计算物理作业 7

王一杰[a]

[a] 中国科学技术大学

2021 年 11 月 29 日

1 Problem 1

1.1 程序设定

问题描述： 编写用 Verlet 速度算法求解三维分子运动方程的程序。

程序设定：

计算步长 $h = 0.001$,

计算步数 $n = 64000$,

势能构造 $V = -\frac{1}{r} - \frac{1}{|\overrightarrow{r} - \overrightarrow{r_0}|}$, $where\ \overrightarrow{r_0} = (1, 1, 1)$，即具有两个力心的平方反比吸引力。

分子初始参数：

质量：$m = 1$

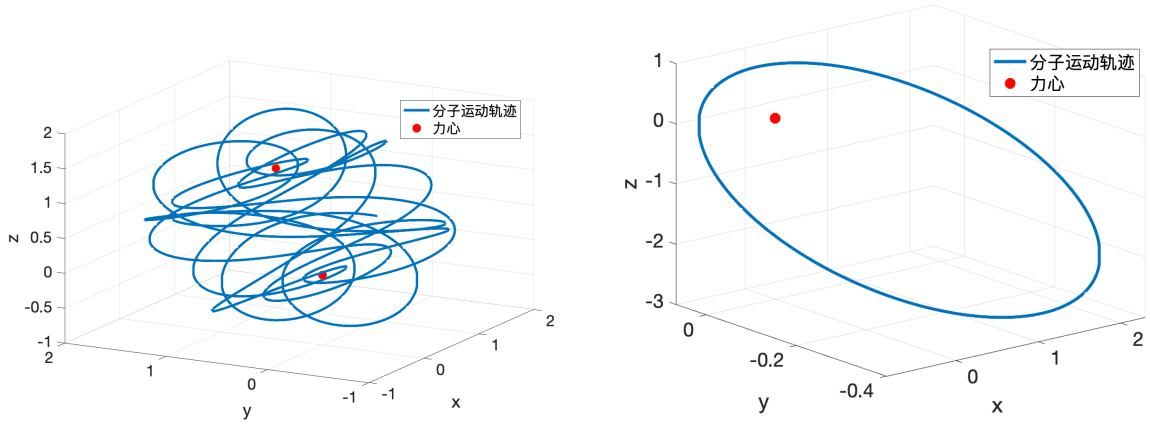
初速度：$(v_x, v_y, v_z) = (-1, 0.1, 0.5)$,

初始位置：$(x, y, z) = (1, 0, 0.5)$,

1.2 程序实现

程序实现如下：

```
n=64000;                                    %计算步数
h=0.001;                                    %计算步长
x=zeros(n+1,3);                             %位置储存单元
v=zeros(n+1,3);                             %速度储存单元
x(1,1)=1;                                   %位置初始化
x(1,2)=0;
x(1,3)=0.5;
v(1,1)=-1;                                  %速度初始化
v(1,2)=0.1;
v(1,3)=0.5;
F=zeros(2,3);                               %作用力存储单元
for i=1:1:n                                 %演化计算
    for j=1:1:3
        F(1,j)=-x(i,j)/(x(i,1)^2+x(i,2)^2+x(i,3)^2)^1.5-(x(i,j)-1)/((x(i,1)-1)^2+(x(i,2)-1)
            ^2+(x(i,3)-1)^2)^1.5;            %F_i计算
        x(i+1,j)=x(i,j)+h*v(i,j)+0.5*F(1,j)*h^2;    %x_i+1计算
    end
```



(a) 双力心轨迹　　(b) 单力心轨迹

图 1: Problem 1 运行结果

```
17      for j=1:1:3
18          F(2,j)=-x(i+1,j)/(x(i+1,1)^2+x(i+1,2)^2+x(i+1,3)^2)^1.5-(x(i+1,j)-1)/((x(i+1,1)-1)
                ^2+(x(i+1,2)-1)^2+(x(i+1,3)-1)^2)^1.5;   %F_i+1计算
19          v(i+1,j)=v(i,j)+0.5*h*(F(2,j)+F(1,j));          %V_i+1计算
20      end
21  end
```

1.3 运行结果

演化结果如图 1 所示，计算结果期望相符，图片分别展示了两个力心的轨迹和单个力心的轨迹（即 $V=-1/r$），与预期相符。可见，计算结果正确。

2 Problem 2

2.1 程序设定

总能量固定的单原子系统的分子动力学模拟。
元胞为 $L_x=L_y=L_z=10$，划分为 $10\times10\times10$ 的正方形网格，
元胞内原子数 $N=64$，
原子质量 $m=1$，
位势为 Lenard-Jones 势，其中 $\varepsilon=\sigma=1$，
边界条件为周期性边界条件，初始位置是随机分布在正则节点上，
初始速度为按 $[-1\ 1]$ 随机分布。
分子动力学模拟步长取为 $\Delta t=0.02$，
模拟 $100\sim200$ 步后原子的速度分布和位置分布。

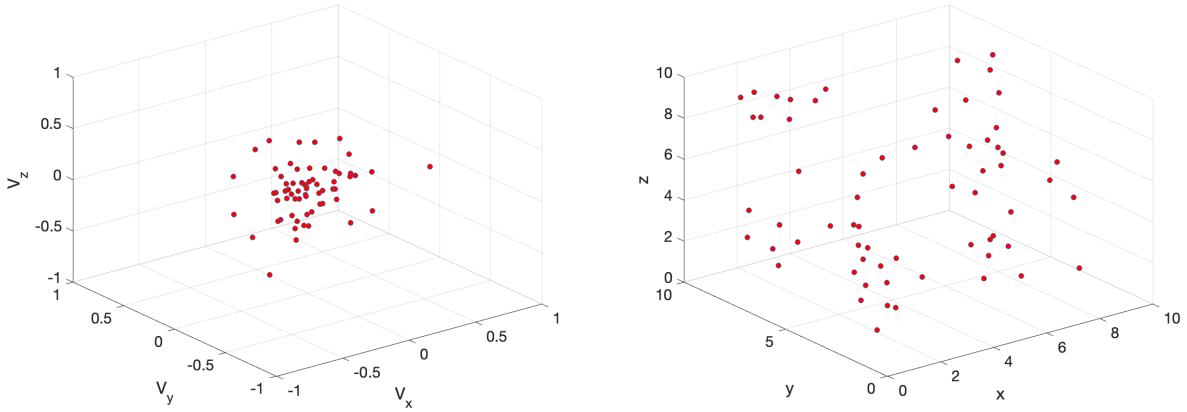
2.2 程序实现

```
1  N=64;                    %粒子数
2  step=200;                %计算步数
```

```
h=0.02;                          %计算步长
x=zeros(N,3);                    %位置储存单元
x1=zeros(N,3);                   %位置临时储存单元
xr=zeros(N,3);                   %相对位置储存单元
flag=0;
E=zeros(step,1);                 %Energy
T=2;                             %温度设定
rc=4;                            %截断半径

I=1;
while(I<N+1)                     %位置随机生成，此处需注意一个格点只能生成一个粒子，否则
    会出现nan
    flag=0;
    x(I,:)=round(0.5+9*rand(1,3));
    for k=1:1:I-1
        if (x(k,1)==x(I,1))&&(x(k,2)==x(I,2))&&(x(k,3)==x(I,3))
            flag=1;
        end
    end
    if flag==1
        I=I-1;
    end
    I=I+1;
end
v=(2*rand(N,3)-1);               %速度随机生成
F1=zeros(64,3);                  %作用力存储单元
F2=zeros(64,3);                  %作用力存储单元

for i=1:1:step                   %演化过程
   for n=1:1:N
       for r=1:1:3               %作用力的初始置0
           F1(n,r)=0;
           F2(n,r)=0;
       end
       for k=1:1:N               %最小像力约定下的相对位置计算
           if k~=n
               for m=1:1:3
                   if x(n,m)-x(k,m)>5
                       xr(k,m)=x(k,m)+10;
                       xr(k,m)=xr(k,m)-x(n,m);
                   elseif x(n,m)-x(k,m)<=-5
                       xr(k,m)=x(k,m)-10;
                       xr(k,m)=xr(k,m)-x(n,m);
                   else
```

```
                    xr(k,m)=x(k,m)-x(n,m);
                end
            end
            for m=1:1:3      %最小像力约定下的相互作用力计算
                F1(n,m)=F1(n,m)-48*xr(k,m)/(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^7+24*xr(k,m)
                    /(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^4;
             if (xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)>rc^2
                    F1(n,m)=F1(n,m)+48*xr(k,m)/(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^7-24*xr(
                        k,m)/(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^4; %截断修正
             end
            end
        end
    end
    for m=1:1:3              %位置演化
       x1(n,m)=mod(x(n,m)+h*v(n,m)+0.5*F1(n,m)*h^2,10);
    end
end

for n=1:1:N
    for k=1:1:N              %最小像力约定下的新时刻相对位置计算
        if k~=n
            for m=1:1:3
                if x1(n,m)-x1(k,m)>5
                    xr(k,m)=x1(k,m)+10;
                    xr(k,m)=xr(k,m)-x1(n,m);
                elseif x1(n,m)-x1(k,m)<=-5
                    xr(k,m)=x1(k,m)-10;
                    xr(k,m)=xr(k,m)-x1(n,m);
                else
                    xr(k,m)=x1(k,m)-x1(n,m);
                end
            end

            for m=1:1:3      %最小像力约定下的新时刻相互作用力计算
                F2(n,m)=F2(n,m)-48*xr(k,m)/(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^7+24*xr(k,m)
                    /(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^4;
                if (xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)>rc^2
                    F2(n,m)=F2(n,m)+48*xr(k,m)/(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^7-24*xr(
                        k,m)/(xr(k,1)^2+xr(k,2)^2+xr(k,3)^2)^4; %截断修正
                end
            end
        end
    end
    for m=1:1:3              %速度演化
```

```
            v(n,m)=v(n,m)+0.5*(F1(n,m)+F2(n,m))*h;
        end
    end
    beta=0;
    for n=1:1:N
        for m=1:1:3
            beta=beta+v(n,m)^2;
        end
    end
    beta=(T*(N-1)/(beta*16))^0.5;                %速度约化因子计算
    for n=1:1:N
        for m=1:1:3
            v(n,m)=beta*v(n,m);                  %速度修正
        end
    end
    for n=1:1:N
       for m=1:1:3
          x(n,m)=x1(n,m);                        %位置临时储存单元输入进位置储存单元
       end
    end
    for n=1:1:N
        for m=1:1:3
            E(i)=E(i)+v(n,m)^2;                  %动能计算
        end
    end
    E(i)=E(i)/2;
    %-------------------------------以下用于监视演化过程
      % figure(1)
      % plot3(v(:,1),v(:,2),v(:,3),'o','MarkerSize',5,'MarkerFaceColor','r')
      % xlim([-0.4,0.4])
      % ylim([-0.4,0.4])
      % zlim([-0.4,0.4])
      % grid on
      % figure(2)
      % plot3(x(:,1),x(:,2),x(:,3),'o','MarkerSize',5,'MarkerFaceColor','r')
      % xlim([0,10])
      % ylim([0,10])
      % zlim([0,10])
      % grid on
      % pause(0.01)
    %-------------------------------演化监视绘图结束
end
figure(1)                                        %速度相空间绘制
plot3(v(:,1),v(:,2),v(:,3),'o','MarkerSize',5,'MarkerFaceColor','r')
```



(a) 速度相空间　　(b) 位置相空间

图 2: Problem 2 运行结果

```
130 xlim([-1,1])
131 ylim([-1,1])
132 zlim([-1,1])
133 xlabel('V_x')
134 ylabel('V_y')
135 zlabel('V_z')
136 set(gca,'FontSize',14)
137 grid on
138
139 figure(2)                                    %位置相空间绘制
140 plot3(x(:,1),x(:,2),x(:,3),'o','MarkerSize',5,'MarkerFaceColor','r')
141 xlim([0,10])
142 ylim([0,10])
143 zlim([0,10])
144 xlabel('x')
145 ylabel('y')
146 zlabel('z')
147 set(gca,'FontSize',14)
148 grid on
```

演化结果如图 2 所示，分别展现了速度相空间和位置相空间的原子分布，计算结果期望相符（速度满足 Maxwell 速度分布率，位置均匀分布）。可见，计算结果正确。