Lesson 8 : Introduction to Numerical Analysis

Now that we have a working understanding of IEEE 754 Format, we move onto our introduction to the field of Numerical Analysis. We will borrow Lloyd Trefethen's definition of this field:

> "Numerical analysis is the study of algorithms for the problems of continuous mathematics".*

* Note: This definition is quite sophisticated and originates from deep thought on the part of Trefethen and many other scholars who've worked in this field. I highly encourage you to read through Trefethen's 5 page article "The Definition of Numerical Analysis." I've provided a link to this article on our homepage.

Let's focus our attention on the pivotal words and ideas in this definition. With this in mind, we see that numerical analysis involves:

- the study of $\boxed{\text{algorithms}}$

A. One of the central jobs of a numerical analyst is to $\boxed{\text{devise algorithms}}$ to solve specific, applied problems that arise in science, engineering, and mathematics.

B. hidden between the spaces in the phrase "devising algorithms to solve problems" is the fact that the algorithms a numerical analyst produces should actually solve the problem(s) being considered. Thus, a numerical analyst should be able to verify and certify that the solution produced by her algorithm is indeed viable solutions to the original problem. With this in mind, a numerical analyst must also be quite skilled at $\boxed{\text{analyzing algorithms}}$ in order to verify that her solution(s) actually make sense in the context of her original problem.

Numerical analysis involves:

☐ Problems of $\boxed{\text{continuous mathematics}}$

A. The word "continuous" signifies that real or complex variable are involved in the problem. In other words, in order to get an exact solution to our problem, we need to describe fundamental quantities or operations on those quantities using type II numbers in $\mathbb{R}$ (or $\mathbb{C}$).

B. Remember, though, that type II numbers cannot be represented exactly on a finite-memory, binary computer. Thus, a very important part of a numerical analyst business of designing algorithms is to accurately approximate all type II numbers using type I data, performing operations on type I data, yield a type I result, and certifying that the type I output of the algorithm accurately approximates some type II solution to the original problem.

Numerical Analysis involves:

☐ Problems of $\boxed{\text{continuous mathematics}}$, continued...

C. Continuous variables in mathematics are the opposite of discrete variables. Continuity and the real number system require the sophisticated concepts of limits and infinite decimal expansions to represent most $x \in \mathbb{R}$ exactly. In the continuous paradigm, we can guarantee to produce an approximation $\hat{x}$ of any $x \in \mathbb{R}$ to an arbitrarily small level of accuracy $\varepsilon > 0$ that we desire with

$$|x - \hat{x}| < \varepsilon.$$

We even have the option to for $\hat{x} = x$ "exactly" if desired. This represents a fantasy for an applied mathematician and a simple axiom for pure mathematicians. Numerical analysts develop algorithms to solve continuous problems on finite-memory binary computers.

D. Discrete variables in mathematics are countable (in a finite amount of time) and exist in bit-size pieces. For example, canonical counting problems involving subsets of $\mathbb{N}$ or $\mathbb{Z}$ are great introductions to discrete variables since each element of $\mathbb{Z}$ is easily distinguished from it's closest neighbor and can be written exactly without the use of limits. Algorithms for discrete problems are the focus of other computer scientists outside the field of numerical analysis.

Numerical Analysis involves:

- ☐ work that is $\boxed{\boxed{\text{applied}}}$

A. All of the work of numerical analysis is applied to solve problems arising in science, engineering, and mathematics. This work is applied daily and successfully to thousands of application areas on millions of binary computers around the world.

B. "Problems of continuous mathematics" are the problems that science and engineering are built upon;

> * without numerical methods, science and engineering as practiced today would grind to a halt quite quickly.

* This observation (highlighted in red above) is a fundamental reason why this ENGR 11 course and Math 2B are pre-requisites for so many STEM majors... If you plan to work in this space, the tools of numerical analysis will very likely be part of your future

Numerical Analysis involves:

☐ $\boxed{\text{Approximating unknowns}}$

A. most problems of continuous mathematics CANNOT be solved exactly by finite algorithms.*
This rather grim reality would hold true even if we were able to work in exact arithmetic (which we can't do on a binary computer).

B. Our first introduction to approximating unknown quantities was our study of IEEE 754 Floating-point numbers. This topic and a related discussion of floating-point arithmetic, is focused on the simplest of approximation problems:

how can we approximate unknown type II numbers $x \in \mathbb{R}$ accurately and efficiently on a binary computer. Compared to the larger field of numerical analysis, floating-point format/arithmetic is a "small and rather tedius" subtopic of approximations that sets the stage for a much larger study.

* contrast this with the fact discrete problems in computer science can be solved exactly by finite algorithms on binary computers

Numerical Analysis involves:

☐ $\boxed{\text{Approximating unknowns}}$ continued...

c. Not only do numerical analysts want to design algorithms that work on binary computation machines with finite memory and produce output that verifiably (and approximately) solves important applied problems, but we also want to create algorithms that lead to rapid "convergence" of approximation.

At this stage in history, we've had 80+ years (1945 - present day) of Research & Development into this field by some very dedicated thinkers.

In fact, a central pride of the field of numerical analysis is that for many applied problems, our predecessors have invented clever algorithms that converge to desired approximate output exceedingly fast.

Numerical Analysis involves:

☐ $\boxed{\text{Analysis of accuracy and stability}}$ of numerical algorithms *

A. We remember again that most problems of continuous mathematics CANNOT be solved exactly by finite algorithms.

Numerical analysts are painfully aware of this fact and yet we dedicate ourselves to developing algorithms to approximate solutions to problems of continuous math.

While designing algorithms, we care about a number of important features of our approximations including:

i. the motivations behind our problem

ii. the algorithmic derivations that lead to our algorithm

iii. the effects of finite-precision arithmetic within our algorithm

iv. the efficiency, speed, and convergence properties of our algorithm.

It is within this framework that the very traditional topic of error analysis fits.

* A very famous and incredibly insightful textbook entitled "Accuracy and Stability of Numerical Algorithms, 2E" by Nicholas J. Higham gives a state-of-the-art overview of this subtopic of numerical analysis and is quite useful as a reference for professional numerical analysts.

Let's start to develop an algorithm to compute approximations for

$$f(x) = \sqrt{x}$$

for $x \in \mathbb{R}$. Notice, there are actually two approximations happening

Start with $x \in \mathbb{R}$ (infinite precision, type II number

$\xrightarrow[\text{IEEE 754}]{\text{approximate w/}}$

transformed into $\hat{x} \in \mathbb{Q}_I$ (finite precision, type I number)

$\downarrow$

OBSTACLE

algorithm to produce $\hat{y} = \sqrt{\hat{x}}$ $\downarrow$

Calculate $\hat{y} \in \mathbb{Q}_I$ to approx $\sqrt{\hat{x}}$ (finite precision using IEEE 754 Arithmetic

$\xleftarrow{\text{verify \& certify}}$

End with $\sqrt{\hat{x}} = \hat{y} \approx y = \sqrt{x}$