

Лабораторная работа №2

23 апреля 2021 г.

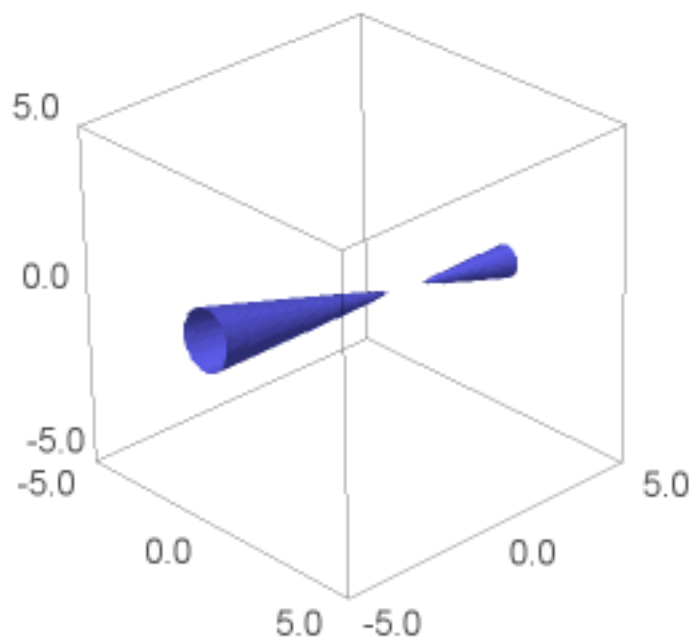
Задание 3. Привести уравнение к каноническому виду

Определяем переменные x , y , z и функцию $func$ моего варианта.

```
var("x y z")  
func = 11*x**2 - 2*x*y - 2*x*z + 2*y*z + 9*z**2 - 4*x + y + z
```

График данной функции:

```
plot_start = implicit_plot3d(func, (x,-5,5), (y,-5,5), (z,-5,5), plot_points=100)
```



Составляем матрицу квадратичной формы A , столбец коэффициентов линейной формы a и единичную матрицу E .

```
A = matrix([[11, -1, -1], [-1, 0, 1], [-1, 1, 9]])
a = matrix([[-2], [1/2], [1/2]])
E = matrix([[1, 0, 0], [0,1,0], [0,0,1]])
```

Находим корни характеристического уравнения.

```
var("l l1 l2 l3")
lvct = vector([l1, l2, l3])
lambdas = solve(det(A-l*E), l)
```

Находим взаимно перпендикулярные единичные собственные векторы $s1, s2, s3$, соответствующие корням $l1, l2, l3$.

```
svcts = []
for el in lambdas:
    nums = []
    lhs = (A-el.rhs()*E)*lvct
    res = solve([lhs[0] == 0, lhs[1] == 0, lhs[2] == 0], l1, l2, l3)[0]
    for i in range(len(res)):
        if len(res[i].rhs().variables()) == 0:
            nums.append(res[i].rhs())
        else:
            varname = res[i].rhs().variables()[0]
            d = {varname: 1}
            nums.append(res[i].rhs().subs(d).n())
    svcts.append(vector(nums))

for i in range(len(lambdas)):
    lambdas[i] = real_part(lambdas[i].rhs().n())
```

Нормируем эти векторы.

```
normvcts = []
for el in svcts:
    norm_lvct = (el / sqrt((el*el).n())).n()
    normvcts.append(norm_lvct)
```

Получаем матрицу S^T и получаем $a' = S^T * a$.

```
ST = matrix([[0, 0, 0],[0, 0, 0],[0, 0, 0]])
for i in range(len(normvcts)):
    for j in range(len(normvcts[i])):
        ST[i,j] = normvcts[i][j]

a_ = ST*a
show("a\': ", tuple(a_))
```

Так как $a' = 0$, то получаем канонический вид нашей функции.

```
kanon_func = lambdas[0]*x^2 + lambdas[1]*y^2 + lambdas[2]*z^2
show(kanon_func)
```

$$8.61752630079049 * x^2 - 0.180640477902440 * y^2 + 11.5631141771119 * z^2$$

График функции, приведенной к каноническому виду:

```
plot_end = implicit_plot3d(kanon_func, (x,-5,5), (y,-5,5), (z,-5,5), plot_points=100)
```

