

Инструментарий на основе Python-библиотек и экосистемы Jupyter для решения научных и прикладных задач

Башагин М.В., Бежанян Т.Ж., Бутенко Ю.А., Воронцов А.С.,
Зуев М.И., Киракосян М.Х., Матвеев М.А., Нечаевский А.В.,
Пряхина Д.И., Рахронов И.Р., Рахронова А.Р., Стрельцова О.И.

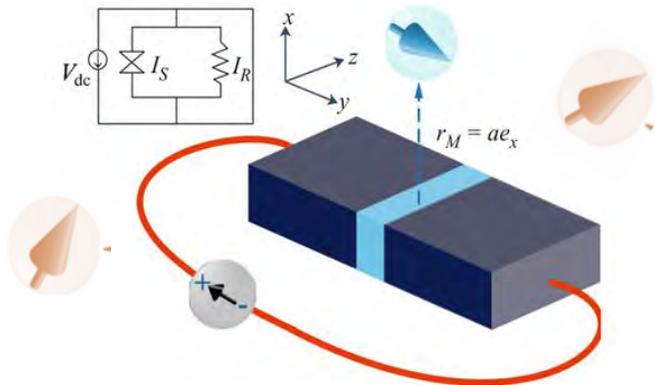
ЛИТ, ЛТФ ОИЯИ

Осенняя Школа по информационным технологиям ОИЯИ

16-20 октября 2023



Блок символьных вычислений



$$\gamma_{m_i} = -\frac{\mu_0}{2\Phi_0} \int d\mathbf{r}_i \frac{\mathbf{M}_i \times \mathbf{r}_i}{r^3}$$

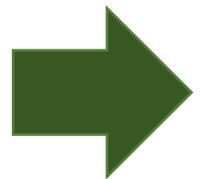
$$B_{12}(r_{12}, m_1) = \frac{\mu_0}{4\pi} \left(\frac{3(m_1 \cdot \hat{r})\hat{r}}{b^5} - \frac{m_1}{b^3} \right)$$



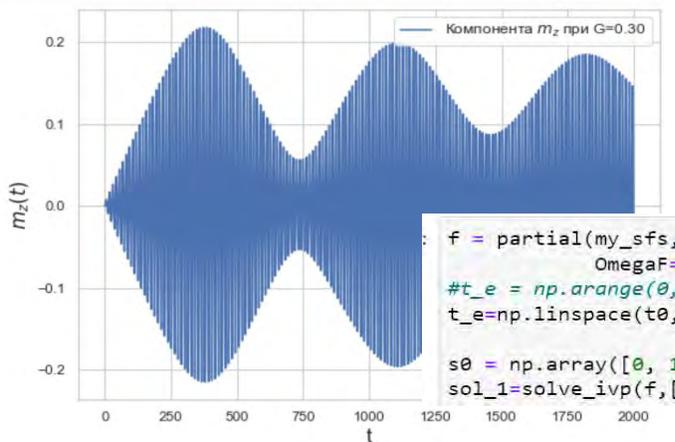
SymPy is a Python library for symbolic mathematics.



matplotlib is a main library for building graphs, diagrams in Python.



Блок численных расчетов и анализа



```
f = partial(my_sfs, G=G, alpha=alpha, k=k, \
            OmegaF=OmegaF, V=V)
#t_e = np.arange(0, 25, 0.0001)
t_e=np.linspace(t0,tf,100000)

s0 = np.array([0, 1, 0])
sol_1=solve_ivp(f,[t0,tf],s0, t_eval=t_e, method='RK45')
```



SciPy is an open-source software for mathematics, science, and engineering.

Ускорение многопараметрических расчетов



Joblib is a set of tools to provide lightweight pipelining in Python



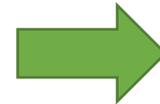
Numba is an open source JIT compiler that translates a subset of Python and NumPy code into fast machine code.

Процесс проведения численных исследований

**Математическая
постановка
задачи**



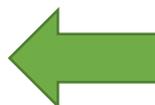
**Разработка
вычислительной
схемы**



**Подбор библиотек,
апробация на модельных
расчетах, визуализация**



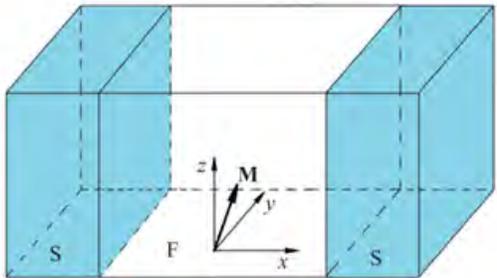
**Анализ
результатов,
построение
графиков,
диаграмм....**



**Проведение
расчетов на
вычислительных
платформах**



**Программная
реализация**



Периодичность появления интервалов обращения магнитного момента джозефсоновского φ_0 -перехода

Математическая постановка задачи

Основные уравнения представлены в работе [1]. Ниже приведена задача Коши для системы уравнений в безразмерном виде. Динамика магнитного момента M рассматриваемой системы описывается уравнением Ландау-Лифшица-Гильберта:

$$\frac{dm_x}{dt} = -\frac{1}{1 + M^2 \alpha^2} \{m_y H_z - m_z H_y + \alpha[m_x(M, H) - H_x]\},$$

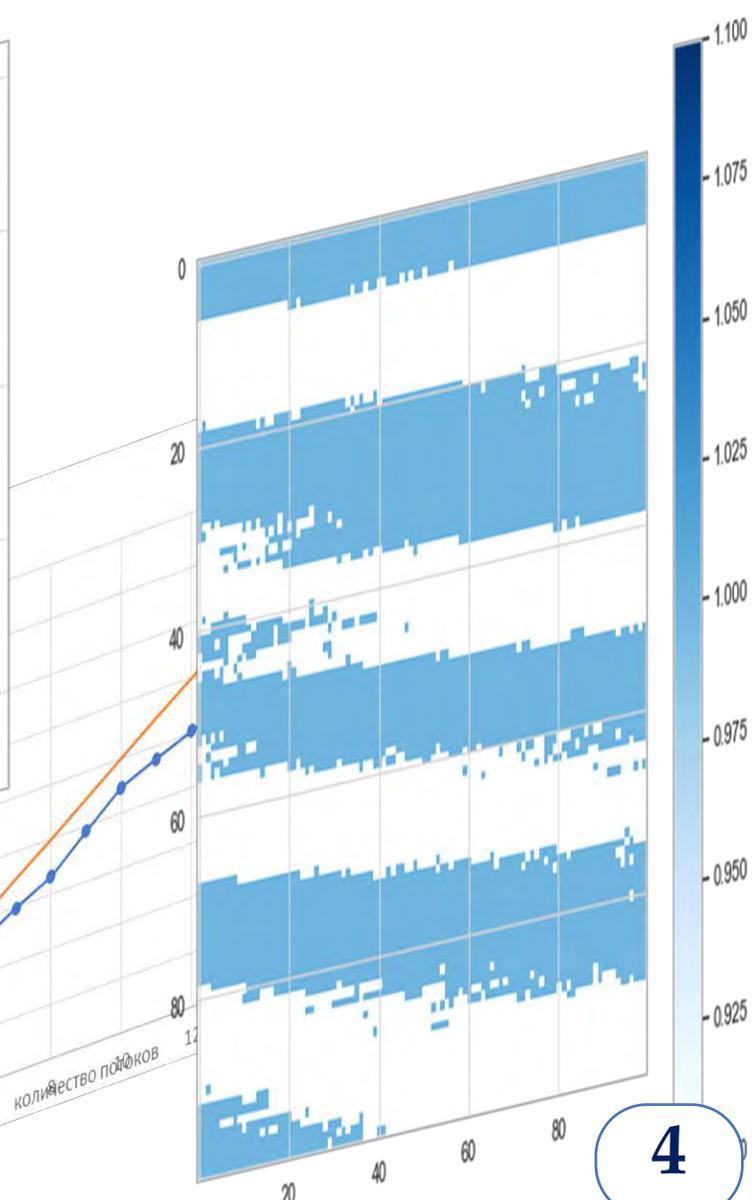
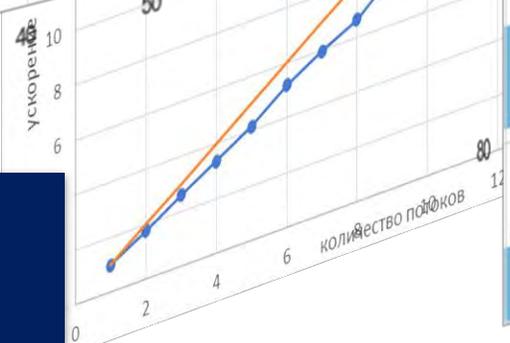
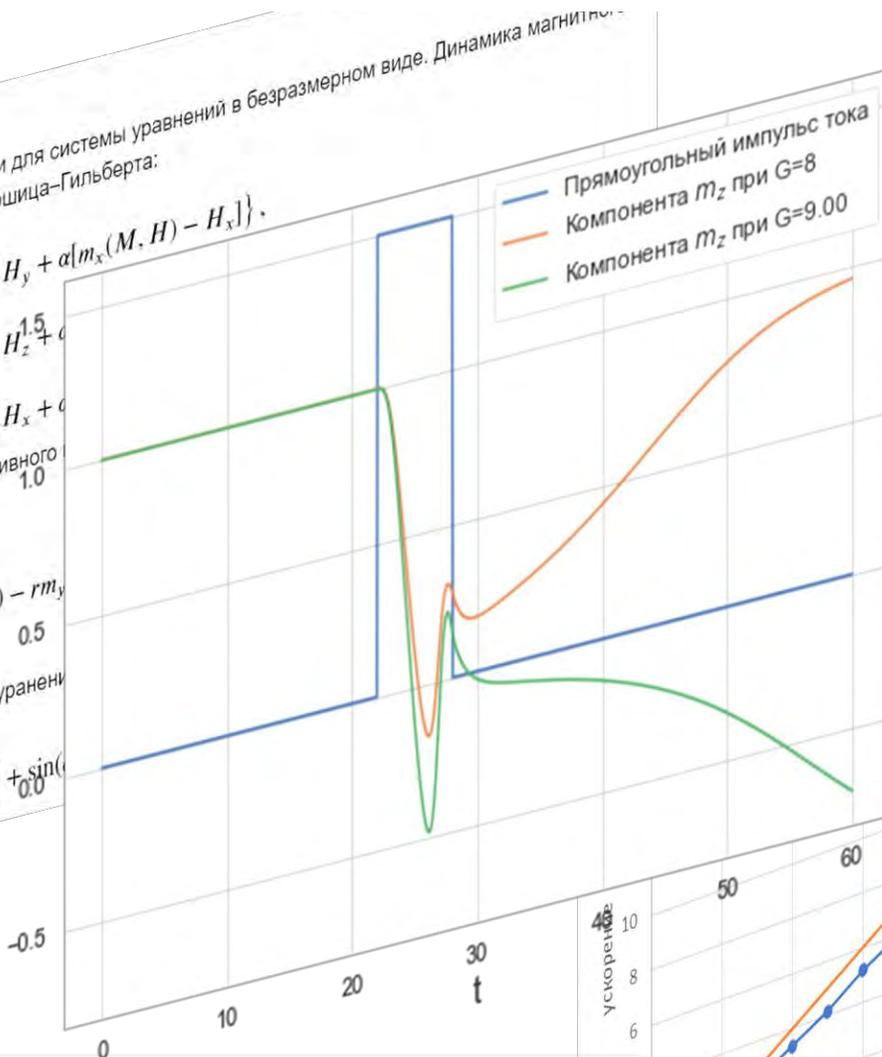
$$\frac{dm_y}{dt} = -\frac{1}{1 + M^2 \alpha^2} \{m_z H_x - m_x H_z + \alpha[m_y(M, H) - H_y]\},$$

$$\frac{dm_z}{dt} = -\frac{1}{1 + M^2 \alpha^2} \{m_x H_y - m_y H_x + \alpha[m_z(M, H) - H_z]\},$$

где $M = [m_x, m_y, m_z]$ - компоненты магнитного момента, компоненты эффективного магнитного поля $H_x(t) = 0$, $H_y = Gr \sin(\phi(t) - r m_y)$, $H_z(t) = m_z(t)$ определяются следующим образом:

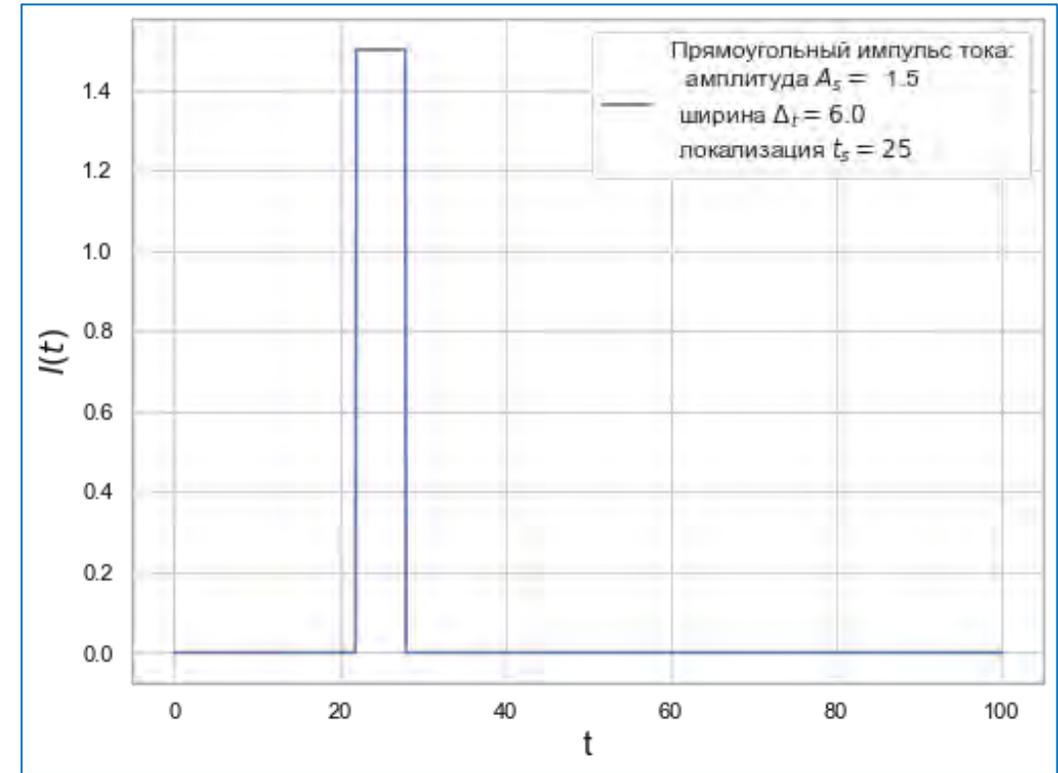
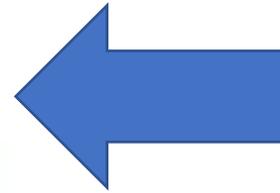
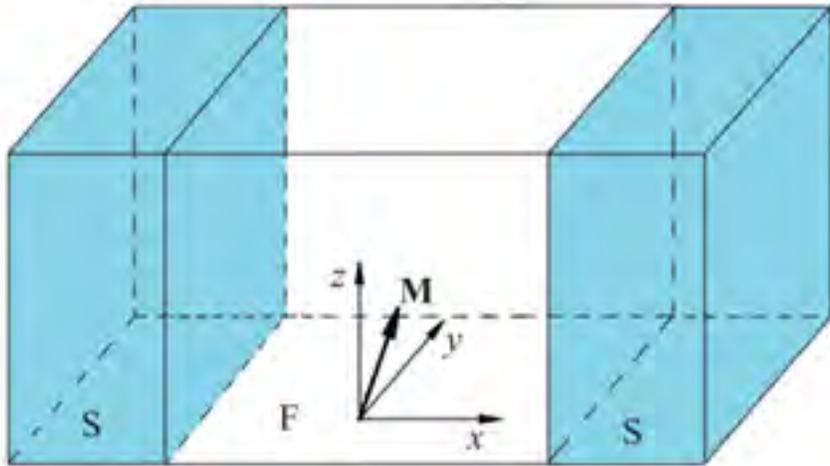
$$I = w \left(\frac{d\phi}{dt} - r \frac{dm_y}{dt} \right) + \sin(\phi)$$

при этом уравнение на джозефсоновскую разность фаз $\phi(t)$ определяется уравнением джозефсоновского контакта, измеренный в единицах критического тока I_c :



Периодичность появления интервалов обращения магнитного момента джозефсоновского φ_0 -перехода

Физическая постановка задачи



Исследовать временные зависимости компонент магнитного момента M при различных значениях параметров φ_0 -перехода, на основе которых можно установить интервалы параметров, где происходит его переверот от **1** к **-1**.

Математическая постановка задачи



Динамика магнитного момента рассматриваемой системы описывается уравнением Ландау-Лифшица-Гильберта:

$$\frac{dm_x}{dt} = -\frac{1}{1+M^2\alpha^2} \left\{ m_y H_z - m_z H_y + \alpha \left[m_x (M, H) - H_x \right] \right\},$$

$$\frac{dm_y}{dt} = -\frac{1}{1+M^2\alpha^2} \left\{ m_z H_x - m_x H_z + \alpha \left[m_y (M, H) - H_y \right] \right\},$$

$$\frac{dm_z}{dt} = -\frac{1}{1+M^2\alpha^2} \left\{ m_x H_y - m_y H_x + \alpha \left[m_z (M, H) - H_z \right] \right\},$$

$$\frac{d\varphi}{dt} = -\frac{1}{w} \left(\sin(\varphi - rm_y) + r \frac{dm_y}{dt} \right) + \frac{1}{w} I,$$

Начальные условия:

$$m_x(0) = 0, \quad m_y(0) = 0, \quad m_z(0) = \mathbf{1}, \quad \varphi(0) = 0$$

$M = [m_x, m_y, m_z]$ – компоненты магнитного момента, компоненты эффективного поля:

$$H_x(t) = 0,$$

$$H_y(t) = Gr \sin(\varphi(t) - rm_y(t)),$$

$$H_z(t) = m_z(t),$$

параметрами модели:

G – отношение энергии Джозефсона к энергии магнитной анизотропии,

r – константа спин-орбитального взаимодействия,

α – диссипация Гилберта,

$w = \mathbf{1}$ в рамках этого исследования.

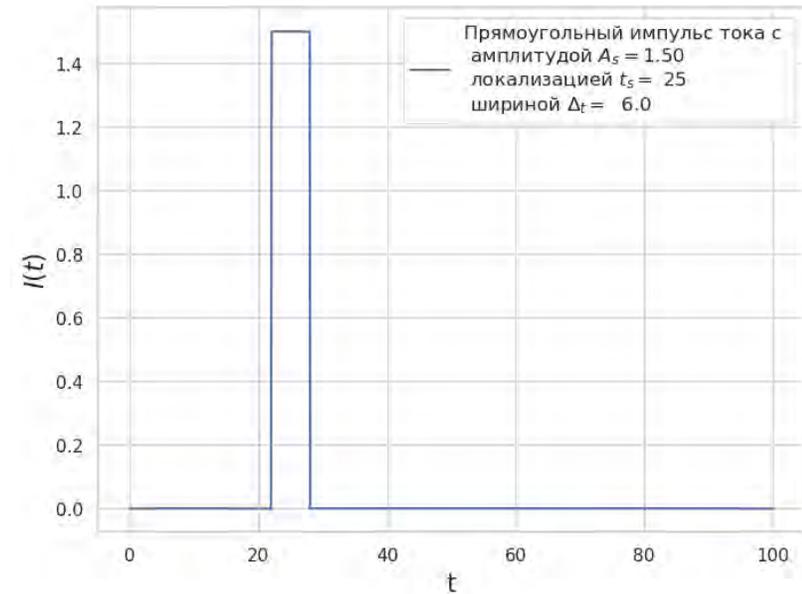
Разработка вычислительной схемы

1. Численное решение начальной задачи

$$\begin{cases} \frac{dS}{dt} = F(t, S, p), \\ S|_{t=0} = S_0. \end{cases}$$

при фиксированных значениях параметров проведение расчетов для прямоугольного импульса тока:

$$I = I(t) = \begin{cases} A_s, & t \in [t_s - \Delta t / 2, t_s + \Delta t / 2], \\ 0, & \text{otherwise.} \end{cases}$$



2. Для проведения исследований зависимости решений от параметров требуется разработка параллельной реализации программы.

Подбор библиотек, апробация на модельных расчетах, визуализация

1. Численное решение начальной задачи

$$\begin{cases} \frac{dS}{dt} = F(t, S, p), \\ S|_{t=0} = S_0. \end{cases}$$

при фиксированных значениях параметров проведение расчетов для прямоугольного импульса тока:

$$I = I(t) = \begin{cases} A_s, & t \in [t_s - \Delta t / 2, t_s + \Delta t / 2], \\ 0, & \text{otherwise.} \end{cases}$$

2. Для проведения исследований зависимости решений от параметров требуется разработка параллельной реализации программы.



SciPy is an open-source software for mathematics, science, and engineering.

Ускорение многопараметрических расчетов



Joblib is a set of tools to provide lightweight pipelining in Python



Numba is an open source JIT compiler that translates a subset of Python and NumPy code into fast machine code.



HPCLab component
VM with JupyterHub and SLURM [<https://jlabhpc.jinr.ru>]

- ❑ Intel Xeon Gold 6126 (24 Cores @ 2.6 GHz)
- ❑ 32 GB RAM

Educational component
JupyterLab Server [<https://studhub.jinr.ru>]
[<https://studhub2.jinr.ru>]

- ❑ 2x Intel Xeon Gold 6152 (22 Cores @ 2.1 GHz)
- ❑ 512 GB RAM

Computation component
Server with NVIDIA Volta [<https://jhub1.jinr.ru>]
[<https://jhub2.jinr.ru>]

- ❑ 2x Intel Xeon Gold 6148 (20 Cores @ 2.4 GHz)
- ❑ 4x **NVIDIA Tesla V100** SXM2 32 GB HBM2
- ❑ 512 GB RAM

[<https://jhub3.jinr.ru>]

- ❑ 2x Intel Xeon E5 2698v4 (20 Cores @ 2.2 GHz)
- ❑ 8x **NVIDIA Tesla V100** SXM2 16 GB HBM2
- ❑ 512 GB RAM

Экосистема для практических занятий



Jupyter Notebook

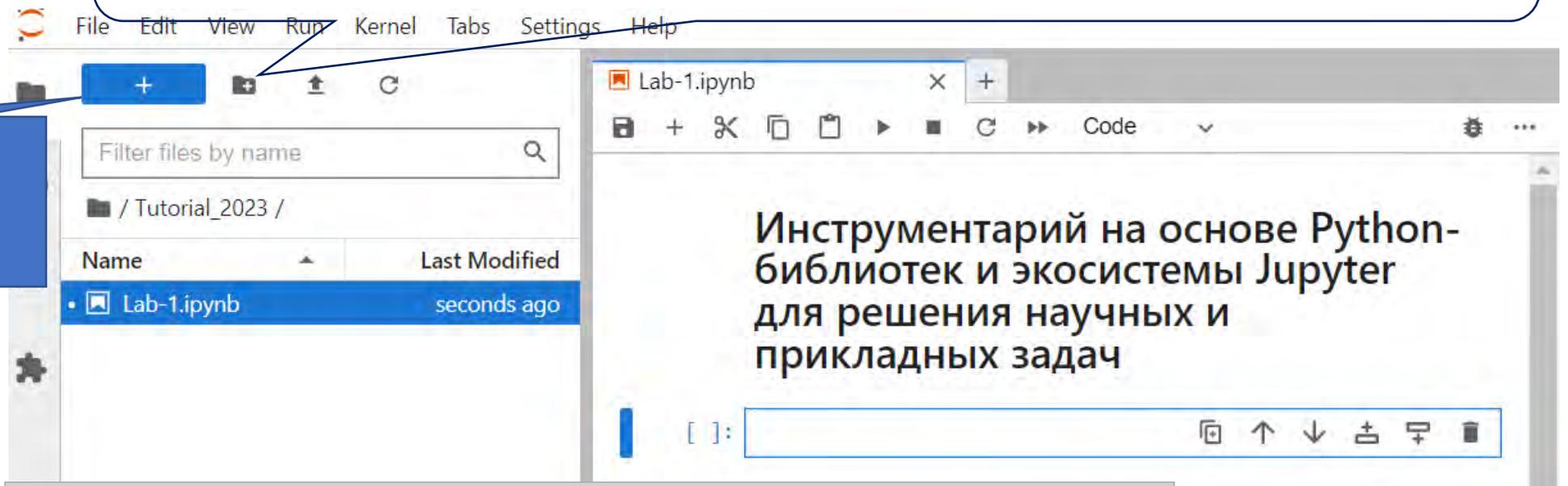
представляет собой интерактивную среду для запуска программного кода в браузере.

VM <https://studhub.jinr.ru>

(авторизация через учетную запись GitLab)
установлен JupyterLab

1. Создайте папку для практических занятий, для этого нажмите на **+** в левой части экрана.

2. Создать новый файл Python 3



Login: tut001- tut100



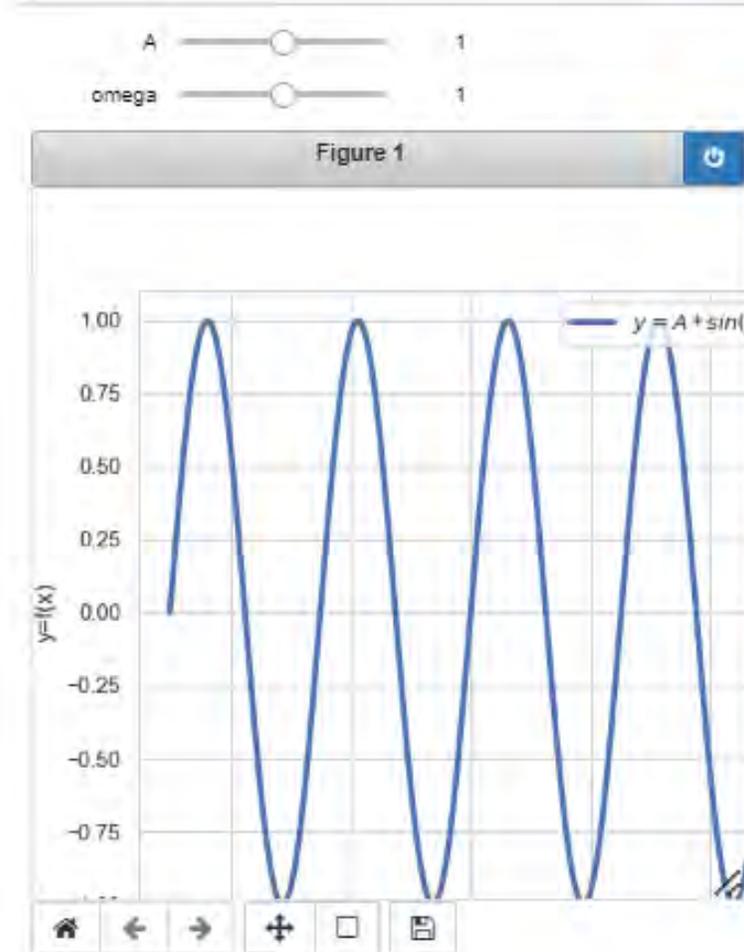
Matplotlib + Jupyter Widgets

```
%matplotlib widget
```

```
@interact
def show_sin(A=1,omega=1):
    t=np.linspace(-4*np.pi, 4*np.pi, 150 , endpoint=True)
    fig = plt.figure(figsize=(6,6))
    # plot sin
    plt.plot(t, f_sin(t,A,omega),label='$y=A*\sin(\omega_t)$ ',linewidth=3.0)
    plt.xlabel('t ')
    plt.ylabel('y=f(x)')

    plt.legend(loc='upper right')

    plt.show()
```



Численное решение задачи Коши: библиотека SciPy

Задача Коши: Рассмотрим решение начальной задачи (*Initial value problem*) для системы обыкновенных дифференциальных уравнений первого порядка, разрешенных относительно производной:

$$\begin{cases} \frac{dy(t)}{dt} = f(t, y(t)), \\ y|_{t=t_0} = y_0, \end{cases} \quad (1)$$

где $y = (y_1, \dots, y_n)^T$ - вектор-функция.

Пример 1: Численно решить задачу Коши:

$$\begin{cases} \frac{dy}{dt} = y \cos(t), \\ y(0) = y_0. \end{cases} \quad (2)$$

Для сравнения приведем аналитическое решение задачи (2):

$$y_{exact} = y_0 e^{\sin(t)}.$$

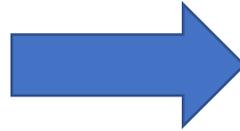
[Библиотека SciPy](#)



Решение задачи Коши для ОДУ

Задача 1. Численно решить задачу Коши:

$$\begin{cases} \frac{dy(t)}{dt} = y \cos(t), \\ y|_{t=0} = y_0. \end{cases}$$



Задача 2. Численно решить задачу Коши:

$$\begin{cases} \frac{dy(t)}{dt} = y \cos(\omega t), \\ y|_{t=0} = y_0. \end{cases}$$

Задача 3. Линеаризованное уравнение на магнитный момент:

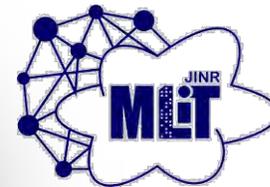
$$\frac{d^2 m_y}{dt^2} + 2\alpha\omega_J \frac{dm_y}{dt} + \omega_F^2 m_y = \omega_F^2 G r \sin(\omega_J t)$$



$$m_y = y_1$$

Система ОДУ:

$$\begin{cases} \frac{dy_1}{dt} = y_2, \\ \frac{dy_2}{dt} = -2\alpha\omega_J y_2 - \omega_F^2 y_1 + \omega_F^2 G r \sin(\omega_J t) \\ y_1|_{t=0} = y_{10} \\ y_2|_{t=0} = y_{20} \end{cases}$$



Инструментарий на основе Python-библиотек и экосистемы Jupyter для решения научных и прикладных задач



jupyter {book}

<http://studhub.jinr.ru:8080/books>