LBM: Usage & Coding

AC290r - Simone Melchionna

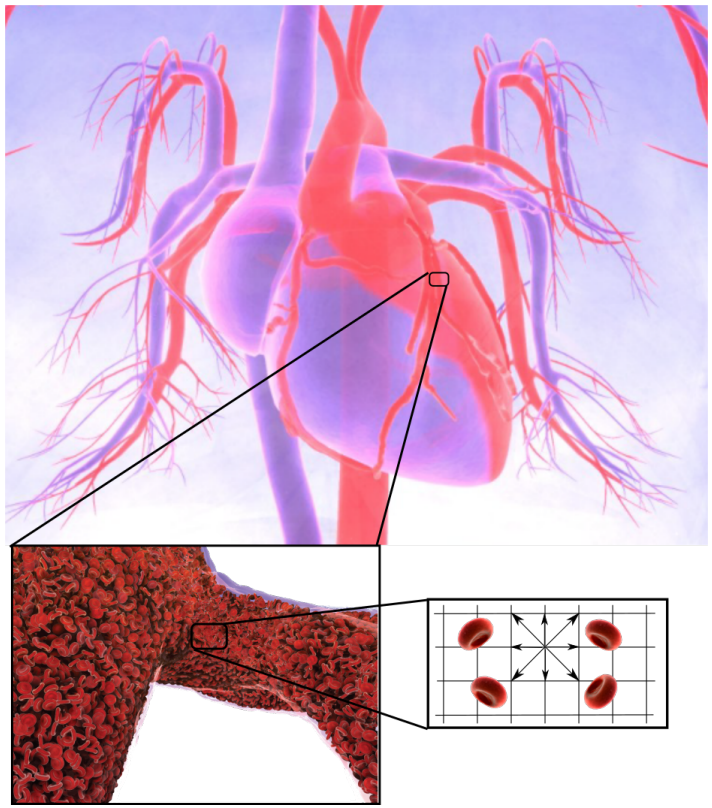
National Research Council Italy & Lexma Technology LLC

April 3, 2019

Module objectives

Hemodynamics is at the cross-road between engineering, medicine and high-performance computing. In this module we will:

* simulate complex geometric systems, like multi-branched arterial trees, with LBM$^+$
* familiarize with multi-physics and multi-scale concepts that are ubiquitous in life science applications.



Lattice Boltzmann Classics

At its core, LBM utilizes a cubic grid where each node communicates with its neighbor nodes the fluid populations

$$f_p(x, t)$$

the index p labels a set of vectors connecting grid neighbors, $\mathbf{c}_p$. These vectors are de facto discrete velocities that connect 1st, 2nd or farther node neighbors.

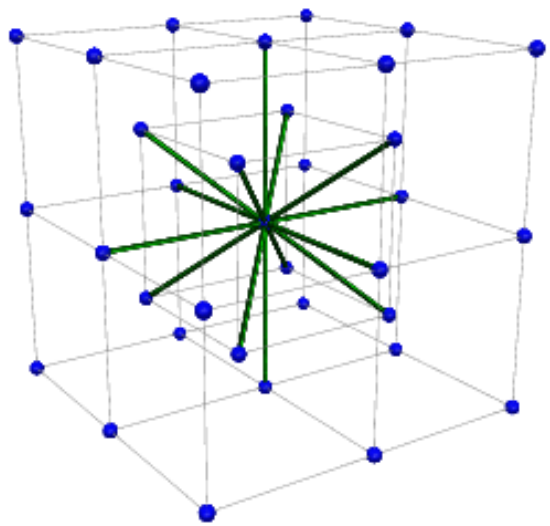


Figure: Cubic cartesian grid in 3D

Advancing Populations

Populations move according to the following rule, here written in Bhatnagar-Gross-Krook (BGK) form:

$$f_p(x + hc_p, t + h) = f_p(x, t) + \omega h \left[f_p^{eq}(\rho, \mathbf{u}) - f_p \right](x, t) + w_p \frac{\mathbf{c}_p \cdot \mathbf{g}}{c_s^2}$$

where ω is a relaxation frequency, h the timestep (often assumed to be unit $h = 1$ in simplified notation) and $\mathbf{g}$ the body acceleration. $f_p^{eq}$is the (pseudo) equilibrium:

$$f_p^{eq}(\rho, \mathbf{u}) = w_p \rho \left[1 + \frac{\mathbf{u} \cdot \mathbf{c}_p}{c_S^2} + \frac{(\mathbf{c}_p \cdot \mathbf{u})^2 - c_S^2 u^2}{2c_S^4} \right]$$

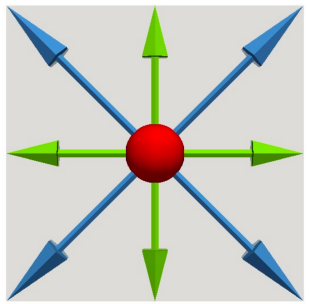
with w_p normalized weights $\sum_p w_p = 1$ and the lattice speed of sound$c_S = 1/\sqrt{3}$.
If populations coincide with equilibrium then the fluid is inviscid.

DnQm nomenclature

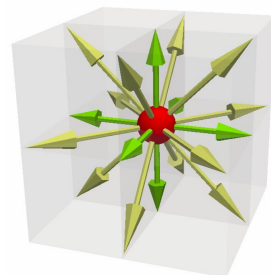
For 2D or 3D systems, we can choose different sets for $\mathbf{c}_p$ and w_p. Common choices are:

* D2Q9 in 2D: with four 1st neighbors, four second neighbors and one population with zero speed (rest population)
* D3Q19 in 3D: with six 1st neighbors, twelve second neighbors and one population with zero speed (rest population)

i	0	1	2	3	4	5	6	7	8
w_i	$\frac{4}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
c_{ix}	0	+1	0	−1	0	+1	−1	−1	+1
c_{iy}	0	0	+1	0	−1	+1	+1	−1	−1



i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
w_i	$\frac{1}{3}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
c_{ix}	0	+1	−1	0	0	0	0	+1	−1	+1	−1	0	0	+1	−1	+1	−1	0	0
c_{iy}	0	0	0	+1	−1	0	0	+1	−1	0	0	+1	−1	−1	+1	0	0	+1	−1
c_{iz}	0	0	0	0	0	+1	−1	0	0	+1	−1	+1	−1	0	0	−1	+1	−1	+1



Macro Quantities

f_p^{eq} is constructed starting from macroscopic quantities, fluid density:

$$\rho(\mathbf{x}, t) = \sum_p f_p(\mathbf{x}, t)$$

and momentum density (or current), $\mathbf{J}(\mathbf{x}, t) = \rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t)$, so that fluid velocity is:

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)} \sum_p \mathbf{c}_p f_p(\mathbf{x}, t)$$

It is not accidental that populations and equilibrium share the first two moments: $\sum_p f_p = \sum_p f_p^{eq}$ and $\sum_p \mathbf{c}_p f_p = \sum_p \mathbf{c}_p f_p^{eq}$.

Back to NSE

Given the Navier-Stokes equation (NSE)

$$\frac{\partial(\rho u_\alpha)}{\partial t} + \frac{\partial \rho_\alpha u_\alpha u_\beta}{\partial x_\beta} = -\frac{\partial p}{\partial x_\alpha} + \frac{\partial}{\partial x_\beta}\left[\eta\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right) + \left(\eta_B - \frac{2}{3}\eta\right)\delta_{\alpha\beta}\frac{\partial u_\gamma}{\partial x_\gamma}\right] + g_\alpha$$

with η and η_B shear and bulk dynamic viscosities and g body acceleration, from the condition $\rho = const \longrightarrow \frac{\partial u_\gamma}{\partial x_\gamma} = \nabla \cdot \mathbf{u} = 0$ descends the Incompressible Navier-Stokes equation

$$\frac{d\mathbf{u}}{dt} = \left(\frac{\partial}{\partial t} + \nabla\cdot\right)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{g}$$

as reproduced by LBM at times longer than the collisional one. Kinematic viscosity relates to the BGK relaxation frequency by

$$\nu = c_s^2\left(\frac{1}{\omega} - \frac{1}{2}\right)$$

Equation of State

The Equation of State (EOS) of the simple ideal gas law relates pressure to density

$$p = \rho R T$$

(R: specific gas constant) and the isoentropic EOS reads $p = p_0 \left(\frac{\rho}{\rho_0}\right)^{\gamma}$ with $\gamma = c_P/c_V$: adiabatic index and c_P, c_V heat capacities at constant pressure and volume (with $c_P = c_V + R$). At constant temperature, we obtain the isoentropic ideal gas law

$$p = \rho R T_0$$

In practice, the ideal gas approximation is equivalent to take pressure around a reference value and Taylor expand, so that

$$p \simeq p_0 + c_S^2 \rho$$

having introduced the speed of sound $c_S^2 = \left(\frac{\partial p}{\partial \rho}\right)_S$. Thus for the isoentropic EOS: $c_S = \sqrt{\gamma R T_0}$ and for the isothermal EOS: $c_S = \sqrt{R T_0}$.

BGK - LBM works implicitly with isothermal EOS.

Collide and Stream

Let's analyse again the LBM core and rewrite it as two elementary steps:

1. compute the temporary post-collisional populations:

$$f_p^* = (1 - \omega) f_p + \omega f_p^{eq}$$

2. stream the post-collisional populations to neighboring sites

$$f_p(x + c_p, t + 1) = f_p^*(x, t)$$

The two steps illustrates the basic mechanism: computing $f_p^*$is a local operation, while streaming is not.

LBM Pseudo-code

Consider a first array storing populations: $pop[p, ifl]$ with $p = 0, ..., 18$ and ifl running over the nodes.
A simple scheme is to use a second array for storing the post-streaming populations. However, this would require unnecessary swaps and/or copies. Instead, let's use pointers through the following pseudo-code:

1. allocate the arrays $pop1$ and $pop2$ and the pointers pop and $tpop$ set as : $pop \rightarrow pop1$ and $tpop \rightarrow pop2$
2. initialize populations and store in pop
3. compute the post-collisional populations:
$pop = (1 - \omega) \times pop + \omega \times popeq(\rho, u)$
4. stream data and store to the array pointed by the second pointer: $tpop = Stream(pop)$
5. Swap pointers:
   if odd timestep: $pop \rightarrow pop1$ and $tpop \rightarrow pop2$
   if even timestep: $pop \rightarrow pop2$ and $tpop \rightarrow pop1$

Goto 3. and repeat until stop signal.
As apparent, the sequence avoids memory copies.

The LBM stability range

We don't analyse the LBM stability in detail but, as a rule of thumb, optimal stability occurs when the rest equilibrium population is positive, then

$$|\mathbf{u}| < \sqrt{\frac{2}{3}} \frac{\Delta x}{\Delta t}$$

Being a representative value $u = 0.1$, then kinematic viscosity that provides stable results is $5 \times 10^{-2} < \nu < 1$.
When we will work on the channel flow, we will play with these parameters to see how LBM responds.

Initial Conditions

Initial values for density and velocity usually obey some initial conditions of the type:

$$\rho(x, t = 0) = \rho_0(x)$$

$$u(x, t = 0) = u_0(x)$$

(pressure and density are basically the same thing). Typically, one sets uniform initial density and zero initial velocity.
Whatever is the initial state, the initial populations can be set as equilibrium values:

$$f_p(x, 0) = f_p^{eq}(\rho_0(x), u_0(x))$$

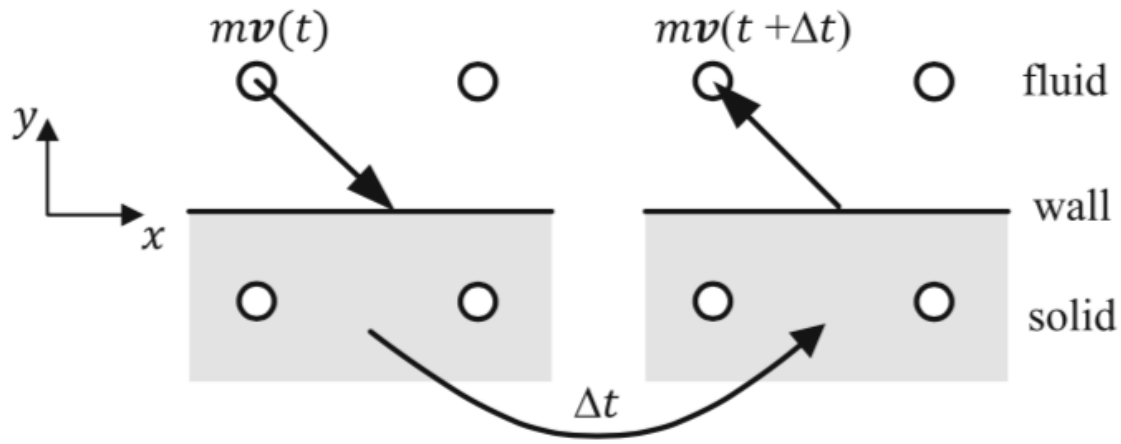
which assumes a specific condition that will evolve via a transient towards the steady or unsteady final state.

Boundary Conditions

What happens at walls and open boundaries (inlets or outlets)?
In general boundary conditions (BC) fall under following the generic condition of a PDE $b_1 \frac{\partial \phi}{\partial n}\Big|_{x_b,t} + b_2\phi(x_b,t) = b_3$ with ϕ density or velocity, n the wall normal, b_1, b_2, b_3 constants.
Choosing $\phi = \mathbf{u} = 0$ is the no-slip boundary conditions.
In general, we call Dirichlet BC when $b_1 = 0$ and $b_2 \neq 0$(imposing the field value), Neumann BC when $b_1 \neq 0$ and $b_2 = 0$ (imposing the flux) and Robin BC when $b_1 \neq 0$ and $b_2 \neq 0$ (imposing a relation between flux and field value).
At any boundary, we can impose one of the following rules:
* static or moving (locally planar) wall surfaces, with a condition of zero-flux
* inlet/outlet surfaces with given conditions on pressure or velocity

No-Slip BC

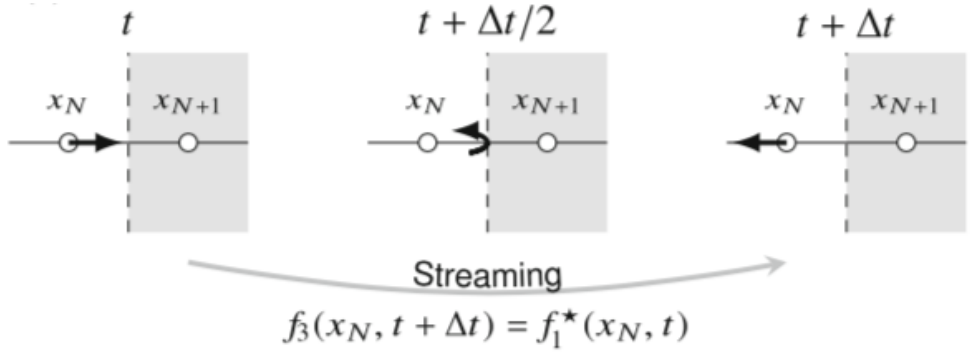
No-slip BC reproduce most of macroscopic near-wall behavior.



Given the collision & streaming distinction:

$$f_p(x + c_p, t + 1) = f_p^*(x, t)$$

clearly some populations will fall off from "fluid" nodes to no-man's land (say, on wall nodes).
No-slip BC are imposed by reflecting populations at walls!



Call for p population its conjugate (reversed) $\bar{p}$ and correct the post-streaming at near-wall node x_N as

$$f_{\bar{p}}(x_N, t+1) = f_p^*(x_N, t)$$

The scheme modifies in presence of a moving wall having density and velocity ρ_w and $\mathbf{u}_w$ as:

$$f_{\bar{p}}(x_N, t+1) = f_p^*(x_N, t) - 2w_p\rho_w \frac{\mathbf{c}_p \cdot \mathbf{u}_w}{c_S^2}$$

For inlets and outlets instead a first approximation is to substitute populations with their equilibrium values:

$$f_p(x_{io}, t+1) = f_p^{eq}(\rho_0, \mathbf{u}_0)$$

followed by streaming.
Question: when does this approximation hold in a channel flow ?

LBM for low Mach flows

LBM is a quasi-compressible method with ideal gas thermodynamics. Can we simulate a real liquid ?
Let's note that in the NSE only the gradient of pressure enters.
Therefore, $p = c_S^2 \rho$ can be always re-expressed as $p = c_S^2(\rho_0 + \rho')$
at constant entropy, we can map the ideal gas to a real gas by setting $c_S^2 \rho_0$ to a reference pressure. Consider now:
Reynolds number $Re = \frac{t_{diff}}{t_{conv}} = \frac{ul}{\nu}$ Mach number $Ma = \frac{t_{sound}}{t_{conv}} = \frac{u}{c_S}$
$Re \ll 1$ is where momentum diffusion is efficient, the realm of microfluidics and biophysics
$Re \gg 1$ is where momentum diffusion is slow and inertia dominates, the realm of turbulence and engineering
$Ma \ll 1$ for incompressible flows, such as for liquids, the realm of microfluidics and biophysics
$Ma > 1$ for increasingly compressible fluids, from transonic flows to shock waves, again engineering
In the following, we will focus on low Re/low Ma regimes.
However, LBM is highly versatile...

Let's recall the adimensional form of the NSE

$$\frac{d\tilde{\mathbf{u}}}{dt} = -\tilde{\nabla}\tilde{p} + \frac{1}{Re}\tilde{\nabla}^2\mathbf{u} + \tilde{\mathbf{F}}$$

where: $\tilde{u} = \frac{u}{u_{char}}$, $\tilde{p} = \frac{p}{\rho u_{char}^2}$, $\tilde{F} = \frac{Fl_{char}}{\rho u_{char}^2}$, $\frac{\partial}{\partial \tilde{t}} = \frac{l_{char}}{u_{char}}\frac{\partial}{\partial t}$, $\frac{\partial}{\partial \tilde{x}} = l_{char}\frac{\partial}{\partial x}$ and, once we match Re, we can simulate incompressible flows. For compressible systems, we also expect to match both Re and Ma. Finally, another important number is the Knudsen number

$$Kn = \frac{l_{mfp}}{\ell} = \frac{Ma}{Re_\ell}$$

where l_{mfp} is the mean free path and ℓ a given lengthscale. Kn has basically two regimes:
Kn $\ll 1$: NSE holds on the ℓ scale
Kn > 1: kinetic regime holds on the ℓ scale, since molecular collisions have not acted for sufficiently long times.

Mapping LBM to physical units

Consider Δx grid spacing in physical units (say, in MKS units) and $\mathbf{U}$ a characteristic velocity of the problem (say the inlet velocity in MKS).
From now on, we will write LBM quantities in lowercase and those in physical units in uppercase .
Being LBM a time-explicit technique, we want to express the timestep Δt in physical units. Since
$\mathbf{u}\frac{\Delta x}{\Delta t} = \mathbf{U}$:

$$\Delta t = \frac{\mathbf{u}}{\mathbf{U}}\Delta x$$

Setting also the fluid mass in the unit cell Δm allows us to dimensionalize all derived quantities.

Numerical and physical kinematic viscosity (ν_{ph}) map as:

$$\nu = \frac{\Delta t}{\Delta x^2} \nu_{ph}$$

Obviously $\nu \sim 1/\Delta x$ and therefore this is equivalent to matching the Reynolds number of the problem since $UL/\nu_{ph} = u\Delta x\mathcal{N}/\nu$, where $\mathcal{N}$ is the number of mesh nodes over the characteristic length L.
If we need to increase ν to stabilize a simulation, then we need to reduce Δx which in turn implies reducing Δt proportionally.

Grids and Meshes: the cartesian structure

As you already know, LBM works with an underlying set of nodes where the fluid flows between walls. Before we move on, let's define a mesh and a grid. In CFD there is no firm distinction but in general a grid is a set of elements that have a well defined structure, square or rectangular grids being the most prototypical; mesh is a more general set of elements.
They may be unstructured and use various shapes of elements, sometimes even mixing elements of different types in the same mesh. For instance, a closed surface made by a set of triangles creates a water-tight system.
LBM uses the cartesian grid to fills computational space, a measure of resolution being $1/\Delta x$.
The cartesian structure provides a simple and easily manageable object, in fact, the grid can be constructed by simply setting nodes coordinates to integer values.

Grid management

The simplicity to construct a cartesian mesh is a main advantage of LBM.
Suppose that you have a closed-shell surface, say a sphere, and you need to construct the LBM grid inside the surface.
* Would you think of a good (possibly linear) algorithm to create the nodes inside the surface ?
* Would you method work in presence of multiple non-intersecting surfaces ?
In general, a solid model of real-life conduits is made of curved surfaces. Adapting LBM to handle curved surfaces requires extra algorithmic work not covered here.
However, an approximation is to represent the real surface by staircasing, that is, approximate the curved surface by cartesian nodes that identify the inner and outer regions.
How accurate is staircasing? It can be proven that it is first-order accurate and, in the case of special alignment with the wall (say along the x, y , z axes), it is second-order accurate.

Grid management(2)

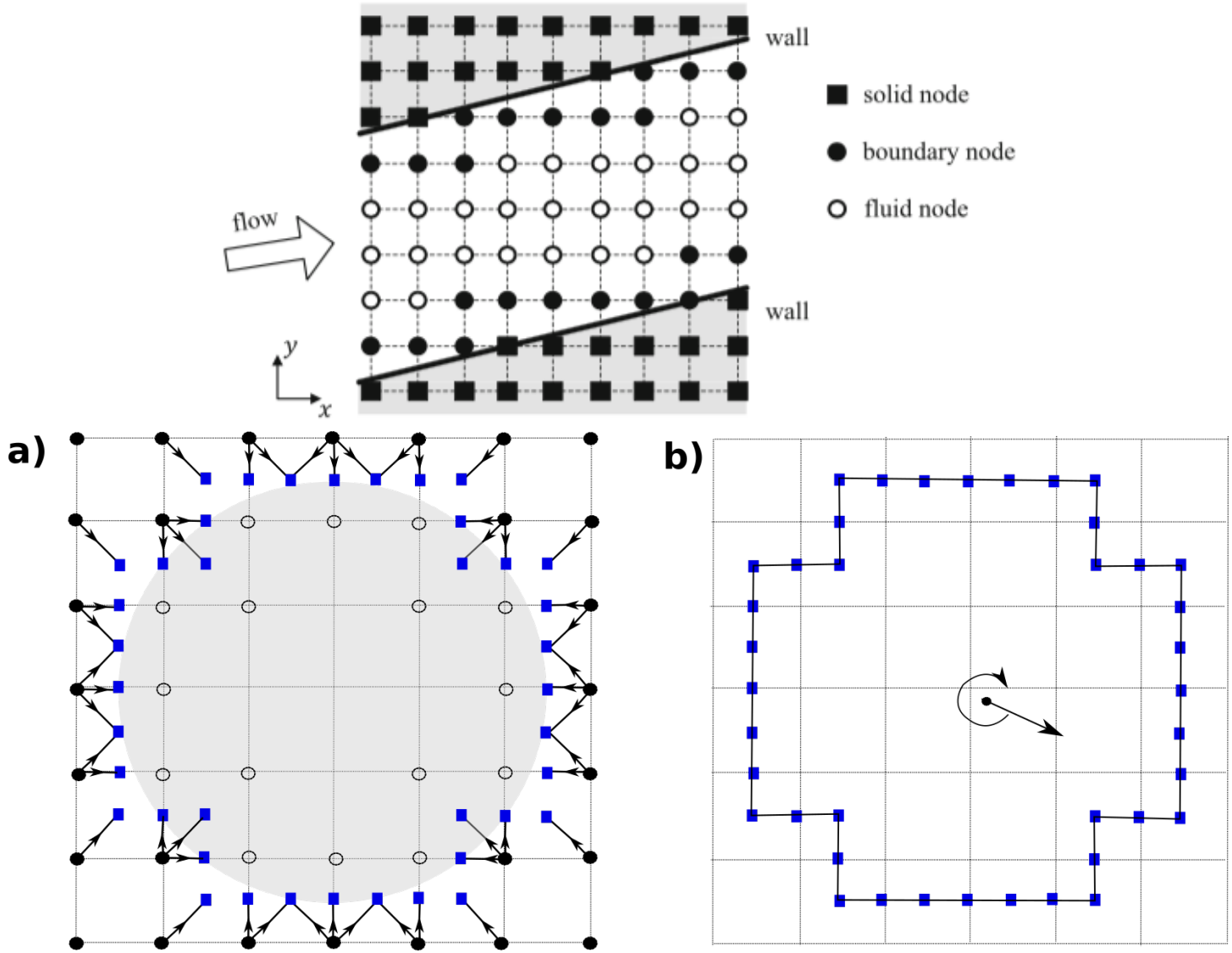


Figure: Staircasing to represent non-aligned or curved surfaces

Grid management(3)

Even a multibranched artery, typical example of smooth and topologically complex object, can be staircased.
This operation is usually performed automatically and with no human guidance!
In addition, we can push the resolution to high values to get a systematically better approximation to the real system, minimizing the roughness (dissipation) arising from staircasing. Adding more nodes is relatively cheap in LBM in terms of floating point operations, therefore increasing resolution is always the best option.

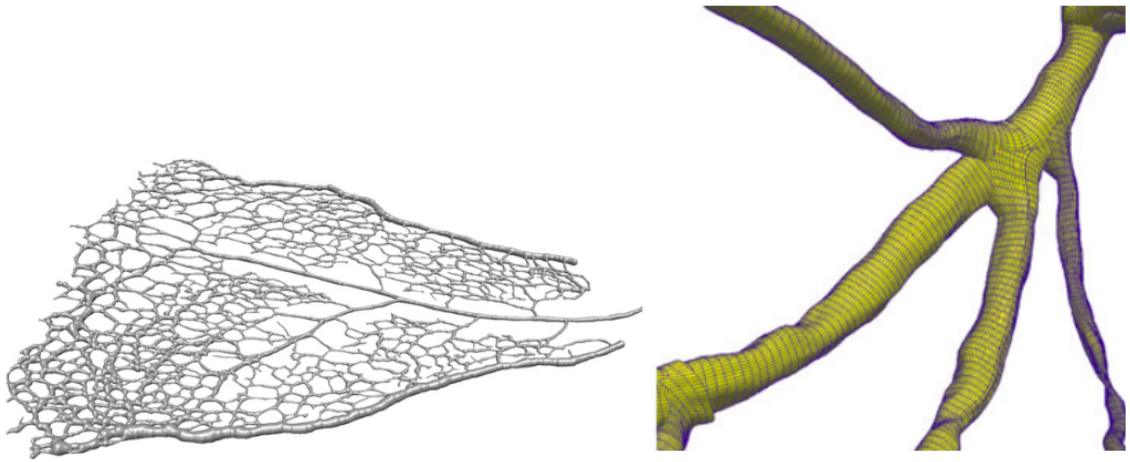


Figure: A capillary network (left) and human coronary arteries (right)

Direct vs Indirect addressing

Let's call the nodes i, j, k and their identity is given by the array $flg[i, j, k] = 1, 2, 3, 4$ for fluid, wall, inlet, outlet nodes.
Streaming LBM populations can be done by addressing nodes and its neighbors directly or indirectly.
Direct addressing is the most natural and simple option: for every mesh node i, j, k and array $A[i, j, k]$, one can address the node neighbors as $i \pm 1, j, k$, $i, j \pm 1, k$ and $i, j, k \pm 1$, as much as $i \pm 1, j \pm 1, k$, $i, j \pm 1, k \pm 1$ and $i \pm 1, j, k \pm 1$. We can also know the identity of the post-streaming destination node.
But direct addressing becomes increasingly ackward if the geometry is non-trivial since streaming will become stuffed with conditionals.
Indirect addressing is the next option. Here for each discrete velocity labelled p, the connectivity matrix $Conn[p, i, j, k]$ point to the neighbor node i', j', k'. The connectivity matrix also allows accessing the neighbor nodes in a compact way.

Sparsity

It's crucial to simulate large and complex physiological systems that are naturally sparse. Typically, only 10% of space is filled by blood conduits while the rest is just empty space, which ultimately makes storage of populations a very important issue.
One strategy is to compactify storage by using one dimensional arrays: the array $A[ifl]$ where the index ifl runs over the number of (fluid, wall, inlet, outlet) nodes in the system. Also, let us consider the mapping that maps the i, j, k triplet in row-major order: $i4 = nx \times ny \times k + nx \times j + i$. The array $itp[ifl] = i4$ stores the node location, since one can always derive i, j, k from the $i4$ value. Another main benefitis to access any point in the array efficiently. Searching node i^*, j^*, k^* in the array is best down once the itp is sorted: a bisection search method retrieves the node location in logarithmic time (typically searching an index in millions requires ~6 operations).

Sparsity(2)

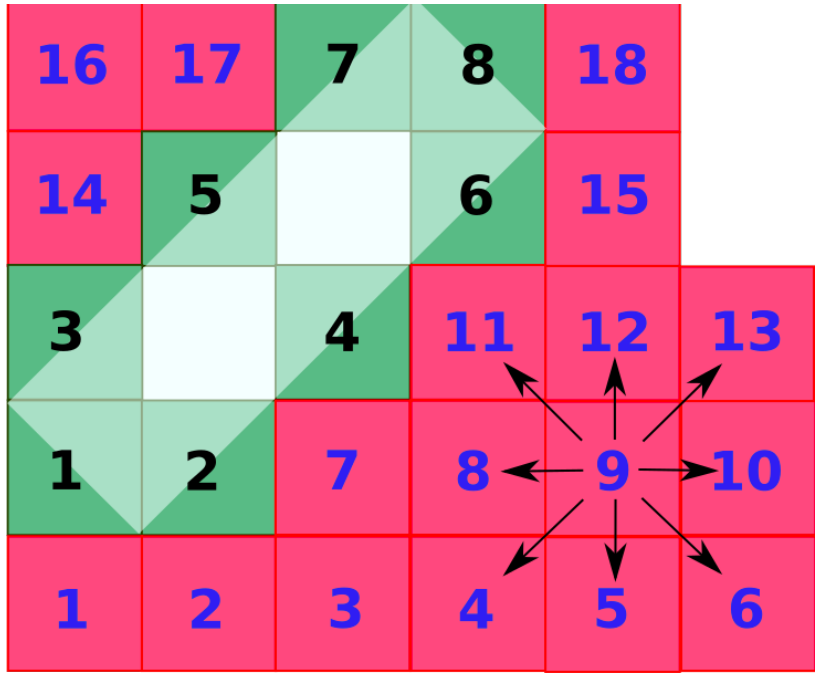


Figure: Indirect addressing and mapping data in row-major order.

How to organize a LBM code

The organization of a LBM code mirrors its simplicity. Let's remember that in LBM there is no need to compute finite differences or intergrals (fluxes). Below is the scheleton of a LBM code.

1. Read input parameters
2. Set populations to equilibrium
3. Set timestep = 0
4. While (timestep < TotalSteps):
5. Boundary Conditions at inlet/outlet nodes
6. Collide populations on fluid nodes
7. Stream populations from fluid, inlet and outlet nodes
8. Bounce-back BC at wall nodes
9. Every n steps, perform analysis and write data on disk
10. timestep += 1 and Goto 4.

How to organize a LBM code(2)

The code skeleton is straightforward and could benefit from optimization techniques that we will cover in the following sections.
For now let's consider the operations related to BCs. In general, there are multiple ways to impose BC at inlet/outlet and wall nodes.
In the pseudo-code scheme proposed here populations are first set at inlet and outlet nodes as equilibrium populations, which will be subsequently streamed to the adjacent fluid nodes. At this point fluid nodes adjacent to inlet/outlet nodes are complete.
For wall nodes the operation is different: each fluid/inlet/outlet node will stream populations and some of them will fall on wall nodes. The bounce back operation will be applied to reverse populations.
At this point all populations are complete.

An idealized blood vessel

Consider a straight artery with a diameter of 5 mm, let's discretize with $\Delta x = 10^{-4} m$, knowing that blood streams at $U = 0.1\text{m/s}$.
What is a LBM typical velocity (in LBM units)?
A value that provides stable results just next to the inlet is $u = 0.1$ implying $\Delta t = 10^{-4} \times 10^{-1}/10^{-1} = 100\mu s$.
Let's imagine that we want to cover a heart beat period ($\sim 1s$).
This means that we need to run for at least 10^4 steps to get reasonable results but most typically for more than 10^5 steps to cover a few heart beats and get converged results.
In reality we will see that this is not sufficient too. Taking for blood the viscosity of water, it follows that $\nu = 10^{-2}$. This value is too small for LBM and a simple BGK scheme would not run at all.
To increase viscosity, we need to increase resolution to $\Delta x = 5 \times 10^{-5}\text{m}$ and Δt will increase proportionally.