Task 1

Вектор – это частный случай матрицы 1xN и Nx1. Повторите материал для векторов, уделяя особое внимание умножению A·B. Вычислите, по возможности не используя программирование: $(5\mathrm{E})^{-1}$, где E – единичная матрица размера 5x5

In [1]:

```
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
```

In [2]:

```
E = np.identity(5)
E
```

In [3]:

```
A = 5*E
A
```

In [4]:

```
#Определитель диагональной матрицы равен произведению элементов стоящих на главной диаг
онали
D = 5**5
D
```

Out[4]:

```
3125
```

In [5]:

```
A11 = 5**4
A11
```

Out[5]:

```
625
```

A11 = Ann = 625

In [6]:

```
A_1 = np.identity(5)*(A11/D)
A_1
```

Out[6]:

```
array([[0.2, 0. , 0. , 0. , 0. ],
       [0. , 0.2, 0. , 0. , 0. ],
       [0. , 0. , 0.2, 0. , 0. ],
       [0. , 0. , 0. , 0.2, 0. ],
       [0. , 0. , 0. , 0. , 0.2]])
```

Проверим: $A \cdot A^{-1} = E$

In [7]:

```
np.dot(A_1, A)
```

Out[7]:

```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

Task 2

Вычислите определитель:

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

$$\Delta = 1 \cdot 0 \cdot 9 + 2 \cdot 6 \cdot 7 + 3 \cdot 8 \cdot 4 - 7 \cdot 0 \cdot 3 - 6 \cdot 8 \cdot 1 - 9 \cdot 4 \cdot 2$$

$$\Delta = 60$$

Task 3

Вычислите матрицу, обратную данной:

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

In [8]:

```
A = np.matrix([[1, 2,3], [4,0,6],[7,8,9]])
A
```

Out[8]:

```
matrix([[1, 2, 3],
        [4, 0, 6],
        [7, 8, 9]])
```

In [9]:

```
A11 = -6*8
A12 = -(4*9-6*7)
A13 = 4*8
A21 = -(2*9 - 3*8)
A22 = 1*9 - 3*7
A23 = -(1*8 - 2*7)
A31 = 2*6
A32 = -(1*6 - 3*4)
A33 = -2*4
```

In [10]:

```
A_1 = np.matrix([[A11, A12, A13], [A21, A22, A23], [A31, A32, A33]])/60
A_inv = A_1.T
A_inv
```

Out[10]:

```
matrix([[-0.8       ,  0.1       ,  0.2       ],
        [ 0.1       , -0.2       ,  0.1       ],
        [ 0.53333333,  0.1       , -0.13333333]])
```

In [11]:

```
np.linalg.inv(A)
```

Out[11]:

```
matrix([[-0.8       ,  0.1       ,  0.2       ],
        [ 0.1       , -0.2       ,  0.1       ],
        [ 0.53333333,  0.1       , -0.13333333]])
```

Проверим:

In [12]:

```
np.dot(A, A_inv)
```

Out[12]:

```
matrix([[ 1.00000000e+00, -2.77555756e-17,  2.77555756e-17],
        [-2.22044605e-16,  1.00000000e+00,  5.55111512e-17],
        [-7.77156117e-16,  2.77555756e-17,  1.00000000e+00]])
```

In [13]:

```
np.dot(A_inv, A)
```

Out[13]:

```
matrix([[ 1.00000000e+00,  0.00000000e+00, -1.66533454e-16],
        [-2.77555756e-17,  1.00000000e+00, -8.32667268e-17],
        [ 2.77555756e-17,  0.00000000e+00,  1.00000000e+00]])
```

Task 4

Приведите пример матрицы 4x4, ранг которой равен 1

In [14]:

```
a = np.matrix([[1,2,3,4], [2,4,6,8], [3,6,9,12], [4,8,12,16]])
a
```

Out[14]:

```
matrix([[ 1,  2,  3,  4],
        [ 2,  4,  6,  8],
        [ 3,  6,  9, 12],
        [ 4,  8, 12, 16]])
```

In [15]:

```
#np.ndim(a)
np.linalg.matrix_rank(a)
```

Out[15]:

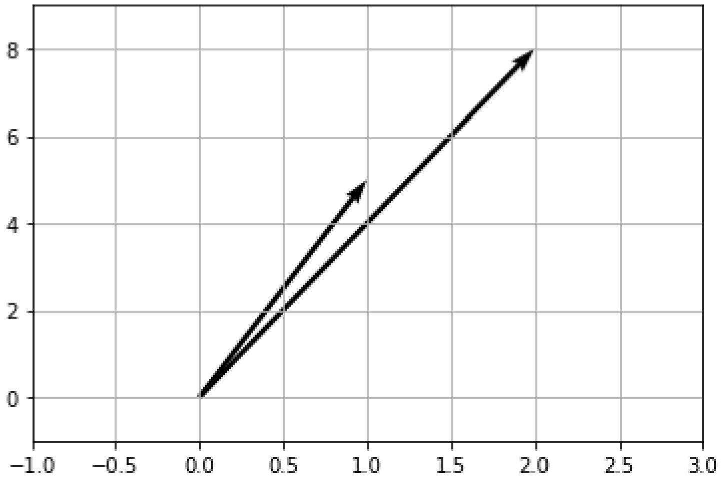
```
1
```

Task 5

Вычислите скалярное произведение двух векторов:
(1, 5) и (2, 8)

In [16]:

```
a = np.array([1,5])
b = np.array([2,8])
X, Y = np.array([0, 0]), np.array([0, 0])
U, V = np.array([a[0], b[0]]), np.array([a[1], b[1]])
plt.quiver(X, Y, U, V, angles='xy', scale_units = 'xy', scale=1)
plt.xlim(-1, 3)
plt.ylim(-1, 9)
plt.grid()
plt.show()
```



In [17]:

```
s = 2 + 5*8
s
```

Out[17]:

```
42
```

Task 6

Вычислите смешанное произведение трех векторов: (1, 5, 0), (2, 8, 7) и (7, 1.5, 3)

In [18]:

```
a = np.array([1,5,0])
b = np.array([2,8,7])
c = np.array([7,1.5,3])
```

$$\vec{a}\, x\, \vec{b} = \begin{vmatrix} i & j & k \\ 1 & 5 & 0 \\ 2 & 8 & 7 \end{vmatrix}$$

In [19]:

```
ab = np.array([(5*7), -7, 8-10])
ab
```

Out[19]:

```
array([35, -7, -2])
```

In [20]:

```
v = np.cross(a, b)
v
```

Out[20]:

```
array([35, -7, -2])
```

In [21]:

```
vc = 35*7 - 7*1.5 - 2*3
vc
```

Out[21]:

```
228.5
```

In [22]:

```
np.inner(v, c)
```

Out[22]:

```
228.5
```

Task 7

Решите линейную систему:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot X = \begin{bmatrix} 12 \\ 2 \\ 1 \end{bmatrix}$$

In [23]:

```
A = np.matrix([[1, 2, 3], [4, 0, 6], [7, 8, 9]])
A
```

Out[23]:

```
matrix([[1, 2, 3],
        [4, 0, 6],
        [7, 8, 9]])
```

In [24]:

```
B = np.matrix([[12], [2], [1]])
B
```

Out[24]:

```
matrix([[12],
        [ 2],
        [ 1]])
```

In [25]:

```
#np.linalg.solve(A, B)
X = np.dot(np.linalg.inv(A), B)
X
```

Out[25]:

```
matrix([[-9.2       ],
        [ 0.9       ],
        [ 6.46666667]])
```

Task 8

Найдите псевдорешение:

$$x + 2y - z = 1$$
$$3x - 4y + 0z = 7$$
$$8x - 5y + 2z = 12$$
$$2x + 0y - 5z = 7$$
$$11x + 4y - 7z = 15$$

In [26]:

```
A = np.matrix([[1, 2, -1], [3, -4, 0], [8, -5, 2], [2, 0, -5], [11, 4, -7]])
A
```

Out[26]:

```
matrix([[ 1,  2, -1],
        [ 3, -4,  0],
        [ 8, -5,  2],
        [ 2,  0, -5],
        [11,  4, -7]])
```

In [27]:

```
B = np.matrix([1, 7, 12, 7, 15]).T
B
```

Out[27]:

```
matrix([[ 1],
        [ 7],
        [12],
        [ 7],
        [15]])
```

In [28]:

```
X, residuals, rnk, s =  np.linalg.lstsq(A, B, rcond=None)
```

In [29]:

```
X
```

Out[29]:

```
matrix([[ 1.13919353],
        [-0.90498444],
        [-0.9009803 ]])
```

In [30]:

```
np.dot(A, X)
```

Out[30]:

```
matrix([[ 0.23020495],
        [ 7.03751834],
        [11.83650981],
        [ 6.78328855],
        [15.21805313]])
```

Task 9

Сколько решений имеет линейная система:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot X = \begin{bmatrix} 12 \\ 2 \\ 1 \end{bmatrix}$$

In [31]:

```
A = np.matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
B = np.matrix([[12], [2], [1]])
```

In [32]:

```
np.linalg.det(A)
```

Out[32]:

```
-9.51619735392994e-16
```

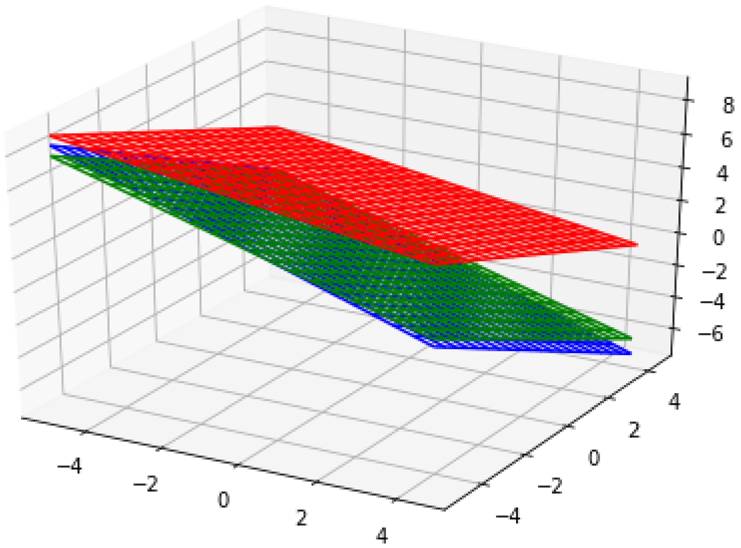
Определитель равен нулю, решений не имеет

In [33]:

```
from pylab import *
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
```

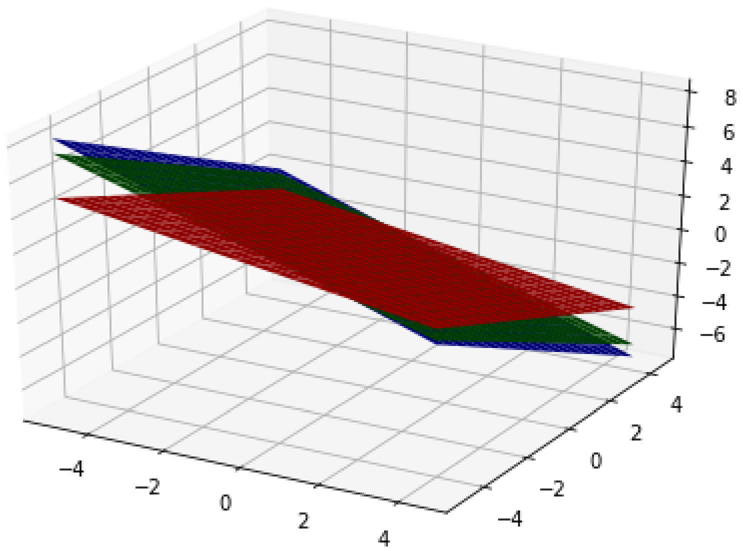
In [34]:

```
fig = figure()
ax = Axes3D(fig)
X = np.arange(-5, 5, 0.5)
Y = np.arange(-5, 5, 0.5)
X, Y = np.meshgrid(X, Y)
Z1 = 4 - 2/3*Y - 1/3*X
Z2 = 2/6 - 2/3*X - 5/6*Y
Z3 = 1/9 - 7/9*X - 8/9*Y
ax.plot_wireframe(X, Y, Z1, color='red')
ax.plot_wireframe(X, Y, Z2, color='green')
ax.plot_wireframe(X, Y, Z3, color='blue')
show()
```



In [35]:

```
fig = figure()
ax = Axes3D(fig)
X = np.arange(-5, 5, 0.5)
Y = np.arange(-5, 5, 0.5)
X, Y = np.meshgrid(X, Y)
Z1 = 0 - 2/3*Y - 1/3*X
Z2 = 0/6 - 2/3*X - 5/6*Y
Z3 = 0/9 - 7/9*X - 8/9*Y
ax.plot_surface(X, Y, Z1, color='red')
ax.plot_surface(X, Y, Z2, color='green')
ax.plot_surface(X, Y, Z3, color='blue')
show()
```



In [36]:

```
A = np.matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
B = np.matrix([[0], [0], [0]])
```

Чтобы система стала совместной изменим вектор B на [0,0,0]. В таком случае система будет иметь тривиальное решение [0,0,0]

Task 10

Вычислите LU-разложение матрицы:

$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & 16 & 21 \\ 4 & 28 & 73 \end{vmatrix}$$

После этого придумайте вектор правых частей и решите полученную линейную систему трех уравнений с данной матрицей.

In [37]:

```
import scipy
import scipy.linalg
```

In [38]:

```
A = np.matrix([[1, 2, 3], [2, 16, 21], [4, 28, 73]])
```

In [39]:

```
P, L, U = scipy.linalg.lu(A)
```

In [40]:

```
print('P\n',P,'\nL\n', L, '\nU\n', U)
```

```
P
 [[0. 1. 0.]
 [0. 0. 1.]
 [1. 0. 0.]]
L
 [[ 1.    0.    0.  ]
 [ 0.25  1.    0.  ]
 [ 0.5  -0.4   1.  ]]
U
 [[  4.    28.    73.  ]
 [  0.    -5.   -15.25]
 [  0.     0.   -21.6 ]]
```

In [41]:

```
np.dot(P.T,A)-np.dot(L,U)
```

Out[41]:

```
matrix([[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]])
```

In [42]:

```
B = np.matrix([1,2,3]).T
B
```

Out[42]:

```
matrix([[1],
        [2],
        [3]])
```

In [43]:

```
Y = np.dot(np.linalg.inv(L), B)
Y
```

Out[43]:

```
matrix([[1.  ],
        [1.75],
        [3.2 ]])
```

In [44]:

```
X = np.dot(np.linalg.inv(U), Y)
X
```

Out[44]:

```
matrix([[ 2.24074074],
        [ 0.10185185],
        [-0.14814815]])
```

Проверим:

In [45]:

```
np.dot(A, X)
```

Out[45]:

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

Значения совпали но строки почему-то сместились... ¯_(ツ)_/¯

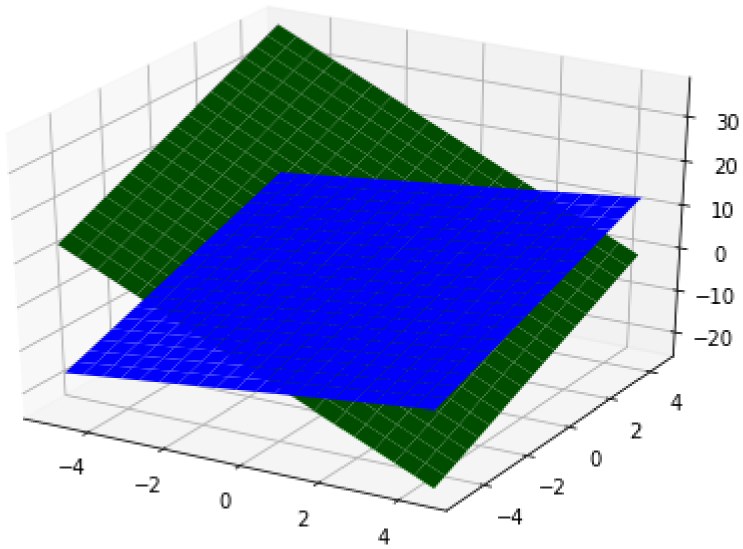
Task 11

Найдите нормальное псевдорешение недоопределенной системы:

$$x + 2y - z = 1$$
$$8x - 5y + 2z = 12$$

In [46]:

```
fig = figure()
ax = Axes3D(fig)
X = np.arange(-5, 5, 0.5)
Y = np.arange(-5, 5, 0.5)
X, Y = np.meshgrid(X, Y)
Z1 = X + 2*Y - 1
Z2 = 6 - 4*X + 5/2*Y
ax.plot_surface(X, Y, Z1, color='blue')
ax.plot_surface(X, Y, Z2, color='green')
show()
```



In [47]:

```
A = np.matrix([[1, 2, -1], [8, -5, 2]])
B = np.matrix([1, 12]).T
```

In [48]:

```
X, res, r, s = np.linalg.lstsq(A,B, rcond=None)
```

In [49]:

```
np.dot(A,X)
```

Out[49]:

```
matrix([[ 1.],
        [12.]])
```

In [50]:

```
# минимум в точке
X
```

Out[50]:

```
matrix([[ 1.38191882],
        [-0.18081181],
        [ 0.0202952 ]])
```

Task 12

Найдите одно из псевдорешений вырожденной системы:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot X = \begin{bmatrix} 2 \\ 5 \\ 11 \end{bmatrix}$$

In [51]:

```
A = np.matrix([[1,2,3],[4,5,6],[7,8,9]])
B = np.matrix([2,5,11]).T
```

In [52]:

```
np.linalg.det(A)
```

Out[52]:

```
-9.51619735392994e-16
```

In [53]:

```
X, res, r, s = np.linalg.lstsq(A,B, rcond=None)
```

In [54]:

```
X
```

Out[54]:

```
matrix([[ 1.25],
        [ 0.5 ],
        [-0.25]])
```