1 question

As I was instructed during tutorial, I copied Formula for Lagrange interpolation from the internet.

Double lagrange (double \*x, double \*y, int n, double xx)

{

int i, j;

double yint, ylag;

yint = 0.0;

for (i = 0; i < n; i++)

{

ylag = 1.0;

for (j = 0; j < n; j++)

{

if (i == j)

continue;

ylag \*= (xx - x[j]) / (x[i] - x[j]);

}

yint += y[i] \* ylag;

}

return yint;

}

2 question

I created a new file manually in online C compiler.

3 question

Firstly I opened file from question 2 for reading. Then using code I create file for interpolation results.

FILE \* fp = fopen ("interpolation\_data.ini", "r");

FILE \* f = fopen ("interpolation\_result.dat", "w");

Then I scanned values from interpolation data and computed interpolation results. I used for loop with double I instead of int I before I was told, that it is not generally used. I didn’t change the code because it worked, but later on I used integer parameter and additional double variable. I printed both in console and in file to be able to quickly verify my code.

fscanf (fp, "%d", &n);

double x[n];

double y[n];

for (int i = 0; i <= n; i++)

{

fscanf (fp, "%lf" "%lf", &x[i], &y[i]);

}

for (double i = x[0]; i <= x[n]; i += (x[n] - x[0]) / 100)

{

printf ("\nValue at x = %lf is equal to: %lf", i,

lagrange (x, y, n, i));

fprintf (f, "\n%lf\t" "%lf", i, lagrange (x, y, n, i));

}

4 question

5 question

I created two new arrays x2 and y2 to distinguish them from the ones from previous exercises. I created function Bad which returns value of given formula

double Bad(double x){

return(1.0/(10.0\*x\*x));

}

Then I filled x2 with uniformly distributed nodes between -1 and 1. Then I added corresponding values to y2.

for(int i = 0; i < d; i++){

x2[i] = -1.0 + (i\*(2.0/(d-1)));

y2[i] = Bad(x2[i]);

}

Later I created code to compute interpolate values of g for points from exercise 3. I received, however, strange results which indicates my fault somewhere, but I couldn’t find it. I attach only graph for number of interpolation base points equal to 6 because if I include n = 20 and 40 first graph is nearly vertical line.

int d = 6;

double x2[d];

double y2[d];

double Bad(double x){

return(1.0/(10.0\*x\*x));

}

for(int i = 0; i < d; i++){

x2[i] = -1.0 + (i\*(2.0/(d-1)));

y2[i] = Bad(x2[i]);

}

double point = 0.0;

for(int j = 0; j < d; j++){

fprintf(g, "%lf\t %.3lf\n",point, lagrange(x2, y2, d, point));

point += 0.1;

}

6 question

I examined, copied and modified formula for Newtons interpolation polynominal from the internet.

double Newton(double \*ax, double \*ay, double x, double n){

double h=ax[1]-ax[0];

double diff[MAXN+1][ORDER+1];

double p, yp;

double nr = 1.0;

double dr = 1.0;

for (int i=0;i<=n-1;i++)

diff[i][1] = ay[i+1]-ay[i];

for (int j=2;j<=ORDER;j++)

for(int i=0;i<=n-j;i++)

diff[i][j] = diff[i+1][j-1] - diff[i][j-1];

int i=0;

while (!(ax[i]>x))

i++;

i--;

p = (x-ax[i])/h;

yp = ay[i];

for (int k=1;k<=ORDER;k++)

{

nr \*=p-k+1;

dr \*=k;

yp +=(nr/dr)\*diff[i][k];

}

return yp;

}