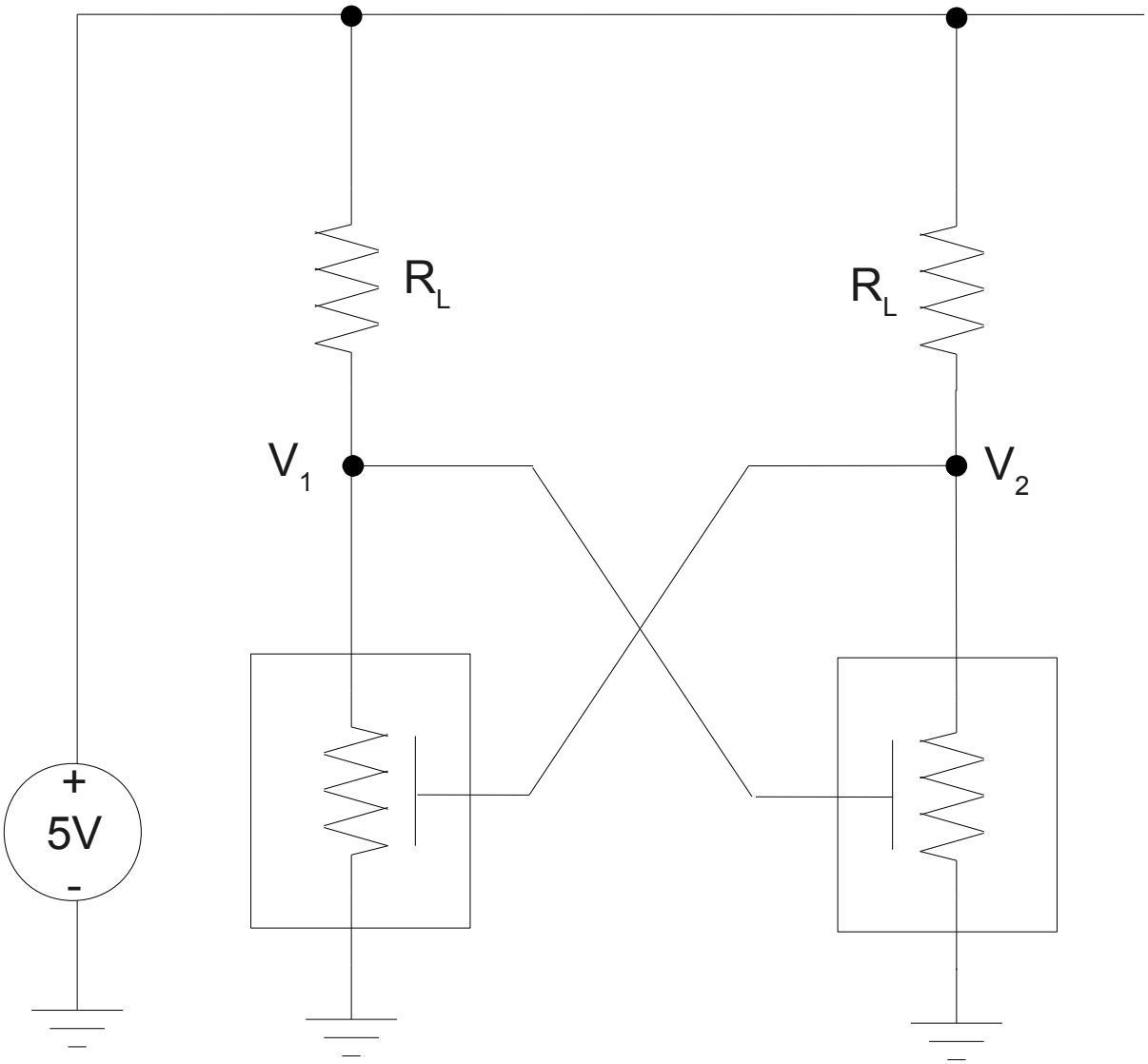
Physics 3340/8300-005 - "Computational Physics" - Fall 2013
Homework 7
Assigned: Monday, 21 October
Due: Friday, 25 October 11:59pm by Blackboard upload

Reading:

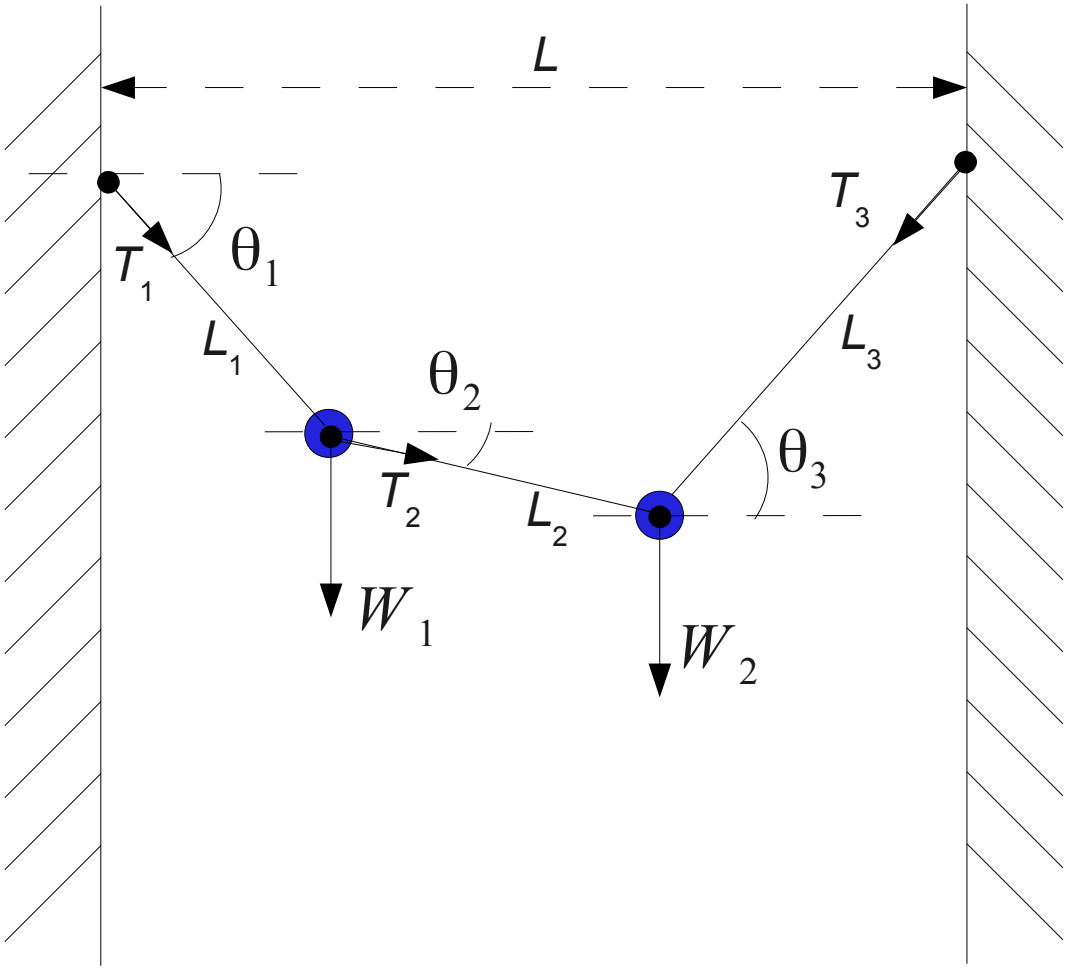
Cheney and Kincaid section 3.2, subsection entitled “Systems of Nonlinear Equations”
Cheney and Kincaid section 4.3, subsection entitled “First Derivative Formulas via Taylor Series”
http://www.physics.smu.edu/fattarus/SysRootsLab.html

Problems:

1) Test whether a new solid state device may be used to store a digital bit in the flip flop circuit of the figure below. See the lecture slides for the electrical behavior of each device. As in those slides, the device current I_D is given by $I_D = \frac{V_D}{R_0} \cdot g(V_G)$ where the gate function $g(V_G) = \frac{e^{(\frac{V_G - V_T}{V_X})}}{1 + e^{(\frac{V_G - V_T}{V_X})}}$ V_D is the voltage across the conductive path in the device, V_G is the voltage at the gate relative to ground, and the parameters in the gate function have the values V_T=2.5V and V_X=0.5V. For these devices the parameter R_0 = 750 ohms, and assume no current flows into the gate terminals. Use $R_L = 8 \times 10^3$ ohms in the circuit. Set up two nonlinear equations that express the Kirchoff's current law at the two nodes V_1 and V_2. Be careful to account for the sign of the device current correctly in KCL. Then use the sys_newton_raphson() function to solve this system as in the first section of the class lab exercise on nonlinear systems. Use the nonlin_sys.c program as a template program, and fill in the 2x2 Jacobian matrix analytically. As in the lab exercise, the program should accept two command line arguments for initial guesses for the two node voltages. This circuit may successfully store a bit of digital information if it has two stable electrical states, corresponding to 0 and 1 binary values. You should find two different starting guesses for the node voltages that cause the Newton-Raphson algorithm to converge to two different sets of the voltages. Submit your main program code that calculates the two node voltages and your two sets of starting guesses. Hint: Ignore any state you will converge to when starting the Newton-Raphson algorithm with two identical starting guess voltages, as this will be an unstable operating point.


R_L
R_L
V_1
V_2
+
5V
-

2) Solve the following static structure by finding the tensions in the three strings T_1, T_2 and T_3, and the angles the three strings make with the horizontal, θ_1, θ_2, and θ_3 :



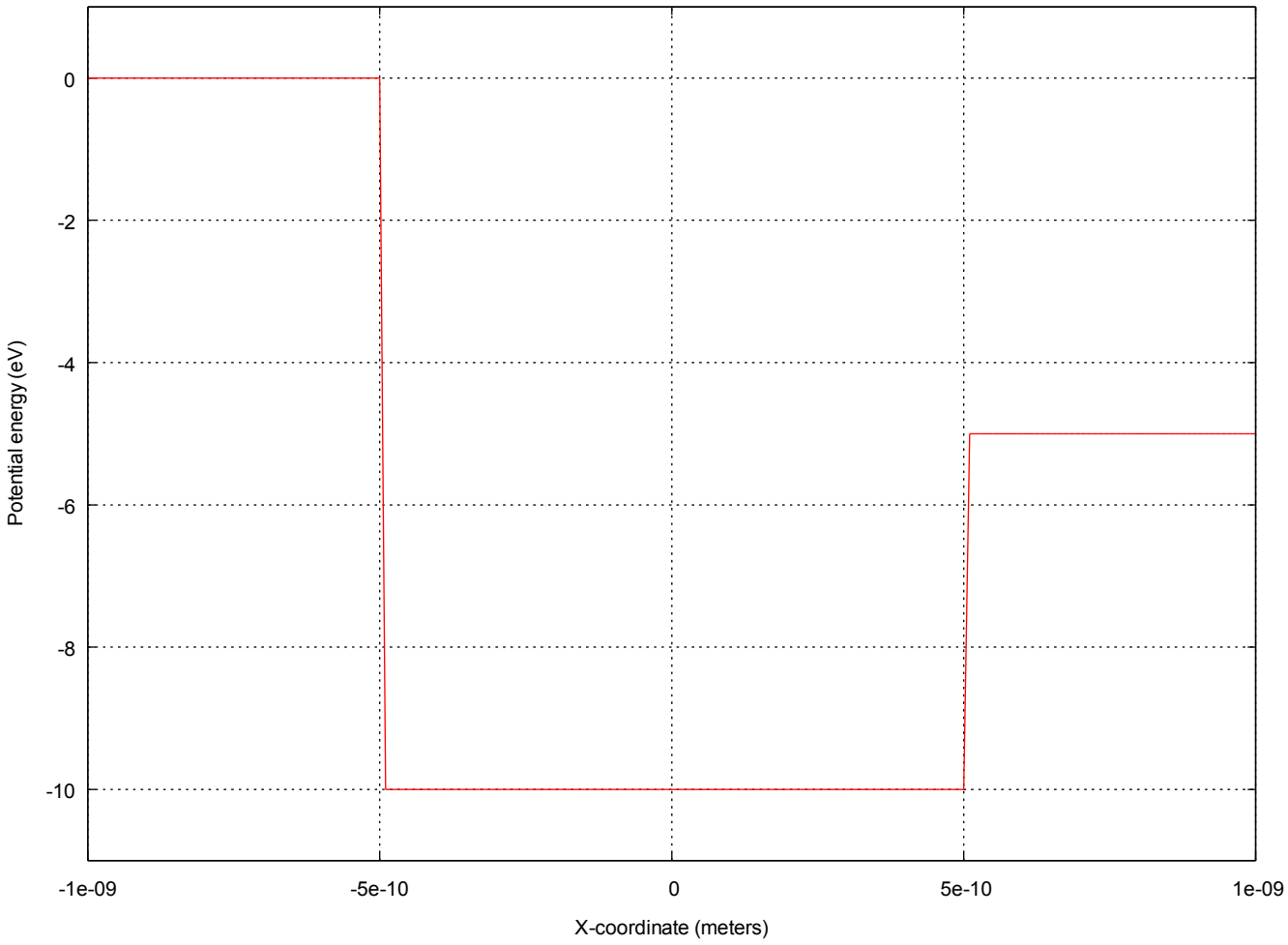
Two weights are connected to the joints in the strings, with weight W_1 and W_2. The lengths of the three strings are L_1, L_2 and L_3, the distance between the two walls that the outer strings are tied to is L, and the two strings are tied to the walls at the same altitude. Note that with flexible strings supporting the weights, the angles of the three tension vectors are now unknown, and part of the set of six variables to solve for. This produces nonlinear equations that must be solved with the multidimensional Newton-Raphson algorithm.

Set up your equations of static equilibrium in the form $\begin{matrix} f_1(\vec{x})=0 \\ f_2(\vec{x})=0 \\ f_3(\vec{x})=0 \\ f_4(\vec{x})=0 \\ f_5(\vec{x})=0 \\ f_6(\vec{x})=0 \end{matrix}$ where the argument vector x is

$\vec{x} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$. The first two equations simply express that the sum of the *x* displacements of each string must be L, and the sum of the *y* displacements of the strings must total to 0. Then the next four equations will be the sum of the *x* components and *y* components of the forces on each weight must of course be zero for the static case.

Then use the sys_newton_raphson() function to solve this system as in the first section of the class lab exercise on nonlinear systems. You may use the nonlin_sys.c program as a template, but adjust the tolerance values at the top of the program appropriately for your units. Following that program, write your main program to accept six command line arguments, for the initial guesses for the six argument components. Use the general numerical Jacobian function as presented in class. Use either radians or degrees for the angles θ_1, θ_2, and θ_3.You may find the constants `RADTODEG` and `DEGTORAD` in the class include file constants.h to be helpful. Use arbitrary units for the dimensions and the forces. Assume L=30 units, L_1=12, L_2=8, L_3=15, W_1=100 units and W_2=175 units. Submit your main program code.

3) Find the lowest three allowed energy levels for an electron bound in a potential well following the plot:



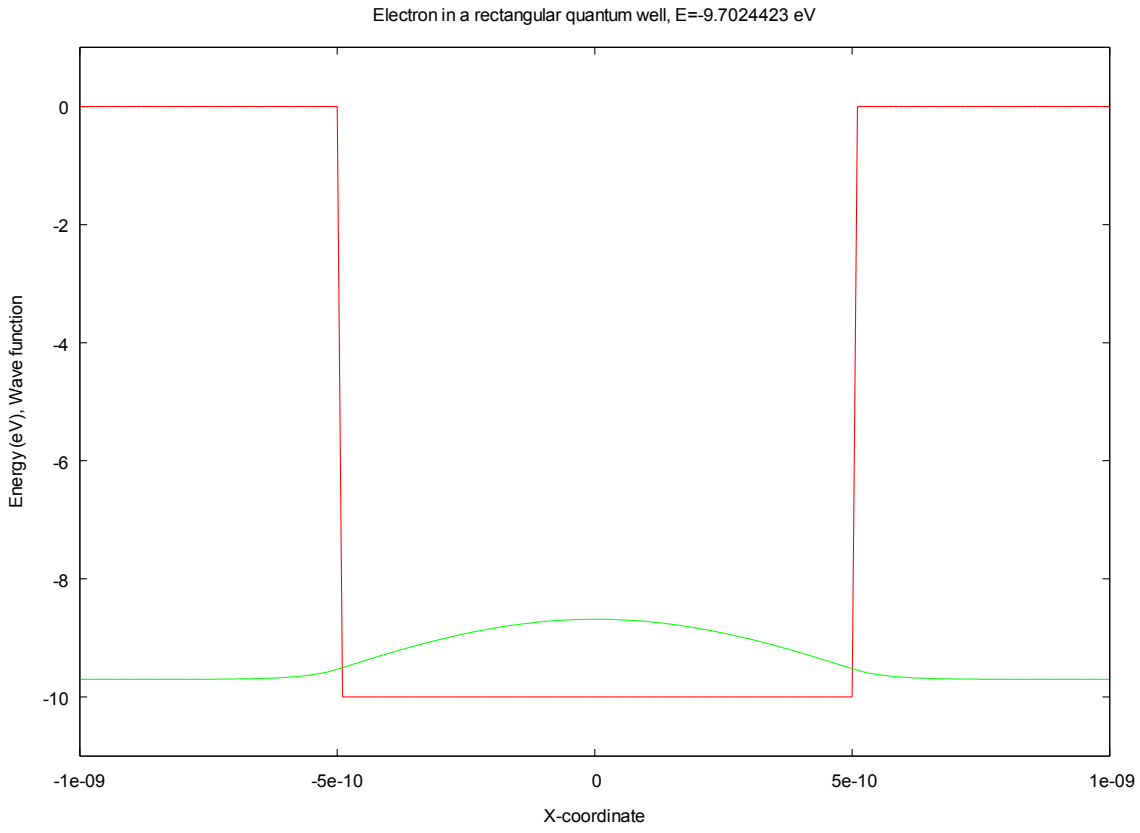
The well width is 1nm, the potential energy left of the well is 0eV, right of the well is -5eV, and inside the well is -10eV.

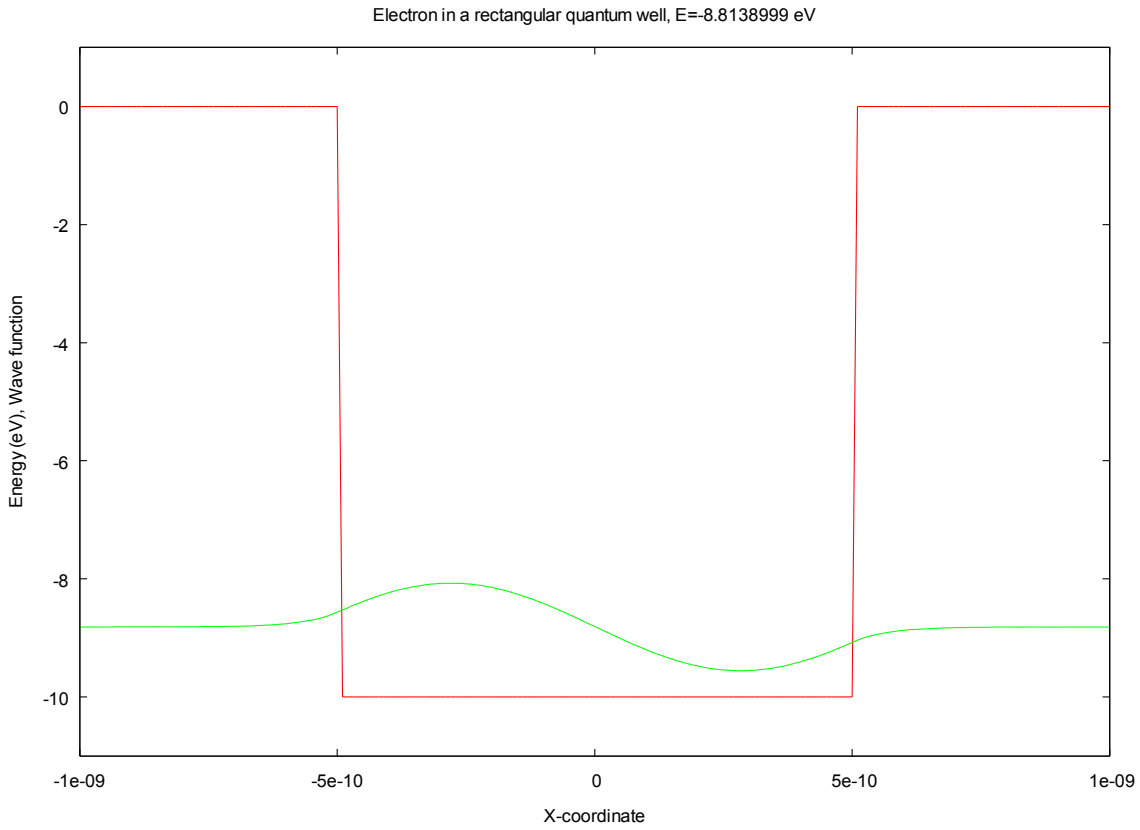
Due to the constraints of normalizability, we know that to the left of the well the wave function solution will follow a growing exponential, $\psi(x)=Ae^{k_1 x}$ with A arbitrarily set to 1. To the right of the well, the wave function solution will follow a decaying exponential, $\psi(x)=Be^{-k_3 x}$ Inside the well the wave function will be oscillatory, $\psi(x)=C\cos(k_2 x)+D\sin(k_2 x)$ The relationship between the electron energy and the three wave numbers in the three regions will be

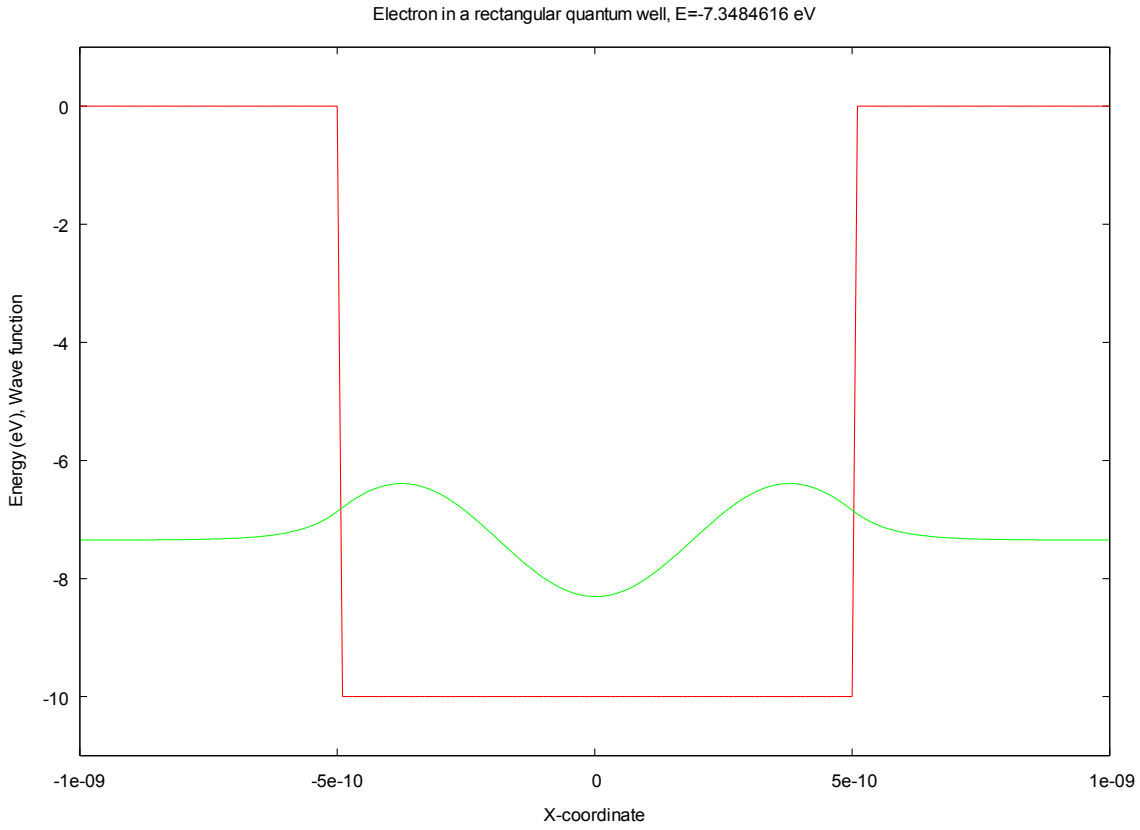
$k_1=\sqrt{\frac{-2mE}{\hbar^2}}$ $k_2=\sqrt{\frac{2m(E+V_0)}{\hbar^2}}$ $k_3=\sqrt{\frac{2m(-V_1-E)}{\hbar^2}}$ with V_0=10eV, V_1=5eV, and the electron mass mc^2=0.5110 MeV. The further constraint that the wave function be continuous in value and first derivative over the two regional boundaries determines a system of four nonlinear equations as a function of B, C, D and E. The allowed electron energies will be the lowest three values of E contained in three solutions of this system.

Use the general numerical Jacobian function as presented in class. A numerical increment and decrement of 10^{-6} for the B, C and D coefficients and 10^{-3} eV for the E level seems to work well in my code. I used convergence tolerance values of 10^{-12} for all four variables.

To start with good initial guesses, you can assume the energy levels will be close to those for a symmetrical well with a barrier height of 10eV on both sides, which were shown in class to be the solutions to a one-dimensional nonlinear equation. The following plots of the lowest three wave functions for the symmetrical case should be helpful in arriving at good initial guesses for the coefficient variables, or at least their signs:







Make a command line argument for each variable starting guess. Submit your source code and your sets of initial values that you get to converge to the three energy levels. Hint: Because of the presence of the

exponential function in the general solution, the coefficients B, C and D in the converged solutions will in general be much less than 1. For the lowest energy level, try initial values of 10^{-2} for B and 10^{-3} for C.