Librería Terablood v 0.1

Simulación de partículas deformables

10/06/2011

Universidad de Los Andes

Oscar Castillo, Andrés Gonzalez

Manual

Librería Terablood v 0.1

Oscar Castillo Orduz

Universidad de Los Andes – Facultad de Ingeniería

Departamento de Ingeniería Mecánica

2011-06-21

1. **Introducción**

En presenta una guía rápida para utilizar la librería Terablood sobre la plataforma Eclipse. El objetivo es que el lector sea capaz de implementar tres casos básicos (*Corte, Poiseuille, Multiple*) sobre deformación de células debido a la imposición de un flujo externo.

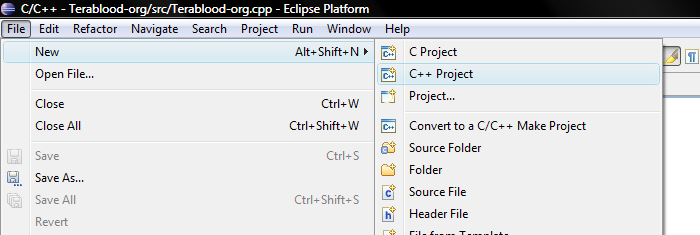
Eclipse: IDE (Integrated Development Environment) permite el desarrollo profesional de aplicaciones en diversos lenguajes de programación para este manual se ha utilizado la versión 3.4 para desarrollo sobre C++. Más información puede ser encontrada en <http://www.eclipse.org/downloads/> en la versión para desarrollo en C++. Desde esta página se descarga una carpeta que contiene el ejecutable de eclipse. Eclipse NO requiere instalación tan solo se debe descomprimir en cualquier ubicación Ej. C:/. Una vez haya descomprimido la carpeta encontrará el ejecutable eclipse.exe el cual es el punto de entrada a la aplicación. Para más información sobre eclipse puede consultar el siguiente enlace en el cual presentan el manejo básico de la interfaz.

<http://cupi2.uniandes.edu.co/sitio/index.php/cursos/apo1/nivel-1?start=5>

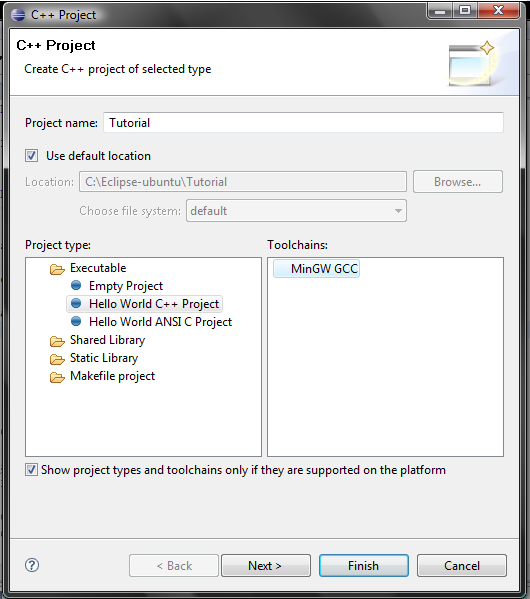
**2. Creando la estructura del proyecto**

Antes de comenzar debe tener a su disposición el entorno eclipse y la carpeta Terablood que contiene los archivos fuente de la librería.

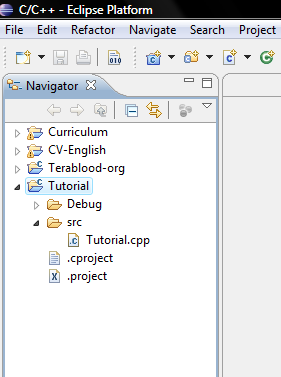
1. Crear un nuevo proyecto en eclipse.



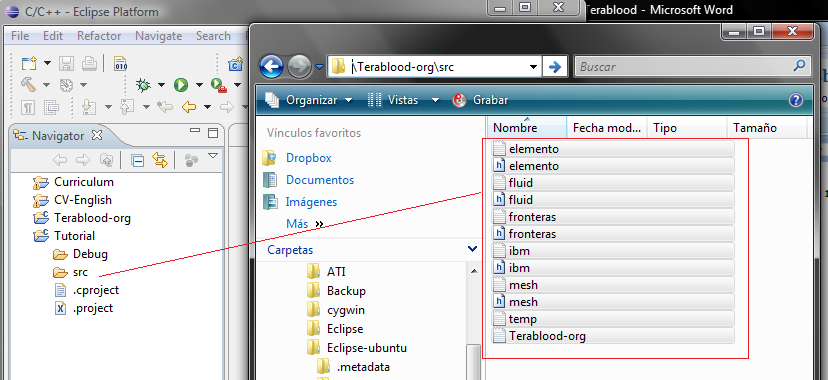
1. Asignar el compilador por defecto y dar un nombre al proyecto

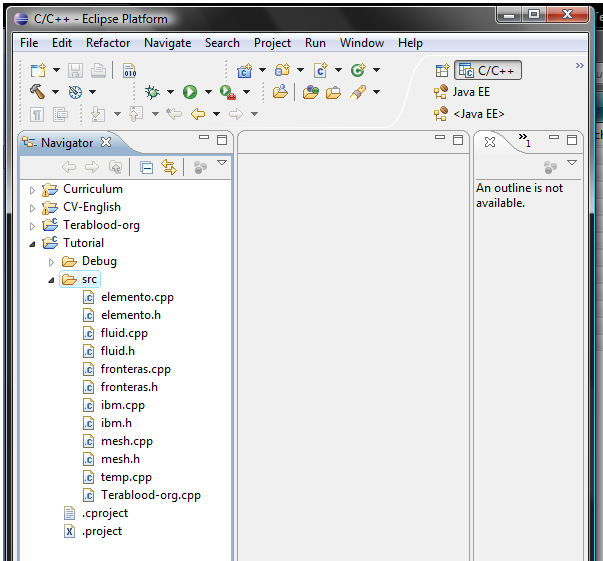


1. Eliminar el contenido del proyecto. En este caso se debe eliminar todo el contenido de la carpeta src.

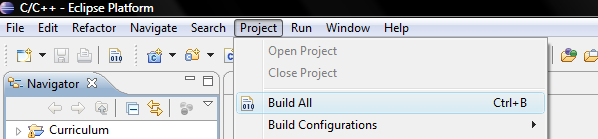


1. Copiar el contenido de Terablood/src a la carpeta Tutorial/src el resultado se muestra a continuación.

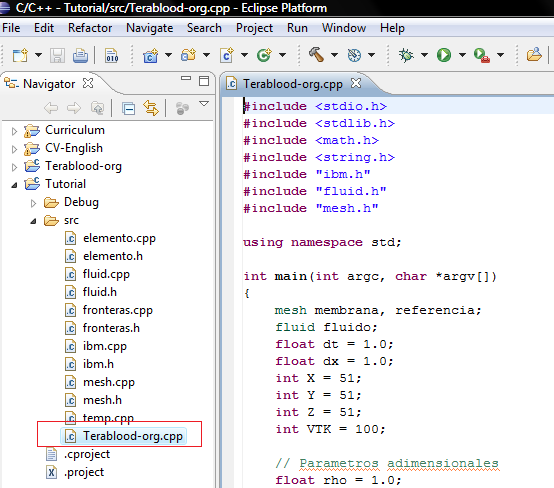


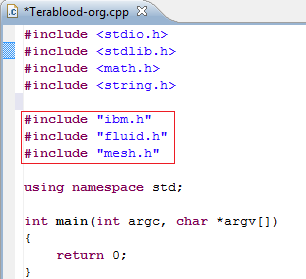


1. Reconstruir el proyecto para tener en cuenta el nuevo los cambios en el código fuente.

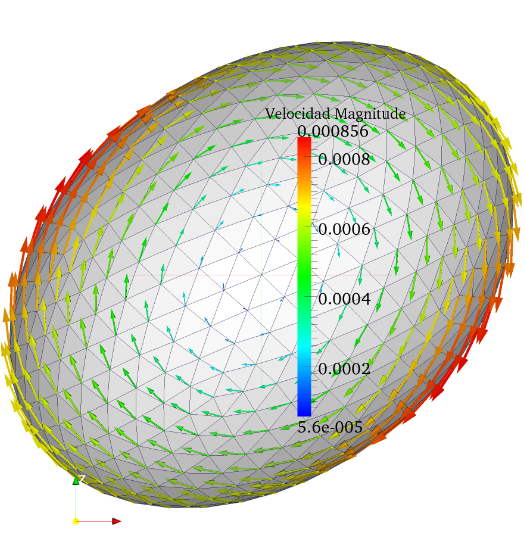


1. Modificar el archivo de entrada Terablood.cpp, borrar todo el contenido que se encuentra en este archivo a excepción de aquellas líneas que son indispensables para la ejecución. **Tenga en cuenta que está incluyendo las siguientes librerías ibm.h, fluid.h, mesh.h. Estas representan los tres elementos fundamentales del método Immersed Boundary, Fluido en Lattice Boltzmann y el modelo de la membrana.**

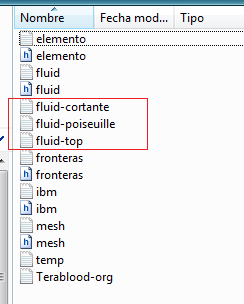




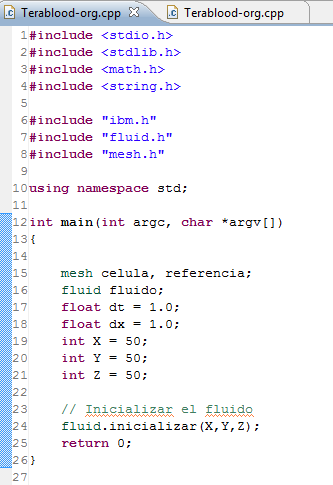
Caso 1: Célula suspendida en flujo cortante



1. En este punto se ha de decidir el tipo de simulación que se quiere implementar. Para este primer paso se quiere implementar una célula suspendida en un flujo cortante. Tal como se muestra a continuación. El flujo cortante se genera aplicando una velocidad a la pared superior e inferior de igual magnitud pero dirección opuesta.
2. El primer paso para cualquier simulación es definir el tipo de flujo que se debe implementar. En la carpeta Terablood/src encuentra los siguientes archivos fluid-cortante.cpp, fluid-poiseuille.cpp, fluid-top.cpp. Para este caso vamos a utilizar el archivo fluid-cortante.cpp.

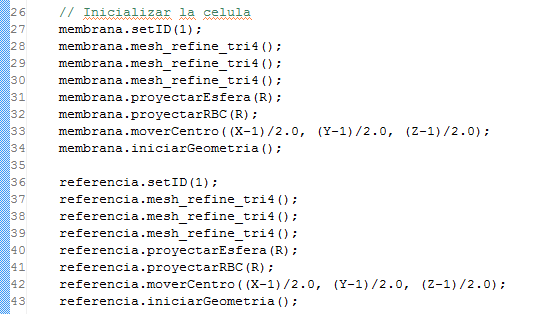


1. Copiar el contenido de fluid-cortante.cpp al archivo fluid.cpp. Esto es necesario debido a que la implementación de cada tipo de flujo debe manejar condiciones de frontera diferentes.
2. Las librerías ibm.h y mesh.h son genéricas y funcionan para cualquier tipo de configuración, es decir no hay necesidad de realizar ninguna modificación a estos archivos. Únicamente se debe modificar el archivo fluido.cpp.
3. Hasta el momento se han realizado los ajustes del código para el caso de flujo cortante. Ahora vamos a comenzar con la creación de la geometría de la célula y la geometría del fluido.
4. En el archivo Terablood-org.cpp agregar las siguientes líneas:



La línea 15 muestra la creación de dos objetos de la clase mesh.cpp y la línea 16 crea un objeto de la clase fluid. La función inicializar de la clase fluid crea todas las estructuras en memoria para almacenar las variables macro del fluido así como las celdas del método Lattice Boltzmann (LBM) D3Q19. ***Por cada célula que se quiera modelar se deben construir 2 objetos del tipo mesh, el primero corresponde a la malla que se deforma en el tiempo y la segunda es la referencia desde la cual se van a calcular las deformaciones de cada elemento en el algoritmo de elementos finitos FEM.***

13, La creación de las estructuras para las células requiere de una secuencia ordenada de pasos, no se puede alterar a forma en que se llaman los métodos debido a que cada función depende de las estructuras que crean los métodos anteriores. El orden se muestra a continuación.



La línea 27 asigna un identificador único a la célula, se utiliza para configuraciones en las cuales existen múltiples estructuras en la simulación. De la línea 28 a 30 se realiza el refinamiento dividiendo cada cara del icosaedro en 4 nuevos triángulos, el resultado se muestra a continuación, por cada llamado de la función mesh\_refine\_tri4() se realiza un paso de refinamiento. La línea 40 proyecta cada punto del icosaedro refinado sobre la superficie de una esfera. Se han implementado 3 tipos de proyecciones posibles Esfera, Elipsoide y Red Blood Cells.

|  |  |
| --- | --- |
|  |  |

**Es importante que antes de realizar la proyección sobre cualquier geometría se haga una proyección sobre una esfera.** La línea 41 puede ser omitida para realizar una simulación de flujo cortante sobre la esfera. Por defecto la célula es creada con coordenadas de centro (0,0,0). Antes de trasladar la célula a cualquier otro punto sobre el espacio se deben realizar las correcciones en orientación.

Ejemplo: Rotar la célula 45° en X, 90° en Y y 135° en Z.

referencia.rotarEstructura(3.1416\*(1./4.), 3.1416\*(1./2.), 3.1416\*(3./4.));

Una vez se logre la orientación deseada se puede trasladar la célula a la coordenada deseada tal como lo muestra la línea 33. **La malla de fluido se crea de forma ascendente en el sentido positivo de cada eje, por tanto para ubicar la célula en el centro del flujo de debe trasladar desde la posición (0,0,0) creada por defecto hasta ((X-1)/2., (Y-1)/2., (Z-1)/2.) que es el centro de la malla del fluido. Por último se llama la función inicarGeometria() que crea y calcula todas la propiedades de la malla para calcular las fuerzas en cada nodo como curvaturas, deformaciones y otras estructuras auxiliares. Todas las operaciones realizadas en la malla deformable se deben replicar exactamente y en el mismo orden en la referencia.**

1. Una vez terminada la configuración geométrica se deben crear todos los parámetros a-dimensionales que describen el problema para el caso del flujo cortante el parámetro que describe el fenómeno es el número capilar definido como:



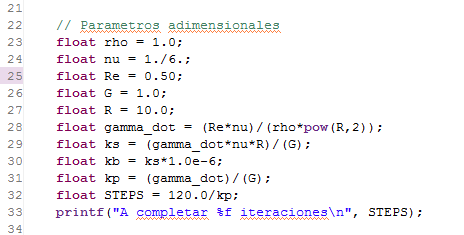
En donde,

, Es la tasa de corte del flujo cortante no perturbado. *Es la magnitud de la placa superior e inferior.*

, Viscosidad cinemática del fluido circundante.

, Radio o longitud característica de la célula.

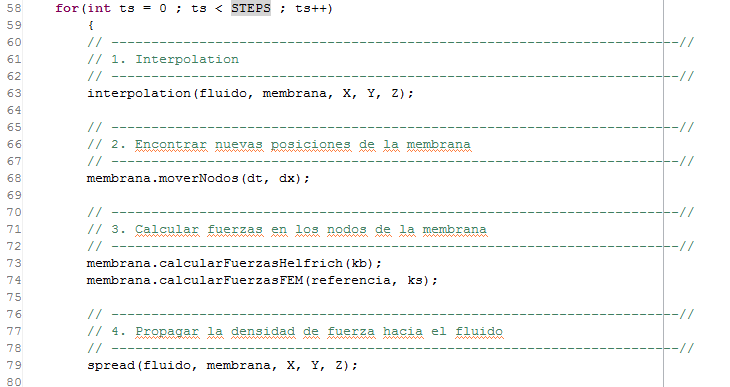
, Modulo de elasticidad en la membrana. (En el modelo de Skalak los dos módulos característicos son iguales). Sin embargo para modelos diferentes habrá que revisar la definición de cada número adimensional.

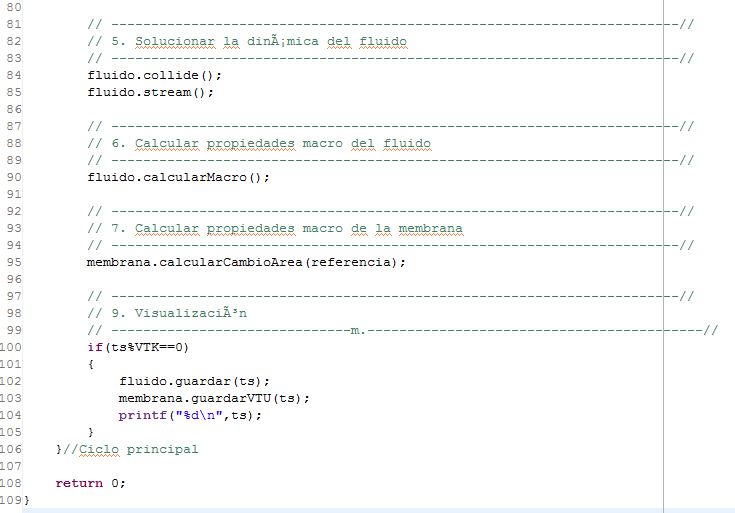


Finalmente el análisis adimensional debe proporcionar todas las variables físicas para poder empezar la simulación. Para este caso las variables que nos interesan son: Modulo de elasticidad, tasa de corte, y tasa de corte reducida para reportar el avance en pasos de tiempo normalizados. Para el fluido basta con inicializar el valor de velocidad característica para este caso la velocidad de la pared superior e inferior.



1. El siguiente paso es iniciar el ciclo principal de simulación, que consiste en los siguientes pasos:
   1. Interpolación – (Propagar la velocidad desde la malla de fluido a los nodos de la membrana).
   2. Encontrar nueva posición de los nodos de la membrana. (Aplicando el algoritmo Adams-Bazchfort ó Euler).
   3. Calcular las fuerzas debido a la deformación. (Aplicar el algoritmo de elementos finitos).
   4. Calcular las fuerzas de doblamiento utilizando el modelo de Helfrich.
   5. Calcular fuerzas adicionales.
   6. Spread – (Propagar las fuerzas nodales desde la membrana hacia la malla del fluido).
   7. Collision and Streaming.
   8. Calcular las propiedades macro del fluido.
   9. Calcular el cambio de área local (cada elemento) de la membrana respecto a la geometría inicial.
   10. Finalmente, el paso de visualización.



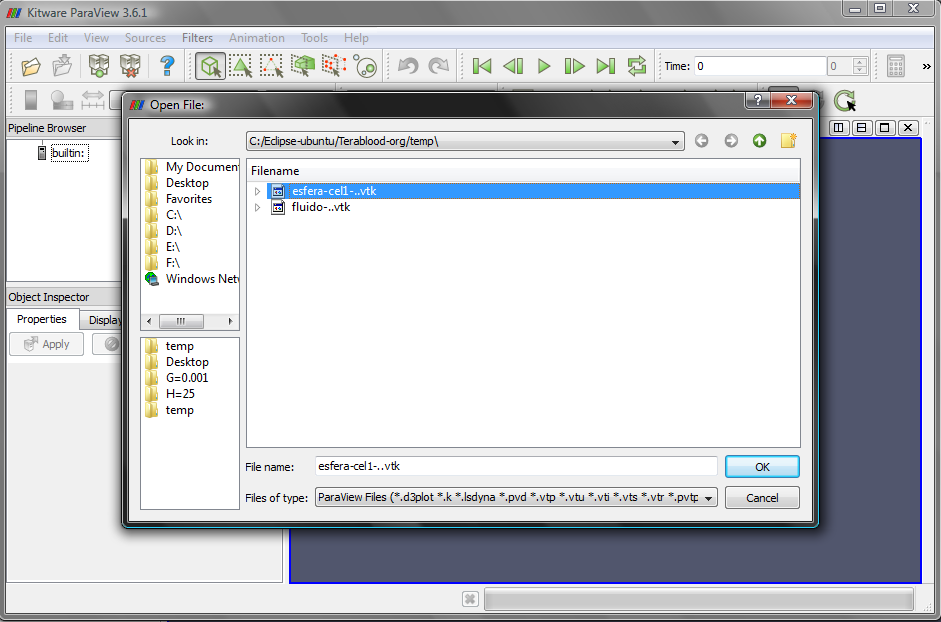


1. Post-procesamiento: La salida del programa se guarda en una carpeta /temp. Se guardan dos tipos de archivos:

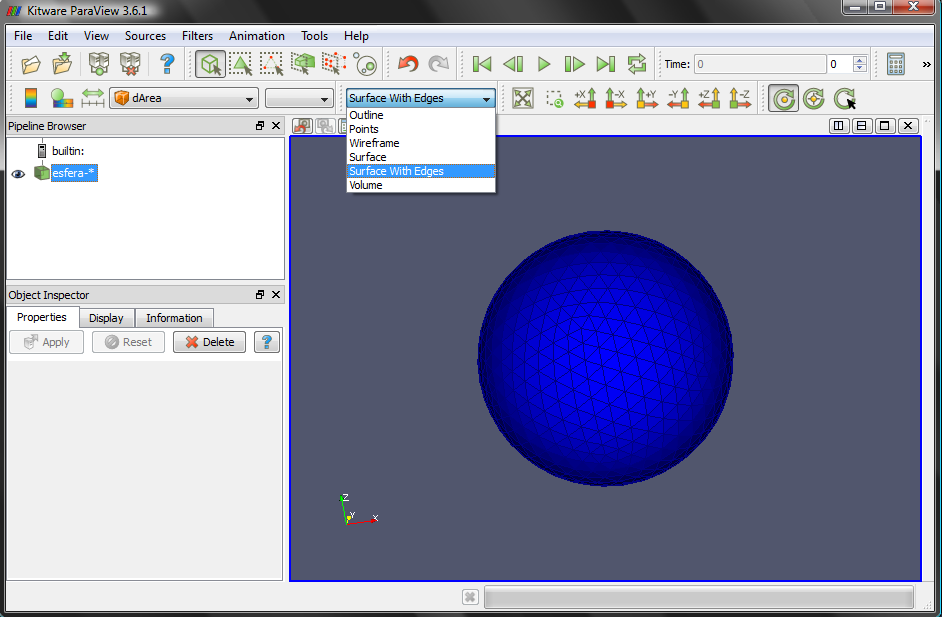
El primero con formato esfera-cell#-PASO.vtk, para este archivo # representa el ID de cada celula necesario para identificar cada estructura cuando se implementan varias células. PASO se refiere al número de iteración al cual pertenece el archivo.

El segundo con formato fluido-PASO.vtk, este archivo guarda todas las variables macro del fluido en una malla estructurada de acuerdo al estándar VTK.

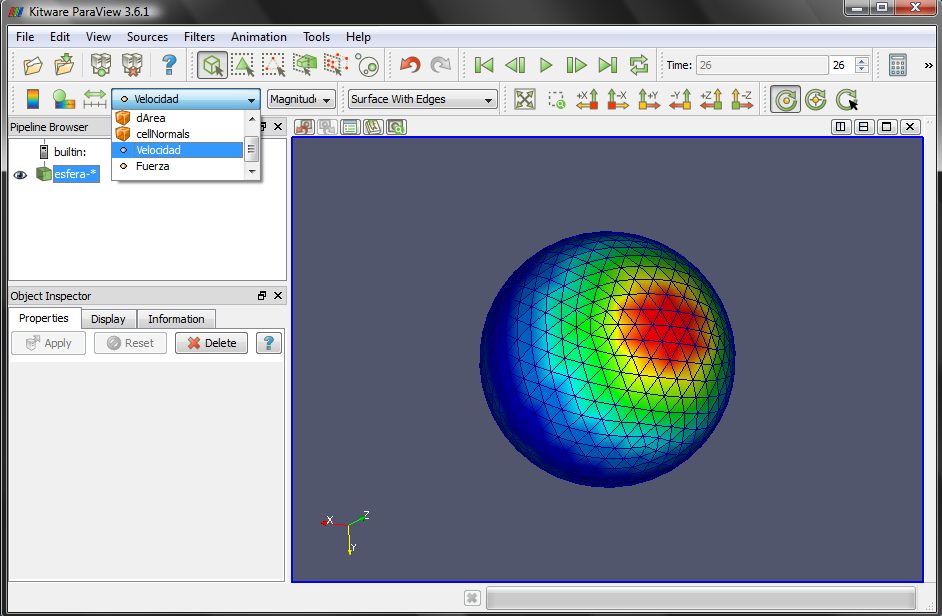
1. Estos archivos generados pueden ser leídos con cualquier paquete de post-procesamiento, sin embargo, se recomienda utilizar Paraview que es bastante flexible y brinda todas las herramientas necesarias para analizar los resultados.



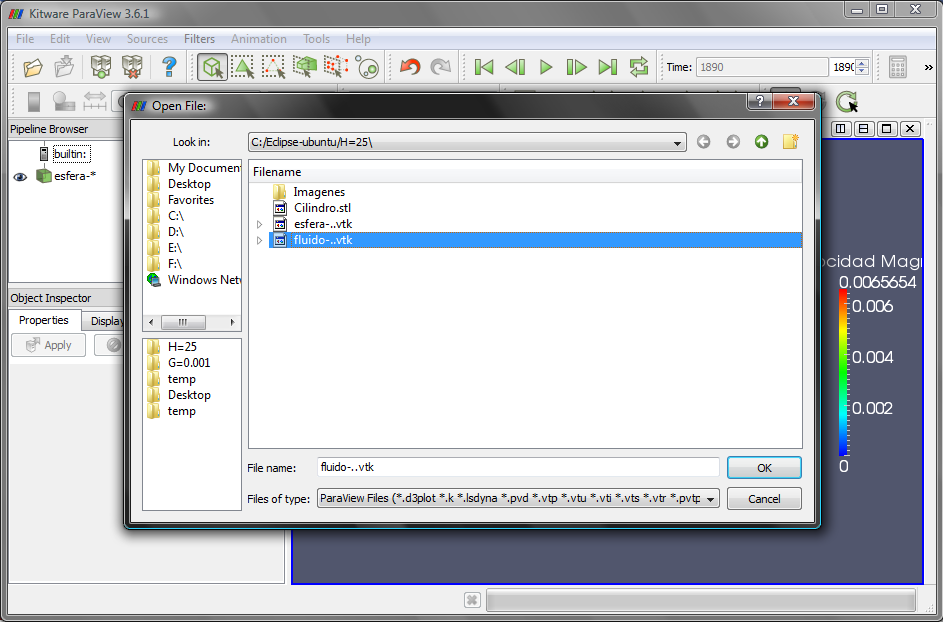
Cambiar las opciones de visualización



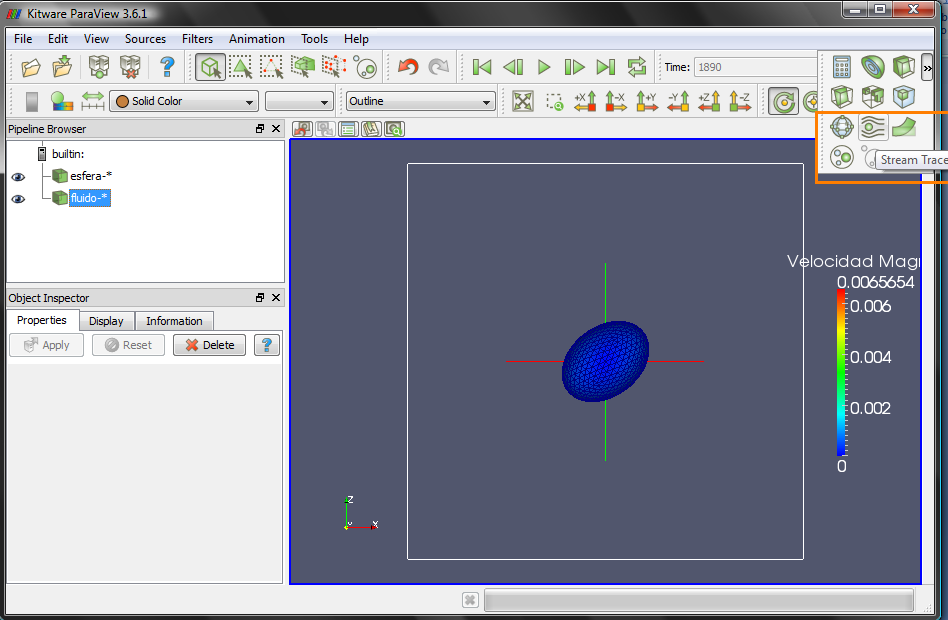
Mostrar la magnitud de la velocidad sobre la superficie



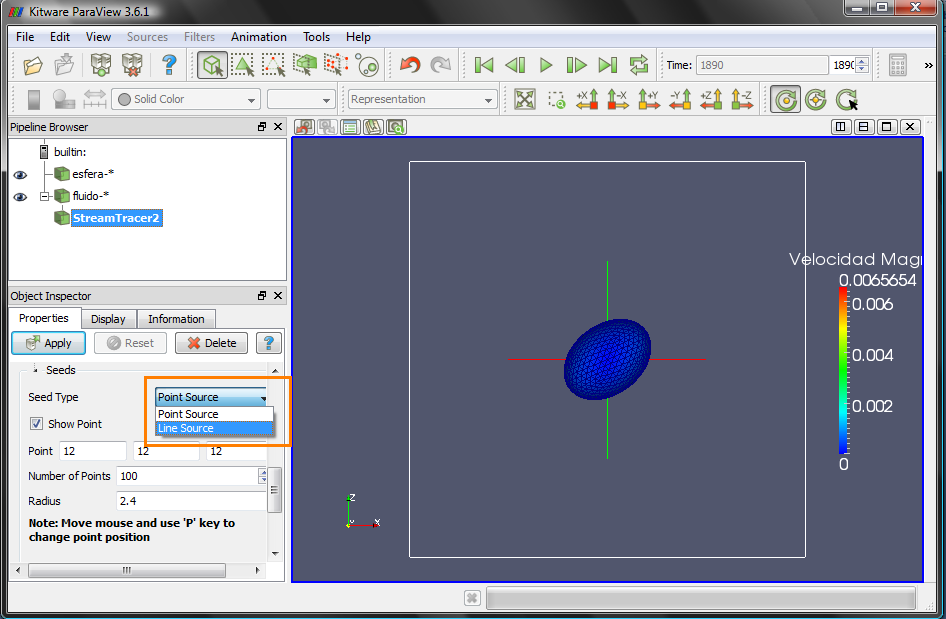
Ahora se debe sobreponer la estructura del fluido a la visualización



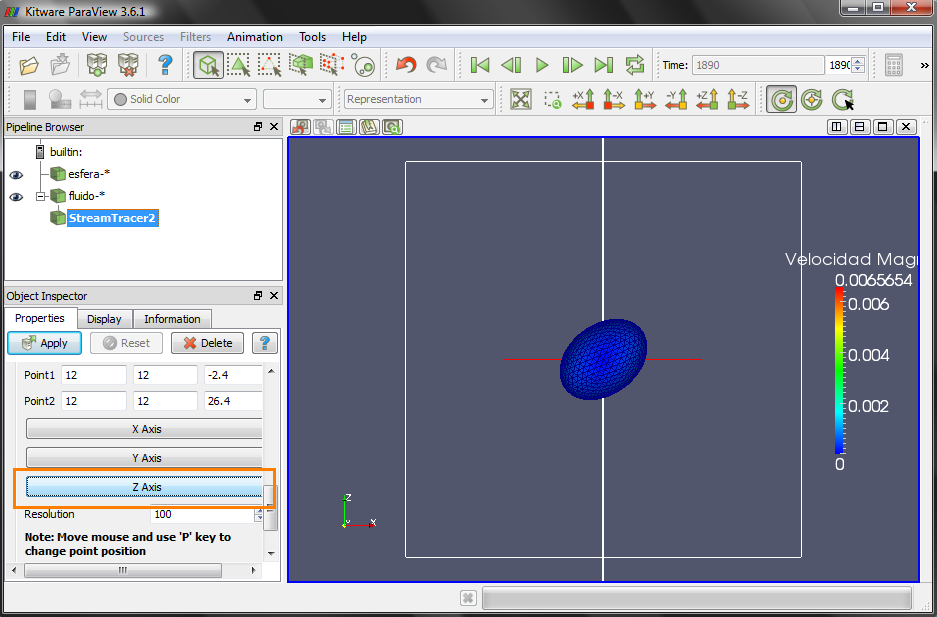
Una vez se ha importado el archivo se procede a mostrar las líneas de flujo. Se debe señalar el fluido en la parte izquierda en la ventana Pipeline Browser y luego aplicar la herramienta Stream Tracer.



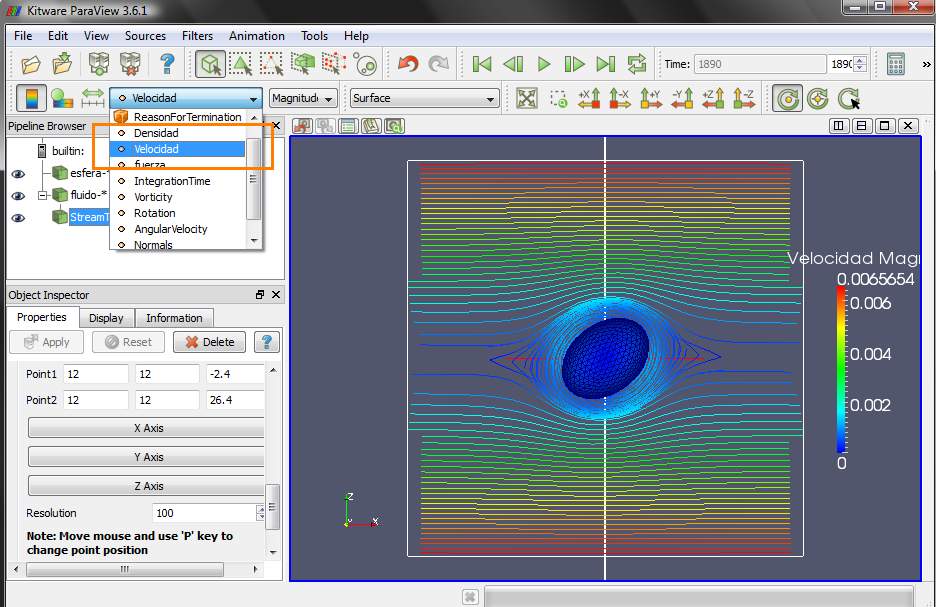
Esta herramienta tiene dos modos de funcionamiento, en el primer modo toma un punto en el espacio y traza todas las líneas de flujo que están contenidas alrededor de un radio r. El segundo modo consiste en trazar líneas de flujo que están sobre un plano. En este ejemplo se utiliza el segundo modo que proporciona mayor claridad al fenómeno en estudio. Se debe seleccionar Line Source en el panel Properties de la ventana Object Inspector. Se debe ubicar de acuerdo a los ejes mostrados en la parte izquierda inferior de la ventana de visualización.



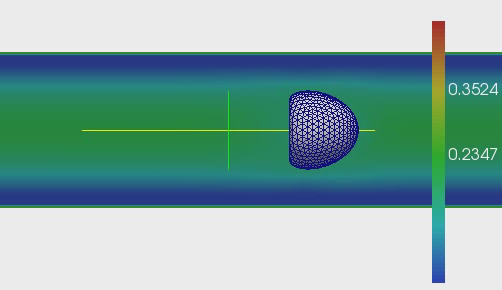
Para la situación de flujo cortante se debe especificar que muestre las líneas de flujo sobre el plano Z, es decir que la línea se debe seleccionar sobre el eje Z.



El resultado se muestra a continuación. Hasta aquí se ha mostrado una forma rápida para verificar los resultados que son producidos por el algoritmo, sin embargo, se recomienda revisar los manuales e información técnica del software Paraview.



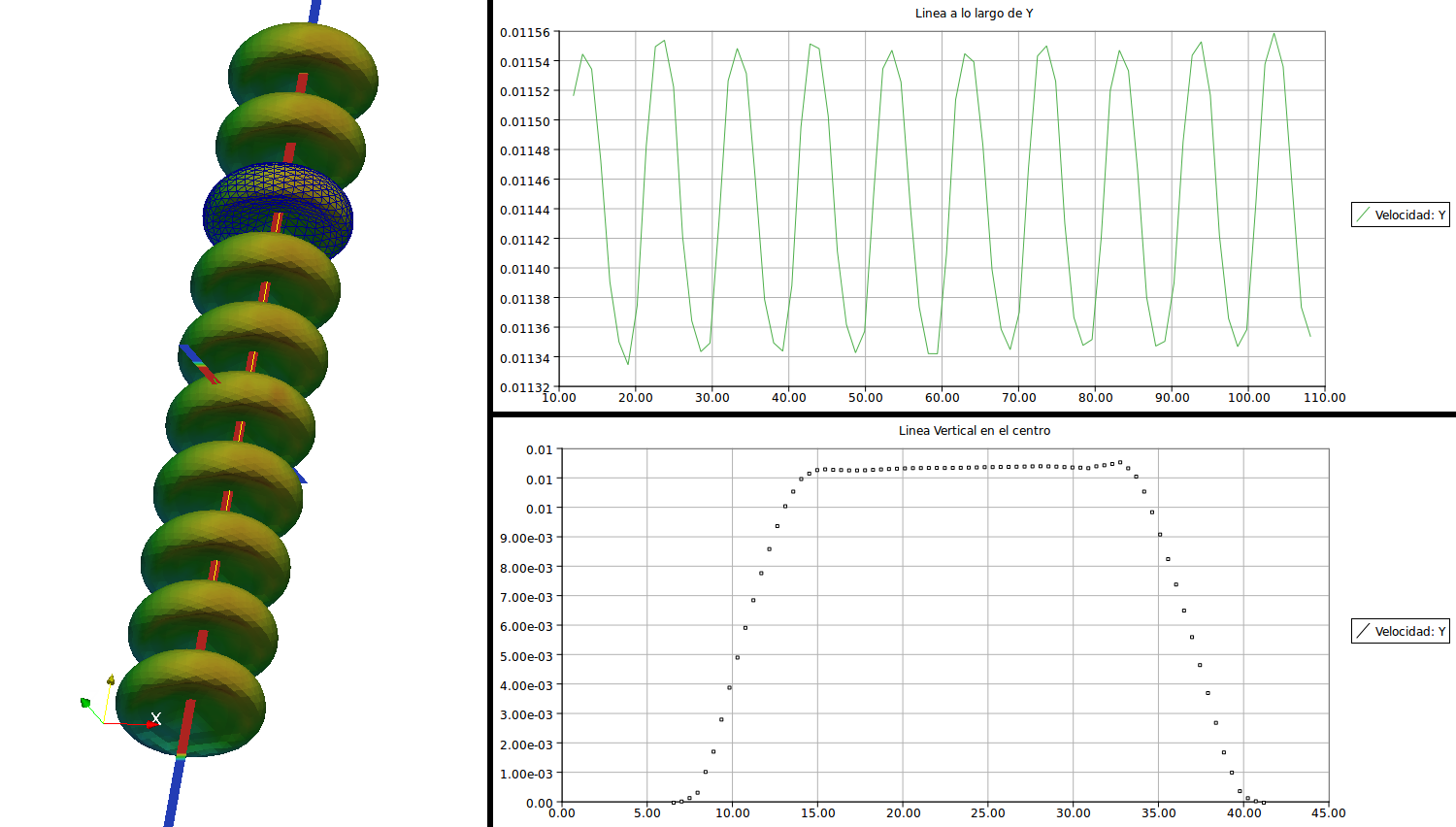
Caso 2: Célula suspendida en flujo de Poiseuille



El primer paso es definir el tipo de flujo que se va a utilizar, al igual que en el paso 8 debemos reemplazar el contenido del archivo fluid.cpp por el contenido del archivo fluid-poiseuille.cpp. Este último archivo implementa las condiciones de frontera necesarias para simular el flujo de Poiseuille las cuales incluyen un dominio periódico en la dirección Y y Bounce-back alrededor de un cilindro con eje en dirección Y. El flujo parabólico se desarrolla debido a un diferencial de presión en los extremos del ducto. Con tan solo hacer la modificación al archivo del fluido.cpp ya se ha incluido el efecto de Poiseuille. Lo que resta es realizar el análisis dimensional y modificar las dimensiones de la red de fluido.

Típicamente en la dirección del flujo la red es mayor. En esta versión ya han sido implementadas las condicionas de frontera periódicas en la dirección Y que es la dirección del flujo por defecto.

Caso 3: Múltiples Células suspendidas en flujo de Poiseuille



El punto de partida para este tercer caso es el caso 2 en el cual se implementa el flujo una sola célula suspendida en flujo de Poiseuille. La idea general es realizar N veces el procedimiento de inicialización y operaciones realizadas en el ciclo principal, en donde N es el número de células que están siendo simuladas.