Project 4: Runge-Kutta ODE Solver

Ethan Storer

Numerical Methods ME 355

Section A

Code Written on December 3, 2022

Report Submitted on December 3, 2022

I have neither given or received, nor have I tolerated others’ use of unauthorized aid.

-Ethan Storer

Figures:

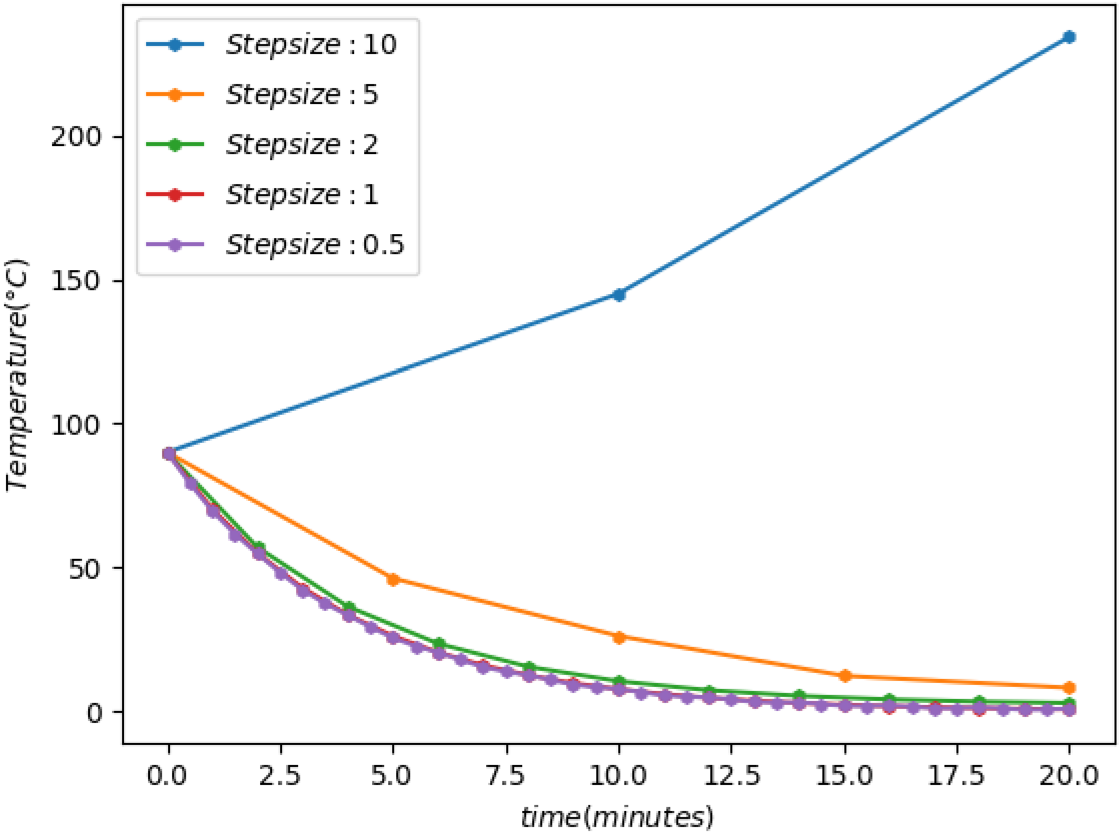

Figure 1. As RungeKutta.py with Heun's method uses smaller step sizes,, the curve approximates a closer fit to the true temperature curve. The final temperature also becomes more accurate.

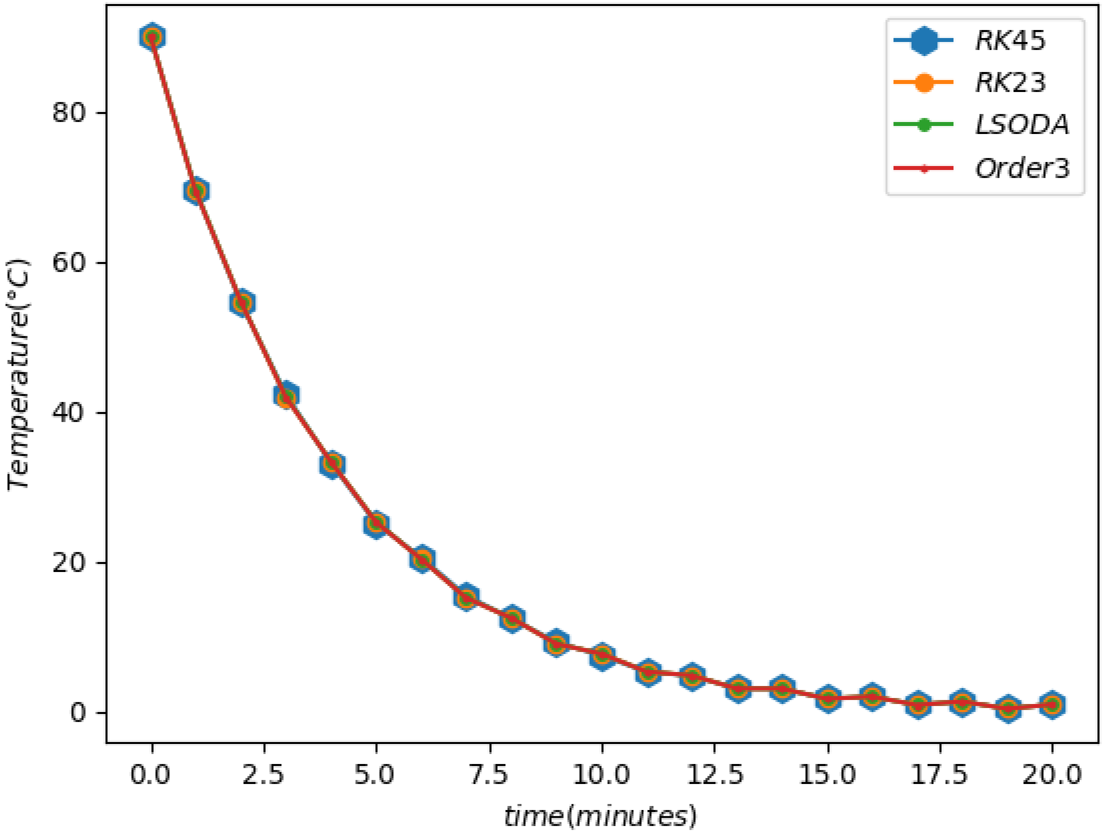

Figure 2. The Scipy methods plot each point almost identically. The marker icons are sized and stacked to show each plotted method. RK23 solves the same way as Order3, but it assumes the error of a 2nd Order Runge-Kutta Method. RK45 solves with a 5th Order Runge-Kutta method but assumes the accuracy of a 4th Order Runge-Kutta method. RK23 and RK45 have a margin of safety for error. The LSODA method switches between the Adams method for nonstiff differential equations and the Backward Differentiation Formula for stiff differential equations. LSODA seems to have more utility than the Explicit Runge-Kutta methods since it combines two specialized methods to become a general tool.

Deliverables:

Table 1. Accuracy increases with computation time. Each Scipy method takes at least 3 times as long as the Heun and Order 3 methods. Given the Scipy methods are all running two methods, the amount of increased computation time makes sense. The 2nd Order Heun method and Order 3 have the poorest accuracy, but they were the fastest methods. At 20 iterations, they take almost one millisecond to compute.

Method	Temperature at 20 min	Number of Iterations	Average Step Size	Speed (milliseconds)
(1) Heun's	234.309	2	10	0.0-0.99
(2) Heun's	8.127	4	5	
(3) Heun's	2.84	10	2	
(4) Heun's	0.681	20	1	
(5) Heun's	0.911	40	0.5	
Order 3	0.984	20	1	1.00
Scipy RK45	0.973	20	1	2.99
Scipy RK23	0.977	20	1	7.97
Scipy LSODA	0.979	20	1	2.99