

1. Introducción a Fluidsim

Fluidsim es una librería científica escrita en **Python**, diseñada para la simulación numérica de flujos mediante métodos **pseudoespectrales**. Está pensada principalmente para investigación y docencia en dinámica de fluidos.

El simulador no es una aplicación con interfaz gráfica, sino un **framework de simulación**, en el cual el usted define un conjunto de parámetros físicos y numéricos, y luego ejecuta la simulación mediante scripts en Python.

2. Filosofía y arquitectura del simulador

Fluidsim sigue una arquitectura modular y orientada a objetos. Cada simulación se construye a partir de los siguientes componentes:

- Un **solver**, que define las ecuaciones físicas a resolver.
- Un conjunto de **parámetros**, que controlan el dominio, la discretización, la integración temporal y la física del sistema.
- Un sistema de **tiempo**, encargado de la integración temporal.
- Un sistema de **salida**, que gestiona los archivos de resultados.

El usuario interactúa principalmente con el solver y con los parámetros, mientras que el resto del flujo es gestionado internamente por Fluidsim.

3. Concepto de solver en Fluidsim

En Fluidsim, un solver representa un **modelo físico específico**. Por ejemplo, existen solvers para:

- Navier–Stokes en 2D
- Navier–Stokes en 3D
- Turbulencia cuasi-geostrófica

Cada solver define: las ecuaciones diferenciales a resolver, el método numérico utilizado y las variables dinámicas del sistema.

Puede ver todos los `solvers` disponibles en la [documentación](#), ahí mismo podrá ver los parámetros que componen cada uno.

4. Parámetros de simulación

Los parámetros de una simulación se almacenan en un objeto jerárquico, el cual contiene tanto parámetros físicos como numéricos.

Estos parámetros se crean mediante el método:

```
from fluidsim.solvers.ns2d.solver import Simul
params = Simul.create_default_params()
```

Este objeto contiene información sobre:

- El tamaño del dominio espacial.
- La resolución numérica.
- El paso de tiempo.
- Las constantes físicas del modelo.

5. Ciclo de vida de una simulación

Una simulación típica en Fluidsim sigue siempre el mismo flujo:

1. Creación del objeto de parámetros.
2. Inicialización del solver.
3. Integración temporal.
4. Escritura de resultados.

Es decir:

```
#Creación del objeto
sim = Simul(params)
#Se inicia la simulación
sim.time_stepping.start()
```

Al ejecutar `start()`, Fluidsim se encarga automáticamente de iterar en el tiempo, aplicar los operadores espectrales y guardar los resultados según la configuración.

6. Ejemplo mínimo completo

En esta sección se explicará un ejemplo básico de forma progresiva. La idea es dar a entender cómo se hacen las simulaciones con esta herramienta, en general el algoritmo a seguir es el mismo, y consiste principalmente en cómo manipular los parámetros de un solver. Por suerte el resto del trabajo lo maneja el Simulador.

Primero se elige el modelo físico con el que vamos a trabajar, digamos que en este caso nos interesa trabajar con Navier-Stokes2D para un fluido incompresible. Para ello debemos importar `Simul`, que es el objeto principal de una simulación:

```
from fluidsim.solvers.ns2d.solver import Simul
```

Ahora viene una de las partes más importantes, configurar los parámetros de nuestra simulación. Aquí podemos modificar elementos como: Dominio espacial, Resolución, integración temporal y parámetros físicos como la viscosidad y forzamientos:

```
#Creamos el objeto params
params = Simul.create_default_params()
#Ahora podemos modificar cosas:

#Puntos de discretización
params.oper(nx = 64
           ny = 64)

#Tiempo final de la simulación (Empieza en t=0)
params.time_stepping.t_end = 5.0
```

Es importante recalcar todo el trabajo que Fluidsim hace por debajo, en este caso el paso del tiempo lo calcula el simulador según la estabilidad numérica.

Ahora le damos vida a la simulación como objeto:

```
sim = Simul(params)
```

Aquí suceden muchas cosas de forma interna: se construye la malla espectral, se inicializan campos, se configuran operadores FFT, se crean los manejadores de salida.

Finalmente:

```
sim.time_stepping.start()
```

Aquí se inicia el bucle de integración temporal. Fluidsim ahora: avanza en el tiempo, resuelve las ecuaciones, guarda los datos e imprime información en pantalla.

Conceptualmente, el simulador “cobra vida”.

El código completo se verá así:

```
from fluidsim.solvers.ns2d.solver import Simul

params = Simul.create_default_params()

params.oper(nx = 64
            , ny = 64
            , time_stepping.t_end = 5.0)

sim = Simul(params)
sim.time_stepping.start()
```

Con este entendimiento de la estructura básica de una simulación, se le recomienda al usuario, recrear los tutoriales y ejemplos que presenta la [documentación oficial](#).

7. Ejecución, salida y post-procesamiento

Fluidsim guarda los resultados automáticamente en un directorio de salida, el cual contiene archivos con los campos simulados.

Estos archivos pueden analizarse posteriormente mediante scripts en Python o herramientas como Jupyter Notebook.