#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <time.h>

#include "MT.h" //乱数を生成するヘッダーファイル

#define pi 3.141592

double dis(double[][3],double[][3],int,int);

double ang(double[][3],double[][3],double[][3],int,int,int);

const double avogadro=6.0221367e+23;

const double kb=1.380648e-23;

int main()

{

int i,j,k,l,m;

int ix,iy,iz;

int loop=4001;//メインループ繰り返し数

int count=0;//メインループカウント

double t,dt=0.5e-15;//時間

double angs=1e-10;//1[Å]=1e-10[m]

double ev=1.60218e-19;//1[eV]=1.60218e-19[J]

double ev\_angs=ev/angs;//速度の更新の際に単位変換として利用

double T0=293.15,T,x;//温度[K](T0:設定温度)

double co;//角度(cosθ)

//原子パラメータ

int par1\_in\_unit=8;//単位格子内の原子数

int unit\_num\_x=2,unit\_num\_y=2,unit\_num\_z=2;//x,y,z方向の単位格子の数

int par1=par1\_in\_unit\*unit\_num\_x\*unit\_num\_y\*unit\_num\_z;//基本セル中の原子(Si)の数

int par1\_all=par1\*18;//ミラーセルも含めた計算に用いる全原子の数(Si)

int par2=10;//基本セル１つ当たりに近づける原子(Cl)の数

int par2\_all=par2\*9;//ミラーセルも含めた計算に用いる全原子の数(Cl)

int num\_j[par1\_all];//カットオフ距離内の原子の保管用(j)

int num\_k[par1\_all];//カットオフ距離内の原子の保管用(k)

double c1[par1\_all][3],c2[par2\_all][3];//座標［Å］

double v1[par1\_all][3],v2[par2\_all][3],v;//速度［m/s］

double f1[par1\_all][3],f2[par2\_all][3];//力［N］

double m1=28.0855/avogadro\*1e-3;//Siの原子質量[kg]

double m2=35.453/avogadro\*1e-3;//Clの原子質量[kg]

double a=5.4301;//Siの格子定数[Å]

double a\_x=a\*unit\_num\_x;//基本セルのx方向の1辺の長さ

double a\_y=a\*unit\_num\_y;//基本セルのy方向の1辺の長さ

double a\_z=a\*unit\_num\_z;//基本セルのz方向の1辺の長さ

//tersoffパラメータ(Si)

double lam1=3.2394;//[Åe-1]

double lam2=1.3258;//[Åe-1]

double beta=3.3675e-1;

double n=2.2956e+1;

double c=4.8381;

double d=2.0417;

double h=0.;

double R=3.0;//Å

double D=0.2;//Å

double A=3.2647e+3;//[eV]

double B=9.5373e+1;//[eV]

double f[3];

double dz[3],dz\_i[3],dz\_j[3],dz\_k[3];

double fr,fa,fc\_ij,fc\_ik,dfc\_ij,dfc\_ik,b,db,g,dg,z;

double r\_ij,r\_ik,r\_jk;

//morseパラメータ

//Cl\_Si

double E1=4.35057;

double a1=1.43163;

double ra1=2.1;

double cut1=5.25;//SiとClのカットオフ距離[Å]

//Cl\_Cl

double E2=1.757;

double a2=1.2956;

double ra2=2.5;

double cut2=5.2;//ClとClのカットオフ距離[Å]

//・・・・・初期条件・・・・・

//・・・初期位置・・・

//Si

i=0;

for(iz=0;iz<unit\_num\_z;iz++){

for(iy=0;iy<unit\_num\_y;iy++){

for(ix=0;ix<unit\_num\_x;ix++){

c1[i][0]=0.+a\*ix;

c1[i][1]=0.+a\*iy;

c1[i][2]=0.+a\*iz;

i++;

c1[i][0]=0.+a\*ix;

c1[i][1]=1./2.\*a+a\*iy;

c1[i][2]=1./2.\*a+a\*iz;

i++;

c1[i][0]=1./2.\*a+a\*ix;

c1[i][1]=0.+a\*iy;

c1[i][2]=1./2.\*a+a\*iz;

i++;

c1[i][0]=1./2.\*a+a\*ix;

c1[i][1]=1./2.\*a+a\*iy;

c1[i][2]=0.+a\*iz;

i++;

c1[i][0]=1./4.\*a+a\*ix;

c1[i][1]=1./4.\*a+a\*iy;

c1[i][2]=1./4.\*a+a\*iz;

i++;

c1[i][0]=1./4.\*a+a\*ix;

c1[i][1]=3./4.\*a+a\*iy;

c1[i][2]=3./4.\*a+a\*iz;

i++;

c1[i][0]=3./4.\*a+a\*ix;

c1[i][1]=1./4.\*a+a\*iy;

c1[i][2]=3./4.\*a+a\*iz;

i++;

c1[i][0]=3./4.\*a+a\*ix;

c1[i][1]=3./4.\*a+a\*iy;

c1[i][2]=1./4.\*a+a\*iz;

i++;

}}}

//Cl

init\_genrand((unsigned)time(NULL));//現在時刻で乱数の初期化

for(i=0;i<par2;i++){L4:

c2[i][0]=unit\_num\_x\*a\*genrand\_res53();//genrand\_res53():0~1の乱数

c2[i][1]=unit\_num\_y\*a\*genrand\_res53();//0~2aの範囲の値を与える

c2[i][2]=(unit\_num\_z-1)\*a+3./4.\*a+cut1;//基本格子(上面のSi原子)からカットオフ距離だけ離す

if (c2[i][0]==c2[i-1][0]&&c2[i][1]==c2[i-1][1])goto L4;

}

//ミラーセルにコピー

j=1;

for(iy=-1;iy<=1;iy++){

for(ix=-1;ix<=1;ix++){if(ix==0&&iy==0)ix=1;

//Siの上段(原子番号=par1+1,2・・・)

k=par1\*j;

for(i=0;i<par1;i++){

c1[k+i][0]=c1[i][0]+a\_x\*ix;

c1[k+i][1]=c1[i][1]+a\_y\*iy;

c1[k+i][2]=c1[i][2];

}

//Cl(原子番号=par2+1,2・・・)

k=par2\*j;

for(i=0;i<par2;i++){

c2[k+i][0]=c2[i][0]+a\_x\*ix;

c2[k+i][1]=c2[i][1]+a\_y\*iy;

c2[k+i][2]=c2[i][2];

}

j++;

}}

for(iy=-1;iy<=1;iy++){

for(ix=-1;ix<=1;ix++){

//Siの上段

k=par1\*j;

for(i=0;i<par1;i++){

c1[k+i][0]=c1[i][0]+a\_x\*ix;

c1[k+i][1]=c1[i][1]+a\_y\*iy;

c1[k+i][2]=c1[i][2]-a\_z;

}

j++;

}}

//・・・初期速度・・・

//Si

for(i=0;i<par1;i++){

for(j=0;j<3;j++){

v1[i][j]=0;

}}

//Cl

double r1,r2;

r1=sqrt(2\*kb\*T0/m2);//熱確速度

r2=2\*pi;//2π

for(i=0;i<par2;i++){

v2[i][0]=r1\*sqrt(-log(genrand\_res53()))\*cos(r2\*genrand\_res53());

v2[i][1]=r1\*sqrt(-log(genrand\_res53()))\*cos(r2\*genrand\_res53());

v2[i][2]=-r1\*sqrt(-log(genrand\_res53()))\*fabs(cos(r2\*genrand\_res53()));//基盤原子の方向となるように負にする

}

//・・・力の初期条件・・・

for(j=0;j<3;j++){

//Si

for(i=0;i<par1;i++){

f1[i][j]=0;

}

//Cl

for(i=0;i<par2;i++){

f2[i][j]=0;

}}

//・・・・・メインループ・・・・・

for(t=0;t<dt\*loop;t+=dt,count++){

//・・・位置の更新・・・

for(j=0;j<3;j++){

//Si

for(i=0;i<par1;i++){

v1[i][j]=v1[i][j]+dt/2\*f1[i][j]/m1\*ev\_angs;//速度の更新でも使用するので計算回数を減らすために保管しておく(ev\_angsでm/sに変換しているf:ev/Å→J/m)

c1[i][j]=c1[i][j]+dt\*v1[i][j]/angs;//angsでmをÅに変換している

}

//Cl

for(i=0;i<par2;i++){

v2[i][j]=v2[i][j]+dt/2\*f2[i][j]/m2\*ev\_angs;

c2[i][j]=c2[i][j]+dt\*v2[i][j]/angs;

}}

//・・・周期条件・・・

//基本セルの外に出た粒子の修正

//Si

for(i=0;i<par1;i++){

if(c1[i][0]<0){

while(c1[i][0]<0){

c1[i][0]+=a\_x;

}}

else if(c1[i][0]>a\_x){

while(c1[i][0]>a\_x){

c1[i][0]-=a\_x;

}}

if(c1[i][1]<0){

while(c1[i][1]<0){

c1[i][1]+=a\_y;

}}

else if(c1[i][1]>a\_y){

while(c1[i][1]>a\_y){

c1[i][1]-=a\_y;

}}}

//Cl

for(i=0;i<par2;i++){

if(c2[i][0]<0){

while(c2[i][0]<0){

c2[i][0]+=a\_x;

}}

else if(c2[i][0]>a\_x){

while(c2[i][0]>a\_x){

c2[i][0]-=a\_x;

}}

if(c2[i][1]<0){

while(c2[i][1]<0){

c2[i][1]+=a\_y;

}}

else if(c2[i][1]>a\_y){

while(c2[i][1]>a\_y){

c2[i][1]-=a\_y;

}}}

//ミラーセルにコピー

j=1;

for(iy=-1;iy<=1;iy++){

for(ix=-1;ix<=1;ix++){if(ix==0&&iy==0)ix=1;

//Siの上段(原子番号=par1+1,2・・・)

k=par1\*j;

for(i=0;i<par1;i++){

c1[k+i][0]=c1[i][0]+a\_x\*ix;

c1[k+i][1]=c1[i][1]+a\_y\*iy;

c1[k+i][2]=c1[i][2];

}

//Cl(原子番号=par2+1,2・・・)

k=par2\*j;

for(i=0;i<par2;i++){

c2[k+i][0]=c2[i][0]+a\_x\*ix;

c2[k+i][1]=c2[i][1]+a\_y\*iy;

c2[k+i][2]=c2[i][2];

}

j++;

}}

//・・・力の初期化・・・

for(j=0;j<3;j++){

//Si

for(i=0;i<par1;i++){

f1[i][j]=0;

}

//Cl

for(i=0;i<par2;i++){

f2[i][j]=0;

}}

//・・・力の計算・・・

//Si

for(i=0;i<par1\_all;i++){

//カットオフ距離内の原子の保管(j)

for(j=0,l=0;j<par1\_all;j++){

if(j!=i){

if(dis(c1,c1,i,j)<R+D){

num\_j[l]=j;

l++;

}}}

//カットオフ距離内の原子の保管(k)

for(j=0;j<l;j++){

num\_k[j]=num\_j[j];

}

for(j=0;j<l;j++){

//i原子とj原子間におけるfrとfaの計算

r\_ij=dis(c1,c1,i,num\_j[j]);

fr=A\*exp(-lam1\*r\_ij);

fa=-B\*exp(-lam2\*r\_ij);

//i原子とj原子間におけるfcとd(fc)/drの計算

if(R-D<r\_ij){

fc\_ij=1./2.\*(1-sin(pi/2\*(r\_ij-R)/D));

dfc\_ij=-pi/(4\*D)\*cos(pi/2\*(r\_ij-R)/D);

}

else{

fc\_ij=1;

dfc\_ij=0;

}

//ζ(=z)の計算(i原子とk原子間におけるfcの計算)

z=0;

for(k=0;k<l;k++){

if(num\_k[k]!=num\_j[j]){

r\_ik=dis(c1,c1,i,num\_k[k]);

if(R-D<r\_ik){

fc\_ik=1./2.\*(1-sin(pi/2\*(r\_ik-R)/D));

}

else{

fc\_ik=1;

}

g=1+c\*c/d/d-c\*c/(d\*d+pow(h-ang(c1,c1,c1,i,num\_j[j],num\_k[k]),2));

z+=fc\_ik\*g;

}}

//bの計算

b=1+pow(beta,n)\*pow(z,n);

if(z!=0)db=-1./2.\*pow(beta,n)\*pow(z,n-1)\*pow(b,-1/(2\*n)-1);

else db=0;

b=pow(b,-1/(2\*n));

//Fi(力)の計算(第1項)

for(k=0;k<3;k++){

f[k]=(-dfc\_ij\*fr-fc\_ij\*(-lam1\*fr))\*(c1[i][k]-c1[num\_j[j]][k])/r\_ij;//-lam1\*fr=dfr

f[k]+=-b\*(dfc\_ij\*fa+fc\_ij\*(-lam2\*fa))\*(c1[i][k]-c1[num\_j[j]][k])/r\_ij;//-lam2\*fa=dfa

f[k]\*=1./2.;//第1項に×1/2していることになっている

f1[i][k]+=f[k];

f1[num\_j[j]][k]-=f[k];

}

for(k=0;k<l;k++){

if(num\_k[k]!=num\_j[j]){

r\_ik=dis(c1,c1,i,num\_k[k]);

if(R-D<r\_ik){

fc\_ik=1./2.\*(1-sin(pi/2\*(r\_ik-R)/D));

dfc\_ik=-pi/(4\*D)\*cos(pi/2\*(r\_ik-R)/D);

}

else{

fc\_ik=1;

dfc\_ik=0;

}

//gとdgの計算

co=ang(c1,c1,c1,i,num\_j[j],num\_k[k]);

g=1+c\*c/d/d-c\*c/(d\*d+pow(h-co,2));

dg=-2\*c\*c\*(h-co)/pow(d\*d+pow(h-co,2),2);

//∂ζ/∂riの計算(第1項)

for(m=0;m<3;m++){

dz[m]=g\*dfc\_ik\*(c1[i][m]-c1[num\_k[k]][m])/r\_ik;

//∂ζ/∂riの計算

dz\_i[m]=dz[m]+fc\_ik\*dg\*((co/r\_ij-1/r\_ik)\*(c1[num\_j[j]][m]-c1[i][m])/r\_ij+(co/r\_ik-1/r\_ij)\*(c1[num\_k[k]][m]-c1[i][m])/r\_ik);

//Fi(力)の計算

f1[i][m]-=1./2.\*(fc\_ij\*fa\*db\*dz\_i[m]);//第2項に×1/2していることになっている

//∂ζ/∂rjの計算

dz\_j[m]=fc\_ik\*dg/r\_ij\*(-co\*(c1[num\_j[j]][m]-c1[i][m])/r\_ij+(c1[num\_k[k]][m]-c1[i][m])/r\_ik);

//Fj(力)の計算

f1[num\_j[j]][m]-=1./2.\*(fc\_ij\*fa\*db\*dz\_j[m]);//第2項に×1/2していることになっている

//∂ζ/∂rkの計算(-∂ζ/∂rk)

dz\_k[m]=-dz[m]+fc\_ik\*dg/r\_ik\*(-co\*(c1[num\_k[k]][m]-c1[i][m])/r\_ik+(c1[num\_j[j]][m]-c1[i][m])/r\_ij);

//Fk(力)の計算

f1[num\_k[k]][m]-=1./2.\*(fc\_ij\*fa\*db\*dz\_k[m]);//第2項に×1/2していることになっている

}}}}}

//Cl

for(i=0;i<par2\_all;i++){

//近づける原子と基盤原子間に働く力の計算

for(j=0;j<par1\_all;j++){

r\_ij=dis(c2,c1,i,j);

if(r\_ij<cut1){

for(k=0;k<3;k++){

f[k]=-(2\*a1\*E1\*(exp(-a1\*(r\_ij-ra1))-exp(-2\*a1\*(r\_ij-ra1))))\*(c2[i][k]-c1[j][k])/r\_ij;//-∂U/∂r\*(c2-c1)/rでc1→c2方向が正

f2[i][k]+=f[k];

f1[j][k]-=f[k];

}}}

//近づける原子同士の間に働く力の計算

for(j=i+1;j<par2\_all;j++){

r\_ij=dis(c2,c2,i,j);

if(r\_ij<cut2){

for(k=0;k<3;k++){

f[k]=-(2\*a2\*E2\*(exp(-a2\*(r\_ij-ra2))-exp(-2\*a2\*(r\_ij-ra2))))\*(c2[i][k]-c2[j][k])/r\_ij;//-∂U/∂r\*(c2i-c2j)/rでj→i方向が正

f2[i][k]+=f[k];//(斥力が働いていてi原子のほうが正方向にあるときf2[i][k]>0)

f2[j][k]-=f[k];

}}}}

//・・・速度の更新・・・

for(i=0;i<3;i++){

//Si

for(j=0;j<par1;j++){

v1[j][i]=v1[j][i]+dt/2\*f1[j][i]/m1\*ev\_angs;

}

//Cl

for(j=0;j<par2;j++){

v2[j][i]=v2[j][i]+dt/2\*f2[j][i]/m2\*ev\_angs;

}}

//・・・温度の計算・・・

for(i=0,T=0;i<par1;i++){

T+=m1\*(pow(v1[i][0],2)+pow(v1[i][1],2)+pow(v1[i][2],2));

}

for(i=0;i<par2;i++){

T+=m2\*(pow(v2[i][0],2)+pow(v2[i][1],2)+pow(v2[i][2],2));

}

T/=3\*(par1+par2)\*kb;

//・・・温度の補正・・・

x=sqrt(T0/T);

//・・・速度の更新(温度による補正)・・・

for(i=0;i<3;i++){

//Si

for(j=0;j<par1;j++){

v1[j][i]\*=x;

}

//Cl

for(j=0;j<par2;j++){

v2[j][i]\*=x;

}}

//・・・ファイル出力・・・

if(count%50==0){

FILE \*outputfile;//出力ストリーム(ファイルを定義)

char filename[100];

sprintf(filename, "MD%06d.txt",count/50);//MD%06d.txtという文字列を配列filenameに挿入

outputfile = fopen(filename, "w");//ファイルを書き込み用にオープン(MD%06d.txtというファイルを開く)

if (outputfile == NULL) {//オープンに失敗した場合

printf("cannot open\n");// エラーメッセージを出して

exit(1);// 異常終了

}

for(i=0;i<par1;i++){

fprintf(outputfile,"%d 2 %lf %lf %lf\n",i,c1[i][0],c1[i][1],c1[i][2]);//MD%06d.txtというファイルに書く

}

// for(i=par1;i<par1\*18;i++){

//fprintf(outputfile,"%d 3 %lf %lf %lf\n",i,c1[i][0],c1[i][1],c1[i][2]);//ファイルに書く(周期境界条件)今回は必要ない

//}

for(i=0;i<par2;i++){

fprintf(outputfile,"%d 0 %lf %lf %lf\n",i,c2[i][0],c2[i][1],c2[i][2]);//MD%06d.txtというファイルに書く(cl),[通し番号][粒子種][x座標][y座標][z座標]

}

// for(i=par2;i<par2\*9;i++){

//fprintf(outputfile,"%d 0 %lf %lf %lf\n",i,c2[i][0],c2[i][1],c2[i][2]);//ファイルに書く(周期境界条件)今回は必要ない

//}

fclose(outputfile);// ファイルをクローズ(閉じる)

}}

//・・・終了後・・・

printf("fin\n");

}

//・・・・・サブプログラム・・・・・

//・・・距離の計算プログラム・・・

double dis(double atm\_i[][3],double atm\_j[][3],int i,int j)

{

int k;

double r=0;

for(k=0;k<3;k++){

r+=pow(atm\_i[i][k]-atm\_j[j][k],2);

}

r=sqrt(r);

return r;

}

//・・・角度(cos)のプログラム・・・

double ang(double atm\_i[][3],double atm\_j[][3],double atm\_k[][3],int i,int j,int k)

{

double r\_ij,r\_ik,y,cos;

double ij[3],ik[3];

int l;

for(l=0;l<3;l++){

ij[l]=atm\_i[i][l]-atm\_j[j][l];

ik[l]=atm\_i[i][l]-atm\_k[k][l];

y+=ij[l]\*ik[l];

}

r\_ij=dis(atm\_i,atm\_j,i,j);

r\_ik=dis(atm\_i,atm\_k,i,k);

cos=y/(r\_ij\*r\_ik);

return cos;

}