1. (10 points) [Pencil] Derive Equations 34 and 35.

2. (10 points) [MATLAB] Complete the codes on pages 8–10 (see also the handout given out on Friday, 9/29; attached at the end of this assignment for your convenience).

3. (20 points) [MATLAB] i) Improve the FiniteSquareWell_Wavefunctions code on pages 12–14. Annotate it with proper labels for the energy levels and the wavefunctions. Don't forget to label the pertinent points along the horizontal axis. Notice how the wavefunctions leak out of the boundaries of the well. Since $\psi^*\psi dx$ represents the probability of finding the particle in a region dx this implies that there is a finite probability of finding the particle in the classically forbidden region. This is known as *quantum tunneling.*

   ii) Now vary certain parameters of the finite square well, run your program, and observe what happens to the energy levels, the wavefunctions, and to their shapes. Vary the

   (a) width
   (b) barrier height, and
   (c) mass of the particle

   both one variable at a time and in tandem. Please explain the behavior that you observe, in your own words.

4. (10 points) [Pencil] Derive Equations 40 and 41 and show that Equations 36 and 37 can indeed be recast as Equation 42.

5. (10 points) [Pencil] Show that $P_{\text{forbidden}}$ can be written as Equation 51, namely,

$$P_{\text{forbidden}} = \begin{bmatrix} \cosh(\beta(x-b)) & \frac{1}{\beta}\sinh(\beta(x-b)) \\ \beta\sinh(\beta(x-b)) & \cosh(\beta(x-b)) \end{bmatrix}.$$

$$\beta\cos(\alpha a) - \alpha\sin(\alpha a) = 0 \qquad \text{:Even States} \tag{34}$$

$$\alpha\cos(\alpha a) + \beta\sin(\alpha a) = 0 \qquad \text{:Odd States} \tag{35}$$

typical sinusoidal form

$$\psi(x) = A\sin(\alpha x) + B\cos(\alpha x) \tag{36}$$

and its derivative is

$$\psi'(x) = A\alpha\cos(\alpha x) - B\alpha\sin(\alpha x) \tag{37}$$

$$A = \sin(\alpha b)\psi(b) + \frac{1}{\alpha}\cos(\alpha b)\psi'(b) \tag{40}$$

$$B = \cos(\alpha b)\psi(b) - \frac{1}{\alpha}\sin(\alpha b)\psi'(b) \tag{41}$$

```matlab
% This program plots a function for the "even & odd condition" in the
% finite square well and calculates the energy levels.
%
function FiniteWell  % This is the main program with a corresponding end
  %
  clear; close all; clc

  % Specify constants:

  a    = 0.3;               % Well WIDTH 2a (in nanometers)
  V0   = 12.0;              % Well DEPTH V0 in eV
  Mass = 1.0;               % Mass in units of electron masses
  h_bar_SQ = 0.076199682;   % In units of Mass*eV-nm^2
  E = 0:0.1:12;             % Energy is expressed in eV.

  figure('Name', 'The "even" condition','NumberTitle','off');
  plot(E,even(E)); grid
  xlabel('Energy (eV)', 'FontSize', 18, 'FontWeight', 'bold', 'Interpreter', 'Latex')
  ylabel('$$F_{even } $$', 'FontSize', 18, 'FontWeight', 'bold', 'Interpreter', 'Latex')
  title('$$F_{even } $$  as a function of energy', 'FontSize', 18, ...
      'FontWeight', 'bold', 'Interpreter', 'Latex')


  % INSERT the fzero commands here to find the EVEN roots


  % REPEAT the above for the ODD roots

  fprintf('The odd and even energies of the finite potential well alternates. \n')
  fprintf('  \n')
  fprintf('    Even 1    Odd 1    Even 2    Odd 2 \n')
  fprintf('________________________________________ \n')
  disp([even_1, odd_1, even_2, odd_2])
  fprintf(' All energies in eV \n')

  function F_even = even(E) % This is a subfunction or subroutine
    % This FUNCTION evaluates the EVEN condition for the finite square well

    alpha  = sqrt(2*Mass*E/h_bar_SQ);
    beta   = sqrt(2*Mass*(V0-E)/h_bar_SQ);
    F_even = beta.*cos(alpha*a) - alpha.*sin(alpha*a);
  end % function even(E)

  % REPEAT the above for the ODD function

end % function FiniteWell
```