Chemistry 4010 Fall 2019 Assignment 7 Solutions

1. The Hamiltonian is

$$H = \frac{p^2}{2m} + D_e \left(e^{-2a(x-x_e)} - 2e^{-a(x-x_e)}\right).$$

The equations of motion are

$$\begin{aligned}
\frac{dx}{dt} &= \frac{\partial H}{\partial p} = p/m. \\
\frac{dp}{dt} &= -\frac{\partial H}{\partial x} \\
&= -D_e \left(-2ae^{-2a(x-x_e)} + 2ae^{-a(x-x_e)}\right) \\
&= 2aD_e \left(e^{-2a(x-x_e)} - e^{-a(x-x_e)}\right)
\end{aligned}$$

2. Because of its `implicitplot()` function, MAPLE is the simpler tool to use for this task.[1] Here is my Maple session:

Define the Hamiltonian:

$H := (x,p) \mapsto 1/2\,\frac{p^2}{m} + De\,\left(\mathrm{e}^{-2\,a(x-xe)} - 2\,\mathrm{e}^{-a(x-xe)}\right)$

$$H := (x,p) \mapsto 1/2\,\frac{p^2}{m} + De\,\left(\mathrm{e}^{-2\,a(x-xe)} - 2\,\mathrm{e}^{-a(x-xe)}\right)$$

Define the parameters:

$m := 1.6 \times 10^{-27}$

$$m := 1.600000000 \times 10^{-27}$$

$De := 9.0 \times 10^{-19}$

$$De := 9.0000000000 \times 10^{-19}$$

$a := 2.0 \times 10^{10}$

$$a := 20000000000.0$$

$xe := 0.0000000001$

$$xe := 0.0000000001000000000$$

I'm going to store the values of H that I want to use in a list:

$Hvals := [-7.0 \times 10^{-19}, -2.0 \times 10^{-19}]$

$$Hvals := [-7.0000000000 \times 10^{-19}, -2.0000000000 \times 10^{-19}]$$

The largest orbit will have the highest energy. To find its extent (minimum and maximum values of x and p), we can use `solve()`:

$xlim := solve\left(Hvals_2 = H(x,0)\right)$

[1] MATLAB has a `fimplicit()` function that does the same thing, but this function is not available in OCTAVE at this time.

$$xlim := 0.0000000002068184195,\ 6.838545034 \times 10^{-11}$$

$plim := solve\left(Hvals_2 = H\left(xe, p\right)\right)$

$$plim := -4.732863826 \times 10^{-23},\ 4.732863826 \times 10^{-23}$$

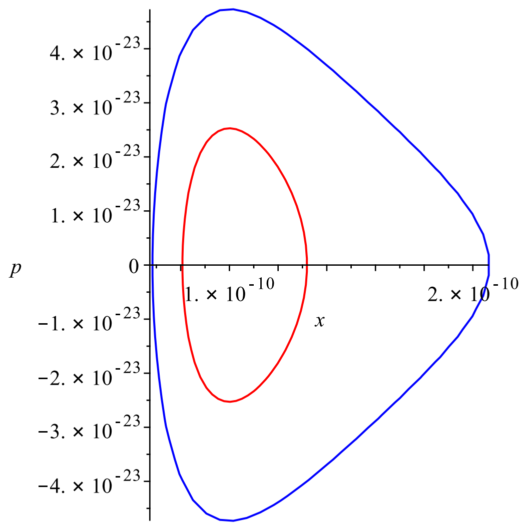
I now have everything I need to call `implicitplot()`.

with(plots):

$p1 := implicitplot(Hvals_2 = H(x,p), x = xlim_2..xlim_1, p = plim_1..plim_2, color = blue):$

$p2 := implicitplot(Hvals_1 = H(x,p), x = xlim_2..xlim_1, p = plim_1..plim_2, color = red):$

$display\left([p1, p2]\right)$



3.

$$
\begin{aligned}
\frac{dV}{dx} &= D_e\left(-2ae^{-2a(x-x_e)} + 2ae^{-a(x-x_e)}\right)\\
&= 2aD_e\left(-e^{-2a(x-x_e)} + e^{-a(x-x_e)}\right)\\
\frac{d^2V}{dx^2} &= 2aD_e\left(2ae^{-2a(x-x_e)} - ae^{-a(x-x_e)}\right)\\
&= 2a^2D_e\left(2e^{-2a(x-x_e)} - e^{-a(x-x_e)}\right)\\
\therefore k &= \left.\frac{d^2V}{dx^2}\right|_{x=x_e} = 2a^2D_e\\
\therefore \nu &= \frac{1}{2\pi}\sqrt{\frac{k}{m}} = \frac{a}{2\pi}\sqrt{\frac{2D_e}{m}}\\
&= \frac{2\times 10^{10}\,\mathrm{m}^{-1}}{2\pi}\sqrt{\frac{2(9\times 10^{-19}\,\mathrm{J})}{1.6\times 10^{-27}\,\mathrm{kg}}}\\
&= 1\times 10^{14}\,\mathrm{Hz}.
\end{aligned}
$$

4. A frequency of 1×10^{14} Hz corresponds to a period of 1×10^{-14} s. I would therefore get 100 periods of oscillation if I set the total integration time to 10^{-12} s. To have any hope of representing the solution accurately, I probably need at least 20 points per period, so I will set $h = 5\times 10^{-16}$ s.

   The semi-implicit Euler scheme is the following:

$$
\begin{aligned}
x_i &= x_{i-1} + hp_{i-1}/m,\\
p_i &= p_{i-1} + 2ahD_e\left(e^{-2a(x_i-x_e)} - e^{-a(x_i-x_e)}\right).
\end{aligned}
$$

   My code is the following:

```
% Implicit-Euler integration for a particle in a Morse potential

% Parameters:
m = 1.6e-27
De = 9e-19
a = 2e10
xe = 1e-10

% Initial conditions:
x(1) = 2e-10
p(1) = 0

% Housekeeping variables
i = 1;

% Data to store
H(1) = p(1)^2/(2*m) + De*(exp(-2*a*(x(1)-xe)) - 2*exp(-a*(x(1)-xe)))
t(1) = 0;
```

```
% Numerical parameters
h = 5e-16
tmax = 1e-12;

% Main loop
while t(i)<tmax
    i = i + 1;
    t(i) = t(i-1) + h;
    x(i) = x(i-1) + h*p(i-1)/m;
    p(i) = p(i-1) + 2*a*De*h*(exp(-2*a*(x(i)-xe)) - exp(-a*(x(i)-xe)));
    H(i) = p(i)^2/(2*m) + De*(exp(-2*a*(x(i)-xe)) - 2*exp(-a*(x(i)-xe)));
end

% Calculate the mean value of H
meanH=mean(H)

% Calculate the period by finding differences between successive peaks in x.
% Note: findpeaks() is not guaranteed to be in every Matlab or Octave
% installation. You may have to do this calculation by hand, e.g. by reading
% the values of t at a few peaks from a graph.
% Even if findpeaks() is present, in Octave you have to explicitly load the
% signal package to access it. For Matlab, comment out the following line.
pkg load signal
[pkval,pkpos] = findpeaks(x);
periods = t(pkpos(2:end))-t(pkpos(1:end-1));
per = mean(periods)
freq = 1/per
```

The value of the Hamiltonian oscillates with a mean of -2.24×10^{-19} J (Fig. 1). The initial value of H was -2.27×10^{-19} J, so this is very reasonable. Although I only show the first few oscillations in the figure, the oscillations continue for as long as I integrated without any noticeable drift away from their mean.

The vibrational frequency found by my simulation is 5.47×10^{13} Hz (calculated using Octave). (I get a slightly lower frequency of 5.37×10^{13} Hz running the same code in Matlab, no doubt due to some small differences in the `findpeaks()` functions in the two systems.) This is a bit more than half of the frequency estimated based on the harmonic oscillator equations. Note that we are using a high energy here, far above the bottom of the potential well where the harmonic approximation would be expected to apply, so this discrepancy is not too surprising.

Figure 2 shows the orbit in phase space. Compared to the exact result, this orbit lacks symmetry with respect to a change of sign of p. The size of the orbit is however approximately correct.

5. With $h = 1.8 \times 10^{-14}$ s, x just increases throughout the simulation. This behavior continues until about $h = 1.2 \times 10^{-15}$ s. At this point, we see a few oscillations before x starts to increase (Fig. 3). This behavior persists until about $h = 7.9 \times 10^{-16}$, where we see our first sustained oscillations. Since the period for this h is 1.8×10^{-14} s, this

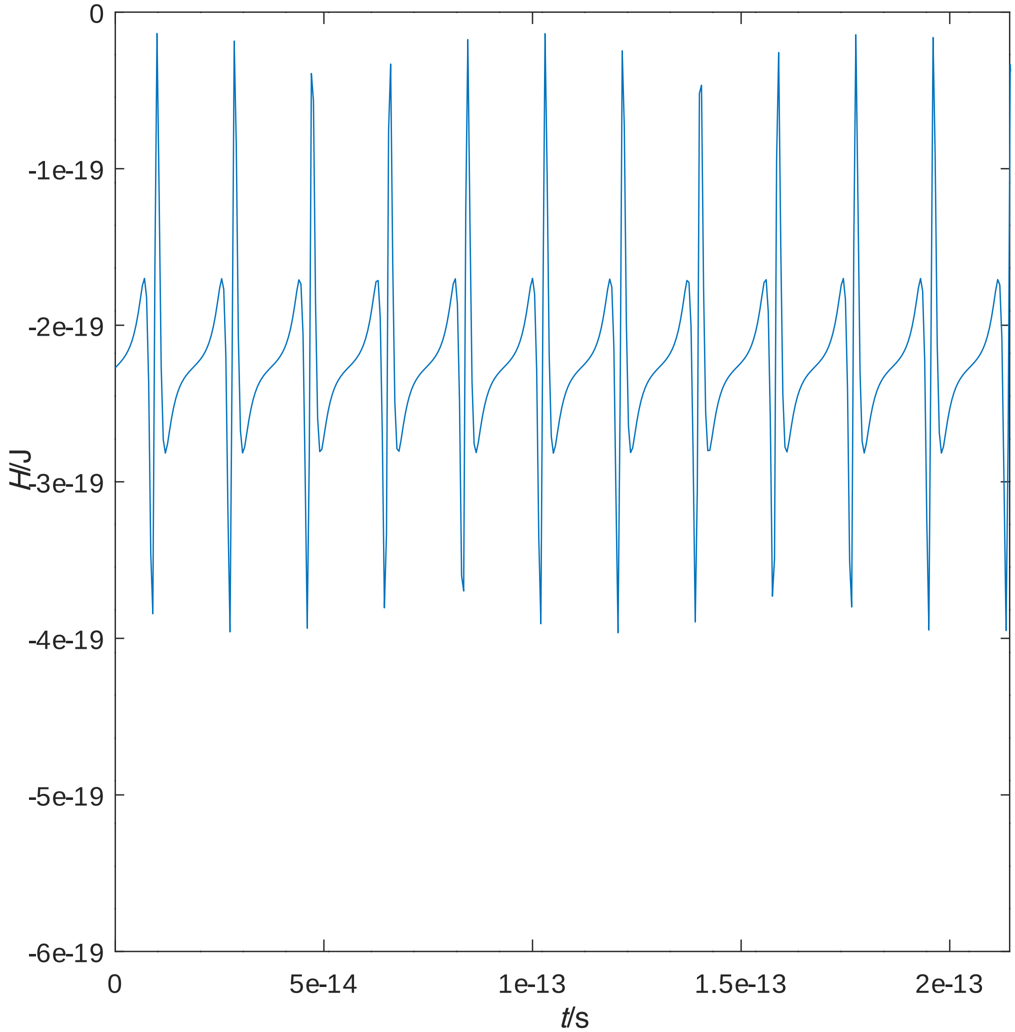


Figure 1: Value of the Hamiltonian for the first 2×10^{-13} s of the simulation

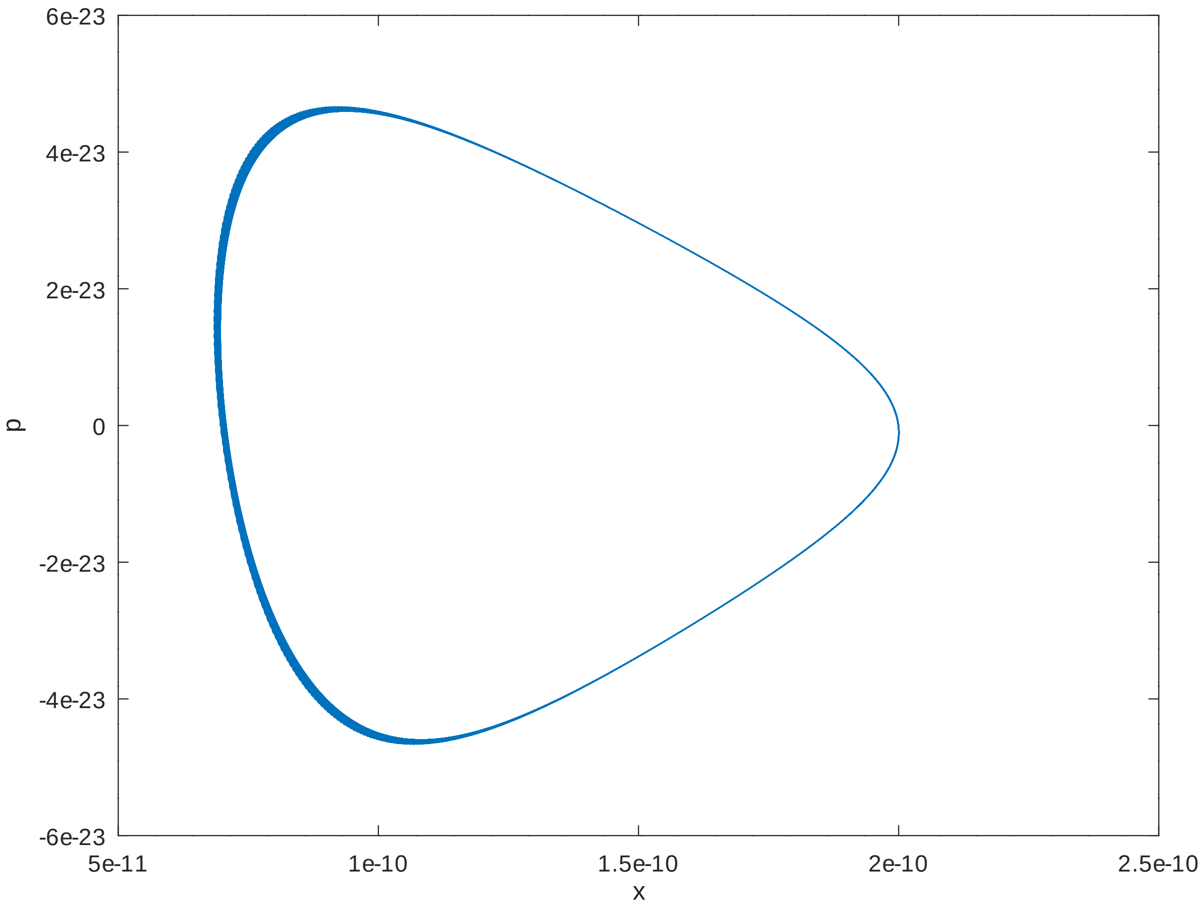


Figure 2: Orbit of the Morse oscillator calculated using the implicit Euler method with $h = 5 \times 10^{-16}\,\mathrm{s}$.

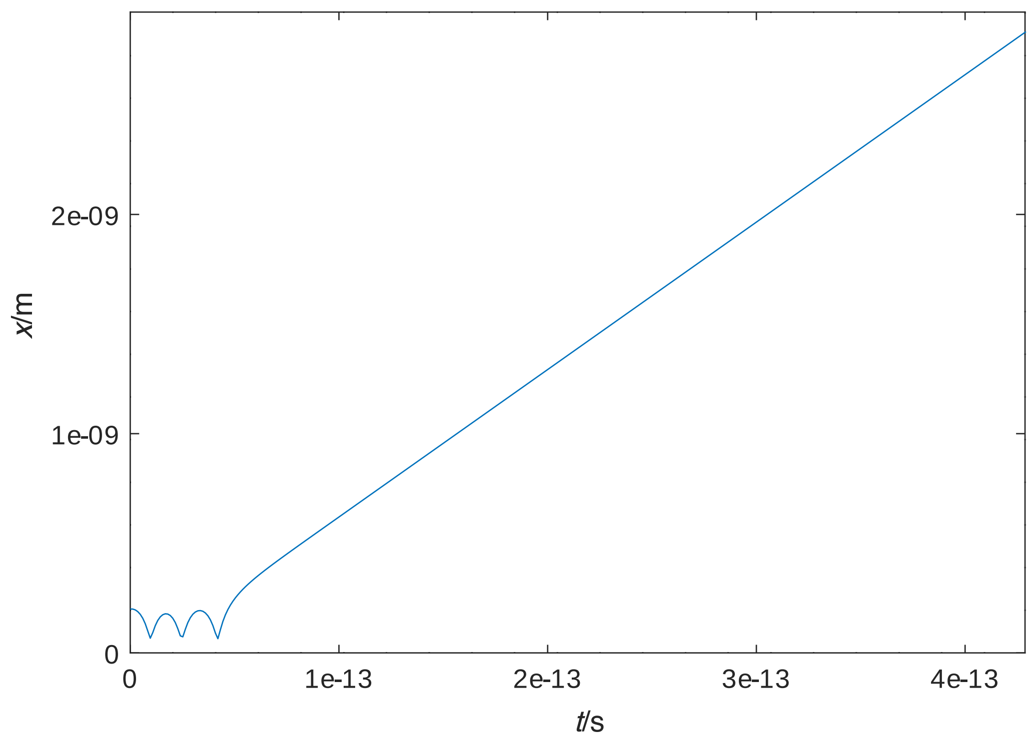


Figure 3: Computed solution for $h = 1.2 \times 10^{-15}$ s.

corresponds to about 23 points per period, which is similar to what I had originally aimed for when choosing h.

We can keep playing this game to see what happens. I wrote a version of my program that varies h systematically and collects some statistics as it does so. Here is the program:

```
% Implicit-Euler integration for a particle in a Morse potential
% Vary the step size h, and store the frequency for each h.

% Parameters:
m = 1.6e-27
De = 9e-19
a = 2e10
xe = 1e-10

% Housekeeping variables
j = 1;

% Data to store

% Numerical parameters
h(1) = 7.9e-16;
hend = 3.6e-17;
tmax = 1e-12;

% Comment out the following line if using Matlab.
pkg load signal

% Loop over values of h
while h(end) > hend
  clear x t

  i = 1;
  t(1) = 0;
  x(1) = 2e-10;
  p = 0;
  H(j,1) = p^2/(2*m) + De*(exp(-2*a*(x(1)-xe)) - 2*exp(-a*(x(1)-xe)));

  while t<tmax
    i = i + 1;
    t(i) = t(i-1) + h(j);
    x(i) = x(i-1) + h(j)*p/m;
    p = p + 2*a*De*h(j)*(exp(-2*a*(x(i)-xe)) - exp(-a*(x(i)-xe)));
    H(j,i) = p^2/(2*m) + De*(exp(-2*a*(x(i)-xe)) - 2*exp(-a*(x(i)-xe)));
  end


  % Calculate the mean value of H
  meanH(j)=mean(H(j,:));

  % Calculate the period by finding differences between successive peaks in x.
```

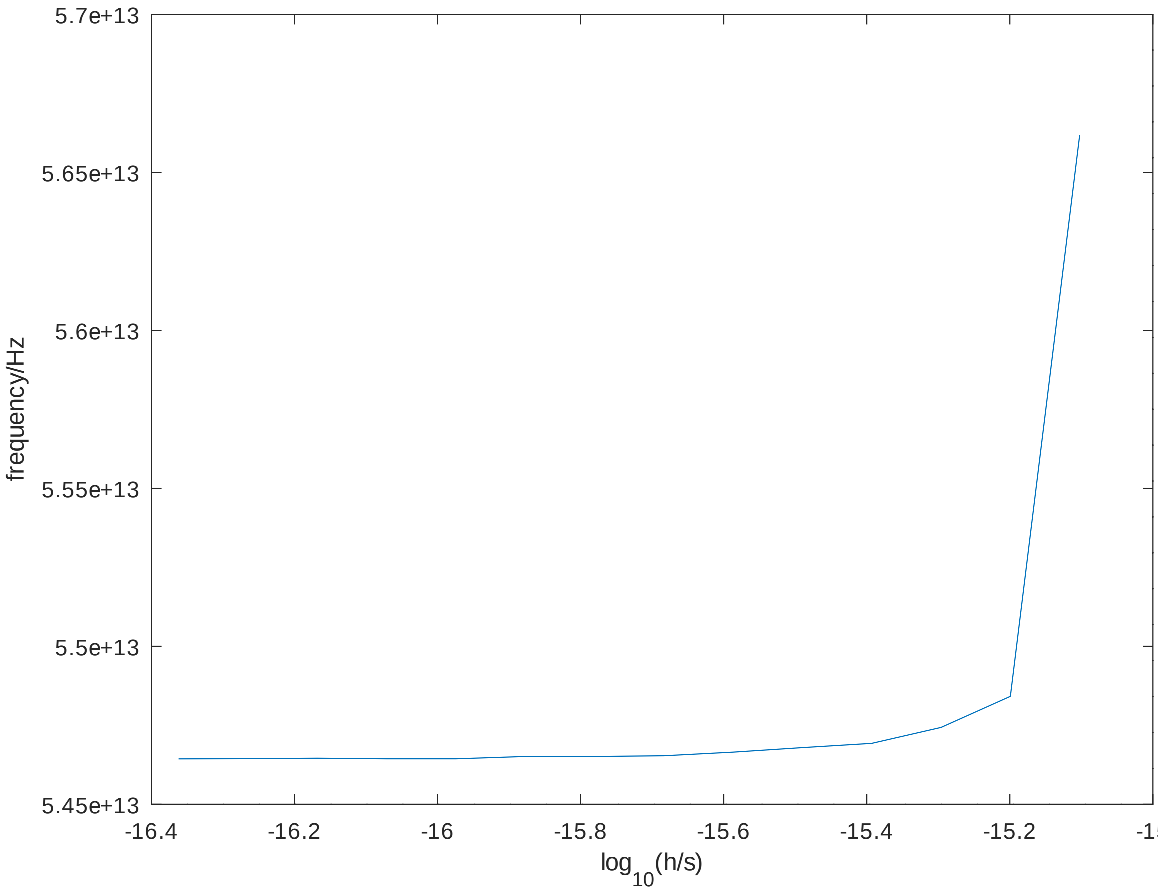


Figure 4: Frequency vs h

```
  [pkval,pkpos] = findpeaks(x);
  periods = t(pkpos(2:end))-t(pkpos(1:end-1));
  per = mean(periods);
  freq(j) = 1/per;

  j=j+1;
  h(j) = 0.8*h(j-1);

end

h=h(1:end-1);
```

The value of the Hamiltonian is more-or-less independent of h. The frequency decreases somewhat as h is decreased from the largest value considered, but this decrease is not dramatic (of the order of 3%, Fig. 4). This is comforting, because it means that this numerical method is relatively robust: as soon as h is small enough to give limit cycles, consistent estimates of the frequency are obtained.