Introducción a OpenFoam

Presentación general de OpenFoam

E. Martín[1], M. Meis[2], F. Varas[1,3]
1: Universidad de Vigo, 2: Vicus Desarrollos Tecnológicos, 3: Universidad Politécnica de Madrid

PROGRAMA COOPERACIÓN TRANSFRONTERIZA ESPAÑA – PORTUGAL COOPERAÇÃO TRANSFRONTEIRIÇA 2007 – 2013

0682_CLOUDPYME2_1_E

Unión Europea FEDER
Invertimos en su futuro

Organización del curso

Software Libre: OpenFoam (CFD)

- Horario y distribución de sesiones

	Miércoles, 18 de Junio	Jueves, 19 de Junio	Viernes, 20 de Junio
10:00 – 12:00	Introducción a OpenFOAM	Esquemas numéricos en OpenFOAM	Cálculo paralelo y postptocesado
12:00 – 14:00	Práctica	Práctica	Práctica
16:00 – 18:00	Modelos físicos OpenFOAM	Desarrollo de resolvedores en OpenFOAM	
18:00 – 20:00	Práctica	Práctica	

- Profesorado:
  - FERNANDO VARAS (UNIVERSIDAD DE VIGO)
  - ELENA MARTÍN (UNIVERSIDAD DE VIGO)
  - MARCOS MEIS (VICUS DESARROLLOS TECNOLÓGICOS Y UNIVERSIDAD DE VIGO)

Organización del curso

Software Libre: OpenFoam (CFD)

- Ejercicios prácticos
  - Flujos isotermos incompresibles I: análisis de una cavidad 2D y un codo 3D
  - Flujos isotermos incompresibles II: análisis de un perfil aerodinámico
  - Flujos con transferencia de calor I: flujo en un codo 3D con transferencia de calor
  - Flujos con transferencia de calor II: análisis térmico de una habitación
  - Flujos con frontera libre: análisis de rotura de una presa

Software Libre: OpenFoam (CFD)

- Enlaces de interés
  - Página web oficial de OpenFOAM:
    http://www.openfoam.com/
  - Wiki no oficial sobre OpenFOAM :
    http://openfoamwiki.net/index.php/Main_Page
  - PhD course in CFD with Open Source software :
    http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2009/
  - OpenFOAM-extend
    http://sourceforge.net/projects/openfoam-extend/
  - Texto "Lectures in Computational Fluid Dynamics of Incompressible Flow" por J. M. McDonough :
    http://www.engr.uky.edu/~acfd/me691-lctr-nts.pdf

Organización del curso

Software Libre: OpenFoam (CFD)

- Enlaces de interés
  - Notas del curso "Computational Hydraulics" por Dr David Apsley :

    http://personalpages.manchester.ac.uk/staff/david.d.apsley/lectures/comphydr/index.htm
  - CFD Online: Recursos relativos a dinámica de fluidos computacional :

    http://www.cfd-online.com/
  - Foro de discusión sobre OpenFOAM en CFD Online :

    http://www.cfd-online.com/Forums/openfoam/
  - Links on-line de modelos de turbulencia

    http://www.cfd-online.com/Wiki/Turbulence_modeling

¿Qué es OpenFoam?

- OPENFOAM® (Open Field Operation And Manipulation) es una herramienta de simulación numérica basada en esquemas de volúmenes finitos (básicamente, se trata de una biblioteca de utilidades para implementar un esquema de volúmenes finitos) especialmente orientada para la simulación numérica en mecánica de fluidos (cfd).
- Es un código de software libre (con acceso, por tanto, al código fuente) **programado en C++** y producido por opencfd ltd.
- OPENFOAM® es capaz de resolver tanto flujos sencillos como complejos (incluyendo flujos turbulentos, fenómenos de transferencia de calor y reacciones químicas), así como problemas en otros campos (como problemas electromagnéticos).
- Una de las principales ventajas, al margen de su flexibilidad, es su **capacidad para ser ejecutado en paralelo**. La implementación del cálculo distribuido (basada en el uso de mpi y directamente explotable por parte de procesadores multinúcleo, clúster o redes de ordenadores) no añade ninguna dificultad al usuario, por lo que la capacidad de cálculo del código está únicamente limitada por el hardware disponible

¿Qué es OpenFOAM?

- OpenFOAM is first 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 problem in continuum mechanics; and **utilities**, that are designed to perform tasks that involve data manipulation

- One of the strengths of OpenFOAM is that **new solvers and utilities can be created by its users** with some pre-requisite knowledge of the underlying method, physics and programming techniques involved
- OpenFOAM is **supplied with pre- and post-processing environments**. The interface to the pre- and post-processing are themselves OpenFOAM utilities, thereby ensuring consistent data handling across all environments

¿Qué es OpenFOAM?

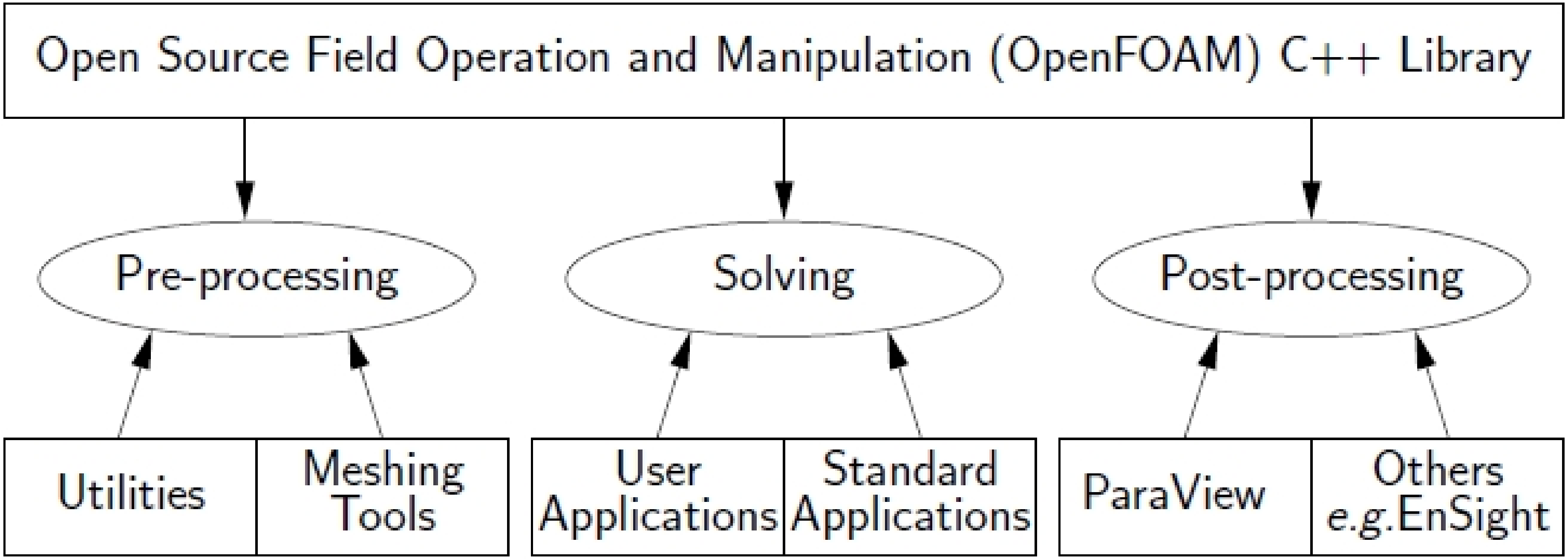


Figure 1.1: Overview of OpenFOAM structure.

Estructura de OpenFOAM

Structure of OpenFOAM

- **applications**: source files of all the executables:
  - solvers
  - utilities
  - bin
  - test
- **bin**: basic executable scripts.
- **doc**: pdf and Doxygen documentation.
  - Doxygen
  - Guides-a4
- **lib**: compiled libraries.
- **src**: source library files.
- **test**: library test source files.
- **tutorials**: tutorial cases.
- **wmake**: compiler settings.

Código fuente

- Comandos útiles para navergar en las fuentes de OpenFOAM:
  - *app* = $WM_PROJECT_DIR/applications
  - *sol* = $WM_PROJECT_DIR/applications/solvers
  - *util* = $WM_PROJECT_DIR/applications/utilities
  - *src* = $WM_PROJECT_DIR/src
- Variables de entorno:
  - $FOAM_APP = $WM_PROJECT_DIR/applications
  - $FOAM_SOLVERS = $WM_PROJECT_DIR/applications/solvers
  - $FOAM_UTILITIES = $WM_PROJECT_DIR/applications/utilities
  - $FOAM_SRC = $WM_PROJECT_DIR/src

CloudPYME

Estructura de archivos del software

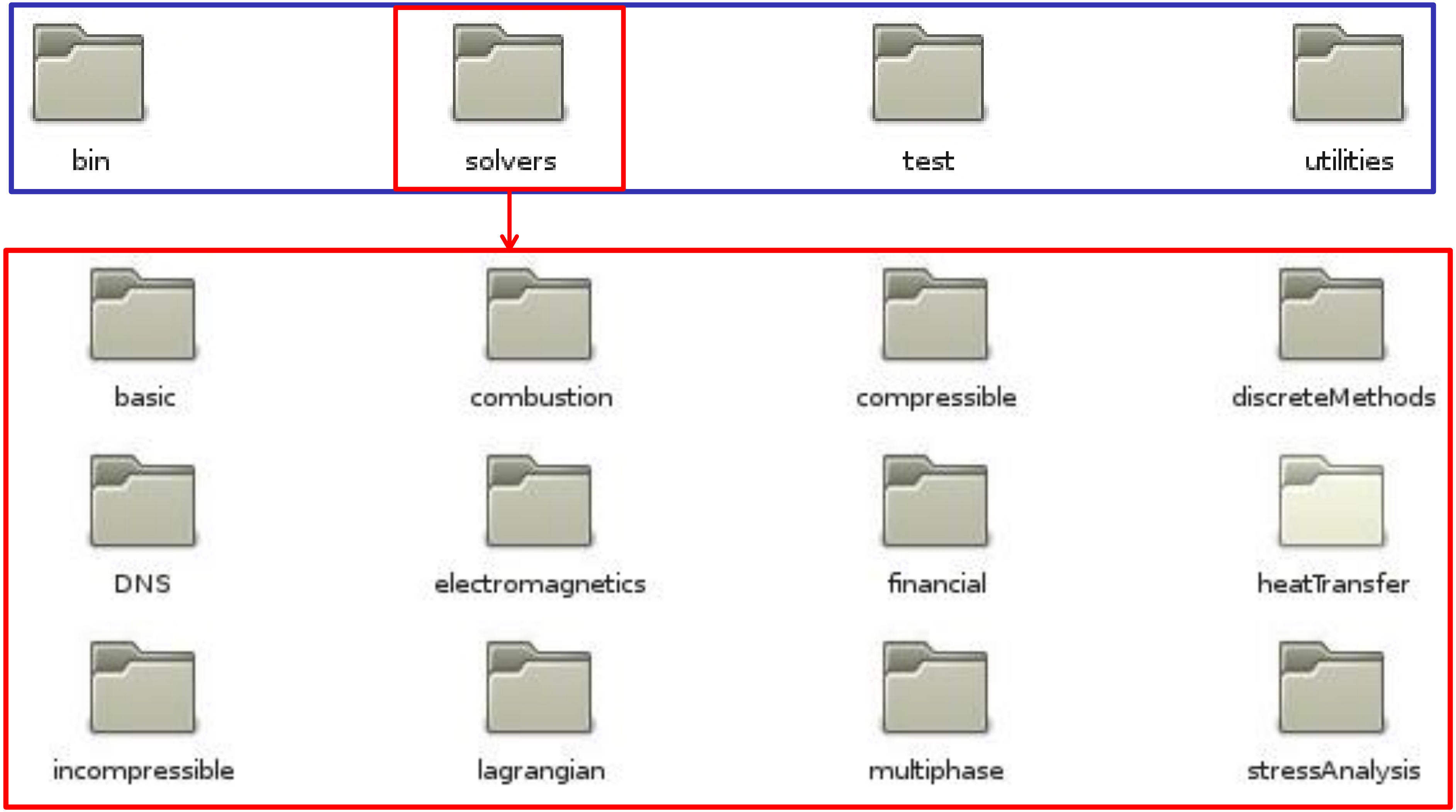
PROGRAMA COOPERACIÓN TRANSFRONTERIZA ESPAÑA – PORTUGAL COOPERAÇÃO TRANSFRONTEIRIÇA 2007 – 2013

0682_CLOUDPYME2_1_E

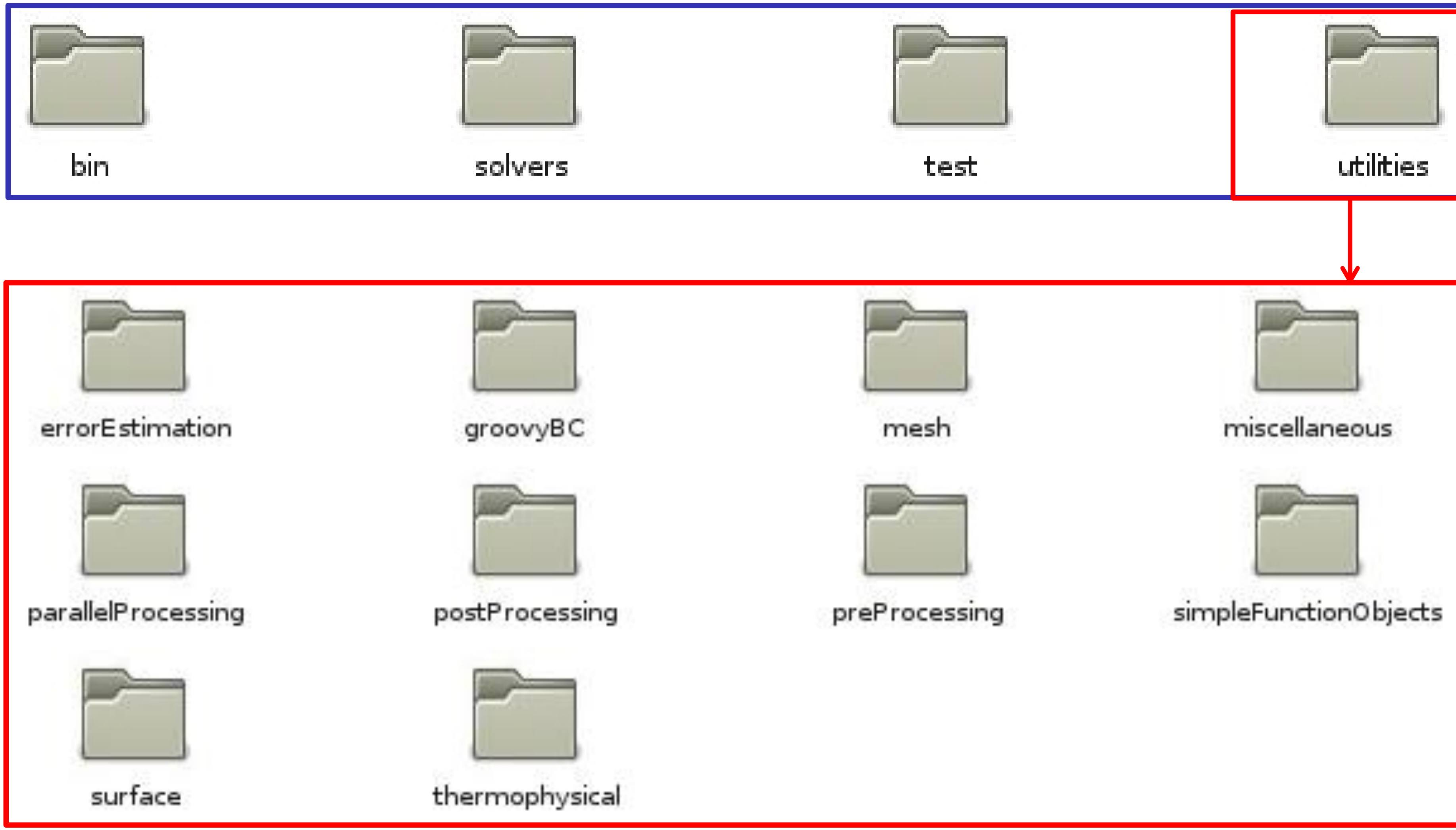
Unión Europea
FEDER
Invertimos en su futuro

CloudPYME

Estructura de archivos del software



Estructura de archivos del software



Estructura de archivos de un caso

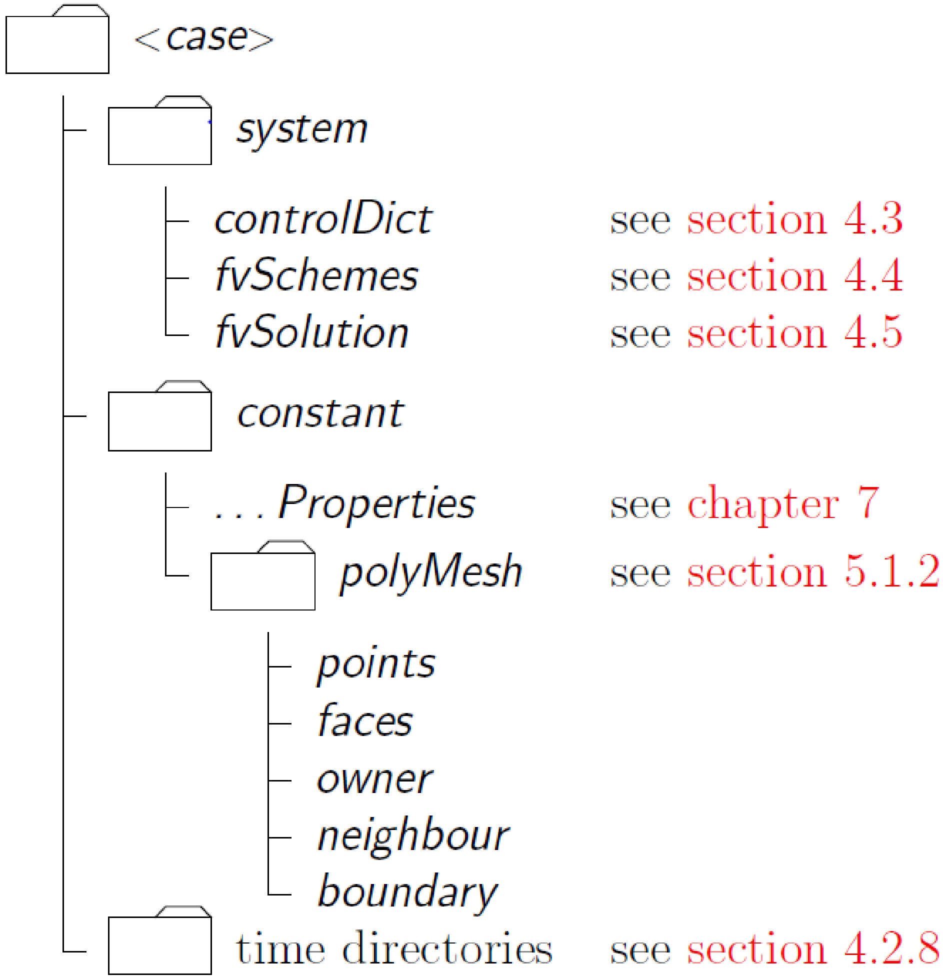


Figure 4.1: Case directory structure

Tutorial: Cavity

- Geometría 2D, incompresible, laminar, isotermo

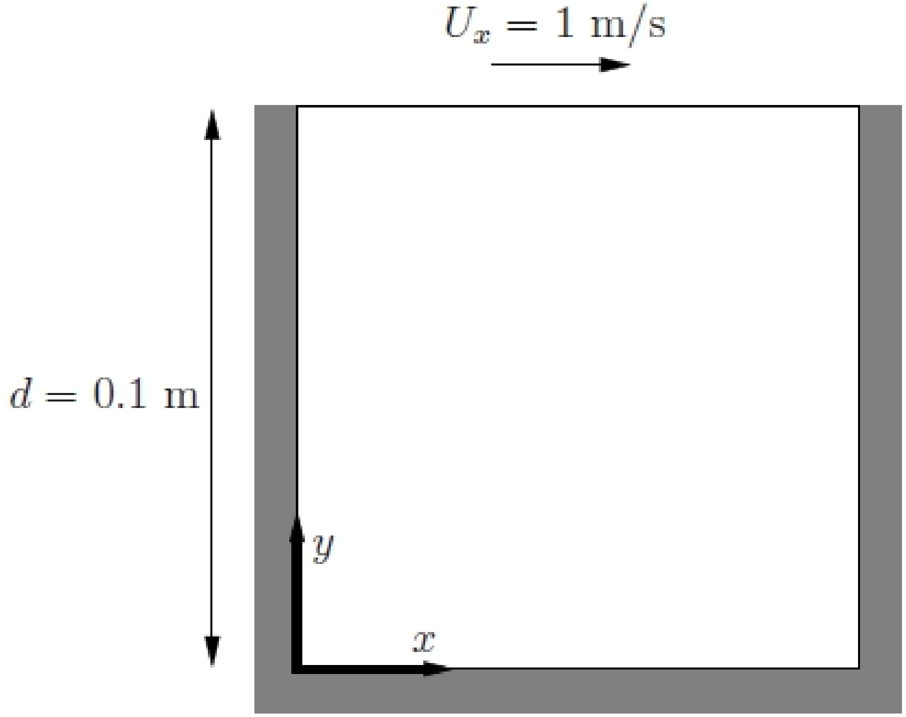
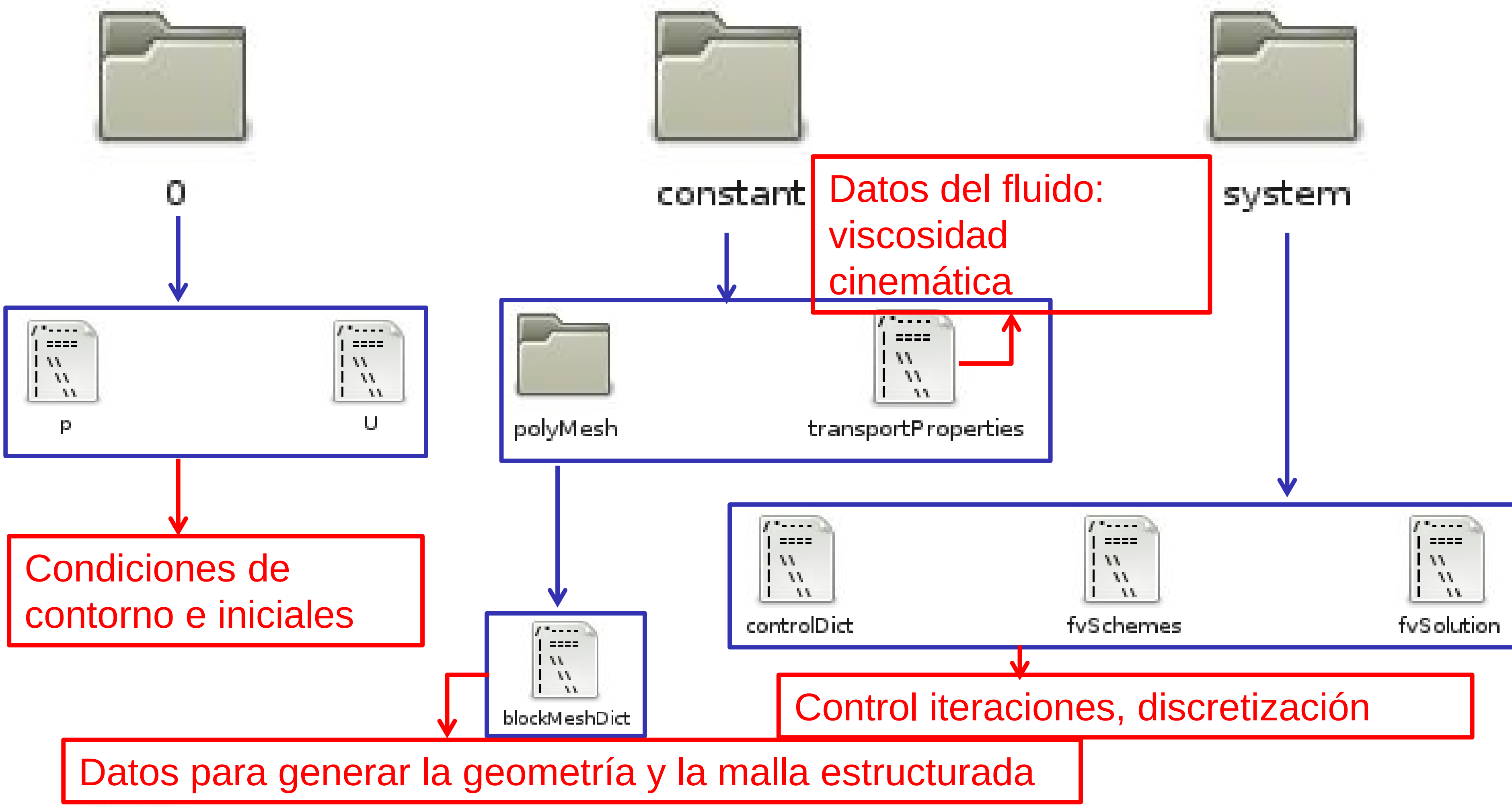


Figure 2.1: Geometry of the lid driven cavity.

Estructura de carpetas y ficheros

- Caso “cavity”:



CloudPYME

Fichero: “transportProperties”

transportProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | 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;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

// ************************************************************************* //
```

Cabecera OpenFoam

Viscosidad cinemática fluido

Dimensiones de las variables

CloudPYME

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

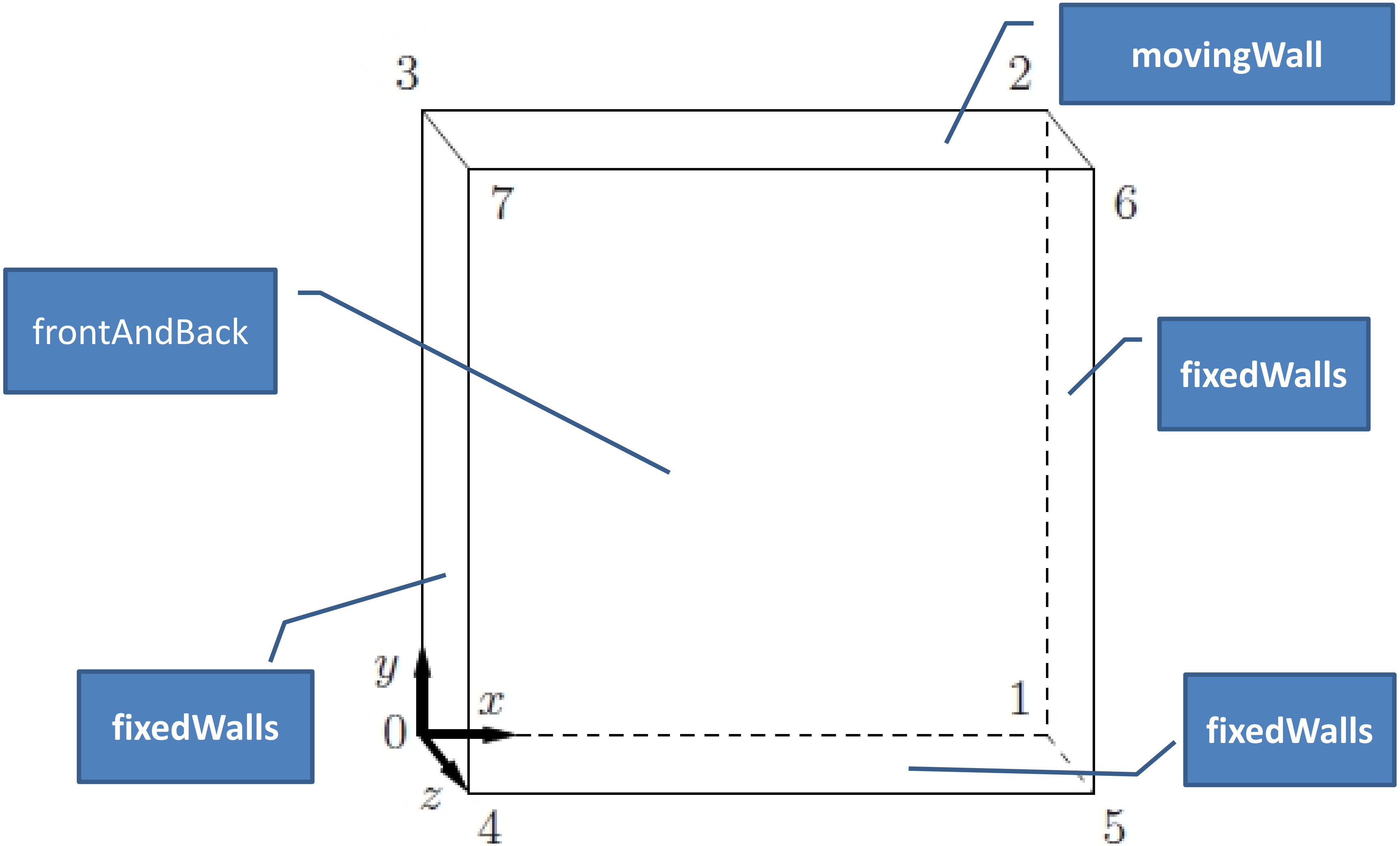


Figure 2.2: Block structure of the mesh for the cavity.

Fichero: “blockMeshDict”

blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | 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

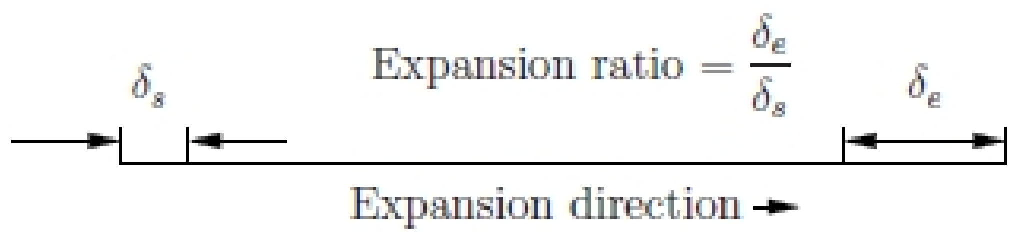


Figure 5.6: Mesh grading along a block edge

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
(
);

// ************************************************************************* //
```

Contornos/fronteras

Nombres de los contornos

Tipo de contorno

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

- Generación de malla: Ejecutar en un terminal en el directorio del caso:

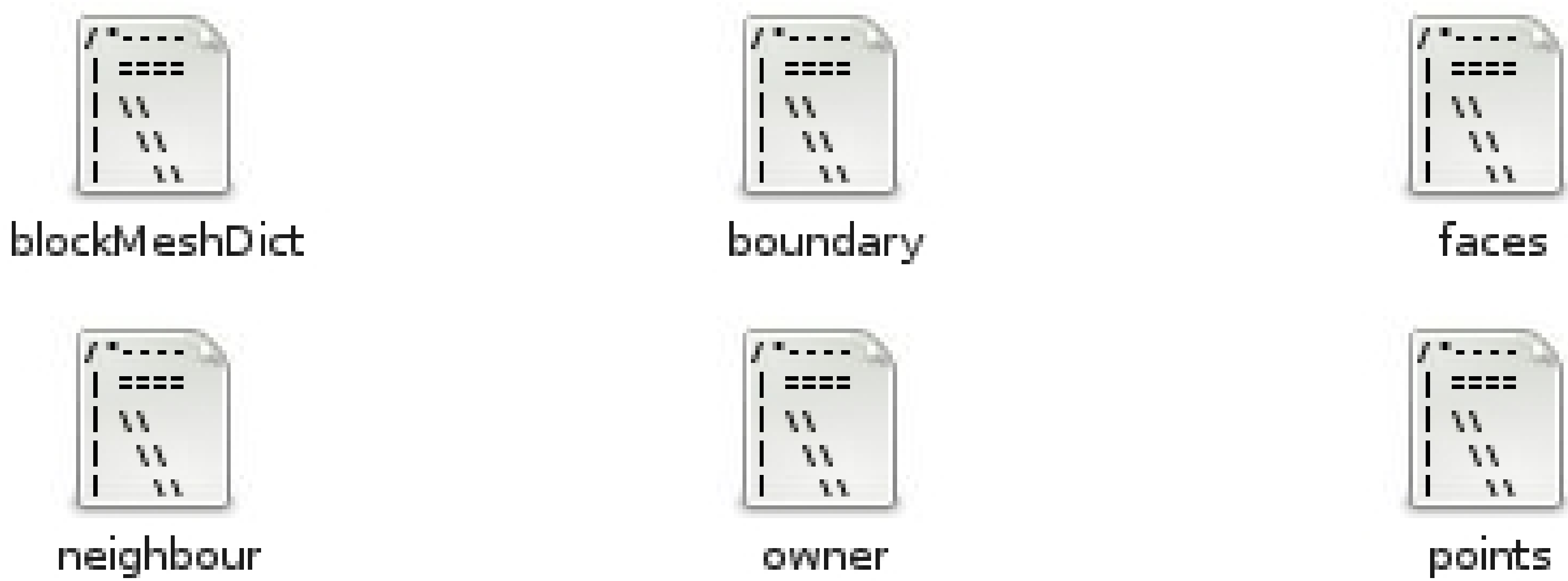
  startFoam –v 2.3.0

  blockMesh > log.blockMesh

  checkMesh > log.checkMesh

Generación de malla con blockMesh

➢ Archivos generados:



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**

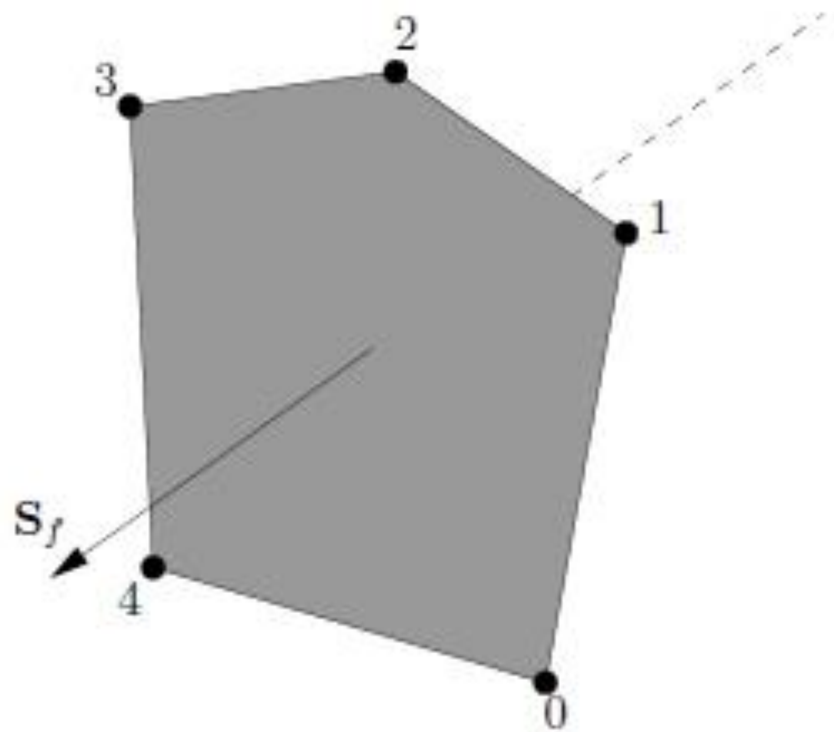


Figure 5.1: Face area vector from point numbering on the face

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

Base type	patch wall	symmetry empty wedge cyclic processor
Primitive type	fixedValue fixedGradient zeroGradient mixed directionMixed calculated	
Derived type	*e.g.*inletOutlet	

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 `value` 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 `inletDirection`	`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 fixedGradient pressure based on the atmospheric pressure gradient	—
Types derived from mixed		
inletOutlet	Switches $\mathbf{U}$ and p between fixedValue and zeroGradient depending on direction of $\mathbf{U}$	`inletValue`, `value`
outletInlet	Switches $\mathbf{U}$ and p between fixedValue and zeroGradient depending on direction of $\mathbf{U}$	`outletValue`, `value`
pressureInletOutletVelocity	Combination of pressureInletVelocity and inletOutlet	`value`
pressureDirected-InletOutletVelocity	Combination of pressureDirectedInletVelocity and inletOutlet	`value`, `inletDirection`
pressureTransmissive	Transmits supersonic pressure waves to surrounding pressure p_∞	`pInf`
supersonicFreeStream	Transmits oblique shocks to surroundings at p_∞, T_∞, $\mathbf{U}_\infty$	`pInf`, `TInf`, `UInf`
Other types		
slip	zeroGradient if ϕ is a scalar; if ϕ is a vector, normal component is fixedValue zero, tangential components are zeroGradient	—
partialSlip	Mixed zeroGradient/ slip condition depending on the `valueFraction`; = 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

CloudPYME

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         Gauss linear orthogonal;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         orthogonal;
}

fluxRequired
{
    default         no;
    p               ;
}
```

Scheme	Description
`Euler`	First order, bounded, implicit
`localEuler`	Local-time step, 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 $\nabla\bullet$
`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         Gauss linear orthogonal;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         orthogonal;
}

fluxRequired
{
    default         no;
    p               ;
}
```

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
`filteredLinear2`	Linear with filtering for high-frequency ringing
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          smoothSolver;
        smoother        symGaussSeidel;
        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
Diagonal solver for explicit systems	diagonal

†PCG for symmetric matrices, PBiCG for asymmetric

Table 4.12: Linear solvers.

Ejecución del caso “cavity”

- Recordatorio: Generación de malla:
  - Ejecutar en un terminal en el directorio del caso:

    blockMesh > log.blockMesh

    checkMesh > log.checkMesh

  - Ejecutar paraFoam para visualizar la malla antes de resolver y comprobar que todo es correcto:

    paraFoam &

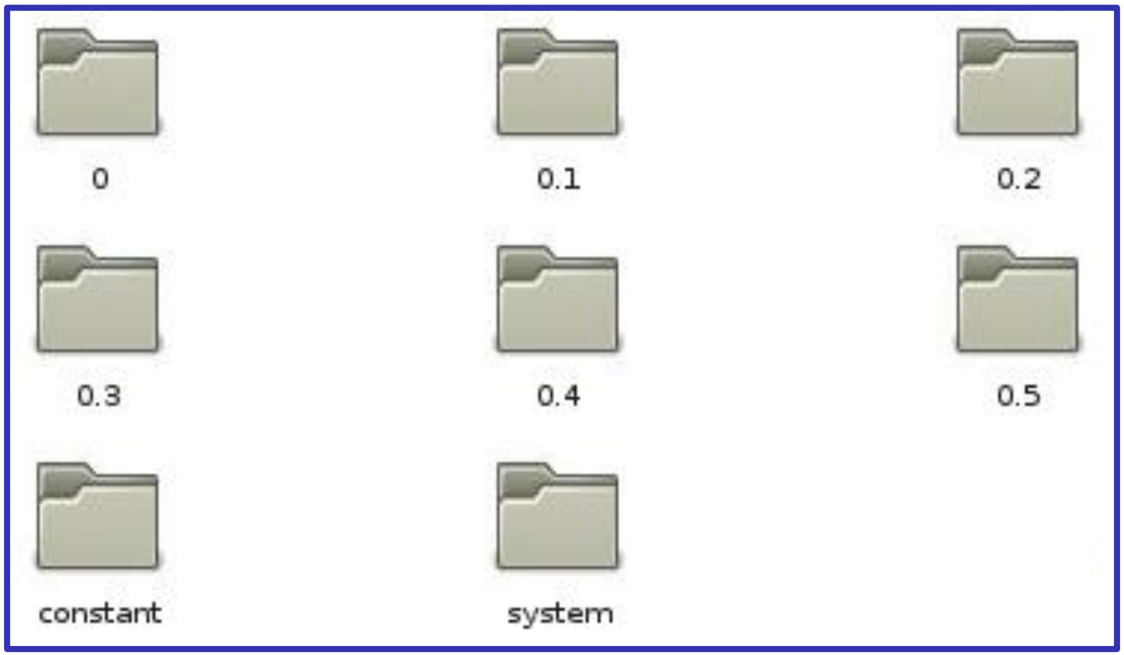
PROGRAMA COOPERACIÓN TRANSFRONTERIZA ESPAÑA – PORTUGAL COOPERAÇÃO TRANSFRONTEIRIÇA 2007 – 2013

Unión Europea FEDER

Invertimos en su futuro

Ejecución del caso "cavity"

- Ejecución del solver: Ejecutar en un terminal en el directorio del caso:
  - icoFoam > log.icoFoam ⟶ Aparecen archivos resultados para cada instante de tiempo



Ejecución del caso “cavity”

- Visualización de los residuos: Ejecutar en un terminal en el directorio del caso:

  gnuplot Residuals –

- Es necesario tener en el caso el fichero Residuals:

Postprocesado

- Postprocesado:
  - Alternativa 1:

    foamToVTK -time 0:0.5 > log.foamToVTK

    – Transforma los ficheros de resultados a formato VTK

    paraview &

    – Visualiza los resultados obtenidos en la simulación
  - Alternativa 2:

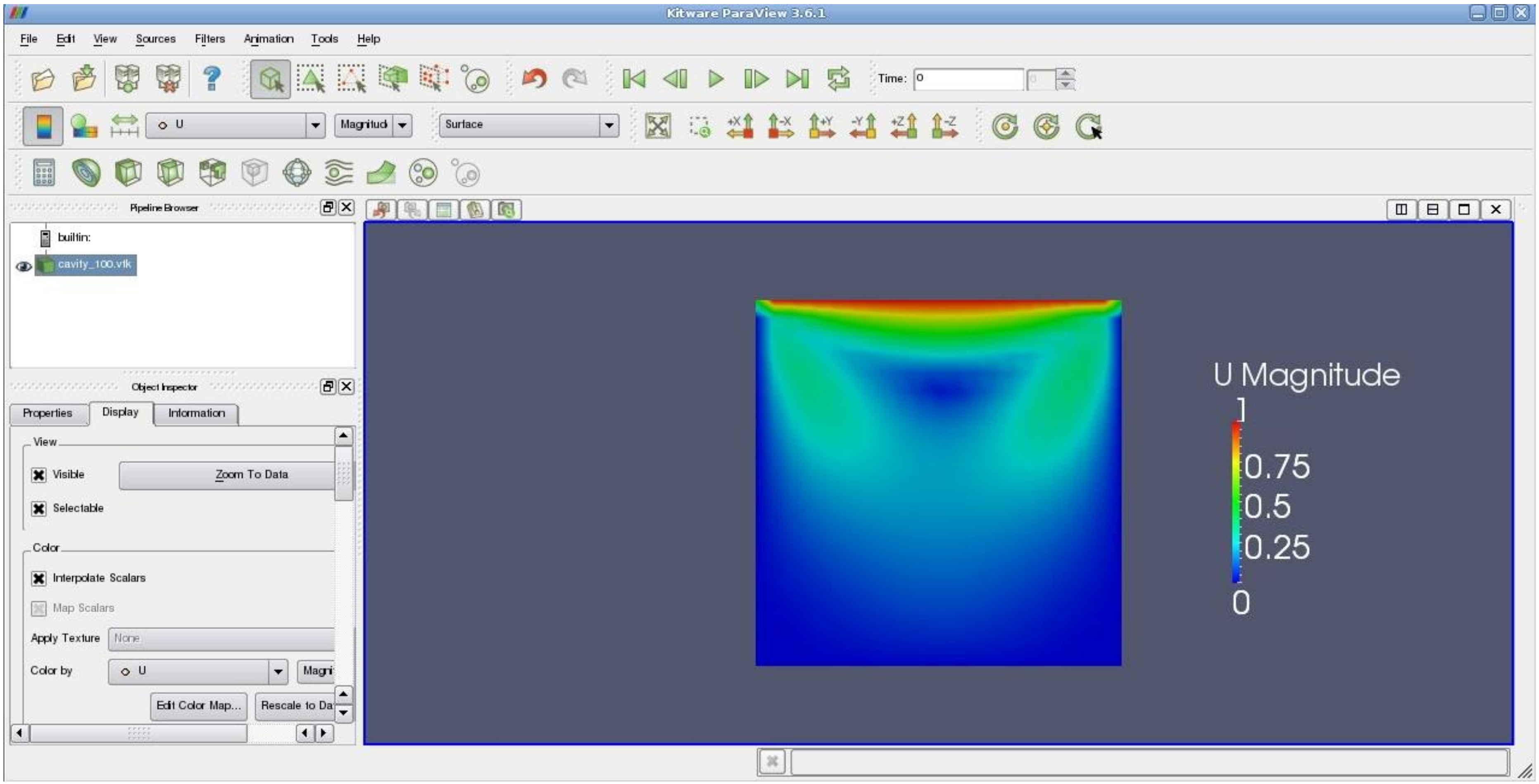
    paraFoam &

Scripts de ejecución

➢ Fichero script_exe:

```
//generación malla estructurada
blockMesh > log.blockMesh

//chequeo de malla
checkMesh > log.checkMesh

//ejecución del solver icoFoam
icoFoam > log.icoFoam

//visualización de residuos
gnuplot Residuals -

//generación de ficheros de resultados en formato VTK
foamToVTK > log.foamToVTK

//visualización de resultados con paraview
paraview &

//alternativa de visualización: sustituir los dos últimos pasos anteriores por la visualización en paraFoam:
paraFoam &
```

Postprocesado



Scripts automáticos de limpieza y ejecución del caso

Ficheros Allclean y Allrun:

➢ Fichero Allclean:

```
#!/bin/sh
cd ${0%/*} || exit 1    # run from this directory

# Source tutorial clean functions
. $WM_PROJECT_DIR/bin/tools/CleanFunctions

cleanCase
rm -rf constant/polyMesh/boundary
rm -rf VTK
rm -rf Residuals.png
```

- **Ejecución:**

  ./Allclean

Scripts automáticos de limpieza y ejecución del caso

Ficheros Allclean y Allrun:

➢ Fichero Allrun:

```
#!/bin/sh
cd ${0%/*} || exit 1    # run from this directory

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

runApplication blockMesh
runApplication checkMesh
runApplication icoFoam
runApplication gnuplot Residuals
runApplication foamToVTK
runApplication paraview
```

- **Ejecución:**

  ./Allrun

Ejercicio: generación de malla más fina

- **Tamaño de celdas del mallado en x e y: 1mm**
- **0.1 m/0.001 m = 100 celdas en x e y**
- **Necesario editar y modificar BlockMeshDict**

- **Volver a ejecutar el caso**