9.9.

b) Now do the same for all states but using the operators a and $a^\dagger$.

$$a = \frac{1}{\sqrt{2\hbar m\omega}}\left(-ip + m\omega x\right)$$
$$a^\dagger = \frac{1}{\sqrt{2\hbar m\omega}}\left(ip + m\omega x\right)$$
$$x = \sqrt{\frac{\hbar}{2m\omega}}\left(a^\dagger + a\right)$$
$$p = i\sqrt{\frac{\hbar m\omega}{2}}\left(a^\dagger - a\right)$$

$$\begin{aligned}\langle x\rangle &= \langle n|x|n\rangle = \sqrt{\frac{\hbar}{2m\omega}}\langle n|a^\dagger + a|n\rangle \\ &= \sqrt{\frac{\hbar}{2m\omega}}\left[\langle n|a^\dagger|n\rangle + \langle n|a|n\rangle\right] = \sqrt{\frac{\hbar}{2m\omega}}\left[\langle n|\sqrt{n+1}|n+1\rangle + \langle n|\sqrt{n}|n-1\rangle\right] \\ &= \sqrt{\frac{\hbar}{2m\omega}}\left[\sqrt{n+1}\langle n|n+1\rangle + \sqrt{n}\langle n|n-1\rangle\right] = 0 \quad \text{since} \quad \langle n|m\rangle = \delta_{nm}\end{aligned}$$

$$\begin{aligned}\langle p\rangle &= \langle n|p|n\rangle = \sqrt{\frac{\hbar}{2m\omega}}\langle n|a^\dagger - a|n\rangle \\ &= \sqrt{\frac{\hbar}{2m\omega}}\left[\langle n|a^\dagger|n\rangle - \langle n|a|n\rangle\right] = \sqrt{\frac{\hbar}{2m\omega}}\left[\langle n|\sqrt{n+1}|n+1\rangle - \langle n|\sqrt{n}|n-1\rangle\right] \\ &= \sqrt{\frac{\hbar}{2m\omega}}\left[\sqrt{n+1}\langle n|n+1\rangle - \sqrt{n}\langle n|n-1\rangle\right] = 0 \quad \text{since} \quad \langle n|m\rangle = \delta_{nm}\end{aligned}$$

Note also that $\langle n|a^2|n\rangle = 0$ and $\langle n|\left(a^\dagger\right)^2|n\rangle = 0$ in a similar manner, so that

$$\begin{aligned}\langle x^2\rangle &= \langle n|x^2|n\rangle = \frac{\hbar}{2m\omega}\langle n|\left(a^\dagger + a\right)^2|n\rangle = \frac{\hbar}{2m\omega}\langle n|\left(a^\dagger\right)^2 + a^\dagger a + aa^\dagger + a^2|n\rangle \\ &= \frac{\hbar}{2m\omega}\langle n|a^\dagger a + aa^\dagger|n\rangle = \frac{\hbar}{2m\omega}\langle n|\sqrt{n}\sqrt{n} + \sqrt{n+1}\sqrt{n+1}|n\rangle \\ &= \frac{\hbar}{2m\omega}\left(2n+1\right) = \frac{\hbar}{m\omega}\left(n + \tfrac{1}{2}\right)\end{aligned}$$

$$\begin{aligned}\langle p^2\rangle &= \langle n|p^2|n\rangle = -\frac{\hbar m\omega}{2}\langle n|\left(a^\dagger - a\right)^2|n\rangle = -\frac{\hbar m\omega}{2}\langle n|\left(a^\dagger\right)^2 - a^\dagger a - aa^\dagger + a^2|n\rangle \\ &= \frac{\hbar m\omega}{2}\langle n|a^\dagger a + aa^\dagger|n\rangle = \frac{\hbar}{2m\omega}\langle n|\sqrt{n}\sqrt{n} + \sqrt{n+1}\sqrt{n+1}|n\rangle \\ &= \frac{\hbar m\omega}{2}\left(2n+1\right) = \hbar m\omega\left(n + \tfrac{1}{2}\right)\end{aligned}$$

9.11 a) Normalize:

$$|\psi(t=0)\rangle = A\left[|0\rangle + 2e^{i\pi/2}|1\rangle\right]$$

$$\begin{aligned}1 = \langle\psi|\psi\rangle &= A^*\left(\langle 0| + 2e^{-i\pi/2}\langle 1|\right)A\left(|0\rangle + 2e^{i\pi/2}|1\rangle\right)\\ &= |A|^2(1+4) = |A|^2\,5\\ &\Rightarrow A = \tfrac{1}{\sqrt{5}}\end{aligned}$$

b) Time evolution

$$\begin{aligned}|\psi(t)\rangle &= \tfrac{1}{\sqrt{5}}\left(e^{-iE_0t/\hbar}|0\rangle + 2e^{i\pi/2}e^{-iE_1t/\hbar}|1\rangle\right)\\ &= e^{-i\omega t/2}\,\tfrac{1}{\sqrt{5}}\left(|0\rangle + 2e^{i\pi/2}e^{-i\omega t}|1\rangle\right)\end{aligned}$$

c) Expectations values with ladder ops:

$$\begin{aligned}\langle x\rangle &= \langle\psi(t)|x|\psi(t)\rangle = \sqrt{\tfrac{\hbar}{2m\omega}}\langle\psi(t)|a^\dagger + a|\psi(t)\rangle\\ &= \sqrt{\tfrac{\hbar}{2m\omega}}e^{+i\omega t/2}\,\tfrac{1}{\sqrt{5}}\left(\langle 0| + 2e^{-i\pi/2}e^{+i\omega t}\langle 1|\right)\left(a^\dagger + a\right)e^{-i\omega t/2}\,\tfrac{1}{\sqrt{5}}\left(|0\rangle + 2e^{i\pi/2}e^{-i\omega t}|1\rangle\right)\\ &= \sqrt{\tfrac{\hbar}{2m\omega}}\,\tfrac{1}{5}\left[2e^{i\pi/2}e^{-i\omega t}\langle 0|a|1\rangle + 2e^{-i\pi/2}e^{+i\omega t}\langle 1|a^\dagger|0\rangle\right]\\ &= \sqrt{\tfrac{\hbar}{2m\omega}}\,\tfrac{2}{5}\left[ie^{-i\omega t}\sqrt{1} - ie^{+i\omega t}\sqrt{1}\right] = \sqrt{\tfrac{\hbar}{2m\omega}}\,\tfrac{4}{5}\sin\omega t\end{aligned}$$

Momentum expectation value:

$$\begin{aligned}\langle p\rangle &= \langle\psi(t)|p|\psi(t)\rangle = i\sqrt{\tfrac{m\omega\hbar}{2}}\langle\psi(t)|a^\dagger - a|\psi(t)\rangle\\ &= i\sqrt{\tfrac{m\omega\hbar}{2}}e^{+i\omega t/2}\,\tfrac{1}{\sqrt{5}}\left(\langle 0| + 2e^{-i\pi/2}e^{+i\omega t}\langle 1|\right)\left(a^\dagger - a\right)e^{-i\omega t/2}\,\tfrac{1}{\sqrt{5}}\left(|0\rangle + 2e^{i\pi/2}e^{-i\omega t}|1\rangle\right)\\ &= i\sqrt{\tfrac{m\omega\hbar}{2}}\,\tfrac{1}{5}\left[-2e^{i\pi/2}e^{-i\omega t}\langle 0|a|1\rangle + 2e^{-i\pi/2}e^{+i\omega t}\langle 1|a^\dagger|0\rangle\right]\\ &= i\sqrt{\tfrac{m\omega\hbar}{2}}\,\tfrac{2}{5}\left[-ie^{-i\omega t}\sqrt{1} - ie^{+i\omega t}\sqrt{1}\right] = \sqrt{\tfrac{m\omega\hbar}{2}}\,\tfrac{4}{5}\cos\omega t\end{aligned}$$

Ehrenfest's theorem in this case is

$$\langle p\rangle = m\frac{d}{dt}\langle x\rangle = m\frac{d}{dt}\left[\sqrt{\tfrac{\hbar}{2m\omega}}\,\tfrac{4}{5}\sin\omega t\right] = m\sqrt{\tfrac{\hbar}{2m\omega}}\,\tfrac{4}{5}\left[\omega\cos\omega t\right] = \sqrt{\tfrac{m\omega\hbar}{2}}\,\tfrac{4}{5}\cos\omega t$$

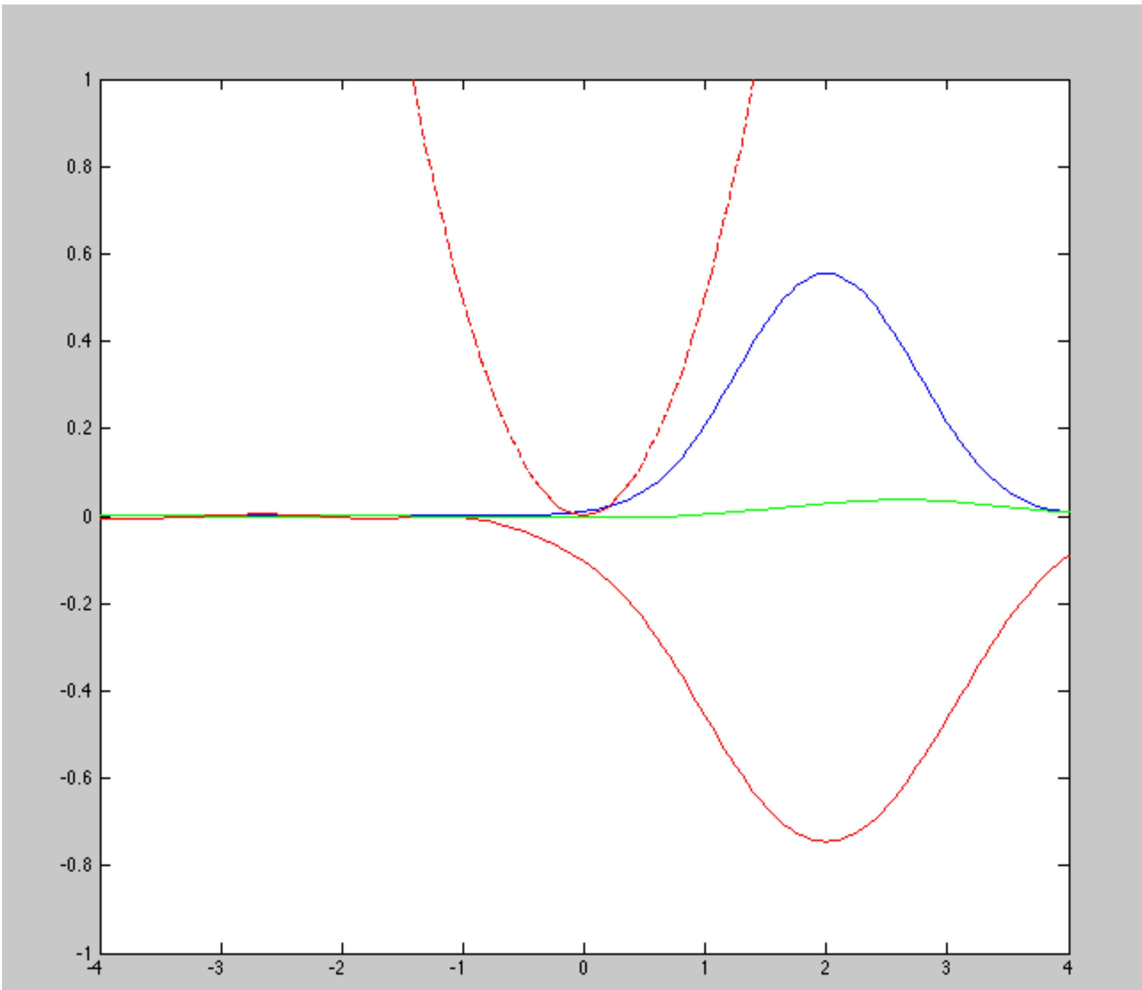
So it is satisfied.

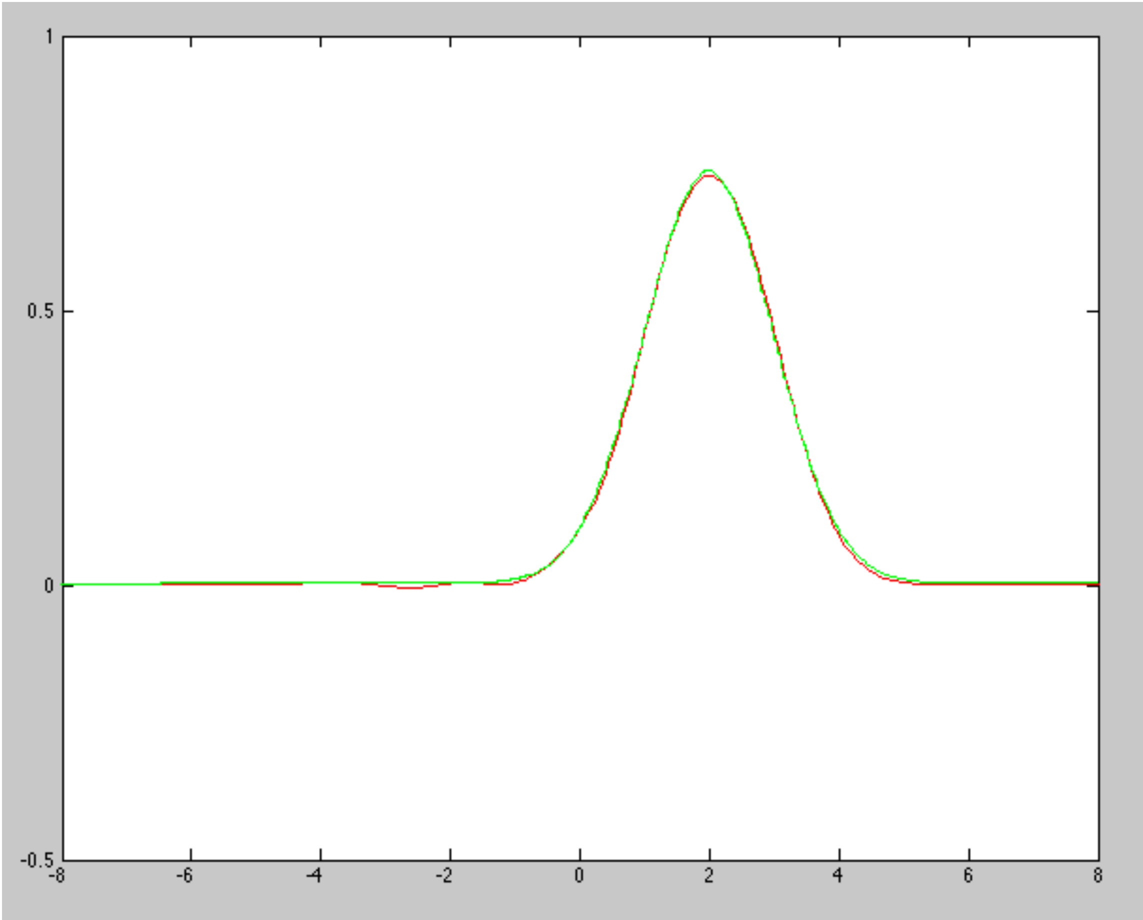
```
%****************************************************************
% Program 9.20
% Feel free to play with this code
%****************************************************************
clear
% Calculate and plot orthogonal polynomials
% PARAMETERS:
Order=25;            %number of energy eigenfunctions used (as order
```

```
                    %increases the approximation improves!)
Xoffset=5;          %amount of offset
dx=0.05;            %spacing of interval
L=8;                %size of region we are interested in
X=-L:dx:L ;         %support on x axis - fills X vector from -L to L
Period = 5;         %for later on in code - how many period to look at
TimeInterval = 0.1;    %Steps for animation of plots later...
number_points=size(X,2);     %how many points used on x-axis?  "size" fcn

Phi_x_0_offset=(pi^(-0.25))*(exp(-(((X-Xoffset).^2)/2)));  % Build offset gnd state

for n=0:Order;

Phi_x_n(n+1,:)=(pi^(-0.25))*(((2.^n)*factorial(n))^(-0.5))*hermite(n,X).*(exp(-
((X.^2)/2)));

C(n+1)=dx*Phi_x_n(n+1,:)*(Phi_x_0_offset)';  %Even though strange looking, this is
                                    %the integral that finds the c_n
                                    %coefficients of the overlap
                                    %integral.  (It is an inner product!) Result is a
                                    %row vector with c_n values.
Check_C_n(n+1)=(Xoffset^n/(2^n*factorial(n))^(0.5))*exp(-(Xoffset^2/4));

end
Check=C
Check2=Check_C_n
Sumcheck=(C*C');
%Checks the squared sums of the Cn's to see how close they come to
%saturating the normalization condition.  Should be close to 1 if we are
%good.
ApproximationFunction=Phi_x_n'*C';
%This is simple, elegant and correct.  The Matrix Phi_x_n' operates on the
%column vector C' via matrix multiplication to produce our approximation
%function.

%ApproximationFunction=sum((C'*ones(1,number_points)).*Phi_x_n);

%The above is a very short bit of code that multiplies the c_n coefficients by
%the eigenfunctions to produce our approximation of the wavefunction.  It does the
same thing
%as the for loop below.  Working inside out,it does it by producing column matrix of
%coefficients, and taking outer product of it with matrix of ones of the same size as
%support X.  This produces a matrix of n rows by size of X columns matrix. Element
by
%element muliplication (.*) with the the Phi_x_n produces a matrix of legendre_lth
```

```
%order*c_n. Sum just sums up columns to produce full approximation function,
NewF2.


%Code below does the same thing as the single line above, but in more pedestrian
way

%ApproximationFunction=0*X;
%for i=1:7;
%ApproximationFunction = ApproximationFunction + Funct2(i,:)*C(i);
%end

%Calculation of the Expectation Value of the Energy

EnergySpectrum=0.5:1:(Order+0.5);
%Creates an energy spectrum vector with energies in units of h_bar*omega

Expected_Energy=(C.^2)*EnergySpectrum'
%Uses the energy spectrum vector to calculate the expectation value of the
%energy -->  Sum((c_n^2)*E_N)

%Calcuation of the time dependent state
for T=0:TimeInterval:2*pi*Period
  %Time is T. Period is set up above and is how many period

  Factor=C.*(exp(-1i*EnergySpectrum*T));
  %Factor is c_n*exp(-i*omega*t) = c_n*exp(-i*E_n*t/h_bar)

  Approx_Phi_x_t=Phi_x_n'*Factor';
  %This is the time dependent approximation function at time = T
  ProbDensity=Approx_Phi_x_t.*conj(Approx_Phi_x_t);
  %Now plot the functions and the envelope
  figure(1)
  %plot(X,ProbDensity,'b',X,0.5*X.*X,'--r');

plot(X,ProbDensity,X,real(Approx_Phi_x_t),'r',X,imag(Approx_Phi_x_t),'g',X,0.5*X.*X,'-
-r');

  axis([-Xoffset-2,Xoffset+2,-1,1]);
  %slows things down so we can see the animation
  pause(0.1);
  %

end

%Propogator_with_Time=Propogator*exp(t)
```

```
%TimeDepApproxFcn=sum((C'*ones(1,number_points)).*Phi_x_n)

figure(2)
plot(X,ApproximationFunction,'r',X,Phi_x_0_offset,'--g');
axis([min(X),max(X),-0.5,1]);
shg
```






```
>> Plot9_20b

Check =

    0.3679    0.5203    0.5203    0.4248    0.3004    0.1900    0.1097    0.0586    0.0293


Check2 =

    0.3679    0.5203    0.5203    0.4248    0.3004    0.1900    0.1097    0.0586    0.0293


Expected_Energy =

    2.4977
```