Lab Three
Computational Physics
Kigali Institute of Science and Technology

December 2011

In this lab you will first learn about how to define your own function in matlab. You will then define functions for the two functions that arise when solving for the energy levels for the finite square well.

Once you have these functions defined, use the plot and froot commands to answer the the questions at the end of this sheet. **Hand in your answers to these questions at the end of lab today.** You may work in groups of two or three if you wish.

Creating your own function. Matlab has lots of built-in functions, but sometimes you will need to define your own. Doing so is not hard once you get the hang of it. Let's say you want to make a function $f(x) = sin(2x)$. Let's call this function `doublesine`. To create this function, make a matlab `.m` file called `doublesine`. Type the following in the file:

```
function [x] = doublesine(t)
% This is an example function
% the function is sin(2t)
% Written by __________
% December 22, 2011

x = sin(2*t);
```

Then, in the matlab window, try typing `doublesine(2)`. Then try typing `sin(2*x)`. You should get the same answers.

(It make take a little bit of time to get this to work as I figure out how matlab in the lab is set up. As usual, some patience may be needed.)

When defining a function, the name of the file needs to have the same name as the function. You can then plot your function. For example,

```
x = 0:0.1:10
plot(x,doublesine(x))
```

Functions for the Energy Levels for the Finite Well.
Below are the functions for the even and odd conditions that we derived in class. You can download them from `http://hornacek.coa.edu/dave/KIST` if you don't want to type them in. Save the first function in a file called `even.m` and the second in a file called `odd.m`.

```
function [F_even] = even(E)
% This is the function that corresponds to the equation that
% arises from the EVEN condition when matching wavefunctions.
% F(E) = beta * cosine(alpha*a) - alpha * sine(alpha*a)
% This is for a finite square well with DEPTH V0 and WIDTH 2A.
% Mass is expressed in electron masses.
% Energy is expressed in eV.
% Distance is expressed in nanometers.
%
% Specify parameters:
a    = 0.3; % in nm
V0   = 20; % in eV
Mass = 511000; % in eVc^2
% Specify constants:
```

```
hc = 1240; % in eVnm
% Evaluate the function and return.
alpha  = sqrt( 8*pi^2*Mass*E./(hc^2) );
beta   = sqrt(8*pi^2*Mass*(V0-E)/(hc^2));
F_even = beta.*cos(alpha*a) - alpha.*sin(alpha*a);

function [F_odd] = odd(E)
% This is the function that corresponds to the equation that
% arises from the EVEN condition when matching wavefunctions.
% F(E) = beta * cosine(alpha*a) - alpha * sine(alpha*a)
% This is for a finite square well with DEPTH V0 and WIDTH 2A.
% Mass is expressed in electron masses.
% Energy is expressed in eV.
% Distance is expressed in nanometers.
%
% Written by David Feldman.  21 Dec 2011
% for Computational Physics class at KIST
%
% Specify constants:
a    = 0.3; % in nm
V0   = 20; % in eV
Mass = 511000; % in eVc^2
%
hc = 1240; % in eVnm
% Evaluate the function and return.
alpha  = sqrt(8*pi^2*Mass*E./(hc^2));
beta   = sqrt(8*pi^2*Mass*(V0-E)/(hc^2));
F_odd  = beta.*sin(alpha*a) + alpha.*sin(alpha*a);
```

You now have the two functions `even(E)` and `odd(E)`. You can change V_0 and experiment. You can plot the functions, and then use the `fzero` command to solve for the roots.

Exercises. Do these exercises and turn in your answers at the end of the lab session. For all of these problems, keep the width fixed at $a = 0.3$ nm. For each question, include a short explanation of how you solved the problem.

1. First, set $V_0 = 100000$. This corresponds to an approximately infinite well. Make plots of `even(E)` and `odd(E)` and find the approximate location of the lowest solution. Then use the `fzero` command to find a more exact answer for the lowest energy level. Is your answer what you would expect?

2. Determine all allowed energies if $V_0 = 20$. (Remember that our equations do not hold if $E > V_0$.)

3. For what range of V_0 are there exactly three bound states?

4. Is there any V_0 so small that there is no bound state?