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.

Developed at Barcelona Supercomputing Center – Centro Nacional de Supercomputación

Oriol LEHMKUHL
Leading Researcher

Jordi MUELA CASTRO
Recognized Researcher

Lucas GASPARINO
First Stage Researcher

Users & Developers at KTH

Mohammad UMAIR
Postdoctoral Researcher

Ricardo VINUESA
Principal Investigator

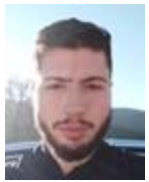
Fran. ALCÁNTARA-ÁVILA
Postdoctoral Researcher

Optimization Team at NTUA

Sotirios Xydis
Assistant Professor

G. Anagnostopoulos
PhD Student

K. Iliakis
Postdoc Researcher



P.-E. Eleftherakis
PhD Student

Marcial SANCHIS
PhD Student

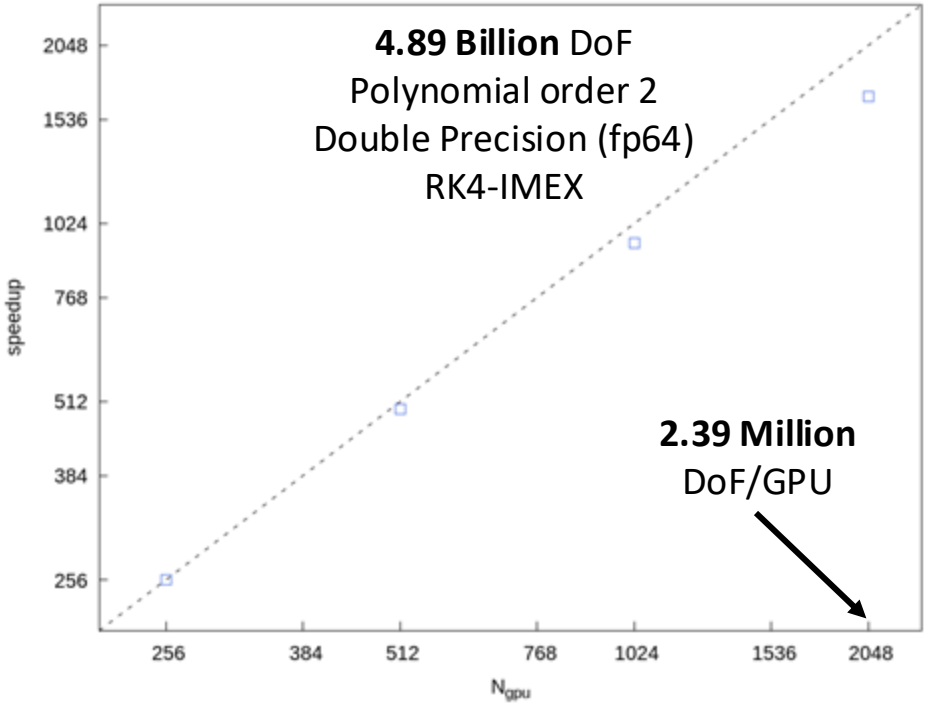
Cristiano PIMENTA
PhD Student (Volvo & KTH)

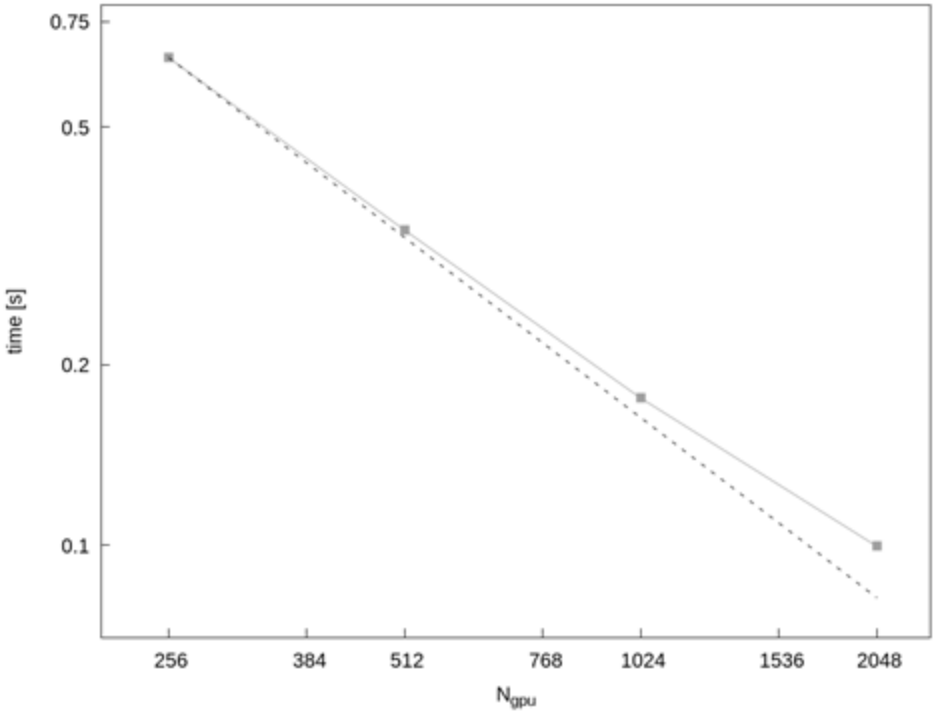
Pol SUÁREZ
PhD Student

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

SOD2D Scaling

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





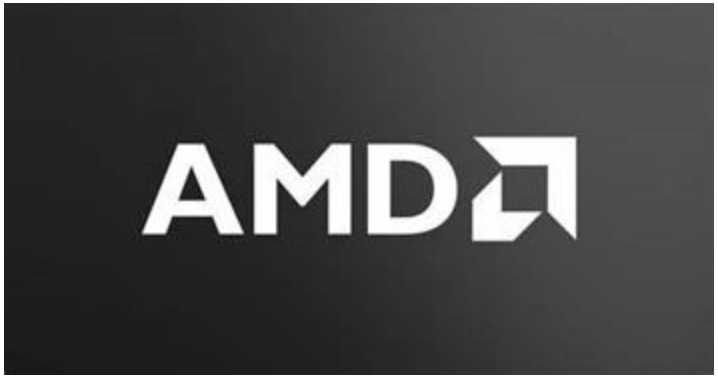
Carried out in **MareNostrum 5 ACC** (Accelerated Partition)

NVIDIA **Hopper H100** 64GB HBM2

Source: Development team at

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

SOD2D LUMI Porting and Optimization Project



KTH

Mohammad UMAIR
Postdoctoral Researcher

Ricardo VINUESA
Principal Investigator

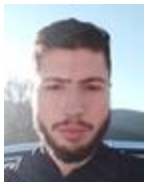
Fran. ALCÁNTARA-ÁVILA
Postdoctoral Researcher

NTUA

Sotirios Xydis
Assistant Professor

G. Anagnostopoulos
PhD Student

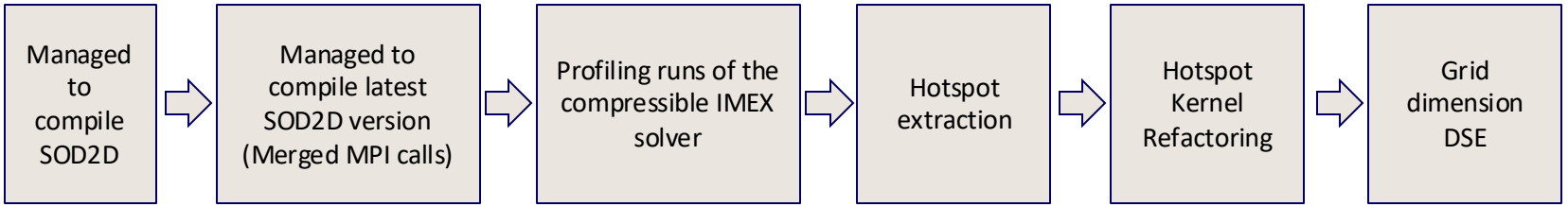
K. Iliakis
Postdoc Researcher



P.-E. Eleftherakis
PhD Student

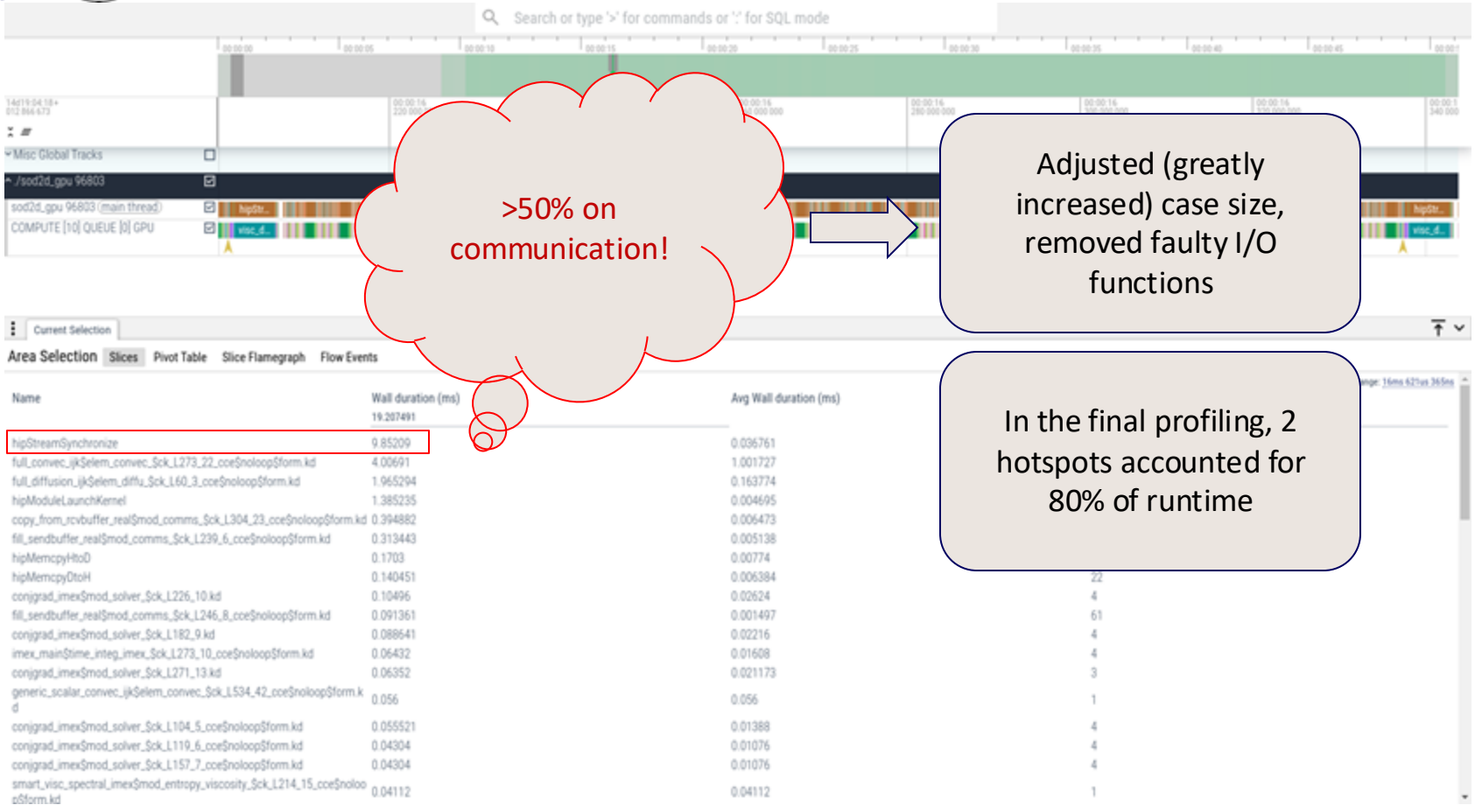
With help from Jing Gong and Jonathan Vincent from PDC Center for High Performance Computing, and Jean-Yves Vet from Hewlett Packard Enterprise (HPE).

Accomplished goals



Indicative Specification	AMD MI250X (LUMI)	NVIDIA H100	NVIDIA DGX A100 (per GPU)
Peak FP64 FLOPS	47.9 TFLOPS	30 TFLOPS	19.5 TFLOPS
Peak FP32 FLOPS	**47.9 TFLOPS**	**60 TFLOPS**	**39 TFLOPS**
Memory Bandwidth	3.28 TB/s	2.0 TB/s	1.6 TB/s
Peak Power	500W (LUMI)	700W	500W
SOD2D norm SLOWDOWN	~6x	1x	2x
SOD2D PPW deterioration	~4.3x	1x	~1.4x

Initial Profiling



>50% on communication!

Adjusted (greatly increased) case size, removed faulty I/O functions

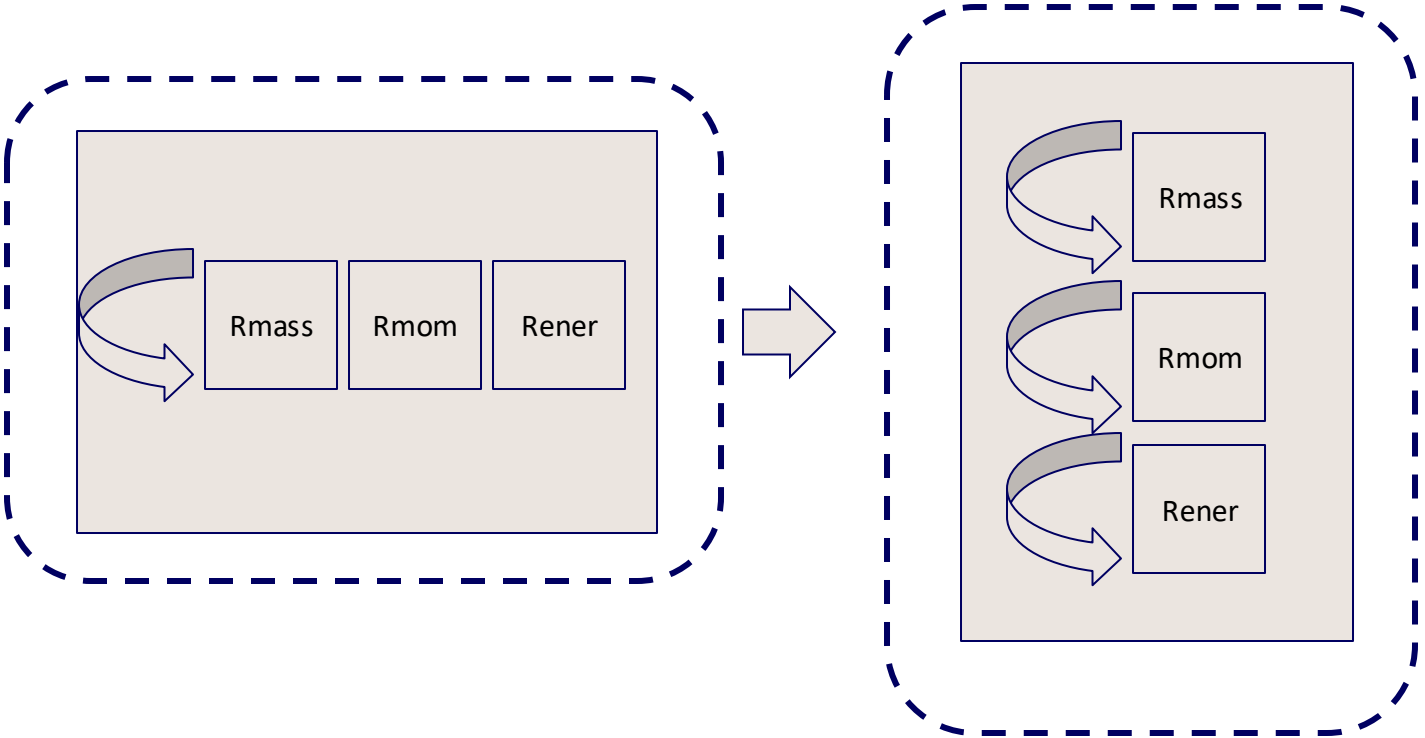
In the final profiling, 2 hotspots accounted for 80% of runtime

Register spilling of the hotspots

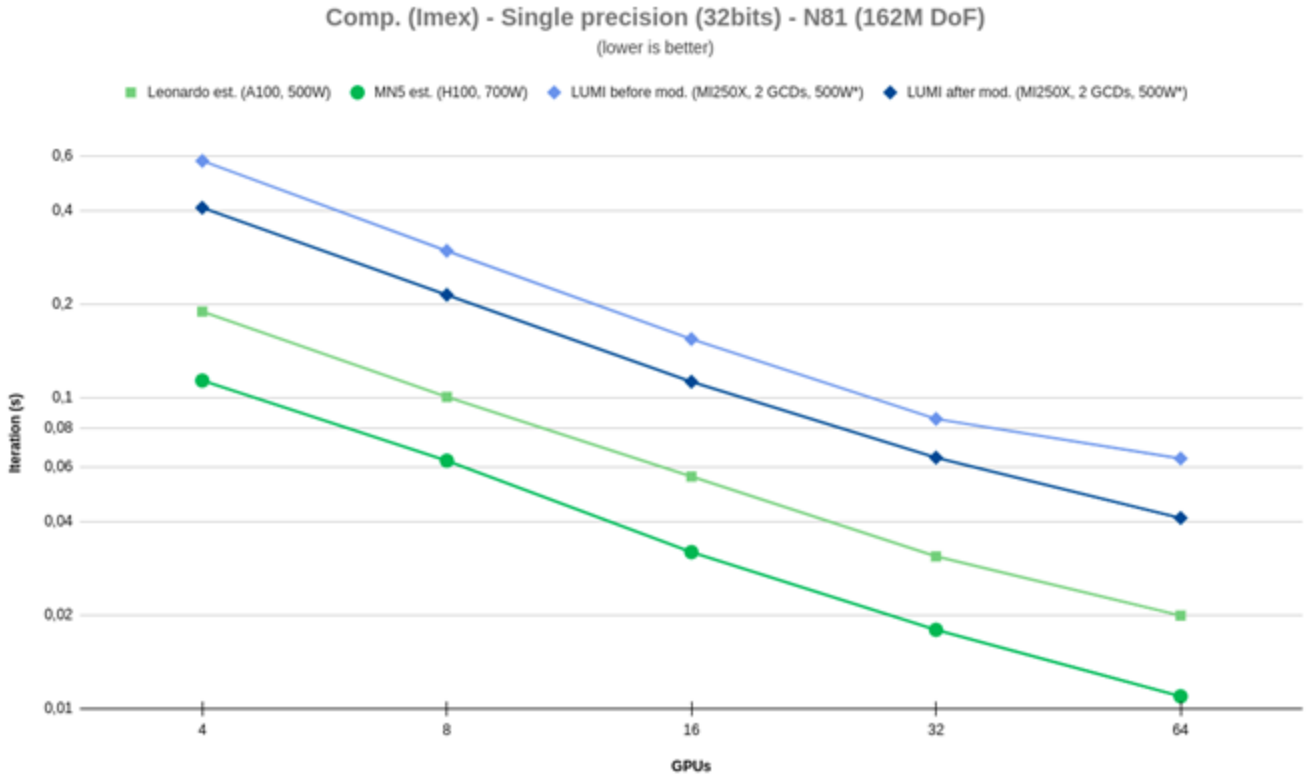
7.1 Wavefront Launch Stats

Metric_ID	Metric	Avg	Min	Max	Unit
7.1.0	Grid Size	16384000.00	16384000.00	16384000.00	Work items
7.1.1	Workgroup Size	256.00	256.00	256.00	Work items
7.1.2	Total Wavefronts	0.00	0.00	0.00	Wavefronts
7.1.3	Saved Wavefronts	0.00	0.00	0.00	Wavefronts
7.1.4	Restored Wavefronts	0.00	0.00	0.00	Wavefronts
7.1.5	VGPRs	128.00	128.00	128.00	Registers
7.1.6	AGPRs	0.00	0.00	0.00	Registers
7.1.7	SGPRs	96.00	96.00	96.00	Registers
7.1.8	LDS Allocation	4608.00	4608.00	4608.00	Bytes
7.1.9	Scratch Allocation	624.00	624.00	624.00	Bytes/workitem

Kernel Code refactoring to reduce spilling



Scaling runs and refactoring speed-up



Speed-up


Avg Iteration Duration and Gangs vs Worker Number
Avg_iteration (10^-3s)
Gangs
Default (128)
Optimal (320)
Avg Iteration Duration (10^-3s)
Number of Gangs
Worker Number
6.8
7.0
7.2
7.4
7.6
50
100
150
200
250
100
200
300
400
500

Recap

- With N=86, full_convec_ijk took 53.60% of the iteration time
- Splitting the kernel in 3 sped up that kernel by deducing register spilling.
- Changing the launch bounds led to an other performance improvment on that kernel
- Now the sum of the 3 split kernels takes about ~33%. which is about the same duration as the next dominant kernel full_diffusion_ijk .Overall iteration improvment is about 30%.
- Final slowdown is ~2x compared to NVIDIA A100

Future work

We will focus more on this kernel

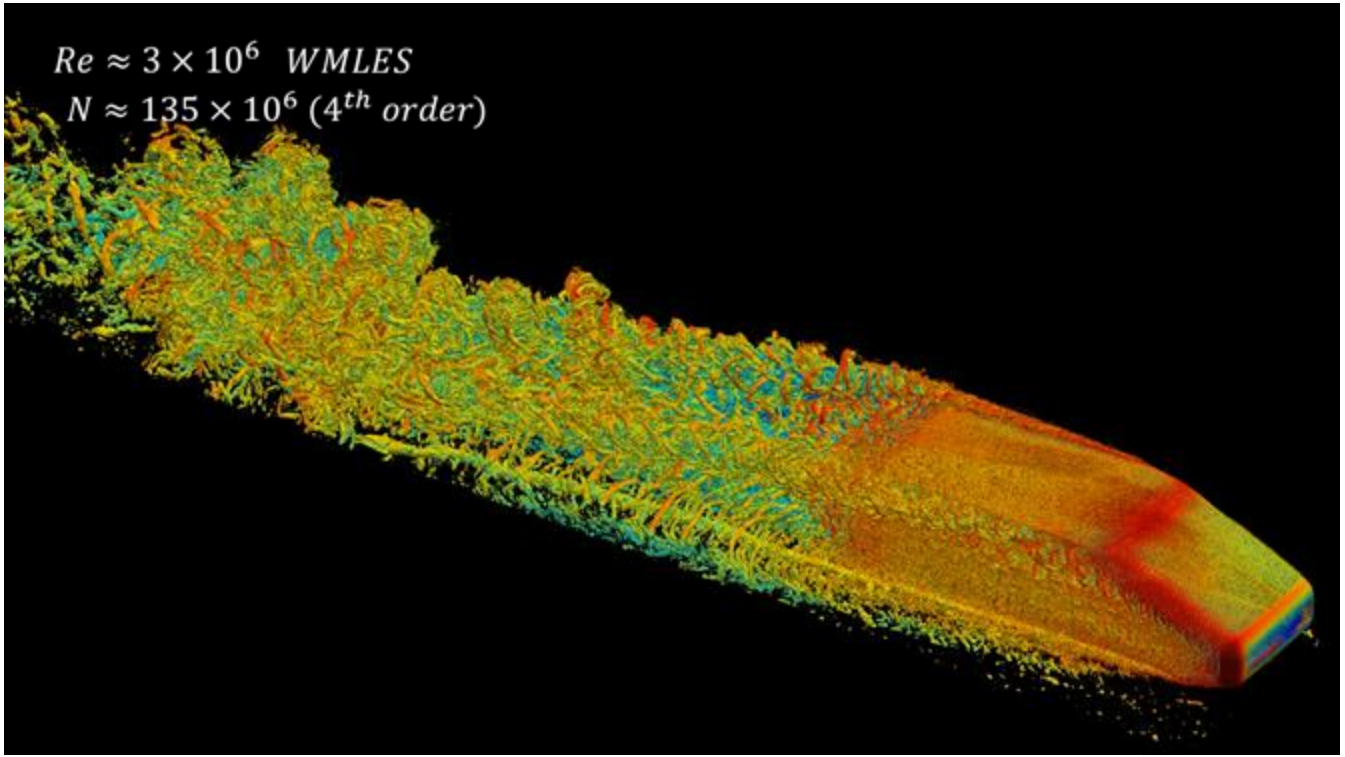
- The idea is to refactor this kernel and then move on to the next big kernel
- There has to be a compromise between performance and readability
- SOD2D is a continuously evolving code
- Many new features are being implemented almost every month

Autotuning

- Full integration of the autotuning framework is required:
  - Changing nvtx profiling calls
  - Scripting changes to adjust to slurm and interface with AMD profilers
- More autotuning runs should be included:
  - Different solvers
  - At scale
  - Different examples

The effort here will reflect in better future

© 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.