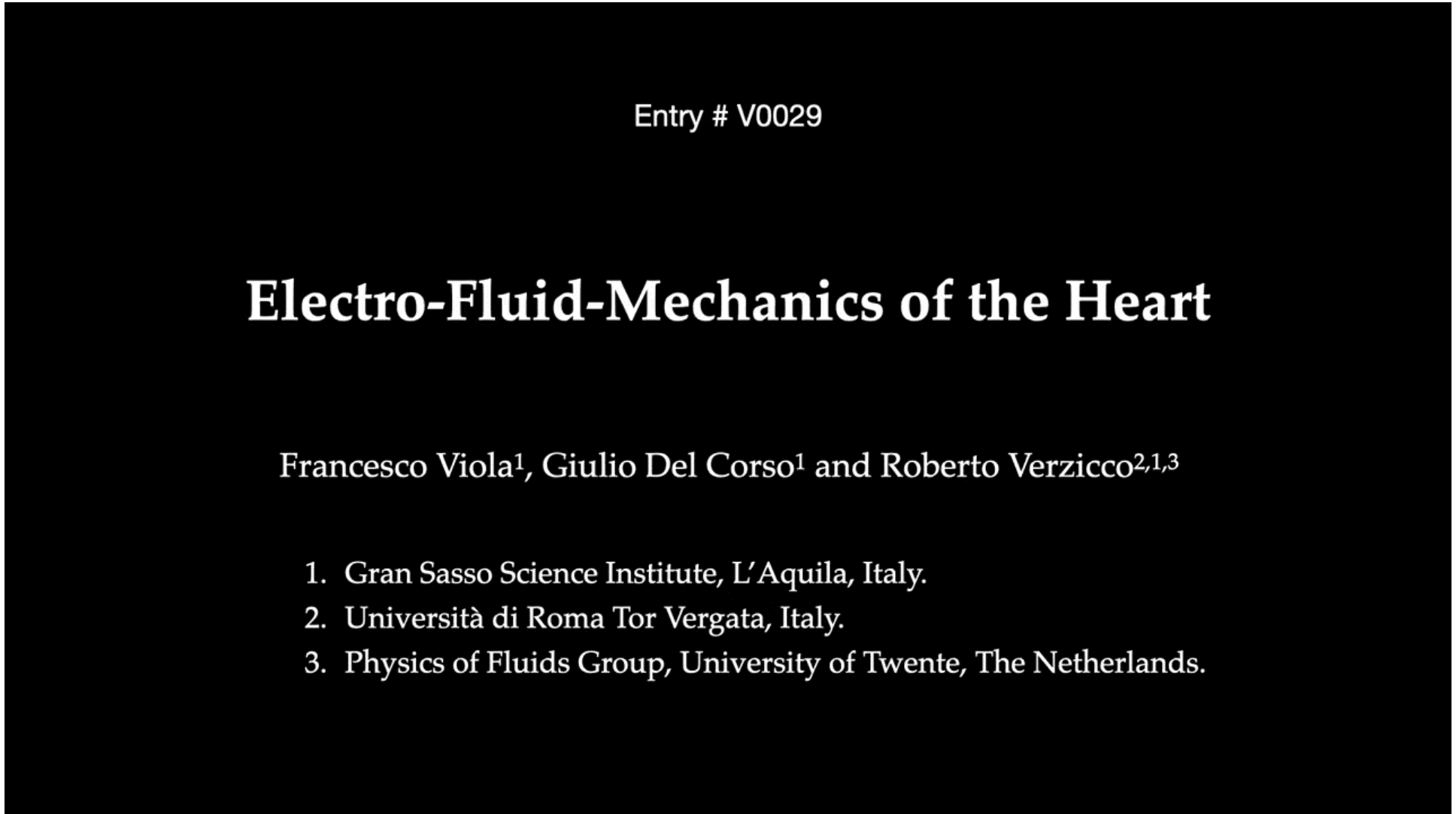
ME 7310 Lecture 21

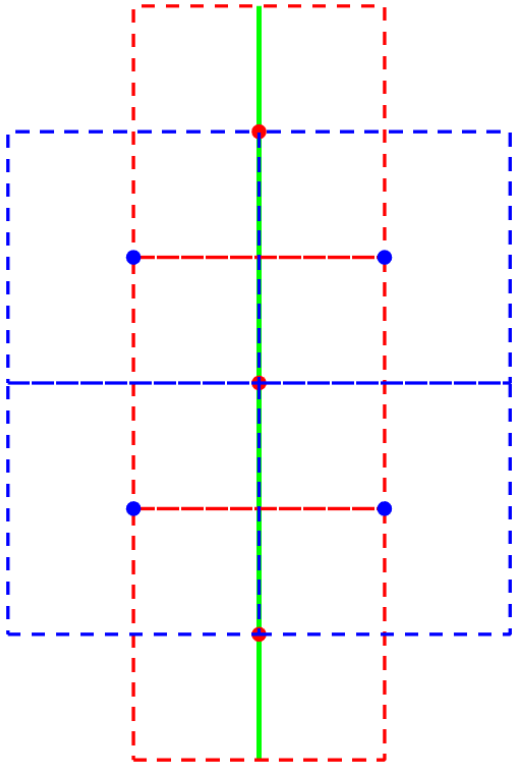
Professor Allshouse

4/2/2024



Inlet boundary conditions

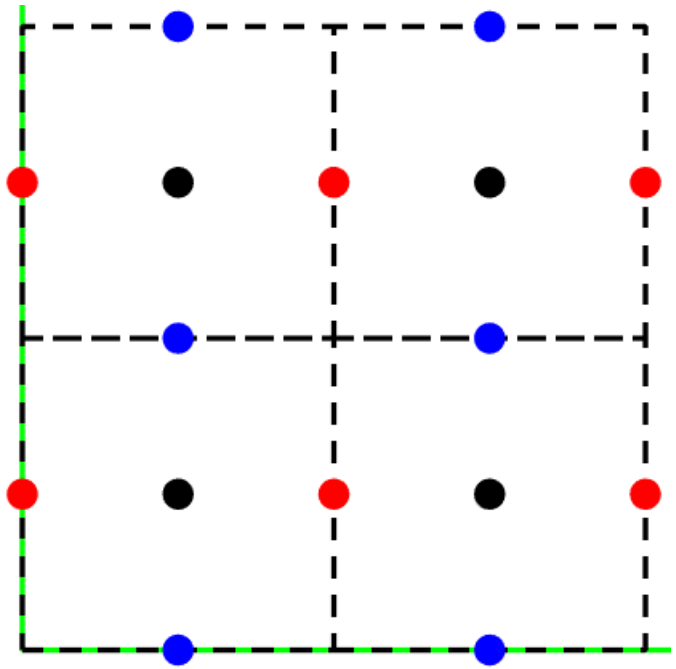
- Two types of inlet conditions: velocity based and pressure based
- Velocity based: both velocity components are set ON the boundary
  - Ghost cells could be used for tangential component
- Pressure based still requires a velocity direction (usually normal to the wall)



Outlet boundary conditions

- If there is an inlet there needs to also be an outlet
- Outlet boundary condition impacts local dynamics and should be set well down stream of the region of interest.
- Fully developed outlet boundary
  - Fully developed flow assumes there are no gradients in the streamwise direction
  - Generally this is taken as no gradients in the wall normal direction.
- Specified outlet pressure
  - Velocity still has zero gradient in the streamwise direction
  - Pressure requires a ghost cell to set the value ON the boundary

Fully developed outlet boundary conditions



Inlet

```
elseif u_type(i) == 2

    % Inlet boundary condition
    A_u(i,i) = 1;
    b_u(i) = u_lef;
```

```
%% If the cell is Neighboring the left boundary
elseif v_type(i) == 3
    A_v(i,i)      =  Fe_v(i) +Fn_v(i)-Fs_v(i)+De+2*Dw+Dn+Ds;
    A_v(i,i+1)    =  Fn_v(i)-Dn;
    A_v(i,i-1)    = -Fs_v(i)-Ds;
    A_v(i,i+Ny_v) =  Fe_v(i)-De;
    b_v(i) = -(p_n(i)-p_s(i))/dy + 2*(Fw_v(i)+Dw)*v_lef;
```

```
% Boundaries on the west face
elseif p_type(i) == 3
    A_p(i,i) = Ce(i) + Cn(i) + Cs(i);
    A_p(i,i+Ny_p) = -Ce(i);
    A_p(i,i+1) = -Cn(i);
    A_p(i,i-1) = -Cs(i);
```

```
%% Calculate the velocity correction for all cells IN the domain

    u_correction = 0*u_guess;
    u_correction(:,2:end-1) = (pc_W-pc_E)/dx./Ap_u(:,2:end-1);

    u_correction(u_type==-1) = 0;
```

Outlet

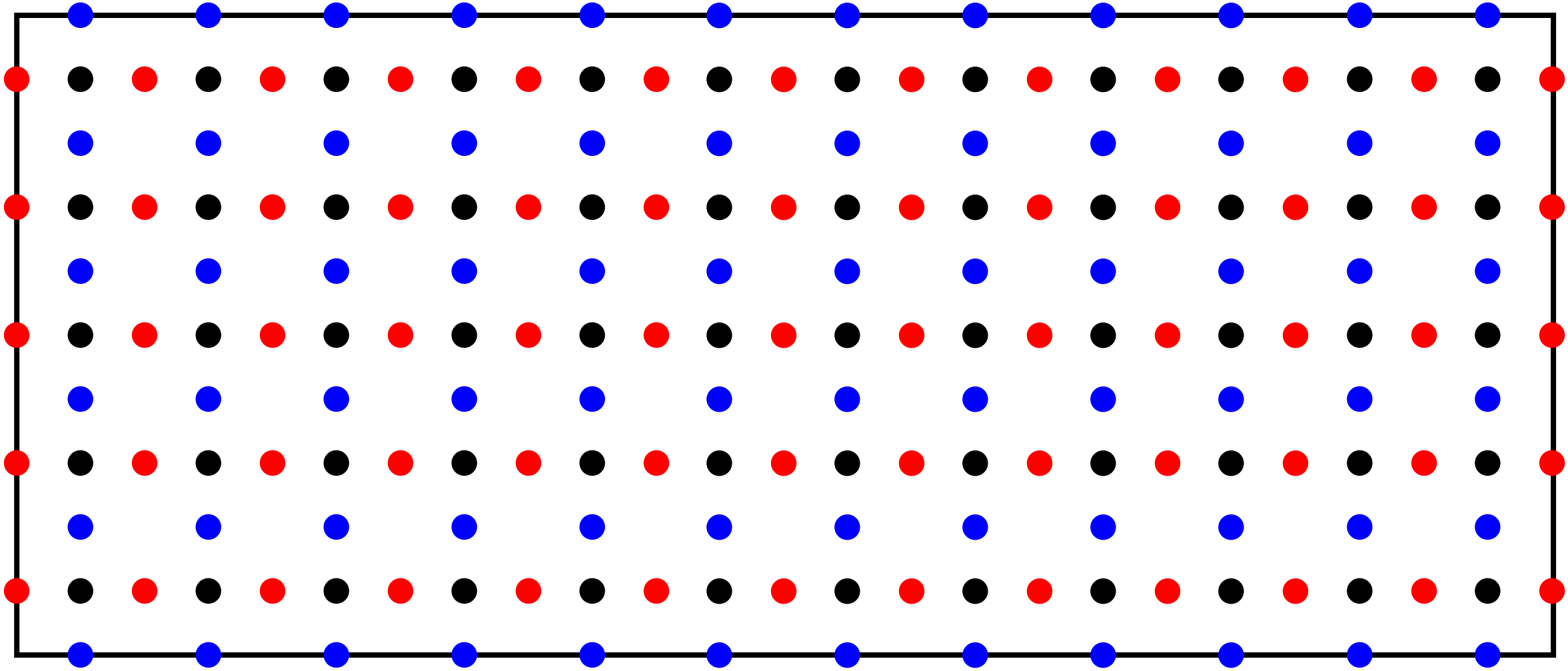
```
%% If the cell is ON the right boundary
if u_type(i) == 1

    % Outlet boundary condition
    A_u(i,i) = 1;
    A_u(i,i-Ny_u) = -1;
    b_u(i) = 0;
```

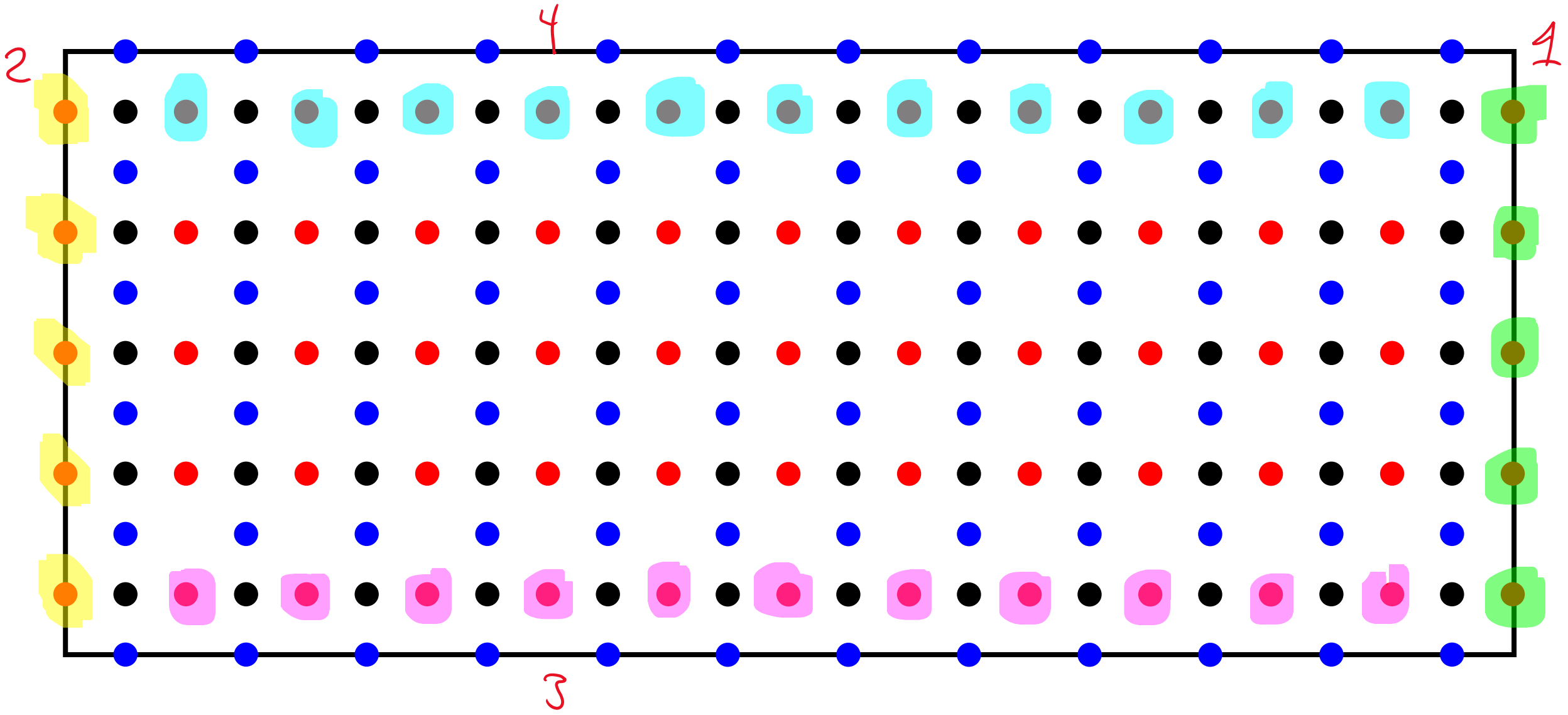
```
%% If the cell is Neighboring the right boundary
elseif v_type(i) == 4
    A_v(i,i)      =  2*Fe_v(i)-Fw_v(i)+Fn_v(i)-Fs_v(i)+Dw+Dn+Ds;
    A_v(i,i+1)    =  Fn_v(i)-Dn;
    A_v(i,i-1)    = -Fs_v(i)-Ds;
    A_v(i,i-Ny_v) = -Fw_v(i)-Dw;
    b_v(i) = -(p_n(i)-p_s(i))/dy;
```

```
% Boundaries on the east face
elseif p_type(i) == 6
    A_p(i,i) = Cw(i) + Cn(i) + Cs(i);
    A_p(i,i+1) = -Cn(i);
    A_p(i,i-1) = -Cs(i);
    A_p(i,i-Ny_p) = -Cw(i);
```

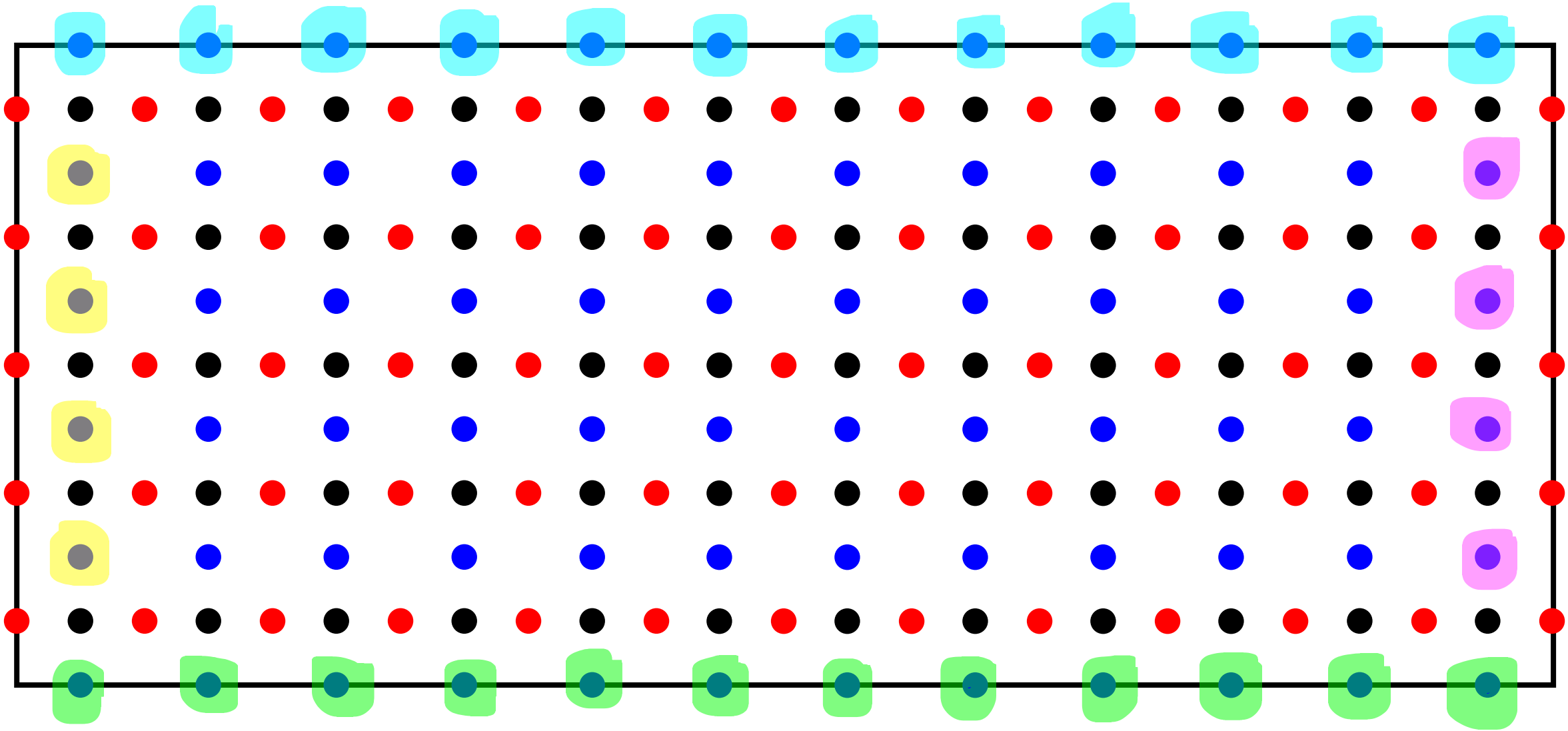
```
u_guess = u_star + u_correction;

v_guess = v_star + v_correction;

p_guess = p_guess + alpha_p*p_correction;
```

```
%% Impose flux conservation

m_in = sum(u_guess(:,1)*dy);
m_out = sum(u_guess(:,end-1)*dy);

u_guess(:,end) = m_in/m_out*u_guess(:,end-1);
```

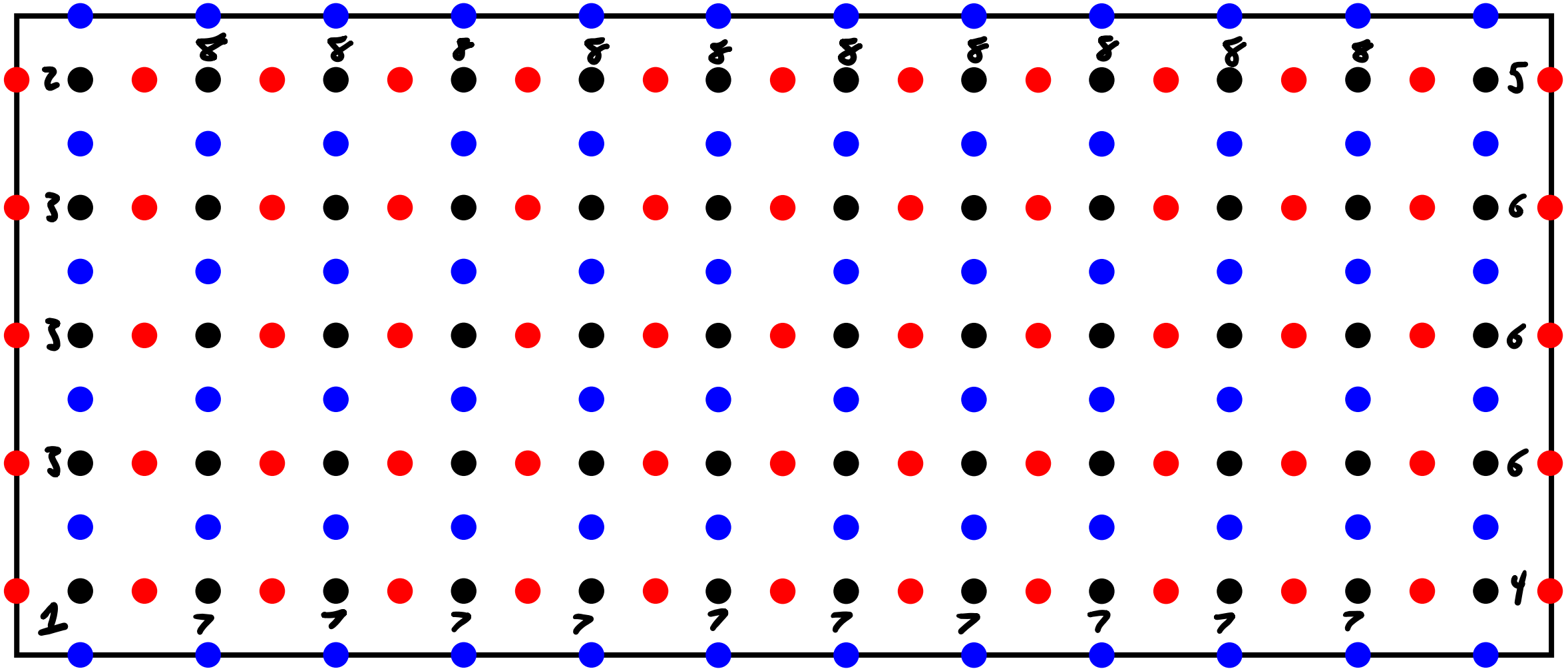


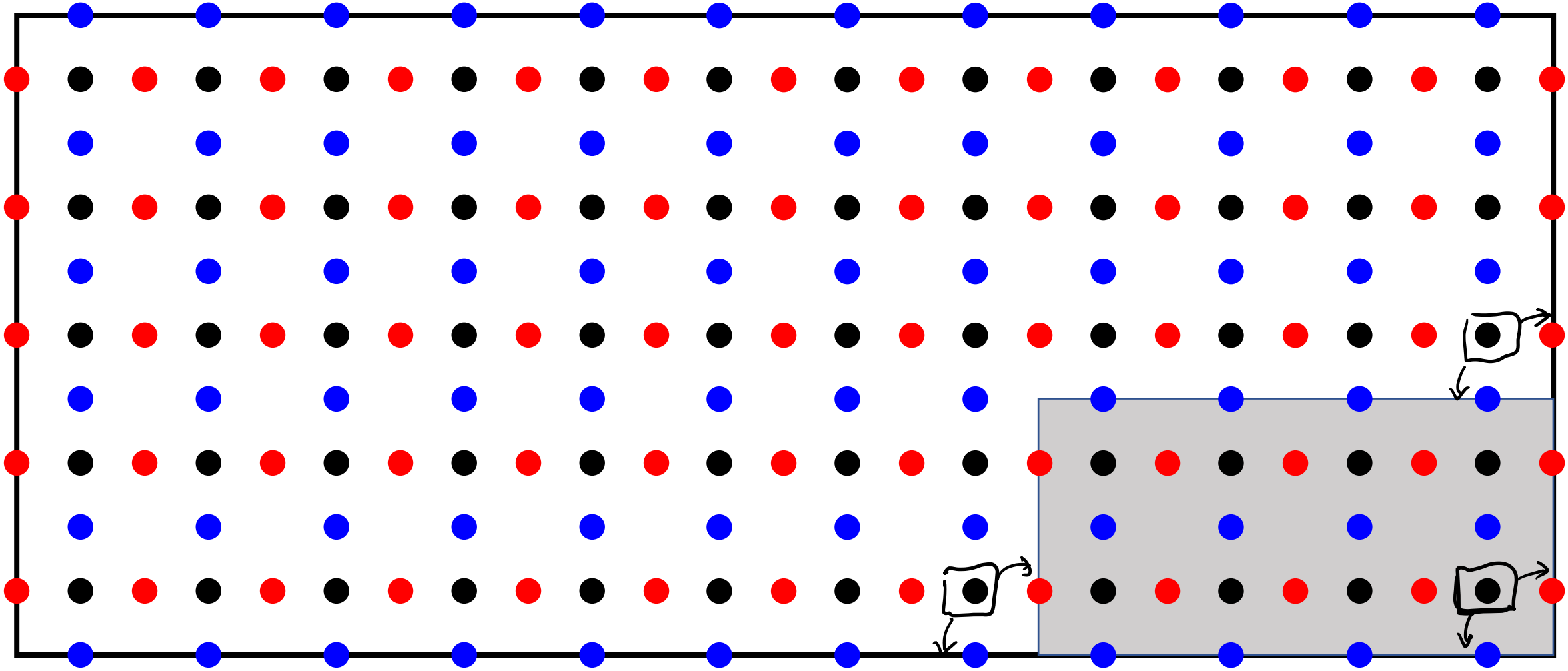


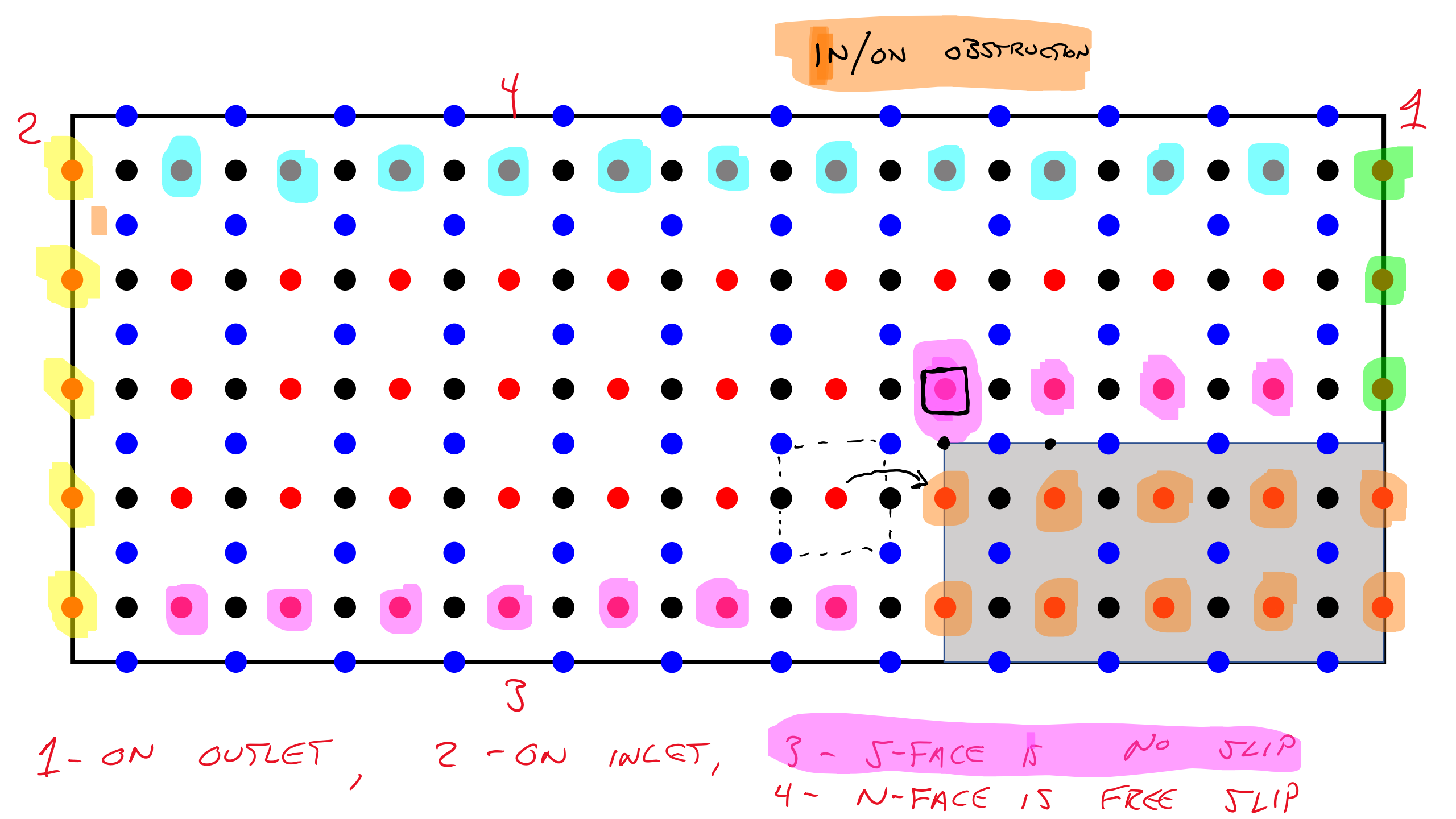
1 - ON OUTLET, 2 - ON INLET, 3 - S-FACE IS NO SLIP
4 - N-FACE IS FREE SLIP

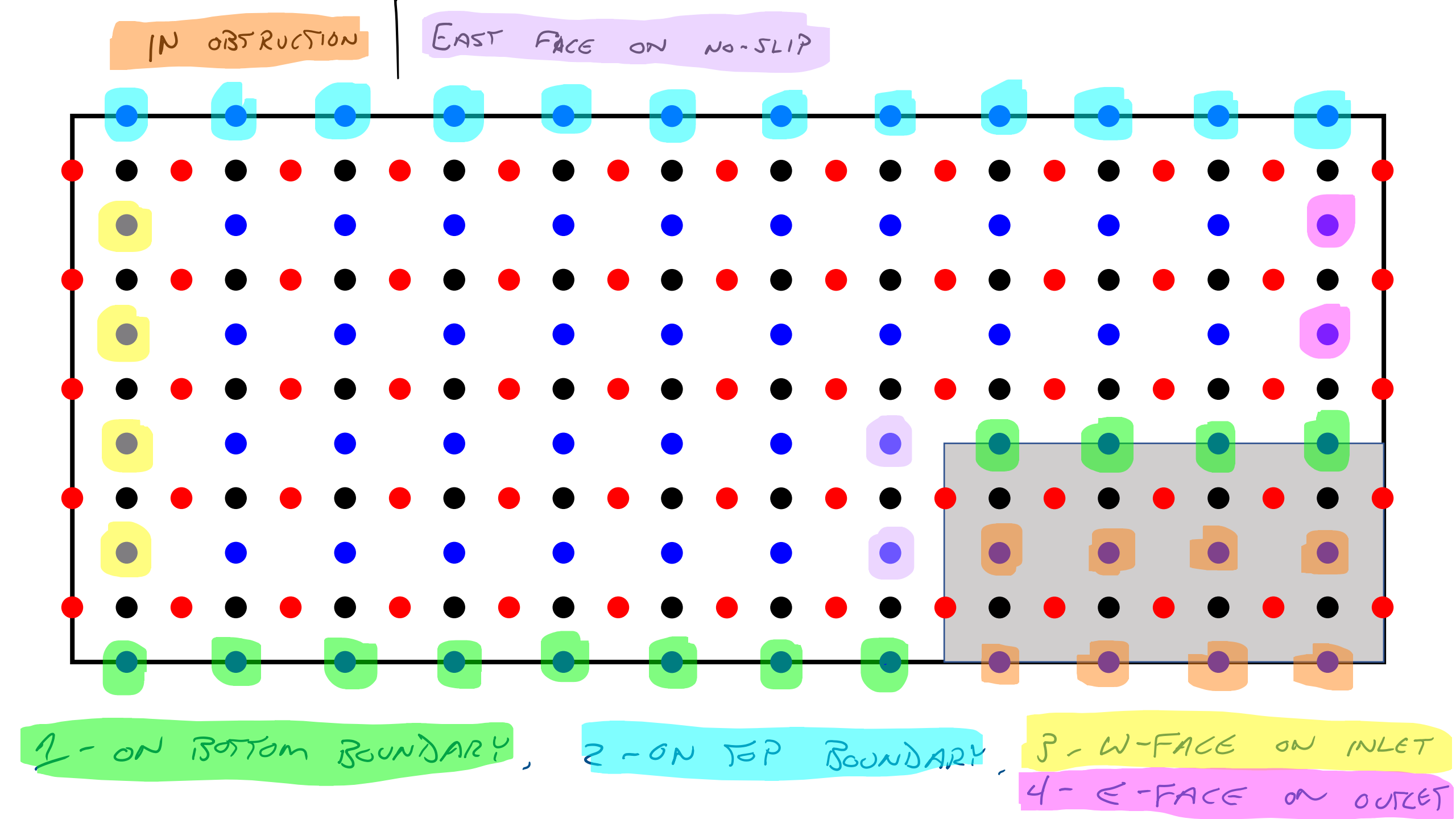


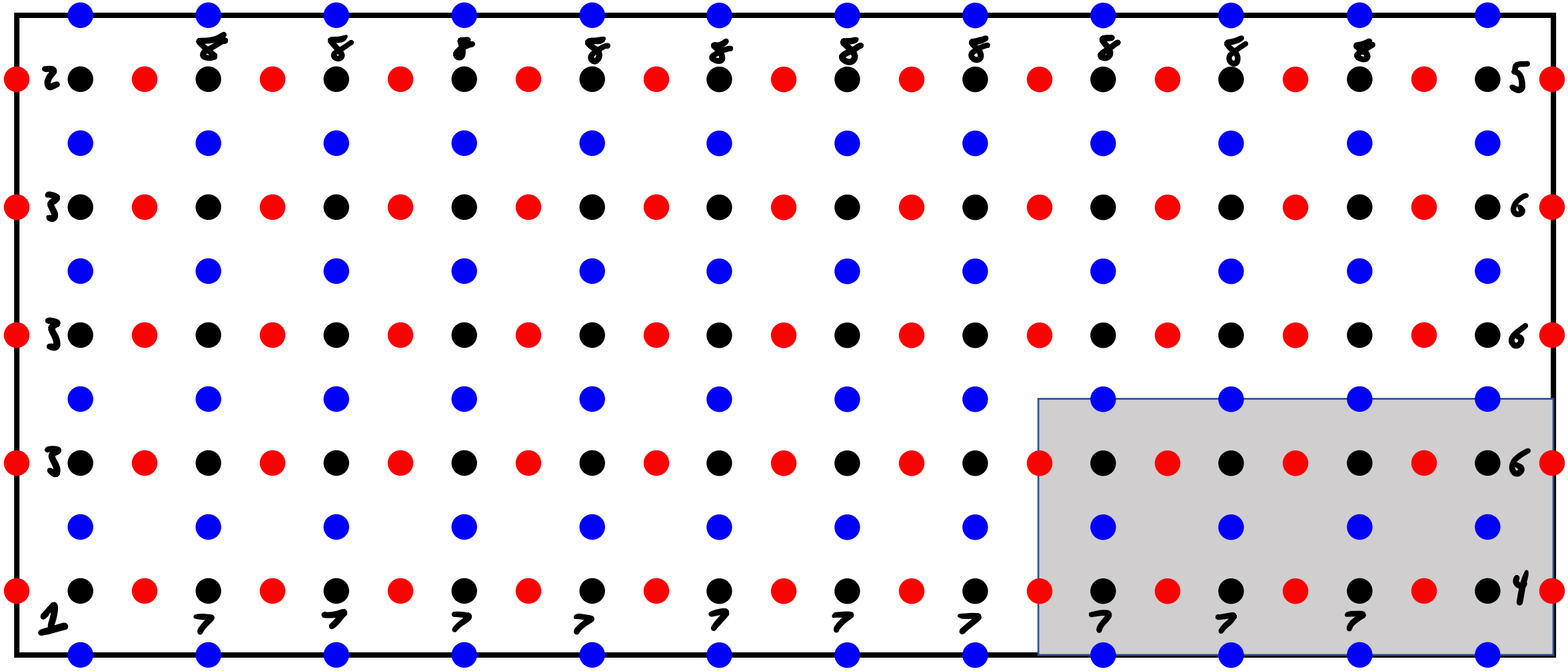
1 - ON BOTTOM BOUNDARY, 2 - ON TOP BOUNDARY, 3 - W-FACE ON INLET
4 - E-FACE ON OUTLET






IN/ON OBSTRUCTION
1
2
3
4
1 - ON OUTLET, 2 - ON INLET,
3 - S-FACE IS NO SLIP
4 - N-FACE IS FREE SLIP


IN OBSTRUCTION
EAST FACE ON NO-SLIP
1 - ON BOTTOM BOUNDARY
2 - ON TOP BOUNDARY
3 - W-FACE ON INLET
4 - E-FACE ON OUTLET



Changes to get to blocked flow

- We need to update the cell types to incorporate the cell
- Apply appropriate boundary condition to those cells

Types of cells

- U-types
  - Centers on Inlet
  - Center on Outlet
  - North on free-slip
  - South on no-slip
  - (Interior)

  - ~~On the front of the obstruction~~
  - Cell centers In/on the obstruction
  - ~~Cells above obstruction~~

- V-types
  - Centers on bottom (no-slip)
  - Centers on top (free-slip)
  - West face on inlet
  - East face on outlet

  - In the obstruction
  - East face on no-slip boundary

- P-types
- Boundary on
  - South and West
  - North and West
  - West
  - South and East
  - North and East
  - East
  - South
  - North

  - In the obstruction