Round-off errors

Round-off errors occur when you do not have enough digits to store the answer.

ex) 8-bit integer : $\frac{7}{2} = 3$ ← not enough digits to store 3.5

In double precision the precision is 2^{-52}

Roughly 2.22×10^{-16} in base-10

Called machine precision & given by eps in Matlab.

ex) Addition problem w/ a machine precision of $10^{-3} = 0.001$ (you can store 3 sig-figs)

Exact: $0.99 + 0.0044 + 0.0042 = 0.9986$

3-digit w/ rounding

$$(0.99 + 0.0044) + 0.0042 = 0.994 + 0.0042$$
$$= 0.998$$

$$0.99 + (0.0044 + 0.0042) = 0.99 + 0.0086$$
$$= 0.999$$

Frequent in subtraction and division.

Re-arranging the calculation can reduce these errors.

ex) Roots of $x^2 - bx + 1 = 0$, b is large

$$r = \sqrt{b^2 - 4} \qquad x_1 = \frac{b+r}{2} \qquad x_2 = \frac{b-r}{2}$$

If b is large then b & r are close to each other

If $b = 110$ & $r = (110^2 - 4)^{1/2} = 109.9818\ldots$

True answer is $109.99\ldots$ and $0.00909166\ldots$

Now consider a machine with $\varepsilon = 10^{-3}$ (3 sig figs) using chopping

$\Rightarrow b = 110 \qquad r = 109$

actually 109.5

$$x_1 = \frac{b+r}{2} = \frac{(110 + 109)}{2} = \frac{219}{2} = 109$$ ← 3 digits correct

$$x_2 = \frac{b-r}{2} = \frac{(110 - 109)}{2} = \frac{1}{2} = 0.5$$ ← 0 digits correct

but: $$x_2 = \frac{b-r}{2} \frac{b+r}{b+r} = \frac{b^2 - r^2}{2(b+r)} = \frac{4}{2(b+r)} = \frac{2}{b+r}$$

Then $$x_2 = \frac{2}{b+r} = \frac{2}{110+109} = \frac{2}{219} = 0.00913$$ ← one digit & much closer

Why we care: When solving $\underline{A}\,\underline{x} = \underline{b}$ we will decompose $\underline{A}$.

To do this you do repeated operation on $\underline{A}$. These use $+, -, *, /$

Certain algorithms are less susceptible to round-off errors (more stable) & this give better answers.