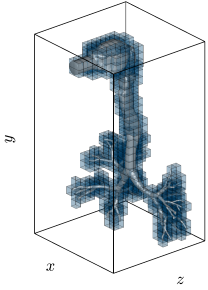
High-performance GPU-accelerated DNS solver for human airways

Andrea Portioli[a], Marco Atzori[a], Andrea Benassi[b], Maurizio Quadrio[a]



EFDC2, Dublin, 29/08

[a]Department of Aerospace Science and Technology, Politecnico di Milano, Campus Bovisa, Milano, 20156, Italy.
[b]Chiesi Farmaceutici S.p.A., Parma, Italy.

Why the human airways?

Several health applications can benefit from CFD data of the human lungs (drug delivery, COPDs...).

Main challenges:

- **Complex geometry** ($\approx 2^{23}$ bifurcations)
- **Wide scale separation** from mouth-throat region to the alveoli
- **Moving surfaces** drive the **inspiration cycle**

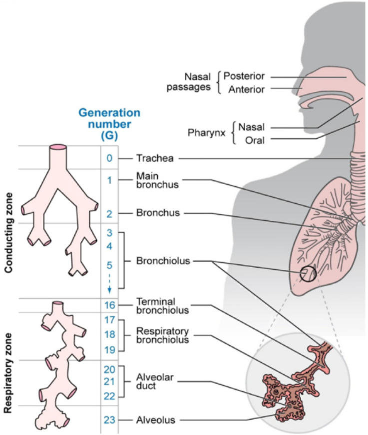


Image from: P. Koullapis et al., Regional aerosol deposition in the human airways: The SimInhale benchmark case and a critical assessment of in silico methods, *Eur. J. Pharm. Sci.*, 2018

Why a new DNS solver?

Flow field in the human airways **exhibit complex phenomena**[1]:

- Transition to turbulence
- Relaminarization
- Vortex shedding
- Secondary flows
- Separation

→

- **Limited effectiveness** of turbulence models
- Generation of an effective body-fitted mesh **is challenging**

[1] M. Atzori et al., Direct numerical simulations of inhalation in a 23-generation lung model, 2025, *Submitted*

AMPHIBIOUS

A **M**ulti**P**latform **H**igh-performance **I**mmersed-**B**oundary **I**ncompressible **O**pen-so**U**rce **S**olver

a) **Immersed boundary** method[1]

b) **Iterative** pressure correction algorithm

c) Domain decomposition based on **Building-CUBE** method (BCM) [2,3]

---

[1] P. Luchini et al., A simple and efficient second-order immersed-boundary method for the incompressible Navier–Stokes equations, *J. Comp. Phys.*, 2025

[2] K. Nakahashi & L. Kim, Building-Cube Method for Large-Scale, High Resolution Flow Computations, *42nd AIAA Aerospace Sciences Meeting & Exhibit* 2004

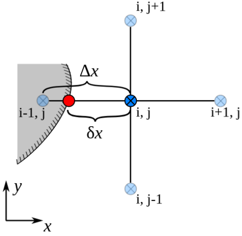
[3] N. Jansson et al., CUBE: A scalable framework for large-scale industrial simulations, *Int. J. High Perform. Comp. Appl.*, 2018

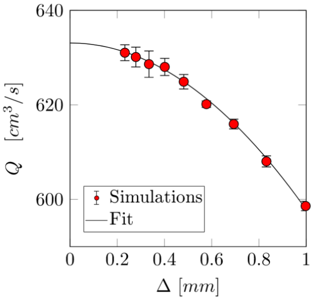
a) The IB method

Hypothesis:

Laplacian is dominant near the wall

- **2nd order accurate** in space and time
- **Stable** due to implicit correction
- **Efficient** in memory and computation

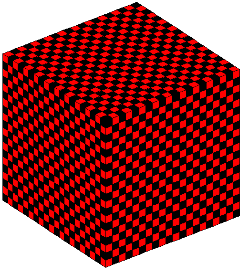




Images from: P. Luchini et al., A simple and efficient second-order immersed-boundary method for the incompressible Navier–Stokes equations, *J. Comp. Phys.*, 2025

b) Pressure correction

- **Weighted red-black Jacobi** method
- **Matrix free**
- Only requires local communications across subdomains
- Tolerance controlled through input parameters



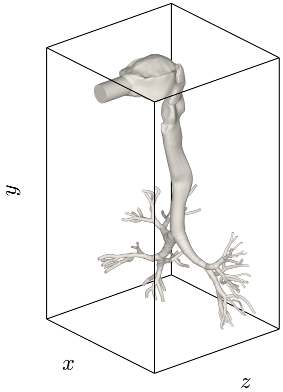
c) Building-Cube method

Domain decomposition and parallelization are separated

- **Cube**: portion of the domain discretised with a Cartesian grid
- Cubes in solid regions are discarted (pruned)
- Each process can manage multiple cubes

Solves common problems of structured grids for complex geometries:

- Memory wasted in solid regions
- Load balancing
- Flexibility

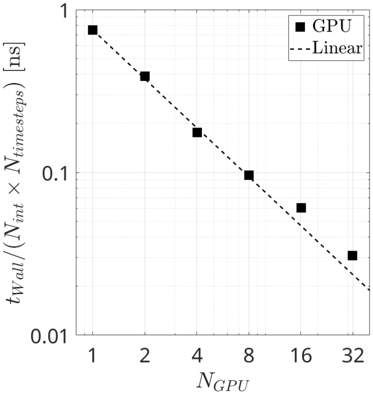


Pros & Cons

- Keeps performance of structured grids (especially important on GPUs)
- Significant memory savings in complex geometries
- Load balancing
- Keeping the same discretization: more cubes $\rightarrow$ more MPI communications

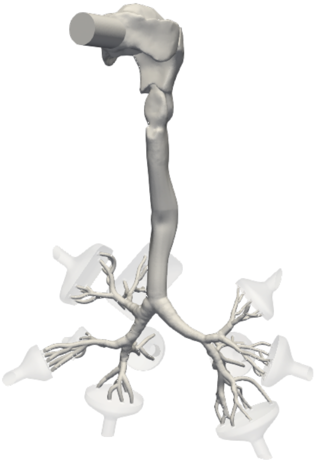
GPU offloading

- OpenACC for **portability**
- One GPU per process
- GPU-Aware MPI with RDMA and Peer-to-Peer memory access
- Multiple cubes **allow overlapping** computations and asynchronous communications



SimInhale benchmark

- **Only the first 5 generations** are considered
- **Non-deformable** walls
- **Steady inhalation** driven by a pressure difference



P. Koullapis et al., Regional aerosol deposition in the human airways: The SimInhale benchmark case and a critical assessment of in silico methods, *Eur. J. Pharm. Sci.*, 2018

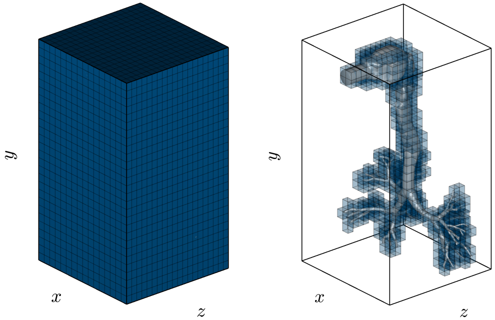
Performance

Δ	0.1 mm
N_{tot}	$\approx 8.35 \times 10^9$
N_{int}	$\approx 134 \times 10^6$
N_{int}/N_{tot}	$\approx 1.6\%$
$Cubes_{tot}$	8960
$Cubes_{int}$	665
$Cubes_{int}/Cubes_{tot}$	$\approx 7.5\%$

$(\Delta_{DNS} \approx 0.05$ mm$)$

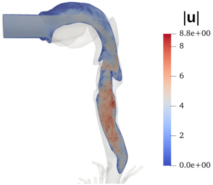
Using 32 NVIDIA A100 GPUs on Leonardo Booster:

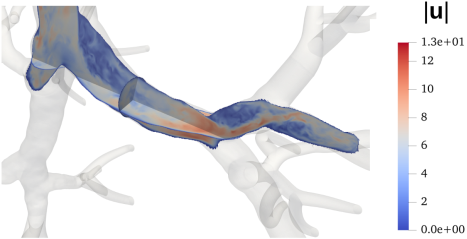
0.17 ns per internal point per timestep



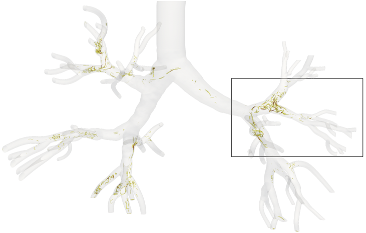
Flow field and resolution

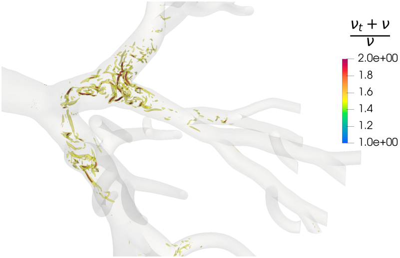
Flow field





A-posteriori resolution check using WALE Subgrid-Scale model





Conclusions

AMPHIBIOUS:

- Performs well on complex geometries
- Fast
- Open source

https://amphibious.cfd