```
In [1]: import sympy
import numpy as np
from sympy import Expr, Eq, latex, plot_implicit, Matrix, plot, solve, linsolve, nonlinsolve, symbols, eye, Point, Line
from sympy import pi as Pi
from sympy import cos as Cos
from sympy import sin as Sin
import matplotlib.pyplot as plt
```

Занятие 12

Алгебра

Линейные операторы на плоскости и в пространстве

Линейный оператор осуществляет линейное отображение линейного пространства на линейное пространство:
$$A:\ Ax = y, \quad x \in X, y \in Y,$$
при этом
$$A(a_1x' + a_2x'') = a_1Ax' + a_2Ax'' = a_1y' + a_2y'',$$
где a_1 и a_2 - числа, x' и x'' - элементы линейного пространства X, y' и y'' - элементы линейного пространства Y.

X и Y могут быть разными или совпадать.

x' и x'' - прообразы при линейном отображении A.

y' и y'' - образы x' и x'' при линейном отображении A.

Множество образов всех векторов пространства X при линейном отображении A называется образом линенйого оператора.

Ввиду линейности оператора его образ полностью определяется образами базисных векторов $(e_1, e_2, \dots, e_n)$ пространства X, поскольку любой вектор x пространства X можно представить в виде $x = x_1e_1 + x_2e_2 + \dots + x_ne_n$, откуда из линейности оператора следует $Ax = A(x_1e_1 + x_2e_2 + \dots + x_ne_n) = x_1Ae_1 + x_2Ae_2 + \dots + x_nAe_n$.

Раскладывая векторы Ae_k, $k = 1, 2, .., n$ по базису $f_1, f_2, \dots, f_m$ пространства Y, получим $Ae_k = a_{1k}f_1 + a_{2k}f_2 + \dots + a_{mk}f_m$, поэтому
$$Ax = x_1Ae_1 + \dots + x_nAe_n = x_1(a_{11}f_1 + a_{21}f_2 + \dots + a_{m1}f_m) + \dots + x_n(a_{1n}f_1 + a_{2n}f_2 + \dots + a_{mn}f_m).$$
Сгруппируем слагаемые по базисным векторам $f_1, f_2, \dots, f_m$:
$$Ax = (x_1a_{11} + \dots + x_na_{1n})f_1 + (x_1a_{21} + \dots + x_na_{2n})f_2 + \dots + (x_1a_{m1} + \dots + x_na_{mn})f_m$$
Таким образом, координаты $y_1, y_2, \dots, y_m$ образа Ax вектора x выражаются через координаты x_1, x_2,...+x_n вектора x в базисе $(e_1, e_2, \dots, e_n)$ пространства X и координаты a_{ij} образов базисных векторов $f_1, f_2, \dots, f_m$ пространства Y при линейном отображении A:
$$y_j = x_1a_{j1} + x_2a_{j2} + \dots + x_na_{jn}, \quad j = 1, \dots m,$$
в матричной форме это запишется как $y = Ax$, где A - матрица с элементами a_{ij} - i-я координата j-го базисного вектора e_j, т.е. матрица A состоит из векторов - столбцов образов базисных векторов.

Пример 1.

Построим матрицу оператора А, поворачивающего вектор на угол α против часовой стрелки. Такой оператор переводит вектор $(1, 0)$ в вектор $(\cos\alpha, \sin\alpha)$, $(0, 1)$ в $(-\sin\alpha, \cos\alpha)$, поэтому матрица A имеет вид
$$\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

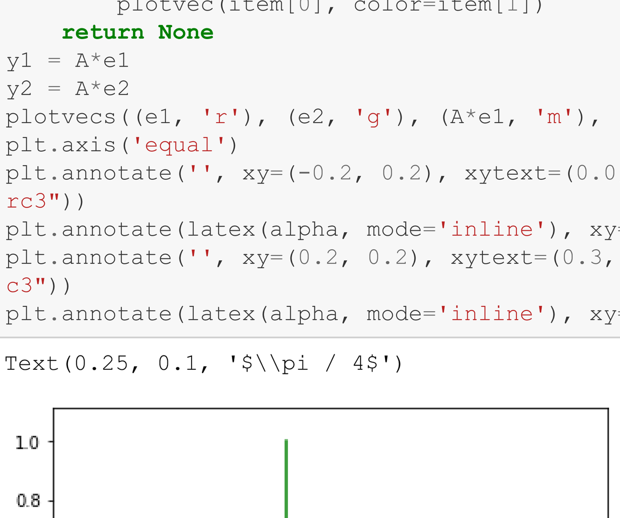
```
In [2]: alpha = Pi/4
A = Matrix([[Cos(alpha), -Sin(alpha)], [Sin(alpha), Cos(alpha)]])
e1 = Matrix([1, 0])
e2 = Matrix([0, 1])
display(A*e1, A*e2)
```

$$\begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

$$\begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

```
In [3]: def plotvec(vect, color='r'):
    plt.plot([0, vect[0]], [0, vect[1]], color)
    return None
def plotvecs(*vecscolors):
    for item in vecscolors:
        plotvec(item[0], color=item[1])
    return None
y1 = A*e1
y2 = A*e2
plotvecs((e1, 'r'), (e2, 'g'), (A*e1, 'm'), (A*e2, 'y'))
plt.axis('equal')
plt.annotate('', xy=(-0.2, 0.2), xytext=(0.0, 0.3), arrowprops=dict(arrowstyle="->", connectionstyle="arc3"))
plt.annotate(latex(alpha, mode='inline'), xy=(-0.2, 0.3), fontsize=16)
plt.annotate('', xy=(0.2, 0.2), xytext=(0.3, 0.0), arrowprops=dict(arrowstyle="->", connectionstyle="arc3"))
plt.annotate(latex(alpha, mode='inline'), xy=(0.25, 0.1), fontsize=16)
```

```
Out[3]: Text(0.25, 0.1, '$\\pi / 4$')
```



Проверим, что при умножении матрицы А на произвольный вектор плоскости получается вектор, образующий угол с исходным вектором. Воспользуемся возможностями модуля Geometry:

```
In [4]: from sympy.abc import x, y
O = Point(0, 0)
M = Point(x, y)
X = Matrix([x, y])
res = A*X
B = Point(res[0], res[1])
OM = Line(O, M)
OB = Line(O, B)
OM.angle_between(OB).expand().simplify()
```

Out[4]: $\frac{\pi}{4}$

Пример 2.

Построить матрицу оператора, переводящего базисные векторы i, j и k трехмерного пространства в векторы $(1, 2, 3)$, $(-2, 1, 4)$ и $(2, 0, 5)$. Матрица такого оператора
$$\begin{pmatrix} 1 & -2 & 2 \\ 2 & 1 & 0 \\ 3 & 4 & 5 \end{pmatrix}$$

```
In [5]: A = Matrix([[1, -2, 2], [2, 1, 0], [3, 4, 5]])
A
```

Out[5]: $$\begin{bmatrix} 1 & -2 & 2 \\ 2 & 1 & 0 \\ 3 & 4 & 5 \end{bmatrix}$$

Проверим, что при умножении базисных векторов на эту матрицу действительно получаются заданные векторы:

```
In [17]: i = Matrix([1, 0, 0])
j = Matrix([0, 1, 0])
k = Matrix([0, 0, 1])
display(A*i, A*j, A*k)
```

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 2 \\ 0 \\ 5 \end{bmatrix}$$

Найдем вектор, в который оператор переводит вектор
$$\begin{pmatrix} 3 \\ -7 \\ 9 \end{pmatrix}$$

```
In [7]: vec = Matrix([3, -7, 9])
display(A*vec)
```

$$\begin{bmatrix} 35 \\ -1 \\ 26 \end{bmatrix}$$

Мартица линейного оператора в новом базисе

Пусть в новом базисе у оператора А матрица A_{new}, тогда получить образ вектора x можно так: зная координаты x в новом базисе, находим его координаты в старом базисе с помощью преобразования, определяемого матрицей P, потом к полученному вектору в старом базисе применяем оператор А, умножая вектор в старом базисе на матрицу А в старом же базисе, после чего переводим полученный образ из старого базиса в новый умножением на матрицу P^{-1}. В итоге получается цепочка умножений $P^{-1}AP$, это произведение и есть матрица опреатора А в новом базисе, где P - матрица перехода от нового базиса к старому.

Тонкость такая: базисные векторы нового базиса в новом базисе будут иметь координаты $(1, 0, 0)$, $(0, 1, 0)$ и $(0, 0, 1)$, точно как базисные векторы срарого базиса в старом базисе, поэтому умножая вектор v с координатами в новом базисе на матрицу P мы получим координаты вектора v в старом базисе. Если требуется по координатам в старом базисе найти координаты в новом, нужно умножать на обратную матрицу, т.е. P^{-1}.

Мартица линейного оператора в новом базисе
$$A_{new} = P^{-1}AP \qquad (1)$$

Пример 3.

Найти матрицу оператора А примера 2 в базисе из векторов $(1, 2, 0)$, $(1, 0, 3)$ и $(0, 1, 5)$.

Вначале составим матрицу перехода, она будет такой же, как матрица некоторого оператора, переводящего текущий базис $(1, 0, 0)$, $(0, 1, 0)$ и $(0, 0, 1)$ в новый:
$$P = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 3 & 5 \end{pmatrix}$$
Затем найдем матрицу оператора в новом базисе по формуле (1)

```
In [8]: P = Matrix([[1,1,0],[2,0,1],[0,3,5]])
A_new = P**(-1)*A*P
display(P, A_new)
```

$$\begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 3 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ -3 & 6 & 8 \\ 4 & 0 & 1 \end{bmatrix}$$

Проследим за трансформацией новых базисных векторов f_1, f_2, f_3. Вначале найдем их образы при операторе А, умножая их на матрицу А в старом базисе, поскольку мы знаем координаты новых базисных векторов в старом базисе:

```
In [9]: f1 = Matrix((1,2,0))
f2 = Matrix((1,0,3))
f3 = Matrix((0,1,5))
display(A*f1, A*f2, A*f3)
```

$$\begin{bmatrix} -3 \\ 4 \\ 11 \end{bmatrix}$$

$$\begin{bmatrix} 7 \\ 2 \\ 18 \end{bmatrix}$$

$$\begin{bmatrix} 8 \\ 1 \\ 29 \end{bmatrix}$$

Мы получили координаты образов векторов нового базиса в старом базисе, теперь нужно найти координаты полученных векторов в новом базисе. Для этого нужно умножить векторы на матрицу, обратную к матрице P:

```
In [10]: display(P**(-1)*A*f1, P**(-1)*A*f2, P**(-1)*A*f3)
```

$$\begin{bmatrix} 0 \\ -3 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 6 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 8 \\ 1 \end{bmatrix}$$

Собственные числа и собственные векторы линейного оператора.

Пусть для некоторого ненулевого вектора x и числа λ выполняется
$$Ax = \lambda x, \quad x \neq 0,$$
тогда λ - собственное число линейного оператора A, а x - собственный вектор, соответствующий собственному числу λ.

Собственные числа и векторы не зависят от выбора базиса!

Находить собственные числа можно, решая характеристическое уравнение $|A - \lambda E| = 0$, где E - единичная матрица подходящей размерности.

Собственные векторы получаются как решения СЛАУ $(A - \lambda E)x = 0$, где λ - соответствующее собственное число.

В sympy есть средства вычисления собственных чисел и векторов

eigenvals (словарь, ключи - собственные числа, значения - их кратности)

eigenvect (список кортежей, первый элемент кортежа - собственное число, второй - его кратность, третий - список собственных векторов, соответствующих этому собственному числу)

Если у оператора в трехмерном пространстве есть три собственных вектора, то в базисе из этих векторов матрица оператора диагональная, на диагонали - собственные числа оператора.

Пример 3.

Дана матрица оператора В:
$$\begin{pmatrix} 51 & -8 & -12 \\ 9 & 33 & -18 \\ -18 & -6 & 66 \end{pmatrix}$$

Найти матрицу оператора В в базисе из собственных векторов.

Вначале найдем собственные векторы:

```
In [11]: B = Matrix([[51,-8,-12],[9,33,-18],[-18,-6,66]])
res = B.eigenvects()
for item in res:
    print(item[0], item[1])
    display(*item[2])
```

```
30 1
```

$$\begin{bmatrix} \frac{4}{3} \\ 2 \\ 1 \end{bmatrix}$$

```
45 1
```

$$\begin{bmatrix} \frac{4}{3} \\ -\frac{1}{2} \\ 1 \end{bmatrix}$$

```
75 1
```

$$\begin{bmatrix} -\frac{1}{3} \\ -\frac{1}{2} \\ 1 \end{bmatrix}$$

Составим матрицу перехода, она состоит из столбцов с координатами собственных векторов в старом базисе:

```
In [12]: P = res[0][2][0].row_join(res[1][2][0]).row_join(res[2][2][0])
display(P, P.det())
```

$$\begin{bmatrix} \frac{4}{3} & \frac{4}{3} & -\frac{1}{3} \\ 2 & -\frac{1}{2} & -\frac{1}{2} \\ 1 & 1 & 1 \end{bmatrix}$$

$$-\frac{25}{6}$$

Найдем матрицу в базисе из собственных векторов по формуле (1) перед примером 3:

```
In [13]: B_new = P**(-1)*B*P
B_new
```

Out[13]: $$\begin{bmatrix} 30 & 0 & 0 \\ 0 & 45 & 0 \\ 0 & 0 & 75 \end{bmatrix}$$

Поменяем в матрице Р последовательность векторов, поменяем местами второй и третий векторы:

```
In [14]: P_new = P[:,0].row_join(P[:,2]).row_join(P[:,1])
display(P_new)
B_new_1 = P_new**(-1)*B*P_new
B_new_1
```

$$\begin{bmatrix} \frac{4}{3} & -\frac{1}{3} & \frac{4}{3} \\ 2 & -\frac{1}{2} & -\frac{1}{2} \\ 1 & 1 & 1 \end{bmatrix}$$

Out[14]: $$\begin{bmatrix} 30 & 0 & 0 \\ 0 & 75 & 0 \\ 0 & 0 & 45 \end{bmatrix}$$

Заметим, что изменилась последовательность собственных чисел на главной диагонали, причем в соответствии с изменением порядка следования собственных векторов в матрице перехода.

Пример 4

Найдем собственные числа по определению, т.е. как корни характеристического многочлена $|B - \lambda E| = 0$

```
In [15]: lam = symbols('lamda')
lambdas = solve((B - lam*eye(3)).det())
lambdas
```

```
Out[15]: [30, 45, 75]
```

Для каждого собственного числа найдем собственный вектор, решив СЛАУ $(B - \lambda E)x = 0$:

```
In [16]: evects = set()
zero_vect = Matrix((0, 0, 0))
k1, k2, k3, k4 = symbols('k1:5')
display((B - lam*eye(3)).row_join(zero_vect))
for lam in lambdas:
    evects = sympy.Union(evects, linsolve((B - lam*eye(3)).row_join(zero_vect), k1, k2, k3))
display(*evects)

evects = [p.subs(symb, 1) for p in evects for symb in p.free_symbols ]
Matrix(evects).transpose()
```

$$\begin{bmatrix} 51-\lambda & -8 & -12 & 0 \\ 9 & 33-\lambda & -18 & 0 \\ -18 & -6 & 66-\lambda & 0 \end{bmatrix}$$

$$\left(-\frac{k_3}{3},\ -\frac{k_3}{2},\ k_3\right)$$

$$\left(\frac{4k_3}{3},\ -\frac{k_3}{2},\ k_3\right)$$

$$\left(\frac{4k_3}{3},\ 2k_3,\ k_3\right)$$

Out[16]: $$\begin{bmatrix} -\frac{1}{3} & \frac{4}{3} & \frac{4}{3} \\ -\frac{1}{2} & -\frac{1}{2} & 2 \\ 1 & 1 & 1 \end{bmatrix}$$