МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

(ФГБОУ ВО «ВятГУ»)

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

Лабораторная работа № 1 по дисциплине

«Вычислительная математика»

«ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ»

Вариант 7

Выполнил студент группы ИВТб-2301-04-00 / Жеребцов К. А./

Проверил преподаватель / Исупов К. С./

Киров 2021

**Цель работы**

Изучить численные методы решения нелинейных уравнений: комбинированный метод и метод простых итераций.

**Задание**

1. Построить график функции f(x) и отделить один из корней уравнения: f(x).

2. Сузить интервал изоляции корня, если необходимо, проверив условие: M<=2m.

3. Уточнить корень с погрешностью e<=0,00001 двумя численными методами: комбинированным методом и методом итераций.

4. Проверить полученное значение корня, используя систему Wolfram Alpha.

Уравнение: x+ln(x)-0,5=0

Интервал: [0,1;1,0]

**Теоретическая часть**

1. Локализация корней

Теорема о существовании корней:

Если функция f(x) непрерывна на отрезке [a, b] и принимает на концах отрезка значения разных знаков, то внутри отрезка существует один или нечетное число корней уравнения f(x) = 0.

Теорема о единственности корня:

Если функция f(x) непрерывна и монотонна на отрезке [a, b] и

принимает на концах отрезка значения разных знаков, то внутри

отрезка содержится корень уравнения f(x) = 0 и притом только один.

Теорема о единственности корня:

Если функция f(x) непрерывна и монотонна на отрезке [a, b] и

принимает на концах отрезка значения разных знаков, то внутри

отрезка содержится корень уравнения f(x) = 0 и притом только один.

1. Комбинированный метод

Методы хорд и касательных дают приближения корня с разных сторон. Поэтому их часто применяют в сочетании друг с другом, тогда уточнение корня происходит быстрее.

Пусть дано уравнение f(x) = 0, корень отделен на отрезке [a, b].

Рассмотрим случай, когда f ‘(x) f ’’(x)>0 (рисунок 1).

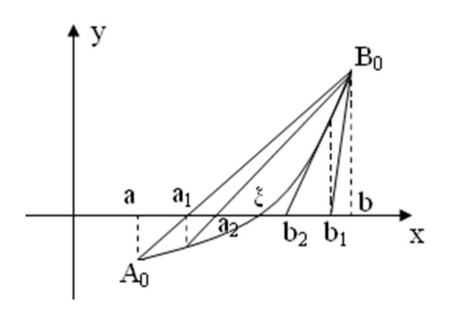
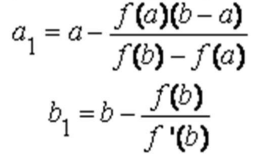
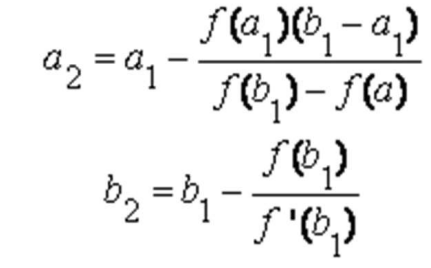


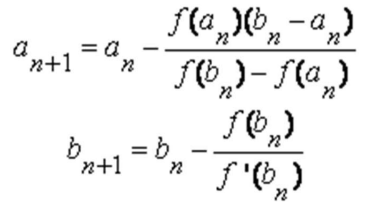
Рисунок 1.

В этом случае метод хорд дает приближенное значение корня с недостатком (конец b неподвижен, неподвижным будет тот конец интервала изоляции, для которого знак функции совпадает со знаком ее второй производной f ``(x), а последовательные приближения лежат по другую сторону корня.), а метод касательных – с избытком (за начальное приближение берем точку b). Тогда вычисления следует проводить по формулам:



Теперь корень ξ заключен в интервале [a1, b1]. Применяя к этому отрезку комбинированный метод, получим: 

и т. д.



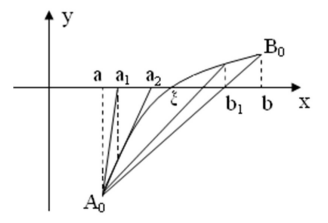
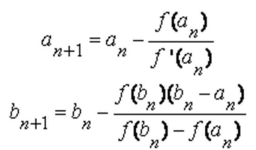
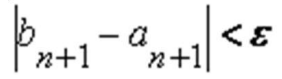
Если же f ‘(x) f ’’(x)<0 (рисунок 2), то, рассуждая аналогично, получим следующие формулы для уточнения корня уравнения: 

Рисунок 2.



Вычислительный процесс прекращается, как только выполнится условие:



1. Метод простых итераций

Представим исходное уравнение в виде x=φ(x). (2.21)

Пусть нам известно начальное приближение к корню х0 (х0 ∈ [a, b]). Подставив его в правую часть уравнения (2.21) получим новое приближение x1 = φ(x0), затем аналогичным образом получим x2 = φ(x1) и так далее, xk+1 = φ(xk), k = 0, 1, 2, ... (2.22)

Для того чтобы итерационный процесс (2.22) был сходящимся, необходимо, чтобы абсолютная величина производной φ`(x) в окрестности корня была меньше единицы. Если это условие выполняется на отрезке [a, b], на котором локализован корень, то в качестве начального приближения можно взять любую точку из этого отрезка. Скорость сходимости зависит от абсолютной величины производной |φ`(x)|: чем меньше |φ`(x)| вблизи корня, тем быстрее сходится процесс.

Переход от уравнения (2.1) к уравнению в итерационной форме (2.21) можно осуществить различными способами в зависимости от вида функции f(x). При таком переходе необходимо построить функцию φ(x) так, чтобы выполнялось условие сходимости.

Рассмотрим один из общих алгоритмов перехода от уравнения

f(x) = 0 к уравнению x = φ(x). Умножим левую и правую части уравнения на произвольную константу k и добавим к обеим частям неизвестное x. При этом корни исходного уравнения не изменятся

x + k \* f(x) = x + 0 \* k или x = x + k \* f(x) (2.24)

Уравнение (2.24) эквивалентно уравнению (2.21) с функцией

φ(x) = x + k \* f(x). Произвольный выбор константы k позволяет обеспечить выполнение условия сходимости.

Поскольку в данном случае φ`(x) = 1 + k \* f `(x), значение k следует выбирать, так чтобы в окрестности корня выполнялось условие | φ`(x)| = |1 + k \* f `(x)| < 1. (2.25)

Наибольшую скорость сходимости в методе простых итераций получим при φ`(x) = 0.

Этого можно добиться, если выбрать параметр k зависящим от x в виде k(x) = - 1 / f `(x) (2.26)

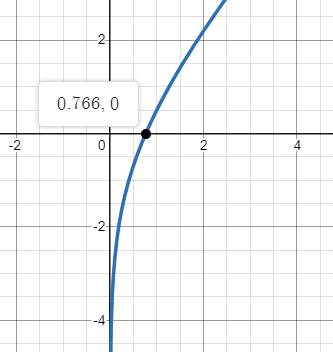
При этом итерационная формула (2.22) переходит в формулу Ньютона



Таким образом, метод Ньютона можно трактовать как частный случай метода простых итераций, обладающий максимальной скоростью сходимости.

**Расчетная часть**

График функции



Исходное уравнение

x + ln(x) – 0.5 = 0

Исходная функция

f(x) = x + ln(x) – 0.5

Первая производная

f ’= 1 + 1 / x

Вторая производная

f ’’ = -1 / x^2

Преобразование исходного уравнения к виду φ(x) = x

φ(x) = x – f(x) / k

φ(x) = x - x + (ln(x) – 0.5) / 6

Критерий останова для комбинированного метода:

|xk+1 – x̃k+1| < ε, так как M = 1, m = 2; M <= 2m, где



Критерий останова для метода итераций:

k\*|Xn+1 - Xn| <= E, так как сходимость монотонная

**Листинг**

Комбинированной метод:

procedure komb(a, b, e: real);

var

x0: real;

x11, x12: real;

m, mm: real;

begin

writeln('Исходное уравнение x+ln(x)-0,5=0; Интервал [',a,';',b,']');

writeln('f`(x)=1+1/x; f`(',a,')=', f1(a), '; f`(',b,')=', f1(b), ';');

writeln('f``(x)=-1/x^2; f``(',a,')=', f2(a), '; f``(',b,')=', f2(b), ';');

m := min(abs(f1(a)), abs(f1(b)));

mm := max(abs(f2(a)), abs(f2(b)));

ostanov(m, mm);

if f(a) \* f2(a) > 0 then writeln('Значение по недостатку вычисляется методом хорд, а по избытку методом касательных')

else if f(b) \* f2(a) > 0 then writeln('Значение по недостатку вычисляется методом касательных, а по избытку методом хорд');

writeln('------------------------------------------------------------');

writeln('n | Bn | f(Bn) | f(An)-f(Bn) | f(An)\*(An-Bn) | h | Bn+1 ');

if f(a) \* f2(a ) > 0 then begin

x12 := b - f(b) / f1(b);

x11 := a - ((a - b) \* f(a) / (f(a) - f(b)));

writeln(p, ' | ', b:6:5, ' | ', f(b):6:5, ' | ', f(a) - f(b):6:5, ' | ', f(a) \* (a - b):6:5, ' | ', (f(a) \* (a - b)) / (f(a) - f(b)):6:5, ' | ', x12:6:5, ' | ', abs(x12 - x11):6:9);

inc(p);

while abs(x12 - x11) >= e do

begin

a := x11;

b := x12;

x12 := b - f(b) / f1(b);

x11 := a - ((a - b) \* f(a) / (f(a) - f(b)));

writeln(p, ' | ', b:6:5, ' | ', f(b):6:5, ' | ', f(a) - f(b):6:5, ' | ', f(a) \* (a - b):6:5, ' | ', (f(a) \* (a - b)) / (f(a) - f(b)):6:5, ' | ', x12:6:5, ' | ', abs(x12 - x11):6:9);

inc(p);

end;

writeln('Корень: ', x12:6:5)

end

else

if f(b) \* f2(a) > 0 then begin

x11 := a - f(a) / f1(a);

x12 := b - ((b - a) \* f(b) / (f(b) - f(a)));

writeln(p, ' | ', a:6:5, ' | ', f(a):6:5, ' | ', f(b) - f(a):6:5, ' | ', f(b) \* (b - a):6:5, ' | ', (f(b) \* (b- a)) / (f(b) - f(a)):6:5, ' | ', x12:6:5, ' | ', abs(x12 - x11):6:9);

inc(p);

while abs(x12 - x11) >= e do

begin

a := x11;

b := x12;

x12 := b - f(b) / f1(b);

x11 := a - ((a - b) \* f(a) / (f(a) - f(b)));

writeln(p, ' | ', a:6:5, ' | ', f(a):6:5, ' | ', f(b) - f(a):6:5, ' | ', f(b) \* (b - a):6:5, ' | ', (f(b) \* (b- a)) / (f(b) - f(a)):6:5, ' | ', x12:6:5, ' | ', abs(x12 - x11):6:9);

inc(p);

end;

writeln('Корень: ', x12:6:5)

end;

end;

Метод простых итераций

procedure iter(a, b, e: real);

var

c1, c2: real;

x1, x2: real;

q, qq, n, k: real;

i: integer;

begin

writeln('Исходное уравнение x+ln(x)-0,5=0; Интервал [',a,';',b,']');

c1 := abs(f1(a));

c2 := abs(f1(b));

if c1 < c2 then n := c2 else n := c1;

i := sign(f1(a));

if i = -1 then k := -(n / 2 + 0.5) else k := n / 2 + 0.5;

writeln('Канонический вид phi(x) = x - (x + ln(x)-0,5)/', k, ';');

writeln('phi`(x) = 5/6 - 1/6x; phi`(',a,') = ', phi1(a):3:2, '; phi`(',b,') = ',phi1(b):3:2);

writeln('|phi`(x)| < 1 и sign(phi`(',a,'))<>sign(phi`(',b,')) -> монотонная сходимость');

q := max(abs(phi1(a)), abs(phi1(b)));

qq := 1 / (1 - q);

writeln('Критерий останова: ', qq:3:2, '\*|Xn+1 - Xn| <= E');

writeln('------------------------------------------------------------');

writeln('n | Xn | phi(Xn) | ', qq:3:2, ' \* |Xn+1 - Xn| ');

x1 := a;

x2 := phi(x1, k);

writeln(p, ' | ', X1:6:5, ' | ', x2:6:5, ' | ', qq \* abs(x2 - x1):6:5);

p := p + 1;

while qq \* abs(x2 - x1) > e do

begin

x1 := x2;

x2 := phi(x1, k);

writeln(p, ' | ', X1:6:5, ' | ', x2:6:5, ' | ', qq \* abs(x2 - x1):6:5);

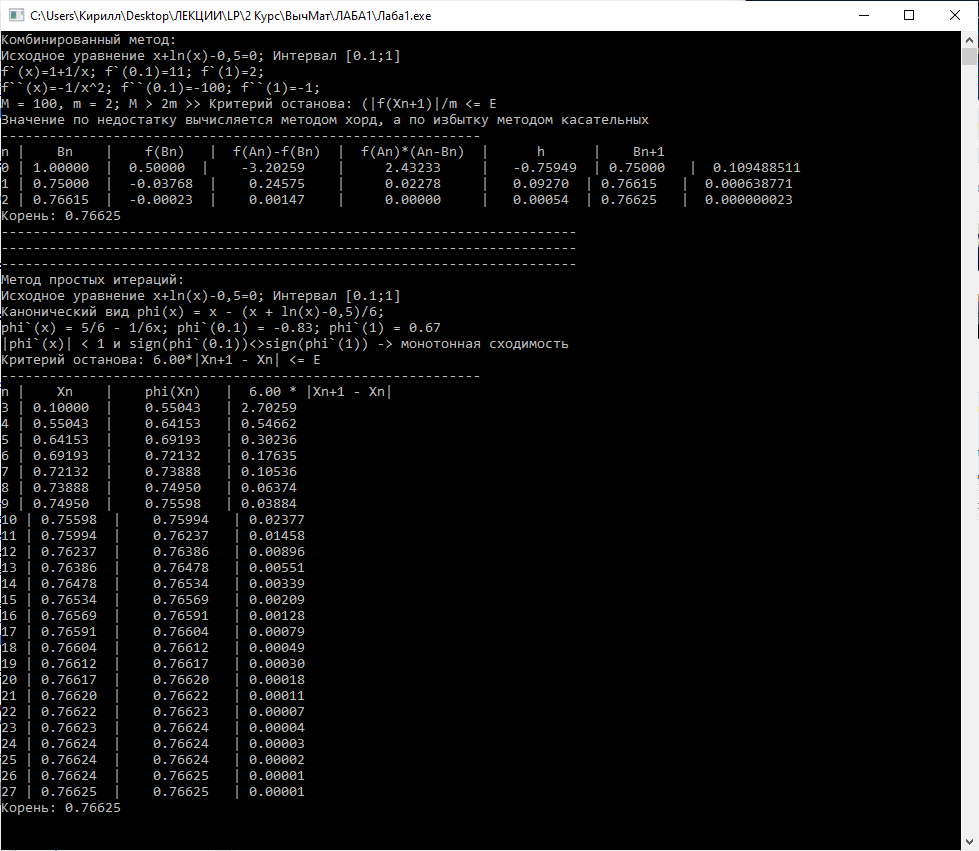
p := p + 1;;

end;

writeln('Корень: ', x2:6:5)

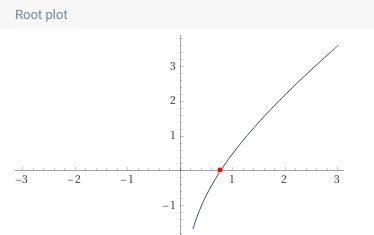
end;

**Результат работы программы:**

****

**Проверка в системе Wolfram Alpha**







**Вывод**

Были изучены численные методы решения нелинейных уравнений: комбинированный метод и метод простых итераций.