

TESIS CARRERA DE GRADO EN INGENIERÍA
NUCLEAR

**IMPLEMENTACIÓN Y VALIDACIÓN DEL
TRANSPORTE DE ESCALARES POR FLUIDOS EN UN
CÓDIGO DE ELEMENTOS FINITOS.**

Guido Giuntoli

Dr. Mariano Cantero
Director

Dr. Enzo Dari
Co-director

Miembros del Jurado

Dr. Claudio Padra

Dra. Lourdes Torres

Junio de 2015

Mecánica Computacional – Centro Atómico Bariloche

Instituto Balseiro
Universidad Nacional de Cuyo
Comisión Nacional de Energía Atómica
Argentina

A mi tía Dani y a mi abuelo Angel.

Índice de contenidos

Índice de contenidos	v
Resumen	vii
Abstract	ix
1. Introducción	1
1.1. Motivación	1
1.2. Esquema del trabajo	2
2. Resolución numérica de las ecuaciones de Navier-Stokes	3
2.1. Ecuaciones de Navier-Stokes	3
2.2. El método de los elementos finitos	4
2.3. Implementación computacional	8
2.4. Integración numérica	10
2.5. Procesamiento en paralelo	11
2.6. Validación	12
3. La ecuación de advección-difusión	15
3.1. Formulación	15
3.2. El esquema de pasos fraccionados	16
3.3. El método de advección semi-lagrangiana	17
3.4. Elementos finitos en la ecuación de difusión	19
3.5. Condiciones de contorno	21
3.5.1. Condición de Dirichlet	22
3.5.2. Condición de Neumann	22
3.5.3. Condición de Robin	22
3.6. Resultados en 1D	24
4. Implementación eficiente del transporte	29
4.1. Esquema numérico	29
4.2. Advección semi-lagrangiana eficiente	30

4.3. Procesamiento en paralelo de la advección	32
4.4. Validación	33
4.5. Evaluación del tiempo de cálculo	36
5. Conclusiones	39
5.1. Comentarios finales	39
5.2. Trabajo futuro	40
A. Declaración de actividades	43
A.1. Actividades de práctica profesional supervisada	43
A.2. Actividades de proyecto y diseño	43
Bibliografía	45
Índice de figuras	47
Índice de tablas	49
Agradecimientos	51

Resumen

En este trabajo se evaluó y validó un código de elementos finitos para la resolución de las ecuaciones de Navier-Stokes, el cual está siendo desarrollado en conjunto por el Departamento de Mecánica Computacional de la Comisión Nacional de Energía Atómica de Argentina y la Universidad de San Pablo, Brazil. Además se implementó y validó un esquema numérico, capaz de simular el transporte de escalares en fluidos para problemas de ingeniería. El esquema implementado es un esquema de pasos fraccionados, en donde se aplica primero un paso de advección a través de la técnica de advección semi-lagrangiana y luego un paso de difusión a través del método de elementos finitos. Con el objetivo de lograr un código eficiente que sirva para resolver problemas de gran tamaño, el código se implementó con procesamiento en paralelo.

Palabras clave: ELEMENTOS FINITOS, NAVIER-STOKES, ADVECCIÓN SEMI-LAGRANGIANA

Abstract

A finite element code for solving Navier-Stokes equations was evaluated and validated. The code is a joint development between the Department of Computational Mechanics of “Comisión Nacional de Energía Atómica” of Argentina and the University of San Pablo, Brazil. Part of this work was devoted at implementing a numerical scheme for the transport of a scalar property. This was achieved by a fractional stepping in which advection is treated with a semi-lagrangian scheme and diffusion with a traditional finite element scheme. The code is implemented with parallel processing in order to allow for solving large scale problems efficiently.

Keywords: NAVIER-STOKES, FINITE ELEMENTS, SEMI-LAGRANGIAN ADVECTION

Capítulo 1

Introducción

“[...] Al fin largaron y se terminaron las discusiones sobre qué hacer, a que ritmo largar y cómo plantear la ejecución de la carrera. [...]”

— Crónica del Maratón Tres Ciudades Patagónicas 2009, Nicolás Ternavasio.

1.1. Motivación

El desarrollo de nuevas tecnologías ha generado una demanda creciente en el modelado y análisis de sistemas cada vez más complejos. En ingeniería muchos de estos problemas pertenecen al campo de la Mecánica de Fluidos y en los mismos se ve involucrado el transporte de propiedades escalares. Ejemplos de estos problemas son: el flujo de dos fases, en donde la concentración de la fase dispersa se modela como un escalar ([1]), la transferencia de calor donde se modela el transporte de la temperatura ([2]) o el transporte de contaminantes en el medio ambiente ([3]). La gran mayoría de estos problemas carecen de una solución analítica, por lo que es necesario resolverlos con técnicas numéricas.

Los avances en el desarrollo de nuevas técnicas numéricas y las implementaciones del procesamiento distribuido han contribuido a que el modelado numérico y la simulación asistida por computadoras se convierta en una herramienta de gran utilidad en las distintas etapas de diseño en ingeniería. Estos problemas en general implican altos Números de Reynolds (Re), en estas circunstancias existe un amplio rango de escalas espaciales y temporales involucradas que hacen que el costo computacional sea alto. Así, para el caso de la simulación directa de turbulencia (DNS por sus siglas en Inglés), el costo computacional crece aproximadamente con Re^3 . Existen otras técnicas numéricas como son la simulación de grandes vórtices (o LES, “Large Eddy Scale” en Inglés)

o la simulación de las ecuaciones promediadas (o RANS, “Reynolds-averaged Navier Stokes” en Inglés) las cuales permiten resolver problemas con mayor Re a menor costo pero con un menor grado de detalle que la simulación directa.

En el grupo de mecánica computacional del Centro Atómico Bariloche se cuenta con una basta experiencia en desarrollo de códigos de cálculo para el análisis e investigación de problemas de fluidos. En colaboración con un grupo de investigación de la Universidad de San Pablo (USP), Brazil, se está desarrollando un código de elementos finitos con el objetivo de crear una herramienta de cálculo que utilice la técnica DNS para resolver problemas con geometrías complejas.

El objetivo de este trabajo es avanzar en la validación del código de cálculo realizando simulaciones en dos y tres dimensiones de problemas conocidos y comparándolos con los ya publicados en la literatura. Esto último demanda un análisis detallado del código y de las técnicas numéricas empleadas. Por último, se busca implementar sobre el código el cálculo numérico del transporte de escalar utilizando el método de advección semi-lagrangiana a través del procesamiento distribuido.

1.2. Esquema del trabajo

En el capítulo 2 se explica el método de los elementos finitos implementado en el código para resolver las ecuaciones de Navier-Stokes en flujos incompresibles. Al final del capítulo se comparan los resultados calculados con el código con los de otras literaturas.

En el capítulo 3, se plantea una forma de implementar un método para la resolución numérica de la ecuación de advección-difusión. Se realiza un análisis sobre la aplicación de las condiciones de contorno y se reportan los resultados en problemas 1D utilizando el esquema planteado.

En el capítulo 4 se expone la forma en la que se implementó computacionalmente el método para la resolución de la ecuación de transporte de escalares. En el mismo se contrastan los resultados del código con algunos problemas conocidos de ingeniería que involucran el transporte de escalares y se evalúa el rendimiento en tiempo de cómputo.

Finalmente, en el capítulo 5 se resumen las conclusiones principales de este trabajo y se da la línea de trabajo futura a seguir en lo referido al mismo.

Capítulo 2

Resolución numérica de las ecuaciones de Navier-Stokes

“[...] Cuando ya llevaban mas de veinte minutos de carrera, pasaron por la marca del kilómetro seis y de pronto el Pampeano Giménez salió disparado hacia delante, como un noqueador que decreta que se acabó el round de estudio y empieza a tirar mamporros.[...]”

— Crónica del Maratón Tres Ciudades Patagónicas 2009, Nicolás Ternavasio.

En este capítulo se analiza la implementación del método de los elementos finitos utilizado para resolver numéricamente las ecuaciones de Navier-Stokes (N-S) a través del uso de la técnica de procesamiento en paralelo. Al final del capítulo se realiza la validación del código a través de la comparación de resultados conocidos de la literatura.

2.1. Ecuaciones de Navier-Stokes

Las ecuaciones de N-S son las que gobiernan la dinámica de fluidos. Estas se deducen a partir del principio de conservación de momento y de la masa, para entender la deducción de las mismas a partir de las leyes fundamentales de la física se recomienda ver [4]. En este trabajo se analiza el caso de flujo incompresible que es descrito por las siguientes ecuaciones:

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} - \mu \nabla^2 \mathbf{u} + \nabla p = \mathbf{F}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \quad (2.1)$$

donde (\mathbf{u}, p) son las incógnitas del problema, \mathbf{u} es el campo de velocidad y p el de presión dentro de un dominio Ω y en un intervalo temporal $(0, T)$. Por otra parte ρ es la densidad, μ la viscosidad y \mathbf{F} es la fuerza volumétrica sobre el fluido, estos constituyen los datos del problema.

Las condiciones de contorno para este problema se pueden expresar como:

$$\left\{ \begin{array}{ll} \mathbf{u} = \mathbf{u}_D & \text{en } \Gamma_n \cap \Gamma_t \times (0, T), \\ \mathbf{u} \cdot \mathbf{n} = u_n \quad \mathbf{s} = \boldsymbol{\tau}_t & \text{en } \Gamma_n - \Gamma_n \cap \Gamma_t \times (0, T), \\ \mathbf{n} \times \mathbf{u} \times \mathbf{n} = \mathbf{u}_t \quad r = \tau_n & \text{en } \Gamma_t - \Gamma_n \cap \Gamma_t \times (0, T), \\ \boldsymbol{\tau} = \boldsymbol{\tau}_D & \text{en } \Gamma - (\Gamma_n \cup \Gamma_t) \times (0, T), \end{array} \right. \quad (2.2)$$

donde Γ_n y Γ_t son las porciones del contorno $\Gamma = \partial\Omega$ en las cuales se impondrá condiciones de Dirichlet sobre una de las componentes de la velocidad, es decir, sobre la velocidad normal (u_n) o sobre la velocidad tangencial (\mathbf{u}_t). Por otro lado, \mathbf{s} y r son las tensiones tangenciales y normales en el borde.

Puede verse en las condiciones de contorno 2.2 que en las porciones de Γ donde únicamente se impondrá el valor de una sola de las componentes de la velocidad, se deberá imponer además el valor sobre las tensiones correspondientes a las direcciones complementarias.

Por último, la condición inicial se expresa:

$$\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{con} \quad \nabla \cdot \mathbf{u}_0 = 0 \quad \text{en } \Omega. \quad (2.3)$$

2.2. El método de los elementos finitos

El primer paso del esquema numérico consiste en hallar una discretización temporal de las ecuaciones, para esto se eligió un esquema del tipo Euler implícito. Definiendo una discretización temporal de la forma $\{t_0 \dots t_n, t_{n+1} \dots\}$, las ecuaciones quedan:

$$\left\{ \begin{array}{l} \frac{\rho}{\Delta t} \mathbf{u}^{n+1} + \rho \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1} - \mu \nabla^2 \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{F}^{n+1} + \frac{\rho}{\Delta t} \mathbf{u}^n, \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \end{array} \right. \quad (2.4)$$

donde (\mathbf{u}^n, p^n) y $(\mathbf{u}^{n+1}, p^{n+1})$ son aproximaciones de la solución a tiempo t_n y t_{n+1} respectivamente.

En general, como se puede ver en [5], se pide que $\mathbf{u} \in \mathbf{V}$ y $p \in L$; donde L y \mathbf{V} se definen como:

$$L = \left\{ q : \int_{\Omega} q^2 < \infty \right\} \quad \text{y} \quad \mathbf{V} = \{ \mathbf{w} : [\mathbf{w}]_k \in L(\Omega) \quad \wedge \quad \partial[\mathbf{w}]_k / \partial x_i \in L(\Omega) \}, \quad (2.5)$$

con $k, i = 1, 2, 3$ para los casos en tres dimensiones.

El método de elementos finitos consiste en hallar una formulación variacional o débil de las ecuaciones 2.4. Para ello se multiplica la primer ecuación de 2.4 por una función genérica $\mathbf{w} \in \mathbf{V}_0$ y la segunda ecuación por una función $q \in L_0$, donde:

$$L_0 = \left\{ q : q \in L \text{ y } \int_{\Omega} q = 0 \right\} \quad (2.6)$$

y

$$\mathbf{V}_0 = \{ \mathbf{w} : \mathbf{w} \in \mathbf{V} \text{ y } \mathbf{n} \cdot \mathbf{w} = 0 \text{ en } \Gamma_n \text{ y } \mathbf{n} \times \mathbf{w} = 0 \text{ en } \Gamma_t \}, \quad (2.7)$$

luego las ecuaciones se integran en Ω .

Con la finalidad de lograr resultados más precisos y obtener un sistema de ecuaciones lineales, se sustituye $\mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1}$ por $(\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1} + \frac{1}{2}(\nabla \cdot \mathbf{u}^n) \mathbf{u}^{n+1}$, esto se justifica en [6] y se denomina forma anti-simétrica del término convectivo. Por otro lado las condiciones de contorno 2.2, con esta formulación, se puedan imponer de manera más cómoda si se reescribe $\nabla^2 \mathbf{u} = \nabla \cdot (\nabla \mathbf{u} + \nabla^T \mathbf{u})$.

Con estas consideraciones las ecuaciones terminan quedando:

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\rho}{\Delta t} \mathbf{u}^{n+1} \cdot \mathbf{w} + \int_{\Omega} \rho \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1} + \frac{1}{2} (\nabla \cdot \mathbf{u}^n) \mathbf{u}^{n+1} \right) \cdot \mathbf{w}, \\ - \int_{\Omega} \mu \nabla \cdot (\nabla \mathbf{u}^{n+1} + \nabla^T \mathbf{u}^{n+1}) \cdot \mathbf{w} + \int_{\Omega} \nabla p^{n+1} \cdot \mathbf{w}, \\ = \int_{\Omega} \mathbf{F}^{n+1} \cdot \mathbf{w} + \int_{\Omega} \frac{\rho}{\Delta t} \mathbf{u}^n \cdot \mathbf{w} \quad \forall \mathbf{w} \in \mathbf{V}, \\ \int_{\Omega} \nabla \cdot \mathbf{u}^{n+1} q = 0 \quad \forall q \in L_0. \end{array} \right. \quad (2.8)$$

Aplicando la fórmula de Green al término de fuerzas viscosas y considerando las propiedades del espacio \mathbf{V}_0 , se llega a:

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\rho}{\Delta t} \mathbf{u}^{n+1} \cdot \mathbf{w} + \int_{\Omega} \rho \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1} + \frac{1}{2} (\nabla \cdot \mathbf{u}^n) \mathbf{u}^{n+1} \right) \cdot \mathbf{w}, \\ + \int_{\Omega} \mu (\nabla \mathbf{u}^{n+1} + \nabla^T \mathbf{u}^{n+1}) : \nabla \mathbf{w} + \int_{\Omega} (\nabla \cdot \mathbf{w}) p^{n+1}, \\ = \int_{\Omega} \mathbf{F}^{n+1} \cdot \mathbf{w} + \int_{\Omega} \frac{\rho}{\Delta t} \mathbf{u}^n \cdot \mathbf{w} + \int_{\Gamma - \Gamma_n} r \mathbf{w} \cdot \mathbf{n} + \int_{\Gamma - \Gamma_{\tau}} \mathbf{s} \cdot \mathbf{w} \times \mathbf{n}, \\ \forall \mathbf{w} \in \mathbf{V}_0, \\ \int_{\Omega} \nabla \cdot \mathbf{u}^{n+1} q = 0 \quad \forall q \in L_0, \end{array} \right. \quad (2.9)$$

donde r y \mathbf{s} son:

$$\left\{ \begin{array}{l} r = -p + \mathbf{n} \cdot (\nabla \mathbf{u} + \nabla^T \mathbf{u}) \cdot \mathbf{n} \text{ en } \Gamma - \Gamma_n, \\ \mathbf{s} = \mu \mathbf{n} \cdot (\nabla \mathbf{u} + \nabla^T \mathbf{u}) \times \mathbf{n} \text{ en } \Gamma - \Gamma_{\tau}. \end{array} \right. \quad (2.10)$$

La formulación con velocidad y presión de las ecuaciones de N-S a la que se llegó, también conocida como formulación u-P, constituye un problema de elementos finitos mixtos. En principio los espacios vectoriales que se utilizan para representar a la velocidad y a la presión son distintos. De hecho, como se explica en [5], la existencia de la condición de incompresibilidad hace que estos espacios deban cumplir con la condición LBB para garantizar existencia y unicidad en la solución débil.

Para facilitar la implementación numérica, se eligieron los mismos espacios vectoriales para representar a ambos campos. Como esta elección de espacios no cumple la condición LBB, se realizó una estabilización de las ecuaciones originales incorporando un término a la ecuación de continuidad como se aprecia en la ecuación 2.11.

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\rho}{\Delta t} \mathbf{u}^{n+1} \cdot \mathbf{w} + \int_{\Omega} \rho \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1} + \frac{1}{2} (\nabla \cdot \mathbf{u}^n) \mathbf{u}^{n+1} \right) \cdot \mathbf{w}, \\ + \int_{\Omega} \mu (\nabla \mathbf{u}^{n+1} + \nabla^T \mathbf{u}^{n+1}) : \nabla \mathbf{w} + \int_{\Omega} (\nabla \cdot \mathbf{w}) p^{n+1}, \\ = \int_{\Omega} \mathbf{F}^{n+1} \cdot \mathbf{w} + \int_{\Omega} \frac{\rho}{\Delta t} \mathbf{u}^n \cdot \mathbf{w} + \int_{\Gamma-\Gamma_n} r \mathbf{w} \cdot \mathbf{n} + \int_{\Gamma-\Gamma_{\tau}} \mathbf{s} \cdot \mathbf{w} \times \mathbf{n}, \\ \forall \mathbf{w} \in \mathbf{V}_0, \\ \int_{\Omega} \nabla \cdot \mathbf{u}^{n+1} q = \int_{\Omega} \frac{\tau}{\rho} \nabla p^{n+1} \cdot \nabla q \quad \forall q \in L_0, \end{array} \right. \quad (2.11)$$

donde τ es un parámetro que se ajusta según el problema que se resuelve. A medida que este parámetro tiende a cero las ecuaciones tienden a adoptar su forma original. Este esquema de estabilización se puede ver en [7].

El paso siguiente consiste en encontrar una aproximación para (\mathbf{u}, p) . Llamando (\mathbf{u}^h, p^h) a la solución aproximada y pidiendo $(\mathbf{u}^h, p^h) \in (\mathbf{V}^h, L^h)$ donde $\mathbf{V}^h \subset \mathbf{V}$ y $L^h \subset L$, la misma se puede representar como:

$$\mathbf{u}^h = \sum_i^{N_v} \mathbf{u}_i \alpha_i(\mathbf{x}) \quad \text{y} \quad p^h = \sum_i^{N_p} p_i \beta_i(\mathbf{x}), \quad (2.12)$$

donde $\alpha_i(\mathbf{x})$ y $\beta_i(\mathbf{x})$ son bases de los espacios \mathbf{V}^h y L^h , \mathbf{u}_i y p_i son incógnitas que representan a los campos de velocidad y presión en N_v y N_p puntos del dominio respectivamente. Como se mencionó anteriormente, los espacios de las aproximaciones de cada componente de velocidad es igual al de la presión, de esa forma se consideró $\alpha_i = \beta_i = \psi_i$ y $N_v = N_p = N$.

Reemplazando las aproximaciones 2.12 en las ecuaciones 2.11, se llega al siguiente

sistema de ecuaciones lineales:

$$\left\{ \begin{array}{l} \sum_{i=1}^N u_{k,i}^{n+1} \int_{\Omega} \left[\frac{\rho}{\Delta t} \psi_i \psi_j + \rho (\mathbf{u}^n \cdot \nabla) \psi_i + \frac{\rho}{2} (\nabla \cdot \mathbf{u}^n) \psi_i \cdot \psi_j + \mu (\nabla \psi_i + \nabla^T \psi_i) \cdot \nabla \psi_j \right], \\ \sum_{i=1}^N p_i^{n+1} \int_{\Omega} \frac{\partial \psi_j}{\partial x_k} \psi_i = \sum_i^N \int_{\Omega} F_{k,i}^{n+1} \psi_i \psi_j + \int_{\Omega} \frac{\rho}{\Delta t} u_{k,i}^n \psi_j, \\ + \int_{\Gamma-\Gamma_n} r \mathbf{w} \cdot \mathbf{n} + \int_{\Gamma-\Gamma_\tau} \mathbf{s} \cdot \mathbf{w} \times \mathbf{n} \quad \text{para } \forall j = 1 \dots N \quad k = 1, 2, 3, \\ \sum_{k=1}^3 \sum_{i=1}^N u_{k,i}^{n+1} \int_{\Omega} \nabla \cdot \psi_i \psi_j - \sum_{i=1}^N p_i^{n+1} \int_{\Omega} \frac{\tau}{\rho} \nabla \psi_i \cdot \nabla \psi_j = 0 \quad \text{para } j = 1 \dots N, \end{array} \right. \quad (2.13)$$

donde $u_{k,i} = [\mathbf{u}_i]_k$ representa cada una de las componentes de la velocidad en el nodo i . Este problema puede ser representado como un problema matricial de la forma:

$$\mathbf{A} \mathbf{x} = \mathbf{b}, \quad (2.14)$$

donde \mathbf{A} es una matriz de $4N \times 4N$ y \mathbf{x} y \mathbf{b} son vectores de $4N$ componentes que representan las incógnitas y el vector del lado derecho (RHS) respectivamente.

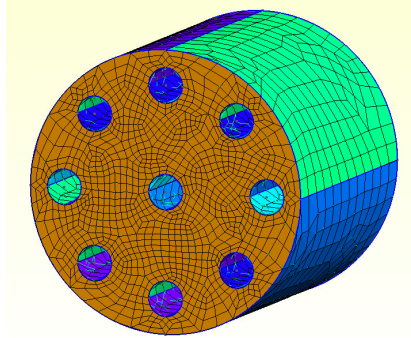


Figura 2.1: Ejemplo de una malla de elementos hexaédricos.

Por otra parte es necesario definir los nodos donde (\mathbf{u}_i^h, p_i^h) representa a la solución, para ello se discretiza el dominio en una malla que cubre el dominio Ω . La misma esta compuesta por N nodos \mathbf{x}_i relacionados entre si por porciones volumétricas llamadas elementos. En este trabajo las mallas implementadas son con elementos hexaédricos tales como los que se aprecian en la figura 2.1.

Finalmente se deben definir las funciones de forma ψ_i , en general se pide que:

$$\psi_i(\mathbf{x}_j) = \delta_{i,j}. \quad (2.15)$$

Por otro lado son localmente polinómicas en los elementos adyacentes al nodo \mathbf{x}_i y se anulan en el resto de los elementos. De esta forma en cada elemento existen 8 funciones de forma que no se anulan dentro del mismo.

La forma funcional de las mismas se define dentro de un elemento local, en este caso es un cubo en las coordenadas $r, s, t \in [-1, 1]$, como se verá, esto facilita la implementación numérica.

Las 8 funciones de forma en el elemento local se definen como:

$$\left\{ \begin{array}{ll} \hat{\psi}_1(r, s, t) = \frac{1}{8}(1+r)(1+s)(1+t), & \hat{\psi}_2(r, s, t) = \frac{1}{8}(1-r)(1+s)(1+t), \\ \hat{\psi}_3(r, s, t) = \frac{1}{8}(1-r)(1-s)(1+t), & \hat{\psi}_4(r, s, t) = \frac{1}{8}(1+r)(1-s)(1+t), \\ \hat{\psi}_5(r, s, t) = \frac{1}{8}(1+r)(1+s)(1-t), & \hat{\psi}_6(r, s, t) = \frac{1}{8}(1-r)(1+s)(1-t), \\ \hat{\psi}_7(r, s, t) = \frac{1}{8}(1-r)(1-s)(1-t), & \hat{\psi}_8(r, s, t) = \frac{1}{8}(1+r)(1-s)(1-t), \end{array} \right. \quad (2.16)$$

donde s, r, t se denominan coordenadas locales.

Para definir las funciones de forma en un elemento genérico se define la siguiente transformación de coordenadas locales a globales:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \sum_{i=1}^8 \hat{\psi}_i(r, s, t) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \quad (2.17)$$

donde $[x_i \ y_i \ z_i]^T$ son las coordenadas de los vértices o nodos del elemento. De esa forma:

$$\psi_i(x, y, z) = \hat{\psi}_i(r, s, t). \quad (2.18)$$

En la figura 2.2 se esquematiza el mapeo de coordenadas locales a globales a través de la transformación 2.17.

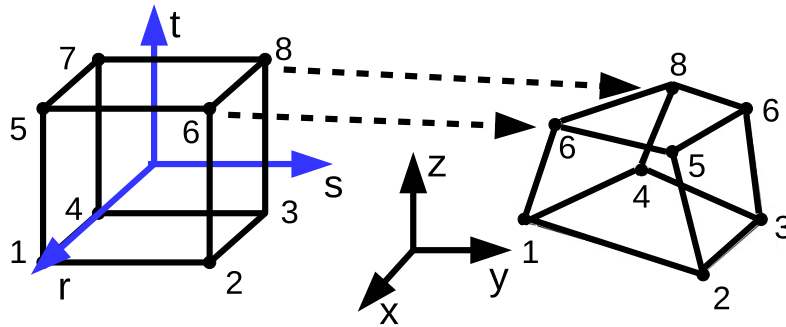


Figura 2.2: Mapeo de coordenadas locales a globales.

2.3. Implementación computacional

Para construir la matriz y el RHS de 2.14, se deben realizar integrales en las que intervienen operaciones entre las funciones de forma en el dominio Ω . La manera de

hacerlo es escribiendo las integrales como:

$$\int_{\Omega} \dots = \sum_e^{N_e} \int_{\Omega_e} \dots, \quad (2.19)$$

donde Ω_e es el volumen de un elemento “ e ” y N_e es el número total de elementos que conforman la malla. De esta forma, para realizar las integrales que constituyen el sistema de ecuaciones 2.13, se recorren cada uno de los elementos de la malla y se construye una matriz elemental cuyos coeficientes son los aportes de cada elemento sobre la integral total. Luego se suman cada uno de los coeficientes de la matriz elemental sobre los coeficientes correspondientes en la matriz global. Para construir el RHS se realiza un procedimiento similar construyendo un vector elemental.

Este proceso de recorrer elementos, armar la matriz elemental y el vector elemental para luego sumarlos en la matriz global y el RHS es lo que se conoce como “ensamblaje”.

En este código los elementos de la malla poseen 8 nodos con 4 incógnitas cada uno, por lo tanto existen 32 incógnitas por elemento, esto implica que la matriz elemental posee un tamaño de 32×32 .

Eligiendo un ordenamiento de las incógnitas de la forma:

$$\mathbf{x} = [u_{x,1} \quad u_{y,1} \quad u_{z,1} \quad p_1 \quad u_{x,2} \quad u_{y,2} \quad u_{z,2} \quad p_2 \dots u_{x,N} \quad u_{y,N} \quad u_{z,N} \quad p_N]^T, \quad (2.20)$$

la matriz elemental queda:

$$A_e = \begin{pmatrix} C_{e,1,1} & 0 & 0 & B_{e,x,1,1} & \dots & C_{e,N_e,1} & 0 & 0 & B_{e,x,N_e,1} \\ 0 & C_{e,1,1} & 0 & B_{e,y,1,1} & \dots & 0 & C_{e,N_e,1} & 0 & B_{e,y,N_e,1} \\ 0 & 0 & C_{e,1,1} & B_{e,z,1,1} & \dots & 0 & 0 & C_{e,N_e,1} & B_{e,z,N_e,1} \\ B_{e,x,1,1} & B_{e,y,1,1} & B_{e,z,1,1} & E_{e,1,1} & \dots & B_{e,x,1,N_e} & B_{e,y,1,N_e} & B_{e,z,1,N_e} & E_{e,N_e,1} \\ \hline C_{e,1,2} & 0 & 0 & B_{e,x,1,2} & \dots & C_{e,N_e,2} & 0 & 0 & B_{e,x,N_e,2} \\ 0 & C_{e,1,2} & 0 & B_{e,y,1,2} & \dots & 0 & C_{e,N_e,2} & 0 & B_{e,y,N_e,2} \\ 0 & 0 & C_{e,1,2} & B_{e,z,1,2} & \dots & 0 & 0 & C_{e,N_e,2} & B_{e,z,N_e,2} \\ B_{e,x,2,1} & B_{e,y,2,1} & B_{e,z,2,1} & E_{e,2,1} & \dots & B_{e,x,2,N_e} & B_{e,y,2,N_e} & B_{e,z,2,N_e} & E_{e,N_e,2} \\ \hline \dots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \hline C_{e,1,N_e} & 0 & 0 & B_{e,x,1,N_e} & \dots & C_{e,N_e,N_e} & 0 & 0 & B_{e,x,N_e,N_e} \\ 0 & C_{e,1,N_e} & 0 & B_{e,y,1,N_e} & \dots & 0 & C_{e,N_e,N_e} & 0 & B_{e,y,N_e,N_e} \\ 0 & 0 & C_{e,1,N_e} & B_{e,z,1,N_e} & \dots & 0 & 0 & C_{e,N_e,N_e} & B_{e,z,N_e,N_e} \\ B_{e,x,N_e,1} & B_{e,y,N_e,1} & B_{e,z,N_e,1} & E_{e,1,N_e} & \dots & B_{e,x,N_e,N_e} & B_{e,y,N_e,N_e} & B_{e,z,N_e,N_e} & E_{e,N_e,N_e} \end{pmatrix},$$

donde:

$$C_{e,i,j} = \int_{\Omega_e} \frac{\rho}{\Delta t} \psi_i \psi_j + \rho (\mathbf{u}^n \cdot \nabla \psi_i) \psi_j + \frac{\rho}{2} (\nabla \cdot \mathbf{u}^n) \psi_i \psi_j + \mu \nabla \psi_i \cdot \nabla \psi_j, \quad (2.21)$$

$$B_{e,x_k,i,j} = \int_{\Omega_e} \psi_i \frac{\partial \psi_j}{\partial x_k}, \quad (2.22)$$

y

$$E_{e,i,j} = \int_{\Omega_e} \frac{\tau}{\rho} \nabla \psi_i \cdot \nabla \psi_j. \quad (2.23)$$

Por otra parte el vector elemental es:

$$b_e = [M_{e,x,1} \quad M_{e,y,1} \quad M_{e,z,1} \quad 0 \dots M_{e,x,N_e} \quad M_{e,y,N_e} \quad M_{e,z,N_e} \quad 0]^T, \quad (2.24)$$

en donde:

$$M_{e,x_k,1} = \int_{\Omega} F_{k,i}^{n+1} \psi_i \psi_j + \int_{\Omega} \frac{\rho}{\Delta t} u_{k,i}^n \psi_j + \int_{\Gamma-\Gamma_n} r \mathbf{w} \cdot \mathbf{n} + \int_{\Gamma-\Gamma_\tau} \mathbf{s} \cdot \mathbf{w} \times \mathbf{n}. \quad (2.25)$$

Finalmente, para averiguar el valor de la solución \mathbf{x} resta resolver el sistema matricial.

2.4. Integración numérica

Para realizar la integración dentro de los elementos y poder armar la matriz elemental, el procedimiento utilizado en este código se basa en utilizar el cambio de coordenadas 2.17 tal como se explica en [8]. De esta forma todas las integrales se pueden hacer dentro del elemento local y así, al implementar las integrales con la cuadratura de Gauss, se utilizan siempre los mismos puntos y pesos de Gauss.

Por ejemplo, para calcular el primer coeficiente del término 2.21 el procedimiento es:

$$\begin{aligned} \int_{\Omega_e} \psi_i \psi_j &= \int_{\Omega_{s,r,t}} \hat{\psi}_i \hat{\psi}_j \det(\mathbf{J}), \\ &= \sum_{gp} \hat{\psi}_i(\mathbf{x}_{gp}) \hat{\psi}_j(\mathbf{x}_{gp}) \det(\mathbf{J}) w_{gp}, \end{aligned}$$

donde \mathbf{x}_{gp} y w_{gp} son los puntos y los pesos de Gauss definidos en el elemento local como:

$$\mathbf{x}_{gp} = (\pm 1/\sqrt{3}, \pm 1/\sqrt{3}, \pm 1/\sqrt{3}), \quad w_{gp} = 1/4, \quad \text{con } gp = 1 \dots 8, \quad (2.26)$$

y \mathbf{J} es la matriz Jacobiana transpuesta, que se define como:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial r} x_i & \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial r} y_i & \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial r} z_i \\ \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial s} x_i & \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial s} y_i & \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial s} z_i \\ \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial t} x_i & \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial t} y_i & \sum_{i=1}^8 \frac{\partial \hat{\psi}_i}{\partial t} z_i \end{pmatrix}. \quad (2.27)$$

Del mismo modo, para calcular las integrales que contienen gradientes de las funciones de forma, se debe aplicar la relación que surge de la regla de la cadena, estas es:

$$\nabla_{r,s,t}\hat{\psi}_i = \mathbf{J}\nabla_{x,y,z}\psi_i. \quad (2.28)$$

Luego, el valor del gradiente transformado al elemento local es:

$$\nabla_{x,y,z}\psi_i = \mathbf{J}^{-1}\nabla_{r,s,t}\hat{\psi}_i. \quad (2.29)$$

Finalmente este tipo de integral se calcula como:

$$\begin{aligned} \int_{\Omega_e} \nabla\psi_i \cdot \nabla\psi_j &= \int_{\Omega_{s,r,t}} (\mathbf{J}^{-1}\nabla_{r,s,t}\hat{\psi}_i) \cdot (\mathbf{J}^{-1}\nabla_{r,s,t}\hat{\psi}_j) \det(\mathbf{J}), \\ &= \sum_{gp} (\mathbf{J}^{-1}\nabla_{r,s,t}\hat{\psi}_i)(\mathbf{x}_{gp}) \cdot (\mathbf{J}^{-1}\nabla_{r,s,t}\hat{\psi}_j)(\mathbf{x}_{gp}) \det(\mathbf{J}) w_{gp}. \end{aligned}$$

2.5. Procesamiento en paralelo

El procesamiento en paralelo es una técnica computacional que permite disminuir los tiempos de cálculo y hace posible la resolución numérica de problemas de gran tamaño. El modelo de programación utilizado en este código es del tipo SPMD (“Single Program Multiple Data” en Inglés). Es decir, cada uno de los procesadores ejecutan el mismo programa pero con diferentes variables y datos. Para hacer esto, en el código, se utilizaron las bibliotecas de PETSc (“Portable, Extensible Toolkit for Scientific Computation” en Inglés).

PETSc es un conjunto de bibliotecas que pueden ser usadas en programas escritos en distintos lenguajes, entre ellos “C”, que es en el que está programado el código. Cada biblioteca manipula una clase particular de objetos, los que se utilizan en este trabajo son matrices, vectores distribuidos y objetos “algorítmicos” como son los resolutores de sistemas lineales, esto se puede ver en detalle en [9]. Estas bibliotecas permiten abstraerse del proceso de intercambio de mensajes ya que el usuario simplemente define la forma en que se particionan los objetos distribuidos y luego las bibliotecas se encargan de realizar las operaciones entre objetos a través de comunicaciones internas con el protocolo MPI (“Message Passing Interface” en Inglés).

La forma de particionar las matrices y los vectores se esquematiza en la figura 2.3, es decir, se asigna un rango de coeficientes de la matriz y de los vectores lo más uniformemente posible a cada proceso. Como las componentes de las matrices y los vectores tienen una relación directa con los nodos de la malla, queda definida, además, una relación entre nodos y procesos. Así, los nodos propios son aquellos que pertenecen a un proceso y los nodos externos son aquellos que no pertenecen.

También es útil definir una relación entre los elementos de la malla y los procesos. De

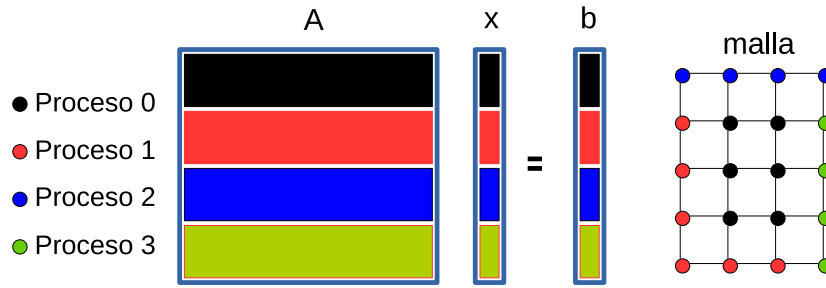


Figura 2.3: Asignación de un rango de componentes de la matriz global y de los vectores a cada proceso.

esta forma, un elemento es propio si al menos uno de sus nodos es propio. Un elemento es externo al proceso si no es propio, es decir, ninguno de sus nodos es propio. Así, por ejemplo, durante el ensamblaje de la matriz global y el RHS cada proceso recorre cada uno de los elementos propios y construye su matriz elemental dejando anulados los bloques que corresponden a nodos externos. Esto reduce el tiempo de ensamblaje conforme se aumenta el número de procesos ya que es una tarea altamente paralelizable.

Una vez ensamblada la matriz global y el vector del RHS, se procede a resolver el sistema. Para la resolución del problema matricial se utilizó el método GMRES (“Generalized Minimal Residual Method” en Inglés), este método está implementado en la biblioteca KSP (“Krilov Space Solver” en Inglés) de PETSc y es llevado a cabo en forma paralela, esto permite reducir el tiempo de cómputo y lograr resolver problemas de mayor tamaño.

2.6. Validación

Para validar la resolución de las ecuaciones de N-S se calculó la solución al problema 2D presentado por [10]. La geometría de este problema es un cuadrado en las coordenadas $x, z \in [0, 1]$ con condiciones de contorno:

$$\begin{cases} u_x = 0, & u_z = 0 & \text{en } x = 0, x = 1, z = 0, \\ u_x = 1, & u_z = 0 & \text{en } z = 1. \end{cases} \quad (2.30)$$

Este código fue desarrollado para la resolución de problemas 3D, por lo tanto, se implementaron mallas de $N \times N \times 1$ elementos prismáticos rectos en las direcciones x, z e y como se ve en la figura 2.4, con condiciones de borde de libre deslizamiento en las paredes $y = 0$ e $y = 1$, es decir:

$$\begin{cases} u_y = 0, & \tau_x = \tau_z = 0 & \text{en } y = 0, y = 1. \end{cases} \quad (2.31)$$

En la forma adimensionalizada de las ecuaciones 2.1, tal como se explica en [4], la

solución estacionaria del problema depende únicamente del número de Reynolds (Re), definido en este caso como:

$$Re = \frac{\rho V L}{\mu}. \quad (2.32)$$

Este problema fue resuelto para con una malla de $N=128$ para los casos con $Re=100$ y 400 ; y con una malla de $N=256$ para el caso de $Re=1000$. La elección en el tamaño de la malla fue igual al trabajo de [10] en el que se utiliza el método de diferencias finitas aplicado a la ecuación de transporte de vorticidad.

Se impuso $\rho = 1$, $V = 1$, $L = 1$ y se ajustó μ de manera de conseguir el Re deseado. Por otra parte, se fijó el parámetro estabilizador $\tau = 1,0e-2$ en base a cálculos previos para estos valores de Re .

Para calcular la solución en estado estacionario se realizaron 100 pasos temporales con $\Delta t = 0,1$ para $Re=100,400$ y 1000 .

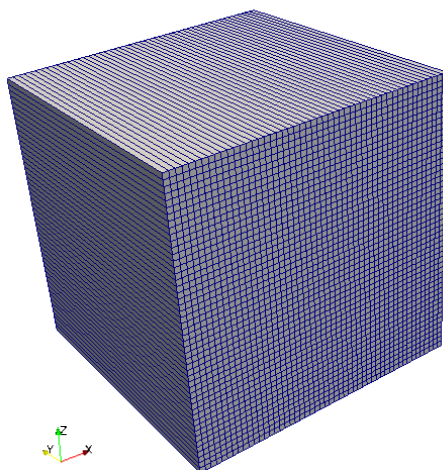


Figura 2.4: Malla utilizada para resolver el problema.

Se calcularon los perfiles de velocidad u_x sobre la línea $x = 0,5$ y los perfiles de u_z sobre $z = 0,5$, los resultados para los distintos Re se muestran en la figura 2.5. En la misma además se muestran los resultados obtenidos en el trabajo [10] y como puede verse los valores en velocidades tienen una buena concordancia.

En la figura 2.6 se comparan los patrones de líneas de corrientes calculadas con el código y las que se obtuvieron en [10] para $Re=100$. En la figura 2.7 se muestran los patrones correspondientes a $Re=400$. Se observa que los patrones de líneas de corriente son similares en ambos casos.

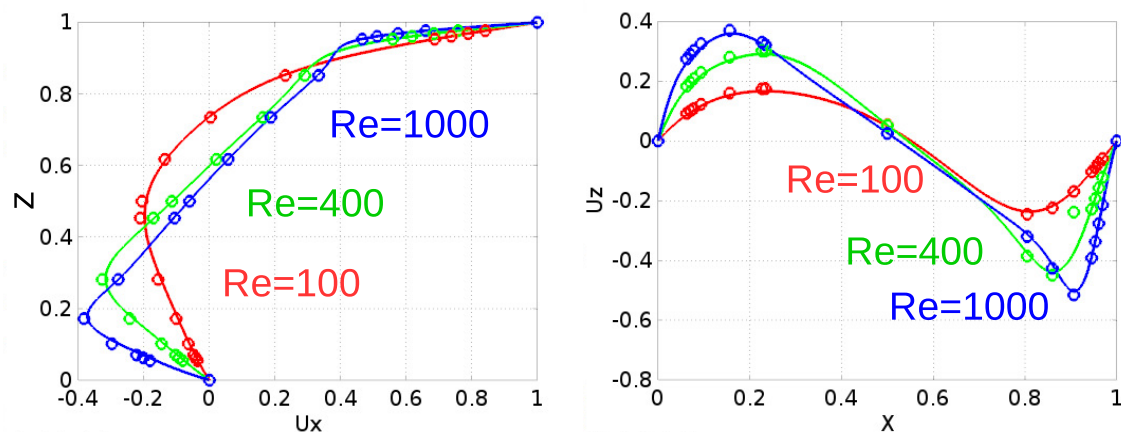


Figura 2.5: A la izquierda, perfil de velocidad u_x sobre la línea $x = 0,5$. A la derecha, perfil de velocidad u_z sobre la línea $z = 0,5$. Con “o” solución obtenida en [10], y en línea continua la solución calculada con el código.

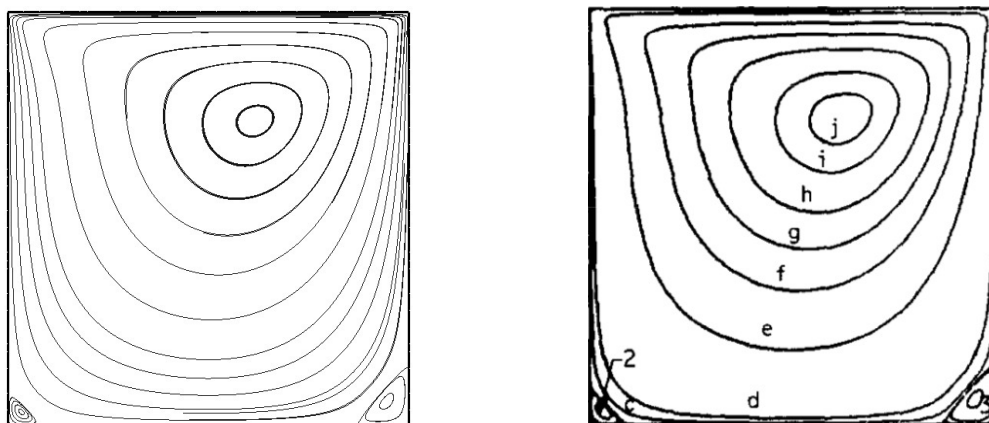


Figura 2.6: Líneas de corrientes obtenidas para $Re=100$. A la izquierda, resultados del código. A la derecha, resultados de [10].

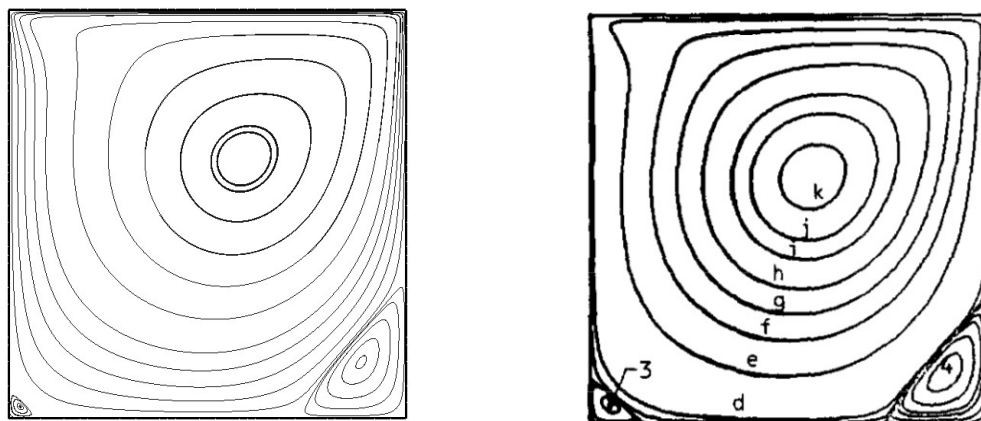


Figura 2.7: Líneas de corrientes obtenidas para $Re=400$. A la izquierda, resultados del código. A la derecha, resultados de [10].

Capítulo 3

La ecuación de advección-difusión

“[...] Pasaron la mitad de la carrera en una hora diecisiete minutos.[...] Las palabras de los demás se habían apagado. Nadie se despegaba del pelotón, pero el se salía de la vaina. [...]”

— Crónica del Maratón Tres Ciudades Patagónicas 2009, Nicolás Ternavasio.

Muchos problemas de ingeniería presentan fenómenos de transporte de escalares por fluidos los cuales se modelan a través de la ecuación de advección-difusión, ver por ejemplo los problemas de transferencia térmica que se plantean en [11]. En este capítulo se explica como se implementa una técnica de resolución numérica para esta ecuación. Básicamente, la técnica consiste en un esquema de pasos fraccionados que combina el método de advección semi-lagrangiana y la difusión a través del método de elementos finitos. Finalmente se plantea una técnica para aplicar las condiciones de contorno en este esquema numérico y se contrastan los resultados en problemas 1D.

3.1. Formulación

Un problema de advección-difusión general, como se puede ver en [12], se puede plantear de la siguiente manera:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - \nabla \cdot (\mathbf{D} \cdot \nabla \phi) + \sigma \phi = S \text{ en } \Omega, \quad (3.1)$$

con condición de contorno

$$\alpha \hat{\mathbf{n}} \cdot \nabla \phi + \beta(\phi - \phi_D) = \gamma \text{ en } \Gamma, \quad (3.2)$$

y condición inicial

$$\phi(\mathbf{x}, t = 0) = \phi_0(\mathbf{x}), \quad (3.3)$$

donde $\phi(\mathbf{x}, t)$ es la cantidad escalar que se transporta dentro de un fluido con velocidad \mathbf{u} confinado en un dominio Ω . Se considera que $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ es un dato para este problema. Por otro lado \mathbf{D} es la matriz diagonal de difusión, σ el coeficiente de reacción, S es la fuente del escalar y α , β y γ son constantes.

El significado físico de cada término es el siguiente:

- $\mathbf{u} \cdot \nabla \phi$: es el término de advección, responsable de transportar al escalar por el movimiento del fluido.
- $\nabla \cdot (\mathbf{D} \cdot \nabla \phi)$: es el término difusivo, hace que el escalar se propague por la existencia de su gradiente.
- $\sigma \phi$ y S constituyen el término de reacción y el término fuente, respectivamente.

En este trabajo no se consideran los términos fuente y de reacción, es decir, $\sigma = 0$ y $S = 0$. Además se supone que la difusión es isotrópica y uniforme, de esa forma, $\nabla \cdot (\mathbf{D} \cdot \nabla \phi) = D \nabla^2 \phi$. Con estas consideraciones, la ecuación de transporte queda:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - D \nabla^2 \phi = 0 \text{ en } \Omega. \quad (3.4)$$

3.2. El esquema de pasos fraccionados

Para la resolución del problema de transporte se implementó un esquema de pasos fraccionados (ó “operator splitting method” en Inglés), este consiste en dividir el operador de transporte en dos partes: un operador de advección \mathbf{L}_1 y otro de difusión \mathbf{L}_2 , es decir:

$$\mathbf{L} = \underbrace{\mathbf{u} \cdot \nabla \phi}_{\mathbf{L}_1} + \underbrace{(-D \nabla^2 \phi)}_{\mathbf{L}_2} = \mathbf{L}_1 + \mathbf{L}_2. \quad (3.5)$$

Luego se aplica cada uno de los operadores en forma secuencial, uno a continuación del otro, es decir:

$$1) \begin{cases} \frac{\partial \phi}{\partial t}(\mathbf{x}, t) + \mathbf{L}_1 \phi(\mathbf{x}, t) = 0 & \text{en } \Omega \times [t_n, t_{n+1}), \\ \phi(\mathbf{x}, t_n) = \phi^n(\mathbf{x}), \end{cases} \quad (3.6)$$

$$2) \begin{cases} \frac{\partial \phi}{\partial t}(\mathbf{x}, t) + \mathbf{L}_2 \phi(\mathbf{x}, t) = 0 & \text{en } \Omega \times [t_n, t_{n+1}), \\ \phi(\mathbf{x}, t_n) = \phi^*(\mathbf{x}), \\ \phi^{n+1}(\mathbf{x}) = \phi(\mathbf{x}, t_{n+1}). \end{cases} \quad (3.7)$$

La primer ecuación a resolver es la ecuación de advección pura, que aquí planteamos resolverla con el método de advección semi-lagrangiana. La segunda ecuación es la de difusión, para este caso, abordamos el problema con el método de elementos finitos. La ventaja principal que posee este esquema frente a otros de un solo paso, como se ejemplifica en [12], es que evita el problema de las oscilaciones (“overshooting and undershooting problem” en Inglés) que se puede presentar en la solución del escalar en regiones de altos gradientes.

La elección en el orden en los pasos, es decir, primero el de advección y luego el de difusión se debe a que de esa manera es posible imponer efectivamente las condiciones de contorno. Este argumento se basa en que la ecuación de advección no requiere condiciones de contorno en las zonas de flujo saliente ($\mathbf{u} \cdot \mathbf{n} > 0$). Si se aplicasen en el orden inverso, luego del paso de advección, el valor del escalar quedaría sin ningún valor impuesto o “flotado” en el borde de flujo saliente.

3.3. El método de advección semi-lagrangiana

La ecuación de advección del problema 3.6 es :

$$\frac{\partial \phi}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \phi = 0, \quad (3.8)$$

con condición inicial:

$$\phi(\mathbf{x}, t_n) = \phi^n(\mathbf{x}). \quad (3.9)$$

Esta es una ecuación hiperbólica, la misma posee solución exacta y es el punto clave que permite construir el esquema numérico.

Para conocer la solución exacta se define el concepto de línea característica. Una línea característica $\mathbf{X}(\mathbf{x}, s; t)$, como se puede ver en [13], es aquella que satisface la ecuación:

$$\frac{d\mathbf{X}(\mathbf{x}, s; t)}{dt} = \mathbf{u}(\mathbf{X}(\mathbf{x}, s; t), t), \quad (3.10)$$

con condición inicial:

$$\mathbf{X}(\mathbf{x}, s; s) = \mathbf{x}. \quad (3.11)$$

Esta representa la trayectoria que sigue una partícula inmersa en un fluido con un campo de velocidad $\mathbf{u}(\mathbf{x}, t)$ que a tiempo $t = s$ está en la posición \mathbf{x} .

Las líneas características se pueden calcular integrando la ecuación 3.10 de la forma:

$$\mathbf{X}(\mathbf{x}, s; t) - \mathbf{x} = \int_s^t \mathbf{u}(\mathbf{X}(\mathbf{x}, s; \tau), \tau) d\tau. \quad (3.12)$$

Así, la solución exacta de la ecuación de advección 3.8 en un intervalo temporal

$[t_n, t_{n+1})$, como puede verse en [12], se escribe:

$$\phi(\mathbf{x}, t_{n+1}) = \phi(\mathbf{X}(\mathbf{x}, t_{n+1}; t_n), t_n). \quad (3.13)$$

Es decir que el valor del escalar a tiempo t_{n+1} es aquel que había a tiempo t_n desplazado a lo largo de las líneas características en ese paso temporal.

La implementación numérica del método de advección semi-lagrangiana consiste primero en discretizar el dominio Ω en una malla de nodos y elementos como la de la figura 3.1. Si \mathbf{x}_i son los nodos que conforman dicha malla y donde se desea conocer la solución $\phi_i^{n+1} = \phi(\mathbf{x}_i, t_{n+1})$, entonces en base a la ecuación 3.13 se tiene que:

$$\phi_i^{n+1} = \phi^n(\mathbf{x}_i^*), \quad (3.14)$$

donde $\phi^n(\mathbf{x}_i^*)$ representa la interpolación de ϕ^n en el punto de partida \mathbf{x}_i^* . Este se calcula con la fórmula 3.12 adaptada al intervalo temporal $[t_n, t_{n+1})$, es decir:

$$\mathbf{x}_i^* = \mathbf{x}_i - \int_{t_n}^{t_{n+1}} \mathbf{u}(\mathbf{X}(\mathbf{x}_i, t_{n+1}, \tau), \tau) d\tau. \quad (3.15)$$

La forma de obtener el valor interpolado $\phi^n(\mathbf{x}_i^*)$ es identificando el elemento de la malla al cual pertenece el punto de partida \mathbf{x}_i^* para luego interpolar el valor de ϕ^n con los valores nodales del elemento ϕ_j^n , esto se ejemplifica en la figura 3.1 para un caso 2D.

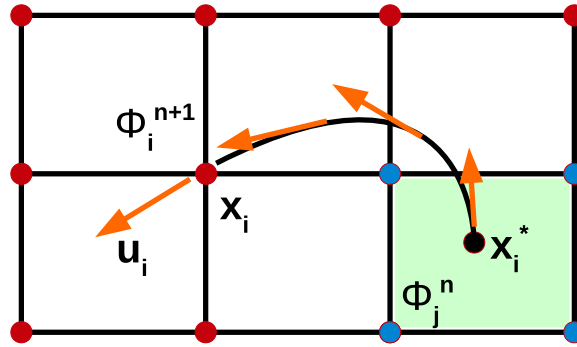


Figura 3.1: Representación del algoritmo de advección semi-lagrangiana para un caso 2D.

Para calcular los puntos de partida \mathbf{x}_i^* numéricamente resolviendo la ecuación 3.15 y conociendo los valores nodales de velocidad \mathbf{u}_i^{n+1} , se utiliza la siguiente fórmula:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - (\Delta t)_k \mathbf{u}^{n+1}(\mathbf{x}_k) & \text{con } k = 1 \dots K, \\ \mathbf{x}_1 = \mathbf{x}_i & (\Delta t)_k = \Delta t / K, \end{cases} \quad (3.16)$$

donde $\mathbf{u}^{n+1}(\mathbf{x}_k)$ es el valor interpolado de \mathbf{u}_i^{n+1} en \mathbf{x}_k y al final $\mathbf{x}_i^* = \mathbf{x}_{K+1}$.

En este trabajo el cálculo de \mathbf{x}^* se realiza a través de la fórmula 3.16 con $K = 1$,

de esta forma se evita el problema de tener que interpolar el valor de velocidad en los puntos \mathbf{x}_k que no sean nodos. Esto conlleva a que, si se utilizan pasos de tiempo demasiado grandes, el cálculo de los puntos de partida puede tener un error significativo, principalmente cuando el gradiente de velocidad es grande. Esto se aprecia en la figura 3.2, donde el uso de $K = 1$ puede dar un resultado con un error inaceptable, es decir, el punto de partida aproximado cae fuera del dominio.

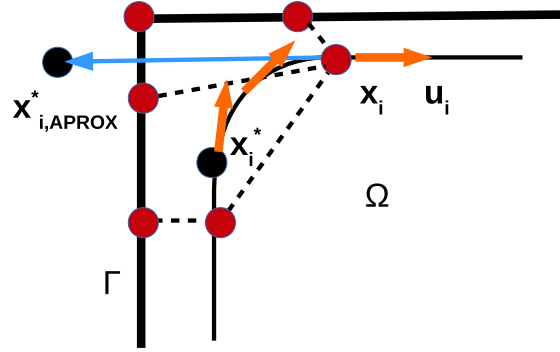


Figura 3.2: Ejemplo con un error considerable en el cálculo de $\mathbf{x}_{i,aprox}^*$ utilizando la fórmula 3.16 con $K = 1$. El valor exacto \mathbf{x}_i^* está dentro del dominio Ω .

3.4. Elementos finitos en la ecuación de difusión

La ecuación de difusión del problema 3.7 es:

$$\frac{\partial \phi}{\partial t} - D \nabla \cdot (\nabla \phi) = 0, \quad (3.17)$$

con condición inicial:

$$\phi(\mathbf{x}, t_n) = \phi^n(\mathbf{x}). \quad (3.18)$$

Esta ecuación se resuelve con el método de elementos finitos aprovechando las estructuras y rutinas empleadas para la resolución de las ecuaciones de N-S. En la discretización temporal se implementó un esquema Euler implícito, de esta forma, la ecuación queda:

$$\phi^{n+1} - D \Delta t \nabla \cdot \nabla \phi^{n+1} = \phi^n. \quad (3.19)$$

Como se explica en [13], se pide que $\phi^{n+1} \in V$, donde:

$$V = \{q : q \in L \text{ y } \partial q / \partial x_i \in L\}, \quad (3.20)$$

con $i = 1, 2, 3$ para un caso en tres dimensiones y siendo L es el espacio definido en 2.5.

Luego se procede a buscar una formulación débil del problema, para ello se multi-

plica la ecuación 3.19 por una función $q \in V_0$, donde:

$$V_0 = \{q : q \in V \text{ y } q = 0 \text{ en } \mathbf{x} \in \Gamma_D\}, \quad (3.21)$$

siendo Γ_D la porción de la frontera donde se impondrán las condiciones de Dirichlet. Integrando en Ω se tiene:

$$\int_{\Omega} \phi^{n+1} q - D\Delta t \int_{\Omega} \nabla \cdot \nabla \phi^{n+1} q = \int_{\Omega} \phi^n q \quad \forall q \in V_0. \quad (3.22)$$

Aplicando la fórmula de Green se llega a la siguiente formulación débil:

$$\int_{\Omega} \phi^{n+1} q + D\Delta t \int_{\Omega} \nabla \phi^{n+1} \cdot \nabla q = \int_{\Omega} \phi^n q + D\Delta t \int_{\Gamma_N} \mathbf{n} \cdot \nabla \phi^{n+1} q \quad \forall q \in V_0, \quad (3.23)$$

donde Γ_N es la porción de Γ donde se impondrán las condiciones de Neumann.

El paso siguiente es aproximar la solución débil ϕ con $\phi^h \in V^h \subset V$ de forma tal de que esta última se pueda expresar como:

$$\phi^h(\mathbf{x}) = \sum_i^N \phi_i \psi_i(\mathbf{x}), \quad (3.24)$$

donde las funciones de forma $\psi_i(\mathbf{x})$ que se utilizan para discretizar la solución son las mismas que se definieron en el capítulo 2 para representar los campos de velocidad y presión.

Reemplazando la aproximación 3.24 en 3.23 se llega al siguiente sistema lineal de ecuaciones:

$$\begin{aligned} \sum_i^N \left[\phi_i^{n+1} \int_{\Omega} \psi_i \psi_j + D\Delta t \phi_i^{n+1} \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j \right] &= \sum_i^N \phi_i^n \int_{\Omega} \psi_i \psi_j \\ + D\Delta t \int_{\Gamma_N} \mathbf{n} \cdot \nabla \phi^{n+1} \psi_j &\quad \text{con } j = 1, 2 \dots N, \end{aligned} \quad (3.25)$$

que escrito en forma matricial es:

$$(\mathbf{M} + D\Delta t \mathbf{K}) \phi = \mathbf{M} \phi^n + \mathbf{f}. \quad (3.26)$$

El ordenamiento de las incógnitas se eligió de la forma:

$$\phi = [\phi_1 \quad \phi_2 \quad \dots \quad \phi_N]^T. \quad (3.27)$$

En 3.26:

$$\mathbf{M}_{i,j} = \int_{\Omega} \psi_i \psi_j \quad , \quad \mathbf{K}_{i,j} = \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j, \quad (3.28)$$

y

$$\mathbf{f}_j = \int_{\Gamma_N} \mathbf{n} \cdot \nabla \phi^{n+1} \psi_j, \quad (3.29)$$

el cual depende de las condiciones de contorno.

Al utilizar las mismas funciones de forma que en el problema de N-S se tiene una variable por cada nodo y de esa forma la matriz elemental queda de un tamaño de 8×8 . La misma se puede expresar como:

$$\mathbf{A}_e = \begin{pmatrix} \int_{\Omega_e} [\psi_1 \psi_1 + \nabla \psi_1 \cdot \nabla \psi_1] & \int_{\Omega_e} [\psi_2 \psi_1 + \nabla \psi_2 \cdot \nabla \psi_1] & \dots & \int_{\Omega_e} [\psi_8 \psi_1 + \nabla \psi_8 \cdot \nabla \psi_1] \\ \int_{\Omega_e} [\psi_1 \psi_2 + \nabla \psi_1 \cdot \nabla \psi_2] & \int_{\Omega_e} [\psi_2 \psi_2 + \nabla \psi_2 \cdot \nabla \psi_2] & \dots & \int_{\Omega_e} [\psi_8 \psi_2 + \nabla \psi_8 \cdot \nabla \psi_2] \\ \vdots & \vdots & \ddots & \vdots \\ \int_{\Omega_e} [\psi_1 \psi_8 + \nabla \psi_1 \cdot \nabla \psi_8] & \int_{\Omega_e} [\psi_2 \psi_8 + \nabla \psi_2 \cdot \nabla \psi_8] & \dots & \int_{\Omega_e} [\psi_8 \psi_8 + \nabla \psi_8 \cdot \nabla \psi_8] \end{pmatrix}.$$

Por otro lado el vector elemental, con un tamaño de 8 coeficientes, es:

$$\mathbf{b}_e = \begin{pmatrix} \sum_{i=1}^8 \phi_i^n \int_{\Omega_e} \psi_i \psi_1 + \int_{\Gamma_{N,e}} \mathbf{n} \cdot \nabla \phi^{n+1} \psi_1 \\ \vdots \\ \sum_{i=1}^8 \phi_i^n \int_{\Omega_e} \psi_i \psi_8 + \int_{\Gamma_{N,e}} \mathbf{n} \cdot \nabla \phi^{n+1} \psi_8 \end{pmatrix}.$$

3.5. Condiciones de contorno

Los tipos de condiciones de contorno que existen son: condición de Dirichlet, de Neumann y de Robin; siendo Γ_D , Γ_N y Γ_B las porciones de Γ en donde se imponen respectivamente. Todas ellas son casos particulares de la condición 3.2. Para completar el esquema numérico se necesita conocer la forma de imponer estas condiciones en el método de pasos de fraccionados.

En la primer etapa del método, la de advección, la única condición posible de implementar es la de Dirichlet en la zona de flujo entrante $\Gamma^- = \{\mathbf{x} \in \Gamma : \mathbf{u}(\mathbf{x}) \cdot \mathbf{n} < 0\}$.

Llamando ϕ_{Adv} y ϕ_{Dif} a los valores de la variable ϕ durante los pasos de advección y difusión respectivamente; en el primer paso se propone exigir una condición de Dirichlet para ϕ_{Adv} de la forma:

$$\phi_{Adv}^{n+1}(\mathbf{x}_i) = \phi^n(\mathbf{x}_i) \quad \forall \mathbf{x}_i \in \Gamma^-, \quad (3.30)$$

independientemente del tipo de condición que se imponga en Γ^- .

Al final del paso de transporte, la condición de borde en ϕ^{n+1} sobre Γ quedará fijada por ϕ_{Dif} según de que condición se trate.

3.5.1. Condición de Dirichlet

Esta condición es equivalente a la ecuación 3.2 haciendo $\alpha = 0$ y $\gamma = 0$, es decir:

$$\phi(\mathbf{x}, t) = \phi_D(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_D. \quad (3.31)$$

En este caso se exige una condición de Dirichlet sobre ϕ_{Dif} de la forma:

$$\phi_{Dif}^{n+1}(\mathbf{x}_i) = \phi_D(\mathbf{x}_i, t_{n+1}) \quad \forall \mathbf{x}_i \in \Gamma_D. \quad (3.32)$$

Así, al final del paso de difusión, el valor de la variable ϕ^{n+1} adoptará el valor de $\phi_D(\mathbf{x}, t_{n+1})$ sobre Γ_D .

3.5.2. Condición de Neumann

Esta condición es equivalente a la ecuación 3.2 haciendo $\beta = 0$ y $\alpha = 1$. La misma se puede escribir como:

$$\mathbf{n} \cdot \nabla \phi(\mathbf{x}, t) = \gamma(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_N. \quad (3.33)$$

En este caso se exige una condición de Neumann sobre ϕ_{Dif} de la forma:

$$\mathbf{n} \cdot \nabla \phi_{Dif}(\mathbf{x}_i) = \gamma(\mathbf{x}_i, t_{n+1}) \quad \forall \mathbf{x}_i \in \Gamma_N. \quad (3.34)$$

Así, al final del paso de difusión, el valor de la variable $\mathbf{n} \cdot \nabla \phi^{n+1}$ adoptará el valor de $\gamma(\mathbf{x}, t_{n+1})$ sobre Γ_N .

3.5.3. Condición de Robin

En este trabajo se analiza el caso particular con $\alpha = -D$, $\beta = \mathbf{n} \cdot \mathbf{u}$, $\gamma = 0$ y $\phi_D = 0$ en la ecuación 3.2. De esta forma, la condición se expresa como:

$$\mathbf{n} \cdot \mathbf{u} \phi - D \mathbf{n} \cdot \nabla \phi = 0 \quad \forall \mathbf{x} \in \Gamma_R. \quad (3.35)$$

Esta condición implica un flujo nulo de ϕ a través de la superficie Γ_R en el caso de un fluido incompresible.

En este caso se propone imponer una condición de Neumann sobre ϕ_{Dif} de la forma:

$$\mathbf{n} \cdot \nabla \phi_{Dif}^{n+1}(\mathbf{x}) = \frac{1}{D \Delta t} \int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t) \phi_{Adv}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_R. \quad (3.36)$$

Lo que incentivó a aplicar este esquema es el hecho de que en la frontera Γ_R el flujo neto del escalar debe ser nulo. Para probar esto, la cantidad neta del escalar Q_{Adv} que

entra a través de un punto $\mathbf{x} \in \Gamma_R$ en el paso de advección durante $\Delta t = [t_n \ t_{n+1})$ puede calcularse como:

$$Q_A = - \int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t) \phi_{Adv}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_R. \quad (3.37)$$

Por otro lado el flujo Q_{Dif} que entra a través de un punto $\mathbf{x} \in \Gamma_R$ durante el paso de difusión puede calcularse como:

$$Q_{Dif} = \int_{t_n}^{t_{n+1}} D \mathbf{n} \cdot \nabla \phi_{Dif}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_R. \quad (3.38)$$

Para que el flujo neto se anule, planteamos que $Q_{Dif} = -Q_{Adv}$, de esa manera se tiene:

$$\int_{t_n}^{t_{n+1}} D \mathbf{n} \cdot \nabla \phi_{Dif}(\mathbf{x}, t) = \int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t) \phi_{Adv}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_R. \quad (3.39)$$

Exigiendo una condición de Neumann para el escalar durante el paso de difusión se tiene la siguiente igualdad:

$$\int_{t_n}^{t_{n+1}} D \mathbf{n} \cdot \nabla \phi_{Dif}(\mathbf{x}, t) = D \mathbf{n} \cdot \nabla \phi_{Dif}^{n+1}(\mathbf{x}) \Delta t \quad \forall \mathbf{x} \in \Gamma_R. \quad (3.40)$$

Finalmente se puede reescribir la condición 3.39 como:

$$D \mathbf{n} \cdot \nabla \phi_{Dif}^{n+1}(\mathbf{x}) \Delta t = \int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t) \phi_{Adv}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_R, \quad (3.41)$$

que se traduce a exigir la condición 3.36.

Considerando un problema en cuyo borde se impone la condición de Robin y recordando que $\phi^n(\mathbf{x})$ representa al campo del escalar previo al paso de advección, $\phi^*(\mathbf{x})$ posterior al mismo y $\phi^{n+1}(\mathbf{x})$ luego del paso de difusión. Del esquema numérico de difusión 3.19 se tiene:

$$\int_{\Omega} \phi_{Dif}^{n+1} = \int_{\Omega} \phi_{Dif}^n + D \Delta t \int_{\Omega} \nabla \cdot \nabla \phi_{Dif}^{n+1}, \quad (3.42)$$

como $\phi_{Dif}^n = \phi^*$ y $\phi_{Dif}^{n+1} = \phi^{n+1}$ se llega a:

$$\int_{\Omega} \phi^{n+1} = \int_{\Omega} \phi^* + D \Delta t \int_{\Omega} \nabla \cdot \nabla \phi_{Dif}^{n+1}. \quad (3.43)$$

Luego, aplicando el Teorema de la Divergencia se tiene:

$$\int_{\Omega} \phi^{n+1} = \int_{\Omega} \phi^* + D \Delta t \int_{\Gamma} \mathbf{n} \cdot \nabla \phi_D^{n+1}. \quad (3.44)$$

Suponiendo que en el paso de advección la cantidad del escalar en el dominio únicamente se modifica por el flujo neto de escalar a través de la superficie, se puede escribir que:

$$\int_{\Omega} \phi^* = \int_{\Omega} \phi^n - \int_{t_n}^{t_{n+1}} \int_{\Gamma} \mathbf{n} \cdot \mathbf{u} \phi_A. \quad (3.45)$$

Reemplazando 3.45 en 3.44 se llega a:

$$\int_{\Omega} \phi^{n+1} = \int_{\Omega} \phi^n - \int_{\Gamma} \int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u} \phi_A + D\Delta t \int_{\Gamma} \mathbf{n} \cdot \nabla \phi_D^{n+1}. \quad (3.46)$$

Reagrupando, se tiene:

$$\int_{\Omega} \phi^{n+1} = \int_{\Omega} \phi^n + \int_{\Gamma} \left[D\Delta t \mathbf{n} \cdot \nabla \phi_D^{n+1} - \int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u} \phi_A \right]. \quad (3.47)$$

Finalmente cumpliendo con la condición 3.36 se llega a:

$$\int_{\Omega} \phi^{n+1} = \int_{\Omega} \phi^n, \quad (3.48)$$

que es el resultado esperado al aplicar la condición de Robin.

Para calcular el flujo de la cantidad advectada se pueden utilizar distintos tipos de métodos de integración numérica. Por ejemplo:

$$\int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t) \phi_{Adv}(\mathbf{x}, t) \simeq \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t_{n+1}) \phi_A^{n+1}(\mathbf{x}) \Delta t \quad \forall \mathbf{x} \in \Gamma_R. \quad (3.49)$$

O una mejor aproximación es:

$$\int_{t_n}^{t_{n+1}} \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t) \phi_{Adv}(\mathbf{x}, t) \simeq \frac{\mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t_{n+1}) \phi_A^{n+1} + \mathbf{n} \cdot \mathbf{u}(\mathbf{x}, t_n) \phi_A^n(\mathbf{x})}{2} \Delta t \quad \forall \mathbf{x} \in \Gamma_R, \quad (3.50)$$

que corresponde a la fórmula del trapecio.

3.6. Resultados en 1D

Con el objetivo de evaluar el esquema numérico de transporte de escalar, se elaboró un código 1D programado en OCTAVE.

Se consideró un perfil de velocidad uniforme $V > 0$ en una malla uniforme de 70 elementos y 71 nodos, donde, $x = 0$ es el borde de flujo entrante y $x = 1$ el borde de flujo saliente. Para el paso de difusión se utilizaron funciones de forma lineales.

Se compararon los resultados del estado estacionario entre el código y las soluciones exactas. En cada problema se simulaban distintos números de Peclet (Pe), que se define

como:

$$Pe = \frac{VL}{D}, \quad (3.51)$$

y es la relación entre la transferencia de escalar por advección y por difusión. En todos los casos se consideró $L = 1$, $D = 1$ y se varió el Pe a través del valor de V .

El primer problema que se resolvió fue aquel con condición de Dirichlet en ambos extremos, es decir:

$$\phi(0) = 1, \quad \phi(1) = 0. \quad (3.52)$$

La solución exacta a este problema es:

$$\phi(x) = \frac{e^{Vx/D} - e^{V/D}}{1 - e^{V/D}}. \quad (3.53)$$

Los resultados del código y la solución exacta para este caso se reportan en la figura 3.3. En la misma se observa una buena concordancia entre el resultado numérico y la solución exacta.

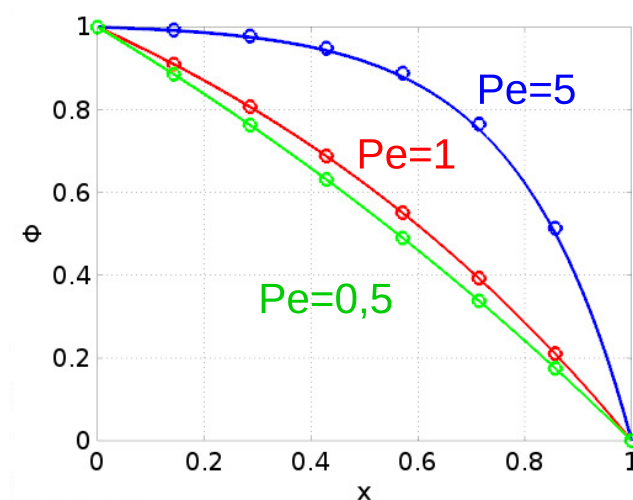


Figura 3.3: Soluciones al problema de transporte 1D con condición de Dirichlet en ambos extremos. Solución exacta con “o” y numérica en línea continua.

En el siguiente caso se impuso una combinación de condiciones de Dirichlet y Neumann en los extremos de flujo entrante y flujo saliente, es decir:

$$\begin{aligned} a) \quad & \phi(0) = 1, \quad \frac{\partial \phi}{\partial x}(1) = 1, \\ b) \quad & \frac{\partial \phi}{\partial x}(0) = 1, \quad \phi(1) = 3, \end{aligned} \quad (3.54)$$

donde las soluciones exactas para cada caso son:

$$\begin{aligned} a) \quad \phi(x) &= \frac{e^{Vx/D} + V/De^{V/D} - 1}{V/De^{V/D}}, \\ b) \quad \phi(x) &= \frac{e^{Vx/D} - e^{V/D} + 3V/D}{V/D}. \end{aligned} \quad (3.55)$$

Los resultados del código y la solución exacta para este caso se muestran en la figura 3.4. Se observa una buena concordancia entre el resultado numérico y el analítico.

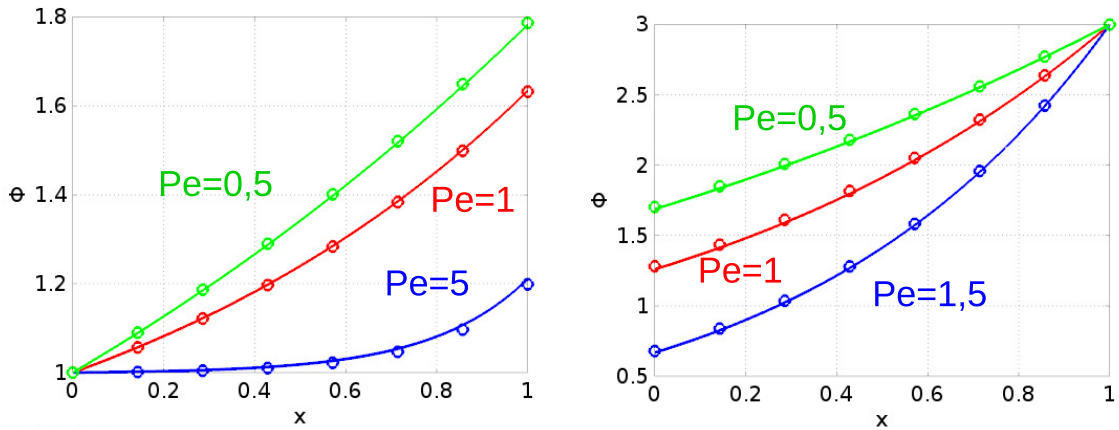


Figura 3.4: Soluciones al problema de transporte 1D. A la izquierda, problema con condición de Dirichlet en $x=0$ y Neumann en $x=1$. A la derecha, problema con condición de Neumann en $x=0$ y Dirichlet en $x=1$. Solución exacta con “o” y numérica en línea continua.

En la siguiente simulación se impuso una combinación de condiciones de Dirichlet y Robin en los extremos de flujo entrante y flujo saliente, es decir:

$$\begin{aligned} a) \quad \phi(0) &= 1, \quad D \frac{\partial \phi}{\partial x}(1) - V\phi(1) = 0, \\ b) \quad D \frac{\partial \phi}{\partial x}(0) - V\phi(0) &= 0, \quad \phi(1) = 1, \end{aligned} \quad (3.56)$$

donde las soluciones exactas para cada caso son:

$$\begin{aligned} a) \quad \phi(x) &= e^{\frac{Vx}{D}}, \\ b) \quad \phi(x) &= e^{\frac{V(x-1)}{D}}. \end{aligned} \quad (3.57)$$

Los resultados del código y la solución exacta para este caso se aprecian en la figura 3.5. Se observa una buena concordancia entre el resultado numérico y la solución exacta.

Finalmente se planteó un problema con condición de Robin en ambos extremos, es decir:

$$D \frac{\partial \phi}{\partial x} - V\phi = 0 \text{ en } x = 0 \text{ y } x = 1. \quad (3.58)$$

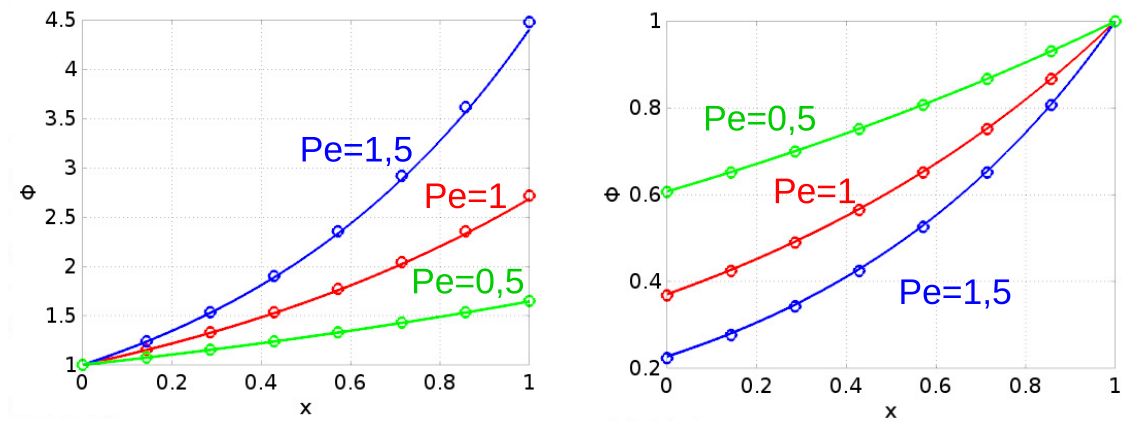


Figura 3.5: Soluciones al problema de transporte 1D. A la izquierda, con condición de Dirichlet en $x=0$ y Robin en $x=1$. A la derecha, problema con condición de Robin en $x=0$ y Dirichlet en $x=1$. Solución exacta con “o” y numérica en línea continua.

La solución exacta a este problema es:

$$\phi(x) = \frac{Vm_0 e^{\frac{Vx}{D}}}{De^{\frac{V}{D}} - 1} \quad \text{con} \quad m_0 = \int_{\Omega} \phi(t=0), \quad (3.59)$$

para obtenerla se planteó que la cantidad del escalar se conserva para todo tiempo, esto se debe a que en este caso el flujo del mismo por la frontera es nulo. Los resultados del código y la solución exacta para este caso se ven en la figura 3.6. En la misma se observa una buena concordancia entre el resultado numérico y el analítico.

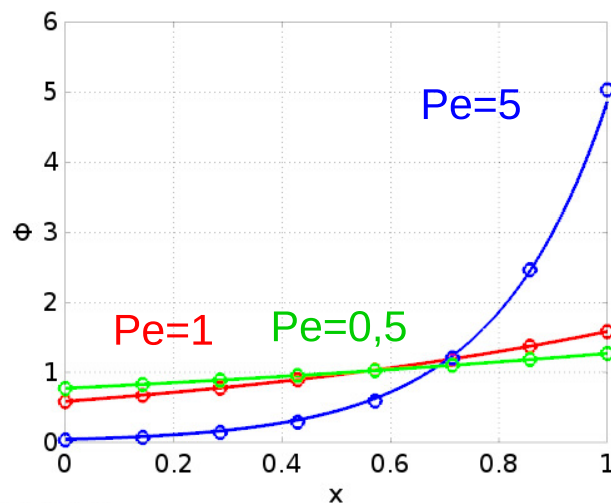


Figura 3.6: Soluciones al problema de transporte 1D con condiciones de Robin en ambos extremos. Solución exacta con “o” y numérica en línea continua.

Con esto se concluye que el esquema numérico de transporte de escalares planteado tiene resultados satisfactorios en el caso 1D.

Capítulo 4

Implementación eficiente del transporte

“[...]Ahora sí, a mi juego me llamaron. Vengo bien, sigo así, y lo voy a alcanzar! [...] Cada golpe de la zapatilla contra el asfalto rumbo a Rawson era una sílaba que le daba fuerza, que lo impulsaba a ganarle al viento.[...]”

— Crónica del Maratón Tres Ciudades Patagónicas 2009, Nicolás Ternavasio.

En este capítulo se explica la forma en que se acopló el transporte de escalares dentro del código. Se describe la implementación del método de advección semi-lagrangiana a través de un algoritmo de búsqueda con árboles octales con procesamiento en paralelo. Finalmente se comparan los resultados del código completo con otros trabajos y se evalúa la eficiencia en tiempo de cálculo.

4.1. Esquema numérico

El problema general de transporte de un escalar en un fluido incompresible en movimiento puede ser descrito con el siguiente conjunto de ecuaciones:

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} - \mu \Delta \mathbf{u} + \nabla p = \mathbf{F}, \\ \nabla \cdot \mathbf{u} = 0, \\ \frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - \nabla(D \nabla \phi) = 0, \end{cases} \quad (4.1)$$

con las condiciones de contorno 2.2 y 3.2, y las condiciones iniciales 2.3 y 3.3 aplicadas simultáneamente en esta formulación general. Se considera que las constantes ρ , μ y la fuente \mathbf{F} pueden depender de la variable escalar ϕ .

Para abordar este problema, implementando las técnicas antes descritas, se siguió un esquema “en etapas”. Éste consiste básicamente en resolver un campo a la vez. Utilizando como datos \mathbf{u}^n , p^n y ϕ^n se calcula \mathbf{u}^{n+1} y p^{n+1} resolviendo el problema de fluidos:

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} - \mu \Delta \mathbf{u} + \nabla p = \mathbf{F}, \\ \nabla \cdot \mathbf{u} = 0. \end{cases} \quad (4.2)$$

Luego utilizando $\mathbf{u} = \mathbf{u}^{n+1}$ y ϕ^n como datos, se calcula ϕ^{n+1} resolviendo el problema de transporte:

$$\begin{cases} \frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - \nabla(D\nabla\phi) = 0. \end{cases} \quad (4.3)$$

El procedimiento se repite sucesivamente hasta alcanzar el tiempo de simulación deseado.

4.2. Advección semi-lagrangiana eficiente

Como se explicó en el capítulo 3 la implementación computacional del método de advección semi-lagrangiana requiere la búsqueda de los elementos de la malla a los cuales pertenecen los puntos de partida \mathbf{x}_i^* . Una vez hallado el elemento se interpola el valor del escalar en ese punto utilizando algunos de los valores nodales del elemento.

En este trabajo se implementó el método de búsqueda con árboles octales. Este se basa en crear una red auxiliar que abarca todo el dominio. Esta red se densifica en función de la malla original como se ve en la figura 4.1 para una caso 2D en el que se utilizan cuadrados. El método se implementa de manera similar para el caso 3D pero con cubos.

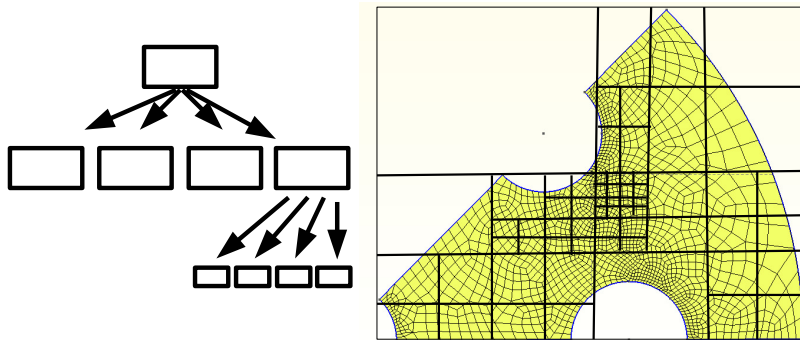


Figura 4.1: Ejemplo de una red auxiliar estructurada como un árbol octal.

El cubo más grande de la red que contiene a todos los elementos de la malla se lo denomina “raíz”, los 8 octantes en que se divide la raíz se los llaman “ramas”, estas se pueden seguir subdividiendo sucesivamente hasta alcanzar un tamaño apto para una búsqueda eficiente que sea un compromiso entre velocidad de búsqueda y uso de la

memoria, esto puede verse en [14]. Las ramas que ya no se dividen se llaman “hojas”, estas almacenan una lista de los elementos de la malla que intersectan con la misma.

Cada rama almacena información sobre su tamaño y su ubicación en el espacio, de esta forma es posible ir calculando, a partir de la raíz, a que hojas pertenecen los puntos de partida \mathbf{x}_i^* . Una vez identificada la misma se recorren cada uno de los elementos que ésta contenga y se busca uno por uno a cuál de ellos pertenece el punto.

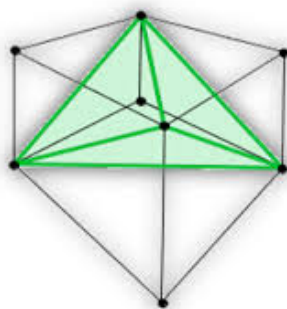


Figura 4.2: Descomposición de un hexaedro en cinco tetraedros.

Para saber si un punto pertenece a un elemento se divide al hexaedro en cinco tetraedros tal como se aprecia en la figura 4.2 y se recorre cada uno de estos tetraedros evaluando si el punto pertenece a alguno de ellos. Considerando un tetraedro de vértices $\mathbf{a} = (x_a, y_a, z_a)$, $\mathbf{b} = (x_b, y_b, z_b)$, $\mathbf{c} = (x_c, y_c, z_c)$ y $\mathbf{d} = (x_d, y_d, z_d)$ que se ordenan de una forma tal que:

$$vol(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) = \frac{1}{6} \begin{vmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{vmatrix} > 0,$$

donde $vol(\cdot, \cdot, \cdot, \cdot)$ es la función que da el volumen del tetraedro conformado con los vértices en su argumento, así, un punto $\mathbf{x}^* = (x, y, z)$ pertenece a ese tetraedro si se cumple simultáneamente que:

$$vol(\mathbf{x}^*, \mathbf{b}, \mathbf{c}, \mathbf{d}) = \frac{1}{6} \begin{vmatrix} x & y & z & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{vmatrix} \geq 0, \quad vol(\mathbf{a}, \mathbf{x}^*, \mathbf{c}, \mathbf{d}) = \frac{1}{6} \begin{vmatrix} x_a & y_a & z_a & 1 \\ x & y & z & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{vmatrix} \geq 0,$$

$$vol(\mathbf{a}, \mathbf{b}, \mathbf{x}^*, \mathbf{d}) = \frac{1}{6} \begin{vmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x & y & z & 1 \\ x_d & y_d & z_d & 1 \end{vmatrix} \geq 0 \text{ y } vol(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{x}^*) = \frac{1}{6} \begin{vmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x & y & z & 1 \end{vmatrix} \geq 0.$$

Una vez identificado el tetraedro al cual pertenece el punto \mathbf{x}^* se interpola el valor

del escalar con los valores nodales ϕ_a, ϕ_b, ϕ_c y ϕ_d de este último con la siguiente fórmula:

$$\phi(\mathbf{x}^*) = \frac{\text{vol}(\mathbf{x}^*, \mathbf{b}, \mathbf{c}, \mathbf{d})\phi_a + \text{vol}(\mathbf{a}, \mathbf{x}^*, \mathbf{c}, \mathbf{d})\phi_b + \text{vol}(\mathbf{a}, \mathbf{b}, \mathbf{x}^*, \mathbf{d})\phi_c + \text{vol}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{x}^*)\phi_d}{\text{vol}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})}. \quad (4.4)$$

4.3. Procesamiento en paralelo de la advección

Con el fin de poder ampliar los alcances de este código a problemas de gran tamaño que requieran velocidad de cómputo y gran cantidad de memoria, se realizó una implementación del método de advección semi-lagrangiana con procesamiento en paralelo. Para este caso se aprovechó la misma asignación de nodos y elementos a cada proceso que se hizo en la resolución de las ecuaciones de N-S.

Para implementar el algoritmo en forma paralela, cada proceso construye su propia red auxiliar con la estructura de árbol octal y almacena sus elementos propios. En cada paso temporal los procesos calculan su propio grupo de puntos de partida \mathbf{x}_i^* utilizando las coordenadas de sus nodos propios \mathbf{x}_i y el valor de la velocidad en el nodo \mathbf{u}_i . Luego identifican que puntos pertenecen a sus elementos propios e interpolan el valor del escalar para asignárselos a la solución ϕ_i^* .

En general, puede existir un grupo de puntos de partida que no pertenezcan a elementos propios, por ejemplo si se utilizan pasos de tiempo demasiado grandes, en ese caso se debe realizar una segunda etapa de búsqueda. Cada proceso carga en un vector distribuido, al cuál pueden acceder todos los procesos, el valor de los puntos de partida propios que no encontró en la primer etapa. Luego cada proceso identifica si alguno de sus elementos propios contiene a algún punto de esta lista, si logra identificar el elemento, interpola el valor del escalar y lo suma a ϕ_i^* que en un principio se encuentra inicializado en cero. Puede haber puntos de esta lista que sean encontrados por dos procesos distintos, en base a esto se divide ϕ_i^* por el número de procesos que encontraron elementos propios con ese punto dentro.

Por otra parte los puntos pueden estar fuera del dominio ya que puede existir una región de flujo entrante, en ese caso se debe tomar una decisión en base a las condiciones de contorno impuestas y a lo discutido en el capítulo 3.

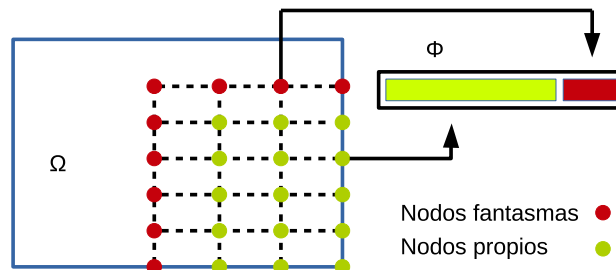


Figura 4.3: Ejemplo de asignación de nodos locales y nodos fantasmas a un proceso .

Para poder interpolar el valor del escalar en elementos propios que contengan nodos externos, también llamados “nodos fantasmas”, se utilizaron objetos de PETSc llamados “VecGhost” en Inglés. Con los mismos se puede acceder a los valores de ϕ_i correspondientes a los nodos propios y además a los valores en los nodos fantasmas en la cola del vector como se aprecia en la figura 4.3.

4.4. Validación

Para validar el funcionamiento del código general se resolvió un problema 2D, este consiste en una cavidad cuadrada en las coordenadas $x, z \in [0, 1]$ acoplada térmicamente. Los resultados obtenidos fueron comparados con el trabajo [15].

Para este problema las ecuaciones 2.1 contienen un término de fuerza boyante $\mathbf{F} = \rho \mathbf{g} \beta (T - T_0)$, donde T es el escalar a transportar en este caso la temperatura del fluido, ρ es la densidad del fluido a temperatura T_0 , β es coeficiente que relaciona cambios de densidad frente a cambios de temperatura y \mathbf{g} es la aceleración de la gravedad.

Las condiciones de contorno para este problema son:

$$\left\{ \begin{array}{l} \mathbf{u} = 0 \text{ en } x = 0, x = 1, z = 0, z = 1, \\ T = T_1 \text{ en } x = 0, \\ T = T_0 \text{ en } x = 1, \\ \frac{\partial T}{\partial \mathbf{n}} = 0 \text{ en } z = 0, z = 1, \end{array} \right. \quad (4.5)$$

Las ecuaciones en su forma estacionaria y adimensionalizada se escriben:

$$\left\{ \begin{array}{l} \mathbf{u}^+ \cdot \nabla_{\mathbf{x}^+} \mathbf{u}^+ - \text{Pr} \Delta_{\mathbf{x}^+} \mathbf{u}^+ + \nabla_{\mathbf{x}^+} p^+ = \text{Pr} \text{ Ra} T^+ \hat{\mathbf{z}}, \\ \nabla_{\mathbf{x}^+} \cdot \mathbf{u}^+ = 0, \\ \mathbf{u}^+ \cdot \nabla_{\mathbf{x}^+} T^+ - \Delta_{\mathbf{x}^+} T^+ = 0, \end{array} \right. \quad (4.6)$$

donde:

$$\mathbf{u}^+ = \frac{\mathbf{u}L}{D}, \quad T^+ = \frac{T - T_0}{T_1 - T_0}, \quad p^+ = \frac{pL^2}{\rho D^2}, \quad \mathbf{x}^+ = \frac{\mathbf{x}}{L}, \quad (4.7)$$

Pr y Ra son los números de Prandtl y Rayleigh definidos como:

$$\text{Pr} = \frac{\nu}{D}, \quad \text{Ra} = \frac{g\beta L^3 (T_1 - T_0)}{D\nu}, \quad (4.8)$$

y $L = 1$ es la longitud característica del problema.

Se realizaron dos simulaciones, una con $\text{Ra}=1\text{e}3$ y otra con $\text{Ra}=1\text{e}4$, ambas con $\text{Pr}=0,71$ al igual que en [15]. Para llegar a la solución estacionaria se calcularon 1000

pasos temporales con $\Delta t = 0,002$ para $Ra=1e3$ y $\Delta t = 0,0005$ para $Ra=1e4$. Por otro lado la discretización espacial se llevó a cabo con una malla de elementos cuboidales rectos como la de la figura 2.4 de $100 \times 100 \times 1$ elementos en las direcciones x , z e y . En la dirección y se fijaron las siguientes condiciones de borde:

$$\begin{cases} u_y = 0, & \tau_x = \tau_z = 0 \text{ en } y = 0, y = 1, \\ \frac{\partial T}{\partial \mathbf{n}} = 0 \text{ en } y = 0, y = 1. \end{cases} \quad (4.9)$$

En la figura 4.4 se representan los perfiles de velocidad u_x^+ en la línea $x^+ = 0,5$ para $Ra=1e3$ y para $Ra=1e4$ obtenidos con el código y contrastados con [15]. Se observa que los perfiles de velocidad calculados con el código son similares a los de [15] para ambos valores de Ra .

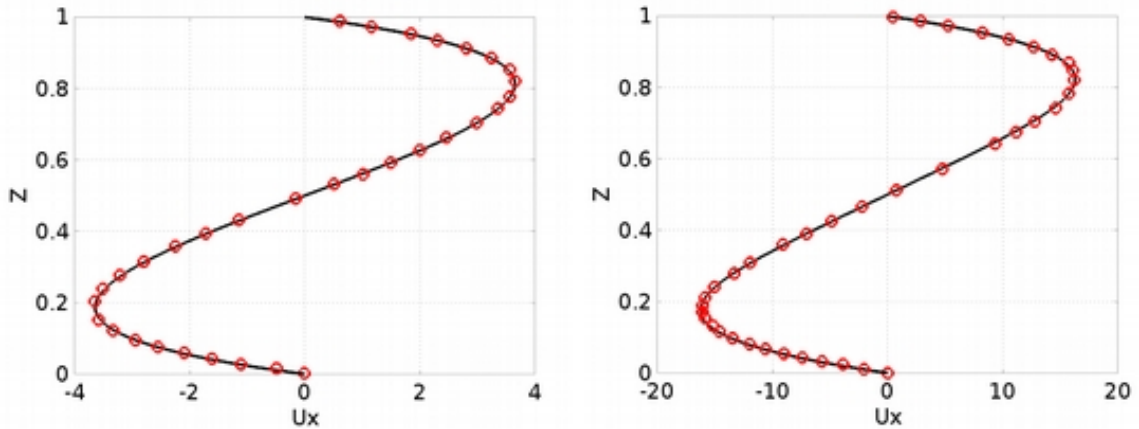


Figura 4.4: Perfiles de velocidad u_x^+ sobre la línea $x^+ = 0,5$. A la izquierda, con $Ra=1e3$. A la derecha, con $Ra=1e4$. Con “o” solución de [15], y en línea continua solución con el código.

En las figuras 4.5, 4.6, 4.7 y 4.8 se representa la distribución de la temperatura obtenida en la simulación y contrastadas con [15] para $Ra = 1e3$ y $Ra = 1e4$. Se observa que los perfiles de temperatura calculados con el código tienen un aspecto similar a los de [15] para ambos valores de Ra .

Por otra parte en las tablas 4.1 y 4.2 se comparan los cálculos de la velocidad máxima en dirección x y z ($\bar{u}_{x,max}^*(x = 0,5)$ y $\bar{u}_{z,max}^*(z = 0,5)$), la ubicación de las mismas ($\bar{z}_{max}(x = 0,5)$ y $\bar{x}_{max}(z = 0,5)$) y el número de Nusselt global (Nu) con los códigos “Fluent” y “OpenFOAM” (resultados reportados en [15]). Se aprecia que los resultados con el código son similares a los reportados en [15] y que el transporte de escalares es correctamente calculado para este caso.

El Número de Nusselt global (Nu) se define como:

$$Nu = \frac{hL}{k}, \quad (4.10)$$

donde h es el coeficiente de transferencia térmico entre las placas en $x = 0$ y $x = 1$, y

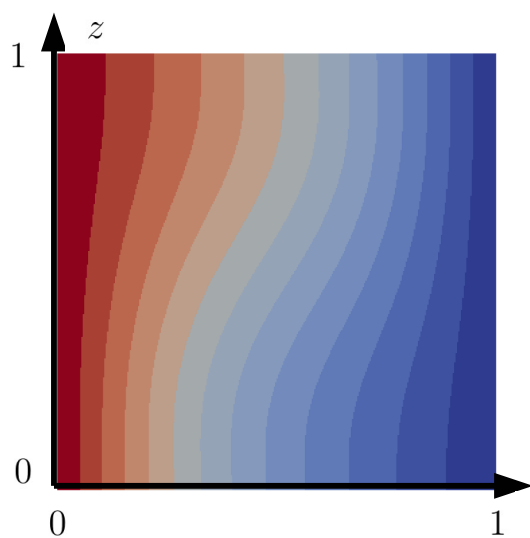


Figura 4.5: Distribución de la temperatura para $Ra = 1e3$ calculada con el código.

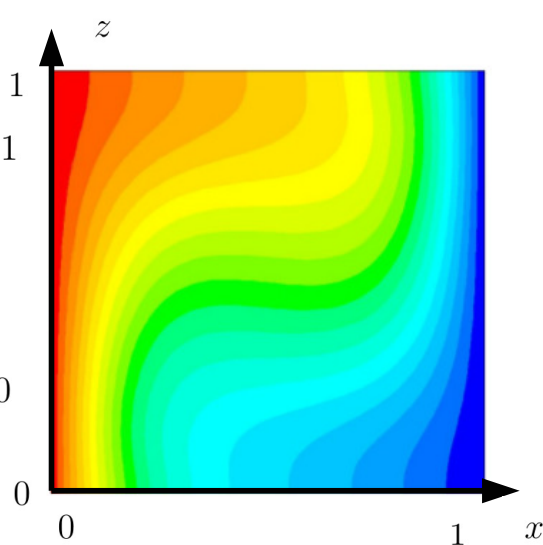


Figura 4.6: Distribución de la temperatura para $Ra = 1e3$ calculada por [15].

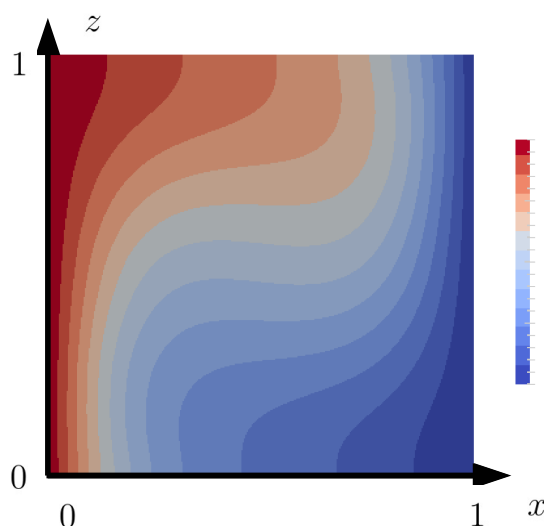


Figura 4.7: Distribución de la temperatura para $Ra = 1e4$ calculada con el código.

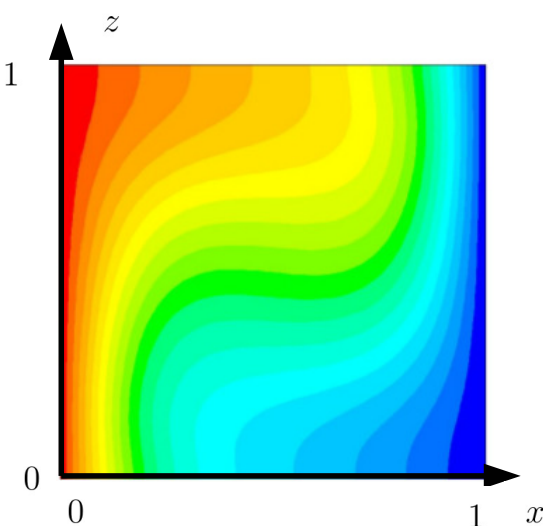


Figura 4.8: Distribución de la temperatura para $Ra = 1e4$ calculada por [15].

Ra=1e3			
	Código	Fluent	OpenFOAM
Nu	1,132	1,113	1,109
$\bar{u}_{x,max}^*(x = 0, 5)$	3,649	3,643	3,640
$\bar{z}_{max}(x = 0, 5)$	0,810	0,817	0,812
$\bar{u}_{z,max}^*(z = 0, 5)$	3,708	3,690	3,700
$\bar{x}_{max}(z = 0, 5)$	0,180	0,182	0,177

Tabla 4.1: Constantes para un $Ra=1e3$ calculadas con el código y resultados de los códigos “Fluent” y “OpenFOAM” de [15].

Ra=1e4			
	Código	Fluent	OpenFOAM
Nu	2,240	2,246	2,222
$u_{x,max}^*(x = 0, 5)$	16,275	16,139	16,281
$z_{max}(x = 0, 5)$	0,830	0,817	0,822
$u_{z,max}^*(z = 0, 5)$	19,732	19,619	19,547
$x_{max}(z = 0, 5)$	0,120	0,119	0,123

Tabla 4.2: Constantes para un Ra=1e4 calculadas con el código y resultados de los códigos “Fluent” y “OpenFOAM” de [15].

k es el coeficiente de conducción térmica. Para calcularlo se utiliza la relación:

$$\int_{\Gamma^*} q'' = h(T_1 - T_0) \Rightarrow h = \int_{\Gamma^*} \frac{q''}{T_1 - T_0}, \quad (4.11)$$

donde q'' es el calor transferido entre placas, Γ^* es la superficie de la placa $x = 0$. Por otro lado utilizando la Ley de Fourier $q'' = k\nabla T \cdot \mathbf{n}$ el Nu puede calcularse como:

$$Nu = \int_{\Gamma^*} \frac{q''L}{k(T_1 - T_0)} = \int_{\Gamma^*} \frac{\nabla T \cdot \mathbf{n}L}{(T_1 - T_0)}. \quad (4.12)$$

4.5. Evaluación del tiempo de cálculo

Para evaluar el tiempo de cálculo del código, y ver como es afectado cuando se computa en forma paralela, se resolvió un problema con un dominio cúbico de $50 \times 50 \times 50$ elementos en las direcciones x, y y z . Se fijó el lado del cubo en $L = 1$, $Pr = 0,71$, $Ra = 1e3$ y el término de fuerza boyante de la forma $\mathbf{F} = \rho \mathbf{g} \beta (T - T_0)$. Por otro lado las condiciones de contorno fueron:

$$\left\{ \begin{array}{l} \mathbf{u} = 0 \text{ en } x = 0, x = 1, y = 0, y = 1, z = 0, z = 1, \\ T = T_1 \text{ en } x = 0, \\ T = T_0 \text{ en } x = 1, \\ \frac{\partial T}{\partial \mathbf{n}} = 0 \text{ en } z = 0, z = 1, y = 0, y = 1. \end{array} \right. \quad (4.13)$$

Las simulaciones fueron realizadas en una computadora con un procesador INTEL I7 con 6 núcleos. En cada una de las simulaciones se realizaron un total de 5 pasos temporales y se promediaron en estos 5 pasos el tiempo en que tarda el código en resolver las ecuaciones de N-S, el paso de advección y el paso de difusión.

En la figura 4.9 se muestran los tiempos de cálculo para distinto número de procesos en el caso en que $\Delta t = 0,001$. En cada una de estas simulaciones, como Δt es pequeño, todos los puntos x_i^* pertenecen a elementos propios y no se necesita de una segunda etapa de búsqueda. Como puede verse, la implementación del algoritmo de advección

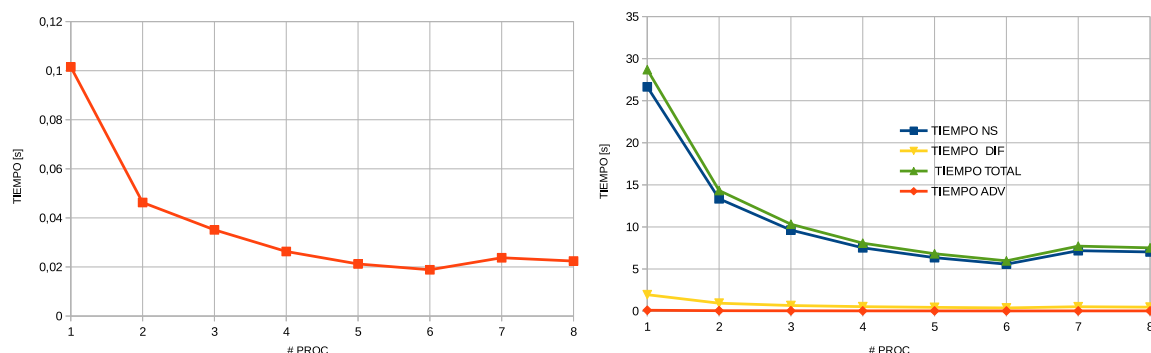


Figura 4.9: Tiempo de cálculo del código en función del número de procesos con $\Delta t=0,001$. A la izquierda, para el paso de advección. A la derecha, para las ecuaciones de N-S, el paso de advección, el de difusión y la suma de todos ellos.

semi-lagrangiana en este caso es paralelizable ya que el tiempo de cómputo disminuye con el número de procesos.

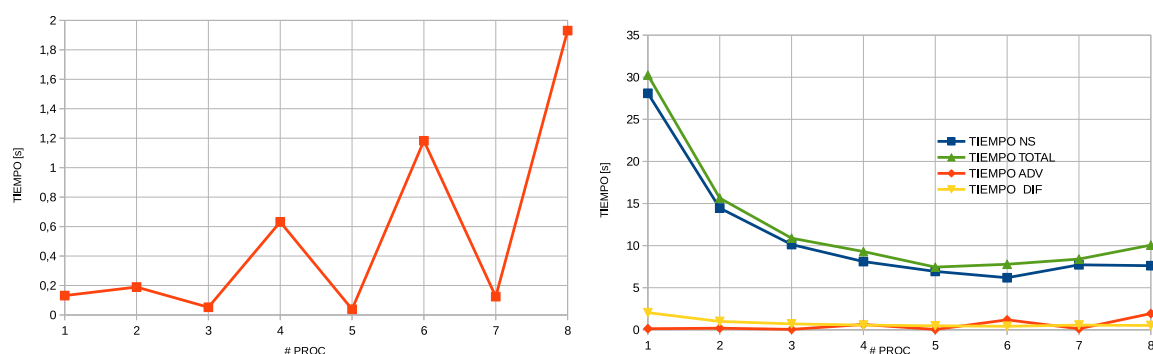


Figura 4.10: Tiempo de cálculo del código en función del número de procesos con $\Delta t=0,1$. A la izquierda, para el paso de advección. A la derecha, para las ecuaciones de N-S, el paso de advección, el de difusión y la suma de todos ellos.

En la figura 4.10 se muestran los tiempos de cálculo para distinto número de procesos en el caso en que $\Delta t=0,1$. En estas circunstancias gran parte de los puntos de partida x_i^* , al ser Δt grande, pertenecen a elementos externos y por lo tanto es necesario realizar una segunda etapa de búsqueda. Como se aprecia, el algoritmo de advección semi-lagrangiana presenta oscilaciones en el tiempo de cálculo según el número de procesos que se utilicen. Esto se debe a que el perfil de velocidades y la asignación de nodos y elementos a procesos hace que en los casos en que se utiliza un número par de procesos exista una gran cantidad de puntos de partida que pertenecen a elementos externos, no así cuando se utiliza un número impar.

Con estas simulaciones puede verse que es importante la forma en que se asignan los nodos de la malla a los procesos ya que esto puede repercutir significativamente en el tiempo de cómputo.

Capítulo 5

Conclusiones

“[...] Al Pampeano lo pasó en el 39 como una luz, como una anécdota, como una víctima irreversible de la Maratón.[...] La banda de llegada le acarició el corazón, los últimos pasos le devolvieron el alma. [...]”

— Crónica del Maratón Tres Ciudades Patagónicas 2009,
Nicolás Ternavasio.

5.1. Comentarios finales

En la primer etapa de este trabajo, se evaluó el esquema numérico utilizado por el código para resolver las ecuaciones de N-S. Esta implementación se validó a través de la resolución de un problema con distintos Números de Reynolds y se mostró que los resultados de estos cálculos eran concordantes con los de otro trabajo tomados como patrón de comparación.

En una segunda etapa, se elaboró un esquema de pasos fraccionados que divide el transporte de un escalar en dos pasos, uno de advección con el método de advección semi-lagrangiana y un paso de difusión con el método de elementos finitos. Por otro lado se propusieron una serie de métodos que permiten imponer condiciones de contorno en este esquema y se comprobó su aplicabilidad en problemas 1D.

El esquema numérico de transporte de escalar fue implementado con técnicas de procesamiento en paralelo y con el método de búsqueda de elementos con árboles octales. Se validó el código resolviendo un problema de transporte de temperatura con distintos Números de Rayleigh y se verificó que los resultados eran concordantes con los de otro trabajo utilizado como patrón de comparación. Finalmente se analizaron los tiempos de cálculo del código completo para distintos números de procesos y distintas situaciones. Se concluyó que el paso de advección incorporado no repercute significati-

vamente en el tiempo total de cómputo.

5.2. Trabajo futuro

El estado de desarrollo actual del código permite simular el transporte de escalares por fluidos, sin embargo, los tiempos de cálculo y la cantidad de memoria para problemas con geometrías complejas son altos. Para que el ensamblaje de las matrices, resolución de los sistemas de ecuaciones y el paso de advección sean efectivos en tiempo y memoria, la asignación de nodos a cada proceso debe ser realizada de forma tal de que estos se encuentren agrupados geoméricamente y exista la mínima cantidad de elementos propios para cada proceso. Para ello se deberá elaborar un algoritmo que realice una asignación eficiente de nodos a los procesos teniendo en cuenta lo anterior.

El algoritmo de advección actual realiza el cálculo de los puntos de partida a través de la fórmula 3.16 con $K = 1$, esto puede introducir un error apreciable en el cálculo de la solución si los pasos de tiempo que se utilizan son grandes. Para obtener menor error y continuar utilizando pasos de tiempo grandes, se debe implementar una estrategia de selección de puntos de partida \mathbf{x}_i^* que sean calculados con $K > 1$. Por ejemplo, si \mathbf{x}_i^* cae fuera del dominio con $K = 1$, se puede volver a calcular el mismo con $K = 6$ como se propone en [16] y ver si efectivamente cae fuera por tratarse de una zona de flujo entrante.

Para ampliar los alcances de este código a problemas generales, este deberá implementar un algoritmo capaz de imponer las condiciones de contorno discutidas en el capítulo 3. Aún así, en el estado actual de desarrollo, es posible imponer condiciones de tipo Dirichlet y Neumann las cuales sirven para modelar una basta cantidad de problemas de ingeniería como se ve en la figura 5.1. En este caso, se buscó simular el transporte de temperatura por el refrigerante, que pasa por un elemento combustible de un reactor nuclear, con el fin de poder calcular la potencia térmica que es capaz de transferir este sistema en distintas condiciones de flujo.

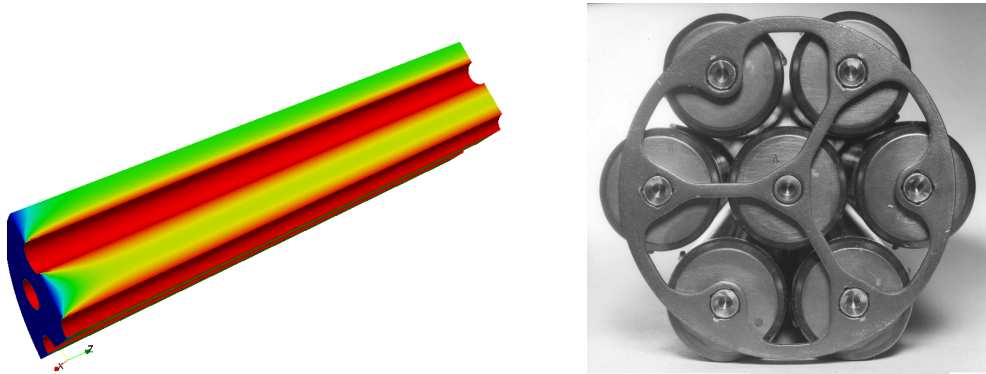


Figura 5.1: A la izquierda, simulación realizada con el código del transporte de temperatura en el refrigerante que atraviesa un elemento combustible de un reactor nuclear. La escala de colores representa la magnitud de la temperatura en el fluido. A la derecha, fotografía del elemento combustible real. Este cálculo se presenta solo a modo de ejemplo del tipo de problema que el código puede resolver.

Apéndice A

Declaración de actividades

A.1. Actividades de práctica profesional supervisada

Las actividades de práctica profesional supervisada incluyeron: el aprendizaje del método de los Elementos Finitos aplicado a las ecuaciones de N-S y a la ecuación de difusión, el aprendizaje del método de Advección Semi-Lagrangiana y el uso del procesamiento en paralelo.

A.2. Actividades de proyecto y diseño

Las actividades de proyecto y diseño se reportan en los capítulos del 1 al 5 de este trabajo. El alumno evaluó y validó el esquema numérico implementado en un código de Elementos Finitos para la resolución de las ecuaciones de Navier-Stokes. A través del método de Advección Semi-Lagrangiana y la técnica búsqueda de elementos con árboles octales, implementó un modelo de transporte de escalares sobre dicho código. Finalmente validó el código y verificó el rendimiento del mismo.

Bibliografía

- [1] Cantero, M., Garcia, M., Balachandar, S. An Eulerian–Eulerian model for gravity currents driven by inertial particles. *International Journal of Multiphase Flow.*, 2008. [1](#)
- [2] Hammami, M., Mseddi, M., Baccar, M. Numerical study of coupled heat and mass transfer in a trapezoidal cavity. *Engineering Applications of Computational Fluid Mechanics.*, **3**, 216–226, 2007. [1](#)
- [3] Hoult, D. Oil Spreading on the Sea. [1](#)
- [4] White, F. Mecánica de Fluidos., 2004. [3](#), [12](#)
- [5] Gunzburguer, M. Finite Element Methods for Viscous Incompressible Flows. 1989. [4](#), [6](#)
- [6] Morinishi, Y. Skew-symmetric form of convective terms and fully conservative finite difference schemes for variable density low-Mach number flows. *Journal of Computational Physics* 229, pág. 276–300, 2010. [5](#)
- [7] An, R., Wang, X. Two-Level Brezzi-Pitkäranta Stabilized Finite Element Methods for the Incompressible Flows. *Hindawi Publishing Corporation*, 2014. [6](#)
- [8] Zienkiewicz, O., Taylor, R. El Método de los Elementos Finitos. Formulación Básica y Problemas Lineales. 1994. [10](#)
- [9] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., et al. PETSc Users Manual. *Argonne National Laboratory.*, 2014. [11](#)
- [10] Ghia, U., Shin, C., Ghia, K. High-Re Solutions for Incompressible Flows Using the Navier-Stokes Equations and a Multigrid Method. *Journal of Computational Physics* 48, pág. 387–411, 1982. [12](#), [13](#), [14](#), [47](#)
- [11] Incropera, DeWitt, Bergman, Lavine. Fundamentals of mass and heat transfer. [15](#)

-
- [12] Donea, J., Huerta, A. Finite Element Methods for Flow Problems. 2003. [15](#), [17](#), [18](#)
- [13] Johnson, C. Numerical Solutions of partial differential equation by the finite element method., 1987. [17](#), [19](#)
- [14] Dari, E., Vénere, M. ALGORITMOS EFICIENTES PARA LA BÚSQUEDA DEL ELEMENTO DE UNA RED QUE CONTIENE UN PUNTO DADO., 1991. [31](#)
- [15] Corzo, S., Marquez, S., Gramajo, D., Nigro, N. Numerical Simulation of Natural Convection Phenomena. *Asociación Argentina de Mecánica Computacional.*, 2011. [33](#), [34](#), [35](#), [36](#), [48](#), [49](#)
- [16] Buscaglia, G., Dari, E. IMPLEMENTATION OF THE LAGRANGE-GALERKIN METHOD FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS. *International journal for numerical methods in fluid*, **15**, 23–36, 1992. [40](#)

Índice de figuras

2.1. Ejemplo de una malla de elementos hexaédricos.	7
2.2. Mapeo de coordenadas locales a globales.	8
2.3. Asignación de un rango de componentes de la matriz global y de los vectores a cada proceso.	12
2.4. Malla utilizada para resolver el problema.	13
2.5. A la izquierda, perfil de velocidad u_x sobre la línea $x = 0,5$. A la derecha, perfil de velocidad u_z sobre la línea $z = 0,5$. Con “o” solución obtenida en [10], y en línea continua la solución calculada con el código.	14
2.6. Líneas de corrientes obtenidas para $Re=100$. A la izquierda, resultados del código. A la derecha, resultados de [10].	14
2.7. Líneas de corrientes obtenidas para $Re=400$. A la izquierda, resultados del código. A la derecha, resultados de [10].	14
3.1. Representación del algoritmo de advección semi-lagrangiana para un caso 2D.	18
3.2. Ejemplo con un error considerable en el cálculo de $\mathbf{x}_{i,aprox}^*$ utilizando la fórmula 3.16 con $K = 1$. El valor exacto \mathbf{x}_i^* está dentro del dominio Ω	19
3.3. Soluciones al problema de transporte 1D con condición de Dirichlet en ambos extremos. Solución exacta con “o” y numérica en línea continua.	25
3.4. Soluciones al problema de transporte 1D. A la izquierda, problema con condición de Dirichlet en $x = 0$ y Neumann en $x = 1$. A la derecha, problema con condición de Neumann en $x = 0$ y Dirichlet en $x = 1$. Solución exacta con “o” y numérica en línea continua.	26
3.5. Soluciones al problema de transporte 1D. A la izquierda, con condición de Dirichlet en $x = 0$ y Robin en $x = 1$. A la derecha, problema con condición de Robin en $x = 0$ y Dirichlet en $x = 1$. Solución exacta con “o” y numérica en línea continua.	27
3.6. Soluciones al problema de transporte 1D con condiciones de Robin en ambos extremos. Solución exacta con “o” y numérica en línea continua.	27
4.1. Ejemplo de una red auxiliar estructurada como un árbol octal.	30

4.2. Descomposición de un hexaedro en cinco tetraedros.	31
4.3. Ejemplo de asignación de nodos locales y nodos fantasmas a un proceso .	32
4.4. Perfiles de velocidad u_x^+ sobre la línea $x^+ = 0,5$. A la izquierda, con $Ra=1e3$. A la derecha, con $Ra=1e4$. Con “o” solución de [15], y en línea continua solución con el código.	34
4.5. Distribución de la temperatura para $Ra = 1e3$ calculada con el código. .	35
4.6. Distribución de la temperatura para $Ra = 1e3$ calculada por [15].	35
4.7. Distribución de la temperatura para $Ra = 1e4$ calculada con el código. .	35
4.8. Distribución de la temperatura para $Ra = 1e4$ calculada por [15].	35
4.9. Tiempo de cálculo del código en función del número de procesos con $\Delta t=0,001$. A la izquierda, para el paso de advección. A la derecha, para las ecuaciones de N-S, el paso de advección, el de difusión y la suma de todos ellos.	37
4.10. Tiempo de cálculo del código en función del número de procesos con $\Delta t=0,1$. A la izquierda, para el paso de advección. A la derecha, para las ecuaciones de N-S, el paso de advección, el de difusión y la suma de todos ellos.	37
5.1. A la izquierda, simulación realizada con el código del transporte de temperatura en el refrigerante que atraviesa un elemento combustible de un reactor nuclear. La escala de colores representa la magnitud de la temperatura en el fluido. A la derecha, fotografía del elemento combustible real. Este cálculo se presenta solo a modo de ejemplo del tipo de problema que el código puede resolver.	41

Índice de tablas

4.1. Constantes para un $Ra=1e3$ calculadas con el código y resultados de los códigos “Fluent” y “OpenFOAM” de [15].	35
4.2. Constantes para un $Ra=1e4$ calculadas con el código y resultados de los códigos “Fluent” y “OpenFOAM” de [15].	36

Agradecimientos

Este trabajo no hubiese sido posible sin la dirección de Mariano Cantero y Enzo Dari, tuve la suerte y el agrado de contar con sus conocimientos desmedidos en la materia, experiencia, dedicación y firmeza sobre las decisiones. Si tuviese la oportunidad de volver al pasado, volvería a elegir el mismo grupo de personas.

Debo agradecer los aportes y la colaboración de Gustavo Buscaglia e Italo Tasso, quienes proveyeron la versión inicial del código y quienes me brindaron su ayuda para el uso del mismo.

Además a la UNR y a la CNEA por darme la formación académica en estos últimos años y a los docentes que participaron.

Por otro lado a todo grupo de MECOM, en especial a Federico Caccia y Ricardo Márquez, que me ayudaron en el proyecto.

Además quiero agradecer a toda mi familia que me llamó durante tres años todos los domingos para saludarme y ver como andaba, sufriendo siempre que me iba mal y alegrándose cuando me iba bien.

También a mis amigos y compañeros de clase que siempre me daban una mano cuando algo no me salía.

