Assignment 4:

This program solves driven morse oscillator model. The hamiltonian is

H(x, p) = p\*\*2/2 + (1 - exp(-x))\*\*2 - a\*x\*sin(wt), where a = 0.04, w = 0.25.

I used two two-dimensional array y and dydt to represent x, p, and the derivatives of them.

y[0] == x

y[1] == p

dy[0]/dt = p

dy[1]/dt = (-2)\*(exp(-x)-exp(-2\*x))+0.04\_dp\*sin(0.25\_dp\*t)

Firstly, I Started 99 trajectories at equally spaced energies between E = 0.01 and E = 0.99 with x(t=0) = 0 and momentum chosen as initial\_p = sqrt(2\*initial\_E) . I computed the time evolution for each of these trajectories with the time step h = 0.1s until arriving maximum time of t = 400. When E > 1 the program will be automatically stopped, since the molecule is ionized. At last, I use Gnuplot to output a graph named driven.png from the data file driven.dat, which shows us how the time for ionizing changes with the initial energy.

The program driven\_Morse.f90 has three major part: main program driven\_Morse and two subroutines derivatives and runge\_kutta\_4.

The subroutine derivatives takes time and the value of x and p as input, outputting the corresponding derivatives for x and p.

The subroutine runge\_kutta\_4 implement the 4th order of runge\_kutta algorithm. It inputs x and p, and their derivatives. What it outputs is the final value for x and p. The subroutine runge\_kutta\_4 calls subroutine derivatives three times.

The Runge-Kutta algorithm are more symmetric than Euler’s method, and it use a trial step at the midpoint of an interval to cancel out lower-order error terms. The second-order formula Requires two function evaluations per step, which makes it better to Euler’s method, because Euler’s method would only reduce the error by a factor of two. The 4th Runge-Kutta algorithm uses Simpson’s formula for integral, thus has error order O(h^5).

As we can see from the graph, when the initial energy is near 0.85, the time for ionization decrease dramatically, which match the prediction from the paper.

