|  |  |  |
| --- | --- | --- |
|  | P3C2T1#yIS1 |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт кибернетики | | |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры | | |

|  |  |
| --- | --- |
|  | |
| **ЛАБОРАТОРНАЯ РАБОТА № 4** | |
| Вариант № 12 | |
|  | |
| Студент группы: КМБО-02-19 Курса: 3 | *С. А. Минеев* |
| Руководитель практики\_\_\_\_\_\_\_\_\_\_\_\_  (Должность) | *А. А. Липатов* |

|  |  |  |
| --- | --- | --- |
| «Отчет представлен к рассмотрению» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2022 г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись студента)* |
|  |  |  |
| «Отчет утвержден.  Допущен к защите» | «\_\_\_»\_\_\_\_\_\_\_\_\_2022 г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись руководителя)* |

Москва 2022

**Оглавление**

[1 Задание № 1 3](#_Toc99875819)

[1.1 Текст задания 4](#_Toc99875820)

[1.2 Теоретическая часть 4](#_Toc99875821)

[1.3 Практическая часть(Результаты работы программы) 4](#_Toc99875822)

[1.4 Приложение № 1 – Примеры работы программы 4](#_Toc99875823)

[1.5 Приложение № 2 – Листинг программы 4](#_Toc99875824)

[2 Задание № 2 4](#_Toc99875825)

[2.1 Текст задания 4](#_Toc99875826)

[2.2 Теоретическая часть 4](#_Toc99875827)

[2.3 Практическая часть(Результаты работы программы) 4](#_Toc99875828)

[2.4 Приложение № 1 – Примеры работы программы 4](#_Toc99875829)

[2.5 Приложение № 2 – Листинг программы 4](#_Toc99875830)

[3 Задание № 3 4](#_Toc99875831)

[3.1 Текст задания 4](#_Toc99875832)

[3.2 Теоретическая часть 4](#_Toc99875833)

[3.3 Практическая часть(Результаты работы программы) 4](#_Toc99875834)

[3.4 Приложение № 1 – Примеры работы программы 4](#_Toc99875835)

[3.5 Приложение № 2 – Листинг программы 4](#_Toc99875836)

1 Задание № 1

### Текст задания

Используя **первую** ***или*** **вторую** формулы Ньютона, вычислить значения функции, заданной таблично для заданных значений аргумента.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Таблица значений функции для варианта № 12 | | | X | Y | | 0.101 | 1.26183 | | 0.106 | 1.27644 | | 0.111 | 1.29122 | | 0.116 | 1.30617 | | 0.121 | 1.32130 | | 0.126 | 1.33660 | | 0.131 | 1.35207 | | 0.136 | 1.36773 | | 0.141 | 1.38357 | | 0.146 | 1.39959 | | 0.151 | 1.41579 | | 0.156 | 1.42683 | | 0.161 | 1.43356 | | |  |  | | --- | --- | | №  Вар. | Таблица значений аргумента | | 12 | *x =* 0.1485-0.002\**n*  *n*  - Номер варианта. | |

### Теоретическая часть

##### 1.2.1 Рассмотрим случай, когда для определённой функции, непрерывной на некотором отрезке [*a, b*] известен некоторый набор(совокупность) *n + 1* аргументов и соответственным им значения функции :

Особенностью в данном задании является тот факт, что аргументы , которые называются ***интерполяционными/аппроксимирующими*** ***узлами*** функции , находятся на некотором ***фиксированном*** расстояние друг от друга, т. е.

Рис. 1 – Иллюстрация расположения равноотстоящих интерполяционных узлов.

Если взять ***упорядоченный*** набор аргументов и пронумеровать по индексу , то зная начальный элемент набора можно однозначно определить

Рассмотрим далее понятие *аппроксимации* и *интерполяции* функций.

##### 1.2.2 Во-первых, согласно аппроксимирующей теореме Вейерштрасса [1885г.] для любой ***непрерывной*** на отрезке функции можно подобрать последовательность многочленов, равномерно сходящихся к этой функции на данном отрезке.

Рис. 2 – Иллюстрация

Именно на основе этой аппроксимирующей теоремы Вейштрасса в математике в конце 70-x годов зародился метод, связанный с нахождением для сложных математических объектов, таких, например, как функции, более простых объектов, а метод был назван аппроксимацией.

###### **Определение № 1:**

**Аппроксимация** (от лат. ***proxima*** — ближайшая, или приближе́ние) – численный метод, состоящий в замене одних объектов (например, тригонометрических функций) другими, в каком-то смысле близкими к исходным, но более простыми (например, алгебраическими функциями).

##### 1.2.3 Во-вторых, из Рис. 2 видно, что при аппроксимации приходится “платить” точностью вычислений при упрощении функции т. к. аппроксимирующая функция строится не по узлам, а по отрезкам связывающие эти узлы, при этом могут иметься потери в конкретных узлах. В связи с этим были разработаны методы, которые строятся непосредственно по узлам функции и носят название методы интерполирования. Методы интерполирования позволяют учитывать все известные интерполирующие узлы и производить оценку точности вычислений при конкретных значениях функции.

###### **Определение № 2:**

**Интерполирование** (от лат. ***inter–polis*** — «разглаженный, подновлённый, обновлённый; преобразованный») – численный метод, заключающийся в нахождении промежуточных значений величины по имеющемуся дискретному набору известных значений.

###### **Задачи интерполяции:**

• Научиться вычислять значение функции в любой наперед заданной точке.

• Построение алгебраических многочленов(полиномов) степени не выше, чем *n*, принимающих в точках значения .

Интерполяция иногда делится на ***два*** вида:

1)  - собственная *интерполяция*.(Интерполяция в узком смысле.)

2)  - *экстраполяция*. (Интерполяция в широком смысле.)

**Факт № 1:** Термин «*интерполяция*» впервые употребил Джон Валлис в своём трактате «Арифметика бесконечных» (1656).

Стоит отметить, что, как и при аппроксимации, так и при интерполяции, имеется подвох в том, что мы не знаем, как поведет себя функция за пределами отрезка, на котором определена функция. В общем случае получается, что чем дальше аргумент (слева или справа от отрезка, на котором строится полином), тем менее точный результат вычислений получается. Поэтому на практике при интерполяции/аппроксимации делается приоритет тем таблицам функции, которые содержат наиболее точные данные и по возможности полностью покрывающие отрезок, на котором проводятся вычисления функции.

Перейдет теперь к непосредственной теории для интерполяции.

##### **1.2.4 Конечная разность**

Когда интерполирующий многочлен – это полином, то вводится такое понятие как конечная разность.

Рассмотрим по подробнее.

Пусть есть какая-то функция имеется приращение

*Первая конечная разность.*

В случае, когда

для *конкретной точки.*

*.*

*– Вторая конечная разность.*

*В общем случае:*

*Общая формула: =; n - ая конечная разность, n =.*

*Начальная формула:*

*--------------------------------------------------------------------------------------------*

*Пример № 1:*

*Дано:*

*Найти:*

*• Все конечные разности.*

*Решение:*

1. *P3(x+) P3(x)= (x+)3 x3=(x+)3 x3= x3+3x2+3x+1- x3 ==3x2+3x+1. Значение первой конечной разности.*

*- Значение второй конечной разности.*

4)

5)

*Ответ:*

*• 3x2+3x+1; Значение* ***первой*** *конечной разности.*

*• Значение* ***второй*** *конечной разности.*

*• Значение* ***третьей*** *конечной разности.*

*•*

*-----------------------------------------------------------------------*

*f – функция.*

###### ***Свойства***

###### ***Свойства f:***

1. *f(x +) = f(x)+ f(x);*
2. *f(x +) =(1+)• f(x);*
3. *n – раз:*

*f(x +n •) = f(x);*

*f(x +n •) =*

##### **1.2.5 Таблицы конечных разностей**

Пусть нам известна некоторая функция и упорядоченная совокупность пар {(, )} :

Что мы знаем ?

Если – равноотстоящие точки, то мы можем вести понятие

Эти формулы дают нам возможность построение конечных разностей в виде двух видов таблиц конечных разностей:

• **Горизонтальные таблицы разностей.**

• **Диагональные таблицы разностей.**

Рассмотрим каждую из таблиц по подробнее.

###### 1.2.1.5.1 Горизонтальные таблицы разностей

Общий вид горизонтальной таблицы конечных разностей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |

Конкретный вид горизонтальной таблицы конечных разностей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* | *= C* |
|  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |

**Пример № 2:**

Дано:

•

Задание:

*• Составить горизонтальную таблицу конечных разностей.*

*Ответ:*

* *Горизонтальная таблица:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | *=* () – ()= 3 |  |  |  |
|  |  | *=* () – ()= 11 |  |  |  |
|  |  | *=* () – ()= 31 |  |  |  |
|  |  | *=* () – ()= 63 |  |  |  |
|  |  | *=* () – ()= 107 |  |  |  |
|  |  |  |  |  |  |

###### 1.2.1.5.2 Диагональные таблицы разностей

Общий вид диагональной таблицы конечных разностей.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |
|  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |
|  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |

Конкретный вид диагональной таблицы конечных разностей.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |
|  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |
|  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* | *= C* |
|  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |

**Пример № 3:**

Дано:

•

Задание:

*• Составить горизонтальную таблицу конечных разностей.*

*Ответ:*

* *Горизонтальная таблица:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | *=* () – ()= 3 |  |  |  |
|  |  | *=* () – ()= 11 |  |  |  |
|  |  | *=* () – ()= 31 |  |  |  |
|  |  | *=* () – ()= 63 |  |  |  |
|  |  | *=* () – ()= 107 |  |  |  |
|  |  |  |  |  |  |

* *Диагональная таблица:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | *=* () – ()= 3 |  |  |  |
|  |  |  |  |  |  |
|  |  | *=* () – ()= 11 |  |  |  |
|  |  |  |  |  |  |
|  |  | *=* () – ()= 31 |  |  |  |
|  |  |  |  |  |  |
|  |  | *=* () – ()= 63 |  |  |  |
|  |  |  |  |  |  |
|  |  | *=* () – ()= 107 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

##### **1.2.6 Обобщенная степень.**

Пусть имеется:

обобщенная степень,

число,

фиксированный шаг;

**Определение № 4:**

**Обобщенной степенью** порядка *n* для числа *х* называется произведение *n* сомножителей, первый из которых равен *х*, а каждый следующий на шаг *h* меньше предыдущего.

Полагают, что:

= 1;

• Если *h* = 0 :

Обобщенная степень **совпадает** с обычной степенью, т. е. = ;

• Если *h* = 1 :

то мы получаем факториал, т. е. = ;

Положим = *h*;

Тогда:

…

Причем, если *k > n*, то такая конечная разность – это есть 0. .

##### **1.2.7 Интерполяционные формулы Ньютона.**

###### 1.2.1.7.1 I Интерполяционная формула Ньютона.[I И. Ф. Н.]

Имеем:

• Некоторая функция *f(x),* известны узлы

•

*•*

*•*

(1)

Это условие можно записать эквивалентным выражением в конечных разностях:

(2)

• Надо найти коэффициенты:

• Положим, что *х =* :

• Получим:

• Положим, что *х = :*

; →

…

• Подставим (3) в формулу (2):

• Получим:

Первая интерполяционная формула Ньютона.

В целом такие расчеты довольно трудоёмки для облегчения использует новую переменную:

И полином принимает вид:

*- Первая формула ньютона, которую используют для расчетов.*

*-----------------------------------------------------------------------------------------------------*

Пример № 2:

*Дано:*

*Задание:*

• Построить интерполирующий многочлен.

*Решение:*

1. Если мы построим на формулу (4), то у нас здесь присутствуют конечные разности. Если говорить по правильному, то степень многочлена должна быть такой сколько у нас конечных разностей. Степень многочлена *n* должна совпадать с тем количеством точек, которые у нас присутствуют точке в таблицы. Но зачастую это не оправдано. И в общем случае делают многочлен с той степенью сколько у нас конечных разностей. Например, если у нас получается многочлен с тремя конечными разностями, то и будет у нас полином третьей степени. Количество суммируемых коэффициентов в многочлене определяется из таблицы разностей. Поэтому сначала строим таблицу конечных разностей.
2. Построение таблицы разностей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | *= 1.698* |  |  | *n = 3* |
|  |  | *=* 1.785 |  |  | *-* |
|  |  | *=* 1.877 |  |  | *-* |
|  |  | *=*1.972 |  |  | *-* |
|  |  |  |  | *-* | *-* |

Посчитаем значение в точке *x* = 3 ∉ [3.5 ; 3.7];

Получим значение приближённое.

При сравнении с данными из таблицы имеем:

3 < 3.5; →

*Ответ:*

**Факт № 2:**

• Эти интерполяционные формулы позволяют вне диапазона этих интерполяционных узлов. Чем дальше от таблицы разностей, тем менее точное значение мы можем получить.

###### 1.2.1.7.2 II Интерполяционная формула Ньютона. [II И. Ф. Н.]

В целом I И. Ф. Н. нужна в основном в том случае, когда надо интерполировать в начале таблицы конечных разностей. Поскольку мы постоянно берем из таблицы то она уже не очень хорошо применима т. к. у нас нет в конце таблицы конечных разностей и в этом случае нужно до заполнять таблицу неизвестными конечными разностями, что не очень удобно. Поэтому была выведена вторая интерполяционная формула Ньютона, как раз для интерполирования в конце таблицы.

Рассмотрим метод интерполирования по II И. Ф. Н.

Имеется набор значений

Имеется шаг: *h*;

Выпишем интерполяционный многочлен в виде:

*(2)*

*;*

*Положим x = ;*

*; =>*

*Положим x =*

*=>*

*…*

*В итоге имеем:*

Сделаем замену:

Получим:

* *Вторая интерполяционная формула Ньютона.*

*---------------------------------------------------------------------------------------------------*

*Пример № 29:*

*Дано:*

*Задание:*

*• Найти: lg(1044) - ?*

*Решение:*

1. *Выписываем конечные разности.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *C = 0* |  |
|  |  |  | *0* | *…* |  |
|  |  | 0,0043 | *-0,001~0* | *…* | *…* |
|  |  | 0,042 | *0* | *…* | *-* |
|  |  | 0,042 | *0* | *-* | *-* |
|  |  | 0,042 | *-* | *-* | *-* |
|  |  | *-* | *-* | *-* | *-* |
|  |  |  |  | *…* |  |

*n = 2;*

*Ответ:*

### Практическая часть(Результаты работы программы)

**Принятые данные :**  
**A :**

|  |  |
| --- | --- |
| x | y |
| 0.101 | 1.26183 |
| 0.106 | 1.27644 |
| 0.111 | 1.29122 |
| 0.116 | 1.30617 |
| 0.121 | 1.3213 |
| 0.126 | 1.3366 |
| 0.131 | 1.35207 |
| 0.136 | 1.36773 |
| 0.141 | 1.38357 |
| 0.146 | 1.39959 |
| 0.151 | 1.41579 |
| 0.156 | 1.42683 |
| 0.161 | 1.43356 |

**X\_0 :**

|  |
| --- |
| 0.1245 |

**Точность:** = 0.00001

**Вычисления :**  
**newVec**

|  |  |
| --- | --- |
| 0.121000 | 1.321300 |
| 0.126000 | 1.336600 |
| 0.131000 | 1.352070 |
| 0.136000 | 1.367730 |
| 0.141000 | 1.383570 |
| 0.146000 | 1.399590 |
| 0.151000 | 1.415790 |
| 0.156000 | 1.426830 |
| 0.161000 | 1.433560 |

**Вычисления :**  
**newVec :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.121000 | 1.321300 | 0.015300 | 0.000170 | 0.000020 | -0.000030 | 0.000040 | -0.000050 | -0.005280 | 0.027480 |
| 0.126000 | 1.336600 | 0.015470 | 0.000190 | -0.000010 | 0.000010 | -0.000010 | -0.005330 | 0.022200 | 0 |
| 0.131000 | 1.352070 | 0.015660 | 0.000180 | -0.000000 | 0.000000 | -0.005340 | 0.016870 | 0 | 0 |
| 0.136000 | 1.367730 | 0.015840 | 0.000180 | 0.000000 | -0.005340 | 0.011530 | 0 | 0 | 0 |
| 0.141000 | 1.383570 | 0.016020 | 0.000180 | -0.005340 | 0.006190 | 0 | 0 | 0 | 0 |
| 0.146000 | 1.399590 | 0.016200 | -0.005160 | 0.000850 | 0 | 0 | 0 | 0 | 0 |
| 0.151000 | 1.415790 | 0.011040 | -0.004310 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.156000 | 1.426830 | 0.006730 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.161000 | 1.433560 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Предварительное выделение элементов для 1 И. Ф. Н. :**  
**newVec :**

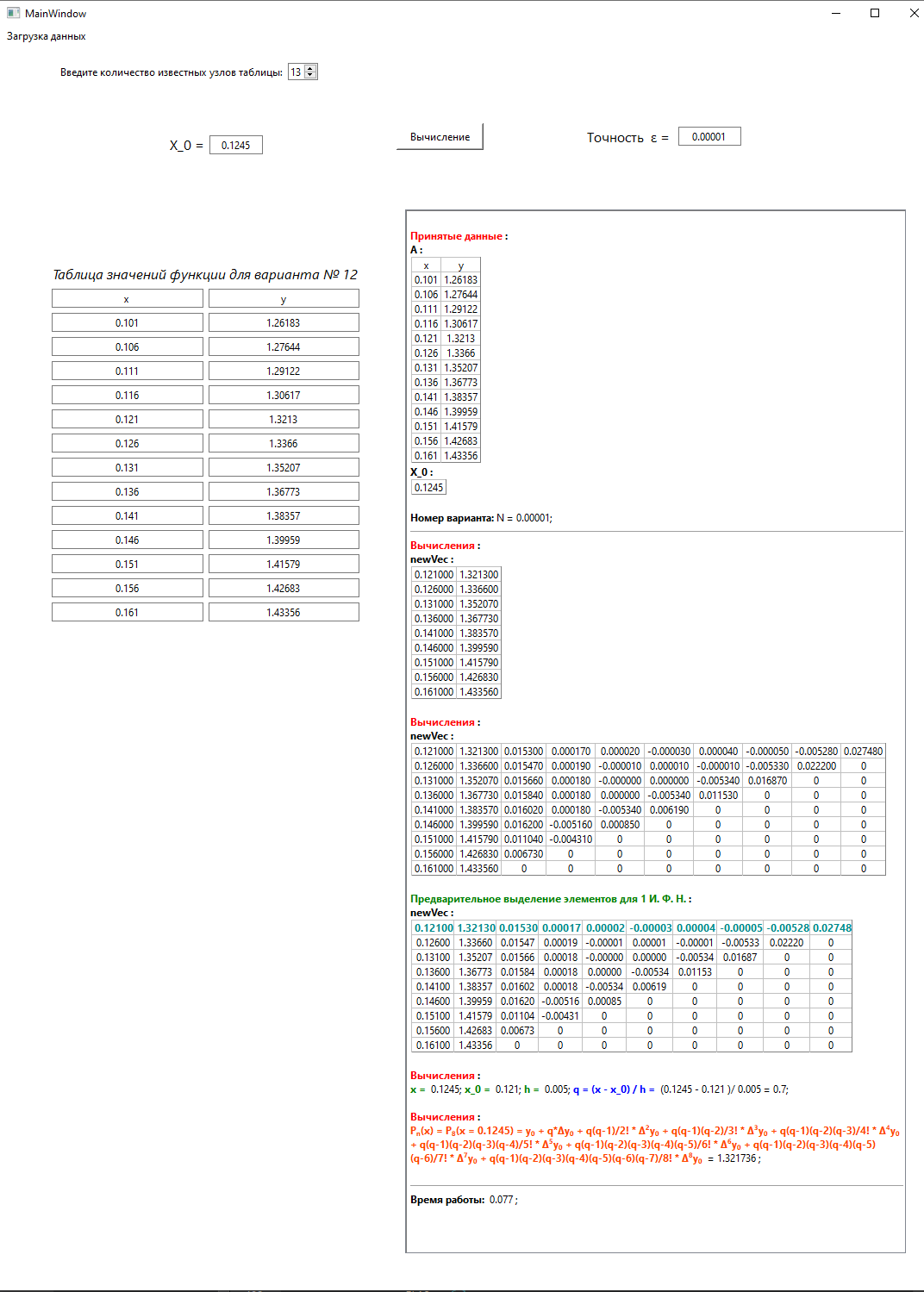
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0.12100** | **1.32130** | **0.01530** | **0.00017** | **0.00002** | **-0.00003** | **0.00004** | **-0.00005** | **-0.00528** | **0.02748** |
| 0.12600 | 1.33660 | 0.01547 | 0.00019 | -0.00001 | 0.00001 | -0.00001 | -0.00533 | 0.02220 | 0 |
| 0.13100 | 1.35207 | 0.01566 | 0.00018 | -0.00000 | 0.00000 | -0.00534 | 0.01687 | 0 | 0 |
| 0.13600 | 1.36773 | 0.01584 | 0.00018 | 0.00000 | -0.00534 | 0.01153 | 0 | 0 | 0 |
| 0.14100 | 1.38357 | 0.01602 | 0.00018 | -0.00534 | 0.00619 | 0 | 0 | 0 | 0 |
| 0.14600 | 1.39959 | 0.01620 | -0.00516 | 0.00085 | 0 | 0 | 0 | 0 | 0 |
| 0.15100 | 1.41579 | 0.01104 | -0.00431 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.15600 | 1.42683 | 0.00673 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.16100 | 1.43356 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Вычисления :**  
**x =** 0.1245; **x\_0 =** 0.121; **h =** 0.005; **q = (x - x\_0) / h =** (0.1245 - 0.121 )/ 0.005 = 0.7;

**Вычисления :**  
**Pn(x) = P8(x = 0.1245) = y0 + q\*Δy0 + q(q-1)/2! \* Δ2y0 + q(q-1)(q-2)/3! \* Δ3y0 + q(q-1)(q-2)(q-3)/4! \* Δ4y0 + q(q-1)(q-2)(q-3)(q-4)/5! \* Δ5y0 + q(q-1)(q-2)(q-3)(q-4)(q-5)/6! \* Δ6y0 + q(q-1)(q-2)(q-3)(q-4)(q-5)(q-6)/7! \* Δ7y0 + q(q-1)(q-2)(q-3)(q-4)(q-5)(q-6)(q-7)/8! \* Δ8y0** = 1.321736 ;

**Время работы:** 0.146 ;

### 1.4 Приложение № 1 – Примеры работы программы



### 1.5 Приложение № 2 – Листинг программы

*//* *Определение* *границ* *элемента* *x* *в* *матрицы* *А* *относительно* *строк.*

int Ga, Gb;

double step\_a, step\_b;

*for*(size\_t i = 0; i < CN; i++){

*if*(i == 0){

step\_a = abs(A[i][0] - B[0]);

Ga = i;

}*else* *if* ( (step\_a > (A[i][0] - B[0])) && (A[i][0] <= B[0])){

step\_a = abs(A[i][0] - B[0]);

Ga = i;

}

*if*(i == 0){

step\_b = abs(A[(CN-1)][0] - B[0]);

Gb = (CN - 1);

}*else* *if* ( (step\_b > (A[(CN-1) - i][0] - B[0])) && (A[(CN-1) - i][0] >= B[0])){

*//qDebug()* *<<* *step\_b;*

step\_b = abs(A[(CN-1) - i][0] - B[0]);

Gb = (CN-1) - i;

}

}

*//* *Метод* *вывода* *данных* *из* *матриц.*

*auto* print\_general = [&](QVector<QVector<double>>& vec) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table border=1 style=\"border-collapse: collapse;\">");

foreach(*auto* str, vec){

strHtml\_A.append("<tr>");

foreach(*auto* elem, str){

elem == 0 ? strHtml\_A.append("<td align=center> " + QString::number(elem) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(elem,'f',6) + " </td>");

}

strHtml\_A.append("</tr>");

}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*switch*(Gb - Ga){

*case*(1):

*while*(Ga != 0){ A.removeFirst(); Ga--; Gb--; CN--;}

qDebug() << Ga;

qDebug() << Gb;

*break*;

}

print\_general(A);

*auto* print\_general\_N\_1 = [&](QVector<QVector<double>>& vec, size\_t Ga, QVector<double>& c) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: green;\">Предварительное выделение элементов для 1 И. Ф. Г. </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table border=1 style=\"border-collapse: collapse;\">");

*for*(size\_t i = 0; i < CN; i++){

strHtml\_A.append("<tr>");

*for*(size\_t j = 0; j < vec[i].count(); j++){

*if*(i == Ga){

vec[i][j] == 0 ? strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',0) + "</b></td>")

: strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',5) + "</b></td>");

c.push\_front(vec[i][j]);

}*else*{

vec[i][j] == 0 ? strHtml\_A.append("<td align=center> " + QString::number(vec[i][j]) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(vec[i][j],'f',5) + " </td>");

}

}

strHtml\_A.append("</tr>");

}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*auto* general\_i = [&]() -> void {

size\_t n = 3;

*auto* FL = [CN = CN] (QVector<QVector<double>> NewVec, size\_t N, double EPS) -> bool {

bool FLAG = *false*;

*for*(size\_t i = 0; i < CN; i++){

FLAG = (abs(NewVec[i][N-1]) > EPS) ? (*true*) : ((FLAG == *true*) ? (*true*) : (*false*));

}

*return* FLAG;

};

uint64\_t COUNT\_R = 0;

*auto* general = [&,CN = CN, print\_general = print\_general, eps = eps, FL = FL, ui=ui, B=B, print\_general\_N\_1=print\_general\_N\_1](*auto* GEN, int64\_t N, QVector<QVector<double>> A) -> void {

QVector<QVector<double>> newVec(CN,QVector<double>(N, 0));

*for*(size\_t i = 0; i < CN; i++){

*for*(size\_t j = 0; j < A[i].count(); j++){

newVec[i][j] = A[i][j];

*if*(j == 1){

uint64\_t COUNT = 0;

std::string number = std::to\_string(A[i][j]);*//A[i][j]*

number = number.substr(0, number.find\_last\_not\_of('0')); *//* *Убираем* *последние* *нули*

size\_t dotFound;

std::stoi(number, *&dotFound*);

COUNT = std::string(number).substr(dotFound).size();

*if*(COUNT > COUNT\_R){

COUNT\_R = COUNT;

}

}

}

}

bool FLAG = *true*;

*for*(size\_t i = 0; i < CN - (N - 2); i++){

newVec[i][N-1] = (double)(newVec[i+1][N - 2] - newVec[i][N - 2]);

uint64\_t COUNT = 0;

std::string number = std::to\_string(newVec[i][N - 1]);

number = number.substr(0, number.find\_last\_not\_of('0')); *//* *Убираем* *последние* *нули;*

size\_t dotFound;

std::stoi(number, *&dotFound*);

COUNT = std::string(number).substr(dotFound).size();

double x = newVec[i][N - 1] - floor(newVec[i][N - 1]);

double fractpart, *//* *дробная* *часть*

intpart; *//* *целая* *часть*

fractpart = modf( newVec[i][N - 1], *&intpart*);

fractpart = QString::number(fractpart, 'f', COUNT\_R).toDouble();

double f = QString::number(fractpart, 'f', COUNT\_R - 1).toDouble();

double f\_i = fractpart - f;

*if*(fractpart - f\_i != 0){

FLAG = *false*;

}

}

*if*( FL(newVec, N, eps) && (N <= CN) && !FLAG){

GEN(GEN, N + 1, newVec);

}*else*{

print\_general(newVec);*//* *print* *one,* *end* *iteration;*

print\_general\_N\_1(newVec, Ga, X);

QString strHtml;

strHtml.clear();

strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

strHtml.append("<b style=\"color: green;\" >x = </b> "+ QString::number(B[0]) + "; ");

strHtml.append("<b style=\"color: green;\" >x\_0 = </b> "+ QString::number(newVec[0][0]) + "; ");

strHtml.append("<b style=\"color: green;\" >h = </b> "+ QString::number(newVec[1][0] - newVec[0][0]) + "; ");

double q = (B[0] - newVec[0][0]) / (newVec[1][0] - newVec[0][0]);

strHtml.append("<b style=\"color: blue;\" >q = (x - x\_0) / h = </b> (" + QString::number(B[0]) + " - " + QString::number(newVec[0][0]) + " )/ " + QString::number(newVec[1][0] - newVec[0][0])

+ " = " + QString::number(q) + "; ");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>");

strHtml.clear();

strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

*auto* P\_n = [&]() -> double {

*auto* ITER = [&](*auto* iter, size\_t i) -> double {

*if*(i >= newVec[0].count() - 2*/\*COLUMN:* *x,* *y;\*/* ){

*return* 0;

}

*auto* factorial = [](*auto* fact, size\_t i) -> double {

*return* (i == 1 || i == 0) ? i*/\*База* *факториала\*/* : i\*fact(fact, i - 1);

};

*auto* Q = [&](*auto* Q, size\_t i) -> double {

*if* (i == 0){

*return* q;

}*else* *if*(i < 0){

*return* 1;

}*else*{

*return* (q-i)\*Q(Q, i-1);

}

};

*return* newVec[0][2*/\*COLUMN:* *x,* *y;\*/* + i]\*Q(Q,i-1) / factorial(factorial, i) + iter(iter, i + 1);

};

*return* newVec[0][1] + ITER(ITER, 1);

};

*switch*(N - 2*/\*COLUMN:* *x,* *y;\*/*){

*case*(8):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[0]) + ") = y<sub>0</sub> "

"+ q\*Δy<sub>0</sub> + q(q-1)/2! \* Δ<sup>2</sup>y<sub>0</sub> + q(q-1)(q-2)/3! \* Δ<sup>3</sup>y<sub>0</sub> + q(q-1)(q-2)(q-3)/4! \* Δ<sup>4</sup>y<sub>0</sub> "

"+ q(q-1)(q-2)(q-3)(q-4)/5! \* Δ<sup>5</sup>y<sub>0</sub> + q(q-1)(q-2)(q-3)(q-4)(q-5)/6! \* Δ<sup>6</sup>y<sub>0</sub> "

"+ q(q-1)(q-2)(q-3)(q-4)(q-5)(q-6)/7! \* Δ<sup>7</sup>y<sub>0</sub> + q(q-1)(q-2)(q-3)(q-4)(q-5)(q-6)(q-7)/8! \* Δ<sup>8</sup>y<sub>0</sub>"

" </b> = " + QString::number(P\_n(), 'f', 6) + " ; ");

*break*;

*case*(2):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[0]) + ") = y<sub>0</sub> "

"+ q\*Δy<sub>0</sub> + q(q-1)/2! \* Δ<sup>2</sup>y<sub>0</sub> "

" </b> = " + QString::number(P\_n(), 'f', 6) + " ; ");

*break*;

*case*(1):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[0]) + ") = y<sub>0</sub> "

"+ q\*Δy<sub>0</sub> </b> = " + QString::number(P\_n(), 'f', 6) + " ; ");

*break*;

*default*:

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[0]) + ") = y<sub>0</sub> "

"+ q\*Δy<sub>0</sub> </b> = " + QString::number(P\_n(), 'f', 6) + " ; ");

*break*;

};

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>");

}

};

general(general, n, A);

};

general\_i();

Ссылка на git-Hub: <https://github.com/MineevS/CHM_3_6.git>

1. Задание № 2

### Текст задания

Используя интерполяционные формулы ***Гаусса***, ***Стирлинга*** и ***Бесселя***, вычислить приближенные значения функции при заданных значениях аргумента, если исходные значения функции представлены таблицей.:

Таблица значений функции .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | Y | X | Y | X | Y |
| 1.50 | 15.132 | 1.75 | 32.812 | 2.00 | 59.158 |
| 1.55 | 17.422 | 1.80 | 37.857 | 2.05 | 64.817 |
| 1.60 | 20.393 | 1.85 | 43.189 | 2.10 | 69.550 |
| 1.65 | 23.994 | 1.90 | 48.689 | 2.15 | 74.782 |
| 1.70 | 28.160 | 1.95 | 54.225 | 2.20 | 79.548 |

Значения аргумента для интерполяционных формул.

|  |  |  |
| --- | --- | --- |
| Гаусса | Стирлинга | Бесселя |
|  |  |  |
| N – номер варианта | | |
| = 1.6+0.006\*12 =  = 1.672 | = 1.749 |  |

### Теоретическая часть

#### Особенности использования интерполяционных формул Ньютона

* Условия использования Интерполяционных формул Ньютона:
* Если , x ***близко*** к , то Используем I ИФН.
* Если >, ***близко*** к , Используем II ИФН.

В частных случаях может возникнуть ситуация, когда мы выходим за пределы таблицы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
| -- | - | - | - | - | - |
|  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |
| -- | - | - | - | - | - |

(--) - находимся близко к значениям таблицы.

Если аргумент функции находится близко к таблице конечных разностей по первому столбцу, то значения И. Ф. Н. будут достаточно точными. Но чем дальше от таблицы требуется найти значение, тем менее точное получается значение для заданной функции. В этом случае прибегают к методу экстраполирования. Процесс нахождения значений полинома за пределами заданного интервала ***экстраполированием***.

**Определение № 4:**

***Экстраполирование(Экстраполяция)*** –распространение результатов, полученных из наблюдений над одной частью некоторого явления, на другую его часть.

**Экстраполяция функции** – продолжение функции за пределы её области определения, при котором продолженная функция (как правило, аналитическая) принадлежит заданному классу функций.

В случае, когда уходим в бесконечность от значений таблицы, то значения интерполяционного многочлена распадается.

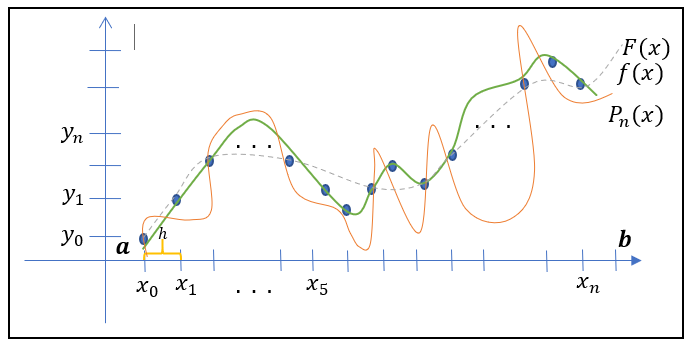


Рис. 2 – Иллюстрация

**Итог:** Интерполяционные формулы Ньютона очень удобно использовать, когда значения ищутся либо в начале таблицы, либо в конце.

В случае, когда надо искать значения где-то в середине таблицы, то использование интерполяционных формул Ньютона не очень подходит, хотя и позволяет найти значение с определенной точностью. Но в целом для этого случая имеются другие интерполяционные формулы, которые делают это более быстро и более просто. Для того, чтобы их рассмотреть нужно ввести такое понятие как таблица центральных разностей.

#### 2.2.2 Таблицы центральных разностей.

Часто, когда интерполирование происходит где-то в центре таблицы конечных разностей, удобно использовать не первые и последние значения, а близкие к тем, которые нам надо получить, но обычно по обе стороны.

Имеем:

• Таблицу конечных разностей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |

• Таблица центральных разностей:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
|  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  | *…* |  |

Появляются отрицательные индексы. Которые помогают получить местоположение узла

, , …;

Формулы для интерполирования на середины: I, II интерполяционные формулы Гаусса, Интерполяционная формула Стирлинга, интерполяционная формула Бесселя.

#### 2.2.3 Интерполяционные формулы Гаусса.

##### 2.2.3.1 I Интерполяционная формула Гаусса. [I И.Ф. Г.]

• Имеется (нечетное) равноотстоящих узлов.

• Имеется функция:

• Известны значения функции в узлах:

• Требуется построить интерполяционный полином , причем этот полином не выше степени .

(Потому что у нас узел).

Отсюда:

Распишем полином:

Предположим, = ;

*…*

Если подставим в полином и введем:

- Первая интерполяционная формула Гаусса.[I И.Ф.Г.]

Первая интерполяционная формула гаусса использует следующие центральные разности:

Интерполяционный многочлен идет ***в вверх*** ***в начало*** таблицы.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  | *…* |  |

##### 2.2.3.2 II Интерполяционная формула Гаусса. [II И.Ф. Г.]

Можно построить ещё одну интерполяционную формулу Гаусса, но на следующих центральных разностях:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |

В отличие от I И.Ф.Г. центральные разности начинаются не с т. е. II И. Ф. Г. еще быстрее приходит в начало таблицы.

Будем иметь:

- Вторая интерполяционная формула Гаусса.

#### 2.2.4 Интерполяционная формула Стирлинга. []

Кроме интерполяционных формул Гаусса, которые все равно требуют интерполирования или подсчета центральных разностей в начале таблицы, поскольку индексы все время уходят в минус, то мы все время приближаемся в начало таблицы. Это не всегда удобно. Для этого существуют еще несколько интерполяционных формул. Одна из них интерполяционная формула Стирлинга. Она представляет собой как среднее арифметическое I и II И. Ф. Г. .

А именно:

Интерполяционная формула Стирлинга.

Интерполяционная формула Стирлинга использует следующие конечные разности:

Таблица выделенных элементов по И. Ф. С.:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |

#### 2.2.5 Интерполяционная формула Бесселя.[И. Ф. Б.]

• И.Ф. Б. берется равноотстоящих узла.

*• h* - шаг.

• Известны значения функции в узлах:

• Требуется построить интерполяционный полином , причем этот полином не выше степени . (Потому, что у нас узел).

Откуда:

Распишем полином:

Возьмем за начальное значение не

Получим:

т. е. в полином подставляем не , а .

Все индексы возрастают на единицу (+1). Получим вспомогательную интерполяционную формулу, которая примет следующий вид:

- Вспомогательный интерполяционный формула Бесселя[В ИФБ].

Если теперь взять среднее арифметическое от предыдущего полинома и вспомогательного, то получим интерполяционный многочлен Бесселя.

Причем, это будет полином, который совпадает с Точек(узлов).

- Интерполяционный полином Бесселя.

Для формулы Бесселя можно заметить, что коэффициенты нечетного порядка имеют коэффициент:

Если

• Интерполяционная формула Бесселя использует следующие конечные разности:

Таблица выделенных элементов по И. Ф. Б.:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |

Так же для общего случая можно тоже упростить формулу, а именно:

Сделаем замену:

- Интерполяционный полином Бесселя. [Упрощенная].

• [Упрощенная] Интерполяционная формула Бесселя использует следующие конечные разности:

Таблица выделенных элементов по И. Ф. Б.:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
|  |  |  |  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  |  |  |  | *…* |  |

#### 2.2.6 Особенности использования интерполяционных формул Стирлинга и Бесселя. + Общая таблица использования Интерполяционных формул для равноотстоящих узлов.

• Если то тогда применяется формула Стирлинга.

• Если то тогда применяется формула Бесселя.

Общая таблица использования Интерполяционных формул:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Формула** | **Условия использования** | **Примечания** |
| 1 | I И. Ф. Н. | Интерполирование в начале таблицы. |  |
| 2 | II И. Ф. Н. | Интерполирование в конце таблицы. |  |
| 3 | I И. Ф. Г. | Используется в И. Ф. C. |  |
| 4 | II И. Ф. Г. | Используется в И. Ф. C. |
| 5 | И. Ф. C. |  |
| 6 | И. Ф. Б. |  |

### Практическая часть(Результаты работы программы)

**Принятые данные :**  
**A :**

|  |  |
| --- | --- |
| x | y |
| 1.50 | 15.132 |
| 1.55 | 17.422 |
| 1.60 | 20.393 |
| 1.65 | 23.994 |
| 1.70 | 28.160 |
| 1.75 | 32.812 |
| 1.80 | 37.857 |
| 1.85 | 43.189 |
| 1.90 | 48.689 |
| 1.95 | 54.225 |
| 2.00 | 59.158 |
| 2.05 | 64.817 |
| 2.10 | 69.550 |
| 2.15 | 74.782 |
| 2.20 | 79.548 |

**X\_G, X\_S, X\_B :**

|  |  |  |
| --- | --- | --- |
| 1.672 | 1.749 | 1.866 |

**Точность :** ε = 0.00001;

**Вычисления :**  
**newVec :**

|  |  |
| --- | --- |
| 1.500 | 15.132 |
| 1.550 | 17.422 |
| 1.600 | 20.393 |
| 1.650 | 23.994 |
| 1.700 | 28.160 |
| 1.750 | 32.812 |
| 1.800 | 37.857 |
| 1.850 | 43.189 |
| 1.900 | 48.689 |
| 1.950 | 54.225 |
| 2.000 | 59.158 |
| 2.050 | 64.817 |
| 2.100 | 69.550 |
| 2.150 | 74.782 |
| 2.200 | 79.548 |

**Вычисления :**  
**newVec :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.500 | 15.132 | 2.290 | 0.681 | -0.051 | -0.014 | -0.000 | 0.000 | 0.001 | -0.003 | 0.006 | -0.504 | 5.454 | -32.669 | 142.642 | -508.333 |
| 1.550 | 17.422 | 2.971 | 0.630 | -0.065 | -0.014 | 0.000 | 0.001 | -0.002 | 0.003 | -0.498 | 4.950 | -27.215 | 109.973 | -365.691 | 0 |
| 1.600 | 20.393 | 3.601 | 0.565 | -0.079 | -0.014 | 0.001 | -0.001 | 0.001 | -0.495 | 4.452 | -22.265 | 82.758 | -255.718 | 0 | 0 |
| 1.650 | 23.994 | 4.166 | 0.486 | -0.093 | -0.013 | 0.000 | -0.000 | -0.494 | 3.957 | -17.813 | 60.493 | -172.960 | 0 | 0 | 0 |
| 1.700 | 28.160 | 4.652 | 0.393 | -0.106 | -0.013 | -0.000 | -0.494 | 3.463 | -13.856 | 42.680 | -112.467 | 0 | 0 | 0 | 0 |
| 1.750 | 32.812 | 5.045 | 0.287 | -0.119 | -0.013 | -0.494 | 2.969 | -10.393 | 28.824 | -69.787 | 0 | 0 | 0 | 0 | 0 |
| 1.800 | 37.857 | 5.332 | 0.168 | -0.132 | -0.507 | 2.475 | -7.424 | 18.431 | -40.963 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.850 | 43.189 | 5.500 | 0.036 | -0.639 | 1.968 | -4.949 | 11.007 | -22.532 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.900 | 48.689 | 5.536 | -0.603 | 1.329 | -2.981 | 6.058 | -11.525 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.950 | 54.225 | 4.933 | 0.726 | -1.652 | 3.077 | -5.467 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.000 | 59.158 | 5.659 | -0.926 | 1.425 | -2.390 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.050 | 64.817 | 4.733 | 0.499 | -0.965 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.100 | 69.550 | 5.232 | -0.466 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.150 | 74.782 | 4.766 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.200 | 79.548 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Предварительное выделение элементов для 1 И. Ф. G. :**  
**newVec :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.500 | 15.132 | 2.290 | 0.681 | -0.051 | -0.014 | -0.000 | **0.000** | **0.001** | -0.003 | 0.006 | -0.504 | 5.454 | -32.669 | 142.642 | -508.333 |
| 1.550 | 17.422 | 2.971 | 0.630 | -0.065 | **-0.014** | **0.000** | 0.001 | -0.002 | 0.003 | -0.498 | 4.950 | -27.215 | 109.973 | -365.691 | 0 |
| 1.600 | 20.393 | 3.601 | **0.565** | **-0.079** | -0.014 | 0.001 | -0.001 | 0.001 | -0.495 | 4.452 | -22.265 | 82.758 | -255.718 | 0 | 0 |
| 1.650 | **23.994** | **4.166** | 0.486 | -0.093 | -0.013 | 0.000 | -0.000 | -0.494 | 3.957 | -17.813 | 60.493 | -172.960 | 0 | 0 | 0 |
| 1.700 | 28.160 | 4.652 | 0.393 | -0.106 | -0.013 | -0.000 | -0.494 | 3.463 | -13.856 | 42.680 | -112.467 | 0 | 0 | 0 | 0 |
| 1.750 | 32.812 | 5.045 | 0.287 | -0.119 | -0.013 | -0.494 | 2.969 | -10.393 | 28.824 | -69.787 | 0 | 0 | 0 | 0 | 0 |
| 1.800 | 37.857 | 5.332 | 0.168 | -0.132 | -0.507 | 2.475 | -7.424 | 18.431 | -40.963 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.850 | 43.189 | 5.500 | 0.036 | -0.639 | 1.968 | -4.949 | 11.007 | -22.532 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.900 | 48.689 | 5.536 | -0.603 | 1.329 | -2.981 | 6.058 | -11.525 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.950 | 54.225 | 4.933 | 0.726 | -1.652 | 3.077 | -5.467 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.000 | 59.158 | 5.659 | -0.926 | 1.425 | -2.390 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.050 | 64.817 | 4.733 | 0.499 | -0.965 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.100 | 69.550 | 5.232 | -0.466 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.150 | 74.782 | 4.766 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.200 | 79.548 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Вычисления :**  
**x =** 1.672; **x\_0 =** 1.65; **h =** 0.05; **q = (x - x\_0) / h =** (1.672 - 1.65 )/ 0.05 = 0.44;

**Вычисления :**  
**Q\_2 :**

|  |  |  |  |
| --- | --- | --- | --- |
| (q + 0) | 0.440 | 4.16600 | 1 |
| (q + 0)(q - 1) | -0.246 | 0.56500 | 2 |
| (q + 0)(q - 1)(q + 2) | -0.355 | -0.07900 | 6 |
| (q + 0)(q - 1)(q + 2)(q - 3) | 0.554 | -0.01400 | 24 |
| (q + 0)(q - 1)(q + 2)(q - 3)(q + 4) | 1.351 | 0.00000 | 120 |
| (q + 0)(q - 1)(q + 2)(q - 3)(q + 4)(q - 5) | -3.457 | 0.00000 | 720 |
| (q + 0)(q - 1)(q + 2)(q - 3)(q + 4)(q - 5)(q + 6) | -11.894 | 0.00100 | 5040 |

**Вычисления :**  
**Pn(x) = P14(x = 1.672) = 25.761779 ;**

**Предварительное выделение элементов для И. Ф. C. :**  
**newVec :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.500 | 15.132 | 2.290 | 0.681 | -0.051 | -0.014 | -0.000 | 0.000 | **0.001** | **-0.003** | **0.006** | -0.504 | 5.454 | -32.669 | 142.642 | -508.333 |
| 1.550 | 17.422 | 2.971 | 0.630 | -0.065 | -0.014 | **0.000** | **0.001** | **-0.002** | 0.003 | -0.498 | 4.950 | -27.215 | 109.973 | -365.691 | 0 |
| 1.600 | 20.393 | 3.601 | 0.565 | **-0.079** | **-0.014** | **0.001** | -0.001 | 0.001 | -0.495 | 4.452 | -22.265 | 82.758 | -255.718 | 0 | 0 |
| 1.650 | 23.994 | **4.166** | **0.486** | **-0.093** | -0.013 | 0.000 | -0.000 | -0.494 | 3.957 | -17.813 | 60.493 | -172.960 | 0 | 0 | 0 |
| **1.700** | **28.160** | **4.652** | 0.393 | -0.106 | -0.013 | -0.000 | -0.494 | 3.463 | -13.856 | 42.680 | -112.467 | 0 | 0 | 0 | 0 |
| 1.750 | 32.812 | 5.045 | 0.287 | -0.119 | -0.013 | -0.494 | 2.969 | -10.393 | 28.824 | -69.787 | 0 | 0 | 0 | 0 | 0 |
| 1.800 | 37.857 | 5.332 | 0.168 | -0.132 | -0.507 | 2.475 | -7.424 | 18.431 | -40.963 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.850 | 43.189 | 5.500 | 0.036 | -0.639 | 1.968 | -4.949 | 11.007 | -22.532 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.900 | 48.689 | 5.536 | -0.603 | 1.329 | -2.981 | 6.058 | -11.525 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.950 | 54.225 | 4.933 | 0.726 | -1.652 | 3.077 | -5.467 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.000 | 59.158 | 5.659 | -0.926 | 1.425 | -2.390 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.050 | 64.817 | 4.733 | 0.499 | -0.965 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.100 | 69.550 | 5.232 | -0.466 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.150 | 74.782 | 4.766 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.200 | 79.548 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Вычисления :**  
**x =** 1.749; **x\_0 =** 1.7; **h =** 0.05; **q = (x - x\_0) / h =** (1.749 - 1.7 )/ 0.05 = 0.98;

**Вычисления :**  
**Q\_3 :**

|  |  |  |  |
| --- | --- | --- | --- |
| q | 0.980 | 4.40900 | 1 |
| q2 | 0.960 | 0.48600 | 2 |
| q(q2 - 12) | -0.039 | -0.08600 | 6 |
| q2(q2 - 12) | -0.038 | -0.01400 | 24 |
| q(q2 - 12)(q2 - 22) | 0.118 | 0.00050 | 120 |
| q2(q2 - 12)(q2 - 22) | 0.116 | 0.00100 | 720 |
| q(q2 - 12)(q2 - 22)(q2 - 32) | -0.948 | -0.00050 | 5040 |
| q2(q2 - 12)(q2 - 22)(q2 - 32) | -0.929 | -0.00300 | 40320 |

**Вычисления :**  
**Pn(x) = P12(x = 1.749) = 32.702115 ;**

**Предварительное выделение элементов для И. Ф. В. :**  
**newVec :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.500 | 15.132 | 2.290 | 0.681 | -0.051 | -0.014 | -0.000 | 0.000 | 0.001 | -0.003 | 0.006 | -0.504 | 5.454 | -32.669 | 142.642 | **-508.333** |
| 1.550 | 17.422 | 2.971 | 0.630 | -0.065 | -0.014 | 0.000 | 0.001 | -0.002 | 0.003 | -0.498 | 4.950 | -27.215 | **109.973** | **-365.691** | **0** |
| 1.600 | 20.393 | 3.601 | 0.565 | -0.079 | -0.014 | 0.001 | -0.001 | 0.001 | -0.495 | 4.452 | **-22.265** | **82.758** | **-255.718** | 0 | 0 |
| 1.650 | 23.994 | 4.166 | 0.486 | -0.093 | -0.013 | 0.000 | -0.000 | -0.494 | **3.957** | **-17.813** | **60.493** | -172.960 | 0 | 0 | 0 |
| 1.700 | 28.160 | 4.652 | 0.393 | -0.106 | -0.013 | -0.000 | **-0.494** | **3.463** | **-13.856** | 42.680 | -112.467 | 0 | 0 | 0 | 0 |
| 1.750 | 32.812 | 5.045 | 0.287 | -0.119 | **-0.013** | **-0.494** | **2.969** | -10.393 | 28.824 | -69.787 | 0 | 0 | 0 | 0 | 0 |
| 1.800 | 37.857 | 5.332 | **0.168** | **-0.132** | **-0.507** | 2.475 | -7.424 | 18.431 | -40.963 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1.850** | **43.189** | **5.500** | **0.036** | -0.639 | 1.968 | -4.949 | 11.007 | -22.532 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.900 | **48.689** | 5.536 | -0.603 | 1.329 | -2.981 | 6.058 | -11.525 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.950 | 54.225 | 4.933 | 0.726 | -1.652 | 3.077 | -5.467 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.000 | 59.158 | 5.659 | -0.926 | 1.425 | -2.390 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.050 | 64.817 | 4.733 | 0.499 | -0.965 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.100 | 69.550 | 5.232 | -0.466 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.150 | 74.782 | 4.766 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2.200 | 79.548 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Вычисления :**  
**x =** 1.866; **x\_0 =** 1.85; **h =** 0.05; **q = (x - x\_0) / h =** (1.866 - 1.85 )/ 0.05 = 0.32;

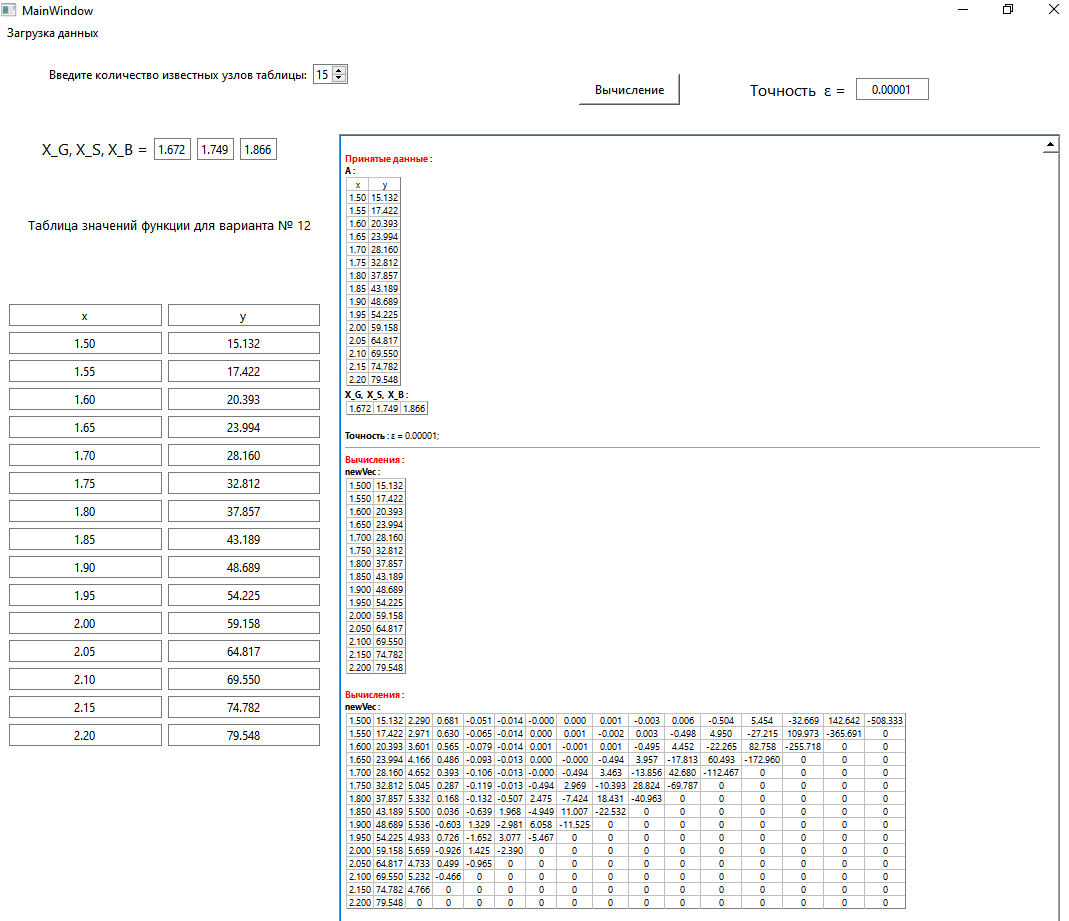
**Вычисления :**  
**Q\_4 :**

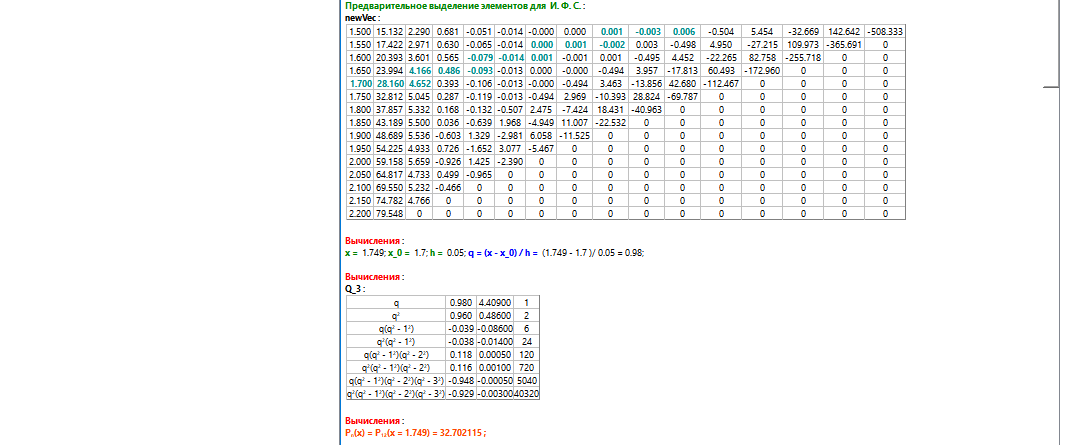
|  |  |  |  |
| --- | --- | --- | --- |
| (q - 0.5) | -0.180 | 5.50000 | 1 |
| (q + 0)(q - 1) | -0.218 | 0.10200 | 2 |
| (q - 0.5)(q + 0)(q - 1) | 0.039 | -0.13200 | 6 |
| (q + 0)(q - 1)(q + 1)(q - 2) | 0.483 | -0.26000 | 24 |
| (q - 0.5)(q + 0)(q - 1)(q + 1)(q - 2) | -0.087 | -0.49400 | 120 |
| (q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3) | -3.000 | 1.23750 | 720 |
| (q - 0.5)(q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3) | 0.540 | 3.46300 | 5040 |
| (q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3)(q + 3)(q - 4) | 36.656 | -4.94950 | 40320 |
| (q - 0.5)(q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3)(q + 3)(q - 4) | -6.598 | -17.81300 | 362880 |
| (q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3)(q + 3)(q - 4)(q + 4)(q - 5) | -741.106 | 19.11400 | 3628800 |
| (q - 0.5)(q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3)(q + 3)(q - 4)(q + 4)(q - 5) | 133.399 | 82.75800 | 39916800 |
| (q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3)(q + 3)(q - 4)(q + 4)(q - 5)(q + 5)(q - 6) | 22394.448 | -72.87250 | 479001600 |
| (q - 0.5)(q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3)(q + 3)(q - 4)(q + 4)(q - 5)(q + 5)(q - 6) | -4031.001 | -365.69100 | 6227020800 |
| (q + 0)(q - 1)(q + 1)(q - 2)(q + 2)(q - 3)(q + 3)(q - 4)(q + 4)(q - 5)(q + 5)(q - 6)(q + 6)(q - 7) | -945439.829 | -254.16650 | 87178291200 |

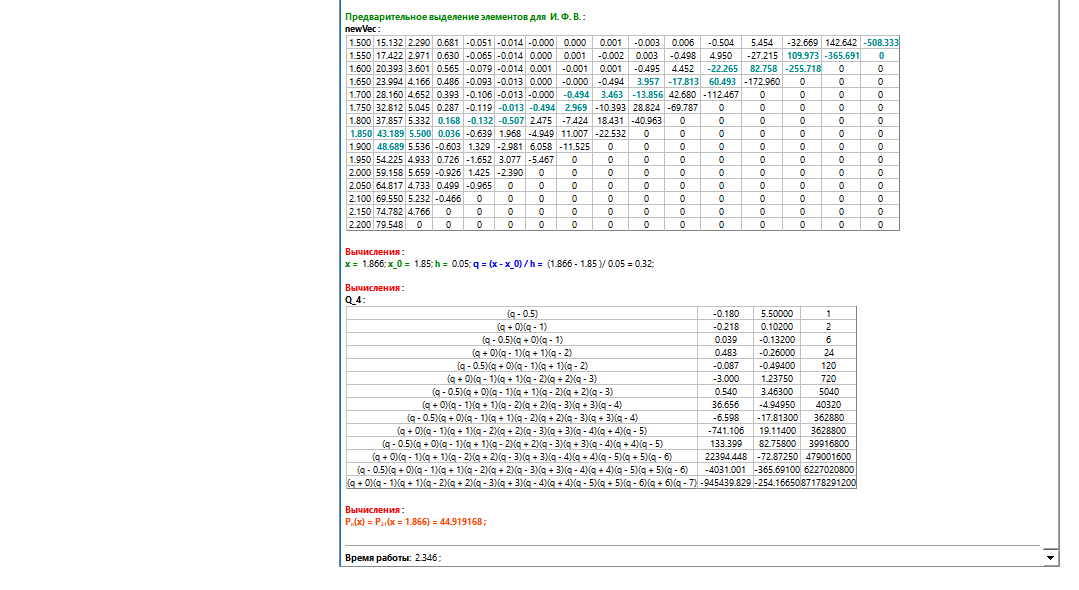
**Вычисления :**  
**Pn(x) = P21(x = 1.866) = 44.919168 ;**

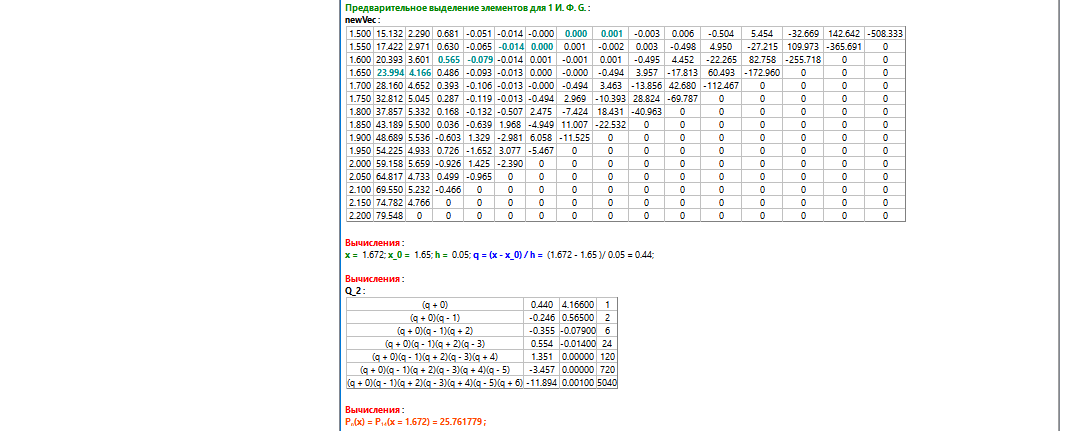
**Время работы:** 3.296 ;

### Приложение № 1 – Примеры работы программы









### Приложение № 2 – Листинг программы

int Ga, Gb;

double step\_a, step\_b;

*auto* print\_general = [&](QVector<QVector<double>>& vec) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table border=1 style=\"border-collapse: collapse;\">");

foreach(*auto* str, vec){

strHtml\_A.append("<tr>");

foreach(*auto* elem, str){

elem == 0 ? strHtml\_A.append("<td align=center> " + QString::number(elem) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(elem,'f',3) + " </td>");

}

strHtml\_A.append("</tr>");

}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*auto* print\_general\_G\_1 = [&](QVector<QVector<double>>& vec, size\_t Ga, QVector<double>& c) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: green;\">Предварительное выделение элементов для 1 И. Ф. Г. </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table border=1 style=\"border-collapse: collapse;\">");

*for*(size\_t i = 0; i < CN; i++){

strHtml\_A.append("<tr>");

*for*(size\_t j = 0; j < vec[i].count(); j++){

*if*(((j == (2\*Ga + 1) - 2\*i || j == (2\*Ga + 2) - 2\*i) && (i <= Ga)) || (i == Ga && (j == 0 || j == 1)) ){

vec[i][j] == 0 ? strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',0) + "</b></td>")

: strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',3) + "</b></td>");

*if* (j != 0 && i == 0){

c.push\_front(vec[i][j]);

}*else* *if*(i > 0 && j != 0){

*static* QVector<double> st;

st.push\_front(vec[i][j]);

*if*(st.count() == 2){

std::copy(st.begin(), st.end(), std::back\_inserter(*c*));

st.clear();

}

}

}*else*{

vec[i][j] == 0 ? strHtml\_A.append("<td align=center> " + QString::number(vec[i][j]) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(vec[i][j],'f',3) + " </td>");

}

}

strHtml\_A.append("</tr>");

}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*auto* print\_general\_S = [&](QVector<QVector<double>>& vec, size\_t Ga, QVector<double>& c) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: green;\">Предварительное выделение элементов для И. Ф. C. </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table border=1 style=\"border-collapse: collapse;\">");

*for*(size\_t i = 0; i < CN; i++){

strHtml\_A.append("<tr>");

*for*(size\_t j = 0; j < vec[i].count(); j++){

*if*((((j % 2 == 0) && (j != 0)) && ( i == (Ga + 1 - 3\*j/2) || i == (2\*Ga + 4 - 3\*j)/2 )) || ((j % 2 == 1) && ( i == (2\*Ga+3\*(1 - j))/2 ))){

vec[i][j] == 0 ? strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',0) + "</b></td>")

: strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',3) + "</b></td>");

*if* (j != 0 && i == 0){ c.push\_back(vec[i][j]);

}*else* *if*(i > 0 && j != 0){ c.push\_back(vec[i][j]);}

}*else*{

(i == Ga && j == 0)

? strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',3) + "</b></td>")

: vec[i][j] == 0

? strHtml\_A.append("<td align=center> " + QString::number(vec[i][j]) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(vec[i][j],'f',3) + " </td>");}}

strHtml\_A.append("</tr>");}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*auto* print\_general\_B\_1 = [&](QVector<QVector<double>>& vec, size\_t Ga, QVector<double>& c) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: green;\">Предварительное выделение элементов для И. Ф. В. </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table border=1 style=\"border-collapse: collapse;\">");

*for*(size\_t i = 0; i < CN; i++){

strHtml\_A.append("<tr>");

*for*(size\_t j = 0; j < vec[i].count(); j++){

*if*( (((j % 2 == 1) && (j != 0)) && ( i == ( (2\*Ga + 3\*(1 - j))/2 ) || i == (2\*Ga + 5 - 3\*j)/2 ))

|| ( ((j % 2 == 0) && (j != 0)) && ( i == ((2\*(Ga + 2) - 3\*j)/2 ) ))){

vec[i][j] == 0 ? strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',0) + "</b></td>")

: strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',3) + "</b></td>");

*if* (j != 0 && i == 0){c.push\_back(vec[i][j]);

}*else* *if*(i > 0 && j != 0){c.push\_back(vec[i][j]);}

}*else*{

(i == Ga && j == 0)

? strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',3) + "</b></td>")

: vec[i][j] == 0

? strHtml\_A.append("<td align=center> " + QString::number(vec[i][j]) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(vec[i][j],'f',3) + " </td>");

}}

strHtml\_A.append("</tr>");} strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*auto* general\_i = [&]() -> void {

size\_t n = 3;

*auto* FL = [CN = CN] (QVector<QVector<double>> NewVec, size\_t N, double EPS) -> bool {

bool FLAG = *false*;

*for*(size\_t i = 0; i < CN; i++){

FLAG = (abs(NewVec[i][N-1]) > EPS) ? (*true*) : ((FLAG == *true*) ? (*true*) : (*false*));

} *return* FLAG;

};

*auto* general = [&](*auto* GEN, int64\_t N, QVector<QVector<double>> A, QVector<double>& C) -> void {

QVector<QVector<double>> newVec(CN,QVector<double>(N, 0));

*for*(size\_t i = 0; i < CN; i++){

*for*(size\_t j = 0; j < A[i].count(); j++){

newVec[i][j]=A[i][j];

}}

*for*(size\_t i = 0; i < CN - (N - 2); i++){

newVec[i][N - 1] = (double)(newVec[i + 1][N - 2] - newVec[i][N - 2]);

}

*if*( FL(newVec, N, eps) && (N <= CN)){

GEN(GEN, N + 1, newVec, C);

}*else*{

print\_general(newVec);*//* *print* *one,* *end* *iteration;*

*//* *I* *И.* *Ф.* *Г.:*

{

*//* *#:*

*for*(size\_t i = 0; i < CN; i++){

*if*(i == 0){ step\_a = abs(A[i][0] - B[0]); Ga = i;

}*else* *if* ( (step\_a > (A[i][0] - B[0])) && (A[i][0] <= B[0])){

step\_a = abs(A[i][0] - B[0]); Ga = i;

}

*if*(i == 0){ step\_b = abs(A[(CN-1)][0] - B[0]); Gb = (CN - 1);

}*else* *if* ( (step\_b > (A[(CN-1) - i][0] - B[0])) && (A[(CN-1) - i][0] >= B[0])){ step\_b = abs(A[(CN-1) - i][0] - B[0]); Gb = (CN-1) - i;}

}

print\_general\_G\_1(newVec, Ga, C);

*if*((!C.empty()) && (C[C.count() - 1] == A[Ga][1])){ QString strHtml; strHtml.clear(); strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

strHtml.append("<b style=\"color: green;\" >x = </b> "+ QString::number(B[0]) + "; ");

strHtml.append("<b style=\"color: green;\" >x\_0 = </b> "+ QString::number(A[Ga][0]) + "; ");

strHtml.append("<b style=\"color: green;\" >h = </b> "+ QString::number(newVec[1][0] - newVec[0][0]) + "; ");

double q = (B[0] - A[Ga][0]) / (newVec[1][0] - newVec[0][0]);

strHtml.append("<b style=\"color: blue;\" >q = (x - x\_0) / h = </b> (" + QString::number(B[0]) + " - "

+ QString::number(A[Ga][0]) + " )/ " + QString::number(newVec[1][0] - newVec[0][0])

+ " = " + QString::number(q) + "; ");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>"); strHtml.clear();

strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

*auto* P\_n = [&]() -> double {

*auto* ITER = [&](*auto* iter, size\_t i) -> double {

*if*(i >= C.count() - 1*/\*COLUMN:* *y;\*/* ) *return* 0;

*auto* factorial = [](*auto* fact, size\_t i) -> double {

*return* (i == 1 || i == 0) ? i*/\*База* *факториала\*/* : i\*fact(fact, i - 1);

};

*auto* Q = [&](*auto* Q, size\_t i) -> double {

*return* (i == 0) ? q

: (i % 2 == 0) ? (q + (i / 2)) \* Q(Q, i - 1)

: (q - ((i + 1) / 2)) \* Q(Q, i - 1);

};

*return* C[C.count() - 1 - i] \* Q(Q, i - 1) / factorial(factorial, i) + iter(iter, i + 1);

};

*return* C[C.count() - 1] + ITER(ITER, 1);

};

*switch*(C.count() - 1*/\*COLUMN:* *y;\*/*){

*case*(3):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *y;\*/*) + "</sub>(x = " + QString::number(B[0]) + ") "

"= " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*case*(2):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[0]) + ") = " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*case*(1):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[0]) + ") = " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*default*:

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *y;\*/*) + "</sub>(x = " + QString::number(B[0]) + ") "

"= " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

};

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>");

}

*//* *#;*

}

*//* *И.* *Ф.* *C.:*

{

*//* *#:*

*for*(size\_t i = 0; i < CN; i++){

*if*(i == 0){

step\_a = abs(A[i][0] - B[1]);

Ga = i;

}*else* *if* ( (step\_a > (A[i][0] - B[1])) && (A[i][0] <= B[1])){

step\_a = abs(A[i][0] - B[1]);

Ga = i;

}

*if*(i == 0){

step\_b = abs(A[(CN-1)][0] - B[1]);

Gb = (CN - 1);

}*else* *if* ( (step\_b > (A[(CN-1) - i][0] - B[1])) && (A[(CN-1) - i][0] > B[1])){

*//qDebug()* *<<* *step\_b;*

step\_b = abs(A[(CN-1) - i][0] - B[1]);

Gb = (CN-1) - i;

}

}

*//qDebug()* *<<* *Ga;*

*//qDebug()* *<<* *Gb;*

print\_general\_S(newVec, Ga, C);

*if*((!C.empty()) && (C[C.count() - 1] == A[Ga][1])){

QString strHtml;

strHtml.clear();

strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

strHtml.append("<b style=\"color: green;\" >x = </b> "+ QString::number(B[1]) + "; ");

strHtml.append("<b style=\"color: green;\" >x\_0 = </b> "+ QString::number(A[Ga][0]) + "; ");

strHtml.append("<b style=\"color: green;\" >h = </b> "+ QString::number(newVec[1][0] - newVec[0][0]) + "; ");

double q = (B[1] - A[Ga][0]) / (newVec[1][0] - newVec[0][0]);

strHtml.append("<b style=\"color: blue;\" >q = (x - x\_0) / h = </b> (" + QString::number(B[1]) + " - "

+ QString::number(A[Ga][0]) + " )/ " + QString::number(newVec[1][0] - newVec[0][0])

+ " = " + QString::number(q) + "; ");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>");

strHtml.clear();

strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

*auto* P\_n = [&]() -> double {

*auto* ITER = [&](*auto* iter, size\_t i) -> double {

*if*(i >= C.count() - 2*/\*COLUMN:* *y;\*/* ) *return* 0;

*auto* factorial = [](*auto* fact, size\_t i) -> double {

*return* (i == 1 || i == 0) ? i*/\*База* *факториала\*/* : i\*fact(fact, i - 1);

};

*static* int STEP = 0;

*auto* Q = [&](*auto* Q, size\_t i) -> double {

int ST = STEP;

double B\_0 = q - STEP;

double B\_1 = q + STEP;

*if*(i == 1){

*return* B\_0;

}*else* *if*(i % 2 == 0){

double V\_1 = B\_1;

double sa = V\_1\*Q(Q, i - 1);

*return* sa;

}*else*{

double V\_0 = B\_0;

STEP++;

*return* V\_0\*Q(Q, i - 1);

}

};

STEP = 0;

*if*(i % 3 != 0){

qDebug() << "i: " << (2\*i + 1)/3 << "STEP: " << STEP << "C\_L: " << C[C.count() - 1 - i] << "C\_R: " << C[C.count() - 2 - i] << ";";

int f = factorial(factorial, (2\*i + 1)/3);

double qr = Q(Q, (2\*i + 1)/3);

qDebug() << "factor(i): " << f << " Q(i): " << QString::number(qr);

STEP = 0;

*return* (C[C.count() - 1 - i] + C[C.count() - 2 - i]) \* Q(Q, (2\*i + 1)/3) / 2\*factorial(factorial, (2\*i + 1)/3) + iter(iter, i + 2);

}*else*{

qDebug() << "i: " << 2\*i/3 << "; STEP:" << STEP << "C: " << C[C.count() - 1 - i] ;

int f = factorial(factorial, (2\*i + 1)/3);

double qr = Q(Q, 2\*i/3);

qDebug() << "factor(i): " << f << " Q(i): " << QString::number(qr);

STEP = 0;

*//qDebug()* *<<* *";* *factor(i):"* *<<* *factorial(factorial,* *(2\*i* *+* *1)/3)* *<<* *";* *Q(i):"* *<<* *Q(Q,* *2\*i/3)* *<<* *";\n";*

*return* (C[C.count() - 1 - i] + 0) \* Q(Q, 2\*i/3) / factorial(factorial, 2\*i/3) + iter(iter, i + 1);

}

*//qDebug()* *<<* *C.count();*

};

*return* C[C.count() - 1] + ITER(ITER, 1);

};

*switch*(C.count() - 1*/\*COLUMN:* *y;\*/*){

*case*(3):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(C.count() - 2*/\*COLUMN:* *y;\*/*) + "</sub>(x = " + QString::number(B[1]) + ") "

"= " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*case*(2):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[1]) + ") = " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*case*(1):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[1]) + ") = " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*default*:

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(C.count() - 2*/\*COLUMN:* *y;\*/*) + "</sub>(x = " + QString::number(B[1]) + ") "

"= " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

};

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>");

}

*//* *#;*

}

*//* *И.* *Ф.* *B.:*

{

*//* *#:*

*for*(size\_t i = 0; i < CN; i++){

*if*(i == 0){

step\_a = abs(A[i][0] - B[2]);

Ga = i;

}*else* *if* ( (step\_a > (A[i][0] - B[2])) && (A[i][0] <= B[2])){

step\_a = abs(A[i][0] - B[2]);

Ga = i;

}

*if*(i == 0){

step\_b = abs(A[(CN-1)][0] - B[2]);

Gb = (CN - 1);

}*else* *if* ( (step\_b > (A[(CN-1) - i][0] - B[2])) && (A[(CN-1) - i][0] > B[2])){

*//qDebug()* *<<* *step\_b;*

step\_b = abs(A[(CN-1) - i][0] - B[2]);

Gb = (CN - 1) - i;

}

}

qDebug() << Ga;

qDebug() << Gb;

print\_general\_B\_1(newVec, Ga, C);

*if*((!C.empty()) && (C[C.count() - 2] == A[Ga][1])){

QString strHtml;

strHtml.clear();

strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

strHtml.append("<b style=\"color: green;\" >x = </b> "+ QString::number(B[2]) + "; ");

strHtml.append("<b style=\"color: green;\" >x\_0 = </b> "+ QString::number(A[Ga][0]) + "; ");

strHtml.append("<b style=\"color: green;\" >h = </b> "+ QString::number(newVec[1][0] - newVec[0][0]) + "; ");

double q = (B[2] - A[Ga][0]) / (newVec[1][0] - newVec[0][0]);

strHtml.append("<b style=\"color: blue;\" >q = (x - x\_0) / h = </b> (" + QString::number(B[2]) + " - "

+ QString::number(A[Ga][0]) + " )/ " + QString::number(newVec[1][0] - newVec[0][0])

+ " = " + QString::number(q) + "; ");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>");

strHtml.clear();

strHtml.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

*auto* P\_n = [&]() -> double {

*auto* ITER = [&](*auto* iter, size\_t i) -> double {

*if*(i >= C.count() - 2*/\*COLUMN:* *y;\*/* ) *return* 0;

*auto* factorial = [](*auto* fact, size\_t i) -> double {

*return* (i == 1 || i == 0) ? i*/\*База* *факториала\*/* : i\*fact(fact, i - 1);

};

*static* int STEP = 0;

*auto* Q = [&](*auto* Q, size\_t i) -> double {

int ST = STEP;

double B\_0 = q - STEP;

double B\_1 = q + STEP;

*if*(i == 1){

*return* B\_0;

}*else* *if*(i % 2 == 0){

double V\_1 = B\_1;

double sa = V\_1\*Q(Q, i - 1);

*return* sa;

}*else*{

double V\_0 = B\_0;

STEP++;

*return* V\_0\*Q(Q, i - 1);

}

};

STEP = 0;

*if*(i % 3 != 0){

qDebug() << "i: " << (2\*i + 1)/3 << "STEP: " << STEP << "C\_L: " << C[C.count() - 1 - i] << "C\_R: " << C[C.count() - 2 - i] << ";";

int f = factorial(factorial, (2\*i + 1)/3);

double qr = Q(Q, (2\*i + 1)/3);

qDebug() << "factor(i): " << f << " Q(i): " << QString::number(qr);

STEP = 0;

*return* (C[C.count() - 1 - i] + C[C.count() - 2 - i]) \* Q(Q, (2\*i + 1)/3) / 2\*factorial(factorial, (2\*i + 1)/3) + iter(iter, i + 2);

}*else*{

qDebug() << "i: " << 2\*i/3 << "; STEP:" << STEP << "C: " << C[C.count() - 1 - i] ;

int f = factorial(factorial, (2\*i + 1)/3);

double qr = Q(Q, 2\*i/3);

qDebug() << "factor(i): " << f << " Q(i): " << QString::number(qr);

STEP = 0;

*return* (C[C.count() - 1 - i] + 0) \* Q(Q, 2\*i/3) / factorial(factorial, 2\*i/3) + iter(iter, i + 1);

}

};

*return* (C[C.count() - 1] + C[C.count() - 2])/2 + ITER(ITER, 2);

};

*switch*(C.count() - 1*/\*COLUMN:* *y;\*/*){

*case*(3):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *y;\*/*) + "</sub>(x = " + QString::number(B[2]) + ") "

"= " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*case*(2):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[2]) + ") = y<sub>0</sub> "

"+ q\*Δy<sub>0</sub> + q(q-1)/2! \* Δ<sup>2</sup>y<sub>0</sub> "

" </b> = " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*case*(1):

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(N - 2*/\*COLUMN:* *x,* *y;\*/*)+ "</sub>(x = " + QString::number(B[2]) + ") = " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

*default*:

strHtml.append("<b style=\"color: orangeRed;\" >P<sub>n</sub>(x) = P<sub>" + QString::number(C.count() - 2*/\*COLUMN:* *y;\*/*) + "</sub>(x = " + QString::number(B[2]) + ") "

"= " + QString::number(P\_n(), 'f', 6) + " ; ");

C.clear();

*break*;

};

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml + "<p></p>");

}

*//* *#;*

}

}

};

general(general, n, A, C);

};

general\_i();

Ссылка на git-Hub: <https://github.com/MineevS/CHM_3_6.git>

1. Задание № 3

### Текст задания

Вычислить значения функции при заданных значениях аргумента, используя ***интерполяционную формулу Ньютона*** для ***не равноотстоящих*** узлов и многочлен ***Лагранжа***.

|  |  |
| --- | --- |
| Таблица значений функции для вариантов 7 – {12} – 13 | |
| X | Y |
| 0.593 | 0.532050 |
| 0.598 | 0.535625 |
| 0.605 | 0.540598 |
| 0.613 | 0.546235 |
| 0.619 | 0.550431 |
| 0.627 | 0.555983 |
| 0.632 | 0.559428 |
| 0.640 | 0.568738 |
| 0.650 | 0.575298 |

Значения аргумента для интерполяционных формул.

|  |  |  |  |
| --- | --- | --- | --- |
| Ньютона с неравноотстоящими узлами | | Многочлена Лагранжа | |
| № вар | Значения аргумента | № вар | Значения аргумента |
| N = 12 | 0.609 + 0.002 \* (от 9 до 3\*N) | N = 12 | 0.629+ 0.003 \* (от 5 до 3\*N) |
| Примечание. Запись «(от В до С) означает, что множитель пробегает значения от В до С с шагом 1. В результате этого получается множество аргументов, для которых надо считать значение интерполируемой функции. | | | |

### Теоретическая часть

#### 3.2.1 Интерполяционные формулы для неравноотстоящих узлов.

Ранее, в задании № 1 были рассмотрены интерполяционные формулы для ***равноотстоящих узлов***, но не изредка приходится встречаться с ситуацией, когда узлы некоторой функции находятся друг от друга не с одним и тем же шагом, а с различными шагами:

В этом случае говорят о неравноотстоящих узлах функции и для вычисления значения в промежуточных аргументах используют ***интерполяционную формулу Ньютона для неравноотстоящих узлов*** и ***интерполяционную формулу Лагранжа***.

#### 3.2.2 Интерполяционная формула Ньютона для неравноотстоящих узлов.

Используя разделённые разности, многочлен Лагранжа можно представить в виде первой интерполяционной формулы Ньютона. Интерполяционная формула Ньютона для функций с не равноотстоящими узлами интерполяции:

— разделённая разность порядка *n*.

Для облегчения расчетов можно воспользоваться таблицей разделённый разностей:

Общий вид таблицы разделенных разностей:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |  |
|  |  |  |  |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  | *…* |  |

#### 3.2.3 Интерполяционная формула Лагранжа для неравноотстоящих узлов.

• Пусть на [*a, b*] дано значение аргумента,

.

• Требуется построить многочлен степени не выше *n*:

() = *, ∀ i* = 0*,* 1*, . . . , n*

• Интерполяционный многочлен Лагранжа:

() =

Интерполяционная формула Лагранжа.

Имеем: на [a, b] задано значение узлов: *= f(), = f(), …, = f().*

Стоит обратить внимание, что здесь не задано шага.

Требуется: построить полином

Рассмотрим полином:

Положим x=

Получим:

Она работает для любых наборов узлов, как ***равноотстоящих***, так и ***неравноотстоящих***. Он имеет единственный вид.

Пример № 31:

Дано: Построить Интерполяционный многочлен Лагранжа на трех узлах.

Функции *y=ln*(*x*).

Решение:

1. = *f*().

= ln() = ln(1) = 0;

= ln() = ln(e) =1;

= ln() = ln() =2;

1. Для подсчёта значения в конкретной точке просто нужно заместо x поставить значение в полином.

#### 3.2.4 Оценка погрешности интерполяционной формулы Лагранжа.

Имеем:

, точку:

=*f*(), =*f*(), …,=*f*();

Нужно построить многочлен Лагранжа как можно ближе к самой функции, чтобы погрешность была минимальной.

Будем считать, что на [*a*,*b*] непрерывна т. е имеет все производные до в узлах : [*a*, *b*].

Введем вспомогательную функцию:

,

Функция *u(x) имеет корень в точках*

• Подберем такой коэффициент , чтобы у появился й корень в какой-то фиксированной точке .

Для того, чтобы существовал такой корень

* *u(x):* корня, и обращается в 0:
* [], [], …, [], [], …, [].

• Для оценки погрешности нужно получить выражение для *k* через производную *u(x).*

• Вспомним, что функция *u(x)* должна иметь *n+1* производную и мы можем сказать, что:

*(x) на будет иметь покрайней мере один корень : ().*

• Теперь рассмотрим производную

Поскольку многочлен имеет n степень, а производная n+1, то такая производная будет равно 0. т. е.

И надо ещё рассмотреть

*Подставляем все в формулу (\*) и получаем:*

*(x) =*;

Соответственно, если вместо x подставить какой-то корень , то значение функции  *(x)* будет равно 0.

*т. е.*

* *()=*
* Формула оценки погрешности сверху,

Получим:

*Пример № 32:*

*Задание:*

*•* Определить, с какой точностью можно посчитать значение функции *: .*

1. Считаем производные:

*;*

*;*

*• посчитаем значение 3 производной в узлах:*

- значение в той точке, в которой считаем.

- Абсолютная погрешность.

### Практическая часть(Результаты работы программы)

**Принятые данные :**  
**A :**

|  |  |
| --- | --- |
| x | y |
| 0.59300 | 0.532050 |
| 0.59800 | 0.535625 |
| 0.60500 | 0.540598 |
| 0.61300 | 0.546235 |
| 0.61900 | 0.550431 |
| 0.62700 | 0.555983 |
| 0.63200 | 0.559428 |
| 0.64000 | 0.568738 |
| 0.65000 | 0.575298 |

**X\_N:**

|  |  |
| --- | --- |
| 9 | 36 |

**X\_L :**

|  |  |
| --- | --- |
| 5 | 36 |

**Точность :** ε = 0.00001;

**Вычисления :**  
**newVec :**

|  |  |
| --- | --- |
| 0.593000 | 0.532050 |
| 0.598000 | 0.535625 |
| 0.605000 | 0.540598 |
| 0.613000 | 0.546235 |
| 0.619000 | 0.550431 |
| 0.627000 | 0.555983 |
| 0.632000 | 0.559428 |
| 0.640000 | 0.568738 |
| 0.650000 | 0.575298 |

**Предварительное выделение элементов для И. Ф. N. с неравноотстающими узлами :**  
**newVec :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | y | [x0 ; x1] | [x0 ; x1 ; x2] | [x0 ; x1 ; x2 ; x3] | [x0 ; x1 ; x2 ; x3 ; x4] | [x0 ; x1 ; x2 ; x3 ; x4 ; x5] | [x0 ; x1 ; x2 ; x3 ; x4 ; x5 ; x6] | [x0 ; x1 ; x2 ; x3 ; x4 ; x5 ; x6 ; x7] | [x0 ; x1 ; x2 ; x3 ; x4 ; x5 ; x6 ; x7 ; x8] |
| 0.593 | **0.532** | **0.715** | **-0.381** | **-0.298** | **27.800** | **-1386.044** | **48507.895** | **940894635.514** | **-88459210557.408** |
| 0.598 | 0.536 | 0.710 | -0.387 | 0.425 | -19.326 | 505.764 | 44270555.765 | -4101280366.258 | 0 |
| 0.605 | 0.541 | 0.705 | -0.378 | -0.135 | -2.130 | 1859869.106 | -168996023.281 | 0 | 0 |
| 0.613 | 0.546 | 0.699 | -0.381 | -0.193 | 65093.289 | -5744951.942 | 0 | 0 | 0 |
| 0.619 | 0.550 | 0.694 | -0.385 | 1757.326 | -147469.933 | 0 | 0 | 0 | 0 |
| 0.627 | 0.556 | 0.689 | 36.519 | -2814.242 | 0 | 0 | 0 | 0 | 0 |
| 0.632 | 0.559 | 1.164 | -28.208 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.640 | 0.569 | 0.656 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.650 | 0.575 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Ответ :**  
**Y :**

|  |  |
| --- | --- |
| x | y |
| 0.627000 | 0.555983 |
| 0.629000 | 0.557314 |
| 0.631000 | 0.558692 |
| 0.633000 | 0.560218 |
| 0.635000 | 0.562031 |
| 0.637000 | 0.564286 |
| 0.639000 | 0.567105 |
| 0.641000 | 0.570503 |
| 0.643000 | 0.574270 |
| 0.645000 | 0.577805 |
| 0.647000 | 0.579890 |
| 0.649000 | 0.578395 |
| 0.651000 | 0.569897 |
| 0.653000 | 0.549203 |
| 0.655000 | 0.508767 |
| 0.657000 | 0.437973 |
| 0.659000 | 0.322286 |
| 0.661000 | 0.142239 |
| 0.663000 | -0.127753 |
| 0.665000 | -0.520775 |
| 0.667000 | -1.079008 |
| 0.669000 | -1.855557 |
| 0.671000 | -2.916538 |
| 0.673000 | -4.343439 |
| 0.675000 | -6.235788 |
| 0.677000 | -8.714140 |
| 0.679000 | -11.923420 |
| 0.681000 | -16.036635 |

**Вычисления :**  
**newVec :**

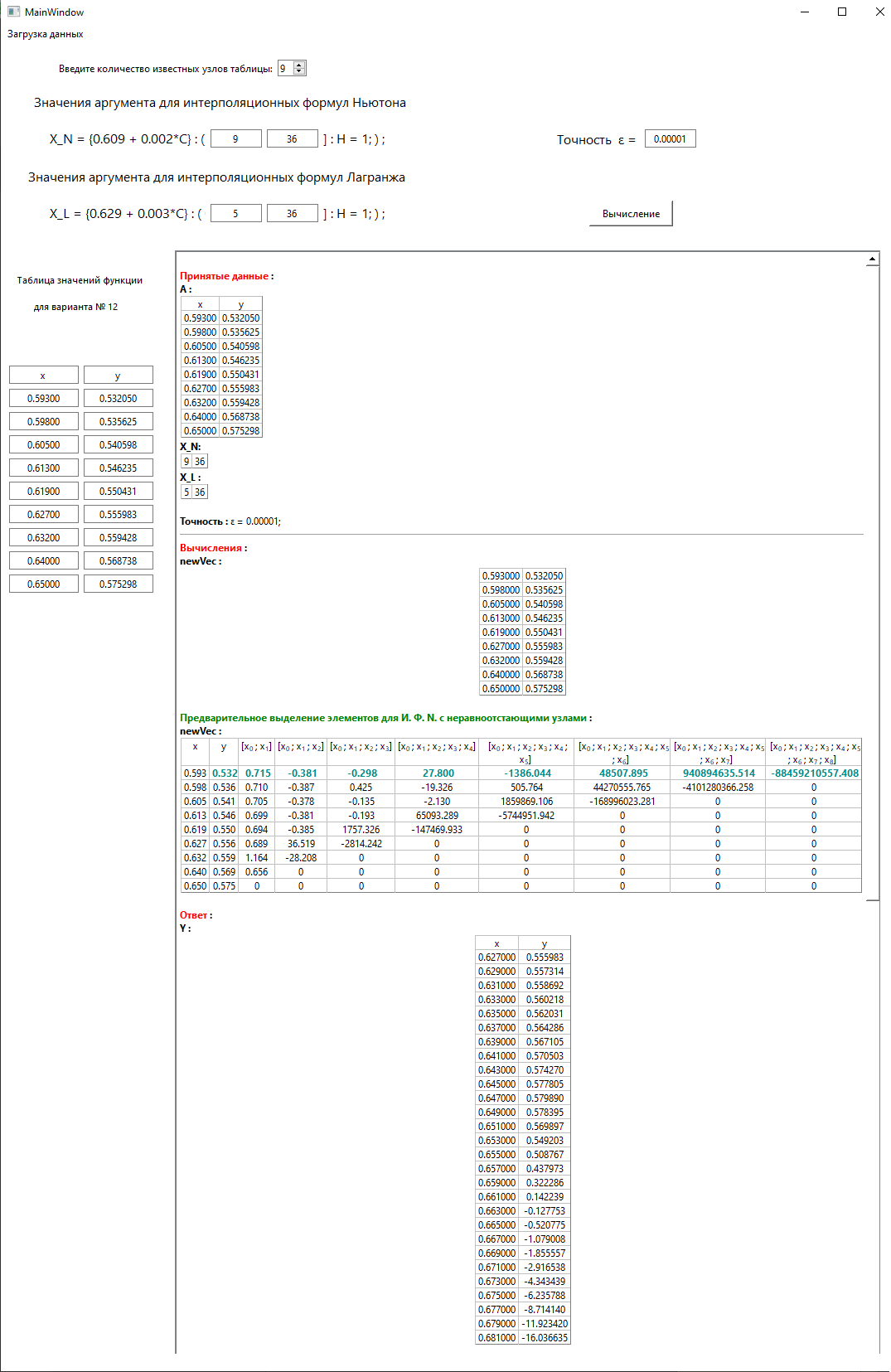
|  |  |
| --- | --- |
| 0.593000 | 0.532050 |
| 0.598000 | 0.535625 |
| 0.605000 | 0.540598 |
| 0.613000 | 0.546235 |
| 0.619000 | 0.550431 |
| 0.627000 | 0.555983 |
| 0.632000 | 0.559428 |
| 0.640000 | 0.568738 |
| 0.650000 | 0.575298 |

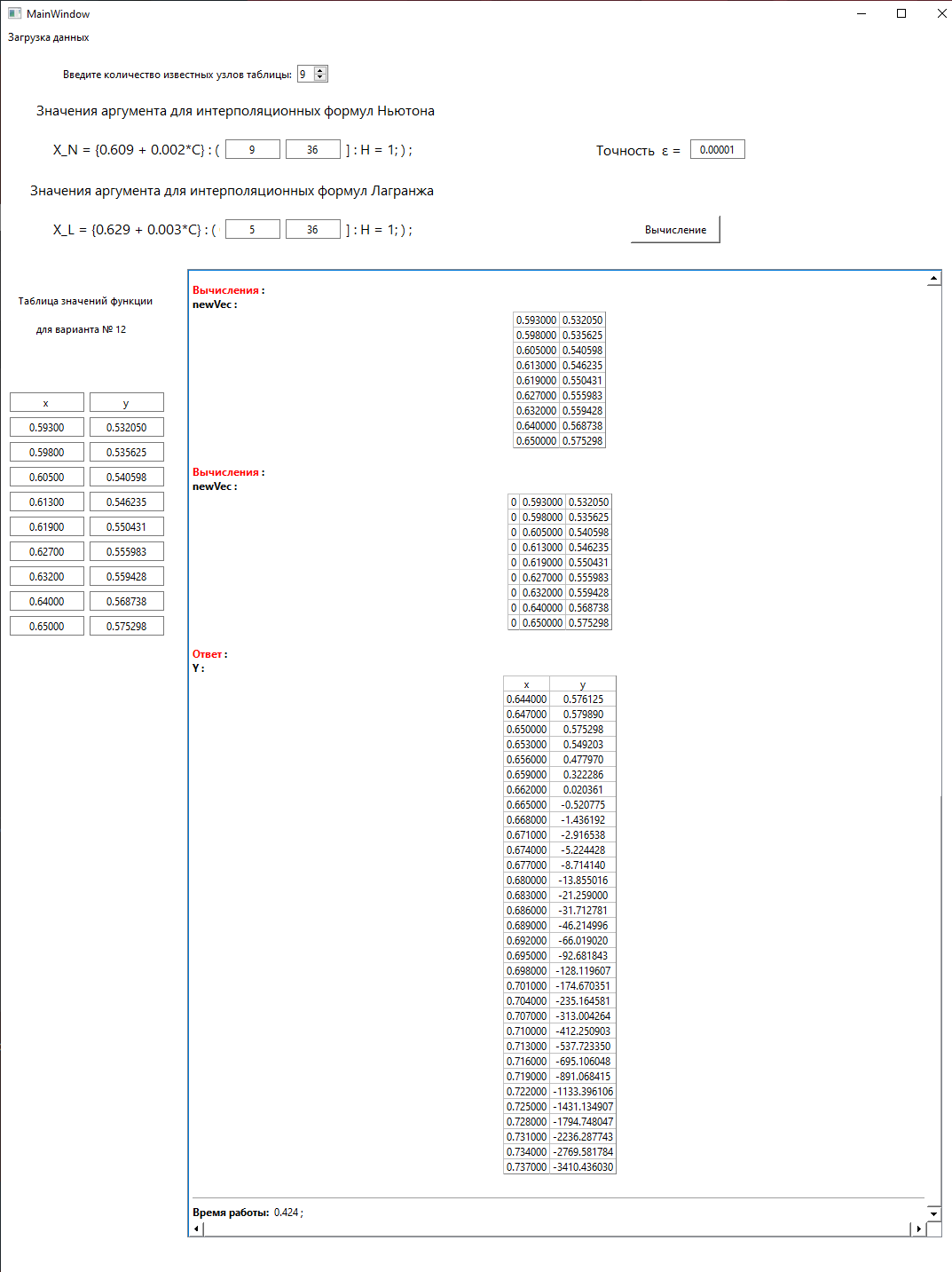
**Ответ :**  
**Y :**

|  |  |
| --- | --- |
| x | y |
| 0.644000 | 0.576125 |
| 0.647000 | 0.579890 |
| 0.650000 | 0.575298 |
| 0.653000 | 0.549203 |
| 0.656000 | 0.477970 |
| 0.659000 | 0.322286 |
| 0.662000 | 0.020361 |
| 0.665000 | -0.520775 |
| 0.668000 | -1.436192 |
| 0.671000 | -2.916538 |
| 0.674000 | -5.224428 |
| 0.677000 | -8.714140 |
| 0.680000 | -13.855016 |
| 0.683000 | -21.259000 |
| 0.686000 | -31.712781 |
| 0.689000 | -46.214996 |
| 0.692000 | -66.019020 |
| 0.695000 | -92.681843 |
| 0.698000 | -128.119607 |
| 0.701000 | -174.670351 |
| 0.704000 | -235.164581 |
| 0.707000 | -313.004264 |
| 0.710000 | -412.250903 |
| 0.713000 | -537.723350 |
| 0.716000 | -695.106048 |
| 0.719000 | -891.068415 |
| 0.722000 | -1133.396106 |
| 0.725000 | -1431.134907 |
| 0.728000 | -1794.748047 |
| 0.731000 | -2236.287743 |
| 0.734000 | -2769.581784 |
| 0.737000 | -3410.436030 |

**Время работы:** 0.242 ;

### Приложение № 1 – Примеры работы программы





### Приложение № 2 – Листинг программы

int Ga, Gb;

double step\_a, step\_b;

*auto* print\_general = [&](QVector<QVector<double>>& vec) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: red;\">Вычисления </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table align=center border=1 style=\"border-collapse: collapse;\">");

*if* (vec.count() == 2) strHtml\_A.append("<tr><td align=center> x </td><td align=center> y </td></tr>" );

foreach(*auto* str, vec){

strHtml\_A.append("<tr>");

foreach(*auto* elem, str){

elem == 0 ? strHtml\_A.append("<td align=center> " + QString::number(elem) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(elem,'f', 6) + " </td>");

}

strHtml\_A.append("</tr>");

}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*auto* print\_general\_o = [&](QVector<QVector<double>>& vec) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: red;\">Ответ </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>Y :</b><table align=center border=1 style=\"border-collapse: collapse;\">");

strHtml\_A.append("<tr><td align=center> x </td><td align=center> y </td></tr>" );

foreach(*auto* str, vec){

strHtml\_A.append("<tr>");

foreach(*auto* elem, str){

elem == 0 ? strHtml\_A.append("<td align=center> " + QString::number(elem) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(elem,'f', 6) + " </td>");

}

strHtml\_A.append("</tr>");

}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

*auto* print\_general\_N = [&](QVector<QVector<double>>& vec, QVector<double>& c) -> void {

strHtml\_A.clear();

strHtml\_A.append("<b style=\"color: green;\">Предварительное выделение элементов для И. Ф. N. с неравноотстающими узлами </b><b style=\"color: black;\">:</b><br>");

strHtml\_A.append("<b>newVec :</b><table align=center border=1 style=\"border-collapse: collapse;\">");

strHtml\_A.append("<tr><td align=center> x </td>"

"<td align=center> y </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub>] </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub> ; x<sub>2</sub>] </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub> ; x<sub>2</sub> ; x<sub>3</sub>] </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub> ; x<sub>2</sub> ; x<sub>3</sub> ; x<sub>4</sub>] </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub> ; x<sub>2</sub> ; x<sub>3</sub> ; x<sub>4</sub> ; x<sub>5</sub>] </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub> ; x<sub>2</sub> ; x<sub>3</sub> ; x<sub>4</sub> ; x<sub>5</sub> ; x<sub>6</sub>] </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub> ; x<sub>2</sub> ; x<sub>3</sub> ; x<sub>4</sub> ; x<sub>5</sub> ; x<sub>6</sub> ; x<sub>7</sub>] </td>"

"<td align=center> [x<sub>0</sub> ; x<sub>1</sub> ; x<sub>2</sub> ; x<sub>3</sub> ; x<sub>4</sub> ; x<sub>5</sub> ; x<sub>6</sub> ; x<sub>7</sub> ; x<sub>8</sub>] </td></tr>");

*for*(size\_t i = 0; i < CN; i++){

strHtml\_A.append("<tr>");

*for*(size\_t j = 0; j < vec[i].count(); j++){

*if*(i == 0 && j > 0){

vec[i][j] == 0 ? strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',0) + "</b></td>")

: strHtml\_A.append("<td align=center><b style=\"color: DarkCyan;\"> " + QString::number(vec[i][j],'f',3) + "</b></td>");

c.push\_front(vec[i][j]);

}*else*{

vec[i][j] == 0 ? strHtml\_A.append("<td align=center> " + QString::number(vec[i][j]) + " </td>")

: strHtml\_A.append("<td align=center> " + QString::number(vec[i][j],'f',3) + " </td>");

}

}

strHtml\_A.append("</tr>");

}

strHtml\_A.append("</table>");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + strHtml\_A + "<p></p>");

};

print\_general(A);

*auto* Duble\_A = A;

*auto* general\_i = [&]() -> void {

size\_t n = 3;

*auto* FL = [CN = CN] (QVector<QVector<double>> NewVec, size\_t N, double EPS) -> bool {

bool FLAG = *false*;

*for*(size\_t i = 0; i < CN; i++){

FLAG = (abs(NewVec[i][N-1]) > EPS) ? (*true*) : ((FLAG == *true*) ? (*true*) : (*false*));

}

*return* FLAG;

};

*auto* general = [&](*auto* GEN, int64\_t N, QVector<QVector<double>>& A, QVector<double>& C) -> void {

QVector<QVector<double>> newVec(CN,QVector<double>(N, 0));

*for*(size\_t i = 0; i < CN; i++){*//* *Копирование* *данных* *из* *А* *в* *newVec.*

*for*(size\_t j = 0; j < A[i].count(); j++){

newVec[i][j] = A[i][j];

}

}

*auto* razd\_raznost = [](*auto* r,QVector<QVector<double>>& A,int64\_t K, int64\_t i) -> double {

*return* (A[i + 1][K - 1] - A[i][K - 1])/((A[i + K - 1][0] - A[i][0]));

};

*for*(size\_t i = 0; i < CN - (N - 2); i++){

newVec[i][N - 1] = razd\_raznost(razd\_raznost, A, N - 1, i);

}

*if*((N <= CN)){*//* *FL(newVec,* *N,* *eps)* *&&*

GEN(GEN, N + 1, newVec, D);

}*else*{

*//* *1.* *Интерполяционный* *многочлен* *Ньютона.*

{

print\_general\_N(newVec, D);*//* *print* *one,* *end* *iteration;*

Y = QVector<QVector<double>>(B[1] - B[0] + 1, QVector<double>(2, 0));

*auto* P\_n = [D = D, A = A](*auto* P\_n, double x, uint32\_t i = 0) -> double {

*if*(i > D.count()) *return* 0;

*auto* R = [A = A](*auto* r, double x, uint32\_t i) -> double {

*if* (i == 1) *return* 1;

*return* (x - A[i - 2][0])\*r(r, x, i - 1);

};

*if* (i == 0){

*return* D[D.count() - 1] + P\_n(P\_n, x, i + 2);

}*else*{

*return* D[D.count() - i]\*R(R, x, i) + P\_n(P\_n, x, i + 1);

}

};

*for*(int i = B[0]; i <= B[1]; i++){

double x = 0.609 + 0.002\*i;

double y = P\_n(P\_n, x);

Y[i - B[0]][0] = x;

Y[i - B[0]][1] = y;

}

print\_general\_o(Y);

}

*//* *2.* *интерполяционный* *многочлен* *лагранжа.*

{

print\_general(Duble\_A);

W = QVector<QVector<double>>(Duble\_A.count(),QVector<double>(3, 0));

*for*(size\_t i = 0; i < Duble\_A.count(); i++){

*for*(size\_t j = 1; j < 3; j++){

W[i][j] = Duble\_A[i][j - 1];

}

}

print\_general(W);

*auto* P\_n\_L = [W = Duble\_A](*auto* P\_n\_L, double x, int32\_t i = 0) -> double {

*if*(i >= W.count()) *return* 0;

*auto* Z = [](*auto* z, double x, QVector<QVector<double>> W, int32\_t i, int32\_t j = 0) -> double {

*if*(i == j){

*return* z(z, x, W, i, j + 1);

}*else* *if*(i >= W.count() || j >= W.count()){

*return* 1;

}*else* {

*return*(x - W[j][0])/(W[i][0] - W[j][0])\*z(z, x, W, i , j + 1);

}

};

*return* W[i][1]\*Z(Z, x, W, i) + P\_n\_L(P\_n\_L, x, i + 1);

};

P = QVector<QVector<double>>(E[1] - E[0] + 1, QVector<double>(2, 0));

*for*(int i = E[0]; i <= E[1]; i++){

double x = 0.629 + 0.003\*i;

double y = P\_n\_L(P\_n\_L, x);

P[i - E[0]][0] = x;

P[i - E[0]][1] = y;

}

print\_general\_o(P);

}

}

};

general(general, n, A, C);

};

general\_i();

Ссылка на git-Hub: <https://github.com/MineevS/CHM_3_6.git>