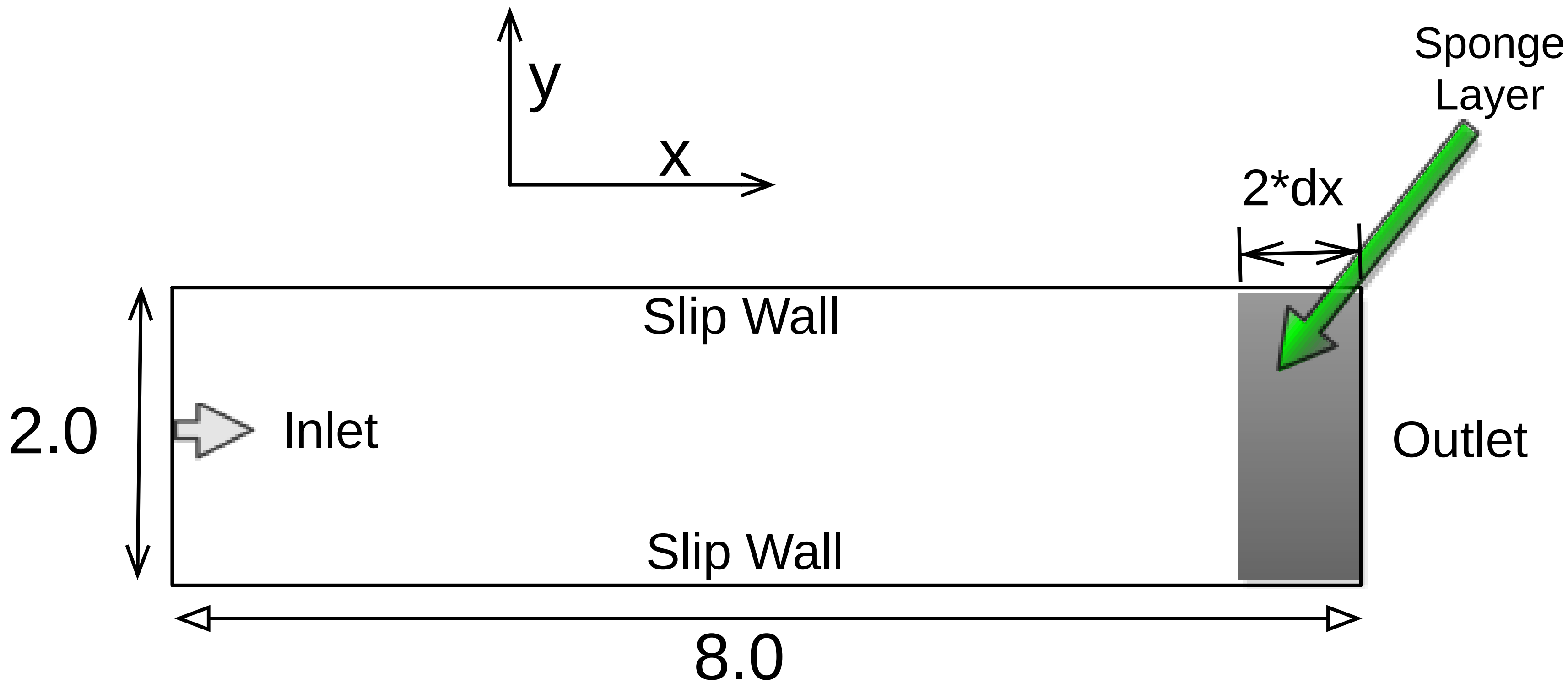
Subsonic jet test case

Plan

- Test case Subsonic jet
- Varying polynomial degrees
- Submitting jobs / Running the test case in parallel
- Visualize results (on Thursday)

Subsonic Jet



Jet radius = 0.1

Simulation Parameters

- Equation system: Euler 2D (inviscid compressible fluid)
- Scheme:
  - Spatial : Modal discontinuous Galerkin
  - Temporal : Explicit Runge-Kutta 4th-order
- Mach Number: 0.4
- Numerical Flux: HLL
- Initial Condition (in dimensionless parameters)
  - Pressure = 1.0, Density = 1.4, Velocity_X = Velocity_Y = 0
- Boundary (Outflow)
  - Inlet
    - Velocity normal = Mach * speed of sound
    - Density amplitude = 2.0

Varying Parameters

Name	Total Number of elements[1]	Order of the scheme (m+1)[2]	Total degrees of freedom
finest	65280	2	1,044,480
fine	4032	8	1,032,192
coarse	240	33	1,045,440

1 - Total number of elements can be varied with varying the level parameter in the mesh input file

2 - Order of the scheme can be set my the variable “m” (max. poly degree) in the solver input file

Output from the Simulation

- There are the following outputs:
  - `timing.res`
  - Restart-files
- `timing.res` contains the general information about the simulation. For example, number of elements, total number of dofs, etc
- The *restart-files* contain the instantaneous simulation data (coefficients of the state variables: rho, m, e), they are the basis for later visualization.

Exercise

- Set up test cases using appropriate meshes and polynomial degree, keeping roughly the same total number of degrees of freedom
  - Use pre-generated mesh data stored in the directories
    - coarse
    - fine
    - finest
  - Change polynomial degree in the input file
    - m=1 (for the finest mesh)
    - m=7 (for the fine mesh)
    - m=32 (for the coarsest mesh)
  - You might also try other settings like m=15 for the coarsest

Workflow (example for coarse mesh)

- Copy the input files to your personal directory:
  `cp -r $KURS/exercises/hpcfdx8 $MYWS`
- Create a setup directory (some name to your liking):
  `mkdir -p $MYWS/hpcfdx8/mysetup`
- Change into that directory:
- `cd $MYWS/hpcfdx8/mysetup`
- Copy ateles.lua into that directory:
  `cp ../ateles.lua .`
- Adapt the config file for your setup:
  `gedit ateles.lua`

Workflow contd.

- Look for the variable "scheme" and set the spatial order of the scheme by defining the maximum polynomial degree (m). (Order of scheme = m+1)

```
scheme = {
  spatial = {
    name = 'modg_2d',
    m    = 32,
  },
  … …
}
```

```
coarse: m = 32
fine:   m = 7
finest: m = 1
```

Workflow contd.

- To submit the jobs use the job scripts present
```
cp ../jet.job .
gedit jet.job
```
- You can set a name for this job
```
#SBATCH --job-name=<Job_name>
```
- And modify its node-count and walltime
```
#SBATCH --nodes=<nNode>
#SBATCH --time=<hours:minutes:seconds>
```
- Submit the job using the "sbatch" command
```
sbatch jet.job
```

Checking the configuration

- It is a good idea to check up on your configuration and make sure that it works!
- Check the log output and the restart directory to make sure everything works as expected.

Stopping a running simulation

- Simulations are set up to cover 30 time units of simulated time
- This will take quite some compute time
- There may be less nodes available, and you might want to reduce the requested number of nodes to fit a free slot

- You can signal a running simulation to stop, by creating an empty file named ‚stop' in the working directory:
  `touch stop`