МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Методы решения систем уравнений

Отчет по лабораторной работе №2 дисциплины

«Вычислительная математика»

Выполнил студент группы ИВТб-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Седов М.Д./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Исупов К.С./

Киров 2018

1 Постановка задачи

Решить систему линейных уравнений 4-го порядка методом Гаусса с точностью е=0,001.

Уравнения системы:

0,73\*x1+1,24\*x2-0,38\*x3-1,43\*x4=0,58

1,07\*x1-0,77\*x2+1,25\*x3+0,66\*x4=-0,66

1,56\*x1+0,66\*x2+1,44\*x3-0,87\*x4=1,24

0,75\*x1+1,22\*x2-0,83\*x3+0,87\*x4=0,92

Решить систему линейных уравнений 4-го порядка с точностью е=0,0001: - методом Зейделя.

Уравнения системы:

x1=0,17\*x1+0,31\*x2-0,18\*x3+0,22\*x4-1,71

x2=-0,21\*x1+0,33\*x3+0,22\*x4+0,62

x3=0,32\*x1-0,18\*x2+0,05\*x3-0,19\*x4-0,89

x4=0,12\*x1+0,28\*x2-0,14\*x3+0,94

Решить систему линейных уравнений 3-го порядка методом обратной матрицы с точностью e=0,001.

Уравнение системы:

3\*x1+2\*x2+7\*x3=0

-5\*x1+4\*x2+x3=2

x1-3\*x2+2\*x3=-7

Решить систему нелинейных уравнений 2-го порядка методом Ньютона с точностью е=0,001.

Уравнения системы:

sin(x)-y-1,32=0

cos(y)-x+0,85=0

Проверить результаты с помощью системы Mathcad.

2 Краткие теоретические сведения

2.1 Метод Гаусса для систем линейных уравнений

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 отделён на отрезке *[a, b].*

Рассмотрим случай, когда f´(x) \* f´´(x) > 0 (рис. 1).

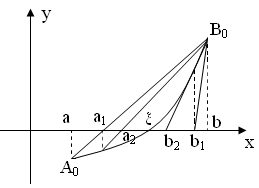


Рис. 1

В этому случае метод хорд дает приближенное значение корня с недостатком (конец b неподвижен), 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 (рис. 2), то, рассуждая аналогично, получим следующие формулы для уточнения корня уравнения:

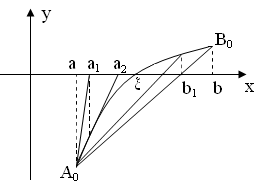


Рис. 2

Вычислительный процесс прекращается, как только выполняется условие:

|bn+1-an+1| < ε

2.2 Метод итераций

Этот метод требует приведения исходного уравнения к каноническому виду: f(x) = 0 -> x = φ(x). Тогда одношаговый итерационный процесс строится по формуле: xn+1 = φ(xn) при выборе любого нулевого приближения из интервала изоляции.

Главное – проверка соблюдения условий сходимости для канонического уравнения: |φ´(x)| < 1 – это обеспечит сходящийся итерационный процесс и получение значения корня уравнения с требуемой точностью за конечное число шагов.

Существует стандартный прием преобразования уравнения к каноническому виду, обеспечивающий сходимость итерационного процесса.

Знак k совпадает со знаком первой производной исходной функции. Итерации продолжают до тех пор, пока не будет выполнено условие:

Если функция φ(x) возрастает, приближенные значения сходятся к точному значению корня монотонно, если же функция φ(x) убывает, то приближенные значения колеблются вокруг точно значения.

3 Выполнение задания

Исходная функция: x \* x - ln(x + 1)

Первая производная: 2 \* x - 1 / (x + 1)

Вторая производная: 2 + 1 / ((x + 1) \* (x + 1))

Значения при уточнении корная комбинированным методом представлены на Рисунке 1.

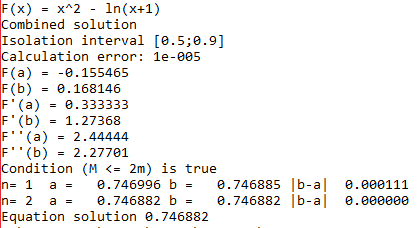


Рисунок 1 – уточнение корня комбинированным методом

Последовательные приближенные значения корня, полученные методом итераций (начальное приближение равно 0), представлены на Рисунке 2.

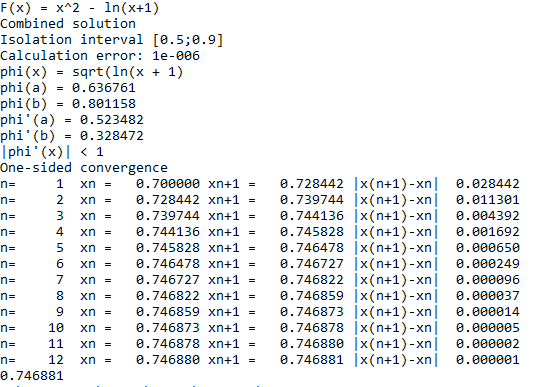


Рисунок 2 – значения корня, полученные методом итераций

4 Экранные формы работы программы

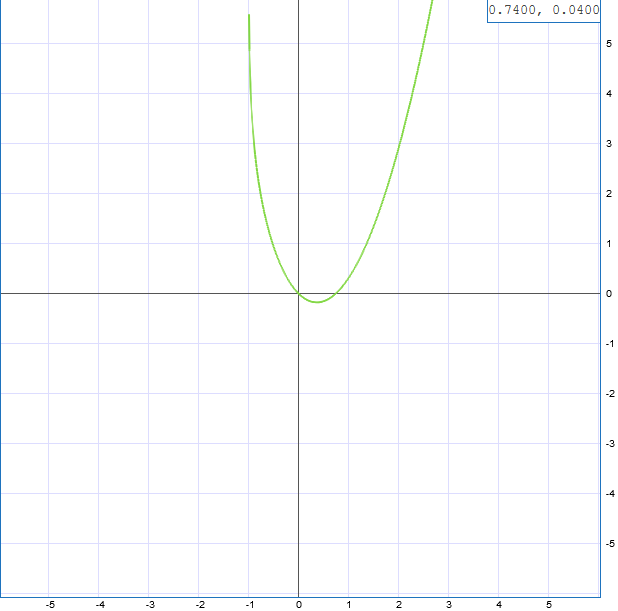


Рисунок 3 – график функции

5 Проверка вычислений корня

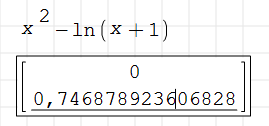


Рисунок 4 – результат проверки корня в система SMath Studio

Приложение А

Листинг программы #1

#include <iostream>

#include <cmath>

#include <iomanip>

#include <algorithm>

using namespace std;

double f(double x) { return x \* x - log(x + 1); }

double f\_1(double x) { return 2 \* x - 1 / (x + 1); }

double f\_2(double x) { return 2 + 1 / ((x + 1) \* (x + 1)); }

int main() {

double a = 0.5, b = 0.9, e = 0.00001;

cout << "F(x) = x^2 - ln(x+1)" << endl;

cout << "Combined solution" << endl;

cout << "Isolation interval [" << a << ";" << b << "]" << endl;

cout << "Calculation error: " << e << endl;

cout << "F(a) = " << f(a) << endl;

cout << "F(b) = " << f(b) << endl;

cout << "F'(a) = " << f\_1(a) << endl;

cout << "F'(b) = " << f\_1(b) << endl;

cout << "F''(a) = " << f\_2(a) << endl;

cout << "F''(b) = " << f\_2(b) << endl;

int n = 1;

if (max(abs(f\_2(a)), abs(f\_2(b))) <= 2 \* max(abs(f\_1(a)), abs(f\_1(b)))) cout << "Condition (M <= 2m) is true" << endl;

if (f(a) \* f\_2(a) > 0) {

a = a - f(a) / f\_1(a);

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

} else {

b = b - f(b) / f\_1(b);

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

}

while (abs(a - b ) > 2 \* e) {

a = a - f(a) / f\_1(a);

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

cout << "n= " << fixed << setprecision(6) << n << " a = " << setw(10) << a << " b = " << setw(10) << b << " |b-a|" << setw(10) << abs(b - a) << endl;

n++;

}

double x = (a + b) / 2;

cout << "Equation solution " << x;

}

Листинг программы #2

#include <iostream>

#include <cmath>

#include <iomanip>

#include <algorithm>

using namespace std;

double phi(double x) { return sqrt(log(x + 1)); }

double phi\_1(double x) { return 1 / (2 \* sqrt(log(x + 1)) \* (x + 1));}

double find(double x, double e) {

double rez;

int iteration = 0;

do {

iteration++;

rez = x;

x = sqrt(log(x + 1));

cout << "n= " << fixed << setprecision(6) << setw(5) << iteration << " xn = " << setw(10) << rez << " xn+1 = " << setw(10) << x << " |x(n+1)-xn|" << setw(10) << abs(rez - x) << endl;

} while (phi\_1(0.5) / (1 - phi\_1(0.5)) \* fabs(rez - x) > e);

return x;

}

int main(int argc, char const \*argv[])

{

double a = 0.5, b = 0.9, e = 0.000001;

cout << "F(x) = x^2 - ln(x+1)" << endl;

cout << "Combined solution" << endl;

cout << "Isolation interval [" << a << ";" << b << "]" << endl;

cout << "Calculation error: " << e << endl;

cout << "phi(x) = sqrt(ln(x + 1)" << endl;

cout << "phi(a) = " << phi(a) << endl;

cout << "phi(b) = " << phi(b) << endl;

cout << "phi'(a) = " << phi\_1(a) << endl;

cout << "phi'(b) = " << phi\_1(b) << endl;

if (abs(phi\_1(a)) < 1 && abs(phi\_1(b) < 1)) cout << "|phi'(x)| < 1" << endl;

cout << "One-sided convergence" << endl;

double x0 = (a + b) / 2;

cout << find(x0, e);

return 0;

}