Федеральное государственное бюджетное образовательное

учреждение высшего образования

РЫБИНСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П.А. СОЛОВЬЕВА

Факультет радиоэлектроники и информатики

Кафедра математического и программного обеспечения электронных вычислительных средств

**Отчет по лабораторной работе №6**

по дисциплине

**Математические методы анализа данных**

на тему

«Интерполирование функций многочлена лагранжа.

Слайн-интерполяция»

Студент группы ИПБ-17 Мельников И.И.

Преподаватель Задорина Н. А.

Рыбинск 2021

1. **Задание**

1. Написать интерполяционный многочлен Лагранжа для функции f(x), которая задана на отрезке [x0,xn] в четырех точках (узлах). Значения функции взять из таблицы заданий Оценить погрешность интерполяции, предполагая, что ½f(n+1)(x)½£1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x  y | 0.2  -2 | 0.4  0 | 0.6  3 | 0.8  0 |

2. Разработать текст программы для приближенного вычисления значений функции f(x) и погрешности интерполяции в любой точке отрезка [x0,xn],

3. На ЭВМ набрать и отладить программу.

4. Провести вычисления функции в точках между заданными узлами. Провести интерполяцию с помощью программы MATHCAD и сравнить результаты.

5. Составить сплайн, заданный интерполяционной таблицей.

6. Проверить практическое совпадение значений «соседних» выражений сплайна в узловых точках, а также совпадение их со значениями функции в узлах интерполяции.

1. **Теоретические сведения**

Интерполяцией называется такой вид точечной аппроксимации, когда аппроксимирующая функция представляет собой алгебраический многочлен (полином) j(x) степени n, который в n+1 точке (узле) xi (i=0,1,...,n), заданных на отрезке [a,b], совпадает со значением аппроксимируемой функции f(x) в этих узлах, т.е. имеем yi=f(xi)=j(xi), i=0,1,...,n.

**Линейная и квадратичная** **интерполяция**.

Отрезок [a,b] делится узлами xi (i=0,1,...,n) на n частичных отрезков [xi-1,xi], при этом x0=a, xn=b.

При линейной интерполяции аппроксимируемая функция y=f(x) заменяется на каждом частичном отрезке [xi-1,xi] (i=1,2,...,n) многочленом первой степени т.е. прямой линией:

 (1)

проходящей через две точки, с координатами xi-1,yi-1=y(xi-1) и xi,yi=y(xi).

Следовательно на каждом отрезке [ xi-1,xi ] имеется своя прямая линия, которая описывается уравнением, проходящим через две точки. В результате для всего отрезка получаем ломаную линию, которая в узлах xi совпадает со значением функции. Коэффициенты ki и bi определяются из следующей системы уравнений:

, i=1,...n. (2)

Из (2) получаем значения неизвестных коэффициентов:

 (3)

Более точной является квадратичная интерполяция. В качестве интерполяционной функции на отрезке [ xi-1,xi+1 ] принимается квадратный трехчлен:

 (4)

Так как это уравнение параболы, то такую интерполяцию также называют параболической. Уравнение параболы содержит три неизвестных коэффициента ai, bi, ci, которые определяются из системы уравнений:

. (5)

Интерполяция для любой точки x отрезка [x0,xn] проходит по трем ближайшим точкам.

При линейной и параболической интерполяции имеются точки, где производная испытывает скачок. При линейной интерполяции это происходит в узлах, а при квадратичной там, где одни три точки заменяются на другие. Этого недостатка лишена интерполяция сплайнами.

**Интерполяционный многочлен Лагранжа.**

Рассмотрим глобальную интерполяцию на отрезке [x0,xn], т.е. построение единого интерполяционного многочлена степени n

 (8)

который в n+1 узле xi (i=0,1,...,n) совпадает со значениями аппроксимирующей функции.

 (9)

Коэффициенты ak (k=0,1,...,n) определяются из системы линейных уравнений (7) n+1 порядка.

При больших n необходимо решать систему линейных уравнений большого порядка, т.е. проводить большой объем вычислений. Избежать этого позволяет обобщенный многочлен Лагранжа степени n:

 (10)

где li(x) - многочлены Лагранжа, определяемые по формулам:

 (11)

где выражения в двойных скобках не должны учитываться, они написаны для пояснения алгоритма, по которому образуются эти многочлены.

Из (10) следует:

 (12)

Отметим, что при n=1 многочлен Лагранжа представляет собой линейную интерполяцию на отрезке [x0,x1], а при n=2 - квадратичную интерполяцию на отрезке [x0,x2].

**Интерполяция сплайнами.**

Сплайн (от англ. слова “splane” - гибкий) это функция, которая на всем отрезке интерполяции непрерывна вместе со своими *k* первыми производными (*k≤m-1)* и на каждом частичном отрезке представляет алгебраический многочлен (полином) степени *m*.

На практике широкое распространение получили сплайны третьей степени, имеющие на [*a,b*] непрерывную, по крайней мере, первую производную. Эти сплайны называются кубическими и обозначаются *S3(x)* (без указания дефекта).

Пусть на отрезке [*a,b*] в узлах сетки D заданы значения некоторой функции

*fi =f(xi), i=0,...,n*.

Интерполяционным кубическим сплайном *S3(x)* называется сплайн

*S3(x)=аi0 +аi1(x - xi)+аi2(x - xi)2 +аi3(x - xi)3*, *x*Î[*xi, xi+1*], (6)

удовлетворяющий условиям

*S3(xi)=f(xi), i=0,...,n*. (7)

Сплайн (6) на каждом из отрезков [*xi, xi+1*],*i=0,...,n-1* определяется четырьмя коэффициентами, и поэтому для его построения на всем промежутке [*a,b*] необходимо определить *4n* коэффициентов. Для их однозначного определения необходимо задать *4n* уравнений.

Условие (7) дает *2n* уравнений, при этом функция *S3(xi)*, удовле­творяющая этим условиям, будет непрерывна во всех внутренних узлах.

Условие непрерывности производных сплайна https://studfiles.net/html/1442/288/html_eO01zEKF0h.P2sg/img-62Ttu6.png,*r=1,2* во всех внутренних узлах *xi*,*i=1,...,n-1* сетки D дает *2(n-1)* равенств.

Вместе получается *4N-2* уравнений.

Два дополнительных условия обычно задаются в виде ограничений на значение производных сплайна на концах промежутка [*a,b*] и называются краевыми условиями.

Наиболее употребительны следующие типы краевых условий:

а) *S'3(а)=f'(а), S'(b)=f'(b)* ;

б) *S"3(а)=f"(а), S"(b)=f"(b)* ;

в) https://studfiles.net/html/1442/288/html_eO01zEKF0h.P2sg/img-8Z4Bl9.png;

г) *S'''3(xp+0)=S'''3(xp-0), р =1, n-1*.

Через краевые условия в конструкцию сплайна включаются параметры, выбирая которые можно управлять его поведением, особенно возле концов отрезка [*a,b*].

Условия типа в) носят названия периодических. Естественно требовать их выполнения в том случае, когда интерполируемая функция периодическая с периодом (*b-a*).

Если известны *f'(x)* или *f"(x)* в точках *а* и *b*, то естественно воспользоваться краевыми условиями типа а) или б).

Если производные неизвестны, то в большинстве случаев наилучшим решением будет применение краевых условий типа г).

Вместо значений производных можно использовать их разностные аналоги. При этом точность интерполяции вблизи концов падает.

Иногда предлагается принимать

*S"3(а)=S"3(b)=0*.

В этом случае вблизи концов точность интерполяции функции и ее первой производной уменьшается и становится соизмеримой с точностью интерполяции сплайном первой степени, что резко ухудшает всю картину.

1. **Результаты работы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x  y | 0.2  -2 | 0.4  0 | 0.6  3 | 0.8  0 |

Был составлен интерполяционный многочлен Лагранжа для функции f(x), которая задана на отрезке [x0,xn] в четырех точках (узлах). Данные были взяты из таблицы значений:

Погрешность интерполяции:



Rn(x) ≈ (x-0.2)\*(x-0.4)\*(x-0.6)\*(x-0.8) \*

**3.1 Исходный текст программы**

Вычисление факториала

long fact(int N){

if (N < 0)

return 0;

if (N == 0)

return 1;

else

return N \* fact(N - 1);

}  
Многочлен Лагранжа

double Lagrange(double\* X, double\* Y, double t){

double polynomial = 0, p1, p2;

int n = sizeof(X);

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

{

p1 = 1; p2 = 1;

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

{

if (i == j)

{

p1 = p1 \* 1; p2 = p2 \* 1;

}

else

{

p1 = p1 \* (t - X[i]);

p2 = p2 \* (X[j] - X[i]);

}

}

polynomial = polynomial + Y[j] \* p1 / p2;

}

return polynomial;

}  
Вычисление погрешности интерполяции

double infelicity(double\* X, double t){

double er = 0, p1;

p1 = 1;

int n = sizeof(X);

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

{

p1 = p1 \* (t - X[i]);

}

er = p1 / fact(n + 1);

return er;

}  
Точка входа в программу

int main(){

setlocale(LC\_ALL, "Russian");

double X[4] = { 0.2, 0.4, 0.6, 0.8 };

double Y[4] = { -2, 0, 3, 0 };

cout << "Ln(0.3) = " << Lagrange(X, Y, 0.3) << "\n ";

cout << "Погрешность = " << infelicity(X, 0.3) << "\n";

cout << "Ln(0.5) = " << Lagrange(X, Y, 0.5) << "\n " ;

cout << "Погрешность = " << infelicity(X, 0.5) << "\n";

cout << "Ln(0.7) = " << Lagrange(X, Y, 0.7) << "\n ";

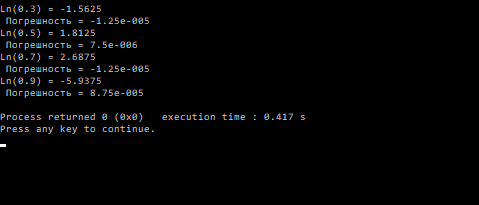
cout << "Погрешность = " << infelicity(X, 0.7) << "\n";

cout << "Ln(0.9) = " << Lagrange(X, Y, 0.9) << "\n ";

cout << "Погрешность = " << infelicity(X, 0.9) << "\n";

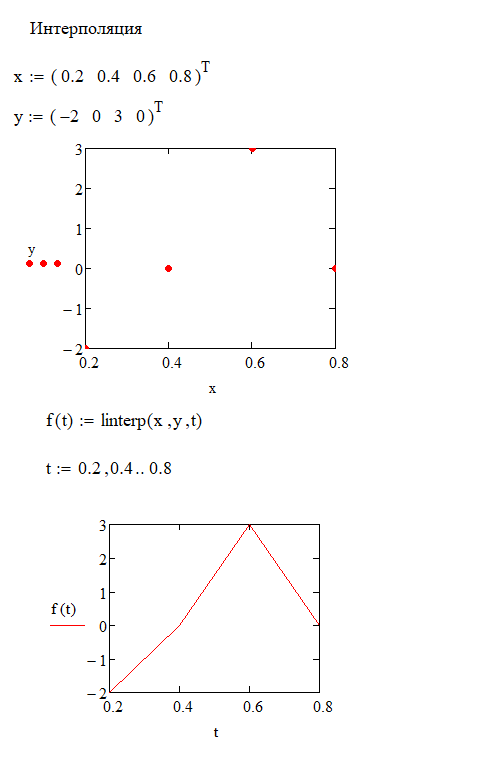
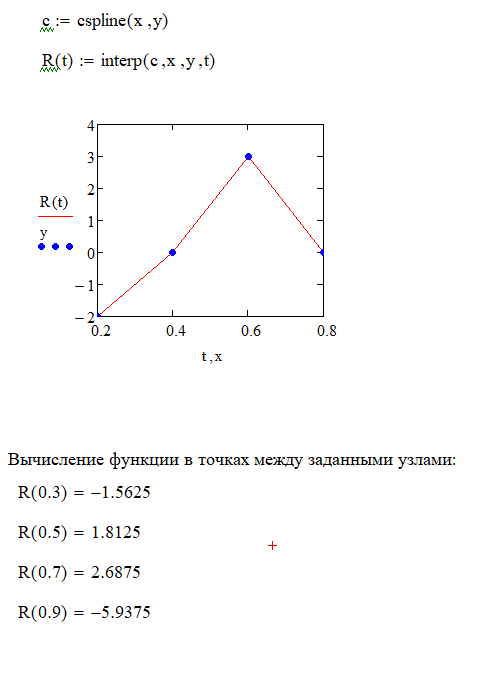
return 0;

}  
**3.2 Результат работы программы**

****

Можно заметить, что погрешность вычислений точек между заданными узлами не значительна.

**3.3 Интерполяция в MathCad**

**** 

Значения почти идентичны со значениями, вычисленными в программе, следовательно, погрешность вычислений тоже не значительная.

**4. Выводы**

В результате выполнения данной работы вычислен многочлен Лагранжа для заданной точками функции f(x), который получился

и его погрешность примерно равна (x-0.2)\*(x-0.4)\*(x-0.6)\*(x-0.8) \* в любой точке отрезка [x0;xn].

Разработана и отлажена на ЭВМ программа нахождения, приближенного значений функции f(x) и погрешности интерполяции в любой точке отрезка [x0;xn].

Проведено интерполирование в Mathcad и построена сплайн-интерполяция, также вычислены промежуточные значения некоторых точек функции f(x):

Сравнив значения, вычисленные программой и в Mathcad можно сделать вывод, что значения совпадают, следовательно, разработанная программа и сплайн-интерполяция, выполненная в Mathcad, примерно одинаково интерполируют функцию, с очень небольшой погрешностью.