Partial differential equations

Dr.ir. Ivo Roghair, Prof.dr.ir. Martin van Sint Annaland

Chemical Process Intensification group
Eindhoven University of Technology

Numerical Methods (6E5X0), 2023-2024

Today's outline

- Introduction

- Instationary diffusion equation
  - Discretization
  - Solving the diffusion equation
  - Non-linear source terms

- Convection
  - Discretization
  - Central difference scheme
  - Upwind scheme

- Conclusions
  - Other methods
  - Summary

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Today's outline

● Introduction

● Instationary diffusion equation
- Discretization
- Solving the diffusion equation
- Non-linear source terms

● Convection
- Discretization
- Central difference scheme
- Upwind scheme

● Conclusions
- Other methods
- Summary

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Overview

Main question

How to solve parabolic PDEs like:

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} - u\frac{\partial c}{\partial x} + R$$

with

$$t = 0; 0 \le x \le \ell \quad \Rightarrow c = c_0$$

$$t > 0; x = 0 \quad \Rightarrow -\mathcal{D}\frac{\partial c}{\partial x} + uc = u_{\text{in}}c_{\text{in}}$$

$$t > 0; x = \ell \quad \Rightarrow \frac{\partial c}{\partial x} = 0$$

accurately and efficiently?

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

What is a PDE?

Partial differential equation

An equation containing a function and their derivatives to multiple independent variables.

Order of PDE

The highest derivative appearing in the PDE

What is a PDE?

Partial differential equation

An equation containing a function and their derivatives to multiple independent variables.

Order of PDE

The highest derivative appearing in the PDE

General second order PDE:

$$A\frac{\partial^2 f}{\partial x^2} + B\frac{\partial^2 f}{\partial x \partial y} + C\frac{\partial^2 f}{\partial y^2} + D\frac{\partial f}{\partial x} + E\frac{\partial f}{\partial y} + Ff = G$$

- Linear equation: Coefficients $A, B, \ldots, G$ do not depend on x and y.
- Non-linear equation: Coefficients $A, B, \ldots, G$ are a function of x and y.

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Classification of PDE's

$$A\frac{\partial^2 f}{\partial x^2} + B\frac{\partial^2 f}{\partial x \partial y} + C\frac{\partial^2 f}{\partial y^2} + D\frac{\partial f}{\partial x} + E\frac{\partial f}{\partial y} + Ff = G$$

The discriminant Δ of a quadratic polynomial is computed as (note: only the higher order coefficients are important):

$$\Delta = B^2 - 4AC$$

Classification of PDE's

$$A\frac{\partial^2 f}{\partial x^2} + B\frac{\partial^2 f}{\partial x \partial y} + C\frac{\partial^2 f}{\partial y^2} + D\frac{\partial f}{\partial x} + E\frac{\partial f}{\partial y} + Ff = G$$

The discriminant Δ of a quadratic polynomial is computed as (note: only the higher order coefficients are important):
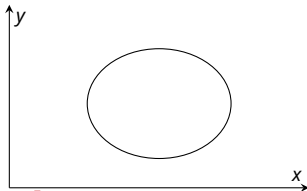
$\Delta = B^2 - 4AC$

- $\Delta < 0 \Rightarrow$ Elliptic equation
  (e.g. Laplace equation for stationary diffusion in 2D)

- $\Delta = 0 \Rightarrow$ Parabolic equation
  (e.g. instationary heat penetration in 1D)

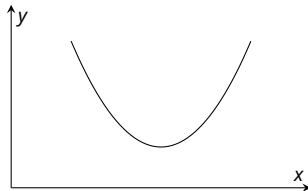- $\Delta > 0 \Rightarrow$ Hyperbolic equation
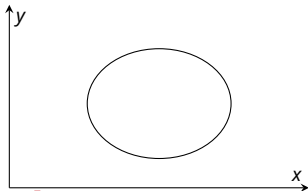 (e.g. wave equation)

Classification of PDE's

$$A\frac{\partial^2 f}{\partial x^2} + B\frac{\partial^2 f}{\partial x \partial y} + C\frac{\partial^2 f}{\partial y^2} + D\frac{\partial f}{\partial x} + E\frac{\partial f}{\partial y} + Ff = G$$

The discriminant Δ of a quadratic polynomial is computed as (note: only the higher order coefficients are important):

$\Delta = B^2 - 4AC$

- $\Delta < 0 \Rightarrow$ Elliptic equation
  (e.g. Laplace equation for stationary diffusion in 2D)

- $\Delta = 0 \Rightarrow$ Parabolic equation
  (e.g. instationary heat penetration in 1D)

- $\Delta > 0 \Rightarrow$ Hyperbolic equation
  (e.g. wave equation)

Introduction
○○○●○○○○○○

Instationary diffusion equation
○○○○○○○○○○○○○○○○○○○

Convection
○○○○○○○○

Conclusions
○○○○○

Classification of PDE's

$$A\frac{\partial^2 f}{\partial x^2} + B\frac{\partial^2 f}{\partial x \partial y} + C\frac{\partial^2 f}{\partial y^2} + D\frac{\partial f}{\partial x} + E\frac{\partial f}{\partial y} + Ff = G$$

The discriminant Δ of a quadratic polynomial is computed as (note: only the higher order coefficients are important):
$\Delta = B^2 - 4AC$

- $\Delta < 0 \Rightarrow$ Elliptic equation
  (e.g. Laplace equation for stationary diffusion in 2D)

- $\Delta = 0 \Rightarrow$ Parabolic equation
  (e.g. instationary heat penetration in 1D)

- $\Delta > 0 \Rightarrow$ Hyperbolic equation
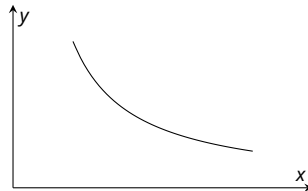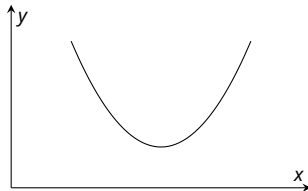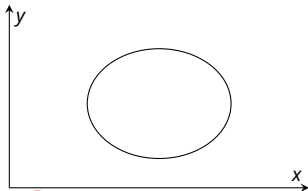 (e.g. wave equation)

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Importance of classification

Different PDE types require different solution techniques because of the difference in range of influence:

- *Characteristics*
  Curves in *xy*-domain along with signal propagation takes place

- *Domain of dependence of point P*
  points in *xy*-domain which influence the value of *f* in point *P*

- *Range of influence of point P*
  points in *xy*-domain which are influenced by the value of *f* in point *P*

Importance of classification

Different PDE types require different solution techniques because of the difference in range of influence:

- *Characteristics*
  Curves in *xy*-domain along with signal propagation takes place

- *Domain of dependence of point P*
  points in *xy*-domain which influence the value of *f* in point *P*

- *Range of influence of point P*
  points in *xy*-domain which are influenced by the value of *f* in point *P*

Importance of classification

Different PDE types require different solution techniques because of the difference in range of influence:

- *Characteristics*
  Curves in *xy*-domain along with signal propagation takes place

- *Domain of dependence of point P*
  points in *xy*-domain which influence the value of *f* in point *P*

- *Range of influence of point P*
  points in *xy*-domain which are influenced by the value of *f* in point *P*

Importance of classification

Different PDE types require different solution techniques because of the difference in range of influence:

- *Characteristics*
  Curves in *xy*-domain along with signal propagation takes place

- *Domain of dependence of point P*
  points in *xy*-domain which influence the value of *f* in point *P*

- *Range of influence of point P*
  points in *xy*-domain which are influenced by the value of *f* in point *P*

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Example elliptic PDE (boundary value problems: BVP)
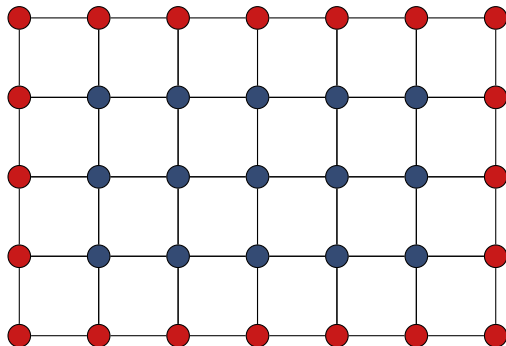
🔵 Grid point at which dependent variable has to be computed

🔴 Grid point at which boundary condition is specified

Typical example: Poisson equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f(x, y)$$

Efficiency (memory requirements, CPU time) of the numerical method is of crucial importance.

Introduction
○○○○○○○●○○

Instationary diffusion equation
○○○○○○○○○○○○○○○○○○○

Convection
○○○○○○○○

Conclusions
○○○○○
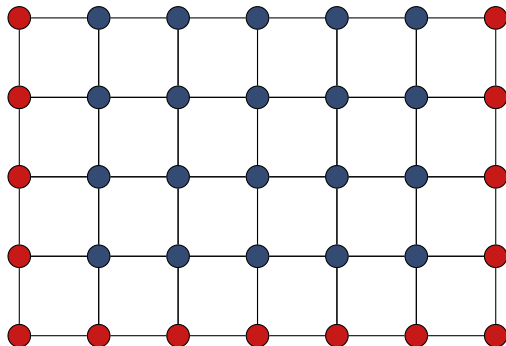
Example parabolic PDE (initial value problem: IVP)

● Grid point at which dependent
variable has to be computed
● Grid point at which boundary
condition is specified

Typical example: Poisson equation

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} = \mathcal{D} \frac{\partial^2 c}{\partial x^2} + R$$

Stability (in numerical sense) of the numerical method is of crucial importance.

Boundary conditions

- Dirichlet or fixed condition: prescribed value of f at boundary

  $f = f_0$ f_0 is a known function

- Neumann condition: prescribed value of derivative of f at boundary

  $\dfrac{\partial f}{\partial n} = q$ q is a known function

- Mixed or Robin condition: relation between f and $\dfrac{\partial f}{\partial n}$ at boundary

  $a\dfrac{\partial f}{\partial n} + bf = c$ a, b and c are known functions

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Boundary conditions

- Dirichlet or fixed condition: prescribed value of f at boundary

  $f = f_0$ f_0 is a known function

- Neumann condition: prescribed value of derivative of f at boundary

  $\dfrac{\partial f}{\partial n} = q$ q is a known function

- Mixed or Robin condition: relation between f and $\dfrac{\partial f}{\partial n}$ at boundary

  $a\dfrac{\partial f}{\partial n} + bf = c$ a, b and c are known functions

Boundary conditions

- Dirichlet or fixed condition: prescribed value of f at boundary

  $f = f_0$ f_0 is a known function

- Neumann condition: prescribed value of derivative of f at boundary

  $$\frac{\partial f}{\partial n} = q$$ q is a known function

- Mixed or Robin condition: relation between f and $\frac{\partial f}{\partial n}$ at boundary

  $$a\frac{\partial f}{\partial n} + bf = c$$ a, b and c are known functions

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Introduction
○○○○○○○○○●

Instationary diffusion equation
○○○○○○○○○○○○○○○○○○○

Convection
○○○○○○○○

Conclusions
○○○○○

Numerical solution method

Finite differences (method of lines, MOL):

1 Discretize spatial domain in discrete grid points

2 Find suitable approximation for the spatial derivatives

3 Substitute approximations in PDE, which gives a system of ODE's, one for every grid points

4 Advance in time with a suitable ODE solver

Alternative methods: collocation, Galerkin or Finite Element methods

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Today's outline

- Introduction

- Instationary diffusion equation
  - Discretization
  - Solving the diffusion equation
  - Non-linear source terms

- Convection
  - Discretization
  - Central difference scheme
  - Upwind scheme
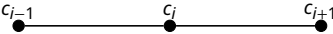
- Conclusions
  - Other methods
  - Summary

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Instationary diffusion equation (Fick's second law)

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2}, \quad \text{with} \quad \begin{array}{l} t = 0; 0 \le x \le \ell \Rightarrow c = c_0 \\ t > 0; x = 0 \Rightarrow c = c_L \\ t > 0; x = \ell \Rightarrow c = c_R \end{array}$$

Second derivative $\dfrac{\partial^2 c}{\partial x^2}$ c_{i-1} c_i c_{i+1}

Instationary diffusion equation (Fick's second law)

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2}, \quad \text{with} \quad \begin{array}{l} t = 0; 0 \le x \le \ell \Rightarrow c = c_0 \\ t > 0; x = 0 \Rightarrow c = c_L \\ t > 0; x = \ell \Rightarrow c = c_R \end{array}$$

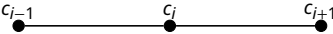Second derivative $\dfrac{\partial^2 c}{\partial x^2}$

$$c_{i+1} = c_i + \left.\frac{\partial c}{\partial x}\right|_i \Delta x + \frac{1}{2}\left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 + \frac{1}{6}\left.\frac{\partial^3 c}{\partial x^3}\right|_i \Delta x^3 + \dots$$

$$c_{i-1} = c_i - \left.\frac{\partial c}{\partial x}\right|_i \Delta x + \frac{1}{2}\left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 - \frac{1}{6}\left.\frac{\partial^3 c}{\partial x^3}\right|_i \Delta x^3 + \dots$$

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Introduction
000000000
Instationary diffusion equation
0●00000000000000000
Convection
00000000
Conclusions
00000

Instationary diffusion equation (Fick's second law)

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2}, \quad \text{with} \quad \begin{array}{l} t = 0; 0 \le x \le \ell \Rightarrow c = c_0 \\ t > 0; x = 0 \Rightarrow c = c_L \\ t > 0; x = \ell \Rightarrow c = c_R \end{array}$$

Second derivative $\dfrac{\partial^2 c}{\partial x^2}$ $\overset{c_{i-1}}{\bullet} \longrightarrow \overset{c_i}{\bullet} \longrightarrow \overset{c_{i+1}}{\bullet}$

$$c_{i+1} = c_i + \left.\frac{\partial c}{\partial x}\right|_i \Delta x + \frac{1}{2}\left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 + \frac{1}{6}\left.\frac{\partial^3 c}{\partial x^3}\right|_i \Delta x^3 + \dots$$

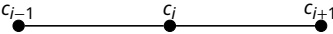$$c_{i-1} = c_i - \left.\frac{\partial c}{\partial x}\right|_i \Delta x + \frac{1}{2}\left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 - \frac{1}{6}\left.\frac{\partial^3 c}{\partial x^3}\right|_i \Delta x^3 + \dots$$

$$c_{i+1} + c_{i-1} = 2c_i + \left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 + \mathcal{O}(\Delta x^4)$$

$$\Rightarrow \left.\frac{\partial^2 c}{\partial x^2}\right|_i = \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Instationary diffusion equation (Fick's second law)

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2}, \quad \text{with} \quad \begin{array}{l} t = 0; 0 \leq x \leq \ell \Rightarrow c = c_0 \\ t > 0; x = 0 \Rightarrow c = c_L \\ t > 0; x = \ell \Rightarrow c = c_R \end{array}$$

Second derivative $\dfrac{\partial^2 c}{\partial x^2}$

$$c_{i+1} = c_i + \left.\frac{\partial c}{\partial x}\right|_i \Delta x + \frac{1}{2}\left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 + \frac{1}{6}\left.\frac{\partial^3 c}{\partial x^3}\right|_i \Delta x^3 + \ldots$$

$$c_{i-1} = c_i - \left.\frac{\partial c}{\partial x}\right|_i \Delta x + \frac{1}{2}\left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 - \frac{1}{6}\left.\frac{\partial^3 c}{\partial x^3}\right|_i \Delta x^3 + \ldots$$

$$c_{i+1} + c_{i-1} = 2c_i + \left.\frac{\partial^2 c}{\partial x^2}\right|_i \Delta x^2 + \mathcal{O}(\Delta x^4)$$

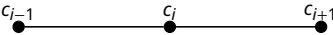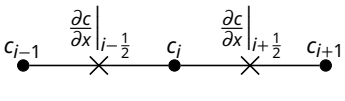$$\Rightarrow \left.\frac{\partial^2 c}{\partial x^2}\right|_i = \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

Due to symmetric discretization: second order (central discretization).

T U / e   EINDHOVEN UNIVERSITY OF TECHNOLOGY

Instationary diffusion equation (Fick's second law)

An alternative discretization:

$$\left.\frac{\partial^2 c}{\partial x^2}\right|_i = \frac{\left.\frac{\partial c}{\partial x}\right|_{i+\frac{1}{2}} - \left.\frac{\partial c}{\partial x}\right|_{i-\frac{1}{2}}}{\Delta x} + \mathcal{O}(\Delta x^2)$$

Instationary diffusion equation (Fick's second law)

An alternative discretization:

$$\left.\frac{\partial^2 c}{\partial x^2}\right|_i = \frac{\left.\frac{\partial c}{\partial x}\right|_{i+\frac{1}{2}} - \left.\frac{\partial c}{\partial x}\right|_{i-\frac{1}{2}}}{\Delta x} + \mathcal{O}(\Delta x^2)$$

$$= \frac{\frac{c_{i+1} - c_i}{\Delta x} - \frac{c_i - c_{i-1}}{\Delta x}}{\Delta x} = \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2}$$

Introduction
000000000

Instationary diffusion equation
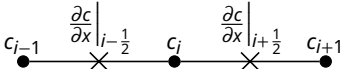0000000000000000000

Convection
00000000

Conclusions
00000

Instationary diffusion equation (Fick's second law)
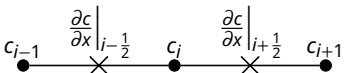
An alternative discretization:

$$\left.\frac{\partial^2 c}{\partial x^2}\right|_i = \frac{\left.\frac{\partial c}{\partial x}\right|_{i+\frac{1}{2}} - \left.\frac{\partial c}{\partial x}\right|_{i-\frac{1}{2}}}{\Delta x} + \mathcal{O}(\Delta x^2)$$

$$c_{i-1} \quad\overset{\left.\frac{\partial c}{\partial x}\right|_{i-\frac{1}{2}}}{\times}\quad c_i \quad\overset{\left.\frac{\partial c}{\partial x}\right|_{i+\frac{1}{2}}}{\times}\quad c_{i+1}$$

$$= \frac{\frac{c_{i+1} - c_i}{\Delta x} - \frac{c_i - c_{i-1}}{\Delta x}}{\Delta x} = \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2}$$

This is convenient for the derivation of $\frac{\partial}{\partial x}\left(\mathcal{D}\frac{\partial c}{\partial x}\right)$:

$$\frac{\partial}{\partial x}\left(\mathcal{D}\frac{\partial c}{\partial x}\right) = \frac{\mathcal{D}_{i+\frac{1}{2}}\left.\frac{\partial c}{\partial x}\right|_{i+\frac{1}{2}} - \mathcal{D}_{i-\frac{1}{2}}\left.\frac{\partial c}{\partial x}\right|_{i-\frac{1}{2}}}{\Delta x} = \frac{\mathcal{D}_{i+\frac{1}{2}}\frac{c_{i+1} - c_i}{\Delta x} - \mathcal{D}_{i-\frac{1}{2}}\frac{c_i - c_{i-1}}{\Delta x}}{\Delta x}$$

$$= \frac{\mathcal{D}_{i+\frac{1}{2}}\,c_{i+1} - \left(\mathcal{D}_{i+\frac{1}{2}} + \mathcal{D}_{i-\frac{1}{2}}\right)c_i + \mathcal{D}_{i-\frac{1}{2}}\,c_{i-1}}{(\Delta x)^2}$$

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Instationary diffusion equation (Fick's second law)

$$\frac{\partial^2 f}{\partial x^2}$$

$i-1 \qquad i-\frac{1}{2} \qquad i \qquad i+\frac{1}{2} \qquad i+1$

Introduction
○○○○○○○○○

Instationary diffusion equation
○○○○●○○○○○○○○○○○○○○○○

Convection
○○○○○○○○○

Conclusions
○○○○○

Instationary diffusion equation (Fick's second law)

$$\frac{\partial^2 f}{\partial x^2}$$

$i-1$ $i-\frac{1}{2}$ i $i+\frac{1}{2}$ $i+1$

$$f_{i+\frac{1}{2}} = f_i + \frac{1}{2}\Delta x \left.\frac{\partial f}{\partial x}\right|_i \Delta x + \frac{1}{2}\left(\frac{1}{2}\Delta x\right)^2 \left.\frac{\partial^2 f}{\partial x^2}\right|_i + \mathcal{O}(\Delta x^3)$$

$$f_{i-\frac{1}{2}} = f_i - \frac{1}{2}\Delta x \left.\frac{\partial f}{\partial x}\right|_i \Delta x + \frac{1}{2}\left(\frac{1}{2}\Delta x\right)^2 \left.\frac{\partial^2 f}{\partial x^2}\right|_i + \mathcal{O}(\Delta x^3)$$

$$f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} = \Delta x \frac{\partial f}{\partial x} + \mathcal{O}(\Delta x^3)$$

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Instationary diffusion equation (Fick's second law)

$\dfrac{\partial^2 f}{\partial x^2}$ $i-1$ $i-\tfrac{1}{2}$ i $i+\tfrac{1}{2}$ $i+1$

$$f_{i+\frac{1}{2}} = f_i + \frac{1}{2}\Delta x \left.\frac{\partial f}{\partial x}\right|_i \Delta x + \frac{1}{2}\left(\frac{1}{2}\Delta x\right)^2 \left.\frac{\partial^2 f}{\partial x^2}\right|_i + \mathcal{O}(\Delta x^3)$$

$$f_{i-\frac{1}{2}} = f_i - \frac{1}{2}\Delta x \left.\frac{\partial f}{\partial x}\right|_i \Delta x + \frac{1}{2}\left(\frac{1}{2}\Delta x\right)^2 \left.\frac{\partial^2 f}{\partial x^2}\right|_i + \mathcal{O}(\Delta x^3)$$

$$f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} = \Delta x \frac{\partial f}{\partial x} + \mathcal{O}(\Delta x^3)$$

$$\Rightarrow \left.\frac{\partial f}{\partial x}\right|_i = \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{\Delta x} + \mathcal{O}(\Delta x^2)$$

Symmetric discretization yields second order!

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Introduction
○○○○○○○○○

Instationary diffusion equation
○○○○○●○○○○○○○○○○○○○○

Convection
○○○○○○○○

Conclusions
○○○○○

Instationary diffusion equation: spatial discretization

Substitution of spatial derivatives yields:

$$\frac{dc_i}{dt} = \mathcal{D}\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2} \quad \text{for } i = 0, \ldots, N$$

For example, using 6 (ridiculously low number!) grid points:

Instationary diffusion equation: spatial discretization

Substitution of spatial derivatives yields:

$$\frac{dc_i}{dt} = \mathcal{D}\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2} \quad \text{for } i = 0,\ldots,N$$

For example, using 6 (ridiculously low number!) grid points:

$c_0 \quad\quad c_1 \quad\quad c_2 \quad\quad c_3 \quad\quad c_4 \quad\quad c_5$

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Instationary diffusion equation: spatial discretization

Substitution of spatial derivatives yields:

$$\frac{dc_i}{dt} = \mathcal{D}\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2} \quad \text{for } i = 0, \ldots, N$$

For example, using 6 (ridiculously low number!) grid points:

$c_0 = c_L$

Instationary diffusion equation: spatial discretization

Substitution of spatial derivatives yields:

$$\frac{dc_i}{dt} = \mathcal{D}\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2} \quad \text{for } i = 0, \ldots, N$$

For example, using 6 (ridiculously low number!) grid points:

$$c_0 = c_L$$

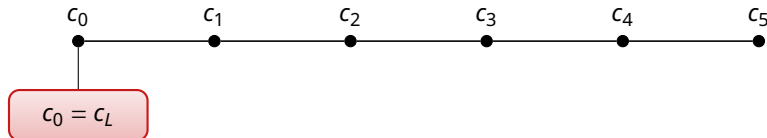$$\frac{dc_1}{dt} = \mathcal{D}\frac{c_0 - 2c_1 + c_2}{\Delta x^2}$$

Instationary diffusion equation: spatial discretization

Substitution of spatial derivatives yields:

$$\frac{dc_i}{dt} = \mathcal{D}\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2} \quad \text{for } i = 0, \ldots, N$$

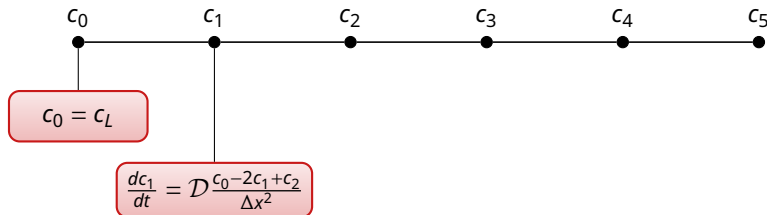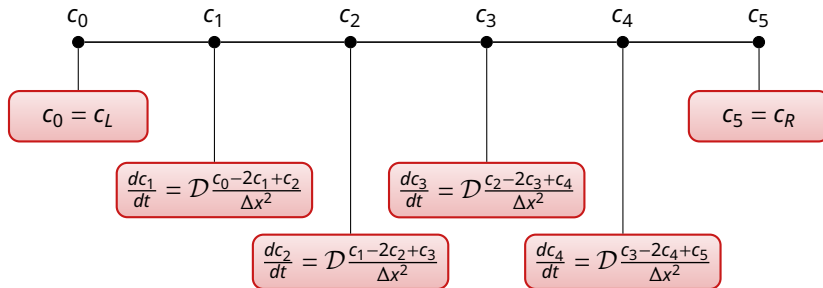For example, using 6 (ridiculously low number!) grid points:

Instationary diffusion equation: boundary conditions

Two options:

1 Keep boundary conditions as additional equations:

$$c_0 = c_L, \frac{dc_1}{dt} = \mathcal{D}\frac{c_0 - 2c_1 + c_2}{\Delta x^2}, \frac{dc_2}{dt} = \mathcal{D}\frac{c_1 - 2c_2 + c_3}{\Delta x^2},$$

$$\frac{dc_3}{dt} = \mathcal{D}\frac{c_2 - 2c_3 + c_4}{\Delta x^2}, \frac{dc_4}{dt} = \mathcal{D}\frac{c_3 - 2c_4 + c_5}{\Delta x^2}, c_5 = c_R$$

2 Substitute boundary conditions to reduce number of equations:

$$\frac{dc_1}{dt} = \mathcal{D}\frac{c_L - 2c_1 + c_2}{\Delta x^2}, \frac{dc_2}{dt} = \mathcal{D}\frac{c_1 - 2c_2 + c_3}{\Delta x^2},$$

$$\frac{dc_3}{dt} = \mathcal{D}\frac{c_2 - 2c_3 + c_4}{\Delta x^2}, \frac{dc_4}{dt} = \mathcal{D}\frac{c_3 - 2c_4 + c_R}{\Delta x^2}$$

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Instationary diffusion equation: boundary conditions

Two options:

1 Keep boundary conditions as additional equations:

$$c_0 = c_L, \frac{dc_1}{dt} = \mathcal{D}\frac{c_0 - 2c_1 + c_2}{\Delta x^2}, \frac{dc_2}{dt} = \mathcal{D}\frac{c_1 - 2c_2 + c_3}{\Delta x^2},$$

$$\frac{dc_3}{dt} = \mathcal{D}\frac{c_2 - 2c_3 + c_4}{\Delta x^2}, \frac{dc_4}{dt} = \mathcal{D}\frac{c_3 - 2c_4 + c_5}{\Delta x^2}, c_5 = c_R$$

2 Substitute boundary conditions to reduce number of equations:

$$\frac{dc_1}{dt} = \mathcal{D}\frac{c_L - 2c_1 + c_2}{\Delta x^2}, \frac{dc_2}{dt} = \mathcal{D}\frac{c_1 - 2c_2 + c_3}{\Delta x^2},$$

$$\frac{dc_3}{dt} = \mathcal{D}\frac{c_2 - 2c_3 + c_4}{\Delta x^2}, \frac{dc_4}{dt} = \mathcal{D}\frac{c_3 - 2c_4 + c_R}{\Delta x^2}$$

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Instationary diffusion equation: temporal discretization

$$\frac{dc_i}{dt} = \mathcal{D}\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2}$$

Time discretization: forward Euler (explicit)

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = \mathcal{D}\frac{c_{i-1}^n - 2c_i^n + c_{i+1}^n}{\Delta x^2}$$

$$\Rightarrow c_i^{n+1} = \text{Fo}\,c_{i-1}^n + (1 - 2\text{Fo})c_i^n + \text{Fo}\,c_{i+1}^n \quad \text{with Fo} = \frac{\mathcal{D}\Delta t}{\Delta x^2}$$

Straightforward updating (explicit equation), simple to implement in a program but stability constraint $\text{Fo} = \frac{\mathcal{D}\Delta t}{\Delta x^2} < \frac{1}{2}$!

Small $\Delta x \Rightarrow$ small $\Delta t \Rightarrow$ patience required ☹

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Introduction
000000000

Instationary diffusion equation
0000000●00000000000

Convection
00000000

Conclusions
00000

Instationary diffusion equation: temporal discretization

$$\frac{dc_i}{dt} = \mathcal{D}\frac{c_{i-1} - 2c_i + c_{i+1}}{\Delta x^2}$$

Time discretization: backward Euler (implicit)

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = \mathcal{D}\frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{\Delta x^2}$$

$$\Rightarrow -\mathrm{Fo}\,c_{i-1}^{n+1} + (1 + 2\mathrm{Fo})c_i^{n+1} - \mathrm{Fo}\,c_{i+1}^{n+1} = c_i^n \quad \text{with}\,\mathrm{Fo} = \frac{\mathcal{D}\Delta t}{\Delta x^2}$$

Requires the solution of a system of linear equations, but no stability constraints!

Note: extension to higher order schemes (with time step adaptation) straightforward. Often second or third order optimal, because for each Euler-like step in the additional order an often large system needs to be solved (not treated in this course).

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Introduction
000000000

Instationary diffusion equation
000000000●0000000000

Convection
00000000

Conclusions
00000

Solving the instationary diffusion equation: example

Solve the diffusion problem using explicit discretization:

$$\frac{\partial c_i}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} \quad \text{with} \quad \begin{aligned} &0 \le x \le \delta, \ \delta = 5 \cdot 10^{-3} \text{ m} \\ &\delta/\Delta x = 100 \text{ grid cells} \\ &\mathcal{D} = 1 \cdot 10^{-8} \text{ m}^2 \text{s}^{-1} \\ &t_{\text{end}} = 5000 \text{ s} \\ &c_L = 1 \text{ mol m}^{-3} \ c_R = 0 \text{ mol m}^{-3} \end{aligned}$$

Solving the instationary diffusion equation: example

Solve the diffusion problem using explicit discretization:

$$\frac{\partial c_i}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} \quad \text{with} \quad \begin{array}{l} 0 \le x \le \delta,\ \delta = 5 \cdot 10^{-3}\ \text{m} \\ \delta/\Delta x = 100\ \text{grid cells} \\ \mathcal{D} = 1 \cdot 10^{-8}\ \text{m}^2\,\text{s}^{-1} \\ t_{\text{end}} = 5000\ \text{s} \\ c_{\text{L}} = 1\ \text{mol}\,\text{m}^{-3}\ c_{\text{R}} = 0\ \text{mol}\,\text{m}^{-3} \end{array}$$

$$c_i^{n+1} = \text{Fo}c_{i-1}^n + (1 - 2\text{Fo})c_i^n + \text{Fo}c_{i+1}^n \quad \text{with Fo} = \frac{\mathcal{D}\Delta t}{\Delta x^2}$$

Introduction
000000000

Instationary diffusion equation
0000000●0000000000

Convection
00000000

Conclusions
00000

Solving the instationary diffusion equation: example

Solve the diffusion problem using explicit discretization:

$$\frac{\partial c_i}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} \quad \text{with} \quad \begin{aligned} &0 \leq x \leq \delta, \ \delta = 5 \cdot 10^{-3} \text{ m} \\ &\delta/\Delta x = 100 \text{ grid cells} \\ &\mathcal{D} = 1 \cdot 10^{-8} \text{ m}^2\,\text{s}^{-1} \\ &t_{\text{end}} = 5000 \text{ s} \\ &c_{\text{L}} = 1 \text{ mol}\,\text{m}^{-3} \ c_{\text{R}} = 0 \text{ mol}\,\text{m}^{-3} \end{aligned}$$

$c_i^{n+1} = \text{Fo}\, c_{i-1}^n + (1 - 2\text{Fo})c_i^n + \text{Fo}\, c_{i+1}^n \quad \text{with Fo} = \frac{\mathcal{D}\Delta t}{\Delta x^2}$

1. Initialise variables
2. Compute time step so that $\text{Fo} \leq \frac{1}{2} \Rightarrow \Delta t = 0.125\text{s}$!
3. Compute 40000 time steps times 100 grid nodes!
4. Store solution

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Introduction
000000000

Instationary diffusion equation
000000000●000000000

Convection
00000000

Conclusions
00000

Solving the instationary diffusion equation: example

Initialise the variables and matrices:

```python
import numpy as np

Nx = 100 # Nx grid points
Nt = 40000 # Nt time steps
D = 1e-8 # m/s
c_L = 1.0; c_R = 0 # mol/m3
t_end = 5000.0 # s
x_end = 5e-3 # m

# Time step and grid size
dt = t_end / Nt
dx = x_end / Nx

# Fourier number
Fo = D * dt / dx / dx

# Initial matrices for solutions (Nx times Nt)
c = np.zeros((Nt + 1, Nx + 1)) # All concentrations are zero
c[:, 0] = c_L # Concentration at the left side
c[:, Nx] = c_R # Concentration at the right side

# Grid node and time step positions
x = np.linspace(0, x_end, Nx + 1)
```

Introduction
ooooooooo
Instationary diffusion equation
oooooooooo**oo**ooooooooo
Convection
oooooooo
Conclusions
ooooo

Solving the instationary diffusion equation: example

Compute the solution (nested time-and-grid loop):

```
for n in range(Nt): # time loop
    for i in range(1, Nx): # Nested loop for grid nodes
        c[n+1, i] = Fo*c[n, i-1] + (1-2*Fo)*c[n, i] + Fo*c[n, i+1];
```

Solving the instationary diffusion equation: example

Compute the solution (nested time-and-grid loop):

```python
for n in range(Nt): # time loop
    for i in range(1, Nx): # Nested loop for grid nodes
        c[n+1, i] = Fo*c[n, i-1] + (1-2*Fo)*c[n, i] + Fo*c[n, i+1];
```
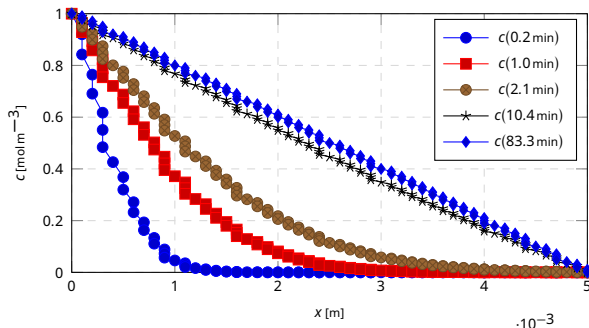
Plotting the solution at $t = \{12.5, 62.5, 125, 625, 5000\}$ s:

```python
# Output times
outt = [12.5, 62.5, 125, 625, 5000]

# Convert+round to time steps
outt_dt = [int(t // dt) for t in outt]

# Plot all time steps at once
import matplotlib.pyplot as plt
plt.plot(x, c[outt_dt, :].T)
plt.show()
```

Solving the instationary diffusion equation: example

A double-loop can impose serious computation times if the number of grid points increases:

```
for n in range(Nt - 1): # time loop
    for i in range(1, Nx): # Nested loop for grid nodes
        c[n+1, i] = Fo * c[n, i-1] + (1 - 2*Fo) * c[n, i] + Fo * c[n, i+1]
```

Remedy: vectorization. Construct a 3-point stencil Laplacian matrix first, then use the matrix product to evolve the simulation:

```
from scipy.sparse import diags

# Construct sparse matrix
e = np.ones(Nx-1)
md = np.concatenate(([1], (1 - 2 * Fo) * e, [1]))
ld = np.concatenate((Fo * e, [0]))
ud = np.concatenate(([0], Fo * e))
A = diags([ld, md, ud], offsets=[-1, 0, 1])

# Time evolution
for n in range(Nt - 1): # time loop
    c[n+1, :] = A.dot(c[n,:])
```

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Introduction
○○○○○○○○○

Instationary diffusion equation
○○○○○○○○○○○○○○●○○○○○○○

Convection
○○○○○○○○

Conclusions
○○○○○

Solving the diffusion equation implicitly

Linear system $A\boldsymbol{x} = \boldsymbol{b}$ from $-\mathrm{Fo}c_{i-1}^{n+1} + (1 + 2\mathrm{Fo})c_i^{n+1} - \mathrm{Fo}c_{i+1}^{n+1} = c_i^n$

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
-\mathrm{Fo} & (1+2\mathrm{Fo}) & -\mathrm{Fo} & 0 & \cdots & 0 \\
0 & -\mathrm{Fo} & (1+2\mathrm{Fo}) & -\mathrm{Fo} & \cdots & 0 \\
0 & 0 & -\mathrm{Fo} & (1+2\mathrm{Fo}) & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1
\end{pmatrix}
\cdot
\begin{pmatrix}
c_0^{n+1} \\
c_1^{n+1} \\
c_2^{n+1} \\
c_3^{n+1} \\
\vdots \\
c_m^{n+1}
\end{pmatrix}
=
\begin{pmatrix}
c_0^n \\
c_1^n \\
c_2^n \\
c_3^n \\
\vdots \\
c_m^n
\end{pmatrix}
$$

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Solving the diffusion equation implicitly

Linear system $A\boldsymbol{x} = \boldsymbol{b}$ from $-\text{Fo}c_{i-1}^{n+1} + (1+2\text{Fo})c_i^{n+1} - \text{Fo}c_{i+1}^{n+1} = c_i^n$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -\text{Fo} & (1+2\text{Fo}) & -\text{Fo} & 0 & \cdots & 0 \\ 0 & -\text{Fo} & (1+2\text{Fo}) & -\text{Fo} & \cdots & 0 \\ 0 & 0 & -\text{Fo} & (1+2\text{Fo}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0^{n+1} \\ c_1^{n+1} \\ c_2^{n+1} \\ c_3^{n+1} \\ \vdots \\ c_m^{n+1} \end{pmatrix} = \begin{pmatrix} c_0^n \\ c_1^n \\ c_2^n \\ c_3^n \\ \vdots \\ c_m^n \end{pmatrix}$$

$1 \times c_0^{n+1} = c_0^n$ (boundary condition)

Solving the diffusion equation implicitly

Linear system $A\boldsymbol{x} = \boldsymbol{b}$ from $-\text{Fo}c_{i-1}^{n+1} + (1 + 2\text{Fo})c_i^{n+1} - \text{Fo}c_{i+1}^{n+1} = c_i^n$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -\text{Fo} & (1 + 2\text{Fo}) & -\text{Fo} & 0 & \cdots & 0 \\ 0 & -\text{Fo} & (1 + 2\text{Fo}) & -\text{Fo} & \cdots & 0 \\ 0 & 0 & -\text{Fo} & (1 + 2\text{Fo}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0^{n+1} \\ c_1^{n+1} \\ c_2^{n+1} \\ c_3^{n+1} \\ \vdots \\ c_m^{n+1} \end{pmatrix} = \begin{pmatrix} c_0^n \\ c_1^n \\ c_2^n \\ c_3^n \\ \vdots \\ c_m^n \end{pmatrix}$$

$1 \times c_0^{n+1} = c_0^n$ (boundary condition)

$-\text{Fo}c_0^{n+1} + (1 + 2\text{Fo})c_1^{n+1} - \text{Fo}c_2^{n+1} = c_1^n$

Solving the diffusion equation implicitly

Linear system $A\boldsymbol{x} = \boldsymbol{b}$ from $-\text{Fo}c_{i-1}^{n+1} + (1 + 2\text{Fo})c_i^{n+1} - \text{Fo}c_{i+1}^{n+1} = c_i^n$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -\text{Fo} & (1+2\text{Fo}) & -\text{Fo} & 0 & \cdots & 0 \\ 0 & -\text{Fo} & (1+2\text{Fo}) & -\text{Fo} & \cdots & 0 \\ 0 & 0 & -\text{Fo} & (1+2\text{Fo}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0^{n+1} \\ c_1^{n+1} \\ c_2^{n+1} \\ c_3^{n+1} \\ \vdots \\ c_m^{n+1} \end{pmatrix} = \begin{pmatrix} c_0^n \\ c_1^n \\ c_2^n \\ c_3^n \\ \vdots \\ c_m^n \end{pmatrix}$$

$1 \times c_0^{n+1} = c_0^n$ (boundary condition)

$-\text{Fo}c_0^{n+1} + (1 + 2\text{Fo})c_1^{n+1} - \text{Fo}c_2^{n+1} = c_1^n$

$-\text{Fo}c_1^{n+1} + (1 + 2\text{Fo})c_2^{n+1} - \text{Fo}c_3^{n+1} = c_2^n$

Solving the diffusion equation implicitly

Linear system $A\boldsymbol{x} = \boldsymbol{b}$ from $-\mathrm{Fo}c_{i-1}^{n+1} + (1 + 2\mathrm{Fo})c_i^{n+1} - \mathrm{Fo}c_{i+1}^{n+1} = c_i^n$

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
-\mathrm{Fo} & (1 + 2\mathrm{Fo}) & -\mathrm{Fo} & 0 & \cdots & 0 \\
0 & -\mathrm{Fo} & (1 + 2\mathrm{Fo}) & -\mathrm{Fo} & \cdots & 0 \\
0 & 0 & -\mathrm{Fo} & (1 + 2\mathrm{Fo}) & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1
\end{pmatrix}
\cdot
\begin{pmatrix}
c_0^{n+1} \\
c_1^{n+1} \\
c_2^{n+1} \\
c_3^{n+1} \\
\vdots \\
c_m^{n+1}
\end{pmatrix}
=
\begin{pmatrix}
c_0^n \\
c_1^n \\
c_2^n \\
c_3^n \\
\vdots \\
c_m^n
\end{pmatrix}
$$

$1 \times c_0^{n+1} = c_0^n$ (boundary condition)

$-\mathrm{Fo}c_0^{n+1} + (1 + 2\mathrm{Fo})c_1^{n+1} - \mathrm{Fo}c_2^{n+1} = c_1^n$

$-\mathrm{Fo}c_1^{n+1} + (1 + 2\mathrm{Fo})c_2^{n+1} - \mathrm{Fo}c_3^{n+1} = c_2^n$

$-\mathrm{Fo}c_2^{n+1} + (1 + 2\mathrm{Fo})c_3^{n+1} - \mathrm{Fo}c_4^{n+1} = c_3^n$

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Solving the diffusion equation implicitly

Linear system $A\boldsymbol{x} = \boldsymbol{b}$ from $-\text{Fo}c_{i-1}^{n+1} + (1 + 2\text{Fo})c_i^{n+1} - \text{Fo}c_{i+1}^{n+1} = c_i^n$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -\text{Fo} & (1+2\text{Fo}) & -\text{Fo} & 0 & \cdots & 0 \\ 0 & -\text{Fo} & (1+2\text{Fo}) & -\text{Fo} & \cdots & 0 \\ 0 & 0 & -\text{Fo} & (1+2\text{Fo}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0^{n+1} \\ c_1^{n+1} \\ c_2^{n+1} \\ c_3^{n+1} \\ \vdots \\ c_m^{n+1} \end{pmatrix} = \begin{pmatrix} c_0^n \\ c_1^n \\ c_2^n \\ c_3^n \\ \vdots \\ c_m^n \end{pmatrix}$$

$1 \times c_0^{n+1} = c_0^n$ (boundary condition)

$-\text{Fo}c_0^{n+1} + (1 + 2\text{Fo})c_1^{n+1} - \text{Fo}c_2^{n+1} = c_1^n$

$-\text{Fo}c_1^{n+1} + (1 + 2\text{Fo})c_2^{n+1} - \text{Fo}c_3^{n+1} = c_2^n$

$-\text{Fo}c_2^{n+1} + (1 + 2\text{Fo})c_3^{n+1} - \text{Fo}c_4^{n+1} = c_3^n$

$1 \times c_m^{n+1} = c_m^n$ (boundary condition)

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Introduction
000000000

Instationary diffusion equation
000000000000000●00000

Convection
00000000

Conclusions
00000

Solving the diffusion equation implicitly in Python

To solve the linear system, we need to define matrix *A*. It is clear that storing many zeros is not efficient in terms of memory. We use a *sparse matrix* format. Two alternative ways to set up the matrix:

Set individual elements of the matrix:

```python
from scipy.sparse import lil_matrix

# Bands in matrix (internal cells)
A = lil_matrix((Nx+1, Nx+1))
for i in range(1, Nx):
    A[i, i-1] = -Fo
    A[i, i] = 1 + 2*Fo
    A[i, i+1] = -Fo

# Set boundary cells, only main diag:
A[0, 0] = 1 # Left
A[Nx, Nx] = 1 # Right
```

Set matrix using bands:

```python
from scipy.sparse import diags

# Bands in matrix (internal cells)
e = np.ones(Nx-1) # Ones for internal cells
md = np.concatenate(([1], e * (1 + 2*Fo), [1])) #
    Main diagonal
ld = np.concatenate((-e * Fo, [0])) # Lower diagonal
ud = np.concatenate(([0], -e * Fo)) # Upper diagonal
A = diags([ld, md, ud], offsets=[-1, 0, 1])
```

Note: The first argument of `diags` defines each column as a diagonal, starting at row 0 (for lower-diagonal) or column 0 (for upper-diagonal).

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Solving the diffusion equation implicitly in Python

The command `plt.spy(A)` shows a figure with the non-zero positions.

nz = 14

Introduction
000000000

Instationary diffusion equation
00000000000000000000

Convection
00000000

Conclusions
00000
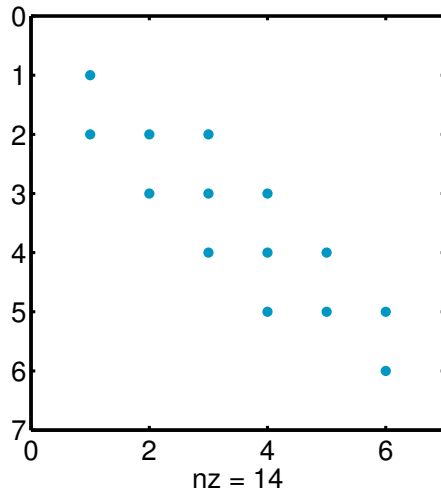
Solving the diffusion equation implicitly in Python

The concentration matrix is initialised and the boundary conditions are set as follows:

```python
# Initial matrices for solutions (Nx times Nt)
c = np.zeros((Nt+1, Nx+1)) # All concentrations are zero
c[:, 0] = c_L # Concentration at left side
c[:, Nx] = c_R # Concentration at right side
```

The right hand side vector (**b**) can now be set during the time-loop:

```python
from scipy.sparse.linalg import spsolve

for n in range(Nt-1): # time loop
    b = c[n, :] # Set right hand side
    solX = spsolve(A, b) # Solve linear system
    c[n+1, :] = solX # Store solution each time step
```

Introduction
○○○○○○○○○

Instationary diffusion equation
○○○○○○○○○○○○○○○○●○○

Convection
○○○○○○○○○

Conclusions
○○○○○

Solving the diffusion equation implicitly in Matlab

Plotting the solution at $t = \{12.5, 62.5, 125, 625, 5000\}$ s.
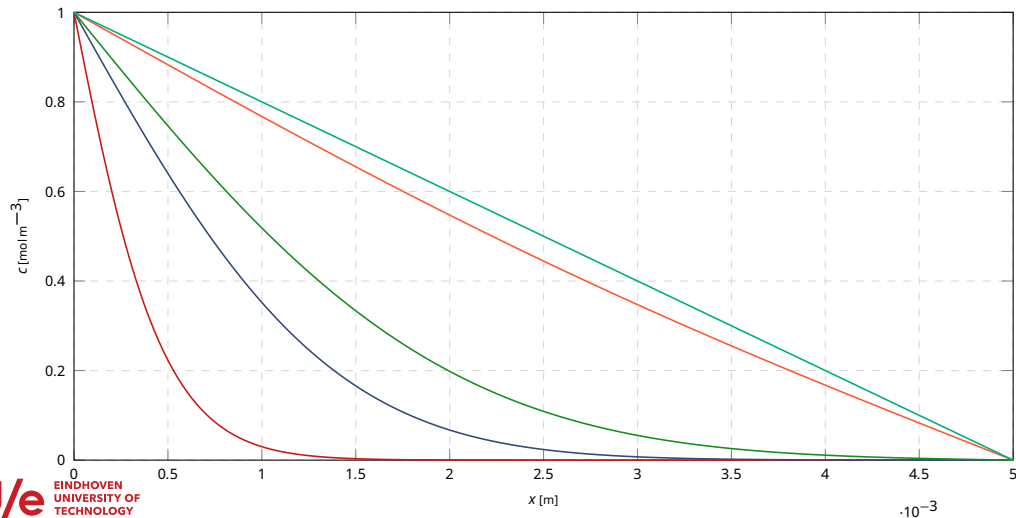
About explicit vs. implicit solutions

- Explicit solution:
  - Easy to implement
  - Very small time steps required.
  - This problem took about 0.5 s.
- Implicit solution:
  - Harder to implement, needs sparse matrix solver
  - No stability constraint
  - This problem took about 0.05 s
- The difference will become much larger for systems with e.g. more grid nodes!

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Extension with non-linear source terms

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} + R(c) \quad \text{with} \quad \begin{array}{l} t = 0; 0 \leq x \leq \ell \Rightarrow c = c_0 \\ t > 0; x = 0 \Rightarrow c = c_L \\ t > 0; x = \ell \Rightarrow c = c_R \end{array}$$

Extension with non-linear source terms

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} + R(c) \quad \text{with} \quad \begin{array}{l} t = 0; 0 \leq x \leq \ell \Rightarrow c = c_0 \\ t > 0; x = 0 \Rightarrow c = c_L \\ t > 0; x = \ell \Rightarrow c = c_R \end{array}$$

- Forward Euler (explicit): simply add to right-hand side

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = \mathcal{D}\frac{c_{i-1}^n - 2c_i^n + c_{i+1}^n}{\Delta x^2} + R(c_i^n)$$

$$\Rightarrow c_i^{n+1} = \text{Fo}c_{i-1}^n + (1 - 2\text{Fo})c_i^n + \text{Fo}c_{i+1}^n + R_i^n \Delta t$$

Introduction
000000000

Instationary diffusion equation
0000000000000000000●

Convection
00000000

Conclusions
00000

Extension with non-linear source terms

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} + R(c) \quad \text{with} \quad \begin{array}{l} t = 0; 0 \le x \le \ell \Rightarrow c = c_0 \\ t > 0; x = 0 \Rightarrow c = c_L \\ t > 0; x = \ell \Rightarrow c = c_R \end{array}$$

• Forward Euler (explicit): simply add to right-hand side

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = \mathcal{D}\frac{c_{i-1}^n - 2c_i^n + c_{i+1}^n}{\Delta x^2} + R(c_i^n)$$

$$\Rightarrow c_i^{n+1} = \text{Fo}c_{i-1}^n + (1 - 2\text{Fo})c_i^n + \text{Fo}c_{i+1}^n + R_i^n \Delta t$$

• Backward Euler (implicit): linearization required

$$R(c_i^{n+1}) = R(c_i^n) + \left.\frac{dR}{dc}\right|_i^n (c_i^{n+1} - c_i^n)$$

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = \mathcal{D}\frac{c_{i-1}^{n+1} - 2c_i^{n+1} + c_{i+1}^{n+1}}{\Delta x^2} + R(c_i^{n+1})$$

$$\Rightarrow -\text{Fo}c_{i-1}^{n+1} + (1 + 2\text{Fo} - \left.\frac{dR}{dc}\right|_i^n \Delta t)c_i^{n+1} - \text{Fo}c_{i+1}^{n+1} = c_i^n + \left(R_i^n - \left.\frac{dR}{dc}\right|_i^n c_i^n\right)\Delta t$$

Today's outline

- ● Introduction

- ● Instationary diffusion equation
    - • Discretization
    - • Solving the diffusion equation
    - • Non-linear source terms

- ● Convection
    - • Discretization
    - • Central difference scheme
    - • Upwind scheme

- ● Conclusions
    - • Other methods
    - • Summary

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Extension with convection terms

$$\frac{\partial c}{\partial t} = \mathcal{D}\frac{\partial^2 c}{\partial x^2} - u\frac{\partial c}{\partial x} + R$$

Discretization of first derivative $\frac{dc}{dx}$,
looks simple but is numerical headache!

Central discretization:

$$\frac{dc}{dx} = \frac{c_{i+1} - c_{i-1}}{2\Delta x}$$

$\Rightarrow$ simple and easy, too bad it doesn't work: yields unstable solutions if convection dominated.

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Central difference scheme of 1st derivative
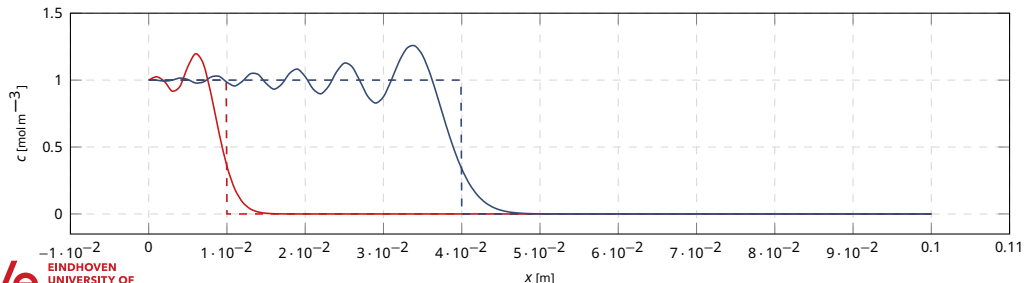
Unsteady convection:

$$\frac{\partial c}{\partial t} = -u \frac{\partial c}{\partial x}$$

Central difference for first derivative:

$$\frac{dc}{dx} = \frac{c_{i+1} - c_{i-1}}{2\Delta x}$$

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

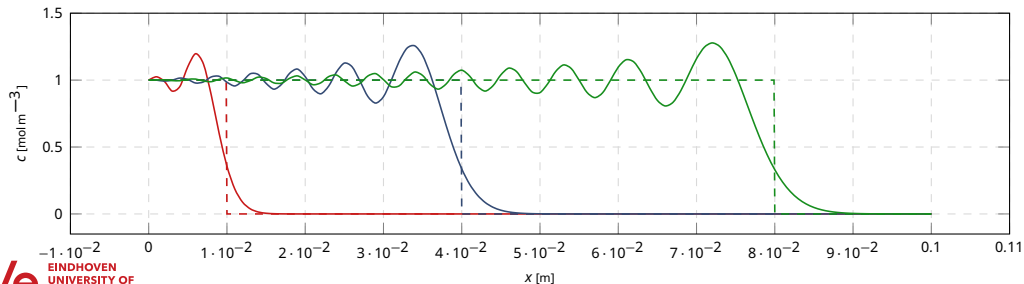# Central difference scheme of 1st derivative

Unsteady convection:

$$\frac{\partial c}{\partial t} = -u \frac{\partial c}{\partial x}$$

Central difference for first derivative:

$$\frac{dc}{dx} = \frac{c_{i+1} - c_{i-1}}{2\Delta x}$$

Forward Euler discretization of temporal and spatial domain:

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = -u \frac{c_{i+1} - c_{i-1}}{2\Delta x} \Rightarrow c_i^{n+1} = c_i^n - u \frac{c_{i+1}^n - c_{i-1}^n}{2\Delta x} \Delta t$$

Central difference scheme of 1st derivative

Unsteady convection:
$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x}$$

Central difference for first derivative:
$$\frac{dc}{dx} = \frac{c_{i+1} - c_{i-1}}{2\Delta x}$$

Forward Euler discretization of temporal and spatial domain:

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = -u\frac{c_{i+1} - c_{i-1}}{2\Delta x} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_{i+1}^n - c_{i-1}^n}{2\Delta x}\Delta t$$

Central difference scheme of 1st derivative

Unsteady convection:

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x}$$

Central difference for first derivative:

$$\frac{dc}{dx} = \frac{c_{i+1} - c_{i-1}}{2\Delta x}$$

Forward Euler discretization of temporal and spatial domain:

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = -u\frac{c_{i+1} - c_{i-1}}{2\Delta x} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_{i+1}^n - c_{i-1}^n}{2\Delta x}\Delta t$$

Central difference scheme of 1st derivative

Unsteady convection:
$$\frac{\partial c}{\partial t} = -u \frac{\partial c}{\partial x}$$

Central difference for first derivative:
$$\frac{dc}{dx} = \frac{c_{i+1} - c_{i-1}}{2\Delta x}$$

Forward Euler discretization of temporal and spatial domain:

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = -u \frac{c_{i+1} - c_{i-1}}{2\Delta x} \Rightarrow c_i^{n+1} = c_i^n - u \frac{c_{i+1}^n - c_{i-1}^n}{2\Delta x} \Delta t$$

Extension with convection terms

Solution: upwind discretization, like CSTR's in series:
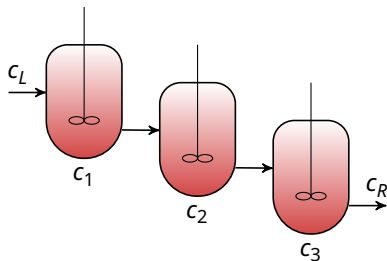
First order upwind: $-u\dfrac{dc}{dx}\bigg|_i = \begin{cases} -u\dfrac{c_i - c_{i-1}}{\Delta x} & \text{if } u \geq 0 \\[3mm] -u\dfrac{c_{i+1} - c_i}{\Delta x} & \text{if } u < 0 \end{cases}$

Extension with convection terms

Solution: upwind discretization, like CSTR's in series:
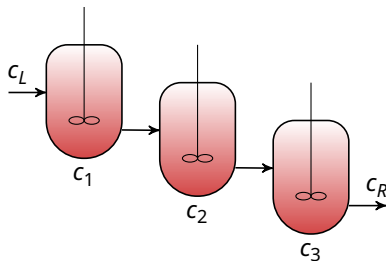
First order upwind: $-u\dfrac{dc}{dx}\bigg|_i = \begin{cases} -u\dfrac{c_i - c_{i-1}}{\Delta x} & \text{if } u \geq 0 \\[2ex] -u\dfrac{c_{i+1} - c_i}{\Delta x} & \text{if } u < 0 \end{cases}$ Stable if $Co = \frac{u\Delta t}{\Delta x} < 1$ (with Co the

Courant number). However, only 1$^{\text{st}}$ order accurate (large smearing of concentration fronts). Higher order upwind requires TVD schemes (trick of the trade)...

Introduction
○○○○○○○○○

Instationary diffusion equation
○○○○○○○○○○○○○○○○○○○○○

Convection
○○○○○●○○○

Conclusions
○○○○○

First order upwind scheme of 1st derivative

Unsteady convection:

$$\frac{\partial c}{\partial t} = -u \frac{\partial c}{\partial x}$$

Upwind scheme for first derivative:

$$-u\frac{dc}{dx}\bigg|_i = \begin{cases} -u\dfrac{c_i - c_{i-1}}{\Delta x} & \text{if } u \geq 0 \\[3mm] -u\dfrac{c_{i+1} - c_i}{\Delta x} & \text{if } u < 0 \end{cases}$$

Introduction
000000000

Instationary diffusion equation
0000000000000000000

Convection
00000●000

Conclusions
00000

First order upwind scheme of 1st derivative

Unsteady convection:
$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x}$$

Upwind scheme for first derivative:
$$-u\frac{dc}{dx}\bigg|_i = \begin{cases} -u\dfrac{c_i - c_{i-1}}{\Delta x} & \text{if } u \geq 0 \\[2ex] -u\dfrac{c_{i+1} - c_i}{\Delta x} & \text{if } u < 0 \end{cases}$$

Forward Euler discretization of temporal and spatial domain:

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = -u\frac{c_{i+1} - c_{i-1}}{2\Delta x}$$

$$\Rightarrow c_i^{n+1} = \begin{cases} c_i^n - u\Delta t\dfrac{c_i - c_{i-1}}{\Delta x} & \text{if } u \geq 0 \\[2ex] c_i^n - u\Delta t\dfrac{c_{i+1} - c_i}{\Delta x} & \text{if } u < 0 \end{cases}$$
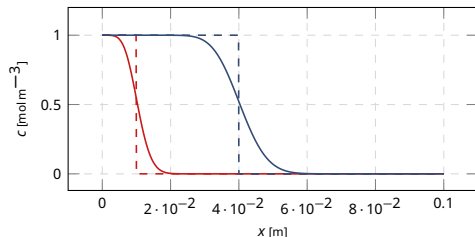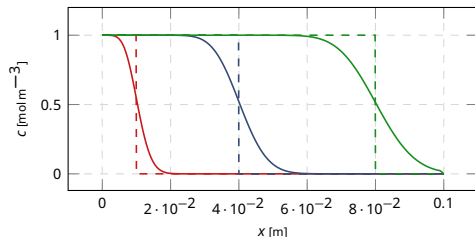
Upwind scheme: example

Unsteady convection through a pipe:

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} \quad \text{with} \quad u = 0.1\,\text{m}\,\text{s}^{-1} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_i - c_{i-1}}{\Delta x}\Delta t$$

Upwind scheme: example

Unsteady convection through a pipe:

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} \quad \text{with} \quad u = 0.1\,\mathrm{m\,s^{-1}} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_i - c_{i-1}}{\Delta x}\Delta t$$
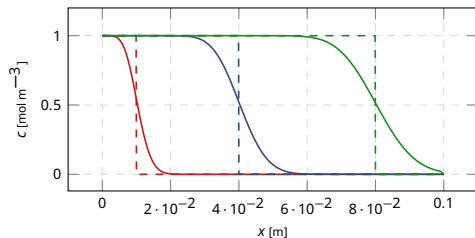
Using 100 grid cells
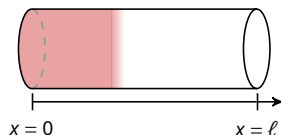
Upwind scheme: example

Unsteady convection through a pipe:

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} \quad \text{with} \quad u = 0.1\,\mathrm{m\,s^{-1}} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_i - c_{i-1}}{\Delta x}\Delta t$$
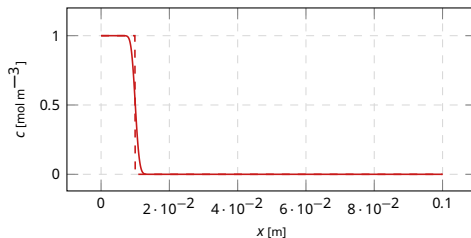
Using 100 grid cells

Upwind scheme: example

Unsteady convection through a pipe:

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} \quad \text{with} \quad u = 0.1\,\text{m}\,\text{s}^{-1} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_i - c_{i-1}}{\Delta x}\Delta t$$

Using 100 grid cells

Introduction
○○○○○○○○○○

Instationary diffusion equation
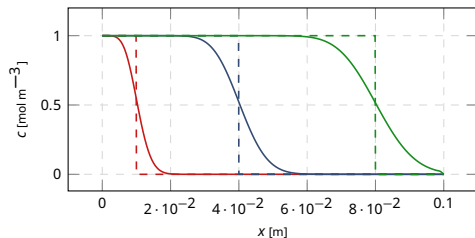○○○○○○○○○○○○○○○○○○○○

Convection
○○○○○○●○○

Conclusions
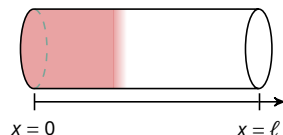○○○○○

Upwind scheme: example

Unsteady convection through a pipe:
$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} \quad \text{with} \quad u = 0.1\,\mathrm{m\,s^{-1}} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_i - c_{i-1}}{\Delta x}\Delta t$$

Using 100 grid cells

Using 1000 grid cells

Introduction
○○○○○○○○○○
Instationary diffusion equation
○○○○○○○○○○○○○○○○○○○○
Convection
○○○○○○●○○
Conclusions
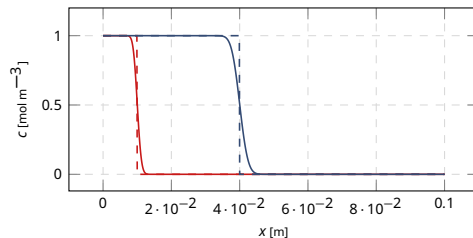○○○○○

Upwind scheme: example

Unsteady convection through a pipe:

$$\frac{\partial c}{\partial t} = -u \frac{\partial c}{\partial x} \quad \text{with} \quad u = 0.1 \, \text{m s}^{-1} \Rightarrow c_i^{n+1} = c_i^n - u \frac{c_i - c_{i-1}}{\Delta x} \Delta t$$

Using 100 grid cells

Using 1000 grid cells

Introduction
○○○○○○○○○○
Instationary diffusion equation
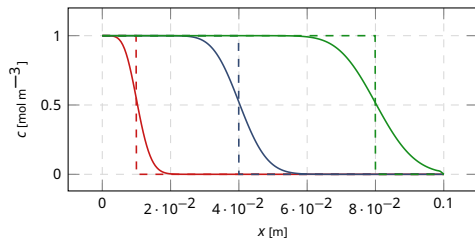○○○○○○○○○○○○○○○○○○○○
Convection
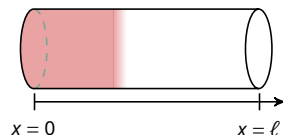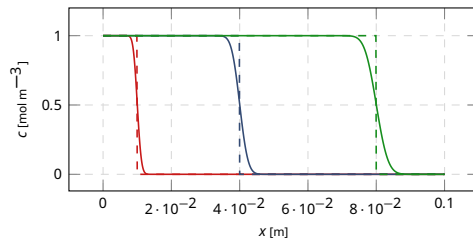○○○○○●○○○
Conclusions
○○○○○

Upwind scheme: example

Unsteady convection through a pipe:
$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} \quad \text{with} \quad u = 0.1\,\text{m s}^{-1} \Rightarrow c_i^{n+1} = c_i^n - u\frac{c_i - c_{i-1}}{\Delta x}\Delta t$$

Using 100 grid cells

Using 1000 grid cells

Introduction
000000000

Instationary diffusion equation
00000000000000000000

Convection
0000000000

Conclusions
00000

Central difference and upwind in Python

The results from the previous slides were computed using this script:

```python
import numpy as np

Nx, Nt = 1000, 10000 # Nc grid points Nt time steps
u = 0.001 # m/s
c_in = 1.0 # mol/m3
t_end = 100.0 # s
x_end = 0.1 # m

# Time step and grid size
dt, dx = t_end/Nt, x_end/Nx

# Courant number
Co = u*dt/dx

# Initial matrices for solutions (Nx times Nt)
c1 = np.zeros((Nt+1, Nx+1)) # All concentrations are zero
c1[:, 0] = c_in # Concentration at inlet (all time steps)
an = np.copy(c1)
c2 = np.copy(c1) # Analytical and upwind solution

# Grid node and time step positions
x = np.linspace(0, x_end, Nx+1)
t = np.linspace(0, t_end, Nt+1)
```

TU/e TECHNOLOGY

Central difference and upwind in Python

(continued)

```python
for n in range(Nt): # time loop
    for i in range(1, Nx): # Nested loop for grid nodes
        # Central difference
        c1[n+1, i] = c1[n, i] - u*((c1[n, i+1] - c1[n, i-1])/(2*dx))*dt
        # Upwind
        c2[n+1, i] = c2[n, i] - u*((c2[n, i] - c2[n, i-1])/dx)*dt
        # Analytical
        an[n+1, i] = (x[i] < u*t[n+1])*c_in
```

Today's outline

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Introduction
○○○○○○○○○

Instationary diffusion equation
○○○○○○○○○○○○○○○○○○

Convection
○○○○○○○○○

Conclusions
○●○○○

Extension to systems of PDE's

- Explicit methods: straightforward extension
- Implicit methods: yields block-tridiagonal matrix (note ordering of equations: all variables per grid cell)

Introduction
○○○○○○○○○

Instationary diffusion equation
○○○○○○○○○○○○○○○○○○○

Convection
○○○○○○○○○

Conclusions
○○●○○

Extension to 2D or 3D systems

Spatial discretization in 2 directions — different methods available:

- Explicit
- Fully implicit
  - 1D gives tri-diagonal matrix
  - 2D gives penta-diagonal matrix
  - 3D gives hepta-diagonal matrix

  Use of dedicated matrix solvers (e.g. ICCG, multigrid, ...)
- Alternating direction implicit (ADI)
  - Per direction implicit, but still overall unconditionally stable

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Further extensions for parabolic PDEs

- Higher order temporal discretization (multi-step) with time step adaptation
- Non-uniform grids with automatic grid adaptation
- Higher-order discretization methods, especially higher order TVD (flux delimited) schemes for convective fluxes (e.g. WENO schemes)
- Higher-order finite volume schemes (Riemann solvers)

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Summary

- Several classes of PDEs were introduced
  - Elliptic, Parabolic, Hyperbolic PDEs
- Diffusion equation: discretization of temporal and spatial domain was discussed
  - Solutions of the diffusion equation using explicit and implicit methods
  - How to add non-linear source terms
- Convection: upwind vs. central difference schemes

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY