Errors

[Start with page about double-precision numbers and machine precision!]

- Mathematical errors / truncation errors

  - E.g from stopping a series expansion

$$u''_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \mathcal{O}(h^2)$$

truncation error

- Round-off errors

  - Numbers only stored with accuracy ~ machine precision

  - For double: ~ 15 digits

  - So almost all numbers stored are approximations:

$$fl(x) \approx x$$

  - Can have catastrophic consequences.

Loss of numerical precision

- a.k.a loss of significance

- Typical case : Substract similar numbers

  → loose significant digits , left with digits affected by round-off

- Example :

  True : $a = 1.0054321$

  Approx : $fl(a) = 1.005$

  - 4 significant digits
  - Rel. error in approx : $\left|\frac{a - fl(a)}{a}\right| \approx 4 \times 10^{-4}$

  &

  $b = 1.0040001$

  $fl(b) = 1.004$

  - 4 sig. digits
  - Rel. err ; $\left|\frac{b - fl(b)}{b}\right| \approx 1 \times 10^{-7}$

  $fl(a)$ and $fl(b)$ are good approx. to a and b.

  Take difference :

  True : $a - b = 0.0014320$

  Appvox : $fl(a) - fl(b) = 0.001$

  - Only 1 sign. digit !
  - Rel error : $\approx 3 \times 10^{-1}$ 30% !

⇒ Loss of precision !

- We are typically interested in <u>relative</u> errors

abs. err : $\Delta = |v_i - u_i|$

rel. err : $\varepsilon = \left| \frac{v_i - u_i}{u_i} \right|$ will study this in proj. 7

[Often look at $\log_{10}(\varepsilon)$ vs $\log_{10}(h)$]

- Typical case for us

  - For "large" step sizes : truncation error dominates
  - For tiny step sizes : round-off errors lead to loss of precision → garbage

<u>Some optimal stepsize gives smallest overall error</u>

- I will put out a code example for this
- You will study this in proj. 7