How to implement a new boundary condition and modify a solver without modifying the solver source code

Joachim Herb, GRS

2019-07-23

OpenFOAM Workshop 2019, Duisburg

Contact data

Email: Joachim.Herb@grs.de

CFD Online: https://www.cfd-online.com/Forums/members/jherb.html

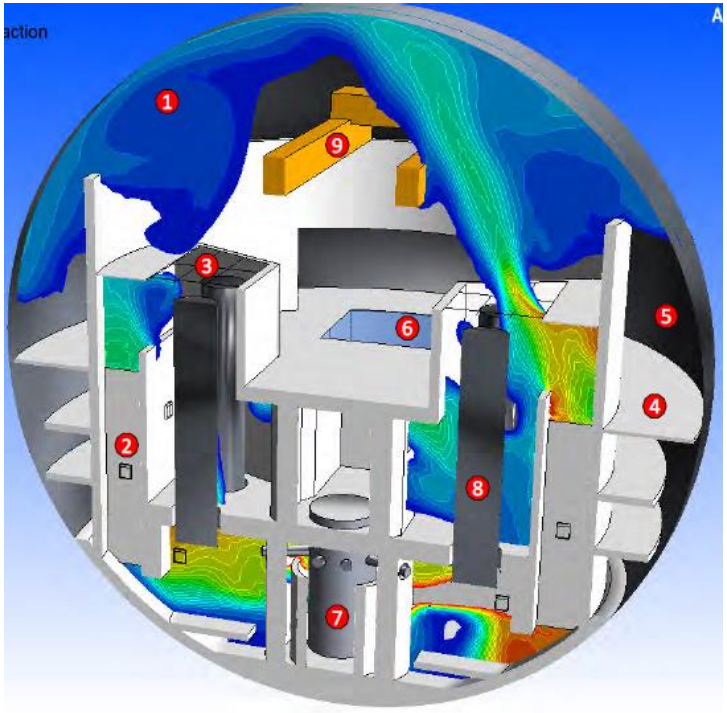
Github: https://github.com/jmozmoz/OFdevelopments

Overview

- Why?
- Existing code
- Physics
- Overview of implementation
- Code/Steps how to implement => github/commits
- Verification

Motivation

- In new reactor designs, passive safety systems play important role
- Need to simulate evaporation and condensation in the containment
- Need to resolve local phenomena'
  (e. g. local concentration of non-condensable gases)
- Long term goal: Use CFD as additional tool for safety analyses of different phenomena in the containment
- Demonstrate applicability of OpenFOAM to reactor safety/containment relevant phenomena
- Implement wall condensation, because
  - BMWi funded implementation exists for CFX with documentation
  - it is missing in OpenFOAM



Schramm et al., GRS-324,
https://doi.org/10.2314/GBV:88267787X

Wall condensation modelling in OpenFOAM

- Since OpenFOAM v3.0+ (ESI)

  thermalHumidityCoupledMixed
  https://www.openfoam.com/releases/openfoam-v3.0+/boundary-conditions.php

  https://develop.openfoam.com/Development/OpenFOAM-plus/tree/master/src/thermophysicalModels/thermophysicalPropertiesFvPatchFields/liquidProperties/humidityTemperatureCoupledMixed

  Problems:

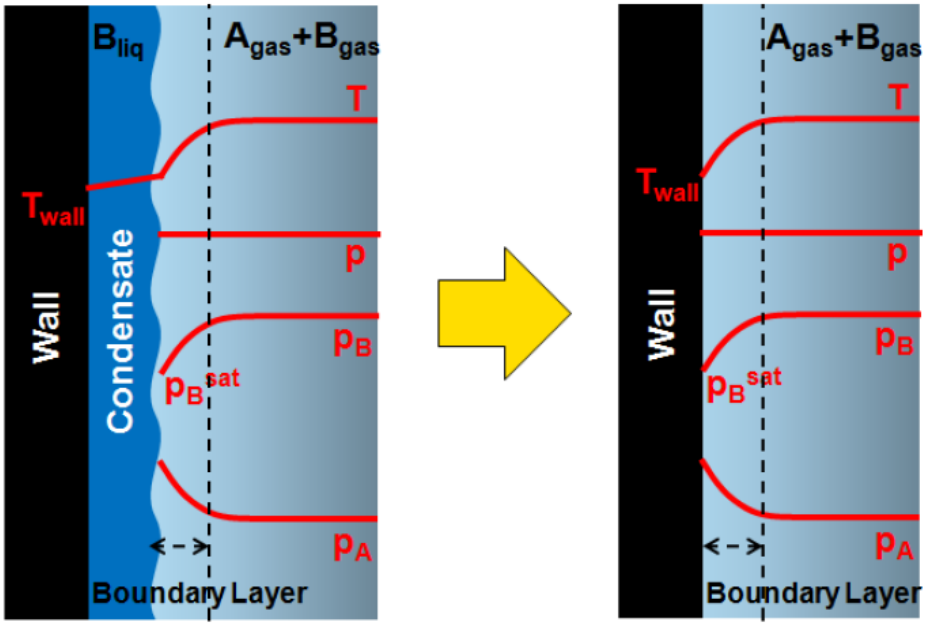
  - "Macroscopic model": Nu(Tsat, Re(L)), h_m(Sh(Re(L), Sc))
    L: Parameter specified by user
  - Possibly wrong implementation of correlation given in T. Bergam et al. Heat and Mass Transfer. 7th Edition. Chapter 10.
    Interpreted as Nusselt number [-] instead of a heat transfer correlation [W/(K m²)]
- chtMultiRegionFoam (OpenFOAM-6)

  Simulate air/vapor mixture (non-reacting) with conjugated heat transfer (https://openfoam.org/release/6/)

  Use thermophysicalProperties of OpenFOAM

Wall condensation modelling in ANSYS CFX

- Implementation of a “microscopic” model for turbulent transport



Zschaeck et al., CFD modelling and validation of wall condensation in the presence of non-condensable gases, Nuclear Engineering and Design **279**, pp 137-146 (2014)
see also https://doi.org/10.2314/GBV:861756150

Physics

Ideas

- Saturation condition at wall

⇒ Vapor concentration at wall given by saturation pressure at wall temperature

- Diffusion in cells next to wall from cell center to wall face

$$\dot{m}_{H_2O} = -\frac{\Gamma_{H_2O,c}}{y_c}\frac{Y_{H_2O,c} - Y_{H_2O,f}}{1 - Y_{H_2O,f}}$$

- Latent heat released at wall
- Mass diffusing to the fluid:
  - Remove it from the fluid
  - Remove specific heat from fluid
  - Add it to wall film

y, wall distance of cell center

$\Gamma_{H_2O,c}$, diffusivity of H_2O in air

$Y_{H_2O,c}$, mass fraction of H_2O at cell center

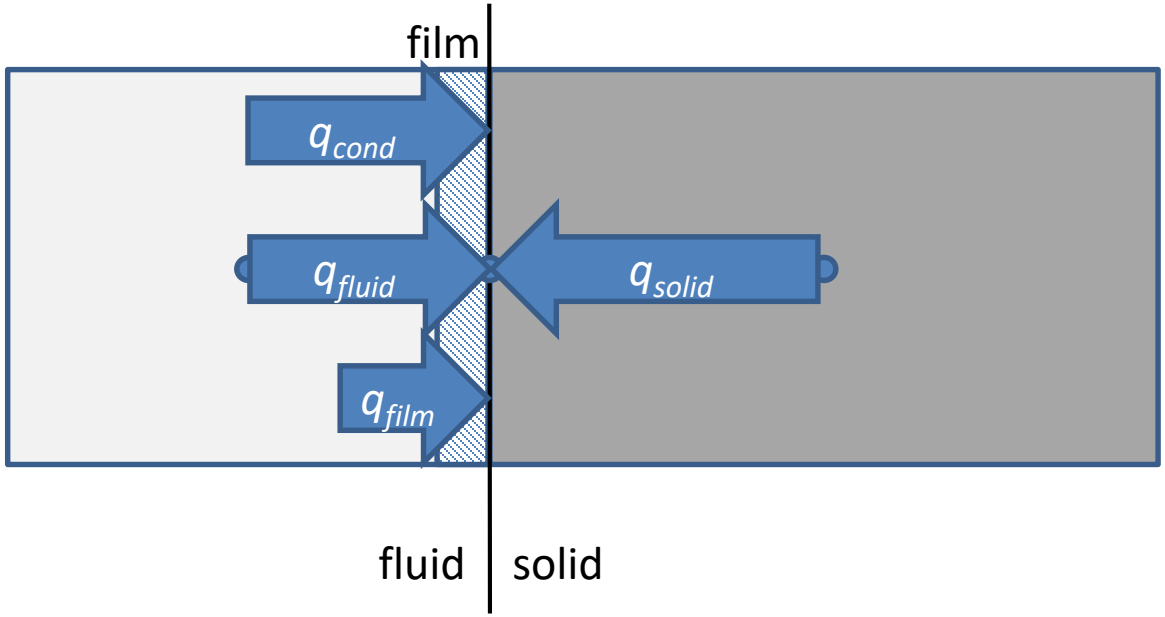
$Y_{H_2O,f}$, mass fraction of H_2O at wall face

Implementation (1/4)

Boundary condition for temperature:

- Using OpenFOAM class hierarchy (`mixedFvPatchScalarField`)
- Saturation condition at the wall
  - Vapor concentration for $T_{sat} = T_{wall}$
  - Diffusion from cell center of cell to wall due to different concentration
  - Energy conservation at wall

$$T_f = wT_r + (1 - w)(T_c + \nabla_\perp T y)$$



$?_f$: face value

$?_c$: cell center value

$?_r$: “reference” value

Implementation (2/4)

Calculate source mass and energy source terms:

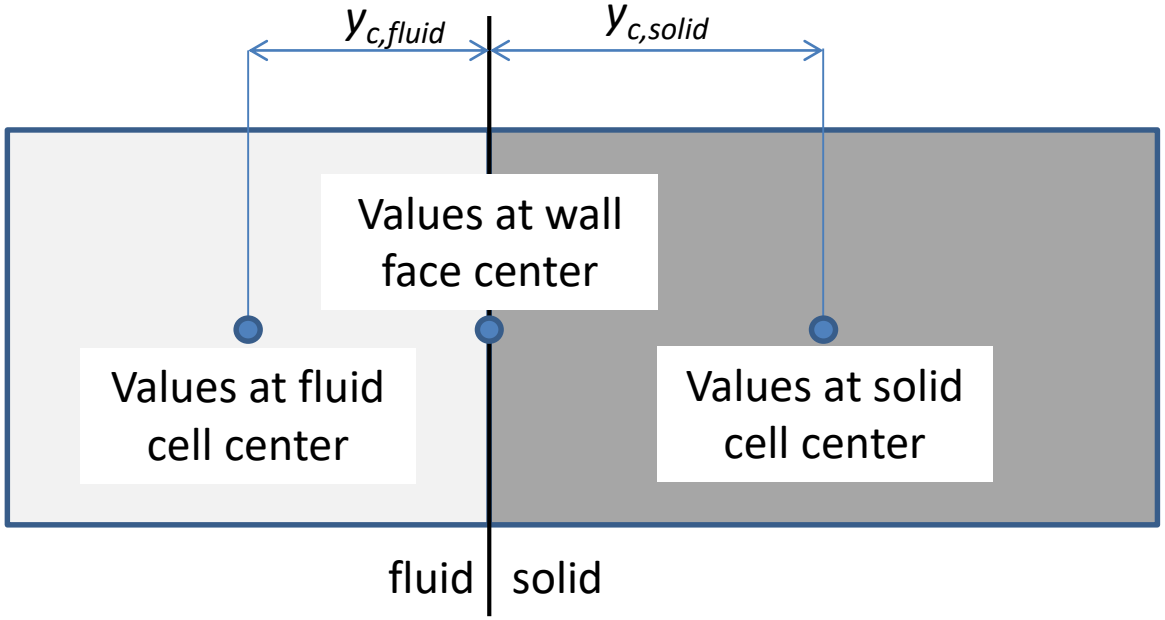
$$\dot{m}_{H_2O} = -\frac{\Gamma_{H_2O,c}}{y_c}\frac{Y_{H_2O,c} - Y_{H_2O,f}}{1 - Y_{H_2O,f}}$$

$$Y_{H_2O,f} = X_{H_2O,f}\frac{M_{H_2O}}{M}$$

$$X_{H_2O,f} = \frac{p_{H_2O,sat}(T_f)}{p_f}$$

$$\Gamma_{H_2O,c} = \frac{\mu}{Sc} + \frac{\mu_t}{Sc_t}$$

$$S_{m,H_2O} = \frac{\dot{m}_{H_2O} A_f}{V_c} \qquad S_{h,H_2O} = S_{m,H_2O} Hs_{H_2O}$$



Water film at wall is used to store mass/energy, but does not effect fluid behavior ($T_{film} = T_{wall}$)

Implementation (3/4)

Add mass source term $S_{m,i}$ to pressure, density, component equations

```
fvScalarMatrix rhoEqn
(
    fvm::ddt(rho)
  + fvc::div(phi)
  ==
    fvOptions(rho)
  + filmMassSource
);
```

In OpenFOAM-7 and OpenFOAM-dev

```
fvScalarMatrix YiEqn
(
fvm::ddt(rho, Yi)
  + mvConvection->fvmDiv(phi, Yi)
  - fvm::laplacian(gammaEff, Yi)
 ==
    reaction.R(Yi)
  + fvOptions(rho, Yi)
  + filmMassSource
);

fvScalarMatrix p_rghEqn
(
    p_rghEqnComp + p_rghEqnIncomp
  ==
    filmMassSource
  + fvOptions(psi, p_rgh,
rho.name())
);
```

Implementation (4/4)

Add energy source term $S_{h,i}$ to energy equation

```
fvScalarMatrix EEqn
(
    fvm::ddt(rho, he) + fvm::div(phi, he)
  + fvc::ddt(rho, K) + fvc::div(phi, K)
…
  - fvm::laplacian(turbulence.alphaEff(), he)
 ==
    rho*(U&g)
…
  + fvOptions(rho, he)
  + filmEnergySource
);
```

Energy conservation at the wall (1/2)

$$q_{fluid} = \Delta_{fluid} \kappa_{fluid,eff} \left(T_{fluid,c} - T_f \right)$$

$$q_{solid} = \Delta_{wall} \kappa_{wall} \left(T_{solid,c} - T_f \right)$$

$$q_{cond} = \dot{m}_{H_2O} L$$

$$q_{film} = \frac{m_{film}}{A_f \Delta t} c_p \left(T_{film}(t) - T_{film}(t - \Delta t) \right)$$

$$= \frac{m_{film}}{A_f \Delta t} c_p \left(T_f - T_f(t - \Delta t) \right)$$

$$q_{fluid} + q_{cond} + q_{solid} - q_{film} = 0$$

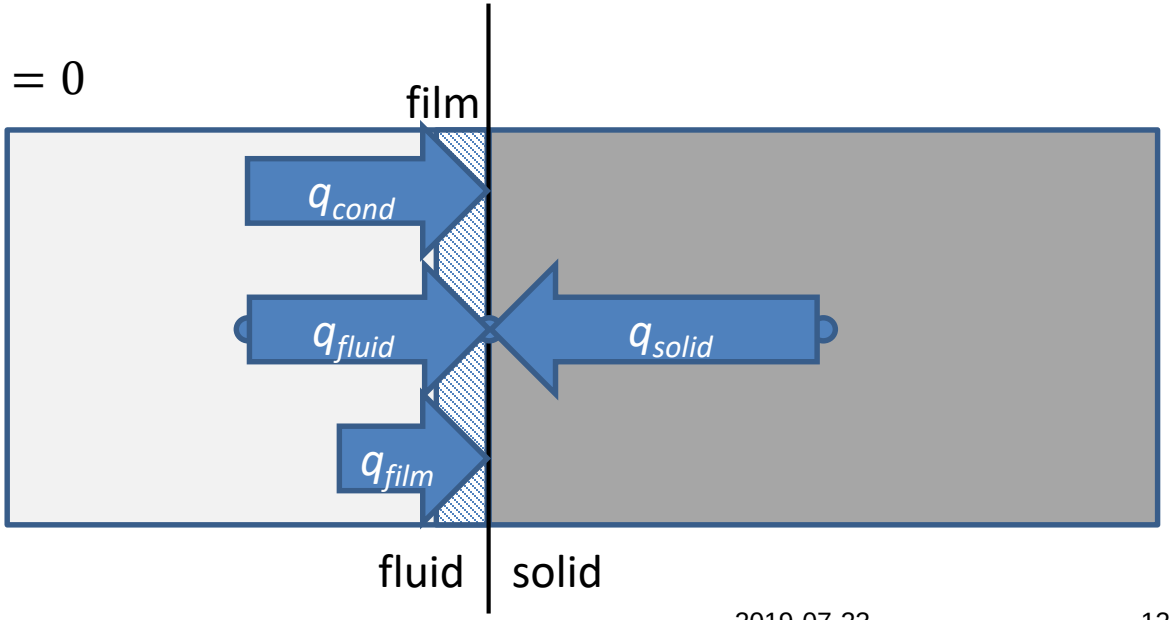
$\Delta = \frac{1}{y}$, inverse wall distance of cell center

κ, thermal conductivity

L, latent heat

A_f, face area

Δt, time step size



Energy conservation at the wall (2/2)

Sort for T_f, T_c, T_r, $\nabla_\perp T$

$$T_f = wT_r + (1 - w)(T_c + \nabla_\perp T y)$$

$$w = \frac{\Delta_{wall}\kappa_{wall} + \frac{m_{film}}{A_w \Delta t} c_p}{\Delta_{fluid}\kappa_{fluid,eff} + \Delta_{wall}\kappa_{wall} + \frac{m_{film}}{A_w \Delta t} c_p}$$

$$T_r = \frac{\Delta_{wall}\kappa_{wall} T_{solid,c} + \frac{m_{film}}{A_w \Delta t} c_p T_p(t - \Delta t) + \dot{m}_i L}{\Delta_{wall}\kappa_{wall} + \frac{m_{film}}{A_w \Delta t} c_p}$$

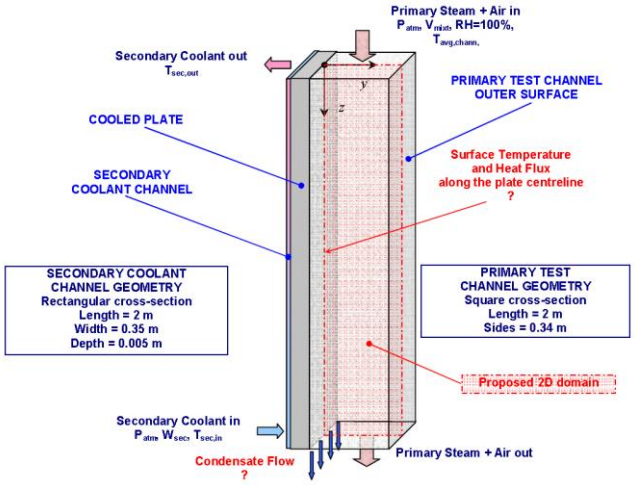
$$\nabla_\perp T = 0$$

Verification

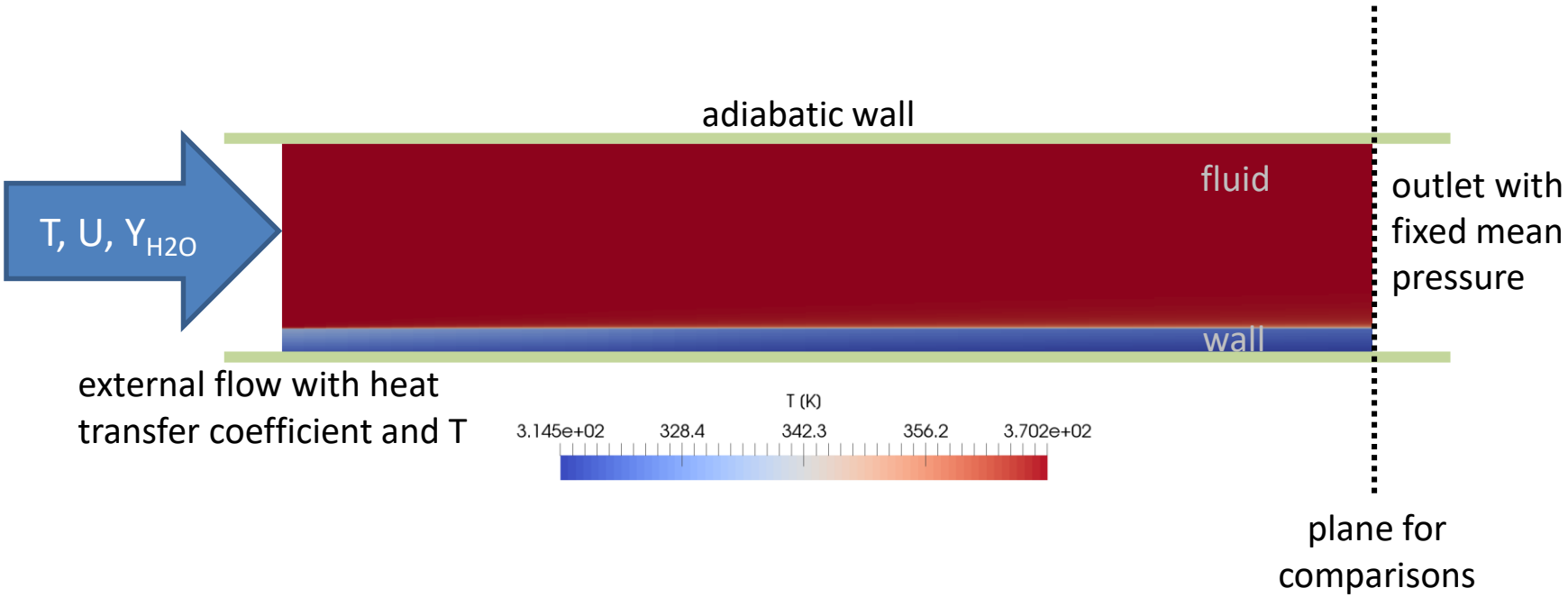
- OpenFOAM function objects (and swak4foam):
  - patch/wall fluxes
  - integrated values
  - Probes
- Jupyter notebook to visulize results

Test Case: Conan Sarnet-Benchmark

- Channel flow with cooled wall
- 2D simulation with modified chtMultiRegionFoam solver (OF-6)



Ambrosini et al., SARnet-2 CONDENSATION BENCHMARK No. 2



Walk through code

https://github.com/jmozmoz/OFdevelopments/tree/master/wallCondensationCoupled

- create a boundary condition derived for the OpenFOAM class `mixedFvPatchScalarField` (based on `turbulentTemperatureRadCoupledMixedFvPatchScalarField`)
- create `fvOptions` derived source terms
- access all necessary fluid and wall fields and properties at the surface
- functions objects in test case
- Python and the libraries pandas, numpy, and matplotlib for postprocessing

How to access fields in boundary conditions (1/2)

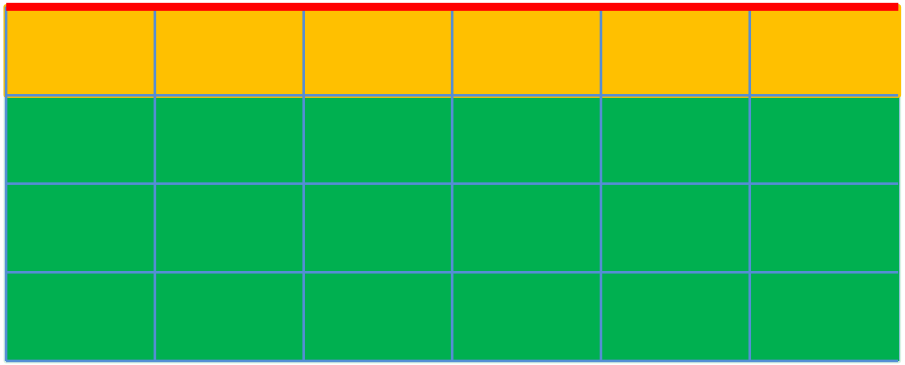
E.g. temperture boundary condition:

- Fields on the patch
  - T: `*this`
  - other scalar fields:

```
const fvPatchScalarField&
patchField =
patch().lookupPatchField<volScalarField, scalar>(fieldName);
```

- fields in cells next to patch

```
const fvPatchScalarField&
patchInternalField =
patchField.patchInternalField()
```



- fields in full mesh
  (including cells next to patch)

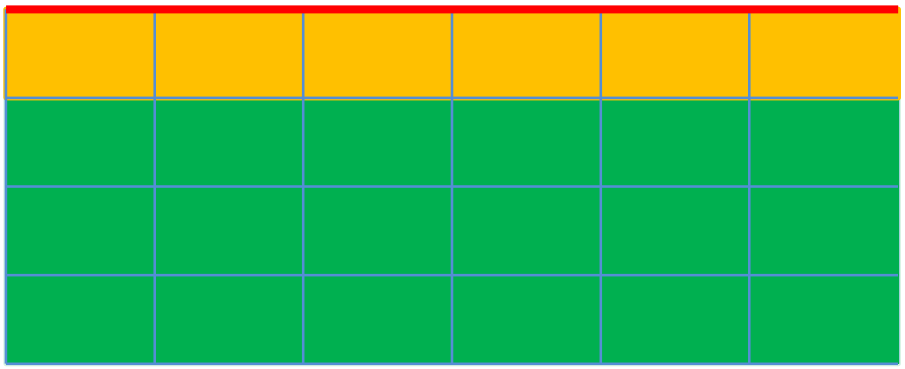
```
const volScalarField&
meshField =
static_cast<const volScalarField&>(patchField.internalField());
```

How to access fields in boundary conditions (2/2)

Finding patchInternalField in full mesh:

```
const labelList& faceCells = patch().faceCells();

forAll(faceCells, faceI)
{
    const label cellI = faceCells[faceI];
    meshField[cellI] == patchInternalField[faceI]; // true!
}
```



Code: Heat release at wall

```
void wallCondensationCoupledMixedFvPatchScalarField::updateCoeffs()
{
...
                const scalar YsatFace = pSatFace/pFace*Mv/Mcomp_;
                const scalar gamma = muCell + mutFace / Sct;

                // mass flux [kg/s/m^2]
                // positive if mass is condensing
                dm[faceI] =
                    gamma*deltaFace
                   *(Ycell - YsatFace)/(1 - YsatFace);

...
 // Heat flux due to change of phase [W/m2]
        dmHfg_ = dm*hPhaseChange;
        dHspec = dm*hRemovedMass;

        forAll(faceCells, faceI)
        {
            const label cellI = faceCells[faceI];

            filmMassSource[cellI] =
               -dm[faceI]
               *magSf[faceI]
               /mesh.cellVolumes()[cellI];

            filmEnergySource[cellI] =
               -dm[faceI]*hRemovedMass[faceI]
               *magSf[faceI]
               /mesh.cellVolumes()[cellI];
        }
...
```

Code: Source terms using fvOptions

```
// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * * * //
Foam::fv::wallCondensationSource::wallCondensationSource(
...
{
    const basicThermo& thermo =
        mesh_.lookupObject<basicThermo>(basicThermo::dictName);
    fieldNames_.setSize(3);
    fieldNames_[0] = thermo.he().name();
    fieldNames_[1] = specieName_;
    fieldNames_[2] = "rho";
//    fieldNames_[3] = "p_rgh"; // we do not need to activate the source
//                              // for p_rgh, because it will be activated
//                              // automatically, if rho is active
    applied_.setSize(fieldNames_.size(), false);
}

void Foam::fv::wallCondensationSource::addSup(fvMatrix<scalar>& eqn, const label fieldi)
{
    eqn += filmMassSourceFluid_;
}

void Foam::fv::wallCondensationSource::addSup(const volScalarField& rho, fvMatrix<scalar>& eqn, const label fieldi)
{
    if (fieldi == 0)
    {
        eqn += filmEnergySourceFluid_;
    }
    else
    {
        eqn += filmMassSourceFluid_;
    }
}
```

Acknowlegments

This work was supported by the German Federal Ministry of Economic Affairs and Energy based on a decision of the German Bundestag within the project RS 1562.

Major parts of the implementation were done at the NUMAP-FOAM School 2018

https://foam-extend.fsb.hr/numap/numap-foam-summer-school-2018/

Supported by:

Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag

GRS is hiring…

https://www.grs.de/karriere/stellenanzeige-2019-09

…

For the Cooling Circuit department located in **Garching**, we are looking for a **Research Assistant m/f/d**

Your tasks

- Further development and validation of CFD simulation codes (particularly OpenFOAM) for thermal hydraulics of nuclear reactor facilities
- Improvement of models for one- and two-phase heat transfer, two-phase flows and fluid-structure-interaction
- Enhancement of single-phase as well as two-phase coupling between CFD codes to CSM codes and thermal hydraulics system codes like ATHLET
- Involvement in development and validation of thermal hydraulics system codes (ATHLET)
- Participation in national and international test and research projects on nuclear safety simulation codes
- Safety analyses for LWR and metal-cooled reactor designs
- Presentation of work results at national and international conferences

…