

## MTE 204 LECTURE 2: Open Source Field Operation and Manipulation (OpenFOAM)

### 2.0 INTRODUCTION

OpenFOAM which stands for Open Source Field Operation and Manipulation it is a C++ library used basically to create **executables** known as **applications**. The applications are in two categories:

**Solvers:** these are designed to solve distinct problems in continuum mechanics.

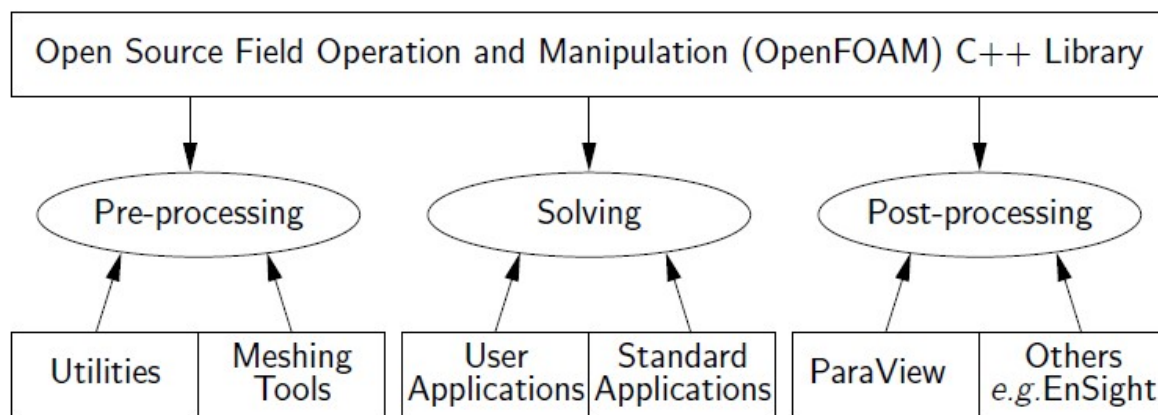
**Utilities:** these are designed to perform tasks that involve data manipulation.

OpenFOAM contains several solvers and utilities which can be used to solve several problems.

One significant advantage of OpenFOAM over most CFD packages is that new solvers and utilities can be created by its users who are equipped with knowledge of the underlying method, physics and programming techniques needed.

OpenFOAM has the pre- and post-processing environments with the OpenFOAM utilities being the interface. These ensure consistent data handling across all environments.

The overall structure of OpenFOAM is shown in figure 2.0.



**Figure 2.0:** OpenFOAM structure

### 2.1 PROGRAMMING IN OpenFOAM

- i. OpenFOAM represents scalars, vectors and matrices as tensor fields. A zero rank tensor is a scalar, a first rank tensor is a vector and a second rank tensor is a matrix.
- ii. The partial differential equations (PDEs) solved involve derivatives of tensor fields with respect to time and space.
- iii. These PDEs must be discretized in time and space before they can be solved them.

## 2.2 BASIC TENSOR CLASSES IN OPENFOAM®

- OpenFOAM® contains a C++ class library named **primitive**.
- In the table below, the basic tensor classes available in OpenFOAM®, with their respective access functions are shown.

Tensor Rank	Common name	Basic class	Access function
0	Scalar	<code>scalar</code>	
1	Vector	<code>vector</code>	<code>x()</code> , <code>y()</code> , <code>z()</code>
2	Tensor	<code>tensor</code>	<code>xx()</code> , <code>xy()</code> , <code>xz()</code> ...

In OpenFOAM®, the tensor shown below

$$\mathbf{T} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Is declared thus:-

```
tensor T(1, 2, 3, 4, 5, 6, 7, 8, 9);
```

The component  $T_{13}$  or  $T_{xz}$  can be accessed using the `xz()` access function.

For instance, in OpenFOAM, the following code line:

```
Info << "Txz = " << T.xz() << endl;
```

Will output:

```
Txz = 3
```

And as opposed to the function “cout” used in C++, “Info” function is used.

## 2.3 ALGEBRAIC TENSOR OPERATIONS IN OPENFOAM®

- Tensor operations operate on the entire tensor entity instead of a series of operations on its components.

- The OpenFOAM® syntax closely mimics the syntax used in written mathematics, using descriptive functions or symbolic operators.
- Some of the algebraic tensor operations are listed in the following table.

Operation	Remarks	Mathematical description	OpenFOAM® description
Addition		$a + b$	<code>a + b</code>
Scalar multiplication		$sa$	<code>s * a</code>
Outer product	rank a, b $\geq 1$	$ab$	<code>a * b</code>
Inner product	rank a, b $\geq 1$	$a \cdot b$	<code>a &amp; b</code>
Double inner product	rank a, b $\geq 2$	$a : b$	<code>a &amp;&amp; b</code>

## 2.4 DIMENSIONAL UNITS IN OPENFOAM®

OpenFOAM® is fully dimensional and for this reason, the following occur:-

- Dimensional checking implementation as a safeguard against implementing a meaningless operation.
- OpenFOAM® encourages the user to attach dimensional units to any tensor and it will perform dimension checking of any tensor operation.
- Units are defined using the **dimensionSet** class tensor, with its units defined using the **dimensioned<Type>** template class, the “<Type>” being scalar, vector, tensor, etc. The **dimensioned<Type>** stores the variable name, the dimensions and the tensor values.

Using the example below, a tensor with dimensions is declared in the following way:

```
dimensionedTensor sigma
(
    "sigma",
    dimensionSet(1, -1, -2, 0, 0, 0, 0),
    tensor(10e6, 0, 0, 0, 10e6, 0, 0, 0, 10e6)
);
```

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

## 2.5 UNITS CORRESPONDENCE IN DIMENSIONSET

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

`dimensionSet (kg, m, s, K, mol, A, cd)`

Parameters with unit  $\text{kg}/\text{ms}^{-2}$  is declared in OpenFOAM as:

```
dimensionedTensor sigma
(
    "sigma",
    dimensionSet(1, -1, -2, 0, 0, 0, 0),  $\equiv \sigma = \begin{pmatrix} 10^6 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^6 \end{pmatrix}$ 
    tensor(1e6,0,0,0,1e6,0,0,0,1e6)
);
```

`dimensionSet (kg, m, s, K, mol, A, cd)`

Compile and run the program below in your created directory and note the output:

Before the main function `main ( )`:

```
#include "dimensionedTensor.H"
```

Before `return(0):`

```
dimensionedTensor sigma
(
    "sigma",
    dimensionSet(1, -1, -2, 0, 0, 0, 0),
    tensor(1e6,0,0,0,1e6,0,0,0,1e6)
);
Info<< "Sigma: " << sigma << endl;
```

This will be your output:

```
Sigma sigma [1 -1 -2 0 0 0 0] (1e+06 0 0 0 1e+06 0 0 0 1e+06)
```

It can be observed that the object `sigma`, which belongs to the `dimensionedTensor` class, contains the name, the dimensions and values.

## 2.6 OpenFOAM® LISTS AND FIELDS

- i. OpenFOAM® frequently needs to store sets of data and perform mathematical operations.
- ii. OpenFOAM® therefore provides an array template class **List<Type>**, making it possible to create a list of any object of class **Type** that inherits the functions of the **Type**. For example a **List** of **vector** is **List<vector>**.
- iii. Lists of the tensor classes are defined in OpenFOAM® by the template class **Field<Type>**. For better code legibility, all instances of **Field<Type>**, e.g. **Field<vector>**, are renamed using typedef declarations as **scalarField**, **vectorField**, **tensorField**, **symmTensorField**, **tensorThirdField** and **symmTensorThirdField**.
- iv. Algebraic operations can be performed between **Field** subject to obvious restrictions such as the fields having the same number of elements.
- v. OpenFOAM® also supports operations between a field and a zero rank tensor, e.g. all values of a **Field** **U** can be multiplied by the **scalar** 2 by simple coding the following line, **U = 2.0 \* U**.