CERFACS

EUROPEAN CENTRE FOR RESEARCH AND ADVANCED TRAINING IN SCIENTIFIC COMPUTING

# Numerics in AVBP

N. Odier
odier@cerfacs.fr

# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- *Shock sensors*

# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- *Shock sensors*

# Mathematical background

- Conservation law (**conservative** formulation)

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x} f(\mathbf{u}) = 0 \qquad \text{with} \qquad \mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}$$

- **Non-conservative** formulation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{a}(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x} = 0 \qquad \text{with} \qquad \mathbf{a}(\mathbf{u}) = \frac{\partial f(\mathbf{u})}{\partial \mathbf{u}} \ \text{Jacobian}$$

- The Cauchy problem (**initial value** problem)

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \dfrac{\partial}{\partial x} f(\mathbf{u}) = 0 \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0 \end{cases}$$

- A solution $\mathbf{u}(x,t), \forall (x,t)$ **satisfying the Cauchy** problem is called a "**Strong**" **solution** (or "**Classical**").

- Such solution **may not exit** (if discontinuity in the initial solution).

➔ For this reason, "**Weak**" **solutions** are of interest.

# Weak solutions for conservation equations

*Will be of interest for finite-elements …*

**Let's consider a function $\phi(\mathbf{x},t)$ defined on a compact support.**

- Let's multiply the conservation equation :

$$\left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x} f(\mathbf{u})\right) \cdot \phi(\mathbf{x},t) = 0$$

- Integration over space-time: *(Scalar product)*

$$\int_{\mathbb{R}} \int_0^\infty \left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x} f(\mathbf{u})\right) \cdot \phi(\mathbf{x},t)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t = 0$$

- Provided $\phi(\mathbf{x},t)$ nullifies on the support boundaries, after **integration by parts**:

$$\int_{\mathbb{R}} \int_0^\infty \left(\mathbf{u}\frac{\partial \phi}{\partial t} + f(\mathbf{u})\frac{\partial \phi}{\partial x}\right)\ \mathrm{d}x\ \mathrm{d}t + \int_{\mathbb{R}} \mathbf{u}_0 \phi(\mathbf{x},0)\ \mathrm{d}x \ = \ 0$$

**<u>Variational formulation for the conservation equation</u>**

➔ No longer derivatives of $\mathbf{u}(\mathbf{x},t)$ !

➔ Derivatives of $\phi(\mathbf{x},t)$ instead (mass-matrix in the FE formalism…)

# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- *Shock sensors*

# Finite Volumes and Finite Elements

**Finite-volume** is based on the **strong formulation** of the conservation equation

- **Conservation** of transported quantities in a **control volume**
- The unknown is approximated by the **mean** over the control volume

**Finite elements** rely on the concept of **variational formulation**, and **weak solutions**

- **Shape functions** are defined within each cell.
- Unknowns are determined at the **node** (*instead average value in cell for FV*)

**Both approaches** rely on :

- Space and time **discretization of the conservation equation**
- Integration over a control volume, **fluxes balance** over the control volume

$$\frac{\partial U}{\partial t} = -\frac{1}{V_\Omega} \int_\Omega \underbrace{\vec{\nabla} \cdot \vec{F}}_{R_\Omega} \mathrm{dV}$$

***Residual***

➔ **Requires to the computation of a residual**

Let's focus on space and time discretization…

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x} f(\mathbf{u}) = 0$$

- The continuous conservation equation is discretized **on a mesh**

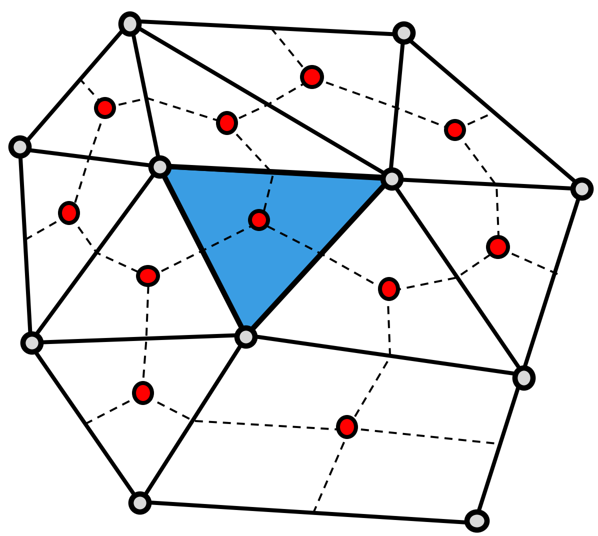
  Allows to store variables at each iteration

  Allows to define control volumes

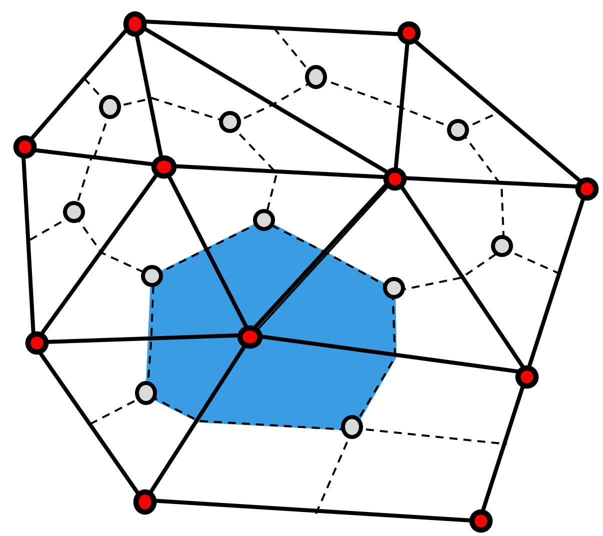
- The continuous conservation equation is discretized **in time**

# Spatial discretization

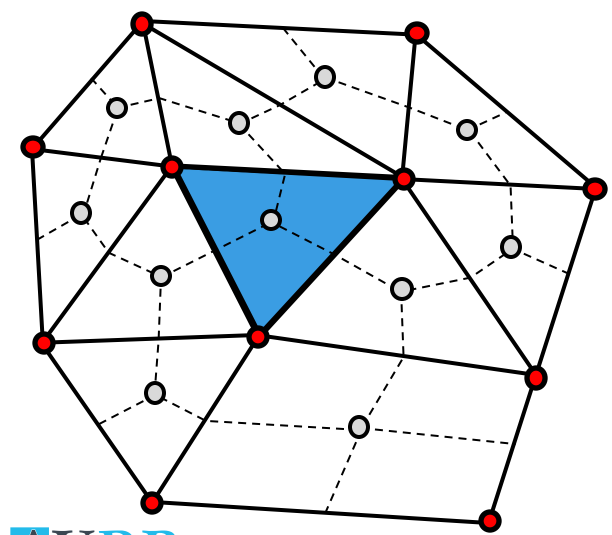
**Cell-centred**



**Vertex-centred**



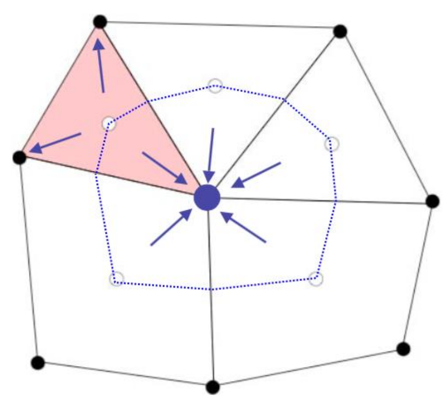
**Cell-vertex**



AVBP

**Variables are stored at the nodes**
**Control volumes are primal cells**

- Primary mesh
- (Median) dual mesh
- Variable storage
- Control volume



# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- *Shock sensors*

Presentation of 2 main Numerical schemes of AVBP

- **Lax-Wendroff** (LW)
- **Two-Step Taylor Galerkin** (TTG) schemes

**In summary, two families of schemes are available in AVBP:**

- **LW** *Finite Volume*
  - − second order accurate in space and time
  - + not expensive
- **TTG4A and TTGC** *Finite Element*
  - + third order accurate in space and time
    dissipation and dispersion properties meeting LES requirements
  - − ~ 2.5 times more expensive than LW

# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

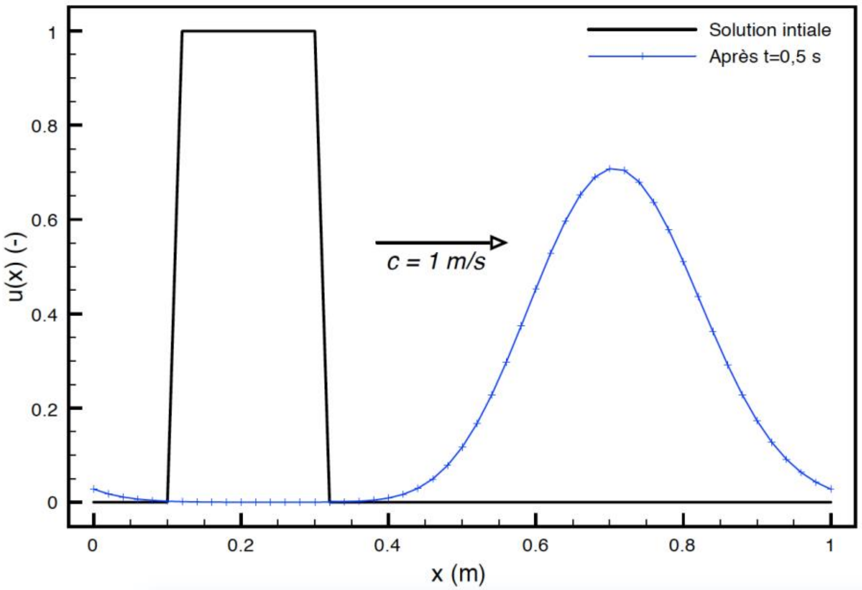
## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- *Shock sensors*

# Properties of AVBP schemes

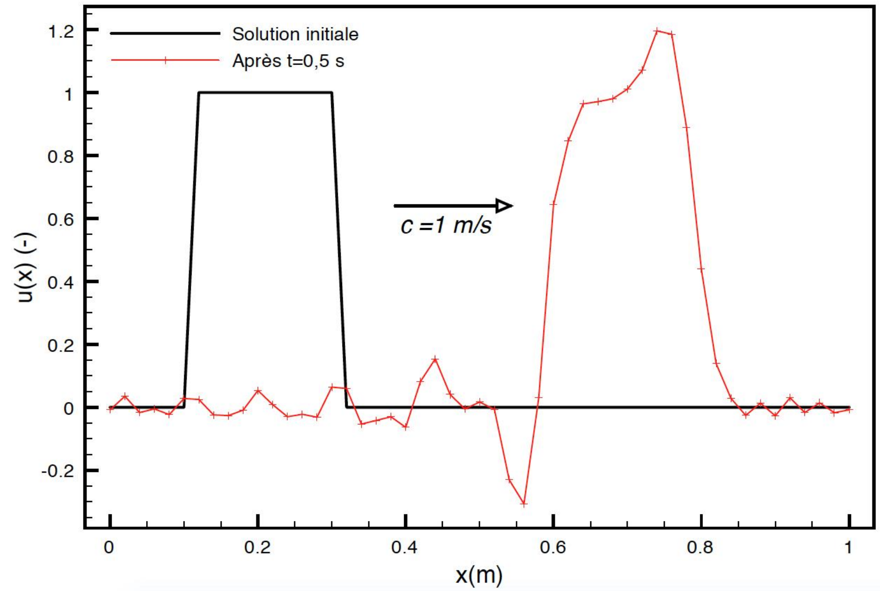
## Mesh discretization introduces errors:

### **Dissipation** error




- Decrease of the wave amplitude
- Smoothing of its gradients

### **Dispersion** error

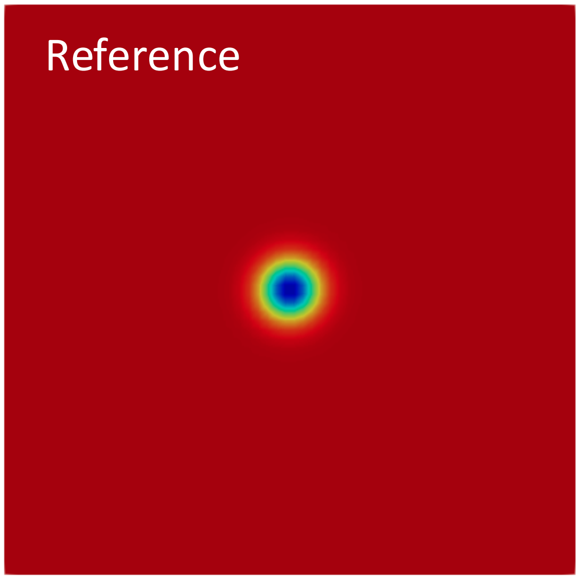



**Dispersive medium:**

A medium in which wave velocities depend on the wave frequency

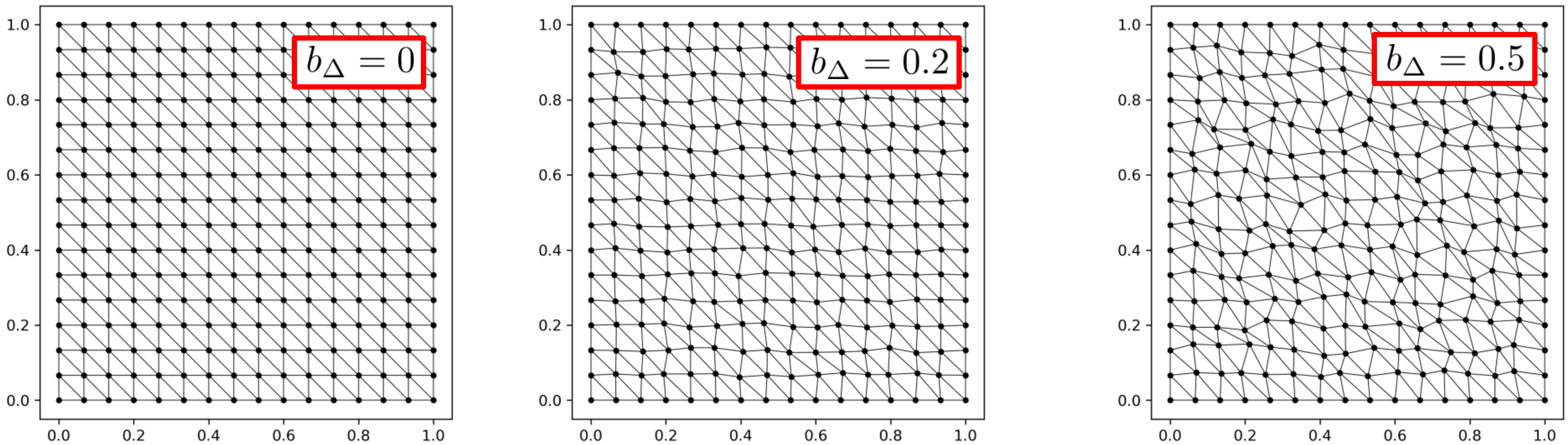
➔ Harmonics of the initial solution are not transported at the same speed

# Application

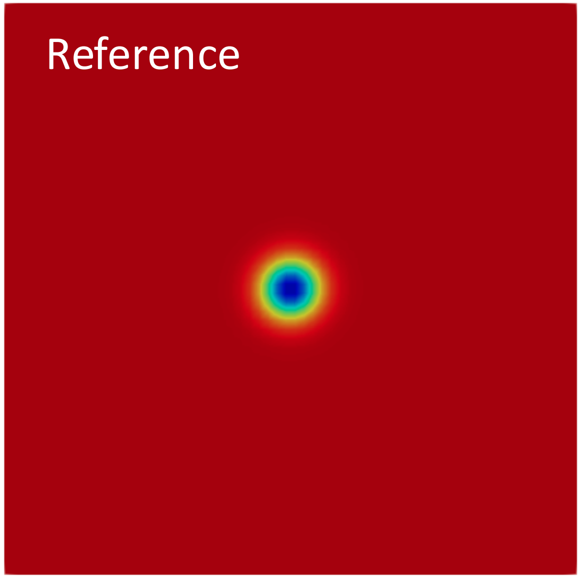



Let’s consider the advection of a 2D isentropic vortex…

Let’s try several schemes, and several mesh qualities



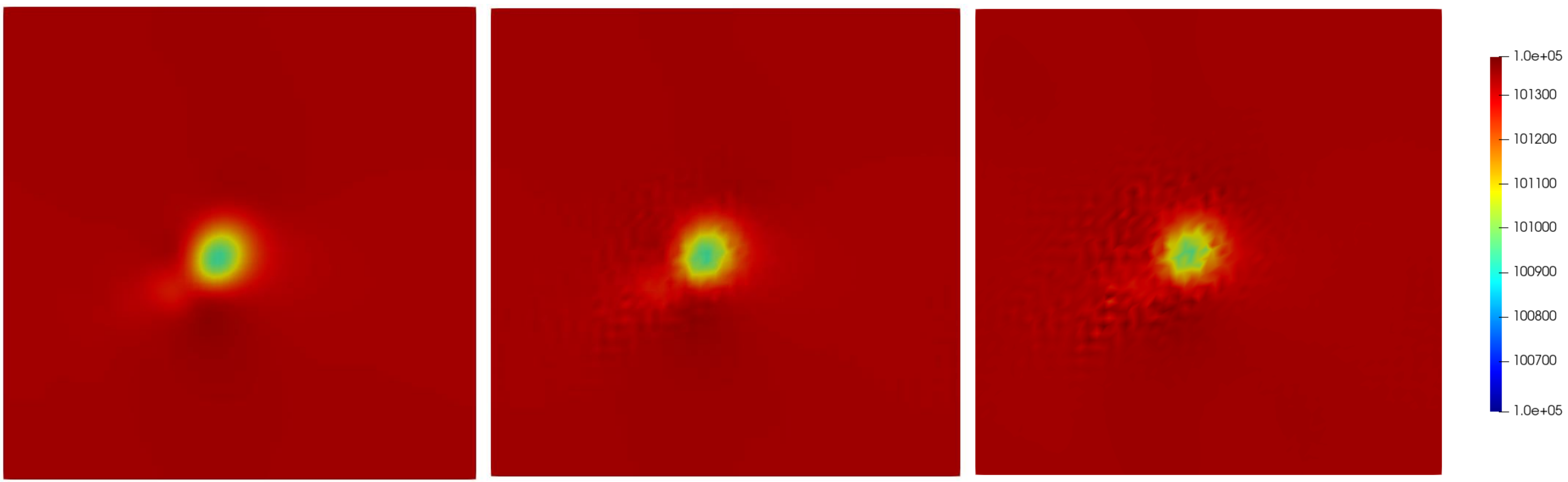
# Lax-Wendroff



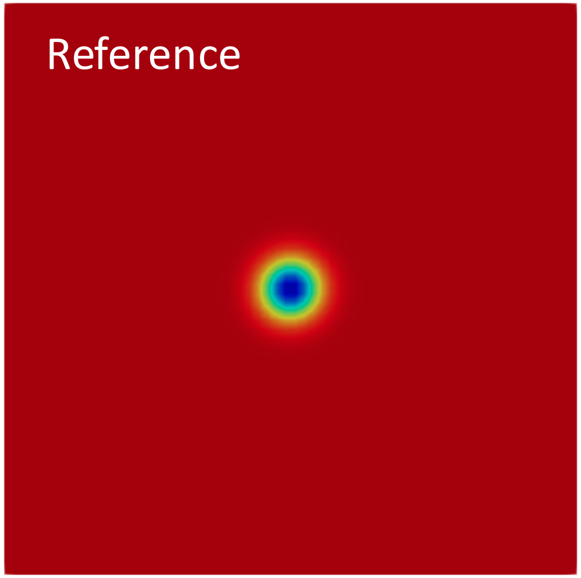

$\Rightarrow$ Dissipative

$\Rightarrow$ Some dispersion

$b_\Delta = 0$ $\quad$ $b_\Delta = 0.2$ $\quad$ $b_\Delta = 0.5$

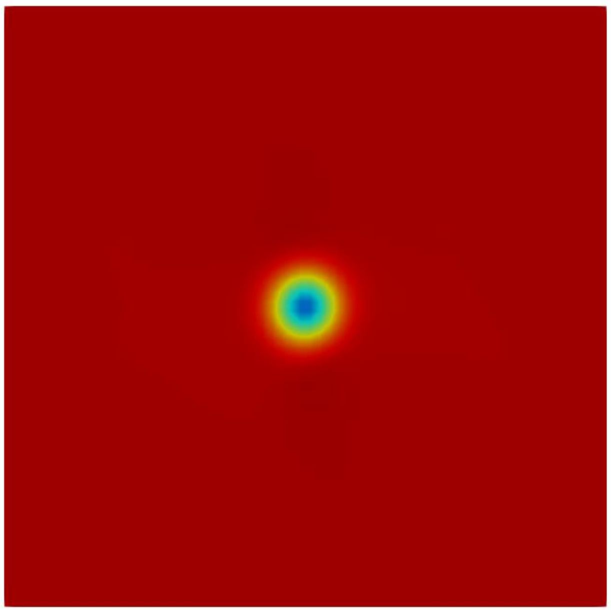


# TTG4A

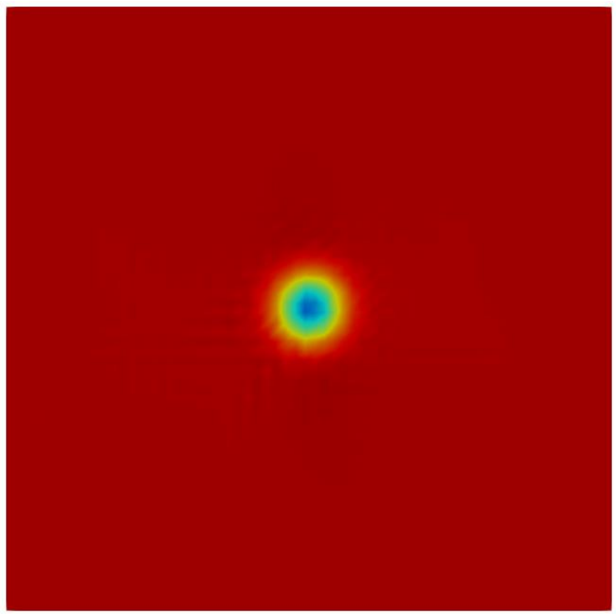



$\Rightarrow$ Still dissipation
$\Rightarrow$ Smaller dispersion

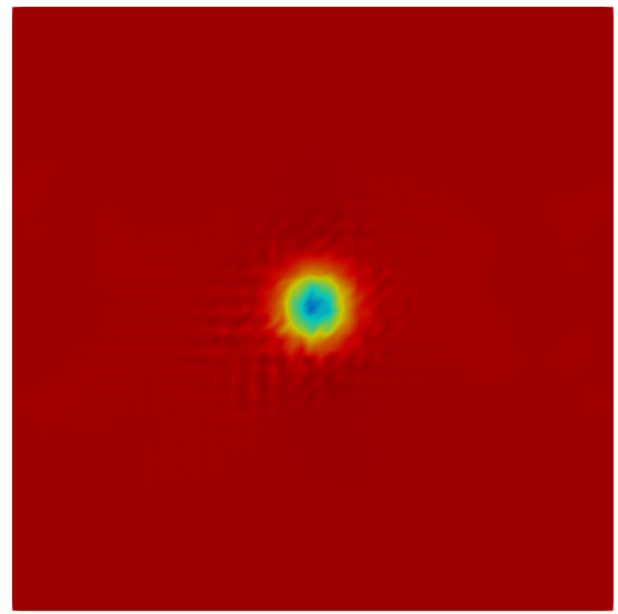
$\boxed{b_\Delta = 0}$

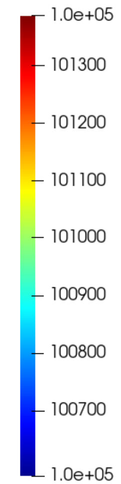


$\boxed{b_\Delta = 0.2}$



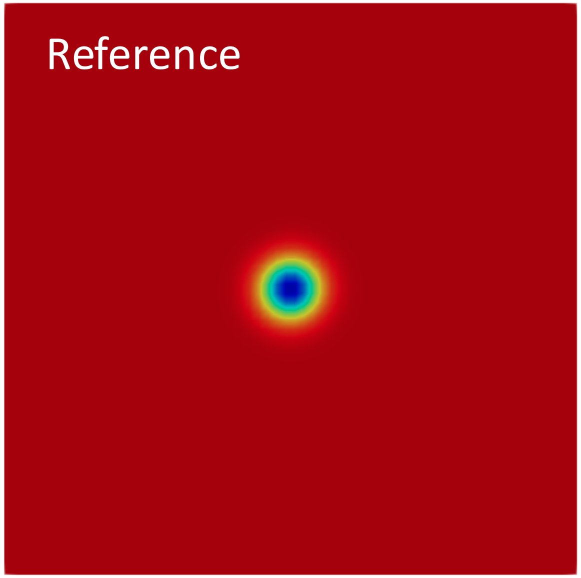
$\boxed{b_\Delta = 0.5}$






CERFACS

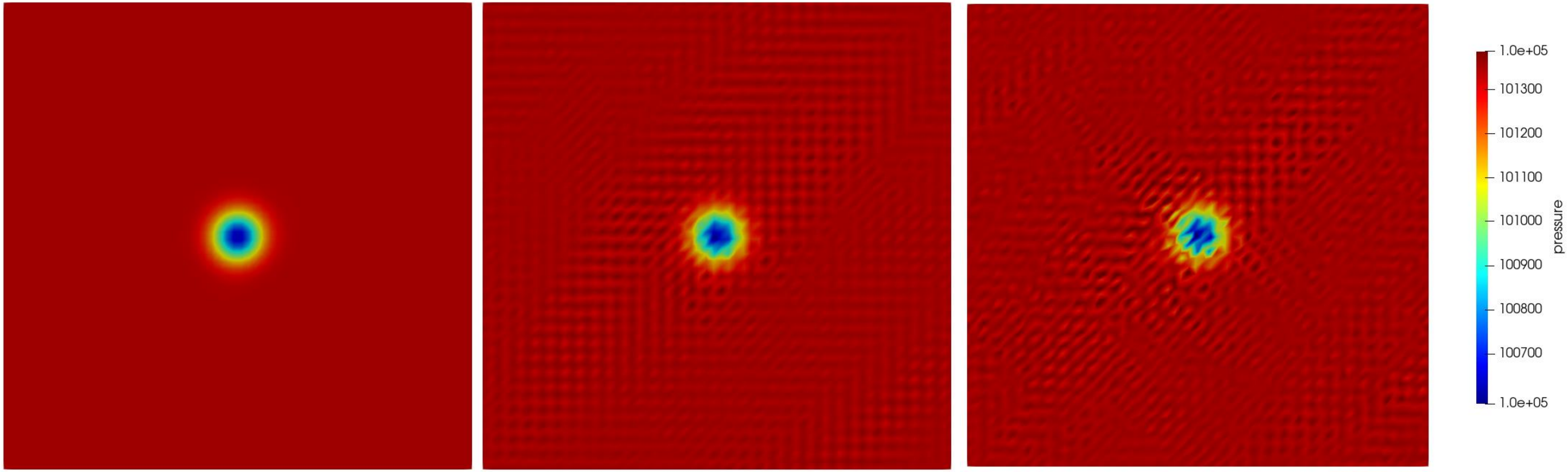
# TTGC




$\Rightarrow$ Very few dissipation

$\Rightarrow$ Large dispersion

$\boxed{b_\Delta = 0}$ $\boxed{b_\Delta = 0.2}$ $\boxed{b_\Delta = 0.5}$



# Partial conclusion

- **Lax Wendroff** is **dissipative**
- **TTG4A** is **also dissipative**, but induces **small dispersion**
- **TTGC very accurate on perfect grids**. **Dispersion** errors however occur for **distorted** grids
- **TTG scheme ~ 2.5** times more expensive

$\Rightarrow$ **No scheme is perfect !**

LW is of interest for LES initialization.

TTG schemes better suited for statistics.

<u>Errors decrease with increasing:</u>

- **Mesh quality**
- **Mesh resolution**
- **Order of numerical scheme**
- **Lower CFL**

**!**

# A simple tip…

*HIP* allows mesh adaptation thanks to the MMG3D library.

**"Mmg3d isofactor –f 1"**

Can significantly improve your initial Centaur / Ansys mesh !

# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- *Shock sensors*

# Wiggles

- AVBP schemes are **centred schemes**.

- In case of **discontinuity in the discrete approximation** of the hyperbolic equation (*e.g.* mesh **refinement interface,** mesh **boundary**):

➔ Creation of a **spurious wave**, with a wavelength $\lambda = 2\Delta x$ :
**"Wiggles", "q-waves"**

Wiggles move forward at a group velocity -c

*Vichnevetsky/Boyles, 1982*
*Vichnevetsky Comp. Math. Appls 1985*
*Vichnevetsky Int. J. Num. methods Fluids 1987*
*Sengupta App. Maths. Comp. 2012*

# Centred schemes and *q-waves*

It can be showed that: when **using centred schemes** *i.e.* $\left[\frac{\mathrm{d}u_n}{\mathrm{d}t} + \left(\frac{u_{n+1} - u_{n-1}}{2h}\right) = 0\right]$

to discretize a continuous **wave equation** $\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0$

leads to a **numerical** (discrete) solution consisting in **2 waves** :

*"p-waves"*

$$\begin{cases} p(\mathbf{x}, t) = \dfrac{u + v}{2} \\ q(\mathbf{x}, t) = \dfrac{u - v}{2} \end{cases}$$

*"q-waves"*

satisfying

$$\begin{cases} \dfrac{\partial p}{\partial t} + c\dfrac{\partial p}{\partial x} = 0 \\ \dfrac{\partial q}{\partial t} - c\dfrac{\partial q}{\partial x} = 0 \end{cases}$$

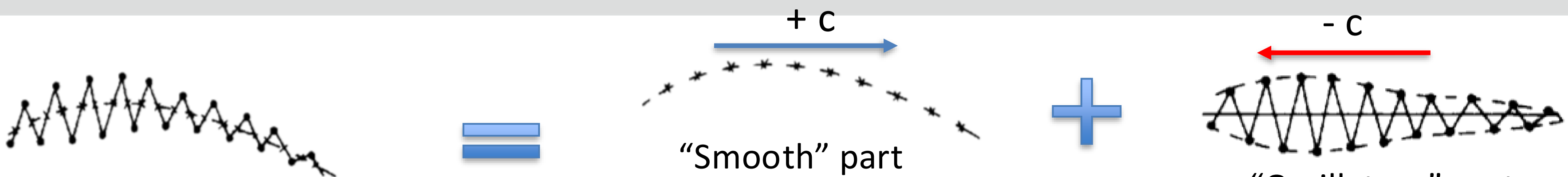
+c

Moving ***forward***

Moving ***backward***

-c

# Centred schemes and *q-waves*



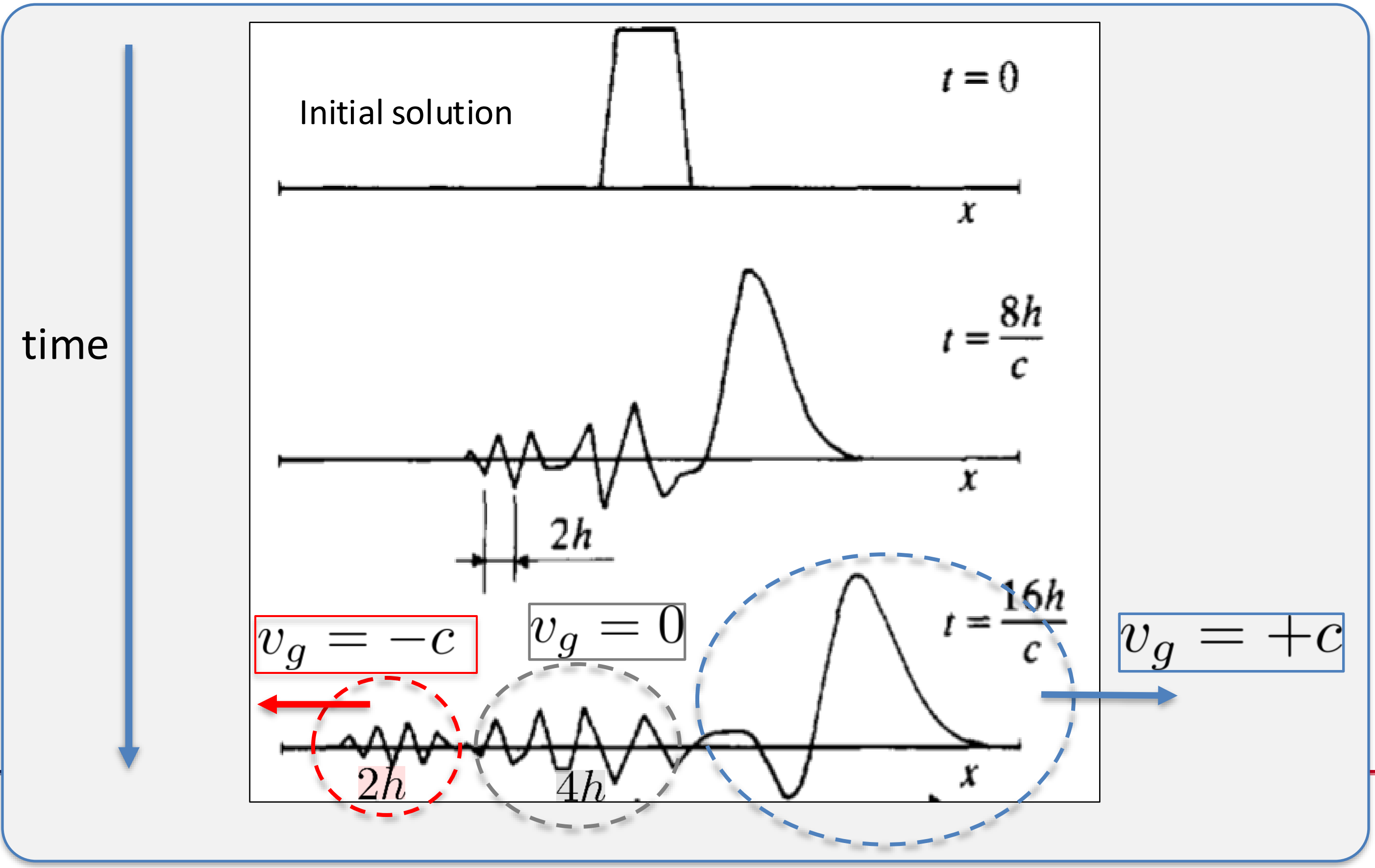

Discrete numerical solution

"Smooth" part

$$p(\mathbf{x},t) = \frac{u+v}{2}$$

"Oscillatory" part

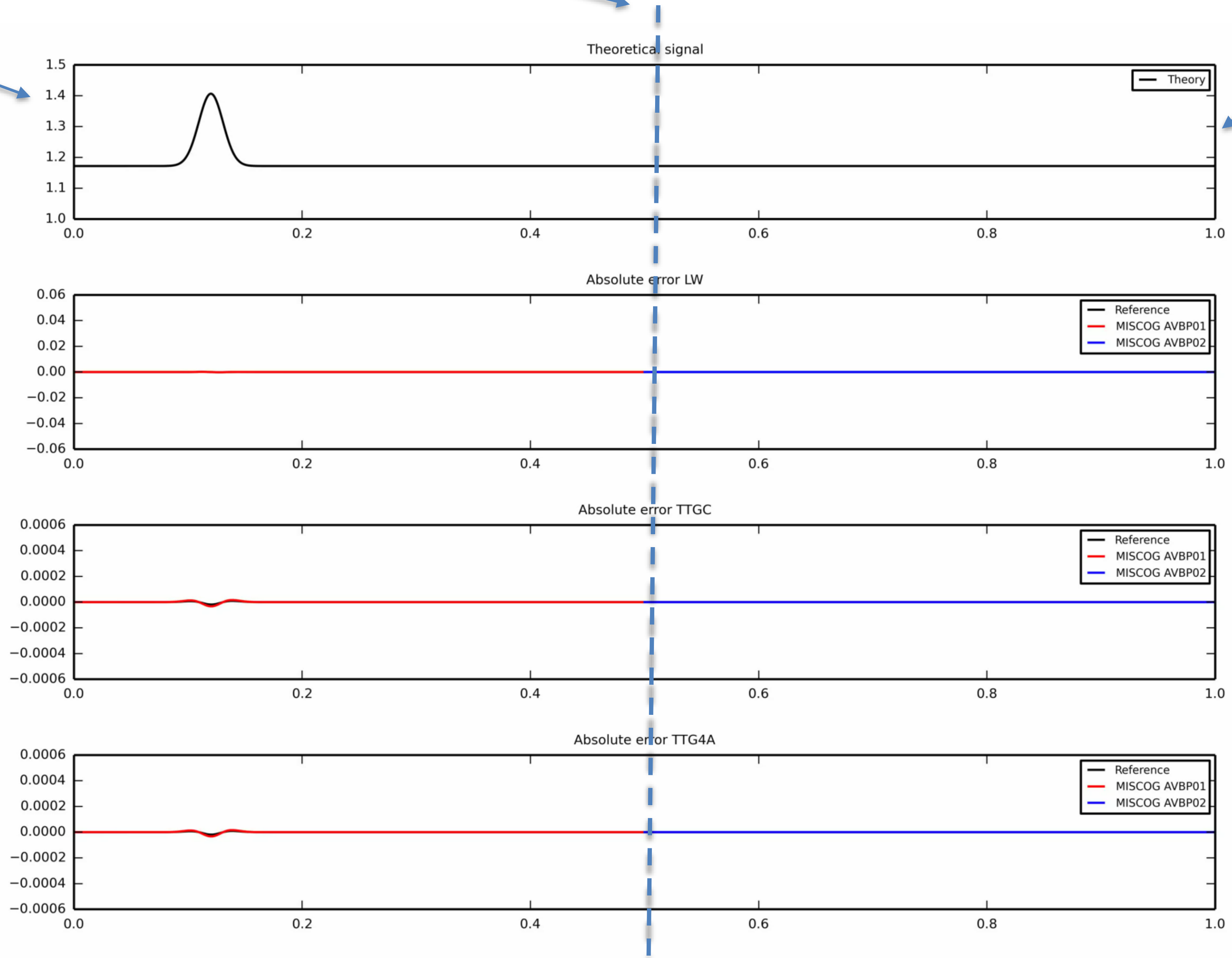
$$q(\mathbf{x},t) = \frac{u-v}{2} = u - \left(\frac{u+v}{2}\right)$$



# Wiggles: illustration

Mesh interface

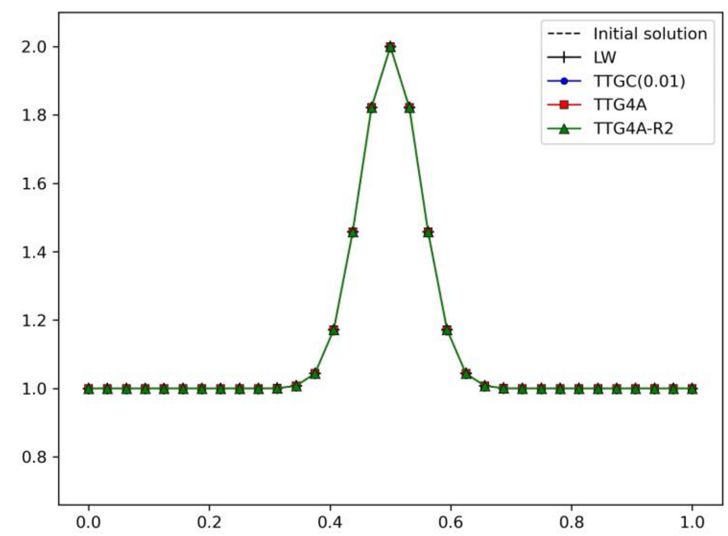
Boundary condition




Boundary condition

q-waves generate p-wave !

# Dispersion errors and *q-waves*

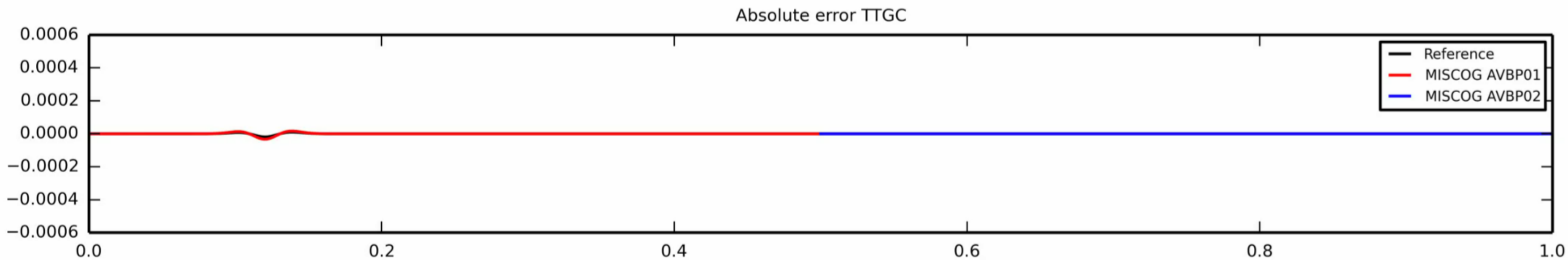



**Dispersion error**

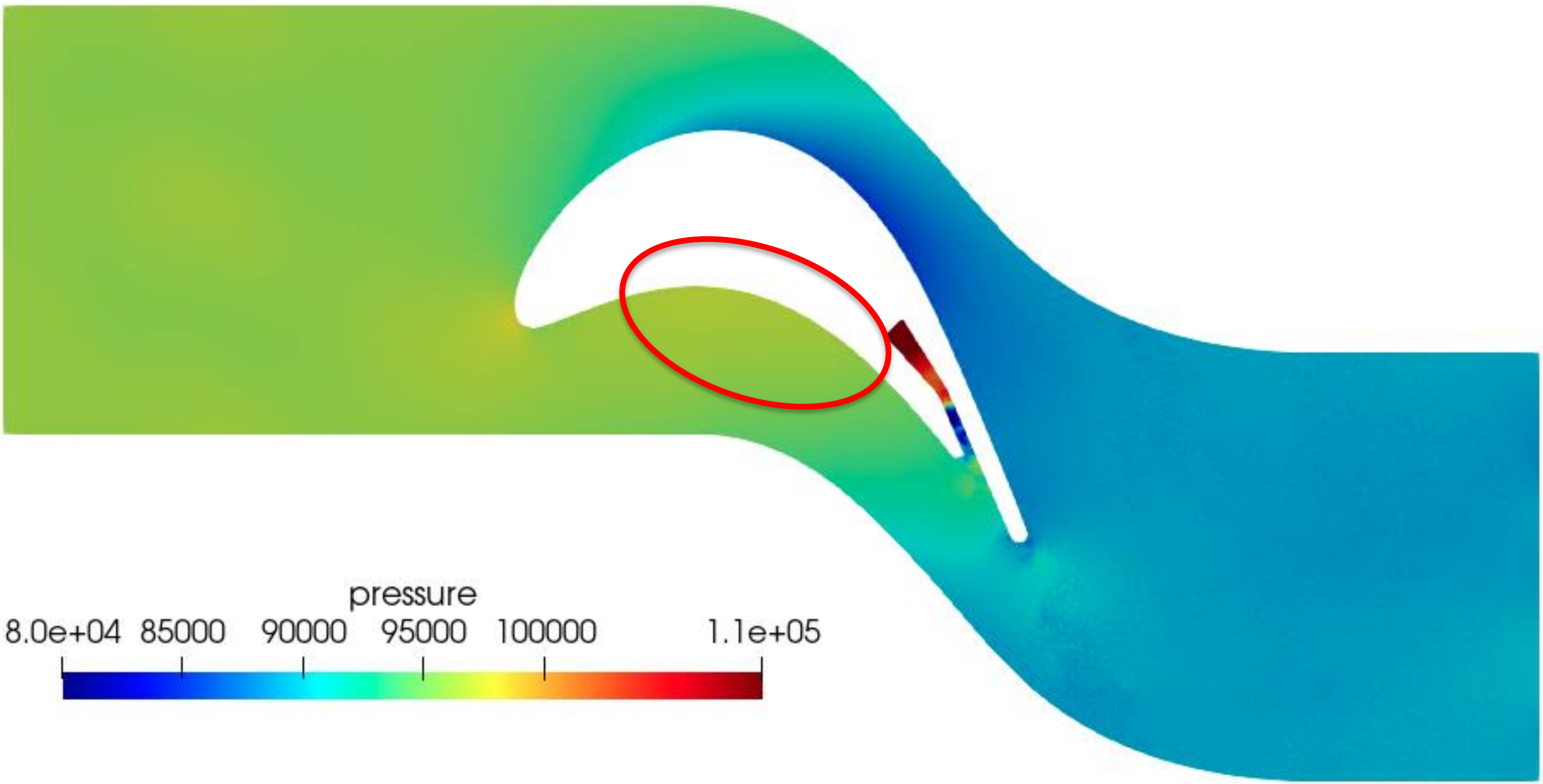
Mostly convected at the phase velocity u+c, along with the physical wave

**q-waves (wiggles)**

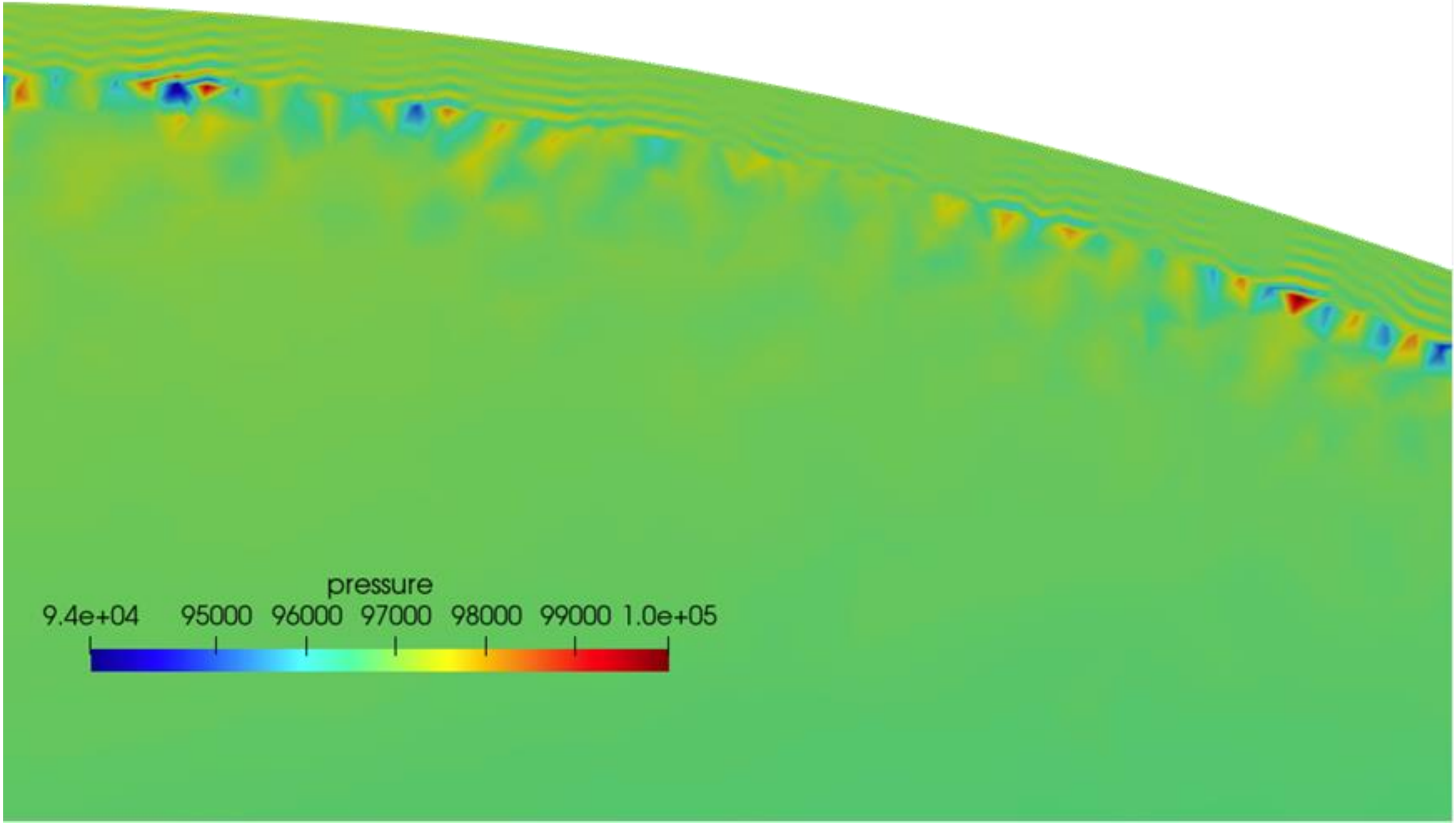
Convected with a group velocity –c, as physical wave interacts with a boundary
q-waves may generate p-waves (physical waves) when interacting back the other boundary



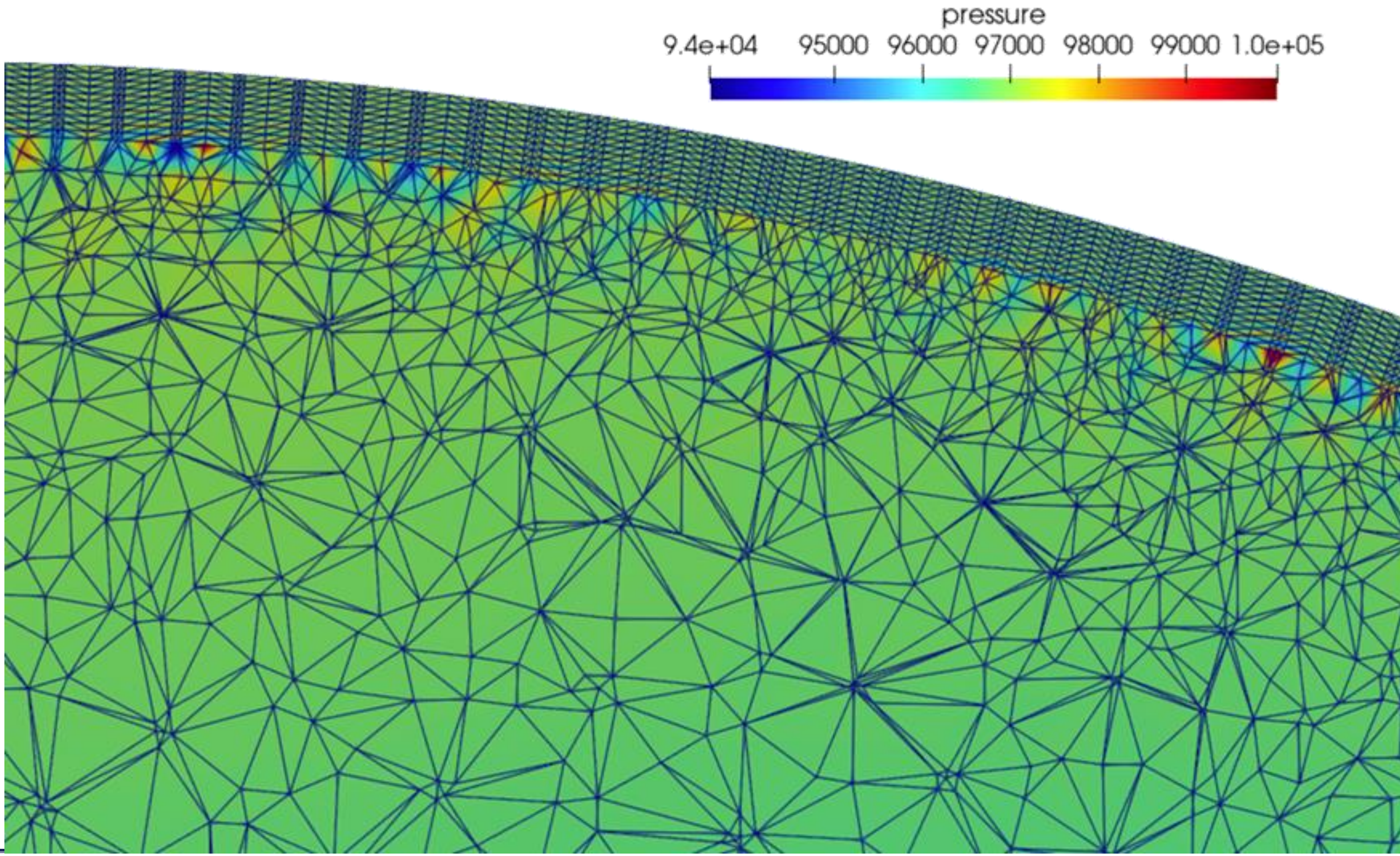
# Issue at mesh interface



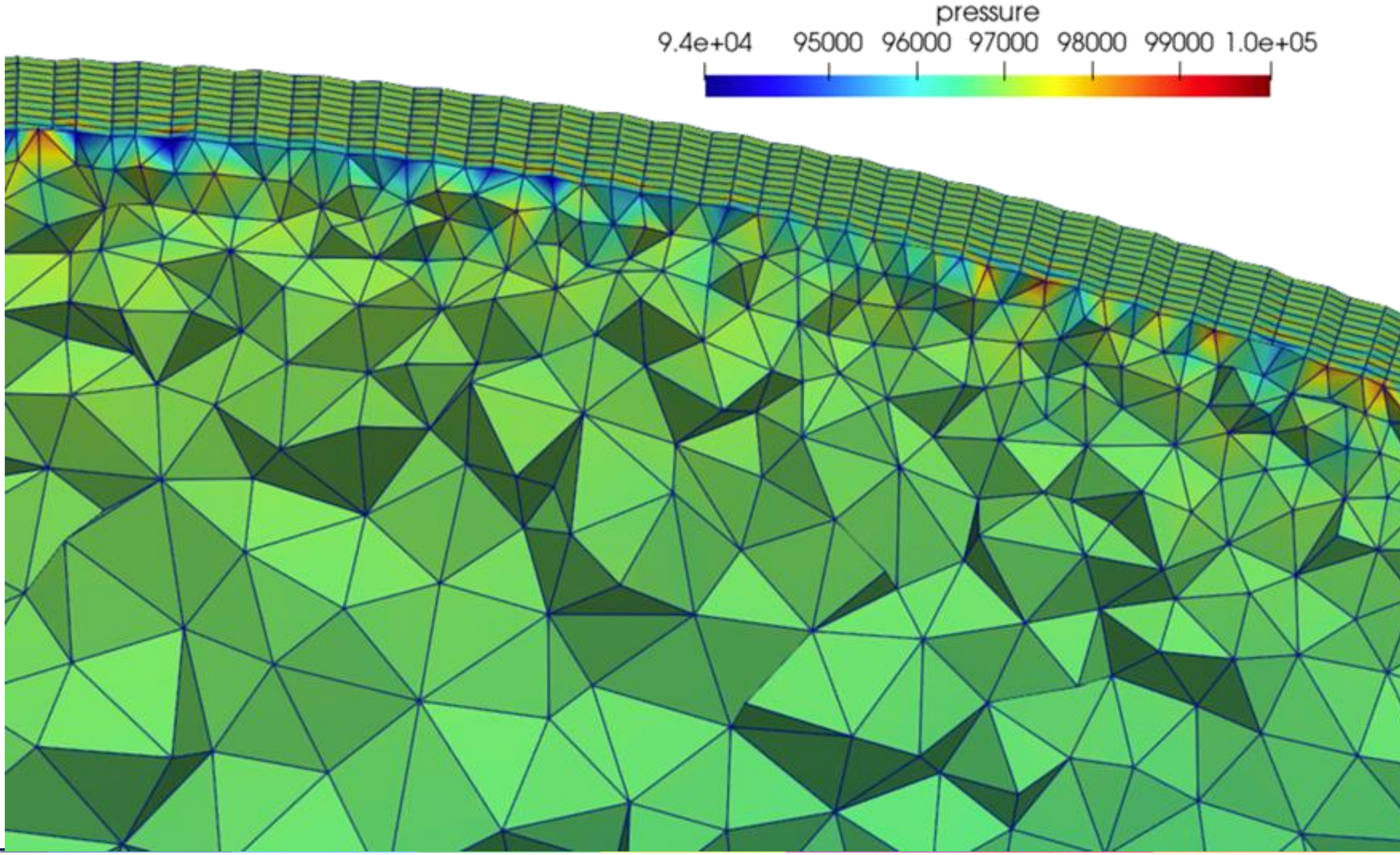
# Issue at mesh interface



# Issue at mesh interface

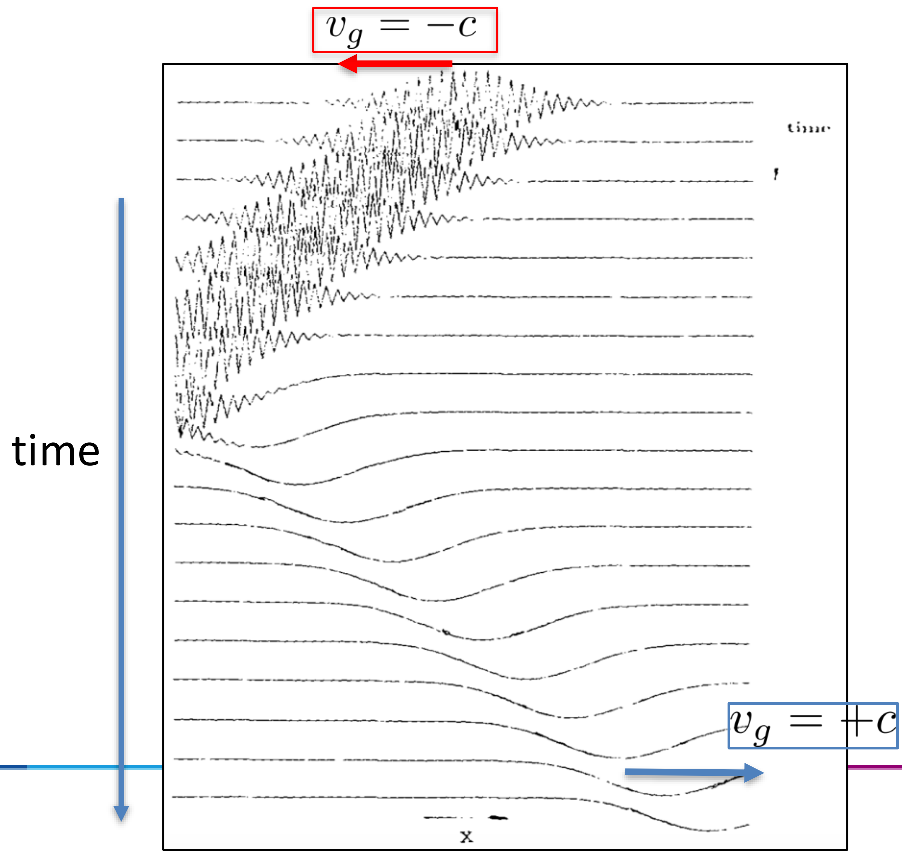


# Issue at mesh interface



# Partial conclusion on Wiggles

- Wiggles are not just dispersion error downstream of physical waves
- Wiggles are actual waves, propagating into the domain, at a group velocity -c
- Wiggles can generate physical waves when interacting with a BC !




*See*
*Vichnevesky Bowles, SIAM 1982*
*Poinsot Veynante Ch.9*
*Sengupta et al. App. Math. Comp. 2012*

# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

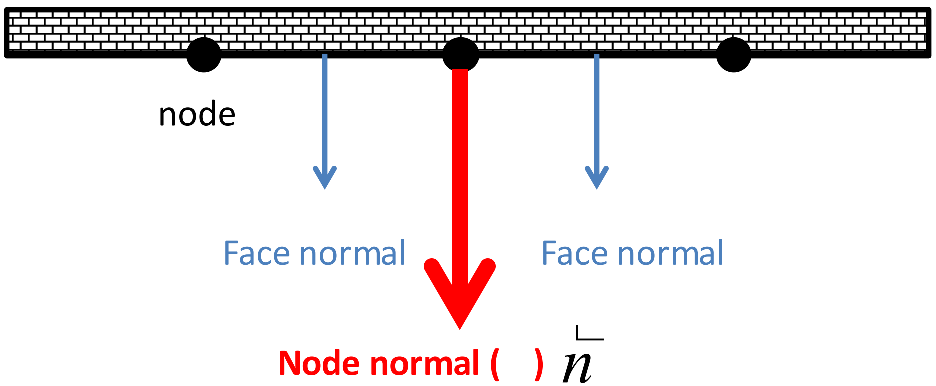
## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- *Shock sensors*

# WALL/ WALL interaction : critical issue on corner

To ensure **mass conservation**, the normal (to the wall) velocity is cancelled: $\boxed{\vec{U}.\vec{n} = 0}$

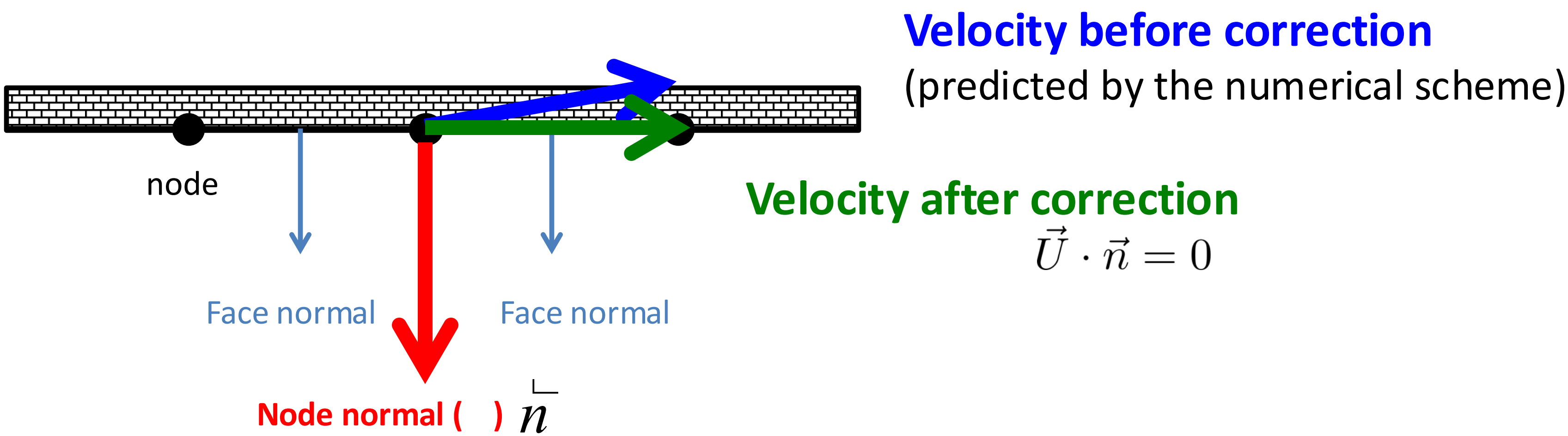
**Node normals are an average** of normals on neighboring surfaces:



# WALL/ WALL interaction : critical issue on corner

To ensure **mass conservation**, the normal (to the wall) velocity is cancelled: $\boxed{\vec{U}.\vec{n}=0}$

**Node normals are an average** of normals on neighboring surfaces:

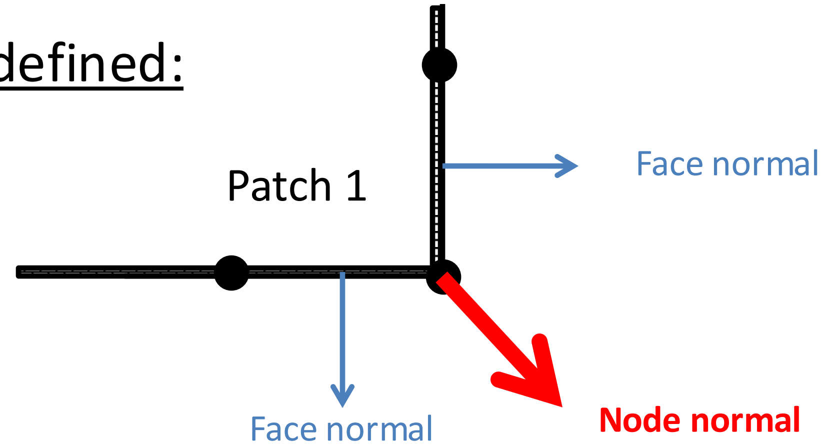


# WALL/ WALL interaction : critical issue on corner

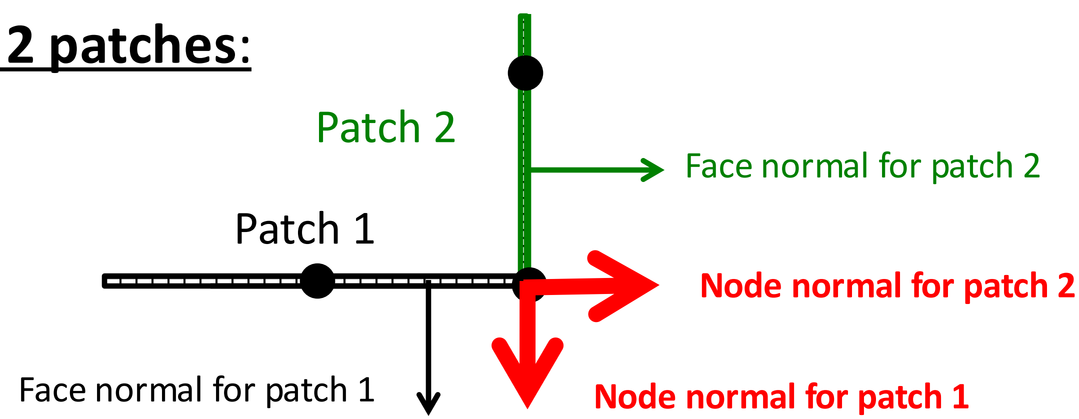
**Wall normal** direction at **corners** is an **ambiguous quantity**!

It depends on how patches are defined :

- If only **1 wall patch** is defined:




- If the wall **is split in 2 patches**:

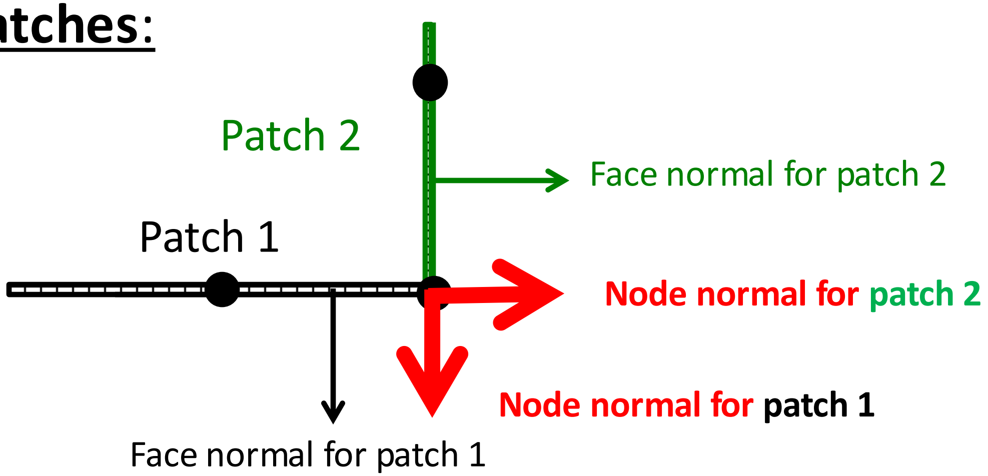


# WALL/ WALL interaction : critical issue on corner

**Wall normal** direction at **corners** is an **ambiguous quantity**!

It depends on how patches are defined :

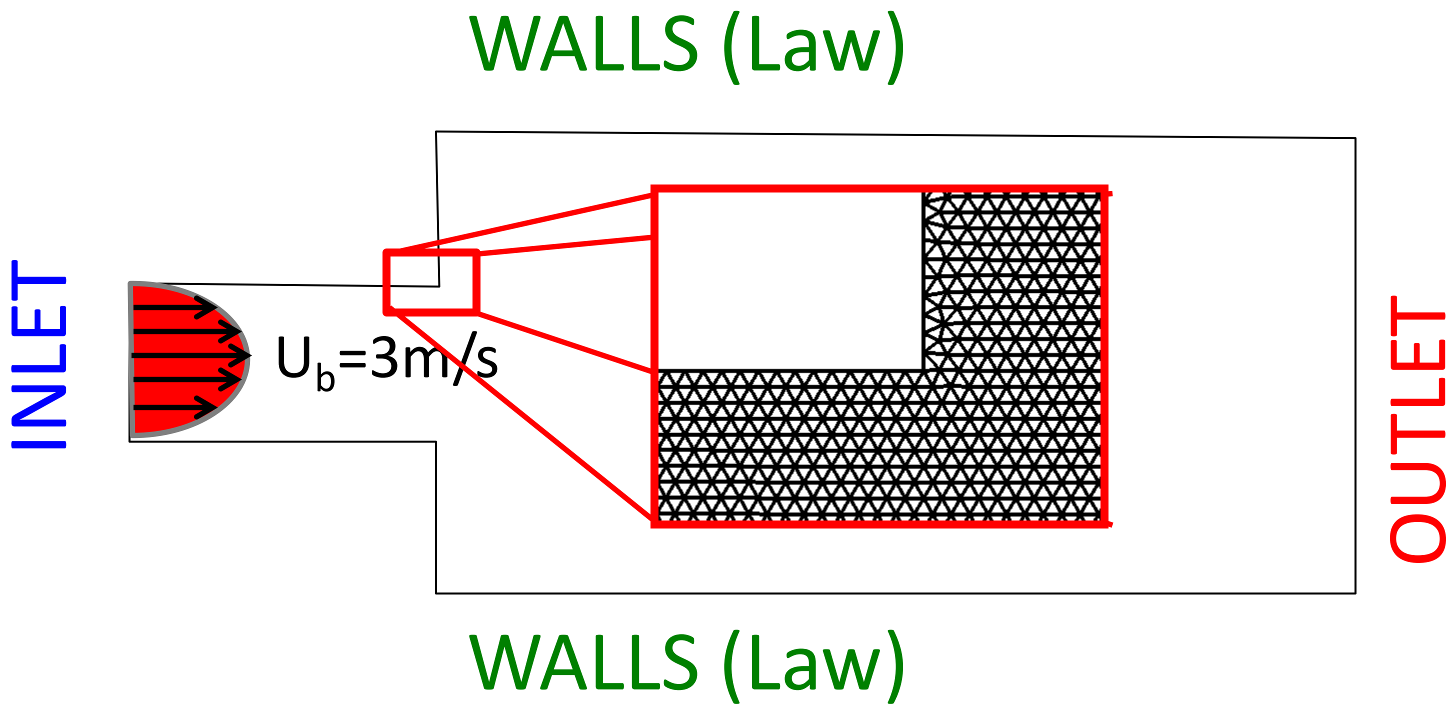
- If the wall is **split in 2 patches**:




→ **Two normals** are defined for the same node !

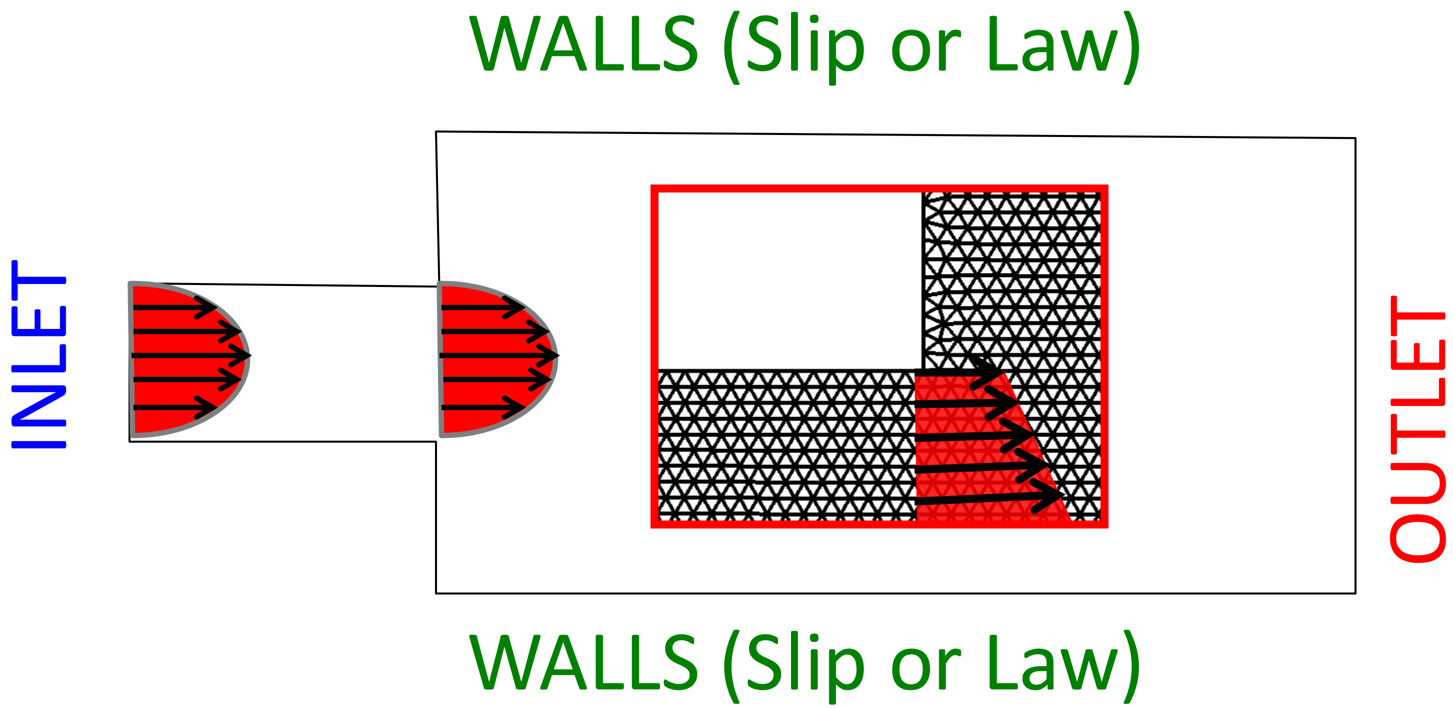
→ **Two treatments** are successively applied

# Example : sudden expansion




**WHAT HAPPENS AT THE CORNER ?**

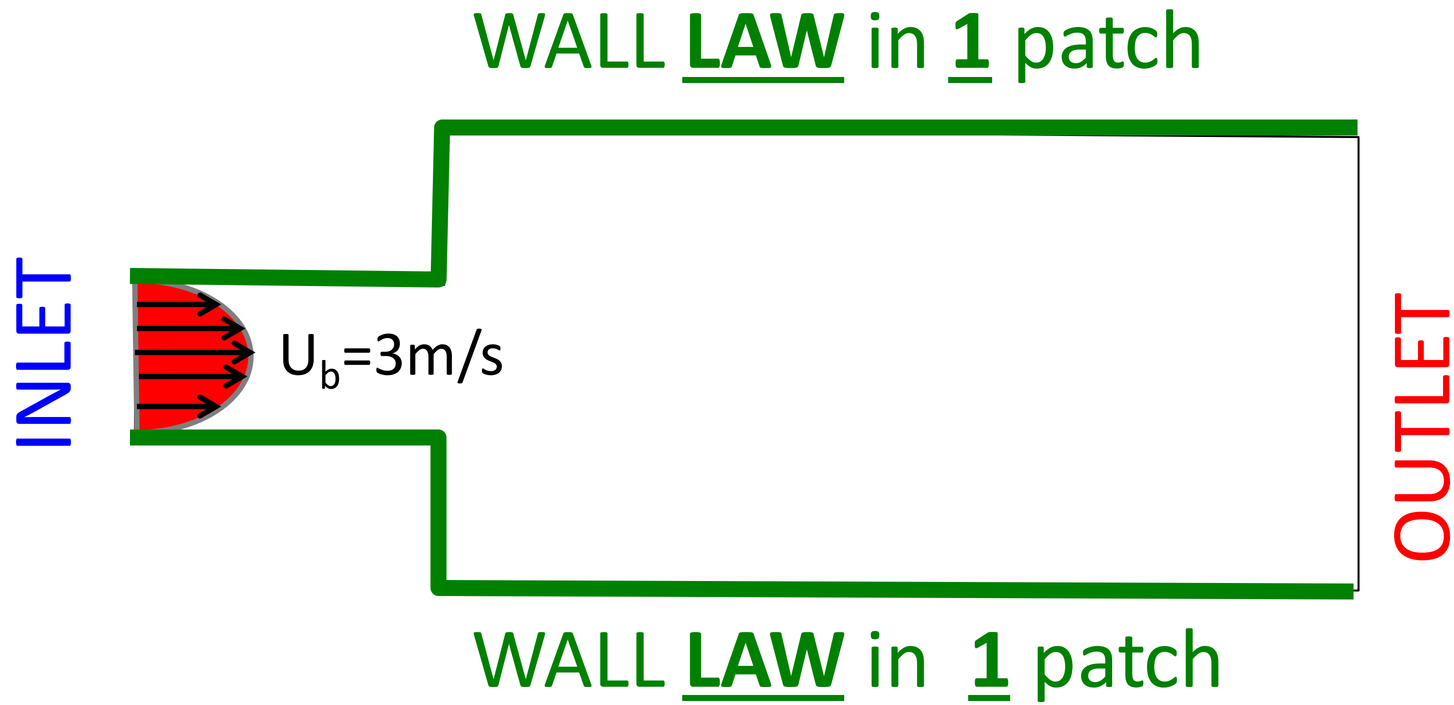
# Example : sudden expansion



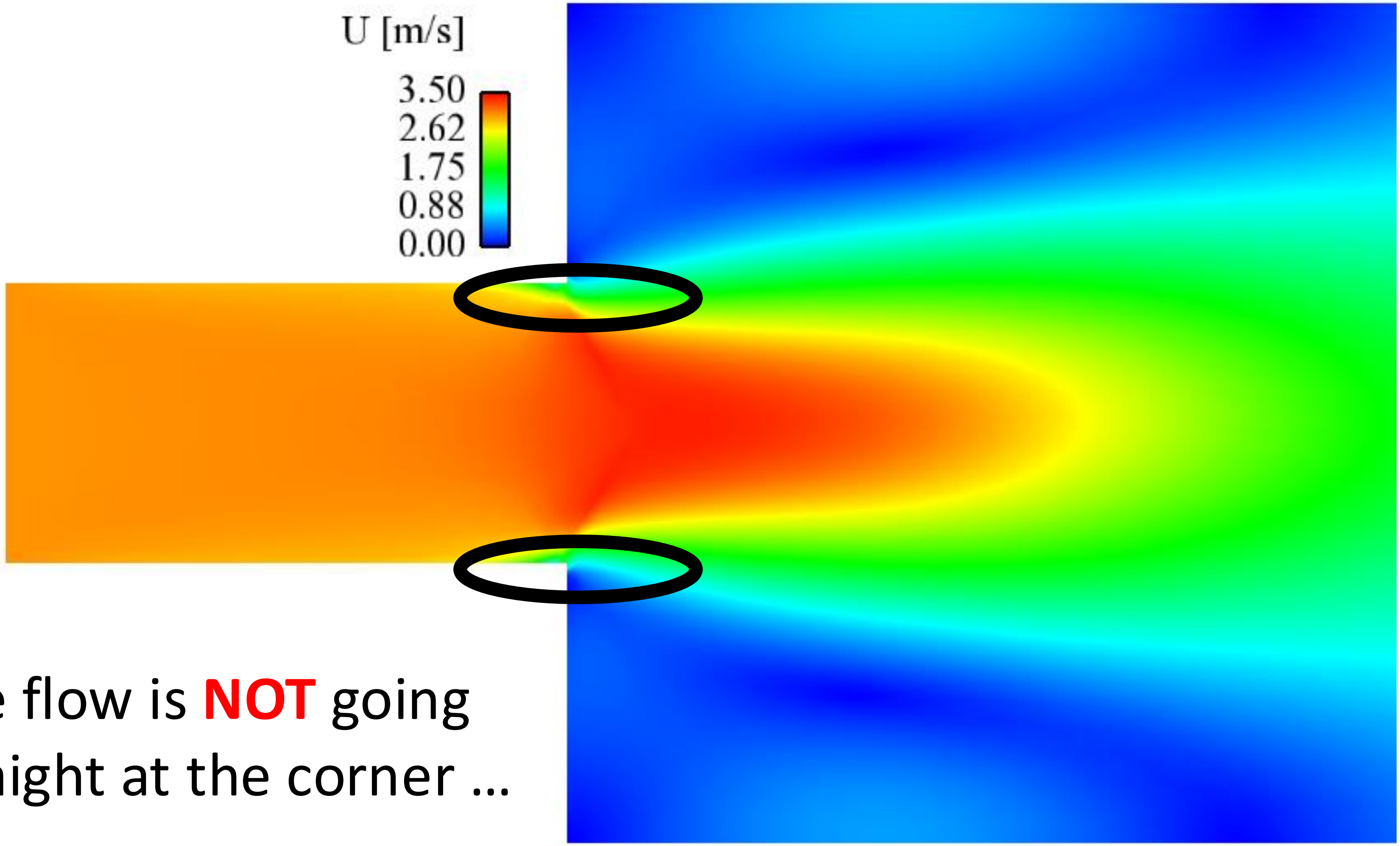

<u>Expected solution</u>: the **flow** should go **straight**

# **Joint** patches test case

➔ First, let's consider **wall Law** using **one <u>single</u>** patch



# **Joint** patches test case



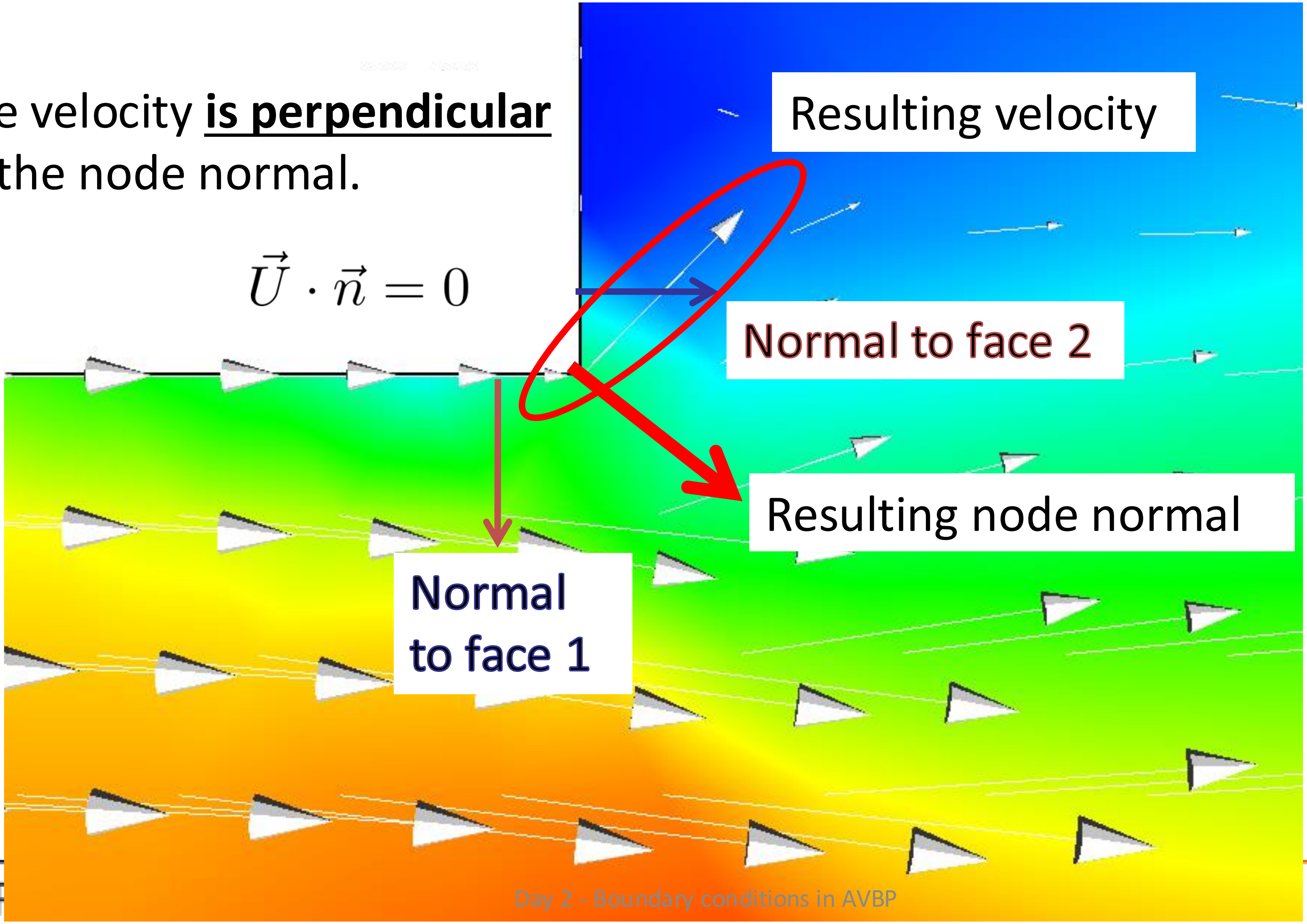

The flow is **NOT** going straight at the corner …


65

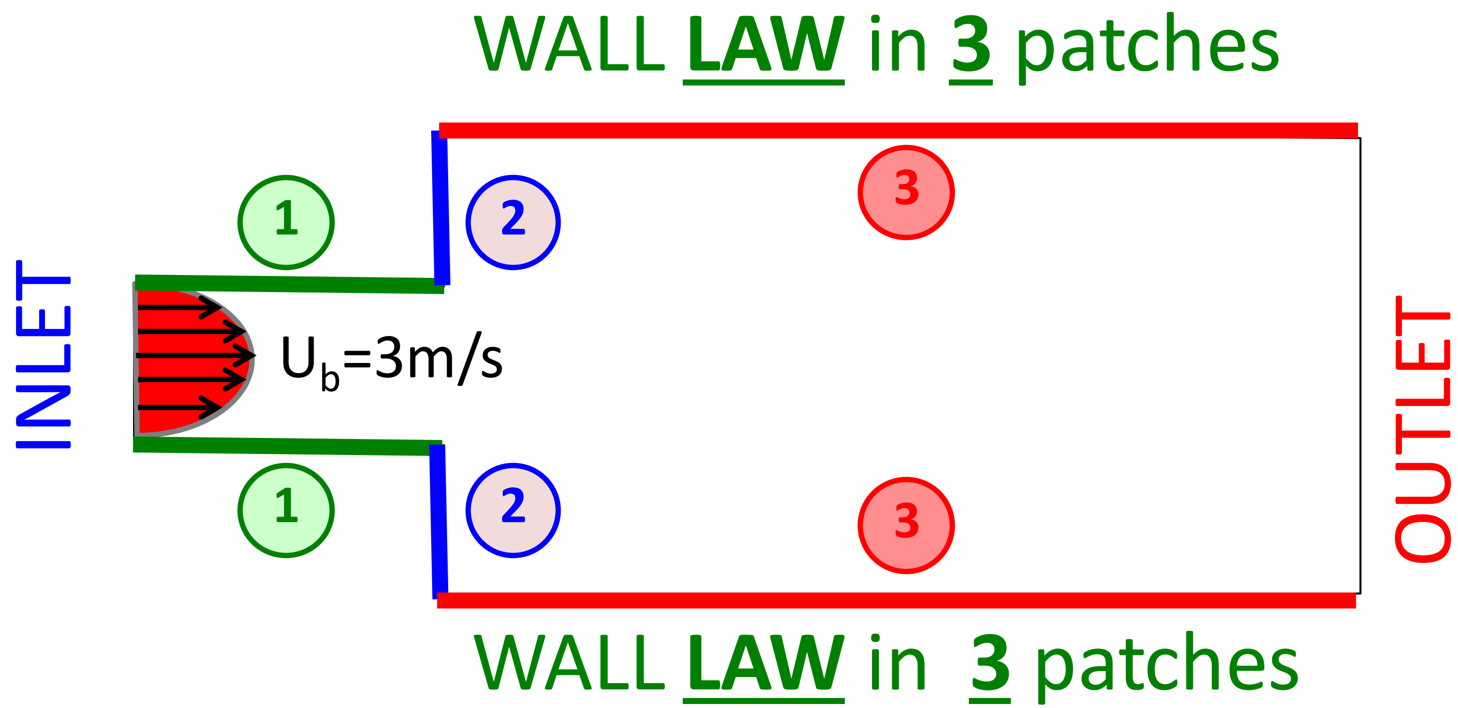
# **Joint** patches test case

The velocity **<u>is perpendicular</u>** to the node normal.

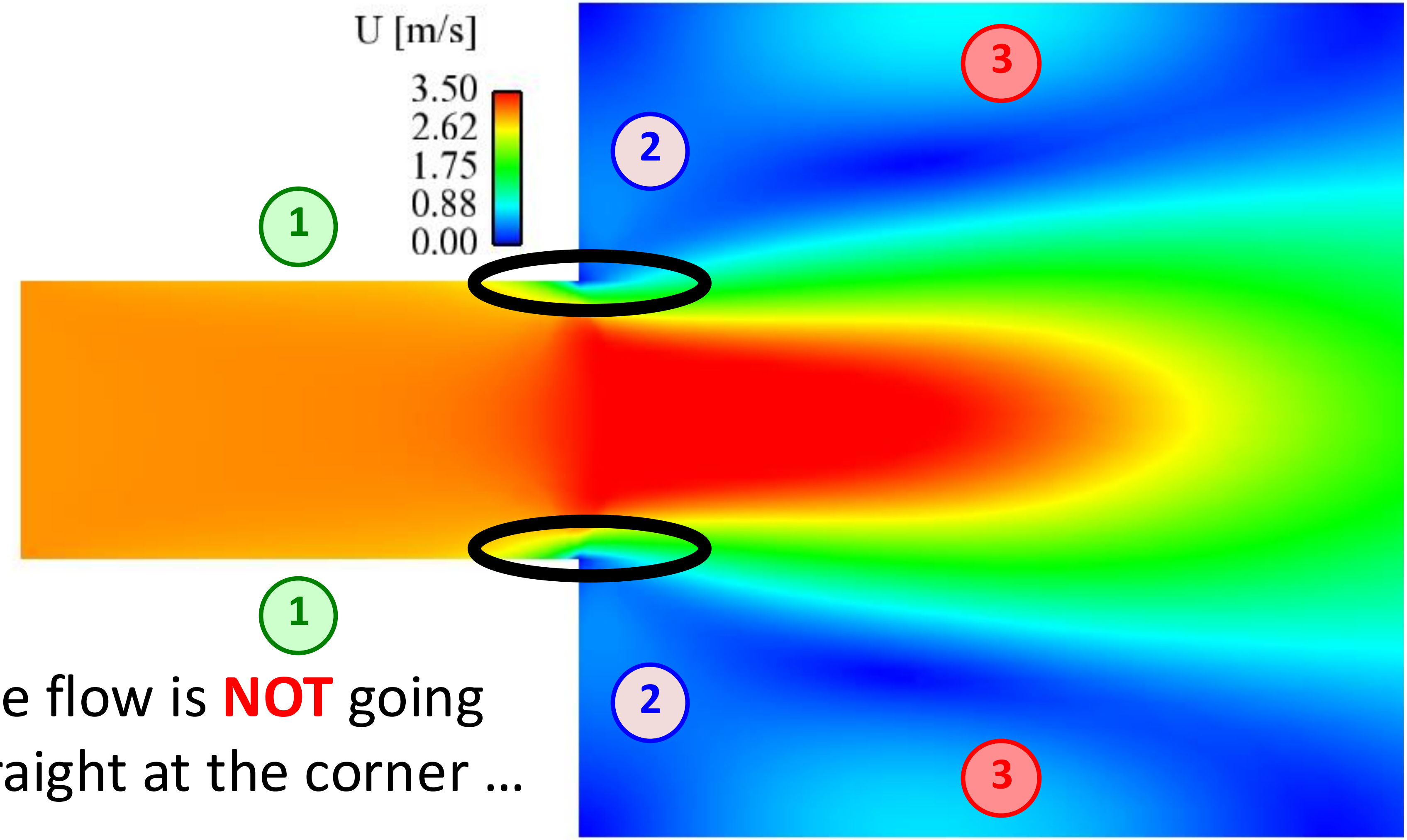
$$\vec{U} \cdot \vec{n} = 0$$



➔ Let's consider **wall Law** using **2 <u>separated</u>** at the corner



# Separated patches test case



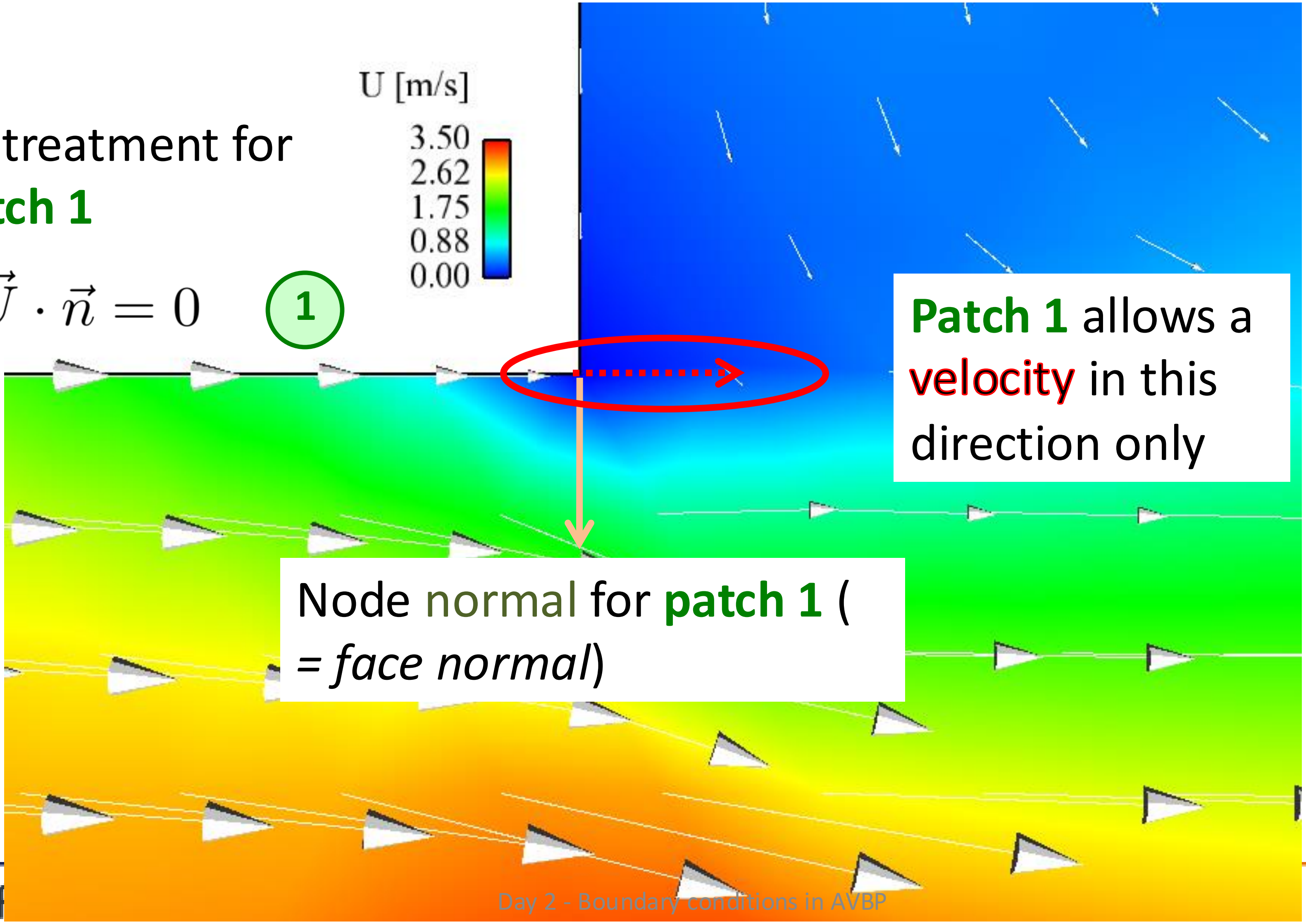

The flow is **NOT** going straight at the corner …

# Separated patches test case

BC treatment for **patch 1**

$$\vec{U} \cdot \vec{n} = 0$$

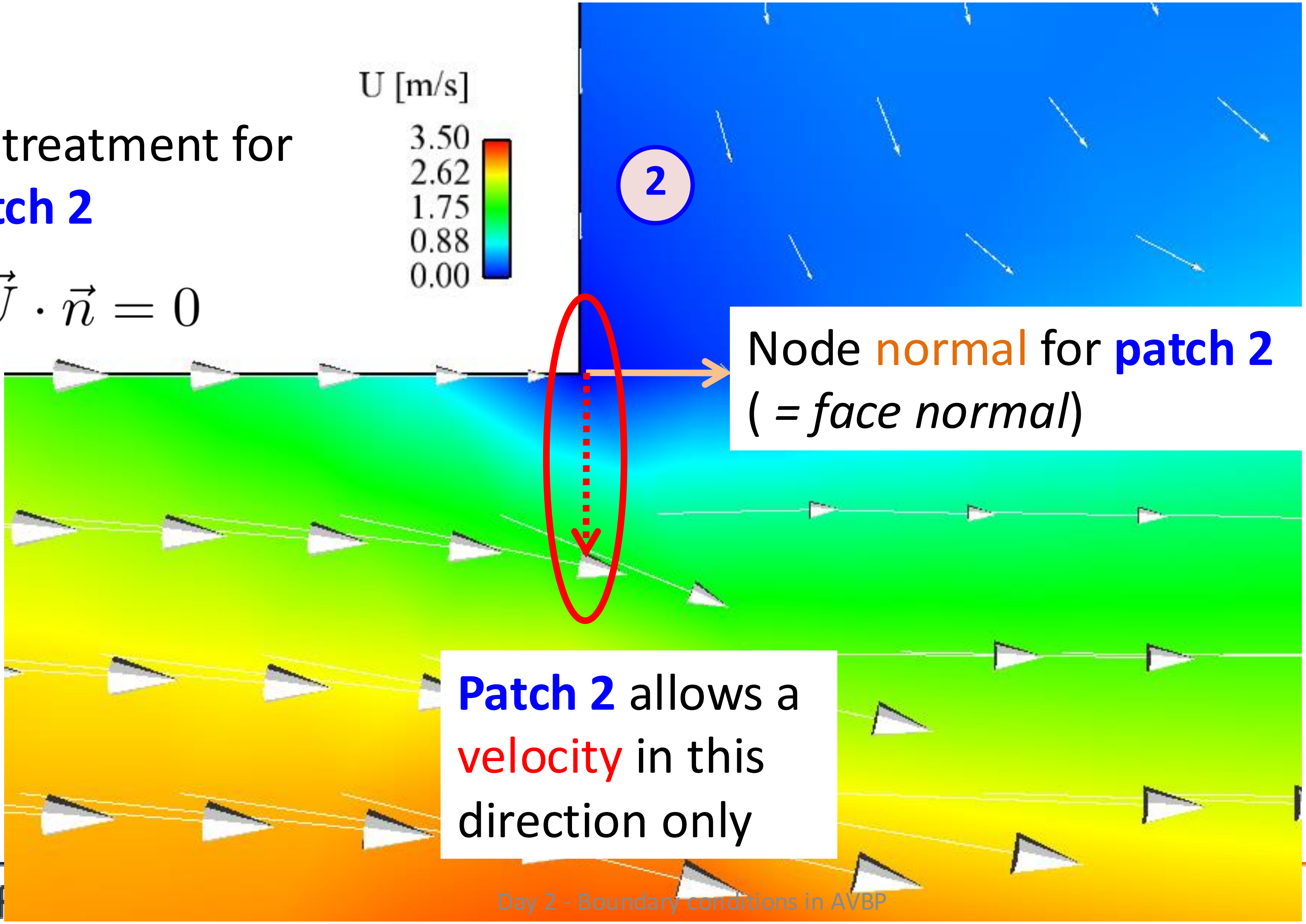
1



# Separated patches test case

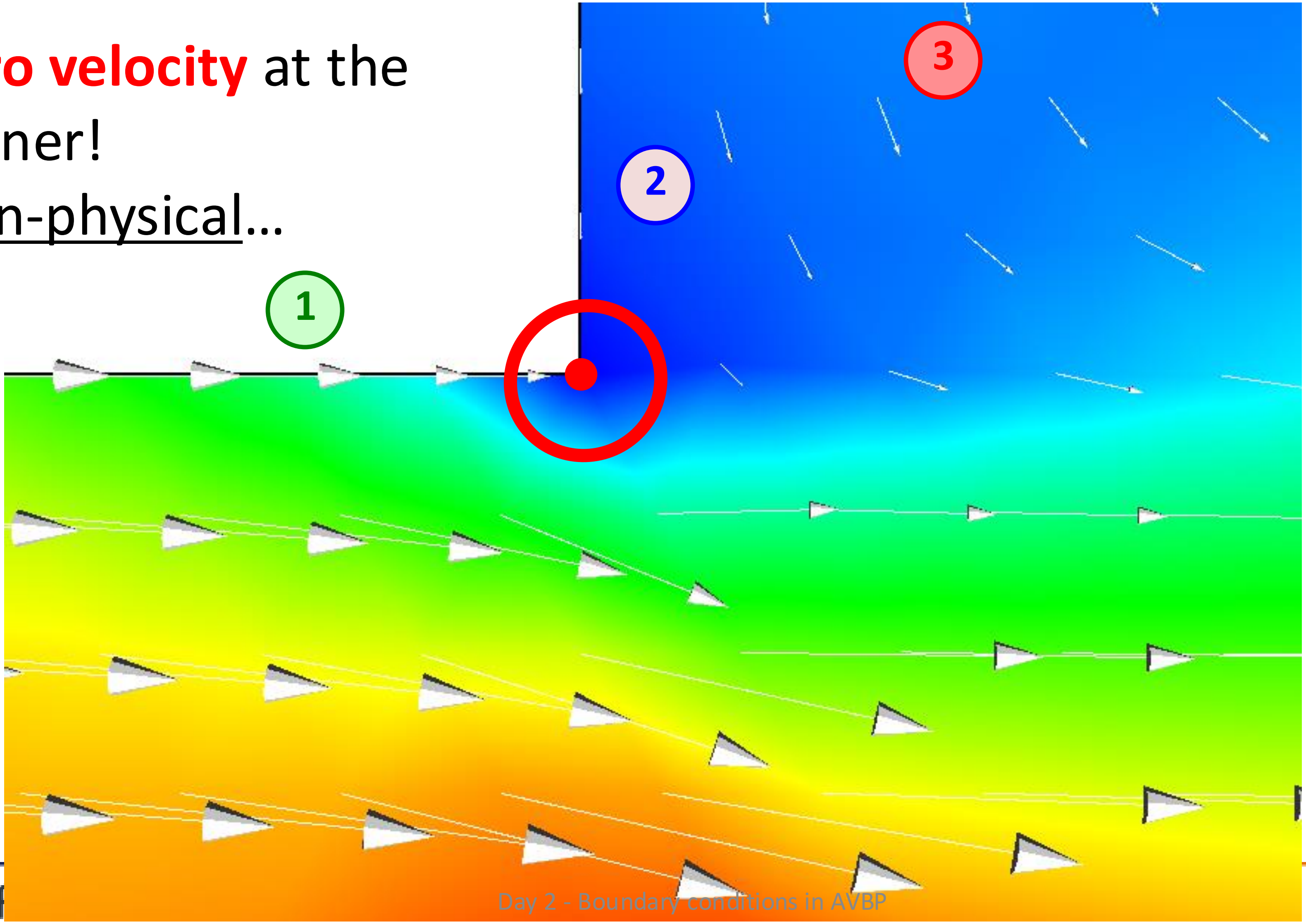
BC treatment for **patch 2**

$$\vec{U} \cdot \vec{n} = 0$$



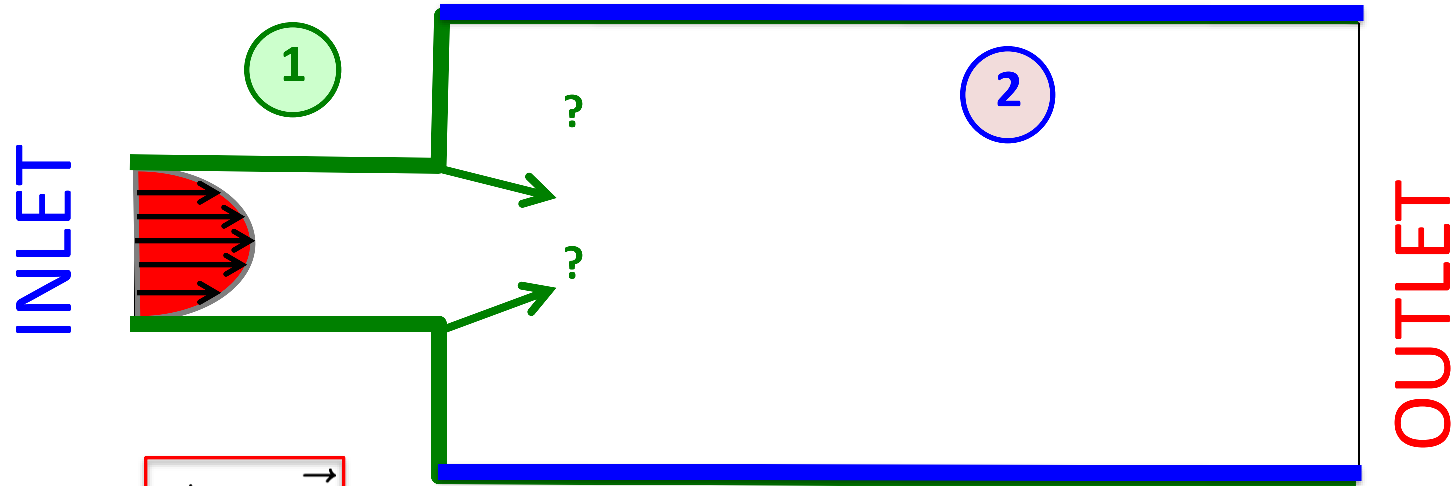
# Separated patches test case

**Zero velocity** at the corner!
Non-physical…



What is the solution ?

➔ The **free-corner procedure** !




Let's impose $\boxed{\vec{n} = \vec{0}}$ :

➔ $\vec{U} \cdot \vec{n} = 0$ is always satisfied.

➔ The flow **direction** is let **free**

➔ Additional treatments are imposed to enforce mass conservation (*PhD P. Schmitt*)

# Asciibound file

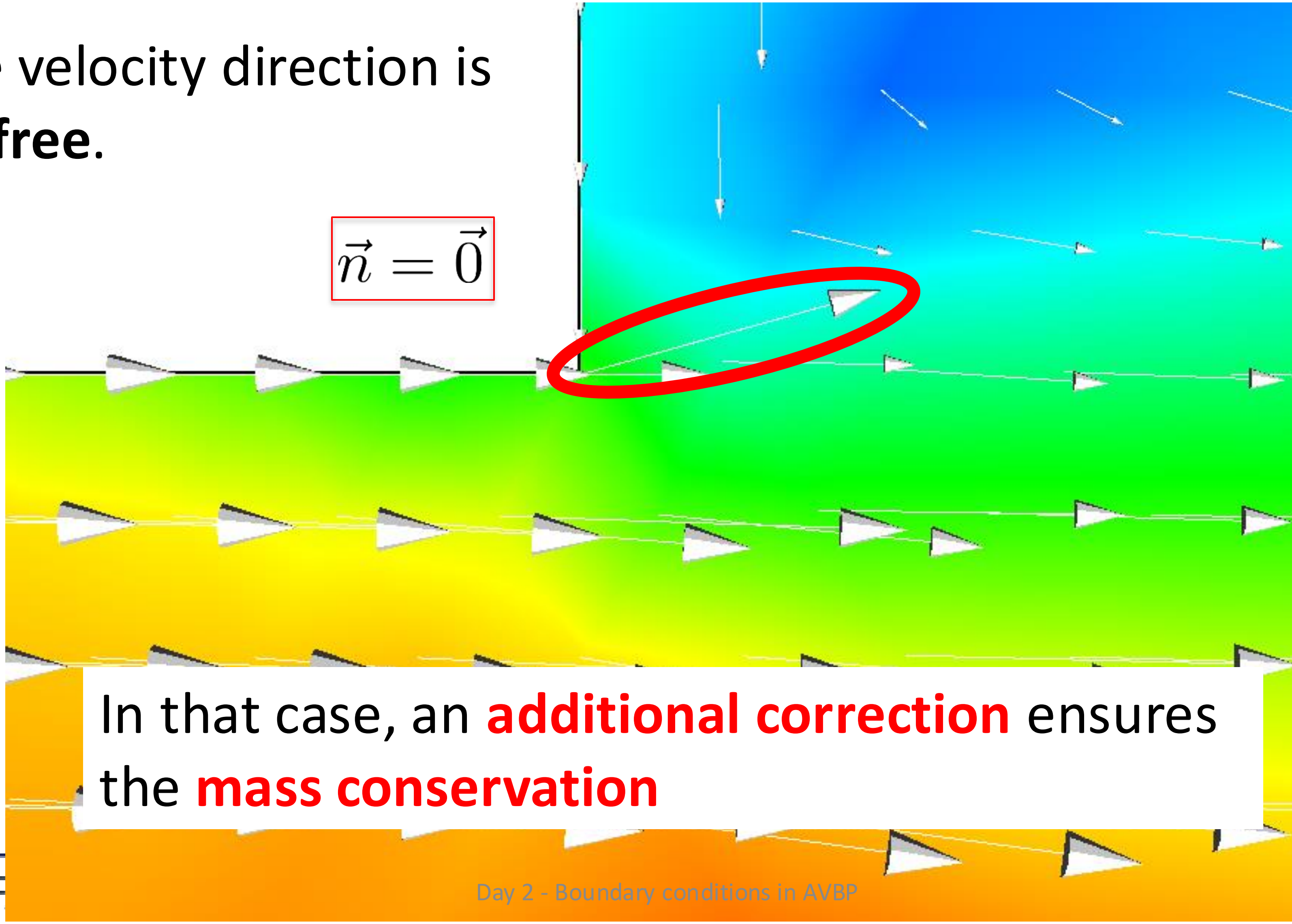
- You must enforce a single patch

```
!-------------------------
patch_name = wall_carter
boundary_condition = WALL_LAW_ADIAB
free_corner_min_angle = 60.0D00
!-------------------------
```

Activates free-corner procedure for angle between face normals > 60°

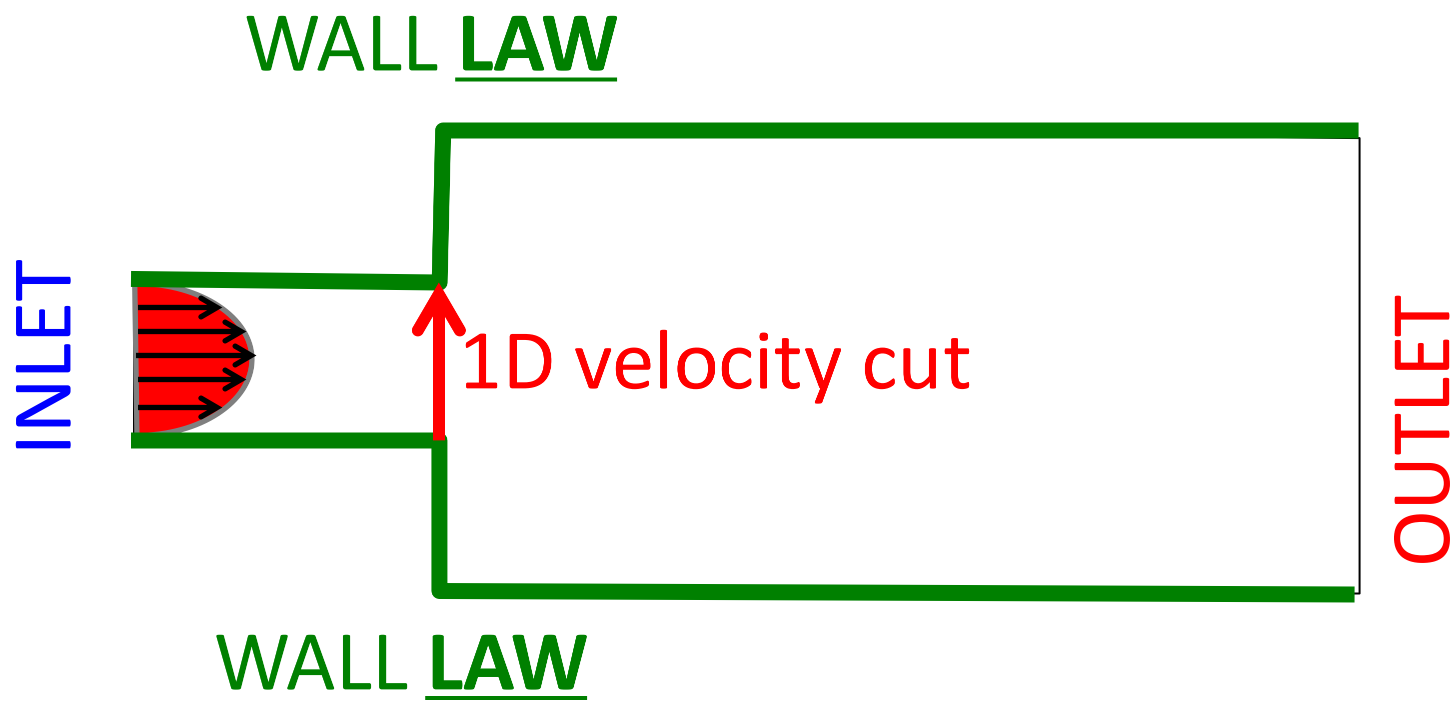
# Example of corner treatments : 1 **single patch** + free corner

The velocity direction is let **free**.



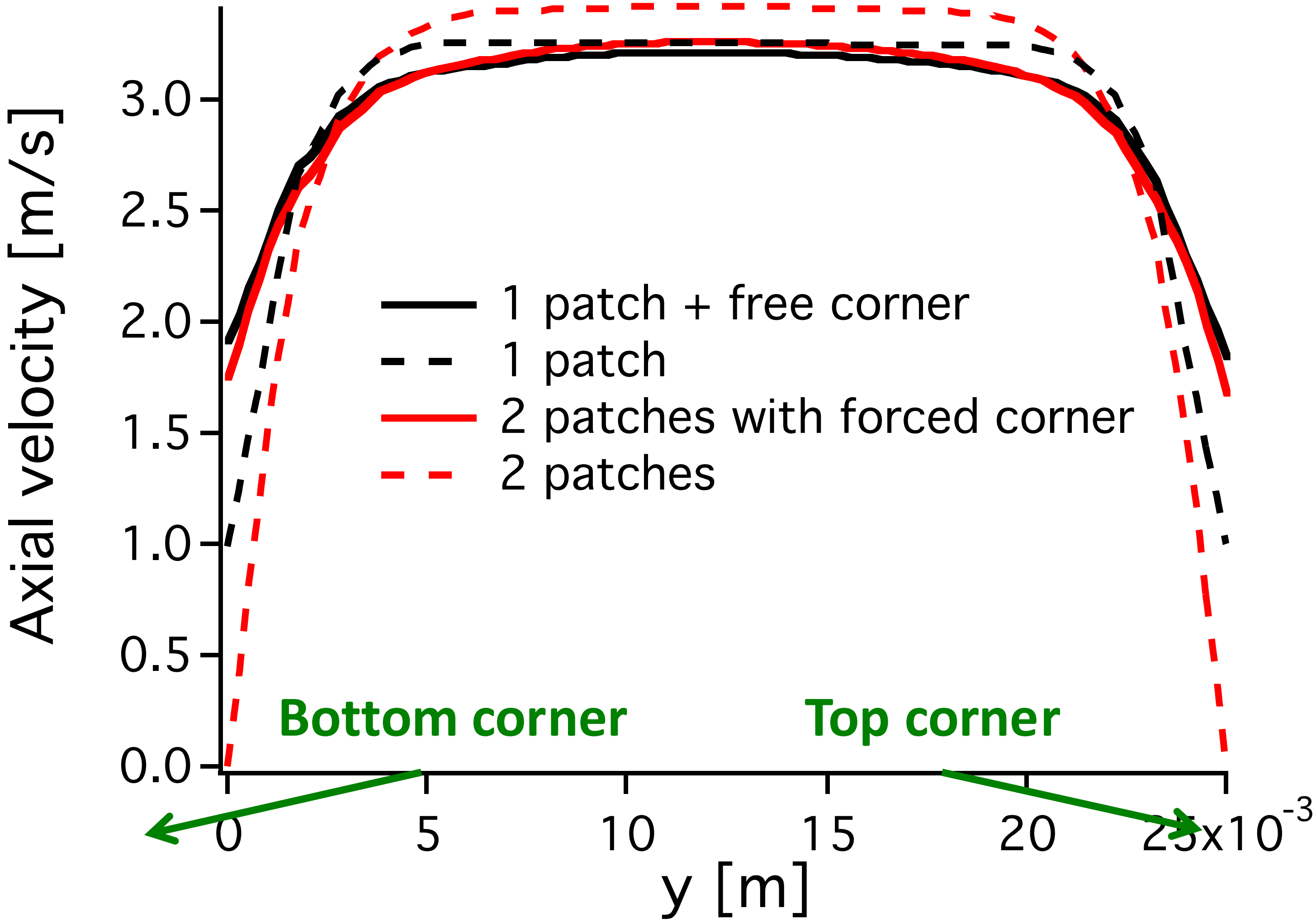

In that case, an **additional correction** ensures the **mass conservation**

# Example : sudden expansion



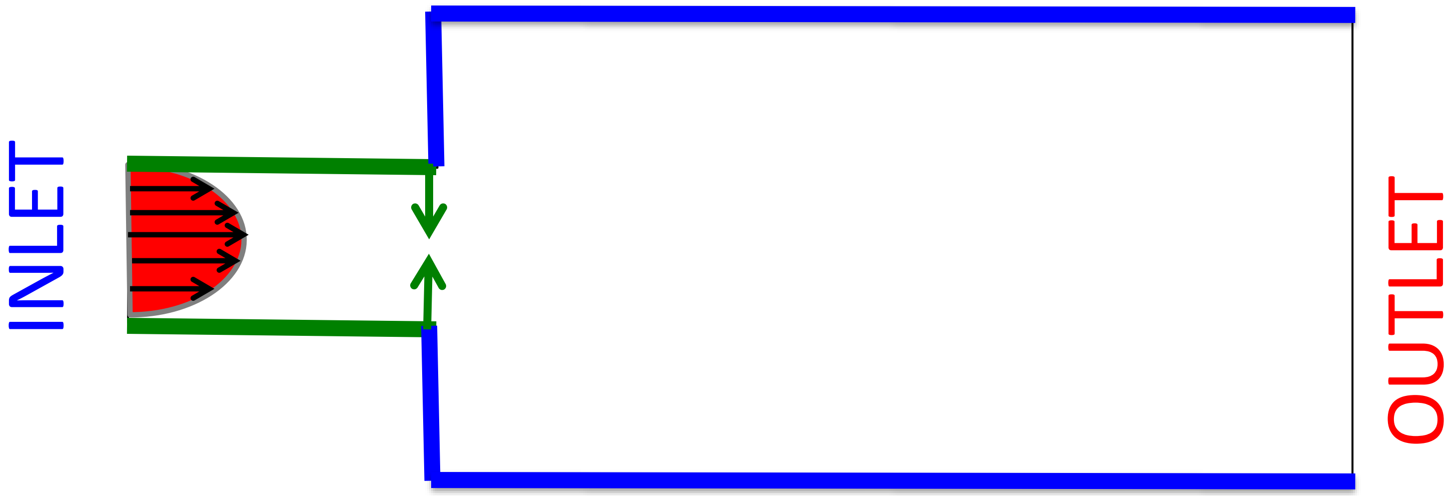

Velcity profiles at the cut position ?

# Example of corner treatments : summary



# Example : sudden expansion

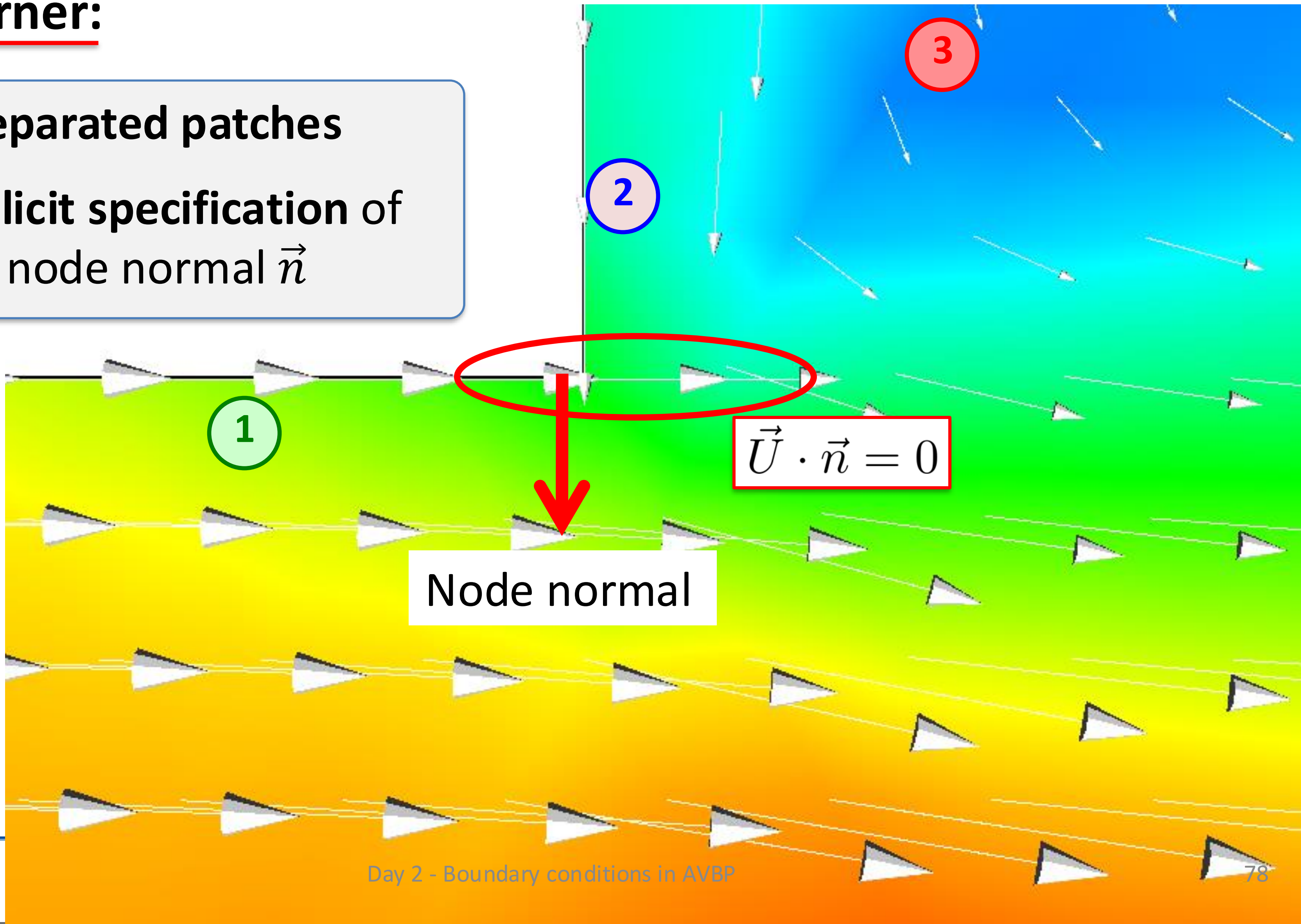
Sometimes, we may want to **enforce** the flow **direction**

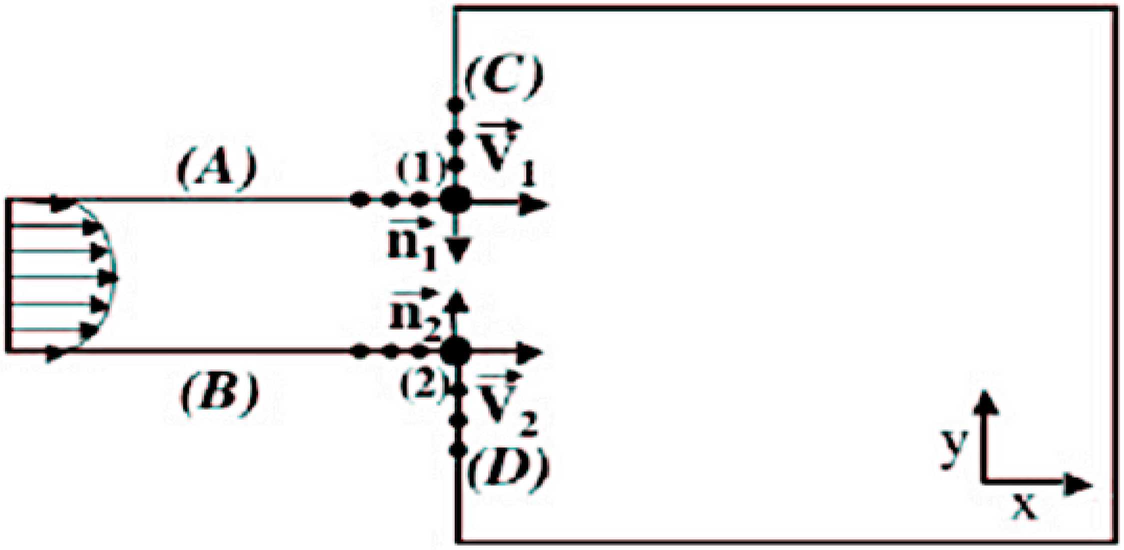



➔ The **forced-corner procedure** !

## Forced-corner:

- **2 separated patches**
- **Explicit specification** of the node normal $\vec{n}$






Separated patches A and B

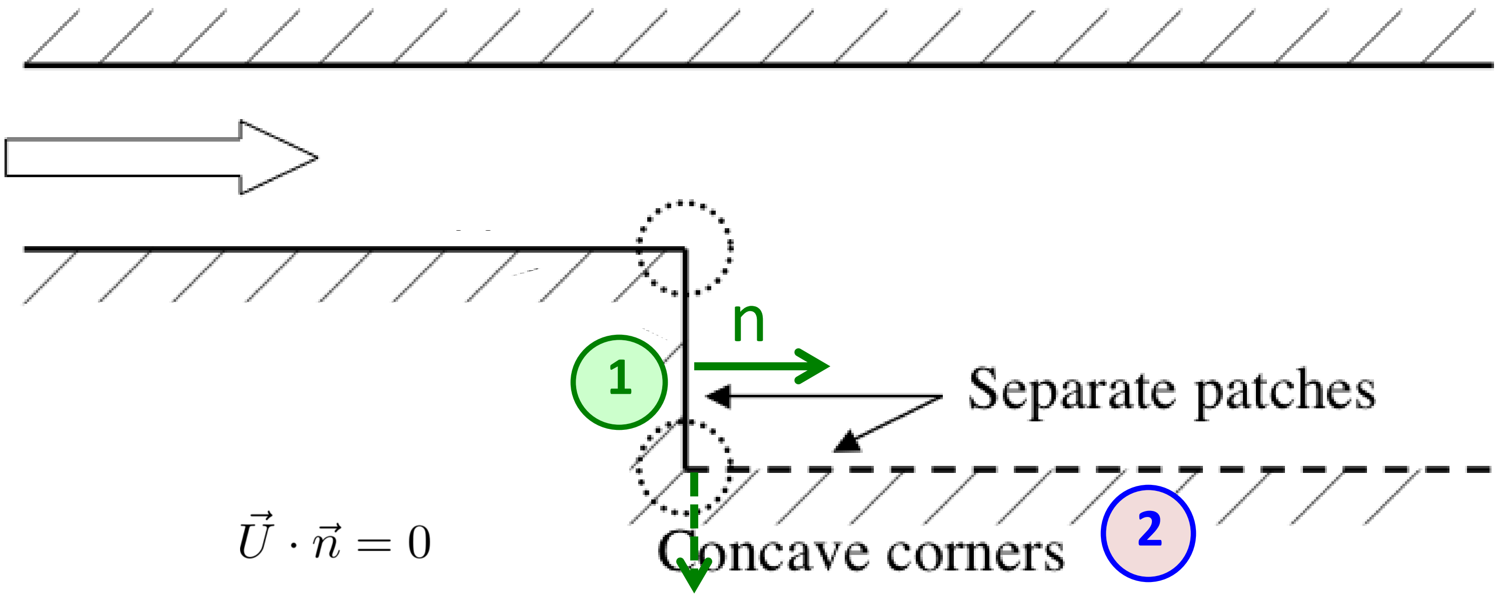
```
$BOUNDARY-PATCHES
patch_name = Patch_A
boundary_condition = WALL_LAW_ADIAB
free_corner_min_angle =   0.60D+02
forced_corner_associated_patch = Patch_C
```

**Patch_A imposes** its normal **to Patch_C**

=> **Enforce** the velocity direction

For **concave corners**, the situation is simpler:

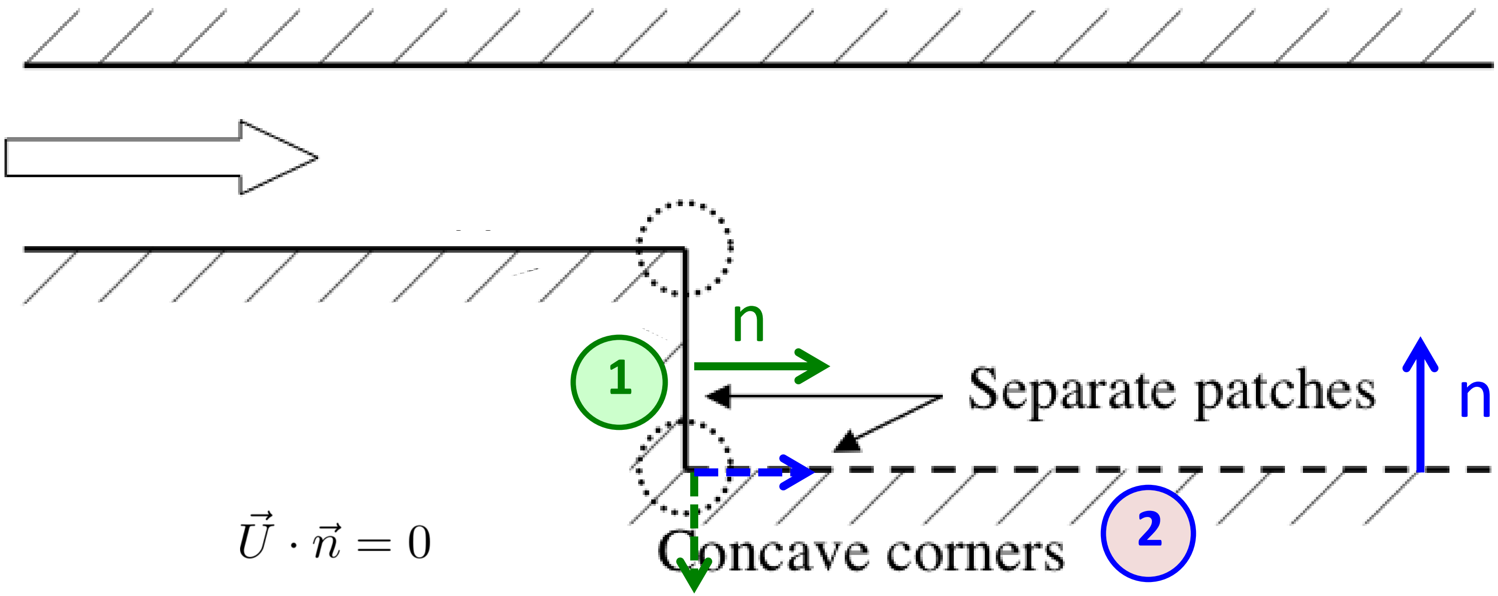
➔ The two patches (**1** and **2**) **must be separated.**




- The first treatment (**patch 1**) implies that the **velocity normal** to **1** is **zero**.

For **concave corners**, the situation is simpler:

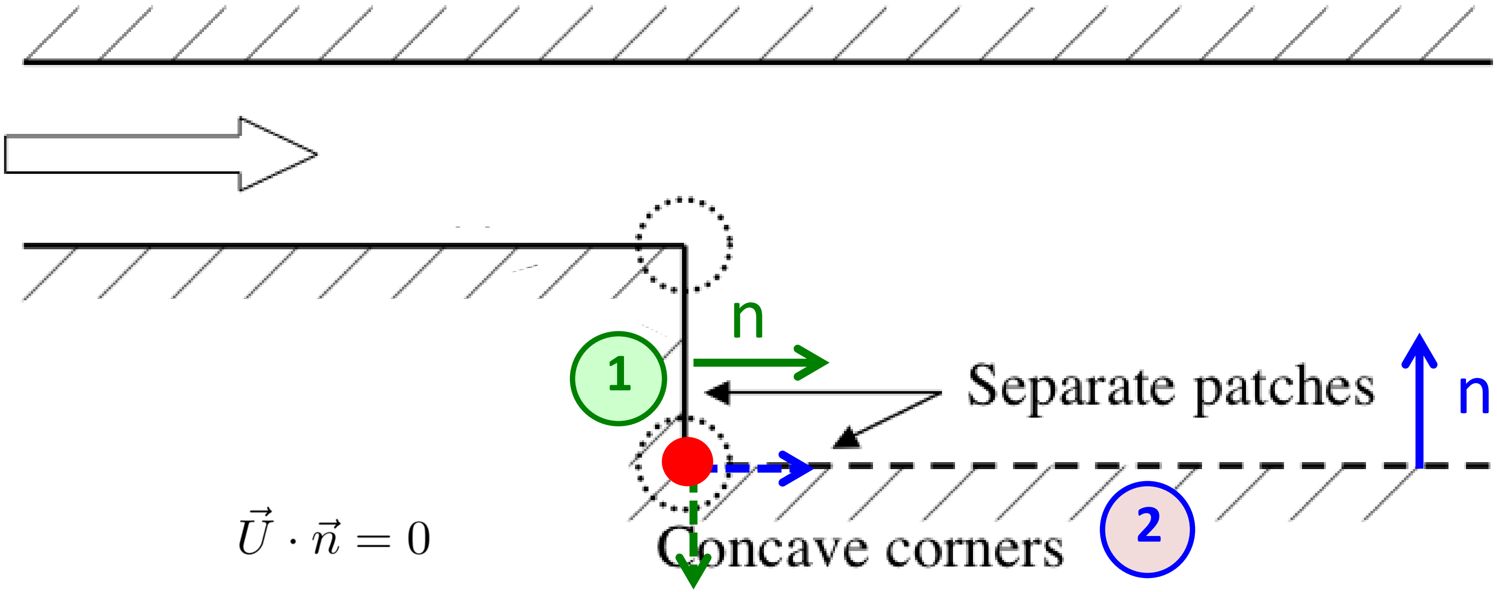
➔ The two patches (**1** and **2**) **must be separated.**




- The first treatment (**patch 1**) implies that the **velocity normal** to **1** is **zero**.
- The second treatment (**patch 2**) implies that the **velocity normal** to **2** is **zero**.

For **concave corners**, the situation is simpler:

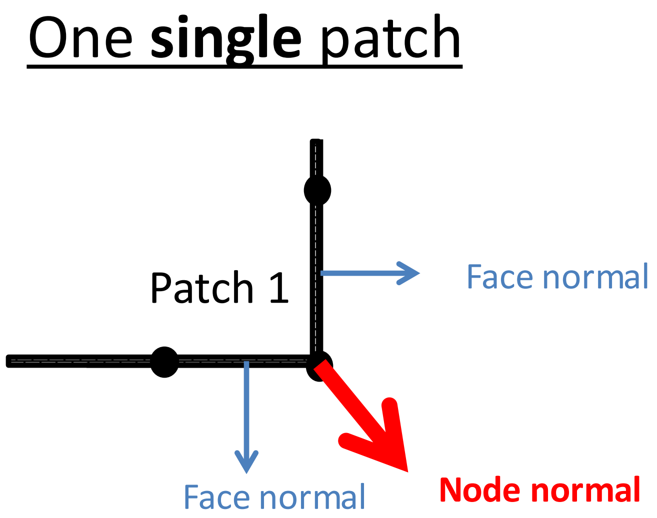
➔ The two patches (**1** and **2**) **must be separated.**

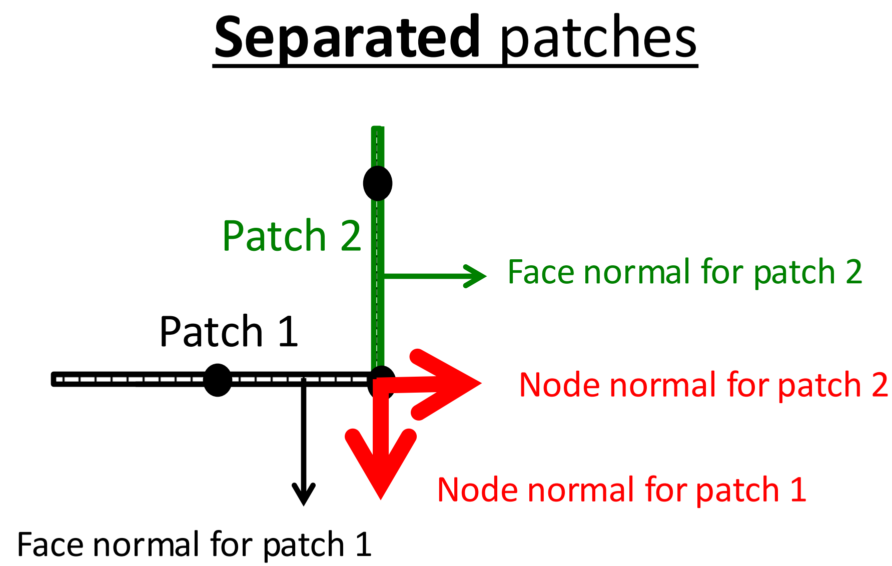



- The first treatment (**patch 1**) implies that the **velocity normal** to **1** is **zero**.
- The second treatment (**patch 2**) implies that the **velocity normal** to **2** is **zero**.

➔ $\vec{\mathrm{U}} = \vec{0}$ at the corner (●), which is physical.

# Partial conclusion on corners




➔ Recommended solution : **free-corner** approach

```
!------------------------
patch_name = wall_carter
boundary_condition = WALL_LAW_ADIAB
free_corner_min_angle = 60.0D00
!------------------------
```

# Outline

## Theoretical insights

- Mathematical background
- Numerical methods

## Description of AVBP schemes

- Lax Wendroff
- Two-step Taylor Galerkin schemes
- Properties of AVBP schemes

## Practical elements

- Wiggles
- Issues at corners
- Artificial viscosity
- Shock sensors

# Artificial viscosity

- Numerical schemes in AVBP are **centred schemes.** They are designed to **be low-dissipative**

- They however suffer from **dispersion errors** (as any numerical scheme)

- When Reynolds number increases, **dissipation** operators **may not allow to damp spurious errors**

→ Solution adopted in AVBP: **artificial viscosity**.

# Artificial viscosity

Artificial viscosity in AVBP combines **two different types of operators** :

$$\boxed{\frac{d\mathbf{U}_j}{dt} = \mathbf{R}_j + \mathbf{D}_j^{(2*)} + \mathbf{D}_j^{(4*)}}$$

residual

2nd order artificial viscosity

4th order artificial viscosity

2nd and 4th order operators rely on **sensors**

➔ Let's have a look in these operators, and sensors…

$$\frac{d\mathbf{U}_j}{dt} = \mathbf{R}_j + \mathbf{D}_j^{(2*)} + \mathbf{D}_j^{(4*)}$$

# Second order AV

- 2nd order viscosity is based on a pseudo-Laplacian formulation
- Allow to damp **non-linearities**, stiff gradients

**smu2** **Sensor** ← **Defined in the run.params file**

(Detect local non-linearities)

(jameson, colin, etc…)

$$\boxed{\mathbf{D}_j^{(2)} = \frac{1}{V_j} \sum_{e \in \mathcal{D}_j} -\frac{\epsilon^{(2)} \zeta_e V_e}{n_v^e \Delta t} \underbrace{(\overline{\mathbf{U}}_e - \mathbf{U}_j)}}$$

Nodal volume

Difference between cell-averaged and nodal value

```
artificial_viscosity_model =  colin
artificial_viscosity_2nd_order =  0.01D+00
artificial_viscosity_4th_order =  0.005D+00
```

← **Sensor**
← **smu2**

$$\frac{d\mathbf{U}_j}{dt} = \mathbf{R}_j + \mathbf{D}_j^{(2*)} + \mathbf{D}_j^{(4*)}$$

# Fourth order AV

- Contrarily to 2nd order operator, 4th order operator **operates in the entire domain**
- Allows to damp **node-to-node oscillations**

**Smu4** **Sensor**

$$\epsilon_e^{(4*)} = \max\left(0, \epsilon^{(4)} - \zeta_e \epsilon^{(2)}\right)$$

$$\boxed{\mathbf{D}_j^{(4)} = \frac{1}{V_j} \sum_{e \in \mathcal{D}_j} \frac{\epsilon_e^{(4*)} V_e}{n_v^e \Delta t} \left[ \left( \frac{1}{n_v^e} \sum_{k \in K_e} \vec{\nabla} \mathbf{U}_k \right) \cdot (\vec{x}_e - \vec{x}_j) - (\mathbf{U}_e - \mathbf{U}_j) \right]}$$

Nodal volume

Bi-Laplacian operator
(approximates a 4th-order derivatives)

➔ **AV4 is applied everywhere**, excepted in region where AV2 occurs

# **Sensors**: Jameson

## **Jameson sensor** at node *k*:

$$\zeta_k = \frac{\left|\Delta_1^k - \Delta_2^k\right|}{\left|\Delta_1^k\right| + \left|\Delta_2^k\right| + |\phi_k|}$$

With:

$\phi_k$: nodal scalar quantity the sensor is based on

$$\Delta_1^k = \bar{\phi}_C - \phi_k$$

$$\Delta_2^k = \nabla\phi_k \cdot (\mathbf{x}_C - \mathbf{x}_k)$$

In AVBP, sensor applied on all conservatives variables, excepted species

## Sensor at cell

$$\zeta_C^{Jameson} = \max_{k \in C} \zeta_k$$

**Operators equivalence for a 1D regular mesh:**

$$\Delta_1^k = \frac{\phi_{k+1} - \phi_k}{2}$$

$$\Delta_2^k = \frac{\phi_{k+1} - \phi_{k-1}}{4}$$

$$\left|\Delta_1^k - \Delta_2^k\right| = \left|\frac{1}{4}\left(\phi_{k+1} - 2\phi_k + \phi_{k-1}\right)\right|$$

- Proportional to 2nd order derivative of ϕ.

➔ Detect variations of $\vec{\nabla}\phi$ (wiggles). **Varies linearly with the perturbation**

- **However, also dissipates small scales of turbulence.**

➔ Not so adapted for LES. May be useful for initialization.

## Colin sensor:

$$\zeta_C^{Colin} = \frac{1}{2}\left(1 + \tanh\left(\frac{\Psi - \Psi_0}{\delta}\right)\right) - \frac{1}{2}\left(1 + \tanh\left(\frac{-\Psi_0}{\delta}\right)\right)$$

with

$$\Psi = \max_{k \in C}\left(0, \frac{\Delta^k}{|\Delta^k + \epsilon_1 \phi_k|}\zeta_C^{Jameson}\right)$$
$$\Delta^k = |\Delta_1^k - \Delta_2^k| - \epsilon^k \max\left(|\Delta_1^k|, |\Delta_2^k|\right)$$
$$\epsilon^k = \epsilon_2\left(1 - \epsilon_3 \frac{\max\left(|\Delta_1^k|, |\Delta_2^k|\right)}{|\Delta_1^k| + |\Delta_2^k| + |\phi_k|}\right)$$

$\Psi_0 = 2.10^{-2}$ $\quad \delta = 1.10^{-2}$

$\epsilon_1 = 1.10^{-2}$ $\quad \epsilon_2 = 0.95$ $\quad \epsilon_3 = 0.5$

- $\xi_C^{Colin}$ **small for**: low amplitude errors, or stiff gradients well resolved by the scheme
- $\xi_C^{Colin}$ **large for**: high amplitude numerical oscillation

➔ **Colin sensor works more in a on/off way than Jameson**

AV2, AV4 applied on $(\rho, \rho E, \rho Y_k)$

- **Colin sensor is better suited for LES than Jameson**
- In AVBP, variables considered for AV with Colin allows good LES of reacting flows



O. Colin. Simulations aux grandes échelles de la combustion turbulente prémélangée dans les statoréacteurs. PhD Thesis, Université de Toulouse, 2000.

# Artificial Viscosity, summary

## Reminder on Artificial Viscosity

- Choices for the sensor (Jameson, Colin, Colin_rhou (SLK), …)
- Choice for the level of 2nd order and 4th order applied (smu2, smu4)
- Choice for the variable on which AV applies

- **AV is required** for LES of actual configurations…

… but an **inaccurate AV can lead to physical issues** (wrong operating point, …)

- Do not confuse: ***subgrid-scale*** viscosity and ***artificial*** viscosity !

See AVBP website (dedicated page) for recommendations !

### Available AV options in ABVP

At present time, one can chose between 8 values for the keyword `artificial_viscosity_model` :

- no: no A.V. ("Academic" LES or true DNS).
- honey: "honey" mode, sensor fully active (equal to 1), the flow is very viscous, should only be used for transients (Initialization of the computation).
- Jameson: A.V. for "aerodynamics" (based on Jameson sensor).
- Jameson_species: A.V. for "aerodynamics" (based on Jameson sensor). An additional sensor is applied for the species mass fractions.
- Colin: A.V. for LES and/or combustion (based on Colin sensor).
- Colin_species: A.V. for LES and/or combustion (based on Colin sensor). An additional sensor is applied for the species mass fractions.
- Colin_rhou: A.V. for LES and/or combustion (based on Colin sensor), applying besides artificial viscosity upon pu.
- Colin_rhou_species: A.V. for LES and/or combustion (based on Colin sensor), applying besides artificial viscosity upon pu. An additional sensor is applied for the species mass fractions.

# AVBP run.params file

```
$RUN-CONTROL
  solver_type = ns
  diffusion_scheme = FE_2delta
  simulation_end_time = 1.00000000d-03
  mixture_name = AIR
  reactive_flow = no
  combustion_model = no
  two_phase_flow = no
  real_gas = no
  LES_model = sigma
  prandtl_turb =   0.60000000D+00
  schmidt_turb =   0.60000000D+00
  convection_scheme = LW
  CFL =   0.90000000D+00
  Fourier =   0.10000000D+00
  compute_chemical_timestep = no
  artificial_viscosity_model =  colin
  artificial_viscosity_2nd_order =  0.01D+00
  artificial_viscosity_4th_order =  0.005D+00
  clip_species = no
$end_RUN-CONTROL
```

Euler / Navier-Stokes

Diffusion scheme.
*(FE_2delta only choice since AVBP_7.7)*

LW / TTG4A / TTGC (+ many others !)

CFL number, if time-step is not imposed

AV Sensor model

smu2

smu4

You should be able to understand, and accurately set these parameters for your case !

CERFACS

# AVBP WEBSITE

AVBP website

Home

General Help
2
Versions
Handbook
QPF table
HIP
Chemistry
Movie
Gallery
Bug store
2
Training

Home

Input Output Tools Docs Install Tutorials FAQ

Copyright

**THIS IS THE WEBSITE OF THE AVBP CODE OF CERFACS**

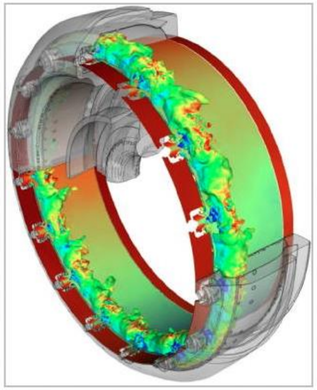
**You can find here examples of simulations as movies or images (in the Gallery). You can also have information on AVBP trainings. If you are an AVBP user, you can also find here information on the AVBP input files, on the scientific manual, the list of test cases (QPFs), the chemistry site of CERFACS and the bug store.**

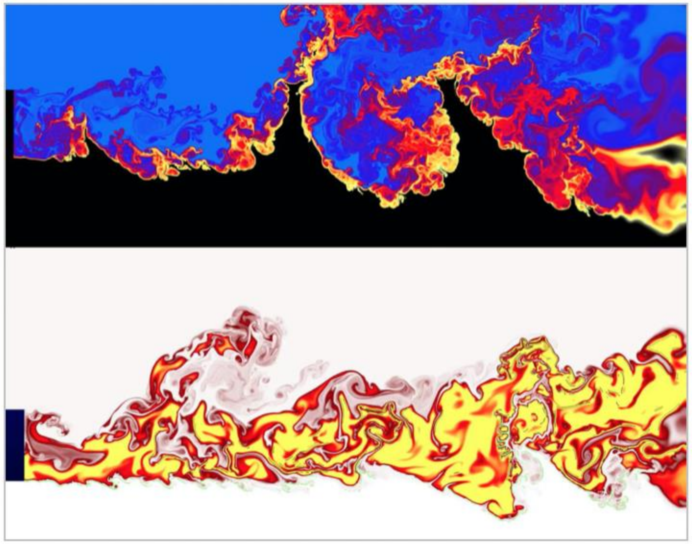
**AVBP: a leader in the field of simulation of turbulent reacting flows**

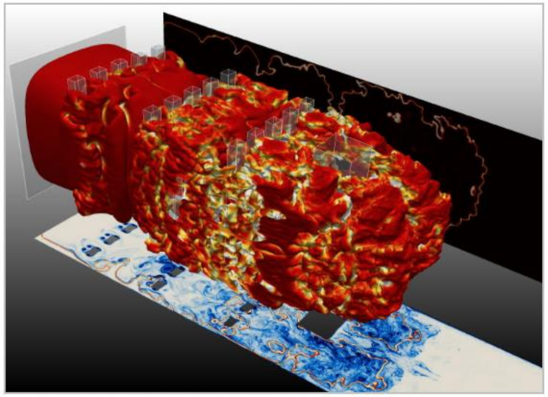
AVBP is a suite of CFD tools to perform DNS and LES of compressible, reacting, turbulent, multispecies flows, on massively parallel CPU and GPU architectures. AVBP is widely used for basic research in multiple laboratories. It is also the design CFD tool for many french companies using LES. The AVBP project is one of the largest CFD teams worldwide focusing on reacting flows with more than 60 researcher scientists and engineers.

**AVBP: a leader in the field of High Performance Computing**

AVBP is at the forefront of HPC developments. It is a reference code in the HPC community, used today in the COEC European Center for Excellence. AVBP is also used for LES of non-reacting flows (turbomachinery), safety CFD (pollution dispersion, virus airborne transport) and environmental problems (solar, wind farms). AVBP is also aplied to multiphysics problems in collaboration with ONERA using the CWIPI coupler technology.







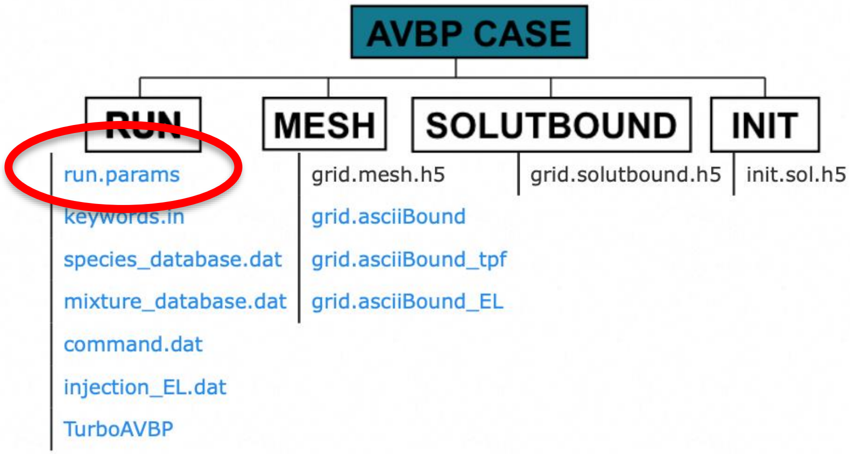
# AVBP website

Home
General help 6
Versions 2
Handbook
QPF table
HIP
Chemistry
Movie
Gallery
Bug store
Training 2

**Input files**

Home › General help › Input files

This is what a typical AVBP simulation looks like:




Several files are mandatory to perform a run with AVBP, others are optional. The following sections and links describe how to generate or fill in these files.

The parameter files

Four parameter files are needed to parametrize a computation with AVBP (two are dedicated to Euler-Lagrange simulation) :

- `keywords.in`: this file is the rule file. It specifies the name of the main input file of your computation (default name = 'run.params') and it also contains a list of keyw
all the options available for a computation in AVBP.
This file is specific to a given version of AVBP and may be retrieved for each distribution of AVBP in the directory `$AVBP_HOME/WORK/INPUT`.
For a technical description of this file, see this page.

| | | | |
|---|---|---|---|
| `convection_scheme.boundary_terms` | character | closed<br>unclosed | Controls the boundary term treatment. Only valid for convective schemes based on a Lax-Wendroff-like method (total discretization).<br>Default values are:<br>• closed if `convection_scheme` = 'TTGC' or 'TTG4A'.<br>• unclosed if `convection_scheme` = 'LW' or 'LW_FE'.<br>• Not used otherwise.<br>More information is available in this document. |
| `diffusion_scheme` | character | FV_4delta<br>FE_2delta | The diffusion scheme:<br>• FV_4delta : the original finite volume 4Δ diffusion scheme.<br>• FE_2delta : the finite element 2Δ diffusion scheme developped by O. Colin.<br>More information is available in this document.<br>Default: FE_2delta if `solver_type` = 'NS'. Not used otherwise. |
| `CFL` | double | - | The CFL number used to compute the convective time-step. |

Very interesting pdf files are available on the website trough this document !

# Good readings

- AVBP-Website
- AVBP Handbook
- AVBP QPF

PhD thesis

- N. Lamarque
- L.M Segui-Troth
- B. Martin

https://elearning.cerfacs.fr

## Books:


- Hirsch, C. (1990) Numerical Computation of Internal and External Flows. Wiley, Hoboken.
- G. Allaire. Analyse numérique et optimisation. Les Editions de l'Ecole Polytechnique, 2012
- J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley–Blackwell, 2003
- T. Poinsot and D. Veynante. Theoretical and Numerical Combustion. R.T. Edwards Inc., 2005. Ch. 9
- R. Vichnevetsky and J. Bowles. Fourier analysis of numerical approximations of hyperbolic equations. SIAM, 1982.
- B. Després, F. Dubois. Systèmes hyperboliques de lois de conservation. Les Editions de l'Ecole Polytechnique, 2005
- Godlewski, E., and Raviart, P. A., *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Springer-Verlag, New York, 1996
- *Numerical Recipes in FORTRAN: The Art of Scientific Computing*