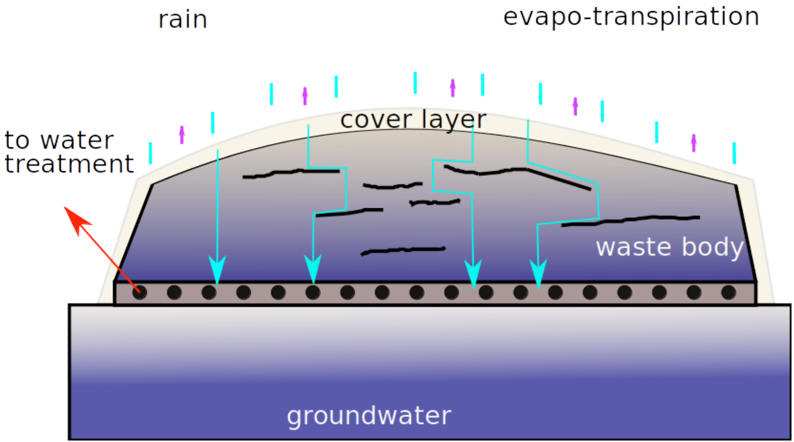
Lecture 1: Introduction Coupled Processes

Timo Heimovaara

Section Geo-Engineering, Delft University of Technology

CIE4365 Period 4, 2021

Modelling the Water balance of a Landfill?



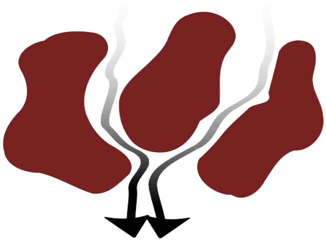


Figure: Conceptual model for a landfill

Conceptual Model similar to a catchment

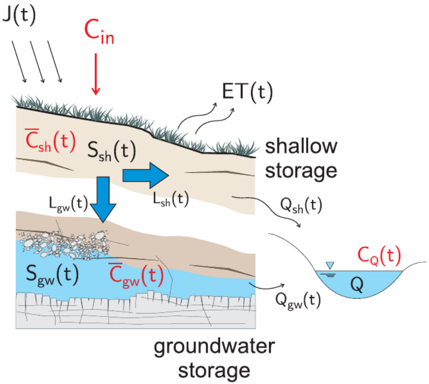
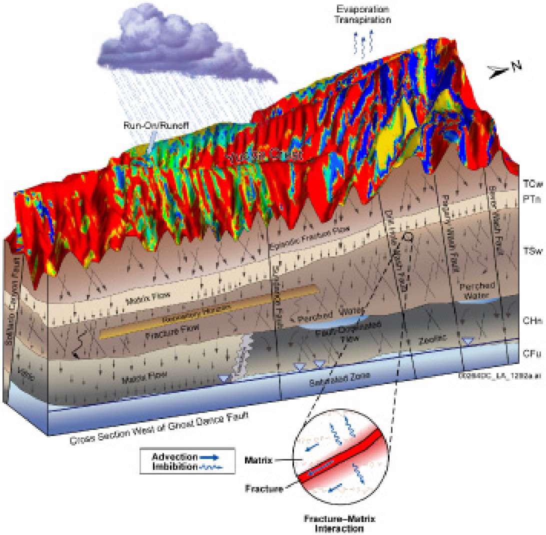
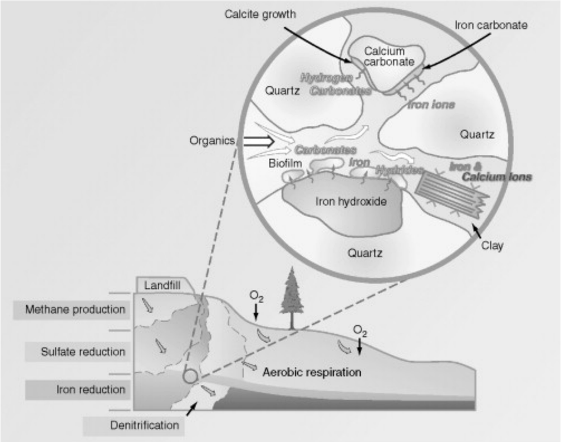


Figure: Water routing in a catchment

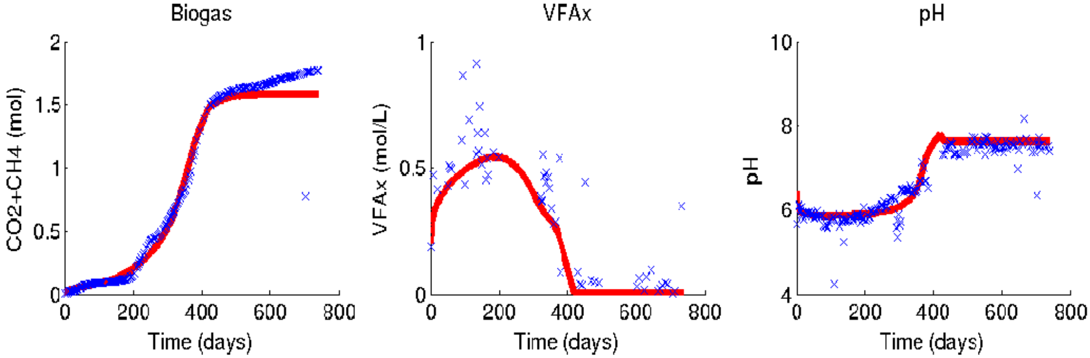
Benettin, P., J. W. Kirchner, A. Rinaldo, and G. Botter (2015), Modeling chloride transport using travel time distributions at Plynlimon, Wales, Water Resour. Res., 51, 3259–3276, doi:10.1002/2014WR016600.

Introduction Coupled Processes in Geo-engineering





Example (Landfill Emissions, André van Turnhout)



Coupled processes

	Gradient				
Flow J	Hydraulic	Electric	Chemical	Thermal	Mechanical
Fluid	Darcy's Law	Electro-osmosis	Normal osmosis	Thermal osmosis	Swelling / shrinkage
Electrical Current	Streaming potential	Ohm's Law	Diffusion & membrane potentials	Seebeck effect	?
Ion	Streaming current	Electrophoresis	Fick's Law	Soret effect	Swelling / Shrinkage
Heat	Isothermal heat transfer	Peltier Effect	Dufour Effect	Fouriers Law	Swelling / Shrinkage
Strain	Swelling / Shrinkage	?	Swelling / Shrinkage	Swelling / Shrinkage	Stress / Strain

Why Matlab?

- Generic tool
- Fourth Generation Programming Language
- Mixed approach: Interpreted and compiled
- Easy to implement concepts
- High level tools available (Debugging, publishing, apps, etc.)
- Very advanced visualisation

Why Python?

- Similar functionality to Matlab
- Free and open source
- More choices, more complicated...

Course schedule

- Differing contact hours (see schedule on Brightspace!)
- Lots of self-study (support available, Timo, Liang, and Aoxi)
- Support in scheduled group sessions in TEAMS (link on Brightspace)
- Virtual classroom lectures can be organized at your request!

Course contents

Lecture 1 Introduction: Lotka Volterra & Conceptual water balance modelling

- Differential Equations
- Finite Differences
- Matlab ODE solvers

Assignment #1: Water Balance Model of a Landfill

Course contents (continued)

Lecture 2 Heat Conductance & Unsaturated water flow
Monday 26 April (Timo Heimovaara)

Lecture 3 Coupled unsaturated water flow, heat transport
Monday 3 May (Timo Heimovaara)

Lecture 4 Coupling Solute transport and biodegradation
Monday 31 May (Timo Heimovaara)

Lecture 5 Hydro-mechanical Coupling
Monday 7 June (Amin Askerinejad)

Group Assignments

- Limited time for lectures: mainly Self Study and group work
- You are encouraged to work in groups (maximum of 4 students per group)
  - Enrol yourself in to a group
    - Best approach: carry out the programming individually or in groups of 2 and have regular meetings on progress to share ideas and check each others code.
    - Exam will require you to be able to solve a problem on your own!
    - Exam is online and therefore it will be a individual computer assignment.

Brightspace

- You can use Brightspace group tools
  - Discussion Fora (group and course wide)
  - File sharing
- Assignments need to be uploaded to Brightspace (group assignments)

Study Load

- Course Load = 5EC ==> 140 hours
- about 14 hours per week (for whole of 10 week period)

Deadlines for Assignments (Due Dates)

- Deadlines of first submission of homework assignments are important:
  - Assignment 1: Friday 30 April (feedback)
  - Assignment 2: Friday 21 May (feedback)
  - Assignment 3: Friday 4 June (feedback)
  - Individual Assignment: Tuesday June 22: 16:30 (100% final grade)
    - choose from list of available assignments
    - final assignments available from June 7 2021

Lotka-Volterra

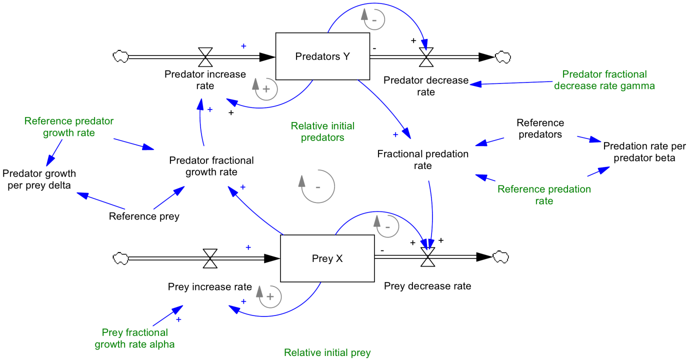
Lotka-Volterra (predator-prey) system

Controls >>
(Page Down)

Adapted from the wikipedia article http://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equation

By Tom Fiddaman, 2011 http://models.metasd.com

The rate equations have been expanded and normalized for clarity, which involves a change in parameters, but the equations are mathematically equivalent to the original



Predator Prey Equations

- Rabbits (R) have unlimited growth (exponential growth, α)
- Foxes (F) grow on rabbits (β)
- Foxes die (γ)

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \alpha R - \beta RF$$

$$\frac{\mathrm{d}F}{\mathrm{d}t} = \delta RF - \gamma F$$

States, Rates & Boundary conditions

- What is a state?
- What is a rate?
- How are they related?

Water and Solute Balance Model

- The equations for the water balance model can be found in Bennettin et al. 2015 which you can download from Brightspace
  - Text of paragraph 4, summarized in figures 4 and 5
- You can download a data from Brightspace:
  - a data file from the Royal Dutch Meteorological Institute with rainfall data for the Berkhout station. This is the closest station to the Wieringermeer landfill;
  - The produced leachate quantity from landfill cell 6 from the Wieringermeer landfill from 2012 to now;
  - The measured concentrations of chloride in this leachate;
- Implement this model in Matlab or Python
  - What are the state and rate variables that you need to calculate?
  - What are the boundary conditions?

Some hints

- Write scripts / routines to read and plot the data available on Brightspace
- Start by developing your implementation strategy with Pseudo Code
- Use the Lotka Volterra example code as base for your own code
- Try to use the built-in PDE solvers of Matlab / Python as much as possible

Forward Discretization, Taylor Series

Forward Taylor-series expansion

$$f(x_0+\Delta x) = f(x_0) + \frac{\partial f(x_0)}{\partial x}\Delta x + \frac{\partial^2 f(x_0)}{\partial x^2}\frac{(\Delta x)^2}{2!} + \frac{\partial^3 f(x_0)}{\partial x^3}\frac{(\Delta x)^3}{3!} + \cdots$$
$$+ \frac{\partial^n f(x_0)}{\partial x^n}\frac{(\Delta x)^n}{n!} + O(\Delta x)^{n+1} \tag{1}$$

so

$$\frac{\partial f(x_0)}{\partial x} = \frac{f(x_0+\Delta x) - f(x_0)}{\Delta x} - \frac{\partial^2 f(x_0)}{\partial x^2}\frac{(\Delta x)}{2!} - \frac{\partial^3 f(x_0)}{\partial x^3}\frac{(\Delta x)^2}{3!} - \cdots$$
$$- \frac{\partial^n f(x_0)}{\partial x^n}\frac{(\Delta x)^{n-1}}{n!} - \cdots \tag{2}$$

$$\frac{\partial f(x_i)}{\partial x} \approx \frac{f_{i+1} - f_i}{\Delta x} + O(\Delta x)$$

Backward Discretization, Taylor Series

Backward Taylor-series expansion

$$f(x_0 - \Delta x) = f(x_0) - \frac{\partial f(x_0)}{\partial x}\Delta x + \frac{\partial^2 f(x_0)}{\partial x^2}\frac{(\Delta x)^2}{2!} - \frac{\partial^3 f(x_0)}{\partial x^3}\frac{(\Delta x)^3}{3!} + \cdots \tag{3}$$

so

$$\frac{\partial f(x_0)}{\partial x} = \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} + \frac{\partial^2 f(x_0)}{\partial x^2}\frac{(\Delta x)}{2!} - \frac{\partial^3 f(x_0)}{\partial x^3}\frac{(\Delta x)^2}{3!}\cdots \tag{4}$$

$$\frac{\partial f(x_i)}{\partial x} \approx \frac{f_i - f_{i-1}}{\Delta x} + O(\Delta x)$$

Adding 2 and 4 and dividing by 2 gives:

$$\frac{\partial f(x_i)}{\partial x} \approx \frac{f_{i+1} - f_{i-1}}{\Delta x} + O(\Delta x)^2$$

Second Derivative

Adding equations 1 and 3 gives:

$$\frac{\partial f^2(x_i)}{\partial x^2} \approx \frac{f_{i+1} - 2f_{i+} f_{i-1}}{(\Delta x)^2} + O(\Delta x)^2$$

First-order Euler

$$y' = \frac{\mathbf{d}y}{\mathbf{d}t} = f(t, y), \qquad y(t_0) = y_0$$

Discretization in time using finite difference:

$$\frac{\mathbf{d}y}{\mathbf{d}t} \approx \frac{\Delta y}{\Delta t} = \frac{y^{k+1} - y^k}{\Delta t} = f(t^k, y^k)$$

so

$$y^{k+1} = y^k + f(t^k, y^k)\Delta t$$

Predictor-Corrector

$$y' = \frac{\mathbf{d}y}{\mathbf{d}t} = f(t, y), \qquad y(t_0) = y_0$$

Iterative approach starting with Euler step:

$$y^{k+1,0} = y^k + f(t^k, y^k)\Delta t$$

followed by a Trapezoidal step:

$$y^{k+1,1} = y^k + \frac{1}{2}\left(f(t^k, y^k) + f(t^k, y^{k+1,0})\right)\Delta t$$

$$y^{k+1,2} = y^k + \frac{1}{2}\left(f(t^k, y^k) + f(t^k, y^{k+1,1})\right)\Delta t$$

$$y^{k+1,n} = y^k + \frac{1}{2}\left(f(t^k, y^k) + f(t^k, y^{k+1,n-1})\right)\Delta t$$

This iteration is repeated until $y^{k+1,n} - y^{k+1,n-1} < \varepsilon_{conv}$ and ε_{conv} is some small number (i.e. 10^{-10})

Runge-Kutta fourth order

$$y' = \frac{\mathbf{d}y}{\mathbf{d}t} = f(t,y), \qquad y(t_0) = y_0$$

The RK4 method is (many resources on Internet):

$$y^{k+1} = y^k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$t^{k+1} = t^k + h$$

where y^{k+1} is the RK4 approximation of y at the next time step and

$$k_1 = hf(t^k, y^k)$$

$$k_2 = hf(t^k + \frac{1}{2}h,\ y^k + \frac{1}{2}k_1)$$

$$k_3 = hf(t^k + \frac{1}{2}h,\ y^k + \frac{1}{2}k_2)$$

$$k_4 = hf(t^k + h,\ y^k + k_3)$$

Matlab ODE solver explicit

Matlab functions (ode45, ode23,:

ode45,ode23, etc.

```
[T,Y] = ode45(@(t,y) MyRates(t, y, Par), [tmin,tmax], yini, options);
```

MyRates(t,y,Par)

```
function dydt = MyRates(t,y,Par)
%Do something with Par
%Do other type of stuff required to calculate the value of dydt
%Calculate the value of dydt
dydt = ....; %Here you make sure that dydt = f(t,y) of your problem
end
```

Matlab ODE solver implicit

ode15i

```
[T,Y] = ode15i(@(t,y,dy) MyRatesImp(t,y,dy,Par),[tmin,tmax],yini, dyini,options);
```

MyRatesImp(t,y,dy,Par)

```
function res = MyRateImp(t,y,dy,Par)
%Do something with Par
%Do other type of stuff required to calculate the value of dydt
%Calculate the value of dydt
res =....; %Here you make sure that res = - dydt +f(t,y) of your problem
end
```