Optimizing SOD2D for LUMI

Mohammad Umair† and **Panagiotis-Eleftherios Eleftherakis**§

†FLOW, Engineering Mechanics, KTH Royal Institute of Technology, Sweden
§ Microprocessors and Digital Systems Laboratory, National Technical University of Athens, Greece

Hackathon: Optimizing for AMD GPUs 2024
14-18 October 2024, Brussels (Belgium)

SOD2D: Spectral high-Order coDe 2 solve partial Differential equations

A new Continuous Galerkin High-Order Spectral Element Method (CG-SEM) code designed to perform numerical simulations of both turbulent compressible and incompressible flows.

Developed with the aim to target large scale Computational Fluid Dynamics simulations for engineering applications by exploiting the capabilities of the leading-edge extreme-scale HPC architectures.

Git repository: https://gitlab.com/bsc_sod2d/sod2d_gitlab/ © MIT license

SOD2D: Spectral high-Order coDe 2 solve partial Differential equations

A new Continuous Galerkin High-Order Spectral Element Method (CG-SEM) code designed to perform numerical simulations of both turbulent compressible and incompressible flows.

Developed with the aim to target large scale Computational Fluid Dynamics simulations for engineering applications by exploiting the capabilities of the leading-edge extreme-scale HPC architectures.

Developed at Barcelona Supercomputing Center – Centro Nacional de Supercomputación


Oriol LEHMKUHL
Leading Researcher


Jordi MUELA CASTRO
Recognized Researcher

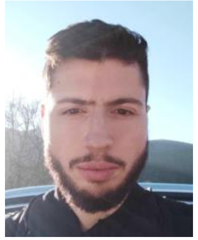

Lucas GASPARINO
First Stage Researcher

Optimization Team at NTUA


Sotirios Xydis
Assistant Professor


G. Anagnostopoulos
PhD Student


K. Iliakis
Postdoc Researcher


P.-E. Eleftherakis
PhD Student

Users & Developers at KTH

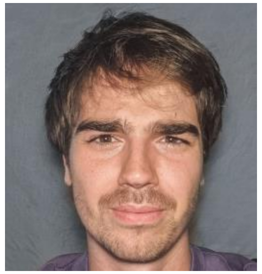

Mohammad UMAIR
Postdoctoral Researcher


Ricardo VINUESA
Principal Investigator


Fran. ALCÁNTARA-ÁVILA
Postdoctoral Researcher


Marcial SANCHIS
PhD Student

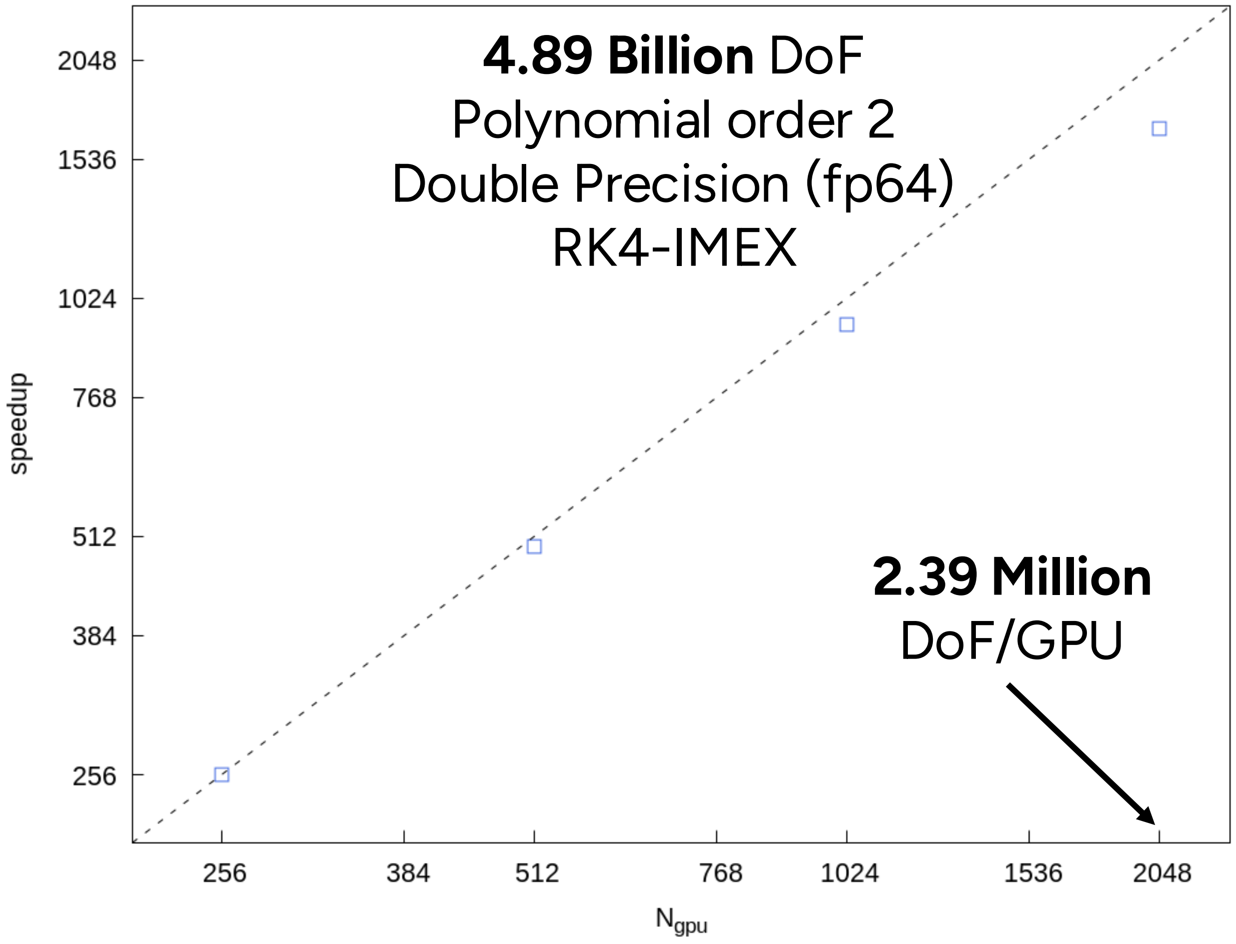

Cristiano PIMENTA
PhD Student (Volvo & KTH)

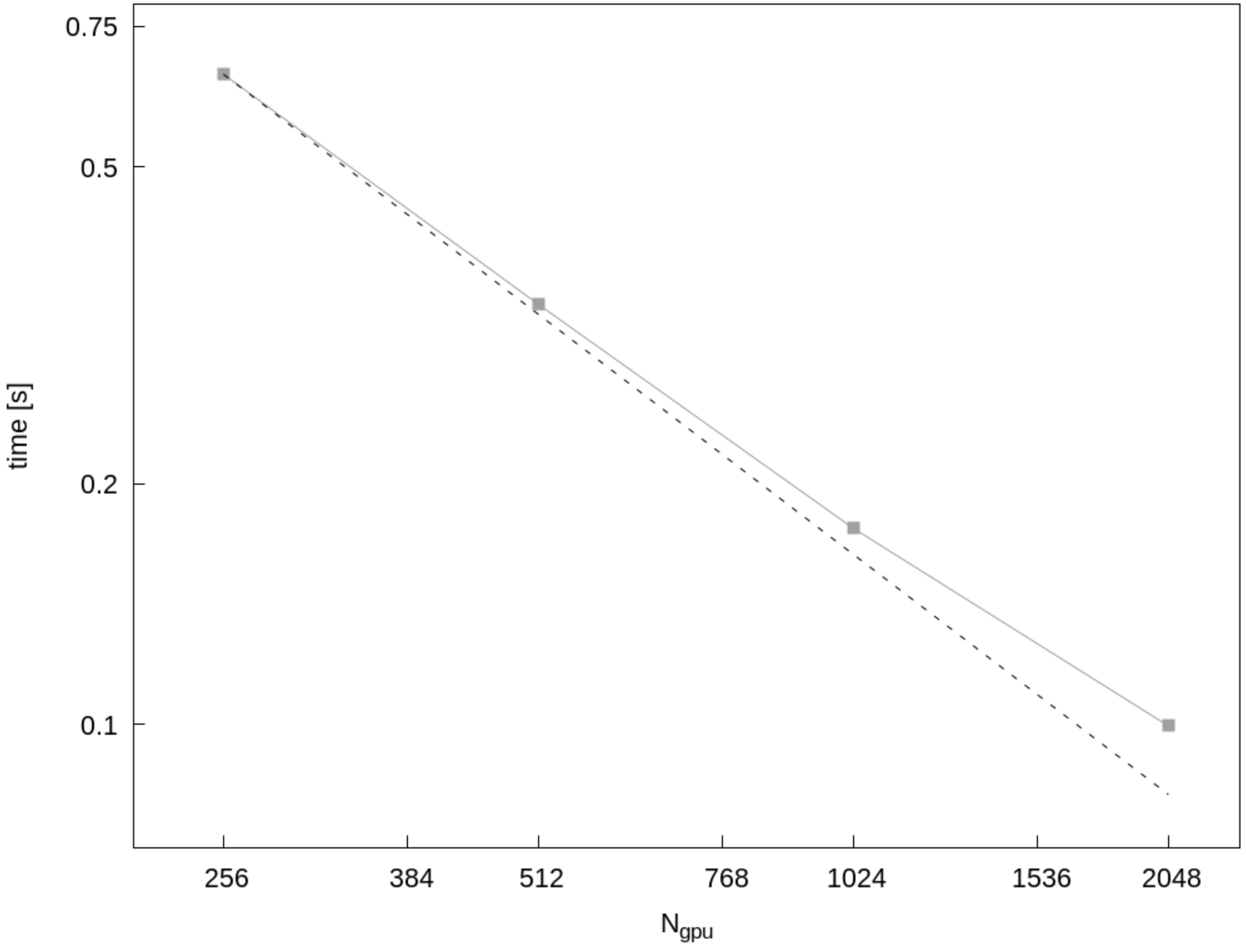

Pol SUÁREZ
PhD Student

Git repository: https://gitlab.com/bsc_sod2d/sod2d_gitlab/ © MIT license

SOD2D Scaling

NASA High-Lift Common Research Model ***(CRM-HL)***





Carried out in **MareNostrum 5 ACC** (Accelerated Partition)

NVIDIA **Hopper H100** 64GB HBM2

Source: Development team at Barcelona Supercomputing Center (Centro Nacional de Supercomputación)

SOD2D LUMI Porting and Optimization Project

SOD2D LUMI Porting and Optimization Project

NVIDIA → AMD

KTH

Mohammad UMAIR
Postdoctoral Researcher

Ricardo VINUESA
Principal Investigator

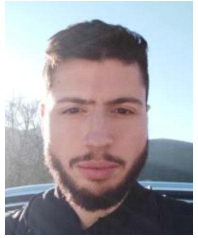
Fran. ALCÁNTARA-ÁVILA
Postdoctoral Researcher

NTUA

Sotirios Xydis
Assistant Professor

G. Anagnostopoulos
PhD Student

K. Iliakis
Postdoc Researcher



P.-E. Eleftherakis
PhD Student

With help from Jing Gong and Jonathan Vincent from PDC Center for High Performance Computing, and Jean-Yves Vet from Hewlett Packard Enterprise (HPE).

SOD2D LUMI Porting and Optimization Project

NVIDIA → AMD

KTH

Mohammad UMAIR
Postdoctoral Researcher

Ricardo VINUESA
Principal Investigator

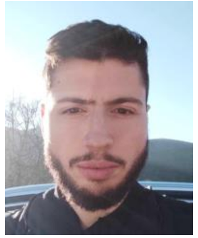
Fran. ALCÁNTARA-ÁVILA
Postdoctoral Researcher

NTUA

Sotirios Xydis
Assistant Professor

G. Anagnostopoulos
PhD Student

K. Iliakis
Postdoc Researcher



P.-E. Eleftherakis
PhD Student

With help from Jing Gong and Jonathan Vincent from PDC Center for High Performance Computing, and Jean-Yves Vet from Hewlett Packard Enterprise (HPE).

SOD2D Auto-Tuning Acceleration Methodology

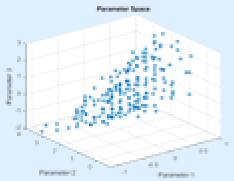
Blackbox exploration

Step 1

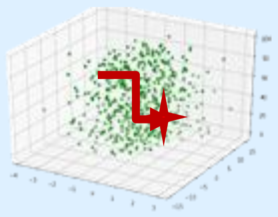
SOD2D
⇩
Profiling
⇩
Hotspots
⇩
Define space
Hotspot
#openacc
pragmas

Step 2

SOD2D



⇩ ⇩
Hyperparameter Search
⇩



Variant 1 (Coupled Configurations)

Configuration ⇨ Hotspots

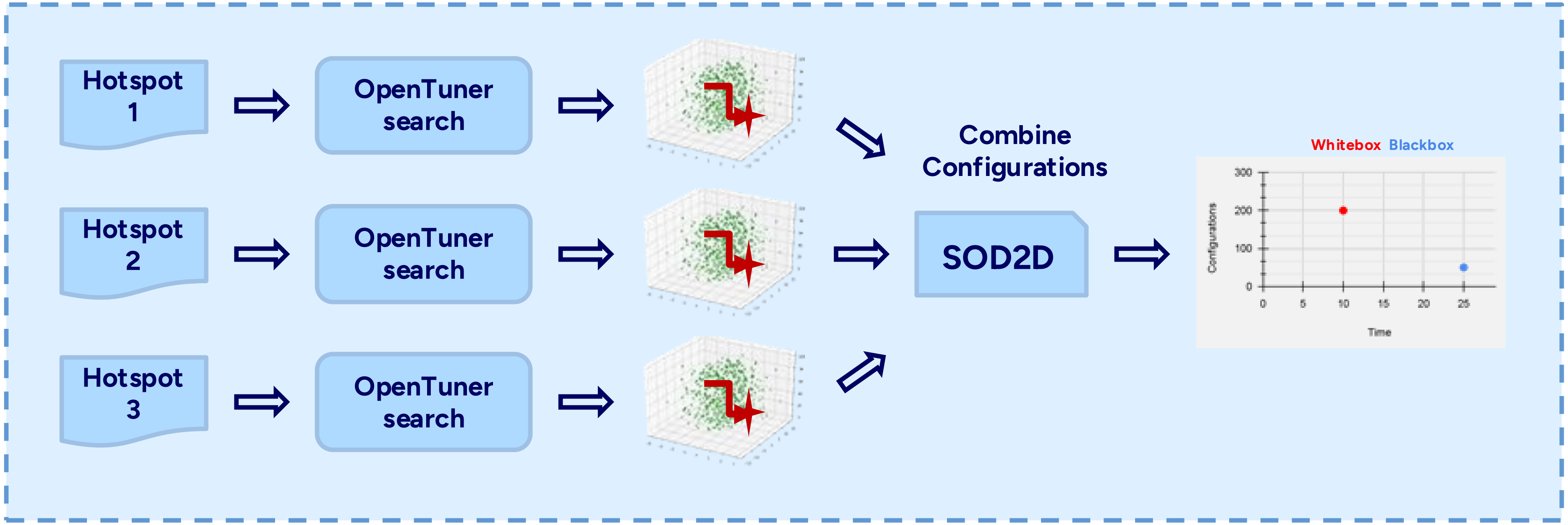
Variant 2 (Decoupled Configurations)

Configuration 1	Configuration 2	Configuration 3	Configuration 4
⇩	⇩	⇩	⇩
Hotspot 1	Hotspot 2	Hotspot 3	Hotspot 4

https://zenodo.org/records/13834895

SOD2D Auto-Tuning Acceleration Methodology

Whitebox Exploration



- Orthogonal optimization searches
- Avoid end-to-end SOD2D simulation

⇩

- Reduced simulation time per config
- Explore more configurations

https://zenodo.org/records/13834895

SOD2D Auto-Tuning Acceleration Results

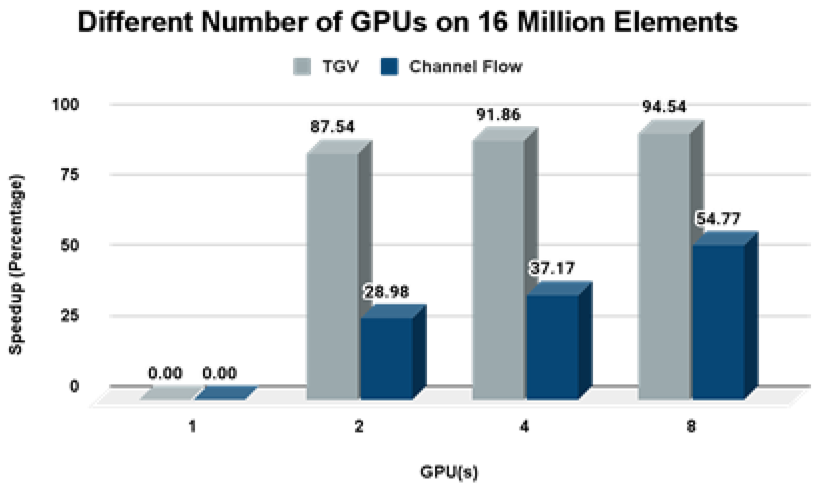
Channel Flow Tuning Results

Case	Performance Gain %	No. GPUs
4m Nodes	7.3%	1
	18.9%	2
8m Nodes	8.4%	1
	19.4%	2

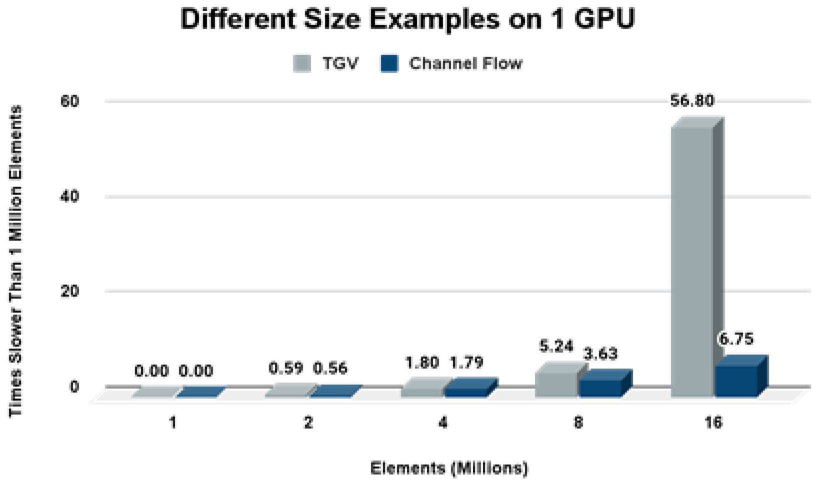
Taylor Green Vortex Tuning Results

Case	Performance Gain %	No. GPUs
KTH Provided Example	10%	1
	15%	2
	~600%	4
	65%	8

Speedup on 4 GPUs Outperforms SOD2D on 8 GPUs ⟹ **Less Resources More Performance**



Speedup Obtained by Multi-GPU System depends on Example Case



~4M Elements Optimal Example Size per GPU

All experiments run in the [Berzelius cluster, NSC]

The work regarding TGV has been a collaboration with KTH and has been accepted as a paper in 2024 Date Conference

https://www.refmap.eu/post/refmap-s-scientific-paper-at-the-date-2024-conference

https://zenodo.org/records/13834895

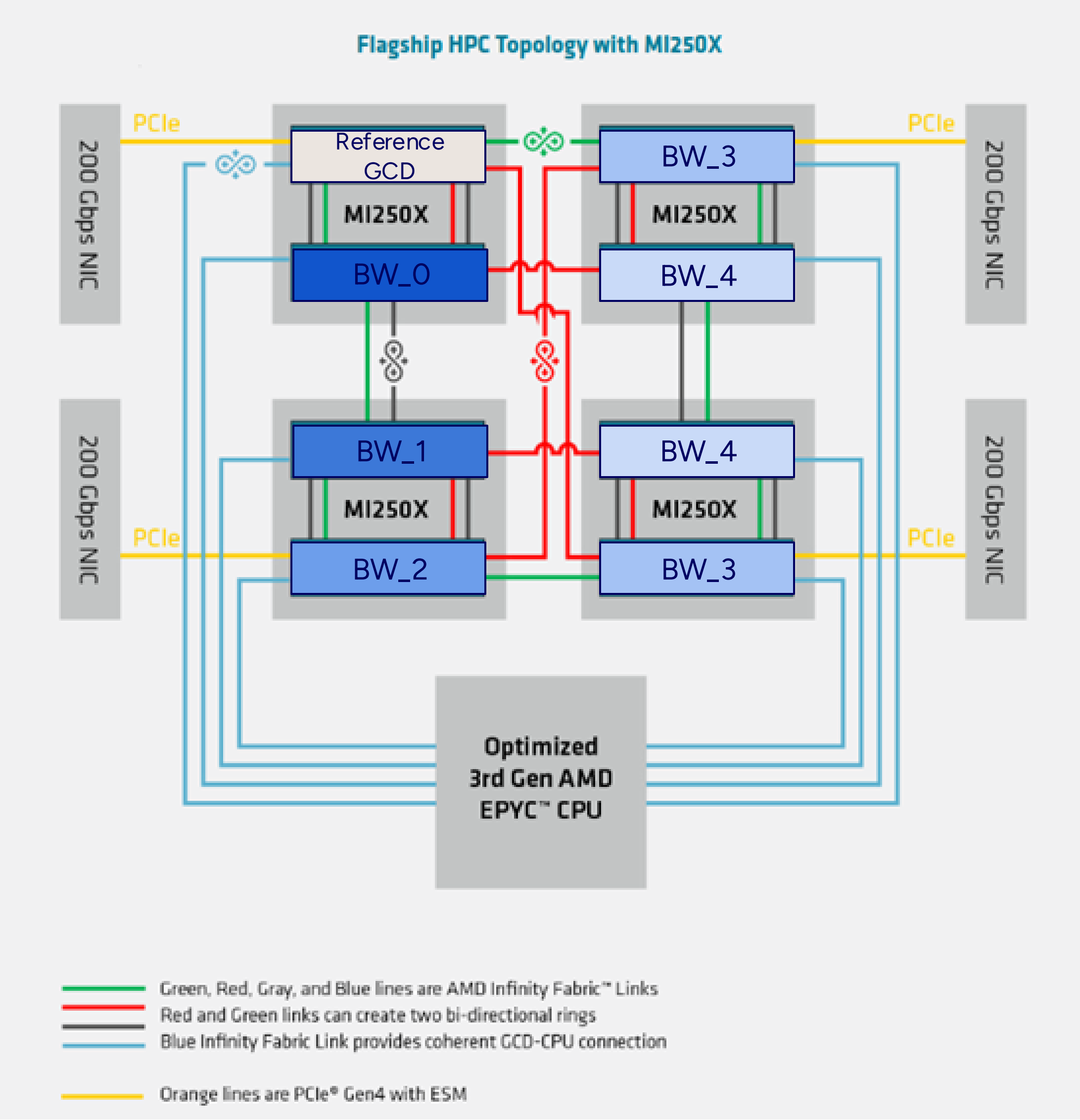
Input Partitioning and Placement Optimization

Stage 0: AMD GPU Optimization

- Performance comparison with NVIDIA GPUs
- OpenACC usage modifications
- Miscellaneous code modifications
- Scale-out performance profiling

Stage 1: Target Topology Microbenchmarking

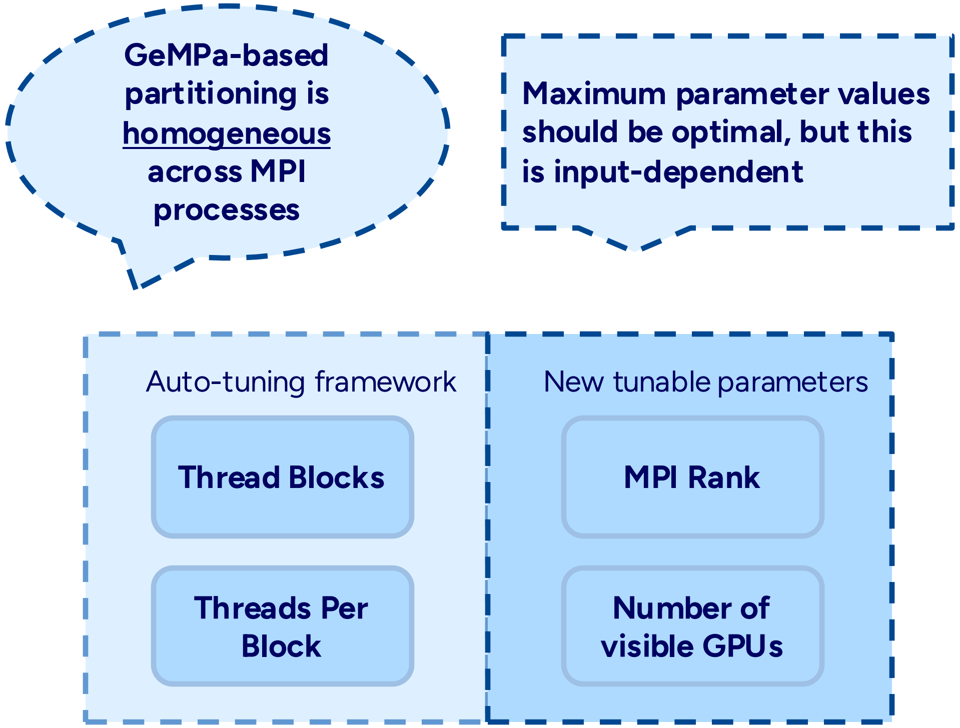
- 7 CPU cores correspond to 1 GPU (GCD) on LUMI
- Non-Uniform GCD communication:
  - **Intra-package:** 4 links
  - **Inter-package:** Either 2 or 1 link
- **Up to 5 different** inter-GCD **bandwidths**.
- **GCD-to-GCD communication microbenchmark** to quantify the above



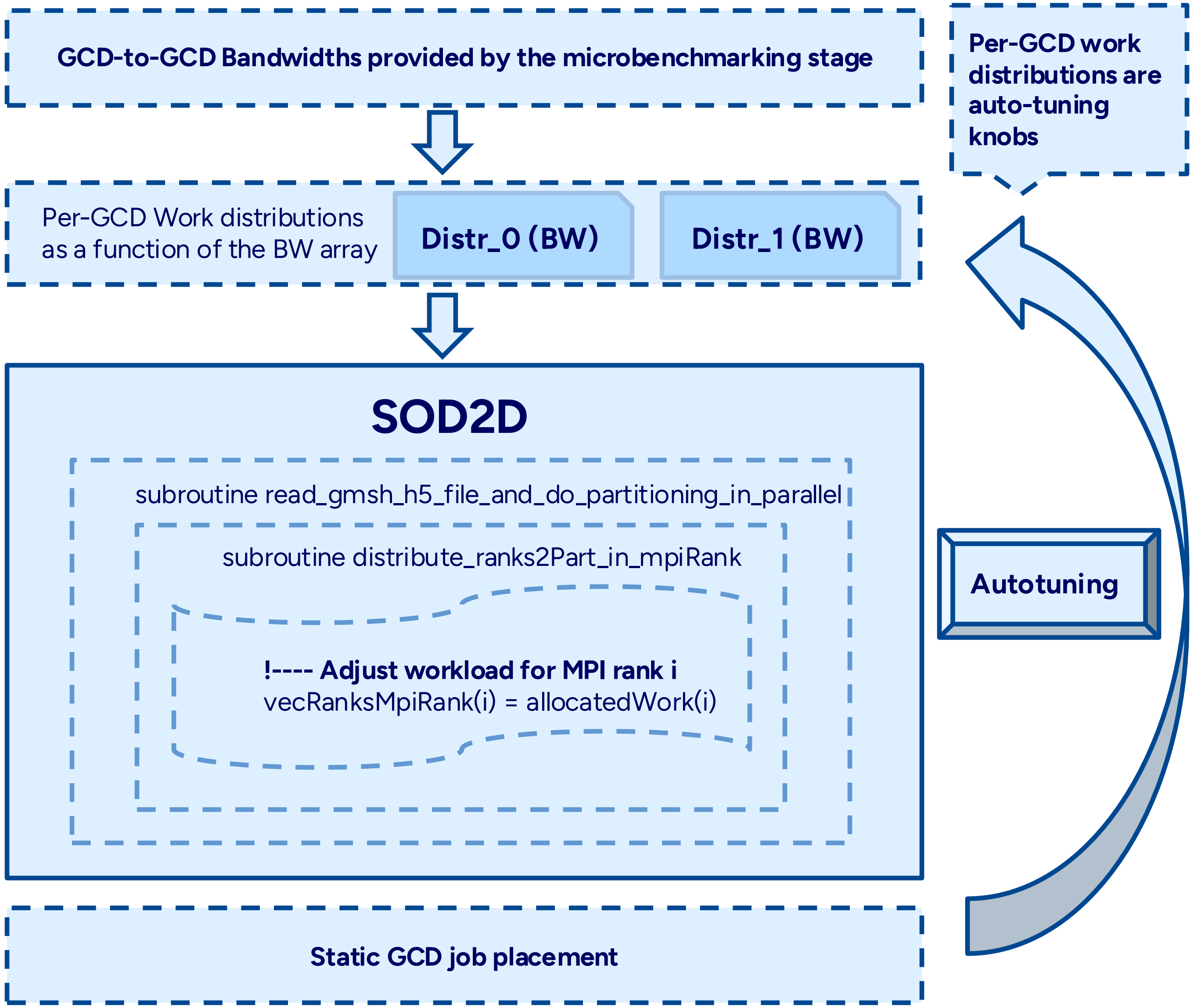
AMD CDNA2 Whitepaper

Input Partitioning and Placement Optimization

Stage 2: Homogeneous Partitioning

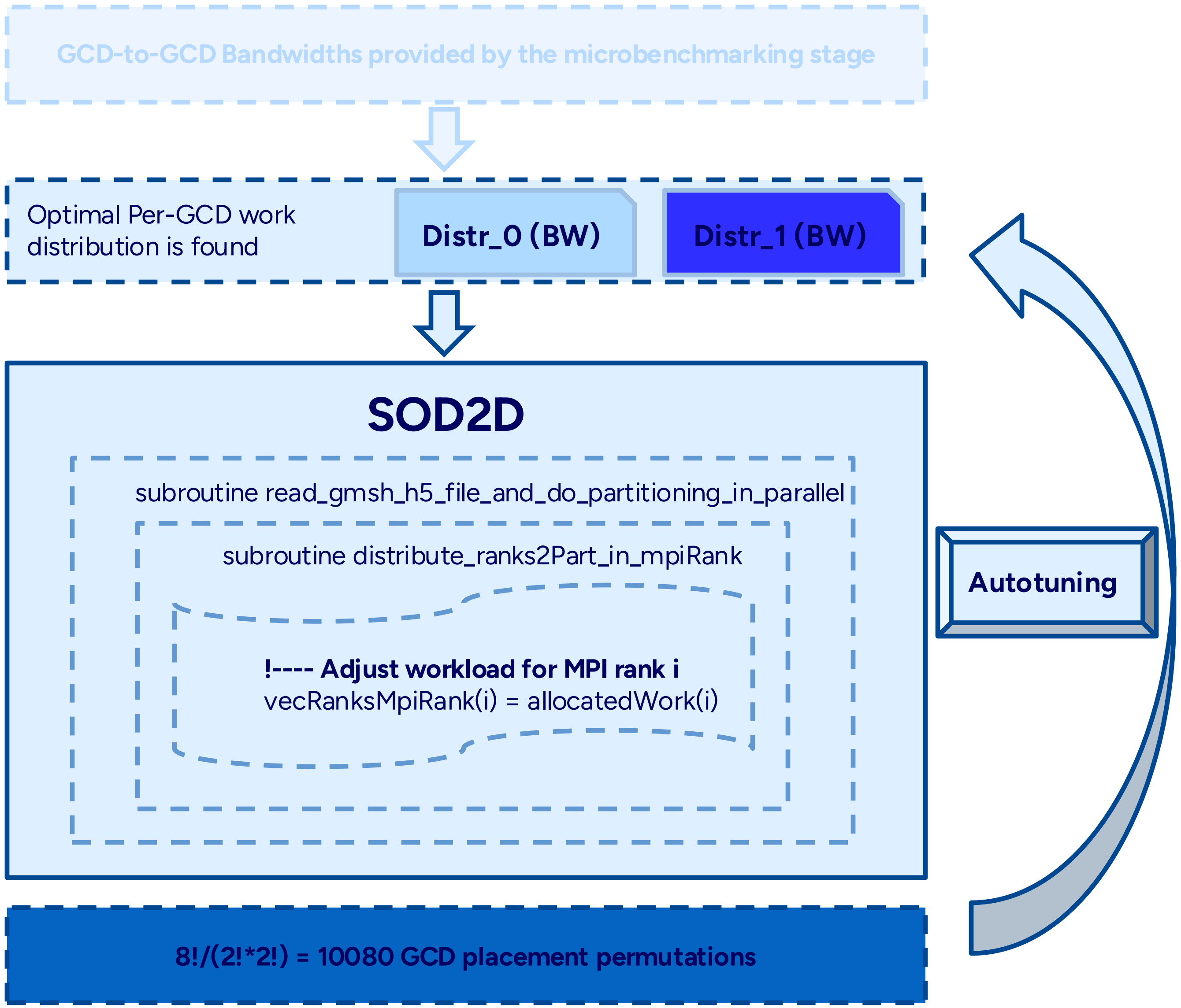


Stage 3: Optimal Heterogeneous Partitioning



Input Partitioning and Placement Optimization

Stage 4: Optimal Heterogeneous Placement

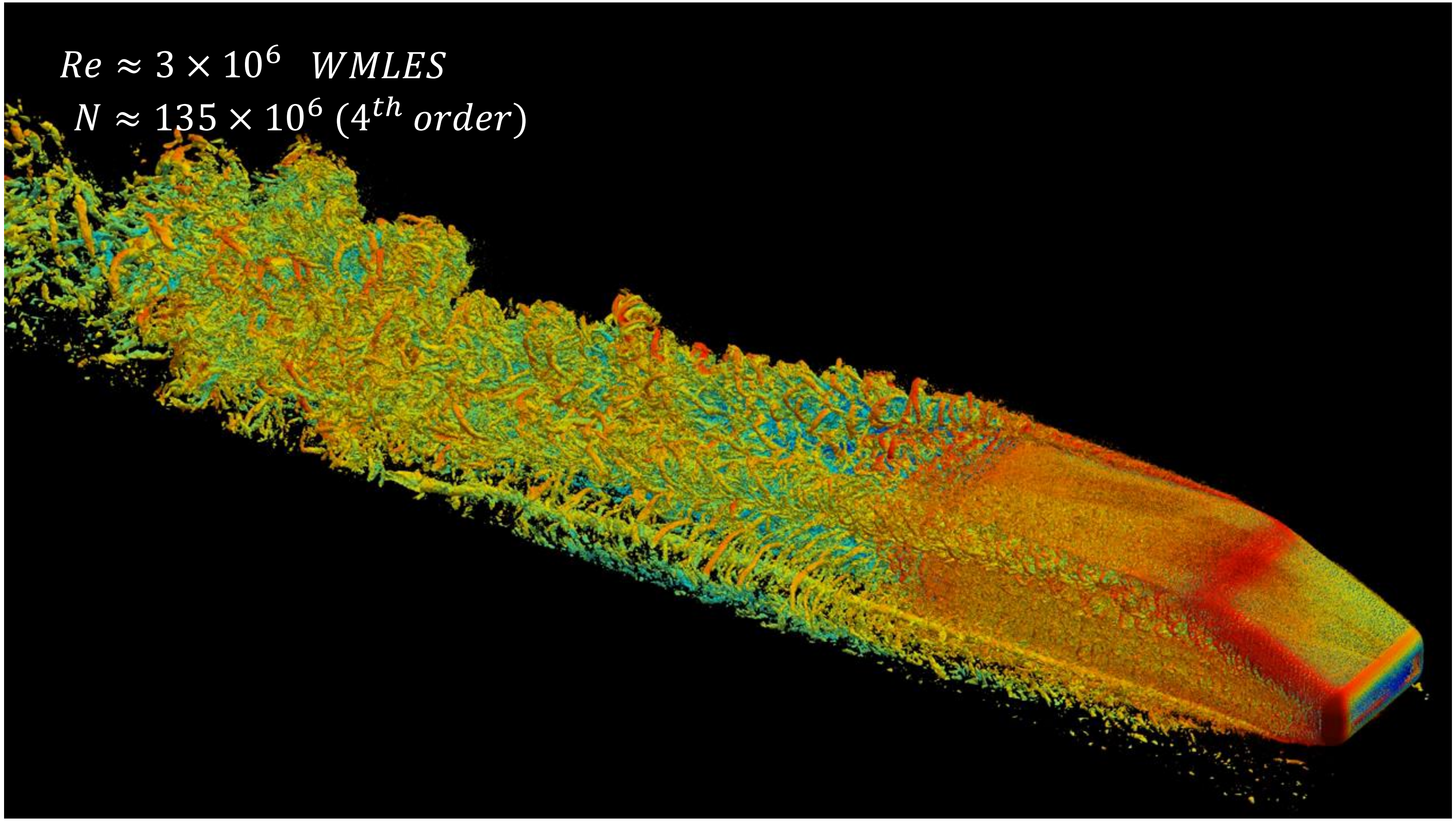


Miscellaneous code optimizations

- Performing **GPU-memory to GPU-memory** MPI operations directly
- Identification of platform-specific **shared memory bottlenecks** and shared memory **tuning**
- **Indirect memory** access optimization
- Loop **tiling auto-tuning**
- Asynchronous **data** transfers and **compute overlap**
- **Kernel fusion**

Windsor Body

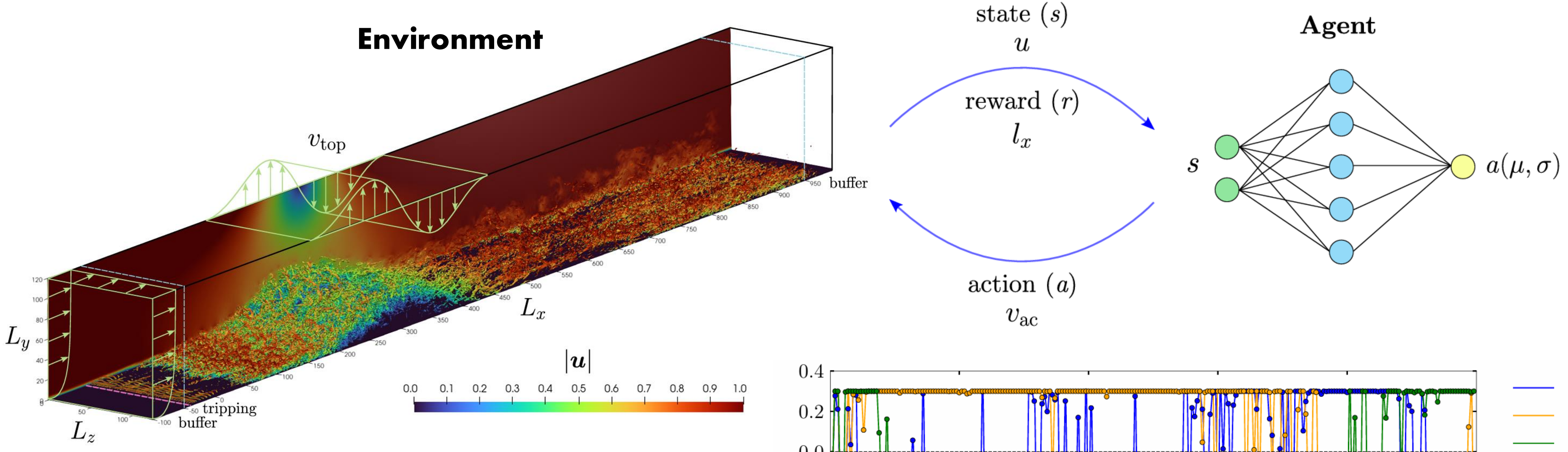
© Benet Eiximeno Franch (UPC) and **Cristiano Pimenta Silva (Volvo & KTH)**



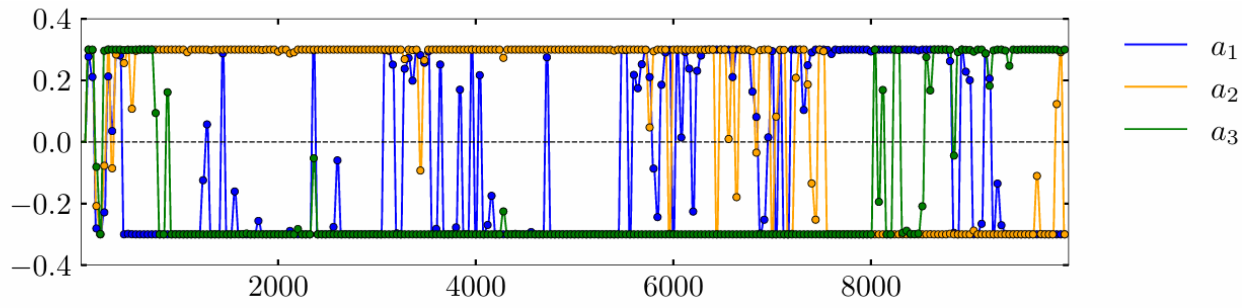
Eiximeno B, Miró A, Rodríguez I, Lehmkuhl O. Toward the Usage of Deep Learning Surrogate Models in Ground Vehicle Aerodynamics. *Mathematics*. 2024; 12(7):998.

Combining CFD and Machine Learning with SOD2D

Separation Control using Deep Reinforcement Learning



	$\overline{l_x}$	$1 - \overline{l_x}/\overline{l_x^*}$
Periodic	129.5 ± 33.4	15.7%
DRL	114.7 ± 6.7	25.3%



Font, B., **Alcántara-Ávila, F.**, Rabault, J., **Vinuesa, R.**, & Lehmkuhl, O. (2024). Active flow control of a turbulent separation bubble through deep reinforcement learning. *Journal of Physics: Conference Series*, *2753*(1), 012022.

Summary

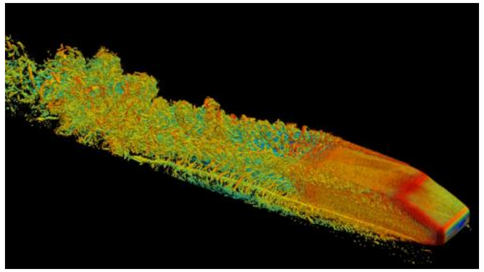
L. Gasparino, F. Spiga, O. Lehmkuhl, **SOD2D: A GPU-enabled Spectral Finite Elements Method for compressible scale-resolving simulations**, Computer Physics Communications, Volume 297, 109067 (2024).

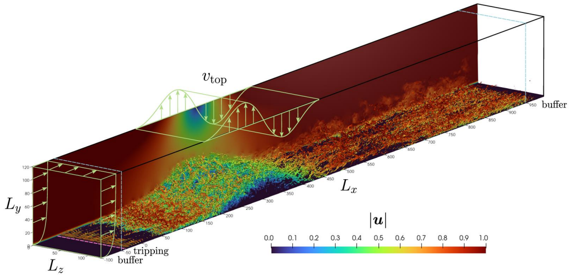
SOD2D: **S**pectral high-**O**rder co**D**e **2** solve partial **D**ifferential equations

- Able to solve compressible and incompressible flows
- Based on Continuous Galerkin Spectral Element Method
- Uses Gauss-Lobatto-Legendre nodes distribution
- RK4 (Compressible) and BDF/EXT3 (Incompressible) time integration
- Fortran90 with MPI for CPUs and OpenACC for GPUs
- HDF5 for Parallel I/O
- SMARTSIM for AI integration

Git repository: https://gitlab.com/bsc_sod2d/sod2d_gitlab/

© MIT license





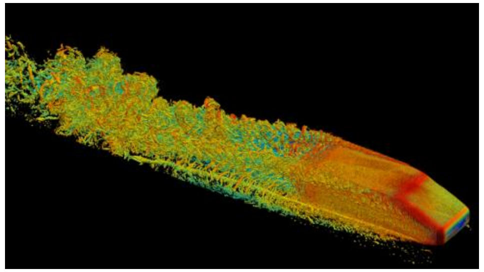
Summary

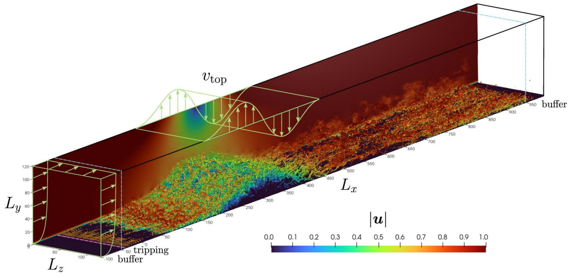
L. Gasparino, F. Spiga, O. Lehmkuhl, **SOD2D: A GPU-enabled Spectral Finite Elements Method for compressible scale-resolving simulations**, Computer Physics Communications, Volume 297, 109067 (2024).

SOD2D: **S**pectral high-**O**rder co**D**e **2** solve partial **D**ifferential equations

- Able to solve compressible and incompressible flows
- Based on Continuous Galerkin Spectral Element Method
- Uses Gauss-Lobatto-Legendre nodes distribution
- RK4 (Compressible) and BDF/EXT3 (Incompressible) time integration
- Fortran90 with MPI for CPUs and OpenACC for GPUs
- HDF5 for Parallel I/O
- SMARTSIM for AI integration

Git repository: https://gitlab.com/bsc_sod2d/sod2d_gitlab/
© MIT license





Summary

L. Gasparino, F. Spiga, O. Lehmkuhl, **SOD2D: A GPU-enabled Spectral Finite Elements Method for compressible scale-resolving simulations**, Computer Physics Communications, Volume 297, 109067 (2024).

SOD2D: **S**pectral high-**O**rder co**D**e **2** solve partial **D**ifferential equations

- Able to solve compressible and incompressible flows
- Based on Continuous Galerkin Spectral Element Method
- Uses Gauss-Lobatto-Legendre nodes distribution
- RK4 (Compressible) and BDF/EXT3 (Incompressible) time integration
- Fortran90 with MPI for CPUs and OpenACC for GPUs
- HDF5 for Parallel I/O
- SMARTSIM for AI integration

Git repository: https://gitlab.com/bsc_sod2d/sod2d_gitlab/

© MIT license