



Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

Programming in OpenFOAM

Lecture 7

Balogh Miklós

March 23, 2015



Table of Contents

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

① Basic concept

② Programming

③ Variables

④ Loops



Difference between commercial and open-source

Programming

Balogh
Miklós

Basic concept

Programming

Variables

Loops

- Commercial solvers (Fluent, CFX, StarCCM++)
 - Black-box (way of implementation is not known)
 - UDF - User Defined Functions (ANSI C, Fortran)
 - Modification is limited
- OpenFOAM
 - The source code is accessible
 - Direct implementation is possible
 - No limitation (limited by common sense)



Applications and libraries

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

- Applications - standalone objects
 - Solvers - icoFoam, simpleFoam, etc.
 - Utilities - blockMesh, setFields, etc.
- Libraries
 - Turbulence models
 - Specific inlet/outlet boundary conditions
 - Specific wall functions
 - Additional source terms



Programming language of OpenFOAM

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

- OpenFOAM is a C++ library (toolbox)
 - Object-oriented, such as C++
 - Provide the mechanism (classes) to declare types and associated operations
 - In verbal and mathematical languages used in science and engineering
- Interpretation
 - Physics: Velocity field and magnitude
 - Math: U and $|U|$
 - OpenFOAM: `volVectorField U` and `volScalarField Umag = mag(U)`



Equation representation in OpenFOAM

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

$$\text{Example: } \frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p$$

Listing 1: solver code

```
1 solve
2 (
3     fvm::ddt(rho, U)
4     + fvm::div(phi, U)
5     - fvm::laplacian(mu, U)
6     ==
7     - fvc::grad(p)
8 );
```



Variables

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

- Local single variables (e.g. in a loop)
 - scalar
 - vector
 - tensor
- Dimensioned single variables
 - dimensionedScalar
 - dimensionedVector
 - dimensionedTensor
- Dimensioned field variables
 - volScalarField (scalarField)
 - volvectorField (vectorField)
 - volTensorField (tensorField)

Listing 2: single variables

```
1 dimensionedScalar nu(transportProperties.lookup("nu"));
2 dimensionedVector flowDir(transportProperties.lookup("flowDir"));
```



Variables - volScalarField

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

Listing 3: volScalarField

```
1 Info<< "Reading field p\n" << endl;  
2 volScalarField p  
3 (  
4     IObject  
5     (  
6         "p",  
7         runtime.timeName(),  
8         mesh,  
9         IObject::MUST_READ,  
10        IObject::AUTO_WRITE  
11    ),  
12    mesh  
13 );
```




Variables - volVectorField

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

Listing 4: volVectorField

```
1 Info<< "Reading field U\n" << endl;  
2 volVectorField U  
3 (  
4     IObject  
5     (  
6         "U",  
7         runTime.timeName(),  
8         mesh,  
9         IObject::MUST_READ,  
10        IObject::AUTO_WRITE  
11    ),  
12    mesh  
13 );
```



Variables - volTensorField

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

Listing 5: volTensorField

```
1 Info<< "Reading field Permeability\n" << endl;  
2 volTensorField Permeability  
3 (  
4     IObject  
5     (  
6         "Permeability",  
7         runTime.timeName(),  
8         mesh,  
9         IObject::MUST_READ,  
10        IObject::AUTO_WRITE  
11    ),  
12    mesh  
13 );
```



Loops and cycles over cells

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

Listing 6: loop over cells

```
1  volScalarField centres_ = mag(wallNorm_ & mesh_.C());
2
3  // Limitation of centres to the ABL height in the cells
4  forAll(centres_, cellI)
5  {
6      if(centres_[cellI] > hABL_.value())
7      {
8          centres_[cellI] = hABL_.value();
9      }
10 }
11
12 // Similar to the following
13 volScalarField centres_ = min(mag(wallNorm_ & mesh_.C()), hABL_);
```



Loops and cycles over faces

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

Listing 7: loop over faces of patches

```
1 // Limitation of centres to the ABL height at the boundaries
2 forAll(patches, patchI)
3 {
4     const fvPatch& curPatch = patches[patchI];
5     forAll(curPatch, faceI)
6     {
7         vector cf = mesh_.Cf().boundaryField()[patchI][faceI];
8         scalar ctemp = mag(wallNorm_.value() & cf);
9         if(ctemp > hABL_.value())
10        {
11            centres_.boundaryField()[patchI][faceI] = hABL_.value();
12        }
13        else
14        {
15            centres_.boundaryField()[patchI][faceI] = ctemp;
16        }
17    }
18 }
```



Compile OpenFOAM sources

Programming

Balogh
Miklós

Basic
concept

Programming

Variables

Loops

Listing 8: compilation

```
1      # Compilation of apps. and utilities
2      wclean
3      wmake
4
5      # Compilation of libraries, such as
6      # - turbulence models
7      # - wall functions
8      # - boundary conditions
9      wclean
10     wmake libso
```