Постановка задачи

Рассмотрим математический маятник длинной L и массой m в поле тяжести g. Тогда уравнение сохранения энергии запишется следующим образом:

$$\frac{mL^2\dot{\alpha}^2}{2} + mgL(1-\cos\alpha) = E_0 = const$$

после дифференцирования по времени получим, де-факто, уравнения Ньютона:

$$mL^2\ddot{\alpha} + mgL\sin\alpha = 0$$

Добавим в нашу систему трение:

$$mL^2\ddot{\alpha} + mgL\sin\alpha + \beta L\dot{\alpha} = 0$$

Введем $\delta = \frac{\beta}{2m}$ и $\omega_0^2 = \frac{g}{L}$ и получим нужное уравнение 2-го порядка:

$$\ddot{\alpha} + 2\delta\dot{\alpha} + \omega_0^2\sin\alpha = 0$$

И последний штрих: добавим в систему так называемое «событие» (оно же «event»): $\dot{\alpha} \to \dot{\alpha} + 0.4\sqrt{\frac{2*E_0}{mL^2}}\mathrm{sign}(\dot{\alpha})$, когда энергия системы падает ниже половины начальной. Сделано это не из физических соображений, а демонстрации ради. Перепишем уравнения в виде ODE:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} \omega \\ -2\delta\omega - \omega_0^2\sin\alpha \end{pmatrix}$$

Описание программы

Итак, теперь перейдем к самой программе. Она состоит из двух частей:

1. Модуль drawing отвечает за рисование четырех анимаций: реального движения маятника и трех различных фазовых диаграмм. Функция draw(ode_sol, t, energy, m, L) принимает на вход пять аргументов, где ode_sol должна содержать массив (n, 2) решений дифференциального уравнения (например, непосредственный вывод solve_ivp или ode_int), t - массив отметок во времени (к которым относятся точки из ode_sol), energy - массив размера (n,) предпосчитанных энергия состояний и два параметра системы m и L.
2. Основной модуль отвечает за решение вышеуказанных уравнений методом solve_ivp, применением «событий» и склейку всего процесса в одно решение.

О методе Рунге - Кутты

Метод этот основан на использовании разложения в ряд Тейлора:

$$f(x,y) = f(x_0,y_0) + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + ...$$

А именно $y_{n+1} = y_n + \frac{h}{6}(p_1 + 2p_2 + 2p_3 + p_4)$, где $p_1 = f(x_n, y_n)$, $p_2 = f(x_n + \frac{h}{2}, y_n + \frac{hp_1}{2})$, $p_3 = f(x_n + \frac{h}{2}, y_n + \frac{hp_2}{2})$ и $p_4 = f(x_n + h, y_n + hp_3)$. Утверждается (и проверять мы не станем, долго и больно), что такой метод дает точность порядка $O(h^5)$, где h - шаг.

Слово о включении «событий» в RHS

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} \omega \\ -2\delta\omega - \omega_0^2\sin\alpha + C\delta(E - \frac{E_0}{2}) \end{pmatrix}$$

Где C - некий коэффициент равный $0.4\sqrt{\frac{2*E_0}{mL^2}}\mathrm{sign}(\dot{\alpha})$ умноженному на нормализацию от дельта-функции, который можно посчитать при большом желании. Как запихнуть дельта-функцию в решатель ODE? Пробуем в третьем модуле test на тестовом примере - и ничего не получим.