**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕНЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.Шухова»**

**(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Дисциплина: Вычислительная математика

Лабораторная работа № 5

Тема: «Решение системы двух нелинейных уравнений с двумя неизвестными методом Ньютона»

Выполнила:

Студент группы ВТ-21

Сидорова Ангелина Сергеевна

Проверила: Бондаренко Т. В.

Белгород 2018

**Цель работы:** изучение и получение практических навыков приближенного решения систем двух нелинейных уравнений с двумя неизвестными методом Ньютона.

Задания к работе

1. Записать для уравнений системы соответствующего варианта задания функции F(x, y) и Φ(x, y). Построить в одной системе координат графики функций F(x, y) и Φ(x, y). Найти точки пересечения графиков функций F(x, y) и Φ(x, y), которые соответствуют решениям системы уравнений.

2. Определить область содержащую одну из точек пересечения графиков функций F(x, y) и Φ(x, y). Выбрать начальное приближение решения системы уравнений ― точку М0 с координатами (x0; y0), принадлежащую выбранной области.

3. Найти частные производные первого порядка по переменным x, y для функций F(x, y) и Φ(x, y) и вычислить значения производных в точке М0. Записать линеаризованную систему, соответствующую исходной нелинейной системе, для выбранного начального приближения М0(x0; y0).

4. Выполнить один шаг численного метода решения системы двух нелинейных уравнений с двумя неизвестными «вручную» и найти следующее приближение к решению системы уравнений (x1; y1). Выполнить проверку правила остановки с точностью ε=0,001.

5. Описать в модуле логическую функцию для приближенного решения системы двух нелинейных уравнений с двумя неизвестными методом Ньютона с заданной точностью ε. Входными данными являются: - функции уравнений системы и их частные производные (f (x, y), ∂f / ∂x, ∂f / ∂y),(g(x, y), ∂g / ∂x, g / ∂y) ; - начальное приближение (x0; y0); - точность решения ε; - максимальное число итераций n. Функция возвращает значение «истина», если приближенное решение системы с заданной точностью получено за число итераций, не превышающее n, при этом вычисленная пара (x; y) – приближенное решение системы двух уравнений с точность решения ε. В противном случае функция возвращает значение «ложь».

6. Составить программу для решения системы двух нелинейных уравнений соответствующего варианта задания.

Вариант 10

Безымянный.png

Задание 1

Запишем выражение для функций, соответствующих уравнениям системы:

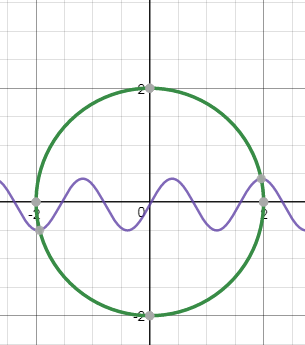
F(x) = sin(4x)-2,2y-0,1

Ф(х) = x2+y2-4

Построим в одной системе координат графики указанных функций:

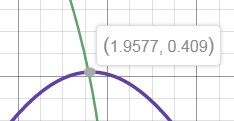
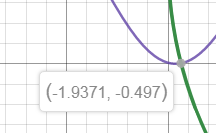
sin(4x)-2,2y-0,1=0

x2+y2-4=0



По внешнему виду графиков функций можно сделать вывод, что система уравнений имеет решение, так как графики функций F(x, y) и Φ(x, y) пересекаются.

Точки пересечения:



Задание 2

М0(2;0,5)

Задание 3

F(x0;y0) = sin(4\*2)-2,2\*0,5-0,1 = -1.06

Ф(x0;y0) = x20+y20-4 = 22+0,52-4 = 0.25

Введем обозначение:

Составим систему линейных уравнений с неизвестными h и l:

Задание 4

Вычисляем следующее приближение к решению системы:

Проверим условие остановки

F(x1;y1) = -0.949

Ф(x1;y1) = 1,983

| F(x1;y1)|+| Ф(x1;y1) | = | -0.949 + 1.983 | > ε = 0.001

Задание 5-6

#include <stdio.h>

#include <stdlib.h>

int meth\_newton(float (\*f)(float x, float y, int f), float (\*g)(float x, float y, int f), float \*x, float \*y, float e, int n)

{

float d,d1,d2;

float h = 0, l = 0;

int k = 0;

do

{

k++;

d = f(\*x,\*y,1)\*g(\*x,\*y,2) - f(\*x,\*y,2)\*g(\*x,\*y,1);

d2 = -f(\*x,\*y,1)\*g(\*x,\*y,0) + g(\*x,\*y,1)\*f(\*x,\*y,0);

d1 = -f(\*x,\*y,0)\*g(\*x,\*y,2) + g(\*x,\*y,0)\*f(\*x,\*y,2);

h = d1/d;

l = d2/d;

\*x = \*x + h;

\*y = \*y + l;

}

while(k < n && fabs(f(\*x,\*y,0)) + fabs(g(\*x,\*y,0)) > e);

return (fabs(f(\*x,\*y,0)) + fabs(g(\*x,\*y,0)) < e);

}

float f(float x, float y, int f)

{

if (f == 0)

return (sin(4\*x)-2.2\*y-0.1);

else

if (f == 1)

return (4\*cos(4\*x));

else

return (-2.2);

}

float g(float x, float y, int f)

{

if (f == 0)

return (x\*x + y\*y - 4);

else

if (f == 1)

return (2\*x);

else

return (2\*y);

}

int main()

{

float x, y, e = 0.001;

int n = 5;

printf("X = ");

scanf("%f",&x);

printf("Y = ");

scanf("%f",&y);

int t = meth\_newton(f, g, &x, &y, e, n);

printf("Reshenie\nX = %f\n",x);

printf("\nY = %f\n",y);

printf("%i",t);

}

