Development and optimization of a coupled multi-GPU LBM-MHFEM solver for vapor transport in the boundary layer over a moist soil

Jakub Klinkovský[a], Andrew C. Trautz[b], Radek Fučík[a], Tissa H. Illangasekare[c]

[a] Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague
[b] Research Civil Engineer – Geotechnical Engineering and Geosciences Branch, US Army Engineer Research and Development Center
[c] Center for Experimental Study of Subsurface Environmental Processes (CESEP), Department of Civil and Architectural Engineering, Colorado School of Mines

2023 SIAM Conference on Computational Science and Engineering
February 28, 2023

Outline

1. Motivation
2. Governing equations
3. Coupling LBM and MHFEM
4. Overview of implementation and optimizations
5. Evaluation of parallel performance

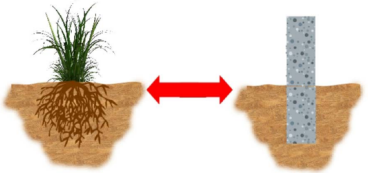
Environmental effects of land-atmospheric interactions

Joint work by Andrew C. Trautz[b] and Tissa H. Illangasekare[c]

[b]US Army Engineer Research and Development Center
[c]Center for Experimental Study of Subsurface Environmental Processes, Colorado School of Mines

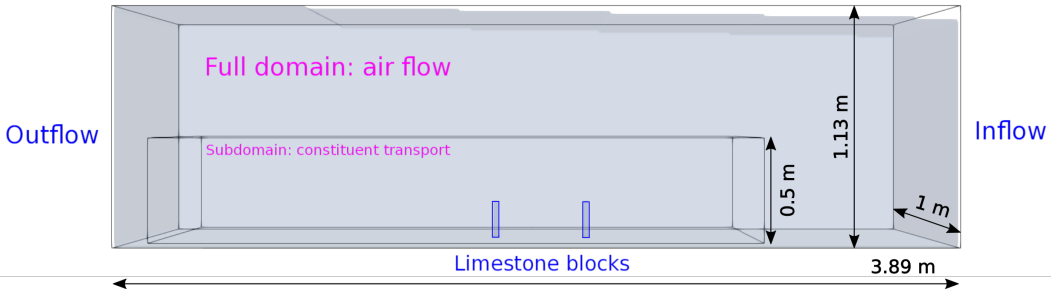
Experiments related to this talk:

- Climate-controlled, low-speed wind tunnel interfaced with a soil tank (CESEP, Colorado School of Mines, USA)
- Designed to study processes with mass flux across the land-atmospheric interface (e.g. water evaporation)
- Live vegetation approximated with limestone blocks



Computational domain

Only part of the wind tunnel above soil surface; 2 identical blocks; different spacings.



Governing equations: air flow and vapor transport

NSE (air flow in $\Omega_1 \times (0, t_{\max})$):

$$\nabla \cdot \vec{v} = 0, \tag{1a}$$

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = -\frac{1}{\rho} \nabla p + \nu \Delta \vec{v}, \tag{1b}$$

- $\vec{v}$ **fluid velocity,**
- ρ fluid density,
- p fluid pressure,
- ν kinematic viscosity of the fluid,

ADE (vapor transport in $\Omega_2 \subset \Omega_1$):

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \vec{v} - D \nabla \phi) = 0, \tag{2a}$$

Or in non-conservative form:

$$\frac{\partial \phi}{\partial t} + \vec{v} \cdot \nabla \phi - \nabla \cdot (D \nabla \phi) = 0. \tag{2b}$$

- $\vec{v}$ **fluid velocity,**
- ϕ relative humidity,
- D diffusion coefficient.

Coupled LBM-MHFEM approach

- Equation (1) – lattice Boltzmann method (LBM)
  - D3Q27, Cumulant collision operator *(M. Geier et al., 2015)*
  - A-A pattern for streaming
  - in-house code implementation *(R. Straka, R. Fučík, P. Eichler, J. Klinkovský et al.)*
  - implementation details later in this talk
- Equation (2) – mixed-hybrid finite element method (MHFEM)
  - *NumDwarf*: numerical scheme for a system of PDEs in a general-coefficient form
  - details in *R. Fučík, J. Klinkovský, J. Solovský, T. Oberhuber, J. Mikyška, Computer Physics Communications 238 (2019)*
- One-way coupling via the velocity field $\vec{v}$
  - Interpolation from the equidistant lattice to the MHFEM mesh

LBM-MHFEM: coupling details

Interpolation of the velocity $\vec{v}$:

- Trilinear or tricubic interpolation
- Evaluation at **cell side centers** (not cell centers) – to satisfy balancing requirements imposed by the MHFEM discretization

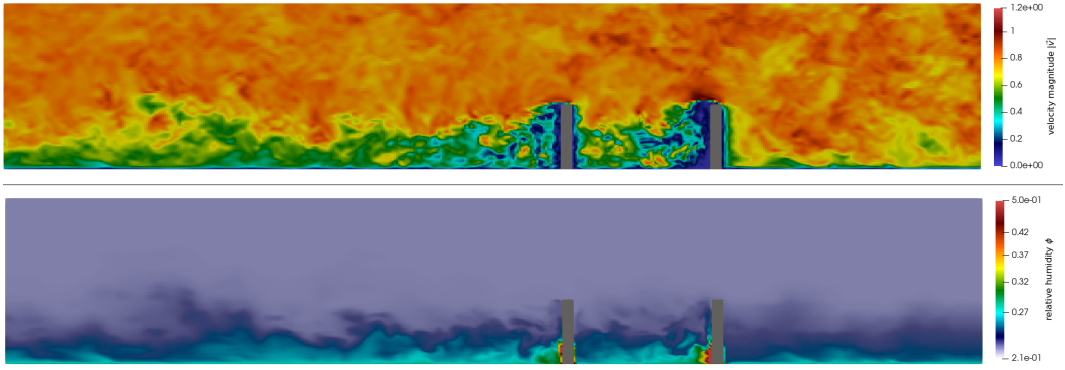
Transport equation:

- $\nabla \cdot \vec{v} = 0$ is not satisfied exactly by the LBM solver (weak compressibility)
- The interpolated velocity field is not locally conservative
- Numerical schemes for the **conservative and non-conservative variants are not equivalent**
- MHFEM discretization of the **non-conservative transport equation** includes a term that compensates for non-zero discrete velocity divergence

Time stepping:

- MHFEM allows to use larger time steps than LBM
- **Adaptive time-stepping** strategy for MHFEM based on a CFL-like condition

Example: simulation of velocity and relative humidity



Implementation overview

Main features:

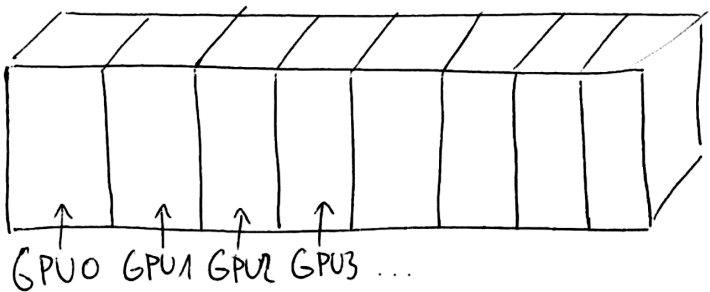
- All parts of the algorithm are computed on a GPU accelerator
- Multi-GPU implementation based on MPI

Custom code in C++ developed using:

- **Template Numerical Library**: `https://tnl-project.org/`
  More details: MS178, talk by Tomáš Oberhuber (Wed. March 1, 3:10-3:25 PM, room D507)
- **CUDA**: `https://docs.nvidia.com/cuda/`
- **Message Passing Interface**: `https://www.mpi-forum.org/`

Domain decomposition for LBM

- Computational domain = several independent subdomains + communication
- Computation: subdomains are processed on different GPUs
- Each MPI rank (process) manages its own GPU and subdomain
- Communication: 9 of 27 distribution functions need to be copied between adjacent subdomains
- For simplicity: only 1D decomposition (our current implementation) – not scalable



Basic LBM algorithm

1. Initialization (read input data, set initial condition, etc.)
2. While the final time is not reached, do **for all lattice sites in parallel**:
   1. Streaming step before collision (pull distribution functions from global memory)
   2. Compute macroscopic quantities (ρ, $\vec{v}$, etc.)
   3. Handle boundary conditions (boundary sites only)
   4. Collision
   5. Streaming step after collision (push distribution functions to global memory)
   6. Output macroscopic quantities to global memory

Note: steps 1 to 6 inside the loop are called **LBM iteration**.
Note: streaming steps before/after collision depend on the streaming pattern.

LBM algorithm with domain decomposition

1. Initialization (read input data, set initial condition, etc.)
2. Copy distribution functions on the boundaries between subdomains
3. While the final time is not reached:
   1. Perform the **LBM iteration** for **all lattice sites** on **all subdomains**
   2. Copy distribution functions on the boundaries between subdomains

LBM with overlapped computation and communication

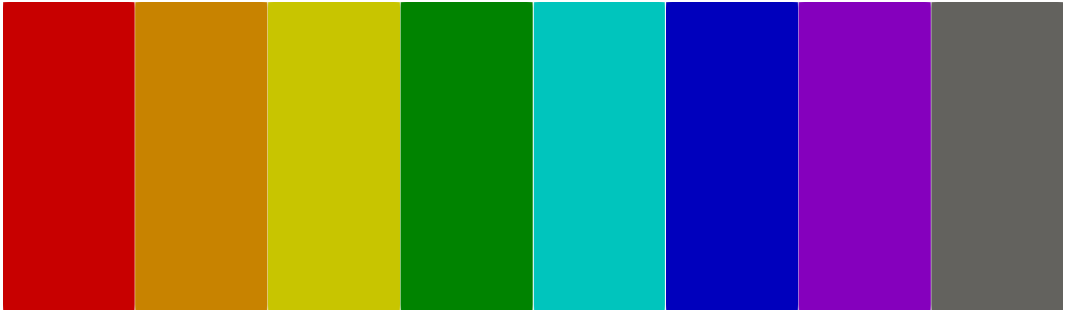
1. Initialization (read input data, set initial condition, etc.)
2. Copy distribution functions on the boundaries between subdomains
3. While the final time is not reached:
   1. On **all subdomains**, start LBM iteration for **lattice sites adjacent to subdomain boundary**
   2. On **all subdomains**, start LBM iteration for **remaining lattice sites**
   3. On **all subdomains**, wait until boundary lattice sites are processed
   4. On **all subdomains**, copy distribution functions on the boundaries between subdomains
   5. On **all subdomains**, wait until the remaining lattice sites are processed

Implemented optimizations

- Domain decomposition with overlapped computation and communication
  (`DistributedNDArraySynchronizer` class from TNL, implementation based on CUDA streams)
- Pipelining for asynchronous communication of relevant distribution functions (9 in each direction)
- Avoiding buffers in communication
  (specific ordering of data in multidimensional arrays is necessary)
- Direct GPU-GPU copies via "CUDA-aware" MPI
- Streaming with the A-A pattern – reduced memory requirements
- **Balancing decomposition of the lattice and mesh**

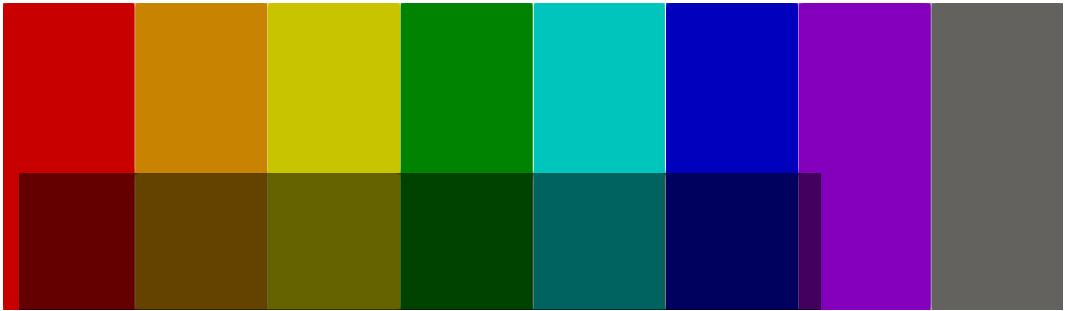
Balancing decomposition of the lattice and mesh

Uniform lattice decomposition: 1/8 of nodes in each subdomain



Balancing decomposition of the lattice and mesh

Uniform lattice decomposition: 1/8 of nodes in each subdomain

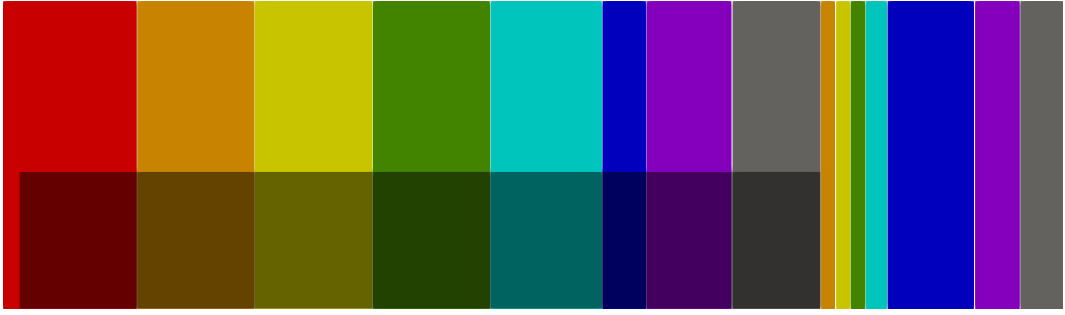


Unstructured mesh decomposition: non-uniform counts of mesh cells

12%	14%	14%	14%	24%	19%	3%	0%

Balancing decomposition of the lattice and mesh

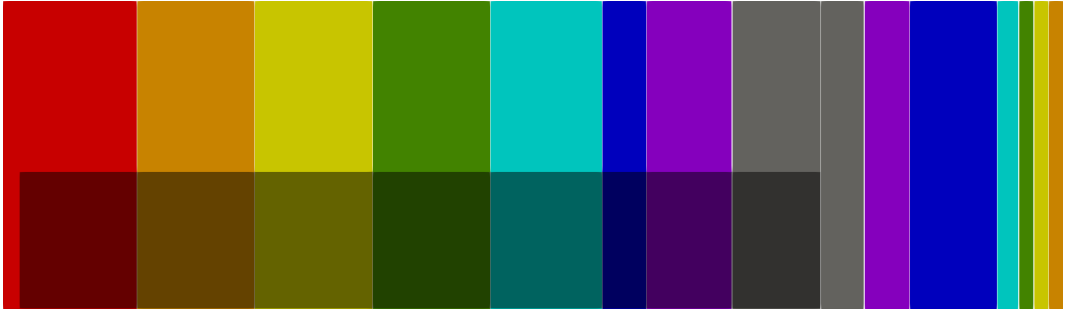
Balanced lattice and mesh decomposition:



Approx. 1/8 of mesh cells **and** approx. 1/8 of lattice nodes per MPI rank.

Balancing decomposition of the lattice and mesh

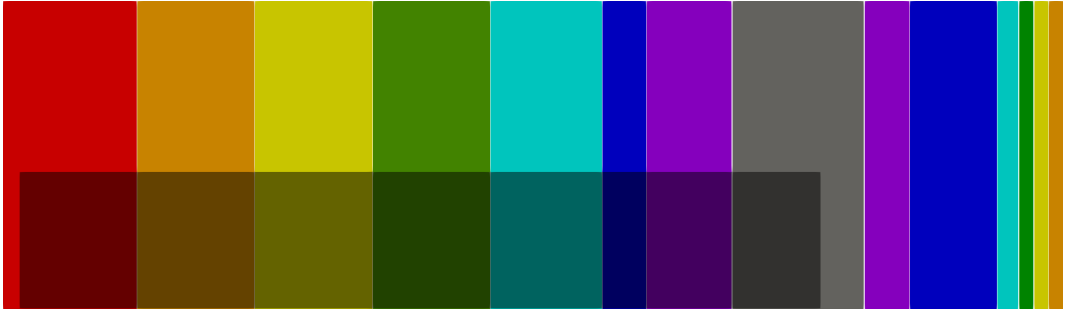
Balanced lattice and mesh decomposition:



Approx. 1/8 of mesh cells **and** approx. 1/8 of lattice nodes per MPI rank.

Balancing decomposition of the lattice and mesh

Balanced lattice and mesh decomposition:



Approx. 1/8 of mesh cells **and** approx. 1/8 of lattice nodes per MPI rank.

Karolina supercomputer – hardware specifications

Accelerated compute nodes in the Karolina supercomputer:

Number of nodes	72
Processors per node	2
CPU model	AMD EPYC 7763 (64 cores, 2.45-3.5 GHz)
Memory per node	1024 GB DDR4 3200 MT/s
Accelerators per node	8
GPU model	Nvidia A100 (40 GB HBM2 memory)
Intra-node connection	NVLink 3.0 (12 sub-links, 25 GB/s per sub-link per direction)
Inter-node connection	$4\times$ 200 Gb/s InfiniBand ports

The supercomputer is operated by IT4Innovations (https://www.it4i.cz/).

LBM: strong scaling on the Karolina supercomputer

Note: only LBM solver (i.e., no coupling with MHFEM)

		single precision			double precision		
N_{nodes}	N_{ranks}	GLUPS	Sp	Eff	GLUPS	Sp	Eff
1	1	5.2	1.0	1.00	2.8	1.0	1.00
1	2	10.2	2.0	0.98	5.5	2.0	1.00
1	4	20.4	3.9	0.98	11.1	4.0	1.01
1	8	41.1	7.9	0.99	22.3	8.1	1.01
2	16	80.4	15.5	0.97	44.1	16.0	1.00
4	32	145.2	28.0	0.87	85.5	31.0	0.97
8	64	258.6	49.8	0.78	153.7	55.7	0.87
16	128	301.1	58.0	0.45	225.1	81.6	0.64

- Lattice size: $512 \times 512 \times 512$
- Each MPI rank uses its own GPU
- GLUPS – billions of lattice updates per second
- Sp – speed-up relative to 1 GPU
- $Eff = Sp/N_{\text{ranks}}$ – parallel efficiency

Note: efficiency limited by 1D domain decomposition (not a problem for weak scaling)

Coupled LBM-MHFEM solver performance

- Decomposition algorithm – amount of work optimized at the cost of increased communication
- Only 1D decomposition is currently implemented – not scalable
- Tested with up to 16 GPUs (Nvidia A-100) on 2 nodes (RCI cluster on FEE CTU):
  - 16×40 GiB $= 640$ GiB memory on the GPUs
  - Up to $3115 \times 800 \times 905 \approx 2.25 \times 10^9$ lattice nodes + approx. 48×10^6 mesh cells
- Not tested on more GPUs (nodes) due to cluster limitations
- Strong scaling in low resolution (approx. 64×10^6 lattice sites and 12×10^6 mesh cells):

N_{ranks}	Time [min]	GLUPS	Eff
1	392	1.0	1.00
2	202	1.9	0.96
4	110	3.7	0.92
8	62	6.4	0.80

Conclusion

- Validated model for vapor transport in air based on LBM and MHFEM
- Fully multi-GPU solver with good scalability
  - The coupled LBM-MHFEM solver needs to be tested on larger supercomputer
  - Future work: optimizations for scalability on more GPUs (e.g. multidimensional decomposition)
- Multidisciplinary work – collaboration between experimental, numerical and computational methodologies
- Future work: extensions of the model (thermodynamics, coupling with porous media, etc.)

Thank you for your attention!

Acknowledgements:

- Czech Science Foundation (project 21-09093S)
- Ministry of Education, Youth, and Sports of the Czech Republic
  (Inter-Excellence grant LTAUSA19021, OP RDE grant CZ.02.1.01/0.0/0.0/16_019/0000765)
- Grant Agency of the Czech Technical University in Prague (project SGS20/184/OHK4/3T/14)
- *e-INFRA CZ* project ID:90140 of Ministry of Education, Youth and Sports of the Czech Republic

Related papers:

- J. Klinkovský, A. C. Trautz, R. Fučík, T. H. Illangasekare: Lattice Boltzmann Method–Based Efficient GPU Simulator for Vapor Transport in the Boundary Layer Over a Moist Soil: Development and Experimental Validation, Computers & Mathematics with Applications, accepted Feb 22, 2023
- R. Fučík, J. Klinkovský, J. Solovský, T. Oberhuber, J. Mikyška: Multidimensional mixed-hybrid finite element method for compositional two-phase flow in heterogeneous porous media and its parallel implementation on GPU, Computer Physics Communications, 2019