Здравствуйте, Надежда Александровна. Ниже представлен листинг кода, который выполняет (и, на всякий случай выводит) все нужные задачи.

import numpy as np

from sympy import \*

A = np.array([[-7,4,0], [0,-1,0], [-1,5,7]])

print(A)

print("\n")

E = np.eye(3)

print(E)

print("\n")

A = np.array([1,2,3])

print(A)

print("\n")

A = np.array ([[-7,4,0], [0,-1,0], [-1,5,7]])

print(A[0,1])

print("\n")

print(A[:,1:3])

print("\n")

A = np.array([[1,2,3], [4,5,6], [7,8,9]])

E = np.eye(3)

print(A-E)

print("\n")

print(3\*A)

print("\n")

А = np.array([[1,2,3], [4,5,6]])

В = np.array([[0,0,0], [2,2,2]])

print(А\*В)

print("\n")

B = np.array([[1,0],[0,-2],[2,1]])

print(B.T)

print("\n")

A = np.array([[-7,4,0],[0,-1,0],[-1,5,7]])

В = np.array([[1,0],[0,-2],[2,1]])

F = А@В

print(F)

print("\n")

F = np.dot(A,B)

print(F)

print("\n")

F = np.matmul(A,B)

print(F)

print("\n")

C = np.array([[7,-3],[1,2]])

print(np.linalg.det(C))

print("\n")

print(np.linalg.inv(C))

print("\n")

print(np.linalg.matrix\_rank(A))

print("\n")

a = Matrix([[1,2,3], [0,-1, 1]])

print(a)

print("\n")

x,y,z = symbols('x у z')

v = Matrix([[1,x],[y,z]])

print(v)

print("\n")

print(eye(3))

print("\n")

print(zeros(2,3))

print("\n")

print(ones(3,2))

print("\n")

print(diag(1,5,-2))

print("\n")

print(diag(-1, ones(2, 2), Matrix([5, 7, 5])))

print("\n")

W = Matrix([[1,2,3]])

print(W)

print("\n")

W = Matrix([[1],[2],[3]])

print(W)

print("\n")

A = Matrix([[1,2,3], [0,-1, 1]])

print(A.shape)

print("\n")

a11, a12, a13, a21, a22, a23, a31, a32, a33 = symbols('a11 a12 a13 a21 a22 a23 a31 a32 a33 ')

A = Matrix([[a11, a12, a13],[a21, a22, a23],[a31, a32, a33]])

print(A)

print("\n")

print(A[2,0])

print("\n")

print(A[:, 1:2])

print("\n")

print(A.row(0))

print("\n")

print(A.col(1))

print("\n")

A.row\_del(0)

print(A)

print("\n")

A.col\_del(1)

print(A)

print("\n")

B = Matrix([[1,2,3], [7,8,9]])

I = B.row\_insert(1, Matrix([[4,5,6]]))

print(I)

print("\n")

D = I.col\_insert(3,Matrix([4,1,6]))

print(D)

print("\n")

V = Matrix([[1,x],[y,z]])

print(V\*V)

print("\n")

print(B.T)

print("\n")

D = Matrix([[0,1],[1,0]])

print(det(D))

print("\n")

print(D.inv())

print("\n")

A = Matrix([[1,8,3],[4,5,6],[7,8,9]])

print(A\*\*-1)

print("\n")

A = Matrix([[1, -1, 2],[2, -2, 4],[-1, 1, 2]])

print(A.rank())

print("\n")

givenMat = Matrix([[1,3,4],[4,-1,0],[3,2,5],[-2,0,1],[4,6,7]])

givenMat.T

print(givenMat.columnspace())

print("\n")

givenMat = Matrix([[1, 2,0,9],[-3,7, 11,5],[-9,4,25,84],[3,12,-5,58]])

print(givenMat.minorEntry(2,1))

print("\n")

print(givenMat.cofactor\_matrix())

print("\n")

row = [0,2]

colum = [2,3]

print(det(givenMat.extract(row,colum)))

print("\n")

givenMat = Matrix([[1,3,2,4,5],[0,0,-1,2,7],[3,9,6,12,15],[5,15,9,26,22],[1,3,1,10,2]])

reduced\_mat, pivots = givenMat.rref()

print(det(reduced\_mat))

print("\n")

Minor = Matrix()

for i in range(reduced\_mat.rows):

row = reduced\_mat.row(i)

position = 0

flag = 0

for j in range(reduced\_mat.cols):

if row[j] != 0:

flag = 1

break

if flag == 1:

Minor = Minor.row\_insert(position, row)

position += 1

print("Базисные строки:")

for row in Minor.tolist():

print(row)

print("\nБазисные столбцы:")

for row in Minor.T.tolist():

print(row)

print("\nБазисный минор:\n")

print(Minor)

print("\n")

A = np.array([[3,2,0],

[1,-1,0],

[0,5,1]])

b = np.array([[2],[4],[-1]])

u = np.linalg.solve(A,b)

print(u)

print("\n")

A = Matrix([[3,2,0],

[1,-1,0],

[0,5,1]])

b = Matrix([[2],[4],[-1]])

x = A\*\*-1 \* b

print(x)

print("\n")