Round-off, truncation and propagation of errors

NC 04 • Module 1: Foundations

Central question	How do computational errors arise and spread through numerical algorithms?
Lecture focus	Finite precision, approximation of infinite processes and propagation of uncertainty.
Prerequisite	Floating-point representation and error measures (NC 02–03).

Learning outcomes

By the end of this lecture, students should be able to...

Distinguish	Identify round-off and truncation errors and explain their sources.
Calculate	Estimate errors caused by rounding, finite series and arithmetic operations.
Analyze	Recognize cancellation, bound propagated errors and select suitable step sizes.

Two major sources of error

Numerical results may differ from their exact mathematical counterparts.

Round-off error	Occurs when a number or arithmetic result is represented with finite precision.
Truncation error	Occurs when an infinite process or exact mathematical operation is replaced by a finite approximation.
Combined effect	A computed approximation can contain both kinds of error, along with errors in input data.

Round-off: chopping and rounding

Consider storing π = 3.14159265... using four significant digits.

Chopping	Discard later digits: 3.141. Absolute error ≈ 0.00059265.
Rounding to nearest	Choose the nearest representable value: 3.142. Absolute error ≈ 0.00040735.
Error bound	For rounding to nearest, absolute error is at most half the spacing between adjacent values.

Floating-point error model

For normal results, assuming rounding to nearest and no overflow or underflow.

Model	$\mathrm{fl}(z) = z(1 + \delta)$, where $\lvert\delta\rvert \le u$ and u is the unit round-off.
Binary64	$u = 2^{-53} \approx 1.11 \times 10^{-16}$; spacing above 1 is 2^{-52}.
Interpretation	The relative error of one correctly rounded operation is small, but many operations can accumulate error.

Worked example: repeated rounding

Use three significant digits and round to nearest after every operation.

First calculation	fl(1.23 + 0.00456) = fl(1.23456) = 1.23.
Second calculation	fl(1.23 + 0.00456) = 1.23 again; both increments disappear.
Comparison	Exact sum = 1.23912; computed sum = 1.23; absolute error = 0.00912.

Truncation error

A finite approximation replaces an exact or infinite mathematical expression.

Example	$e^x = 1 + x + x^2/2! + x^3/3! + \ldots$
Finite approximation	$P_2(x) = 1 + x + x^2/2$. Terms of degree 3 and above are omitted.
Signed truncation error	$R_2(x) = e^x - P_2(x)$. This exists even in exact arithmetic.

Worked example: truncated series

Approximate $e^{0.1}$ using the first three terms of its Taylor series.

Approximation	$P_2(0.1) = 1 + 0.1 + 0.01/2 = 1.105$.
Reference	$e^{0.1} \approx 1.105170185$; absolute truncation error ≈ 0.000170185.
Remainder bound	By Taylor's theorem, $\lvert R_2 \rvert \le e^{0.1}(0.1)^3/6 \approx 0.000184195$.

Order of truncation error

The order describes how fast the approximation error decreases.

Definition	$E(h) = O(h^p)$ means $\lvert E(h)\rvert \leq C\lvert h\rvert^p$ for sufficiently small h, with fixed C.
First order	If the leading error is proportional to h, halving h approximately halves the error.
Second order	If the leading error is proportional to h^2, halving h approximately quarters the error.

Worked example: numerical derivative

Approximate $f'(2)$ for $f(x) = x^2$ using the forward difference.

Formula	$D(h) = [f(2+h) - f(2)]/h = 4 + h$ in exact arithmetic.
Step $h = 0.1$	$D(0.1) = 4.1$; exact derivative = 4; absolute truncation error = 0.1.
Step $h = 0.01$	$D(0.01) = 4.01$; error = 0.01. This is a first-order method.

Propagation of input errors

Small perturbations in input quantities can change the output of a calculation.

Notation	Let $a = x + \Delta x$ and $b = y + \Delta y$ be perturbed input values.
Addition and subtraction	For $z = x \pm y$, $\Delta z = \Delta x \pm \Delta y$, so $\lvert\Delta z\rvert \leq \lvert\Delta x\rvert + \lvert\Delta y\rvert$.
Multiplication	For $z = xy$, $\Delta z = y\Delta x + x\Delta y + \Delta x\Delta y$. The last term is second order.

Worked example: addition and subtraction

Let $x = 20 \pm 0.1$ and $y = 8 \pm 0.2$ denote absolute uncertainty bounds.

Addition	$x + y = 28 \pm 0.3$; possible interval $[27.7, 28.3]$.
Subtraction	$x - y = 12 \pm 0.3$; possible interval $[11.7, 12.3]$.
Interpretation	The absolute uncertainty bound is the same, but the relative uncertainty is larger for the difference.

Worked example: multiplication and division

Let $x = 10 \pm 0.1$ and $y = 5 \pm 0.1$; both quantities are positive.

Product	$xy = 50$ nominally; $\lvert \Delta z \rvert \leq 5(0.1) + 10(0.1) + 0.01 = 1.51$.
Quotient	$x/y = 2$ nominally; exact interval $[9.9/5.1, 10.1/4.9] \approx [1.9412, 2.0612]$.
First-order estimate	Relative uncertainty for product or quotient $\approx 0.1/10 + 0.1/5 = 0.03$ (3%).

Error propagation through functions

Use derivatives to estimate the effect of small perturbations.

One variable	$\Delta f \approx f'(x)\Delta x$, so $\lvert\Delta f\rvert \approx \lvert f'(x)\rvert\,\lvert\Delta x\rvert$.
Several variables	For $z = f(x,y)$, $\Delta z \approx f_x\Delta x + f_y\Delta y$; sum absolute contributions for a first-order bound estimate.
Example	Area $A = \pi r^2$, $r = 10 \pm 0.1$: $\Delta A \approx 2\pi(10)(0.1) = 2\pi$; relative uncertainty $\approx 2\%$.

Catastrophic cancellation

Subtracting nearly equal approximate numbers may amplify relative input errors.

Exact values	$x = 1.234567$, $y = 1.234561$; $x - y = 0.000006$.
Six significant digits	Both round to 1.23457; computed difference = 0, giving 100% relative error.
Avoidance	Reformulate where possible: $\sqrt{(x+1)} - \sqrt{x} = 1/[\sqrt{(x+1)} + \sqrt{x}]$.

Balancing truncation and round-off

Reducing step size indefinitely is not always beneficial in floating-point arithmetic.

Truncation	For a forward difference, the truncation error is typically O(h).
Round-off	Subtraction and division by h can amplify rounding effects; a common estimate is O(u/h).
Practical strategy	Test a range of h values; for a model Ah + Bu/h, the minimum occurs at h = √(Bu/A).

Python demonstration and exercises

Run the derivative experiment for $f(x) = x^2$, then solve the two exercises.

Python	for h in [1e-2, 1e-6, 1e-12, 1e-16]: print(h, ((2+h)**2-4)/h)
Q1	Round 7.3864 to three significant digits; find the absolute round-off error.
Q2	Approximate sin(0.1) by 0.1 (radians); find the absolute truncation error.

Solutions and lecture recap

Next lecture: NC 05 — Taylor Series, Stability and Convergence.

A1	Rounded value = 7.39; absolute round-off error = $\lvert 7.3864 - 7.39 \rvert = 0.0036$.
A2	$\sin(0.1) \approx 0.09983341665$; absolute truncation error ≈ 0.00016658335.
Takeaway	Finite precision causes round-off; finite approximation causes truncation; operations propagate errors.