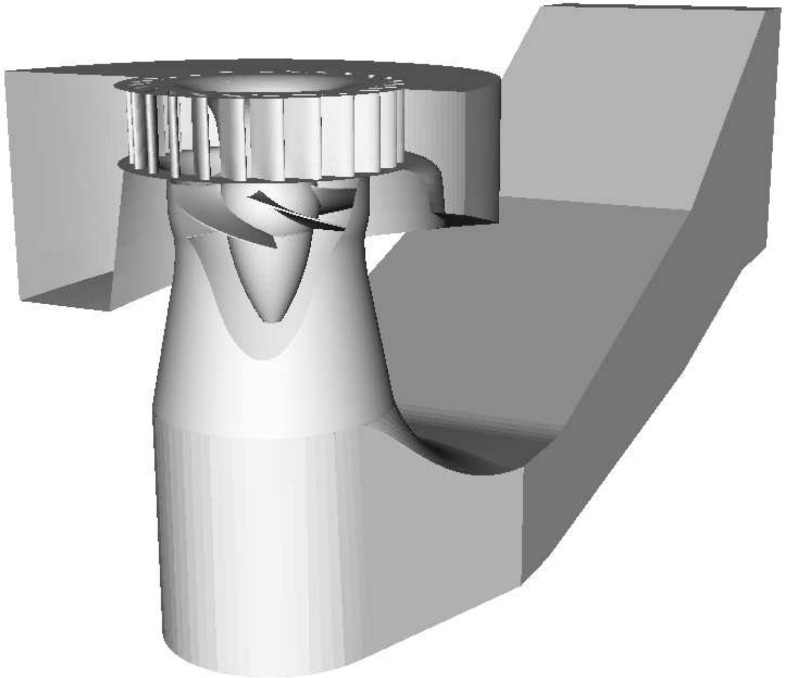
CFD in water turbines – an HPC challenge



Financing:
SVC: Svenskt VattenkraftCentrum (www.svc.nu):
Swedish Energy Agency,
Hydro Power companies[a] (through Elforsk AB),
GE Energy (Sweden) AB,
Waplans Mekaniska Verkstad AB

[a]Vattenfall AB Vattenkraft, Fortum Generation AB, Sydkraft Vattenkraft AB, Skellefteå Kraft AB, Graninge Kraft AB, Jämtkraft AB, Sollefteåforsens AB, Karlstads Energi AB, Gävle Energi AB, Öresundskraft AB

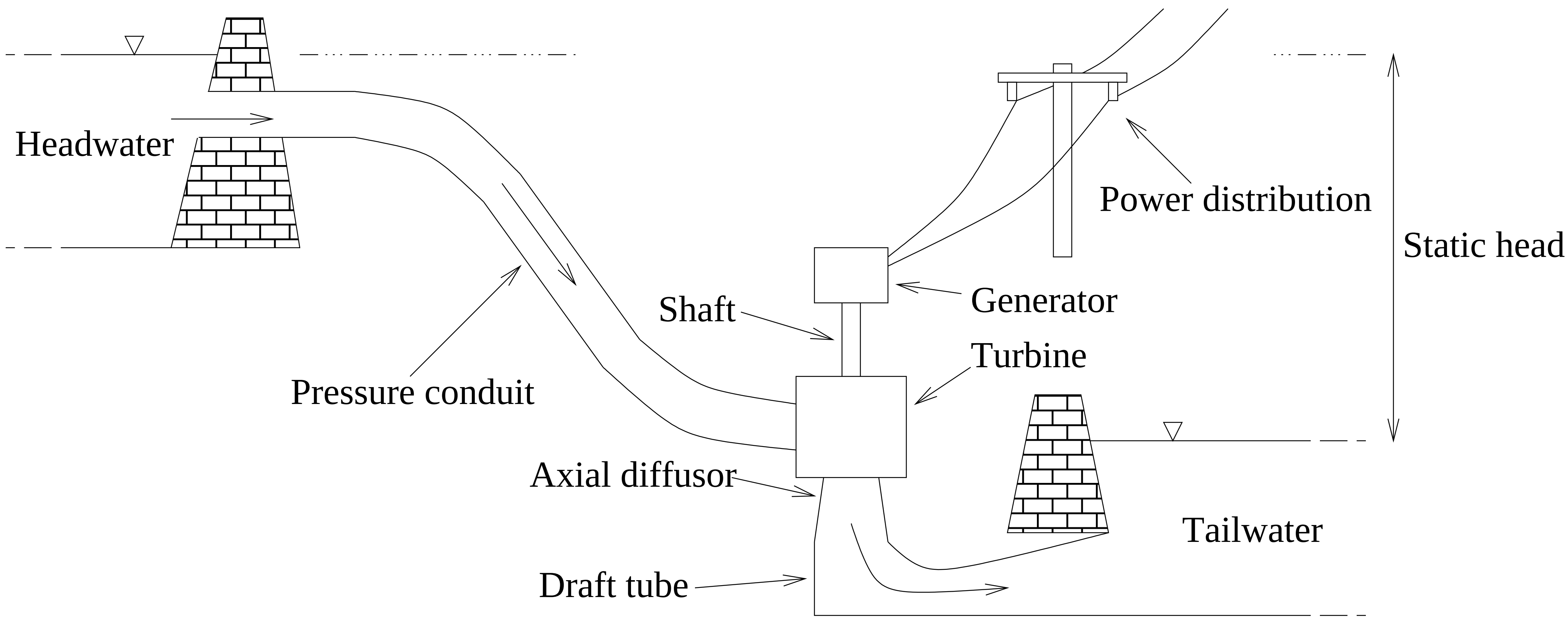
Outline

- Background with description of water power in general and flow features in water turbines in particular.
- Why is CFD in water turbines an HPC challenge?
- The OpenFOAM code
- A parallel performance test

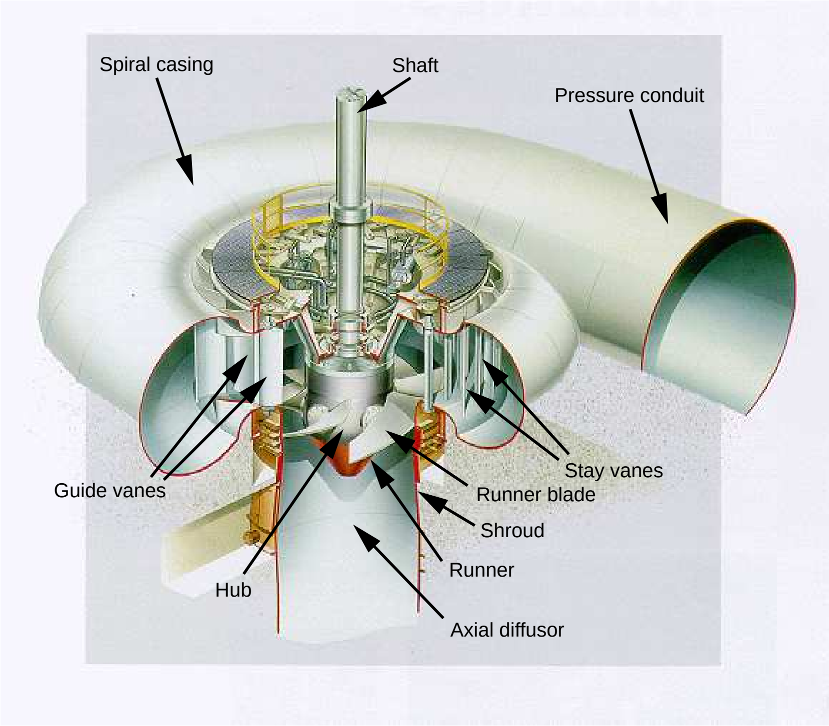
Background

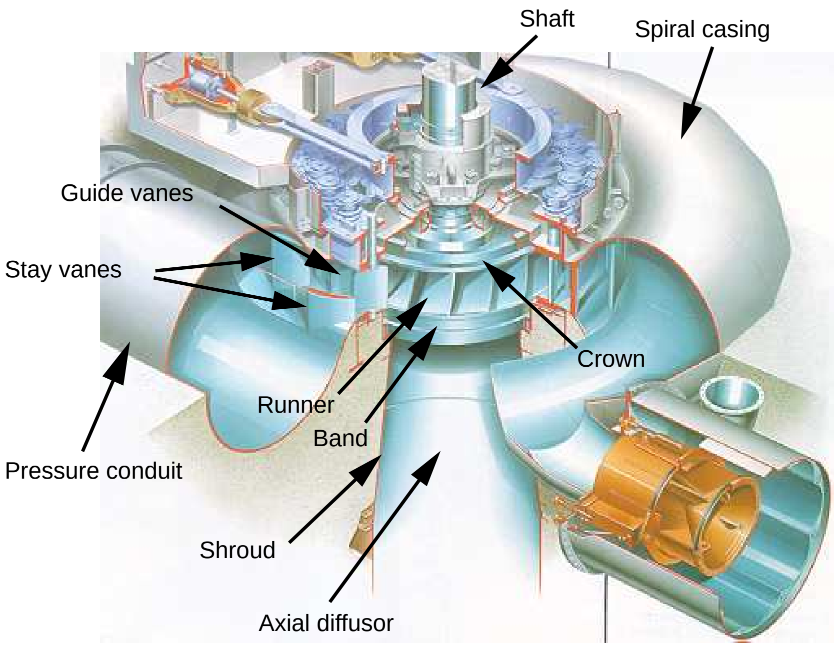
- 50% of the electric power in Sweden is generated by water power.
- Many of the power plants in Sweden are getting old and some major refurbishments are coming up.
- Computational Fluid Dynamics (CFD) is to a large extent used as a design tool for this purpose.
- In order to study the flow in detail enormous HPC facilities and new methods are required.

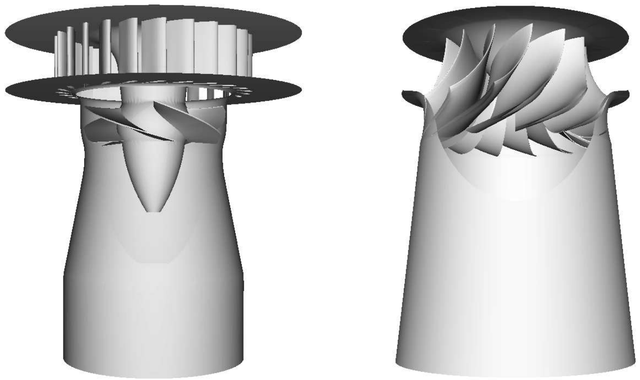
Schematic description of a hydraulic power plant



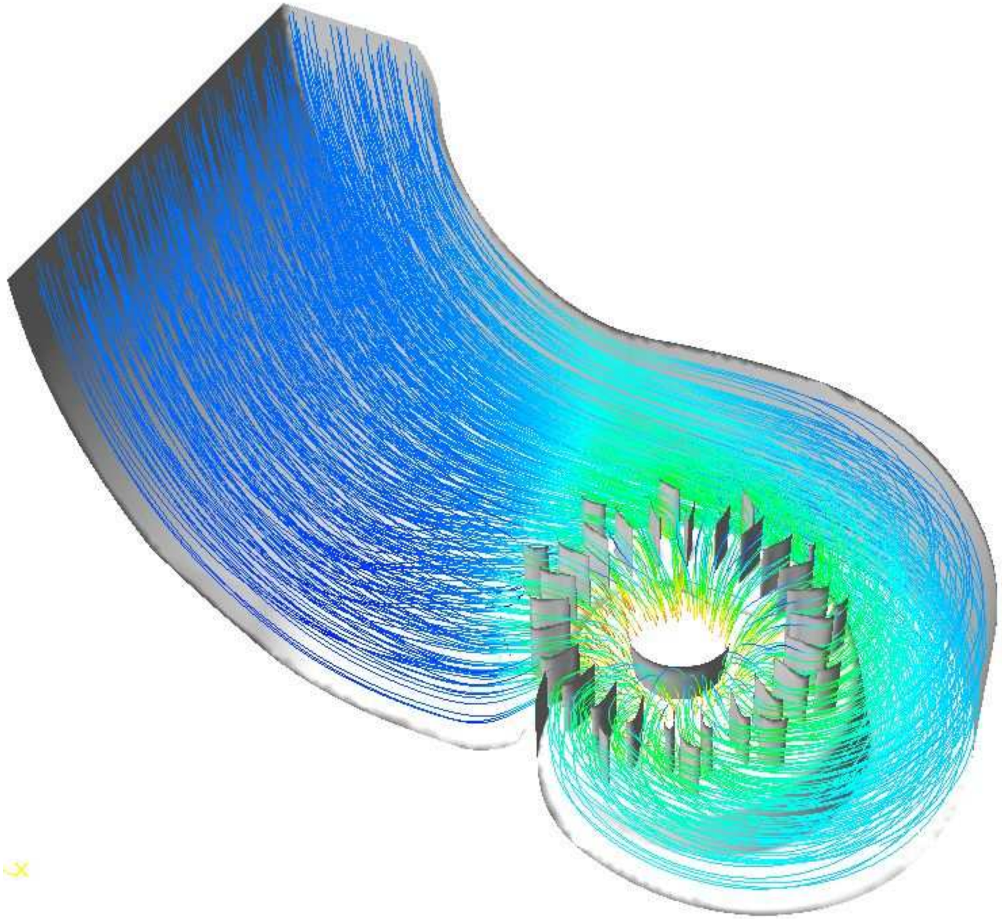
Kaplan and Francis runners

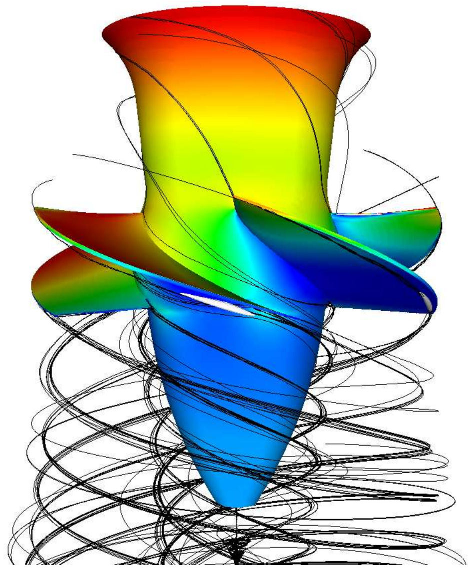


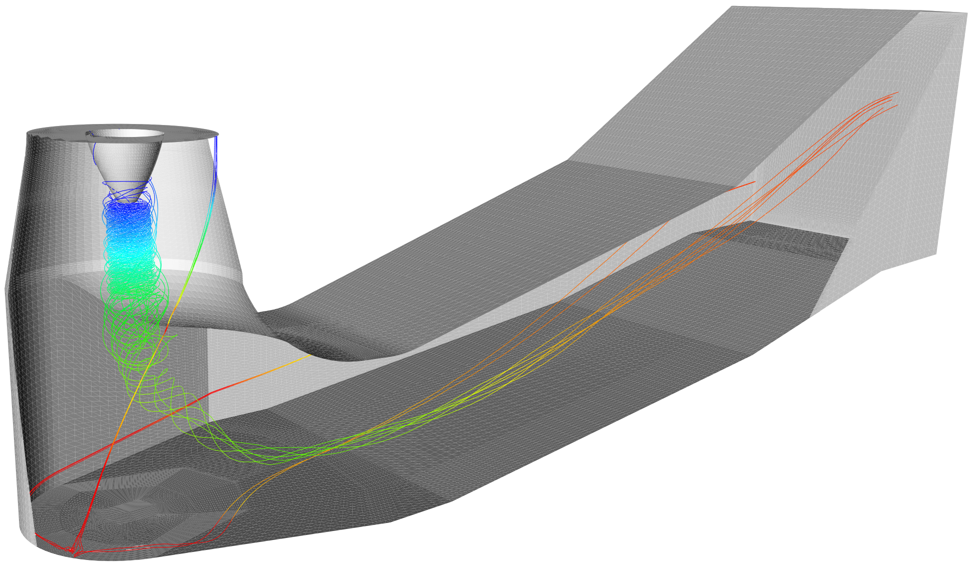




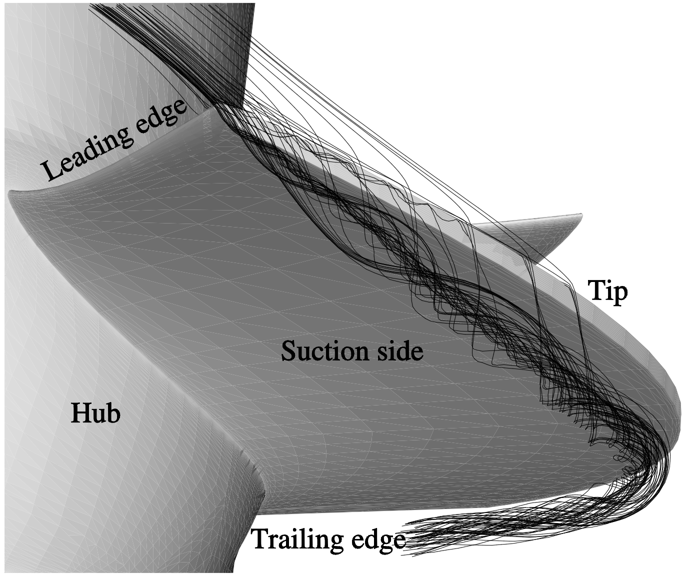
Coupling between flow in different parts of the system

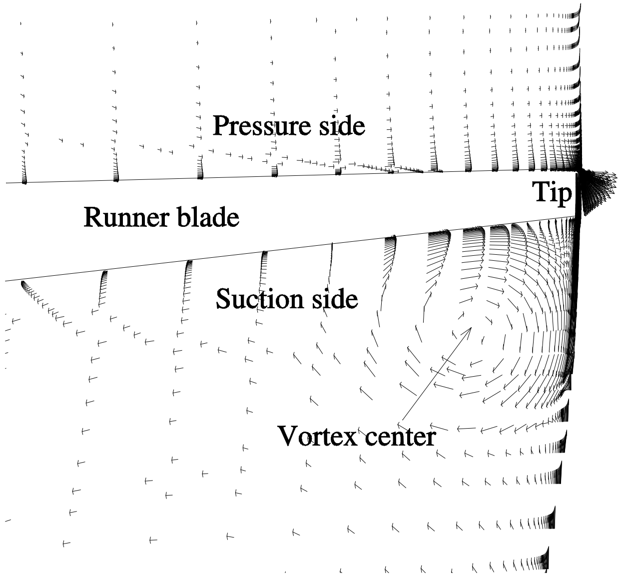


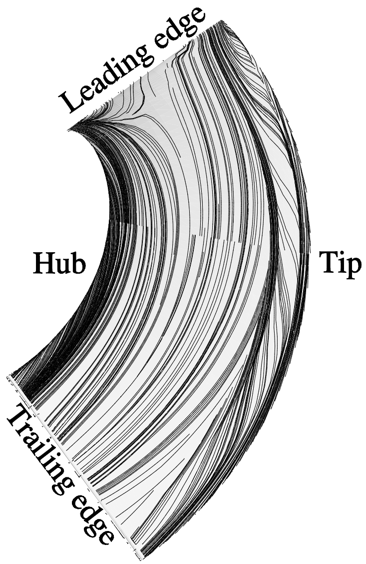




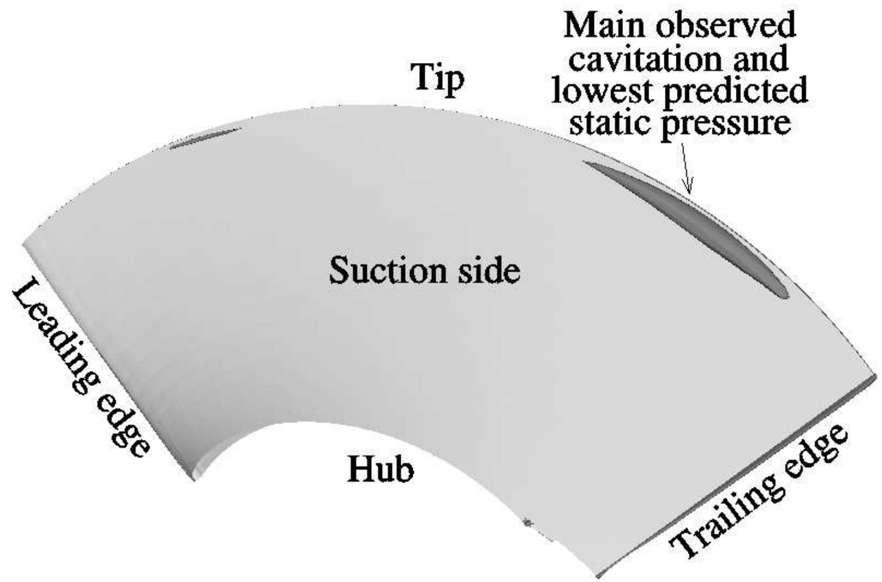
Tip vortex in Kaplan turbines

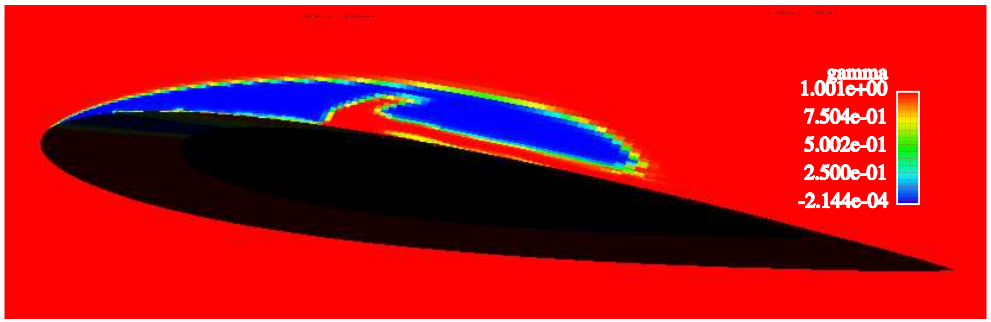


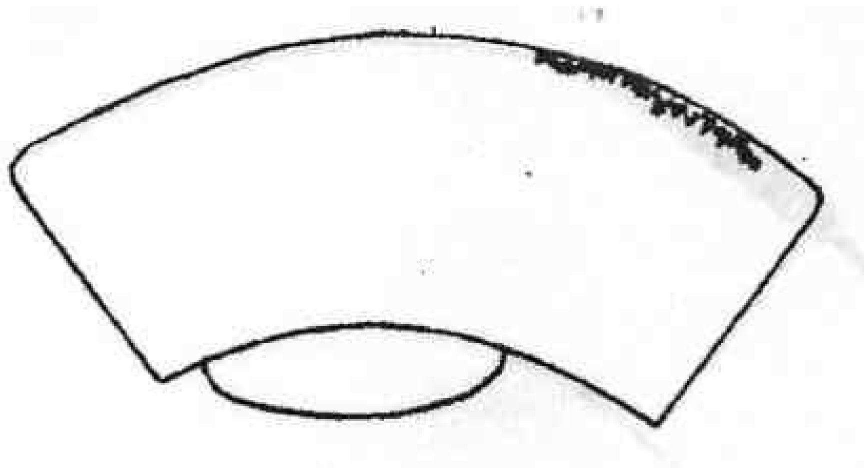




Cavitation and its modelling

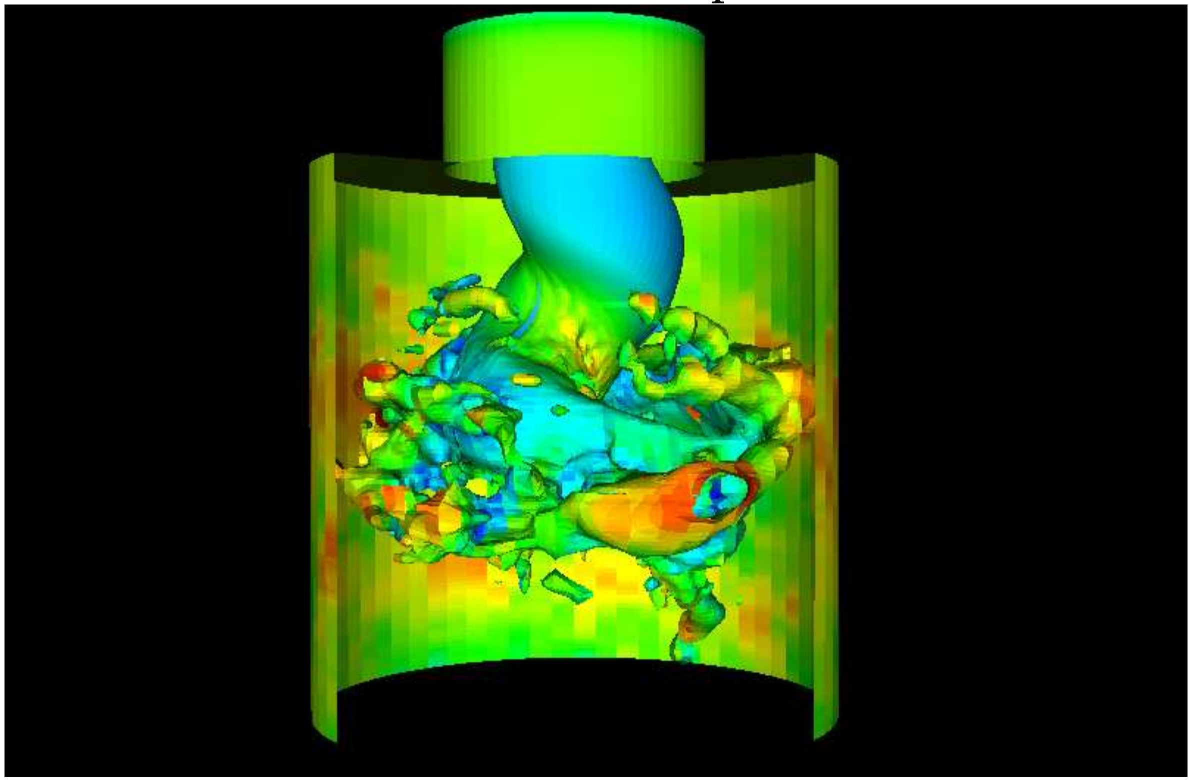




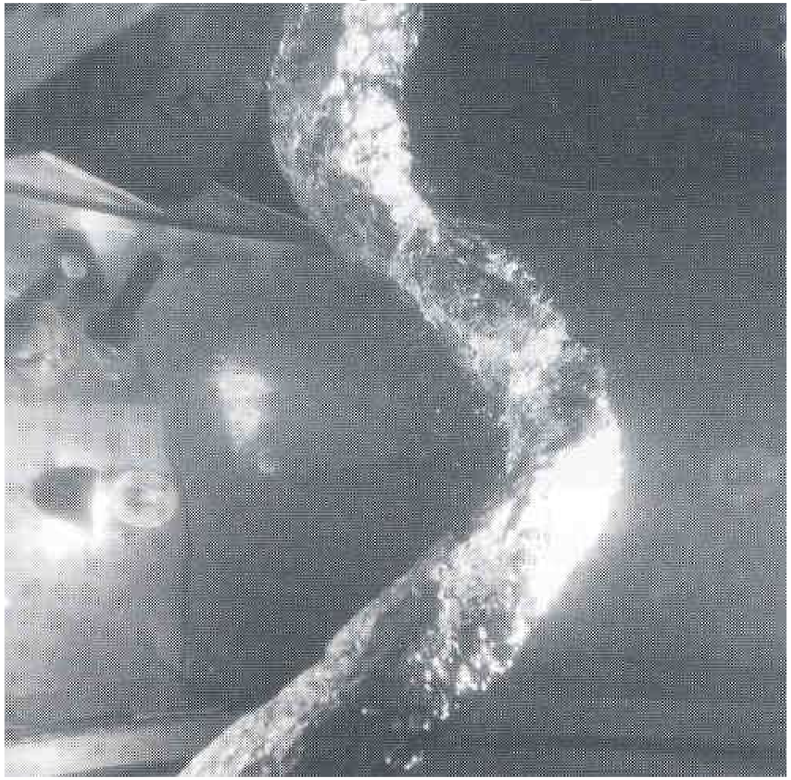


Unsteady flow in draft tubes

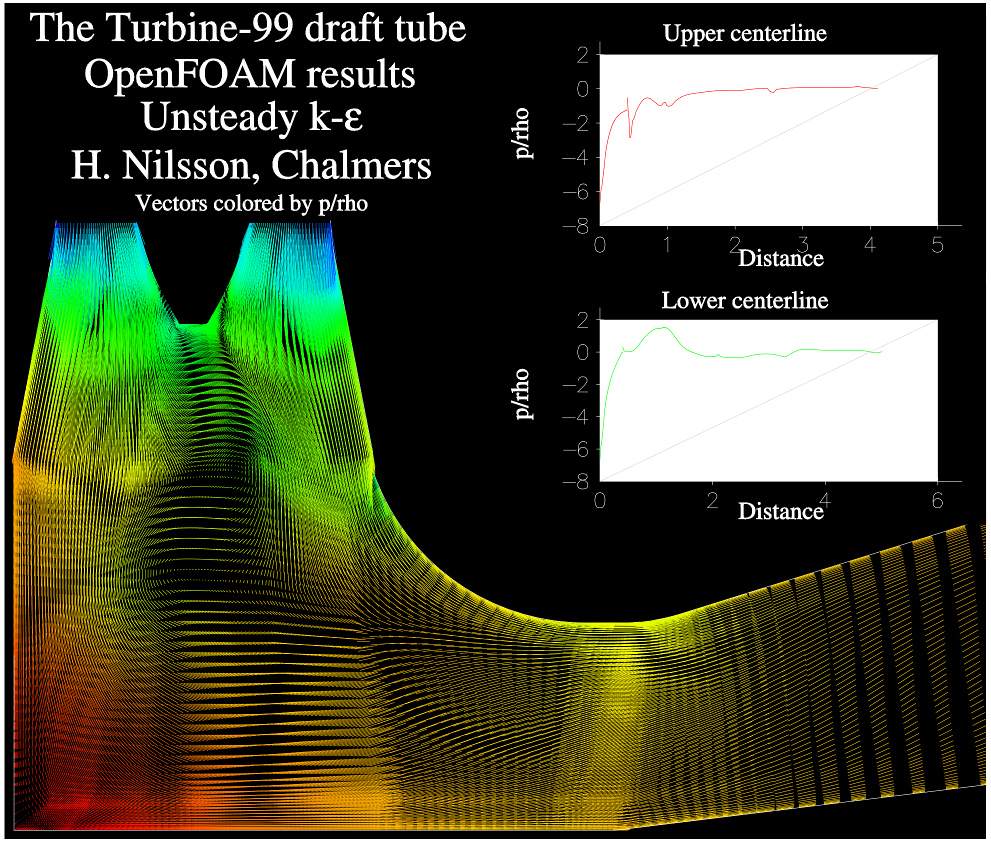
Snapshot of single-phase LES computation.
Iso-surface of static pressure.

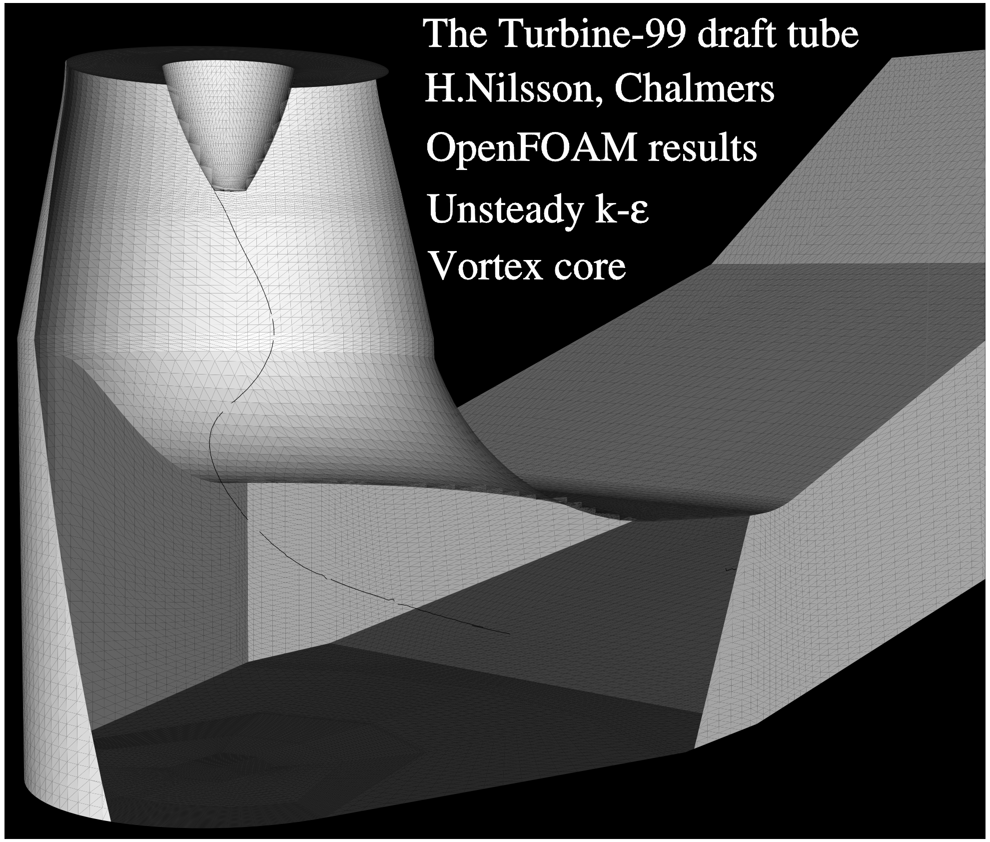


Cavitating vortex rope.



Draft tube results, unsteady $k-\varepsilon$ with wall functions





Why is CFD in water turbines an HPC challenge? (If it hasn't been answered already)

- Resolution of the small scales: High Reynolds numbers ($\sim 10^7$) – high gradients and small flow features (turbulence/wakes/vortices) – needs fine resolution – yields large grid sizes
- Resolution of the large scales: Difficult to set boundary conditions without computing the whole system – yields large domains and grid sizes
- Rotor-stator interaction – a full coupling requires unsteady computations and time consuming methods
- VOF/cavitation – requires high grid resolution at the thin interface, unsteady computations and small timesteps
- Unsteady turbulent flow – LES – requires high resolution, unsteady computations and small timesteps
- Automatic shape optimization – requires many high-quality CFD computations
- Grid generation – an issue for large grids
- Visualization – an issue for large grids

The OpenSource OpenFOAM CFD solver

- OpenFOAM = Open Field Operation and Manipulation, www.openfoam.org
  An OpenSource object oriented C++ tool for solving PDE's
- Preprocessing (grid generator, converters, manipulators, case setup)
- Postprocessing (using OpenSource Paraview)
- Many specialized CFD solvers implemented, e.g.
  - simpleFoam: A finite volume steady-state solver for incompressible, turbulent flow of non-Newtonian fluids, using the SIMPLE algorithm
  - turbFoam: A finite volume solver for unsteady incompressible, turbulent flow of non-Newtonian fluids, using the PISO algorithm
  - icoDyMFoam: Sliding grid
- OpenSource = possibility to have insight into the code
  - * Makes development and tailor-made solvers possible
    - simpleUnsteadyFoam: Unsteady SIMPLE solver
    - cavInterFoam: Cavitation using VOF and the Kunz' cavitation model
  - * Makes research implementations available and results reproducable.
- Access to an international community of OpenFOAM users
- Runs in parallel using automatic/manual domain decomposition.

A parallel performance analysis

- Test case:
  The draft tube with a 10^6 cell grid,
  automatically subdivided into 1, 2, 4, 8 and 16 subdomains.
- Cluster:
  A four-node Linux cluster Dual socket AMD Opteron 280 (2.4 GHz, dual core) with 4GB DDR400 RAM, i.e. 4 cores (CPUs) per node and a total of 16 cores (CPUs).
- Interconnects:
  Gigabit Ethernet through an HP ProCurve 2824 Switch, and an
  Infiniband (PCI-X) through a Silverstorm 9024 Switch.
- Linux version:
  SuSE Linux Enterprise Server, Service pack 3.

Results of the parallel performance analysis

- Wall clock times and normalized wall clock times for three iterations
- Normalized improvement due to the Infiniband network

# CPU	# nodes	ETH (s)	IBA (s)	ETH (speed-up)	IBA (speed-up)	$\frac{IBA\ (speed-up)}{ETH\ (speed-up)}$ (based on speed-up)
1	1	165	163	1.0	1.0	1.0
2	1	86	78	1.9	2.1	1.1
2	2	85	81	1.9	2.0	1.0
4	1	76	72	2.2	2.3	1.0
4	2	64	62	2.6	2.6	1.0
4	4	53	56	3.1	2.9	0.9
8	2	43	41	3.8	4.0	1.0
8	4	41	35	4.0	4.7	1.2
16	4	23	20	7.2	8.2	1.1

- Parallel efficiencies ranging between 45% and 105%
- The distribution of the processes on the available cores has a significant impact
- No significant speed-up when using the Infiniband network

Thank you for your attention!