WSSP 41, Stuttgart 2026

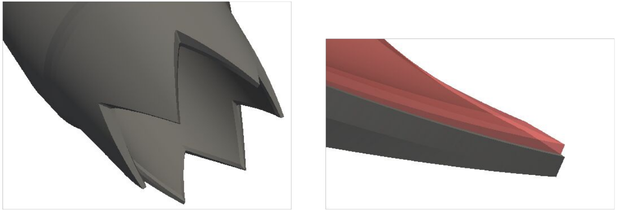
Aeroacoustic Optimization of a Chevron Nozzle on the Hunter HPC System

Matthias Meinke
A. Niemöller, D. Krug
m.meinke@aia.rwth-aachen.de

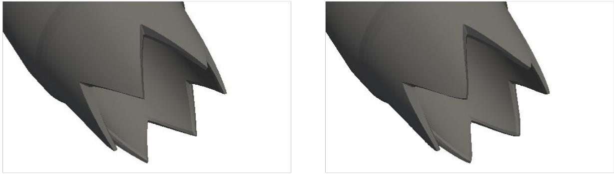
Institute of Aerodynamics
RWTH Aachen University
Aachen, Germany

Optimization of a Chevron Nozzle

- Shape optimization of a chevron nozzle as an engieering HPC application with the motivation to mitigate aeroacoustic noise
- Activities embedded in the European Center of Excellence **EXCELLERAT**
- Noise predictions based on the m-AIA simulation framework



Penetration angle: baseline $\alpha = 5^\circ$ (grey), 8° (red)

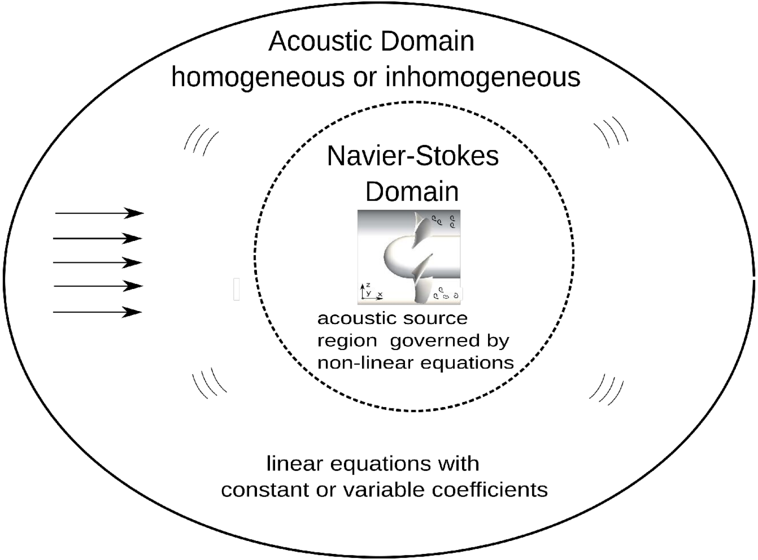


S=0.55

S=0.75

Geometry = S · sinusoidial + (1 - S) · triangular shape

Computational Aeroacoustics (CAA)



Acoustic Domain
homogeneous or inhomogeneous

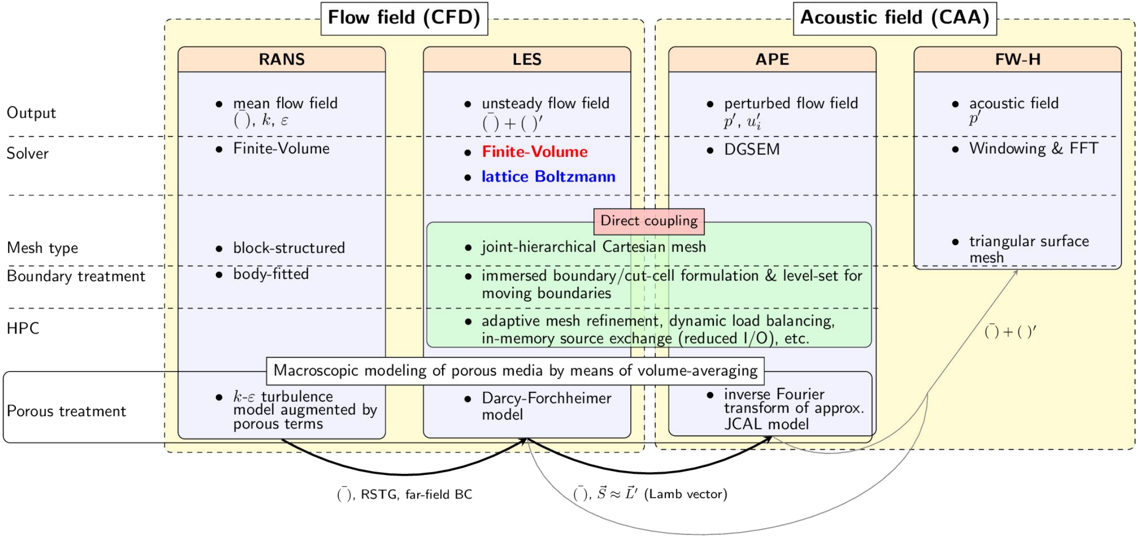
Navier-Stokes Domain

acoustic source region governed by non-linear equations

linear equations with constant or variable coefficients

Zorumski, Comp. Aeroac. 1993

m-AIA Toolchain for Noise Prediction



	Flow field (CFD)		Acoustic field (CAA)	
	RANS	**LES**	**APE**	**FW-H**
Output	mean flow field $\overline{(\,)}$, k, ε	unsteady flow field $\overline{(\,)}+(\,)'$	perturbed flow field p', u_i'	acoustic field p'
Solver	Finite-Volume	**Finite-Volume** **lattice Boltzmann**	DGSEM	Windowing & FFT
Mesh type	block-structured	Direct coupling: joint-hierarchical Cartesian mesh		triangular surface mesh
Boundary treatment	body-fitted	immersed boundary/cut-cell formulation & level-set for moving boundaries		
HPC		adaptive mesh refinement, dynamic load balancing, in-memory source exchange (reduced I/O), etc.		
Porous treatment	k-ε turbulence model augmented by porous terms	Darcy-Forchheimer model	inverse Fourier transform of approx. JCAL model	

Macroscopic modeling of porous media by means of volume-averaging

$\overline{(\,)}$, RSTG, far-field BC

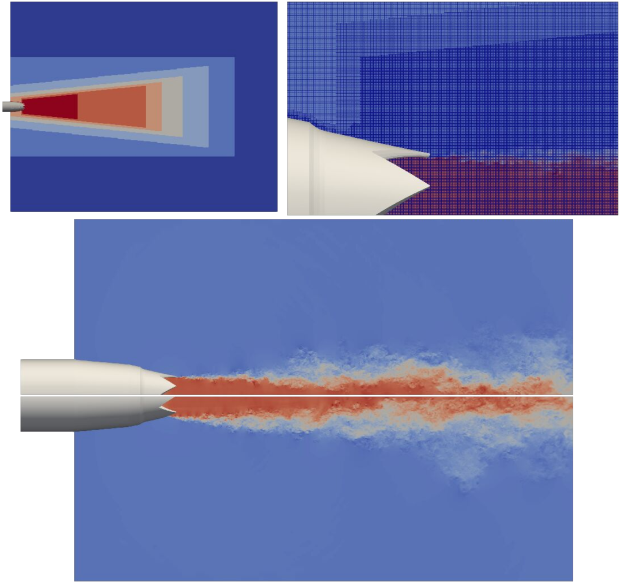
$\overline{(\,)}$, $\vec{S} \approx \vec{L}'$ (Lamb vector)

$\overline{(\,)}+(\,)'$

Simulation of Chevron Nozzle Jets

Shape optimization of nozzle shapes

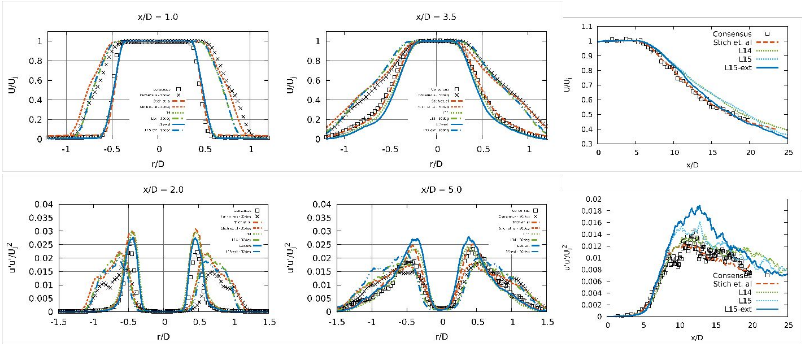
- Automation of aeroacoustic prediction workflow to be tested for increasingly complex problems → Chevron shape optimization
- Large-scale aeroacoustic simulations of the SMC001 chevron nozzle at Re=10^6 → Validation and data analytics
- grid refinement study:
  CFD: 1.5 → 2.1 → 3.7 billion cells
- CAA: 77 million elements → 4.9 billion DOF
- Complete turbulent flow field to acoustic far-field approach: CFD → coupled CFD/CAA → FWH

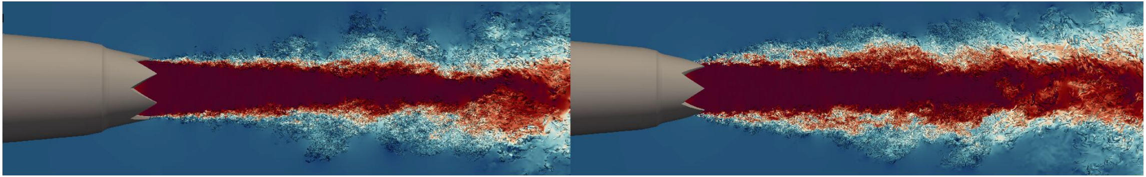


Simulation of Chevron Nozzle Jets

SMC001 Flow field results

- Overall good agreement with reference data
- Overprediction of turbulence intensities downstream of potential core end on highest resolution grid
- Uncertainties: e.g. nozzle synthetic turbulence inlet BC,
- Impact of CFD resolution on acoustic far-field noise?

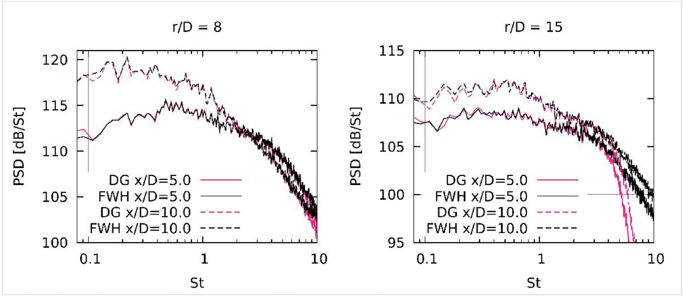


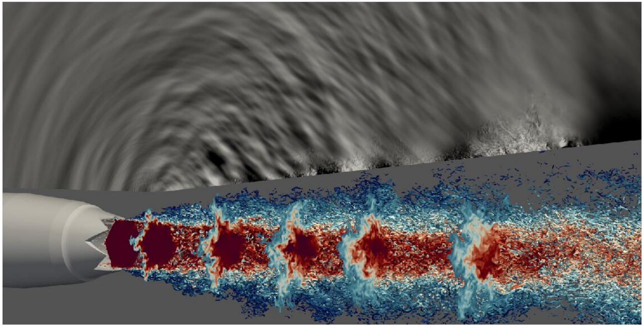


Simulation of Chevron Nozzle Jets

SMC001 Acoustic field

- Comparison of DG and DG-FWH predictions in near far-field
- DG: high-resolution up to r/D=8; frequency cut-off at larger distances
- FWH utilizes high-resolution data sampled on a surface enclosing the acoustic sources
- Negligible impact of exact surface placement
- High surface resolution: almost identical results with 50% less surface elements
- Highest refined case: CFD 3.7B cells, CAA 4.9B DOF
- CFD only: 2048 nodes
  Coupled CFD-CAA: 4096 nodes

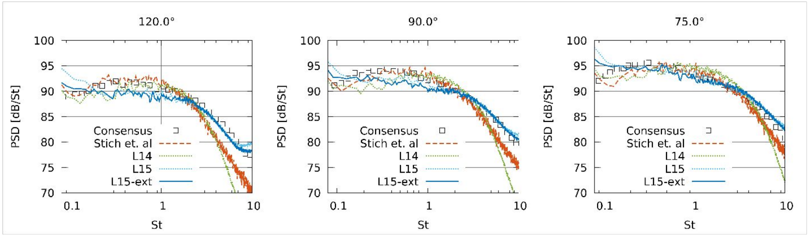


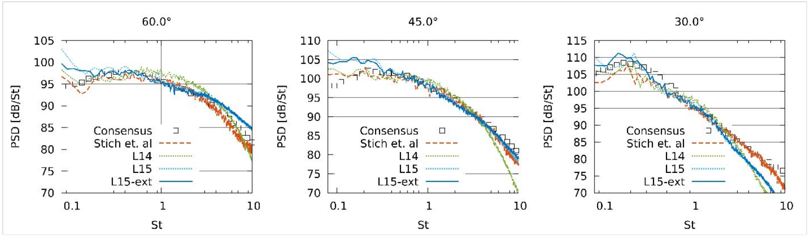


Simulation of Chevron Nozzle Jets

SMC001 Acoustic far-field

- Power spectral density of acoustic pressure on arc r/D = 100
- Overall high level of agreement for highest resolution grid
- Missing high-frequency content and mid-frequency overprediction for lower resolution grid
- Low impact of extension of high-resolution acoustic source region (L15 → L15-ext)



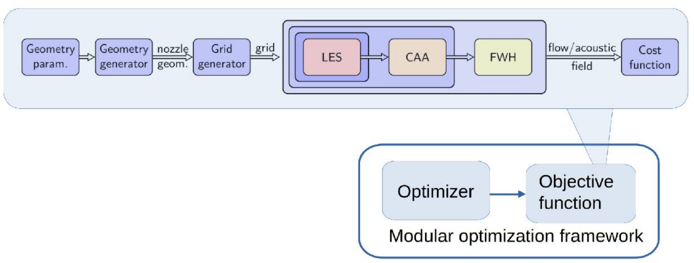


Optimization of a Chevron Nozzle

Workflow Implementation

Single objective function evaluation

- Preprocessing
  - Chevron geometry: Matlab/Octave
  - Mesh generation: joint hierarchical Cartesian grid
- CFD: Finite-Volume solver
- Coupled CFD/CAA: Discontinuous Galerkin method for CAA
- FWH: time or frequency-domain method
- Postprocessing: Python / m-AIA



→ Automated simulation pipeline (easy to use/adapt/extend)

Full workflow automation is necessary for the optimization framework

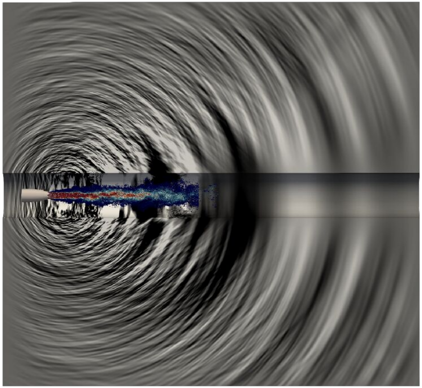
- Input: Chevron nozzle shape, flow parameters, configuration files, jobscript templates
- Optimizer: shape parameters, design space, constraints, . . .
- Output: Objective function (noise reduction, thrust loss), sampled data, slices of flow/acoustic field

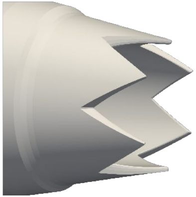
→ Optimized chevron shape parameters

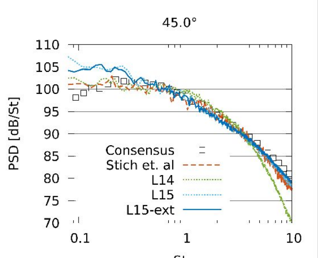
Optimization of a Chevron Nozzle

Case Setup

- Geometry definition based on SMC001 chevron nozzle
- High-resolution simulations performed at $Re_D=1\cdot10^6$
- Flow conditions for shape optimization:
  - Cold jet flow at Ma=0.9, $T_j=0.84$
  - Reynolds number: $Re_D=20{,}000$
  - Smaller scale simulation using O(10-50 $\cdot10^6$) cells
- Objective function
  - lower frequency noise reduction ($St_{min} < f < St_{max}$), range of observer angles
  - thrust loss penalty

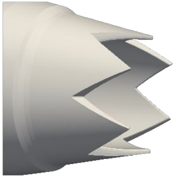


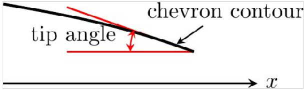


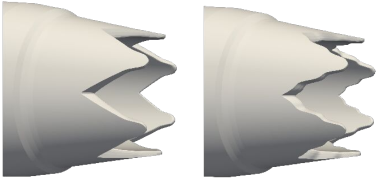


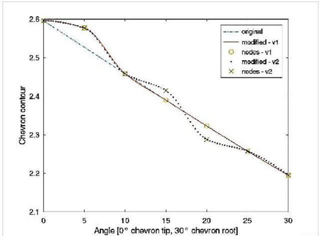
Optimization of a Chevron Nozzle

- Geometry definition based on SMC001 chevron nozzle
- **Geometry parametrization:** 6 symmetric chevron lobes
- Chevron **tip angle**
- Chevron **shape:** triangular baseline
  - Add N points between chevron tip and root, 1 DoF each (e.g. N=5)
  - Piecewise Cubic Hermite Interpolating Polynomial (PCHIP): interpolates data points, continuous first derivative, shape preserving – no overshoots, few oscillations
- Automatic geometry generation in STL format (Matlab/Octave)
- $N+1$ **parameters:** high geometric flexibility



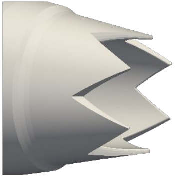


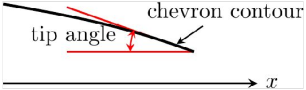


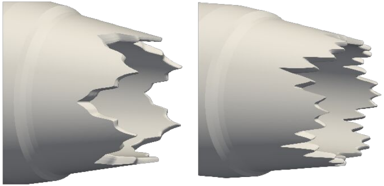


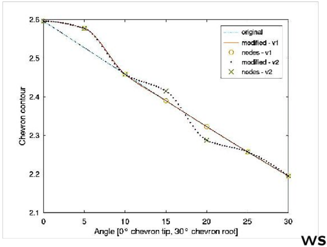
Optimization of a Chevron Nozzle

- Geometry definition based on SMC001 chevron nozzle
- **Geometry parametrization:** 6 symmetric chevron lobes
- Chevron **tip angle**
- Chevron **shape:** triangular baseline
  - Add N points between chevron tip and root, 1 DoF each (e.g. N=5)
  - Piecewise Cubic Hermite Interpolating Polynomial (PCHIP): interpolates data points, continuous first derivative, shape preserving – no overshoots, few oscillations
- Automatic geometry generation in STL format (Matlab/Octave)
- **$N+1$ parameters:** high geometric flexibility





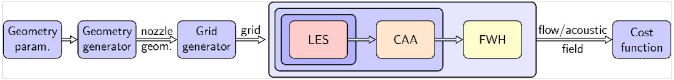


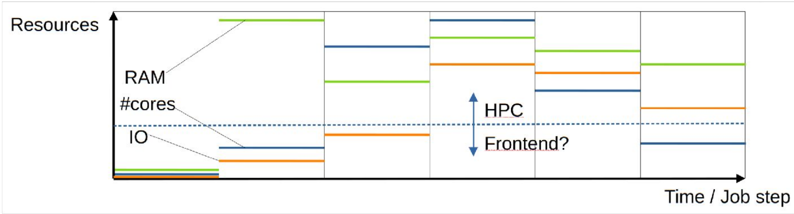


Optimization of a Chevron Nozzle

Workflow

- Dependencies of workflow steps
- Differing resource requirements for each workflow stage
- Simultaneous evaluation of multiple designs by the Bayesian optimizer (FhG)

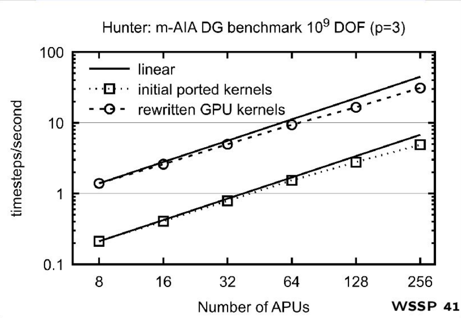

Geometry param.
Geometry generator
nozzle geom.
Grid generator
grid
LES
CAA
FWH
flow/acoustic field
Cost function



Target system: Hunter @ HLRS
AMD MI300A (x4 per node)
128 GB HBM3 per socket

- **Goal:** single function evaluation on 1 node, <1 day runtime, use multiple nodes given decent strong scalability for small scale setup
- **Ongoing work:** optimize APU performance, eliminate remaining bottlenecks, enhance robustness

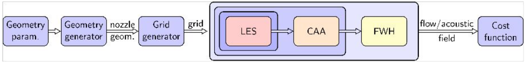
Rewritten DG kernels for MI300A

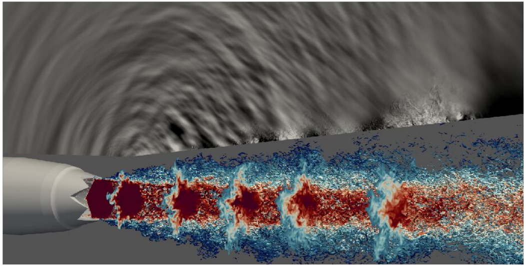


Optimization of a Chevron Nozzle

Simulation stage #1: CFD/LES

- 4 Hunter nodes / 16 MPI ranks, 1.6m 3.3m cells per APU
- **Run #1:** initial jet flow development
  - low CFL number, dynamic load balancing, ≈9min runtime
- **Run #2:** advance to obtain fully developed flow field
  - $T=185\ D/c_0$, 0.0294 s/timestep, ≈25min runtime
- **Run #3:** compute mean flow field
  - $T_{avg}=200\ D/c_0$, ≈30min runtime
- **Overall: ≈40min** ≈65min runtime on 4 nodes
- **Start from fully developed flow field: ≈45min**

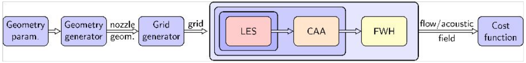




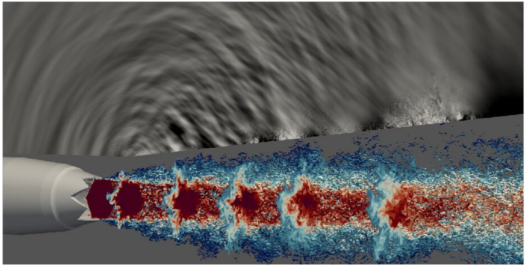
Optimization of a Chevron Nozzle

Simulation stage #2: Coupled CFD-CAA

- 4 Hunter nodes / 16 MPI ranks
- 3.3m FV cells and 6.1m DG DoF per APU, memory ≈16GB per process
- **Run #1:** initial run with dynamic load balancing, no data sampling, ≈8min runtime
- **Run #2:** Acoustic prediction with surface sampling in the near field
  - T_{mboxac}=240 D/c_0, 0.089 s/timestep, ≈75min runtime
- **Overall: 1.4h** runtime on 4 nodes



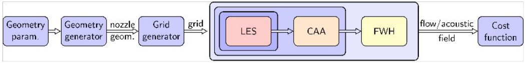
Geometry param.
Geometry generator
nozzle geom.
Grid generator
grid
LES
CAA
FWH
flow/acoustic field
Cost function

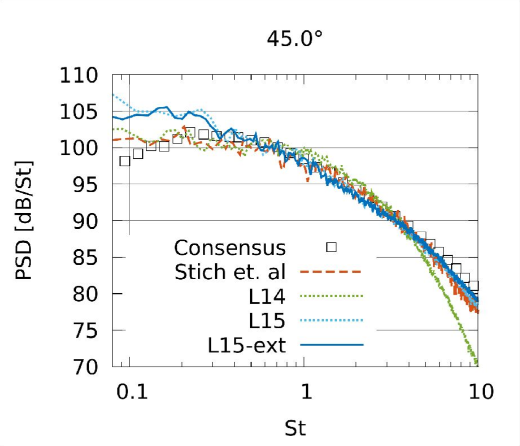


Optimization of a Chevron Nozzle

Simulation stage #3: FWH far-field prediction

- 1 Hunter node / 4 MPI ranks
- FWH surface: 470k surface elements
- 6144 samples, dt=0.03 D/c_0, T=184 D/c_0
- Data size: 86GB
- Welch's method for PSD estimation: window size of 1024 samples, 50% overlap
- ≈90GB memory per process → finer surfaces/more samples per window require more nodes to be used
- 1368 observer points in the far-field at R=100 D
- **Runtime:** ≈ 7min



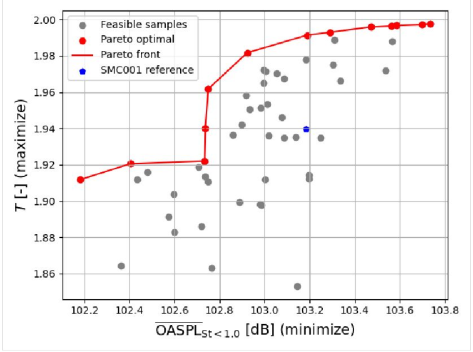


Optimization of a Chevron Nozzle

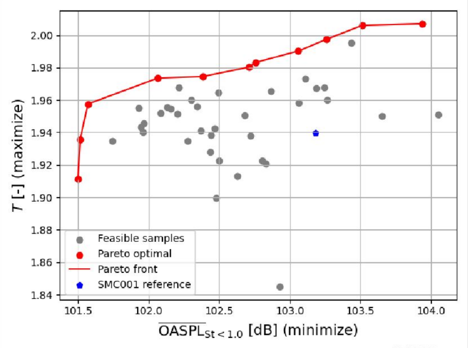
Optimization using 4-shape parameters

- Parameter space:
  alpha $\in [0^\circ, 10\circ]$, x1-x3 $\in [0.0, 1.0]$
- Work in progress: 46 feasible configurations finished
- Overall $\approx$ 100 simulations (2 and 4 shape parameters)
- 4p case more promising in terms of noise reduction
- 4p optimization can be augmented using the 2p results

Pareto front (2 shape parameters)



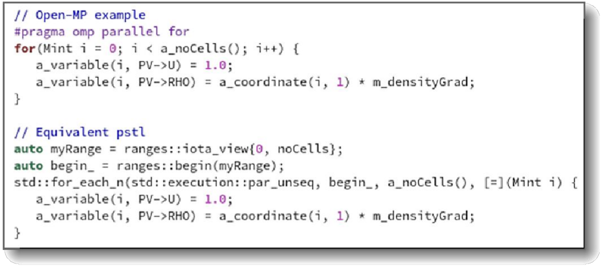
Pareto front (4 shape parameters)



Update of m-AIA APU Porting

m-AIA is ported to APU/GPUs using the **Parallel Standard Template Library (PSTL)**

- Introduced in C++ since version 17
- Adds support of parallel execution to existing functions from <algorithm>
- Examples: `for_each`, `sort`, `find`, `transform_reduce`, ...
- Actual parallel implementation is provided by Vendors (AMD, NVIDIA)
- Other GPU programming models (OpenMP, SYCL, . . .)
- Unifies CPU and GPU acceleration
- GPU: Memory transfer is implicitly managed and triggered by page faults on host/device



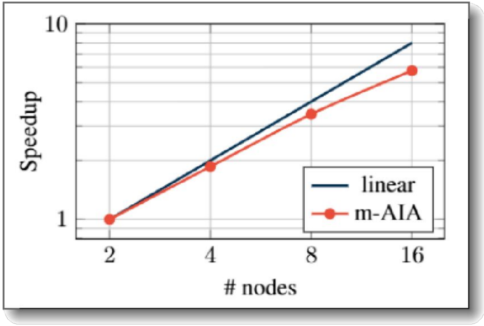
```cpp
// Open-MP example
#pragma omp parallel for
for(Mint i = 0; i < a_noCells(); i++) {
   a_variable(i, PV->U) = 1.0;
   a_variable(i, PV->RHO) = a_coordinate(i, 1) * m_densityGrad;
}

// Equivalent pstl
auto myRange = ranges::iota_view{0, noCells};
auto begin_ = ranges::begin(myRange);
std::for_each_n(std::execution::par_unseq, begin_, a_noCells(), [=](Mint i) {
   a_variable(i, PV->U) = 1.0;
   a_variable(i, PV->RHO) = a_coordinate(i, 1) * m_densityGrad;
}
```

Example for AMD compilation: `clang++ -hipstdpar -offload-arch=gfx ...`

Update of m-AIA APU Porting

m-AIA Lattice Boltzmann

- LB solver has been ported first due to its simple algorithm
- Main compute kernels:
    - Collision (local, high arithmetic intensity)
    - Propagation (non-local, no calculations)
    - MPI exchange



Taylor Green Vortex case with 40M cells

Kernel	CPU 128C AMD Genoa	GPU 1×MI300A	Ratio
Collision	0.0799	0.030730	2.6
Propagation	0.0901	0.029006	3.1
Exchange	0.0593	0.002609	-

Memory bandwidth ratio MI300A/Genoa is 3.8

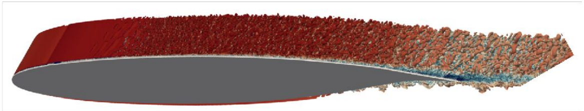
Work in Progess/Future changes:

- Mixed precision to save memory bandwidth
- Further compute kernel fusion
- Interleaving MPI communication with compute kernels

Update of m-AIA APU Porting

m-AIA Finite-Volume Solver

Flow around an airfoil at Re=800,000, M=0.6
$30 \cdot 10^6$ mesh cells



Initial Porting

Timer	CPU [s]	SoA [s]	Speedup
Time-integration	841.33	464.42	1.81

Porting from AoS to SoA

Timer	CPU [s]	SoA [s]	Speedup
Time-integration	841.33	301.63	2.79

Further Optimization

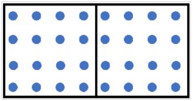
- Fusion of convective and viscous flux kernels
- Reverse flux integral from push to pull

Timer	CPU [s]	SoA [s] Opt.	Speedup
Time-integration	841.33	234.89	3.58

Update of m-AIA APU Porting

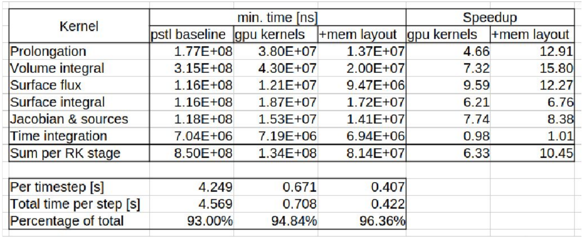
m-AIA Discontinuous Galerkin Method

- Solution of the acoustic perturbation equations for coupled CFD/CAA simulations
- Initial porting, looping over elements/surfaces



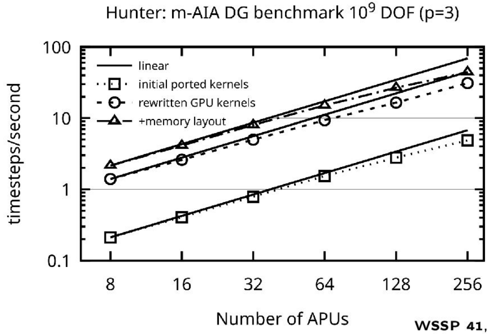
- Optimization A: Looping over nodes directly, merge kernels
- Optimization B: Changing from AoS to SoA memory layout

Initial PSTL	Opt. A	Opt. B
1.0	6.33	10.45



Kernel	min. time [ns]			Speedup	
	pstl baseline	gpu kernels	+mem layout	gpu kernels	+mem layout
Prolongation	1.77E+08	3.80E+07	1.37E+07	4.66	12.91
Volume integral	3.15E+08	4.30E+07	2.00E+07	7.32	15.80
Surface flux	1.16E+08	1.21E+07	9.47E+06	9.59	12.27
Surface integral	1.16E+08	1.87E+07	1.72E+07	6.21	6.76
Jacobian & sources	1.18E+08	1.53E+07	1.41E+07	7.74	8.38
Time integration	7.04E+06	7.19E+06	6.94E+06	0.98	1.01
Sum per RK stage	8.50E+08	1.34E+08	8.14E+07	6.33	10.45

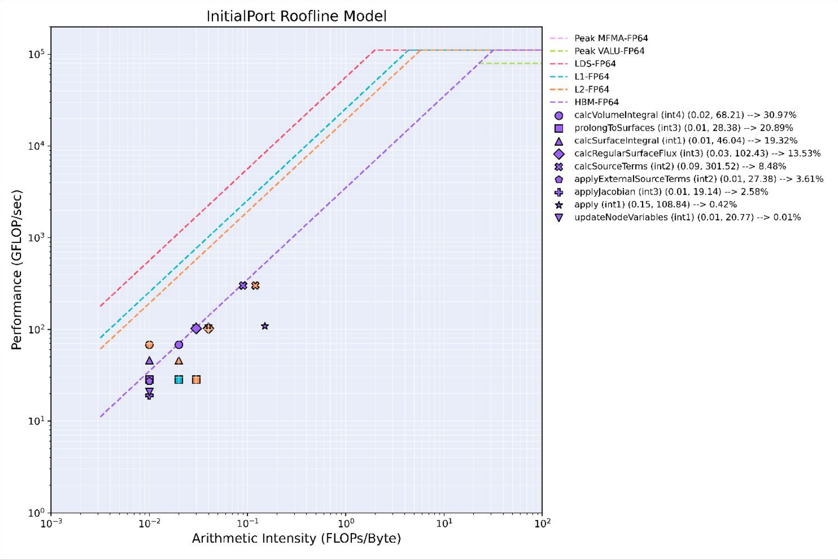
Per timestep [s]	4.249	0.671	0.407
Total time per step [s]	4.569	0.708	0.422
Percentage of total	93.00%	94.84%	96.36%



Update of m-AIA APU Porting

m-AIA Discontinuous Galerkin Method

Initial PSTL Porting looping over elements/surfaces



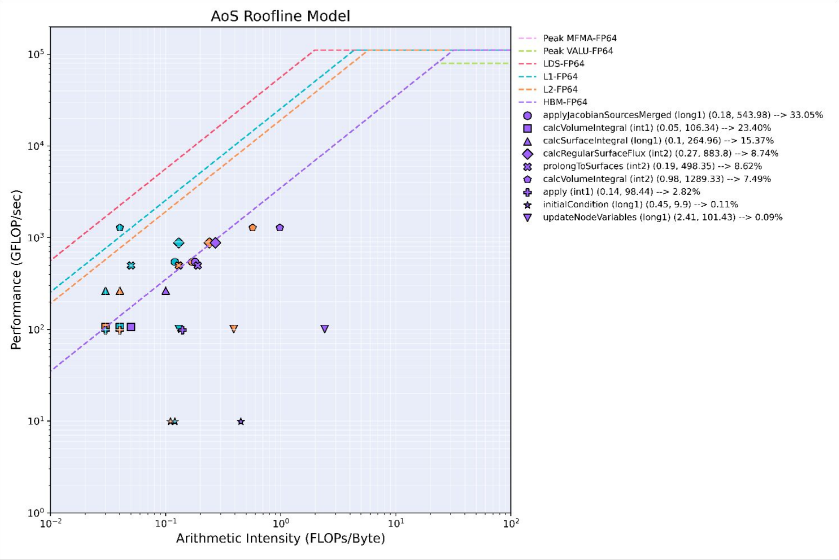
InitialPort Roofline Model

Update of m-AIA APU Porting

m-AIA Discontinuous Galerkin Method

Optimization A:

- Looping over nodes
- Fusion of some compute kernels

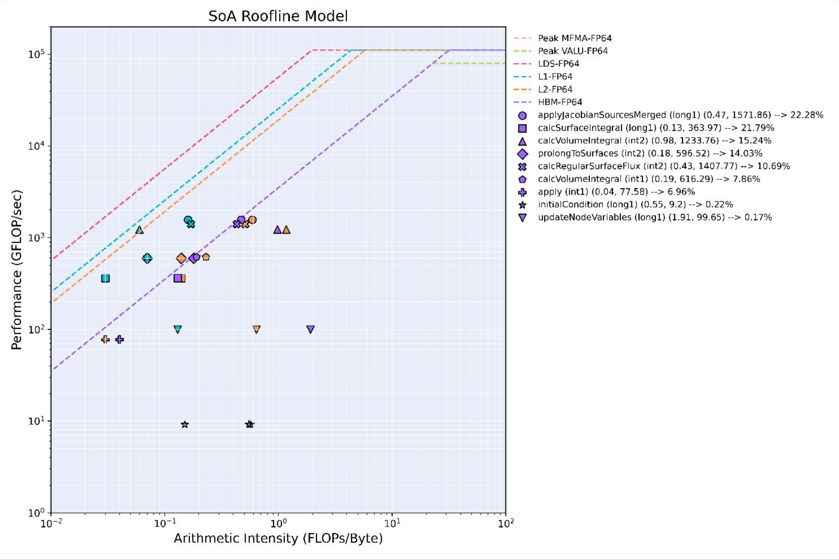


Update of m-AIA APU Porting

m-AIA Discontinuous Galerkin Method

Optimization B

- Changing from AoS to SoA memory layout
- Different memory layouts possible for higher-dimensional data arrays, independent for each data array
- Complex compute kernels with different memory access patterns per Cartesian direction
- Lowest performance: initialization kernels, BC of benchmark case



Summary & Conclusions

- **GPU/APU porting requires substantial effort**
- PSTL seems to be well suited for APU/GPU porting, optimal performance on Hunter may require intermediate HIP kernels (until compiler provides similar performance for PSTL)
- Further optimization of m-AIA is necessary, e.g. in initial and boundary conditions and MPI communication
- Regular meetings with HLRS and AMD specialists are and will be extremely valuable and are planned within the next months

Summary & Conclusions

- **GPU/APU porting requires substantial effort**
- PSTL seems to be well suited for APU/GPU porting, optimal performance on Hunter may require intermediate HIP kernels (until compiler provides similar performance for PSTL)
- Further optimization of m-AIA is necessary, e.g. in initial and boundary conditions and MPI communication
- Regular meetings with HLRS and AMD specialists are and will be extremely valuable and are planned within the next months

Thanks for your attention!

Acknowledgements:

AMD

EXCELLERAT P2

HLRS