# Уравнение Матьё

# Émile Léonard Mathieu

**Эмиль Леонард Матьё** (15 мая 1835, 19 октября 1890) — французский математик и астроном. Наиболее известны его работы по теории групп и математической физике.

Наиболее важные достижения Матьё — функция Матьё, группа Матьё, преобразование Матьё. Он автор учебника по математической физике в 6 томах

# Уравнение Матьё

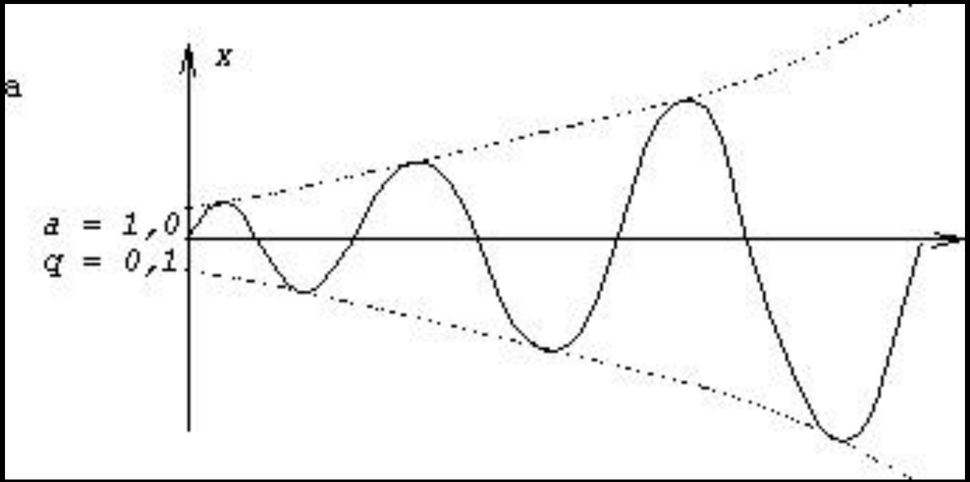
$$\frac{d^2y}{dx^2} + (\mathrm{a} + 16\mathrm{q} \cdot \cos(2\mathrm{x}))\mathrm{y} = 0$$

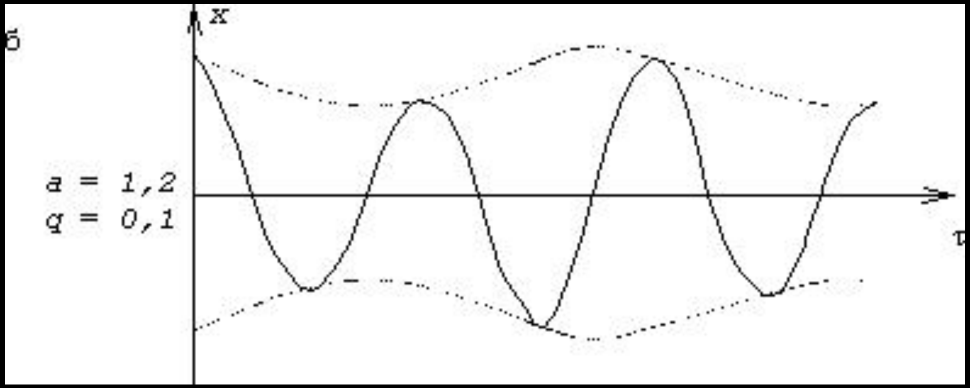
Классическое уравнение теории нелинейных колебаний, выведено в 1868 году в ходе его исследований, связанных с колебаниями эллиптической мембраны барабана.

Эти функции имеют широкий спектр применений в различных областях науки и техники. Например, функции Матьё используются в теории дифференциальных уравнений, физике колебаний, теории волн, электродинамике и оптике. Они представляют собой мощный инструмент для описания и анализа различных физических явлений.

# Вид уравнения Матьё

a) Амплитуды возрастают. Система не устойчива
b) Амплитуды ограничены. Система устойчива



# Задача курсовой $\frac{d^2y}{dx^2} + (\mathrm{a} + 16\mathrm{q} \cdot \cos(2\mathrm{x}))\mathrm{y} = 0$

$$\begin{cases} \frac{d^2U}{dt^2} + (\delta + E \cdot \cos(2\mathrm{t}))\mathrm{U} = 0 \\ \mathrm{U}(0) = A \\ \mathrm{U}'(0) = \mathrm{B} \end{cases}$$

$A = 0.5300355 \cdot x^*$, где $x^*$ – наименьший корень уравнения $x = 1.4^x$

B = 0

δ = 1

$$E = 1.553791 \int_0^1 \frac{\sin(x)}{x^2+1} dx$$

Построить график U(t), оценить погрешность результата и влияние на точность погрешности исходных данных.

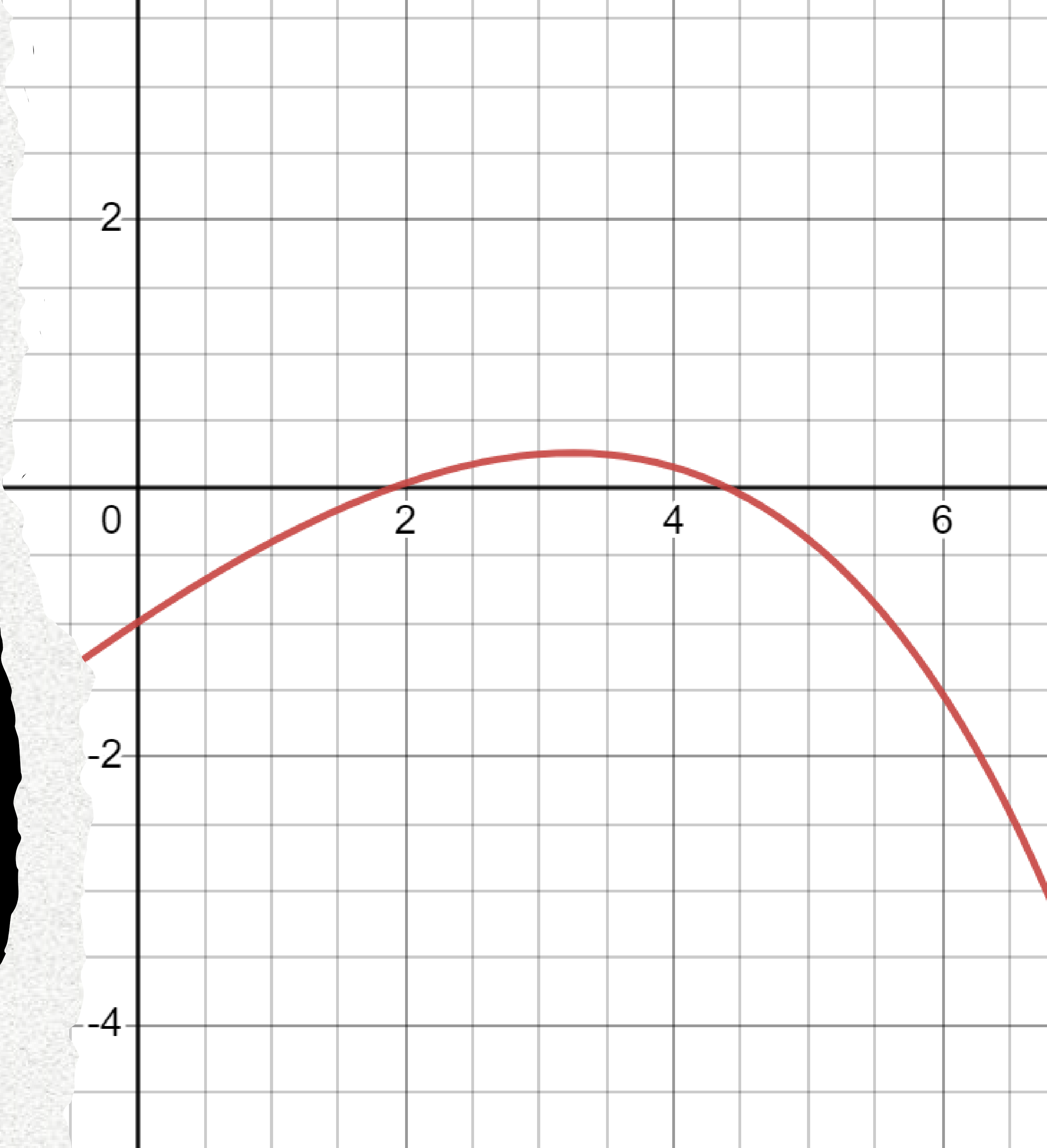
# Разделение на подзадачи

- Найти коэффициент A ($\text{наименьший корень уравнения } x = 1.4^x$).
- Найти коэффициент E ($1.553791 \int_0^1 \frac{\sin(x)}{x^2+1} dx$).
- Привести дифференциальное уравнение второго порядка к системе дифференциальных уравнений первого порядка.
- Решить систему.

# Поиск коэффициента A

$$x = 1.4^x$$
$$x - 1.4^x = 0$$

```
def get_x():
    """
    Возвращает результат решения уравнения x=1.4^x
    """
    x = 2
    eps = 1e-9

    def F(x):
        return x - (1.4 ** x)

    return scipy.optimize.newton(F, x, tol=eps)
```



# Поиск коэффициента E

$$1.553791 \int_0^1 \frac{\sin(x)}{x^2+1} dx$$

$$\int \frac{\sin(x)}{x^2+1} \, dx$$

Недостаточно возможностей для нахождения первообразной данного интеграла

либо решения в элементарных функциях не существует

```
def get_E():
    """
    Возвращает переменную E, взяв интеграл и умножив на константу
    """
    start = 0
    end = 1

    def F(x):
        return np.sin(x) / (x ** 2 + 1)

    return 1.553791 * scipy.integrate.quad(F, start, end)[0]
```

# Система дифференциальных уравнений первого порядка

$$\frac{d^2U}{dt^2} + 0\frac{dU}{dt} + (\delta + E\cdot\cos(2\mathrm{t}))\mathrm{U} = 0$$

$$\frac{d^2U}{dt^2} + \alpha_1\frac{dU}{dt} + \alpha_2\mathrm{U} = 0$$

$$\begin{cases} U^{(2)'} = U^{(1)} \\ U^{(1)'} = -\alpha_1 U^{(1)} - \alpha_2 U^{(2)} \end{cases}$$

$$\begin{cases} U^{(2)'} = U^{(1)} \\ U^{(1)'} = -(\delta + E\cdot\cos(2\mathrm{t}))U^{(2)} \end{cases}$$

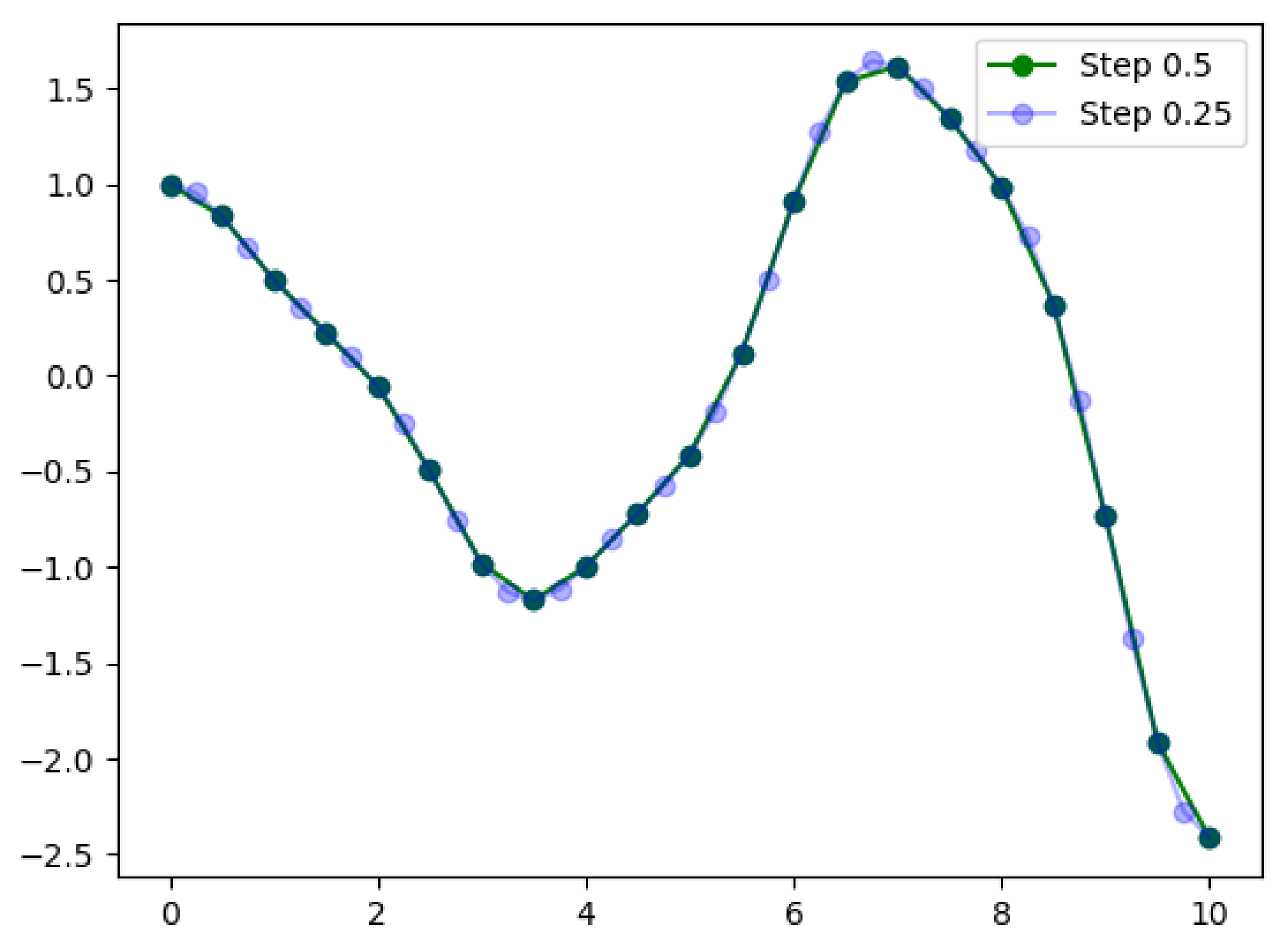
```python
def get_Mathieu_function(delt, E):
    """
    Возвращает функцию Метьё
    """

    def F(t, X):
        dX = np.zeros(X.shape)
        dX[1] = X[0]
        dX[0] = -(delt + E * np.cos(2 * t)) * X[1]
        return dX

    return F
```

# Решение систему дифференциальных уравнений первого порядка

```python
def rkf853(f, T, X0):
    """
    Решает `x' = f(t, x)` для каждого `t` в `T`
    С начальным значением `X0`, используя явный метод Рунге-Кутта 8
    """
    runge = scipy.integrate.ode(f).set_integrator('dop853').set_initial_value(X0, T[0])
    X = [X0, *[runge.integrate(T[i]) for i in range(1, len(T))]]
    return np.array([i[0] for i in X]), np.array([i[1] for i in X])
```

# Результат

Графики U(t) с шагом 0.5 и 0.25



# Как оценить погрешность?

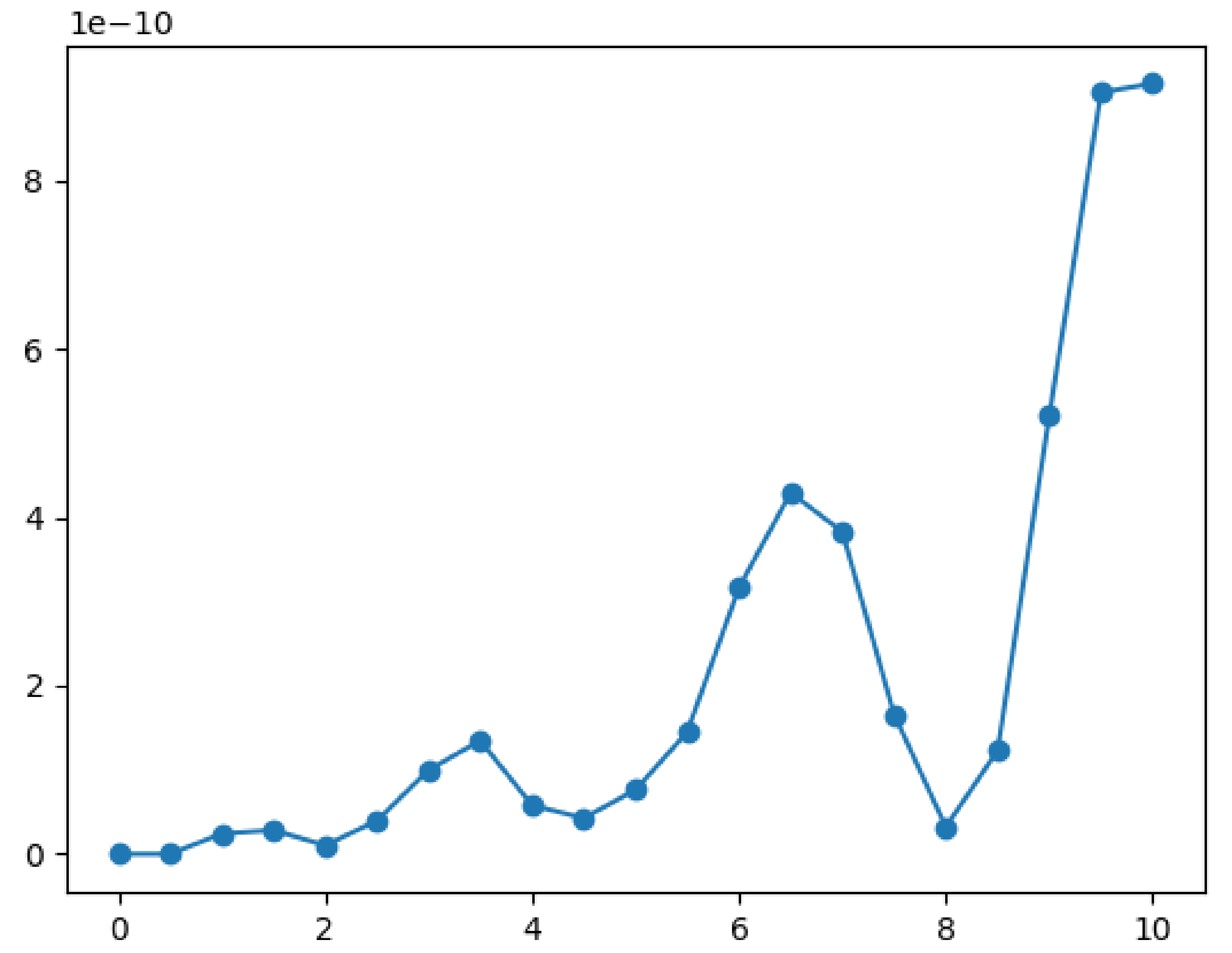
В программе использовалось много различных функций, оценить погрешность который в сумме очень сложно.

# Правило Рунге

**Правило Рунге** — правило оценки погрешности численных методов, было предложено К. Рунге в начале 20 века.

$$\frac{\left|y_{i,h} - y_{i,h/_2}\right|}{2^p - 1}$$

p – степень используемого метода.
В нашем случае p = 8



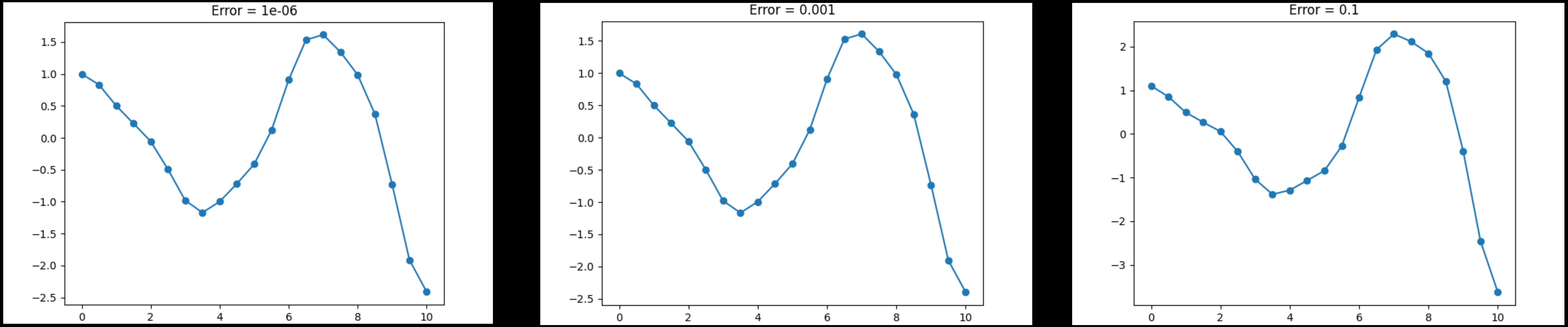

# График погрешности по правилу Рунге

Max error = 9.1503e-10

# Анализ устойчивости системы

А насколько сильно на систему влияет погрешность исходных данных?

Для этой оценки искусственно добавим погрешность в N-й знак каждого исходного значения




# Графики погрешностей

| Error val | Max Error |
|---|---|
| 1e-06 | 2.11771e-06 |
| 1e-05 | 1.03083e-04 |
| 1e-04 | 2.11722e-04 |
| 1e-03 | 1.02919e-02 |
| 1e-02 | 1.10085e-01 |
| 1e-01 | 1.21292e+00 |

# Questions?

# Спасибо за внимание