MSE760 Assignment 1

Lijing Yang

Mechanical Engineering Department, University of Wisconsin – Madison

February 14, 2018

1 Initialize a lattice

The initialization of Ar lattice is done in a function called createLattice, and we could pass the dimension of lattice from command line arguments with usage like:

```
./a.out 5 5 5 0
```

where the first 3 arguments define the size of lattice in x, y, z direction separately, and the last argument defines if there is (1) or not (0) a periodic boundary condition.

Fig. 1 shows the coordinates of the $5 * 5 * 5$ lattice, by rotating the plots in Matlab, I was able to convince myself that they look correct. Note these coordinates are in SI unit, and a print out can be found in output.txt file with the columns defined as

```
cellID (1:(xdim*ydim*zdim)) atomID(1:4) x y z
```

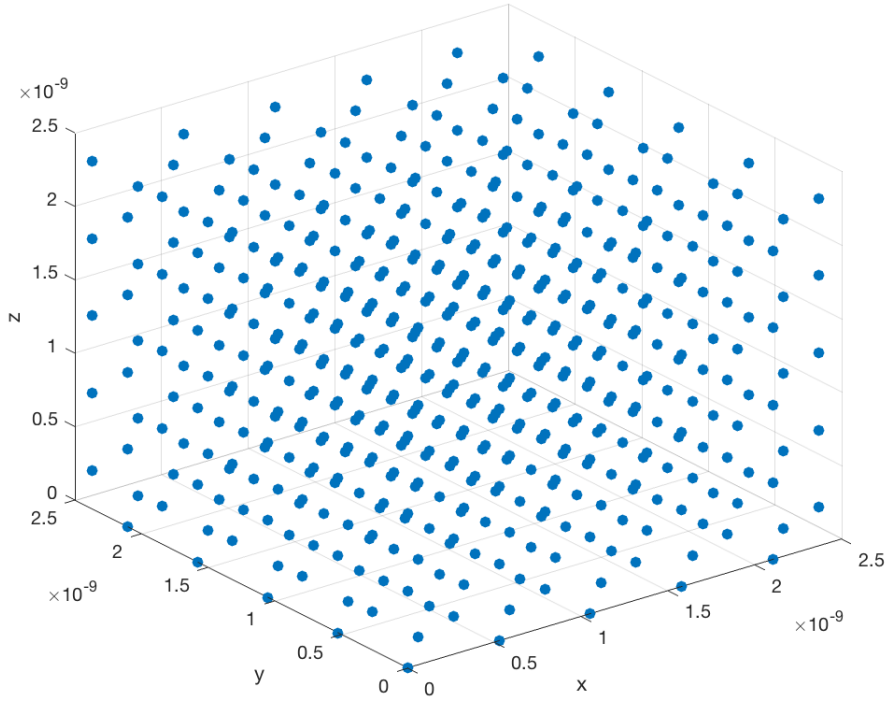

Figure 1: Displacement error of a free response oscillator using various numerical schemes

2 LJ energy of the system

2.1 LJ calculation without PBC

```
// not applying PBC
void noPBC(int xdim, int ydim, int zdim, int currentDim, double a_red,
           double* currentArray, double* currentArrayij){

    int numOfAtoms = 4 * xdim * ydim * zdim;

    for (int i = 0; i < numOfAtoms - 1; i++) {
      for (int j = i + 1; j < numOfAtoms; j++) {
            currentArrayij[j + i * numOfAtoms] = currentArray[i] - currentArray[j];
          }
    } // end of for loop
}
```

2.2 LJ calculation without PBC

```
// applying PBC
void addPBC(int xdim, int ydim, int zdim, int currentDim, double a_red,
           double* currentArray, double* currentArrayij){

    int numOfAtoms = 4 * xdim * ydim * zdim;

    for (int i = 0; i < numOfAtoms - 1; i++) {
      for (int j = i + 1; j < numOfAtoms; j++) {
           currentArrayij[j + i * numOfAtoms] = currentArray[i] - currentArray[j];
           if (currentArrayij[j + i * numOfAtoms] < -(currentDim * a_red) / 2.0)
             currentArrayij[j + i * numOfAtoms] = currentArrayij[j + i * numOfAtoms] +
                 (currentDim * a_red);

           else if (currentArrayij[j + i * numOfAtoms] > (currentDim * a_red) / 2.0)
             currentArrayij[j + i * numOfAtoms] = currentArrayij[j + i * numOfAtoms] -
                 (currentDim * a_red);
           else
             currentArrayij[j + i * numOfAtoms] = currentArrayij[j + i * numOfAtoms];
           }
    } // end of for loop
}
```

2.3 Comparison of convergence

Fig. 2 shows the comparison of convergence both for tests without PBC and with PBC. After we applied PBC, the LJ potential shows a convergence close to theoretical cohesive energy

$-0.089eV$ at size 5, while without applying PBC it requires much more cells to converge. In this test, we only include number of cells up to 20 in each dimension, which yields $32,000$ atoms as a total count.

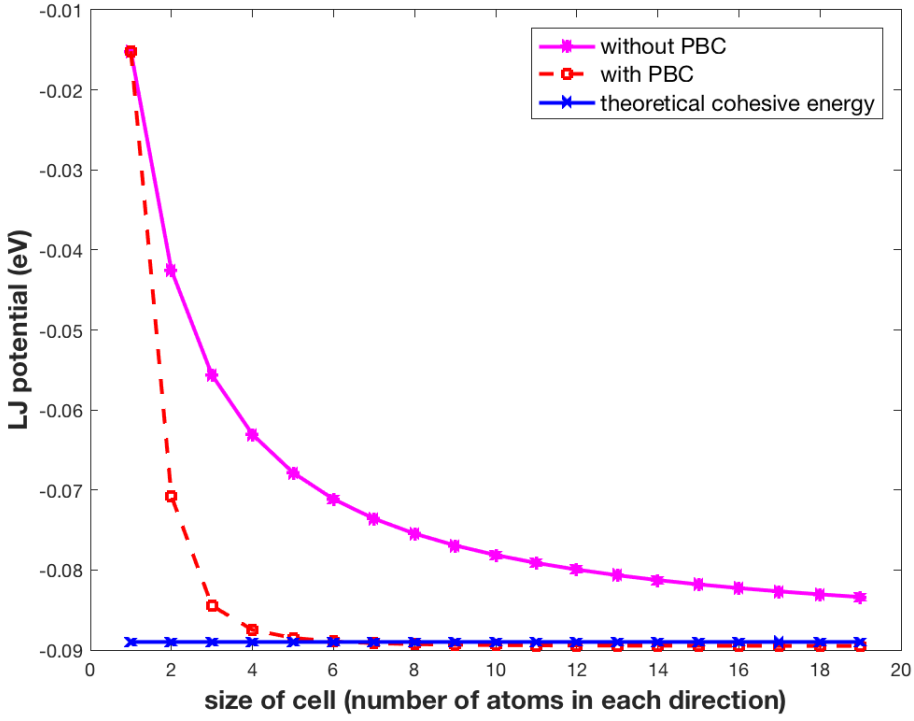

Figure 2: LJ potential comparison of convergence to theoretical cohesive energy

3 Efficiency of Code

In this code, one dimensional arrays are allocated in the memory using malloc in C, which makes the trip to memory less expensive and avoids the issue of allocating dynamic heap memory.

For faster memory access, the code is written to store structure of arrays instead of array of 3D structures, since for the calculation of r_{ij}, we always need to do operations like $x_{ij} = x_i - x_j$, which could be optimized by bringing the 1D array from main memory into cache and take advantage of the temporal and spatial locality.

```
1  // allocate memory for storage of 3D lattice points
2  double *arrayX = (double *)malloc(4 * numberOfAtoms* sizeof(double));
```

```
3  double *arrayY = (double *)malloc(4 * numberOfAtoms* sizeof(double));
4  double *arrayZ = (double *)malloc(4 * numberOfAtoms* sizeof(double));
```

4 Discussion

It could be found from the LJ energy plot that periodic boundary condition helps a lot on the convergence of LJ potential.

It was interesting during the debugging process that I've accidentally passed a_{red} as a int instead of a double, and it created super large LJ energy that took me a while to figure out what went wrong.

Another good habit would be always try to visualize the lattice points when not sure about its correctness, looking at numbers in the order of $1e^{-}10$ would not be helpful to spot the problem. Now that we started using Ovito for visualization, hopefully life would be much easier than before (while Ovito seems not to cooperate on files without time stepping info, needs more practice on it).