

Цель работы: а) освоение методов интерполяции функций; б)

совершенствование навыков по алгоритмизации и программированию вычислительных задач.

Постановка задачи: cоставить схему алгоритма и программу на языке C/C++ решения задачи по теме «Интерполяция» в соответствии с

индивидуальным заданием. Вариант 25

Значения x:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -3.0 | -1.5 | 0.0 | 1.5 | 3.0 |

Значения y:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.4 | 1.2 | 0.8 | 1.7 | 0.5 |

Метод интерполяции: Метод Ньютона

Общая постановка задачи

При решении различных практических задач результаты исследований оформляются в виде таблиц, отображающих зависимость одной или нескольких измеряемых величин от одного определяющего параметра (аргумента). Такого рода таблицы представлены обычно в виде двух или более строк (столбцов) и используются для формирования математических моделей.

Таблично заданные в математических моделях функции обычно записываются в таблицы вида:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Х*** | ***Х0*** | ***Х1*** | ***…*** | ***Хn*** |  |
| ***Y1(X)*** | ***Y(Х0)*** | ***Y(Х1)*** | ***…*** | ***Y(Хn)*** |  |
| ***…*** | ***…*** | ***…*** | ***…*** | ***…*** | **( 1 )** |
| ***Ym(X)*** | ***Y(Х0)*** | ***Y(Х1)*** | ***…*** | ***Y(Хn)*** |  |

Ограниченность информации, представленной такими таблицами, в ряде случаев требует получить значения функций ***Yj(X) (j=1,2,…,m)*** в точках ***Х***, не совпадающих с узловыми точками таблицы ***Хi(i=0,1,2,…,n)***. В таких случаях необходимо определить некоторое аналитическое выражение ***φj(Х)*** для вычисления приближенных значений исследуемой функции ***Yj(X)*** в произвольно задаваемых точках ***Х***. Функция ***φj(Х)*** используемая для определения приближенных значений функции ***Yj(X)*** называется аппроксимирующей функцией (от латинского ***approximo*** - приближаюсь). Близость аппроксимирующей функции ***φj(Х)*** к аппроксимируемой функции ***Yj(X)*** обеспечивается выбором соответствующего алгоритма аппроксимации.

Все дальнейшие рассмотрения и выводы мы будем делать для таблиц, содержащих исходные данные одной исследуемой функции (т. е. для таблиц с ***m=1***).

# Описание метода решения

Для интерполяции функций, заданных таблицами с равноотстоящими значениями аргумента

𝑋𝑖+1−𝑋𝑖=ℎ,где 𝑖=0,1,2…𝑛−1

построение интерполяционных формул и вычисление по формулам заметно упрощается. В записях этих интерполяционных алгоритмов используются разности между значениями функции в соседних узлах интерполяции.

Конечной разностью первого порядка называется



(19)

Из конечных разностей первого порядка образуются конечные разности второго порядка [ 4 ]:



(20)

Аналогично определяются конечные разности третьего, четвёртого и более высоких порядков.

Для функций, заданных таблицами с постоянным шагом изменения аргумента, наиболее часто используются первая или вторая формулы Ньютона, в которых интерполяционная функция определяется как многочлен вида:



(21)

при интерполяции от нулевого узла ***Х0*** или



(22)

при интерполяции от узла ***Хn***.

Значения коэффициентов ***ai*** и ***bi*** в формулах (21) или (22) находятся из условий Лагранжа, определяющих в узлах интерполяции совпадение значений интерполирующей функции со значением таблично заданной функции

***Pn* (*xi* )** ***Yi***

### (см. также формулу (2) в общей постановке задачи интерполяции).

Полагая ***Х=Х0*** , в формуле (21) получим

***Pn(Х0)=a0=Y0 .***

Аналогично для ***Х=Х1***

***Pn(Х1)=a0+a1(X1-X0)=Y1 ,***

### и далее

***a1=(Y1-Y0)/(X1-X0)***

### или, используя введённые обозначения,

***a1= Δy0/(1!h).***

### Продолжая подстановки значений ***Хi ,*** получим

***Pn(Х2)=a0+a1(X2-X0)+ a2 (X2-X0)(X2-X1) =Y2 ,***

### и далее

откуда

***a2\*2h2=Y2 - a0 - a1\*2h* = *Y2 - Y0 - Δy0/h\*2h* = *Y2 - 2 Y1 + Y0 = Δ2y0***

**2 *y***

***a***  **0 .**

**2 2!*h*2**

Проведя аналогичные преобразования для ***Х=Х3*** и ***Х=Х4***, получим

(24)

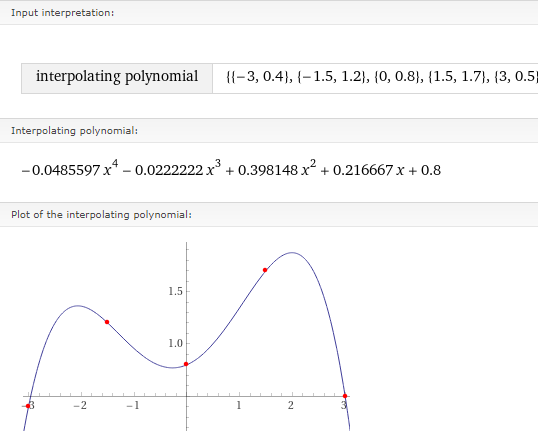


Подставив (24) в (21), получим

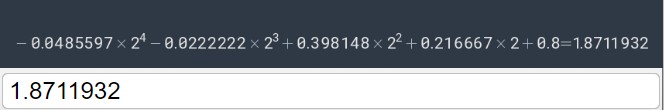


(25)

### Аналитические расчеты



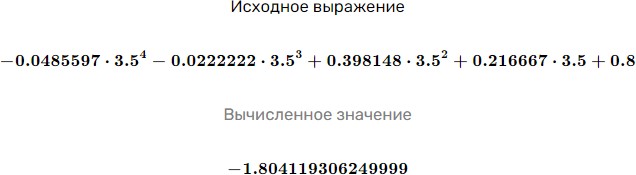
X = 2



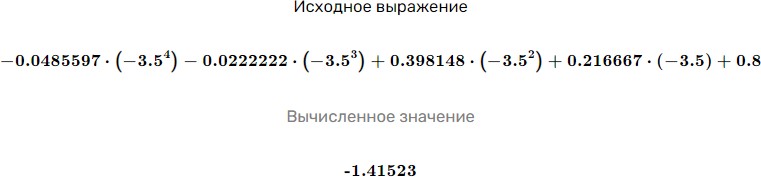
### X = -2.4



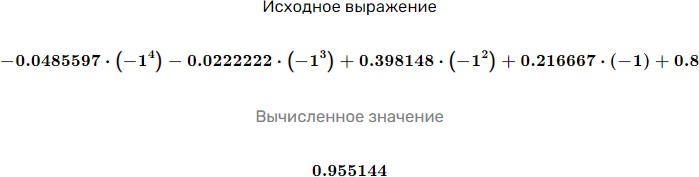
X = 3.5



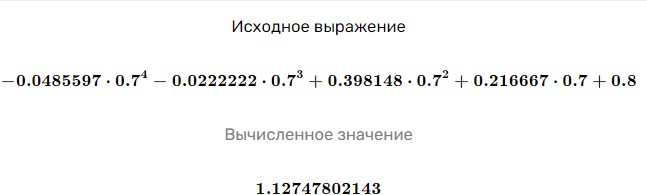
### X = -3.5



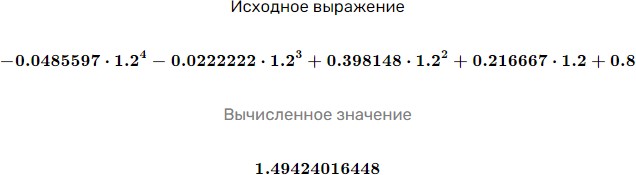
X = -1

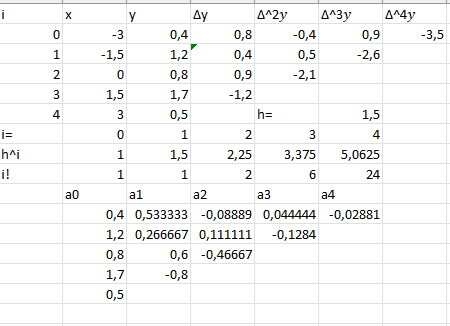


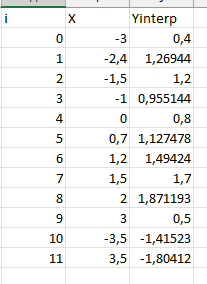
### X = 0.7



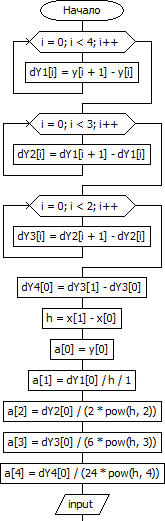
X = 1.2

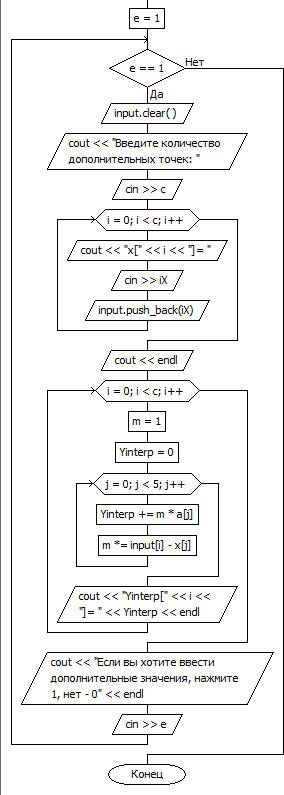




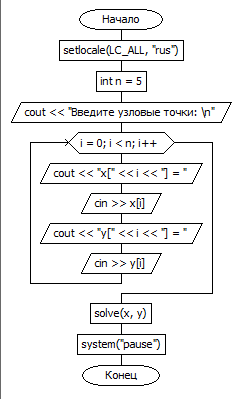




Алгоритм решения Solve



Main



### Листинг программы

#include <iostream> #include <vector> using namespace std;

//void solve(double x[], double y[]);

void solve(double x[], double y[])

{

//вычисление конечных разностей double dY1[4];

double dY2[3]; double dY3[2]; double dY4[1];

for (int i = 0; i < 4; i++)

{

dY1[i] = y[i + 1] - y[i];

}

for (int i = 0; i < 3; i++)

{

dY2[i] = dY1[i + 1] - dY1[i];

}

for (int i = 0; i < 2; i++)

{

dY3[i] = dY2[i + 1] - dY2[i];

}

dY4[0] = dY3[1] - dY3[0];

//вычисление значений коэффициента полинома double h = x[1] - x[0];

double a[5]; a[0] = y[0];

a[1] = dY1[0] / h / 1;

a[2] = dY2[0] / (2 \* pow(h, 2));

a[3] = dY3[0] / (6 \* pow(h, 3));

a[4] = dY4[0] / (24 \* pow(h, 4));

int c;

double iX, Yinterp, m; vector <double> input; int e = 1;

while (e == 1)

{

//Ввод значений x input.clear();

cout << "Введите количество дополнительных точек: "; cin >> c;

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

{

cout << "x[" << i << "]= "; cin >> iX; input.push\_back(iX);

}

cout << endl;

//вычисления y по первой формуле ньютона

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

{

m = 1;

Yinterp = 0;

for (int j = 0; j < 5; j++)

{

Yinterp += m \* a[j]; m \*= input[i] - x[j];

}

cout << "Yinterp[" << i << "]= " << Yinterp << endl;

}

cout << "Если вы хотите ввести дополнительные значения, нажмите 1, нет

- 0" << endl;

cin >> e;

}

}

int main()

{

setlocale(LC\_ALL, "rus"); const int n = 5;

double x[n]; double y[n];

cout << "Введите узловые точки: \n"; for (int i = 0; i < n; i++)

{

cout << "x[" << i << "] = "; cin >> x[i];

cout << "y[" << i << "] = "; cin >> y[i];

}

}

/\*

-3.0

0.4

-1.5

1.2

0.0

0.8

1.5

1.7

3.0

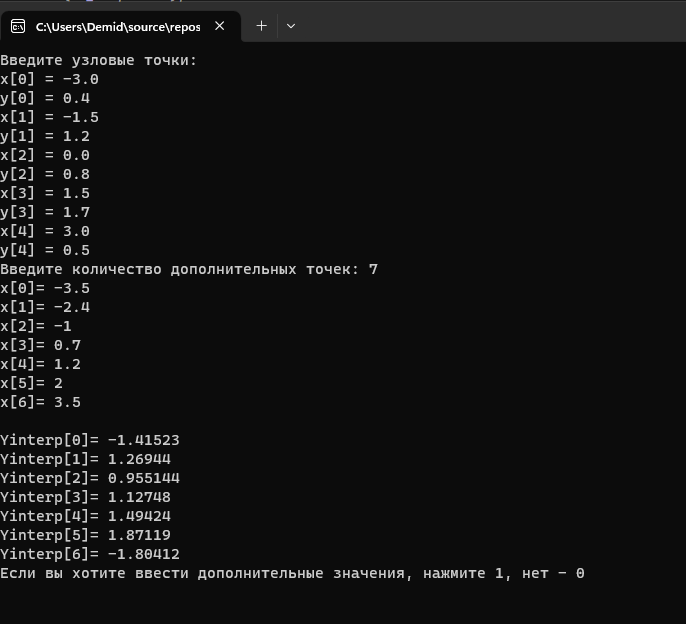
0.5

\*/

solve(x, y);

system("pause");

Результат работы программы



# Сравнение результатов:

Исходя из результатов программных и аналитических расчетов, вычисленные значения функции совпадают

|  |  |  |
| --- | --- | --- |
| X | Аналитические  результаты | Программные  результаты |
| -3.5 | -1,41523 | -1,41523 |
| -2.4 | 1.26944 | 1.26944 |
| -1 | 0.955144 | 0.955144 |
| 0.7 | 1.127478 | 1.12748 |
| 1.2 | 1.49424 | 1.49424 |
| 2 | 1.871193 | 1.87119 |
| 3.5 | -1,80412 | -1,80412 |

**Выводы:** в ходе выполнения данной практической работы были получены знания по интерполированию функции методом Ньютона, также была разработана программа для расчета значений функции на промежутке, по уже заданным x и значениями функции в x.