# 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.

### Цель практической работы

Целью работы является нахождение корня уравнения методом *бисекции* с заданной точностью *Eps*, исследование зависимости числа итераций 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 членов последовательности приближенного значения корня.

### Порядок выполнения первой части практической работы

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";

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;

# 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 .

### Цель практической работы

Целью работы является нахождение корня уравнения методом *хорд* с заданной точностью *Eps*, исследование скорости сходимости и обусловленности метода; исследование обусловленности метода без ограничения итераций заданной точностью *Eps*, когда рассчитывается членов последовательности приближенного значения корня.

### Порядок выполнения практической работы

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.

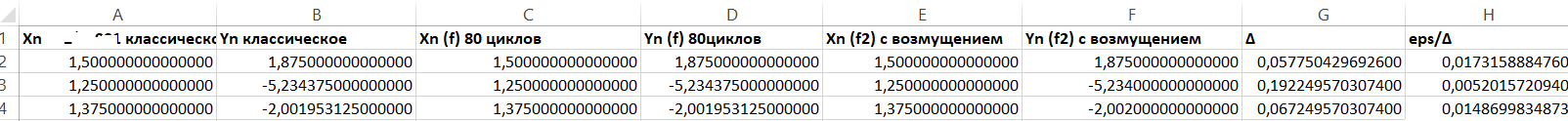
1. По аналогии со второй частью практической работы метода бисекции, провести исследование обусловленности задачи метода *хорд* при построении приближений к корню последовательности и приближенных значений итерационной последовательности после наложения шума на функцию.

**Задание:** используем методы бисекции и хорд. Метод хорд изучаем самостоятельно.

**Варианты** остались с практической 2. Eps или Delta задаем самостоятельно, указывая в работе.

Возмущения описываем в отчете подробно.

Формируем по каждому методу такие данные :

****

Вычисляем.

Повторяем столбцы E, F, G, H с другими возмущениями ещё 2 раза.

Определяем с какого шага задача хорошо обусловлена.

Строим график зависимости N, с которого задача стала хорошо обусловленной от разряда, подвергшегося возмущению.

Делаем сравнительный анализ двух методов.