

DESARROLLO DE UNA INTERFAZ GRÁFICA DE USUARIO DE CÓDIGO ABIERTO PARA OPENFOAM®

Santiago Chialvo^A

^A*Facultad de Ingeniería y Ciencias Hídricas - Universidad Nacional del Litoral*

Área: Ingenierías.

Sub-Área: Informática y Sistemas de Información.

Grupo: X

Palabras clave: Simulación computacional, mecánica de fluidos, Python.

INTRODUCCIÓN

Dentro del área de la simulación computacional, específicamente en problemas de fluíddinámica, existe una amplia variedad de *softwares* que ofrecen las funcionalidades adecuadas para tal fin. La principal desventaja de muchos de ellos es su carácter privativo, obligando al usuario a abonar costosas licencias para su uso.

OpenFOAM® [1] es un software de código abierto, gratis y multiplataforma. Su enorme versatilidad y robustez lo convierten en una de las herramientas más utilizadas en áreas de ciencia e ingeniería, tanto para fines comerciales como académicos. Sin embargo, cuenta con una desventaja: la herramienta no es *user-friendly*, es decir, la curva de aprendizaje para el usuario promedio es lenta.

Las interfaces gráficas de usuario (*GUIs*) simplifican el uso de programas permitiendo una rápida asimilación y manipulación de los datos presentados. Este informe presenta el trabajo de desarrollo de una interfaz gráfica de usuario para OpenFOAM® (de ahora en adelante, *petroSym*) la cual busca facilitar al usuario el uso de esta poderosa herramienta. El desarrollo surge como respuesta a un pedido de trabajo de la empresa Y-TEC. [2]

Proyecto: Combinación de metodologías eulerianas para resolver flujos multifásicos y multifluidos en forma eficiente (CAI+D tipo A, PI 2501 201101 00435 LI).

Director del proyecto: Nigro, Norberto Marcelo

Director del becario/tesista: Nigro, Norberto Marcelo

Co-Director del becario/tesista: Ramajo, Damián Enrique

METODOLOGÍA

Enfoque de la herramienta

En la Figura 1 podemos observar el pipeline básico de una simulación computacional utilizando OpenFOAM®. Los tres grandes bloques diferencian las tareas en las cuales se enfoca petroSym, y cuales son dejadas a OpenFOAM® u otro software externo.

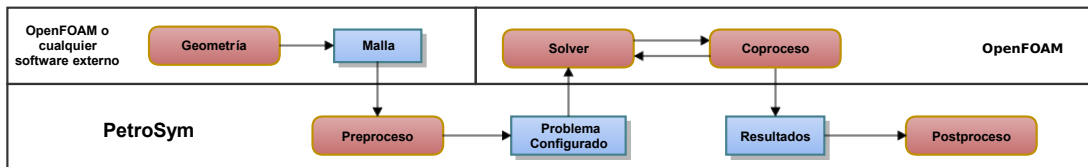


Figura 1: Flujo de trabajo de una simulación computacional utilizando OpenFOAM®.

A partir de la geometría del problema, de un CAD, STL o datos fuente, obtenemos la malla, la cual puede ser generada con cualquier generador externo como ANSA o GID, o bien con utilidades de OpenFOAM® como *blockMesh*. Los *solvers* utilizados para la resolución del problema, que permiten obtener los resultados numéricos, son también provistos por OpenFOAM®.

El enfoque dado a petroSym y su carácter gráfico facilita la tarea de configuración del problema (etapa de preproceso) y visualización de resultados tanto en tiempo real (etapa de coproceso) como una vez obtenidos los resultados finales (etapa de postproceso). Pese a que la interfaz se desvincula de temas referidos a la generación de la malla y los *solvers* utilizados, ofrece funcionalidades que permiten al usuario realizar estas tareas de una manera mas asequible, por ejemplo, evitando errores de *input* de datos (lo cual podría generar un archivo corrupto).

Herramientas de Trabajo

La elección de las herramientas de trabajo fue llevada a cabo mediante una rigurosa búsqueda bibliográfica [3][4], escogiendo aquellas que resultaban mas beneficiosas en cuanto a su interoperabilidad, fiabilidad, y eficiencia.

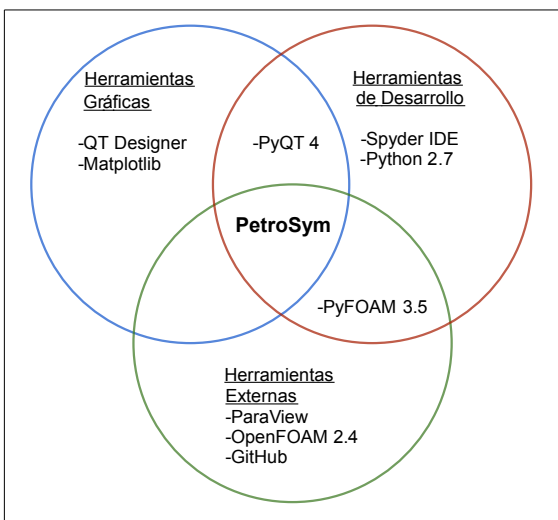


Figura 2: Diferentes herramientas utilizadas para desarrollar PetroSym.

Podemos dividir a las herramientas utilizadas en tres grandes grupos:

1. Herramientas de Desarrollo: Utilizadas para construir el código fuente de la interfaz.
2. Herramientas Gráficas: Utilizadas para construir las ventanas, botones, *widgets* y demás elementos gráficos relacionados a petroSym.
3. Herramientas externas: Necesarias para asegurar el correcto funcionamiento de la interfaz y para mantener el control de versiones de una manera eficiente y robusta.

Muchas de las tareas realizadas por petroSym son secuenciales, y utilizan como entradas las salidas de la tarea ejecutada anteriormente. Con el fin de lograr esta secuencialidad, se utiliza el envío y recepción de señales entre *threads*. Este funcionamiento puede observarse en el lado izquierdo de la Figura 3.

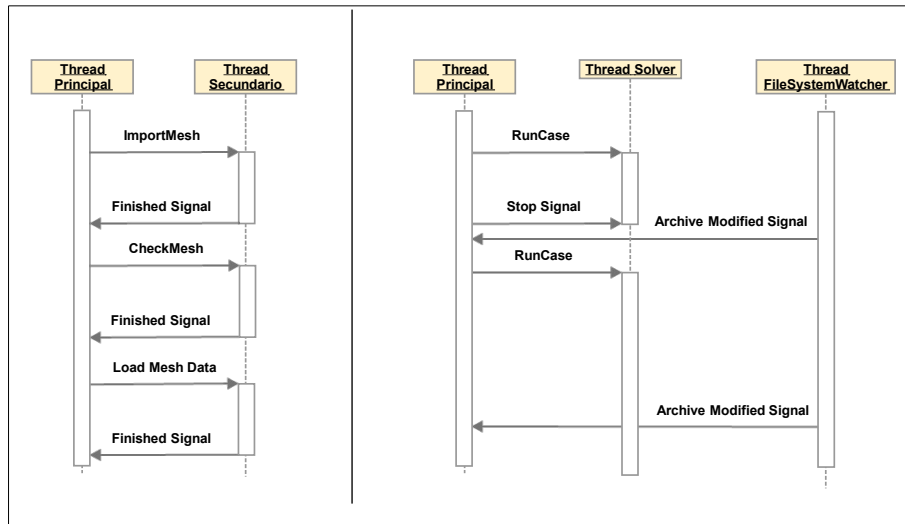


Figura 3: Izquierda: Secuencialidad de tareas. Derecha: Lógica *multithreading*.

La GUI siempre debe correr en el *thread* principal. Todos los *widgets* asociados no puede trabajar en ningún otro *thread*. Para lograr una *non blocking GUI*, es decir, evitar que se bloquee ante eventos externos y sea totalmente responsiva en todo momento, se utilizan *threads* secundarios. En la Figura 3 (derecha) observamos un caso típico de *multithreading* que ocurre durante el uso de petroSym: como mencionamos anteriormente, en el *thread* principal se ejecuta la interfaz gráfica. Si el usuario decide comenzar una simulación, una señal es enviada y el *solver* de OpenFOAM® se ejecuta en un *thread* secundario. Si el usuario decide parar la simulación, nuevamente otra señal será enviada, deteniendo la misma. Aquí vemos la importancia de no utilizar el *thread* principal para estas tareas, ya que si el usuario pulsa algún botón mientras el *thread* principal está ocupado deteniendo la simulación, el mismo esperará a completar la tarea actual para recién dar lugar a la petición del usuario.

RESULTADOS Y USO DE LA HERRAMIENTA

PetroSym provee una interfaz robusta para la importación de la malla a utilizar. Datos geométricos referidos a la malla tales como su volumen, no ortogonalidad, cantidad y tipo de celdas, son presentados de una manera compacta y ordenada al usuario, como se observa en la Figura 4. Además, esta misma lógica es utilizada en las demás pestañas referidas a la configuración del caso, la corrida y el postproceso. Entre las múltiples herramientas desarrolladas, petroSym provee las facilidades necesarias para inyectar un trazador de manera muy sencilla, (Figura 5) siendo posible elegir el *patch* de entrada y el tiempo de inyección. La interfaz además provee al usuario de funcionalidades extras tales como el monitoreo de residuos, seguimiento de campos a partir de *sampling* de la solución, visualización de *snapshots* en tiempo de corrida, entre otros.

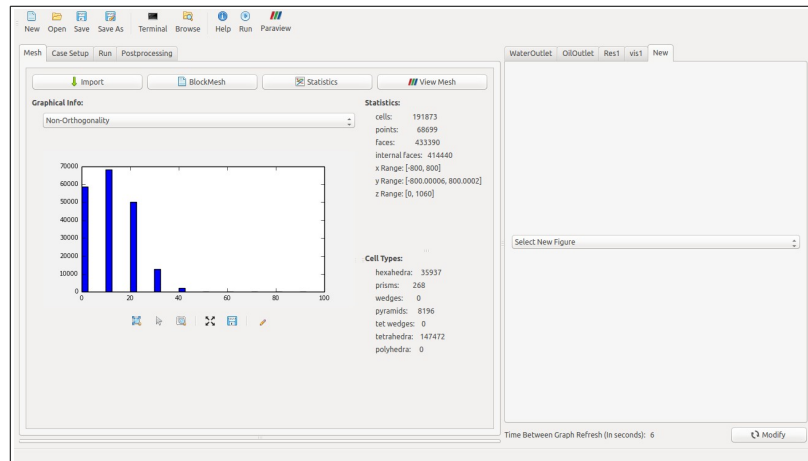


Figura 4: Pestaña de petroSym dedicada a la geometría (*mesh*)

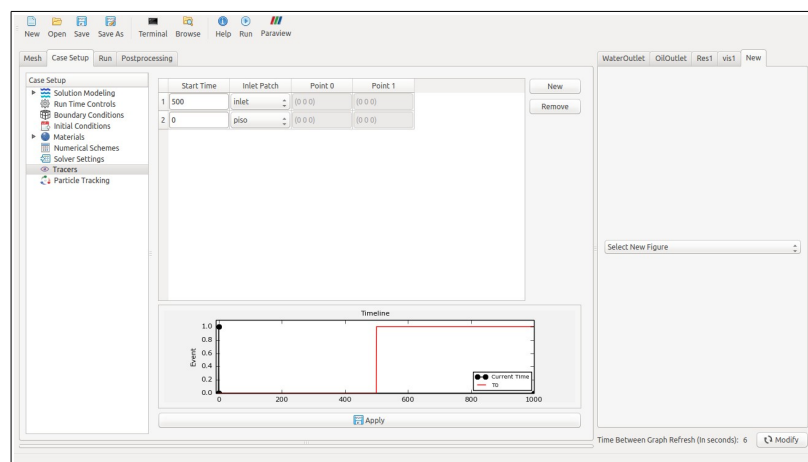


Figura 5: Inyección de un trazador

CONCLUSIONES Y FUTUROS TRABAJOS

El presente informe ha demostrado que petroSym provee las herramientas necesarias para que un usuario nuevo en el área de la simulación computacional utilice OpenFOAM® de una manera sencilla y eficiente. Además, su construcción fue llevada a cabo siguiendo reglas del buen diseño que permiten la reutilización de código, asegurando al mismo tiempo un fácil mantenimiento del mismo y una sencilla manera de adicionar nuevas funcionalidades. Como trabajos futuros, podemos mencionar:

- Intregar nuevos *solvers* a la interfaz.
- Vincular herramientas de ParaView en petroSym.
- Incorporar más de las *function objects* provistas por OpenFOAM®.

BIBLIOGRAFÍA

- [1] OpenCFD Ltd. 2016. OpenFOAM description. Fecha de consulta: 01/09/16. URL: <http://www.openfoam.com/>
- [2] YPF Tecnología. 2016. Y-TEC description. Fecha de consulta: 01/09/16. URL: <https://y-tec.com.ar/Paginas/sobre-ytec.html>
- [3] Daniel Molkentin. 2007. The Book of Qt 4: The Art of Building Qt Applications.
- [4] John Zelle. 2002. Python Programming: An Introduction to Computer Science, 1st Ed.