Assignment 2

ESS101

Albin Ahlskog
Henrik Bertilson

October 16, 2023

1 General Runge-Kutta 2 methods

For the 1a) question to get a model that has an error of order 3, we need to take a newton expansion of the second order. Using the model given in the question we get:

$$x_{k+1} = x(t_k) + \Delta t b_1 f(x) + \Delta t^2 \frac{b_2 a}{2} \dot{f}(x(t_k))$$

which if compared to the given equation we can solve that $(b_1 + b_2 = 1, b_2 a = 0.5)$

For Q1b) RK2 methods are said to be of order 2 since the *global error* is of order 2. The error term e_k is the one-step error of order Δt^3. After $N = \dfrac{T}{\Delta t}$ steps the total error will be $N\Delta t^3 = T\Delta t^2$, so the global error is of order 2 while the one-step error is of order 3.
For Q1c) writing out the systems derivates you get

$$\begin{aligned} n = 1 :& \alpha_1 \\ n = 2 :& \alpha_1 + \alpha_2(t - t_k) \\ n = 1 :& \alpha_1 + \alpha_2(t - t_k) + \alpha_3/2(t - t_k)^2 \end{aligned}$$

comparing it to get values from $x(t_{k+1})$ given we get $(n = 1, b_1 + b_2 = 1)$, $(n = 2, b_1 + b_2 = 1, b_2 c = 0.5)$ and for n=3 we get an term of order 3, which is higher than what RK2 can exactly calculate, which makes it impossible to solve exact.

2 Accuracy & stability

The function `RKsolver.m` was made following the pseudo code in the course book, page 166. When simulating the test system $\dot{x} = \lambda x$ with the given conditions and comparing the results of RK1, RK2 and RK4 to the exact solution the outcome is that RK2 and RK4 are virtually indistinguishable from the exact solution, which is to be expected following the theory.

The accuracy of the different schemes also follow the expected behaviour, even though MATLAB has a hard time keeping up with the minuscule error of RK4 when approaching a δT of 10^-4. Compared to their respective theoretical accuracy, RK1 performs a bit worse, RK2 a bit better and RK4 can't really be with anything the book.

Stability can be checked using $S = \left| \sum_{k=0}^{\mathcal{O}} \dfrac{(\lambda \Delta t)^k}{!k} \right|$ and the results come out as follows: RK1 and RK2 become unstable at $\lambda = -20$ and RK4 at $\lambda = -27.853$.

3 Van-der-Pol oscillator

Running the system of equations defining the Van der Pol-osciallator with the given parameters through MATLABs `ODE45` and plotting the results gives figures behaving exactly as in the figure on page 174 in the course book.

Running `RKsolver.m` on the same system using the RK4 shows that the function gives very good results that are only marginally worse than `ODE45`. Even with the tolerance set to 10^{-8} Δt can be kept at 0.1.

4 IRK

The implicit Runge-Kutta schemes are solved by the function `IRKsolver.m` which was created using various formulas from chapter 8 in the course book and some parts of the PSS codes.

Comparing `IRKsolver.m`, using the given IRK4 scheme, to `RKsolver.m` running the RK4 scheme on the test system $\dot{x} = \lambda x$ shows that they perform equally. However, when comparing them against eachother on the Van der Pol-system, the RK4 scheme performs better as time is advanced, even though they follow each other very closely.