clear all  % clear all variables at first

% Problem 2 (done in class 2/9/16)
% declare symbolic variables

syms h11 h12 h22 cs sn

cmat=[cs sn;-sn cs];  % transformation matrix
hmat=[h11 h12;h12 h22];  % symmetric hamiltonian matrix

% transform
res=cmat.'*hmat*cmat;  % note the period, this ensures the transpose rather
                      % than the Hermitian adjoint

% res designates the result of C'HC

% we want the 1,2 element of res to vanish
expand(res(1,2))  % this gives (cos^2-sin^2)H12 + cos*sin*(H11-H22)

% but cos^2-sin^2=cos(2theta) and cos*sin=sin(2theta)/2
% so cos(2theta) x 2 H12 = sin(2theta)*(H22-H11)

% or tan(2theta)=2 H12/(H22-H11)  or theta = 0.5*atan(2*H12/(H22-H11))

% now the diagonal matrix elements are

expand(res(1,1))
expand(res(2,2))

% which is:  E1=H11*cos(theta)^2+H22*sin(theta)^2 + H12*sin(2theta)
% problem 3

% The eigenvectors correspond to a 2x2 rotation.  C11 = cos(theta) and
% C12= sin(theta), so
% C11=0.122193;C21=-0.99251;
% theta =atan(C21/C11)

% the second eigenvector
C12=-sin(theta);C22=cos(theta);

%display the result
[C12;C22]

% Problem 4

% part a
% program the 2 term recursion relation

syms u n

H=[1 2*u];  % first two hermite polynomials

% note that H(1) is really H_0 because we can't have zero as an index
% or, in general H(m) is really H_{m-1}

% now use the recursion relation
For m=3:5  % this is equivalent to n=2:4
n=m-1;
H(m)=2*u*H(m-1)-2*(m-2)*H(m-2);  % this is equivalent to H_{n+1}=2uH_n-2nH_{n-1}
end

% simplify the final expressions
H=simplify(expand(H));

% check on normalization
syms alpha x
assume(alpha,'positive')

psi0=(alpha/pi)^(1/4)*exp(-alpha*x^2/2);

% convert H_1(u) into function of x
H1=subs(H(2),u,sqrt(alpha)*x);
psi1=(alpha/pi)^(1/4)*H1*exp(-alpha*x^2/2);

% chose the normalization constant so that N^2 int(psi*psi,-inf,inf)=1
N(1)=1/sqrt(int(psi0*psi0,-inf,inf));
N(2)=1/sqrt(int(psi1*psi1,-inf,inf));

% parts b and c
% convert omega to atomic units
omega=4400/219474.6;
mu = 1822.9*1.0073/2 % mu=(ma+mb)/(ma+mb). for identical masses mu=m/2

% omega = sqrt(k/mu) so
k=omega^2*mu

% part d
syms x
v=.5*k*x^2;ezplot(v-omega/2,[-.3 .3])
grid
[xt yt]=ginput(1)
% part e
syms k4
int(v-.5*k4*x^4,-2*xt,2*xt)
% solve for k4
k4=single(solve(ans==0)) % this redefines k4 as numeric

% part f
% zeroth-order energy of v=0

e0=omega/2;
% the first order correction is <0|H'|0>
% H' = Vqo-Vho
hprime=.5*k4*x^4-v;

% put a particular value of alph into psi0
alph =mu*omega; % let alph be the numeric value, keeping alpha % as a symbolic variable

eel=-k/(4*alph)+3*k4/(8*alph^2);
% convert to numeric value
e1=single(eel); % this is the first order correction
% so the energy through first order is
Efirst=e0+e1
% the 2nd order correction is sum v=2 to infinity |<0|H'|v>|^2/(E0-EV)
% this is |<\Psi|H'|2>|^2/(2*omega) + |<\Psi|H'|4>|^2/(4*omega)

zero_two=-k/(4*alph)+3*k4/(4*alph^2);  % <\Psi|H'|2>
zero_four=sqrt(k6)*k4/(4*alph^2);  % <\Psi|H'|4>

e2=zero_two^2/(-2*omega)+zero_four^2/(-4*omega);

Esecond=e0+e1+e2

% part g

% use anonymous functions (alph is a numeric variable and mu is a constant
% kinetic energy

% clear the variables just to make sure I haven't defined them previously
clear T V
T=@(alph)alph/(4*mu)
% and the potential energy
V=@(alph)3*k4./(8*alph.^2); % note periods for vectorization of function
% and the total energy
energ=@(alph)T(alph)+V(alph);
% plot the energy to find the location of the minimum (look in the region
% of the harmonic oscillator alpha, which we know to be ~ 19
plot(energ,[15 25])

% the minimum is around 19.5

% so, find it exactly
alph_best=fminbnd(energ,19,21)
ebest=energ(alph_best)

% part h

% clear all variables for housekeeping
clear all

% syms De re r beta

V_morse=De*(exp(-2*beta*(r-re))-2*exp(-beta*(r-re)))
% the force constant is defined as the 2nd derivative of the potential
% at r=re
k=subs(diff(V_morse,r,2),r,re)
% solve for beta, giving beta=sqrt(k/2*De)
% now we know

omega=4400/219474.6;

mu = 1822.9*1.0073/2;

De=4.52*3.6749e-2  % convert to atomic units
% therefore
k=omega^2*mu
beta=sqrt(k/(2*De))

% part i

% the expression is a power series in v+1/2
% so, using matlab we can write
\% Ev = C(1)(v+1/2)^2 + C(2)(v+1/2)+C(3)
C=[-omega^2/(4*De) omega 0];
\% then we can look for the zero of the polynomial, just like when we
\% looked for a minimum in a polynomial fit
roots(polyder(C))
\% the answer is 16.6. This is vmax+1/2, or Vmax=16 (rounding to an integer)
\% Problem 8
\% define mass and approximate spacing for dvr_quartic_2016.m
mu=918.58
\% loop through five different spacings, saving the energy values
h_val=[0.1 0.05 0.01 0.005 0.001];
\% the length of the hh array
lnh=length(h_val);
\% the length of the hh array
for ih=1:lnh
    hh=h_val(ih);
dvr_quartic_2016;
    ee(ih,:)=ans(4:5); \% save the first two energies in the array ee
end
\% determine the difference between the energies
format short e
delta_e=ee(2:lnh,:)-ee(1:lnh-1,:)
format short
\% the energies calculated using h=0.001 differ less than 1e-5 hartree from
\% those calculated with h=0.005
\% so use 0.005 as the initial spacing
hh=0.005;
\% now vary the masses (H2 HD D2)
mu_val = 918.25*[,5 2/3 1];
\% just look at lowest energy
lnm=length(mu_val);
for imu=1:lnm
    mu=mu_val(imu);
dvr_quartic_2016
    emu(imu)=energy(4)
end
\% plot these energies as a function of 1/sqrt(mu)
plot(1./sqrt(mu_val),emu,'-o');set(gca,'fontsize',16);xlabel('1/sqrt(mu)');
ylabel('E_0')
\% these are almost a straight line, but not quite
\% expand in a power series in 1/sqrt(mu)
cc=polyfit(1./sqrt(mu_val),emu,2)
\% if there were a pure dependence on 1/sqrt(mu) the first coefficient,
\% which is the coefficient of (1/sqrt(mu))^2 = 1/mu, would be zero, which
\% it is not
\% to see the non-linear dependence, use a larger range of mu
mu_valn=918.25*[1.2;1.2;2];
lnnm=length(mu_valn);
for imu=1:lnmm
    mu=mu_valn(imu);
    dvr_quartic_2016
    emun(imu)=energy(4) % just look at lowest energy
end

plot(1./sqrt(mu_valn),emun,1./sqrt(mu_valn),emu,'o') % Problem 5

syms a11 a12 a22
A=[a11 a12;a12 a22] % symmetric matrix

syms u11 u12 u22
U=[u11 u12;0 u22] % upper triangular matrix

res=U.'*U % use .' to get transpose not hermitian adjoint

% this should equal A
% so we have three equations
% u11^2=a11 or u11=sqrt(a11) no problem with sqrt since A is positive
% (or zero)
% then u11*u12=a12 or u12=a12/u11 or u12 = a12/sqrt(a11)
% finally u12^2+u22^2 = a22 or u22=a22-u12^2 = a22-a12^2/a11
UU=subs(U,[u11 u12 u22],[sqrt(a11) a12/sqrt(a11) sqrt(a22-a12^2/a11)]);

% check that this works
UU.'*UU-A % it works, this is zero

% now, let's try a 3x3 matrix

syms u11 u12 u13 u22 u23 u33
U3=[u11 u12 u13;0 u22 u23;0 0 u33];
syms a11 a12 a13 a22 a23 a33
A=[a11 a12 a13;a22 a23 a33]

res=U*U; % the upper left 2x2 matrix is identical so that the same relations hold as before
U3(1:2,1:2)=UU

res=U3.'*U3;

% let's work with the remaining elements
% first the 1,3 element sqrt(a11)*u13; this has to equal a13
% thus u13=a13/sqrt(a11);
U3=subs(U3,u13,a13/sqrt(a11));
res=U3.'*U3;

% now, the 2,3 element has to equal a23
U3=subs(U3,u23,solve(res(2,3)==a23,u23)); % this give the value of u23, which is pretty complicated
res=U3.'*U3;

% now, the 3,3 element has to equal a33
solve(res(3,3)==a33);
uu33=ans(1); % take positive square root
U3=subs(U3,u33,uu33)

% complicated expression, but the result is correct
simplify(U3.'*U3-A) % this equals zero

%% Problem 6
hmat=[.29 -.44;-.44 -.39];
smat=[1 .68;.68 1];
b2=1/sqrt(1-smat(1,2)^2);
a2=-smat(1,2)*b2;
% the gram-schmidt coefficients are (column order) this is matrix i call
% in the notes D_GS^(2)
Dgs=[1 a2;0 b2]
% this normalizes the overlap matrix
Dgs'*smat*Dgs % this equals 1
% then in the GS basis the matrix of the Hamiltonian is
Hgs=Dgs'*hmat*Dgs;
% now, diagonalize this
[Cgs, Egs]=eig(Hgs);
% the eigenvectors are orthogonal, so that Cgs'*Cgs=1
Cgs*Cgs'
% the eigenvectors in the original basis are
C=Dgs*Cgs;
% compare with generalized eigenvalue solution
[CC, EEgs]=eig(hmat,smat);
CC-C % this should be zero