Modelling and Simulation Assignment 4

Varun G Hegde
Arshad N

21 October 2019

1 Introduction

This assignment consists of three parts. The first part concentrate on Implicit Runge-Kutta method based on approach Integration of implicit ODE.The second part is about Gauss-Legendre collocation method and this is solved by Matlab codes.The last part is DAE Integration where we widely-used implicit integration techniques, and their behaviour for the simulation of various dynamic systems.

2 Implicit RK1

(a) Consider the implicit ODE

$$F\left(\frac{x_{k+1} - x_k}{\Delta t}, x_k, u(t_k)\right) = 0 \tag{1}$$

To prove this Implicit ODE is in Explicit Euler scheme.
From the lecture notes we take the equation (6.19)

$$x_{k+1} = x_k + \Delta t K_1 \tag{2}$$

where $\mathrm{K}_1 = f(x_k, u(t_k))$ substituting in eqn (2).,

$$x_{k+1} = x_k + \Delta t f(x_k, u(t_k)) \tag{3}$$

Rearranging eqn (3) we get

$$f(x_k, u(t_k)) = \frac{x_{k+1} - x_k}{\Delta t} = \dot{x} \tag{4}$$

From the given $\mathrm{F}(\dot{x}, x, u) = 0$, substituting (4), we get

$$F\left(\frac{x_{k+1} - x_k}{\Delta t}, x_k, u(t_k)\right) = 0 \tag{5}$$

Hence proved.

(b) The Explicit Euler scheme can be written as

$$x_{k+1} = x_k + \Delta t f(x_{k+1}, u_{k+1}) \tag{6}$$

or

$$f(\dot{x}, x_{k+1}, u_{k+1}) = 0 \tag{7}$$

Algorithm: Implicit Euler Scheme

Input: $\mathrm{x}_0, u(t_0).......u(t_{N-1})\Delta t$
for $k = 0, ..., N-1$

Got an good approximate or x_{k+1} (Neighborhood of the solution of the convergence)

Introduce a While loop to compute the solution for Δx_{k+1} such that $\|r(x_{k+1}, x_k, u_{k+1})\|$ is within tolerance

while $\|r(x_{k+1}, x_k, u_{k+1})\| > Tol$

$$\frac{\partial r(x_{k+1}, x_k, u_{k+1})}{\partial x_{k+1}} \Delta x_{k+1} + r(x_{k+1}, x_k, u_{k+1}) = 0 \tag{8}$$

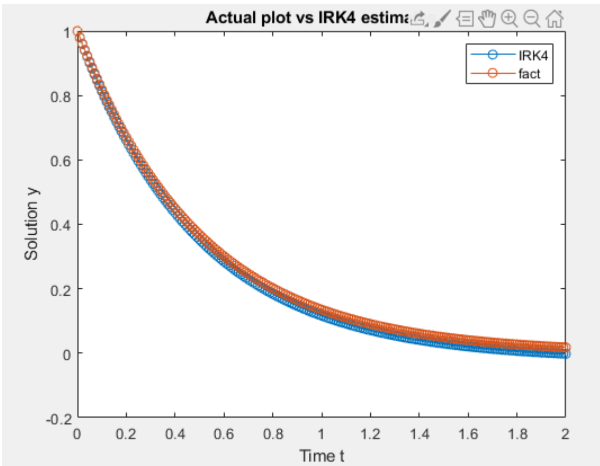
where r is given by .

$$x_{k+1} \leftarrow x_{k+1} + \alpha \Delta x_{k+1} \tag{9}$$

for some step size $\alpha \in]\ 0,1]$ (a full step α =1 generally works for implicit integrators)
Then compute to return $\|r(x_{k+1}, x_k, u_{k+1})\|$ for while loop
return $\mathrm{x}_1, ..., N$

3 Gauss-Legendre IRK

(a)
A Matlab script to implement the IRK4 on a problem posed in assignment3 was deployed . It was observed that IK4 yielded a much lower global in contrast to RK2. The following observation was made through the plot attached below,



(b)

A Matlab script to implement the IRK4 on Vander-pol function was also deployed . But there was a difficulty in obtaining a invertible matrix as the jacobian were not square matrix when we considered 2 variables for the Vander-Pol function and was yielding 4*2 matrix. But we are supposed to observe the same result as 3a).

Due to the error faced in the question 2 , we spent a lot of time trying to correct the same but eventually we ran out of time. thus very less progress could be made on problem 4. Thus please allow us to resubmit the same and will adhere to its deadline.