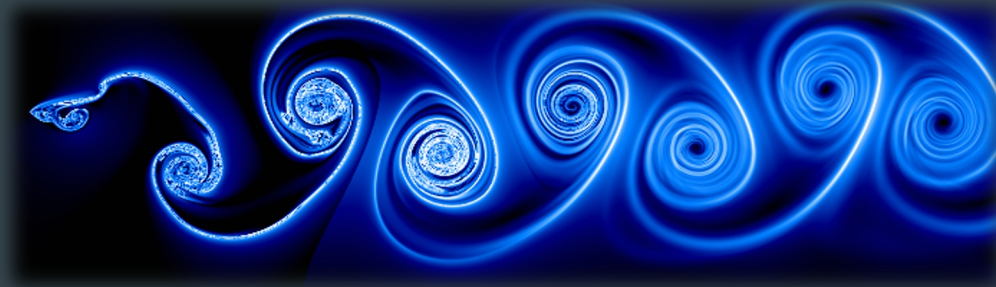


# Getting Started with CFD using OpenFOAM



Open  FOAM

Dr. Chandan Bose

Assistant Professor of Aerospace Engineering

University of Birmingham, UK

[www.chandanbose.com](http://www.chandanbose.com)

*Explore it! Tame it! Use it!*



UNIVERSITY OF  
BIRMINGHAM

"THIS OFFERING IS NOT APPROVED OR ENDORSED BY  
OpenCFD LIMITED, PRODUCER AND DISTRIBUTOR OF THE  
OpenFOAM SOFTWARE AND OWNER OF OpenFOAM® AND  
OpenCFD® TRADEMARKS."

...


# Bio-Inspired Fluid-Structure Interaction Laboratory, University of Birmingham



UNIVERSITY OF  
BIRMINGHAM







PageDiscussion

## Technical Committee for D

Contents [hide]

1 Committee Members of D&T


1.1 Our goals

1.2 Committee Workflow


1.3 How you can get involved

### Committee Members of D&T

The tutorial collection is managed and supported by E



József Nagy, Committee chair



Andrew Heather, Co-chair

What links here

Related changes

Special pages

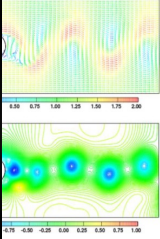
Printable version

Permanent link

Page information


Cite this page





1993

21



OpenFOAM®  
Journal

# Open Field of Operation And Manipulation (FOAM)

- A set of open-source libraries for continuum mechanics (GNU Public License)
- C++ programming language: Object Oriented Programming (OOP)
- Based on Finite Volume Method (FVM)

## Why OpenFOAM?

- Open source, has a large community – Free (GNU Public License)!!
  - An efficient library of C++ modules (OOP):  
**High level programming!**
  - FVM solvers can be tailored for a specific need ...
  - Extensive multi-physics capabilities ...
  - Cross-platform coupling – FSI!
- No black magic!!
  - Not a single executable – understanding of source code is must!
  - No native GUI...
  - Not so well documented...

# Historical Timeline of OpenFOAM

- **Henry Weller** and **Hrvoje Jasak** are the main initial contributors to its development.

**1989** Project was initially started under the title of **FOAM** at **Imperial college**, London. The idea was to create a simulation platform more powerful than **FORTTRAN**. Due to its object oriented features **C++** was chosen as the programming language.

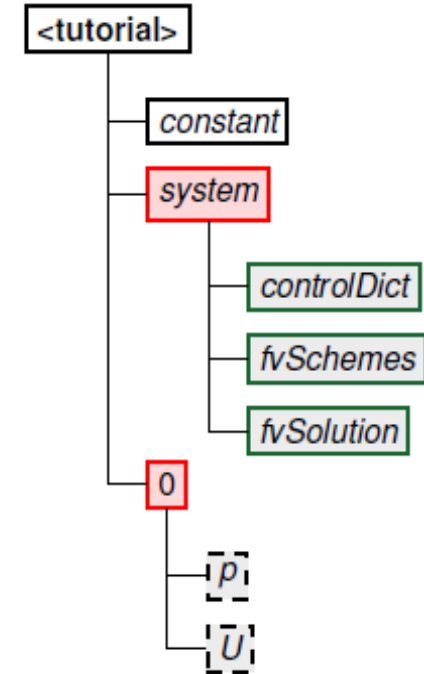
**10<sup>th</sup> Dec 2004** **OpenCFD Ltd** was founded and first version (v1.0) of **OpenFOAM** was released under the **GNU GPL** license.

**8<sup>th</sup> Aug 2011** **OpenCFD** was acquired by **Silicon Graphics International (SGI)**.

**15<sup>th</sup> Aug 2012** **OpenCFD** became a wholly owned subsidiary of **ESI group**.

# Primary OpenFOAM Forks

- OpenFOAM Foundation Release (official standard release)
- [www.openfoam.org](http://www.openfoam.org) : OpenFOAM- v11
- OpenFOAM + (official development release) – ESI OpenCFD
- [www.openfoam.com](http://www.openfoam.com) OpenFOAM- v2306
- FOAM – extend (community driven development branch)
- [www.openfoamwiki.net](http://www.openfoamwiki.net) foam-extend-5





# High Level Programming with OpenFOAM?

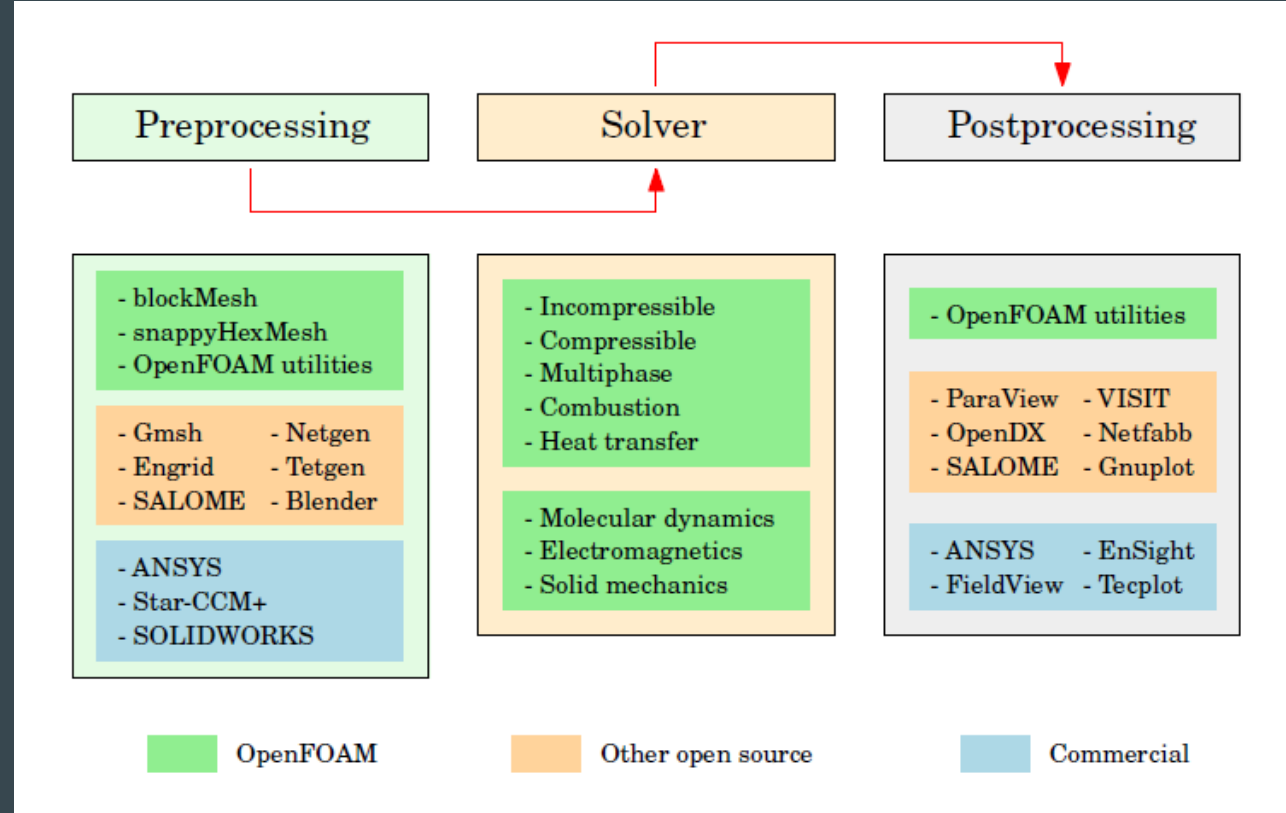
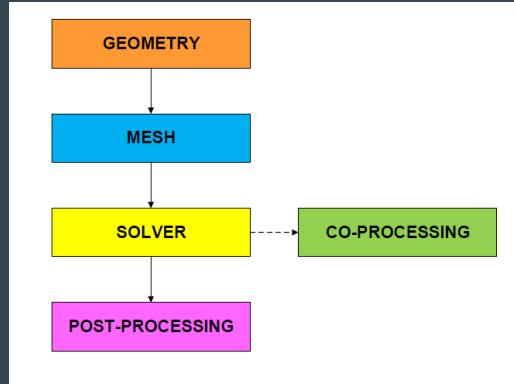
$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \phi \mathbf{U} - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p \longrightarrow$$

```
solve
(
    fvm::ddt(rho,U)
  + fvm::div(phi,U)
  - fvm::laplacian(mu,U)
  ==
  - fvc::grad(p)
);
```

# OpenFOAM VS Commercial Softwares?

	OpenFOAM	Commercial Software (ANSYS, ABAQUS etc.)
1. Cost effectiveness	✓	✗
2. Parallel computing	✓	✓
3. Source code	✓	✗
4. Redistribution of code	✓	✗
5. Collaborative development	✓	✗
6. Documentation	✗	✓
7. GUI / user friendliness	✗	✓

# Simulation Workflow



# Getting Started ...

```
$ source PATH-TO-DIR/etc/bashrc
```



```
~/.bashrc
```

System “bashrc”

→ Working with multiple versions of OpenFOAM

→ alias of7='source /opt/openfoam7/etc/bashrc'

→ Looking into the “etc” folder ...

→ OpenFOAM Environment Variables

foam

\$WM\_PROJECT\_USER\_DIR

Check the path of working directory

\$WM\_PROJECT\_USER\_DIR

\$ echo \$WM\_PROJECT\_USER\_DIR

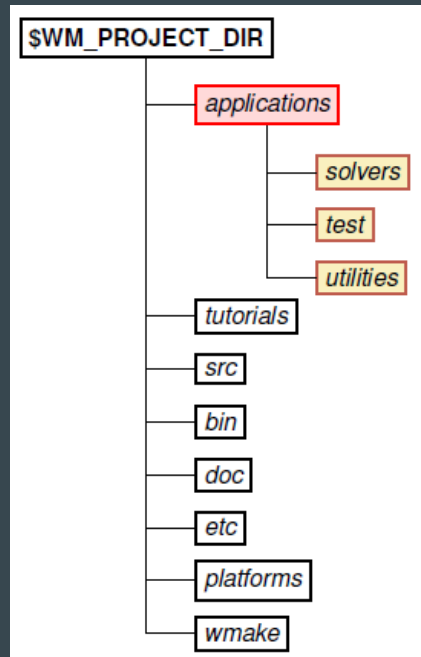
run → \$FOAM\_RUN

Check the path of run directory

tut → \$FOAM\_TUTORIALS

\$ echo \$FOAM\_RUN

src → \$FOAM\_SRC



\$ env | grep ^FOAM\_

# OF Aliases

```
cbose1991@DESKTOP-Q10GJQE:~$ cd OpenFOAM/OpenFOAM-v2012/etc/config.sh/  
cbose1991@DESKTOP-Q10GJQE:~/OpenFOAM/OpenFOAM-v2012/etc/config.sh$ ls  
CGAL          ccmio          functions      mgridgen       scotch  
FFTW          cmake          gperftools    mpi            settings  
adios2        compiler       hypre         paraview       setup  
aliases       completion_cache kahip         paraview-system unset  
bash_completion example        metis         petsc          vtk  
cbose1991@DESKTOP-Q10GJQE:~/OpenFOAM/OpenFOAM-v2012/etc/config.sh$ nano aliases
```

```
# Change directory aliases  
# ~~~~~  
alias foam='cd ${WM_PROJECT_DIR:?}'  
  
alias src='cd ${WM_PROJECT_DIR:?}/src'  
alias lib='cd ${FOAM_LIBBIN:?}'  
alias app='cd ${WM_PROJECT_DIR:?}/applications'  
alias sol='cd ${WM_PROJECT_DIR:?}/applications/solvers'  
alias util='cd ${WM_PROJECT_DIR:?}/applications/utilities'  
alias tut='cd ${FOAM_TUTORIALS:-${WM_PROJECT_DIR:?}/tutorials}'  
alias run='cd ${FOAM_RUN:-${WM_PROJECT_USER_DIR:?}/run}'  
  
alias ufoam='cd ${WM_PROJECT_USER_DIR:?}'  
alias uapp='cd ${WM_PROJECT_USER_DIR:?}/applications'  
alias usol='cd ${WM_PROJECT_USER_DIR:?}/applications/solvers'  
alias util='cd ${WM_PROJECT_USER_DIR:?}/applications/utilities'
```

# Basic Structure

```
$ env | grep ^FOAM_
```

```
cbose1991@DESKTOP-Q10GJQE: ~  
cbose1991@DESKTOP-Q10GJQE:~$ env | grep ^FOAM_  
FOAM_TUTORIALS=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/tutorials  
FOAM_RUN=/home/cbose1991/OpenFOAM/cbose1991-v2012/run  
FOAM_APP=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/applications  
FOAM_API=2012  
FOAM_UTILITIES=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/applications/utilities  
FOAM_APPBIN=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/platforms/linux64Gcc63DPInt32Opt/bin  
FOAM_SOLVERS=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/applications/solvers  
FOAM_EXT_LIBBIN=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/ThirdParty/platforms/linux64Gcc63DPInt32/lib  
FOAM_USER_APPBIN=/home/cbose1991/OpenFOAM/cbose1991-v2012/platforms/linux64Gcc63DPInt32Opt/bin  
FOAM_SIGFPE=  
FOAM_LIBBIN=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/platforms/linux64Gcc63DPInt32Opt/lib  
FOAM_SRC=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/src  
FOAM_ETC=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/etc  
FOAM_SITE_APPBIN=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/site/2012/platforms/linux64Gcc63DPInt32Opt/bin  
FOAM_SITE_LIBBIN=/home/cbose1991/OpenFOAM/OpenFOAM-v2012/site/2012/platforms/linux64Gcc63DPInt32Opt/lib  
FOAM_INST_DIR=/opt  
FOAM_USER_LIBBIN=/home/cbose1991/OpenFOAM/cbose1991-v2012/platforms/linux64Gcc63DPInt32Opt/lib  
FOAM_MPI=openmpi-4.0.3  
FOAM_JOB_DIR=/opt/jobControl  
cbose1991@DESKTOP-Q10GJQE:~$
```



# Basic Structure

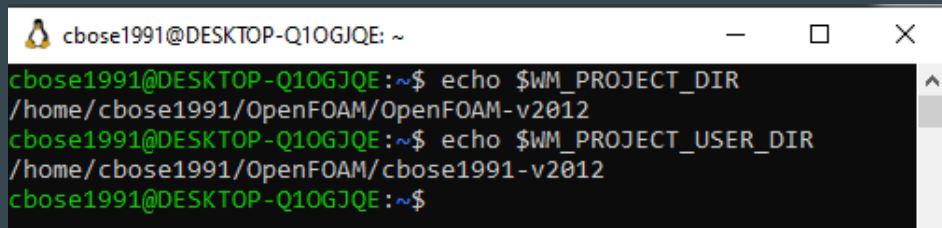
Check the path of working directory

**\$ echo \$WM\_PROJECT\_USER\_DIR**

Check the path of run directory

**\$ echo \$FOAM\_RUN**

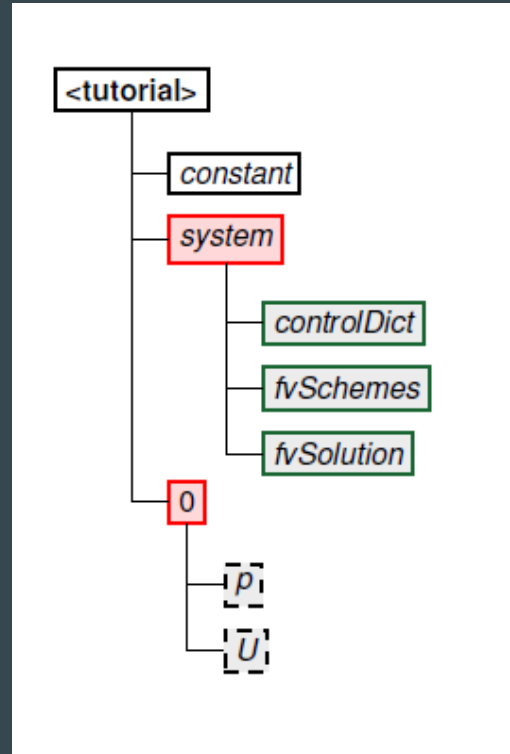
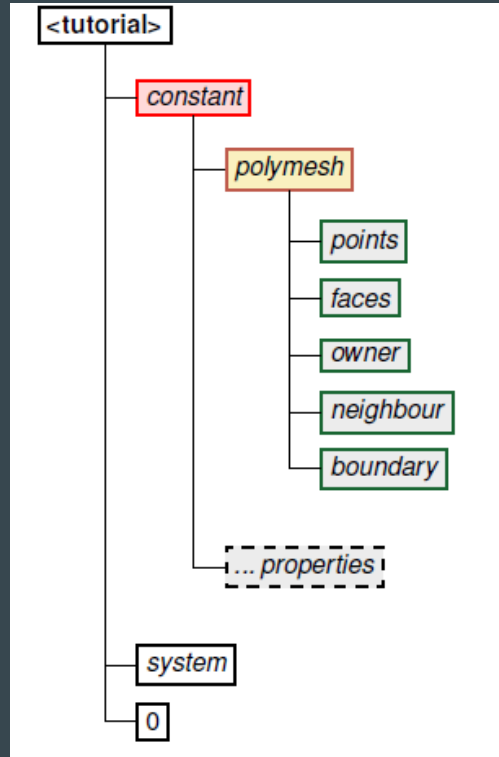
```
$HOME/OpenFOAM/  
├── $WM_PROJECT_USER_DIR  
└── $WM_PROJECT_DIR
```



```
cbose1991@DESKTOP-Q10GJQE: ~  
cbose1991@DESKTOP-Q10GJQE:~$ echo $WM_PROJECT_DIR  
/home/cbose1991/OpenFOAM/OpenFOAM-v2012  
cbose1991@DESKTOP-Q10GJQE:~$ echo $WM_PROJECT_USER_DIR  
/home/cbose1991/OpenFOAM/cbose1991-v2012  
cbose1991@DESKTOP-Q10GJQE:~$
```

```
cbose1991@DESKTOP-Q10GJQE:~$ tree -L 1 -d $WM_PROJECT_DIR  
/home/cbose1991/OpenFOAM/OpenFOAM-v2012  
├── META-INFO  
├── ThirdParty  
├── applications  
├── bin  
├── doc  
├── etc  
├── modules  
├── platforms  
├── src  
├── tutorials  
└── wmake
```

# OpenFOAM Case Structure



```
cavity
├── 0
│   ├── U
│   └── p
├── constant
│   └── transportProperties
└── system
    ├── PDRblockMeshDict
    ├── blockMeshDict
    ├── controlDict
    ├── fvSchemes
    └── fvSolution
```

## Incompressible flow

Solver	Description
1. <i>icoFoam</i>	Transient solver for incompressible, laminar flow of Newtonian fluids.
2. <i>simpleFoam</i>	Steady state solver for incompressible, turbulent flow.
3. <i> pisoFoam</i>	Transient solver for incompressible flow.

## Solver names describe their functionality

*simple*: **SIMPLE** algorithm used in steady-state solvers.

*piso*: **PISO** algorithm used in transient solvers  $\alpha_{CFL} < 1$ .

*pimple*: **PIMPLE** algorithm used in transient solvers.

*DyM*: Supports dynamic mesh (eg. mesh refinement, moving meshes).

# Initial and Boundary Conditions

## Standard initial conditions

```
<< cat $FOAM_RUN/cavity/0/U
```

```
.....
17 dimensions      [0 1 -1 0 0 0 0]; // Dimensions in SI base units [kg m s K mol A cd]
18
19 internalField    uniform (0 0 0); // Cell values
.....
```

## Initial and boundary conditions

```
<< cat $FOAM_RUN/cavity/0/U
```

```
.....
21 boundaryField    // Dictionary to specify boundary conditions
22 {
23     movingWall    // Boundary patch name
24     {
25         type       fixedValue; // Boundary condition type
26         value       uniform (1 0 0); // Boundary values
27     }
28
29     fixedWalls
30     {
31         type        noSlip;
32     }
33
34     frontAndBack
35     {
36         type        empty; // Boundary condition type for 2D simulations
37     }
38 }
.....
```

	BC type	Data	Example
1.	<i>fixedValue</i>	value	$U = (5, 10, 0)$
2.	<i>fixedGradient</i>	gradient	$\frac{\partial T}{\partial n} = 3.5$
3.	<i>zeroGradient</i>	–	$\frac{\partial p}{\partial n} = 0$

## controlDict dictionary

```
<< cat $FOAM_RUN/cavity/system/controlDict
```

```
18 application    icoFoam;    // Name of solver
19
20 startFrom       startTime;  // Useful for restart simulations
21
22 startTime       0;          // Starting time
23
24 stopAt          endTime;    // Stopping criteria
25
26 endTime         0.5;        // End time
27
28 deltaT          0.005;      // Time stepping (Default value of 1 for steady state solvers)
29
30 writeControl     timeStep;   // Criteria for writing results
31
32 writeInterval    20;        // Interval for writing results
33
34 purgeWrite       0;         // Disable rewriting over time directories
35
36 writeFormat      ascii;     // Format for writing results
37
38 writePrecision   6;         // Precision for writing results
39
40 writeCompression off;       // Disable compression for wrting results
41
42 timeFormat       general;    // Format for writing time directories
43
44 timePrecision    6;         // Precision for writing time directories
45
46 runTimeModifiable true;     // Allows modification of settings during the run
```

```
<< cat $FOAM_RUN/cavity/system/fvSchemes
```

```
18 ddtSchemes
19 {
20     default      Euler;           // Time discretisation schemes
21 }
23 gradSchemes      // Gradient evaluation schemes
24 {
25     default      Gauss linear;
26     grad(p)      Gauss linear;
27 }
29 divSchemes        // Discretisation of the convective terms
30 {
31     default      none;
32     div(phi,U)   Gauss linear;
33 }
35 laplacianSchemes  // Discretisation of the Laplacian term
36 {
37     default      Gauss linear orthogonal;
38 }
40 interpolationSchemes // Method of interpolation
41 {
42     default      linear;
43 }
45 snGradSchemes     // Surface normal gradient scheme
46 {
47     default      orthogonal;
48 }
```



```
18 solvers
19 {
20     p
21     {
22         solver      PCG;           // Solver to solve the system
23         preconditioner DIC;       // Preconditioner type
24         tolerance   1e-06;
25         relTol      0.05;
26     }
27     pFinal
28     {
29         $p;
30         relTol      0;
31     }
32     U
33     {
34         solver      smoothSolver;
35         smoother     symGaussSeidel;
36         tolerance    1e-05;
37         relTol       0;
38     }
39 }
40 }
41 }
42
43 PISO                      // PISO controls
44 {
45     nCorrectors      2;
46     nNonOrthogonalCorrectors 0;
47     ....
48 }
49 }
```

# High Level OOP with OpenFOAM

Open  FOAM

```
1 #include <iostream>
2 using namespace std;
3 // main() is where program execution begins.
4 int main ()
5 {
6 cout << "Hello OpenFOAM"; //prints Hello OpenFOAM
7 return 0;
8 }
```

C++ code

Code.H

Code.C

$$\sigma = \begin{pmatrix} 10^6 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^6 \end{pmatrix}$$

The standard declarations → namespace **std**

OpenFOAM declarations belong to namespace **Foam**

Explicit naming in OpenFOAM:

**Foam::function()** ;

where **function()** is a function defined in namespace Foam.

```
1 dimensionedTensor sigma
2 ( "sigma",
3 dimensionSet(1, -1, -2, 0, 0, 0, 0),
4 tensor(1e6,0,0,0,1e6,0,0,0,1e6)
5 );
```

Units are defined using the **dimensionSet** class **tensor**

**dimensionSet (k, m, s, K, mol, A, cd)**

Object orientation focuses on the **objects** instead of the functions.

- **Space and time:** `polyMesh`, `fvMesh`, `Time`
- **Field algebra:** `Field`, `DimensionedField` and `GeometricField`
- **Boundary conditions:** `fvPatchField` and derived classes
- **Finite Volume discretisation:** `fv` and `fvm` namespace
- **Sparse matrices:** `lduMatrix`, `fvMatrix` and linear solvers

# objectRegistry in OpenFOAM

The top level `objectRegistry` associates with the `Time` class.

How would a dictionary like `transportProperties` be read into OpenFOAM?

```
IOdictionary transportProperties
(
    IOobject
    (
        "transportProperties",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    )
);
```

`runtime.timeName()`

`runtime.system()`

```
volScalarField p
(
    IOobject
    (
        "p",
        runTime.timeName(), //directory name
        mesh,
        IOobject::MUST_READ, //read controls
        IOobject::AUTO_WRITE //write controls
    ),
    mesh
);
```

- `transportProperties` is the name of the file containing the dictionary.
- `runTime.constant()`, the instance, gives the position of the dictionary, which is, in this case, contained in the constant directory of the considered case.
- The `objectRegistry` is represented by the mesh.

# Finite Volume discretisation

Finite Volume Method implemented in 3 parts

- **Surface interpolation**: cell-to-face data transfer
  - Finite Volume Calculus (fvc): given a field, create a new field
  - Finite Volume Method (fvm): create a matrix representation of an operator, using FV discretisation
- 
- **Explicit** → Evaluate derivative based on known **GeometricField** values – functions grouped into namespace fvc ::
  - **Implicit** → Evaluate derivative based on unknown values. This creates a matrix equation

# Equation Discretization in OpenFOAM

geometricField

<https://cpp.openfoam.org/>

**fvc::** (Finite Volume Calculus) → **Explicit**

**fvm::** (Finite Volume Method) → **Implicit**

fvMatrix

fvc:: ddt (A)

fvm:: ddt (ρ, A)

fvc:: div (A)

fvm:: div (phi, A)

fvc:: laplacian (A)

fvm:: laplacian (mu, A)

fvc:: grad (A)

PDE

**Ax = b**

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \phi \mathbf{U} - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p \longrightarrow$$

```
solve
(
    fvm::ddt(rho,U)
  + fvm::div(phi,U)
  - fvm::laplacian(mu,U)
  ==
  - fvc::grad(p)
);
```



# Sparse matrices and Linear Solvers

- Finite Volume matrix class: **fvMatrix**
- Derived from **lduMatrix**, with a reference to the solution field
- PCG - Preconditioned conjugate gradient solver for symmetric lduMatrices.
- PBiCG - Preconditioned bi-conjugate gradient solver for asymmetric lduMatrices.
- GAMG - Geometric agglomerated algebraic multigrid solver (also named Generalised geometric-algebraic multi-grid in the manual).
- smoothSolver - Iterative solver using smoother for symmetric and asymmetric matrices.
- diagonalSolver - Iterative solver for explicit systems.

**lduMatrix** is a general matrix class in which the coefficients are stored as three arrays, one for the upper triangle, one for the lower triangle and a third for the diagonal.

# Pre-conditioners for Linear Solvers

A preconditioned iterative solver solves the system

$$M^{-1}Ax = M^{-1}b$$



**Left preconditioning**

- The chosen preconditioner should make sure that convergence for the preconditioned system is much faster than for the original one.
- This leads to  $M$  (mostly) being an easily invertible approximation to  $A$ .
- Preconditioner leads to a faster propagation of information through the computational mesh.

Preconditioner	Keyword
Diagonal incomplete-Cholesky (symmetric)	DIC
Faster diagonal incomplete-Cholesky (DIC with caching)	FDIC
Diagonal incomplete-LU (asymmetric)	DILU
Diagonal	diagonal
Geometric-algebraic multi-grid	GAMG
No preconditioning	none

# Source Code of icoFoam

```
cbose1991@DESKTOP-Q10GJQE: /opt/openfoam7/applications/solvers/incompressible/icoFoam
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/platforms/linux64GccDPInt32Opt/bin$ reset
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/platforms/linux64GccDPInt32Opt/bin$ app
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications$ pwd
/opt/openfoam7/applications
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications$ ls
Allwmake solvers test utilities
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications$ cd solvers/
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications/solvers$ ls
DNS      combustion      discreteMethods  financial      incompressible  multiphase
basic    compressible     electromagnetics heatTransfer    lagrangian      stressAnalysis
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications/solvers$ cd incompressible/
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications/solvers/incompressible$ ls
adjointShapeOptimizationFoam  icoFoam          pim
boundaryFoam                  nonNewtonianIcoFoam  pis
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications/
oam/
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications/
ls
Make createFields.H icoFoam.C
cbose1991@DESKTOP-Q10GJQE:/opt/openfoam7/applications/
```

```
cbose1991@DESKTOP-Q10GJQE: ~/OpenFOAM/OpenFOAM...
icoFoam/
├── Make
│   ├── files
│   └── options
├── createFields.H
└── icoFoam.C
```

# Source Code Compilation

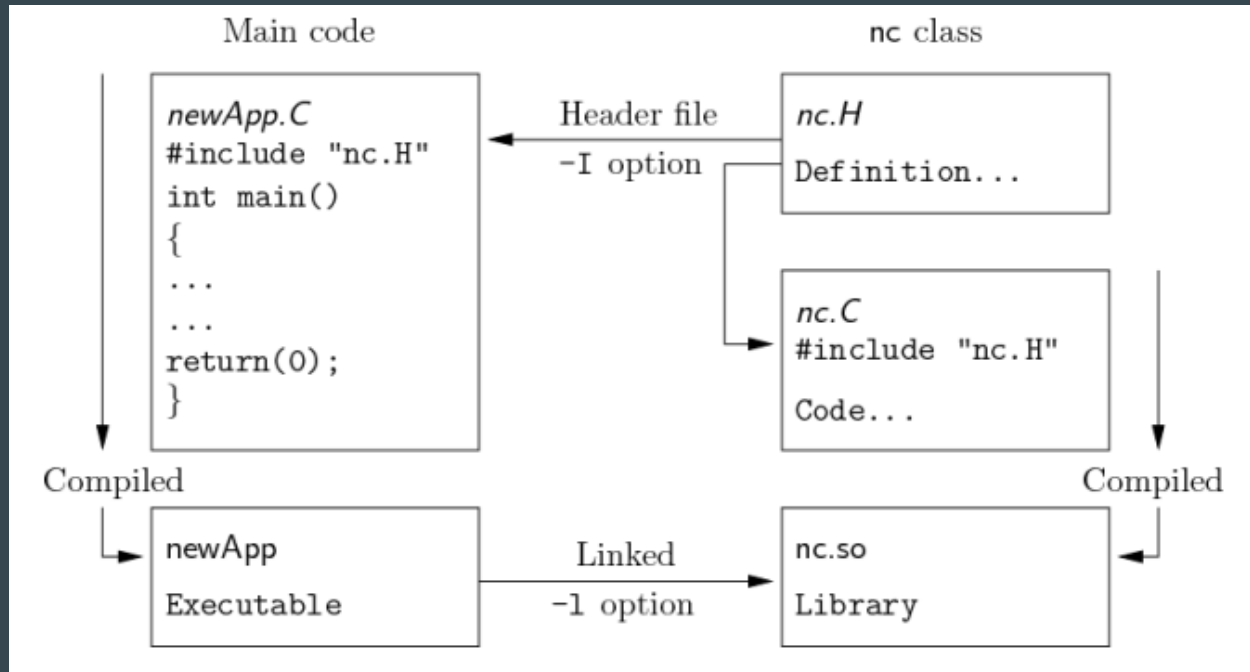
wmake

`$FOAM_APPBIN`

`$FOAM_USER_APPBIN`

`$FOAM_LIBBIN`

`$FOAM_USER_LIBBIN`



And that's it.

Hope you liked it. 😊