

MFCyE - Proyecto

Método de volúmenes finitos

8 de marzo de 2022

La empresa Zeleros ha conseguido desarrollar casi por completo su sistema de transporte de tipo Hyperloop, pero se ha encontrado con un pequeño problema extra que ha de solucionar: la refrigeración de las unidades de levitación electromagnética. Debido a la baja densidad del aire en el interior del conducto en el que se desplaza el vehículo, la refrigeración convectiva puede ser tan baja que haga falta forzar el movimiento de un flujo de refrigerante por los electroimanes. El caso es que está dedicando todos sus recursos al desarrollo de los algoritmos de control y a afinar el diseño aerodinámico interno del sistema propulsivo, por lo que externaliza el dimensionado del sistema de refrigeración. Es una gran oportunidad para demostrar vuestra capacidad ingenieril, así que os ponéis manos a la obra para preparar un proyecto que presentar. ¡A ver si os contratan!

Tras las primeras reuniones para decidir cómo proceder, os dais cuenta de que necesitaréis realizar simulaciones mediante CFD para poder definir por completo el sistema. Queréis evitar invertir grandes cantidades de dinero en licencias, además de reducir el *vendor locking*, por lo que primero os decidís por una aproximación de software libre al problema. El caso es que, además, querríais desarrollar vuestro propio código, para poder tenerlo en vuestro portafolio para trabajos futuros. Como primer objetivo, vais a programar la ecuación de conservación de la energía en un dominio bidimensional de geometría arbitraria, resolviendo la misma mediante volúmenes finitos. Para esta primera versión, supondréis que el fluido de trabajo es incompresible y que se mueve a una velocidad dependiente del tiempo que habrá sido obtenida mediante otros medios (ya programaréis esa parte en otro momento). Las propiedades del fluido, la malla, el método de integración temporal y las condiciones de contorno han de ser seleccionables, para tener un sistema flexible que pueda adaptarse a las necesidades de las fases posteriores del proyecto. El código ha de ser válido para cualquier número de condiciones de contorno, además, y el dominio de cálculo puede tener forma arbitraria. La ecuación que hay que resolver es la siguiente, junto con unas condiciones iniciales dependientes de la posición y el campo de velocidades dependiente de la posición y del tiempo (notad que las variables en negrita denotan vectores):

$$\rho \cdot c_v \cdot \frac{\partial T}{\partial t} + \rho \cdot c_v \cdot \nabla \cdot (\mathbf{u} \cdot T) - \nabla \cdot (k \cdot \nabla T) = 0$$

$$T(\mathbf{x}, 0) = f(\mathbf{x})$$

$$\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$$

donde ρ es la densidad del fluido, c_v es el calor específico, T es la temperatura, t es el tiempo, \mathbf{u} es el vector velocidad (dependiente de la posición y del tiempo), k es la conductividad térmica, \mathbf{x} es el vector posición y f es una función arbitraria dependiente de la posición.

Para las primeras pruebas que presentaréis a Zeleros, **escribiréis un informe de un máximo de 3500 palabras (más figuras y anexos con el código) sobre peculiaridades encontradas durante el proyecto.** Algunas cosas interesantes pueden ser el efecto del orden del integrador, del tamaño de la malla en diferentes direcciones, estimar la estabilidad numérica. . . El equipo de ingeniería de cohetes os calificará el proyecto. Para obtener la calificación máxima, será necesario lo siguiente:

- El código estará escrito en C, C++, Fortran, Java, Julia, Python o MATLAB/GNU Octave. Para más información sobre bibliotecas utilizables, lo mejor que podéis hacer es consultar con el equipo docente de la asignatura de Mecánica de Fluidos Computacional y Experimental.
- El código será lo suficientemente modular como para que se puedan **seleccionar las condiciones de contorno, el integrador temporal, la función que obtenga el campo de velocidades y el método de cálculo de flujos convectivos y conductivos por parte del usuario.**
- Toda la selección descrita en el punto anterior se realizará en la llamada principal al simulador, y **tendrá que poderse hacer sin necesidad de usar una interfaz gráfica para poder automatizar el cálculo y sin que éste tenga que modificar el cuerpo del programa.**
- La solución, tras la integración temporal, **tendrá que converger a la solución correcta del problema** (con las limitaciones esperables debidas a la malla y al orden de los esquemas numéricos utilizados).
- Todas las funciones, métodos, variables y clases contarán con **documentación apropiada**, dejando claras sus interfaces.
- Se implementarán **dos métodos para calcular los flujos convectivos y otros dos para calcular los flujos conductivos.**
- El problema será resoluble con **integradores implícitos**, por lo que tendrá que expresarse en forma matricial.
- A cada paso de integración, **la matriz del sistema tendrá que recalcularse y no se podrá suponer que es constante.**
- Las figuras contarán con **calidad suficiente.**
- Cualquier referencia utilizada será nombrada en la **bibliografía** del trabajo.
- Se rellenarán todos los apartados que aparecen en la **plantilla** del trabajo.
- El análisis de resultados se realizará correctamente, y **tendrá que ver con detalles sobre la resolución numérica.**
- Se incluirá un **análisis de coste de cálculo** y un **análisis de convergencia.**
- Se verificará la solución con las de una solución proporcionada por el equipo docente para un caso simple.
- En el caso de que se programe en MATLAB/GNU Octave, se usará exactamente la misma estructura de código trabajada durante la primera sesión de prácticas.
- Se usarán los datos de la malla y la asignación de superficies para las condiciones de contorno en el formato disponible en PoliformaT.