PH2150A Scientific Computing Skills Team Project
Motion of a pendulum with the Runge–Kutta method

# 1 Introduction

In this project you will investigate the oscillations of a pendulum by solving the equation of motion numerically. We will not make any approximation about the amplitude of the motion being small, and we will consider the damping effects of a viscous medium. The solution to this problem cannot be written down in closed form, but it turns out to be relatively simple to find a numerical solution.

Specifically, consider a pendulum of length $l$. The angle $\varphi$ with respect to the vertical obeys the 2nd order differential equation

$$\ddot{\varphi} + 2\beta\dot{\varphi} + \omega_{\mathrm{c}}^2 \sin\varphi = 0 \ , \tag{1}$$

where $\beta \geq 0$ represents the effect of a viscous medium and $\omega_{\mathrm{c}} = \sqrt{g/l}$ is the characteristic frequency of small oscillations in the absence of damping. By defining $\omega = \dot{\varphi}$, equation (1) can be rewritten as two coupled 1st order equations,

$$\dot{\varphi} = \omega \tag{2}$$

$$\dot{\omega} = -2\beta\omega - \omega_{\mathrm{c}}^2 \sin\varphi \ . \tag{3}$$

The goal of this project is to investigate numerical solutions to these equations. The idea behind the numerical methods is to start from a given initial condition and take small steps in the independent variable (here time, $t$). The derivatives of the dependent variables (here the angle of the pendulum, $\varphi$ and the angular speed $\omega$) at $t$ are used to construct approximations for a slightly later time, $\varphi(t + \Delta t)$ and $\omega(t + \Delta t)$. The techniques are described in more detail below.

# 2 The assignment

Your assignment is

- to find out as much as you can about the solution $\varphi(t)$ obtained with different numerical techniques. These should include the Euler method and two variations of the Runge–Kutta method;
- to display the solutions as plots of $\varphi(t)$ and $\omega(t)$ versus $t$ and also as curves in the $(\varphi, \omega)$ plane, and to should show how these look with and without the effects of a viscous medium;
- to investigate how the accuracy of the solution depends on the step size $\Delta t$ for each of the methods used.

If time permits, you should also look at a pendulum with a periodic driving force, i.e. solve the equation

$$\ddot{\varphi} + 2\beta\dot{\varphi} + \omega_{\mathrm{c}}^2 \sin\varphi = \alpha \cos\omega_{\mathrm{d}} t \ , \tag{4}$$

where $\alpha$ represents the force's magnitude and $\omega_{\mathrm{d}}$ its frequency. Choose appropriate values of the parameters (try, say, $\omega_{\mathrm{c}}^2 = 1$, $\beta = 0.1$, $\alpha = 0.7$, $\omega_{\mathrm{d}} = 0.6$). Investigate solutions of this problem for different initial conditions.

## 3 The Runge–Kutta method

Our two coupled differential equations are of the form

$$\begin{aligned} \dot{\varphi} &= \omega \equiv f(t, \varphi, \omega) \ , & (5) \\ \dot{\omega} &= -2\beta\omega - \omega_{\mathrm{c}}^2 \sin\varphi \equiv g(t, \varphi, \omega) \ . & (6) \end{aligned}$$

These equations define the functions $f$ and $g$. In the general method for solving differential equations that we will use, $f$ and $g$ could depend on all of the variables of the problem. In our example, however, they do not depend explicitly on the time and indeed $f(t, \varphi, \omega) = \omega$ does not depend on $\varphi$. We can consider discrete steps in time of size $\Delta t$, and define

$$t_n = t_0 + n\Delta t \ , \quad n = 0, 1, \ldots \ . \tag{7}$$

where the initial conditions for our pendulum are given as $\varphi(t_0) = \varphi_0$ and $\omega(t_0) = \omega_0$. We can approximate the solution at time $t_{n+1}$ in terms of the solution at time $t_n$ by

$$\varphi(t_{n+1}) \approx \varphi(t_n) + \Delta t \, \dot{\varphi}(t_n) \tag{8}$$

$$\omega(t_{n+1}) \approx \omega(t_n) + \Delta t \, \dot{\omega}(t_n) \ . \tag{9}$$

To obtain, for example, $\dot{\omega}(t_n)$ for the right-hand side of (9), one evaluates the corresponding right-hand side of (3) using $\varphi(t_n)$ and $\omega(t_n)$. By starting at $t_0$ and repeating this rule with a sufficiently small step size $\Delta t$, the solution at an arbitrary later time $t_n$ can be found. This is called the *Euler method*. It is not widely used in practice since far better methods (namely, the Runge–Kutta method) are available, but it contains the main idea. Since the approximation for $\varphi_{n+1}$ and $\omega_{n+1}$ is based on a 1st order Taylor expansion about the solution at time $t_n$, you can easily show that the error is proportional to $(\Delta t)^2$.

An improvement over equations (8)–(9) is provided by the Runge-Kutta method (pronunciation: Roong-eh-Kutta). The problem with the Euler method was that the approximation for, say, $\varphi_{n+1}$ used the derivative $\dot{\varphi}$ evaluated at the time $t_n$. But in the course of stepping from $t_n$ to $t_{n+1}$, the derivative changes. A better guess would use the derivative averaged over the interval between $t_n$ and $t_{n+1}$. As an approximation to this we can use the derivative at a point half-way between $t_n$ and $t_{n+1}$, i.e. at $t_{n+\frac{1}{2}} = \frac{1}{2}(t_n + t_{n+1})$. Our rule is then

$$\varphi_{n+1} = \varphi_n + \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, \varphi(t_n + \tfrac{1}{2}\Delta t), \omega(t_n + \tfrac{1}{2}\Delta t))\,, \tag{10}$$

$$\omega_{n+1} = \omega_n + \Delta t\, g(t_n + \tfrac{1}{2}\Delta t, \varphi(t_n + \tfrac{1}{2}\Delta t), \omega(t_n + \tfrac{1}{2}\Delta t))\,. \tag{11}$$

The problem is that we don't have $\varphi(t_n + \frac{1}{2}\Delta t)$ or $\omega(t_n + \frac{1}{2}\Delta t)$. For these, however, we can use the approximations

$$\varphi(t_n + \tfrac{1}{2}\Delta t) \approx \varphi_n + \tfrac{1}{2}\Delta t\, f(t_n, \varphi_n, \omega_n)\,, \tag{12}$$

$$\omega(t_n + \tfrac{1}{2}\Delta t) \approx \omega_n + \tfrac{1}{2}\Delta t\, g(t_n, \varphi_n, \omega_n)\,. \tag{13}$$

That is, our method of finding $\varphi_{n+1}$ and $\omega_{n+1}$ from $\varphi_n$ and $\omega_n$ now consists of the following steps:

$$\begin{aligned}
k_1 &= \Delta t\, f(t_n, \varphi_n, \omega_n), & l_1 &= \Delta t\, g(t_n, \varphi_n, \omega_n), \\
k_2 &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, \varphi_n + \tfrac{1}{2}k_1, \omega_n + \tfrac{1}{2}l_1), & l_2 &= \Delta t\, g(t_n + \tfrac{1}{2}\Delta t, \varphi_n + \tfrac{1}{2}k_1, \omega_n + \tfrac{1}{2}l_1), \\
\varphi_{n+1} &= \varphi_n + k_2\,, & \omega_{n+1} &= \omega_n + l_2\,.
\end{aligned} \tag{14}$$

Note that we still only needed the values of $\varphi$ and $\omega$ at the time $t_n$ to determine the values $t_{n+1}$. You can show that the error from this method is proportional to $(\Delta t)^3$.

The most widely used version of the Runge–Kutta method uses the same basic idea, but a more sophisticated formula for the average slope, based on the derivatives at the beginning, middle and end of the step. The algorithm is

$$\begin{aligned}
k_1 &= \Delta t\, f(t_n, \varphi_n, \omega_n), & l_1 &= \Delta t\, g(t_n, \varphi_n, \omega_n), \\
k_2 &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, \varphi_n + \tfrac{1}{2}k_1, \omega_n + \tfrac{1}{2}l_1), & l_2 &= \Delta t\, g(t_n + \tfrac{1}{2}\Delta t, \varphi_n + \tfrac{1}{2}k_1, \omega_n + \tfrac{1}{2}l_1), \\
k_3 &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, \varphi_n + \tfrac{1}{2}k_2, \omega_n + \tfrac{1}{2}l_2), & l_3 &= \Delta t\, g(t_n + \tfrac{1}{2}\Delta t, \varphi_n + \tfrac{1}{2}k_2, \omega_n + \tfrac{1}{2}l_2), \\
k_4 &= \Delta t\, f(t_n + \Delta t, \varphi_n + k_3, \omega_n + l_3), & l_4 &= \Delta t\, g(t_n + \Delta t, \varphi_n + k_3, \omega_n + l_3), \\
\varphi_{n+1} &= \varphi_n + \tfrac{k_1}{6} + \tfrac{k_2}{3} + \tfrac{k_3}{3} + \tfrac{k_4}{6}, & \omega_{n+1} &= \omega_n + \tfrac{l_1}{6} + \tfrac{l_2}{3} + \tfrac{l_3}{3} + \tfrac{l_4}{6}\,.
\end{aligned} \tag{15}$$

Note that both $k_1$ and the corresponding quantity $l_1$ must both be found before proceeding to find $k_2$ and $l_2$, both of which are needed to find $k_3$ and $l_3$, and so forth. This procedure can clearly be generalised to $n$ coupled equations, and hence we can also use it to solve a single $n$th order equation.

# 4 Teamwork

You will need to divide up the work so that each member has specific tasks to accomplish and the team achieves its goals to the fullest extent possible. Below are some suggestions on possible ways of separating the project into individual tasks.

In Java you will probably define a class called, say, `PenVector`, that contains $\varphi$ and $\omega$. Your program should create an object of type `PenVector` and initialise it with the starting conditions $\varphi(t_0) = \varphi_0$, $\omega(t_0) = \omega_0$. You should also define a class which has methods to evaluate the derivatives $\dot{\varphi}$ and $\dot{\omega}$. In addition you should:

- Implement an `Euler` class with a method that carries out the updating rule described by equations (8) and (9).
- Implement an `RK` class with a method that carries out the updating rule described by equations (15). This method should have the same name and return types as the corresponding method in the `Euler` class so that they can be used interchangeably. You can easily do this as well with the updating rule described by equations (14).
- Store the data values produced by successive updates so you can pass them to a plotting routine.

You should write a driver program with the `main` method that creates objects using the classes above and executes the various methods in the desired way.

## 5 References

The Runge–Kutta method is discussed in most books on differential equations. A complete description can be found in

W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes*, 2nd edition, Cambridge University Press, Cambridge (1992).

A discussion of the numerical solution of the pendulum problem can be found in

N.J. Giordano, *Computational Physics*, Prentice Hall (1997).

G. Cowan, 19 November, 2001