DIY Project report: Adaptive step-size RK4 on double pendulum

Adhilsha A - 2011006

Project report for P346, School of Physical sciences, NISER

(Dated: December 1, 2022)

In this project, I will be coding an adaptive step-size Runge Kutta 4^{th} order (ASRK4) in python language and applying it to a specific case of Double pendulum. The results show the merit of using ASRK4 instead of fixed step-size Runge Kutta 4^{th} order (RK4).

I. INTRODUCTION

We have already studied and coded the fixed step-size step-size Runge-Kutta 4^{th} order (RK4) in the class itself. Here, we will be re-creating an adaptive step-size step-size Runge-Kutta 4^{th} order (ASRK4) which alters the step-size depending on the related error that can occur if the step is used. The goal is to achieve a certain precision/tolerance level without wasting many steps. The main obstacles for the recoding are:

- to form a notion of possible error related to step-size
- setting limits for reduction and expansion of step-size

This will be applied in the case of a Double pendulum (2 dimensions) which is also known as Chaos pendulum. It is a pendulum with another pendulum attached to the end, forming a simple physical system that is extremely sensitive to initial conditions. We will be using ASRK4 to tackle this problem more accurately.

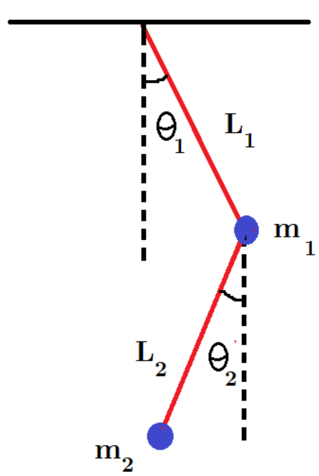

FIG. 1. Double pendulum setup

II. PROBLEM: DOUBLE PENDULUM

As mentioned before, the problem at hand is double pendulum to which we apply some standard physical assumption to make the problem a bit simpler. The conditions and physical form is explained below. The theory explained below is referenced from *myPhysicsLab.com*[1], where you can also find a simple simulation of a double pendulum using fixed step-size Runge-Kutta.

In the double pendulum, one end of the pendulum if fixed to an arbitrary pivot point. From the pivot point, a massless rigid rod with a bob is hanged. From this first bob, a second bob is connected via another massless rigid rod. The first rod and bob from the top are subscripted with 1 and the latter subscripted with 2. The length of the rigid rod, the mass of the bob, and the angle made by the rod with the perpendicular angle are noted by L, m, and θ. Thus, L_1, m_1, and θ_1 correspond to the first bob while L_2, m_2, and θ_2 correspond to the second bob. The overall physical setup is shown in Figure 1.

The aim is to obtain the change in θ_1 and θ_2 with the passage of time while keeping a certain level of accuracy. This can be achieved by adapting the step size of RK4 to obtain the said accuracy.

A. Importance

Here we will be solving only an instance of the chaotic pendulum which is very sensitive to the initial conditions. We can take this as an analogy to any problem whose path across time is "chaotic" thus any fixed step size would not be able to solve the problem accurately as it might miss a very sharp turn of the path. We can see how the ASRK4 adapts to the chaotic pendulum thus showing how it can adapt to a chaotic path ahead of any physical problem which is explained by a set of first-order Ordinary differential equations (ODEs).

B. Why adaptive step-size RK4?

RK4 is already known for its accuracy for a given step size among its peers[2]. By making it adaptive, we will further enhance its performance while trying to keep it reasonably simple. As mentioned before, the ASRK4 will adapt the step size according to the possible error estimate. i.e. it will spend more steps on the larger in-

$$\theta_1' = \omega_1 \tag{1}$$
$$\theta_2' = \omega_2 \tag{2}$$
$$\omega_1' = \frac{-g(2m_1+m_2)sin\theta_1 - m_2 g sin(\theta_1-2\theta_2) - 2sin(\theta_1-\theta_2)m_2(\omega_2^2 L_2 + \omega_1^2 L_1 cos(\theta_1-\theta_2))}{L_1(2m_1+m_2-m_2 cos(2\theta_1-2\theta_2))} \tag{3}$$
$$\omega_2' = \frac{2sin(\theta_1-\theta_2)(\omega_1^2 L_1(m_1+m_2) + g(m_1+m_2)cos\theta_1 + \omega_2^2 L_2 m_2 cos(\theta_1-\theta_2))}{L_2(2m_1+m_2-m_2 cos(2\theta_1-2\theta_2))} \tag{4}$$

FIG. 2. The equations of motion of Double pendulum as first-order ODEs

crease in the variable under analysis while increasing the step-size and moving fast where the change is relatively smaller.

In simpler terms, ASRK4 can detect sudden turns in paths and reduce step size to increase accuracy while reducing the step size and moving fast when the path doesn't vary much. Note that this path can be any variable under question. The results will clarify this statement further.

C. Equations for Double pendulum

The complete derivation to the equations of motion is rather overexplained as we have studied it many times over at different levels. For a system of Double pendulum as explained before, we can arrive at the equation of motion as given in Figure 2. Now, for solving the problem, it is just to solve the obstacles in creating a general ASRK4.

III. ADAPTIVE STEP-SIZE RK4 (ASRK4)

As mentioned before, the first obstacle is having a notion of error in relation to step size. To set this notion, we will be starting an initial step-size h_i and an initial solution $[\theta_1, \omega_1, \theta_2, \omega_2]$ at $t = t_i$. We will perform double the step size in one go, and two single steps in another go to arrive at the solution at $t = t_i + 2h$ in two different ways. As expected the smaller steps will be more accurate than the bigger steps. The in RK4 method is in the order of h^5 which is ch^5. The errors in both methods can be written as:

$$X_{h+h} + c(h)^5 = X_{2h} + c(2h)^5 \tag{5}$$
$$X_{h+h} - X_{2h} = 30ch^5 \tag{6}$$
$$ch^5 = abs\left(\frac{X_{h+h} - X_{2h}}{30}\right) \tag{7}$$
$$error = abs\left(\frac{X_{h+h} - X_{2h}}{30}\right) \tag{8}$$

Then the new stepsize h' for a target accuracy δ is:

$$h' = h\left(\frac{h\delta}{error}\right)^{1/4} \tag{9}$$

The second obstacle was setting limits on change in step-size. As we are more intersted in accuracy, we will allow the step size to reduce till the error reaches the designated accuracy. i.e. till $h\delta/error$ is less than 1.

For the upper limit on growing we limit to twice the step size every time.

IV. APPLICATION ON SPECIFIC CASE

The application of the same to a specific case is attached in the folder as "code.py" and outputs as different graph.

There in 0-2 second graphs (Figure 3 and Figure 4), we can see the step size steady for RK4 and step size varying with change in path for ASRK4. we can also see that the error in the values for this specific case accumulates and deviate the path a lot while using fixed size RK4, whereas the ASRK4 adapts and follows solutions. See 0-10 sec graph (Figure 5 and 6).

[1] myphysicslab.com. *Double pendulum*. `https://www.myphysicslab.com/pendulum/double-pendulum-en.html`.

[2] By peers, I mean methods that have more or less the same level of simplicity in coding and application.

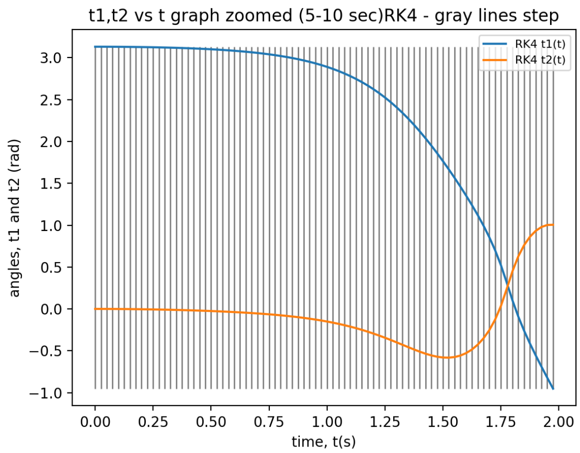

FIG. 3. 0-2 second graph RK4

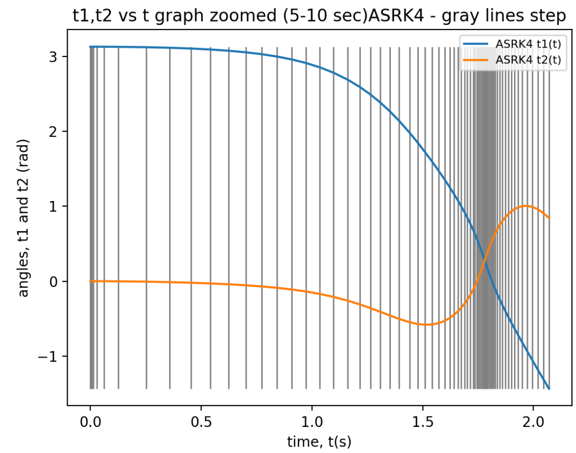

FIG. 4. 0-2 second graph ASRK4

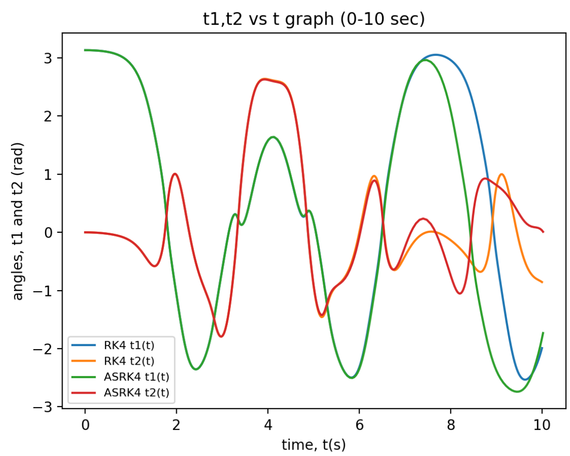

FIG. 5. 0-10 second graph ASRK4-RK4 comparison

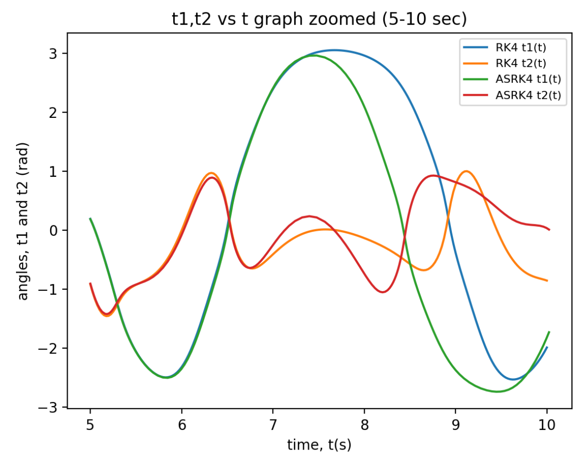

FIG. 6. 7-10 second graph ASRK4-RK4 comparison