Math 342: Project 2

Connor Emmons

Documentation: The main Project 2 MatLab script and all required dependencies are located in the Project 2 folder found here: https://github.com/Connor-Lemons/Emmons-Math-342. No other resources used.

Problem 1: Arrange the parameters of RK4 in a Butcher Tableu

$$w_0 = \alpha \tag{1}$$

$$K_1 = hf(t_k, w_k) \tag{2}$$

$$K_2 = hf\left(t_k + \frac{h}{2}, w_k + \frac{1}{2}K_1\right) \tag{3}$$

$$K_3 = hf\left(t_k + \frac{h}{2}, w_k + \frac{1}{2}K_2\right) \tag{4}$$

$$K_4 = hf(t_{k+1}, w_k + K_3) \tag{5}$$

$$w_{k+1} = w_k + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \tag{6}$$

The Butcher Tableu representation of this method is:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Problem 2: For the three RK methods shown below, given the Butcher Tableau representation, give the equations which define those methods.

Ralston Method

$$\begin{array}{c|cc} 0 & & \\ \frac{2}{3} & \frac{2}{3} & \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

Heun Third-Order Method

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{3} & \frac{1}{3} & & \\
\frac{2}{3} & 0 & \frac{2}{3} & \\
\hline
 & \frac{1}{4} & 0 & \frac{3}{4}
\end{array}
$$

Runge-Kutta 3/8 Rule

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{3} & \frac{1}{3} & & & \\
\frac{2}{3} & -\frac{1}{3} & 1 & & \\
1 & 1 & -1 & 1 & \\
\hline
 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8}
\end{array}
$$

The equations for the Ralston Method are:

$$w_0 = \alpha \tag{1}$$

$$K_1 = hf(t_k, w_k) \tag{2}$$

$$K_2 = hf\left(t_k + \frac{2}{3}h, w_k + \frac{2}{3}K_1\right) \tag{3}$$

$$w_{k+1} = w_k + \frac{1}{4}(K_1 + 3K_2) \tag{4}$$

The equations for the Heun Third-Order Method are:

$$w_0 = \alpha \tag{5}$$

$$K_1 = hf(t_k, w_k) \tag{6}$$

$$K_2 = \mathrm{hf}\left(t_k + \frac{1}{3}h, w_k + \frac{1}{3}K_1\right) \tag{7}$$

$$K_3 = hf\left(t_k + \frac{2}{3}h, w_k + \frac{2}{3}K_2\right) \tag{8}$$

$$w_{k+1} = w_k + \frac{1}{4}(K_1 + 3K_3) \tag{9}$$

The equations for the Runga-Kutta 3/8 Rule are:

$$w_0 = \alpha \tag{10}$$

$$K_1 = hf(t_k, w_k) \tag{11}$$

$$K_2 = hf\left(t_k + \frac{1}{3}h, w_k + \frac{1}{3}K_1\right) \tag{12}$$

$$K_3 = hf\left(t_k + \frac{2}{3}h, w_k - \frac{1}{3}K_1 + K_2\right) \tag{13}$$

$$K_4 = hf(t_k + h, w_k + K_1 - K_2 + K_3) \tag{14}$$

$$w_{k+1} = w_k + \frac{1}{8}(K_1 + 3K_2 + 3K_3 + K_4) \tag{15}$$

Problem 3: Implement the three methods discussed in Problem 2 in MatLab.

Code can be found in the GitHub page or attached at the end of this document.

https://github.com/Connor-Lemons/Emmons-Math-342/tree/main/Project%202

Problem 4: Implement the Runge-Kutta-Fehlberg method according to Algorithm 5.3.

Code can be found in the GitHub page or attached at the end of this document.

https://github.com/Connor-Lemons/Emmons-Math-342/tree/main/Project%202

Problem 5: Rewrite the equation $y'(t) = kx(t)y(t)$, where $x(t)$ is the number of susceptible (non-infective) individuals and y(t) is the number of infective individuals, as a function solely of $y(t)$ and m, which represents the total population.

Note that the the sum of the number of non-infective individuals and the number of infective individuals is the size of the population. This gives:

$$m = x(t) + y(t) \tag{1}$$

Solving for $x(t)$ gives:

$$x(t) = m - y(t) \tag{2}$$

Making this substitution in the original equation gives:

$$y'(t) = k\big(m - y(t)\big)y(t) \tag{3}$$

$$y'(t) = kmy(t) - ky(t)^2 \tag{4}$$

Problem 6: Use the methods described in Problems 1-4 to obtain various estimated solutions to equation (4) in Problem 5.

Parameters:

$$m = 100000 \tag{1}$$

$$y(0) = 1000 \tag{2}$$

$$k = 2 \times 10^{-6} \tag{3}$$

$$t_i = 0\ days \tag{4}$$

$$t_f = 30\ days \tag{5}$$

The results of these methods can be found in the Github page or attached at the end of this document.

https://github.com/Connor-Lemons/Emmons-Math-342/tree/main/Project%202

The final estimation of the number of infective individuals from each method are shown below. Note that all values are rounded to the nearest integer.

$h = 2$:

Ralston: $y(30) = 79319$

Heun: $y(30) = 80231$

RK38: $y(30) = 80289$

RK4: $y(30) = 80288$

$h = 1$:

Ralston: $y(30) = 80028$

Heun: $y(30) = 80287$

RK38: $y(30) = 80295$

RK4: $y(30) = 80295$

RKF45: $y(30) = 80296$

All estimations are relatively similar. The higher the order of the method, the more accurate it is likely to be. Running the methods with a smaller h also likely produced more accurate results, with the most accurate method being the RKF45 method due to its high order and optimizing of h to match a specified tolerance level.