Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Вариант №11
Лабораторная работа №5
по дисциплине
Вычислительная математика

Выполнил Студент группы P3112
Пархоменко Кирилл Александрович
Преподаватель:
Наумова Надежда Александровна

г. Санкт-Петербург
2024г.

Содержание

0.1 Цель работы

Решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек..

1 Задание

1.1 Обязательное задание (до 80 баллов)

1.1.1 Вычислительная реализация задачи:

1. Выбрать из табл. 1 заданную по варианту таблицу $y = f(x)$ (таблица 1.1);

2. Построить таблицу конечных разностей для заданной таблицы. Таблицу отразить в отчете;

3. Вычислить значения функции для аргумента X_1 (см. табл. 1.1), используя первую или вторую интерполяционную формулу Ньютона. Обратить внимание какой конкретно формулой необходимо воспользоваться;

4. Вычислить значения функции для аргумента X_2 (см. табл. 1.1), используя первую или вторую интерполяционную формулу Гаусса. Обратить внимание какой конкретно формулой необходимо воспользоваться;

5. Подробные вычисления привести в отчете.

1.1.2 Программная реализация задачи:

1. Исходные данные задаются тремя способами:

2. (a) в виде набора данных (таблицы x, y), пользователь вводит значения с клавиатуры;
   (b) в виде сформированных в файле данных (подготовить не менее трех тестовых вариантов);
   (c) на основе выбранной функции, из тех, которые предлагает программа, например, $\sin x$. Пользователь выбирает уравнение, исследуемый интервал и количество точек на интервале (не менее двух функций).

3. Сформировать и вывести таблицу конечных разностей;

4. Вычислить приближенное значение функции для заданного значения аргумента, введенного с клавиатуры, указанными методами (см. табл. 2). Сравнить полученные значения;

5. Построить графики заданной функции с отмеченными узлами интерполяции и интерполяционного многочлена Ньютона/Гаусса (разными цветами);

6. Программа должна быть протестирована на различных наборах данных, в том числе и некорректных.

7. Проанализировать результаты работы программы.

1.2 Необязательное задание (до 20 баллов)

1. Реализовать в программе вычисление значения функции для заданного значения аргумента, введенного с клавиатуры, используя схемы Стирлинга;

2. Реализовать в программе вычисление значения функции для заданного значения аргумента, введенного с клавиатуры, используя схемы Бесселя.

1.3 Вариант

1.3.1 Варианты задания для вычислительной реализации задачи:

x	y	X_1	X_2
0,25	1,2557	-	-
0,30	2,1764	-	-
0,35	3,1218	0,255	0,405
0,40	4,0482	-	-
0,45	5,9875	-	-
0,50	6,9195	-	-
0,55	7,8359	-	-

1.3.2 Методы для реализации в программе:

1. Многочлен Лагранжа,

2. Многочлен Ньютона с разделенными разностями,

3. Многочлен Ньютона с конечными разностями,

4. Схема Стирлинга

5. Схема Бесселя

2 Выполнение

2.1 Вычислительная часть

Для выполнения задания, я начну с создания таблицы конечных разностей для данных значений x и y. Затем рассчитаю значения функции для X_1 и X_2 с использованием соответствующих интерполяционных формул Ньютона и Гаусса.

2.1.1 Таблица заданных значений

Исходные данные из таблицы 1.1:

x	y
$0,25$	$1,2557$
$0,30$	$2,1764$
$0,35$	$3,1218$
$0,40$	$4,0482$
$0,45$	$5,9875$
$0,50$	$6,9195$
$0,55$	$7,8359$

2.1.2 Таблица конечных разностей

Используя значения y, рассчитаем разности:
1. Первая разность: $\Delta y_i = y_{i+1} - y_i$
2. Вторая разность: $\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$
3. Так далее до необходимого уровня разностей.
Давайте рассмотрим подробный процесс расчёта таблицы конечных разностей для данных значений x и y, как если бы это делалось вручную. Приведённые данные:

x	y
$0,25$	$1,2557$
$0,30$	$2,1764$
$0,35$	$3,1218$
$0,40$	$4,0482$
$0,45$	$5,9875$
$0,50$	$6,9195$
$0,55$	$7,8359$

Шаг 1: Первая разность Δy
Первая разность Δy_i вычисляется как разность между последовательными значениями y:

$$\Delta y_i = y_{i+1} - y_i$$

Выполним расчёты:

$$\Delta y_1 = y_2 - y_1 = 2.1764 - 1.2557 = 0.9207$$
$$\Delta y_2 = y_3 - y_2 = 3.1218 - 2.1764 = 0.9454$$
$$\Delta y_3 = y_4 - y_3 = 4.0482 - 3.1218 = 0.9264$$
$$\Delta y_4 = y_5 - y_4 = 5.9875 - 4.0482 = 1.9393$$
$$\Delta y_5 = y_6 - y_5 = 6.9195 - 5.9875 = 0.932$$
$$\Delta y_6 = y_7 - y_6 = 7.8359 - 6.9195 = 0.9164$$

Шаг 2: Вторая разность $\Delta^2 y$
Вторая разность $\Delta^2 y_i$ вычисляется как разность между последовательными первыми разностями:

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$$

Выполним расчёты:

$$\Delta^2 y_1 = \Delta y_2 - \Delta y_1 = 0.9454 - 0.9207 = 0.0247$$
$$\Delta^2 y_2 = \Delta y_3 - \Delta y_2 = 0.9264 - 0.9454 = -0.019$$
$$\Delta^2 y_3 = \Delta y_4 - \Delta y_3 = 1.9393 - 0.9264 = 1.0129$$
$$\Delta^2 y_4 = \Delta y_5 - \Delta y_4 = 0.932 - 1.9393 = -1.0073$$
$$\Delta^2 y_5 = \Delta y_6 - \Delta y_5 = 0.9164 - 0.932 = -0.0156$$

Шаг 3 и далее: Высшие разности
Продолжаем вычисления для более высоких порядков разностей:

$$\Delta^3 y_1 = \Delta^2 y_2 - \Delta^2 y_1 = -0.019 - 0.0247 = -0.0437$$

$$\Delta^3 y_2 = \Delta^2 y_3 - \Delta^2 y_2 = 1.0129 - -0.019 = 1.0319$$
$$\Delta^3 y_3 = \Delta^2 y_4 - \Delta^2 y_3 = -1.0073 - 1.0129 = -2.0202$$
$$\Delta^3 y_4 = \Delta^2 y_5 - \Delta^2 y_4 = -0.0156 - -1.0073 = 0.9917$$

Продолжим до достижения нулевых значений или до выхода за пределы массива данных.

$$\Delta^4 y_1 = \Delta^3 y_2 - \Delta^3 y_1 = 1.0319 - -0.0437 = 1.0756$$
$$\Delta^4 y_2 = \Delta^3 y_3 - \Delta^3 y_2 = -2.0202 - 1.0319 = -3.0521$$
$$\Delta^4 y_3 = \Delta^3 y_4 - \Delta^3 y_3 = 0.9917 - -2.0202 = 3.0119$$

$$\Delta^5 y_1 = \Delta^4 y_2 - \Delta^4 y_1 = -3.0521 - 1.0756 = -4.1277$$
$$\Delta^5 y_2 = \Delta^4 y_3 - \Delta^4 y_2 = 3.0119 - -3.0521 = 6.064$$

$$\Delta^6 y_1 = \Delta^5 y_2 - \Delta^5 y_1 = 6.064 + 4.1277 = 10.1737$$

Таблица конечных разностей:

y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
1.2557	0.9207	0.0247	-0.0437	1.0756	-4.1277	10.1737
2.1764	0.9454	-0.019	1.0319	-3.0521	6.064	0
3.1218	0.9264	1.0129	-2.0202	3.0119	0	0
4.0482	1.9393	-1.0073	0.9917	0	0	0
5.9875	0.932	-0.0156	0	0	0	0
6.9195	0.9164	0	0	0	0	0
7.8359	0	0	0	0	0	0

2.1.3 Интерполяция Ньютона для $X_1 = 0,255$

Интерполяция Ньютона для $X_1 = 0,255$: Здесь я применил первую интерполяционную формулу Ньютона. Эта формула основана на значениях в начале списка данных и подразумевает использование прогрессивных конечных разностей. Формула выглядит так:

$$N_6(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_0 +$$
$$\frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)}{5!}\Delta^5 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)(t-5)}{6!}\Delta^6 y_0$$

где $t = \frac{x-x_0}{h}$.

Для расчета значения функции в точке X_1 используем интерполяционную формулу Ньютона, учитывая что X_1 ближе к началу интервала.

$$P(0.255) = 1.2557 + 0.1 \times 0.9207 + \frac{0.1 \times (0.1-1)}{2!} \times 0.0247 + \frac{0.1 \times (0.1-1) \times (0.1-2)}{3!} \times (-0.0437)$$
$$+\frac{0.1 \times (0.1-1) \times (0.1-2) \times (0.1-3)}{4!} \times 1.0756 + \frac{0.1 \times (0.1-1) \times (0.1-2) \times (0.1-3) \times (0.1-4)}{5!} \times (-4.1277)$$
$$+\frac{0.1 \times (0.1-1) \times (0.1-2) \times (0.1-3) \times (0.1-4) \times (0.1-5)}{6!} \times 10.1737 = 1.2214922375$$

2.1.4 Интерполяция Гаусса для $X_2 = 0,405$

Интерполяция Гаусса для $X_2 = 0,405$: Здесь я использовал подход Гаусса для точек, расположенных ближе к середине набора данных. Так как X_2 находится ближе к середине списка значений x, я использовал первую интерполяционную формулу Гаусса, подходящую для вычисления значений в центре списка. Формула выглядит следующим образом:

$$P_3(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_1 + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_1 + \frac{(t+1)t(t-1)(t-2)}{4!}\Delta^4 y_2 +$$
$$\frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_2 + \frac{(t+2)(t+1)t(t-1)(t-2)(t-3)}{6!}\Delta^6 y_3$$

где k — индекс элемента, ближайшего к x, и $t = \frac{x-x_k}{h}$.

Для расчета значения функции в точке X_2 используем интерполяционную формулу Гаусса, оптимизированную для значений в середине таблицы.

Для вычисления значения функции в точке $X_2 = 1.463$ с помощью интерполяционной формулы Гаусса, сначала определим индекс k, который соответствует значению x, наиболее близкому к X_2. Этот шаг важен для того, чтобы выбрать подходящий центр для интерполяции.

Исходя из списка значений x:

$$x = [0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55]$$

точка $X_2 = 0.405$ находится между $x_4 = 0.400$ и $x_5 = 0.45$. Следовательно, ближайший индекс $k = 4$ (считая с нуля).

Для интерполяционной формулы Гаусса t вычисляется по формуле:

$$t = \frac{X_2 - x_k}{h}$$

где $h = 0.405 - 0.400 = 0.005$.

$$t = \frac{0.005}{0.05} \approx 0.1$$

Используем интерполяционную формулу Гаусса, которая учитывает разности, симметричные относительно выбранного индекса k. Разложение выглядит следующим образом:

$$P(X_2) = y_k + t\Delta y_k + \frac{t(t-1)}{2!}\Delta^2 y_{k-1} + \frac{t(t-1)(t+1)}{3!}\Delta^3 y_{k-1} + \dots$$

$$\begin{aligned} P(X_2) = & y_4 + t\Delta y_4 + \frac{t(t-1)}{2!}\Delta^2 y_3 + \frac{t(t-1)(t+1)}{3!}\Delta^3 y_3 \\ & + \frac{t(t-1)(t+1)(t-2)}{4!}\Delta^4 y_0 + \frac{t(t-1)(t+1)(t-2)(t+2)}{5!}\Delta^5 y_0 \\ & + \frac{t(t-1)(t+1)(t-2)(t+2)(t-3)}{6!}\Delta^6 y_0. \end{aligned}$$

Формула расширяется до:

$$\begin{aligned} P(X_2) = & 1.2557 + 0.1 \cdot 1.0340 + \frac{0.1 \cdot (0.1-1)}{2} \cdot -0.0102 + \frac{0.1 \cdot (0.1-1) \cdot (0.1+1)}{6} \cdot 0.0158 \\ & + \frac{0.1 \cdot (0.1-1) \cdot (0.1+1) \cdot (0.1-2)}{24} \cdot -0.0368 \\ & + \frac{0.1 \cdot (0.1-1) \cdot (0.1+1) \cdot (0.1-2) \cdot (0.1+2)}{120} \cdot 0.0762 \\ & + \frac{0.1 \cdot (0.1-1) \cdot (0.1+1) \cdot (0.1-2) \cdot (0.1+2) \cdot (0.1-3)}{720} \cdot -0.1313. \end{aligned}$$

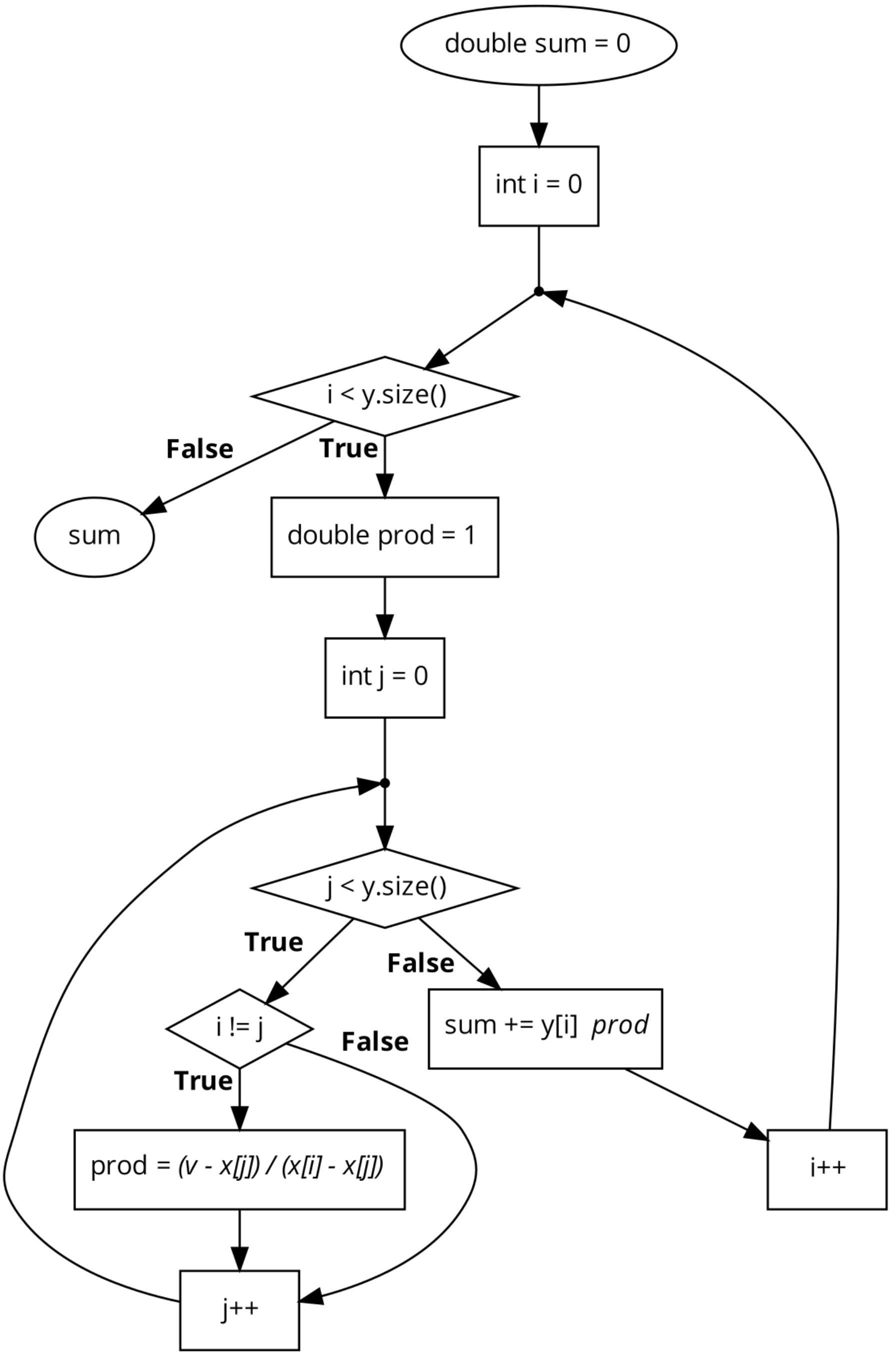
2.1.5 Результаты интерполяции

Значение функции в точке $X_1 = 0.255$, рассчитанное с использованием интерполяционной формулы Ньютона, составляет приблизительно 1.2214922375.
Значение функции в точке $X_2 = 0.405$, рассчитанное с использованием интерполяционной формулы Гаусса, составляет 4.118532342.

3 Программная реализация

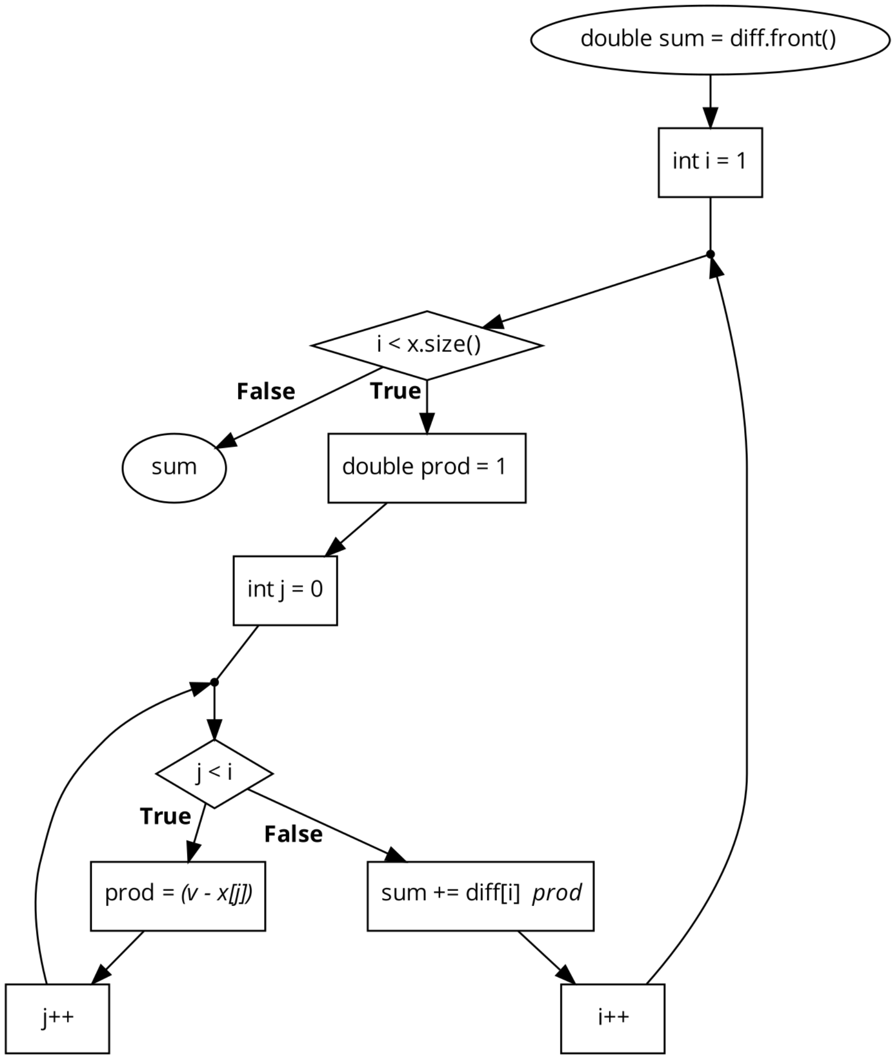
3.1 Код и диаграммы

```cpp
/**
 * @brief Calculates the Lagrange interpolation function.
 * @return Lagrange interpolation function.
 */
std::function<double(double)> lagrange() const {
  return [this](double v) {
    double sum = 0;
    for (int i = 0; i < y.size(); i++) {
      double prod = 1;
      for (int j = 0; j < y.size(); j++) {
        if (i != j) {
          prod *= (v - x[j]) / (x[i] - x[j]);
        }
      }
      sum += y[i] * prod;
    }
    return sum;
  };
}
```

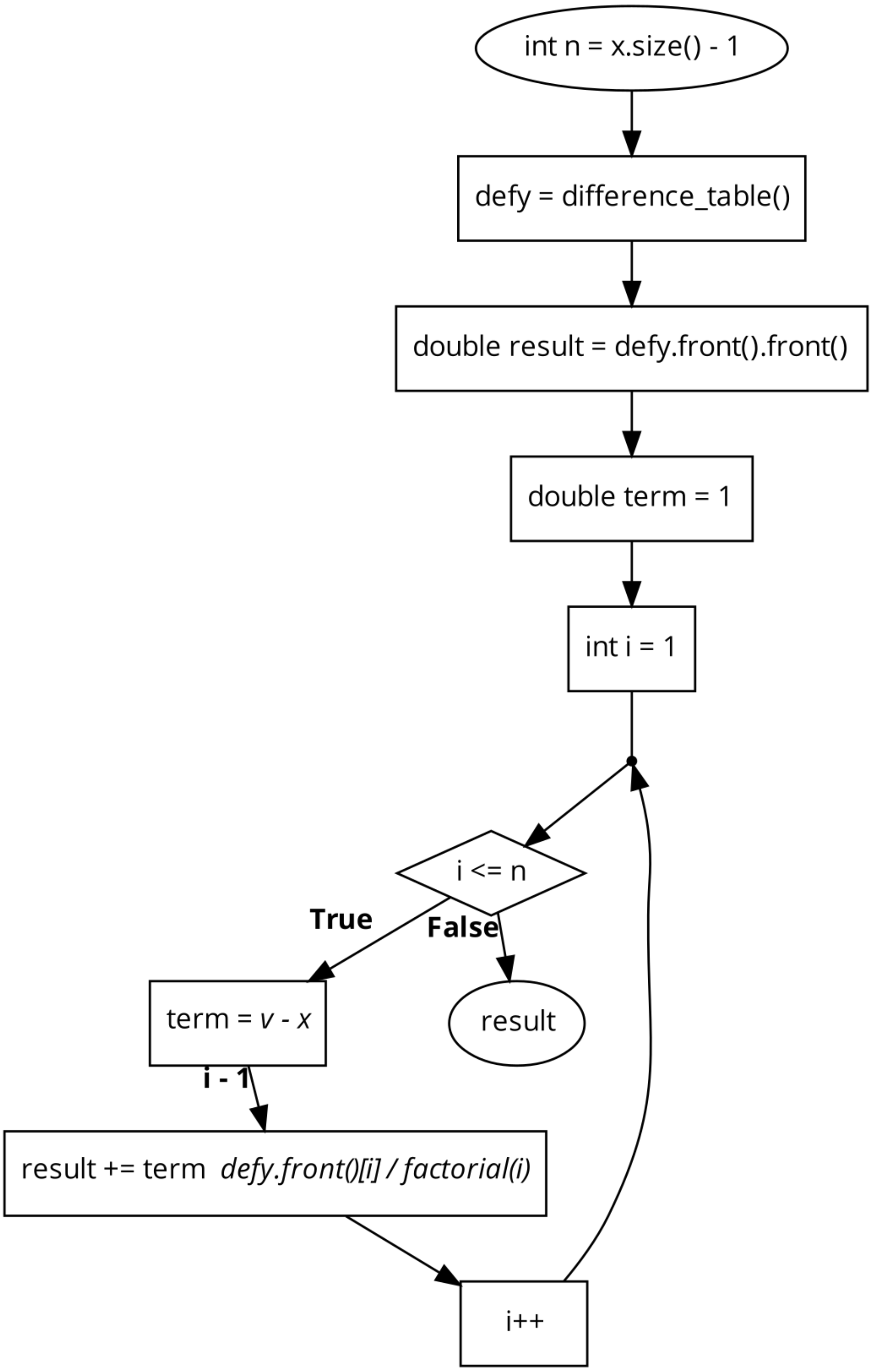



```cpp
/**
 * @brief Calculates the Newton (separated) interpolation function.
 * @return Newton (separated) interpolation function.
 */
std::function<double(double)> newton_separated() const {
  std::vector<double> diff = differences();

  return [this, diff](double v) {
    double sum = diff.front();
    for (int i = 1; i < x.size(); i++) {
      double prod = 1;
      for (int j = 0; j < i; j++) {
        prod *= (v - x[j]);
      }

      sum += diff[i] * prod;
    }
```

```
    return sum;
  };
}
```

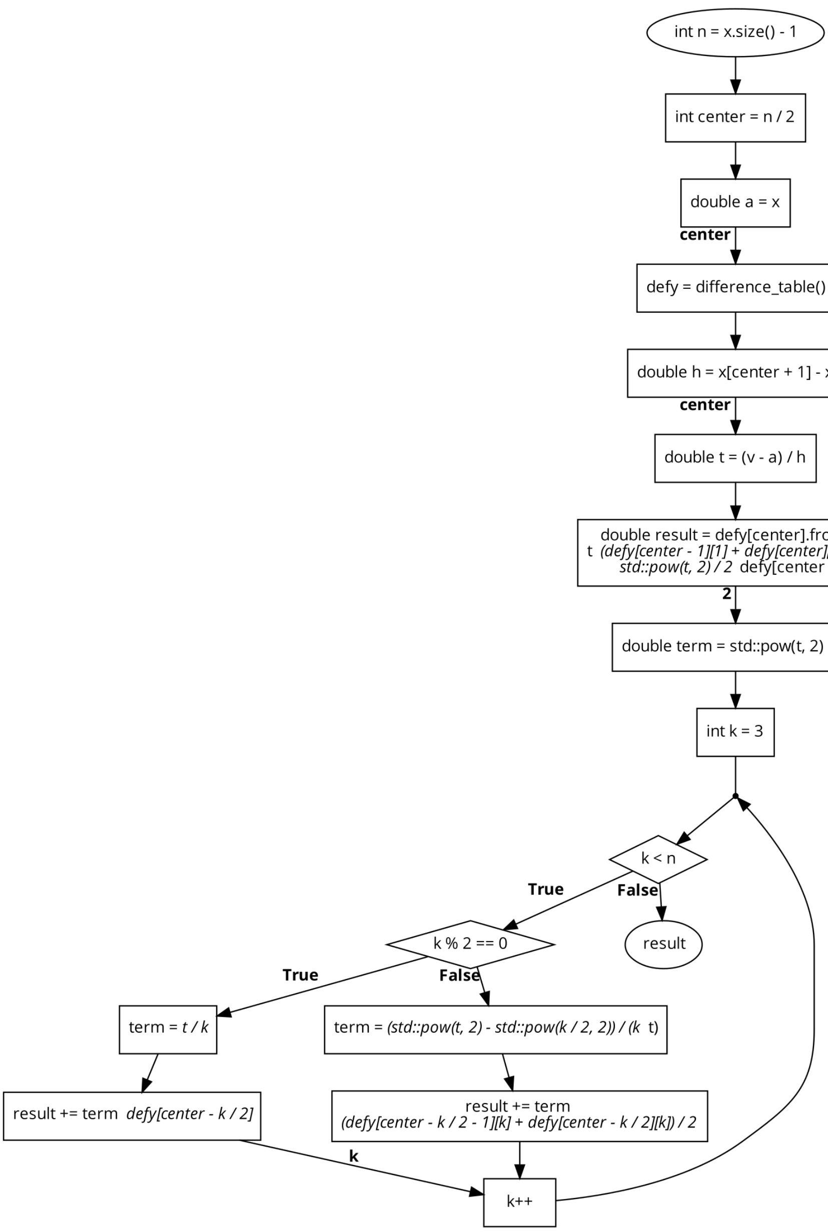



```
/**
 * @brief Calculates the Newton (finite differences) interpolation function.
 * @return Newton (finite differences) interpolation function.
 */
std::function<double(double)> newton_finite() const {
  int n = x.size() - 1;
  std::vector<std::vector<double>> defy = difference_table();

  return [this, n, defy](double v) {
    double result = defy.front().front();
    double term = 1;

    for (int i = 1; i <= n; i++) {
      term *= v - x[i - 1];
      result += term * defy.front()[i] / factorial(i);
    }

    return result;
  };
}
```



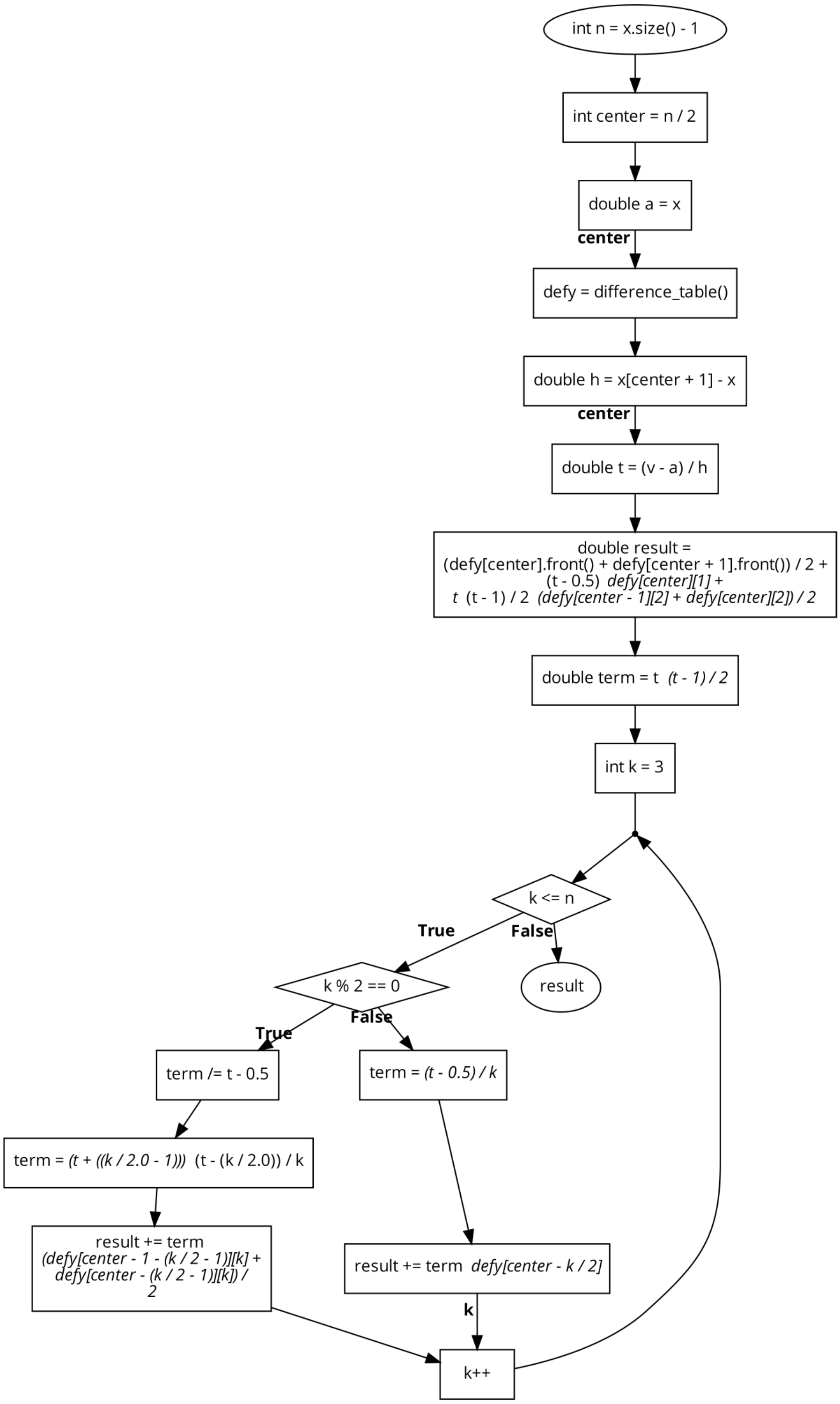

```
/**
 * @brief Calculates the Stirling interpolation function.
 * @return Stirling interpolation function.
 */
std::function<double(double)> stirling() const {
  int n = x.size() - 1;
  int center = n / 2;
  double a = x[center];
  std::vector<std::vector<double>> defy = difference_table();

  return [this, n, center, a, defy](double v) {
    double h = x[center + 1] - x[center];
    double t = (v - a) / h;

    double result = defy[center].front() +
```

```
                t * (defy[center - 1][1] + defy[center][1]) / 2 +
                std::pow(t, 2) / 2 * defy[center - 1][2];
    double term = std::pow(t, 2) / 2;

    for (int k = 3; k < n; k++) {
      if (k % 2 == 0) {
        term *= t / k;
        result += term * defy[center - k / 2][k];
      } else {
        term *= (std::pow(t, 2) - std::pow(k / 2, 2)) / (k * t);
        result += term *
               (defy[center - k / 2 - 1][k] + defy[center - k / 2][k]) / 2;
      }
    }

    return result;
  };
}
```



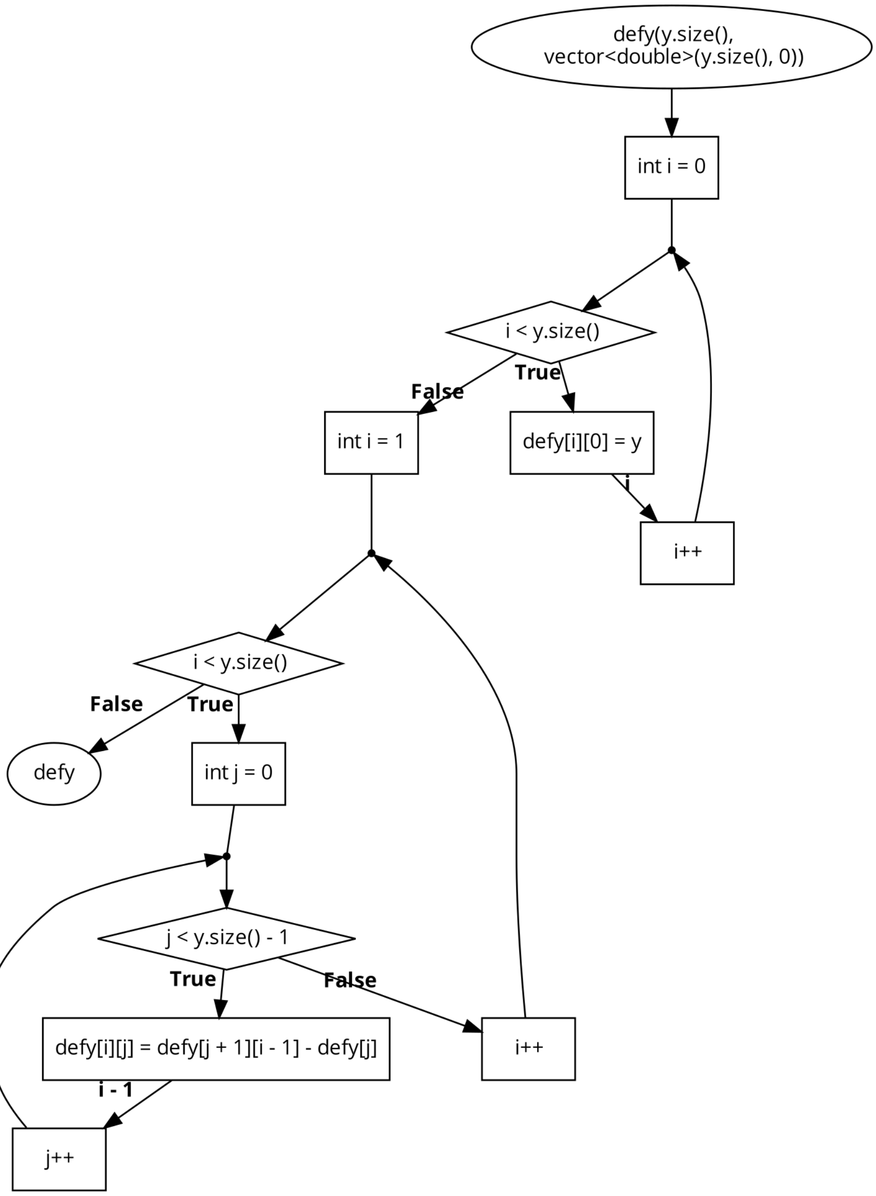

```
/**
```

```
 * @brief Calculates the Bessel interpolation function.
 * @return Bessel interpolation function.
 */
std::function<double(double)> bessel() const {
  int n = x.size() - 1;
  int center = n / 2;
  double a = x[center];

  std::vector<std::vector<double>> defy = difference_table();

  return [this, n, center, a, defy](double v) {
    double h = x[center + 1] - x[center];
    double t = (v - a) / h;
    double result =
        (defy[center].front() + defy[center + 1].front()) / 2 +
        (t - 0.5) * defy[center][1] +
        t * (t - 1) / 2 * (defy[center - 1][2] + defy[center][2]) / 2;
    double term = t * (t - 1) / 2;

    for (int k = 3; k <= n; k++) {
      if (k % 2 == 0) {
        term /= t - 0.5;
        term *= (t + ((k / 2.0 - 1))) * (t - (k / 2.0)) / k;
        result += term *
               (defy[center - 1 - (k / 2 - 1)][k] +
                defy[center - (k / 2 - 1)][k]) /
               2;
      } else {
        term *= (t - 0.5) / k;
        result += term * defy[center - k / 2][k];
      }
    }

    return result;
  };
}
```



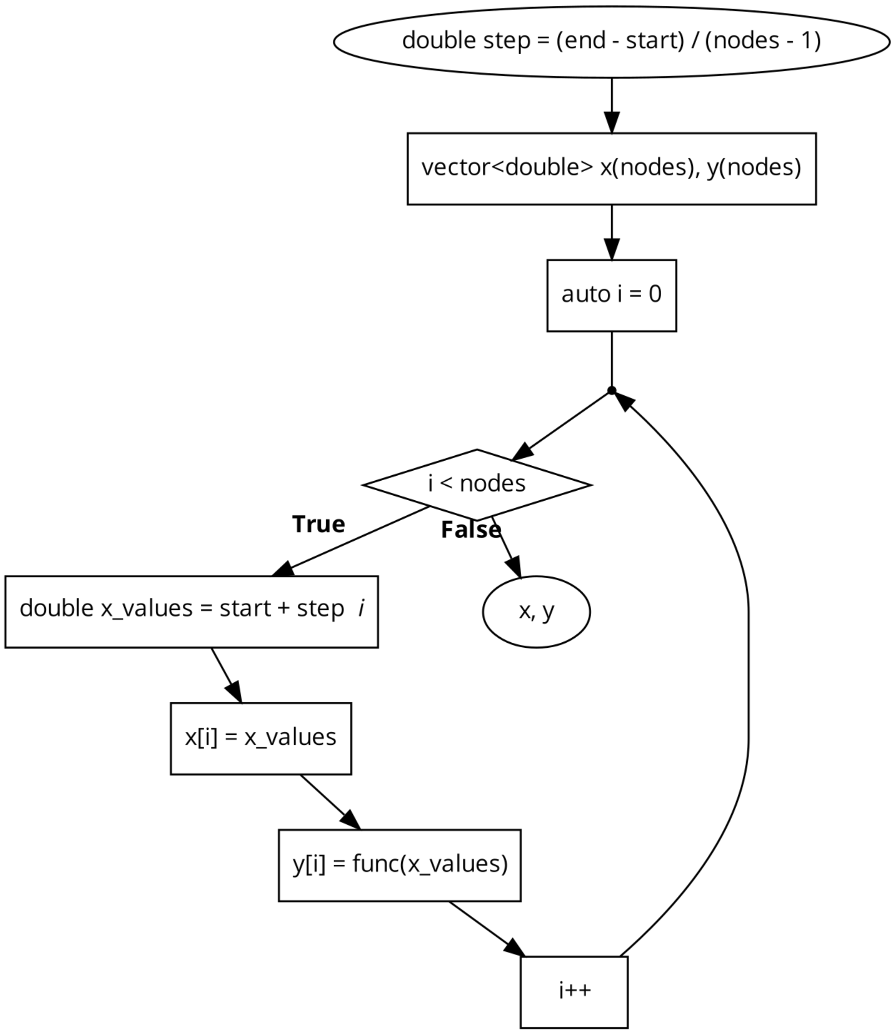

```
/**
 * @brief Calculates the difference table for interpolation.
 * @return Difference table.
 */
std::vector<std::vector<double>> difference_table() const {
  std::vector<std::vector<double>> defy(y.size(),
                                       std::vector<double>(y.size(), 0));

  for (int i = 0; i < y.size(); i++) {
    defy[i][0] = y[i];
  }

  for (int i = 1; i < y.size(); i++) {
```

```
  for (int j = 0; j < y.size() - 1; j++) {
   defy[i][j] = defy[j + 1][i - 1] - defy[j][i - 1];
  }
 }

 return defy;
}
```




```
/**
 * @brief Generates function values for a given function within a range.
 * @param func The function.
 * @param start The start of the range.
 * @param end The end of the range.
 * @param nodes The number of nodes to generate.
 * @return A pair of vectors representing the x and y coordinates of the
 * generated function values.
 */
static std::pair<std::vector<double>, std::vector<double>>
generate_func_values(std::function<double(double)> func, double start,
          double end, int nodes) {
 double step = (end - start) / (nodes - 1);
```

```

  std::vector<double> x(nodes), y(nodes);
  for (auto i = 0; i < nodes; i++) {
   double x_values = start + step * i;
   x[i] = x_values;
   y[i] = func(x_values);
  }
  return {x, y};
}
```



3.2 Ссылка на Github с кодом

GitHub

4 Заключение

При работе были изучены метод интерполяции по точкам.

Список литературы

[1] Слайды с лекций (2023). // Кафедра информатики и вычислительной техники – Малышева Татьяна Алексеевна, к.т.н., доцент.