



Science and
Technology
Facilities Council

Scientific Computing



CCP-WSI
a Collaborative Computational Project
in Wave Structure Interaction

Introduction to programming solvers in OpenFOAM®

Dr. Wendi Liu

The screenshot shows a dark-themed website for a workshop. At the top, it says "OpenFOAM Parallel Performance Engineering Workshop". Below that are two blue buttons: "Register" and "Agenda". Underneath are four small icons with text: a calendar icon for "5 - 6 June 2023", a clock icon for "Time TBC", and a location pin icon for "Daresbury Laboratory, Keckwick Lane, WA4 4AD". At the bottom right of the page is a large image of a server rack with the text "OpenFOAM" below it.

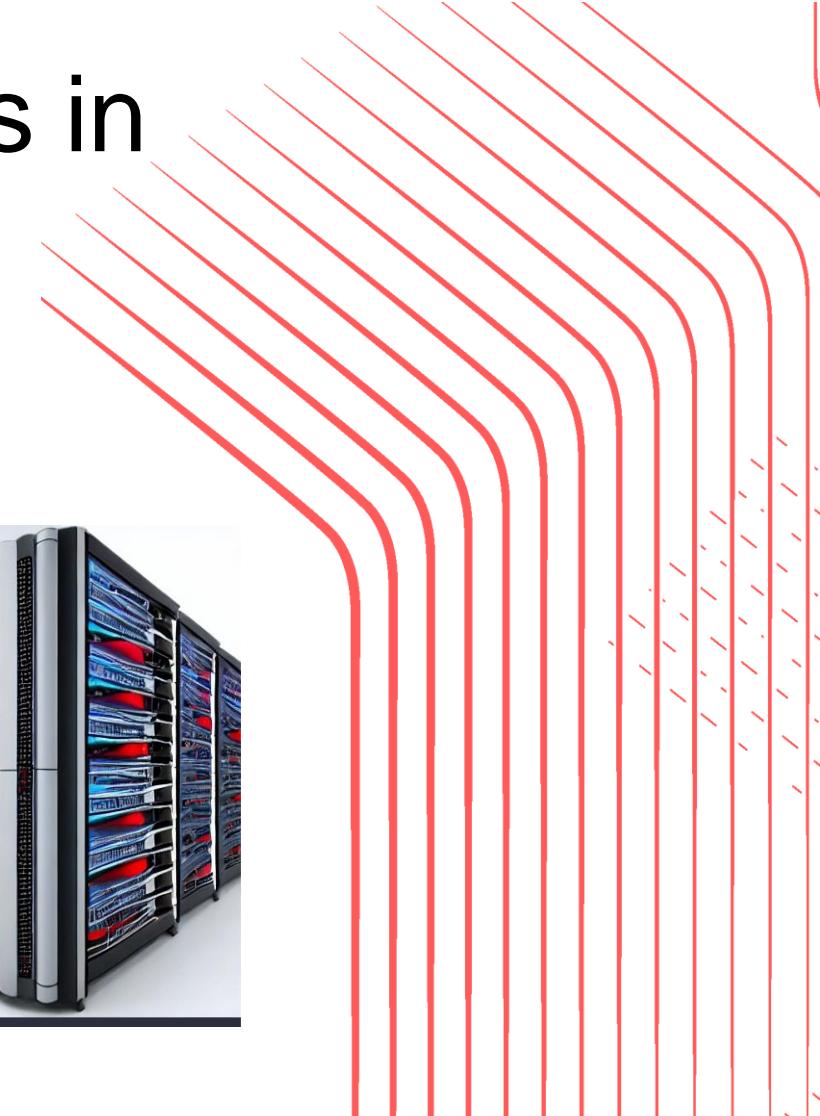


Table of Content

1 Building Blocks in OpenFOAM®

2 Solver Structure

3 Hands On Session



Science and
Technology
Facilities Council

Scientific Computing



Science and
Technology
Facilities Council

Scientific Computing

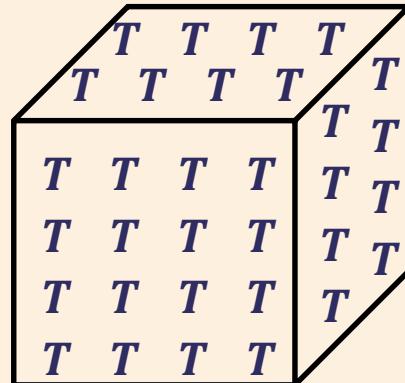
Building Blocks in OpenFOAM®



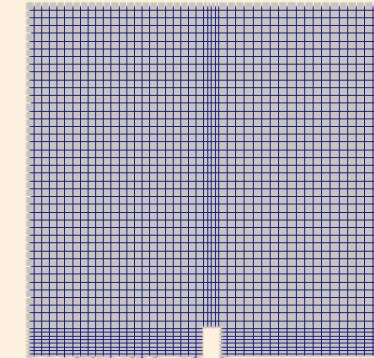
Building Blocks in OpenFOAM®

$$\mathbf{T} = T_{ij} = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1n} \\ T_{21} & T_{22} & \cdots & T_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{m1} & T_{m2} & \cdots & T_{mn} \end{pmatrix}$$

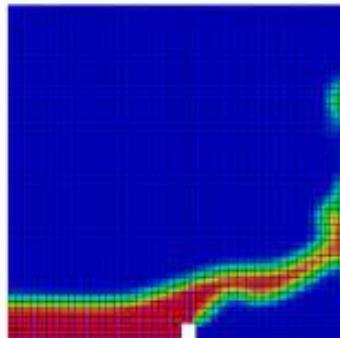
Tensor



Field



Discretisation



Solver

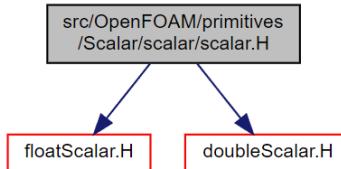
Building Blocks in OpenFOAM®

Tensor classes

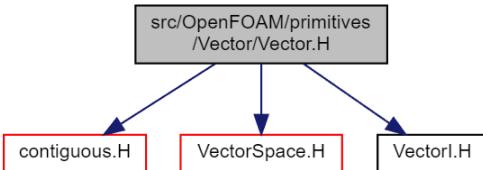
s

$$\boldsymbol{v} = v_i = (v_1 \quad v_2 \quad v_3)$$

Rank 0

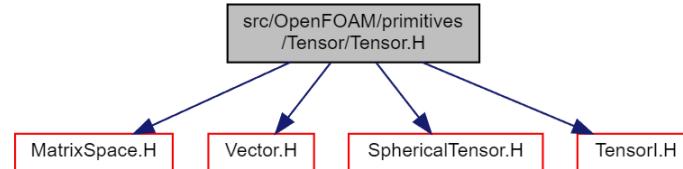


Rank 1



Rank 2

$$\boldsymbol{T} = T_{ij} = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1n} \\ T_{21} & T_{22} & \cdots & T_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{m1} & T_{m2} & \cdots & T_{mn} \end{pmatrix}$$



Access function:
 $x()$, $y()$, $z()$

Scalar

Vector

Access function:
 $xx()$, $xy()$, $xz()$...

Tensor

Building Blocks in OpenFOAM®

Algebraic tensor operations – snapshots from Programmer's Guide v2212

Operation	Comment	Mathematical Description	Description in OpenFOAM
Addition		$a + b$	$a + b$
Subtraction		$a - b$	$a - b$
Scalar multiplication		$s a$	$s * a$
Scalar division		a/s	a / s
Outer product	rank $a, b \geq 1$	ab	$a * b$
Inner product	rank $a, b \geq 1$	$a \cdot b$	$a \& b$

Continued on next page

OpenFOAM-v2212

Continued from previous page	Operation	Comment	Mathematical Description	Description in OpenFOAM
Double inner product	rank $a, b \geq 2$	$a : b$	$a \&& b$	
Cross product	rank $a, b = 1$	$a \times b$	$a ^ b$	
Square		a^2	$\text{sqr}(a)$	
Magnitude squared		$ a ^2$	$\text{magSqr}(a)$	
Magnitude		$ a $	$\text{mag}(a)$	
Power	$n = 0, 1, \dots, 4$	a^n	$\text{pow}(a, n)$	
Component average	$i = 1, \dots, N$	\bar{a}_i	$\text{cmptAv}(a)$	
Component maximum	$i = 1, \dots, N$	$\max(a_i)$	$\text{cmptMax}(a)$	
Component minimum	$i = 1, \dots, N$	$\min(a_i)$	$\text{cmptMin}(a)$	
Scale		$\text{scale}(a, b)$	$\text{cmptMultiply}(a, b)$	
Geometric transformation		transforms a using tensor T	$\text{transform}(T, a)$	

Operations exclusive to tensors of rank 2

Transpose	T^T	$\text{T.T}()$
Diagonal	$\text{diag } T$	$\text{diag}(T)$
Trace	$\text{tr } T$	$\text{tr}(T)$
Deviatoric component	$\text{dev } T$	$\text{dev}(T)$
Symmetric component	$\text{symm } T$	$\text{symm}(T)$
Skew-symmetric component	$\text{skew } T$	$\text{skew}(T)$
Determinant	$\det T$	$\det(T)$
Cofactors	$\text{cof } T$	$\text{cof}(T)$
Inverse	$\text{inv } T$	$\text{inv}(T)$
Hodge dual	$* T$	$*T$

Operations exclusive to scalars			
Sign (boolean)		$\text{sgn}(s)$	$\text{sign}(s)$
Positive (boolean)		$s \geq 0$	$\text{pos}(s)$
Negative (boolean)		$s < 0$	$\text{neg}(s)$
Limit	n scalar	$\text{limit}(s, n)$	$\text{limit}(s, n)$
Square root		\sqrt{s}	$\text{sqrt}(s)$
Exponential		$\exp s$	$\text{exp}(s)$
Natural logarithm		$\ln s$	$\text{log}(s)$
Base 10 logarithm		$\log_{10} s$	$\text{log10}(s)$
Sine		$\sin s$	$\text{sin}(s)$
Cosine		$\cos s$	$\text{cos}(s)$
Tangent		$\tan s$	$\text{tan}(s)$
Arc sine		$\text{asin } s$	$\text{asin}(s)$
Arc cosine		$\text{acos } s$	$\text{acos}(s)$
Arc tangent		$\text{atan } s$	$\text{atan}(s)$
Hyperbolic sine		$\sinh s$	$\text{sinh}(s)$
Hyperbolic cosine		$\cosh s$	$\text{cosh}(s)$
Hyperbolic tangent		$\tanh s$	$\text{tanh}(s)$
Hyperbolic arc sine		$\text{asinh } s$	$\text{asinh}(s)$
Hyperbolic arc cosine		$\text{acosh } s$	$\text{acosh}(s)$
Hyperbolic arc tangent		$\text{atanh } s$	$\text{atanh}(s)$
Error function		$\text{erf } s$	$\text{erf}(s)$
Complement error function		$\text{erfc } s$	$\text{erfc}(s)$

Continued on next page

OpenFOAM-v2212

Continued from previous page	Operation	Comment	Mathematical Description	Description in OpenFOAM
Logarithm gamma function			$\ln \Gamma s$	$\text{lgamma}(s)$
Type 1 Bessel function of order 0			$J_0 s$	$\text{j0}(s)$
Type 1 Bessel function of order 1			$J_1 s$	$\text{j1}(s)$
Type 2 Bessel function of order 0			$Y_0 s$	$\text{y0}(s)$
Type 2 Bessel function of order 1			$Y_1 s$	$\text{y1}(s)$

a, b are tensors of arbitrary rank unless otherwise stated
 s is a scalar, N is the number of tensor components

Table 2.2: Algebraic tensor operations in OpenFOAM

Building Blocks in OpenFOAM®

Dimensional units

- Units are defined using the dimensionSet class

```
dimensionSet pressureDims(1, -1, -2, 0, 0, 0, 0);
```

No.	Property	Unit	Symbol
1	Mass	kilogram	k
2	Length	metre	m
3	Time	second	s
4	Temperature	Kelvin	K
5	Quantity	moles	mol
6	Current	ampere	A
7	Luminous intensity	candela	cd

S.I. base units of measurement

Building Blocks in OpenFOAM®

Field class

- Lists of the tensor classes are defined by `Field<Type>`
 - All instances of `Field<Type>` are renamed using `typedef` as `tensorField`, `vectorField`, etc.

Field< Type > Class Template Reference

Generic templated field type. More...

Inherits FieldBase, and List< Type >.

Inherited by CompactIOField< Foam::Field< label >, label >, CompactIOField< Foam::Field< scalar >, scalar >, CompactIOField< Foam::Field< vector >, Foam::Vector >, CompactIOField< Foam::Field< Foam::SphericalTensor >, Foam::SphericalTensor >, CompactIOField< Foam::Field< Foam::SymmTensor >, Foam::SymmTensor >, CompactIOField< Foam::Field< Tensor >, Foam::Tensor >, DimensionedField< symmTensor, Foam::volMesh >, DimensionedField< symmTensor, volMesh >, DimensionedField< scalar, Foam::volMesh >, DimensionedField< scalar, volMesh >, DimensionedField< Type, pointMesh >, DimensionedField< vector, Foam::pointMesh >, DimensionedField< vector, Foam::volMesh >, DimensionedField< vector, volMesh >, DimensionedField< tensor, Foam::volMesh >, DimensionedField< tensor, volMesh >, DimensionedField< Foam::Vector, Foam::volMesh >, DimensionedField< Foam::Vector, Foam::pointMesh >, DimensionedField< scalar, Foam::surfaceMesh >, DimensionedField< scalar, surfaceMesh >, DimensionedField< scalar, Foam::pointMesh >, DimensionedField< scalar, pointMesh >, DimensionedField< vector, pointMesh >, DimensionedField< scalar, Foam::areaMesh >, DimensionedField< scalar, areaMesh >, DimensionedField< scalar, Foam::edgeMesh >, DimensionedField< Foam::Vector, Foam::areaMesh >, DimensionedField< Foam::SphericalTensor, Foam::areaMesh >, DimensionedField< Foam::SphericalTensor, Foam::edgeMesh >, DimensionedField< Foam::SymmTensor, Foam::areaMesh >, DimensionedField< Foam::SymmTensor, Foam::edgeMesh >, DimensionedField< Foam::Tensor, Foam::areaMesh >, DimensionedField< Foam::Tensor, Foam::edgeMesh >, DimensionedField< Type, Foam::areaMesh >, DimensionedField< Type, Foam::edgeMesh >, DimensionedField< Foam::Vector, Foam::surfaceMesh >, DimensionedField< Foam::SphericalTensor, Foam::volMesh >, DimensionedField< Foam::SymmTensor, Foam::volMesh >, DimensionedField< Foam::Tensor, Foam::volMesh >, DimensionedField< Foam::Tensor, Foam::surfaceMesh >, DimensionedField< Type, Foam::volMesh >, DimensionedField< Type, Foam::pointMesh >, DimensionedField< vector, Foam::surfaceMesh >, DimensionedField< scalar, Foam::pointGeoMesh >, DimensionedField< Foam::SphericalTensor, Foam::pointMesh >, DimensionedField< Foam::SymmTensor, Foam::pointMesh >, DimensionedField< Foam::Tensor, Foam::pointMesh >, DimensionedField< tensor, Foam::areaMesh >, DimensionedField< tensor, areaMesh >, DimensionedField< vector, Foam::areaMesh >, DimensionedField< vector, areaMesh >, DimensionedField< scalar, Foam::edgeMesh >, DynamicField< Foam::Vector >, DynamicField< label >, GlobalField< vector >, IOField< label >, IOField< scalar >, IOField< Foam::Vector >, IOField< Foam::SphericalTensor >, IOField< Foam::SymmTensor >, IJKField< scalar >, IJKField< Foam::Vector >, IJKField< vector >, IJKField< bool >, IJKField< Foam::SymmTensor >, faPatchField< scalar >, faPatchField< Foam::Vector >, faPatchField< Foam::SphericalTensor >, faPatchField< Foam::SymmTensor >, faPatchField< Foam::Tensor >, faPatchField< tensor >, faPatchField< vector >, faePatchField< scalar >, faePatchField< Foam::Vector >, faePatchField< Foam::SphericalTensor >, faePatchField< Foam::SymmTensor >, faePatchField< Foam::Tensor >, fvPatchField< symmTensor >, fvPatchField< scalar >, fvPatchField< vector >, fvPatchField< tensor >, fvPatchField< Foam::Vector >, fvPatchField< Foam::SphericalTensor >, fvPatchField< Foam::SymmTensor >, fvPatchField< Foam::Tensor >, fvsPatchField< scalar >, fvsPatchField< Foam::Vector >, fvsPatchField< Foam::SphericalTensor >, fvsPatchField< Foam::SymmTensor >, fvsPatchField< Foam::Tensor >, fvsPatchField< vector >, valuePointPatchField< vector >, DimensionedField< Type, GeoMesh >, GlobalField< Type >, IOField< Type >, IJKField< Type >, faPatchField< Type >, faePatchField< Type >, fvPatchField< Type >, fvsPatchField< Type >, rawIOField< Type >, and valuePointPatchField< Type >.

Collaboration diagram for Field< Type >:

```
graph TD; FieldBase --> Field["Field< Type >"]; ListType["List< Type >"] --> Field;
```

The diagram shows a UML class hierarchy. At the top level, there are two red-bordered boxes: 'FieldBase' on the left and 'List< Type >' on the right. Below them, a larger grey-bordered box contains the text 'Field< Type >'. Two blue arrows point from the 'FieldBase' and 'List< Type >' boxes down to the 'Field< Type >' box, indicating that 'Field< Type >' inherits from both 'FieldBase' and 'List< Type >'.

[legend]



Building Blocks in OpenFOAM®

Overview of discretisation

- Spatial discretisation**

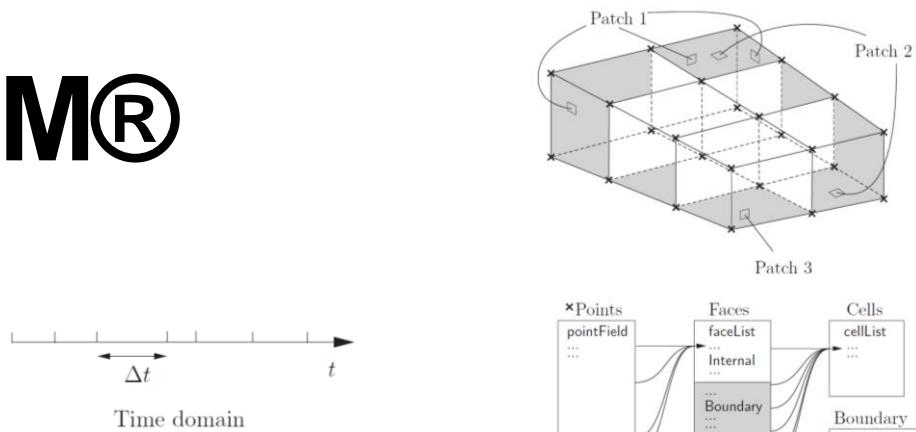
Defining the solution domain by a set of points that fill and bound a region of space when connected;

- Temporal discretisation**

(For transient problems) dividing the time domain into a finite number of time intervals, or steps;

- Equation discretisation**

Generating a system of algebraic equations in terms of discrete quantities defined at specific locations in the domain, from the PDEs that characterise the problem.



Temporal discretisation

Term description	Implicit / Explicit	Text expression	fvm::fvc:: functions
Laplacian	Imp/Exp	$\nabla^2\phi$ $\nabla \cdot \Gamma \nabla \phi$	laplacian(phi) laplacian(Gamma, phi)
Time derivative	Imp/Exp	$\frac{\partial \phi}{\partial t}$ $\frac{\partial \rho \phi}{\partial t}$	ddt(phi) ddt(rho, phi)
Second time derivative	Imp/Exp	$\frac{\partial}{\partial t} \left(\rho \frac{\partial \phi}{\partial t} \right)$	d2dt2(rho, phi)
Convection	Imp/Exp	$\nabla \cdot (\psi)$ $\nabla \cdot (\psi \phi)$	div(psi, scheme)* div(psi, phi, word)* div(psi, phi)
Divergence	Exp	$\nabla \cdot \chi$	div(chi)
Gradient	Exp	$\nabla \chi$ $\nabla \phi$	grad(chi) gGrad(phi) lsGrad(phi) snGrad(phi) snGradCorrection(phi)
Grad-grad squared	Exp	$ \nabla \nabla \phi ^2$	sqrGradGrad(phi)
Curl	Exp	$\nabla \times \phi$	curl(phi)
Source	Imp Imp/Exp†	$\rho \phi$	Sp(rho, phi) SuSp(rho, phi)

†if `fvm::SuSp` source is discretised implicit or explicit depending on the sign of `rho`.

†An explicit source can be introduced simply as a `vol<Type>Field`, e.g. `rho*phi`.

Function arguments can be of the following classes:

`phi`: `vol<Type>Field`

`Gamma`: scalar `volScalarField`, `surfaceScalarField`, `volTensorField`, `surfaceTensorField`.

`rho`: scalar, `volScalarField`

`psi`: `surfaceScalarField`.

`chi`: `surface<Type>Field`, `vol<Type>Field`.

Equation discretisation



Science and
Technology
Facilities Council

Scientific Computing

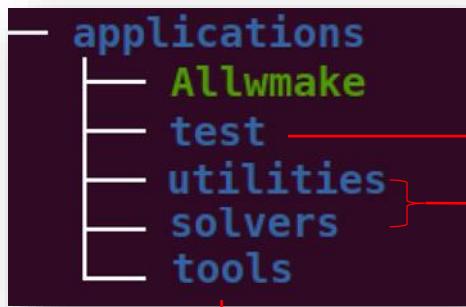
Solver Structure



Solver Structure

Folder structure of applications/

- Contains executable applications



Tools that may be useful for developers or advanced users

Unit tests for classes & functions.
Useful resource to show the usage of classes & algorithms.

Categorised in groups according to physical & mathematical models

Categorised in groups according to type of classes

The screenshot shows the 'applications' section of the OpenFOAM API Guide. The left sidebar lists categories like Namespaces, Classes, and Files. The main content area shows a dependency graph for the 'applications' directory:

```
graph TD; utilities --> src[ ]; solvers --> src[ ]; applications --> src[ ];
```

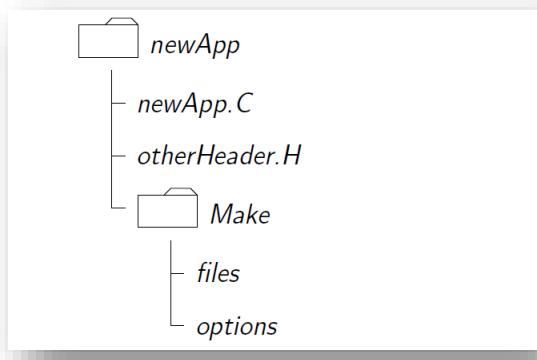
Annotations on the right side highlight specific parts of the graph:

- A red dashed box encloses the 'utilities', 'solvers', and 'applications' nodes, with the text "make use of various libs" below it.
- Red arrows point from the text "Categorised in groups according to physical & mathematical models" to the 'applications' node in the graph.
- Red arrows point from the text "Categorised in groups according to type of classes" to the 'src' node in the graph.

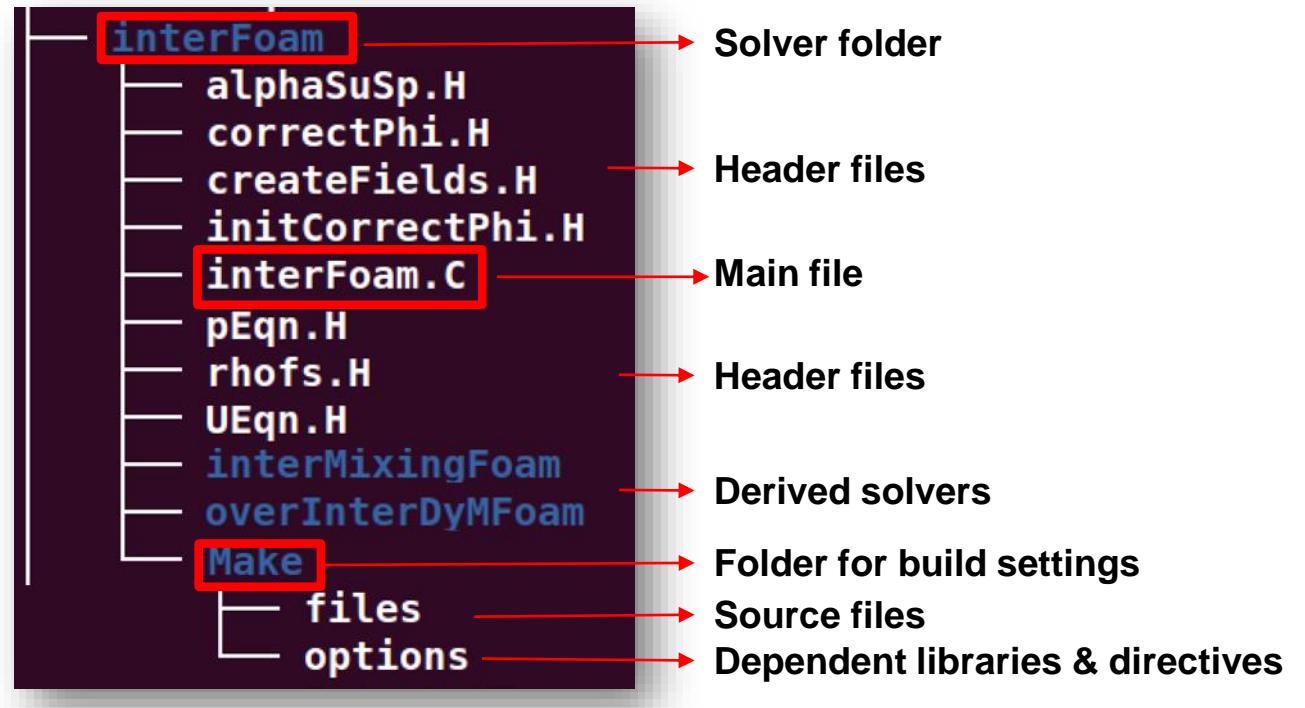
At the bottom of the page, there is a note: "OPENFOAM® is a registered trademark of OpenCFD Ltd. Generated by doxygen 1.8.17".

Solver Structure

Folder structure of applications/solvers/multiphase/interFoam



Directory structure for an application



Solver Structure

High-level overview of interFoam

Hydrostatic pressure effects

For cases that the hydrostatic pressure contribution

$$\rho(\mathbf{g} \cdot \mathbf{h})$$

is important, e.g. for buoyant and multiphase cases, it is numerically convenient to solve for an alternative pressure defined by

$$p' = p - \rho(\mathbf{g} \cdot \mathbf{h}).$$

In OpenFOAM solver applications the p' pressure term is named `p_rgh`. The momentum equation

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\mu_{\text{eff}} \nabla \mathbf{u}) = -\nabla p + \rho\mathbf{g}$$

is transformed to use p' :

$$p' = p - \rho(\mathbf{g} \cdot \mathbf{h}).$$

After the following substitutions:

$$\begin{aligned} -p &= -p' - \rho(\mathbf{g} \cdot \mathbf{h}) \\ -\nabla p &= -\nabla(p') - \nabla(\rho(\mathbf{g} \cdot \mathbf{h})) \\ &= -\nabla(p') - \rho\mathbf{g} \cdot \nabla\mathbf{h} - \mathbf{h} \cdot \nabla(\rho\mathbf{g}) \\ &= -\nabla(p') - \rho\mathbf{g} \cdot \mathbf{I} - \mathbf{g} \cdot \mathbf{h} \nabla(\rho) - \cancel{\rho\mathbf{h} \cdot \nabla(\mathbf{g})}^0 \\ &= -\nabla(p') - \rho\mathbf{g} - \mathbf{g} \cdot \mathbf{h} \nabla\rho \end{aligned}$$

where, for CFD meshes the term $\nabla\mathbf{h}$ is given by the gradient of the cell centres, which equates to the tensor \mathbf{I} , the momentum equation becomes:

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\mu_{\text{eff}} \nabla \mathbf{u}) = -\nabla p' - \mathbf{g} \cdot \mathbf{h} \nabla\rho$$

For constant density applications this can be further simplified to

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\mu_{\text{eff}} \nabla \mathbf{u}) = -\nabla p'$$

interFoam

Overview

- Category: Multiphase
- transient
- incompressible
- multiphase: 2 immiscible phases
- isothermal
- Turbulence
- Finite volume options

Equations

- PIMPLE algorithm
- Hydrostatic pressure effects

Input requirements

Mandatory fields:

- p : pressure [Pa]
- p_{rgh} : pressure - hydrostatic contribution [Pa]
- U : velocity [m/s]

Physical models

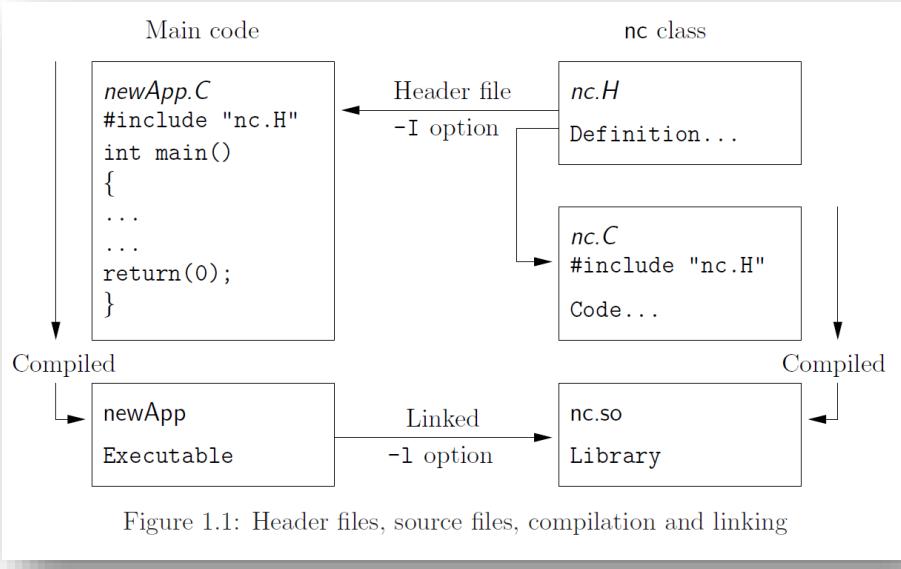
- turbulence: constant/turbulenceProperties
- finite volume options : constant/fvOptions (optional)

Solution controls

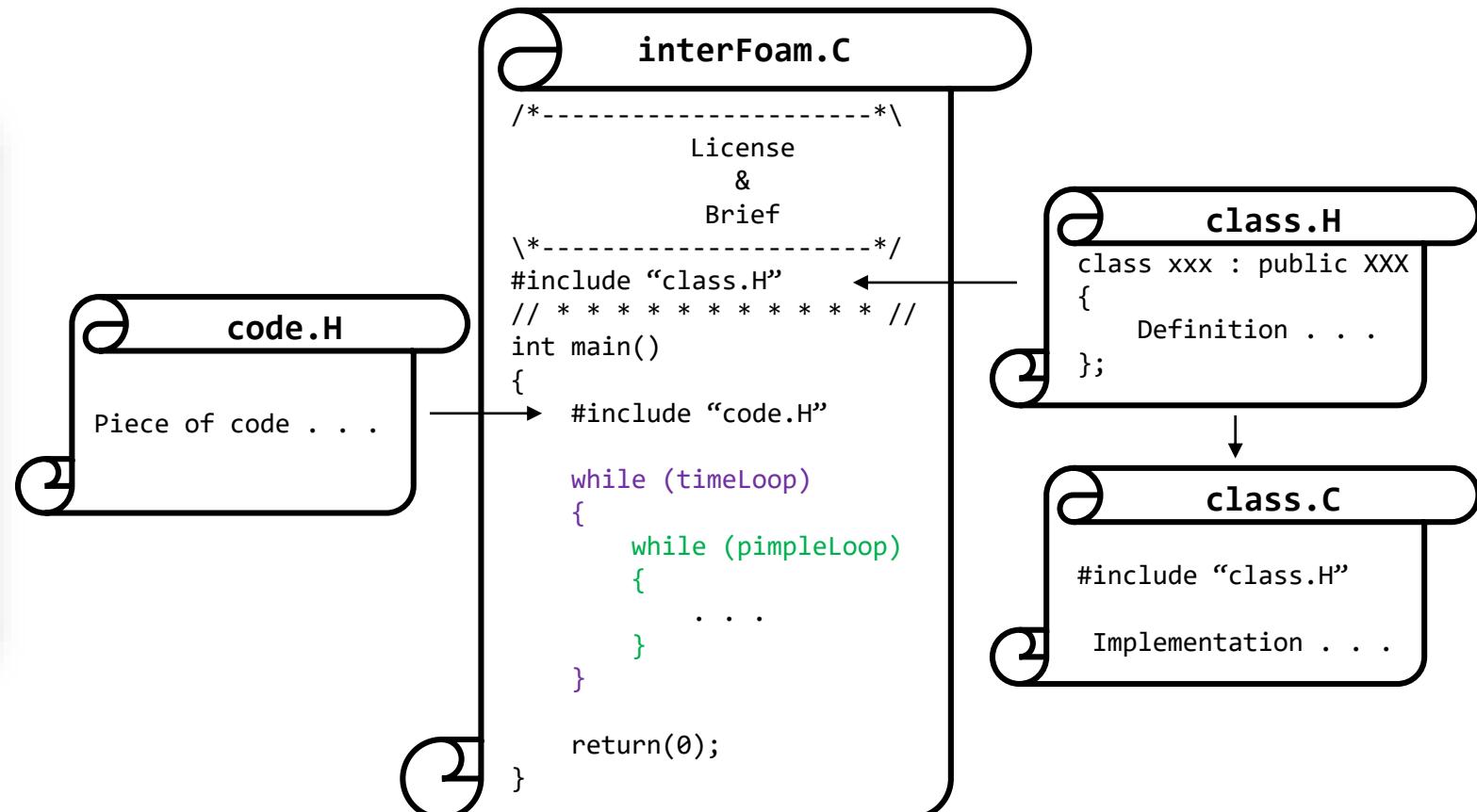
- Schemes
- Linear equation solvers
- controlDict

Solver Structure

A walk-through of interFoam



Header files, source files, compilation and linking



High-level structure of `interFoam`

Solver Structure

A walk-through of interFoam

```
interFoam.C
/*
=====
License
&
Brief
\*-
#include "class.H"
// * * * * *
int main()
{
    #include "code.H"

    while (timeLoop)
    {
        while (pimpleLoop)
        {
            ...
        }
    }

    return(0);
}
```



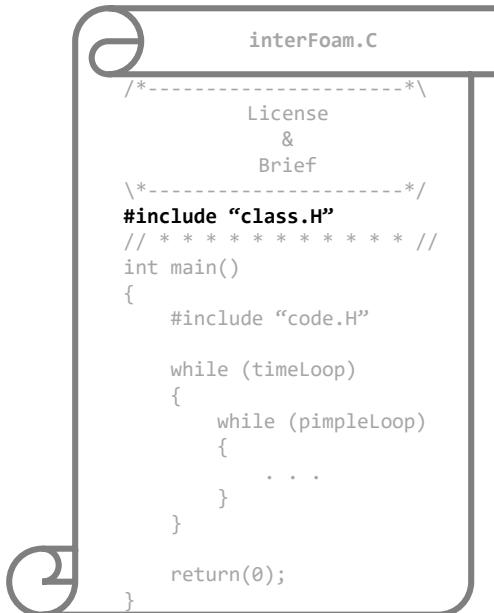
interFoam.C

Go to the documentation of this file.

```
1  /*
2  =====
3  \\\   / F ield           | OpenFOAM: The Open Source CFD Toolbox
4  \\\   / O peration        |
5  \\\   / A nd              | www.openfoam.com
6  \\\  M anipulation       |
7  -----
8  Copyright (C) 2011-2017 OpenFOAM Foundation
9  Copyright (C) 2020 OpenCFD Ltd.
10 -----
11 License
12     This file is part of OpenFOAM.
13
14 OpenFOAM is free software: you can redistribute it and/or modify it
15 under the terms of the GNU General Public License as published by
16 the Free Software Foundation, either version 3 of the License, or
17 (at your option) any later version.
18
19 OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
20 ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
21 FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
22 for more details.
23
24 You should have received a copy of the GNU General Public License
25 along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
26
27 Application
28     interFoam
29
30 Group
31     grpMultiphaseSolvers
32
33 Description
34     Solver for two incompressible, isothermal immiscible fluids using a VOF
35     (volume of fluid) phase-fraction based interface capturing approach,
36     with optional mesh motion and mesh topology changes including adaptive
37     re-meshing.
38
39 */
40
```

Solver Structure

A walk-through of interFoam



```
interFoam.C
/*
 *-----*
 * License
 * &
 * Brief
 *-----*/
#include "class.H"
// * * * * *
int main()
{
    #include "code.H"

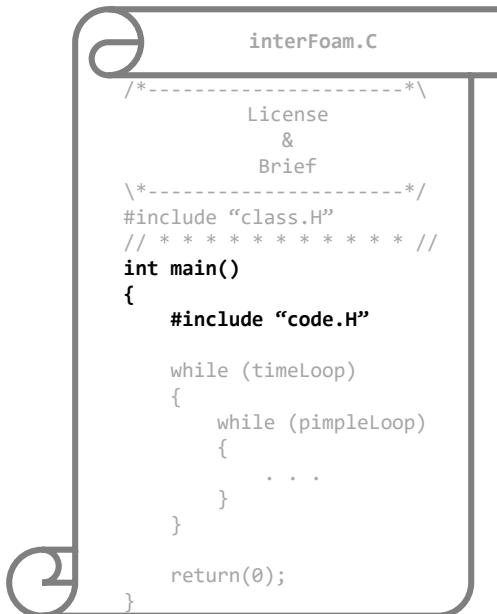
    while (timeLoop)
    {
        while (pimpleLoop)
        {
            ...
        }
    }

    return(0);
}
```

```
33 Description
34     Solver for two incompressible, isothermal immiscible fluids using a VOF
35     (volume of fluid) phase-fraction based interface capturing approach,
36     with optional mesh motion and mesh topology changes including adaptive
37     re-meshing.
38
39 /**
40
41 #include "fvCFD.H"
42 #include "dynamicFvMesh.H"
43 #include "CMULES.H"
44 #include "EulerDdtScheme.H"
45 #include "localEulerDdtScheme.H"
46 #include "CrankNicolsonDdtScheme.H"
47 #include "subCycle.H"
48 #include "immiscibleIncompressibleTwoPhaseMixture.H"
49 #include "incompressibleInterPhaseTransportModel.H"
50 #include "turbulentTransportModel.H"
51 #include "pimpleControl.H"
52 #include "fvOptions.H"
53 #include "CorrectPhi.H"
54 #include "fvcSmooth.H"
55
56 // * * * * *
57
58 int main(int argc, char *argv[])
59 {
```

Solver Structure

A walk-through of interFoam



```
interFoam.C

/*
 *----*
 License
 &
 Brief
 \*-----*/
#include "class.H"
// * * * * * * * * * * *
int main()
{
    #include "code.H"

    while (timeLoop)
    {
        while (pimpleLoop)
        {
            ...
        }
    }

    return(0);
}
```

```
58 int main(int argc, char *argv[])
59 {
60     argList::addNote
61     (
62         "Solver for two incompressible, isothermal immiscible fluids"
63         " using VOF phase-fraction based interface capturing.\n"
64         "With optional mesh motion and mesh topology changes including"
65         " adaptive re-meshing."
66     );
67
68     #include "postProcess.H"
69
70     #include "addCheckCaseOptions.H"
71     #include "setRootCaseLists.H"
72     #include "createTime.H"
73     #include "createDynamicFvMesh.H"
74     #include "initContinuityErrs.H"
75     #include "createDyMControls.H"
76     #include "createFields.H"
77     #include "createAlphaFluxes.H"
78     #include "initCorrectPhi.H"
79     #include "createUfIfPresent.H"
80
81     if (!LTS)
82     {
83         #include "CourantNo.H"
84         #include "setInitialDeltaT.H"
85     }
86 }
```

Solver Structure

A walk-through of interFoam

```
interFoam.C

/*
-----*\
License
&
Brief
\*-----*/
#include "class.H"
// * * * * * * * * * * * *
int main()
{
    #include "code.H"

    while (timeLoop)
    {
        while (pimpleLoop)
        {
            . . .
        }
    }
    return(0);
}
```

Solver Structure

A walk-through of interFoam

```
interFoam.C
/*
-----*
License
&
Brief
\*-----*/
#include "class.H"
// * * * * *
int main()
{
    #include "code.H"

    while (timeLoop)
    {
        while (pimpleLoop)
        {
            ...
        }
    }

    return(0);
}
```

33

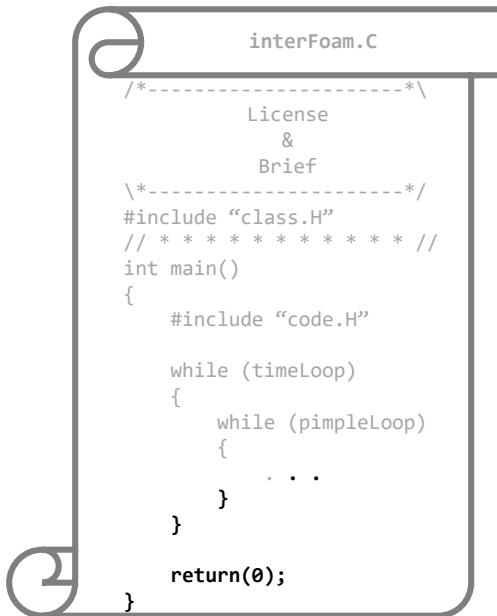
```
Info<< "Reading transportProperties\n" << endl;
immiscibleIncompressibleTwoPhaseMixture mixture(U, phi);
```

createFields.H

```
109 // --- Pressure-velocity PIMPLE corrector loop
110 while (pimple.loop())
111 {
112     if (pimple.firstIter() || moveMeshOuterCorrectors)
113     {
114         mesh.update();
115
116         if (mesh.changing())
117         {
118             // Do not apply previous time-step mesh compression flux
119             // if the mesh topology changed
120             if (mesh.topoChanging())
121             {
122                 talphaPhi1Corr0.clear();
123             }
124
125             gh = (g & mesh.C()) - ghRef;
126             ghf = (g & mesh.Cf()) - ghRef;
127
128             MRF.update();
129
130             if (correctPhi)
131             {
132                 // Calculate absolute flux
133                 // from the mapped surface velocity
134                 phi = mesh.Sf() & Uf();
135
136                 #include "correctPhi.H"
137
138                 // Make the flux relative to the mesh motion
139                 fvc::makeRelative(phi, U);
140
141                 mixture.correct();
142
143             }
144
145             if (checkMeshCourantNo)
146             {
147                 #include "meshCourantNo.H"
148             }
149         }
150     }
151 }
```

Solver Structure

A walk-through of interFoam



```
151 #include "alphaControls.H"
152 #include "alphaEqnSubCycle.H"
153
154 mixture.correct();
155
156 if (pimple.frozenFlow())
157 {
158     continue;
159 }
160
161 #include "UEqn.H"
162
163 // --- Pressure corrector loop
164 while (pimple.correct())
165 {
166     #include "pEqn.H"
167 }
168
169 if (pimple.turbCorr())
170 {
171     turbulence->correct();
172 }
173
174 runTime.write();
175
176 runTime.printExecutionTime(Info);
177 }
178
179 Info<< "End\n" << endl;
180
181 return 0;
182 }
183
184
185
186 // ****//
```



Science and
Technology
Facilities Council

Scientific Computing

Hands On Session

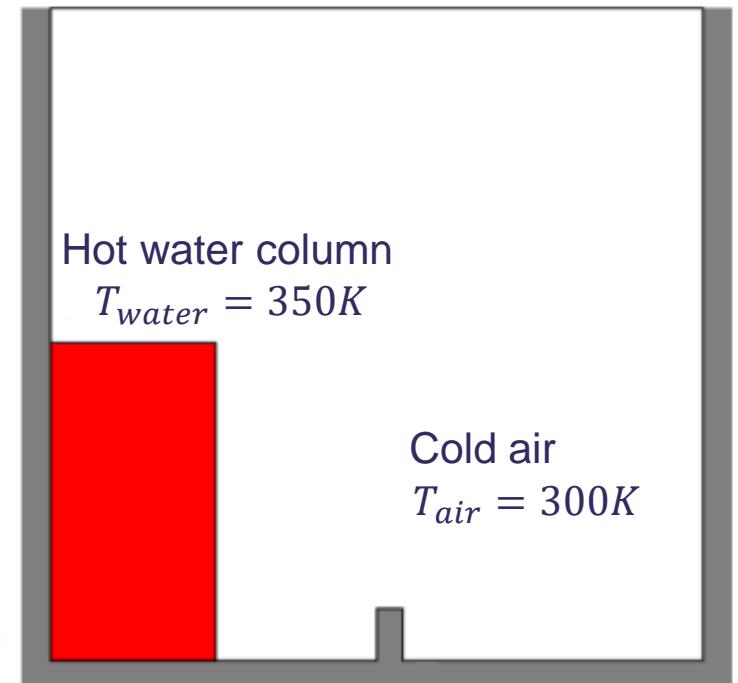


Hands On Session

A 2-D dam break with hot water column

A column of hot water with a temperature of $T_{water} = 350K$ at rest located behind a membrane on the left side of a tank. The rest of the tank is filled with cold air with a temperature of $T_{air} = 300K$. At time $t = 0s$, the membrane is removed and the column of hot water collapses. During the collapse, the hot water impacts an obstacle at the bottom of the tank and exchanges heat continuously with the surrounding cold air.

Hint: Modify the `interFoam` solver to implement unsteady passive scalar transport equations for heat exchange.



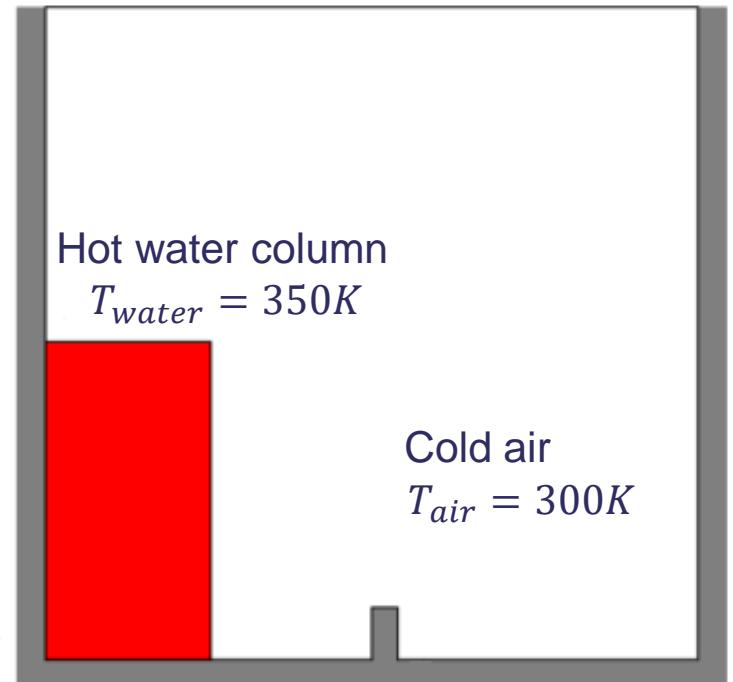
Geometry of the dam break

Hands On Session

Step 0: Analysis and planning

$$\frac{\partial \rho T}{\partial t} + \nabla \cdot (\rho \mathbf{U} T) - \nabla \cdot (D_{eff} \nabla T) = 0$$

$$D_{eff} = \frac{\alpha k_1}{C_{v1}} + \frac{(1 - \alpha)k_2}{C_{v2}}$$



Geometry of the dam break

Hands On Session

Step 1: Preparation

- Open the “*OpenFOAM Sandbox*” terminal environment from *Applications >> Software*
- Source the compiled copy of OpenFOAM from the course_material directory

```
source /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/etc/bashrc
```

Hands On Session

Step 1: Preparation

- Create OpenFOAM user directory and a subfolder for the modified solver

```
mkdir -p $WM_PROJECT_USER_DIR/applications && cd $WM_PROJECT_USER_DIR/applications
```

```
cp -r /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/applications/solvers/multiphase/interFoam/ interThermoFoam  
cp -r /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/applications/solvers/multiphase/VoF/ .
```

```
rm -r interThermoFoam/interMixingFoam interThermoFoam/overInterDyMFoam
```

Hands On Session

Step 1: Preparation

- Create a subfolder in the OpenFOAM® user directory for the test case

```
cd $WM_PROJECT_USER_DIR && mkdir run && cd run
```

```
cp -r /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-  
v2212/tutorials/multiphase/interFoam/laminar/damBreakPermeable/ .
```

Hands On Session

Step 2: Rename the application

- Rename source file interFoam.C

```
cd $WM_PROJECT_USER_DIR/applications/interThermoFoam
```

```
mv interFoam.C interThermoFoam.C
```

- Update config file Make/files

```
interThermoFoam.C  
EXE = $(FOAM_USER_APPBIN)/interThermoFoam
```

Update interFoam.C
into interThermoFoam.c

Update FOAM_APPBIN
into FOAM_USER_APPBIN

Hands On Session

Step 2: Rename the application

- Update source file comments

```
Application
  interThermoFoam
Group
  grpMultiphaseSolvers
Description
  Solver for two incompressible, isothermal immiscible (volume of fluid) phase-fraction based interface capturing approach, with optional mesh motion and mesh topology changes including adaptive re-meshing, with simplified thermal equations.
/*
*=-----*
```

Update `interFoam.C`
into `interThermoFoam.c`

Add a description of the new functionality

```
/*
=====
 \ \ / F ield
  \ \ / O peration
   \ \ / A nd
    \ \ M anipulation
=====
 OpenFOAM: The Open Source CFD Toolbox
 www.openfoam.com

Copyright (C) 2011-2017 OpenFOAM Foundation
Copyright (C) 2020 OpenCFD Ltd.

License
  This file is part of OpenFOAM.

  OpenFOAM is free software: you can redistribute it and/or modify it
  under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
  ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
  for more details.

  You should have received a copy of the GNU General Public License
  along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Application
  interThermoFoam
Group
  grpMultiphaseSolvers
Description
  Solver for two incompressible, isothermal immiscible fluids using a VOF (volume of fluid) phase-fraction based interface capturing approach, with optional mesh motion and mesh topology changes including adaptive re-meshing, with simplified thermal equations.
*/
#include "fvCFD.H"
#include "dynamicFvMesh.H"
#include "CMULES.H"
-- INSERT --
```

`interThermoFoam.C`

Hands On Session

Step 3: Adding new Entries into source codes

- Add definitions of k and C_v in `createFields.H`

A instance named k_1
Name of the dimensioned scalar is k
The dimension of k : $[W/(m \cdot K)] = [M^1 L^1 t^{-3} T^{-1} Q^0 C^0 l^0]$
Lookup transportProperties under the sub-dictionary phase1Name()
Lookup keyword k

A instance named C_{v1}
Name of the dimensioned scalar is C_v
The dimension of C_v : $[J/(kg \cdot K)] = [M^0 L^2 t^{-2} T^{-1} Q^0 C^0 l^0]$
Lookup transportProperties under the sub-dictionary phase1Name()
Lookup keyword C_v

Add k and C_v after the definition of mixture

```
immiscibleIncompressibleTwoPhaseMixture mixture(U, phi);  
  
volScalarField& alpha1(mixture.alpha1());  
volScalarField& alpha2(mixture.alpha2());  
  
const dimensionedScalar& rho1 = mixture.rho1();  
const dimensionedScalar& rho2 = mixture.rho2();  
  
dimensionedScalar k1  
(  
    "k",  
    dimensionSet(1, 1, -3, -1, 0, 0, 0),  
    mixture.subDict(mixture.phase1Name()),  
    "k"  
);  
  
dimensionedScalar k2  
(  
    "k",  
    dimensionSet(1, 1, -3, -1, 0, 0, 0),  
    mixture.subDict(mixture.phase2Name()),  
    "k"  
);  
  
dimensionedScalar Cv1  
(  
    "Cv",  
    dimensionSet(0, 2, -2, -1, 0, 0, 0),  
    mixture.subDict(mixture.phase1Name()),  
    "Cv"  
);  
  
dimensionedScalar Cv2  
(  
    "Cv",  
    dimensionSet(0, 2, -2, -1, 0, 0, 0),  
    mixture.subDict(mixture.phase2Name()),  
    "Cv"  
);
```

`createFields.H`

Hands On Session

Step 3: Adding new Entries into source codes

- Initialise the temperature field in `createFields.H`

Create scalar field T
Create object for input/output operations
Name of the dictionary file to read/write
runtime directory
Object registry
Read the dictionary in the runtime directory
write the value in the runtime directory
Link object to mesh

```
(  
    "Cv",  
    dimensionSet(0, 2, -2, -1, 0, 0, 0),  
    mixture.subDict(mixture.phase1Name()),  
    "Cv"  
)  
  
dimensionedScalar Cv2  
(  
    "Cv",  
    dimensionSet(0, 2, -2, -1, 0, 0, 0),  
    mixture.subDict(mixture.phase2Name()),  
    "Cv"  
)  
  
volScalarField T  
(  
    IOobject  
    (  
        "T",  
        runTime.timeName(),  
        mesh,  
        IOobject::MUST_READ,  
        IOobject::AUTO_WRITE  
    ),  
    mesh  
)  
  
// Need to store rho for ddt(rho, U)  
volScalarField rho  
(  
    IOobject  
    (  
        "rho",  
        runTime.timeName(),  
        mesh,  
        IOobject::READ_IF_PRESENT  
    ),  
    alpha1*rho1 + alpha2*rho2  
)  
rho.oldTime();
```

`createFields.H`

Hands On Session

Step 4: Implement model equations

- Create a new header file TEqn.H in the source folder

```
cd $WM_PROJECT_USER_DIR/applications/interThermoFoam
```

```
touch TEqn.H
```

- Define D_{eff} in TEqn.H and solve the passive temperature PDE

$$D_{eff} = \frac{\alpha k_1}{C_{v1}} + \frac{(1 - \alpha)k_2}{C_{v2}}$$

$$\frac{\partial \rho T}{\partial t} + \nabla \cdot (\rho \mathbf{U} T) - \nabla \cdot (D_{eff} \nabla T) = 0$$

```
volScalarField Deff
(
    "Deff"
    (alpha1*k1/Cv1 + (scalar(1)-alpha1)*k2/Cv2)
);

solve
(
    fvm::ddt(rho, T)
    +fvm::div(rhoPhi, T)
    -fvm::laplacian(Deff, T)
);
```

TEqn.H

Hands On Session

Step 5: Include TEqn.H into solution algorithm

```
    mixture.correct();

    if (pimple.frozenFlow())
    {
        continue;
    }

    #include "UEqn.H"
    #include "TEqn.H" -> Include TEqn.H

    // --- Pressure corrector loop
    while (pimple.correct())
    {
        #include "pEqn.H"
    }

    if (pimple.turbCorr())
    {
```

interThermoFoam.C



Science and
Technology
Facilities Council

Scientific Computing

Hands On Session

Step 6: Compile interThermoFoam

```
cd $WM_PROJECT_USER_DIR/applications/interThermoFoam
```

```
wmake
```

```
we11608$ wmake
Making dependencies: interThermoFoam.C
g++ -std=c++11 -m64 -pthread -DOPENFOAM=2212 -DWM_DP -DWM_LABEL_SIZE=32 -Wall -Wextra -Wold-style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -Wno-unknown-pragmas -O3 -DNRepository -ftemplate-depth=100 -I./VoF -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/phaseSystemModels/twoPhaseInter/incompressibleInterPhaseTransportModel/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/finiteVolume/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/meshTools/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/sampling/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/dynamicFvMesh/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/twoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/interfaceProperties/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/twoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/phaseIncompressible/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/immiscibleIncompressibleTwoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude/postProcess.H:147,
    from InterThermoFoam.C:68,
./createFields.H: In function 'int main(int, char**)':
./createFields.H:45:8: error: expected ')' before 'dimensionSet'
45 |     dimensionSet(1, 1, -3, -1, 0, 0, 0),
   |             ^
./createFields.H:44:1: note: to match this '('
44 |     dimensionSet(1, 1, -3, -1, 0, 0, 0),
   |     ^
./createFields.H:47:29: error: invalid use of non-static member function 'const Foam::word& Foam::twoPhaseMixture::phaseName() const'
47 |     mixture.suodict(mixture.phaseName),
   |     ^
In file included from /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/incompressible/lnInclude/incompressibleTwoPhaseMixture.H:43,
                  from /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/immiscibleIncompressibleTwoPhaseMixture/lnInclude/immiscibleIncompressibleTwoPhaseMixture.H:41,
```

Hint: Check spelling, and semicolons if failed

Failed?

or

Succeeded?

```
we31608$ wmake
Making dependencies: interThermoFoam.C
g++ -std=c++11 -m64 -pthread -DOPENFOAM=2212 -DWM_DP -DWM_LABEL_SIZE=32 -Wall -Wextra -Wold-style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -Wno-unknown-pragmas -O3 -DNRepository -ftemplate-depth=100 -I./VoF -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/phaseSystemModels/twoPhaseInter/incompressibleInterPhaseTransportModel/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/finiteVolume/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/meshTools/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/sampling/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/dynamicFvMesh/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/twoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/phaseIncompressible/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/immiscibleIncompressibleTwoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude/postProcess.H:147,
    from InterThermoFoam.C:68,
./createFields.H: In function 'int main(int, char**)':
./createFields.H:45:8: error: expected ')' before 'dimensionSet'
45 |     dimensionSet(1, 1, -3, -1, 0, 0, 0),
   |             ^
./createFields.H:44:1: note: to match this '('
44 |     dimensionSet(1, 1, -3, -1, 0, 0, 0),
   |     ^
./createFields.H:47:29: error: invalid use of non-static member function 'const Foam::word& Foam::twoPhaseMixture::phaseName() const'
47 |     mixture.suodict(mixture.phaseName),
   |     ^
In file included from /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/incompressible/lnInclude/incompressibleTwoPhaseMixture.H:43,
                  from /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/phaseSystemModels/twoPhaseInter/incompressibleInterPhaseTransportModel/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/twoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/meshTools/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/sampling/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/dynamicFvMesh/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/twoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/phaseIncompressible/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/immiscibleIncompressibleTwoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude/postProcess.H:147,
    from InterThermoFoam.C:68,
./createFields.H: In function 'int main(int, char**)':
./createFields.H:45:8: error: expected ')' before 'dimensionSet'
45 |     dimensionSet(1, 1, -3, -1, 0, 0, 0),
   |             ^
./createFields.H:44:1: note: to match this '('
44 |     dimensionSet(1, 1, -3, -1, 0, 0, 0),
   |     ^
./createFields.H:47:29: error: invalid use of non-static member function 'const Foam::word& Foam::twoPhaseMixture::phaseName() const'
47 |     mixture.suodict(mixture.phaseName),
   |     ^
In file included from /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/incompressible/lnInclude/incompressibleTwoPhaseMixture.H:43,
                  from /mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/phaseSystemModels/twoPhaseInter/incompressibleInterPhaseTransportModel/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/twoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/meshTools/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/sampling/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/dynamicFvMesh/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/transportModels/twoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/phaseIncompressible/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/TurbulenceModels/immiscibleIncompressibleTwoPhaseMixture/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude -I/mnt/ceph-training/course_materials/OpenFOAM/OpenFOAM-v2212/src/OpenFOAM/lnInclude/postProcess.H:147,
    from InterThermoFoam.C:68,
```



Science and
Technology
Facilities Council

Scientific Computing

Hands On Session

Step 7: Prepare the test case

- Setup temperature field T in `0.orig` folder

```
cd $WM_PROJECT_USER_DIR/run/damBreakPermeable
```

```
cp 0.orig/alpha.water 0.orig/T
```

- Change object to T
- Change dimension of T
- Change internal Field value as 300

Hands On Session

Step 7: Prepare the test case

- Setup initial condition of water temperature T in setFieldsDict

```
=====
 \ \ / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
  \ \ / O p e r a t i o n | Version: v2212
   \ \ / A n d          | Website: www.openfoam.com
    \ \ / M a n i p u l a t i o n
\*----- C++ -----
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;
    object     setFieldsDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

defaultFieldValues
(
    volScalarFieldValue alpha.water 0
);

regions
(
    boxToCell
    {
        box (0 0 -1) (0.2461 0.292 1);
        fieldValues
        (
            volScalarFieldValue alpha.water 1
            volScalarFieldValue T 350
        );
    }
);

// ****

```

setFieldsDict



Set water temperature as 350

Hands On Session

Step 7: Prepare the test case

- Setup fluid thermal properties in transportProperties

```
/*-----*\n| Field | Operation | OpenFOAM: The Open Source CFD Toolbox\n|       | And       | Version: v2212\n|       | Manipulation | Website: www.openfoam.com\n*-----*/\nFoamFile\n{\n    version      2.0;\n    format       ascii;\n    class        dictionary;\n    object       transportProperties;\n}\n// * * * * *\n\nphases          (water air);\n\nwater\n{\n    transportModel Newtonian;\n    nu             1e-06;\n    rho            1000;\n    k              0.66; red arrow\n    Cv             3900;\n}\n\nair\n{\n    transportModel Newtonian;\n    nu             1.48e-05;\n    rho            1;\n    k              0.03;\n    Cv             718; red arrow\n}\n\nsigma          0.07;\n\n// *****\n
```

transportProperties

Setup k and C_v for both phases

Hands On Session

Step 7: Prepare the test case

- Setup linear solver for T in fvSolution

```
        "(U|T|k|epsilon).*"  
    {  
        solver      smoothSolver;  
        smoother   symGaussSeidel;  
        tolerance  1e-06;  
        relTol     0;  
        minIter    1;  
    }  
  
PIMPLE  
{
```

fvSolution

Add T in smooth linear solver

- Setup divSchemes for $\operatorname{div}(\rho\phi, T)$ in fvSchemes

```
divSchemes  
{  
    div(rhoPhi,U) Gauss linearUpwind grad(U);  
    div(rhoPhi,T) Gauss upwind;  
    div(phi,alpha) Gauss vanLeer;  
    div(phirb,alpha) Gauss linear;  
    div(rhoPhi,k) Gauss upwind;  
    div(rhoPhi,epsilon) Gauss upwind;  
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;  
}
```

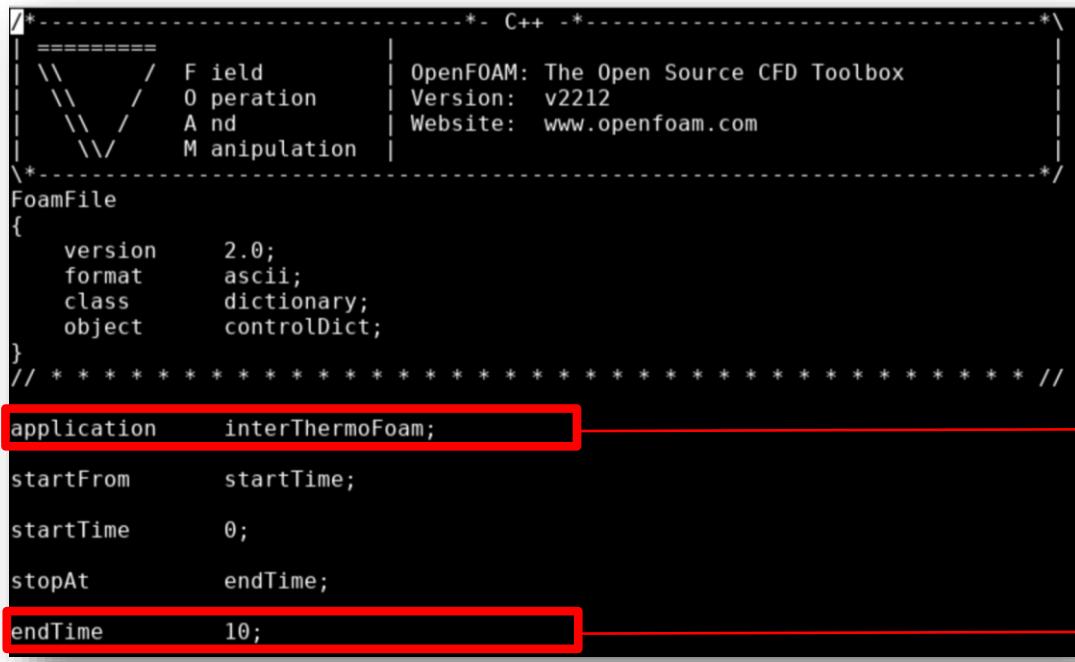
fvSchemes

Setup $\operatorname{div}(\rho\phi, T)$ by Gauss upwind

Hands On Session

Step 7: Prepare the test case

- Change the application name and set the endTime as 10s in controlDict



The screenshot shows a terminal window displaying the contents of a file named `controlDict`. The file is a `FoamFile` with the following structure:

```
/*-----\n---- / Field          OpenFOAM: The Open Source CFD Toolbox\n---- / Operation      Version: v2212\n---- / And           Website: www.openfoam.com\n---- / Manipulation\n\\-----*/\nFoamFile\n{\n    version      2.0;\n    format       ascii;\n    class        dictionary;\n    object       controlDict;\n}\n// * * * * *\n\napplication      interThermoFoam;\nstartFrom        startTime;\nstartTime        0;\nstopAt          endTime;\nendTime          10;
```

Two specific lines are highlighted with red boxes:

- The line `application interThermoFoam;`
- The line `endTime 10;`

Red arrows point from these highlighted lines to the right, each accompanied by a descriptive text label:

- Change the application name as `interThermoFoam`**
- Extend the endTime to 10s**

`controlDict`

Hands On Session

Step 8: Execute the test case

```
cd $WM_PROJECT_USER_DIR/run/damBreakPermeable
```

```
./Allrun
```

```
apptainer-openfoam2206:~/OpenFOAM/we31608-v2212/run/damBreakPermeable/  
we31608$ ./Allrun  
Restore 0/ from 0.orig/  
Running blockMesh on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
Running setFields on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
Running decomposePar on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
Running interThermoFoam (4 processes) on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
[
```

Running ...

```
apptainer-openfoam2206:~/OpenFOAM/we31608-v2212/run/damBreakPermeable/  
we31608$ ./Allrun  
Restore 0/ from 0.orig/  
Running blockMesh on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
Running setFields on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
Running decomposePar on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
Running interThermoFoam (4 processes) on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
Running redistributePar (4 processes) on /home/we31608/OpenFOAM/we31608-v2212/run/damBreakPermeable  
apptainer-openfoam2206:~/OpenFOAM/we31608-v2212/run/damBreakPermeable/  
we31608$ [
```

Done

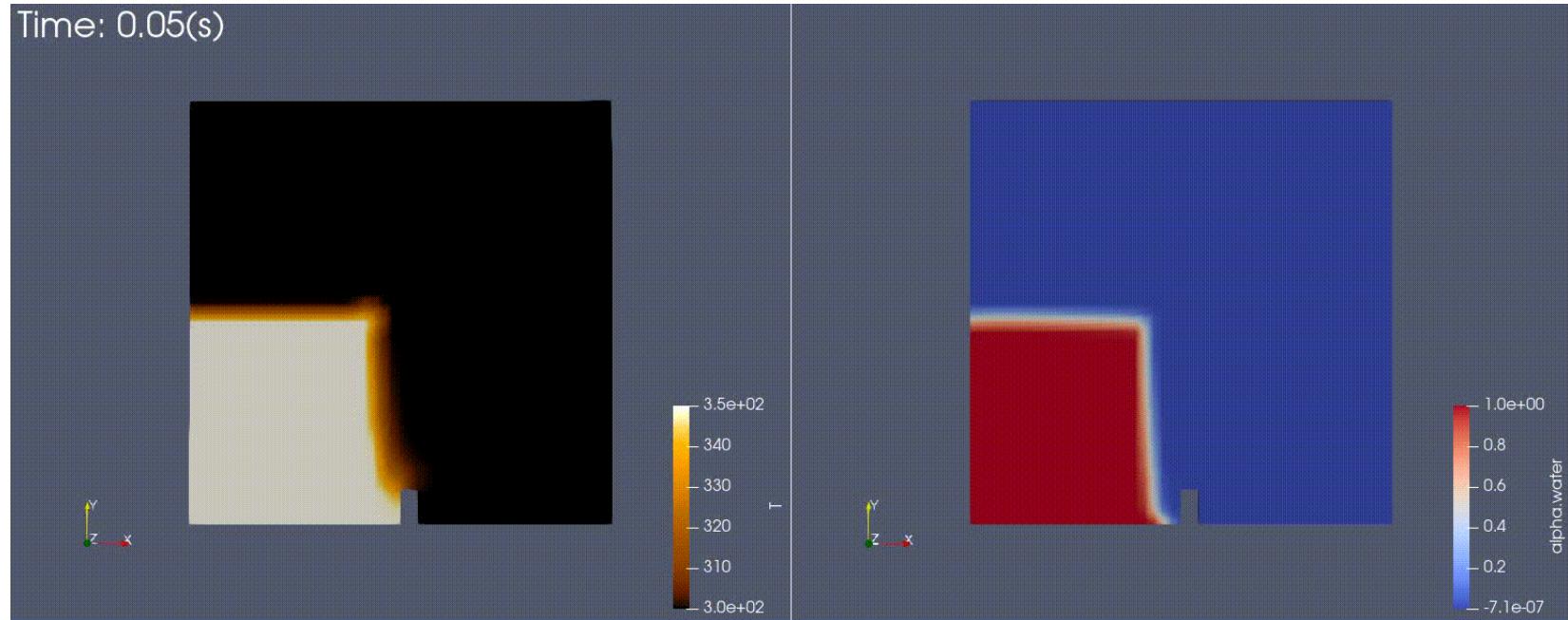
Hands On Session

Step 9: Check the result

```
cd $WM_PROJECT_USER_DIR/run/damBreakPermeable
```

```
touch open.foam
```

- Open the “*Paraview*” terminal from
Applications >> Utilities





Science and
Technology
Facilities Council

Scientific Computing

Questions?



References

OpenFOAM: Programmer's Guide v2212

<https://sourceforge.net/projects/openfoam/files/v2212/ProgrammersGuide.pdf/download>

OpenFOAM: Extended Code Guide v2112

<https://www.openfoam.com/documentation/guides/v2112/doc/>

Maric, T., Höpken, J. and Mooney, K.G., *The OpenFOAM® Technology Primer*. Sourceflux, 2014.



Science and
Technology
Facilities Council

Scientific Computing



Science and
Technology
Facilities Council

Scientific Computing

Thank you

A large, abstract graphic element occupies the right side of the slide. It features a red triangle pointing downwards on a red background. Overlaid on this are several white line patterns: a series of horizontal lines at the top, a cluster of vertical dots in the middle, and two sets of parallel curved lines on the right side.

scd.stfc.ac.uk

[@SciComp_STFC](https://twitter.com/SciComp_STFC)

STFC Scientific Computing

wendi.liu@stfc.ac.uk