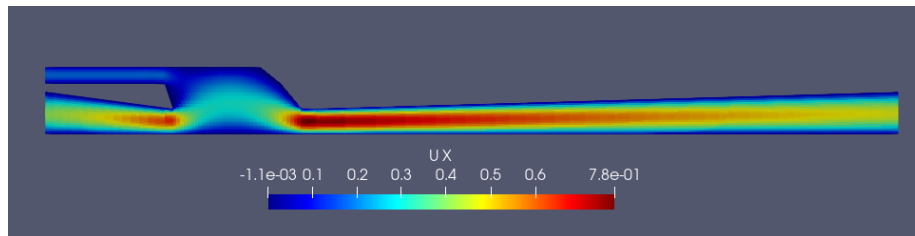




Informe N°1: Estudio de un caso RTD



Resumen:

Preparación de un caso para estudio RTD (Residence Time Distribution) con *OpenFOAM*. Definición de la malla con *blockMesh* y simulación con solvers *simpleFoam* (obtención campo de velocidades) y *scalarTransportFoam* (distribución del campo escalar).

Autor: Guillermo Rolle
Supervisor: Dr. Ing. César Pairetti

Diciembre 2019

Contents

1	Introducción	2
1.1	Información preliminar del problema	2
1.2	Preparación del caso	2
2	Mallado	3
3	Preparación de la simulación	6
3.1	Condiciones iniciales	6
3.2	Diccionarios en <i>system</i>	6
4	Simulación: <i>simpleFoam</i>	6
5	Preparación del caso RTD	7
5.1	Condiciones iniciales	8
5.2	Propiedades del fluido	8
5.3	Diccionarios en <i>system</i>	9
6	Simulación: <i>scalarTransportFoam</i>	9
7	Post proceso	10

1 Introducción

El objetivo de este trabajo es estudiar la calidad de mezclado en el tubo de salida de dos fluidos diferentes. En particular, en este trabajo se pensó un dispositivo de mezcla de agroquímicos en línea. Es decir, que la mezcla se realice en el momento en que se requiere en lugar de mezclar la cantidad suficiente de productos previo a salir al campo.

El problema de la técnica convencional radica en que la mezcla no es almacenable en depósito para un posterior uso obligándola a ser tratada adecuadamente para reducir la contaminación generada por estos químicos.

Los casos de simulación acá presentados se encuentran en el repositorio https://github.com/guillerolle/adscpcion_cfd. Descargar o clonar repositorio para obtener los archivos necesarios para las simulaciones.

Nota: los resultados de las simulaciones no se incluyen en el repositorio para evitar consumir espacio en el mismo. Sólo se encuentran los diccionarios y archivos importantes para realizar las simulaciones. Las simulaciones corren a cuenta de cada uno.

1.1 Información preliminar del problema



Figure 1: Geometría del problema

El fluido principal (agua) ingresa a alta presión por la entrada 1 succionando el fluido secundario (agroquímico) que ingresa por la entrada 2. La homogeneidad de la mezcla se evaluará en la salida del dispositivo.

En este trabajo se supondrá que el agroquímico tiene las mismas propiedades que el agua (densidad y viscosidad). De esta manera, se realizará una simulación para obtener el campo de velocidades para un único fluido en ambas entradas.

Luego, a partir del campo obtenido en estado estacionario, se hará otra simulación para estudiar la difusión y convección de un parámetro escalar que en este caso representará la concentración de agroquímico en cada punto del dominio de la simulación. Un valor de 1 indicará que el fluido es 100% agroquímico en ese punto y un valor de 0 indicará que es 100% agua (0% agroquímico).

1.2 Preparación del caso

Copiaremos un caso de los tutoriales para preparar el nuestro a partir de ese. Una vez abierta una terminal y cargado el ambiente de OpenFOAM, ejecutar:

```
$ run  
$ mkdir 01
```

Linux tips:
El símbolo '.' representa el directorio actual

```
$ cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/pitzDaily ./01
$ cd 01
```

En la carpeta de nuestro caso *01_rtd* tendremos la siguiente estructura de archivos:

```
01
|-- 0
|   |-- epsilon
|   |-- f
|   |-- k
|   |-- nut
|   |-- nuTilda
|   |-- omega
|   |-- p
|   |-- U
|   '-- v2
|-- constant
|   |-- transportProperties
|   '-- turbulenceProperties
'-- system
    |-- blockMeshDict
    |-- controlDict
    |-- fvSchemes
    |-- fvSolution
    '-- streamlines.
```

3 directories, 16 files

En primer lugar, la simulación que vamos a realizar es laminar por lo que tendremos que modificar las propiedades de turbulencia. Abrir `constant/turbulenceProperties` con el editor de textos de preferencia y eliminaremos los archivos `0/epsilon`, `0/f`, `0/k`, `0/nut`, `0/nuTilda`, `0/omega`, `0/v2`. Por ejemplo:

```
$ cd 0
$ rm epsilon f k nut nuTilda omega v2
$ cd ../constant
$ nano turbulenceProperties
[...]
simulationType laminar; <- MODIFICAR ESTA LINEA
[...]
```

Ver archivo completo: [constant/turbulenceProperties](#)

2 Mallado

Se utilizará el mallador *blockMesh* de OpenFOAM. En la figura 2 se puede ver la disposición de los vértices y bloques para crear el archivo *blockMeshDict*.

Linux tips:
man cmd
muestra el manual del comando *cmd*.
Por ejemplo,
man cd

Linux tips:
'..' representa el directorio superior

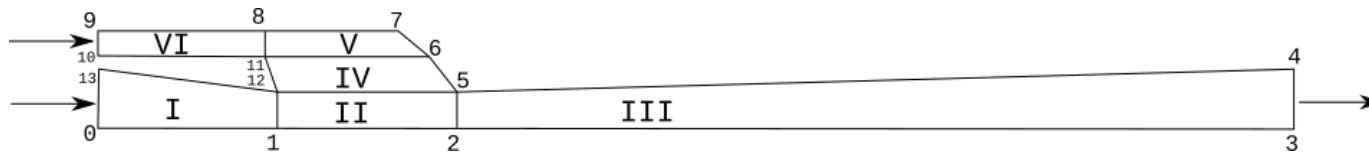


Figure 2: Vértices y bloques para blockMesh

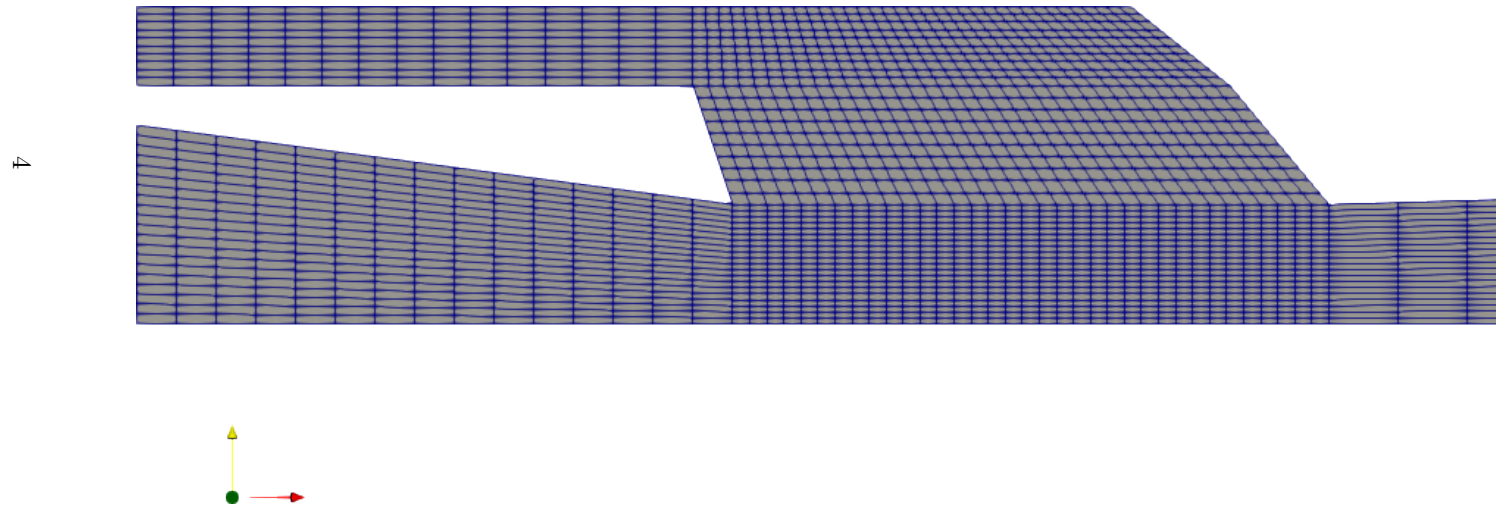


Figure 3: Malla creada con blockMesh (zoom)

Ver archivo completo: [system/blockMeshDict](#)

Una vez definido el archivo *system/blockMeshDict* ejecutar los siguientes comandos para crear la malla con *blockMesh*:

```
$ blockMesh
```

Linux tips:

touch <file>
crea un archivo
vacío con el
nombre *<file>*

Si el comando se ejecuta correctamente podemos mirar la malla en *ParaView* (3). Observar la ortogonalidad de la malla.

```
$ touch rtd.foam
```

```
$ paraview rtd.foam &
```

O, equivalentemente:

```
$ paraFoam
```

Linux tips:

El símbolo *&* al
final de la línea
permite seguir
utilizando la
terminal actual

Antes de continuar con el caso es conveniente observar la salida del comando *checkMesh* para comprobar la calidad del mallado. Por ejemplo, una malla "no-ortogonal" puede resultar en fallas en la resolución numérica del caso. Ver el siguiente documento sobre [calidad de mallado](#).

```
$ checkMesh
```

```
[...]
```

```
Checking geometry...
```

```
Overall domain bounding box (0 0 -0.001) (0.1 0.008 0)
```

```
Mesh has 2 geometric (non-empty/wedge) directions (1 1 0)
```

```
Mesh has 2 solution (non-empty) directions (1 1 0)
```

```
All edges aligned with or perpendicular to non-empty directions.
```

```
Boundary openness (7.19353e-18 -4.50911e-17 -3.44031e-15) OK.
```

```
Max cell openness = 2.13343e-16 OK.
```

```
Max aspect ratio = 11.4754 OK.
```

```
Minimum face area = 6.35714e-08. Maximum face area = 1.75071e-06.
```

```
Face area magnitudes OK.
```

```
Min volume = 6.35714e-11. Max volume = 4.35313e-10. Total volume =
```

```
4.8025e-07. Cell volumes OK.
```

```
Mesh non-orthogonality Max: 50.9374 average: 16.2357 <- CUANTO MAS
```

```
ALTO EL VALOR MAXIMO DE NO-ORTOGONALIDAD, MAYORES SERAN LOS
```

```
ERRORES NUMERICOS DEBIDOS LA MALLA!!
```

```
Non-orthogonality check OK.
```

```
Face pyramids OK.
```

```
Max skewness = 0.928968 OK.
```

```
Coupled point location match (average 0) OK.
```

```
Mesh OK.
```

```
End
```

OpenFOAM tips:

El comando
paraFoam
ejecuta
esencialmente
los dos
comandos
anteriores para
abrir ParaView

3 Preparación de la simulación

Después de crear la malla hay que ajustar los archivos `0/U`, `0/p`, `system/fvSchemes`, `system/fvSolutions` y `system/controlDict`.

3.1 Condiciones iniciales

El campo de velocidades podemos definirlo como nulo en todo el dominio porque impondremos condiciones de presión. En las entradas y salidas utilizamos la condición *zeroGradient*.

Ver archivo completo de condición inicial de velocidad: [0/U](#)

En cuanto a las presiones, impondremos 300MPa en la entrada de agua y un valor de 0 en la salida. En la entrada de agroquímico nos interesa que la presión sea mayor que la presión del agua en el empalme de los tubos. De esta manera permitiría al agroquímico a ser succionado por el flujo de agua.

Gracias a la tobera en la entrada, existe una depresión en la zona de unión. Impondremos un valor de 300MPa para asegurarnos que la presión sea mayor en ese lugar.

Ver archivo completo de condición inicial de presión: [0/p](#)

3.2 Diccionarios en system

Estos diccionarios contienen información sobre los solvers, pasos de tiempo, métodos numéricos, etc. El más importante es [system/controlDict](#). Cabe destacar que *simpleFoam* es un solver estacionario, así que lo que se define como tiempo en este diccionario, es en realidad un número que se le asigna a cada iteración. Se pone un deltaT igual a 1 para facilitar la lectura de los archivos generados.

```
[...]
deltaT 1; // Solamente sirve para enumerar las iteraciones. No afecta
         a la simulacion
[...]
```

Los otros dos contienen información sobre los esquemas numéricos y tolerancias.

Ver archivo: [system/fvSchemes](#)

Ver archivo: [system/fvSolution](#)

4 Simulación: *simpleFoam*

Ahora estamos listos para ejecutar la simulación primer simulación para obtener el campo de velocidades en el dispositivo. En el directorio principal del caso:

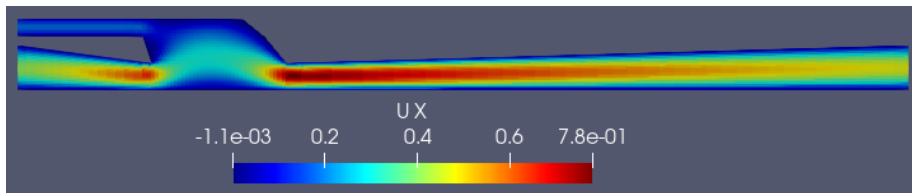
```
$ simpleFoam | tee log
```

Linux tips:

El símbolo | toma la salida del primer comando (izquierda) y la usa como entrada del segundo (derecha)

Si todo se ejecuta correctamente podremos abrirlo con ParaView (o clickear en 'reload' si ya está abierto):

```
$ paraview rtd.foam &
```



Linux tips:

El comando *tee* *file* guarda lo que recibe como entrada en el archivo *file*. Ver *man tee*

Figure 4: Campo de velocidad

En este caso nos interesaría saber si el flujo de agroquímico efectivamente es expulsado por el tubo de salida. Podemos comprobar esto, sabiendo que la simulación es laminar, con un filtro en ParaView. Presionar **Ctrl + Space** y buscar el filtro *StreamTracer*. Podemos ubicar el filtro alineado con el eje Y en el tubo de salida y mostraremos unas 50 líneas de traza.

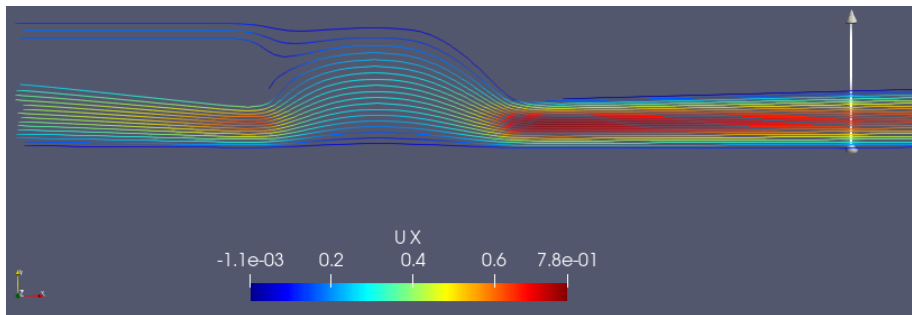


Figure 5: Líneas de traza

Las líneas de traza correspondientes al tubo superior indican que, efectivamente, el agroquímico es succionado en el dispositivo.

Podemos cargar el archivo de estado de ParaView *rtd_pU.pvsm* y nos abrirá el entorno de ParaView con las gráficas y filtros correspondientes. La simulación debe estar realizada anteriormente.

5 Preparación del caso RTD

Como base para la simulación del transporte escalar copiaremos otro de los tutoriales en una carpeta nueva. En la carpeta principal del caso:

```
$ mkdir RTD
$ cp -r $FOAM_TUTORIALS/basic/scalarTransportFoam/pitzDaily ./RTD
$ cd RTD
```


Tendremos la siguiente estructura de archivos:

```
RTD
|-- 0
|   |-- T
|   '-- U
|-- constant
|   '-- transportProperties
'-- system
    |-- blockMeshDict
    |-- controlDict
    |-- fvSchemes
    '-- fvSolution
```

3 directories, 7 files

Copiamos la malla del caso anterior en el caso RTD.

```
$ cp -r ../constant/polyMesh ./constant
```

Como antes, podemos comprobar que la malla esté creada correctamente ejecutando

```
$ paraFoam
```

5.1 Condiciones iniciales

Para el campo de velocidad usaremos el obtenido en la simulación anterior. En este caso, la última iteración es la 110. Por lo tanto

```
$ cp ../110/U ./0
```

La concentración de agroquímico será representada en este caso por la variable T . Un valor de 1 indicaría que el fluido es agroquímico puro y un valor de 0 indicaría que el fluido en ese lugar es únicamente agua.

Entonces, para el *inlet_one* pondremos una condición tipo *fixedValue* con valor 0 y para *inlet_two* un valor de 1. Para el *outlet* pondremos condición *zeroGradient*.

Por ejemplo, para *inlet_one*:

```
inlet_one
{
    type    fixedValue;
    value   uniform 0;
}
```

Ver el archivo completo: [RTD/0/T](#)

5.2 Propiedades del fluido

En la carpeta *constant* debemos definir un valor para el coeficiente de difusión del agroquímico. Propondremos un valor de 0.00005 (ojo! corroborar con el repositorio el valor más actualizado):

```
...
DT 0.00005; // Para OpenFOAM-plus (v1906)
DT DT [0 2 -1 0 0 0 0] 0.00005; // Para OpenFOAM-dev (7)
...
```

Ver el archivo completo: [RTD/constant/transportProperties](#)

OpenFOAM

tips:

En la versión [dev](#) es necesario definir las unidades de la difusividad mientras que en la [plus](#) no.

5.3 Diccionarios en *system*

Nuevamente, estos diccionarios contienen información sobre los solvers, pasos de tiempo, métodos numéricos, etc. El más importante es [RTD/system/controlDict](#). Esta vez, como el solver *scalarTransportFoam* es un solver transitorio, las variables de tiempo sí representan tiempos y no iteraciones.

```
...
endTime 2; // en segundos ambas. Chequear ultima version en repositorio
deltaT 0.001;
...
```

Los otros dos contienen información sobre los esquemas numéricos y tolerancias.

Ver archivo: [RTD/system/fvSchemes](#)

Ver archivo: [RTD/system/fvSolution](#)

6 Simulación: *scalarTransportFoam*

Ya tenemos el caso RTD preparado y podemos ejecutar la simulación con el solver *scalarTransportFoam*. Cabe destacar que este es un solver transitorio a diferencia de *simpleFoam* que es estacionario.

```
$ scalarTransportFoam | tee log
$ paraFoam
```

Podemos cargar el archivo de estado de ParaView *RTD.pvsm* para abrir las gráficas y filtros correspondientes.

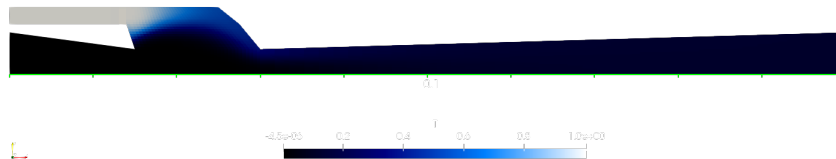


Figure 6: Concentración de agroquímico ($t = 2s$)

7 Post proceso

Nos interesa conocer la concentración promedio en toda el área de salida del dispositivo. En la carpeta *system* del caso RTD copiaremos el archivo *patchAverage* de la carpeta *etc* de OpenFOAM. En la carpeta principal RTD ejecutar:

```
$ cp $FOAM_ETC/caseDicts/postProcessing/surfaceFieldValue/patchAverage ./
  system
$ nano system/patchAverage
```

En este archivo debemos modificar los campos *patchName* y *field names* con nuestros valores de interés. Los cambios son los siguientes:

```
...
name outlet;
fields (T);
...
```

Ver el archivo completo: [RTD/system/patchAverage](#)

Podemos ejecutar el post-proceso con el siguiente comando.

```
$ postProcess -func 'patchAverage'
```

Esto nos va a crear una carpeta *postProcessing/patchAverage/0* con el archivo *surfaceFieldValue_0.dat*. Cambiamos de directorio para analizar más fácilmente estos datos:

```
$ cd postProcessing/patchAverage/0
```

Ese archivo contiene el valor de *areaAverage(T)* para cada instante de tiempo dispuestos en columna. Para ver el contenido del archivo en la terminal ejecutar:

```
$ cat surfaceFieldValue_0.dat
```

Este archivo podemos leerlo directamente en *gnuplot* para realizar una gráfica de la concentración en función del tiempo.

```
$ gnuplot
gnuplot> plot 'surfaceFieldValue\_0.dat' w l title "outlet"
gnuplot> set xlabel "Tiempo [s]"
gnuplot> set ylabel "Concentracion []"
gnuplot> replot
```

Observamos que la concentración en la salida converge a un valor de aproximadamente 15%.

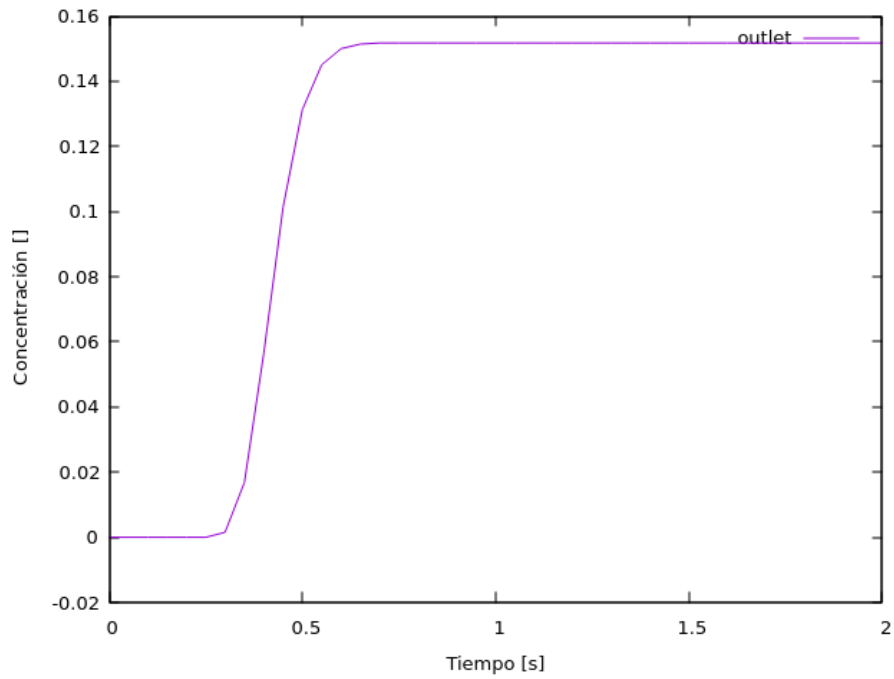


Figure 7: Concentración en la salida en función del tiempo

Podemos también observar el perfil de concentración sobre la salida para comprobar la homogeneidad de la mezcla. Esto se ve en rojo en la figura 8.

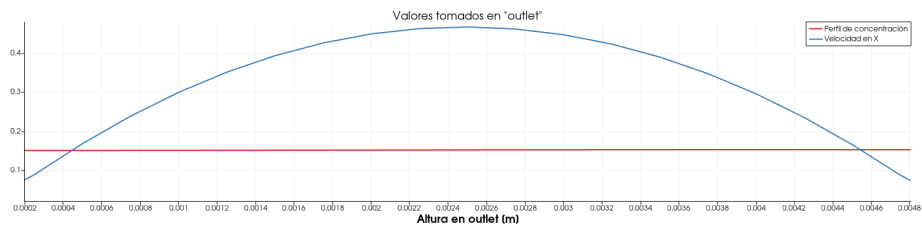


Figure 8: Distribución de valores sobre toda la altura del outlet