1. Nested Arithmetic

Evaluating a function is a common step in many numerical algorithms. The most kind of function is polynomials as many more complicated functions in fact use their Taylor expansions as their representations, which are after all polynomials. Polynomials of order k are of the form

$$p_k(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_k x^k.$$

Evaluating powers of a number thus has become particularly important. We don't want to lose significant digits there (or not lose too many).

Example. Suppose $x = 4.71$. Evaluate the polynomial $p_3(x) = 1.5 + 3.2x - 6.1x^2 + x^3$ at this point.

We first form a table of straightforward computation (naive computation).

	x	x^2	x^3	$6.1x^2$	$3.2x$
exact	4.71	22.1841	104.487111	135.32301	15.072
3-digit chopping	4.71	22.1	104	134	15.0
3-digit rounding	4.71	22.2	105	135	15.1

Firstly, we know $x^2 = 4.71^2 = 22.1841$. This rounds to 22.2 if we use three-digit rounding. Then,

$$\begin{aligned} fl\left(x^3\right) &= x \otimes x \otimes x \\ &= (x \otimes x) \otimes x \\ &= fl\left(fl(x) \otimes fl(x)\right) \otimes x \\ &= fl\left(4.71^2\right) \otimes x \\ &= 22.2 \otimes x \\ &= fl\left(fl(22.2) \times fl(x)\right) \\ &= fl(22.2 \cdot 4.71) \\ &= fl(135.42) \\ &= 135 \end{aligned}$$

which then rounds to 105. Similarly, $6.1x^2 = 135.42$ and rounds to 135, and $3.2x = 15.072$ and rounds to 15.1.

The exact result is

$$p_3(4.71) = -14.263899.$$

With three-digit chopping, we have

$$f(4.71) = ((104 - 134) + 15.0) + 1.5 = -13.5, \implies Error_{rel} = \left|\frac{-14.263899 + 13.5}{-14.263899}\right| \approx 0.05$$

and with three-digit rounding, we have

$$f(4.71) = ((105 - 135) + 15.1) + 1.5 = -13.4, \implies Error_{rel} = \left|\frac{-14.263899 + 13.4}{-14.263899}\right| \approx 0.06$$

(Check these!). Both rounding yield considerable relative error.

The root of the problem lies in the number of arithmetic computations performed by naive/direct computation. Let us count the number of floating-point operations (flops, in short). In the polynomial $p_3(x) = x^3 - 6.1x^2 + 3.2x + 1.5$, we have

	$+/-$	$\times/\div$
x^3	1	2
$6.1x^2$	1	1
$3.2x$	1	1

which totals 7 flops – note in $6.1x^2$, we already know x^2 from computing x^3, so the only multiplication is $6.1 \times x^2$. To reduce this number, we consider a nested formulation of the polynomial.

$$\begin{aligned} p_3(x) &= x^3 - 6.1x^2 + 3.2x + 1.5 \\ &= \left(x^2 - 6.1x + 3.2\right) x + 1.5 \\ &= \left((x - 6.1)\, x + 3.2\right) x + 1.5 \end{aligned}$$

Now, we still have three $+/-$, but the number of multiplication reduces to just two times. With this formulation, we only incur a total of 5 flops.

Indeed, still using three-digit chopping but now employing the nested polynomial, we have

$$f(4.71) = ((4.71 - 6.1)\, 4.71 + 3.2)\, 4.71 + 1.5 = -14.2.$$

We observe that we already hit three significant digits. The relative error of this calculation is

$$Error_{rel} = \left| \frac{-14.263899 + 14.2}{-14.263899} \right| \approx 0.0045,$$

far better than direct computation.

Remark. Moral of the story: you **always** put polynomials in nested form before doing **any** computations/evaluations.

2. Well-conditioned Problem

In calculus, we have learned the notion of continuity, that is, we say a function $f(x)$ is continuous at some point $x = a$ if and only if the following statement is true: for every $\epsilon > 0$, we can find a $\delta > 0$ such that

$$|f(x) - f(a)| < \epsilon$$

if $|x - a| < \delta$.

However, this is awfully abstract. What it really means is that given an input very close to $x = a$ (closeness measured by δ), then the output won't be also so far away from the true output (measured by ϵ).

This definition helps us establish something call **well-conditioned** problems. Suppose we are asked to solve a system of equations in matrix form, i.e.,

$$Ax = b.$$

We say this problem is **well-conditioned** if for every perturbation δA and δb, the solution $\tilde{x}$ to the perturbed problem

$$(A + \delta A)\, \tilde{x} = (b + \delta b)$$

is not too far off from the true solution x. In other words, this problem doesn't go crazy when nudged. Otherwise, we call the problem **ill-conditioned**.