

Introducción a OpenFoam

Presentación general de OpenFoam

E. Martín¹, M. Meis², F. Varas^{1,3}

1: Universidad de Vigo, 2: Vicus Desarrollos Tecnológicos, 3:
Universidad Politécnica de MADrid

Organización del curso

- <http://www.dma.uvigo.es/cursos.html>
 - Software Libre: OpenFoam (CFD)
 - Horario y distribución de sesiones

	Jueves 26 de Enero	Viernes 27 de Enero
10:00-12:00		Cálculo paralelo y postproceso sobre ejemplo "rotura de una presa"
12:00-14:00	Presentación general de OpenFOAM sobre ejemplo "cavidad + elbow 3D"	Ejemplo práctico: Hot room
16:00-18:00	Modelos y esquemas numéricos	Ejemplo práctico: Airfoil turbulento 2D
18:00-20:00	Resolvedores sobre ejemplo elbow 3D térmico	

- Ejercicios prácticos
- Enlaces de interés

¿Qué es OpenFoam?

Applications, Solvers, and Utilities

- OpenFOAM is first and foremost a *C++ library*, used primarily to create executables, known as *applications*. The applications fall into two categories: *solvers*, that are each designed to solve a specific continuum mechanics problem; and *utilities*, that are designed to perform tasks that involve data manipulation.
- OpenFOAM is distributed with a large number of applications, but soon any advanced user will start developing new applications for his/ her special needs. The basic way to do this is to find and copy an application that almost does what is needed, and then to modify it by copy/paste from other applications that has some features that are needed.
- Special applications for pre- and post-processing are included in OpenFOAM. Converters to/from other pre- and post-processors are available.

¿Qué es OpenFoam?

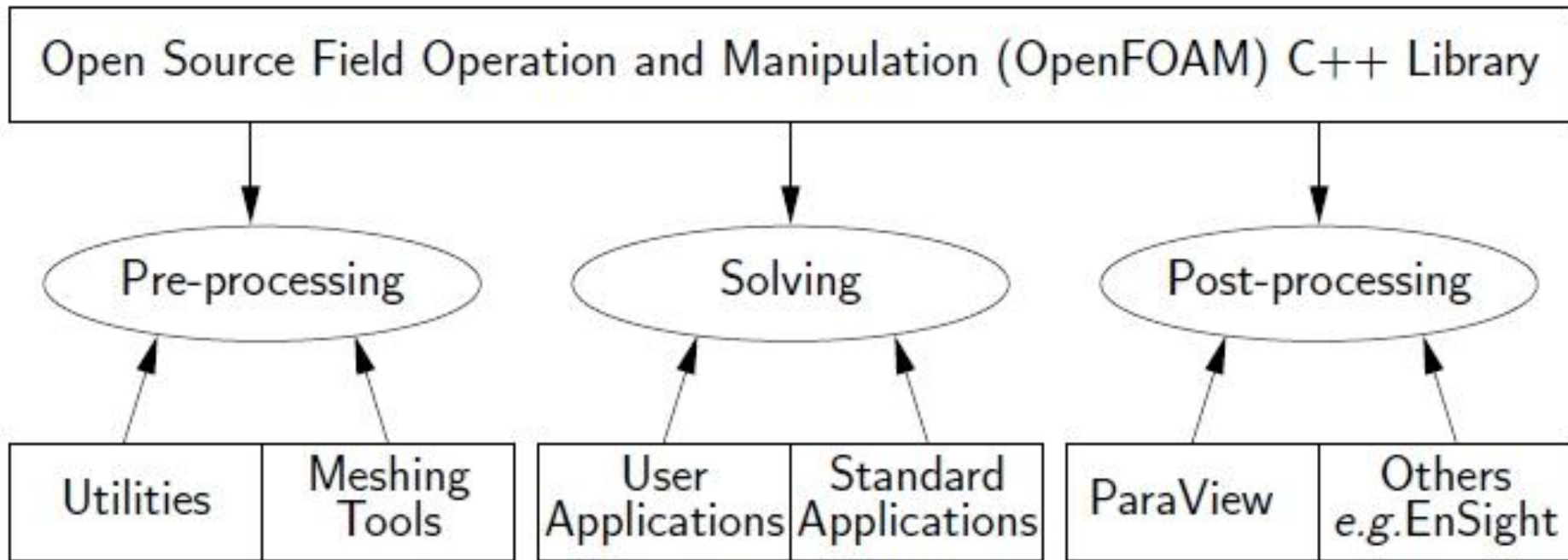


Figure 1.1: Overview of OpenFOAM structure.

Estructura de archivos del software



applications



bin



doc



etc



lib



src



tutorials

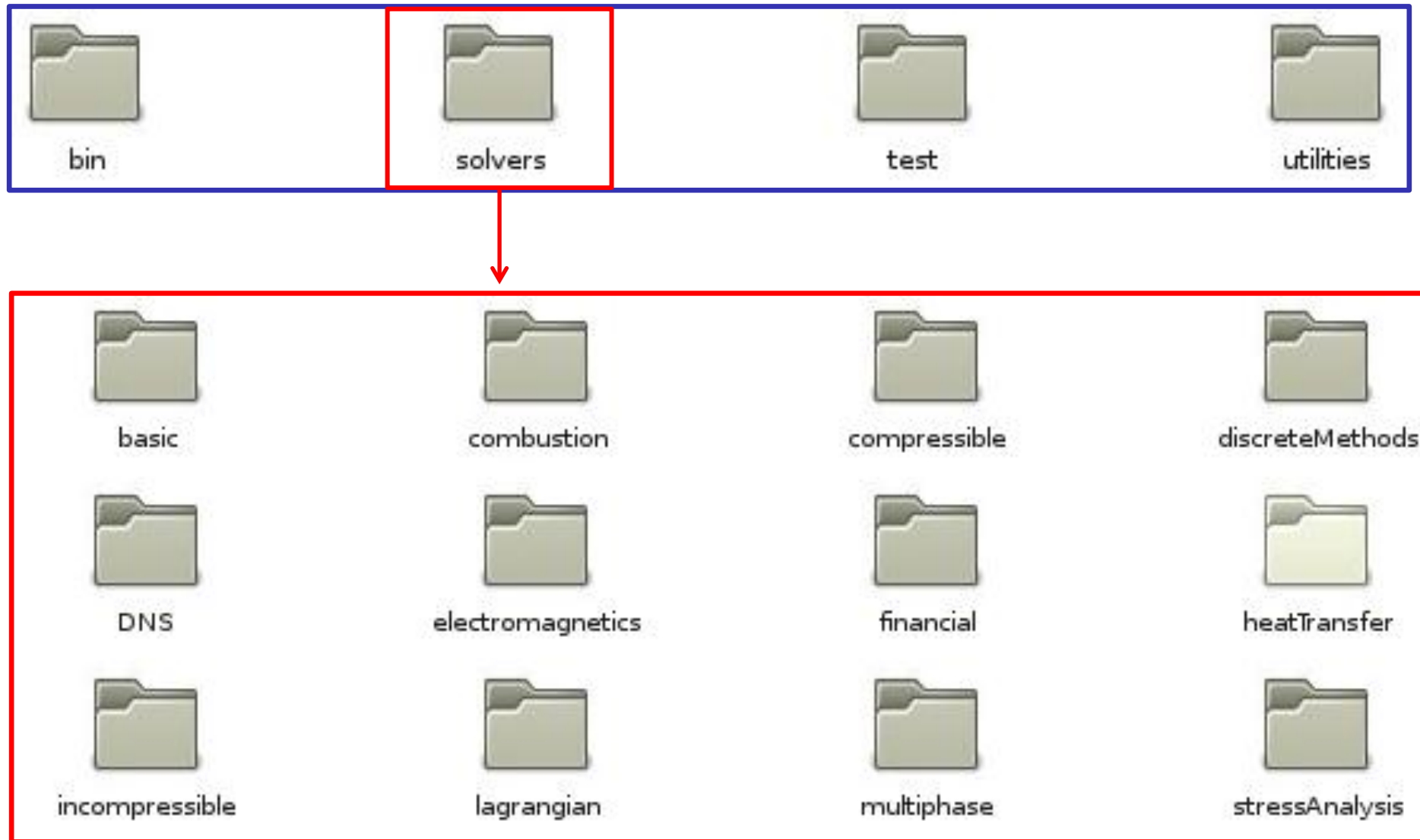


wmake

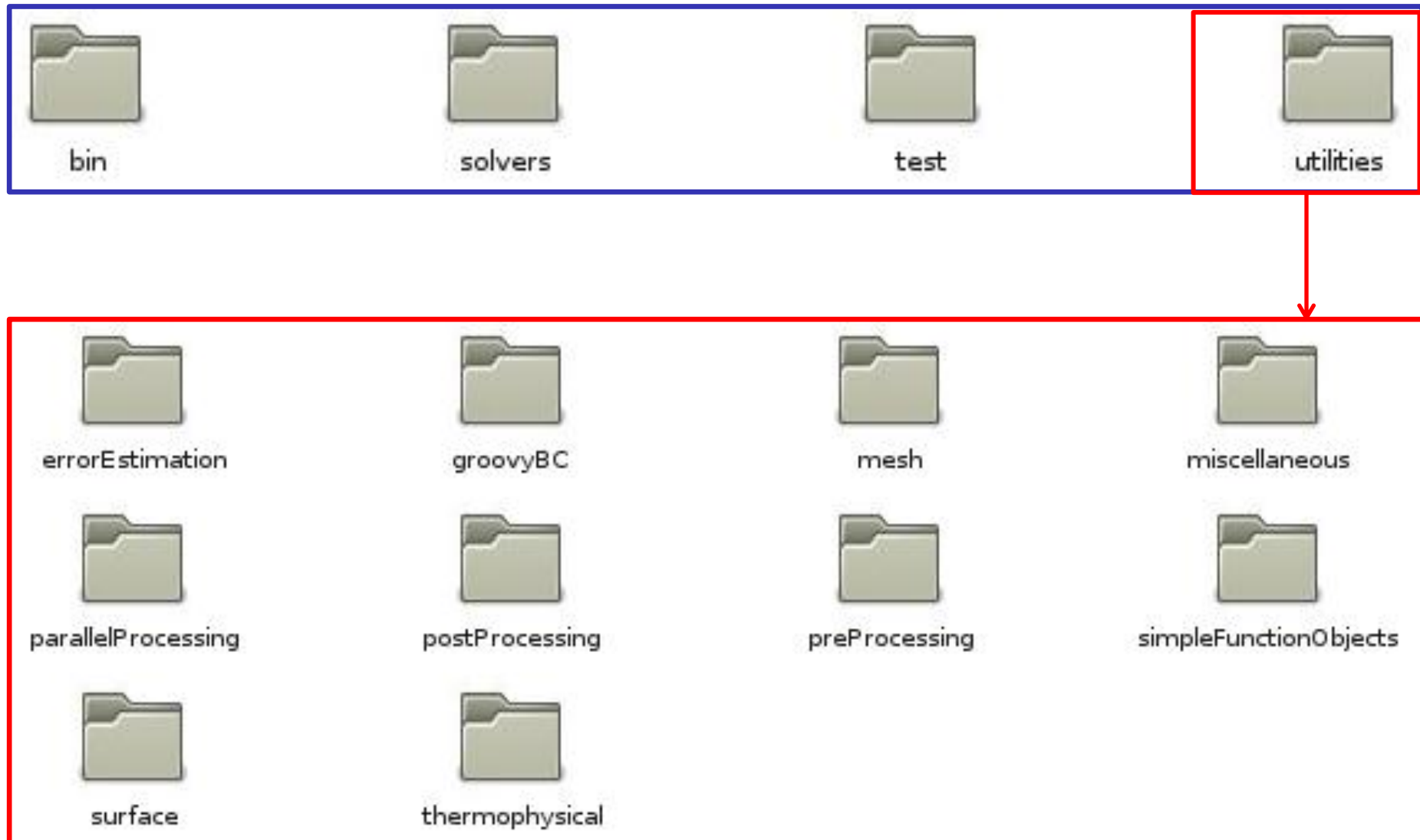


Allwmake

Estructura de archivos del software



Estructura de archivos del software



Estructura de archivos de un caso

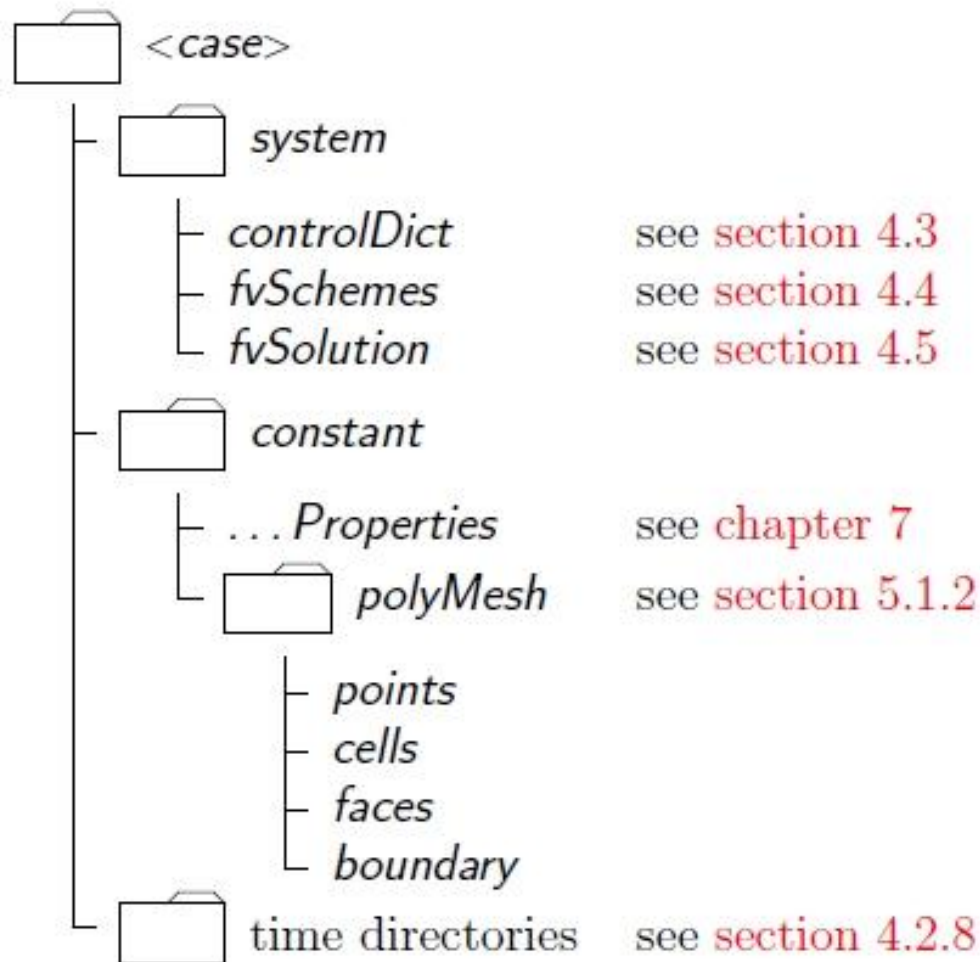


Figure 4.1: Case directory structure

Tutorial: Cavity

- Geometría 2D, incompresible, laminar, isotermo

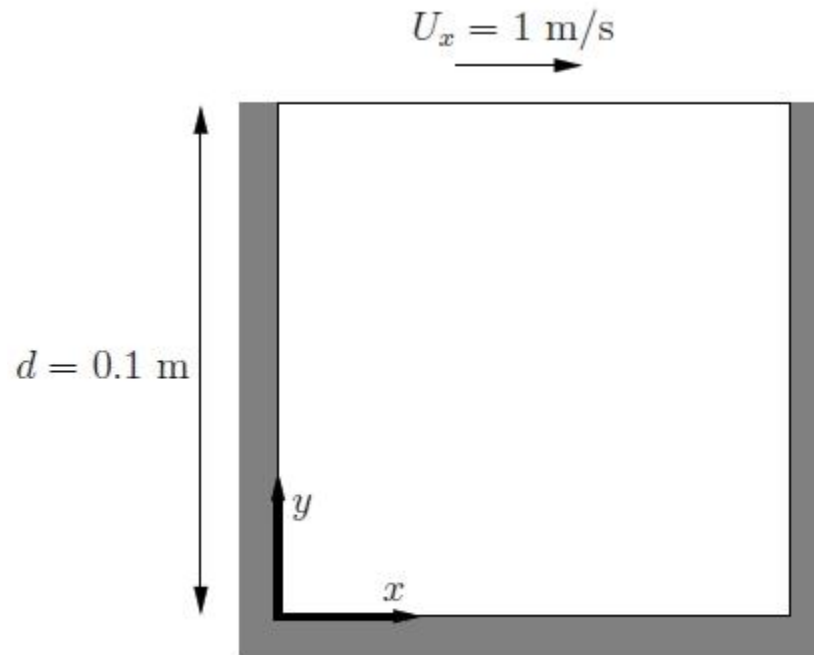
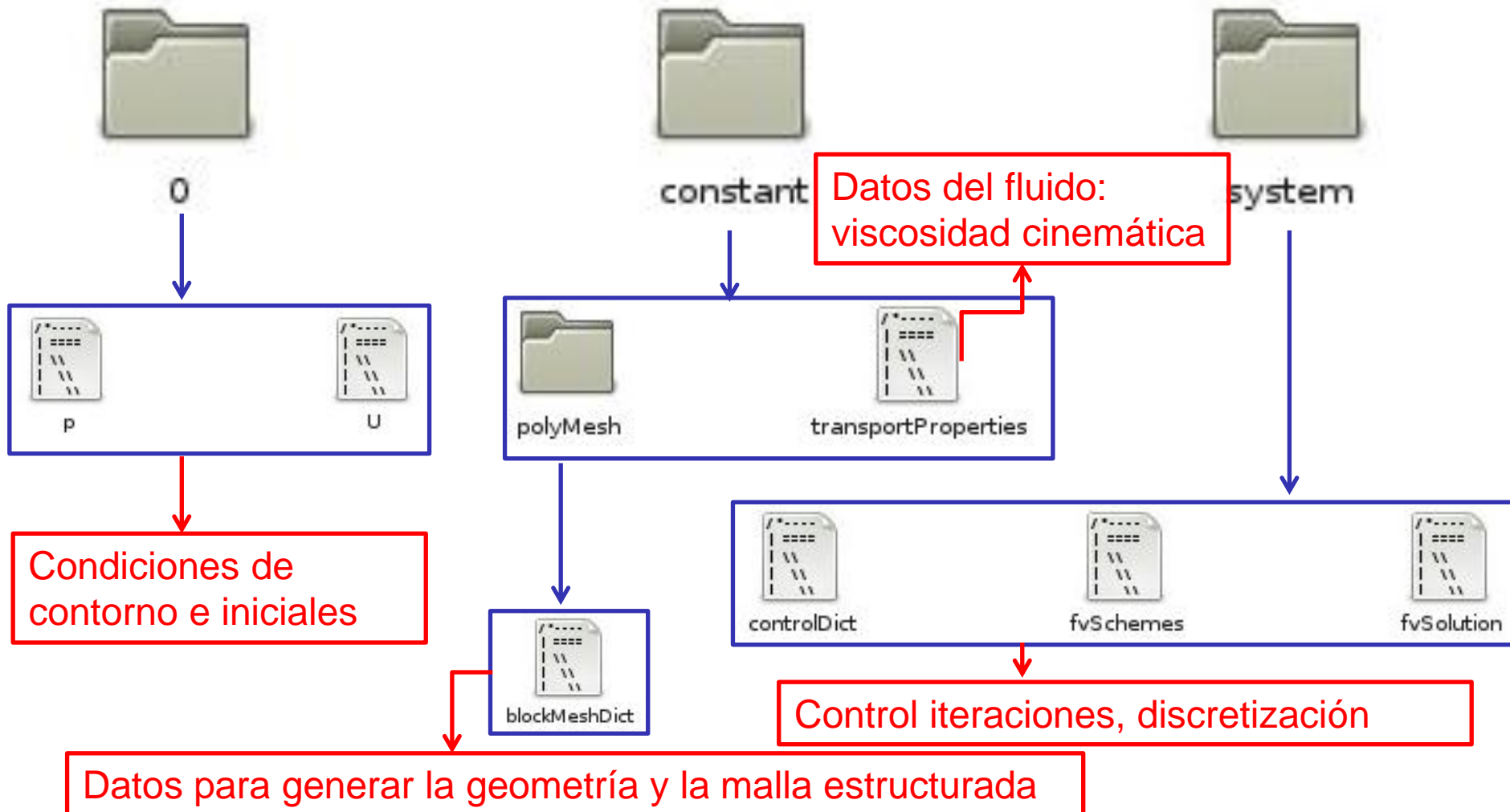


Figure 2.1: Geometry of the lid driven cavity.

Estructura de carpetas y ficheros

- Caso “cavity”:



Fichero: "transportProperties"

```
transportProperties X
/*-----*-- C++ --*-----*/
|=====|
| \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ \ / O p e r a t i o n | Version: 1.7.0
| \ \ / A n d | Web: www.OpenFOAM.com
| \ \ M a n i p u l a t i o n |
/*-----*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       transportProperties;
}

// *****

nu          nu [ 0 2 -1 0 0 0 0 ] 0.01;

// *****
```

Cabecera OpenFoam

Viscosidad cinemática fluido

Dimensiones de las variables

Dimensiones de las variables

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

Table 1.3: S.I. base units of measurement

Cavity: geometría

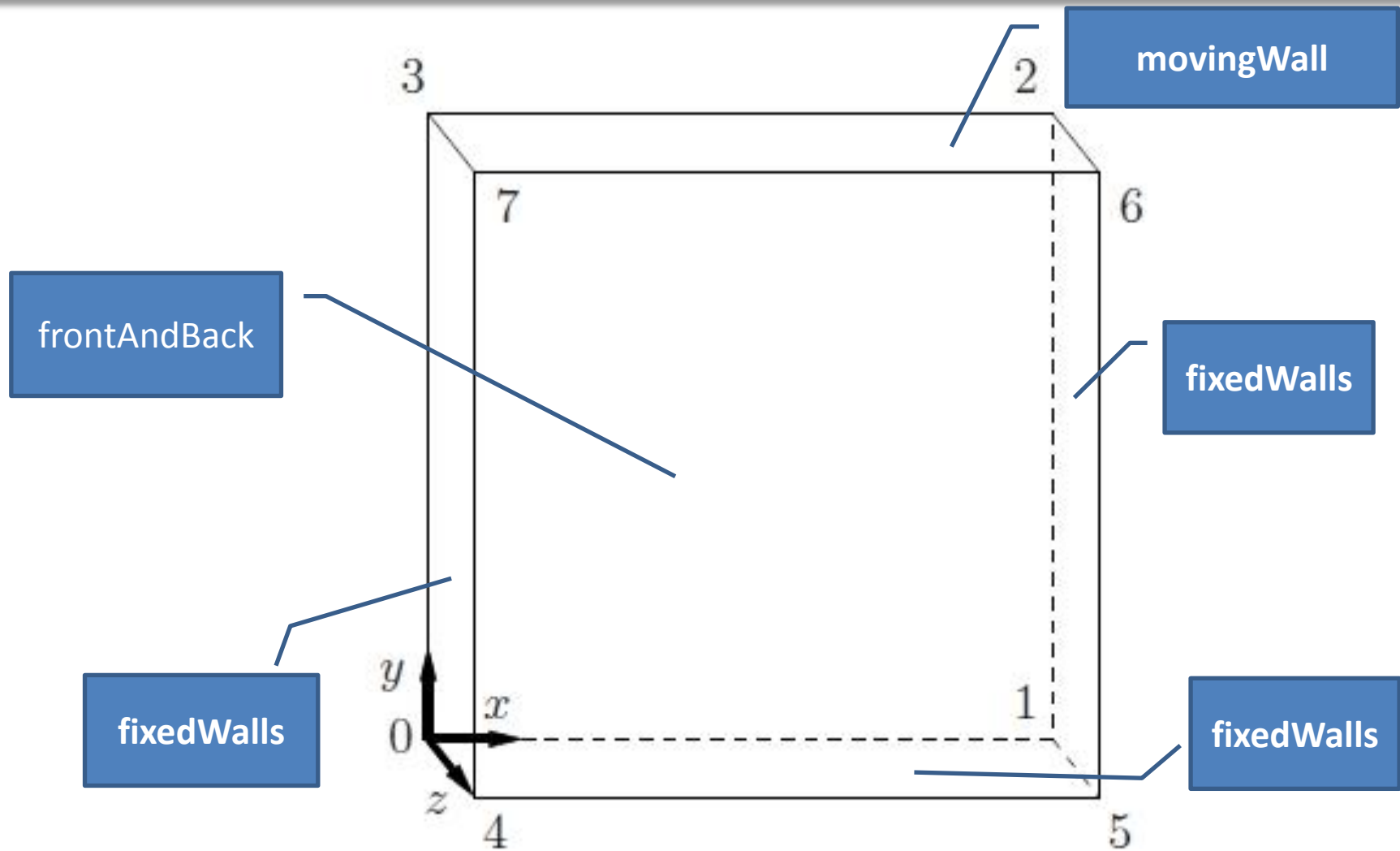


Figure 2.2: Block structure of the mesh for the cavity.

Fichero: “blockMeshDict”

```
blockMeshDict
/*----- C++ -----*/
|=====|
|  \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
|  \  /  O peration     | Version: 1.7.0
|  \  /  A nd           | Web: www.OpenFOAM.com
|  \  /  M anipulation  |
|=====|
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}
// *****

convertToMeters 0.1;

vertices
(
    (0 0 0)
    (1 0 0)
    (1 1 0)
    (0 1 0)
    (0 0 0.1)
    (1 0 0.1)
    (1 1 0.1)
    (0 1 0.1)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
);
```

Cabecera OpenFoam

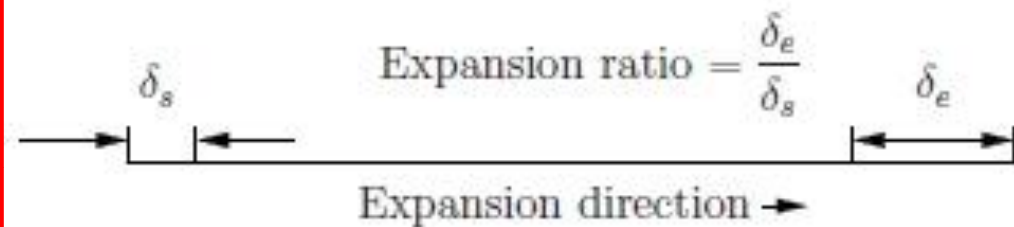


Figure 5.6: Mesh grading along a block edge

Fichero: “blockMeshDict”

```
edges
(
);

patches
(
    wall movingWall
    (
        ( 3 7 6 2 )
    )
    wall fixedWalls
    (
        ( 0 4 7 3 )
        ( 2 6 5 1 )
        ( 1 5 4 0 )
    )
    empty frontAndBack
    (
        ( 0 3 2 1 )
        ( 4 5 6 7 )
    )
);

mergePatchPairs
(
);

// ..... //
```

The diagram illustrates the structure of the `patches` section in a `blockMeshDict` file. Red boxes highlight the following elements:

- `patches`: Points to the label "Contornos/fronteras".
- `movingWall`: Points to the label "Nombres de los contornos".
- `fixedWalls`: Points to the label "Nombres de los contornos".
- `frontAndBack`: Points to the label "Tipo de contorno".

The numbers inside the parentheses (e.g., 3 7 6 2, 0 4 7 3, 2 6 5 1, 1 5 4 0, 0 3 2 1, 4 5 6 7) represent the indices of the mesh edges that form each boundary.

Tipos de contornos

Selection Key	Description
patch	generic patch
symmetryPlane	plane of symmetry
empty	front and back planes of a 2D geometry
wedge	wedge front and back for an axi-symmetric geometry
cyclic	cyclic plane
wall	wall — used for wall functions in turbulent flows
processor	inter-processor boundary

Table 5.2: Basic patch types.

Generación de malla propia de OF

Mesh generation

blockMesh	A multi-block mesh generator
extrude2DMesh	Takes 2D mesh (all faces 2 points only, no front and back faces) and creates a 3D mesh by extruding with specified thickness
extrudeMesh	Extrude mesh from existing patch (by default outwards facing normals; optional flips faces) or from patch read from file
snappyHexMesh	Automatic split hex mesher. Refines and snaps to surface

- Ejecutar en un terminal en el directorio del caso:
 - blockMesh

Generación de malla con blockMesh

- Archivos generados:



blockMeshDict



boundary



faces



neighbour



owner



points

Generación de malla con blockMesh

- Archivo “boundary”: contiene los contornos de la geometría
- “points”: coordenadas 3D de los vértices de la malla
- “faces”: construcción de las caras de las celdas de la malla a partir del número de cada vértice

Archivo
“boundary”

```
3
{
    movingWall
    {
        type            wall;
        nFaces           20;
        startFace        760;
    }
    fixedWalls
    {
        type            wall;
        nFaces           60;
        startFace        780;
    }
    frontAndBack
    {
        type            empty;
        nFaces           800;
        startFace        840;
    }
}
```

Vector normal a cada cara

- Sentido antihorario

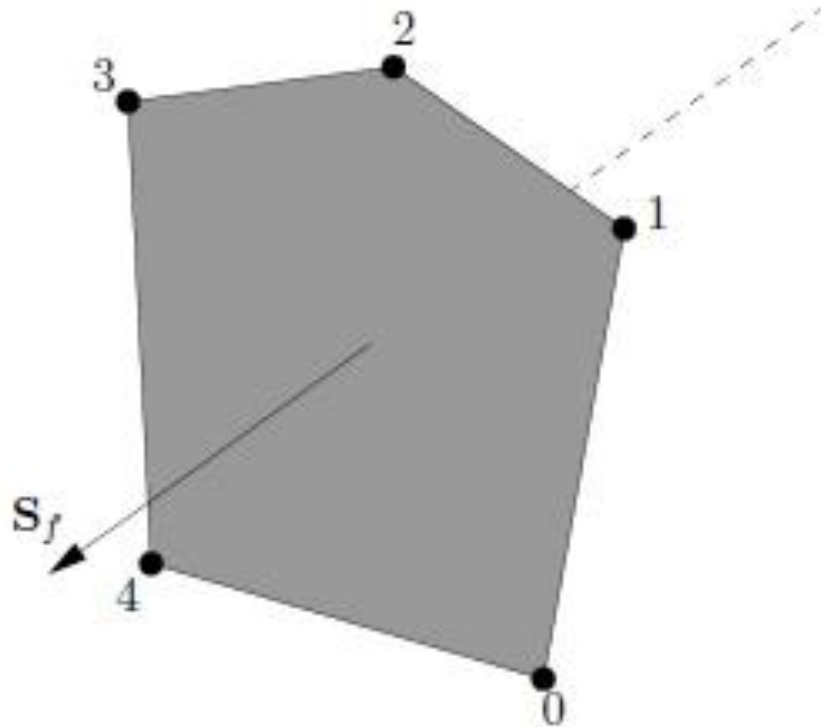


Figure 5.1: Face area vector from point numbering on the face

Condiciones iniciales y de contorno: “U”

```
dimensions      [0 1 -1 0 0 0 0];  
internalField   uniform (0 0 0);  
boundaryField  
{  
    movingWall  
    {  
        type      fixedValue;  
        value      uniform (1 0 0);  
    }  
    fixedWalls  
    {  
        type      fixedValue;  
        value      uniform (0 0 0);  
    }  
    frontAndBack  
    {  
        type      empty;  
    }  
}
```

Condición inicial de U

Condiciones de contorno

Tipos de condiciones de contorno

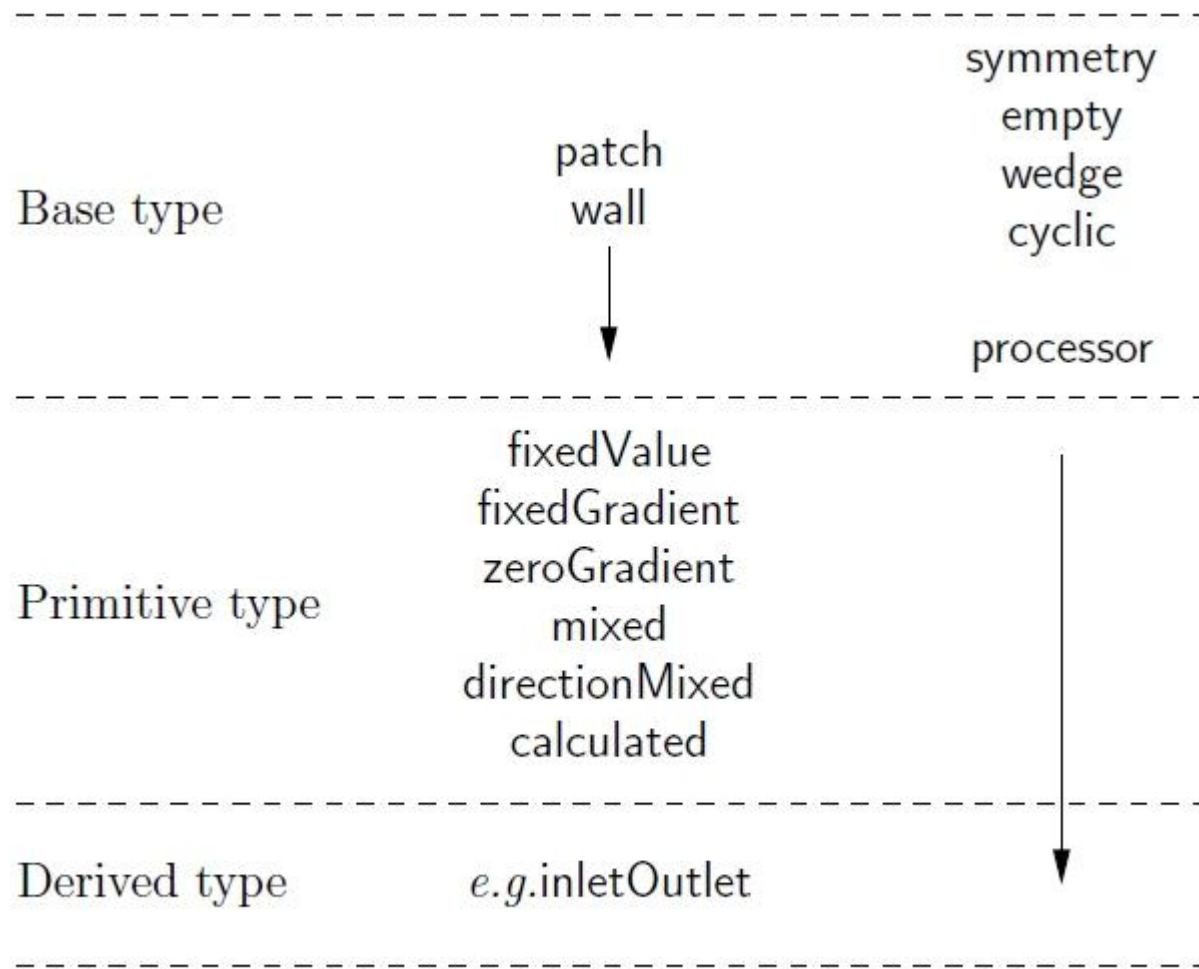


Figure 5.2: Patch attributes

Tipos de condiciones de contorno

Type	Description of condition for patch field ϕ	Data to specify
fixedValue	Value of ϕ is specified	value
fixedGradient	Normal gradient of ϕ is specified	gradient
zeroGradient	Normal gradient of ϕ is zero	—
calculated	Boundary field ϕ derived from other fields	—
mixed	Mixed fixedValue/ fixedGradient condition depending on the value in valueFraction	refValue, refGradient, valueFraction, value
directionMixed	A mixed condition normal to the patch with a fixedGradient condition tangential to the patch	refValue, refGradient, valueFraction, value

Table 5.3: Primitive patch field types.

Tipos de condiciones de contorno

5.2.4 Derived types

There are numerous derived types of boundary conditions in OpenFOAM, too many to list here. Instead a small selection is listed in [Table 5.4](#). If the user wishes to obtain a list of all available model, they should consult the OpenFOAM source code. Derived boundary condition source code can be found at the following locations:

- in *\$FOAM_SRC/finiteVolume/fields/fvPatchFields/derived*
- within certain model libraries, that can be located by typing the following command in a terminal window

```
find $FOAM_SRC -name "*derivedFvPatch*"
```

- within certain solvers, that can be located by typing the following command in a terminal window

```
find $FOAM_SOLVERS -name "*fvPatch*"
```


Tipos de condiciones de contorno

Types derived from fixedValue		Data to specify
movingWallVelocity	Replaces the normal of the patch <code>value</code> so the flux across the patch is zero	value
pressureInletVelocity	When p is known at inlet, \mathbf{U} is evaluated from the flux, normal to the patch	value
pressureDirectedInletVelocity	When p is known at inlet, \mathbf{U} is calculated from the flux in the <code>inletDirection</code>	value, inletDirection
surfaceNormalFixedValue	Specifies a vector boundary condition, normal to the patch, by its magnitude; +ve for vectors pointing out of the domain	value
totalPressure	Total pressure $p_0 = p + \frac{1}{2}\rho \mathbf{U} ^2$ is fixed; when \mathbf{U} changes, p is adjusted accordingly	p0
turbulentInlet	Calculates a fluctuating variable based on a scale of a mean value	referenceField, fluctuationScale
Types derived from fixedGradient/zeroGradient		
fluxCorrectedVelocity	Calculates normal component of \mathbf{U} at inlet from flux	value
wallBuoyantPressure	Sets <code>fixedGradient</code> pressure based on the atmospheric pressure gradient	—
Types derived from mixed		
inletOutlet	Switches \mathbf{U} and p between <code>fixedValue</code> and <code>zeroGradient</code> depending on direction of \mathbf{U}	inletValue, value
outletInlet	Switches \mathbf{U} and p between <code>fixedValue</code> and <code>zeroGradient</code> depending on direction of \mathbf{U}	outletValue, value
pressureInletOutletVelocity	Combination of <code>pressureInletVelocity</code> and <code>inletOutlet</code>	value
pressureDirectedInletOutletVelocity	Combination of <code>pressureDirectedInletVelocity</code> and <code>inletOutlet</code>	value, inletDirection
pressureTransmissive	Transmits supersonic pressure waves to surrounding pressure p_∞	pInf
supersonicFreeStream	Transmits oblique shocks to surroundings at $p_\infty, T_\infty, \mathbf{U}_\infty$	pInf, TInf, UInf
Other types		
slip	<code>zeroGradient</code> if ϕ is a scalar; if ϕ is a vector, normal component is <code>fixedValue</code> zero, tangential components are <code>zeroGradient</code>	—
partialSlip	Mixed <code>zeroGradient</code> / <code>slip</code> condition depending on the <code>valueFraction</code> ; = 1 for slip	valueFraction
Note: p is pressure, \mathbf{U} is velocity		

Table 5.4: Derived patch field types.

Condiciones iniciales y de contorno: “U”

```
dimensions      [0 1 -1 0 0 0 0];  
internalField   uniform (0 0 0);  
boundaryField  
{  
    movingWall  
    {  
        type      fixedValue;  
        value      uniform (1 0 0);  
    }  
    fixedWalls  
    {  
        type      fixedValue;  
        value      uniform (0 0 0);  
    }  
    frontAndBack  
    {  
        type      empty;  
    }  
}
```

Condición inicial de U

Condiciones de contorno

Condiciones iniciales y de contorno: “p”

```
dimensions      [0 2 -2 0 0 0 0];  
internalField   uniform 0;  
boundaryField  
{  
    movingWall  
    {  
        type      zeroGradient;  
    }  
    fixedWalls  
    {  
        type      zeroGradient;  
    }  
    frontAndBack  
    {  
        type      empty;  
    }  
}
```

Condición inicial de p

Condiciones de contorno
Para la presión p

Ficheros de “system”: controlDict

```
application      icoFoam;
startFrom        startTime;
startTime        0;
stopAt           endTime;
endTime          0.5;
deltaT           0.005;
writeControl      timeStep;
writeInterval     20;
purgeWrite        0;
writeFormat       ascii;
writePrecision    6;
writeCompression  uncompressed;
timeFormat        general;
timePrecision     6;
runTimeModifiable yes;
```

Application Solver

Paso temporal

Solvers: ejemplos

'Basic' CFD codes

laplacianFoam	Solves a simple Laplace equation, e.g. for thermal diffusion in a solid
potentialFoam	Simple potential flow solver which can be used to generate starting fields for full Navier-Stokes codes
scalarTransportFoam	Solves a transport equation for a passive scalar

Incompressible flow

boundaryFoam	Steady-state solver for 1D turbulent flow, typically to generate boundary layer conditions at an inlet, for use in a simulation
channelFoam	Incompressible LES solver for flow in a channel
icoFoam	Transient solver for incompressible, laminar flow of Newtonian fluids
nonNewtonianIcoFoam	Transient solver for incompressible, laminar flow of non-Newtonian fluids
pimpleDyMFoam	Transient solver for incompressible, flow of Newtonian fluids on a moving mesh using the PIMPLE (merged PISO-SIMPLE) algorithm
pimpleFoam	Large time-step transient solver for incompressible, flow using the PIMPLE (merged PISO-SIMPLE) algorithm
pisoFoam	Transient solver for incompressible flow
shallowWaterFoam	Transient solver for inviscid shallow-water equations with rotation
simpleFoam	Steady-state solver for incompressible, turbulent flow

Solvers: ejemplos

Compressible flow

rhoCentralFoam	Density-based compressible flow solver based on central-upwind schemes of Kurganov and Tadmor
rhoPimpleFoam	Transient solver for laminar or turbulent flow of compressible fluids for HVAC and similar applications
rhoPisoFoam	Transient PISO solver for compressible, laminar or turbulent flow
rhoPorousSimpleFoam	Steady-state solver for turbulent flow of compressible fluids with RANS turbulence modelling, and implicit or explicit porosity treatment
rhoPsonicFoam	Pressure-density-based compressible flow solver
rhoSimpleFoam	Steady-state SIMPLE solver for laminar or turbulent RANS flow of compressible fluids
rhoSonicFoam	Density-based compressible flow solver
sonicDyMFoam	Transient solver for trans-sonic/supersonic, laminar or turbulent flow of a compressible gas with mesh motion
sonicFoam	Transient solver for trans-sonic/supersonic, laminar or turbulent flow of a compressible gas
sonicLiquidFoam	Transient solver for trans-sonic/supersonic, laminar flow of a compressible liquid

Solvers: ejemplos

Heat transfer and buoyancy-driven flows

buoyantBoussinesqPisoFoam	Transient solver for buoyant, turbulent flow of incompressible fluids
buoyantBoussinesqSimpleFoam	Steady-state solver for buoyant, turbulent flow of incompressible fluids
buoyantPisoFoam	Transient solver for buoyant, turbulent flow of compressible fluids for ventilation and heat-transfer
buoyantSimpleFoam	Steady-state solver for buoyant, turbulent flow of compressible fluids
buoyantSimpleRadiationFoam	Steady-state solver for buoyant, turbulent flow of compressible fluids, including radiation, for ventilation and heat-transfer
chtMultiRegionFoam	Combination of heatConductionFoam and buoyantFoam for conjugate heat transfer between a solid region and fluid region

Ficheros de “system”: fvSchemes

```

ddtSchemes
{
    default Euler;
}

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

divSchemes
{
    default none;
    div(phi,U) Gauss linear;
}

laplacianSchemes
{
    default none;
    laplacian(nu,U) Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
}

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

snGradSchemes
{
    default corrected;
}
    
```

Scheme	Description
Euler	First order, bounded, implicit
CrankNicholson ψ	Second order, bounded, implicit
backward	Second order, implicit
steadyState	Does not solve for time derivatives

Table 4.11: Discretisation schemes available in *ddtSchemes*.

Keyword	Category of mathematical terms
interpolationSchemes	Point-to-point interpolations of values
snGradSchemes	Component of gradient normal to a cell face
gradSchemes	Gradient ∇
divSchemes	Divergence $\nabla \cdot$
laplacianSchemes	Laplacian ∇^2
timeScheme	First and second time derivatives $\partial/\partial t, \partial^2/\partial^2 t$
fluxRequired	Fields which require the generation of a flux

Table 4.5: Main keywords used in *fvSchemes*.

Ficheros de “system”: fvSchemes

```

ddtSchemes
{
    default Euler;
}

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

divSchemes
{
    default none;
    div(phi,U) Gauss linear;
}

laplacianSchemes
{
    default none;
    laplacian(nu,U) Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
}

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

snGradSchemes
{
    default corrected;
}
    
```

Centred schemes

linear	Linear interpolation (central differencing)
cubicCorrection	Cubic scheme
midPoint	Linear interpolation with symmetric weighting

Upwinded convection schemes

upwind	Upwind differencing
linearUpwind	Linear upwind differencing
skewLinear	Linear with skewness correction
QUICK	Quadratic upwind differencing

TVD schemes

limitedLinear	limited linear differencing
vanLeer	van Leer limiter
MUSCL	MUSCL limiter
limitedCubic	Cubic limiter

NVD schemes

SFCD	Self-filtered central differencing
Gamma ψ	Gamma differencing

Table 4.6: Interpolation schemes.

Ficheros de “system”: fvSolution

```
solvers
{
    p
    {
        solver          PCG;
        preconditioner   DIC;
        tolerance        1e-06;
        relTol           0;
    }

    U
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0;
    }
}

PISO
{
    nCorrectors          2;
    nNonOrthogonalCorrectors 0;
    pRefCell              0;
    pRefValue             0;
}
```

Linear Solvers

Solver	Keyword
Preconditioned (bi-)conjugate gradient	PCG/PBiCG†
Solver using a smoother	smoothSolver
Generalised geometric-algebraic multi-grid	GAMG
†PCG for symmetric matrices, PBiCG for asymmetric	

Table 4.12: Linear solvers.

Ejecución del caso “cavity”

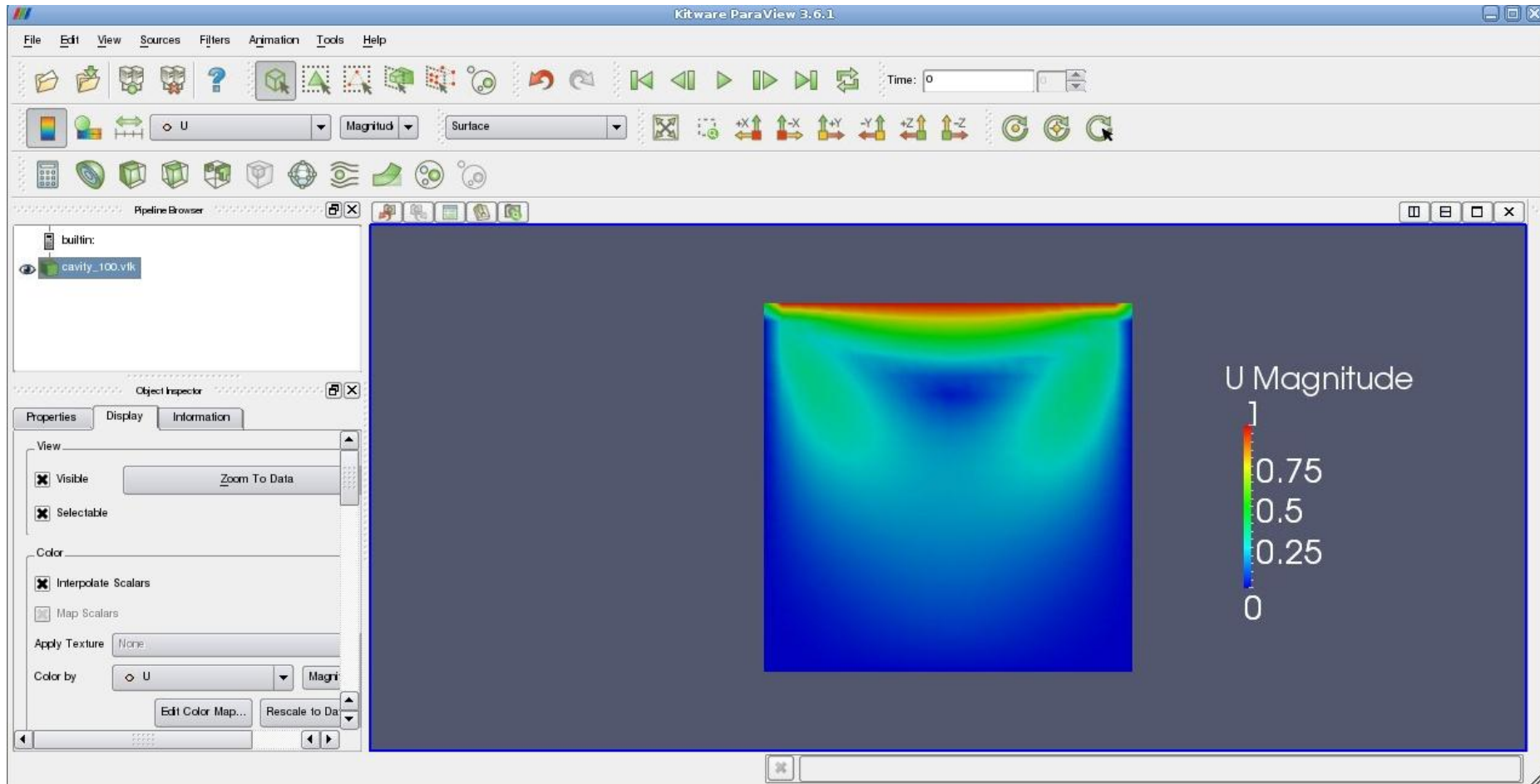
- Ejecutar en un terminal en el directorio del caso:
 - `icoFoam > log` → Aparecen archivos resultados para cada instante de tiempo



Postprocesado

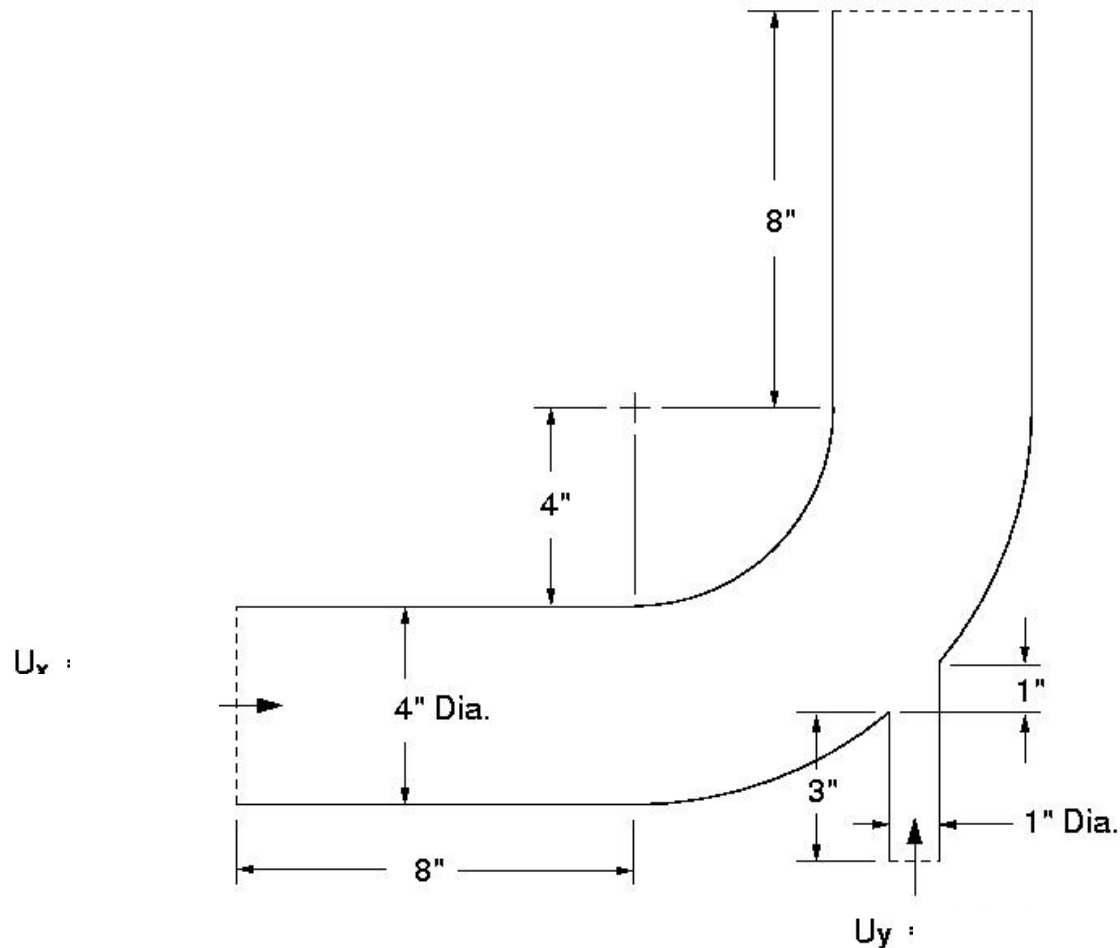
- Postprocesado:
 - foamToVTK -time 0:0.5
 - Transforma los ficheros de resultados a formato VTK
 - Paraview &
 - Visualiza los resultados obtenidos en la simulación

Postprocesado

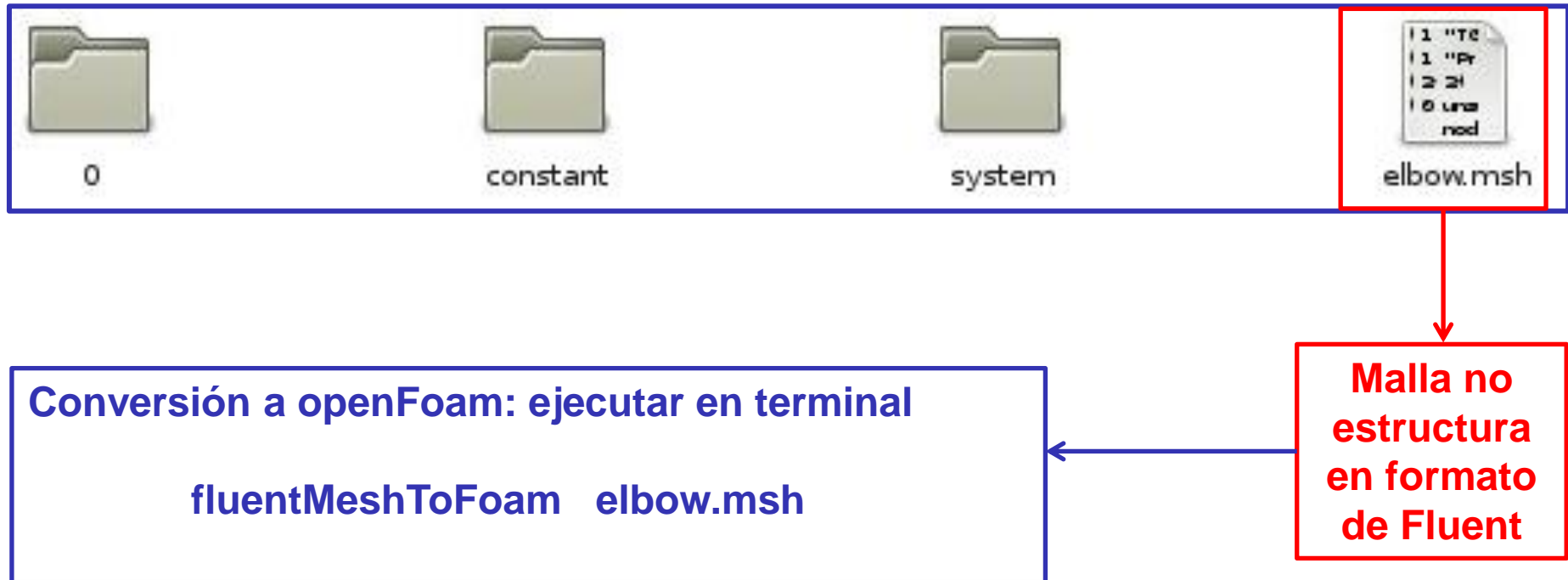


Tutorial “elbow_2D”

- Ejemplo 2D, laminar, incompresible e isotermo



Tutorial “elbow”



Conversiones de mallas

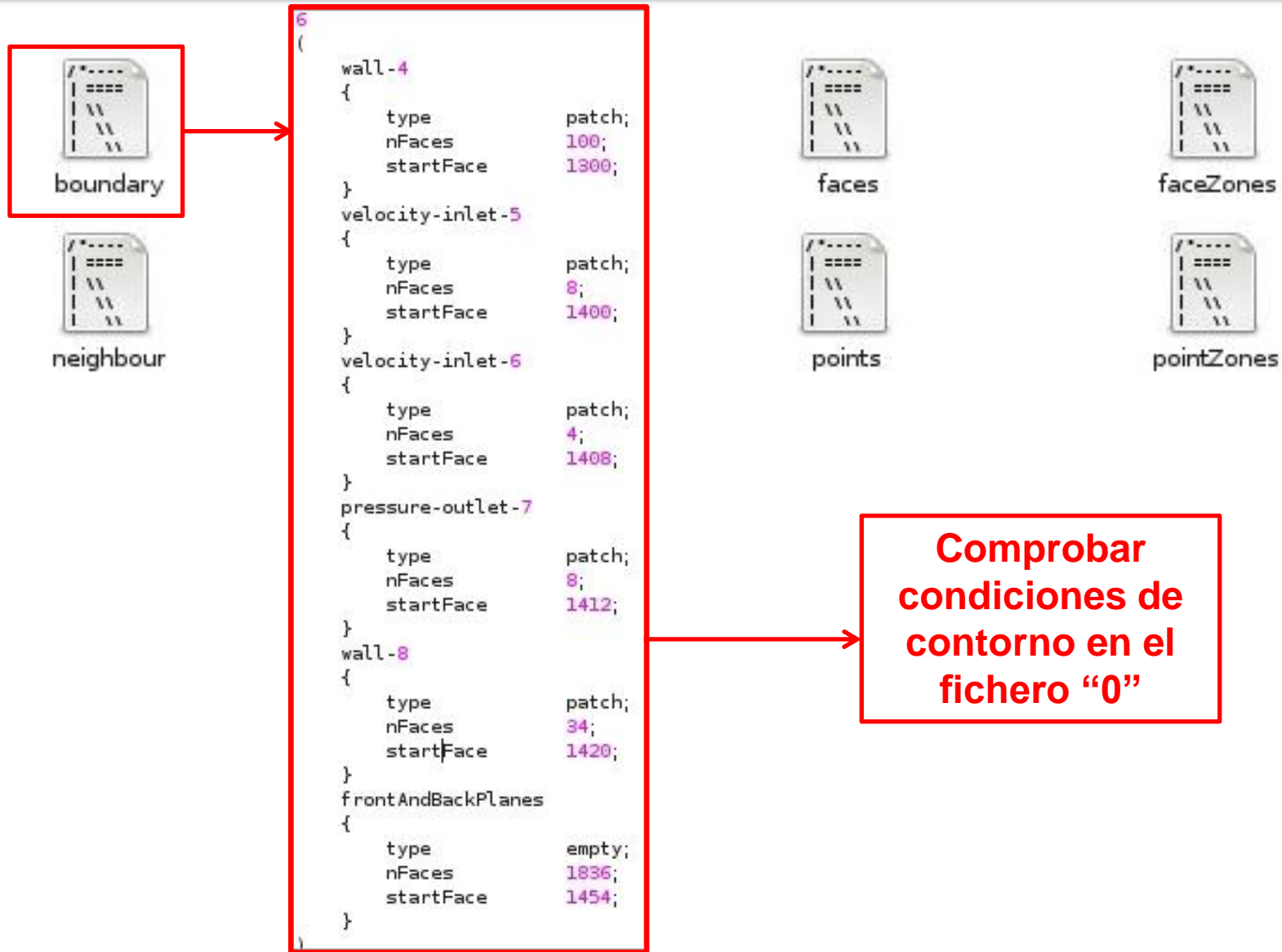
Mesh conversion

ansysToFoam	Converts an ANSYS input mesh file, exported from I-DEAS, to OpenFOAM format
cfx4ToFoam	Converts a CFX 4 mesh to OpenFOAM format
fluent3DMeshToFoam	Converts a Fluent mesh to OpenFOAM format
fluentMeshToFoam	Converts a Fluent mesh to OpenFOAM format including multiple region and region boundary handling
foamMeshToFluent	Writes out the OpenFOAM mesh in Fluent mesh format
foamToStarMesh	Reads an OpenFOAM mesh and writes a PROSTAR (v4) bnd/cel/vrt format
gambitToFoam	Converts a GAMBIT mesh to OpenFOAM format
gmshToFoam	Reads .msh file as written by Gmsh
ideasUnvToFoam	I-Deas unv format mesh conversion
kivaToFoam	Converts a KIVA grid to OpenFOAM format
mshToFoam	Converts .msh file generated by the Adventure system
netgenNeutralToFoam	Converts neutral file format as written by Netgen v4.4
plot3dToFoam	Plot3d mesh (ascii/formatted format) converter

Conversiones de mallas

<code>polyDualMesh</code>	Calculate the dual of a <code>polyMesh</code> . Adheres to all the feature and patch edges
<code>sammToFoam</code>	Converts a STAR-CD SAMM mesh to OpenFOAM format
<code>star4ToFoam</code>	Converts a STAR-CD (v4) PROSTAR mesh into OpenFOAM format
<code>starToFoam</code>	Converts a STAR-CD PROSTAR mesh into OpenFOAM format
<code>tetgenToFoam</code>	Converts <code>.ele</code> and <code>.node</code> and <code>.face</code> files, written by <code>tetgen</code>
<code>writeMeshObj</code>	For mesh debugging: writes mesh as three separate OBJ files which can be viewed with e.g. <code>javaview</code>

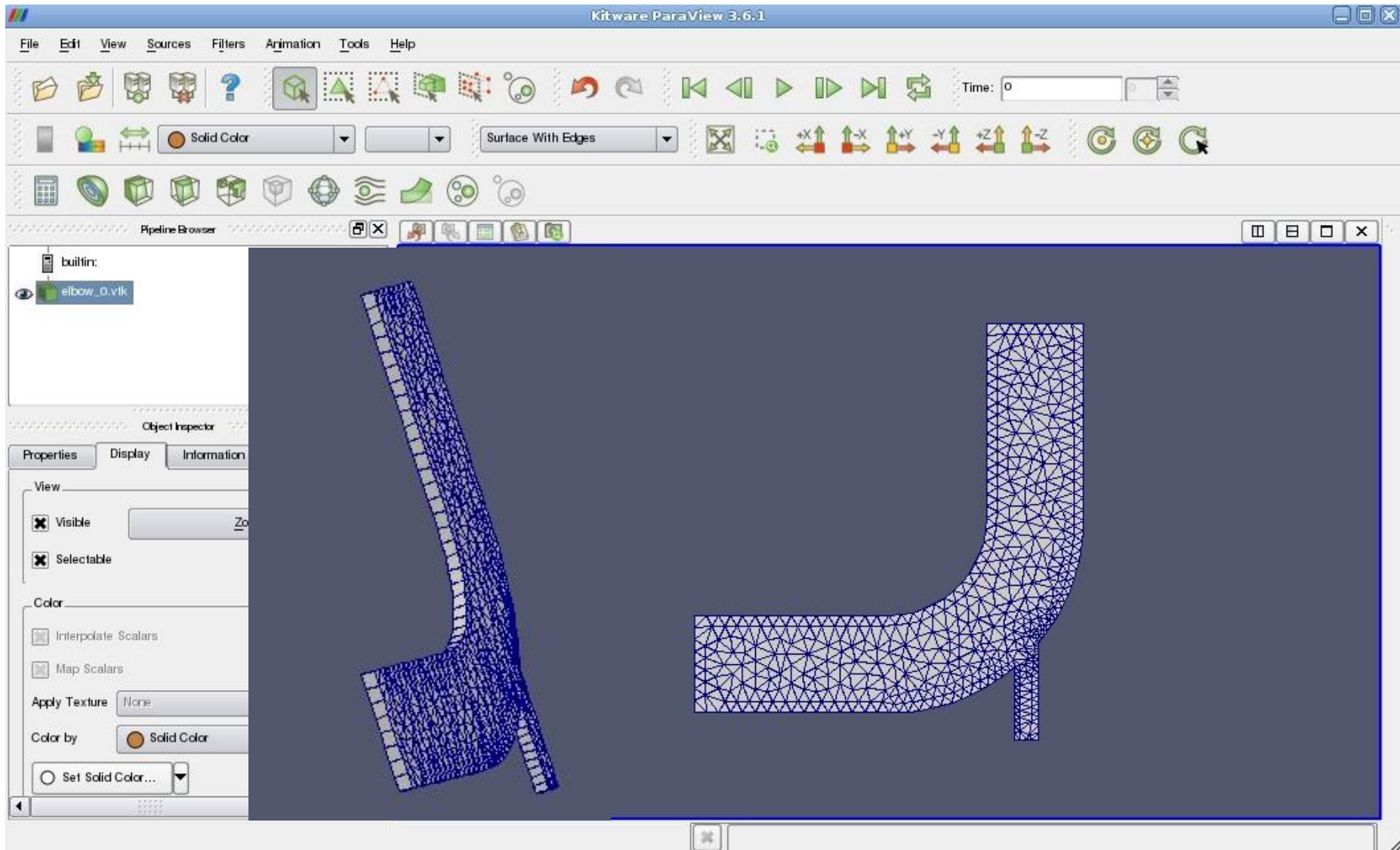
Directorio “polyMesh”



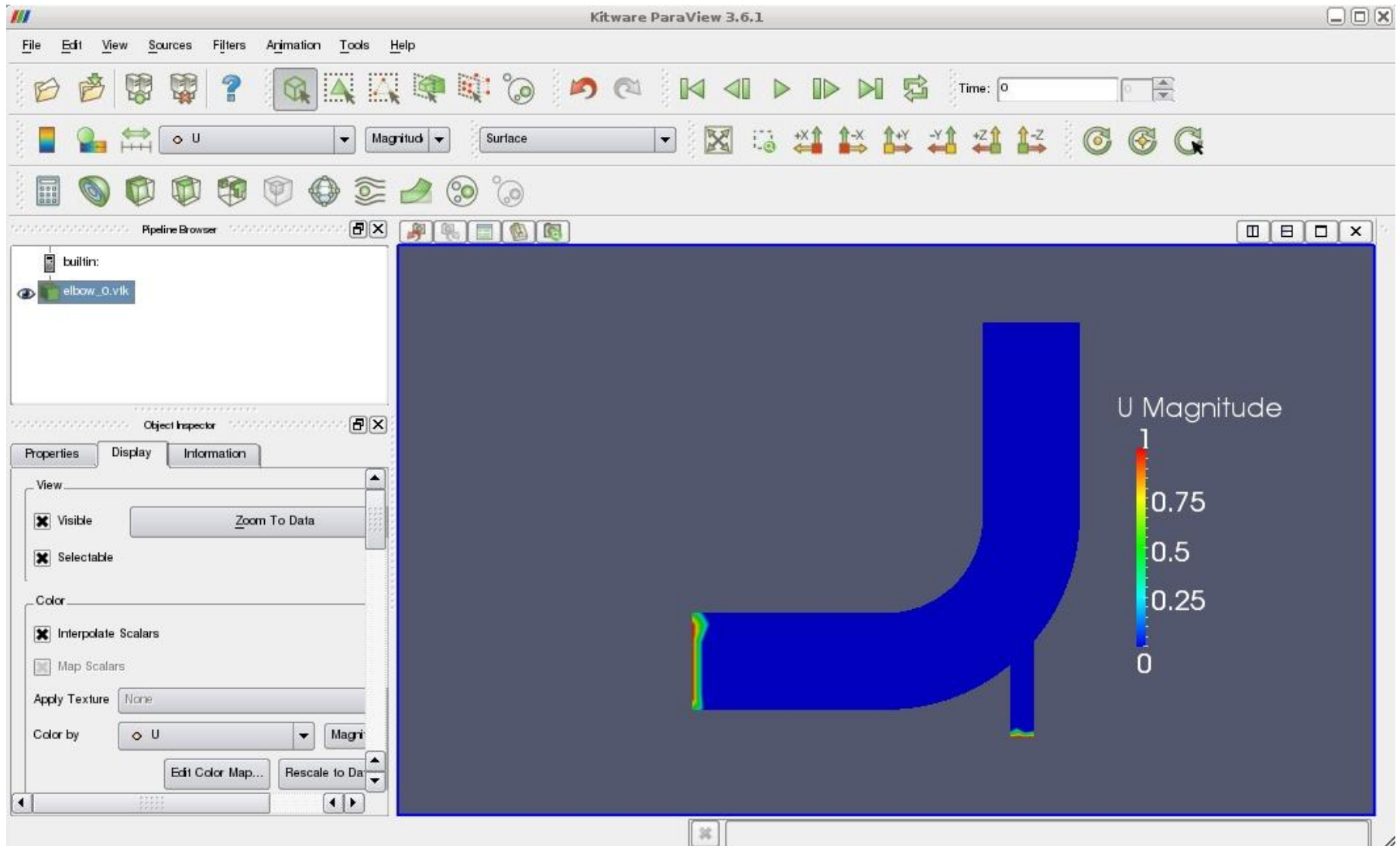
Visualización de geometría y malla

- Preprocesado:
 - foamToVTK -time 0
 - Transforma los ficheros de las condiciones iniciales a formato VTK
 - Paraview &
 - Visualiza la geometría, mallado y condiciones iniciales de la simulación

Visualización de geometría y malla

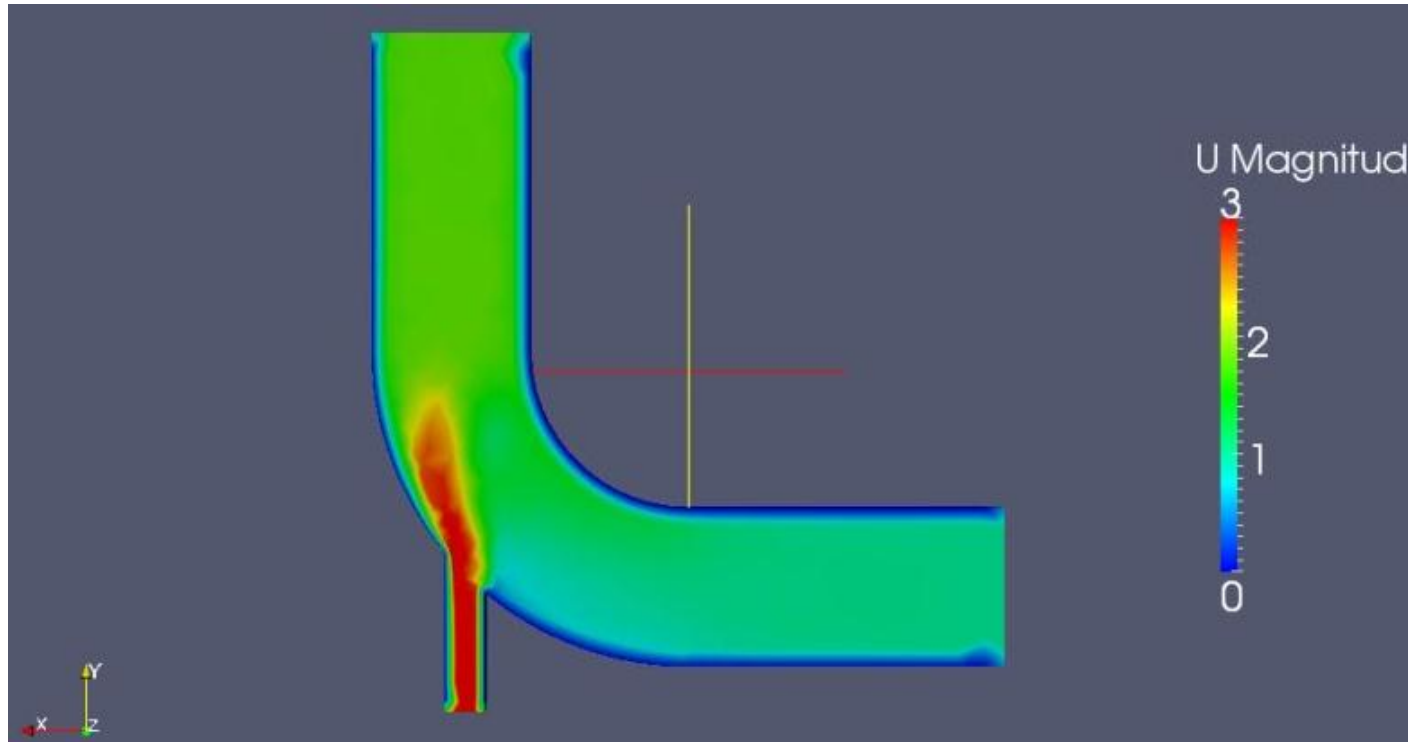


Visualización de geometría y malla



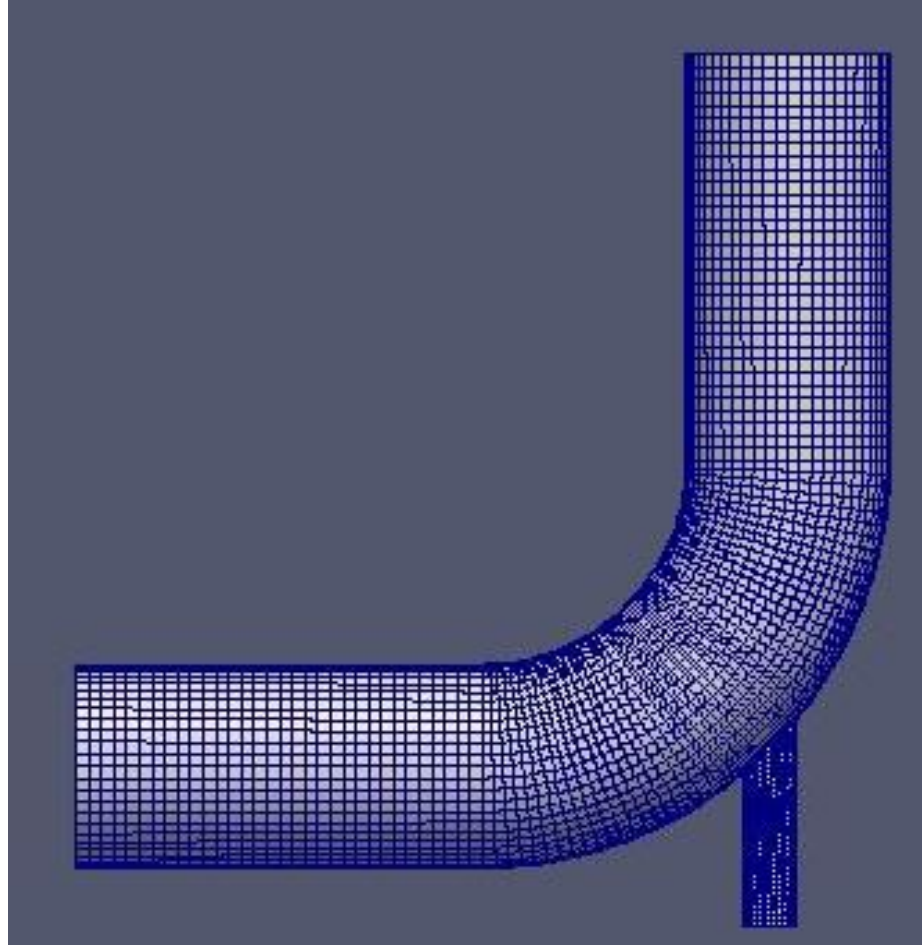
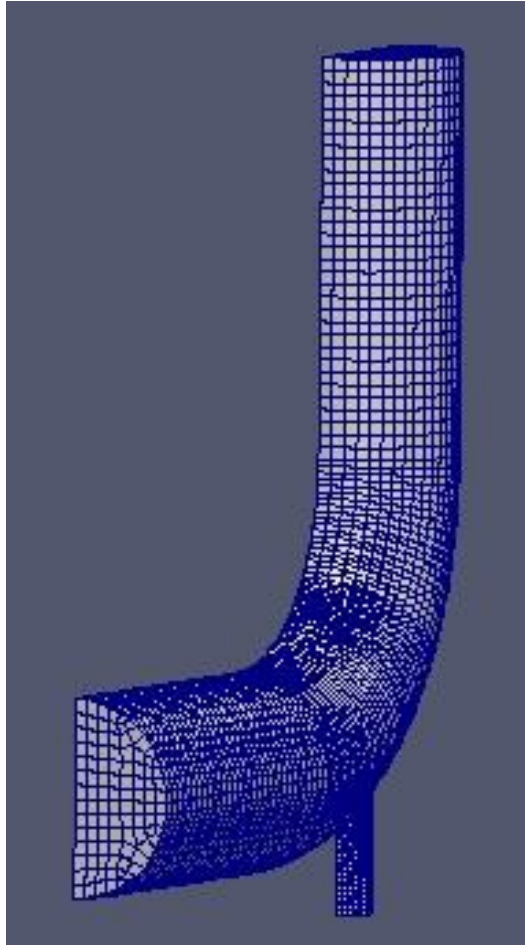
Ejecución y resultados del caso

- icoFoam > log
- foamToVTK -time 0:0.5
- Paraview &



Tutorial “elbow_3D”

- Ejemplo 3D, laminar, incompresible e isotermo



Archivo “polyMesh/boundary”

```
5
(
  wall
  {
    type            patch;
    nFaces          3630;
    startFace       38612;
  }
  symmetry
  {
    type            symmetryPlane;
    nFaces          2018;
    startFace       42242;
  }
  pressure-outlet-7
  {
    type            patch;
    nFaces          100;
    startFace       44260;
  }
  velocity-inlet-6
  {
    type            patch;
    nFaces          40;
    startFace       44360;
  }
  velocity-inlet-5
  {
    type            patch;
    nFaces          100;
    startFace       44400;
  }
)
```

“0/U”

```
dimensions      [0 1 -1 0 0 0];
internalField    uniform (0 0 0);
boundaryField
{
  wall
  {
    type          fixedValue;
    value          uniform (0 0 0);
  }
  symmetry
  {
    type          symmetryPlane;
  }
  velocity-inlet-5
  {
    type          fixedValue;
    value          uniform (0 3 0);
  }
  velocity-inlet-6
  {
    type          fixedValue;
    value          uniform (0 3 0);
  }
  pressure-outlet-7
  {
    type          zeroGradient;
  }
}
```

Resultados Tutorial “elbow_3D”

