

# Математические пакеты

## Введение в пакет SageMath

Сучков Андрей Игоревич

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

12 декабря 2020 г.

- SageMath (Sage, **SAGE** – **S**ystem for **A**lgebra and **G**eometry **E**xperimentation) – система компьютерной алгебры, покрывающая много областей математики, включая алгебру, комбинаторику, вычислительную математику и матанализ.
- Первая версия Sage была выпущена 24 февраля 2005 года в виде свободного программного обеспечения с лицензией GNU GPL.
- Первоначальной целью проекта было «создание открытого программного обеспечения альтернативного системам Magma, Maple, Mathematica, и MATLAB».
- Разработчиком Sage является Уильям Стейн – математик Вашингтонского университета.
- SageMath использует синтаксис Python

- ❶ Для создания достойной альтернативы системам Magma, Maple, Mathematica, и MATLAB потребуются сотни или тысячи человеко-лет, если начинать процесс разработки с нуля.
- ❷ Существует большое количество готового математического ПО с открытым исходным кодом, но написанного на различных языках программирования, из которых наиболее встречаемыми являются C, C++, Fortran и Python.

# SageMath

## Цели SageMath

- 1 Полезность
- 2 Эффективность
- 3 Свободный и открытый исходный код
- 4 Легкая компиляция
- 5 Взаимодействие
- 6 Хорошая документация
- 7 Расширяемость
- 8 Дружественность

- # – однострочные комментарии
- = – присваивание
- +, -, \*, / – простейшие арифметические операции
- ^, \*\* – возведение в степень
- % – взятие остатка
- // – целочисленное деление

### Листинг 1: Математические функции

```
1 sqrt(3.4) # return 1.84390889145858
2
3 sin(pi/3) # return 1/2*sqrt(3)
4
5 n(sin(pi/3)) # return 0.866025403784439
6 sin(pi/3).numerical_approx() # equivalent
7
8 N(sin(pi/3), digits=5) # return 0.86602
```

### Листинг 2: Определение функций

```
1 def is_even(n):  
2     return n % 2 == 0  
3  
4 ## Example  
5 is_even(2)    # return True  
6 is_even(3)    # return False  
7  
8 def is_divisible_by(number, divisor=2):  
9     return number % divisor == 0  
10  
11 ## Example  
12 is_divisible_by(6, 2)    # return True  
13 is_divisible_by(6)       # return True  
14 is_divisible_by(6, 5)    # return False
```

### Листинг 3: Циклы

```
1 def even(n):  
2     v = []  
3     for i in range(3,n):  
4         if i % 2 == 0:  
5             v.append(i)  
6     return v  
7  
8 ## Example  
9 even(10)    # return [4, 6, 8]
```



# Базовая алгебра и вычисления

Решение уравнений: функция `solve`

## Листинг 4: Точное решение уравнений

```
1 x = var('x')
2 solve(x^2 + 3*x + 2, x)
3 ## return [x == -2, x == -1]
4
5 x, b, c = var('x b c')
6 solve([x^2 + b*x + c == 0], x)
7 ## return [x == -1/2*b - 1/2*sqrt(b^2 - 4*c), x == -1/2*b +
   1/2*sqrt(b^2 - 4*c)]
```

# Базовая алгебра и вычисления

Решение уравнений: функция `find_root`

## Листинг 5: Численное решение уравнений

```
1 theta = var('theta')
2 solve(cos(theta) == sin(theta), theta)
3 ## return [sin(theta) == cos(theta)]
4
5 phi = var('phi')
6 find_root(cos(phi)==sin(phi), 0, pi/2)
7 ## return 0.785398163397448
```

# Базовая алгебра и вычисления

Дифференцирование, интегрирование и т.д.

## Листинг 6: Дифференцирование

```
1 u = var('u')
2 diff(sin(u), u)
3 ## return cos(u)
4
5 diff(sin(u^2), u, 4)
6 ## return 16*u^4*sin(u^2) - 48*u^2*cos(u^2) - 12*sin(u^2)
7
8 x, y = var('x, y')
9 f = x^2 + 17*y^2
10 f.diff(x) # return 2*x
11 f.diff(y) # return 34*y
```

# Базовая алгебра и вычисления

Дифференцирование, интегрирование и т.д.

## Листинг 7: Интегрирование

```
1 x = var('x')
2 integral(x*sin(x^2), x) # return -1/2*cos(x^2)
3
4 integral(x/(x^2+1), x, 0, 1) # return 1/2*log(2)
5
6 f = 1/((1+x)*(x-1))
7 f.partial_fraction(x) # return -1/2/(x + 1) + 1/2/(x - 1)
```

## Листинг 8: Разложение на простые дроби

```
1 x = var('x')
2
3 f = 1/((1+x)*(x-1))
4 f.partial_fraction(x) # return -1/2/(x + 1) + 1/2/(x - 1)
```

# Базовая алгебра и вычисления

## Решение дифференциальных уравнений

### Листинг 9: Функция `desolve`

```
1 x = var('x')
2 y = function('y')(x)
3
4 DE = diff(y, x) + y - 1
5 desolve(DE, [y, x])
6 ## return (_C + e^x)*e^(-x)
```

# Базовая алгебра и вычисления

## Решение дифференциальных уравнений

### Листинг 10: Преобразование Лапласа

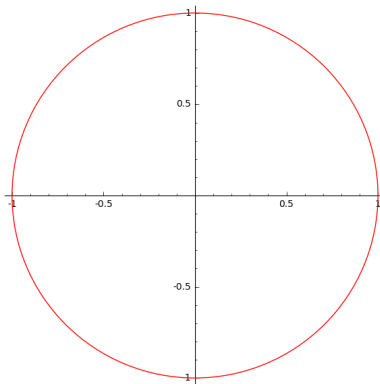
```
1 s = var("s")
2 t = var("t")
3
4 f = t^2*exp(t) - sin(t)
5
6 f.laplace(t, s)
7 ## return -1/(s^2 + 1) + 2/(s - 1)^3
```

# Построение графиков

## Двумерные графики

### Листинг 11: Построение окружности

```
1 circle((0,0), 1, rgbcolor=(1,0,0))
```

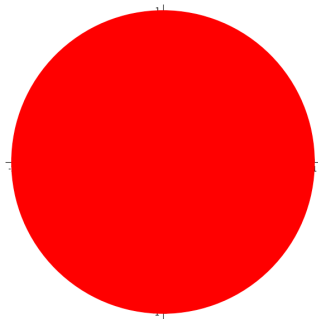


# Построение графиков

## Двумерные графики

### Листинг 12: Построение круга

```
1 circle((0,0), 1, rgbcolor=(1,0,0), fill=True)
2 ## equivalent
3 c = circle((0,0), 1, rgbcolor=(1,0,0), fill=True)
4 c.show() # equivalent: show(c)
```





# Построение графиков

## Двумерные графики

### Листинг 13: Объединение графиков

```
1 p1 = plot(cos, (-2*pi, 2*pi), rgbcolor=(1,0,0))  
2 p2 = plot(sin, (-2*pi, 2*pi), rgbcolor=(0,1,0))  
3 show(p1 + p2)
```

