

**Mitja Morgut, Enrico Nobile, Paola Ranut**  
*DIA - Dipartimento di Ingegneria e Architettura*  
*Università degli Studi di Trieste*

*Esercitazione di Termofluidodinamica Computazionale*

## **Introduzione a OpenFOAM**

**(B) Calcolo del flusso turbolento in un gomito a 90°**



Aprile 2013



## Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Operazioni preliminari</b>	<b>4</b>
2.1	Export della mesh in formato Fluent . . . . .	4
2.2	Trasferimento della mesh . . . . .	6
<b>3</b>	<b>Setup della simulazione</b>	<b>7</b>
3.1	Conversione della mesh . . . . .	7
3.2	Definizione delle proprietà fisiche del fluido di lavoro . . . . .	8
3.3	Scelta del modello di turbolenza . . . . .	9
3.4	Condizioni al contorno . . . . .	10
3.5	Solver setup . . . . .	16
<b>4</b>	<b>Esecuzione della simulazione</b>	<b>19</b>
4.1	Calcolo seriale . . . . .	19
4.2	Calcolo parallelo . . . . .	19
<b>5</b>	<b>Post processing</b>	<b>20</b>
5.1	Perdita di carico . . . . .	20
5.2	Contour del campo di velocità . . . . .	24



## 1 Introduzione

In questa esercitazione viene ripetuta, in OpenFOAM, l'analisi del flusso turbolento in un gomito a 90°, eseguita in precedenza con ANSYS CFX durante l'esercitazione *Calcolo del flusso turbolento in un gomito a 90° con ANSYS CFX*.

Come già accennato nella prima parte di questa esercitazione, *Introduzione a OpenFOAM - (A) Installazione di OpenFOAM su macchine Windows tramite VirtualBox*, OpenFOAM è un OpenSource CFD toolbox mediante il quale è possibile, utilizzando uno dei molteplici solutori pre-compilati, risolvere/analizzare innumerevoli casi/problemi termofluidodinamici.

In questo caso sarà utilizzato il solutore `simpleFoam`, sviluppato per la previsione di flussi turbolenti stazionari. La simulazione sarà eseguita utilizzando la mesh strutturata generata durante l'esercitazione *Utilizzo del Generatore di Mesh ANSYS-ICEM CFD - (B) Flusso in un gomito a 90°*. Le operazioni di post-processing saranno eseguite con il programma, anch'esso Open-Source, *paraview*.

Questa esercitazione mira ad insegnare allo studente le nozioni base di OpenFOAM per poter analizzare problemi/casi di flussi turbolenti stazionari.

Chi volesse estendere le proprie conoscenze è invitato a consultare la guida ufficiale del programma, (vedi Desktop macchina virtuale) e visionare i molteplici tutorials contenuti nella directory:

```
student@ubuntu:/opt/openfoam220/tutorials
```

Utile può essere la consultazione del forum on-line:

<http://www.cfd-online.com/Forums/openfoam/>

## 2 Operazioni preliminari

Nel contesto di questa esercitazione prima di passare al setup delle simulazioni dovete operare le seguenti operazioni.

1. Esportare, in formato Fluent (.msh), la mesh generata durante l'esercitazione *Utilizzo del Generatore di Mesh ANSYS-ICEM CFD - (B) Flusso in un gomito a 90°*.
2. Trasferire la mesh nella cartella /Esercitazione/MyGomito della macchina virtuale

### 2.1 Export della mesh in formato Fluent

In questo caso dovete esportare la mesh in formato Fluent, poichè nella fase successiva essa sarà convertita dal formato Fluent al formato di OpenFOAM tramite l'esecuzione di un apposito comando base.

Per esportare la mesh generata in precedenza con ICEM operate nel seguente modo:

1. Dal Tab **Output**, selezionate l'icona **Select Solver** cerchiata di Fig. 1.  
Impostate **Output Solver** → **Fluent\_V6**.  
(La scelta del **Common Structural Solver** è superflua).

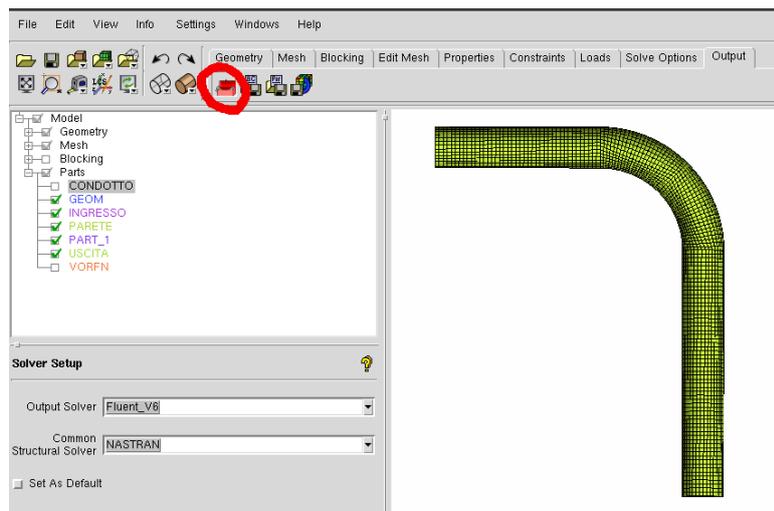


Figura 1: Selezione formato

2. Schiacciate sull'icona **Write Input** cerchiata di Fig. 2.
  - 2.1 Il meshatore vi chiederà se desiderate salvare il progetto. Scegliete **YES**
  - 2.2 Alla comparsa della finestrella analoga a quella di Fig. 3 ,schiacciate su **Open**.
  - 2.3 All'apparire dell'ultima finestrella utile per il setup dell'export assicuratevi di **scalare la mesh** del fattore 0.001 in tutte le tre direzioni come indicato in Fig. 4.  
*In questo caso è necessario scalare da subito la mesh poichè in OpenFOAM le dimensioni sono indicate sempre in metri*
  - 2.4 Settate il nome del file di output e cliccate su **Done**

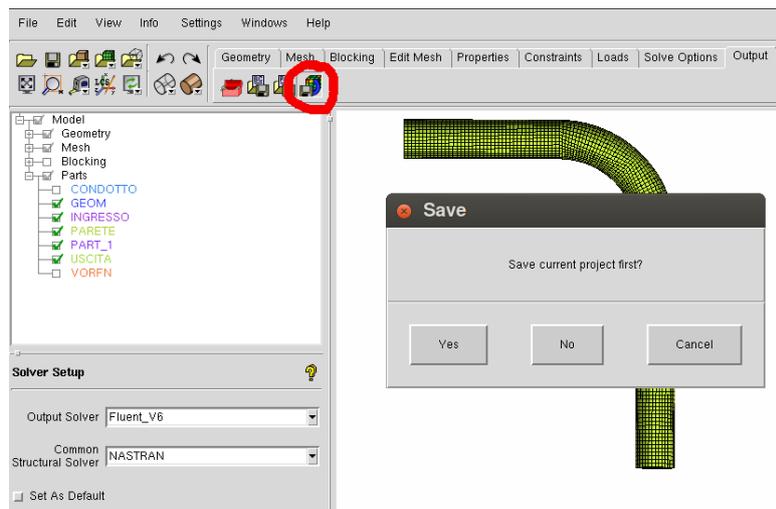


Figura 2: Salvataggio mesh

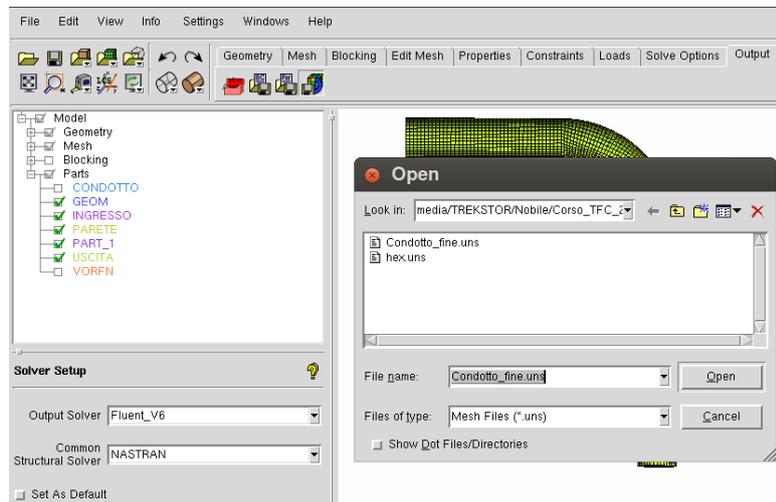


Figura 3: Apertura mesh export

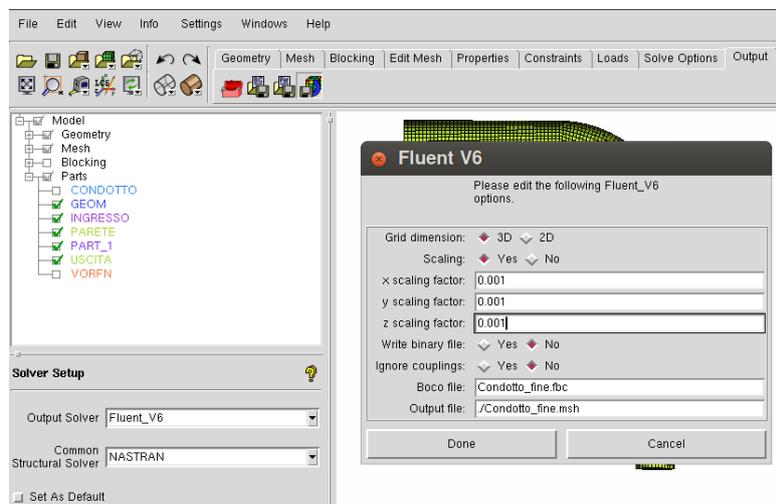


Figura 4: Setup del export

## 2.2 Trasferimento della mesh

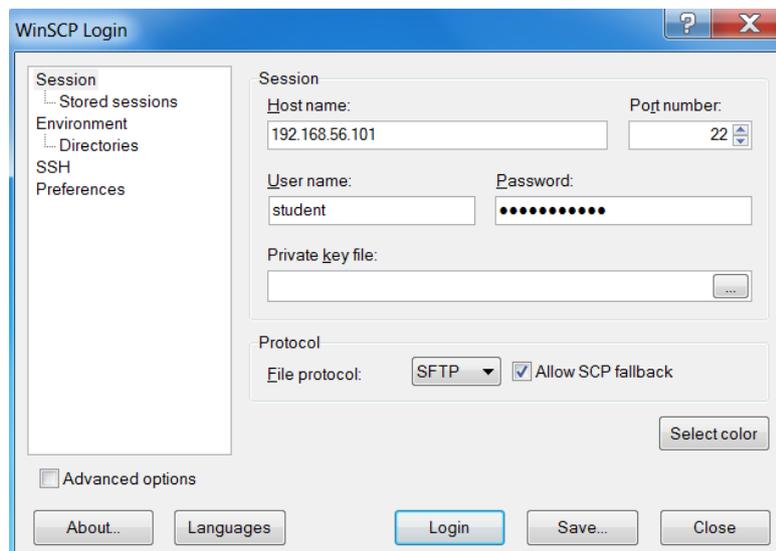
1. Avviate da *VirtualBox* la macchina virtuale *ubuntu*.
2. Alla comparsa della finestra di login, la quale evidenzia il corretto funzionamento di *ubuntu* (virtuale), avviate in Windows il programma *winSCP*.
3. Con *winSCP* Trasferite il file della vostra mesh, in formato *.msh*, da Windows a Ubuntu nella directory `Esercitazione/MyGomito`.

Per completezza in Fig. 5 viene nuovamente proposto il setup per potersi connettere alla macchina virtuale dove:

**Host name:** 192.168.56.101

**User name:** student

**Password:** student2013



**Figura 5:** Setup di winSCP

## 3 Setup della simulazione

### 3.1 Conversione della mesh

Per convertire la mesh dal formato Fluent (.msh) al formato di OpenFOAM operate nel seguente modo:

1. Aprite una finestra terminale cliccando su 

**1.1:** Nella console digitate in sequenza:

```
cd Esercitazione
cd MyGomito
```

**1.2:** Una volta che vi siete posizionati in:

```
student@ubuntu~/Esercitazioni/MyGomito$
per procedere alla conversione della mesh eseguite l'istruzione:
```

```
fluent3DMeshToFoam nome_della_vostra_mesh.msh
```

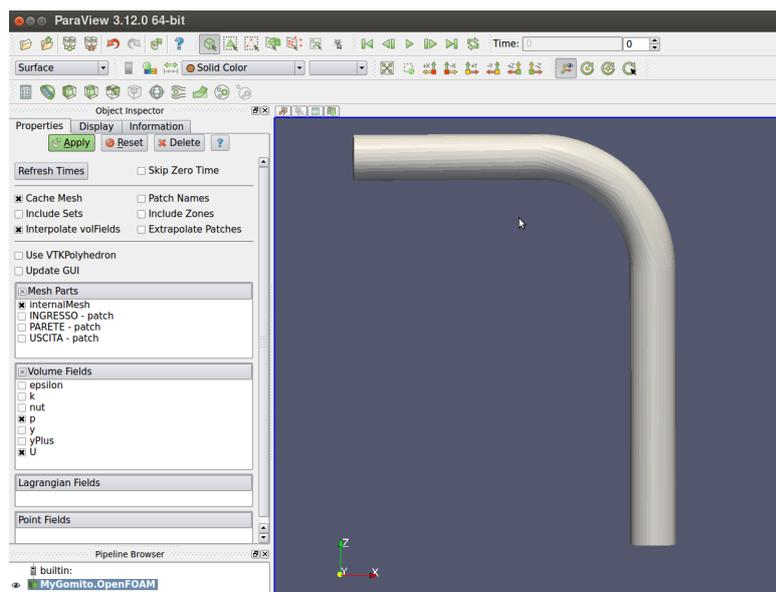
*Si precisa che a seguito della conversione i dati della mesh, in formato OpenFOAM, sono stati salvati/scritti nella cartella `../MyGomito/constant/polyMesh`*

Per verificare se la mesh è stata importata in maniera corretta può essere utilizzato il programma *paraview*, il quale sarà successivamente utilizzato anche per le operazioni di post-processing.

1. Per caricare il caso in *paraview* digitate `paraFoam`

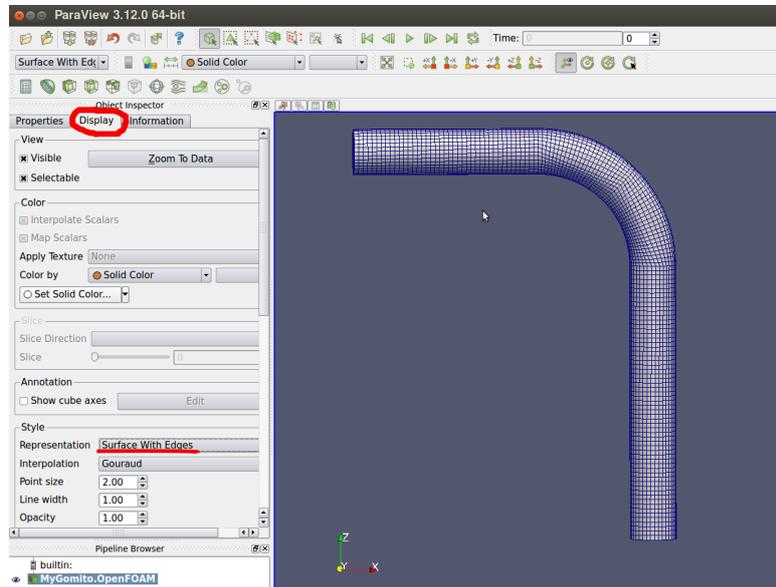
```
student@ubuntu:~/Esercitazioni/MyGomito$ paraFoam
```

**2.1:** Aperto il programma, schiacciate su **Apply**  
*Dovreste ottenere il risultato di Fig. 6.*



**Figura 6:** Visualizzazione solida del modello

**2.2:** Per visualizzare la mesh di superficie selezionate il tab **Display** cerchiato in Fig. 7 e poi dal menu a tendina di **Style** → **Representation** selezionate **Surface with edges**.



**Figura 7:** Visualizzazione mesh di superficie

### 3.2 Definizione delle proprietà fisiche del fluido di lavoro

In questo caso basta definire il valore della viscosità cinematica dell'acqua  $\nu = 8.92 \times 10^{-7} \text{ m}^2/\text{s}$ . Procedete nel seguente modo:

1. Nella console digitate:

```
cd constant
```

2. Visualizzate il contenuto della directory `../MyGomito/constant` digitando:

```
ls
```

*Dovreste notare il file `transportProperties`.*

3. Aprite il file `transportProperties` digitando:

```
gedit transportproperties
```

*Dovreste ottenere il risultato analogo a quello di Fig. 8. Poichè per questa esercitazione il valore della viscosità è già stato impostato al valore corretto chiudete `gedit` e proseguite con il `setup`.*

```

/*-----* C++ *-----*/
=====
Field      | OpenFOAM: The Open Source CFD Toolbox
Operation  | Version: 2.2.0
And        | Web: www.OpenFOAM.org
Manipulation

FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       transportProperties;
}
// *****

transportModel Newtonian;
nu              nu [ 0 2 -1 0 0 0 ] 8.92e-07;
// *****

```

Figura 8: Snapshot del file transportProperties letto con gedit

### 3.3 Scelta del modello di turbolenza

Il modello di turbolenza viene definito nel file `RASProperties` che si trova sempre nella directory `../MyGomito/constant`.

Analogamente a quanto fatto in precedenza per aprire/editare il file digitate:

```
gedit RASproperties
```

Per completezza in Fig. 9 è presentato lo snapshot di questo file visualizzato con gedit.

```

/*-----* C++ *-----*/
=====
Field      | OpenFOAM: The Open Source CFD Toolbox
Operation  | Version: 2.2.0
And        | Web: www.OpenFOAM.org
Manipulation

FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       RASProperties;
}
// *****

RASModel      kEpsilon;
turbulence    on;
printCoeffs   on;

```

Figura 9: Snapshot del file RASProperties letto con gedit

*Anche in questo caso non serve modificare il file poichè le simulazioni saranno eseguite proprio con il modello di turbolenza  $k - \epsilon$ .*

### 3.4 Condizioni al contorno

Per definire le condizioni al contorno dovete opportunamente editare:

1. il file `boundary` che si trova nella directory `../MyGomito/constant/polyMesh`
2. i file `U`, `p`, `k`, `epsilon`, `nut` che si trovano in `../MyGomito/0`

1. Per modificare il file `boundary` in questo caso procedete nel seguente modo:

**1.1:** Nella console digitate:

```
cd polyMesh
gedit boundary
```

**1.2:** Per la parte PARETE sostituite: `type patch` con `type wall`.

Non modificate le altre parti del file. Le eventuali differenze nel numero di `nFaces` e `startFace` del vostro file con quello presentato nel Listing 1 sono dovute ad un diverso grado di raffinamento della mesh.

**1.3:** Chiudete `gedit`

**Listing 1:** `../MyGomito/constant/polyMesh/boundary`

```
FoamFile
{
  version 2.0;
  format ascii;
  class polyBoundaryMesh;
  location "constant/polyMesh";
  object boundary;
}

3
(
  INGRESSO
  {
    type patch;
    nFaces 225;
    startFace 97011;
  }
  PARETE
  {
    type wall;
    nFaces 5328;
    startFace 97236;
  }
  USCITA
  {
    type patch;
    nFaces 225;
    startFace 102564;
  }
)
```

2. In questo caso per motivi didattici i file `U`, `p`, `epsilon`, `k`, `nut` sono già stati settati in maniera corretta. Nelle pagine a seguire sono presentati i listati dei file.

Per completezza/chiarzza, soprattutto per gli utenti meno esperti di Linux, si precisa che per spostarsi (utilizzando la console) dalla directory `../MyGomito/constant/polyMesh` alla directory `../MyGomito/0` in questo caso bisogna digitare:

```
cd ..
cd ..
cd 0
```

Inoltre, per visualizzare ad. es. il file `U` basta digitare come fatto finora:

```
gedit U
```

**Listing 2:** `../MyGomito/0/U`

```
FoamFile
{
    version 2.0;
    format ascii;
    class volVectorField;
    object U;
}

dimensions [0 1 -1 0 0 0 0];

internalField uniform (0 0 0);

boundaryField
{
    INGRESSO
    {
        type fixedValue;
        value uniform (0.2 0 0);
    }

    USCITA
    {
        type zeroGradient;
    }

    PARETE
    {
        type fixedValue;
        value uniform (0 0 0);
    }
}
```

Listing 3: ../MyGomito/0/p

```

FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  object p;
}

dimensions [0 2 -2 0 0 0 0];

internalField uniform 0;

boundaryField
{
  INGRESSO
  {
    type zeroGradient;
  }

  USCITA
  {
    type fixedValue;
    value uniform 0;
  }

  PARETE
  {
    type zeroGradient;
  }
}

```

**NOTA**

Prestando attenzione all' unità di misura:

```
dimensions [0 2 -2 0 0 0 0];
```

potete notare che la pressione in questo caso è espressa in  $m^2/s^2$ .

Questo è dovuto al fatto che le simulazioni sono eseguite considerando un flusso incomprimibile. Per tali condizioni la densità rimane costante. Quindi le equazione di continuità e di quantità di moto (per convenienza) possono essere divise con la densità  $\rho$ .

Nel caso dell'equazione della quantità di moto si ha:

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot [\mu (\nabla \mathbf{U} + (\nabla \mathbf{U})^T)] \quad (1)$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) = -\nabla \frac{p}{\rho} + \nabla \cdot [\nu (\nabla \mathbf{U} + (\nabla \mathbf{U})^T)] \quad (2)$$

**Listing 4:** ../MyGomito/0/k

```
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  location "0";
  object k;
}

dimensions [0 2 -2 0 0 0 0];

internalField uniform 0.0006;

boundaryField
{
  INGRESSO
  {
    type fixedValue;
    value uniform 0.0006;
  }
  USCITA
  {
    type zeroGradient;
  }

  PARETE
  {
    type kqRWallFunction;
    value uniform 0.0006;
  }
}
```

**Listing 5:** ../MyGomito/0/epsilon

```
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  location "0";
  object epsilon;
}

dimensions [0 2 -3 0 0 0 0];

internalField uniform 0.036;

boundaryField
{
  INGRESSO
  {
    type fixedValue;
    value uniform 0.036;
  }
  USCITA
  {
    type zeroGradient;
  }
  PARETE
  {
    type epsilonWallFunction;
    value uniform 0.036;
  }
}
```

**Listing 6:** ../MyGomito/0/nut

```
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  location "0";
  object nut;
}

dimensions [0 2 -1 0 0 0 0];

internalField uniform 0;

boundaryField
{
  INGRESSO
  {
    type calculated;
    value uniform 0;
  }
  USCITA
  {
    type calculated;
    value uniform 0;
  }
  PARETE
  {
    type nutkWallFunction;
    value uniform 0;
  }
}
```

### 3.5 Solver setup

Il setup del solutore viene definito attraverso i file `controlDict`, `fvSchemes` e `fvSolution` contenuti nella directory `../MyGomito/system`.

Si fa notare che:

in `fvSchemes` vengono settati gli schemi di interpolazione

in `fvSolution` vengono specificati e settati i solutori di equazioni lineari

in `controlDict` viene settato il numero massimo di iterazioni del ciclo SIMPLE (Semi Implicit Multiple Pressure Linked Equations).

Di seguito sono presentati i contenuti dei file sopracitati. Si fa notare che nei file `fvSchemes` e `fvSolution` appare la variabile `omega` ( $\omega$ ) relativa ai modelli di turbolenza della classe  $k - \omega$  anche se le simulazioni sono in questo caso eseguite esclusivamente con il modello di turbolenza  $k - \epsilon$ . Questo è stato fatto per darvi la possibilità di eseguire, in un secondo momento, anche le simulazioni con il modello di turbolenza SST semplicemente sostituendo nel file `../constant/transportProperties` la stringa `kEpsilon` con la stringa `kOmegaSST` e definendo in `../MyGomito/0` la grandezza `omega`.

**Listing 7:** `../MyGomito/system/controlDict`

```
application simpleFoam;

startFrom latestTime;

startTime 0;

stopAt endTime;

endTime 1000;

deltaT 1;

writeControl timeStep;

writeInterval 500;

purgeWrite 0;

writeFormat ascii;

writePrecision 6;

writeCompression uncompressed;

timeFormat general;

timePrecision 6;

runTimeModifiable yes;
```

**Listing 8:** MyGomito/system/fvSchemes

```
ddtSchemes
{
    default steadyState;
}

gradSchemes
{
    default Gauss linear ;
    grad(p) Gauss linear ;
    grad(U) Gauss linear ;
}

divSchemes
{
    default none;
    div(phi,U) Gauss linearUpwind grad(U);
    div(phi,k) Gauss upwind;
    div(phi,omega) Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div((nuEff*dev(T(grad(U)))) Gauss linear;
}

laplacianSchemes
{
    default none;
    laplacian(nuEff,U) Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
    laplacian(DkEff,k) Gauss linear corrected;
    laplacian(DomegaEff,omega) Gauss linear corrected;
    laplacian(DepsilonEff,epsilon) Gauss linear corrected;
    laplacian(1,p) Gauss linear corrected;
}

interpolationSchemes
{
    default linear;
    interpolate(U) linear;
}

snGradSchemes
{
    default corrected;
}

fluxRequired
{
    default no;
    p;
}
```

**Listing 9:** ../MyGomito/system/fvSolution

```
solvers
{
  p GAMG
  {
    tolerance 1e-12;
    relTol 0.001; // Originale era 0.0001
    smoother GaussSeidel;
    cacheAgglomeration true;
    nCellsInCoarsestLevel 500;
    agglomerator faceAreaPair;
    mergeLevels 1;
  };

  U smoothSolver
  {
    smoother GaussSeidel;
    nSweeps 2;
    tolerance 1e-10;
    relTol 0.01;
  };

  k smoothSolver
  {
    smoother GaussSeidel;
    nSweeps 2;
    tolerance 1e-10;
    relTol 0.01;
  };

  epsilon smoothSolver
  {
    smoother GaussSeidel;
    nSweeps 2;
    tolerance 1e-10;
    relTol 0.01;
  };

  omega smoothSolver
  {
    smoother GaussSeidel;
    nSweeps 2;
    tolerance 1e-10;
    relTol 0.001;
  };
}

SIMPLE
{
  nNonOrthogonalCorrectors 2;
  pRefCell 0;
  pRefValue 0;
}

relaxationFactors
{
  p 0.5;
  U 0.4;
  k 0.4;
  omega 0.4;
  epsilon 0.4;
}
```

## 4 Esecuzione della simulazione

In questo caso le simulazioni vengono eseguite in regime stazionario utilizzando il solutore *simpleFoam*. A seconda della capacità del vostro laptop la simulazione può essere lanciata in modalità seriale o parallela.

### 4.1 Calcolo seriale

Per eseguire la simulazione utilizzando un'unica CPU (core) posizionatevi in `../MyGomito` e poi digitate `simpleFoam`.

```
student@ubuntu:~/Esercitazione/MyGomito$ simpleFoam
```

La simulazione in accordo con quanto settato nel file `../system/controlDict` dovrebbe terminare dopo 1000 iterazioni.

### 4.2 Calcolo parallelo

Per poter eseguire le simulazioni in parallelo dovete

1. Editare il file `../system/decomposeParDict`. Nel caso di due processori si ha:

**Listing 10:** `../MyGomito/system/decomposeParDict`

```
FoamFile
{
  version 2.0;
  format ascii;
  class dictionary;
  location "system";
  object decomposeParDict;
}

numberOfSubdomains 2;

method scotch;

distributed no;

roots ( );
```

2. Eseguire l'istruzione `decomposePar`

```
student@ubuntu:~/Esercitazione/MyGomito$ decomposePar
```

*Verificate la comparsa dell'directory `processor0` e `processor1`.*

3. Digitare:

```
mpirun -np 2 simpleFoam -parallel
```

Anche in questo caso la simulazione dovrebbe terminare dopo 1000 iterazioni.

**Attenzione:** per poter visualizzare i risultati bisogna in questo caso ricomporre il caso. Per fare ciò usate l'istruzione `reconstructPar`.

```
student@ubuntu:~/Esercitazione/MyGomito$ reconstructPar
```

*Eseguito `reconstructPar` dovrebbe apparirvi la directory `1000`.*

## 5 Post processing

In questa sezione viene descritto come effettuare con il programma *paraview*, delle operazioni di post-processing analoghe a quelle eseguite in precedenza con CFX-Post nell'esercitazione: *Calcolo del flusso turbolento in un gomito a 90° con ANSYS-CFX*.

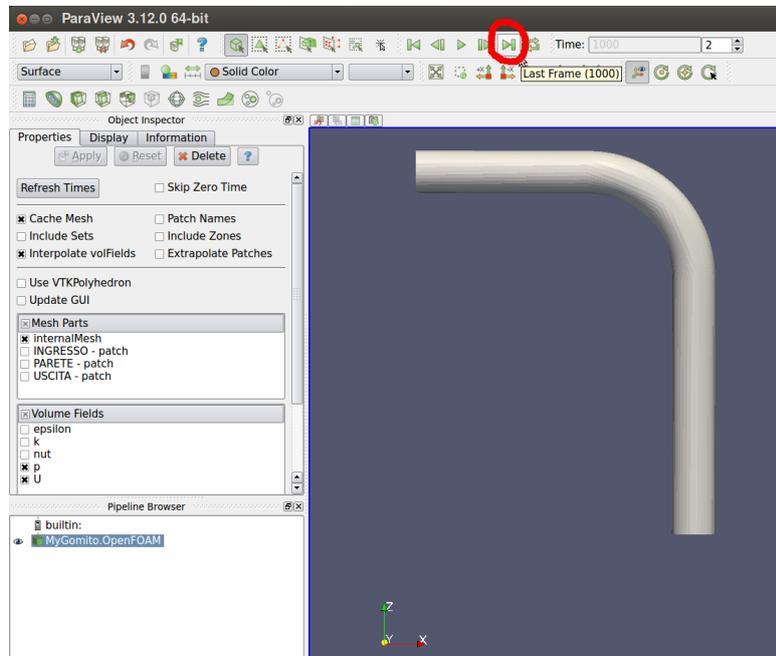
In particolare di seguito viene spiegato come valutare la caduta di pressione dovuta al gomito e come estrarre il contour del campo di velocità in un determinato piano.

### 5.1 Perdita di carico

1. Lanciate *paraview* con il comando `paraFoam`.

```
student@ubuntu:~/Esercitazione/MyGomito$ paraFoam
```

2. Avviato il programma cliccate, prima sull'icona cerchiata di Fig. 10 e poi su **Apply**. In questo modo caricherete l'ultimo timestep disponibile.



**Figura 10:** Caricamento ultimo time-step

3. Per ricavare i dati relativi al *Piano monte* ( $x = 0$  mm) operate nel seguente modo:

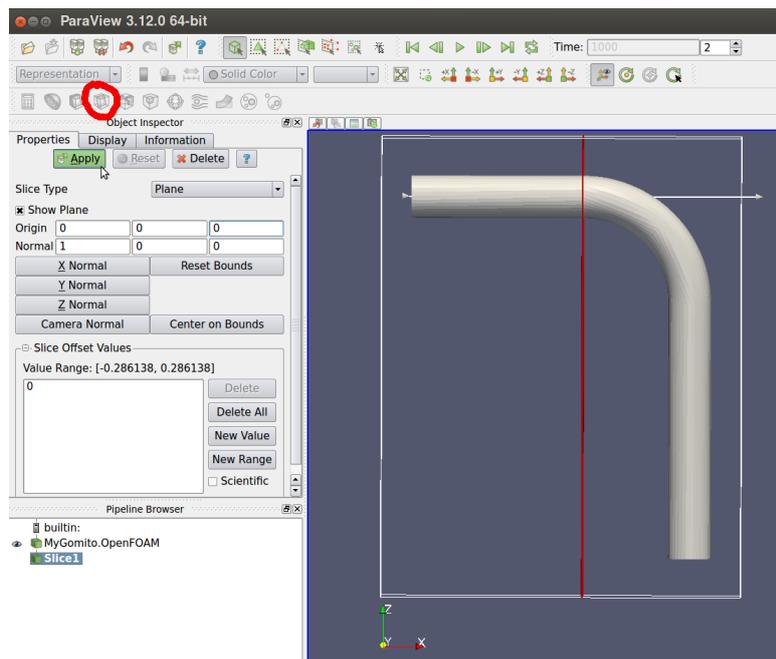
**3.1:** Cliccate sul simbolo cerchiato di Fig.11

**3.2:** Settate i campi **Origin** e **Normal** come indicato in Fig.11

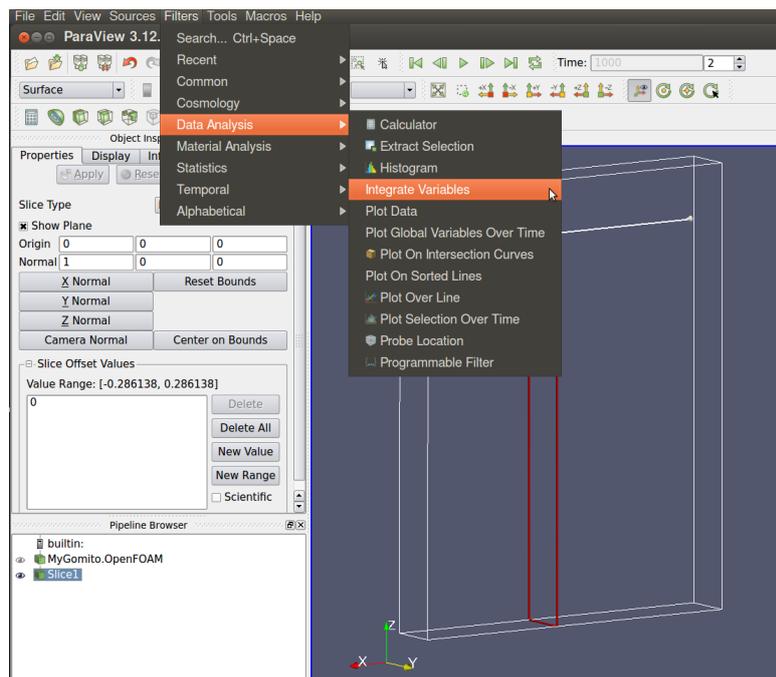
**3.3:** Cliccate su **Apply**

**3.4:** Dal menu ad albero lasciate attiva solo la visualizzazione dello **Slice1**.

**3.5:** spostate il cursore del mouse sulle barra superiore del desktop in modo tale da far comparire i vari Tab. Scegliete: **Filters** → **Data Analysis** → **Integrate Variables**, come evidenziato in Fig. 12



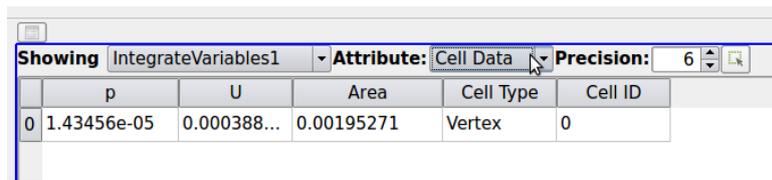
**Figura 11:** Slice1 (Piano monte,  $x=0$  mm)



**Figura 12:** Slice1, integrate variables

### 3.6: premete Apply.

*Dovrebbe comparirvi la vista di un foglio di calcolo analogo a quello di Fig. 13.*



	p	U	Area	Cell Type	Cell ID
0	1.43456e-05	0.000388...	0.00195271	Vertex	0

**Figura 13:** Slice1, spreadsheet

**3.7:** Nel foglio di calcolo settate **Attribute** uguale a CellData come visibile in Fig.13

**3.8:** Infine per calcolare il valore medio della pressione (in Pascal)

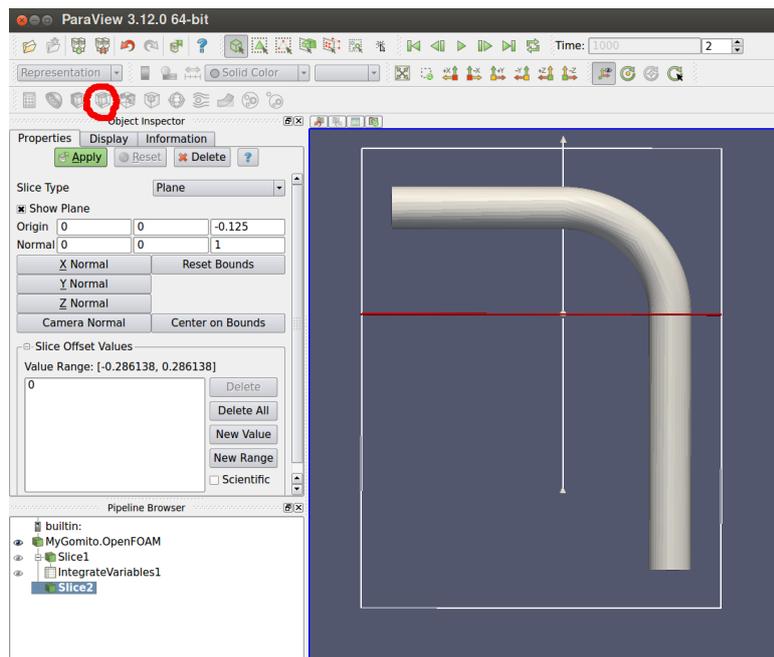
$$P1 = p/Area \cdot \rho = (1.43456 \times 10^{-5} / 0.00195271) \cdot 997 = 7.324 Pa$$

4. Per valutare la pressione in *Piano valle*, ( $z=-0.125m$ ) operate in modo analogo.

**4.1:** Cliccate sul simbolo cerchiato di Fig.14

**4.2:** Settate i campi **Origin** e **Normal** come in Fig. 14

**4.3:** Cliccate su **Apply**



**Figura 14:** Slice2 (Piano valle,  $z=-125$  mm)

**4.4:** Visualizzate esclusivamente lo Slice2.

**4.5:** Spostate il cursore del mouse sulle barra superiore del desktop in modo tale da far comparire il vari Tab. Scegliete **Filters** → **Data Analysis** → **Integrate Variables**, come evidenziato in Fig. 15.

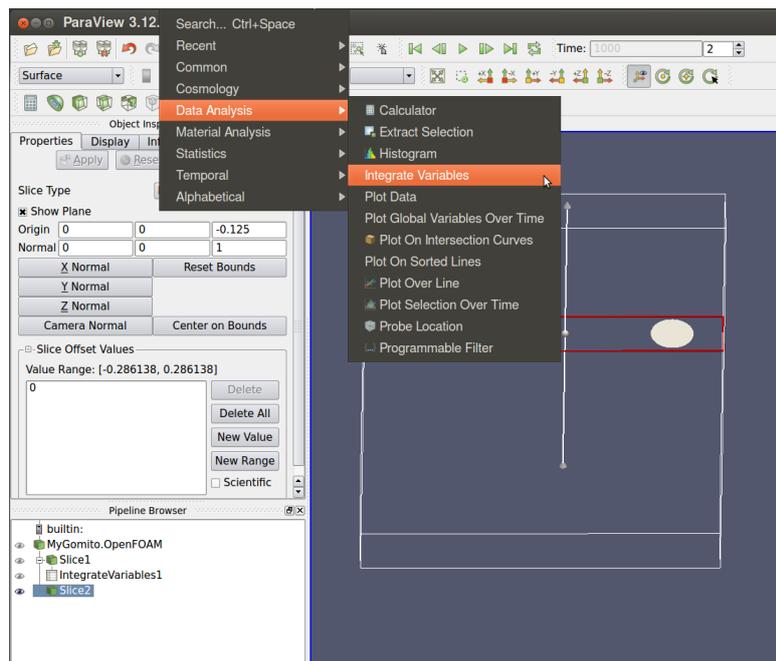


Figura 15: Slice2, integrate variables)

4.6 premete **Apply**. Dovrebbe ricomparirvi la vista del foglio di calcolo.

Showing	IntegrateVariables2	Attribute:	Cell Data	Precision:	6
	p	U	Area	Cell Type	
0	5.69132e-06	2.29798...	0.00195258	Vertex	

Figura 16: Slice2, spreadsheet

4.7 Assicuratevi che nel foglio di calcolo l'opzione **Showing** sia settata su IntegrateVariables2 e che **Attribute** sia su cellData, come visibile in Fig.16.

4.8 Ricavate il valore medio della pressione (in Pascal)

$$P2 = p/Area \cdot \rho = (5.69132 \times 10^{-6} / 0.00195258) \cdot 997 = 2.906 Pa$$

5. Il valore della caduta di pressione tra il *piano monte* ed il *piano valle* è quindi:

$$P1 - P2 = 7.324 - 2.906 = 4.418 Pa$$

## 5.2 Contour del campo di velocità

Per visualizzare il contour del campo di velocità nella sezione *Piano Valle* (Slice2) operate nel seguente modo:

1. Assicuratevi che sia visibile solo l'entità Slice2.

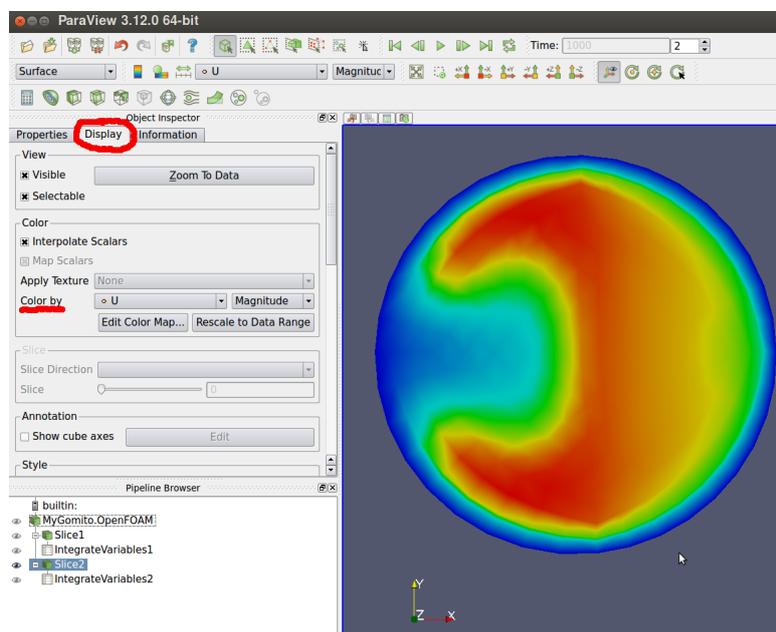
2. Come indicato in Fig.17:

**2.1** schiacciate/selezionate il tab **Display**

**2.2** settate l'opzione **Color by** con **U** e **Magnitude**

**2.3** schiacciate **Rescale to Data Range**

*Dovreste ottenere un campo di velocità analogo a quello visibile in Fig. 17.*



**Figura 17:** Slice2, contour del campo di velocità

3. Per visualizzare la legenda:

**3.1** schiacciate su **Edit Color Map**

**3.2** al comparire dalla finestrella di Fig.18 selezionate il Tab **Color Legend** e barrate poi l'opzione **Show Color Legend**

**3.3** Per ruotare la visuale ed ottenere il risultato di Fig. 19 dovete muovere il mouse in senso verticale tenendo premuti contemporaneamente il tasto sinistro del mouse e il testo SHIFT della tastiera.

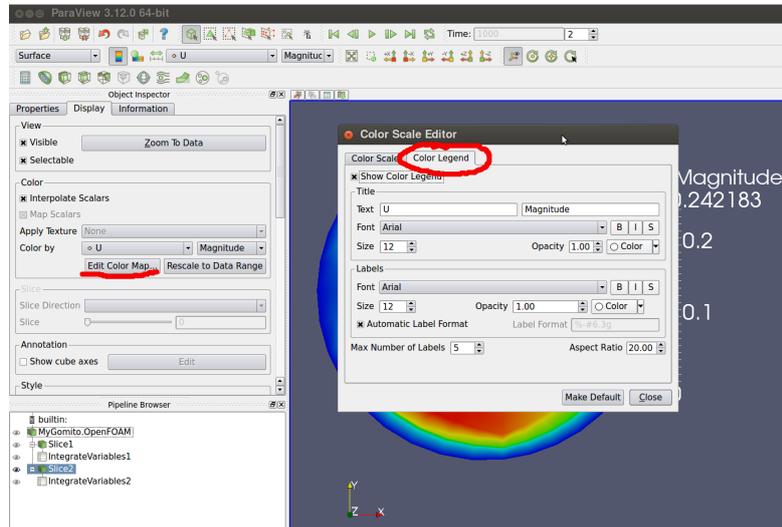


Figura 18: Slice2, contour del campo di velocità e della legenda

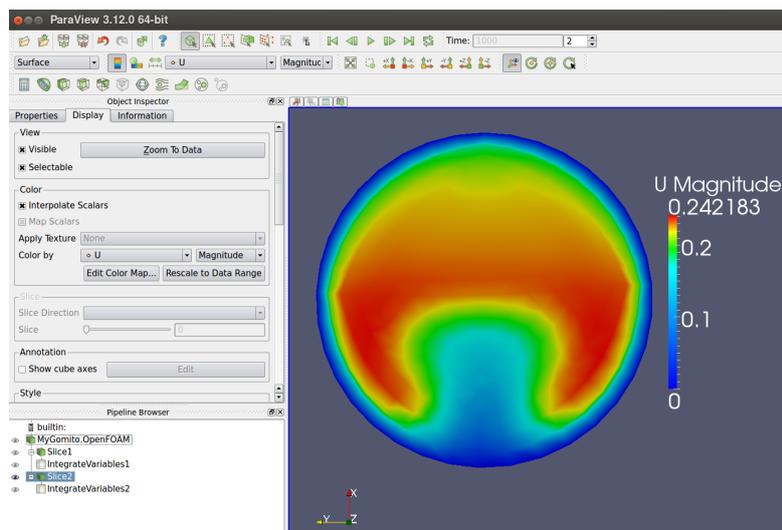


Figura 19: Slice2, contour del campo di velocità ruotato