LES of turbulent reacting flows

T. Poinsot

- **Research director CNRS, IMFT**
- **Head CFD team at CERFACS**
- **French Academy of Sciences**
- **Senior research fellow Stanford**

Ghent, January 2020

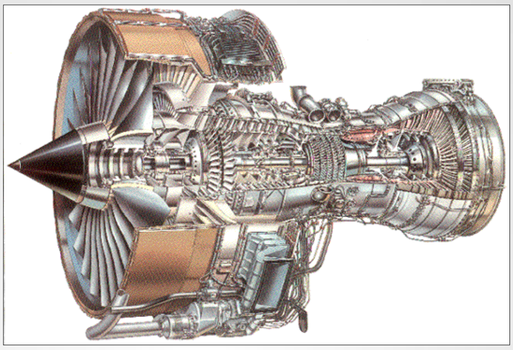
OUTLINE

- Energy policies and combustion
- Tools to simulate reacting flows
- Turbulent premixed flames and explosions
- Deep learning for turbulence combustion interaction

ENERGY AND COMBUSTION

ENERGY ON EARTH TODAY = COMBUSTION

Combustion: more than 85 percent of the energy produced



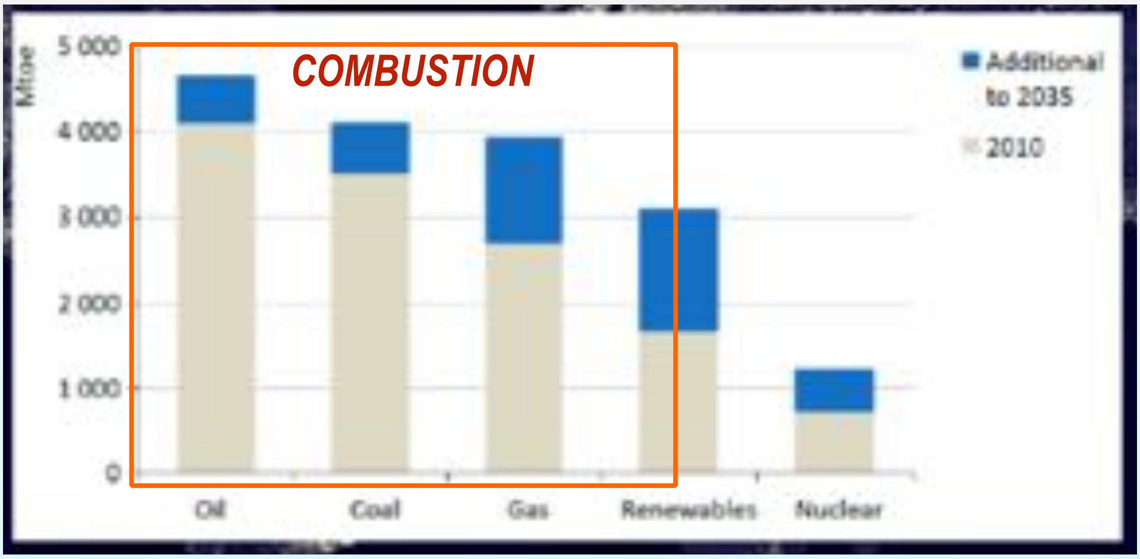
COMBUSTION OVERVIEW

Two important equations:

ENERGY ON EARTH TODAY = COMBUSTION

ENERGY ON EARTH TOMORROW = COMBUSTION

IN MOST SCENARIOS, THE ABSOLUTE ENERGY PRODUCTION USING COMBUSTION RISES BECAUSE THE INCREASE OF GLOBAL ENERGY NEEDS CANNOT BE SATISFIED BY RENEWABLES SOURCES ONLY…



Whatever the scenario is, need the best combustion systems: optimize efficiency, minimize pollutants and CO2 emission

Importance of Combustion

Total Primary Energy Supply in 2016: **13.7 Gtoe**
(10 Gtoe in 2000)

13.9% renewable
- Hydro (2.5%)
- Bio & waste (9.8%)
- Solar, geo, etc. (1.6%)

4.9% Nuclear

81.2% Fossil fuel
- Coal (27.1%)
- Oil (32%)
- Natural gas (22.1%)

Total for combustion: 91%
(91.5% in 1990)

1 toe = 41.855 GJ = 11.628 MWh
https://www.iea.org/statistics

So:

- ★We burn a lot
- ★We will keep burning a lot
- ★COMBUSTION SCIENCE MUST ALLOW US TO DO THIS WITHOUT WASTING FUEL, INCREASING POLLUTION, KILLING PEOPLE AND CHANGING THE GLOBAL CLIMATE

The place of simulation

- Of course, everyone knows and agrees that we need simulation to design better combustion systems
- The real question is: which type of simulation ?
- I will try to convince you today that this should be LES: Large Eddy Simulation

Which equations ?

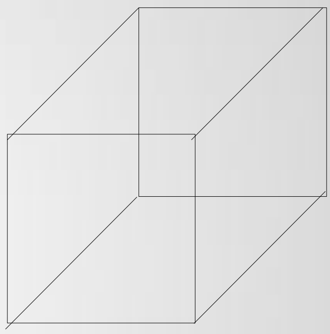
The reacting Navier Stokes equations:

- are well known
- are exact !

THE MAIN PROBLEM REMAINS TURBULENCE !

Do we really know what turbulence is ?

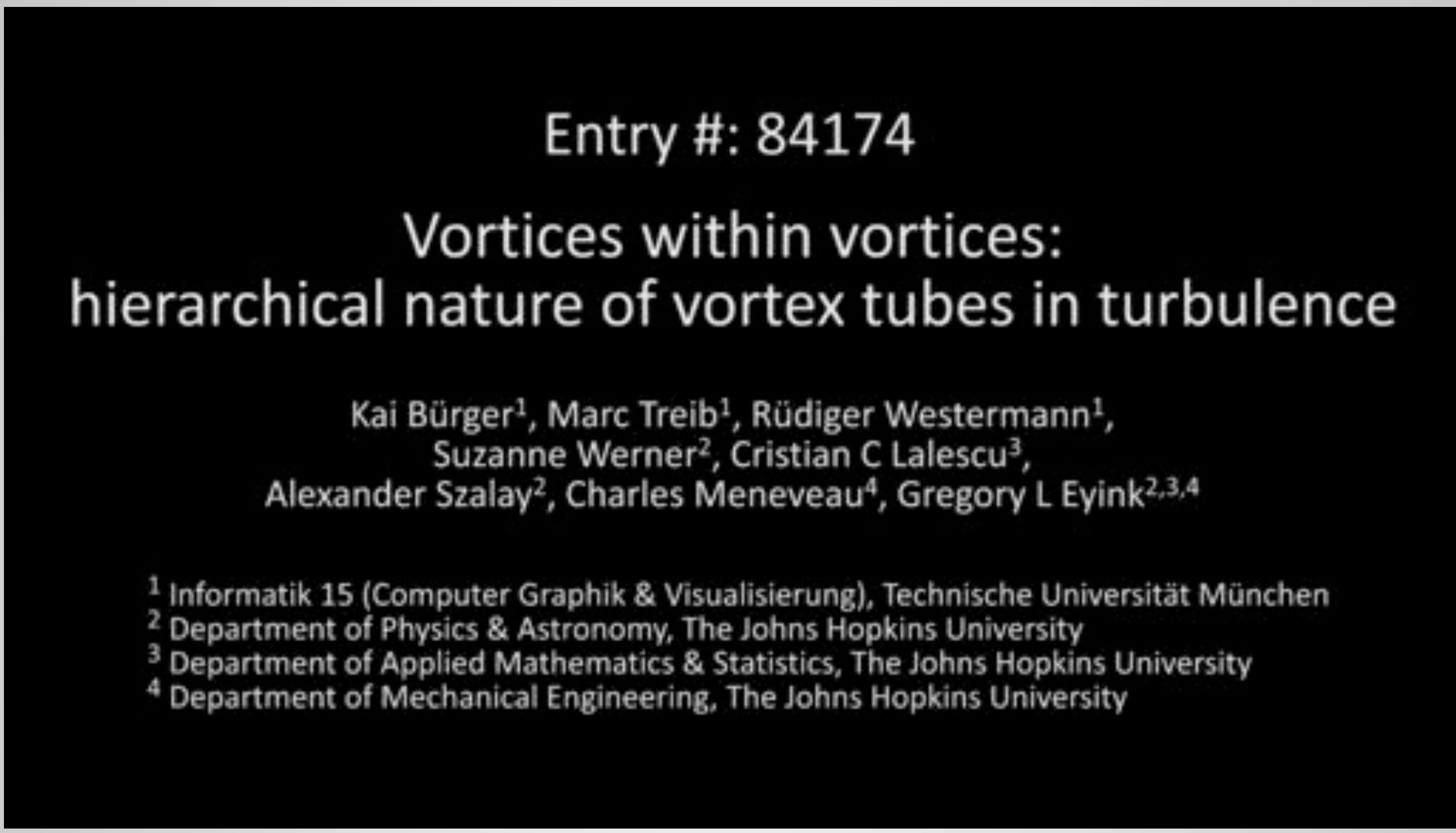
Visualization of vortices in a square box of isotropic turbulence (no combustion)
1 billion points



Vortices within vortices:
hierarchical nature of vortex tubes in turbulence

Kai Bürger[1], Marc Treib[1], Rüdiger Westermann[1],
Suzanne Werner[2], Cristian C Lalescu[3],
Alexander Szalay[2], Charles Meneveau[4], Gregory L Eyink[2,3,4]

[1] Informatik 15 (Computer Graphik & Visualisierung), Technische Universität München
[2] Department of Physics & Astronomy, The Johns Hopkins University

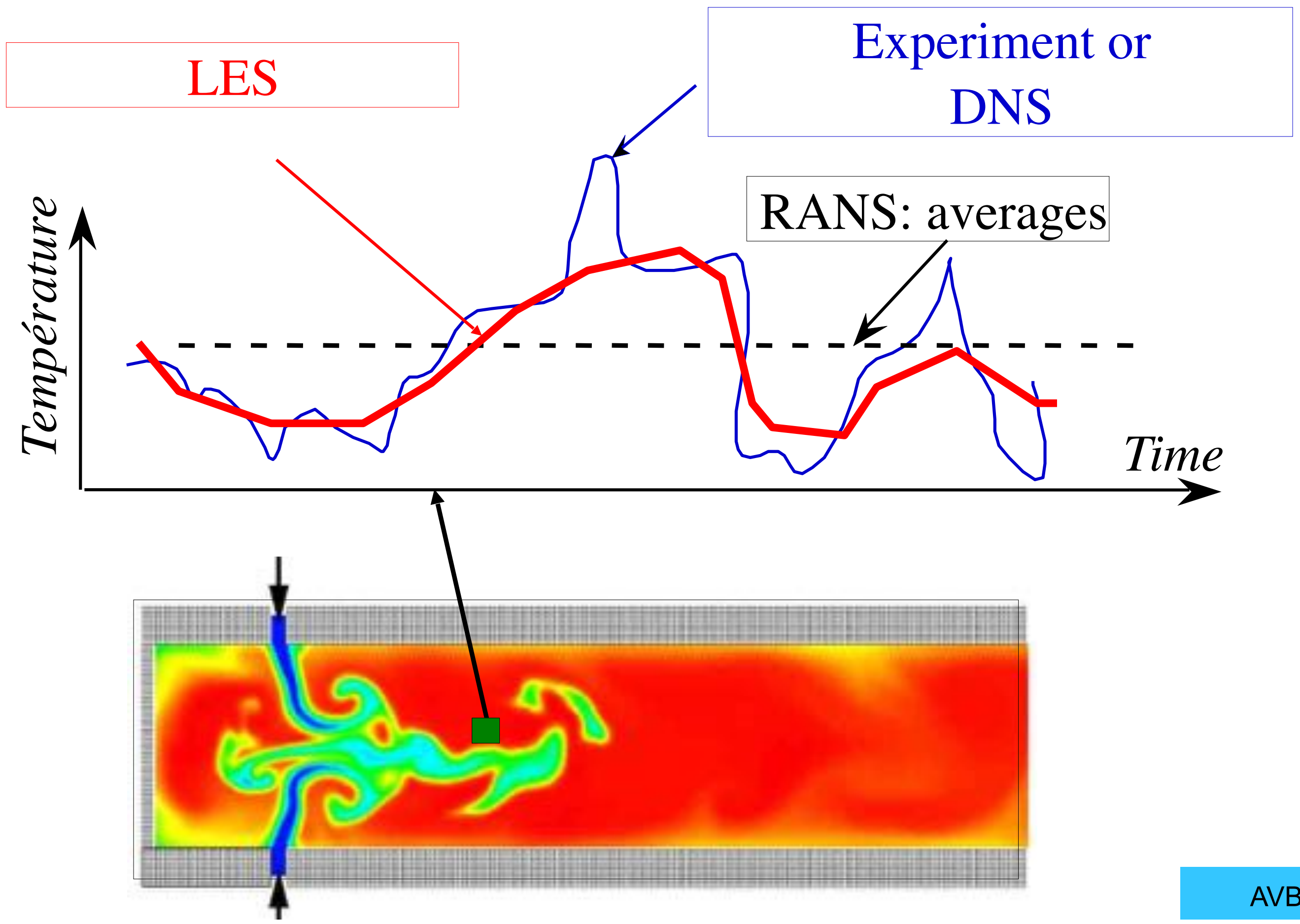


Fully unsteady

Three dimensional

Can we really compute this ?

Methods for turbulent flows CFD:



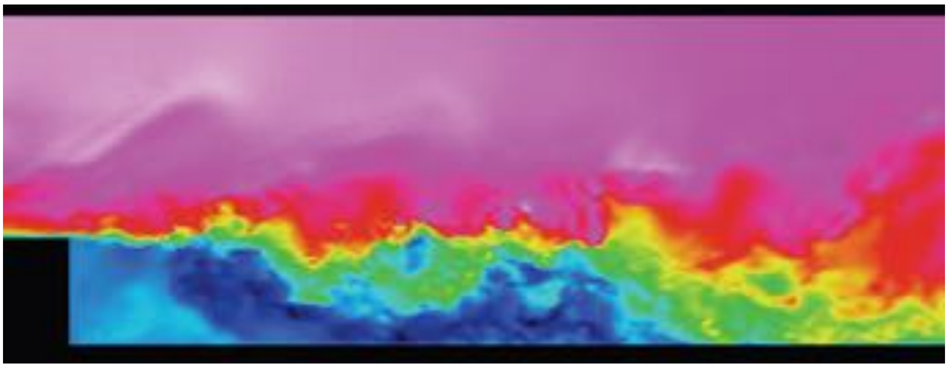
A strong difference between RANS and LES averaging:

- In RANS, averaging is performed over time (or realizations). By definition, RANS variables do not depend on time

- In LES averaging (filtering) is performed locally over space (a small zone around each point). LES variables are time-dependent quantities

Source: Rémy Fransen, 3rd INCA colloquium, ONERA, Toulouse (2011)

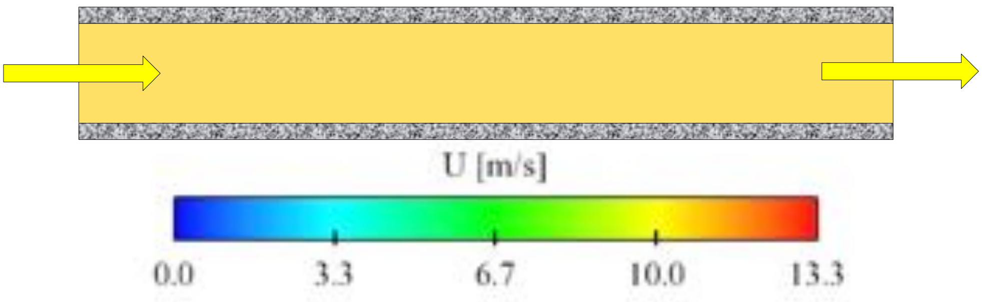
RANS

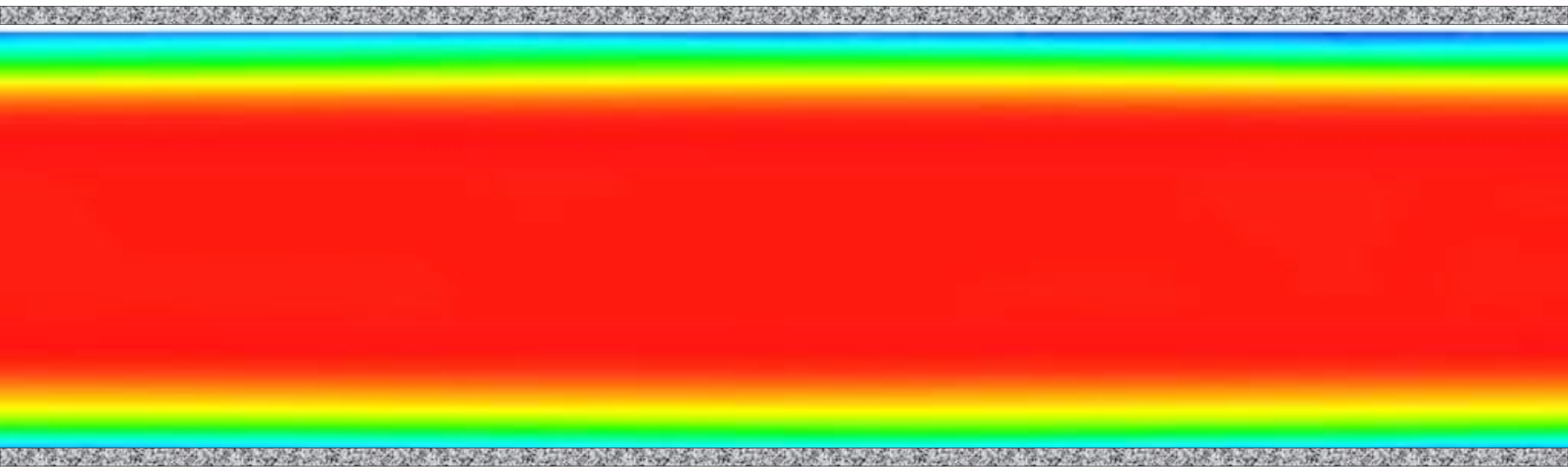


Source: Rémy Fransen, 3rd INCA colloquium, ONERA, Toulouse (2011)

LES

Same duct computed with RANS and then with LES:





t = 0.440 s

NON REACTING

OK, we should not do RANS.

So, what do we do ?

LES.... or even DNS !

However, taking a simulation code from RANS to LES is a big step

Ch. 4 Section 4.8

OUTLINE

- Energy policies and combustion
- Tools to simulate reacting flows
- Turbulent premixed flames and explosions
- Deep learning for turbulence combustion interaction

TURBULENCE SUB GRID MODELS

Apparently, LES and RANS models for turbulent viscosity are not very different:

RANS, time averagec

$$\frac{\partial \overline{\rho}\widetilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u}_i\widetilde{u}_j) + \frac{\partial \overline{p}}{\partial x_j} = \frac{\partial}{\partial x_i}\left[\overline{\tau}_{ij} - \overline{\rho}\boxed{(\widetilde{u_iu_j} - \widetilde{u}_i\widetilde{u}_j)]}\right.$$

$$\boxed{\widetilde{u_iu_j} - \widetilde{u}_i\widetilde{u}_j} = -\nu_t\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial \widetilde{u}_k}{\partial x_k}\right)$$

LES, space filtered

$$\frac{\partial \overline{\rho}\widetilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u}_i\widetilde{u}_j) + \frac{\partial \overline{p}}{\partial x_j} = \frac{\partial}{\partial x_i}\left[\overline{\tau}_{ij} - \overline{\rho}\boxed{(\widetilde{u_iu_j} - \widetilde{u}_i\widetilde{u}_j)]}\right.$$

$$\boxed{\widetilde{u_iu_j} - \widetilde{u}_i\widetilde{u}_j} = -\nu_t\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial \widetilde{u}_k}{\partial x_k}\right)$$

$\nu_t = \mu_t/\rho$ **Turbulent viscosity**

Ch. 4

Seen from the Fortran lines, the only difference between LES and RANS is: turbulent viscosity

RANS, time averaged

Ch. 4 Section 4.7.3

$$\nu_t = c_\mu \frac{k^2}{\epsilon}$$

LES, space filtered

Ch. 4 Section 4.7.3

$$\nu_t = (C_S \Delta)^2 \left|\overline{S}\right| = (C_S \Delta)^2 \left(2\overline{S}_{ij}\overline{S}_{ij}\right)^{1/2}$$

But... in practice:

1/ What makes a code a good LES code is not only changing the expression of the turbulent viscosity (for example replacing the k-eps model by the Smagorinski model)

2/ What is needed usually to write a good LES code is to restart from zero and build a code which is fully « LES compatible » ?

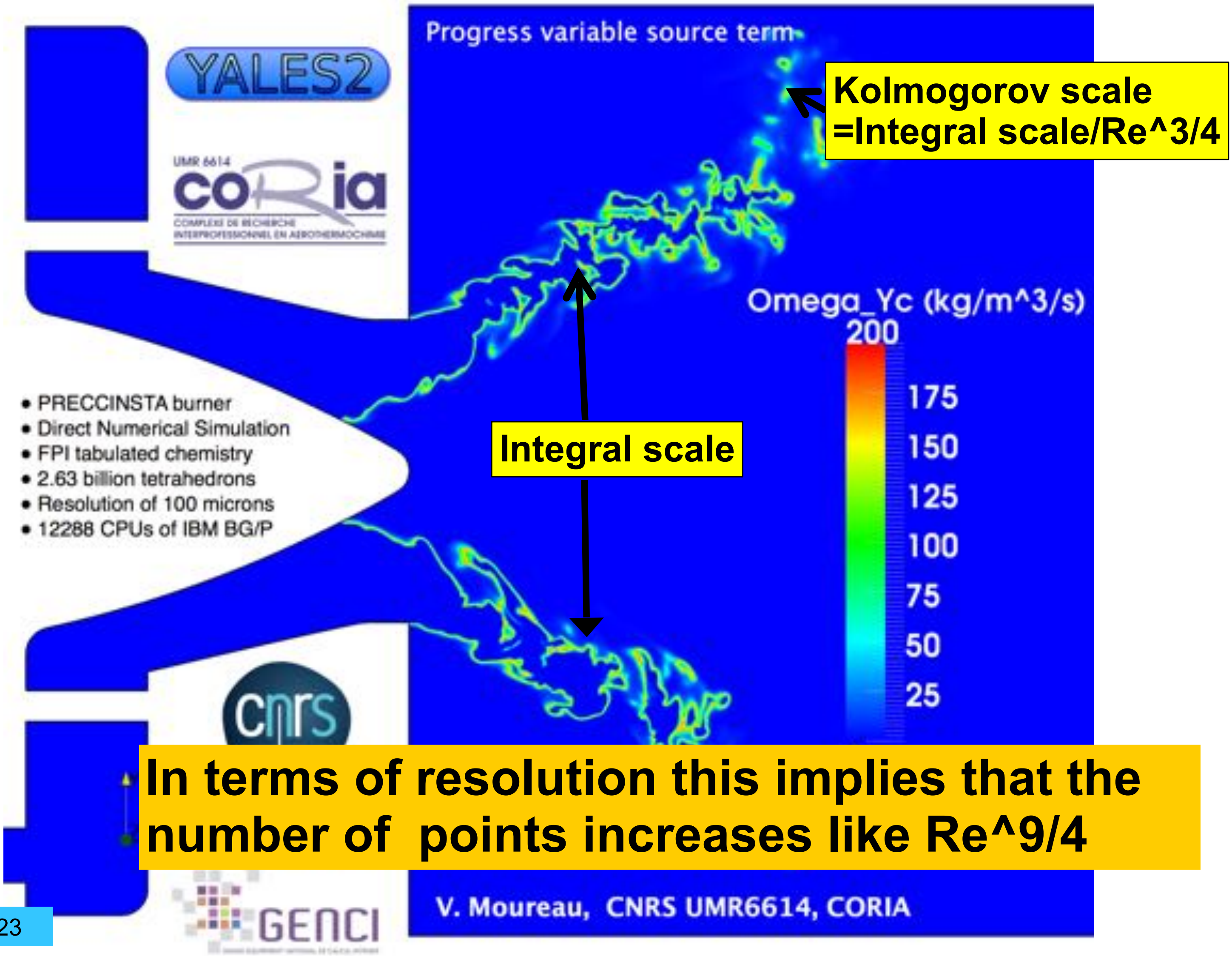
-> Why ?

A more cynical view at the true difference between codes doing LES, RANS and DNS ?

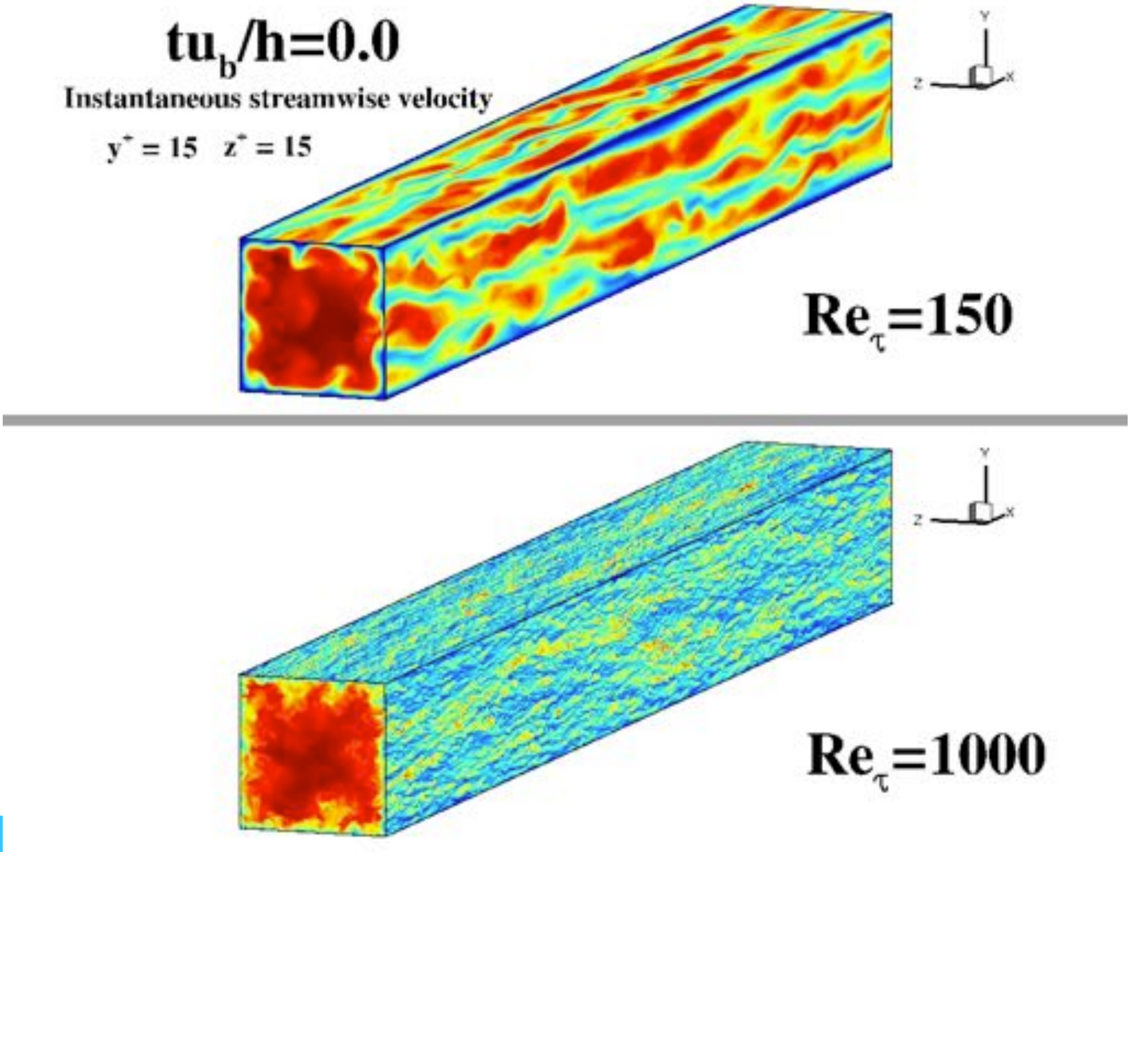
All applications of interest have large Reynolds numbers:

$$\mathrm{Re}(real) = UL/\upsilon$$

A large Reynolds number implies a large difference between large and small spatial scales and therefore a huge number of grid points Which we simply dont have



In terms of resolution this implies that the number of points increases like Re^9/4


tu_b/h=0.0
Instantaneous streamwise velocity
y+ = 15   z+ = 15
Re_τ=150
Re_τ=1000
Y
Z
X

So: we cant resolve all scales associated to large Reynolds numbers and real chambers... This was true in the 60s and it is still true today...

Had to find a solution !

We use two tricks (we call them 'models'):

- **turbulence models: add turbulent viscosity** ν_t
- **dissipative schemes: add numerical viscosity** ν_a

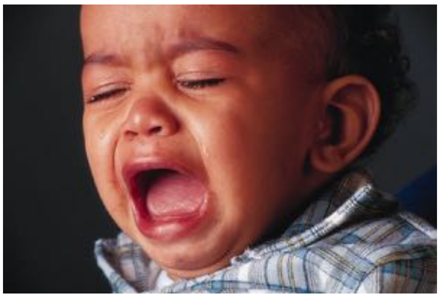
$$\frac{\partial}{\partial t}\rho u_j + \boxed{\frac{\partial}{\partial x_i}\rho u_i u_j} = -\frac{\partial p}{\partial x_j} \boxed{+ \frac{\partial \tau_{ij}}{\partial x_i}}$$

<u>This is the :</u>

- non-linear term

- source of turbulence

- ennemy of all CFD codes



<u>This is the:</u>

- viscous term

- linear term

- damper of turbulence

- friend of all PhD students

So... what is turbulent viscosity ?
It is the easiest solution when the Navier Stokes equations are averaged (in RANS) or filtered (in LES) to model the non linear terms:

$$\frac{\partial}{\partial t}\rho u_j + \frac{\partial}{\partial x_i}\rho u_i u_j = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_i}$$

$$\tau_{ij} = -\frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Filter these equations in space (LES) or average them in time (RANS):

$$f = \widetilde{f} + f''$$

Replace in momentum:

$$\frac{\partial \overline{\rho}\widetilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u}_i\widetilde{u}_j) + \frac{\partial \overline{p}}{\partial x_j} = \frac{\partial}{\partial x_i}\left[\overline{\tau}_{ij} - \overline{\rho}\boxed{(\widetilde{u_i u_j} - \widetilde{u}_i\widetilde{u}_j)}\right]$$

$$\mathcal{T}_{ij} = \boxed{\widetilde{u_i u_j} - \tilde{u}_i\tilde{u}_j} = \boxed{\frac{\delta_{ij}}{3}\mathcal{T}_{kk} - 2\,\nu_t\left(\widetilde{S}_{ij} - \frac{\delta_{ij}}{3}\widetilde{S}_{kk}\right)}$$

The SGS term is modeled using a turbulent viscosity (fully compressible expression here)

Ch. 4 Section 4.7.3

$$\frac{\partial \overline{\rho}\widetilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u}_i\widetilde{u}_j) + \frac{\partial \overline{p}}{\partial x_j} = \frac{\partial}{\partial x_i}\left[\overline{\tau}_{ij} - \boxed{\overline{\rho}\,(\widetilde{u_i u_j} - \widetilde{u}_i\widetilde{u}_j)}\right]$$

$$\frac{\partial \overline{\rho}\widetilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u}_i\widetilde{u}_j) + \frac{\partial \overline{p}}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right]$$

USING A TURBULENT VISCOSITY MODEL IS EQUIVALENT TO ADDING A (COMPLICATED) TURBULENT VISCOSITY TO THE LAMINAR ONE

Ch. 4 Section 4.7.3

Very dangerous model: it transforms a non-linear term (source of turbulence) into a viscous term (which damps turbulence).

1/ Now this term plays a role similar to laminar viscosity

2/ The political interpretation

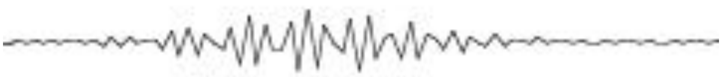
3/ Ultimate reason: this was a good way to get our codes to work !

What is artificial viscosity ?: here, the viscous term is introduced through the numerical scheme

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

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} + u_i \frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x} = o(\Delta x^2)$$

Numerical analysis 101: centered schemes have … problems: they generate wiggles as soon as the resolution is not sufficient



Introduce artificial viscosity:

$$\frac{f_i^{n+1}-f_i^n}{\Delta t}+u_i\frac{f_{i+1}^n-f_{i-1}^n}{2\Delta x}=\boxed{\nu_a\frac{f_{i+1}^n+f_{i-1}^n-2f_i^n}{\Delta x^2}}$$

Makes the scheme more stable and able to handle gradien
However, in practice we are not solving:

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

But:

$$\frac{\partial f}{\partial t}+u\frac{\partial f}{\partial x}=\boxed{\nu_a}\frac{\partial^2 f}{\partial x^2}$$

Upwind schemes are NOT a solution:

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} + u_i \frac{f_i^n - f_{i-1}^n}{\Delta x} = o(\Delta x)$$

Using: $f_{i-1}^n = f_i^n - \Delta x \frac{\partial f}{\partial x} + 1/2 \Delta x^2 \frac{\partial^2 f}{\partial x^2}$

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 1/2(u\Delta x)\frac{\partial^2 f}{\partial x^2} + o(\Delta x^2)$$

An upwind scheme is like a centered scheme with a numerical viscosity equal to *1/2 u* Δ*x*

Turbulent flow solvers combine:

- **turbulent viscosity** ν_t
- **numerical viscosity** ν_a

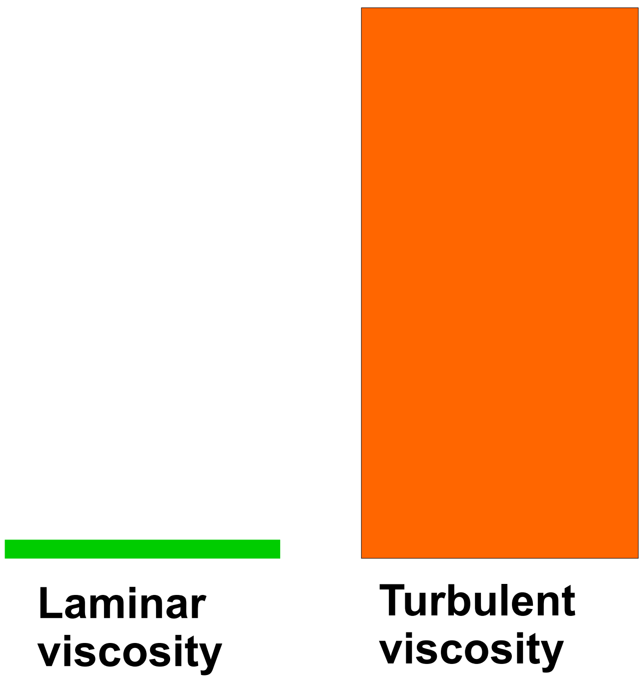
to allow the code to run. But at which price ?
In practice, the Reynolds number seen by the code is :

$$\mathrm{Re}(num) = UL/(\upsilon + \upsilon_t + \upsilon_a)$$

which is much smaller than the Reynolds of the flow.

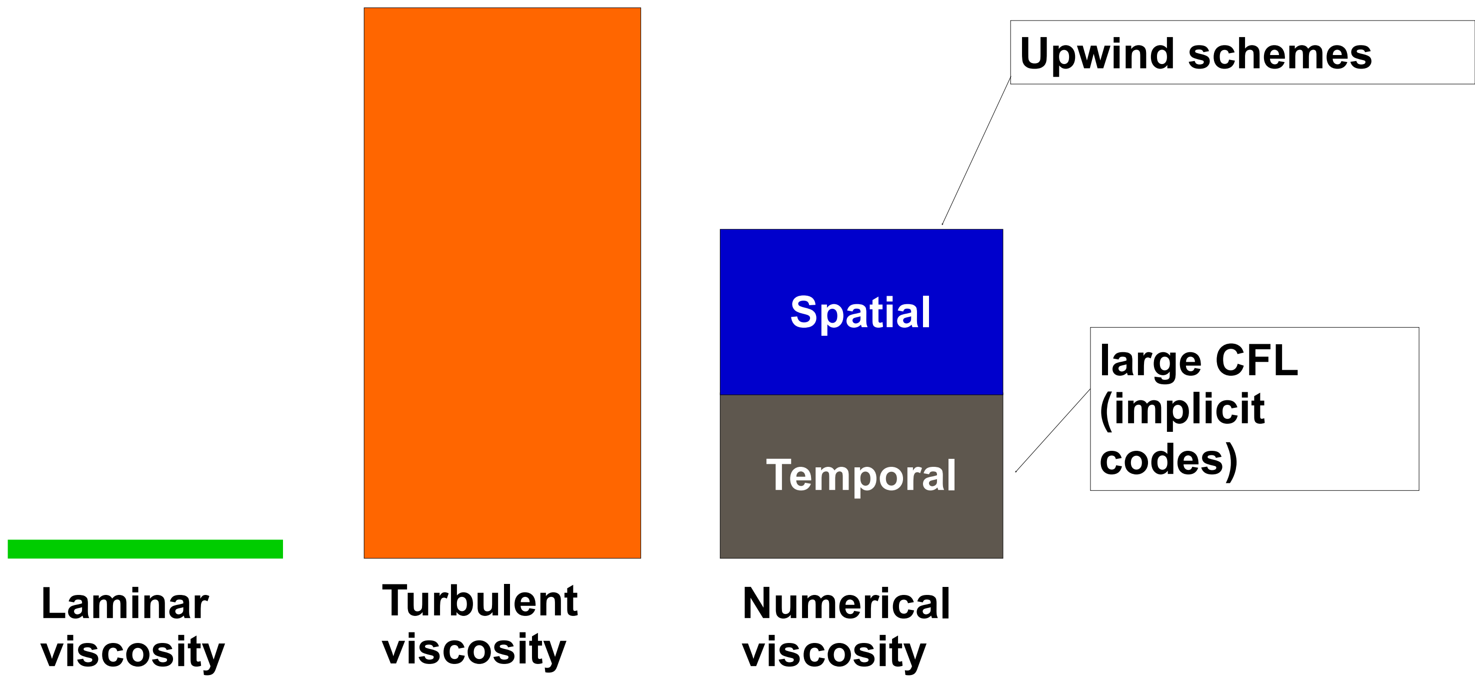
It can even be smaller than the critical Reynolds number to have turbulence in this flow.
==> We are not computing the same flow…: instead of a high Re turbulent flow, we are computing a laminar flow

RANS: turbulent viscosity is very large ==> the Reynolds of the code is so small that the flow is steady (ie laminar)



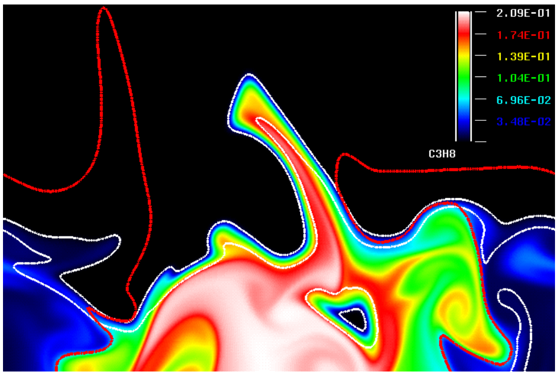
The code sees a laminar flow

RANS: since the flow is so viscous, might as well use numerical viscosity too to make it faster and more robust !



$$\mathrm{Re} = UL/(\upsilon + \upsilon_t + \upsilon_a) << \mathrm{Re}(real)$$

CODE DNS: nothing more than the laminar viscosity



High-order scheme
+
Small CFL

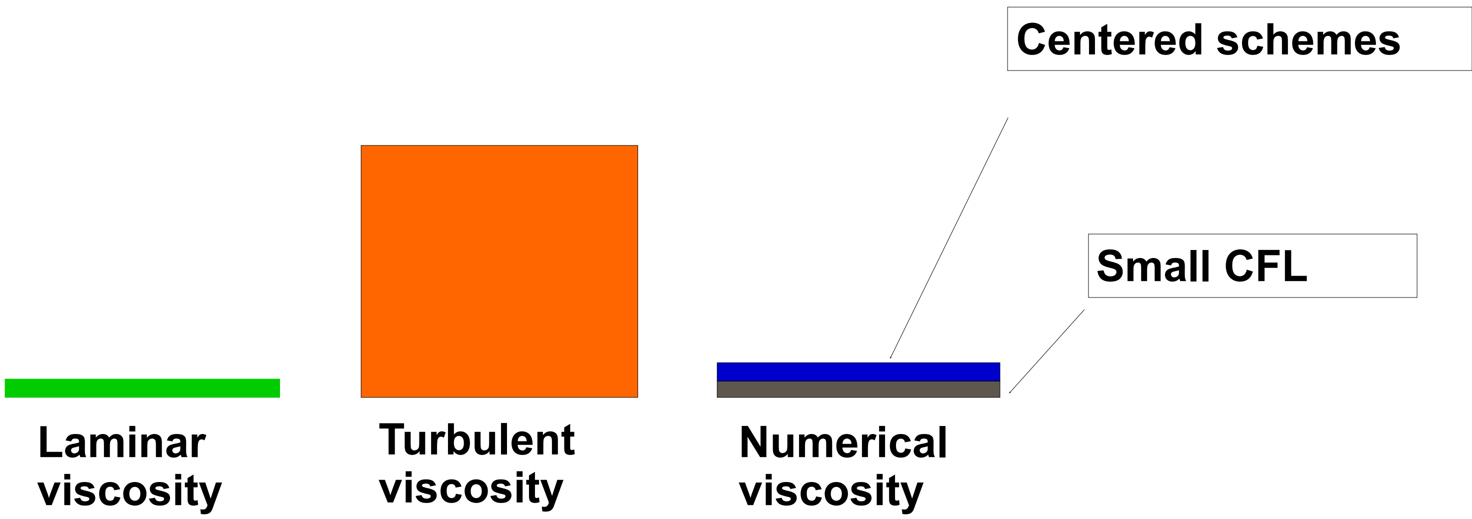
Laminar viscosity

Turbulent viscosity

Numerical viscosity

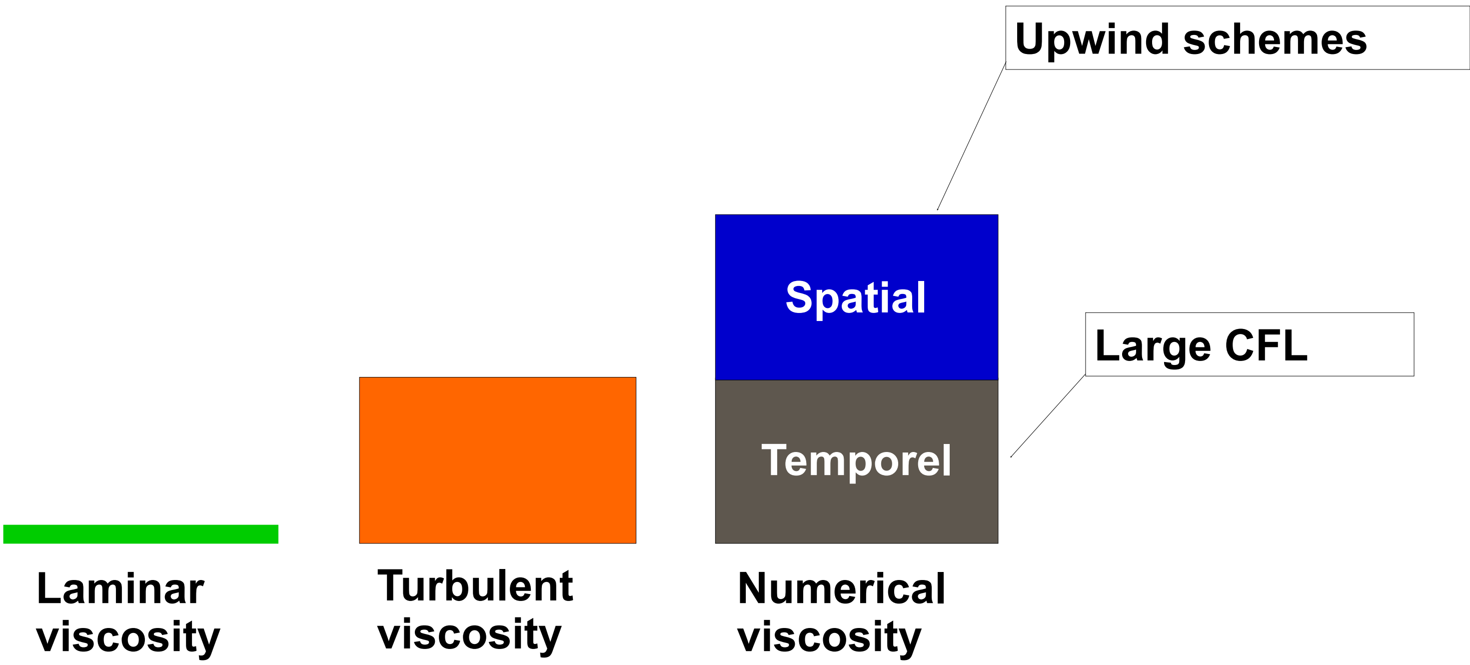
$$\mathrm{Re}(num) = UL/(\upsilon + \upsilon_t + \upsilon_a) = UL/\upsilon$$

A good LES code: turbulent viscosity reduced and limited numerical viscosity ==> Reynolds turbulent smaller than the true one but large enough for the flow to be turbulent



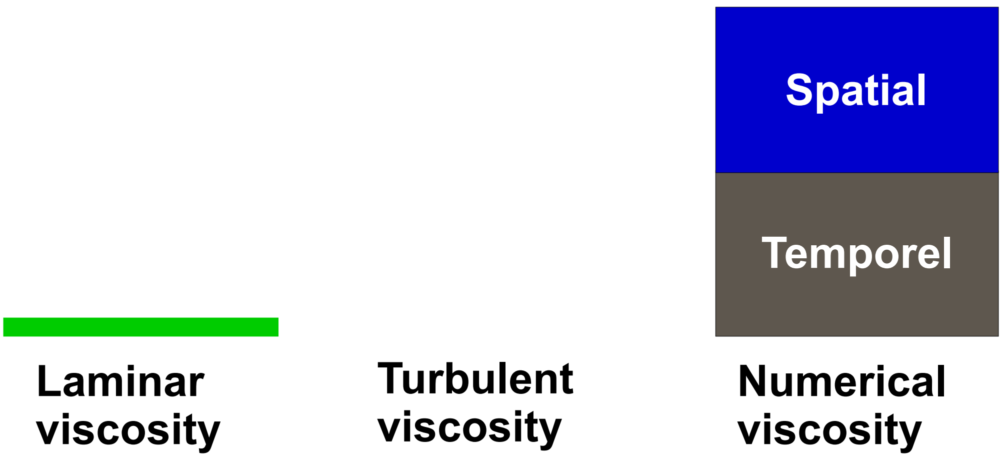
$$\mathrm{Re} = UL/(\upsilon + \upsilon_t + \upsilon_a) > \mathrm{Re}(crit)$$

DANGERS of RANS codes used for LES:



$$\mathrm{Re}(num) = UL/(\upsilon + \upsilon_t + \upsilon_a) \text{ too small}$$

Classical test: remove the turbulent viscosity



CONCLUSION:

A good LES needs:

- ➢ **high order schemes**
- ➢ **small time steps**

(Otherwise it is LESWE: Large Eddy Simulation Without Eddy…)

- ➢ **This will require important CPU time**
- ➢ **THIS IS IMPOSSIBLE IF WE DO NOT USE MASSIVELY PARALLEL MACHINES**

Even if we have the CPU power, is it easy to do ? Actually NO ! Computing waves (vortices or acoustic waves or entropy waves) is tough.

LES: it is all about waves !

LES must propagate:

- **vortices,**
- **acoustic waves**
- **chemical species.**

This impacts our choices for numerical techniques

‘Not all LES codes are equal’ (Stanford motto)

DISPERSION / DISSIPATION

IN THE REAL WORLD:

- **A medium is dispersive if the speed at which waves propagate depends on their frequency.**
- **A medium is dissipative if waves are dissipated when they propagate.**

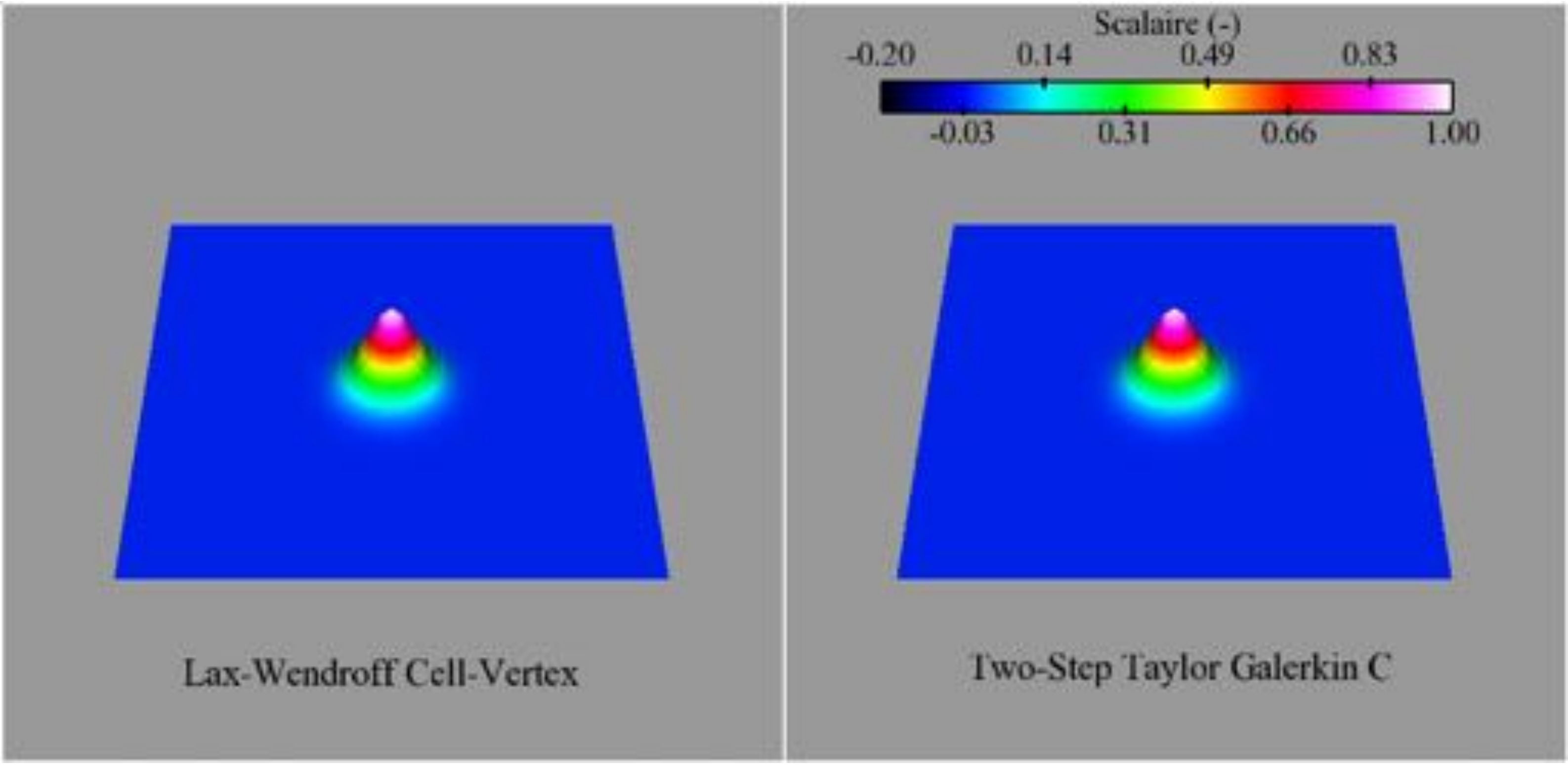
Example: Air is not dispersive for sound waves. But it is dissipative for high frequency waves.

IN THE NUMERICAL WORLD:

Building a numerical technique which respects the dispersive and dissipative properties of gases is almost impossible.
For LES, this is bad news.

Example: convecting a scalar 'bump' in homogeneous flow with two methods:

- **Lax Wendroff (2nd order)**
- **TTGC (3rd/4th order)**



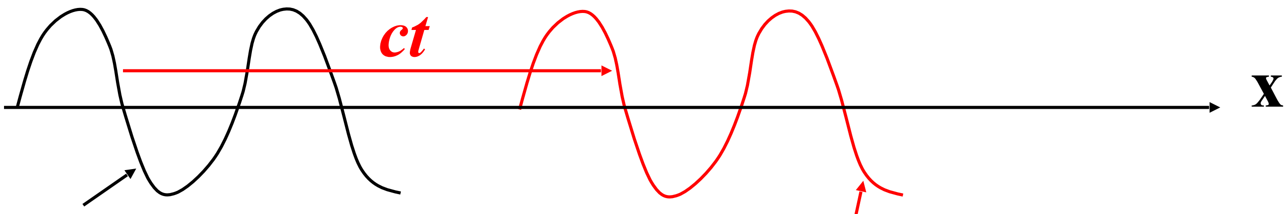
Can we study these questions without writing a code ?

Yes !... consider the simplest case of one-dimensional convection equation at speed c:

$$\frac{\partial}{\partial t}u + c\frac{\partial}{\partial x}u = 0$$

For this equation, we can derive analytically what the results of a given scheme with perfect time advancement would be. This equation is neither dispersive nor dissipative by nature: all signals are transported at speed *c* without any modification

The exact solution for this wave problem is a convection at speed c:



$$u(x,t=0) = v(0)\ \exp\left[2j\pi\omega x\right]$$

u(x,t) exact

$$\boxed{u(x,t) \quad = \quad u(x-ct,\ \ t=0) \ = \ v(0)\ \exp\left[2j\pi\omega(x-ct)\right]}$$

Being able to predict this convection speed is crucial for acoustics but also for turbulence (to convect vortices or entropy waves).

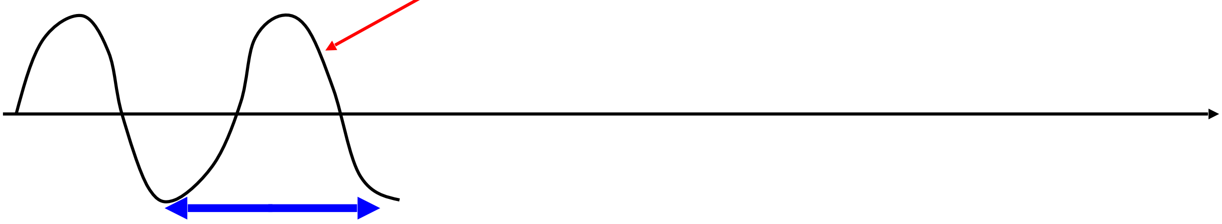
What happens in codes ? space is discretized…
Take the simplest finite difference example:

Discretize x axis: $x = i\Delta x$

Assume that u is a sinusoidal function of space (pulsation ω):

$$u(i\Delta x, t) = u_i(t) = v(t)\exp(2\pi j\ \omega\ i\Delta x)$$

$j^2=-1$



$$\lambda = 1/\omega$$

What does a second order code do ? Suppose we discretize this equation in space on a grid of spacing Δx and assume we have perfect time advancement:

$$\frac{\partial}{\partial t}u_i + c\frac{u_{i+1}-u_{i-1}}{2\Delta x}=0 \qquad \textbf{(1)}$$

For sinusoidal wave propagation: $u_i = v(t)\exp(2\pi j\,\omega\, i\Delta x)$

Replacing *u* by *v(t)* in Eq. (1) leads to:

$$\frac{\partial}{\partial t}v = -cjv\,\sin(2\pi\omega\Delta x)/\Delta x$$

Or: $$v(t) = v(0)\exp\left(-cj\,\frac{\sin(2\pi\omega\Delta x)}{\Delta x}t\right)$$

The numerical solution for this problem is:

$$\boxed{u(x,t) = v(0)\exp\left[2j\pi\omega\left(x-c\frac{\sin(2\pi\omega\Delta x)}{2\pi\omega\Delta x}t\right)\right]}$$

Comparing the exact and the numerical solution:

Exact:
$$u(x,t) = v(0) \exp\left[2j\pi\omega(x-ct)\right]$$

Numerical:
$$u(x,t) = v(0) \exp\left[2j\pi\omega\left(x-c\boxed{\frac{\sin(2\pi\omega\Delta x)}{2\pi\omega\Delta x}}t\right)\right]$$

The numerical scheme is dispersive: the speed is not right

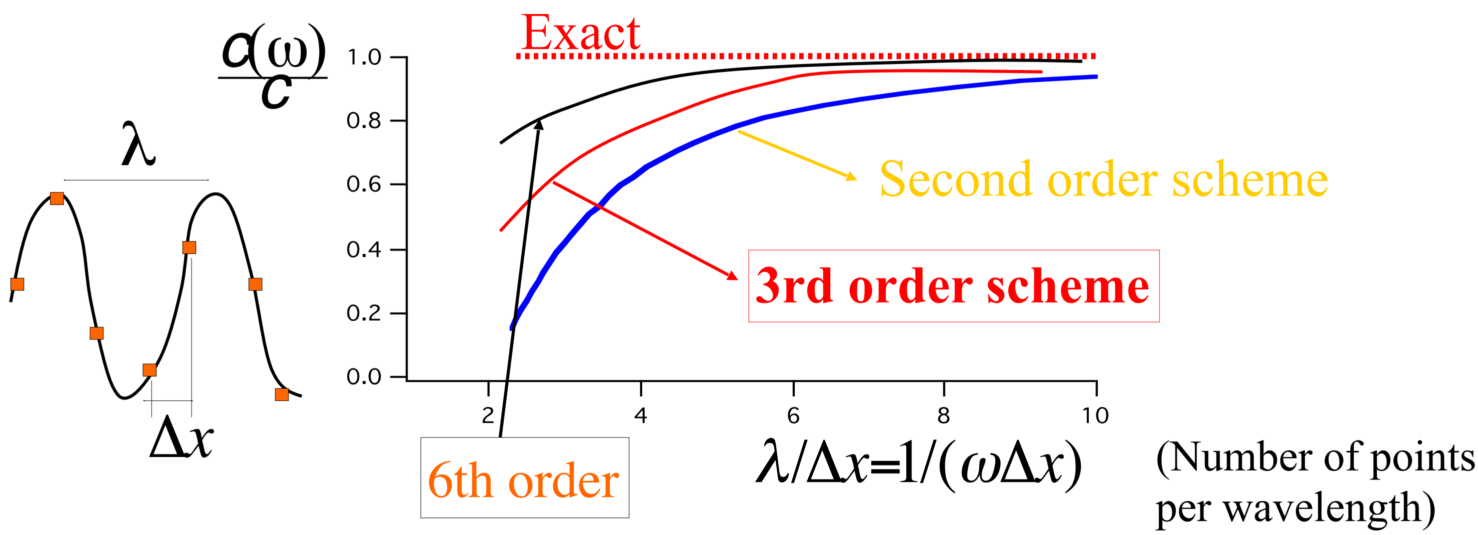
Comparing the speeds:

Exact: $u(x,t) = v(0) \exp\left[2j\pi\omega(x-ct)\right]$

Numerical: $u(x,t) = v(0) \exp\left[2j\pi\omega\left(x-c\frac{\sin(2\pi\omega\Delta x)}{2\pi\omega\Delta x}t\right)\right]$

shows that the numerical scheme makes the flow 'dispersive'; different wavelengths ω are propagated at different speeds c(ω):

$$\boxed{\frac{c(\omega)}{c} = \frac{\sin(2\pi\omega\Delta x)}{2\pi\omega\Delta x}}$$



This is not good news for second-order schemes: they do not propagate waves at the right speed as soon as the resolution (ie the number of points per wavelength $\lambda/\Delta x$) is not very high.

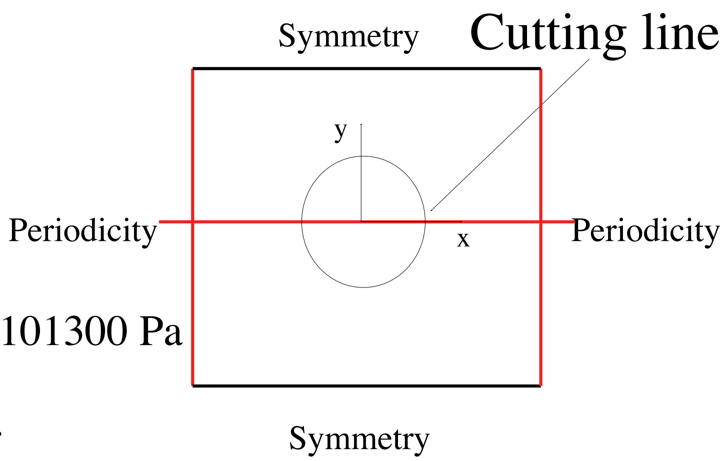
Higher order schemes do MUCH better.

2D Vortex convection

Convecting vortices is the basic feature of LES

$$U_x = 100\, m/s \qquad p - p_0 = -\frac{\rho\Gamma^2}{2R_c^2} e^{-\frac{x^2+y^2}{R_c^2}}$$

$$v = \frac{\Gamma}{R_c^2} x e^{-\frac{x^2+y^2}{2R_c^2}} \qquad u = -\frac{\Gamma}{R_c^2} y e^{-\frac{x^2+y^2}{2R_c^2}}$$

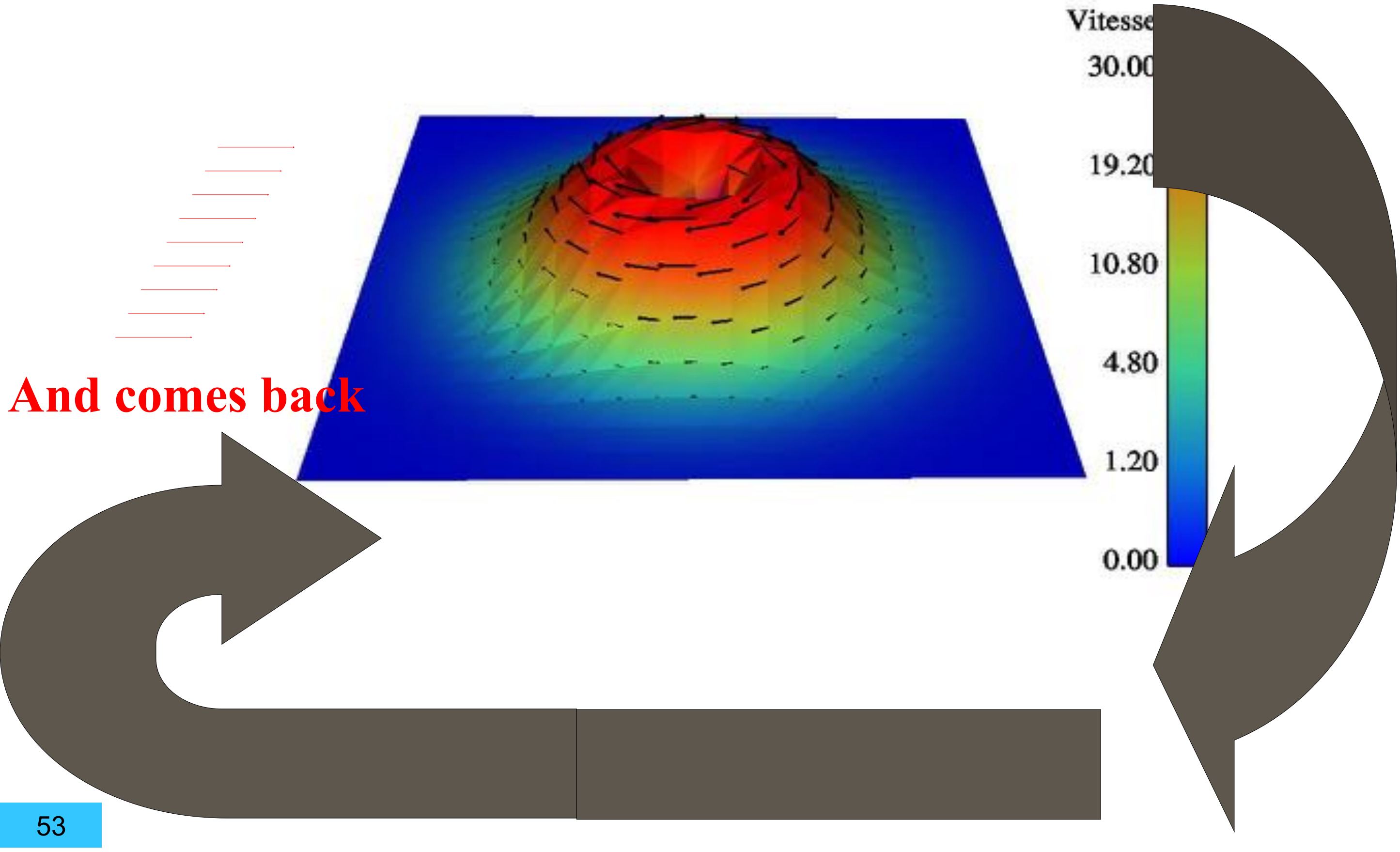


- $\Gamma = 1$ m/s^2 ; $R_c = 19.45*10^{-3}$ m ; $p_0 = 101300$ Pa
- 2D structured mesh 30X30 elements.
- Analytical solution
- Tested LW (2nd order) and TTGC (3rd order) numerical scheme.
- Acoustic CFL = 0.7
- Tested also other codes. CFX 5.7 using a 2nd order centred finite volume scheme. Fluent. Openfoam, etc

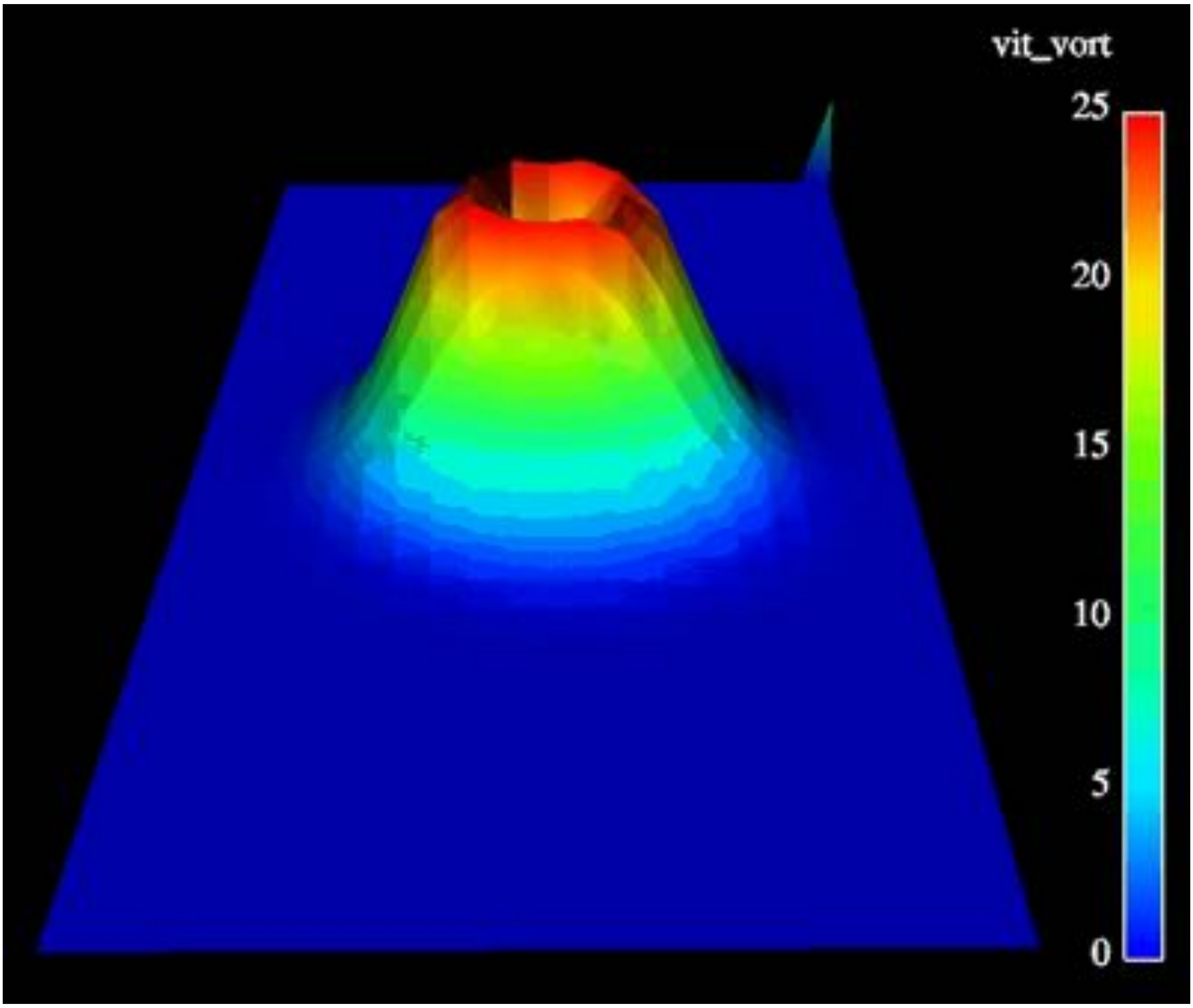
Moureau et al, JCP 2005

Periodic on all sides

Vortex leaves



And comes back


vit_vort
25
20
15
10
5
0

eLearning at CERFACS

http://elearning.cerfacs.fr/numerical/benchmarks/vortex2d/index.php

Hfactor ERC-CNRS ERC Feedback GoogleMap YAHOO CERFACSwww WEBMAIL ASSURANCES/BANQUES TV METEO ORANGE VTT

eLearning at CERFACS

CERFACS e-Learning@cerfacs

Numerical methods for CFD - *Benchmarks*

The CERFACS CO-VO test (COnvection of a VOrtex) for DNS and LES codes

Abstract

This site is a platform open to all groups interested in comparing their DNS/LES codes on a very simple case: the convection of a vortex on a mean, constant speed flow. It represents the simplest prototype of what high fidelity codes must do in DNS or LES: convect vortices over long distances at the right speed and the right amplitude. The computation is performed without viscosity and the expected solution is simply the initial vortex convected without deformation. The computation is performed in a periodic box in which the vortex turns for 10, 20, 30 and 40 turn over times. Comparing the solution at these instants with the initial solution is an excellent qualification of the solvers accuracy.

A small grid (80 by 80) is used for accuracy while a three dimensional grid is used to measure speed and efficiency on parallel machines.

CERFACS has tested some of the codes available in France and you are welcome to look at results but also to repeat the same tests and send us the results. We will incorporate them in the web site. The document referenced below provides all information to repeat the test which is extremely fast and simple.

Associated ressources

- Case description and results (fr)
- Test case description
- ...ults of the test case

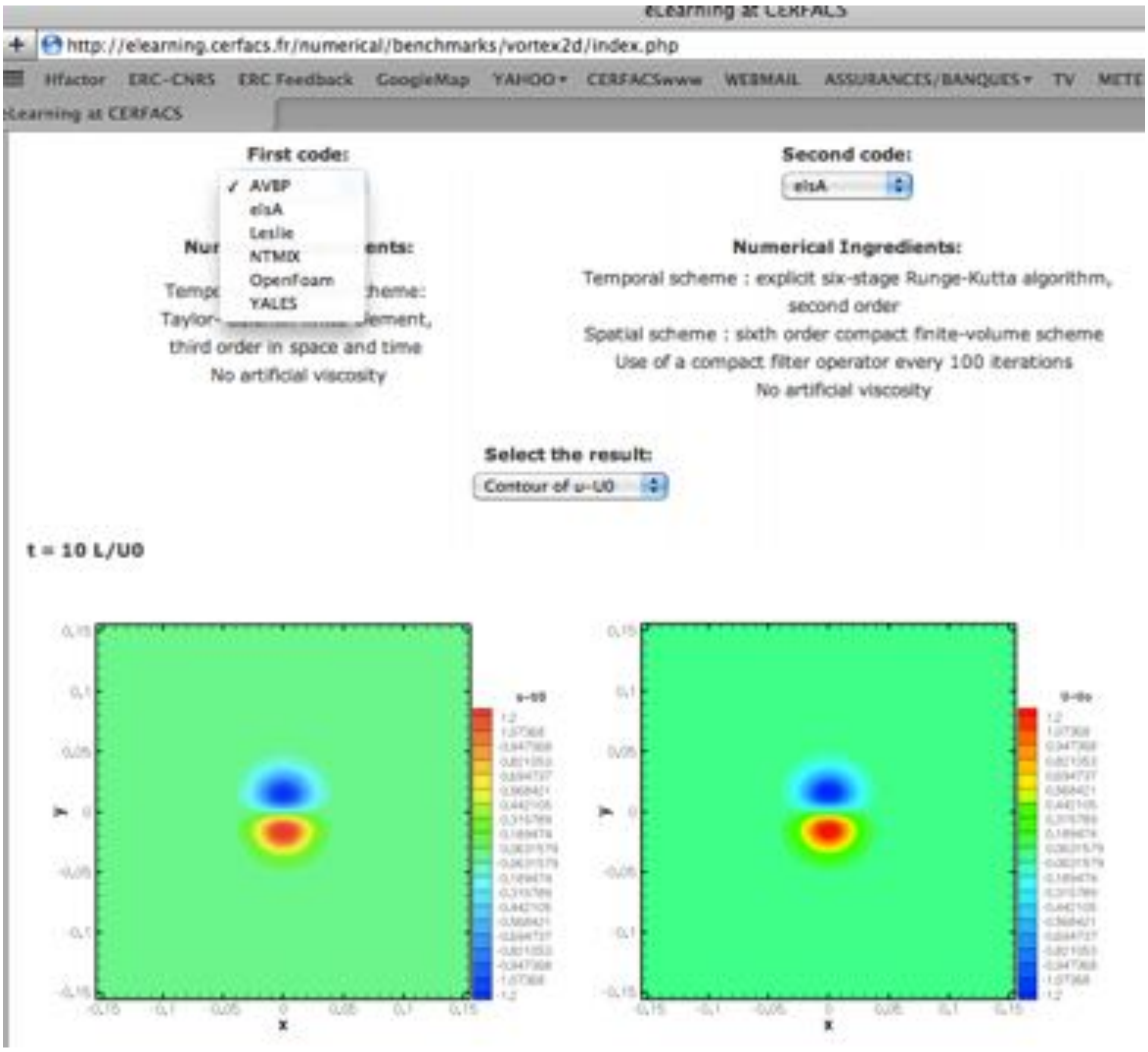
Home

Topics

Combustion
Turbomachine
Environmenta
Numerical met
Advices for ph
Miscellaneous
Coupling

Formations

Computation c flame tempera
Stability, dispe dissipation
Mesh and discr
Unstable mode combustor


eLearning at CERFACS
http://elearning.cerfacs.fr/numerical/benchmarks/vortex2d/index.php
First code:
AVBP
elsA
Leslie
NTMIX
OpenFoam
YALES
Second code:
elsA
Numerical Ingredients:
Temporal scheme : explicit six-stage Runge-Kutta algorithm, second order
Spatial scheme : sixth order compact finite-volume scheme
Use of a compact filter operator every 100 iterations
No artificial viscosity
third order in space and time
No artificial viscosity
Select the result:
Contour of u-U0
t = 10 L/U0

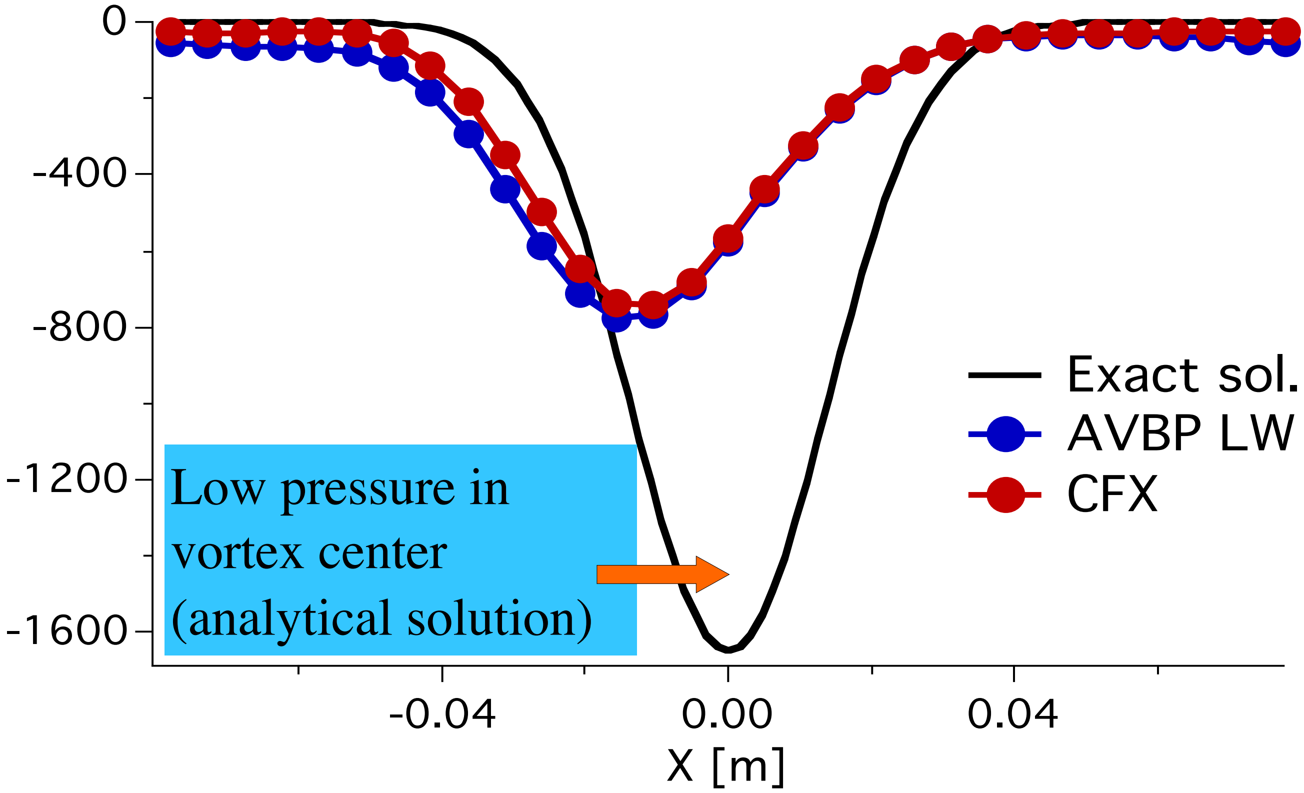
Comparing three schemes:

- 2nd order Lax Wendroff in AVBP (code CERFACS)
- 2nd order CFX (or Fluent or Openfoam)
- 3rd order TTGC in AVBP (Oxford/CERFACS: Colin and Rudgyard, *J. Comp. Phys.* 162 (2000)).

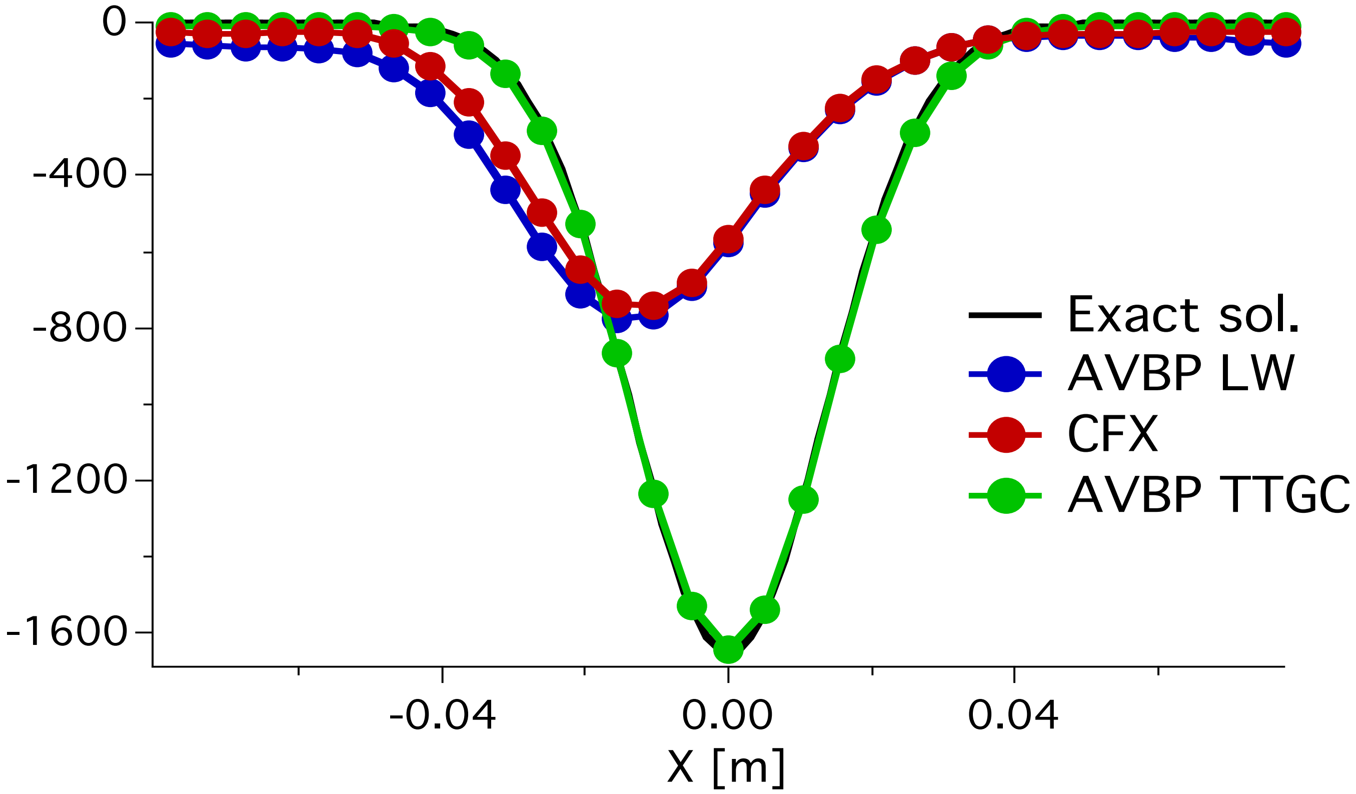
Results after three turn over times

Time step is set by CFL = c $\Delta t/\Delta x$ = 0.7

Relative pressure [Pa]



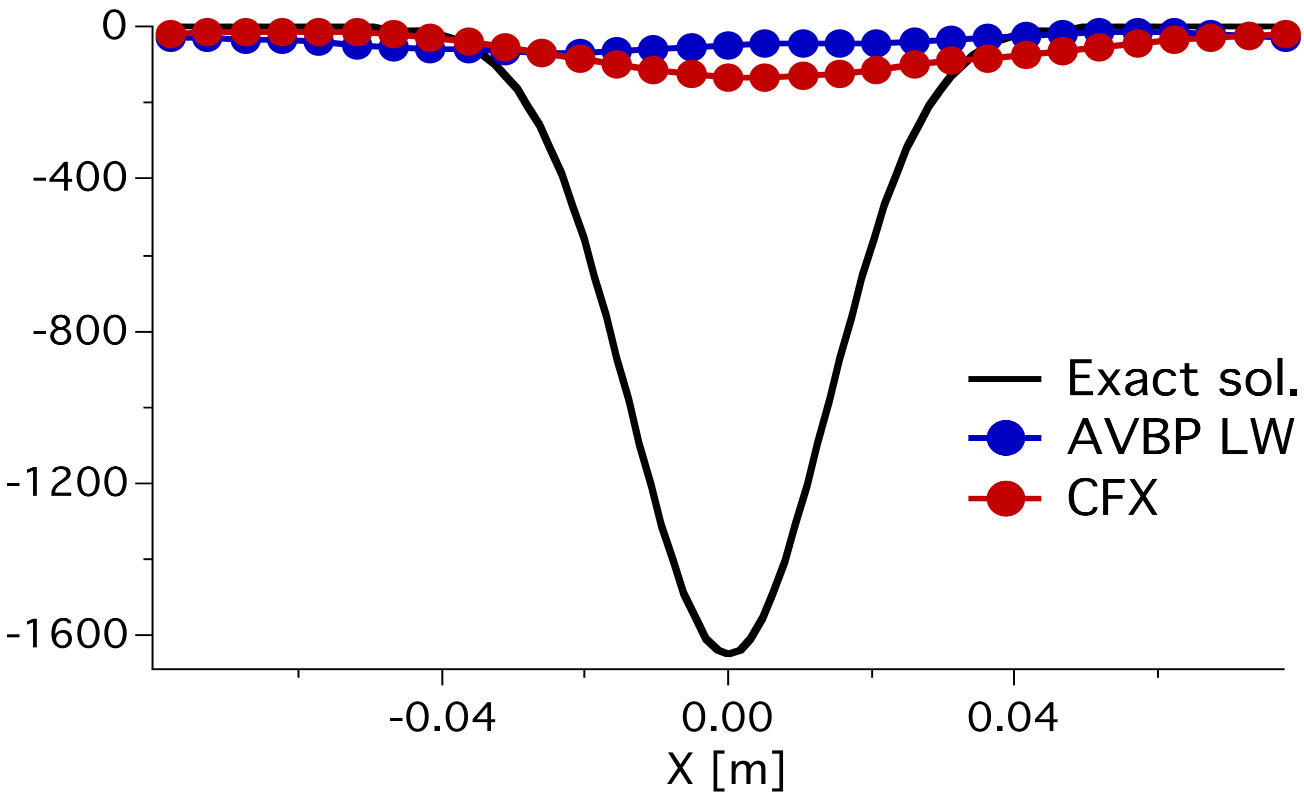
Relative pressure [Pa]



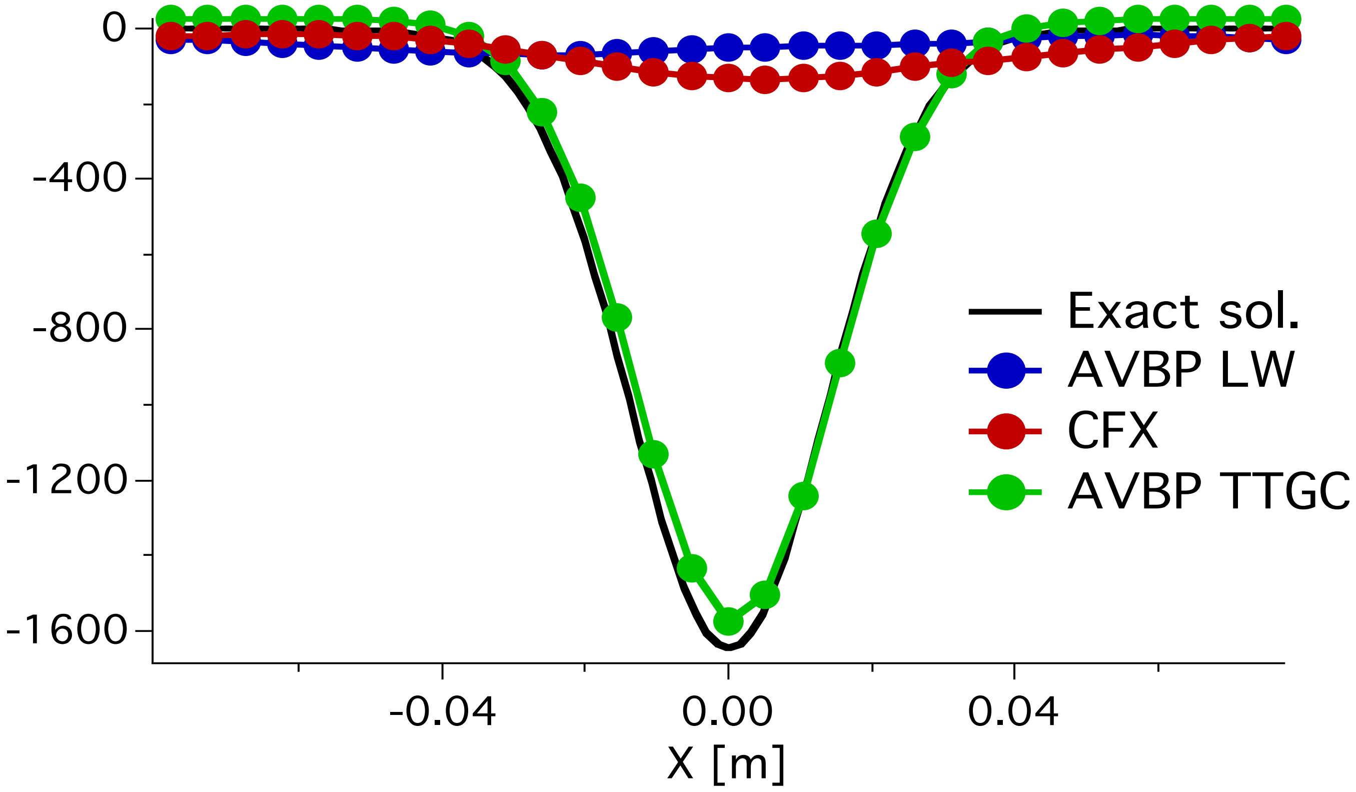
Results after ten turn over times

CFL = 0.7

Relative pressure [Pa]



Relative pressure [Pa]



So high order, explicit schemes are better
==> well known : they are a *MUST* for DNS codes

In the DNS community (which uses structured grids):
- Spectral schemes (not many in combustion)
- Pseudo spectral schemes
- Finite differences: 6th, 8th, 10th order in space

BUT IT IS NOT SIMPLE TO CONSTRUCT AN EXPLICIT HIGH-ORDER SCHEME ON UNSTRUCTURED MESHES !
1st order: easy
2nd order: OK
3rd order: much more difficult
4th order: ouch !

OUTLINE

- Energy policies and combustion
- Tools to simulate reacting flows
- Turbulent premixed flames and explosions
- Deep learning for turbulence combustion interaction

Turbulent premixed flames: What is our main problem ?

- SIZE RATIO S: the size of the domain / the flame thickness
- S is systematically small in DNS
- S is large in real atmospheric flames: 10 cm/ 0.5mm= 200

And S is huge in two cases:

- High pressure flames (aerospace applications): because the flame thickness is small
- Large domains (explosions) because the domain size is large
- A 30 cm / 60 bar aeronautical chamber and a 20 m / 1 bar explosion raise the same modeling difficulties

This issue is not limited to 'flames'

- For all chemical reactions in turbulent flows, the upscaling problem is a major one
- Validations and calibration of simulation tools are often performed on small scale systems, at atmospheric pressure
- In other words, models which are working in small scale, low pressure devices may fail miserably in real, large, high pressure systems.

INTRODUCTION: EXPLOSIONS

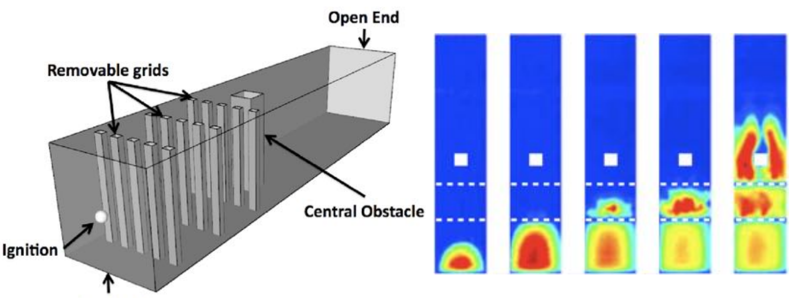

Buncefield, UK

IGNITION IN BUILDINGS

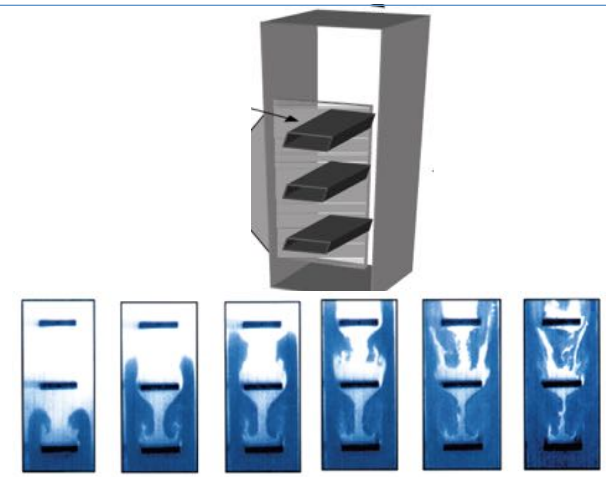
- **When there is a gas leak in a building (for example an offshore platform), the consequences can be dramatic**

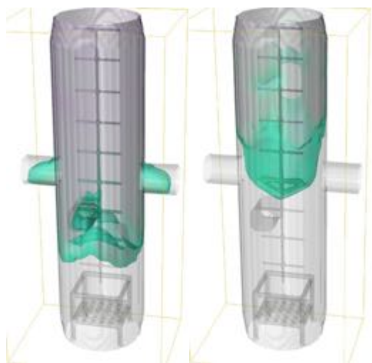
When turbulent flames become too fast:

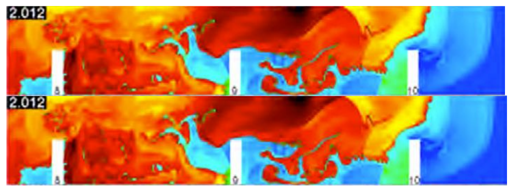
Explosions are studied in venting chambers



Kent, *J. et al. 5th Asia-Pacific Conf. Combust. , Adelaide, Australia (2005)*

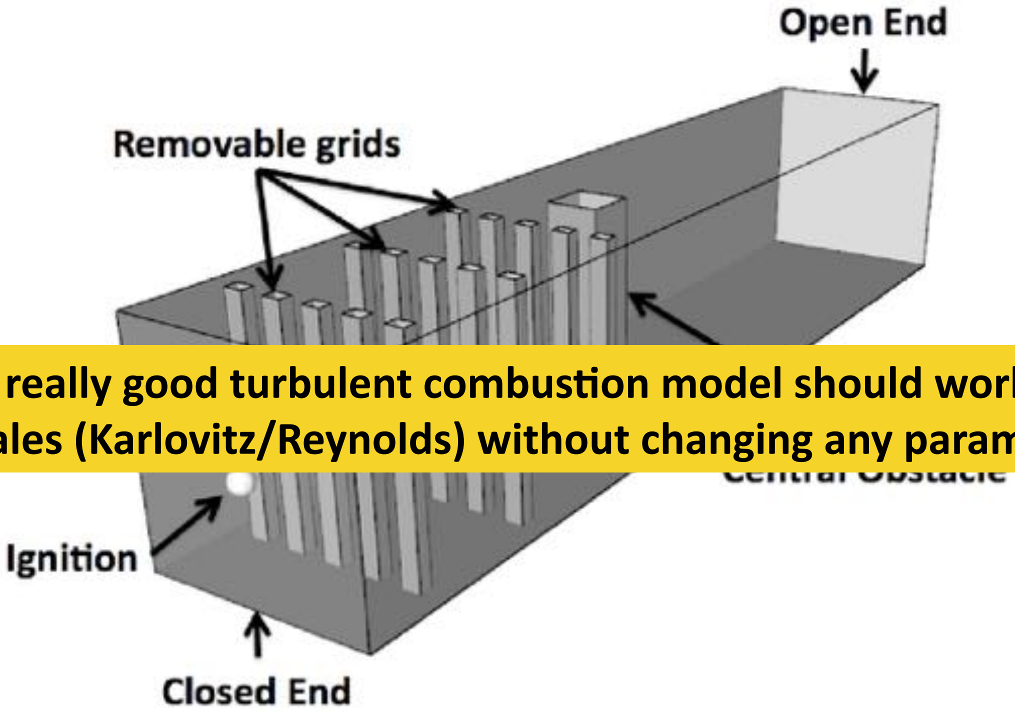

Patel, S. *et al.* *Proc. Combust. Inst. (2002)*


Makarov, D. *et al.* Int. *Journal Hydrogen Energy. (2010)*



Dorofeev, S.B. *Proc. Combust. Inst. (2011)*

Venting chambers allow a variation in scales which is unseen in other systems



A really good turbulent combustion model should work at all scales (Karlovitz/Reynolds) without changing any parameter...

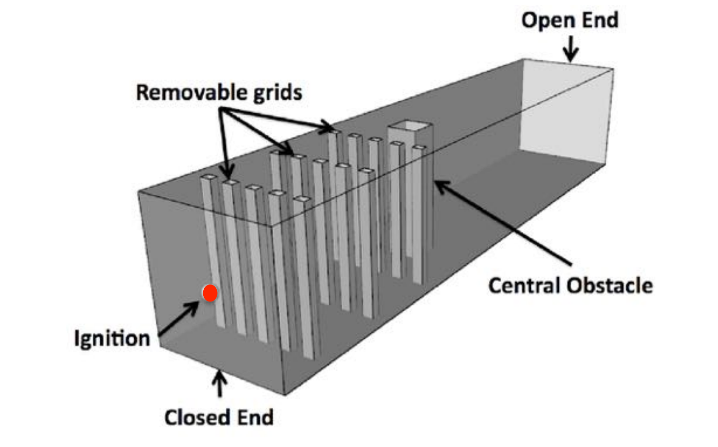
A.R. Masri et al., Industrial & Engineering Chemistry Research, 2012

O. Vermorel, P. Quillatre and T. Poinsot and Ph. Ricoux. LES of explosions in venting chamber: a test case for premixed turbulent combustion models. Comb. Flame. 2017, 183, 207-224.

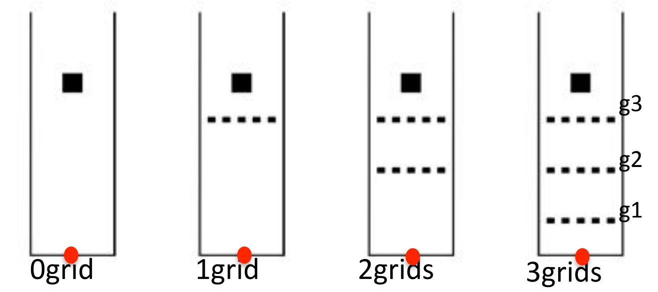
The initial Sydney bomb: 25 cm long

Sydney Explosion Chamber [1]

- Box :
  - 0.05 x 0.05 x 0.25 m^3 (small-scale)
  - 0.3 x 0.3 x 1.5 m^3 (medium-scale)
- Fully filled with Fuel/Air mixture
- Fuels : C_3H_8 or CH_4 or H_2
- One central square obstruction
- 3 turbulence generating grids (removable)
- Laser ignition at the closed end of the chamber in the initially quiet mixture

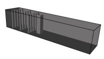
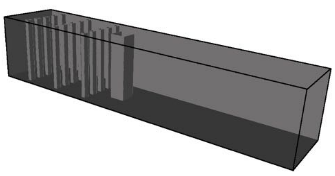
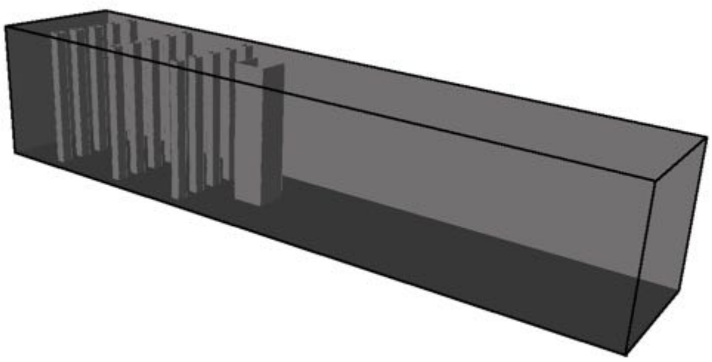


Different configurations studied:

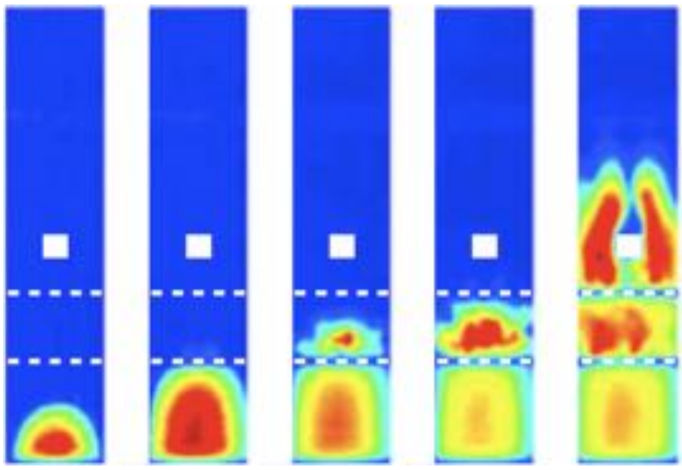


[1] Masri, A.R. Al-Harbi, A. Meares, S. and Ibrahim, S ."A Comparative Study of Turbulent Premixed Flames Propagating Past Repeated Obstacles", *Industrial & Engineering Chemistry Research (2012)*

Same setup - three sizes:

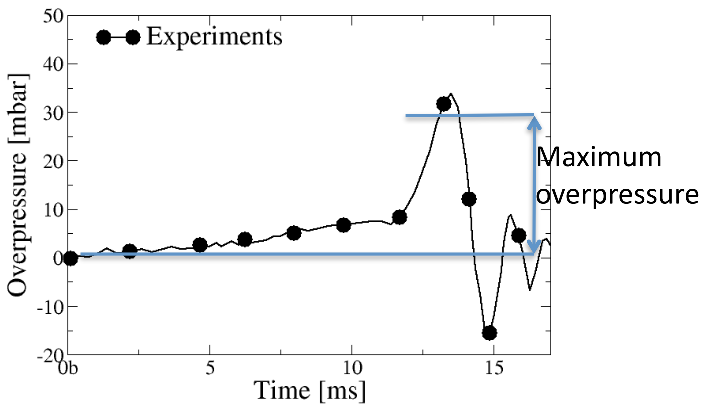
	SCALE	VOLUME	
	X 1	X 1	Masri setup University of Sydney
	X 6	X 216	Scaled-up reproduction of Masri setup (x6) - 1,5m Experiments by GEXCON
	X 24	X 13824	Scaled-up reproduction of Masri setup (x24) - 6,1m Experiments by GEXCON

Experimental images of flame propagation [2]:



8ms 10ms 11ms 11,5ms 12ms

Overpressure generated P-P0 [mbar]



[2] Gubba, S.R. *et al.*, *Combust. Sci. Tech. (2008).*

Complex problem mixing:

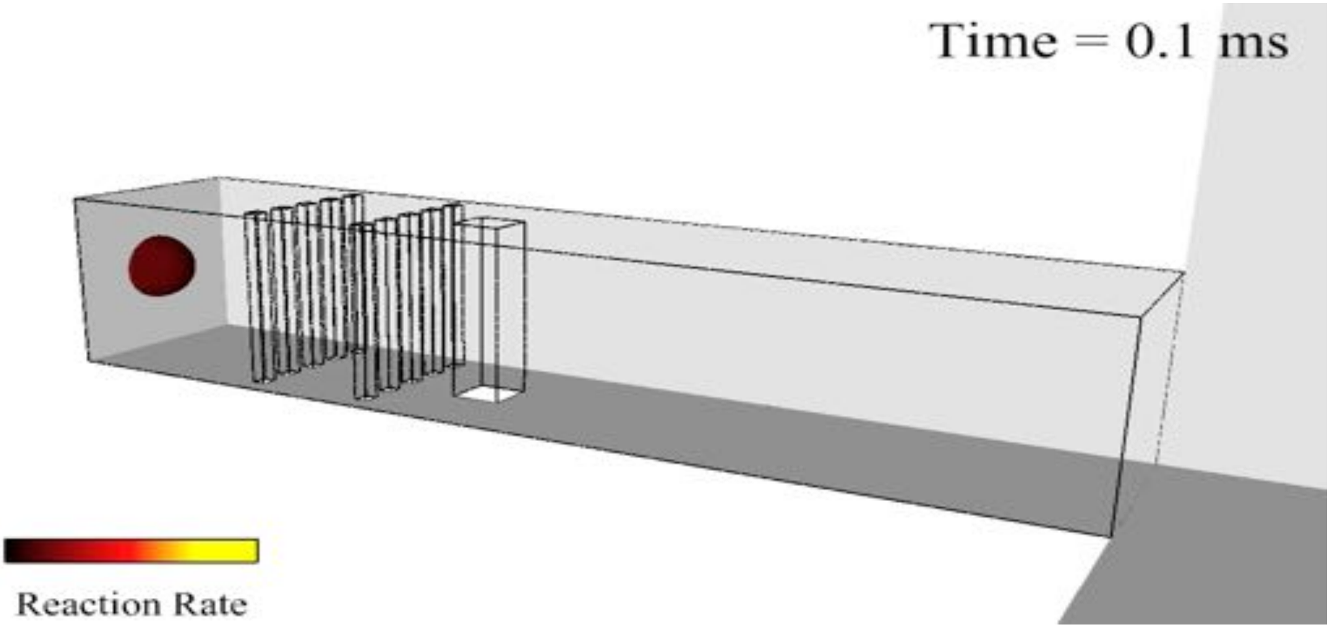
- Ignition
- Laminar phase propagation
- Transition to turbulence
- Turbulent propagation
- Relaminarisation

Comparison with experimental data:

- Flame structure
- Flame position
- **Maximum overpressure**
- **Influence of adding/removing grids**
- **Influence of fuel**

Results – Small Scale Chamber

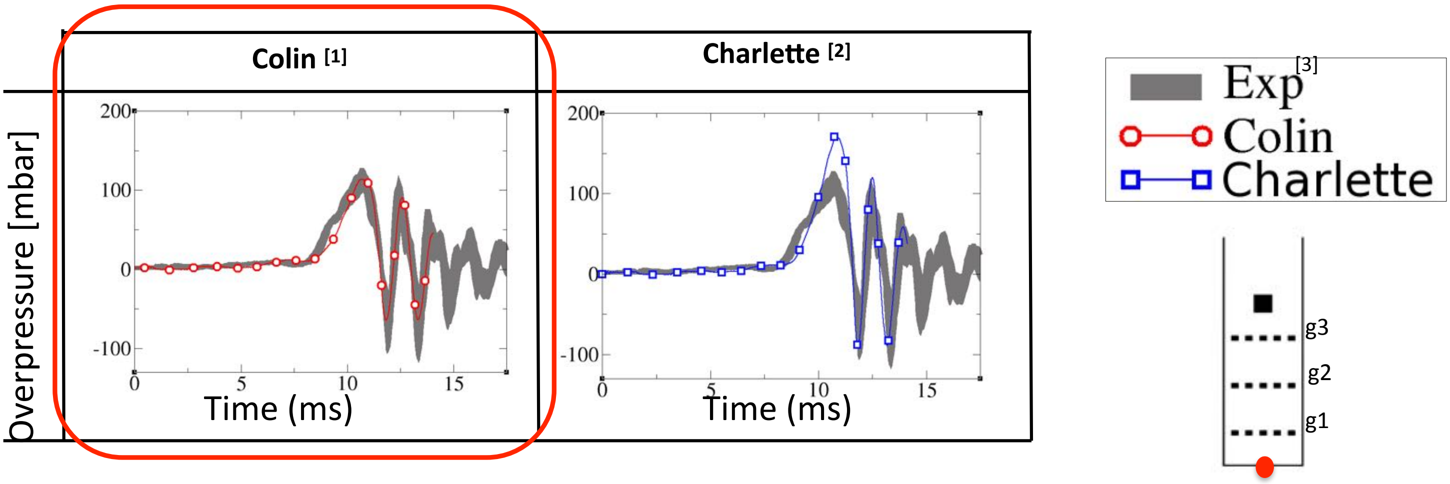
Flame Propagation



- Long laminar phase controls the flame shape and speed before it touches the obstacles
- Fast acceleration when flame becomes turbulent
- Acoustic oscillations at the end of combustion

Results – Small Scale Chamber

Choice of the turbulent combustion model



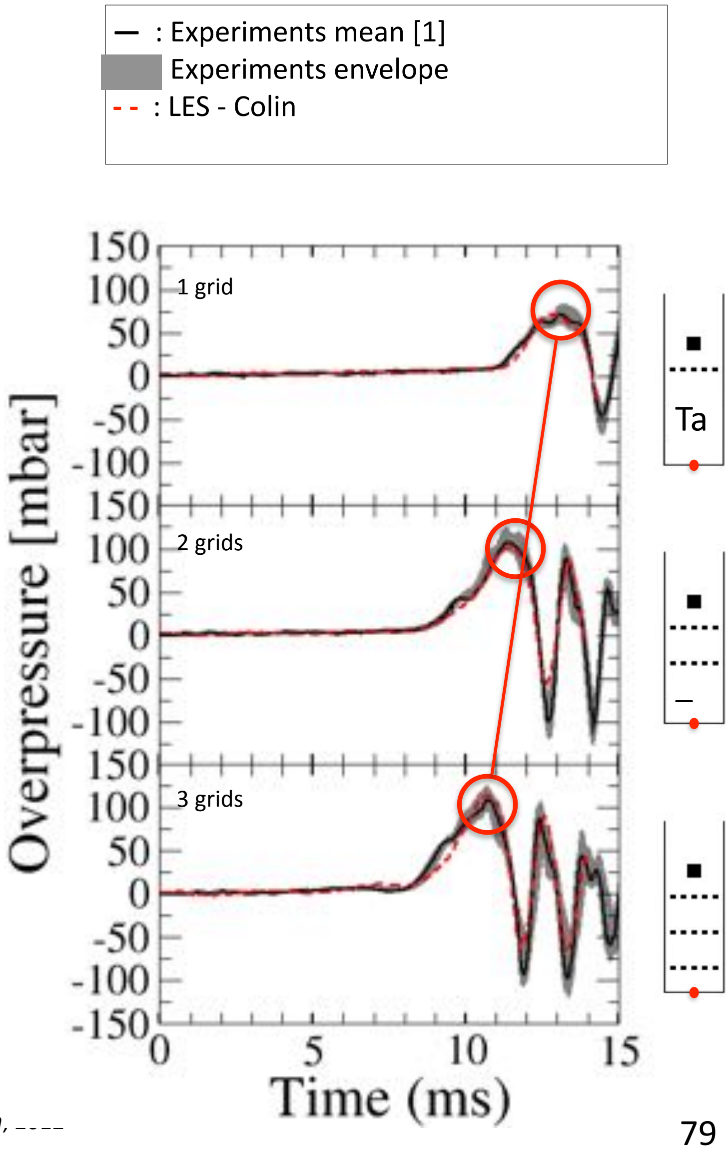
- Over-estimation of the maximum overpressure reached by Charlette's model.
- Colin's model gives the right behavior.

Turbulent combustion model for small scale chamber simulations: **Colin**

[1] Colin et al, *Physics of fluids*, 2000
[2] Charlette et al, *Combustion and Flame*, 2002
[3] Masri, et al, *Industrial & Engineering Chemistry Research*, 2012

Results – Small Scale

Influence of the number of grids :

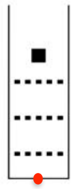


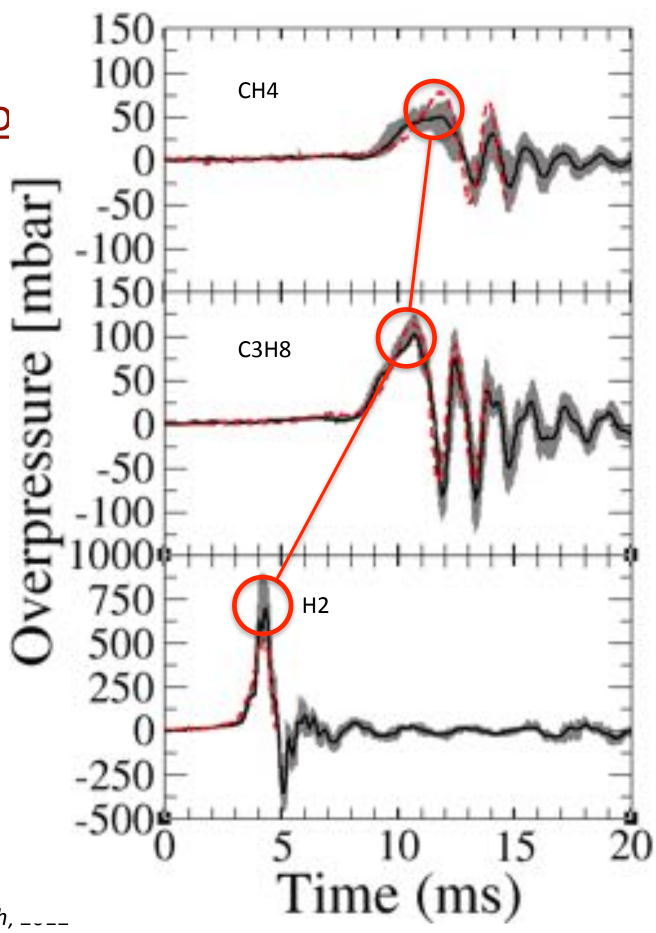
[1] Masri, et al, *Industrial & Engineering Chemistry Research*, _ _ _ _

Results – Small Scale

Influence of the fuel type : Response to flame properties S_l^0, d_l^0

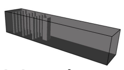
— : Experiments mean [1]
▮ : Experiments envelope
- - : LES - Colin





[1] Masri, et al, *Industrial & Engineering Chemistry Research*, ____

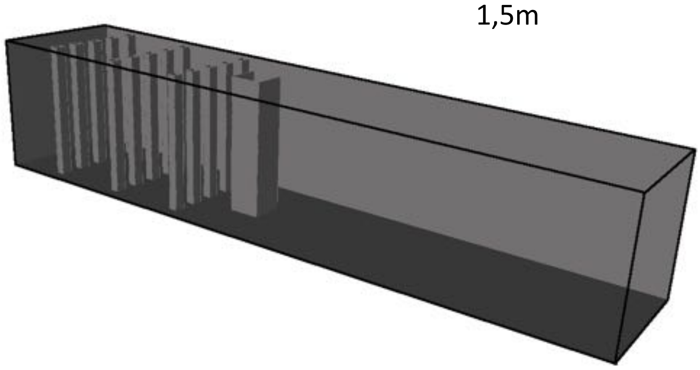
Scaling things up: by 6



Masri setup
25cm

x6

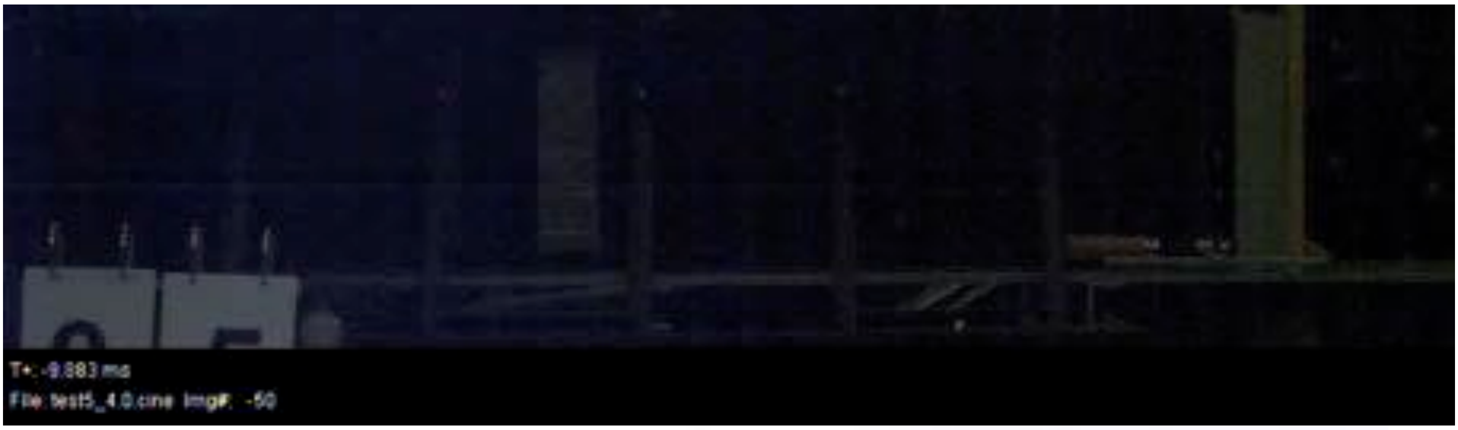
Scaled-up reproduction of Masri setup (x6)
1,5m



Results – Medium Scale Chamber

Flame Propagation

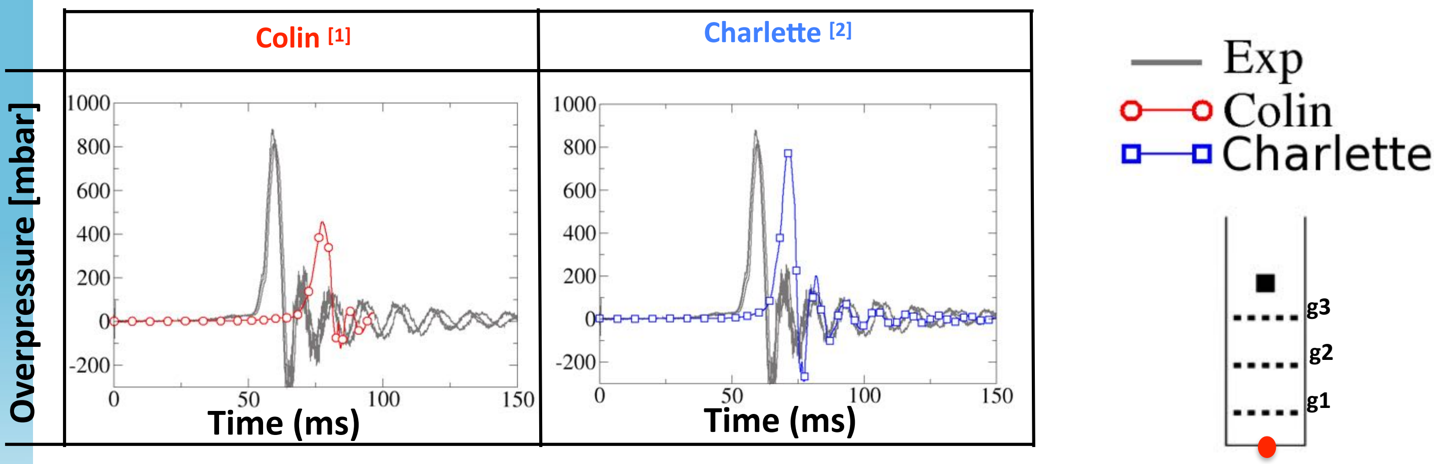
Experiments



LES

Results – Medium Scale Chamber

Choice of the turbulent combustion model



The turbulent combustion model which worked perfectly for small scale does not work for the medium size chamber simulations…
Results even worse for the large scale chamber

O. Vermorel, P. Quillatre and T. Poinsot and Ph. Ricoux. LES of explosions in venting chamber: a test case for premixed turbulent combustion models. Comb. Flame. 2017, 183, 207-224.

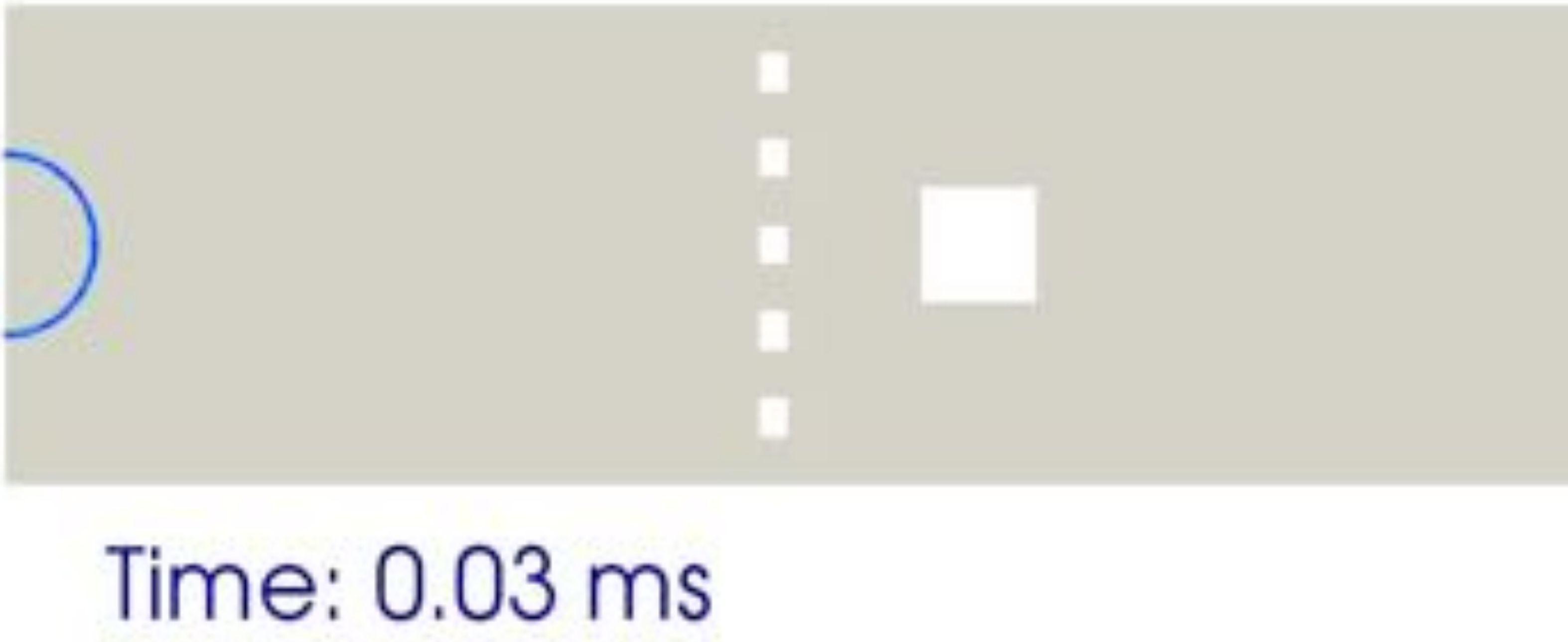
Implications for turbulent combustion models

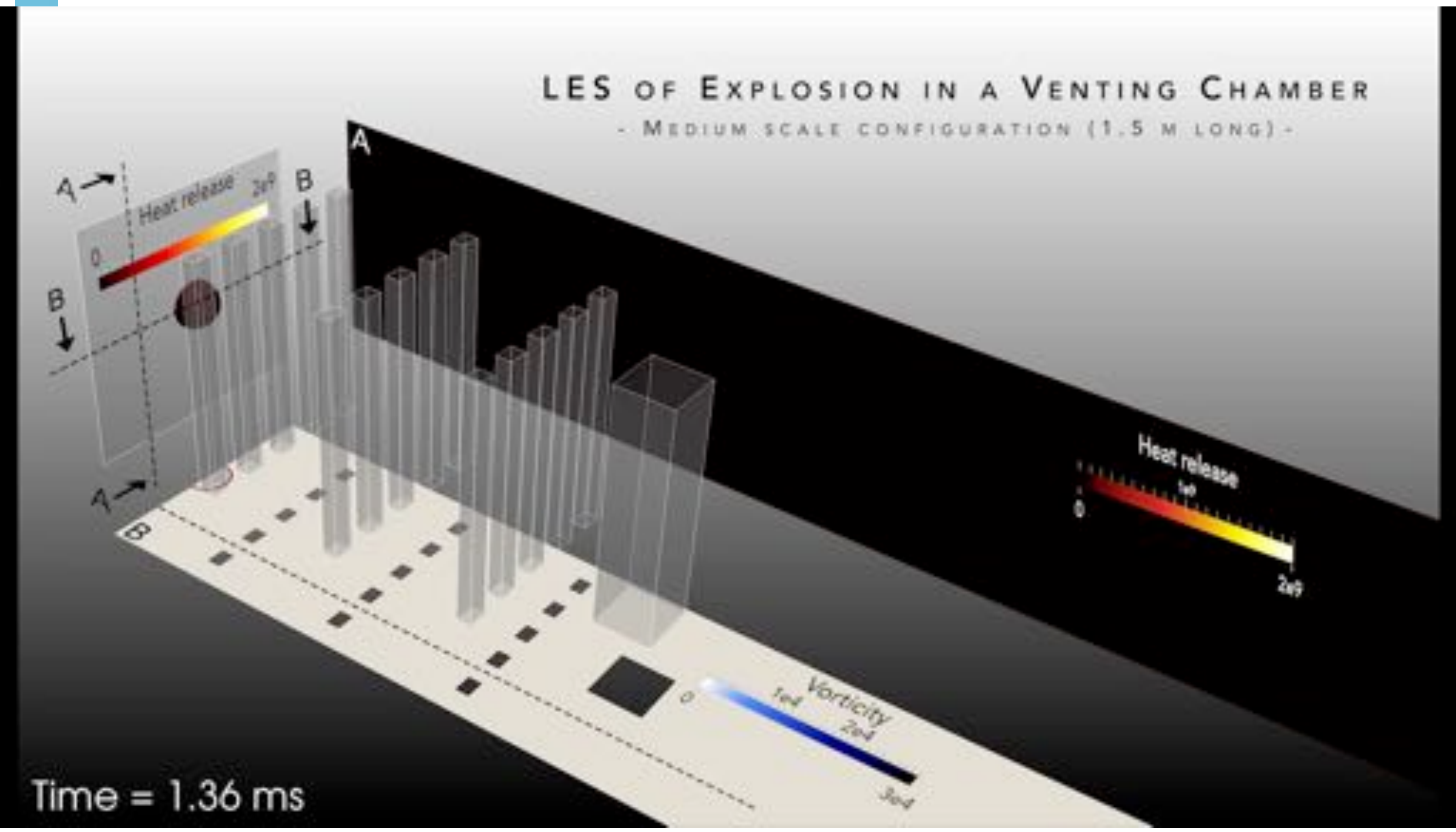
- Going from a volume of 1 to a volume of 24^3= 13000 shows that a standard (good!) model has problems for upscaling
- Solutions:
  - use more points !
  - use Deep Learning (Lapeyre et al, Comb. Flame 2019)

USING MORE GRID POINTS ?!

- Grid refinement can replace models !
- Adding more points when the scale increases is a simple but expensive way of solving the problem
- This requires very large computers. Example: the INCITE BG machines

1 billion cell LES:

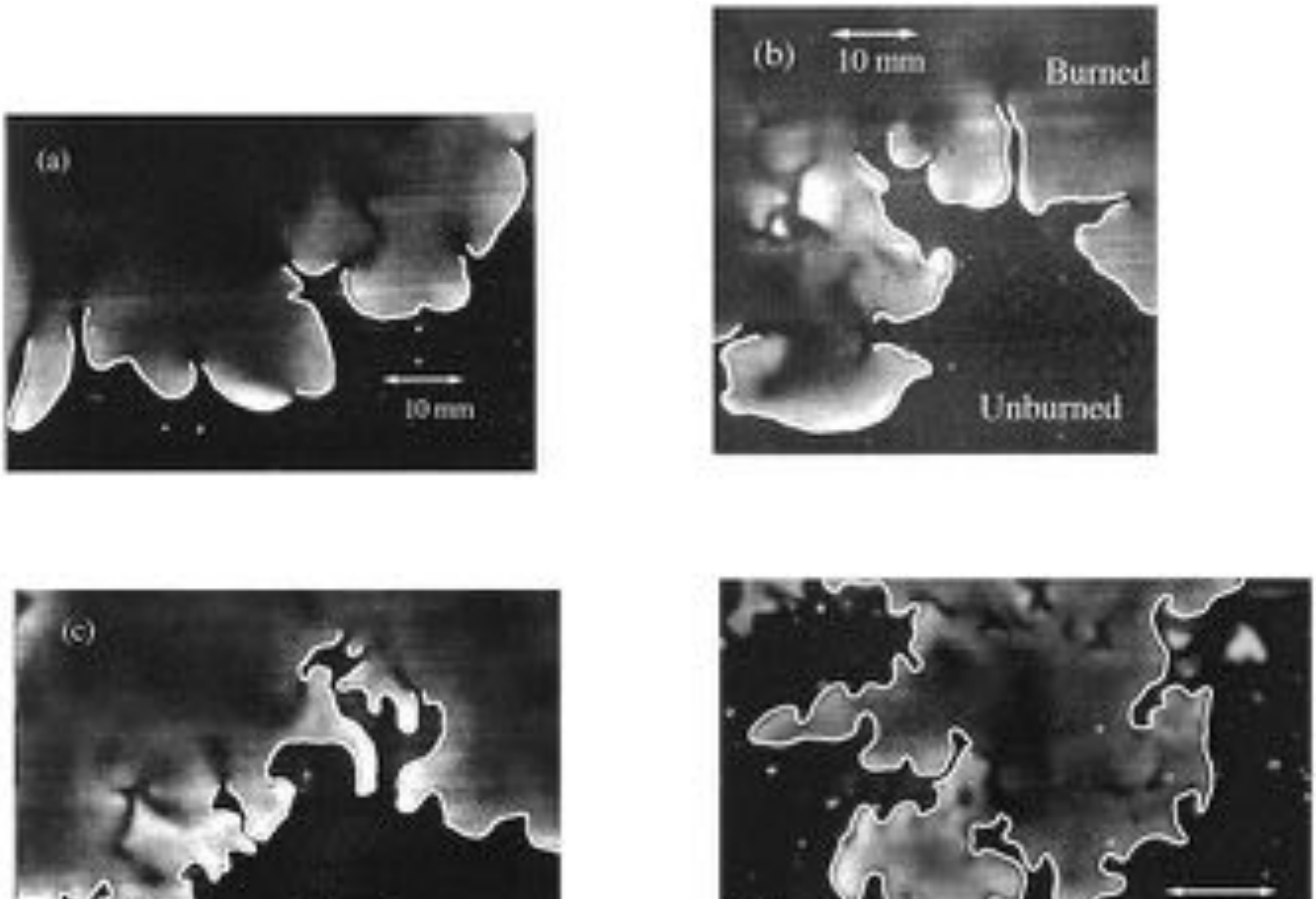



LES of Explosion in a Venting Chamber
- Medium scale configuration (1.5 m long) -
A
B
Heat release
0
2e9
Vorticity
0
1e4
2e4
3e4
Time = 1.36 ms

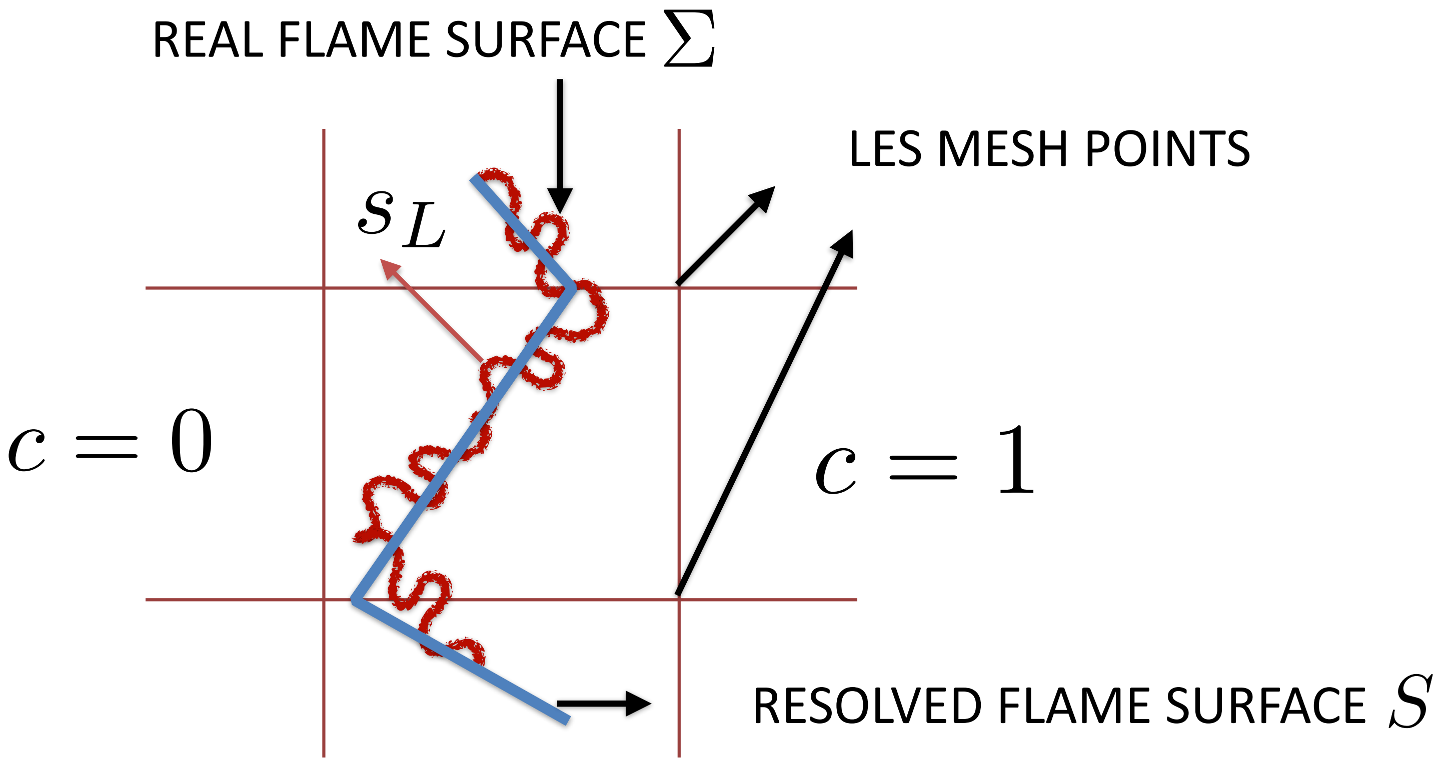
OUTLINE

- Energy policies and combustion
- Tools to simulate reacting flows
- Turbulent premixed flames and explosions
- Deep learning for turbulence combustion interaction

Premixed turbulent flames:

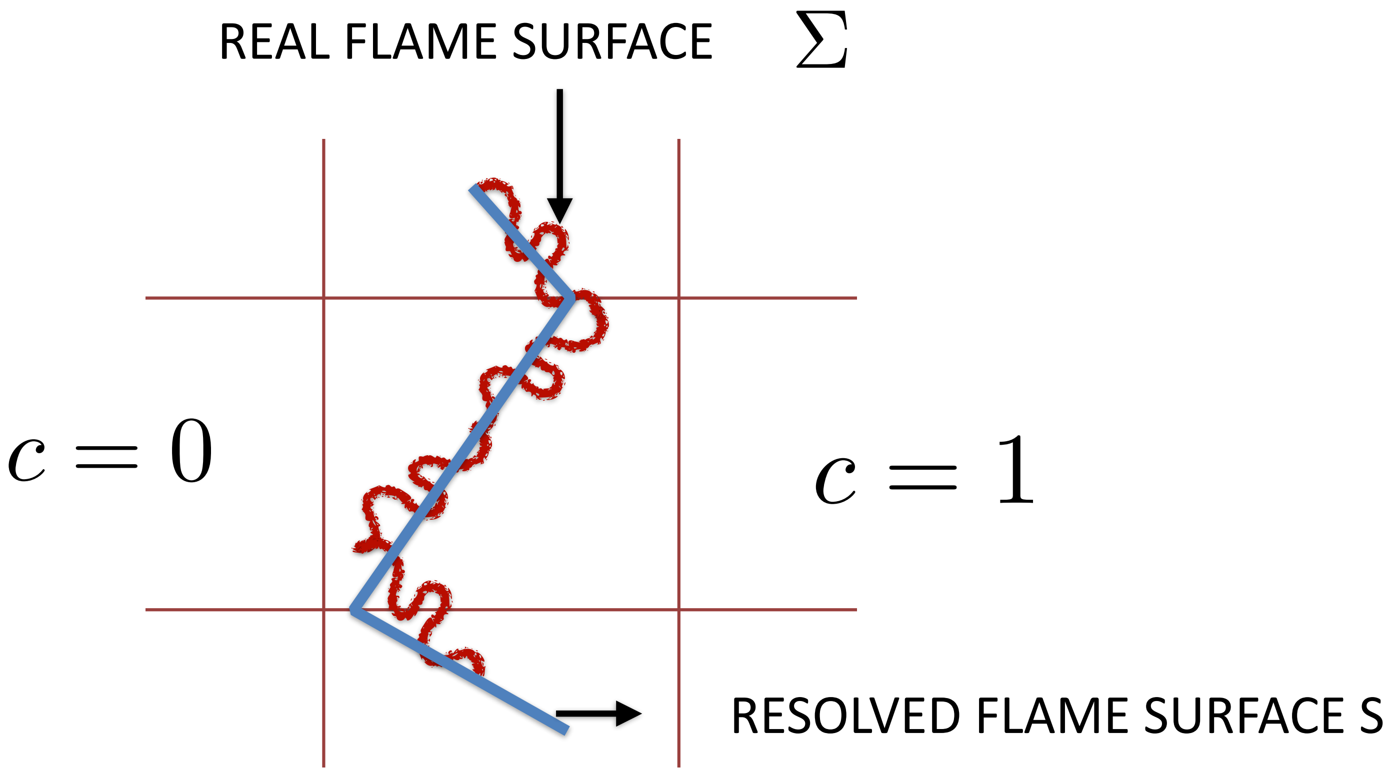


Closure problem in turbulent premixed flames: finding the sub grid surface in LES



[1] Butler, T. D. & O'Rourke, P. J. (1977). Symp. (Int.) Combust. 16, 1503 – 1515.

Can I guess Σ knowing s and the resolved temperature field c

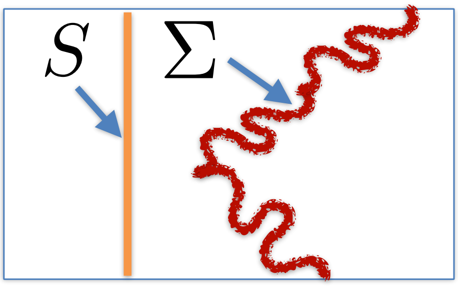


$$\Xi = \frac{\overline{|\nabla c|}}{|\nabla \overline{c}|} = f(\overline{c}, u', \dots)$$

$$\overline{|\nabla \mathbf{c}|} = \Xi |\nabla \overline{c}|$$

TOTAL FLAME SURFACE

RESOLVED FLAME SURFACE



$$\Xi = \frac{\Sigma}{S} = \frac{\overline{|\nabla c|}}{|\nabla \overline{c}|}$$

Convolutional neural network (CNN) approach:

$\overline{\mathbf{c}}, \overline{|\nabla \mathbf{c}|}$ defined over a subdomain Ω

$$f_{CNN}(\overline{\mathbf{c}}) = \frac{\overline{|\nabla \mathbf{c}|}}{|\nabla \overline{\mathbf{c}}|}$$

$$f_{CNN} : \mathbb{R}^{\Omega} \mapsto \mathbb{R}^{\Omega}$$

Efficiency functions f :

- …
- 1989 - Gouldin [5] fractal
- 2000 - Colin *et al.* [6]
- 2002 - Charlette *et al.* [7]
- 2011 - Wang *et al.* [2]
- …

[5] Gouldin, F. C., Bray, K. N. C., & Chen, J. Y. (1989). Chemical closure model for fractal flamelets. *Combustion and flame*, *77*(3-4), 241-259.

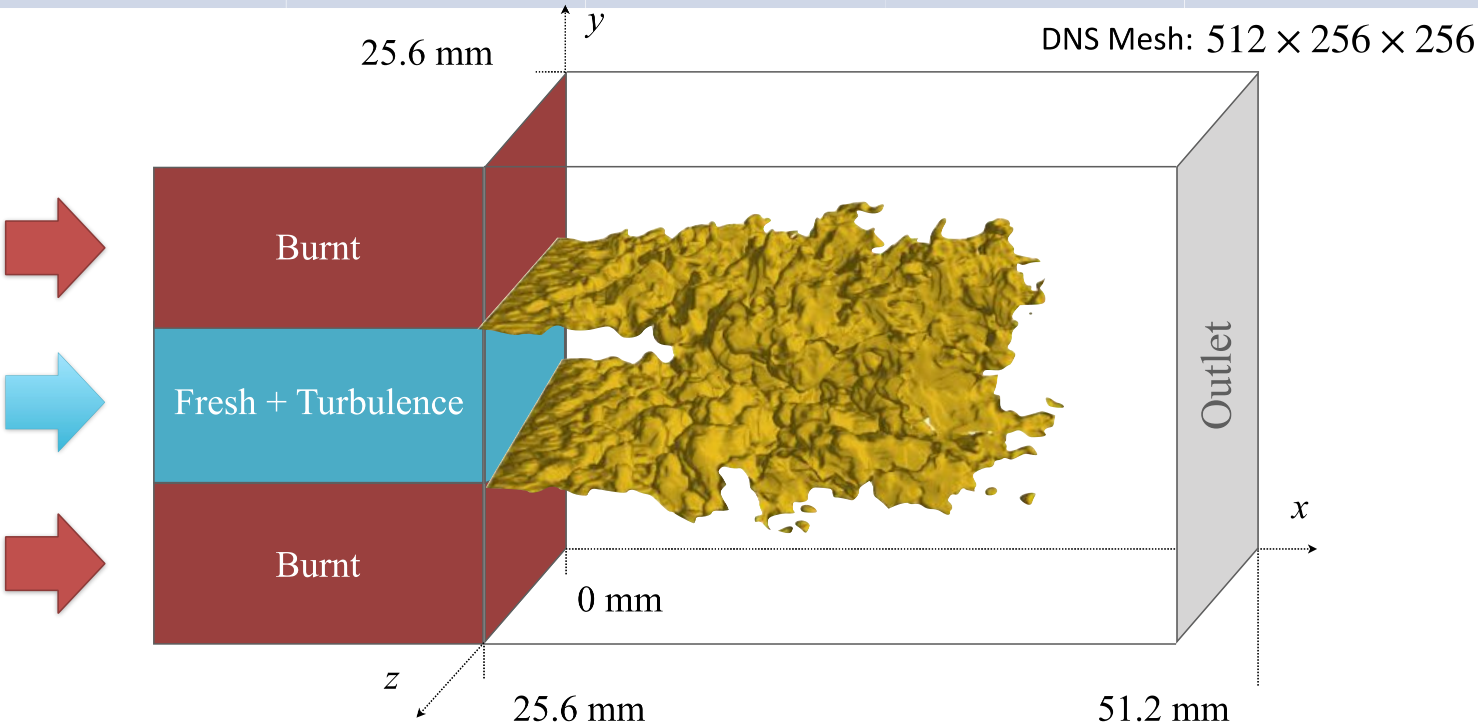
[6] Colin, O., Ducros, F., Veynante, D., & Poinsot, T. (2000). A thickened flame model for large eddy simulations of turbulent premixed combustion. Physics of fluids, 12(7), 1843-1863.

[7] Charlette, F., Meneveau, C., & Veynante, D. (2002). A power-law flame wrinkling model for LES of premixed turbulent combustion Part I: non-dynamic formulation and initial tests. Combustion and Flame, 131(1-2), 159-180.

Numerical setup to train and test the CNN for premixed turbulent flame: a turbulent Bunsen burner

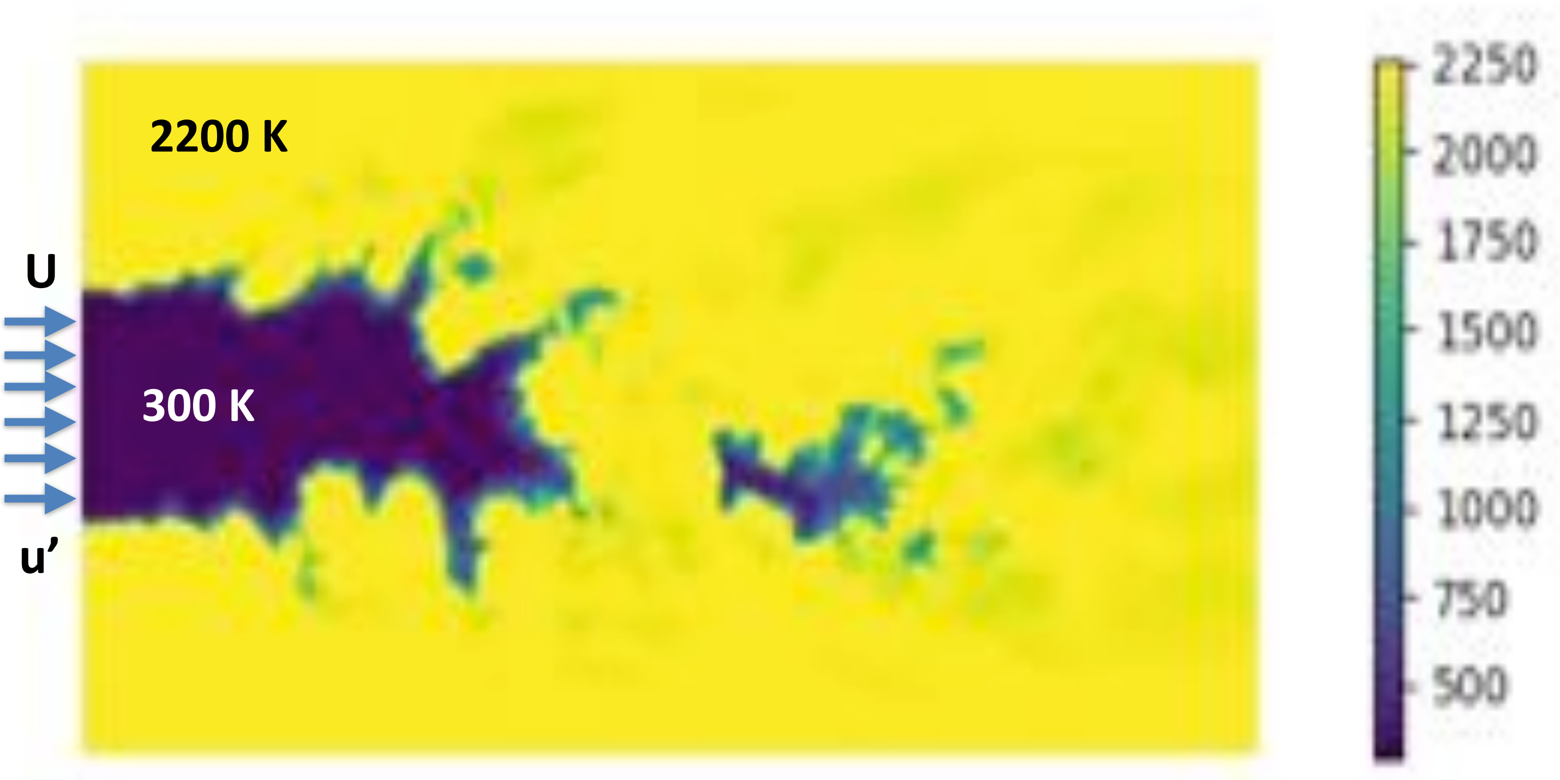
The DNS used to train the CNN:

Fuel	Fresh gases velocity	Injected turbulent intensity	Re_τ	Ka
Methane - Air φ=1	***10 m/s***	***5% / 10%***	***5 and 10***	***< 1***

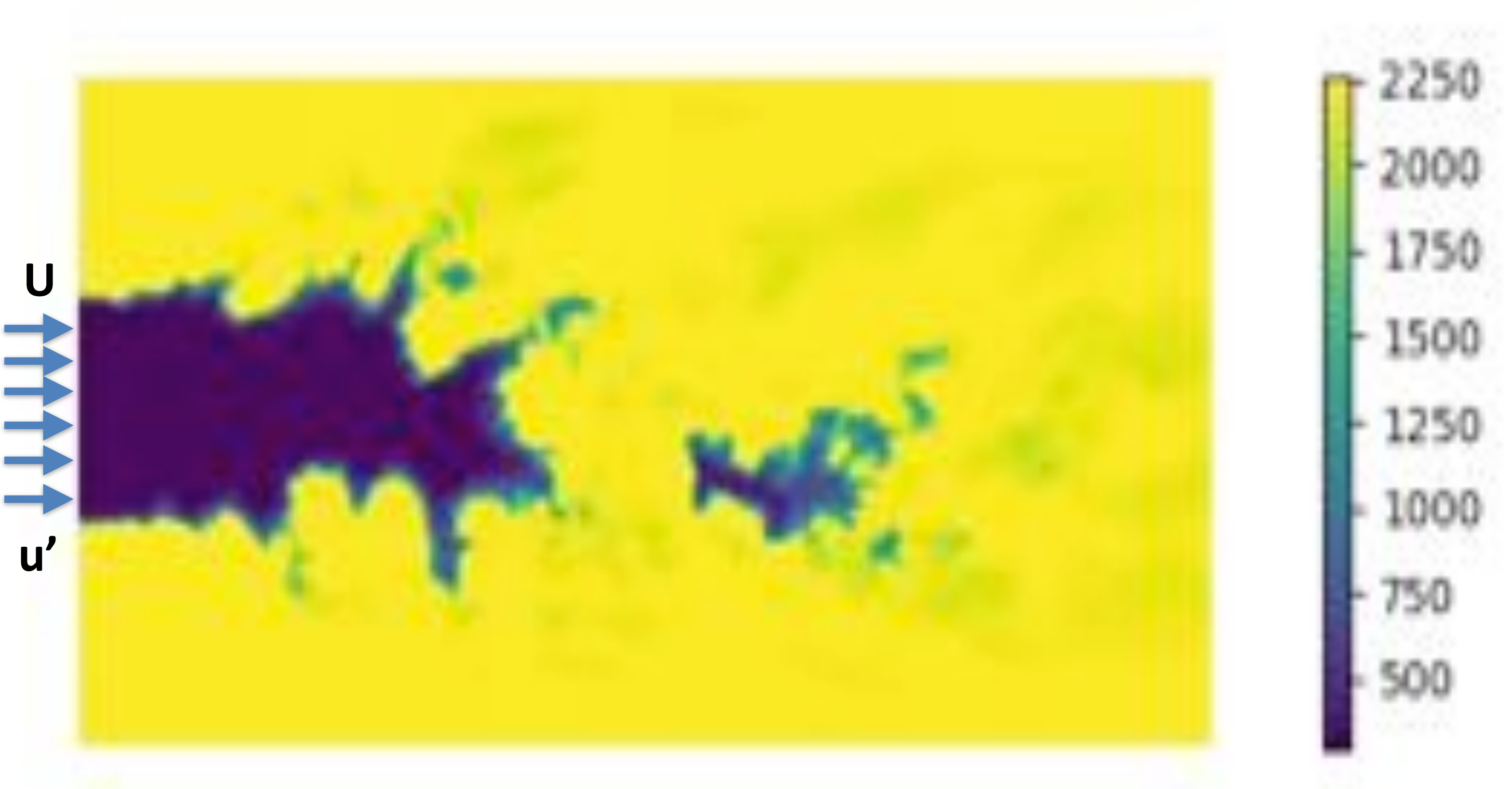


Similar to: [8] Bell, J. B., Day, M. S., Grcar, J. F., Lijewski, M. J., Driscoll, J. F., & Filatyev, S. A. (2007). Numerical simulation of a laboratory-scale turbulent slot flame. *Proceedings of the combustion institute*, *31*(1), 1299-1307.

The DNS used to train the CNN:



The problem depends on two parameters: the mean inlet velocity U and the RMS turbulent velocity u'



Simulations

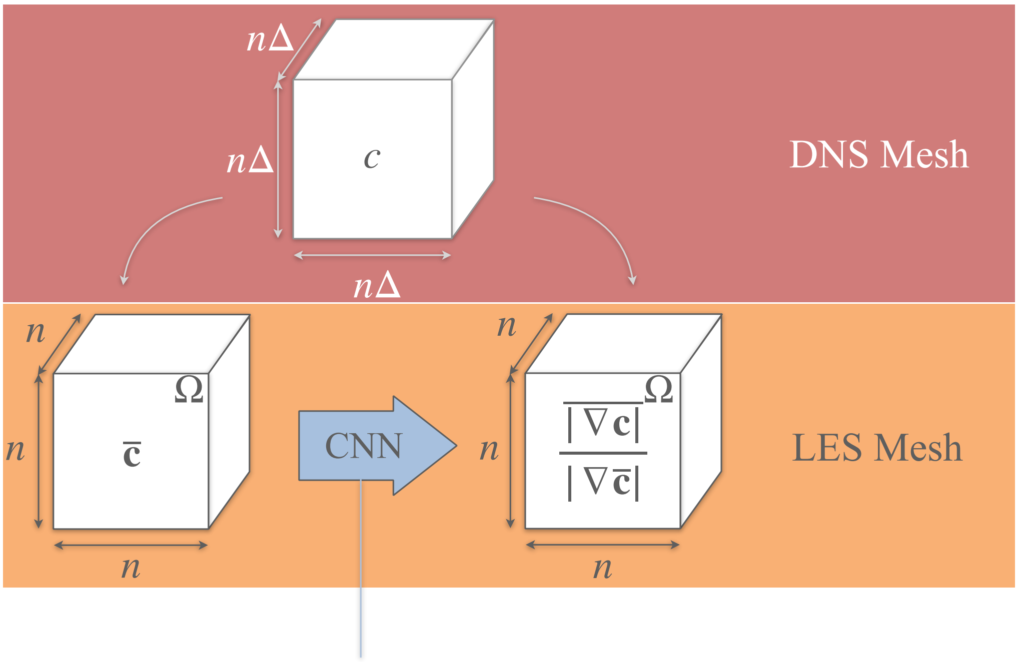
Name	u' / S_L	Inlet velocity	Resolution	Turbulent combustion model	Comparison	
Train 1	*1.23*	*Constant*	*DNS*	*Resolved*	*ø*	Train
Train 2	*2.47*	*Constant*	*DNS*	*Resolved*	*ø*	
Mushroom	*1.23*				*A priori*	Test

TRAINING THE NETWORK ON DNS

A PRIORI TEST ON DNS DATA

A priori study

Building the dataset

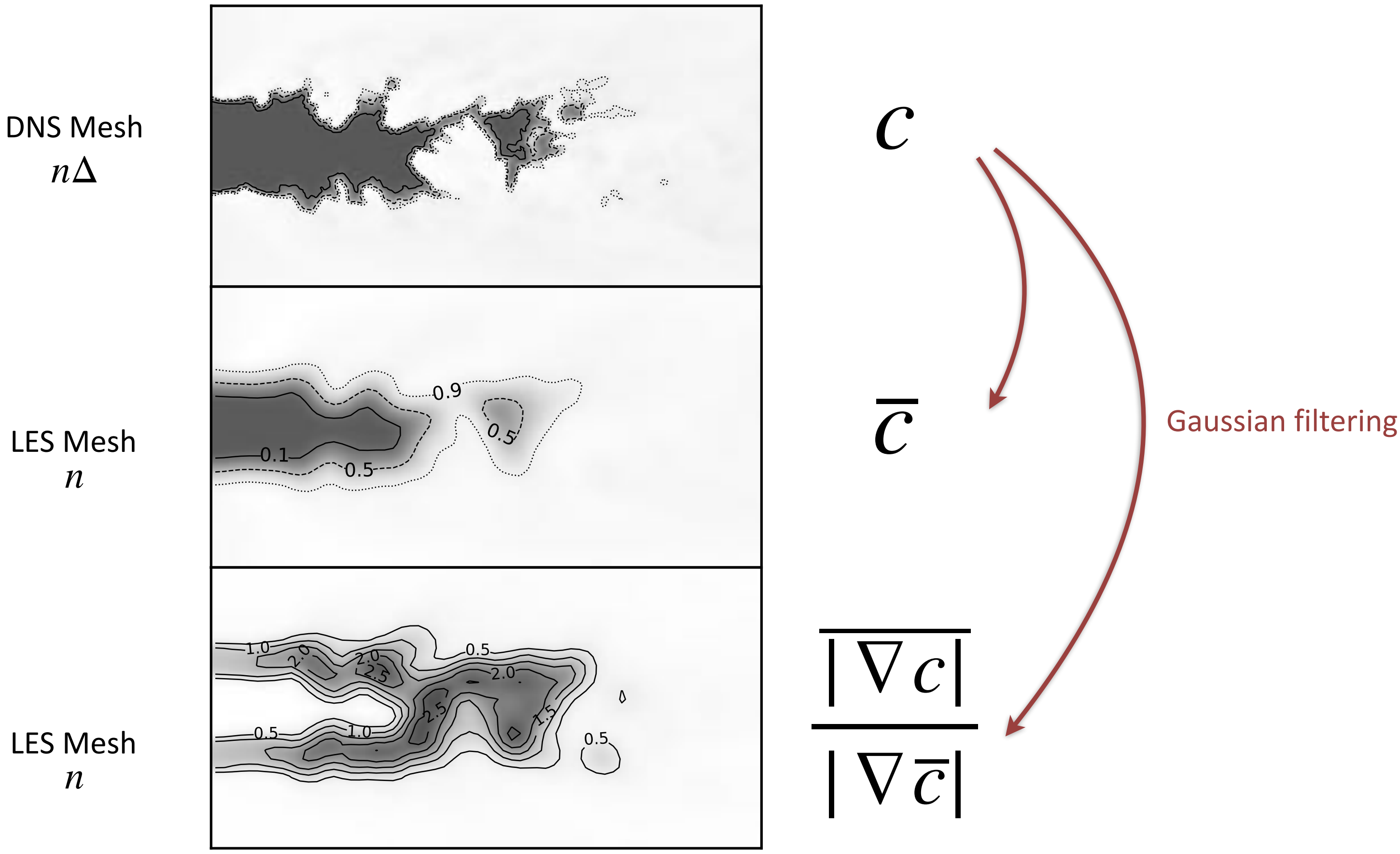


Convolutional neural network

Gaussian filtering equivalent to flame thickening Δ

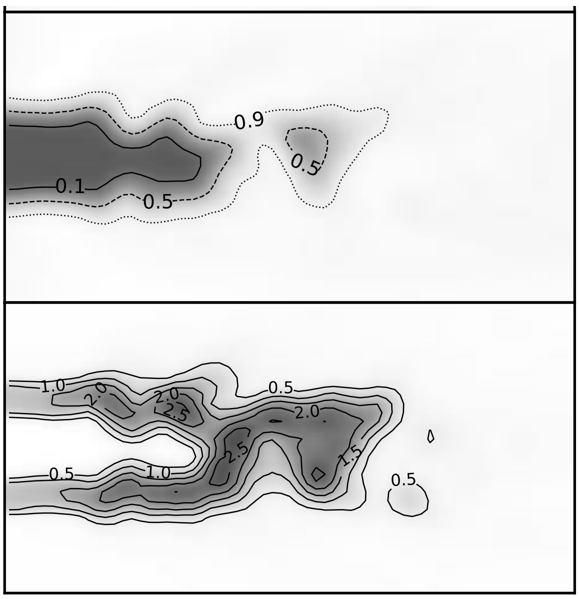
$$F_\Delta(n) = \begin{cases} e^{-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2} & \text{if } n \in [1,N] \\ 0 & \text{otherwise} \end{cases}$$

When the CNN learns the dataset:



When the CNN is used:

LES Mesh
n



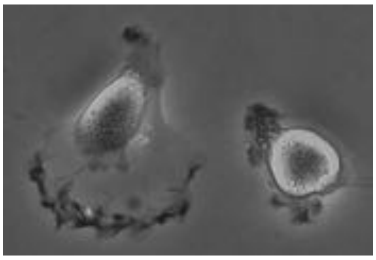
LES Mesh
n

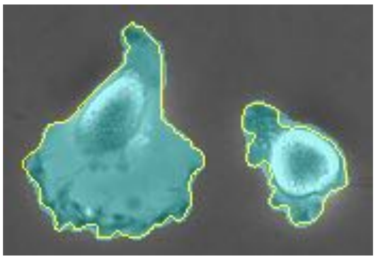
$$\bar{c}$$

$$\frac{\overline{|\nabla c|}}{|\nabla \bar{c}|}$$

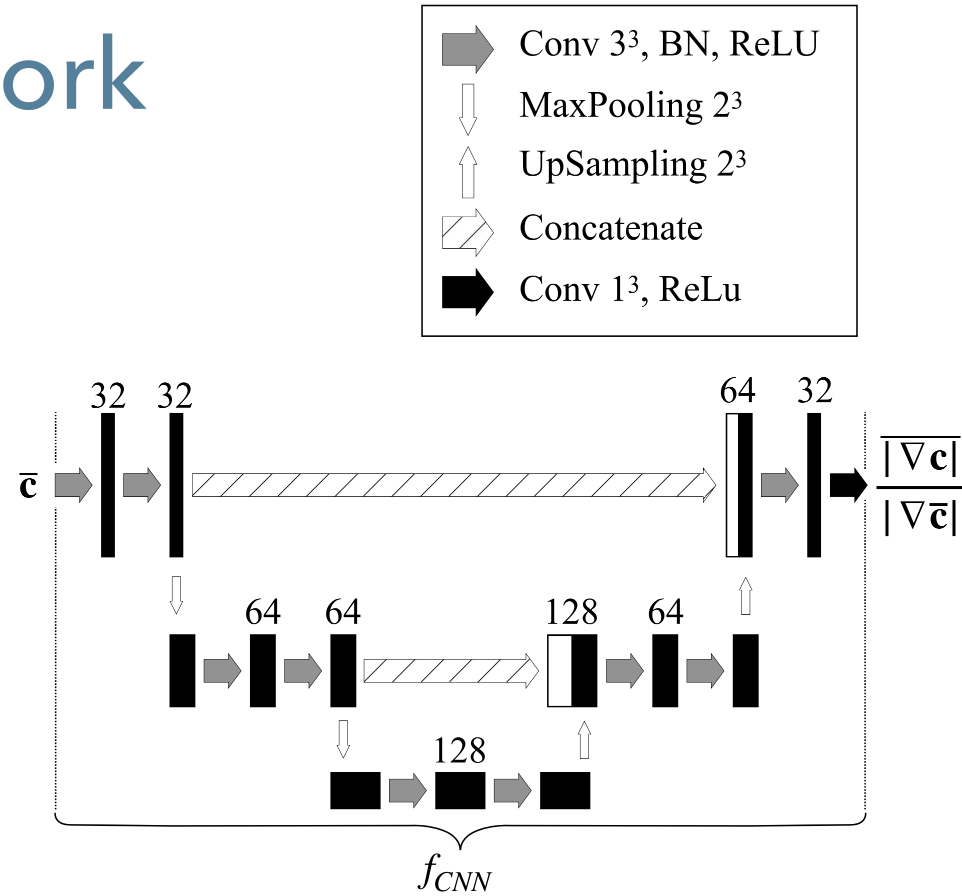
Neural network

Input





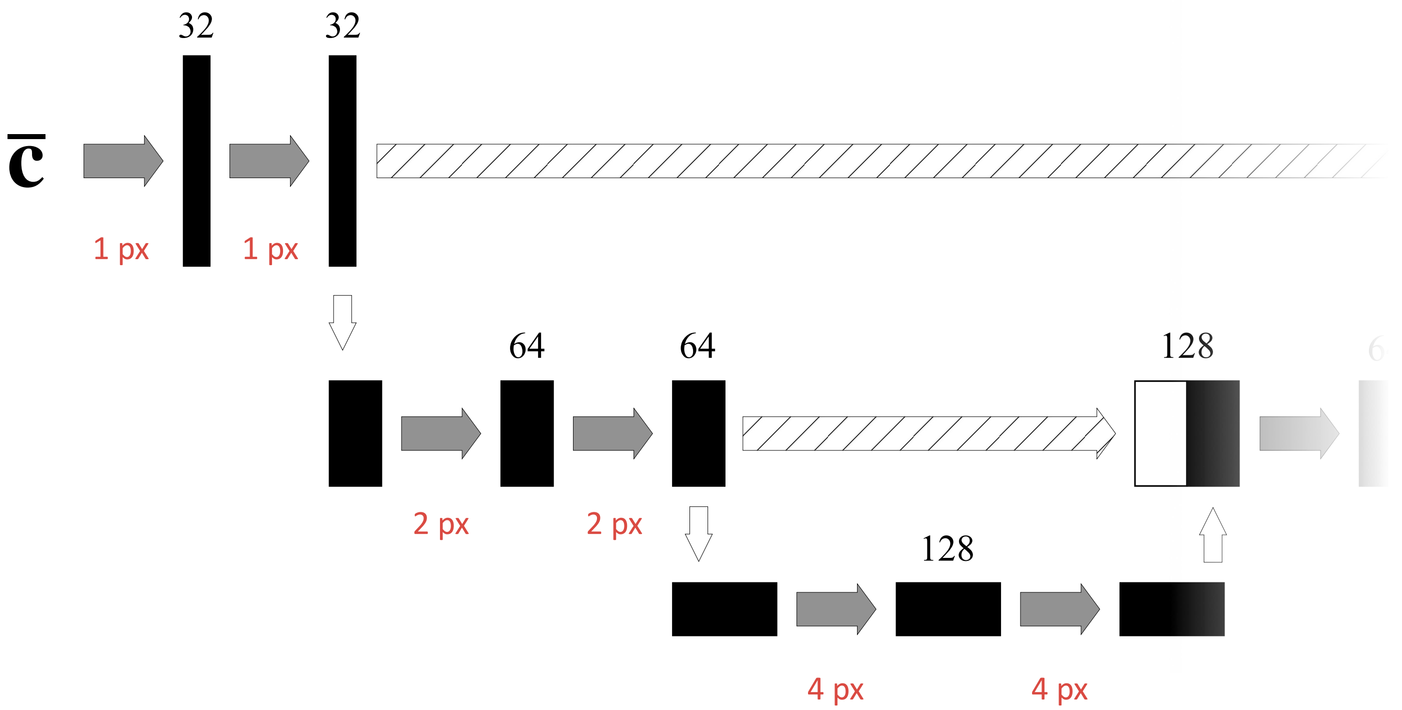
Segmented image



Architecture is adapted from a medical image segmentation network [9]

[9] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

Neural network



- Information propagates 14 pixels sideways ≈ maximum size of learned structures
- Network is trained on 16^3 inputs. Fully convolutional so that the full field is explored and used for training

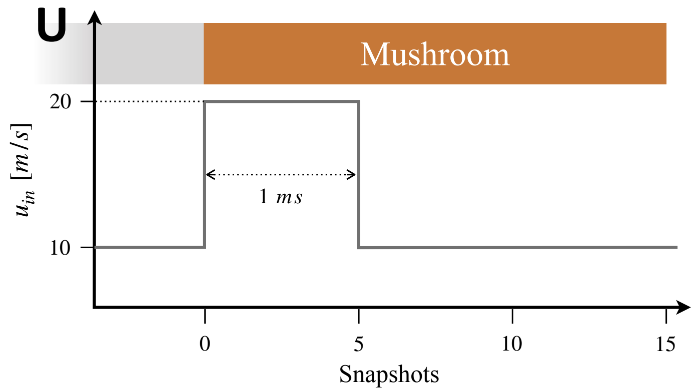
Simulations

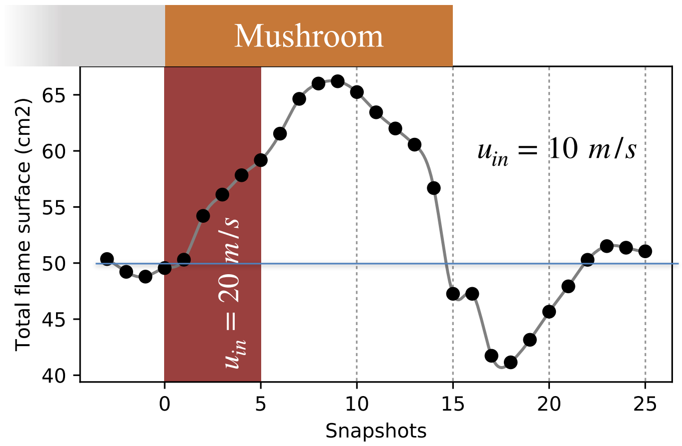
Name	u' / S_L	Inlet velocity	Resolution	Turbulent combustion model	Comparison	
Train 1	1.23	Constant	DNS	Resolved	ø	Train
Train 2	2.47	Constant	DNS	Resolved	ø	
Mushroom	1.23				A priori	Test
PULSE_DNS	1.23	Sinewave	DNS	Resolved	ø	
PULSE_CNN	1.23	Sinewave	LES	CNN [9]	A posteriori	Coupled
PULSE_DYN	1.23	Sinewave	LES	Dynamic [2]	A posteriori	

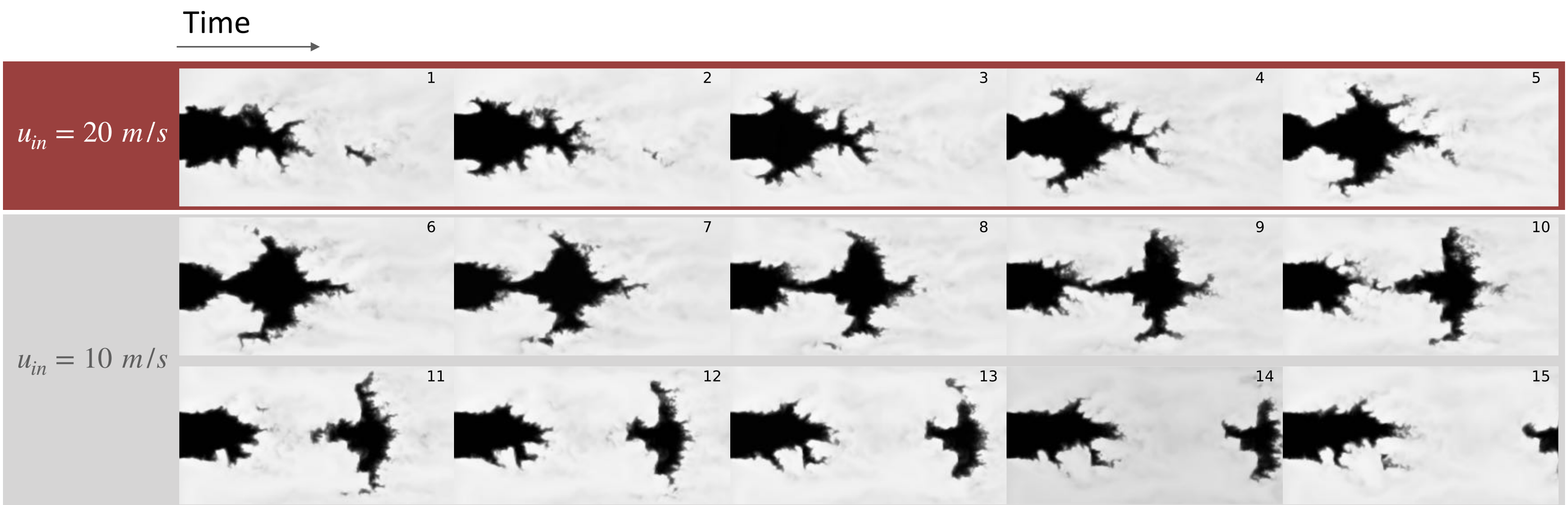
A PRIORI TEST ON DNS DATA

DNS / LES code: AVBP (cerfacs.fr/en/computational-fluid-dynamics-softwares)

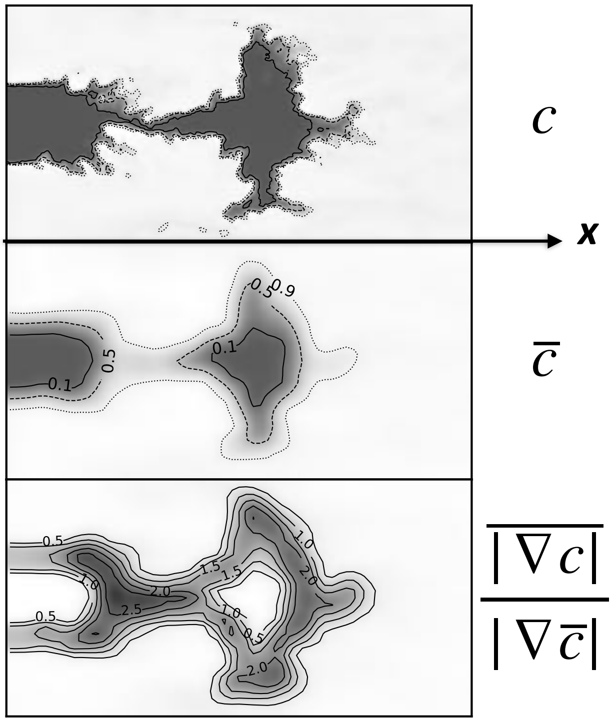
A priori tests for a pulsated flame:



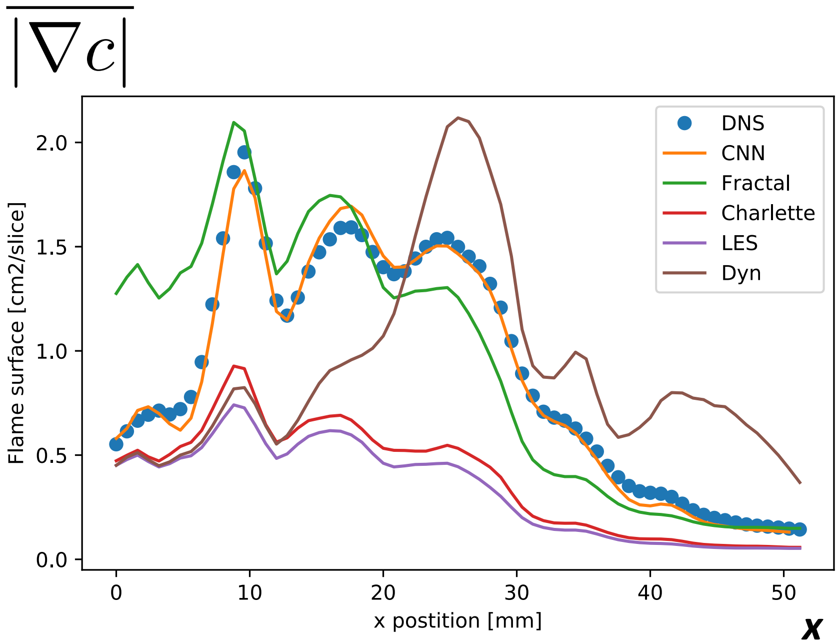




A priori tests for a pulsated flame:



Snapshot at highest flame surface



[10] Lapeyre, C. J., Misdariis, A., Cazard, N., Veynante, D., & Poinsot, T. (2018). Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *arXiv preprint arXiv:1810.03691. Submitted to Combust. Flame*

In terms of computers:

- The CNN must be integrated in the LES code to compute flame wrinkling but the inference time (evaluation of f_{CNN}) becomes too long on CPU: GPUs are much better
- -> an hybrid architecture CPU/GPU is needed

$\overline{c}$

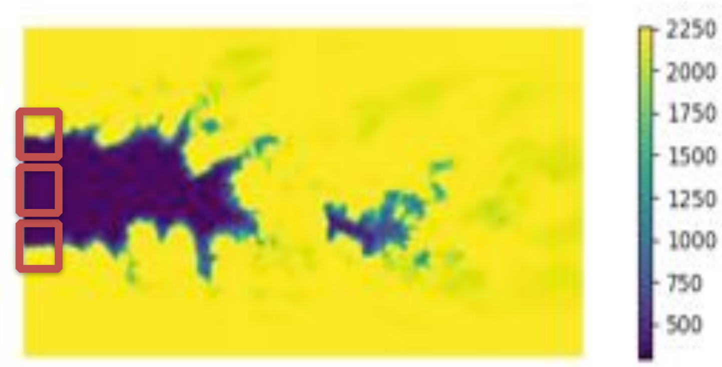
$f_{CNN}(\overline{c})$

CPU : Navier-Stokes solver
(AVBP)

GPU : CNN
(TensorFlow)

A comment on locality:

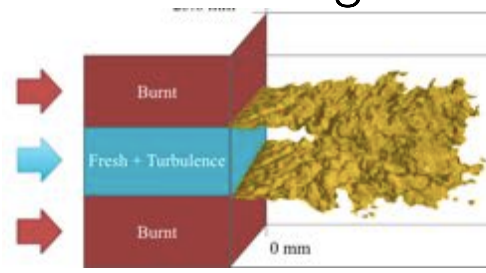
- During training, the CNN learns what wrinkling is, over the WHOLE domain



- During application, the CNN uses only points in a 14 pixel wide box… but it remembers what he has learnt during the training phase

Another comment on generality

- The CNN has learnt to predict sub grid flame wrinkling in this configuration and this one only



- How general is this knowledge ?
  - We dont know
  - Since we do not understand how the CNN works, we have no way to determine its range of validity: do we need to train the CNN for each flame (in which case we would need a DNS for each flame, which we do not have) ?
  - For the moment: we try it on other flames

Conclusion

Combining:

- LES is good and will take over other methods
- But it is expensive and not easy: specific codes must be built
- And subgrid models are still needed: they remain the weakest part of the modeling

Bibliography

[1] Butler, T. D. & O'Rourke, P. J. (1977). Symp. (Int.) Combust. 16, 1503 – 1515.

[2] Wang, G., Boileau, M., & Veynante, D. (2011). Combustion and Flame, 158(11), 2199-2213.

[3] Poinsot, T. & Veynante, D. (2011). Theoretical and Numerical Combustion, Third Edition

[4] Marble, F. E. & Broadwell, J. E. (1977). The coherent flame model for turbulent chemical reactions, Tech. Rep. Tech. Rep. TRW-9-PU, Project Squid.

[5] Gouldin, F. C., Bray, K. N. C., & Chen, J. Y. (1989). Chemical closure model for fractal flamelets. Combustion and flame, 77(3-4), 241-259.

[6] Colin, O., Ducros, F., Veynante, D., & Poinsot, T. (2000). A thickened flame model for large eddy simulations of turbulent premixed combustion. Physics of fluids, 12(7), 1843-1863.

[7] Charlette, F., Meneveau, C., & Veynante, D. (2002). A power-law flame wrinkling model for LES of premixed turbulent combustion Part I: non-dynamic formulation and initial tests. Combustion and Flame, 131(1-2), 159-180.

[8] Bell, J. B., Day, M. S., Grcar, J. F., Lijewski, M. J., Driscoll, J. F., & Filatyev, S. A. (2007). Numerical simulation of a laboratory-scale turbulent slot flame. Proceedings of the combustion institute, 31(1), 1299-1307.

[9] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

[10] Lapeyre, C. J., Misdariis, A., Cazard, N., Veynante, D., & Poinsot, T. (2018). Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *arXiv preprint arXiv:1810.03691. Submitted to Combust. Flame*