СПб НИУ ИТМО

кафедра ИПМ

Вычислительная математика

Лабораторная работа № 3

Вычисление функции с помощью интерполяционного полинома Ньютона

Вариант 1.2

Работу выполнил:

Студент II курса

Группы № 2120

Журавлев Виталий

Санкт-Петербург

2013 г.

**Цель работы:**

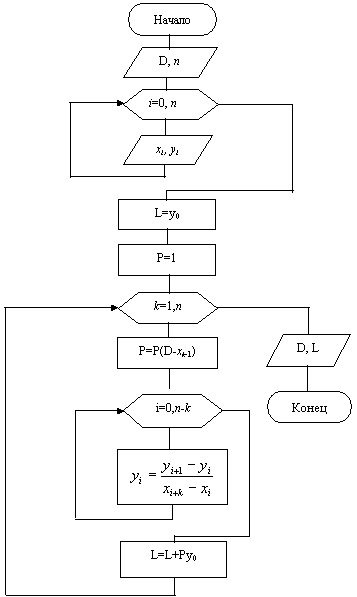
Составить подпрограмму для вычисления значений функции, заданной таблично на решетке X0..XN, с помощью интерполяционного полинома Ньютона заданной степени K.

**Описание метода:**

**Интерполяционный полином в форме Ньютона**

|  |  |  |
| --- | --- | --- |
| Интерполяционный многочлен легко определяется если его построить в виде: | | |
|  | *Pn(x) = С0 + С1(x - x0) + C2(x - x0) (x - x1) + ...+ Cn(x - x0)(x - x1) ... (x - xn-1)* | (5) |
| Исходя из [условия интерполяции](http://www.cde.spbstu.ru/Num_Met/Interpol/interpol_t.html) ( http://www.cde.spbstu.ru/Num_Met/Interpol/Images/Pol_inter2.gif при http://www.cde.spbstu.ru/Num_Met/Interpol/Images/Pol_inter3.gif ) для коэффициентов *Ci* получим систему уравнений треугольного вида | | |
|  | *f(x0) = С0 f(x1) = С0 + С1(x1 - x0) f(x2) = С0 + С1(x1 - x0) + C2(x2 - x0)(x2 - x1)* ***. . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . .*** *f(xn) = С0 + С1(xn - x0) + C2(xn - x0)(xn - x1) + ...+ Cn(xn - x0)(xn - x1) ... (xn- xn-1*) |  |
| Из этой системы легко находятся: | | |
|  | *http://www.cde.spbstu.ru/Num_Met/Interpol/Images/newton_pol2.gif http://www.cde.spbstu.ru/Num_Met/Interpol/Images/newton_pol3.gif http://www.cde.spbstu.ru/Num_Met/Interpol/Images/newton_pol4.gif*и так далее. |  |
| Величины, стоящие в правой части приведённых выше равенств, получили название ***разделённых разностей****,* соответственно, нулевого, первого и второго порядков. Для них приняты обозначения *f[xi],* *f[xi,xi-1], f[xi,xi-1,xi-2]* и т.д. С учётом этих обозначений выражение [(5)](http://www.cde.spbstu.ru/Num_Met/newton_pol.html#y9) можно переписать в виде : | | |
|  | *Pn(x) = f[x0] + f[x1,x0](x - x0) + f[x2,x1,x0](x - x0)(x - x1) + ...                                  + f[xn,xn-1,...x0](x - x0)(x - x1)...(x - xn-1)* | (6) |
| Можно показать, что | | |
|  | http://www.cde.spbstu.ru/Num_Met/Interpol/Images/newton_pol5.gif | (7) |
| Выражения (6) и (7) определяют интерполяционный полином в форме Ньютона. Вычисление полинома в Ньютоновской форме удобно при последовательном дополнении сетки *(n+2)*-м узлом и наращивании порядка интерполяционного полинома. При этом необходимо вычислить лишь одно дополнительное слагаемое  *f[xn+1,xn,...x0](x - x0)(x - x1)...(x - xn)*в выражении (6). | | |

**Алгоритм:**



**Код основных методов:**

…

if (Points != null)

{

coeffX = weight / (2 \* (maxX + maxX / 7));

coeffY = height / (2 \* (maxY + maxY / 7));

float x1, x2, y1, y2;

for (float x = Points[0].X; x <= Points[count - 1].X - 0.1f; x += 0.1f)

{

x1 = coeffX \* x + weight / 2;

x2 = coeffX \* (x + 0.1f) + weight / 2;

//график по предполагаемой функции

y1 = -coeffY \* (float)f(x) + height / 2;

y2 = -coeffY \* (float)f(x + 0.1f) + height / 2;

graph.DrawLine(truegraph, x1, y1, x2, y2);

//график по полиному

y1 = -coeffY \* NewtonF(x) + height / 2;

y2 = -coeffY \* NewtonF(x + 0.1f) + height / 2;

graph.DrawLine(approxgraph, x1, y1, x2, y2);

}

...

}

//нахождение значения функции с помощью полинома

private float NewtonF(float x)

{

float[] masY = new float[count];

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

masY[i] = Points[i].Y;

double summ = masY[0], multiplier = (x - Points[0].X) / (Points[1].X - Points[0].X), q = 1;

int factI = 1;

for (int i = 1; i <= count - 1; i++)

{

q \*= multiplier;

factI \*= i;

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

masY[j] = masY[j + 1] - masY[j];

summ += masY[0] \* q / factI;

multiplier--;

}

**Вывод:**

В ходе выполнения лабораторной работы я изучил принцип интерполяции на примере рассмотренного мной метода вычисления функции с помощью интерполяционного полинома Ньютона.