Міністерство освіти та Науки України

Національний технічний університет “Харківський політехнічний інститут”

Кафедра «Програмна інженерія і інформаційні технології управління»

Індивідуальне завдання №5

з дисципліни

«Чисельні методи»

Виконали:

студенти групи КН-34б

Рузняєв Д.А.

Костюк I.Ю.

Перевірив:

Проф. Гужва В.О.

Харків

2016

**МЕТОД НЬЮТОНА-КОТЕСА**

В выражении http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image020.png мы называли http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image022.png коэффициентами, исходя из их смысла. Однако, правильнее эти величины называть весовыми коэффициентами. Величину http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image007_0004.png, определяющую погрешность численного интегрирования, также называют остатком.

Для семейства методов Ньютона-Котеса можно записать общее выражение:

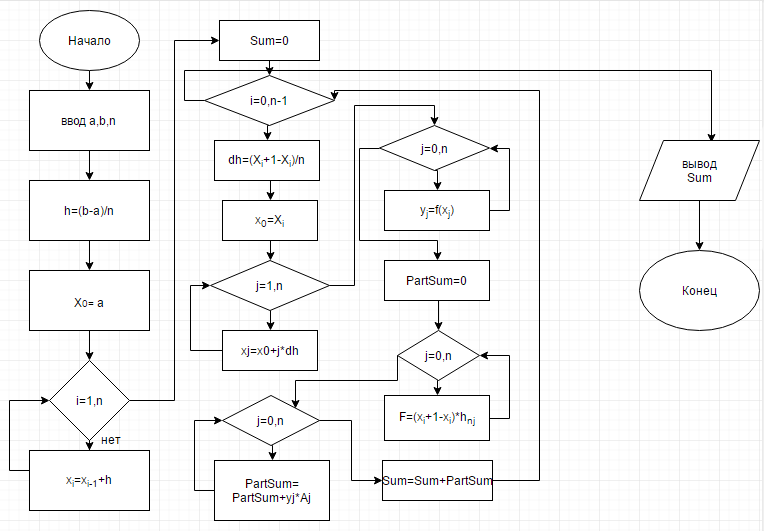
http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image009_0003.png                                                             (2.16)

где n – порядок метода Ньютона-Котеса, N – количество частичных отрезков, http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image011_0003.png, http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image013_0002.png, http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image015_0002.png.

Коэффициенты Hi могут быть заданы в табличной форме:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | H0 | H1 | H2 | H3 | H4 | H5 | H6 |
| 1 | **1/2** | **1/2** |  |  |  |  |  |
| 2 | **1/6** | **2/3** | **1/6** |  |  |  |  |
| 3 | **1/8** | **3/8** | **3/8** | **1/8** |  |  |  |
| 4 | **7/90** | **16/45** | **6/45** | **16/45** | **7/90** |  |  |
| 5 | **19/288** | **75/288** | **50/288** | **50/288** | **75/288** | **19/288** |  |
| 6 | **41/840** | **216/840** | **27/840** | **272/840** | **27/840** | **216/840** | **41/840** |

БЛОК СХЕМА МЕТОДА НЬЮТОНА-КОТЕСА



ИСХОДНЫЙ КОД МЕТОДА НЬЮТОНА-КОТЕСА

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace nk

{

class Program

{

double y(double x)

{

return (3.5\*Math.Pow(x, 2) + x) / (Math.Pow(x, 4) + 1);

}

static void Main(string[] args)

{

Program ex = new Program();

double a = 0;

double b = 1.2;

double h = 2 / 10.0;

Console.WriteLine("h={0}", h);

double[] x = new double[7];

x[0] = 0;

Console.WriteLine("X0={0}", x[0]);

for (int i = 1; i < 7; i++)

{

x[i] = x[0] + i \* h;

Console.WriteLine("x{1}={0}", x[i], i);

}

double result = 0;

double[] Y = new double[7];

Y[0] = ex.y(x[0]);

Console.WriteLine("Y[0]={0}", Math.Round(Y[0], 3));

for (int i = 1; i < Y.Length; i++)

{

Y[i] = ex.y(x[i]);

Console.WriteLine("Y[{0}]={1}", i, Math.Round(Y[i], 3));

}

int[,] koef = new int[,]{{1,0,0,0,0,0,0,0,0,0},

{1,1,0,0,0,0,0,0,0,0},

{1,4,1,0,0,0,0,0,0,0},

{1,3,3,1,0,0,0,0,0,0},

{7,32,12,32,7,0,0,0,0,0},

{19,75,50,50,75,19,0,0,0,0},

{41,216,27,272,27,216,41,0,0,0}};

float[] mltp = new float[] { 1, 1 / 2f, 1 / 3f, 1 / 8f, 1 / 90f, 2 / 288f, 1 / 840f };

double Sum, PartSum;

for (int j = 0; j < 6; j++)

{

PartSum = 0;

for (int i = 0; i <= 7; i++)

PartSum += koef[7, i] \* f(a + (i + j \* 7) \* h);

Sum += mltp[7] \* PartSum \* h;

}

Console.WriteLine(Math.Round(Sum, 3));

}

}

}

РУЧНОЙ ПРОСЧЕТ МЕТОДОМ НЬЮТОНА-КОТЕСА

*Студента Рузняева Д.А*

n=6

, где h = , h = 0.2

Найдем значения F(x) в этих точках

X0=0; Y0=0

X1=0.2; Y1=0.339

X2=0.4; Y1=0.936

X3=0.6; Y2=1.647

X4=0.8; Y3=2.157

X5=1; Y1=2.25

X6=1.2; Y4=2.03

Используя таблицу, получим:

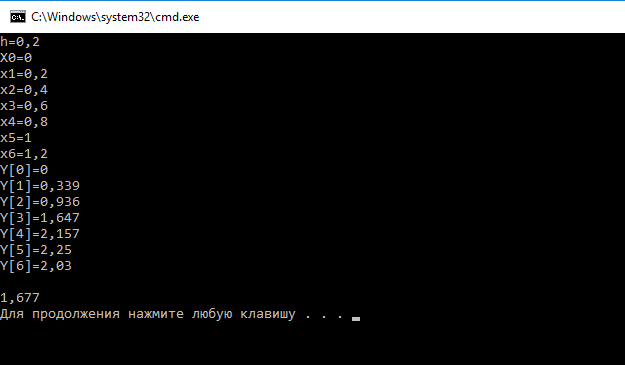
=

+0.0693321429+0.578+0.099)=

=1.677

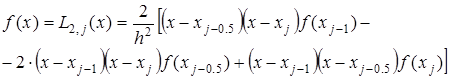
ПРОСЧЕТ С ПОМОЩЬЮ ПРОГРАММЫ МЕТОДОМ НЬЮТОНА-КОТЕСА

*Студента Рузняева Д.А.*

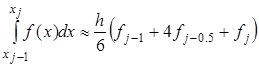


МЕТОД СИМПСОНА

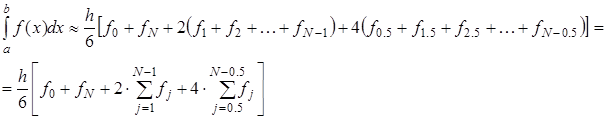
В этом методе подынтегральная функция на частичном отрезке http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image004.png аппроксимируется параболой, проходящей через три точки http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image005_0002.png, http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image007_0003.png, http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image024.png, то есть интерполяционным многочленом Лагранжа второй степени:

                (1)

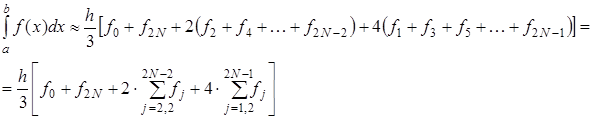
Проведя интегрирование, получим:

                                                      (2)

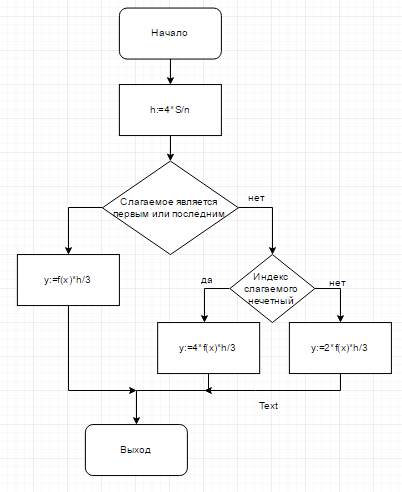
Это и есть формула Симпсона или формула парабол. На отрезке http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image007.png формула Симпсона примет вид:

   (3)

Можно разбить отрезок интегрирования http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image007.png на четное количество 2*N* равных частей с шагом http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image020_0000.png. Тогда можно построить параболу на каждом сдвоенном частичном отрезке http://aco.ifmo.ru/el_books/numerical_methods/lectures/glava2/glava2_clip_image004.png и переписать выражения (2.12-2.14) без дробных индексов. Тогда формула Симпсона примет вид:

     (4)

БЛОК-СХЕМА МЕТОДА СИМПСОНА



ИСХОДНЫЙ КОД МЕТОДА СИМПСОНА

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Simpson

{

class Program

{

double y(double x)

{

return (Math.Pow(x, 2) + 3.8) / (Math.Pow(x, 3) + 7.1);

}

static void Main(string[] args)

{

Program ex = new Program();

double a = 0;

double b = 2.2;

double h = 2.2 / 10.0;

Console.WriteLine("Найдем h={0}", h);

double[] x = new double[5];

x[0] = 0;

Console.WriteLine("X0={0}", x[0]);

for (int i = 1; i < 5; i++)

{

x[i] = x[0] + i \* h;

Console.WriteLine("x{1}={0}", x[i], i);

}

double result = 0;

double[] Y = new double[5];

Y[0] = ex.y(x[0]);

Console.WriteLine("Y[0]=0{0}",Math.Round(Y[0],3));

for (int i = 1; i < Y.Length; i++)

{

Y[i] = ex.y(x[i]);

Console.WriteLine("Y[{0}]={1}", i, Math.Round(Y[i], 3));

}

result += (h / 3) \* (Y[0] + Y[Y.Length - 1]);

Console.Write("h/3\*(Y[0]+Y[{0}]", Y.Length - 1);

for (int i = 1; i < Y.Length; i++)

{

if (i % 2 != 0)

{

result += ((4 \* h) / 3) \* (ex.y(x[i]));

Console.Write("+4\*y({0})", i);

}

if (i % 2 == 0 && i != 4)

{

result += ((2 \* h) / 3) \* (ex.y(x[i]));

Console.WriteLine("+2\*y({0})", 1);

}

}

Console.WriteLine("]=" + Math.Round(result, 3));

}

}

}

РУЧНОЙ ПОДСЧЕТ МЕТОДА СИМПСОНА

2,2

∫ (x^2 + 3,8)/(x^3 +7,1) dx

0

h = от 0 до 2,2 ,следовательно делим на 10 части.

h = 0,22

h/3(y0 + 4(y1+y3) +2(y2 +y4))

f(x) = (x^2 + 3,8)/(x^3 +7,1)

x0 = 0 y0 = 0.535

x1 = 0,22 y1 = 0.541

x2 = 0,44 y2 = 0.556

x3 = 0,66 y3 = 0.573

x4 = 0,88 y4 = 0.588

2,2

∫ (x^2 + 3,8)/(x^3 +7,1) dx = h/3\*(Y[0]+Y[4]+4\*y(1)+2\*y(1)+4\*y(3)]=0.491

0

ПРОСЧЕТ С ПОМОЩЬЮ ПРОГРАММЫ МЕТОДА СИМПСОНА

