SUBITOP Understanding subduction zone topography through modelling of coupled shallow and deep processes

modelling workshop

Days 2+3: Introduction to numerical modelling

Jeroen van Hunen

Durham University
Department of Earth Sciences

community.dur.ac.uk/jeroen.van-hunen/Subitop

General Learning Objectives

- How to use Python as a modelling tool
- How to describe physical processes mathematically:
  - Heat diffusion and advection
  - Fluid flow
- Mathematical concepts required to build numerical models:
  - Finite differences
  - Discretization
  - Time-stepping
  - Boundary conditions
  - Coupled equations
- How to construct basic numerical models in Python:
  - Heat advection-diffusion models
  - Mantle convection
- How to critically evaluate numerical models

General Setup

1. Tue AM Introduction; 0D, 1D; radioactive decay, diffusion
2. Tue PM Extension to 2D models
3. Wed AM Advection-diffusion equation
4. Wed PM Coupled equations: convection model

Today's aims:

- The basic steps and processes behind building a numerical model
- How timestepping works and be able to compare different timestepping techniques
- Using resolution tests to check your model against analytical solutions
- Modelling a radioactive decay system
- The diffusion process and its governing equation
- Numerical modelling of spatially varying processes
- How to apply finite difference techniques to model 1D time-dependent heat diffusion

A first example: Filling a bath tub

A very basic example: open tap & fill bath tub. What is amount of water in the tub through time?

- If we assume that it fills at a constant rate of 2.6 litres/sec,

  we get a basic equation ('governing equation') of:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = b, \quad b = 2.6$$

- Initially the bath is empty

  Initial condition: $V(0) = 0$

Analytical solution:

$$V(t) = 0 + 2.6t = 2.6t$$

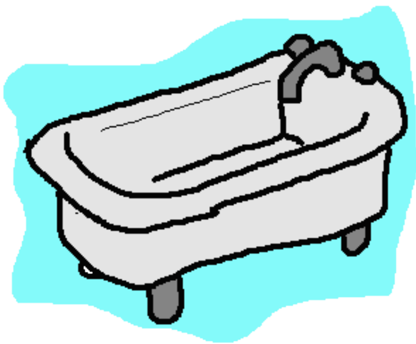
A first example: Filling a bath tub

Numerical approach:

Governing Equation:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = b$$ (infinitesimal calculus expression)

$$\frac{\Delta V}{\Delta t} = \frac{V_{new} - V_{old}}{\Delta t} = b$$ (discrete expression)



$$V_{new} = V_{old} + b\Delta t$$

(unknowns left, known variables to right)

Numerical time step: try Δt = 2 seconds

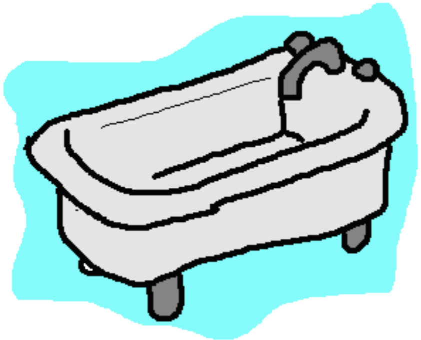
Timestep 1: $V^1 = V^0 + b\Delta t = 0 + 2.6 * 2 = 5.2$ litres

Timestep 2: $V^2 = V^1 + b\Delta t = 5.2 + 2.6 * 2 = 10.4$ litres

and so on …

Second example: Draining a bath tub

- Possible differential equation: $\dfrac{dV}{dt} = -aV$
- Analytical solution: $V(t) = V^0 \exp(-at)$
- with V^0 the amount of water at $t=0$
- Numerical approach: $\dfrac{V^{new} - V^{old}}{\Delta t} = -aV^{old}$



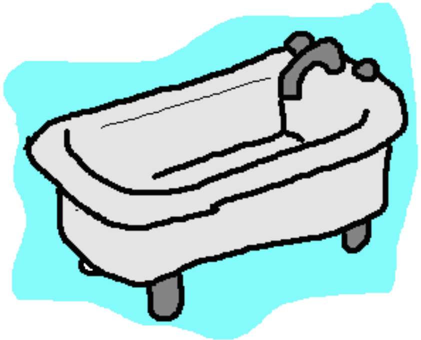
Let's take V^0=500 litres, and a = 0.01:

Timestep 1: $V^1 = V^0 - a\Delta t * V^0 = 500 - 0.01 * 2 * 500 = 490.0$

Timestep 2: $V^2 = V^1 - a\Delta t * V^0 = 490 - 0.01 * 2 * 490 = 480.2$

Third example: combination of the previous two

$$\frac{dV}{dt} = -aV + b$$



- Analytical solution for an initially empty bath:

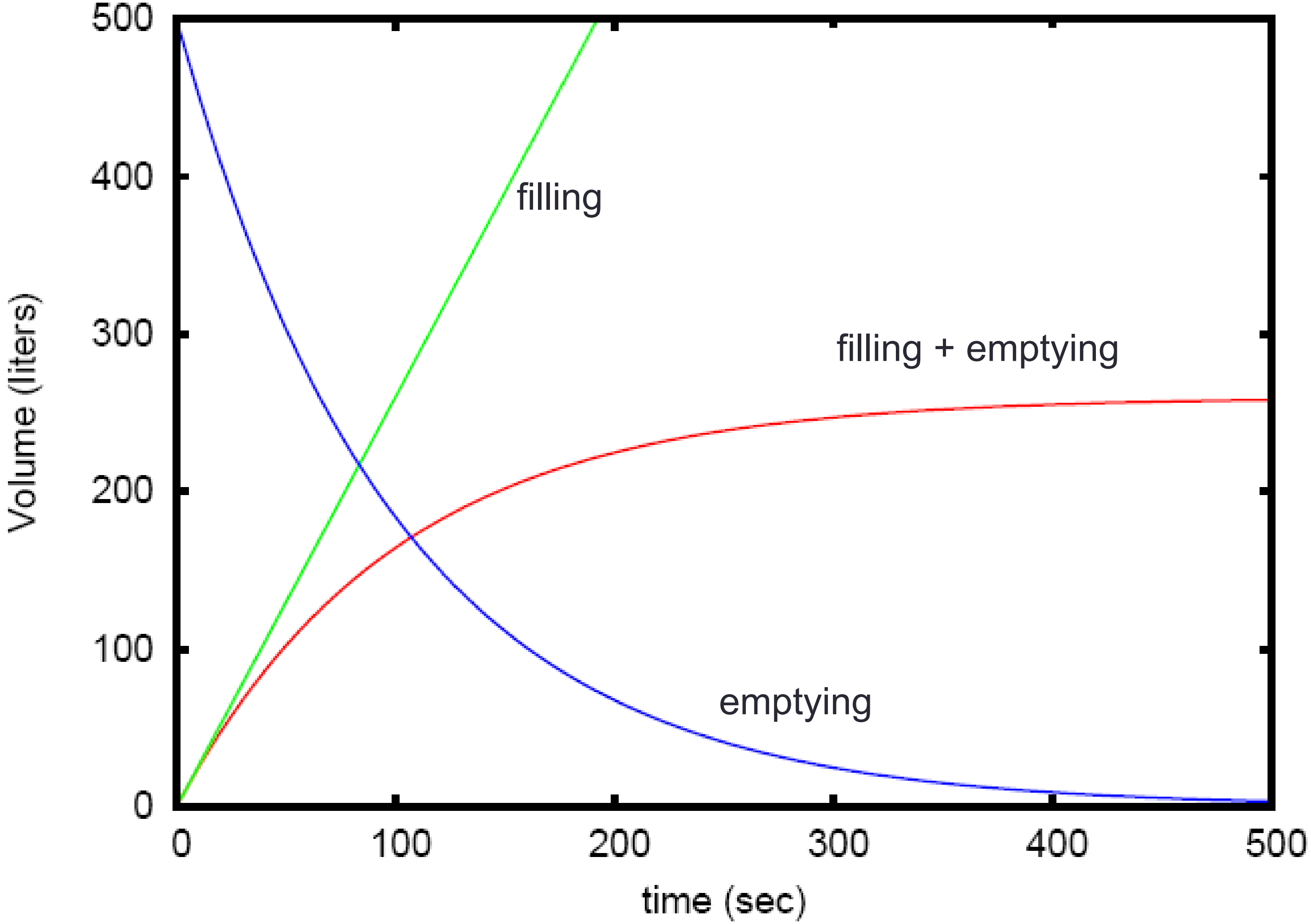
$$V(t) = \frac{b}{a}\left(e^{-at} - 1\right)$$

- Numerical approach:

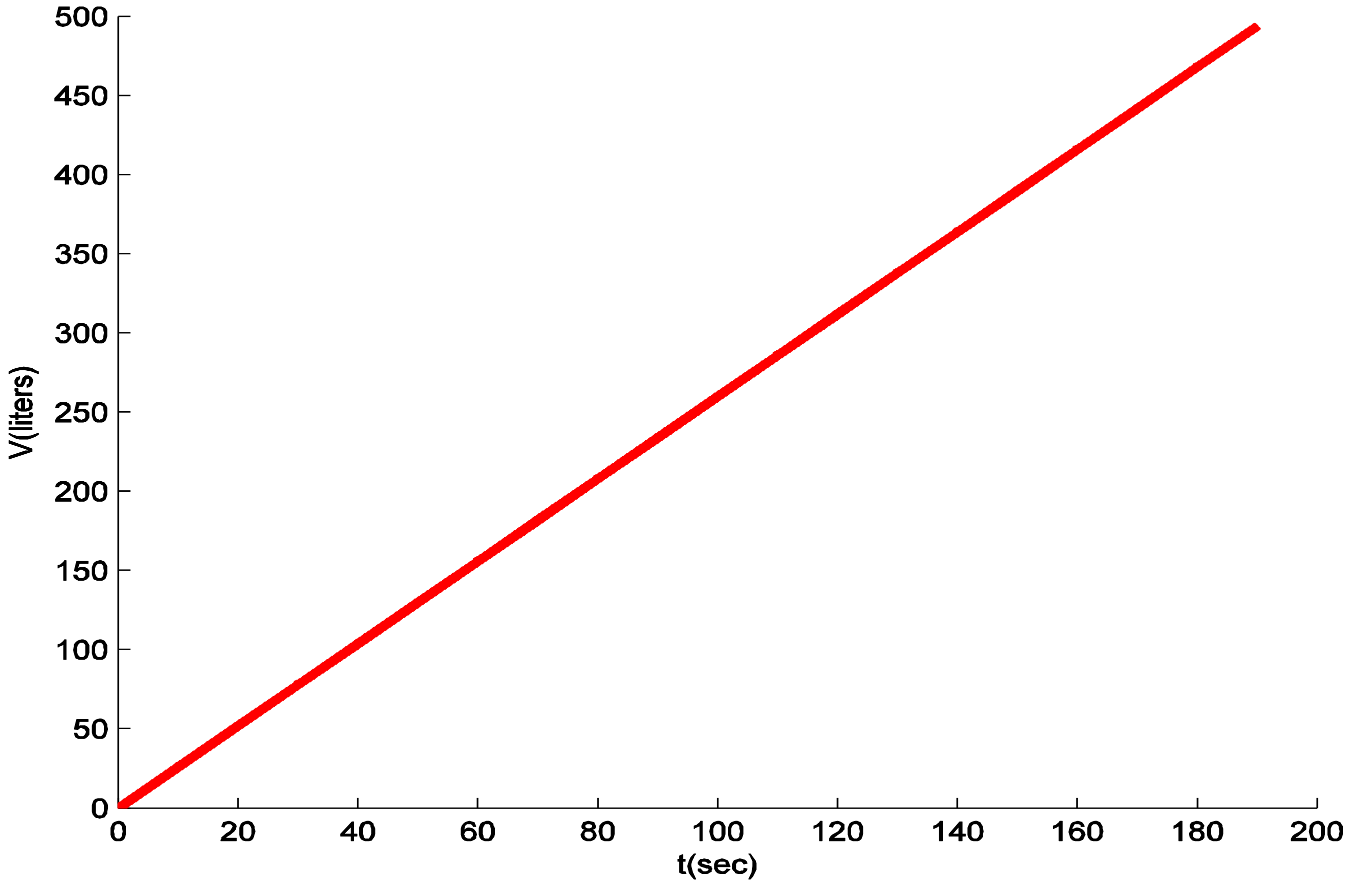
Time step 1: $V^1 = V^0 + dt * (-aV_0 + b) = 0 + 2*(-0.01*0 + 2.6) = 5.2$

Time step 2: $V^2 = V^1 + dt * (-aV_1 + b) = 5.2 + 2*(-0.01*5.2 + 2.6) = 10.296$

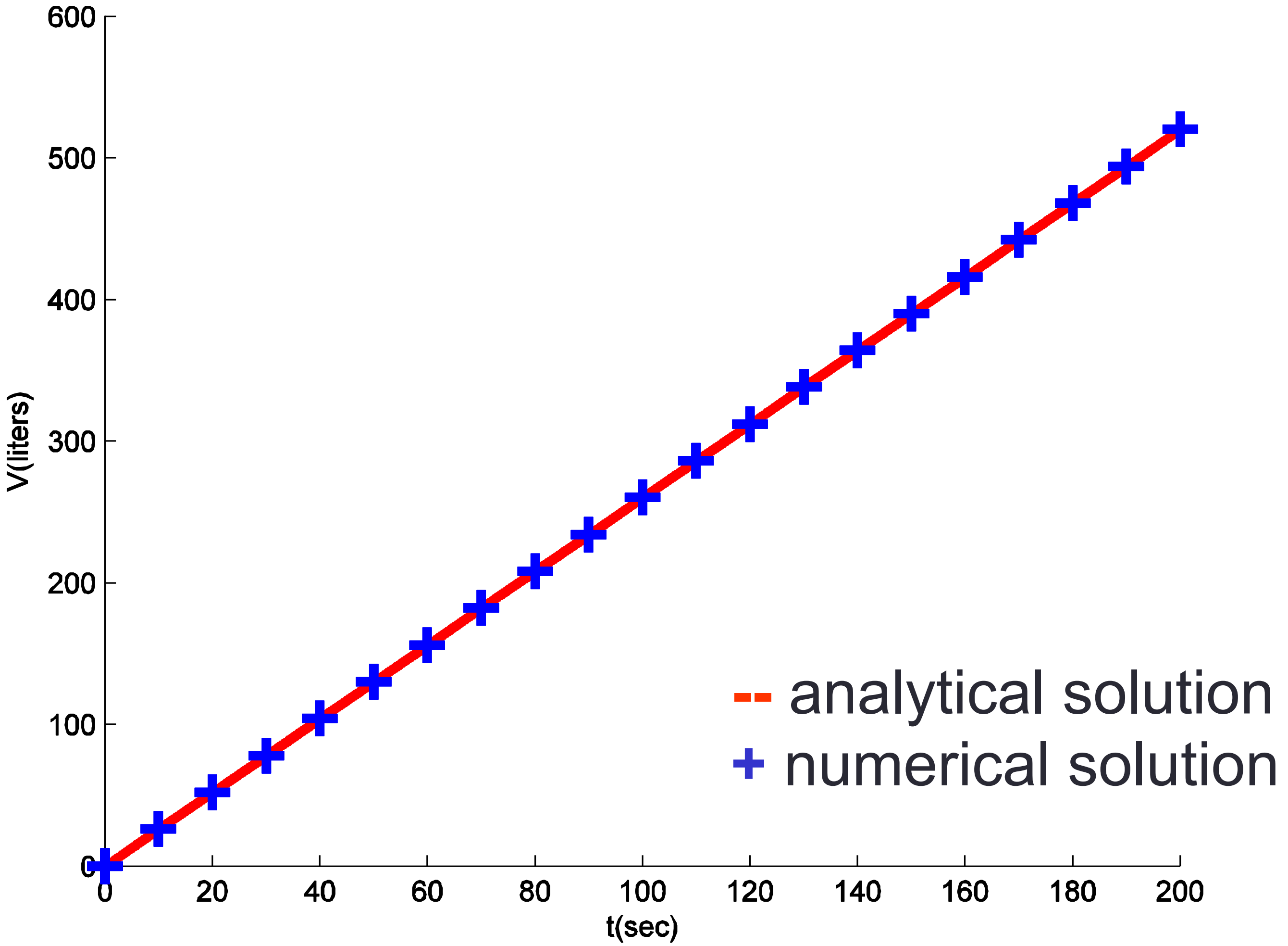
Analytical Solutions



Accuracy of the model: Filling the bathtub



Accuracy of the model: Filling the bathtub



```
import numpy as np
import matplotlib.pyplot as plt

# Initialisation of the parameters:
v_ini  = 500            # Volume at time t=0
v_old  = v_ini
a      = -0.01          # parameter related to emptying bathtub
t      = 0              # set start time to 0
dt     = 25             # time step size
nt     = 25             # number of time steps
volume = np.zeros(nt)   # create some arrays to put solutions in
time   = np.zeros(nt)

# Start time-stepping
for it in range(0,nt):
    # save old solutions in arrays:
    volume[it] = v_old
    time[it]   = t
    v_new      = v_old + dt*a*v_old # Calculate new solution
    t          = t + dt             # Update time
    v_old      = v_new              # Prepare for next time step

# Plot solution:
plt.clf()
plt.figure(1)
plt.plot (time,volume,'o-')
plt.xlabel('t(sec)')
plt.ylabel('V(litres)')
plt.title('Bath tub volume through time')
plt.show()
```

initialisation

calculation

plotting

Python code for calculation of draining of a bath tub

Practical 1, Part 1:

- Take a look at our first numerical model.
- Add the analytical solution. Is the solution perfect?
  - Why or why not?
- What effect does the size of the time step have on the accuracy of the model
  - Why?

https://community.dur.ac.uk/jeroen.van-hunen/Subitop/session1.html

Finite difference approximation

- Taylor expansion: $$f(t+\Delta t) = f(t) + \Delta t \frac{df}{dt} + \frac{\Delta t^2}{2!}\frac{d^2 f}{dt^2} + \frac{\Delta t^3}{3!}\frac{d^3 f}{dt^3} + \frac{\Delta t^4}{4!}\frac{d^4 f}{dt^4} + \dots$$
- Truncate: $$f(t+\Delta t) = f(t) + \Delta t \frac{df}{dt} + O(\Delta t^2)$$
- Re-arrange: $$f(t+\Delta t) - f(t) = \Delta t \frac{df}{dt} + O(\Delta t^2)$$

$$\frac{f(t+\Delta t) - f(t)}{\Delta t} = \frac{df}{dt} + O(\Delta t)$$

or: $$\boxed{\frac{df}{dt} = \frac{f(t+\Delta t) - f(t)}{\Delta t} + O(\Delta t)}$$

Thus: derivative of function $h(t)$ at time $t \approx$ forward difference of the function over a time step Δt.

This approximation has additional terms the largest of which includes the factor Δt.

Time stepping Methods

- Differential equation

$$\frac{dV}{dt} = -aV$$

- Discretization

$$\frac{V^{new} - V^{old}}{\Delta t} = -aV$$

$$\frac{V^{new} - V^{old}}{\Delta t} = -a\underline{V^{old}}$$

forward Euler method

$$\frac{V^{new} - V^{old}}{\Delta t} = -a\underline{V^{new}}$$

backward Euler method

Time stepping Methods

Re-arranging backward Euler equation:

$$\frac{V^{new} - V^{old}}{\Delta t} = -aV^{new}$$

Time stepping Methods

Re-arranging backward Euler equation:

$$\frac{V^{new} - V^{old}}{\Delta t} = -aV^{new}$$

$$V^{new} - V^{old} = -a\Delta t V^{new}$$

$$V^{new}(1 + a\Delta t) = V^{old}$$

$$V^{new} = \frac{V^{old}}{(1 + a\Delta t)}$$

Time stepping Methods

Discretization

$$\frac{V^{new} - V^{old}}{\Delta t} = -aV$$

$$\frac{V^{new} - V^{old}}{\Delta t} = -aV^{old}$$

forward Euler method

$$\frac{V^{new} - V^{old}}{\Delta t} = -aV^{new}$$

backward Euler method

$$\frac{V^{new} - V^{old}}{\Delta t} = -a\left(\frac{V^{old} + V^{new}}{2}\right)$$

Crank-Nicholson method

Practical 1, Part 2:

- Using finite difference techniques to model radioactive decay
  - Work in small groups to discuss key modelling decisions about how to describe the geological process mathematically, what other information you'll need, how long a time to model, etc.
  - Try out new Python commands and techniques you'll need for your model
  - Build the model and implement different time stepping methods yourself
  - If time permits, build a larger model to explore the Earth's secular cooling (Practical 1, Extras, part A)

`https://community.dur.ac.uk/jeroen.van-hunen/Subitop/session1.html`