1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский политехнический университет Петра Великого
3. —
4. Институт компьютерных наук и технологий
5. **Кафедра «Информационная безопасность компьютерных систем»**

**ОТЧЕТ**

**по лабораторной работе № 1**

1. по дисциплине «Вычислительная математика»
2. Выполнил
3. студент гр. 23508/4 Е.Г. Проценко
4. Проверил
5. профессор С.М. Устинов
6. Санкт-Петербург
7. 2016

# Формулировка задания (Вариант 29)

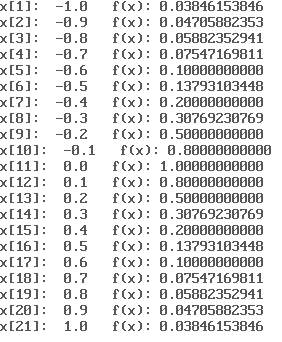
Для функции по узлам построить полином Лагранжа 20-й степени и сплайн-функцию S(x). Вычислить значения всех трех функций в точках . Результаты отобразить графически.

Используя программу QUANC8, вычислить два интеграла:

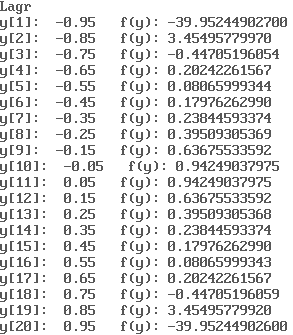
, для и для .

# Результаты работы

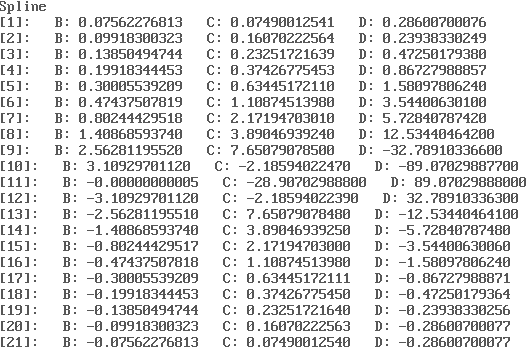
1. Получены точные значения для функции по узлам

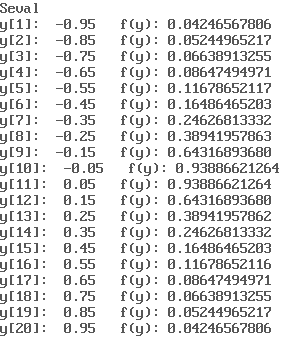


1. Полином Лагранжа 20-й степени. Вычислены значения функции в точках .

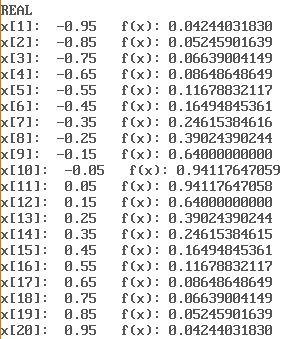


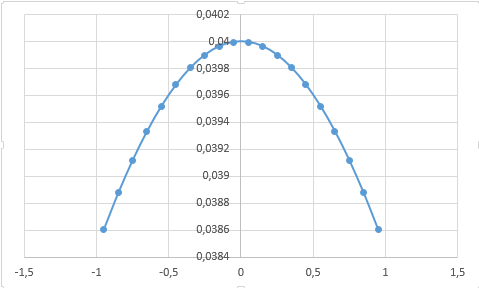
1. Коэффициенты для Spline и построена сплайн-функция. Вычислены значения функции в точках для нее.

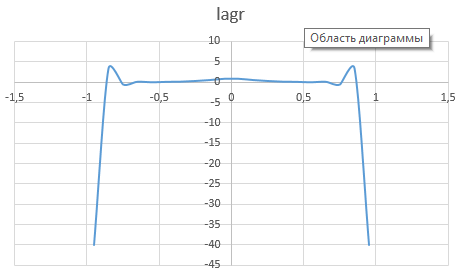


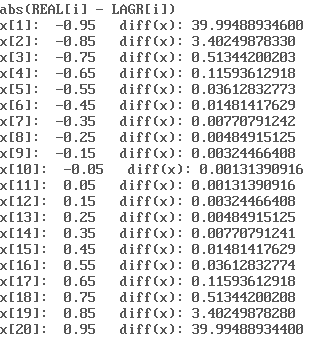


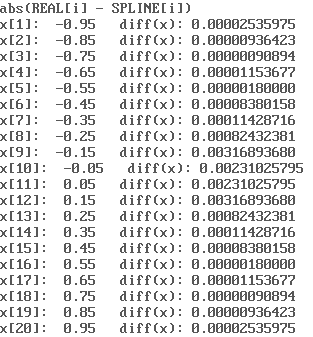
1. Настоящие значения функции в точках .



1. Графики исходной функции и сплайн-функции похожи, а график функции Лагранжа сильно отличается на краях отрезка.  
   

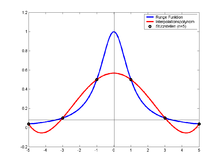


1. По графикам было бы невозможно определить какая функция строит исходную функцию точнее, поэтому для каждой точки бы вычитан модуль отклонения  
   

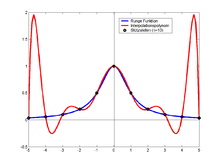


Видно, что интерполирование по Лагранжу точнее в середине отрезка, зато имеет огромную погрешность на его концах, сплайн-интерполирование дает лучший результат там.

Это связано с так называемым Феноменом Рунге. С возрастание степени полинома погрешность интерполяции стремится к бесконечности



Функция Рунге ([плотность вероятности](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%BE%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B2%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B8) [распределения Коши](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%9A%D0%BE%D1%88%D0%B8)) и интерполяционный полином 5-й степени



Функция типа Рунге f(x) = \frac{1}{1+x^2} и интерполяционный полином 10-й степени

1. Результаты работы QUANC8  
   

# вывод

Я познакомился на практике с построением интерполяционного полинома Лагранжа, с работой сплайн-функции, quanc8.

Интерполирование по Лагранжу оказалось очень неточным, особенно на концах отрезка. Это связано со спецификой функции. Для подобных функций интерполяционный полином Лагранжа будет иметь все большее отклонение, чем дальше мы от начала координат. В таком случае, конечно, интерполирование по Лагранжу не подходит, тогда в дело вступает сплайн интерполирование.

4 Приложение

Листинг написанной программы:

uses CRT, FMM, MATH;

Var

i: integer;

j: float;

x\_vector, f\_vector, y\_vector, f\_vector\_lagr, f\_vector\_s, f\_vector\_r: floatvector;

B, C, D: floatvector;

xk, yk: float;

abserr, relerr: float;

nofun: longint;

flag: float;

result: float;

errest: float;

{$F+}

function FUN(x: float) : float;

begin

FUN := 1/(1 + 25 \* x \* x);

end;

function FUN\_QUAN(x: float) : float;

begin

FUN\_QUAN := power(abs(1 - x \* x), -1);

end;

function FUN\_QUANUAN(x: float) : float;

begin

FUN\_QUANUAN := power(abs(1 - x \* x), -0.5);

end;

function LAGR(x\_vect, f\_vect: floatvector; N: integer; x: float) : float;

Var

i, j: integer;

total, result: float;

begin

total := 0;

result := 1;

for i:=1 to N + 1 do begin

for j:=1 to N + 1 do begin

if i <> j then begin

result := result \* (x - x\_vect[j])/(x\_vect[i] - x\_vect[j]);

end;

end;

total := total + result \* f\_vect[i];

result := 1;

end;

LAGR := total;

end;

{$F-}

begin

clrscr;

{PART I}

i := 1;

xk := -1;

while i <= 21 do begin

x\_vector[i] := xk;

f\_vector[i] := FUN(xk);

writeln('x[', i, ']: ', xk:2:1, ' f(x): ',f\_vector[i]:13:12);

i := i + 1;

xk := xk + 0.1;

end;

readln;

{LAGR}

writeln;

writeln('Lagr');

i := 1;

yk := -0.95;

while i <= 20 do begin

y\_vector[i] := yk;

f\_vector\_lagr[i] := LAGR(x\_vector, f\_vector, 20, yk);

writeln('y[', i, ']: ', yk:3:2, ' f(y): ',f\_vector\_lagr[i]:13:12);

{writeln('y[', i, ']: ', yk, ' f(y): ',f\_vector\_lagr[i]);}

i := i + 1;

yk := yk + 0.1;

end;

readln;

{SPLINE}

spline(21, x\_vector, f\_vector, B, C, D);

writeln;

writeln('Spline');

for i := 1 to 21 do begin

writeln('[', i, ']: B: ' , B[i]:13:12, ' C: ', C[i]:13:12, ' D: ', D[i]:13:12);

end;

readln;

{SEVAL}

writeln('Seval');

i := 1;

{yk := -0.95;}

while i <= 20 do begin

f\_vector\_s[i] := seval(21, y\_vector[i], x\_vector, f\_vector, B, C, D);

writeln('y[', i, ']: ', y\_vector[i]:3:2, ' f(y): ',f\_vector\_s[i]:13:12);

{writeln('y[', i, ']: ', y\_vector[i], ' f(y): ',f\_vector\_s[i]);}

i := i + 1;

{yk := yk + 0.1;}

end;

readln;

{REAL}

writeln('REAL');

i := 1;

while i <= 20 do begin

f\_vector\_r[i] := FUN(y\_vector[i]);

writeln('x[', i, ']: ', y\_vector[i]:2:1, ' f(x): ',f\_vector\_r[i]:13:12);

i := i + 1;

end;

readln;

{REAL - Lagr}

writeln('abs(REAL[i] - LAGR[i])');

i := 1;

while i <= 20 do begin

writeln('x[', i, ']: ', y\_vector[i]:2:1, ' diff(x): ', abs(f\_vector\_r[i] - f\_vector\_lagr[i]):13:12);

i := i + 1;

end;

readln;

{REAL - SPLINE}

writeln('abs(REAL[i] - SPLINE[i])');

i := 1;

while i <= 20 do begin

writeln('x[', i, ']: ', y\_vector[i]:2:1, ' diff(x): ', abs(f\_vector\_r[i] - f\_vector\_s[i]):13:12);

i := i + 1;

end;

readln;

{Quanc}

writeln('QUANC8');

abserr := 0.1;

relerr := 0.1;

quanc8(@FUN\_QUAN, 0, 2.14, abserr, relerr, result, errest, nofun, flag);

writeln('Result: ', result:13:12);

quanc8(@FUN\_QUANUAN, 0, 2.14, abserr, relerr, result, errest, nofun, flag);

writeln('Result: ', result:13:12);

readln;

end.