

# **SIMULACIÓN DE TRANSITORIOS CON SOLVER PIMPLEFOAM**

## **TRANSIENT SIMULATION WITH PIMPLEFOAM SOLVER**

**Guillermo Rolle**

**Palabras clave:** CFD, OpenFOAM, pimpleFoam

**Resumen.** Este documento provee una guía para realizar una simulación de estado transitorio en OpenFOAM utilizando el solver pimpleFoam. La malla es importada de un informe anterior al igual que las condiciones de contorno e iniciales. El informe provee también una breve explicación sobre técnicas de post-proceso y conclusiones sobre el método implementado.

**Keywords:** CFD, OpenFOAM, pimpleFoam

**Abstract.** This document provides a guide to perform a transient state simulation in OpenFOAM using the pimpleFoam solver. Mesh is imported from a previous report like boundary and initial conditions. The report also provides a brief explanation of post-processing techniques and conclusions about the implemented method.

## 1. INTRODUCCIÓN

En el [informe anterior](#) se hicieron simulaciones en 3D sobre el mezclador de agroquímicos. Los campos de velocidad y presión fueron obtenidos con un solver estacionario. Se había comprobado que el solver no era capaz de reducir los residuales al valor requerido por no poder obtenerse una solución estacionaria. En este informe se simula el mismo caso pero con el solver transitorio [pimpleFoam](#). Se usa una copia de la malla previa y el informe se basará exclusivamente en las modificaciones del caso para ejecutar el nuevo solver.

Figura 1: Caso 3D en cuestión

Este informe es parte de la [serie de informes](#) basados en un mezclador de agroquímicos en línea. El material correspondiente para este caso se encuentra en el [repositorio online](#).

## 2. PREPARACIÓN DEL CASO

Usamos el [caso 3D anterior](#) como base para preparar el nuevo caso. Copiamos además el [meshCase](#) para generar la malla. Notar que no modificaremos los parámetros de mallado pero aprovecharemos el script [Allrun](#) que requiere de la existencia de este caso.

No son muchas las modificaciones que hay que hacer para pasar de una simulación con `simpleFoam` a una con `pimpleFoam`.

Primero, en el archivo [controlDict](#) debemos llenar el campo `application` con `pimpleFoam`. Además, en el solver estacionario el campo `deltaT` representa el paso en la numeración de iteraciones y no representa más que un número de iteración. En un solver transitorio en cambio, sí es importante ya que representa el paso de tiempo entre una iteración y la siguiente. Lo mismo aplica para el campo `endTime`. Para la simulación de transitorio simularemos un tiempo total de 2 segundos con paso de tiempo de 1ms como se ve en la [figura 2](#). Por último, aumentamos la frecuencia de escritura para poder analizar mejor las variaciones en el tiempo.

<code>application</code>	<code>simpleFoam;</code>	→	←	<code>application</code>	<code>pimpleFoam;</code>
<code>startFrom</code>	<code>startTime;</code>			<code>startFrom</code>	<code>startTime;</code>
<code>startTime</code>	<code>0;</code>			<code>startTime</code>	<code>0;</code>
<code>stopAt</code>	<code>endTime;</code>			<code>stopAt</code>	<code>endTime;</code>
<code>deltaT</code>	<code>1;</code>	→	←	<code>deltaT</code>	<code>0.001;</code>
<code>endTime</code>	<code>2000;</code>	→	←	<code>endTime</code>	<code>2;</code>
<code>writeControl</code>	<code>timeStep;</code>			<code>writeControl</code>	<code>timeStep;</code>
<code>writeInterval</code>	<code>100.0;</code>	→	←	<code>writeInterval</code>	<code>5;</code>

Figura 2: cambios en `controlDict`: caso estacionario (izquierda) vs caso transitorio (derecha)

En el archivo [fvSchemes](#) modificamos el campo `ddtSchemes` y agregamos una línea a `gradSchemes`. Los cambios se muestran en la [figura 3](#).

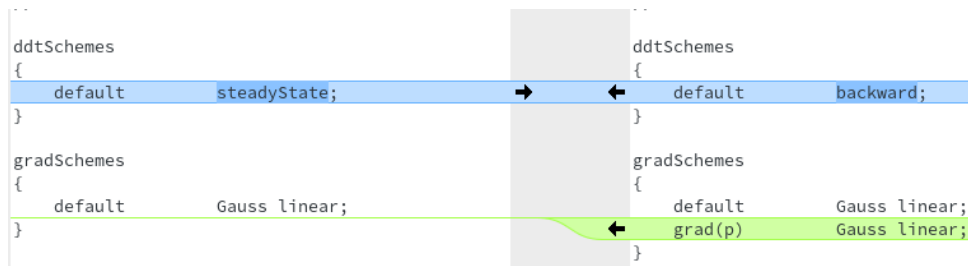


Figura 3: cambios en fvSchemes: caso estacionario (izquierda) vs caso transitorio (derecha)

Por último, modificamos las tolerancias de los solvers para p y U en el archivo **fvSolution**. Además, agregamos los modificadores para el campo PIMPLE. los cambios se muestran en la figura 4

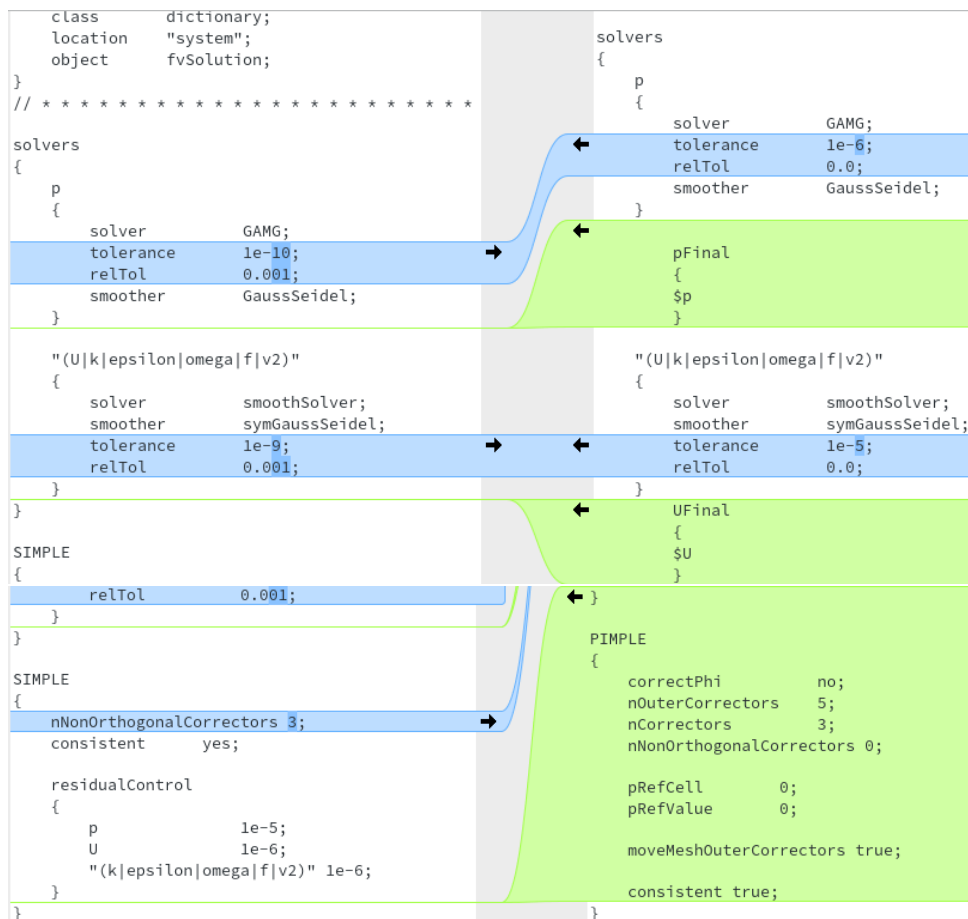


Figura 4: cambios en fvSolution: caso estacionario (izquierda) vs caso transitorio (derecha)

### 3. SIMULACIÓN SIMPLEFOAM

Una vez creada la malla y definidas las propiedades del solver (modelo estacionario, flujo incompresible, agua, etc.) abrimos la sección **CfdSolver** en **FreeCAD** y escribimos el caso. Antes de ejecutarlo, debemos realizar algunas modificaciones en los archivos *fvSolution* y *fvSchemes*. En el repositorio se encuentran los archivos *fvSolution.bkp* y *fvSchemes.bkp*. Reemplazar los anteriores por éstos últimos para mejorar la convergencia de la simulación.

Se sugiere utilizar una utilidad como **Meld** para comparar archivos y/o carpetas. En la figura 5 se muestra esta utilidad comparando los archivos mencionados.

Figura 5: Comparación archivos fvSolution.bkp (izquierda) con fvSolution (derecha)

Volviendo a FreeCAD, podremos ejecutar la simulación con *RUN*. Observar que la evolución de los residuales (figura 6) comienza a oscilar sobre un valor aproximadamente constante a partir de la iteración número 1000 sin llegar nunca a las tolerancias definidas en *system/fvSolution*.

Figura 6: Evolución de residuales con simpleFoam

Sin necesidad de esperar a que termine de correr la simulación, podemos ver la evolución del perfil en ParaView. En la figura 7 se muestra el campo de velocidades para una iteración bastante avanzada de la simulación (1700). Al comparar con pasos de tiempo consecutivos, no se aprecian grandes diferencias en los campos de velocidad o presión, por lo que asumiremos que la simulación fue satisfactoria. FreeCAD descompone el caso en varios procesadores, por lo que para ver los resultados en ParaView debemos seleccionar "DecomposedCase".

Figura 7: Campo de velocidad U obtenido

#### 3.1. Reconstrucción del caso

Ya sabemos de los informes anteriores que necesitamos copiar el campo de presión y velocidad obtenidos con esta simulación en otro directorio que contenga el caso de la simulación RTD. Como el caso fue ejecutado en varios procesadores, en el directorio principal no encontraremos los pasos de simulación. Éstos se encuentran en las carpetas *processor\** y los archivos están divididos (caso descompuesto).

Para reconstruir el caso debemos ejecutar el comando `$ reconstructPar` en el directorio *case*. Ahora encontraremos los campos de presión y velocidad para la iteración 2000 en el directorio principal.

### 4. SIMULACIÓN SCALARTRANSPORTFOAM

Preparamos el caso de transporte escalar copiando la malla en *meshCase/constant/polyMesh* en el directorio *constant* del caso RTD. También traeremos de la simulación de *simpleFoam* el campo de velocidad 2000/U al directorio de condiciones iniciales 0/U en RTD. Ajustamos las condiciones en *system* según los valores del repositorio

#### 4.1. Descomposición en varios procesadores

En el diccionario `system/decomposeParDict` se indica la cantidad de subprocesos que se usarán para correr la simulación. Ajustar este valor en función de la cpu dónde se ejecuta. En este caso, se utilizarán 4 procesadores.

Ejecutamos `$ createPatch -overwrite` y luego `$ decomposePar -force` en el directorio principal y aparecerán ahora directorios `processor*`. Ahora, con el siguiente comando podremos iniciar la simulación:

```
$ mpirun -n 4 scalarTransportFoam -parallel |  
>tee log.scalarTransportFoam
```

En la figura 8 se ven los residuales de esta simulación. Recordando que es una simulación en estado transitorio, se puede concluir que el solver llega a una solución constante a partir del 1er segundo de simulación aproximadamente.

Figura 8: Evolución de residuales con simpleFoam

## 5. POST-PROCESO

Vamos a usar principalmente *ParaView* para analizar los resultados. En la figura 9 podemos ver unas líneas de traza coloreadas en función de la concentración de agroquímico. Se hizo la muestra sobre una línea paralela al eje 'Y' para facilitar la interpretación del campo tridimensional. Se observa que la mayor parte del mezclado se realiza, en realidad, justo después de la cámara de mezclado. Este fenómeno se puede adjudicar a que el flujo es laminar (por el modelo propuesto en la simulación), dificultando el mezclado dentro de la cámara.

Figura 9: Líneas de traza coloreadas según concentración de agroquímico

Con una rápida inspección visual parece ser que a la salida la concentración de agroquímico es aproximadamente 0.7. Usando la función de posproceso definida en el caso de OpenFOAM como *patchAverage* calculamos la evolución de la concentración en la superficie de salida. El valor final es de 72 %. En la figura 10 se muestra esta evolución.

Figura 10: Evolución la concentración en la salida del dispositivo

Si hacemos varios cortes en distintas secciones del dispositivo podemos ver cómo varía la distribución de concentración en el eje 'X', como se muestra en la figura 11. Observar que el agroquímico no llega a difundir perfectamente en la salida (color verde).

Sería conveniente analizar más detalladamente la concentración en la superficie de salida del dispositivo. Si bien debería mantener una simetría respecto del eje del dispositivo, en la figura 12 queda en evidencia que esto no sucede así. Una posible razón de este inconveniente, es la malla ortogonal alineada con los ejes cartesianos en todo el volumen de la muestra.

Esto puede implicar errores en la interpolación entre celdas. Será diferente para celdas contenidas en el plano XY (que debería tener componentes de velocidad únicamente en X e Y) que una que esté girada a 45 grados respecto a un plano paralelo al YZ. En este caso, el método debería interpolar en dirección a vértices de las celdas hexahedricas y no sobre caras o aristas, generando diferencias entre celdas de una misma sección. En la figura 13 se muestra esta idea.

Figura 11: Distribución de concentración de agroquímico en varias secciones del dispositivo

Figura 12: Concentración en Outlet. Valores por celda

Además, hay que tener en cuenta que el método no pudo reducir los residuales por más de  $10^{-4}$ , generando estas discrepancias con lo esperado. En ciertos casos, estas diferencias pueden ser perfectamente aceptables mientras que en otros no. Esto es criterio del ejecutante y de las exigencias que debe cumplir la simulación.

## 6. CONCLUSIONES

En este informe preparamos y ejecutamos el primer caso tridimensional del dispositivo. Generamos una malla sobre un modelo arbitrario creado en *FreeCAD* y ejecutamos las simulaciones correspondientes. Surgieron dificultades con la convergencia de las simulaciones, mostrando de alguna manera la influencia del mallado sobre los resultados.

Además, se hace notar que los resultados esperados en cuanto a la concentración promedio en el outlet debería ser similar a los del **informe anterior** (34 %) pero en este caso dio valores muy diferentes (72 %). Revisando las condiciones impuestas, los valores de presión en los inlets debía ser de 300 kPa, pero por error se hicieron las simulaciones con 0.3 kPa. No se realiza la simulación nuevamente ya que éstos valores generan mayores inconvenientes en la convergencia (con los mismos diccionarios en *system* los residuales no bajan de  $10^{-2}$ ) y se considera un trabajo innecesario para los fines prácticos de este informe y adscripción. Se prefiere continuar con una simulación más avanzada en el siguiente informe.

Aprovechando que las condiciones de la simulación no permiten resolver adecuadamente el caso en estado estacionario, el siguiente informe se va a realizar utilizando un solver transitorio **pimpleFoam**.

Figura 13: Malla ortogonal en un caso simétrico respecto a un eje