Simulation tools for 3D reacting flows

T. Poinsot\textsuperscript{1,2,3}, D. Veynante\textsuperscript{4}, F. Nicoud\textsuperscript{2}, B. Cuenot\textsuperscript{2}, L. Selle\textsuperscript{1}, G. Lartigue\textsuperscript{2,5}, L. Gicquel\textsuperscript{2}, V. Moureau\textsuperscript{5}, S. Candel\textsuperscript{4} and others!

\textsuperscript{1}IMFT, CNRS, Toulouse France
\textsuperscript{2}/\textsuperscript{3}CERFACS, Toulouse France
\textsuperscript{3}CTR, Stanford and NASA Ames
\textsuperscript{4}EM2C, Paris
\textsuperscript{5}CORIA, Rouen

Copyright Dr T. Poinsot 2015
The reacting Navier Stokes equations and the codes to solve them

The differences between RANS, DNS and LES

LES requires computing vortices and acoustic waves in CFD: is it simple?

High Performance Computing and combustion: tabulation, reduced schemes, etc
<table>
<thead>
<tr>
<th></th>
<th>LAMINAR</th>
<th>TURBULENT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0D codes:</strong> T=f(t)</td>
<td>CHEMKIN, COSILAB CANTERA</td>
<td>PSR</td>
</tr>
<tr>
<td>Yk=gk(t)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PREMIXED</strong></td>
<td>CHEMKIN, COSILAB CANTERA: ‘premix’</td>
<td><strong>TURBULENT CODES</strong></td>
</tr>
<tr>
<td><strong>DIFFUSION</strong></td>
<td>CHEMKIN, COSILAB CANTERA: ‘oppdiff’</td>
<td></td>
</tr>
</tbody>
</table>
Thermodynamics: equation of state
A mixture of k= 1 to N perfect gases.
Composition characterized by mass fractions Yk=mk/m

\[
\frac{1}{W} = \sum_{k=1}^{N} \frac{Y_k}{W_k}
\]

Molar weight of the mixture

\[
P = \rho \, r \, T
\]

with

\[
r = \frac{R}{W} = \sum_{k=1}^{N} \frac{Y_k}{W_k} \, R = \sum_{k=1}^{N} Y_k \, r_k
\]

Ch. 1 Section 1.1
MASS CONSERVATION: one equation for each species \( k = 1 \) to \( N \)

\[ \rho_k = \rho \ast Y_k \] is the partial density of species \( k \)

\[ \frac{\partial \rho_k}{\partial t} + \frac{\partial}{\partial x_j} (\rho_k u_j) = - \frac{\partial}{\partial x_j} [J_{j,k}] + \dot{\omega}_k \]

- Diffusion
- Reaction rate

Ch. 1
ENERGY:

\[ \frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_j} (\rho E u_j) = - \frac{\partial}{\partial x_j} [u_i (P \delta_{ij} - \tau_{ij}) + q_j] + \dot{w}_T + Q_r \]

These equations are unfiltered, instantaneous.

They are exact; they contain turbulence.

The main issue is how to handle turbulence!
Methods for turbulent flows:

- LES
- Experiment or DNS
- RANS: averages

Ch. 4 Section 4.2
Instantaneous and averaged flows have not much in common

Chen et al
31st Symp.
In spectral space:

Energy $E(k)$ vs. wave number $k$:

- Modeled in RANS
- Computed in DNS
- Computed in LES
- Modeled in LES

$E(k)$ peaks at $k_C$.
A big 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 is performed locally over space (a small zone around each point). LES variables are time-dependent quantities.
Same duct computed with RANS and then with LES:

\[ t = 0.440 \, \text{s} \]
To resolve this velocity profile, we need:
- 10 points in the transverse direction
- 2 in the longitudinal direction (there is no gradient)

This is a very cheap RANS computation which can be done with only 20 points
To resolve this velocity profile, we need:
- many points in the transverse direction because gradients are stronger
- many points in the longitudinal direction because vortices have similar lengths in both directions
- a resolution in time because all velocities fluctuate

This LES will require many points and many time iterations
Your eyes are actually averaging like RANS codes: LES reveals the eddies hidden behind the ‘mean’:

Visualization of Turbulent Flame

Sandia Flame-E

Matthias Ihme
Marcus Herrmann
Heinz Pitsch

Center for Integrated Turbulence Simulations

Stanford University
COMPARISON OF RANS AND LES:

RANS

LES
RANS has other problems:
- “The mean does not mean anything” (Berkeley)
RANS has other problems:

- Certain phenomena can simply not be averaged: ignition, quenching, instabilities
Decrease the fuel mass fraction at inlet and watch for quenching:
RANS has other problems:

- Certain phenomena can simply not be averaged: ignition, quenching, instabilities

- Acoustic waves are not handled correctly by RANS formulations:
  - they are intrinsically unsteady mechanisms which can not be ‘averaged’
  - the solvers used in RANS utilize large turbulent and numerical viscosities which kill acoustic waves
Example of experiment which cannot be computed with RANS:

D. Mejia PhD IMFT
Response to harmonic forcing by a loud speaker:
Response to music:

RANS has other problems:

- Certain phenomena can simply not be averaged: ignition, quenching, instabilities

- Acoustic waves are not handled correctly by RANS formulations:
  - they are intrinsically unsteady mechanisms which can not be ‘averaged’
  - the solvers used in RANS utilize large turbulent and numerical viscosities which kill acoustic waves

- Thermoacoustics (combustion instabilities) is by nature unsteady and can’t be computed with RANS
WHAT WOULD THE MEAN OF THIS FLOW MEAN?
We classify flames as:

LAMINAR OR TURBULENT

But flames can be LAMINAR AND TURBULENT:
- In space
- In time
LAMINAR AND TURBULENT IN SPACE: Some parts of a combustor may actually be laminar. Models should degenerate to laminar flames when the flow becomes laminar.

- Most RANS models don’t... (EBU or Magnussen for example). Actually no turbulent combustion model can handle full chemistry with precision.
LAMINAR AND TURBULENT IN TIME: flames can begin as laminar and become turbulent:
- Explosions
- Piston engines
Sydney or Gexcon experiments

- Box: 5x5x25 cm
- Fuel/Air mixture at $\Phi=1$
- Fuel:
  - LPG (95% $C_3H_8$)
  - CNG (90% $CH_4$)
  - H2
- One central square obstruction
- 3 turbulence generating grids
Piston engines: flames begin as laminar kernels and become turbulent later.

A model for ignition in a piston engine must account for the initial laminar phase. This initial phase lasts longer when the flow is less turbulent.
Surprise for beginners:

RANS codes do turbulent flames but can NOT do laminar flames:
• RANS models do not resolve flame fronts
• RANS models resolve only the mean position of flames: the laminar structure is fully modelled
• RANS codes are usually very dissipative and cannot track flame dynamics: a RANS flame front has a thickness of the order of 1 cm
• Molecular transport is usually not modelled in many RANS formulations (needed for laminar flames)
OK, we should not do RANS.

So, what do we do?

LES.... or even DNS!

Going from RANS to LES is a big step:
- Introduction of turbulence at inlets?
- Turbulence models?
- Introduction of chemistry?
In RANS codes, you only had to choose values for k and epsilon at inlets.
In LES codes, you must generate explicit turbulence.
Two different issues:
- A turbulence spectrum must be chosen to construct a turbulent signal \( u,v,w \) \((x,y,z=z_0,t)\).
- This signal must then be imposed at the inlet of the LES. If this is a compressible flow, this must be done without introducing noise. In a subsonic flow this is difficult.


Turbulence models: apparently, LES and RANS are not very different

RANS, time averaged

\[
\frac{\partial \rho \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_i} (\rho \bar{u}_i \bar{u}_j) + \frac{\partial p}{\partial x_j} = \frac{\partial}{\partial x_i} \left( \bar{\tau}_{ij} - \rho \bar{u}_i \bar{u}_j \right)
\]

\[
\bar{u}_i \bar{u}_j = -\nu_t \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \bar{u}_k}{\partial x_k} \right)
\]

LES, space filtered

\[
\frac{\partial \rho \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i} (\rho \tilde{u}_i \tilde{u}_j) + \frac{\partial p}{\partial x_j} = \frac{\partial}{\partial x_i} \left[ \bar{\tau}_{ij} - \rho (\tilde{u}_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j) \right]
\]

\[
\tilde{u}_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j = -\nu_t \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \tilde{u}_k}{\partial x_k} \right)
\]

\[\nu_t = \frac{\mu_t}{\rho}\]

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

**RANS, time averaged**

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

**LES, space filtered**

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

---

Ch. 4 Section 4.7.3
All applications of interest have large Reynolds numbers:

\[ \text{Re}(\text{real}) = \frac{UL}{\nu} \]

Large Reynolds number -> large difference between largest and smallest spatial scales -> large number of grid points which we simply don't have in general!
Integral scale

Kolmogorov scale

= \frac{\text{Integral scale}}{\text{Re}^{3/4}}

In terms of resolution this implies that the number of points increases like $\text{Re}^{9/4}$
In a channel (diameter= 1cm,length 1cm, air at 300 K), number of points required:
- if the flow is laminar: 10 in each direction is enough (up to Re=2000, u=3 m/s)
- if the flow is faster than 3 m/s:
So: we **can not** 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** $\nu_t$
- dissipative schemes: add **numerical viscosity** $\nu_a$

Back up for a while: can we see ‘turbulence’ in our equations?

Yes....
\[ \frac{\partial}{\partial t} \rho u_j + \frac{\partial}{\partial x_i} \rho u_i u_j = - \frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_i} \]

This is the:
- non-linear term
- source of turbulence
- ennemy of all CFD codes

This is the:
- viscous term
- linear term
- damper of turbulence
- friend of all PhD students
So... what is turbulent viscosity?
It is the easiest (not the only one) solution when the Navier Stokes equations are averaged (in RANS) or filtered (in LES) to model the non linear terms due to the multiplications of correlations.

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

\[
\tau_{ij} = -\frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} + \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)
\]
Filter these equations in space (LES) or average them in time (RANS):

\[ f = \tilde{f} + f'' \]

Replace in momentum:

\[
\frac{\partial \tilde{\rho} \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i} (\tilde{\rho} \tilde{u}_i \tilde{u}_j) + \frac{\partial \tilde{p}}{\partial x_j} = \frac{\partial}{\partial x_i} [\bar{\tau}_{ij} - \bar{\rho} (\bar{u}_i \bar{u}_j - \tilde{u}_i \tilde{u}_j)]
\]

\[ \bar{\tau}_{ij} = \bar{u}_i \bar{u}_j - \tilde{u}_i \tilde{u}_j \]

HOW CAN WE MODEL THIS TERM?
We will model it in a form similar to the laminar term:

\[
\frac{\partial \rho \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i} (\rho \tilde{u}_i \tilde{u}_j) + \frac{\partial p}{\partial x_j} = \frac{\partial}{\partial x_i} [\tau_{ij} - \rho (\bar{u}_i \bar{u}_j - \tilde{u}_i \tilde{u}_j)]
\]

Laminar viscosity:

\[
\tau_{ij} = \rho \nu \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right)
\]

Turbulent viscosity:

\[
\tau_{ij} = u_i u_j - \tilde{u}_i \tilde{u}_j = -\nu_t \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right)
\]
\[
\frac{\partial \rho \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i} (\rho \tilde{u}_i \tilde{u}_j) + \frac{\partial p}{\partial x_j} = \frac{\partial}{\partial x_i} [\tau_{ij} - \rho (\tilde{u}_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j)]
\]

\[
\frac{\partial \rho \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_i} (\rho \tilde{u}_i \tilde{u}_j) + \frac{\partial p}{\partial x_j} = \bar{\rho} (\nu + \nu_t) \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right)
\]

Careful: all diagonal terms have been hidden in the pressure

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:

\[
\frac{f_i^{n+1} - f_i^n}{\Delta t} + u_i \frac{f_i^{n+1} - f_i^{n-1}}{2\Delta x} = \nu_a \frac{f_i^{n+1} + f_i^{n-1} - 2f_i^n}{\Delta x^2}
\]

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

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

But:

\[
 \frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = \nu_a \frac{\partial^2 f}{\partial x^2}
\]
Upwind schemes are NOT a solution:

\[
\frac{f_i^{n+1} - f_i^n}{\Delta t} + u_i \frac{f_i^n - f_i^{n-1}}{\Delta x} = o(\Delta x)
\]

Using:

\[
f_{i-1}^n = f_i^n - \Delta x \frac{\partial f}{\partial x} + \frac{1}{2}\Delta x^2 \frac{\partial^2 f}{\partial x^2}
\]

An upwind scheme is like a centered scheme with a numerical viscosity equal to \(\frac{1}{2} u \Delta x\)
Turbulent flow solvers combine:

- turbulent \textit{viscosity} $\nu_t$
- numerical \textit{viscosity} $\nu_a$

to allow the code to run. But at which price?

In practice, the Reynolds number seen by the code is:

$$Re(num) = \frac{UL}{(\nu + \nu_t + \nu_a)}$$

which is much \textbf{smaller} than the Reynolds of the flow. It can even be smaller than the \textbf{critical} Reynolds number to have turbulence in this flow.

$\implies$ 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)
RANS: since the flow is so viscous, might as well use numerical viscosity

\[
\text{Re} = \frac{UL}{(\nu + \nu_t + \nu_a)} \ll \text{Re(real)}
\]
CODE DNS: nothing more than the laminar viscosity

\[ \text{Re}(\text{num}) = \frac{UL}{(\nu + \nu_t + \nu_a)} = \frac{UL}{\nu} \]
A good LES code: turbulent viscosity reduced and limited numerical viscosity

\[ \text{Re} = \frac{UL}{(\nu + \nu_t + \nu_a)} > \text{Re}(\text{crit}) \]
DANGERS of RANS codes used for LES:

\[ \text{Re}(\text{num}) = \frac{UL}{(\nu + \nu_t + \nu_a)} \text{ too small} \]
Classical test: remove the turbulent viscosity

And see if something changes...
If nothing changes, this is not LES.
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 must propagate vortices and acoustic waves. This impacts our choices for numerical techniques:

- How do codes propagate physical waves? NOT SO WELL...
- Do codes propagate ‘other’ non-physical waves? YES...

NEED TO START DISCUSSING:

- Dissipation: waves are attenuated by the numerical scheme
- Dispersion: the speed of waves is affected by the numerical scheme and depends on its wavelength
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 (2\textsuperscript{nd} 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 $c$:

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

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 $c$:

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

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

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 \( \omega \)):

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

\[ j^2 = -1 \]

\[ \lambda = 1/\omega \]
What does a second order code do? Suppose we have perfect time advancement:

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

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

Replacing $u$ by $\nu(t)$ in Eq. (1) leads to:

$$\frac{\partial \nu}{\partial t} = -cj \nu \sin(2\pi \omega \Delta x)/\Delta x$$

Or:

$$\nu(t) = \nu(0) \exp(-cj \frac{\sin(2\pi \omega \Delta x)}{\Delta x} t)$$

The numerical solution for this problem will be:

$$u(x,t) = \nu(0) \exp\left[2j\pi \omega (x - c \frac{\sin(2\pi \omega \Delta x)}{2\pi \omega \Delta x} t)\right]$$
Comparing the exact and the numerical solution:

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

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

The numerical scheme is dispersive: the speed is not right BUT the amplitude of the signal is right===> no damping is introduced by the centered scheme. Good for LES/DNS

This property is true of all centered schemes but not of upwind schemes. For the first order upwind scheme, for example, the numerical solution would be:

\[
 u(x,t) = v(0) \exp\left(-\frac{ct}{\Delta x}(1-\cos2\pi\omega\Delta x)\right) \exp\left[2j\pi\omega(x-c\frac{\sin(2\pi\omega\Delta x)}{2\pi\omega\Delta x}t)\right]
\]

Numerical dissipation factor!
Moreover, for the centered scheme, comparing now the speeds:

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

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

The numerical scheme makes the flow ‘dispersive’; different wavelengths \( \omega \) are propagated at different speeds \( c(\omega) \):

\[
\frac{c(\omega)}{c} = \frac{\sin(2 \pi \omega \Delta x)}{2 \pi \omega \Delta x}
\]

\( \omega \Delta x = \Delta x / \lambda \)
This is not good news for second-order schemes: they do not propagate waves at the right speed as soon as the resolution (i.e., the number of points per wavelength $\lambda/\Delta x$) is not very high. Higher order schemes do better.
A simple example in one dimension
• Second order centered scheme
• Fourth order Runge Kutta time integration

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

Black: code
Cyan: exact

Periodic boundaries
2D Vortex convection

Convecting vortices is the basic feature of LES

\[ U_x = 100 \text{m/s} \]
\[ p - p_0 = -\frac{\rho \Gamma^2}{2R_c^2} e^{-\frac{x^2+y^2}{R_c^2}} \]
\[ \nu = \frac{\Gamma}{R_c^2} xe^{-\frac{x^2+y^2}{2R_c^2}} \]
\[ u = -\frac{\Gamma}{R_c^2} ye^{-\frac{x^2+y^2}{2R_c^2}} \]

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

Moureau et al, JCP 2005
30 by 30 box: It contains MANY vortices
FOR LES, THINGS ARE EVEN MORE COMPLICATED:
- At the cut off scale (for sizes of the order of a few mesh size), there is ALWAYS activity and vorticity. In an LES, we have many vortices which are resolved with only a few points.
- Worse news: we use these scales to compute turbulent viscosity
- Need to be accurate at all scales and not only at the largest ones

These vortices exist in the LES especially near $k_c$

These vortices exist in the real world but not in the LES
Periodic on all sides

Vortex leaves

And comes back
Comparing three schemes:
• 2nd order Lax Wendroff in AVBP (code CERFACS)
• 2nd order CFX
• 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 $\text{CFL} = c \frac{\Delta t}{\Delta x} = 0.7$

If you want to do this with your code: type ‘CERFACS CO-VO elearning’ on google
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 resources

- Case description and results (fr)
- Test case description
- Results of the test case
First code:
- AVBP
- elsA
- Leslie
- NTMIX
- OpenFoam
- YALES

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

Select the result:
- Contour of \( u-U_0 \)

\[ t = 10 \frac{L}{U_0} \]
Low pressure in vortex center (analytical solution)
Velocity $y \ [m/s]$
Velocity $y \text{ [m/s]}$
Results after ten turn over times

CFL = 0.7
Relative pressure [Pa]
Relative pressure [Pa]
Velocity \( y \, [\text{m/s}] \)
Velocity y [m/s]
Influence of time step: the famous implicit/explicit controversy

Essential question for LES:
➢ explicit codes use a CFL-limited time step: no question on how to choose the time step, just take the max possible value for stability

➢ implicit codes can use any time step for stability. BUT for precision, the question is open! The fact that the code can be run does not mean that the result is good ....

➢ In the present test case, an implicit code was used with various CFL up to 10 to 40 corresponding to convective CFL numbers from 1 to 5.

→ Results after 3 turn around times
Relative pressure [Pa]
IMPLICIT AND LES ARE A COMBINATION WHICH STILL REQUIRE SOME WORK!
Conclusion: (1) small time steps are needed and (2) high-order 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!
CODES FOR LES

91

FLUENT
STAR CCM
OPENFOAM

N3S
KIVA
C3D

DISCONTINUOUS GALERKIN (ARGO, CENAERO)
SPECTRAL DIFFERENCES (JAGUAR, CERFACS)

AVBP (CERFACS)
YALES 2 (CORIA)
CHARLES (CTR)

CODES DNS

1 2 3 4 5 6
PRECISION ORDER
Structured meshes can not be a good approach for combustion chambers… PROBLEM FOR LES!

Care is needed to propagate physical waves (vortex and acoustic waves) at the right speed and amplitude.

But there are worse news: ‘other’ waves are created and propagated by codes. These waves travel in ‘packets’.

- Vichnevetski and Bowles 1982, SIAM studies in Applied Mechanics
- Sengupta 2004 Fundamentals of Computational Fluid Dynamics
- Poinsot Veynante TNC 2011
The group velocity of wiggles

In dispersive flows, the group velocity $Vg(\omega)$ of wave packets differ from $c(\omega)$… Illustration for the advection equation: $\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$

Typical wave packet: envelope (long wave length $\Lambda$) + high frequency wave (short wave length $\lambda$)
Numerical wave packet:
• Wavelength=2 $\Delta x$
• Propagating in the wrong direction!

Initial condition:
A typical wave packet
Do we understand this? See theory in waves books. For fluid flow and centered schemes, see Vichnevetski and Bowles 1982, 1986 or Sengupta 2004. The main result is that the group velocity $V_g$ of wave packets is given by:

$$
\frac{V_g(\omega)}{c} = \frac{d}{d\omega} \left[ \frac{\omega c(\omega)}{c} \right]
$$

For the second-order scheme:

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

$$
\frac{V_g(\omega)}{c} = \cos(2\pi\omega\Delta x)
$$

$$
\lambda/\Delta x = 1/(\omega\Delta x)
$$
Codes propagate two types of waves depending on the sign of Vg:

- **Physical waves called ‘p’ (long wave):** they go with the flow
- **Numerical waves (short wave) called ‘q’ going against the flow**
For the centered second order scheme, the group velocity of wiggles (waves with a wavelength equal to 2 $\Delta x$) is

$$\frac{V_g(\omega)}{c} = -1$$

For higher order schemes, this velocity gets ... worse. For example, a compact sixth-order scheme (Baum et al 1994):

$$\frac{V_g(\omega)}{c} = -4.3$$

Here, wiggles propagate upstream at -4.3 times the sound speed.
Why do we care about ‘q’ waves? They perturb the flow. They also reflect into ‘p’ waves at boundaries!
Numerical ‘q’ waves are difficult to avoid:
1/ They can be created by stiff initial conditions, even if the user does not see them
2/ Numerical ‘q’ waves are also generated when physical ‘p’ waves interact with boundaries.
As a consequence, numerical waves can induce false numerical resonances (see Buelle and Huerre CTR 1988, Poinsot and Lele 1992, Poinsot and Veynante 2005):

Will happen even in supersonic flows
Some may wonder WHY we discuss this: they have never seen this problem in commercial codes!!
This is due to the viscosity of RANS codes. Example: non dissipative scheme (left) and typical RANS scheme (right)

- **NON DISSIPATIVE SCHEME** (LES or DNS)
- **DISSIPATIVE SCHEME** (RANS)
Conclusion:
- LES is good
- LES is difficult and expensive
- Those who propose LES at the cost of RANS, with RANS codes and options (implicit, low order) are probably not saying the truth....

=> Massively parallel machines are the only solution to make LES affordable