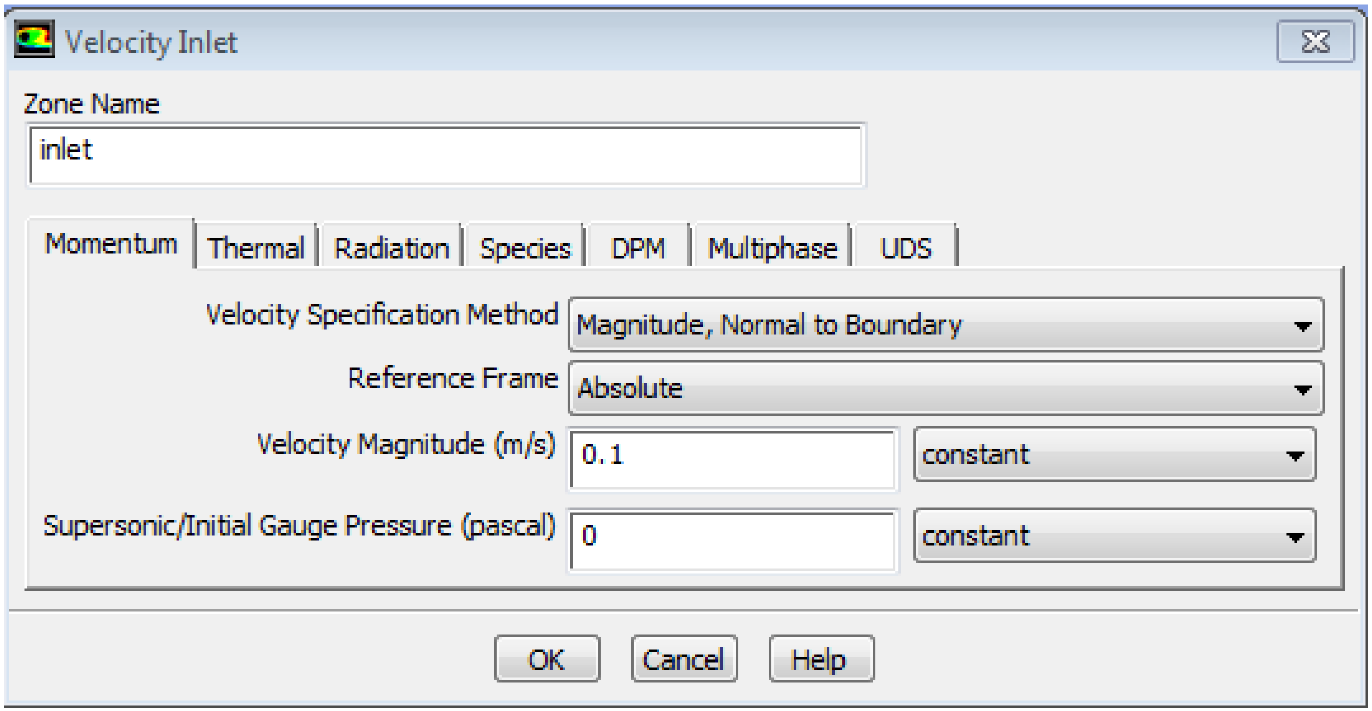
Boundary Conditions

Available BC types in Fluent

- Inlet
  - Velocity inlet
  - Pressure inlet
  - Mass flow inlet
  - Inlet vent
  - Intake fan
- Exit
  - Pressure outlet
  - Pressure far-field
  - Outflow
  - Exhaust fan
- Wall
  - Symmetric
  - Axisymmetric
  - Periodic

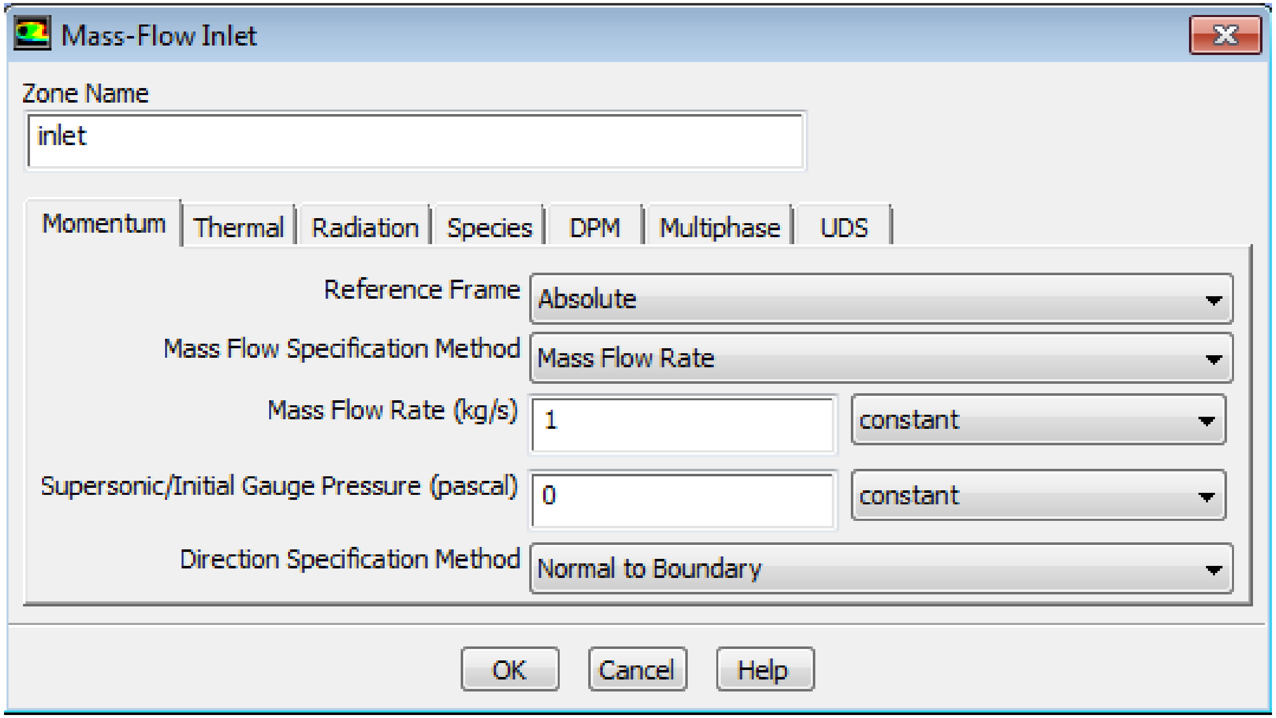
Velocity inlet

- Define the velocity and scalar properties of the flow at the boundary



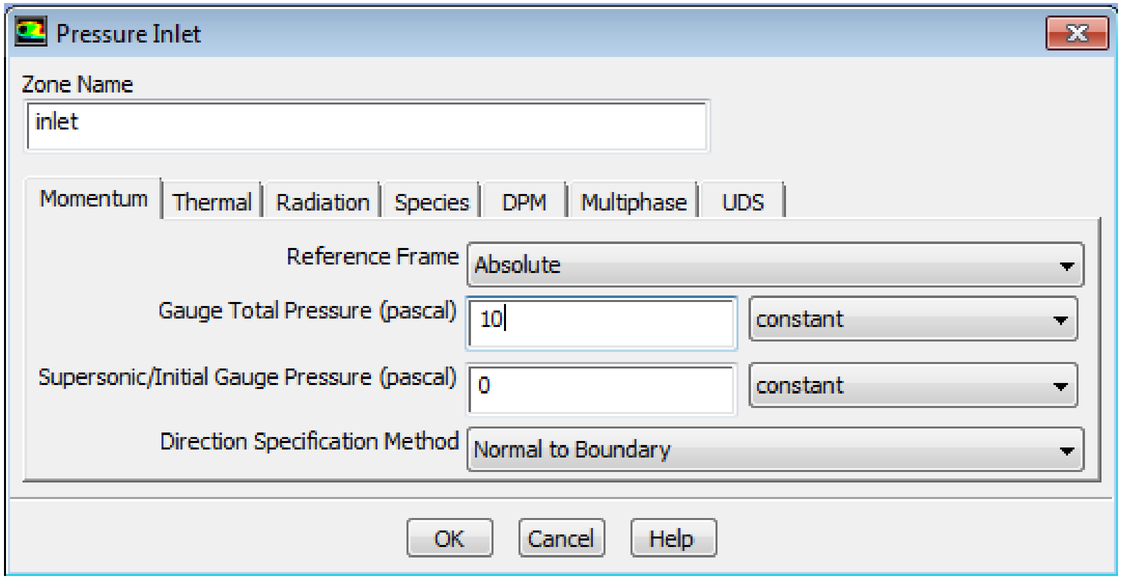
Mass flow inlet

- Define the mass flow rate for compressible flows at the boundary



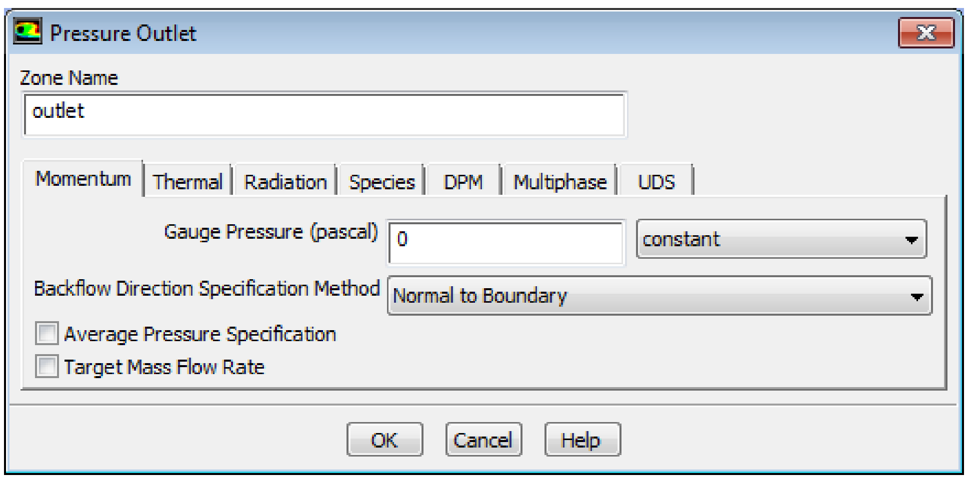
Pressure inlet

- Define the total pressure and other scalar properties at the boundary
- $p_{\text{total}} = p_{\text{static}} + 0.5\rho v^2$



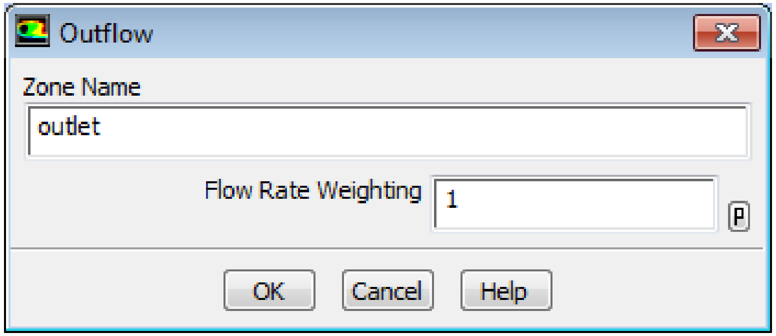
Pressure outlet

- Define the static pressure (and also other scalar variables, in case of backflow) at the boundary



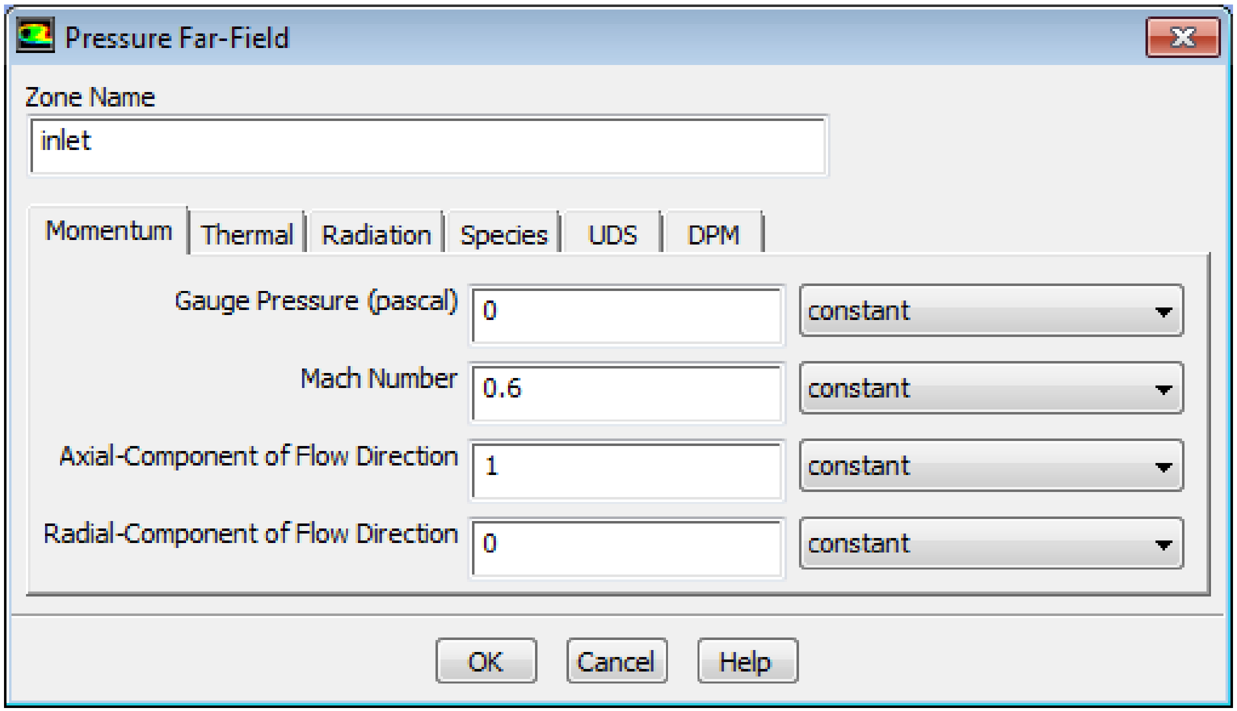
Outflow

- Assume zero streamwise gradient for all flow variables except pressure
- Not appropriate for compressible flows
- Cannot use with pressure inlet



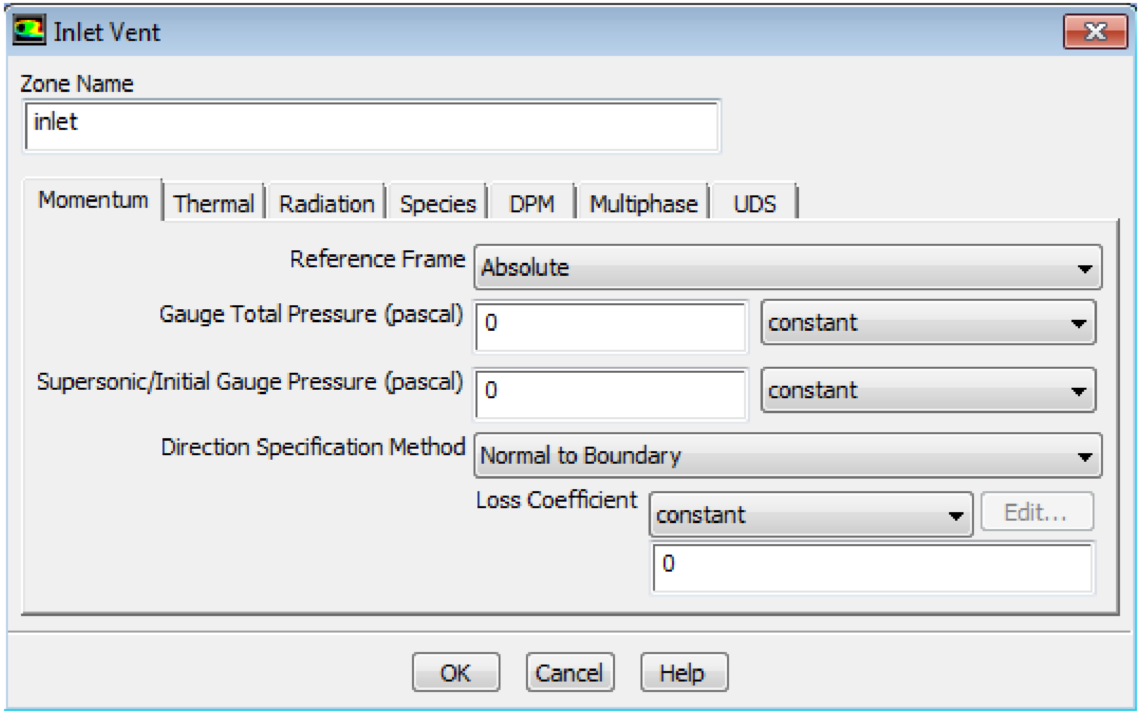
Pressure far-field

- Model a free-stream compressible flow at infinity



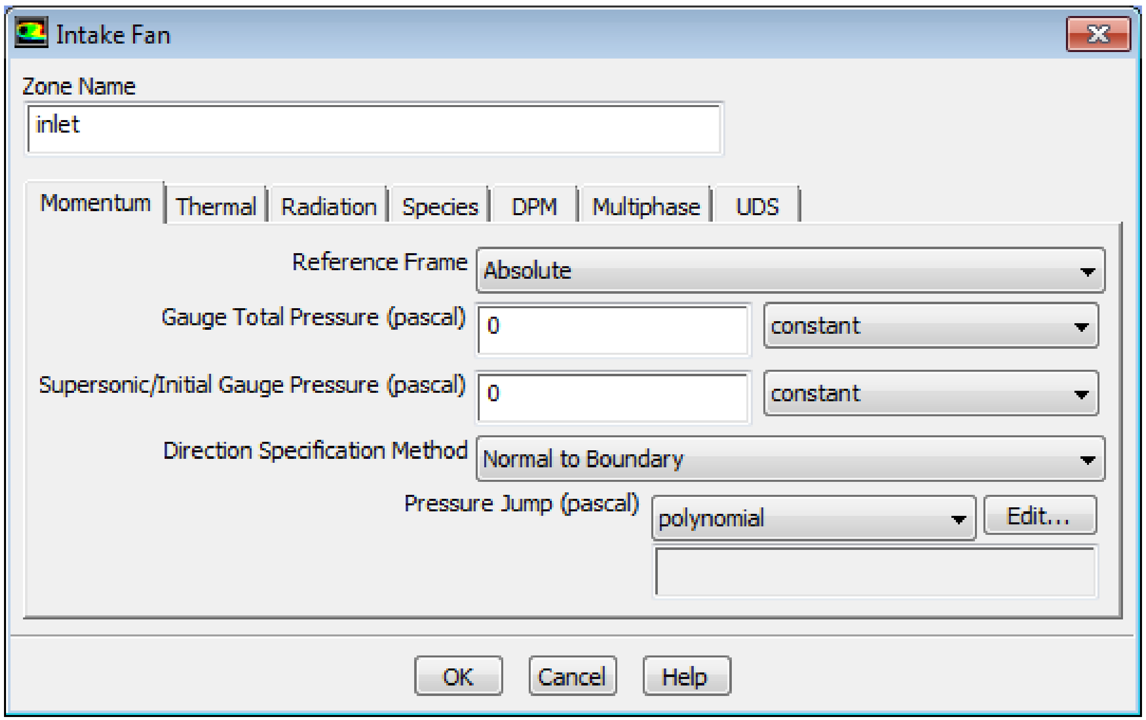
Inlet/outlet vent

- Model an inlet/outlet vent with a specified loss coefficient



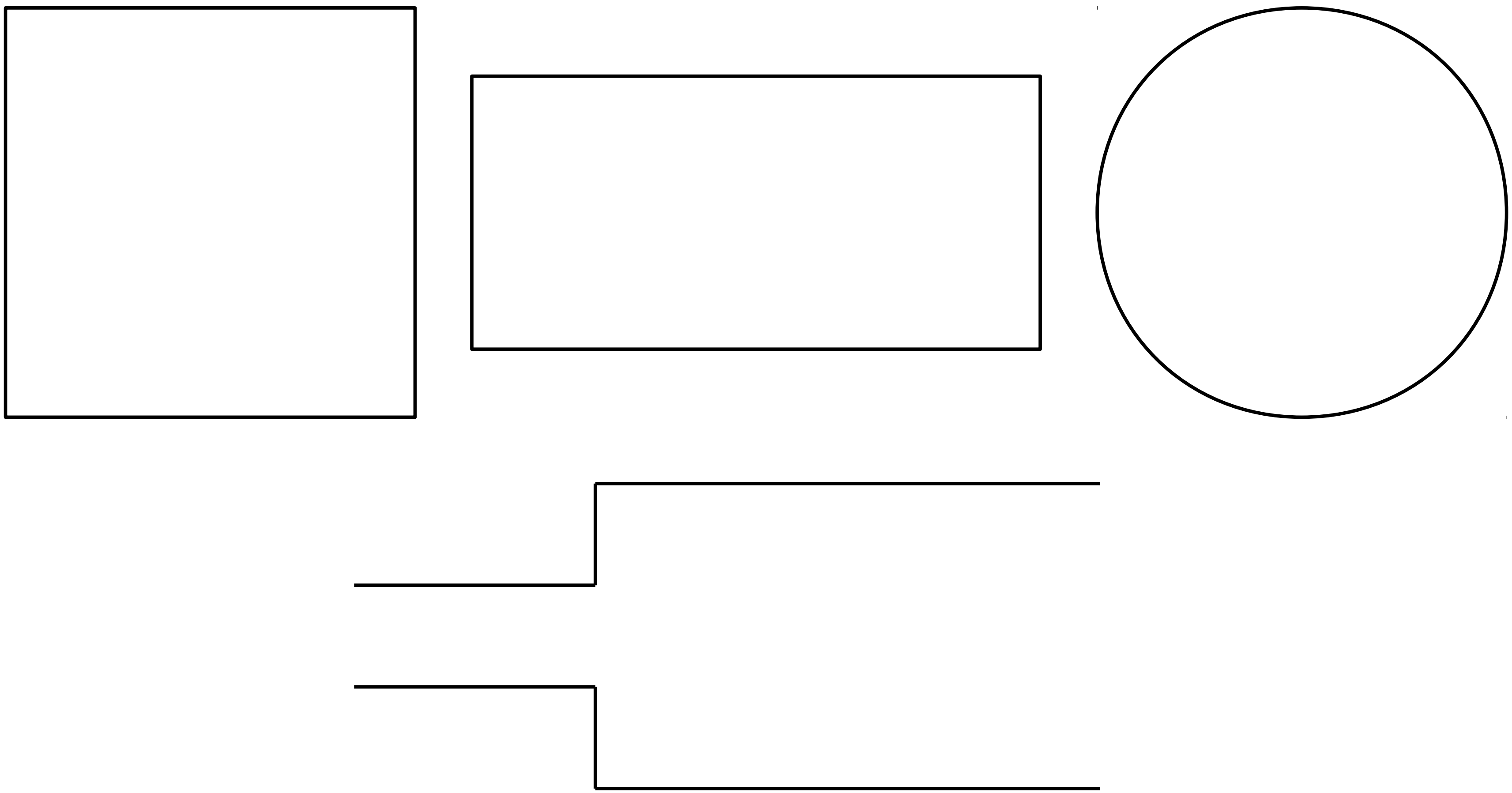
Intake/exhaust fan

- Model an external intake/exhaust fan with a specified pressure jump

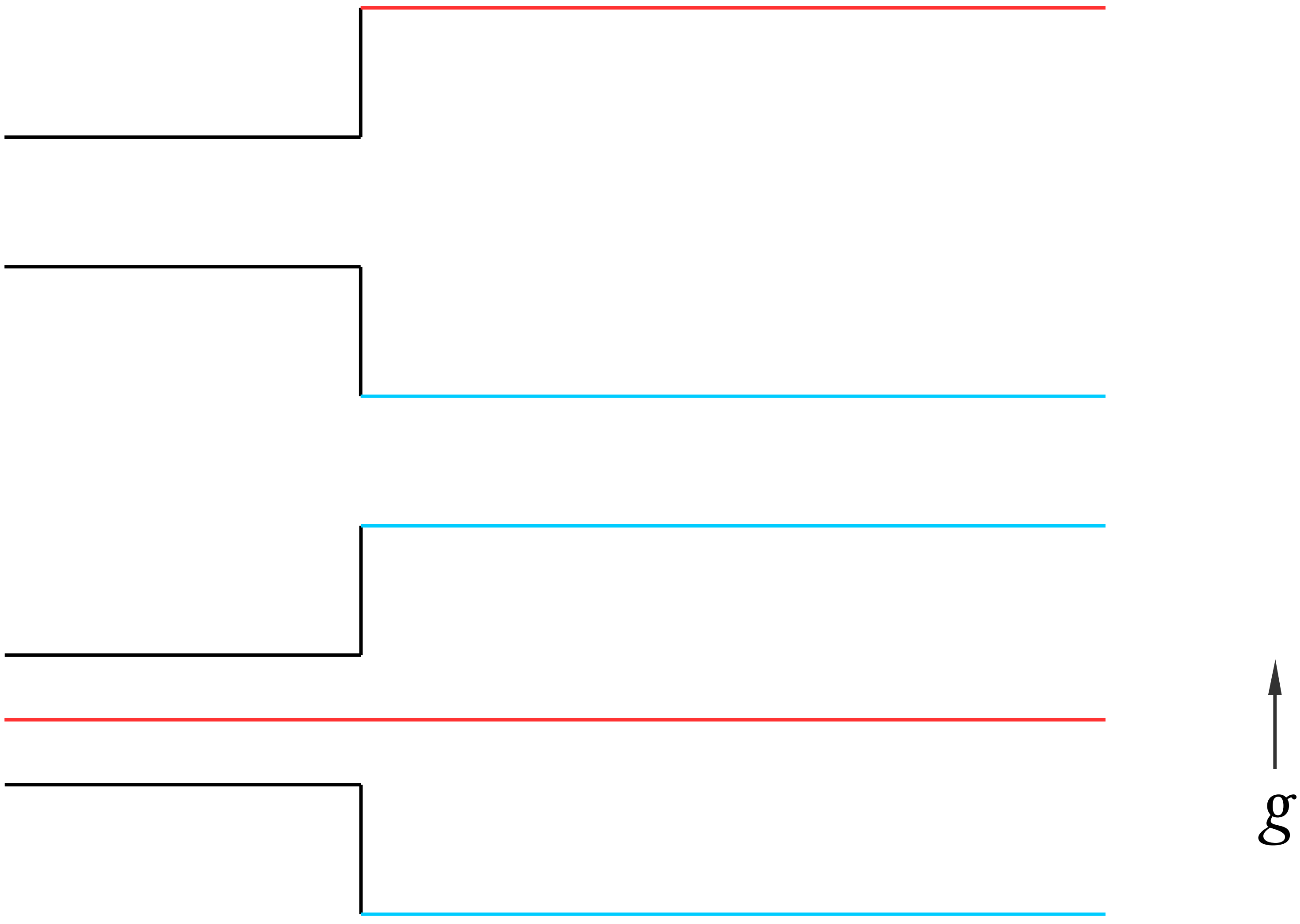


Symmetry BC

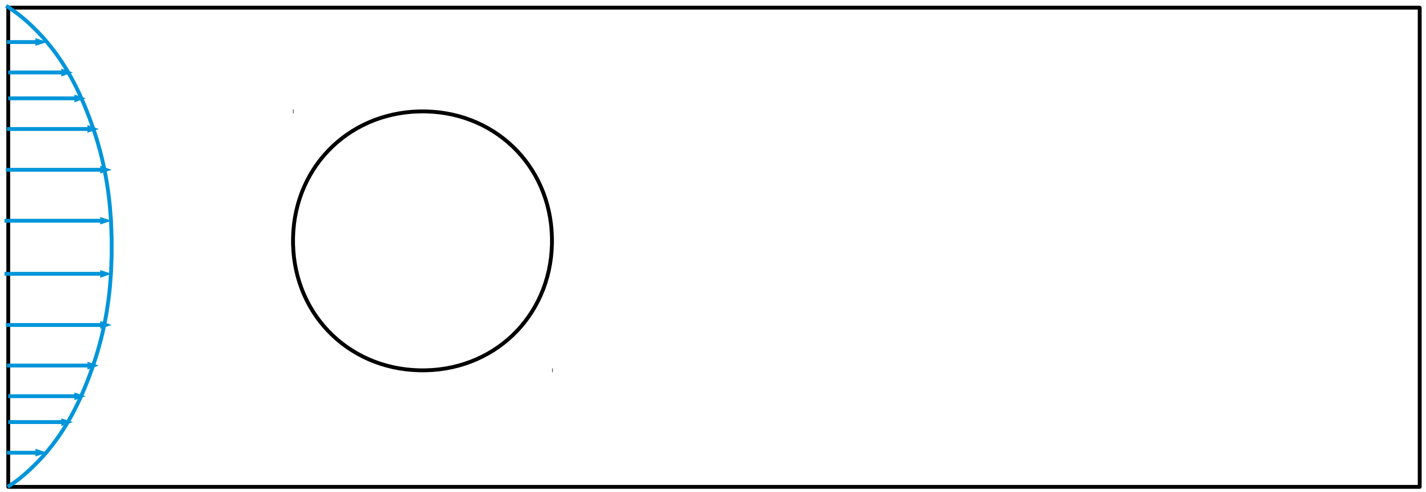
Are these domains symmetric?



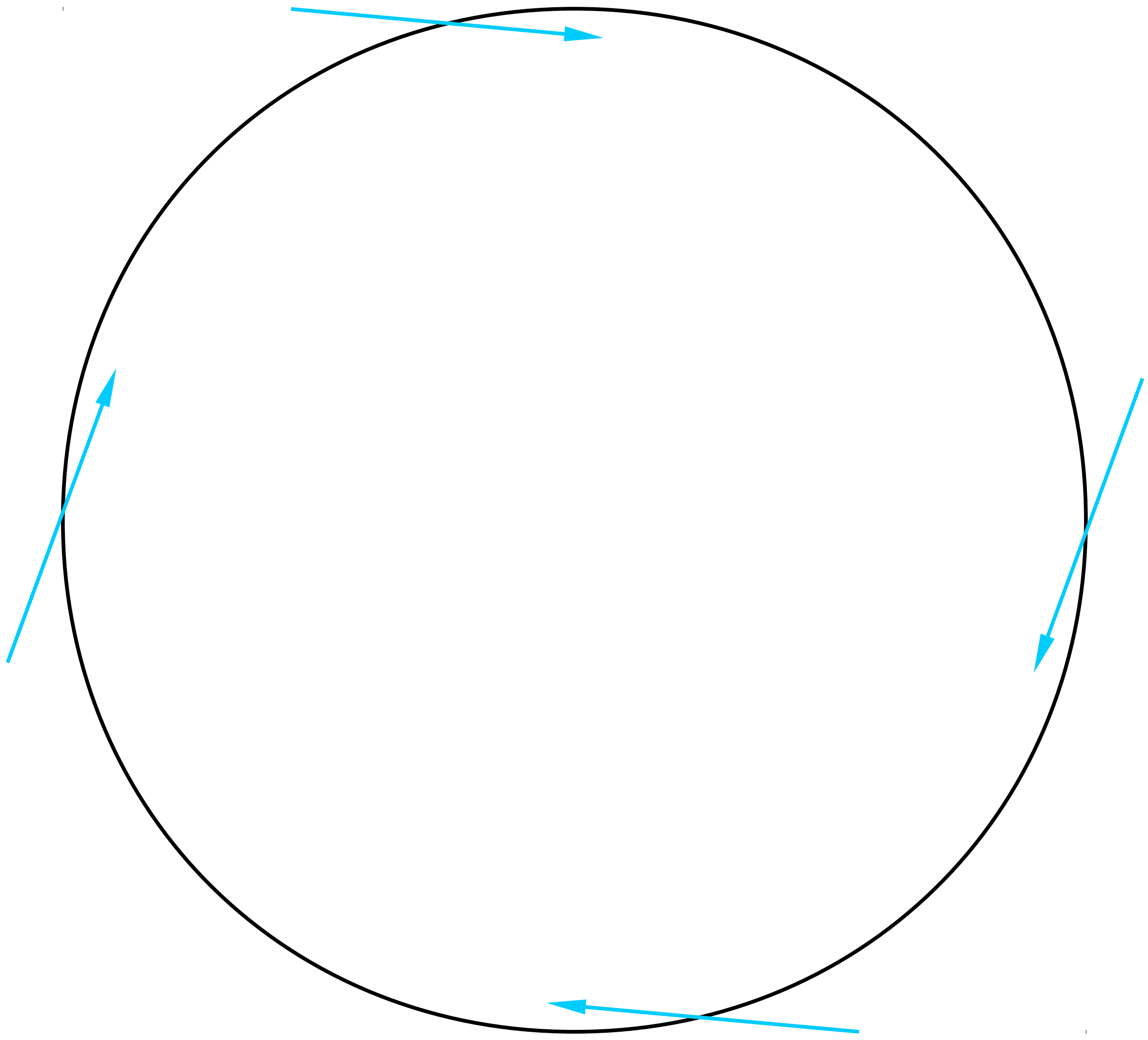
Are these domains symmetric?



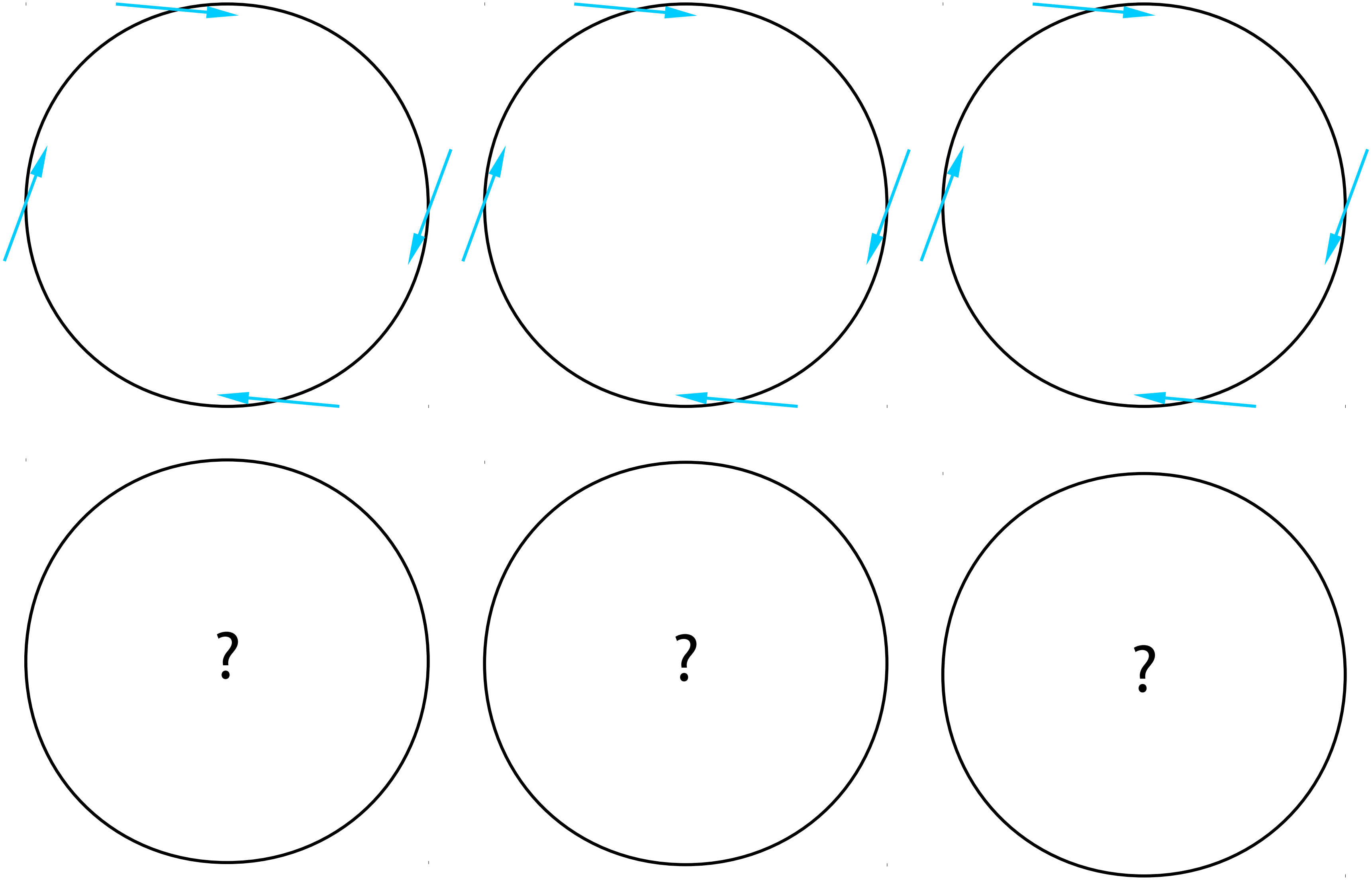
Are these domains symmetric?



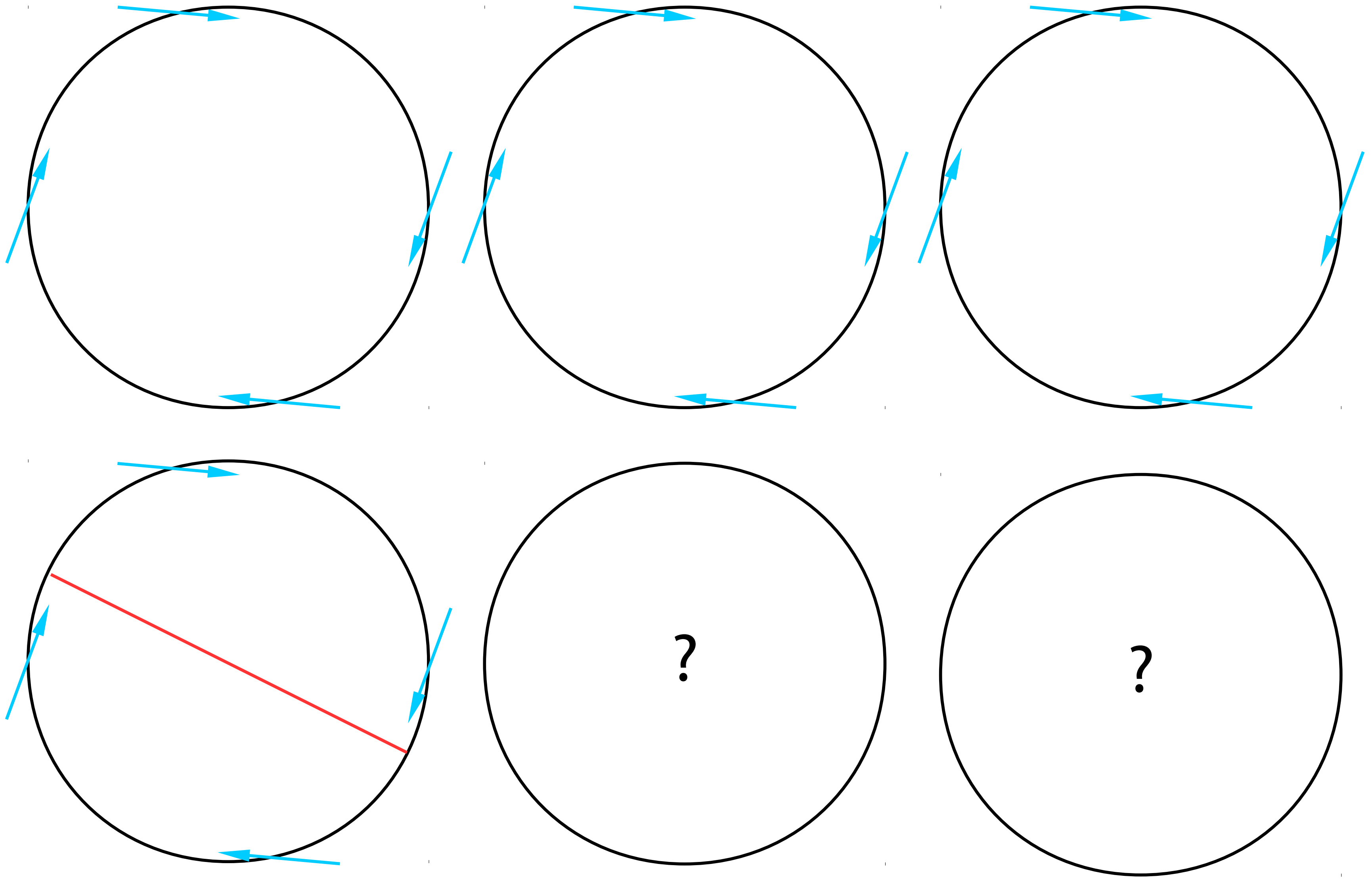
How do we symmetrize this domain?



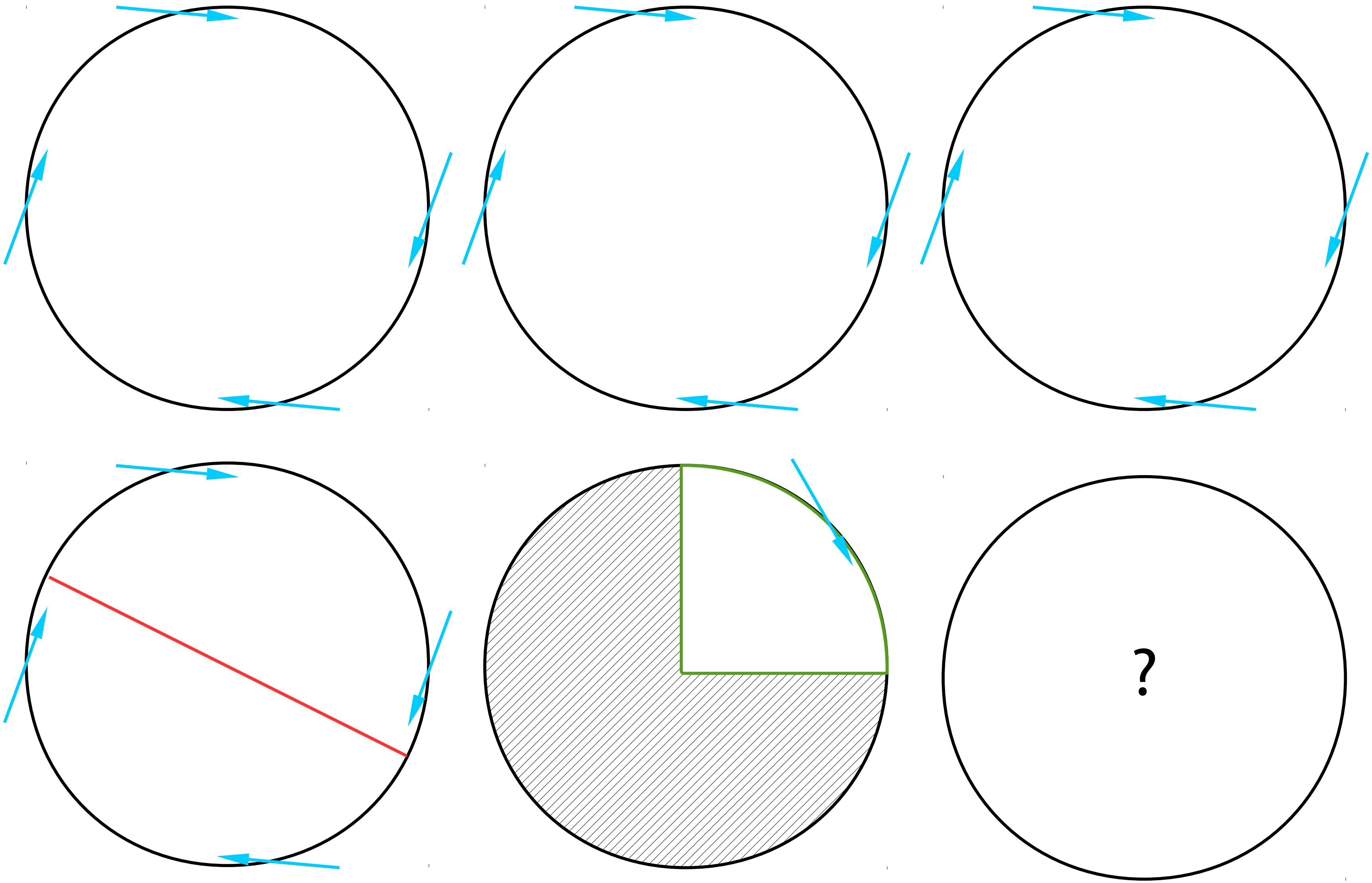
How do we symmetrize this domain?



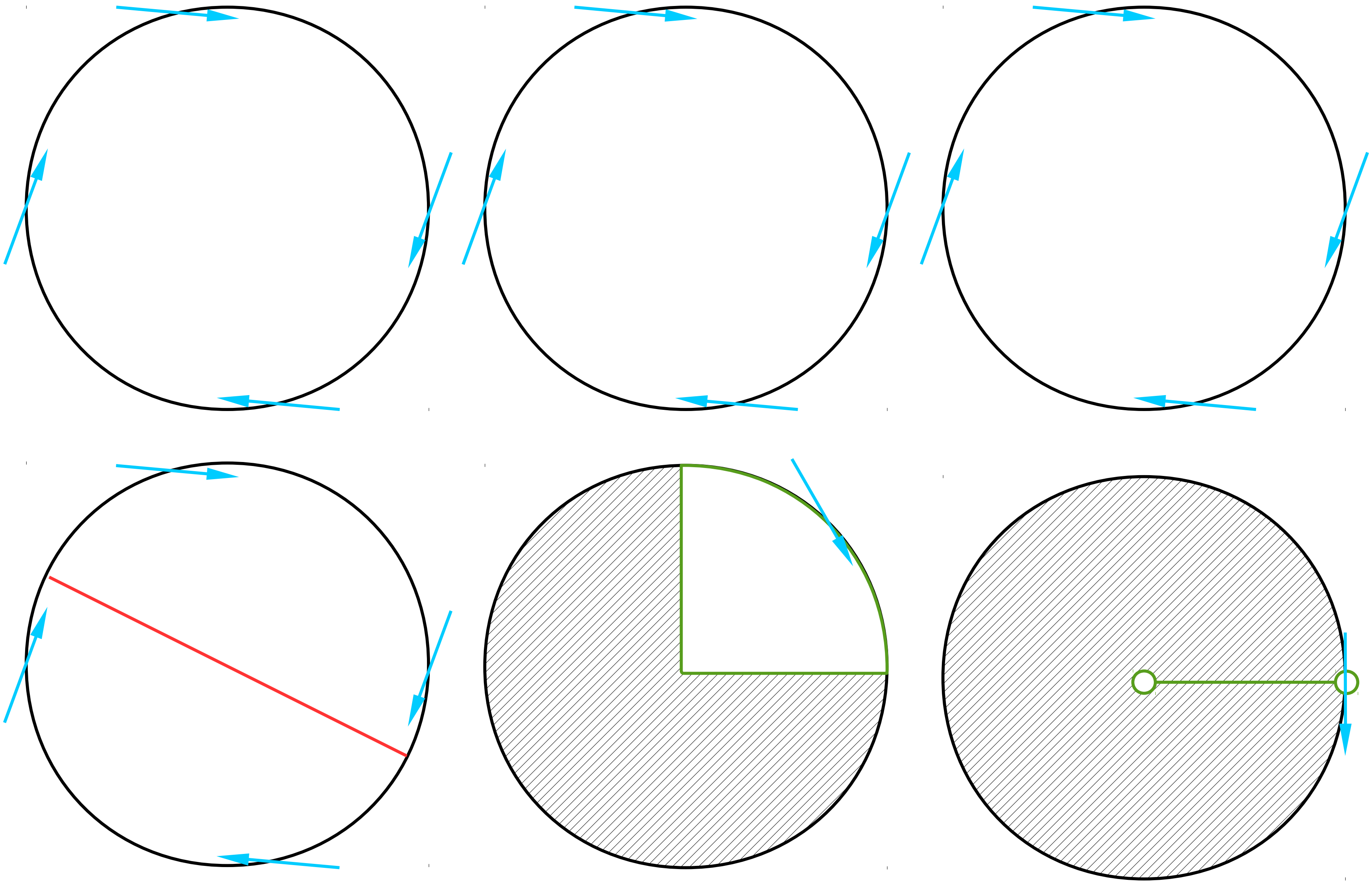
How do we symmetrize this domain?



How do we symmetrize this domain?



How do we symmetrize this domain?



UDF C Macro Language

- `DEFINE_PROFILE`—specify profile condition on boundary face zone
- `DEFINE_TRANSIENT_PROFILE`—specify profile condition on boundary face zone as function of time
- `DEFINE_PROPERTY`—specify material property (density, viscosity, thermal conductivity, rate of strain, mixing laws, ...)
- `DEFINE_SOURCE`—specify custom source term (mass, momentum, energy, species mass fractions, radiation model, user-defined scalars); written for *single* cell

UDF: DEFINE_PROFILE

$$p(y) = 1.1 \times 10^5 - 0.1 \times 10^5 \left(\frac{y}{0.0745}\right)^2$$

```
/*********************************************************************
  UDF for specifying steady-state parabolic pressure profile boundary
  profile for a turbine vane
 *********************************************************************/

#include "udf.h"
DEFINE_PROFILE(pressure_profile,t,i)
{
  real x[ND_ND];
  real y;
  face_t f;
  begin_f_loop(f,t)
  /* this will hold the position vector */
  {
    F_CENTROID(x,f,t);
    y = x[1];
    F_PROFILE(f,t,i) = 1.1e5 - y*y/(.0745*.0745)*0.1e5;
  }
  end_f_loop(f,t)
}
```

UDF: DEFINE_PROPERTY

```
/**********************************************************************
  UDF that simulates solidification by specifying a temperature-
  dependent viscosity property
 **********************************************************************/
#include "udf.h"
DEFINE_PROPERTY(cell_viscosity,c,t)
{
  real mu_lam;
  real temp = C_T(c,t);
  if (temp > 288.)
    mu_lam = 5.5e-3;
  else if (temp > 286.)
    mu_lam = 143.2135 - 0.49725 * temp;
  else
    mu_lam = 1.;
  return mu_lam;
}
```

UDF: DEFINE_SOURCE

$$source = -0.5 C_2 \rho y \mid v_x \mid v_x$$

```
/*******************************************************************
  UDF for specifying an x-momentum source term in a spatially
  dependent porous media
 *******************************************************************/
#include "udf.h"
#define C2 100.0
DEFINE_SOURCE(xmom_source,c,t,dS,eqn)
{
  real x[ND_ND];
  real con, source;
  C_CENTROID(x,c,t);
  con = C2*0.5*C_R(c,t)*x[1];
  source = -con*fabs(C_U(c, t))*C_U(c,t);
  dS[eqn] = -2.*con*fabs(C_U(c,t));
  return source;
}
```