Computational Physics: Assignment 4

Sarah Roberts

November 9, 2018

Part 1: Integration. Functions were written in python to implement the Trapezoid method and Simpson's method. The included code first tests the methods on

$$\int_{-1}^{1} x^2 + 2x + 2 = 4.666...$$

The trapezoid method converges to 4.663 for $n = 1532$ steps, which has a percent error of 0.065% while Simpson's method converges to 4.665 for $n = 521$, which has a percent error of 0.027%. As expected, Simpson's method converges much more quickly than the Euler method.

After testing with a simple integral, the same functions were applied to approximating

$$\int_{-100}^{100} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}} = 1.0$$

The trapezoid method approximated an integral of 0.9986 for $n = 712$, which has percent error of 0.1404%. Simpson's rule approximated the integral as 0.9962 after $n = 253$ iterations, with a percent error of 0.3773%. The following two plots show the error as a function of number of steps n for each of the two methods, when evaluating the latter integral.

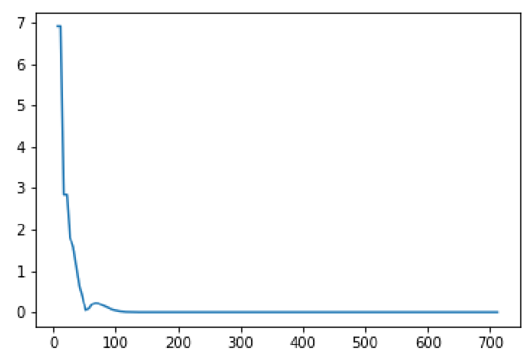

FIGURE 1. `trap.png` Percent error as a function of n when approximating the integral with trapezoid method.

Part 2: Ordinary Differential Equations (ODEs). Functions to approximate the solution for an ordinary differential equation were written in Python. They were applied to the ODE:

$$\frac{dx}{dt} = 4x - t + 1$$

$$\text{where } x(0) = 1$$

First, Euler's method is called for $n = 1000$, so the method is applied to 1000 evenly spaced steps on the domain $t = [0, 1]$. A plot of this solution and the analytic solution is shown below. Then, the Runge-Kutta function is called for $n = 500$. A plot with this solution is shown below and also includes the analytic solution and the Euler approximation.

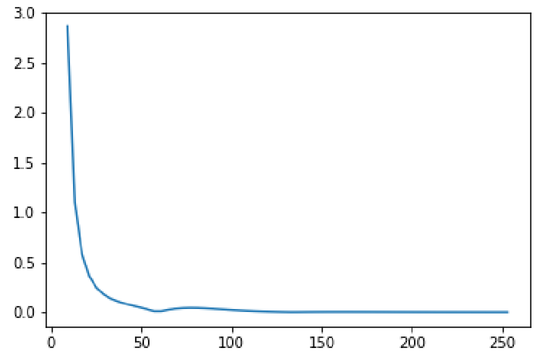

FIGURE 2. `simpson.png` Percent error as a function of n when approximating the integral with Simpson's method. Note the different scale on the x axis, compared to Figure 1. Simpson's method converges much more quickly than the trapezoid method.

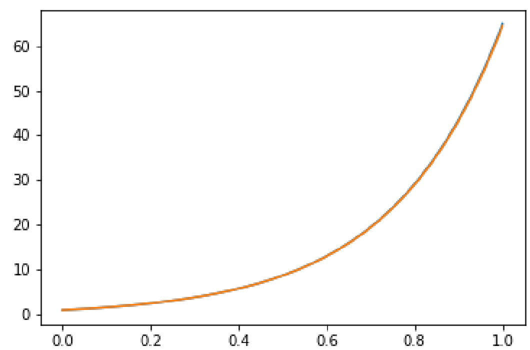

FIGURE 3. `euler.png` This figures shows the analytic solution in blue and the Euler approximation in orange.

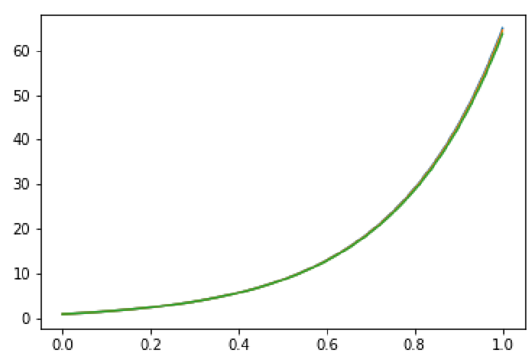

FIGURE 4. `rungekutta.png` This figures shows the analytic solution in blue, the Euler approximation in orange, and the Runge-Kutta approximation in green.

Finally, the error in the final step ($t = 1.0$) was compared between the Euler and Runge Kutta methods for varying step size. The error was calculated as a fraction of the correct value, according to the analytic solution

$$x(t) = \frac{1}{16}(4t + 19e^{4t} - 3)$$
$$\text{err} = \frac{|x(t) - x_{appx}(t)|}{x(t)}$$

where $x_{appx}(t)$ is the numerical approximation of x at t.

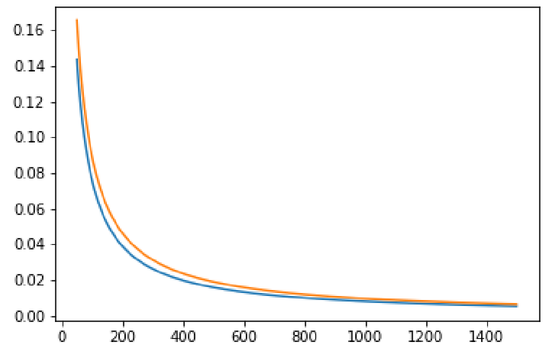

FIGURE 5. `eulerRKErrors.png` The relative error in the approximation of x(t = 1.0) was compared between the Euler and Runge-Kutta methods for varying step size. The Euler method is shown in orange and Runge-Kutta in blue. The x axis is number of steps (more steps is a smaller step size) and y axis is the error (smaller is better).