Ansys Fluent – nozzle flow CPU versus GPU runs

15.6.2023

Lewin Stein

ZUSE INSTITUTE BERLIN

Content

1. Ansys Fluent overview & licensing
2. Fluent TUI & example call
3. Setup of testcase – RANS of 3D steady subsonic nozzle flow
4. Tweaks needed for native GPU solver
5. Live demo: runtime & energy consumption CPU vs. GPU

Ansys www.hlrn.de/doc/display/PUB/Ansys+Suite

- big commercial engineering suite incl. two CFD solvers (unstructured* FVM):
  - **Fluent**: cell-centered, general (e.g. more mesh types), GPU acceleration*, more support ← todays focus
  - CFX: vertex-centered, specialized for turbomachinery
- HPC ready: brings own MPI, shared/distributed memory mode and pinning is auto deduced*
- **Fluent** solver differentiation (both for steady and transient flow):
  - pressure-based (not/mildly compressible)

    → segregated (predictor-corrector approach e.g. SIMPLE)

    → coupled (faster convergence* but larger system to solve/save in memory simultaneously)
  - density-based (highly compressible)

    → coupled – implicit

    → coupled – explicit (e.g. Runge-Kutta time stepper)

* apart of exceptions!

Ansys - licensing

- Do you fullfill the Ansys license conditions (industry financed projects are prohibited) ?
  If yes, apply at support@hlrn.de to become a Ansys group member and use our licenses.
- Alternatively, you can bring your own license: www.hlrn.de/doc/display/PUB/How+to+bring+your+own+license
- If you want to use our licenses always add `#SBATCH -L ansys` to your submission script !

licenses in Berlin

scontrol show lic	license type	nr.	module add (CPU cluster)	module add (Berlin GPU cluster)
aa_r	research: job number	25	19.0-2 (2018) 2019r2 2020r2 2022r1 2022r2 2023r1	2022r2_IntelMPI 2023r1_IntelMPI
aa_r_hpc	research: total core number beyond 4 (if GPU native, one A100 ≅ 72-99 cores ?)	2498		
aa_t_a	teaching: job number (4 core max.)	250		

Fluent example call

SLURM script:

```
#!/bin/bash
#SBATCH -t 00:10:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=96
#SBATCH -p standard96:test
#SBATCH -L ansys

module load ansys/2023r1
srun hostname -s > hostfile
```

hyperthreading is inadvisable

cas & dat input file

textUI.jou

```
file/read-cas initial_run.cas.h5
solve/iterate 10
file/write-case-data outputfile
exit
```

```
fluent 2d -g -t${SLURM_NTASKS} -mpi=intel -cnf=hostfile -i textUI.jou
```

- 2d → dim.
- -g → noGUI
- -t${SLURM_NTASKS} → task# (bind-to core is default)
- -mpi=intel → MPI type
- -cnf=hostfile → nodename list (bcn1001 bcn1001 bcn1001…)
- -i textUI.jou → instead of interactive input

Fluent example job - daytime sea breeze

www.hlrn.de/doc/display/PUB/Fluent

public

SEITENHIERARCHIE

Fluent

General computational fluid dynamics solver (cell-centered FVM). GPUs are supported.

General Information

ⓘ To obtain and checkout a product license please read Ansys Suite first.

Documentation and Tutorials

ⓘ Besides the official documentation and tutorials (see Ansys Suite), another alternative source is: https://cfd.ninja/tutorials

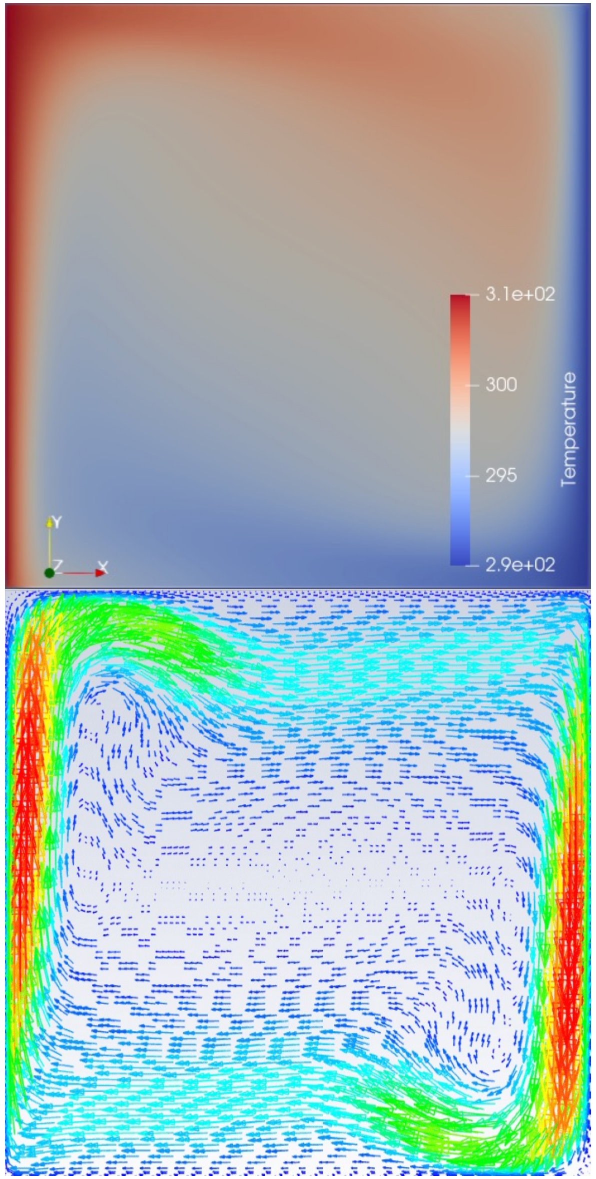
Example Jobscripts

The underlying test case described here can be downloaded here: NaturalConvection_SimulationFiles.zip

Job for 2 nodes with 40 tasks (on 40 cpu-cores) per node

```
#!/bin/bash
#SBATCH -t 00:10:00
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=40
#SBATCH -L ansys
#SBATCH -p medium
#SBATCH --mail-type=ALL
#SBATCH --output="cavity.log.%j"
#SBATCH --job-name=cavity_on_cpu

module load ansys/2019r2
srun hostname -s > hostfile
echo "Running on nodes: ${SLURM_JOB_NODELIST}"

fluent 2d -g -t${SLURM_NTASKS}  -ssh  -mpi=intel -cnf=hostfile << EOFluentInput >cavity.out.$SLURM_JOB_ID
      file/read-case initial_run.cas.h5
      parallel/partition/method/cartesian-axes 2
      solve/initialize/initialize-flow
      solve/iterate 100
      exit
      yes
EOFluentInput
```

constant hot boundary (land)



constant cold boundary (sea)

Testcase: 3D subsonic nozzle flow

- Source: Ansys Fluent Tutorial Guide (2023 R1)
  Ch. 8: Modeling Transient Compressible Flow – requires geometry file of nozzle
  Access only for Ansys group members:
  `/sw/eng/ansys_inc/v231/doc_tutorials/Fluent_Tutorial_Package/00 Ansys_Fluent_Tutorial_Guide_2023_R1.pdf`
  `/sw/eng/ansys_inc/v231/doc_tutorials/Fluent_Tutorial_Package/transient_compressible.zip`

- Old 2D guide (but freely accessible) 2D version – setup of nozzle geometry is included
  `https://alfaproject.ir/wp-content/uploads/2020/02/part6.pdf`

- Our 3D testcase is adapted from the source above. All needed files are included in
  `www.hlrn.de/doc/download/attachments/81232167/fluent_demo_cpu_vs_gpu.zip`
  These are:
  - nozzle_gpu_supported.cas.h5 → geometry & solver setup (adapted for the native GPU solver)
  - 00_submit_me_cpu.sh* → Slurm script to submit the job to a standard 96-core node
  - 00_submit_me_gpu.sh* → Slurm script to submit the job to a A100-GPU (with one host core)
  - tui_input.jou → Ansys text user interface commands to iterate the steady solver 1000 times
  - postprocessing_cmds.txt* → bash commands to check mass conservation & to extract runtimes/energy

* mashine specific

Ansys Fluent - GPU support

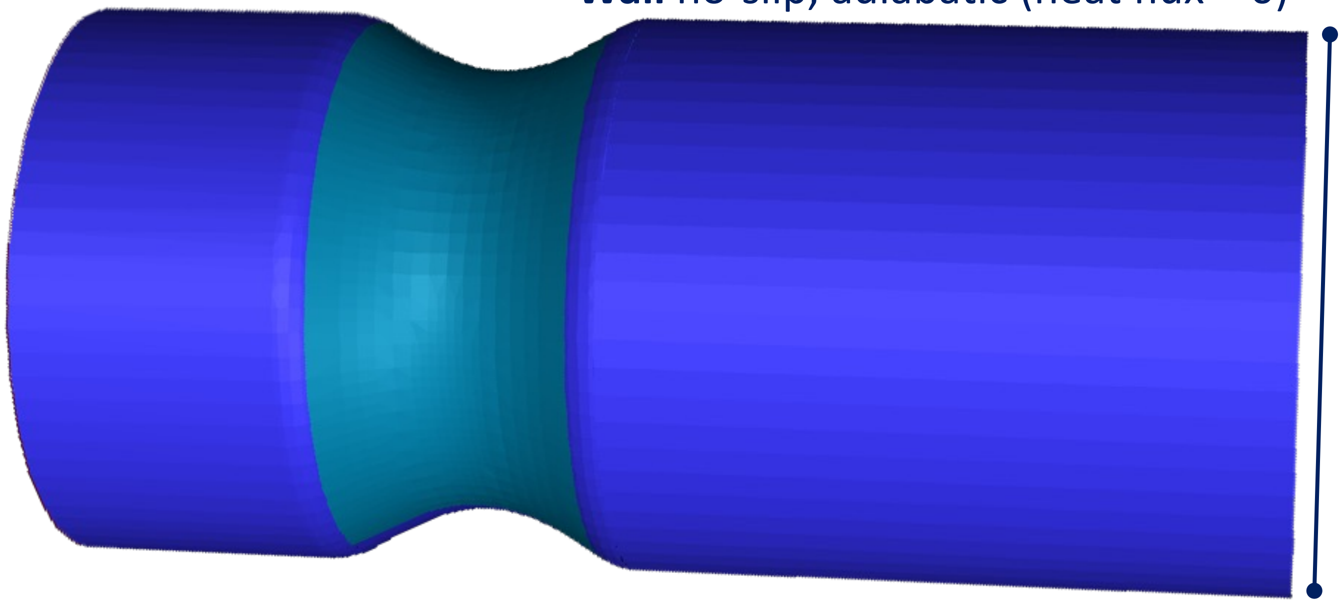
- Since 2014 GPU offload “-gpgpu” → best if linear eq. system solver of multigrid method dominates
- Since 2023 GPU native “-gpu” beta feature → most work is done by GPU, minimized CPU-GPU data movements
- The number of CPU-cores (e.g. ntasks-per-node=72) must be an integer multiple the GPUs (e.g. gres=gpu:4), all nodes must have the same layout

```
GPU solver defaults are activated for the following unsupported settings (2)
==================================================
Area    Settings        From        To
--------------------------------------------------
solver  coupling scheme coupled     simple
solver  method          pseudo time turned off


> solve/initialize/hyb-initialization
Preparing GPU solver, please wait ...
------------------------------------------------------------------------------------------------------------------------
Rank  PID     Core   OS       Host     Device  Name                    Version  Memory(GB)  Bandwidth(GB/s)  Cores
------------------------------------------------------------------------------------------------------------------------
0     719200  1/144  lnamd64  bgn1010  0/4     NVIDIA A100-SXM4-80GB   8.0      79.1995     1944.58          108x64
------------------------------------------------------------------------------------------------------------------------

Non-released features (1)
*****************************
Feature            Exposure
-----------------------------
Compressible Flow  Beta
```

Subsonic nozzle flow – setup

Wall no-slip, adiabatic (heat flux = 0)

Pressure inlet
u = hybrid init.
p = 0.9 atm
T = 300°C



0.2 m

Pressure outlet
u = hybrid init.
p = 0.7369 atm
T = 300°C

Air as ideal gas
c_p = 1006.43 J/kg/K (specific heat)
λ = 0.0242 W/m/K (therm. conductivity)
μ = 1.7894 10^{-5} kg/m/s (dyn. viscosity)
M = 28.966 g/mol (molecular weight)

Turbulence model
RANS: SST k–ω (2 eqn, shear stress transport)

Solution methods and models (FVM)
density-based (coupled, compressible)
viscous, energy eq. with viscous heating (λ term)
flux type: Roe flux-difference splitting (FDS)

- **time discretization**

steady, implicit, pseudo time-stepping, CFL = 25

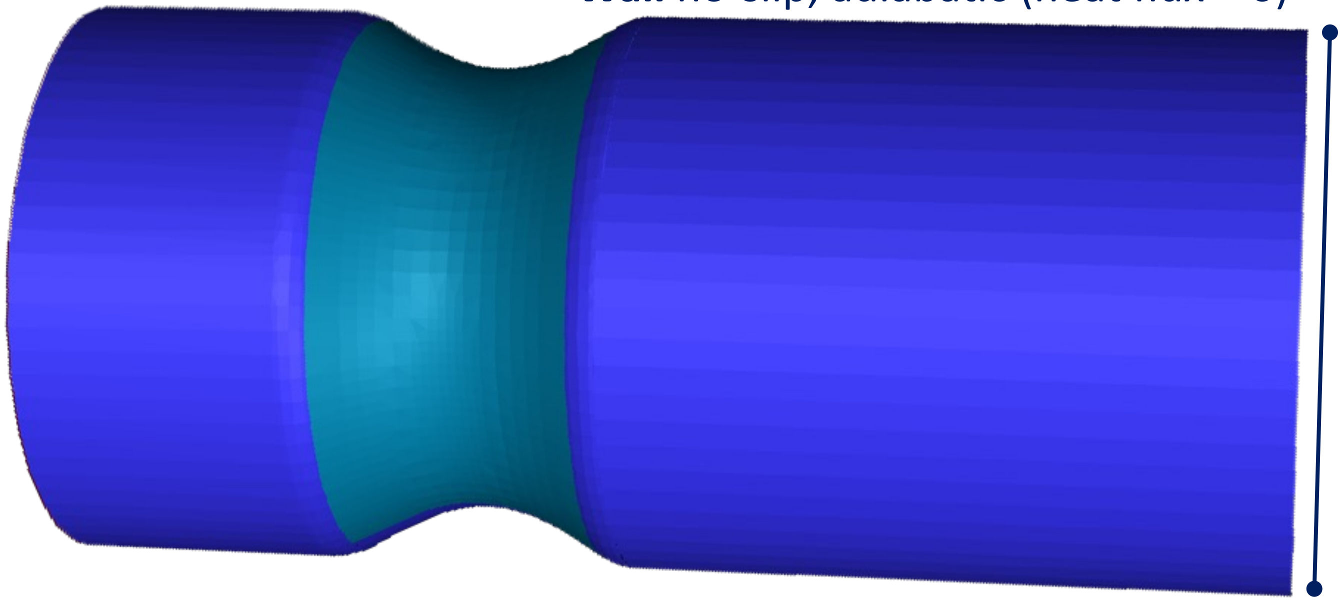
- **spatial discretization**

Gradient terms: least squares cell-based
Other transport terms (flow, k, ω): 2^{nd} order upwind

Subsonic nozzle flow – setup for GPU native

Wall no-slip, adiabatic (heat flux = 0)

Pressure inlet
u = hybrid init.
p = 0.9 atm
T = 300°C



0.2 m

Pressure outlet
u = hybrid init.
p = 0.7369 atm
T = 300°C

Air as ideal gas

c_p = 1006.43 J/kg/K (specific heat)
λ = 0 ? (therm. conductivity)
μ = 1.7894 10^{-5} kg/m/s (dyn. viscosity)
M = 28.966 g/mol (molecular weight)

Turbulence model

RANS: SST k–ω (2 eqn, shear stress transport)

Solution methods and models (FVM)

pressure-based (segregated, mildly compressible)
viscous, energy eq. ~~with viscous heating (λ-term)~~
flux type: Rhie-Chow - momentum based

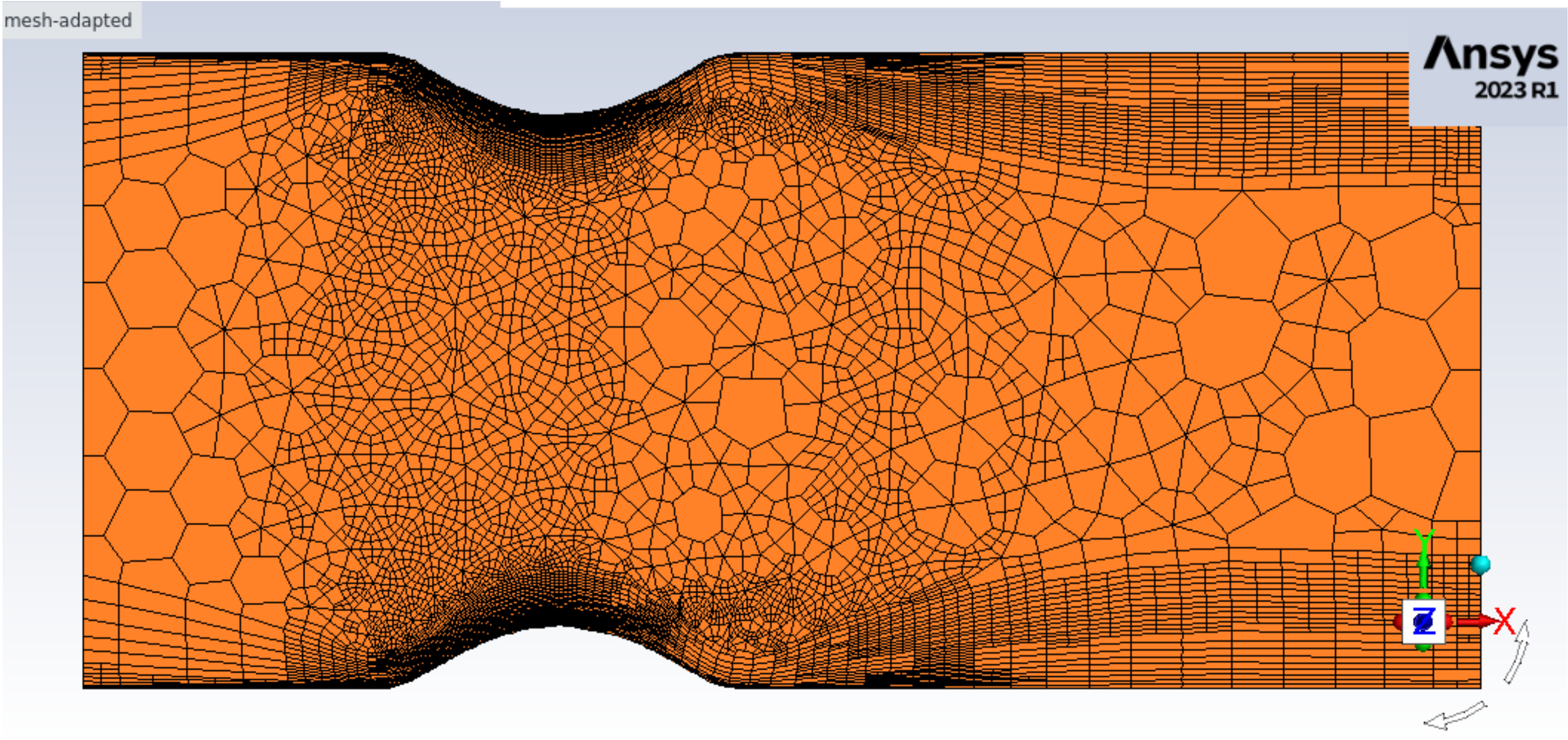
- **time discretization**

steady, SIMPLE, ~~pseudo time-stepping,~~ CFL = 25

- **spatial discretization**

Gradient terms: least squares cell-based
Other transport terms (flow, k, ω): 2nd order upwind

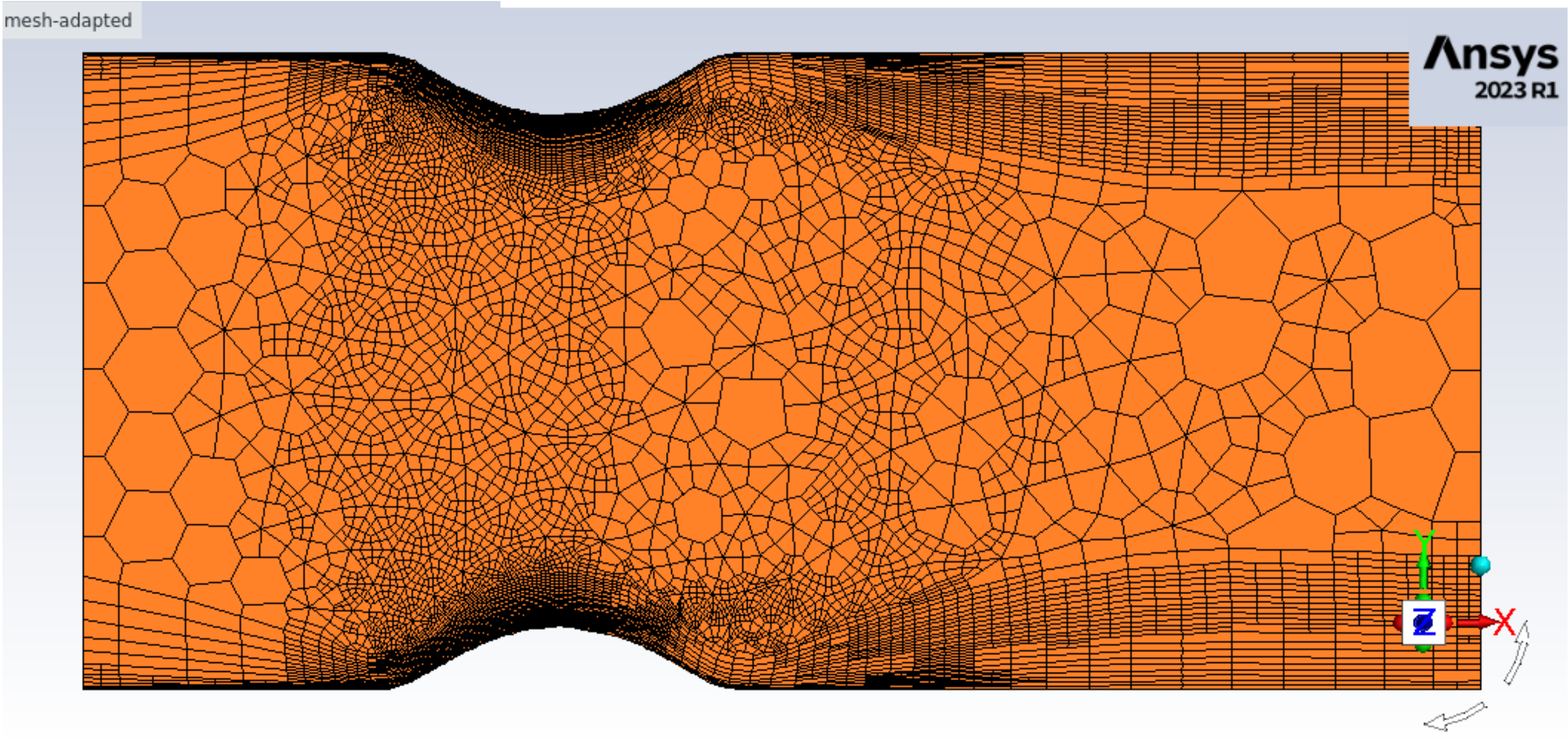
Subsonic nozzle flow – automatic adaption



Volume Mesh
Elements = poly-hexcore
Min. Cell Length = 5 mm
Max. Cell Length = 20 mm

Automatic Mesh Adaption
Frequency (iteration) = 20
Criterion: shock indicator - density-based

Subsonic nozzle flow – automatic adaption



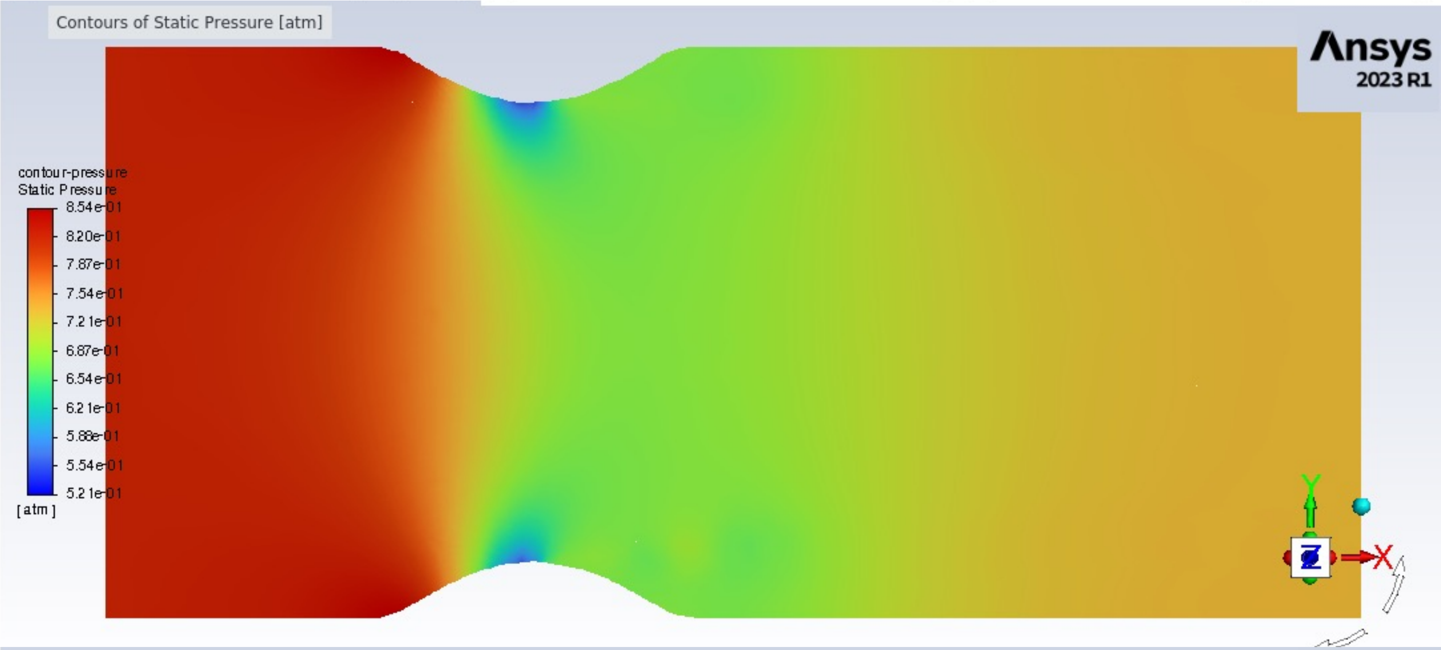
Volume Mesh
Elements = poly-hexcore
Min. Cell Length = 5 mm
Max. Cell Length = 20 mm

~~**Automatic Mesh Adaption**~~
~~Frequency (iteration) = 20~~
~~Criterion: shock indicator - density-based~~

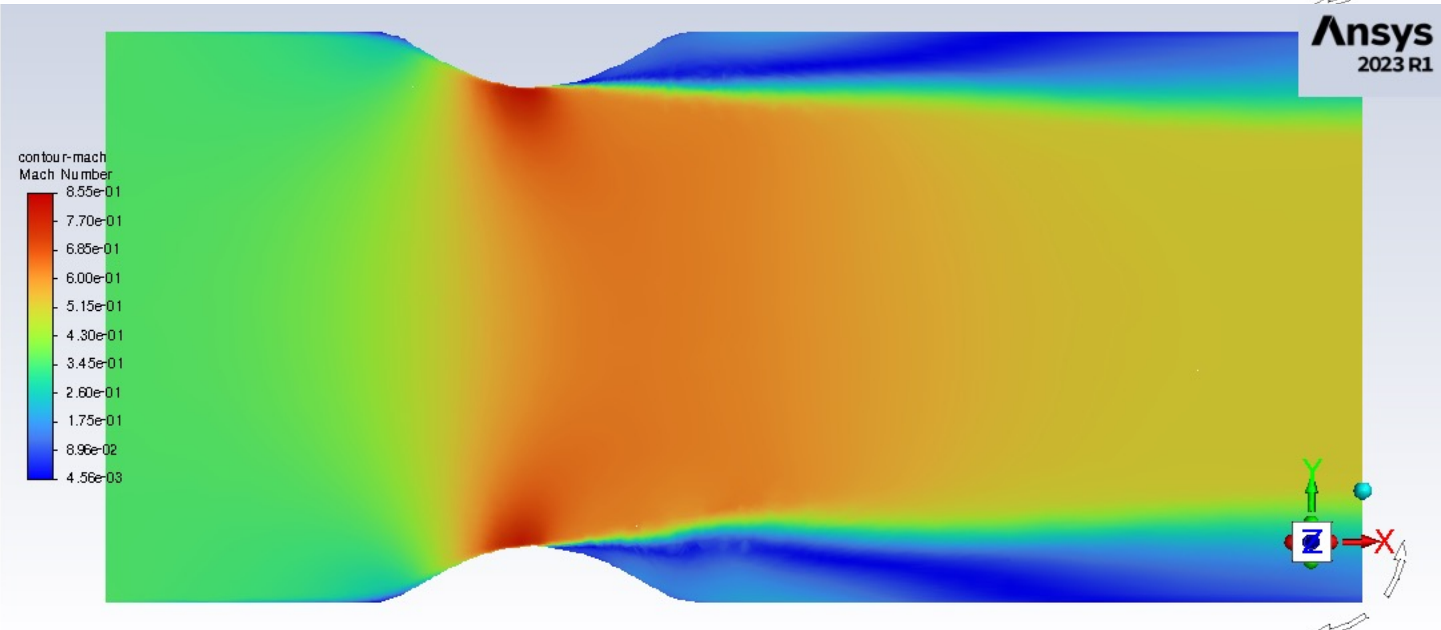
deactivated for GPU native

Subsonic nozzle flow - solution

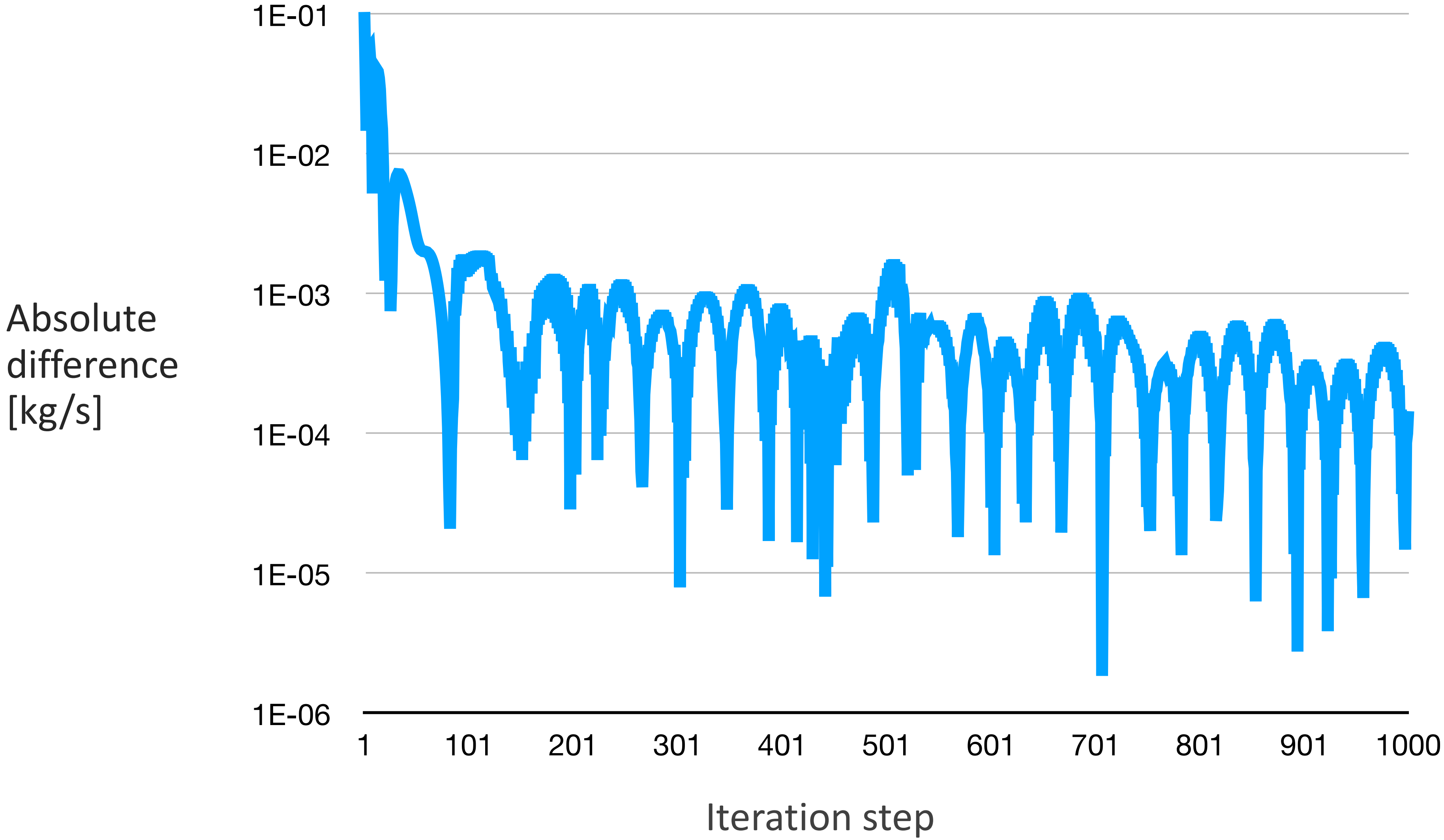
pressure



Mach number



Mass flow rate - difference inlet/oulet



Subsonic nozzle flow – CPU vs. GPU

ZIB
ZUSE INSTITUTE BERLIN

Node type	Time per step [s]	Speedup (normalized)	Energy per step [Watt s]	Energy used per step (normalized)
CPU only – 96 core (360GB)	0.114	1	161.251	2.1
GPU offload – 1 core, 1 A100 (80GB)	6.035	0.02		
GPU native – 1 core, 1 A100 (80GB)	0.037	3.1	76.597	1
GPU native – 4 core, 4 A100 (4x80GB)	0.054	2.1		

Thank You

Any questions or feedback ?