"Is it a solver? Is it a function object? It's a region solver"

A new feature in swak4foam

Bernhard F.W. Gschaider

HFD Research GesmbH

Virtual Linz
16. October 2020

Outline

1. Introduction
   - This presentation
   - Who is this?
   - Adaptive boundary conditions
2. What is already there
   - OpenFOAM
   - swak4Foam
3. simpleRegionSolvers
   - The library
   - Set-up
4. Examples
   - Adding heat conduction
   - Flow boundary conditions
   - Structure mechanics
5. The End
   - Limitations and outlook
   - Goodbye

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

What it's about

- Topic of the presentation is a new part of `swak4Foam`
- It allows using solvers as boundary conditions
  - Build simple "multi-physics" cases

Outline

1. Introduction
   - This presentation
   - Who is this?
   - Adaptive boundary conditions
2. What is already there
   - OpenFOAM
   - swak4Foam
3. simpleRegionSolvers
   - The library
   - Set-up
4. Examples
   - Adding heat conduction
   - Flow boundary conditions
   - Structure mechanics
5. The End
   - Limitations and outlook
   - Goodbye

Bernhard Gschaider

- Working with OpenFOAM™ since it was released
  - Still have to look up things in Doxygen
- I am not a core developer
  - But I don't consider myself to be an *Enthusiast*
- My involvement in the OpenFOAM™-community
  - Janitor of the `openfoamwiki.net`
  - Author of two additions for OpenFOAM™

    swak4foam Toolbox to avoid the need for C++-programming

    PyFoam Python-library to manipulate OpenFOAM™ cases and assist in executing them
  - Organizing committee for the OpenFOAM™ *Workshop*
- The community-activies are not my main work but *collateral damage* from my real work at ...

Heinemann Fluid Dynamics Research GmbH

The company

- Subsidary company of *Heinemann Oil*
  - Reservoir Engineering
  - Reservoir management

Description

- Located in Leoben, Austria
  - . . . and Vienna
- Works on
  - Fluid simulations
    - OpenFOAM™ and Closed Source
  - Software development for CFD
    - mainly OpenFOAM™
- Industries we worked for
  - Automotive
  - Processing
  - . . .

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Boundary conditions with physics

- Usually the boundary conditions we set are "one way"
  - "Hey. This is the value/gradient" adapt to it
- But sometimes the boundary is influenced by the fluid
  - Walls heat up and conduct to other places
  - Things move
  - The pressure influences the velocity at the inlet
    - There is no `uniform` velocity in real life

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 `simpleRegionSolvers`
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Outline

1. Introduction
   - This presentation
   - Who is this?
   - Adaptive boundary conditions
2. What is already there
   - OpenFOAM
   - swak4Foam
3. simpleRegionSolvers
   - The library
   - Set-up
4. Examples
   - Adding heat conduction
   - Flow boundary conditions
   - Structure mechanics
5. The End
   - Limitations and outlook
   - Goodbye

Inhomogenous boundary conditions

- OpenFOAM has multiple ways to specify non-uniform boundary conditions
  - `coded` if you like C++ programming
  - reading data from files and playing it back
- All these are fine if you know beforehand how the boundary will look like

The thermal baffle

- The `thermalBaffle` condition allows simulating a conducting wall
  - Specified as a boundary condition
  - OpenFOAM automatically generates a mesh by extruding from the patch
  - User has to specify thermal properties
  - Heat conduction is solved in the sheet
    - Heat is exchanged with the fluid via the boundary conditions
- Disadvantages:
  - Geometry is limited to sheets
  - Boundary conditions are predefined (limits applications)

The *conjugate heat transfer* solver

- The `chtMultiRegionFoam` allows simulating a system with multiple regions
  - Of compressible, turbulent fluids
  - Heat conducting solids
- All these exchange heat via boundary conditions
  - Explicitly coupled
- Fine if no additional physics is needed
  - Multiphase or combustion would require writing a completely new solver

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

What is swak4Foam

From https://openfoamwiki.net/index.php/Contrib/swak4Foam

swak4Foam stands for SWiss Army Knife for Foam. Like that knife it rarely is the best tool for any given task, but sometimes it is more convenient to get it out of your pocket than going to the tool-shed to get the chain-saw.

- It is the result of the merge of
  - funkySetFields
  - groovyBC
  - simpleFunctionObjects

  and has grown since
- The goal of swak4Foam is to make the use of C++ unnecessary
  - Even for complex boundary conditions etc
- We'll use here. But there will be no deeper explanation

groovyBC

- Part of `swak4Foam` that allows using expressions for boundary conditions
  - the are non-uniform in space
  - change in time
- Allow storing values
  - this "memory" allows pseudo-physical behavior

If it is an expression swak4Foam can do it

- swak4Foam has
  - function objects
  - fvOption-objects
- that allow manipulating the solution
- all of them controlled by expressions
- but if we can't write it as an expression they don't help

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 `simpleRegionSolvers`
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Solver as a function object

- Idea of the `simpleRegionSolvers`-library is
  - Re-implement a solver
    - Works on a separate mesh
  - Wrap the implementation into a function object
    - The solver is called at the end of every time-step
  - The main region and the separate region exchange information via boundary conditions
    - This requires `mapped` patches
- Currently there are 4 solvers in the library
  - Each of them shamelessly stole from an existing solver

heatConductionRegionSolver

- Solves the heat conduction in a solid
- Re-implements the solid part of `chtMultiRegionFoam`
  - But uses the temperature instead of the energy in the equation
    - Using the energy is not a good idea with non-homogeneous conductivities
- This was the first solver implemented for this library
  - . . . because I needed it to at conjugate heat transfer to a solver that was already complicated

rhoPimpleFlowRegionSolver

- Solves a compressible, turbulent fluid
  - Transient (but can be used steady as well)
- Re-implements `rhoPimpleFoam` form OF v1906
  - Moving mesh-parts have been removed

potentialFlowRegionSolver

- Solves the potential flow equation
- Re-implements `potentialFoam`
  - Which is the same in all OpenFOAM-versions since . . . forever

solidDisplacementRegionSolver

- Solves structural equations in a solid
  - Stress and displacement
- Re-implements solidDisplacementFoam from v1906
  - The tractionDisplacement-boundary condition is ported as well as it is hard-linked to solidDisplacementFoam
  - a groovyTractionDisplacement allows calculating the pressure and the traction

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Adding a solver

This needs two things in the `controlDict`

1. load the library
2. add a function object
   - the only required parameter is `meshRegion` with the name of the region the solver works in

constrolDict

```
libs (
    "libsimpleRegionSolvers.so"
);
functions
{
    inChannel {
        type rhoPimpleFlowRegionSolver;
        meshRegion inChannel;
    }
}
```

Creating a region

This is the hard part

- A sub-directory with the name of the `meshRegion` has to be added and filled with the necessary files in

0	fields that the solver needs
constant	material properties etc
system	`fvSolution` and `fvSchemes`

- It is almost like setting up a separate case

Mesh setup

- A mesh has to be constructed inside `constant/<meshRegion>/polyMesh`
  - This depends on your meshing workflow
- Boundaries that should exchange with other regions have to be `mapped` or `mappedWall`
  - Easiest in `blockMeshDict`

A patch in `blockMeshDict`

```
boundary
(
    inlet
    {
        type mappedPatch;
        faces
        (
            (0 1 12 11)
        );
        sampleMode      nearestPatchFace;
        sampleRegion    inChannel;
        samplePatch     inChannel_to_pitz;
    }
```

Boundary conditions

- The mapped-patch allows copying/interpolating values from the associated patch in the other mesh
  - This is built into OpenFOAM
  - The patches can have different face sizes

0/U

Getting the velocity from the other mesh

```
boundaryField
{
    inlet
    {
        type mapped;
        setAverage 0;
        average (10 0 0);
        value           uniform (10 0 0);
    }
```

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Adding a full baffle

- This is a tutorial case for `reactingFoam`
  - Methane goes into a chamber
  - Mixes with air
  - Burns
- The inlet is a pipe that goes into the chamber
  - one side faces the fuel inlet the other the chamber
    - these sides can have different temperatures
  - the tutorial assumes that the wall is adiabatic
- In this demo a solid region was added for the pipe wall
  - Heat conduction solved in that solid
  - If the pipe-wall is heated on one side (chamber for instance) the heat will propagate to the other side (fuel inlet)
- The following picture show the inlet region

DLR_A_LTS Result

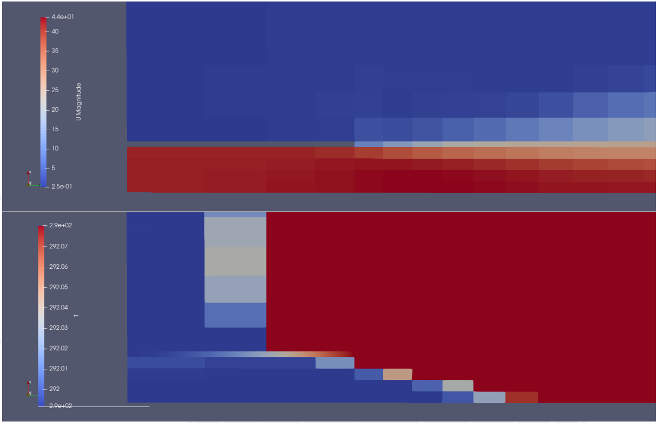

Figure: Temperature and Velocity

The baffle makes no difference

. . . . for this case

- In this case this doesn't make a big difference
    - The flame is far from the pipe
        - And there is no radiation
    - Temperature differences in the solid far less than 1 degree Celsius
        - So the adiabatic BC of the tutorial is justified
- But I had cases where it made a difference

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Improved inlet turbulence for a LES case

- This is based on a LES-tutorial for `rhoPimpleFoam`
  - The popular `pitzDaily`-case
- The inlet pipe of the original case is rather short
  - To develop a proper velocity and turbulence profile
- In this demo a pipe was added to the inlet
  - With a coarser mesh
  - Flow is solved with a $k - \epsilon$ model there
  - The outlet values for the velocity and k are used as an inlet condition for the "proper" simulation

LES with a RAS inlet

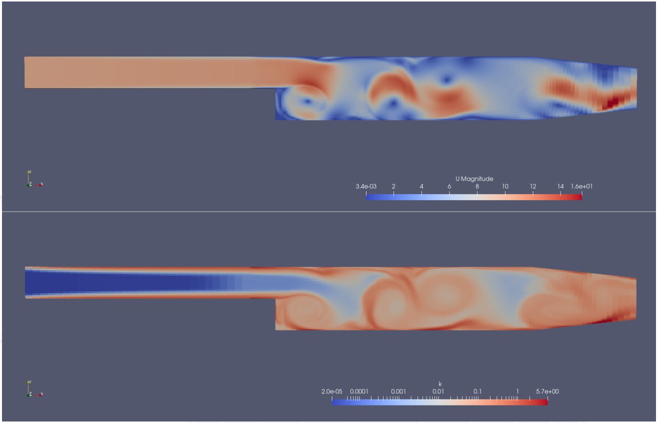

Figure: Velocity and turbulence

That looks strange

- We see a "spike" in the values of k on the wall at the beginning of the proper case
    - Wht is the reason?

Reason for the jump on the boundary

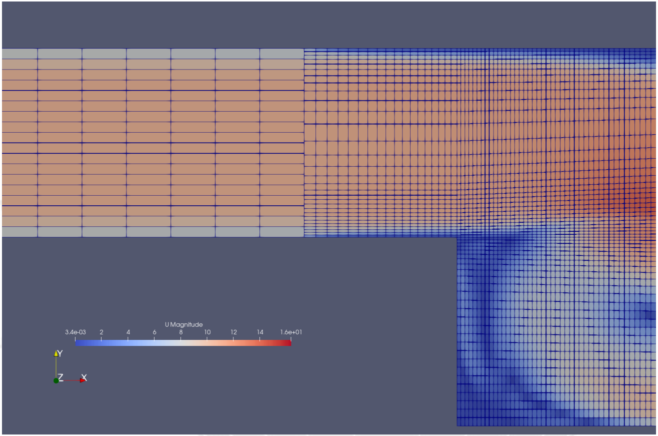

Figure: Different mesh resolutions

Removing uniform inlet or outlet

- To avoid artificial `uniform`-ness on the inlet and outlet of the `pitzDaily` case we use "simpler" flow solvers:
  - potential flow
    - the outlet velocity is used as an inlet velocity for one of them. The pressure calculated there is used instead of the `uniform` pressure
    - the outlet velocity of the other potential flow is used as the inlet condition for the central region
- The tutorial case for `simpleFoam` is used

A potential flow at each end

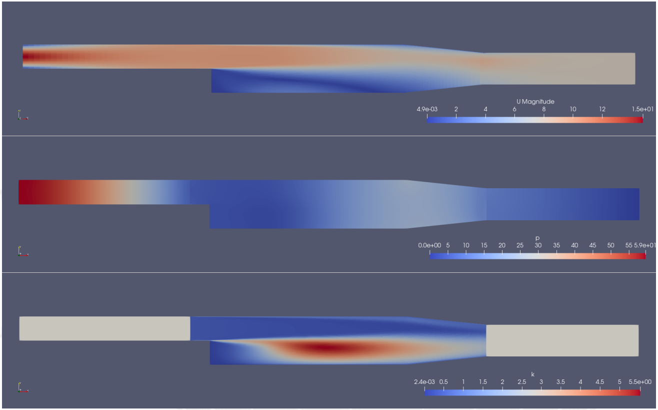

Figure: Velocity, pressure and turbulence

Why not make the inlet longer?

- The velocity on the `outlet` of the `pitzDaily` is now non-uniform
  - In some cases slight improvements of the convergence have been seen because of this
- In our case the potential flow on the inlet didn't make much difference
  - In fact: it "destroyed" the parabolic inlet velocity
- Why not use a longer mesh on each hand:
  - Sometimes we don't want the "full physics" in these inlets/outlets

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

An old case

- This is a demo case for a proof-of-concept FSI (Fluid Structure Interaction) solver that was developed yeaaaars ago
    - (icoStructFoam if you want to Google)
- Flow through a short channel past an obstacle
    - The obstacle is some soft material ("thing") whose inside is fixed on a "stick"
- The traction of the fluid deforms the solid
- The solver used is pimpleFoam
    - With an added structure solver
        - groovyTractionDisplacement uses the fluid shear stress to calculate the traction

Thing on a stick deformation

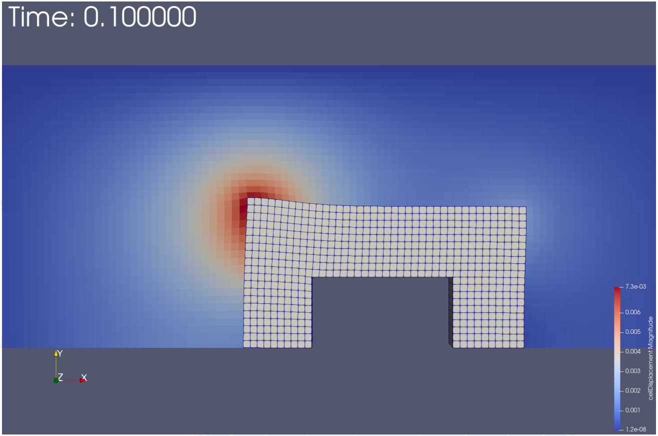

Figure: Mesh deformation

Thing on a stick velocity

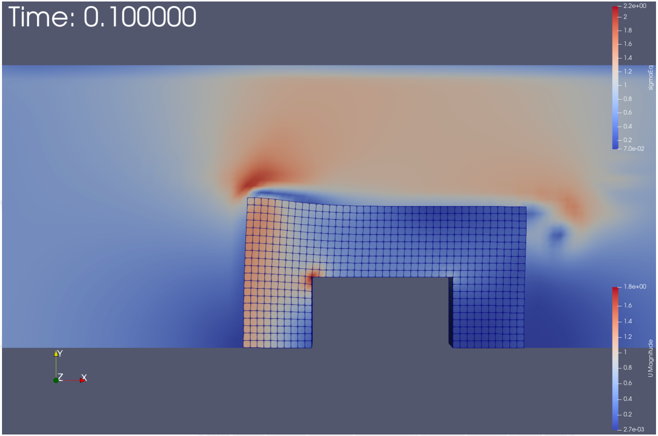

Figure: Velocity in the fluid and stress in the solid

FSI benchmark case

- This is the port of a tutorial-case from `foam-extend 4.1`
  - A benchmark-case for FSI by Hron and Turek
- A channel flow
- In the middle of the channel is a solid "pole"
- On this pole a flexible sheet is attached
  - The flow going past this sheets excites a periodic movement in the sheet
    - It goes up and down
- The base solver is `pimpleFoam`
  - a solid region was added
  - material properties were taken from the `foam-extend` tutorial

Hron and Turek result

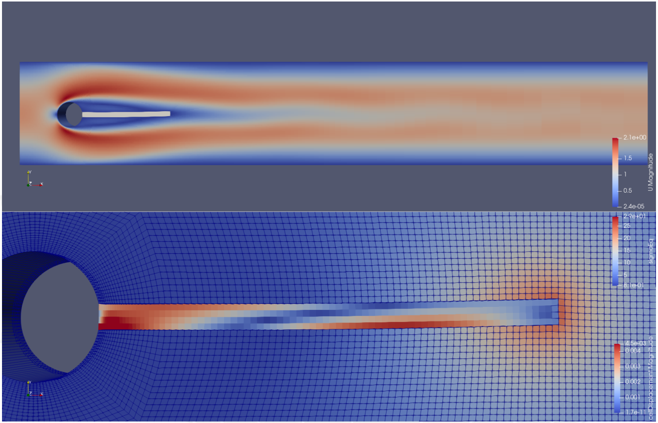

Figure: Velocity of the fluid and deformation of the fluid mesh

Solid oscillations

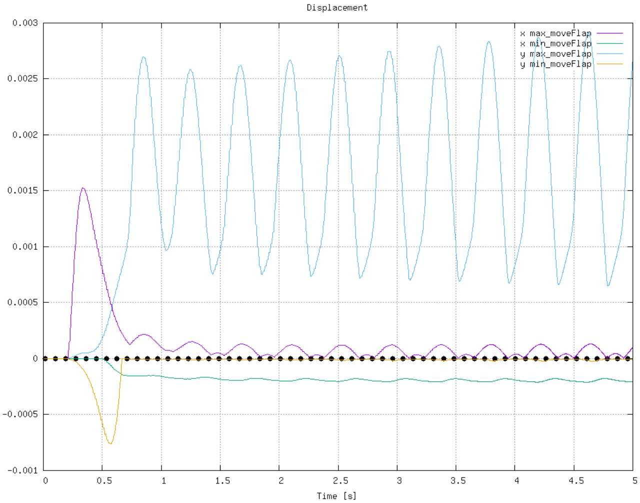

Figure: Maximum displacements of the solid

Problems with that case

- This case shows the limitations of the approach
  - it was unstable so relaxation had to be added
  - the displacements are one order of magnitude smaller than they should be
- The reasons are
  - The explicit coupling of the regions
    - Only at the end of the time-steps do they exchange information
  - The formulation from `solidDisplacementFoam` is only valid for small deformations

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

Outline

1. Introduction
   - This presentation
   - Who is this?
   - Adaptive boundary conditions
2. What is already there
   - OpenFOAM
   - swak4Foam
3. simpleRegionSolvers
   - The library
   - Set-up
4. Examples
   - Adding heat conduction
   - Flow boundary conditions
   - Structure mechanics
5. The End
   - Limitations and outlook
   - Goodbye

What these region solvers can't do

- Simulations that require a tight coupling and highly transient effects are problematic
  - Heat conduction: no problem
  - FSI: needs work
- The interpolation on the boundaries may lead to numerical errors

Supported versions

- The library was developed on v1906
  - then ported to OF 7 and `foam-extend 4.1`
    - Even to OF 2.3
  - Works on OF8 and v1912 an v2006 as well
- Currently some of the solvers do not work on `foam-extend` and `2.3` because of large differences in the infra-structure
- All the cases shown here can be found in `Examples/RegionSolver`
  - Work on v2006 out-of-the box
  - OF8 needs some adaptions

Further developments

- The 4 solvers in the library cover most conceivable applications
  - No sense to port more specialized solvers
    - Except for a structural solver for big deformations
- A tighter (more stable) coupling using the `fvOption`-mechanism is investigated
  - Maybe this allows properly simulating the Hron/Turek-case

Outline

1 Introduction
- This presentation
- Who is this?
- Adaptive boundary conditions

2 What is already there
- OpenFOAM
- swak4Foam

3 simpleRegionSolvers
- The library
- Set-up

4 Examples
- Adding heat conduction
- Flow boundary conditions
- Structure mechanics

5 The End
- Limitations and outlook
- Goodbye

No questions

Thanks for listening

Suggestions? Comments?

Offers for help?

OpenFOAM Workshop

16th OpenFOAM-Workshop

8.-11. June 2021

http://openfoamworkshop.org/

License of this presentation

This document is licensed under the *Creative Commons Attribution-ShareAlike 3.0 Unported* License (for the full text of the license see
`http://creativecommons.org/licenses/by-sa/3.0/legalcode`).
As long as the terms of the license are met any use of this document is fine (commercial use is explicitly encouraged).
Authors of this document are:

Bernhard F.W. Gschaider original author and responsible for the strange English grammar. Contact him for a copy of the sources if you want to extend/improve/use this presentation