Part 3 of GLEON modeling workshop 2022

Coding a 1D transport model

Robert Ladwig

rladwig2@wisc.edu @hydrobert robertladwig.github.io robertladwig

Diffusive transport

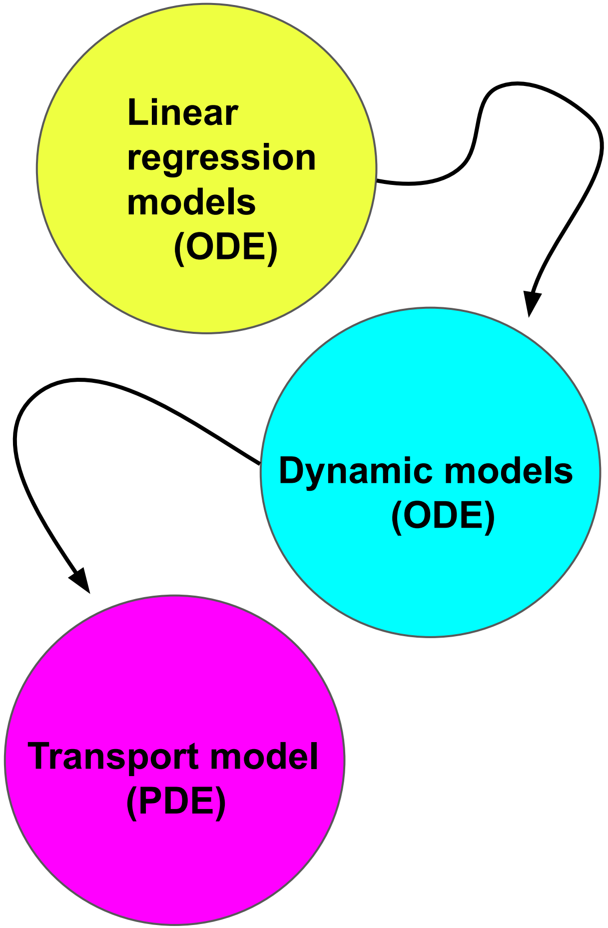
Learning goals:

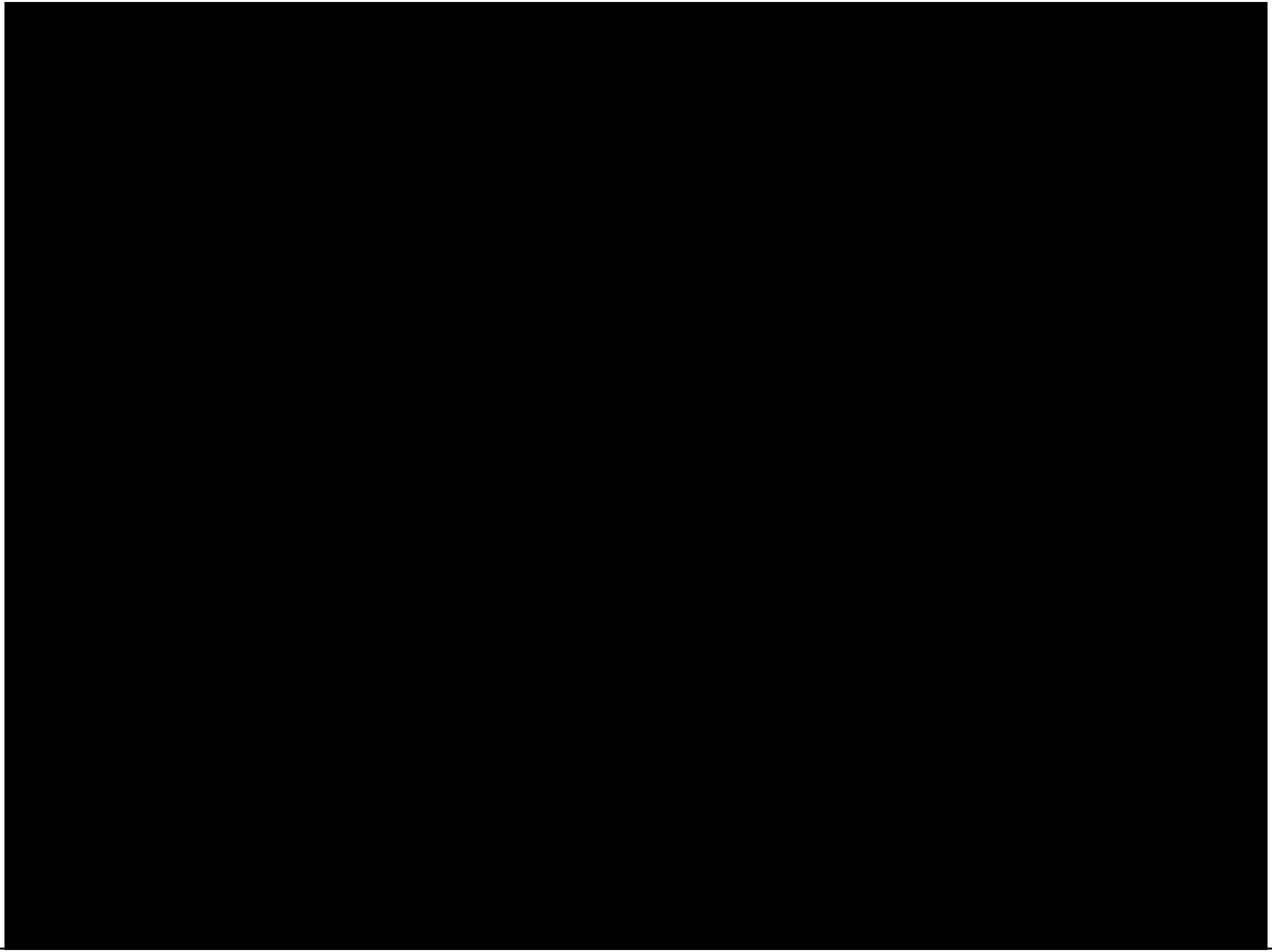
1) Understanding the one-dimensional diffusion equation

2) How can we discretize the equation?

3) How to code up the diffusion model in R and visualize the output

4) Understanding model assumptions and uncertainties





Video by Engin Akyurt

What's diffusion?

- in aquatic systems, **diffusion** is one of the main transport processes
  - **molecular diffusion**: random Brownian motion
  - **turbulent diffusion**: large scale motion by eddies

What’s diffusion?

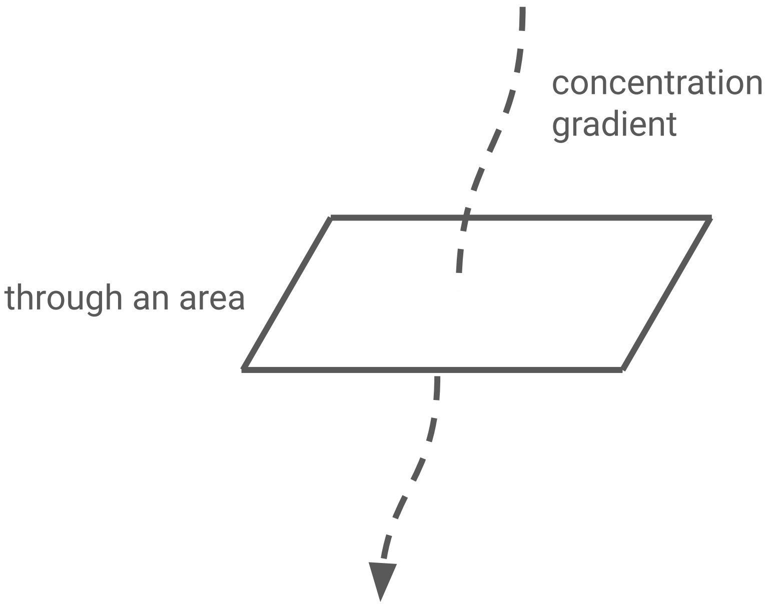
- in aquatic systems, **diffusion** is one of the main transport processes
    - **molecular diffusion**: random Brownian motion
    - **turbulent diffusion**: large scale motion by eddies

- motion by a diffusion coefficient and a (velocity/concentration) gradient

- **let’s model the diffusion of a (passive) tracer in a lake**
    - passive: no reaction/transformation

Vertical axis

- code a model over **time and space**
- transport of a concentration over time



1D diffusion equation

- code a model over **time and space**

diffusion coefficient

$$\frac{dC}{dt} = K\frac{d^2C}{dz^2}$$

change of concentration over time

second-order derivative of the concentration over the vertical axis z

Remainder

- partial differential equation (PDE): **several variables and their derivatives**

$$\frac{dC}{dt} = K \frac{d^2 C}{dz^2}$$

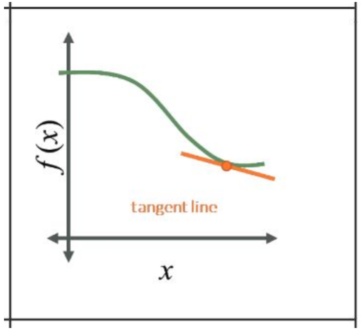
Remainder

- partial differential equation (PDE): **several variables and their derivatives**

$$\frac{dC}{dt} = K\frac{d^2C}{dz^2}$$

- first-order derivative (slope of a function)

$$f'(x) = \frac{\partial y}{\partial x}$$



Remainder

$$\frac{dC}{dt} = K\frac{d^2C}{dz^2}$$

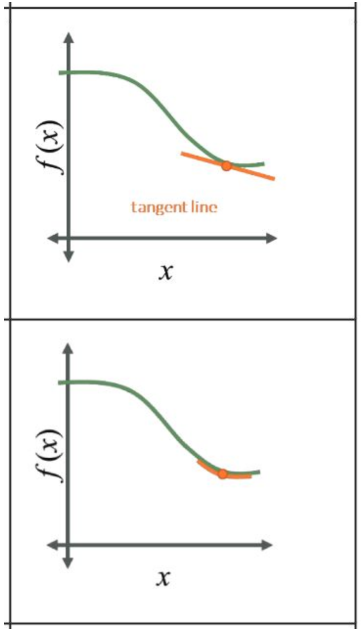
- partial differential equation (PDE): **several variables and their derivatives**

- first-order derivative (slope of a function)

$$f'(x) = \frac{\partial y}{\partial x}$$

- second-order derivative (how fast that slope changes)

$$f'”(x) = \frac{\partial^2 y}{\partial x^2}$$



Check units

- code a model over **time and space**
- **check the units!**

area over time

$$\frac{dC}{dt} = K\frac{d^2C}{dz^2}$$

mass over volume and time

mass over volume and area

$$\frac{g}{m^3 s} = \frac{m^2}{s}\frac{g}{m^3 m^2}$$

Discretization

- luckily, we can discretize this using a **Central Difference Scheme**
- use of Taylor expansion (series expansion of a function)

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + O(a^4)$$

- centered differencing in space, forward differencing in time

Approximating the solution

FTCS (forward in time, centered in space)

- the power of loops!

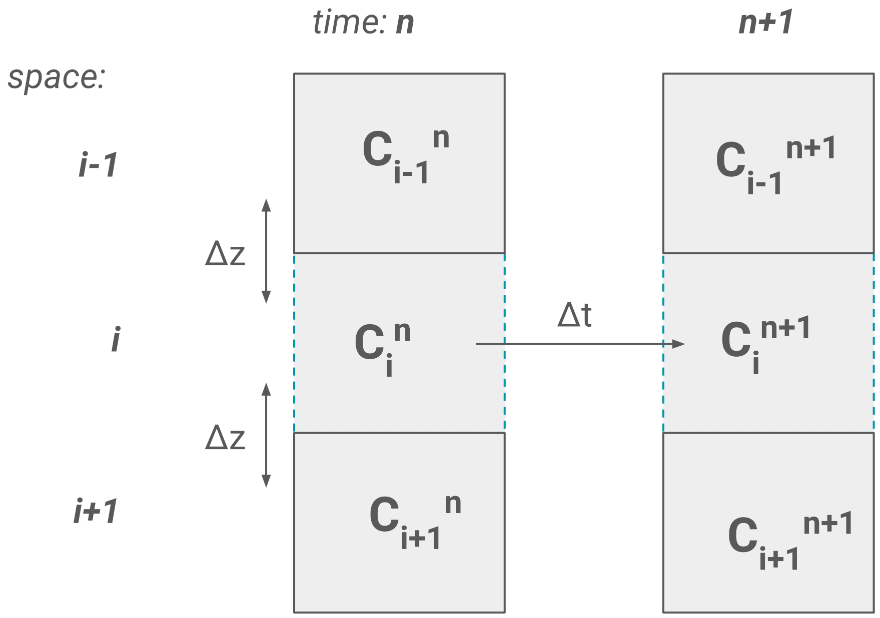
$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta z^2}$$

Approximating the solution

FTCS (forward in time, centered in space)

- the power of loops!

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta z^2}$$

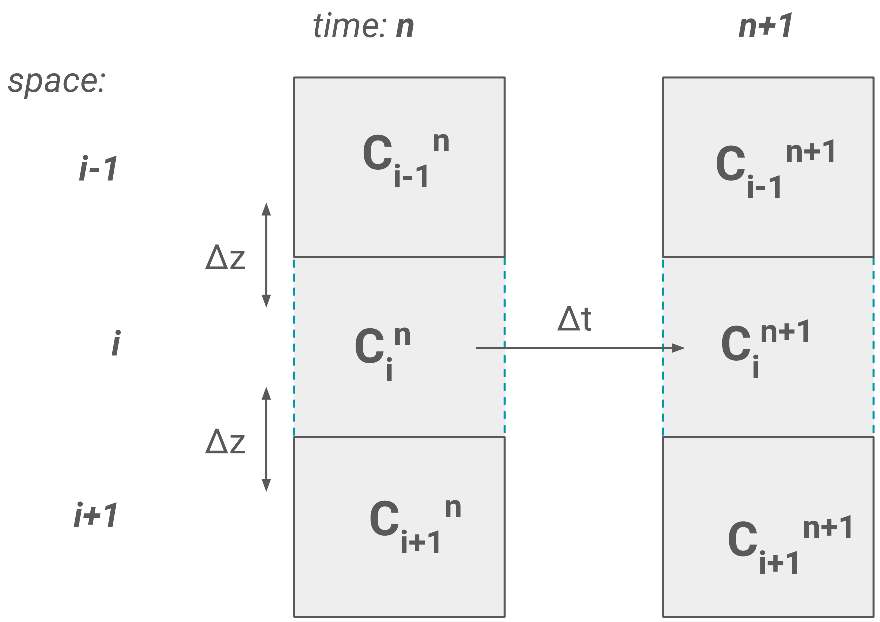


Approximating the solution

FTCS (forward in time, centered in space)

- the power of loops!

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta z^2}$$



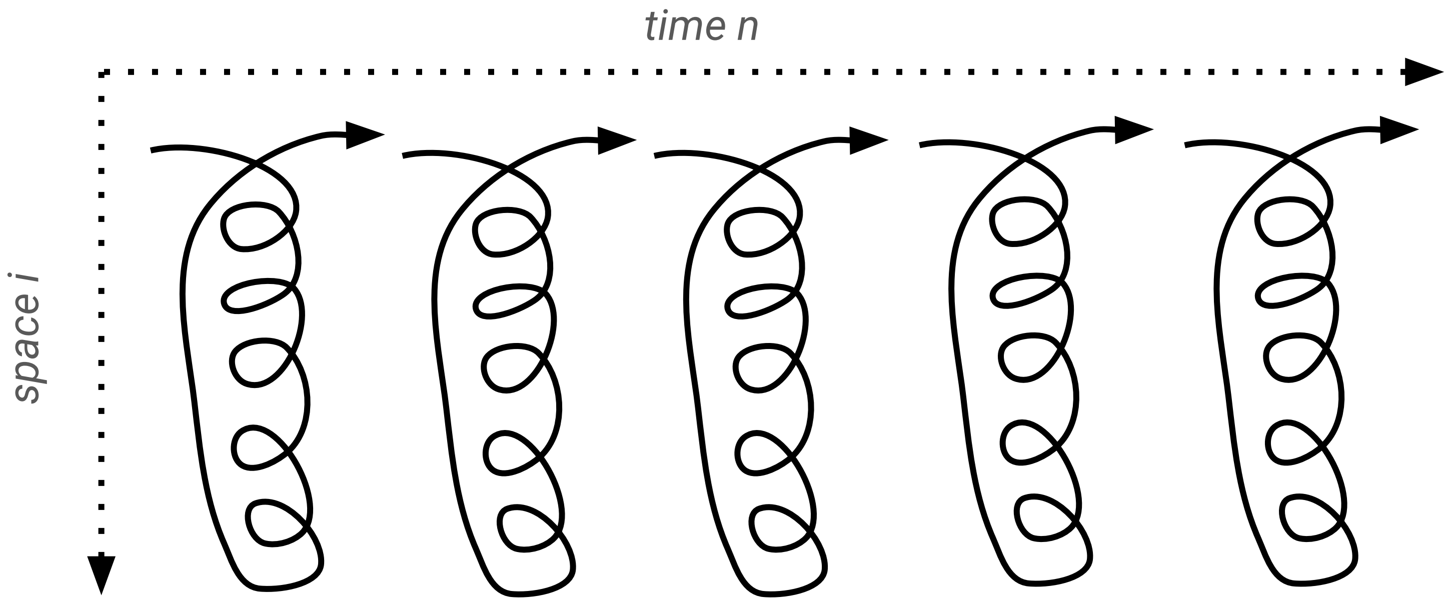
Beware!
this is an **explicit** scheme, therefore its numerical stability depends on the time step size

$$\frac{u\Delta t}{\Delta x} \leq 1$$

Using nested loops

FTCS (forward in time, centered in space)
- the power of loops!

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta z^2}$$

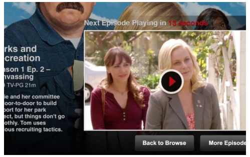


Using nested loops

FTCS (forward in time, centered in space)

- the power of loops!

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta z^2}$$

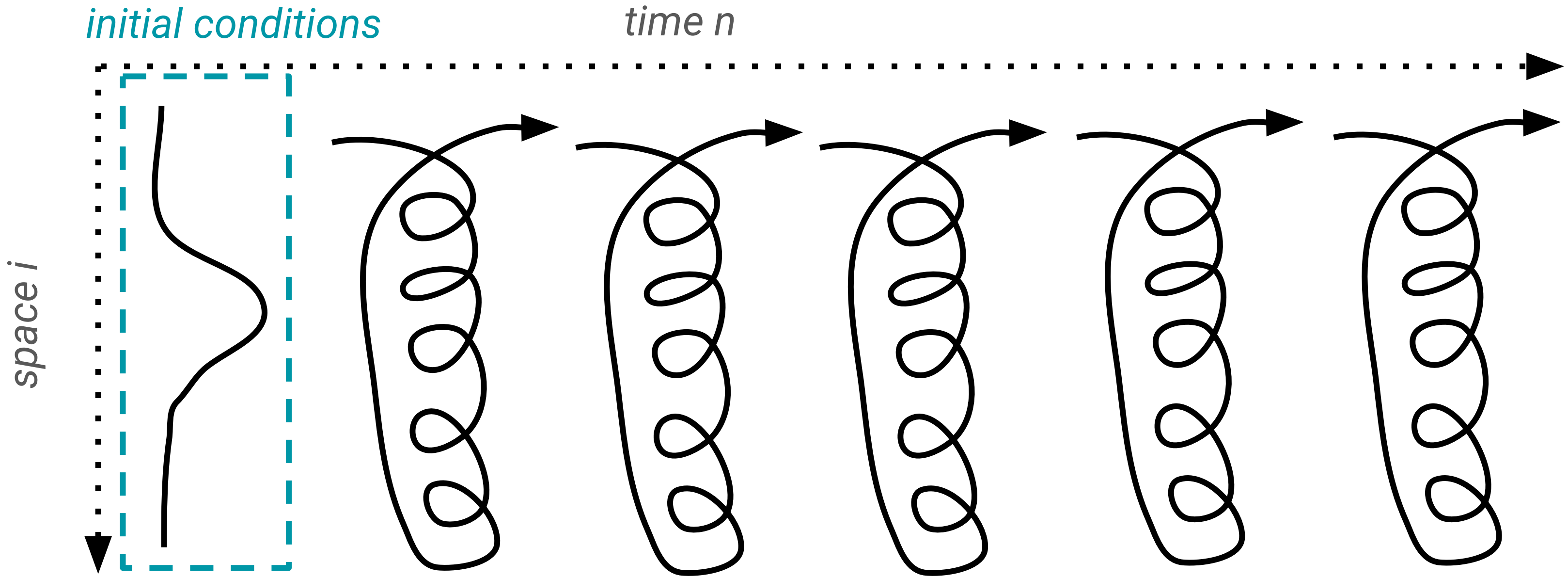


Using nested loops

FTCS (forward in time, centered in space)

- the power of loops!

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta z^2}$$



Coding it in R

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1} - 2C_i + C_{i-1}}{\Delta z^2}$$

i is space index, *n* is time index

```
> time = 100
> space = 100
> conc <- matrix(0, nrow = space, ncol = time)
# our results in a matrix: 100 seconds times 100 m over the depth
> K = 0.5 # diffusion coefficient, unit: m2/s
> dx = 1 # our spatial step, unit: m
> dt = 1 # our time step, unit: s
> conc[, 1] = dnorm(seq(1,100,1), mean = 50, sd = 1) * 100
# initial conc. is defined vertically through a normal distribution, unit: -

> for (n in 2:ncol(conc)){ # time index
      for (i in 2:(nrow(conc)-1)){ # space index
        conc[i, n] = conc[i, n-1] + K * dt / dx**2 * (conc[i+1, n-1] - 2 * conc[i, n-1] +
     conc[i-1, n-1]) # our FTCS scheme
      }
 }
```

Look at results

Visualise it!

```
> library(tidyverse)
> library(reshape2)

> time =  paste0(seq(1,ncol(conc)))
> df <- data.frame(cbind(time, t(conc)) )
> colnames(df) <- c("time", as.character(paste0(seq(1,nrow(conc)))))

> m.df <- reshape2::melt(df, "time")
> m.df$time <- time

> ggplot(m.df, aes(as.numeric(time), as.numeric(variable))) +
     geom_raster(aes(fill = as.numeric(value)), interpolate = TRUE) +
     scale_fill_gradientn(limits = c(-1,100),
                          colours = rev(RColorBrewer::brewer.pal(11, 'Spectral')))+
     theme_minimal()  +xlab('Time') +
     ylab('Depth') +
     labs(fill = 'Conc. [%]')
```

Play around with it

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1} - 2C_i + C_{i-1}}{\Delta z^2}$$

i is space index, *n* is time index

```
> time = 100
> space = 100
> conc <- matrix(0, nrow = space, ncol = time)
# our results in a matrix: 100 seconds times 100 m over the depth
> K = 0.5 # diffusion coefficient, unit: m2/s
> dx = 1 # our spatial step, unit: m
> dt = 1 # our time step, unit: s
> conc[, 1] = dnorm(seq(1,100,1), mean = 50, sd = 1) * 100
# initial conc. is defined vertically through a normal distribution, unit: -

> for (n in 2:ncol(conc)){ # time index
      for (i in 2:(nrow(conc)-1)){ # space index
        conc[i, n] = conc[i, n-1] + K * dt / dx**2 * (conc[i+1, n-1] - 2 * conc[i, n-1] +
     conc[i-1, n-1]) # our FTCS scheme
      }
 }
```

Play around with it

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1} - 2C_i + C_{i-1}}{\Delta z^2}$$

i is space index, *n* is time index

```
> time = 100
> space = 100
> conc <- matrix(0, nrow = space, ncol = time)
# our results in a matrix: 100 seconds times 100 m over the depth
> K = 0.5 # diffusion coefficient, unit: m2/s
> dx = 1 # our spatial step, unit: m
> dt = 1 # our time step, unit: s
> conc[, 1] = dnorm(seq(1,100,1), mean = 50, sd = 1) * 100
# initial conc. is defined vertically through a normal distribution, unit: -

> for (n in 2:ncol(conc)){ # time index
      for (i in 2:(nrow(conc)-1)){ # space index
        conc[i, n] = conc[i, n-1] + K * dt / dx**2 * (conc[i+1, n-1] - 2 * conc[i, n-1] +
     conc[i-1, n-1]) # our FTCS scheme
      }
 }
```

Play around with it

$$C_i^{n+1} = C_i^n + K\Delta t \frac{C_{i+1} - 2C_i + C_{i-1}}{\Delta z^2}$$

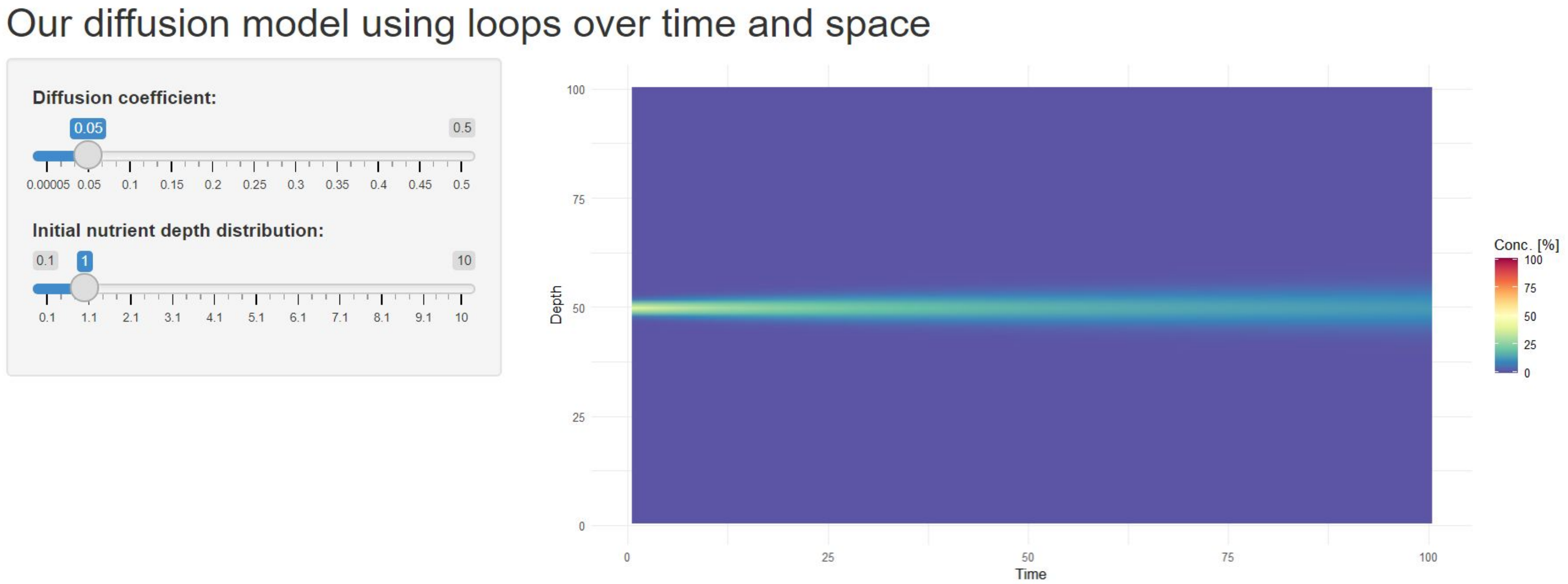
i is space index, *n* is time index

```
> time = 100
> space = 100
> conc <- matrix(0, nrow = space, ncol = time)
# our results in a matrix: 100 seconds times 100 m over the depth
> K = 0.5 # diffusion coefficient, unit: m2/s
> dx = 1 # our spatial step, unit: m
> dt = 1 # our time step, unit: s
> conc[, 1] = dnorm(seq(1,100,1), mean = 50, sd = 1) * 100
# initial conc. is defined vertically through a normal distribution, unit: -

> for (n in 2:ncol(conc)){ # time index
      for (i in 2:(nrow(conc)-1)){ # space index
        conc[i, n] = conc[i, n-1] + K * dt / dx**2 * (conc[i+1, n-1] - 2 * conc[i, n-1] +
     conc[i-1, n-1]) # our FTCS scheme
      }
  }
```

Shiny app

- test our loop model: shorturl.at/asx56
- play around with the diffusion coefficient and initial concentration distribution



Assumptions and uncertainties

- missing terms:
    (1) advection
    (2) reaction

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2}$$

$$\frac{\partial C}{\partial t} = K\frac{\partial^2 C}{\partial z^2} + R$$

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2} + R$$

Assumptions and uncertainties

- missing terms:
    1. advection (**negligible**)
    2. reaction (**important**)

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2}$$

$$\frac{\partial C}{\partial t} = K\frac{\partial^2 C}{\partial z^2} + R$$

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2} + R$$

Assumptions and uncertainties

- missing terms:
  1. advection (**negligible**)
  2. reaction (**important**)

- discretization: explicit
  → **numerical stability**!

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2}$$

$$\frac{\partial C}{\partial t} = K\frac{\partial^2 C}{\partial z^2} + R$$

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2} + R$$

Assumptions and uncertainties

- missing terms:
  1. advection (**negligible**)
  2. reaction (**important**)

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2}$$

$$\frac{\partial C}{\partial t} = K\frac{\partial^2 C}{\partial z^2} + R$$

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2} + R$$

- discretization: explicit
  → **numerical stability**!

- missing diffusion at top & bottom

Assumptions and uncertainties

- missing terms:
  (1) advection (**negligible**)
  (2) reaction (**important**)

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2}$$

$$\frac{\partial C}{\partial t} = K\frac{\partial^2 C}{\partial z^2} + R$$

- -

$$\frac{\partial C}{\partial t} = -w\frac{\partial C}{\partial z} + K\frac{\partial^2 C}{\partial z^2} + R$$

- discretization: explicit
  → **numerical stability**!

- missing diffusion at top & bottom

- how to derive **K**? (turbulence…)

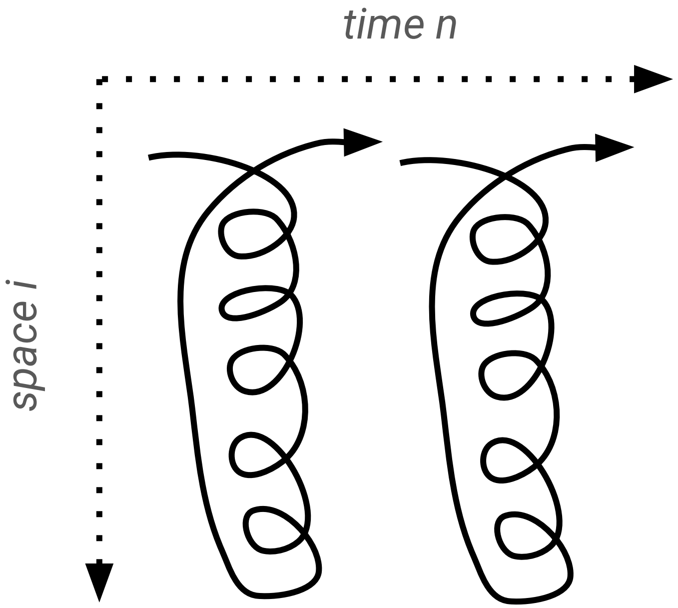
Summing up

$$\frac{dC}{dt} = K\frac{d^2C}{dz^2}$$

- 1D diffusion equation
- discretization
- coding it using nested loops
- model parameters
- uncertainties & assumptions

- add YOUR water quality:

$$\frac{\partial C}{\partial t} = K\frac{\partial^2 C}{\partial z^2} + R$$



FTCS I

- applying Taylor expansion: $$\frac{dC}{dt} = K\frac{d^2C}{dz^2}$$

- forwards: $$C_{i+1} = C_i + \Delta z\frac{\partial C}{\partial z} + \frac{\Delta z^2}{2!}\frac{\partial^2 C}{\partial z^2} + \frac{\Delta z^3}{3!}\frac{\partial^3 C}{\partial z^3} + O(\Delta z^4)$$
- backwards: $$C_{i-1} = C_i - \Delta z\frac{\partial C}{\partial z} + \frac{\Delta z^2}{2!}\frac{\partial^2 C}{\partial z^2} - \frac{\Delta z^3}{3!}\frac{\partial^3 C}{\partial z^3} + O(\Delta z^4)$$

- sum them up to get: $$C_{i+1} + C_{i-1} = 2C_i + \Delta z^2\frac{\partial^2 C}{\partial z^2} + O(\Delta z^4)$$
- and re-arrange: $$\frac{\partial^2 C}{\partial z^2} = \frac{C_{i+1} - 2C_i + C_{i-1}}{\Delta z^2} + O(\Delta z^4)$$

FTCS II

- applying Taylor expansion: $$\frac{dC}{dt} = K\frac{d^2C}{dz^2}$$

- *i* is space index, *n* is time index $$\frac{dC}{dt} = K\frac{C_{i+1} - 2C_i + C_{i-1}}{\Delta z^2}$$

- forward differencing in time: $$\frac{dC}{dt} = \frac{C_i^{n+1} - C_i^n}{\Delta t}$$

$$\boxed{C_i^{n+1} = C_i^n + K\Delta t\frac{C_{i+1} - 2C_i + C_{i-1}}{\Delta z^2}}$$

FCTS (forward in time, central in space)