Практическое задание 4.1

In [1]:

```
import sympy as sym
import numpy as np
np.set_printoptions(precision=2, suppress=True)
import warnings
warnings.filterwarnings("ignore")
```

1. Решить систему уравнений методом Гаусса:

$$\begin{cases} x_1 + x_2 - x_3 - 2x_4 = 0, \\ 2x_1 + x_2 - x_3 + x_4 = -2, \\ x_1 + x_2 - 3x_3 + x_4 = 4. \end{cases}$$

Преобразуем расширенную матрицу системы путем элементарных преобразований:

$$\left(\begin{array}{cccc|c} 1 & 1 & -1 & -2 & 0 \\ 2 & 1 & -1 & 1 & -2 \\ 1 & 1 & -3 & 1 & 4 \end{array}\right)$$

Вычтем из 3 строки 1

$$\left(\begin{array}{cccc|c} 1 & 1 & -1 & -2 & 0 \\ 2 & 1 & -1 & 1 & -2 \\ 0 & 0 & -2 & 3 & 4 \end{array}\right)$$

Вычтим из 2 строки 1, умноженную на 2

$$\left(\begin{array}{cccc|c} 1 & 1 & -1 & -2 & 0 \\ 0 & -1 & 1 & 5 & -2 \\ 0 & 0 & -2 & 3 & 4 \end{array}\right)$$

Мы видим, что система имеет *бесконечное количество решений.* Запишем его в виде *общего решения*.

Оставшаяся матрица соответствует системе

$$\begin{cases} x_1 + x_2 - x_3 - 2x_4 = 0, \\ -x_2 + x_3 + 5x_4 = -2, \\ -2x_3 + 3x_4 = 4. \end{cases}$$

В данном случае можно рассматривать в качестве *свободного* параметра $x_4 = c$. Тогда, выражая остальные переменные через c, получим:

$$x_4 = c,$$
$$x_3 = -2 + 3/2 \cdot \mathrm{c},$$
$$x_2 = 13/2 \cdot c,$$

$$x_1 = -2 - 3 \cdot c.$$

2. Проверить на совместность и выяснить, сколько решений будет иметь система линейных уравнений:

а) $\begin{cases} 3x_1 - x_2 + x_3 = 4, \\ 2x_1 - 5x_2 - 3x_3 = -17, \\ x_1 + x_2 - x_3 = 0; \end{cases}$

Поскольку я самостоятельно написал функцию для решения уравнений методом Гауса, то я достаточно хорошо с ним разобрался и здесь для экономии времени предлагаю использовать данную функцию. Запишем уравнение в матричном виде и воспользуемся функцией.

In [2]:

```
A = np.array([[3., -1., 1., 4.], [2., -5., -3., -17.], [1., 1., -1., 0.]])
A
```

Out[2]:

```
array([[  3.,  -1.,   1.,   4.],
       [  2.,  -5.,  -3., -17.],
       [  1.,   1.,  -1.,   0.]])
```

In [3]:

```
def solv_equations(A):
    for nrow in range(len(A)):
        for i in range(nrow + 1):
            if nrow != len(A) - 1:
                A[nrow + 1] = A[nrow + 1] - A[i] * (A[nrow + 1][i] / A[i][i])
            else:
                break
    print("Диагональная матрица:")
    print(A)
    roots = []
    if np.count_nonzero(A[len(A)-1])==2:
        for i in reversed(range(1, len(A) + 1)):
            x = (A[i - 1][len(A)] / A[i - 1][i - 1])
            roots.append(x)
            A[:, i - 1] = A[:, i - 1] * x
            A[:, len(A)] = A[:, len(A)] - A[:, i - 1]
        print("Данная СЛАУ является совместной со следующими корнями: ", roots[::-1])

    elif np.count_nonzero(A[len(A)-1]) < 2 or np.count_nonzero(A[len(A)-2]) < 2:
        print("Диагональная матрица:")
        print(A)
        print("Данная СЛАУ является несовместной и решений не имеет")

    elif np.count_nonzero(A[len(A)-1]) > 2:
        print("Диагональная матрица:")
        print(A)
        print("Данная СЛАУ является совместной и имеет бесконечное множество решений")
```

In [4]:

```
solv_equations(A)
```

```
Диагональная матрица:
[[  3.     -1.      1.      4.   ]
 [  0.     -4.33  -3.67 -19.67]
 [  0.      0.    -2.46  -7.38]]
Данная СЛАУ является совместной со следующими корнями:  [1.0000000000000002,
2.0000000000000004, 3.0]
```

Проверим

In [5]:

```
M1 = np.array([[3., -1., 1.], [2, -5, -3], [1, 1, -1]])
v1 = np.array([4., -17., 0])
np.linalg.solve(M1, v1)
```

Out[5]:

```
array([1., 2., 3.])
```

Видим, что корни найдены правильно

б) $\begin{cases} 2x_1 - 4x_2 + 6x_3 = 1, \\ x_1 - 2x_2 + 3x_3 = -2, \\ 3x_1 - 6x_2 + 9x_3 = 5; \end{cases}$

Запишем уравнение в матричном виде и воспользуемся функцией.

In [6]:

```
A = np.array([[2., -4., 6., 1.], [1, -2, 3, -2], [3, -6, 9, 5]])
A
```

Out[6]:

```
array([[ 2., -4.,  6.,  1.],
       [ 1., -2.,  3., -2.],
       [ 3., -6.,  9.,  5.]])
```

In [7]:

```
solv_equations(A)
```

```
Диагональная матрица:
[[ 2.  -4.   6.   1. ]
 [ 0.   0.   0.  -2.5]
 [ nan  nan  nan  nan]]
Диагональная матрица:
[[ 2.  -4.   6.   1. ]
 [ 0.   0.   0.  -2.5]
 [ nan  nan  nan  nan]]
Данная СЛАУ является несовместной и решений не имеет
```

In [8]:

```
M1 = np.array([[2., -4., 6.], [1, -2, 3], [3, -6, 9]])
v1 = np.array([1., -2., 5])
np.linalg.solve(M1, v1)
```

```
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
<ipython-input-8-11e31800b9dd> in <module>
      1 M1 = np.array([[2., -4., 6.], [1, -2, 3], [3, -6, 9]])
      2 v1 = np.array([1., -2., 5])
----> 3 np.linalg.solve(M1, v1)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\linalg\linalg.py in solve(a,
b)
    401     signature = 'DD->D' if isComplexType(t) else 'dd->d'
    402     extobj = get_linalg_error_extobj(_raise_linalgerror_singular)
--> 403     r = gufunc(a, b, signature=signature, extobj=extobj)
    404
    405     return wrap(r.astype(result_t, copy=False))

C:\ProgramData\Anaconda3\lib\site-packages\numpy\linalg\linalg.py in _raise_l
inalgerror_singular(err, flag)
     95
     96 def _raise_linalgerror_singular(err, flag):
---> 97     raise LinAlgError("Singular matrix")
     98
     99 def _raise_linalgerror_nonposdef(err, flag):

LinAlgError: Singular matrix
```

Видим, что данное СЛАУ действительно не имеет решний

в) $\begin{cases} x_1 + 2x_2 + 5x_3 = 4, \\ 3x_1 + x_2 - 8x_3 = -2. \end{cases}$

Запишем уравнение в матричном виде и воспользуемся функцией.

In [9]:

```
A = np.array([[1., 2., 5., 4.], [3, 1, -8, -2]])
A
```

Out[9]:

```
array([[ 1.,  2.,  5.,  4.],
       [ 3.,  1., -8., -2.]])
```

In [10]:

```
solv_equations(A)
```

```
Диагональная матрица:
[[  1.   2.   5.   4.]
 [  0.  -5. -23. -14.]]
Диагональная матрица:
[[  1.   2.   5.   4.]
 [  0.  -5. -23. -14.]]
Данная СЛАУ является совместной и имеет бесконечное множество решений
```

3. Проверить на совместность и выяснить, сколько решений будет иметь система линейных уравнений, заданная расширенной матрицей

$$\tilde{A} = \left(\begin{array}{cccc|c} 1 & 3 & -2 & 4 & 3 \\ 0 & 5 & 0 & 1 & 2 \\ 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 0 & 2 & 1 \end{array}\right).$$

Запишем уравнение в матричном виде и воспользуемся функцией.

In [11]:

```
A = np.array([[1., 3., -2., 4., 3.], [0., 5., 0., 1., 2.], [0., 0., 3., 0., 4.], [0., (
A
```

Out[11]:

```
array([[ 1.,  3., -2.,  4.,  3.],
       [ 0.,  5.,  0.,  1.,  2.],
       [ 0.,  0.,  3.,  0.,  4.],
       [ 0.,  0.,  0.,  2.,  1.]])
```

In [12]:

```
solv_equations(A)
```

```
Диагональная матрица:
[[ 1.  3. -2.  4.  3.]
 [ 0.  5.  0.  1.  2.]
 [ 0.  0.  3.  0.  4.]
 [ 0.  0.  0.  2.  1.]]
Данная СЛАУ является совместной со следующими корнями:  [2.7666666666666666,
0.3, 1.3333333333333333, 0.5]
```

Проверим

In [13]:

```
M1 = np.array([[1., 3., -2., 4.], [0., 5., 0., 1.], [0., 0., 3., 0.], [0., 0., 0., 2.]]
v1 = np.array([ 3., 2., 4., 1.])
np.linalg.solve(M1, v1)
```

Out[13]:

```
array([2.77, 0.3 , 1.33, 0.5 ])
```

Видим, что корни найдены правильно

4. Дана система линейных уравнений, заданная расширенной матрицей

$$\tilde{A} = \left(\begin{array}{ccc|c} 1 & 2 & 3 & a \\ 4 & 5 & 6 & b \\ 7 & 8 & 9 & c \end{array}\right).$$

Найти соотношение между параметрами a, b и c, при которых система является несовместной.

После преобразования расширенной матрицы данной системы линейных уравнений путем элементарных преобразований получаем следующую диагональную матрицу:

$$\tilde{A} = \left(\begin{array}{ccc|c} 1 & 2 & 3 & a \\ 0 & -3 & -6 & -4a + b \\ 0 & 0 & 0 & a - 2b + \mathrm{c} \end{array}\right).$$

Из данной диагональной матрицы мы видим, что система будет несовместной при условии, что $a - 2b + \mathrm{c} \neq 0$. Если $a - 2b + \mathrm{c} = 0$, то сисема будет иметь бесконечное множество решений.