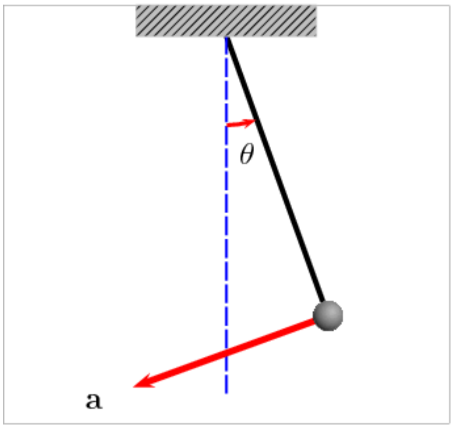
Report on the Nonlinear Simple Pendulum

Baalateja Kataru

Introduction

The Simple Pendulum system consists of a bob tied to a string that is suspended from a support. Upon supplying an initial acceleration to the pendulum, it starts to exhibit oscillatory behavior due to the opposing natures of the bob's momentum that tends to push the bob away from the mean position (middle) and the force of gravity acting as a restoring force that tends to pull the bob towards the mean position.



Equation of Motion

The equation of motion for the simple pendulum system is a **2nd order nonlinear differential equation**:

$$\ddot{\theta} + \frac{g}{L}\sin\theta = 0$$

Analytical Solution

If one does not linearize by applying the small angle approximation, the nonlinear differential equation of motion does not have a closed form solution that involves elementary functions, but instead must be solved numerically using a computer.

Numerical Solution

Converting to a system of differential equations

To solve the equation of motion numerically, we first need to convert the 2nd order differential equation

$$\ddot{\theta} + \frac{g}{L}\sin(\theta(t)) = 0$$

into a 1st order system of differential equations upon which standard numerical methods like Euler and Runge-Kutta that exist for solving first order differential equations can be applied.

We do that by defining a new variable ω

$$\omega(t) = \dot{\theta}$$

Physically, ω corresponds to the angular velocity of the pendulum, i.e., the rate of change of angular displacement with time. With this variable introduced, the 2nd order DE can be written as the following first order system of DEs

$$\begin{aligned} \dot{\theta} &= \omega(t) \\ \dot{\omega} &= -\frac{g}{L}\sin(\theta(t)) \end{aligned}$$

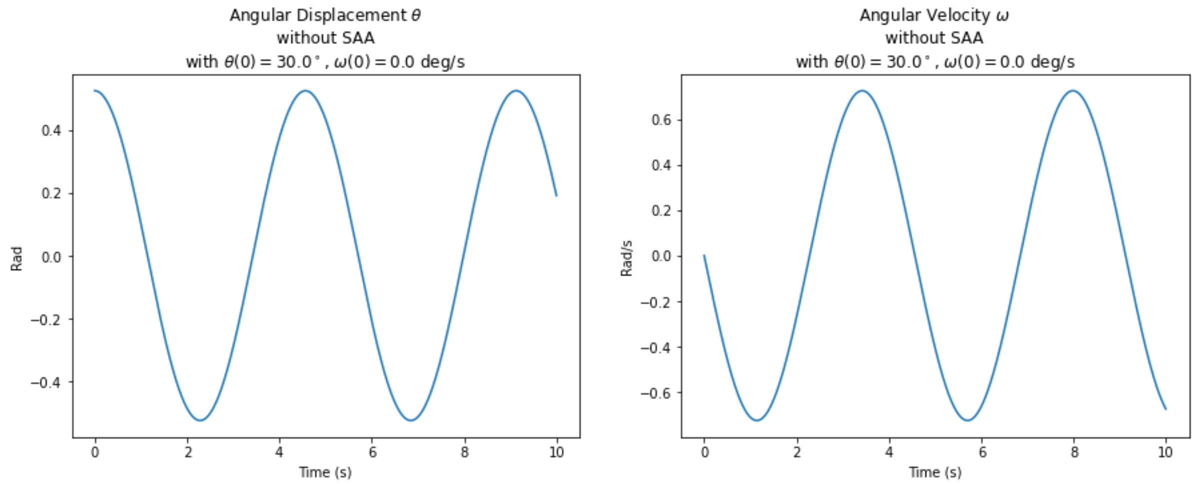
Now, we can employ the Runge-Kutta 4 method to solve this system numerically.

Results

In Python 3.10, we implemented RK4 to solve the DE system for a pendulum of length 5 meters, iterating from 0 to 10 seconds for 10000 time steps with the initial conditions

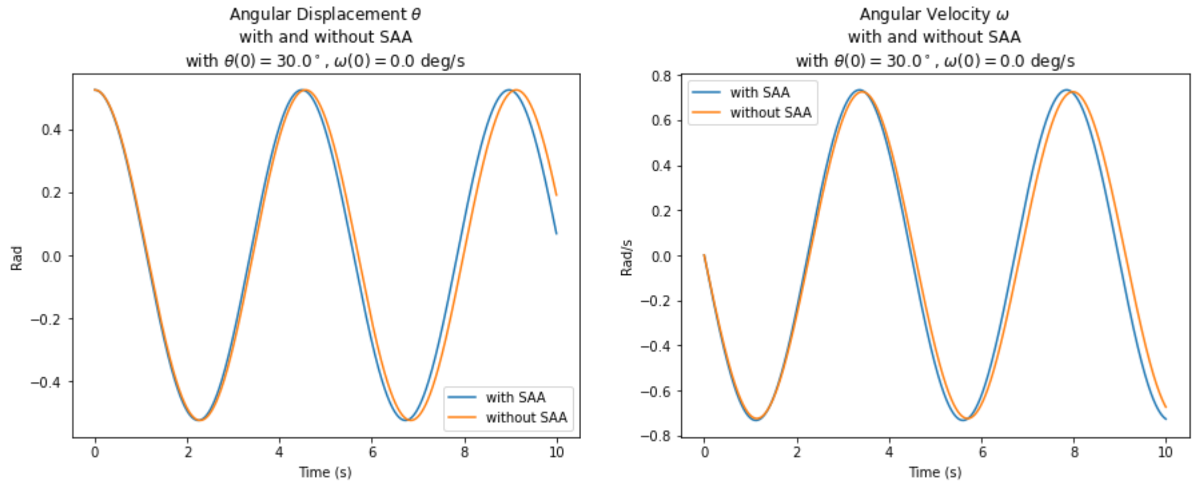
$$\theta(0) = 30^\circ$$
$$\omega(0) = 0$$

Upon solving, we obtain the following solutions for $\theta(t)$ and $\omega(t)$,



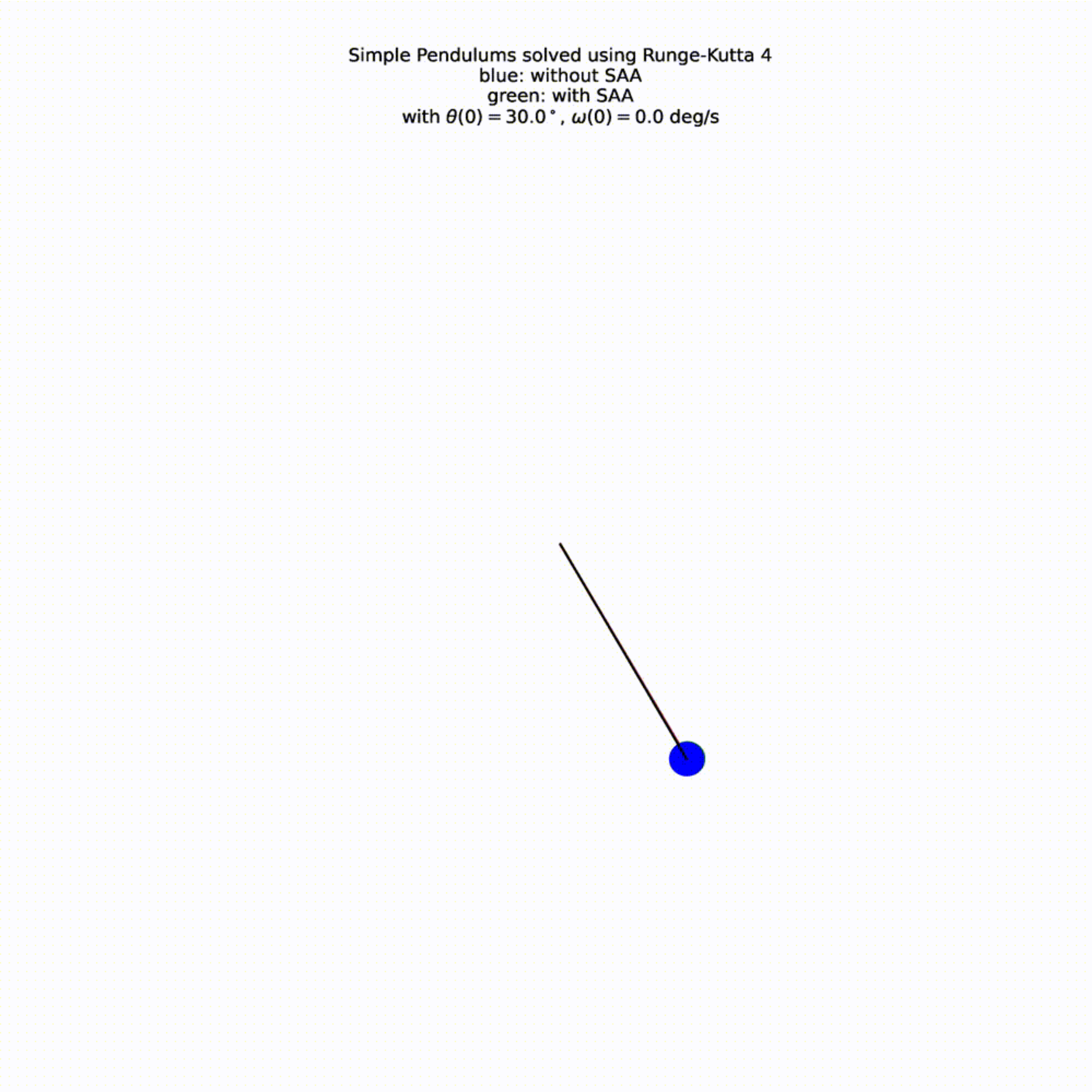
Observe how the solutions seem to be sinusoidal. This makes sense as the lack of any external forces means that the amplitude of oscillation for all time is the same as initial amplitude of oscillation which is the angular displacement specified in the initial conditions $\theta(0) = 30^\circ$, and this amplitude is comfortably in the range where the small angle approximation is valid so our solutions resembles those obtained by applying the approximation.

To highlight this further, we solve the DE system again with all the same parameters twice — once with and once without the small angle approximation (SAA) and compare results.



As expected, the differences between the solutions are negligible. However, notice that as time goes on the differences get larger. This is due to the nonlinear nature of the pendulum's motion creeping in, because as time increases the nonlinear effects compound and become more significant, making it difficult to ignore them when predicting the pendulum's behavior.

To illustrate this further, we used Python to create a handy simulation that details this increase of nonlinear influence over time visually by simulating the motion of two pendulums each with and without SAA. The full simulation as a .gif file can be found here.

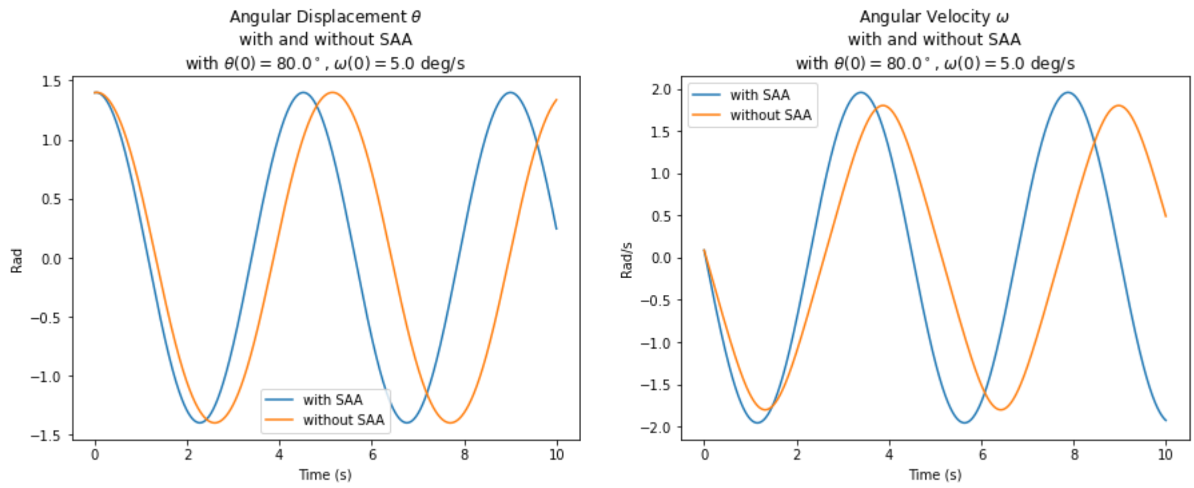


Apart from the passage of time, the nonlinear effects of the system become important when the oscillations are not small enough that SAA can be applied without large error.

As a demonstration, we solve the DE system again for all the same parameters except now we change the initial conditions to be

$$\theta(0) = 80^\circ$$
$$\omega(0) = 5 \text{ deg}/s$$

Thereby increasing the initial displacement (and therefore the amplitude of all oscillations) outside the range of applicability of SAA and also supplying an initial velocity. Then the solutions we obtain are



Which clearly shows that the larger the amplitude, the quicker and more the solutions diverge, proving why SAA has the criteria that it does for using it.

References

The code written for the analysis and simulations presented above is located in this Jupyter Notebook:

Simple-and-coupled-pendulums/Simple Pendulum - Runge-Kutta.ipynb at main · BK-Modding/Simple-and-coupled-pendulums

Jupyter Notebooks written while attempting to understand and solve the Simple Pendulum and Coupled Pendulum systems. - Simple-and-coupled-pendulums/Simple Pendulum - Runge-Kutta.ipynb at main · BK-...

https://github.com/BK-Modding/Simple-and-coupled-pendulums/blob/main/Simple%20Pendulum%20-%20Runge-Kutta.ipynb