Авиадвигатель ОДК

ON THE COMPUTATION OF AIRCRAFT ENGINE FAN NOISE GENERATION USING HIGH ORDER NUMERICAL METHODS ON GRAPHIC PROCESSING UNITS

L. U. Gomzikov[1], **E. V. Koromyslov**[1,2,3], M. V. Usanin[1], A. A. Siner[1,2]

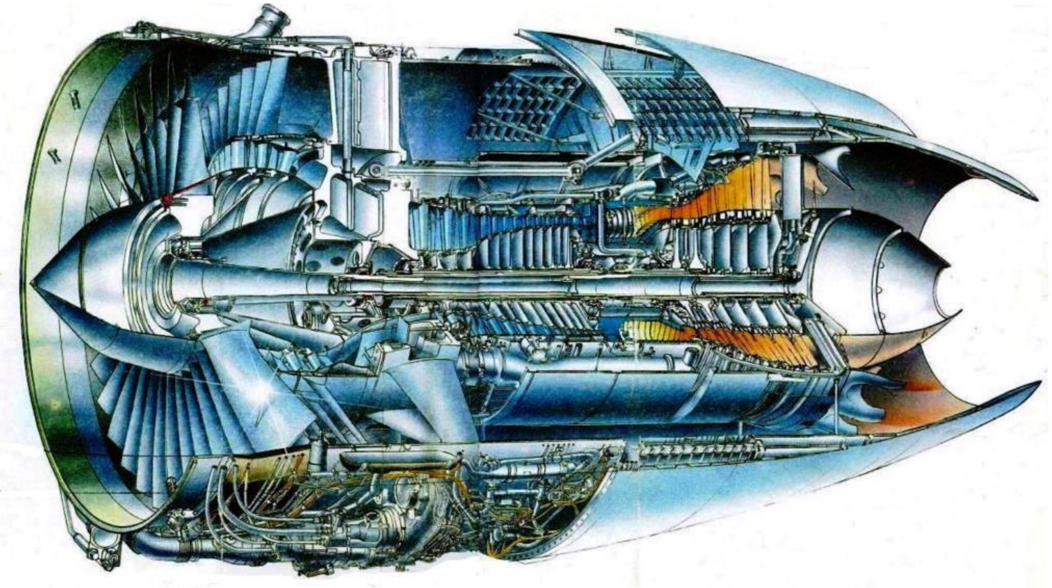
[1]OJSC "Aviadvigatel", Perm, Russia, koromyslov@avid.ru

[2]Perm State National Research University, Perm, Russia

[3]Institute of Continuous Media Mechanics UrB RAS, Perm, Russia

Aircraft engine noise and emission

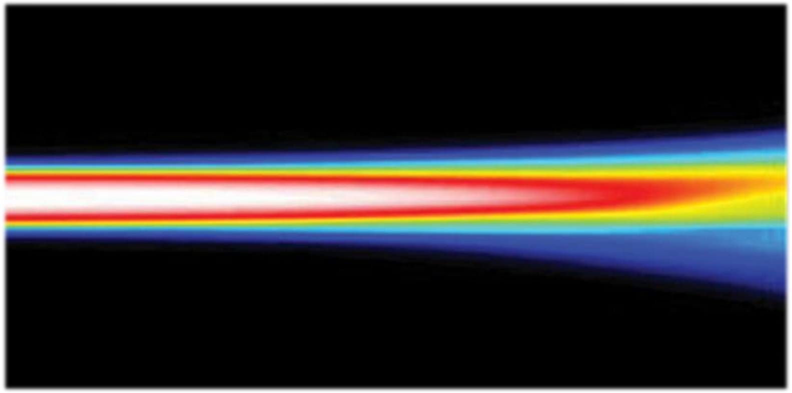
- Ecological standards are constantly tightened

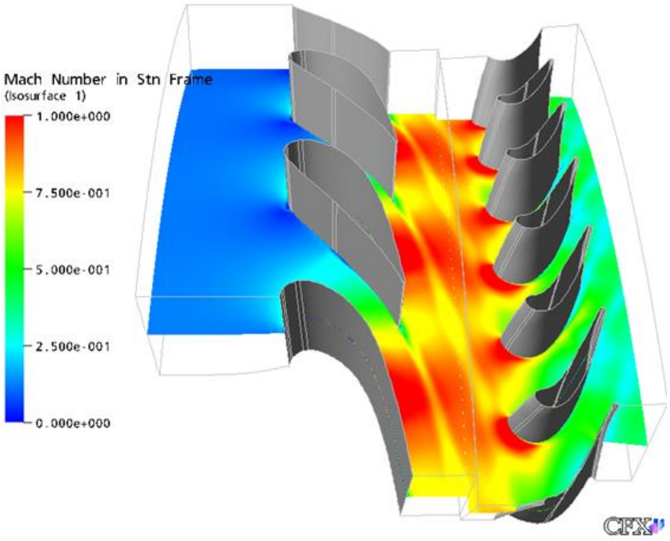


- Noise sources:
  - Fan rotation
  - Jet nozzle
- Emission source:
  - Non-uniform combustion at the combustion chamber

Engineering gas dynamic simulations (1)

- **Currently:**
  - **Gas flow modeling based on RANS equations with turbulence modeling**
  - **Low order numerical schemes**
  - **Steady-state solutions**
  - **Computation of only part of the domain, exploiting periodicity if possible**

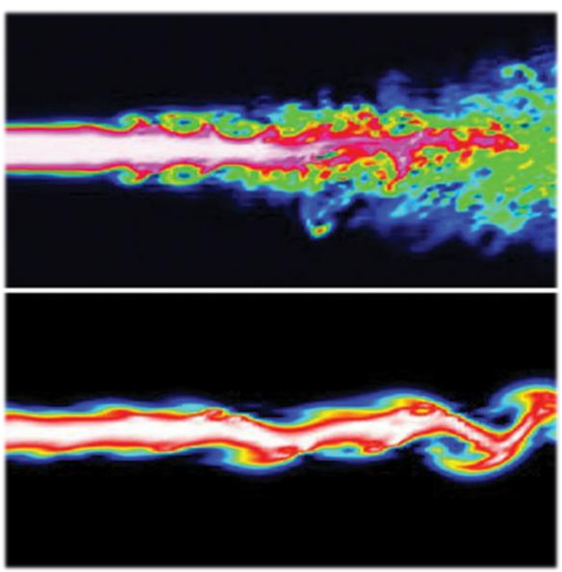


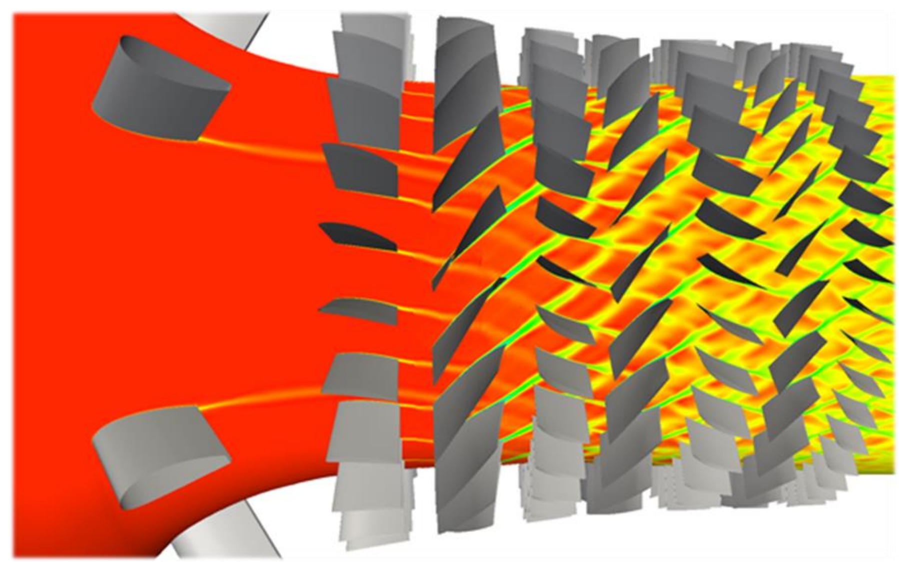


Engineering gas dynamic simulations (2)

- **Desired:**
  - **Simulation based on “first principles” (LES, DNS)**
  - **High order numerical schemes**
  - **Unsteady solutions**
  - **Computation of whole domain**

  => High requirements for computational resources





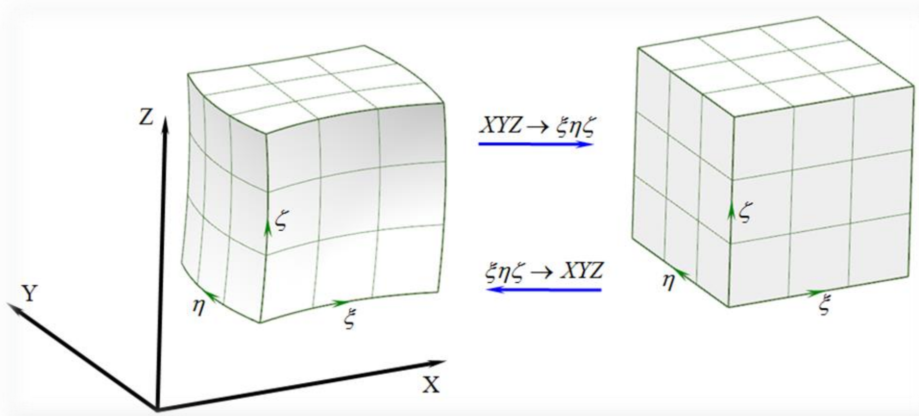
GHOST CFD Solver: Capabilities (1)

- **Full Navier-Stokes equations without *explicit* turbulence modeling**
- **Finite differences on structured curvilinear meshes**
- **DRP schemes for spatial derivatives computation**

  (13 points stencil in each direction, 4th order – C. Bogey, C. Bailly)
- **Optimized 6-stage low storage explicit Runge-Kutta scheme for time derivatives (4th order)**
- **Selective filtering of high frequency waves which are poorly resolved on the computational mesh (*LES-RF*)**

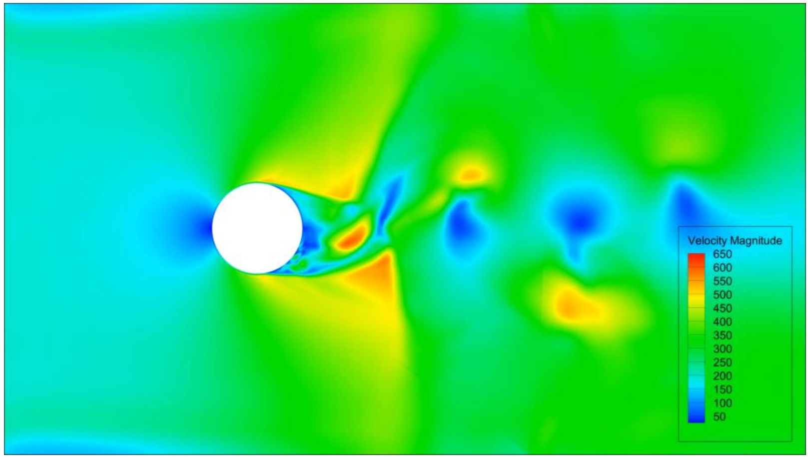
$$\frac{\partial U}{\partial t} + A_x \frac{\partial U}{\partial x} + A_y \frac{\partial U}{\partial y} + A_z \frac{\partial U}{\partial z} = F_v$$

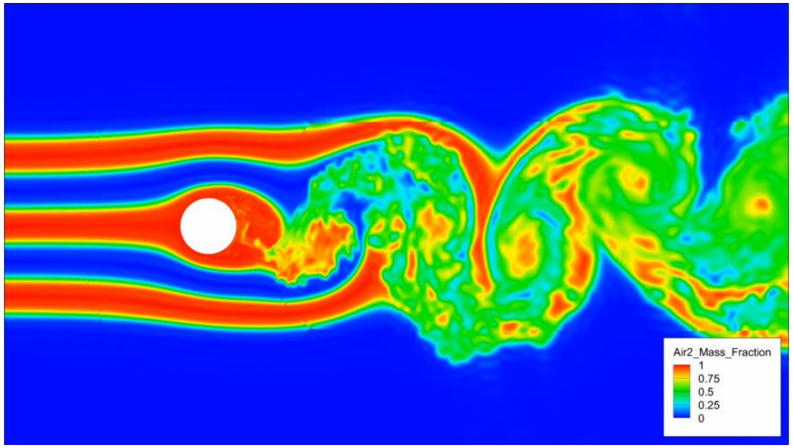
$$U = [\rho \quad u \quad v \quad w \quad p]^T$$

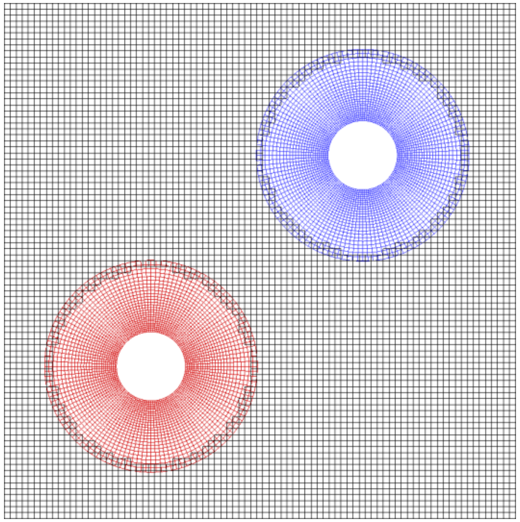


GHOST CFD Solver: Capabilities (2)

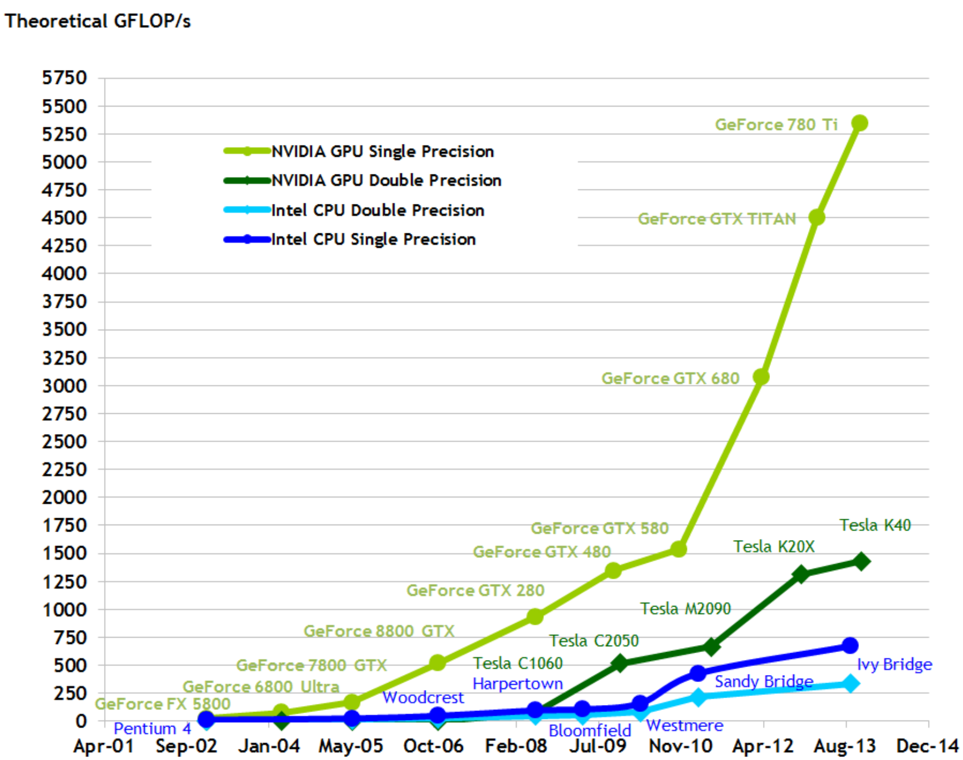
- **Shock-Capturing filtering for transonic flows**
- **Support for multiple gas species (via mass fraction transport equation)**
- **Overset (CHIMERA) 2-layer meshes with relative mesh components movement support (2nd order Lagrangian interpolation)**

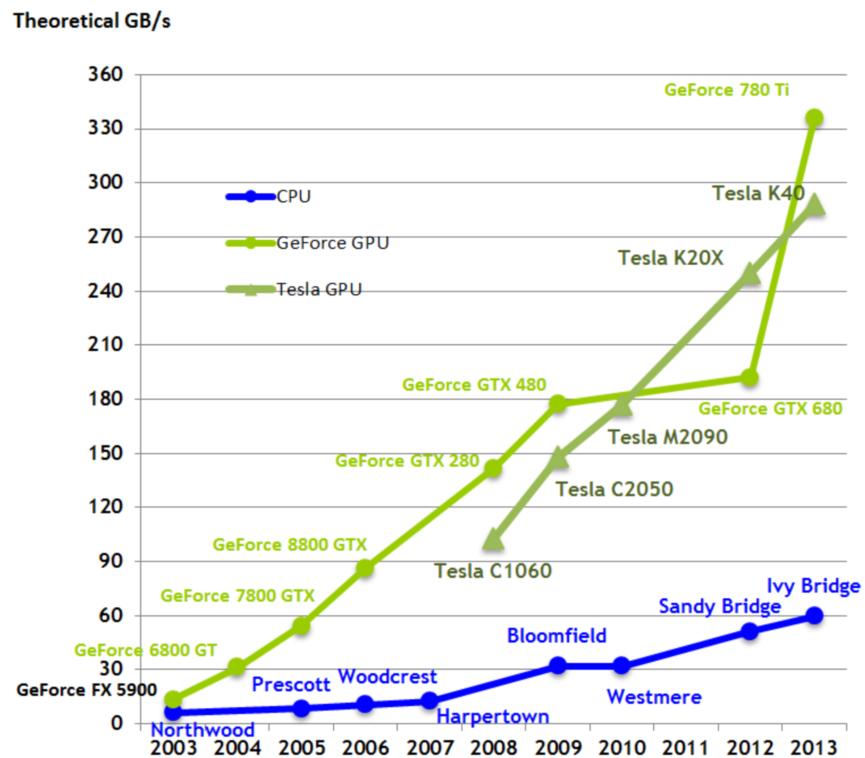






GPU vs CPU: Performance





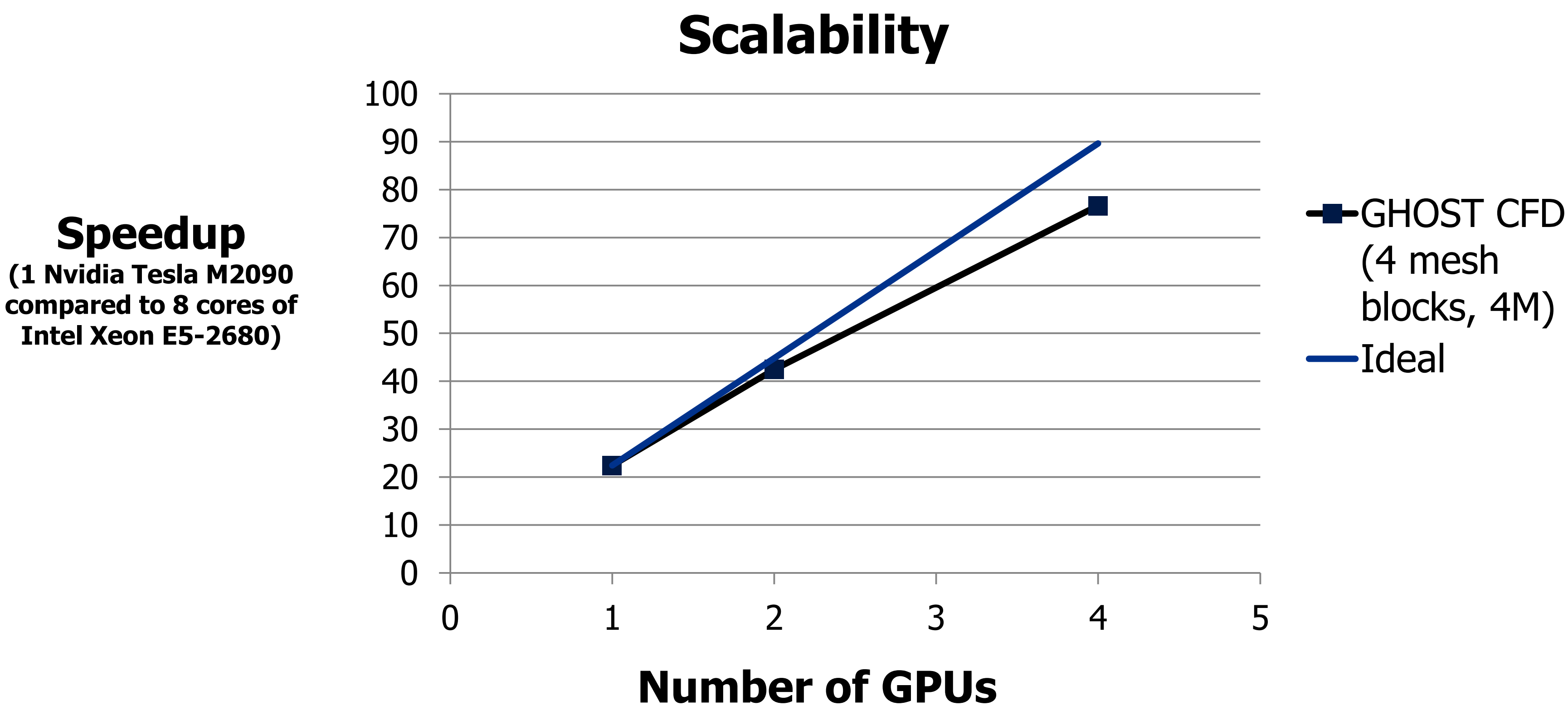
Performance

Memory bandwidth

GHOST CFD Solver: Performance

- **Multiple GPU located on multiple cluster nodes supported**
- **Single GPU speedup:** **<u>12-20x</u>**

  (1 Nvidia Tesla M2090 compared to 8 cores of Intel Xeon E5-2680)



Авиадвигатель
ОДК

Fan noise computation: Outline

- **Allows to optimize fan and outlet guide vane blades for noise reduction. Vital due to acoustic norms tightening.**

- **Approaches:**

  - **Empirical (estimation based on available experimental data)**
    - **Pros: Very fast**
    - **Cons: Requires huge experimental base; Non feasible for new designs.**

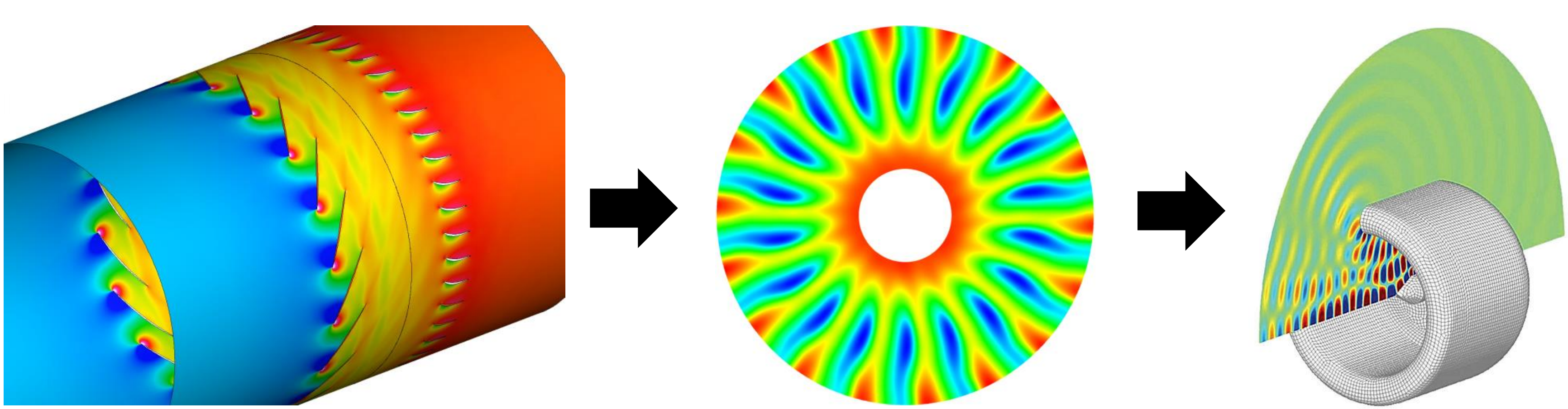
  - **Simplified (2D models, noise sources instead of real blades, generalized periodicity usage etc)**
    - **Pros: Fast**
    - **Cons: Non feasible for modern engine designs (Very hard or impossible to implement for inhomogeneous blade rows; Low blade geometry dependence=> hard to optimize blade)**

  - **Direct (3D computations of blade noise generation followed by the computation of obtained signal propagation)**
    - **Pros: Robust; Free from heavy empiricism; Allows evaluation of new designs;**
    - **Cons: Requires huge computational resources and long computational times, which is highly undesirable for engineering;**

Fan noise computation: Current approach

1. **Computation of noise generation with commercial 3D CFD software (ANSYS Fluent \ CFX)**
2. **Modal decomposition at inlet and outlet planes with commercial MSC Software Actran iTM.**
3. **Noise propagation computation (in axisymmetric setting) with MSC Software Actran TM.**

- The approach is currently under verification.
- Typical computational time: about 1.5 months for one variant (not good)



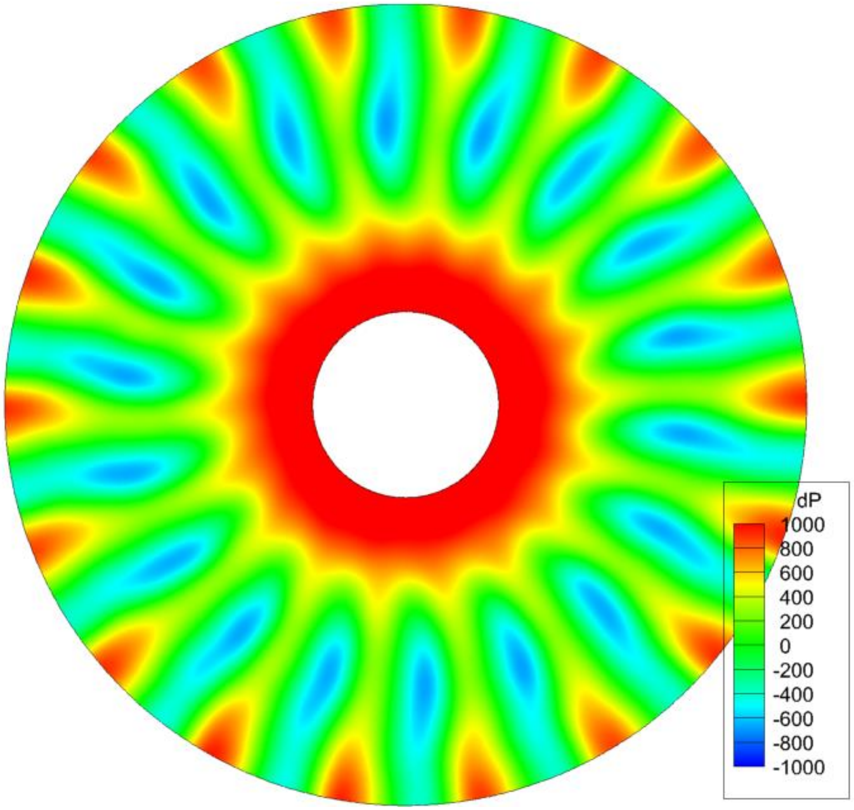
Fan noise computation: GHOST CFD utilization

- GHOST CFD for computation of noise generation:
  - High order schemes, optimized for wave propagation => possibility to use coarser meshes;
  - High performance => faster computations (however computational time profit may vary due to explicit scheme timestep size restriction)

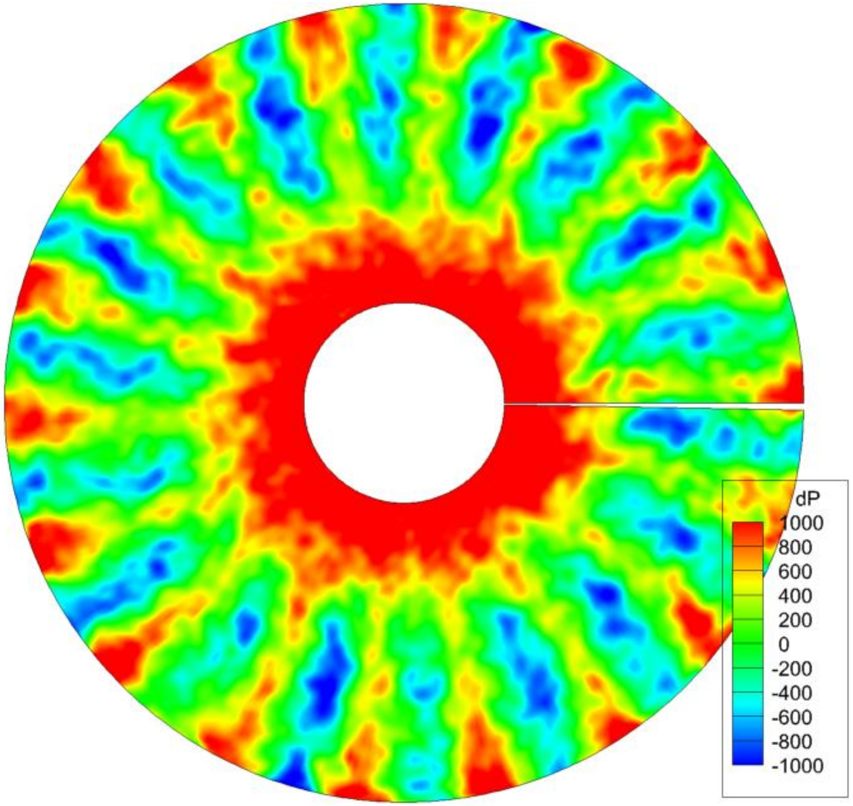
- For model problem :
  - ANSYS CFX:
    - 10-15 days \ revolution (2500 timesteps) with fine mesh (~60M cells)
    - 7 days for coarse mesh (~30M cells)
  - GHOST CFD:
    - 2 days \ revolution (100000 timesteps!) with coarse mesh (~30M cells)

- 3 revolutions needed for complete computation

Fan noise computation: Comparison (Pressure near the fan)

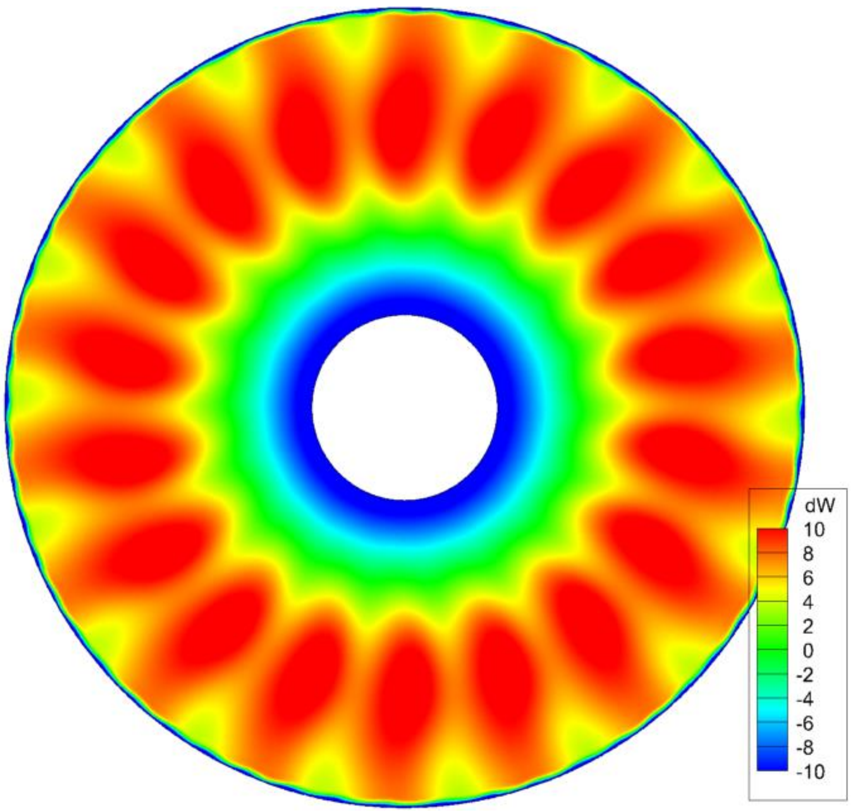


ANSYS CFX

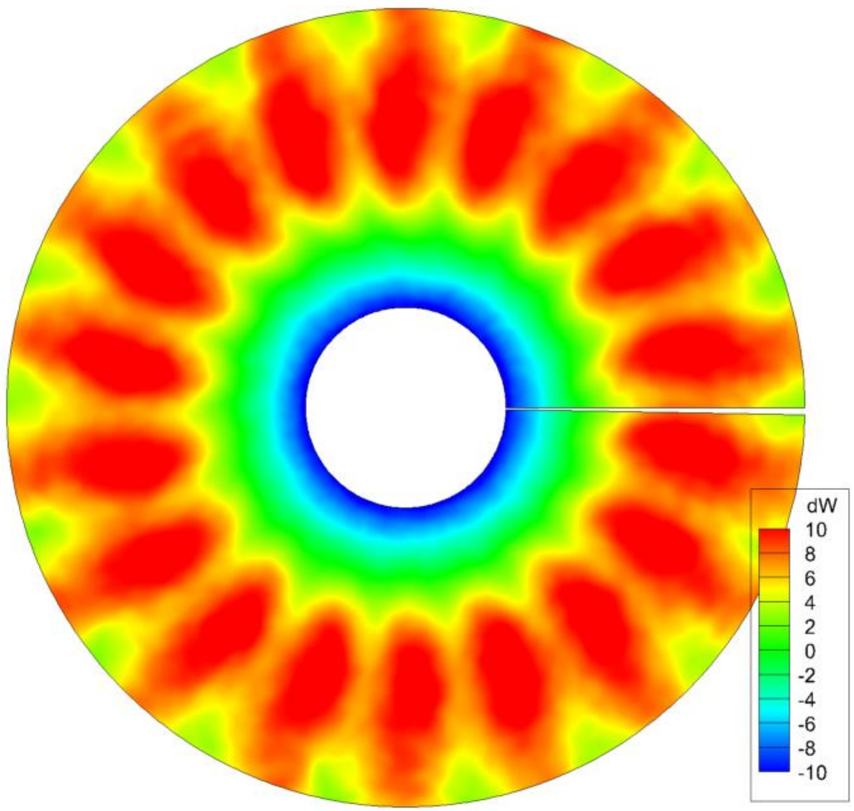


GHOST CFD

Fan noise computation: Comparison (Axial velocity near the fan)

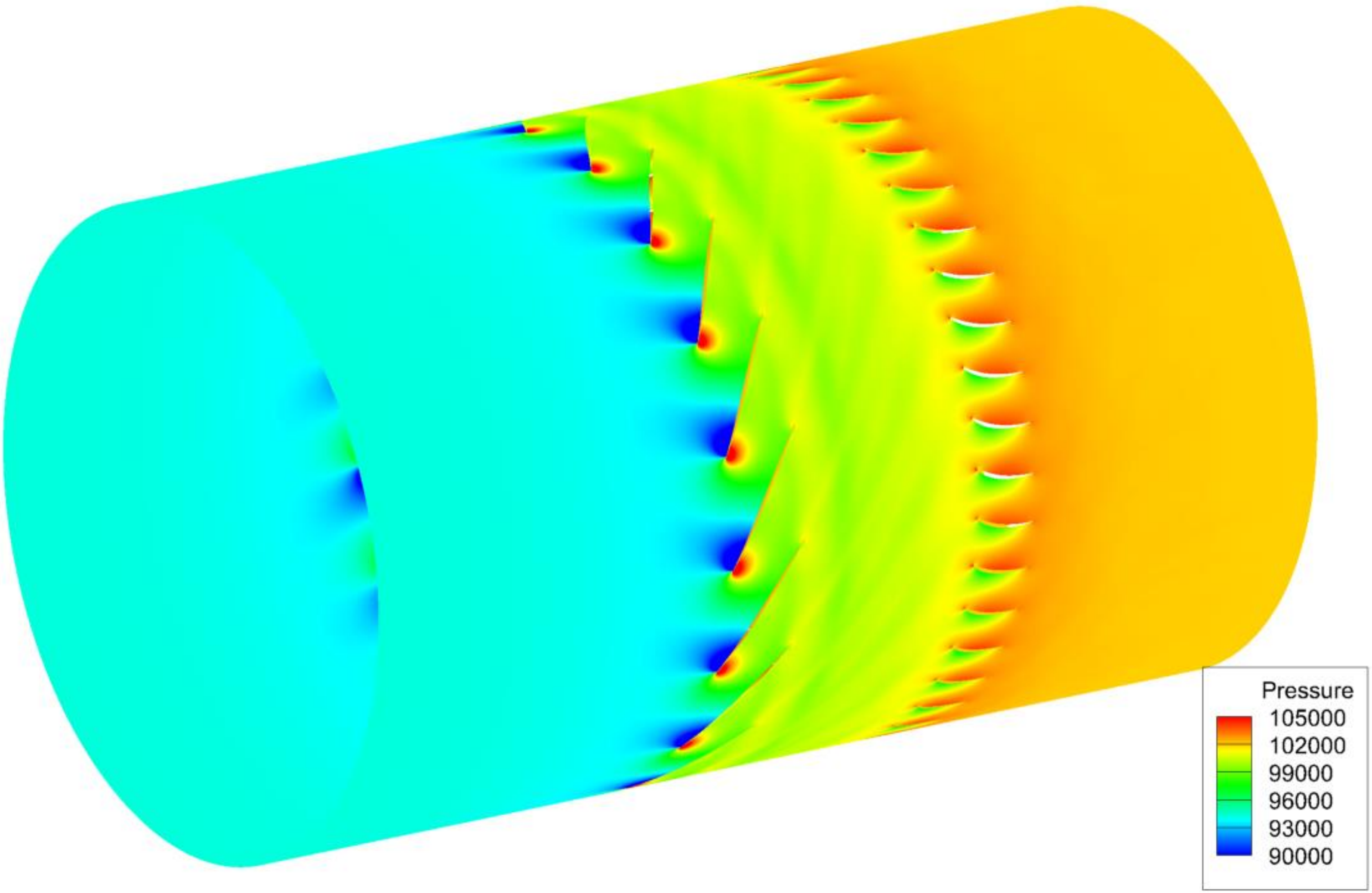


ANSYS CFX



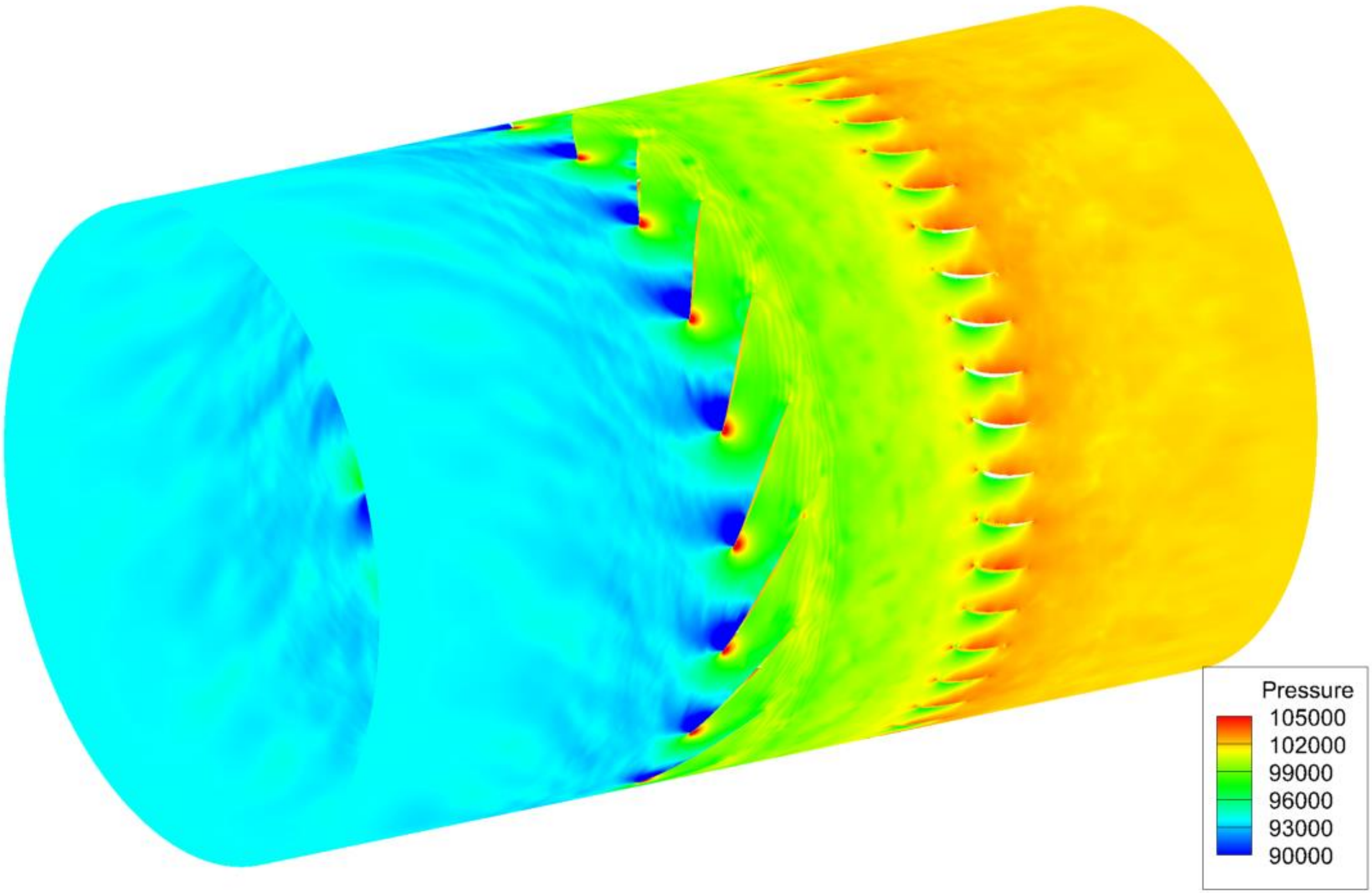
GHOST CFD

Fan noise computation: Comparison (Pressure in the domain) (1)



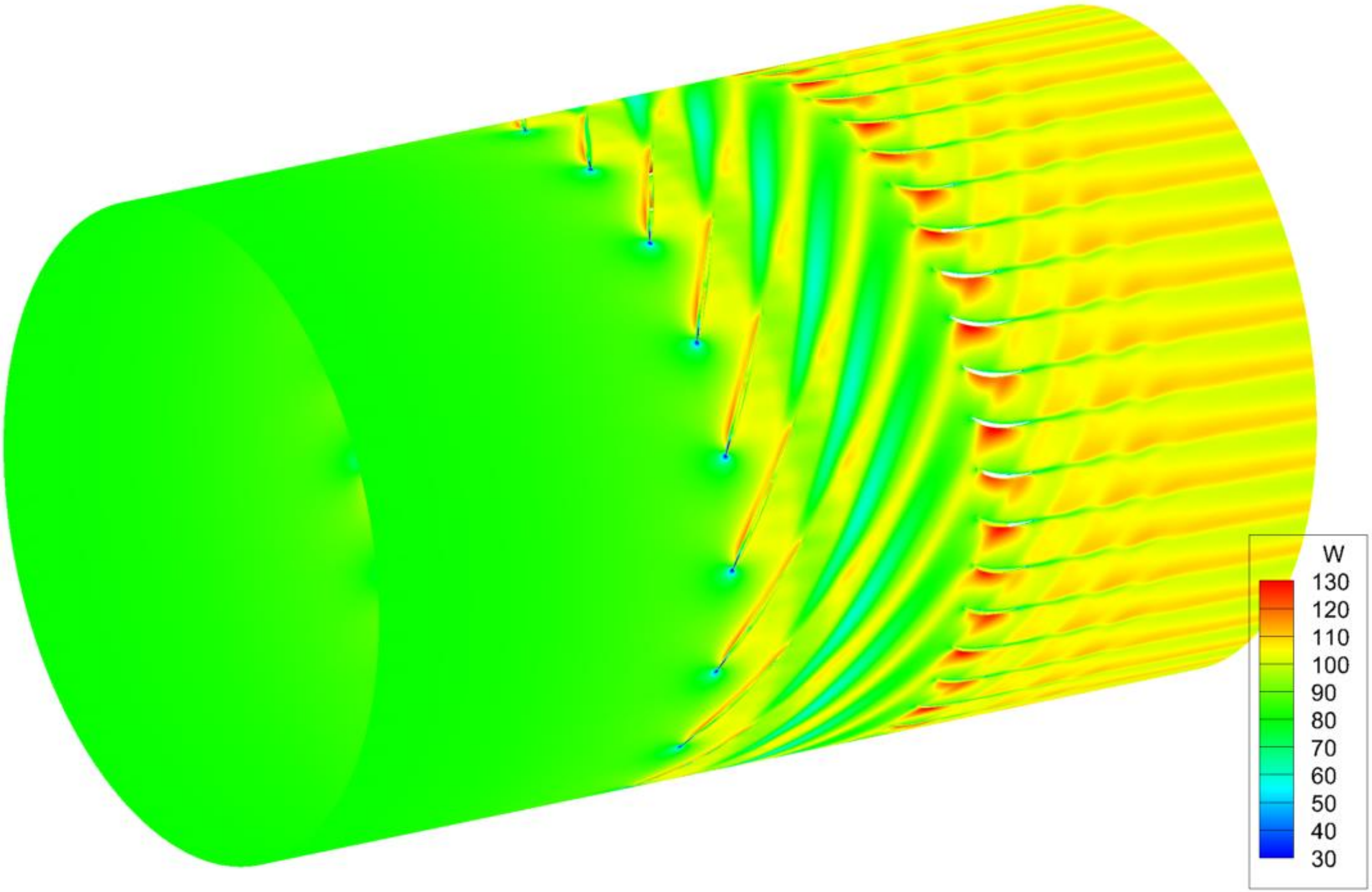
ANSYS CFX

Fan noise computation: Comparison (Pressure in the domain) (2)



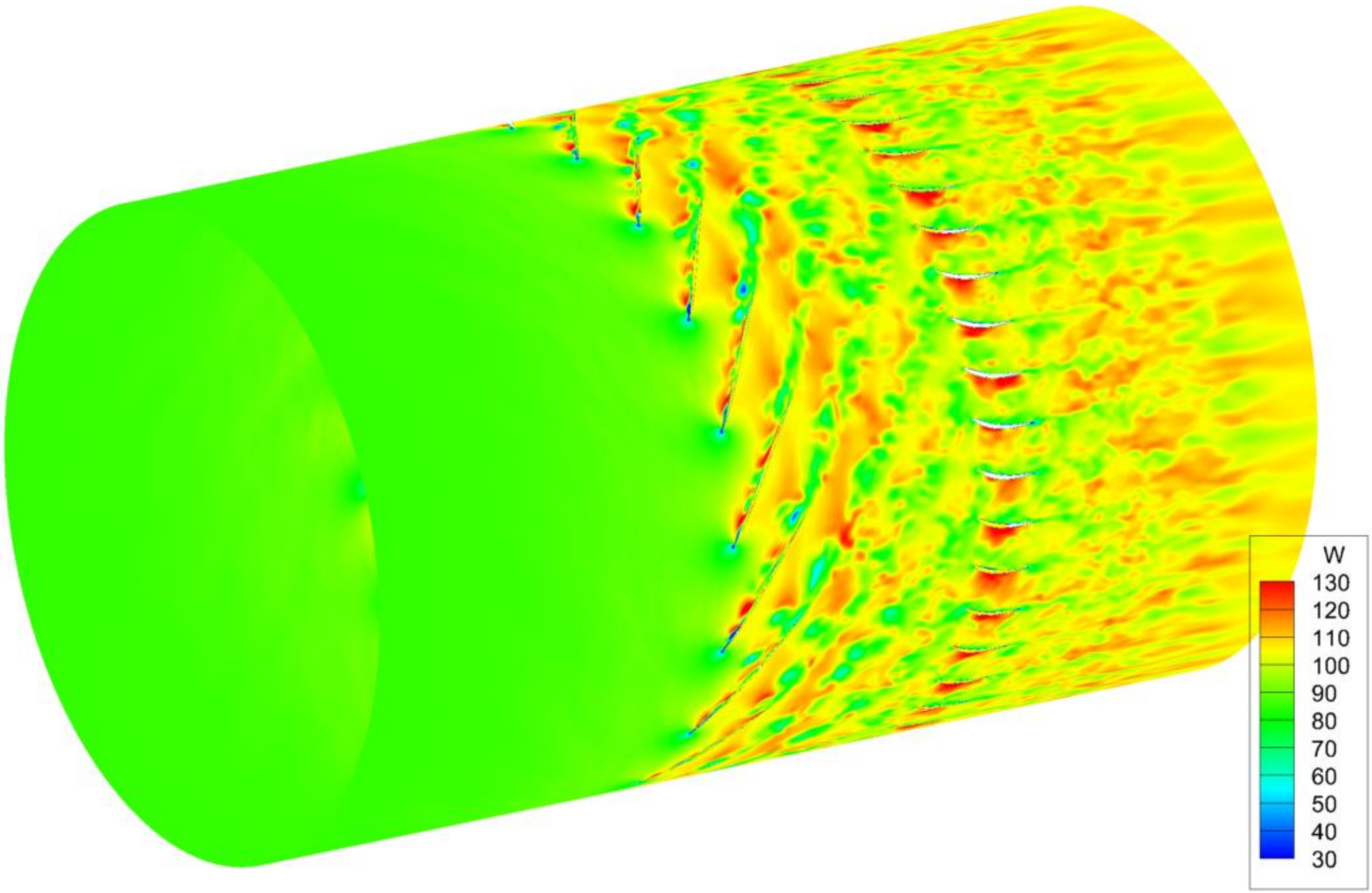
GHOST CFD

Fan noise computation: Comparison (Axial velocity in the domain)



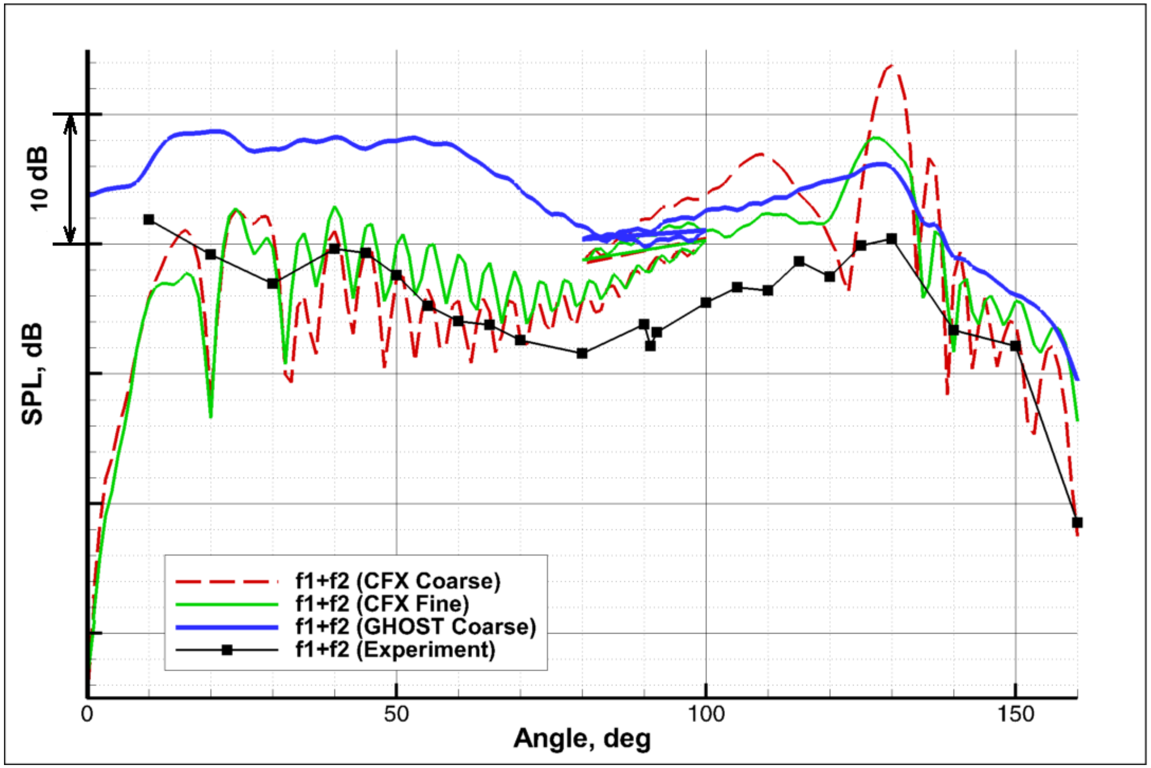
ANSYS CFX

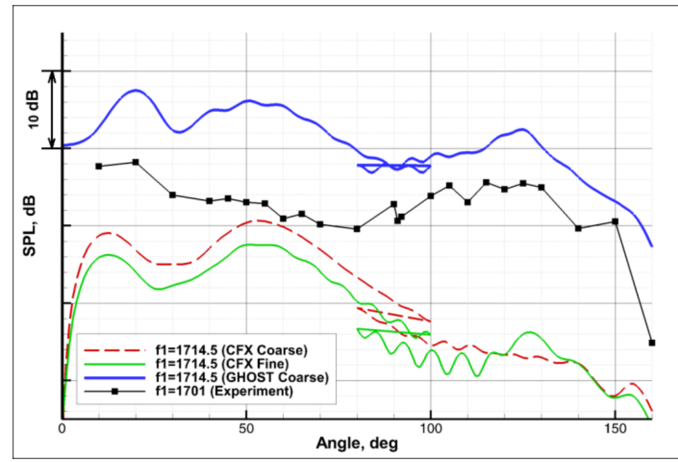
Fan noise computation: Comparison (Velocity in the domain)

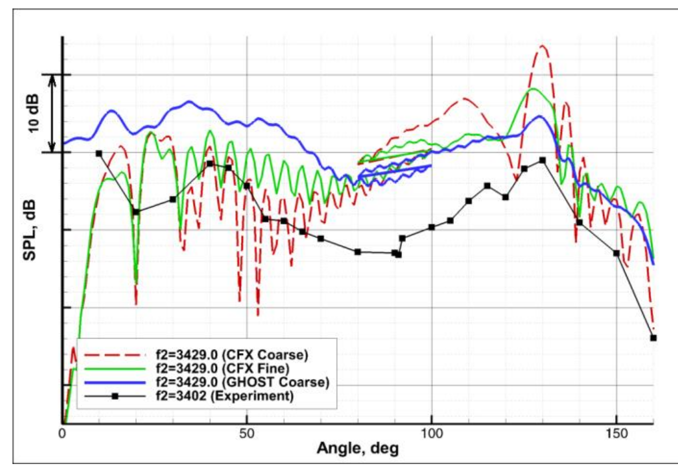


GHOST CFD

Fan noise computation: Results(1)

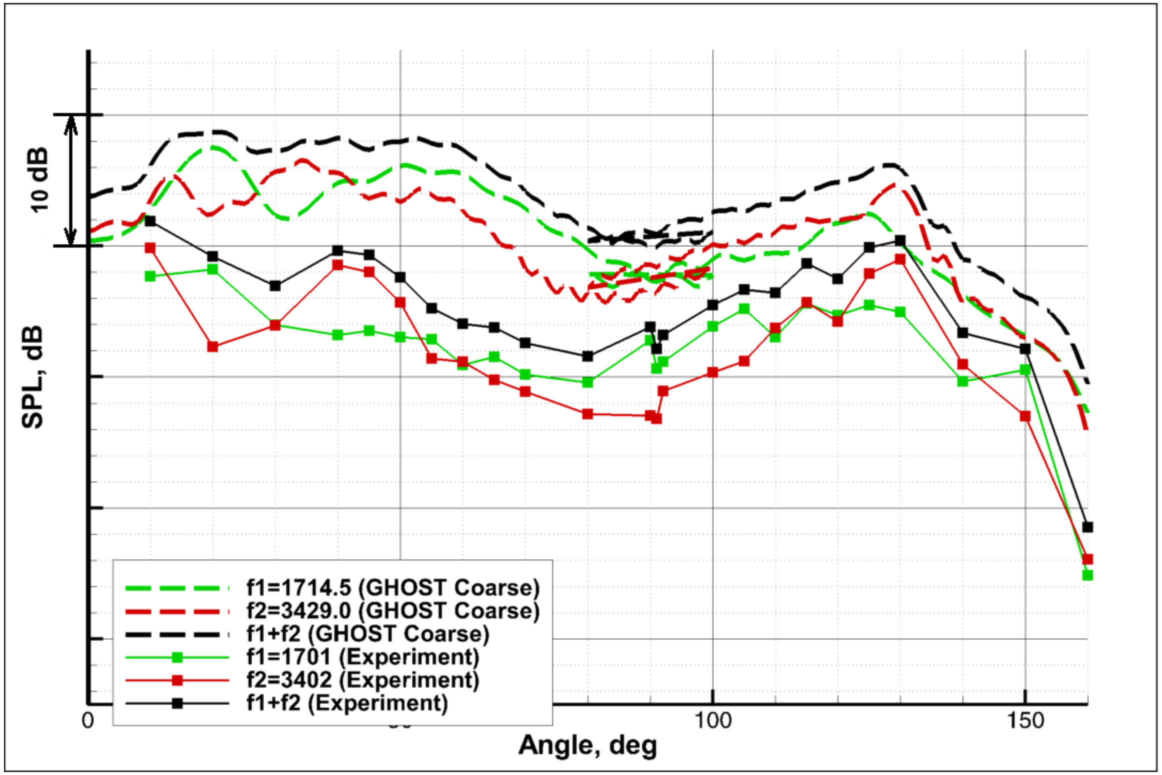




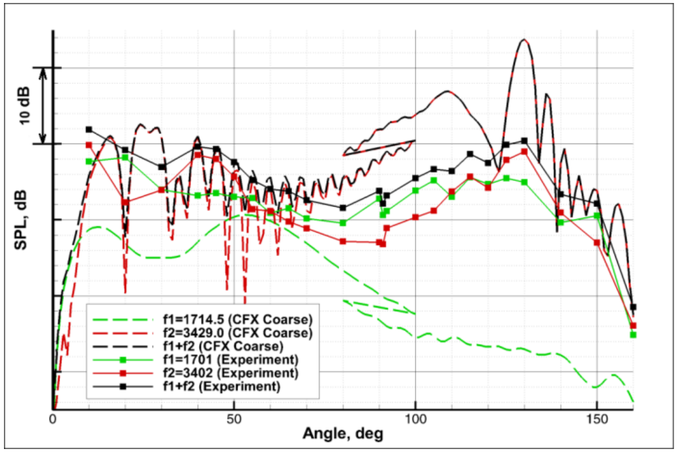


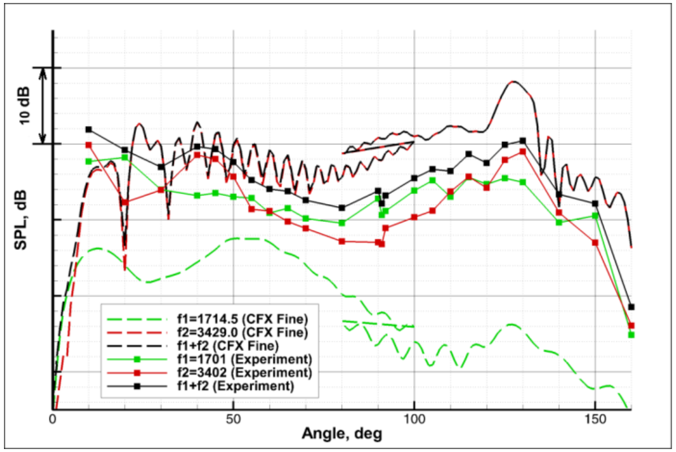
SPL for different frequencies and their sum

Fan noise computation: Results(2)



SPL for different frequencies and their sum





Conclusions & Future work

Conclusions:

- Preliminary work for the computation of fan noise generation with GPU-based GHOST CFD solver was done.
- The solver showed acceleration of simulation about 3 times in terms of physical time compared to commercial software and potentially seems more accurate.
- Current results look promising, however indicate that some solver modifications are needed (presumably wall treatment).

Future work:

- Implementation of wall functions (or other wall treatment)
- Further solver improvement for faster computations (better optimization of the algorithms and code)

Thank you for attention!