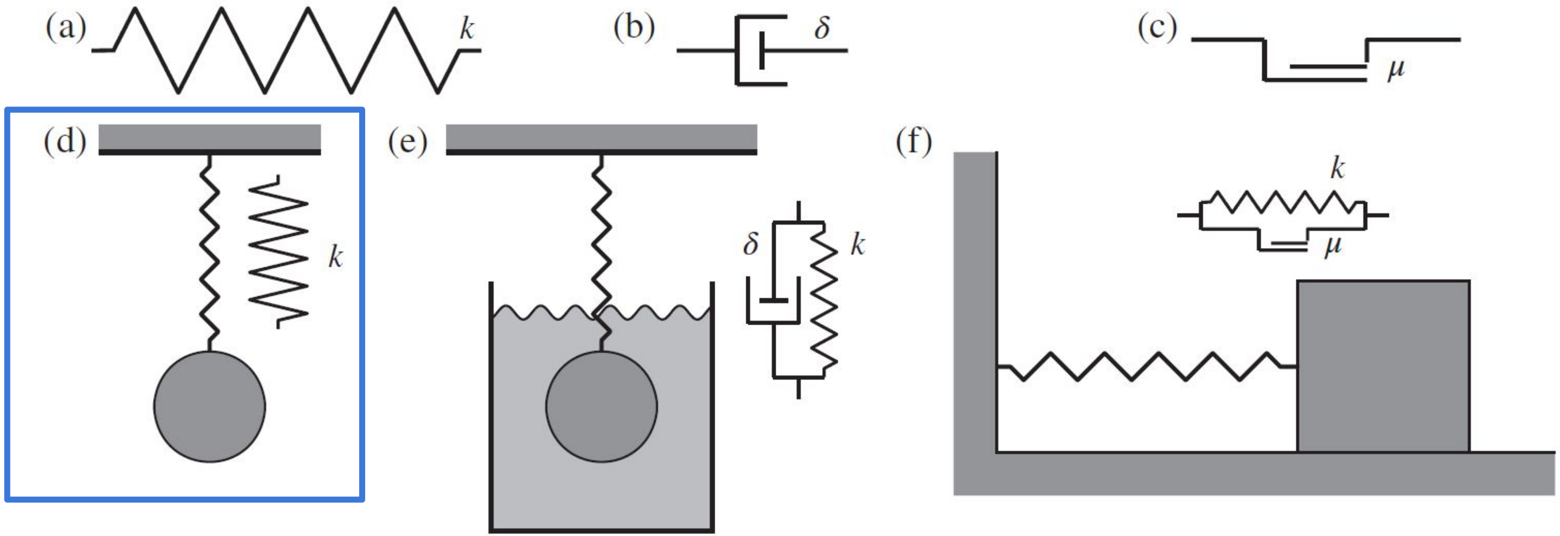
Image source: EDEM Webinar

Calculation Cycle



Hand-Calculation Example

Six classic mechanical model used in DEM analysis



Undamped linear oscillator and Damped linear oscillator:

PENNSTATE 1855

Hand Calculation Example

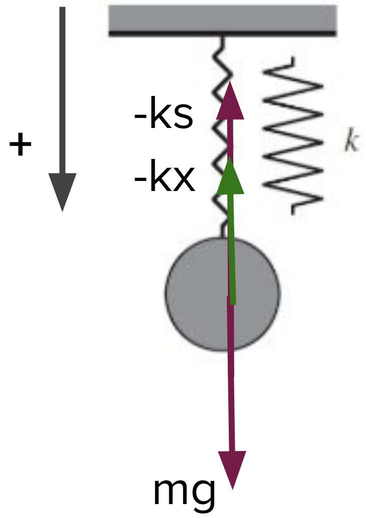
Undamped linear oscillator:

- At Equilibrium Point

a) Force created by Gravity F = mg

b) Restoration force of the spring to oppose the pulling force by gravity F = -ks

c) Restoration force of the spring F = -kx

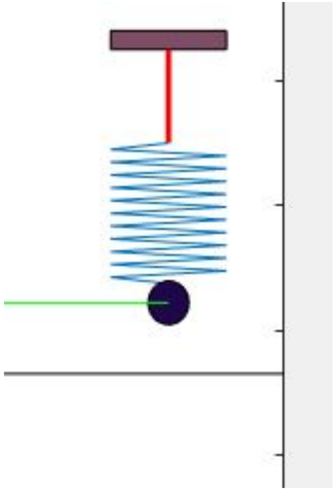


- The object release from the height above the equilibrium position at the beginning.

Hand-Calculation Example

Undamped linear oscillator:

Newton's Law: Total force applied to the body = Motion of the body



$$F = ma$$

$$-kx + \underline{mg - ks}$$

$\rightarrow 0$

$$-kx$$

$$a = \frac{d^2x}{dt^2}$$

$$ma = m\frac{d^2x}{dt^2}$$

$$-kx = m\frac{d^2x}{dt^2}$$

Hand-Calculation Example

Undamped linear oscillator:

$$m\frac{d^2x}{dt^2} + kx = 0 \;\rightarrow\; x = e^{nt} \;\rightarrow\; mn^2 + k = 0 \;\rightarrow\; n_{1,2} = \pm\sqrt{-\frac{k}{m}} = \pm\sqrt{\frac{k}{m}}\,i = \pm\omega_0 i$$

$$x(t) = C_1\cos(\omega_0 t) + C_2\sin(\omega_0 t) \qquad x(t) = A\cos(\omega_0 t - \phi)$$

$$B.C.1:\; x(t=0) = x_0 \;\rightarrow\; A = x_0$$

$$B.C.2:\; x(t\rightarrow\infty) = 0 \;\rightarrow\; \phi = \tan^{-1}\left(\frac{A\sin\phi}{A\cos\phi}\right)$$

A: the amplitude of the displacement

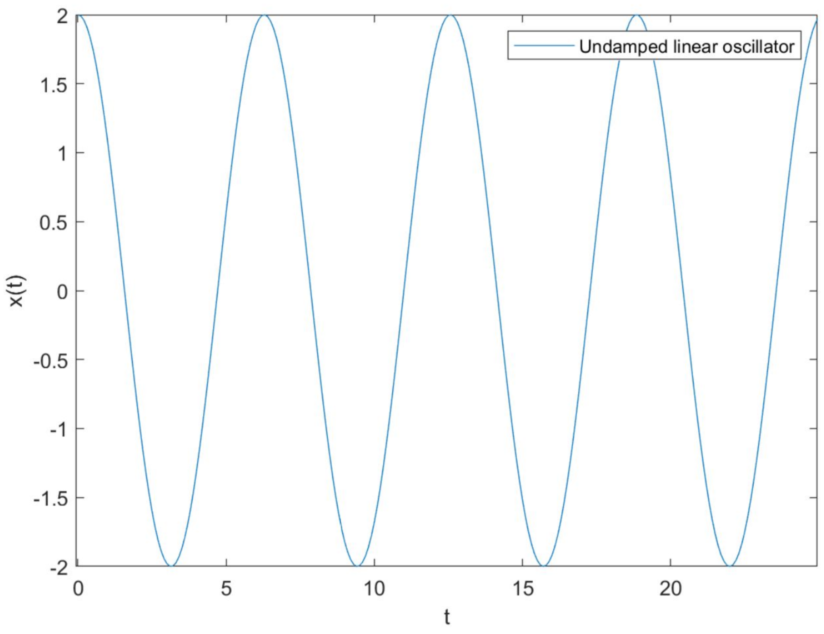
ϕ: the phase shift or phase angle of the displacement.

Hand-Calculation Example

Undamped linear oscillator:

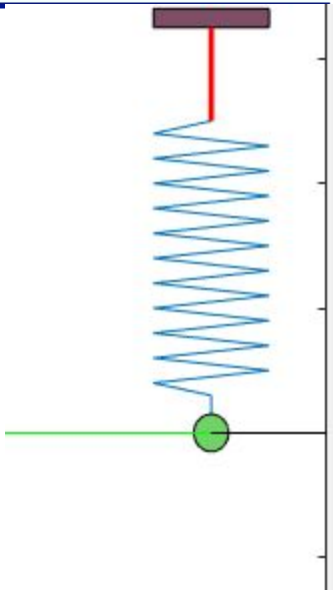
$$x(t) = A\cos\left(\sqrt{\frac{k}{m}}t - \phi\right)$$

$$v = \frac{dx}{dt} = -C \cdot \sin\left(\sqrt{\frac{k}{m}}t - \phi\right)$$



Hand-Calculation Example

Damped linear oscillator:



Newton's Law: Total force applied to the body = Motion of the body

$$F = ma$$

$$-kx + \underbrace{mg - ks}_{0} - \beta\frac{dx}{dt}$$

$$a = \frac{d^2x}{dt^2}$$

$$ma = m\frac{d^2x}{dt^2}$$

$$-kx - \beta\frac{dx}{dt}$$

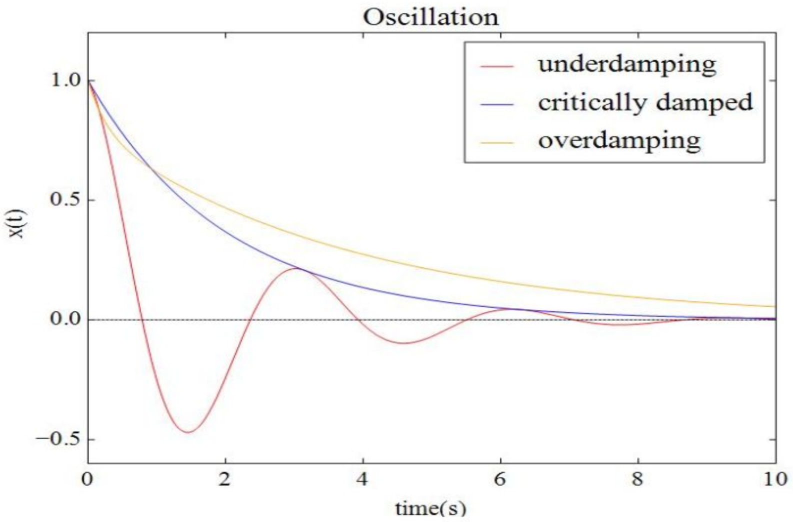
$$-kx - \beta\frac{dx}{dt} = m\frac{d^2x}{dt^2}$$

Hand-Calculation Example

Damped linear oscillator:

$$m\frac{d^2x}{dt^2}+\beta\frac{dx}{dt}+kx \rightarrow x=e^{nt} \rightarrow mn^2+\beta n+k=0 \rightarrow n_{1,2}=-\frac{\beta\pm\sqrt{\beta^2-4mk}}{2m}$$

$$\begin{cases} \beta^2-4mk>0\,, Overdamped \\ \beta^2-4mk=0\,, Critical\ damping \\ \beta^2-4mk<0\,, Underdamped \end{cases}$$



Hand-Calculation Example

Damped linear oscillator:

$$n_{1,2} = -\frac{\beta \pm \sqrt{\beta^2 - 4mk}}{2m} \qquad \omega = \frac{\sqrt{|\beta^2 - 4mk|}}{2m} \qquad n_{1,2} = -\frac{\beta}{2m} \pm i\omega$$

$$x(t) = C_1 e^{n_1 t} + C_2 e^{n_2 t} = C_1 e^{\left(-\frac{\beta}{2m} + i\omega\right)t} + C_2 e^{\left(-\frac{\beta}{2m} - i\omega\right)t}$$

$$x(t) = C_1 e^{t\left(-\frac{\beta}{2m}\right)} * e^{i\omega t} + C_2 e^{t\left(-\frac{\beta}{2m}\right)} * e^{-i\omega t}$$

$$x(t) = e^{t\left(-\frac{\beta}{2m}\right)} * \left(C_1 e^{i\omega t} + C_2 e^{-i\omega t}\right)$$

$$x(t) = e^{t\left(-\frac{\beta}{2m}\right)} * [C_1' \cos(\omega t) + C_2' \sin(\omega t)]$$

$$x(t) = A e^{t\left(-\frac{\beta}{2m}\right)} * \cos[\omega t - \phi]$$

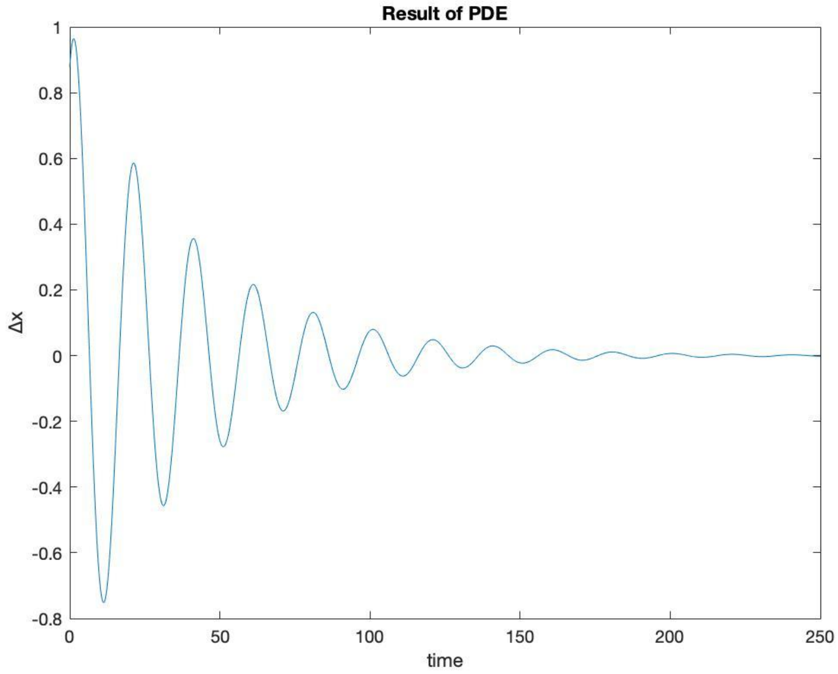
Hand-Calculation Example

Damped linear oscillator:

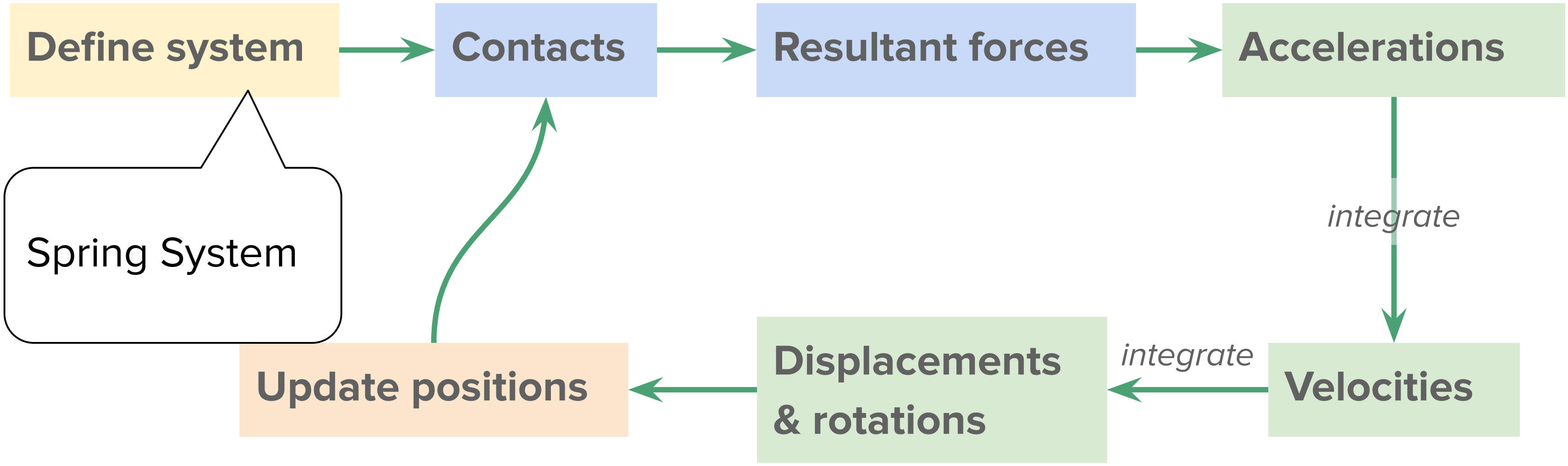
$$x(t) = Ae^{t\left(-\frac{\beta}{2m}\right)} * \cos[\omega t - \phi\,]$$

A: the amplitude of the displacement

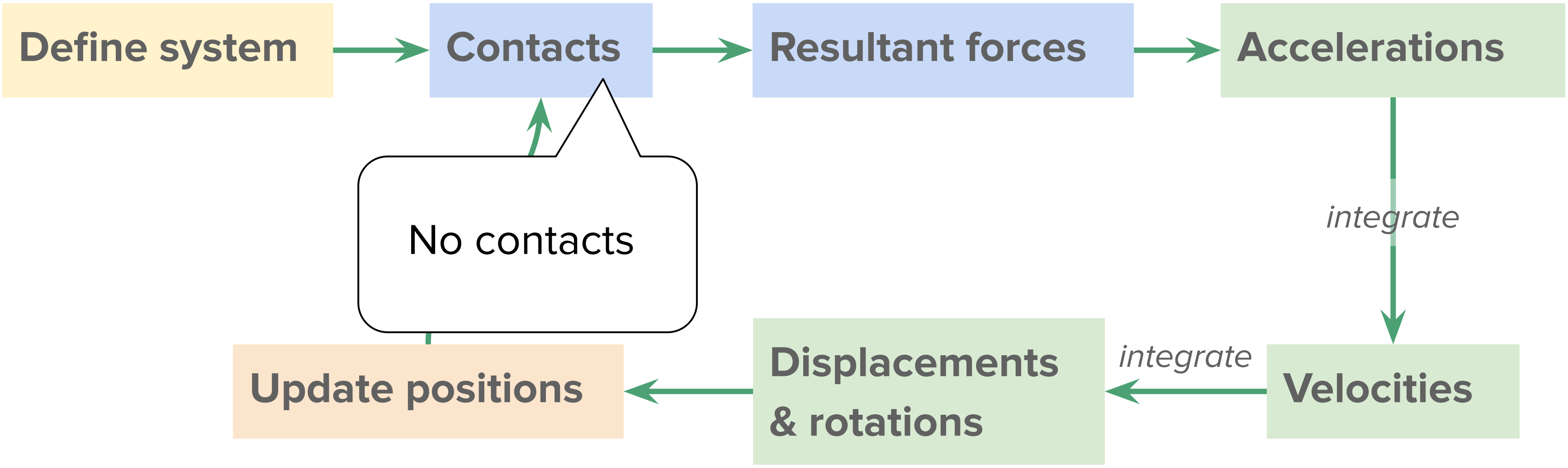
ϕ: the phase shift or phase angle of the displacement.



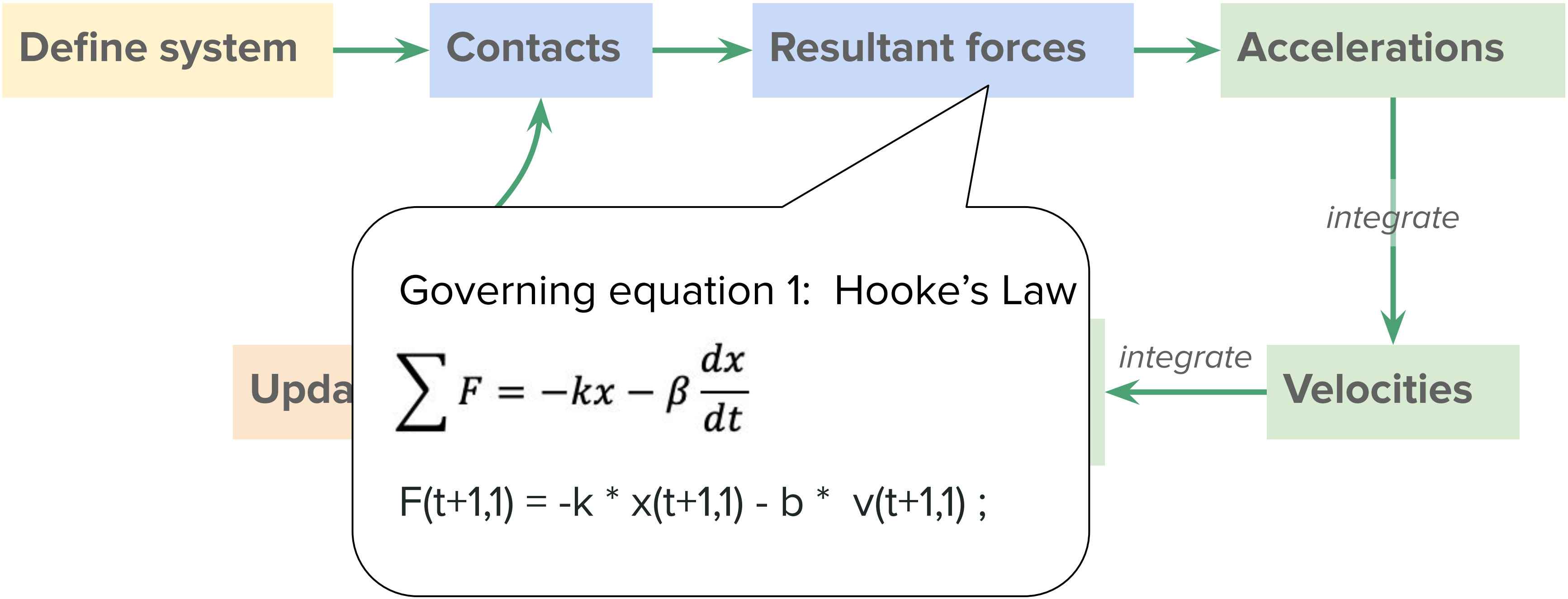
Calculation Cycle



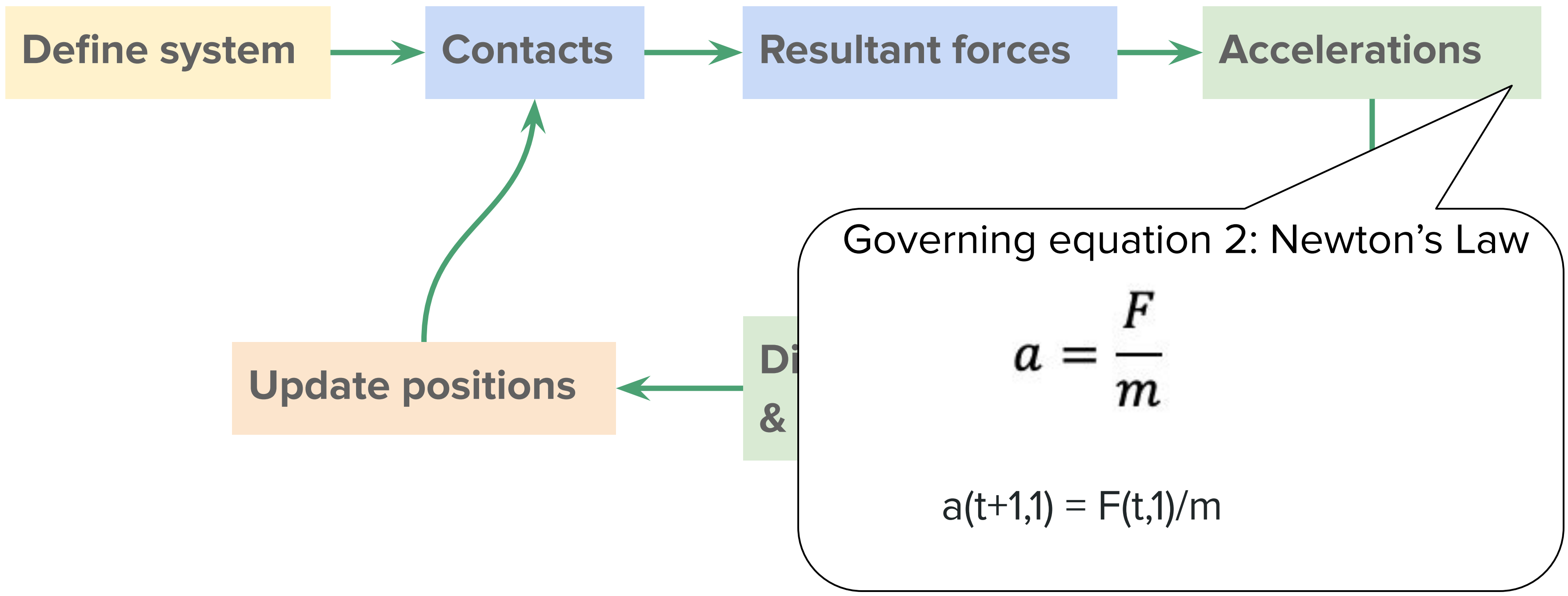
Calculation Cycle



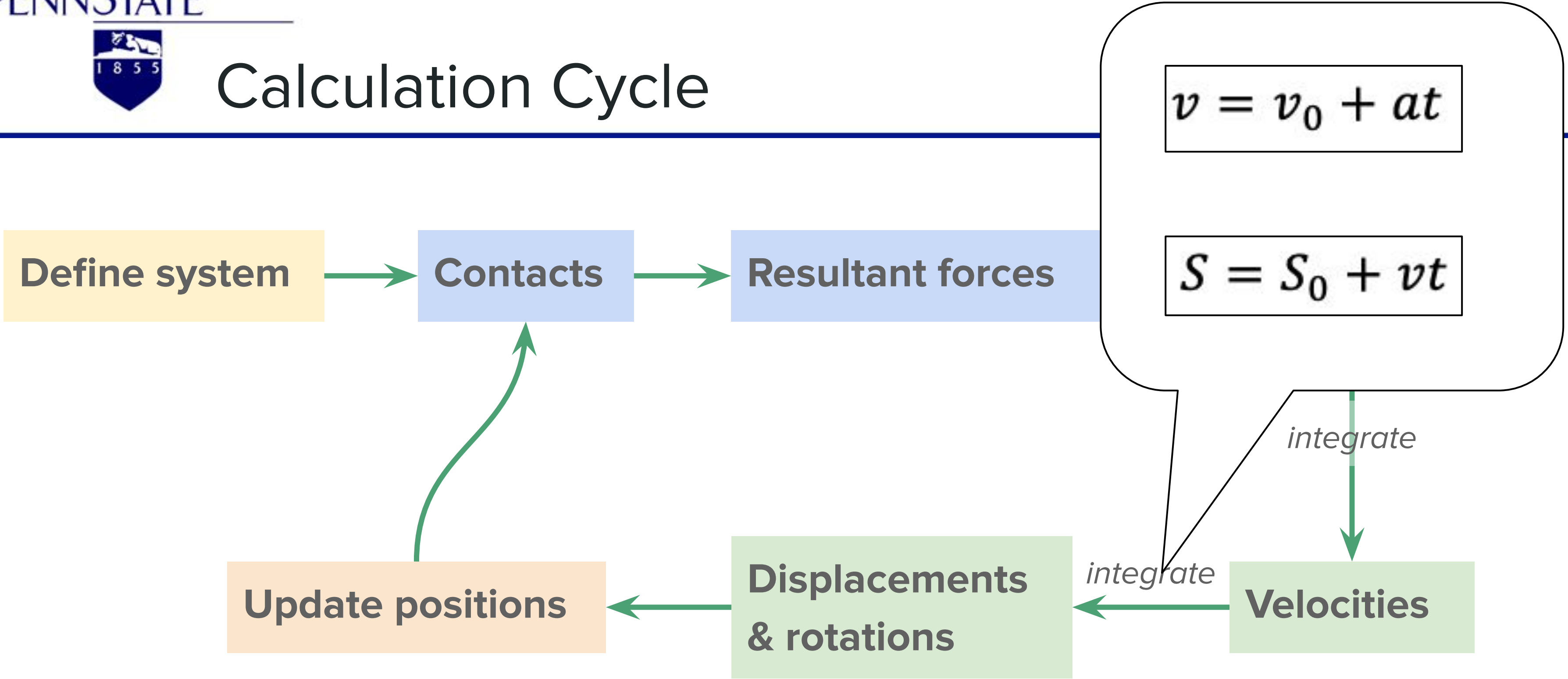
Calculation Cycle



Calculation Cycle



Calculation Cycle



Hand-Calculation Example

Code

```
g = 9.81;
k = 0.2;
m = 2;
b = 0.3;
TimeStep = 0.1;

x = zeros(1000,1);
x(1,1) = -1;
x(2:1000,1) = nan;
```

```
v = zeros(1000,1);
v(1,1) = 0;
v(2:1000,1) = nan;

F = zeros(1000,1);
F(1,1) = (k * x(1,1)-b*v(1,1));
F(2:1000,1) = nan;

time = zeros(1,1000);
a = zeros(1000,1);
```

PENNSTATE 1855

Hand-Calculation Example

Code

```
g = 9.81;
k = 0.2;
m = 2;
b = 0.3;
TimeStep = 0.1;

x = zeros(1000,1);
x(1,1) = -1;
x(2:1000,1) = nan;
```

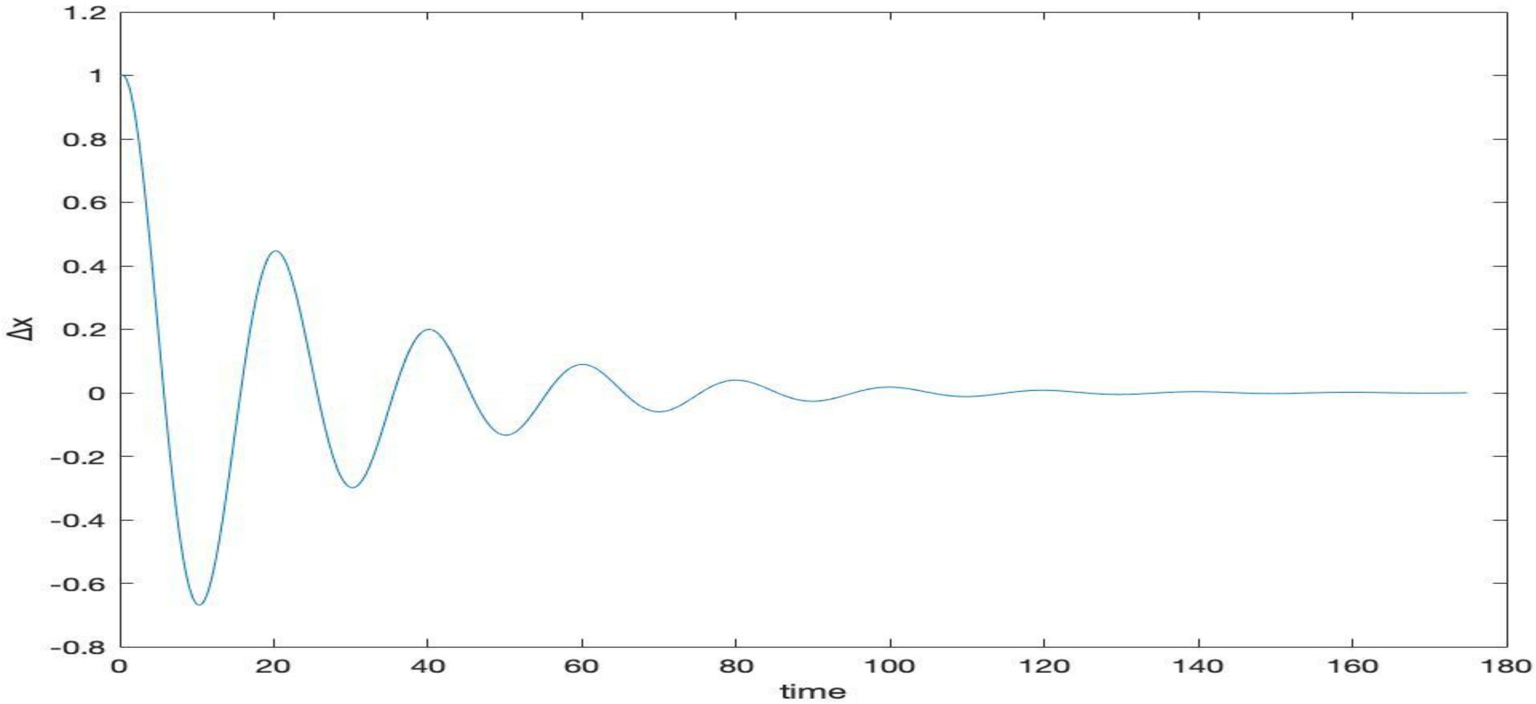
```
v = zeros(1000,1);
v(1,1) = 0;
v(2:1000,1) = nan;

F = zeros(1000,1);
F(1,1) = (k * x(1,1)-b*v(1,1));
F(2:1000,1) = nan;

time = zeros(1,1000);
a = zeros(1000,1);
```

Hand-Calculation Example

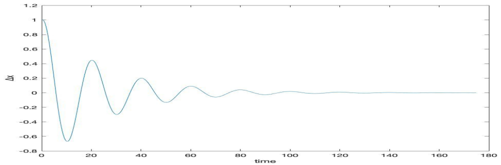
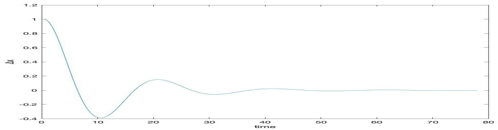
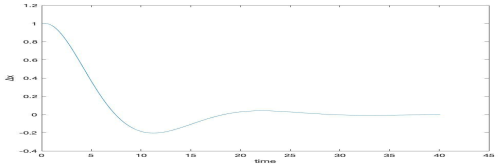
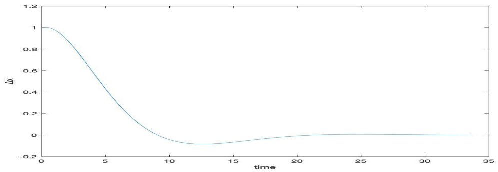
Code

```
for t = 1:1:1000
   time(1,t+1) = t * TimeStep ;

   a(t+1,1) = F(t,1)/m;

   v(t+1,1) = v(t,1) + a(t,1) * TimeStep;

   x(t+1,1) = x(t,1) + v(t,1)*TimeStep;

    if x(t+1,1) > 0
      if  x(t+1,1) <= 15
         F(t+1,1) = -k * x(t+1,1) - b *  v(t+1,1) ;
      else
         x(t+1,1) = 15;
         v(t+1,1) = 0;
         F(t+1,1) = -k * x(t+1,1);
      end
    elseif x(t+1,1) >= -15

         F(t+1,1) = (-k * x(t+1,1) - b *  v(t+1,1));

    else

         x(t+1,1) = -15;
         v(t+1,1) = 0;
         F(t+1,1) = -k * x(t+1,1);
    end


end

plot(time,-x);
```

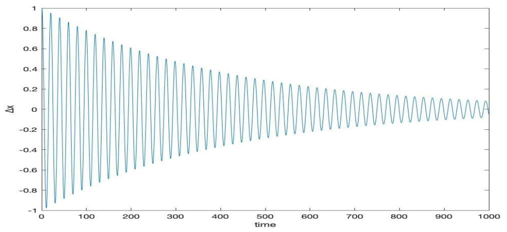
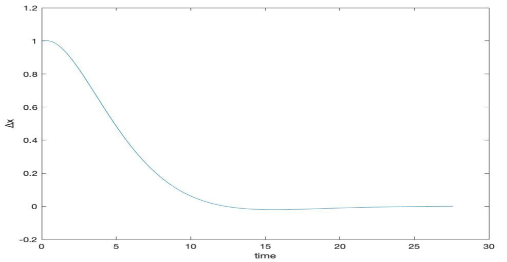
Hand-Calculation Example



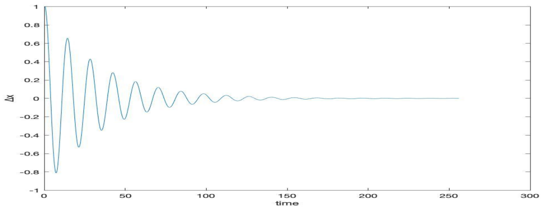
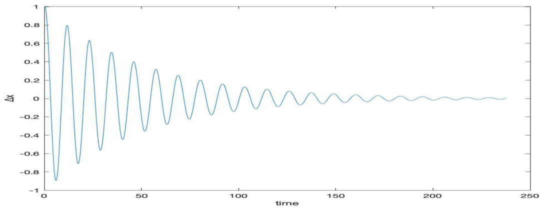
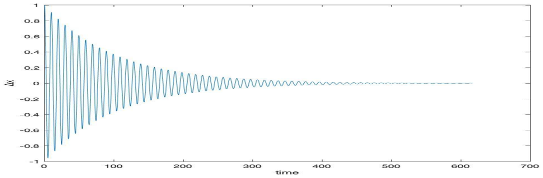
Hand-Calculation Example k = 0.2

Damping Coefficient	Time	Figure
0.2	174.7	
0.4	78	
0.6	40.1	
0.8	33.6	

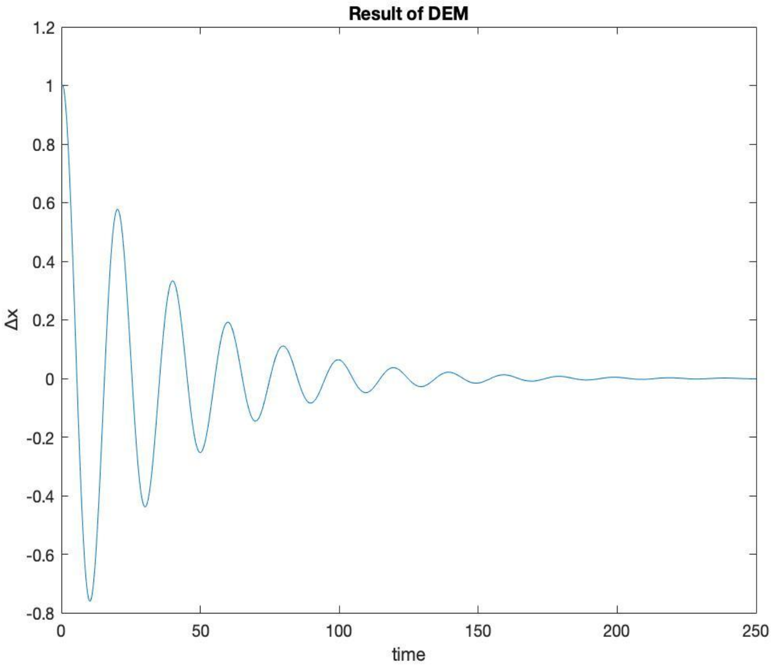
Hand-Calculation Example k = 0.2

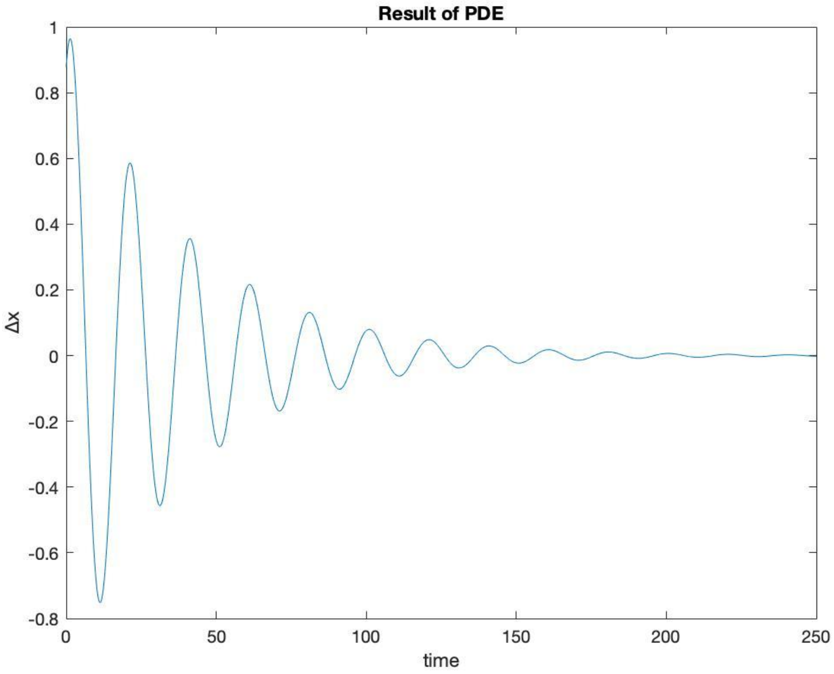
Damping Coefficient	Time	Figure
0.05	>1000	
0.99	27.6	

Hand-Calculation Example b = 0.2

Spring constant	Time	Figure
0.4	212	
0.6	237.5	
0.8	616.6	

Comparison between DEM result and PDE result





Numerical Example

Bouncing ball

Basic assumption:

- Particles are soft
- Continuity of force
- Viscous damping proportional to velocity

Damped/Dissipated

$$F(x) = \begin{cases} mg - Dv & for\ x \geq 0 \\ mg - kx - Dv & for\ x < 0 \end{cases}$$

Govern Equation

Without Dissipation

$$F(x) = \begin{cases} mg & for\ x \geq 0 \\ mg - kx & for\ x < 0 \end{cases}$$

```
function [dy]=bouncing_ball(t,y)
% bouncing ball without dissipation
global g
global k
if (y(2)>=0)
dy=[g
y(1)];
else
dy=[g-k*y(2)
y(1)];
end
return
```

Parameters:
K is set to be 100 and D is set to be 1.
The initial height is set to be 4
The time range is set to be [0 10]

PENNSTATE

1855

Numerical Example

Bouncing ball

Bouncing ball without dissipation

```
clear, format compact
global k, k=100;
global g, g=-9.81;
global D, D=1;
x0=4
v0=1

tspan=[0 10]
[t,y]=ode23('bouncing_ball',tspan,[v0 x0]);
[t1,y1]=ode23('bouncing_balldis',tspan,[v0 x0]);

figure(1)
plot(t,y(:,2),'ro', t1,y1(:,2), 'bo')
legend('Bouncing ball without dissipation', ...
    'Bouncing ball with dissipation')
```

```
function [dy]=bouncing_ball(t,y)
% bouncing ball without dissipation
global g
global k
if (y(2)>=0)
dy=[g
y(1)];
else
dy=[g-k*y(2)
y(1)];
end
return
```

```
function varargout = ode23(ode,tspan,y0,options,varargin)
%ODE23  Solve non-stiff differential equations, low order method.
```

Numerical Example

Bouncing ball

Bouncing ball with dissipation

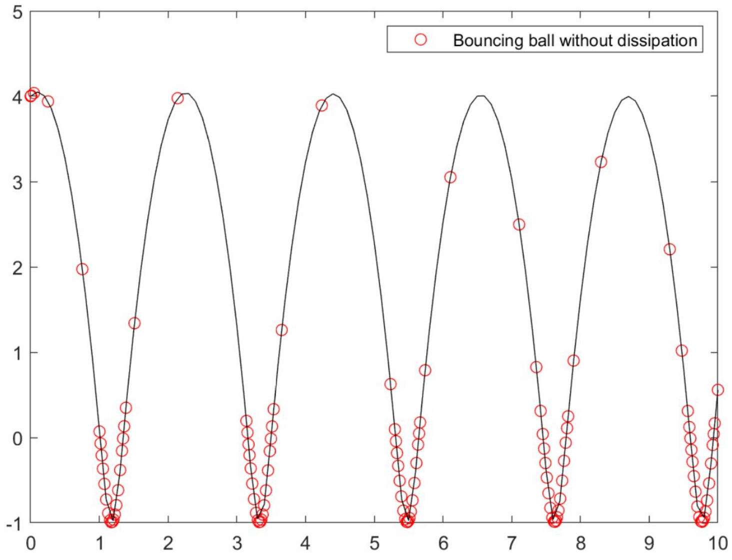
```
clear, format compact
global k, k=100;
global g, g=-9.81;
global D, D=1;
x0=4
v0=1

tspan=[0 10]
% [t,y]=ode23('bouncing_ball',tspan,[v0 x0]);
[t1,y1]=ode23('bouncing_balldis',tspan,[v0 x0]);

figure(1)
plot(t,y(:,2),'ro', t1,y1(:,2), 'bo')
legend('Bouncing ball without dissipation', ...
    'Bouncing ball with dissipation')
```

```
function [dy]=bouncing_balldis(t,y)
% bouncing ball without dissipation
global g
global k
global D
f_el=-k*y(2)
f_damp=-D*y(1)
f_tot=f_el+f_damp
if (sign(f_tot*f_el)<0)
f_tot=0
end

if (y(2)>=0)
dy=[g-D*y(1)
y(1)];
else
dy=[g+f_tot
y(1)];
end
return
```

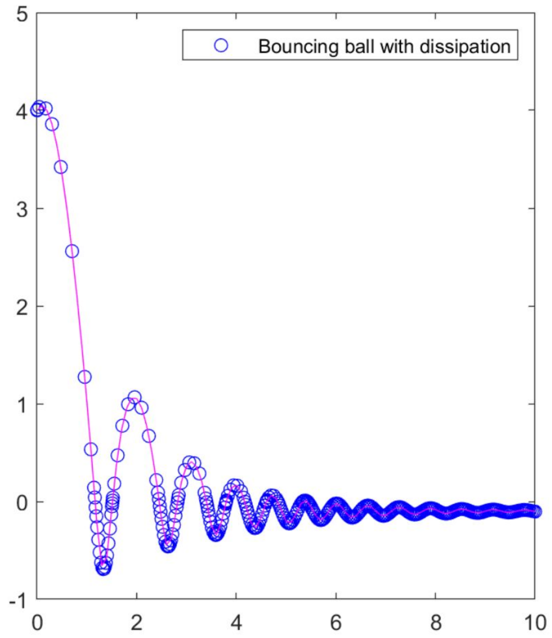
Numerical Example

Bouncing ball

Bouncing ball without dissipation



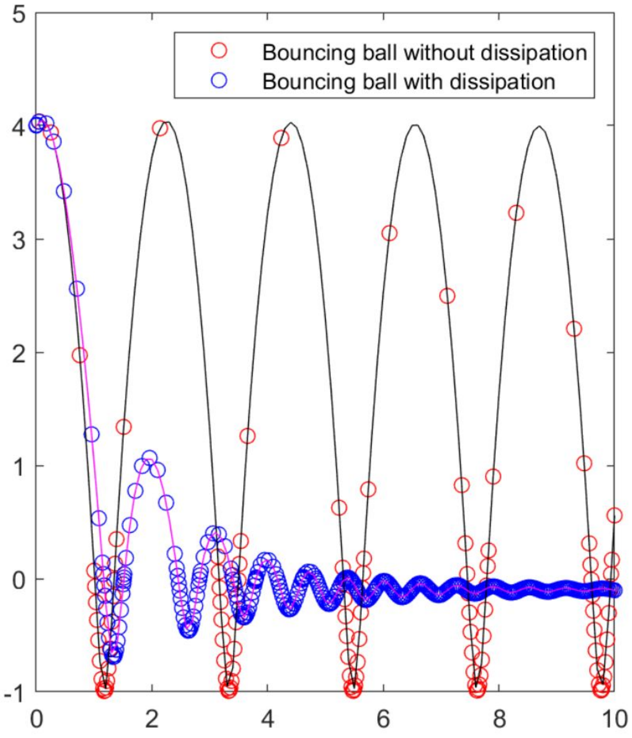
Bouncing ball with dissipation

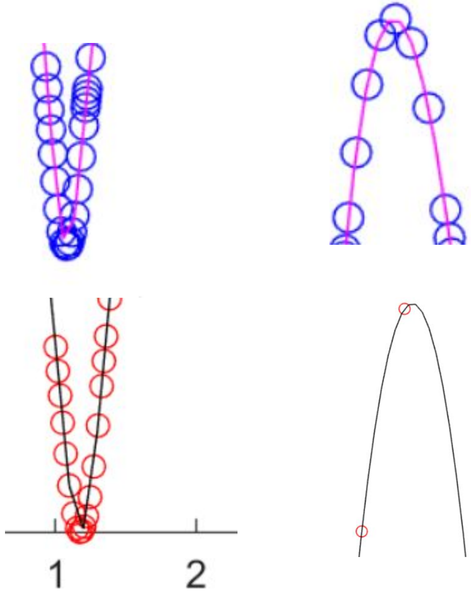


Numerical example

Bouncing ball

Comparison





dissipated

undissipated

- ➢ Bouncing ball with dissipation have a smaller time steps

PENNSTATE

Numerical Example

Modeling of Polygonal Particle-1D

Basic assumption:

- No dissipation /No damping
- Soft and round particle
- Perfect Elastic collision
- Rotation is ignored

External Force:

- Gravity force
- Elastic force

Initialization:

- Particle number (1 to 5)
- Same radius(0.5) , mass(1)
- Same Young Modulus (1000)
- Minimum(0) and Maximum vertical displacement(2*n+2)
- Initial vertical position and velocity: y0:[2 0 4 0 ... 10 0]
- End time: 4

Numerical Example

Modeling of Polygonal Particle - 1D

```
clear all
format compact
n_part=2
% initialize radius and mass
global rad, rad(1:n_part)=0.5;
global m, m(1:n_part)=1;
global E, E=1000; % Young's modulus
global lmax, lmax=2*n_part+2;
global lmin, lmin=0;
global g, g=-9.81;
% initialize positions and velocities=0
r0=2*[1:n_part];
v0=r0*0;
y0(1:2:2*n_part-1)=r0;
y0(2:2:2*n_part)=v0;
t_end=4
[t,y]=ode113('DEMround1D',[0 t_end],y0);
hold on
for i=1:n_part
plot(t,y(:,2*i-1),'bo-')
end
axis([0 max(t) lmin-.5 lmax+.5])
```

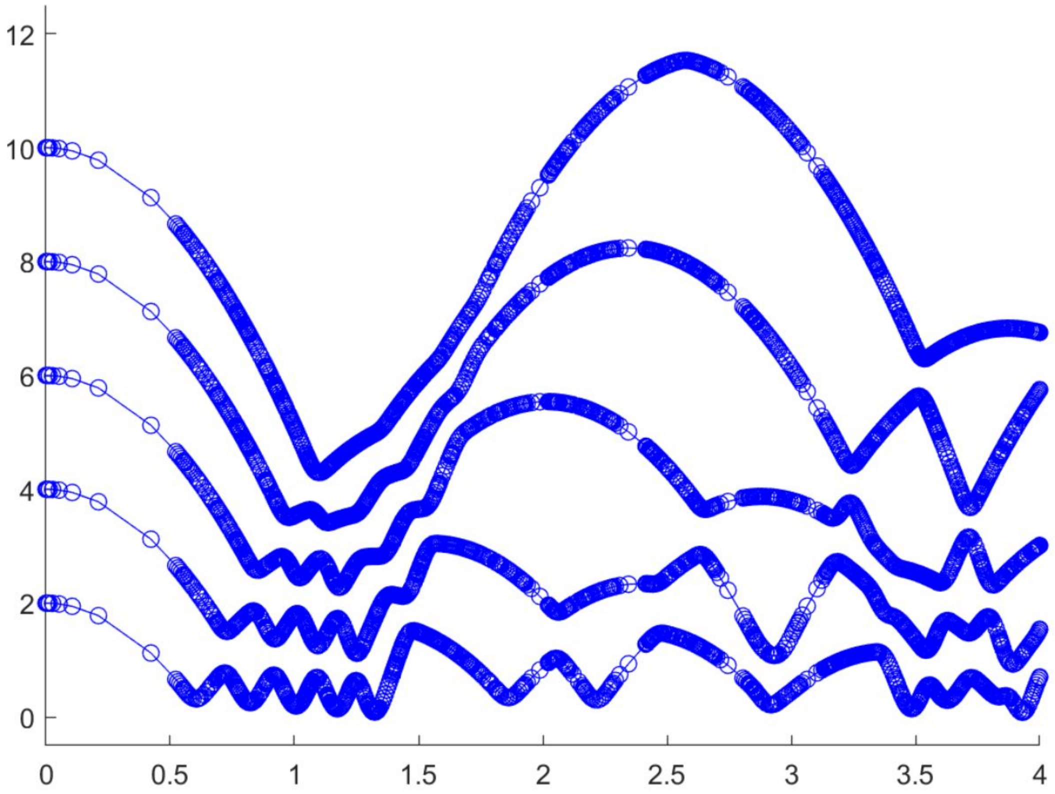
```
function [dydt]=DEMround1D(t,y);
global m rad E lmax lmin g
n_part=length(m);
if length(y)~=2*length(m)
error('length of y must be twice the length of m')
end
if length(rad)~=length(m)
error('length of r must be twice the length of m')
end
a=zeros(1,n_part);
for i_part=1:n_part
x1=y(2*i_part-1); % position of first particle
rad1=rad(i_part);
% Particle-Particle Interaction
for j_part=i_part+1:n_part
x2=y(2*j_part-1); % position of second particle
rad2=rad(j_part);
if (abs(x2-x1)<(rad(i_part)+rad(j_part))) % overlap
forcemagnitude=E*abs(abs(x1-x2)-(rad1+rad2));
forcedirection=sign(x1-x2);
f=forcemagnitude*forcedirection;
a(i_part)=a(i_part)+f;
a(j_part)=a(j_part)-f; % use action=reaction
end
end
% Particle-wall Interaction
if (x1-rad1)<lmin
a(i_part)=a(i_part)-E*((x1-rad1)-lmin);
end
```

particle interaction

Numerical example

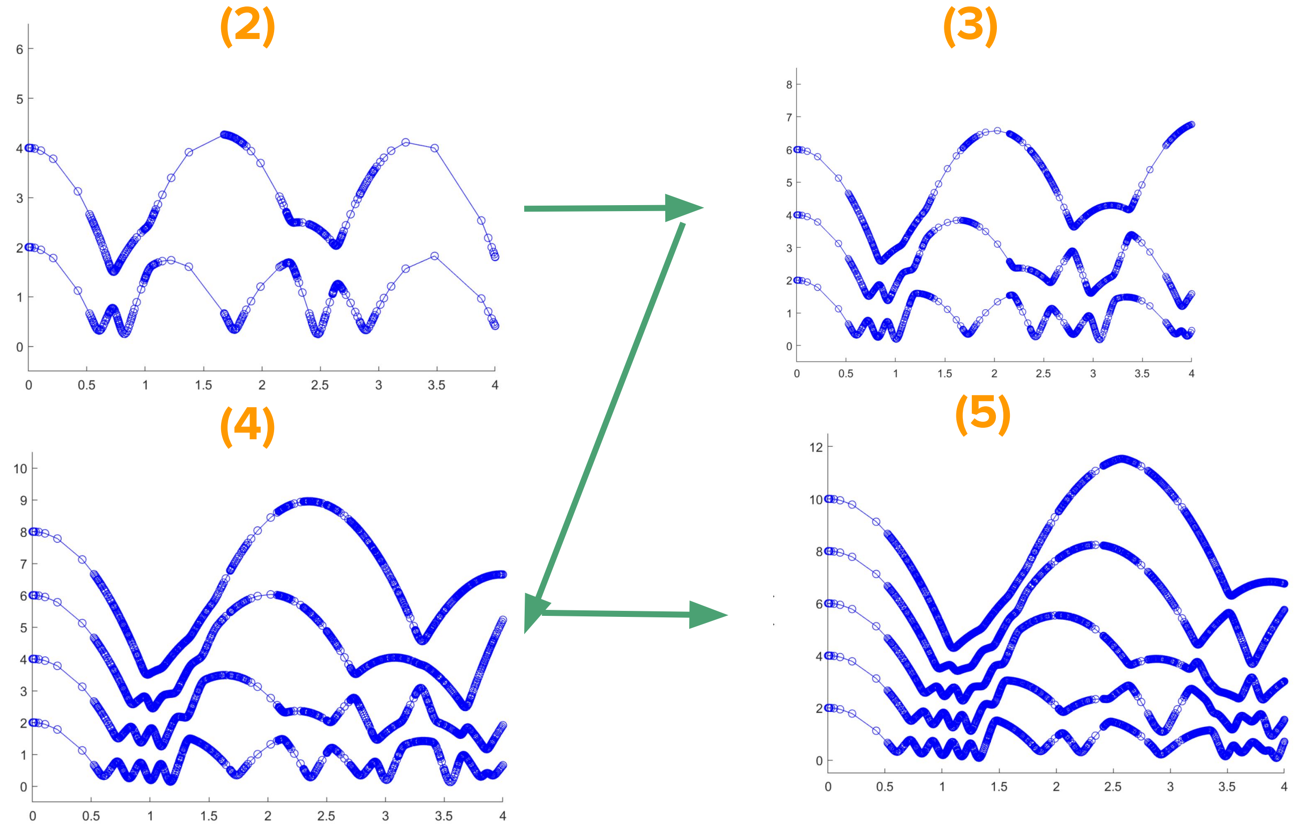
Modeling of Polygonal Particle-1D

Result



Numerical example

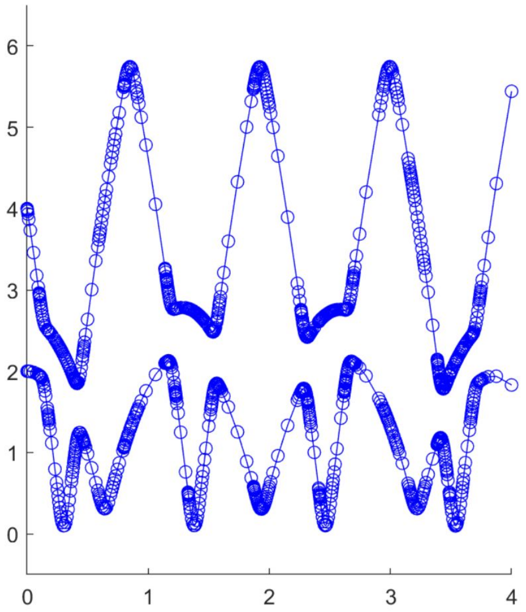
Modeling of Polygonal Particle-1D



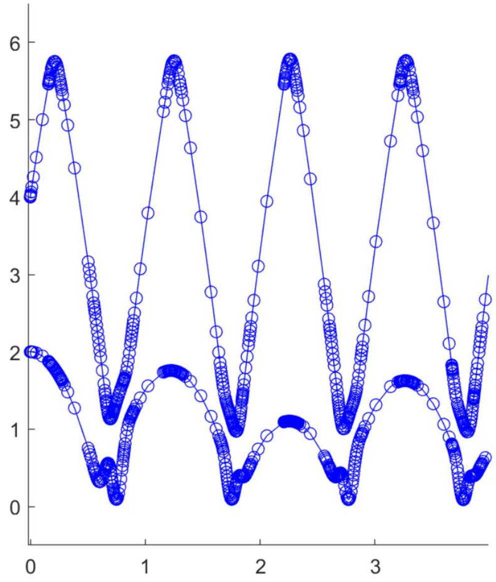
Numerical example

Modeling of Polygonal Particle-1D

Y0: [2 0 4 -10]



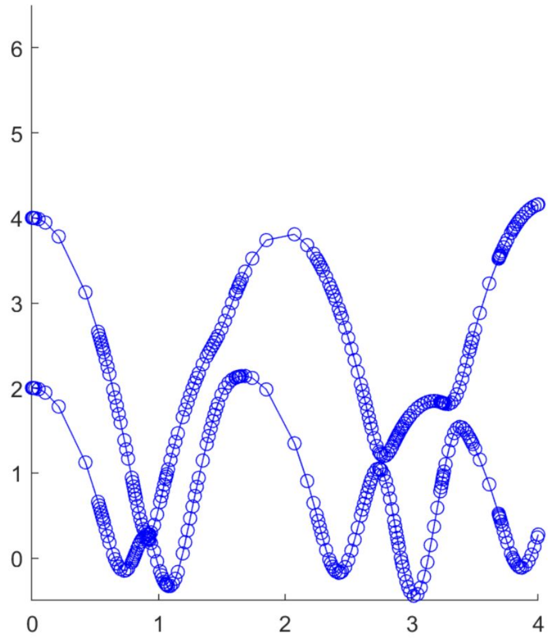
Y0: [2 0 4 10]



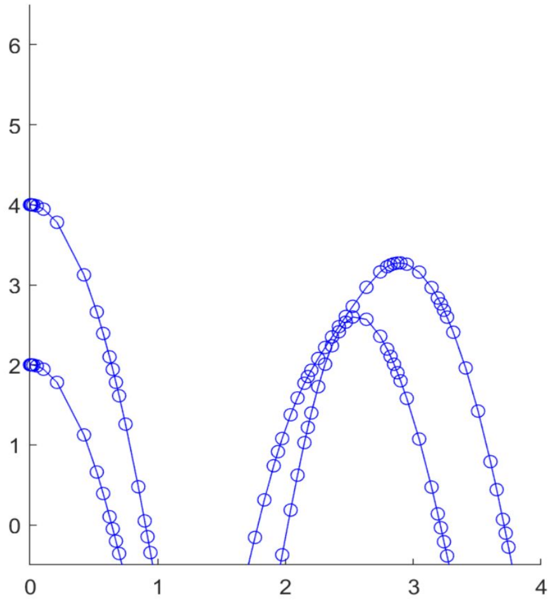
Numerical example

Modeling of Polygonal Particle-1D

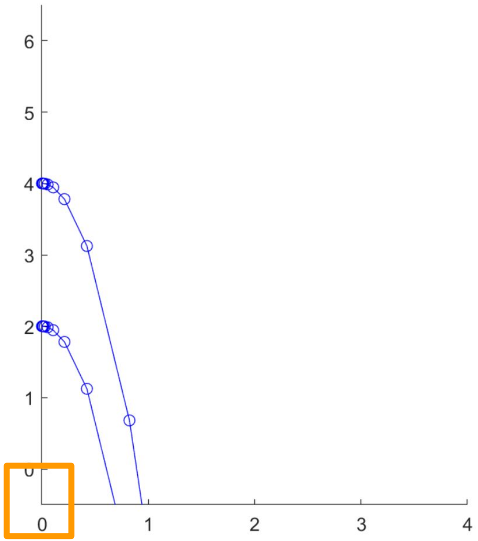
E = 100



E = 10



E = 0



Numerical example

Initialization:

- Particle number (18)
- Same radius(0.5) , mass(1)
- Same Young Modulus (10000)
- Minimum(0) and Maximum vertical displacement(2*n+2). Maximum and minimum horizontal displacement is the same with vertical.
- Initial position and velocity vector
  y0:[X1 randx1 Y1 randy1 ... X18 randx18 Y18 randy18]: 72*1 = 18 * 4
- End time: 5

Numerical Example

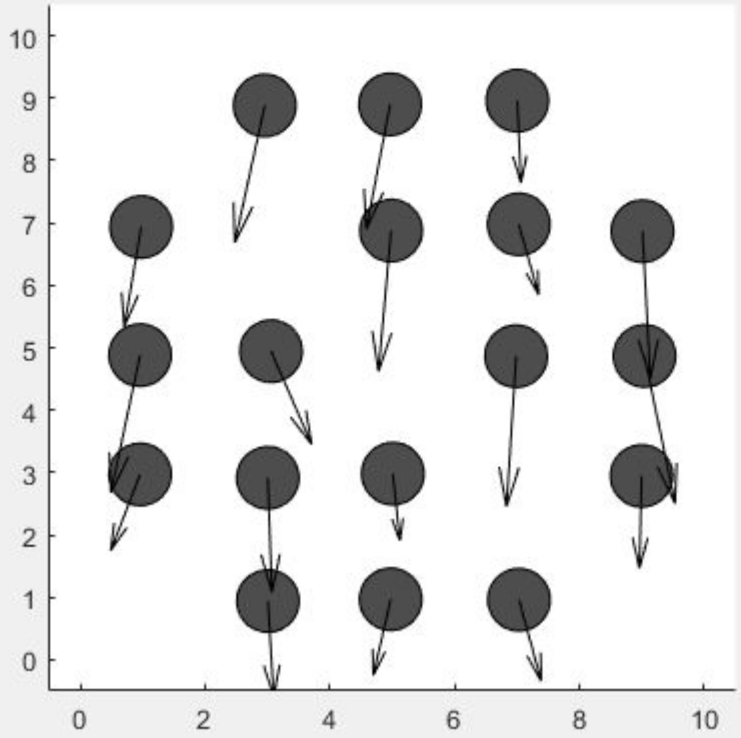
Modeling of Polygonal Particle - 2D

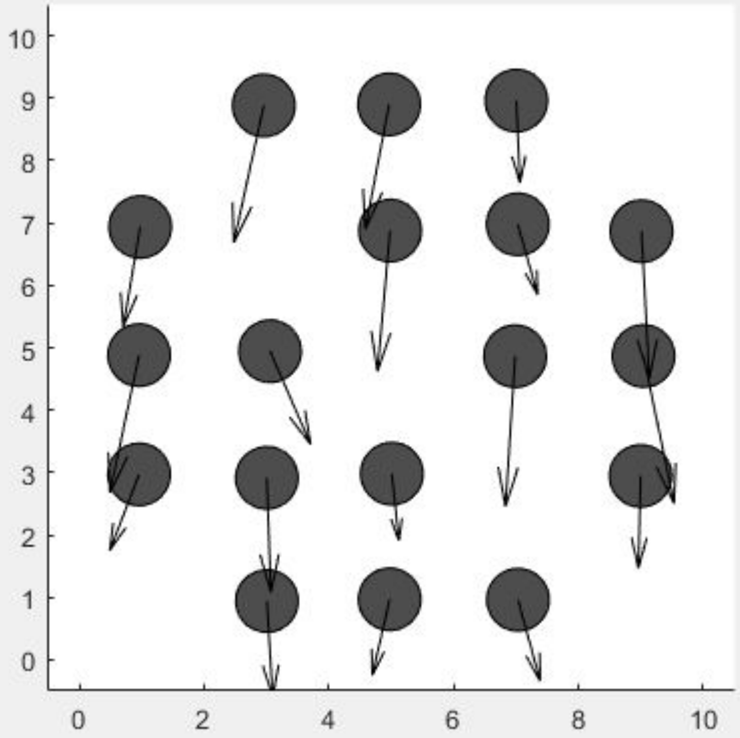
```
clear all
format compact
clf
n_part=18
% initialize radius and mass
global rad, rad(1:n_part)=0.5;
global m, m(1:n_part)=1;
global E, E=10000; % Young's modulus
global lmaxx, lmaxx=n_part/2+1;
global lminx, lminx=0;
global lmaxy, lmaxy=n_part/2+1;
global lminy, lminy=0;
global g, g=-9.81;
% initialize positions and velocities=0
rand('seed',5)
r0_x=2*mod([1:n_part],5)+1;
v0_x=rand(size(r0_x))-0.5;
r0_y=sort(r0_x);
v0_y=rand(size(r0_y))-0.5;
y0(1:4:4*n_part-3)=r0_x;
y0(2:4:4*n_part-2)=v0_x;;
y0(3:4:4*n_part-1)=r0_y;
y0(4:4:4*n_part)=v0_y;;
t_end=5;
[t,y]=ode113('DEMround2Dnorot',[0:0.005:t_end],y0);
figure(1)
[X,Y,Z]=cylinder(rad,55); % outline for plotting
```

```
function [dydt]=DEMround2Dnorot(t,y);
global m rad E lmax lmin lmaxx lminx lmaxy lminy g
n_part=length(m);
if length(y)~=4*length(m)
error('length of y must be four times the length of m')
end
if length(rad)~=length(m)
error('length of r must be four times the length of m')
end
a=zeros(2,n_part);
for i_part=1:n_part
r1=[y(4*i_part-3)
y(4*i_part-1)]; % position of first particle
rad1=rad(i_part);
% Particle-Particle Interaction
for j_part=i_part+1:n_part
r2=[y(4*j_part-3)
y(4*j_part-1)]; % position of second particle
rad2=rad(j_part);
if (norm(r1-r2)<(rad(i_part)+rad(j_part)))
forcemagnitude=E*abs(norm(r1-r2)-(rad1+rad2));
forcedirection=(r1-r2)/norm(r1-r2);
f=forcemagnitude*forcedirection;
a(:,i_part)=a(:,i_part)+f;
a(:,j_part)=a(:,j_part)-f;
end
end
% Particle-wall Interaction
if (r1(1)-rad1)<lminx
a(1,i_part)=a(1,i_part)-E*((r1(1)-rad1)-lminx);
```

Numerical Example

Modeling of Polygonal Particle - 2D



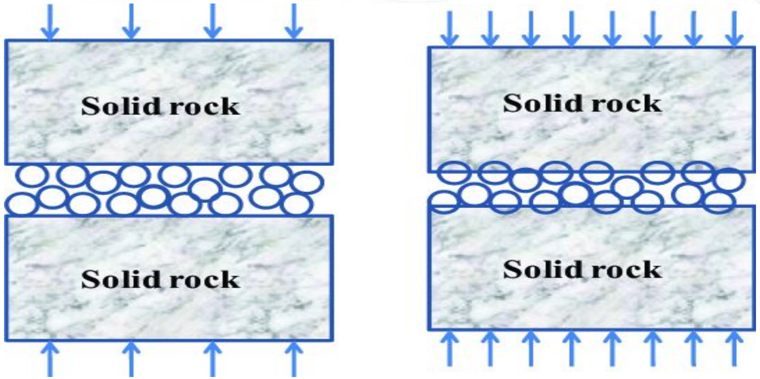


Application:Proppant Deformation

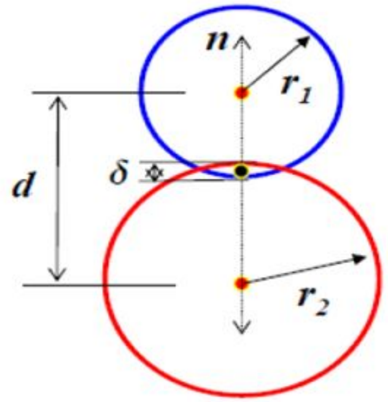
- Hydraulic Fracture
- Proppant Deformation

solid material (sand, ceramic), keep fracture open

- LIGGGHTS (open source)
- assumptions
    - shape is spherical
    - each grain is individual
    - can affect each other only touching points

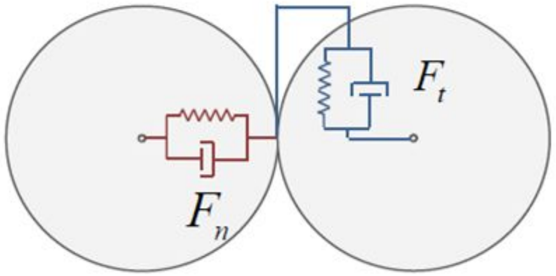


Application



Overlap between two particles

$$\delta = r_1 + r_2 - d$$



Linear spring dashpot model assumed for DEM

$$\overrightarrow{F_n} = k_n\vec{\delta} + c_n\overrightarrow{\Delta v_n}$$

$$\overrightarrow{F_t} = k_t \left| \int_{t_{c,0}}^{t} \Delta v_t(\tau) d\tau \right| \vec{t} + c_t\overrightarrow{\Delta v_t}$$

- The magnitude of tangential force is bounded by the friction force

$$\max(|\overrightarrow{F_t}|) = |\mu\overrightarrow{F_n}|$$

Where:

Δv_n = normal relative velocity at the contact point

Δv_t = relative tangential velocity

τ = tangential vector at the contact point

k_n = elastic contact for normal contact

k_t = elastic contact for tangential contact

$t_{c,}0$ = time at which the contact begins

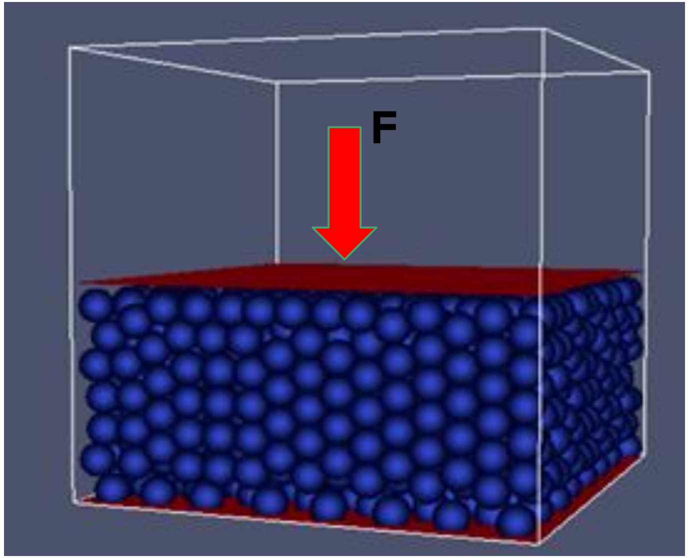
Application

Simulation

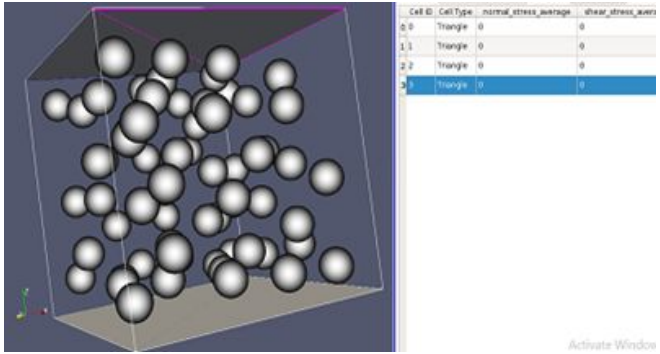
- 3x3x3 mm cubic shape
- proppant types and their features

Proppant Type	Poisson's Ratio	Young's Modulus (Gpa)
Silica Sand	0.17	70
Resin Coated	0.19	220
Ceramic	0.22	375

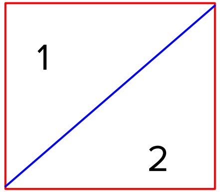
- three diameter (0.005 mm, 0.01 mm, 0.015 mm)
- grains are elastic
- no temperature affect

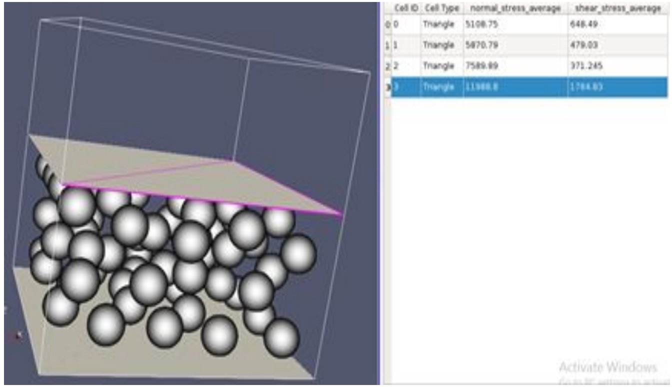


Application (Simulation)

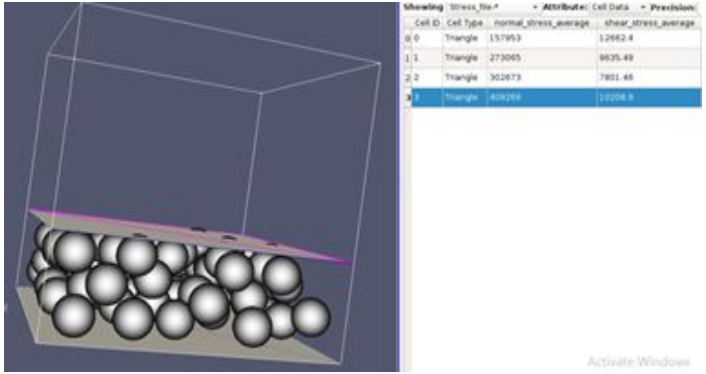


Step-1



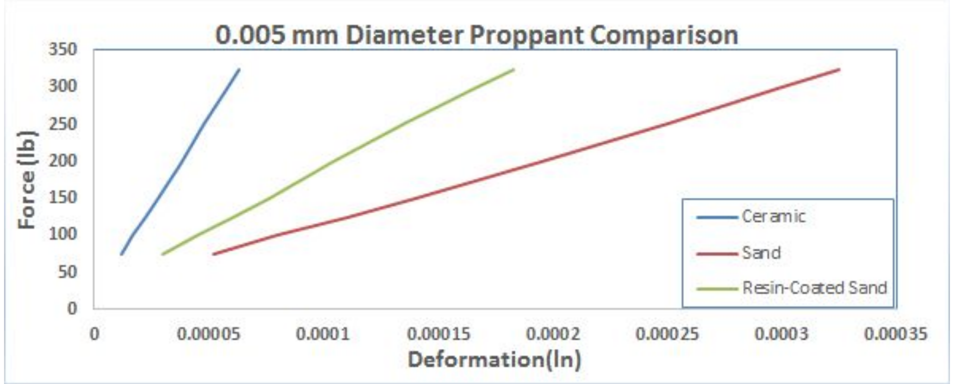


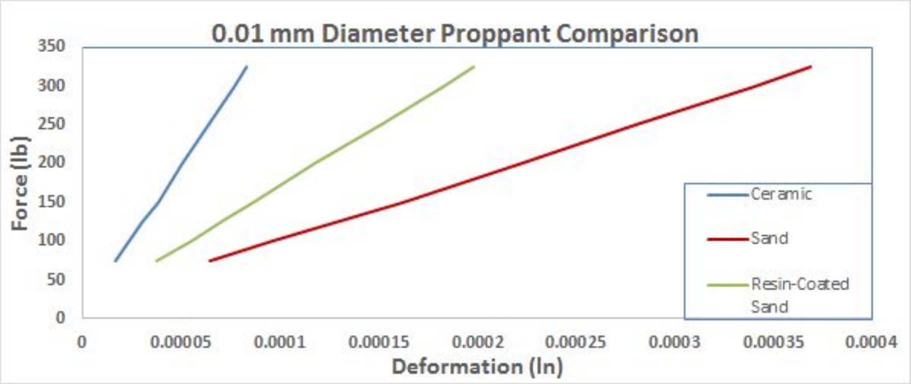
Step -2

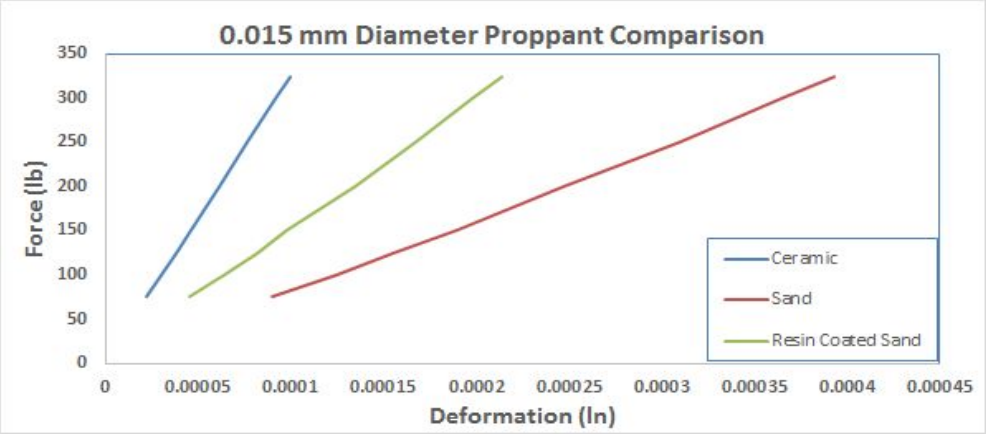


Step-3

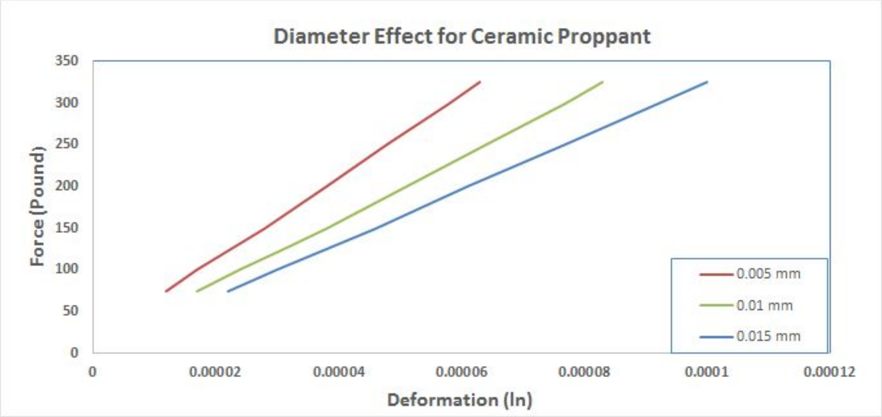
Application (Results)

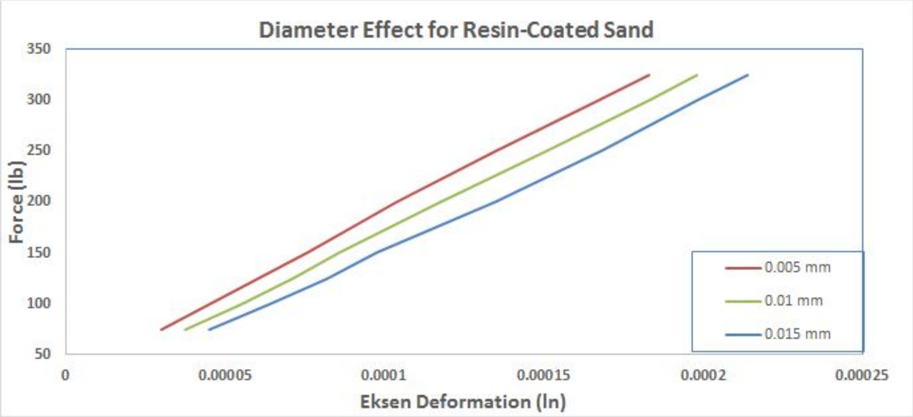


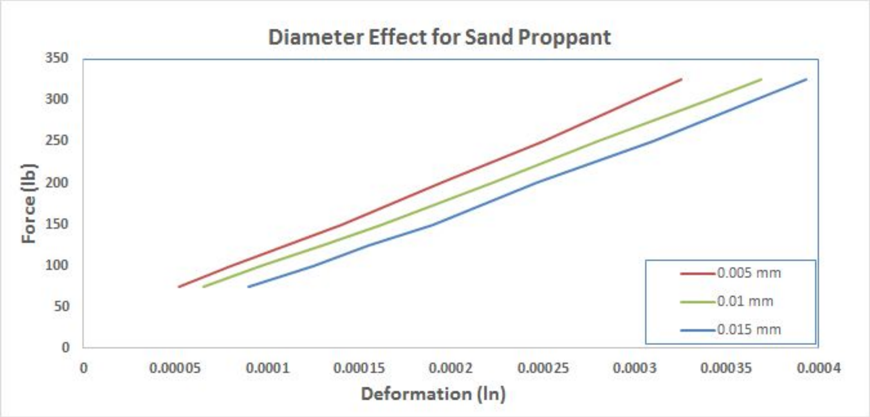




Application (Results)







Numerical Model (Code)

```
#Ceramic
atom_style    granular
atom_modify map array
boundary                m m m
newton                  off
#echo                   both
communicatesingle vel yes
units                   s
region                  reg block -0.005 0.005 -0.0025 0.0025 0.0000 0.008 units box
create_box              1 reg
neighbor                0.002 bin
neigh_modify   delay 0
#Material properties required for new pair styles
fix                     m1 all property/global youngsModulus peratomtype 375.e6
fix                     m2 all property/global poissonsRatio peratomtype 0.22
fix                     m3 all property/global coefficientRestitution peratomtypepair 0.3 0.3
fix                     m4 all property/global coefficientFriction peratomtypepair 0.5 0.5
f#New pair style
pair_style gran model hertz tangential history #Hertzian without cohesion
pair_coeff
timestep                0.00001
 fix xwalls1 all wall/gran model hertz tangential history primitive type 1 xplane -0.0115
fix xwalls2 all wall/gran model hertz tangential history primitive type 1 xplane +0.015
fix ywalls1 all wall/gran model hertz tangential history primitive type 1 yplane -0.015
fix ywalls2 all wall/gran model hertz tangential history primitive type 1 yplane +0.015
fix zwalls1 all wall/gran model hertz tangential history primitive type 1 zplane  0.00
fix zwalls2 all wall/gran model hertz tangential history primitive type 1 zplane  0.03
#servo wall
fix                     cad all mesh/surface/stress file meshes/lwall.stl type 1 stress on
fix                     cad2 all mesh/surface/stress/servo file meshes/uwall.stl type 1 com 0. 0. 0. ctrlPV
force axis 0. 0. 0.01 target_val -0.5 vel_max 0.001 kp 2069.
fix                     geometry all wall/gran model hertz tangential history mesh n_meshes 2 meshes cad
cad2
#distributions for insertion
```

Young's modulus

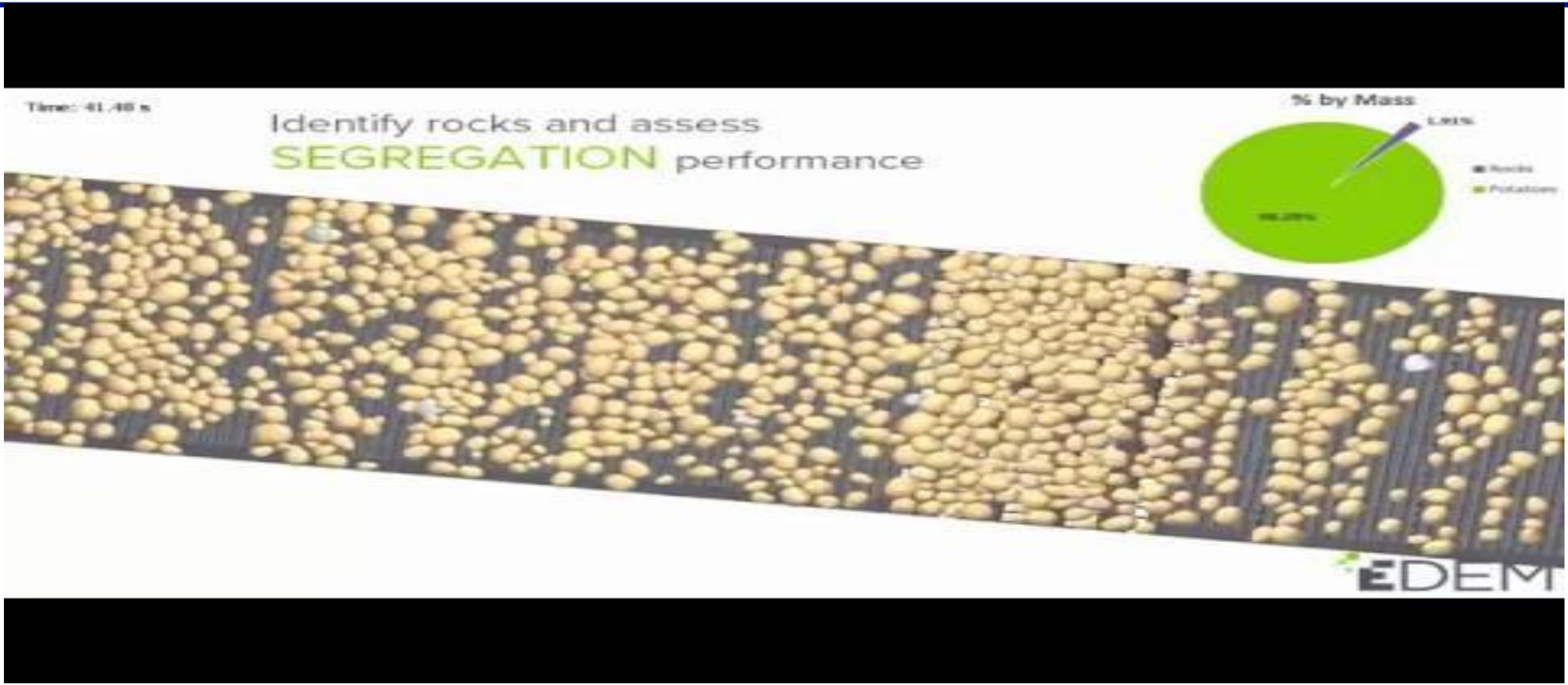
Poisson's ratio

Friction coefficient

Dimension of figure

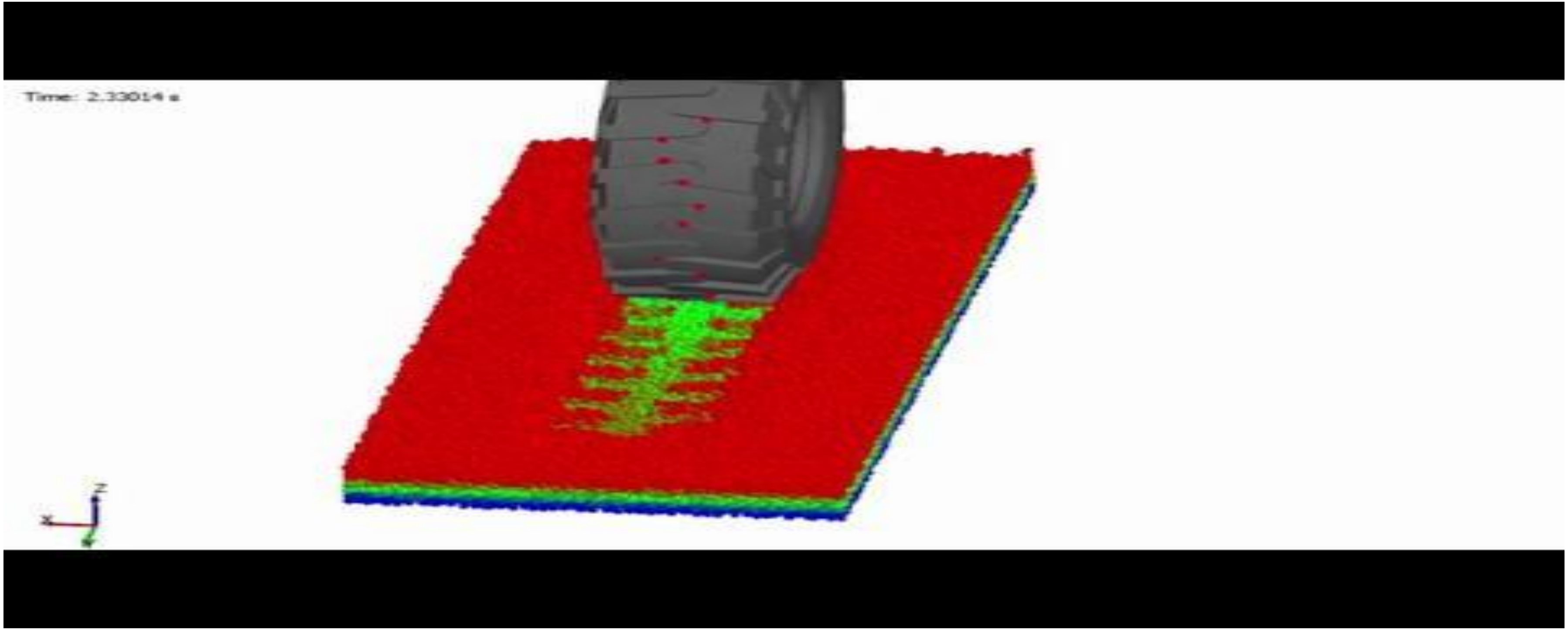
```
fix                     pts1 all particletemplate/sphere 15485863 atom_type 1 density constant 2500 radius
constant 0.015
fix                     pts2 all particletemplate/sphere 15485867 atom_type 1 density constant 2500 radius
constant 0.015
fix                     pdd1 all particledistribution/discrete 32452843 2 pts1 0.3 pts2 0.7

#parameters for gradually growing particle diameter
variable                alphastart equal 0.25
variable                alphatarget equal 0.67
variable                growts equal 50000
variable                growevery equal 40
variable                relaxts equal 20000

#region and insertion
group                   nve_group region reg

#particle insertion
fix                     ins nve_group insert/pack seed 32452867 distributiontemplate pdd1 &
                        maxattempt 200 insert_every once overlapcheck yes all_in yes vel constant 0. 0. 0. &
                        region reg volumefraction_region 0.9

#apply nve integration to all particles that are inserted as single particles
fix                     integr nve_group nve/sphere

#output settings, include total thermal energy
#compute                1 all erotate/sphere
#thermo_style           custom step atoms ke c_1 vol
#thermo                 1000
#thermo_modify          lost ignore norm no

#insert the first particles
run                     1
dump                    dmp all custom/vtk 350 post/packing_*.vtk id type type x y z ix iy iz vx vy vz fx fy fz
omegax omegay omegaz radius
dump                    D_stl all stl 400 post/Wall-*.stl
dump                    l_stl all mesh/gran/VTK 400 post/Stress_file-*.vtk stress
unfix                   ins
run                     340000
```

Applications: Potato Harvest

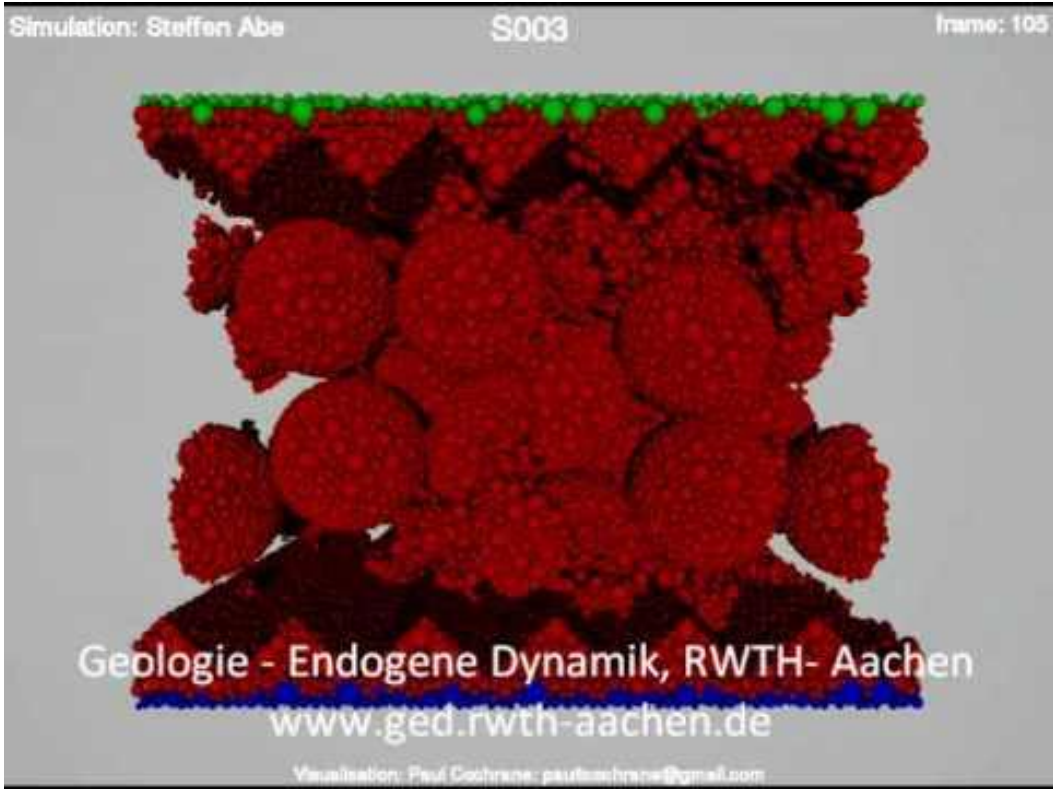


PENNSTATE

Applications: Tractor Tire-Soil Interaction

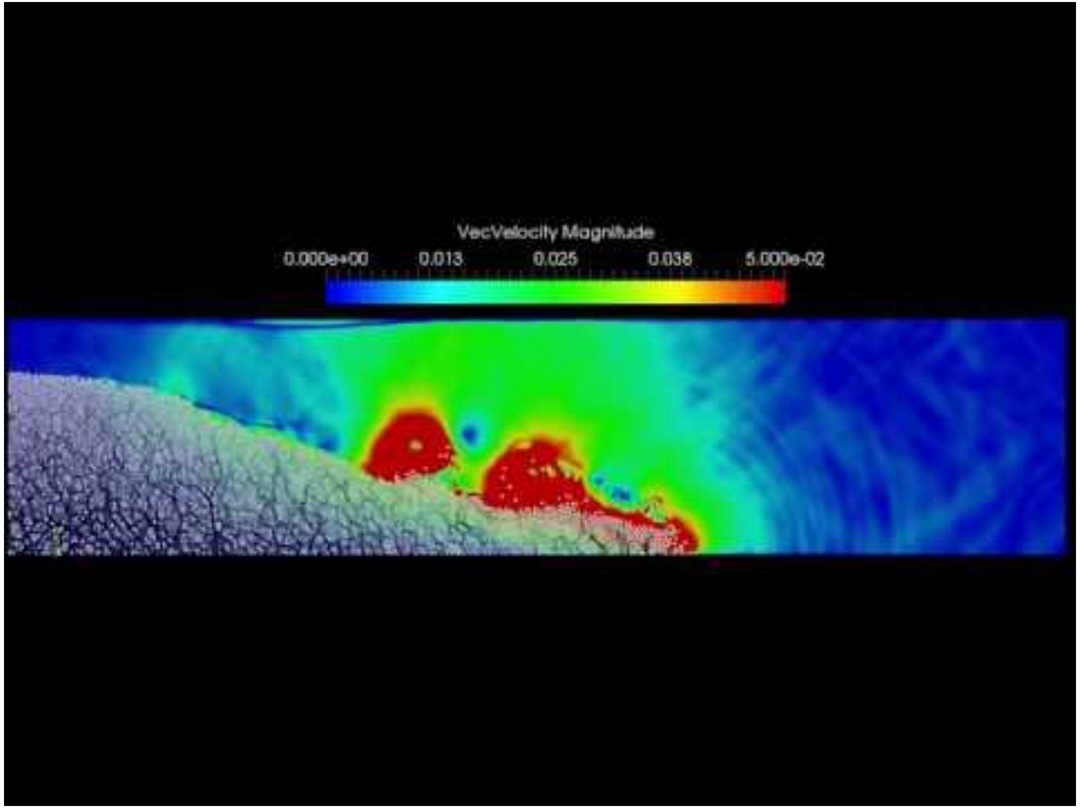


Applications: Fault Gouge



Abe & Mair 2009 : Effects of gouge fragment shape on fault friction: New 3D modelling results. Geophysical Research Letters. 36, L23302

Applications: Granular Avalanche



Krishna Kumar, University of Cambridge

PENNSTATE

References

1. EDEM Webinar: https://www.edemsimulation.com/event/webinar-what-is-dem/
2. http://jenike.com/discrete-element-modeling/
3. Cundall, P. A.; Strack, O. D. L. (March 1979). "A discrete numerical model for granular assemblies". Géotechnique. 29 (1): 47–65. doi:10.1680/geot.1979.29.1.47.
4. Shi, Gen-Hua (February 1992). "Discontinuous Deformation Analysis: A New Numerical Model For The Statics And Dynamics of Deformable Block Structures". Engineering Computations. 9 (2): 157–168. doi:10.1108/eb023855.
5. Williams, G.N.; Pande, G.; Beer, J.R. (1990). Numerical Methods in Rock Mechanics. Chichester: Wiley. ISBN 978-0471920212.
6. Granular avalanche: https://www.youtube.com/watch?v=Rlb50Ed6H6Y
7. https://www.edemsimulation.com/resources-learning/video-gallery/
8. http://www.sharetechnote.com/html/DE_Modeling_Example_SpringMass.html#SingleSpring_SimpleHarmonic_Vert_Damp
9. http://hyperphysics.phy-astr.gsu.edu/hbase/oscda.html
10. https://ebookcentral.proquest.com/lib/pensu/reader.action?docID=1745057
11. http://farside.ph.utexas.edu/teaching/315/Waves/node10.html