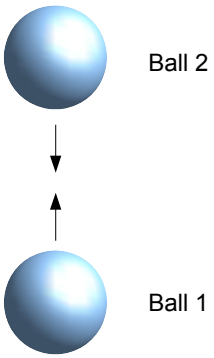

Figure 1: Two identical balls collide

Dear LAMMPS admins,

I'm trying to illustrate here a bug within the granular package of LAMMPS.

I performed collisions between two identical balls, as indicated in figure 1.

In all the simulation, there is no gravity. Equal but opposite velocities were given to the balls as initial condition.

The normal force on ball 2 is calculated in LAMMPS and compared to independent calculations by Matlab. For the first two simulations, γ_n is set to zero, that is, visco-elastic damping is turned off for the purpose of pinning down the problem.

The force calculation by Matlab is based on equation 1, which can be found on page 426, LAMMPS user manual:

$$F = \sqrt{\frac{d-r}{d}}(K_n(d-r) - \gamma_n m_{eff} V_n) \tag{1}$$

$d = R_1 + R_2$ is the contact distance between two balls of radius $R_1$and R_2

r is the distance between the centers of contacting balls

K_n is the elastic constant for normal contact

γ_n is the viscoelastic constants for normal contact

$m_{eff} = m_i m_j/(m_i + m_j)$ is the effective mass of two balls of mass m_i and m_j

V_n is the normal component of the relative velocity of the two balls

To my surprise, the force calculated this way (the red curve) does NOT match the force calculated by LAMMPS (the blue circles). However, the forces calculated by LAMMPS can be matched by calculation based on equation 2(the green curve).

$$F = \sqrt{d-r}(Kn(d-r) - \gamma_n m_{eff} V_n) \tag{2}$$

It is believed that equation 1 is correct. It is listed in LAMMPS user manual, and it is also seen in literature, for example, the experimental paper by Laurent Labous[1], etc.

This finding is further confirmed by looking into the source code (pair_gran_hertzian.cpp, line 145-154):

```
// normal damping term
// this definition of DAMP includes the extra 1/r term
```

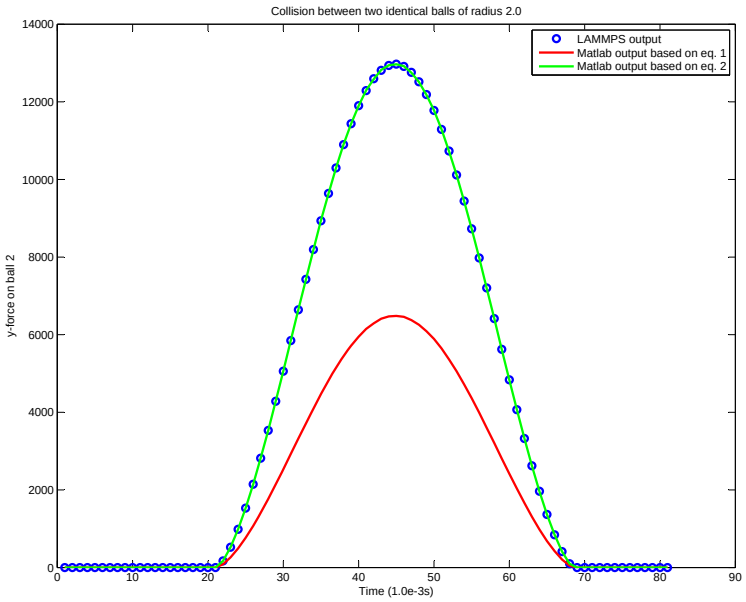

Figure 2: y-force on ball 2 with radius 2

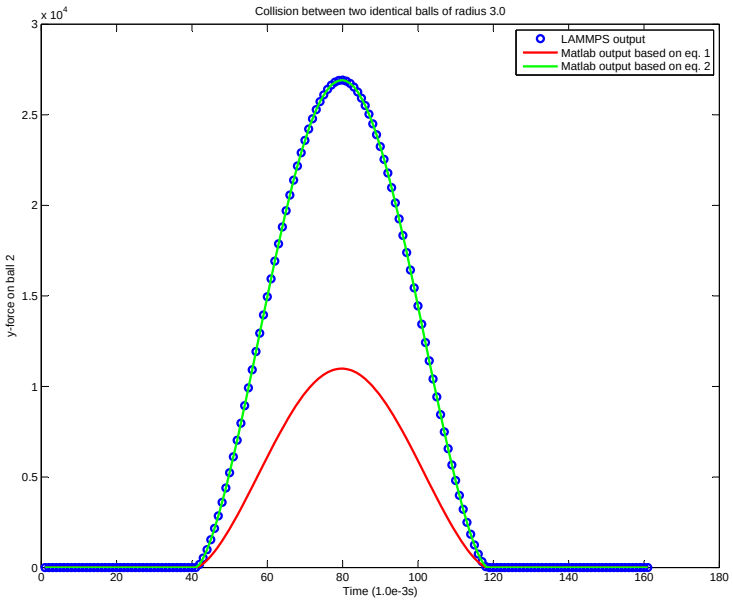

Figure 3: y-force on ball 2 with radius 3

```
xmeff = rmass[i]*rmass[j] / (rmass[i]+rmass[j]);
if (mask[i] & freeze_group_bit) xmeff = rmass[j];
if (mask[j] & freeze_group_bit) xmeff = rmass[i];
damp = xmeff*gamman_dl*vnnr/rsq;
ccel = xkk*(radsum-r)/r - damp;
rhertz = sqrt(radsum - r);
ccel = rhertz * ccel;
```

To me, it is a bit surprised to find out that LAMMPS implemented normal contact force (for granular particles) by a single parameter K_n instead of Young's modulus E and Poisson's ratio ν. As this will impose an unnecessary restriction of mono-dispersity to the systems that LAMMPS can simulate, because K_n is actually a function of d, ($K_n = Ed$). Even for mono-dispersity, the forces so calculated are off by a factor of $\sqrt{d}$. So, please look into this matter and fix the bug.

Along this line of investigation, I also find the following paragraph misleading (page 198, LAMMPS user manual):

"IMPORTANT NOTE: Some models in LAMMPS treat particles as extended spheres or ellipsoids, as opposed to point particles. In 2d, the particles will still be spheres or ellipsoids, not circular disks or ellipses, meaning their moment of inertia will be the same as in 3d."

What I found is that the mass of the balls is calculated differently for 2d and 3d simulations.

$M = \pi r^2 \rho$ for 2d

and

$M = \frac{4}{3}\pi r^3 \rho$for 3d.

That is, for 2d simulations, mass of the balls IS calculated as circular disks with unit thickness.

1 The input scripts are as following:

1.1 The collision.src file (used as ./lmp_serial < collision.src)

```
# collision test
dimension 3
atom_style granular
boundary p f p
newton off
read_data collision.in
# gamma_n set to zero, no visco-elastic damping
```

pair_style gran/hertzian 200000.0 0.0 0.5 0

timestep 0.000001

fix 1 all nve/sphere

fix ywalls all wall/gran yplane 0 100 50 0

dump 1 all custom 100 collision_gamma.out tag x y vx vy radius fx fy

run 1000000

1.2 The collision.in file

first line will always be ignored.

collision test of granular package of LAMMPs

header section

2 atoms

1 atom types

0 40 xlo xhi

0 40 ylo yhi

-10 10 zlo zhi

body section

Atoms

1 1 6.0 1.0 14.0 9.0 0.0

2 1 6.0 1.0 14.0 21.0 0.0

Velocities

1 0.0 5.0 0.0 0.0 0.0 0.0

2 0.0 -5.0 0.0 0.0 0.0 0.0

1.3 Matlab script

%collision between two identical balls

%parameters as in LAMMPS

kn = 200000.0;

dens=1.0;

gamma_n=0.0;

d=6.0; % twice the radius, R1+R2

r=ball2_y-ball1_y; % balls are lined up in x-direction

vn=ball2_vy-ball1_vy; % relative velocity

meff=0.5*(4.0/3.0*pi*(d/2.0)^3)*dens; %effective mass

```
fy=sqrt((d-r)/d).*(kn*(d-r)-meff*gamma_n*vn); % equation for calculate normal force as in page 426, LAMMPS manual
fy_m = sqrt(d)*fy;
plot(ball2_fy(5600:10:7200),'bo')
hold on
plot(real(fy(5600:10:7200)),'r')
plot(real(fy_m(5600:10:7200)),'g')
xlabel('Time (1.0e-3s)')
ylabel('y-force on ball 2')
```

References

[1] Anthony D. Rosato Laurent Labous and Rajesh N. Dave. Measurements of collisional properties of spheres using high-speed video analysis. *PHYSICAL REVIEW E*, 56:5717–5725, 1997.