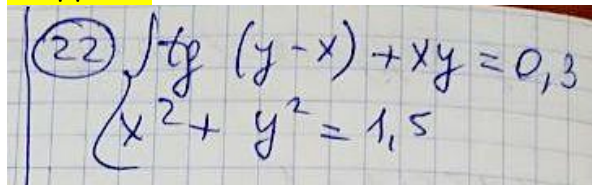


Метод Простих Ітерацій

Підготувала:
Студентка ПМІ-23
Шувар Софія

Мета: реалізувати метод простих ітерацій для розв'язання систем нелінійних рівнянь. Продемонструвати роботу програми на конкретному прикладі.

Завдання:

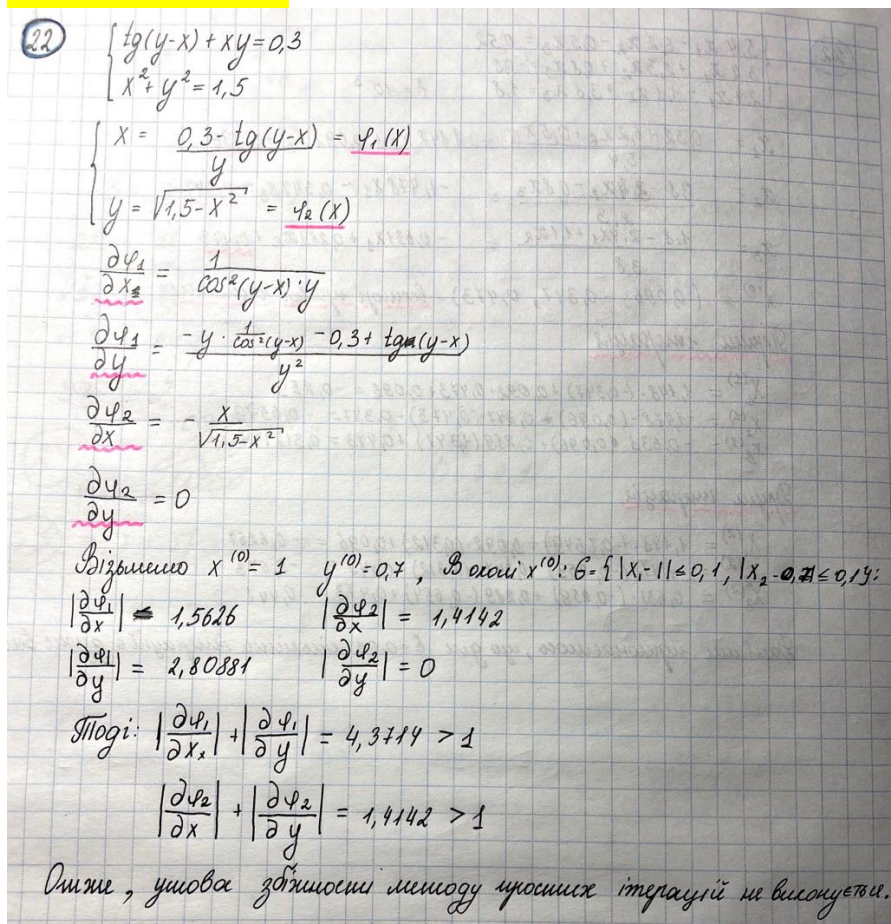


$$(22) \begin{cases} \operatorname{tg}(y-x) + xy = 0,3 \\ x^2 + y^2 = 1,5 \end{cases}$$

Хід роботи:

1. Розв'язати систему нелінійних рівнянь за допомогою методу простих ітерацій.
2. Реалізувати метод простих ітерацій на довільній мові програмування.
3. Продемонструвати результати роботи алгоритму на конкретному прикладі.

Розв'язок системи:



$$(22) \begin{cases} \operatorname{tg}(y-x) + xy = 0,3 \\ x^2 + y^2 = 1,5 \end{cases}$$

$$\begin{cases} x = \frac{0,3 - \operatorname{tg}(y-x)}{y} = \varphi_1(x) \\ y = \sqrt{1,5 - x^2} = \varphi_2(x) \end{cases}$$

$$\frac{\partial \varphi_1}{\partial x} = \frac{1}{\cos^2(y-x) \cdot y}$$

$$\frac{\partial \varphi_1}{\partial y} = \frac{-y \cdot \frac{1}{\cos^2(y-x)} - 0,3 + \operatorname{tg}(y-x)}{y^2}$$

$$\frac{\partial \varphi_2}{\partial x} = -\frac{x}{\sqrt{1,5 - x^2}}$$

$$\frac{\partial \varphi_2}{\partial y} = 0$$

Візьмемо $x^{(0)} = 1$ $y^{(0)} = 0,4$, Область $x^{(0)}$: $G = \{ |x_1| \leq 0,1, |x_2 - 0,4| \leq 0,1 \}$

$$\left| \frac{\partial \varphi_1}{\partial x} \right| = 1,5626 \quad \left| \frac{\partial \varphi_2}{\partial x} \right| = 1,4142$$

$$\left| \frac{\partial \varphi_1}{\partial y} \right| = 2,80881 \quad \left| \frac{\partial \varphi_2}{\partial y} \right| = 0$$

Період: $\left| \frac{\partial \varphi_1}{\partial x} \right| + \left| \frac{\partial \varphi_1}{\partial y} \right| = 4,3714 > 1$

$$\left| \frac{\partial \varphi_2}{\partial x} \right| + \left| \frac{\partial \varphi_2}{\partial y} \right| = 1,4142 > 1$$

Отже, умова збіжності методу простих ітерацій не виконується.

Реалізація алгоритму:

Для реалізації алгоритму я обрала мову програмування Python. Алгоритм написаний з використанням бібліотеки numpy та sympy.

Глобальні змінні

MAX_ITERATIONS – максимальна кількість ітерацій, у випадку перевищення програма завершає роботу.

message – змінна для збереження інформації про ітерації.

```
1 from sympy import diff, tan, sqrt, symbols
2 import numpy as np
3
4 MAX_ITERATIONS = 100
5 message = ""
6
```

Функція isIterable :

Аргументи: J - функція $F'(x)$, що вертає квадратну матрицю значень x та y, xx – початкове наближення (x_0, y_0).

Функція перевіряє чи виконується умова збіжності методу простих ітерацій.

```
6
7 def isIterable(J, xx):
8     J_0 = J(xx)
9     print(J_0)
10    if abs(J_0[0][0]) + abs(J_0[0][1]) >= 1 or abs(J_0[1][0]) + abs(J_0[1][1]) >= 1:
11        raise ValueError('Limit condition is not suitable for the method')
12
```

Функція iteration (реалізація алгоритму):

Аргументи: F – функція системи нелінійних рівнянь, що вертає масив значень в точках x та y, J - функція $F'(x)$, що вертає квадратну матрицю значень x та y, xx – початкове наближення (x_0, y_0), eps – значення E.

Метод простих ітерацій складається з наступних частин:

Рядок 17 – перевіряємо чи виконується умова збіжності, якщо ні, припиняємо роботу;

Рядок 18 – розпочинаємо цикл довжини максимальної кількості ітерацій;

Рядок 20 – обчислюємо нові значення F в x_i та y_i :

Рядок 25 – зупиняємо ітерацію, якщо при кожній наступній ітерації значення x змінюються менше ніж на E.

Рядок [23-24] – заповнюємо глобальну змінну message інформацією про ітерації та різницю.

```

13
14 def iteration(F, J, xx, eps):
15     global message
16     try:
17         isIterable(J, xx)
18         for iter in range(MAX_ITERATIONS):
19             x_new = F(xx)
20             delta = np.absolute(x_new-xx)
21             xx = x_new
22
23             message += "Iteration " + str(iter + 1) + ": " + str(xx) + "\n"
24             message += "Difference: " + str(delta) + "\n\n"
25             if np.all(delta < eps):
26                 break
27         return xx
28     except ValueError as e:
29         message += e
30

```

Функція demonstration(вивід усіх отриманих результатів):

f_1, f_2 – рівняння системи;

$F(x)$ – функція системи нелінійних рівнянь, що вертає масив значень в точках x та y ;

$J(x)$ - функція $F'(x)$, що вертає квадратну матрицю значень x та y ;

Функції введені для конкретного прикладу.

Початкове наближення: $x_0 = 1$ та $y_0 = 0.7$

$E = 0.001$

```

30
31 def demonstration():
32     x, y = symbols('x y')
33     f1 = (0.3-tan(y-x))/y
34     f2 = sqrt(1.5 - x**2)
35
36     def F(xx):
37         return np.array(
38             [f1.subs((x, xx[0]), (y, xx[1])),
39              f2.subs((x, xx[0]), (y, xx[1]))])
40
41     def J(xx):
42         return np.array(
43             [[diff(f1, x).subs([(x, xx[0]), (y, xx[1])]), diff(f1, y).subs([(x, xx[0]), (y, xx[1])])],
44              [diff(f2, x).subs([(x, xx[0]), (y, xx[1])]), diff(f2, y).subs([(x, xx[0]), (y, xx[1])])]])
45
46     x = iteration(F, J, xx=np.array([1, 0.7]), eps=0.001)
47     print("\nIterations:")
48     print(message)
49     print("Solution:")
50     print(x)
51

```

Результати роботи програми на конкретному прикладі:

Результат для $x_0 = 1$ та $y_0 = 0.7$

```
Iterations:  
Limit condition is not suitable for the method  
Solution:  
None
```

Оскільки не виконується умова збіжності. Продемонструю роботу програми для прикладу запропонованого на практичній.

```
Iterations:  
Iteration 1: [3.47850542618522 2.69412117498195]  
Difference: [0.0214945738147820 0.494121174981946]  
  
Iteration 2: [3.58915326065225 2.68669194458339]  
Difference: [0.110647834467028 0.00742923039855903]  
  
Iteration 3: [3.64614285626131 2.72450035842661]  
Difference: [0.0569895956090680 0.0378084138432255]  
  
Iteration 4: [3.68541936555836 2.74356564121392]  
Difference: [0.0392765092970455 0.0190652827873032]  
  
Iteration 5: [3.71067963940584 2.75655080368217]  
Difference: [0.0252602738474805 0.0129851624682527]  
  
Iteration 6: [3.72706822712172 2.76483720905500]  
Difference: [0.0163885877158769 0.00828640537282732]  
  
Iteration 7: [3.73765153877681 2.77018663751045]  
Difference: [0.0105833116550955 0.00534942845545228]  
  
Iteration 8: [3.74447927236044 2.77363011033154]  
Difference: [0.00682773358362976 0.00344347282109281]  
  
Iteration 9: [3.74887962057556 2.77584706998936]  
Difference: [0.00440034821511626 0.00221695965782231]  
  
Iteration 10: [3.75171391589199 2.77727396975216]  
Difference: [0.00283429531643664 0.00142689976280153]  
  
Iteration 11: [3.75353879559169 2.77819226392160]  
Difference: [0.00182487969969181 0.000918294169436784]  
  
Iteration 12: [3.75471346591194 2.77878319000551]  
Difference: [0.00117467032025065 0.000590926083905874]  
  
Iteration 13: [3.75546947743708 2.77916343362252]  
Difference: [0.000756011525144817 0.000380243617011278]  
  
Solution:  
[3.75546947743708 2.77916343362252]
```

Висновок

Виконуючи дану практичну роботу, я навчилась розв'язувати систему нелінійних рівнянь за допомогою методу простих ітерацій та реалізувала алгоритм розв'язання на мові Python з використанням можливостей бібліотек numpy та sympy.