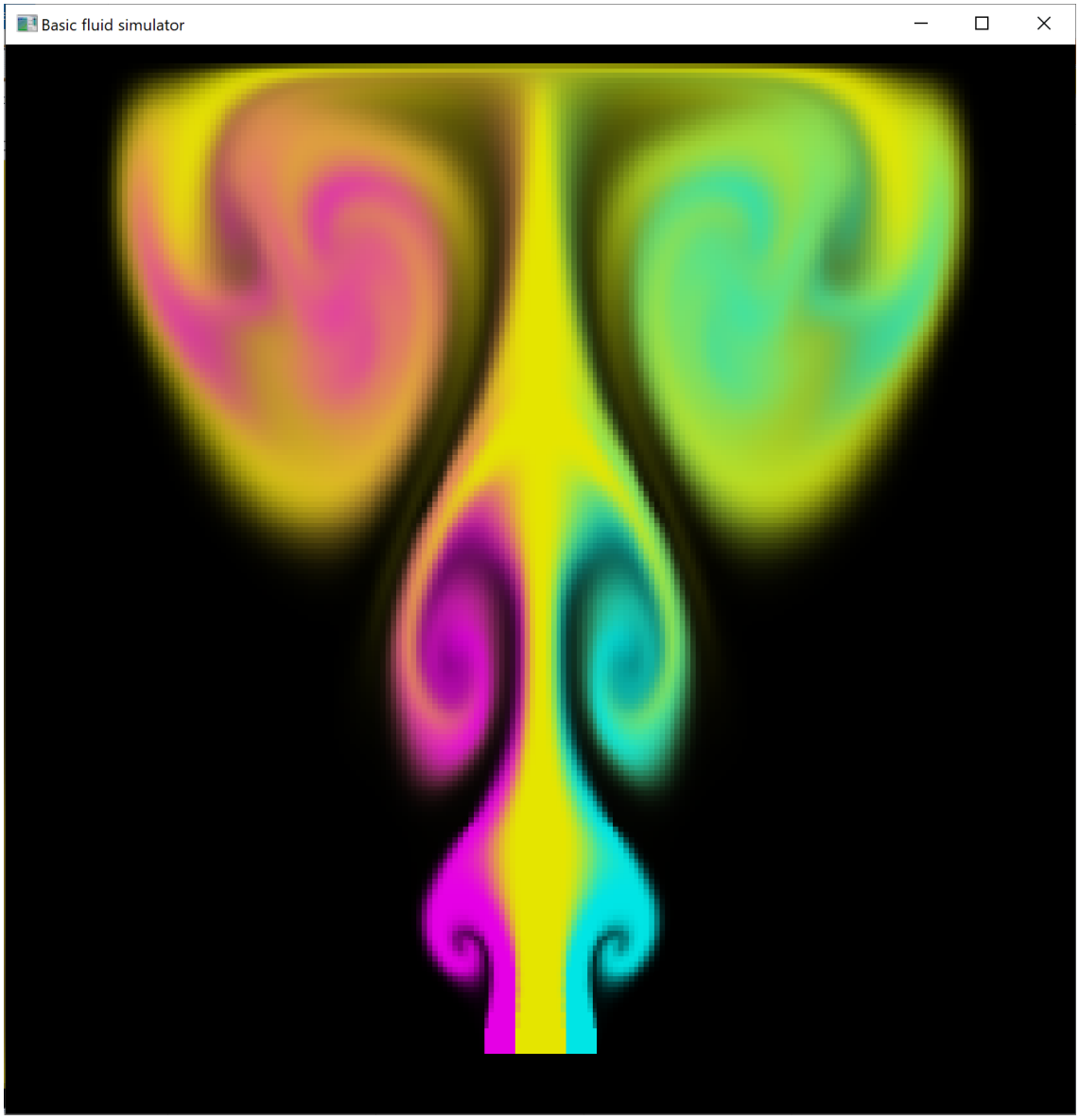
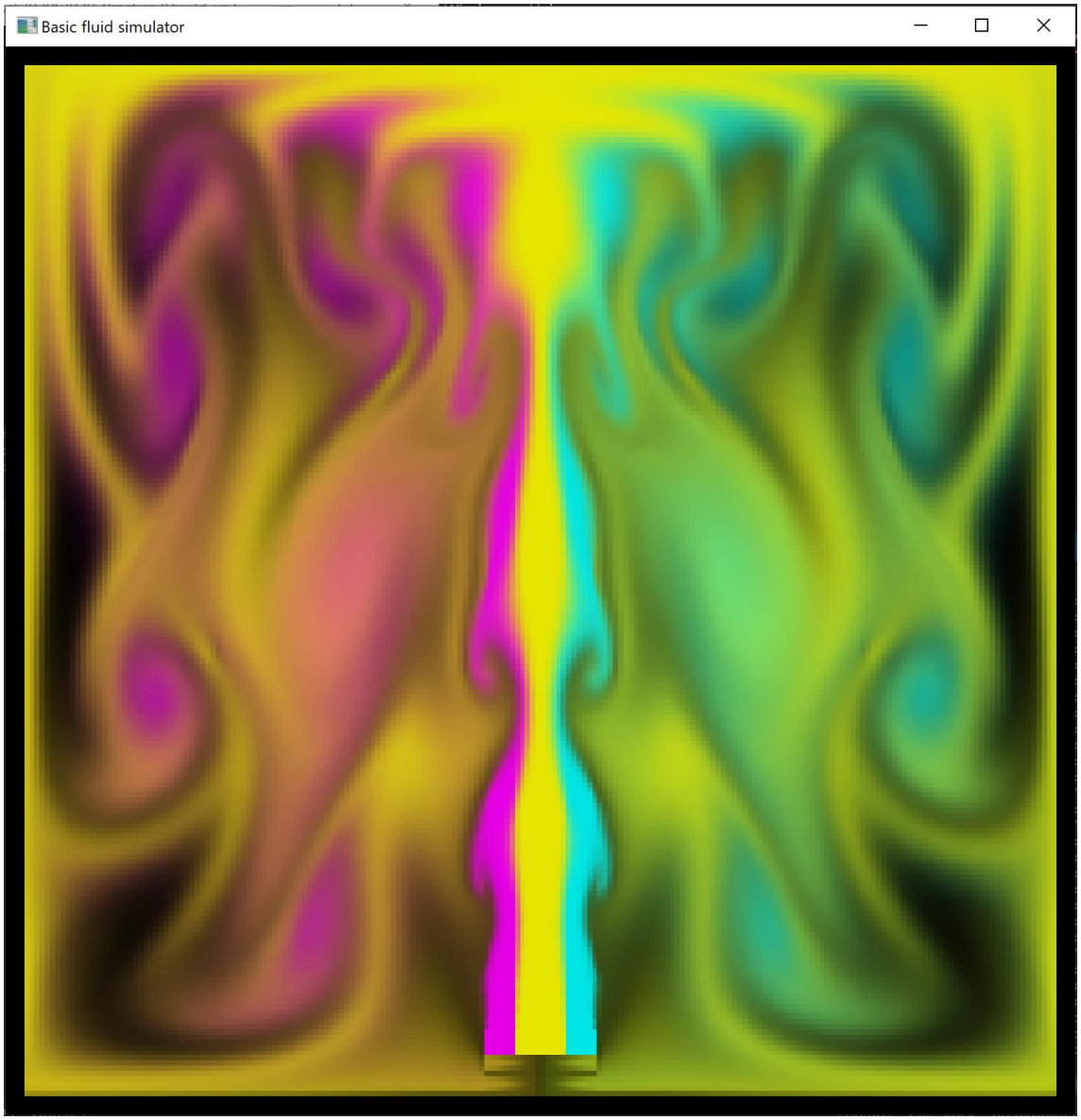
Animación Avanzada, Curso 2020-2021

Práctica 3a: Simulador de Fluidos Euleriano

(Asignación 25 de Marzo; Entrega Domingo 18 de Abril a las 23:55)




Basic fluid simulator

¿Cómo entregar la práctica?

Enviar una copia del fichero *Fluid2Exercise.cpp* antes del 18 de Abril a las 23:55. Se habilitará el proceso de entrega a través del campus virtual de la asignatura durante la semana previa a la entrega. En caso de que surja algún problema con el campus virtual se podrá realizar la entrega enviando el resultado de la práctica a través de email <*ivan.alduan@urjc.es*>.

Especificaciones Generales:

Se proporciona un proyecto sobre *Microsoft Visual Studio 2017/2019* preconfigurado con todos los ficheros de código que se encargan de inicializar y configurar la simulación, ejecutar el bucle de simulación, y visualizar los resultados utilizando la librería *freeglut*. La práctica también se puede realizar en otra plataforma (Linux, Mac) y compiladores, utilizando el fichero *CMakeLists* proporcionado y suponiendo que dicha plataforma disponga de una implementación de *Glut*.

El ejercicio consiste en la programación de cada paso necesario para la simulación de un fluido en 2D Euleriano siguiendo la teoría de las clases. La implementación de cada uno de los pasos necesarios para el funcionamiento de la simulación se encuentra en *Fluid2Exercise.cpp*. Todo nuevo código ha de ser añadido en el fichero *Fluid2Exercise.cpp*. La práctica está preparada para que no se necesiten librerías adicionales. Todo el código adicional que se precise ha de estar incluido en el fichero *Fluid2Exercise.cpp*.

La práctica se evaluará intercambiando el fichero *Fluid2Exercise.cpp*, compilando y ejecutando, con lo cual no compilará si se precisan otros ficheros.

De forma excepcional, en caso de que se realicen modificaciones adicionales necesarias para la evaluación de la práctica, se deberán incluir en la entrega todos los ficheros afectados y una breve explicación de los cambios realizados. La práctica está preparada para que se pueda obtener la máxima calificación modificando únicamente el fichero de entrega *Fluid2Exercise.cpp*.

No se requiere memoria explicativa, la evaluación se realizará a partir del código fuente.

Explicación de la Demo:

La demo contiene todo el framework necesario para realizar una simulación de fluidos Euleriana como la de la imagen, a falta de que el alumno proporcione las rutinas que componen los pasos necesarios para simular un fluido: emisión de tinta RGB, advección de las diferentes propiedades del fluido, viscosidad, gravedad y cálculo de presiones para proyectar el fluido a un estado incompresible.

El dominio en el que desarrolla su dinámica el fluido se considera fijo en el espacio y no parametrizable para evitar que sea necesario ajustar la cámara de nuevo para la correcta visualización de la rejilla. El número de celdas por cada dimensión es variable y se recomienda al alumno testear diferentes resoluciones y comprobar cómo el tamaño de las celdas afecta al resultado final de la simulación y al tiempo de cómputo. Otros parámetros configurables son la gravedad, la densidad del fluido, el coeficiente de viscosidad dinámica y el paso de tiempo.

Al iniciar la demo esta se encuentra detenida.

La tecla *'s'* permite pausar/continuar la simulación.

La tecla *'g'* permite visualizar la discretización subyacente a la simulación.

La tecla *'ESC'* permite salir del programa.

Clases Index2, Array2, Grid2, Fluid2:

Estas clases contienen todo el código necesario para gestionar los datos del simulador.

La clase Index2 sirve para manejar índices de celdas en un dominio 2D.

La clase Array2 permite almacenar cada propiedad de la rejilla y proporciona una manera sencilla de acceder a los datos almacenados mediante coordenadas 1D o 2D según convenga.

La clase Grid2 es la representación de una rejilla regular 2D y contiene diversos métodos relacionados con el manejo de rejillas 2D que pretenden facilitar la realización de la práctica al alumno.

La clase Fluid2 utiliza todas las clases anteriores y representa el estado completo del simulador mediante diferentes arrays para cada propiedad.

Clases SparseMatrix, PCGSolver:

La clase SparseMatrix es una representación de matrices dispersas mediante el formato CRS (Compressed Row Storage).

La clase PCGSolver implementa la resolución de un sistema lineal mediante el método de Gradiente Conjugado Precondicionado, con un precondicionador basado en una factorización de Cholesky incompleta. Estas clases se proporcionan al alumno para facilitarle la construcción y solución del sistema de Poisson para calcular las presiones del fluido. Las clases son lo suficientemente sencillas para ser utilizadas como *cajas negras* de manera que el alumno no tenga que conocer los detalles de implementación.