1 system and simulation setup

Two atoms, one of 1e, the other -2e. They are fix-positioned in the middle of unit cell. The length of x direction of the box is increased in a fashion similar to Successione di Fibonacci, and cut-off radius increased by 10 until 5000, atom distance until 2500.

Two jobs are generated, one with pair style lj/cut/coul/long, the other with pair style lj/cut/coul/cut under no PBC, implying cut-off radius larger than atom distance.

2 simulation data

See attached excel file

3 data analysis and conclusion

1) when unit cell size is big, ecoul+elong -> ecoul(no PBC). For example, see 5000/2200/10-200

2) when unit cell size is big, distance=10, cut-off radius has significant effect on energy calculation

3) in LAMMPS code, two pair styles have different treatment for coulombic energy, in that a prefactor is introduced.

In pair\_lj\_cut\_coul\_cut.cpp, see L137, or as follows:

if (eflag) {

if (rsq < cut\_coulsq[itype][jtype])

ecoul = factor\_coul \* qqrd2e \* qtmp\*q[j]\*sqrt(r2inv);

else ecoul = 0.0;

if (rsq < cut\_ljsq[itype][jtype]) {

evdwl = r6inv\*(lj3[itype][jtype]\*r6inv-lj4[itype][jtype]) -

offset[itype][jtype];

evdwl \*= factor\_lj;

} else evdwl = 0.0;

}

In pair\_lj\_cut\_coul\_long.cpp, see L183; for related params, see L138-162, or as follows:

if (!ncoultablebits || rsq <= tabinnersq)

ecoul = prefactor\*erfc;

if (rsq < cut\_coulsq) {

if (!ncoultablebits || rsq <= tabinnersq) {

r = sqrt(rsq);

grij = g\_ewald \* r;

expm2 = exp(-grij\*grij);

t = 1.0 / (1.0 + EWALD\_P\*grij);

erfc = t \* (A1+t\*(A2+t\*(A3+t\*(A4+t\*A5)))) \* expm2;

prefactor = qqrd2e \* qtmp\*q[j]/r;

forcecoul = prefactor \* (erfc + EWALD\_F\*grij\*expm2);

if (factor\_coul < 1.0) forcecoul -= (1.0-factor\_coul)\*prefactor;

} else {

union\_int\_float\_t rsq\_lookup;

rsq\_lookup.f = rsq;

itable = rsq\_lookup.i & ncoulmask;

itable >>= ncoulshiftbits;

fraction = (rsq\_lookup.f - rtable[itable]) \* drtable[itable];

table = ftable[itable] + fraction\*dftable[itable];

forcecoul = qtmp\*q[j] \* table;

if (factor\_coul < 1.0) {

table = ctable[itable] + fraction\*dctable[itable];

prefactor = qtmp\*q[j] \* table;

forcecoul -= (1.0-factor\_coul)\*prefactor;

}

}

} else forcecoul = 0.0;

4) in order to have more accurate ecoul if elong is missing in calculation, we need: bigger cut-off radius, bigger unit cell, and smaller distance.