ALGONQUIN COLLEGE

SCHOOL OF ADVANCED TECHNOLOGY

ICT - Applications & Programming
Computer Engineering Technology – Computing Science

Numerical Computing – CST8233

Term: Summer 2024

A. Kadri, 2024S

Assignment #2 – Menus and Non-Linear Regression and ODE

In this assignment, you will use menus to prompt users with different options. You will also use non-linear regression to find the best-fit of some data and use this to extrapolate new data.

Objectives

- Create a menu system to prompt users with different options,
- Use regression to find the best fit of some data, and
- Use that fit to extrapolate new data.

Grades:

8% of your final course mark

Deadline

August 9th, 2024, 11:59 PM

Question 1

Important Note

Create only one script file for this assignment. To be able to test the code, I should be able to execute the "`bestFitFun()`" function, which for this assignment serves as the equivalent of "`main()`" in other languages. You may create other functions if you wish, but we will only use that one function to launch your code for marking.

Tasks

You are given a file called "`rocket.xlsx`" that contains data on the distance travelled by a rocket launched to the moon. The distance is measured in meters every ten seconds. Write a program using R to do the following:

- Prompts the user to enter a file name. The program must verify that this file exists in the same directory. If it does not, report an error to the user and have them try again.

- Read that data in your script, and save it in a data frame named as "`rocket_data`".

- Your code must present two menus to the user:

- First menu: asks if the user would like to perform non-linear regression to find the best fit, or to quit.
- If the user chooses to find the best fit, then the program will use both the power and exponential models to find the best fit.
  Hint: the program must find the constants of each model and then calculate the sum of the square of the residuals and determine the best model based on the value of each model.

- The program must show the results of each model and state the best model that fits the data.

- Second menu: will ask the user if they wish to extrapolate using the best fit model. If the user chooses to extrapolate, the program will prompt the user to enter a value to which they would like to extrapolate. The program will show the extrapolated value.

• The sequence of your program is illustrated below:

```
MENU
 1. Best Fit
 2. Quit
```

If the user selects 1, they should see this:

```
Please enter the name of the file to open:
```

The user will enter "`rocket.xslx`", at which point your code will load the data using the file name provided by the user. If the user selects 2, the program exists. If the user enters a wrong file name, then the program should show this message:

```
File does not exist, please enter the name of the file to
open:
```

Once loaded, view the data frame, i.e., `rocket_data`. Then, show the results of both fittings, the power and exponential models as below. Please note that the program should show the values of all constants and the Sr for each equation.

```
Power Model:
d = a * t^b
Sr = …

Exponential Model:
D = a * e^(b*t)
Sr = …

The best fit model is … model.
```

Once the best fit model is found, the user will be prompted with the second menu as follows:

```
MENU
 1. Extrapolation
 2. Main Menu
```

If the user selects 2, go back one menu. If the user selects 1, the code will ask the user what time they would like to extrapolate to:

```
Please enter the time to extrapolate to:
```

Calculate the estimated travelled distance and print the result.

- Plot the original data given in the excel sheet along with the best-fit function on the same graph. You must use the right titles for the graph, x-axis (Time-sec) and y-axis (Distance-meter).

- Finally, save the plot as `best_fit.pdf` in the current directory.

Question 2

Important Note

Create only one script file for this assignment. To be able to test the code, I should be able to execute the "`ODEsolver)`" function, which for this assignment serves as the equivalent of "`main()`" in other languages.

You may create other functions if you wish, but we will only use that one function to launch your code for marking.

Tasks

For the thin, glass-walled mercury thermometer system shown in Figure 1, assume that the temperature of the bath changes based on certain chemical process occurring between two substances reacting with each other inside the bath. It is found that the equation that describes this process is given as follows:

$$\frac{d\theta(t)}{dt} + 2\theta(t) = \cos 4t$$

It can be found that the actual solution of the response of the thermometer, $\theta(t)$, is given by the following equation:

$$\theta(t) = 0.1\cos 4t + 0.2\sin 4t + 2.9\, e^{-2t}$$

The ODE given above can be solved using many numerical methods, such as Euler's and Runge-Kutta 2nd Order Methods.

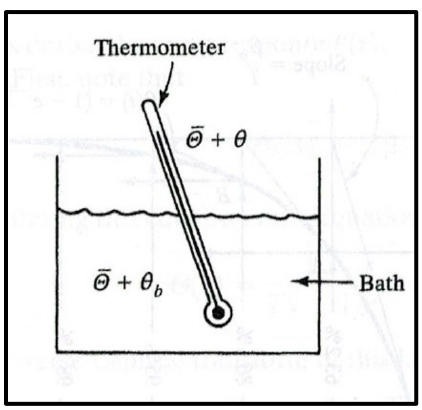

Figure 1 Thin, glass-walled mercury thermometer system

1. Write an R function that computes the solution $\theta(t)$ using Euler's Method. For this step, use the following information: $h = 0.8, 0.2, 0.05$, $\theta_0 = 3$ ℃, $0 \leq t \leq 2$ second. Find the discrete values of $\theta(t)$ at each h step value.

2. Write an R function that uses Runge-Kutta 4th method to solve the same ODE using the following information: $h = 0.8, 0.2, 0.05$, $\theta_0 = 3$ ℃, $0 \leq t \leq 2$ second. Find the discrete values of $\theta(t)$ at each h step value.

3. Calculate the relative error of the resultant solution at each time for each h step. Your output of your code should show a table that shows the exact temperature, the estimated temperature, and the relative error. The user will choose one method and one step size.

Example Output

The output of the code should look like below. The results of test case when using Euler's and Runge-Kutta for $h = 0.2$ are shown in the table below.

```
>> Choose the method for solving the ODE:
1. Euler's Method
2. Runge-Kutta 4th Order Method

>> 1
>> Choose step size "h" (0.8, 0.2, 0.05)
>> 0.2

Time(second)   Exact Temp(C)  Estimated Temp(C)   Percentage
Error(%)
0.2            2.157          2.000               7.28
0.4            1.500          1.339               10.71
0.6            0.935          0.798               14.66
0.8            0.474          0.331               30.13
1.0            0.176          -0.001              100.54
1.2            0.073          -0.131              280.86
1.4            0.128          -0.061              148.01
1.6            0.241          0.118               50.86
1.8            0.299          0.270               9.76
2.0            0.236          0.283               19.89

>> Choose the method for solving the ODE:
1. Euler's Method
2. Runge-Kutta 4th Order Method
```

```
>> 2
>> Choose step size "h" (0.8, 0.2, 0.05)
>> 0.2

Time(second)   Exact Temp(C)  Estimated Temp(C)   Percentage
Error(%)
0.2            2.157          2.157               0.01
0.4            1.500          1.500               0.02
0.6            0.935          0.935               0.03
0.8            0.474          0.474               0.06
1.0            0.176          0.176               0.15
1.2            0.073          0.073               0.34
1.4            0.128          0.128               0.15
1.6            0.241          0.241               0.05
1.8            0.299          0.299               0.02
2.0            0.236          0.236               0.00

>> Choose the method for solving the ODE:
1. Euler's Method
2. Runge-Kutta 4th Order Method
```