# Практическая работа №4

**Цель работы.** Целью работы является нахождение корня уравнения методами *Ньютона* и *простых итераций* с заданной точностью *Eps*, исследование скорости сходимости и обусловленности метода.

**Основные теоретические положения.**

**Метод Ньютона.** В случае, когда известно хорошее начальное приближение решения уравнения , эффективным методом повышения точности является метод *Ньютона*. Он состоит в построении итерационной последовательности (1)

(1)

где Последовательность сходится к корню уравнения . По теореме о сходимости метода *Ньютона* должен быть простым корнем уравнения в отсекающем промежутке этого корня функция – дважды непрерывно дифференцируема и

Для оценки погрешности *n*-го приближения корня предлагается пользоваться неравенством (2)

(2)

где - наибольшее значение модуля второй производной на отрезке [a,b]; - наименьшее значение модуля первой производной на отрезке [a,b]. Таким образом, если , то . Это означает, что при хорошем начальном приближении корня после каждой итерации число верных десятичных знаков в очередном приближении удваивается, т.е. процесс сходится очень быстро и имеет место квадратическая сходимость.

Если необходимо найти корень с точностью *ε*, то итерационный процесс можно прекращать, когда выполняется неравенство (3)

(3)

Если на *(n-1)*-м шаге очередное приближение не удовлетворяет условию окончания процесса, то вычисляются величины и следующие приближение корня . При выполнении условия (3) величина принимается за приближенное значение корня *с*, вычисленное с точностью *ε*.

**Постановка задачи**. В практической работе предлагается, использовать программы - функции NEWTON (листинг 4) и Round (листинг 1).

Листинг 4 – Функция NEWTON

double NEWTON(double X, double Eps, int& N) {

// extern double F1(double);

double Y, Y1, DX, Eps0;

N = 0;

double m1 = 3.29; // наименьшее значение модуля 1-ой производной

double M2 = 16.424; // наибольшее значение модуля 2-ой производной

Eps0 = sqrt(2 \* m1 \* Eps / M2);

do {Y = F(X);

if (Y == 0.0) {

return X;}

Y1 = F1(X);

if (Y1 == 0.0) {

puts("Производная обратилась в ноль\n");

exit(1);}

DX = Y / Y1;

X -= DX;

N++;

} while (fabs(DX) >= Eps0);

return X;}Этапы выполнения практической работы

**I часть**:

1. Графически или аналитически отделить корень уравнения (найти отрезки [Left, Right], на которых функция . удовлетворяет условиям теоремы о сходимости метода Ньютона).//см лаба 1-3
2. Проверить функцию на выпуклость вверх или вниз, выбрать начальное приближение корня так чтобы.
3. Получить аналитическое выражение функций и Оценить снизу величину оценить сверху величину
4. По заданному Eps сосчитать условие окончания итерационного процесса Eps2=
5. Составить подпрограммы-функции вычисления , , предусмотрев округление их значений с заданной точностью Delta.
6. Составить головную программу, вычисляющую корень уравнения и содержащую обращение к подпрограммам , , Round, NEWTON и индикацию результатов.
7. Провести вычисления по программе. Исследовать скорость сходимости метода и чувствительность метода к ошибкам в исходных данных.

**II часть:**

(Аналогия с бисекцией): посчитать Xn и Yn с «чистой» функцией,

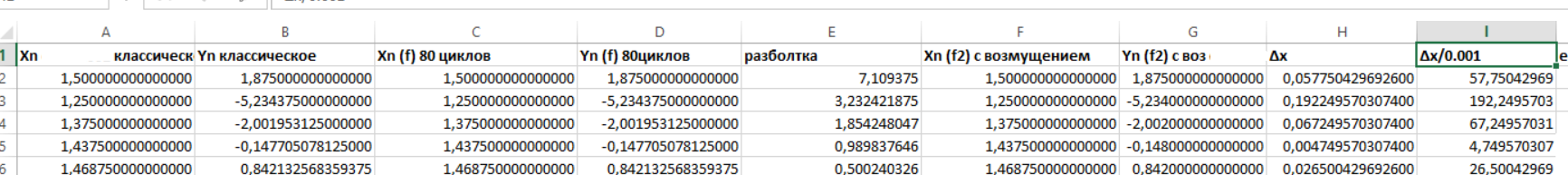
внести возмущения в f(x),

изменить условие цикла до 80 итераций.

**Новое:** подсчитать «разболтку» в столбце Е. Как только вычисления попадают в промежуток неопределенности. Определить номер, с которого начинается «разболтка».

**на входе y, на выходе х**. Поэтому eps должен быть связан с х, delta c y (проверить код программы).

В столбце I считаем и сравниваем с .

****

**Метод простых итераций.** Метод *простых итераций* решения уравнения заключается в замене исходного уравнения эквивалентным ему уравнением и построении последовательности , сходящейся при к точному решению

Корень уравнения является точкой пересечения двух графиков и . Сходимость метода зависит от вида функции . В зависимости от величины модуля первой производной метод может сходиться и расходиться.

Достаточные условия сходимости метода *простых итераций* формулируются следующей теоремой:

Если функция определена, дифференцируема и принадлежит отрезку , то существует число, такое что на , и последовательность,сходится к единственному решению на уравнения при (1)

(1)

Если , то |, если , то |.

Рассмотрим один шаг итерационного процесса. Исходя из найденного на предыдущем шаге значения , вычисляется . Если , то полагается и выполняется очередная итерация. Если же , то вычисления заканчиваются и за приближенное значение корня принимается величина . Погрешность результата вычислений зависит от знака производной : при : погрешность определения корня составляет , а при , погрешность не превышает . Здесь - число, такое, что на отрезке [a,b]. Существование числа является условием сходимости метода в соответствии с отмеченной выше теоремой.

Для применения метода *простых итераций* определяющее значение имеет выбор функции , в уравнении , эквивалентном исходному. Функцию необходимо подбирать так, чтобы . Это обусловливается тем, что если , на отрезке [a,b], то последовательные приближения . будут колебаться около корня *c*, если же , то последовательные приближения будут сходиться к корню *c* монотонно.

Число обусловленности метода *простых итераций* (2)

**Постановка задачи.** В практической работе предлагается использовать программы функции ITER и PHI (листинг 5).

Листинг 5 - функции ITER и PHI

double ITER (double X0, double Eps, int& N, double a, double b) {

if (Eps <= 0.0) {

puts("Неверное задание точности\n");

exit(1); }

double X1 = PHI(X0, a, b);

double X2 = PHI(X1, a, b);

for (N = 2; (X1 - X2) \* (X1 - X2) > fabs((2 \* X1 - X0 - X2) \* Eps); N++) {

X0 = X1;

X1 = X2;

X2 = PHI(X1, a, b); }

return X2; }

double PHI(double x, double a, double b) {

if (x == 0) {

printf("деление на 0!");

exit(1); }

double min = min\_F1(a, b);

double max = max\_F1(a, b);

double s = x - 2 / (min + max) \* (F(x));

s = Round(s, delta);

return(s); }

Необходимо осуществить следующие шаги:

1) Графически или аналитически отделить корень уравнения и получить отрезок [Left, Right].

2) Сосчитать найти на отрезке [Left, Right] – минимальное значение – максимальное значение .

3) Преобразовать уравнение к виду, удобному для итераций .

3) Выбрать начальное приближение , лежащее на [Left, Right].

4) Составить подпрограммы для вычисления значений ,, предусмотрев округление вычисленных значений с точностью Delta.

5) Составить головную программу, вычисляющую корень уравнения и содержащую обращение к программам , PHI, ITER и индикацию результатов.

6) Провести вычисления по программе. Исследовать скорость сходимости и обусловленность метода.

Сделать сравнительную таблицу с результатами последних 4 методов.

**Варианты** соответствуют номеру в списке группы.