

Classical trajectories

B: Triatomic reaction dynamics with L-J potentials

You will now use the velocity Verlet algorithm to study the dynamics of systems with three atoms (A, B, and C), first of all when the atoms interact via a L-J potential function and then when the atoms interact via a LEPS potential. To make things simple, we will restrict the dynamics of the atoms to be along a line, i.e. only colinear configurations of the atoms will be allowed. It is, first of all, instructive to visualize the potential energy surface. There are only two independent variables in this problem, the distance between atoms A and B, and the distance between atoms B and C. The interaction potential can, therefore, be represented as a surface in 3-D space. With a L-J potential between each pair of atoms, the total potential is

$$V(r_{AB}, r_{BC}) = V_{AB}^{LJ}(r_{AB}) + V_{BC}^{LJ}(r_{BC}) + V_{AC}^{LJ}(r_{AB} + r_{BC})$$

where

$$V_{ij}^{LJ}(r) = 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r} \right)^{12} - \left(\frac{\sigma_{ij}}{r} \right)^6 \right)$$

The parameters in the LJ potential function will be different for different types of atoms, for example the parameters will be different for Ar-Ar interaction than a Ar-Kr interaction.

To keep things simple, we will assume all three atoms are of the same kind, and choose units in such a way that the numerical value of both ϵ and σ are unity.

```
-----  
% param.m  
epsilonAB=1;  
epsilonBC=1;  
epsilonAC=1;  
sigmaAB=1;  
sigmaBC=1;  
sigmaAC=1;  
-----
```

The potential function can be defined in Matlab as

```
-----  
function y=potent(rAB,rBC)  
  
param  
  
y=4*epsilonAB*((sigmaAB./rAB).^12-(sigmaAB./rAB).^6)+4*epsilonBC*((sigmaBC./rBC).^12-(sigmaBC./rBC).^6)+4*epsilonAC*((sigmaAC./(rAB+rBC)).^12-(sigmaAC./(rAB+rBC)).^6);  
-----
```

To keep things simple, we will assume all three atoms are of the same kind, and choose units in such a way that the numerical value of both ϵ and σ are unity.

This interaction potential which depends on two independent variables can be visualized as a surface using the Matlab function mesh.

```
>> rAB=[0.9:0.02:2];
>> rBC=[0.9:0.02:2];
>> [rAB,rBC]=meshgrid(rAB,rBC);
>> mesh(rAB,rBC,potent(rAB,rBC))
```

Note how the potential energy increases when either the AB or the BC distance becomes small. It is hard to see the minimum potential from this angle. The viewing angle can be changed with the view option. For example,

```
>> view(-138,14)
```

Q1: What configuration of the three atoms gives a minimum potential energy (i.e. what values of rAB and rBC)?

Another way of visualizing the potential surface is a contour plot. Usually, this is more convenient, but it is important to keep the 3D image in mind. A contourplot is generated by the contour function or countourf function in Matlab.

```
>> contourf(rAB,rBC,potent(rAB,rBC))
```

Matlab chooses some equal spacing of the contours. It can be better to specify explicitly which values of the energy the contours should correspond to (not necessarily equal spacing). The energies of the contours are defined by the vector v. Also, it will be better for later purposes to increase the range in r1 and r2, and increase the number of points used to calculate the contours (to improve the resolution in the contour lines)

```
-----
% cntr.m
```

```
rAB=[0.9:0.01:3.5];
rBC=[0.9:0.01:3.5];
[rAB,rBC]=meshgrid(rAB,rBC);
v=[-1.75 -1.5 -1.25 -1.0 -0.75 -0.5 -0.25, 0.0 1.0 5.0 10.0];
contourf(rAB,rBC,potent(rAB,rBC),v)
-----
```

Notice the two potential 'valleys' corresponding to A-B distance of ca. 1.2 and corresponding to B-C distance of ca. 1.2. Where the two valleys meet, i.e. both distances at ca. 1.2, there is a potential well. The L-J potential function describes an interaction where all the atoms optimally pack together in a trimer with a 'bond length' of ca. 1.2.

In order to calculate classical trajectories for the system, we need to be able to evaluate the force from the potential function. First of all, defining a function that gives the derivative of the potential energy with respect to the A-B distance

```
-----
function y=dvdr1(rAB,rBC)
```

```
param
```

```
y=4*epsilonAB*(-12*sigmaAB^12*rAB.^-13+6*sigmaAB^6*rAB.^-7)+4*epsilonAC*(
-12*sigmaAC^12*(rAB+rBC).^-13+6*sigmaAC^6*(rAB+rBC).^-7);
-----
```

You should verify that this is the correct expression by carrying out the differentiation yourself. It is not so hard for this potential. Later, you will derive the expression for the derivatives for the LEPS potential which is much more complicated. Then, it is really valuable to know how to do this analytically in Matlab. We also need the derivative of the potential with respect to the B-C distance

```
-----
function y=dvdr2(rAB,rBC)
```

param

$$y=4*\epsilon_{BC}*(-12*\sigma_{BC}^{12}*r_{BC}^{-13}+6*\sigma_{BC}^6*r_{BC}^{-7})+4*\epsilon_{AC}*(-12*\sigma_{AC}^{12}*(r_{AB}+r_{BC})^{-13}+6*\sigma_{AC}^6*(r_{AB}+r_{BC})^{-7});$$

The force on each one of the three atoms can be obtained from the derivatives dV/dr_{AB} and dV/dr_{BC} . Using the chain rule, we have for atom A

$$f_A = -\frac{dV}{dr_{AB}} \frac{dr_{AB}}{dx_A} = \frac{dV}{dr_{AB}}$$

since $r_{AB}=x_B-x_A$, and $dr_{AB}/dx_B = -1$. The force on atom B contains contributions from both dV/dr_{AB} and dV/dr_{BC}

$$f_B = -\frac{dV}{dr_{AB}} + \frac{dV}{dr_{BC}}$$

and the force on atom C is

$$f_C = -\frac{dV}{dr_{BC}}$$

The Matlab commands for implementing the velocity Verlet algorithm in the first part of this exercise (the calculations of the diatomic molecule) can now be extended to the three atom case. Let x_A denote the position of atom A and m_A the mass of atom A, and similarly for the other atoms. We will choose all the masses to be equal and choose units so that the numerical value is unity. Then, specifying the time step size, and total time for the trajectory and then the initial conditions, position and velocity.

```
mA=1.0;
mB=1.0;
mC=1.0;

h=0.02;
totaltime=4.0;
nsteps=round(totaltime/h)+1;

xA(1)=-3.0;
xB(1)=0.0;
xC(1)=1.3;
vA(1)=1.0;
vB(1)=0.0;
vC(1)=0.0;
```

Q2: Describe in words what the initial conditions chosen here correspond to.

The force at the initial step can then be calculated.

```
rABdist(1)=xB(1)-xA(1);
rBCdist(1)=xC(1)-xB(1);
fA(1)=dvdr1(rABdist(1),rBCdist(1));
fB(1)=-dvdr1(rABdist(1),rBCdist(1))+dvdr2(rABdist(1),rBCdist(1));
fC(1)=-dvdr2(rABdist(1),rBCdist(1));
```

The loop over time steps is (read carefully these commands and make sure you understand them, compare with the equations for the velocity Verlet algorithm in the lecture notes)

```
-----  
for n=2:nsteps  
    xA(n)=xA(n-1)+h*vA(n-1)+1/2*h^2*fA(n-1)/mA;  
    xB(n)=xB(n-1)+h*vB(n-1)+1/2*h^2*fB(n-1)/mB;  
    xC(n)=xC(n-1)+h*vC(n-1)+1/2*h^2*fC(n-1)/mC;  
  
    rABdist(n)=xB(n)-xA(n);  
    rBCdist(n)=xC(n)-xB(n);  
  
    fA(n)= dvdr1(rABdist(n),rBCdist(n));  
    fB(n)=-dvdr1(rABdist(n),rBCdist(n))+dvdr2(rABdist(n),rBCdist(n));  
    fC(n)=-dvdr2(rABdist(n),rBCdist(n));  
  
    vA(n)=vA(n-1)+1/2*h*(fA(n)+fA(n-1))/mA;  
    vB(n)=vB(n-1)+1/2*h*(fB(n)+fB(n-1))/mB;  
    vC(n)=vC(n-1)+1/2*h*(fC(n)+fC(n-1))/mC;  
end  
-----
```

The trajectory can now be plotted

```
>> plot(rABdist,rBCdist,'k')
```

It is easier to understand the plot by superimposing it on the contour plot of the potential surface

```
>> cntr  
>> hold on  
>> plot(rABdist,rBCdist,'y')
```

An animation can help illustrate the motion of the atoms along the trajectory

```
-----  
nres=5;  
hold off  
  
% This loop picks out every fifth set of coordinates  
for n=1:nres:nsteps  
    ptA((n-1)/5+1)=xA(n);  
    ptB((n-1)/5+1)=xB(n);  
    ptC((n-1)/5+1)=xC(n);  
end  
  
nanimate=round(nsteps/nres);  
  
for t=1:nanimate  
    plot(ptA(t),0,'ob',ptB(t),0,'ok',ptC(t),0,'oy','MarkerSize',41)  
    axis([-5 5 -5 5])  
    axis off  
    M(t) = getframe;  
end  
  
movie(M,5)  
-----
```

Arrange things so that you can see the plot of the trajectory on top of the contour plot of the potential surface

at the same time that you can see the animation. You can for example make a postscript file of the potential surface with trajectory, so that you can view it alongside the animation. The command to use is

```
>> print -depsc 'potplot'.eps
```

Make sure you understand how the trajectory plot and the animation are related, they are two different illustrations of the same atomic motion.

Q3: Explore the energy dependence of the dynamics. First of all, make sure you have you have all the relevant commands for carrying out the velocity Verlet calculations (initial conditions and the loop over time) in one m file to make repeated execution easier. Make sure you clear all variables having to do with the dynamics. Then, change the initial velocity of atom A to 0.2. The total time of the simulation needs to be increased, `totaltime = 15` would be appropriate in this case. When do you see an exchange reaction and when does a three atom complex form? This is also a function of the phase of the vibration of the BC molecule when the A atom hits. Try changing the phase of the B-C vibration without increasing the amount of vibrational energy in the BC molecule before the collision (how can that be done easily?).

Q4: Increase the velocity of the A atom to 5. At this high energy the attractive part of the interaction between A and B-C is quite irrelevant and the collision is well described as a collision of hard spheres (analogous to billiard balls).