Disclaimer

“This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.”

Introductory OpenFOAM® Course

From 16th to 20th February, 2015

University of Genoa, DICCA

Dipartimento di **I**ngegneria **C**ivile, **C**himica e **A**mbientale

Your Lecturer

Joel GUERRERO

joel.guerrero@unige.it

UNIVERSITÀ DEGLI STUDI
DI GENOVA

guerrero@wolfdynamics.com

wolf dynamics

Today's lecture

1. **Boundary and initial conditions**

Boundary conditions and initial conditions

- First at all, when we use OpenFOAM® to find the approximate solution of the governing equations, we are solving an Initial Boundary Value Problem (IBVP).
- In an IBVP, we need to impose appropriate boundary conditions and initial conditions.
- No need to say that the boundary conditions and initial conditions need to be physically realistic.
- Boundary conditions are a required component of the numerical method, they tell what is going on the boundaries of the domain. You can see them as source terms.
- Initial conditions are also a required component of the numerical method, they define the initial state of the problem.

Boundary conditions and initial conditions

A few words about boundary conditions

Boundary conditions and initial conditions

- Boundary conditions can be divided into three fundamental or mathematical types:
    - Dirichlet boundary conditions.
    - Neumann boundary conditions.
    - Robin Boundary conditions.
- By the way, during this discussion the semantics is not important, that depends of how you want to call the boundary conditions or how they are named in the solver, *i.e.*,
    - in, inlet, inflow, velocity inlet, incoming flow and so on.
    - out, outlet, outflow, velocity outlet, outgoing flow and so on.
    - solid wall, non-penetrating wall, fixed wall and so on.

Boundary conditions and initial conditions

- When we use a **Dirichlet boundary condition**, we prescribe the value of a variable at the boundary.
- When we use a **Neumann boundary condition**, we prescribe the gradient normal to the boundary.
- **Robin boundary conditions**, are a mixed of Dirichlet boundary conditions and Neumann boundary conditions.
- You can use any of these three boundary conditions in OpenFOAM®.

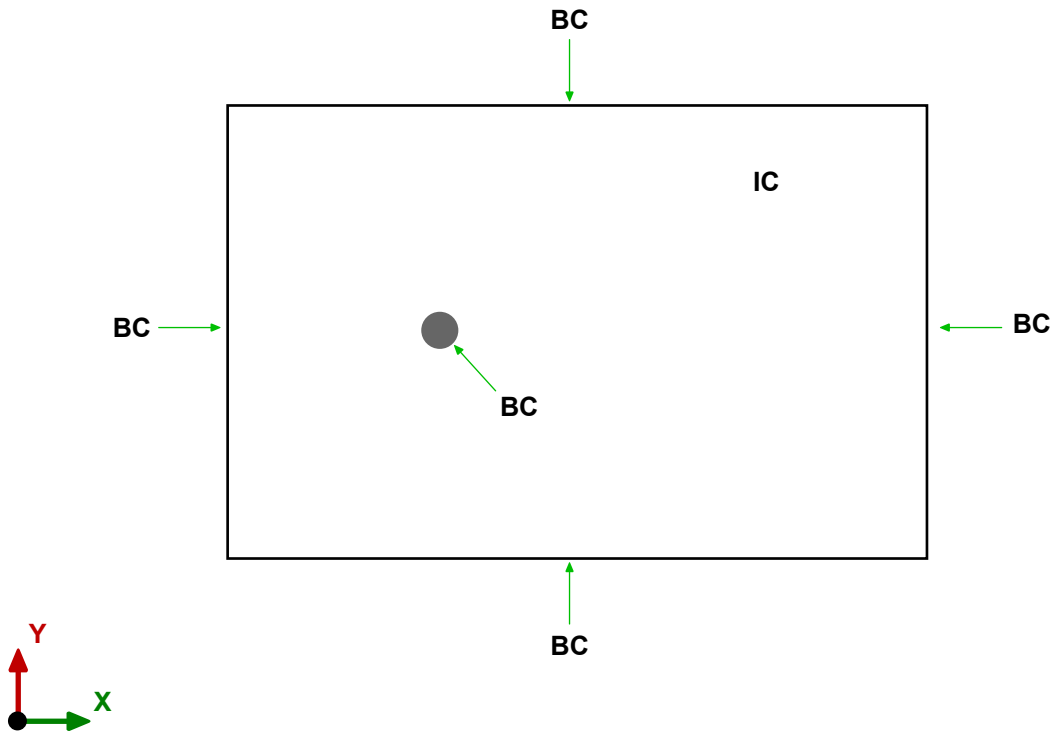
Boundary conditions and initial conditions

- Defining boundary conditions involves:
    - Determining the boundary condition type.
    - Finding the location of the boundary condition in the domain.
    - Giving the required physical information.

- The choice of the boundary conditions depend on:
    - Geometrical considerations.
    - Physics involved.
    - Information available at the boundary condition location.
    - Numerical considerations.

Boundary conditions and initial conditions

- To define boundary conditions you need to know the location of the boundaries (where they are in your mesh), and supply the information at the boundaries.
- You must know what information is required at the boundaries, you must know the physics involved.



Boundary conditions and initial conditions

A few words about initial conditions

Boundary conditions and initial conditions

- Initial conditions can be divided into two groups:
    - Uniform initial conditions.
    - Non-uniform initial conditions.

- For non-uniform initial conditions, the value used can be obtained from:
    - Another simulation, including a solution with different grid resolution.
    - A potential solver.
    - Experimental results.
    - A mathematical function
    - Reduced order model.

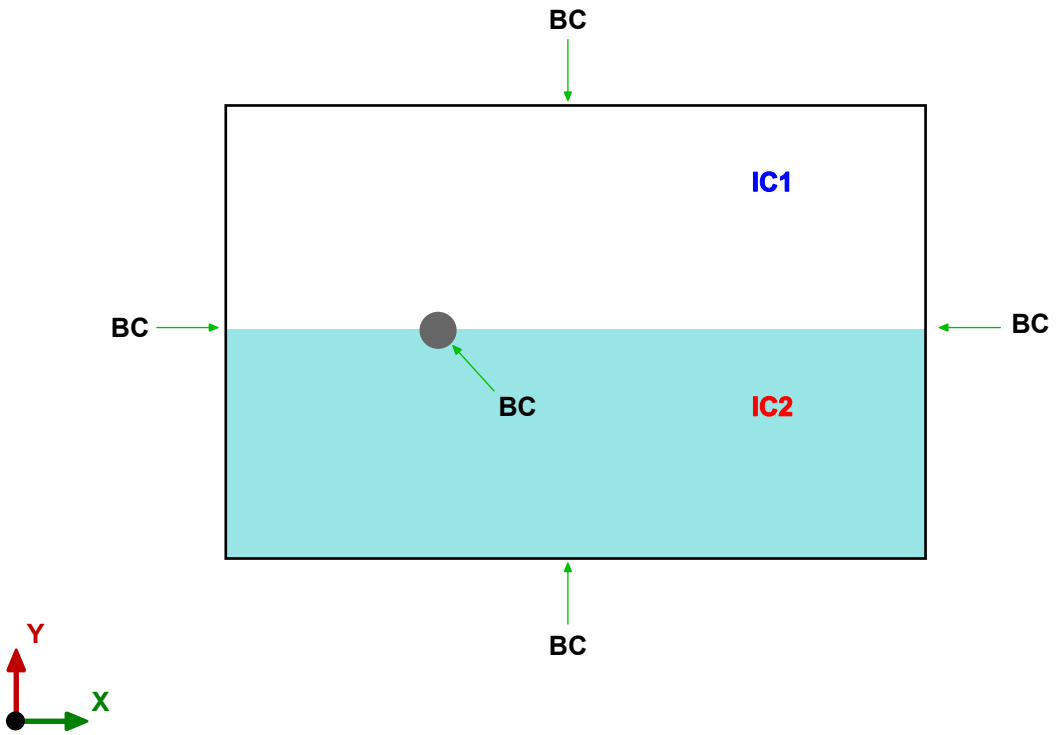
Boundary conditions and initial conditions

- Defining initial conditions involves:
    - Determining the initial condition type.
    - Finding the location of the initial condition in the domain.
    - Giving the required physical information.

- The choice of the initial conditions depend on:
    - Geometrical considerations.
    - Physics involved.
    - Information available.
    - Numerical considerations.

Boundary conditions and initial conditions

- For initial conditions, you need to supply the initial information or initial state of your problem.
- This information can be a uniform value or a non-uniform value.
- You can apply the initial conditions to the whole domain or zones of the domain (patches).



Boundary conditions and initial conditions

A few considerations and guidelines

Boundary conditions and initial conditions

- Remember:
  - Poorly defined boundary conditions can have a significant impact on your solution.

  - Sometimes, finding the right boundary conditions can be really tricky.
  - Initial conditions are as important as the boundary conditions.
  - A good initial condition can improve the stability and convergence rate.
  - On the other hand, unphysical initial conditions can slow down the convergence rate or can cause divergence.

  - You need to define boundary conditions and initials conditions for every single variable you are solving.

Boundary conditions and initial conditions

- The required values of the boundary conditions and initial conditions depend on the equations you are solving and physical models used, *e.g.*,
    - For incompressible, and laminar flows you will need to set only the velocity and pressure.
    - If you are solving a turbulent compressible flow you will need to set velocity, pressure, temperature and the turbulent variables.
    - For multiphase flows you will need to set the primitives variables for each phase. You might also need to initialize the phases patches.
    - If you are doing turbulent combustion or chemical reactions, you will need to define the species, reactions and primitive and turbulent variables.
- By the way, you also need to define the physical properties.
- **Remember, the physics is yours.**

Boundary conditions and initial conditions

- **General guidelines when choosing the boundary conditions and initial conditions:**
    - Boundary conditions and initial conditions need to be physically realistic.
    - Minimize grid skewness, non-orthogonality, growth rate, and aspect ratio near the boundaries. You do not want to introduce diffusion errors early in the simulation, specially close to the inlets.
    - Try to avoid large gradients in the direction normal to the boundaries near inlets and outlets. That is to say, put your boundaries far away from where things are happening.

Boundary conditions and initial conditions

- **General guidelines when choosing the boundary conditions and initial conditions:**
    - Do not force the flow at the outlet, use a zero normal gradient for all flow variables except pressure. The solver extrapolates the required information from the interior.
    - Be careful with backward flow at the outlets (flow coming back to the domain) and backward flow at inlets (reflection waves), they required special treatment.
    - If possible, select inflow and outflow boundary conditions such that the flow either goes in or out normal to the boundaries.
    - Use zero gradient boundary conditions only with incompressible flows and when you are sure that the flow is fully developed.

Boundary conditions and initial conditions

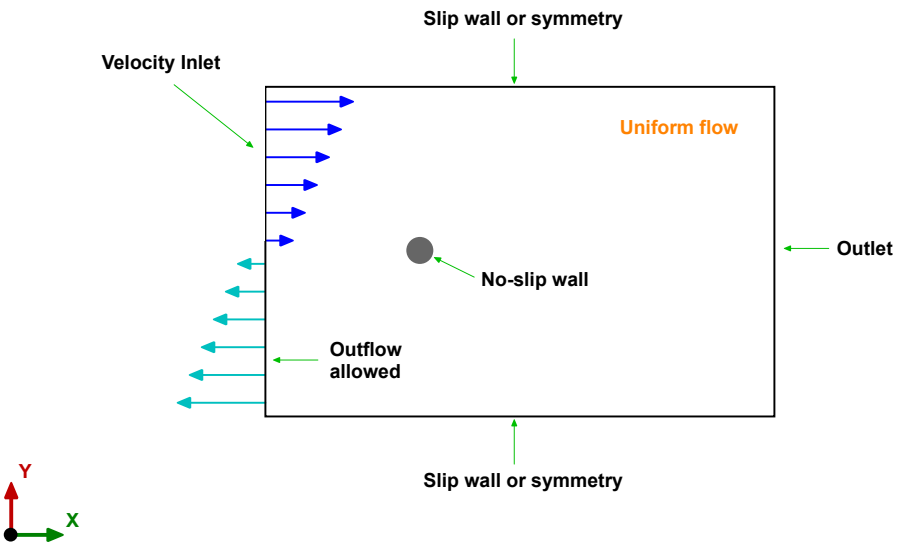
- **General guidelines when choosing the boundary conditions and initial conditions:**
    - At least one boundary condition should specify pressure (total or static).
    - Outlets that discharge to the atmosphere can use a static pressure boundary condition. This is interpreted as the static pressure of the environment into which the flow exhausts.
    - Inlets that take flow into the domain from the atmosphere can use a total pressure boundary condition (e.g. open window).
    - Mass flow inlets produce a uniform velocity profile at the inlet.

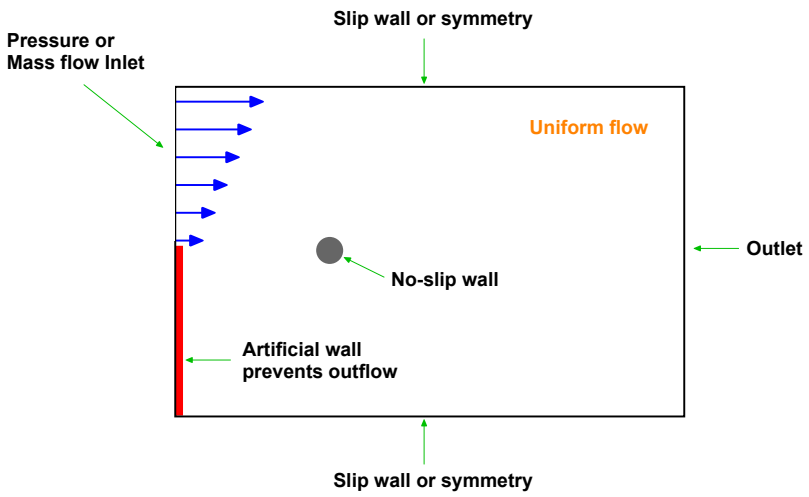
Boundary conditions and initial conditions

- **General guidelines when choosing the boundary conditions and initial conditions:**
  - The mass flow distribution at mass flow outlets is based on the upstream values.
  - Pressure specified boundary conditions allow a natural velocity profile to develop.
  - Velocity inlet boundary conditions are intended for incompressible flow.
  - For compressible flows use mass flow inlets or pressure boundary conditions.

Boundary conditions and initial conditions

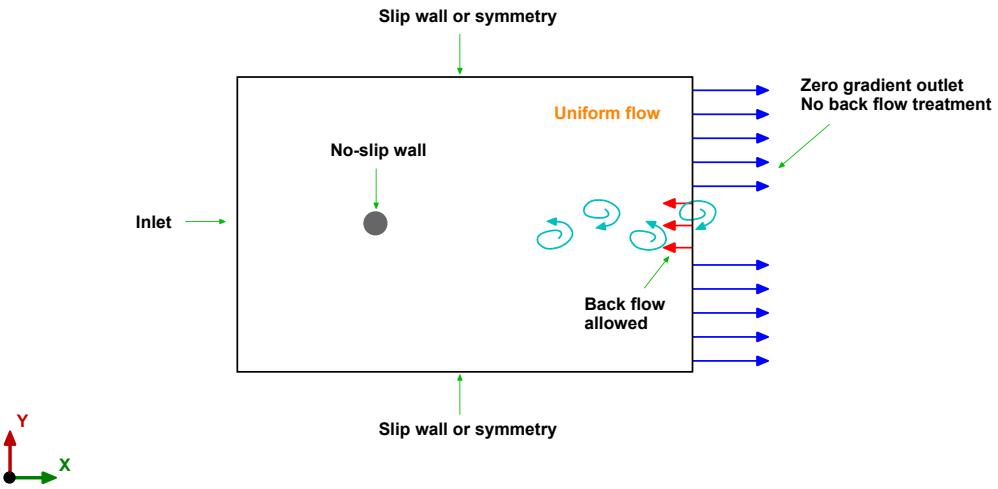
- **Inlets and outlets boundary conditions:**
  - Inlets are for regions where inflow is expected; however, inlets might support outflow when a velocity profile is specified.
  - Pressure boundary conditions do not allow outflow at the inlets.
  - Velocity specified inlets are intended for incompressible flows.
  - Pressure and mass flow inlets are suitable for compressible and incompressible flows.
  - Same concepts apply to outlets, which are regions where outflow is expected.

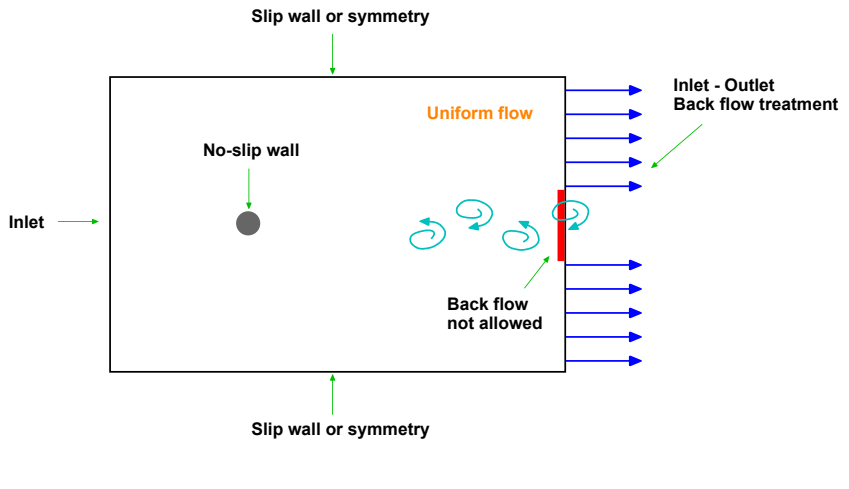




Boundary conditions and initial conditions

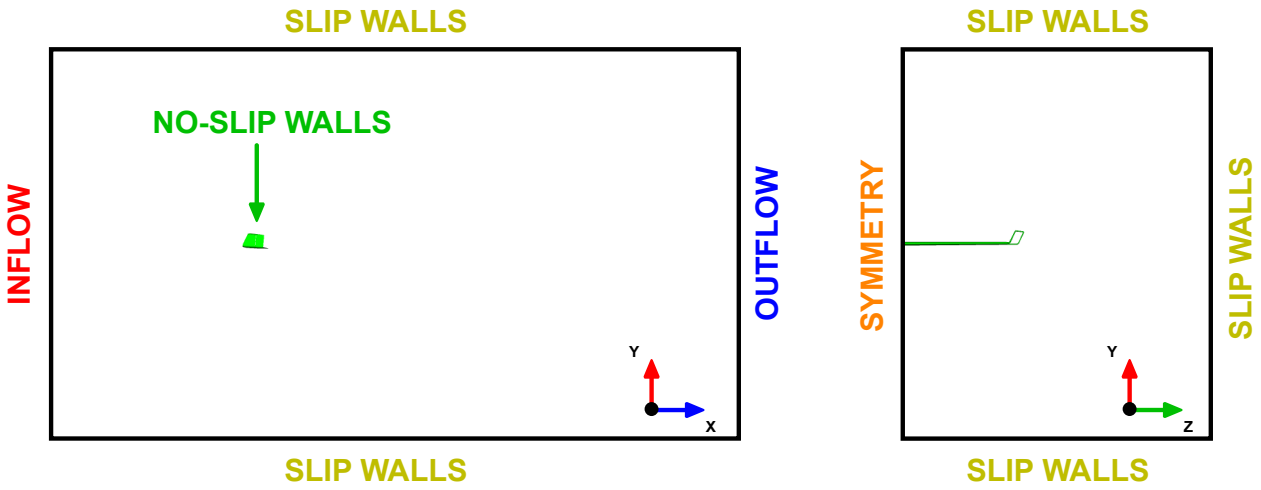
- **Zero gradient and backflow boundary conditions:**
  - Zero gradient boundary conditions extrapolates the values from the domain. They require no information.
  - Zero gradient boundary conditions can be used at inlets and outlets.
  - Backflow boundary conditions provide a generic outflow/inflow condition, with specified inflow/outflow for the case of return flow.
  - In the case of a backflow outlet, when the flux is positive (out of domain) it applies a Neumann boundary condition (zero gradient), and when the flux is negative (into of domain), it applies a Dirichlet boundary condition (fixed value).
  - Same concept applies to backflow inlets.





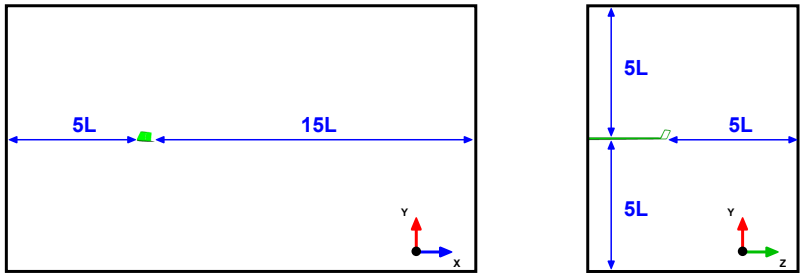
Boundary conditions and initial conditions

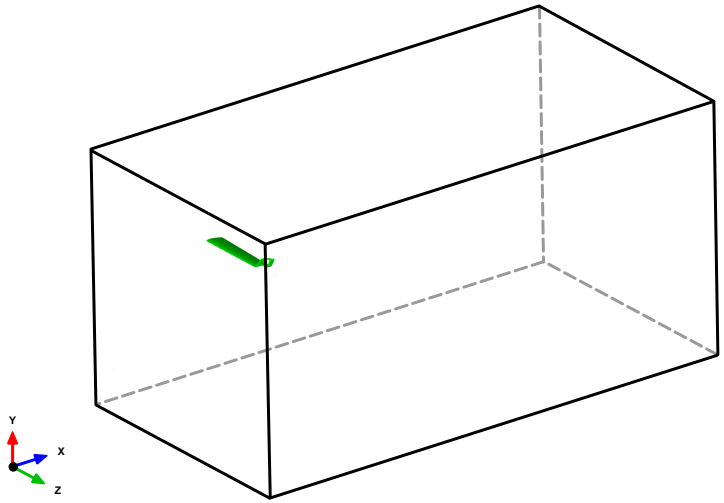
- **Symmetry boundary conditions:**
  - I do not like to use symmetry boundary conditions. They are a big simplification of the problem, specially if you are dealing with turbulence.
  - However, they help to reduce mesh dimension and they speed-up the computations.
  - Have in mind that symmetry boundary conditions only apply to **planar faces**.
  - To use symmetry boundary conditions, both the geometry and the flow field must be symmetric.
  - Mathematically speaking, setting a symmetry boundary condition is equivalent to:
    - Zero normal velocity at the symmetry plane.
    - Zero normal gradients of all variables at symmetry plane.



Boundary conditions and initial conditions

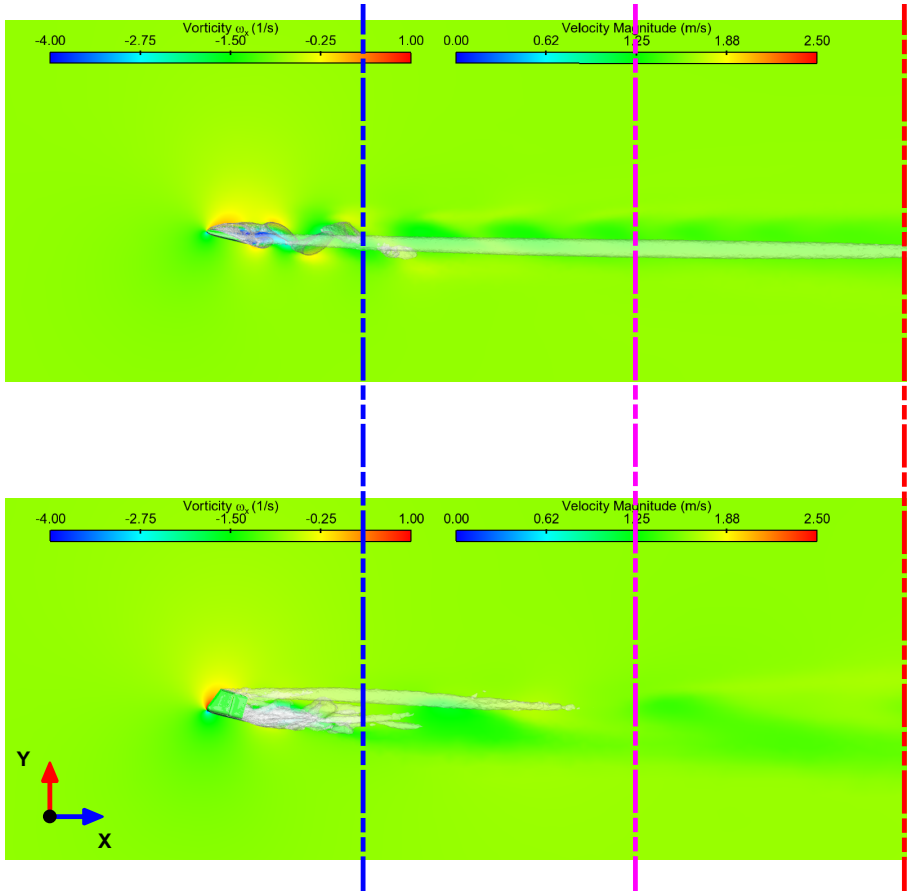
- **Domain dimensions (when the dimensions are not known):**
  - As a general guideline you can use the dimensions illustrated in the figure, where **L** is a reference length. In this case, **L** is the wing chord.
  - Always verify that there are no significant gradients normal to any of the boundaries patches. If there are, you should consider increasing the domain dimensions.

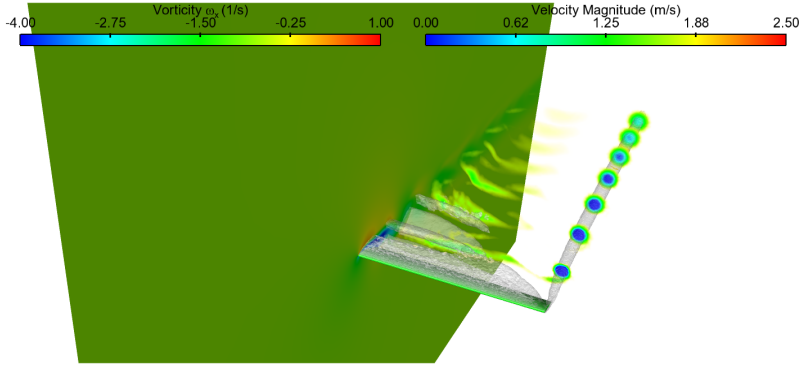


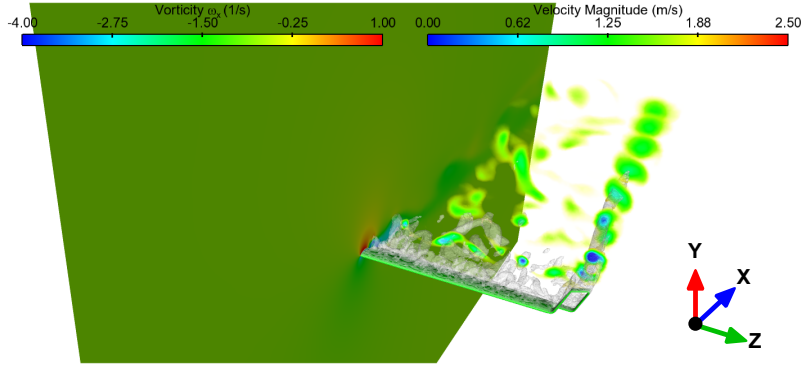


Boundary conditions and initial conditions

- **Location of the outlet boundary condition:**
  - Place outlet boundary conditions as far as possible from recirculation zones or backflow conditions, by doing this you increase the stability.
  - Remember, backflow conditions requires special treatment.







Boundary conditions and initial conditions

Boundary conditions in OpenFOAM®

Boundary conditions and initial conditions

- OpenFOAM® distinguish between **base type** boundary conditions and **numerical type** boundary conditions.
  - **Base type**. This boundary condition is based on geometry information or an inter-processor communication link (halo boundaries).
  - **Numerical type**. This boundary condition assigns the value to the field variables in the given patch. It can be of primitive type or derived type.

- Some **base type** and **numerical type** boundary conditions are paired. That is, you define the same type for the **base type** and the **numerical type**.

Boundary conditions and initial conditions

- OpenFOAM® distinguish between **base type** boundary conditions and **numerical type** boundary conditions.

	Unpaired	**Paired**

Base type	`patch`
	`wall`
		`symmetry`
		`empty`
		`wedge`
		`cyclic`
		`processor`

Numerical type	`fixedValue`
	`fixedGradient`
	`zeroGradient`
	`fixedFluxPressure`
	`inletOutlet`
	`totalPressure`
	`uniformFixedValue`
	
	and so on …
		`symmetry`
		`empty`
		`wedge`
		`cyclic`
		`processor`

Boundary conditions and initial conditions

- **Base type** boundary conditions are defined in the file **`boundary`** located in the directory **`constant/polyMesh`**
- The file **`boundary`** is automatically created when you generate or convert the mesh.
- When you convert a mesh to OpenFOAM® format, you might need to manually modify the file **`boundary`**. This is because the conversion utilities do not recognize the boundary type of the original mesh.
- Remember, if a **base type** boundary condition is missing, OpenFOAM® will complain and will tell you where and what is the error.
- Also, if you misspelled something OpenFOAM® will complain and will tell you where and what is the error.

Boundary conditions and initial conditions

- **Numerical type** boundary conditions are defined in the field variables dictionaries located in the directory **0** (*e.g.* `U`, `p`).
- When we talk about **numerical type** boundary conditions we are referring to Dirichlet, Neumann or Robin boundary conditions.
- In OpenFOAM®, **numerical type** boundary conditions can be of primitive or derived type.
- You need to manually create the field variables dictionaries (*e.g.* **`0/U`**, **`0/p`**). I hope you know how to do it, because we have done it.
- Remember, if you forget to define a boundary condition, OpenFOAM® will complain and will tell you where and what is the error.
- Also, if you misspelled something OpenFOAM® will complain and will tell you where and what is the error.

Boundary conditions and initial conditions

- The name of the **base type** boundary condition and the name of the **numerical type** boundary condition needs to be the same, if not, OpenFOAM® will complain.
- Pay attention to this, specially if you are converting the mesh from another format.

0/U	**0/p**	**constant/polyMesh/boundary**
inlet	inlet	inlet
oUtlet	outlet	outlet
top	top	top
bottom	bottom	Bottom
cylinder	cylinder	cylinder
sym	sym	sym

Boundary conditions and initial conditions

- The following **base type** boundary conditions are paired. That is, the type needs to be same in the **boundary** dictionary and field variables dictionaries (*e.g.* **U**, **p**).

constant/polyMesh/boundary	**0/U - 0/p**
symmetry	symmetry
symmetryPlane	symmetryPlane
empty	empty
wedge	wedge
cyclic	cyclic
processor	processor

Boundary conditions and initial conditions

- The following **numerical type** boundary conditions can be any of the primitive or derived type available in OpenFOAM®. Mathematically speaking; they can be Dirichlet, Neumann or Robin boundary conditions.

constant/polyMesh/boundary	**0/U - 0/p**
patch	fixedValue fixedGradient zeroGradient fixedFluxPressure inletOutlet freeStream slip totalPressure uniformFixedValue **and so on …**

Boundary conditions and initial conditions

- The **`wall`** base type boundary condition is defined as follows:

constant/polyMesh/boundary	**0/U**	**0/p**
wall	type fixedValue; value uniform (0 0 0);	zeroGradient

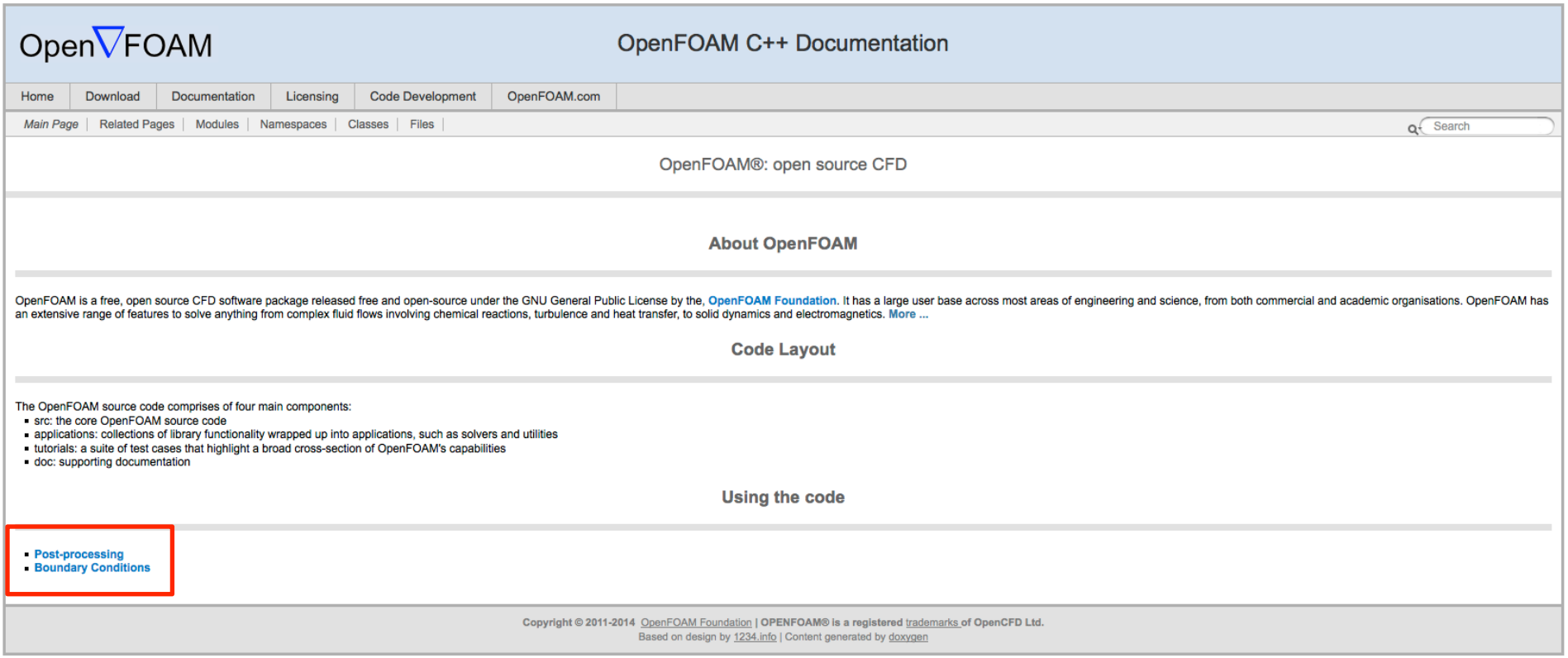
- This boundary condition is not contained in the **`patch`** base type boundary condition group, because specialize modeling options can be used on this boundary condition.
- An example is turbulence modeling, where turbulence can be generated or dissipated at the walls.

Boundary conditions and initial conditions

- There are numerous primitive and derived type boundary conditions implemented in OpenFOAM®.
- You can find the source code of the numerical boundary conditions in the following directory:
    - **`$FOAM_SRC/finiteVolume/fields`**
- The wall boundary conditions for the turbulence models (wall functions), are located in the following directory:
    - **`$FOAM_SRC/turbulenceModels/`**

Boundary conditions and initial conditions

- To get more information about all the boundary conditions available in OpenFOAM® you can read the Doxygen documentation, just look for the **Using the code** section at the bottom of the page.
- If you did not compile the Doxygen documentation, you can access the information online, http://www.openfoam.org/docs/cpp/



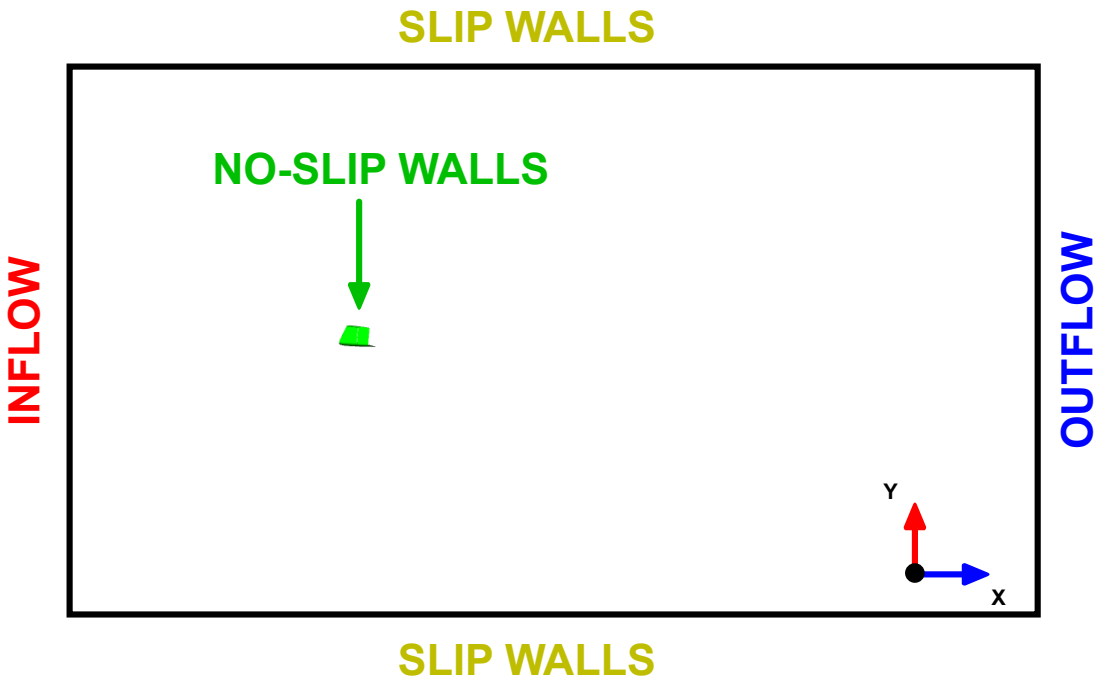
Boundary conditions and initial conditions

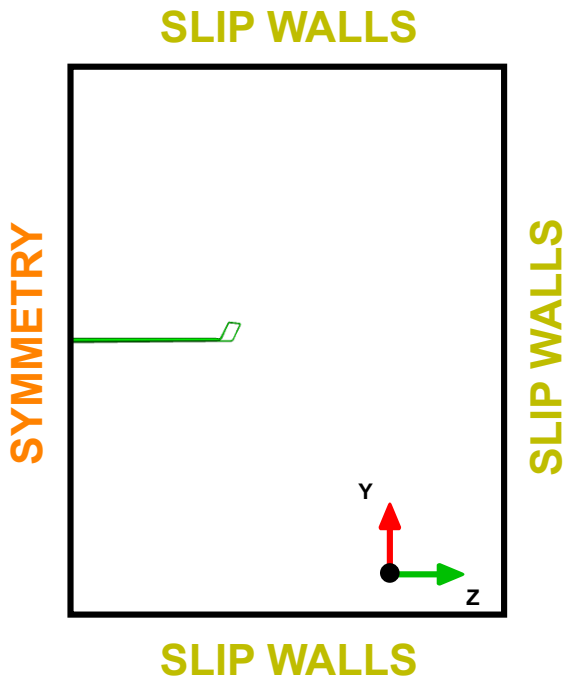
A few boundary conditions setup

Boundary conditions and initial conditions

- **Boundary conditions setup:**

At inflow:	At outflow:	At no-slip wall:	At symmetry plane:	At slip walls:
U = fixed value	U = back flow	$U = 0$	U = symmetry	U = slip
p = zero gradient	p = fixed value	p = zero gradient	p = symmetry	p = slip
T = fixed value	T = back flow	T = zero gradient	T = symmetry	T = slip
κ = fixed value	κ = back flow	κ = zero gradient or fixed value	κ = symmetry	κ = slip
ω = fixed value	ω = back flow	ω = zero gradient or fixed value	ω = symmetry	ω = slip

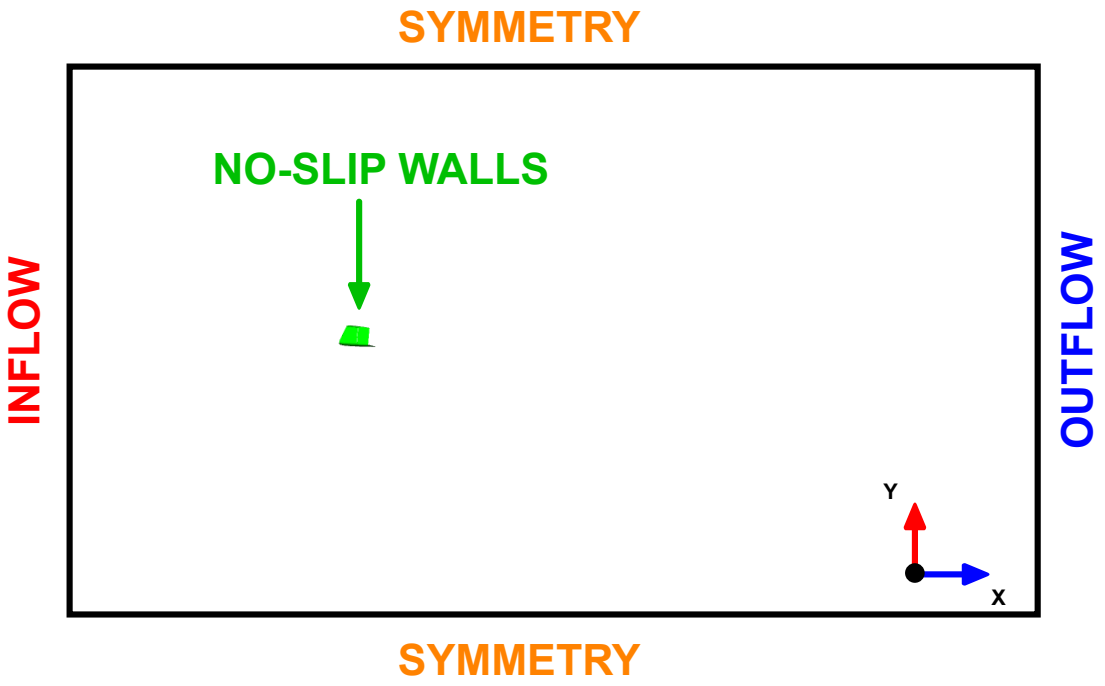


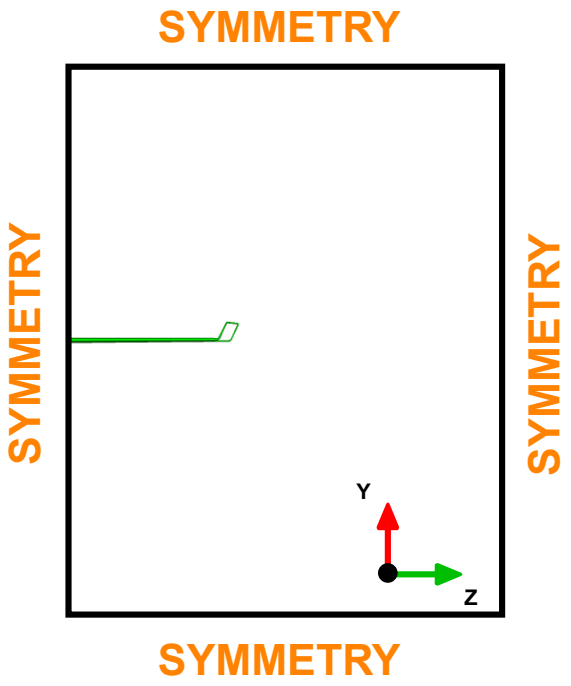


Boundary conditions and initial conditions

- **Boundary conditions setup:**

At inflow:	At outflow:	At no-slip wall:	At symmetry plane:
U = fixed value	U = back flow	$U = 0$	U = symmetry
p = zero gradient	p = fixed value	p = zero gradient	p = symmetry
T = fixed value	T = back flow	T = zero gradient	T = symmetry
κ = fixed value	κ = back flow	κ = zero gradient or fixed value	κ = symmetry
ω = fixed value	ω = back flow	ω = zero gradient or fixed value	ω = symmetry

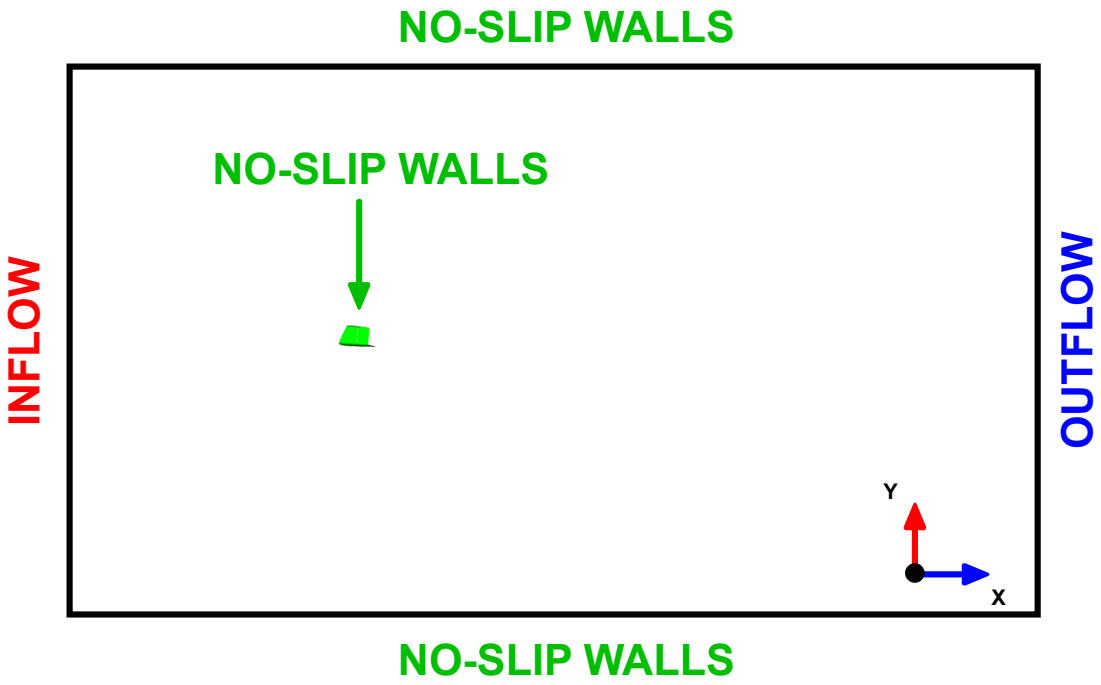


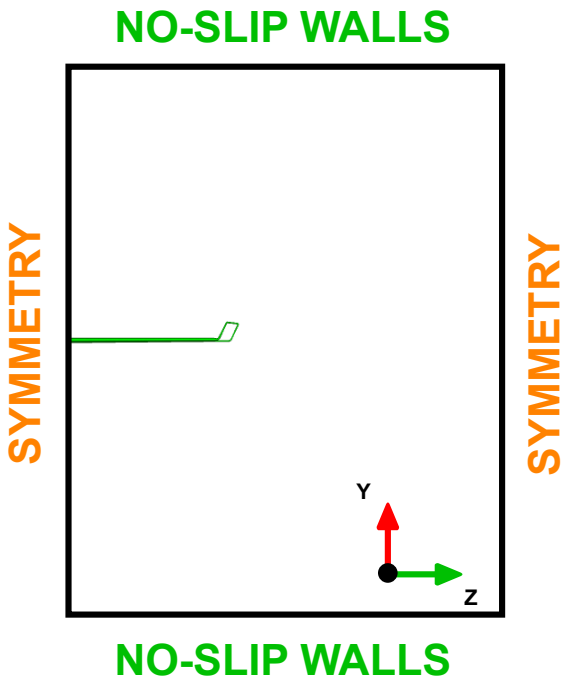


Boundary conditions and initial conditions

- **Boundary conditions setup:**

At inflow:	At outflow:	At no-slip wall:	At symmetry plane:
U = fixed value	U = back flow	$U = 0$	U = symmetry
p = zero gradient	p = fixed value	p = zero gradient	p = symmetry
T = fixed value	T = back flow	T = zero gradient	T = symmetry
κ = fixed value	κ = back flow	κ = zero gradient or fixed value	κ = symmetry
ω = fixed value	ω = back flow	ω = zero gradient or fixed value	ω = symmetry

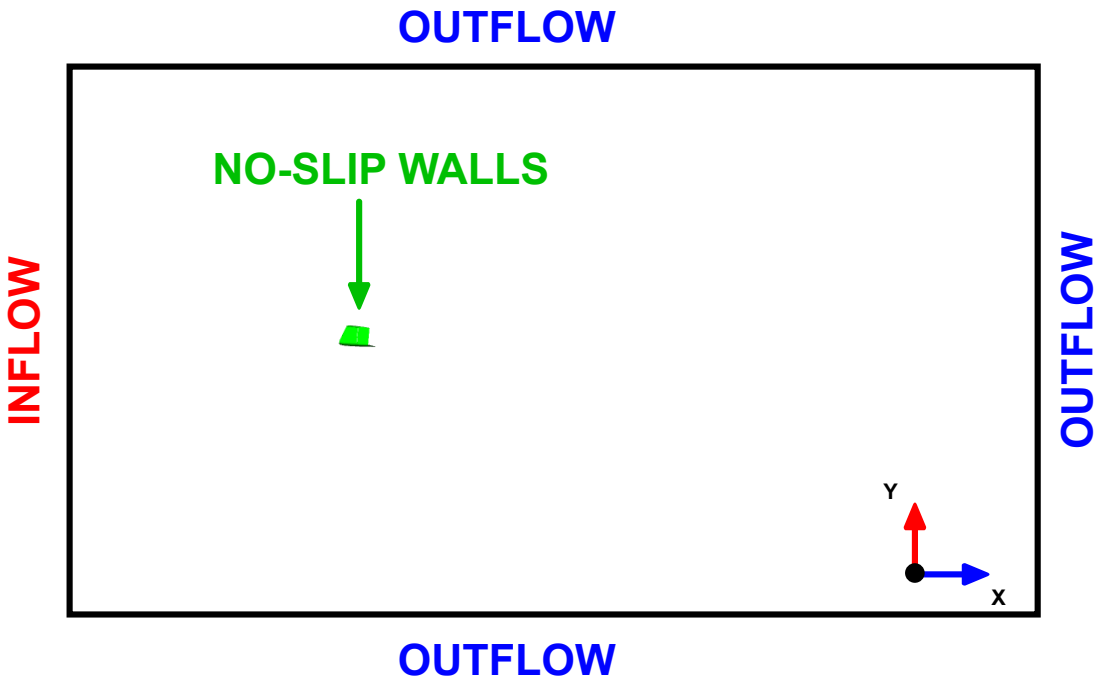


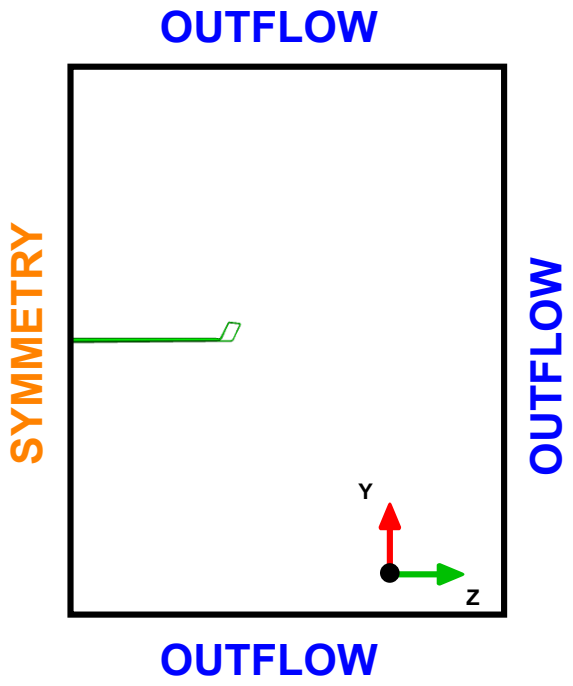


Boundary conditions and initial conditions

- **Boundary conditions setup:**

At inflow:	At outflow:	At no-slip wall:	At symmetry plane:
U = fixed value	U = back flow	$U = 0$	U = symmetry
p = zero gradient	p = fixed value	p = zero gradient	p = symmetry
T = fixed value	T = back flow	T = zero gradient	T = symmetry
κ = fixed value	κ = back flow	κ = zero gradient or fixed value	κ = symmetry
ω = fixed value	ω = back flow	ω = zero gradient or fixed value	ω = symmetry

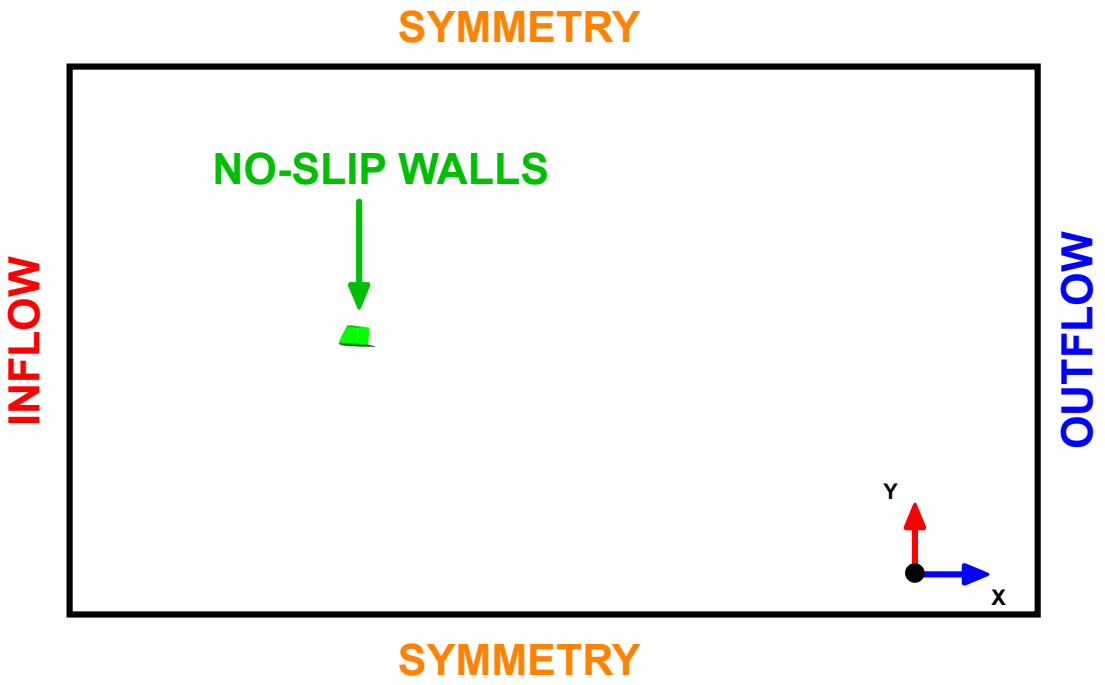


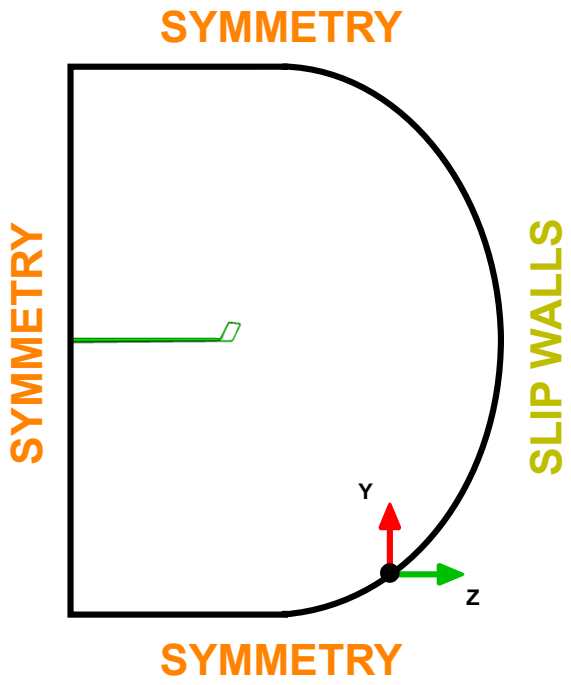


Boundary conditions and initial conditions

- **Boundary conditions setup:**

At inflow:	At outflow:	At no-slip wall:	At symmetry plane:	At slip walls:
U = fixed value	U = back flow	$U = 0$	U = symmetry	U = slip
p = zero gradient	p = fixed value	p = zero gradient	p = symmetry	p = slip
T = fixed value	T = back flow	T = zero gradient	T = symmetry	T = slip
κ = fixed value	κ = back flow	κ = zero gradient or fixed value	κ = symmetry	κ = slip
ω = fixed value	ω = back flow	ω = zero gradient or fixed value	ω = symmetry	ω = slip

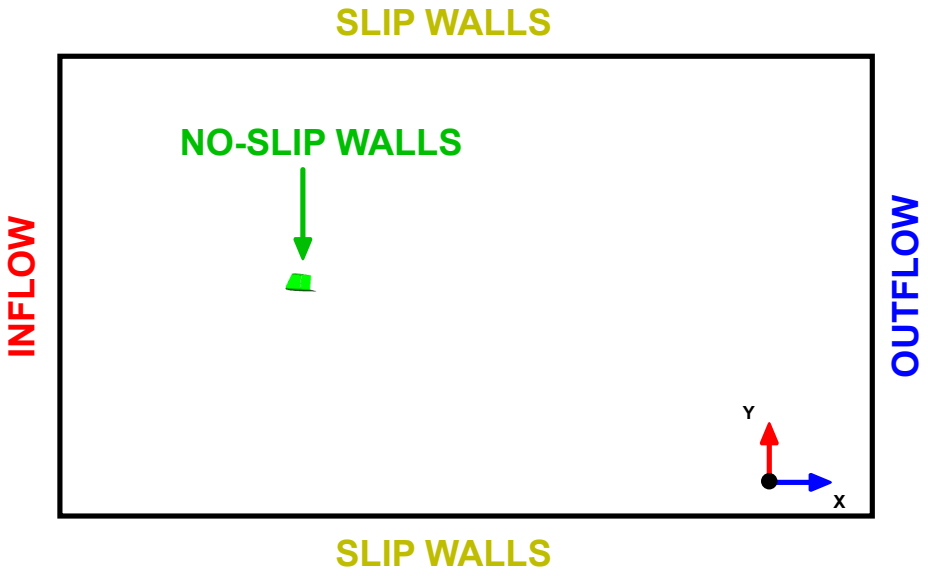


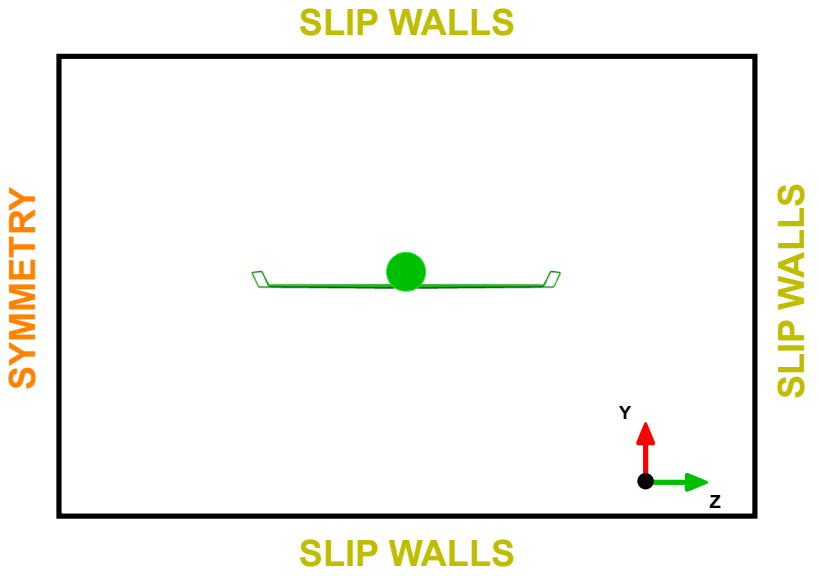


Boundary conditions and initial conditions

- **Boundary conditions setup:**

At inflow:	At outflow:	At no-slip wall:	At symmetry plane:	At slip walls:
U = fixed value	U = back flow	$U = 0$	U = symmetry	U = slip
p = zero gradient	p = fixed value	p = zero gradient	p = symmetry	p = slip
T = fixed value	T = back flow	T = zero gradient	T = symmetry	T = slip
κ = fixed value	κ = back flow	κ = zero gradient or fixed value	κ = symmetry	κ = slip
ω = fixed value	ω = back flow	ω = zero gradient or fixed value	ω = symmetry	ω = slip

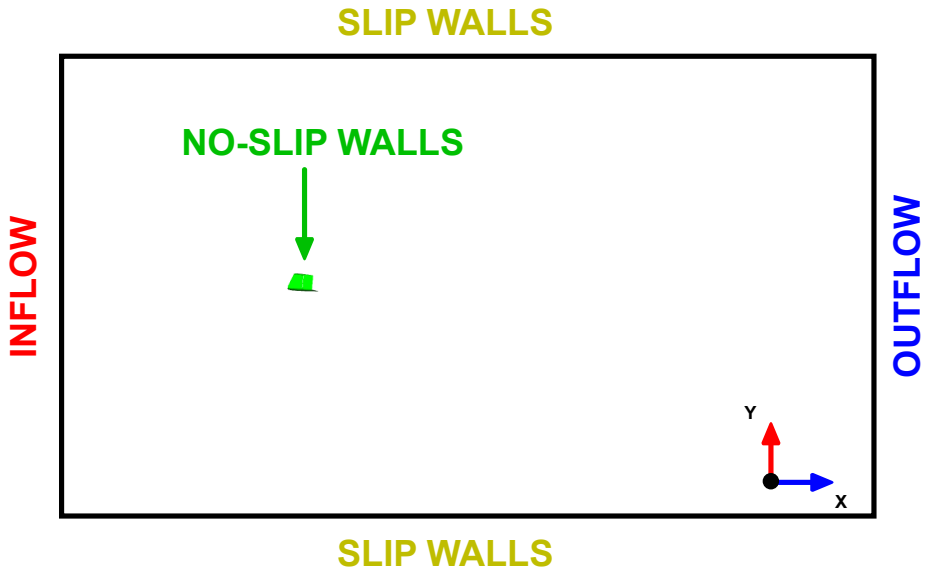


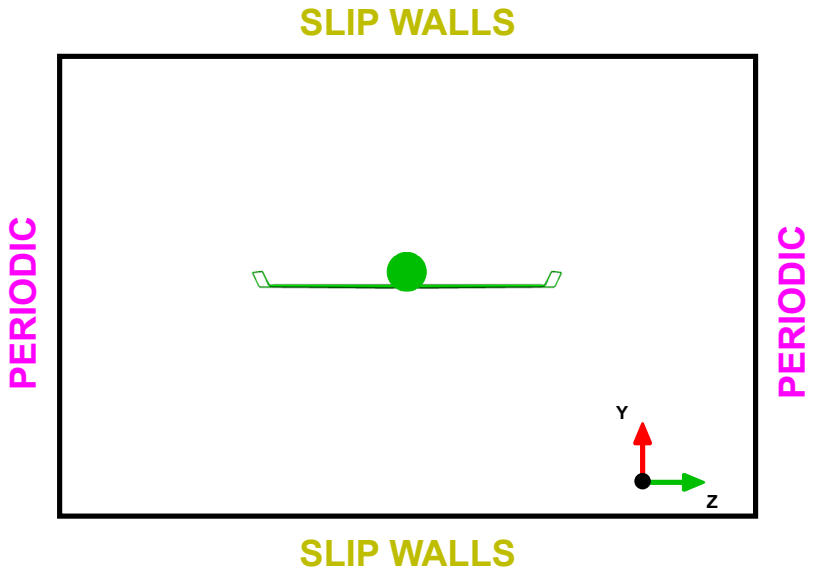


Boundary conditions and initial conditions

- **Boundary conditions setup:**

At inflow:	At outflow:	At no-slip wall:	At periodic patches:	At slip walls:
U = fixed value	U = back flow	$U = 0$	U = periodic	U = slip
p = zero gradient	p = fixed value	p = zero gradient	p = periodic	p = slip
T = fixed value	T = back flow	T = zero gradient	T = periodic	T = slip
κ = fixed value	κ = back flow	κ = zero gradient or fixed value	κ = periodic	κ = slip
ω = fixed value	ω = back flow	ω = zero gradient or fixed value	ω = periodic	ω = slip





Boundary conditions and initial conditions

- So far we have not study turbulence, however, in all previous configurations we are missing one boundary condition?

- Do you know which one?

- Do not worry we are going to address the boundary conditions needed for turbulence modeling later on.

Boundary conditions and initial conditions

- All the previous boundary conditions setup will generate a well posed problem.

- However, some of them are more stable.
  - A very robust and stable configuration is to set the inlet velocity and the static pressure at outlet. The inlet total pressure is an implicit result of the solution.
  - You can also set mass flow rate at inlet and the static pressure at outlet. The inlet total pressure will be adjusted to get the given mass flow.
  - Less robust but still stable, total pressure at inlet with static pressure at outlet. This configuration is sensitive to initial conditions.

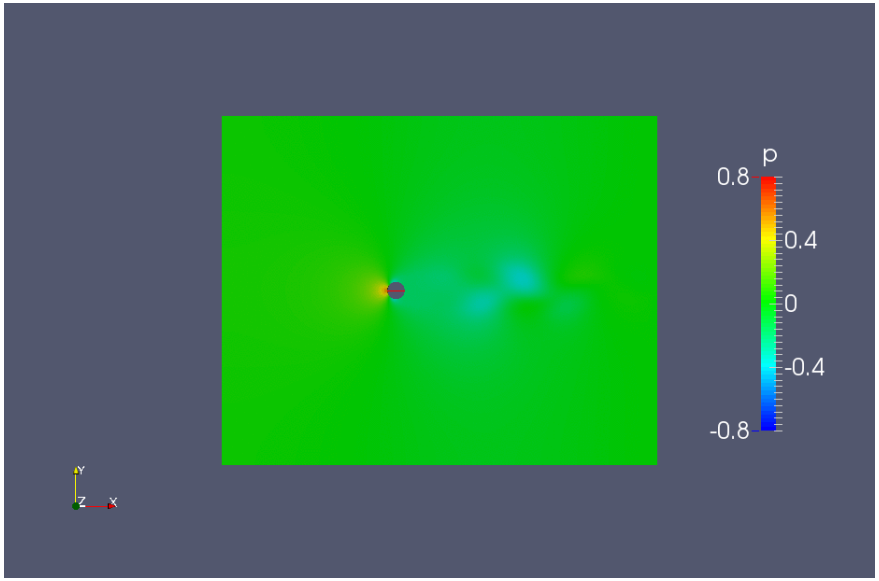
Boundary conditions and initial conditions

- And some configurations are unreliable.
    - Inlet velocity, mass flow or total pressure at inlet and zero gradient at outlet. These combinations should be avoided because the static pressure level is not fixed.
    - Velocity at outlet. Unless the flow is fully developed, you will be forcing the flow to an unrealistic condition.
    - Zero gradients at inlet and outlet, this will result in an ill-posed problem.
    - Velocity at inlet and outlet. The configuration is unstable.

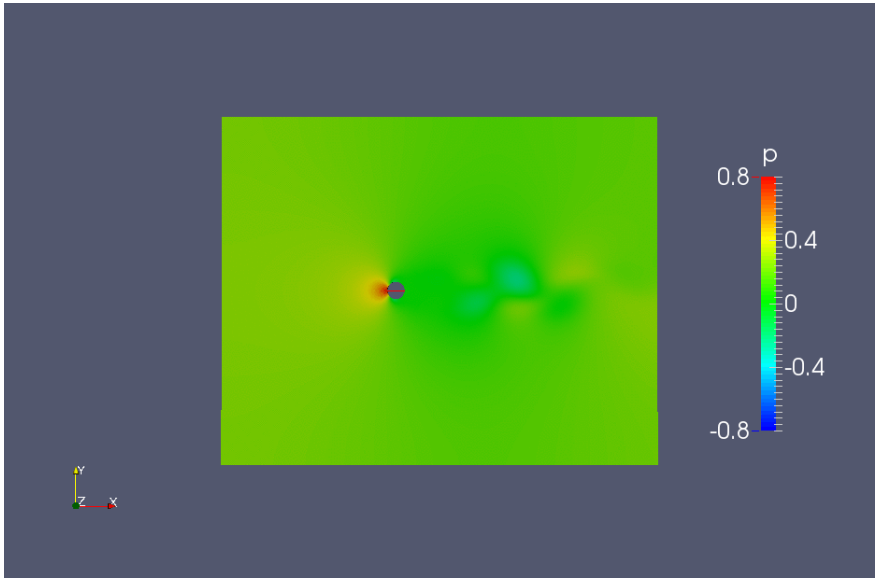
Boundary conditions and initial conditions

- And some configurations are unreliable.
  - Inlet velocity and pressure zero gradient at outlet. This combination should be avoided because the static pressure level is not fixed.
  - Qualitatively speaking the results are very different.
  - Visualization or co-processing helped us to identify a potential problem.

BCs 1.
Inlet velocity and fixed outlet pressure



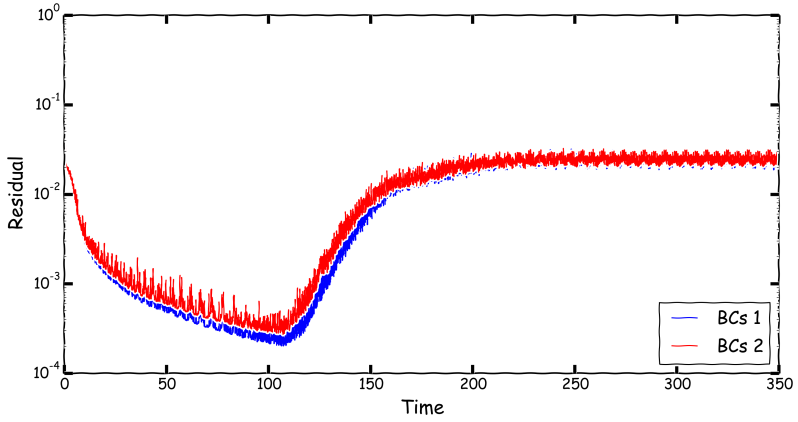
BCs 2.
Inlet velocity and zero gradient outlet pressure



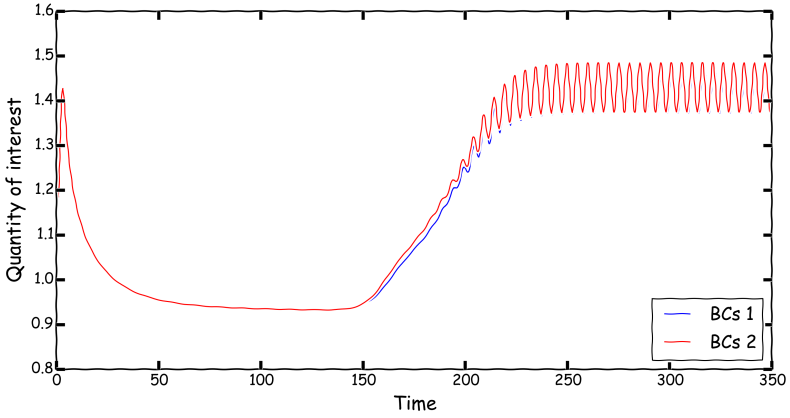
Boundary conditions and initial conditions

- And some configurations are unreliable.
  - Inlet velocity and pressure zero gradient at outlet. This combination should be avoided because the static pressure level is not fixed.
  - However, quantitatively speaking the results seem to be fine.
  - Very misleading.

Residual plot for pressure



Quantity of interest – Force coefficient on the body

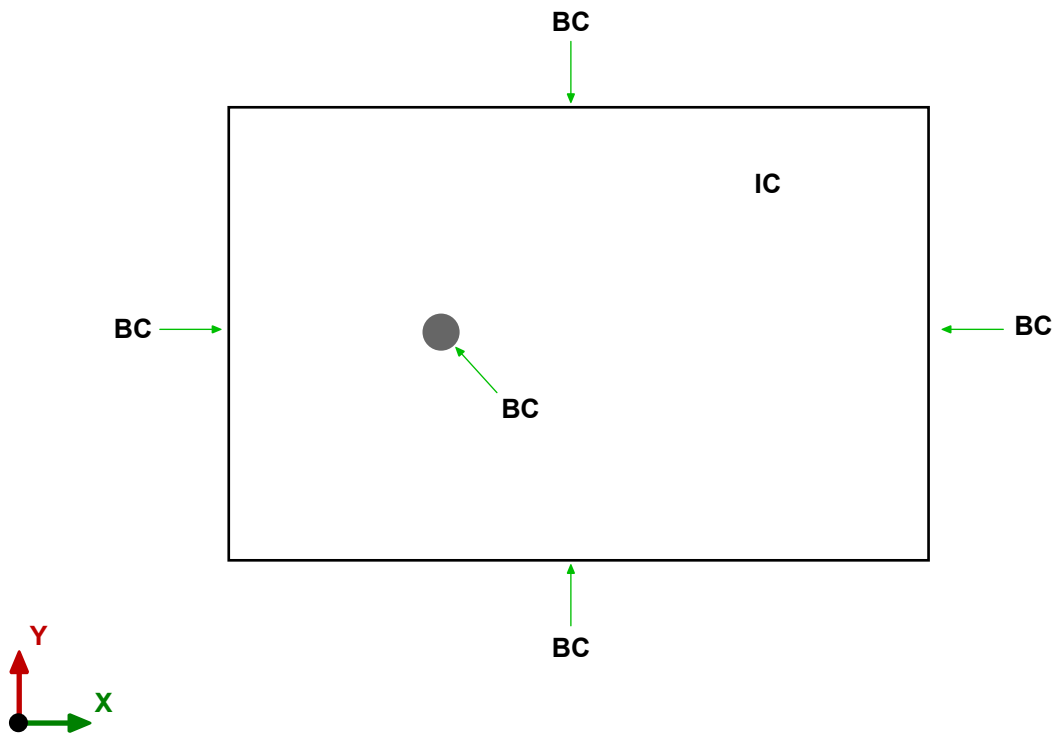


Boundary conditions and initial conditions

A mental exercise

Boundary conditions and initial conditions

- Let us do a mental exercise.
- What boundary conditions will you use for this case?
- By the way, there is no wrong or right answer (maybe there are a few bad choices), everything depends of what do you want to do.

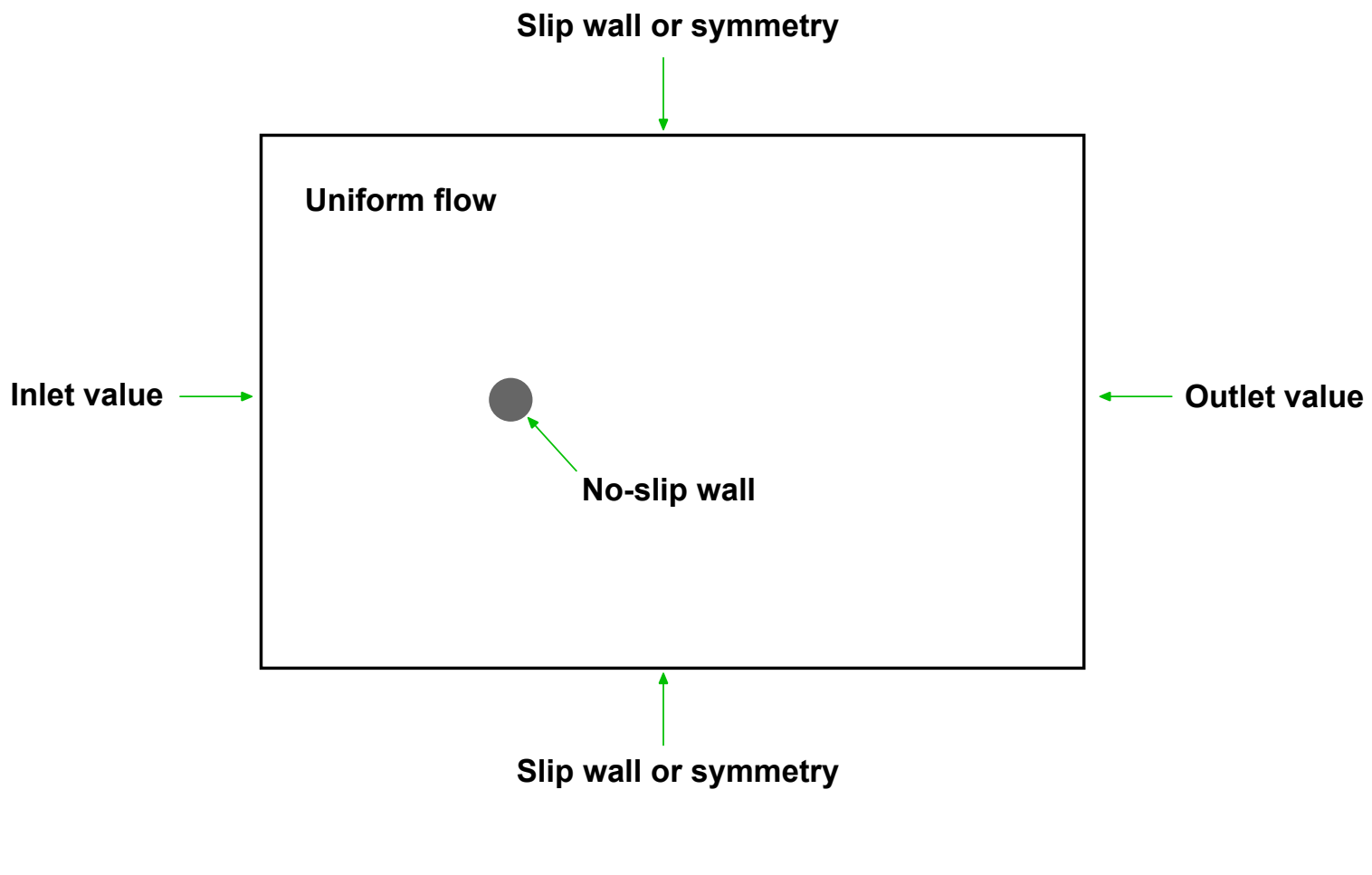


Types of boundary conditions (BC):
Dirichlet
Neumann
Robin

Initial contions (IC):
Uniform or non-uniform value in the whole domain
Patches with uniform or non-uniform values

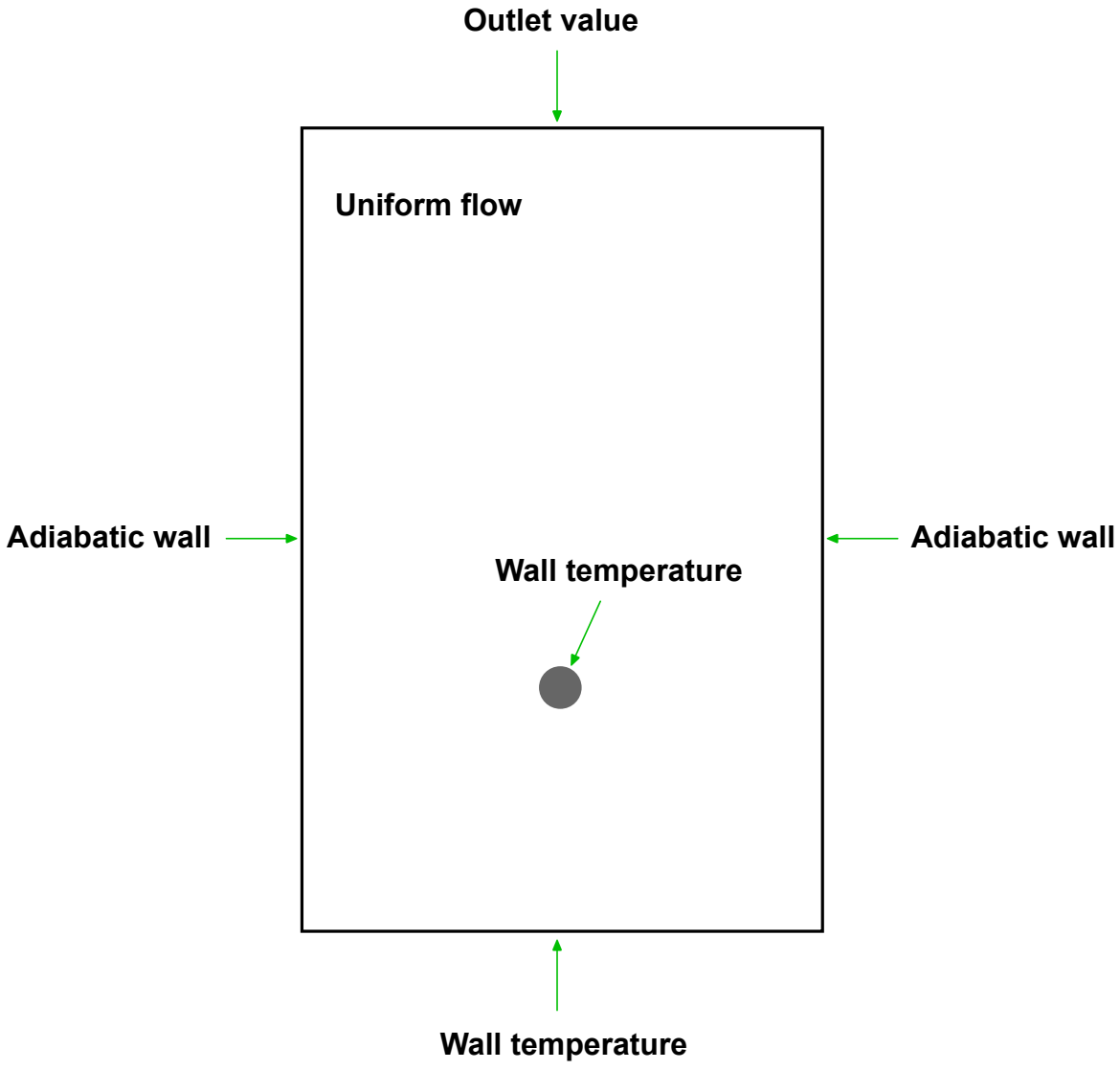
Boundary conditions and initial conditions

- One of the possible scenarios, external aerodynamics.



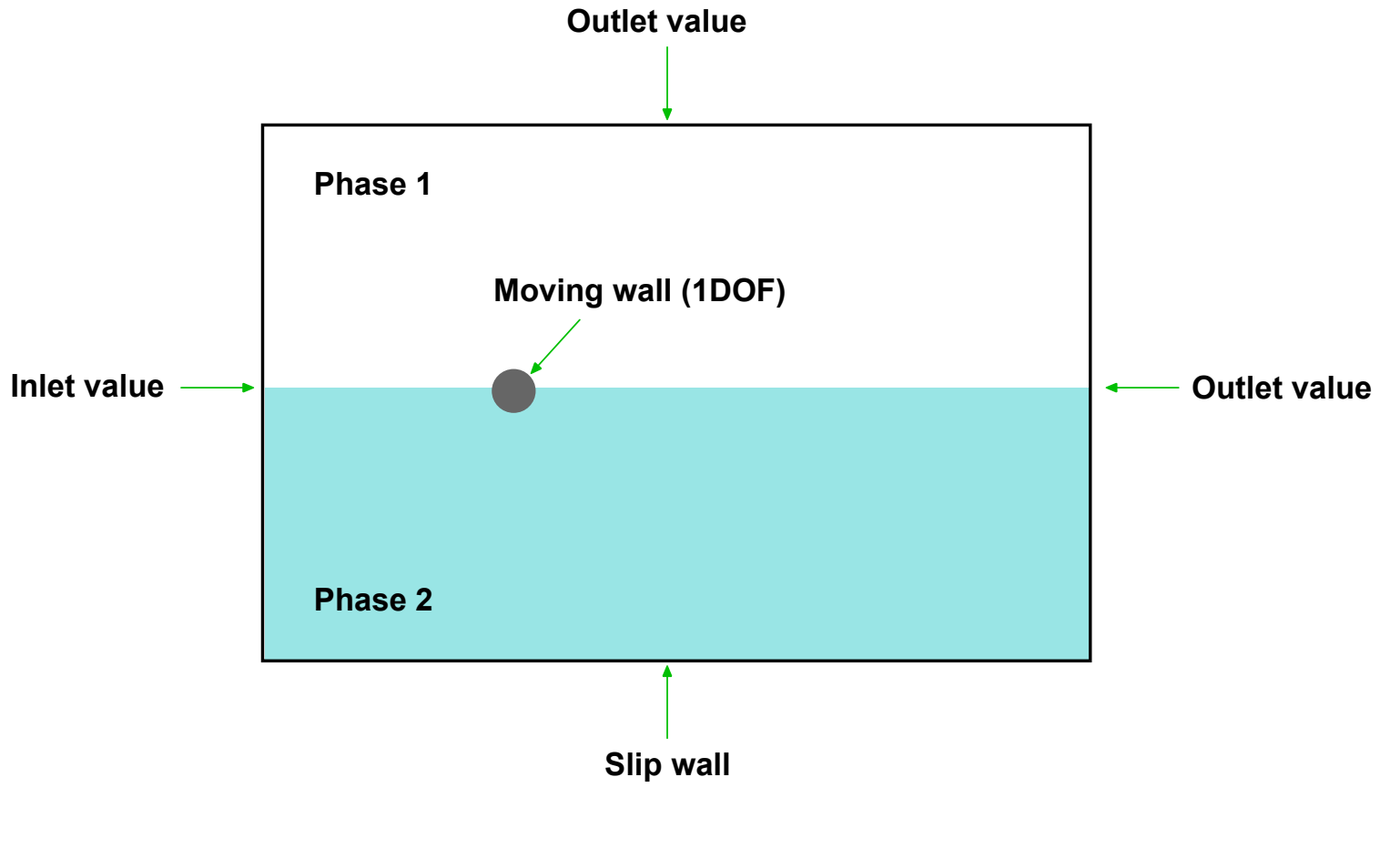
Boundary conditions and initial conditions

- Another possible scenarios, buoyant flow and a heated body.



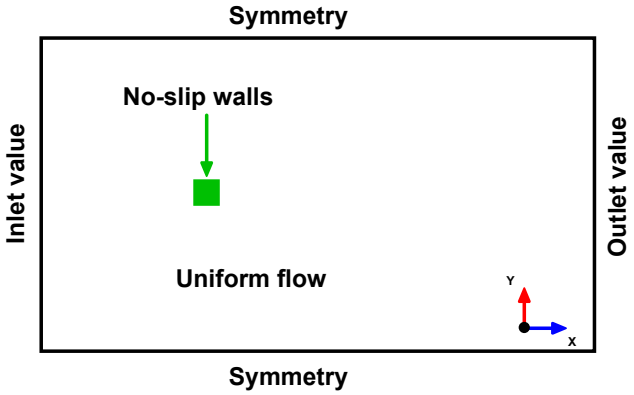
Boundary conditions and initial conditions

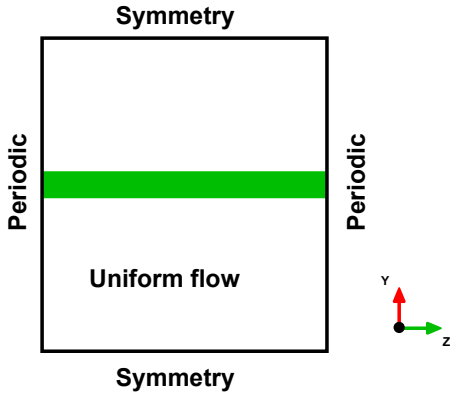
- For those interested in marine applications, a virtual towing tank and 1DOF simulation.

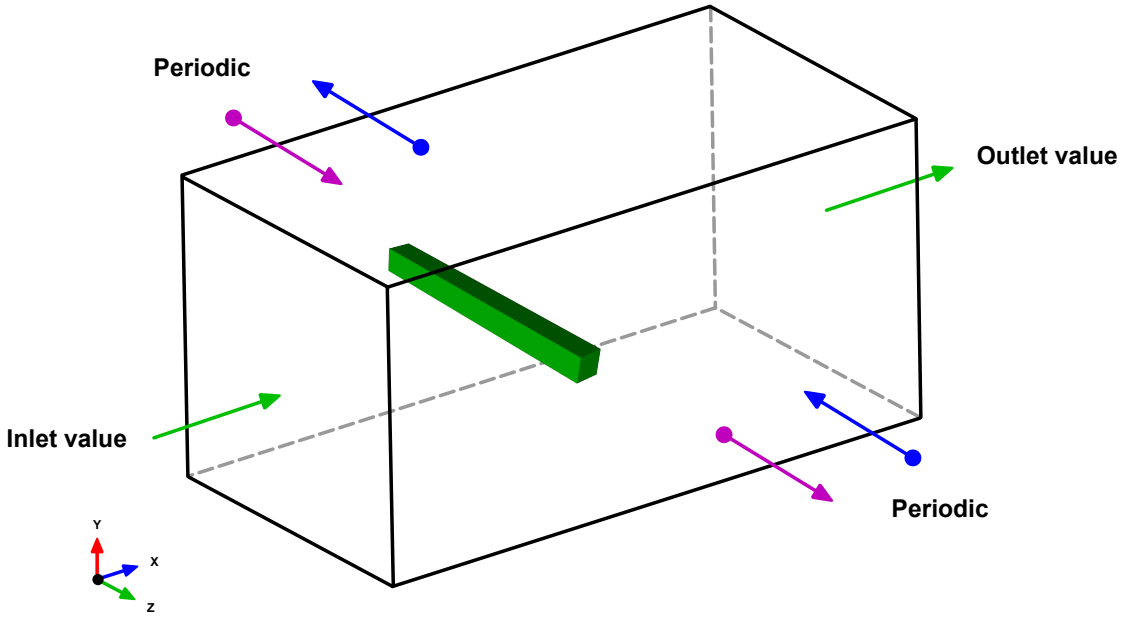


Boundary conditions and initial conditions

- And if you go to the real world (3D), nothing change. You just need to impose the boundary conditions in the third dimension.







Thank you for your attention