

Комп'ютерний практикум №2

Розв'язання систем лінійних алгебраїчних рівнянь (СЛАР) прямими методами

Виконав:

Студент 3 курсу ФТІ

групи ФІ-92

Поночевний Назар Юрійович

Варіант 14

Завдання:

Розв'язати систему рівнянь з кількістю значущих цифр $m = 6$. Якщо матриця системи симетрична, то розв'язання проводити за методом квадратних коренів, якщо матриця системи несиметрична, то використати метод Гауса. Вивести всі проміжні результати (матриці A , що отримуються в ході прямого ходу методу Гауса, матрицю зворотного ходу методу Гауса, або матрицю T та вектор u для методу квадратних коренів), та розв'язок системи. Навести результат перевірки: вектор нев'язки $r = |b - Ax|$, де x - отриманий розв'язок.

```
"""
Solving a system of equations
"""

# ----- Input -----

import numpy as np

A = np.array([[2.12, 0.42, 1.34, 0.88],
              [0.42, 3.95, 1.87, 0.43],
              [1.34, 1.87, 2.98, 0.46],
              [0.88, 0.43, 0.46, 4.44]])

b = np.array([11.172, 0.115, 0.009, 9.349]).reshape(-1, 1)

# ----- Code -----

def square_root_method(M):
    A = np.copy(M)
    n = len(A)
```

```

T = np.zeros_like(A)

for i in range(n):
    T[i, i] = A[i, i] ** 0.5
    T[i, i + 1:] = A[i, i + 1:] / T[i, i]
    for j in range(i + 1, n):
        A[j, j:] = A[j, j:] - T[i, j] * T[i, j:]

return T.T

def backward_pass1(T, b):
    y = np.zeros_like(b)
    n = len(y)

    y[0, 0] = b[0, 0] / T[0, 0]
    for i in range(1, n):
        suma = sum([T[i, j] * y[j, 0] for j in range(i)])
        y[i, 0] = (b[i, 0] - suma) / T[i, i]

    return y

def backward_pass2(tT, y):
    x = np.zeros_like(y)
    n = len(x)

    x[n - 1, 0] = y[n - 1, 0] / tT[n - 1, n - 1]
    for i in range(n - 2, -1, -1):
        suma = sum([tT[i, j] * x[j, 0] for j in range(n - 1, i, -1)])
        x[i, 0] = (y[i, 0] - suma) / tT[i, i]

    return x

def main():
    T = square_root_method(A)
    print(f"T:\n{T}")
    print(f"\nT':\n{T.T}")

    y = backward_pass1(T, b)
    print(f"\ny:\n{y}")

    x = backward_pass2(T.T, y)
    print(f"\nx:\n{x}")

```

```
r = np.sum(np.abs(b - A.dot(x)))
print(f"r: {r}")

if __name__ == "__main__":
    main()
```

```
T:
[[ 1.45602198  0.          0.          0.          ]
 [ 0.28845718  1.96641614  0.          0.          ]
 [ 0.92031578  0.81596579  1.21128803  0.          ]
 [ 0.60438648  0.13001336 -0.16702293  2.00746528]]

T':
[[ 1.45602198  0.28845718  0.92031578  0.60438648]
 [ 0.          1.96641614  0.81596579  0.13001336]
 [ 0.          0.          1.21128803 -0.16702293]
 [ 0.          0.          0.          2.00746528]]

y:
[[ 7.6729611 ]
 [-1.06707869]
 [-5.10353219]
 [ 1.99151322]]

x:
[[ 7.22006384]
 [ 1.0833107 ]
 [-4.07651719]
 [ 0.99205363]]
r: 2.086872341600099e-15
```