

Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

МЕТОДИ ОБЧИСЛЕНЬ
Комп'ютерний практикум №2
Розв'язання систем лінійних алгебраїчних рівнянь (СЛАР) прямими методами
Варіант 3 (13)

Виконала:
студентка 3 курсу
гр. ФБ-92
Шатковська Діана

Київ - 2022

1. Допрограмний етап

Дана СЛАР (Варіант 3):

$$\begin{pmatrix} 8,30 & 2,62 & 4,10 & 1,90 & | & -10,65 \\ 3,92 & 8,45 & 8,78 & 2,46 & | & 12,21 \\ 3,77 & 7,21 & 8,04 & 2,28 & | & 15,45 \\ 2,21 & 3,65 & 1,69 & 6,99 & | & -8,35 \end{pmatrix}$$

Не є симетричною, тому для її розв'язання застосуємо метод Гауса.

2. Програмний етап

Лістинг програми:

```
# Calc Methods Lab 2 (Solving Linear Systems) by Diana Shatkovska
FB-92, Variant 3
from numpy import array, matmul, subtract

a = array([[8.30, 2.62, 4.10, 1.90],
           [3.92, 8.45, 8.78, 2.46],
           [3.77, 7.21, 8.04, 2.28],
           [2.21, 3.65, 1.69, 6.99]], float)

b = array([-10.65, 12.21, 15.45, -8.35], float)
n = len(b)

print(f"The given matrix is : \n{a}")
print(f"The b vector is : \n{b}")
print('-----')

# 1st step: elimination
def elimination(matrix, b_vector):
    for k in range(n-1):
        for i in range(k+1, n):
            if matrix[i, k] == 0:
                continue
            factor = matrix[k, k]/matrix[i, k]
            for j in range(k, n):
                matrix[i, j] = matrix[k, j] - matrix[i, j]*factor
            b_vector[i] = b_vector[k] - b_vector[i]*factor
        print(f"{k}-{i} step:\n{matrix}")
```

```

        print(f"b vector: \n{b_vector}\n")
    return matrix, b_vector

triangled_a, triangled_b = elimination(a, b)

print(f"The triangled matrix is : \n{triangled_a}")
print(f"The b vector is : \n{triangled_b}")
print('-----')

# 2nd step: back-substitution
def back_substitution(triangled_matrix, b_vector):
    x_vector = [0] * n
    x_vector[n-1] = b_vector[n-1]/triangled_matrix[n-1, n-1]

    for i in range(n-2, -1, -1):
        sum_ax = 0
        for j in range(i+1, n):
            sum_ax += triangled_matrix[i, j] * x_vector[j]
        x_vector[i] = (b_vector[i] - sum_ax) / triangled_matrix[i, i]
    return x_vector

x = back_substitution(triangled_a, triangled_b)
print(f"The solution is : \n{x}")

# calculating r = |b-Ax|
r_vector = abs(subtract(b, matmul(a, x)))
# r_vector = [f'{el:.6f}' for el in r_vector]
print(f"Вектор невязки: {r_vector}")

```

Результат виконання програми:

The given matrix is :

```
[[8.3 2.62 4.1 1.9 ]
 [3.92 8.45 8.78 2.46]
 [3.77 7.21 8.04 2.28]
 [2.21 3.65 1.69 6.99]]
```

The b vector is :

```
[-10.65 12.21 15.45 -8.35]
```

0-1 step:

```
[[ 8.3      2.62      4.1      1.9      ]
 [ 0.      -15.27158163 -14.49030612 -3.30867347]
 [ 3.77     7.21     8.04     2.28     ]
 [ 2.21     3.65     1.69     6.99     ]]
```

b vector:

```
[-10.65 -36.50280612 15.45 -8.35 ]
```

0-2 step:

```
[[ 8.3      2.62      4.1      1.9      ]
 [ 0.      -15.27158163 -14.49030612 -3.30867347]
 [ 0.      -13.2534748 -13.60079576 -3.11962865]
 [ 2.21     3.65     1.69     6.99     ]]
```

b vector:

```
[-10.65 -36.50280612 -44.66458886 -8.35 ]
```

0-3 step:

```
[[ 8.3      2.62      4.1      1.9      ]
 [ 0.      -15.27158163 -14.49030612 -3.30867347]
 [ 0.      -13.2534748 -13.60079576 -3.11962865]
 [ 0.      -11.0881448 -2.24705882 -24.3520362 ]]
```

b vector:

```
[-10.65 -36.50280612 -44.66458886 20.70972851]
```

1-2 step:

```
[[ 8.3      2.62      4.1      1.9      ]
 [ 0.      -15.27158163 -14.49030612 -3.30867347]
 [ 0.      0.      1.18148303 0.28598108]
 [ 0.      -11.0881448 -2.24705882 -24.3520362 ]]
```

b vector:

```
[-10.65 -36.50280612 14.96286044 20.70972851]
```

1-3 step:

```
[[ 8.3      2.62      4.1      1.9      ]
 [ 0.      -15.27158163 -14.49030612 -3.30867347]
 [ 0.      0.      1.18148303 0.28598108]
 [ 0.      0.      -11.39545636 30.23112201]]
```

b vector:

[-10.65 -36.50280612 14.96286044 -65.0260907]

2-3 step:

```
[[ 8.3        2.62        4.1        1.9        ]  
 [ 0.       -15.27158163 -14.49030612 -3.30867347]  
 [ 0.        0.        1.18148303  0.28598108]  
 [ 0.        0.        0.        3.42034943]]
```

b vector:

[-10.65 -36.50280612 14.96286044 8.22094327]

The triangled matrix is :

```
[[ 8.3        2.62        4.1        1.9        ]  
 [ 0.       -15.27158163 -14.49030612 -3.30867347]  
 [ 0.        0.        1.18148303  0.28598108]  
 [ 0.        0.        0.        3.42034943]]
```

The b vector is :

[-10.65 -36.50280612 14.96286044 8.22094327]

The solution is :

[-4.773098987064921, -9.595050885653148, 12.08269043946597, 2.4035390059155906]

Вектор нев'язки: [1.77635684e-15 2.84217094e-14 0.00000000e+00 0.00000000e+00]

Висновки:

При виконанні цього комп'ютерного практикуму було розглянуто метод Гауса для розв'язання СЛАР та запрограмовано його мовою Python. За отриманими результатами, та, зокрема, перевірки вектору нев'язки було отримано досить точний розв'язок СЛАР із похибкою аж у 14-15му розряді десяткового числа.