Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе 1**

**Дисциплина**: Вычислительная математика

Вариант 1

Выполнил студент гр. 3530901/90003 В.С. Андрианов

(подпись)

Преподаватель В.Н. Цыган

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

**Содержание**

1. **Техническое задание**
2. **Цель работы**
3. **Ход выполнения работы**
   1. **Объяснение подхода к решению поставленной задачи**
   2. **Описание входных и выходных данных для работы исследуемых процедур**
   3. **Листинг программы**
   4. **Результат выполнения программы**
4. **Вывод**

**1. Техническое задание**

Для функции F(x) = по узлам = -1 + 0.1k (k=0, 1, …, 20) построить

полином Лагранжа L(x) 20-й степени и сплайн-функцию S(x). Также

построить графики трех указанных функций – f(x), L(x), S(x). Затем сравнить

значения трех интегралов: . Первый интеграл вычислить

аналитически, а для вычисления второго и третьего использовать программу

QUANC8.

**2. Цель работы**

Использовать в рабочих целях реализацию численных методов, таких как Lagrange, Spline, Quanc8

**3. Ход выполнения работы**

**3.1 Объяснение подхода к решению поставленной задачи**

Сначала создаем пустые массивы, которые будут хранить узлы, значения функции в этих узлах, коэффициенты для работы комбинации процедур spline – seval.

Далее создаем дополнительные функции. Первая функция – функция, которая рассчитывает узлы интерполирования (= -1 + 0.1k (k=0, 1, …, 20)). Вторая функция – функция, которая рассчитывает значение F(x) в узлах (F(x) = ). Третья и четвертая функции – это функции-обертки (spline-seval и lagrange) для работы процедуры Quanc8.

После дополнительных функций идет главная функция. В ней описывается вся основная работа. Сначала находим узлы интерполирования и значения F(x) в узлах. Далее используем процедуру spline, которая решает систему относительно и заполняет массивы коэффициентов b, c, d (то есть коэффициентами системы: .

Далее проходим по массиву узлов, параллельно строим полином Лагранжа двадцатой степени, и, с помощью процедуры seval, решаем систему в узлах интерполирования. Выводим в консоль значения функций F(x), L(x), S(x).

Далее (c помощью процедуры Quanc8) находим значения интегралов: . Выводим результаты в консоль.

* 1. **Описание входных и выходных данных для работы исследуемых процедур**

В процедуру **Lagrange(n, X, Y, p)** передаются:

n – целое число, количество точек, по которому строится полином

X – вектор, элементами которого являются узлы интерполирования

Y – вектор, элементами которого являются значения функции F(x) в узлах интерполирования

p – текущая точка

В процедуру **Spline(n, X, F, B, C, D)** передаются:

n – число точек

X – вектор, элементами которого являются узлы интерполирования

Y – вектор, элементами которого являются значения функции F(x) в узлах интерполирования

B, C, D – векторы с коэффициентами, результат работы

В процедуру **Seval(n, p, X, F, B, C, D)** передаются:

n – целое число, количество точек, по которому строится полином

p – текущая точка

X – вектор, элементами которого являются узлы интерполирования

Y – вектор, элементами которого являются значения функции F(x) в узлах интерполирования

В процедуру **QUANC8 (FUN, A, B, ABSERR, RELERR, RESULT, ERREST, NOFUN, FLAG)** передаются:

FUN – имя подпрограммы-функции, вычисляющей значение подынтегральной функции f(x)

A, B – нижний и верхний пределы интегрирования

ABSERR и RELERR – границы абсолютной и относительной погрешностей

RESULT – значение интеграла

ERREST – оценка погрешности, выполненная программой

NOFUN – количество вычислений подынтегральной функции, использованных для получения результата

FLAG – индикатор надежности результата. Нулевое значение этой переменной отвечает относительной надежности результата, а ненулевое, свидетельствует об отклонениях от нормального хода выполнения программы

RESULT, ERREST, NOFUN, FLAG – результат работы

* 1. **Листинг программы**

1. #include <iostream>  
   #include "lagrange.h"  
   #include "SPLINES.H"  
   #include "quanc8.h"  
     
   // массивы в которых будут храниться:  
   // точки по которым ищем значения функции и узлы для построения полиномов  
   double arrayX[21], \*x = arrayX;  
     
   // значения функции в точках, которые хранятся в массиве выше  
   double arrayF[21], \*f = arrayF;  
     
   // массив, в котором хранятся коэффициенты b (сплайн интерполяции)  
   double arrayB[21], \*b = arrayB;  
     
   // массив, в котором хранятся коэффициенты c (сплайн интерполяции)  
   double arrayC[21], \*c = arrayC;  
     
   // массив, в котором хранятся коэффициенты d (сплайн интерполяции)  
   double arrayD[21], \*d = arrayD;  
     
     
   // функция, которая вычисляет k точек, по которым строятся полиномы  
   double X(int k) {  
    return -1 + 0.1 \* k;  
   }  
     
   // функция, которая считает значение заданной функции  
   double F(double t) {  
    return (1/(1 + 25 \* pow(t, 2)));  
   }  
     
   // функция-обертка для нахождения значения полинома Лагранжа с помощью функции quanc8  
   double lagrangeForQuanc(double t) {  
    return lagrange(20, x, f, t);  
   }  
     
   // функция-обертка для нахождения значения сплайн-полинома с помощью функции quanc8  
   double splineForQuanc(double t) {  
    double \*p = &t;  
    return seval(20, p, x, f, b, c, d);  
   }  
     
   int main() {  
     
    // заполняем массивы arrayX и arrayF  
    for (int i = 0; i < 21; i++) arrayX[i] = X(i);  
    for (int i = 0; i < 21; i++) arrayF[i] = F(arrayX[i]);  
     
    // интерполируем сплайнами  
    spline(20, x, f, b, c, d);  
     
    for (int i = 0; i < 21; i++) {  
     
    // интерполируем полиномом Лагранжа и ищем значение функции в точках arrayX  
    double resLagrange = lagrange(20, x, f, arrayX[i]);  
     
    // рещаем систему, состоящую из полиномов третьей степени  
    double spline = seval(20, &arrayX[i], x, f, b, c, d);  
     
    // выводим результаты в консоль  
    if (i == 0) printf("\n x F(x) L(x) S(x)\n");  
      
    if (i < 9) printf("%f %f %f %f\n", arrayX[i], arrayF[i], resLagrange, spline);  
    if (i > 9) printf(" %f %f %f %f\n", arrayX[i], arrayF[i], resLagrange, spline);  
    }  
     
    // абсолютная и относительная погрешность  
    double abserr = 1.0e-14, relerr = 0.0;  
     
    // реальная погрешность  
    double errest, \*Errest = &errest;  
     
    // индикатор надежности  
    double flag, \*Flag = &flag;  
     
    // количество внутренних вычислений функции  
    int nofun, \*Nofan = &nofun;  
     
    // результат нахождения интеграла  
    double res, \*result = &res;  
     
    // использование функции, которая находит значения функции в точках x  
    double (\*FUN1) (double x); FUN1 = F;  
     
    // использование функции-обертки интерполирования полиномом Лагранжа  
    double (\*FUN2) (double x); FUN2 = lagrangeForQuanc;  
     
    // использование функции-обертки интерполирования сплайнами  
    double (\*FUN3) (double x); FUN3 = splineForQuanc;  
     
    printf("\nFunction Integration result Errest NoFan Flag\n");  
     
    // нахождение и вывод интеграла от функции  
    quanc8(FUN1, -1.0, 1.0, abserr, relerr, result, Errest, Nofan, Flag);  
    printf("F(x) %f %f %d %f\n", \*result, errest, nofun, flag);  
     
    // нахождение и вывод интеграла от полинома Лагранжа  
    quanc8(FUN2, -1.0, 1.0, abserr, relerr, result, Errest, Nofan, Flag);  
    printf("Lagrange %f %f %d %f\n", \*result, errest, nofun, flag);  
     
    // нахождение и вывод интеграла от сплайнов  
    quanc8(FUN3, -1.0, 1.0, abserr, relerr, result, Errest, Nofan, Flag);  
    printf("Spline %f %f %d %f\n", \*result, errest, nofun, flag);  
     
    return 0;  
      
   }

**3.4 Результат выполнение программы**

x – узлы интерполирования

F(x) – значения функции в x

L(x) – значения полинома Лагранжа в x

S(x) – значения Сплайн интерполяции в x

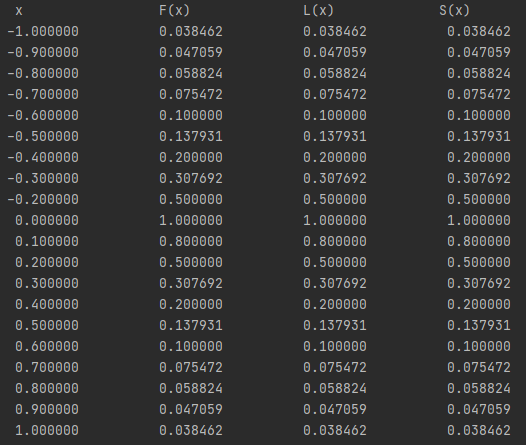


Рис. 1 Значения функций в узлах интерполирования

Из рисунка 1 видно, что значения функций одинаковы в узлах интерполирования. Ожидался именно такой результат, так как были построены интерполяционные полиномы (от которых и требуется точное значение в узлах интерполирования) от заданной функции.

Для детального рассмотрения функций L(X) и S(x) рассмотрим их значения между узлами интерполирования.

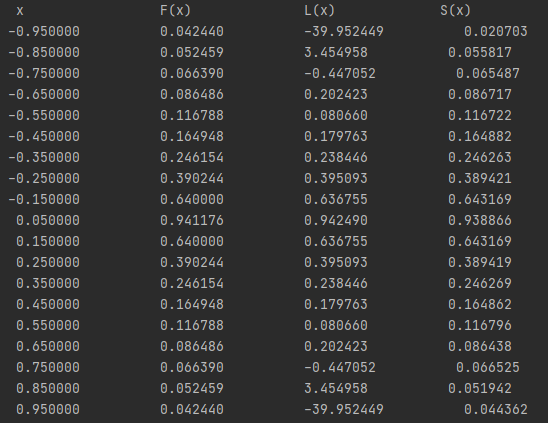
****

Рис. 2 Значения функций между узлами интерполирования

На рисунке 2 представлены значения функций между узлами интерполирования.

Для представления отклонений интерполяционных полиномов от заданной функции построим таблицу.

Табл.1 - Значения функций между узлами интерполирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | F(x) | L(x) | S(x) | |F(x)-L(x)| | |F(x)-S(x)| |
| -0,95 | 0,04244 | -39,9524 | 0,020703 | 39,994889 | 0,021737 |
| -0,85 | 0,052459 | 3,454958 | 0,055817 | 3,402499 | 0,003358 |
| -0,75 | 0,06639 | -0,44705 | 0,065487 | 0,513442 | 0,000903 |
| -0,65 | 0,086486 | 0,202423 | 0,086717 | 0,115937 | 0,000231 |
| -0,55 | 0,116788 | 0,08066 | 0,116722 | 0,036128 | 6,6E-05 |
| -0,45 | 0,164948 | 0,179763 | 0,164882 | 0,014815 | 6,6E-05 |
| -0,35 | 0,246154 | 0,238446 | 0,246263 | 0,007708 | 0,000109 |
| -0,25 | 0,390244 | 0,395093 | 0,389421 | 0,004849 | 0,000823 |
| -0,15 | 0,64 | 0,636755 | 0,643169 | 0,003245 | 0,003169 |
| 0,05 | 0,941176 | 0,94249 | 0,938866 | 0,001314 | 0,00231 |
| 0,15 | 0,64 | 0,636755 | 0,643169 | 0,003245 | 0,003169 |
| 0,25 | 0,390244 | 0,395093 | 0,389419 | 0,004849 | 0,000825 |
| 0,35 | 0,246154 | 0,238446 | 0,246269 | 0,007708 | 0,000115 |
| 0,45 | 0,164948 | 0,179763 | 0,164862 | 0,014815 | 8,6E-05 |
| 0,55 | 0,116788 | 0,08066 | 0,116796 | 0,036128 | 8E-06 |
| 0,65 | 0,086486 | 0,202423 | 0,086438 | 0,115937 | 4,8E-05 |
| 0,75 | 0,06639 | -0,44705 | 0,066525 | 0,513442 | 0,000135 |
| 0,85 | 0,052459 | 3,454958 | 0,051942 | 3,402499 | 0,000517 |
| 0,95 | 0,04244 | -39,9524 | 0,044362 | 39,994889 | 0,001922 |

Из таблицы 1 видно, что интерполяционный полином Лагранжа плохо описывает F(x) в начале и конце промежутка. Также стоит отметить, что отклонение полинома Лагранжа в точках между узлами интерполирования больше, чем у сплайн интерполяции.

На рисунках 3, 4 и 5 представлены графики F(x), L(x), S(x).

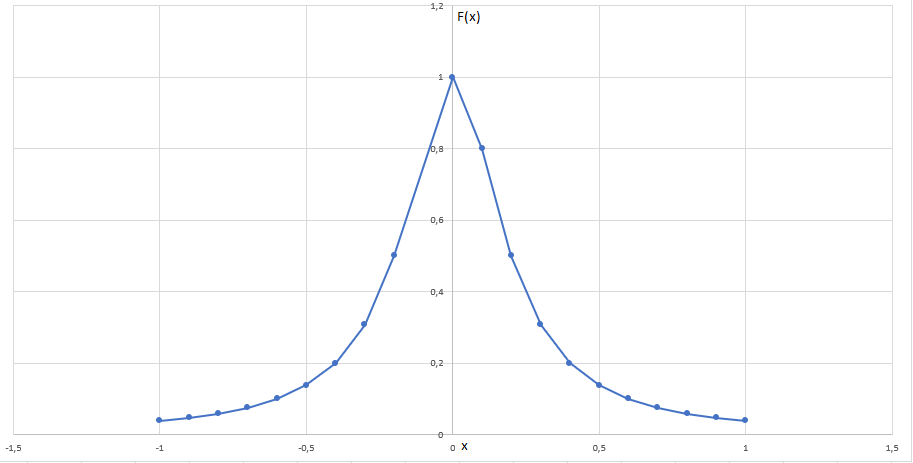


Рис. 3 График F(x)

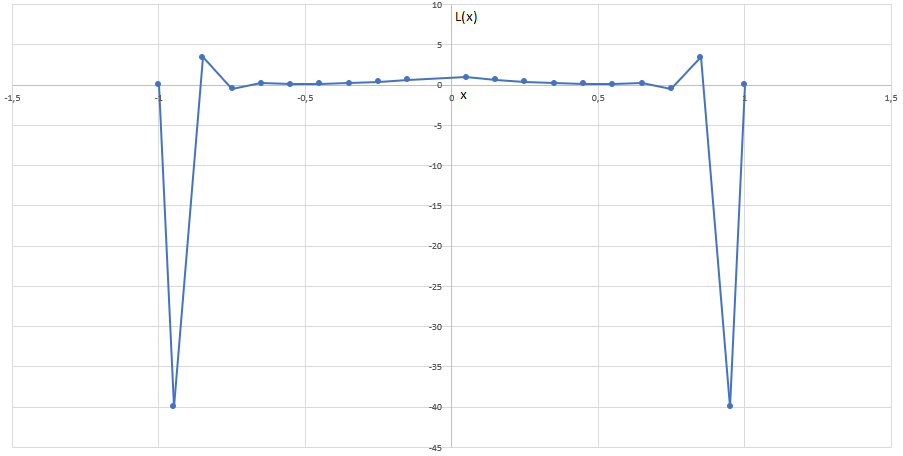


Рис. 4 График L(x)

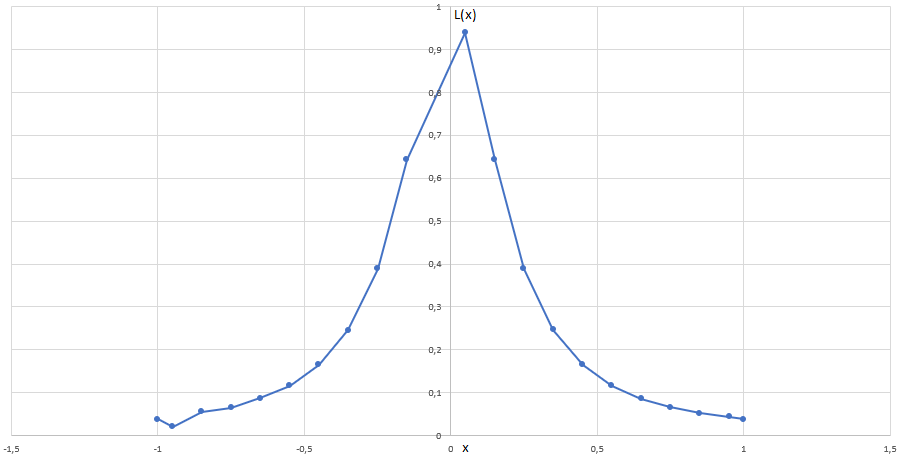


Рис. 5 График S(x)

Из графиков видно, что сплайн интерполяция лучше описывает изначальную функцию.

Теперь аналитически вычислим интеграл:

Рассчитаем интеграл для функций F(x), L(x) и S(x) с помощью процедуры Quanc8.

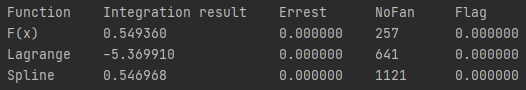


Рис. 6 Результаты работы процедуры Quanc8

Из рисунка 6 видно, что значение интеграла от сплайн-функции близко к значению интеграла от заданной функции. Также стоит сказать, что интеграл от полинома Лагранжа значительно отличается от интегралов от других функций, причину такого различия можно увидеть на рисунке 4 (график функции L(x)).

Реальная погрешность работы Quanc8 во всех случаях равна 0. Также видно, что меньше всего вычислений ушло на функцию F(x) (257), а больше всего на функцию L(x) (1121). Индикатор надежности во всех случаях равен 0, то есть результатам можно доверять.

**4. Вывод**

В ходе выполнения работы было проведено исследование работы нескольких процедур, а именно: Lagrange, Spline-Seval, Quanc8.

В результате использования этих процедур были получены интерполяционный полином Лагранжа и Сплайн интерполяция. Оба метода интерполяции были сравнены между собой. По всем функциям были построены графики, которые отражают поведение функций между узлами интерполирования.

Также были найдены интегралы от полученных функций с помощью процедуры Quanc8. Как и ожидалось сплайн-интерполяция довольно хорошо описывает функцию на всем промежутке. Полином Лагранжа хуже описывает F(x), чем сплайн интерполяция., но на концах отрезка полином Лагранжа имеет сильные “скачки”. Из-за этих “скачков” результат работы Quanc8 сильно отличается от эталонного значения.