**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



**СИСТЕМИ НЕЛІНІЙНИХ РІВНЯНЬ. МЕТОД НЬЮТОНА ТА ε-АЛГОРИТМ**

**Лабораторна робота №3**

**Виконав:**

**ст. гр. ІР-21**

**Левицький М. І.**

**Прийняла:**

**Доцент каф. КСА**

**Дзелендзяк У.Ю.**

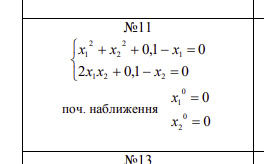
**Львів 2020**

**Мета роботи:** ознайомитися з найпоширенішим ітераційним методом розв’язування систем нелінійних рівнянь – методом Ньютона та екстраполяційним методом – ε -алгоритмом.

**ЗАВДАННЯ**

|  |  |  |
| --- | --- | --- |
| № п/п |  | Вхідні дані |
| 11 | Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по по всій матриці ) | e = 0.00001 |

Рівняння для завдання:



**Код програми :**

f\_func = [  
 lambda x\_1**,** x\_2: x\_1 \*\* 2 + x\_2 \*\* 2 + 0.1 - x\_1**,** lambda x\_1**,** x\_2: 2 \* x\_1 \* x\_2 + 0.1 - x\_2  
]  
  
f\_x\_func = [  
 [  
 lambda x\_1**,** x\_2: 2 \* x\_1 - 1**,** lambda x\_1**,** x\_2: 2 \* x\_2  
 ]**,** [  
 lambda x\_1**,** x\_2: 2 \* x\_2**,** lambda x\_1**,** x\_2: 2 \* x\_1 - 1  
 ]  
]  
  
n = 2  
E = [  
 [0.0**,** 0.0]**,** [0.0**,** 0.0]  
]  
V = [  
 [0.0**,** 0.0]**,** [0.0**,** 0.0]  
]  
C = [  
 [0.0**,** 0.0]**,** [0.0**,** 0.0]  
]  
  
Y = [0.0**,** 0.0]  
X = [0.0**,** 0.0]  
  
INVERS = [  
 [0.0**,** 0.0]**,** [0.0**,** 0.0]  
]  
  
e = 0.00001  
  
f\_values = [0**,** 0]  
J = [[0**,** 0]**,** [0**,** 0]]  
  
x = [0.0**,** 0.0]  
x\_old = [0.0**,** 0.0]  
  
  
def is\_not\_ok():  
 is\_Ok = True  
 for i in range(n):  
 is\_Ok = is\_Ok and abs((x[i] - x\_old[i]) / x[i]) < e  
 x\_old[i] = x[i]  
  
 for i in range(n):  
 pass  
 return is\_Ok  
  
  
def main():  
 counter = 0  
 for i in range(n):  
 for j in range(n):  
 C[i][j] = 0  
 INVERS[i][j] = 0.0  
 if i == j:  
 E[i][j] = 1  
  
 while True:  
 counter += 1  
 for i in range(n):  
 f\_values[i] = f\_func[i](x[0]**,** x[1])  
 for j in range(n):  
 J[i][j] = f\_x\_func[i][j](x[0]**,** x[1])  
  
 for b in range(n):  
 # метод Гауса  
 inx = []  
 P = []  
  
 for i in range(n):  
 inx.append(i)  
  
 for i in range(n):  
 for j in range(n):  
 V[i][j] = (J[i][j])  
 P.append(E[i][b])  
  
 for k in range(n):  
 # сортування  
 max\_sth = abs(V[k][k])  
 w = k  
 h = k  
 for l in range(k**,** n):  
 for f in range(k**,** n):  
 if max\_sth < abs(V[l][f]):  
 max\_sth = abs(V[l][f])  
 h = l  
 w = f  
  
 value = P[k]  
 P[k] = P[h]  
 P[h] = value  
  
 for d in range(n):  
 value = V[k][d]  
 V[k][d] = V[h][d]  
 V[h][d] = value  
  
 z = inx[k]  
 inx[k] = inx[w]  
 inx[w] = z  
  
 for d in range(n):  
 if d < k:  
 value = C[d][k]  
 C[d][k] = C[d][w]  
 C[d][w] = value  
 else:  
 value = V[d][k]  
 V[d][k] = V[d][w]  
 V[d][w] = value  
 # кінець сортування  
  
 Y[k] = P[k] / V[k][k]  
 for i in range(k + 1**,** n):  
 P[i] -= V[i][k] \* Y[k]  
 for j in range(k + 1**,** n):  
 C[k][j] = V[k][j] / V[k][k]  
 V[i][j] -= V[i][k] \* C[k][j]  
  
 for i in range(n - 1**,** -1**,** -1):  
 X[i] = Y[i]  
 for j in range(i + 1**,** n):  
 X[i] -= C[i][j] \* X[j]  
  
 for i in range(n):  
 if inx[i] != i:  
 z = inx[i]  
 value = X[i]  
 X[i] = X[z]  
 X[z] = value  
 inx[i] = inx[z]  
 inx[z] = z  
 # кінець методу Гауса  
  
 for i in range(n):  
 INVERS[i][b] = X[i]  
  
 for i in range(n):  
 for j in range(n):  
 x[i] = x[i] - INVERS[i][j] \* f\_values[j]  
  
 if is\_not\_ok():  
 break  
 return counter  
  
  
print("count: "**,** main())  
print(x)

# check  
for i in range(n):  
 print(f\_func[i](x[0]**,**x[1]))

**Результат програми:**

