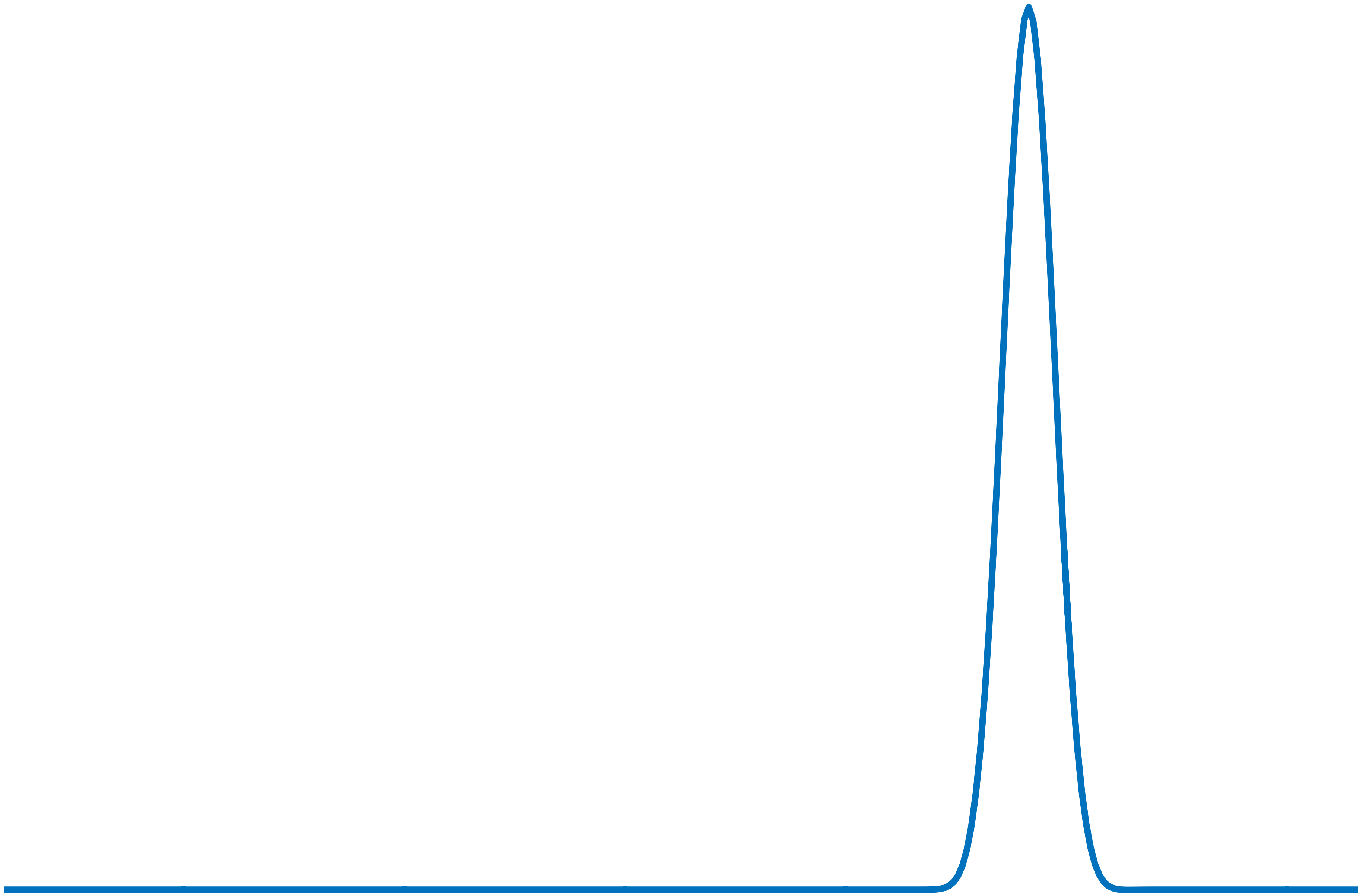
Numerical simulation of 1D convection - diffusion equation



P. Billuart
P. Parmentier
G. Winckelmans

LMECA2660
Homework

Goals

- Validation of the periodic approximation
- Development of new high order decentered scheme
  - ★ investigate the dispersion and dissipation error and assessment against other schemes
- Write a C code to
  - ★ investigate convection and convection-diffusion phenomena,
  - ★ investigate the numerical properties of different discretizations of the convective and diffusive terms.

Problem statement

- Convection-diffusion equation (1D-problem) :

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = \nu\,\frac{\partial^2 u}{\partial x^2}$$

with a Gaussian function as initial condition

$$u(x,0) = \frac{Q}{\sqrt{\pi\,\sigma_0^2}}\,\exp\left(-\frac{x^2}{\sigma_0^2}\right)$$

- Analytical solution (diffusing function moving at a constant velocity c)

$$u(x,t) = \frac{Q}{\sqrt{\pi\,(\sigma_0^2 + 4\nu t)}}\,\exp\left(-\frac{(x-ct)^2}{(\sigma_0^2 + 4\nu t)}\right)$$

Periodic approximation

- The Gaussian function has an unbounded support

  -> impossible to solve that problem numerically

- We use a periodic domain of period L instead of an unbounded domain
  - ⋆ Valid approximation as long as $L >> \sigma_0$

The first part of the homework will allow to assess the validity and the quality of this approximation.

Reminder on Fourier Transforms and Fourier Series

- Fourier Transform (not periodic and continuous function)

The Fourier transform $\widehat{f}(k)$ of a function $f(x)$ is defined as :

$$\widehat{f}(k) = \mathcal{F}(f(x)) \quad = \quad \int_{-\infty}^{\infty} f(x) \exp(-\imath kx)\, dx ,$$

while the inverse transform is defined as :

$$f(x) = \mathcal{F}^{-1}\left(\widehat{f}(k)\right) \quad = \quad \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{f}(k) \exp(\imath kx)\, dk .$$

- For a **periodic** signal: Fourier series (discrete)

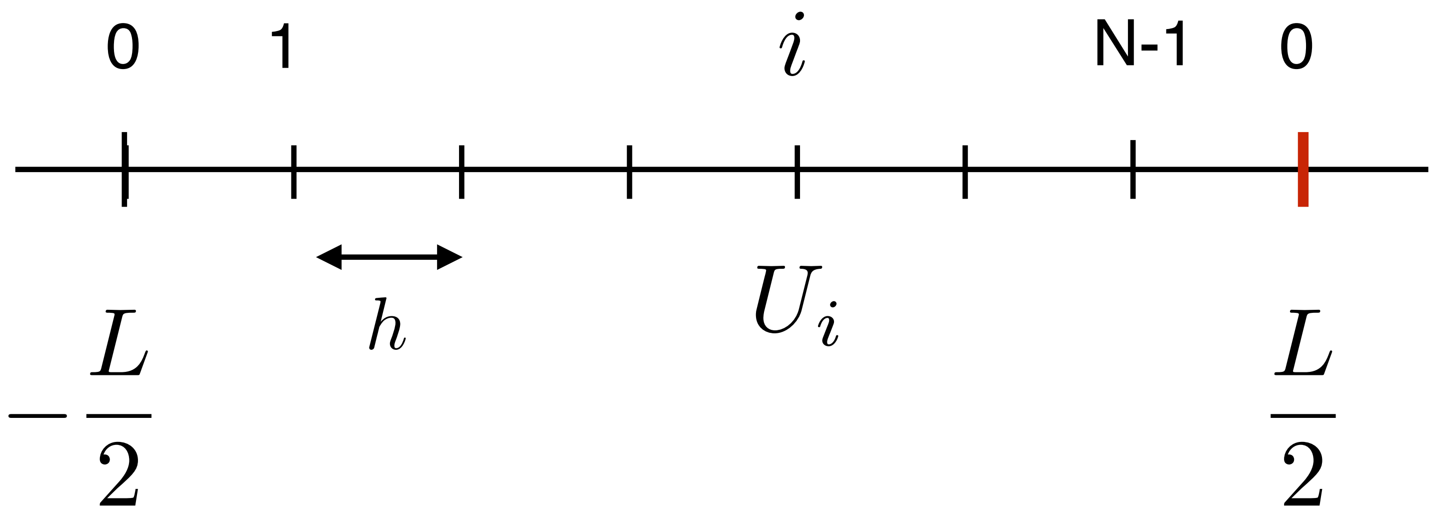
$$f(x) \quad = \quad \sum_{j=-\infty}^{\infty} \widehat{F}(k_j) \exp(\imath k_j x) ,$$

$$\text{where} \quad \widehat{F}(k_j) \quad = \quad \frac{1}{L} \int_{-L/2}^{L/2} f(x) \exp(-\imath k_j x)\, dx$$

Reminder on Fourier Transforms and Fourier Series

- Discretization of the function

Spatial domain



$$x_i = -\frac{L}{2} + ih$$

- Discrete and periodic Fourier Series

Frequency domain

$$f_i = f(x_i) = \sum_{j=-N/2}^{N/2} \widehat{F}(k_j) \exp(\imath k_j x_i) \; ,$$

$$\text{where} \quad \widehat{F}_j = \widehat{F}(k_j) = \frac{1}{N} \sum_{i=0}^{N-1} f(x_i) \exp(-\imath k_j x_i) \; .$$

Reminder on Fourier Transforms and Fourier Series

- The Gaussian function considered is a real-valued function. Therefore,
  - ★ Coefficients $\widehat{F}(k_j)$ are complex conjugates (real-valued function):

$$\widehat{F}(-k_j) = \widehat{F}_r(-k_j) + \imath\, \widehat{F}_\imath(-k_j) = \left(\widehat{F}(k_j)\right)^* = \widehat{F}_r(k_j) - \imath\, \widehat{F}_\imath(k_j)$$

  - ★ N being even, a flip-flop mode exists.

$$f(x_i) = \underbrace{\widehat{F}_r(k_0)}_{\textit{Zero mode}} + \sum_{j=1}^{N/2-1} 2\left[\widehat{F}_r(k_j)\cos(k_j x_i) - \widehat{F}_\imath(k_j)\sin(k_j x_i)\right] + \underbrace{2\widehat{F}_r(k_{N/2})\cos(k_{N/2} x_i)}_{\textit{Flip-flop mode}}$$

Discrete Fourier Series in Matlab

- The Fourier coefficients $\widehat{F}(k)$ may be obtained using the fft function in Matlab.
- Caution: the result has to be divided by the N (see the documentation, this is not performed by Matlab).
- Except that detail, Matlab provides the N coefficients in the following form:

Indices of the vector returned by the fft function

1	2		N/2	N/2+1	N/2+2		N
$\widehat{F}(k_0)$	$\widehat{F}(k_1)$	$\cdots$	$\widehat{F}(k_{N/2-1})$	$\widehat{F}(k_{N/2})$	$\widehat{F}(k_{-(\frac{N}{2}-1)})$	$\cdots$	$\widehat{F}(k_{-1})$

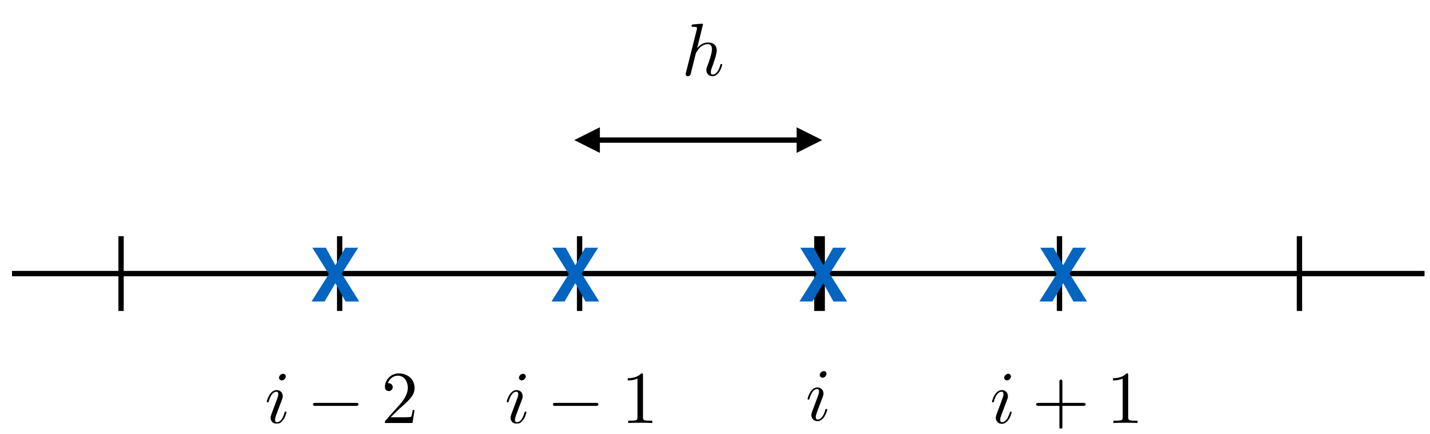
Zero mode (purely real)

Flip-flop mode (purely real)

Development of a partially decentered scheme

- A new discretization for the convective term

$$\frac{\partial u}{\partial t} + \boxed{c\,\frac{\partial u}{\partial x}} = \nu\,\frac{\partial^2 u}{\partial x^2}$$



★ Order of convergence

★ Truncation error

★ Comparison of the dispersion and/or diffusion errors with respect to the exact case (modal analysis: obtain the modified wavenumbers of the decentered scheme)

Produce a C-code

$$\boxed{\frac{\partial u}{\partial t}} + \boxed{c\,\frac{\partial u}{\partial x}} = \boxed{\nu\,\frac{\partial^2 u}{\partial x^2}}$$

Temporal integration: RK4

Convective term:

- *Second order explicit (E2)*
- *E4*
- *Decentered scheme (DS)*
- *Fourth order implicit (I4)*
- *I6*

Diffusive term:

- *Second order explicit (E2)*

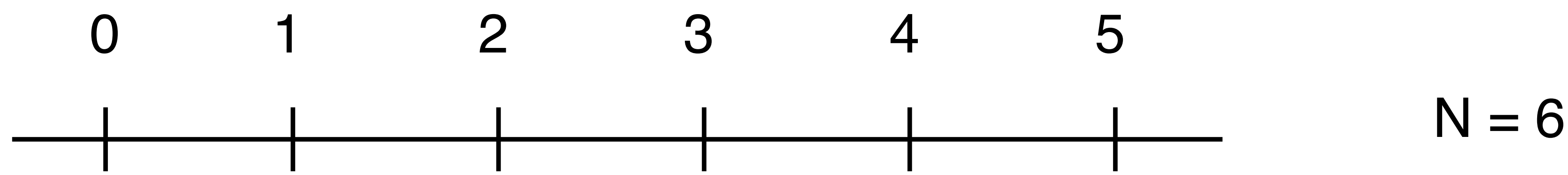
- Lost in C? -> **Google** your issue!

  - http://stackoverflow.com

  - https://openclassrooms.com/courses/apprenez-a-programmer-en-c (french only!)

FDs and periodic domain

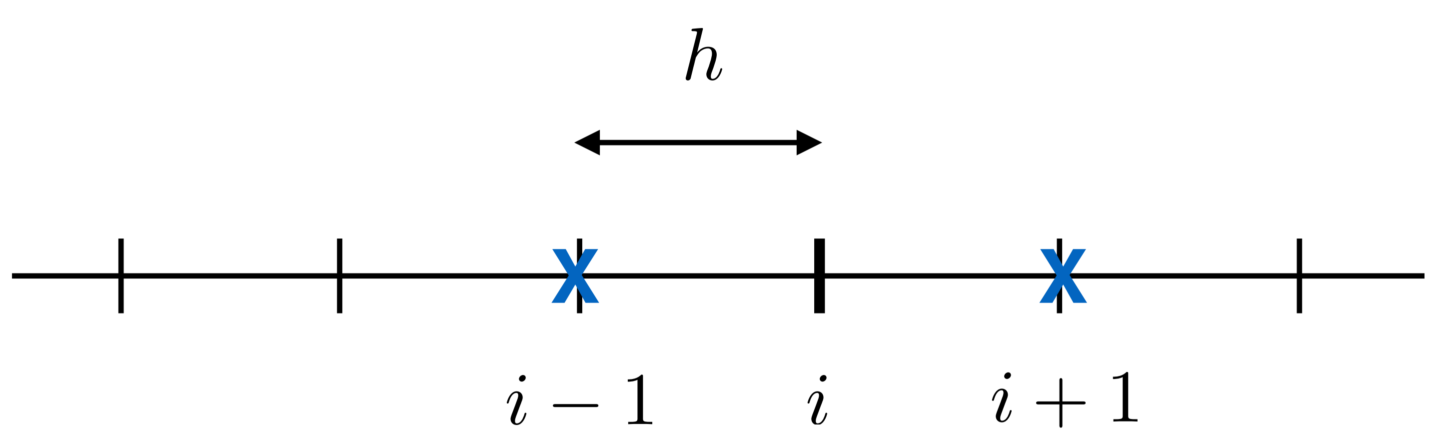
- Spatial discretization: N points numbered from 0 to N-1



- Computation of the first order derivative using a E2 scheme

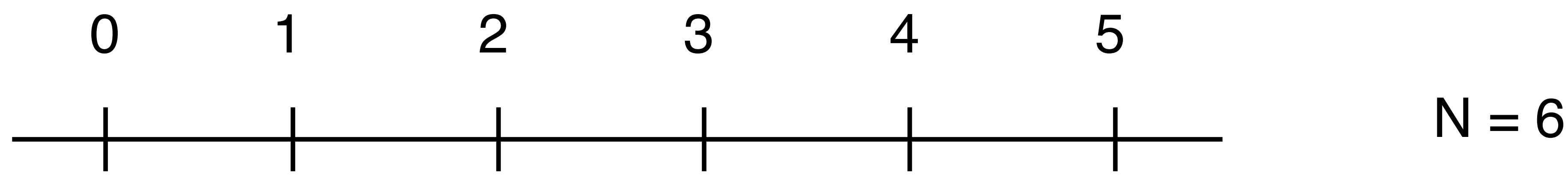
$$\left.\frac{\partial u}{\partial x}\right|_i = \frac{U_{i+1} - U_{i-1}}{2h} + \mathcal{O}(h^2)$$

★ Point in the middle of the domain



FDs and periodic domain

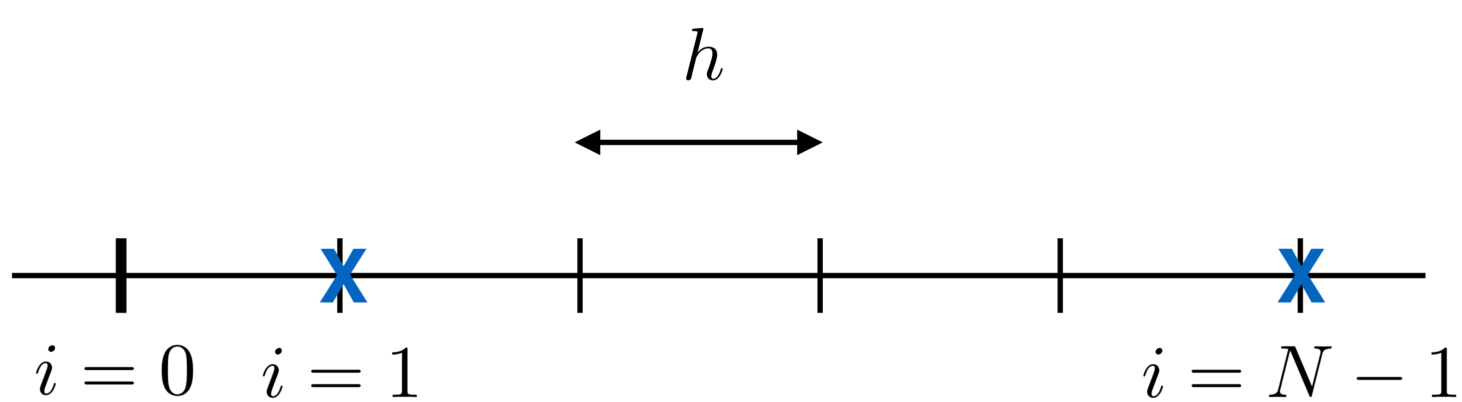
- Spatial discretization: N points numbered from 0 to N-1



- Computation of the first order derivative using a E2 scheme

$$\left.\frac{\partial u}{\partial x}\right|_i = \frac{U_{i+1} - U_{i-1}}{2h} + \mathcal{O}(h^2)$$

  ★ Point on the edge of the domain (periodic condition)



C language: dos and don’ts

- Periodic domain: modulo is your best friend!

```
0  % 10 = 0
1  % 10 = 1
10 % 10 = 10
```

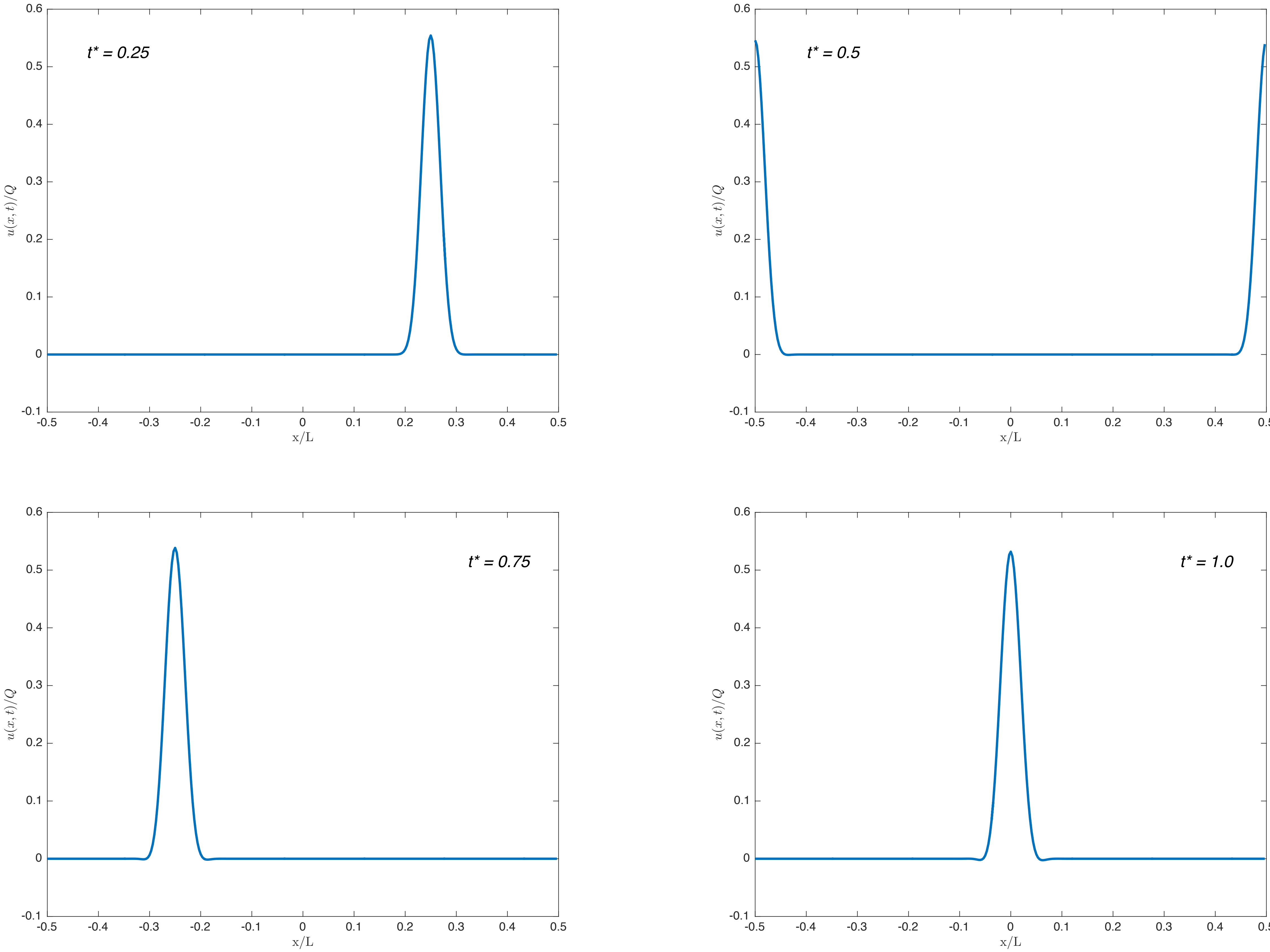
- never do a *if* condition inside a *for* loop

```
for (i=0; i<n; ++i){
   if (i==0) ….
   else ….
}
```

- use **calloc** instead of **malloc**

```
double* x = (double*) calloc(n,sizeof(double))
```

Pure convection case: example



Practical information

- Hand over: **16th March at 6 pm**
- Paper version (report) + Moodle (report + code)
- No fancy covers
- French or English
- Questions?

  Wednesday and Friday 4pm to 6pm (starting next week)

  Stevin a.078

  philippe.billuart@uclouvain.be

  p.parmentier@uclouvain.be (!!)