MCEN 3030

23 Jan 2024

Last time:
- Pseudocode
- Writing functions with loops

Today:
- Function Practice
- Numerical Error
- Debugging/ Debug Mode

A few comments on functions

1) To "call" a function from the command window, the function.m must be saved in the "working directory" & the function file must begin with

function [out1, out2] = fxn_name(in1, in2)

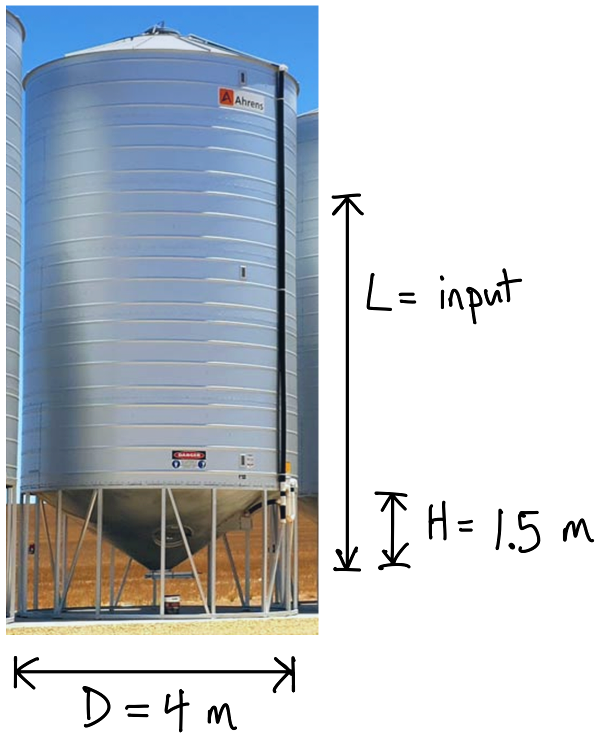
2) Functions should have, as their last line,

end

3) Functions can be embedded inside scripts or other functions, but can then only be called from within that script/function.

4) Functions can also call other functions in the "working directory".

Let's write, from scratch, a function that contains a function in it.



From an input height L, determine the volume of grain in the silo

Numerical Error

Big picture: We are trying to model reality, and must ask the question: How accurately do these computational models portray the real physics?

- How trustworthy are our calculations?

In regards to modeling reality, we should pay attention to significant digits —

320 N $\longrightarrow$ two significant digits

320.2 N $\longrightarrow$ four

320.20 N $\longrightarrow$ five

$\longrightarrow$ Mechanical engineering measurements are probably not going to have 5 sig figs. Maybe 4, probably 3. So we need our computations to produce similar accuracy.

Definitions of error

Generic definition (mathematical)

$$E = (\text{true value}) - (\text{approximate value})$$

analytical value
experimental value
validation value

computed value

An often more meaningful answer:

Relative Error

$$e = \frac{(\text{true value}) - (\text{approximate value})}{(\text{true value})}$$

(×100 if you want a %)

In some cases, it is not known what the "true value" is.

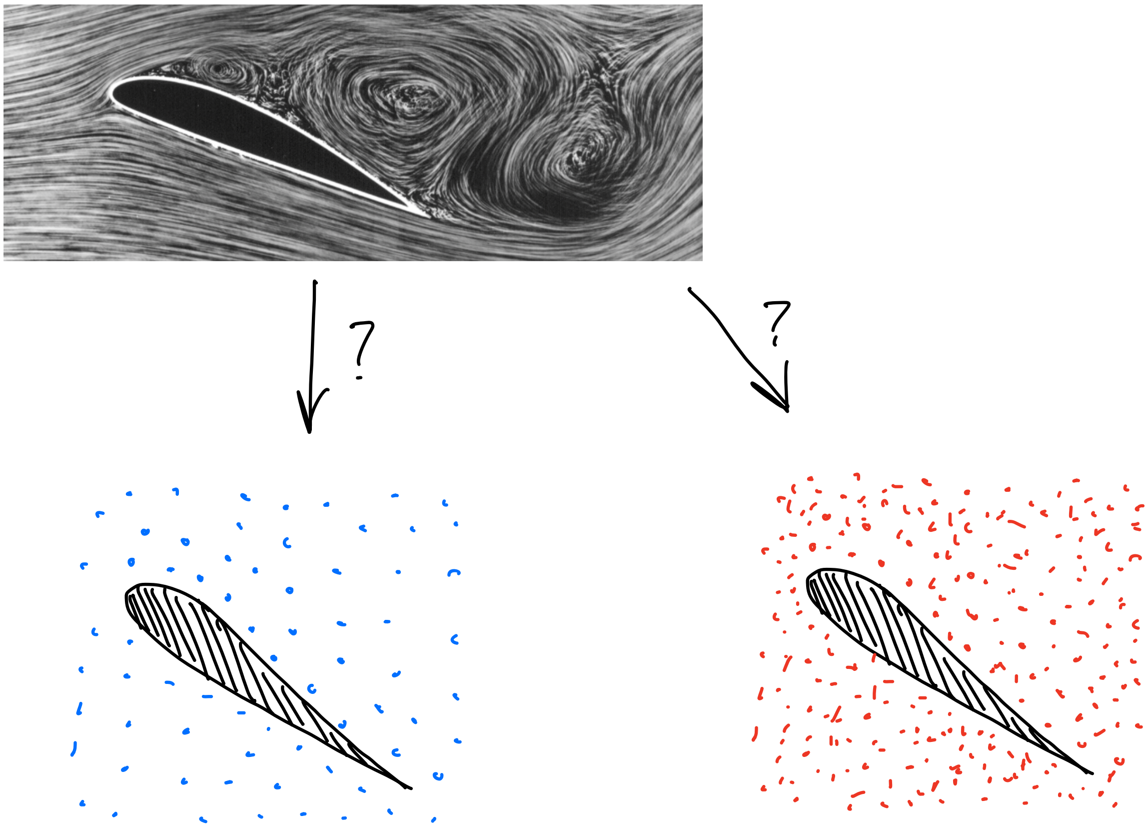
In such a case, we might define our numerical accuracy based on the change in output value over successive iterations

$$\epsilon = \frac{(\text{new value}) - (\text{old value})}{(\text{new value})} \qquad \left(\times 100 \text{ to make it a } \%\right)$$

Criterion for "convergence" on a (hopefully good) solution

$$|\epsilon| < \epsilon_{\text{acceptable}}$$

Trade-off: time & error



More points → better, more detailed answer.

But computational cost typically scales up more than linearly.

How do we know the acceptable amount of error?

- With a validation case / experiment, get to the same number of sig figs:
    - One standard: relative error of

From Chapra → $(0.5 \times 10^{2-n})\%$ where n = # of sig figs

- Diminishing returns: one more iteration changes result by 0.01%? Probably OK to call it good.