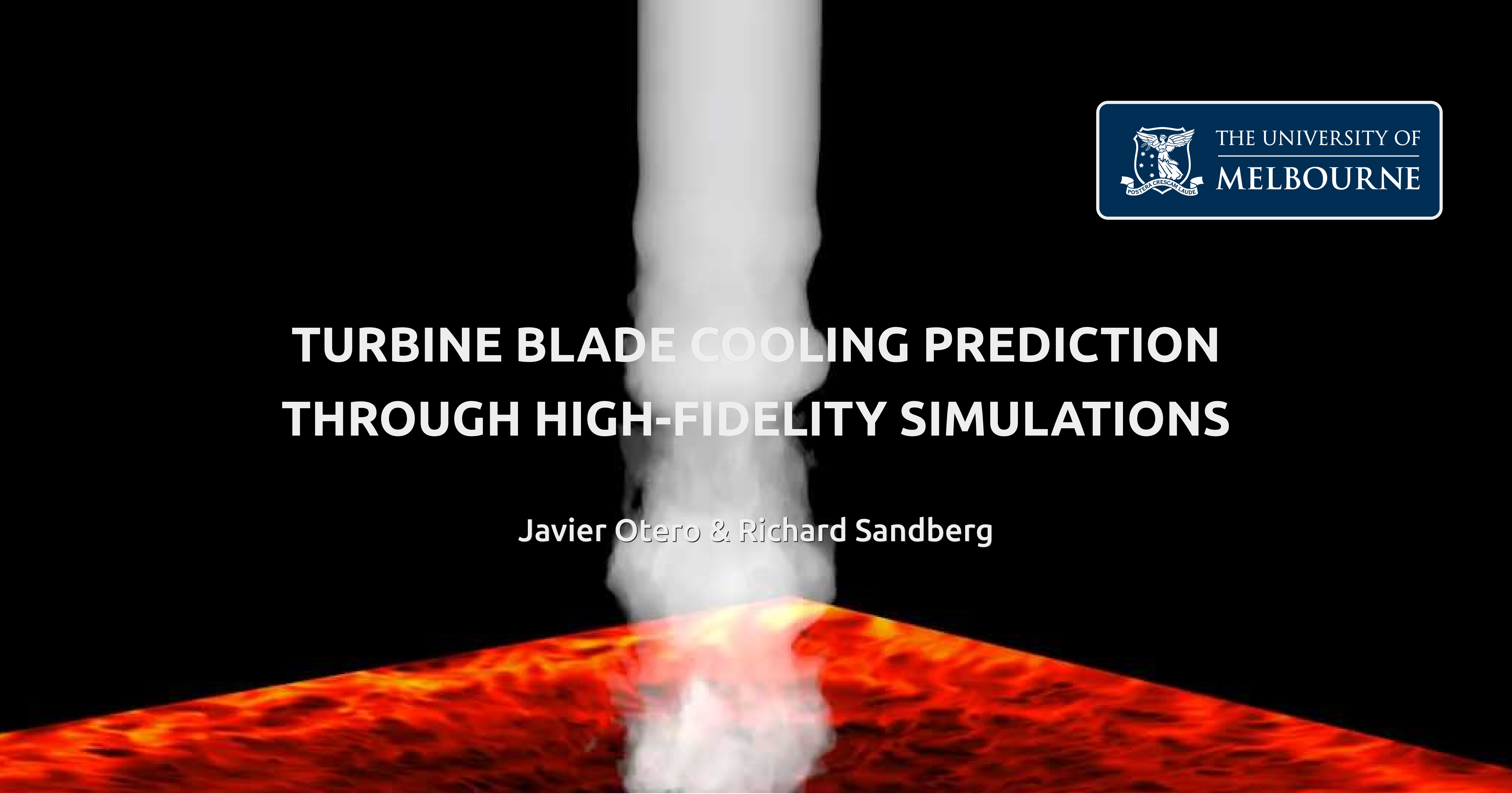
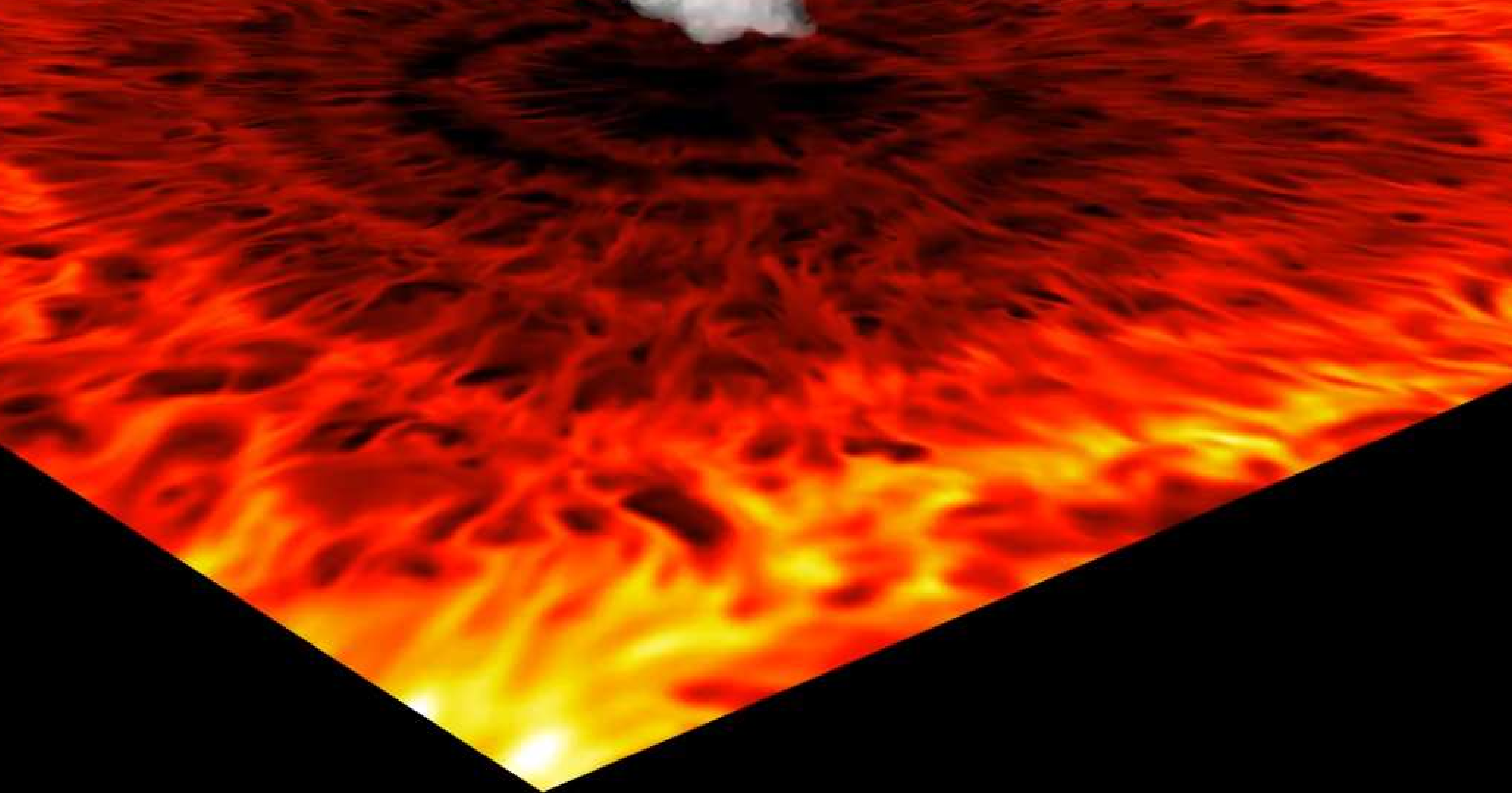


TURBINE BLADE COOLING PREDICTION THROUGH HIGH-FIDELITY SIMULATIONS

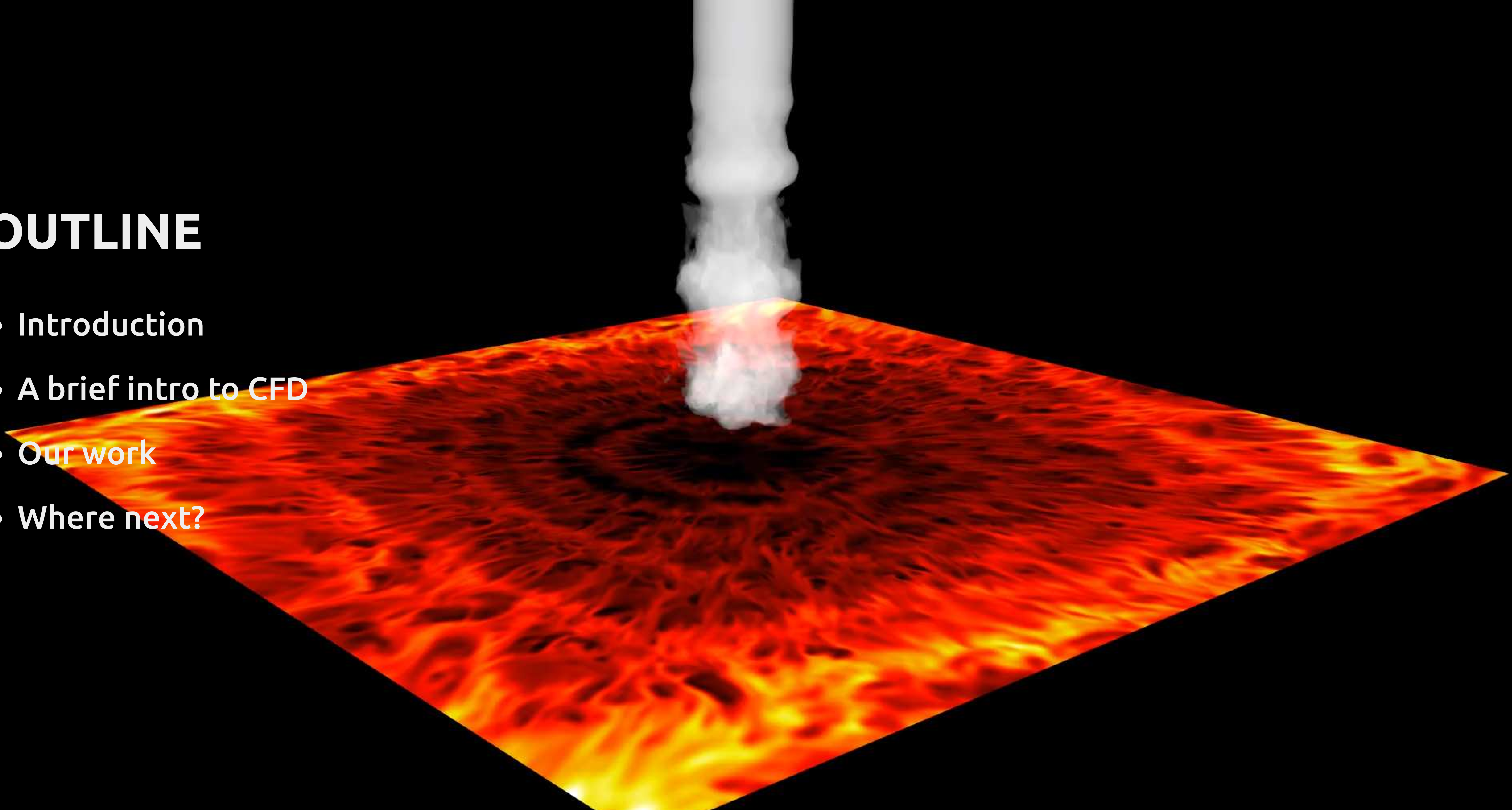
Javier Otero & Richard Sandberg

THE UNIVERSITY OF MELBOURNE



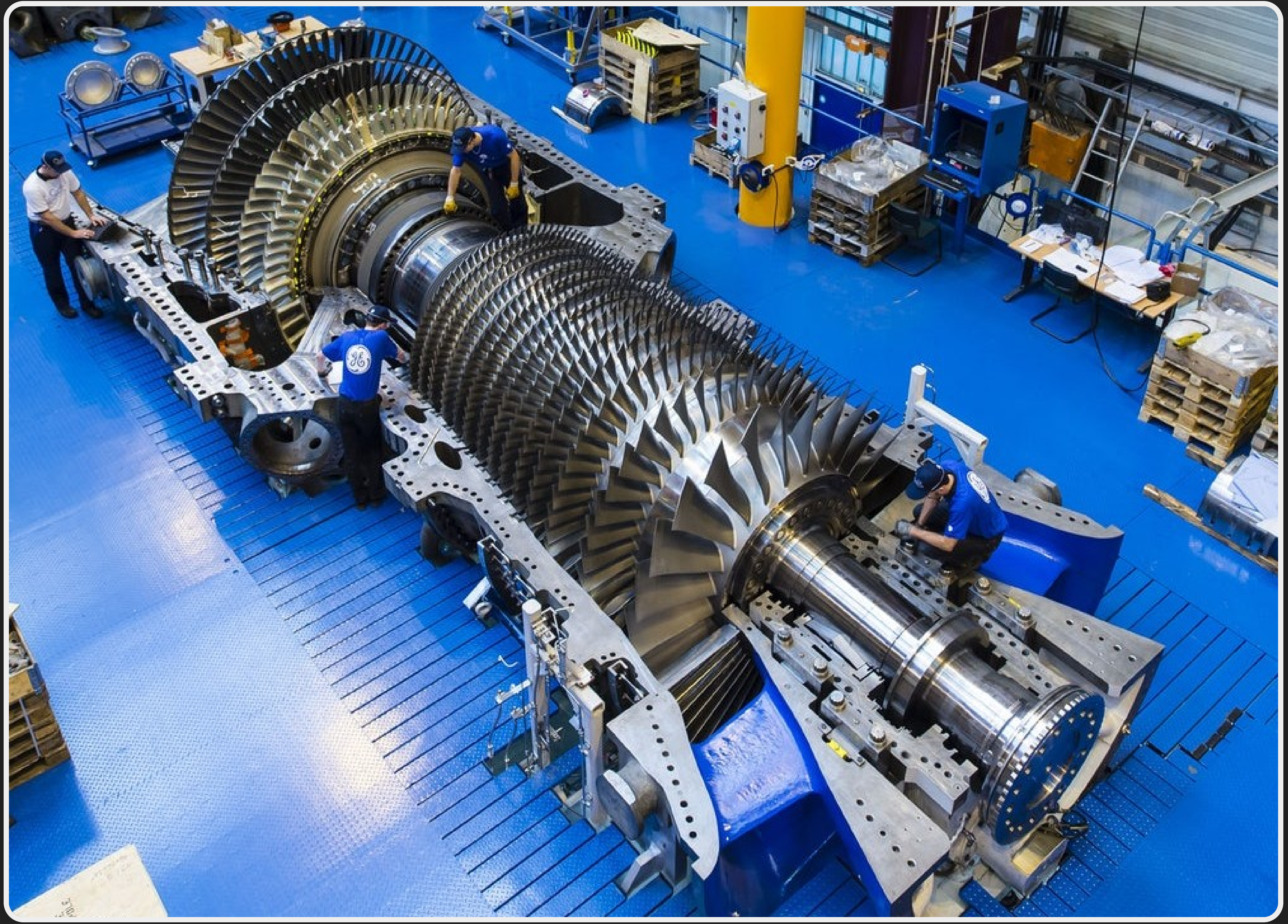
OUTLINE

- Introduction
- A brief intro to CFD
- Our work
- Where next?



TURBINE BLADE COOLING

TURBINE BLADE COOLING

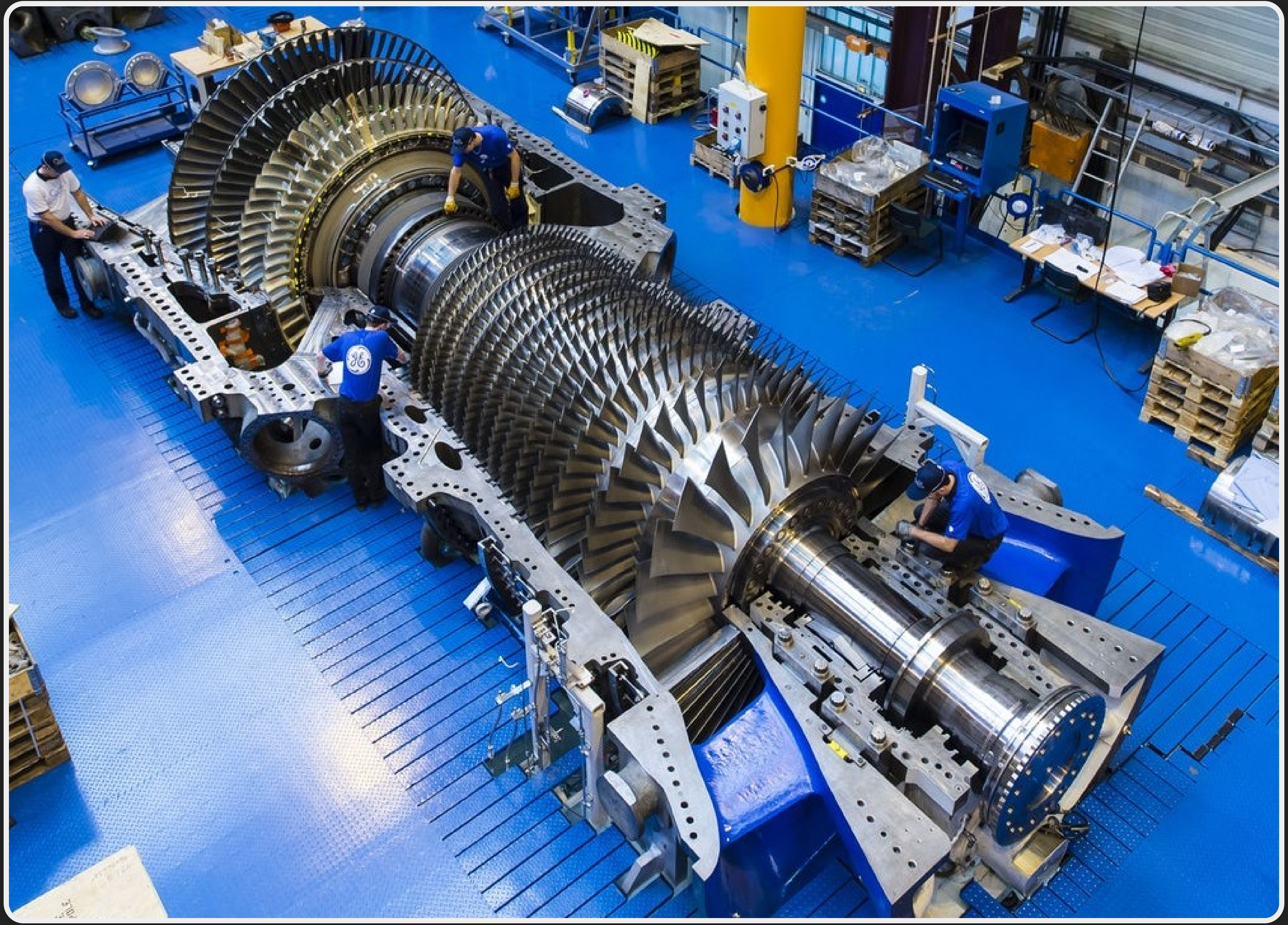


From ge.com

Why do we need cooling on the turbine blades?

- Higher turbine entry temperatures leads to higher thermodynamic efficiency.
- Lower fuel consumption!
- Engine components are exposed to extreme temperatures.
- The most critical components are the turbine blades.
- They operate at temperatures beyond their melting point.
- How is that even possible?
- Because they're fitted with a cooling system.

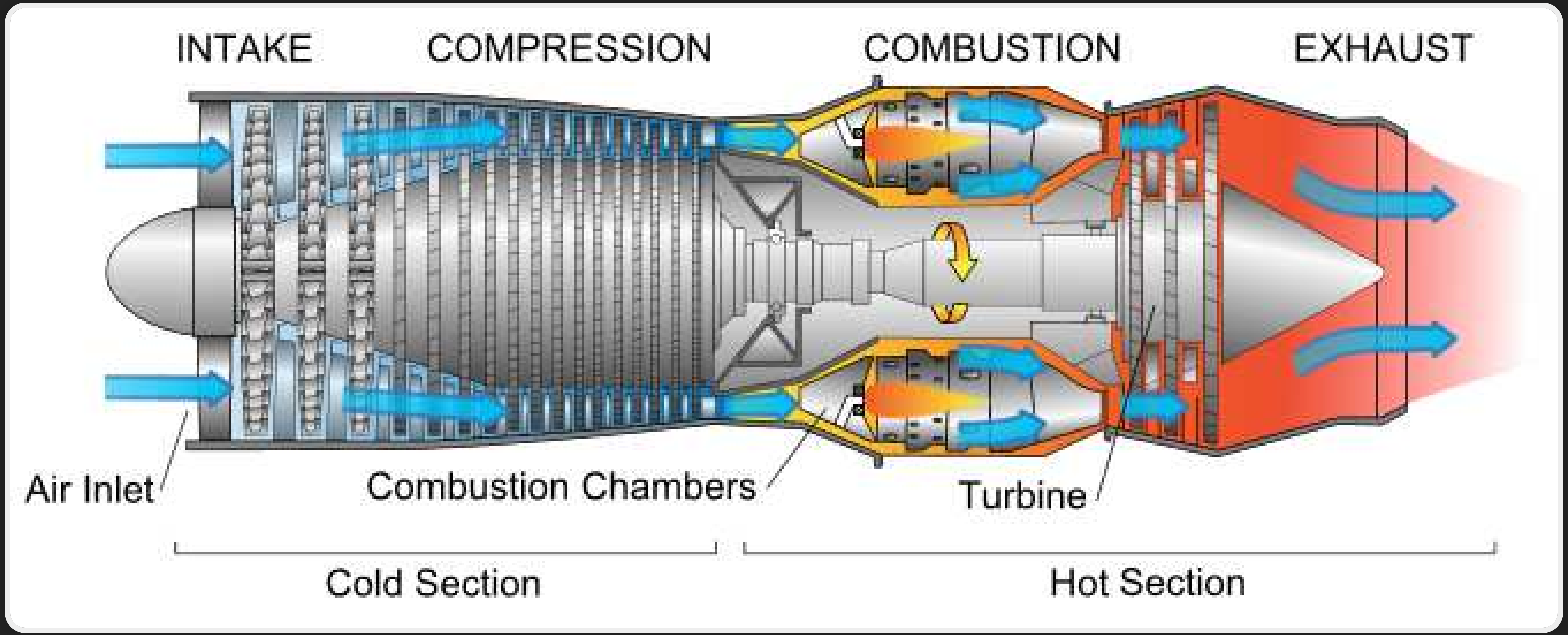
TURBINE BLADE COOLING



From ge.com

How does the cooling work?

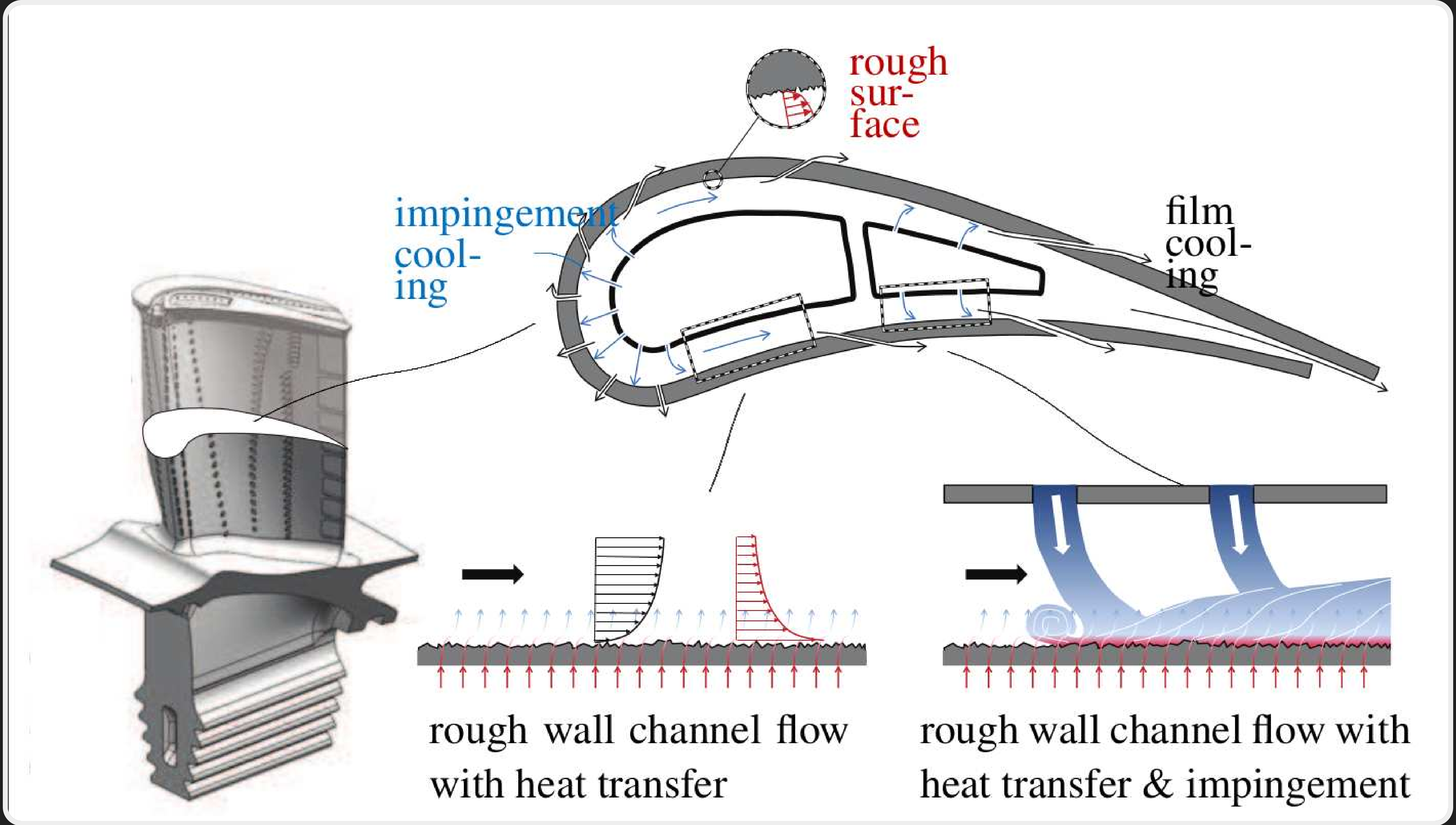
- The cooling system bypasses cold air from the compressor directly into the turbine blades.

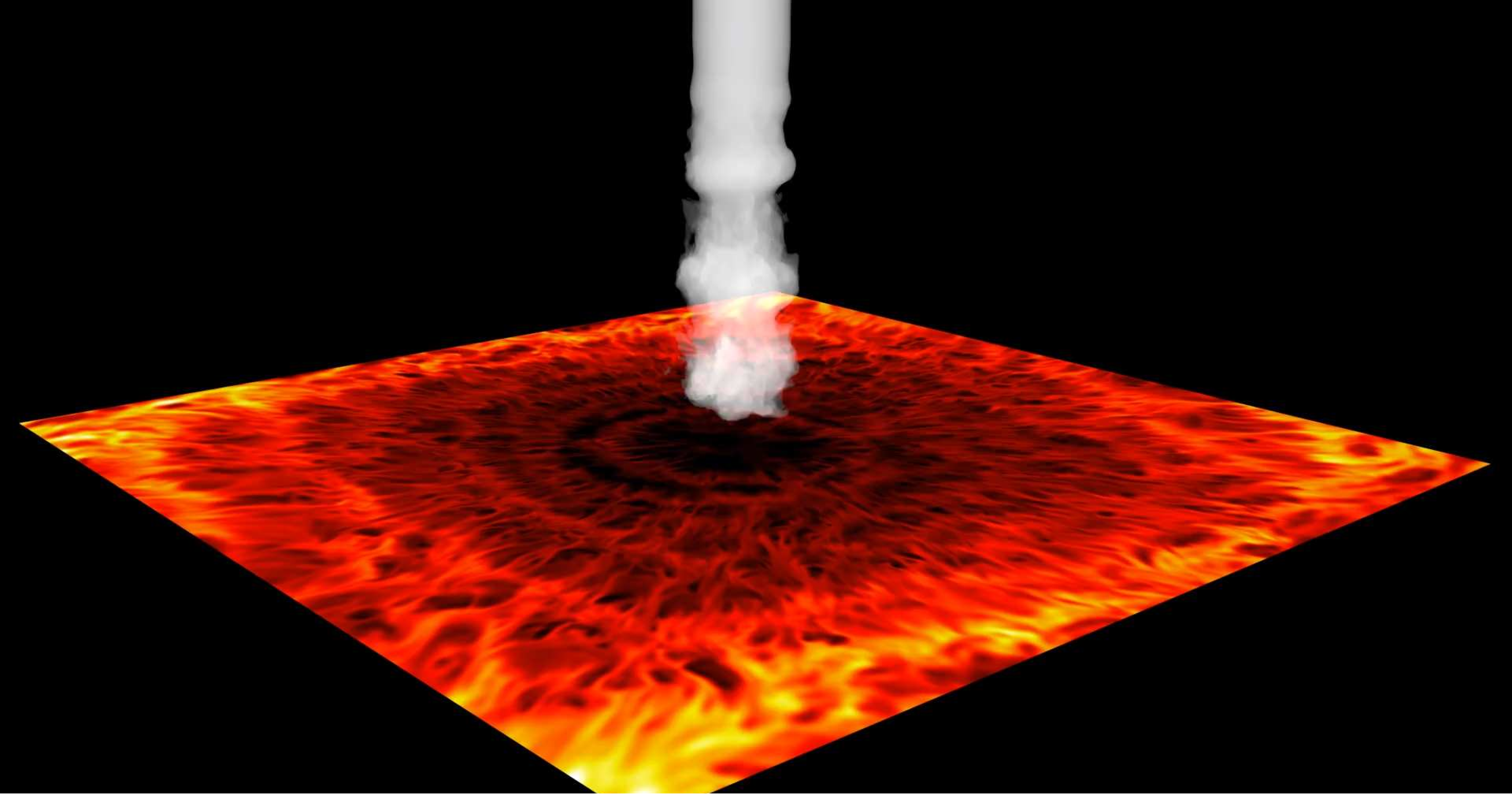


From cset.mnsu.edu

How does the cooling work?

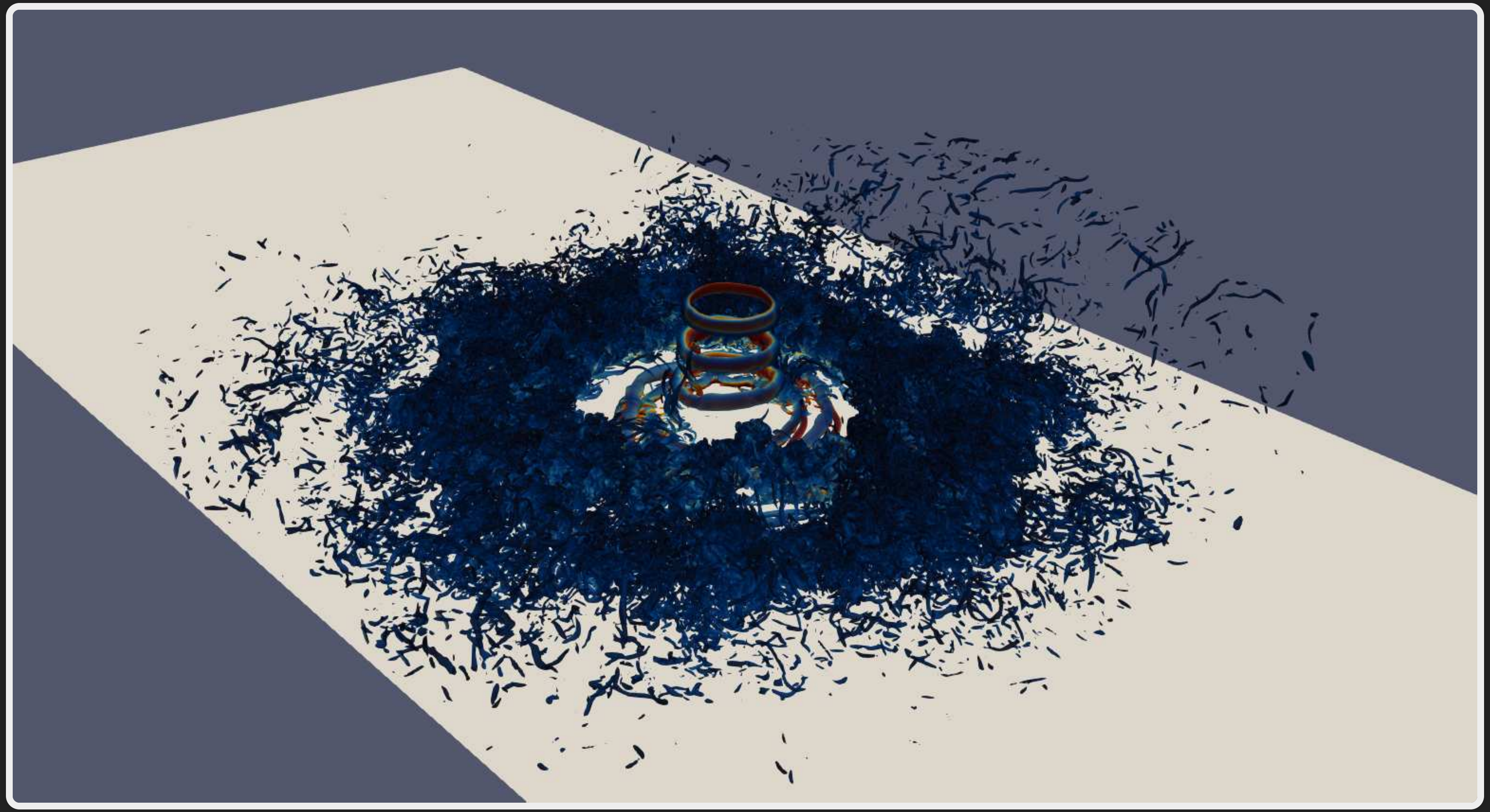
- This cold air injection forms rows of cold jets which impinge onto a heated surface.

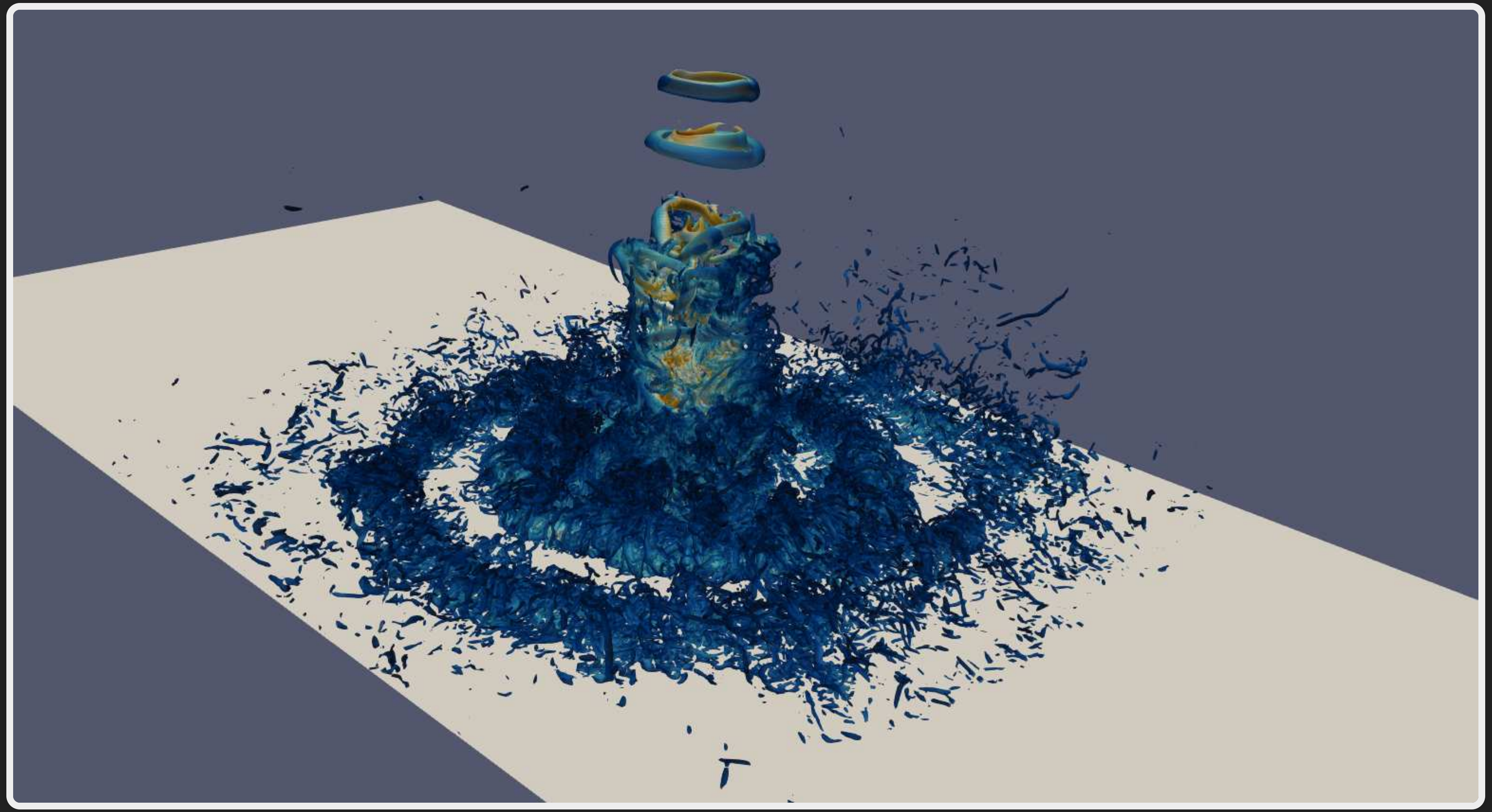




Parameter dependency

- Different nozzle-to-plate distances





Computational Fluid Dynamics

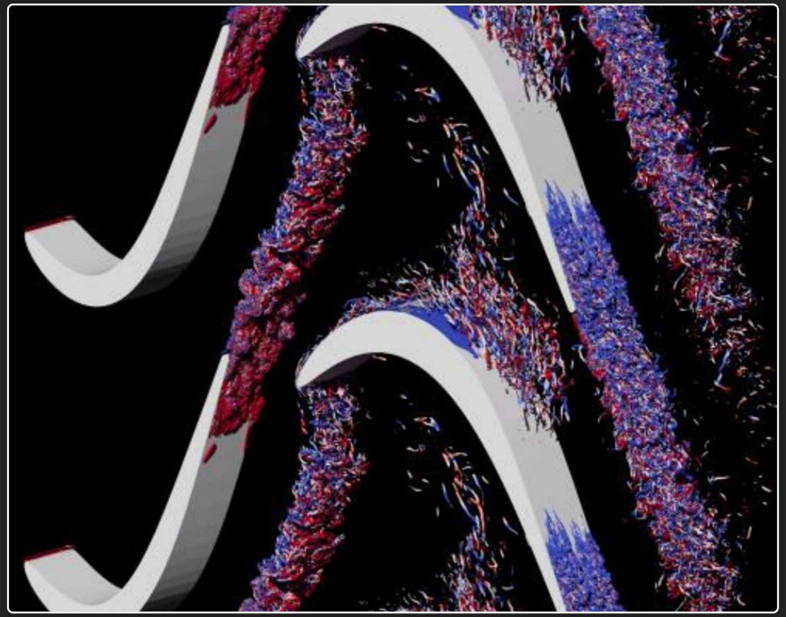
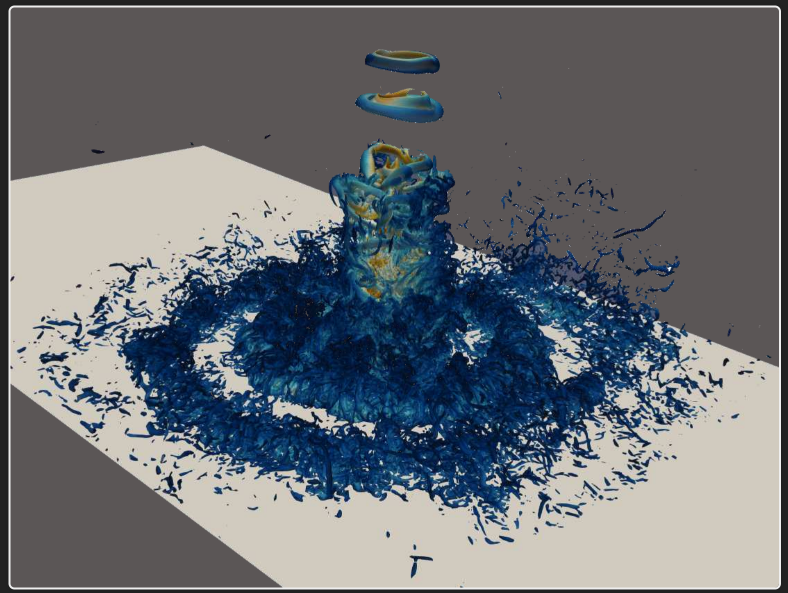
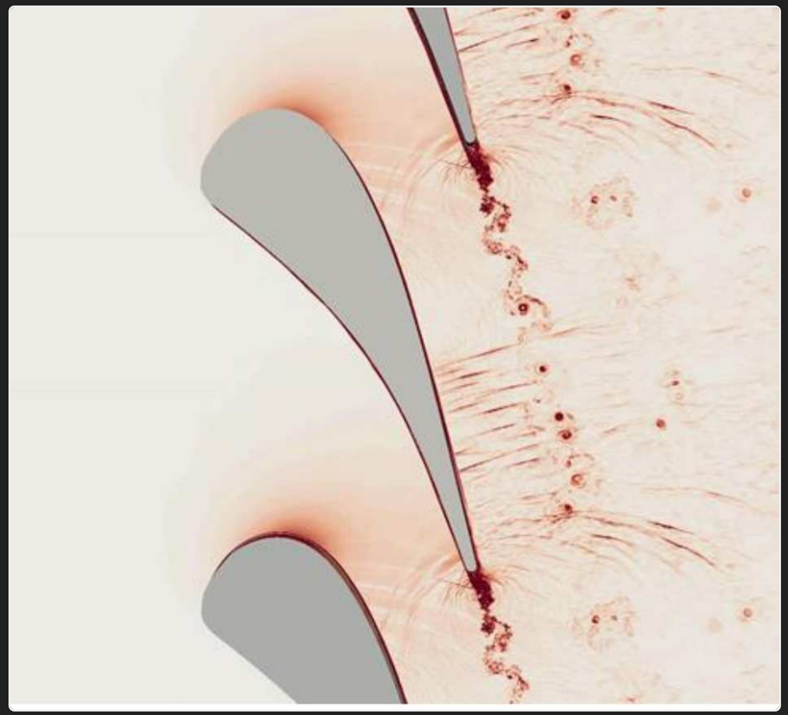
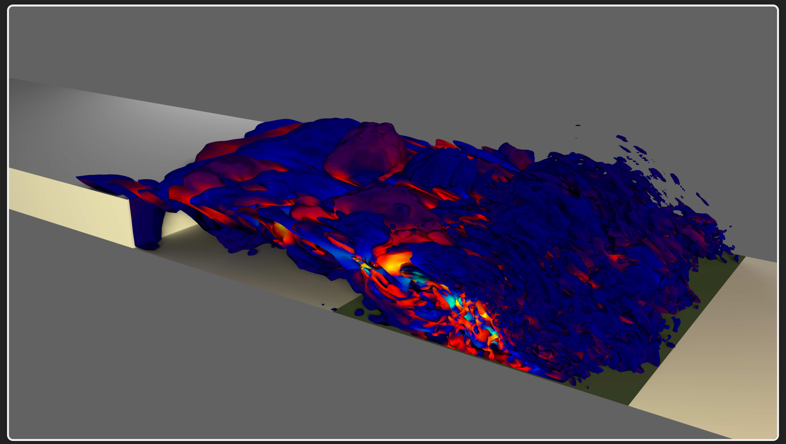
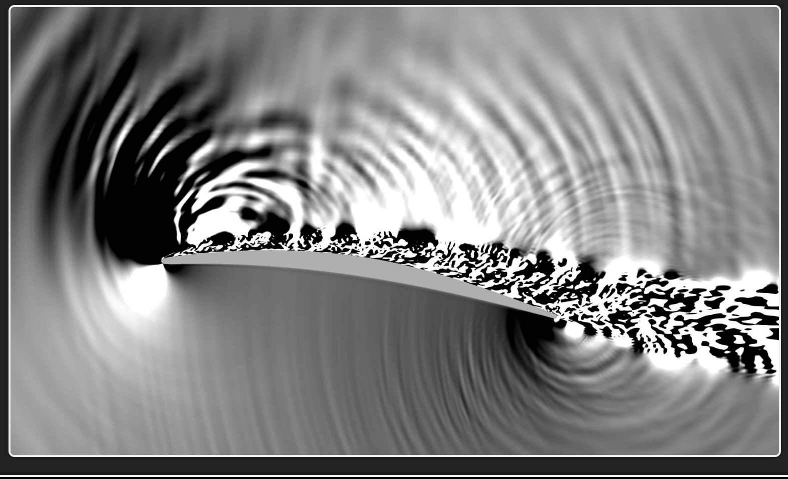
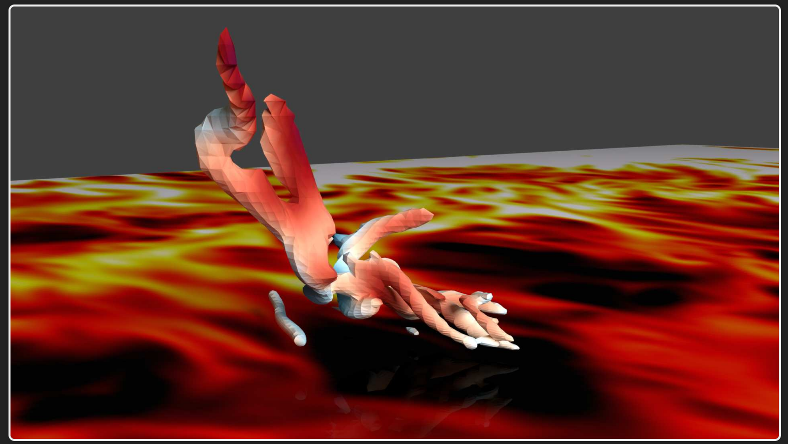
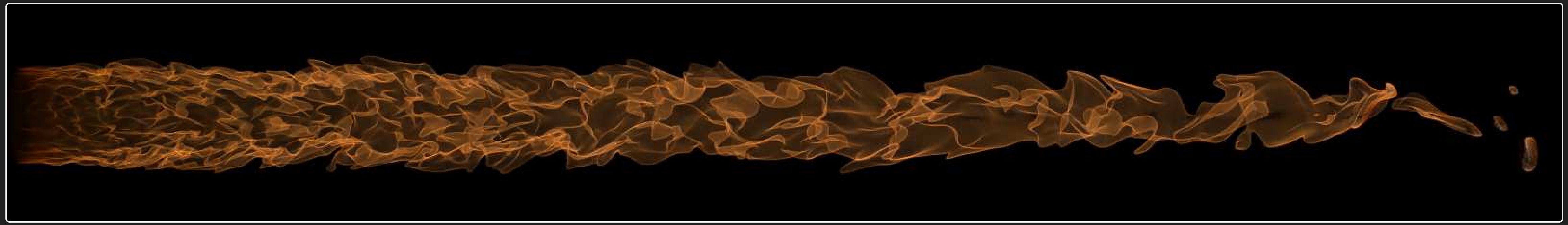
- Governed by the Navier-Stokes equations

- Flow model
  - Compressible Flows
    - High-speed flows & large temperature fluctuations
  - Incompressible
    - Low-speed flows & low temperature fluctuations
    - More complex numerical methods

- Main modelling approaches:
  - Direct Numerical Simulation (DNS) - Very expensive
  - Large Eddy Simulation (LES) - Expensive
  - Reynolds-Averaged Navier-Stokes (RANS) - Cheap

HiPSTAR

High-Performance Solver for Turbulence and Aeroacoustics Research

HiPSTAR

- Compressible solver
- DNS and LES
- Supports complex geometries
- High order of accuracy
  - More computing intensive
  - Lower resolution required
  - Need less storage
- Support for GPU computing
  - Our code is very efficient!

HiPSTAR

- How efficient?

  - CPU vs GPU test at Pawsey Supercomputing Centre (WA)
    - CPU runtime: 1h 31min
    - GPU runtime: 4min 30s
  - Test was on old generation GPUs. New ones are 1.5x faster
  - We've also got more improvements coming up (+20% performance improvement)

- We've got access to the 1st and 5th most powerful supercomputers in the world
  - Summit (Oak Ridge National Laboratory, US)
  - Piz Daint (Swiss National Supercomputing Centre, Switzerland)
  - Full list at top500.org

WHAT HAS BEEN DONE?

A QUICK REVIEW

- Industry
  - Mostly RANS and experimental work
  - Cannot afford to run higher fidelity methods during product design
  - High uncertainty means large safety factors and suboptimal operating conditions
- Academia
  - Experimental studies
  - Incompressible LES at a very low resolution (different results on similar studies)
  - Only one incompressible DNS study

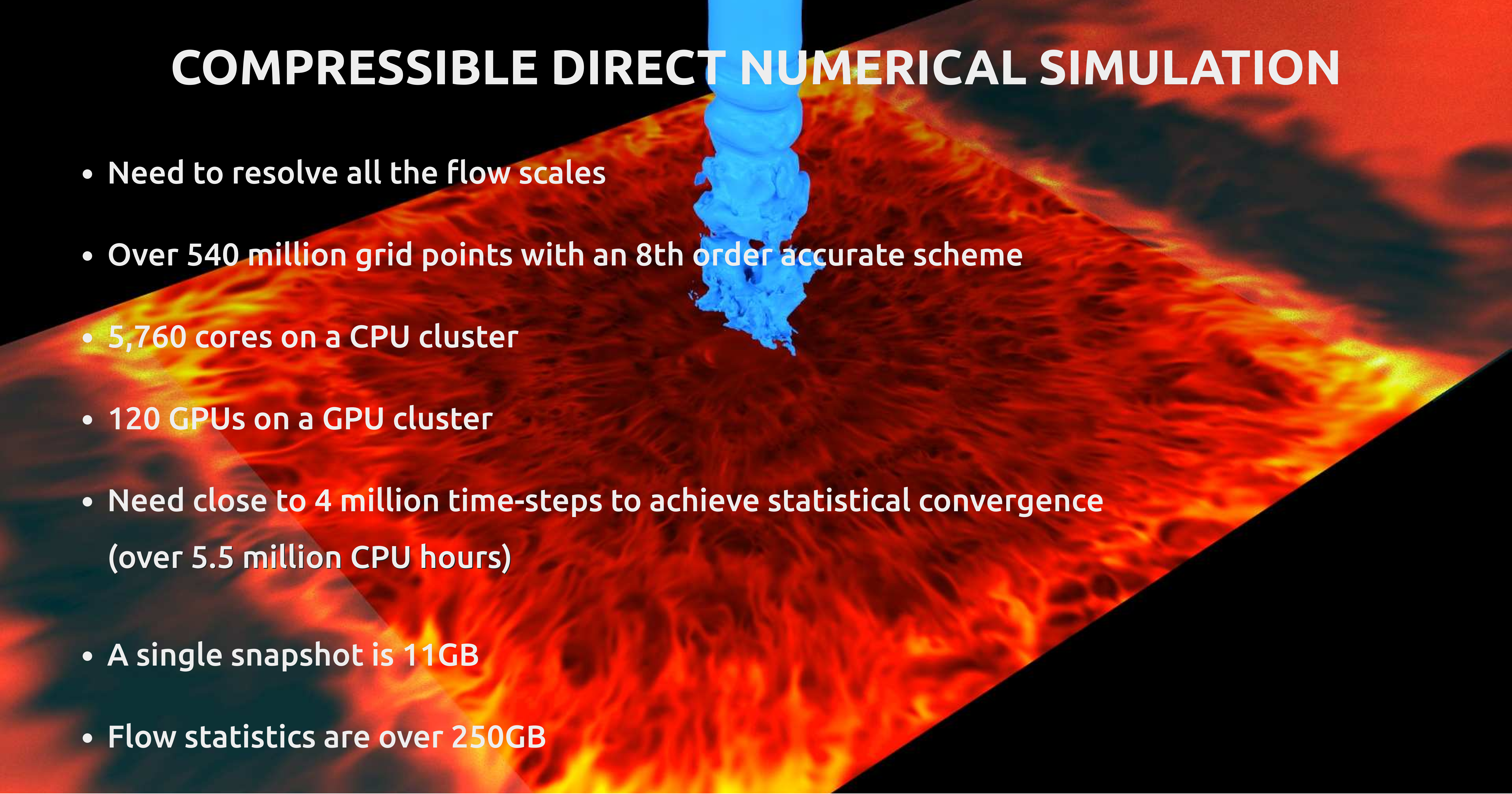
THE GAP

- Need for more high-fidelity numerical studies at realistic engine conditions ($Re \geqslant 10,000$)
- Experimental studies cannot provide the necessary insight to fully understand the cooling phenomena.
- A higher accuracy on the cooling prediction will allow the designers to push their product closer to the optimal regime with confidence
- The more air we use for cooling, the less efficient the engine is

OUR WORK

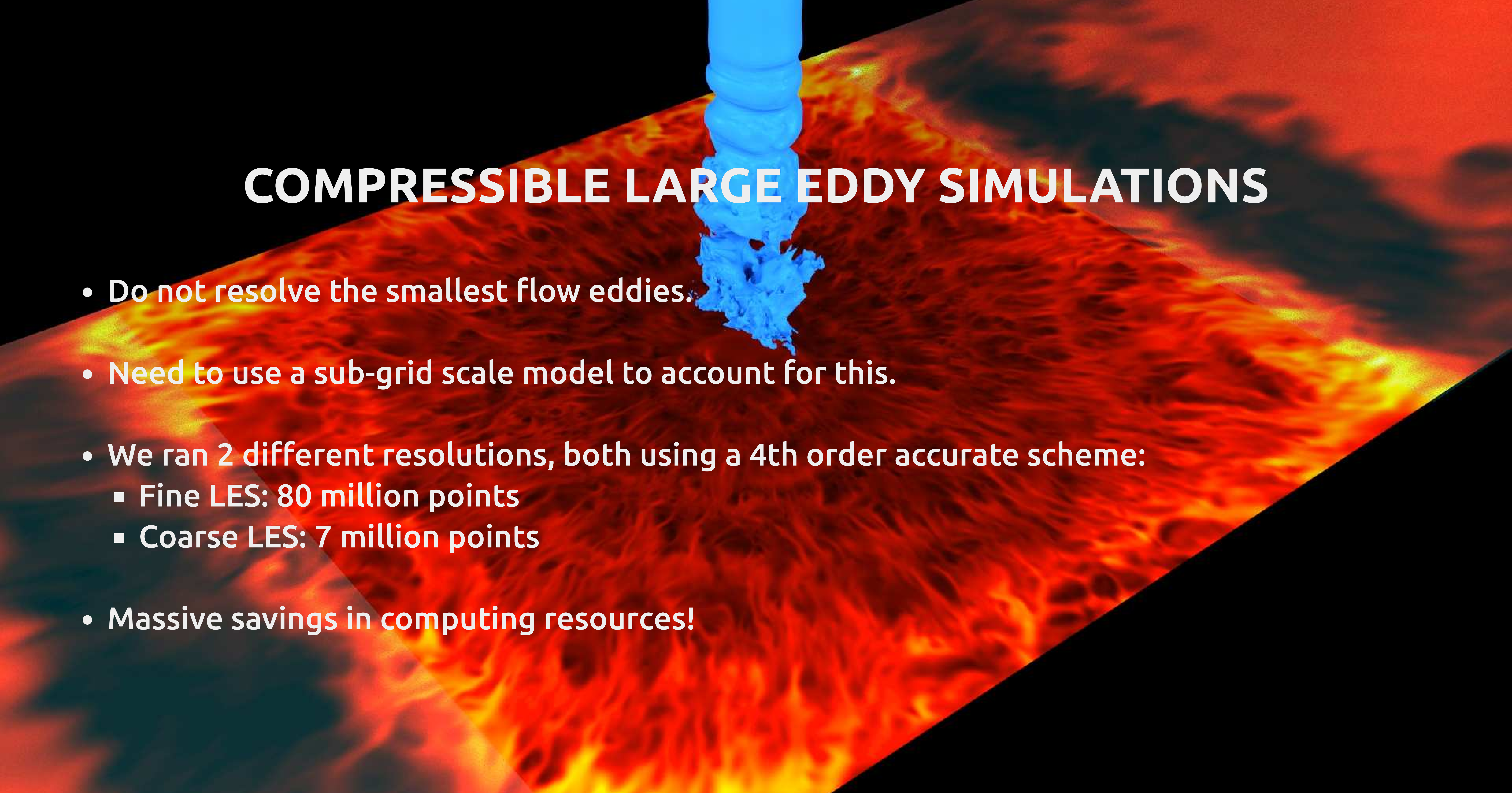
OUR WORK

- Run full compressible DNS
- Investigate how lower fidelity methods (LES or RANS) compare
- Study the interaction of multiple jets
- Study the effect of cross-flow



COMPRESSIBLE DIRECT NUMERICAL SIMULATION

- Need to resolve all the flow scales
- Over 540 million grid points with an 8th order accurate scheme
- 5,760 cores on a CPU cluster
- 120 GPUs on a GPU cluster
- Need close to 4 million time-steps to achieve statistical convergence (over 5.5 million CPU hours)
- A single snapshot is 11GB
- Flow statistics are over 250GB

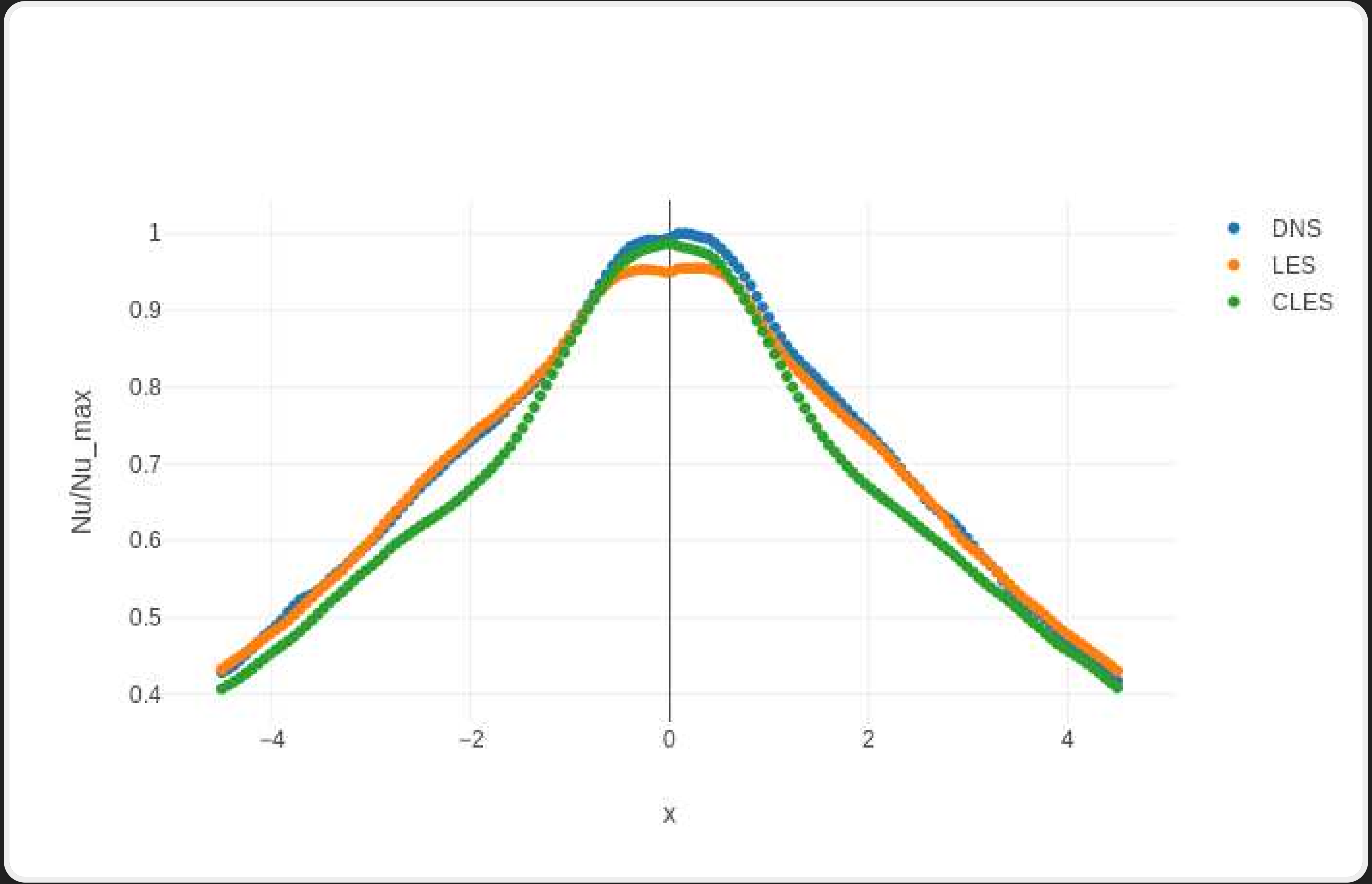


COMPRESSIBLE LARGE EDDY SIMULATIONS

- Do not resolve the smallest flow eddies.
- Need to use a sub-grid scale model to account for this.
- We ran 2 different resolutions, both using a 4th order accurate scheme:
  - Fine LES: 80 million points
  - Coarse LES: 7 million points
- Massive savings in computing resources!

Comparing results

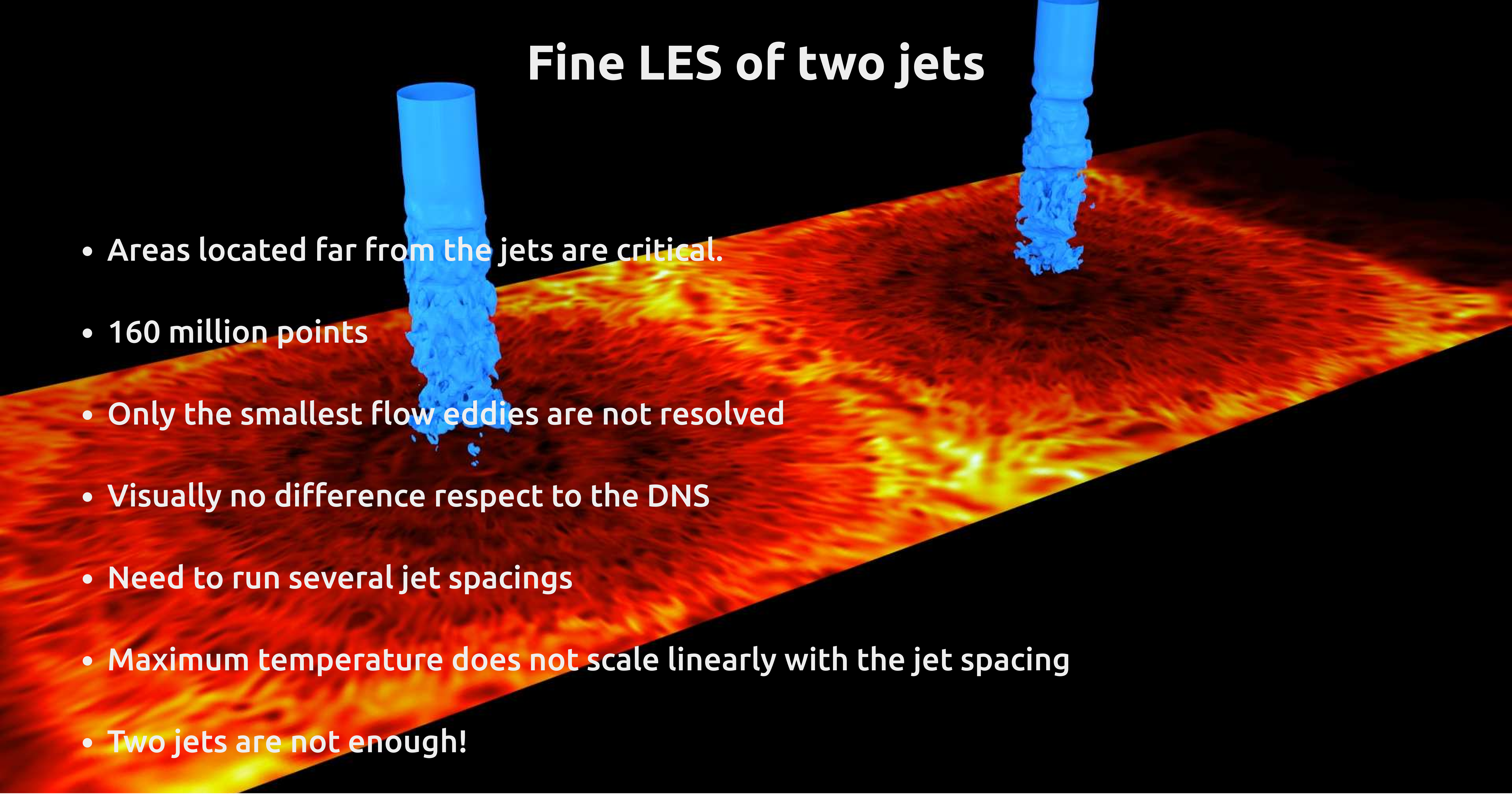
- Cooling at the impingement wall



Comparing results

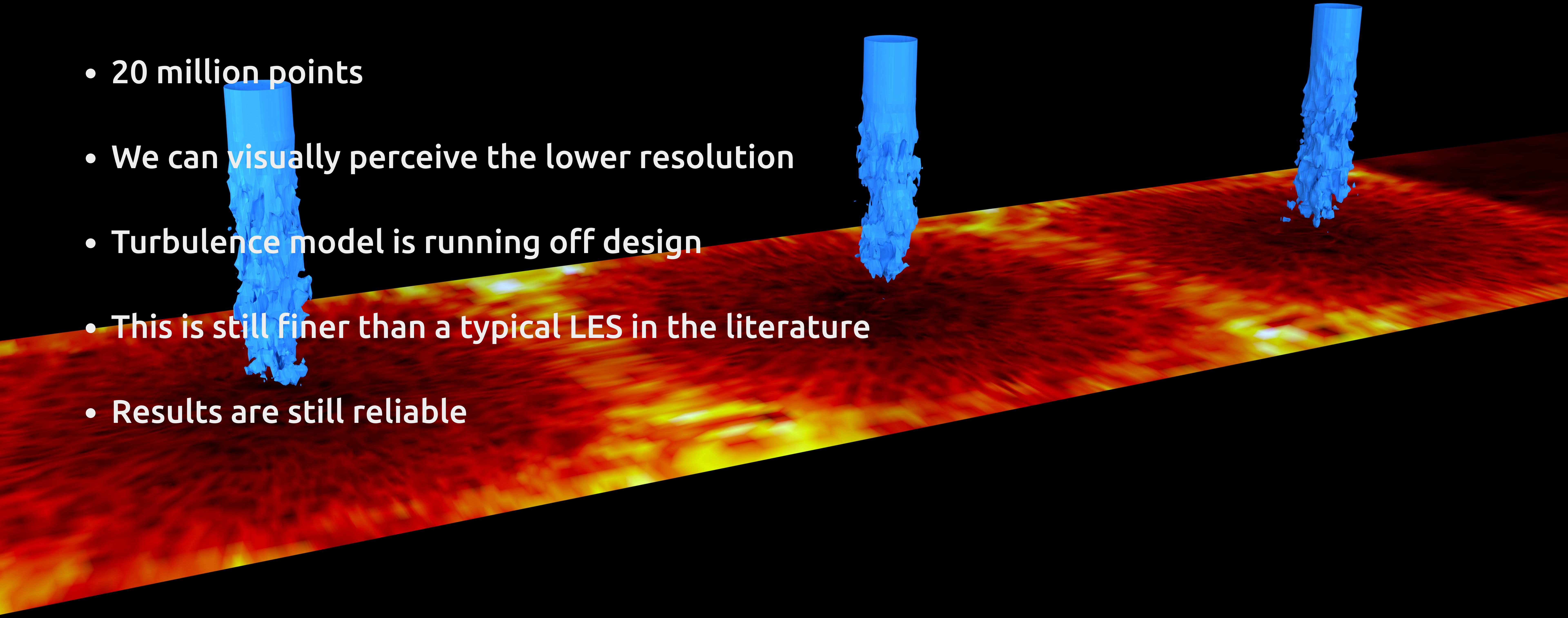
- The fine LES virtually matches the DNS results
- The coarse LES shows a fairly good agreement (a DNS is 110 times more expensive)
- RANS results (OpenFOAM) are far from good
- We can confidently use any LES to run a parametric study

Fine LES of two jets



- Areas located far from the jets are critical.
- 160 million points
- Only the smallest flow eddies are not resolved
- Visually no difference respect to the DNS
- Need to run several jet spacings
- Maximum temperature does not scale linearly with the jet spacing
- Two jets are not enough!

Coarse LES of three jets

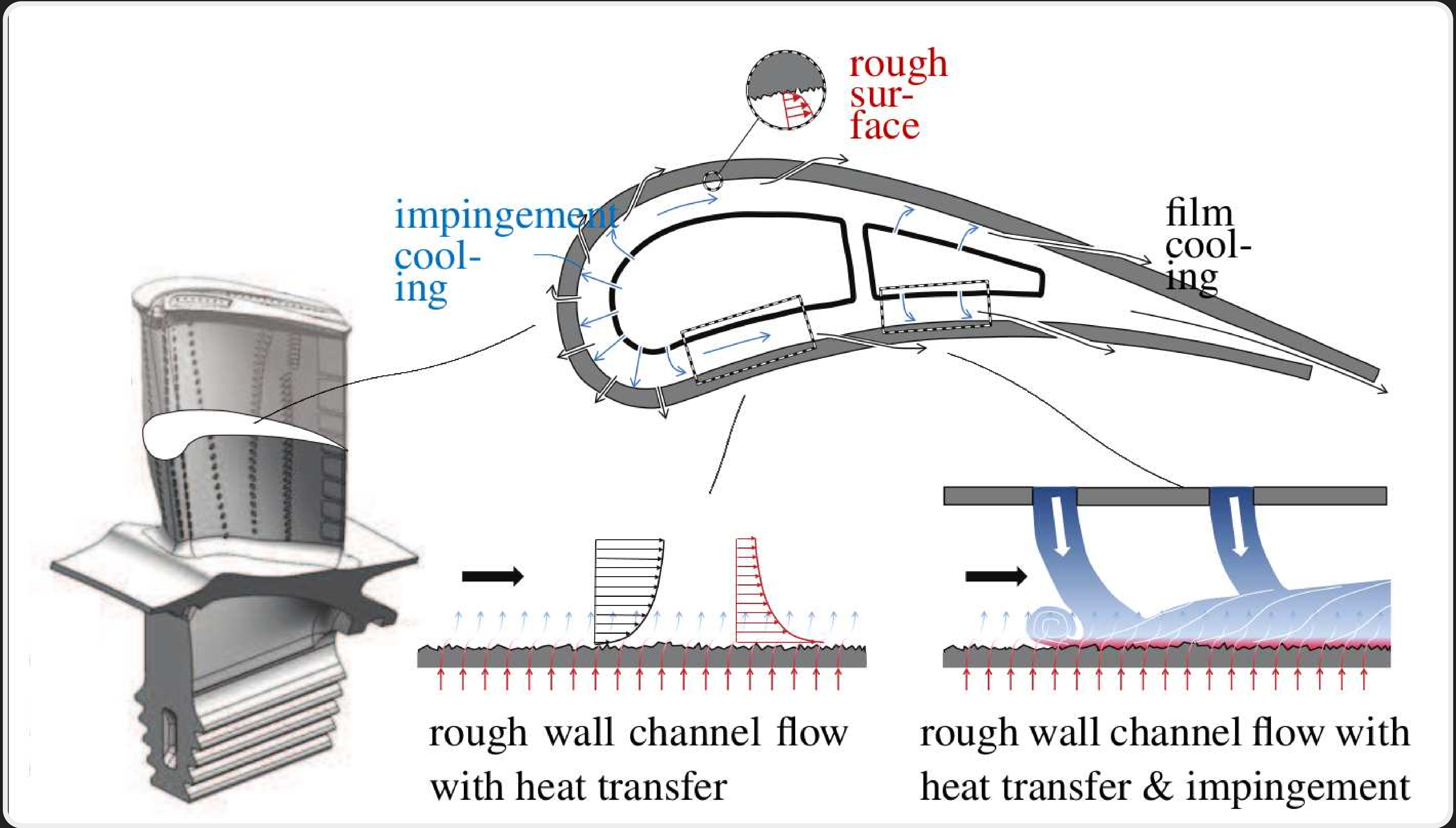


- 20 million points
- We can visually perceive the lower resolution
- Turbulence model is running off design
- This is still finer than a typical LES in the literature
- Results are still reliable

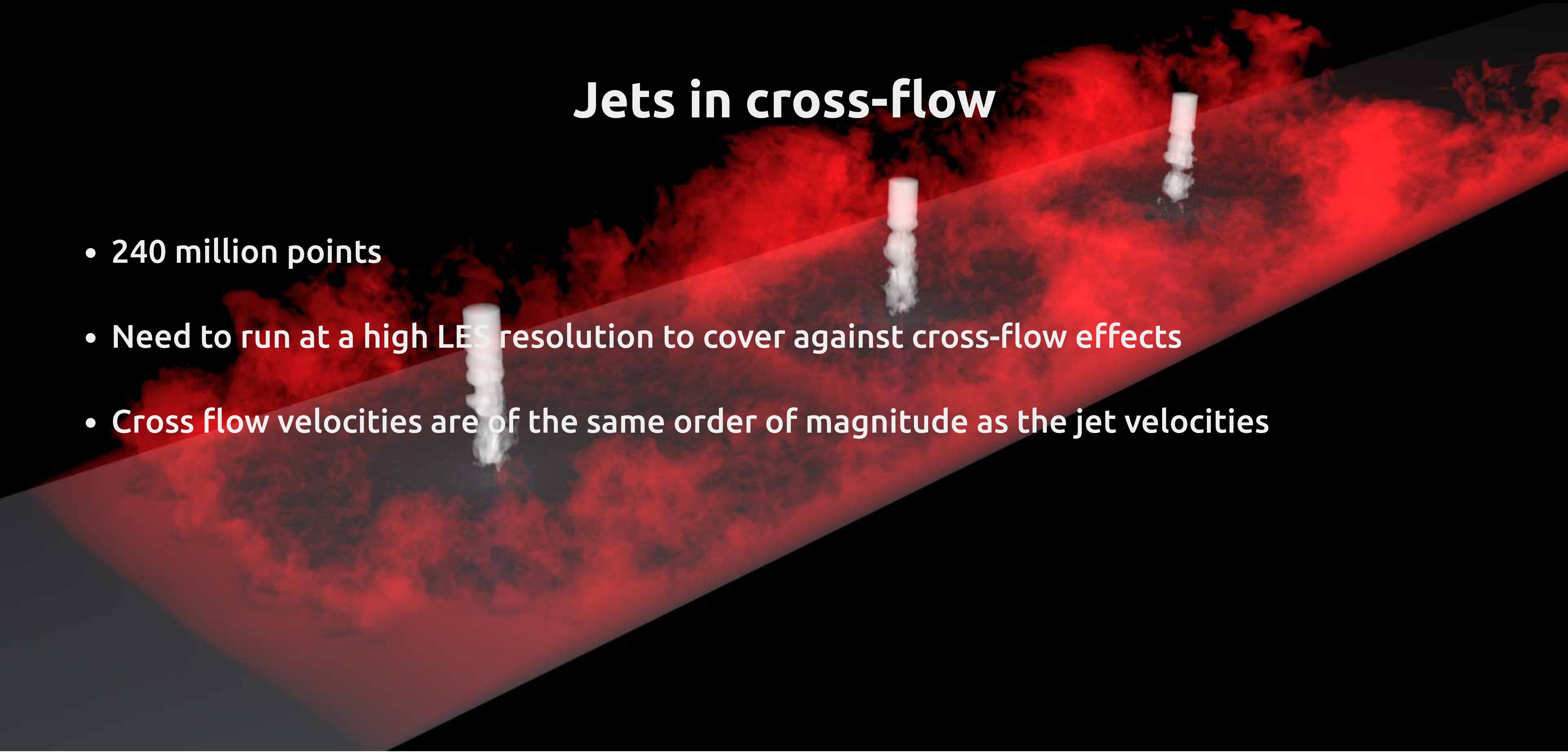
Need to make the setup a bit more realistic

- We've looked at single and multiple impinging jet setups
- But those cases are still not realistic enough

Jets in cross-flow



Jets in cross-flow



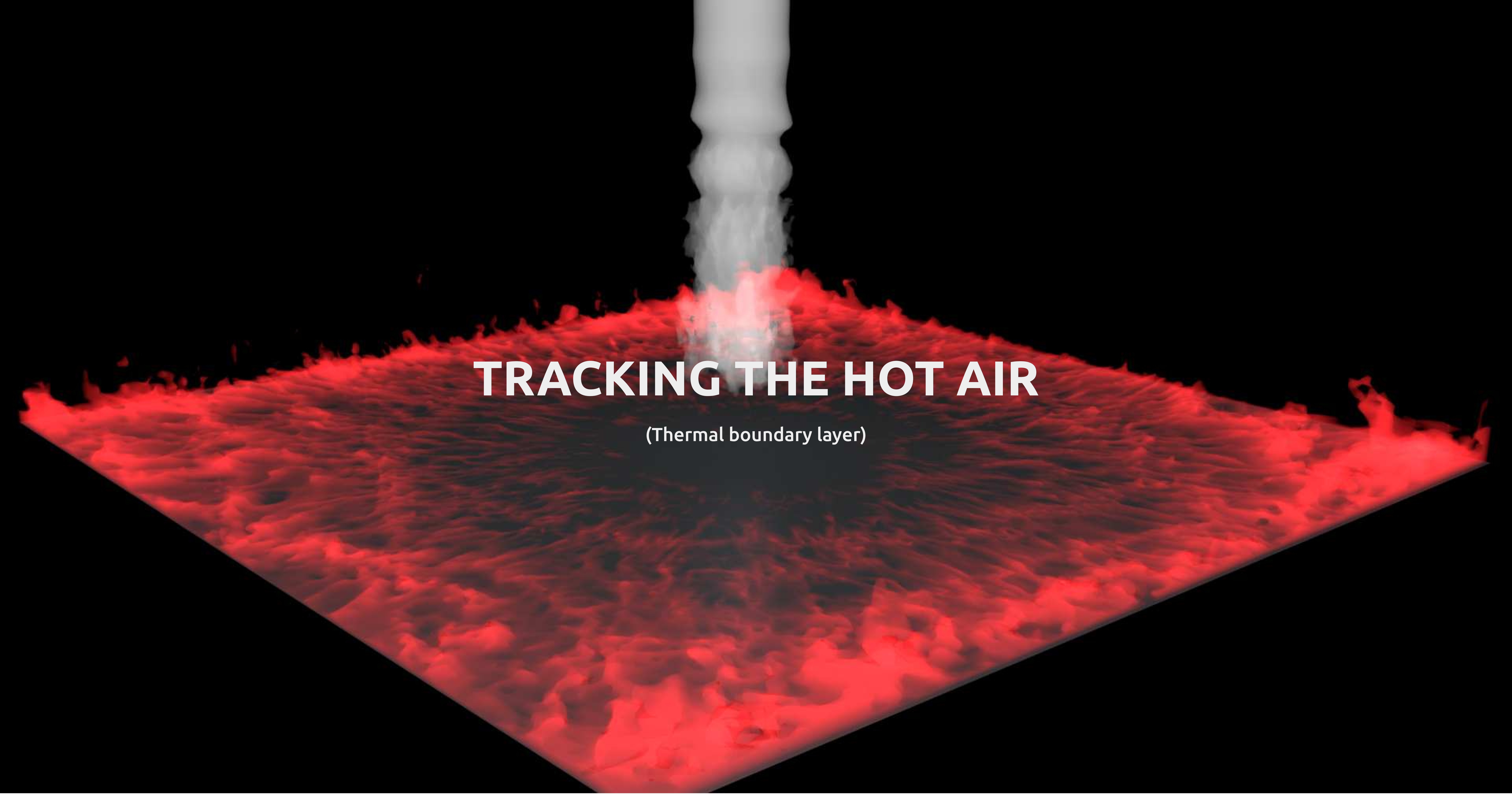
- 240 million points
- Need to run at a high LES resolution to cover against cross-flow effects
- Cross flow velocities are of the same order of magnitude as the jet velocities

What do we do with all the data?

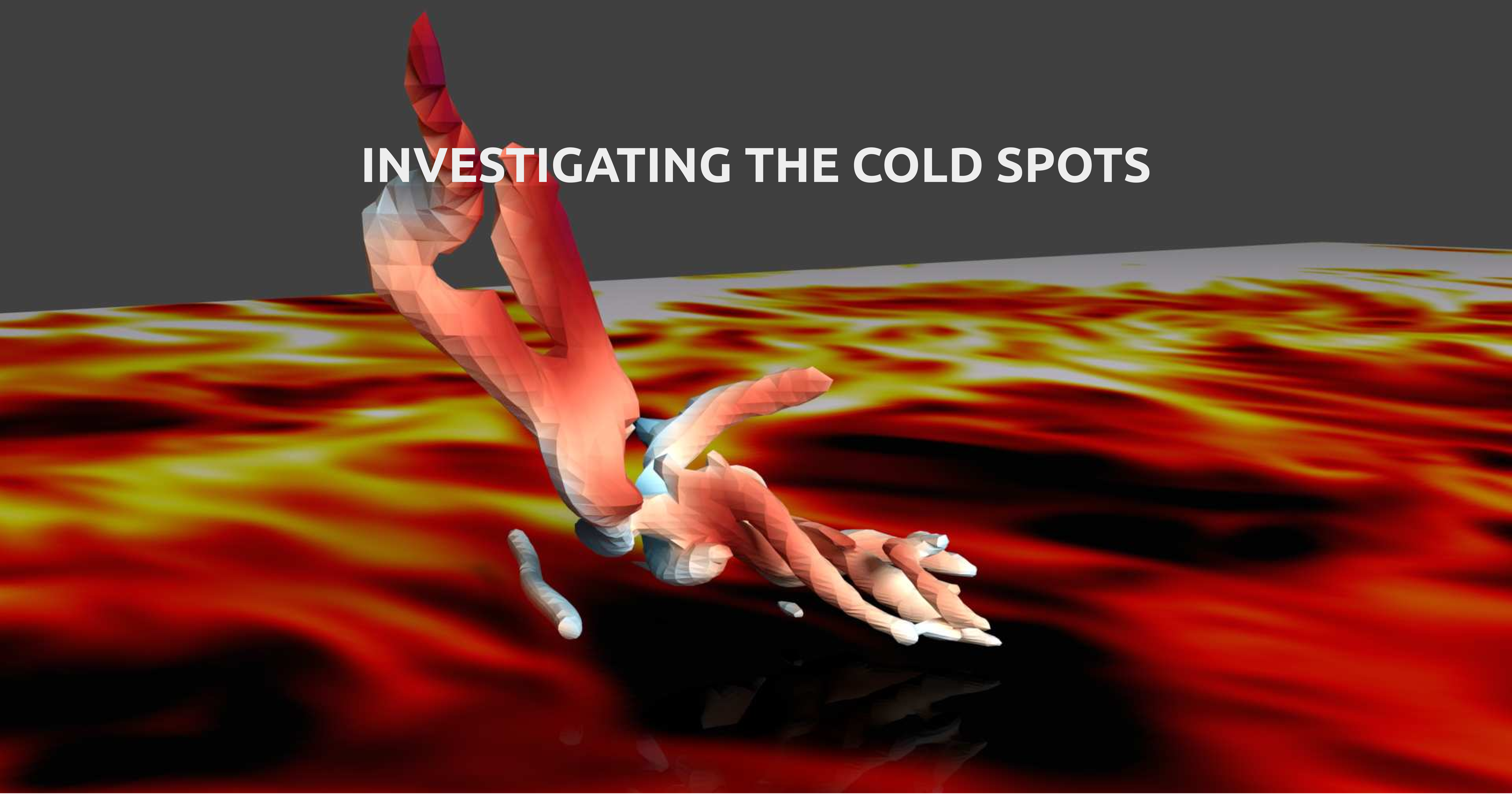
- For industrial applications, the main interest is on the time-averaged quantities
- The data produced through DNS is also an excellent database for fundamental research purposes ...
- and also to make very nice pictures

TRACKING THE HOT AIR

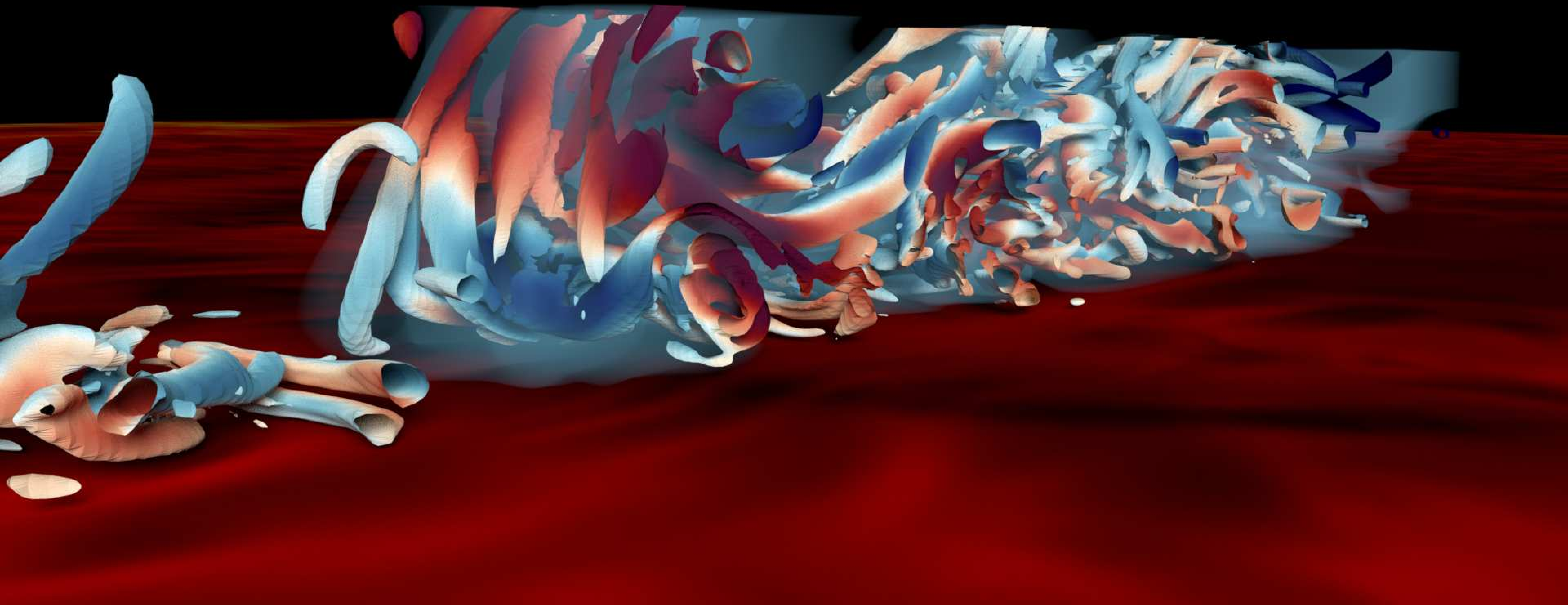
(Thermal boundary layer)



INVESTIGATING THE COLD SPOTS


INVESTIGATING THE COLD SPOTS

VISUALISING THE COLD FLOW



WHERE NEXT

MACHINE LEARNING

Turbulence modelling

- Use our in-house machine learning tool to improve current turbulence and heat-transfer models
  - Evolutionary algorithm for symbolic regression (GEP)
- This applies to both LES and RANS methods

MACHINE LEARNING

Pattern recognition

- Use NNs to classify cold and hot spots based on their physics
- Easier post-processing
- Could also be applied in scenarios where not all the necessary data is available (experiments)


ANY QUESTIONS?