Michael Cowan

Computational Physics

Problem Set 7

1. In this problem we are given 9 data points and asked to calculate the least square linear fit using the procedure outlined in class. I followed the procedure exactly, and had the program both calculate the linear fit and plot it along with the given data points.

Code:

1. clear all;
2. n=input('Enter the number of data points to be fit: ');
3. %enter the data points
4. for i=1:n
5. A=input('Enter a data point [xi,yi]: ');
6. x(i)=A(1);
7. y(i)=A(2);
8. end
9. %calculate the matrix elements
10. gamma=sum(x.^2);
11. delta=sum(x);
12. beta=sum(x.\*y);
13. ksi=sum(y);
14. M=[gamma,delta;delta,n];
15. c=[beta;ksi];
16. r=M^-1\*c; %calculate the slope and y-intercept
17. disp('The slope of the least square linear fit is:');
18. disp(r(1));
19. disp('The y-intercept of the least square linear fit is:');
20. disp(r(2));
21. %construct the linear fit from the endpoints
22. xg=[0,x(n)];
23. yg=[r(2),r(1)\*x(n)+r(2)];
24. %plot the data points and the linear fit
25. figure(1); clf;
26. plot(x,y,'o',xg,yg,'--');
27. xlabel('x');
28. ylabel('y');
29. title('Data points (x,y) with least square linear fit');

Results:

Enter the number of data points to be fit: 9

Enter a data point [xi,yi]: [1,1]

Enter a data point [xi,yi]: [2,1]

Enter a data point [xi,yi]: [3,3.5]

Enter a data point [xi,yi]: [4,5]

Enter a data point [xi,yi]: [5,4]

Enter a data point [xi,yi]: [6,8]

Enter a data point [xi,yi]: [7,4]

Enter a data point [xi,yi]: [8,7]

Enter a data point [xi,yi]: [9,12]

The slope of the least square linear fit is:

1.1000

The y-intercept of the least square linear fit is:

-0.44444

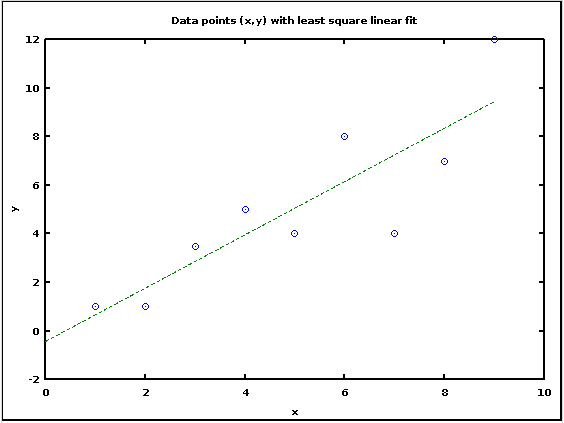


Figure 1: The given data points with the least square linear fit

2. Garcia Chapter 5 Problem 12

In this problem we are given data for the CO2 concentrations at two measuring stations and asked to find the least squares linear fit for each. The process is identical to that used in the previous problem, except that the data was imported from a text file.

The problem also asks us to find when the CO2 concentration will be 10% above its 1981 value for both locations. The question is ambiguously worded, so I interpreted it to be asking when the CO2 concentration will first be 10% above the maximum value reached in 1981.

Code:

1. clear all;
2. %import the data from a text file
3. ma=dlmread('Mauna.txt');
4. ba=dlmread('Barrow.txt');
5. %transpose the data into a row matrix
6. ma=ma';
7. ba=ba';
8. %find the number of data points
9. n1=size(ma,2);
10. n2=size(ba,2);
11. %create a vector with the dates
12. x1 = [0:14/365:229\*14/365] +1981;
13. x2 = [0:14/365:229\*14/365] +1981;
14. %calculate the matrix elements
15. gamma1=sum(x1.^2);
16. delta1=sum(x1);
17. beta1=sum(x1.\*ma);
18. ksi1=sum(ma);
19. gamma2=sum(x2.^2);
20. delta2=sum(x2);
21. beta2=sum(x2.\*ba);
22. ksi2=sum(ba);
23. M1=[gamma1,delta1;delta1,n1];
24. M2=[gamma2,delta2;delta2,n2];
25. c1=[beta1;ksi1];
26. c2=[beta2;ksi2];
27. r1=M1^-1\*c1; %calculate the slope and y-intercept
28. r2=M2^-1\*c2; %calculate the slope and y-intercept
29. disp('The annual rate of increase of CO2 in ppm at Mauna Loa is:');
30. disp(r1(1));
31. disp('The annual rate of increase of CO2 in ppm at Barrow, AK is:');
32. disp(r2(1));
33. %calculate the linear fit
34. xg1=[x1(1),x1(n1)];
35. yg1=[r1(1)\*x1(1)+r1(2),r1(1)\*x1(n1)+r1(2)];
36. figure(1); clf;
37. plot(x1,ma,'+',xg1,yg1,'--');
38. xlabel('Year');
39. ylabel('CO2 (ppm)');
40. title('Carbon Dioxide concentration at Mauna Loa');
41. %calculate the linear fit
42. xg2=[x2(1),x2(n2)];
43. yg2=[r2(1)\*x2(1)+r2(2),r2(1)\*x2(n2)+r2(2)];
44. figure(2); clf;
45. plot(x2,ba,'+',xg2,yg2,'--');
46. xlabel('Year');
47. ylabel('CO2 (ppm)');
48. title('Carbon Dioxide concentration at Barrow,AK');
49. %calculating when the CO2 concentration increases by 10 percent from the 1981 level
50. y1=max(ma(1:26))\*1.1;
51. y2=max(ba(1:26))\*1.1;
52. x1\_10=(y1-r1(2))/r1(1);
53. x2\_10=(y2-r2(2))/r2(1);
54. fprintf('The CO2 level at Mauna Loa will be 10 percent above its 1981 value in %g \n',floor(x1\_10));
55. fprintf('The CO2 level at Barrow,AK will be 10 percent above its 1981 value in %g \n',floor(x2\_10));

Results:

The annual rate of increase of CO2 in ppm at Mauna Loa is:

1.5470

The annual rate of increase of CO2 in ppm at Barrow, AK is:

1.5475

The CO2 level at Mauna Loa will be 10 percent above its 1981 value in 2006

The CO2 level at Barrow,AK will be 10 percent above its 1981 value in 2007

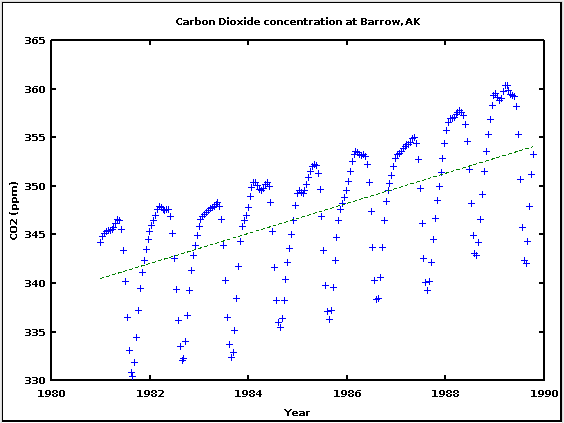
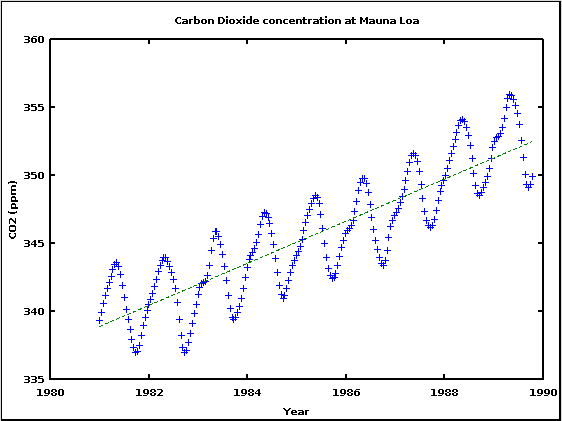


Figure 2: CO2 concentration measured at Mauna Loa (left) and Barrow, AK (right)

I know that the year I calculated for when the CO2 level rises by 10% is not very accurate because I used the linear fit of all of the data to calculate it. Rather than telling us when the value first exceeds 110% of the 1981 maximum value, it really tells us when the average value for the year will exceed 110% of the 1981 maximum value. A better way to calculate this would have been to do a linear fit for the maximum CO2 values each year.

3. In this problem we solve the Schrödinger Equation for the Quantum Simple Harmonic Oscillator using the explicit matrix method derived in class. We do breaking the space up into N intervals of width a. We then construct the Hamiltonian matrix with the term 2α+Vn along the diagonal and –α to the immediate left and right of the diagonal, with Vn being the potential at the nth interval and α=-ħ2/2ma2. We can then find the energy from the eigenvalues of the Hamiltonian matrix and the wavefunctions from the eigenvectors. This method is particularly useful because you do not have to assume the basis states in order to find the energy levels of the system.

Code:

1. clear all;
2. %set parameters
3. hbar=1;
4. m=1;
5. omega=1;
6. L=input('Input the width over which to calculate: ');
7. N=input('Input the number of steps: ');
8. a=L/N;
9. alpha=hbar^2/(2\*m\*a^2);
10. x=(-L/2):a:(L/2);
11. %construct the Hamiltonian matrix
12. for i=1:N+1
13. for j=1:N+1
14. if i==j
15. V(j)=0.5\*m\*omega^2\*x(j)^2;
16. H(i,j)=2\*alpha+V(j);
17. elseif ((j==i-1 && i-1~=0) || (j==i+1 && i+1~=N+2))
18. H(i,j)=-alpha;
19. else
20. H(i,j)=0;
21. end
22. end
23. end
24. [psi,E]=eig(H); %find the eigenvectors and eigenvalues
25. fprintf('The energy of the ground state is %g \n',E(1,1));
26. fprintf('The energy of the first excited state is %g \n',E(2,2));
27. figure(1); clf;
28. plot(x,psi(:,1));
29. xlabel('Position');
30. ylabel('\Psi');
31. title('Ground state wave function of the QSHO potential');
32. figure(2); clf;
33. plot(x,psi(:,2));
34. xlabel('Position');
35. ylabel('\Psi');
36. title('First excited state wave function of the QSHO potential');

Results:

Input the width over which to calculate: 50

Input the number of steps: 500

The energy of the ground state is 0.499687

The energy of the first excited state is 1.49844

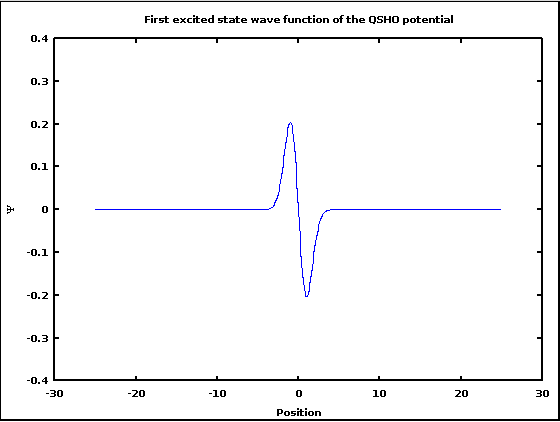
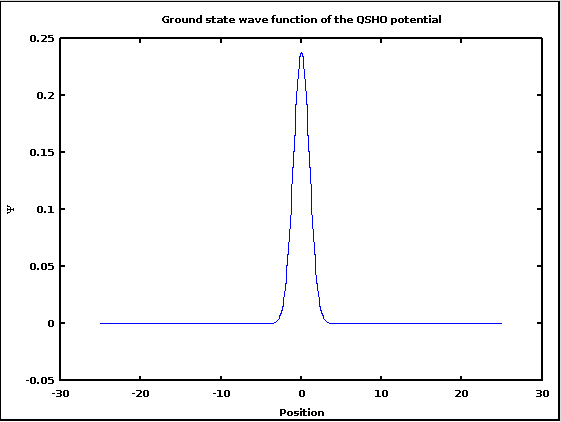


Figure 3: The wave functions for the ground state and the first excited state

The wave functions have the appropriate shape (I looked it up in Griffiths). The energy for the QSHO is given by (n+1/2)ħω, so with ħ=ω=1, the energies for the first two states should be 0.5 and 1.5. As you can see above, my program successfully calculated the ground state energy to within ~0.06% and the first excited state energy to within ~0.1%.

One oddity that I encountered while doing the calculation is that if I made L, the width over which I calculated the wavefunctions, too small, the wavefunctions ended up being inverted from what was expected. I’m not sure what causes this, as the calculated energy levels were still close to their correct values. I suspect it has something to do with the wavefunctions not being calculated far enough into the classically forbidden region.