

# "Elaboración de un código libre de CFD para el análisis de un fluido en estado transitorio mediante la aplicación de las ecuaciones de Navier Stokes y el método de Lattice Boltzmann"

María José Acosta Ayala

Ricardo Ortega del Ángel

Asesor: Rafael Medina Noguerón

Instituto Politécnico Nacional  
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Ticomán  
Ciudad de México

16 de julio de 2020

## **Resumen**

Se desarrolla un código abierto de CFD en Python basado en la técnica de diferencias finitas capaz de aplicar las ecuaciones de Navier Stokes, así como el método de Lattice Boltzmann con el fin de analizar un fluido en estado transitorio para el caso general del escalón. Se pretende que todos los alumnos del Instituto Politécnico Nacional tengan acceso al código con el fin de darles la oportunidad de estudiarlo y mejorarlo, poniendo así en práctica la ideología del *software libre*.

- Resultados
- Conclusiones

## **Abstarct**

# Índice general

<b>I</b>	<b>Estado del Arte</b>	<b>1</b>
<b>1.</b>	<b>El método de diferencias finitas en códigos de CFD</b>	<b>2</b>
1.1.	Introducción . . . . .	2
1.2.	Métodos de discretización . . . . .	3
1.3.	Historia del método de las diferencias finitas . . . . .	4
1.4.	Situación actual . . . . .	5
1.4.1.	Softwares de carácter industrial basados en el método de discretización de diferencias finitas . . . . .	5
1.5.	Ventajas y Desventajas de MDF, MVF y MEF . . . . .	8
<b>2.</b>	<b>El método de lattice Boltzmann</b>	<b>9</b>
2.1.	Introducción . . . . .	9
2.2.	Orígenes del Método de Lattice Boltzmann . . . . .	10
2.2.1.	Autómatas celulares . . . . .	10
2.2.2.	De LGCA al Método de lattice boltzmann . . . . .	13
2.2.3.	Las ecuaciones de Lattice Boltzmann . . . . .	15
2.3.	Códigos de Lattice Boltzmann existentes . . . . .	18
2.3.1.	OpenLB . . . . .	18
2.3.2.	Palabos . . . . .	19
2.3.3.	PowerFLOW . . . . .	19
<b>II</b>	<b>Marco teórico</b>	<b>20</b>
<b>3.</b>	<b>Navier-Stokes y el método de las diferencias finitas, fundamentos teóricos</b>	<b>21</b>
3.1.	Ecuaciones de Navier Stokes . . . . .	21
3.1.1.	Conservación de la masa . . . . .	22
3.1.2.	Conservación de momento . . . . .	22
3.1.3.	Conservación de la energía . . . . .	26
3.2.	Turbulencia . . . . .	26
3.2.1.	Tamaño, tiempo y velocidad de las escalas de turbulencia . . . . .	26
3.2.2.	Estudio estadístico de turbulencia . . . . .	29
3.2.3.	Ecuaciones de Navier Stokes para flujo turbulento . . . . .	29
3.2.4.	Flujos turbulentos en la proximidad de una pared . . . . .	32
3.3.	Discretización por el método de las diferencias finitas . . . . .	32

3.4.	Condiciones de frontera . . . . .	37
3.4.1.	Neumann . . . . .	38
3.4.2.	Dirichlet . . . . .	38
3.5.	Generación de mallas . . . . .	38
3.5.1.	Tipos de configuración de malla . . . . .	39
<b>4.</b>	<b>El método de Lattice-Boltzmann fundamentos teóricos</b>	<b>43</b>
4.1.	Presión, Temperatura y Energía cinética . . . . .	43
4.2.	Función de distribución de probabilidad de velocidades de Maxwell . . . . .	47
4.2.1.	El espacio de velocidades tri-dimensional . . . . .	49
4.3.	Función de distribución de probabilidad de velocidades de Boltzmann . . . . .	50
4.4.	Ecuación de Boltzmann . . . . .	52
4.4.1.	Ecuación de transporte de Boltzmann . . . . .	52
4.5.	Aproximación BGKW . . . . .	53
4.6.	Arreglos lattice . . . . .	54
4.6.1.	Arreglo tri-dimensional . . . . .	54
<b>III</b>	<b>Simulaciones de Navier-Stokes y Lattice Boltzmann</b>	<b>58</b>
<b>5.</b>	<b>Caso de Estudio y condiciones</b>	<b>59</b>
5.1.	Caso del escalón (forward-facing step) . . . . .	59
5.2.	Condiciones generales para la simulación utilizando las ecuaciones de Navier-Stokes para un fluido tri-dimensional bajo efectos turbulentos . . . . .	60
5.3.	Condiciones generales para la simulación utilizando el método de lattice Boltzmann para un fluido tri-dimensional bajo efectos turbulentos . . . . .	60
<b>6.</b>	<b>Desarrollo del código</b>	<b>61</b>
6.1.	Programación del método de lattice boltzmann para un fluido turbulento tri-dimensional . . . . .	61
6.1.1.	Flujo incompresible e isotérmico . . . . .	61
6.2.	Programación de las ecuaciones de Navier-Stokes de un fluido turbulento tri-dimensional . . . . .	84
6.2.1.	Flujo bidimensional . . . . .	84
6.2.2.	Flujo tridimensional . . . . .	87
<b>IV</b>	<b>Resultados y Conclusiones</b>	<b>89</b>
<b>7.</b>	<b>Resultados y conclusiones de códigos en 2D</b>	<b>90</b>
<b>8.</b>	<b>Resultados y conclusiones de códigos en 3D</b>	<b>91</b>
<b>V</b>	<b>Anexos</b>	<b>92</b>

# Sección I

## Estado del Arte

# Capítulo 1

## El método de diferencias finitas en códigos de CFD

### 1.1. Introducción

El método favorito de la industria para resolver un fluido es la aplicación de la teoría del medio continuo mediante las ecuaciones de Navier-Stokes. Este método consiste en la obtención de una solución aproximada de dichas ecuaciones utilizando métodos numéricos, los cuales son programados y ejecutados mediante computadoras tomando en cuenta múltiples variables físicas en un cierto número de iteraciones y un cierto tiempo de simulación dado, con el fin de obtener un resultado final convergente y cercano a la realidad.

El uso de un método numérico incorrecto para resolver el fluido puede causar inestabilidad, imprecisión en el resultado final o consumir más tiempo computacional del que es necesario para converger, esto también depende de la manera en la que el algoritmo de solución del software maneje los errores acumulados de los métodos numéricos en las diferentes variables del fluido como la viscosidad, la presión, la velocidad, densidad, etc.

Se puede apreciar entonces, que la dinámica de fluidos computacional (**CFD**) es multidisciplinaria, incluyendo aspectos puramente matemáticos, computacionales e ingenieriles, y el resultado final es la suma total de la interacción de diferentes procesos que involucran estas tres disciplinas.

Estos procesos deben ser compatibles unos con otros. Se deben utilizar las ecuaciones correspondientes al estudio que se realizará, el método de discretización espacial y temporal deben ser suficientemente precisos para representar el dominio o volumen de control (*procesos matemáticos*), el método de solución en el cual se manejan los errores acumulados propios de las iteraciones del método numérico debe ser el más eficiente (*proceso ingenieril*) y por último se debe usar una computadora con capacidades adecuadas para entregar resultados a tiempo (*proceso computacional*).

Para el ingeniero de CFD resulta difícil encontrar la correcta combinación de variables que funcionen en un caso de estudio. Como se ha mencionado antes, una variable cualquiera que sea incorrecta puede causar la divergencia del modelo, que se obtengan resultados que no son confiables o simplemente los resultados pueden tardar en llegar más de lo esperado. Es por esto que el ingeniero de CFD debe conocer cuales son las ventajas y desventajas del

sinfín de opciones que se pueden usar así como sus aplicaciones, ya que el enfoque debe ser el correcto desde el inicio del planteamiento de una simulación. En esta situación se incluyen las herramientas de trabajo, que son los múltiples softwares de CFD, comerciales, de tipo libre y open source. Es importantísimo que se conozcan las bases matemáticas y físicas sobre las cuales está hecho el software puesto que, aunque todos los softwares resuelven el fluido, no todos lo hacen de la misma manera, una diferencia fundamental entre ellos por ejemplo, es **el método de discretización** del dominio.

## 1.2. Métodos de discretización

Existen diferentes métodos de discretización. Cada uno ofrece ciertas ventajas y desventajas y por ello las aplicaciones son diferentes. Los métodos de discretización más comunes son los siguientes:

- Método de diferencias finitas (FDM)
- Método de volúmenes finitos (FVM)
- Método de elementos finitos (FEM)

### El método de las diferencias finitas

Es el enfoque más antiguo para discretizar las ecuaciones diferenciales parciales de Navier-Stokes en su forma diferencial. Se toma una ecuación diferencial parcial en forma diferencial y se sustituye por un conjunto finito de ecuaciones algebraicas en términos de una o más variables dependientes evaluadas en las coordenadas indicadas. Un ejemplo de un conjunto de ecuaciones algebraicas son las series de Taylor.

El método de las diferencias finitas dificulta la solución del fluido alrededor de geometrías complejas, por otro lado, para geometrías fácilmente adaptables a una malla cartesiana (*figuras cuadradas*), la solución se ha encontrado precisa y estable. Se debe tomar en cuenta que existe un error de truncamiento, naturalmente se obtiene al usar un método de aproximación y no un método matemático exacto.

Algunos esquemas de diferencias finitas son los siguientes:

- Diferencia progresiva (forward)
- Diferencia regresiva (backward)
- Diferencia central
- Crank-Nicholson
- McCormack

## El método de volúmenes finitos

Es definitivamente el más popular entre los software de CFD comerciales, de tipo libre y de código abierto, este método se basa en discretizar la forma integral de las ecuaciones diferenciales parciales de Navier-Stokes dando lugar a la formación de soluciones débiles.

Se establece que las variables deben cumplir una continuidad desde la entrada a la salida en diminutos volúmenes que juntos forman un dominio.

Es parecido al método de diferencias finitas e incluso en ciertas condiciones es equivalente a este.

## El método de elementos finitos

Subdivide un modelo tri-dimensional en elementos muy pequeños de tamaño finito dando lugar a la formación de millones de nodos los cuales forman un sistema de millones de ecuaciones que se resuelve según el nivel de precisión requerido, pueden ser fenómenos lineales o no lineales. Se requiere una función de forma para resolver el fluido en cada nodo.

Actualmente se requiere demasiado tiempo computacional para resolver CFD con este método en geometrías sencillas y complejas.

### 1.3. Historia del método de las diferencias finitas

El método de las diferencias finitas ha sido utilizado para desarrollar códigos de CFD de capacidades limitadas, debido a su alta precisión pero poca adaptabilidad a geometrías complejas y en algunas ocasiones lentitud de obtener un resultado convergente. Se puede encontrar en el área académica y de investigación más que en una aplicación industrial con casos de estudio sencillos como el escalón, ahmed body, un cilindro perpendicular al flujo, esfera, etc.

Las aproximaciones por diferencias finitas son los más simples y antiguos métodos para resolver ecuaciones diferenciales parciales. Ya era usado por Euler en el año 1700 aplicado a una dimensión y fue extendido a a dos dimensiones por Runge en 1908. Sin embargo, la verdadera investigación de las técnicas de diferencias finitas en aplicaciones numéricas comenzó en los años 50 con la aparición de las primeras computadoras que ofrecieron las condiciones mas convenientes para indagar en los métodos numéricos.

Esta nueva tecnología fue aprovechada por el laboratorio nacional de Los Álamos, en donde se desarrolló el primer modelo funcional de CFD en 1957 utilizando el método de discretización de diferencias finitas.

En 1967, Douglas Aircraft desarrolló un método funcional de análisis en tres dimensiones para CFD. El análisis fue muy básico y solo funcionaba para resolver fluido alrededor de perfiles alares, poco después fue conocido como el **método de paneles**, en el cual la geometría analizada era simplificada significativamente para que los cálculos fueran fáciles de realizar.

Las ecuaciones potenciales fueron incorporadas al CFD por Boeing en la década de 1970. Las ecuaciones de euler para flujos transónicos fueron incorporadas a los códigos de CFD en 1981. En 1990 la tecnología de las computadoras y el CFD se desarrollo tanto que los fabricantes de autos vieron la posibilidad de adoptar estas tecnologías. En 1995 General Motors



y Ford incorporaron el CFD como una herramienta mas para el diseño de sus vehículos.

## 1.4. Situación actual

El método de las diferencias finitas actualmente es raro de encontrar en cualquier software de CFD. Hay ciertos software especializados que lo usan, sin embargo es común que utilicen esta metodología como una opción elegida por el usuario mas no una metodología estándar en el software.

En el ámbito académico se pueden encontrar un sinnúmero de códigos basados en diferencias finitas, pues el objetivo es introducir a los estudiantes a la formación en la dinámica de fluidos computacional, esto se logra fácilmente con el método de diferencias finitas ya que es el más sencillo de entender y de programar. Es también sin duda, el método sobre el cual se basan las otras metodologías.

En el ámbito industrial, la metodología de las diferencias finitas no es muy útil ya que no resulta eficiente para esta aplicación. La industria requiere de soluciones rápidas, precisas y de bajo costo computacional. El método de las diferencias finitas cumple con estas condiciones siempre y cuando se utilicen geometrías sencillas, pero no es eficiente cuando hablamos de geometrías complejas como un perfil alar, un avión, un auto, entre otros.

Las limitantes en el método de las diferencias finitas no han provocado la eliminación completa de este método, existen ciertos softwares comerciales aplicados a la industria que utilizan esta metodología.

### 1.4.1. Softwares de carácter industrial basados en el método de discretización de diferencias finitas

#### FLOW-3D

Este software es desarrollado en Los Álamos, lugar tradicional del CFD, en el cual fue originado el primer código de CFD funcional de la historia.

Flow 3D presume de un mundo de aplicaciones entre las cuales se pueden encontrar:

- Aerodinámica
- Termodinámica
- Fundición y diseño de moldes
- Diseño de vehículos marítimos
- Estudios de estabilidad para vehículos aéreos y marítimos
- Simulación de olas.
- Estudio de interacción fluido-sólido.
- Estudios con interacción entre dos o más fluidos.

En su ultima versión ofrece los tres métodos de discretización conocidos entre los cuales esta el método de diferencias finitas. El método de diferencias finitas es utilizado en conjunto con el método de volúmenes finitos, con un algoritmo capaz de aplicar ambos métodos según convenga en las diferentes partes de una geometría compleja. Esto permite a Flow 3D acelerar el tiempo de simulación sin perder versatilidad en su aplicación ni precisión en sus resultados finales.

## TAIR (Transonic Airfoil Analysis)

Es un código que fue hecho con lenguaje Fortran, usaba un rápido algoritmo que resuelve ecuaciones conservativas completamente potenciales para campos de fluidos transónicos alrededor de perfiles alares.

Contaba con un procedimiento capaz de generar mallas curvilíneas bien adaptadas a una geometría de un perfil alar cualquiera mediante métodos de diferencias finitas, así como una sección para seguir la convergencia de cerca. Dicha sección automáticamente actualizaba los parámetros de la solución con el fin de incrementar la velocidad a la cual se llega a la convergencia o en dado caso prevenir que la solución divergiera. En modo de *default* sólo eran cambiados tres datos de entrada de un caso a otro: el número de Mach en corriente libre, ángulo de ataque y coordenadas del perfil alar. Variables como factores de relajación, aceleración y coeficientes de amortiguamiento temporales permanecían fijos. Con esta herramienta solo el 1 % de los casos divergía.

TAIR utilizaba una subrutina llamada GRGEN, la cual generaba una malla hecha mediante la metodología de diferencias finitas.

A pesar de su gran precisión y fácil convergencia las aplicaciones de TAIR eran limitadas, pues como su nombre lo dice solo se utilizaba para flujos subsónicos y transónicos en 2D alrededor de perfiles alares.

## HEAT 2 and HEAT 3

Es un código de computadora para flujos bidimensionales (HEAT 2) y tridimensionales (HEAT 3) en estado transitorio y estacionario usado principalmente por universidades e institutos de investigación.

Sus aplicaciones son:

- Problemas de conducción de calor
- Puentes térmicos
- Estimación de temperaturas en la superficie
- Riesgos de condensación
- Optimización de aislantes
- Análisis de sistemas de calentamiento de pisos
- Análisis de marcos de ventanas.

Ambos programas resuelven la ecuación de conducción de calor por el método explícito de las diferencias finitas.

### Canonical Navier Stokes (CaNS)

Es un código para simulaciones de fluidos de requerimientos computacionales masivos. Su principal objetivo es resolver cualquier fluido incompresible, newtoniano del cual se pueda obtener los mejores beneficios utilizando un algoritmo de solución basado en la transformada rápida de Fourier (FFT). CaNS aprovecha la eficiencia del método FFT para resolver la ecuación de Poisson discretizada por el método de diferencias finitas.

En dos de tres direcciones la malla es regular y el solucionador puede resolver la siguiente combinación de condiciones de frontera:

- Neumann-Neumann
- Dirichlet-Dirichlet
- Periodic

En la tercera dirección del dominio, la malla puede ser no uniforme permitiendo el refinamiento de la misma en las zonas de mayor interés.

### Dune (Distributed and Unified Numerics Environment)

Es un programa hecho en lenguaje C++ para resolver ecuaciones diferenciales parciales mediante la fácil implementación del método de elementos finitos, volúmenes finitos y **diferencias finitas**.

DUNE es de carácter libre registrado bajo la licencia pública. La principal idea de DUNE es crear interfaces que permitan el uso eficiente de antiguas y nuevas librerías, soportando aplicaciones de alto desempeño computacional. No es un software basado en el método de diferencias finitas, sin embargo permite la implementación de este y cualquier modificación al algoritmo de MDF.

### FDS

Es un software de CFD hecho para la simulación de fluidos afectados por el fuego, sin embargo sus herramientas y fácil interfaz permiten ampliar sus aplicaciones, entre las cuales se pueden encontrar flujos de baja velocidad impulsados por fuentes térmicas y simulación de humo y transferencia de calor.

Utiliza el método de diferencias finitas en una malla rectangular. Se pueden utilizar modelos de turbulencia LES. El código está hecho en Fortran y fue desarrollado por el Laboratorio Nacional de Investigación de Incendios de Estados Unidos.

## 1.5. Ventajas y Desventajas de MDF, MVF y MEF

Discretización	Ventajas	Desventajas
Diferencias finitas	<ul style="list-style-type: none"> <li>■ Fácil de programar.</li> <li>■ Alta precisión.</li> <li>■ Fácil convergencia.</li> <li>■ Fácil aplicación en simulaciones numéricas directas.</li> </ul>	<ul style="list-style-type: none"> <li>■ Sólo se puede usar para geometrías sencillas de forma cuadrangular.</li> <li>■ Para aumentar precisión solo se puede disminuir el tamaño de celda, lo cual incrementa el costo computacional.</li> <li>■ No puede aplicarse mas que en coordenadas cartesianas.</li> <li>■ Existe un error de truncamiento.</li> </ul>
Volúmenes finitos	<ul style="list-style-type: none"> <li>■ Se adapta a geometrías complejas</li> <li>■ El resultado obtenido es preciso</li> <li>■ Menor costo computacional al refinar zonas de alta importancia</li> <li>■ La discretización se lleva a cabo directamente en el espacio físico del problema, por lo tanto no se necesitan transformaciones entre sistemas de coordenadas.</li> <li>■ Bajo ciertas condiciones resulta equivalente al método de las diferencias finitas.</li> </ul>	<ul style="list-style-type: none"> <li>■ La programación no es sencilla.</li> <li>■ Alto costo computacional para geometrías sencillas de forma cuadrangular.</li> <li>■ Se pueden obtener soluciones débiles.</li> </ul>
Elementos finitos	<ul style="list-style-type: none"> <li>■ Se puede utilizar para geometrías complejas.</li> <li>■ Entrega una solución continua.</li> </ul>	<ul style="list-style-type: none"> <li>■ Difícil de programar.</li> <li>■ Alto costo computacional.</li> <li>■ Requiere de una función de forma.</li> </ul>

# Capítulo 2

## El método de lattice Boltzmann

### 2.1. Introducción

Existen dos enfoques principales para resolver las ecuaciones de transporte presentes en la dinámica de fluidos (calor, masa y momentum), el enfoque continuo mejor conocido como teoría del medio continuo y el enfoque discreto. En la teoría del medio continuo se pueden obtener ecuaciones diferenciales parciales u ordinarias asumiendo la condición de conservación de la energía, masa y momentum en un volumen de control infinitesimal. Sin embargo, resolver dichas ecuaciones se torna complejo o a veces imposible, es por esto que se recurre a las metodologías de discretización mencionadas en el capítulo anterior.

Las discretizaciones por diferencias finitas, volúmenes finitos y elementos finitos desde la teoría del medio continuo convierten las ecuaciones diferenciales parciales en un sistema de ecuaciones algebraicas que pueden ser resueltas de una manera iterativa hasta alcanzar valores convergentes.

La escala que se aprecia en estos métodos de solución es macroscópica, se asume que las partículas del fluido se comportan de una manera similar en todo el dominio (*Teoría del medio continuo*).

Por otro lado, desde el enfoque discreto se analiza cada una de las partículas que forman un dominio, para ello se deben identificar las fuerzas inter-moleculares, la posición y velocidad de cada partícula en cada instante de tiempo y resolver ecuaciones diferenciales ordinarias. A este nivel no hay definición de variables como la temperatura, presión, viscosidad, etc. Ya que dichas variables están definidas para un conjunto de partículas denominado fluido. Se relaciona la temperatura y presión a la energía cinética de las partículas y la frecuencia con la que estas "*bombardean las fronteras*" del volumen de control. Este método se llama **dinámica molecular (MD)**, tiene gran versatilidad pero su costo computacional es exageradamente alto.

Afortunadamente existe un punto medio entre estas dos maneras de abordar la dinámica de fluidos y se llama el **Método de Lattice Boltzmann (LBM)**. La principal idea de este método es eliminar la brecha entre la escala micrométrica de la dinámica molecular y la escala macrométrica de la teoría del medio continuo. El método de lattice Boltzmann pretende utilizar lo mejor de los dos mundos, no estudia el comportamiento de cada partícula en el fluido pero tampoco las considera a todas como similares. El método de lattice Boltzmann

agrupa colecciones de partículas mediante conceptos de mecánica estadística y las trata como unidades en el volumen de control. La propiedad de cada agrupación de partículas es representada por una función de distribución. (Figura 2.1)

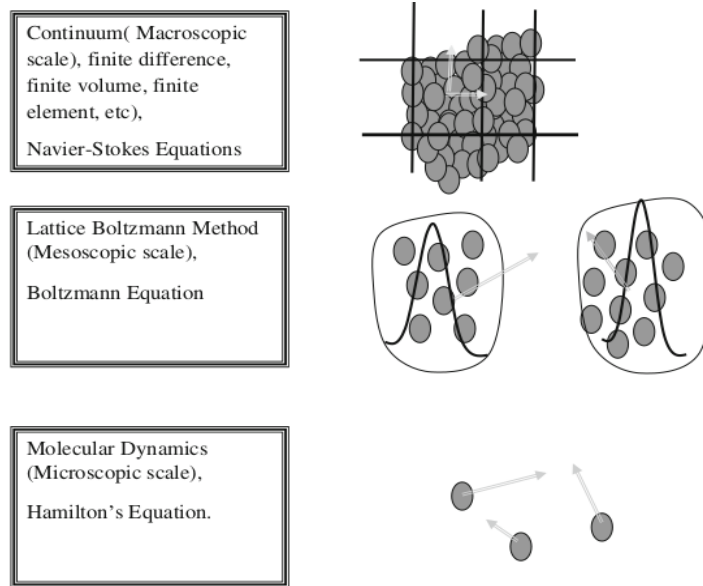


Figura 2.1: Ilustración de la escala macrómetrica en el medio continuo, la escala micrométrica en la dinámica molecular y la mesoescala que ofrece LBM.

El acercamiento mesoscópico que ofrece el método de lattice boltzmann en la mayoría de los casos, ofrece un incremento en la precisión con la que se capturan los fenómenos a analizar sin sacrificar el costo computacional, sobre todo en geometrías complejas y en zonas de desprendimiento de flujo en donde una malla de cualquier método continuo se debe refinar a un tamaño muy pequeño.

## 2.2. Orígenes del Método de Lattice Boltzmann

### 2.2.1. Autómatas celulares

En 1988 el Método de lattice Boltzmann, también conocido en castellano como *el método de boltzmann por redes* fue presentado por McNamara y Zanetti para dar solución a los inconvenientes mostrados por los autómatas celulares de un gas reticular (**Lattice Gas Cellular Automata, LGCA**). La investigación de dicho autómata prometía una forma revolucionaria de llegar a la solución de un fluido, sin embargo presentaba muchas dificultades.

Un autómata celular es un modelo matemático hecho para un sistema que evoluciona en el tiempo, se compone de un conjunto de celdas con distintos valores o estados que son alterados de un instante a otro en unidades de tiempo discreto (timestep).

Este conjunto de celdas evoluciona siguiendo lo establecido en una expresión matemática y la interacción resultante con los valores de las celdas vecinas, a esto se le ha llamado *regla de transición local*.

La idea de los autómatas celulares fue usarlos para modelar sistemas naturales que pueden ser descritos como una colección masiva de objetos simples que interactúen entre si.

Los elementos básicos de un autómata celular son:

- **Una rejilla o cuadrículado:**

Consiste en un espacio n-dimensional dividido homogéneamente, en donde cada división es llamada célula o celda.

- **Conjunto de valores o estados:**

Cada célula puede tomar un valor a partir de un conjunto de caracteres o alfabeto el cual usualmente puede ser expresado en valores o colores.

- **Configuración inicial:**

Es la asignación de un valor o estado a cada uno de las celdas de un reticulado.

- **Vecindad:**

Define un conjunto de celdas que se consideran adyacentes a una célula dada.

- **Función de transición local:**

Es la regla de evolución que determina el comportamiento del autómata. Define como debe cambiar de estado cada celda dependiendo de su estado anterior y de los estados anteriores de su vecindad. Suele darse como una expresión algebraica o un grupo de ecuaciones.

#### 2.2.1.1. Autómata celular para un gas reticular (LGCA)

Los autómatas celulares a pesar de utilizar reglas simples pueden simular sistemas complejos, lo cual resulta un requisito esencial si se requieren como herramienta de solución de fenómenos físicos como lo es el comportamiento de un fluido.

Los autómatas celulares son fáciles de implementar y especialmente son adecuados para computadoras de procesamiento masivos en paralelo. Son también incondicionalmente numéricamente estables.

Para programar un autómata celular existen ciertas ideas que se deben abordar:

- **Leyes de conservación:** Algunas leyes de la física están basadas en la conservación de cantidades. Las ecuaciones de Navier-Stokes por ejemplo, expresan la conservación de la masa, momentum y energía. Los autómatas celulares usados para una simulación deben mantener esta propiedad. Uno de los problemas principales en la construcción de un autómata celular de gas reticular es la aparición de falsos valores invariantes. En algunos casos estos invariantes ni siquiera están presentes en un fenómeno físico natural.

Un ejemplo de un invariante es la conservación de la diferencia del número de partículas que fluyen en direcciones contrarias dentro de una celda en el modelo LGCA HPP. Esta conservación adicional no tiene un homólogo en el mundo macroscópico de la dinámica de fluidos. A estas cantidades de conservación adicionales no reales se les llama invariantes espurios.

- **Fenómenos físicos transitorios:** Los fenómenos físicos que no son estáticos implican la transportación de ciertas cantidades. La propagación (*transporte de cantidades*) de la información en un autómata celular no es posible para la mayoría de las reglas de transición locales.
- **Un autómata celular programado correctamente:** El comportamiento físico deseado en un autómata celular para un gas reticular, debe revelar en su límite macroscópico lo mismo que se puede obtener con teoría de mecánica estadística, esto en pocas palabras significa que en su límite macroscópico se deben poder obtener las ecuaciones de Navier-Stokes. Esta condición implica que la aplicación de ciertos conceptos de la mecánica estadística requieran que las reglas locales sean reversibles<sup>1</sup>.

Para que un autómata celular pueda ser ocupado en la solución de fenómenos físicos se deben cumplir las tres ideas antes mencionadas, sin embargo solo algunos tipos de celdas en los autómatas celulares cumplen las ideas que se discuten anteriormente. No es claro que forma deben tener las celdas del autómata celular o como construir el mismo con la certeza de cumplir los 3 enunciados anteriores. La forma de la celda modifica las propiedades de vecindad del autómata celular y con ello sus reglas de transición local, por ello algunas formas no permiten la obtención de las ecuaciones de Navier-Stokes en sus límites macroscópicos.

Para simplificar el problema de construir un autómata celular funcional para fenómenos físicos el LGCA divide la regla local en una **regla de colisión y otra de propagación**, de esta manera se asegura de que la regla de transición local sea capaz de transportar información. Por esta diferencia algunos autores no consideran el LGCA como un autómata celular si no solo es nombrado como *gas reticular*.

El primer modelo de autómata celular de retícula de gas se propuso en 1973 por Hardi, Pomeau y Pazzis. Al modelo se le nombró HPP por las iniciales de sus tres nombres. Se utilizaron las reglas de transición local para lograr que la masa y el momento se conservaran. El modelo HPP es un modelo bidimensional, donde el fluido se considera como un conjunto de partículas virtuales, las cuales se encuentran colocadas en los nodos de una malla cuadrada (Figura 2.2). En cada nodo hay cuatro celdas. Estas celdas pueden estar vacías u ocupadas por lo menos por una partícula. Este principio de exclusión (principio de exclusión de Pauli) es característico para todos los autómatas celulares de retícula de gas.

Este sistema evoluciona de manera determinista y procede con un cambio local de colisiones (sólo están involucradas las partículas en un nodo) y de propagación a lo largo de enlaces con los nodos vecinos. En la etapa de colisión en el modelo HPP, cuando dos partículas se mueven a un mismo nodo con velocidades opuestas, sus vectores de velocidad girarán 90° después de la colisión; en cualquier otro caso, no se produce una colisión y las velocidades de las partículas no cambian.

A pesar de que el modelo HPP satisface las leyes básicas de conservación, las variables no satisfacen las ecuaciones de Navier Stokes en el límite macroscópico. En realidad, el modelo HPP fue diseñado como un modelo de partículas en lugar de un método computacional para

---

<sup>1</sup>Reversibilidad es la capacidad de un sistema macroscópico de experimentar cambios de estado físicos, sin un aumento en la entropía, resultando posible volver al estado inicial cambiando las condiciones que provocaron dichos cambios



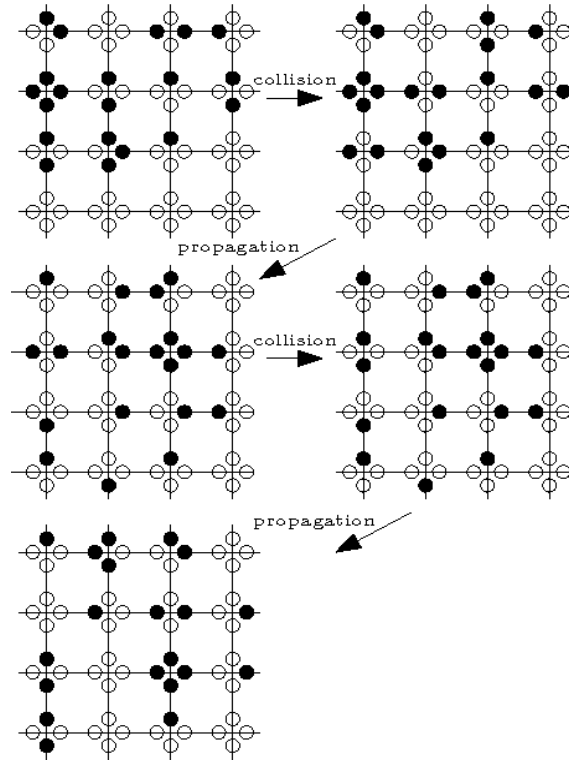


Figura 2.2: Evolución del modelo HPP.

flujos macroscópicos. Aunque, la idea básica detrás del modelo HPP abrió una nueva ventana para llegar a la solución de un fluido.

No obstante, en 1986 Frisch, Hasslacher y Pomeau se percataron de que utilizando una malla hexagonal (Figura 2.3) podían obtener las ecuaciones de Navier-Stokes y propusieron el modelo FHP, conocido así por las iniciales de los autores nuevamente. El uso de una malla hexagonal permitió la eliminación del invariante adicional que conservaba la diferencia de partículas que fluían en direcciones opuestas. Al no tener invariantes se podía llegar a la obtención de las ecuaciones de Navier-Stokes. Poco después, se propusieron muchas ampliaciones del método y generalizaciones como lo fue el uso de una celda con forma de hipercubo centrado en las caras (FCHC) para simulaciones 3D.

### 2.2.2. De LGCA al Método de lattice boltzmann

A pesar de que se encontraron modelos funcionales de autómatas celulares mediante el LGCA. Este padecía en ciertos aspectos listados a continuación

- 1.-Generaba mucho ruido estadístico
- 2.-Solo permitía el uso de un número de Reynolds bajo
- 3.-Funciones de complejidad exponencial en el tiempo computacional
- 4.-Invariantes espurios

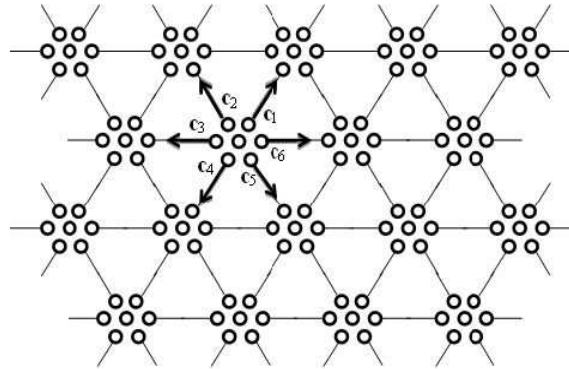


Figura 2.3: Modelo FHP.

### 2.2.2.1. Ruido estadístico

Como cualquier método de partículas, LGCA estaba expuesto a una gran cantidad de fluctuaciones estadísticas. El problema era que aun se tenían demasiados datos innecesarios provenientes del uso de las ecuaciones de dinámica molecular para la solución de las colisiones de partículas, la generación de datos era masiva aun y cuando se ocuparan funciones de distribución.

### 2.2.2.2. Número de Reynolds bajo

El máximo número de Reynolds que se podía lograr con una simulación de LGCA es controlado por el máximo número de colisiones por unidad de tiempo que el autómata es capaz de soportar. El problema es una vez mas que el estudio de los autómatas celulares no deja de lado el hecho de que resolver dinámica molecular requiere un alto costo computacional. En los años 80's el Reynolds más alto que se logró estudiar con el LGCA fue de  $Re = 20$ .

### 2.2.2.3. Funciones de complejidad exponencial en el tiempo computacional

Como se ha mencionado anteriormente no todas las celdas permitían la obtención de las ecuaciones de Navier Stokes en un limite macroscópico. Esto se dificultó aún más cuando el fenómeno requería ser estudiado en 3D. Una solución a esto fue el uso de celdas de la forma de un hipercubo centrado en las caras (FHCH). El uso de este cubo como celda tri-dimensional en un autómata celular tenía las propiedades adecuadas para lograr la obtención de las ecuaciones de Navier-Stokes, sin embargo el costo computacional era regido por la función  $O(2^b)$  en donde  $b$  es el número de colisiones por unidad de tiempo. Entre más colisiones se analizaran en el espacio-fase<sup>2</sup> el tiempo computacional incrementaba de manera exponencial, por ejemplo si se analizaban 24 colisiones por unidad de tiempo la computadora debía analizar  $2^{24} = 16,128$  millones de operaciones booleanas por cada celda, lo cual requería de un altísimo costo computacional.

<sup>2</sup>El espacio-fase es el lugar abstracto en donde se consideran todas las posibilidades de colisión de las partículas en estudio

Una vez mas se había encontrado una solución para terminar con los problemas de forma de las celdas de los autómatas, sin embargo esta solución arrojaba un nuevo problema que en pocas palabras era el incremento exponencial del tiempo computacional para llegar a la solución.

#### 2.2.2.4. Invariantes espurios

Debido a la ausencia de fuentes y sumideros externos los fluidos en un medio continuo conservan la masa, momentum y energía, nada más y nada menos que eso. Los fluidos discretos en cambio, necesitan de valores de conservación espurios adicionales a la conservación de masa y momentum. A pesar de funcionar en los autómatas celulares, los invariantes deben ser usados con cuidado en las simulaciones ya que en algunas ocasiones pueden provocar la obtención de resultados erróneos. Se desarrollaron varios métodos para identificar invariantes espurios y crear soluciones ante estos, sin embargo no era posible darle solución a todos. Estos invariantes espurios se encuentran en las reglas de transición local.

### 2.2.3. Las ecuaciones de Lattice Boltzmann

#### 2.2.3.1. Ecuación no-lineal de Lattice Boltzmann ( LBE )

La primera ecuación de lattice boltzmann fue generada en respuesta a los múltiples problemas que presentaba el LGCA específicamente al ruido estadístico. Poco después, se descubrió que las otras anomalías del LGCA podían ser trabajadas mediante el enfoque de la ecuación de lattice boltzmann, de esta manera LBE se convirtió en un tema de investigación. La investigación de la LBE fue en dirección a mantener lo mejor del LGCA y arreglar todos los problemas que este presentaba.

Esta fue propuesta por McNamara y Zanetti en 1988 con el explicito propósito de hacer a un lado el problema de ruido estadístico. La idea es simple, utilizar un promedio de grupos de población de estudio. Este cambio de perspectiva es el mismo que plantea la teoría cinética continua, en donde se establece que en vez de estudiar moléculas individuales podemos crear colecciones de ciertas variables que representen una nube de datos microscópicos promediados a lo largo de un periodo de tiempo discreto.

Se trabaja con promedios obtenidos en ciertas regiones de un reticulado o enrejado, de manera que los valores que forman el promedio varíen muy poco en tiempo y espacio. El promedio de estas variables es representado por una función de distribución la cual será usada para cubrir una cierta área del reticulado en vez de un solo nodo como se muestra en la figura 2.4.

Los valores promedio resultantes permiten la obtención de agrupaciones de propiedades que pueden ser tratadas como una sola, esto dará lugar a un incremento en el espaciado (incremento de micro-escala a meso-escala) y un nuevo timestep.

La ecuación de lattice boltzmann es la siguiente:

$$\Delta_i f_i = C_i(f_i, \dots, f_b) \quad (2.1)$$

Es una transcripción de la ecuación de micro-dinámica de LGCA reemplazando las partículas ( $n_i$ ) por una distribución de estas ( $f_i$ ). El ruido estadístico es eliminado ya que por

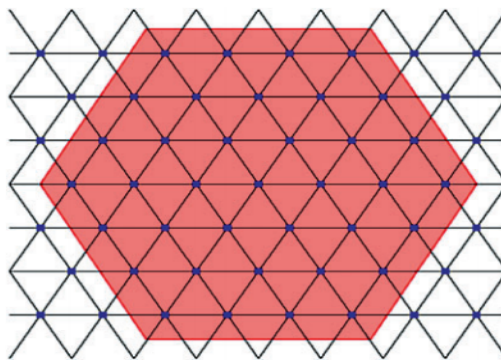


Figura 2.4: Representación de una región en un reticulado hexagonal.

definición ( $f_i$ ) es una cantidad promedio que permite la obtención de datos estadísticos suavizados. Se entiende al término  $C_i$  como el operador de colisiones definido para los automatas celulares en el cual se obtiene como resultado una partícula en un nodo (generación con operador booleano +1), se elimina la partícula (aniquilación con operador booleano -1) o no hay ningún cambio (operador 0).

El término  $\Delta_i f_i$  es la diferencia de número de partículas en la celda debido al cambio de estado de esta.

El mérito de la ecuación de boltzmann planteada por McNamara y Zanetti es el cambio de perspectiva en la solución del fluido, perspectiva que permitió avanzar a un punto medio entre las soluciones de dinámica molecular y las de teoría del medio continuo con Navier-Stokes.

El precio a pagar con este nuevo planteamiento no es algo que se deba despreciar por supuesto. En el lado de la física, los efectos de estudiar cada partícula se pierden claramente. La solución que se pueda alcanzar mediante la ecuación de lattice boltzmann no será exacta ya que se usan promedios de cantidades.

A pesar de que el uso de la ecuación de boltzmann demuestra ser menos revolucionario que el uso del LGCA, hay algunas ideas que mantienen la investigación de la ecuación de boltzmann en desarrollo como son las siguientes:

Primeramente, la LBE se puede utilizar fácilmente y de manera eficiente en cualquier tipo de computadora. En segundo lugar, puede adaptar aspectos mesoscopicos de la física que no son fáciles de lograr en los modelos basados en la teoría del medio continuo.

Es preciso mencionar que a pesar de que el ruido estadístico fue eliminado, aun quedaban muchas anomalías del LGCA a las cuales se debía dar solución. En particular, la indisponibilidad de lograr simulaciones en 3D era igual de mala que en el LGCA, esto involucra los problemas con el bajo número de Reynolds y la complejidad exponencial de tiempo computacional. La LBE aun mantenía un operador de colisiones de múltiples partículas. Es aquí en donde se encontraba el problema del tiempo computacional, se debían resolver las colisiones necesariamente para describir fenómenos de gran escala en la dinámica de fluidos, y como se ha mencionado dicho operador era no lineal lo cual provocaba inestabilidades numéricas y un aumento en la complejidad de solución. Si se quería atacar el problema del tiempo computacional se debía trabajar con el operador de colisiones.

### 2.2.3.2. La Ecuación cuasi-lineal de lattice boltzmann

El problema presente en el LGCA de la inviabilidad de la LBE no lineal para simulaciones en 3D no duro mucho. Higuera y Jiménez propusieron una solución casi al mismo tiempo que McNamara y Zanetti. El objetivo final siempre fue llegar a las simulaciones macroscópicas de la dinámica de fluidos, para lograr esto la intención era identificar los grados de libertad innecesarios en el movimiento de las partículas y trabajar en la elaboración de procedimientos sistemáticos para deshacerse de estos.

Higuera y Jimenez plantearon la ecuación cuasi-lineal como se muestra a continuación.

$$\Delta_i f_i = A_{ij}(f_i, \dots, f_j^e) \quad (2.2)$$

El factor distintivo en esta ecuación es el uso de la matriz  $A_{ij}$ . Anteriormente se debía almacenar el estado inicial de la celda y el estado siguiente, después leer estos datos y resolver las colisiones de las partículas. Sin embargo, gracias a la matriz  $A_{ij}$  obtenida con la expansión chapman-enskog, ya no se necesitaba almacenar cada estado si no se podían resolver las colisiones de manera analítica. La función exponencial de tiempo computacional cambia de  $O(2^b)$  a  $O(b^2)$  lo cual utilizando el ejemplo anterior de complejidad exponencial cambia de un número de  $2^{24} = 16,128$  millones de operaciones booleanas en cada celda a  $24^2 = 576$  operaciones. Esto permitía la simulación de fenómenos en 3D, aunque la limitante de un bajo número de Re aun existía y se sumaba otro problema más, la ecuación de boltzmann cuasi-lineal quedó expuesta a inestabilidades numéricas.

### 2.2.3.3. Ecuación de boltzmann con colisiones

Ahora que los dos inconvenientes más significativos de los modelos de LGCA, el ruido estadístico y la inviabilidad de lograr simulaciones en 3D, han sido arreglados, la atención se tornó hacia la tercer mayor limitante. La dificultad de resolver difusividad con altos valores de momentum, en otras palabras el LGCA está limitado al uso de bajos números de Re.

El problema era de nuevo que la restricción de difusividad es causa de que la ecuación original de boltzmann no lineal y la ecuación cuasi-lineal no pueden producir una sola colisión que el LGCA no permita, en pocas palabras la naturaleza del LGCA aun estaba presente en las ecuaciones de boltzmann y aunque la matriz  $A_{ij}$  permitió el cálculo de colisiones en un solo paso y de manera analítica, la cantidad de partículas en dicha matriz se obtenía de un cálculo anterior, el cual era la obtención de la función de distribución de probabilidad de las partículas según lo que las leyes de la microdinámica permitieran. La idea fue entonces que el cálculo de colisiones no fuera dictado por una teoría microscópica sino que fuera un parámetro libre sustentado en algo totalmente diferente. El nombre de este sustento es *descomposición espectral*.

Con la descomposición espectral se logró establecer a voluntad una cantidad de colisiones virtuales mediante una función de distribución de equilibrio discreta.

La ecuación de boltzmann con colisiones permitió el estudio de fenómenos 3D a bajas viscosidades (Altos numeros de Re) sin riesgo de sufrir inestabilidades numéricas.

### 2.2.3.4. El modelo de Lattice Bhatnagar-Gross-Krook

Se considera como el fin del desarrollo de las ecuaciones de lattice boltzmann y fue utilizado por Cercignani en 1990 para simplificar la ecuación de boltzmann. La matriz  $A_{ij}$  en la ecuación de boltzmann funcionaba como un factor de relajación sobre el tiempo en la función de distribución de equilibrio. Sin embargo era un factor múltiple de relajación. El hecho de establecer una condicionante de conservación en la matriz para las variables de masa y momentum forzaba a la función de distribución de equilibrio local ( $f_i$ ) a regresar a su estado inicial según los valores inversos de la diagonal de la matriz  $A_{ij}$ . La pregunta que surgió entonces fue *¿Por que no utilizar solo un factor de relajación sobre el tiempo que controle la viscosidad y mantenga la simulación dentro de los límites establecidos por las leyes de la conservación y estabilidad numérica?*

Esto significa pasar de un factor de relajación múltiple a un factor único de relajación. De esta manera todas las variables utilizan la misma escala de relajación sobre el tiempo  $\tau = 1/\omega$ . Con tal cambio la ecuación de boltzmann se convierte en:

$$\Delta_i f_i = -\omega(f_i, \dots, f_j^e) \quad (2.3)$$

La ecuación 2.3 se discretiza y se asume que es válida únicamente a lo largo de direcciones específicas (*enlaces*). A partir de este punto se obtienen diferentes modelos de redes entre los cuales se pueden encontrar *D1Q3*, *D2Q5*, *D3Q15*, mismos modelos que son el fundamento de la programación del Método de lattice boltzmann <sup>3</sup>

## 2.3. Códigos de Lattice Boltzmann existentes

### 2.3.1. OpenLB

Es un código libre basado totalmente en el método de lattice boltzmann, se puede utilizar para simulaciones en 2D y 3D. Esta hecho en lenguaje C++ y utiliza archivos con sub-rutinas establecidas que se pueden crear mediante el comando *makefile*. Ofrece una gran versatilidad de condiciones de frontera para redes ya programadas las cuales gracias a la orientación de objetos se pueden combinar y facilmente utilizar en una misma simulación.

Aún conserva ciertos aspectos de los antecesores del LBM, como lo son la función de colisión y propagación de las partículas, la cual se puede usar en 2D y 3D también.

El usuario puede encontrar un camino fácil a plantear su simulación asi como para la obtención de datos, ya que de igual manera existen sub-rutinas programadas que permiten la extracción de resultados específicos de la simulación.

Cuenta con herramientas que permiten un ahorro de tiempo computacional según la simulación que se realice.

Resulta en una fácil implementación para simulaciones en paralelo, utiliza paraVIEW como visualizador de resultados y arroja datos en formato VTK.

Usualmente se usa para investigación y simulación de casos muy específicos para la industria. No se ha implementado formalmente en la industria ingenieril ya que este mercado esta dominado por los software libres basados en la teoría del medio continuo.

<sup>3</sup>Para un mejor entendimiento de los modelos de redes ver el capítulo 4

### 2.3.2. Palabos

Esta programado en lenguaje C++. Prácticamente no tiene dependencias externas, únicamente usa MPI y Posix en UNIX y por ello es fácil de utilizar en distintas plataformas, incluyendo la implementación en paralelo. Contiene interfaces de programación adicionales para lenguajes python y java. No cuenta con una interfaz gráfica y se necesita cierta programación para correr una simulación.

Se recomienda únicamente a investigadores e ingenieros con bases solidas acerca del método de lattice boltzmann. Es capaz de combinar una vasta cantidad de modelos físicos y aun asi mantener la eficiencia que lo caracteriza en simulaciones en paralelo.

Se puede usar para simulaciones en 2D y 3D. El código puede guardar datos en formato ASCII o archivos binarios lo cuales directamente pueden producir imagenes de formato GIF o se pueden generar archivos VTK que permiten una visualización en paraVIEW.

Se pueden guardar datos de la simulación en cualquier momento.

### 2.3.3. PowerFLOW

Es el único código de CFD comercial utilizado en la industria. Su potencial se ubica en la industria automotriz para velocidades en donde el flujo se puede considerar incompresible. Utiliza los dos enfoques descritos al inicio de este capítulo, el continuo y el discreto. Para zonas de desprendimiento de flujo se debe establecer regiones en donde se construirán las redes de boltzmann con un determinado tamaño. De esta manera resuelve el fluido alrededor del carro mediante las ecuaciones de Navier-Stokes y en las zonas de alta probabilidad de desprendimiento de flujo, utiliza el método de lattice boltzmann, el cual solo esta disponible en 3 modos de Redes(D3Q15, D3Q19 y D3Q27) según el nivel de detalle que se necesite. Es utilizado por compañías como Ford y Volvo en la industria automotriz comercial y en la industria de carreras para categorías GT, NASCAR y LeMans.

Los resultados indican alta cercanía al túnel de viento, con un costo computacional aceptable, poca probabilidad de divergencia en la solución y puede resultar fácilmente manejado por el usuario. Como factores en contra se puede encontrar la alta exigencia de una malla se superficie triangular de buena calidad por parte del software, lo cual se ve reflejado en el aumento de tiempo en la etapa de pre-proceso de una geometría compleja. También es importante mencionar que el costo de adquisición de licencias para este software es muy alto.

Su uso es práctico al utilizar PowerCASE (software de pre-proceso compatible con PowerFLOW) en windows y correr PoweFLOW en modo paralelizado en un ambiente UNIX con sistemas de computo de alto rendimiento.

## Sección II

### Marco teórico



## Capítulo 3

# Navier-Stokes y el método de las diferencias finitas, fundamentos teóricos

### 3.1. Ecuaciones de Navier Stokes

Las ecuaciones de Navier Stokes surgen como una propuesta para describir matemáticamente los cambios de propiedades de un fluido durante interacciones dinámicas o térmicas. Algunos ejemplos de estas propiedades son la velocidad, presión, temperatura, densidad y viscosidad. Dichas ecuaciones, reciben el nombre gracias a las aportaciones Cloude-Louis Navier y George Gabriel Stokes.

Las ecuaciones de Navier Stokes se definen como un conjunto de ecuaciones expresadas con derivadas parciales no lineales que describen el movimiento de un fluido. Las ecuaciones son ajustables dependiendo el contenido del problema y se expresan con base en los principios de conservación de masa, momento y energía.

Al día de hoy, no se dispone de una solución general para este conjunto de ecuaciones, y en ciertas condiciones no es posible hallar una solución analítica; por lo que se puede recurrir al análisis numérico para determinar una solución aproximada. Este proceso es que llevan a cabo diversos software de CFD, los cuales resuelven de forma iterativa las ecuaciones hasta asegurar la convergencia.

Para generar un modelo matemático conveniente, es fundamental observar el flujo de un fluido basado en sus propiedades cinemáticas. Por consiguiente, el movimiento del fluido puede investigarse con métodos lagrangianos o eulerianos. La descripción lagrangiana se basa en el monitoreo de una partícula de fluido que es lo suficientemente grande como para detectar propiedades. En el método euleriano, no se sigue ninguna partícula específica a través del camino; en cambio, se examina el campo de velocidad en función del tiempo y la posición.

Las ecuaciones de Navier-Stokes se expresan como la ecuación de continuidad para la masa; las ecuaciones de conservación para momento; y ecuación de conservación de la energía para la primera ley de la termodinámica. Todas estas ecuaciones se consideran simultáneamente para examinar los campos de fluidos y los flujos.

### 3.1.1. Conservación de la masa

De acuerdo con las leyes de la física, la masa en el volumen de control no se puede crear ni destruir. Es por ello que la ecuación de conservación de la masa (Ecuación 3.1), también expresada como ecuación de continuidad, establece que la diferencia de flujo másico en el sistema entre la sección de entrada y la de salida es cero:

$$\frac{D\rho}{Dt} + \rho (\nabla \cdot \vec{V}) = 0 \quad (3.1)$$

donde  $\rho$  es la densidad,  $V$  la velocidad y  $\nabla$  representa el operador nabla para expresar la divergencia;

$$\vec{\nabla} = \vec{i} \frac{\partial}{\partial x} + \vec{j} \frac{\partial}{\partial y} + \vec{k} \frac{\partial}{\partial z} \quad (3.2)$$

Mientras que la densidad es constante, se hace la suposición de flujo es incompresible con el propósito de simplificar la continuidad como se indica en la ecuación 3.3, lo que indica un proceso en estado estacionario:

$$\frac{D\rho}{Dt} = 0 \longrightarrow \nabla \cdot \vec{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (3.3)$$

### 3.1.2. Conservación de momento

La descripción se configura de acuerdo con la expresión de la Segunda Ley del Movimiento de Newton:

$$F = m \cdot a = m \cdot \frac{DV}{Dt} \quad (3.4)$$

donde  $F$  es la fuerza neta aplicada a cualquier objeto,  $a$  es la aceleración y  $m$  es la masa. En caso de que se esté estudiando un fluido como objeto, es conveniente dividir la ecuación entre volumen para generar una derivación en términos de densidad de la siguiente manera:

$$f = \rho \cdot \frac{DV}{Dt} \quad (3.5)$$

en el que  $f$  es la fuerza ejercida sobre la partícula por unidad de volumen ( $f = F/Vol$ ), y la densidad es la masa por unidad de volumen ( $\rho = m/Vol$ ). La  $\frac{D}{Dt}$  indica la derivada sustancial de la siguiente manera:

$$\frac{D(\cdot)}{Dt} = \frac{\partial(\cdot)}{\partial t} + u \frac{\partial(\cdot)}{\partial x} + v \frac{\partial(\cdot)}{\partial y} + w \frac{\partial(\cdot)}{\partial z} = \frac{\partial(\cdot)}{\partial t} + V \cdot \nabla(\cdot) \quad (3.6)$$

Por otro lado, se puede descomponer a la fuerza por unidad de volumen en  $f_{cuerpo}$  y  $f_{superficie}$ .

$$f = f_{cuerpo} + f_{superficie} \quad (3.7)$$

donde  $f_{cuerpo}$  es la fuerza aplicada sobre toda la masa de partículas

$$F_{cuerpo} = m \cdot g$$

$$\frac{F_{cuerpo}}{Vol} = \frac{m}{Vol} \cdot g$$

$$f_{cuerpo} = \rho \cdot g$$

$\rho$  es la densidad del fluido y  $g$  es la aceleración gravitacional.

Por otro lado, la fuerza externa que se aplica en la superficie de las partículas es  $f_{superficie}$ , y se expresa mediante presión y fuerzas viscosas (ecuación 3.8). Para esta deducción, se hará uso de la Figura

$$f_{superficie} = f_{presion} + f_{viscosas} \quad (3.8)$$

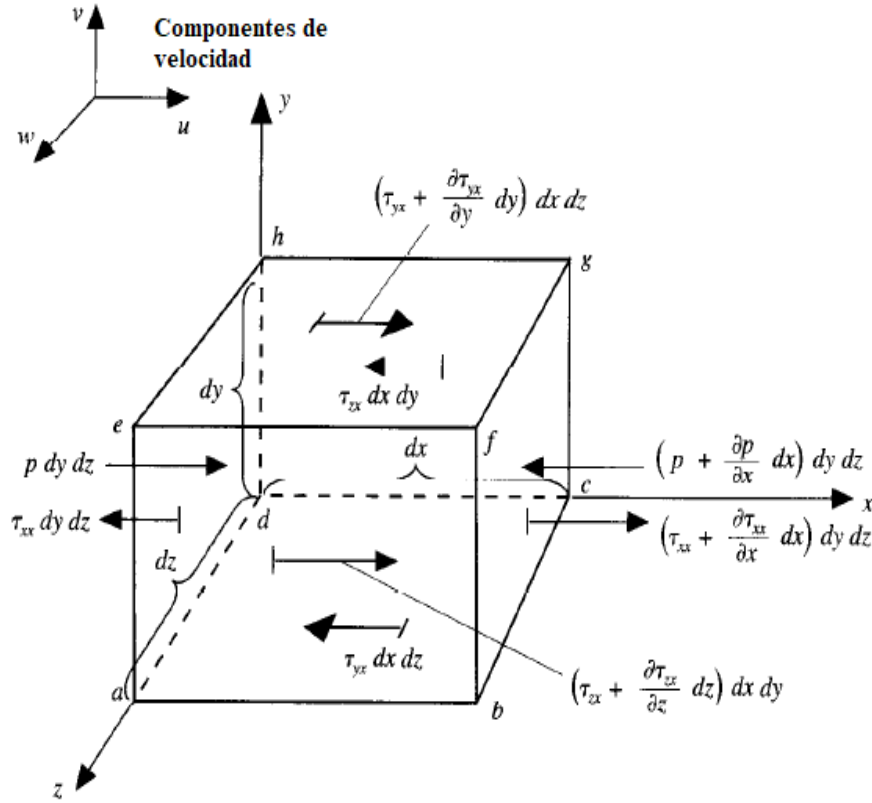


Figura 3.1: Elemento fluido infinitesimal y en movimiento. Sólo se muestran las fuerzas en la dirección  $x$ . Modelo utilizado para la derivación del componente  $x$  de la ecuación de momento. (Anderson (1995))

Cuando un fluido se encuentra en reposo, la presión hidrostática local  $P$  es el único esfuerzo que actúa en cualquier superficie del elemento de fluido, el cual siempre actúa hacia dentro y normal a la superficie. Por lo tanto, para un fluido en reposo se puede expresar el tensor de esfuerzo de la siguiente manera:

$$F_{presion} = \sigma_{ij} \cdot Area = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \cdot Area =$$

$$\begin{bmatrix} pdydz - (p + \frac{\partial p}{\partial x} dx)dydz & 0 & 0 \\ 0 & pdxdz - (p + \frac{\partial p}{\partial y} dy)dxdz & 0 \\ 0 & 0 & pdxdy - (p + \frac{\partial p}{\partial z} dz)dxdy \end{bmatrix}$$

Simplificando y dividiendo ambos lados entre volumen, que en este caso es  $Vol = dx \cdot dy \cdot dz$  obtenemos que:

$$f_{presion} = \begin{bmatrix} -\frac{\partial p}{\partial x} & 0 & 0 \\ 0 & -\frac{\partial p}{\partial y} & 0 \\ 0 & 0 & -\frac{\partial p}{\partial z} \end{bmatrix} = -\nabla p$$

Por último, queda agregar las fuerzas viscosas. Se considerará el fluido como Newtoniano, donde por definición el tensor de esfuerzo es linealmente proporcional al tensor de razón de deformación. Siguiendo el mismo procedimiento que para la fuerzas externas se tiene que:

$$F_{viscosas} = \tau_{ij} \cdot Area = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \cdot Area =$$

$$\begin{bmatrix} -\tau_{xx}dydz + (\tau_{xx} + \frac{\partial \tau_{xx}}{\partial x} dx)dydz & -\tau_{xy}dxdz + (\tau_{xy} + \frac{\partial \tau_{xy}}{\partial x} dx)dydz & -\tau_{xz}dxdy + (\tau_{xz} + \frac{\partial \tau_{xz}}{\partial x} dx)dydz \\ -\tau_{yx}dxdz + (\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy)dxdz & -\tau_{yy}dxdz + (\tau_{yy} + \frac{\partial \tau_{yy}}{\partial y} dy)dxdz & -\tau_{yz}dydz + (\tau_{yz} + \frac{\partial \tau_{yz}}{\partial y} dy)dxdz \\ -\tau_{zx}dxdy + (\tau_{zx} + \frac{\partial \tau_{zx}}{\partial z} dz)dxdy & -\tau_{zy}dxdy + (\tau_{zy} + \frac{\partial \tau_{zy}}{\partial z} dz)dxdy & -\tau_{zz}dxdy + (\tau_{zz} + \frac{\partial \tau_{zz}}{\partial z} dz)dxdy \end{bmatrix}$$

Simplificando y dividiendo ambos lados entre volumen ( $Vol = dx \cdot dy \cdot dz$ ) obtenemos que:

$$f_{viscosas} = \begin{pmatrix} \frac{\partial \tau_{xx}}{\partial x} & \frac{\partial \tau_{xy}}{\partial x} & \frac{\partial \tau_{xz}}{\partial x} \\ \frac{\partial \tau_{yx}}{\partial y} & \frac{\partial \tau_{yy}}{\partial y} & \frac{\partial \tau_{yz}}{\partial y} \\ \frac{\partial \tau_{zx}}{\partial z} & \frac{\partial \tau_{zy}}{\partial z} & \frac{\partial \tau_{zz}}{\partial z} \end{pmatrix}$$

Estableciendo que el tensor de esfuerzo viscoso para un fluido newtoniano incompresible con propiedades constantes es  $\tau_{ij} = 2\mu\epsilon_{ij}$ , donde  $\epsilon_{ij}$  es el tensor de razón de deformación. Así, la expresión anterior quedaría como

$$f_{viscosas} = \begin{bmatrix} \frac{\partial}{\partial x} 2\mu \left[ \frac{1}{2} \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial x} \right) \right] & \frac{\partial}{\partial x} 2\mu \left[ \frac{1}{2} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] & \frac{\partial}{\partial x} 2\mu \left[ \frac{1}{2} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \right] \\ \frac{\partial}{\partial y} 2\mu \left[ \frac{1}{2} \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right] & \frac{\partial}{\partial y} 2\mu \left[ \frac{1}{2} \left( \frac{\partial v}{\partial y} + \frac{\partial v}{\partial y} \right) \right] & \frac{\partial}{\partial y} 2\mu \left[ \frac{1}{2} \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right] \\ \frac{\partial}{\partial z} 2\mu \left[ \frac{1}{2} \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right] & \frac{\partial}{\partial z} 2\mu \left[ \frac{1}{2} \left( \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \right] & \frac{\partial}{\partial z} 2\mu \left[ \frac{1}{2} \left( \frac{\partial w}{\partial z} + \frac{\partial w}{\partial z} \right) \right] \end{bmatrix} =$$

$$\begin{bmatrix} 2\mu \frac{\partial^2 u}{\partial x^2} & \mu \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & \mu \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ \mu \frac{\partial}{\partial y} \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & 2\mu \frac{\partial^2 v}{\partial y^2} & \mu \frac{\partial}{\partial y} \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\ \mu \frac{\partial}{\partial z} \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) & \mu \frac{\partial}{\partial z} \left( \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) & 2\mu \frac{\partial^2 w}{\partial z^2} \end{bmatrix}$$

Ahora, considerando únicamente la componente  $x$  y sustituimos los términos encontrados para la ecuación 3.7 lo que da como resultado:

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \rho \cdot g_x + 2\mu \frac{\partial^2 u}{\partial x^2} + \mu \frac{\partial}{\partial y} \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) + \mu \frac{\partial}{\partial z} \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \quad (3.9)$$

Cabe resaltar que mientras las componentes de velocidad sean funciones suaves de  $x$ ,  $y$  y  $z$ , el orden de diferenciación es irrelevante. Por ejemplo, la primera parte del último término en la ecuación 3.9 se puede reescribir como:

$$\mu \frac{d}{dz} \left( \frac{\partial w}{\partial x} \right) = \mu \frac{d}{dx} \left( \frac{\partial w}{\partial z} \right)$$

Con esta suposición y sabiendo que la ecuación de continuidad es igual a cero ( $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$ ), la ecuación 3.9 queda de la siguiente manera:

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \rho \cdot g_x + \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (3.10)$$

Así, para cada dimensión cuando la velocidad es  $V(u, v, w)$ , las ecuaciones de Navier-Stokes para un flujo tridimensional incompresible se pueden expresar de la siguiente manera::

$$\rho \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = \rho g_x - \frac{\partial p}{\partial x} + \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (3.11)$$

$$\rho \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = \rho g_y - \frac{\partial p}{\partial y} + \mu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \quad (3.12)$$

$$\rho \left( \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = \rho g_z - \frac{\partial p}{\partial z} + \mu \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \quad (3.13)$$

$p$ ,  $u$ ,  $v$  y  $w$  son incógnitas en las que se busca una solución utilizando la ecuación de continuidad y las condiciones de frontera. Por último, se combinan las tres componentes en una ecuación vectorial:

$$\underbrace{\rho \frac{DV}{Dt}}_I = - \underbrace{\nabla p}_{II} + \underbrace{\rho \cdot g}_{III} + \underbrace{\mu \nabla^2 V}_{IV} \quad (3.14)$$

*I*: Término de convección

*II*: Fuerza aplicada en la superficie (presión)

*III*: Peso específico

*IV*: Fuerza viscosa

Además, la ecuación de la energía debe considerarse si hay alguna interacción térmica en el problema a estudiar.

### 3.1.3. Conservación de la energía

La conservación de la energía es la primera ley de la termodinámica en la que se establece que la energía no se puede crear ni destruir durante un proceso; sólo puede cambiar de forma. Este principio de conservación de la energía también se puede expresar como el cambio neto (aumento o disminución) de la energía total del sistema durante un proceso es igual a la diferencia entre la energía que entra y la energía que sale del sistema durante un proceso. Las interacciones de energía se ubican en las fronteras del sistema cuando lo cruzan, y representan la energía que gana o pierde un sistema durante un proceso.

El calor, el trabajo y el flujo másico son las tres formas en las que la energía se puede transferir hacia o desde un sistema. Sin embargo, las únicas dos formas de interacción de la energía relacionadas con una masa fija o sistema cerrado son las transferencias de calor y de trabajo.

Uno de los tipos comunes de ecuación de la energía es la siguiente:

$$\rho \left[ \underbrace{\frac{\partial h}{\partial t}}_I + \underbrace{\nabla \cdot (hV)}_{II} \right] = - \underbrace{\frac{\partial p}{\partial t}}_{III} + \underbrace{\nabla \cdot (k \nabla T)}_{IV} + \underbrace{\phi}_V \quad (3.15)$$

*I*: Cambio local con el tiempo

*II*: Término convectivo

*III*: Trabajo por presión

*IV*: Flujo de calor

*V*: Término de disipación de calor

## 3.2. Turbulencia

Los flujos turbulentos se caracterizan por tener fluctuaciones caóticas en el espacio y tiempo, la existencia de numerosas escalas de remolinos (*eddies*), alta difusividad, un número de Reynolds grande, disipación de energía en calor y una “memoria corta”; más adelante se hablará más a detalle de ellas.

Cabe recalcar que la turbulencia es una característica de los flujos de fluidos y no de los fluidos. Cuando el número de Reynolds es suficientemente alto, la mayor parte de la dinámica de la turbulencia es la misma en todos los fluidos (líquidos o gases). Las características principales de los flujos turbulentos no están controladas por las propiedades moleculares del fluido.

### 3.2.1. Tamaño, tiempo y velocidad de las escalas de turbulencia

Para que se mantenga un flujo turbulento, es necesario suministrar algún tipo de energía a las grandes escalas de remolinos, que posteriormente alimentan a remolinos más pequeños,

a este proceso se le llama cascada de energía. Así, se tienen tres tipos de escalas o tamaños para dichos eddies: escala integral, escala de Taylor y la escala de Kolmogorov. Una vez alcanzado el tamaño más pequeño de eddies (escala de Kolmogorov), la energía cinética es transformada en calor por disipación viscosa. **Este proceso está ligado con el alargamiento de vórtices que sólo ocurre cuando se analiza en tres dimensiones, por lo que en casos especiales de dos dimensiones no se tiene este tipo de cascada.**

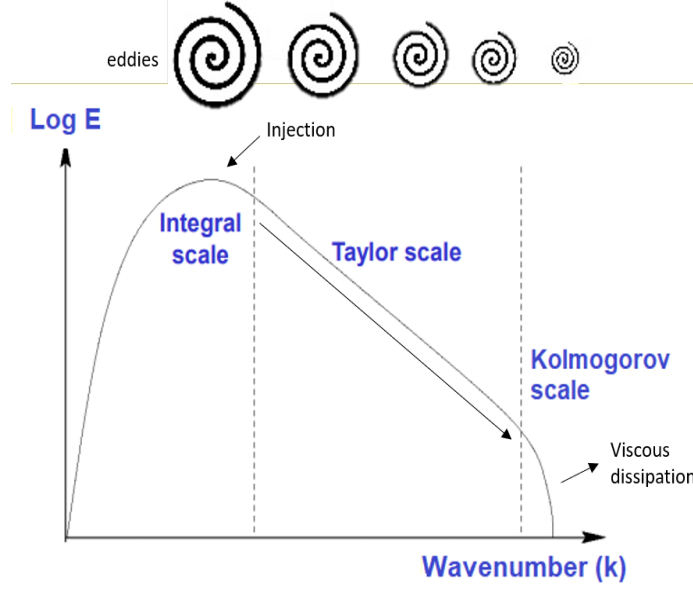


Figura 3.2: La energía cinética de la turbulencia ( $E$ ), para las diferentes escalas de movimiento (o longitudes de onda).

La difusión tiene un efecto importante, ya que después de ciertas interacciones entre vórtices, la turbulencia olvida sus condiciones iniciales. Por lo tanto, en muchas ocasiones la turbulencia tiende a un estado estadísticamente autosimilar, es decir que conserva .

La disipación viscosa es usualmente denotada como  $\epsilon$  y se define como la razón de cambio en la cual la energía cinética turbulenta se convierte en calor. También significa la destrucción de gradientes de velocidad fluctuantes por la acción de esfuerzos viscosos. Entonces, sucesivamente pequeños remolinos se disipan por la viscosidad molecular cerca de la pared, por lo que se dice que la escala o tamaño que alcanzan los remolinos más pequeños es independiente de las restricciones geométricas externas ya que dependen solamente de la viscosidad molecular y la disipación viscosa. La longitud de un vórtice (diámetro), tiempo y velocidad de un eddie en la escala de kolmogorov se denota como:

$$longitud = \eta \quad tiempo = t_\eta \quad velocidad = v_\eta$$

Haciendo un análisis dimensional, estableciendo que la viscosidad cinemática es  $\nu = \frac{m^2}{s}$  y la disipación  $\epsilon = \frac{m^2}{s^3}$  se tiene que la longitud de Kolmogorov es:

$$\eta = \nu^a \epsilon^b \quad donde \quad a = \frac{3}{4} \quad y \quad b = -\frac{1}{4}$$

por lo que

$$\eta = \left( \frac{\nu^3}{\epsilon} \right)^{1/4} \quad (3.16)$$

Consecuentemente el tiempo y la velocidad serán:

$$t_\eta = \left( \frac{\nu}{\epsilon} \right)^{1/2} \quad v_\eta = (\nu\epsilon)^{1/4}$$

Debido a que los esfuerzos viscosos gobiernan en este nivel (escala de kolmogorov), el número de Reynolds es considerado la unidad:

$$Re_\eta = \frac{\eta v_n}{\nu} = \frac{\left( \frac{\nu^3}{\epsilon} \right)^{1/4} (v\epsilon)^{1/4}}{\nu} = 1$$

Se puede concluir entonces, que los eddies en esta escala no dependen de la geometría si no únicamente de la viscosidad y la energía.

En orden ascendente se tiene la microescala de Taylor, la cual se puede ver como la escala en la que las fluctuaciones turbulentas están fuertemente relacionadas y por lo tanto es una longitud característica de fluctuación instantánea del gradiente de velocidad. Entonces, utilizando la longitud de la la microescala de Taylor ( $\lambda$ ), la disipación se puede expresar como:

$$\epsilon \sim \frac{\nu K}{\lambda^2} \quad (3.17)$$

Se considera que la microescala de Taylor es la escala intermedia que se encuentra entre la escala de Kolmogorov y la escala macroscópica (escala integral).

Para obtener la escala integral, al igual que para la escala de Kolmogorov, se puede realizar un análisis dimensional.

$$\Lambda = longitud \quad \epsilon = \frac{longitud^2}{tiempo^3} \quad V = \frac{longitud}{tiempo}$$

Una posible combinación dimensional para encontrar la longitud de la escala integral es:

$$\Lambda = \frac{V^3}{\epsilon} \quad (3.18)$$

y asumiendo que la velocidad  $V \sim \sqrt{K}$ , se tiene que la escala integral es igual a:

$$\Lambda \sim \frac{K^{3/2}}{\epsilon} \quad (3.19)$$



### 3.2.2. Estudio estadístico de turbulencia

Las magnitudes físicas, como la densidad y la velocidad, se definen operacionalmente, de modo que sus valores pueden ser determinados mediante la medición. No obstante, las definiciones operacionales de la probabilidad (por ejemplo, en términos de promedio de tiempo o promedio de un ensamble) aunque a menudo se utilizan, no son satisfactorias, aunque en la actualidad se consideren más precisas.

Se puede tomar como ejemplo el juego de moneda cara o cruz (águila o sol), donde se lanza la moneda un número de veces  $N = 1,000,000$ . Al final del juego se determina que hay una mayor probabilidad ( $p$ ) de que resulte cara, sin embargo, algunas otras veces resulta en cruz. Algo parecido sucede con la turbulencia. Al tener un ensamble de variables  $q^n(x, y, z, t); n = 1 \dots N$ , para un número de experimentos numéricos o físicos  $N$ , el promedio se puede expresar como:

$$\langle q \rangle = \bar{q} = \frac{1}{N} \sum_{n=1}^{n=N} q^{(n)}$$

La desviación instantánea del valor promedio ( $\bar{q}$ ), se conoce como fluctuación y usualmente se denota con una apóstrofo:

$$q' = q - \bar{q}$$

por lo que  $q$  será igual al valor promedio más las fluctuaciones ( $q = \bar{q} + q'$ ); la varianza será  $\overline{q'^2}$ ; y la intensidad de la fluctuación se define como la media cuadrática (**rms** del inglés *root mean square*) de las fluctuaciones  $\sqrt{\overline{q'^2}}$ .

### 3.2.3. Ecuaciones de Navier Stokes para flujo turbulento

La no linealidad de las ecuaciones de Navier Stokes es la base de la riqueza del fenómeno en dinámica de fluidos. Cuando se trata con flujos turbulentos, los términos no lineales dan origen a nuevos términos aparte de los componentes de velocidad promedio y presión promedio. Esto quiere decir que las ecuaciones contienen más incógnitas que número de ecuaciones, a lo que se le conoce como el problema de cierre de la turbulencia (*the turbulence closure problem*).

Así es como se introduce la descomposición de Reynolds, donde la velocidad y la presión total se divide en una parte promedio y otra fluctuante. Las parte promedio se denota con letras mayúsculas y la fluctuante con un apóstrofo, como se muestra a continuación:

$$u_i = U_i + u'_i \quad y \quad p = P + p'$$

La forma más compacta de obtener las deducciones al introducir la descomposición de Reynolds es planteando las ecuaciones con ayuda de tensores cartesianos donde la dirección está denotada con las coordenadas  $(x_1, x_2, x_3)$  o lo que es lo mismo  $(x, y, z)$  y la velocidad instantánea como  $(u_1, u_2, u_3)$  o  $(u, v, w)$ .

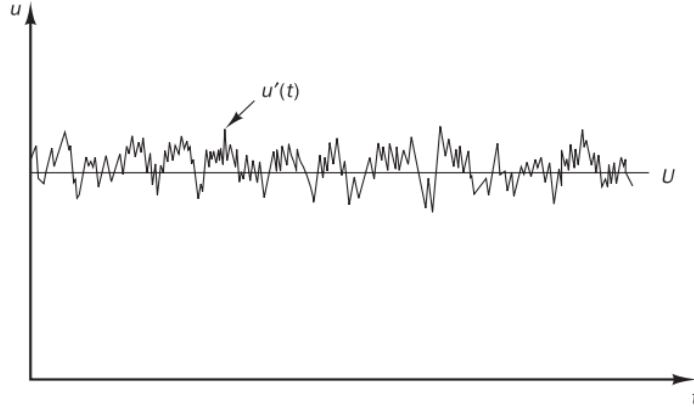


Figura 3.3: La imagen muestra lo que se busca lograr con la descomposición de Reynolds. La velocidad  $u$  corresponde a la suma de la velocidad promedio  $U$  más las fluctuaciones  $u'$ .

### 3.2.3.1. Condición de continuidad

Para flujos incompresibles, como ya se mencionó anteriormente, se tiene la ecuación 3.3 donde no se tiene divergencia. Al tratarse de un flujo turbulento, esta condición se mantiene para la velocidad promedio y para la fluctuante:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = \left( \frac{\partial U}{\partial x} + \frac{\partial u'}{\partial x} \right) + \left( \frac{\partial V}{\partial y} + \frac{\partial v'}{\partial y} \right) + \left( \frac{\partial W}{\partial z} + \frac{\partial w'}{\partial z} \right) = 0 \quad (3.20)$$

### 3.2.3.2. La ecuación de Reynolds

La ecuación de Reynolds se utiliza para describir los valores promedio del flujo, la cual se deduce de la ecuación de Navier Stokes para momento. Para esta deducción se utilizará la ecuación en la dirección de  $x$ :

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (3.21)$$

Este tipo de "promedio de Reynolds" (*Reynolds averaging*) es útil solo cuando las cantidades promediadas no cambian significativamente con el tiempo, de modo que el valor promedio de una componente fluctuante y sus derivadas durante dicho rango de tiempo es cero. Por lo tanto se pueden hacer las siguientes consideraciones como ejemplo:

$$\begin{aligned} \overline{u'} &= \overline{v'} = \overline{w'} = 0 \\ \overline{U} &= U \\ \overline{(V + v')(U + u')} &= \overline{VU + Vu' + v'U + v'u'} = \overline{VU} + \overline{v'u'} \end{aligned}$$

A la ecuación 3.21 se le incorpora la descomposición de Reynolds y se saca el promedio quedando como resultado:

$$\begin{aligned} \frac{\partial U}{\partial t} + \overline{(U + u') \frac{\partial(U+u')}{\partial x}} + \overline{(V + v') \frac{\partial(U+u')}{\partial y}} + \overline{(W + w') \frac{\partial(U+u')}{\partial z}} \\ = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \left( \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} \right) \end{aligned}$$

Desarrollando todos los términos llegamos a:

$$\begin{aligned} \frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + \overline{u' \frac{\partial u'}{\partial x}} + V \frac{\partial U}{\partial y} + \overline{v' \frac{\partial u'}{\partial y}} + W \frac{\partial U}{\partial z} + \overline{w' \frac{\partial u'}{\partial z}} = \\ -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \left( \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} \right) \end{aligned}$$

Recordando la regla de la cadena y despejando se pueden escribir los siguientes términos como:

$$\begin{aligned} \overline{\frac{\partial u' u'}{\partial x}} &= \overline{u' \frac{\partial u'}{\partial x}} + \overline{u' \frac{\partial u'}{\partial x}} \quad \rightarrow \quad \overline{u' \frac{\partial u'}{\partial x}} = \overline{\frac{\partial u' u'}{\partial x}} - \overline{u' \frac{\partial u'}{\partial x}} = \overline{\frac{\partial u' u'}{\partial x}} \\ \overline{\frac{\partial u' v'}{\partial y}} &= \overline{u' \frac{\partial v'}{\partial y}} + \overline{v' \frac{\partial u'}{\partial y}} \quad \rightarrow \quad \overline{v' \frac{\partial u'}{\partial y}} = \overline{\frac{\partial u' v'}{\partial y}} - \overline{u' \frac{\partial v'}{\partial y}} = \overline{\frac{\partial u' v'}{\partial y}} \\ \overline{\frac{\partial u' w'}{\partial z}} &= \overline{u' \frac{\partial w'}{\partial z}} + \overline{w' \frac{\partial u'}{\partial z}} \quad \rightarrow \quad \overline{w' \frac{\partial u'}{\partial z}} = \overline{\frac{\partial u' w'}{\partial z}} - \overline{u' \frac{\partial w'}{\partial z}} = \overline{\frac{\partial u' w'}{\partial z}} \end{aligned}$$

Observamos que aparece la ecuación de continuidad en los términos anteriores, cuyo valor es cero. Por lo tanto, la ecuación de Reynolds para  $x$ ,  $y$  y  $z$  es:

$$\begin{aligned} \frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} + W \frac{\partial U}{\partial z} = \\ -\frac{1}{\rho} \frac{\partial P}{\partial x} + \frac{\partial}{\partial x} \left( \nu \frac{\partial U}{\partial x} - \overline{u' u'} \right) + \frac{\partial}{\partial y} \left( \nu \frac{\partial U}{\partial y} - \overline{u' v'} \right) + \frac{\partial}{\partial z} \left( \nu \frac{\partial U}{\partial z} - \overline{u' w'} \right) \\ \frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} + V \frac{\partial V}{\partial y} + W \frac{\partial V}{\partial z} = \\ -\frac{1}{\rho} \frac{\partial P}{\partial y} + \frac{\partial}{\partial x} \left( \nu \frac{\partial V}{\partial x} - \overline{v' u'} \right) + \frac{\partial}{\partial y} \left( \nu \frac{\partial V}{\partial y} - \overline{v' v'} \right) + \frac{\partial}{\partial z} \left( \nu \frac{\partial V}{\partial z} - \overline{v' w'} \right) \\ \frac{\partial W}{\partial t} + U \frac{\partial W}{\partial x} + V \frac{\partial W}{\partial y} + W \frac{\partial W}{\partial z} = \\ -\frac{1}{\rho} \frac{\partial P}{\partial z} + \frac{\partial}{\partial x} \left( \nu \frac{\partial W}{\partial x} - \overline{w' u'} \right) + \frac{\partial}{\partial y} \left( \nu \frac{\partial W}{\partial y} - \overline{w' v'} \right) + \frac{\partial}{\partial z} \left( \nu \frac{\partial W}{\partial z} - \overline{w' w'} \right) \end{aligned}$$

Estas ecuaciones son muy parecidas a las de Navier Stokes, sin embargo aparece un nuevo término análogo al tensor de esfuerzos viscosos y es conocido como el esfuerzo turbulento o el tensor de esfuerzos de Reynolds (ecuación 3.22 y 3.23).

$$\tau = -\rho \overline{u'_i u'_j} \quad (3.22)$$

$$\tau_{ij} = -\rho \begin{pmatrix} \overline{u' u'} & \overline{u' v'} & \overline{u' w'} \\ \overline{v' u'} & \overline{v' v'} & \overline{v' w'} \\ \overline{w' u'} & \overline{w' v'} & \overline{w' w'} \end{pmatrix} \quad (3.23)$$

Este tensor es simétrico y por lo tanto tiene seis componentes independientes. Esto claramente ilustra el problema de cierre de la turbulencia. La ecuación de continuidad 3.20 junto con la ecuación de Reynolds constituyen cuatro ecuaciones donde se tienen diez incógnitas: la velocidad ( $U, V, W$ ), la presión ( $P$ ) y seis componentes del esfuerzo de Reynolds.

Por otro lado, se puede observar que los últimos términos de la ecuación de Reynolds corresponde a la divergencia total del tensor de esfuerzos  $\frac{\partial}{\partial x_j} \left( \nu \frac{\partial U_i}{\partial x_j} - \overline{u'_i u'_j} \right)$ , donde la primera parte representa el flujo neto de momento causado por movimientos moleculares, y la segunda parte el flujo neto de momento  $\rho u'_i$  causado por fluctuaciones macroscópicas de velocidad  $u'_j$ .

### 3.2.3.3. Energía cinética del flujo promedio y de las fluctuaciones turbulentas

Para esta sección, partiremos de la ecuación de la mecánica clásica de la energía cinética:

$$K = \frac{1}{2} m V^2$$

la cual representa la energía asociada con el movimiento de la partícula (velocidad) y su masa.

En el caso de un flujo turbulento, la energía cinética total ( $K^{(tot)}$ ) por unidad de masa es igual a la suma de la energía cinética promedio ( $K^{(M)} = \frac{1}{2} U_i U_i$ ) más la energía cinética turbulenta ( $K = \frac{1}{2} \overline{u'_i u'_i}$ ), esto es:

$$K^{(tot)} = K^{(M)} + K = \frac{1}{2} U_i U_i + \frac{1}{2} \overline{u'_i u'_i} \quad (3.24)$$

### 3.2.4. Flujos turbulentos en la proximidad de una pared

En la capa límite turbulenta, al igual que la laminar, existe la condición de no deslizamiento en las paredes sólidas. No obstante, la capa límite turbulenta se divide en dos regiones: interna y externa. Cada una de estas regiones cuenta con una escala de longitud que rige la dinámica cerca de la pared. Mientras más grande sea el número de Reynolds, se pueden apreciar mejor estas dos capas.

La región interna se caracteriza por ser una zona delgada donde las fuerzas viscosas predominan, haciendo que la velocidad se adhiera la condición de no deslizamiento. Por lo tanto, en esta región se tiene un mayor gradiente, mayores esfuerzos cortantes y mayor resistencia al avance a diferencia de una capa límite laminar.

En la región externa, los esfuerzos viscosos son despreciables en comparación de los esfuerzos turbulentos.

## 3.3. Discretización por el método de las diferencias finitas

La discretización es un proceso en el que una expresión matemática, como una ecuación diferencial o una integral que involucra funciones con un número infinito de valores, son aproximadas por diferentes expresiones algebraicas más sencillas aplicadas sobre un número finito de puntos o volúmenes discretos en un dominio. Esto quiere decir que un problema de cálculo se puede transformar en un problema algebraico que resulte programable.

Un método de discretización es por medio de diferencias finitas, donde se considera que las variables dependientes sólo existen en puntos discretos llamados nodos.

El método de las diferencias finitas se puede explicar utilizando la Figura 3.4 donde se representa una sección discretizada de un dominio. De forma general se establece que el

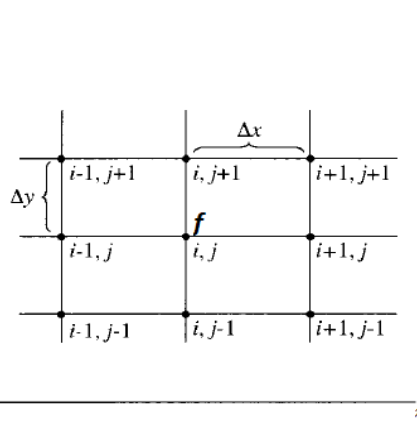


Figura 3.4: Malla de puntos discretos.

espacio entre los nodos en el eje  $x$  es uniforme y esta dado por  $\Delta x$ , y el espacio entre los nodos en el eje  $y$  es uniforme y dado por  $\Delta y$ . Usualmente el espacio entre los nodos no es uniforme en el dominio físico, por lo que se hace una transformación del dominio físico al lógico donde se tiene un espacio uniforme entre nodos, y es en el dominio computacional donde se hacen los cálculos numéricos.

En la Figura 3.4 se observa que los puntos de la malla están identificados por los subíndices  $i$  en el eje  $x$  y  $j$  sobre el eje  $y$ . Dicho esto, consideraremos los subíndices como coordenadas de un plano, por lo que al punto  $f$  le corresponden las coordenadas  $(i, j)$ . Siguiendo la misma lógica, el punto a la derecha del punto  $f$  tendrá las coordenadas  $(i + 1, j)$ , el punto a la izquierda  $(i - 1, j)$ , el punto arriba  $(i, j + 1)$  y el punto debajo  $(i, j - 1)$ .

La representación más común de diferencias finitas es utilizando las series de Taylor. Mediante las series de Taylor se aproxima una función en un punto determinado utilizando una serie de suma de potencias que se calculan a partir de derivadas.

Para ejemplificar esto, ahora asumiremos que el punto  $f$  se trata de una propiedad del fluido ( $f(x)$ ) como se muestra en la figura 3.5. En este caso la propiedad  $f(x + \Delta x)$  representa el nodo  $f(i + 1, j)$  y se expresa en términos de una serie de Taylor partiendo del punto  $x$ .

$$f(x + \Delta x) = f(x) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial^2 f}{\partial x^2} \frac{(\Delta x)^2}{2} + \frac{\partial^3 f}{\partial x^3} \frac{(\Delta x)^3}{6} + \dots \quad (3.25)$$

Lo que se busca con esto es llegar a una aproximación que se asemeje lo más posible a la gráfica real, siendo en este caso  $f(x)$  en la figura 3.5. Al agregar más términos a la ecuación 3.25, se obtiene una aproximación más precisa, de hecho, se convierte en una representación exacta de sólo cuando se lleva un número infinito de términos en el lado derecho.

Veamos el caso computacional de la ecuación 3.25 y retomando la figura 3.4.

$$f_{i+1,j} = f_{i,j} + \left( \frac{\partial f}{\partial x} \right)_{i,j} (\Delta x) + \left( \frac{\partial^2 f}{\partial x^2} \right)_{i,j} \frac{(\Delta x)^2}{2} + \left( \frac{\partial^3 f}{\partial x^3} \right)_{i,j} \frac{(\Delta x)^3}{6} + \dots \quad (3.26)$$

Ahora, para alcanzar el objetivo de diferencias finitas, se despeja a  $\left( \frac{\partial f}{\partial x} \right)_{i,j}$ :

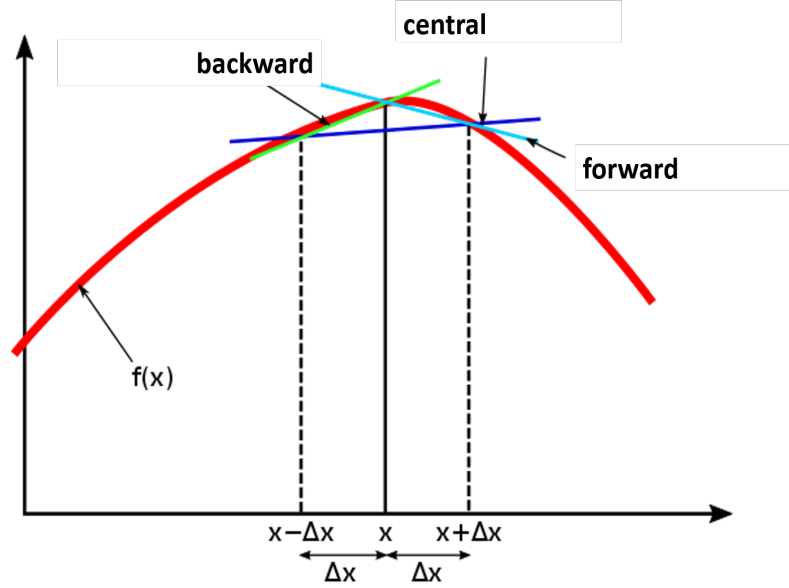


Figura 3.5: Aproximación por diferencias finitas donde se ilustran los tres tipos: forward, backward y central.

$$\left(\frac{\partial f}{\partial x}\right)_{i,j} = \underbrace{\frac{f_{i+1,j} - f_{i,j}}{\Delta x}}_{\text{Representación de la diferencia finita}} - \underbrace{\left(\frac{\partial^2 f}{\partial x^2}\right)_{i,j} \frac{\Delta x}{2} - \left(\frac{\partial^3 f}{\partial x^3}\right)_{i,j} \frac{\Delta x^2}{6} + \dots}_{\text{Error por truncamiento}} \quad (3.27)$$

Por lo que la derivada parcial real evaluada en el punto  $(i, j)$  se da en el lado izquierdo de la ecuación 3.27. El error por truncamiento nos indica lo que está siendo omitido de la ecuación. Como en este caso el término de orden más bajo en el error de truncamiento es  $\Delta x$  elevado a la primera potencia, esta expresión de diferencia finita se denomina como *precisión de primer orden*. Así, la ecuación 3.27 se puede reescribir como:

$$\left(\frac{\partial f}{\partial x}\right)_{i,j} = \frac{f_{i+1,j} - f_{i,j}}{\Delta x} + O(\Delta x) \quad (3.28)$$

Dado que la ecuación 3.28 se calcula a partir del nodos  $(i, j)$  e  $(i+1, j)$  es una aproximación de primer orden de tipo forward para  $\left(\frac{\partial f}{\partial x}\right)_{i,j}$ , ya que no se utiliza ningún punto a la izquierda de  $f(x)$ . Note que esto se ilustra en la figura 3.5 para los nodos  $x$  y  $x + \Delta x$ .

De forma similar se puede obtener una aproximación de primer orden de tipo backward, calculada a partir del nodo  $(i-1, j)$  como se muestra a continuación:

$$f_{i-1,j} = f_{i,j} + \left(\frac{\partial f}{\partial x}\right)_{i,j} (-\Delta x) + \left(\frac{\partial^2 f}{\partial x^2}\right)_{i,j} \frac{(-\Delta x)^2}{2} + \left(\frac{\partial^3 f}{\partial x^3}\right)_{i,j} \frac{(-\Delta x)^3}{6} + \dots$$

o de otra forma

$$f_{i-1,j} = f_{i,j} - \left( \frac{\partial f}{\partial x} \right)_{i,j} (\Delta x) + \left( \frac{\partial^2 f}{\partial x^2} \right)_{i,j} \frac{(\Delta x)^2}{2} - \left( \frac{\partial^3 f}{\partial x^3} \right)_{i,j} \frac{(\Delta x)^3}{6} + \dots \quad (3.29)$$

Despejando para  $\left( \frac{\partial f}{\partial x} \right)_{i,j}$ , se obtiene:

$$\left( \frac{\partial f}{\partial x} \right)_{i,j} = \frac{f_{i,j} - f_{i-1,j}}{\Delta x} + O(\Delta x) \quad (3.30)$$

Se conoce como backward dado que no utiliza ningún nodo del lado derecho de  $f$ , al contrario del tipo forward.

No obstante, en algunas aplicaciones de CFD, se requiere un grado de exactitud mayor. Por lo tanto, para construir una diferencia finita de segundo orden, basta con restar la diferencia finita tipo backward (ecuación 3.29) de la diferencia finita tipo forward (ecuación 3.26):

$$f_{i+1,j} - f_{i-1,j} = 2 \left( \frac{\partial f}{\partial x} \right)_{i,j} (\Delta x) + 2 \left( \frac{\partial^3 f}{\partial x^3} \right)_{i,j} \frac{(\Delta x)^3}{6} + \dots \quad (3.31)$$

Al igual que en los casos anteriores, se despeja para  $\left( \frac{\partial f}{\partial x} \right)_{i,j}$  y se obtiene:

$$\left( \frac{\partial f}{\partial x} \right)_{i,j} = \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x} + O(\Delta x)^2 \quad (3.32)$$

A este tipo de diferencia finita se le conoce como central de segundo orden debido al grado del error de truncamiento. Este tipo de diferencia está calculada entre los nodos  $(i+1, j)$  e  $(i-1, j)$ .

Hasta ahora sólo hemos estudiado tipos de diferencias finitas para derivadas parciales de primer orden. Sin embargo, dado que en esta investigación se tratará con flujos viscosos utilizando ecuaciones de Navier Stokes con derivadas parciales de segundo orden, es necesario hacer la discretización para dichos casos.

Para obtener diferencia finita de tipo central de segundo orden para una derivada parcial de segundo orden, basta con sumar la diferencia tipo forward y backward como se muestra a continuación:

$$f_{i+1,j} + f_{i-1,j} = 2f_{i,j} + \left( \frac{\partial^2 f}{\partial x^2} \right)_{i,j} (\Delta x)^2 + \left( \frac{\partial^4 f}{\partial x^4} \right)_{i,j} \frac{(\Delta x)^4}{12} \dots \quad (3.33)$$

Despejando para  $\left( \frac{\partial^2 f}{\partial x^2} \right)_{i,j}$  se tiene que:

$$\left( \frac{\partial^2 f}{\partial x^2} \right)_{i,j} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 \quad (3.34)$$

Cuando se quieren discretizar derivadas mixtas como  $\frac{\partial^2 f}{\partial x \partial y}$ , se derivan con respecto a  $y$  las ecuaciones 3.26 y 3.29, resultando en:

$$\left(\frac{\partial f}{\partial y}\right)_{i+1,j} = \left(\frac{\partial f}{\partial y}\right)_{i,j} + \left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} (\Delta x) + \left(\frac{\partial^3 f}{\partial x^2 \partial y}\right)_{i,j} \frac{(\Delta x)^2}{2} + \left(\frac{\partial^4 f}{\partial x^3 \partial y}\right)_{i,j} \frac{(\Delta x)^3}{6} + \dots \quad (3.35)$$

y

$$\left(\frac{\partial f}{\partial y}\right)_{i-1,j} = \left(\frac{\partial f}{\partial y}\right)_{i,j} - \left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} (\Delta x) + \left(\frac{\partial^3 f}{\partial x^2 \partial y}\right)_{i,j} \frac{(\Delta x)^2}{2} - \left(\frac{\partial^4 f}{\partial x^3 \partial y}\right)_{i,j} \frac{(\Delta x)^3}{6} + \dots \quad (3.36)$$

Luego, se resta la ecuación 3.36 de la ecuación 3.35:

$$\left(\frac{\partial f}{\partial y}\right)_{i+1,j} - \left(\frac{\partial f}{\partial y}\right)_{i-1,j} = 2 \left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} (\Delta x) + \left(\frac{\partial^4 f}{\partial x^3 \partial y}\right)_{i,j} \frac{(\Delta x)^3}{3} + \dots \quad (3.37)$$

y finalmente se despeja a  $\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j}$ :

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} = \frac{\left(\frac{\partial f}{\partial y}\right)_{i+1,j} - \left(\frac{\partial f}{\partial y}\right)_{i-1,j}}{2\Delta x} - \left(\frac{\partial^4 f}{\partial x^3 \partial y}\right)_{i,j} \frac{(\Delta x)^2}{6} \quad (3.38)$$

Se puede observar que ahora en denominador tenemos la derivada parcial  $\frac{\partial f}{\partial y}$  evaluada en los puntos de la malla  $(i+1, j)$  e  $(i-1, j)$  (figura 3.4). Esto nos lleva a pensar que se trata de una diferencia finita de tipo central de segundo orden en el eje  $x$ . No obstante, la derivada parcial es con respecto al eje  $y$ , es decir, el eje en el que la coordenada  $j$  varía. Por lo tanto, dichas derivadas se pueden sustituir por diferencias finitas de tipo central de segundo orden, pero tomando como puntos centrales las coordenadas  $(i+1, j)$  e  $(i-1, j)$  y modificando  $j$ .

$$\begin{aligned} \left(\frac{\partial f}{\partial y}\right)_{i+1,j} &= \frac{f_{i+1,j+1} - f_{i+1,j-1}}{2\Delta y} + O(\Delta y)^2 \\ \left(\frac{\partial f}{\partial y}\right)_{i-1,j} &= \frac{f_{i-1,j+1} - f_{i-1,j-1}}{2\Delta y} + O(\Delta y)^2 \end{aligned}$$

Por consiguiente, la ecuación 3.38 da como resultado:

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} = \frac{f_{i+1,j+1} + f_{i-1,j-1} - f_{i-1,j+1} - f_{i+1,j-1}}{4\Delta x \Delta y} + O[(\Delta x)^2, (\Delta y)^2] \quad (3.39)$$

Siguiendo la metodología anterior, se pueden obtener las aproximaciones con respecto a  $y$ . Para resumir, las aproximaciones respecto a  $x$  y  $y$  se enlistan en las tablas 3.1 y 3.2.



Derivada	Aproximación por diferencias finitas	Tipo
$\left(\frac{\partial f}{\partial x}\right)_{i,j}$	$\frac{f_{i+1,j}-f_{i,j}}{\Delta x}$	Forward
	$\frac{f_{i,j}-f_{i-1,j}}{\Delta x}$	Backward
	$\frac{f_{i+1,j}-f_{i-1,j}}{2\Delta x}$	Central de primer orden
$\left(\frac{\partial^2 f}{\partial x^2}\right)_{i,j}$	$\frac{f_{i+1,j}-2f_{i,j}+f_{i-1,j}}{(\Delta x)^2}$	Central de segundo orden
$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j}$	$\frac{f_{i+1,j+1}-f_{i+1,j-1}+f_{i-1,j+1}-f_{i-1,j-1}}{4\Delta x \Delta y}$	Mixta

Tabla 3.1: Aproximación por diferencias finitas para las derivadas en el eje  $x$ .

Derivada	Aproximación por diferencias finitas	Tipo
$\left(\frac{\partial f}{\partial y}\right)_{i,j}$	$\frac{f_{i,j+1}-f_{i,j}}{\Delta y}$	Forward
	$\frac{f_{i,j}-f_{i,j-1}}{\Delta y}$	Backward
	$\frac{f_{i,j+1}-f_{i,j-1}}{2\Delta y}$	Central de primer orden
$\left(\frac{\partial^2 f}{\partial y^2}\right)_{i,j}$	$\frac{f_{i,j+1}-2f_{i,j}+f_{i,j-1}}{(\Delta y)^2}$	Central de segundo orden
$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j}$	$\frac{f_{i+1,j+1}+f_{i-1,j-1}-f_{i-1,j+1}-f_{i+1,j-1}}{4\Delta x \Delta y}$	Mixta

Tabla 3.2: Aproximación por diferencias finitas para las derivadas en el eje  $y$ .

### 3.4. Condiciones de frontera

Al resolver ecuaciones diferenciales parciales donde buscamos encontrar el valor de la variable potencial en algún volumen de control, puede ser necesario imponer restricciones a

dicha variable en la superficie límite de ese espacio para obtener una solución única. Los dos tipos principales de estas condiciones de frontera son las condiciones de frontera de Neumann y Dirichlet.

### 3.4.1. Neumann

Una condición de frontera de Neumann impone la restricción de que la derivada direccional de una variable ( $\phi$ ) tiene valor en un lugar específico. Esta condición es usualmente utilizada cuando la derivada direccional normal a alguna superficie límite, denominada derivada normal  $\frac{d\phi}{dn}$ , es cero. Hay una serie de ocasiones en las que condición de frontera de Neumann en un sistema puede ser cero.

### 3.4.2. Dirichlet

## 3.5. Generación de mallas

Como se vio en la sección anterior, para el método de diferencias finitas se requiere que los cálculos se realicen sobre una colección de puntos de cuadrícula discretos conocida como malla. La forma en que se determina dicha cuadrícula se llama generación de malla y es un aspecto importante para el CFD. El tipo de malla que se elija para un problema determinado puede hacer o romper la solución numérica. Debido a esto, la generación de malla se ha convertido en una entidad por sí misma en CFD (Anderson (1995)).

La generación numérica de mallas surgió de la necesidad de encontrar solución a las ecuaciones diferenciales parciales de la mecánica de fluidos aplicadas en regiones físicas con una geometría compleja. De esta forma fue posible transformar un problema con una complicación en la forma física a un dominio más simple. Tales transformaciones pueden ser vistas como el establecer un sistema de coordenadas arbitrarias curvilíneo para el dominio físico.

Generalmente, se busca transformar un dominio físico a un cuadrado en dos dimensiones o a un cubo en tres dimensiones de tal manera que la frontera del cuadrado o del cubo corresponde a la frontera del dominio físico. El cuadrado o cubo es llamado dominio lógico o computacional y la transformación da origen a una frontera que conforma el sistema de coordenadas (Figura 3.6). La terminología empleada es flexible, por ejemplo, a las transformaciones se les conoce como mapas; el dominio físico es también llamado espacio físico, región física u objeto físico; y a la región lógica también se le conoce como dominio o espacio.

Para realizar este proceso, es necesario que el Jacobiano de la transformación no sea cero, ya que de ser cero la transformación falla en preservar las propiedades físicas y matemáticas esenciales de las ecuaciones alojadas.

Para el Jacobiano diferente de cero, la transformación tiene una inversa, que es el dominio computacional asignado al dominio físico (figura 3.7). La transformación inversa lleva los puntos del dominio computacional al dominio físico, donde forma una malla ver. Además, cuando los algoritmos son implementados en códigos computacionales, los cálculos son realizados en un espacio lógico, por lo tanto se puede pensar que la transformación es del espacio lógico al espacio físico.

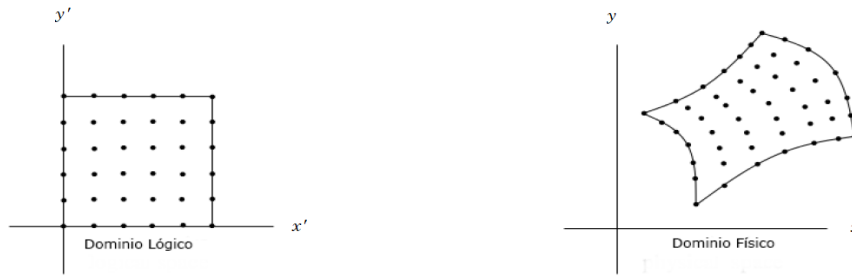


Figura 3.6: Cuadrícula de puntos en el dominio físico y lógico.

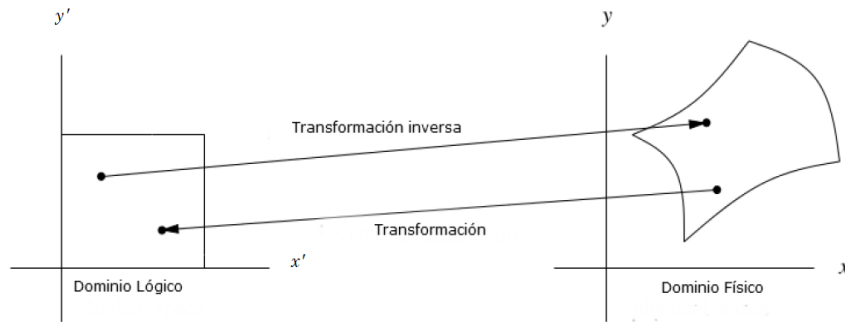


Figura 3.7: Transformación, mapeado, o sistema de coordenadas.

Por otro lado, las transformaciones que contienen un punto cero en el Jacobiano se llaman plegadas; uno de los objetivos más importantes en la generación de mallas es evitar una transformación plegada.

### 3.5.1. Tipos de configuración de malla

Hay tres tipos principales de estructuras de malla (y algunas modificaciones de estas) que se han empleado en intentos de resolver numéricamente las ecuaciones de Navier Stokes. Estos son: i) no escalonados, ii) escalonados y iii) "parcialmente escalonados". Estos, junto con sus modificaciones comunes, se muestran en la Figura 3.8. En cada caso, mostramos una sola celda de cuadrícula 2D y se indican las ubicaciones dentro de dicha celda donde se calculará cada una de las variables  $(u, v, p)$  McDonough (2013).

#### 3.5.1.1. No escalonados o escalonado natural

La cuadrícula no escalonada que se muestra en la Figura 3.8 (a) tiene todas las variables definidas en los en los vértices de las celdas, por lo que desde el punto de vista del análisis numérico de ecuaciones diferenciales parciales sería la opción más viable para la discretización de cualquier sistema. Con este tipo de cuadrícula, la implementación de las condiciones de frontera es sencilla ya que las variables discretas están definidas en cada vértice y coinciden con los puntos de la malla. Esto implica que las condiciones de Dirichlet se pueden

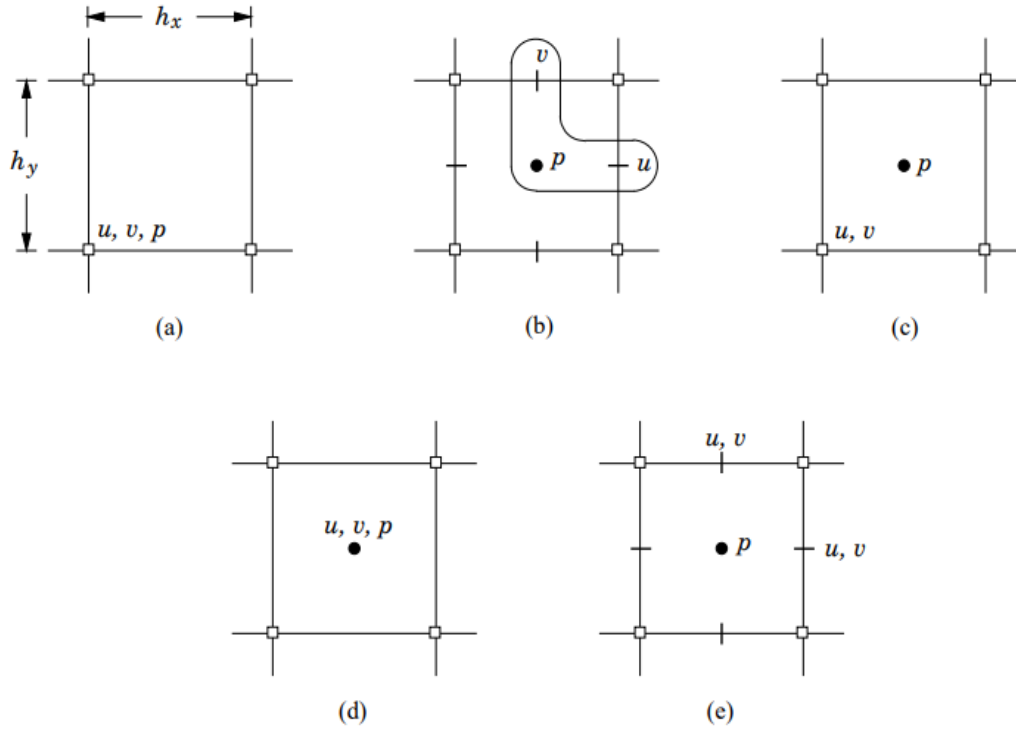


Figura 3.8: Estructuras de malla alternativas para resolver las ecuaciones de Navier Stokes para flujo incompresibles en 2D (a) escalonado natural o no escalonado, (b) escalonado, (c) escalonado parcialmente, (d) escalonado centrado en la celda y (e) escalonado con múltiples ecuaciones de momento McDonough (2013).

implementar sin ninguna aproximación, algo que es altamente deseable.

### 3.5.1.2. Escalonada

Este tipo de cuadrícula recibe el nombre de escalonada porque la ubicación de cada variable es diferente alrededor de la celda. La Figura 3.8 (b) muestra una agrupación típica de variables que resulta útil al momento de codificar algoritmos computacionales. Harlow y Welch en 1965 introdujeron este tipo de cuadrícula al construir el método de marcador y célula (marker-and-cell, MAC), y fue la primera configuración con la que se podía resolver las ecuaciones de Navier Stokes de manera confiable. En este tipo de configuración no existe el problema del desacoplamiento entre presión y velocidad a diferencia de la configuración no escalonada. No obstante, como se ve fácilmente, la condición de no deslizamiento no puede satisfacerse exactamente con este tipo de configuración. Esta dificultad llevó a la búsqueda de otros arreglos de cuadrícula.

### 3.5.1.3. Parcialmente escalonada

Una de esas configuraciones alternativas es la cuadrícula ALE (arbitraria lagrangiana-euleriana) de Hirt et al. (1997), que en términos generales denominamos "parcialmente es-

calonados" porque la presión se calcula en el centro de la celda como en la configuración escalonada, pero ambos componentes de la velocidad se calculan en los puntos naturales de la red de diferencias finitas como en el caso no escalonado.

Está claro que esta cuadrícula proporciona un remedio al problema de satisfacer la condición de no deslizamiento, aunque en algunos puntos límites no es posible especificar la presión exactamente (lo que también sucede para la cuadrícula escalonada). Sin embargo, los principales problemas con este enfoque son que el algoritmo necesario para implementar el método ALE es bastante complicado y puede ocurrir el desacoplamiento de la presión y la velocidad, pero, por lo general, con menos gravedad que en las no escalonadas. Como consecuencia de estas dificultades, los métodos ALE rara vez se usan.

#### 3.5.1.4. Centrada en la celda no escalonada

La cuadrícula centrada en la celda "no escalonada" (a veces, y más apropiadamente, denominada cuadrícula de ubicación conjunta) ha experimentado mucha popularidad tras la aparición del documento por Rhie y Chow en 1983. A menudo se la llama simplemente una cuadrícula no escalonada, pero como se puede ver en la Figura 3.8 (a) y (d), no se trata de la misma configuración. Si bien, todos los componentes de la solución se calculan en el centro de la celda. De hecho, las aproximaciones empleadas para este tipo de configuración son bastante similares a las utilizadas para las ecuaciones de flujo compresible. No obstante, es necesario alterar también la ecuación de Poisson de presión para que este enfoque tenga éxito en el caso incompresible.

Es importante recalcar que las variables de la cuadrícula de ubicación conjunta no satisfacen todas las condiciones de frontera, independientemente de si trata de flujo estable o turbulento. Por lo tanto, los resultados son generalmente menos precisos que los calculados en una cuadrícula escalonada. Sin embargo, esta forma de cuadrícula es ampliamente utilizada en software comerciales de CFD.

#### 3.5.1.5. Escalonada con múltiples ecuaciones de momento

La última configuración de malla para las variables dependientes de la que se hablará aquí consiste en la cuadrícula escalonada habitual que se muestra en la Figura 3.8 (b) pero con todos los componentes de velocidad calculados en cada pared de la celda, como se indica en la Figura 3.8 (e). Este enfoque está motivado por la inquietud respecto a la precisión de la interpolación de las componentes de velocidad en las paredes de la celda durante la construcción de los términos advectivos de la ecuación de momento. Se debe tener en cuenta que se realizan interpolaciones muy similares a la configuración no escalonada centrados en la celda para construir los flujos requeridos en la pared. Además, calcular todos los componentes de velocidad en todas las paredes de la celda satisface exactamente la condición del no deslizamiento y sin flujo, lo que no es posible con la configuración escalonada habitual como se ha señalado anteriormente. La configuración escalonada con múltiples ecuaciones de momento conduce a aproximadamente el doble de la aritmética que el método escalonado habitual para soluciones en 2D, y el triple en 3D.

Por otro lado, utilizando un promedio la configuración de cuadrícula escalonada no se pierde precisión, y por lo tanto hay pocas razones para aceptar la aritmética adicional re-

querida en la configuración escalonada con múltiples ecuaciones de momento a menos que en un problema específico la precisión de las condiciones de frontera sean un problema particularmente grave.

# Capítulo 4

## El método de Lattice-Boltzmann fundamentos teóricos

### 4.1. Presión, Temperatura y Energía cinética

Para entender la naturaleza hemos intentado descifrar la composición de la materia, y lo mas que sabemos es que cualquier sustancia esta hecha por moléculas y sub-moleculas. Las moléculas se han visualizado para entendimiento humano como esferas solidas moviéndose aleatoriamente de manera conservativa en un espacio. El movimiento de estas partículas satisface la conservación de la masa, momentum y energía, entonces la segunda ley de newton se puede aplicar para el estudio de estas partículas.

La segunda ley de newton establece que los cambios en un impulso lineal  $Fdt$  son iguales a los cambios en la cantidad de movimiento  $d(mv)$  de una partícula. De esta definición se obtiene la ecuación que indica la segunda ley de newton como la conocemos.

$$F = \frac{d(mv)}{dt} \quad (4.1)$$

Asumiendo que la masa es constante la ecuación anterior se puede simplificar a

$$F = m \frac{dv}{dt} = ma \quad (4.2)$$

En donde  $a$  es el vector aceleración. En dinámica de partículas es relevante saber la posición de las partículas en cada paso de tiempo.

Considerando a  $F$  en la ecuación (4.1) y (4.2) como una función conocida se puede obtener la velocidad de una partícula y después mediante la definición de velocidad la posición de esta relativa al origen.

$$v = \frac{dr}{dt} \quad (4.3)$$

Si la función  $F$  cambia, la partícula de masa  $m$  se moverá a diferente velocidad cambiando de  $v$  a  $v + vdt$  y su posición cambiará de  $r$  a  $r + rdt$  en donde  $r$  puede representar cualquier lugar en el espacio sin una dirección específica en el eje coordenado.

Cuando la función  $F$  se mantiene constante la partícula se moverá libremente de un lugar a otro sin cambiar dirección ni velocidad.

Consideremos que  $F$  se mantiene constante, que las partículas están en movimiento dentro de un contenedor sin interactuar unas con otras y agregaremos la variable de energía interna al sistema.

La magnitud de la velocidad de cada una de las partículas incrementará así como también la energía interna del sistema. Esto es provocado por el movimiento inducido de las partículas de un lado a otro del contenedor debido al choque de estas con las paredes del recipiente. En la escala microscópica que ofrece la dinámica molecular no existan como tal los conceptos de temperatura y presión sin embargo mediante el planteamiento descrito anteriormente se puede llegar a obtener definiciones como temperatura y presión utilizando la energía interna del sistema directamente relacionada con la velocidad de las partículas mediante el concepto de energía cinética.

Por ejemplo, al incremento en la energía cinética de las partículas se le relaciona con el concepto de temperatura, a la fuerza aplicada por las partículas en las paredes del contenedor con cierta frecuencia y por unidad de área se le relaciona con la presión. Un análisis mas profundo se podrá apreciar a continuación.

Consideremos una partícula moviéndose con una velocidad  $v_x$  en el interior de un tubo con longitud  $L$  y golpeando continuamente las paredes al final del tubo (fig. 4.1). La fuerza ejercida por la partícula en una de las paredes es igual a la relación de cambio del momentum de esta partícula ( $F = m \frac{dv}{dt}$ ). Tomando en cuenta que este comportamiento se observa en ambas paredes finales del tubo y que se asume un choque perfectamente elástico en cada colisión tenemos que.

$$F\Delta t = mv_x - (-mv_x) = 2mv_x \quad (4.4)$$

En donde  $\Delta t$  es el tiempo entre la colisión de la partícula en una pared y otra y  $2mv_x$  representa la suma de los momentum resultantes por el choque de las partículas en las dos paredes del cilindro. Recordemos que en la ecuación (4.4) solo se está considerando el movimiento de las partículas en dirección  $x$ .

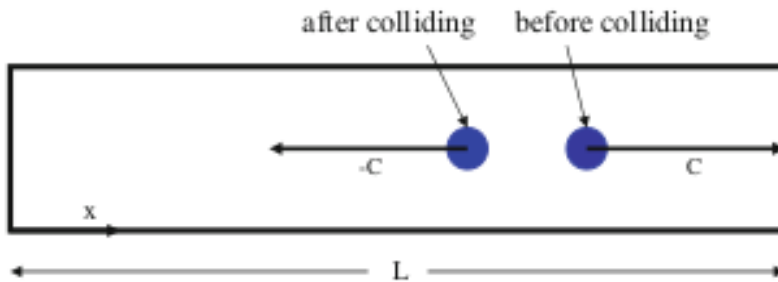


Figura 4.1: Cilindro de Longitud  $L$  con una partícula chocando en ambas caras finales del contenedor.

Utilizando la ecuación  $d = vt$  y despejando  $t$  podemos obtener:



$$\Delta t = \frac{L}{v_x} \quad (4.5)$$

La ecuación (4.5) representa el tiempo que tarda una partícula en llegar de un extremo a otro. Para obtener el tiempo en el que regresa a su punto de origen se tiene:

$$\Delta t = \frac{2L}{v_x} \quad (4.6)$$

Sustituyendo la ecuación (4.6) en la ecuación (4.4) tenemos que:

$$F \left( \frac{2L}{v_x} \right) = 2mv_x \quad (4.7)$$

En donde:

$$F = \frac{mv_x^2}{L} \quad (4.8)$$

La ecuación (4.8) describe la fuerza con la que la partícula golpea ambas caras del tubo en función de una masa constante  $m$ , una longitud constante  $L$  y una cierta velocidad  $v_x$ .

Considerando que un tubo no contiene solo una partícula, si no  $N$  número de partículas la ecuación (4.8) se puede generalizar a:

$$F = \frac{Nmv_x^2}{L} \quad (4.9)$$

En donde ahora la fuerza que golpea ambas caras del tubo es ejercida por  $N$  partículas.

Por otro lado, sabemos que el vector velocidad esta formado por la suma de sus tres componentes de velocidad en  $x$ ,  $y$  y  $z$ :

$$v^2 = v_x^2 + v_y^2 + v_z^2 \quad (4.10)$$

Recordemos que las partículas en el contenedor no son afectadas por ninguna fuerza externa, que las colisiones solo son con las paredes y estas choques son totalmente elásticos. Podemos asumir entonces que no hay razón alguna por la cual la velocidad de las partículas cambie en las direcciones  $x$  y  $z$  por lo tanto se puede decir que:

$$v^2 = 3v_x^2 \quad (4.11)$$

Entonces la ecuación (4.9) cambia a:

$$F = \frac{Nmv^2}{3L} \quad (4.12)$$

La ecuación (4.12) representa la fuerza de las partículas chocando contra las paredes del contenedor alejadas a una distancia  $L$  con una misma velocidad en sus tres componentes asumiendo choques elásticos y no interacción entre ellas. Para obtener la presión sobre una de las caras del cilindro necesitamos dividir la fuerza entre el área de una de las caras del cilindro.

$$P = \frac{Nmv^2}{3LA} \quad (4.13)$$

En donde  $LA$  se vuelven el volumen  $V$  completo del contenedor transformando la ecuación (4.13) en:

$$P = \frac{Nmv^2}{3V} \quad (4.14)$$

Si multiplicamos la ecuación (4.14) por  $\frac{3}{2}$  en ambos lados de la ecuación obtenemos lo siguiente:

$$P = \frac{2N(KE)}{3V} \quad (4.15)$$

En donde KE es la energía cinética  $KE = \frac{mv^2}{2}$ .

Por último se puede simplificar la ecuación (4.15) representando el número de moléculas por unidad de volumen con la variable  $n = \frac{N}{V}$ .

Entonces:

$$P = \frac{2}{3}nKE \quad (4.16)$$

La presión en una escala microscópica se relaciona con la energía cinética de las moléculas.

En este simple modelo se desprecian los efectos de las interacciones moleculares y el tamaño de la molecula. A pesar de ello, para un gas a temperatura ambiente, los resultados son sorprendentemente confiables. En un sistema real, la partícula tiene un volumen y las colisiones tienen lugar entre ellas.

Experimentalmente se ha encontrado que para gases lejos de condiciones críticas, la ecuación de estado puede representarse como:

$$PV = nRT \quad (4.17)$$

En donde:

$$n = \frac{N}{N_A}$$

$N_A$  = número de avogadro

$R$  = la constante de gases

Igualando las ecuaciones (4.17) y (4.16) se tiene que:

$$\frac{2N(KE)}{3V} = \frac{NRT}{N_A V} \quad (4.18)$$

Entonces la energía cinética KE es:

$$KE = \frac{3RT}{2N_A} \quad (4.19)$$

Utilizando la constante de Boltzmann  $k = \frac{R}{N_A}$   
La ecuación 4.20 queda entonces como:

$$KE = \frac{3kT}{2} \quad (4.20)$$

En donde T es:

$$T = \frac{2(KE)}{3k} \quad (4.21)$$

La conclusión es que la temperatura y la presión en el mundo macroscópico son solo la energía cinética de las moléculas multiplicada por factores escalares en cuyos respectivos casos son el numero de las moléculas por unidad de volumen  $n$  y la constante de boltzmann  $k$

## 4.2. Función de distribución de probabilidad de velocidades de Maxwell

La idea de calcular las trayectorias de un gran número de moléculas para lograr una visión macroscópica de un fluido era inalcanzable. Es por esto que Maxwell introdujo una nueva idea.

El conocimiento de la velocidad y posición de cada molécula en cada instante de tiempo ya no era importante como se describe en la ecuación (4.3). En cambio una función de distribución de probabilidad sería el parámetro con el cual se caracterizarían las partículas. Esta distribución respondería a la pregunta ¿Que probabilidad existe de encontrar moléculas con un cierto rango de velocidades en un instante de tiempo dado?.

Para un gas de  $N$  partículas con velocidades en la dirección  $x$  la probabilidad de encontrar partículas con velocidades entre  $v_x$  y  $v_x + dv_x$  esta indicada por:

$$\frac{vp_x}{N} = f(v_x) \quad (4.22)$$

En donde  $vp_x$  es el número de eventos probables de que una partícula se encuentre dentro del rango  $v_x$  y  $v_x + dv_x$  y  $N$  el número total de eventos, esto no solo incluye el rango establecido anteriormente si no todas las velocidades posibles en  $x$ .

La misma definición se puede usar para las direcciones  $y$  y  $z$ .

De esta manera la probabilidad tomando en cuenta las tres direcciones quedaría de la siguiente forma:

$$\frac{vp}{N} = f(v_x)f(v_y)f(v_z) \quad (4.23)$$

Debido a que  $f(v_x), f(v_y), f(v_z)$  son fracciones de la cantidad total de moléculas, si estas son integradas (suma total) deben de arrojar como resultado el total de partículas ( $N$ ) en un contenedor, expresado en fracciones porcentuales la suma total debe ser igual a 1.

$$\iiint f(v_x)f(v_y)f(v_z)dv_xdv_ydv_z = 1 \quad (4.24)$$

La función de distribución de velocidades no debe depender de la dirección si no solo de la velocidad de las partículas. Se puede decir entonces que una fracción de velocidades que se encuentra dentro de un rango (*fig. 4.2*) es igual a la velocidad total dentro de un contenedor multiplicada por una función fraccionaria desconocida eq. (4.26).

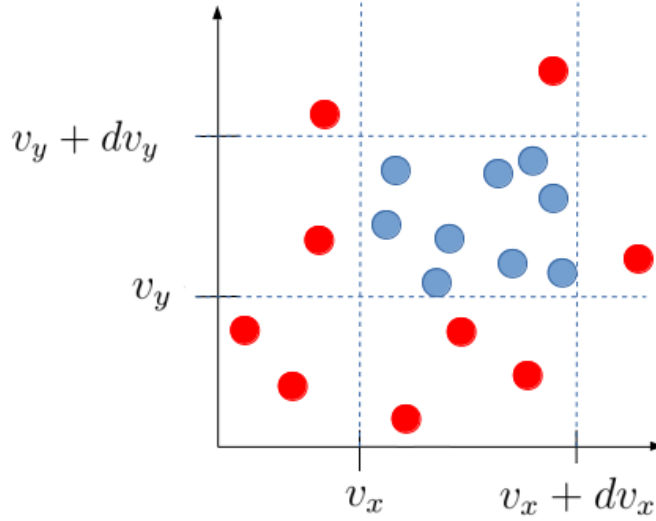


Figura 4.2: Solo una fracción  $f(v_x)f(v_y)$  de velocidades se encuentra en azul mientras que el resto  $N$  es el conjunto total de velocidades (puntos rojos y azules).

Recordando que la velocidad total en 3D es  $v^2 = (v_x^2 + v_y^2 + v_z^2)$ , tenemos que:

$$f(v_x)f(v_y)f(v_z) = \Omega(v_x^2 + v_y^2 + v_z^2) \quad (4.25)$$

$$f(v_x)f(v_y)f(v_z) = \Omega v^2 \quad (4.26)$$

El valor de la función de velocidades debe ser positivo y debe encontrarse entre 0 y 1, por lo tanto se utiliza el termino de velocidad elevado al cuadrado  $v^2$ .

En donde  $\Omega$  es entonces una función desconocida, sin embargo sabemos que una función exponencial o logarítmica decreciente podría satisfacer el resultado de una función de distribución de velocidades ya que esta va de 0 a 1 y para ser positiva su exponente debe estar elevado al cuadrado, por lo tanto podemos decir que la función de distribución de velocidades para una dimensión  $x$  puede utilizar la siguiente forma logarítmica:

$$f(v_x) = Ae^{-B(v_x)^2} \quad (4.27)$$

En donde A y B son constantes.

Esto resulta conveniente ya que evitamos encontrar un valor de una función desconocida  $\Omega$ .

Si aplicamos la definición de la eq. (4.27) para  $f(v_x)f(v_y)f(v_z)$  podemos sustituir el término del lado derecho de la ecuación (4.27) con lo siguiente:

$$f(v_x)f(v_y)f(v_z) = Ae^{-B(v_x)^2}Ae^{-B(v_y)^2}Ae^{-B(v_z)^2} \quad (4.28)$$

En donde según la propiedad de los logaritmos  $e^C e^D = e^{C+D}$  podemos decir que:

$$f(v_x)f(v_y)f(v_z) = A^3 e^{-B(v_x^2+v_y^2+v_z^2)} \quad (4.29)$$

Recordando de las ecuaciones 4.26 y 4.27 en donde  $v^2 = v_x^2 + v_y^2 + v_z^2$  obtenemos que:

$$f(v) = A^3 e^{-B(v^2)} \quad (4.30)$$

La ecuación (4.30) describe la función de probabilidad de distribución de velocidades para una partícula que cumple con una velocidad en un intervalo de  $v$  a  $v + dv$ . Esta función es la respuesta a la pregunta ¿Que probabilidad existe de encontrar una partícula estudiada cumpliendo con un cierto rango de velocidades en un instante de tiempo dado? sin embargo necesitamos saber la probabilidad de encontrar que varias partículas cumplan con un intervalo dado de velocidades.

#### 4.2.1. El espacio de velocidades tri-dimensional

Es importante pensar en la distribución de velocidades como un espacio tri-dimensional con valores de velocidad  $(v_x, v_y, v_z)$ , en donde cada velocidad es representada por un punto con coordenadas correspondientes a sus componentes en las direcciones  $x, y, z$ .

Si consideramos un vector velocidad resultante de las tres componentes anclado al origen y variamos su sentido y dirección mas no su magnitud rotandolo infinitas veces hasta ocupar todo el espacio tri-dimensional posible, estaremos dibujando una esfera, la cual contiene diferentes valores de componentes de velocidad en  $x, y$  y  $z$ , es decir, diferentes posiciones del espacio de velocidades pero que cumplen con un mismo vector velocidad  $v$ . Si a este espacio esférico le asignamos un rango  $v$  a  $v + dv$ , las partículas que cumplan con este intervalo de velocidades ya no serán las contenidas en la esfera completa, si no solo en una parte de ella, las partículas estarían contenidas entre la esfera con radio  $v$  y la esfera con radio  $v + dv$  fig. 4.3.

El volumen de este espacio entre esferas (*zona azul de la figura 4.3*) esta dado por  $4\pi v^2 dv$ , asumiendo que cada partícula es un punto en el espacio entre esferas podemos decir que el número por el cual debemos multiplicar el término derecho de la ecuación (4.30) para obtener la probabilidad de encontrar varias partículas con un rango de velocidades dado es dicho volumen.

Por lo tanto:

$$f(v_s) = 4\pi v^2 dv A^3 e^{-B(v^2)} \quad (4.31)$$

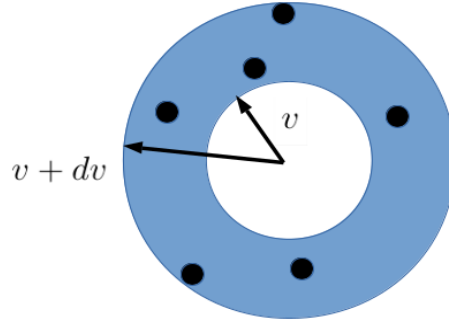


Figura 4.3: Si imaginamos una esfera cortada por la mitad podríamos observar las velocidades (puntos negros) que cumplen con un rango dado en un espacio de velocidades tri-dimensional.

En donde las constantes A y B son respectivamente:

$$A = \sqrt{\frac{b}{\pi}} \quad (4.32)$$

$$B = \frac{m}{2kT} \quad (4.33)$$

Debemos ser observadores en como el uso de la temperatura según lo visto en la sección anterior se incluye en la constante B.

Sustituyendo ambas constantes en la ecuación (4.31) y definiendo los limites de velocidad entre  $v$  y  $v + dv$

$$f(v) = \int_v^{v+dv} 4\pi v^2 \left( \frac{m}{2kT\pi} \right)^{\frac{3}{2}} e^{\frac{-mv^2}{2kT}} \quad (4.34)$$

La ecuación (4.34) finalmente nos arroja como resultado la probabilidad de encontrar partículas con cierta velocidad entre  $v$  y  $v + dv$  en un espacio tri-dimensional.

### 4.3. Función de distribución de probabilidad de velocidades de Boltzmann

Boltzmann demostró que la entropía  $S$  de un sistema con una energía  $E$  dada, esta relacionada con el número de micro-estados  $W$  disponibles mediante la ecuación (4.35).

$$S = k \log(W) \quad (4.35)$$

En donde  $k$  es la constante de Boltzmann.

Se considera que un sistema macroscópico puede encontrarse en ciertos micro estados, los cuales varían respecto al tiempo, es decir el macro-estado cambia según los posibles micro

estados que pueda adquirir. Los micro estados son aquellos posibles arreglos en los que puede encontrarse a un macro estado *fig. 4.4*.

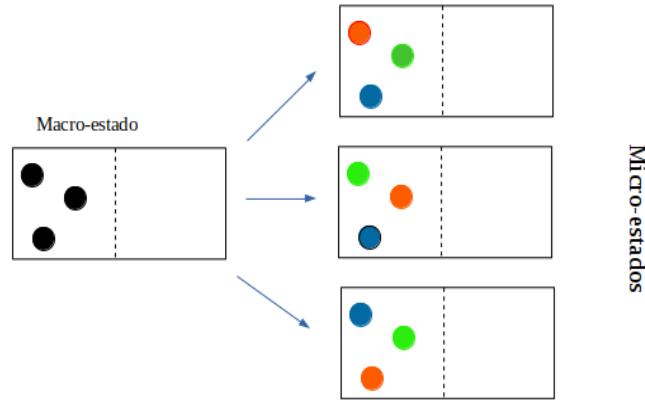


Figura 4.4: El macro-estado es una caja con partículas en la mitad de la izquierda y nada en la mitad derecha, las partículas pueden estar acomodadas en diferentes configuraciones según el paso del tiempo, a estas diferentes configuraciones en las que puede encontrarse a un macro-estado se les conoce como micro-estados.

Boltzmann entonces estableció que para cualquier sistema largo o pequeño en equilibrio térmico a una temperatura  $T$ , la probabilidad de encontrarse en un estado en particular con una energía  $E$  es proporcional a  $e^{\frac{-E}{kT}}$  por lo tanto:

$$f(E) = Ae^{\frac{-E}{kT}} \quad (4.36)$$

En donde  $A$  esta definida por la siguiente ecuación (4.37)

$$A = \sqrt{\frac{m}{2\pi kT}} \quad (4.37)$$

Sustituyendo la ec. (4.37) y siendo  $E = \frac{1}{2}mv^2$  tenemos que para obtener la probabilidad de encontrar una partícula en un cierto rango de velocidad en dirección  $x$  utilizamos la siguiente función:

$$f(v_x) = \sqrt{\frac{m}{2\pi kT}} e^{\frac{-mv_x^2}{2kT}} \quad (4.38)$$

Planteando la ecuación (4.38) para obtener la probabilidad de encontrar partículas con una velocidad sin importar la dirección de esta tenemos:

$$f(v) = \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} e^{\frac{-mv^2}{2kT}} \quad (4.39)$$

Multiplicando la ecuación (4.39) por el volumen esférico descrito anteriormente se tiene que:

$$f(v) = \int_v^{v+dv} 4\pi v^2 \left( \frac{m}{2\pi kT} \right)^{\frac{3}{2}} e^{\frac{-mv^2}{2kT}} \quad (4.40)$$

Finalmente la (4.40) es la ecuación de la distribución de maxwell-boltzmann, a partir de la cual se obtiene **la ecuación de boltzmann** que es la base del método de lattice boltzmann.

## 4.4. Ecuación de Boltzmann

### 4.4.1. Ecuación de transporte de Boltzmann

Una descripción estadística de un sistema se puede explicar mediante la función de distribución  $f(r, v, t)$  en donde  $f(r, v, t)$  es el número de moléculas en un tiempo  $t$ , con una posición  $r$  y una velocidad  $v$ .

Si se aplica una fuerza externa  $F$  al sistema como un pistón empujando un gas de masa  $m$  en un volumen de control, cambiará la velocidad de las partículas de  $v$  a  $v + Fdt$  y la posición de esta de  $r$  a  $r + vdt$ .

El número de moléculas empujadas por el pistón en el volumen de control debe ser el mismo en el intervalo  $r$  a  $r + vdt$  si no existen colisiones entre las partículas.

Por lo tanto el número de moléculas antes y después de aplicarse la perturbación es el mismo.

Esto da lugar a la siguiente ecuación:

$$f(r, v, t) = f(r + vdt, v + Fdt, t + dt) \quad (4.41)$$

$$f(r, v, t) - f(r + vdt, v + Fdt, t + dt) = 0 \quad (4.42)$$

Al existir colisiones entre ellas la fuerza externa aplicada no sería la única fuerza actuando sobre las partículas y la dirección y velocidad cambiaría para cada una de ellas como consecuencia de las colisiones. Esto significa que algunas podrían salir del volumen de control en el intervalo de  $r$  a  $r + vdt$  y entonces la ecuación (4.42) no es igual a cero.

En la ecuación el resultado es el número de partículas inicial multiplicado por una función desconocida que da como resultado un incremento o un decremento en el número de moléculas, mostrando así un cambio en la cantidad de partículas antes y después de haber sido perturbadas.

Esta función desconocida es conocida como el operador de colisión  $\Omega$ .

La ecuación de transporte de boltzmann se convierte en:

$$f(r, v, t) - f(r + vdt, v + Fdt, t + dt) = \Omega f(r, v, t) \quad (4.43)$$

Desde un punto de vista infinitesimal tenemos:

$$f(r, v, t)drdv - f(r + vdt, v + Fdt, t + dt)drdv = \Omega f(r, v, t)drdvdt \quad (4.44)$$



Dividiendo ambos lados de la ecuación entre  $drdvdt$

$$\frac{f(r, v, t) - f(r + vdt, v + Fdt, t + dt)}{dt} = \Omega f(r, v, t) \quad (4.45)$$

Se puede expresar la suma de la función de distribución  $f(r, v, t) - f(r + vdt, v + Fdt, t + dt)$  como una diferencia infinitesimal. Por lo tanto:

$$\Omega(f) = \frac{df}{dt} \quad (4.46)$$

Ya que  $f$  esta en función de  $r, v$  y  $t$  entonces  $df$  se puede expandir como:

$$df = \frac{\partial f}{\partial r} dr + \frac{\partial f}{\partial v} dv + \frac{\partial f}{\partial t} dt \quad (4.47)$$

$$\frac{df}{dt} = \frac{\partial f}{\partial r} \frac{dr}{dt} + \frac{\partial f}{\partial v} \frac{dv}{dt} + \frac{\partial f}{\partial t} \quad (4.48)$$

En donde:

$$\frac{dr}{dt} = v$$

$$\frac{dv}{dt} = \frac{F}{m} = a$$

Por lo tanto la ecuación (4.48) se puede escribir de la siguiente forma:

$$\Omega(f) = \frac{\partial f}{\partial r} v + \frac{\partial f}{\partial v} a + \frac{\partial f}{\partial t} \quad (4.49)$$

o también como:

$$\Omega(f) = \frac{\partial f}{\partial r} v + \frac{\partial f}{\partial v} \left( \frac{F}{m} \right) + \frac{\partial f}{\partial t} \quad (4.50)$$

Para un sistema sin una fuerza externa. la ecuación de Boltzmann es:

$$\Omega(f) = \frac{\partial f}{\partial r} v + \frac{\partial f}{\partial t} \quad (4.51)$$

## 4.5. Aproximación BGKW

La ecuación de boltzmann (4.51) es una ecuación integro-diferencial cuya solución es muy complicada y de alto costo computacional. Sin embargo se encontró un enfoque distinto para obtener esta solución, la cual es la aproximación BGKW.

El operador de colisión  $\Omega$  fue reemplazado por:

$$\Omega(f) = \omega(f^{eq} - f) \quad (4.52)$$

En donde  $\omega$  es la frecuencia de colisión  $\omega = \frac{1}{\tau}$  y  $\tau$  es un factor de relajación.

Por otro lado,  $f^{eq}$  es la función de distribución de equilibrio local que es la función de distribución de maxwell-boltzmann ec (4.40) y por último  $f$  es el número de moléculas en un instante de tiempo  $t$  con una posición  $r$  y una velocidad  $v$ .

Sustituyendo la ecuación (4.52) en la ecuación (4.51) tenemos que la ecuación de boltzmann para un sistema sin una fuerza externa es:

$$\frac{\partial f}{\partial r}v + \frac{\partial f}{\partial t} = \frac{1}{\tau}(f^{eq} - f) \quad (4.53)$$

La ecuación (4.53) es la herramienta de trabajo del método de lattice boltzmann para simulaciones de CFD y puede ser discretizada como:

$$f_i(r + v_i\Delta t, t + \Delta t) = f_i(r, t) + \frac{\Delta t}{\tau} [f_i^{eq}(r, t) - f_i(r, t)] \quad (4.54)$$

En donde  $i$  se refiere a una dirección específica en los enlaces de un arreglo ó enrejado, conocido en inglés como *lattice*, en cada arreglo residen partículas ficticias descritas con una función de distribución. Algunas de estas partículas se mueven a lo largo de direcciones específicas hacia los nodos adyacentes considerando un dominio creado por varios enrejados. El número de direcciones y enlaces depende del arreglo que se este utilizando.

## 4.6. Arreglos lattice

La terminología usada en lattice boltzmann para referirnos a los diferentes tipos de arreglos existentes es:

$$DnQm$$

En donde  $n$  representa la dimensión del problema (1D,2D,3D) y  $m$  indica el número de velocidades en el modelo, también es el número de enlaces.

### 4.6.1. Arreglo tri-dimensional

En general dos modelos son utilizados en simulaciones de tres dimensiones, el  $D3Q15$  y  $D3Q19$ .

En el modelo  $D3Q15$  existen 15 enlaces que funcionan como vectores de velocidad cada uno caracterizado por una función de distribución de probabilidad  $f$ .

En la fig 4.5 se puede apreciar lo siguiente:

La función  $f_0$  tiene velocidad cero, los nodos 1,2,3 y 4 están en el centro de las caras este, norte, oeste y sur respectivamente. Los nodos 5 y 6 se encuentran al frente y atrás. Los nodos 7,8,9,10,11,12,13 y 14 están en las esquinas del enrejado. Los 15 vectores velocidad para las funciones de distribución son los siguientes:

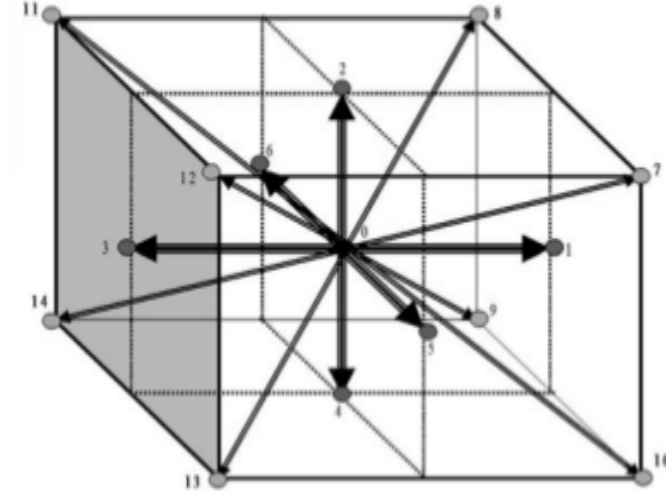


Figura 4.5: Representación gráfica del arreglo D3Q15 utilizado en LBM.

$$f_0 \longrightarrow v(0, 0, 0)$$

$$f_1 \longrightarrow v(1, 0, 0)$$

$$f_2 \longrightarrow v(0, 1, 0)$$

$$f_3 \longrightarrow v(-1, 0, 0)$$

$$f_4 \longrightarrow v(0, -1, 0)$$

$$f_5 \longrightarrow v(0, 0, 1)$$

$$f_6 \longrightarrow v(0, 0, -1)$$

$$f_7 \longrightarrow v(1, 1, 1)$$

$$f_8 \longrightarrow v(1, 1, -1)$$

$$f_9 \longrightarrow v(1, -1, -1)$$

$$f_{10} \longrightarrow v(1, -1, 1)$$

$$f_{11} \longrightarrow v(-1, 1, -1)$$

$$f_{12} \longrightarrow v(-1, 1, 1)$$

$$f_{13} \longrightarrow v(-1, -1, 1)$$

$$f_{14} \longrightarrow v(-1, -1, -1)$$

Cada función de distribución tiene un factor de ponderación que corresponde a su distancia relativa con el centro del enrejado. Para el modelo  $D3Q15$  esos factores son:

$$f_0 \longrightarrow \frac{16}{72}$$

$$f_1 \text{ a } f_6 \longrightarrow \frac{8}{72}$$

$$f_7 \text{ a } f_{14} \longrightarrow \frac{1}{72}$$

Para el modelo  $D3Q19$  (*fig.4.6*) los vectores son los siguientes:

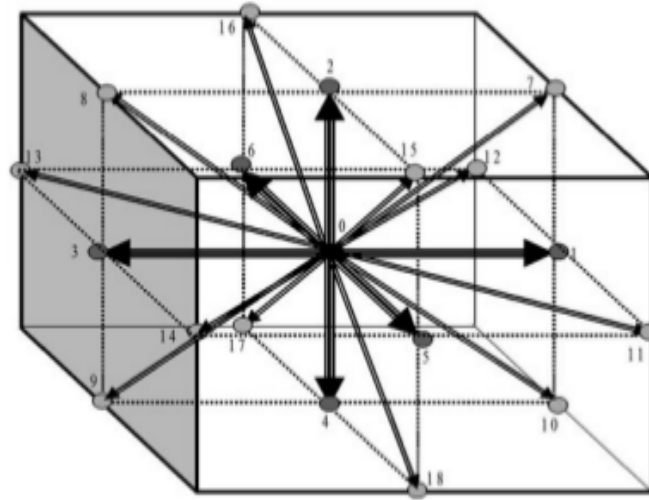


Figura 4.6: Representación gráfica del arreglo D3Q19 utilizado en LBM.

$$f_0 \longrightarrow v(0, 0, 0)$$

$$f_1 \longrightarrow v(1, 0, 0)$$

$$f_2 \longrightarrow v(0, 1, 0)$$

$$f_3 \longrightarrow v(-1, 0, 0)$$

$$f_4 \longrightarrow v(0, -1, 0)$$

$$f_5 \longrightarrow v(0, 0, 1)$$

$$f_6 \longrightarrow v(0, 0, -1)$$

$$f_7 \longrightarrow v(1, 1, 0)$$

$$f_8 \longrightarrow v(-1, 1, 0)$$

$$f_9 \longrightarrow v(-1, -1, 0)$$

$$f_{10} \longrightarrow v(1, -1, 0)$$

$$f_{11} \longrightarrow v(1, 0, 1)$$

$$f_{12} \longrightarrow v(1, 0, -1)$$

$$f_{13} \longrightarrow v(-1, 0, -1)$$

$$f_{14} \longrightarrow v(-1, 0, 1)$$

$$f_{15} \longrightarrow v(0, 1, 1)$$

$$f_{16} \longrightarrow v(0, 1, -1)$$

$$f_{17} \longrightarrow v(0, -1, -1)$$

$$f_{18} \longrightarrow v(0, -1, 1)$$

Con sus respectivos factores de ponderación.

$$f_0 \longrightarrow \frac{12}{36}$$

$$f_1 \text{ a } f_6 \longrightarrow \frac{2}{36}$$

$$f_7 \text{ a } f_{18} \longrightarrow \frac{1}{36}$$

## Sección III

# Simulaciones de Navier-Stokes y Lattice Boltzmann

# Capítulo 5

## Caso de Estudio y condiciones

### 5.1. Caso del escalón (forward-facing step)

El flujo sobre un escalón al frente (*forward-facing step*) sumergido en una capa límite turbulenta, produce estructuras de flujo complejas que permanecen en gran medida sin explicación. Es una configuración canónica que tiene importancia tanto para la investigación de la dinámica de fluidos fundamental como para el diseño de ingeniería aplicada, por ejemplo, difusores, intercambiadores de calor y cámaras de combustión (Iftekhhar et al., 2015). Igualmente, esta geometría se puede generar a partir de la instalación de paneles en el borde de ataque de alas, barquillas y superficies de control, así como de la instalación de paneles de acceso, puertas y ventanas en el fuselaje de una aeronave (Costantini et al., 2015).

No obstante, es más común encontrar investigaciones acerca del escalón atrás (*backward-facing step*) ya que en el caso al frente existen dos regiones de recirculación. Asimismo, los investigadores enfrentan dificultades en la obtención de datos precisos en la región de recirculación, además de que el comportamiento del flujo a un parámetro de flujo individual depende de numerosos parámetros (Sherry et al., 2009). En esta ocasión, nos referiremos al caso del escalón al frente simplemente como el caso general del escalón.

El flujo sobre el escalón tiene diversas características incluyendo puntos de separación y readherencia, regiones de recirculación (vórtices) y capas límite que se ilustran en la Figura **XX**. La capa límite turbulenta se acerca al obstáculo de izquierda a derecha y a medida que aumenta el bloqueo del escalón hacia arriba, se forma un gradiente de presión adverso. El flujo se separa a aguas arriba y se vuelve a adherir a la pared del escalón.

- 5.2. Condiciones generales para la simulación utilizando las ecuaciones de Navier-Stokes para un fluido tridimensional bajo efectos turbulentos
- 5.3. Condiciones generales para la simulación utilizando el método de lattice Boltzmann para un fluido tridimensional bajo efectos turbulentos



# Capítulo 6

## Desarrollo del código

### 6.1. Programación del método de lattice boltzmann para un fluido turbulento tri-dimensional

#### 6.1.1. Flujo incompresible e isotérmico

La ecuación de lattice boltzmann para un fenómeno descrito por el problema de transporte originalmente es no lineal. Mediante la aproximación BGK el término de advección es reemplazado por un paso de "streaming."° dicho en otras palabras por un paso en el cual las partículas se mueven de un nodo a otro. No hay necesidad de resolver la ecuación de Poisson en cada time step para satisfacer la ecuación de continuidad, esto reduce drásticamente el tiempo computacional comparado con el método numérico de Navier-Stokes.

LBM puede ser considerado un método explícito en donde no es necesario resolver ecuaciones simultáneas en cada paso de tiempo. Los pasos de colisión y movimiento de las partículas (streaming) son locales y es fácil resolver la simulación en diferentes CPUs en un proceso paralelo.

LBM simula flujo incompresible por debajo de un Mach pequeño y con variaciones de densidad débiles. En LBM el número Mach está definido por:

$$Ma = \frac{u}{c_s} \quad (6.1)$$

En donde  $u$  es la velocidad macroscópica del dominio y  $c_s$  la velocidad del sonido.

El número de Reynolds  $Re$  está definido por:

$$Re_{LBM} = \frac{uN}{\nu} \quad (6.2)$$

En donde  $u$  es la velocidad mesoscópica (LBM) del dominio,  $N$  el número de lattices y  $\nu$  la viscosidad cinemática cuyo valor puede ser elegido arbitrariamente. Se aconseja que para altos valores de  $Re$  se incremente el número de lattices o se utilicen valores pequeños de  $\nu$ , el uso de valores inferiores de  $\nu$  puede causar problemas de estabilidad. Para un mayor control de las propiedades del flujo se recomienda usar la ecuación 6.4, en la que se usa la viscosidad dinámica y no la viscosidad cinemática.

$$\nu = \frac{\mu}{\rho} \quad (6.3)$$

Por lo tanto:

$$Re_{LBM} = \frac{uN\rho}{\mu} \quad (6.4)$$

#### 6.1.1.1. Arreglo D2Q9

Para lograr desarrollar el código para flujo tri-dimensional turbulento debemos dominar el código para flujo bi-dimensional sin turbulencia.

En el método de lattice boltzmann el flujo en 2D se desarrolla con el arreglo D2Q9.

Recordemos la ecuación BGK-LBM:

$$f_k(x + \Delta x, t + \Delta t) = f_k(x, t)[1 - \omega] + \omega f_k^{eq}(x, t) \quad (6.5)$$

en donde:

$$f_k^{eq} = w_k \rho(x, t) \left[ 1 + \frac{c_k \cdot U}{c_s^2} + \frac{1}{2} \frac{(c_k \cdot U^2)}{c_s^4} - \frac{1}{2} \frac{u^2}{c_s^2} \right] \quad (6.6)$$

por lo tanto:

$$f_k(x + \Delta x, t + \Delta t) = f_k(x, t)[1 - \omega] + \omega w_k \rho(x, t) \left[ 1 + \frac{c_k \cdot U}{c_s^2} + \frac{1}{2} \frac{(c_k \cdot U^2)}{c_s^4} - \frac{1}{2} \frac{u^2}{c_s^2} \right] \quad (6.7)$$

en donde:

$$c_s = \frac{c_k}{\sqrt{3}} \quad (6.8)$$

$$c_k = \frac{\Delta x}{\Delta t} i + \frac{\Delta y}{\Delta t} j \quad (6.9)$$

$$U = ui + vj \quad (6.10)$$

La ecuación 6.7 es la ecuación LBM-BGK con términos extras en la función de equilibrio que son necesarios para la ecuación de conservación de momentum y condición de no linealidad de la velocidad.

También es necesario usar el arreglo D2Q9 para asegurar la isotropía e invariancia galileana.

Los parámetros que controlan el flujo incompresible serán el número de Reynolds y las restricciones geométricas del problema. Entonces, para cualquier problema de flujo incompresible debemos tener un Re adecuado a la geometría que se utilizará. (similitud de geometría)

### 6.1.1.2. Número de Mach y Reynolds

El análisis del error en LBM necesita de una extensiva expansión multi-escala que va mas allá de los límites de esta tesis.

Dicho análisis demuestra que LBM se asemeja al uso de las ecuaciones de Navier-Stokes para flujo incompresible a un número de Mach bajo. Al utilizar aire usualmente se usa un  $Ma < 0.33$ . Un valor mas alto presentaría el enrarecimiento del aire y con ello habría discontinuidades que podrían afectar los resultados de la simulación.

El error en LBM es del orden de  $Ma^2$ . La viscosidad cinemática esta relacionada con el factor de relajación  $\omega$  mediante la ecuación 6.11:

$$\nu = \frac{\Delta x^2}{3\Delta t}(\omega - 0.5) \quad (6.11)$$

Utilizando  $Re = \frac{UL}{\nu}$  y reemplazando  $\nu$  por la ec. 6.11 tenemos que:

$$Re = \frac{3\Delta t UL}{\Delta x^2(\omega - 0.5)} \quad (6.12)$$

Para el numero de Mach tenemos que:

$$Ma = \frac{\Delta x}{L\sqrt{3}}(\omega - 0.5)Re \quad (6.13)$$

En aplicaciones prácticas el  $Re$  macroscópico se iguala a  $Re_{LBM}$ , después  $u$  y  $\nu$  pueden ser arbitrariamente seleccionados dentro del rango que asegure la estabilidad de la solución. En este caso  $N$  tendrá que cambiar según los valores de  $u_{LBM}$  y  $\nu$  así lo requieran.

### 6.1.1.3. Condiciones de Frontera

**1.-Condición de Rebote.** Es usada para modelar un cuerpo solido, en otras palabras esta condición de frontera representa una condición de no deslizamiento. En LBM, existen dos versiones para esta condición. La primera es simular el rebote a la mitad de la celda del enrejado, esto provoca una disminución de dominio que es igual a la mitad del tamaño de la celda en ese nodo *fig.6.1*

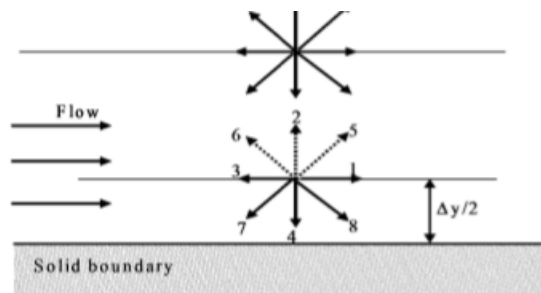


Figura 6.1: Representación de la condición de frontera de rebote sin tocar la pared realmente. Versión 1

La segunda versión es extender el dominio de manera que la pared este ubicada exactamente a la mitad de la celda. *fig. 6.2*

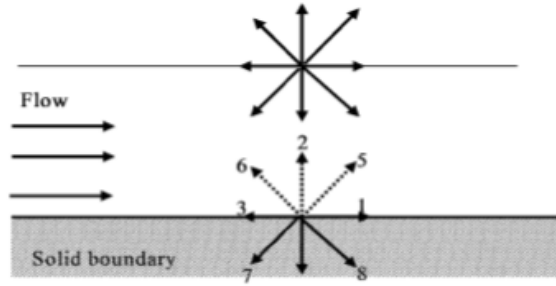


Figura 6.2: Representación de la condición de frontera de rebote con el dominio extendido mas allá de la pared. Versión 2

El concepto de la condición de rebote es simple, en las figuras anteriores  $f_4, f_7$  y  $f_8$  son funciones conocidas y sus valores se obtienen en el paso de streaming. Se asume que cuando una de estas funciones de distribución que representan partículas en movimiento, hagan contacto con la pared, regresarán al dominio sin salir de el, conservando el mismo ángulo y magnitud pero en dirección opuesta por lo tanto se puede expresar lo siguiente:

$$f_2 = f_4 \quad (6.14)$$

$$f_5 = f_7 \quad (6.15)$$

$$f_6 = f_8 \quad (6.16)$$

## 2.-Condición con velocidad conocida.

En general, la función de distribución puede ser dividida en dos partes, la función de equilibrio y la función de no equilibrio.

$$f = f^{eq} + f^{neq} \quad (6.17)$$

Para los nodos ubicados en una entrada del lado izquierdo de un ducto horizontal, la ecuación anterior se puede escribir de la siguiente manera:

$$f_{izq} = f_f + (f_{izq}^{eq} - f_f^{eq}) \quad (6.18)$$

Por ejemplo, en la *fig.6.2* las funciones de distribución  $f_1, f_5$ , y  $f_8$ , son desconocidas ya que no hay un dominio a la izquierda de la entrada de flujo de donde se puedan obtener valores por medio del paso de streaming.

Entonces:

$$f_1 = f_3 + (f_1^{eq} - f_3^{eq}) \quad (6.19)$$

$$f_5 = f_7 + (f_5^{eq} - f_7^{eq}) \quad (6.20)$$

$$f_8 = f_6 + (f_8^{eq} - f_6^{eq}) \quad (6.21)$$

### **3.-Condición de frontera abierta.**

En algunos problemas la velocidad de la salida del flujo no es posible determinarse. Normalmente las funciones de distribución desconocidas se obtienen por extrapolación.

En las figuras 6.1 y 6.2 la parte derecha de la imagen representa la salida de flujo. Las funciones  $f_1, f_5$  y  $f_8$  se obtienen por medio del paso de streaming, y las funciones desconocidas  $f_3, f_6$  y  $f_7$  se obtienen por medio de las siguientes ecuaciones:

$$f_{3,n} = 2f_{3,n-1} - f_{3,n-2} \quad (6.22)$$

$$f_{6,n} = 2f_{6,n-1} - f_{6,n-2} \quad (6.23)$$

$$f_{7,n} = 2f_{7,n-1} - f_{7,n-2} \quad (6.24)$$

Sin embargo, en algunos casos las ecuaciones anteriores terminan en soluciones inestables.

El uso de una interpolación de primer orden ha mostrado mayor estabilidad que los esquemas de interpolación de segundo orden según la bibliografía.

Otro enfoque es asumir que la densidad a la salida es conocida y se mantiene constante. Si esta es la condición entonces se pueden usar las siguientes ecuaciones:

$$u_x = -1 + \frac{f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)}{\rho_{salida}} \quad (6.25)$$

$$f_3 = f_1 - \frac{2}{3}\rho_{salida}u_x \quad (6.26)$$

$$f_6 = f_8 - \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_{salida} \quad (6.27)$$

$$f_7 = f_5 + \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_{salida} \quad (6.28)$$

#### 6.1.1.4. Planteamiento del problema en un arreglo D2Q9 y código

Para el problema general del escalón el planteamiento es el siguiente:

Se tiene aire a una velocidad macroscópica  $U$ , el cual fluye alrededor de un escalón de dimensiones  $x, y$ . Simulando un túnel de viento, en donde la sección de prueba restringe el flujo por medio de una pared superior en inferior.

Un segundo planteamiento es el estudio del mismo escalón en un volumen abierto con el escalón sobre un piso estático.

Se deben obtener los campos de velocidad y presión para comparar los métodos de LBM y NS con el fin de obtener las justas comparaciones y conclusiones.

Se simulará el caso del escalón en un volumen abierto como primera aproximación.

El dominio es rectangular y cuenta con tres fronteras abiertas. El escalón esta situado sobre una pared inferior (suelo).

En esta primera aproximación se deben utilizar las condiciones de frontera de la siguiente manera:

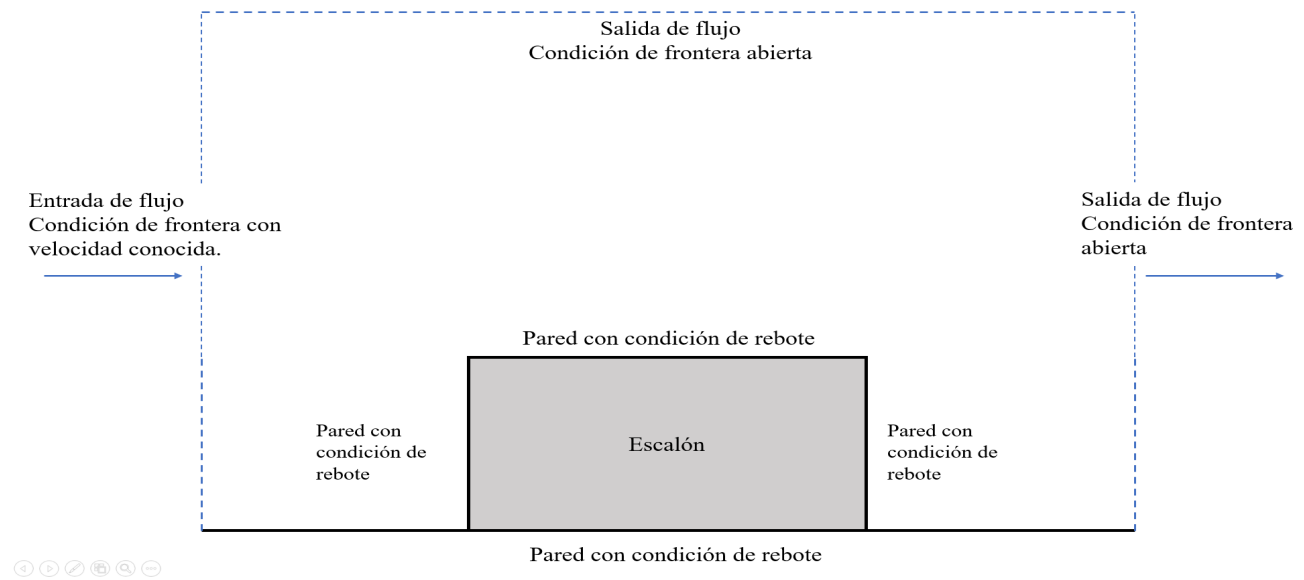


Figura 6.3: Diagrama de simulación para un escalón situado en un volumen abierto

La segunda aproximación es el planteamiento del escalón en una sección de prueba de un túnel de viento. El dominio es rectangular y tiene una entrada de flujo, una salida y dos paredes que restringen y estrangulan el flujo. Las condiciones de frontera se plantean de la siguiente manera:

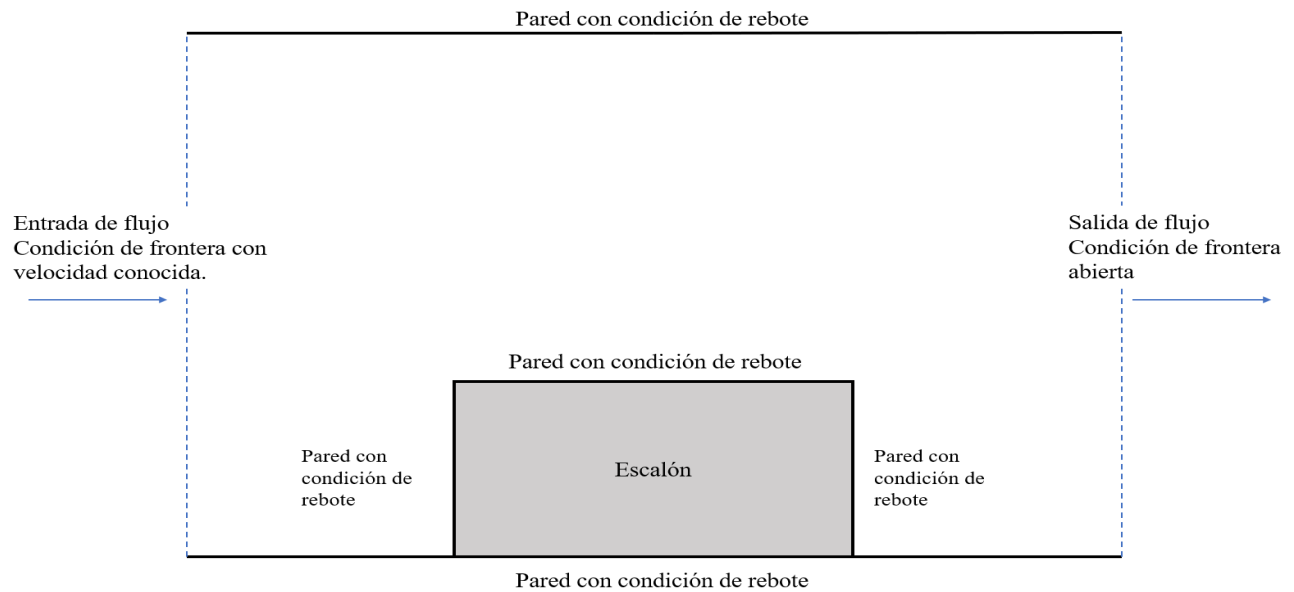
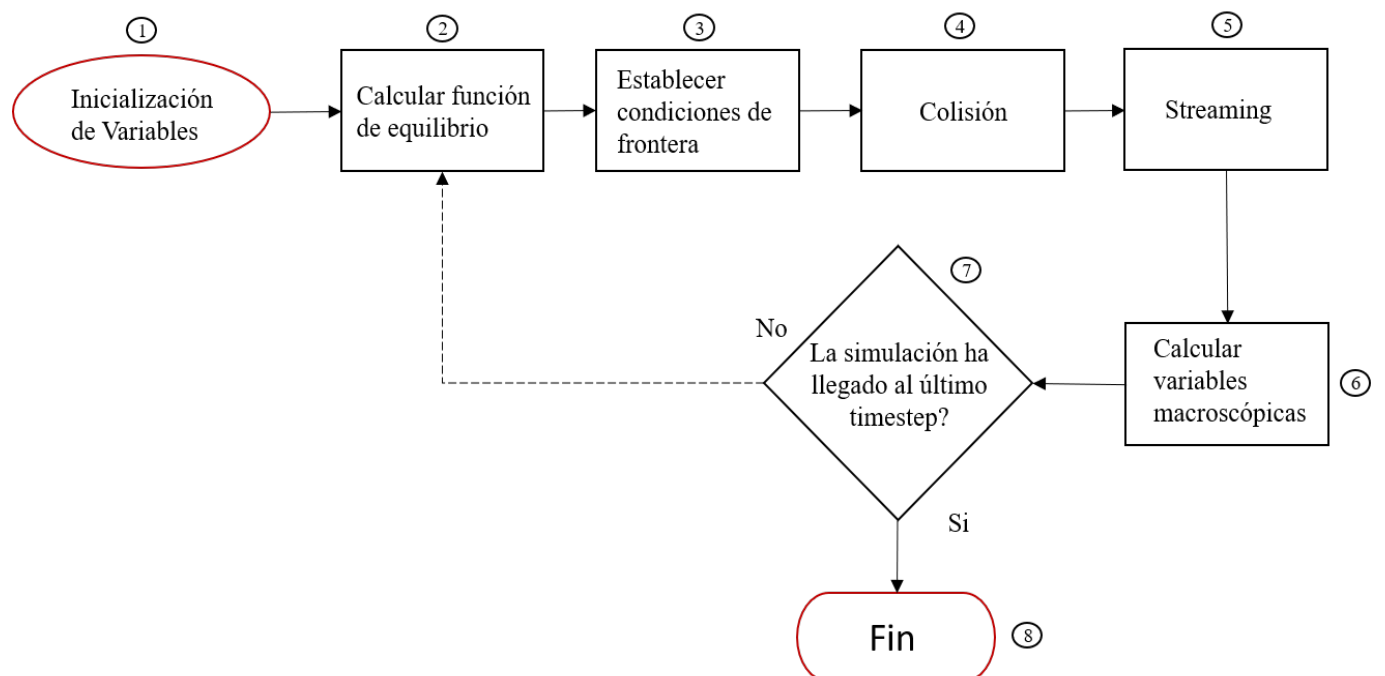


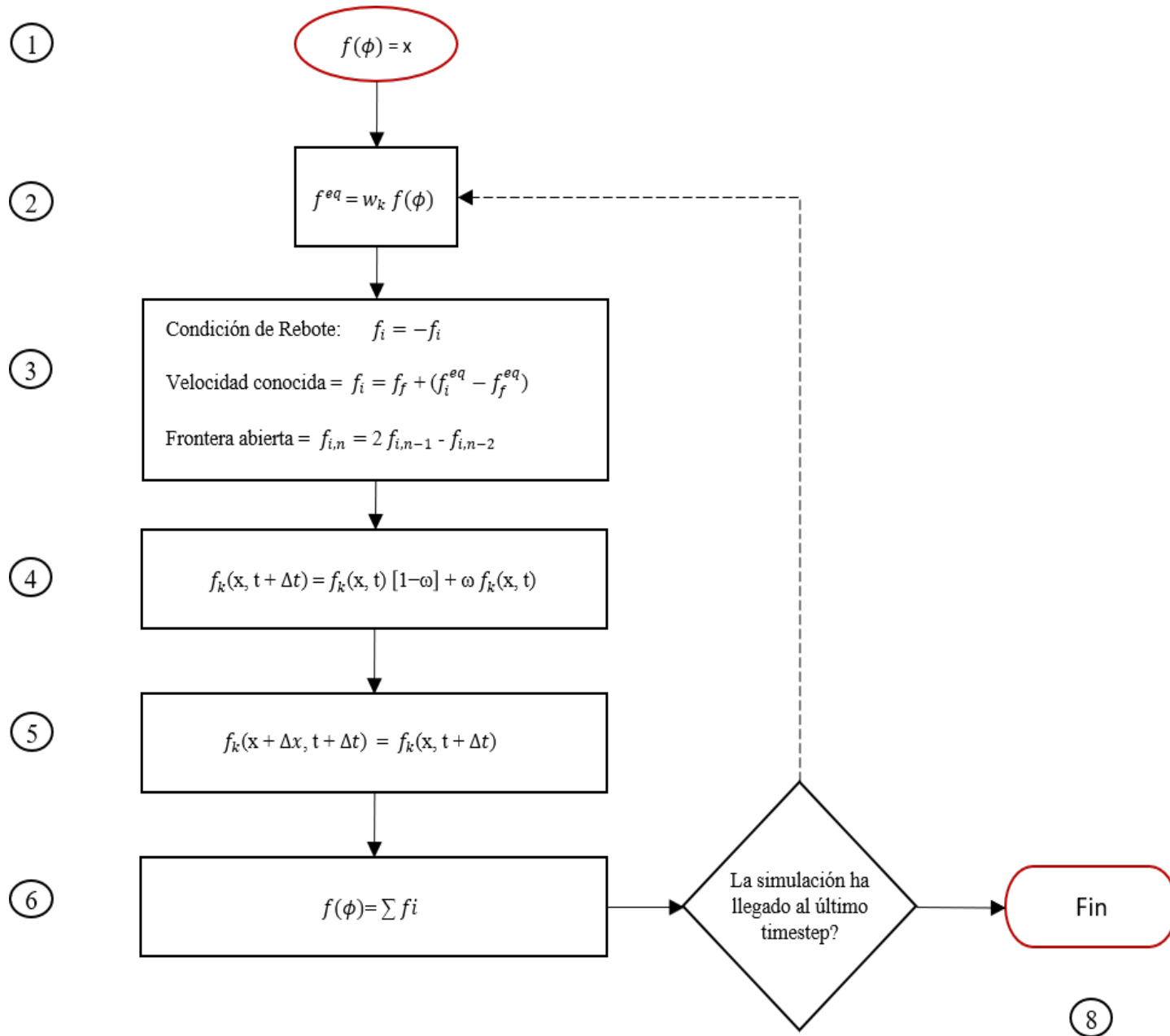
Figura 6.4: Diagrama de simulación para un escalón en un túnel de viento

El algoritmo general de cualquier código de LBM es el siguiente:



Observando el diagrama anterior podemos asegurar que el método de Lattice Boltzmann consiste en el cumplimiento satisfactorio de cinco pasos.

El diagrama anterior puede ser descrito con las ecuaciones del método de Lattice Boltzmann de la siguiente manera:

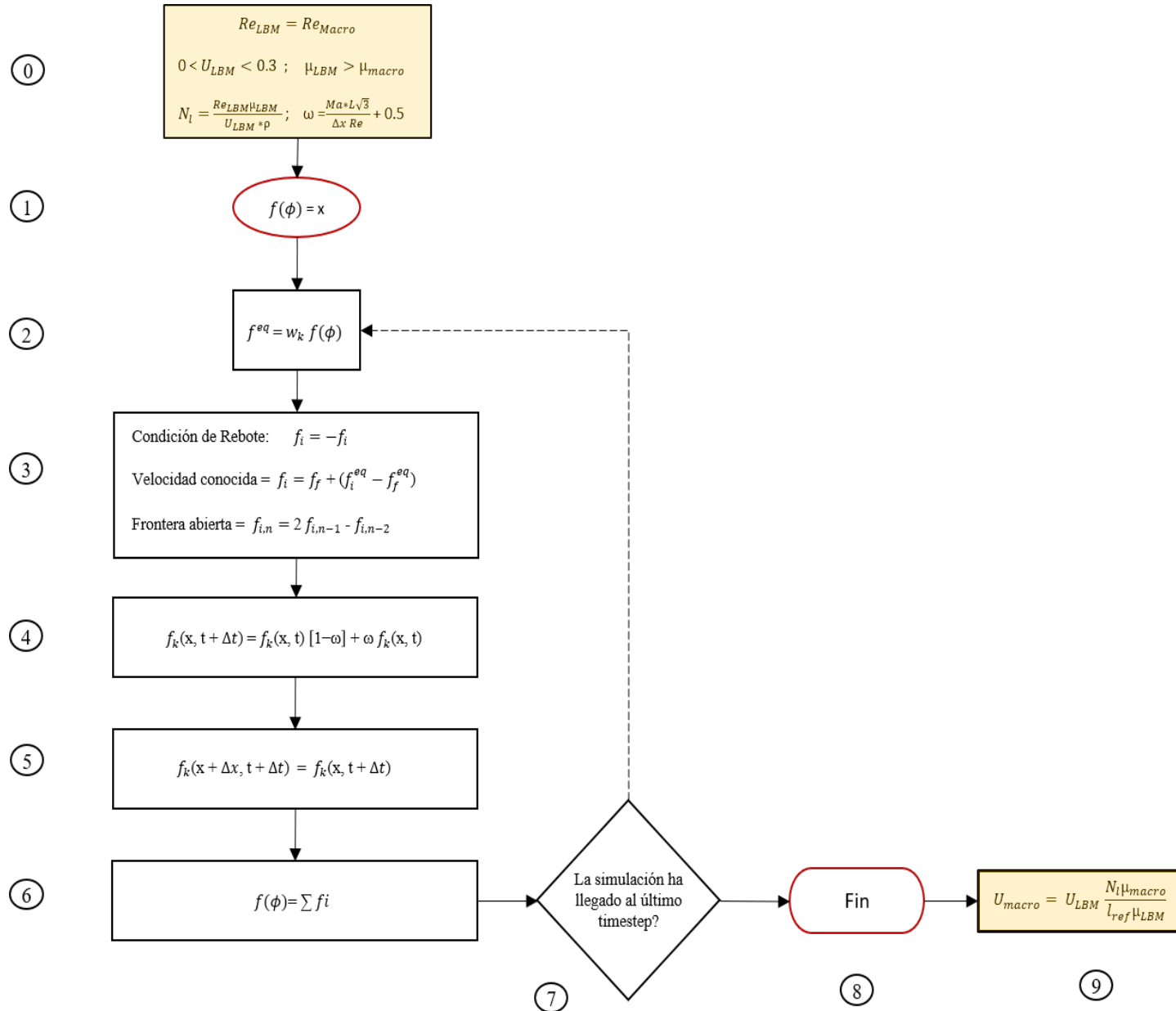


El proceso es descrito en 8 pasos, en el caso particular del escalón se usaran tres tipos de condiciones de frontera, las cuales están descritas en el paso 4.

Desglosaremos el diagrama anterior con el fin de permitir un mayor entendimiento de la programación del código del método de lattice boltzmann para un escalón con un arreglo D2Q9 de enrejados o lattices.



Recordemos que el método de lattice boltzmann trabaja en un régimen de flujo con  $Ma < 0.33$ . El valor de  $Ma$  entonces, puede restringir el estudio del flujo. Sin embargo, se puede cambiar de una escala macroscópica a una mesoscópica y viceversa para obtener los resultados deseados, concentrando el estudio en el uso del número de  $Re$  de manera que no haya un estado enrarecido del aire que genere inestabilidades pero el régimen de flujo sea el requerido en una escala macroscópica. Se debe entonces agregar dos pasos al diagrama de LBM. Un paso 0 y un paso 9, los cuales indicarán la transformación de escala de un régimen macroscópico a un régimen estable mesoscópico de LBM mediante el uso de la definición del número de  $Re$  para LBM. Estos dos pasos se explicarán con detalle mas adelante.



### Paso 0. Macro-escala a Meso-escala

El paso 0 es el mas sencillo. Solo se deben asignar las variables que registrarán el flujo, sus valores y la transformación de escalas de un enfoque macroscópico a un enfoque mesoscópico mediante el Re.

---

```

1  import os ; from os import * ; import shutil
2  from numpy import *; from numpy.linalg import *
3  import matplotlib.pyplot as plt; from matplotlib import cm; import cv2
4  import numpy as np ; import glob
5
6  maxIter = 20000                                # Número total de timesteps.
7  frame = 200 ;
8
9  ##### Definición de variables #####
10 rho = 1.22                                     # Densidad
11 lenx, leny = 3, 1                             # Dimensiones del dominio en "x" y "y"
12 radius,U,vis = 0.4,5,1e-5                     # Definición de un obstáculo (escalón)
13
14 ##### Transformación a mesoescala #####
15 lref=leny                                     # Longitud de Referencia para Re
16 Re = lref*rho*U/vis                           # Número Re macroscópico
17 uLB = 0.1; vis_lbm = 1e-4 # Velocidad y viscosidad dinámica mesoscópica
18 nx = int(Re*vis_lbm/(uLB*rho)) # Obtención Nodos en x con Re macroscópico.
19 ny = int(leny*nx/lenx)
20 ly=ny-1.0; q = 9                             # "q" número de funciones de velocidad
21 dx = lenx/nx ; dt = dx                       # dx en escala mesoscópica.
22 cx = nx/2; cy=ny; Ma = 0.22                  # Mach elegido arbitrariamente
23 Ma_err = Ma**2
24 omega = (Ma/((dx/(nx*((1/3)**2)))*Re))+0.5   # Factor de relajación omega
25
26 print("\nOmega: "+str(omega)+"\n"+"Ma: "+str(Ma)+"\n"+"Ma_err= "+str(Ma_err)
27 +" "+str(Ma_err*100)+"perc"+ "\n"+"nx: "+ "\n"+str(nx)+"\n"+"ny: "+str(ny))
28 os.system("pause")

```

---

Para calcular el Re a escala macroscópica se usa la ecuación 6.29 y para calcular el Re a escala mesoscópica se usa la ecuación 6.30 de la siguiente página.

$$Re_{macro} = \frac{U_{macro} * l_{ref} * \rho}{\mu_{macro}} \quad (6.29)$$

$$Re_{LBM} = \frac{U_{LBM} * N_l * \rho}{\mu_{LBM}} \quad (6.30)$$

La velocidad  $u_{LBM}$  y  $\mu_{LBM}$  controlarán la estabilidad de la solución en LBM, mientras que  $N_l$  despejado en la ecuación 6.31 mantendrá el flujo en el mismo régimen permitiendo que  $Re_{macro}$  y  $Re_{LBM}$  sean iguales.

$$N_l = \frac{Re_{macro} * \mu_{LBM}}{U_{LBM} * \rho} \quad (6.31)$$

$$\omega = \frac{Ma * N_l * \sqrt{3}}{Re} + 0.5 \quad (6.32)$$

Para fines prácticos se definen las variables en escala macroscópica (líneas 10-12 del código anterior) y se transforma a la escala mesoscópica de la siguiente manera.

1.- Se igualan  $Re_{macro}$  y  $Re_{LBM}$ .

$$Re_{macro} = Re_{LBM}$$

2.- Se elige arbitrariamente un valor de la velocidad  $U_{LBM}$  mayor a 0 y menor a 1.  
Por ejemplo:

$$U_{LBM} = 0.1$$

$$N_l = \frac{Re_{macro} * \mu_{LBM}}{0.1 * \rho}$$

3.- Se elige un valor de viscosidad arbitrariamente mayor a  $\mu_{macro}$ .

Por ejemplo:

$$\mu_{macro} = 1e^{-4}$$

Entonces:

$$\mu_{LBM} = 1e^{-5}$$

y  $N_l$  es:

$$N_l = \frac{Re_{macro} * 1e^{-5}}{0.1 * \rho}$$

4.- Se despeja  $N_l$  de la ecuación y se obtiene el numero de nodos.

Suponiendo que  $\rho = 1.22$  y  $Re_{macro} = 6e^5$

$$N_l = \frac{6e^5 * 1e^{-5}}{0.1 * 1.22}$$

$$N_l = 499$$

5.- Utilizando la ecuación 6.32 se obtiene el factor de relajación  $\omega$  utilizando un valor  $Ma$  menor a 0.33. El usuario establecerá un valor de  $Ma$  adecuado para su simulación. En el caso del presente estudio se utilizará el valor de  $Ma = 0.22$  ya que ofrece un error  $LBM_{error} = 4.8\%$ .

$$\omega = \frac{0.22 * 499 * \sqrt{3}}{6e^5} + 0.5$$

$$\omega = 0.5033$$

Lo anterior se encuentra ilustrado en la siguiente imagen:

```
Macro Re: 609999.9999999999
Omega: 0.5033260607164541
Ma: 0.22
Ma_error= 0.0484 4.84%
nx:
499
ny: 166
Press any key to continue . . .
```

Figura 6.5: Impresión de los valores iniciales en pantalla para el arreglo D2Q9 de LBM en python

En este caso se definió el flujo con una velocidad  $U = 5m/s$ , viscosidad  $\mu = 1e - 05$ , densidad  $\rho = 1.22kg/m^3$  y una longitud de referencia  $l_{ref} = l_y [m]$ . Siendo el valor de  $l_y[m] = 1$  (línea 11 del código anterior)

Los valores arbitrarios utilizados para LBM fueron  $U_{LBM} = 0.1$  y  $\mu_{LBM} = 1e - 04$ .

**Paso 1. Inicialización de variables.**

Recordando las ecuaciones 6.5, 6.6, 6.7, 6.8, 6.9 y 6.10

$C_k$  es cada una de las funciones de distribución de velocidad representadas en vectores unitarios.

Para el arreglo D2Q9 se tienen los siguientes vectores

$$\begin{array}{lllll} C_0 = (0, 0), & C_1 = (1, 0), & C_2 = (0, 1), & C_3 = (-1, 0), & C_4 = (0, -1), \\ C_5 = (1, 1), & C_6 = (-1, 1), & C_7 = (-1, -1), & C_8 = (1, -1) \end{array}$$

Se creará un arreglo en python, en donde se asignarán valores igual a -1, 0, y 1 a los ejes X, Y del arreglo mediante ciclos *for*. Esto resultará en una matriz de 1 x 1 conteniendo así los valores de los vectores de velocidad  $c_i$  unitarios según su coordenada (x,y). Se expresa en python de la siguiente manera.

---

```
c = array([(x,y) for x in [0,-1,1] for y in [0,-1,1]])
```

---

Se deben asignar los valores de ponderación correspondientes al arreglo D2Q9 de LBM. Estos factores son los siguientes:

$$w_0 = 4/9, \quad w_1 = w_2 = w_3 = w_4 = 1/9, \quad w_5 = w_6 = w_7 = w_8 = 1/36$$

Por lo tanto en python podemos decir que:

---

```
t = 1./36. * ones(q)
t[asarray([norm(ci)<1.1 for ci in c])] = 1./9.; t[0] = 4./9.
```

---

En donde la variable  $c$  indica las diferentes velocidades.

Con el ciclo *for* se toman velocidades desde  $c_0$  hasta  $c_8$  y se asigna un valor  $t_i$  a la lista  $t$ . Si el valor total del vector  $c$  es menor a 1.1, es decir, si  $c = c_1 = c_2 = c_3 = c_4$  el valor de ponderación  $t$  es  $t = 1/9$ . Si  $c = c_0$  entonces  $t = 4/9$  y si el valor de  $c$  es cualquier otro que no haya sido previamente asignado entonces  $t$  será  $t = 1/36$ .

La velocidad se define con la siguiente línea de código.

---

```
vel = fromfunction(lambda d,x,y: (1-d)*uLB,(2,nx,ny))
```

---

En donde el valor 1 en el término  $(1 - d)$  indica la componente en  $x$  y su respectiva proporción en  $y$ . Si el valor de 1 se cambia por cualquier otra cifra menor a 1, la componente en  $x$  será menor y la componente en  $y$  incrementará desde cero.

## Paso 2. Función de Equilibrio

Es necesario recordar la ecuación 6.6. En donde existe un término que multiplica a las variables  $w_k \rho$ . Este término le asignaremos la variable  $B$  y se manipulará para llegar a la forma de la ecuación 6.37

$$f_k^e q = w_k \rho B \quad (6.33)$$

$$B = 1 + \frac{C_k \cdot U}{C_s^2} + \frac{1}{2} \frac{(C_k \cdot U)^2}{C_s^4} - \frac{1}{2} \frac{U^2}{C_s^2} \quad (6.34)$$

Utilizando la ecuación 6.8, el término  $B$  adquiere la siguiente forma.

$$B = 1 + \frac{C_k \cdot U}{\frac{1}{3}} + \frac{1}{2} \frac{(C_k \cdot U)^2}{\frac{1}{3^2}} - \frac{1}{2} \frac{U^2}{\frac{1}{3}} \quad (6.35)$$

$$B = 1 + 3C_k \cdot U + \frac{3^2}{2} (C_k \cdot U)^2 - \frac{3}{2} U^2$$

Se sabe que  $U$  es  $U = u + v$  y por lo tanto:

$$U^2 = u^2 + v^2 \quad (6.36)$$

Usando 6.36 en 6.35 tenemos que:

$$B = 1 + 3C_k \cdot U + \frac{3^2}{2} (C_k \cdot U)^2 - \frac{3}{2} u^2 + v^2 \quad (6.37)$$

La ecuación 6.37 es la forma del término  $B$  que utilizaremos para programar la función de equilibrio en LBM.

---

```
def equilibrium(rho,u):                                # Función de distribución de equilibrio.
    cu = dot(c,u.transpose(1,0,2))
    usqr = 3./2.*(u[0]**2+u[1]**2)
    feq = zeros((q,nx,ny))
    for i in range(q):
        feq[i,:,:] = rho*t[i]*(1.+3*cu[i]+(3**2*0.5*cu[i]**2)-usqr)
    return feq
```

---

### Paso 3. Condiciones de Frontera

Se deben definir las condiciones de frontera en python. Utilizando la *fig.6.4* y la *fig.6.3*, sabemos que se deben utilizar tres tipos de condiciones de frontera.

#### Condición de rebote

La condición de frontera de rebote esta definida por las ecuaciones 6.14, 6.15 y 6.16. Lo cual es puede expresar en otros términos como:

$$c_i = -c_i \quad (6.38)$$

En python se define como:

---

```
noslip = [c.tolist().index((-c[i]).tolist()) for i in range(q)]
```

---

En donde se copia una lista mediante la función `tolist()` con los valores negativos de cada velocidad, indicando así que la trayectoria de las partículas sera en dirección opuesta a los vectores de velocidad definidos en las ecuaciones 6.14, 6.15, 6.16 y en el diagrama de la *fig.6.1.1.4*

En el caso del escalón en un volumen abierto la condición de rebote estará definida en el escalón y en la pared inferior, la cual se entiende como el suelo sobre el cual se posiciona el objeto de estudio.

Para definir en que posición se utilizará la condición de rebote se hace lo siguiente:

---

```
LowerWall = fromfunction(lambda x,y: y==ny, (nx,ny))
Step=fromfunction(lambda x,y: (x-cx)**120+(y-cy)**120<rad_nodes**110, (nx,ny))
```

---

Se puede observar en las líneas anteriores la creación de una función temporal *lambda* en python, la cual asigna valores en dos ejes *x* y *y* con los nodos especificados por los arreglos *nx* y *ny*.

Estas líneas de código aislará los valores de los arreglos *nx* y *ny* para asignar la condición de rebote mas adelante en el código.

#### Condición de frontera de velocidad conocida

La condición de frontera de velocidad conocida se expresa según la ecuación 6.18. Esta condición se conoce también como la condición de frontera Zou/He.

Por otra parte tenemos la condición de frontera abierta definida por la ecuación 6.22, 6.23 y 6.24

Antes de programar estas dos condiciones debemos seguir cierta estrategia para mantener la simplicidad del método de LBM en python. Para ello, agruparemos las funciones de distribución de velocidades de cada lattice en 5 grupos.

Los primeros 3 grupos serán  $i_1, i_2$  e  $i_3$ . Ilustrados en la siguiente imagen.

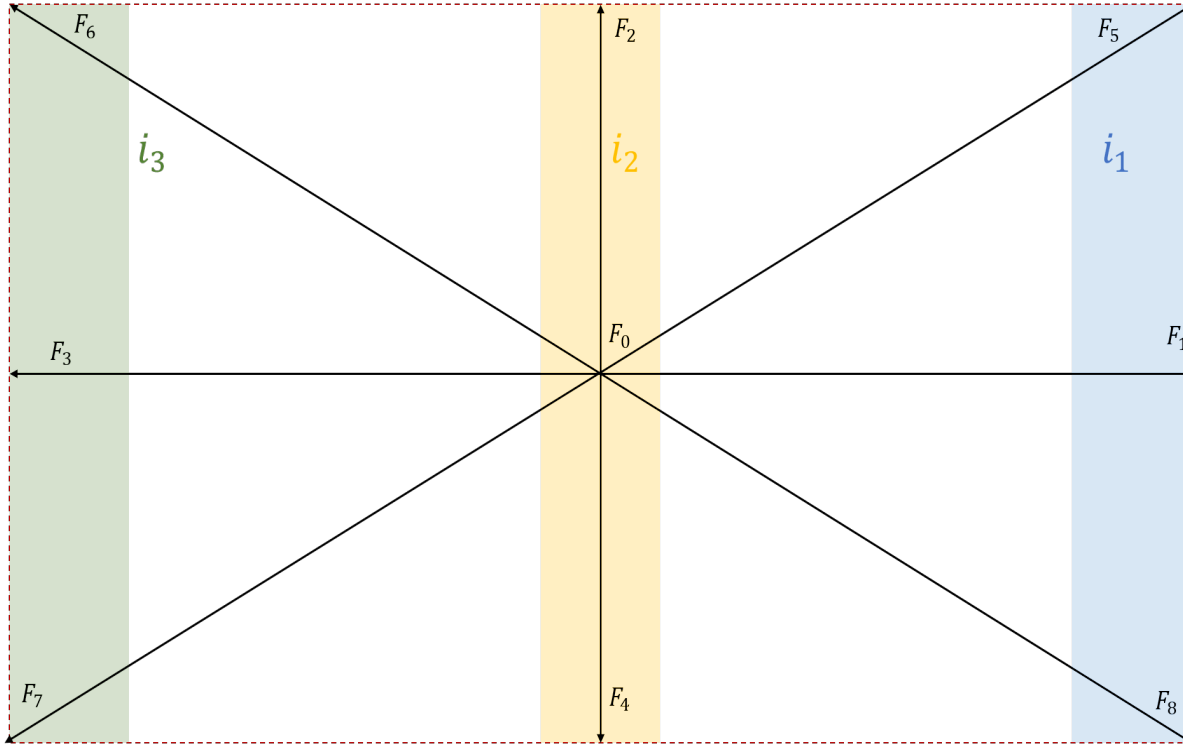


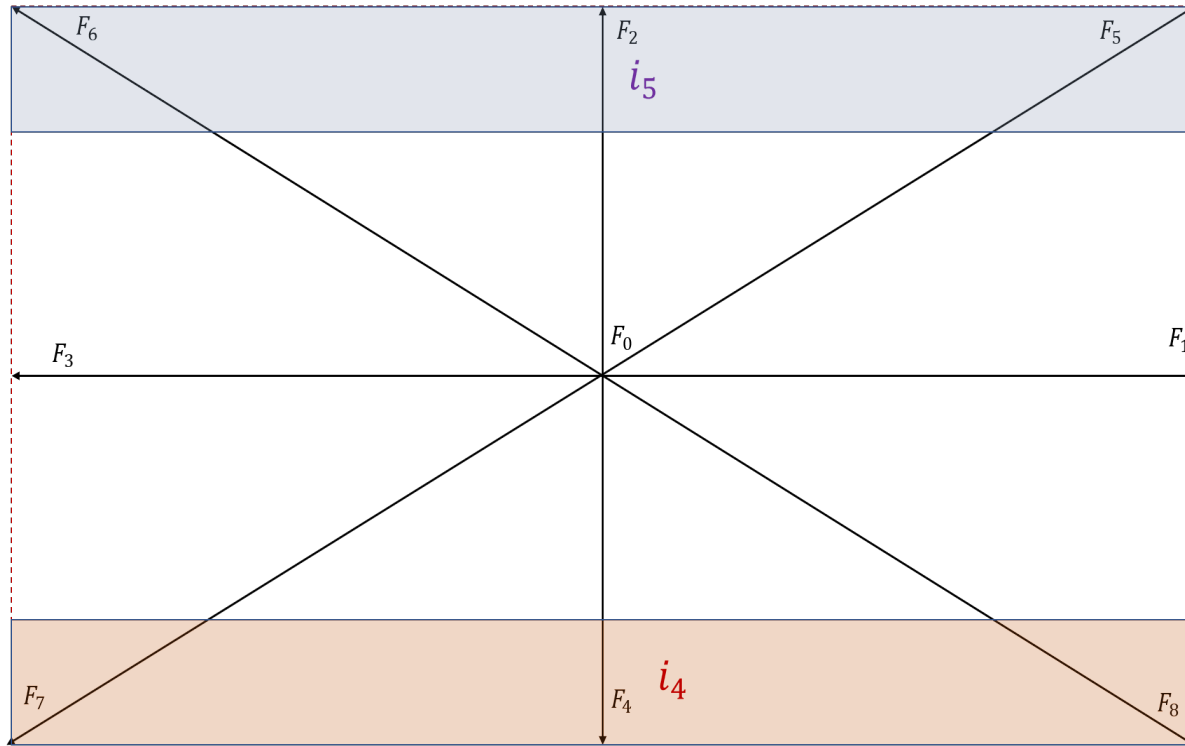
Figura 6.6: Agrupaciones  $i_1, i_2$  e  $i_3$

El grupo  $i_1$  corresponde a las funciones  $f_5, f_1$  y  $f_8$ ,  $i_2$  a  $f_2, f_0$  y  $f_4$  y por último  $i_3$  a  $f_6, f_3$  y  $f_7$

Los siguientes dos grupos serán utilizados para el caso en donde el volumen esta abierto y el escalón posicionado sobre una condición de frontera de rebote.

El grupo  $i_4$  corresponde a las funciones  $f_7, f_4$  y  $f_8$  y la agrupación  $i_5$  a  $f_6, f_2$  y  $f_5$ .



Figura 6.7: Agrupaciones  $i_4, i_5$ 

En python estas agrupaciones se expresan como:

---

```
i1 = arange(q)[asarray([ci[0]<0 for ci in c])] # f1,f5,f6
i2 = arange(q)[asarray([ci[0]==0 for ci in c])] # f2,f0,,f4
i3 = arange(q)[asarray([ci[0]>0 for ci in c])] # f3,f6,f7

i4 = arange(q)[asarray([ci[1]<0 for ci in c])] # f7,f4,f8
i5 = arange(q)[asarray([ci[1]>0 for ci in c])] # f6,f2,f5
```

---

Usando los grupos  $i_1$  e  $i_3$  y la definición de la condición de frontera de velocidad conocida por medio de la ecuación 6.18. En python se expresa que la condición Zou/He es:

---

```
fin[i3,0,:] = fin[i1,0,:] + feq[i3,0,:] - fin[i1,0,:]
```

---

El proceso para programar en python la condición de frontera abierta es el mismo, solo cambian la ecuación y el grupo de funciones que se utilizarán.

En caso de que se use esta condición para el escalón en un túnel de viento se expresa de la siguiente manera.

---

```
fin[i1,-1,:] = fin[i3,-2,:]
```

---

En caso de que se use esta condición para el escalón en un dominio abierto se expresa de la siguiente manera.

---

```
fin[i5,:,0] = fin[i4,:,1]
```

---

### Paso 4. Colisión

La ecuación 6.5, describe la colisión y movimiento de las partículas después del impacto entre ellas. la ecuación 6.5 se puede manipular para llegar una forma que resulte mas fácil de programar.

$$\begin{aligned}
 f_k(x + \Delta x, t) &= f_k(x, t)[1 - \omega] + \omega f_k^{eq}(x, t) \\
 f_k(x + \Delta x, t) &= f_k(x, t) - \omega f_k(x, t) + \omega f_k^{eq}(x, t) \\
 f_k(x + \Delta x, t) &= f_k(x, t) - \omega[f_k(x, t) - f_k^{eq}(x, t)]
 \end{aligned} \tag{6.39}$$

### Paso 5. Streaming

$$\begin{aligned}
 f_{k_{i+1}}(x + \Delta x, t + \Delta t) &= f_k(x + \Delta x, t) \\
 f_k(x + \Delta x, t + \Delta t) &= f_k(x, t) - \omega[f_k(x, t) - f_k^{eq}(x, t)]
 \end{aligned} \tag{6.40}$$

La ecuación 6.40 contiene los pasos 4 y 5, y se expresa en python de la siguiente manera:

---

```
fout = fin - omega * (fin - feq)
```

---

El paso de streaming se explica en la figura 6.8.

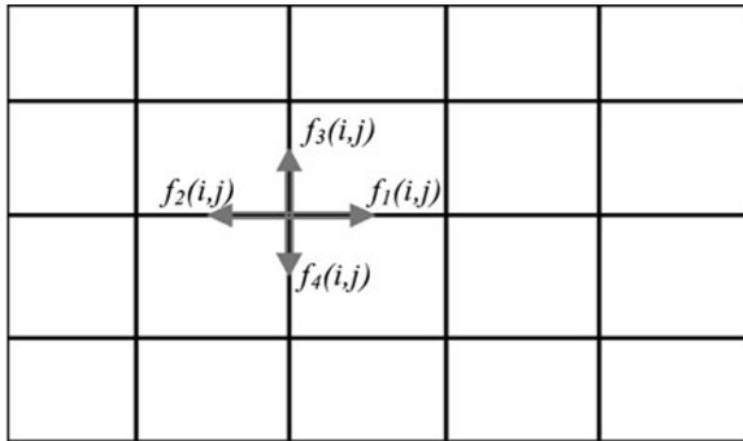


Figura 6.8: Ilustración del paso de streaming en un arreglo D2Q4

En un arreglo D2Q4, con 4 funciones de distribución de velocidad durante el paso de streaming la función  $f_1$  se mueve de  $f_1(i, j)$  a  $f_1(i + 1, j)$ , en el caso de  $f_2$  la función se mueve de  $f_2(i, j)$  a  $f_2(i - 1, j)$ , para  $f_3$  el movimiento es de  $f_3(i, j)$  a  $f_3(i, j + 1)$  y por último  $f_4$  se mueve de  $f_4(i, j)$  a  $f_4(i, j - 1)$ .

Elevando la complejidad a un arreglo D2Q9 los movimientos de las funciones son los siguientes según la figura 6.9

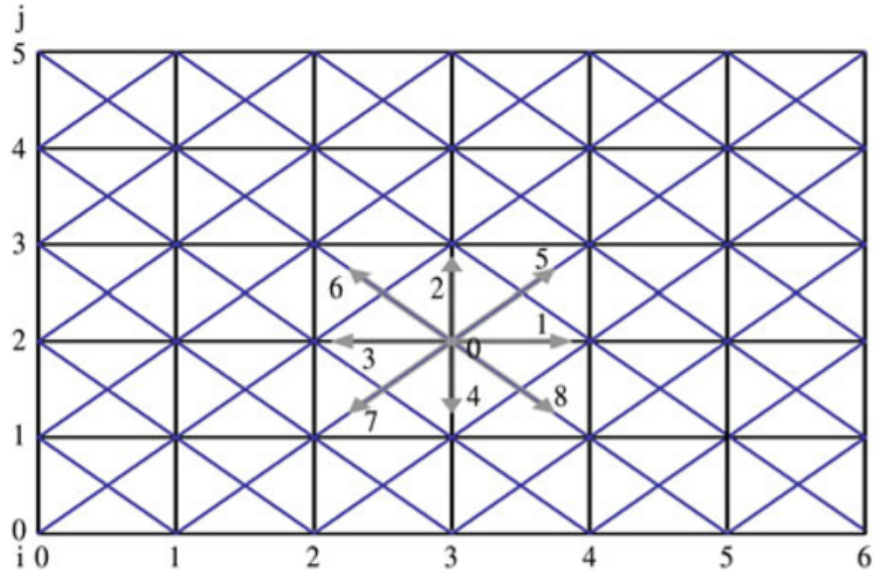


Figura 6.9: Ilustración del paso de streaming en un arreglo D2Q9

$$f_0(i, j) \text{ a } f_0(i, j)$$

$$f_1(i, j) \text{ a } f_1(i + 1, j)$$

$$f_2(i, j) \text{ a } f_2(i, j + 1)$$

$$f_3(i, j) \text{ a } f_3(i - 1, j)$$

$$f_4(i, j) \text{ a } f_4(i, j - 1)$$

$$f_5(i, j) \text{ a } f_5(i + 1, j + 1)$$

$$f_6(i, j) \text{ a } f_6(i - 1, j + 1)$$

$$f_7(i, j) \text{ a } f_7(i - 1, j - 1)$$

$$f_8(i, j) \text{ a } f_8(i + 1, j - 1)$$

### Paso 6. Cálculo de las variables macroscópicas

Después del paso de streaming es necesario recalcular las variables macroscópicas de cada nodo.

Recordemos que las 9 funciones de una misma variable multiplicadas por un factor de ponderación indican la probabilidad de una de esas funciones (partículas) de ir en cierta dirección y formar parte de el siguiente nodo. Es por esta razón que la variable macroscópica debe re-calcularse después del paso de streaming, ya que al moverse de un nodo a otro los valores de las funciones no son los mismos en dichos nodos.

El cálculo de una variable macroscópica en este caso la densidad  $\rho$  solo es la suma de cada una de las funciones de velocidad  $\sum f_k$  en cada lattice. Véase ecuación 6.41

$$\sum_{k=0}^8 f_k(i, j) \quad (6.41)$$

En python utilizaremos la siguientes líneas de código.

---

```
sumpop = lambda fin: sum(fin,axis=0)
rho = sumpop(fin)
```

---

A partir de este punto se repite el paso número 2 del diagrama de flujo 6.1.1.4.

Un timestep en el código de LBM se cumplirá cuando se cumplan los pasos del número 2 al número 6 del diagrama mencionado.

### Paso 7. Identificación de timestep y toma de decisión

El punto número 7 en el diagrama 6.1.1.4 es el indicador de salida de un ciclo for en python. El ciclo for terminará cuando el código haya alcanzado el ultimo timestep especificado.

### Paso 8. Indicador del final de la simulación

Cuando el código cumple el último timestep sale del ciclo for y se da por terminada la simulación.

---

```

1  for time in range(maxIter):
2
3      rho = sumpop(fin)    ;   u = dot(c.transpose(), fin.transpose((1,0,2)))/rho
4
5      u[:,0,:] = vel[:,0,:] # Left wall: compute density from known populations.
6      rho[0,:] = 1./(1.-u[0,0,:]) * (sumpop(fin[i2,0,:])+2.*sumpop(fin[i1,0,:]))
7
8      feq = equilibrium(rho,u)
9      fin[i3,0,:] = fin[i1,0,:] + feq[i3,0,:] - fin[i1,0,:] # Limite izquierdo
10                                     # Zou/He .
11      fin[i5,:,0] = fin[i4,:,1] # Pared superior; condición abierta.
12      #fin[i1,-1,0] = fin[i3,-2,:] #Límite derecho; condición abierta
13      fout = fin - omega * (fin - feq) # Collision step.
14      for i in range(q):
15          # fout[i,UpperWall] = fin[noslip[i],UpperWall]
16          fout[i,LowerWall] = fin[noslip[i],LowerWall]
17          fout[i,Step] = fin[noslip[i],Step]
18      for i in range(q): # Streaming step.
19          fin[i,:,:] = roll(roll(fout[i,:,:],c[i,0],axis=0),c[i,1],axis=1)

```

---

En el ciclo for se pueden observar los pasos del 2 al 6.

- Línea 8 (Paso 2)  
Cálculo de la función de equilibrio utilizando los valores iniciales  $f_k$
- Líneas 9 a 12, (Paso 3)  
Cálculo de condiciones de frontera con velocidad conocida y del tipo abierta.
- Líneas 13, (Paso 4)  
Colisión inicial sin obstáculo.
- Líneas 14 a 17, (Paso 3)  
Definición de condición de frontera en paredes y obstáculos .
- Líneas 18 y 19, (Paso 5)  
Paso de streaming.
- Línea 3, (Paso 6)  
Cálculo de las variables macroscópicas.

### Paso 9. Transformación de escala mesoscópica a macroscópica

En el Paso 0 se hizo una transformación de escala macroscópica a mesoscópica para poder resolver el flujo en un régimen adecuado al método de lattice boltzmann. Ahora, debemos pasar de una escala mesoscópica a una escala macroscópica para obtener los resultados deseados.

El método es el mismo que se usó en el Paso 0 igualando las ecuaciones 6.29 y 6.30.

La diferencia es que ahora despejaremos  $U_{macro}$  de la ecuación, y utilizaremos los valores finales de la simulación como  $U_{LBM}$ .

Se tiene que:

$$\begin{aligned}
 Re_{macro} &= Re_{LBM} \\
 \frac{U_{macro} * l_{ref} * \rho}{\mu_{macro}} &= \frac{U_{LBM_{final}} * N_l * \rho}{\mu_{LBM}} \\
 U_{macro} &= \frac{U_{LBM_{final}} * N_l * \rho * \mu_{macro}}{\mu_{LBM} * l_{ref} * \rho} \\
 U_{macro} &= \frac{U_{LBM_{final}} * N_l * \mu_{macro}}{\mu_{LBM} * l_{ref}} \tag{6.42}
 \end{aligned}$$

En python los expresaremos de la siguiente manera

---

```

1   macro_factor = (nx*vis)/(vis_lbm*lref)
2   cp=plt.clf()
3   plt.imshow(sqrt(((u[0]*macro_factor)**2)+((u[1]*macro_factor)**2)
4   ).transpose(),
5   cmap=cm.jet)
6   plt.title("Macro-scale velocity [m/s]");   plt.colorbar(cp)
7   plt.savefig(str(p)+"-"+str(tv)+"/vel."+str(time/frame).zfill(4)
8   +str(time)+".png")

```

---

La ecuación 6.42 se divide en un factor de macro escala y la velocidad  $U_{LBM}$  por practi-  
cidad.

$$U_{macro} = U_{LBM_{final}} * factor_{Macro} \tag{6.43}$$

La ecuación 6.43 se aprecia en las líneas 1 y 3 del código anterior. En donde  $u[0]$  y  $u[1]$  son  $u$  y  $v$ .

Las líneas siguientes del código grafican los contornos de velocidad y sus respectivos valores en un enfoque macroscópico. Siendo este el final del código.

## 6.2. Programación de las ecuaciones de Navier-Stokes de un fluido turbulento tri-dimensional

### 6.2.1. Flujo bidimensional

El problema que se presenta en primera instancia para resolver las ecuaciones de Navier Stokes, es que las tres ecuaciones de momento están intrincadamente acopladas porque cada componente de velocidad aparece en cada ecuación de momento y en la ecuación de continuidad. El problema más complejo a resolver es el papel que juega la presión, ya que aparece en las tres ecuaciones de momento, pero evidentemente no hay alguna otra ecuación para ella.

Si se conoce el gradiente de presión, el proceso de obtención de ecuaciones discretizadas para las velocidades a partir de las ecuaciones de momento es exactamente el mismo que el de los esquemas explicados en el Capítulo 3. Sin embargo, si se desea calcular el campo de presión como parte de la solución, su gradiente no se conoce de antemano.

Si el flujo es compresible, la ecuación de continuidad puede usar como la ecuación de transporte para la densidad y, además de las ecuaciones de momento, la ecuación de energía es para la temperatura. Así, la presión se puede obtener a partir de la densidad y la temperatura utilizando la ecuación de estado  $p = p(\rho, T)$ . No obstante, si el flujo es incompresible, la densidad es constante y, por definición, no está vinculada a la presión. En este caso, el acoplamiento entre presión y velocidad introduce una restricción en la solución: si se aplica el campo de presión correcto en las ecuaciones de momento, el campo de velocidad resultante debería satisfacer la continuidad.

Una forma de salir de esta dificultad es construir un campo de presión que garantice que se satisfaga la continuidad; tal relación se puede obtener tomando la divergencia de la ecuación de momento. Para lograr esto, se propone la ecuación de Poisson para la presión (Ecuación 6.44).

$$\frac{\partial \psi^2}{\partial^2 x} + \frac{\partial \psi^2}{\partial^2 y} = b \quad (6.44)$$

Donde  $\psi$  y  $b$  pueden ser funciones reales o complejas. En este caso, como se necesita una ecuación que defina el gradiente de presión,  $\psi$  tomará el valor de la presión, es decir,  $\psi = p$ . Únicamente haría falta definir la función  $b$ , para ello se utilizan las ecuaciones de momento a las cuales se les aplicará divergencia como se muestra en la Ecuaciones 6.45 y 6.46, con el fin de que el término de presión sea de segundo orden.

$$\frac{\partial}{\partial x} \left[ \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right] \quad (6.45)$$

$$\frac{\partial}{\partial y} \left[ \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \right] \quad (6.46)$$



Sumando estas dos ecuaciones se obtiene:

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + 2 \left( \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \right) + \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + u \frac{\partial^2 u}{\partial x^2} + u \frac{\partial^2 v}{\partial x \partial y} + v \frac{\partial^2 u}{\partial x \partial y} + v \frac{\partial^2 v}{\partial y^2} - \\ \nu \left[ \frac{\partial}{\partial x} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial}{\partial y} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \right] = - \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \end{aligned}$$

Como se puede observar, los últimos términos agrupados corresponden a los primeros de la ecuación de Poisson. Así, reagrupando los términos:

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + 2 \left( \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \right) + \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + u \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + v \frac{\partial}{\partial y} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) - \\ \nu \left[ \frac{\partial}{\partial x^2} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + \frac{\partial}{\partial y^2} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right] = - \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \end{aligned}$$

y simplificando con ayuda de la ecuación de continuidad, la ecuación de Poisson que liga la presión y la velocidad para dos dimensiones es:

$$- \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) = 2 \left( \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \right) + \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 \quad (6.47)$$

Una estrategia de solución iterativa que se puede adoptar es el algoritmo SIMPLE de Patankar y Spalding (1972). SIMPLE es un acrónimo de Método Semi-Implícito para Ecuaciones Vinculadas a Presión (*Semi-Implicit Method for Pressure Linked Equations*) y es un procedimiento numérico ampliamente utilizado para resolver problemas de dinámica de fluidos y transferencia de calor a través de las ecuaciones de Navier Stokes.

El algoritmo es iterativo. Los pasos básicos en de solución son los siguientes:

1. Establecer las condiciones de frontera.
2. Calcular los gradientes de velocidad y presión.
3. Resolver la ecuación de momento discretizada para calcular el campo de velocidad intermedia.
4. Resolver la ecuación de corrección de presión.
5. Actualizar el gradiente de presión.
6. Actualizar las condiciones de frontera de presión.
7. Corregir las velocidades en cada celda.

Una variante modificada es el algoritmo SIMPLER (SIMPLE Revised), que fue introducido por Patankar en 1979. Cada iteración del algoritmo SIMPLER requiere aproximadamente el doble de operaciones que el algoritmo SIMPLE original, ya que deben resolverse dos ecuaciones de presión de Poisson y dos conjuntos de ecuaciones de momento. Esta desventaja generalmente se ve compensada por una convergencia significativamente más rápida, por lo que SIMPLER es, en general, más eficiente.

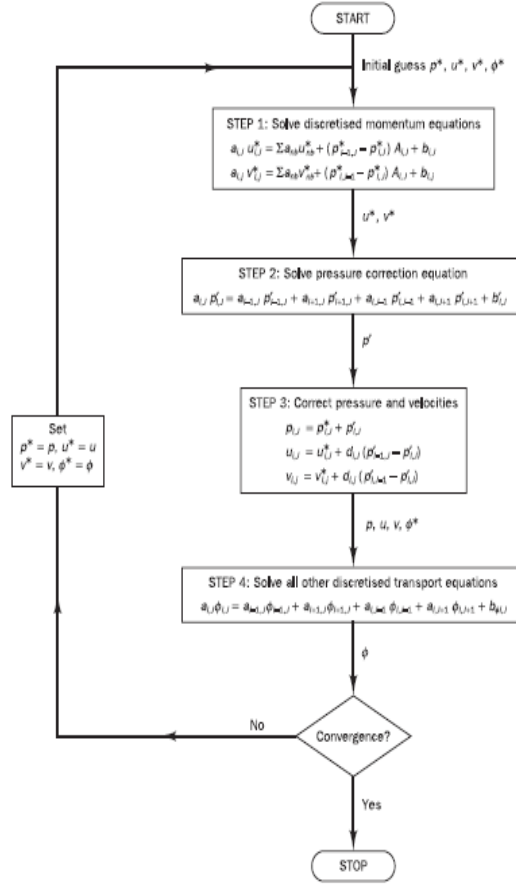


Figura 6.10: Diagrama de flujo del algoritmo SIMPLE (**Corregir**).

1. Establecer las condiciones de frontera.
2. Proponer los gradientes de presión y de velocidades.
3. Resolver la ecuación de Poisson discretizada.
4. Resolver la ecuación de corrección de presión.
5. Actualizar el gradiente de presión.
6. Actualizar las condiciones de frontera de presión.
7. Resolver las ecuaciones de momento con la presión actualizada.
8. Corregir las velocidades en cada celda.
9. Actualizar las condiciones de frontera de velocidad.

$$\begin{aligned}
 u_{i,j}^{n+1} = & u_{i,j}^n - u_{i,j}^n \frac{\Delta t}{\Delta x} (u_{i,j}^n - u_{i-1,j}^n) - v_{i,j}^n \frac{\Delta t}{\Delta y} (u_{i,j}^n - u_{i,j-1}^n) \\
 & - \frac{\Delta t}{2\rho\Delta x} (p_{i+1,j}^n - p_{i-1,j}^n) + \nu \left( \frac{\Delta t}{\Delta x^2} (u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n) \right. \\
 & \left. + \frac{\Delta t}{\Delta y^2} (u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n) \right)
 \end{aligned}$$

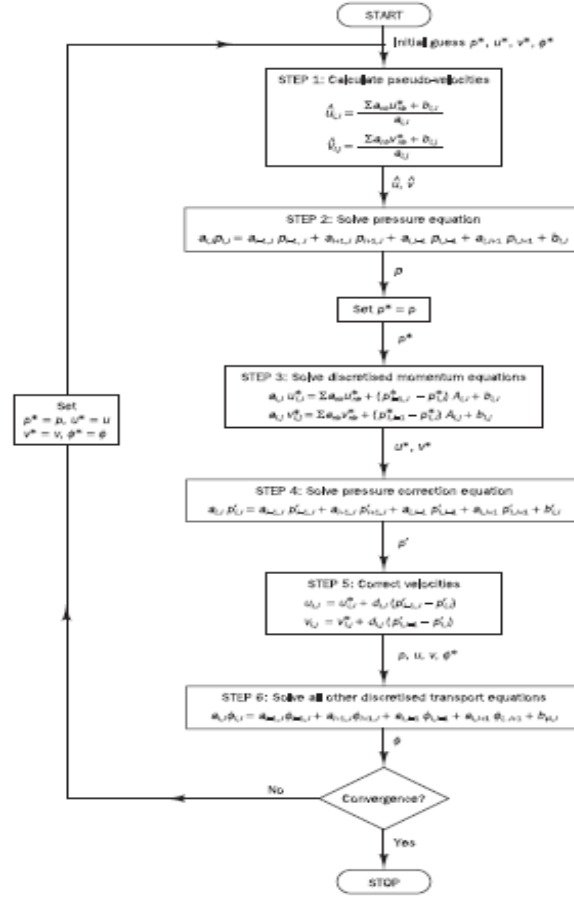


Figura 6.11: Diagrama de flujo del algoritmo SIMPLER (Corregir).

y para  $v$  es:

$$\begin{aligned}
 v_{i,j}^{n+1} = & v_{i,j}^n - u_{i,j}^n \frac{\Delta t}{\Delta x} (v_{i,j}^n - v_{i-1,j}^n) - v_{i,j}^n \frac{\Delta t}{\Delta y} (v_{i,j}^n - v_{i,j-1}^n) \\
 & - \frac{\Delta t}{2\rho\Delta x} (p_{i+1,j}^n - p_{i-1,j}^n) + \nu \left( \frac{\Delta t}{\Delta x^2} (v_{i+1,j}^n - 2v_{i,j}^n + v_{i-1,j}^n) \right. \\
 & \left. + \frac{\Delta t}{\Delta y^2} (v_{i,j+1}^n - 2v_{i,j}^n + v_{i,j-1}^n) \right)
 \end{aligned}$$

Reacomodando la ecuación de Poisson discretizada, tenemos:

$$\begin{aligned}
 P_{i,j}^n = & \frac{\Delta y^2 (P_{i+1,j}^n + P_{i-1,j}^n) + \Delta x^2 (P_{i,j+1}^n + P_{i,j-1}^n)}{2(\Delta x^2 + \Delta y^2)} - \frac{\rho \Delta x^2 \Delta y^2}{2(\Delta x^2 + \Delta y^2)} \\
 & \left[ \frac{1}{\Delta t} \left( \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta x} \right)^2 - \left( \frac{u_{i,j+1}^n - u_{i,j-1}^n}{\Delta y} \right) \left( \frac{v_{i+1,j}^n - v_{i-1,j}^n}{\Delta x} \right) \right. \\
 & \left. - \left( \frac{v_{i,j+1}^n - v_{i,j-1}^n}{2\Delta y} \right)^2 \right]
 \end{aligned}$$

### 6.2.2. Flujo tridimensional

Para un flujo tridimensional se sigue el mismo procedimiento que para el flujo bidimensional, sólo que se agrega el término de velocidad  $w$  en  $z$ . La Ecuación 6.48 muestra la ecuación genérica de Poisson para tres dimensiones.

$$\frac{\partial \psi^2}{\partial^2 x} + \frac{\partial \psi^2}{\partial^2 y} + \frac{\partial \psi^2}{\partial^2 z} = b \quad (6.48)$$

Cada una de las ecuaciones de momento se multiplica por  $\frac{\partial}{\partial x}$ ,  $\frac{\partial}{\partial y}$  y  $\frac{\partial}{\partial z}$  respectivamente (Ecuaciones 6.49 -6.51).

$$\frac{\partial}{\partial x} \left[ \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \right] \quad (6.49)$$

$$\frac{\partial}{\partial y} \left[ \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \right] \quad (6.50)$$

$$\frac{\partial}{\partial z} \left[ \left( \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \right] \quad (6.51)$$

Sumando estas tres ecuaciones da como resultado:

$$\begin{aligned} & \frac{\partial}{\partial t} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + 2 \left( \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \right) + 2 \left( \frac{\partial u}{\partial z} \frac{\partial w}{\partial x} \right) + 2 \left( \frac{\partial v}{\partial z} \frac{\partial w}{\partial y} \right) + \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial z} \right)^2 + \\ & u \frac{\partial^2 u}{\partial x^2} + u \frac{\partial^2 v}{\partial x \partial y} + u \frac{\partial^2 w}{\partial x \partial z} + v \frac{\partial^2 u}{\partial x \partial y} + v \frac{\partial^2 v}{\partial y^2} + v \frac{\partial^2 w}{\partial y \partial z} + w \frac{\partial^2 u}{\partial x \partial z} + w \frac{\partial^2 v}{\partial y \partial z} + w \frac{\partial^2 w}{\partial z^2} - \\ & \nu \left[ \frac{\partial}{\partial x} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + \frac{\partial}{\partial y} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) + \frac{\partial}{\partial z} \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \right] = - \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} \right) \end{aligned}$$

Reagrupando los términos, se llega a:

$$\begin{aligned} & \frac{\partial}{\partial t} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + 2 \left( \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \right) + 2 \left( \frac{\partial u}{\partial z} \frac{\partial w}{\partial x} \right) + 2 \left( \frac{\partial v}{\partial z} \frac{\partial w}{\partial y} \right) + \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial z} \right)^2 + \\ & u \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + v \frac{\partial}{\partial y} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + w \frac{\partial}{\partial z} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) - \\ & \nu \left[ \frac{\partial}{\partial x^2} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + \frac{\partial}{\partial y^2} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + \frac{\partial}{\partial z^2} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] = - \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} \right) \end{aligned}$$

y simplificando con ayuda de la ecuación de continuidad, la ecuación de Poisson que liga la presión y la velocidad para tres dimensiones es:

$$\begin{aligned} - \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} \right) &= 2 \left( \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \right) + 2 \left( \frac{\partial u}{\partial z} \frac{\partial w}{\partial x} \right) + 2 \left( \frac{\partial v}{\partial z} \frac{\partial w}{\partial y} \right) + \\ & \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial z} \right)^2 \end{aligned} \quad (6.52)$$

## Sección IV

# Resultados y Conclusiones

## Capítulo 7

### Resultados y conclusiones de códigos en 2D

## Capítulo 8

# Resultados y conclusiones de códigos en 3D

## Sección V

### Anexos



## Apéndice A

Código completo de Lattice Boltzmann para un arreglo D2Q9 y post-procesamiento de datos.

```

1  import os ; from os import * ; import shutil
2  from numpy import *; from numpy.linalg import *
3  import matplotlib.pyplot as plt; from matplotlib import cm
4  import cv2
5  import numpy as np
6  import glob
7  ##### Flow definition #####
8  maxIter = 20000 # Total number of time iterations.
9  frame = 200 ;
10 rho = 1.22
11 lenx, leny = 3, 1
12 radius,U,vis = 0.4,5,1e-5 ##### Macro Variables setup
13 lref=leny
14 Re      = lref*rho*U/vis; print("Macro Re: ",Re)
15 uLB     = 0.1;    vis_lbm = 1e-4 ;
16 nx = int(Re*vis_lbm/(uLB*rho));    ny = int(leny*nx/lenx);    ly=ny-1.0
17 q = 9
18 dx = lenx/nx ; dt = dx                #Lattice dimensions and populations.
19 cx = nx/2; cy=ny;    rad_nodes=int(radius*nx/lenx); Ma = 0.22
20 Ma_err = Ma**2                # Coordinates of step and radius size.
21 nulb    = uLB*nx/Re; omega = (Ma/((dx/(nx*((1/3)**2)))*Re))+0.5
22 print("\nOmega: "+str(omega)+"\n"+"Ma: "+str(Ma)+"\n"+
23 "Ma_error= "+str(Ma_err)+"    "+str(Ma_err*100)+"perc"+" \n"+"nx: "+str(nx)+" \n"+
24 str(nx)+" \n"+"ny: "+str(ny))
25
26
27 os.system("pause")
28
29 # Post-process data definition before running Simulation #####
30
31 crpth=os.path.abspath(os.getcwd())
32 tv = "total_velocity" ; vxf = "Velocity_X" ; vyf = "Velocity_Y"
33 p = "results" ###Folders names

```

```

34
35 shutil.rmtree(str(p))
36 #Comment if first time to run script, Uncomment if script was ran to
37 overwrite results folder
38
39 os.mkdir(str(p)) ; os.chdir(str(p)) #Results folder definition
40 os.mkdir(str(tv)) ; os.mkdir(str(vxf)) ; os.mkdir(str(vyf))
41 #Creation of folder inside results directory
42
43 os.chdir(str(crpth))
44
45 ##### Lattice Constants #####
46 c = array([(x,y) for x in [0,-1,1] for y in [0,-1,1]])
47 # Lattice velocities.
48
49 t = 1./36. * ones(q) # Lattice weights.
50 t[asarray([norm(ci)<1.1 for ci in c])] = 1./9.; t[0] = 4./9.
51
52 noslip = [c.tolist().index((-c[i]).tolist()) for i in range(q)]
53
54 i1 = arange(q)[asarray([ci[0]<0 for ci in c])]
55 # Unknown on right wall. (f1,f5,f6)
56
57 i2 = arange(q)[asarray([ci[0]==0 for ci in c])]
58 # Vertical middle. (f2,f0,,f4)
59
60 i3 = arange(q)[asarray([ci[0]>0 for ci in c])]
61 # Unknown on left wall. (f3,f6,f7)
62
63 i4 = arange(q)[asarray([ci[1]<0 for ci in c])] # f7,f4,f8
64 i5 = arange(q)[asarray([ci[1]>0 for ci in c])] # f6,f2,f5
65
66 ##### Function Definitions #####
67
68 sumpop = lambda fin: sum(fin,axis=0)
69 # Helper function for density computation.
70

```

```

71 def equilibrium(rho,u): # Equilibrium distribution function.
72     cu = dot(c,u.transpose(1,0,2))
73     usqr = 3./2.*(u[0]**2+u[1]**2)
74     feq = zeros((q,nx,ny))
75     for i in range(q):
76         feq[i,:,:] = rho*t[i]*(1.+3*cu[i]+(3**2*0.5*cu[i]**2)-usqr)
77     return feq
78
79
80 ##### Setup: obstacle and velocity inlet with perturbation #####
81 #UpperWall = fromfunction(lambda x,y: y==0, (nx,ny))
82 LowerWall = fromfunction(lambda x,y: y==ny, (nx,ny))
83 Step=fromfunction(lambda x,y:(x-cx)**120+(y-cy)**120<rad_nodes**110,(nx,ny))
84
85 vel = fromfunction(lambda d,x,y: (1-d)*uLB,(2,nx,ny))
86 ##### Change 1 to any lower value to change vx vy components.
87
88 print("Initialized velocity values \n"); print(vel) ; os.system("pause")
89
90 feq = equilibrium(rho,vel); fin = feq.copy()
91
92 ##### Main time loop #####
93
94 for time in range(maxIter):
95
96     rho = sumpop(fin)
97     u = dot(c.transpose(), fin.transpose((1,0,2)))/rho
98     # Calculate macroscopic density and velocity.
99
100     u[:,0,:] =vel[:,0,:] # Left wall:
101     #compute density from known populations.
102
103     rho[0,:]= 1./(1.-u[0,0,:])*(sumpop(fin[i2,0,:])+2.*sumpop(fin[i1,0,:]))
104
105     feq = equilibrium(rho,u)
106     fin[i3,0,:] = fin[i1,0,:] + feq[i3,0,:] - fin[i1,0,:] #Límite izquierdo
107                                     # Zou/He.

```

```

108     fin[i5,:,0] = fin[i4,:,1] # Right wall: outflow condition.
109     #fin[i1,-1,0] = fin[i3,-2,:] #Pared derecha; condición abierta
110     fout = fin - omega * (fin - feq) # Collision step.
111
112     for i in range(q):
113         # fout[i,UpperWall] = fin[noslip[i],UpperWall]
114         fout[i,LowerWall] = fin[noslip[i],LowerWall]
115         fout[i,Step] = fin[noslip[i],Step]
116
117     for i in range(q): # Streaming step.
118         fin[i,:,:) = roll(roll(fout[i,:,:),c[i,0],axis=0),c[i,1],axis=1)
119
120
121     if (time%frame==0): # Visualization Frames
122         macro_factor = (nx*vis)/(vis_lbm*lref)
123         cp=plt.clf()
124         plt.imshow(sqrt(((u[0]*macro_factor)**2)+
125             ((u[1]*macro_factor)**2)).transpose(),cmap=cm.jet)
126         plt.title("Macro-scale velocity [m/s]"); plt.colorbar(cp)
127         plt.savefig(str(p)+"/"+str(tv)+"/vel."+
128             str(time/frame).zfill(4)+str(time)+".png")
129
130         vx=plt.clf(); plt.imshow((u[0]*macro_factor).transpose(),cmap=cm.jet)
131         plt.title("Macro-scale velocity X [m/s]"); plt.colorbar(vx)
132         plt.savefig(str(p)+"/"+str(vxf)+"/vel_X."+
133             str(time/frame).zfill(4)+str(time)+".png")
134
135         vy=plt.clf(); plt.imshow((u[1]*macro_factor).transpose(),cmap=cm.jet)
136         plt.title("Macro-scale velocity Y [m/s]"); plt.colorbar(vx)
137         plt.savefig(str(p)+"/"+str(vyf)+"/vel_X."+
138             +str(time/frame).zfill(4)+str(time)+".png")
139
140     ##### VIDEOS #####
141
142     img_array = []
143     for filename in glob.glob(str(p)+"/"+str(tv)+'/*.*png'):
144         img = cv2.imread(filename) ; height, width, layers = img.shape ; size =

```

---

```
145     (width,height) ; img_array.append(img)
146 out = cv2.VideoWriter(str(p)+"/"+str(tv)+'project.avi',
147 cv2.VideoWriter_fourcc(*'DIVX'), 15, size)
148 for i in range(len(img_array)):
149     out.write(img_array[i])
150 out.release()
151
152 for filename in glob.glob(str(p)+"/"+str(vxf)+'/*.png'):
153     img = cv2.imread(filename) ; height, width, layers = img.shape ; size =
154     (width,height) ; img_array.append(img)
155 out = cv2.VideoWriter(str(p)+"/"+str(vxf)+'project.avi',
156 cv2.VideoWriter_fourcc(*'DIVX'), 15, size)
157 for i in range(len(img_array)):
158     out.write(img_array[i])
159 out.release()
160
161 for filename in glob.glob(str(p)+"/"+str(vyf)+'/*.png'):
162     img = cv2.imread(filename) ; height, width, layers = img.shape ; size =
163     (width,height) ; img_array.append(img)
164 out = cv2.VideoWriter(str(p)+"/"+str(vyf)+'project.avi',
165 cv2.VideoWriter_fourcc(*'DIVX'), 15, size)
166 for i in range(len(img_array)):
167     out.write(img_array[i])
168 out.release()
```

---

# Bibliografía

- Anderson, J. D. (1995). *Computational fluid dynamics : the basics with applications*. McGraw-Hill series in mechanical engineering. McGraw-Hill, New York.
- Costantini, M., Risius, S., and Klein, C. (2015). Experimental investigation of the effect of forward-facing steps on boundary layer transition. *Procedia IUTAM*, 14:152 – 162. IUTAM\_ABCM Symposium on Laminar Turbulent Transition. Recuperado el 01 de julio de 2019, de <http://www.sciencedirect.com/science/article/pii/S2210983815000620>.
- Hirt, C. W., Amsden, A. A., and Cook, J. L. (1997). An arbitrary lagrangian-eulerian computing method for all flow speeds. pages 203–216. <https://pdfs.semanticscholar.org/8ed9/3eddb72dcd457947e42836d50202dbc8e35c.pdf>.
- Iftekhar, H., Shao, W., and Agelin-Chaab, M. (2015). Reynolds number effect on forward facing step in a pressure gradient. In *Proceedings of the 25th CANCAM*. Recuperado el 01 de julio de 2019, de <https://tinyurl.com/y5847g98>.
- McDonough, J. M. (2013). *Computational fluid dynamics of Incompressible Flow: Mathematics, Algorithms and Implementations*. Departments of Mechanical Engineering and Mathematics University of Kentucky, Kentucky. <http://web.engr.uky.edu/acfd/me691-lctr-nts.pdf>.
- Rhie, C. M. and Chow, W. L. (1983). *Numerical study of the turbulent flow past an airfoil with trailing edge separation*. AIAA J .11.
- Sherry, M., Jacono, D., Sheridan, J., Mathis, R., and Marusic, I. (2009). Flow separation characterisation of a forward facing step immersed in a turbulent boundary layer. *Sixth International Symposium on Turbulence and Shear Flow Phenomena*, pages 1325–1330. Recuperado el 01 de julio de 2019, de <https://tinyurl.com/y5v544au>.