HOMEWORK 1

Jacob Cunningham[1]

Monday 20th January, 2025

Abstract

Homework discussion, calculations, and answers are provided herein.

Keywords Numerical Methods, Mechanical Engineering

1 Effects of roundoff and truncation errors on numerical accuracy

The one-sided finite difference scheme to approximate the first derivative of a function f is defined as:

$$f_h(x) = \frac{f(x+h) - f(x)}{h} \approx f'(x), \tag{1}$$

where h is the step size if no roundoff error exists, then the accuracy of the scheme is dermined soley by **truncation error**

$$t_e(h) = |f'(x) - f_h(x)| = O(h), \quad f'(x) = f_h(x) + O(h), \tag{2}$$

where the big O - notation means there exists a constant K such that $O < K \cdot h$ for all h. In the presence of roundoff errors, x cannot be represented exactly; instead, it is represented by the rounded value $\tilde{x}$ with the associated roundoff error $r = |\tilde{x} - x|$.

1.1 Part A

Show that the total error of the finite-difference approximation consists of both truncation error $t_e(h)$ due to the finite-difference scheme and the roundoff error r:

$$\epsilon(h) := |f'(x) - f_h(\tilde{x})| = O(h) + \frac{r}{h} \tag{3}$$

Hint: Start from $f'(x) = f_h(x) + O(h)$, and consider that the computation of the finite difference $f_h(x)$ approximation already involves roundoff errors, e.g., $f(x) = \tilde{f}(x) + r$, $f(x+h) = \tilde{f}(x) + h + r$.

> **Answer 1A**
> Answer goes here.

1.2 Part B

From Part A, what can you say about the numerical accuracy of the finite difference scheme as the step size h is continually decreased?

---

[1]Correspondence to: jjc132@pitt.edu

> **Answer 1B**
> Answer goes here.

1.3 Part C

The second order central difference approximation is defined as

$$f_h^c(x) = \frac{f(x + h - f(x - h))}{2h} \approx f'(x) \tag{4}$$

and has a truncation error of $t_e(h) = |f'(x) - f_h^c(x)| = O(h^2)$.

Within the notebook `Week2_FD_students.ipynb` available on Canvas Module 1, add a function to evaluate the finite difference approximation f_h^c for the derivative of $f(x) = \sin(x)$ with step size h at a fixed $x = x_0 = 1$. For the same array h of step sizes as in the notebook, calculate the array consisting of errors between the finite difference approximation f_h^c and the exact derivative f'. Plot the approximation error for the central differences scheme as a function of step size h. Display your plot in log format in addition tot he previous plot for forward differences.

> **Answer 1C**
> Answer goes here.

1.4 Part D

Now repeat the same above steps for the function $g(x) = \sin(100x)$ and its derivative $g'(x)$. Add your plots for the forward difference and central difference numerical errors to the same plot. Make sure to use a different linestyle, and specify your legend entries. Looking at your plot, what can you say about the accuracy of each method to approximate the derivative for $\sin(x)$ and $\sin(100x)$ at $x_0 = 1$? Can you explain why?

> **Answer 1D**
> Answer goes here.

2 Errors in Scientific Computing

The sine function is given by the infinite series

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \tag{5}$$

2.1 Part A

Calculate the forward and backward errors if we approximate the sine function by using only the first term in the series for $x = 0.1$, 0.5, 1.0.

> **Answer 2A**
> Answer goes here.

2.2 Part B

Calculate the forward and backward errors if we approximate the sine function by using only the first two terms in the series for $x = 0.1$, 0.5, 1.0.

> **Answer 2B**
> Answer goes here.

3 Condition Number & Stability

This is Exercise 1 from Section 1.4, page 26 of Driscoll and Braun [2018]. Exercises are also available at the end of each section of a chapter in the online textbook. Refer to your textbook for cross-referenced equations and tables.

Consider the formulas

$$f(x) = \frac{1 - \cos x}{\sin x}, \quad g(x) = \frac{2\sin^2 x/2}{\sin x}, \tag{6}$$

which are mathematically equivalent, but they suggest evaluation alrogirthms that can behave quite differently in floating point arithmetic.

3.1 Part A

Using (1.2.6), find the relative condition number of f. Note that because f & g are mathematically equivalent, their condition numbers are the same. Show that the condition number approaches to 1 as $x \to 0$, which means it is possible to compute the function accurately near zero.

> **Answer 3A**
> Answer goes here.

3.2 Part B

Compute $f(10^{-6})$ using a sequence of four elementary operations. Using Table 1.1 on page 13, make a table like the one shown in Demo 1.4.1 in the book that shows the result of each elementary result and the numerical value of the condition number of that step.

> **Answer 3B**
> Answer goes here.

3.3 Part C

Repeat Part B for $g(10^{-6})$, which has six elementary steps.

> **Answer 3C**
> Answer goes here.

3.4 Part D

Based on your answers to Part B & C, is $f(10^{-6})$ or $g(10^{-6})$ more accurate (provide your answer as Markdown text in your Jupyter notebook)?

> **Answer 3D**
> Answer goes here.

4 Floating Point Arithmetic

Create an example calculation to demonstrate that floating point addition is not associative. Repeat the exercise for floating point multiplication.

Answer 4
Answer goes here.

Original article

This article is available online at the following URL: https://jacob-cunningham-ds.github.io/me2060/hw1-code-cunningham

References

T. A. Driscoll and R. J. Braun. *Fundamentals of Numerical Computation*. 2018.