Phys 512 Assignment 3

Ian Hendricksen

October 5, 2021

All relevant outputs from my script can be found in the file `A3 Print Output.txt`.

1. Following from Numerical Recipes' method for canceling the leading-order error term (5th order), we can implement Runge-Kutta first by taking a full step of length h, and then two half steps of length $h/2$. We then have two evaluations of $y(x+h)$ as

$$\begin{aligned} y(x+h) &= y_h + (2h)^5\phi + 0(h^6) \\ y(x+h) &= y_{h/2} + 2(h^5)\phi + O(h^6), \end{aligned} \tag{1}$$

where y_h is evaluated using Runge-Kutta for one full h step, and $y_{h/2}$ is evaluated using Runge-Kutta for two $h/2$ steps (Eqn. 17.2.1 in text). The fifth-order error terms reflect the manner in which y_h and $y_{h/2}$ are calculated. If we define $\Delta \equiv y_{h/2} - y_h$, we can rewrite (1) as

$$y(x+h) = y_{h/2} + \Delta/15 + O(h^6), \tag{2}$$

and we have thus canceled the leading 5th-order error term. This will be our definition to use in our function `rk4_stepd`. This uses 11 function evaluations every time we want to integrate a new y value. One full step of length h requires 4 function evaluations (i.e. the number of function evaluation of `rk4_step`), and each half step of length $h/2$ requires 4 function evaluations, but we evaluate the same $k1$ for the full step and the first half step, so we don't have to re-evaluate this value.

We evaluate the ODE and initial condition

$$\frac{dy}{dx} = \frac{y}{1+x^2} \quad y(-20) = 1 \tag{3}$$

from x = [-20, 20] using both `rk4_step` and `rk4_stepd`, with 200 steps. The explicit solution is

$$y(x) = c_0 e^{\tan^{-1}(x)} \quad c_0 = \frac{1}{e^{\tan^{-1}(-20)}}. \tag{4}$$

Defining our error estimate as the standard deviation of the difference between the exact solution and each `rk4` evaluation, we find that the error of `rk4_step` is 0.00011776140531974458, and the error of `rk4_stepd` is 0.02890610767690144 (both printed each time the script is run).

Now we want to integrate the ODE using `rk4_stepd` such that we evaluate the function the same number of times as rk4_step. For 200 steps, rk4_step evaluates the function 800 times. For `rk4_stepd` to evaluate the function 800 times, `rk4_stepd` needs to have $800/11 \sim 73$ steps.

2.

3.