Table of Contents

Example 9.1

```
%From Giordano and Nakanishi
%Initializing a molecular dynamics simulation

function[x,y,vx,vy] = TheInitiator(N,L,dt,dr,v0)
```

Initialize parameters

```
%needs the following inputs:

% N :Number of particles
% L :Size of system
% dt: time step
% dr: max position component deviation dr from the vertices of a regular array
% v0: max initial velocity component magnitude
%
```

Set initial positions and velocities of the particles

```
gSpace = L/(ceil(sqrt(N))+1); %grid spacing between adjacent initial positions alo
gSpacex = gSpace:gSpace:L-gSpace;
gSpacey = gSpace:gSpace:L-gSpace;
```

```
Error using TheInitiator (line 18)
Not enough input arguments.
```

Create a grid of equally spaced points and assign the N particles to them

```
gy=[]; %initialize gy (y coords of particles initially)
for i=1:ceil(sqrt(N))
    gy=horzcat(gy,gSpacey); %concatenate on one instance of gSpacey
end
gy = gy(1:N); %make sure gy is as long as gx will be
for i=1:N %iterate  through all N particles
```

```
    %if i<ceil(sqrt(N))
        gx(i) = gSpacex(ceil(i/(ceil(sqrt(N)))));
   % end
end
```

Displace each particle randomly from its starting position

```
for i=1:N
    x0(i) = gx(i) + 2*(rand-0.5)*dr; %update x according to Ex 9.1
    y0(i) = gy(i) + 2*(rand-0.5)*dr; %update y according to Ex 9.1
    vx0(i) = 2*(rand-0.5)*v0; %calculate a random vx0 for particle i
    vy0(i) = 2*(rand-0.5)*v0; %calculate a random vy0 for particle i
    xPrev(i) = x0(i) - vx0(i)*dt; %backwards-calculate xPrev for future use of the
    yPrev(i) = y0(i) - vy0(i)*dt; %do the same for yPrev
end
x = [xPrev;x0];
y = [yPrev;y0];
vx =[vx0;vx0];
vy = [vy0;vy0];

end
```

Plot the results

hold on plot(gx,gy,'b.') plot(x0,y0,'r.') plot(xPrev,yPrev,'g.') axis([0 L 0 L]) ylabel('Y') xlabel('X')

Published with MATLAB® R2014a