

## Physics 452: Homework #11

Due Tuesday, Feb. 24, 5:00PM, 2008

1. Explore problem 5 from Exam 1 numerically (solution available online).

(a) Show that the expression for the perturbative energy correction can be written in

scaled units as  $E'_{pert} = \frac{3\alpha'}{4\omega'^2} - \frac{9\alpha'^2}{4\omega'^5}$ .

(b) Find how large  $\alpha$  can get before 1<sup>st</sup> and 2<sup>nd</sup>-order perturbation theory misses the ground-state energy by about 10%. Assume that  $\omega = \hbar/ma^2$ , which is  $\omega' = 1$  in scaled unit. Express your answer as a number times  $\hbar^2/(ma^6)$ . HINT: Try matlab code Schrod below.

2. Explore the problem 8.1.

(a) Show that the WKB energy for the ground state can be written in scaled units as

$$E'_{WKB, ground} = \frac{\pi^2}{2} \left[ 1 + \frac{V'_0}{2\pi^2} \right]^2.$$

(b) Find how large  $V_0$  can get before WKB misses the ground-state energy by 10%.

Express your answer as a number times  $\hbar^2/(ma^2)$ . HINT: Try matlab code Eigen below.

(c) For  $V_0$  found in part (b), find an expression for the WKB wave function and graph it on top of the one found numerically. Don't forget to normalize.

```
function Schrod
% This program integrates the 1-D Schrodinger equation for
% a specified initial wave function and static potential.
close all;
% Number of spatial points on grid
nmax=2048;
% Width of grid in units of Bohr radii
xWidth=10;
dx=xWidth/nmax;
x=-dx*(nmax/2):dx:dx*(nmax/2-1);
% Potential in units of hBar^2/(mass*a^2)
alpha=0;
w=1;
V=w^2*x.^2/2+alpha*x.^4;
psi=exp(-x.^2/2);
% Number of time steps
nsteps=4096;
samp=zeros(1,nsteps);
% Integration time in units of mass*a^2/hBar
```

```

tmax=200;
dt=tmax/nsteps;
% Number of movie frames displayed
frames=100;
nframe=round(nsteps/frames);
% Integrate Schrodinger equation using FFT method
dnu=1/(dx*nmax);
nu=-dnu*(nmax/2):dnu:dnu*(nmax/2-1);
nu=fftshift(nu);
for n=1:nsteps;
    % Plot wavefunction occasionally
    if rem(n,nframe)==0
        plot(x,real(psi),'g',
x,imag(psi),'y',x,(abs(psi)).^2,'b',x,V,'r')
        ylim([-1.5 1.5])
        xlabel('x')
        ylabel('|\psi|^2 (blue), Re{\psi} (green), Im{\Psi}
(yellow), V (red)')
        text(-.8*xWidth/2,.75, strcat('t =',num2str(n*dt)))
        drawnow
    end
    %Update wavefunction for next time iteration.
    psi=psi.*exp(-i*dt*V);
    psi2=fft(psi);
    psi2=psi2.*exp(-i*dt/2*(2*pi*nu).^2);
    psi=ifft(psi2);
    % Sample wavefunction at some point at each time
iteration
    % (for later calculation of frequency)
    samp(n)=psi(nmax/2);
end
%
%
%Find the frequency spectrum of the sampled point.
spec=fft(samp);
spec=fftshift(spec);
dfreq=2*pi/(dt*nsteps);
freq=-dfreq*(nsteps/2):dfreq:dfreq*(nsteps/2-1);
%plot the frequency spectrum.
figure
plot(freq,(abs(spec)).^2,'b')
xlim([-10,0])
% Determine the frequency corresponding to the maximum
% of the spectrum. This is an eigen energy.
[maxvalue,n] = max(transpose(abs(spec).^2));
fprintf('Eigen Energy = %7.3f.\n',-freq(n))

```

```

function Eigen
% This code solves for an eigen state of the infinite
% square well with a square step on the left half
% in the bottom.
global V0
% Choose the height of the distortion V0.
V0=0;
% Provide a rough guess of the eigen value E.
E = 1;
trialSol = bvpinit(linspace(-0.5,0.5,10),@InitialGuess,E);
sol = bvp4c(@diffeq,@bc,trialSol);
fprintf('Eigen Energy = %7.3f.\n',sol.parameters)
x = linspace(-0.5,0.5);
psi = deval(sol,x);
plot(x,psi(1,:))
title('Eigenfunction of potential.')
xlabel('x')
ylabel('psi')
% -----
% Derivatives: The top element is the derivative of psi;
% the second element is the derivative of psi-prime.
% The value of the potential step V0 can be modified.
function dydx = diffeq(x,psi,E)
global V0
dydx = [ psi(2)
        -2*(E-V0*(1-sign(x))/2)*psi(1)];
% -----
% Specify boundary conditions: Set psi to zero on left
% and right; set the derivative to -1 on right (arbitrary
% choice --- only affects normalization).
function res = bc(psiLeft,psiRight,E)
res = [ psiLeft(1)
        psiRight(1)
        psiRight(2)+1 ];
% -----
% An initial guess of the solution (top)
% and its derivative (bottom)
function psi0 = InitialGuess(x)
psi0 = [ cos(pi*x)
        -pi*sin(pi*x) ];

```