1 LabB series on Numerical Analysis and its applications: Lab 1B: Iteration and error (version 1) Due Sun Jan 28, 11:59 PM, 2024

Lab1B, Lab2B, Lab3B, Lab4B will be a series of Chapters about Numerical Analysis and its applications. Lab1B will be an introduction to the series and will focus on iteration and error.

1.1 Title, Team Members and Abstract

In your Lab1B you can use the above title *Iteration and error* or alter the title if you have some thing that suits your Lab1B better. Below the title you should state the names and ID of team members.

You should provide a 1/4 page abstract that serves as an abstract for Lab1B, as well as mentioning it is an abstract for the series of 4 Labs. I know that sounds unfair, but this is the way all writers start. You can make a reasonable guess on what might happen in the series .. such as you will introduce and explore numerical analysis and its applications. The abstract should be mostly verbal and contain few if any equations and be accessible to a wide audience. Typically most people make a sketch of their abstract and revise it as the rest of the work is completed.

1.2 Introduction

In Lab1B (and the other LabB's) you will be asked to provide a 3/4 page Introduction that serves as an Introduction for Lab1B, as well as mentioning it is part of the series of 4 Labs. Again you can make a reasonable guess on what might happen in the series .. such as you will introduce and explore numerical analysis and its applications. The abstract should be mostly verbal and some (but not too many) equations and be accessible to a wide audience. Typically most people make a sketch of their Introduction and revise it as the rest of the work is completed.

I would also like you to mention some background and historical references. For example what is the historical motivation (it could be ancient e.g. Babylonian history) or more recent 20-th century history. Provide a couple of references which are put in the Reference section of your Lab1B.

1.3 Particular aspects of this Lab

In this lab you will investigate the effect of rounding and truncation errors in computing some simple expressions. A helpful tip: If you keep a word-processor file open and paste in results, plots, and explanations as you go along it will save you a lot of time getting the report together.

1.4 Taylor series and error

Recall that the truncated Taylor series of a function $f(x)$ about a point x_0 is given by:

$$T_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k \tag{1}$$

with remainder term $R_n = \frac{f^{(n+1)}(z)}{(n+1)!}(x - x_0)^{n+1}$ for some z depending on n between x and x_0.

Problem 0

For $f(x) = e^{-x}$ show that the task of evaluating (1) about $x_0 = 0$ yields

$$T_n(x) = \sum_{k=0}^{n} \frac{(-1)^k}{k!}x^k \tag{2}$$

and find the form of the remainder (truncation error) term.

Problem 1

(a) Find the Taylor series with $n + 1$ terms and the associated truncation error for

$$F(x) = \frac{e^{-x} - 1 + x}{x^2}. \tag{3}$$

by substituting the series (2) for e^{-x} into $F(x)$. *Don't use the general formula of Taylor series for $F(x)$.* Your expression for the Taylor series should be general, similar to the series for e^{-x} (see Lab1A for examples of such computations). For numerical evaluation it is useful to simplify the $F(x)$ after substituting the first few terms of $e^{-x} = 1 - x + x^2/2 + \cdots$.

(b) Construct an expression that bounds the truncation error, assuming $n > 2$, for a given value of x. See especially the review material and examples related to Taylor's series with remainder in Section 0.5 of the Text by Sauer.

Problem 2

(a) Modify an M-file for Taylor Series from Lab1A (e.g. for $\exp(x)$ without using Horner) to compute and plot the Taylor series and $F(x)$ from problem 1 above. (an M-file is expected for this part).
A couple of implementation hints that may be useful:
(i) Note that the loop does not have to start with 1.
(ii) Note that the first term in the series is now $1/2$ rather than 0 so you will need to change the initialization. A useful function to do this is the Matlab `ones` function which you can multiply by 0.5 to get the appropriate initialization.
(iii) When calculating $F(x)$ with a vector argument x instead of a scalar, you need to use the vector operators for division ./ and powers .^ (i.e. with the . modifying the normal operator).

(b) Plot a graph of $F(x)$ and also its Taylor series with 4 terms (up to the cubic term) using 50 points over the interval $(-0.01, 0, 01)$. If you have done this correctly, the curves should be essentially indistinguishable.

Problem 3

(a) Change the range of your plot in problem 2b above to $(-10^{-7}, +10^{-7})$. The two curves should now look very different.

(b) Change the script so that it plots the absolute and relative errors (two different plots, one for absolute error, one for relative error)related to the difference between the Taylor series and $F(x)$ over the interval $(-10^{-6}, +10^{-6})$. Make sure you use the Matllab `abs` function to obtain the absolute value of the error. Use a log scale on the $y-$axis.

(c) To identify the source of the errors plotted in (b) it is useful to compare to a reasonable bound for the truncation error expected for the Taylor series. Come up with a such a bound and plot its magnitude on the same log-linear plot from (b). You should have a different bound for $x < 0$ and $x > 0$. For $x > 0$ you can use the result mentioned in class for error of an alternating series. The case $x < 0$ is much harder.

(d) Consider the results from (c). From this you should be able to conclude that the problem is roundoff error rather than truncation error. **Why?** Explain what is making the roundoff errors so large and give an argument as to whether the Taylor series or the directly computed function is more accurate.

Problem 4 In this problem we want to create a matlab function that can evaluate $F(x)$ to a given accuracy for any value of x.

(a) Create a new m-File, called `myfofx.m`. Unlike our previous script files, we would like this to be a function, just like $\sin(x)$ that returns a number. i.e. we would like to be able to use the function in commands like:

```
>> y=myfofx(0.001)
```

or

```
>> plot(x,myfofx(x))
```

To do this, we need to put the line

```
function [fx]=myfofx(x)
```

as the first non-commented line in the m-file. Then, `fx` will be the variable that we will use in the m-file that, at the end of the calculation, contains our answer. Note that unlike previous cases where `x` and `T` were scalars you will not need to set the value of x as it will be passed to the function. In this case, just return the value of the $n = 8$ Taylor series (i.e. replace the `fprintf` statement with `fx=T` before the `return`). Test the function by evaluating the result on the command line for `myfofx(1)` and `myfofx(2)`. Make sure you are getting the correct result before proceeding.

(b) If we are close to $x = 0$ our previous results suggest that it would be better to use the Taylor series. However, if we are far away from $x = 0$, the accuracy of our Taylor series starts to degrade due to truncation error. Use an `if-else` construct to modify your function to use the Taylor series when $-10^{-6} < x < +10^{-6}$ and the explicit expression for $F(x)$ outside of this range. If you are unsure of how to use the `if-else` construct described in the matlab tutorial, look it up in the matlab help. In particular, the examples given there often very helpful. Other matlab functions and statements that may, or may not, be useful is the absolute value function `abs` and the `OR` statement.

(c) Demonstrate that your function works by plotting its result over the same ranges of x used in problems 2 and 3.

Summary and Conclusions

This section should not be packed with equations, but instead verbal descriptions.

In this section you will briefly summarize your results and what you did in this lab.

You should also indicate how the method could be applied to other problems and make suggestions for future work.

References

Include some references.

Teamwork Statement

This shoud be a paragraph, outlining current team work problems and ideas.

Code Appendix

Plot Appendix