Lecture 1:

A first numerical example:

Radioactive decays

Learning outcomes of course

- Scientific computing for complex (=realistic) problems
- Basic numerical methods and their implementation
  - solving differential equations
  - root finding: solutions for $f(x) = 0$
  - numerical integration
  - Monte Carlo methods & simulation
- Assignments in Python notebooks

https://dmaitre.phyip3.dur.ac.uk/notebooks/nm/

Material

- The course is entirely based on
  Giordano & Nakanishi: "Computational physics"
- Course homepage at

https://www.ippp.dur.ac.uk/~krauss/Lectures/NumericalMethods/index.html

Scientific computing:

- Start with a physical system/phenomenon, typically isolated, self-contained $\Longrightarrow$ idealised
- Formulate a mathematical model, for example pendulum:

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = \ddot{\theta} = -\frac{g}{l}\,\sin\theta\,.$$

- Sometimes must simplify further for analytical result (here: small angle approximation, i.e. $\sin\theta \to \theta$)
- Computer enters to go beyond highly idealised cases: no analytical solution $\to$ approximate numerically!
- Implement/code numerical solution, but then the real work only begins:
  - code verification (aka "debugging")
  - model validation/refinement (against "reality")

Some vocabulary

- Solutions of mathematical models: typically functions
  like, e.g., $\theta(t)$ for pendulum or similar.
- Functions are maps:

$$\begin{array}{rcl} \text{set of input points} & \longrightarrow & \text{set of output points} \\ \text{domain} & \longrightarrow & \text{range}. \end{array}$$

- Computations evaluate functions.
- An algorithm is a recipe to transform inputs to outputs.
- In the digital world (computers),
  the domain and range are discrete, and
  the algorithm terminates after a finite number of steps

Types of errors

- Truncation errors:
  Many functions given as series
  like, e.g., $\sin x = x - \frac{x^3}{3!} + \ldots$ or similar.
  In their evaluation, computer must stop at some point.
- Accuracy errors:
  Computer uses finite number of bits for representing a number, therefore finite precision only. This leads to round-off errors.
  Example: $10^{30} + 1 - 10^{30} = 0$.
- Discretisation errors:
  Representing smooth functions with discrete steps.
  Should decrease with step-size.

These types of error can accumulate!

$\Longrightarrow$ Can lead to instabilities!

(Correct algorithm produces wrong solution - no issue here)

Starting with physics: Radioactive decays

- Simple differential equation (1st order):

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\frac{N}{\tau} \stackrel{!}{=} f(N)\,,$$

where $N(t)$ is the number of atoms in the material at any given time t, and τ is the time-constant of the decay. It is linked to the half-life $T_{1/2}$ like

$$T_{1/2} = \tau \cdot \ln 2\,.$$

- Analytical solution is equally straightforward:

$$N(t) = N(0) \exp\left(-\frac{t}{\tau}\right)\,.$$

Numerical solution: discretisation

- Basic idea: continuous time $t \leftrightarrow$ discrete times t_i $i \in N$.
- How does this work out?
  - Remember definition of derivative:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{N(t+\Delta t) - N(t)}{\Delta t}$$

  - Approximate $\Delta t \to 0$ with Δt "small enough":

$$\frac{\mathrm{d}N}{\mathrm{d}t} \approx \frac{N(t+\Delta t) - f(t)}{\Delta t}$$

  - Therefore rewrite differential eqn as difference eqn:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\frac{N}{\tau} \Longleftrightarrow N(t+\Delta t) - N(t) = -\frac{N(t)\Delta t}{\tau}\,.$$

Numerical solution: discretisation (cont'd)

- With discrete times:

$$
\begin{aligned}
N_{i+1} = N(t_{i+1}) &= N(t_i) - N(t_i)\frac{\Delta t}{\tau} \\
t_{i+1} &= t_i + \Delta t
\end{aligned}
$$

- Simple general solution for first order differential eqn's:

  Euler method

  The first order differential equation $\frac{\mathrm{d}\underline{x}}{\mathrm{d}t} = \underline{f}(\underline{x},\, t)$ can be solved numerically (vector form) by

$$
\begin{aligned}
\underline{x}_{i+1} &= \underline{x}_i + \underline{f}(\underline{x}_i,\, t)\Delta t \\
t_{i+1} &= t_i + \Delta t\,.
\end{aligned}
$$

Error estimate (for discretisation error)

- To estimate error inherent in Euler method, start with Taylor expansion of

$$x(t+\Delta t) = x(t) + \frac{\mathrm{d}x(t)}{\mathrm{d}t}\Delta t + \frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2}\frac{(\Delta t)^2}{2!} + \dots$$

- Algorithm uses first two terms, but ignores all others starting at the third one.

$\Longrightarrow$ Error per step: $\mathcal{O}[(\Delta t)^2]$

- But number of steps from t_0 to $t_{\mathrm{end}} \sim 1/\Delta t$!

$\Longrightarrow$ Overall error: $\mathcal{O}[(\Delta t)]$

Pseudo-code for solution

Main program

- Input of initial conditions
- Initialise classes "Radioactive" and "DEq_Solver"
- Calculate the trajectory.
- Print/plot the result.

Initialisation of physics problem (in "Radioactive")

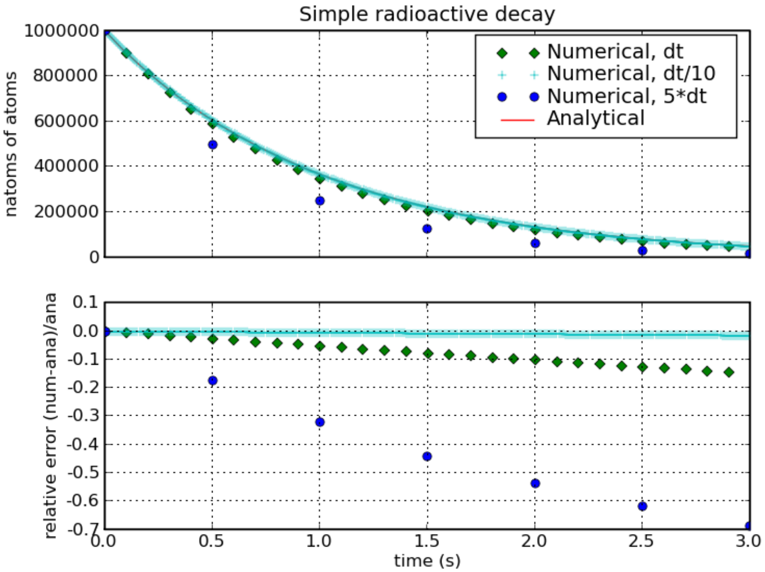
- Fix $t_0 = 0$, $t_{\rm end}$, N_0, τ
- δt part of the calculation, not the physics

Calculation (in "DEq_Solver")

- Iterate time steps until $t_i \geq t_{\rm end}$ is reached:

$$\begin{aligned} t_{i+1} &= t_i + \Delta t \\ \underline{x}(t_{i+1}) = \underline{x}_{i+1} &= \underline{x}_i + \underline{f}(\underline{x}_i,\, t_i)\Delta t\,. \end{aligned}$$

Results


Simple radioactive decay
Numerical, dt
Numerical, dt/10
Numerical, 5*dt
Analytical
natoms of atoms
1000000
800000
600000
400000
200000
0
relative error (num-ana)/ana
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
0.0
0.5
1.0
1.5
2.0
2.5
3.0
time (s)

Summary

- Computer methods paramount when realistic physical situations/phenomena to be described quantitatively. There, typically, analytical solutions are not available, enforcing the use of numerical approaches.
- Various types of error intrinsic to using computers: truncation, accuracy, discretisation.
- A classical problem in physics: solving differential eqn's. Strategy (like nearly always): discretisation and rewriting differential eqn's as difference eqn's.
  First method: Euler method - accurate to first order.