**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по практической работе №3**

**по дисциплине «Вычислительная математика»**

**Тема: Метод бисекции и метод хорд.**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1304 |  | Чернякова В.А. |
| Преподаватель |  | Попова Е.В. |

Санкт-Петербург

2022

## Цель работы.

## Целью работы является нахождение корня уравнения f(x)=0 методом бисекции с заданной точностью Eps, исследование зависимости числа итераций k от точности Eps при заданном изменении Eps, исследование обусловленности метода (чувствительность к ошибкам в исходных данных); исследование обусловленности метода без ограничения итераций заданной точностью Eps, когда рассчитывается N>30 членов последовательности приближенного значения корня.

Целью работы также является нахождение корня уравнения f(x)=0 методом хорд с заданной точностью Eps, исследование скорости сходимости и обусловленности метода; исследование обусловленности метода без ограничения итераций заданной точностью Eps, когда рассчитывается N>30 членов последовательности приближенного значения корня.

## Основные теоретические положения.

**Метод бисекции.** Пусть заданная функция - непрерывна на отрезке , тогда существует точка *c*, в которой значение функции равно нулю, т.е. =0 (следствие из теоремы Больцано-Коши) [8]. В методе *бисекции* строят последовательность вложенных друг в друга отрезков, на концах которых функция имеет разные знаки. Каждый последующий отрезок получается делением пополам предыдущего. Процесс построения последовательности отрезков позволяет найти корень уравнения =0 с любой заданной точностью.

Опишем *(n-1)-*ю итерацию метода. На *(n-1)-*м шаге найден отрезок такой, что . Разделим данный отрезок пополам точкой и вычислим (рисунок 1).

Рисунок 1 – Деление отрезка пополам

Если =0, то - корень уравнения, при этом ошибка вычислений равна нулю, и задача хорошо обусловлена. Если 0, то из двух половин отрезка выбирается та, на концах которой функция имеет противоположные знаки, поскольку искомый корень лежит на этой половине. Таким образом, , , если ; или , , если .

Пусть во входных данных задачи задан *ε,* являющийся точностью искомого корня, тогда процесс деления пополам продолжается до тех пор, пока длина отрезка не станет меньше *2ε*. За значение корня функции с требуемой точностью *ε* принимается координата середины отрезка.

Другой подход вычислительного эксперимента заключается в отказе от ограничения *2ε*-окрестностью количества итераций. Строятся две последовательности приближений корня с двойной точностью и с наложенным шумом. Вычисляются значение числа обусловленности и абсолютная погрешность в точке , полученной без искажения функции при большом числе итераций . Проверяется неравенство и фиксируется при каких значениях итерационной переменной метод можно считать хорошо обусловленным.

Метод *бисекции* является несложным и надежным методом поиска простого корня уравнения (простым называется корень дифференцируемой функции , если и ).

Этот метод сходится для любых непрерывных функций , в том числе недифференцируемых. Скорость его сходимости невысока. Для достижения точности *ε* необходимо совершить порядка итераций. Это означает, что для получения каждых трех верных десятичных знаков необходимо совершить около 10 итераций.

Плюсы метода бисекции:

* Всегда сходится.
* Не требует гладкости.

Минусы метода:

* Сходимость относительно медленная.
* Не применяется для поиска кратных корней.

Для определения скорости сходимости обозначим середину *n*-го отрезка за точку . Погрешность приближения к корню определяется по формуле |. Из этой оценки видно, что метод *бисекции* сходится со скоростью геометрической прогрессии, знаменатель которой . По сравнению с другими методами метод *бисекции* сходится довольно медленно. Критерий окончания итерационного процесса определяется неравенством. При его выполнении, за приближенное значение корня принимают которое приближено к корню с точностью .

В случае, когда попадает в интервал неопределенности корня , знак вычисленного приближенного значения функции может быть неверным, и последующие итерации не имеют смысла. Однако этот метод следует признать очень надежным; он гарантирует точность приближения, примерно равную радиусу интервала неопределенности.

**Метод хорд.** Метод хорд также применяется в задаче нахождения корня уравнения =0. Пусть найден отрезок , на котором функция меняет знак. Для определенности положим >0, . В методе *хорд* процесс итераций состоит в том, что в качестве приближений к корню уравнения =0 принимаются значения . . . точек пересечения хорды с осью абсцисс, как это показано на рисунке 2.

*с*

Рисунок 2 – Построение хорд в используемом методе

Сначала находится уравнение хорды АВ (прямой, проходящей через 2 точки): . Для точки пересечения ее с осью абсцисс получается уравнение . Далее сравниваются знаки величин и и для рассматриваемого случая оказывается, что корень находится в интервале , так как . Отрезок отбрасывается. Следующая итерации состоит в определении нового приближения - точки пересечения хорды с осью абсцисс и т.д. Итерационный процесс продолжается до тех пор, пока значение не станет по модулю меньше заданного числа .

Прекращение итерационного процесса происходит при ложном значении условия сравнения модуля функции и Eps (листинг 4).

Другой подход вычислительного эксперимента заключается по аналогии с методом *бисекции* в отказе от сравнения значения функции и заданного . Строятся две последовательности приближений корня с двойной точностью и с наложенным шумом. Вычисляются значение числа обусловленности и абсолютная погрешность в точке полученной при большом числе итераций с функцией без искажений. Вычисляется . Проверяется неравенство и фиксируется при каких значениях итерационной переменной метод можно считать хорошо обусловленным.

Алгоритмы методов *бисекции* и *хорд* похожи, однако метод хорд в ряде случаев дает более быструю сходимость итерационного процесса, причем успех его применения, как и метода *бисекции*, гарантирован. Метод обладает линейной сходимостью, которая зависит от выбора начального приближения .

## Порядок выполнения практической работы.

## Часть 1.

В работе предлагается использовать модули программы, вычисляющие функции BISECT и Round (листинг 1).

Порядок выполнения работы должен быть следующим:

1) Графически или аналитически отделить корень уравнения (найти отрезки [Left, Right], на которых функция . удовлетворяет условиям теоремы Коши).

2) Составить подпрограмму вычисления функции

3) Составить подпрограмму вычисления функции , для вычисления .

4) Составить головную программу, содержащую обращение к подпрограммам , BISECT, Round и вывод результатов.

5) Провести вычисления по программе. Построить график зависимости числа итераций от *Eps*.

6) Исследовать чувствительность метода к ошибкам в исходных данных. Ошибки в исходных данных моделировать с использованием программы Round, округляющей значения функции с заданной точностью Delta.

В коде прописывается вывод и (листинг 2).

Листинг 2 – Вывод значений и

std::cout << "V delta: " <<………………………. << "\n";

std::cout << "V delta max: " << eps / delta << "\n";

## Часть 2.

Во второй части работы модули программы, вычисляющие функции BISECT и F, модифицируются с новым условием прохождения цикла (листинг 2) и округления функции методом усечения.

1. Методом бисекции построить итерационную последовательность приближений к корню последовательности , где и задано в варианте. Вычисления проводить с двойной точностью.
2. Вычислить значение числа обусловленности по формуле , принимая за значение корня приближение полученное при числе итераций равное
3. Внести неточности в значения вычисляемой функции методом усечения точных значений функции до знака после запятой, заданного параметром . Получить приближенные значения итерационной последовательности .
4. Вычислить абсолютные величины разностей между значениями и приближенной итерационной последовательности где
5. Определить при каких значениях выполняется соотношение и задача является хорошо обусловленной.

Листинг 2 – Фрагмент модифицированной программы «BISECT»

while((\*N)<80)

{x = (left + right) / 2;

## Часть 3.

В практической работе предлагается, использовать программы-функции HORDA (листинг 4) и Round (листинг 1).

Листинг 4 – Функция HORDA

double HORDA(double Left, double Right, double Eps, int &N) {

double FLeft = F(Left);

double FRight = F(Right);

double X, Y;

if (FLeft \* FRight > 0.0) {

puts("Неверное задание интервала\n");

exit(1);}

if (Eps <= 0.0) {puts("Неверное задание точности\n");

exit(1);}

N = 0;

if (FLeft == 0.0) {

return Left;}

if (FRight == 0.0) {

return Right;}

do {X = Left - (Right - Left) \* FLeft / (FRight - FLeft);

Y = F(X);

if (Y == 0.0) {

return X;}

if (Y \* FLeft < 0.0) {

Right = X;

FRight = Y;}

else {Left = X;

FLeft = Y;}

N++;} while (fabs(Y) >= Eps);

return X;}Порядок выполнения лабораторной работы:

1) Графически или аналитически отделить корень уравнения (найти отрезки [Left, Right], на которых функция . удовлетворяет условиям теоремы Коши).

2) Составить подпрограмму вычисления функции предусмотрев округление значений функции с заданной точностью Delta с использованием программы Round.

3) Составить подпрограмму вычисления функции , для вычисления .

4) Составить головную программу, вычисляющую корень уравнения и содержащую обращение к подпрограмме , HORDA, Round и индикацию результатов.

5) Провести вычисления по программе. Теоретически и экспериментально исследовать скорость сходимости и обусловленность метода.

1. По аналогии со второй частью практической работы метода бисекции, провести исследование обусловленности задачи метода *хорд* при построении приближений к корню последовательности и приближенных значений итерационной последовательности после наложения шума на функцию.

## Выполнение работы.

*Вариант 28.*

*Часть 1.*

1. Функция *f(x) = 1/cos(cx) + 1* тригонометрическая и, следовательно,

является периодической, то есть значения функции через регулярные промежутки времени повторяются.

В область допустимых значений данной функции не входят значения,

принадлежащие интервалу *(π/2c + πn/c),* где *n* – целое число, так как функция *cos(cx)* принимает значение равное 0.

При *с*, принимающем значения от 3 до 3.5, корень находится на отрезке [3; 4].

Пример графика функции при *с* = 3 см. рис. 1. Корни , где *n*

целое число.

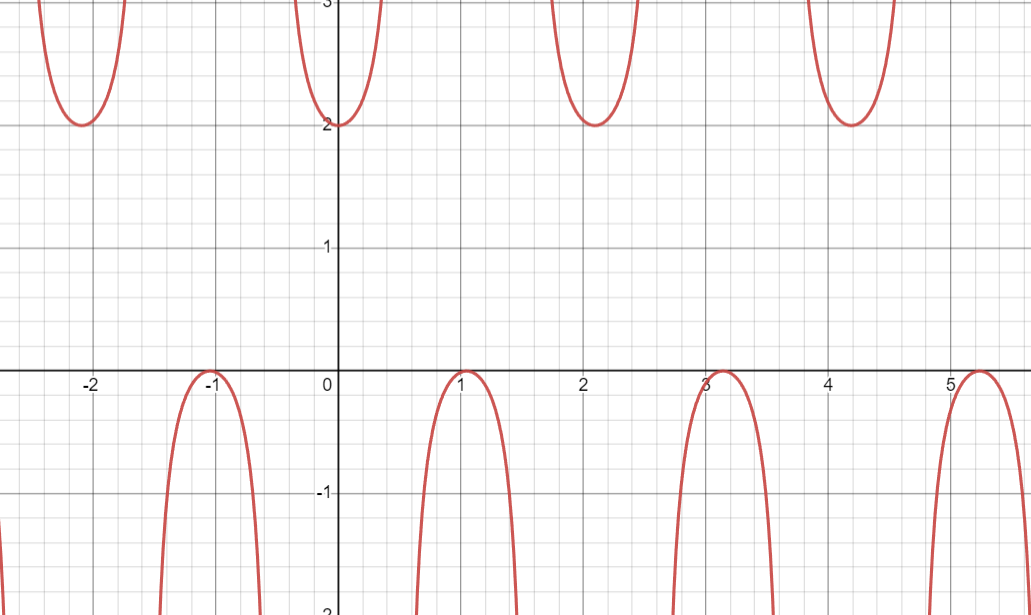


Рисунок 1 – график функции f(x) = 1/cos(cx) + 1 при с = 3.

1. Была составлена подпрограмма *F*() для вычисления функции 𝑓(𝑥) для

параметра с вводимая с клавиатуры. Функция *F()* принимает аргумент x и возвращает значение функции *1/cos(cx) + 1*.

*double F(double x){*

*return (1/(cos(c\*x)))+1;*

*}*

1. Была составлена подпрограмма *Derivative()* для вычисления функции

для параметра с вводимого с клавиатуры. Функция *Dertivate()* принимает аргумент *x* и возвращает значение производной функции *1/cos(cx) + 1 в точке x*.

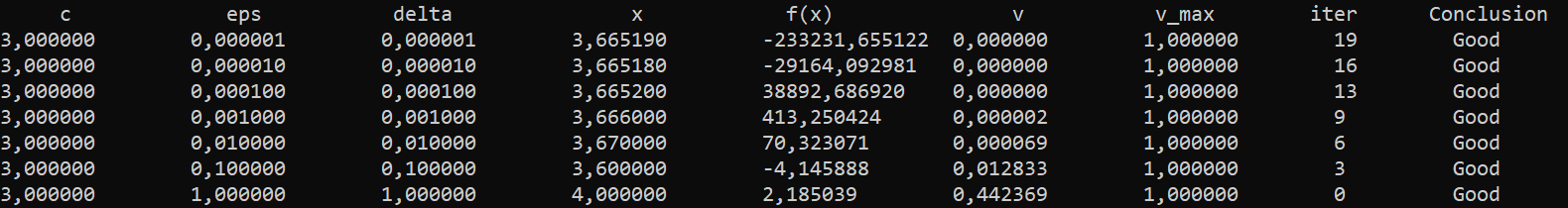
*double Derivative(double x){*

*int F\_ = (c\*sin(c\*x))/(pow(cos(c\*x), 2));*

*return F\_;*

*}*

1. Составлена головная программы - main, содержащая обращение к подпрограмме *F*, *Derivative* и программам-функциям *BISECT*, *Round* и осуществляющая вывод результатов на экран. Разработанный код представлен в приложении.
2. Проведение вычислений и анализ.

Для построения графика зависимости числа итераций от *eps* проведем вычисление по программе.

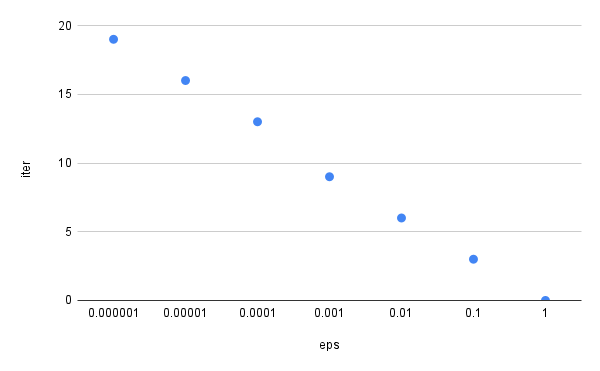
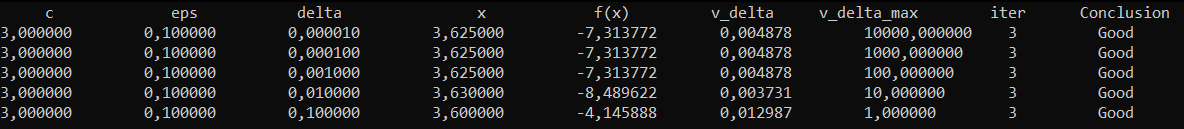
Рисунок 2 – вычисление программы при постоянном параметре с равном 3, и одновременном варьировании *delta* и *eps* от 0.000001 до 1.

Рисунок 3 – график зависимости числа итераций от *Eps*.

Вывод: Чем меньше значение *Eps*, тем больше число итераций.

1. Исследуем чувствительность метода к ошибкам в исходных данных. Ошибки в исходных данных моделируем с использованием программы *Round*, округляющей значения функции с заданной точностью *Delta*.

Рисунок 4 – вычисления программы при постоянном значении *с* равном 3 и *eps* равном 0.1 и варьирующемся значении *delta* от 0.00001 до 0.1.

Вывод: При любом значении *delta* от 0.00001 до 0.1 и постоянных *с* = 3 и *eps* = 0.1 функция хорошо обусловлена.

*Часть 2.*

1. Методом бисекции построим итерационную последовательность приближений к корню последовательности , где . Вычисления проводим с двойной точностью.

На основе функции *BISECT* была создана новая функция *BISECT2*, в которой был изменен цикл *while*:

*while ((Right - Left) >= E)*

Теперь цикл работает пока выполняется следующее условие:

*while (N <= 80)*

Для вывода итерационной последовательности приближений к корню последовательности.

Также добавлен вывод в консоль очередного значения корня для каждой новой итерации.

*std::cout<<"x"<<N<<" "<<X<<"\n";*

Таблица 1 – Итерационная последовательность приближений к корню последовательности

|  |  |
| --- | --- |
| x0 | 3.5 |
| x1 | 3.75 |
| x2 | 3.625 |
| x3 | 3.6875 |
| x4 | 3.65625 |
| x5 | 3.67188 |
| x6 | 3.66406 |
| x7 | 3.66797 |
| x8 | 3.66602 |
| x9 | 3.66504 |
| x10 | 3.66553 |
| x11 | 3.66528 |
| x12 | 3.66516 |
| x13 | 3.66522 |
| x14 | 3.66519 |
| x15 | 3.66518 |
| x16 | 3.66518 |
| x17 | 3.66519 |
| x18 | 3.66519 |
| x19 | 3.66519 |
| x20 | 3.66519 |
| x21 | 3.66519 |
| x22 | 3.66519 |
| x23 | 3.66519 |
| x24 | 3.66519 |
| x25 | 3.66519 |
| x26 | 3.66519 |
| x27 | 3.66519 |
| x28 | 3.66519 |
| x29 | 3.66519 |
| x30 | 3.66519 |
| x31 | 3.66519 |
| x32 | 3.66519 |
| x33 | 3.66519 |
| x34 | 3.66519 |
| x35 | 3.66519 |
| x36 | 3.66519 |
| x37 | 3.66519 |
| x38 | 3.66519 |
| x39 | 3.66519 |
| x40 | 3.66519 |
| x41 | 3.66519 |
| x42 | 3.66519 |
| x43 | 3.66519 |
| x44 | 3.66519 |
| x45 | 3.66519 |
| x46 | 3.66519 |
| x47 | 3.66519 |
| x48 | 3.66519 |
| x49 | 3.66519 |
| x50 | 3.66519 |
| x51 | 3.66519 |
| x52 | 3.66519 |
| x53 | 3.66519 |
| x54 | 3.66519 |
| x55 | 3.66519 |
| x56 | 3.66519 |
| x57 | 3.66519 |
| x58 | 3.66519 |
| x59 | 3.66519 |
| x60 | 3.66519 |
| x61 | 3.66519 |
| x62 | 3.66519 |
| x63 | 3.66519 |
| x64 | 3.66519 |
| x65 | 3.66519 |
| x66 | 3.66519 |
| x67 | 3.66519 |
| x68 | 3.66519 |
| x69 | 3.66519 |
| x70 | 3.66519 |
| x71 | 3.66519 |
| x72 | 3.66519 |
| x73 | 3.66519 |
| x74 | 3.66519 |
| x75 | 3.66519 |
| x76 | 3.66519 |
| x77 | 3.66519 |
| x78 | 3.66519 |
| x79 | 3.66519 |

1. Вычислим значение числа обусловленности по формуле *,* где *xN*(*N* = 79) = 3.66519

*F(x) = 1/cos(cx)+1*

*F’(x) = (c\*sin(c\*x))/cos(c\*x)^2*

Подставляем *с* = 3, *x* = 3.66519

*F’(x)* = -1.6516e+30

*|F’(x)|* = 1.6516e+30

= 1/1.6516e+30 = 6.05472e-31

1. Вносятся неточности в значения вычисляемой функции методом усечения точных значений функции до знака после запятой, заданного параметром .

При выводе *x\** вызывается функция *truncRound(),* которая округляет значение в соответствии с *eps = 0.01* – два знака после запятой.

Получаются приближенные значения итерационной последовательности .

Вычисляются абсолютные величины разностей между значениями и приближенной итерационной последовательности где

Определяется при каких значениях выполняется соотношение и задача является хорошо обусловленной.

Для вычислений *с* = 3, *eps* = 0.01, . Интервал – от 3 до 4.

Таблица 2 – Вычисления по условиям из пунктов 3-5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **Xn** | **X\*n** | **i** | **Conclusion** |
| 0 | 3.5 | 3.5 | 0 | GOOD |
| 1 | 3.75 | 3.75 | 0 | GOOD |
| 2 | 3.625 | 3.62 | 0.005 | BAD |
| 3 | 3.6875 | 3.68 | 0.0075 | BAD |
| 4 | 3.65625 | 3.65 | 0.00625 | BAD |
| 5 | 3.67188 | 3.67 | 0.001875 | BAD |
| 6 | 3.66406 | 3.66 | 0.0040625 | BAD |
| 7 | 3.66797 | 3.66 | 0.00796875 | BAD |
| 8 | 3.66602 | 3.66 | 0.00601562 | BAD |
| 9 | 3.66504 | 3.66 | 0.00503906 | BAD |
| 10 | 3.66553 | 3.66 | 0.00552734 | BAD |
| 11 | 3.66528 | 3.66 | 0.0052832 | BAD |
| 12 | 3.66516 | 3.66 | 0.00516113 | BAD |
| 13 | 3.66522 | 3.66 | 0.00522217 | BAD |
| 14 | 3.66519 | 3.66 | 0.00519165 | BAD |
| 15 | 3.66518 | 3.66 | 0.00517639 | BAD |
| 16 | 3.66518 | 3.66 | 0.00518402 | BAD |
| 17 | 3.66519 | 3.66 | 0.00518784 | BAD |
| 18 | 3.66519 | 3.66 | 0.00518974 | BAD |
| 19 | 3.66519 | 3.66 | 0.0051907 | BAD |
| 20 | 3.66519 | 3.66 | 0.00519117 | BAD |
| 21 | 3.66519 | 3.66 | 0.00519141 | BAD |
| 22 | 3.66519 | 3.66 | 0.00519153 | BAD |
| 23 | 3.66519 | 3.66 | 0.00519147 | BAD |
| 24 | 3.66519 | 3.66 | 0.00519144 | BAD |
| 25 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 26 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 27 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 28 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 29 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 30 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 31 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 32 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 33 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 34 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 35 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 36 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 37 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 38 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 39 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 40 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 41 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 42 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 43 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 44 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 45 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 46 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 47 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 48 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 49 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 50 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 51 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 52 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 53 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 54 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 55 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 56 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 57 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 58 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 59 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 60 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 61 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 62 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 63 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 64 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 65 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 66 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 67 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 68 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 69 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 70 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 71 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 72 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 73 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 74 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 75 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 76 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 77 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 78 | 3.66519 | 3.66 | 0.00519143 | BAD |
| 79 | 3.66519 | 3.66 | 0.00519143 | BAD |

Вывод: При N равном 0 и 1 функция хорошо обусловлена, в других случаях – плохо обусловлена.

*Часть 3.*

1. Методом хорд построим итерационную последовательность приближений к корню последовательности , где . Вычисления проводим с двойной точностью.

На основе функции *HORDA* была создана новая функция *HORDA2*, в которой был изменен цикл *while*:

*while (fabs(Y) >= Eps)*

Теперь цикл работает пока выполняется следующее условие:

*while (N < 80)*

Для вывода итерационной последовательности приближений к корню последовательности.

Также добавлен вывод в консоль очередного значения корня для каждой новой итерации.

*std::cout<<"x"<<N<<" "<<X<<"\n";*

Таблица 3 – Итерационная последовательность приближений к корню последовательности

|  |  |
| --- | --- |
| x0 | 3.04273 |
| x1 | 3.06232 |
| x2 | 3.07459 |
| x3 | 3.08321 |
| x4 | 3.08968 |
| x5 | 3.09476 |
| x6 | 3.09886 |
| x7 | 3.10226 |
| x8 | 3.10513 |
| x9 | 3.10758 |
| x10 | 3.10971 |
| x11 | 3.11158 |
| x12 | 3.11323 |
| x13 | 3.1147 |
| x14 | 3.11602 |
| x15 | 3.11721 |
| x16 | 3.1183 |
| x17 | 3.11928 |
| x18 | 3.12019 |
| x19 | 3.12102 |
| x20 | 3.12178 |
| x21 | 3.12249 |
| x22 | 3.12315 |
| x23 | 3.12377 |
| x24 | 3.12434 |
| x25 | 3.12488 |
| x26 | 3.12538 |
| x27 | 3.12586 |
| x28 | 3.1263 |
| x29 | 3.12672 |
| x30 | 3.12712 |
| x31 | 3.1275 |
| x32 | 3.12785 |
| x33 | 3.12819 |
| x34 | 3.12852 |
| x35 | 3.12882 |
| x36 | 3.12912 |
| x37 | 3.12939 |
| x38 | 3.12966 |
| x39 | 3.12992 |
| x40 | 3.13016 |
| x41 | 3.1304 |
| x42 | 3.13062 |
| x43 | 3.13084 |
| x44 | 3.13104 |
| x45 | 3.13124 |
| x46 | 3.13143 |
| x47 | 3.13162 |
| x48 | 3.1318 |
| x49 | 3.13197 |
| x50 | 3.13213 |
| x51 | 3.13229 |
| x52 | 3.13245 |
| x53 | 3.1326 |
| x54 | 3.13274 |
| x55 | 3.13288 |
| x56 | 3.13302 |
| x57 | 3.13315 |
| x58 | 3.13328 |
| x59 | 3.1334 |
| x60 | 3.13352 |
| x61 | 3.13364 |
| x62 | 3.13375 |
| x63 | 3.13386 |
| x64 | 3.13397 |
| x65 | 3.13407 |
| x66 | 3.13417 |
| x67 | 3.13427 |
| x68 | 3.13436 |
| x69 | 3.13446 |
| x70 | 3.13455 |
| x71 | 3.13464 |
| x72 | 3.13472 |
| x73 | 3.13481 |
| x74 | 3.13489 |
| x75 | 3.13497 |
| x76 | 3.13505 |
| x77 | 3.13512 |
| x78 | 3.1352 |
| x79 | 3.13527 |

1. Вычислим значение числа обусловленности по формуле *,* где *xN*(*N* = 79) = 3.13527

*F(x) = 1/cos(cx)+1*

*F’(x) = (c\*sin(c\*x))/cos(c\*x)^2*

Подставляем *с* = 3, *x* = 3.13527

*F’(x)* = 0.0569156

*|F’(x)|* = 0.0569156

= 1/ 0.0569156 = 17.5699

1. Вносятся неточности в значения вычисляемой функции методом усечения точных значений функции до знака после запятой, заданного параметром .

При выводе *x\** вызывается функция *truncRound(),* которая округляет значение в соответствии с *eps = 0.01* – два знака после запятой.

Получаются приближенные значения итерационной последовательности .

Вычисляются абсолютные величины разностей между значениями и приближенной итерационной последовательности где

Определяется при каких значениях выполняется соотношение и задача является хорошо обусловленной.

Для вычислений с = 3, eps = 0.01, . Интервал – от 3 до 4.

Таблица 4 – Вычисления по условиям из пунктов 3-5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **Xn** | **X\*n** | **del\_i** | **Conclusion** |
| 0 | 3.04273 | 03.04 | 0.00273148 | GOOD |
| 1 | 3.06232 | 03.06 | 0.00232263 | GOOD |
| 2 | 3.07459 | 03.07 | 0.00458733 | GOOD |
| 3 | 3.08321 | 03.08 | 0.00320944 | GOOD |
| 4 | 3.08968 | 03.08 | 0.00968246 | GOOD |
| 5 | 3.09476 | 03.09 | 0.00475743 | GOOD |
| 6 | 3.09886 | 03.09 | 0.00886208 | GOOD |
| 7 | 3.10226 | 3.1 | 0.00226118 | GOOD |
| 8 | 3.10513 | 3.1 | 0.0051288 | GOOD |
| 9 | 3.10758 | 3.1 | 0.00758473 | GOOD |
| 10 | 3.10971 | 3.1 | 0.00971449 | GOOD |
| 11 | 3.11158 | 3.11 | 0.00158093 | GOOD |
| 12 | 3.11323 | 3.11 | 0.00323143 | GOOD |
| 13 | 3.1147 | 3.11 | 0.0047024 | GOOD |
| 14 | 3.11602 | 3.11 | 0.00602237 | GOOD |
| 15 | 3.11721 | 3.11 | 0.00721401 | GOOD |
| 16 | 3.1183 | 3.11 | 0.00829561 | GOOD |
| 17 | 3.11928 | 3.11 | 0.00928206 | GOOD |
| 18 | 3.12019 | 3.12 | 0.000185663 | GOOD |
| 19 | 3.12102 | 3.12 | 0.00101665 | GOOD |
| 20 | 3.12178 | 3.12 | 0.0017836 | GOOD |
| 21 | 3.12249 | 3.12 | 0.00249378 | GOOD |
| 22 | 3.12315 | 3.12 | 0.00315339 | GOOD |
| 23 | 3.12377 | 3.12 | 0.00376774 | GOOD |
| 24 | 3.12434 | 3.12 | 0.0043414 | GOOD |
| 25 | 3.12488 | 3.12 | 0.00487837 | GOOD |
| 26 | 3.12538 | 3.12 | 0.00538211 | GOOD |
| 27 | 3.12586 | 3.12 | 0.00585565 | GOOD |
| 28 | 3.1263 | 3.12 | 0.00630168 | GOOD |
| 29 | 3.12672 | 3.12 | 0.00672256 | GOOD |
| 30 | 3.12712 | 3.12 | 0.00712039 | GOOD |
| 31 | 3.1275 | 3.12 | 0.00749703 | GOOD |
| 32 | 3.12785 | 3.12 | 0.00785417 | GOOD |
| 33 | 3.12819 | 3.12 | 0.00819329 | GOOD |
| 34 | 3.12852 | 3.12 | 0.00851575 | GOOD |
| 35 | 3.12882 | 3.12 | 0.00882276 | GOOD |
| 36 | 3.12912 | 3.12 | 0.00911541 | GOOD |
| 37 | 3.12939 | 3.12 | 0.00939471 | GOOD |
| 38 | 3.12966 | 3.12 | 0.00966155 | GOOD |
| 39 | 3.12992 | 3.12 | 0.00991677 | GOOD |
| 40 | 3.13016 | 3.13 | 0.000161108 | GOOD |
| 41 | 3.1304 | 3.13 | 0.00039526 | GOOD |
| 42 | 3.13062 | 3.13 | 0.000619855 | GOOD |
| 43 | 3.13084 | 3.13 | 0.000835474 | GOOD |
| 44 | 3.13104 | 3.13 | 0.00104265 | GOOD |
| 45 | 3.13124 | 3.13 | 0.00124187 | GOOD |
| 46 | 3.13143 | 3.13 | 0.0014336 | GOOD |
| 47 | 3.13162 | 3.13 | 0.00161824 | GOOD |
| 48 | 3.1318 | 3.13 | 0.0017962 | GOOD |
| 49 | 3.13197 | 3.13 | 0.00196782 | GOOD |
| 50 | 3.13213 | 3.13 | 0.00213345 | GOOD |
| 51 | 3.13229 | 3.13 | 0.0022934 | GOOD |
| 52 | 3.13245 | 3.13 | 0.00244796 | GOOD |
| 53 | 3.1326 | 3.13 | 0.00259739 | GOOD |
| 54 | 3.13274 | 3.13 | 0.00274196 | GOOD |
| 55 | 3.13288 | 3.13 | 0.00288189 | GOOD |
| 56 | 3.13302 | 3.13 | 0.00301741 | GOOD |
| 57 | 3.13315 | 3.13 | 0.00314872 | GOOD |
| 58 | 3.13328 | 3.13 | 0.00327602 | GOOD |
| 59 | 3.1334 | 3.13 | 0.0033995 | GOOD |
| 60 | 3.13352 | 3.13 | 0.00351932 | GOOD |
| 61 | 3.13364 | 3.13 | 0.00363564 | GOOD |
| 62 | 3.13375 | 3.13 | 0.00374862 | GOOD |
| 63 | 3.13386 | 3.13 | 0.0038584 | GOOD |
| 64 | 3.13397 | 3.13 | 0.00396511 | GOOD |
| 65 | 3.13407 | 3.13 | 0.00406889 | GOOD |
| 66 | 3.13417 | 3.13 | 0.00416985 | GOOD |
| 67 | 3.13427 | 3.13 | 0.00426811 | GOOD |
| 68 | 3.13436 | 3.13 | 0.00436377 | GOOD |
| 69 | 3.13446 | 3.13 | 0.00445694 | GOOD |
| 70 | 3.13455 | 3.13 | 0.00454771 | GOOD |
| 71 | 3.13464 | 3.13 | 0.00463618 | GOOD |
| 72 | 3.13472 | 3.13 | 0.00472243 | GOOD |
| 73 | 3.13481 | 3.13 | 0.00480655 | GOOD |
| 74 | 3.13489 | 3.13 | 0.00488861 | GOOD |
| 75 | 3.13497 | 3.13 | 0.00496869 | GOOD |
| 76 | 3.13505 | 3.13 | 0.00504686 | GOOD |
| 77 | 3.13512 | 3.13 | 0.00512319 | GOOD |
| 78 | 3.1352 | 3.13 | 0.00519775 | GOOD |
| 79 | 3.13527 | 3.13 | 0.00527059 | GOOD |

Вывод: При любых N функция хорошо обусловлена.

*Работа с возмущениями.*

Функции округления для возмущений –

*double Round\_Truncation(double Y) {*

*if (Y >= 0) {*

*return floor(Y);//round down*

*} else {*

*return ceil(Y);//round up*

*}*

*}*

Возмущение 1. Осуществляется округление до двух знаков после запятой.

Возмущение 2. Осуществляется округление до трех знаков после запятой.

Возмущение 3. Осуществляется округление до четырех знаков после запятой.

Для следующих таблиц вычисления происходили следующим образом:

* *Xn* и *Yn* *классическое* – работа обычной функции *BISECT()/HORDA().* На каждом шаге работа цикла выводится значение переменной *X* и соответствующее от него *F(X);*
* *Xn(f)* и *Yn(f)* *80 циклов* – работа новых функций *BISECT2()/HORDA2().* На каждом шаге работа цикла до его окончания выводится значение переменной *X* и соответствующее ему *F(X);*
* *Xn(f)* и *Yn(f) с возмущением* – к классическим значением применяется округлениe*;*
* *Delta* – значение получается вычитание из последнего значения в столбце *Xn(f) 80 циклов* строчек, соотвествующих столбцу *Xn(f) с возмущением*;
* *Eps/delta* – используемое значение *eps* в данном случае делиться на полученные значения *delta*;
* *Заключение* – осуществляется проверка обусловленности функции. Если полученное раннее значение *v\_del* меньше получившегося *eps/delta* функция хорошо обусловлена, иначе плохо.

**Таблица по методу бисекции.**

с = 3, v\_del = 6,05E-31

* Eps = 0,01

Таблица 5 – работа метода с округлением до 2 знаков после запятой.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **Xn классическое** | **Yn классическое** | **Xn(f) 80 циклов** | **Yn (f) 80 циклов** | **Xn(f) с возмущением** | **Yn (f) с возмущением** | **delta** | **eps/delta** | **Заключение** |
| 0 | 3,5 | -1,10289 | 3,5 | -1,10289 | 3,5 | -1,11 | 0,16519 | 0,06053635208 | ХОРОШО |
| 1 | 3,75 | 4,97315 | 3,75 | 4,97315 | 3,75 | 4,97 | 0,08481 | 0,1179106237 | ХОРОШО |
| 2 | 3,625 | -7,31377 | 3,625 | -7,31377 | 3,62 | -7,32 | 0,04519 | 0,2212878956 | ХОРОШО |
| 3 | 3,6875 | 15,9531 | 3,6875 | 15,9531 | 3,68 | 15,95 | 0,01481 | 0,6752194463 | ХОРОШО |
| 4 | 3,65625 | -36,2841 | 3,65625 | -36,2841 |  |  |  |  |  |
| 5 | 3,67188 | 50,8769 | 3,67188 | 50,8769 |  |  |  |  |  |
| 6 |  |  | 3,66406 | -294,266 |  |  |  |  |  |
| 7 |  |  | 3,66797 | 121,021 |  |  |  |  |  |
| 8 |  |  | 3,66602 | 405,435 |  |  |  |  |  |
| 9 |  |  | 3,66504 | -2186,7 |  |  |  |  |  |
| 10 |  |  | 3,66553 | 993,316 |  |  |  |  |  |
| 11 |  |  | 3,66528 | 3633,11 |  |  |  |  |  |
| 12 |  |  | 3,66516 | -11001,4 |  |  |  |  |  |
| 13 |  |  | 3,66522 | 10845,1 |  |  |  |  |  |
| 14 |  |  | 3,66519 | 1,51E+06 |  |  |  |  |  |
| 15 |  |  | 3,66518 | -22165,7 |  |  |  |  |  |
| 16 |  |  | 3,66518 | -44994,2 |  |  |  |  |  |
| 17 |  |  | 3,66519 | -92759,2 |  |  |  |  |  |
| 18 |  |  | 3,66519 | -197688 |  |  |  |  |  |
| 19 |  |  | 3,66519 | -455079 |  |  |  |  |  |
| 20 |  |  | 3,66519 | -1,30E+06 |  |  |  |  |  |
| 21 |  |  | 3,66519 | -1,94E+07 |  |  |  |  |  |
| 22 |  |  | 3,66519 | 3,27E+06 |  |  |  |  |  |
| 23 |  |  | 3,66519 | 7,86E+06 |  |  |  |  |  |
| 24 |  |  | 3,66519 | 2,65E+07 |  |  |  |  |  |
| 25 |  |  | 3,66519 | -1,44E+08 |  |  |  |  |  |
| 26 |  |  | 3,66519 | 6,49E+07 |  |  |  |  |  |
| 27 |  |  | 3,66519 | 2,36E+08 |  |  |  |  |  |
| 28 |  |  | 3,66519 | -7,37E+08 |  |  |  |  |  |
| 29 |  |  | 3,66519 | 6,96E+08 |  |  |  |  |  |
| 30 |  |  | 3,66519 | 2,48E+10 |  |  |  |  |  |
| 31 |  |  | 3,66519 | -1,52E+09 |  |  |  |  |  |
| 32 |  |  | 3,66519 | -3,24E+09 |  |  |  |  |  |
| 33 |  |  | 3,66519 | -7,44E+09 |  |  |  |  |  |
| 34 |  |  | 3,66519 | -2,13E+10 |  |  |  |  |  |
| 35 |  |  | 3,66519 | -2,95E+11 |  |  |  |  |  |
| 36 |  |  | 3,66519 | 5,42E+10 |  |  |  |  |  |
| 37 |  |  | 3,66519 | 1,33E+11 |  |  |  |  |  |
| 38 |  |  | 3,66519 | 4,84E+11 |  |  |  |  |  |
| 39 |  |  | 3,66519 | -1,51E+12 |  |  |  |  |  |
| 40 |  |  | 3,66519 | 1,43E+12 |  |  |  |  |  |
| 41 |  |  | 3,66519 | 5,23E+13 |  |  |  |  |  |
| 42 |  |  | 3,66519 | -3,11E+12 |  |  |  |  |  |
| 43 |  |  | 3,66519 | -6,60E+12 |  |  |  |  |  |
| 44 |  |  | 3,66519 | -1,51E+13 |  |  |  |  |  |
| 45 |  |  | 3,66519 | -4,25E+13 |  |  |  |  |  |
| 46 |  |  | 3,66519 | -4,54E+14 |  |  |  |  |  |
| 47 |  |  | 3,66519 | 1,18E+14 |  |  |  |  |  |
| 48 |  |  | 3,66519 | 3,20E+14 |  |  |  |  |  |
| 49 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 50 |  |  | 3,66519 | -2,33E+15 |  |  |  |  |  |
| 51 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 52 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 53 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 54 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 55 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 56 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 57 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 58 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 59 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 60 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 61 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 62 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 63 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 64 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 65 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 66 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 67 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 68 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 69 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 70 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 71 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 72 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 73 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 74 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 75 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 76 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 77 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 78 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 79 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |

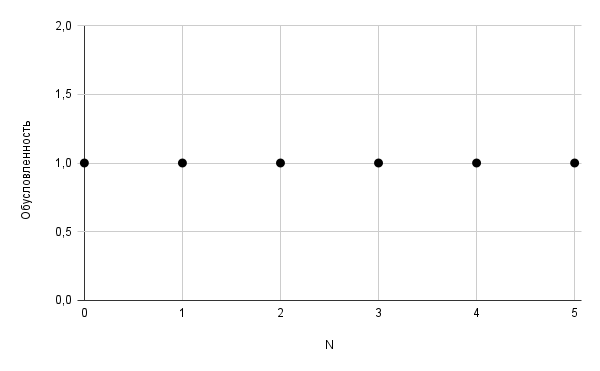


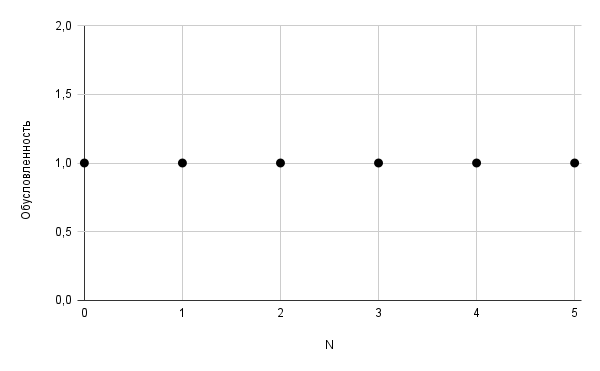
Рисунок 5 – График зависимости N, с которого задача стала хорошо обусловленной от разряда, подвергшегося возмущению. На графике 0 соответствует плохой обусловленности, 1 – хорошей.

Вывод: с первой же итерации задача является хорошо обусловленной.

* Eps = 0,001

Таблица 6 – работа метода с округлением до 3 знаков после запятой.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **Xn классическое** | **Yn классическое** | **Xn(f) 80 циклов** | **Yn (f) 80 циклов** | **Xn(f) с возмущением** | **Yn (f) с возмущением** | **delta** | **eps/delta** | **Заключение** |
| 0 | 3,5 | -1,10289 | 3,5 | -1,10289 | 3,5 | -1,103 | 0,16519 | 0,006053635208 | ХОРОШО |
| 1 | 3,75 | 4,97315 | 3,75 | 4,97315 | 3,75 | 4,973 | 0,08481 | 0,01179106237 | ХОРОШО |
| 2 | 3,625 | -7,31377 | 3,625 | -7,31377 | 3,625 | -7,314 | 0,04019 | 0,0248818114 | ХОРОШО |
| 3 | 3,6875 | 15,9531 | 3,6875 | 15,9531 | 3,687 | 15,953 |  |  |  |
| 4 | 3,65625 | -36,2841 | 3,65625 | -36,2841 | 3,656 | -36,285 |  |  |  |
| 5 | 3,67188 | 50,8769 | 3,67188 | 50,8769 | 3,671 | 50,876 |  |  |  |
| 6 |  |  | 3,66406 | -294,266 |  |  |  |  |  |
| 7 |  |  | 3,66797 | 121,021 |  |  |  |  |  |
| 8 |  |  | 3,66602 | 405,435 |  |  |  |  |  |
| 9 |  |  | 3,66504 | -2186,7 |  |  |  |  |  |
| 10 |  |  | 3,66553 | 993,316 |  |  |  |  |  |
| 11 |  |  | 3,66528 | 3633,11 |  |  |  |  |  |
| 12 |  |  | 3,66516 | -11001,4 |  |  |  |  |  |
| 13 |  |  | 3,66522 | 10845,1 |  |  |  |  |  |
| 14 |  |  | 3,66519 | 1,51E+06 |  |  |  |  |  |
| 15 |  |  | 3,66518 | -22165,7 |  |  |  |  |  |
| 16 |  |  | 3,66518 | -44994,2 |  |  |  |  |  |
| 17 |  |  | 3,66519 | -92759,2 |  |  |  |  |  |
| 18 |  |  | 3,66519 | -197688 |  |  |  |  |  |
| 19 |  |  | 3,66519 | -455079 |  |  |  |  |  |
| 20 |  |  | 3,66519 | -1,30E+06 |  |  |  |  |  |
| 21 |  |  | 3,66519 | -1,94E+07 |  |  |  |  |  |
| 22 |  |  | 3,66519 | 3,27E+06 |  |  |  |  |  |
| 23 |  |  | 3,66519 | 7,86E+06 |  |  |  |  |  |
| 24 |  |  | 3,66519 | 2,65E+07 |  |  |  |  |  |
| 25 |  |  | 3,66519 | -1,44E+08 |  |  |  |  |  |
| 26 |  |  | 3,66519 | 6,49E+07 |  |  |  |  |  |
| 27 |  |  | 3,66519 | 2,36E+08 |  |  |  |  |  |
| 28 |  |  | 3,66519 | -7,37E+08 |  |  |  |  |  |
| 29 |  |  | 3,66519 | 6,96E+08 |  |  |  |  |  |
| 30 |  |  | 3,66519 | 2,48E+10 |  |  |  |  |  |
| 31 |  |  | 3,66519 | -1,52E+09 |  |  |  |  |  |
| 32 |  |  | 3,66519 | -3,24E+09 |  |  |  |  |  |
| 33 |  |  | 3,66519 | -7,44E+09 |  |  |  |  |  |
| 34 |  |  | 3,66519 | -2,13E+10 |  |  |  |  |  |
| 35 |  |  | 3,66519 | -2,95E+11 |  |  |  |  |  |
| 36 |  |  | 3,66519 | 5,42E+10 |  |  |  |  |  |
| 37 |  |  | 3,66519 | 1,33E+11 |  |  |  |  |  |
| 38 |  |  | 3,66519 | 4,84E+11 |  |  |  |  |  |
| 39 |  |  | 3,66519 | -1,51E+12 |  |  |  |  |  |
| 40 |  |  | 3,66519 | 1,43E+12 |  |  |  |  |  |
| 41 |  |  | 3,66519 | 5,23E+13 |  |  |  |  |  |
| 42 |  |  | 3,66519 | -3,11E+12 |  |  |  |  |  |
| 43 |  |  | 3,66519 | -6,60E+12 |  |  |  |  |  |
| 44 |  |  | 3,66519 | -1,51E+13 |  |  |  |  |  |
| 45 |  |  | 3,66519 | -4,25E+13 |  |  |  |  |  |
| 46 |  |  | 3,66519 | -4,54E+14 |  |  |  |  |  |
| 47 |  |  | 3,66519 | 1,18E+14 |  |  |  |  |  |
| 48 |  |  | 3,66519 | 3,20E+14 |  |  |  |  |  |
| 49 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 50 |  |  | 3,66519 | -2,33E+15 |  |  |  |  |  |
| 51 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 52 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 53 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 54 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 55 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 56 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 57 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 58 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 59 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 60 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 61 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 62 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 63 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 64 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 65 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 66 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 67 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 68 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 69 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 70 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 71 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 72 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 73 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 74 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 75 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 76 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 77 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 78 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 79 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |

Рисунок 6 – График зависимости N, с которого задача стала хорошо обусловленной от разряда, подвергшегося возмущению. На графике 0 соответствует плохой обусловленности, 1 – хорошей.

Вывод: с первой же итерации задача является хорошо обусловленной.

* Eps = 0,0001

Таблица 7 – работа метода с округлением до 4 знаков после запятой.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **Xn классическое** | **Yn классическое** | **Xn(f) 80 циклов** | **Yn (f) 80 циклов** | **Xn(f) с возмущением** | **Yn (f) с возмущением** | **delta** | **eps/delta** | **Заключение** |
| 0 | 3,5 | -1,10289 | 3,5 | -1,10289 | 3,5 | -1,1029 | 0,16519 | 0,0006053635208 | ХОРОШО |
| 1 | 3,75 | 4,97315 | 3,75 | 4,97315 | 3,75 | 4,9731 | 0,08481 | 0,001179106237 | ХОРОШО |
| 2 | 3,625 | -7,31377 | 3,625 | -7,31377 | 3,625 | -7,3138 | 0,04019 | 0,00248818114 | ХОРОШО |
| 3 | 3,6875 | 15,9531 | 3,6875 | 15,9531 | 3,6875 | 15,9531 | 0,02231 | 0,004482294935 | ХОРОШО |
| 4 | 3,65625 | -36,2841 | 3,65625 | -36,2841 | 3,6562 |  |  |  |  |
| 5 | 3,67188 | 50,8769 | 3,67188 | 50,8769 | 3,6718 |  |  |  |  |
| 6 |  |  | 3,66406 | -294,266 |  |  |  |  |  |
| 7 |  |  | 3,66797 | 121,021 |  |  |  |  |  |
| 8 |  |  | 3,66602 | 405,435 |  |  |  |  |  |
| 9 |  |  | 3,66504 | -2186,7 |  |  |  |  |  |
| 10 |  |  | 3,66553 | 993,316 |  |  |  |  |  |
| 11 |  |  | 3,66528 | 3633,11 |  |  |  |  |  |
| 12 |  |  | 3,66516 | -11001,4 |  |  |  |  |  |
| 13 |  |  | 3,66522 | 10845,1 |  |  |  |  |  |
| 14 |  |  | 3,66519 | 1,51E+06 |  |  |  |  |  |
| 15 |  |  | 3,66518 | -22165,7 |  |  |  |  |  |
| 16 |  |  | 3,66518 | -44994,2 |  |  |  |  |  |
| 17 |  |  | 3,66519 | -92759,2 |  |  |  |  |  |
| 18 |  |  | 3,66519 | -197688 |  |  |  |  |  |
| 19 |  |  | 3,66519 | -455079 |  |  |  |  |  |
| 20 |  |  | 3,66519 | -1,30E+06 |  |  |  |  |  |
| 21 |  |  | 3,66519 | -1,94E+07 |  |  |  |  |  |
| 22 |  |  | 3,66519 | 3,27E+06 |  |  |  |  |  |
| 23 |  |  | 3,66519 | 7,86E+06 |  |  |  |  |  |
| 24 |  |  | 3,66519 | 2,65E+07 |  |  |  |  |  |
| 25 |  |  | 3,66519 | -1,44E+08 |  |  |  |  |  |
| 26 |  |  | 3,66519 | 6,49E+07 |  |  |  |  |  |
| 27 |  |  | 3,66519 | 2,36E+08 |  |  |  |  |  |
| 28 |  |  | 3,66519 | -7,37E+08 |  |  |  |  |  |
| 29 |  |  | 3,66519 | 6,96E+08 |  |  |  |  |  |
| 30 |  |  | 3,66519 | 2,48E+10 |  |  |  |  |  |
| 31 |  |  | 3,66519 | -1,52E+09 |  |  |  |  |  |
| 32 |  |  | 3,66519 | -3,24E+09 |  |  |  |  |  |
| 33 |  |  | 3,66519 | -7,44E+09 |  |  |  |  |  |
| 34 |  |  | 3,66519 | -2,13E+10 |  |  |  |  |  |
| 35 |  |  | 3,66519 | -2,95E+11 |  |  |  |  |  |
| 36 |  |  | 3,66519 | 5,42E+10 |  |  |  |  |  |
| 37 |  |  | 3,66519 | 1,33E+11 |  |  |  |  |  |
| 38 |  |  | 3,66519 | 4,84E+11 |  |  |  |  |  |
| 39 |  |  | 3,66519 | -1,51E+12 |  |  |  |  |  |
| 40 |  |  | 3,66519 | 1,43E+12 |  |  |  |  |  |
| 41 |  |  | 3,66519 | 5,23E+13 |  |  |  |  |  |
| 42 |  |  | 3,66519 | -3,11E+12 |  |  |  |  |  |
| 43 |  |  | 3,66519 | -6,60E+12 |  |  |  |  |  |
| 44 |  |  | 3,66519 | -1,51E+13 |  |  |  |  |  |
| 45 |  |  | 3,66519 | -4,25E+13 |  |  |  |  |  |
| 46 |  |  | 3,66519 | -4,54E+14 |  |  |  |  |  |
| 47 |  |  | 3,66519 | 1,18E+14 |  |  |  |  |  |
| 48 |  |  | 3,66519 | 3,20E+14 |  |  |  |  |  |
| 49 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 50 |  |  | 3,66519 | -2,33E+15 |  |  |  |  |  |
| 51 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 52 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 53 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 54 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 55 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 56 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 57 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 58 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 59 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 60 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 61 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 62 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 63 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 64 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 65 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 66 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 67 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 68 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 69 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 70 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 71 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 72 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 73 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 74 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 75 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 76 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 77 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 78 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |
| 79 |  |  | 3,66519 | 7,42E+14 |  |  |  |  |  |

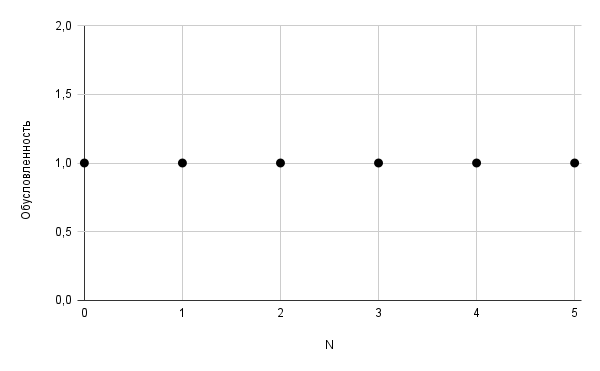


Рисунок 7 – График зависимости N, с которого задача стала хорошо обусловленной от разряда, подвергшегося возмущению. На графике 0 соответствует плохой обусловленности, 1 – хорошей.

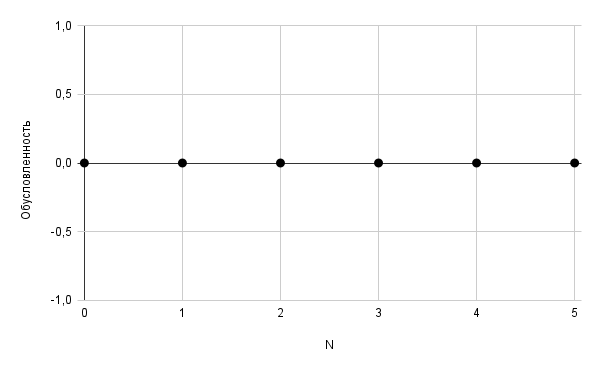
**Таблица по методу хорд.**

с = 3, v\_del = 1,76E+01

* Eps = 0,01

Таблица 8 – работа метода с округлением до 2 знаков после запятой.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **Xn классическое** | **Yn классическое** | **Xn(f) 80 циклов** | **Yn (f) 80 циклов** | **Xn(f) с возмущением** | **Yn (f) с возмущением** | **delta** | **eps/delta** | **Заключение** |
| 0 | 3,04273 | -0,0456526 | 3,04273 | -0,0456526 | 3,04 | -0,05 | 0,09527 | 0,1049648368 | ПЛОХО |
| 1 | 3,06232 | -0,0289588 | 3,06232 | -0,0289588 | 3,06 | -0,03 | 0,07527 | 0,1328550551 | ПЛОХО |
| 2 | 3,07459 | -0,0205495 | 3,07459 | -0,0205495 | 3,07 | -0,03 | 0,06527 | 0,1532097441 | ПЛОХО |
| 3 | 3,08321 | -0,0155372 | 3,08321 | -0,0155372 | 3,08 | -0,02 | 0,05527 | 0,1809299801 | ПЛОХО |
| 4 | 3,08968 | -0,0122498 | 3,08968 | -0,0122498 | 3,08 | -0,02 |  |  |  |
| 5 | 3,09476 | -0,00995277 | 3,09476 | -0,00995277 | 3,09 | -0,01 |  |  |  |
| 6 |  |  | 3,09886 | -0,0082732 |  |  |  |  |  |
| 7 |  |  | 3,10226 | -0,00700195 |  |  |  |  |  |
| 8 |  |  | 3,10513 | -0,00601324 |  |  |  |  |  |
| 9 |  |  | 3,10758 | -0,00522709 |  |  |  |  |  |
| 10 |  |  | 3,10971 | -0,00459047 |  |  |  |  |  |
| 11 |  |  | 3,11158 | -0,0040669 |  |  |  |  |  |
| 12 |  |  | 3,11323 | -0,00363057 |  |  |  |  |  |
| 13 |  |  | 3,1147 | -0,00326273 |  |  |  |  |  |
| 14 |  |  | 3,11602 | -0,00294951 |  |  |  |  |  |
| 15 |  |  | 3,11721 | -0,00268041 |  |  |  |  |  |
| 16 |  |  | 3,1183 | -0,00244737 |  |  |  |  |  |
| 17 |  |  | 3,11928 | -0,00224412 |  |  |  |  |  |
| 18 |  |  | 3,12019 | -0,00206572 |  |  |  |  |  |
| 19 |  |  | 3,12102 | -0,0019082 |  |  |  |  |  |
| 20 |  |  | 3,12178 | -0,0017684 |  |  |  |  |  |
| 21 |  |  | 3,12249 | -0,0016437 |  |  |  |  |  |
| 22 |  |  | 3,12315 | -0,00153198 |  |  |  |  |  |
| 23 |  |  | 3,12377 | -0,00143148 |  |  |  |  |  |
| 24 |  |  | 3,12434 | -0,00134072 |  |  |  |  |  |
| 25 |  |  | 3,12488 | -0,00125847 |  |  |  |  |  |
| 26 |  |  | 3,12538 | -0,00118368 |  |  |  |  |  |
| 27 |  |  | 3,12586 | -0,00111548 |  |  |  |  |  |
| 28 |  |  | 3,1263 | -0,00105309 |  |  |  |  |  |
| 29 |  |  | 3,12672 | -0,000995865 |  |  |  |  |  |
| 30 |  |  | 3,12712 | -0,00094325 |  |  |  |  |  |
| 31 |  |  | 3,1275 | -0,000894756 |  |  |  |  |  |
| 32 |  |  | 3,12785 | -0,000849958 |  |  |  |  |  |
| 33 |  |  | 3,12819 | -0,000808487 |  |  |  |  |  |
| 34 |  |  | 3,12852 | -0,000770018 |  |  |  |  |  |
| 35 |  |  | 3,12882 | -0,000734265 |  |  |  |  |  |
| 36 |  |  | 3,12912 | -0,000700976 |  |  |  |  |  |
| 37 |  |  | 3,12939 | -0,000669928 |  |  |  |  |  |
| 38 |  |  | 3,12966 | -0,000640922 |  |  |  |  |  |
| 39 |  |  | 3,12992 | -0,000613782 |  |  |  |  |  |
| 40 |  |  | 3,13016 | -0,000588349 |  |  |  |  |  |
| 41 |  |  | 3,1304 | -0,000564483 |  |  |  |  |  |
| 42 |  |  | 3,13062 | -0,000542055 |  |  |  |  |  |
| 43 |  |  | 3,13084 | -0,000520952 |  |  |  |  |  |
| 44 |  |  | 3,13104 | -0,000501071 |  |  |  |  |  |
| 45 |  |  | 3,13124 | -0,000482318 |  |  |  |  |  |
| 46 |  |  | 3,13143 | -0,000464609 |  |  |  |  |  |
| 47 |  |  | 3,13162 | -0,000447867 |  |  |  |  |  |
| 48 |  |  | 3,1318 | -0,000432023 |  |  |  |  |  |
| 49 |  |  | 3,13197 | -0,000417013 |  |  |  |  |  |
| 50 |  |  | 3,13213 | -0,000402779 |  |  |  |  |  |
| 51 |  |  | 3,13229 | -0,000389268 |  |  |  |  |  |
| 52 |  |  | 3,13245 | -0,000376432 |  |  |  |  |  |
| 53 |  |  | 3,1326 | -0,000364227 |  |  |  |  |  |
| 54 |  |  | 3,13274 | -0,00035261 |  |  |  |  |  |
| 55 |  |  | 3,13288 | -0,000341546 |  |  |  |  |  |
| 56 |  |  | 3,13302 | -0,000330998 |  |  |  |  |  |
| 57 |  |  | 3,13315 | -0,000320936 |  |  |  |  |  |
| 58 |  |  | 3,13328 | -0,000311329 |  |  |  |  |  |
| 59 |  |  | 3,1334 | -0,000302151 |  |  |  |  |  |
| 60 |  |  | 3,13352 | -0,000293376 |  |  |  |  |  |
| 61 |  |  | 3,13364 | -0,000284981 |  |  |  |  |  |
| 62 |  |  | 3,13375 | -0,000276944 |  |  |  |  |  |
| 63 |  |  | 3,13386 | -0,000269244 |  |  |  |  |  |
| 64 |  |  | 3,13397 | -0,000261864 |  |  |  |  |  |
| 65 |  |  | 3,13407 | -0,000254786 |  |  |  |  |  |
| 66 |  |  | 3,13417 | -0,000247992 |  |  |  |  |  |
| 67 |  |  | 3,13427 | -0,000241469 |  |  |  |  |  |
| 68 |  |  | 3,13436 | -0,000235202 |  |  |  |  |  |
| 69 |  |  | 3,13446 | -0,000229177 |  |  |  |  |  |
| 70 |  |  | 3,13455 | -0,000223382 |  |  |  |  |  |
| 71 |  |  | 3,13464 | -0,000217806 |  |  |  |  |  |
| 72 |  |  | 3,13472 | -0,000212437 |  |  |  |  |  |
| 73 |  |  | 3,13481 | -0,000207266 |  |  |  |  |  |
| 74 |  |  | 3,13489 | -0,000202283 |  |  |  |  |  |
| 75 |  |  | 3,13497 | -0,000197478 |  |  |  |  |  |
| 76 |  |  | 3,13505 | -0,000192844 |  |  |  |  |  |
| 77 |  |  | 3,13512 | -0,000188372 |  |  |  |  |  |
| 78 |  |  | 3,1352 | -0,000184055 |  |  |  |  |  |
| 79 |  |  | 3,13527 | -0,000179885 |  |  |  |  |  |

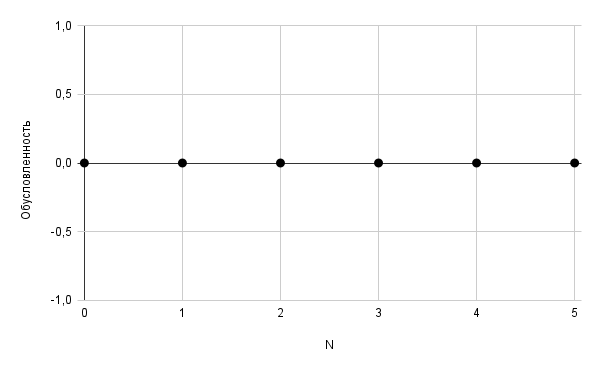
Рисунок 8 – График зависимости N, с которого задача стала хорошо обусловленной от разряда, подвергшегося возмущению. На графике 0 соответствует плохой обусловленности, 1 – хорошей.

Вывод: с первой же итерации задача является плохо обусловленной.

* Eps = 0,001

Таблица 9 – работа метода с округлением до 3 знаков после запятой.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **Xn классическое** | **Yn классическое** | **Xn(f) 80 циклов** | **Yn (f) 80 циклов** | **Xn(f) с возмущением** | **Yn (f) с возмущением** | **delta** | **eps/delta** | **Заключение** |
| 0 | 3,04273 | -0,0456526 | 3,04273 | -0,0456526 | 3,042 | -0,046 | 0,09327 | 0,1072156106 | ПЛОХО |
| 1 | 3,06232 | -0,0289588 | 3,06232 | -0,0289588 | 3,062 | -0,029 | 0,07327 | 0,1364815068 | ПЛОХО |
| 2 | 3,07459 | -0,0205495 | 3,07459 | -0,0205495 | 3,074 | -0,021 | 0,06127 | 0,1632120124 | ПЛОХО |
| 3 | 3,08321 | -0,0155372 | 3,08321 | -0,0155372 | 3,083 |  |  |  |  |
| 4 | 3,08968 | -0,0122498 | 3,08968 | -0,0122498 | 3,089 |  |  |  |  |
| 5 | 3,09476 | -0,00995277 | 3,09476 | -0,00995277 | 3,094 |  |  |  |  |
| 6 |  |  | 3,09886 | -0,0082732 |  |  |  |  |  |
| 7 |  |  | 3,10226 | -0,00700195 |  |  |  |  |  |
| 8 |  |  | 3,10513 | -0,00601324 |  |  |  |  |  |
| 9 |  |  | 3,10758 | -0,00522709 |  |  |  |  |  |
| 10 |  |  | 3,10971 | -0,00459047 |  |  |  |  |  |
| 11 |  |  | 3,11158 | -0,0040669 |  |  |  |  |  |
| 12 |  |  | 3,11323 | -0,00363057 |  |  |  |  |  |
| 13 |  |  | 3,1147 | -0,00326273 |  |  |  |  |  |
| 14 |  |  | 3,11602 | -0,00294951 |  |  |  |  |  |
| 15 |  |  | 3,11721 | -0,00268041 |  |  |  |  |  |
| 16 |  |  | 3,1183 | -0,00244737 |  |  |  |  |  |
| 17 |  |  | 3,11928 | -0,00224412 |  |  |  |  |  |
| 18 |  |  | 3,12019 | -0,00206572 |  |  |  |  |  |
| 19 |  |  | 3,12102 | -0,0019082 |  |  |  |  |  |
| 20 |  |  | 3,12178 | -0,0017684 |  |  |  |  |  |
| 21 |  |  | 3,12249 | -0,0016437 |  |  |  |  |  |
| 22 |  |  | 3,12315 | -0,00153198 |  |  |  |  |  |
| 23 |  |  | 3,12377 | -0,00143148 |  |  |  |  |  |
| 24 |  |  | 3,12434 | -0,00134072 |  |  |  |  |  |
| 25 |  |  | 3,12488 | -0,00125847 |  |  |  |  |  |
| 26 |  |  | 3,12538 | -0,00118368 |  |  |  |  |  |
| 27 |  |  | 3,12586 | -0,00111548 |  |  |  |  |  |
| 28 |  |  | 3,1263 | -0,00105309 |  |  |  |  |  |
| 29 |  |  | 3,12672 | -0,000995865 |  |  |  |  |  |
| 30 |  |  | 3,12712 | -0,00094325 |  |  |  |  |  |
| 31 |  |  | 3,1275 | -0,000894756 |  |  |  |  |  |
| 32 |  |  | 3,12785 | -0,000849958 |  |  |  |  |  |
| 33 |  |  | 3,12819 | -0,000808487 |  |  |  |  |  |
| 34 |  |  | 3,12852 | -0,000770018 |  |  |  |  |  |
| 35 |  |  | 3,12882 | -0,000734265 |  |  |  |  |  |
| 36 |  |  | 3,12912 | -0,000700976 |  |  |  |  |  |
| 37 |  |  | 3,12939 | -0,000669928 |  |  |  |  |  |
| 38 |  |  | 3,12966 | -0,000640922 |  |  |  |  |  |
| 39 |  |  | 3,12992 | -0,000613782 |  |  |  |  |  |
| 40 |  |  | 3,13016 | -0,000588349 |  |  |  |  |  |
| 41 |  |  | 3,1304 | -0,000564483 |  |  |  |  |  |
| 42 |  |  | 3,13062 | -0,000542055 |  |  |  |  |  |
| 43 |  |  | 3,13084 | -0,000520952 |  |  |  |  |  |
| 44 |  |  | 3,13104 | -0,000501071 |  |  |  |  |  |
| 45 |  |  | 3,13124 | -0,000482318 |  |  |  |  |  |
| 46 |  |  | 3,13143 | -0,000464609 |  |  |  |  |  |
| 47 |  |  | 3,13162 | -0,000447867 |  |  |  |  |  |
| 48 |  |  | 3,1318 | -0,000432023 |  |  |  |  |  |
| 49 |  |  | 3,13197 | -0,000417013 |  |  |  |  |  |
| 50 |  |  | 3,13213 | -0,000402779 |  |  |  |  |  |
| 51 |  |  | 3,13229 | -0,000389268 |  |  |  |  |  |
| 52 |  |  | 3,13245 | -0,000376432 |  |  |  |  |  |
| 53 |  |  | 3,1326 | -0,000364227 |  |  |  |  |  |
| 54 |  |  | 3,13274 | -0,00035261 |  |  |  |  |  |
| 55 |  |  | 3,13288 | -0,000341546 |  |  |  |  |  |
| 56 |  |  | 3,13302 | -0,000330998 |  |  |  |  |  |
| 57 |  |  | 3,13315 | -0,000320936 |  |  |  |  |  |
| 58 |  |  | 3,13328 | -0,000311329 |  |  |  |  |  |
| 59 |  |  | 3,1334 | -0,000302151 |  |  |  |  |  |
| 60 |  |  | 3,13352 | -0,000293376 |  |  |  |  |  |
| 61 |  |  | 3,13364 | -0,000284981 |  |  |  |  |  |
| 62 |  |  | 3,13375 | -0,000276944 |  |  |  |  |  |
| 63 |  |  | 3,13386 | -0,000269244 |  |  |  |  |  |
| 64 |  |  | 3,13397 | -0,000261864 |  |  |  |  |  |
| 65 |  |  | 3,13407 | -0,000254786 |  |  |  |  |  |
| 66 |  |  | 3,13417 | -0,000247992 |  |  |  |  |  |
| 67 |  |  | 3,13427 | -0,000241469 |  |  |  |  |  |
| 68 |  |  | 3,13436 | -0,000235202 |  |  |  |  |  |
| 69 |  |  | 3,13446 | -0,000229177 |  |  |  |  |  |
| 70 |  |  | 3,13455 | -0,000223382 |  |  |  |  |  |
| 71 |  |  | 3,13464 | -0,000217806 |  |  |  |  |  |
| 72 |  |  | 3,13472 | -0,000212437 |  |  |  |  |  |
| 73 |  |  | 3,13481 | -0,000207266 |  |  |  |  |  |
| 74 |  |  | 3,13489 | -0,000202283 |  |  |  |  |  |
| 75 |  |  | 3,13497 | -0,000197478 |  |  |  |  |  |
| 76 |  |  | 3,13505 | -0,000192844 |  |  |  |  |  |
| 77 |  |  | 3,13512 | -0,000188372 |  |  |  |  |  |
| 78 |  |  | 3,1352 | -0,000184055 |  |  |  |  |  |
| 79 |  |  | 3,13527 | -0,000179885 |  |  |  |  |  |

Рисунок 9 – График зависимости N, с которого задача стала хорошо обусловленной от разряда, подвергшегося возмущению. На графике 0 соответствует плохой обусловленности, 1 – хорошей.

Вывод: с первой же итерации задача является плохо обусловленной.

* Eps = 0,0001

Таблица 10 – работа метода с округлением до 4 знаков после запятой.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **Xn классическое** | **Yn классическое** | **Xn(f) 80 циклов** | **Yn (f) 80 циклов** | **Xn(f) с возмущением** | **Yn (f) с возмущением** | **delta** | **eps/delta** | **Заключение** |
| 0 | 3,04273 | -0,0456526 | 3,04273 | -0,0456526 | 3,0427 | -0,0457 | 0,09257 | 0,001080263584 | ПЛОХО |
| 1 | 3,06232 | -0,0289588 | 3,06232 | -0,0289588 | 3,0623 | -0,029 | 0,07297 | 0,001370426203 | ПЛОХО |
| 2 | 3,07459 | -0,0205495 | 3,07459 | -0,0205495 | 3,0745 | -0,0206 | 0,06077 | 0,001645548791 | ПЛОХО |
| 3 | 3,08321 | -0,0155372 | 3,08321 | -0,0155372 | 3,0832 | -0,0156 | 0,05207 | 0,001920491646 | ПЛОХО |
| 4 | 3,08968 | -0,0122498 | 3,08968 | -0,0122498 | 3,0896 |  |  |  |  |
| 5 | 3,09476 | -0,00995277 | 3,09476 | -0,00995277 | 3,0947 |  |  |  |  |
| 6 |  |  | 3,09886 | -0,0082732 |  |  |  |  |  |
| 7 |  |  | 3,10226 | -0,00700195 |  |  |  |  |  |
| 8 |  |  | 3,10513 | -0,00601324 |  |  |  |  |  |
| 9 |  |  | 3,10758 | -0,00522709 |  |  |  |  |  |
| 10 |  |  | 3,10971 | -0,00459047 |  |  |  |  |  |
| 11 |  |  | 3,11158 | -0,0040669 |  |  |  |  |  |
| 12 |  |  | 3,11323 | -0,00363057 |  |  |  |  |  |
| 13 |  |  | 3,1147 | -0,00326273 |  |  |  |  |  |
| 14 |  |  | 3,11602 | -0,00294951 |  |  |  |  |  |
| 15 |  |  | 3,11721 | -0,00268041 |  |  |  |  |  |
| 16 |  |  | 3,1183 | -0,00244737 |  |  |  |  |  |
| 17 |  |  | 3,11928 | -0,00224412 |  |  |  |  |  |
| 18 |  |  | 3,12019 | -0,00206572 |  |  |  |  |  |
| 19 |  |  | 3,12102 | -0,0019082 |  |  |  |  |  |
| 20 |  |  | 3,12178 | -0,0017684 |  |  |  |  |  |
| 21 |  |  | 3,12249 | -0,0016437 |  |  |  |  |  |
| 22 |  |  | 3,12315 | -0,00153198 |  |  |  |  |  |
| 23 |  |  | 3,12377 | -0,00143148 |  |  |  |  |  |
| 24 |  |  | 3,12434 | -0,00134072 |  |  |  |  |  |
| 25 |  |  | 3,12488 | -0,00125847 |  |  |  |  |  |
| 26 |  |  | 3,12538 | -0,00118368 |  |  |  |  |  |
| 27 |  |  | 3,12586 | -0,00111548 |  |  |  |  |  |
| 28 |  |  | 3,1263 | -0,00105309 |  |  |  |  |  |
| 29 |  |  | 3,12672 | -0,000995865 |  |  |  |  |  |
| 30 |  |  | 3,12712 | -0,00094325 |  |  |  |  |  |
| 31 |  |  | 3,1275 | -0,000894756 |  |  |  |  |  |
| 32 |  |  | 3,12785 | -0,000849958 |  |  |  |  |  |
| 33 |  |  | 3,12819 | -0,000808487 |  |  |  |  |  |
| 34 |  |  | 3,12852 | -0,000770018 |  |  |  |  |  |
| 35 |  |  | 3,12882 | -0,000734265 |  |  |  |  |  |
| 36 |  |  | 3,12912 | -0,000700976 |  |  |  |  |  |
| 37 |  |  | 3,12939 | -0,000669928 |  |  |  |  |  |
| 38 |  |  | 3,12966 | -0,000640922 |  |  |  |  |  |
| 39 |  |  | 3,12992 | -0,000613782 |  |  |  |  |  |
| 40 |  |  | 3,13016 | -0,000588349 |  |  |  |  |  |
| 41 |  |  | 3,1304 | -0,000564483 |  |  |  |  |  |
| 42 |  |  | 3,13062 | -0,000542055 |  |  |  |  |  |
| 43 |  |  | 3,13084 | -0,000520952 |  |  |  |  |  |
| 44 |  |  | 3,13104 | -0,000501071 |  |  |  |  |  |
| 45 |  |  | 3,13124 | -0,000482318 |  |  |  |  |  |
| 46 |  |  | 3,13143 | -0,000464609 |  |  |  |  |  |
| 47 |  |  | 3,13162 | -0,000447867 |  |  |  |  |  |
| 48 |  |  | 3,1318 | -0,000432023 |  |  |  |  |  |
| 49 |  |  | 3,13197 | -0,000417013 |  |  |  |  |  |
| 50 |  |  | 3,13213 | -0,000402779 |  |  |  |  |  |
| 51 |  |  | 3,13229 | -0,000389268 |  |  |  |  |  |
| 52 |  |  | 3,13245 | -0,000376432 |  |  |  |  |  |
| 53 |  |  | 3,1326 | -0,000364227 |  |  |  |  |  |
| 54 |  |  | 3,13274 | -0,00035261 |  |  |  |  |  |
| 55 |  |  | 3,13288 | -0,000341546 |  |  |  |  |  |
| 56 |  |  | 3,13302 | -0,000330998 |  |  |  |  |  |
| 57 |  |  | 3,13315 | -0,000320936 |  |  |  |  |  |
| 58 |  |  | 3,13328 | -0,000311329 |  |  |  |  |  |
| 59 |  |  | 3,1334 | -0,000302151 |  |  |  |  |  |
| 60 |  |  | 3,13352 | -0,000293376 |  |  |  |  |  |
| 61 |  |  | 3,13364 | -0,000284981 |  |  |  |  |  |
| 62 |  |  | 3,13375 | -0,000276944 |  |  |  |  |  |
| 63 |  |  | 3,13386 | -0,000269244 |  |  |  |  |  |
| 64 |  |  | 3,13397 | -0,000261864 |  |  |  |  |  |
| 65 |  |  | 3,13407 | -0,000254786 |  |  |  |  |  |
| 66 |  |  | 3,13417 | -0,000247992 |  |  |  |  |  |
| 67 |  |  | 3,13427 | -0,000241469 |  |  |  |  |  |
| 68 |  |  | 3,13436 | -0,000235202 |  |  |  |  |  |
| 69 |  |  | 3,13446 | -0,000229177 |  |  |  |  |  |
| 70 |  |  | 3,13455 | -0,000223382 |  |  |  |  |  |
| 71 |  |  | 3,13464 | -0,000217806 |  |  |  |  |  |
| 72 |  |  | 3,13472 | -0,000212437 |  |  |  |  |  |
| 73 |  |  | 3,13481 | -0,000207266 |  |  |  |  |  |
| 74 |  |  | 3,13489 | -0,000202283 |  |  |  |  |  |
| 75 |  |  | 3,13497 | -0,000197478 |  |  |  |  |  |
| 76 |  |  | 3,13505 | -0,000192844 |  |  |  |  |  |
| 77 |  |  | 3,13512 | -0,000188372 |  |  |  |  |  |
| 78 |  |  | 3,1352 | -0,000184055 |  |  |  |  |  |
| 79 |  |  | 3,13527 | -0,000179885 |  |  |  |  |  |

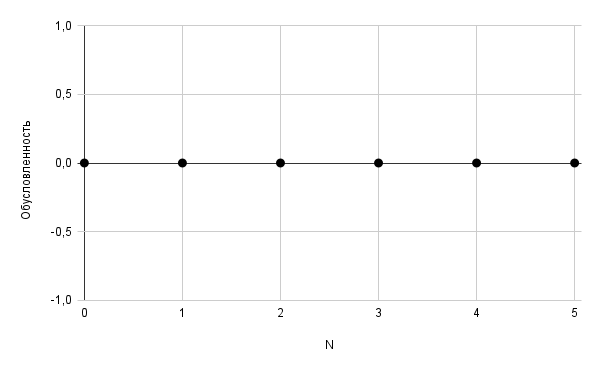


Рисунок 10 – График зависимости N, с которого задача стала хорошо обусловленной от разряда, подвергшегося возмущению. На графике 0 соответствует плохой обусловленности, 1 – хорошей.

Вывод: с первой же итерации задача является плохо обусловленной.

## Общие выводы. Сравнение двух методов.

При выбранных значениях c, eps и интервала метод бисекции и метод хорд выполняют вычисления за одинаковое число итераций.

Обусловленность для бисекции при каждом из трех возмущений всегда хорошая, для метода хорд же всегда плохая.

Скорость сходимости метода бисекции невысока. Для достижения точности *ε* необходимо совершить порядка итераций

Метод хорд обладает линейной сходимостью, которая зависит от выбора начального приближения .

# приложение

#include <iostream>

#include <cmath>

using namespace std;

double delta, eps, c = 3;

int n;

double Round\_Truncation(double Y) {

if (Y >= 0) {

return floor(Y);//round down

} else {

return ceil(Y);//round up

}

}

double Derivative(double x){

return (c\*sin(c\*x))/(pow(cos(c\*x), 2));

}

double F(double x){

//return (1/(cos(c\*x)))+1;

return (1/(cos(c\*Round\_Truncation(x\*10000)/10000)))+1;

}

double Round(double X, double Delta) {

if (Delta <= 1E-9){

puts("Неверно задана точность округления\n");

exit(1);

}

if (X>0.0)

return (Delta\*(long((X / Delta) + 0.5)));

else

return (Delta\*(long((X / Delta) - 0.5)));

}

double BISECT(double Left, double Right, double Eps, int &N){

double E = fabs(Eps)\*2.0;

double FLeft = F(Left);

double FRight = F(Right);

double X = (Left + Right) / 2.0;

double Y;

if (FLeft\*FRight>0.0) {

puts("neverno zadan interval\n");

exit(1);

}

if (Eps <= 0.0) {

puts("neverno zadana tochnost\n");

exit(1);

}

N = 0;

if (FLeft == 0.0)

return Left;

if (FRight == 0.0)

return Right;

while ((Right - Left) >= E) {

X = 0.5\*(Right + Left), eps;

//Y = F(X);

Y = Round\_Truncation(F(X)\*10000)/10000;

X = Round\_Truncation(X\*10000)/10000;

std::cout<<X<<" "<<Y<<"\n";

if (Y == 0.0)

return (X);

if (Y\*FLeft < 0.0)

Right = X;

else {

Left = X;

FLeft = Y;

}

N++;

};

return(X);

}

double BISECT2(double Left, double Right, double Eps, int &N){

double E = fabs(Eps)\*2.0;

double FLeft = F(Left);

double FRight = F(Right);

double X = (Left + Right) / 2.0;

double Y;

if (FLeft\*FRight>0.0) {

puts("neverno zadan interval\n");

exit(1);

}

if (Eps <= 0.0) {

puts("neverno zadana tochnost\n");

exit(1);

}

N = 0;

if (FLeft == 0.0)

return Left;

if (FRight == 0.0)

return Right;

while (N < 80) {

X = 0.5\*(Right + Left);

Y = F(X);

if (Y == 0.0)

return (X);

if (Y\*FLeft < 0.0){

Right = X;

}

else {

Left = X;

FLeft = Y;

}

N++;

};

return(X);

}

double BISECT2\_1(double Left, double Right, double Eps, int &N, int v\_del){

double E = fabs(Eps)\*2.0;

double FLeft = F(Left);

double FRight = F(Right);

double X = (Left + Right) / 2.0;

double Y;

if (FLeft\*FRight>0.0) {

puts("neverno zadan interval\n");

exit(1);

}

if (Eps <= 0.0) {

puts("neverno zadana tochnost\n");

exit(1);

}

N = 0;

if (FLeft == 0.0)

return Left;

if (FRight == 0.0)

return Right;

//std::cout<<"N\tXn\tX\*n\tdel\_i\tConclusion\n";

while (N < 80) {

X = 0.5\*(Right + Left);

Y = F(X);

if (Y == 0.0)

return (X);

if (Y\*FLeft < 0.0){

Right = X;

}

else {

Left = X;

FLeft = Y;

}

//std::cout<<N<<"\t"<<X<<"\t"<<Outrage2(X)<<"\t"<<fabs(X-Outrage2(X))<<"\t";

//if (fabs(X-Outrage2(X)) <= v\_del\*eps)

//std::cout<<"GOOD\n";

//else

//std::cout<<"BAD\n";

N++;

};

return(X);

}

double HORDA(double Left, double Right, double Eps, int &N) {

double FLeft = F(Left);

double FRight = F(Right);

double X, Y;

if (FLeft \* FRight > 0.0) {

puts("Неверное задание интервала\n");

exit(1);

}

if (Eps <= 0.0) {

puts("Неверное задание точности\n");

exit(1);

}

N = 0;

if (FLeft == 0.0) {

return Left;

}

if (FRight == 0.0) {

return Right;

}

do {

X = Left - (Right - Left) \* FLeft / (FRight - FLeft);

X = Round\_Truncation(X\*10000)/10000;

//Y = F(X);

Y = Round\_Truncation(F(X)\*10000)/10000;

//std::cout<<X<<" "<<Y<<"\n";

std::cout<<Round\_Truncation(X\*10000)/10000<<" "<<Y<<"\n";

if (Y == 0.0) {

return X;

}

if (Y \* FLeft < 0.0) {

Right = X;

FRight = Y;

}

else {

Left = X;

FLeft = Y;

}

N++;

} while (fabs(Y) >= Eps);

return X;

}

double HORDA2(double Left, double Right, double Eps, int &N) {

double FLeft = F(Left);

double FRight = F(Right);

double X, Y;

if (FLeft \* FRight > 0.0) {

puts("Неверное задание интервала\n");

exit(1);

}

if (Eps <= 0.0) {

puts("Неверное задание точности\n");

exit(1);

}

N = 0;

if (FLeft == 0.0) {

return Left;

}

if (FRight == 0.0) {

return Right;

}

do {

X = Left - (Right - Left) \* FLeft / (FRight - FLeft);

Y = F(X);

if (Y == 0.0) {

return X;

}

if (Y \* FLeft < 0.0) {

Right = X;

FRight = Y;

}

else {

Left = X;

FLeft = Y;

}

//std::cout<<X<<" "<<F(X)<<"\n";

N++;

} while (N < 80);

return X;

}

double HORDA2\_1(double Left, double Right, double Eps, int &N, double v\_del) {

double FLeft = F(Left);

double FRight = F(Right);

double X, Y;

if (FLeft \* FRight > 0.0) {

puts("Неверное задание интервала\n");

exit(1);

}

if (Eps <= 0.0) {

puts("Неверное задание точности\n");

exit(1);

}

N = 0;

if (FLeft == 0.0) {

return Left;

}

if (FRight == 0.0) {

return Right;

}

do {

X = Left - (Right - Left) \* FLeft / (FRight - FLeft);

Y = F(X);

if (Y == 0.0) {

return X;

}

if (Y \* FLeft < 0.0) {

Right = X;

FRight = Y;

}

else {

Left = X;

FLeft = Y;

}

//std::cout<<X<<"\t"<<Outrage2(X)<<"\t"<<fabs(X-Outrage2(X))<<"\t";

//if (fabs(X-Outrage2(X)) <= v\_del\*eps)

//std::cout<<"GOOD\n";

//else

//std::cout<<"BAD\n";

N++;

} while (N < 80);

return X;

}

int main(){

setlocale(LC\_ALL, "Russian");

/\*std::cout<<"Часть 1 пункт 5\n";

c = 3;

printf("%6s\t%14s\t%14s\t%14s\t%14s\t%14s\t%14s\t%10s\t%10s\n", "c", "eps", "delta", "x", "f(x)", "v\_delta", "v\_delta\_max", "iter", "Conclusion");

for(eps = 0.000001, delta = 0.000001; eps <= 1 && delta <= 1; eps \*= 10, delta \*= 10){

double x = Round(BISECT(3, 4, eps, n), delta);

double derivative = fabs(Derivative(x));//вычисление абсолютного значения

double v = 1/derivative;

printf("%6f\t%6f\t%6f\t%6f\t%6f\t%6lf\t%6f\t%d\t", c, eps, delta, x, F(x), v, eps/delta, n);

if(v < eps/delta)

printf("%6s\n", "Good");

else

printf("%6s\n", "Bad");

}

std::cout<<"Часть 1 пункт 6\n";

c = 3;

eps = 0.1;

n = 0;

printf("%6s\t%14s\t%14s\t%14s\t%14s\t%14s\t%14s\t%10s\t%10s\n", "c", "eps", "delta", "x", "f(x)", "v\_delta", "v\_delta\_max", "iter", "Conclusion");

for(delta = 0.00001; delta <= 0.1; delta \*= 10){

double x = Round(BISECT(3, 4, eps, n), delta);

double derivative = fabs(Derivative(x));//вычисление абсолютного значения

double v = 1/derivative;

printf("%6f\t%6f\t%6f\t%6f\t%6f\t%6lf\t%6f\t%d\t", c, eps, delta, x, F(x), v, eps/delta, n);

if(v < eps/delta)

printf("%6s\n", "Good");

else

printf("%6s\n", "Bad");

}

//BISECT

c = 3;

eps = 0.01;

n = 0;

double X79 = BISECT2(3, 4, eps, n);

double FX79\_ = Derivative(X79);

double v\_del = 1/FX79\_;

c = 3;

eps = 0.01;

n = 0;

BISECT2\_1(3, 4, eps, n, v\_del);\*/

c = 3;

eps = 0.01;

n = 0;

BISECT(3, 4, eps, n);

//HORDA

/\*c = 3;

eps = 0.01;

n = 0;

double X79 = HORDA2(3, 4, eps, n);\*/

//double FX79\_ = Derivative(X79);

//double v\_del = 1/FX79\_;

//c = 3;

//eps = 0.01;

//n = 0;

//HORDA2\_1(3, 4, eps, n, v\_del);

//eps = 0.01;

//n = 0;

//HORDA(3, 4, eps, n);

return 0;

}