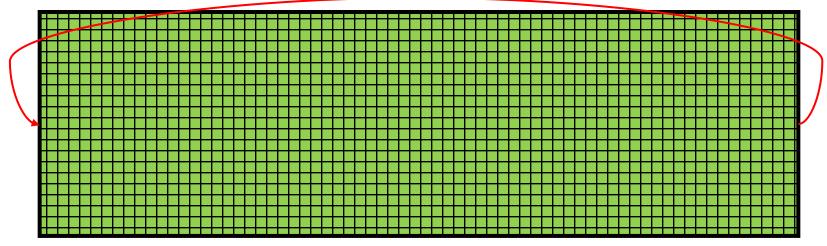
Adventures in CFD and Geometry

> 3DTea (Finally!!) 21-Nov-2024

Picture generated using ChatGPT Prompt – title of the talk

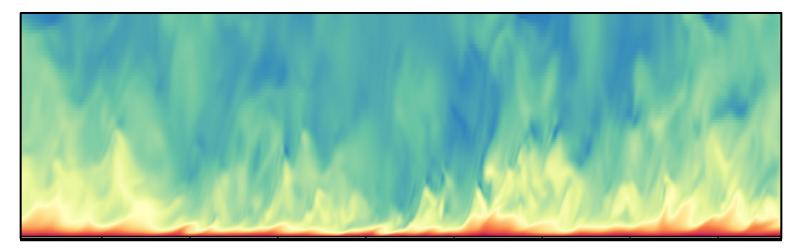


Part 1: Fake It Till You Make It



Standard Flow Configuration

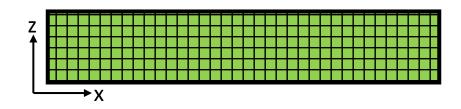
How to best populate the grid cell values, to reach the final state "as quickly as possible"?



Target Final State to Achieve

Computational Cost ~ $N_x \times N_y \times N_z$

Cost is estimated using: $N_{CPU} \times T$ (hours)

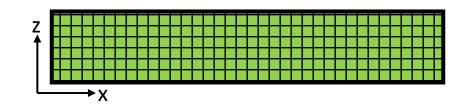


Reference	Flow	Re	Hours	Core type	Cores	Model	Memory (GB
6. Mansour ³⁴	Channel	595	185	CPU	64	IBM SP2	4.096
7. Kim ⁶	Channel	180	62.5	CPU	4	Cray-XMP	56 Mb
8. Jiménez ⁷	Channel	2003	2929.69	CPU	2048	PowerPC 970FX	4096
10. Lee ^{9,10}	Channel	5186	381.47	CPU	52 4288	PowerPC A2	512 TB
11. Alfonsi ¹¹	Channel	200	51.39	CPU/GPU	6 CPU/ 1 GPU	Xeon X5660 Nvidia C-1060	28
12. Alfonsi ¹¹	Channel	400	237.5	CPU/GPU	6 CPU/ 1 GPU	Xeon X5660 Nvidia C-1060	28
13. Alfonsi ¹¹	Channel	600	461.11	CPU/GPU	18 CPU/	Xeon X5660	84

17. Vela-Martín ¹⁶	Channel	2000	507.81	GPU	128	Tesla P-100	2048
18. Vela-Martín ¹⁶	Channel	5303	2734.38	GPU	512	Tesla P-100	8192

Computational Cost ~ $N_x \times N_y \times N_z$

Cost is estimated using: $N_{CPU} \times T$ (hours)



Reference	Flow	Re	Hours	Core type	Cores	Model	Memory (GB)	Region	kWh	Mass (kg)
6. Mansour ³⁴	Channel	595	185	CPU	64	IBM SP2	4.096	California	237.74	51.46
7. Kim ⁶	Channel	180	62.5	CPU	4	Cray-XMP	56 Mb	California	5.01	1.08
8. Jiménez ⁷	Channel	2003	2929.69	CPU	2048	PowerPC 970FX	4096	Spain	1.28×10^{5}	2.18×10^{4}
10. Lee ^{9,10}	Channel	5186	381.47	CPU	52 4288	PowerPC A2	512 TB	Illinois	4.13×10^{6}	$1.1 imes 10^6$
11. Alfonsi ¹¹	Channel	200	51.39	CPU/GPU	6 CPU/ 1 GPU	Xeon X5660 Nvidia C-1060	28	Italy	25.17	8.15
12. Alfonsi ¹¹	Channel	400	237.5	CPU/GPU	6 CPU/ 1 GPU	Xeon X5660 Nvidia C-1060	28	Italy	116.3	37.66
13. Alfonsi ¹¹	Channel	600	461.11	CPU/GPU	18 CPU/	Xeon X5660	84	Italy	677.41	219.37
17. Vela-Martín ¹⁶	Channel	2000	507.81	GPU	128	Tesla P-100	2048	Switzerland	$2.78 imes 10^4$	320.08
18. Vela-Martín ¹⁶	Channel	5303	2734.38	GPU	512	Tesla P-100	8192	Switzerland		6894

Computational Cost ~ $N_x \times N_y \times N_z$

Reference

6. Mansour³⁴

8. Jiménez⁷

10. Lee^{9,10}

11. Alfonsi¹¹

12. Alfonsi¹¹

13. Alfonsi¹¹

7. Kim⁶

Cost is estimated using: $N_{CPU} \times T$ (hours)

Flow

Channel 595

Channel 5186

Channel 600

Channel

Channel

Channel

Re

180

Channel 2003 2929.69

200

400

Hours Core type

CPU

CPU

CPU

CPU

461.11 CPU/GPU 18 CPU/

CPU/GPU 6 CPU/

CPU/GPU 6 CPU/

185

62.5

381.47

51.39

237.5

Cores

64

4

2048

52 4288

1 GPU

1 GPU

Model

IBM SP2

Cray-XMP

PowerPC

970FX

PowerPC A2

Xeon X5660

Nvidia C-1060

Xeon X5660

Nvidia C-1060

Xeon X5660

84

Italy

677.41

219.37

				Corporate clients	Contact Career	FAQ 🌐 Global Eng	glish \checkmark
	Сту	climate R	Q =				
		Calculate		Choose		Pay	
	Your f	-	IS to: Paris (FR), CDC	à, One way, Economy Cl	ass, ca. 400 km, 1 trav	eller	
	CO ₂ a	mount: 0.199 t			200 kg	of CO2	
				Flight fr	om Ams	terdam ·	- Paris
Memory (GB)	Region	kWh	Mass (kg)	_			
4.096 56 Mb 4096	California California Spain	237.74 5.01 1.28×10^{5}	51.46 1.08 $2.18 imes 10^4$	6			
512 TB 28	Illinois Italy	4.13 × 10 ⁶ 25.17	1.1×10^{6} 8.15				
28	Italy	116.3	37.66				

17. Vela-Martín ¹⁶	Channel	2000	507.81	GPU	128	Tesla P-100	2048	Switzerland	$2.78 imes 10^4$	320.08
18. Vela-Martín ¹⁶	Channel	5303	2734.38	GPU	512	Tesla P-100	8192	Switzerland	$5.98 imes 10^5$	6894

Computational Cost ~ $N_x \times N_y \times N_z$

Reference

6. Mansour³⁴

8. Jiménez⁷

10. Lee^{9,10}

11. Alfonsi¹¹

12. Alfonsi¹¹

13. Alfonsi¹¹

7. Kim⁶

Cost is estimated using: $N_{CPU} \times T$ (hours)

Flow

Channel 595

Channel 5186

Channel 600

Channel

Channel

Channel

Re

180

Channel 2003 2929.69

200

400

Hours Core type

CPU

CPU

CPU

CPU

461.11 CPU/GPU 18 CPU/

CPU/GPU 6 CPU/

CPU/GPU 6 CPU/

185

62.5

381.47

51.39

237.5

Cores

64

4

2048

52 4 2 8 8

1 GPU

1 GPU

Model

IBM SP2

Cray-XMP

PowerPC

970FX

PowerPC A2

Xeon X5660

Nvidia C-1060

Xeon X5660

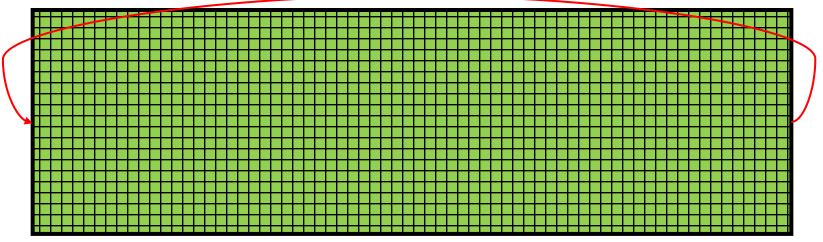
Nvidia C-1060

Xeon X5660

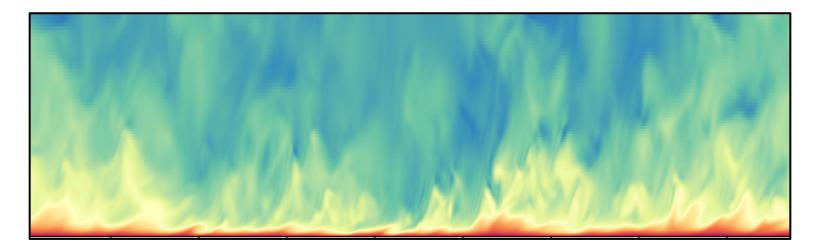
				Corporate clients	Contact Career	FAQ 🌐 Global Engl	ish 🗸
	Сту	climate R	Q =				
		Calculate		Choose		Pay	
	Your f	light:					
	From:	Amsterdam (NL), AN	IS to: Paris (FR), CD0	G, One way, Economy Cla	ass, ca. 400 km, 1 trav	eller	
	CO ₂ a	mount: 0.199 t			200 kg	of CO2	
				Flight fr	om Ams	sterdam -	Paris
Memory (GB)	Region	kWh	Mass (kg)				
4.096 56 Mb 4096 512 TB	California California Spain Illinois	237.74 5.01 1.28×10^{5} 4.13×10^{6}	51.46 1.08 2.18×10^4 1.1×10^6	6			
28	Italy	25.17	8.15				
28 84	Italy Italy	116.3 677.41	37.66 219.37				

17. Vela-Martín ¹⁶	Channel	2000	507.81	GPU	128	Tesla P-100	2048	Switzerland	$2.78 imes 10^4$	320.08
 18. Vela-Martín ¹⁶	Channel	5303	2734.38	GPU	512	Tesla P-100	8192	Switzerland	$5.98 imes 10^5$	6894

Lots of CO2 output for no reason! We better find a fast way to spin up to the final state.



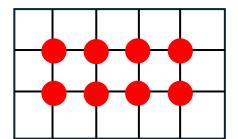
Standard Flow Configuration

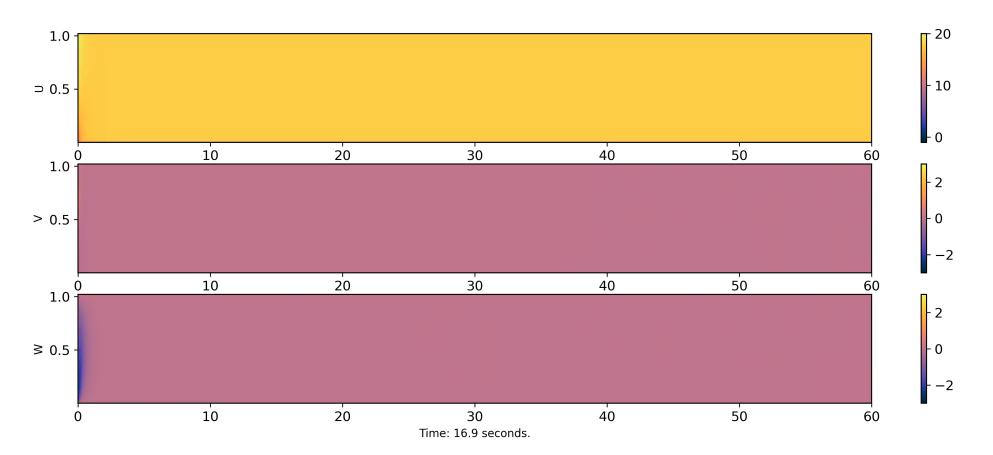


Target Final State to Achieve

Use an existing tool – differently! – No Math Version

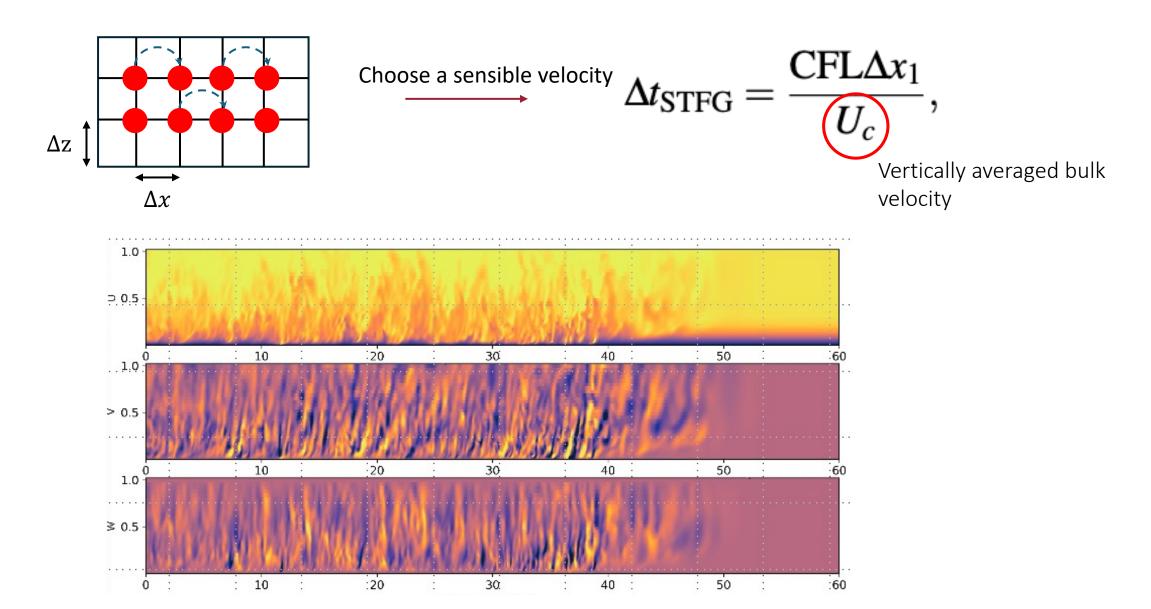
Instead of using the synthetic turbulence generator as an inflow, populate the grid with data!



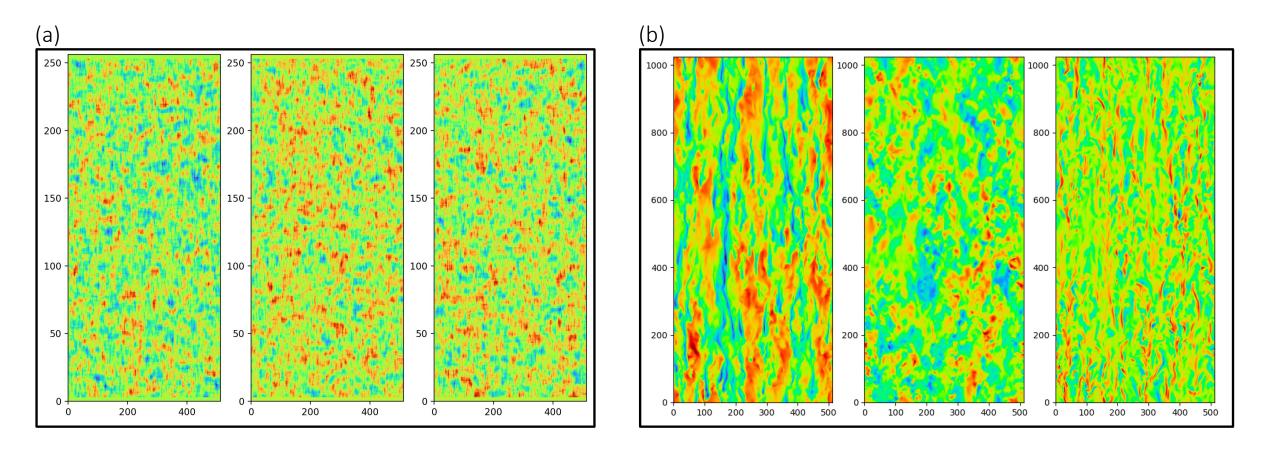


Use an existing tool – differently! – No Math Version

Instead of using the synthetic turbulence generator as an inflow, populate the grid with data!



Can you guess which one is fake and which one is real?



BreatheLab affiliates are NOT allowed to answer/respond!

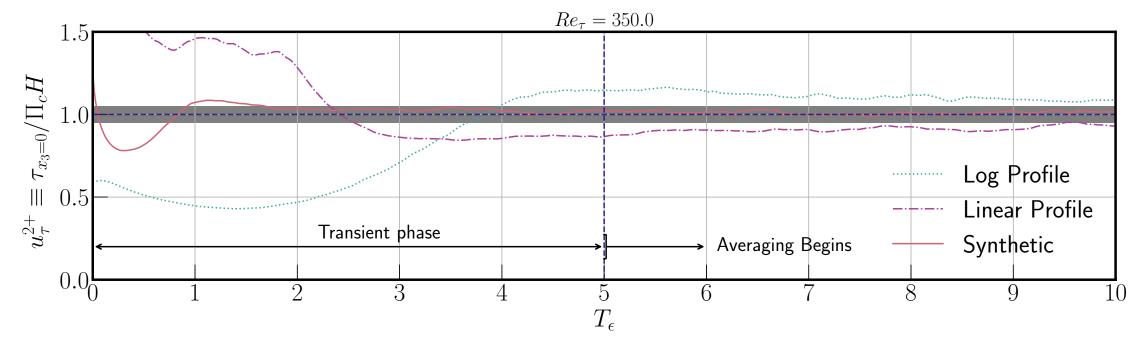
Using artificial turbulence to achieve swift converged turbulence statistics in a pressuredriven channel flow

A. Patil and C. García-Sánchez

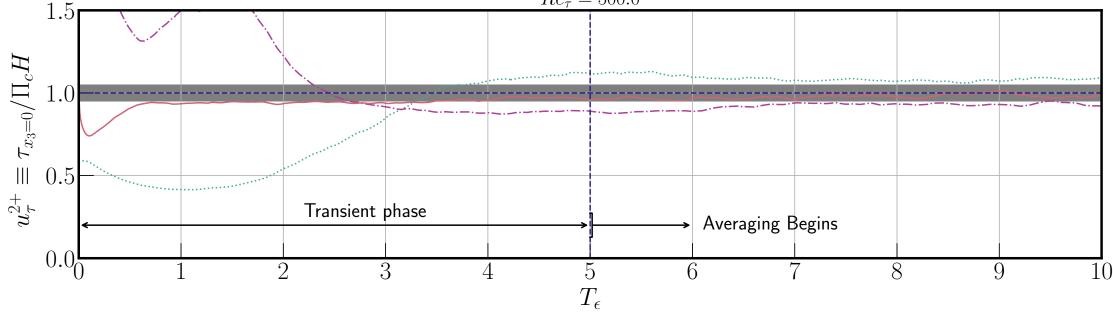
Supplementary Video 2

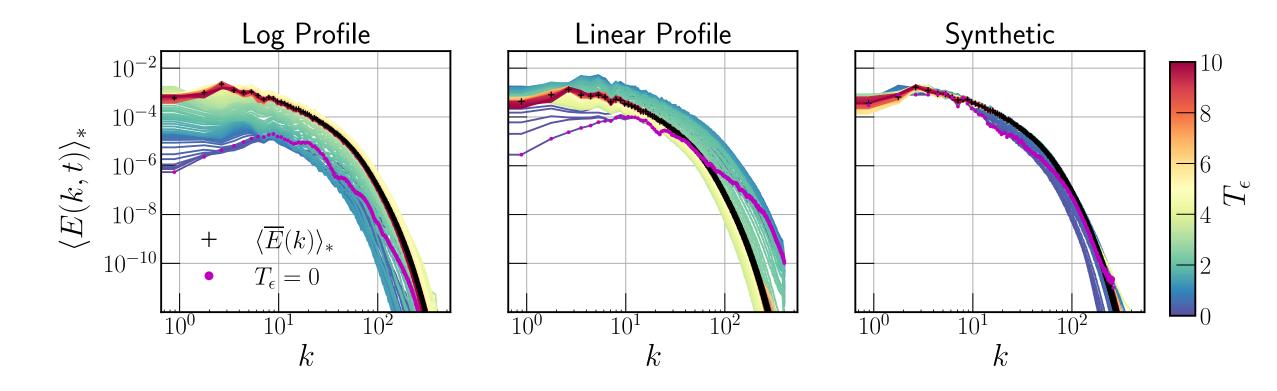
Comparison of transition to turbulence, friction Reynolds number: 500





 $Re_{\tau} = 500.0$





A factor of 7-8 faster compared to conventional methods

End of Part 1

Part 2: GenSDF

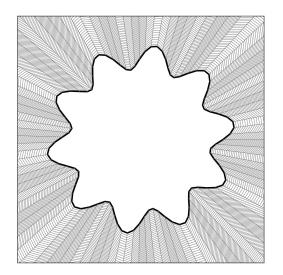
But Why?

Memory hungry trimesh library Slow python code and many more.....

<u>Solution</u>

Modern Fortran + MPI based implementation

- Low memory requirements
- Easy to port to existing CFD solvers in Fortran
- Good excuse to program :)



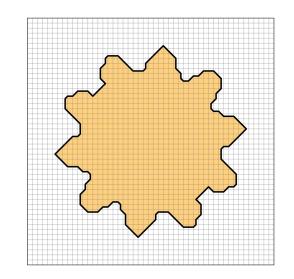
🗋 runall_mpi.sh	Updating paper draft and README.md	last year
🗋 tests.py	Update bounding box in tests.py	last year
🕒 translateSTL.py	code cleanup	last year
🗋 visualiseSDF.py	Code clean up	last year
다 README 화 BSD-3-Clause license		∅ :Ξ

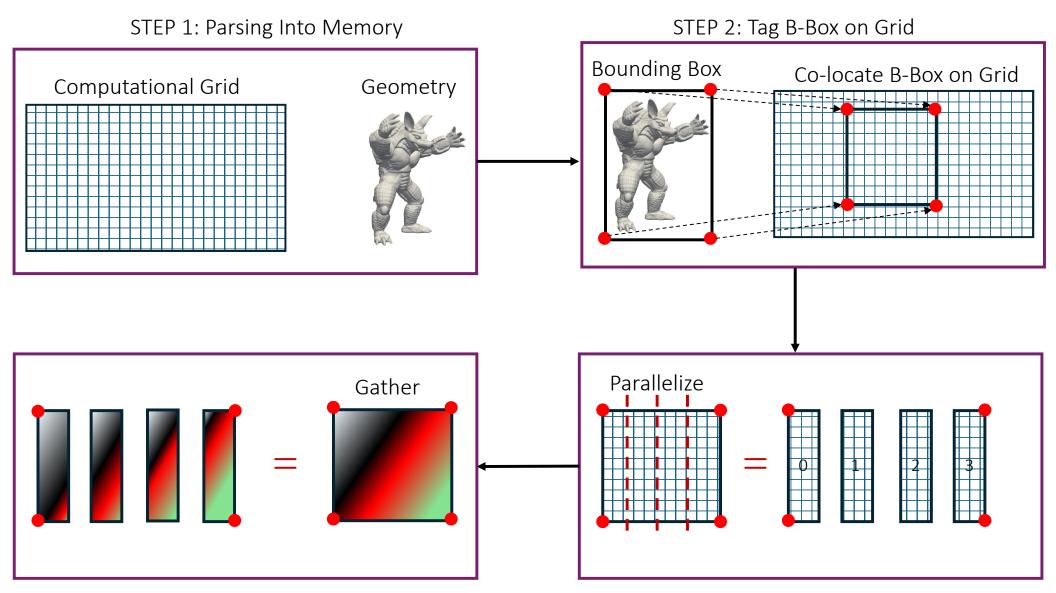


Setup the virtual environment

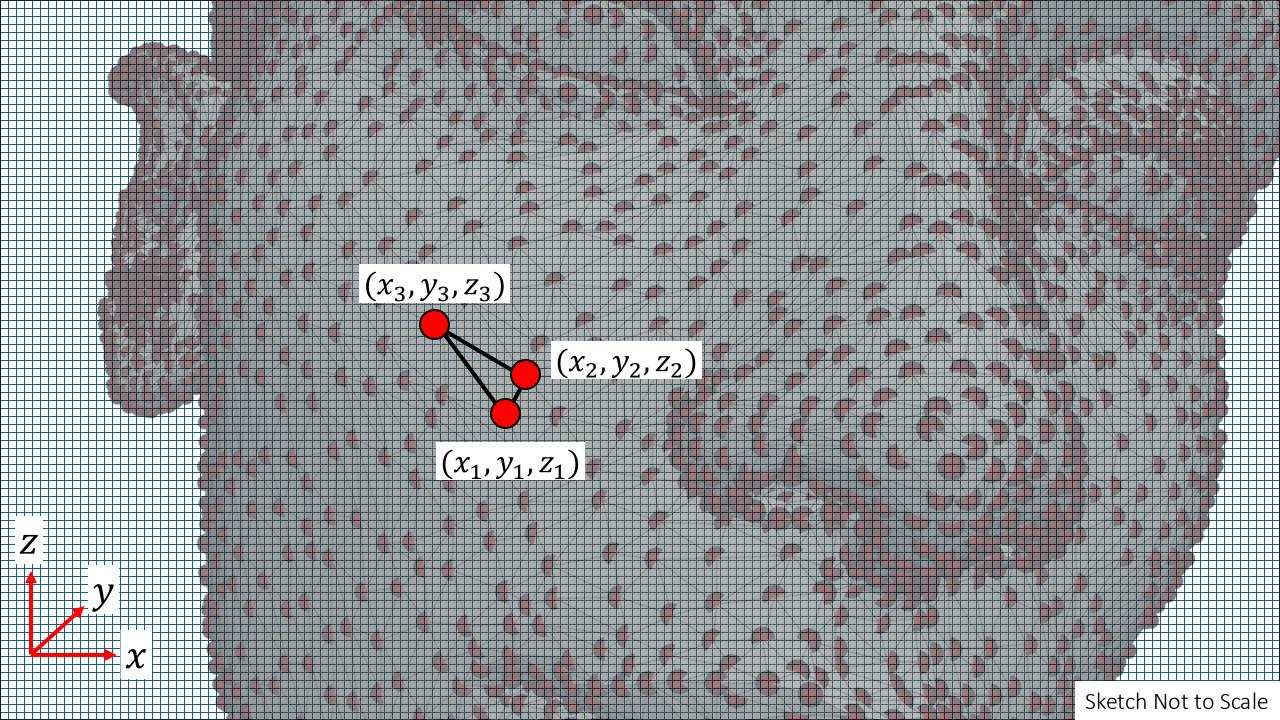
- Install virtual environment [See https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/]
- python3 -m pip install --user virtualenv
- Create the virtual environment for a specific python version .and. say yes to the prompt question python3 -m venv <pathtosave>/stl2sdf
- Install the required libraries
- pip install -r requirements.txt

To deactivate the special python environment use [Note: dont do this if you wish to run the code...] deactivate



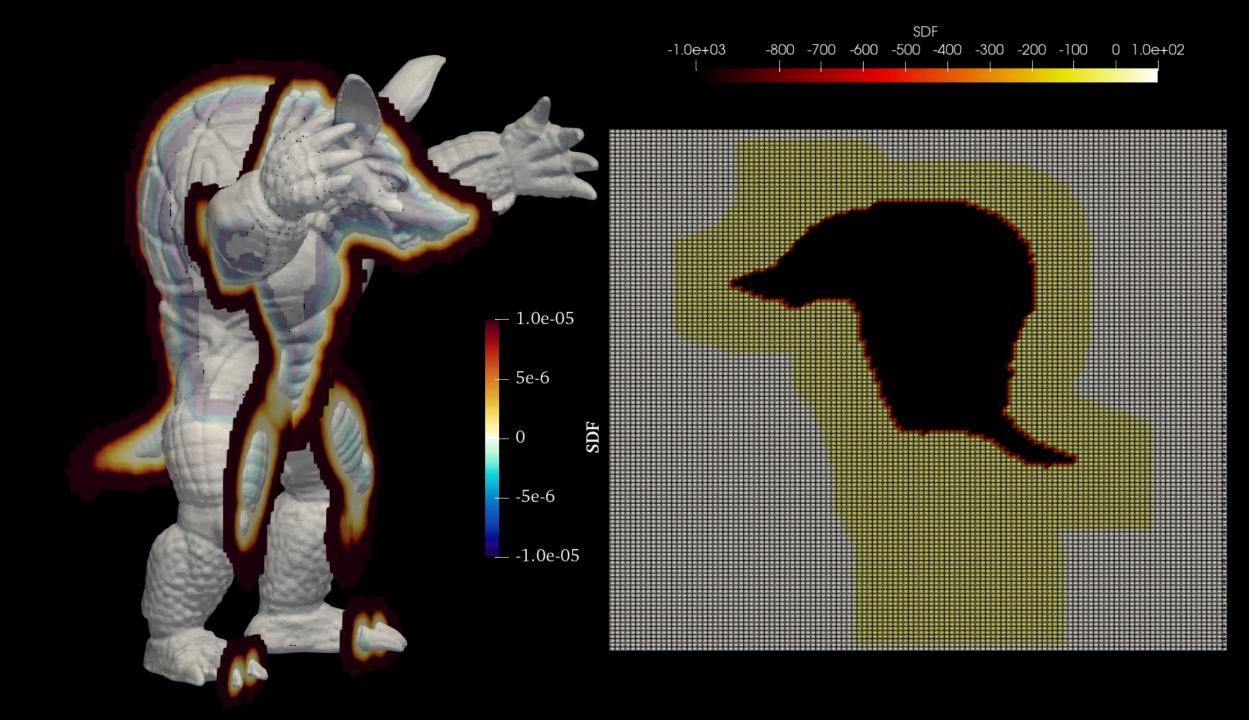


STEP 3: Decompose workload



18:04 | testrun | > mpirun -np 8 ./gensdf_mpi

<pre>**** Starting with 8 MPI ranks *** **** Starting with 8 MPI ranks ***</pre>
<pre>*** Successfully finished setting up the grid spacing *** Successfully read OBJ file: data/sphere_clipped.obj Number of vertices: 5856000 Number of normals: 5856000 Number of normals: 11700000</pre>
Number of faces: 11704000 Geometry is bounded by (minimum) 8.11000000000003E-006 1.0190000000001E-005 0.1100000000000000 Geometry is bounded by (maximum) 1.99998403000000 0.399989810000000 0.13000000000000000 *** Min-Max Index-Value pair *** Min-Max Index-Value pair *** 0.1300000000000000000000000000000000000
100.00% 11704000/11704000 Elapsed: 103.17s Remaining: 0.00s **** Writing output data to file *** Done with file write in 0.2325000000000082 seconds u-faces
*** Calculating the signed-distance-field v-faces *** 100.00% 11704000/11704000 Elapsed: 103.39s Remaining: 0.00s *** Writing output data to file *** Done with file write in 0.2323120000000030 seconds v-faces
*** Calculating the signed-distance-field v-faces *** 100.00% 11704000/11704000 Elapsed: 104.08s Remaining: 0.00s *** Writing output data to file *** Done with file write in 0.2327329999999963 seconds v-faces
<pre>*** Calculating the signed-distance-field Cell-Center *** 100.00% 11704000/11704000 Elapsed: 103.48s Remaining: 0.00s *** Writing output data to file *** Done with file write in 0.232099999999998 seconds Cell-Center *** Calculation for SDF completed in 445.86988200000000 seconds ***</pre>



Thank you!