Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра Автоматизированных Систем Управления (АСУ)

Отчет по контрольной (лабораторной) работе № 1

по дисциплине «Вычислительная математика»

Выполнил: ст. гр. з-422П8-5

Жданов А.А. «\_\_» \_\_\_2023г.

Проверил: к.т.н., доц. каф. АСУ \_\_\_\_\_\_\_\_

Романенко В. В. «\_\_» \_\_\_2023г.

Томск 2023

Оглавление

1 Решение уравнений с одной переменной

1.1 Задание

1.2 Теоретический материал

1.3 Алгоритм решения

1.4 Результаты работы программы

2 Решение задач линейной алгебры

2.1 Задание

2.2 Теоретический материал

2.3 Алгоритм решения

2.4 Результаты работы программы

Выводы

Список использованных источников

Приложение А Листинги программ

1 Решение уравнений с одной переменной

1.1 Задание

Написать программу отделения корней их уточнения методом дихотомии.

Входные данные:

функция f(x);

интервал [a,b];

точность по аргументу и по функции.

Выходные данные: корни ξ , точность;

значения функции f (ξ);

количество итераций n ;

количество вычислений функции f (x) ;

время счета.

Вариант 6: f(x) = - 3cos(x)

1.2 Теоретический материал

Для решения уравнения был избран метод дихотомии или половинного деления. У данного метода теоретическая идея довольно проста — делить заданный отрезок пополам (отсюда, собственно и название), подставлять значение срединной точки в уравнение до тех пор, пока разница между правой и левой частями уравнения не будет равной, или менее заданной точности, либо длина отрезка также не станет менее заданной (тогда, решение находится на одном из концов разделенного отрезка). Для задания лабораторной работы в программе устанавливается точность равная 0,0001 как по функции, так и по аргументу.

1.3 Алгоритм решения

В программе реализованы 3 метода: метод непосредственно половинного деления; метод возвращающий значение заданной функции в какой-либо точке; метод определяющий знак функции в какой-либо точке.

Функция осуществляющая метод половинного деления получает в качестве параметров значения начала и конца заданного отрезка (в программе этот интервал определен как [0,2]), значение длины отрезка, и значение функции в точке. Далее запускается цикл деления отрезка пополам до тех пор пока абсолютное значение функции в точке не будет равным или менее установленного параметра, если это значение превышает установленный параметра запускается условный оператор, который переназначает начало или конец отрезка в зависимости от знака функции в точке — если знак функции в начале отрезка совпадает со знаком функции в текущей точке, то началу отрезка присваиватся значение середины отрезка, в том случае если знак не совпадает значение присваивается концу отрезка, и процесс деления пополам начинается вновь.

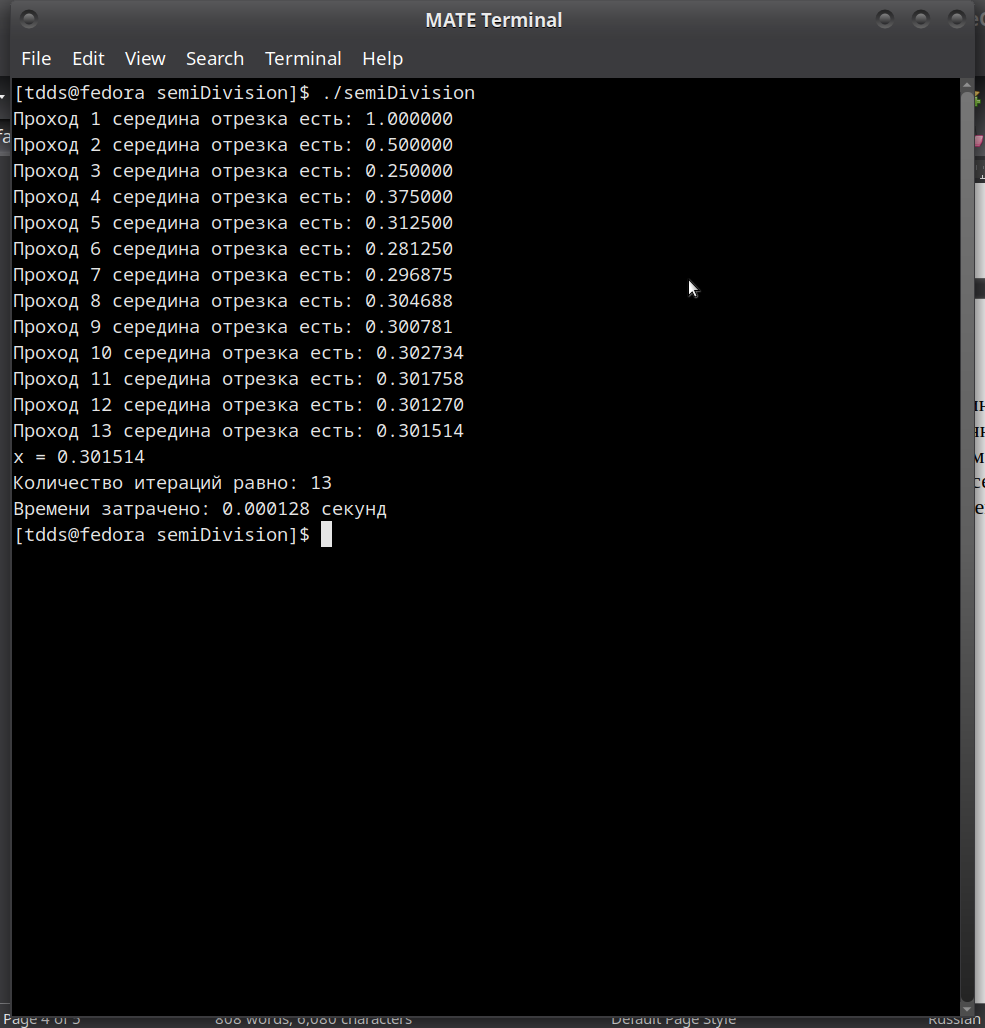
Метод возвращающий значение функции в точке получает параметр, являющийся нулем на первом шаге, а далее этим параметром становится середина отрезка. Метод подставляет полученный параметр в функцию (согласно варианту), вычисляет её значение, и возвращает его.

Метод определяющий знак функции в точке принимает как параметр значение текущей точки и возвращает 1, либо -1, соответственно знаку текущей точки.

В главной функции находятся: счетчик времени выполнения программы; выражения выводящие на экран все подставлямые середины отрезков, количество итераций, и количество затраченного на расчет времени.

1.4 Результаты работы программы

Результат работы программы показан на рис. 1

Рисунок 1.

2. Решение задач линейной алгебры

2.1 Задание

Написать программу решения системы линейных алгебраических уравнений методом Гаусса.

Входные данные: порядок системы n; матрица системы A; правая часть системы b.

Выходные данные: промежуточные векторы и матрицы; решение системы; невязка.

Вариант 6:

3.2x+5.4y+4.2z+2.2p=11.4

2.1x+3.2y+3.1z+1.1p=9.2

1.2x+0.4y-0.8z-0.8p=0.4

4.7x+10.4y+9.7z+9.7p=30.4

Написать программу вычисления определителя матрицы методом Гаусса. Входные данные: порядок системы n; матрица системы A.

Выходные данные: значение определителя.

Написать программу вычисления обратной матрицы методом Гаусса.

Входные данные: порядок системы n ; матрица системы A.

Выходные данные: промежуточные векторы и матрицы; обратная матрица; невязка.

2.2 Теоретический материал

Метод Гаусса отличается относительно малым количеством итераций при решении СЛАУ. Кроме того, при помощи него хорошо описаны в литературе методы обращения матриц, и вычисления определителей.

Метод заключается в следующем — для начала, при помощи элементарных преобразований, система приводится к т.н. треугольному (сверху или снизу) виду, т.е. уравнения системы содержат столько неизвестных с ненулевыми коэффициентами, каков их номер по счету в системе (для треугольных снизу систем, для систем треугольных сверху — наоборот). Далее, в действие приводится метод обратной подстановки, т.е. неизвестные вычисляются в обратном порядке, и затем последовательно подставляются в уравнения все еще содержащие неизвестные, тем самым, постепенно находя все неизвестные.

Вычисление определителя методом Гаусса вообще не представляет какой-либо сложности — его значение равно произведению коэффициентов, стоящих на главной диагонали, приведенной к треугольному виду матрицы.

Для вычисления обратной матрицы также используется метод Гаусса, с тем отличием, что вместо свободных членов заданной СЛАУ, подставляются значения единичной матрицы «постолбцово». Т.е. получается четыре системы уравнений имеющих одну и ту же матрицу и различные свободные члены. Решая эти четыре системы уравнений получится 16 значений неизвестных, которые в свою очередь являются значениями обратной матрицы. Невязка вычисляется подстановкой членов столбца в первоначальную систему, и сравнивается со свободными членами единичной («постолбцово») матрицы. Разность между ними и будет невязкой обратной матрицы.

2.3 Алгоритм работы программы

Программы состоит из «пустых» методов, которые ничего не возвращают, но преобразуют переменные. Переменными здесь являются матрицы, а также векторы.

Вначале система проверяет нет ли у заданной матрицы нулей на осевых элементах (элементах, стоящих на главной диагонали). Метод проверки основан на цикличном проходе по всем строкам, а затем по всем элементам строки. Если находится нуль, то к строке применяются элементарные преобразования — каждый член следующей строки умножается на -1 и прибавляется к текущей. Здесь можно отметить, что подобные преобразования несут некий риск, т.к. возможна ситуация, когда в следующей строке соответствующий член тоже равен нулю, преобразование не получится. Но подразумевается, что в учебных заданиях не может быть сингулярных, или систем без решения.

Далее вызывается метод преобразующий, проверенную, и при необходимости, преобразованную матрицу системы, к треугольному виду.

Приложение А

#include <stdio.h>

#include <math.h>

#include <time.h>

float dihot(float a, float b, float e, float e1);//Инициализация метода дихотомии

float func( float x);// Инициализация метода заданной функции

int sign(float x);// Инициализация метода определяющего знак

int n = 0;

int main()

{

clock\_t start, end;

start = clock();

printf("x = %f\n", dihot(0, 2, 0.0001, 0.0001));// Вывод на экран решения уравнения. Параметром вывода является результат метода реализующего, собственно, метод дихотомии. В свою очередь, параметрами функции являются начало и конец отрезка, на котором находится решение, эти цифры можно менять; точность с которой будет решаться уравнение, значение, с которым будет сравниваться значение заданной функции на предмет приближения к нулю (по известной теореме, это и будет решением уравнения) соответственно.

end = clock();

double time\_taken = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("Количество итераций равно: %d\n", n);

printf("Времени затрачено: %f секунд\n",time\_taken);

}

float func( float x)//Метод возвращает значение заданной функции, вычисленной от текущего параметра

{

return sqrt(4\*x+7)-3\*cos(x); //Вариант 6

}

int sign(float x)//Метод возвращает 1 либо -1 в зависимости от поступившего параметра. Параметром является текущее вычисленное х.

{

int res;

res=0;

if (x<0) res=-1;

if (x>0) res= 1;

return res;

}

float dihot(float a, float b, float e, float e1)//Метод посредством цикла while делит заданный отрезок пополам до тех пор пока разница между концами отрезка не будет меньше или равна параметру е, либо абсолютное значение функции в текущей точке приблизится к нулю менее чем параметр е1

{

float x;

while (b-a>e)

{

n++;

x=(a+b)/2;// Вычисление значения искомой переменной, собственно, половинное деление/метод дихотомии

printf("Проход %d середина отрезка есть: %f \n", n, x);

if (fabs(func(x))<e1)

break; // Если асолютное значение функции в соответсвующей точке приближается к нулю с заданной точностью, цикл прекращается, и метод возвращает текущее значение х, которое и является решением уравнения с заданной точностью

if (sign(func(a))==sign(func(x))) {

a=x; // Значению начальной точки присваивается значение текущей искомой переменной, если значение функции в начально точке равно значению функции в точке х

}else

b=x; // Если предыдущее условие не выполнено значение переменной х присваивается концу заданного отрезка, на котором ищется решение

}

return x; //Метод возвращает значение переменной удовлетворяющее первому условию в цикле выше

}