MA 322: Lab Assignment #5

Due on Sunday, September 6, 2015

Jiten Chandra Kalita

Silvi Pandey-130123045

Contents

PROBLEM 1

```
#include<iostream>
#include<math.h>
#include<stdlib.h>
#include<vector>
#include<stdio.h>

using namespace std;
double function(double val);
void GetCoefficients(vector<double>&coefficients,int interval,double a
                     ,double b,double h);
void GetDiff(vector<double>coefficients,int interval,double a,double b
             ,double h,int interval1,double a1,double b1,double h1);
double polynomial(double x,vector<double> coefficients,int interval,
                  double a,double b,double h);
int main()
{
    double a,b,h,a1,b1,h1;
    int interval,interval1;
    a=0;b=6;interval=11;
    h=(double)(b-a)/interval;
    vector<double>coefficients;
    GetCoefficients(coefficients,interval,a,b,h);
    for(int i=0;i<interval;i++)
       printf("a%d %0.6lf\n",i,coefficients[i]);
    cout<<endl;
    a1=0;b1=8;interval1=33;
    h1=(double)(b1-a1)/interval1;
    GetDiff(coefficients,interval,a,b,h,interval1,a1,b1,h1);
}
double polynomial(double x,vector<double>coefficients,int interval,
                  double a,double b,double h)
{
    double sum=0;
    double prod=1;
    for(int i=0;i<interval;i++)
    {
        prod=1;
        for(int j=0;j<i;j++)
            prod=prod*(x-(a+j*h));
        sum+=coefficients[i]*prod;
    }
    return sum;
}
void GetDiff(vector<double>coefficients,int interval,double a,double b
             ,double h,int interval1,double a1,double b1,double h1)
{
    for(int i=0;i<interval1;i++)
    {
        printf("For %0.6lf : %0.6lf\n",a1+i*h1,fabs(function(a1+i*h1)-
                    polynomial(a1+i*h1,coefficients,interval,a,b,h)));
    }
```

```
}
double function(double x)
{
    return atan(x);
}
void GetCoefficients(vector<double>&coefficients,int interval,double a,
                     double b,double h)
{
    double dp[interval][interval];
    for(int i=0;i<interval;i++)
        dp[i][i]=function(a+(double)(i*h));
    int j;
    for(int l=1;l<interval;l++)
    {
        for(int i=0;i<interval;i++)
        {
            j=i+l;
            if(j>=interval)
                break;
            dp[i][i+l]=(dp[i+1][i+l]-dp[i][i+l-1])/((a+(i+l)*h)-(a+i*h));

        }
    }
    for(int i=0;i<interval;i++)
        coefficients.push_back(dp[0][i]);
}
```

OUTPUT

(a)COEFFICIENTS

```
a0 0.000000
a1 0.900699
a2 -0.284418
a3 0.043813
a4 0.005107
a5 -0.005625
a6 0.002085
a7 -0.000546
a8 0.000115
a9 -0.000020
a10 0.000003


Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

(b)mod(F(X)-P(X))

```
For 0.000000 : Error = 0.000000
For 0.250000 : Error = 0.000676
For 0.500000 : Error = 0.000252
For 0.750000 : Error = 0.000208
For 1.000000 : Error = 0.000146
For 1.250000 : Error = 0.000026
For 1.500000 : Error = 0.000062
For 1.750000 : Error = 0.000011
For 2.000000 : Error = 0.000025
For 2.250000 : Error = 0.000015
For 2.500000 : Error = 0.000009
For 2.750000 : Error = 0.000015
For 3.000000 : Error = 0.000000
For 3.250000 : Error = 0.000013
For 3.500000 : Error = 0.000007
For 3.750000 : Error = 0.000011
For 4.000000 : Error = 0.000016
For 4.250000 : Error = 0.000006
For 4.500000 : Error = 0.000033
For 4.750000 : Error = 0.000013
For 5.000000 : Error = 0.000070
For 5.250000 : Error = 0.000104
For 5.500000 : Error = 0.000149
For 5.750000 : Error = 0.000678
For 6.000000 : Error = 0.000000
For 6.250000 : Error = 0.007717
For 6.500000 : Error = 0.038345
For 6.750000 : Error = 0.127970
For 7.000000 : Error = 0.349808
For 7.250000 : Error = 0.840848
For 7.500000 : Error = 1.842061
For 7.750000 : Error = 3.757016
For 8.000000 : Error = 7.234915

Process returned 0 (0x0)   execution time : 0.079 s
Press any key to continue.
```

CONCLUSION

(a)In the interval [0,6] the absolute value of the difference between the function value and the value of the polynomial is very small(of the order of 0.0001)
(b)In the interval (6,8] the error term is quite high(of the order of 1)
(c)The polynomial interpolates quite accurately but doesnot extrapolate well.

PROBLEM 2

```
#include<iostream>
#include<math.h>
#include<stdlib.h>
#include<stdio.h>

using namespace std;
int main()
{
     double TVal[7];
     double alpha=2;
     double delx=0.125;
     double a[7],b[7],c[7],f[7];
     b[0]=-(2+2*alpha*alpha*delx);
     for(int i=0;i<7;i++)
     {
          a[i]=1;
          b[i]=b[0];
          c[i]=1;
          f[i]=0;
     }
     f[6]=-100;
     for(int i=1;i<7;i++)
     {
          b[i]=b[i]-(c[i-1]*a[i])/b[i-1];
          f[i]=f[i]-(f[i-1]*a[i])/b[i-1];
     }
     TVal[6]=f[6]/b[6];
     for(int i=5;i>=0;i--)
     {
          TVal[i]=(f[i]-c[i]*TVal[i+1])/b[i];
     }
     for(int i=0;i<7;i++)
     {
          printf("T%d %0.6lf\n",i+1,TVal[i]);
     }
}
```

OUTPUT

(a)For Δ x = 0.125

```
T1 0.101317
T2 0.303951
T3 0.810537
T4 2.127660
T5 5.572442
T6 14.589666
T7 38.196555

Process returned 0 (0x0)   execution time : 0.070 s
Press any key to continue.
```

(a)For Δ x = 0.0625

```
T1 0.585946
T2 1.464866
T3 3.076219
T4 6.225681
T5 12.487984
T6 24.994278
T7 49.997711

Process returned 0 (0x0)   execution time : 0.020 s
Press any key to continue.
```

PROBLEM 3(a)

```
#include<iostream>
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

using namespace std;
int low=-5;
int high=5;
int points=41;
double h=(double)(high-low)/(points-1);

double function(double x);
void ThomasMethod(double c[],double a[]);
double poly(double x,int j,double a,double b,double c,double d);

int main()
{
     double coefficient[points-1][4];
     double a[points],c[points-2];
     for(int i=0;i<points;i++)
          a[i]=function(low+i*h);
     ThomasMethod(c,a);
     for(int i=0;i<points-1;i++)
     {
          coefficient[i][0]=a[i];
          if(i==0)
               coefficient[i][2]=0;
          else
               coefficient[i][2]=c[i-1];
     }
     for(int i=0;i<points-1;i++)
     {
          if(i!=0)
               coefficient[i][1]=(a[i+1]-a[i])/h-h*(2*c[i-1]+c[i])/3;
          else
               coefficient[i][1]=(a[i+1]-a[i])/h-h*c[i-1]/3;
     }
     for(int i=0;i<points-1;i++)
     {
```

```
          if(i==points-2)
               coefficient[i][3]=-coefficient[i][2]/(3*h);
          coefficient[i][3]=(coefficient[i+1][2]-coefficient[i][2])/(3*h);
     }
    for(int k=0; k<points-1; k++)
        cout<<"$: "<<k+1<<" "<<coefficient[k][0]<<" "<<coefficient[k][1]<<" "
        <<coefficient[k][2]<<" "<<coefficient[k][3]<<endl;
     int qlow=0;
     int qhigh=5;
     int qpoints=101;
     double qh=(double)(qhigh-qlow)/(qpoints-1);
     int j=20;
     for(int i=0;i<101;i++)
     {
          if(qlow+i*qh>low+j*h)
               j++;
          cout<<poly(qlow+i*qh,j,coefficient[j][0],coefficient[j][0],coefficient[j][0]
             ,coefficient[j][0])-function(qlow+i*qh)<<endl;
     }
}
double poly(double x,int j,double a,double b,double c,double d)
{
     return a+b*(x-(low+j*h))+c*(x-(low+j*h))*(x-(low+j*h))+d*(x-(low+j*h))*
     (x-(low+j*h))*(x-(low+j*h));
}
double function(double x)
{
     return 1/(1+x*x);
}
void ThomasMethod(double c[],double a[])
{
     double    A[points-2],B[points-2],C[points-2],F[points-2];
     for(int i=0;i<points-2;i++)
     {
          A[i]=h;
          B[i]=4*h;
          C[i]=h;
          F[i]=(3*(a[i+2]-a[i+1])-3*(a[i+1]-a[i]))/h;
     }
     for(int i=1;i<points-2;i++)
     {
          B[i]=B[i]-C[i]*A[i]/B[i-1];
          F[i]=F[i]-F[i-1]*A[i]/B[i-1];
     }
     c[points-3]=F[points-3]/B[points-3];
     for(int i=points-4;i>=0;i--)
     {
          c[i]=(F[i]-C[i]*c[i+1])/B[i];
     }
}
```

OUTPUT

```
For 0 0
For 0.05 -0.214447
For 0.1 -0.172099
For 0.15 -0.122466
For 0.2 -0.0651855
For 0.25 -0.303676
For 0.3 -0.251831
For 0.35 -0.195569
For 0.4 -0.134869
For 0.45 -0.0697008
For 0.5 -0.29
For 0.55 -0.235274
For 0.6 -0.179054
For 0.65 -0.121228
For 0.7 -0.0616209
For 0.75 -0.241563
For 0.8 -0.193756
For 0.85 -0.145989
For 0.9 -0.0979862
For 0.95 -0.0494367
For 1 -0.189024
For 1.05 -0.150941
For 1.1 -0.113318
For 1.15 -0.0758388
For 1.2 -0.0381775
For 1.25 -0.145052
For 1.3 -0.115747
For 1.35 -0.0868728
For 1.4 -0.0581455
For 1.45 -0.0292822
For 1.5 -0.111538
For 1.55 -0.0891015
For 1.6 -0.0669604
```

```
For 1.65 -0.0448828
For 1.7 -0.0226386
For 1.75 -0.0867788
For 1.8 -0.0694491
For 1.85 -0.0522914
For 1.9 -0.0351197
For 1.95 -0.0177499
For 2 -0.0685567
For 2.05 -0.0549782
For 2.1 -0.0414821
For 2.15 -0.0279187
For 2.2 -0.0141401
For 2.25 -0.0550347
For 2.3 -0.0442239
For 2.35 -0.0334361
For 2.4 -0.0225497
For 2.45 -0.0114441
For 2.5 -0.0448653
For 2.55 -0.036121
For 2.6 -0.0273623
For 2.65 -0.0184888
For 2.7 -0.00940098
For 2.75 -0.0371008
For 2.8 -0.0299222
For 2.85 -0.0227066
For 2.9 -0.0153699
For 2.95 -0.00782872
For 3 -0.0310811
For 3.05 -0.0251071
For 3.1 -0.0190831
For 3.15 -0.0129379
```

```
For 3.2 -0.0066004
For 3.25 -0.026345
For 3.3 -0.0213118
For 3.35 -0.016222
For 3.4 -0.0110141
For 3.45 -0.005627
For 3.5 -0.0225671
For 3.55 -0.0182794
For 3.6 -0.013932
For 3.65 -0.00947167
For 3.7 -0.0048453
For 3.75 -0.019515
For 3.8 -0.0158257
For 3.85 -0.0120761
For 3.9 -0.00821973
For 3.95 -0.00420983
For 4 -0.0170203
For 4.05 -0.0138171
For 4.1 -0.0105548
For 4.15 -0.00719197
For 4.2 -0.00368741
For 4.25 -0.014959
For 4.3 -0.0121554
For 4.35 -0.0092945
For 4.4 -0.00633944
For 4.45 -0.0032535
For 4.5 -0.0132392
For 4.55 -0.0107673
For 4.6 -0.00824041
For 4.65 -0.00562553
For 4.7 -0.00288969
For 4.75 -3.93839e+294
For 4.8 -4.11199e+294
For 4.85 -4.29548e+294
For 4.9 -4.49255e+294
For 4.95 -4.70692e+294
For 5 -3.75083e+294

Process returned 0 (0x0)   execution time : 0.303 s
Press any key to continue.
```

PROBLEM 3(b)

```
#include<iostream>
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

using namespace std;
int low=-5;
int high=5;
int points=11;
double h=(double)(high-low)/(points-1);

double function(double x);
int main()
{
     double dp[points][points];
     for(int i=0;i<points;i++)
          dp[i][i]=function(low+i*h);
     int j;
     for(int l=1;l<points-1;l++)
```

```
20      {
            for(int i=0;i<points;i++)
            {
                j=i+1;
                if(j>=points)
25                  break;
                dp[i][j]=(dp[i+1][j]-dp[i][j-1])/((low+j*h)-(low+i*h));
            }
        }
        for(int i=0;i<points;i++)
30          cout<<dp[0][i]<<" ";
        cout<<endl;
}
double function(double x)
{
35      return 1/(1+x*x);
}
```

OUTPUT

COEFFICIENT

```
Coefficient 1 : 0.0384615
Coefficient 2 : 0.020362
Coefficient 3 : 0.0104072
Coefficient 4 : 0.00633484
Coefficient 5 : 0.00429864
Coefficient 6 : -0.0020362
Coefficient 7 : -0.00113122
Coefficient 8 : 0.00108597
Coefficient 9 : -0.000429864
Coefficient 10 : 0.000113122
Coefficient 11 : 7.06327e-304


Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

PROBLEM 4(a)

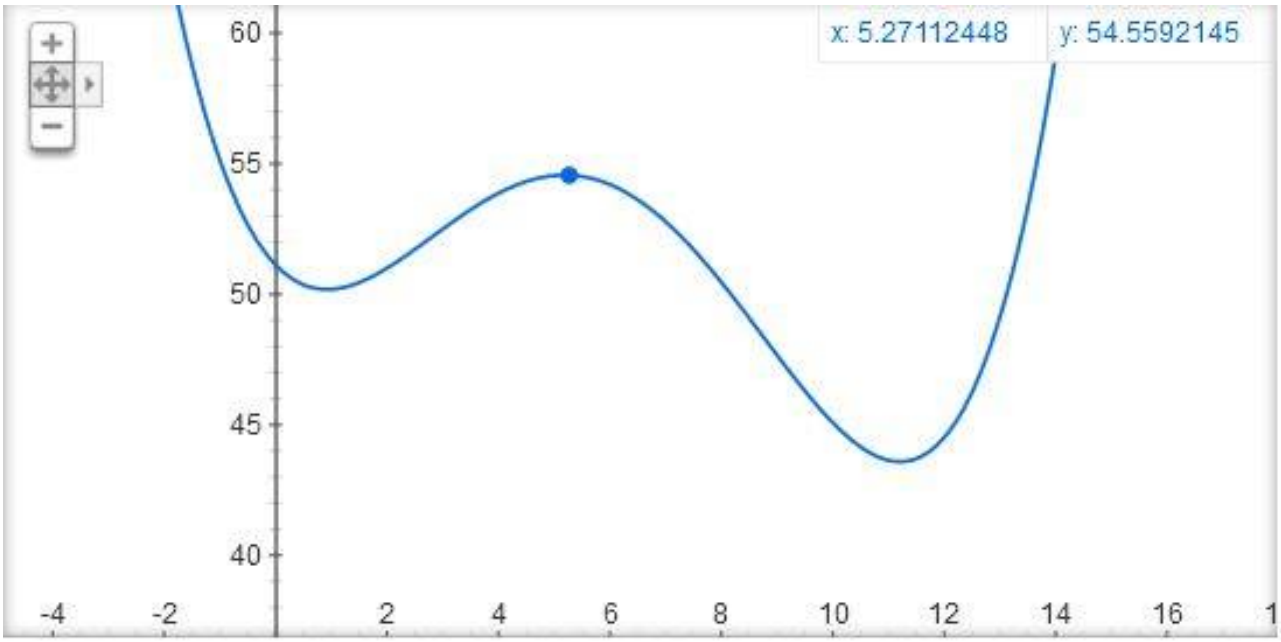
```
#include<iostream>
#include<math.h>
#include<stdlib.h>
#include<stdio.h>
5

using namespace std;

int low=0;
int high=13;
10 int points=5;
double functionSpeed(double t);
double NewtonPoly(double coefficient[],double val,double xVal[]);
```

```
int main()
{
     FILE *fp=fopen("speed.txt","w");
     double dp[points][points];
     double xVal[5];
     xVal[0]=0;xVal[1]=3;xVal[2]=5;xVal[3]=8;xVal[4]=13;
     for(int i=0;i<points;i++)
          dp[i][i]=functionSpeed(xVal[i]);
     int j;
     for(int l=1;l<points-1;l++)
     {
          for(int i=0;i<points;i++)
          {
               j=i+l;
               if(j>=points)
                    break;
               dp[i][j]=(dp[i+1][j]-dp[i][j-1])/(xVal[j]-xVal[i]);
          }
     }
     double coefficient[points];
     for(int i=0;i<points;i++)
          coefficient[i]=dp[0][i];
     for(int i=0;i<points;i++)
          fprintf(fp,"%0.6lf\n",dp[0][i]);
     cout<<NewtonPoly(coefficient,10,xVal);
}
double NewtonPoly(double coefficient[],double x,double xVal[])
{
     double sum=0;
     double prod;
     for(int i=0;i<points;i++)
     {
          prod=1;
          for(int j=0;j<i;j++)
               prod=prod*(x-xVal[j]);
          sum+=prod*coefficient[i];
     }
     return sum;
}
double functionSpeed(double t)
{
     if(t==0)
          return 75;
     if(t==3)
          return 77;
     if(t==5)
          return 80;
     if(t==8)
          return  74;
     if(t==13)
          return 72;
}
```

OUTPUT

(a) Speed at t=10s : 55.4167 feet/sec
(b) Distance travelled : 779.75 feet

PROBLEM 4(b)



(a)As visible in the plot,the car just touches 55 miles/hour speed in the interval t : [0,13]
(b)Predicted maximum Speed of car : 55miles/hour(approx)