Physics 452: Homework #14

Due Thursday, Mar. 5, 5:00PM, 2009

1. This is a follow-up on prob. 9.7. Consider an electron in a potential well defined by $V(x)=\frac{1}{2}m\omega^2x^2+\alpha x^4$. Let $\alpha = 1\cdot\hbar^2/ma^6$, where a is a unit of length (say, the Bohr radius $0.528\times10^{-10}\,m$) used to scale position: $x'\equiv x/a$. Similarly, let the natural frequency of the oscillator be $\omega = 1\cdot\hbar/ma^2$. We also apply an external oscillating electric field (i.e., laser), represented by the potential $V_L(x,t)=eE_Lx\cos(\omega_L t)$, where E_L and ω_L are the strength and frequency of the field. Let the field strength be $E_L=\frac{1}{10}\cdot\hbar^2/ema^3$.

(a) What intensity produces this field (in units of W/cm^2), assuming linearly polarized light for which $I=\varepsilon_o cE_L^2/2$?

(b) The eigen energies of the ground state and first excited state are $E_0 = 0.804\,\hbar^2/ma^2$ and $E_1 = 2.738\,\hbar^2/ma^2$. What is the ideal applied frequency ω_L for coupling these two states? Convert your answer also to a wavelength in nanometers.

(c) If the electron starts out in the ground state and the external field is applied, at some special future time, what is the highest probability possible of finding the electron in the 1st excited state? HINT: This is intended to be easy.

(d) Compute the strength of $H'_{01}\equiv\langle\psi_0|exE_L\psi_1\rangle\cos(\omega_L t)$. HINT: This must be determined numerically, since we do not have analytical expressions for the eigen states. The eigen functions are available in files that can be down loaded. The simple program MatElem is useful for computing the integral.

(e) Calculate the amount of time required to complete one Rabi oscillation. How many laser periods correspond to this same time? See if it agrees with the numerical simulation Schrod2.

(f) Determine the 'detuning' $\Delta\omega\equiv\omega_L-\omega_o$, where $\omega_o\equiv(E_1-E_0)/\hbar$, that will cause the probability of finding the electron in the excited state to reach a maximum of only 50% during Rabi oscillations. Also express your answer as a wavelength shift $\Delta\lambda\sim\Delta\omega\,2\pi c/\omega_L^2$ in nanometers.

(g) For the case in part (f), what is the amount of time required for a full Rabi oscillation? Compare with a numerical simulation Schrod2. HINT: You will need to adjust ω_L slightly in the simulation to account for the detuning.

```
function Schrod2
```

```
% This program integrates the 1-D Schrodinger equation for
% a specified intial wave function and static potential.
close all;
% Laser field strength and frequency.
EL=.1;
wL=1.934;
load x0.txt
load p0.txt
load V0.txt
x=transpose(x0);
psi=transpose(p0);
V=transpose(V0);
nmax=length(x);
dx=(x(nmax)-x(1))/(nmax-1);
% Integration time in units of mass*a^2/hBar.
tmax=200;
nsteps=4000;
dt=tmax/nsteps;
% Number of movie frames displayed
frames=400;
nframe=round(nsteps/frames);
% Integrate Schrodinger equation using FFT method
dnu=(1/dx)/(nmax-1);
nu=-dnu*(nmax/2):dnu:dnu*(nmax/2-1);
nu=fftshift(nu);
for n=1:nsteps;
   t=n*dt;
   Vp=EL*x*sin(wL*t);
   Vtot=V+Vp;
   % 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',x,Vp,'r')
    ylim([-1 1])
    xlabel('x')
    ylabel('|\psi|^2 (B), Re{\psi} (G),Im{\psi} (Y), V (R)')
    text(x(nmax/8),.75,strcat('t =',num2str(t)))
    drawnow
  end
  %Update wavefunction for next time iteration.
  psi=psi.*exp(-i*dt*Vtot/2);
  psi2=fft(psi);
  psi2=psi2.*exp(-i*dt/2*(2*pi*nu).^2);
  psi=ifft(psi2);
  psi=psi.*exp(-i*dt*Vtot/2);
end

function matElem
```

```
% Load in arrays for position and two wave functions.
load x0.txt
load p0.txt
load p1.txt
x=transpose(x0);
psi0=transpose(p0);
psi1=transpose(p1);
%Plot, just to see.
plot(x,psi0,'b',x,psi1,'r')
xlabel('x')
ylabel('\psi0 (Blue), \psi1 (Red)')
% Compute and integral (field strength not included).
a=psi1.*x.*psi0;
sum=trapz(x,a);
sum
```

For your reference, this program was used to get the eigen states (not needed).

```
function Eigen2
% This code solves for an eigen state of the infinite
% square well with a square step on the left half
% in the bottom.
%
% Provide a rough guess of the eigen value E.
E = 1;
trialSol = bvpinit(linspace(0,4,10),@InitialGuess,E);
sol = bvp4c(@diffeq,@bc,trialSol);
fprintf('Eigen Energy  = %7.3f.\n',sol.parameters)
x = linspace(0,4);
psi = deval(sol,x);
plot(x,psi(1,:))
title('Eigenfunction of potential.')
xlabel('x')
ylabel('psi')
% Append other side of wave function.
nmax=length(psi);
for n=1:nmax-1;
  p1(n)=psi(1,nmax+1-n); % Minus sign if odd state.
  %p1(n)=-psi(1,nmax+1-n);
  x1(n)=-x(1,nmax+1-n);
end
for n=nmax:2*nmax-1;
  p1(n)=psi(1,n-nmax+1);
  x1(n)=x(1,n-nmax+1);
end
% Normalize wave function.
prob=abs(p1).^2;
sum=trapz(x1,prob);
p1=p1/sqrt(sum);
```

```
% Represent with 2048 points.
num=2048;
dx=(x1(2*nmax-1)-x1(1))/(num-1);
x2=x1(1):dx:x1(2*nmax-1);
p2=interp1(x1,p1,x2,'spline');
% Save x and psi to files.
fid=fopen('p0.txt','w');
fprintf(fid,'%12.8f \n',p2);
fid=fopen('x0.txt','w');
fprintf(fid,'%12.8f \n',x2);
% And save the potential too.
w=1;
alpha=1;
V0=w^2*x2.^2/2+alpha*x2.^4;
fid=fopen('V0.txt','w');
fprintf(fid,'%12.8f \n',V0);
% -----------------------------------------------------------
% 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)
w=1;
alpha=1;
dydx = [ psi(2)
         -2*(E-(w^2*x.^2/2+alpha*x.^4))*psi(1)];
% -----------------------------------------------------------
% Specify boundary conditions:
function res = bc(psiLeft,psiRight,E)
% For an even state, set psi to 1 on left and zero on
right;
% set the derivative to zero at left.
res = [  psiLeft(1)-1
         psiLeft(2)
         psiRight(1)];
% For an odd state, set psi to zero on left and zero on
right;
% set the derivative to 1 at left.
%res = [  psiLeft(1)
%          psiLeft(2)-1
%          psiRight(1)];
%
% -----------------------------------------------------------
% An initial guess of the solution (top)
% and its derivative (bottom)
function psi0 = InitialGuess(x)
psi0 = [  exp(-x.^2)*(1+x.^2)
          -0*x.*exp(-x.^2) ];
```