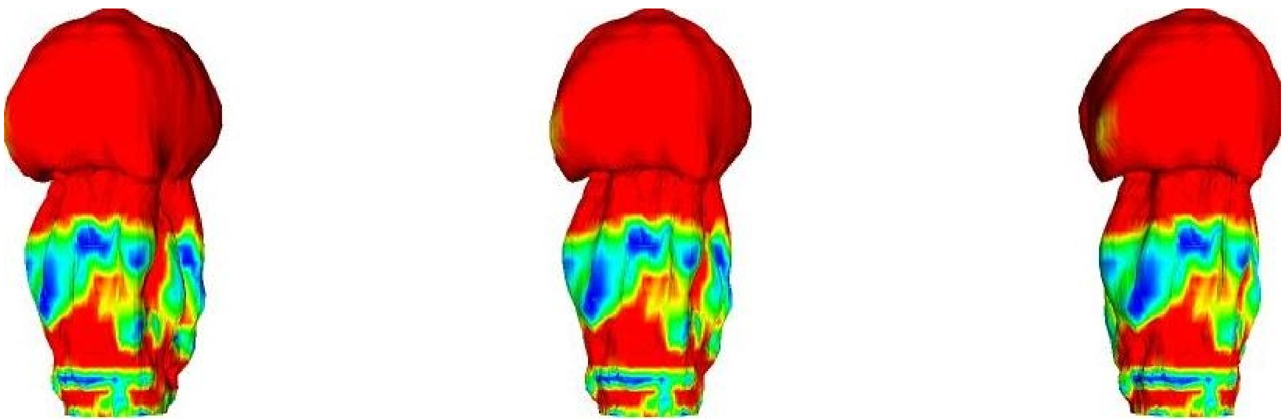
A Grand Challenge of Computational Fluid Dynamics: Simulation of Turbulence



Timo Siikonen

TEKNILLINEN KORKEAKOULU

Contents

- What is numerical simulation of turbulence?
- About the numerical techniques applied
- Case 1 : A jet in a cross flow as calculated using large eddy simulation
- Case 2 : Several jets in a combustion reactor
- The resources required for the simulation of turbulence
- About more demanding cases

TEKNILLINEN KORKEAKOULU

Two definitions of turbulence:

• A state of violent disturbance and disorder (as in politics or social conditions generally); “the industrial revolution was a period of great turbulence” (wordnet.princeton.edu)

• In fluid dynamics, turbulence or turbulent flow is a flow regime characterized by low momentum diffusion, high momentum convection, and rapid variation of pressure and velocity in space and time. The (dimensionless) *Reynolds number* characterizes whether flow conditions lead to laminar or turbulent flow (en.wikipedia.org)

Turbulence modeling in computational fluid dynamics (CFD)

- A traditional way is based on the time-averaged Navier-Stokes equations. Additional modeling is needed for the closure of the problem (Reynolds stresses need to be modeled).

- Simulation of turbulence means that we try to catch turbulent phenomena totally (DNS) or partly (LES) in the numerical solution of the field equations. Thus the solution is always time-dependent and the time scales are small requiring a small time step in the simulation.

Numerical simulation of turbulence

- In direct numerical simulation (DNS) no modeling is applied and the role of numerical error is insignificant. The whole spectrum of turbulence is simulated.

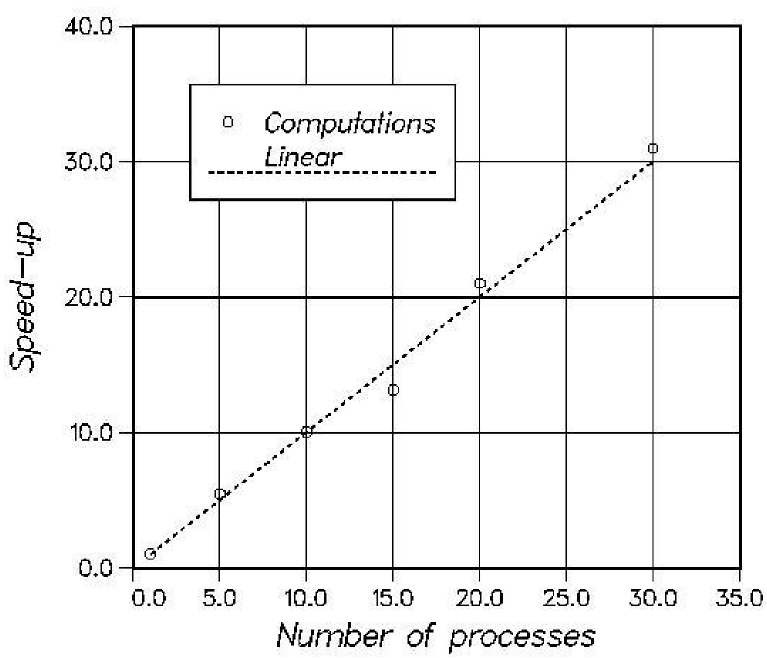
- In large eddy simulation a part (the longest wavelengths) of turbulence are simulated. Short wavelengths are either modeled or not, but always affected by the numerics.

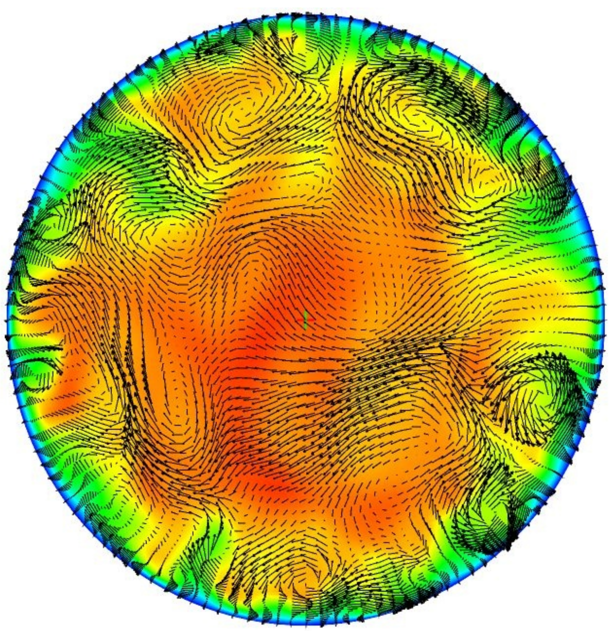
- Reynolds number is an important parameter for characterizing the flow type. DNS is feasible only for low Reynolds numbers.

Present computations

- LES using an in-house code at TKK. Simulations made by Petri Majander.
- A jet in a cross flow at $Re_D = 46700$
- A challenge: large-eddy simulation of a flow field inside a (cold) combustion reactor

- Boundary-fitted finite-volume method, structured mesh
- Pressure correction (projection) method for incompressible flow
- Parallelization by dividing the physical domain between the processors
- Implicit time stepping and iteration within time steps
- Message passing only at the explicit stage of the solution
- Parallelization with even blocks nearly linear in IBMSC with 1-30 processors

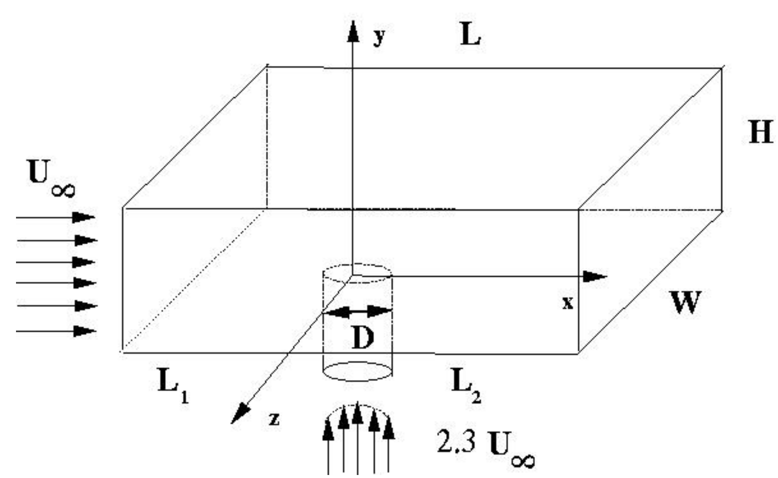




Case: a jet in a cross-flow at $Re_D = 46700$

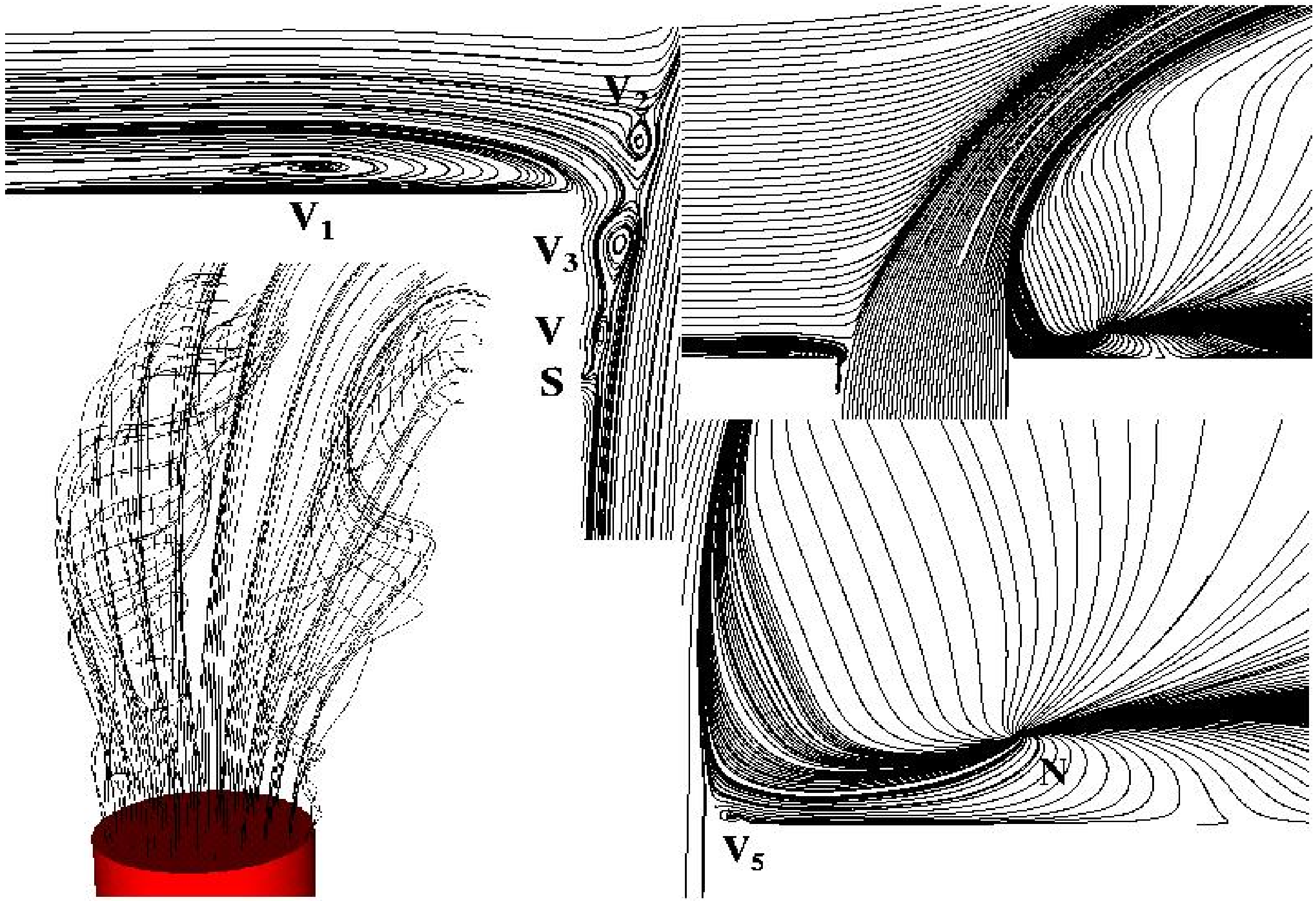
- Jet-to-cross-flow velocity ratio = 2.3
- Experiments by Crabb, Durao and Whitelaw (1981,J.Fluids Engng,103).
- 2 764 800 control volumes equally distributed into 25 blocks/processors
- The bulk cross-flow travels from the inlet to the outlet in 5200 time steps
- The computation time was around ten cross-flow-through times
- The statistics were collected during the last six flow-through times
- This took around 40 days with IBMSC

		EXP.	LES
$L = L_1 + L_2$	(D)	6 + 70	3 + 10
W	(D)	18.1	8
H	(D)	11.8	7



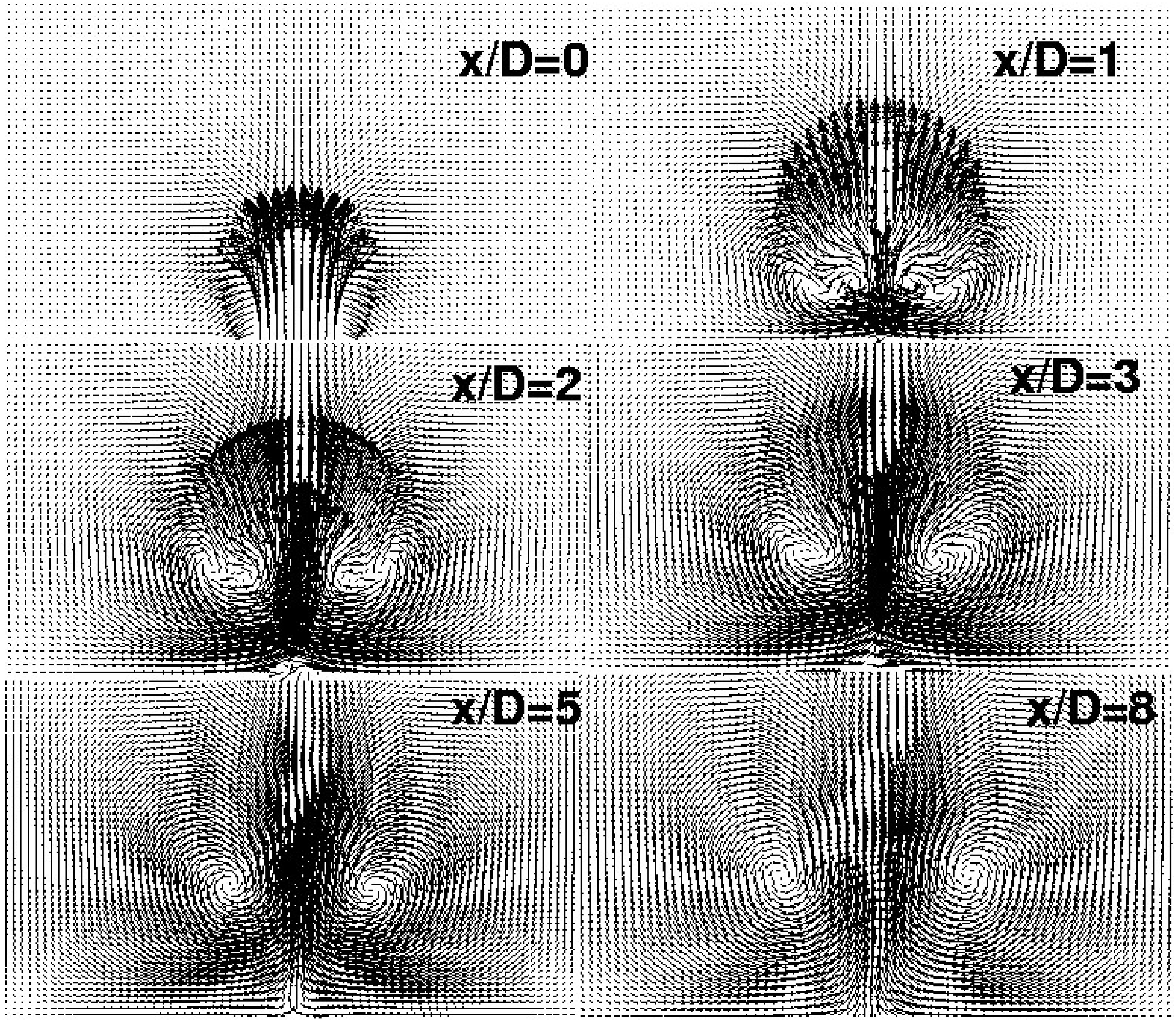
TEKNILLINEN KORKEAKOULU

- Jet in a cross-flow : averaged streamlines in the symmetry plane

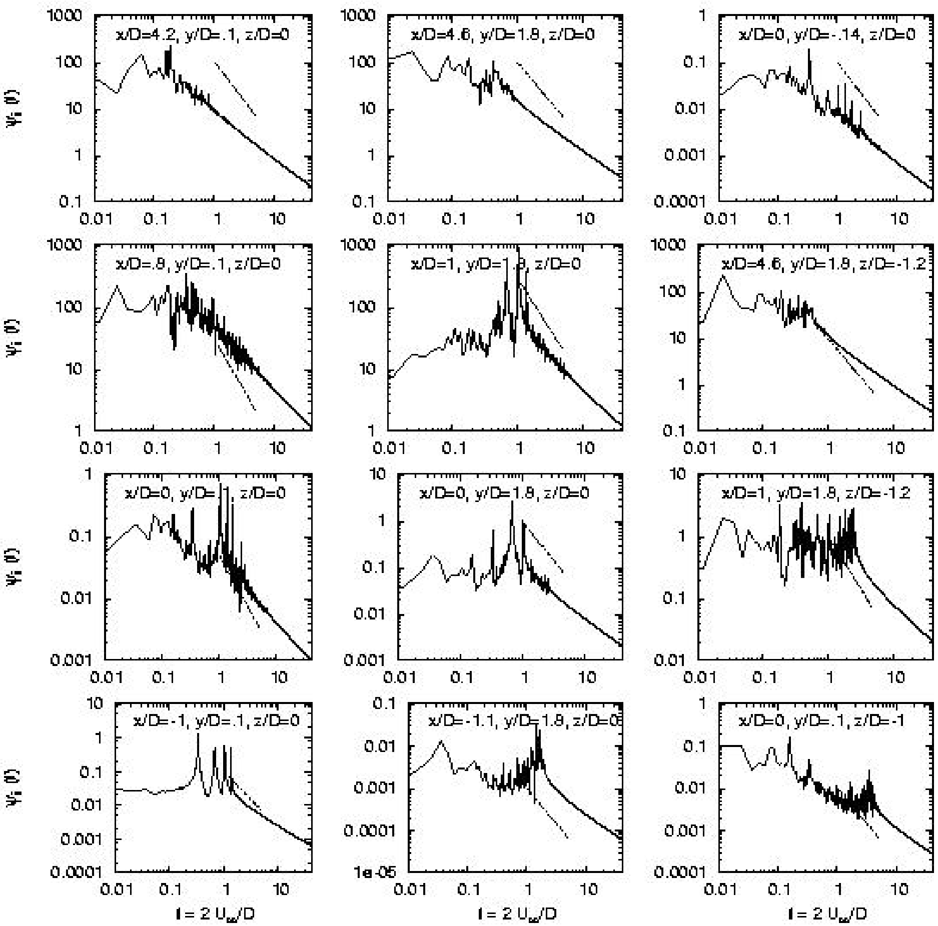


TEKNILLINEN KORKEAKOULU

- Jet in a cross-flow : an evolution of the counter-rotating vortex pair



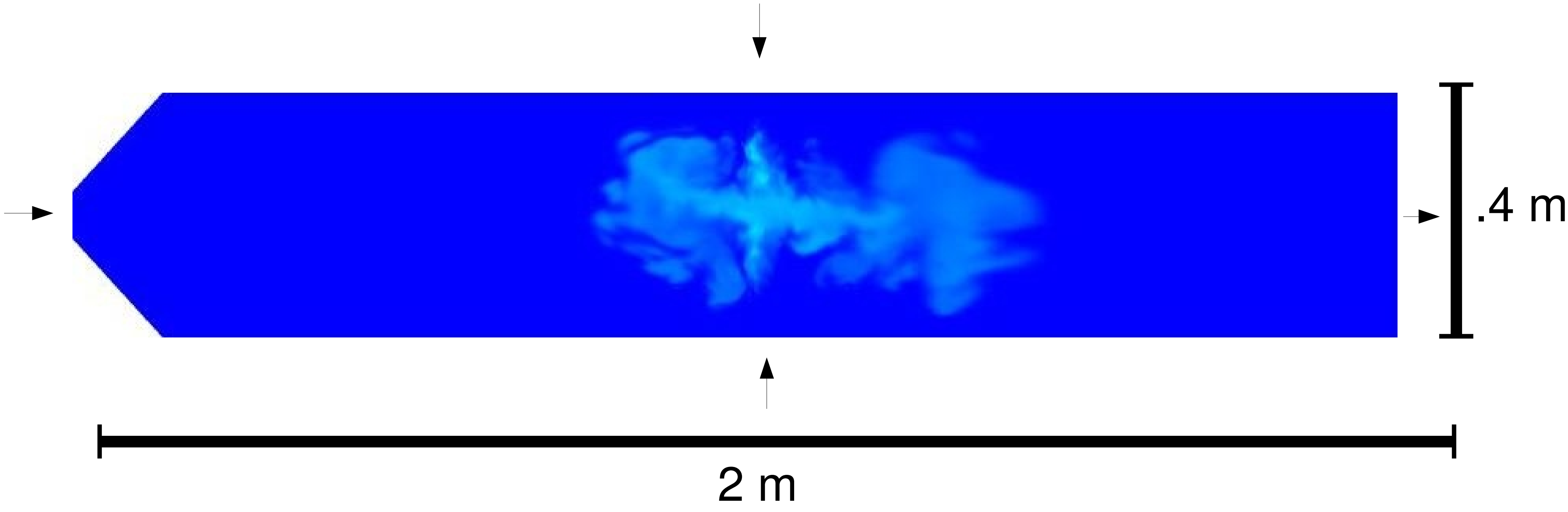
- Jet in a cross-flow : time series are recorded
- Spectra can be used to analyze the resolution of the computation
- An animation of the 2nd invariant of the velocity gradient shows vortices



TEKNILLINEN KORKEAKOULU

A challenge: large-eddy simulation of a flow field inside a (cold) combustion reactor

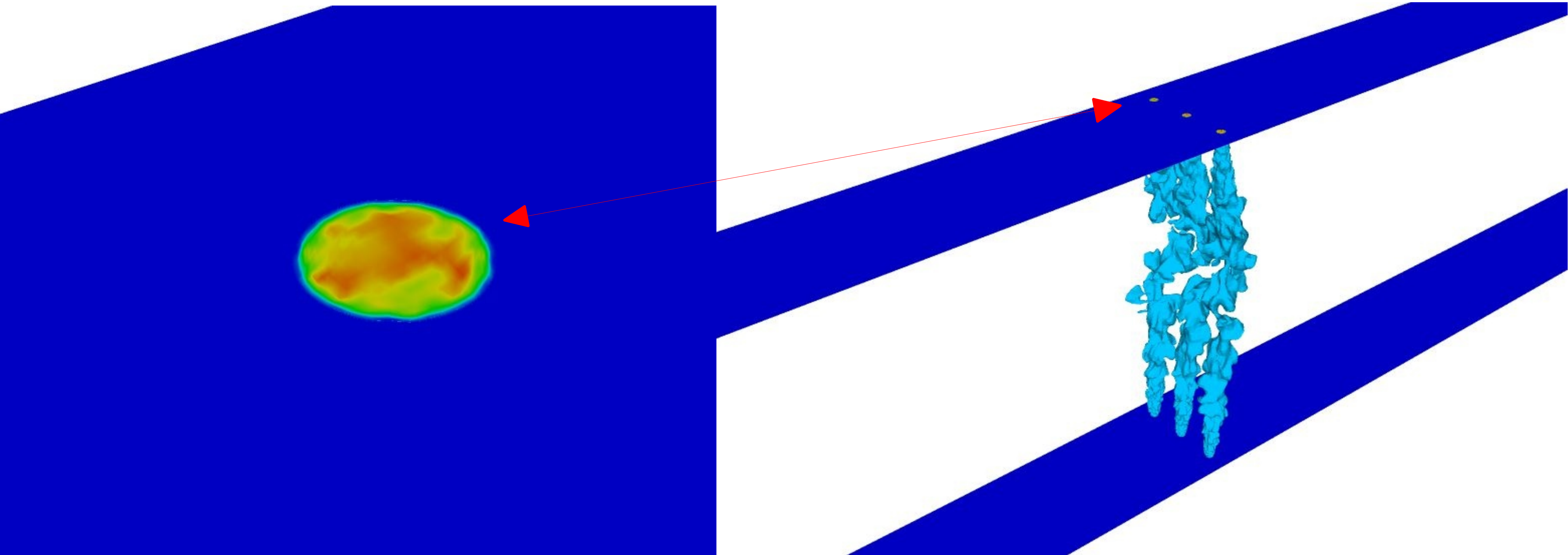
- Industrial interest: a uniform spreading of the secondary air
- A scale-down model pilot reactor at TUT includes 8 opposing jet pairs
- Size 0.4 m x 0.4 m x 2.5 m
- In a current mesh 3 opposing jet pairs and 4 million control volumes



TEKNILLINEN KORKEAKOULU

A challenge: large-eddy simulation of a flow field inside a (cold) combustion reactor

- Large range of scales : expensive computation
- Turbulent pipe flow (D=0.8 cm) is fed into the domain (total length=200 cm)



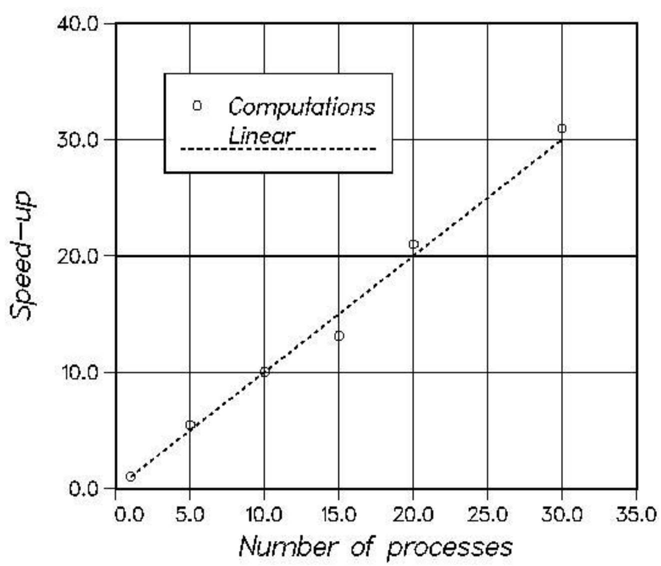
Experiences about the computer runs

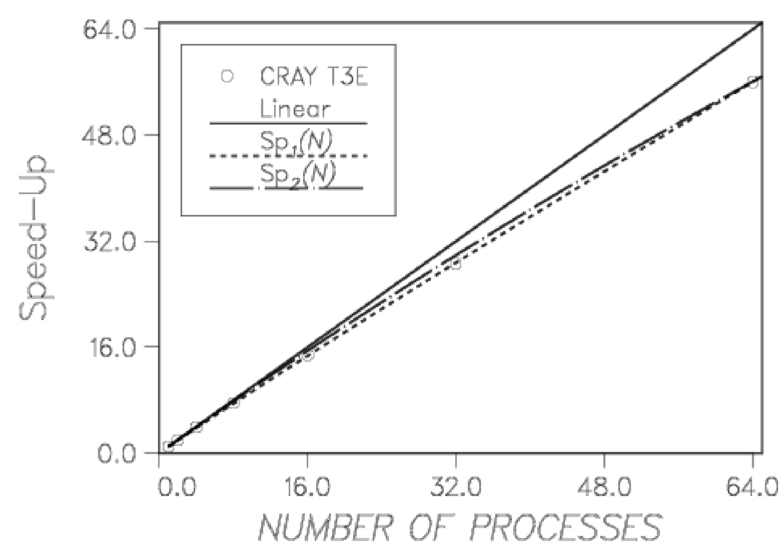
- The jet in a cross flow case took 40 CPU days with 25 processors
- The combustion reactor has been running about one month with 30 processors
- Utilization of the CSC resources has been necessary in this case
- The problem size would be suitable for a smaller (local) cluster

TEKNILLINEN KORKEAKOULU

Two ways to consider the efficiency of parallelization

- In scale-up the problem size is scaled with the number of CPUs
- In speed-up a problem size is fixed and should be solved as quickly as possible
- A scale-up performance can be linear and turbulence simulation is ideally suited for this kind of parallelization (increase in Reynolds number)
- A speed-up performance is more critical and never linear => It is more economical to increase the problem size than to solve the case quickly





An ultimate example

- We have 32 computer runs that take 32 CPU days each and a resource of 32 CPUs
- In a single processor mode all cases are completed in 32 days
- Furthermore, since in CFD we do not know how long it takes, we can at least look at the preliminary results daily (or weekly)
- In a parallel mode the first case is finished after about one day, but the execution of all cases takes maybe 40 days (speed-up is non-linear)
- Many cases end to a disaster (input errors etc.)
- It is always my computer run that is the last one in a queue

=> A parallelization is quite useless

TEKNILLINEN KORKEAKOULU

How can we help the rabbit to win the race?

Lessons learned: Back to good old SGI days

It was noticed that the local RISC machine performance was in most cases better than that of the famous super-computers

At present there is no money left, but...

..the local (Linux) clusters are remarkably cheap and efficient. A local cluster could have been used in the present cases.

About DNS and LES

- In a 3D time-dependent simulation the computational work is proportional to X * Y * Z * t
- In DNS CPU time is about Re^3 and memory requirement Re^(9/4)
- In LES the computer capacity requirement is smaller
- However, as the Reynolds number is increased there is no capacity limit, all the resources available can be spent

Back to the resource problem

- In spite of any remedies a computing centre will loose the race in ordinary (CFD) simulations
- The given examples were still in a class of 'a small memory size'
- Rather than computing cases quickly we should concentrate on a really large-scale computing
- In the large scale computing a software applied should show a linear scale-up performance
- By increasing the Reynolds number turbulence simulation is in an eternal need of resources and the problem scales up
- At the moment the necessary know-how is available for a real 'grand challenge in CFD'

Conclusions

- Turbulence simulation is a good way to waste computer resources
- A skillful simulation may serve as a benchmark for decades (e.g. a channel flow case by Kim, Moin and Moser in 1987)
- One way to deliver resources: If a simulation calender time is weeks or even months, the project has earned more resources