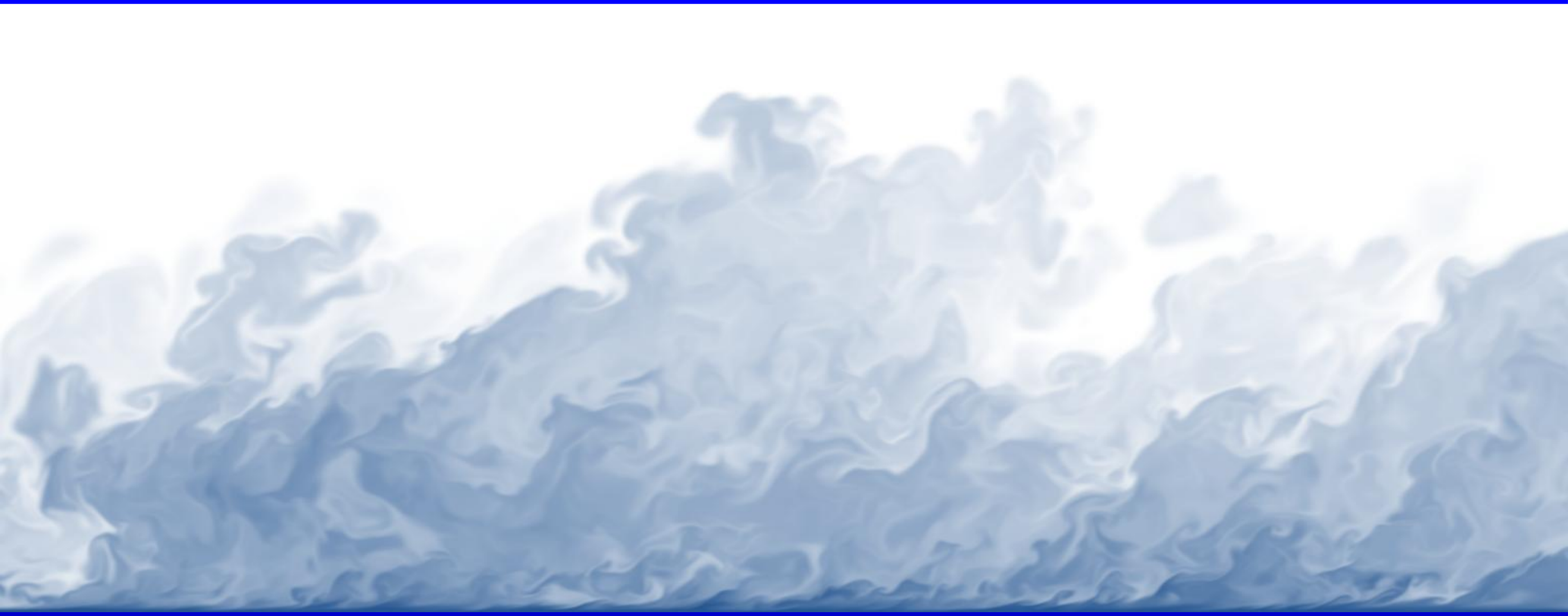
IAG
Institute of Aerodynamics
und Gas Dynamics



One Year of HUNTER from a User Perspective: Progress, Performance, and Patience with NS3D

Christoph Wenzel

WSSP 2026

April, 2026

Introduction – Boundary-layer group

Christoph Wenzel

Boundary Layers

Tobias Gibis

DNS Compressible Turbulence

Jason Appelbaum

DNS Compressible Turbulence

Abissan Suntharajan

DNS Compressible Turbulence

Leander Düntsch

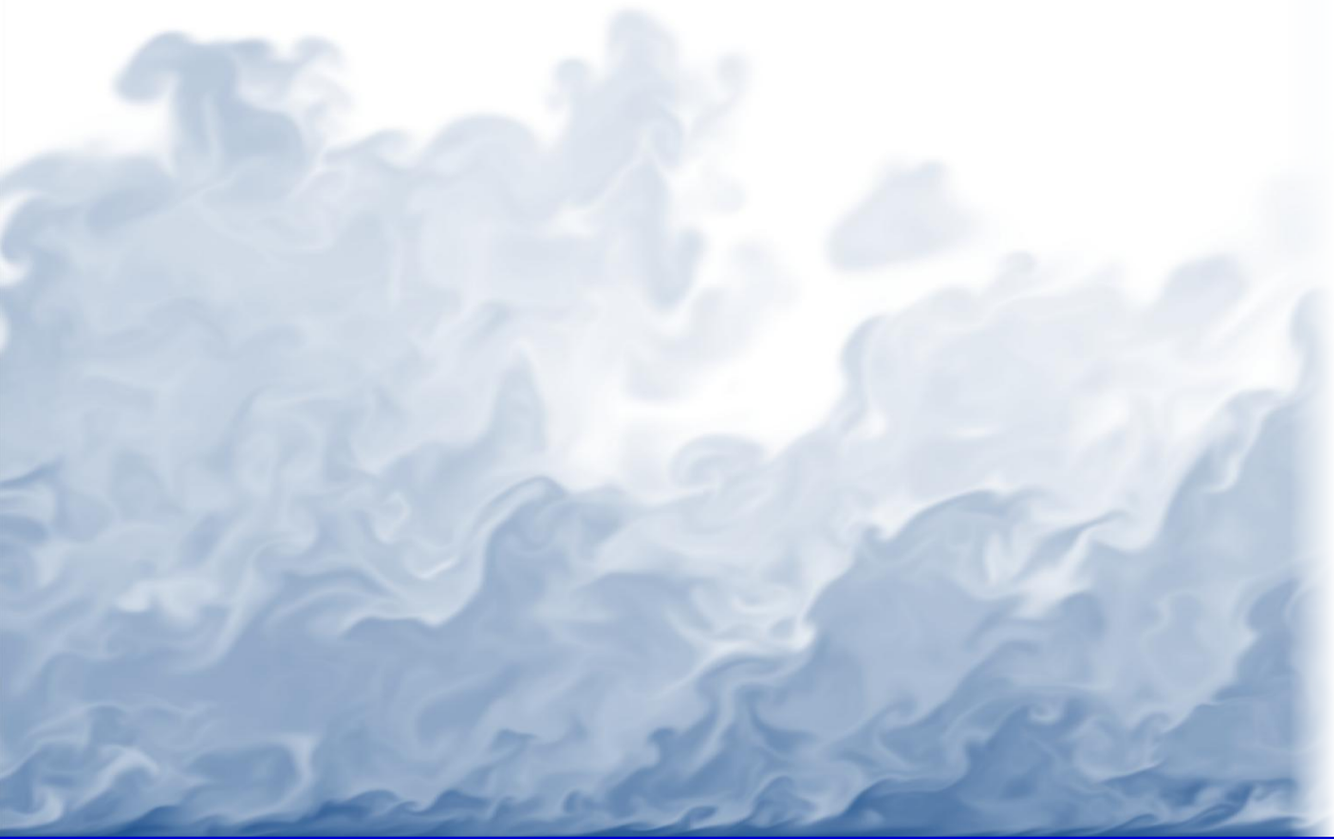
DNS Transition Control

Tristan Römer

EXPERIMENT Transition Control

Institute of Aerodynamics and Gas Dynamics
University of Stuttgart
70569 Stuttgart
Germany

IAG
Institute of Aerodynamics
und Gas Dynamics



Introduction

NS3D GPU Porting Status

Performance insights

- Performance sensitivity to
  - MPI decomposition
  - Rank ordering
- Scaling behaviour

Summary/Conclusions

IAG
Institute of Aerodynamics
und Gas Dynamics

Rendering: Wenzel

Christoph Wenzel

Paradigm

In a certain sense, understanding boundary layers can be assessed by the ability to predict their behavior for arbitrary boundary conditions.

Task:

Solve the governing equations:

Conservation of mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j\right) = 0$$

Conservation of momentum:

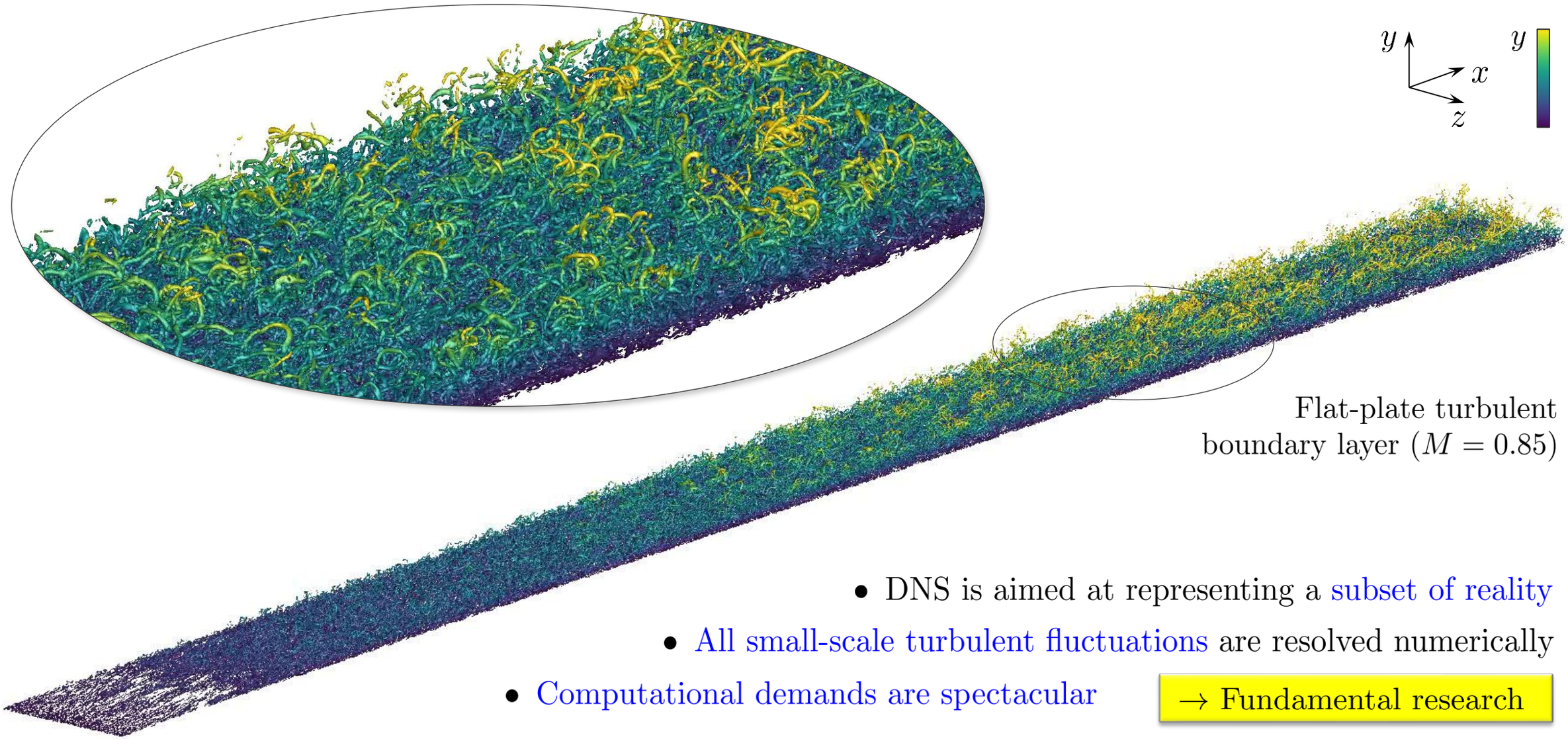
$$\frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_j}\left(u_i \rho u_j\right) = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j}$$

Conservation of total inner energy:

$$\frac{\partial}{\partial t}\left(\rho E\right) + \frac{\partial}{\partial x_j}\left(u_j \rho H\right) = \frac{\partial}{\partial x_j}\left(u_i \sigma_{ij}\right) + \frac{\partial q_j}{\partial x_j}$$

with: $\sigma_{ij} \approx 2\mu\left(S_{ij} - \frac{1}{3} S_{kk}\delta_{ij}\right)$ $S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ and: $q_j = -k\frac{\partial T}{\partial x_j}$

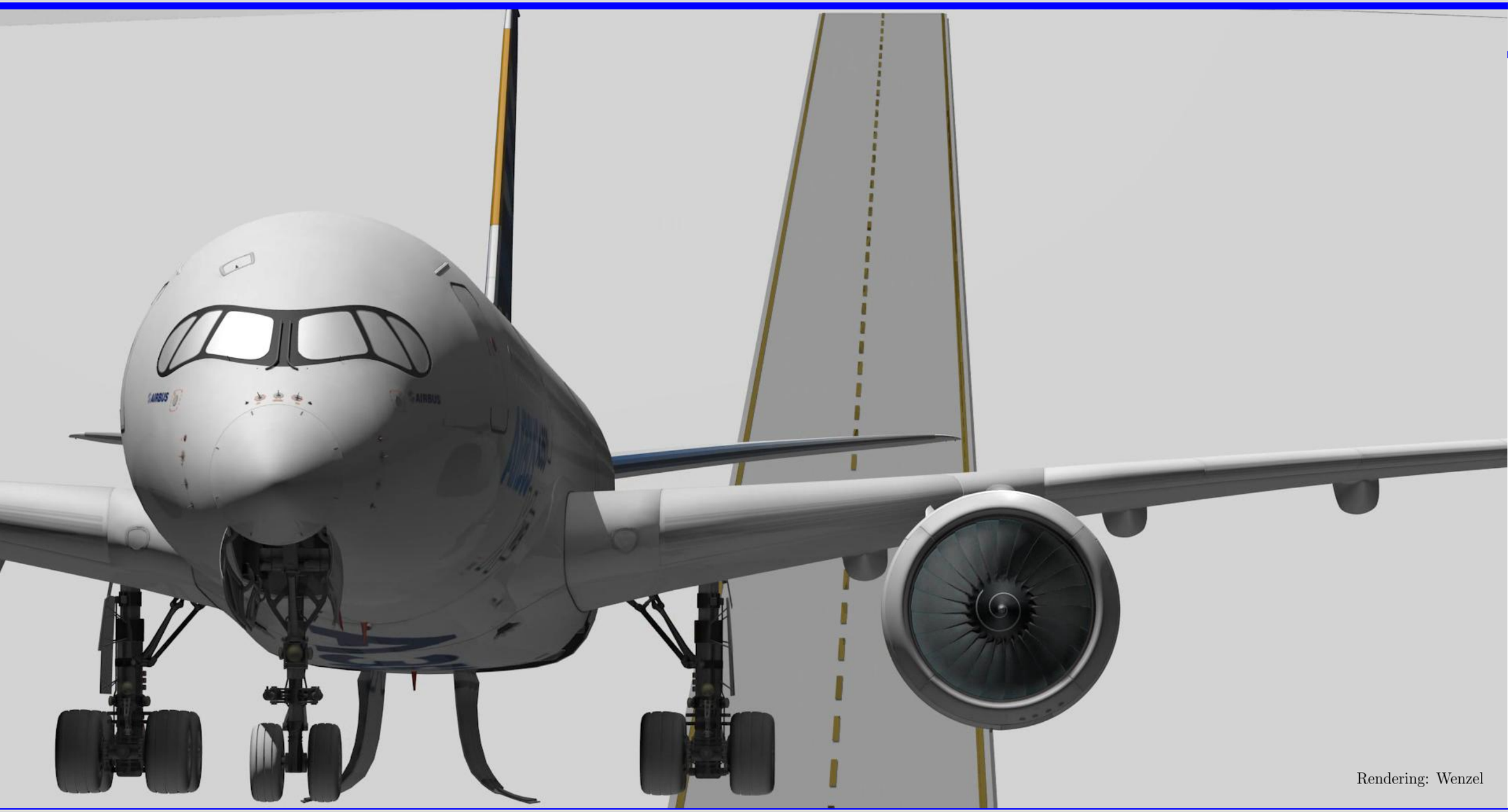
IAG
Institute of Aerodynamics
und Gas Dynamics



Flat-plate turbulent
boundary layer ($M = 0.85$)

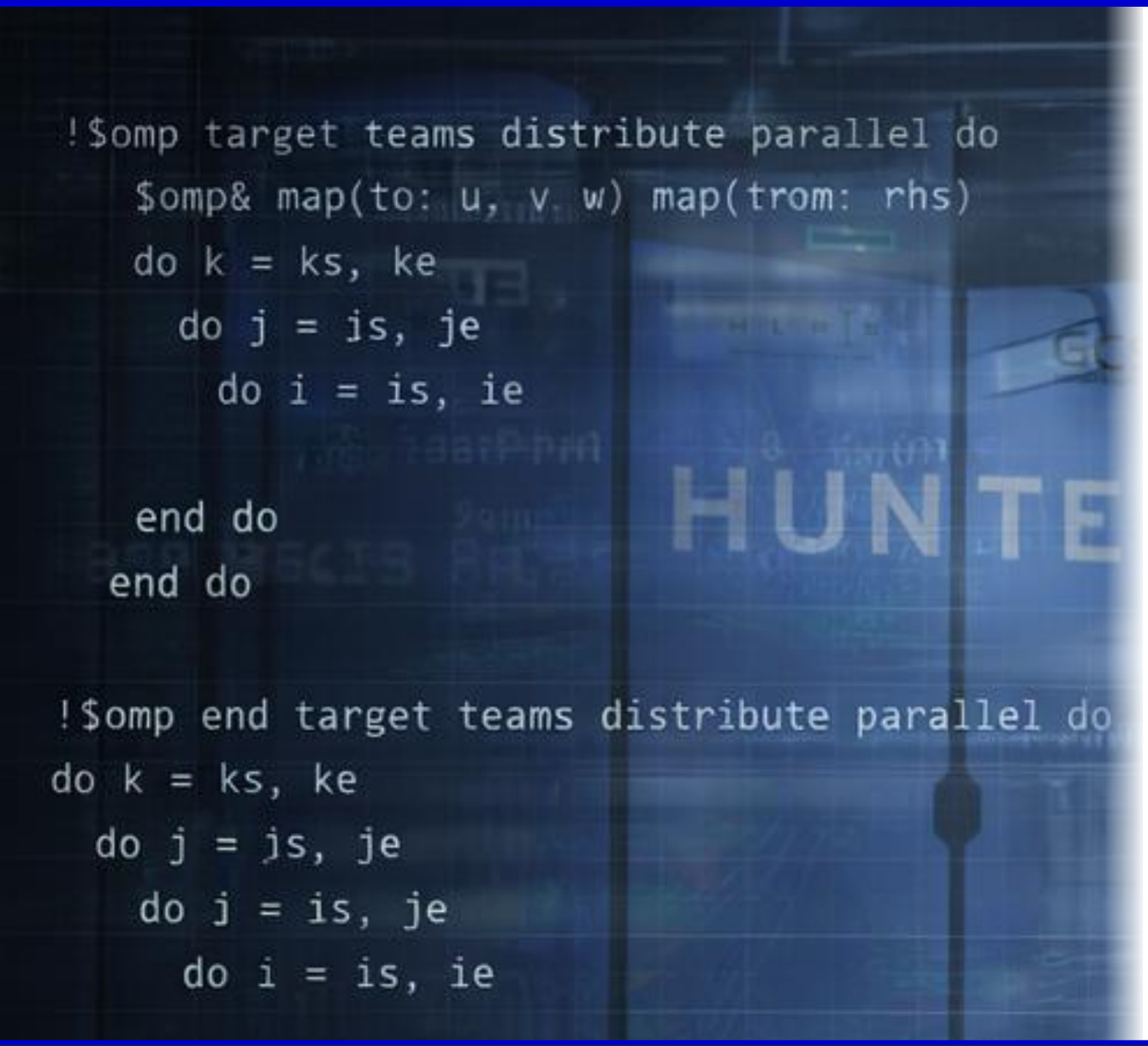
- DNS is aimed at representing a subset of reality
- All small-scale turbulent fluctuations are resolved numerically
- Computational demands are spectacular

→ Fundamental research



Rendering: Wenzel

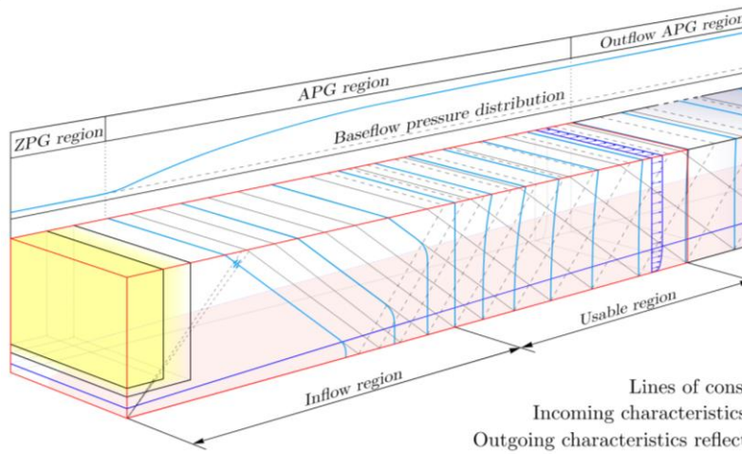
IAG
Institute of Aerodynamics
und Gas Dynamics



NS3D: Basic information

- NS3D is a compressible Navier–Stokes finite-difference solver for fundamental research
- First version developed between 2005–2008
- Current version stems from a major rewrite and refactoring (2017–2018)

Language	Modern Fortran
Domain Decomposition	Full 3D domain decomposition
Parallelization (CPU)	Hybrid: MPI + OpenMP
GPU Porting	staying with OpenMP



NS3D: Porting to GPU

- **Setup and early decisions (2024)**
  - GPU port of NS3D using OpenMP, with support from HLRS, HPE, and AMD
- **September 2024**
  - Start of porting effort using partially ported version for Hawk AI (NVIDIA A100) by HPE
  - Major issues: differences between NVHPC, Cray, and AMD compilers
  - AMD ecosystem (Cray, AMD) less mature
  - Many compiler bugs and missing features significantly slowed development
- **March 2025**
  - First correct Taylor–Green vortex with all numerical schemes
- **June 2025**
  - First correct turbulent boundary layer case
- **August 2025**
  - First stable GPU version for production-case development
- **September 2025: Status at WSSP**
  - Some rare issues remain
  - Workarounds still present in parts of the code

2 days per week each

IAG
Institute of Aerodynamics und Gas Dynamics

- **Status September 2025**
  - Production cases are running stably (small and large scale)

?

2 days per week each

- Many setbacks e.g. due to software-stack updates exposing previously hidden bugs in our code (mainly affecting large cases), along with limited MPI decomposition robustness
- Extensive debugging effort, largely relying on trial-and-error (and a fair amount of guesswork)
- Mystery bugs: random NaNs or Infs in major arrays, HDF5 interface randomly fails

- **Today (April 2026)**
  - Production cases are running stably (small and large scale)

A small collection of problems

- Derived types (e.g. `Pvec%u`) not reliably usable inside kernels
  → Solution: Use pointer-based access before entering kernels (e.g. `u` → `Pvec%u`)
- Cray compiler has a limited number of arrays per kernel (≈ 120 arrays)
  → Solution: Split very large kernels
- NaNs/Infs are not reliably detected by the compiler
  → Solution: Custom GPU kernels to check for NaNs at regular intervals
- `firstprivate` scalars are not passed correctly to `declare target` functions
  → Solution: Use the `value` attribute
- Arrays slicing in kernel routines causes issues
  → Solution: Avoid slicing inside kernels
  → Note: Rewrite critical parts (e.g. reduced I/O) using explicit loops
- ...

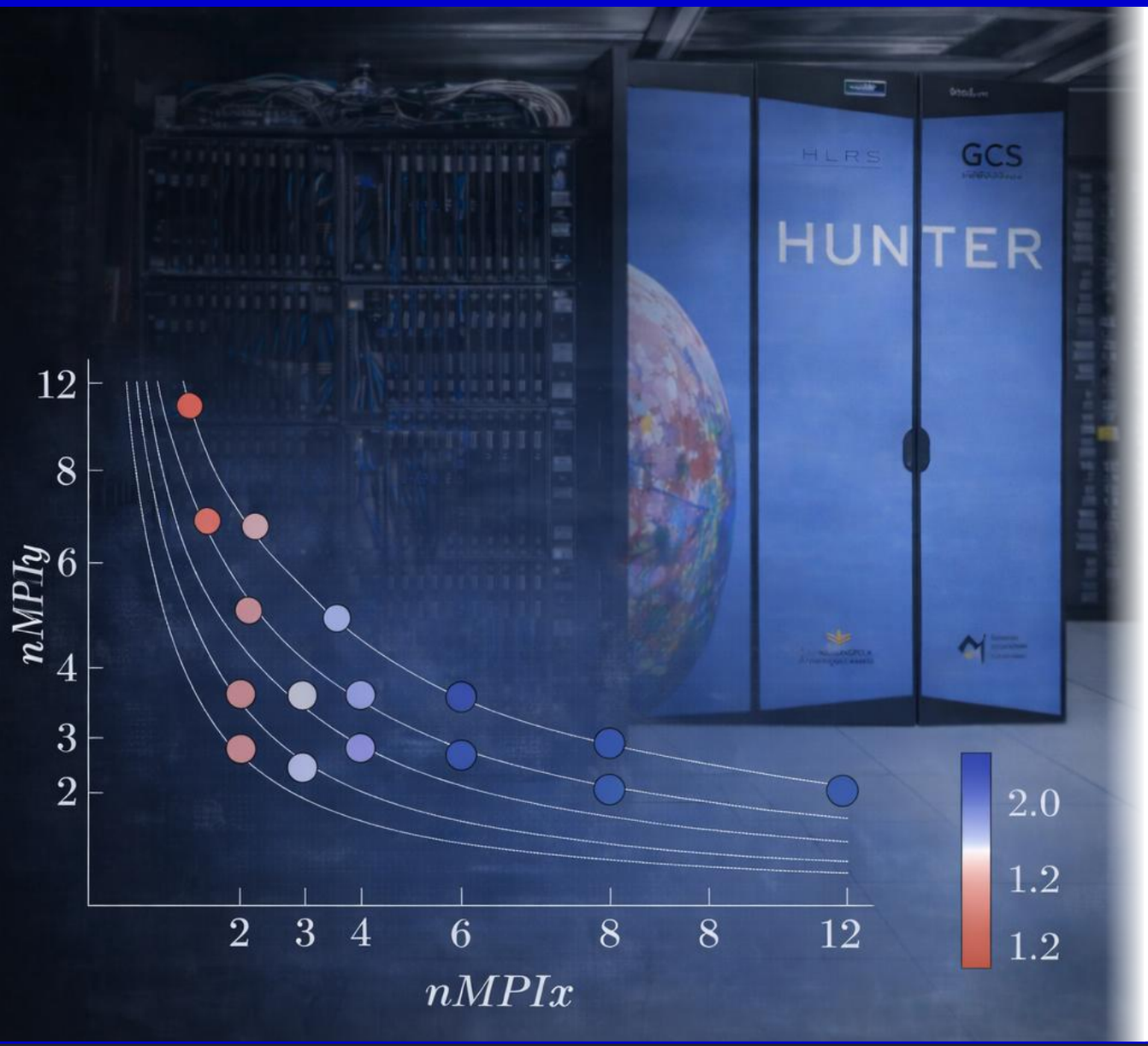
NS3D modernization for heterogeneous GPU systems

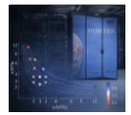
IAG
Institute of Aerodynamics
und Gas Dynamics



- Ported NS3D to multiple GPU platforms (LUMI, Leonardo)
  - Extended support beyond USM-based architectures
  - Introduced a new memory management strategy
  - Addressed compiler differences in OpenMP offloading
  - Enabled modern Fortran data structures on GPUs
  - Improved code stability through systematic debugging and runtime checks
  - Started performance optimization focusing on the main bottlenecks

- **Current status:** Stable production runs established; final NS3D_{neo}.develop version expected soon
- NS3D_{neo}.develop runs in both USM and non-USM modes





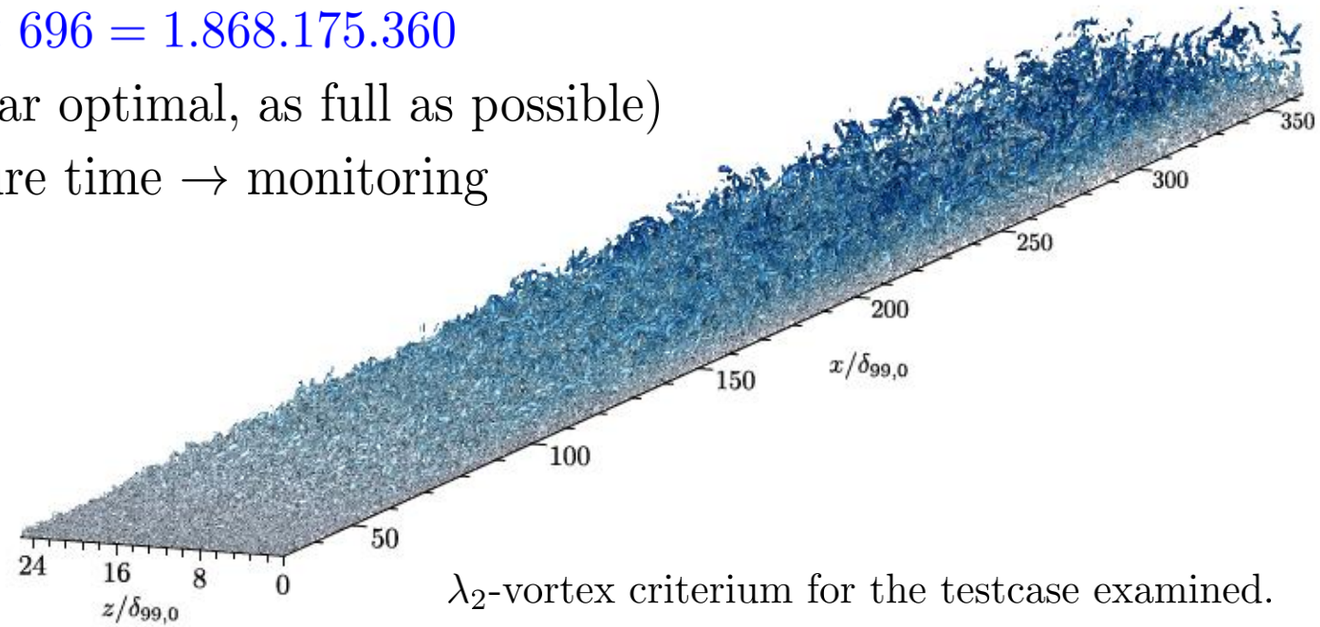
Characterization of NS3D$_{neo}$.develop on HUNTER

IAG
Institute of Aerodynamics
und Gas Dynamics

- DNS of a spatially developing turbulent boundary layer
- Grid stretching in streamwise and wall-normal direction (requiring grid transformations)
- Turbulent inflow condition (random-number-based)
- Spatial filtering is applied

Domain size has been chosen to fit on 12 nodes (48 MPI ranks, 1 MPI rank per APU):

- Total number of grid points: $5592 \times 480 \times 696 = 1.868.175.360$
- Grid points per MPI rank: 38.920.320 (near optimal, as full as possible)
- Procedure: Run 1000 time steps → measure time → monitoring
- Postprocessing

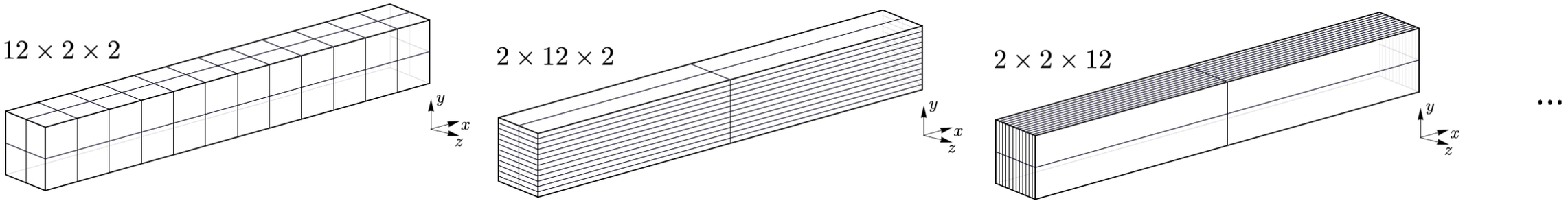


λ_2-vortex criterium for the testcase examined.

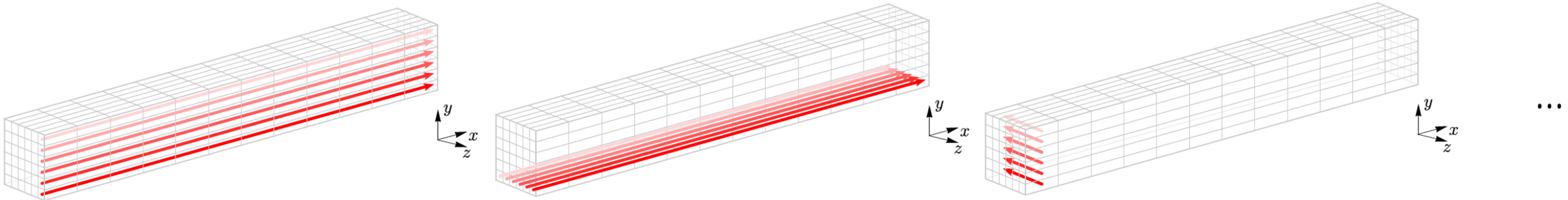
Performance Sensitivity: MPI Decomposition and Rank Ordering

IAG
Institute of Aerodynamics
und Gas Dynamics

All permutations of the 3D MPI domain decomposition were tested ($12 \times 2 \times 2$, $2 \times 12 \times 2$, ...), with a fixed total of 48 MPI ranks:



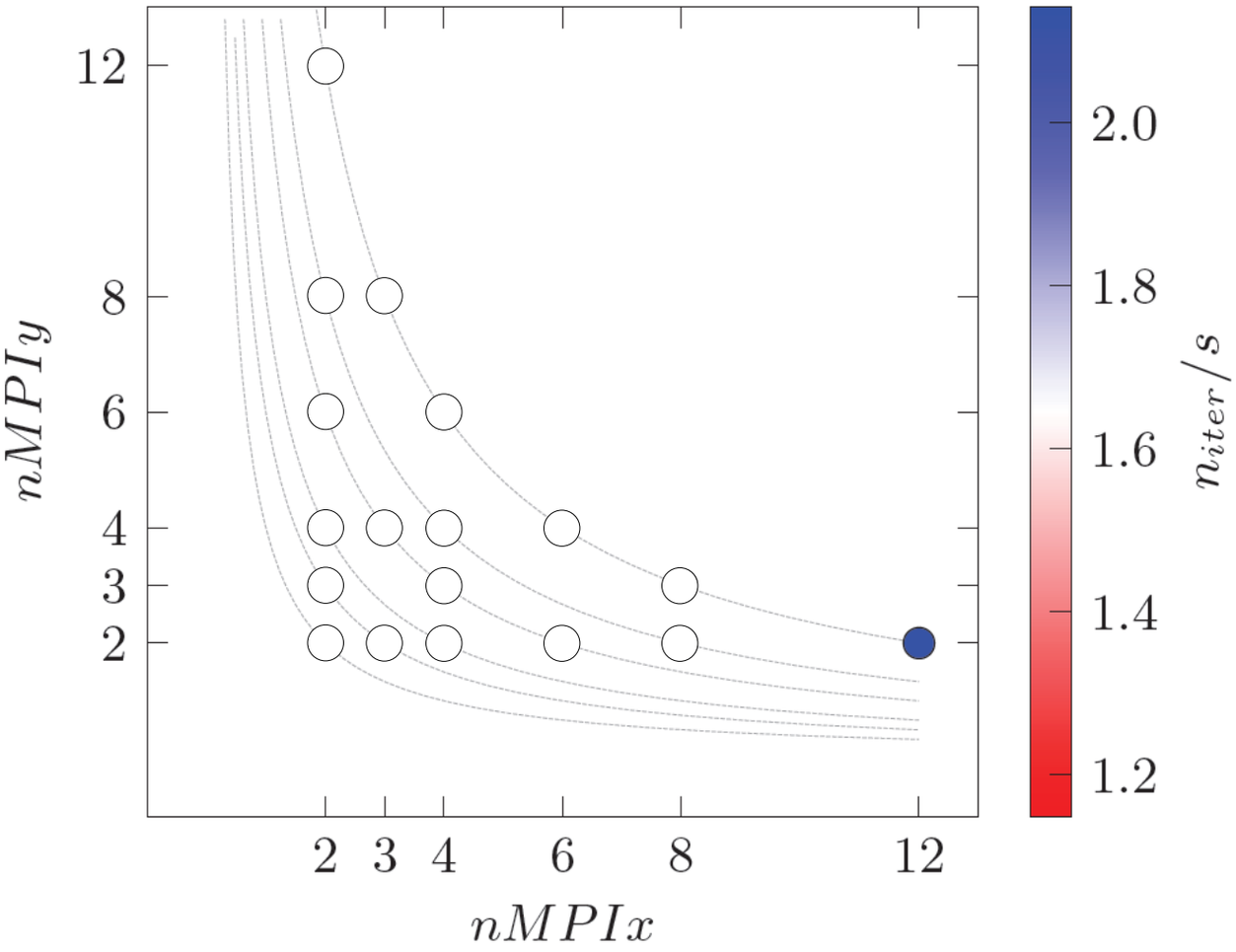
All permutations of the 3D MPI rank ordering were tested (`xyz`, `xzy`, ...):

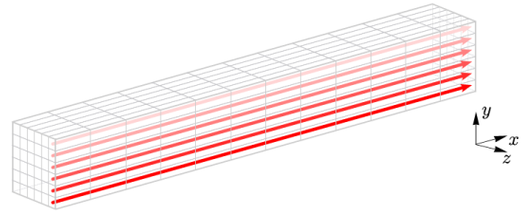


Total: 108 cases

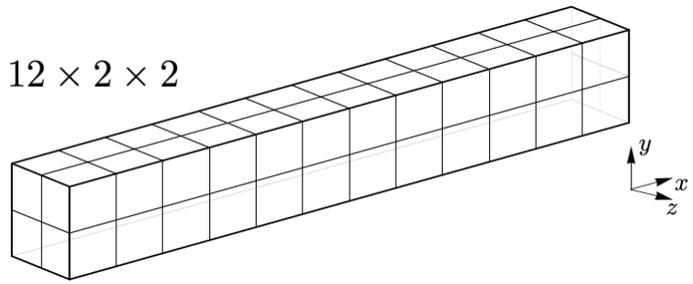
MPI decomposition: $12 \times 2 \times 2$ (xyz-ordering)

Institute of Aerodynamics
und Gas Dynamics



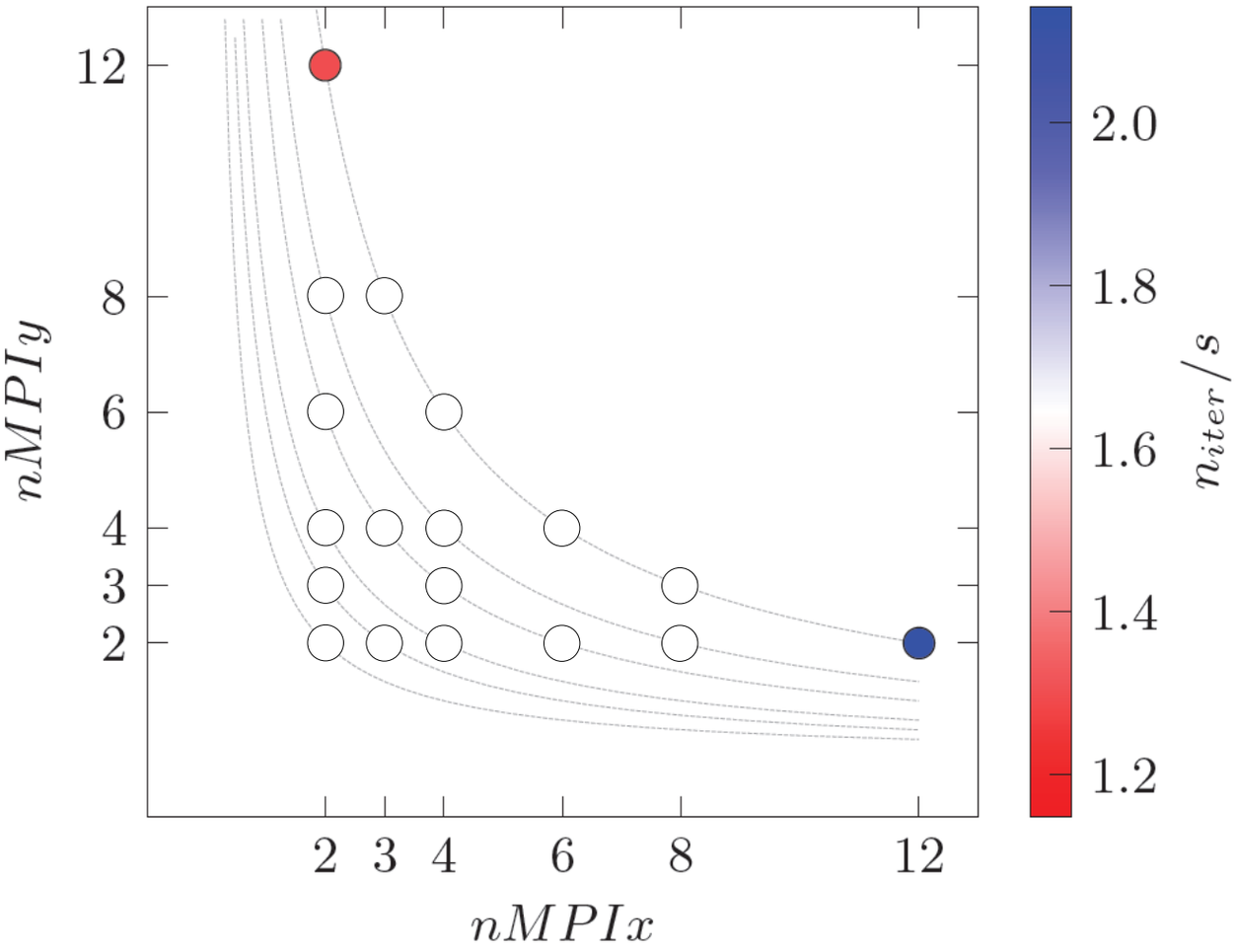


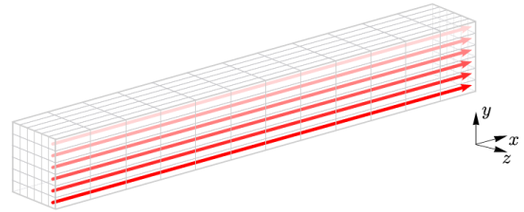

```
1  do k = 1, ...
2     do j = 1, ...
3        do i = 1, ...
4                 ...
5        end do
6     end do
7  end do
```



MPI decomposition: $2 \times 12 \times 2$ (xyz-ordering)

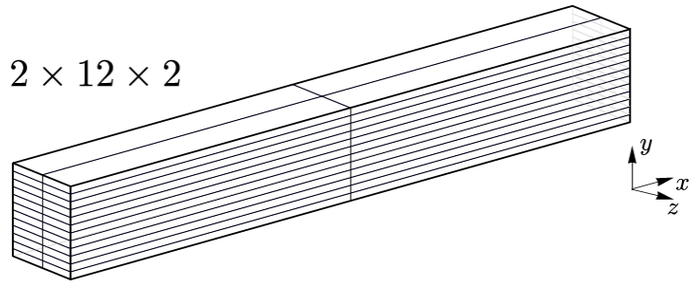
Institute of Aerodynamics
und Gas Dynamics





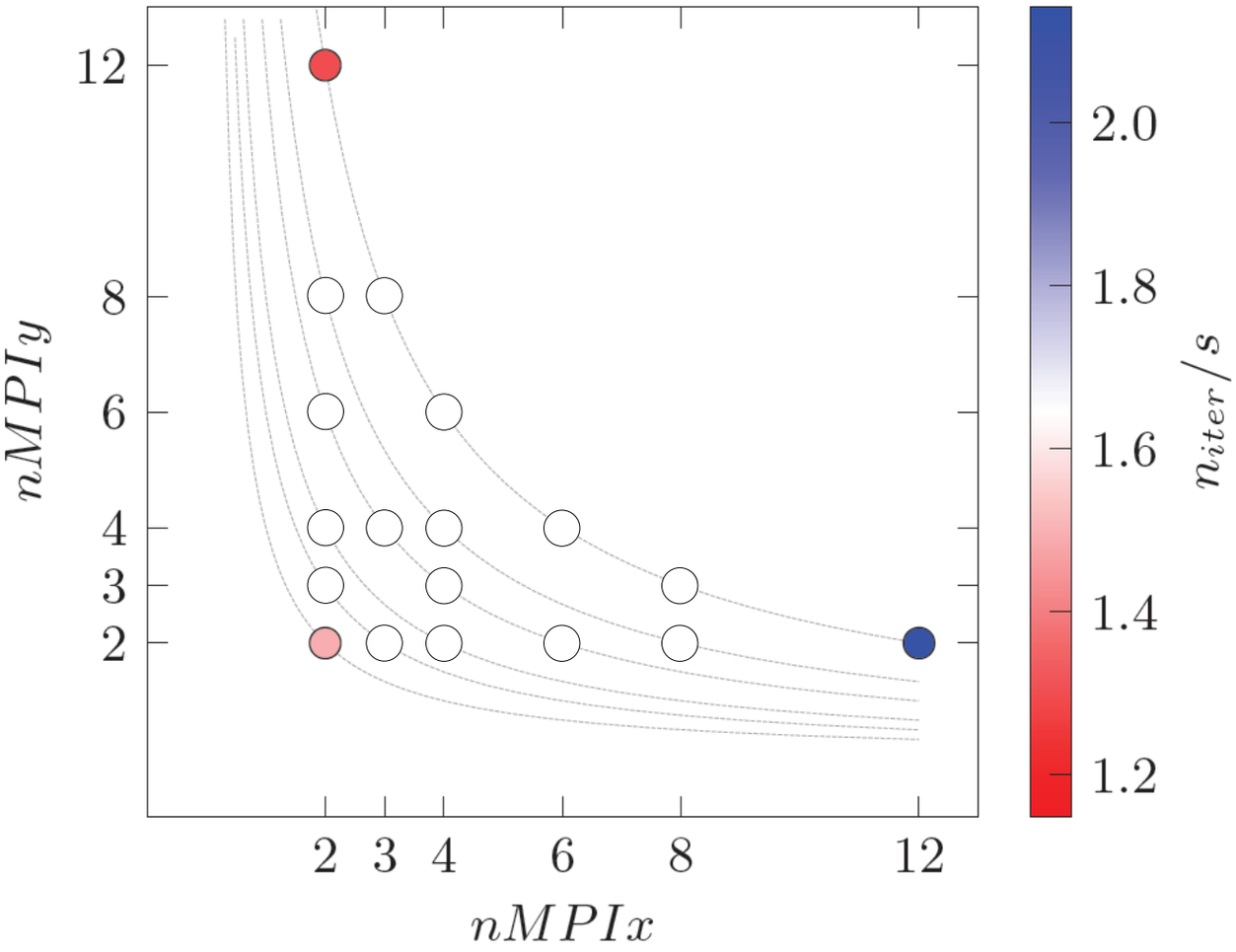

```
1  do k = 1, ...
2     do j = 1, ...
3        do i = 1, ...
4                 ...
5        end do
6     end do
7  end do
```

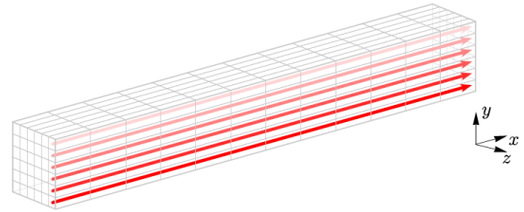
$2 \times 12 \times 2$



MPI decomposition: $2 \times 2 \times 12$ (xyz-ordering)

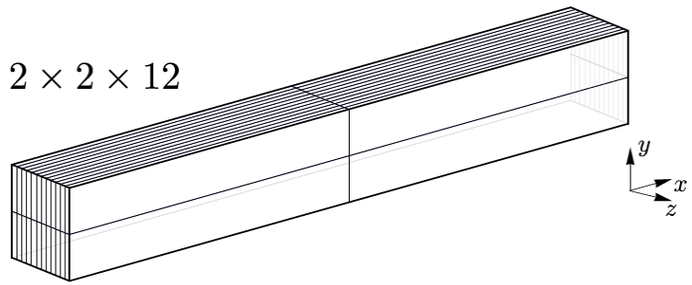
Institute of Aerodynamics
und Gas Dynamics





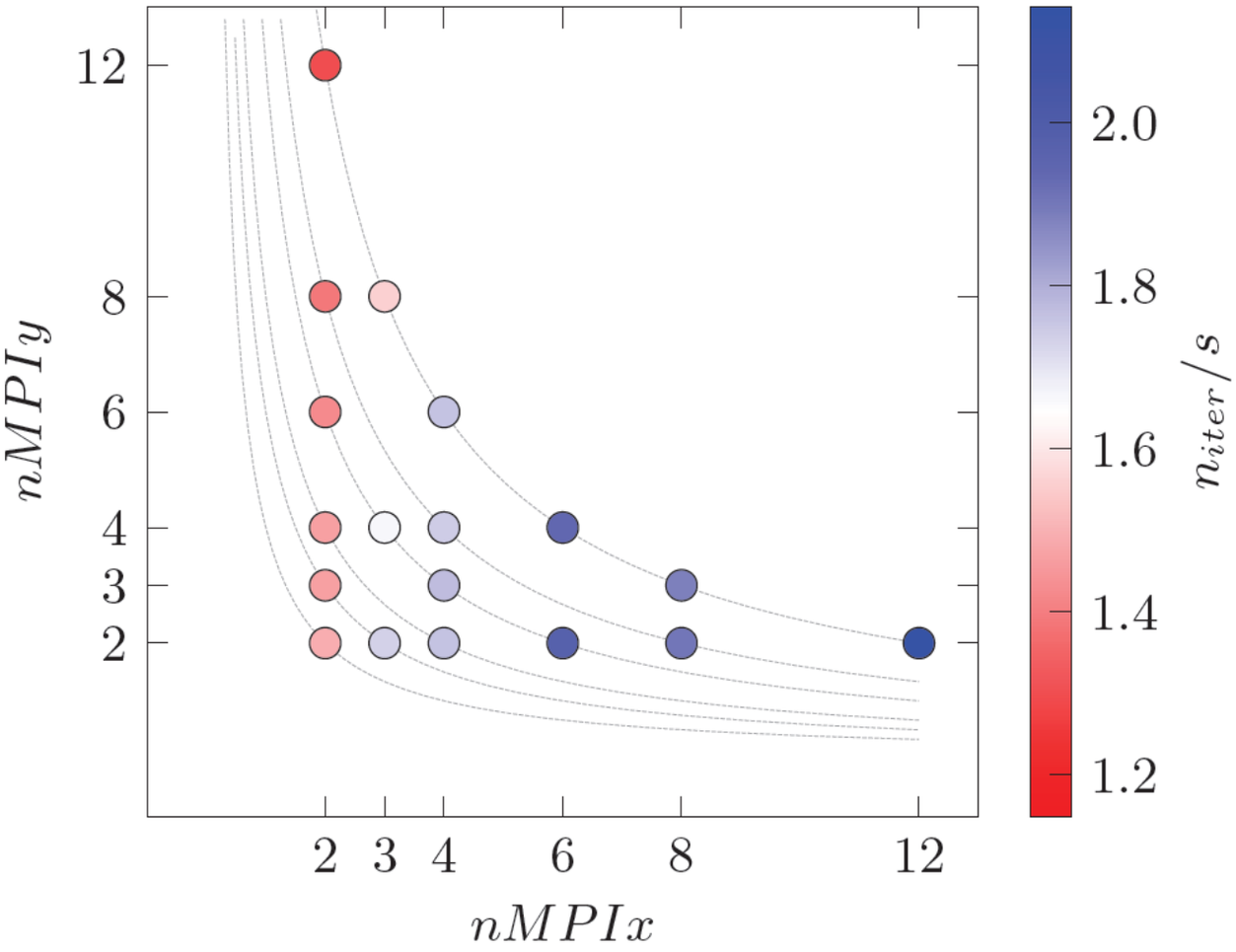

```
1  do k = 1, ...
2     do j = 1, ...
3        do i = 1, ...
4                 ...
5        end do
6     end do
7  end do
```

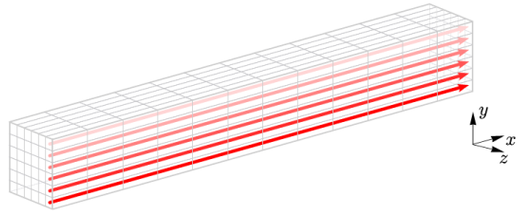
$2 \times 2 \times 12$



MPI decomposition (xyz-ordering)

Institute of Aerodynamics
und Gas Dynamics

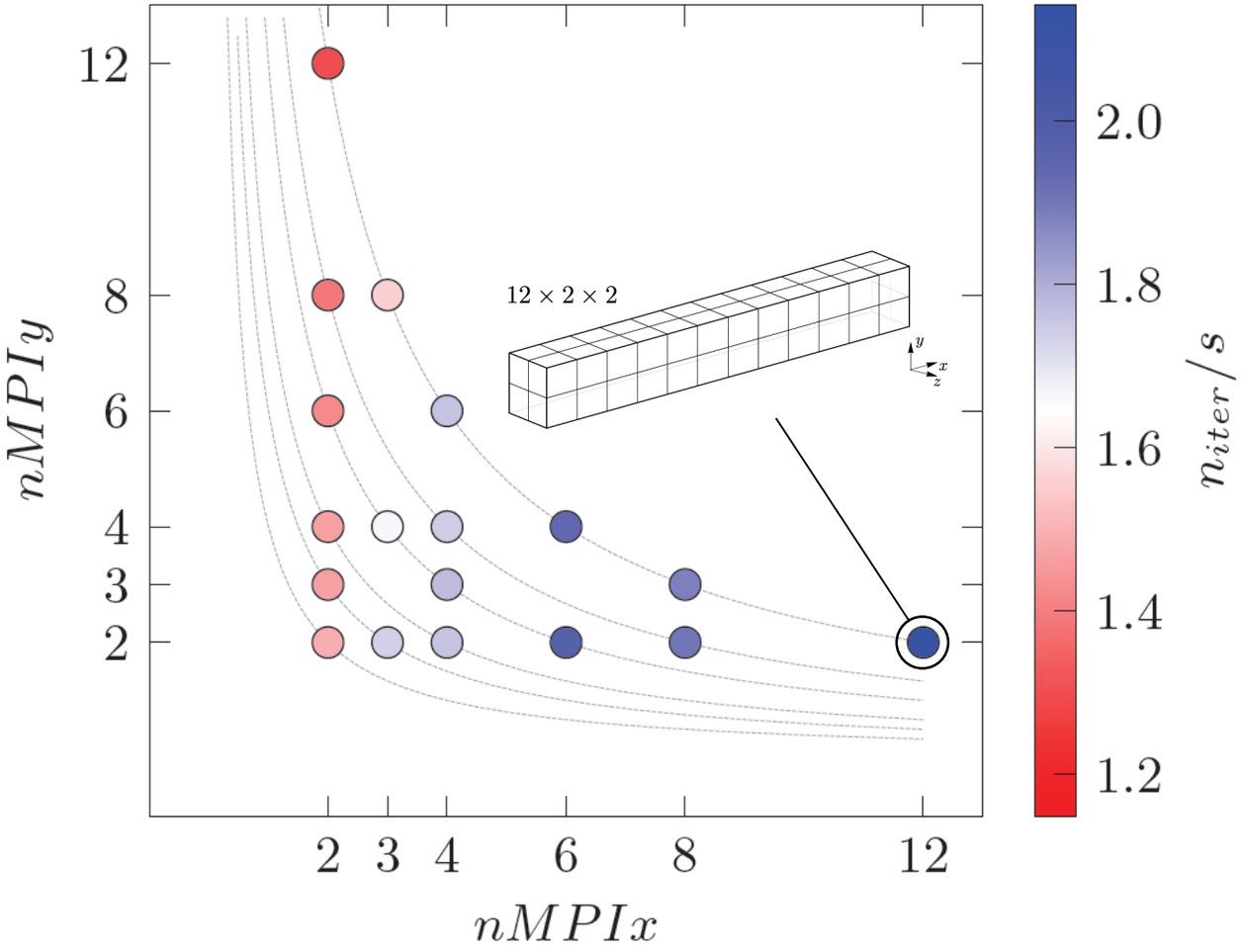


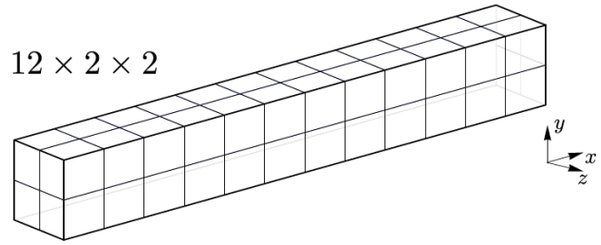


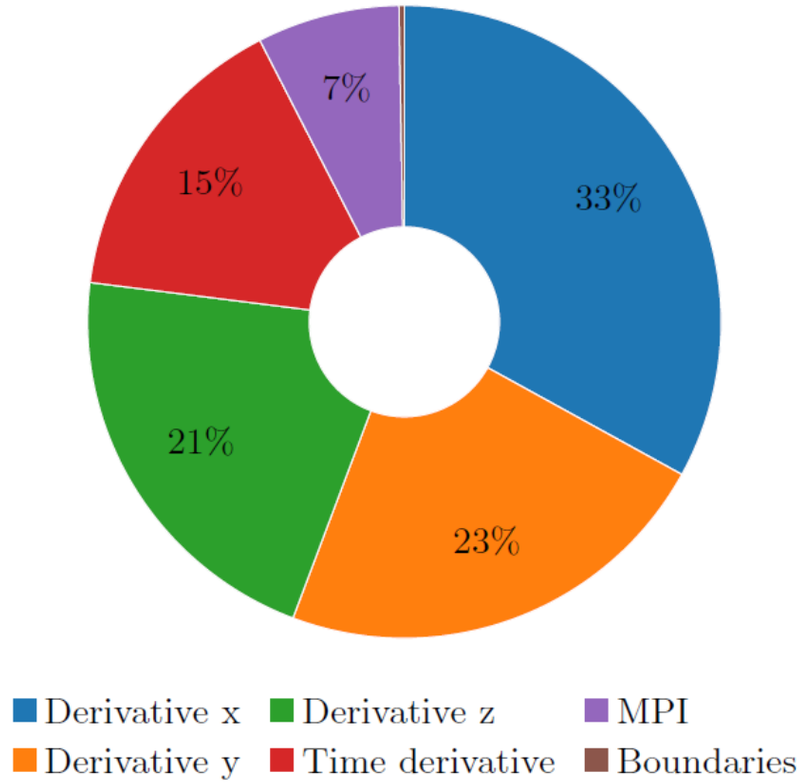

```
1  do k = 1, ...
2     do j = 1, ...
3        do i = 1, ...
4                 ...
5        end do
6     end do
7  end do
```

MPI decomposition: $12 \times 2 \times 2$ (xyz-ordering)

Institute of Aerodynamics
und Gas Dynamics

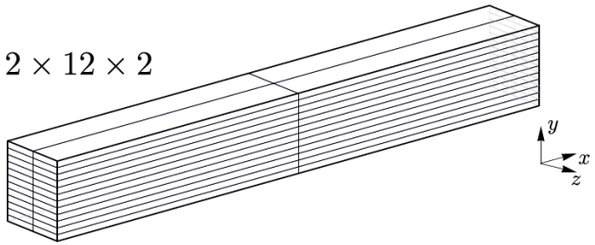


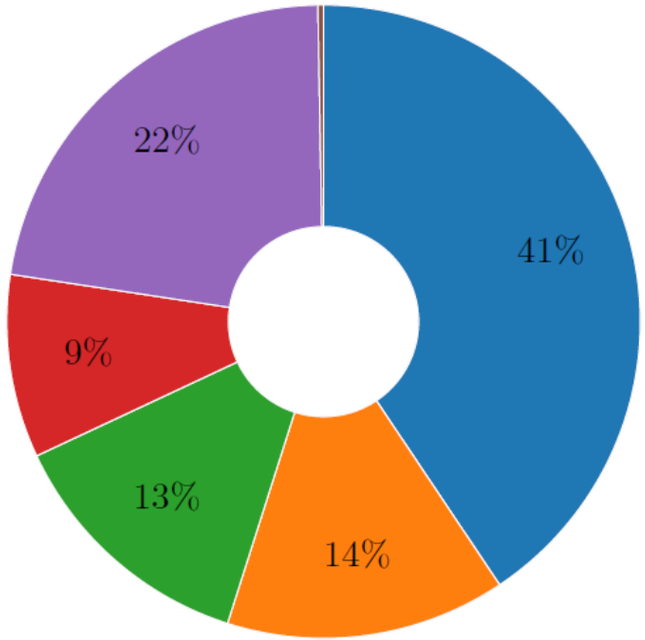


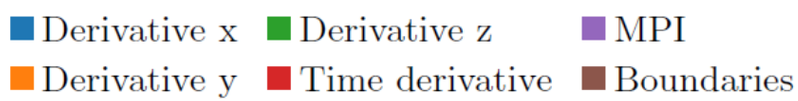


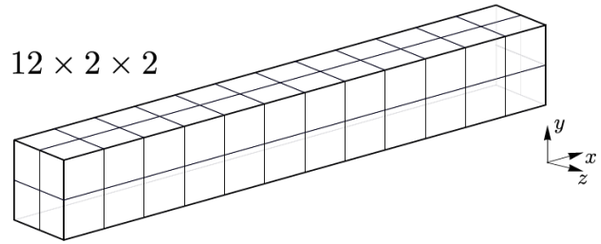
MPI decomposition: $12 \times 2 \times 2$ (xyz-ordering)

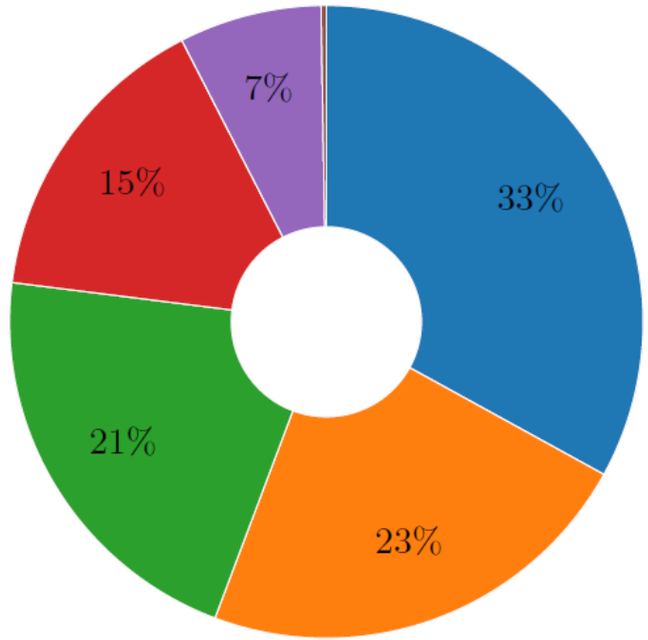
IAG
Institute of Aerodynamics
und Gas Dynamics

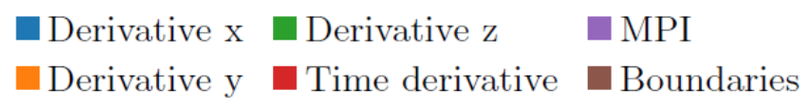




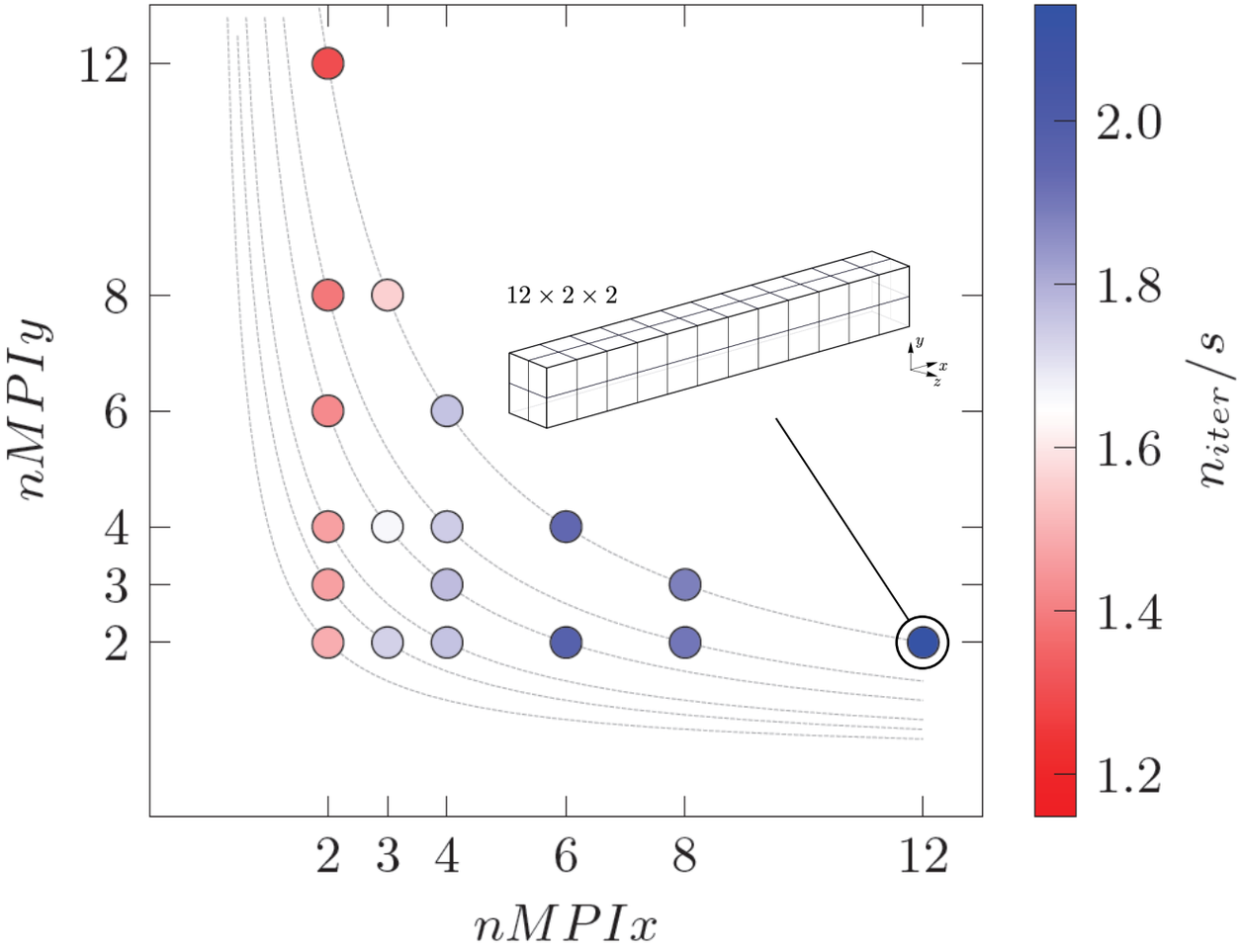


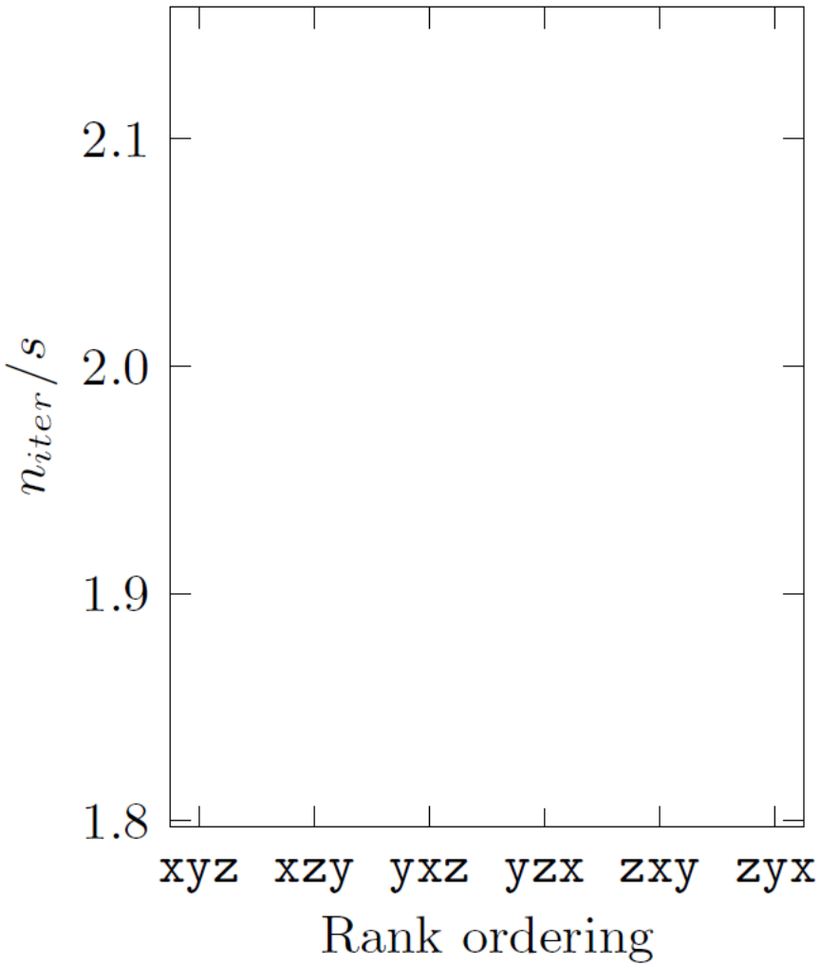






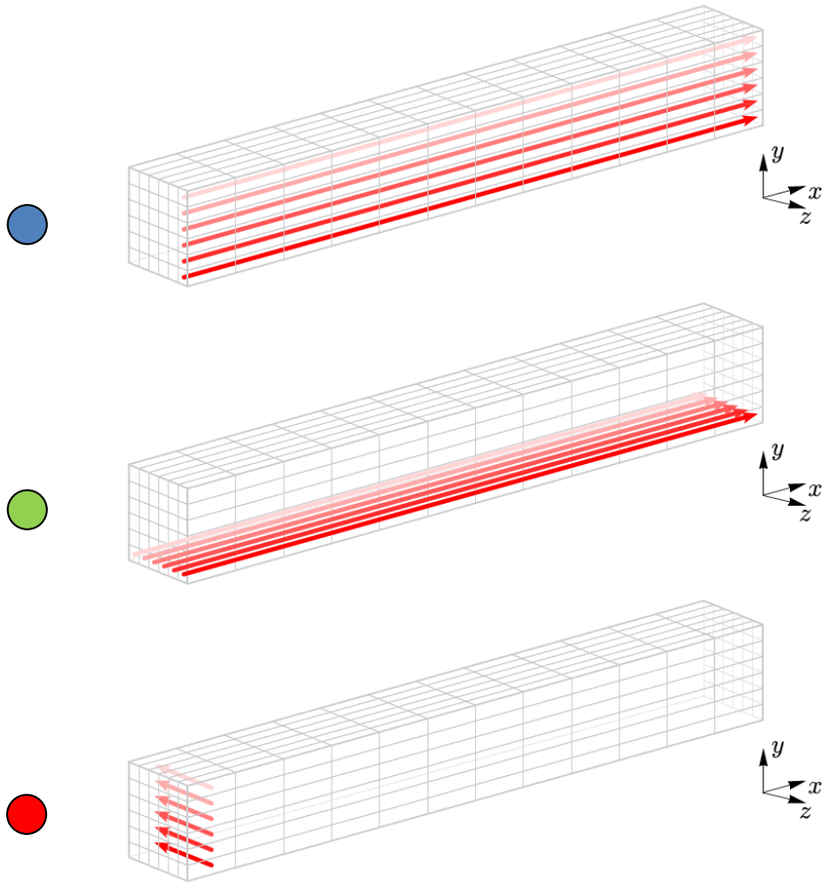
MPI decomposition: Influence of rank ordering

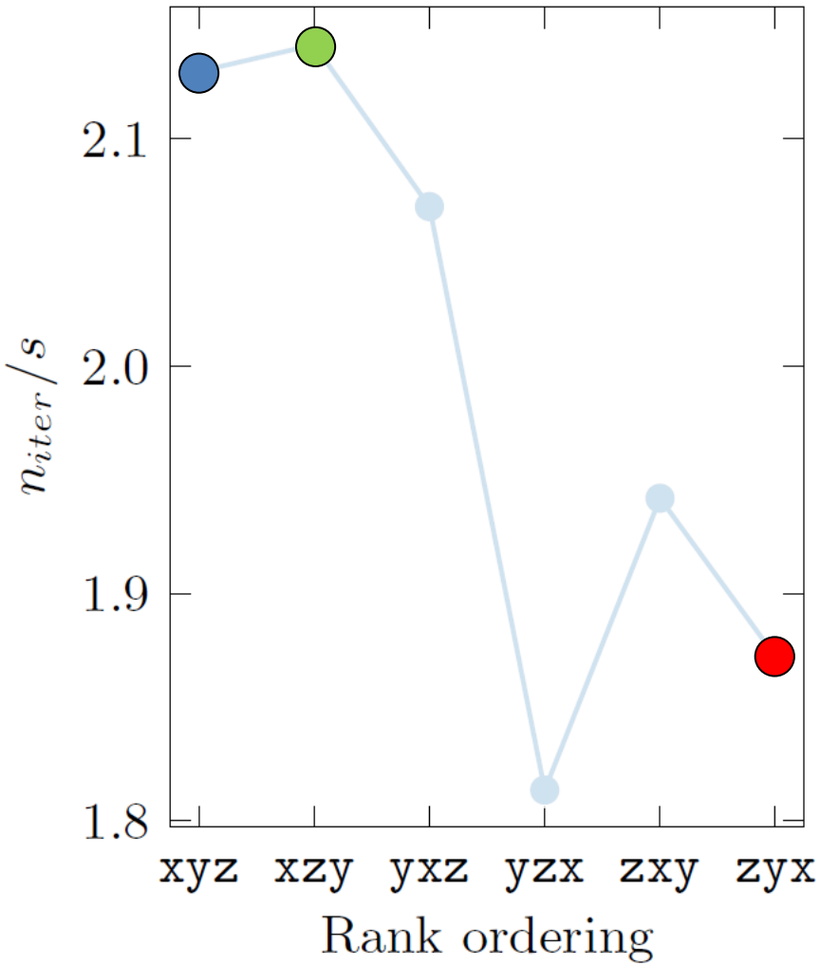




MPI decomposition: Influence of rank ordering

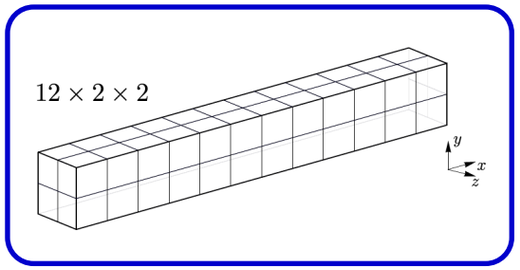
Institute of Aerodynamics
und Gas Dynamics

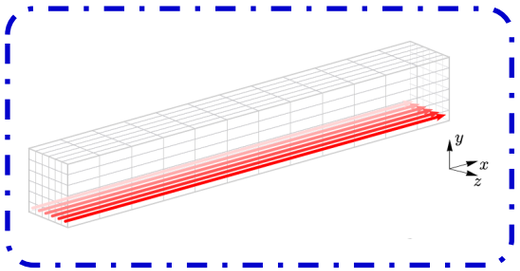


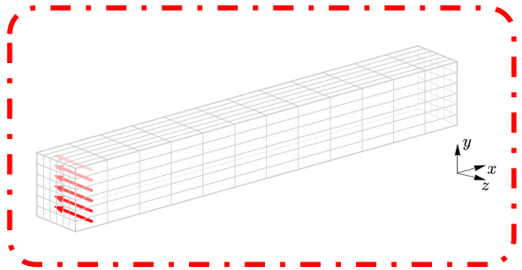


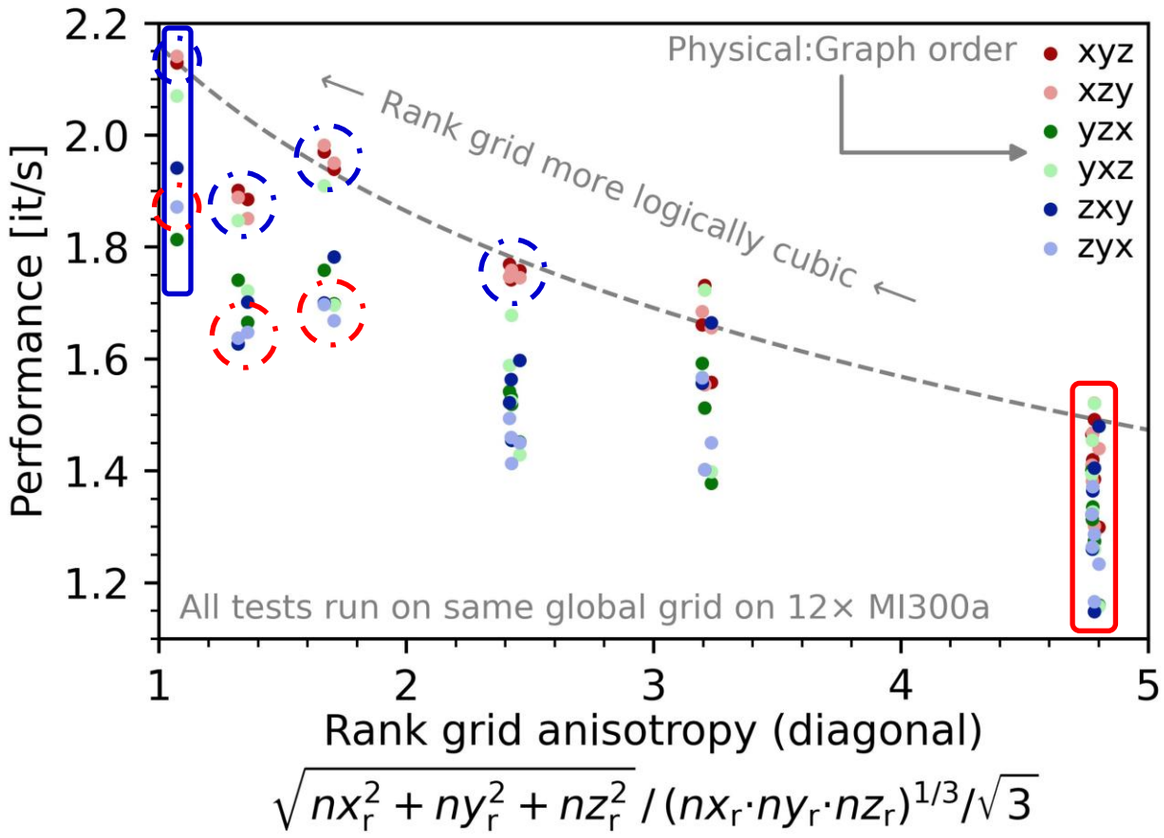
Summary: Bringing everything together...

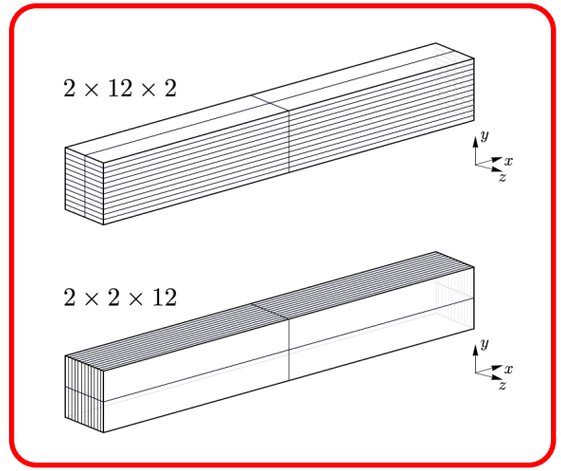
Institute of Aerodynamics
und Gas Dynamics







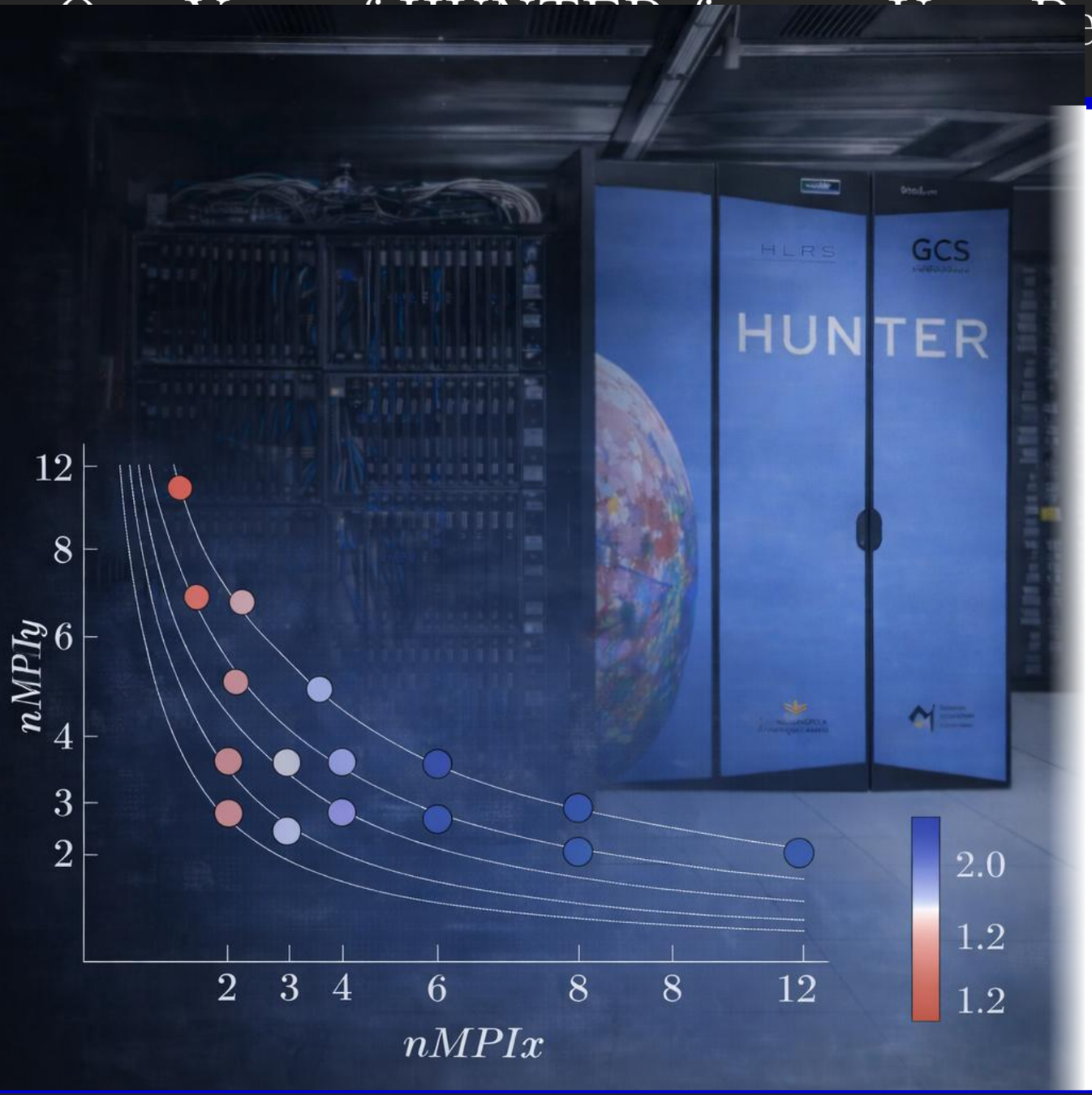




Summary: Bringing everything together...

If you are looking for long hanging fruits in performance, check you case design!

IAG
Institute of Aerodynamics
und Gas Dynamics



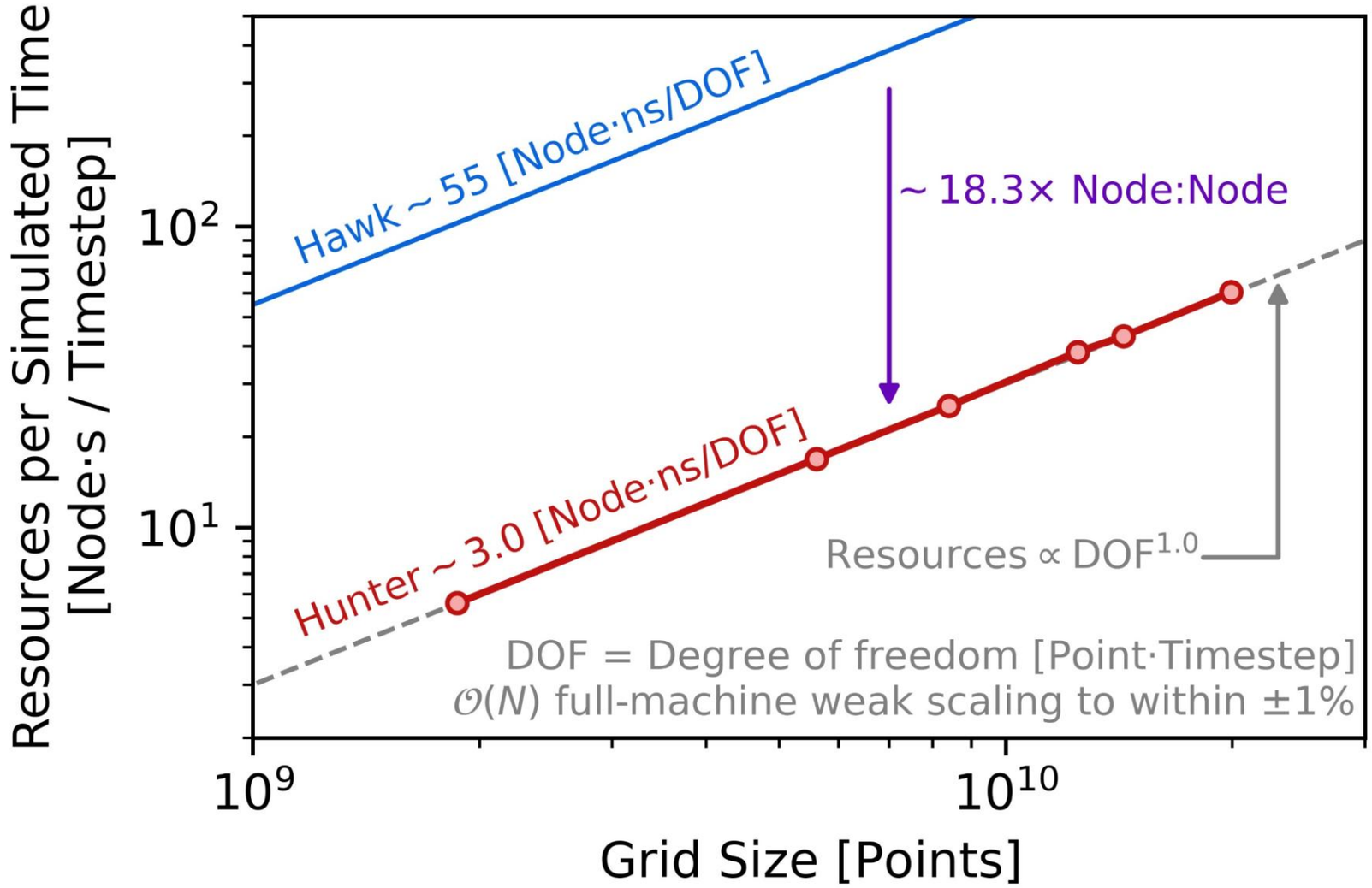
IAG
Institute of Aerodynamics und Gas Dynamics

- Same near-production test case as before:

n_{nodes}	$n_{\text{gridPoints}}$	n_x	n_y	n_z	$n_{\text{MPI},x}$	$n_{\text{MPI},y}$	$n_{\text{MPI},z}$	$n_{x,\text{rank}}$	$n_{y,\text{rank}}$	$n_{z,\text{rank}}$
12	1.868.175.360	5592	480	696	12	2	2	466	240	348
36	5.604.526.080	7456	720	1044	16	3	3	466	240	348
54	8.406.789.120	8388	720	1392	18	3	4	466	240	348
80	12.454.502.400	9320	960	1392	20	4	4	466	240	348
92	14.322.677.760	10718	960	1392	23	4	4	466	240	348
128	19.927.203.840	14912	960	1392	32	4	4	466	240	348

- Same test procedure as before
- All cases were computed twice for consistency
- Different MPI rank orderings were tested with consistent results, therefore only results for (`xyz`) are shown
- Lessons learned previously were applied, resulting in MPI domains that are as cubic as possible (in terms of n_x, n_y, n_z per MPI rank)

IAG
Institute of Aerodynamics
und Gas Dynamics



Introduction

NS3D GPU Porting Status

Performance insights

- Performance sensitivity to
  - MPI decomposition
  - Rank ordering
- Scaling behaviour

Summary/Conclusions

Summary and conclusions

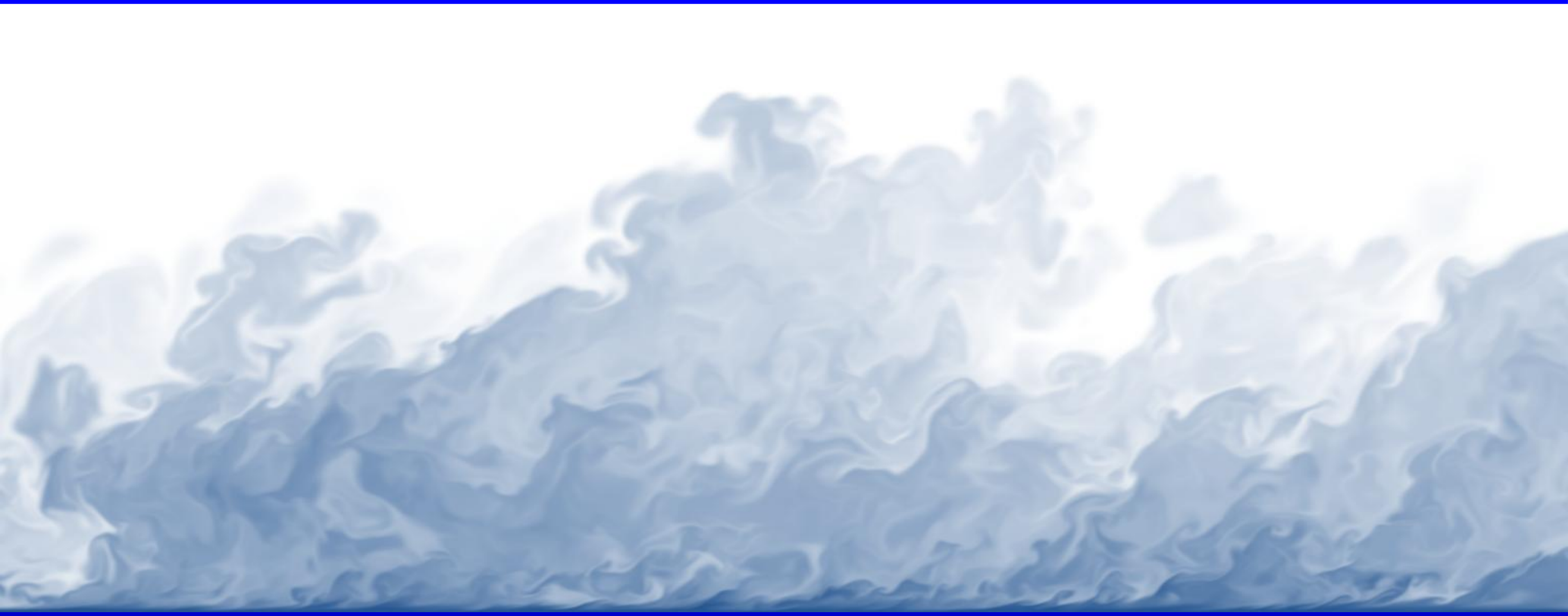
- **Starting point (Sept. 2025):** First production cases available, but limited robustness
- Many setbacks e.g. due to software-stack updates exposing previously hidden bugs in our code (mainly affecting large cases), along with limited MPI decomposition robustness
- Extensive debugging effort, largely relying on trial-and-error (and a fair amount of guesswork)
- Remaining mystery bugs: random NaNs or Infs in major arrays, HDF5 interface randomly fails

- **Current status:** (Largely) stable production runs; final NS3D_{neo}.develop version expected soon
- NS3D_{neo}.develop runs in both USM and non-USM modes

- **Performance insights (non-optimised version!):**
  - Strong sensitivity to MPI decomposition and rank ordering (up to $\times 2$)
  - Excellent weak scaling for near-optimal setups ($\sim$40M grid points per MPI rank)
  - Near-cubic MPI subdomains significantly improve performance
  - `xyz` rank ordering advantageous for current implementation

→ Next steps: Optimization!

IAG
Institute of Aerodynamics
und Gas Dynamics



One Year of HUNTER from a User Perspective: Progress, Performance, and Patience with NS3D

Christoph Wenzel

WSSP 2026

April, 2026