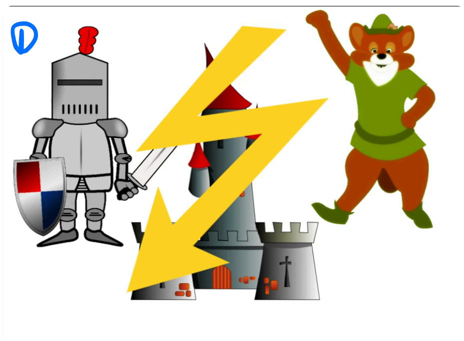
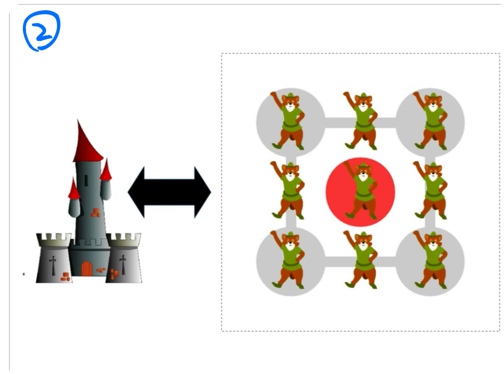
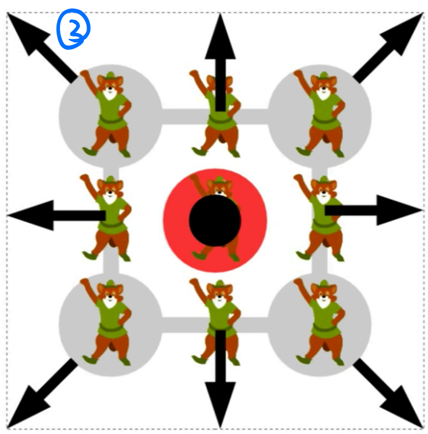
Short Intro to Lattice Boltzmann Method

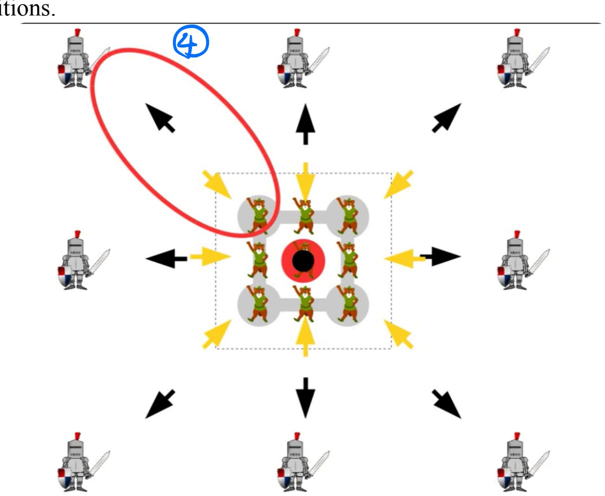
Video Title: Of Foxes, Attackers, ... and the Lattice Boltzmann Method
Link: https://www.youtube.com/watch?v=trvSBGyK74g



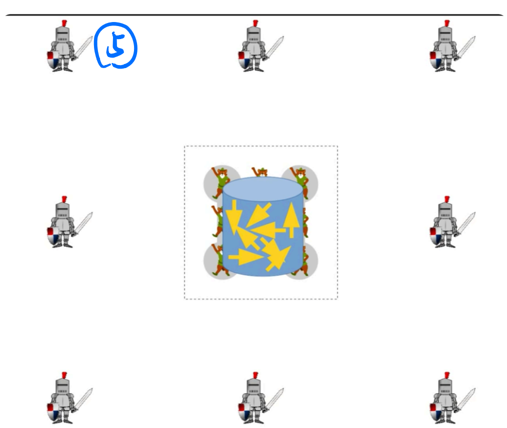


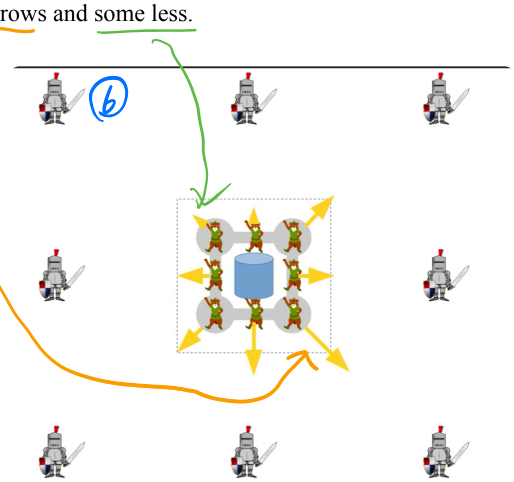
An analogy (example): Once upon a time, foxes live in the castle and attackers want to attack the castle. Looking from the top of the castle, foxes take these strategic positions to defend their home. 8 foxes stay at the castle walls and 1 fox stay in the center to distribute arrows to his mates. All attacker also move to their positions.

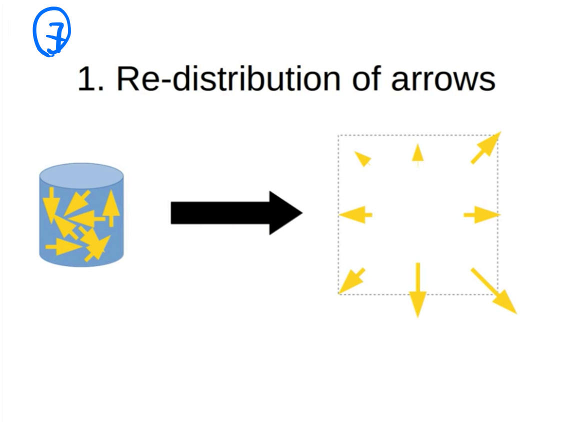


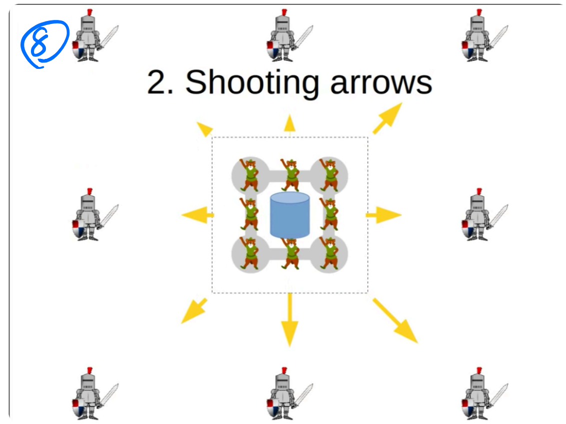


The battle begins: The attackers shot arrows into the castle and the foxes shot arrows towards the attackers. Note: Each attacker shot only towards one particular fox that he could directly spot and vice versa! In order NOT to run out of arrows, the foxes collected the arrows of the attackers in a big box. With the help of the fox in the center, they redistributed the arrows according to the needs of each individual defender. Some foxes obtained more arrows and some less.

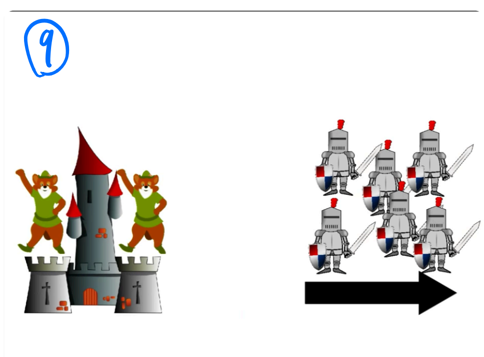


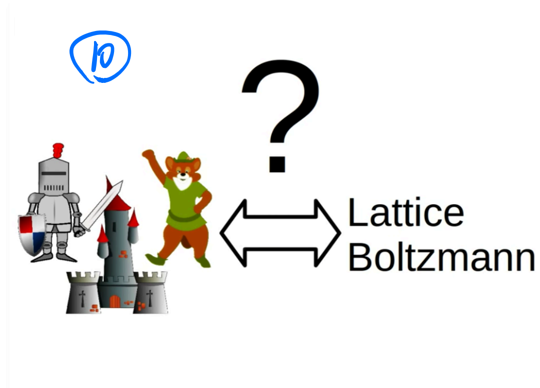






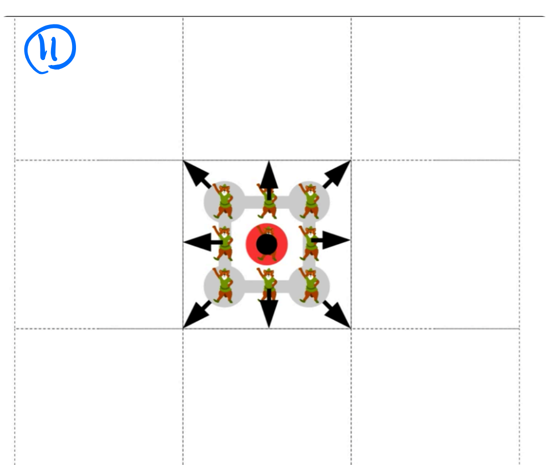
By this clever move, that is, redistribution of arrows and shooting the arrows towards the attackers, the attackers get scared and run off. The foxes successfully defended their home.

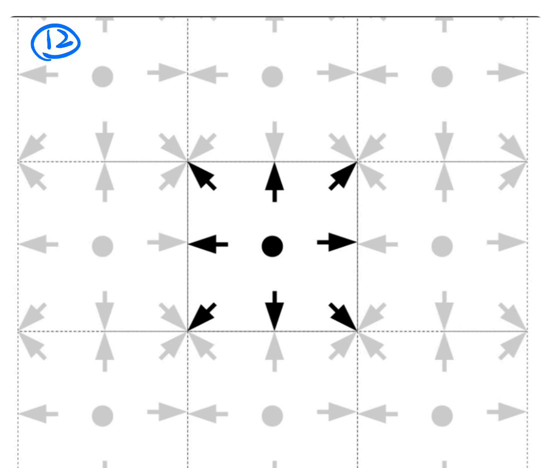


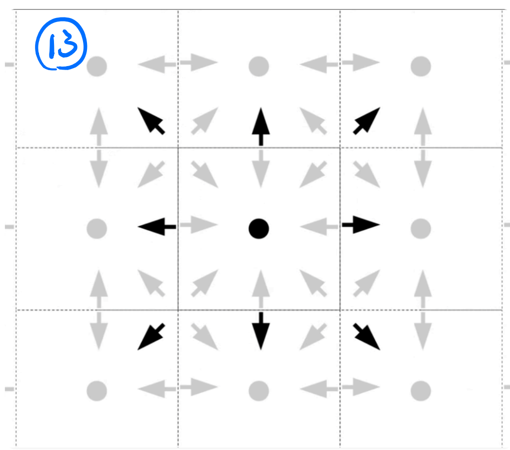


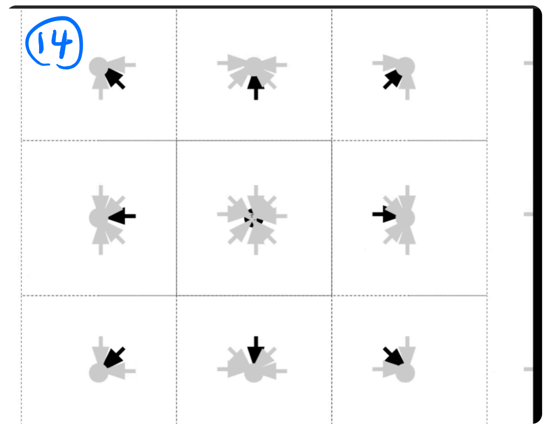
So, what does the story of foxes and attackers have to do with the lattice Boltzmann method --- LBM?

In the classical LBM, we first discretize space and turn our castle into a single Cartesian grid cell. The grid cells are typically chosen quadrantic (象限的) in 2D or cubic in 3D, but other forms exist as well. The arrows that the foxes shoot towards the attackers correspond to particles. In fact, **each arrow corresponds to a whole set of particles**, the so-called particle populations or **particle distribution functions**. In the current sketch, we restrict the motion of the particles to 9 potential moves. **Either a particle moves to the respective neighboring cell or resides in the current cell**.



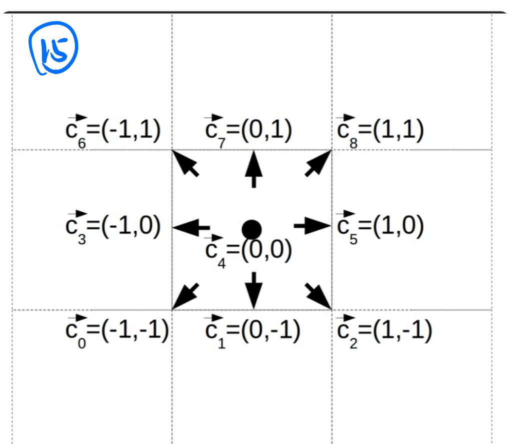


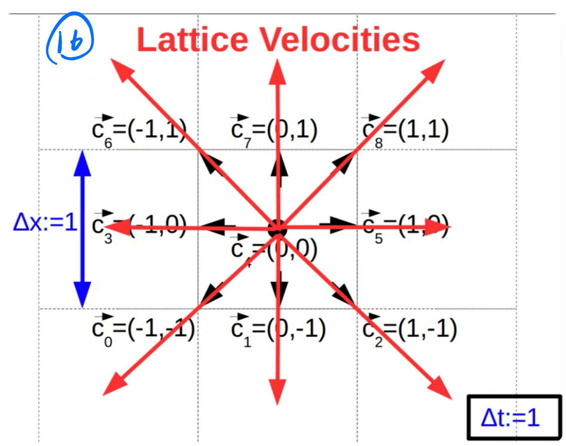




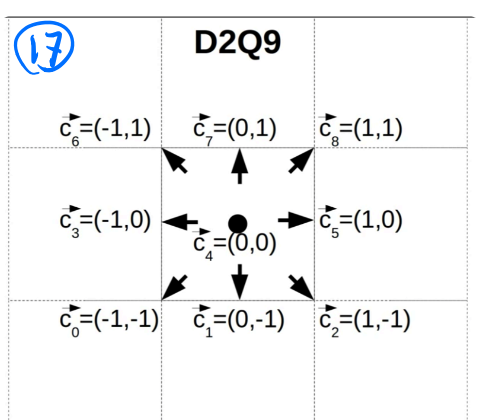
In the following, we denote these 9 directions that the particles can move into by C_0, C_1, C_2, etc.

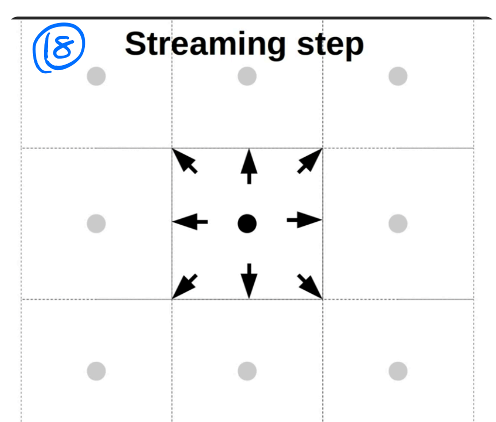
C_0 - C_8 are scaled such that for a given time interval Δt, our particles move exactly from one cell to a neighboring one. To keep things simple, we will assume that our mesh size Δx = Δt = 1. The vectors C_0 - C_8 are called lattice velocities.

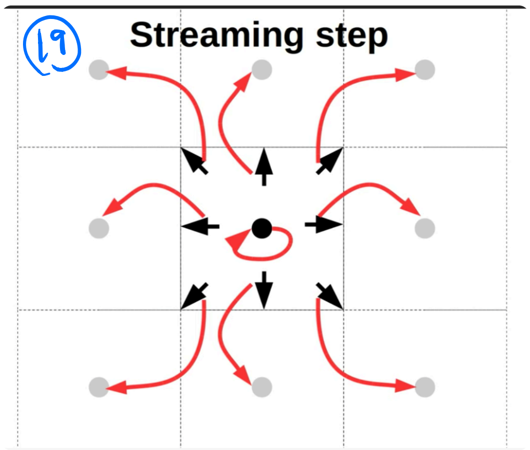


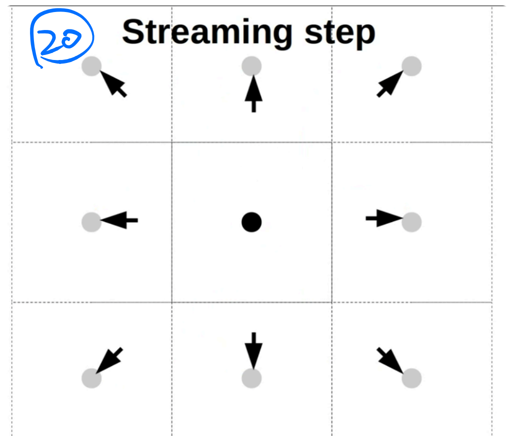


This very specific case, that is, a 2-dimensional grid with 9 velocity directions is called D2Q9 discretization. **The transport of the particle distributions to neighboring cells** is called **streaming** step in LBM. **It basically models the convective transport in the fluid flow**.

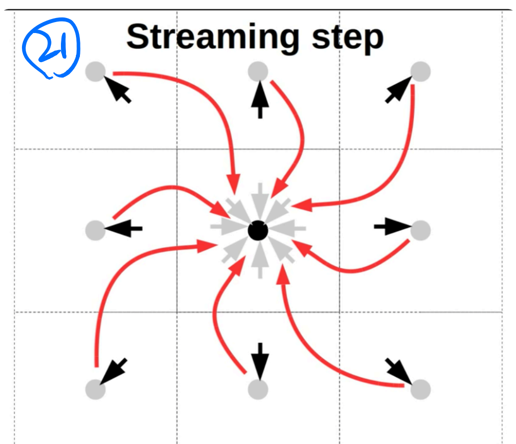


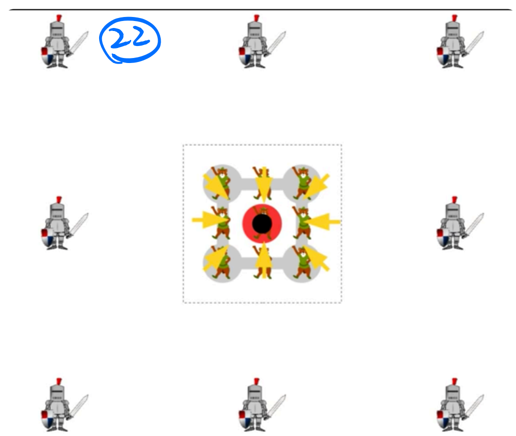




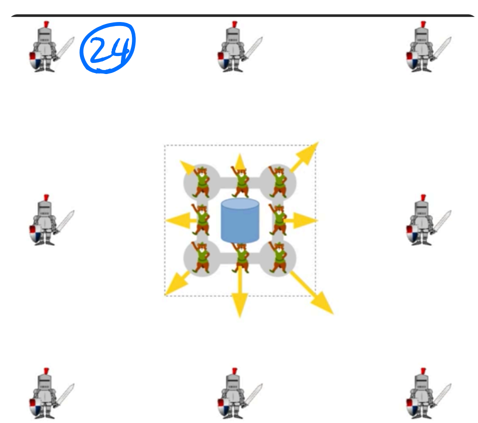


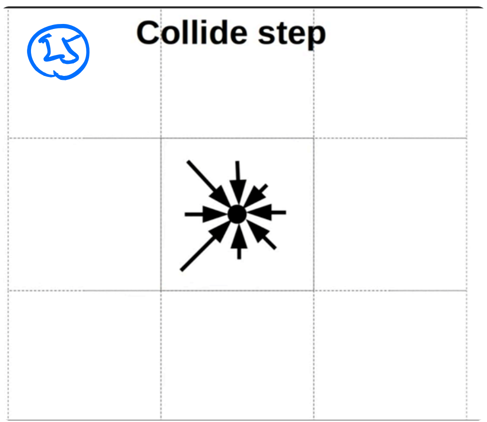
All distributions are copied to copied to the neighboring cells. This also fills the current cell with a new set of distributions.

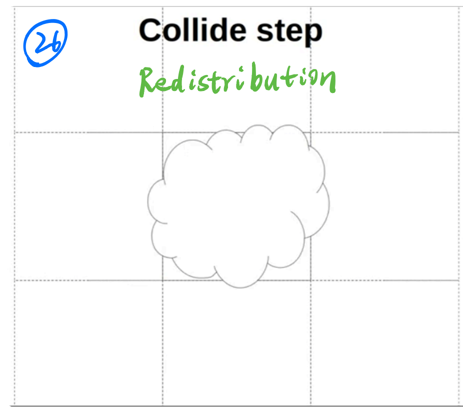




Our foxes have also already shown us the diffusive transport. Locally inside their castle, they collect the arrows in a box and rapidly redistribute the arrows among each other. We basically do the same thing in the LBM in the **collide step**.

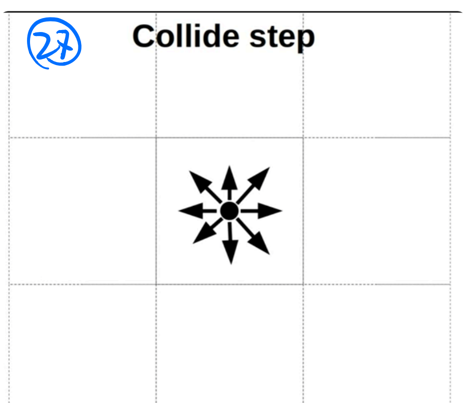


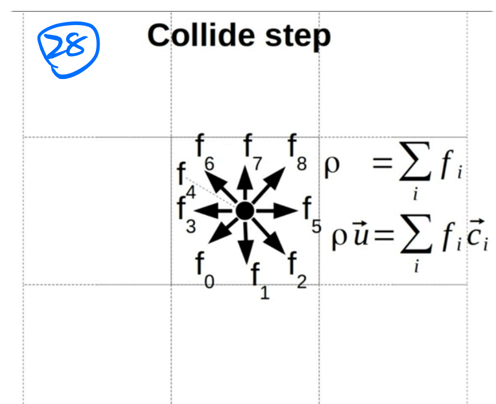




We let the particles collide locally inside each grid cell and so redistribute our particle populations. Both mass and momentum need to be conserved during collision.

If we denote each particle population with $f_0, f_1, f_2 \dots f_8$, we can compute the mass per grid cell, ρ, from the sum of the distributions. The momentum arises from the sum of the terms $f_i * C_i$.





Collide step: BGK model

29

$$\rho = \sum_i f_i$$

$$\rho \vec{u} = \sum_i f_i \vec{c}_i$$

$$w_{0,2,6,8} = \frac{1}{36} \quad w_{1,3,5,7} = \frac{1}{9} \quad w_4 = \frac{4}{9} \quad c_s = \frac{1}{\sqrt{3}}$$

Speed of sound

$$\|\vec{u}\| \ll c_s$$

$$f_i^{eq} = w_i \rho \left(1 + \frac{\vec{c}_i \cdot \vec{u}}{c_s^2} + \frac{(\vec{c}_i \cdot \vec{u})^2}{2 c_s^4} - \frac{\vec{u} \cdot \vec{u}}{2 c_s^2}\right)$$

$\frac{1}{\tau}$: collision frequency ; τ : relaxation time

$$f_i := f_i - \frac{1}{\tau}\left(f_i - f_i^{eq}\right) \quad \tau \in (0.5, 2)$$

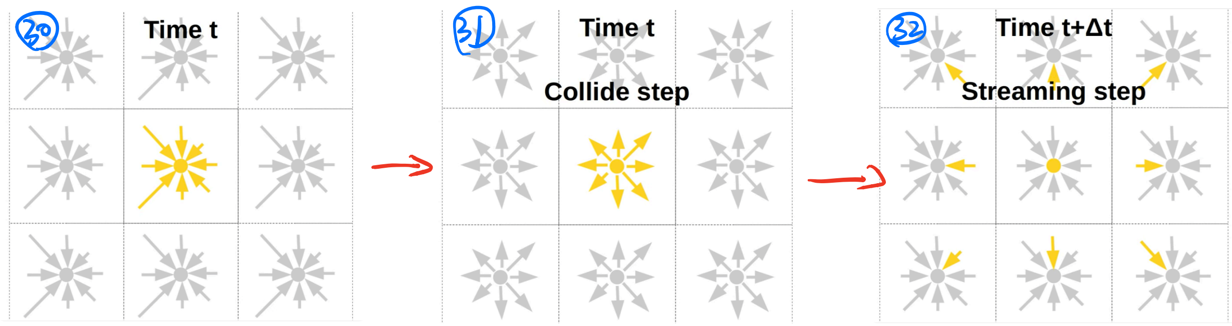
Now, how do we model the collision process? The most common collision model is given by the so-called Bhatnagar-Gross-Krook (BGK) model.

First, we compute mass and momentum inside a single grid cell from the distributions that have just entered our cells. Then we compute the equilibrium distribution fi_eq.

Computing mass and momentum from the equilibrium distributions returns the same values for ρ and u as if we used f_0 - f_8. This is basically due to a clever choice of our lattice weights W_i. You may just do the computation by hand and check out the isotropic structure of lattice velocities and weights.

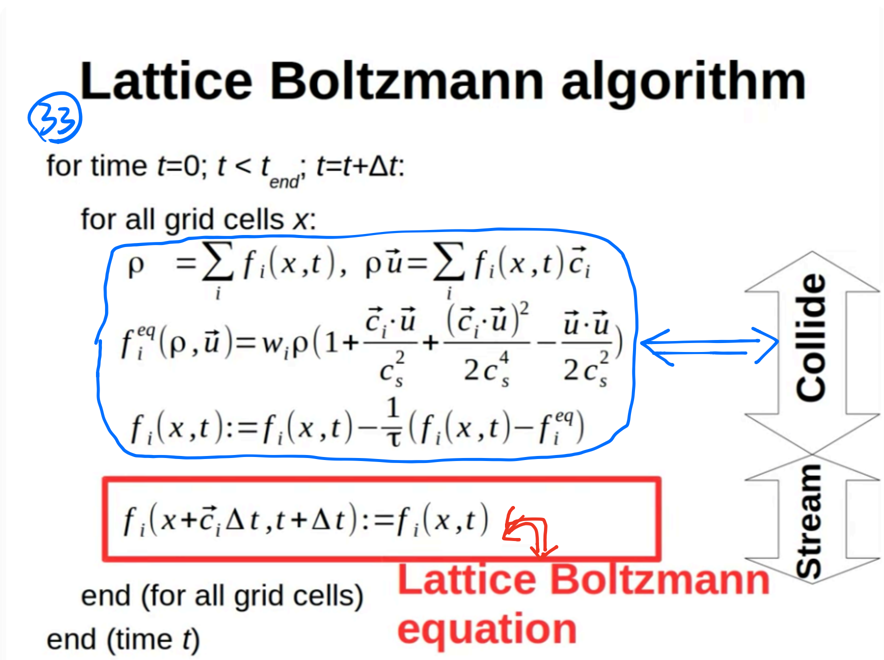
The expression of the equilibrium function can be derived from the Maxwell-Boltzmann distribution in the low velocity limit. That means for flow velocities $\boldsymbol{u}$, which are much smaller than the speed of sound Cs.

Finally, we **relax** our distributions **f_i towards** the **equilibrium distribution** at a given collision frequency $1/\tau$. τ is chosen such that the correct viscosity of the considered fluid is obtained. For $\tau = 1$, the relaxation process sets the distributions f_i exactly to the corresponding counterparts f_i_eq of the equilibrium distribution. That means that our BGK model basically pushes our distributions f_i towards the equilibrium state. Due to the numerical stability, τ needs to be in the range (0.5, 2).

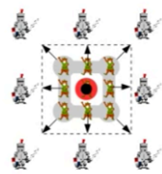


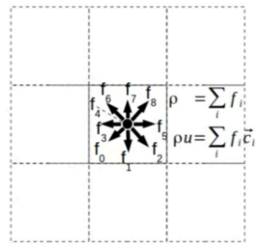
Now, let's put the collide step and streaming step together. Starting from time t, the particle distributions first collide locally in each cell, resembling the collide step. Afterwards, the collided distributions are streamed to the neighboring cells. Yellow distributions leave the center cell and gray distributions from neighboring cells enter the center cell. These gray-colored distributions and yellow distributions for the rest population now correspond to the particle distributions at time t+Δt. We can finally sketch the lattice Boltzmann algorithm.

For a given number of time steps, we loop through all grid cells and execute the collide step in each cell. Afterwards, we carry out the streaming step to obtain the new distribution values at the next times step. The highlighted update rule for the distributions f_i is know as the Lattice Boltzmann Equation BGK form.



34





Castle invasion	Lattice Boltzmann
Foxes/attackers	Particle distributions f_i
Directions for shooting arrows	Lattice velocities $\vec{c}_i$
Re-distribution of arrows → conserve number of arrows	Collide step → conserve mass, momentum
Shooting arrows	Streaming step

$$f_i(x+\vec{c}_i \Delta t, t+\Delta t) := f_i(x,t) - \frac{1}{\tau}\left(f_i(x,t) - f_i^{eq}\right)$$