Numerical Analysis

Assignment 2

Modifying 4 stage

Runge-Kutta (R.K) Method

Abdul Rehman Ansari

24487

Equation used:

, (0) = 87/1.23

Constants used:

At Initial Height = 1:

C1 = 1 Theta 1 = NA Weight 1 = NA

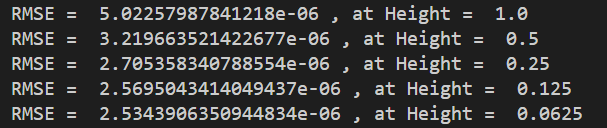
C2 = 2 Theta 2 = 0.25 Weight 2 = 0.25

C3 = 3 Theta 3 = 0.50 Weight 3 = 0.50

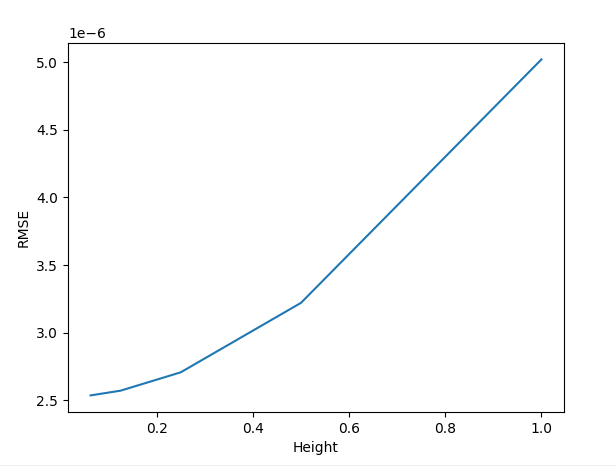
C4 = 4 Theta 4 = 0.75 Weight 4 = 0.75

*NOTE: During calculations, C1, C2, C3, C4 were summed and then divided by the sum to keep their sum always equal to 0*

Errors:



Height-RMSE Graph:



Methodology:

* A program that performs the Runge-Kutta method (R.K Method) was developed in such a way that its weights, thetas and constants could be changed.
* The program was fed a differential equation generated through the given method and my ERP ID (24487).
* The constants were tweaked and the calculated y coordinates were stored
* A python library (scipy) was used to calculate very close approximations to the actual y coordinates, which were stored separately
* The calculated and “actual” y coordinates were used to calculate the Root Mean Squared Error (RMSE) of y coordinates
* The RMSE for different Heights was stored and then graphed to study the relation between RMSE and the height of the interval.
* This process was repeated for several combinations until an acceptable range of RMSE was acquired

Findings:

* No matter what constants were used, RMSE was found to be decreasing as the height decreased.
* For some constants, the change in RMSE in relation to Height was found to be linear. For others, it was found to be quadratic.
  + Note : this could be due to the approximation error of the scipy function used
* It was also found that Each constant is crucial to getting a precise answer as keeping any constant 0 would increase the error substantially
* Whenever a constant was tweaked, it would increase the error generated when compared to the Runge-Kutta (R.K) method, which somewhat shows evidence of the R.K method’s constants being tweaked to minimize the error for most equations possible.
* For me, it was found that keeping the difference between the thetas and weights minimized the error and changing the constants resulted in an increase in RMSE

Appendix

import matplotlib.pyplot as plt

from scipy.integrate import odeint

import numpy as np

import math

HEIGHT = 1  # Original Set Height

LOWERLIMIT = 0

UPPERLIMIT = 87 # a = last two digits of my ERP = 87

INITIAL\_Y = 87/1.23

constants = [1,2,3,4]

theta = [0.25,0.5,0.75]

weights = [0.25,0.5,0.75]

INITIAL\_X = LOWERLIMIT

def ComputeDerivitive(y,x):

    dydt = (87\*x + 1.23\*y)/(1.23\*x + 87\*y)

    return dydt

def CorrectAnswer(x):

    t = [0,x]

    y = odeint(ComputeDerivitive,INITIAL\_Y,t)

    return y[1]

def computeAbsoluteSquareError(x,y):

    correct\_answer = CorrectAnswer(x)

    return (y - correct\_answer) \*\* 2

def constant\_sum(constants):

    sum = 0

    for i in constants:

        sum += i

    return sum

def RK\_Method(h, initial\_y, initial\_x):

    error = 0

    y = initial\_y

    x = initial\_x

    constant =  constant\_sum(constants)

    for i in range(int(UPPERLIMIT/h)):

        k1 = h \* ComputeDerivitive(y,x)

        k2 = h \* ComputeDerivitive(y + k1 \* theta[0],x + h \* weights[0])

        k3 = h \* ComputeDerivitive(y + k2 \* theta[1], x + h \* weights[1])

        k4 = h \* ComputeDerivitive(y + k3 \* theta[2], x + h \* weights[2])

        y = y + ((constants[0] \* k1 + constants[1] \* k2 + constants[2] \* k3 + constants[3] \* k4) / constant)

        x += h

        #print(computeAbsoluteSquareError(x,y))

        error += computeAbsoluteSquareError(x,y)

    return math.sqrt((error)/(UPPERLIMIT/h))

def main():

    errors = []

    heights = []

    for i in range(0,5):

        errors.append(RK\_Method(HEIGHT/(2 \*\* i), INITIAL\_Y, INITIAL\_X))

        print(" RMSE = ", errors[-1], ", at Height = ", HEIGHT / 2 \*\* i)

        heights.append(HEIGHT/(2 \*\* i))

    print(heights)

    plt.plot(heights, errors)

    plt.xlabel("Height")

    plt.ylabel("RMSE")

    plt.show()

main()