How to simulate a Plasmatron test with COOLFluiD

N. Villedieu and A. Lani

April 24, 2013

Outline

How to install COOLFluiD?

Loading of the sources and compilation

- The first step is ALWAYS: `module load cf2-2012.3/lammpi`
- To get the sources:
  - For the kernel (part of the code that is common to every applications)

```
svn co
   https://coolfluidsrv.vki.ac.be/svn/
       coolfluid/Sources/Kernel/trunk COOLFLUID
```

  - For the plugins:
    1. Adapt the coolfluid.conf and put it in the COOLFLUID directory
    2. `cd COOLFLUID`
    3. `./prepare.pl -mods-up`
  - Compilation:
    1. `./prepare.pl -build=release`
    2. `cd builds/x86_64/release/`
    3. `make -j6`

How to install COOLFluiD

Adaptation of coolfluid.conf

- Path of the two first lines:

```
coolfluid_dir=/villedie/workspace/COOLFLUID
basebuild_dir=/villedie/workspace/COOLFLUID/
                                    builds/x86_64
```

- The path to the dependencies

```
mpi_dir=/ardisksrv1/projects/cf2/2012.3/lam
```

The special case of Rocks5

- Read the wiki about rocks5
- Get a node `qcompile`
- Follow the steps mentioned in slide 1

How to run COOLFluiD

- `module load coolfluid2/2010.0`
- If needed `lamboot`
- Create symbolic links to the executable

```
ln -sf /villedie/workspace/COOLFLUID/
      builds/x86_64/release/src/
      Solver/coolfluid-solver .
ln -sf /villedie/workspace/COOLFLUID/
      builds/x86_64/release/src/
      Solver/coolfluid-solver.xml .
```

- Run
  - In serial:

    `./coolfluid-solver --scase ./torch_4_res.CFcase`
  - In parallel (it may be necessary to do lamboot first) :

```
mpirun -np 2 ./coolfluid-solver --scase
                              ./torch_4_res.CFcase
```

How to simulate a Plasmatron test

What can we simulate?

- It is possible to simulate the torch+chamber+probe
- The flow is considered as LTE

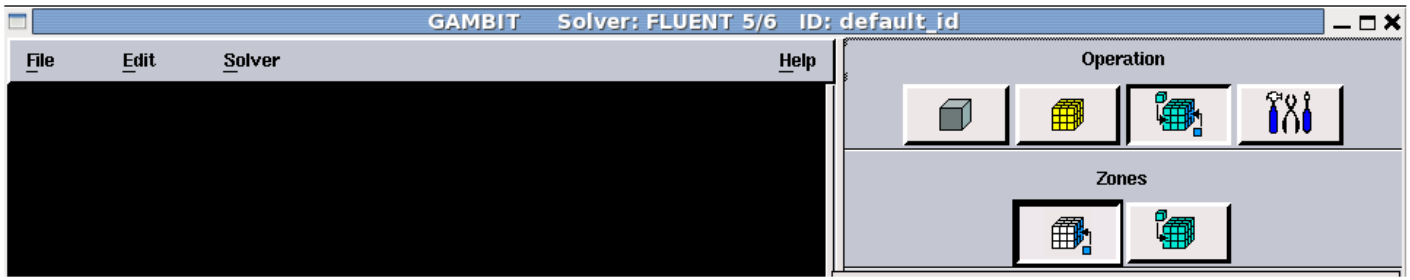
Steps

1. Doing the mesh
2. Simulation of the torch with simple Euler boundary conditions
3. Torch with full boundary conditions
4. Extrapolation of the torch solution in the chamber
5. Chamber with 1^{st} order scheme
6. Chamber with 2^{nd} order scheme

How to do the mesh

Few instructions for Gambit

- It is advisable to do the mesh on a scaled ($\approx$ 100) geometry
- You should register all the boundaries:
  - Click Solver $\rightarrow$ generic
  - Click the green button on the left (in Operation) $\rightarrow$ white button in zones
  - Register all your boundaries name
- Save the mesh in .neu: Click File $\rightarrow$ export $\rightarrow$ mesh and give a name



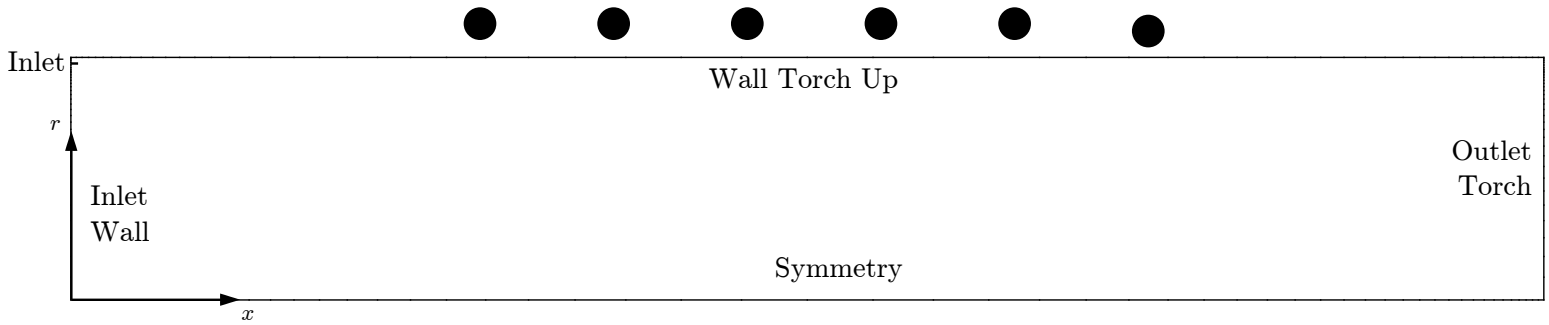
How to do the mesh

You need to do two meshes

- One for the torch
- One for the torch+chamber+probe: The torch block should keep the same

Mesh requirements for the torch

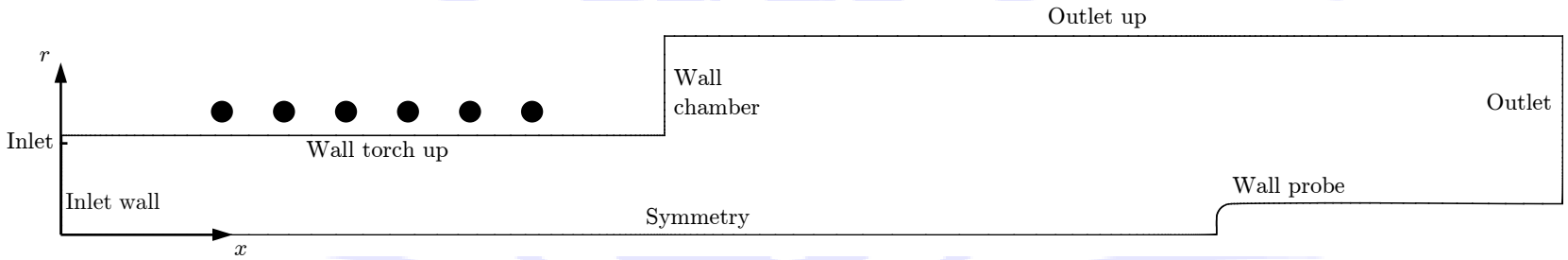
- The mesh should not be very fine: $1800 < \sharp\text{nodes} < 2500$
- It should be refined near the walls and the outlet



Meshing Torch+chamber+probe

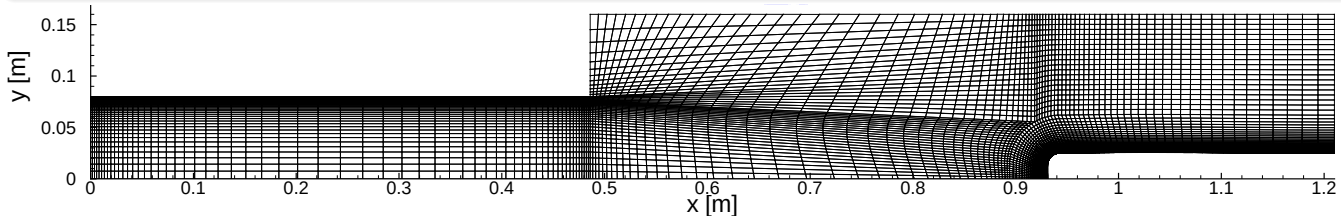
Definition of the domain

- Only the part of the chamber close to the probe is modeled
  - Probe at $x = 445mm \Rightarrow$ domain is cut at $x = 1210mm$
  - Ray of the probe $25mm \Rightarrow$ the chamber is cut at $y = 160mm$



Doing the mesh

- Should start from the torch mesh
- Refine near the probe (mesh size around $5\mu m$)
- Coarsening around the top boundary



First phase of the simulation

torch_4.CFcase

The CFcase is divided in two parts:

- First part, does not need to be changed (setup of FV, LTE...)
- Second part, needs to be adapted to your case

Loading the mesh

```
Simulator.SubSystem.Default.listTRS = InnerFaces...
Simulator.SubSystem.CFmeshFileReader.Data.FileName =
                                        torch_4.CFmesh
```

- The extension should be **.CFmesh**
- If the Gambit file was done with a scaling:

```
Simulator.SubSystem.CFmeshFileReader.Data.
                                    ScalingFactor=1000
```

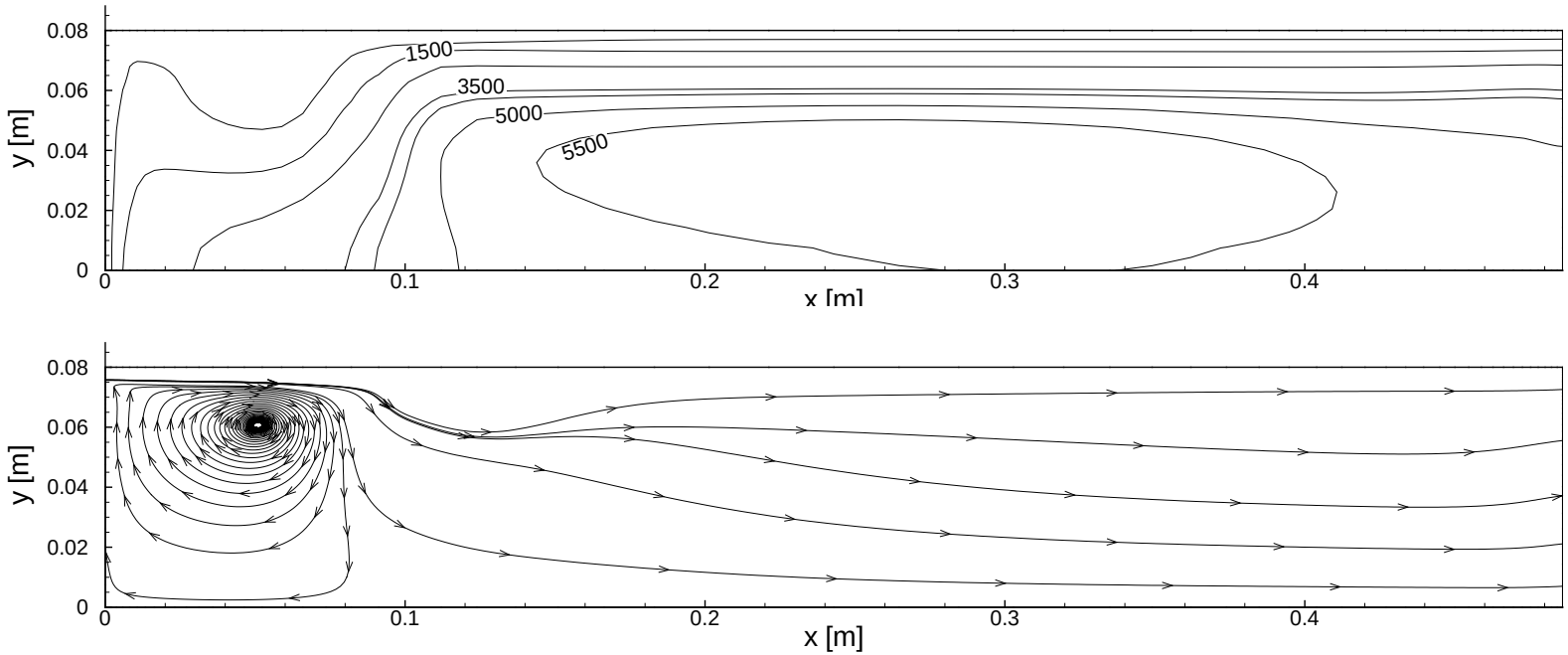
First phase of the simulation

Stopping the simulation

It is possible to setup a condition on the residual:

```
Simulator.SubSystem.StopCondition=Norm
Simulator.SubSystem.Norm.valueNorm=-2.0
```

- Here it should be stopped it by hand (Clt + C) when temperature and streamlines are as follows (around 500-1500 iterations).



First phase of the simulation

CFL

For this phase the best is to use an interactive CFL

```
Simulator.SubSystem.NewtonIterator.Data.CFL.
                                    ComputeCFL=Interactive
Simulator.SubSystem.NewtonIterator.Data.CFL.
                                    Interactive.CFL=0.0001
Simulator.SubSystem.InteractiveParamReader.
                                    FileName=./torch.inter
Simulator.SubSystem.InteractiveParamReader.readRate=5
```

The CFL should change slowly

Outputs

It is possible to use several outputs: COOFluiD, tecplot, paraview....

```
Simulator.SubSystem.OutputFormat=Tecplot CFmesh
```

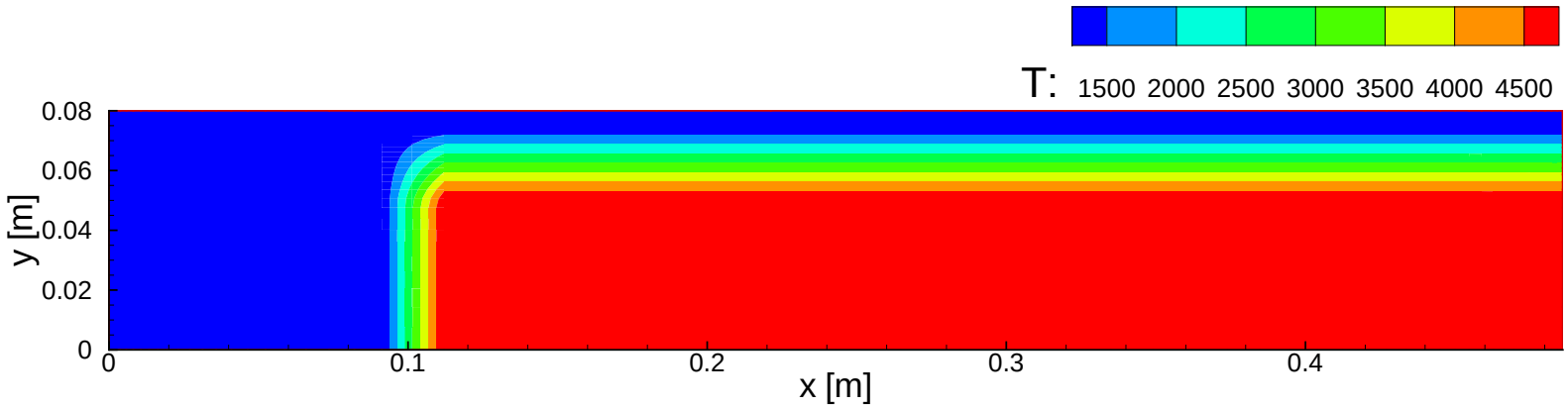
You can define file names....

```
Simulator.SubSystem.CFmesh.FileName=torch_4-out.CFmesh
Simulator.SubSystem.Tecplot.SaveRate=50
Simulator.SubSystem.Tecplot.Data.SurfaceTRS=Wall_torch_in
```

First phase of the simulation

Initialization

```
Simulator.SubSystem.CellCenterFVM.InitComds=InitState
Simulator.SubSystem.CellCenterFVM.InitNames=InField
Simulator.SubSystem.CellCenterFVM.InField.applyTRS
                                        =InnerFaces
Simulator.SubSystem.CellCenterFVM.InField.Vars=x y
Simulator.SubSystem.CellCenterFVM.InField.Def=0.   \
          if(y>.075,if(y<.08,100.,0.),if(x>.2,0.,0.))\
                                  0.\
                                  ...
```



First phase of the simulation

Extrapolation

- Extrapolation from cell centers to nodes for the viscous gradients and visualization

```
Simulator.SubSystem.CellCenterFVM.Data.
   NodalExtrapolation=DistanceBasedGMoveMultiTRS
Simulator.SubSystem.CellCenterFVM.Data.
   DistanceBasedGMoveMultiTRS.TrsPriorityList=Wall_torch...
Simulator.SubSystem.CellCenterFVM.Data.
   DistanceBasedGMoveMultiTRS.TRSName=Wall_torch...
```

- Impose strongly wall temperature
- Impose strongly electro-magnetic field to zero (only in this transient phase)

```
Simulator.SubSystem.CellCenterFVM.Data.
  DistanceBasedGMoveMultiTRS.Wall_torch_in.
                                ValuesIdx=1 2 3 4 5
Simulator.SubSystem.CellCenterFVM.Data.
    DistanceBasedGMoveMultiTRS.Wall_torch_up.
                                Values=0 0 350 0 0
```

First phase of the simulation

Boundary conditions

- List of the commands:

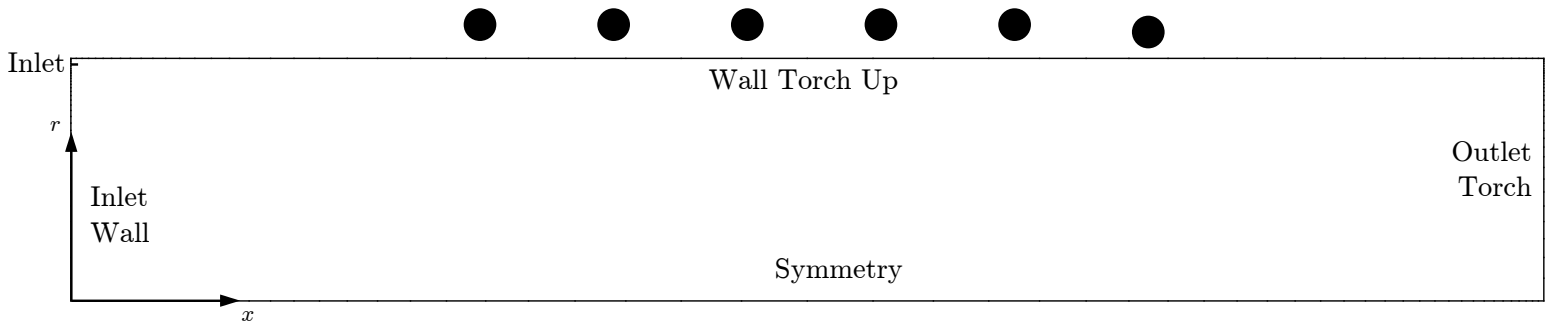
```
Simulator.SubSystem.CellCenterFVM.BcComds=MirrorICPFVMCC
```

- List of names:

```
Simulator.SubSystem.CellCenterFVM.BcNames=BcTorchWallUp
```

- TRS where it is applied:

```
Simulator.SubSystem.CellCenterFVM.BcOutletTorch
                                 .applyTRS=Outlet_torch
```



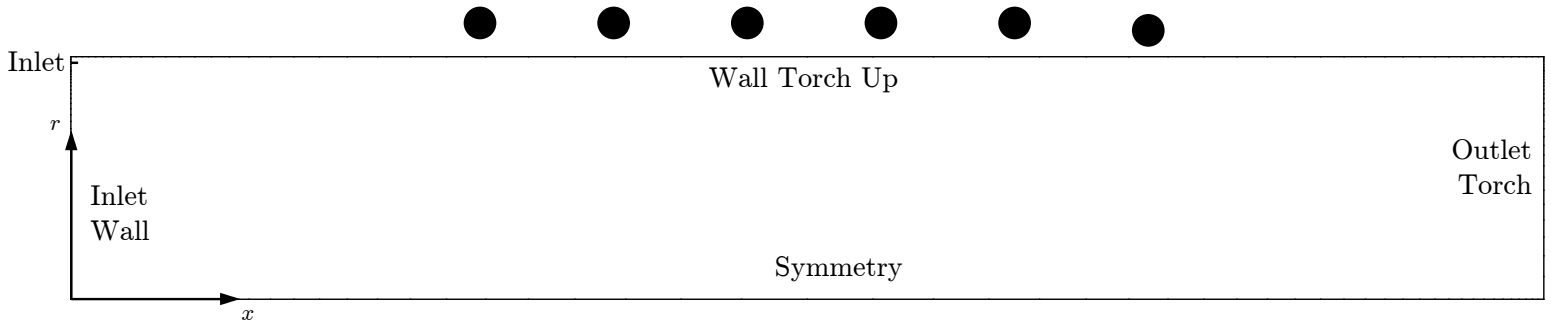
First phase of the simulation

Outlet Torch: `SubOutletICP2DPuvtFVMCC`

- At the Outlet we impose no pressure perturbations:
  `Simulator.SubSystem.CellCenterFVM.BcOutletTorch.P=0.0`

Walls: `NoSlipWallIsothermalICPPvtFVMCC`

- At the wall we impose isothermal temperature:
  `Simulator.SubSystem.CellCenterFVM.BcTorchWallUp.TWall=350.`



First phase of the simulation

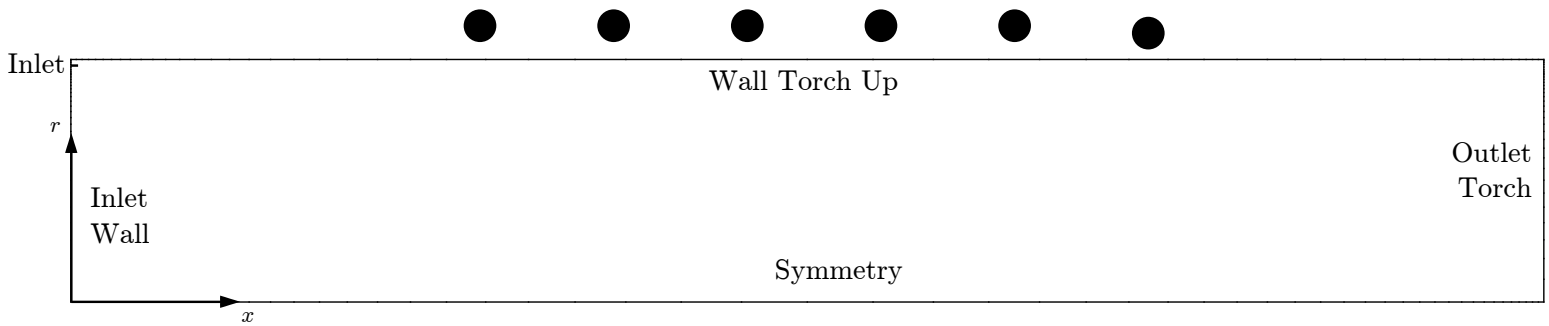
Inlet:`SubInletICP2DPuvtUVTFVMCC`

- At the inlet we imposse the mass flow, temperature and the geometry is defined

```
Simulator.SubSystem.CellCenterFVM.BcInlet.MassFlow=16.
Simulator.SubSystem.CellCenterFVM.BcInletT=350.
Simulator.SubSystem.CellCenterFVM.BcInlet.
                                        InletRadii=.075 .08
```

Symmetry line:`MirrorICPFVMCC`

All radial gradients are set to zero



First phase of the simulation

Definition of the coils and current

- Definition of the PLASMA POWER

```
Simulator.SubSystem.DataProcessing.JouleHeatSource.
          DesiredPower=90.
```

- Definition of the setup of the coils

```
Simulator.SubSystem.DataProcessing.JouleHeatSource.
          NbCoils=6
Simulator.SubSystem.DataProcessing.JouleHeatSource.
          RadiusCoils= .109 .109 .109 .109 .109 .109
Simulator.SubSystem.DataProcessing.JouleHeatSource.
          ZPositionCoils=.127 .177 .227 .277 .327 .377
```

- Writing the Joule Heat in a file:

```
Simulator.SubSystem.DataProcessing.JouleHeatSource.
          OutputFileElCurrent=.\elCurrent.plt
```

Second phase

Differences with the first phase

In this phase the electromagnetic field is imposed to zero in the far-field by the boundary conditions.

Setup

- We use the output of the previous simulation to restart the simulation:

  ```
  Simulator.SubSystem.CellCenterFVM.Restart=true
  ```

- The Boundary conditions also modify the EM:

  ```
  Simulator.SubSystem.CellCenterFVM.BcComds
                =EpComputingNoSlipWallIsothermalICPPvtFVMCC
  ```

Second phase

CFL

The CFL can be increased faster:

```
Simulator.SubSystem.NewtonIterator.Data.CFL.Value=0.01
Simulator.SubSystem.NewtonIterator.Data.CFL.
          ComputeCFL=Function
Simulator.SubSystem.NewtonIterator.Data.CFL.
          Function.Def=if(i>80,5,1)
```

Iteration	**CFL**
$1 \sim 10$	0.01
$10 \sim 45$	0.1
$45 \sim 100$	1
$100 \sim 295$	10
> 295	100

Extrapolation

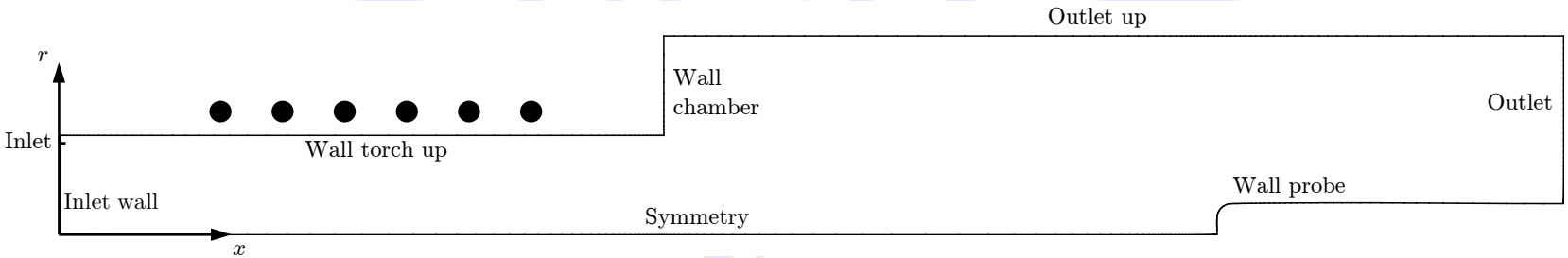
- The solution in the torch is used to approximate the solution in the chamber

Installing the code

- You need to change the path in:
  - build/Makefile
  - build/CMakeFiles/initialcond.dir/depend.intern
  - build/CMakeFiles/initialcond.dir/DependInfo.cmake
  - build/CMakeFiles/initialcond.dir/build.make
- Adapt mesh names in main.cpp
- compile
- Run: `build/initialcond`

First Order Chamber

- The CFcase for the full geometry is very similar to the one of phase two, the main changes are:
  - The TRS names and mesh name
  - The boundary conditions:
    - Wall Chamber ⇒ Isothermal wall (with an eventual blowing)
    - Outlet up ⇒ SubOutlet
    - Outlet ⇒ SuperOutlet
    - Wall Probe ⇒ Isothermal Wall



Second Order Chamber

- The second order simulation should start from the 1^{st} order one

What changes

- The polynomial reconstruction:

```
Simulator.SubSystem.CellCenterFVM.Data.
                              PolyRec=LinearLS2D
Simulator.SubSystem.CellCenterFVM.Data.
                              LinearLS2D.gradientFactor=1.
```

- Add a limiter (because of the high gradients in temperature)

```
Simulator.SubSystem.CellCenterFVM.Data.LinearLS2D.
                              limitRes=-15.
Simulator.SubSystem.CellCenterFVM.Data.
                              Limiter=Venktn2D
Simulator.SubSystem.CellCenterFVM.Data.Venktn2D.
                              coeffEps=1.0
```

- Add a condition on gradients at the boundaries

```
Simulator.SubSystem.CellCenterFVM.BcTorchWallIn.
                              ZeroGradientFlags=1 0 0 0 1 1
```

Remarks

Is it necessary to redo this always?

- For a new geometry: YES
- For a geometry where solutions exists:
  - For change of pressure around $2000 \sim 4000 Pa$: NO
    You just need to adjust the pressure and velocity
    ```
    Simulator.SubSystem.ICPLTE2D.refValues = 1.  100.
    100.  10000.  0.01 0.01
    Simulator.SubSystem.ICPLTE2D.ConvTerm.uInf = 100.
    Simulator.SubSystem.ICPLTE2D.ConvTerm.p0Inf =
    10000.
    ```
  - For change of the power $30 \sim 40 kW$: NO
  - Otherwise YES

Remarks

Possible issues

It is possible that few problems occurs:

- Problems around walls:
  - Near the probe: need for refinements
  - Near Inlet wall: use adiabatic wall
- Creation of a recirculation on the top region: impose a small horizontal velocity out of the jet

Conclusion

- If you want to understand better I advise you to read:
  - The report of Sartori: VKI SR 2010-01.
  - The report of PerisNavarro: SR 2011-22.
- From the ICP computation, people usually want to compute the NDP's For this I porpose to do another lecture