Math 4513

Solutions to Homework 7

1. Write a Maple program that finds the optimal set of $n+1$ nodes on a specified interval $[a, b]$. (Write it in such a way that it takes the numbers n, a, and b as input parameters.)

```
#values for n , a, b must already be defined to run this routine

# the function maps the interval [-1,1] linearly onto [a,b]
f := t -> (b+a)/2 + (b-a)*t/2:
# determine the n+1 roots of the (n+1)th Chebyshev polynomial
for i from 0 to n do
   t[i] := cos((2*i+1)*Pi/(2*n+2)));
od:
# map the roots t[i] to the interval [a,b]
for i from 0 to n do
   x[i] := f(t[i]);
od:
```

2. Write a Maple program that finds the Newton form of the interpolation polynomial of degree n corresponding to a set consisting of $n+1$ data points. (Write in such a way that it takes the number n and two arrays x[i] and y[i] as input parameters.)

```
# the number n and the arrays x[i], y[i], i=0..n, must be predefined
#
F := array[0..n,0..n]:

#initialize array F[i,0]
for i from 0 to n do
   F[i,0] := Y[i]:
   od:
#apply recursion relations
for j from 1 to n do
   for i from 0 to n-j do
         F[i,j] := (F[i+1,j-1] - F[i,j-1])/(X[i+j] - X[i]):
   od:
od:
#construct Newton form of interpolation polynomial
p := F[0,0]:  #  = Y[0]:
g := 1:
for i from 1 to n do
      g := (x-X[i-1])*g:
      p := p + F[0,i]*g:
od:

print(p);
```

3. Write a Maple program that converts the Newton form of an interpolation polynomial to a polynomial in standard form:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

```
# if p is the Newton form of the interpolation polynomial then
# collect(p,x) will the polynomial expanded in powers of x)
P := collect(p,x);
```

4. In this last problem you are to use the tools you've developed above to find a polynomial that best fits a set of experimental data. Go to

`http://www.math.okstate.edu/~binegar/4513-F98/Interpolation_Problem.html`

There you will find a program that takes as input a set of nodes x_i and produces, as output, a corresponding table of *experimental data*. Pick an appropriate choice of nodes, determine the corresponding experimental data, and find a polynomial (in standard form) that best fits your data.

The following data table was obtained from the WWW experiment.

x	f(x)
-5	15258786
-4	1864131
-3	132854
-2	4083
-1	66
0	11
1	6
2	459
3	33218
4	671091
5	6781686

Here is the program we'll use to find the corresponding interpolation polynomial.

```
n := 10:
X := array(0..n,[-1.0,-0.8,-0.6,-0.4,-0.2,0.0,0.2,0.4,0.6,0.8,1.0]):
Y := array(0..n);
Y[0] := 66.0:
Y[1] := 36.7179869184:
Y[2] := 23.7636048896:
Y[3] := 17.2222688256:
Y[4] := 13.4375000064:
Y[5] := 11.0:
Y[6] := 9.3055555584:
Y[7] := 8.0612330496:
Y[8] := 7.1102253056:
Y[9] := 6.3792344064:
Y[10] := 6.0:

F := array[0..n,0..n]:


#initialize array F[i,0]
for i from 0 to n do
   F[i,0] := Y[i]:
   od:
```

```
#apply recursion relations
for j from 1 to n do
   for i from 0 to n-j do
         F[i,j] := (F[i+1,j-1] - F[i,j-1])/(X[i+j] - X[i]):
            od:
            od:
#construct Newton form of interpolation polynomial
p := F[0,0]:  #  = Y[0]:
g := 1:
for i from 1 to n do
   g := (x-X[i-1])*g:
   p := p + F[0,i]*g:
od:
#write interpolation polynomial in standard form
P := collect(p,x):

#print result
for i from 0 to n do
c := coeff(P,x,i):
c := evalf(c,3):
lprint(`coefficient of x to the`,i,`is`,c):
od:
```

This program produces the following output.

```
coefficient of x to the   0   is   11.0
coefficient of x to the   1   is   -10.0
coefficient of x to the   2   is   9.00
coefficient of x to the   3   is   -8.00
coefficient of x to the   4   is   7.00
coefficient of x to the   5   is   -6.00
coefficient of x to the   6   is   5.00
coefficient of x to the   7   is   -4.00
coefficient of x to the   8   is   3.00
coefficient of x to the   9   is   -2.00
coefficient of x to the   10   is   1.00
```

Thus,

$$P(x) = x^{10} - 2x^9 + 3x^8 - 4x^7 + 5x^6 - 6x^5 + 7x^4 - 8x^3 + 9x^2 - 10x + 11$$