# Implementation of elastic-plastic model in PDLAMMPS

R. Rahman and J.T. Foster
Center for Simulation, Visualization and
Real-time prediction (SiViRt)
The University of Texas at San Antonio
San Antonio, TX 78249
Email: rezwanur.rahman@utsa.edu,
john.foster@utsa.edu

October 7, 2013

## 1 Introduction

In this documentation the discussion is focused on implementation of elastic plastic peridynamics model in PDLAMMPS [PLPS08],[Pli95]. The peridynamic elastic-plastic formulation used in this work was developed by John Mitchell at Sandia National Lab [Mit11]. In the PDLAMMPS a new pair-peri-style is added in order to incorporate the elastic-plastic formulation. The new source codes `pair-peri-eps.cpp` and `pair-peri-eps.h` are introduced. Besides, `fix-peri-neigh.cpp` and `fix-peri-neigh.h` are modified for introducing elastic-plastic solid model (EPS).

## 2 Algorithm and implementation

The total extension state can be decomposed into two parts [Mit11] [1]:

[1] For detail please see the document on implementation of viscoelasticity in state based peridynamics model [Mit11]

$$\begin{aligned}
\text{Total extension: } e\left(Y\right) &= e^{i}\left(Y\right)+e^{d}\left(Y\right) && (1)\\
\text{Volumetric extension: } e^{i}\left(Y\right) &= \frac{\theta\left(Y\right)\left|X\right|}{3} && (2)\\
\text{Deviatoric extension: } e^{d}\left(Y\right) &= \left|Y\right|-\left|X\right|-\frac{\theta\left(Y\right)\left|X\right|}{3} && (3)
\end{aligned}$$

Here, $|Y|$, $|X|$ and $\theta$ are the reference state, defromation state and dilation state, respectively. The deviatoric extension ca be written as:

$$e^{d}\left(Y\right)=e^{de}\left(Y\right)+e^{dp(i)}\left(Y\right) \tag{4}$$

Here, $e^{de}\left(Y\right)$ and $e^{dp(i)}$ are the elastic and plastic parts of the deviatoric extension. The elastic-plastic force scaler state can be written as:

$$t=t^{i}+t^{d}=-\frac{3p}{m}\omega\underline{x}+\alpha\omega\left(e^{d}-e^{dp}\right) \tag{5}$$

$p$, $k$, $\alpha$, $t^i$ and $t^d$ are hydrostatic pressure, bulk modulus, elastic properties volumetric and deviatoric scaler force states, respectively. $\alpha=\frac{15\mu}{m}$; where, $\mu$, $m$ are the shear modulus and weighted volume, respectively. The influence function: $\omega\left\langle\xi\right\rangle=\frac{1}{\|\xi\|}$, $\|\xi\|$ is the scaler reference state. At the current or $(n+1)^{th}$ timestep the scaler trial force state $t^{d}_{trial}$ and it's norm $\left\|t^{d}_{trial}\right\|$ are calculated.

$$t^{d}_{trial}=\frac{15\mu}{m}\omega\left(e^{d}_{n+1}-e^{dp}_{n+1}\right). \tag{6}$$

Equation. 6 is implemented in the function in `pair_peri_eps.cpp`:

```
double PairPeriEPS::compute_DeviatoricForceStateNorm(int i)
{
  int j,jj,jnum,itype,jtype;
  double xtmp,ytmp,ztmp,delx,dely,delz;
  double xtmp0,ytmp0,ztmp0,delx0,dely0,delz0;
  double rsq,r,dr;
  double delta;
  double tdtrial;
  double norm = 0.0;
  double **x = atom->x;
```

```
int *type = atom->type;
double **x0 = atom->x0;
double *s0 = atom->s0;
int nlocal = atom->nlocal;
double *vfrac = atom->vfrac;
double vfrac_scale = 1.0;
double lc = domain->lattice->xlattice;
double half_lc = 0.5*lc;
double **r0   =
    ((FixPeriNeigh *) modify->fix[ifix_peri])->r0;
int **partner =
    ((FixPeriNeigh *) modify->fix[ifix_peri])->partner;
int *npartner =
    ((FixPeriNeigh *) modify->fix[ifix_peri])->npartner;
double *wvolume =
    ((FixPeriNeigh *) modify->fix[ifix_peri])->wvolume;
double **deviatorPlasticextension =
  ((FixPeriNeigh *) modify->fix[ifix_peri])->deviatorPlasticextension;
int periodic =
   domain->xperiodic || domain->yperiodic || domain->zperiodic;

// compute the dilatation theta

  xtmp = x[i][0];
  ytmp = x[i][1];
  ztmp = x[i][2];
  xtmp0 = x0[i][0];
  ytmp0 = x0[i][1];
  ztmp0 = x0[i][2];
  jnum = npartner[i];
  itype = type[i];

  for (jj = 0; jj < jnum; jj++) {
    if (partner[i][jj] == 0) continue;
    j = atom->map(partner[i][jj]);
     // check if lost a partner without first breaking bond
    if (j < 0) {
      partner[i][jj] = 0;
```

```
  continue;
}
delx = xtmp - x[j][0];
dely = ytmp - x[j][1];
delz = ztmp - x[j][2];
if (periodic) domain->minimum_image(delx,dely,delz);
rsq = delx*delx + dely*dely + delz*delz;
delx0 = xtmp0 - x0[j][0];
dely0 = ytmp0 - x0[j][1];
delz0 = ztmp0 - x0[j][2];
if (periodic) domain->minimum_image(delx0,dely0,delz0);
r = sqrt(rsq);
dr = r - r0[i][jj];
if (fabs(dr) < 2.2204e-016) dr = 0.0;

      // scale vfrac[j] if particle j near the horizon
double vfrac_scale;

jtype = type[j];
double delta = cut[itype][jtype];

// scale vfrac[j] if particle j near the horizon

if ((fabs(r0[i][jj] - delta)) <= half_lc)
  vfrac_scale = (-1.0/(2*half_lc))*(r0[i][jj]) +
    (1.0 + ((delta - half_lc)/(2*half_lc) ) );
else vfrac_scale = 1.0;

double ed = dr - (theta[i] * r0[i][jj])/3;
double edPNP1 = deviatorPlasticextension[i][jj];

jtype = type[j];
delta = cut[itype][jtype];

double omega_plus  =
    influence_function(-1.0*delx0,-1.0*dely0,-1.0*delz0);
double omega_minus =
    influence_function(delx0,dely0,delz0);
```

```
        double stretch = dr / r0[i][jj];

        tdtrial = ( 15 * shearmodulus[itype][itype]) *
         ((omega_plus * theta[i] / wvolume[i]) +
         ( omega_minus * theta[j] / wvolume[j] ) ) * (ed - edPNP1);

        norm += tdtrial * tdtrial * vfrac[j] * vfrac_scale;
      }
    return sqrt(norm);
  }
```

The yield function $f\left(t^d_{trial}\right)$ is written based on $\left\|t^d_{trial}\right\|$ and yield stress $\sigma_Y$ (Eq. 6).

$$f\left(t^d_{trial}\right) = \frac{\left\|t^d_{trial}\right\|^2}{2} - \frac{25\sigma_Y^2}{8\pi\delta^5} \tag{7}$$

Here, $\delta$ is the horizon. If $f\left(t^d_{trial}\right) < 0$ the step is elastic. Otherwise, it is plastic. If the step is elastic:

$$t^d_{n+1} = t^d_{trial}, \tag{8}$$

$$e^{dp}_{n+1} = e^{dp}_n. \tag{9}$$

For plastic step:

$$\Delta\lambda = \frac{1}{\alpha}\left[\frac{\left\|t^d_{trial}\right\|}{\sqrt{2\psi_0}} - 1\right], \tag{10}$$

$$t^d_{n+1} = \sqrt{2\psi_0}\frac{t^d_{trial}}{\left\|t^d_{trial}\right\|}, \tag{11}$$

$$e^{dp}_{n+1} = e^{dp}_n + \Delta\lambda t^d_{n+1}. \tag{12}$$

The return algorithm is implemented in `PairPeriEPS::compute(int,int)`.

# 3 LAMMPS command for PD EPS

There is no significant change in the LAMMPS input script for elastic-plastic model. For PD EPS the LAMMPS commands are:

```
pair_style peri/eps
pair_coeff i j Bulk_modulus Shear_modulus s00 alpha Yield_Stress
```

# 4 Conclusion

The LAMMPS implementation of peridynamic elastic-plastic model is still in beta phase. Any bug or issue can be informed to the authors through `rezwanur.rahman@utsa.edu` .

# References


[Mit11] John A. Mitchell. A nonlocal, ordinary, state-based plasticity model for peridynamics. *Sandia National Lab Report*, 3166:1–34, 2011.

[Pli95] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J Comp Phys*, 117:1–19, 1995.

[PLPS08] Michael L. Parks, Richard B. Lehoucq, Steven J. Plimpton, and Stewart A. Silling. Implementing peridynamics in molecular dynamics code. *Computer physics communication*, 179:777–783, 2008.


# 5 Appendix

In the current updated version of PDLAMMPS there are two new compute commands are included. One is **compute ID group-ID plasticity/atom**, and **compute ID group-ID dilatation/atom**. Compute style plasticity/atom is applicable for pair style peri/eps and compute style dilatation/atom is applicable for pair styles peri/lps, peri/ves and peri/eps. In compute style plasticity, $\lambda$ for each peridynamic node is calculated. The dilatation at each peridynamics node $\theta$ is computed by compute dilatation/atom command. Similar to compute style damage/atom, these compute commands store per atom information.