Simulation sur les nouvelles architectures massivement parallèles : un exemple en mécanique des fluides

A. Cadiou, M. Buffat, L. Le Penven, J. Montagnier

Laboratoire de Mécanique des Fluides et d'Acoustique
CNRS, Université Lyon I, École Centrale de Lyon, INSA de Lyon

LyonCalcul

Part I

Présentation de la problématique physique

Numerical simulation in fluid mechanics

Le Laboratoire de Mécanique des Fluides et d'Acoustique

Domaines de recherche

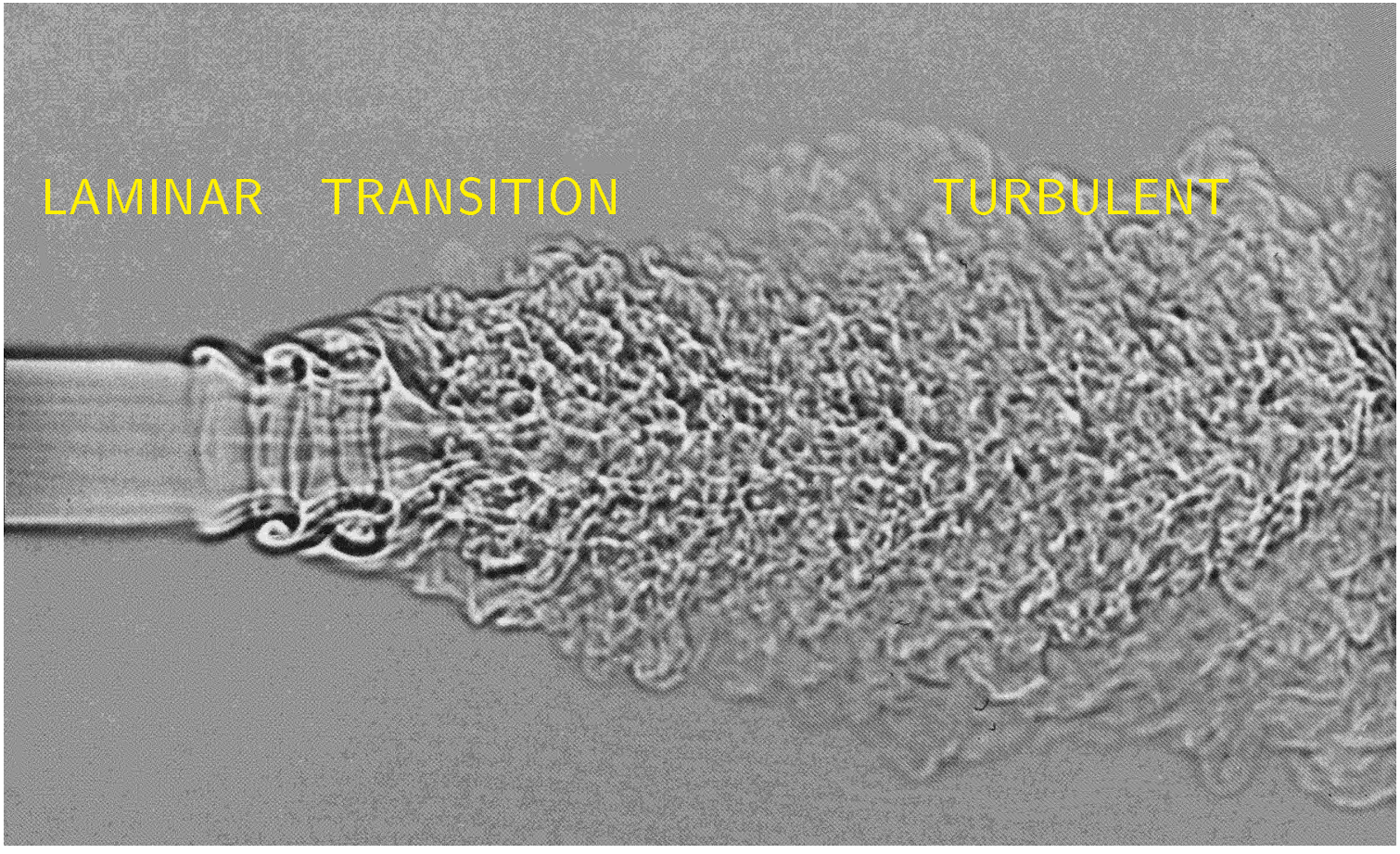
- Physique et modélisation de la turbulence,
- instabilités hydrodynamiques,
- écoulements diphasiques,
- mécanique des fluides environnementale,
- aérodynamique interne,
- phénomènes thermiques couplés,
- aéroacoustique,
- propagation acoustique,
- méthodes de résolution numérique des équations de Navier-Stokes,
- contrôle actif ou passif des écoulements,
- microfluidique.

Le Laboratoire de Mécanique des Fluides et d'Acoustique

Secteurs d'application et partenaires industriels

- **Aéronautique et Spatial :** SAFRAN/SNECMA, CNES, ONERA, Turbomeca, EADS, Dassault
- **Automobile** Renault, Volvo, Valeo, Delphi, IAV, CNRT, PO, Pôles LUTB, Moveo, ID4car
- **Environnement - Bruit** DGA, CNES, ONERA, SNCF, Eurocopter, EADS, PSA, CEA DAM
- **Environnement - atmosphérique, hydrologique** EDF, INERIS, CEA, CEMAGREF, Total, Londres, Turin, Air Parif...
- **Génie des procédés, Energie** CEA, Aventis, Geoservices, Andritz...

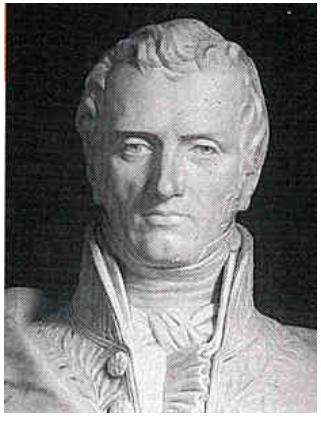
Laminar/turbulent transition



Jet flow: Landis-Shapiro, ombroscopy

TURBULENT = 3D, multiple space-time scales, seemingly random behaviour, for practical applications : details are not predictable.

Navier-Stokes equations



Claude Louis NAVIER

(1785-1836)

For incompressible, homogeneous fluid:

velocity $\boldsymbol{V}$, pressure P

$$\partial_t \boldsymbol{V} + \boldsymbol{V}.\nabla \boldsymbol{V} = -1/\rho \ \nabla p + \nu \Delta \boldsymbol{V}$$
$$\nabla.\boldsymbol{V} = 0$$

density ρ, viscosity ν

+ initial and boundary conditions

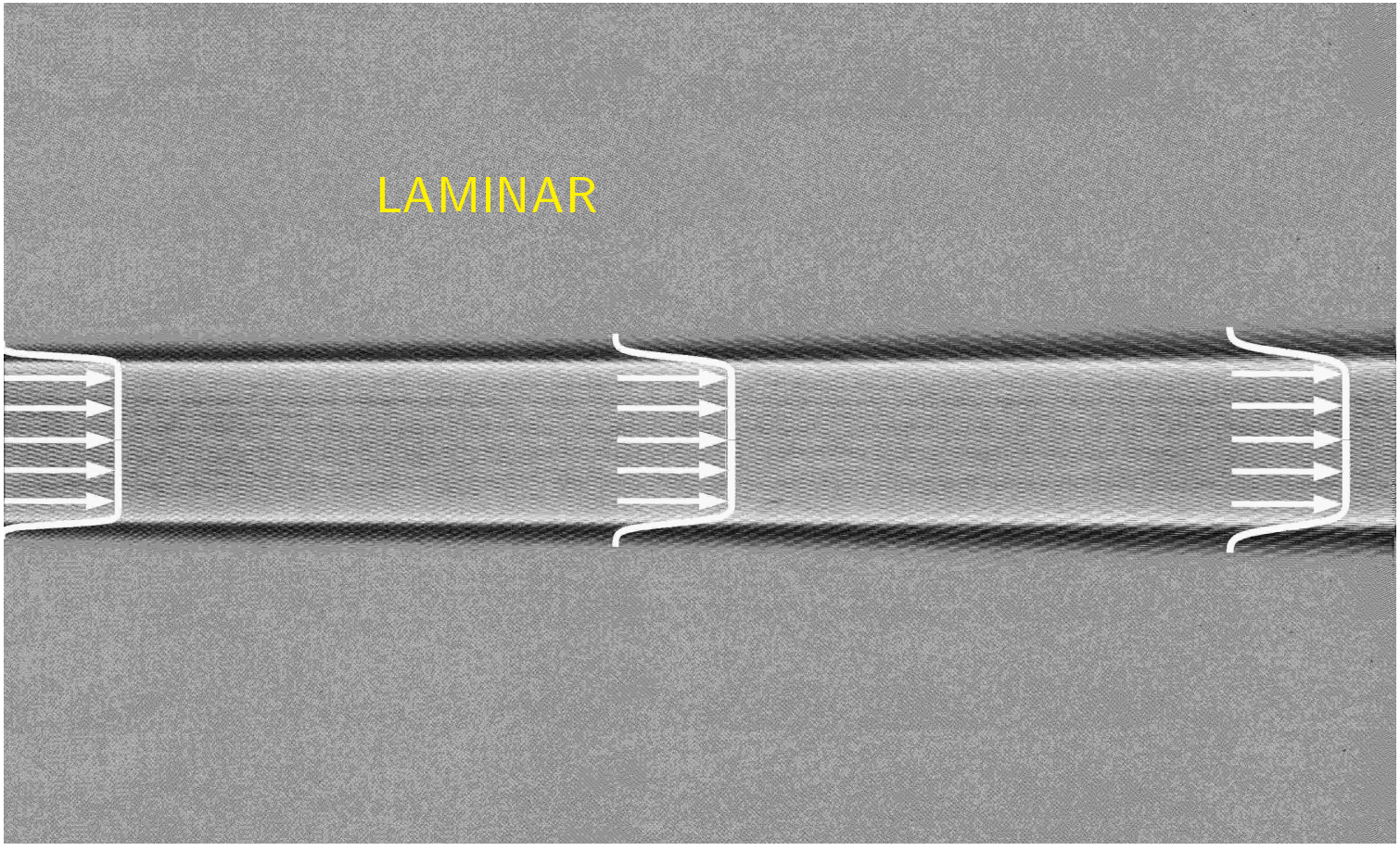
$$\mathcal{NS}(\boldsymbol{V}) = 0$$

Well-known equations, but open mathematical problems ....

George Gabriel STOKES

(1819-1903)

Laminar/turbulent transition

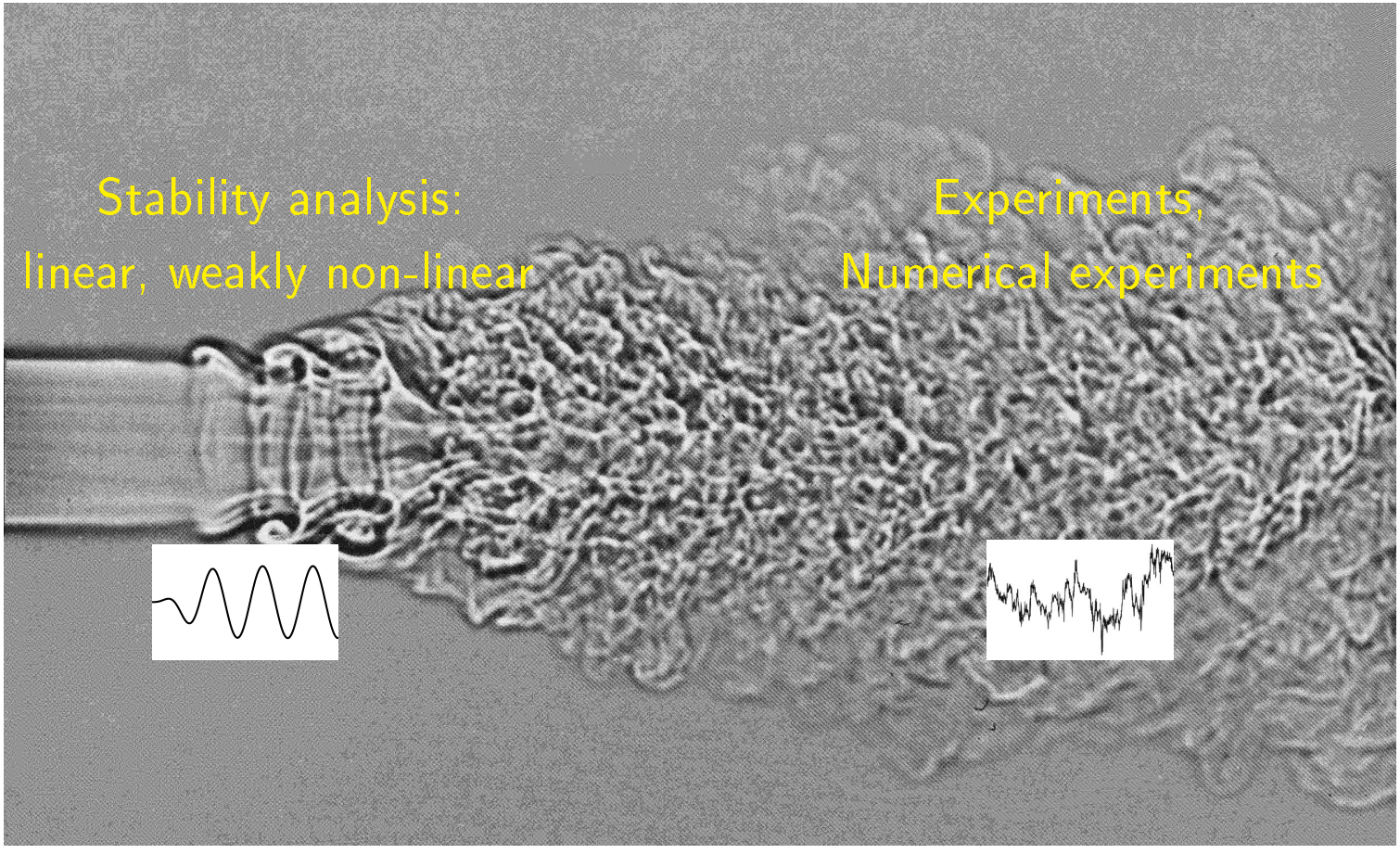


A "laminar" solution exists : $\mathcal{NS}(\boldsymbol{V}_0) = 0$.

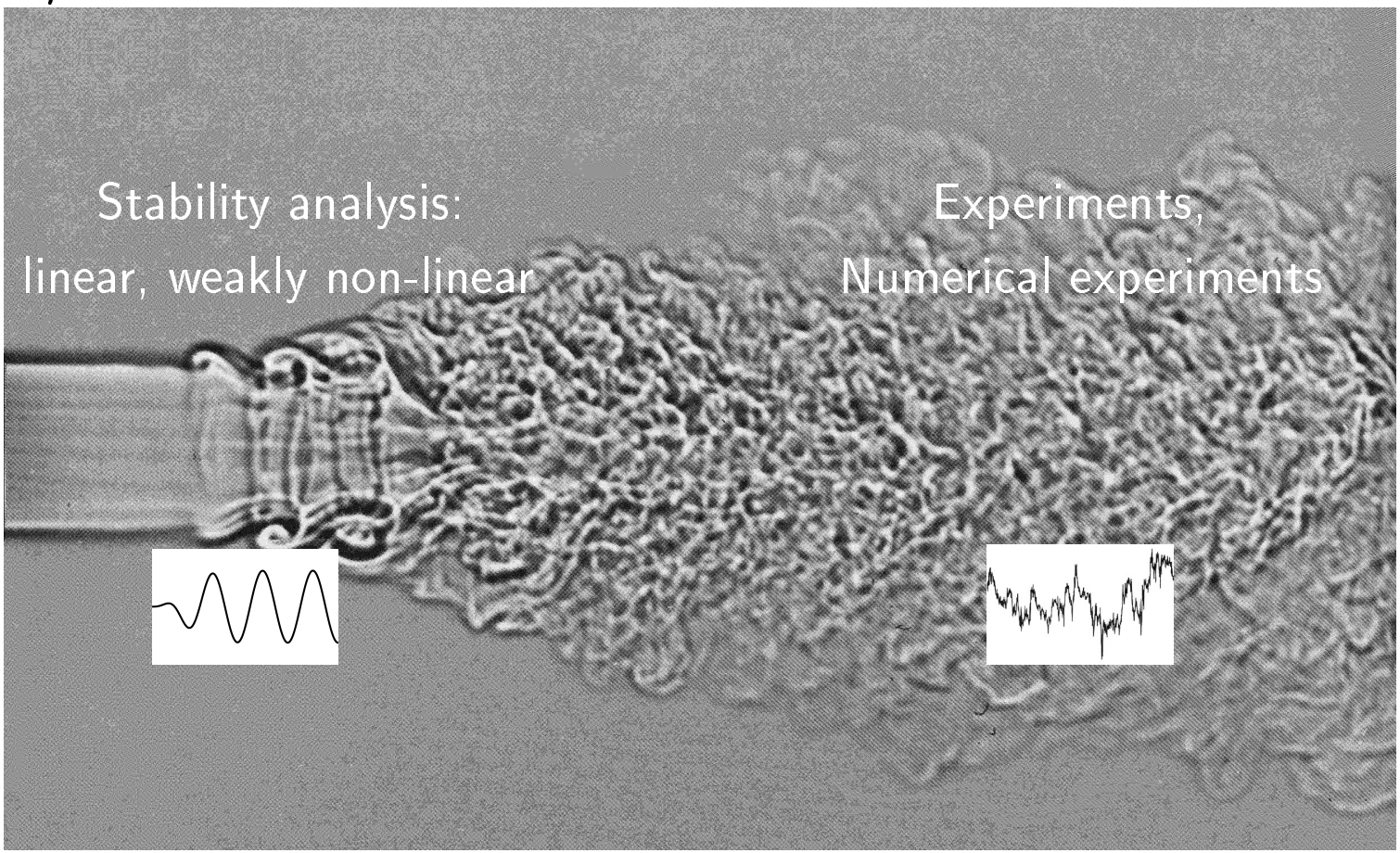
Stability of $\boldsymbol{V}_0$ under perturbations? $\boldsymbol{V} = \boldsymbol{V}_0 + \boldsymbol{v}$

Linear stability problem : $\frac{d\mathcal{NS}}{d\boldsymbol{V}}|_{\boldsymbol{V_0}}\boldsymbol{v} = 0$ shows that $|\boldsymbol{v}| \nearrow$: $\boldsymbol{V}_0$ unstable.

Laminar/turbulent transition



Laminar/turbulent transition



Identify scenarii for transition to turbulence,

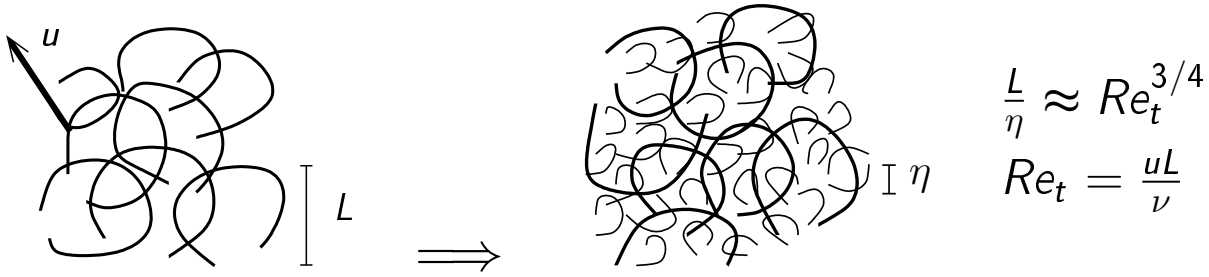
Explain why turbulence is self-sustaining as it convects downstream,

Design statistical models for well-developped turbulence.

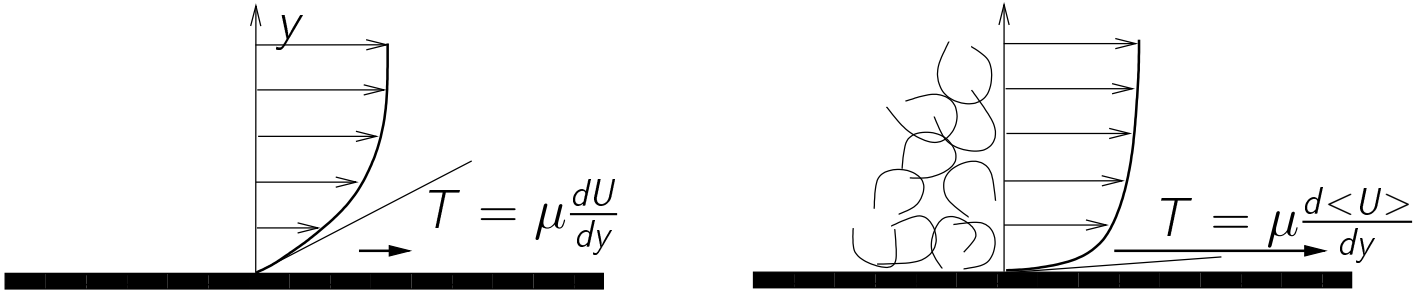
$\rightarrow$ Equations for the averaged variables: $< \boldsymbol{V}(\boldsymbol{x}, t) >$, ...

Practical consequences of turbulence

- Promote energy transfer to smaller scales and increase dissipation.



- Increase spatial transfer for mass, momentum, energy.
  - momentum transfer on solid wall ⇒ force,

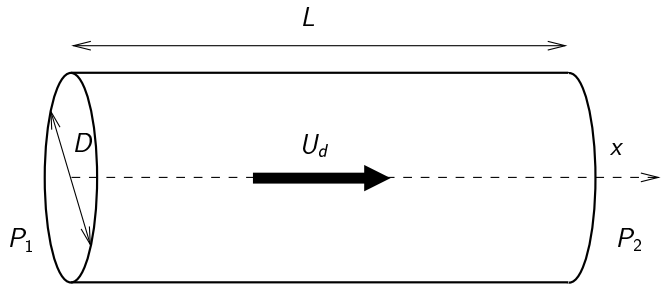


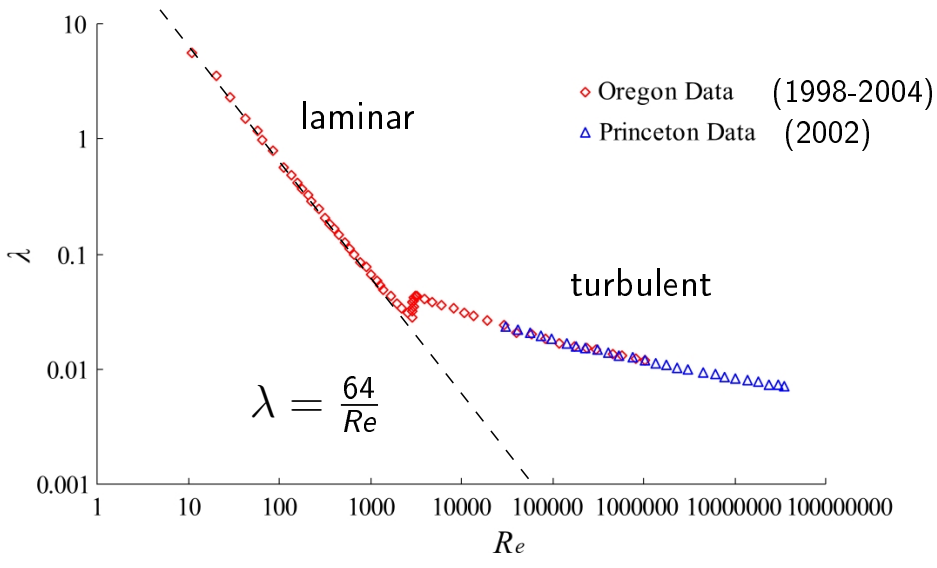
- Consequences may be favorable or not, depending on applications.

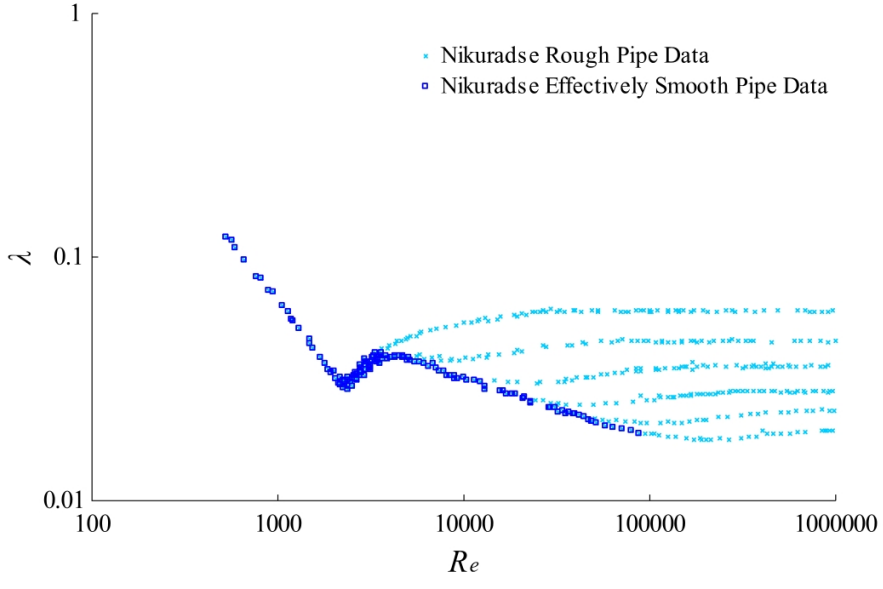
Example : pressure loss in circular pipes

$$P_2 - P_1 = \lambda \frac{L}{D} \frac{\rho U_d^2}{2}, \; Re = \frac{U_d D}{\nu}$$

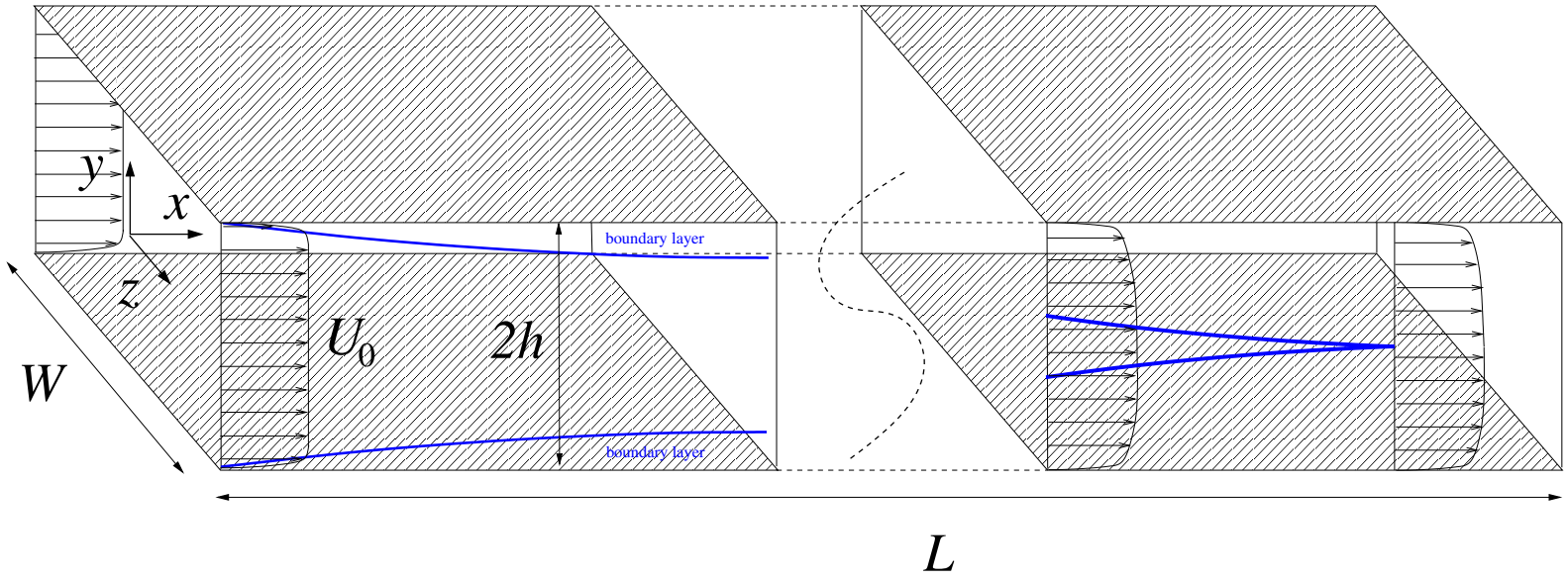
Pressure loss coefficient λ





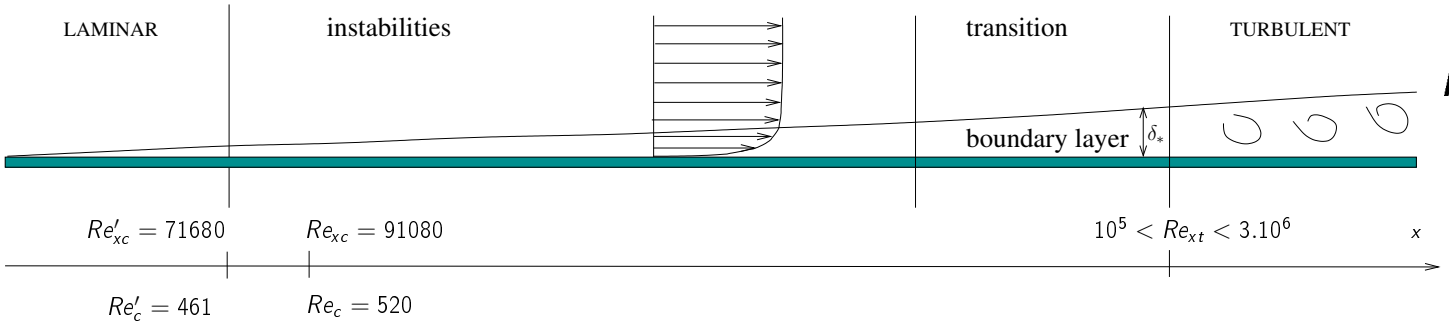


The present study



- Developping flow in a plane channel (2 parallel solid walls, ΔP).
- For large U_0, turbulent transition inside the boundary layers.
- Evolves towards well-developped state ($\frac{d<.>}{dx} = 0$).
- Numerical simulations :
  - Long geometries ($L \sim 100h$), grid refinement near the walls.
  - Effects of entrance perturbations.

Transition in boundary layers



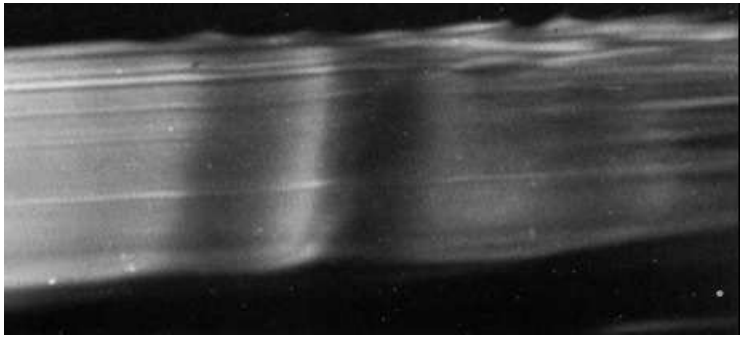
$$Re_x = \frac{U_\infty x}{\nu}$$

$$Re = \frac{U_\infty \delta_*}{\nu}$$

M. Matsubara et P.H. Alfredsson

- moderate level of perturbation
- "streaks" (3D)

→ "by-pass" transition (lower Re_x)



H. Werlé

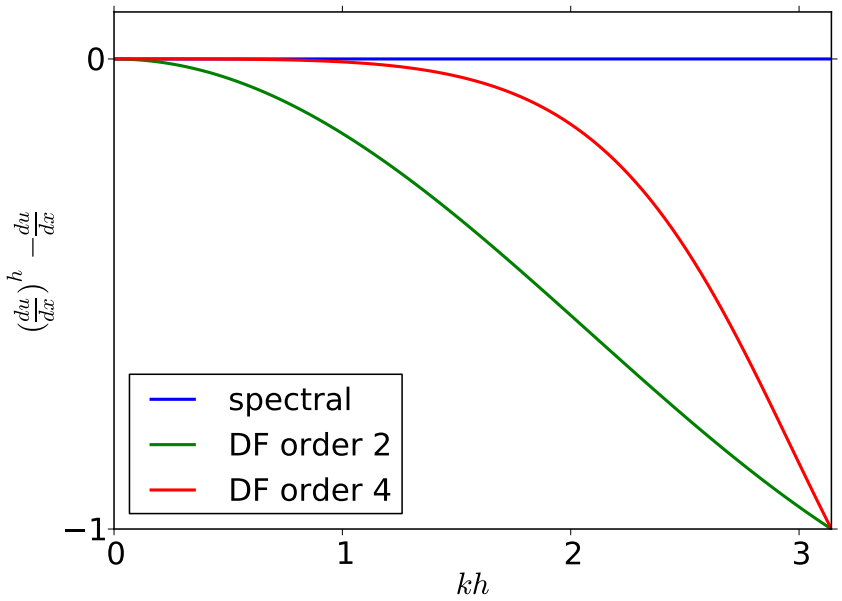
- low level of perturbation
- Tollmien-Schlichting waves (2D)

→ transition

Numerical method

- Numerical experiment: need to resolve the flow at all scales .
- As $\left(\frac{L}{\eta}\right)^3 \sim Re^{9/4}$ $\nearrow$, increasingly stringent condition for turbulence.
- Spectral methods are attractive, due to their high spatial accuracy.

- Spatial derivatives are calculated exactly.
- Exponential convergence for smooth solutions (faster than FE, FD ...).



- Since the 70's, extensively applied to simulation of turbulent flows but, their implementation on new HPC must be carefully considered.

Partie II

Mise en oeuvre sur les plateformes HPC

Adaptation du code NadiaSpectral aux nouvelles architectures

Incompressible Navier-Stokes equations

Governing equations

$$\begin{aligned} \frac{\partial U}{\partial t} + U.\nabla U &= -\nabla p + \frac{1}{Re}\Delta U \\ \nabla.U &= 0 \\ & U(t=0) = U_0 \\ & U|_{\partial\Omega} \end{aligned}$$

Galerkin formulation using an orthogonal decomposition of the velocity

$$U = U_{OS}(U.e_y) + U_{SQ}((\nabla \times U).e_y)$$

spectral approximation

$$U(t,x,y,z) = \sum_i \hat{U}_i(t)\alpha_i(x,y,z)$$

Numerical method

Spectral coefficients with $N_x \times N_y \times N_z$ modes

$$U(x,y,z,t) = \sum_{m=-N_x/2}^{N_x/2} \sum_{p=-N_z/2}^{N_z/2} \left[\sum_{n=0}^{N_y-1} \alpha_{OS,n}^{mp} \hat{U}_{OS,n}^{mp} + \sum_{n=0}^{N_y-1} \alpha_{SQ,n}^{mp} \hat{U}_{SQ,n}^{mp} \right]$$

- Optimal representation of a solenoidal velocity field
- Elimination of the pressure

Spectral approximation

- Fourier-Chebyshev approximation with a Galerkin formulation
- Time integration with Crank Nicolson / Adams Bashforth scheme

Resolution of coupled systems for non-linear advective terms

At each time step, $N_x \times N_z$ linear systems of dimension $N_y - 3$ are solved

$$A_{OS}^{mp} \alpha_{OS}^{mp} = b_{OS}^{mp}$$

$$A_{SQ}^{mp} \alpha_{SQ}^{mp} = b_{SQ}^{mp}$$

A_{OS}^{mp} and A_{SQ}^{mp} are sparse matrices (resp. 7D and 5D)
$b^{mp} = b^{mp}(\alpha_{SQ}^{mp}, \alpha_{OS}^{mp})$
contains non-linear terms
(convolution products coupling every α_n^{mp})

$\Rightarrow$ b is calculated in physical space
$\Rightarrow$ must perform FFTs in each direction

Per iteration, i.e. at each time step,
27 FFT (direct or inverse) are performed

Challenge: from 100 to 10000 cores

Present configuration: computational domain size $150 \times 2 \times 3.2$

- $17280 \times 192 \times 384$ modes (~ 1.3 billion of modes)
- travel 1 length with it=300000 iterations. (~ 16 millions of FFT)

Memory constraint

- $N = N_x \times N_y \times N_z$, with N very large
  - large memory requirement ($\sim 520 Go$)
  - BlueGene/P 0.5 Go per core $\Rightarrow \sim 1000$ cores needed
- $N_x >> N_y, N_z$, elongated in one direction
  - 1D domain decomposition $\Rightarrow$ limited to ~ 100 cores
  - can only simulate a 10 times shorter channel length

Wall clock time constraint

- CPU time $100h \sim 4$ days on ~ 8000 cores
  - with 100 cores (if possible), 80 times slower, $8000h \sim 1$ year

Outline

Implementation on HPC platforms

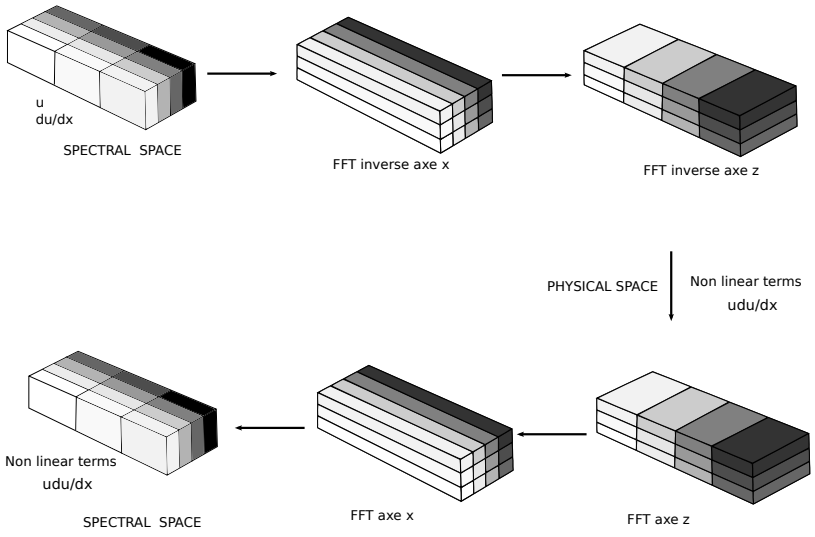
- MPI strategy to scale from $O(100)$ to $O(10\,000)$ core
- Hybrid strategy to migrate on many-core platform
- Additional constraint for optimization
- Data manipulation during simulation
- Data manipulation for analysis and post-treatment

Outline

Implementation on HPC platforms

- MPI strategy to scale from $O(100)$ to $O(10\,000)$ core
- Hybrid strategy to migrate on many-core platform
- Additional constraint for optimization
- Data manipulation during simulation
- Data manipulation for analysis and post-treatment

2D domain decomposition



- Chebyshev between walls
  (y direction, $N_y + 1$ modes)
- 2D FFT in periodical directions
  (x direction and z direction)
- Transpose from
  $y-$pencil to $x-$pencil,
  $x-$pencil to $z-$pencil and back

Increase the number of MPI processes and reduce wall clock time

- 1D decomposition: MPI $\leq N_y$

  $17280 \times 192 \times 384 \rightarrow$ max. of MPI processes: nproc=192
- 2D decomposition: MPI $\leq N_y \times N_z$

  $17280 \times 192 \times 384 \rightarrow$ max. of MPI processes: nproc=73 728
- Perform data communications and remapping
- Choose data rearrangement to limit the increase in communications

Outline

Implementation on HPC platforms

- MPI strategy to scale from $O(100)$ to $O(10\,000)$ core
- Hybrid strategy to migrate on many-core platform
- Additional constraint for optimization
- Data manipulation during simulation
- Data manipulation for analysis and post-treatment

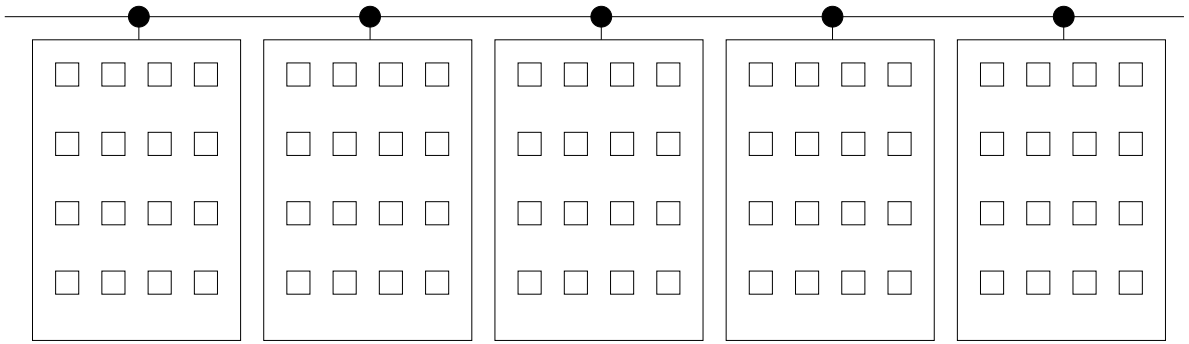
Constraints related to modern many-cores platforms

Tendancy towards many-cores platforms

- Limited number of nodes
- Increase of cores per node (BlueGene/P = 4 - SuperMUC = 16)

Increase MPI processes

- allow larger number of modes within the same wall clock time
- limit the memory available per processus



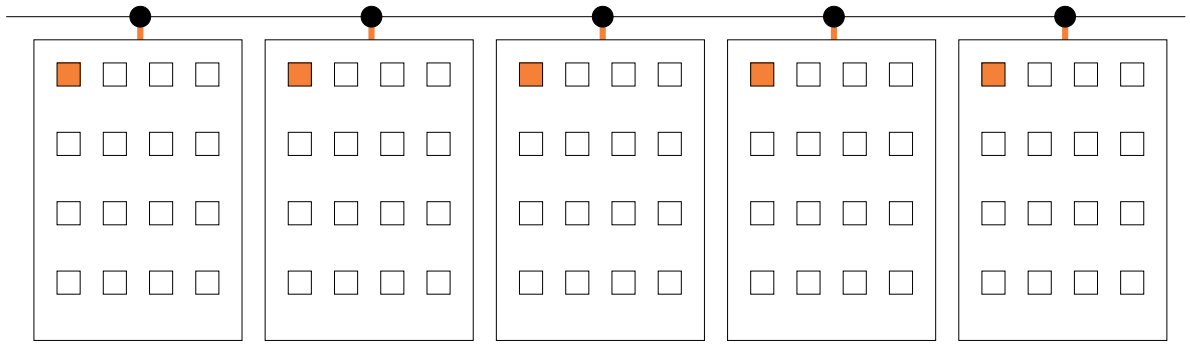
Constraints related to modern many-cores platforms

Tendancy towards many-cores platforms

- Limited number of nodes
- Increase of cores per node (BlueGene/P = 4 - SuperMUC = 16)

Increase MPI processes

- allow larger number of modes within the same wall clock time
- limit the memory available per processus



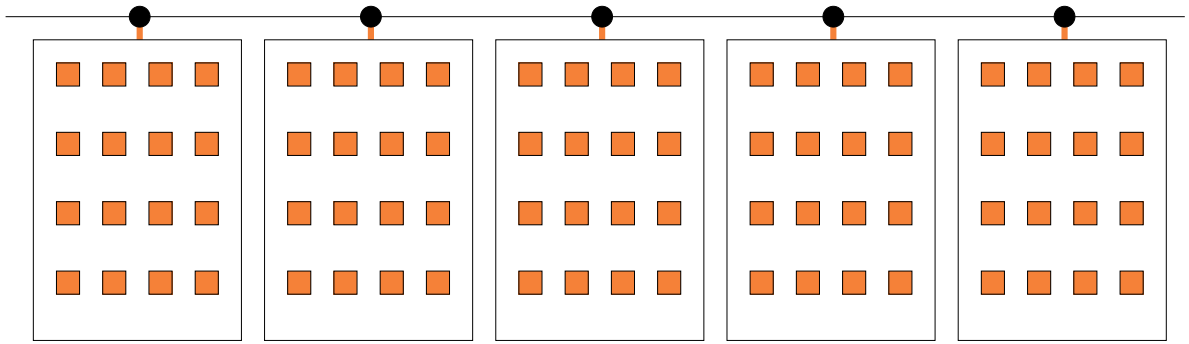
Constraints related to modern many-cores platforms

Tendancy towards many-cores platforms

- Limited number of nodes
- Increase of cores per node (BlueGene/P $= 4$ - SuperMUC $= 16$)

Increase MPI processes

- allow larger number of modes within the same wall clock time
- limit the memory available per processus



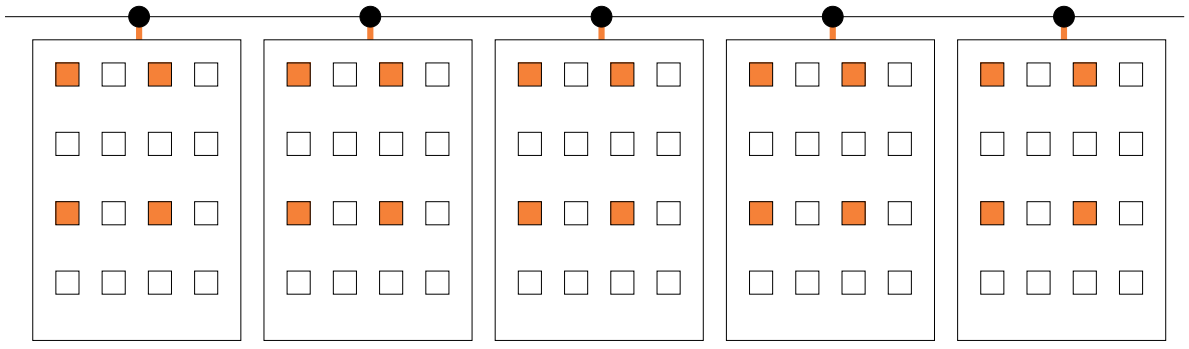
Constraints related to modern many-cores platforms

Tendancy towards many-cores platforms

- Limited number of nodes
- Increase of cores per node (BlueGene/P = 4 - SuperMUC = 16)

Increase MPI processes

- allow larger number of modes within the same wall clock time
- limit the memory available per processus



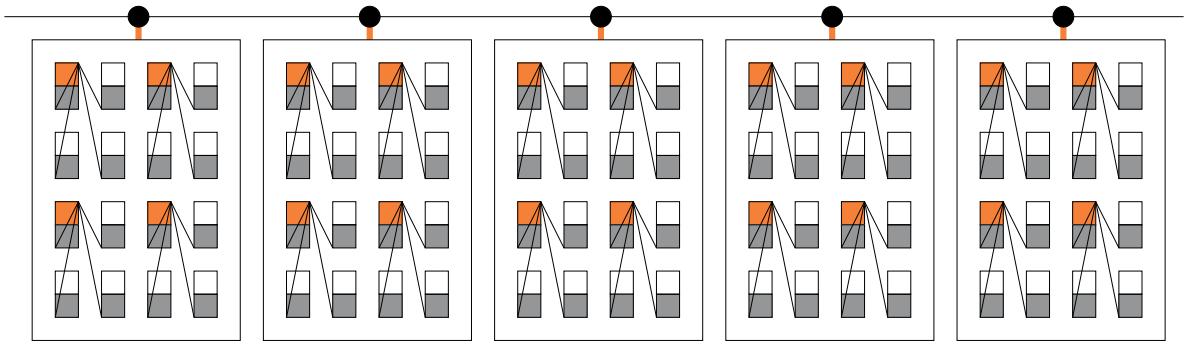
Constraints related to modern many-cores platforms

Tendancy towards many-cores platforms

- Limited number of nodes
- Increase of cores per node (BlueGene/P $= 4$ - SuperMUC $= 16$)

Increase MPI processes

- allow larger number of modes within the same wall clock time
- limit the memory available per processus



Hybrid OpenMP/MPI

Suitable for recent many-core platforms

- Reduces the number of MPI processes
  - Reduces the number of communications
  - Increases the available memory size per node

Modification for many threads

- Time of thread creation exceeds inner loop time execution
- Implementation of explicit creation of threads
- Recover full MPI performance and allow further improvment.

Outline

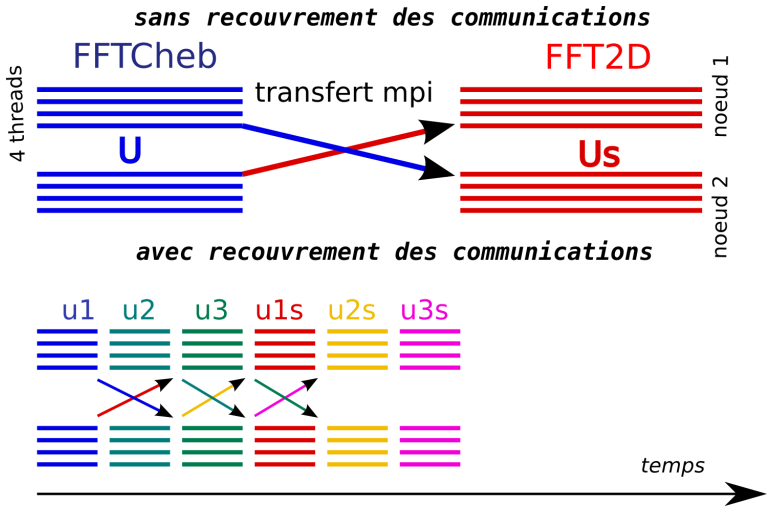
Implementation on HPC platforms

- MPI strategy to scale from $O(100)$ to $O(10\,000)$ core
- Hybrid strategy to migrate on many-core platform
- Additional constraint for optimization
- Data manipulation during simulation
- Data manipulation for analysis and post-treatment

More than domain decomposition ...

Tasks parallelization : mask communications by execution time

- reduces by 20% time per iteration
- less loss in communications - waist $\sim$ 10%



Placement of processus

- specific on each platform, optimize interconnection communications
- avoid threads to migrate from one core to another

  example: TORUS versus MESH in BlueGene/P platform - 50% faster

Outline

Implementation on HPC platforms

- MPI strategy to scale from $O(100)$ to $O(10\,000)$ core
- Hybrid strategy to migrate on many-core platform
- Additional constraint for optimization
- Data manipulation during simulation
- Data manipulation for analysis and post-treatment

Problems related to the very large calculations
Data manipulation during simulation

Data Input/Output and storage

- Large data
  - case $17280 \times 192 \times 384$: one velocity field ~ 30 Go

$\Rightarrow$ Use parallel IO (each processes writes its own data)

- Large amount of file, could rapidly exceeds inode or quota limit
  - case $17280 \times 192 \times 384$: statistics 8 fields on ~ 8000 processes (~ 300 Go)
  - write ~ 150 times data during travel length ($L_x = 150$) (disk quota ~ 3 To)

$\Rightarrow$ wrap in tar archive file or separated directory

- Manage the large amount of data generated

$\Rightarrow$ Use of predefined parallel format (VTK, HDF5, NetCFD, ...)

beware not to add useless complexity for regular structured data

$\Rightarrow$ Optimize data transfert between platform

HPC simulations require every layer of HPC ressources

Tier-0, PRACE

1. JUGENE and JUQUEEN, Jülich, Germany
2. CURIE, Bruyères-le-Châtel, France
3. SuperMUC, Garching, Germany

Tier-1, GENCI

1. IDRIS, Orsay
2. CINES, Montpellier
3. TGCC, Bruyères-le-Châtel

Tier-2, Fédération Lyonnaise de Modélisation et Sciences Numériques

P2CHPD, la Doua
Many thanks to Christophe Péra

Problems related to the very large calculations
Data manipulation after simulation

Data processing

- Part of the analysis is performed during simulation
- Part of it is explored afterwards
  - Entails spatial derivation, eigenvalues evaluation ...
  - Preserve the same accuracy than in the simulation
  - Should be interactive and when ready on batch mode

$\Rightarrow$ Should be done locally because of data storage

$\Rightarrow$ Must be parallel, but on a smaller scale

Problems related to the very large calculations
Data manipulation after simulation

Data 3D visualization

- Cannot be performed directly on HPC platforms
- Preserve accuracy of the simulation
- Should be interactive and when ready on batch mode

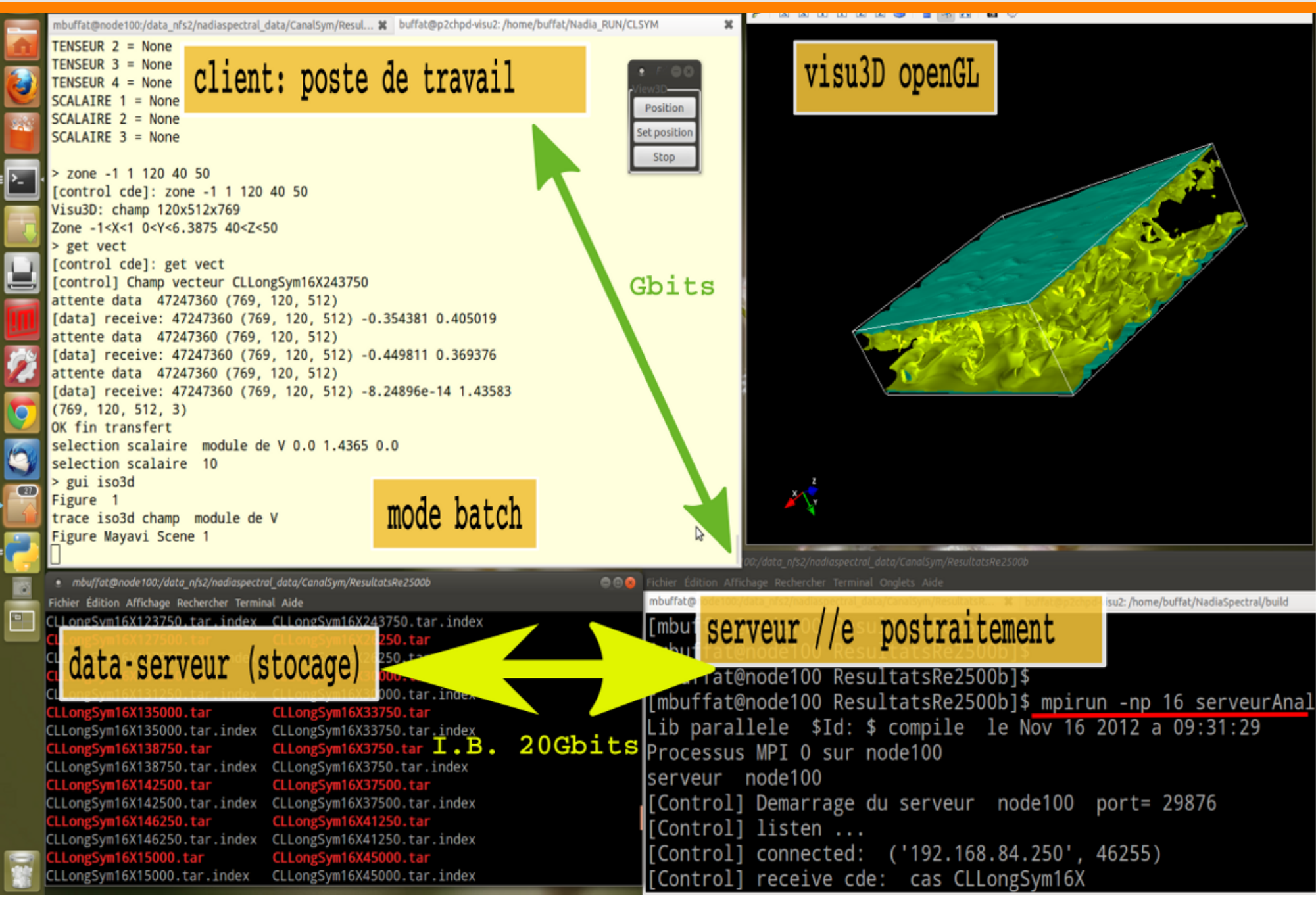
$\Rightarrow$ Must be done with remote access

$\Rightarrow$ Must be parallel, but on a smaller scale

Problems related to the very large calculations
Data manipulation after simulation

Organization

- Data transferred on a local data server - with rapid network access
- Processing is performed on a Mecocenter platform
- Client/Server system to access from laptop


client: poste de travail
visu3D openGL
Position
Set position
Stop
Gbits
mode batch
data-serveur (stocage)
I.B. 20Gbits
serveur //e postraitement
TENSEUR 2 = None
TENSEUR 3 = None
TENSEUR 4 = None
SCALAIRE 1 = None
SCALAIRE 2 = None
SCALAIRE 3 = None
> zone -1 1 120 40 50
[control cde]: zone -1 1 120 40 50
Visu3D: champ 120x512x769
Zone -1<X<1 0<Y<6.3875 40<Z<50
> get vect
[control cde]: get vect
[control] Champ vecteur CLLongSym16X243750
attente data 47247360 (769, 120, 512)
[data] receive: 47247360 (769, 120, 512) -0.354381 0.405019
attente data 47247360 (769, 120, 512)
[data] receive: 47247360 (769, 120, 512) -0.449811 0.369376
attente data 47247360 (769, 120, 512)
[data] receive: 47247360 (769, 120, 512) -8.24896e-14 1.43583
(769, 120, 512, 3)
OK fin transfert
selection scalaire module de V 0.0 1.4365 0.0
selection scalaire 10
> gui iso3d
Figure 1
trace iso3d champ module de V
Figure Mayavi Scene 1
CLLongSym16X135000.tar
CLLongSym16X135000.tar.index
CLLongSym16X138750.tar
CLLongSym16X138750.tar.index
CLLongSym16X142500.tar
CLLongSym16X142500.tar.index
CLLongSym16X146250.tar
CLLongSym16X146250.tar.index
CLLongSym16X15000.tar
CLLongSym16X15000.tar.index
CLLongSym16X33750.tar
CLLongSym16X33750.tar.index
CLLongSym16X3750.tar
CLLongSym16X3750.tar.index
CLLongSym16X37500.tar
CLLongSym16X37500.tar.index
CLLongSym16X41250.tar
CLLongSym16X41250.tar.index
CLLongSym16X45000.tar
CLLongSym16X45000.tar.index
[mbuffat@node100 ResultatsRe2500b]$ mpirun -np 16 serveurAnal
Lib parallele $Id: $ compile le Nov 16 2012 a 09:31:29
Processus MPI 0 sur node100
serveur node100
[Control] Demarrage du serveur node100 port= 29876
[Control] listen ...
[Control] connected: ('192.168.84.250', 46255)
[Control] receive cde: cas CLLongSym16X

What was achieved for HPC simulations

A suitable development and software environment

- code C++
- BLAS, GSL
- MPI/OpenMP - optimized libraries (e.g. FFTW, MKL)
- cmake, git
  - swig interface Python and a C++ library derived from the code
  - python, mpi4py, numpy, matplotlib, mayavi ...

Development of a parallel strategy for the code

- revisit parallel strategy of the code
- revisit strategy for the analysis

Resulting method

Characterictics

- Efficient solver for hybrid multicore massively parallel platforms
  - Original coarse grained MPI/OpenMP strategy
  - Tasks overlapping
- Pre- and post- processing tools for smaller MPI platforms
  - parallel VTK format (paraview)
  - Parallel Client/Server programs in Python calling a spectral library
  - 2D/3D parallel visualization - (matplotlib/mayavi)

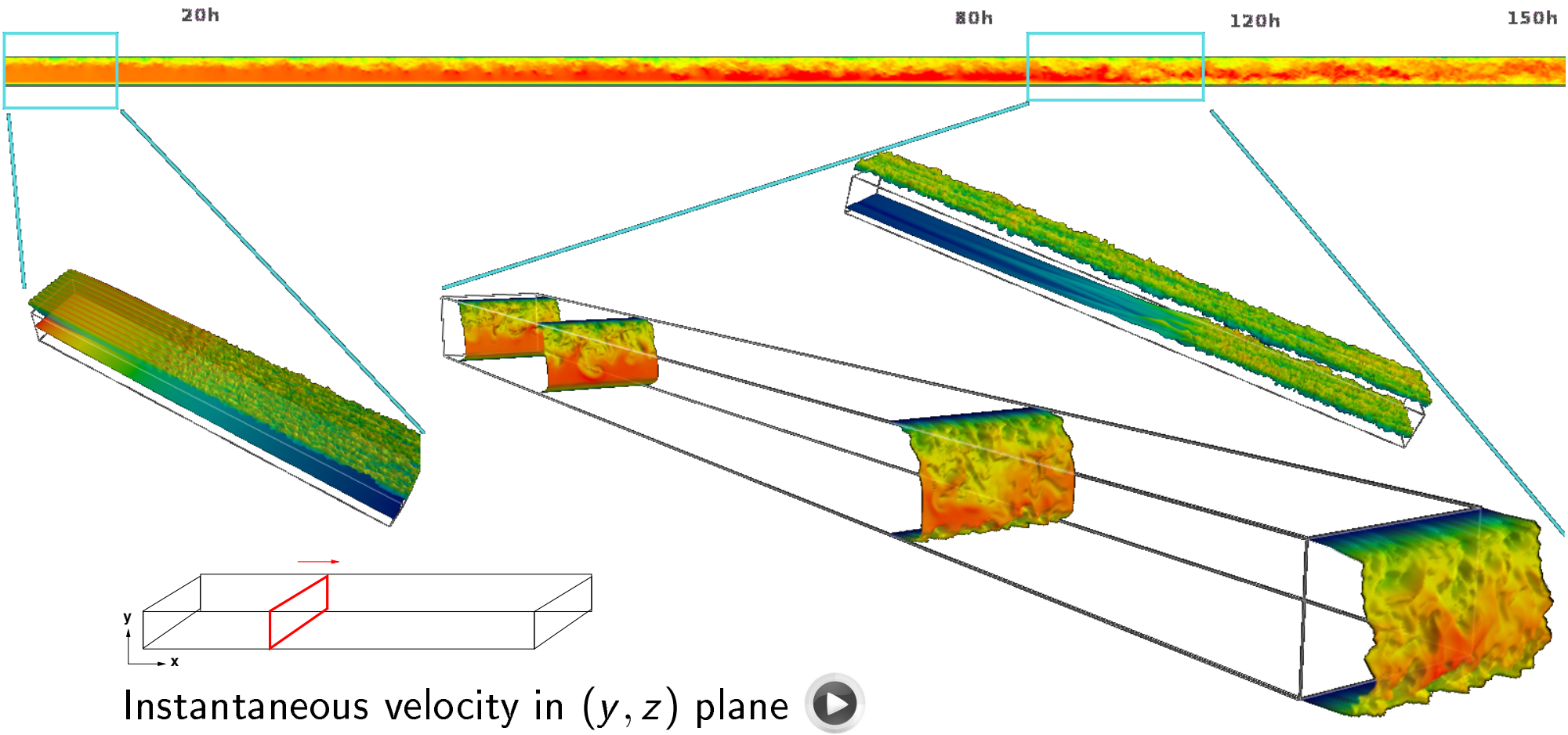
Properties

- Fairly portable
- Small time spent in communications $\sim 10\%$
- Rapid wall clock time for a global scheme
  (1.3s/it on BlueGene/P - 0.2s/it on SuperMUC)

DNS of turbulent transition in channel entrance flow

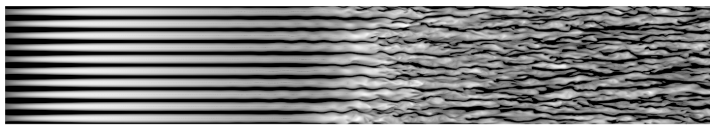
First transition

Second transition



Instantaneous velocity in (y, z) plane

DNS of turbulent transition in channel entrance flow



To read more :
J. Montagnier, A. Cadiou, M. Buffat, L. Le Penven,
Towards petascale spectral simulations for transition analysis in wall bounded flow (2012), Int. Journal for Numerical Methods in Fluids, doi:10.1002/fld.3758