

*Simulación Numérica de
Flujo de Fluido Viscoso e
Incompresible en 2D*

Juan Rubén Medina Eliett
Miguel Omar Fagúndez Córcega

26 de noviembre de 2003



UNIVERSIDAD DE CARABOBO
Facultad Experimental de Ciencias y Tecnología
Departamento de Computación

*Simulación Numérica de
Flujo de Fluido Viscoso e Incompresible en 2D*

Autores:

Juan Rubén Medina Eliett

Miguel Omar Fagúndez Córcega

Tutor: Prof. Carlos E. Cadenas R.

Co-Tutor: Prof. Germán Larrazabal

Trabajo Especial de Grado presentado ante la Ilustre Universidad de Carabobo
como credencial de mérito para optar al título de Licenciado en Computación.

Bárbula, 26 de noviembre de 2003

Resumen

En este trabajo se presentan los resultados preliminares de la simulación de flujo de fluidos en una y dos dimensiones basada en métodos conservativos, resolviendo numéricamente las ecuaciones de Navier-Stokes y de compresibilidad artificial. Se presentan cuatro esquemas numéricos para 1D, dos explícitos y dos implícitos, haciendo uso de los métodos de Operadores de Soporte 1-2-1 y el método Castillo-Grone 2-2-2, respectivamente. En 2D se desarrollan e implementan igualmente bajo un esquema implícito los métodos mencionados. En estas técnicas creadas recientemente, los operadores discretos mantienen las propiedades conservativas de los operadores continuos de divergencia, gradiente y laplaciano. Aun están siendo probadas a diferentes niveles desde el punto de vista conceptual, y es por ello que esta investigación estaría colocando precedentes tanto en el área numérica, como en el campo de aplicación al cual está siendo enfocada. Se propone una metodología de desarrollo basada únicamente en la simplicidad, comunicación, retroalimentación y reutilización del código desarrollado. En la resolución del sistema de ecuaciones lineales obtenido en los esquemas implícitos, se usó la biblioteca UCSparseLib desarrollada en ANSI C. Además, se crearon herramientas necesarias para la visualización final de resultados haciendo uso de OpenGL.

Qué maravilloso es tener tan poco que pedir y tanto que agradecer

A Jehová Dios, A mis viejos Lourdes y Francisco, A mis dos hermanas

Juan Rubén Medina Eliett

A Dios por darme toda la fuerza de la vida, a mis padres Zulay del Valle Córcega y Miguel Omar Fagúndez Montenegro por ser los responsables de la persona que soy, a Saulymar por ser siempre fuente de apoyo e inspiración.

Los quiero a todos.

Miguel Omar Fagúndez Córcega

Agradecimiento

“Para hacer una torta de manzana, primero tienes que crear un universo.”

Carl Sagan

A muchas personas se les va la vida intentando comprender los cómo y los por qué de las infinitas maravillas que existen en nuestro planeta y fuera de él, proponiendo modelos matemáticos o teorías que satisfagan su curiosidad y la de otros. Esta pasión por lo que denominamos *ciencia*, nos hace sentir inteligentes y orgullosos de nuestro trabajo, hasta el punto de no reconocer muchas veces la existencia de un Ser Superior quien ha diseñado de manera tan perfecta las leyes físicas y todo cuanto vemos en nuestro entorno. A tí Jehová, gracias por la vida y por habernos permitido lograr nuestras metas en este trabajo. Por darme la perseverancia necesaria para no flaquear cuando las cosas se complican, sobre todo en esas madrugadas de cansancio, hambre y sueño.

A la persona que más quiero con toda mis fuerzas, mi vieja, quien sin importar los sacrificios que hemos pasado juntos, siempre me ha demostrado cuánto significa para ella y las cosas que está dispuesta a hacer por mí. Por perdonar mis errores, por tu disposición a dar todo para vernos felices, por estar siempre pendiente de que no me falte nada, por enojarte cuando quieres que apague la computadora y vaya a dormir, por enseñarme las cosas de la vida... gracias por ser mi mamá.

Por si fuera poco, tengo dos hermanas que son mucho más que eso. Morela y Leticia, a ustedes tampoco tendré nunca una forma de pagarles las cosas que han hecho por mí y el apoyo que siempre me han dado en medio de tantas circunstancias. Todo lo que soy se lo debo principalmente a tres mujeres... ustedes.

De las mejores cosas que me pasaron durante mi recorrido en la universidad, indudablemente está el haber compartido con personas maravillosas. Patty, tienes un corazón que no cabe en el universo; gracias por estar siempre dispuesta a soportarme, a escucharme, a aconsejarme, a perdonarme, a apoyarme... te quiero mucho.

Afortunadamente contamos siempre con la simpatía de personas que han creído mucho en nosotros, en nuestro trabajo y en lo que podemos hacer. Entre ellos, sin duda alguna, nuestro tutor Carlos Cadenas... gracias por tu disposición casi incondicional, por soportar que nunca fuéramos tan puntuales en las horas de consulta, por estudiar siempre más para explicarnos mejor, por tomarnos en cuenta para las cosas importantes que quieres continuar desarrollando, por demostrarnos que valió la pena seguir haciendo esto a pesar de las tentadoras ofertas de trabajo que rechazamos porque nos obligaban a mudarnos lejos; gracias por confiar en nosotros y por permitir que los recursos del departamento de matemáticas estuvieran más disponibles que los de nuestro propio departamento. A Germán, por hacernos reír con las frases y ocurrencias que siempre tienes a la mano, por las críticas sanas contra nuestro código y por querer mostrar a todos con orgullo nuestro trabajo. Al profesor Joel por ser todo un profesor, por su vocación enseñando aspectos importantes de las matemáticas computacionales.

A los profesores Jaqueline y Amadís, infinitamente gracias por no vernos como simples grupos de alumnos y preocuparse personalmente por los problemas y cosas que nos afectan. A la profesora Reyna por habernos formado con sólidas bases algorítmicas y a la profesora Desireé por su atención y anécdotas profesionales. Profesora Rosa, gracias por su interés en sembrar la creatividad en cada uno de sus estudiantes.

Irónicamente, es imposible dejar de mencionar a la gran mayoría de profesoras adscritas a nuestro departamento de computación, quienes en su fanatismo “metodoilógico” sólo consiguen exteriorizar una profunda ignorancia en las verdaderas *ciencias* computacionales. Sus criterios para la toma de decisiones, generalmente faltos de profesionalismo, ética, conocimiento y salpicados de resentimiento o frustraciones personales, afectan directa e indirectamente la formación y evaluación de los estudiantes, los cuales son capaces de percibir sin muchos inconvenientes tales deficiencias académicas. Aunque no de la mejor manera, nos han dado aún más fuerzas para continuar demostrando alrededor del mundo la calidad y nivel de nuestras investigaciones y desarrollos ya reconocidos muchas veces por la comunidad científica internacional, lo que quizás explique su inquina hacia los estudiantes y algunos otros profesores. Por eso, *gracias!*

Gracias de verdad a los compañeros de clase que siempre alegraron el camino, las ganas de seguir adelante, y que creían a veces que éramos los más locos por habernos metido en este paquete con los matemáticos... Laurita, Jonathan, Alnardo, Daniel, Luisiño, Víctor Esteller y José Angel. A Juancito, gracias por ahorrarnos los tropezones que ya venías dándote con los métodos. A Miguel, por tener más paciencia que yo en sacar y revisar los cientos de páginas de cuentas de este trabajo, por ser un amigo casi imposible de amotinar y por mantener siempre la serenidad en medio de las presiones.

A Ruberth por portarse siempre como todo un hermano mayor, a Wuanelyers y Manuel por las múltiples segundas y a todos los que han puesto un poco de sí para hacer de este trabajo un orgullo que no nos pertenece a nosotros, sino a ustedes...

Juan Rubén Medina Eliett

Agradecimiento

Es muy difícil agradecer a todas las personas que ayudaron a realizar este trabajo de investigación así que si falta alguien por favor discúlpeme y muy dentro de ustedes sabrán que se los voy a agradecer toda la vida.

Primero le doy gracias a Dios, por la fuerza de vida que hace posible que esté aquí, por no abandonarme en los momentos mas difíciles y siempre sentir que no estaba solo en esas madrugadas de trabajo. A mi madre Zulay, no tengo palabras para agradecerte, eres infinitamente importante para mí, por tu apoyo, tus consejos, tus regaños, todo hace que mejore como persona cada día, gracias mamá. A mi padre Miguel, por ser el mejor padre del mundo y estar siempre cuando te necesito, por transmitirme tranquilidad y confianza, te quiero muchísimo. A mis hermanos por su presencia y preocupación por tratar de comprender el trabajo que estoy haciendo. A mi abuelo por estar pendiente de mí después de esas largas horas de trabajo hasta la madrugada. A toda mi familia por su preocupación, adoro la familia que tengo. Al Prof. Cadenas por creer en nosotros desde el principio, por todo su apoyo, y esas largas horas para entender los métodos numéricos que estábamos aprendiendo, Gracias!!! ...ah, acuérdate que vamos a seguir trabajando!!!. A German por sus consejos en la programación, por las buenas críticas de nuestro trabajo y esos comentarios tan oportunos. A todos mis compañeros Patricia, Alnardo, Laurita, Fernando, Víctor, Bena, Carmirt, a todos, con ustedes compartí muchas cosas antes y durante la tesis. Al Prof. Joel por sus consejos, gracias por ser uno de los mejores profesores con que cuenta la facultad. A la Prof. Jacqueline que es fuente de inspiración para la mayoría de los estudiantes. Al Prof. Amadis por su colaboración y estímulo durante toda nuestra

carrera. A Juancito Rodríguez, chamo gracias!!!, porque al principio no tenía ni idea de los métodos y tú me aclaraste dudas básicas y luego todos los resultados fueron exitosos. A mi compañero de tesis Juan, por todo el esfuerzo y empeño en hacer de nuestra investigación un mejor trabajo cada día. A mi novia Saulymar por estar a mi lado estos 5 años de estudios, animándome y acompañándome en la buenas y en las malas, no me imagino este pequeño recorrido de mi vida sin tenerte a mi lado. Gracias!!!

Garcias a todos!!!

Miguel Omar Fagúndez Córcega

Índice general

1. PLANTEAMIENTO DEL PROBLEMA	1
1.1. Planteamiento del Problema	1
1.2. Objetivo General y Específicos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	5
1.3. Justificación	6
1.4. Antecedentes	8
2. MARCO TEÓRICO	12
2.1. Conceptos Básicos	13
2.1.1. Mecánica de fluidos	13
2.1.2. Ecuaciones Diferenciales y Condiciones de Frontera	17
2.1.3. Operadores de Divergencia, Gradiente y Laplaciano	19

2.2. Métodos de Solución Numérica	19
2.2.1. Diferencia Finita Simple	21
2.2.2. Volumen Finito	22
2.2.3. Elemento Finito	22
2.2.4. Diferencia Finita Mimética	24
2.3. Generación de Mallados	25
2.3.1. Mallados en 1D	25
2.3.2. Mallados en 2D	28
2.4. Método de Operadores de Soporte 1-2-1	30
2.5. Método Castillo-Grone 2-2-2	34
2.5.1. Operadores discretos de segundo orden	36
2.6. Método Implícito de Crank-Nicolson	37
2.7. Condiciones de Frontera en CFD	37
2.8. Métodos de Compresibilidad Artificial	39
2.9. Resolución de Sistemas de Ecuaciones Lineales	40
2.9.1. Uso de la librería UCSparseLib	41
2.10. OpenGL como API para la Visualización	45

2.11. Propuesta metodológica para el desarrollo del Software	47
2.11.1. Planificación	48
2.11.2. Diseño	48
2.11.3. Desarrollo	49
2.11.4. Pruebas	50
3. DESARROLLO	51
3.1. Solución Numérica de las Ecuaciones de Navier-Stokes y Compresibilidad Artificial en 1D	52
3.1.1. Operadores de Soporte para Mallados Uniformes. Método Explícito	52
3.1.2. Operadores de Soporte para Mallados Uniformes. Método Implícito	56
3.1.3. Método Castillo-Grone para Mallados Uniformes. Método Explícito 2-2-2	63
3.1.4. Método Castillo-Grone para Mallados Uniformes. Método Implícito 2-2-2	64
3.2. Solución Numérica de las Ecuaciones de Navier-Stokes y Compresibilidad Artificial en 2D	65

3.2.1. Método Operadores de Soporte para Mallados Uniformes. Método Implícito 1-2-1	65
3.3. Implementación de las subrutinas de la Simulación Numérica . . .	91
3.3.1. Pre-procesador	91
3.3.2. Post-procesador	92
3.3.3. Estructuras de Datos	93
3.3.4. Prototipos de las Funciones y Procedimientos principales .	94
3.4. Otros desarrollos	96
3.5. Visualización	96
3.5.1. Bibliotecas Involucradas	97
4. RESULTADOS Y CONCLUSIONES	98
4.1. Resultados Numéricos	98
4.1.1. Pruebas en 1D	99
4.1.2. Pruebas en 2D	101
4.2. Conclusiones y Recomendaciones	105
4.3. Trabajos Futuros	107
A. Código Fuente de la Implementación	111

Introducción

El control de flujo de fluidos ha emergido recientemente como un importante área de investigación científica y tecnológica. Esto se debe a la amplia gama de importantes aplicaciones en las que es extremadamente útil tener un conocimiento apropiado de la mecánica de fluidos. En biomecánica el flujo de sangre y fluido cerebral son de particular interés; en meteorología e ingeniería oceánica, para entender el movimiento del aire y las corrientes oceánicas, se requiere del conocimiento de la mecánica de fluidos; los ingenieros químicos deben comprender la mecánica de fluidos para diseñar los diferentes equipos de procesamiento químico; los ingenieros aeronáuticos utilizan su conocimiento de fluido para incrementar al máximo la fuerza de elevación, reducir al mínimo el retardo de aeronaves y para diseñar motores de reacción; los ingenieros mecánicos diseñan bombas, turbinas, motores de combustión interna, compresores de aire, equipos de aire acondicionado, para el control de contaminación y plantas eléctricas con base en el conocimiento apropiado de la mecánica de fluidos; y los ingenieros civiles también utilizan los resultados obtenidos en el estudio de mecánica de fluidos para comprender el transporte de sedimentos y la erosión en ríos, la contaminación del aire y agua, y así diseñar sistemas de tuberías, plantas de tratamiento de aguas negras, canales de irrigación, sistemas de control de inundaciones, presas y estadios deportivos cubiertos.

Surge entonces la simulación computacional de la dinámica de fluidos (*CFD*, por sus siglas en inglés) como una herramienta para la evaluación e investigación de situaciones como estas, sin necesidad de realizarlas, lo que ha llevado a muchos investigadores a dedicarse a resolver los problemas de orden numérico para cumplir con estos objetivos, logrando así reducir costos en tiempo, dinero y esfuerzo.

Desde hace algunas décadas, se ha trabajado en el desarrollo de teorías y métodos numéricos que son de gran importancia en la resolución de ecuaciones diferenciales de primero y segundo orden, los cuales permiten obtener aproximaciones cada vez más exactas a fin de modelar algún proceso dentro de un entorno controlado. Nuevas técnicas numéricas, como los Métodos de Diferencias Finitas Miméticas, son evaluadas con el fin de brindar a los investigadores la facilidad de resolver problemas en menor tiempo y reducir los costos de muchas pruebas en comparación al desarrollo manual. En estas técnicas creadas recientemente, los operadores discretos mantienen las propiedades conservativas de los operadores continuos de divergencia, gradiente y laplaciano. Aún están siendo probadas a diferentes niveles desde el punto de vista conceptual, y es por ello que esta investigación estaría colocando precedentes tanto en el área numérica, como en el campo de aplicación al cual está siendo enfocada.

El deseo de simular procesos reales, entre ellos los de una refinería de crudo pesado (petróleo), hace muy atractivo este tipo de investigación, ya que la visualización del comportamiento de los fluidos dentro de la refinería es un gran paso en el diseño de proyectos de esa índole; también por el uso de múltiples métodos para resolver las ecuaciones diferenciales que se involucran en este tipo de problemas.

En el primer capítulo se da a conocer el problema a resolver, los objetivos que

se persiguen, la justificación y los antecedentes en las áreas involucradas. En el capítulo dos se introducen algunos conceptos básicos de la dinámica de fluidos computacional y las ecuaciones diferenciales implicadas. Luego se exponen algunas de las formulaciones más importantes para resolver ecuaciones diferenciales. El análisis que se hace de los métodos es en términos de los procedimientos para obtener una solución aproximada, por lo que está orientado al cómo se construye la formulación y no en el por qué la formulación funciona, ya que ello escapa de los alcances del presente trabajo. Es por ello que los conceptos matemáticos se mantendrán lo más simples posible. En el capítulo tres se hace el desarrollo numérico de las ecuaciones planteadas en el capítulo uno, en términos de las formulaciones miméticas. También se introduce la estructura y funcionamiento de los métodos iterativos para la resolución de sistemas de ecuaciones implantados en la biblioteca UCSparcelib, usada como marco de trabajo en el desarrollo del software. A continuación, se presenta la implementación de las funciones necesarias, explicando las estructuras de datos creadas y los prototipos de las principales funciones. En el capítulo cuatro se muestran los resultados de las pruebas realizadas, así como también las conclusiones, recomendaciones y posibles trabajos futuros. En el apéndice A está el código fuente de parte de la implementación realizada.

Como un aspecto de notación, todas las variables escalares y vectores o matrices involucrados, se expresan en letras itálicas (ie: x , u , y) y letra negrilla (ie: \mathbf{u} , \mathbf{f} , \mathbf{A}) respectivamente, mientras que para aquellas variables o instrucciones involucradas en la implementación computacional se escribirán con letra tipo (i.e: `ii`, `TMatrix`)

Capítulo 1

PLANTEAMIENTO DEL PROBLEMA

1.1. Planteamiento del Problema

Muchos algoritmos de simulación involucran necesariamente la solución numérica de ecuaciones diferenciales ordinarias y en derivadas parciales. La mayoría de ellas pueden ser formuladas utilizando operadores diferenciales invariantes de primer orden, tales como la *divergencia* y el *gradiente*. Por otro lado, es posible construir muchos esquemas de diferencias finitas miméticos en base a los operadores discretos de DIVERGENCIA Y GRADIENTE, pero en la mayoría de los casos los operadores discretos clásicos no mantienen algunas propiedades físicas de los operadores continuos. Por ello ha surgido la necesidad de desarrollar métodos que permitan obtener tales operadores y que satisfagan las leyes de conservación, simetría y los teoremas tradicionales del cálculo vectorial, como el teorema de Green, entre otros.

A finales del año 2002 y a comienzos del presente año, se han creado técnicas para

generar operadores discretos miméticos tanto de divergencia como de gradiente. Debido a la novedad de esta técnica, la misma no ha sido aplicada a muchos de los problemas clásicos de la mecánica continua y tampoco se han hecho comparaciones exhaustivas con otros métodos miméticos, como por ejemplo el método de Operadores de Soporte 1-2-1, el cual tiene una base teórica sólida. Esto ha incentivado la creación de aplicaciones diversas en diferentes campos de la ciencia.

En este trabajo se pretende simular numéricamente el flujo de fluido basado en un sistema de ecuaciones diferenciales conocido como ecuaciones de *Navier - Stokes* para el fluido real no estacionario, viscoso e incompresible:

$$\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \right) \quad (1.1)$$

$$\frac{\partial v_y}{\partial t} + v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} \right) \quad (1.2)$$

donde: ν : es la viscosidad cinemática del fluido (constante)

ρ : es la densidad del fluido (constante)

p : es la presión del fluido en (x, y)

t : es el tiempo

v : es el vector velocidad

$(x, y) \in \Omega \subset \mathbb{R}^2$ variables dependientes

Lo anterior se puede escribir en función de los operadores continuos de divergencia, gradiente y laplaciano, como:

$$\frac{Dv}{\partial t} = -\frac{1}{\rho}\nabla p + \nu\nabla^2 v \quad \text{en } \Omega \quad (1.3)$$

En donde :

$$\begin{aligned} \nabla^2 v &= (\Delta v_x, \Delta v_y)^t \\ \nabla p &= \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y} \right)^t \\ \frac{Dv}{\partial t} &= \frac{\partial v}{\partial t} + v \cdot \nabla v \\ \nabla v &= \left(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \right)^t \\ v &= (v_x, v_y)^t \\ \Delta a &= \frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2} \quad a : \mathbb{R}^2 \rightarrow \mathbb{R} \end{aligned}$$

En (1.1) y (1.2) se plantearon las ecuaciones de *Navier - Stokes* para el flujo de fluido incompresible en 2D, pero es necesario definir otra ecuación (ecuación de continuidad) que teóricamente garantice la incompresibilidad, que permita completar el sistema y que este tenga solución única:

$$\nabla \cdot v = 0 \quad \text{en } \Omega \quad (1.4)$$

$$(1.5)$$

Al tratar numéricamente este sistema de ecuaciones y realizando una discretización del dominio, se obtiene un conjunto de n ecuaciones lineales con n incógnitas, representado por la forma:

$$\mathbf{Ax}=\mathbf{b}$$

donde: $\mathbf{A} \in \mathbb{R}^{n \times n}$ es la matriz de coeficientes

$\mathbf{x} \in \mathbb{R}^n$ es el vector de incógnitas

$\mathbf{b} \in \mathbb{R}^n$ es el vector de términos independientes

Para hallar una solución numérica al sistema de ecuaciones en derivadas parciales, se opta por el Método de Operadores de Soporte 1-2-1 [1] y el Método Castillo-Grone 2-2-2 [4] para construir operadores discretos miméticos. Al surgir un sistema de ecuaciones lineales, el mismo debe ser resuelto por alguna biblioteca numérica. De esta manera se obtienen los resultados que permitan evaluar las ventajas y desventajas de los métodos planteados. Posteriormente, es deseable poder visualizar de alguna manera el comportamiento dinámico de la simulación a través de su progreso en el tiempo, lo que exige la utilización de técnicas de visualización científica. En este trabajo se utiliza la biblioteca numérica UCSparseLib como herramienta de desarrollo para el programa de simulación y OpenGL como herramienta en la creación del software de visualización.

1.2. Objetivo General y Específicos

1.2.1. Objetivo General

Desarrollar módulos de software capaces de simular numéricamente el flujo de fluido viscoso, incompresible y no estacionario dentro de un dominio físico 1D y 2D dadas por (1.1), (1.2) y (1.4).

1.2.2. Objetivos Específicos

1. Estudiar el Método de Operadores de Soporte y Método Castillo-Grone para construir operadores discretos miméticos, a través de una revisión bibliográfica detallada en las diferentes áreas de investigación y así obtener herramientas numéricas necesarias en la resolución del problema planteado.
2. Estudiar y comprender la biblioteca UCSparseLib, elegida como herramienta de trabajo por su disponibilidad de código abierto y eficiencia en la resolución de sistemas de ecuaciones lineales.
3. Diseñar módulos de software para UCSparseLib que permitan resolver el sistema de ecuaciones diferenciales parciales, aplicando tanto el Método de Operadores de Soporte 1-2-1, como el Método Castillo-Grone 2-2-2 para construir operadores discretos miméticos.
4. Diseñar programas para la simulación numérica de flujo de fluido haciendo uso de los conocimientos y elementos ya considerados y de las condiciones físicas planteadas.
5. Implementar (3) y (4) de manera eficiente y con filosofía multiplataforma para ser acoplado a la biblioteca UCSparseLib.
6. Diseñar aplicaciones que permitan realizar un análisis comparativo entre los diferentes métodos para construir operadores discretos miméticos.
7. Implementar el diseño creado en (6), con el propósito de comprobar las ventajas y desventajas numéricas de cada método.

8. Diseñar un post-procesador que permita visualizar el comportamiento dinámico de la simulación.
9. Implementar el diseño desarrollado en (8) haciendo uso de OpenGL como biblioteca para la creación y manipulación de gráficos.

1.3. Justificación

La simulación como una herramienta para la evaluación e investigación de situaciones de la vida real sin necesidad de realizarlas, ha llevado a que muchos investigadores se dediquen a resolver los problemas de orden numérico para cumplir con estos objetivos, logrando así reducir costos en tiempo, dinero y esfuerzo. Es por ello que desde hace algunas décadas, se ha trabajado en el desarrollo de teorías y métodos numéricos que son de gran importancia en la resolución de ecuaciones diferenciales de primer y segundo orden, las cuales permiten obtener aproximaciones cada vez más exactas a fin de modelar algún proceso dentro de un entorno controlado.

El deseo de simular procesos reales, entre ellos los de una refinería de crudo pesado (petróleo), hace muy atractivo este tipo de investigación, ya que la visualización del comportamiento de los fluidos dentro de la refinería es un gran paso en el diseño de proyectos de esa índole.

Surge entonces la oportunidad de evaluar nuevas técnicas numéricas como los Métodos de Diferencias Finitas Miméticas, con el fin de brindar a los investigadores la facilidad de resolver problemas en menor tiempo, obtener

soluciones más precisas y reducir los costos de muchas pruebas en comparación al desarrollo manual. En estas técnicas creadas recientemente, los operadores discretos mantienen las propiedades conservativas de los operadores continuos de divergencia, gradiente y laplaciano. Aún está siendo probada a diferentes niveles desde el punto de vista conceptual y de aplicación, con ecuaciones diferenciales diversas que permitan determinar sus ventajas y desventajas. Esta investigación estará orientada a un dominio físico 1D y 2D, ya que el método de Diferencias Finitas Miméticas propuesto para resolver las ecuaciones diferenciales de *Navier-Stokes*, no posee un desarrollo completo en 3D. Es por ello que esta investigación estaría colocando precedentes tanto en el área numérica, como en el campo de aplicación al cual está siendo enfocada.

Dentro de un paquete de simulación, el tiempo es de gran interés por parte de los investigadores. Debido a que la resolución numérica de cada sistema de ecuaciones lineales asociado a un paso de tiempo de un problema puede tomar de minutos a horas de cómputo, el proceso completo para una simulación podría tardar de semanas a meses, haciendo que los resultados no se obtengan en el momento deseado. Para mejorar el tiempo que consume resolver los sistemas lineales, se emplean bibliotecas de software con métodos de resolución eficientes.

La representación gráfica de tales resultados, permite comprender visualmente el comportamiento dinámico de la simulación. Para ello es necesario considerar el uso de OpenGL como la más importante biblioteca para hardware gráfico existente en la actualidad, lo que ofrece mayor soporte, portabilidad y excelentes tiempos de respuesta en este aspecto de la simulación.

1.4. Antecedentes

Como se ha mencionado en la sección anterior, el interés de este trabajo es la Simulación Numérica de Flujo de Fluido, la cual es una área particular de la Simulación Numérica en Ingeniería. En esta sección, se presenta una breve descripción de algunos de los antecedentes en las áreas de interés. Sólo se hace referencia a publicaciones recientes, ya que éstas a su vez hacen cita de trabajos previos y clásicos de gran importancia en la Simulación Numérica.

En Enero de 1998 J.M. Hyman and M. Shashkov [2] presentan un estudio que describe cómo incorporar las condiciones de frontera para los Métodos de Diferencias Finitas Miméticos, para que así las aproximaciones resultantes imiten las identidades de operadores diferenciales del cálculo vectorial y tensorial. La aproximación es válida para una gran clase de ecuaciones diferenciales parciales y se describe la ecuación de Poisson con las condiciones de frontera tipo Dirichlet, Neumann y Robin.

En 1998, B. Jiang [5], demuestra cómo utilizar el Método de Elementos Finitos de Mínimos Cuadrados basado en la formulación velocidad-presión-vorticidad de primer orden, con el fin de resolver varios problemas relacionados con flujo viscoso incompresible, entre ellos una adaptación rigurosa de las ecuaciones de *Navier-Stokes* y sus condiciones de frontera adicionales.

En [6] J. Stam propone por primera vez un modelo incondicionalmente estable para producir flujos de fluidos complejos que permite la interacción de los mismos con otros elementos en un escenario de simulación en 2D y 3D. Esto lo logra a

través del modelo matemático de *Navier-Stokes* y un uso eficiente de técnicas de graficación por computadora.

Posteriormente en [7] J. Stam presenta la implementación de un *solver* para fluidos, consistente con las ecuaciones de flujo de fluidos que producen campos de velocidad, que contienen estructuras rotacionales incompresibles y que reaccionan dinámicamente a fuerzas ejercidas por el usuario. Este *solver* especializado en fluidos que cubren el espacio, permite tomar ventaja de la Transformada de Fourier, lo que simplifica considerablemente muchos aspectos del mismo y puede ser usado como primitiva de movimientos básicos en diferentes aplicaciones en computación gráfica.

Para diciembre del año 2002, P. Valera [8] presenta su trabajo especial de grado, en el que desarrolla un conjunto de estructuras de datos y subrutinas implementadas en el lenguaje de programación C++ para simular la deformación de tejido blando simple (venas). Utiliza un modelo para deformaciones lineales sobre un material isotrópico homogéneo, donde se aproxima la geometría a un conjunto finito de elementos y se calcula la fuerza ejercida sobre cada partícula y sus vecinas a través del método de los elementos finitos, para después simular su evolución usando modelos incrementales iterativos. Cada subrutina implementada está basada en el enfoque de la API *Virtual Vision Machine* (VVM) para generar las visualizaciones.

Para el 2003 G. Larrazabal, J. Castillo y C.R. Torres en [10] presentan un trabajo en el área de la Dinámica de Fluidos, en donde se desarrolla un algoritmo eficiente y robusto para la simulación numérica del flujo estratificado 2D. El algoritmo

se prueba simulando el flujo en una placa descendente infinito en el piso con un flujo uniforme. El programa desarrollado permite alcanzar buenas *aproximaciones* en pocas iteraciones para una alta gama de números de Reynolds. La forma de resolver el sistema de ecuaciones lineales dispersos es mediante la biblioteca *UCSparseLib*.

Como es bien conocido, la etapa de procesamiento es la que consume el mayor tiempo de CPU en una simulación numérica como las descritas en este trabajo. Por ello, es importante mencionar que la tarea de resolver sistemas lineales dispersos de ecuaciones es el núcleo computacional de esta etapa. Las rutinas para resolver dichos sistemas deben ser una perfecta caja negra, es decir, que no dependan de la geometría del problema ni del tipo de discretización utilizada. Además, estas rutinas deben contener diferentes métodos de solución, debido a que no existe un método general para resolver sistemas lineales dispersos de ecuaciones. En la actualidad es posible hacer uso de distintas bibliotecas numéricas capaces de resolver sistemas lineales de gran magnitud, entre ellas:

- PECTs (*Portable, Extensible Toolkit for Scientific Computation*), es un conjunto de herramientas de manejo de datos para solución de problemas científicos de gran escala. Está diseñada para resolver problemas que involucren sistemas de ecuaciones en derivadas parciales, ha sido desarrollada en ANSI C y puede ser utilizada con otros lenguajes como Fortran.
- *The NIST Sparse BLAS (National Institute of Standards and Technology Sparse Basic Linear Algebra Subprogram)* es un conjunto de subrutinas que proveen un núcleo computacional para operaciones fundamentales con matrices esparcidas. Está desarrollada en ANSI C y pueden ser llamadas

desde Fortran a través del *Sparse BLAS Toolkit*.

- *SuperLU* es una biblioteca de propósitos generales para la solución directa de sistemas de ecuaciones lineales. Está desarrollada en ANSI C y puede ser llamada desde Fortran.

Es claro que existen muchas bibliotecas numéricas y que son de dominio público. Sin embargo, para desarrollar este trabajo será utilizada la biblioteca *UCSparseLib*, la cual ha sido creada por el Dr. Germán Larrazabal [9] en la Universidad de Carabobo, y cuya finalidad es la resolución de sistemas lineales densos y esparcidos. Está desarrollada en ANSI C y debido a la disponibilidad del código fuente y asesorías, será una importante herramienta en el desarrollo de este trabajo.

Capítulo 2

MARCO TEÓRICO

En este capítulo se introducen algunos conceptos básicos de la dinámica de fluidos computacional, los fundamentos de las ecuaciones gobernantes, y algunos modelos clásicos de ello. Luego, se mostrarán las formulaciones de los métodos de diferencias finitas miméticas, que permiten aproximar la solución haciendo una descomposición de las ecuaciones diferenciales de segundo orden en un sistema de ecuaciones lineales. Posteriormente se analizan algunos métodos de discretización del dominio continuo (mallados), el uso de las bibliotecas UCSparselib y OpenGL, sobre las cuales se basa la implementación propuesta, y finalmente es presentada la metodología seguida en el diseño y desarrollo de los productos de software resultantes de esta investigación.

2.1. Conceptos Básicos

2.1.1. Mecánica de fluidos

Los *fluidos* se definen como todo material que no sea sólido y que puede “fluir”. Son fluidos los líquidos y los gases; aún con sus grandes diferencias, sus comportamientos como fluidos se describe con las mismas ecuaciones básicas. La diferencia entre uno y otro está en su compresibilidad. Para cualquier sustancia, el estado líquido existe a una temperatura mayor que la del estado sólido, debido a que tiene una mayor agitación térmica y las fuerzas moleculares no son suficientes para mantener a las mismas en posiciones fijas. Lo común que tienen con los sólidos es que si actúan fuerzas externas de compresión, surgen grandes fuerzas atómicas que se resisten. En el estado gaseoso las moléculas tienen un continuo movimiento al azar y ejercen fuerzas muy débiles unas con otras y la separación promedio entre las moléculas de un gas, es mucho más grande que las dimensiones de las mismas.

El estudio de la dinámica de fluidos es similar al estudio clásico de la dinámica de sólidos, en la que se estudia el movimiento bajo la acción de fuerzas aplicadas. Se hace uso de los mismos principios de: conservación de masa, conservación del momentum y conservación de la energía termodinámica. Las ecuaciones de movimiento son dinámicas y las ecuaciones de continuidad son cinemáticas.

Propiedades de los fluidos

Entre las principales propiedades relacionadas con los fluidos y analizadas en el desarrollo de este trabajo de investigación, están:

- Densidad: Es la cantidad de masa por unidad de volumen.
- Peso específico: Es el peso por unidad de volumen.
- Viscosidad: Se refiere a la pegajosidad interna de un fluido. Es una de las propiedades que influye en la potencia necesaria para mover una superficie aerodinámica a través de la atmósfera. La velocidad de deformación de un fluido está directamente ligada a su viscosidad.
- Compresibilidad: Es la deformación provocada en el fluido por cambios de presión, lo que da como resultado un incremento de densidad.
- Tension superficial: es una propiedad originada por las fuerzas de atracción entre las moléculas.

Movimientos de fluido

Los movimientos de fluido se manifiestan de diferentes maneras. Algunos pueden ser descritos con facilidad, en tanto que otros requieren de un conocimiento completo de las leyes de física. En aplicaciones de ingeniería es importante describir los movimientos de fluidos tan simplemente como puedan ser justificados. Esto en general depende de la precisión requerida y de las suposiciones simplificadoras de las ecuaciones de movimiento, las cuales son muy difíciles de

resolver. Algunas suposiciones comunes utilizadas para simplificar una situación de flujo tienen que ver con las propiedades del fluido. Por ejemplo, en ciertas condiciones, la viscosidad puede afectar el flujo de manera significativa; en otras, los efectos viscosos pueden ser omitidos, con lo que se simplifican en gran medida las ecuaciones sin que se alteren significativamente las predicciones. La compresibilidad de un gas en movimiento deberá ser tomada en cuenta si las velocidades son muy altas, mas los efectos de compresibilidad no tienen que ser tomados en cuenta para predecir las fuerzas del viento que actúan en edificios o para predecir cualquier otra cantidad física que sea un efecto directo del viento. Después de estudiar los movimientos de fluidos, las suposiciones apropiadas utilizadas deberán ser más que obvias. El análisis de problemas de flujo de fluidos complejo a menudo se simplifica mediante la visualización de patrones de flujo, los que permiten desarrollar un mayor entendimiento intuitivo y ayudan a formular el problema matemático. En este trabajo se considera la simulación de los llamados fluidos Newtonianos, los cuales comprenden la gran mayoría, entre ellos el agua, aire y aceite. La principal característica de este tipo de fluidos es que el esfuerzo cortante es proporcional al gradiente de velocidad.

Clasificación de los flujos de fluido

- Flujos viscosos e invíscidos: Un flujo puede ser clasificado de una manera general como flujo viscoso o flujo invíscido. Un flujo invíscido es aquel en el que los efectos viscosos no influyen significativamente en el flujo y por lo tanto son ignorados. En un flujo viscoso los efectos de viscosidad son importantes y no pueden ser ignorados. Para modelar un flujo invíscido analíticamente, simplemente la viscosidad se hace cero; esto obviamente

hace que todos los efectos viscosos sean cero. Con base en la experiencia, se encuentra que la clase principal de flujos que pueden ser modelados como flujos invíscidos, son los flujos externos, es decir, los flujos que existen en el exterior de un cuerpo, tales como el flujo alrededor de una superficie aerodinámica o una superficie hidrodinámica. Los flujos viscosos incluyen la amplia clase de flujos internos, tales como flujos en tubos y conductos y en canales abiertos. En flujos como esos los efectos viscosos provocan pérdidas sustanciales y responden a las inmensas cantidades de energía que deben ser utilizadas para transportar petróleo y gas por oleoductos. La condición no deslizante que produce una velocidad cero en la pared, y los esfuerzos cortantes resultantes, conducen directamente a estas pérdidas.

- Flujo laminar: En un flujo laminar el fluido fluye sin mezclado significativo de sus partículas próximas entre sí. El flujo es ordenado y predecible, el movimiento se produce en capas o láminas y las soluciones matemáticas son factibles. En este flujo las partículas se mueven en trayectorias independientes de las partículas de capas adyacentes.
- Flujo turbulento: Los movimientos del fluido varían irregularmente de tal suerte que las cantidades, tales como velocidad y presión, muestran una variación aleatoria con el tiempo y las coordenadas espaciales. Las cantidades físicas con frecuencia se describen mediante promedios estadísticos. En este sentido, un flujo turbulento “continuo” puede ser definido como un flujo en el que las cantidades físicas promedio dependen del tiempo y no cambian con éste.

2.1.2. Ecuaciones Diferenciales y Condiciones de Frontera

La mayoría de los fenómenos naturales son descritos por modelos matemáticos. Entre estos, las ecuaciones diferenciales son muy frecuentes y existen muchos tipos de ellas. En general, una ecuación diferencial puede escribirse como

$$A(u) = f \quad \text{en } \Omega, \quad (2.1)$$

donde A es un operador diferencial, u y f son funciones suficientemente diferenciables para satisfacer las restricciones del operador A , y u se conoce como *función solución* o *variable dependiente*; por lo que A puede verse como una función que depende de u . El operador A se dice que es lineal en su argumento u , sí y solo sí se cumple que:

$$A(\alpha u + \beta v) = \alpha A(u) + \beta A(v). \quad (2.2)$$

para todo escalar α, β , y funciones u, v . Si $f \equiv 0$ en (2.1), se dice que la ecuación es *homogénea*, en caso contrario, *no homogénea*. Además, Ω en (2.1) se conoce como el *dominio* de la ecuación diferencial, y para este contexto siempre se definirá real.

Cuando la variable dependiente es función de una variable, el dominio es un segmento de recta y su frontera son los dos puntos extremos. Cuando la variable dependiente es función de dos variables, el dominio es, comúnmente, una región del plano y su frontera es la curva que encierra (o delimita) la región.

Para garantizar la unicidad de la solución de la ecuación diferencial, es necesario imponer algunas condiciones sobre la variable dependiente. Cuando estas restricciones se aplican a los valores de la función solución y/o a sus derivadas en

los puntos de la frontera, se conoce como un *problema de valor de frontera*. Las condiciones de frontera se denotan como:

$$B(u) = g \quad \text{sobre } \Gamma, \quad (2.3)$$

donde B es un operador diferencial, u la variable dependiente y g es una función definida en la frontera Γ . Dependiendo de cómo se formulan las condiciones de frontera se pueden tener diferentes tipos. Cuando se condiciona únicamente el valor de la variable dependiente, se conoce como condición de *Dirichlet*, y tiene la forma:

$$hu = g \quad \text{sobre } \Gamma. \quad (2.4)$$

donde h es una función definida en Γ .

Cuando la condición prescribe el valor de la derivada de la variable dependiente, se conoce como condición de *Newmann*:

$$a \frac{\partial u}{\partial \vec{n}} = g \quad \text{sobre } \Gamma, \quad (2.5)$$

donde \vec{n} es el vector normal unitario externo a la frontera y a una función definida en Γ .

La condición donde se involucra tanto la variable dependiente como su derivada, se conoce como condición de *Robin*, *Newmann generalizada* o *mixta*, y es de la forma:

$$a \frac{\partial u}{\partial \vec{n}} + hu = g \quad \text{sobre } \Gamma \quad (2.6)$$

Si en cualquiera de los casos, $g \equiv 0$, se denomina condición *homogénea*, en otro caso se denomina *no homogénea*.

2.1.3. Operadores de Divergencia, Gradiente y Laplaciano

Entre los operadores más comunes, está el operador ∇ denominado *nabla*, el cual permite obtener la Divergencia, el Gradiente y Laplaciano de funciones escalares o vectoriales según sea el caso. Cada una de estas operaciones se define como:

Gradiente:

$$\nabla u = \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} \quad (2.7)$$

Divergencia:

$$\nabla \cdot \vec{A} = \frac{\partial AX}{\partial x} + \frac{\partial AY}{\partial y} \quad (2.8)$$

Laplaciano:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.9)$$

2.2. Métodos de Solución Numérica

Como ya se ha mencionado, un número significativo de problemas físicos y de ingeniería, incluyen ecuaciones en derivadas parciales. Es posible obtener soluciones analíticas de tales ecuaciones físico-matemáticas sólo en casos especiales, o bien puede ser extremadamente laborioso. Como una alternativa a ello, surgen los métodos numéricos para poder obtener soluciones aproximadas de la ecuación diferencial.

Las funciones incógnitas en ecuaciones diferenciales, son funciones de argumentos

continuos. Por ejemplo, la función $u(x, y, z, t)$ de la ecuación de calor, retorna el valor de temperatura en cada punto del espacio en el tiempo t . Esto significa que en el momento de tiempo t , puede asociarse a cada punto con coordenadas (x, y, z) un valor de temperatura $u(x, y, z, t)$. En el caso más simple, 1D estacionario, u es la función de un argumento espacial $x = x_1$, $u(x)$ y $x \in [a, b]$. La función $u(x)$ de argumento continuo $x \in [a, b]$, es elemento de un espacio de funciones H , y alguna norma es utilizada para comparar dos funciones $u(x), v(x) \in H$. Por ejemplo:

$$\|u\|_C = \max_{a \leq x \leq b} |u(x)| \quad (2.10)$$

$$\|u\|_{L_2} = \left(\int_a^b u^2(x) dx \right)^{1/2} \quad (2.11)$$

La escogencia de la norma para la comparación de estas funciones, depende de la interpretación física de la función u .

Para comprender lo que significa la *función de argumentos discretos*, se puede considerar el proceso de medición en un punto del espacio por un instrumento. Como resultado de tales mediciones, se obtiene un conjunto de números, cada uno de los cuales corresponde al valor en un punto. Estos valores pueden ser enumerados de alguna manera, por ejemplo asignando un número al instrumento que toma cada medición, y puede ser denotado como u_i . En algunas aplicaciones físicas, los instrumentos pueden arrojar valores aproximados sobre un pequeño volumen o área, por lo que en este caso, los valores no corresponden a un punto en específico. Es claro que solo el índice i no representa información sobre la ubicación del instrumento con numeración i ; es por ello que la noción de función con argumentos discretos incluye información sobre la ubicación de los puntos, volúmenes u otra figuras geométricas y los valores relacionados a estos elementos

geométricos. Ejemplos de mallados y funciones de mallados son dados en la siguiente sección.

2.2.1. Diferencia Finita Simple

Este es el método más antiguo para hallar solución numérica de ecuaciones en derivadas parciales, introducido por Euler en el siglo XVIII. Es además muy fácil de aplicar en geometrías simples. En cada punto del mallado, la ecuación diferencial es aproximada reemplazando las derivadas parciales por aproximaciones en términos de los valores nodales de las funciones. Aunque en principio el método puede ser aplicado sobre cualquier tipo de mallado, es común que se desarrolle únicamente para mallados estructurados.

Por medio de expandir la serie de Taylor o interpolación polinomial, se obtienen aproximaciones para la primera y segunda derivada de las variables con respecto a las coordenadas.

Como ya se mencionó, en mallados estructurados, el método de diferencias finitas es simple y efectivo, y es especialmente sencillo obtener esquemas de orden alto en mallados regulares. La desventaja que presentan es que la conservación no es considerada a menos que se dedique un análisis adicional. Además, la restricción para sólo geometrías simples es una desventaja significativa en flujos complejos.

2.2.2. Volumen Finito

Este método hace uso de la forma integral de la ecuación de conservación como punto de inicio. El dominio de la solución es dividido en un número finito de volúmenes de control contiguos y las ecuaciones de conservación son aplicadas a cada volumen. Es posible adaptar el método a cualquier tipo de mallado, por lo que es apropiado para geometrías complejas. Posiblemente las aproximaciones de Volumen Finito sean las más sencillas de entender y programar, y todos los términos que necesitan ser aproximados tienen interpretación física, lo que lo hace muy popular en ingeniería.

La desventaja en comparación con los esquemas de Diferencias Finitas, es que el desarrollo para órdenes mayores a dos, es mucho más difícil de realizar en tres dimensiones. Esto se debe a que el enfoque de Volumen Finito requiere tres niveles de aproximación: interpolación, diferenciación e integración.

2.2.3. Elemento Finito

Es similar al método de Volumen Finito en muchas maneras. El dominio es fraccionado en un conjunto de volúmenes discretos o elementos finitos que están generalmente estructurados; en 2D es común el uso de triángulos o cuadriláteros, mientras que en 3D son tetraedros o hexaedros. La característica distintiva de este método es que las ecuaciones son multiplicadas por una *función de peso* antes de ser integrada en el dominio. En la forma más simple del método de Elementos Finitos, la solución es aproximada por una función polinomial en cada elemento,

de manera de garantizar continuidad de la solución a través de las fronteras de los elementos. Tal función puede ser construida a partir de los valores en las esquinas de los elementos.

Esta aproximación es entonces sustituida dentro de la integral de la ley de conservación y las ecuaciones a ser resueltas son obtenidas a través de la derivada de la integral con respecto a cada valor nodal igual a cero; esto corresponde a seleccionar la mejor solución dentro de un conjunto de funciones permitidas (la de residual mínimo). El resultado es un conjunto de ecuaciones algebraicas no lineales.

Una importante ventaja de este método es la capacidad de adaptarse a geometrías arbitrarias, y existe una extensa bibliografía dedicada a la construcción de mallados para métodos de elementos finitos. Los mallados son fácilmente refinados por medio de simplemente subdividir cada elemento. Además, los métodos de elementos finitos son fáciles de analizar matemáticamente y es posible demostrar sus propiedades de optimalidad para ciertos tipos de ecuaciones.

La principal desventaja, compartida con cualquier método que utilice mallados no uniformes, es que las matrices de las ecuaciones linealizadas no son estructuradas como aquellas en mallados uniformes, haciendo más difícil hallar métodos de soluciones eficientes.

2.2.4. Diferencia Finita Mimética

Entre los métodos más universales y efectivos, usado ampliamente en la actualidad para hallar soluciones aproximadas a ecuaciones físico-matemáticas, está el método de *diferencias finitas miméticas*, el cual resuelve la ecuación original reemplazando los operadores continuos de *divergencia*, *gradiente* y *laplaciano* por aproximaciones numéricas que conserven propiedades importantes de los operadores originales.

Las soluciones para las ecuaciones diferenciales son conseguidas en dos etapas:

- El desarrollo del esquema de diferencias finitas (una aproximación en diferencias para la ecuación diferencial sobre un mallado).
- El cálculo de la solución de las ecuaciones en diferencias, el cual es escrito en forma de un sistema de alto orden de ecuaciones algebraicas lineales o no lineales.

La esencia del método puede ser resumida como sigue:

1. El dominio continuo (por ejemplo, un intervalo, un rectángulo o un dominio con forma arbitraria), es reemplazado por un conjunto discreto de puntos (nodos).
2. En lugar de una función con argumentos continuos, se considera una con argumentos discretos. El valor de la función es definida en los nodos del

mallado o en otros elementos del mallado (por ejemplo, en celdas que tienen nodos en sus vértices) y es llamada *función de mallado*.

3. Los valores de derivadas dentro de la ecuación diferencial y en las condiciones de frontera, son aproximados por las expresiones en diferencias. Así el problema diferencial es transformado en un sistema de ecuaciones algebraicas lineales o no lineales, lo que constituye el *esquema de diferencias finitas*.

Debido a lo reciente de la creación de este método, aún está siendo probado a diferentes niveles, desde el punto de vista conceptual y de aplicación con ecuaciones diferenciales diversas que permitan determinar sus ventajas y desventajas. Esta investigación estará orientada a un dominio físico 1D y 2D, ya que el método de Diferencias Finitas Miméticas propuesto para resolver las ecuaciones diferenciales de *Navier-Stokes*, no posee un desarrollo completo en 3D.

2.3. Generación de Mallados

2.3.1. Mallados en 1D

Un ejemplo simple de mallado es el usado en el caso 1D. Sea el dominio de la variable x el intervalo $0 \leq x \leq 1$. Se definen en este intervalo M nodos x_i , $i = 1, \dots, M$:

$$0 = x_1 < x_2 < \dots < x_i < x_{i+1} < \dots < x_{M-1} < x_M = 1$$

Estos nodos dividen el intervalo $[0, 1]$ en $M - 1$ subintervalos llamados celdas:

$$[x_i, x_{i+1}] \quad i = 1, \dots, M - 1$$

Para la identificación de los nodos, en este caso de mallado 1D *no uniforme* se usa el índice i . Si

$$x_i = h * (i - 1) \quad h = 1/(M - 1)$$

entonces el mallado es *uniforme*.

En el caso de *mallados uniformes*, la densidad de la distribución de los nodos puede ser caracterizada por el parámetro h , el cual tiende a cero cuando el número de nodos incrementa.

Para *mallados no uniformes* usualmente se asume que existen ciertas constantes C_{max} y C_{min} , las cuales no dependen de h , y

$$C_{min}h \leq x_{i+1} - x_i \leq C_{max}h \quad (2.12)$$

Si esta supocisión llamada *regularidad* es válida, entonces si h tiende a cero la longitud de cada celda del mallado también tiende a cero. Por ello, para el caso no uniforme, el mallado está caracterizado por h y constantes C_{max} y C_{min} .

Mallados No Uniformes

Para la aplicación práctica de un mallado no uniforme, es útil e importante comprender la diferencia básicamente entre dos tipos.

El primer tipo de mallado no uniforme es el *mallado suave*. Se asume que existe una función suave $x = x(\xi)$, el cual transforma el segmento $[0, 1]$ en él mismo. Esta transformación debe ser uno a uno. Para el caso 1D una condición suficiente para esto, es que la función $x(\xi)$ sea monótona creciente. Supóngase que en el segmento $0 \leq \xi \leq 1$ hay un mallado uniforme con nodos $\xi_i = \frac{i-1}{M-1}, i = 1, \dots, M$. El mallado $x_i; i = 1, \dots, M$ en el segmento $0 \leq x \leq 1$ es llamado un *mallado suave* si las coordenadas están dadas por:

$$x_i = x(\xi_i)$$

y la función $x(\xi)$ es una *función suave*. (Nótese que el espacio de la variable ξ es llamada generalmente *espacio lógico*, y el espacio de la variable x es llamado *espacio físico*).

Una clase más amplia de mallados no uniformes son los llamados *no suaves*, los cuales no pueden ser construidos utilizando una transformación. Un ejemplo de tales mallados es uno en el que la longitud entre nodos es alternada por $h, 2h, h, \dots$

Para los mallados no uniformes es asumido que existen condiciones de regularidad como (2.12), la cual provee alguna información sobre la distribución de los nodos.

Tipos de Funciones de Mallado

Hay dos tipos principales de funciones de argumentos discretos en un caso 1D. En el primer caso, los valores de la función corresponden a los nodos. Es llamado *discretización nodal* y la función de mallado u^h es un conjunto (vector) de M

números:

$$u^h = u_i^h, i = 1, \dots, M$$

Usando la interpretación de función de mallado como el resultado de mediciones, se puede decir en el caso de discretización nodal, el instrumento mide calores exactamente en los nodos. Si es posible medir solamente el valor promedio sobre la celda (x_i, x_{i+1}) , entonces el valor de las funciones del mallado corresponden a las celdas, lo que se denomina *discretización por celda*, lo que hace que el valor de la función en este caso corresponda a la celda en general como un objeto geométrico. Para denotar los valores de funciones por celda, se utilizan índices medios; es decir que la notación $u_{i+1/2}^h$ es usada por el valor de la función de mallado en la celda (x_i, x_{i+1}) . Otra manera es haciendo referencia al índice u_i^h del nodo izquierdo de la celda.

Para comparar funciones de mallados tal como en el caso continuo, son usadas análogos discretos de las normas (2.10) y (2.11) para $u^h \in H$:

$$\|u^h\| = \max_{1 \leq i \leq M-1} |u_i^h| \tag{2.13}$$

$$\|u^h\| = \left(\sum_{i=1}^{M-1} (u_i^h)^2 * (x_{i+1} - x_i) \right)^{1/2} \tag{2.14}$$

2.3.2. Mallados en 2D

El ejemplo más simple de un mallado en 2D es llamado *producto tensorial* en un rectángulo. Sea el dominio de las variables x y y el rectángulo:

$$R = 0 \leq x \leq a \times 0 \leq y \leq b$$

Sobre los intervalos $0 \leq x \leq a$ y $0 \leq y \leq b$ es construido un mallado 1D como en la sección previa:

$$0 = x_1 < x_2 < \cdots < x_i < x_{i+1} < \cdots < x_{M-1} < x_M = a,$$

$$0 = y_1 < y_2 < \cdots < y_i < y_{j+1} < \cdots < x_{N-1} < x_N = b$$

El conjunto de nodos con coordenadas (x_i, y_j) sobre el plano es llamado mallado tensorial o mallado rectangular no uniforme. Claramente, el mallado consiste en los puntos en donde las líneas $x = x_i$ y $y = y_j$ se intersectan. En el caso de mallados de producto tensorial, la densidad puede ser caracterizada por el parámetro

$$h = \max(h_x, h_y) = \max\left(\frac{a}{M-1}, \frac{b}{N-1}\right)$$

el cual tiende a cero cuando el número de nodos incrementa.

El mallado es llamado *uniforme* en alguna dirección (x , y o ambas), si el mallado 1D en esa dirección es uniforme. Si el mallado es uniforme en ambas direcciones y los pasos h_x y h_y son iguales, el mallado es denominado *cuadrado*.

Para la identificación de cada nodo, se utilizan dos índices i, j . El rectángulo con vértices $(i, j), (i + 1, j), (i + 1, j + 1), (i, j + 1)$ (llamado celda) (ver figura 2.1) se usa para la identificación de esa celda.

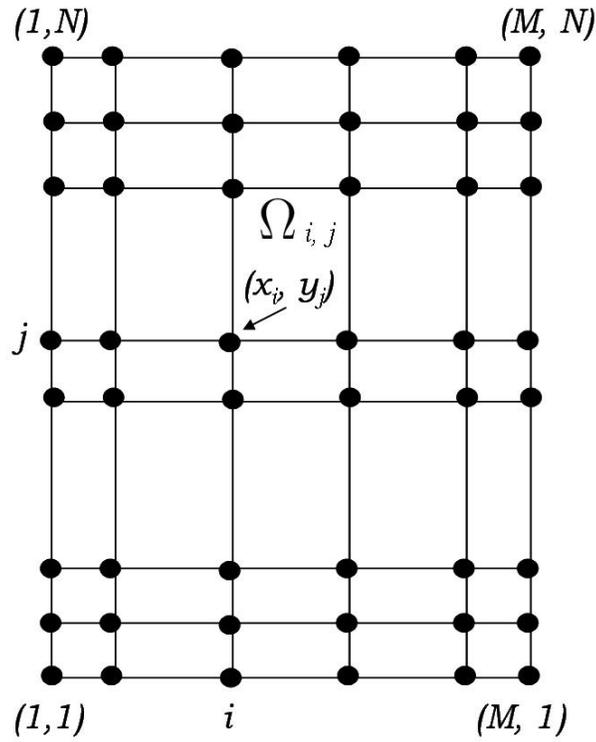


Figura 2.1: Mallado tensorial

2.4. Método de Operadores de Soporte 1-2-1

Existen dos tipos básicos de restricciones impuestas sobre los esquemas de diferencias finitas. La primera es la aproximación y las restricciones de estabilidad, lo que determina la convergencia de la solución aproximada a medida que el mallado se hace mas pequeño. El segundo tipo de restricciones se refieren a la retención que hace el esquema de diferencias finitas de propiedades importantes de las ecuaciones diferenciales originales. Cuando se calcula la solución numérica de las ecuaciones diferenciales parciales, los operadores de diferencia que imitan propiedades cruciales de los operadores diferenciales son generalmente más precisas que las que no lo hacen. Propiedades como la simetría, conservación,

relación de dualidad e identidades entre gradiente, rotor y divergencia son muy importantes, pero el autor de [1] considera la conservación como la más importante de las propiedades.

Independientemente de la aproximación utilizada, en la determinación del esquema de diferencias finitas, se considera la construcción de análogos de diferencias finitas para los operadores diferenciales invariantes básicos de primer orden, tales como grad, div y curl de la física-matemática. Las ventajas de la representación operacional de los esquemas de diferencias finitas, llega a ser más clara en el caso de mallados lógicos rectangulares y de mallados no estructurados, en los cuales las propiedades de los operadores de diferencia aseguran la conservación.

El método de Operadores de Soporte ha sido desarrollado por A.Favorskii, A. Samarskii, M. Shashkov y V. Tishkin luego de muchos años de estudio, y han logrado resumir el proceso de construcción de esquemas de diferencias finitas conservativas para ecuaciones de la física-matemática utilizando el método de Operadores de Soporte, en cinco pasos principales:

1. Escribir las ecuaciones originales en términos de los operadores diferenciales invariantes de primer orden: div, grad y curl.
2. Es importante comprender que las propiedades de los operadores diferenciales invariantes implican las leyes de conservación y otras características importantes del sistema original de ecuaciones diferenciales. En tal conexión, resaltan dos diferentes clases de propiedades en los operadores. La primera clase contiene las propiedades de un operador simple, y la segunda, las

propiedades que se conectan con dos o más operadores. Una de las principales relaciones del segundo tipo y que es usada para derivar aproximaciones consistentes para los operadores de diferencia div y grad en la construcción de esquemas de diferencias es:

$$\int_V \text{div} \vec{A} dV + \int_V (\vec{A}, \text{grad} \varphi) dV = \oint_S \varphi(\vec{A}, \vec{n}) dS \quad (2.15)$$

donde V es el volumen arbitrario con superficie S y vector normal unitario \vec{n} ; φ es una función escalar arbitraria; \vec{A} es una función vectorial arbitraria.

3. Escoger donde serán localizadas en el mallado las funciones escalares, vectoriales y tensoriales. Se consideran dos posibilidades: la primera es usar discretización nodal para las funciones escalares y discretización de celda centrada para funciones vectoriales de componentes Cartesianas, y una segunda posibilidad es utilizar discretización nodal para funciones vectoriales de componentes Cartesianas y de celda centrada para las funciones escalares.
4. Seleccionar el operador *primal*. Este debe ser uno de los operadores diferenciales de primer orden utilizados en la formulación de las ecuaciones originales. Por definición existe un análogo de diferencias para este operador. A partir de allí, todos los operadores de diferencias pueden ser derivados de la definición de discretización del operador primal y los análogos de diferencia de las identidades integrales. En el caso de discretización nodal de la función vectorial, es natural hacer uso de la divergencia como operador primal, y en el caso de discretización nodal de la función escalar es natural el operador grad como primal.
5. Las fórmulas para otros operadores de diferencias, llamados operadores

derivados, son obtenidas utilizando la expresión para el operador primal y la forma de los análogos en diferencias de las identidades integrales.

6. Finalmente, si se tienen análogos discretos para los operadores diferenciales, los cuales conforman las ecuaciones diferenciales originales, se puede generar el esquema de diferencias finitas sustituyendo los operadores diferenciales por operadores discretos.

Para construir un sistema de operadores de diferencia análogos de *div* y *grad* en el caso de una dimensión que satisfaga las condiciones ya formuladas, considérese que $V = [0, 1]$ y $\vec{A} = AX, 0, 0$. Esto es, \vec{A} tiene sólo una componente. Todas las funciones escalares y los componentes de los vectores dependen solo de la variable x .

En el caso donde *grad* y *div* son ambos igual a du/dx , debe construirse dos análogos de diferencia diferentes para la primera derivada. En 1D la identidad (2.15) queda:

$$\int_0^1 \varphi \frac{dAX}{dx} dx + \int_0^1 AX \frac{d\varphi}{dx} dx = \varphi(1)AX(1) - \varphi(0)AX(0)$$

Por simplicidad, se asume que la función φ es igual a cero en $x = 0$ y $x = 1$. Así, el lado derecho de la ecuación anterior es igual a cero y puede ser escrita como:

$$\int_0^1 \varphi \frac{dAX}{dx} dx + \int_0^1 AX \frac{d\varphi}{dx} dx = 0$$

Por ello, en el caso 1D los análogos de diferencia de la primera derivada deben imitar algunos análogos de diferencia de esta identidad.

Al principio se consideró el caso donde la función de diferencia escalar pertenece a HN , $u^h \in HN$ y $u_1^h = u_M^h = 0$. Esto significa que en este caso se considerarán las funciones discretas $u^h = (u_2^h, u_3^h, \dots, u_{M-1}^h)$. Es natural entonces seleccionar D_x como operador primal (el cual es análogo de la primera derivada) y $D_x : HN \rightarrow HC$. El valor en la celda i es

$$(D_x u^h)_i = \frac{u_{i+1}^h - u_i^h}{x_{i+1} - x_i} \quad (2.16)$$

La aproximación del Operador Divergencia queda definida por

$$DIV_{i+1/2} f = \frac{f_{i+1} - f_i}{h} \quad \text{en } \Omega \quad (2.17)$$

El Operador Gradiente se define como:

$$\begin{aligned} GRAD_0 f &= \frac{f_{1/2} - f_0}{h/2} && \text{en } \Gamma \\ GRAD_i f &= \frac{f_{i+1/2} - f_{i-1/2}}{h} && \text{en } \Omega \\ GRAD_N f &= \frac{f_N - f_{N-1/2}}{h/2} && \text{en } \Gamma \end{aligned} \quad (2.18)$$

Siendo h el espaciado uniforme del mallado $h = \Delta x$, con $\Delta x = x_{i+1} - x_i$

2.5. Método Castillo-Grone 2-2-2

En [4], J. Castillo y R. Grone muestran la manera de construir aproximaciones discretas de orden alto para la derivada en mallados uniformes unidimensionales y que satisfagan una ley de conservación global. Básicamente el problema consiste en hallar buenas aproximaciones de la divergencia ($\nabla \cdot$) y el gradiente (grad) que satisfagan el análogo discreto del teorema de la divergencia (2.15):

$$\int_{\Omega} \nabla \cdot \vec{v} f dV + \int_{\Omega} \vec{v} \text{grad} f dV = \int_{\partial\Omega} f \vec{v} \cdot \vec{n} dS \quad (2.19)$$

Para estos dos operadores existen tres ideas muy relacionadas: el *teorema de la divergencia* (2.19), conservación local y conservación global. La conservación local es un caso especial del teorema de la divergencia donde $f = 1$ y Ω una sola celda, mientras que la conservación global es (2.19) con $f = 1$ aplicado a la región completa bajo consideración. Las discretizaciones que poseen propiedades análogas a estas, son denominadas *miméticas*. Como tanto el teorema de la divergencia como la conservación local implican la conservación global, es relativamente sencillo hallar discretizaciones que satisfagan análogos a la conservación global.

En una configuración unidimensional dentro del intervalo $[0, 1]$, (2.19) es simplemente integración por partes:

$$\int_0^1 \frac{dv}{dx} f dx + \int_0^1 v \frac{df}{dx} dx = v(1)f(1) - v(0)f(0)$$

En las discretizaciones se utiliza el mallado de Operadores de Soporte, construido con $N > 0$, el cual es el número de celdas, y $h = 1/N$. Los nodos o puntos en el mallado son $x_i = ih$, $0 \leq i \leq N$. Las celdas están dadas por el intervalo $[ih, (i+1)h]$ con centro $x_{i+\frac{1}{2}} = (i + \frac{1}{2})h$, $0 \leq i \leq N - 1$. La divergencia discreta opera en los valores v , mientras que el gradiente discreto en los valores f .

2.5.1. Operadores discretos de segundo orden

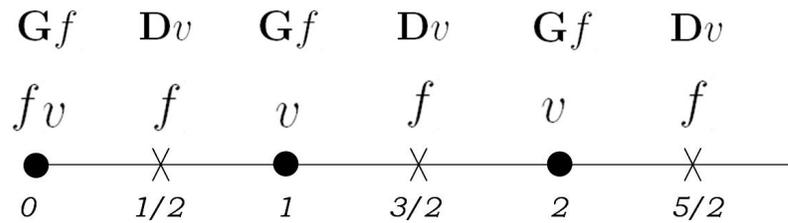
La divergencia discreta más simple está definida por

$$(\mathbf{D}v)_{i+\frac{1}{2}} = \frac{v_{i+1} - v_i}{h} \quad 0 \leq i \leq N - 1 \quad (2.20)$$

Y el gradiente discreto está definido por:

$$\begin{aligned} (\mathbf{G}f)_0 &= \frac{-\frac{4}{3}f_0 + \frac{3}{2}f_{\frac{1}{2}} - \frac{1}{6}f_{\frac{3}{2}}}{h/2} \\ (\mathbf{G}f)_i &= \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{h} \quad 1 \leq i \leq N - 1 \\ (\mathbf{G}f)_N &= \frac{\frac{4}{3}f_N - \frac{3}{2}f_{N-\frac{1}{2}} + \frac{1}{6}f_{N-\frac{3}{2}}}{h/2} \end{aligned} \quad (2.21)$$

donde nuevamente la definición de \mathbf{G} en los puntos de la frontera es la dada por la aproximación de Operadores de Soporte. La divergencia \mathbf{D} es una aproximación de segundo orden, y el gradiente \mathbf{G} es de segundo orden tanto en los nodos internos como en la frontera. La siguiente figura ilustra las posiciones de los valores de $\mathbf{D}v$ y $\mathbf{G}f$ en el mallado:



Los esquemas miméticos contruidos con esta aproximación, conservan las propiedades fundamentales de los operadores continuos originales, permitiendo a las aproximaciones discretas de las ecuaciones en derivadas parciales imitar propiedades críticas, incluyendo las leyes de conservación y simetría en la solución de los problemas físicos subyacentes.

2.6. Método Implícito de Crank-Nicolson

El método de Crank-Nicolson se corresponde con la regla del trapecio para ecuaciones diferenciales ordinarias y se basa en diferencias centrales para hacer aproximaciones de segundo orden en el tiempo. Esta aproximación para la primera derivada es como sigue:

$$\partial_t v x_i^{n+1/2} = \frac{\partial_t v x_i^{n+1} + \partial_t v x_i^n}{2} \quad (2.22)$$

Los desarrollos basados en el esquema de Crank-Nicolson son incondicionalmente estables, consistentes y convergentes para todos los valores en los pasos de tiempo, alcanzando aproximaciones considerablemente mejores que con métodos explícitos. Es utilizado conjuntamente con diferencias centrales espaciales con el propósito de alcanzar mayores precisiones.

2.7. Condiciones de Frontera en CFD

Todos los problemas en la dinámica de fluidos computacional son definidos en términos de las condiciones iniciales y de frontera. Es importante que el usuario especifique correctamente estos valores y comprenda el papel que desempeñan en los algoritmos numéricos. En los problemas no estacionarios, los valores iniciales de todas las variables del fluido necesitan ser establecidas en todos los puntos del dominio. Computacionalmente, esto significa inicializar apropiadamente las estructuras de datos utilizadas. Entre las condiciones de frontera comunmente utilizadas, se pueden mencionar:

- Condiciones de Entrada.
- Condiciones de Salida.
- Condiciones de Pared.
- Presiones prescritas o constantes.
- Simetría.
- Periodicidad (o condiciones de frontera cíclicas).

Haciendo uso de las condiciones de frontera, es posible establecer arbitrariamente la magnitud de una variable en un nodo del mallado con el propósito de simular situaciones específicas. Adicionalmente, es útil modificar los valores de algunas variables en nodos internos, con el fin de simular obstáculos dentro del dominio. El sistema de las ecuaciones de fluido discretizado puede ser resuelto normalmente sin tener que tratar tales obstáculos de forma separada. Algunas condiciones de fronteras más complejas, pueden incluir movimientos dependientes del tiempo, fronteras rotacionales o factores de aceleración, condiciones especiales para flujos supersónicos, entre otras. Debido a la importancia de combinar correctamente los diferentes tipos de condiciones de frontera dentro de una simulación con el fin de obtener los resultados y aproximaciones deseadas, es tarea del usuario establecerlas partiendo de un cuidadoso análisis de los aspectos involucrados y de la experiencia que tenga en identificar las condiciones de la situación planteada.

2.8. Métodos de Compresibilidad Artificial

Los flujos compresibles es un área de gran importancia en la mecánica de fluidos por sus aplicaciones en aerodinámica y diseño de turbinas, lo que ha llevado al desarrollo de métodos para su solución numérica. Surge entonces la posibilidad de adaptar tales métodos a la solución de flujos incompresibles. La mayor diferencia entre las ecuaciones de flujo compresible y los que no lo son, es netamente de carácter matemático. La versión compresible contiene la derivada en el tiempo de la densidad. En el caso de los flujos incompresibles la densidad es constante, lo cual elimina la opción de colocar tal derivada. Esto plantea que la derivada en el tiempo de la presión es una clara elección para estos casos, lo cual significa que realmente no se resuelven las ecuaciones de incompresibilidad. Como resultado, se ha cuestionado mucho la idea de utilizar métodos de compresibilidad artificial en problemas de flujo incompresible; sin embargo, ha sido ampliamente probado. Esta propuesta fue hecha por primera vez por Chorin [?] en 1967, planteándola entre otras versiones posibles basadas en los métodos compresibles. En resumen, la propuesta esencial es agregar la derivada en el tiempo de la presión a la ecuación de continuidad:

$$\frac{1}{\beta} \frac{\partial p}{\partial t} + \frac{\partial V_x}{\partial x} = 0, \quad (2.23)$$

donde β es un parámetro de compresibilidad artificial cuyo valor es clave en el desempeño de este método. Obviamente, mientras mayor sea el valor de β , mayor será la “incompresibilidad” en las ecuaciones considerando el caso de densidad constante.

El factor crucial en la convergencia de este método, es la escogencia del parámetro

β , debido a que el valor óptimo es dependiente del problema. Aún así, algunos autores han sugerido procedimientos automáticos para tal selección, ya que valores demasiado grandes requieren de un correcto campo de velocidad para poder satisfacer la ecuación de continuidad incompresible. También es posible determinar el menor valor aceptable para β a través de la propagación de velocidad de las ondas de presión. Obviamente, $1/(\beta\Delta t)$ debe ser pequeño comparado con los coeficientes de los otros términos de la ecuación, lo cual es necesario si se desea una rápida convergencia.

2.9. Resolución de Sistemas de Ecuaciones Lineales

En muchos casos el resultado del proceso de discretización numérica, es un sistema de ecuaciones algebraicas lineales o no lineales dependiendo de la naturaleza de las ecuaciones diferenciales. Aun en los casos no lineales, los métodos de solución envuelven resolver sistemas lineales. Es por ello que se hace necesario el uso de maneras eficientes para hallar soluciones a tales sistemas.

Las matrices resultantes de la discretización de ecuaciones en derivadas parciales son siempre esparcidas, es decir, la mayoría de sus elementos son nulos. Este hecho permite definir estructuras eficientes que almacenan y manipulan sólo los coeficientes relevantes para resolver el sistema.

Entre los métodos directos se encuentran la Eliminación Gaussiana, Descomposición LU, Método de Solución Tridiagonal (*Algoritmo de Thomas*) y el método de

Reducción Cíclica. Cualquier sistema de ecuaciones puede ser resuelto con estos algoritmos, pero desafortunadamente el costo de la triangulación en matrices esparcidas es extremadamente alto. Esto justifica el uso de métodos iterativos, los cuales sistemáticamente refinan una solución inicial. El más simple de estos métodos es el de Jacobi y entre los más conocidos e implementados están el método Gauss-Seidel, SOR (*Successive Over-Relaxation*), método de Stone (Descomposición LU incompleta), Gradiente Conjugado, Gradiente Biconjugado, CGSTAB, GMRES y Multigrid.

2.9.1. Uso de la librería UCSparseLib

UCSparseLib es una biblioteca portable para la resolución serial y paralela de sistemas lineales densos y esparcidos. Los sistemas lineales considerados son de la forma:

$$\mathbf{Ax}=\mathbf{b}$$

donde A es una matrix $n \times n$ densa o esparcida de gran tamaño suministrada por el usuario, y puede ser regular o irregular en su estructura, b es vector de términos independientes. Las investigaciones sobre técnicas de matrices esparcidas han llegado a incrementar su complejidad, y promete continuar acentuándose debido a la necesidad creciente de diseñar algoritmos eficientes para modernas supercomputadoras. Aún cuando existe gran número de paquetes y herramientas para resolver computacionalmente matrices densas de pequeño tamaño, hay también la necesidad de herramientas similares o bibliotecas de propósito general para el trabajo con matrices esparcidas. Una colección de unos cuantos programas

básicos para realizar tareas comunes y elementales, puede ser muy útil y reducir el tiempo dedicado a implementar y probar algoritmos para matrices esparcidas. Los paquetes Linpack y Eispack desarrollados en los años 70, han sido de gran ayuda en varias áreas de la computación científica, ahorrando grandes cantidades de esfuerzo y tiempo. Por otro lado, es muy frecuente que los investigadores de la computación de matrices esparcidas, codifiquen sus propias subrutinas para los modos de almacenamiento de la matriz o de su reordenamiento acorde con ciertas permutaciones; una de las razones pudiera ser la ausencia de estándares. Por ello, para la misma estructura de datos básica, puede estar en uso un gran número de variaciones.

UCSparseLib es una biblioteca de propósito general debido a que puede manipular matrices generales no simétricas y con patrones de estructura irregulares. Los patrones de las posiciones nulas no necesariamente deben ser simétricos. Desarrollada en ANSI C, UCSparseLib resuelve sistemas lineales esparcidos y densos. Entre las funcionalidades incluidas, y que han sido de utilidad en el desarrollo de este trabajo, están:

- Rutinas para leer y escribir matrices utilizando un formato simple. También es posible transformar una matriz a un formato postscript.
- Operaciones básicas aplicadas a vectores, multiplicación de matrices, *solvers* triangulares y reordenamiento de matrices.
- *Solvers* Iterativos como *Gauss-Seidel*, *Jacobi*, Gradiente Conjugado, GM-RES(m), BiCGstab y BiCG.
- Rutinas de utilidades adicionales como *timers*, tiempos de CPU y manejo

de memoria.

La siguiente figura muestra la organización general de la biblioteca:

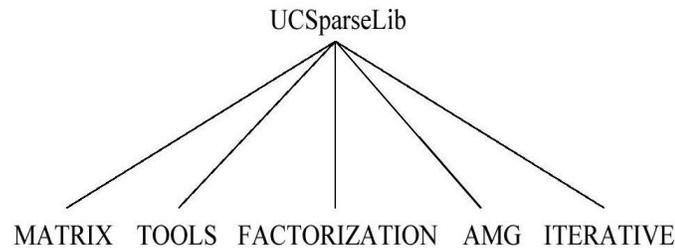


Figura 2.2: Organización General de UCSparseLib

Estructura de datos para matrices esparcidas

Una de las mayores dificultades encontradas en el cálculo de matrices esparcidas, es la variedad de tipos de matrices resultantes en aplicaciones prácticas. El propósito de un esquema de almacenamiento debe ser entonces, ganar eficiencia en términos de utilización de memoria y en operaciones aritméticas. Como resultado, se han propuesto muchas maneras diferentes de almacenar matrices esparcidas tomando ventajas de la estructura de las matrices o del problema específico del cual surgen. Uno de los esquemas más comunes de almacenamiento es el *Compressed Sparse Row* (CSR). En este esquema, todas las entradas no cero son almacenadas fila por fila en un arreglo de reales unidimensional, junto a un arreglo entero que contenga los índices de sus columnas y arreglos de apuntadores a cada una de tales filas. El orden de los elementos dentro de cada fila no es importante, lo cual simplifica el esquema de almacenamiento. En UCSparseLib una matriz es un objeto llamado `TDMatrix`, el cual es creado utilizando la siguiente rutina:

```
TDMatrixCreate(TDMatrix *M, TMatFormat format, int rows, int cols);
```

donde M es una matriz, y debe ser declarada como TDMatrix M. El formato format es el modo de almacenamiento, el cual puede ser MATRIX_DENSE_R, MATRIX_DENSE_C, MATRIX_SDENSE_R, MATRIX_SDENSE_C, MATRIX_DIAG, MATRIX_CSR, MATRIX_CSC, MATRIX_SCS_R, MATRIX_SCS_C. rows y cols son las dimensiones de la matriz.

Un componente importante del objeto TDMatrix es la estructura TDSparseVec. Esta estructura representa un vector esparcido y es:

```
typedef struct
{
    int      nz;      /* No null elements */
    int      diag;   /* Index of the diagonal element */
    int      *id;    /* col/row values */
    double   *val;   /* row/col elements */
} TDSparseVec;
```

Para acceder la fila i de la matriz TDMatrix M, se utiliza la siguiente macro:

```
For_TDMatrix_Row( M, i, row, mode ){
    .
    .
    .
}
```

donde row es un TDSparseVec y mode puede ser: ACCESS_READ, ACCESS_WRITE Y ACCESS_RW.

Existen también macros para acceder columnas o fila/columnas si el usuario no conoce el formato de la matriz. Las macros son: `For_TDMatrix_Col` y `For_TDMatrix_RC`, respectivamente.

2.10. OpenGL como API para la Visualización

OpenGL [20] es una interfaz de software para hardware gráfico. Esta interfaz consiste en aproximadamente 150 comandos diferentes para especificar los objetos y operaciones necesarias para producir aplicaciones gráficas interactivas.

OpenGL está diseñado como una línea de procesos, y es una interfaz independiente de hardware que puede ser implementada en muchas plataformas diferentes.

Con el fin de satisfacer estas cualidades, no posee comandos para la creación de ventanas u obtener entradas del usuario, por lo que se debe hacer uso de algún controlador del sistema de ventanas propio del hardware en particular. De igual manera, OpenGL no provee comandos de alto nivel para la creación de objetos complejos; en su lugar, sólo define un pequeño conjunto de primitivas geométricas tales como puntos, líneas y polígonos con los cuales es posible generar cualquier modelo deseado. Es por ello que se hace necesaria la construcción de todos los elementos que conforman los gráficos deseados para este trabajo, como una capa de abstracción de software que tiene a OpenGL y otras bibliotecas auxiliares como base.

Debido a que es posible realizar muchas cosas con el sistema gráfico de OpenGL, un programa puede llegar a ser complicado. Sin embargo, la estructura básica

puede ser realmente sencilla: inicializar ciertas variables de estado que controlan la forma en la cual OpenGL dibuja y los objetos a ser dibujados. Entre la terminología de la computación gráfica, la palabra *Render* implica el proceso en el cual un computador crea imágenes a partir de modelos. Estos modelos u objetos son construidos con primitivas geométricas (puntos, líneas y polígonos) que son especificadas por sus vértices.

El render final de la imagen consiste en píxeles dibujados en la pantalla; un píxel es el elemento visible más pequeño que el hardware gráfico puede colocar en la pantalla. La información sobre los píxeles (como por ejemplo, el color que deben tener), es organizada en la memoria como mapas de bits. Un mapa de bits es un área de memoria que almacena un bit de información para cada píxel en la pantalla; un bit podría indicar cuán rojo se supone que debe ser un píxel en particular, por ejemplo.

Como ya se ha mencionado, OpenGL provee un conjunto de comandos primitivos pero poderoso, y todos los tipos de dibujo de más alto nivel deben ser hechos en términos de tales comandos. Además, todos los programas hechos en OpenGL deben hacer uso de un mecanismo base de algún sistema específico de manejo de ventanas. Existe un número considerable de bibliotecas que permiten simplificar algunas de tales tareas de programación, como:

- OpenGL Utility Library (GLU), la cual contiene varias rutinas que hacen uso de los comandos de bajo nivel de OpenGL para llevar a cabo tareas como configuración de las matrices que especifican la orientación y proyección de las vistas, y dibujo de superficies. Esta biblioteca es distribuida como

parte de todas las implementaciones de OpenGL.

- OpenGL Utility Toolkit (GLUT) [21] es un conjunto de herramientas de ventanas independiente del sistema, que simplifica con pocas rutinas la necesidad de crear el entorno de dibujo requerido por OpenGL. La mayoría de las variables de estado se encuentran inicializadas por defecto en valores que facilitan el desarrollo en programas simples. Las rutinas igualmente requieren pocos parámetros. Por estas razones de simplicidad, GLUT no hace uso de manejadores, apuntadores o estructuras de datos nativas a algún sistema en particular; incluso provee de su propio conjunto (limitado) de fuentes.

2.11. Propuesta metodológica para el desarrollo del Software

Se plantea una metodología que se fundamente en la simplicidad, comunicación y retroalimentación o reutilización del código desarrollado (reciclado de código), haciendo que todo el proceso de desarrollo sea más simple que muchos de los métodos existentes. Entre sus objetivos está potenciar al máximo el trabajo en equipo. Esto implica que los diseños del software deben ser claros y sencillos. El código es revisado continuamente mediante la programación en pareja y se realizan pruebas todo el tiempo, no solo de cada nueva rutina (pruebas unitarias), sino también de integración al añadirlas al proyecto, o después de modificar cualquiera existente (integración continua). Se (re)diseña en todo momento (*refactoring*), dejando el código siempre en el estado más simple. Y las iteraciones son radicalmente más cortas de lo que es usual en otras metodologías, con el fin

de beneficiarse de la retroalimentación tan a menudo como sea posible. No es considerada la burocracia documental requerida en desarrollos no pertenecientes a las áreas de investigación científica.

A continuación se describen las etapas contempladas en cada uno de los ciclos de desarrollo de este proyecto:

2.11.1. Planificación

Se plantea la planificación como un diálogo permanente en el cual se decide el alcance del proyecto, la prioridad (qué debe ser hecho en primer lugar), la composición de las versiones (qué debería incluir cada una de ellas), y las fechas de inicio y culminación de las mismas. En cada iteración del desarrollo se planifican las actividades a realizarse, lo que convierte la comunicación en parte esencial del proyecto.

2.11.2. Diseño

Debido a que siempre costará menos tiempo implementar un diseño sencillo que uno complejo, se propone siempre realizar las cosas de la manera más sencilla posible. Si alguna parte de la implementación resulta especialmente compleja, se replantea ('divide y vencerás'). Se evita caer en la tentación de ir añadiendo funcionalidades adicionales no planificadas, aún incluso se conozca exactamente cómo implementarlas, es decir, centrarse en la tarea que se ha fijado y hacerla lo mejor posible. Se programa lo que se ha fijado sin perder tiempo en desarrollo de

código que no se sabe si será utilizado. En esta etapa se considera la reutilización como fortaleza en el diseño planteado.

2.11.3. Desarrollo

El código es desarrollado siguiendo los estándares definidos en UCSparceLib para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo. Esto es decisivo para poder plantear con éxito la propiedad colectiva del código.

Cuando los *test* son creados antes que el código, la implementación del código será mucho más rápida. El tiempo empleado en desarrollar un *test* y algo de código para probarlo es aproximadamente el mismo tiempo que se emplea en crear exclusivamente dicho código. La creación de las unidades de *test* ayudarán a tener una visión a cerca del *cómo*, en definitiva, del comportamiento del programa, lo que resulta especialmente beneficioso en el diseño de sistemas de software complicados.

Todo el código que formará parte del plan, será desarrollado por los dos miembros del equipo, quienes trabajarán de forma conjunta en el computador. De esta manera, se incrementará la calidad del software desarrollado sin afectar al tiempo de entrega. Mientras uno de ellos se encarga de pensar la táctica con la cual se va a abordar el problema, el otro se encargará de pensar las estrategias que permiten llevar dichas tácticas a su máximo exponente. Ambos roles son intercambiables.

No será optimizado el código hasta el final. Nunca se tratará de averiguar cuáles serán los posibles cuellos de botella del programa.

2.11.4. Pruebas

Las unidades de *test* o pruebas constituyen unos de los pilares básicos de esta propuesta. Se evita dejar la depuración de todo el software para el final. En cierta manera, una parte del código no será reemplazado si no supera las pruebas que existen para ese código. Después de cada modificación, se deberá emplear los test para verificar que un cambio en la estructura no introduce un cambio en la funcionalidad. Sin embargo, al añadir nuevas capacidades al código, tendrá que ser rediseñada la unidad de prueba, para adaptarse a la nueva funcionalidad. De esta manera, la probabilidad de que exista un fallo en ambos (*test* y código) es menor. De aquí la importancia de la creación de las unidades de prueba antes que el código, para que sean independientes de este.

Las pruebas se convierten en una herramienta de desarrollo, no en un paso de verificación que pueda despreciarse si se cree que el código está bien. El código será implantado cuando supere sus correspondientes unidades de prueba.

Capítulo 3

DESARROLLO

En el capítulo anterior se expusieron diferentes métodos para resolver una ecuación diferencial parcial de segundo orden, bien sea de forma explícita o transformando la ecuaciones diferenciales en un sistema de ecuaciones lineales. En este capítulo se usará el método de diferencias finitas miméticas, tanto Operadores de Soporte como Castillo-Grone 2-2-2, en un sistema de ecuaciones diferenciales de primer grado. Además, se mostrarán detalles de la librería UCSparselib, utilizada como la base principal para el desarrollo computacional del método en la ecuación de estudio de este trabajo. Finalmente, se mostrará el desarrollo procedimientos que permiten representar gráficamente el resultado del cálculo numerico de las ecuaciones implicadas en la simulación de flujo de fluido.

3.1. Solución Numérica de las Ecuaciones de Navier-Stokes y Compresibilidad Artificial en 1D

Particularmente para el caso de una dimensión, las ecuaciones de Navier-Stokes se pueden escribir como:

$$\frac{\partial Vx}{\partial t} + Vx \frac{\partial Vx}{\partial x} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \frac{\partial^2 Vx}{\partial x^2} \quad (3.1)$$

3.1.1. Operadores de Soporte para Mallados Uniformes. Método Explícito

Evaluando en los puntos medios y en el tiempo actual. La ecuación (3.1) queda como:

$$\partial_t Vx_{i+1/2}^n + Vx_{i+1/2}^n \partial_x Vx_{i+1/2}^n = -\frac{1}{\rho} \partial_x p_{i+1/2}^n + \nu \partial_{xx}^2 Vx_{i+1/2}^n \quad (3.2)$$

La aproximación de la derivada con respecto al tiempo que se usará es:

$$\partial_t f = \frac{f_i^{n+1} - f_i^n}{\Delta t} \quad (3.3)$$

Las aproximaciones en los puntos medios pueden ser calculadas como el promedio de los nodos del mallado. Es decir:

$$f_{i+1/2} = \frac{f_{i+1} + f_i}{2} + O(h^2) \quad (3.4)$$

$$f_{i-1/2} = \frac{f_i + f_{i-1}}{2} + O(h^2) \quad (3.5)$$

$$f_{i+3/2} = \frac{f_{i+2} + f_{i+1}}{2} + O(h^2) \quad (3.6)$$

Se debe sustituir el valor de los nodos medios como el promedio de los nodos enteros para aplicar la aproximación del operador gradiente a la ecuación (3.1):

$$\partial_t V x_{i+1/2}^n + V x_{i+1/2}^n \left(\frac{\partial_x V x_{i+1}^n + \partial_x V x_i^n}{2} \right) = -\frac{1}{\rho} \left(\frac{\partial_x p_{i+1}^n + \partial_x p_i^n}{2} \right) + \nu \partial_{xx}^2 V x_{i+1/2}^n$$

Para esta ecuación y sustituyendo por las aproximaciones dadas anteriormente, se obtiene:

$$D_t V x_{i+1/2}^n + V x_{i+1/2}^n \left(\frac{GRADV x_i^n + GRADV x_{i+1}^n}{2} \right) = -\frac{1}{\rho} \left(\frac{GRADp_i^n + GRADp_{i+1}^n}{2} \right) + \nu \left(\frac{GRADV x_{i+1}^n - GRADV x_i^n}{h} \right)$$

Sustituyendo por las aproximaciones correspondientes, la ecuación finalmente queda como:

$$\frac{V x_{i+1/2}^{n+1} - V x_{i+1/2}^n}{\Delta t} + V x_{i+1/2}^n \left(\frac{\frac{V x_{i+1/2}^n - V x_{i-1/2}^n}{h} + \frac{V x_{i+3/2}^n - V x_{i+1/2}^n}{h}}{2} \right) = -\frac{1}{\rho} \left(\frac{\frac{p_{i+1/2}^n - p_{i-1/2}^n}{h} + \frac{p_{i+3/2}^n - p_{i+1/2}^n}{h}}{2} \right) + \nu \left(\frac{\frac{V x_{i+3/2}^n - V x_{i+1/2}^n}{h} - \frac{V x_{i+1/2}^n - V x_{i-1/2}^n}{h}}{h} \right)$$

Aplicando operaciones matemáticas, la ecuación anterior se puede escribir de la siguiente manera:

$$\frac{V x_{i+1/2}^{n+1}}{\Delta t} - \frac{V x_{i+1/2}^n}{\Delta t} - \frac{V x_{i+1/2}^n V x_{i-1/2}^n}{2h} + \frac{V x_{i+1/2}^n V x_{i+3/2}^n}{2h} = \frac{p_{i-1/2}^n}{2h\rho} - \frac{p_{i+3/2}^n}{2h\rho} + \frac{\nu}{h^2} (V x_{i+3/2}^n - 2V x_{i+1/2}^n + V x_{i-1/2}^n)$$

A continuación son despejados los valores conocidos $t = n$ al lado derecho de la ecuación; debido a que solo existe una incógnita en la parte izquierda, simplemente

se sustituyen los valores conocidos del lado derecho. La ecuación general es:

$$Vx_{i+1/2}^{n+1} = \Delta t \left(\frac{p_{i-1/2}^n}{2h\rho} - \frac{p_{i+3/2}^n}{2h\rho} + \frac{\nu}{h^2} (Vx_{i+3/2}^n - 2Vx_{i+1/2}^n + Vx_{i-1/2}^n) \right. \\ \left. + \frac{Vx_{i+1/2}^n}{\Delta t} + \frac{Vx_{i+1/2}^n Vx_{i-1/2}^n}{2h} - \frac{Vx_{i+1/2}^n Vx_{i+3/2}^n}{2h} \right)$$

Con esta ecuación es posible hallar los valores de la incógnita Vx para el siguiente tiempo $n + 1$, conociendo las velocidades del tiempo n .

La ecuación de Compresibilidad Artificial en una dimensión

$$\frac{1}{\beta} \partial_t p + \partial_x Vx = 0$$

se valúa en los nodos medios y para el tiempo n :

$$\frac{1}{\beta} \partial_t p_{i+1/2}^n + \partial_x Vx_{i+1/2}^n = 0$$

Son sustituidas las aproximaciones dadas, tanto para el tiempo como para el espacio:

$$\frac{p_{i+1/2}^{n+1} - p_{i+1/2}^n}{\beta \Delta t} + \frac{Vx_{i+1}^n - Vx_i^n}{h} = 0$$

Desarrollando, se tiene:

$$p_{i+1/2}^{n+1} = p_{i+1/2}^n - \frac{\beta \Delta t}{h} (Vx_{i+1}^n - Vx_i^n)$$

Se sustituye por los promedios en los nodos medios dados por las ecuaciones (3.4):

$$p_{i+1/2}^{n+1} = p_{i+1/2}^n - \frac{\beta \Delta t}{h} \left(\frac{Vx_{i+1/2}^n + Vx_{i+3/2}^n}{2} - \frac{Vx_{i-1/2}^n + Vx_{i+1/2}^n}{2} \right)$$

Al simplificar se obtiene:

$$p_{i+1/2}^{n+1} = p_{i+1/2}^n - \frac{\beta \Delta t}{2h} (Vx_{i+3/2}^n - Vx_{i-1/2}^n)$$

Esta fórmula permite encontrar los valores desconocidos de la presión para el tiempo $n + 1$, considerando que los valores del tiempo n son conocidos.

Condiciones de Frontera para la ecuación de Navier-Stokes

Cuando $i = 0$ se utiliza la aproximación en la frontera izquierda, G_0 , como se muestra a continuación:

$$D_t Vx_{1/2}^n + Vx_{1/2}^n \left(\frac{GRADVx_0^n + GRADVx_1^n}{2} \right) = -\frac{1}{\rho} \left(\frac{GRADp_0^n + GRADp_1^n}{2} \right) + \nu \left(\frac{GRADVx_1^n - GRADVx_0^n}{h} \right)$$

Por lo que la ecuación finalmente queda como:

$$\frac{Vx_{1/2}^{n+1} - Vx_{1/2}^n}{\Delta t} + Vx_{1/2}^n \left(\frac{\frac{Vx_{1/2}^n - Vx_0^n}{h/2} + \frac{Vx_{3/2}^n - Vx_{1/2}^n}{h}}{2} \right) = -\frac{1}{\rho} \left(\frac{\frac{p_{1/2}^n - p_0^n}{h/2} + \frac{p_{3/2}^n - p_{1/2}^n}{h}}{2} \right) + \nu \left(\frac{\frac{Vx_{3/2} - Vx_{1/2}}{h} - \frac{Vx_{1/2} - Vx_0}{h/2}}{h} \right)$$

Simplificando y despejando se obtiene:

$$Vx_{1/2}^{n+1} = \Delta t \left(- \left(\frac{p_{1/2}^n - p_0^n}{h\rho} \right) - \left(\frac{p_{3/2}^n - p_{1/2}^n}{2h\rho} \right) + \nu \frac{Vx_{3/2} - Vx_{1/2}}{h^2} - \nu \frac{Vx_{1/2} - Vx_0}{h^2/2} + \frac{Vx_{1/2}^n}{\Delta t} - Vx_{1/2}^n \frac{Vx_{1/2}^n + Vx_0^n}{h} - Vx_{1/2}^n \frac{Vx_{3/2}^n - Vx_{1/2}^n}{2h} \right)$$

De manera similar se halla la ecuación para la frontera derecha:

Si $i = N - 1$:

$$\begin{aligned}
Vx_{N-1/2}^{n+1} = & Vx_{N-1/2}^n + \Delta t \left(-\frac{1}{\rho} \left(\frac{p_{N-1/2}^n - p_{N-3/2}^n}{2h} + \frac{p_N^n - p_{N-1/2}^n}{h} \right) \right. \\
& + \nu \left(\frac{Vx_N^n - Vx_{N-1/2}^n}{h^2/2} - \frac{Vx_{N-1/2}^n - Vx_{N-3/2}^n}{h^2} \right) \\
& \left. - Vx_{N-1/2}^n \frac{Vx_N^n - Vx_{N-1/2}^n}{h} - Vx_{N-1/2}^n \frac{Vx_{N-1/2}^n - Vx_{N-3/2}^n}{2h} \right)
\end{aligned}$$

Condiciones de Frontera para la ecuación de Compresibilidad Artificial

Si $i = 0$:

$$p_{1/2}^{n+1} = p_{1/2}^n - \frac{\beta \Delta t}{2h} (Vx_{3/2}^n + Vx_{1/2}^n - 2Vx_0^n)$$

Si $i = N - 1$:

$$p_{N-1/2}^{n+1} = p_{N-1/2}^n - \frac{\beta \Delta t}{2h} (2Vx_N^n - Vx_{N-3/2}^n - Vx_{N-1/2}^n)$$

3.1.2. Operadores de Soporte para Mallados Uniformes. Método Implícito

La ecuación de *Navier - Stokes* es evaluada en los puntos medios y en el tiempo $n + 1/2$ (Método de Crank-Nicolson) y se puede escribir como:

$$\partial_t Vx_{i+1/2}^n + Vx_{i+1/2}^n \partial_x Vx_{i+1/2}^{n+1/2} = -\frac{1}{\rho} \partial_x p_{i+1/2}^{n+1/2} + \nu \partial_{xx}^2 Vx_{i+1/2}^{n+1/2}$$

Ahora bien, las derivadas en el tiempo $n + 1/2$, se pueden aproximar con la suma del promedio de las derivadas en el tiempo $n + 1$ y tiempo n en ese mismo punto

$i + 1/2$, por lo que al realizar esto la ecuación anterior queda:

$$\begin{aligned} & \partial_t V x_{i+1/2}^n + V x_{i+1/2}^n \frac{\partial_x V x_{i+1/2}^{n+1} + \partial_x V x_{i+1/2}^n}{2} = \\ & - \frac{1}{\rho} \frac{\partial_x p_{i+1/2}^{n+1} + \partial_x p_{i+1/2}^n}{2} + \nu \frac{\partial_{xx}^2 V x_{i+1/2}^{n+1} + \partial_{xx}^2 V x_{i+1/2}^n}{2} \end{aligned}$$

Sustituyendo por las aproximaciones dadas, tenemos:

$$\begin{aligned} & \frac{V x_{i+1/2}^{n+1} - V x_{i+1/2}^n}{\Delta t} + V x_{i+1/2}^n \left(\frac{\frac{V x_{i+1/2}^{n+1} - V x_{i-1/2}^{n+1}}{h} + \frac{V x_{i+3/2}^{n+1} - V x_{i+1/2}^{n+1}}{h}}{2} + \frac{\frac{V x_{i+1/2}^n - V x_{i-1/2}^n}{h}}{2} \right) \\ & + \frac{\frac{V x_{i+3/2}^n - V x_{i+1/2}^n}{h}}{2} = - \frac{1}{\rho} \left(\frac{\frac{p_{i+1/2}^{n+1} - p_{i-1/2}^{n+1}}{h} + \frac{p_{i+3/2}^{n+1} - p_{i+1/2}^{n+1}}{h}}{2} + \frac{\frac{p_{i+1/2}^n - p_{i-1/2}^n}{h} + \frac{p_{i+3/2}^n - p_{i+1/2}^n}{h}}{2} \right) \\ & + \nu \left(\frac{\frac{V x_{i+3/2}^{n+1} - V x_{i+1/2}^{n+1}}{h} - \frac{V x_{i+1/2}^{n+1} - V x_{i-1/2}^{n+1}}{h}}{2} + \frac{\frac{V x_{i+3/2}^n - V x_{i+1/2}^n}{h} - \frac{V x_{i+1/2}^n - V x_{i-1/2}^n}{h}}{2} \right) \end{aligned}$$

Simplificando y despejando se tiene:

$$\begin{aligned} & \frac{V x_{i+1/2}^{n+1}}{\Delta t} - \frac{V x_{i+1/2}^n V x_{i-1/2}^{n+1}}{4h} + \frac{V x_{i+1/2}^n V x_{i+3/2}^{n+1}}{4h} - \frac{p_{i-1/2}^{n+1}}{4h\rho} + \frac{p_{i+3/2}^{n+1}}{4h\rho} \\ & - \nu \left(\frac{V x_{i+3/2}^{n+1} - 2V x_{i+1/2}^{n+1} + V x_{i-1/2}^{n+1}}{2h^2} \right) = \frac{p_{i-1/2}^n}{4h\rho} - \frac{p_{i+3/2}^n}{4h\rho} \\ & + \nu \left(\frac{V x_{i+3/2}^n - 2V x_{i+1/2}^n + V x_{i-1/2}^n}{2h^2} \right) + \frac{V x_{i+1/2}^n}{\Delta t} + \frac{V x_{i+1/2}^n V x_{i-1/2}^n}{4h} - \frac{V x_{i+1/2}^n V x_{i+3/2}^n}{4h} \end{aligned}$$

Agrupando convenientemente por términos de igual índice:

$$\begin{aligned} & \left(\frac{1}{\Delta t} + \frac{\nu}{h^2} \right) V x_{i+1/2}^{n+1} - \left(\frac{\nu}{2h^2} + \frac{V x_{i+1/2}^n}{4h} \right) V x_{i-1/2}^{n+1} + \left(\frac{V x_{i+1/2}^n}{4h} - \frac{\nu}{2h^2} \right) V x_{i+3/2}^{n+1} \\ & - \frac{p_{i-1/2}^{n+1}}{4h\rho} + \frac{p_{i+3/2}^{n+1}}{4h\rho} = \frac{p_{i-1/2}^n}{4h\rho} - \frac{p_{i+3/2}^n}{4h\rho} + \nu \left(\frac{V x_{i+3/2}^n - 2V x_{i+1/2}^n + V x_{i-1/2}^n}{2h^2} \right) \\ & + \frac{V x_{i+1/2}^n}{\Delta t} + \frac{V x_{i+1/2}^n V x_{i-1/2}^n}{4h} - \frac{V x_{i+1/2}^n V x_{i+3/2}^n}{4h} \end{aligned}$$

El mismo procedimiento es aplicado a la ecuación de Compresibilidad Artificial:

$$\frac{1}{\beta} \partial_t p_{i+1/2}^n + \partial_x V x_{i+1/2}^{n+1/2} = 0$$

por lo que

$$\frac{p_{i+1/2}^{n+1} - p_{i+1/2}^n}{\beta \Delta t} + \frac{\partial_x V x_{i+1/2}^{n+1} + \partial_x V x_{i+1/2}^n}{2} = 0$$

Aplicando las aproximaciones, se tiene

$$\frac{p_{i+1/2}^{n+1} - p_{i+1/2}^n}{\beta \Delta t} + \frac{\frac{V x_{i+1}^{n+1} - V x_i^{n+1}}{h} + \frac{V x_{i+1}^n - V x_i^n}{h}}{2} = 0$$

Se sustituyen los promedios como en el método explícito, obteniendo:

$$\frac{p_{i+1/2}^{n+1} - p_{i+1/2}^n}{\beta \Delta t} + \frac{\frac{\frac{V x_{i+1/2}^{n+1} + V x_{i+3/2}^{n+1}}{2} - \frac{V x_{i-1/2}^{n+1} + V x_{i+1/2}^{n+1}}{2}}{h} + \frac{\frac{V x_{i+1/2}^n + V x_{i+3/2}^n}{2} - \frac{V x_{i-1/2}^n + V x_{i+1/2}^n}{2}}{h}}{2} = 0$$

Finalmente despejando, queda:

$$\frac{p_{i+1/2}^{n+1}}{\beta \Delta t} + \frac{V x_{i+3/2}^{n+1}}{4h} - \frac{V x_{i-1/2}^{n+1}}{4h} = \frac{p_{i+1/2}^n}{\beta \Delta t} - \frac{V x_{i+3/2}^n}{4h} + \frac{V x_{i-1/2}^n}{4h}$$

Condiciones de Frontera para la ecuación de Navier-Stokes

Si $i = 0$:

$$\partial_t V x_{1/2}^n + V x_{1/2}^n \frac{\partial_x V x_{1/2}^{n+1} + \partial_x V x_{1/2}^n}{2} = -\frac{1}{\rho} \frac{\partial_x p_{1/2}^{n+1} + \partial_x p_{1/2}^n}{2} + \nu \frac{\partial_{xx}^2 V x_{1/2}^{n+1} + \partial_{xx}^2 V x_{1/2}^n}{2}$$

Sustituyendo promedios en los nodos medios y las aproximaciones, queda:

$$\begin{aligned} \frac{Vx_{1/2}^{n+1} - Vx_{1/2}^n}{\Delta t} + Vx_{1/2}^n & \frac{\frac{Vx_{1/2}^{n+1} - Vx_0^{n+1}}{h/2} + \frac{Vx_{3/2}^{n+1} - Vx_{1/2}^{n+1}}{h}}{2} + \frac{\frac{Vx_{1/2}^n - Vx_0^n}{h/2} + \frac{Vx_{3/2}^n - Vx_{1/2}^n}{h}}{2} = \\ & \frac{1}{\rho} \frac{\frac{p_{1/2}^{n+1} - p_0^{n+1}}{h/2} + \frac{p_{3/2}^{n+1} - p_{1/2}^{n+1}}{h}}{2} + \frac{\frac{p_{1/2}^n - p_0^n}{h/2} + \frac{p_{3/2}^n - p_{1/2}^n}{h}}{2} \\ & + \nu \frac{\frac{Vx_{3/2}^{n+1} - Vx_{1/2}^{n+1}}{h} - \frac{Vx_{1/2}^{n+1} - Vx_0^{n+1}}{h/2}}{h} + \frac{\frac{Vx_{3/2}^n - Vx_{1/2}^n}{h} - \frac{Vx_{1/2}^n - Vx_0^n}{h/2}}{h} \end{aligned}$$

Desarrollando y simplificando se obtiene:

$$\begin{aligned} \frac{Vx_{1/2}^{n+1}}{\Delta t} - \frac{Vx_{1/2}^n}{\Delta t} + Vx_{1/2}^n & \left(\frac{Vx_{1/2}^{n+1}}{4h} - \frac{Vx_0^{n+1}}{2h} + \frac{Vx_{3/2}^{n+1}}{4h} + \frac{Vx_{1/2}^n}{4h} - \frac{Vx_0^n}{2h} + \frac{Vx_{3/2}^n}{4h} \right) = \\ & - \frac{1}{\rho} \left(\frac{p_{1/2}^{n+1}}{4h} - \frac{p_0^{n+1}}{2h} + \frac{p_{3/2}^{n+1}}{4h} + \frac{p_{1/2}^n}{4h} - \frac{p_0^n}{2h} + \frac{p_{3/2}^n}{4h} \right) \\ & + \nu \left(\frac{Vx_{3/2}^{n+1} - 3Vx_{1/2}^{n+1} + 2Vx_0^{n+1}}{2h^2} + \frac{Vx_{3/2}^n - 3Vx_{1/2}^n + 2Vx_0^n}{2h^2} \right) \end{aligned}$$

Se despejan las incógnitas y agrupan términos semejantes:

$$\begin{aligned} \left(\frac{1}{\Delta t} + \frac{Vx_{1/2}^n}{4h} + \frac{3\nu}{2h^2} \right) Vx_{1/2}^{n+1} + \left(\frac{Vx_{1/2}^n}{4h} - \frac{\nu}{2h^2} \right) Vx_{3/2}^{n+1} + \frac{1}{4h\rho} p_{1/2}^{n+1} + \frac{1}{4h\rho} p_{3/2}^{n+1} = \\ - \frac{1}{\rho} \left(\frac{p_{1/2}^n}{4h} - \frac{p_0^n}{2h} + \frac{p_{3/2}^n}{4h} \right) + \nu \left(\frac{Vx_{3/2}^n - 3Vx_{1/2}^n + 2Vx_0^n}{2h^2} \right) + \frac{Vx_{1/2}^n}{\Delta t} \\ - Vx_{1/2}^n \left(\frac{Vx_{1/2}^n}{4h} - \frac{Vx_0^n}{2h} + \frac{Vx_{3/2}^n}{4h} \right) + \frac{Vx_{1/2}^n}{2h} Vx_0^{n+1} + \frac{p_0^{n+1}}{2h\rho} + \nu \frac{Vx_0^{n+1}}{h^2} \end{aligned}$$

De manera similar se halla la ecuación para la frontera derecha:

Si $i = N - 1$:

$$\begin{aligned} & \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2}^n}{4h} + \frac{3\nu}{2h^2} \right) Vx_{N-1/2}^{n+1} - \left(\frac{Vx_{N-1/2}^n}{4h} + \frac{\nu}{2h^2} \right) Vx_{N-3/2}^{n+1} - \frac{p_{N-3/2}^{n+1}}{4h\rho} - \frac{p_{N-1/2}^{n+1}}{4h\rho} = \\ & \frac{1}{\rho} \left(\frac{p_{N-3/2}^n}{4h} - \frac{p_N^n}{2h} + \frac{p_{N-1/2}^n}{4h} \right) + \nu \left(\frac{2Vx_N^n - 3Vx_{N-1/2}^n + Vx_{N-3/2}^n}{2h^2} \right) \\ & + Vx_{N-1/2}^n \left(\frac{1}{\Delta t} + \frac{Vx_{N-3/2}^n}{4h} - \frac{Vx_N^n}{2h} + \frac{Vx_{N-1/2}^n}{4h} - \frac{Vx_N^{n+1}}{2h} \right) - \frac{p_N^{n+1}}{2h\rho} + \frac{\nu Vx_N^{n+1}}{h^2} \end{aligned}$$

Ecuación de Compresibilidad Artificial

Si $i = 0$:

$$\frac{p_{1/2}^{n+1}}{\beta\Delta t} + \frac{Vx_{3/2}^{n+1}}{4h} + \frac{Vx_{1/2}^{n+1}}{4h} = \frac{p_{1/2}^n}{\beta\Delta t} - \frac{Vx_{3/2}^n}{4h} - \frac{Vx_{1/2}^n}{4h} + \frac{Vx_0^n}{2h} + \frac{Vx_0^{n+1}}{2h}$$

Si $i = N - 1$:

$$\frac{p_{N-1/2}^{n+1}}{\beta\Delta t} - \frac{Vx_{N-1/2}^{n+1}}{4h} - \frac{Vx_{N-3/2}^{n+1}}{4h} = \frac{p_{N-1/2}^n}{\beta\Delta t} - \frac{Vx_N^{n+1}}{2h} - \frac{Vx_N^n}{2h} + \frac{Vx_{N-1/2}^n}{4h} + \frac{Vx_{N-3/2}^n}{4h}$$

La matriz de coeficientes resultante es:

$$A = \begin{bmatrix} w & r_{1/2} & w & u_{1/2} & 0 & \dots & 0 \\ s & t & 0 & t & 0 & \dots & 0 \\ \vdots & & \ddots & & & & \vdots \\ 0 & \dots & -w & -q_{i+1/2} & 0 & r_{i+1/2} & w & u_{i+1/2} & \dots & 0 \\ 0 & \dots & 0 & -t & s & 0 & 0 & t & \dots & 0 \\ \vdots & & & & & & & & \ddots & \vdots \\ 0 & & & \dots & & & -w & -q_{N-1/2} & -w & r_{N-1/2} \\ 0 & & & \dots & & & 0 & -t & s & -t \end{bmatrix}$$

Siendo:

$$\begin{aligned}p_{1/2} &= \frac{1}{\Delta t} + \frac{Vx_{1/2}^n}{4h} + \frac{3\nu}{2h^2} \\p_{N-1/2} &= \frac{1}{\Delta t} - \frac{Vx_{N-1/2}^n}{4h} + \frac{3\nu}{2h^2} \\p_{i+1/2} &= \frac{1}{\Delta t} + \frac{\nu}{h^2} \\q_{i+1/2} &= \frac{Vx_{i+1/2}^n}{4h} + \frac{\nu}{2h^2} \\u_{i+1/2} &= \frac{Vx_{i+1/2}^n}{4h} - \frac{\nu}{2h^2} \\r &= \frac{1}{4h\rho} \\s &= \frac{1}{\beta\Delta t} \\t &= \frac{1}{4h}\end{aligned}$$

Y el vector de incognitas

$$b = \begin{pmatrix} p_{1/2}^{n+1} \\ v_{1/2}^{n+1} \\ \vdots \\ p_{i+1/2}^{n+1} \\ v_{i+1/2}^{n+1} \\ \vdots \\ p_{N-1/2}^{n+1} \\ v_{N-1/2}^{n+1} \end{pmatrix}$$

3.1.3. Método Castillo-Grone para Mallados Uniformes. Método Explícito 2-2-2

Las aproximaciones (2.20) y (2.21) para la divergencia y el gradiente respectivamente, se mantiene igual en los nodos internos, variando en los extremos. Así que las aproximaciones en la frontera de la ecuacion de Navier-Stokes es la siguiente, solo diferenciando del valor del operador gradiente en las fronteras:

Si $i = 0$

$$D_t V x_{1/2}^n + V x_{1/2}^n \left(\frac{GRAD_0 V x + GRAD_1 V x}{2} \right) = -\frac{1}{\rho} \left(\frac{GRAD_0 p + GRAD_1 p}{2} \right) + \nu \left(\frac{GRAD V x_1^n - GRAD V x_0^n}{h} \right)$$

Sustituyendo las aproximaciones y simplificando:

$$V x_{1/2}^{n+1} = V x_{1/2}^n + \Delta t \left(-V x_{1/2}^n \left(\frac{(-4/3)V x_0^n + (3/2)V x_{1/2}^n - (1/6)V x_{3/2}^n}{h} + \frac{V x_{3/2}^n - V x_{1/2}^n}{2h} \right) - \frac{1}{\rho} \left(\frac{(-4/3)p_0^n + (3/2)p_{1/2}^n - (1/6)p_{3/2}^n}{h} + \frac{p_{3/2}^n - p_{1/2}^n}{2h} \right) + \nu \left(\frac{V x_{3/2}^n - V x_{1/2}^n}{h^2} - \frac{(-4/3)V x_0^n + (3/2)V x_{1/2}^n - (1/6)V x_{3/2}^n}{h^2/2} \right) \right)$$

Realizando operaciones matemáticas y agrupación de términos, se obtiene:

$$V x_{1/2}^{n+1} = V x_{1/2}^n + \Delta t \left(\frac{4V x_{1/2}^n V x_0^n}{3h} + \left(\frac{1}{2h} - \frac{3}{2h} \right) (V x_{1/2}^n)^2 + \left(\frac{1}{6h} - \frac{1}{2h} \right) V x_{1/2}^n V x_{3/2}^n + \frac{4p_0^n}{3h\rho} + \left(\frac{1}{2h\rho} - \frac{3}{2h\rho} \right) p_{1/2}^n + \left(\frac{1}{6h\rho} - \frac{1}{2h\rho} \right) p_{3/2}^n + \frac{4\nu V x_0^n}{3h^2/2} - \left(\frac{\nu}{h^2} + \frac{3\nu}{h^2} \right) V x_{1/2}^n + \left(\frac{\nu}{h^2} + \frac{\nu}{3h^2} \right) V x_{3/2}^n \right)$$

Si $i = N - 1$:

$$\begin{aligned}
Vx_{N-1/2}^{n+1} = & Vx_{N-1/2}^n + \Delta t \left(\left(\frac{3}{2h\rho} - \frac{1}{2h\rho} \right) p_{N-1/2}^n + \left(\frac{1}{2h\rho} - \frac{1}{6h\rho} \right) p_{N-3/2}^n - \frac{4p_N^n}{3h\rho} \right. \\
& + \frac{4\nu Vx_N^n}{3h^2/2} - \left(\frac{\nu}{h^2} + \frac{3\nu}{h^2} \right) Vx_{N-1/2}^n + \left(\frac{\nu}{3h^2} - \frac{\nu}{h^2} \right) Vx_{N-3/2}^n + \left(\frac{3}{2h} - \frac{1}{2h} \right) (Vx_{N-1/2}^n)^2 \\
& \left. + \left(\frac{1}{2h} - \frac{1}{6h} \right) Vx_{N-1/2}^n Vx_{N-3/2}^n - \frac{4Vx_{N-1/2}^n Vx_N^n}{3h} \right)
\end{aligned}$$

La ecuación de Compresibilidad Artificial se mantiene igual debido a que las aproximaciones dependen solo del operador Divergencia y la aproximacion hacia adelante de la derivada en tiempo.

3.1.4. Método Castillo-Grone para Mallados Uniformes. Método Implícito 2-2-2

Si $i = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2}^n}{2h} + \frac{2\nu}{h} \right) Vx_{1/2}^{n+1} + \left(\frac{Vx_{1/2}^n}{6h} - \frac{2\nu}{3h} \right) Vx_{3/2}^{n+1} + \frac{1}{2h\rho} p_{1/2}^{n+1} + \frac{1}{6h\rho} p_{3/2}^{n+1} \\
= & -\frac{1}{\rho} \left(\frac{(-8/3)p_0^n + (3)p_{1/2}^n - (1/3)p_{3/2}^n}{4h} + \frac{p_{3/2}^n - p_{1/2}^n}{4h} \right) + \nu \left(\frac{Vx_{3/2}^n - Vx_{1/2}^n}{2h} \right. \\
& \left. - \frac{(-8/3)Vx_0^n + (3)Vx_{1/2}^n - (1/3)Vx_{3/2}^n}{2h} \right) + \frac{Vx_{1/2}^n}{\Delta t} + \frac{2p_0^{n+1}}{3h\rho} + \frac{4\nu Vx_0^{n+1}}{3h} \\
- & Vx_{1/2}^n \left(\frac{(-8/3)Vx_0^n + (3)Vx_{1/2}^n - (1/3)Vx_{3/2}^n}{4h} + \frac{Vx_{3/2}^n - Vx_{1/2}^n}{4h} \right) + \frac{2Vx_{1/2}^n Vx_0^{n+1}}{3h}
\end{aligned}$$

Si $i = N - 1$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2}^n}{2h} + \frac{2\nu}{h^2} \right) Vx_{N-1/2}^{n+1} - \left(\frac{Vx_{N-1/2}^n}{6h} + \frac{2\nu}{3h^2} \right) Vx_{N-3/2}^{n+1} - \frac{p_{N-3/2}^{n+1}}{6h\rho} - \frac{p_{N-1/2}^{n+1}}{2h\rho} \\
& = -\frac{1}{\rho} \left(\frac{p_{N-1/2}^n - p_{N-3/2}^n}{4h} + \frac{(8/3)p_N^n - (3)p_{N-1/2}^n + (1/3)p_{N-3/2}^n}{4h} \right) \\
& + \nu \left(\frac{(8/3)Vx_N^n - (3)Vx_{N-1/2}^n + (1/3)Vx_{N-3/2}^n}{2h^2} - \frac{Vx_{N-1/2}^n - Vx_{N-3/2}^n}{2h^2} \right) \\
& - Vx_{N-1/2}^n \left(\frac{Vx_{N-1/2}^n - Vx_{N-3/2}^n}{4h} + \frac{(8/3)Vx_N^n - (3)Vx_{N-1/2}^n + (1/3)Vx_{N-3/2}^n}{4h} \right) \\
& - \frac{2Vx_{N-1/2}^n Vx_N^{n+1}}{3h} - \frac{2p_N^{n+1}}{3h\rho} + \frac{4\nu Vx_N^{n+1}}{3h^2} + \frac{Vx_{N-1/2}^n}{\Delta t}
\end{aligned}$$

3.2. Solución Numérica de las Ecuaciones de Navier-Stokes y Compresibilidad Artificial en 2D

3.2.1. Método Operadores de Soporte para Mallados Uniformes. Método Implícito 1-2-1

Siendo h_x y h_y el espaciado para cada dimensión de un mallado uniforme, $h_x = \Delta x$ y $h_y = \Delta y$, con $\Delta x = x_{i+1} - x_i$ y $\Delta y = y_{i+1} - y_i$ respectivamente.

Se evalúa en los puntos medios y en el tiempo actual para la ecuacion (1.1):

$$\begin{aligned}
& \partial_t Vx_{i+1/2,j+1/2}^n + Vx_{i+1/2,j+1/2}^n \partial_x Vx_{i+1/2,j+1/2}^n + Vy_{i+1/2,j+1/2}^n \partial_y Vx_{i+1/2,j+1/2}^n = \\
& -\frac{1}{\rho} \partial_x p_{i+1/2,j+1/2}^n + \nu (\partial_{xx}^2 Vx_{i+1/2,j+1/2}^n + \partial_{yy}^2 Vx_{i+1/2,j+1/2}^n)
\end{aligned}$$

Las aproximaciones en los puntos medios se puede calcular como el promedio de

los nodos enteros (en X) del mallado, estos promedios se escriben a continuación:

$$\begin{aligned} f_{i+1/2,j+1/2} &= \frac{f_{i+1,j+1/2} + f_{i,j+1/2}}{2} + O(h^2) \\ f_{i-1/2,j+1/2} &= \frac{f_{i,j+1/2} + f_{i-1,j+1/2}}{2} + O(h^2) \\ f_{i+3/2,j+1/2} &= \frac{f_{i+2,j+1/2} + f_{i+1,j+1/2}}{2} + O(h^2) \end{aligned}$$

Aplicando Crank-Nicolson:

$$\begin{aligned} &\partial_t V x_{i+1/2,j+1/2}^n + V x_{i+1/2,j+1/2}^n \left(\frac{\partial_x V x_{i+1/2,j+1/2}^{n+1} + \partial_x V x_{i+1/2,j+1/2}^n}{2} \right) \\ + V y_{i+1/2,j+1/2}^n &\left(\frac{\partial_y V x_{i+1/2,j+1/2}^{n+1} + \partial_y V x_{i+1/2,j+1/2}^n}{2} \right) = -\frac{1}{\rho} \left(\frac{\partial_x p_{i+1/2,j+1/2}^{n+1} + \partial_x p_{i+1/2,j+1/2}^n}{2} \right) \\ + \nu &\left(\frac{\partial_{xx}^2 V x_{i+1/2,j+1/2}^{n+1} + \partial_{xx}^2 V x_{i+1/2,j+1/2}^n}{2} + \frac{\partial_{yy}^2 V x_{i+1/2,j+1/2}^{n+1} + \partial_{yy}^2 V x_{i+1/2,j+1/2}^n}{2} \right) \end{aligned}$$

Se sustituyen los promedios:

$$\begin{aligned} &\partial_t V x_{i+1/2,j+1/2}^n + V x_{i+1/2,j+1/2}^n \left(\frac{\frac{\partial_x V x_{i+1,j+1/2}^{n+1} + \partial_x V x_{i,j+1/2}^{n+1}}{2} + \frac{\partial_x V x_{i+1,j+1/2}^n + \partial_x V x_{i,j+1/2}^n}{2}}{2} \right) \\ + V y_{i+1/2,j+1/2}^n &\left(\frac{\frac{\partial_y V x_{i+1/2,j+1}^{n+1} + \partial_y V x_{i+1/2,j}^{n+1}}{2} + \frac{\partial_y V x_{i+1/2,j+1}^n + \partial_y V x_{i+1/2,j}^n}{2}}{2} \right) = \\ &-\frac{1}{\rho} \left(\frac{\frac{\partial_x p_{i+1,j+1/2}^{n+1} + \partial_x p_{i,j+1/2}^{n+1}}{2} + \frac{\partial_x p_{i+1,j+1/2}^n + \partial_x p_{i,j+1/2}^n}{2}}{2} \right) \\ + \nu &\left(\frac{\partial_{xx}^2 V x_{i+1/2,j+1/2}^{n+1} + \partial_{xx}^2 V x_{i+1/2,j+1/2}^n}{2} + \frac{\partial_{yy}^2 V x_{i+1/2,j+1/2}^{n+1} + \partial_{yy}^2 V x_{i+1/2,j+1/2}^n}{2} \right) \end{aligned}$$

Despejando y aplicando las aproximaciones, tenemos:

$$\begin{aligned}
& \frac{Vx_{i+1/2,j+1/2}^{n+1} - Vx_{i+1/2,j+1/2}^n}{\Delta t} + \frac{1}{\rho} \left(\frac{p_{i+3/2,j+1/2}^{n+1} - p_{i+1/2,j+1/2}^{n+1} + p_{i+1/2,j+1/2}^{n+1}}{4\Delta x} \right. \\
& \left. - \frac{p_{i-1/2,j+1/2}^{n+1}}{4\Delta x} \right) - \nu \left(\frac{Vx_{i+3/2,j+1/2}^{n+1} - Vx_{i+1/2,j+1/2}^{n+1} - Vx_{i+1/2,j+1/2}^{n+1} - Vx_{i-1/2,j+1/2}^{n+1}}{2(\Delta x)^2} \right. \\
& \left. + \frac{Vx_{i+1/2,j+3/2}^{n+1} - Vx_{i+1/2,j+1/2}^{n+1} - Vx_{i+1/2,j+1/2}^{n+1} - Vx_{i+1/2,j-1/2}^{n+1}}{2(\Delta y)^2} \right) \\
& + Vx_{i+1/2,j+1/2}^n \left(\frac{Vx_{i+3/2,j+1/2}^{n+1} - Vx_{i+1/2,j+1/2}^{n+1} + Vx_{i+1/2,j+1/2}^{n+1} - Vx_{i-1/2,j+1/2}^{n+1}}{4\Delta x} \right) \\
& + Vy_{i+1/2,j+1/2}^n \left(\frac{Vx_{i+1/2,j+3/2}^{n+1} - Vx_{i+1/2,j+1/2}^{n+1} + Vx_{i+1/2,j+1/2}^{n+1} - Vx_{i+1/2,j-1/2}^{n+1}}{4\Delta y} \right) \\
& = -\frac{1}{\rho} \left(\frac{p_{i+3/2,j+1/2}^n - p_{i+1/2,j+1/2}^n + p_{i+1/2,j+1/2}^n - p_{i-1/2,j+1/2}^n}{4\Delta x} \right) \\
& + \nu \left(\frac{Vx_{i+3/2,j+1/2}^n - Vx_{i+1/2,j+1/2}^n - Vx_{i+1/2,j+1/2}^n - Vx_{i-1/2,j+1/2}^n}{2(\Delta x)^2} \right. \\
& \left. + \frac{Vx_{i+1/2,j+3/2}^n - Vx_{i+1/2,j+1/2}^n - Vx_{i+1/2,j+1/2}^n - Vx_{i+1/2,j-1/2}^n}{2(\Delta y)^2} \right) \\
& - Vx_{i+1/2,j+1/2}^n \left(\frac{Vx_{i+3/2,j+1/2}^n - Vx_{i+1/2,j+1/2}^n + Vx_{i+1/2,j+1/2}^n - Vx_{i-1/2,j+1/2}^n}{4\Delta x} \right) \\
& - Vy_{i+1/2,j+1/2}^n \left(\frac{Vx_{i+1/2,j+3/2}^n - Vx_{i+1/2,j+1/2}^n + Vx_{i+1/2,j+1/2}^n - Vx_{i+1/2,j-1/2}^n}{4\Delta y} \right)
\end{aligned}$$

Al simplificar y agrupar términos semejantes, queda:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{\nu}{(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) V x_{i+1/2, j+1/2}^{n+1} + \frac{1}{4\rho\Delta x} p_{i+3/2, j+1/2}^{n+1} - \frac{1}{4\rho\Delta x} p_{i-1/2, j+1/2}^{n+1} \\
& + \left(\frac{V x_{i+1/2, j+1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) V x_{i+3/2, j+1/2}^{n+1} + \left(\frac{V x_{i+1/2, j+1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) V x_{i-1/2, j+1/2}^{n+1} \\
& + \left(\frac{V y_{i+1/2, j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) V x_{i+1/2, j+3/2}^{n+1} + \left(\frac{V y_{i+1/2, j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) V x_{i+1/2, j-1/2}^{n+1} \\
= & -\frac{1}{\rho} \left(\frac{p_{i+3/2, j+1/2}^n - p_{i-1/2, j+1/2}^n}{4\Delta x} \right) + \nu \left(\frac{V x_{i+3/2, j+1/2}^n - 2V x_{i+1/2, j+1/2}^n + V x_{i-1/2, j+1/2}^n}{2(\Delta x)^2} \right) \\
& + \nu \left(\frac{V x_{i+1/2, j+3/2}^n - 2V x_{i+1/2, j+1/2}^n + V x_{i+1/2, j-1/2}^n}{2(\Delta y)^2} \right) - V x_{i+1/2, j+1/2}^n \left(\frac{V x_{i+3/2, j+1/2}^n}{4\Delta x} \right. \\
& \left. - \frac{V x_{i-1/2, j+1/2}^n}{4\Delta x} \right) - V y_{i+1/2, j+1/2}^n \left(\frac{V x_{i+1/2, j+3/2}^n - V x_{i+1/2, j-1/2}^n}{4\Delta y} \right) + \frac{V x_{i+1/2, j+1/2}^n}{\Delta t}
\end{aligned}$$

Para el desarrollo de la ecuación (1.2), se tiene:

$$-\frac{1}{\rho} \frac{\partial p}{\partial y} = -\frac{1}{\rho} \left(\frac{\partial_y p_{i+1/2, j+1/2}^{n+1} + \partial_y p_{i+1/2, j+1/2}^n}{2} \right)$$

y

$$\begin{aligned}
\partial_y p_{i+1/2, j+1/2}^{n+1} &= \frac{\partial_y p_{i+1/2, j+1}^{n+1} + \partial_y p_{i+1/2, j}^{n+1}}{2} \\
\partial_y p_{i+1/2, j+1/2}^n &= \frac{\partial_y p_{i+1/2, j+1}^n + \partial_y p_{i+1/2, j}^n}{2}
\end{aligned}$$

Se aplica la aproximación del operador gradiente:

$$\begin{aligned}
\partial_y p_{i+1/2, j+1/2}^{n+1} &= \frac{p_{i+1/2, j+3/2}^{n+1} - p_{i+1/2, j-1/2}^{n+1}}{4\Delta y} \\
\partial_y p_{i+1/2, j+1/2}^n &= \frac{p_{i+1/2, j+3/2}^n - p_{i+1/2, j-1/2}^n}{4\Delta y}
\end{aligned}$$

Sustituyendo:

$$-\frac{1}{\rho} \frac{\partial p}{\partial y} = -\frac{1}{\rho} \left(\frac{p_{i+1/2, j+3/2}^{n+1} - p_{i+1/2, j-1/2}^{n+1}}{4\Delta y} + \frac{p_{i+1/2, j+3/2}^n - p_{i+1/2, j-1/2}^n}{4\Delta y} \right)$$

Finalmente la ecuación (1.2) de *Navier - Stokes* se puede escribir como:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{\nu}{(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) V y_{i+1/2, j+1/2}^{n+1} + \frac{1}{4\rho\Delta y} p_{i+1/2, j+3/2}^{n+1} - \frac{1}{4\rho\Delta y} p_{i+1/2, j-1/2}^{n+1} \\
& + \left(\frac{V x_{i+1/2, j+1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) V y_{i+3/2, j+1/2}^{n+1} + \left(\frac{V x_{i+1/2, j+1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) V y_{i-1/2, j+1/2}^{n+1} \\
& + \left(\frac{V y_{i+1/2, j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) V y_{i+1/2, j+3/2}^{n+1} + \left(\frac{V y_{i+1/2, j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) V y_{i+1/2, j-1/2}^{n+1} \\
= & -\frac{1}{\rho} \left(\frac{p_{i+1/2, j+3/2}^n - p_{i+1/2, j-1/2}^n}{4\Delta y} \right) + \nu \left(\frac{V y_{i+3/2, j+1/2}^n - 2V y_{i+1/2, j+1/2}^n + V y_{i-1/2, j+1/2}^n}{2(\Delta x)^2} \right) \\
& + \nu \left(\frac{V y_{i+1/2, j+3/2}^n - 2V y_{i+1/2, j+1/2}^n + V y_{i+1/2, j-1/2}^n}{2(\Delta y)^2} \right) - V x_{i+1/2, j+1/2}^n \left(\frac{V y_{i+3/2, j+1/2}^n}{4\Delta x} \right. \\
& \left. - \frac{V y_{i-1/2, j+1/2}^n}{4\Delta x} \right) - V y_{i+1/2, j+1/2}^n \left(\frac{V y_{i+1/2, j+3/2}^n - V y_{i+1/2, j-1/2}^n}{4\Delta y} \right) + \frac{V y_{i+1/2, j+1/2}^n}{\Delta t}
\end{aligned}$$

Aplicando Crank-Nicolson a la ecuación de compresibilidad artificial, se tiene:

$$\begin{aligned}
\frac{1}{\beta} \left(\frac{p_{i+1/2, j+1/2}^{n+1} - p_{i+1/2, j+1/2}^n}{\Delta t} \right) + \frac{\partial_x v x_{i+1/2, j+1/2}^{n+1} + \partial_x v x_{i+1/2, j+1/2}^n}{2} \\
+ \frac{\partial_y v y_{i+1/2, j+1/2}^{n+1} + \partial_y v y_{i+1/2, j+1/2}^n}{2} = 0
\end{aligned}$$

Sustituyendo por las correspondientes aproximaciones discretas, queda:

$$\begin{aligned}
\frac{1}{\beta} \left(\frac{p_{i+1/2, j+1/2}^{n+1} - p_{i+1/2, j+1/2}^n}{\Delta t} \right) + \frac{v x_{i+1, j+1/2}^{n+1} - v x_{i, j+1/2}^{n+1}}{2\Delta x} + \frac{v x_{i+1, j+1/2}^n - v x_{i, j+1/2}^n}{2\Delta x} \\
+ \frac{v y_{i+1/2, j+1}^{n+1} - v y_{i+1/2, j}^{n+1}}{2\Delta y} + \frac{v y_{i+1/2, j+1}^n - v y_{i+1/2, j}^n}{2\Delta y} = 0
\end{aligned}$$

Al sustituirlos los promedios en la ecuación tenemos que:

$$\begin{aligned}
\frac{1}{\beta\Delta t} p_{i+1/2, j+1/2}^{n+1} + \frac{1}{4\Delta x} v x_{i+3/2, j+1/2}^{n+1} - \frac{1}{4\Delta x} v x_{i-1/2, j+1/2}^{n+1} + \frac{1}{4\Delta y} v y_{i+1/2, j+3/2}^{n+1} \\
- \frac{1}{4\Delta y} v y_{i+1/2, j-1/2}^{n+1} = \frac{1}{\beta\Delta t} p_{i+1/2, j+1/2}^n - \frac{1}{4\Delta x} v x_{i+3/2, j+1/2}^n + \frac{1}{4\Delta x} v x_{i-1/2, j+1/2}^n \\
- \frac{1}{4\Delta y} v y_{i+1/2, j+3/2}^n + \frac{1}{4\Delta y} v y_{i+1/2, j-1/2}^n
\end{aligned}$$

Condiciones de Frontera

$i = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,j+1/2}^n}{4\Delta x} + \frac{3\nu}{2(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vx_{1/2,j+1/2}^{n+1} + \left(\frac{Vx_{1/2,j+1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vx_{3/2,j+1/2}^{n+1} \\
& + \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{1/2,j+3/2}^{n+1} - \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vx_{1/2,j-1/2}^{n+1} + \frac{p_{3/2,j+1/2}^{n+1}}{4\rho\Delta x} \\
& + \frac{p_{1/2,j+1/2}^{n+1}}{4\rho\Delta x} = -\frac{1}{\rho} \left(\frac{p_{3/2,j+1/2}^n + p_{1/2,j+1/2}^n - 2p_{0,j+1/2}^n}{4\Delta x} \right) + \nu \left(\frac{Vx_{3/2,j+1/2}^n - 3Vx_{1/2,j+1/2}^n}{2(\Delta x)^2} \right. \\
& + \frac{2Vx_{0,j+1/2}^n}{2(\Delta x)^2} + \left. \frac{Vx_{1/2,j+3/2}^n - 2Vx_{1/2,j+1/2}^n + Vx_{1/2,j-1/2}^n}{2(\Delta y)^2} \right) - Vx_{1/2,j+1/2}^n \left(\frac{Vx_{3/2,j+1/2}^n}{4\Delta x} \right. \\
& + \left. \frac{Vx_{1/2,j+1/2}^n - 2Vx_{0,j+1/2}^n}{4\Delta x} \right) - Vy_{1/2,j+1/2}^n \left(\frac{Vx_{1/2,j+3/2}^n - Vx_{1/2,j-1/2}^n}{4\Delta y} \right) \\
& + \left(\frac{Vx_{1/2,j+1/2}^n}{2\Delta x} \right) Vx_{0,j+1/2}^{n+1} + \frac{\nu}{(\Delta x)^2} Vx_{0,j+1/2}^{n+1} + \frac{p_{0,j+1/2}^{n+1}}{2\rho\Delta x} + \frac{Vx_{1/2,j+1/2}^n}{\Delta t}
\end{aligned}$$

$j = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vy_{i+1/2,1/2}^n}{4\Delta y} + \frac{\nu}{(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vx_{i+1/2,1/2}^{n+1} + \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vx_{i+3/2,1/2}^{n+1} \\
& - \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{i-1/2,1/2}^{n+1} + \left(\frac{Vy_{i+1/2,1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{i+1/2,3/2}^{n+1} + \frac{p_{i+3/2,1/2}^{n+1}}{4\rho\Delta x} \\
& - \frac{p_{i-1/2,1/2}^{n+1}}{4\rho\Delta x} = -\frac{1}{\rho} \left(\frac{p_{i+3/2,1/2}^n - p_{i-1/2,1/2}^n}{4\Delta x} \right) + \nu \left(\frac{Vx_{i+3/2,1/2}^n - 2Vx_{i+1/2,1/2}^n + Vx_{i-1/2,1/2}^n}{2(\Delta x)^2} \right. \\
& + \left. \frac{Vx_{i+1/2,3/2}^n - 3Vx_{i+1/2,1/2}^n + 2Vx_{i+1/2,0}^n}{2(\Delta y)^2} \right) - Vx_{i+1/2,1/2}^n \left(\frac{Vx_{i+3/2,1/2}^n - Vx_{i-1/2,1/2}^n}{4\Delta x} \right) \\
& - Vy_{i+1/2,1/2}^n \left(\frac{Vx_{i+1/2,3/2}^n + Vx_{i+1/2,1/2}^n - 2Vx_{i+1/2,0}^n}{4\Delta y} \right) + \frac{Vx_{i+1/2,1/2}^n}{\Delta t} + \left(\frac{\nu}{(\Delta y)^2} \right) Vx_{i+1/2,0}^{n+1} \\
& + \left(\frac{Vy_{i+1/2,1/2}^n}{2\Delta y} \right) Vx_{i+1/2,0}^{n+1}
\end{aligned}$$

$$i = N - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,j+1/2}^n}{4\Delta x} + \frac{3\nu}{2(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vx_{N-1/2,j+1/2}^{n+1} - \frac{p_{N-1/2,j+1/2}^{n+1}}{4\rho\Delta x} - \frac{p_{N-3/2,j+1/2}^{n+1}}{4\rho\Delta x} \\
& - \left(\frac{Vx_{N-1/2,j+1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{N-3/2,j+1/2}^{n+1} + \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,j+3/2}^{n+1} \\
& - \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,j-1/2}^{n+1} = -\frac{1}{\rho} \left(\frac{2p_{N,j+1/2}^n - p_{N-1/2,j+1/2}^n - p_{N-3/2,j+1/2}^n}{4\Delta x} \right) \\
& \quad - \frac{Vx_{N-1/2,j+1/2}^n}{2\Delta x} Vx_{N,j+1/2}^{n+1} + \nu \left(\frac{2Vx_{N,j+1/2}^n - 3Vx_{N-1/2,j+1/2}^n + Vx_{N-3/2,j+1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \quad \left. + \frac{Vx_{N-1/2,j+3/2}^n - 2Vx_{N-1/2,j+1/2}^n + Vx_{N-1/2,j-1/2}^n}{2(\Delta y)^2} \right) \\
& \quad - \frac{2p_{N,j+1/2}^{n+1}}{4\rho\Delta x} - Vx_{N-1/2,j+1/2}^n \left(\frac{2Vx_{N,j+1/2}^n - Vx_{N-1/2,j+1/2}^n - Vx_{N-3/2,j+1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,j+1/2}^n \left(\frac{Vx_{N-1/2,j+3/2}^n - Vx_{N-1/2,j-1/2}^n}{4\Delta y} \right) + \frac{Vx_{N-1/2,j+1/2}^n}{\Delta t} + \frac{\nu Vx_{N,j+1/2}^{n+1}}{(\Delta x)^2}
\end{aligned}$$

$$j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vy_{i+1/2,M-1/2}^n}{4\Delta y} + \frac{\nu}{(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vx_{i+1/2,M-1/2}^{n+1} + \frac{p_{i+3/2,M-1/2}^{n+1}}{4\rho\Delta x} - \frac{p_{i-1/2,M-1/2}^{n+1}}{4\rho\Delta x} \\
& + \left(\frac{Vx_{i+1/2,M-1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vx_{i+3/2,M-1/2}^{n+1} - \left(\frac{Vx_{i+1/2,M-1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{i-1/2,M-1/2}^{n+1} \\
& - \left(\frac{\nu}{2(\Delta y)^2} + \frac{Vy_{i+1/2,M-1/2}^n}{4\Delta y} \right) Vx_{i+1/2,M-3/2}^{n+1} = -\frac{1}{\rho} \left(\frac{p_{i+3/2,M-1/2}^n - p_{i-1/2,M-1/2}^n}{4\Delta x} \right) \\
& \quad + \frac{Vx_{i+1/2,M-1/2}^n}{\Delta t} + \nu \left(\frac{Vx_{i+3/2,M-1/2}^n - 2Vx_{i+1/2,M-1/2}^n + Vx_{i-1/2,M-1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. - \left(\frac{Vy_{i+1/2,M-1/2}^n}{2\Delta y} \right) Vx_{i+1/2,M}^{n+1} + \frac{2Vx_{i+1/2,M}^n - 3Vx_{i+1/2,M-1/2}^n + Vx_{i+1/2,M-3/2}^n}{2(\Delta y)^2} \right) \\
& \quad + \left(\frac{\nu}{(\Delta y)^2} \right) Vx_{i+1/2,M}^{n+1} - Vx_{i+1/2,M-1/2}^n \left(\frac{Vx_{i+3/2,M-1/2}^n - Vx_{i-1/2,M-1/2}^n}{4\Delta x} \right) \\
& \quad - Vy_{i+1/2,M-1/2}^n \left(\frac{2Vx_{i+1/2,M}^n - Vx_{i+1/2,M-1/2}^n - Vx_{i+1/2,M-3/2}^n}{4\Delta y} \right)
\end{aligned}$$

$$i = 0; j = 0$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,1/2}^n}{4\Delta x} + \frac{Vy_{1/2,1/2}^n}{4\Delta y} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vx_{1/2,1/2}^{n+1} \\
& + \left(\frac{Vx_{1/2,1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vx_{3/2,1/2}^{n+1} + \left(\frac{Vy_{1/2,1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{1/2,3/2}^{n+1} + \frac{p_{3/2,1/2}^{n+1}}{4\rho\Delta x} \\
& + \frac{p_{1/2,1/2}^{n+1}}{4\rho\Delta x} = -\frac{1}{\rho} \left(\frac{p_{3/2,1/2}^n + p_{1/2,1/2}^n - 2p_{0,1/2}^n}{4\Delta x} \right) + \nu \left(\frac{Vx_{3/2,1/2}^n - 3Vx_{1/2,1/2}^n + 2Vx_{0,1/2}^n}{2(\Delta x)^2} \right) \\
& + \frac{Vx_{1/2,3/2}^n - 3Vx_{1/2,1/2}^n + 2Vx_{1/2,0}^n}{2(\Delta y)^2} - Vx_{1/2,1/2}^n \left(\frac{Vx_{3/2,1/2}^n + Vx_{1/2,1/2}^n - 2Vx_{0,1/2}^n}{4\Delta x} \right) \\
& - Vy_{1/2,1/2}^n \left(\frac{Vy_{1/2,3/2}^n + Vy_{1/2,1/2}^n - 2Vy_{1/2,0}^n}{4\Delta y} \right) + \frac{Vx_{1/2,1/2}^n}{\Delta t} + \frac{Vx_{1/2,1/2}^n}{2\Delta x} Vx_{0,1/2}^{n+1} \\
& + \frac{p_{0,1/2}^{n+1}}{2\rho\Delta x} + \frac{\nu}{(\Delta y)^2} Vx_{1/2,0}^{n+1} + \frac{\nu}{(\Delta x)^2} Vy_{0,1/2}^{n+1} + \frac{Vy_{1/2,1/2}^n}{2\Delta y} Vy_{1/2,0}^{n+1}
\end{aligned}$$

$$i = N - 1; j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,M-1/2}^n}{4\Delta x} - \frac{Vy_{N-1/2,M-1/2}^n}{4\Delta y} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,M-1/2}^{n+1} \\
& - \left(\frac{Vx_{N-1/2,M-1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{N-3/2,M-1/2}^{n+1} - \left(\frac{Vy_{N-1/2,M-1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vy_{N-1/2,M-3/2}^{n+1} \\
& - \frac{p_{N-1/2,M-1/2}^{n+1}}{4\rho\Delta x} - \frac{p_{N-3/2,M-1/2}^{n+1}}{4\rho\Delta x} = -\frac{1}{\rho} \left(\frac{2p_{N,M-1/2}^n - p_{N-1/2,M-1/2}^n - p_{N-3/2,M-1/2}^n}{4\Delta x} \right) \\
& + \frac{Vx_{N-1/2,M-1/2}^n}{\Delta t} + \nu \left(\frac{2Vx_{N,M-1/2}^n - 3Vx_{N-1/2,M-1/2}^n + Vx_{N-3/2,M-1/2}^n}{2(\Delta x)^2} \right) \\
& + \frac{\nu}{(\Delta y)^2} Vy_{N-1/2,M}^{n+1} - \frac{p_{N,M-1/2}^{n+1}}{2\rho\Delta x} + \frac{2Vx_{N-1/2,M}^n - 3Vx_{N-1/2,M-1/2}^n + Vx_{N-1/2,M-3/2}^n}{2(\Delta y)^2} \\
& - Vx_{N-1/2,M-1/2}^n \left(\frac{2Vx_{N,M-1/2}^n - Vx_{N-1/2,M-1/2}^n - Vx_{N-3/2,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,M-1/2}^n \left(\frac{2Vy_{N-1/2,M}^n - Vy_{N-1/2,M-1/2}^n - Vy_{N-1/2,M-3/2}^n}{4\Delta y} \right) \\
& - \frac{Vx_{N-1/2,M-1/2}^n}{2\Delta x} Vy_{N,M-1/2}^{n+1} - \frac{Vy_{N-1/2,M-1/2}^n}{2\Delta y} Vx_{N-1/2,M}^{n+1} + \frac{\nu}{(\Delta x)^2} Vx_{N,M-1/2}^{n+1}
\end{aligned}$$

$$i = 0; j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,M-1/2}^n}{4\Delta x} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} - \frac{Vy_{1/2,M-1/2}^n}{4\Delta y} \right) Vx_{1/2,M-1/2}^{n+1} \\
& + \left(\frac{Vx_{1/2,M-1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vx_{3/2,M-1/2}^{n+1} - \left(\frac{Vy_{1/2,M-1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vx_{1/2,M-3/2}^{n+1} \\
& + \frac{p_{3/2,M-1/2}^{n+1}}{4\rho\Delta x} + \frac{p_{1/2,M-1/2}^{n+1}}{4\rho\Delta x} = -\frac{1}{\rho} \left(\frac{p_{3/2,M-1/2}^n + p_{1/2,M-1/2}^n - 2p_{0,M-1/2}^n}{4\Delta x} \right) + \nu \left(\frac{Vx_{3/2,M-1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. - \frac{3Vx_{1/2,M-1/2}^n + 2Vx_{0,M-1/2}^n}{2(\Delta x)^2} + \frac{2Vx_{1/2,M}^n - 3Vx_{1/2,M-1/2}^n + Vx_{1/2,M-3/2}^n}{2(\Delta y)^2} \right) \\
& + \frac{Vx_{1/2,M-1/2}^n}{\Delta t} + \frac{p_{0,M-1/2}^{n+1}}{2\rho\Delta x} - Vx_{1/2,M-1/2}^n \left(\frac{Vx_{3/2,M-1/2}^n + Vx_{1/2,M-1/2}^n - 2Vx_{0,M-1/2}^n}{4\Delta x} \right) \\
& + \nu \left(\frac{Vx_{0,M-1/2}^{n+1}}{(\Delta x)^2} + \frac{Vx_{1/2,M}^{n+1}}{(\Delta y)^2} \right) - Vy_{1/2,M-1/2}^n \left(\frac{2Vx_{1/2,M}^n - Vx_{1/2,M-1/2}^n - Vx_{1/2,M-3/2}^n}{4\Delta y} \right) \\
& \quad + \left(\frac{Vx_{1/2,M-1/2}^n}{2\Delta x} \right) Vx_{0,M-1/2}^{n+1} - \left(\frac{Vy_{1/2,M-1/2}^n}{2\Delta y} Vx_{1/2,M}^{n+1} \right)
\end{aligned}$$

$$i = N - 1; j = 0$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,1/2}^n}{4\Delta x} + \frac{Vy_{N-1/2,1/2}^n}{4\Delta y} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,1/2}^{n+1} - \frac{p_{N-1/2,1/2}^{n+1}}{4\rho\Delta x} \\
& - \frac{p_{N-3/2,1/2}^{n+1}}{4\rho\Delta x} + \left(\frac{Vy_{N-1/2,1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,3/2}^{n+1} - \left(\frac{Vx_{N-1/2,1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{N-3/2,1/2}^{n+1} \\
& = -\frac{1}{\rho} \left(\frac{2p_{N,1/2}^n - p_{N-1/2,1/2}^n - p_{N-3/2,1/2}^n}{4\Delta x} \right) + \nu \left(\frac{2Vx_{N,1/2}^n - 3Vx_{N-1/2,1/2}^n + Vx_{N-3/2,1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{Vx_{N-1/2,3/2}^n - 3Vx_{N-1/2,1/2}^n + 2Vx_{N-1/2,0}^n}{2(\Delta y)^2} \right) - \frac{Vy_{N-1/2,1/2}^n}{2\Delta y} Vx_{N-1/2,0}^{n+1} - \frac{p_{N,1/2}^{n+1}}{2\rho\Delta x} \\
& \quad - Vx_{N-1/2,1/2}^n \left(\frac{2Vx_{N,1/2}^n - Vx_{N-1/2,1/2}^n - Vx_{N-3/2,1/2}^n}{4\Delta x} \right) \\
& \quad - Vy_{N-1/2,1/2}^n \left(\frac{Vx_{N-1/2,3/2}^n Vx_{N-1/2,1/2}^n - 2Vx_{N-1/2,0}^n}{4\Delta y} \right) \\
& \quad + \frac{Vx_{N-1/2,1/2}^n}{\Delta t} + \frac{\nu}{(\Delta y)^2} Vx_{N-1/2,0}^{n+1} + \frac{Vx_{N-1/2,1/2}^n}{2\Delta x} Vx_{N,1/2}^{n+1} - \frac{\nu}{(\Delta x)^2} Vx_{N,1/2}^{n+1}
\end{aligned}$$

Para la ecuación 2 de *Navier - Stokes* las aproximaciones son las siguientes:

$i = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,j+1/2}^n}{4\Delta x} + \frac{3\nu}{2(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vy_{1/2,j+1/2}^{n+1} + \left(\frac{Vx_{1/2,j+1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vy_{3/2,j+1/2}^{n+1} \\
& \quad \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{1/2,j+3/2}^{n+1} - \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vy_{1/2,j-1/2}^{n+1} \\
& \quad + \frac{p_{1/2,j+3/2}^{n+1}}{4\rho\Delta y} - \frac{p_{1/2,j-1/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{p_{1/2,j+3/2}^n - p_{1/2,j-1/2}^n}{4\Delta y} \right) \\
& + \nu \left(\frac{Vy_{3/2,j+1/2}^n - 3Vy_{1/2,j+1/2}^n + 2Vy_{0,j+1/2}^n}{2(\Delta x)^2} + \frac{Vy_{1/2,j+3/2}^n - 2Vy_{1/2,j+1/2}^n + Vy_{1/2,j-1/2}^n}{2(\Delta y)^2} \right) \\
& \quad + \frac{Vy_{1/2,j+1/2}^n}{\Delta t} - Vx_{1/2,j+1/2}^n \left(\frac{Vy_{3/2,j+1/2}^n + Vy_{1/2,j+1/2}^n - 2Vy_{0,j+1/2}^n}{4\Delta x} \right) \\
& - Vy_{1/2,j+1/2}^n \left(\frac{Vy_{1/2,j+3/2}^n - Vy_{1/2,j-1/2}^n}{4\Delta y} \right) + \left(\frac{Vx_{1/2,j+1/2}^n}{2\Delta x} \right) Vy_{0,j+1/2}^{n+1} + \frac{\nu}{(\Delta x)^2} Vy_{0,j+1/2}^{n+1}
\end{aligned}$$

$j = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vy_{i+1/2,1/2}^n}{4\Delta y} + \frac{\nu}{(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vy_{i+1/2,1/2}^{n+1} + \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vy_{i+3/2,1/2}^{n+1} \\
& - \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vy_{i-1/2,1/2}^{n+1} + \left(\frac{Vy_{i+1/2,1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{i+1/2,3/2}^{n+1} + \frac{p_{i+1/2,3/2}^{n+1}}{4\rho\Delta y} \\
& + \frac{p_{i+1/2,1/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{p_{i+1/2,3/2}^n + p_{i+1/2,1/2}^n - 2p_{i+1/2,0}^n}{4\Delta y} \right) + \nu \left(\frac{Vy_{i+3/2,1/2}^n - 2Vy_{i+1/2,1/2}^n}{2(\Delta x)^2} \right) \\
& + \frac{Vy_{i-1/2,1/2}^n}{2(\Delta x)^2} + \frac{Vy_{i+1/2,3/2}^n - 3Vy_{i+1/2,1/2}^n + 2Vy_{i+1/2,0}^n}{2(\Delta y)^2} - Vx_{i+1/2,1/2}^n \left(\frac{Vy_{i+3/2,1/2}^n}{4\Delta x} \right. \\
& \left. - \frac{Vy_{i-1/2,1/2}^n}{4\Delta x} \right) - Vy_{i+1/2,1/2}^n \left(\frac{Vy_{i+1/2,3/2}^n + Vy_{i+1/2,1/2}^n - 2Vy_{i+1/2,0}^n}{4\Delta y} \right) + \frac{\nu}{(\Delta y)^2} Vy_{i+1/2,0}^{n+1} \\
& \quad + \left(\frac{Vy_{i+1/2,1/2}^n}{2\Delta y} \right) Vy_{i+1/2,0}^{n+1} + \frac{p_{i+1/2,0}^{n+1}}{2\rho\Delta y} + \frac{Vy_{i+1/2,1/2}^n}{\Delta t}
\end{aligned}$$

$i = N - 1$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,j+1/2}^n}{4\Delta x} + \frac{3\nu}{2(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vy_{N-1/2,j+1/2}^{n+1} \\
& - \left(\frac{Vx_{N-1/2,j+1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vy_{N-3/2,j+1/2}^{n+1} + \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{N-1/2,j+3/2}^{n+1} \\
& \quad - \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vy_{N-1/2,j-1/2}^{n+1} + \frac{p_{N-1/2,j+3/2}^{n+1}}{4\rho\Delta y} - \frac{p_{N-1/2,j-1/2}^{n+1}}{4\rho\Delta y} = \\
& - \frac{1}{\rho} \left(\frac{p_{N-1/2,j+3/2}^{n+1} - p_{N-1/2,j-1/2}^{n+1}}{4\Delta y} \right) + \nu \left(\frac{2Vy_{N,j+1/2}^n - 3Vy_{N-1/2,j+1/2}^n + Vy_{N-3/2,j+1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{Vy_{N-1/2,j+3/2}^n - 2Vy_{N-1/2,j+1/2}^n + Vy_{N-1/2,j-1/2}^n}{2(\Delta y)^2} \right) - \frac{Vx_{N-1/2,j+1/2}^n Vy_{N,j+1/2}^{n+1}}{2\Delta x} \\
& \quad - \frac{p_{N,j+1/2}^{n+1}}{2\rho\Delta y} - Vx_{N-1/2,j+1/2}^n \left(\frac{2Vy_{N,j+1/2}^n - Vy_{N-1/2,j+1/2}^n - Vy_{N-3/2,j+1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,j+1/2}^n \left(\frac{Vy_{N-1/2,j+3/2}^n - Vy_{N-1/2,j-1/2}^n}{4\Delta y} \right) + \frac{Vy_{N-1/2,j+1/2}^n}{\Delta t} + \frac{\nu Vy_{N,j+1/2}^{n+1}}{(\Delta x)^2}
\end{aligned}$$

$j = M - 1$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vy_{i+1/2,M-1/2}^n}{4\Delta y} + \frac{\nu}{(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vy_{i+1/2,M-1/2}^{n+1} - \frac{p_{i+1/2,M-1/2}^{n+1}}{4\rho\Delta y} - \frac{p_{i+1/2,M-3/2}^{n+1}}{4\rho\Delta y} \\
& + \left(\frac{Vx_{i+1/2,M-1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vy_{i+3/2,M-1/2}^{n+1} - \left(\frac{Vx_{i+1/2,M-1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vy_{i-1/2,M-1/2}^{n+1} \\
& - \left(\frac{\nu}{2(\Delta y)^2} + \frac{Vy_{i+1/2,M-1/2}^n}{4\Delta y} \right) Vy_{i+1/2,M-3/2}^{n+1} = -\frac{1}{\rho} \left(\frac{2p_{i+1/2,M}^n - p_{i+1/2,M-1/2}^n - p_{i+1/2,M-3/2}^n}{4\Delta y} \right) \\
& \quad + \frac{Vy_{i+1/2,M-1/2}^n}{\Delta t} + \nu \left(\frac{Vy_{i+3/2,M-1/2}^n - 2Vy_{i+1/2,M-1/2}^n + Vy_{i-1/2,M-1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{2Vy_{i+1/2,M}^n - 3Vy_{i+1/2,M-1/2}^n + Vy_{i+1/2,M-3/2}^n}{2(\Delta y)^2} \right) - \left(\frac{Vy_{i+1/2,M-1/2}^n}{2\Delta y} \right) Vy_{i+1/2,M}^{n+1} \\
& + \frac{\nu}{(\Delta y)^2} Vy_{i+1/2,M}^{n+1} - \frac{p_{i+1/2,M}^{n+1}}{2\Delta y} - Vx_{i+1/2,M-1/2}^n \left(\frac{Vy_{i+3/2,M-1/2}^n - Vy_{i-1/2,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{i+1/2,M-1/2}^n \left(\frac{2Vy_{i+1/2,M}^n - Vy_{i+1/2,M-1/2}^n - Vy_{i+1/2,M-3/2}^n}{4\Delta y} \right)
\end{aligned}$$

$$i = 0; j = 0$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,1/2}^n}{4\Delta x} + \frac{Vy_{1/2,1/2}^n}{4\Delta y} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vy_{1/2,1/2}^{n+1} \\
& + \left(\frac{Vx_{1/2,1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vy_{3/2,1/2}^{n+1} + \left(\frac{Vy_{1/2,1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{1/2,3/2}^{n+1} + \frac{p_{1/2,3/2}^{n+1}}{4\rho\Delta y} \\
& + \frac{p_{1/2,1/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{p_{1/2,3/2}^n + p_{1/2,1/2}^n - 2p_{1/2,0}^n}{4\Delta y} \right) + \nu \left(\frac{Vy_{3/2,1/2}^n - 3Vy_{1/2,1/2}^n + 2Vy_{0,1/2}^n}{2(\Delta x)^2} \right) \\
& + \frac{Vy_{1/2,3/2}^n - 3Vy_{1/2,1/2}^n + 2Vy_{1/2,0}^n}{2(\Delta y)^2} - Vx_{1/2,1/2}^n \left(\frac{Vy_{3/2,1/2}^n + Vy_{1/2,1/2}^n - 2Vy_{0,1/2}^n}{4\Delta x} \right) \\
& - Vy_{1/2,1/2}^n \left(\frac{Vy_{1/2,3/2}^n + Vy_{1/2,1/2}^n - 2Vy_{1/2,0}^n}{4\Delta y} \right) + \frac{Vy_{1/2,1/2}^n}{\Delta t} + \frac{Vx_{1/2,1/2}^n}{2\Delta x} Vy_{0,1/2}^{n+1} \\
& + \frac{p_{1/2,0}^{n+1}}{2\rho\Delta y} + \frac{\nu}{(\Delta y)^2} Vy_{1/2,0}^{n+1} + \frac{\nu}{(\Delta x)^2} Vy_{0,1/2}^{n+1} + \frac{Vy_{1/2,1/2}^n}{2\Delta y} Vy_{1/2,0}^{n+1}
\end{aligned}$$

$$i = N - 1; j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,M-1/2}^n}{4\Delta x} - \frac{Vy_{N-1/2,M-1/2}^n}{4\Delta y} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vy_{N-1/2,M-1/2}^{n+1} \\
& - \left(\frac{Vx_{N-1/2,M-1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vy_{N-3/2,M-1/2}^{n+1} - \left(\frac{Vy_{N-1/2,M-1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vy_{N-1/2,M-3/2}^{n+1} \\
& - \frac{p_{N-1/2,M-1/2}^{n+1}}{4\rho\Delta x} - \frac{p_{N-1/2,M-3/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{2p_{N-1/2,M}^n - p_{N-1/2,M-1/2}^n - p_{N-1/2,M-3/2}^n}{4\Delta y} \right) \\
& + \nu \left(\frac{2Vy_{N,M-1/2}^n - 3Vy_{N-1/2,M-1/2}^n + Vy_{N-3/2,M-1/2}^n}{2(\Delta x)^2} + \frac{2Vy_{N-1/2,M}^n - 3Vy_{N-1/2,M-1/2}^n}{2(\Delta y)^2} \right) \\
& + \frac{Vy_{N-1/2,M-3/2}^n}{2(\Delta y)^2} - Vx_{N-1/2,M-1/2}^n \left(\frac{2Vy_{N,M-1/2}^n - Vy_{N-1/2,M-1/2}^n - Vy_{N-3/2,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,M-1/2}^n \left(\frac{2Vy_{N-1/2,M}^n - Vy_{N-1/2,M-1/2}^n - Vy_{N-1/2,M-3/2}^n}{4\Delta y} \right) + \frac{Vy_{N-1/2,M-1/2}^n}{\Delta t} \\
& + \frac{\nu}{(\Delta x)^2} Vy_{N,M-1/2}^{n+1} - \frac{Vx_{N-1/2,M-1/2}^n}{2\Delta x} Vy_{N,M-1/2}^{n+1} - \frac{Vy_{N-1/2,M-1/2}^n}{2\Delta y} Vy_{N-1/2,M}^{n+1} \\
& + \frac{\nu}{(\Delta y)^2} Vy_{N-1/2,M}^{n+1} - \frac{p_{N-1/2,M}^{n+1}}{2\rho\Delta y}
\end{aligned}$$

$$i = 0; j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,M-1/2}^n}{4\Delta x} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} - \frac{Vy_{1/2,M-1/2}^n}{4\Delta y} \right) Vy_{1/2,M-1/2}^{n+1} \\
& + \left(\frac{Vx_{1/2,M-1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vy_{3/2,M-1/2}^{n+1} - \left(\frac{Vy_{1/2,M-1/2}^n}{4\Delta y} + \frac{\nu}{2(\Delta y)^2} \right) Vy_{1/2,M-3/2}^{n+1} \\
& - \frac{p_{1/2,M-1/2}^{n+1}}{4\rho\Delta y} - \frac{p_{1/2,M-3/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{2p_{1/2,M}^n - p_{1/2,M-1/2}^n - p_{1/2,M-3/2}^n}{4\Delta y} \right) \\
& + \nu \left(\frac{Vy_{3/2,M-1/2}^n - 3Vy_{1/2,M-1/2}^n + 2Vy_{0,M-1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{2Vy_{1/2,M}^n - 3Vy_{1/2,M-1/2}^n + Vy_{1/2,M-3/2}^n}{2(\Delta y)^2} + \frac{Vy_{1/2,M-3/2}^n}{2(\Delta y)^2} \right) \\
& + \frac{Vy_{1/2,M-1/2}^n}{\Delta t} - Vx_{1/2,M-1/2}^n \left(\frac{Vy_{3/2,M-1/2}^n + Vy_{1/2,M-1/2}^n - 2Vy_{0,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{1/2,M-1/2}^n \left(\frac{2Vy_{1/2,M}^n - Vy_{1/2,M-1/2}^n - Vy_{1/2,M-3/2}^n}{4\Delta y} \right) \\
& + \frac{Vx_{1/2,M-1/2}^n}{2\Delta x} Vy_{0,M-1/2}^{n+1} - \frac{Vy_{1/2,M-1/2}^n}{2\Delta y} Vy_{1/2,M}^{n+1} - \frac{p_{1/2,M}^{n+1}}{2\rho\Delta y} + \nu \left(\frac{Vy_{0,M-1/2}^{n+1}}{(\Delta x)^2} + \frac{Vy_{1/2,M}^{n+1}}{(\Delta y)^2} \right)
\end{aligned}$$

$$i = N - 1; j = 0$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,1/2}^n}{4\Delta x} + \frac{Vy_{N-1/2,1/2}^n}{4\Delta y} + \frac{3\nu}{2(\Delta x)^2} + \frac{3\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,1/2}^{n+1} \\
& + \left(\frac{Vy_{N-1/2,1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,3/2}^{n+1} - \left(\frac{Vx_{N-1/2,1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{N-3/2,1/2}^{n+1} \\
& + \frac{p_{N-1/2,1/2}^{n+1}}{4\rho\Delta y} + \frac{p_{N-1/2,3/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{p_{N-1/2,3/2}^n + p_{N-1/2,1/2}^n - 2p_{N-1/2,0}^n}{4\Delta y} \right) \\
& - \frac{Vy_{N-1/2,1/2}^n}{2\Delta y} Vx_{N-1/2,0}^{n+1} + \frac{p_{N-1/2,0}^{n+1}}{2\rho\Delta y} + \nu \left(\frac{2Vx_{N,1/2}^n - 3Vx_{N-1/2,1/2}^n + Vx_{N-3/2,1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{Vx_{N-1/2,3/2}^n - 3Vx_{N-1/2,1/2}^n + 2Vx_{N-1/2,0}^n}{2(\Delta y)^2} \right) + \frac{Vx_{N-1/2,1/2}^n}{2\Delta x} Vx_{N,1/2}^{n+1} \\
& - \frac{\nu}{(\Delta x)^2} Vx_{N,1/2}^{n+1} - Vx_{N-1/2,1/2}^n \left(\frac{2Vx_{N,1/2}^n - Vx_{N-1/2,1/2}^n - Vx_{N-3/2,1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,1/2}^n \left(\frac{Vx_{N-1/2,3/2}^n Vx_{N-1/2,1/2}^n - 2Vx_{N-1/2,0}^n}{4\Delta y} \right) + \frac{Vx_{N-1/2,1/2}^n}{\Delta t} + \frac{\nu}{(\Delta y)^2} Vx_{N-1/2,0}^{n+1}
\end{aligned}$$

Condiciones de Frontera

Para la 1 Ecuación se tiene:

$i = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,j+1/2}^n}{2\Delta x} + \frac{2\nu}{(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vx_{1/2,j+1/2}^{n+1} + \left(\frac{Vx_{1/2,j+1/2}^n}{6\Delta x} - \frac{2\nu}{3(\Delta x)^2} \right) Vx_{3/2,j+1/2}^{n+1} \\
& + \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{1/2,j+3/2}^{n+1} - \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{1/2,j-1/2}^{n+1} \\
& + \frac{p_{1/2,j+1/2}^{n+1}}{2\rho\Delta x} + \frac{p_{3/2,j+1/2}^{n+1}}{6\rho\Delta x} = -\frac{1}{\rho} \left(\frac{-(8/3)p_{0,j+1/2}^n + 2p_{1/2,j+1/2}^n + (2/3)p_{3/2,j+1/2}^n}{4\Delta x} \right) \\
& + \nu \left(\frac{(8/3)Vx_{0,j+1/2}^n - 4Vx_{1/2,j+1/2}^n + (4/3)Vx_{3/2,j+1/2}^n}{2(\Delta x)^2} \right. \\
& + \left. \frac{Vx_{1/2,j+3/2}^n - 2Vx_{1/2,j+1/2}^n + Vx_{1/2,j-1/2}^n}{2(\Delta y)^2} \right) + \frac{4\nu}{3(\Delta x)^2} Vx_{0,j+1/2}^{n+1} + \frac{2p_{0,j+1/2}^{n+1}}{3\rho\Delta x} \\
& - Vx_{1/2,j+1/2}^n \left(\frac{-(8/3)Vx_{0,j+1/2}^n + 2Vx_{1/2,j+1/2}^n + (2/3)Vx_{3/2,j+1/2}^n}{4\Delta x} \right) \\
& - Vy_{1/2,j+1/2}^n \left(\frac{Vx_{1/2,j+3/2}^n - Vx_{1/2,j-1/2}^n}{4\Delta y} \right) + \frac{Vx_{1/2,j+1/2}^n}{\Delta t} + Vx_{1/2,j+1/2}^n \frac{2Vx_{0,j+1/2}^{n+1}}{3\Delta x}
\end{aligned}$$

$j = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vy_{i+1/2,1/2}^n}{2\Delta y} + \frac{\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vx_{i+1/2,1/2}^{n+1} + \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vx_{i+3/2,1/2}^{n+1} \\
& - \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{i-1/2,1/2}^{n+1} + \left(\frac{Vy_{i+1/2,1/2}^n}{6\Delta y} - \frac{2\nu}{3(\Delta y)^2} \right) Vx_{i+1/2,3/2}^{n+1} + \frac{p_{i+3/2,1/2}^{n+1}}{4\rho\Delta x} \\
& - \frac{p_{i-1/2,1/2}^{n+1}}{4\rho\Delta x} = -\frac{1}{\rho} \left(\frac{p_{i+3/2,1/2}^n - p_{i-1/2,1/2}^n}{4\Delta x} \right) + \nu \left(\frac{Vx_{i+3/2,1/2}^n - 2Vx_{i+1/2,1/2}^n + Vx_{i-1/2,1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{(8/3)Vx_{i+1/2,0}^n - 4Vx_{i+1/2,1/2}^n + (4/3)Vx_{i+1/2,3/2}^n}{2(\Delta y)^2} \right) + \frac{Vx_{i+1/2,1/2}^n}{\Delta t} \\
& \quad - Vy_{i+1/2,1/2}^n \left(\frac{-(8/3)Vx_{i+1/2,0}^n + 2Vx_{i+1/2,1/2}^n + (2/3)Vx_{i+1/2,3/2}^n}{4\Delta y} \right) \\
& - Vx_{i+1/2,1/2}^n \left(\frac{Vx_{i+3/2,1/2}^n - Vx_{i-1/2,1/2}^n}{4\Delta x} \right) + \frac{2Vy_{i+1/2,1/2}^n Vx_{i+1/2,0}^{n+1}}{3\Delta y} + \frac{4\nu}{3(\Delta y)^2} Vx_{i+1/2,0}^{n+1}
\end{aligned}$$

$i = N - 1$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,j+1/2}^n}{2\Delta x} + \frac{2\nu}{(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vx_{N-1/2,j+1/2}^{n+1} - \left(\frac{Vx_{N-1/2,j+1/2}^n}{6\Delta x} \right. \\
& \left. + \frac{2\nu}{3(\Delta x)^2} \right) Vx_{N-3/2,j+1/2}^{n+1} + \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,j+3/2}^{n+1} - \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} \right. \\
& \left. + \frac{\nu}{2(\Delta y)^2} \right) Vx_{N-1/2,j-1/2}^{n+1} - \frac{p_{N-1/2,j+1/2}^{n+1}}{2\rho\Delta x} - \frac{p_{N-3/2,j+1/2}^{n+1}}{6\rho\Delta x} = -\frac{1}{\rho} \left(\frac{(8/3)p_{N,j+1/2}^n}{4\Delta x} \right. \\
& \left. - \frac{2p_{N-1/2,j+1/2}^n}{4\Delta x} - \frac{(2/3)p_{N-3/2,j+1/2}^n}{4\Delta x} \right) + \nu \left(\frac{(8/3)Vx_{N,j+1/2}^n - 4Vx_{N-1/2,j+1/2}^n}{2(\Delta x)^2} \right. \\
& \left. + \frac{(4/3)Vx_{N-3/2,j+1/2}^n + \frac{Vx_{N-1/2,j+3/2}^n - 2Vx_{N-1/2,j+1/2}^n + Vx_{N-1/2,j-1/2}^n}{2(\Delta y)^2} \right) + \frac{Vx_{N-1/2,j+1/2}^n}{\Delta t} \\
& - Vx_{N-1/2,j+1/2}^n \left(\frac{(8/3)Vx_{N,j+1/2}^n - 2Vx_{N-1/2,j+1/2}^n - (2/3)Vx_{N-3/2,j+1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,j+1/2}^n \left(\frac{Vx_{N-1/2,j+3/2}^n - Vx_{N-1/2,j-1/2}^n}{4\Delta y} \right) - \frac{2Vy_{N-1/2,j+1/2}^n Vx_{N,j+1/2}^{n+1}}{3\Delta x} \\
& \quad - \frac{2p_{N,j+1/2}^{n+1}}{3\rho\Delta x} + \frac{4\nu Vx_{N,j+1/2}^{n+1}}{3(\Delta x)^2}
\end{aligned}$$

$$j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vy_{i+1/2, M-1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta y)^2} + \frac{\nu}{(\Delta x)^2} \right) Vx_{i+1/2, M-1/2}^{n+1} + \left(\frac{Vx_{i+1/2, M-1/2}^n}{4\Delta x} \right. \\
& \quad \left. - \frac{\nu}{2(\Delta x)^2} \right) Vx_{i+3/2, M-1/2}^{n+1} - \left(\frac{Vx_{i+1/2, M-1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vx_{i-1/2, M-1/2}^{n+1} \\
& \quad - \left(\frac{Vy_{i+1/2, M-1/2}^n}{6\Delta y} + \frac{2\nu}{3(\Delta y)^2} \right) Vx_{i+1/2, M-3/2}^{n+1} + \frac{p_{i+3/2, M-1/2}^{n+1}}{4\rho\Delta x} - \frac{p_{i-1/2, M-1/2}^{n+1}}{4\rho\Delta x} = \\
& -\frac{1}{\rho} \left(\frac{p_{i+3/2, M-1/2}^n - p_{i-1/2, M-1/2}^n}{4\Delta x} \right) + \nu \left(\frac{Vx_{i+3/2, M-1/2}^n - 2Vx_{i+1/2, M-1/2}^n + Vx_{i-1/2, M-1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{(8/3)Vx_{i+1/2, M}^n - 4Vx_{i+1/2, M-1/2}^n + (4/3)Vx_{i+1/2, M-3/2}^n}{2(\Delta y)^2} \right) - \frac{2Vy_{i+1/2, M-1/2}^n}{3\Delta y} Vx_{i+1/2, M}^{n+1} \\
& \quad - \frac{4\nu Vx_{i+1/2, M}^{n+1}}{3(\Delta y)^2} - Vx_{i+1/2, M-1/2}^n \left(\frac{Vx_{i+3/2, M-1/2}^n - Vx_{i-1/2, M-1/2}^n}{4\Delta x} \right) \\
& -Vy_{i+1/2, M-1/2}^n \left(\frac{(8/3)Vx_{i+1/2, M}^n - 2Vx_{i+1/2, M-1/2}^n - (2/3)Vx_{i+1/2, M-3/2}^n}{4\Delta y} \right) + \frac{Vx_{i+1/2, M-1/2}^n}{\Delta t}
\end{aligned}$$

$$i = 0; j = 0$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2, 1/2}^n}{2\Delta x} + \frac{Vy_{1/2, 1/2}^n}{2\Delta x} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vx_{1/2, 1/2}^{n+1} + \left(\frac{Vx_{1/2, 1/2}^n}{6\Delta x} \right. \\
& \quad \left. - \frac{2\nu}{3(\Delta x)^2} \right) Vx_{3/2, 1/2}^{n+1} + \left(\frac{Vy_{1/2, 1/2}^n}{6\Delta x} - \frac{2\nu}{3(\Delta y)^2} \right) Vx_{1/2, 3/2}^{n+1} + \frac{p_{1/2, 1/2}^{n+1}}{2\rho\Delta x} + \frac{p_{3/2, 1/2}^{n+1}}{6\rho\Delta x} = \\
& -\frac{1}{\rho} \left(\frac{-(8/3)p_{0, 1/2}^n + 2p_{1/2, 1/2}^n + (2/3)p_{3/2, 1/2}^n}{4\Delta x} \right) + \nu \left(\frac{(8/3)Vx_{0, 1/2}^n - 4Vx_{1/2, 1/2}^n + (4/3)Vx_{3/2, 1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{(8/3)Vx_{1/2, 0}^n - 4Vx_{1/2, 1/2}^n + (4/3)Vx_{1/2, 3/2}^n}{2(\Delta y)^2} \right) + \frac{2Vx_{1/2, 1/2}^n}{3\Delta x} Vx_{0, 1/2}^{n+1} + \frac{2Vy_{1/2, 1/2}^n}{3\Delta x} Vx_{1/2, 0}^{n+1} \\
& \quad + \frac{2p_{0, 1/2}^{n+1}}{3\rho\Delta x} + \frac{4\nu Vx_{0, 1/2}^{n+1}}{3(\Delta x)^2} - Vx_{1/2, 1/2}^n \left(\frac{-(8/3)Vx_{0, 1/2}^n + 2Vx_{1/2, 1/2}^n + (2/3)Vx_{3/2, 1/2}^n}{4\Delta x} \right) \\
& \quad + \frac{4\nu Vx_{1/2, 0}^{n+1}}{3(\Delta y)^2} - Vy_{1/2, 1/2}^n \left(\frac{-(8/3)Vx_{1/2, 0}^n + 2Vx_{1/2, 1/2}^n + (2/3)Vx_{1/2, 3/2}^n}{4\Delta x} \right) + \frac{Vx_{1/2, 1/2}^n}{\Delta t}
\end{aligned}$$

$$i = N - 1; j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,M-1/2}^n}{2\Delta x} - \frac{Vy_{N-1/2,M-1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vx_{N-1/2,M-1/2}^{n+1} \\
& - \left(\frac{Vx_{N-1/2,M-1/2}^n}{6\Delta x} + \frac{2\nu}{3(\Delta x)^2} \right) Vx_{N-3/2,M-1/2}^{n+1} - \left(\frac{Vy_{N-1/2,M-1/2}^n}{6\Delta y} + \frac{2\nu}{3(\Delta y)^2} \right) Vy_{N-1/2,M-3/2}^{n+1} \\
& - \frac{p_{N-1/2,M-1/2}^{n+1}}{2\rho\Delta x} - \frac{p_{N-3/2,M-1/2}^{n+1}}{6\rho\Delta x} = -\frac{1}{\rho} \left(\frac{(8/3)p_{N,M-1/2}^n - 2p_{N-1/2,M-1/2}^n - (2/3)p_{N-3/2,M-1/2}^n}{4\Delta x} \right) \\
& + \nu \left(\frac{(8/3)Vx_{N,M-1/2}^n - 4Vx_{N-1/2,M-1/2}^n + (4/3)Vx_{N-3/2,M-1/2}^n}{2(\Delta x)^2} + \frac{(8/3)Vx_{N-1/2,M}^n}{2(\Delta y)^2} \right. \\
& \quad \left. - \frac{4Vy_{N-1/2,M-1/2}^n + (4/3)Vx_{N-1/2,M-3/2}^n}{2(\Delta y)^2} \right) \\
& - Vx_{N-1/2,M-1/2}^n \left(\frac{(8/3)Vx_{N,M-1/2}^n - 2Vx_{N-1/2,M-1/2}^n - (2/3)Vx_{N-3/2,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,M-1/2}^n \left(\frac{(8/3)Vx_{N-1/2,M}^n - 2Vx_{N-1/2,M-1/2}^n - (2/3)Vx_{N-1/2,M-3/2}^n}{4\Delta y} \right) \\
& \quad + \frac{4\nu}{3(\Delta x)^2} Vx_{N,M-1/2}^{n+1} + \frac{4\nu}{3(\Delta y)^2} Vy_{N-1/2,M}^{n+1} + \frac{Vx_{N-1/2,M-1/2}^n}{\Delta t} \\
& - \frac{2Vx_{N-1/2,M-1/2}^n}{3\Delta x} Vx_{N,M-1/2}^{n+1} - \frac{2Vy_{N-1/2,M-1/2}^n}{3\Delta y} Vy_{N-1/2,M}^{n+1} - \frac{2}{3\rho\Delta x} p_{N,M-1/2}^{n+1}
\end{aligned}$$

$$i = 0; j = M - 1$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,M-1/2}^n}{2\Delta x} - \frac{Vy_{1/2,M-1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vx_{1/2,M-1/2}^{n+1} \\
& + \left(\frac{Vx_{1/2,M-1/2}^n}{6\Delta x} - \frac{2\nu}{3(\Delta x)^2} \right) Vx_{3/2,M-1/2}^{n+1} - \left(\frac{Vy_{1/2,M-1/2}^n}{6\Delta y} + \frac{2\nu}{3(\Delta y)^2} \right) Vx_{1/2,M-3/2}^{n+1} \\
& + \frac{p_{1/2,M-1/2}^{n+1}}{2\rho\Delta x} + \frac{p_{3/2,M-1/2}^{n+1}}{6\rho\Delta x} = -\frac{1}{\rho} \left(\frac{-(8/3)p_{0,M-1/2}^n + 2p_{1/2,M-1/2}^n + (2/3)p_{3/2,M-1/2}^n}{4\Delta x} \right) \\
& + \nu \left(\frac{(8/3)Vx_{0,M-1/2}^n - 4Vx_{1/2,M-1/2}^n + (4/3)Vx_{3/2,M-1/2}^n}{2(\Delta x)^2} + \frac{(8/3)Vx_{1/2,M}^n}{2(\Delta y)^2} \right. \\
& \quad \left. - \frac{4Vx_{1/2,M-1/2}^n + (4/3)Vx_{1/2,M-3/2}^n}{2(\Delta y)^2} \right) + \frac{4\nu Vx_{0,M-1/2}^{n+1}}{3(\Delta x)^2} + \frac{4\nu Vx_{1/2,M}^{n+1}}{3(\Delta y)^2} \\
& - Vx_{1/2,M-1/2}^n \left(\frac{-(8/3)Vx_{0,M-1/2}^n + 2Vx_{1/2,M-1/2}^n + (2/3)Vx_{3/2,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{1/2,M-1/2}^n \left(\frac{(8/3)Vx_{1/2,M}^n - 2Vx_{1/2,M-1/2}^n - (2/3)Vx_{1/2,M-3/2}^n}{4\Delta y} \right) \\
& + \frac{Vx_{1/2,M-1/2}^n}{\Delta t} + \frac{2Vx_{1/2,M-1/2}^n}{3\Delta x} Vx_{0,M-1/2}^{n+1} - \frac{2Vy_{1/2,M-1/2}^n}{3\Delta y} Vx_{1/2,M}^{n+1} + \frac{2}{3\rho\Delta x} p_{0,M-1/2}^{n+1}
\end{aligned}$$

$$i = N - 1; j = 0$$

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,1/2}^n}{2\Delta x} + \frac{Vy_{N-1/2,1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vx_{N-1/2,1/2}^{n+1} \\
& - \left(\frac{Vx_{N-1/2,1/2}^n}{6\Delta x} + \frac{2\nu}{3(\Delta x)^2} \right) Vx_{N-3/2,1/2}^{n+1} + \left(\frac{Vy_{N-1/2,1/2}^n}{6\Delta y} - \frac{2\nu}{3(\Delta y)^2} \right) Vx_{N-1/2,3/2}^{n+1}
\end{aligned}$$

$$\begin{aligned}
& -\frac{p_{N-1/2,1/2}^{n+1}}{2\rho\Delta x} - \frac{p_{N-3/2,1/2}^{n+1}}{6\rho\Delta x} = -\frac{1}{\rho} \left(\frac{(8/3)p_{N,1/2}^n - 2p_{N-1/2,1/2}^n - (2/3)p_{N-3/2,1/2}^n}{4\Delta x} \right) \\
& -\frac{2p_{N,1/2}^{n+1}}{3\rho\Delta x} + \nu \left(\frac{(8/3)Vx_{N,1/2}^n - 4Vx_{N-1/2,1/2}^n + (4/3)Vx_{N-3/2,1/2}^n}{2(\Delta x)^2} \right) \\
& + \frac{(8/3)Vx_{N-1/2,0}^n - 4Vx_{N-1/2,1/2}^n + (4/3)Vx_{N-1/2,3/2}^n}{2(\Delta y)^2} + \frac{4\nu}{3(\Delta y)^2} Vx_{N-1/2,0}^{n+1} \\
& -Vx_{N-1/2,1/2}^n \left(\frac{(8/3)Vx_{N,1/2}^n - 2Vx_{N-1/2,1/2}^n - (2/3)Vx_{N-3/2,1/2}^n}{4\Delta x} \right) \\
& -Vy_{N-1/2,1/2}^n \left(\frac{-(8/3)Vx_{N-1/2,0}^n + 2Vx_{N-1/2,1/2}^n + (2/3)Vx_{N-1/2,3/2}^n}{4\Delta y} \right) \\
& + \frac{Vx_{N-1/2,1/2}^n}{\Delta t} - \frac{2Vx_{N-1/2,1/2}^n}{3\Delta x} Vx_{N,1/2}^{n+1} + \frac{2Vy_{N-1/2,1/2}^n}{3\Delta y} Vy_{N-1/2,0}^{n+1} + \frac{4\nu}{3(\Delta x)^2} Vx_{N,1/2}^{n+1}
\end{aligned}$$

Para la 2 ecuación se tiene:

$i = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,j+1/2}^n}{2\Delta x} + \frac{2\nu}{(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vy_{1/2,j+1/2}^{n+1} + \left(\frac{Vx_{1/2,j+1/2}^n}{6\Delta x} - \frac{2\nu}{3(\Delta x)^2} \right) Vy_{3/2,j+1/2}^{n+1} \\
& + \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{1/2,j+3/2}^{n+1} - \left(\frac{Vy_{1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{1/2,j-1/2}^{n+1} + \frac{p_{1/2,j+3/2}^{n+1}}{4\rho\Delta y} \\
& - \frac{p_{1/2,j-1/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{p_{1/2,j+3/2}^n - p_{1/2,j-1/2}^n}{4\Delta y} \right) + \nu \left(\frac{(8/3)Vy_{0,j+1/2}^n - 4Vy_{1/2,j+1/2}^n}{2(\Delta x)^2} \right) \\
& + \frac{(4/3)Vy_{3/2,j+1/2}^n}{2(\Delta x)^2} + \frac{Vy_{1/2,j+3/2}^n - 2Vy_{1/2,j+1/2}^n + Vy_{1/2,j-1/2}^n}{2(\Delta y)^2} + \frac{4\nu}{3(\Delta x)^2} Vy_{0,j+1/2}^{n+1} \\
& -Vx_{1/2,j+1/2}^n \left(\frac{-(8/3)Vy_{0,j+1/2}^n + 2Vy_{1/2,j+1/2}^n + (2/3)Vy_{3/2,j+1/2}^n}{4\Delta x} \right) \\
& -Vy_{1/2,j+1/2}^n \left(\frac{Vy_{1/2,j+3/2}^n - Vy_{1/2,j-1/2}^n}{4\Delta y} \right) + \frac{Vy_{1/2,j+1/2}^n}{\Delta t} + Vx_{1/2,j+1/2}^n \frac{2Vy_{0,j+1/2}^{n+1}}{3\Delta x}
\end{aligned}$$

$j = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vy_{i+1/2,1/2}^n}{2\Delta y} + \frac{\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vy_{i+1/2,1/2}^{n+1} + \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vy_{i+3/2,1/2}^{n+1} \\
& - \left(\frac{Vx_{i+1/2,1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vy_{i-1/2,1/2}^{n+1} + \left(\frac{Vy_{i+1/2,1/2}^n}{6\Delta y} - \frac{2\nu}{3(\Delta y)^2} \right) Vy_{i+1/2,3/2}^{n+1} + \frac{p_{i+1/2,1/2}^{n+1}}{2\rho\Delta y} \\
& \quad + \frac{p_{i+1/2,3/2}^{n+1}}{6\rho\Delta y} = -\frac{1}{\rho} \left(\frac{-(8/3)p_{i+1/2,0}^n + 2p_{i+1/2,1/2}^n + (2/3)p_{i+1/2,3/2}^n}{4\Delta y} \right) \\
& \quad + \nu \left(\frac{Vy_{i+3/2,1/2}^n - 2Vy_{i+1/2,1/2}^n + Vy_{i-1/2,1/2}^n}{2(\Delta x)^2} + \frac{(8/3)Vy_{i+1/2,0}^n - 4Vy_{i+1/2,1/2}^n}{2(\Delta y)^2} \right. \\
& \quad + \left. \frac{(4/3)Vy_{i+1/2,3/2}^n}{2(\Delta y)^2} \right) + \frac{2Vy_{i+1/2,1/2}^n}{3\Delta y} Vy_{i+1/2,0}^{n+1} + \frac{4\nu}{3(\Delta y)^2} Vy_{i+1/2,0}^{n+1} + \frac{Vy_{i+1/2,1/2}^n}{\Delta t} \\
& \quad + \frac{2p_{i+1/2,0}^{n+1}}{3\rho\Delta y} - Vy_{i+1/2,1/2}^n \left(\frac{-(8/3)Vy_{i+1/2,0}^n + 2Vy_{i+1/2,1/2}^n + (2/3)Vy_{i+1/2,3/2}^n}{4\Delta y} \right) \\
& \quad \quad \quad - Vx_{i+1/2,1/2}^n \left(\frac{Vy_{i+3/2,1/2}^n - Vy_{i-1/2,1/2}^n}{4\Delta x} \right)
\end{aligned}$$

$i = N - 1$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,j+1/2}^n}{2\Delta x} + \frac{2\nu}{(\Delta x)^2} + \frac{\nu}{(\Delta y)^2} \right) Vy_{N-1/2,j+1/2}^{n+1} - \left(\frac{Vx_{N-1/2,j+1/2}^n}{6\Delta x} \right. \\
& \quad + \left. \frac{2\nu}{3(\Delta x)^2} \right) Vy_{N-3/2,j+1/2}^{n+1} + \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} - \frac{\nu}{2(\Delta y)^2} \right) Vy_{N-1/2,j+3/2}^{n+1} - \left(\frac{Vy_{N-1/2,j+1/2}^n}{4\Delta y} \right. \\
& \quad + \left. \frac{\nu}{2(\Delta y)^2} \right) Vy_{N-1/2,j-1/2}^{n+1} + \frac{p_{N-1/2,j+3/2}^{n+1}}{4\rho\Delta y} - \frac{p_{N-1/2,j-1/2}^{n+1}}{4\rho\Delta y} = -\frac{1}{\rho} \left(\frac{p_{N-1/2,j+3/2}^n}{4\Delta y} \right. \\
& \quad - \left. \frac{p_{N-1/2,j-1/2}^n}{4\Delta y} \right) + \nu \left(\frac{(8/3)Vy_{N,j+1/2}^n - 4Vy_{N-1/2,j+1/2}^n + (4/3)Vy_{N-3/2,j+1/2}^n}{2(\Delta x)^2} \right. \\
& \quad + \left. \frac{Vy_{N-1/2,j+3/2}^n - 2Vy_{N-1/2,j+1/2}^n + Vy_{N-1/2,j-1/2}^n}{2(\Delta y)^2} \right) + \frac{Vy_{N-1/2,j+1/2}^n}{\Delta t} + \frac{4\nu Vy_{N,j+1/2}^{n+1}}{3(\Delta x)^2} \\
& \quad - Vx_{N-1/2,j+1/2}^n \left(\frac{(8/3)Vy_{N,j+1/2}^n - 2Vy_{N-1/2,j+1/2}^n - (2/3)Vy_{N-3/2,j+1/2}^n}{4\Delta x} \right) \\
& \quad - Vy_{N-1/2,j+1/2}^n \left(\frac{Vy_{N-1/2,j+3/2}^n - Vy_{N-1/2,j-1/2}^n}{4\Delta y} \right) - \frac{2Vy_{N-1/2,j+1/2}^n}{3\Delta x} Vy_{N,j+1/2}^{n+1} - \frac{2p_{N,j+1/2}^{n+1}}{3\rho\Delta y}
\end{aligned}$$

$j = M - 1$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vy_{i+1/2,M-1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta y)^2} + \frac{\nu}{(\Delta x)^2} \right) Vy_{i+1/2,M-1/2}^{n+1} \\
& + \left(\frac{Vx_{i+1/2,M-1/2}^n}{4\Delta x} - \frac{\nu}{2(\Delta x)^2} \right) Vy_{i+3/2,M-1/2}^{n+1} - \left(\frac{Vx_{i+1/2,M-1/2}^n}{4\Delta x} + \frac{\nu}{2(\Delta x)^2} \right) Vy_{i-1/2,M-1/2}^{n+1} \\
& - \left(\frac{Vy_{i+1/2,M-1/2}^n}{6\Delta y} + \frac{2\nu}{3(\Delta y)^2} \right) Vy_{i+1/2,M-3/2}^{n+1} - \frac{p_{i+1/2,M-1/2}^{n+1}}{2\rho\Delta y} - \frac{p_{i+1/2,M-3/2}^{n+1}}{6\rho\Delta y} = \\
& - \frac{1}{\rho} \left(\frac{(8/3)p_{i+1/2,M}^n - 2p_{i+1/2,M-1/2}^n - (2/3)p_{i+1/2,M-3/2}^n}{4\Delta y} \right) + \frac{Vy_{i+1/2,M-1/2}^n}{\Delta t} \\
& - \frac{2Vy_{i+1/2,M-1/2}^n}{3\Delta y} Vy_{i+1/2,M}^{n+1} + \nu \left(\frac{Vy_{i+3/2,M-1/2}^n - 2Vy_{i+1/2,M-1/2}^n + Vy_{i-1/2,M-1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{(8/3)Vy_{i+1/2,M}^n - 4Vy_{i+1/2,M-1/2}^n + (4/3)Vy_{i+1/2,M-3/2}^n}{2(\Delta y)^2} \right) \\
& - \frac{4\nu Vy_{i+1/2,M}^{n+1}}{3(\Delta y)^2} - Vx_{i+1/2,M-1/2}^n \left(\frac{Vy_{i+3/2,M-1/2}^n - Vy_{i-1/2,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{i+1/2,M-1/2}^n \left(\frac{(8/3)Vy_{i+1/2,M}^n - 2Vy_{i+1/2,M-1/2}^n - (2/3)Vy_{i+1/2,M-3/2}^n}{4\Delta y} \right)
\end{aligned}$$

$i = 0; j = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,1/2}^n}{2\Delta x} + \frac{Vy_{1/2,1/2}^n}{2\Delta x} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vy_{1/2,1/2}^{n+1} + \left(\frac{Vx_{1/2,1/2}^n}{6\Delta x} - \frac{2\nu}{3(\Delta x)^2} \right) Vy_{3/2,1/2}^{n+1} \\
& + \left(\frac{Vy_{1/2,1/2}^n}{6\Delta x} - \frac{2\nu}{3(\Delta y)^2} \right) Vy_{1/2,3/2}^{n+1} + \frac{p_{1/2,1/2}^{n+1}}{2\rho\Delta y} + \frac{p_{1/2,3/2}^{n+1}}{6\rho\Delta y} = -\frac{1}{\rho} \left(\frac{-(8/3)p_{1/2,0}^n}{4\Delta y} \right. \\
& \quad \left. + \frac{2p_{1/2,1/2}^n + (2/3)p_{1/2,3/2}^n}{4\Delta y} \right) + \nu \left(\frac{(8/3)Vy_{0,1/2}^n - 4Vy_{1/2,1/2}^n + (4/3)Vy_{3/2,1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{(8/3)Vy_{1/2,0}^n - 4Vy_{1/2,1/2}^n + (4/3)Vy_{1/2,3/2}^n}{2(\Delta y)^2} \right) + \frac{2Vx_{1/2,1/2}^n}{3\Delta x} Vy_{0,1/2}^{n+1} + \frac{2Vy_{1/2,1/2}^n}{3\Delta x} Vy_{1/2,0}^{n+1} \\
& + \frac{2p_{1/2,0}^{n+1}}{3\rho\Delta y} + \frac{4\nu Vy_{0,1/2}^{n+1}}{3(\Delta x)^2} - Vx_{1/2,1/2}^n \left(\frac{-(8/3)Vy_{0,1/2}^n + 2Vy_{1/2,1/2}^n + (2/3)Vy_{3/2,1/2}^n}{4\Delta x} \right) \\
& + \frac{4\nu Vy_{1/2,0}^{n+1}}{3(\Delta y)^2} - Vy_{1/2,1/2}^n \left(\frac{-(8/3)Vy_{1/2,0}^n + 2Vy_{1/2,1/2}^n + (2/3)Vy_{1/2,3/2}^n}{4\Delta x} \right) + \frac{Vy_{1/2,1/2}^n}{\Delta t}
\end{aligned}$$

$i = N - 1; j = M - 1$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,M-1/2}^n}{2\Delta x} - \frac{Vy_{N-1/2,M-1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vy_{N-1/2,M-1/2}^{n+1} \\
& - \left(\frac{Vx_{N-1/2,M-1/2}^n}{6\Delta x} + \frac{2\nu}{3(\Delta x)^2} \right) Vy_{N-3/2,M-1/2}^{n+1} - \left(\frac{Vy_{N-1/2,M-1/2}^n}{6\Delta y} + \frac{2\nu}{3(\Delta y)^2} \right) Vy_{N-1/2,M-3/2}^{n+1} \\
& - \frac{p_{N-1/2,M-1/2}^{n+1}}{2\rho\Delta y} - \frac{p_{N-1/2,M-3/2}^{n+1}}{6\rho\Delta y} = -\frac{1}{\rho} \left(\frac{(8/3)p_{N-1/2,M}^n - 2p_{N-1/2,M-1/2}^n - (2/3)p_{N-1/2,M-3/2}^n}{4\Delta y} \right) \\
& + \nu \left(\frac{(8/3)Vy_{N,M-1/2}^n - 4Vy_{N-1/2,M-1/2}^n + (4/3)Vy_{N-3/2,M-1/2}^n}{2(\Delta x)^2} \right. \\
& \left. + \frac{(8/3)Vy_{N-1/2,M}^n - 4Vy_{N-1/2,M-1/2}^n + (4/3)Vy_{N-1/2,M-3/2}^n}{2(\Delta y)^2} \right) \\
& - Vx_{N-1/2,M-1/2}^n \left(\frac{(8/3)Vy_{N,M-1/2}^n - 2Vy_{N-1/2,M-1/2}^n - (2/3)Vy_{N-3/2,M-1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,M-1/2}^n \left(\frac{(8/3)Vy_{N-1/2,M}^n - 2Vy_{N-1/2,M-1/2}^n - (2/3)Vy_{N-1/2,M-3/2}^n}{4\Delta y} \right) \\
& - \frac{2Vx_{N-1/2,M-1/2}^n Vy_{N,M-1/2}^{n+1}}{3\Delta x} - \frac{2Vy_{N-1/2,M-1/2}^n Vy_{N-1/2,M}^{n+1}}{3\Delta y} - \frac{2}{3\rho\Delta y} p_{N-1/2,M}^{n+1} \\
& + \frac{4\nu}{3(\Delta x)^2} Vy_{N,M-1/2}^{n+1} + \frac{4\nu}{3(\Delta y)^2} Vy_{N-1/2,M}^{n+1} + \frac{Vy_{N-1/2,M-1/2}^n}{\Delta t}
\end{aligned}$$

$i = 0; j = M - 1$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} + \frac{Vx_{1/2,M-1/2}^n}{2\Delta x} - \frac{Vy_{1/2,M-1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vy_{1/2,M-1/2}^{n+1} \\
& + \left(\frac{Vx_{1/2,M-1/2}^n}{6\Delta x} - \frac{2\nu}{3(\Delta x)^2} \right) Vy_{3/2,M-1/2}^{n+1} - \left(\frac{Vy_{1/2,M-1/2}^n}{6\Delta y} + \frac{2\nu}{3(\Delta y)^2} \right) Vy_{1/2,M-3/2}^{n+1} \\
& - \frac{p_{1/2,M-1/2}^{n+1}}{2\rho\Delta y} - \frac{p_{1/2,M-3/2}^{n+1}}{6\rho\Delta y} = -\frac{1}{\rho} \left(\frac{(8/3)p_{1/2,M}^n - 2p_{1/2,M-1/2}^n - (2/3)p_{1/2,M-3/2}^n}{4\Delta y} \right)
\end{aligned}$$

$$\begin{aligned}
& +\nu \left(\frac{(8/3)Vy_{0,M-1/2}^n - 4Vy_{1/2,M-1/2}^n + (4/3)Vy_{3/2,M-1/2}^n}{2(\Delta x)^2} \right. \\
& \quad \left. + \frac{(8/3)Vy_{1/2,M}^n - 4Vy_{1/2,M-1/2}^n + (4/3)Vy_{1/2,M-3/2}^n}{2(\Delta y)^2} \right) \\
& -Vy_{1/2,M-1/2}^n \left(\frac{-(8/3)Vy_{0,M-1/2}^n + 2Vy_{1/2,M-1/2}^n + (2/3)Vy_{3/2,M-1/2}^n}{4\Delta x} \right) \\
& \quad -Vy_{1/2,M-1/2}^n \left(\frac{(8/3)Vy_{1/2,M}^n - 2Vy_{1/2,M-1/2}^n - (2/3)Vy_{1/2,M-3/2}^n}{4\Delta y} \right) \\
& + \frac{Vy_{1/2,M-1/2}^n}{\Delta t} + \frac{2Vx_{1/2,M-1/2}^n}{3\Delta x} Vy_{0,M-1/2}^{n+1} - \frac{2Vy_{1/2,M-1/2}^n}{3\Delta y} Vy_{1/2,M}^{n+1} + \frac{2}{3\rho\Delta y} p_{1/2,M}^{n+1} \\
& \quad + \frac{4\nu Vy_{0,M-1/2}^{n+1}}{3(\Delta x)^2} + \frac{4\nu Vy_{1/2,M}^{n+1}}{3(\Delta y)^2}
\end{aligned}$$

$i = N - 1; j = 0$:

$$\begin{aligned}
& \left(\frac{1}{\Delta t} - \frac{Vx_{N-1/2,1/2}^n}{2\Delta x} + \frac{Vy_{N-1/2,1/2}^n}{2\Delta y} + \frac{2\nu}{(\Delta x)^2} + \frac{2\nu}{(\Delta y)^2} \right) Vy_{N-1/2,1/2}^{n+1} \\
& - \left(\frac{Vx_{N-1/2,1/2}^n}{6\Delta x} + \frac{2\nu}{3(\Delta x)^2} \right) Vy_{N-3/2,1/2}^{n+1} + \left(\frac{Vy_{N-1/2,1/2}^n}{6\Delta y} - \frac{2\nu}{3(\Delta y)^2} \right) Vy_{N-1/2,3/2}^{n+1} \\
& + \frac{p_{N-1/2,1/2}^{n+1}}{2\rho\Delta y} + \frac{p_{N-1/2,3/2}^{n+1}}{6\rho\Delta y} = -\frac{1}{\rho} \left(\frac{-(8/3)p_{N-1/2,0}^n + 2p_{N-1/2,1/2}^n + (2/3)p_{N-1/2,3/2}^n}{4\Delta y} \right) \\
& + \nu \left(\frac{(8/3)Vy_{N,1/2}^n - 4Vy_{N-1/2,1/2}^n + (4/3)Vy_{N-3/2,1/2}^n}{2(\Delta x)^2} \right) \\
& + \frac{(8/3)Vy_{N-1/2,0}^n - 4Vy_{N-1/2,1/2}^n + (4/3)Vy_{N-1/2,3/2}^n}{2(\Delta y)^2} \\
& - Vy_{N-1/2,1/2}^n \left(\frac{(8/3)Vy_{N,1/2}^n - 2Vy_{N-1/2,1/2}^n - (2/3)Vy_{N-3/2,1/2}^n}{4\Delta x} \right) \\
& - Vy_{N-1/2,1/2}^n \left(\frac{-(8/3)Vy_{N-1/2,0}^n + 2Vy_{N-1/2,1/2}^n + (2/3)Vy_{N-1/2,3/2}^n}{4\Delta y} \right) \\
& + \frac{Vy_{N-1/2,1/2}^n}{\Delta t} - \frac{2Vx_{N-1/2,1/2}^n}{3\Delta x} Vy_{N,1/2}^{n+1} + \frac{2Vy_{N-1/2,1/2}^n}{3\Delta y} Vy_{N-1/2,0}^{n+1} + \frac{4\nu}{3(\Delta x)^2} Vy_{N,1/2}^{n+1} \\
& - \frac{2p_{N,1/2}^{n+1}}{3\rho\Delta y} + \frac{4\nu}{3(\Delta y)^2} Vy_{N-1/2,0}^{n+1}
\end{aligned}$$

Ecuación de Compresibilidad Artificial

$$\begin{aligned}
& \frac{1}{\beta\Delta t} p_{i+1/2,j+1/2}^{n+1} + \frac{1}{4\Delta x} vx_{i+3/2,j+1/2}^{n+1} - \frac{1}{4\Delta x} vx_{i-1/2,j+1/2}^{n+1} + \frac{1}{4\Delta y} vy_{i+1/2,j+3/2}^{n+1} \\
& - \frac{1}{4\Delta y} vy_{i+1/2,j-1/2}^{n+1} = \frac{1}{\beta\Delta t} p_{i+1/2,j+1/2}^n - \frac{1}{4\Delta x} vx_{i+3/2,j+1/2}^n + \frac{1}{4\Delta x} vx_{i-1/2,j+1/2}^n \\
& - \frac{1}{4\Delta y} vy_{i+1/2,j+3/2}^n + \frac{1}{4\Delta y} vy_{i+1/2,j-1/2}^n
\end{aligned}$$

Aproximaciones en la Frontera

$i = 0$:

$$\begin{aligned} & \frac{p_{1/2,j+1/2}^{n+1}}{\beta\Delta t} + \frac{vx_{3/2,j+1/2}^{n+1}}{4\Delta x} + \frac{vx_{1/2,j+1/2}^{n+1}}{4\Delta x} + \frac{vy_{1/2,j+3/2}^{n+1}}{4\Delta y} + \frac{vy_{1/2,j+1/2}^{n+1}}{4\Delta y} \\ & - \frac{vy_{1/2,j+1/2}^{n+1}}{4\Delta y} - \frac{vy_{1/2,j-1/2}^{n+1}}{4\Delta y} = \frac{p_{1/2,j+1/2}^n}{\beta\Delta t} + \frac{vx_{0,j+1/2}^{n+1}}{2\Delta x} - \frac{vx_{3/2,j+1/2}^n}{4\Delta x} - \frac{vx_{1/2,j+1/2}^n}{4\Delta x} \\ & + \frac{vx_{0,j+1/2}^n}{2\Delta x} - \frac{vy_{1/2,j+3/2}^n}{4\Delta y} - \frac{vy_{1/2,j+1/2}^n}{4\Delta y} + \frac{vy_{1/2,j+1/2}^n}{4\Delta y} + \frac{vy_{1/2,j-1/2}^n}{4\Delta y} \end{aligned}$$

$j = 0$:

$$\begin{aligned} & \frac{p_{i+1/2,1/2}^{n+1}}{\beta\Delta t} + \frac{vx_{i+3/2,1/2}^{n+1}}{4\Delta x} + \frac{vx_{i+1/2,1/2}^{n+1}}{4\Delta x} - \frac{vx_{i+1/2,1/2}^{n+1}}{4\Delta x} - \frac{vx_{i-1/2,1/2}^{n+1}}{4\Delta x} \\ & + \frac{vy_{i+1/2,3/2}^{n+1}}{4\Delta y} + \frac{vy_{i+1/2,1/2}^{n+1}}{4\Delta y} = \frac{p_{i+1/2,1/2}^n}{\beta\Delta t} + \frac{vy_{i+1/2,0}^{n+1}}{2\Delta y} + \frac{vy_{i+1/2,0}^n}{2\Delta y} - \frac{vy_{i+1/2,3/2}^n}{4\Delta y} \\ & - \frac{vy_{i+1/2,1/2}^n}{4\Delta y} - \frac{vx_{i+3/2,1/2}^n}{4\Delta x} - \frac{vx_{i+1/2,1/2}^n}{4\Delta x} + \frac{vx_{i+1/2,1/2}^n}{4\Delta x} + \frac{vx_{i-1/2,1/2}^n}{4\Delta x} \end{aligned}$$

$i = N - 1$:

$$\begin{aligned} & \frac{p_{N-1/2,j+1/2}^{n+1}}{\beta\Delta t} - \frac{vx_{N-3/2,j+1/2}^{n+1}}{4\Delta x} - \frac{vx_{N-1/2,j+1/2}^{n+1}}{4\Delta x} + \frac{vy_{N-1/2,j+3/2}^{n+1}}{4\Delta y} + \frac{vy_{N-1/2,j+1/2}^{n+1}}{4\Delta y} \\ & - \frac{vy_{N-1/2,j+1/2}^{n+1}}{4\Delta y} - \frac{vy_{N-1/2,j-1/2}^{n+1}}{4\Delta y} = \frac{p_{N-1/2,j+1/2}^n}{\beta\Delta t} - \frac{vx_{N,j+1/2}^{n+1}}{2\Delta x} - \frac{vx_{N,j+1/2}^n}{2\Delta x} + \frac{vx_{N-3/2,j+1/2}^n}{4\Delta x} \\ & + \frac{vx_{N-1/2,j+1/2}^n}{4\Delta x} - \frac{vy_{N-1/2,j+3/2}^n}{4\Delta y} - \frac{vy_{N-1/2,j+1/2}^n}{4\Delta y} + \frac{vy_{N-1/2,j+1/2}^n}{4\Delta y} + \frac{vy_{N-1/2,j-1/2}^n}{4\Delta y} \end{aligned}$$

$j = M - 1$:

$$\begin{aligned} & \frac{p_{i+1/2,M-1/2}^{n+1}}{\beta\Delta t} + \frac{vx_{i+3/2,M-1/2}^{n+1}}{4\Delta x} + \frac{vx_{i+1/2,M-1/2}^{n+1}}{4\Delta x} - \frac{vx_{i+1/2,M-1/2}^{n+1}}{4\Delta x} - \frac{vx_{i-1/2,M-1/2}^{n+1}}{4\Delta x} \\ & - \frac{vy_{i+1/2,M-3/2}^{n+1}}{4\Delta y} - \frac{vy_{i+1/2,M-1/2}^{n+1}}{4\Delta y} = \frac{p_{i+1/2,M-1/2}^n}{\beta\Delta t} - \frac{vx_{i+3/2,M-1/2}^n}{4\Delta x} - \frac{vx_{i+1/2,M-1/2}^n}{4\Delta x} \\ & + \frac{vx_{i+1/2,M-1/2}^n}{4\Delta x} + \frac{vx_{i-1/2,M-1/2}^n}{4\Delta x} - \frac{vy_{i+1/2,M}^{n+1}}{2\Delta y} - \frac{vy_{i+1/2,M}^n}{2\Delta y} \\ & + \frac{vy_{i+1/2,M-3/2}^n}{4\Delta y} + \frac{vy_{i+1/2,M-1/2}^n}{4\Delta y} \end{aligned}$$

$$i = 0; j = 0$$

$$\begin{aligned} & \frac{p_{1/2,1/2}^{n+1}}{\beta\Delta t} + \frac{vx_{3/2,1/2}^{n+1}}{4\Delta x} + \frac{vx_{1/2,1/2}^{n+1}}{4\Delta x} + \frac{vy_{1/2,3/2}^{n+1}}{4\Delta y} + \frac{vy_{1/2,1/2}^{n+1}}{4\Delta y} = \frac{p_{1/2,1/2}^n}{\beta\Delta t} - \frac{vy_{1/2,1/2}^n}{4\Delta y} \\ & - \frac{vy_{1/2,3/2}^n}{4\Delta y} + \frac{vy_{1/2,0}^n}{2\Delta y} - \frac{vx_{1/2,1/2}^n}{4\Delta x} - \frac{vx_{3/2,1/2}^n}{4\Delta x} + \frac{vx_{0,1/2}^n}{2\Delta x} + \frac{vx_{0,1/2}^{n+1}}{2\Delta x} + \frac{vy_{1/2,0}^{n+1}}{2\Delta y} \end{aligned}$$

$$i = N - 1; j = M - 1$$

$$\begin{aligned} & \frac{p_{N-1/2,M-1/2}^{n+1}}{\beta\Delta t} - \frac{vx_{N-1/2,M-1/2}^{n+1}}{4\Delta x} - \frac{vx_{N-3/2,M-1/2}^{n+1}}{4\Delta x} - \frac{vy_{N-1/2,M-1/2}^{n+1}}{4\Delta y} - \frac{vy_{N-1/2,M-3/2}^{n+1}}{4\Delta y} = \\ & \frac{p_{N-1/2,M-1/2}^n}{\beta\Delta t} - \frac{vy_{N-1/2,M}^n}{2\Delta y} + \frac{vy_{N-1/2,M-1/2}^n}{4\Delta y} + \frac{vy_{N-1/2,M-3/2}^n}{4\Delta y} - \frac{vx_{N,M-1/2}^n}{2\Delta x} \\ & + \frac{vx_{N-3/2,M-1/2}^n}{4\Delta x} + \frac{vx_{N-1/2,M-1/2}^n}{4\Delta x} - \frac{vx_{N,M-1/2}^{n+1}}{2\Delta x} - \frac{vy_{N-1/2,M}^{n+1}}{2\Delta y} \end{aligned}$$

$$i = 0; j = M - 1:$$

$$\begin{aligned} & \frac{p_{1/2,M-1/2}^{n+1}}{\beta\Delta t} + \frac{vx_{3/2,M-1/2}^{n+1}}{4\Delta x} + \frac{vx_{1/2,M-1/2}^{n+1}}{4\Delta x} - \frac{vy_{1/2,M-3/2}^{n+1}}{4\Delta y} - \frac{vy_{1/2,M-1/2}^{n+1}}{4\Delta y} = \frac{p_{1/2,M-1/2}^n}{\beta\Delta t} \\ & - \frac{vx_{3/2,M-1/2}^n}{4\Delta x} - \frac{vx_{1/2,M-1/2}^n}{4\Delta x} + \frac{vx_{0,M-1/2}^n}{2\Delta x} - \frac{vy_{1/2,M}^n}{2\Delta y} + \frac{vy_{1/2,M-1/2}^n}{4\Delta y} + \frac{vy_{1/2,M-3/2}^n}{4\Delta y} \\ & + \frac{vx_{0,M-1/2}^{n+1}}{2\Delta x} - \frac{vy_{1/2,M}^{n+1}}{2\Delta y} \end{aligned}$$

$$i = N - 1; j = 0:$$

$$\begin{aligned} & \frac{p_{N-1/2,1/2}^{n+1}}{\beta\Delta t} - \frac{vx_{N-1/2,1/2}^{n+1}}{4\Delta x} - \frac{vx_{N-3/2,1/2}^{n+1}}{4\Delta x} + \frac{vy_{N-1/2,1/2}^{n+1}}{4\Delta y} + \frac{vy_{N-1/2,3/2}^{n+1}}{4\Delta y} = \frac{p_{N-1/2,1/2}^n}{\beta\Delta t} \\ & - \frac{vx_{N,1/2}^n}{2\Delta x} - \frac{vx_{N-1/2,1/2}^n}{4\Delta x} - \frac{vx_{N-3/2,1/2}^n}{4\Delta x} - \frac{vy_{N-1/2,3/2}^n}{4\Delta y} - \frac{vy_{N-1/2,1/2}^n}{4\Delta y} - \frac{vy_{N-1/2,0}^n}{2\Delta y} \\ & - \frac{vx_{N,1/2}^{n+1}}{2\Delta x} + \frac{vy_{N-1/2,0}^{n+1}}{2\Delta y} \end{aligned}$$

3.3. Implementación de las subrutinas de la Simulación Numérica

La mayoría de los programas de simulación ofrecen la posibilidad de acceder a su poder de procesamiento por medio de crear complejas y sofisticadas interfaces de usuario para la entrada de parámetros y análisis de resultados. De ahí, que tales desarrollos contengan tres principales elementos: (i) un pre-procesador, (ii) un *solver* y (iii) un post-procesador. El *solver* o procesador numérico corresponde a la mayoría del tiempo y costo computacional, y esta dado por el desarrollo de las ecuaciones anteriores así como por la resolución de los sistemas lineales.

3.3.1. Pre-procesador

Consiste en la entrada de datos que especifican características importantes del problema a simular. Entre tales aspectos están:

- Definición de la geometría en la región de interés: *dominio* computacional.
- Tipo y modo de subdivisión del dominio.
- Definición de las propiedades del fluido.
- Especificación de las condiciones de frontera en el dominio computacional, las cuales deben coincidir o estar muy cerca de las del dominio del problema.

Aún cuando las interfaces gráficas de usuario resultan muy atractivas por su facilidad de uso y amigabilidad, en este trabajo el pre-procesamiento tiene interfaz

con los parámetros del usuario a través de archivos de entrada. Esto se debe principalmente a la rápida implementación, y a la versatilidad en la realización de grandes cantidades de pruebas de módulos, las cuales constituyen unos de los pilares básicos en el desarrollo, verificación y depuración del software.

3.3.2. Post-procesador

Debido a las crecientes capacidades de las estaciones de trabajo en la generación de gráficos, la mayoría de los paquetes de simulación en CFD están equipados con herramientas versátiles de visualización. En este trabajo inicial, se incluyen funcionalidades tales como el despliegue en pantalla de la geometría del dominio y del mallado, dibujado de vectores y niveles de colores establecidos por una paleta. La escalabilidad de funcionalidades ha sido considerada para permitir modificar fácilmente las existentes, adaptándolas a otras necesidades o simplemente en la creación de nuevas características. Es atrayente además la posibilidad de visualizar dinámicamente los resultados por medio de generar la animación. La inclusión de datos alfanuméricos dentro del entorno gráfico, así como métodos para exportar cada cuadro animación en formato de *bitmap* (.bmp), hacen sentir al usuario a gusto con el entorno y los resultados que obtiene. Estas herramientas son consideradas como la mejor y más revolucionaria forma de comunicar las ideas a los no especialistas.

3.3.3. Estructuras de Datos

En las etapas de diseño del software y análisis de las bibliotecas a utilizar, fue considerada la creación de una estructura que proporcionaría una base sólida en el desarrollo de los módulos propuestos en los objetivos, así como en creaciones futuras. Esta estructura denominada `TDGrid` tiene como objetivo almacenar toda la información referente al dominio computacional sobre el cual es resuelto el problema numérico y se define como sigue:

```
struct TDGrid
{
    int dimension;
    int type;
    int nbnodes[3];
    double space[3];
    double *X;
    double *Y;
    double *Z;
    double *values;
};
```

Es importante resaltar que esta estructura considera el almacenamiento de todos los datos de un dominio computacional en un momento del tiempo. La mayor dimension soportada es 3D, considerando mallados uniformes, no uniformes (tensoriales), así como espacios y subdivisiones diferentes para cada dimensión. Se desarrollaron un conjunto de funciones y procedimientos que operan sobre dicha estructura con el fin de encapsular la manipulación de la información sin preocuparse por la manera en que se guardan o recuperan los datos. Tales prototipos se describen en la siguiente sección.

3.3.4. Prototipos de las Funciones y Procedimientos principales

A continuación se presentarán los prototipos de las principales funciones implementadas para UCSparcelib:

```
void InitGrid1D( TDGrid GG, double value)
```

Inicializa todos los valores del mallado GG con value

```
void CreateGrid1DUniform( TDGrid *GG, int nbnodes, float space)
```

```
void CreateGrid2DUniform(TDGrid *GG, int nbnodesX, float spaceX,  
int nbnodesY, float spaceY)
```

Reserva la memoria necesaria y establece los valores del mallado GG

```
void SetBoundary1D1(TDGrid GG, double fParam)
```

```
void SetBoundary1D2(TDGrid GG, double fParam1, double fParam2)
```

```
void SetBoundary2D1(TDGrid GG, double fParam0, double fParam1,  
double fParam2, double fParam3, double fParam4, double fParam5,  
double fParam6, double fParam7)
```

Procedimientos para establecer los valores de las fronteras del mallado. Los valores son obtenidos desde el cuerpo principal.

```
void CopyGrid( TDGrid SS, TDGrid DD )
```

Copia la informacion contenida en un mallado a otro mallado

```
int getType( const TDGrid GG )
```

```
int getDimension( const TDGrid GG )
```

Permiten obtener valores de las características del mallado almacenado en la estructura

```
void getMinMaxValues( const TDGrid GG, double *min, double *max )
```

```
double getMinValue( const TDGrid GG )
```

```
double getMaxValue( const TDGrid GG )
```

Funciones para buscar y retornar los valores máximo y mínimo, o ambos, en los nodos de un mallado

```
void GridWrite( const TDGrid GG, FILE *file )
```

Escribe en una salida especificada por el usuario, la información contenida en el mallado, siguiendo un formato que permite la fácil legibilidad.

```
void WriteGridValues( const TDGrid GG, FILE *file )
```

Escribe en una salida especificada por el usuario los valores numéricos correspondientes a cada nodo del mallado. Esta función colocada dentro del ciclo del tiempo en la simulación, permite registrar el historial de cambios provocados por las ecuaciones en los valores iniciales del mallado.

```
void DeleteGrid( TDGrid GG )
```

Libera de la memoria toda la información contenida por el mallado **GG**

```
double Error1DUniform( TDGrid Orig, TDGrid Ref, int kk )
```

Calcula el error numérico existente en los valores de cada mallado, a través de la norma de energía y considerando que el mallado **Ref** es **kk** veces más refinado que el mallado **GG**

3.4. Otros desarrollos

Durante la creación y verificación del software desarrollado en este trabajo, fueron creados algunos programas adicionales con el propósito de generar y/o validar casos de prueba y el correcto funcionamiento de las subrutinas individuales o de la integración entre ellas. Para mayor información de estos casos, se puede consultar la documentación y reportes internos adicionales que no forman parte de este informe. Entre estos desarrollos se encuentran aplicaciones para el cálculo del error numérico en caso de tener alguna solución analítica, generación y lectura de archivos para la verificación de secuencias de ejecución, llenado de estructuras, cantidad de iteraciones, entre otros.

3.5. Visualización

El objetivo de la creación de módulos de post-procesamiento para la visualización de los cálculos numéricos, es brindar una herramienta al usuario para comprender los resultados de una forma natural. Es por ello que se ha conseguido desarrollar con bibliotecas estándares, versiones de dicho post-procesador para diferentes plataformas computacionales. Esto se logra realizando una separación clara de estos componentes con el resto de los módulos creados. La interfaz de entrada es nuevamente a través de archivos de datos, los cuales son generados automáticamente por las capas anteriores. Este tipo de entrada ofrece al usuario la posibilidad de procesar los cálculos de la simulación en computadores remotos de mayores prestaciones, obtener los resultados en archivos de texto ASCII y visualizar de manera local los resultados, independientemente del sistema

operativo que posea.

3.5.1. Bibliotecas Involucradas

La principal biblioteca en el desarrollo del post-procesador es OpenGL, pero se ha hecho uso de bibliotecas auxiliares como GLUT, y GLUI. Como ya se ha mencionado, GLUT es una interfaz de software para la creación y manipulación de ventanas independientemente del sistema operativo. Creada en ANSI C, provee además funcionalidades de entrada de usuario que no se hallan en la especificación de OpenGL. Por otra parte, GLUI (*GLUT-Base User Interface*) [22] provee controles tales como botones, combos, spinners y etiquetas entre otros para aplicaciones de OpenGL. Igualmente es independiente del sistema de ventanas, acomplándose a GLUT para recibir eventos externos, tales como movimientos del *mouse*.

Capítulo 4

RESULTADOS Y CONCLUSIONES

4.1. Resultados Numéricos

Las pruebas para las simulaciones fueron evaluadas en base al realismo que debían presentar los resultados de acuerdo a las condiciones proporcionadas y el comportamiento computacional que presenta la simulación. Con respecto al tiempo computacional para el cálculo, depende no solo del tipo de computador, sino de las condiciones, complejidad y magnitud de la situación deseada. Debido a que los resultados son procesados independientemente a la visualización, no existe una relación entre el tiempo de cómputo y la fluidez en que son representados los cuadros de animación. Es importante mencionar que las pruebas representativas mostradas en este trabajo, constituyen apenas unos pocos casos planteados de entre una inmensa cantidad y variedad que pueden ser consultados en la bibliografía o en reportes de experimentos en laboratorios. La intención de ello es

verificar el comportamiento de las ecuaciones en función de los métodos numéricos utilizados, y no hacer una demostración de las verdaderas capacidades de las herramientas que han sido creadas; corresponde a los investigadores en las áreas particulares de la ciencia o de ingeniería, establecer condiciones y escenarios de utilidad práctica.

El contexto de hardware en todas las pruebas fue un procesador Intel Pentium II 400 Mhz, 128Mb RAM. El compilador utilizado fue *gcc* bajo un sistema operativo *Linux Mandrake 8.1*.

4.1.1. Pruebas en 1D

La estructura de la matriz obtenida haciendo uso de métodos implícitos, es como se muestra en la figura 4.1

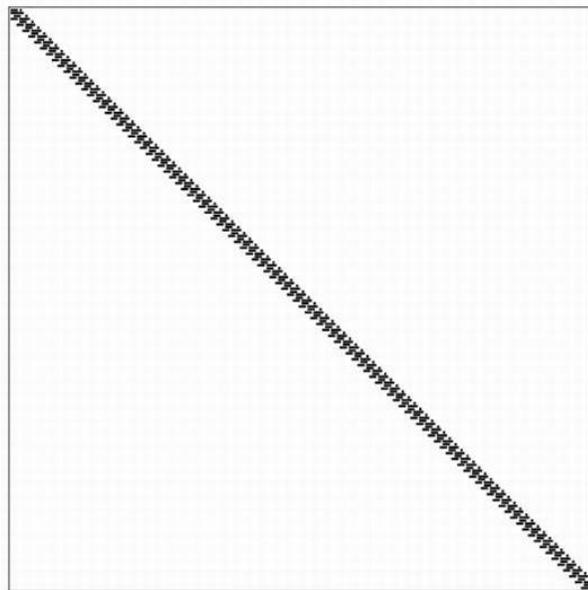


Figura 4.1: Estructura de la matriz para 1D

Prueba 1

Se considera en 1D un flujo laminar, incompresible con viscosidad cinemática constante entre dos placas paralelas separadas por una distancia H . Inicialmente las dos placas están estacionarias y el fluido entre ellas está estancado. De repente en el instante $t = 0$, una de las placas se mueve en la dirección x positiva a una velocidad constante V_0 . Bajo estas consideraciones, las ecuaciones de Navier-Stokes son lineales y las condiciones de fronteras son homogéneas, lo que da de inmediato la solución analítica exacta

$$u/V_0 = 1 - y/H - 2 \sum_{n=1}^{\infty} \frac{1}{n\pi} \text{sen}(n\pi y/H) \exp(-n^2 \pi^2 \nu t/H^2)$$

Esta solución es utilizada para evaluar el comportamiento de la simulación con respecto a los resultados obtenidos a través de las aproximaciones. Es importante resaltar que estos valores exactos pertenecen solo a una incógnita de velocidad. El error calculado a través de la norma de energía dada por (2.14) fue menor a 10^{-2} en todas las pruebas.

Debido a que se tiene la solución exacta para esta situación sencilla, los siguientes gráficos muestran las aproximaciones obtenidas por los métodos, una vez alcanzado el final de la simulación:

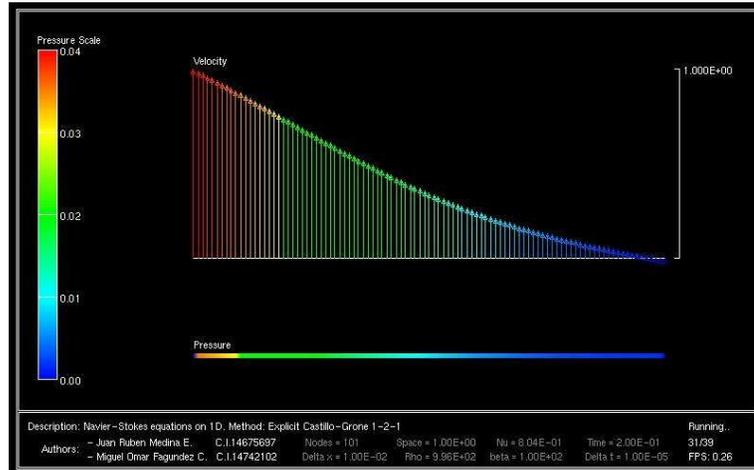


Figura 4.2: Prueba 1(a), Operadores de Soporte 1-2-1 Explícito

Prueba 2

Se considera una velocidad inicial constante para todos los nodos del mallado en el tiempo $t = 0$. El proceso de simulación genera una parábola en el perfil de velocidades.

Prueba 3

Al igual que en la prueba 1, se establece velocidades constantes a cada una de las placas

4.1.2. Pruebas en 2D

La estructura de la matriz obtenida haciendo uso de métodos implícitos en 2D, es como se muestra en la figura 4.1

Prueba 1

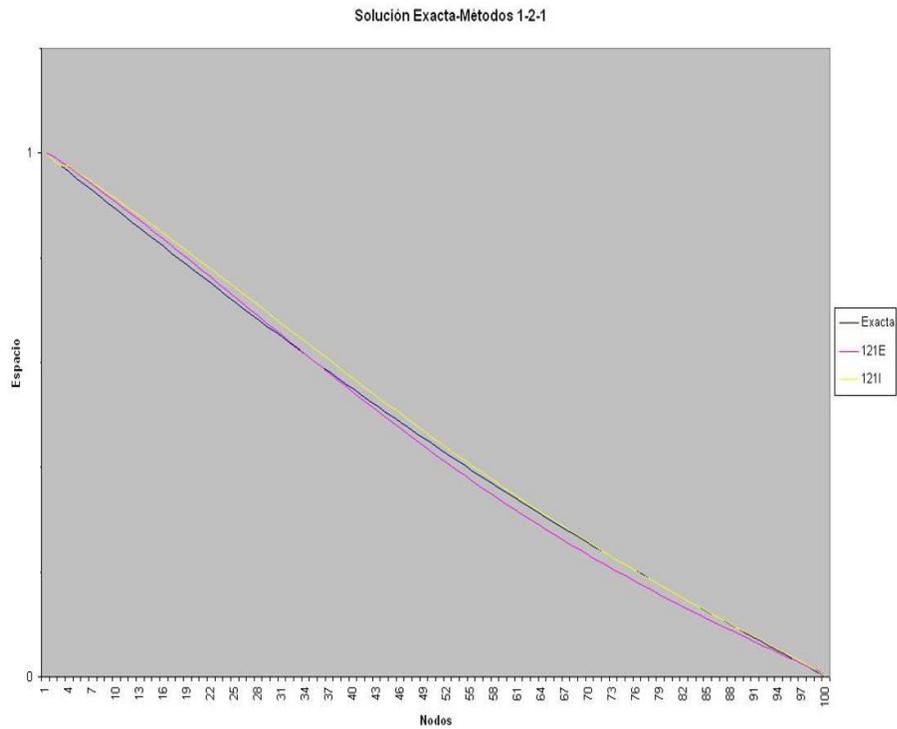


Figura 4.3: Comparación de los resultados del método 1-2-1 con la solución exacta

Al igual que en la prueba inicial, se colocaron casos borde que simulan el movimiento a una velocidad constante de una de las placas que contienen el fluido

Prueba 2

Se establece una velocidad constante e igual para todos los nodos en la frontera derecha. Puede observarse el perfil parabolico que se desarrolla ya avanzada la simulación.

Prueba 3

Por medio de variar la condicion borde en funcion del tiempo, se establece un

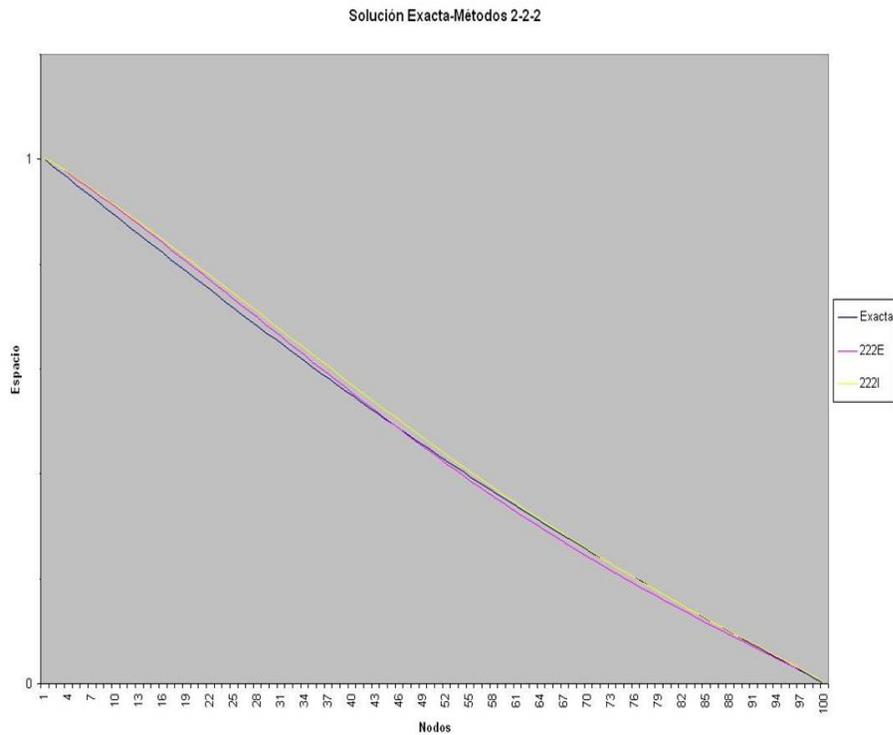


Figura 4.4: Comparación de los resultados del método 2-2-2 con la solución exacta comportamiento senoidal con respecto al eje y y constante con respecto al eje x .

Prueba 4

Igual que en la prueba anterior, pero colocando velocidades constantes y en sentido contrario en los bordes que representen las placas paralelas. Puede observarse los vórtices que se forman a la salida del fluido debido a la acción contraria que ejercen las placas.

Ya que el caso 2D es una particularidad del caso 3D, es posible crear vistas en 3D haciendo uso de los calculos anteriores. Por ejemplo, un corte longitudinal en la tubería que contiene al fluido puede ser representado como en la figura

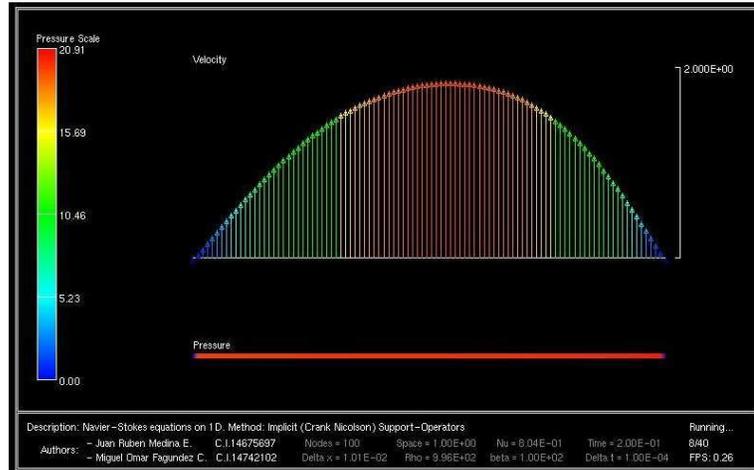


Figura 4.5: Prueba 2(b), Operadores de Soporte 1-2-1 Implícito

En las tablas 4.1 y 4.2 son presentados algunos de los datos obtenidos a partir de los resultados y el cálculo de las pruebas. Esta tabla describe las dimensiones y los tiempos de procesamientos haciendo uso de dos métodos iterativos disponibles en UCSparseLib, resaltando la cantidad de iteraciones requeridas en cada uno con preconditionador y sin preconditionador. Todas las pruebas se realizaron para 100 iteraciones de tiempo de simulación.

NodesX	NodesY	Metodo	c/Pre-c.	s/Precond	Dim	No nulos	dens
15	10	BiCGstab	2	4	450x450	2750	1.36
15	10	GMRES	3	6	450x450	2750	1.36
60	20	BiCGstab	3	-	3600x3600	22480	0.173
60	20	GMRES	6	-	3600x3600	22480	0.173
40	40	BiCGstab	2	6	4563x4563	28587	0.137
40	40	GMRES	4	10	4563x4563	28587	0.137
70	50	BiCBstab	3	-	10143x10143	63767	0.0620
70	50	GMRES	6	-	10143x10143	63767	0.0620
80	80	BiCGstab	3	-	18723x18723	117947	0.0336
80	80	GMRES	6	-	18723x18723	117947	0.0336

Cuadro 4.1: Datos obtenidos de las pruebas

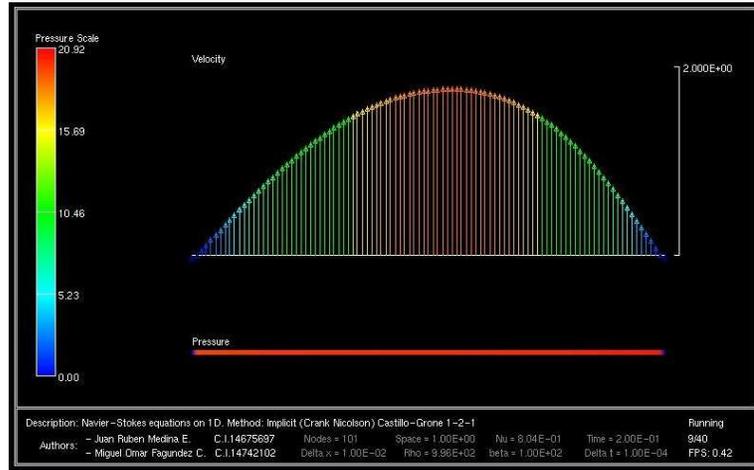


Figura 4.6: Prueba 2(a), Castillo-Grone 2-2-2 Implícito

NodesX	NodesY	Metodo	T.User121	T.Sist121	T.User222	T.Sist222
15	10	BiCGstab	1,250s	0,110s	1,590s	0,100s
15	10	GMRES	1,300s	0,120s	1,740s	0,80s
60	20	BiCGstab	18,520s	1,40s	24,520s	1,10s
60	20	GMRES	17,780s	1,210s	23,560s	1,570s
40	40	BiCGstab	21,800s	1,480s	28,440s	1,540s
40	40	GMRES	20,840s	1,540s	29,410s	1,730s
70	50	BiCBstab	70,310s	3,170s	81,820s	3,950s
70	50	GMRES	68,580s	4,90s	81,120s	4,340s
80	80	BiCGstab	No resuelto	No resuelto	No resuelto	No resuelto
80	80	GMRES	129,670s	9,810s	171,780s	10,740s

Cuadro 4.2: Datos obtenidos de las pruebas

4.2. Conclusiones y Recomendaciones

- Los métodos implementados convergen en todos los casos de prueba.
- Los resultados obtenidos en las simulaciones, en función de las condiciones borde de prueba, presentan una respuesta semejante a la que podría ocurrir en la realidad.
- En todos los casos la tolerancia del error en el cálculo de la solución de

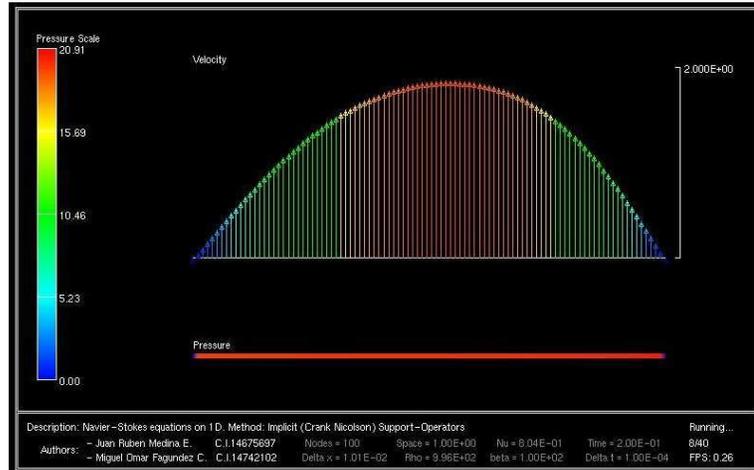


Figura 4.7: Prueba 3(a), Castillo-Grone 2-2-2 Implícito

los sistemas lineales por los métodos BicGSTAB y GMRES fue de 10^{-12} , exactitud generalmente alcanzada en las primeras iteraciones y por lo que no se considera que sea necesario cambiar el método a menos que se necesite mayor precisión.

- En el cálculo de las aproximaciones a las ecuaciones diferenciales a través de los métodos miméticos, el error calculado por la norma de energía aplicada con respecto a una solución analítica, se encuentra por el orden de 10^{-2} , siendo mejor en las aproximaciones formuladas por los métodos 2-2-2.
- El software desarrollado tiene como propósito ser una herramienta para los investigadores en diferentes ramas de la ciencia, donde aplique y sea importante la consideración de los aspectos de la dinámica de fluidos. A pesar de la alta complejidad de las ecuaciones y de las discretizaciones obtenidas en el desarrollo de esta investigación, se superaron las expectativas en los resultados numéricos y por ende visuales.

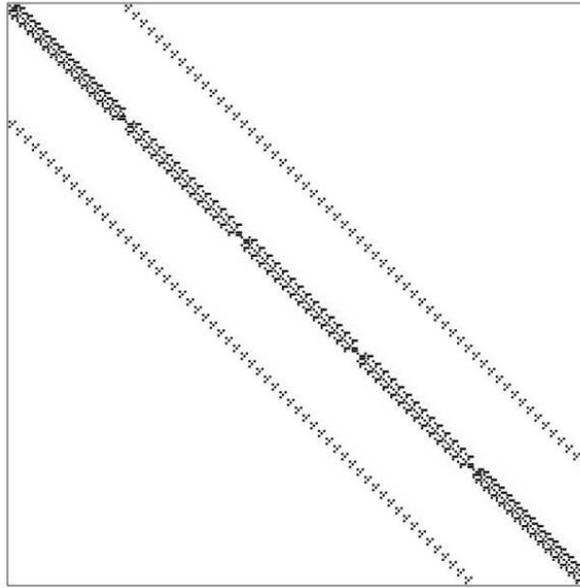


Figura 4.8: Estructura de la matriz para 2D

4.3. Trabajos Futuros

- Concluir el desarrollo de una interfaz gráfica multiplataforma de usuario, que hace amigable la interacción con los parámetros requeridos y otras entradas.
- Reformular las subrutinas implementadas para mejorar los tiempos de cómputo.
- Estudiar el comportamiento de los métodos sobre mallados no uniformes y aplicado a las ecuaciones de la dinámica de fluidos.
- Desarrollar los cálculos y la visualización para resolver problemas en 3D.
- Paralelización de las subrutinas de simulación.
- Profundizar en el estudio y modelos matemáticos para escenarios con diferentes flujos interactuando entre sí, con estructuras elásticas y/o medios

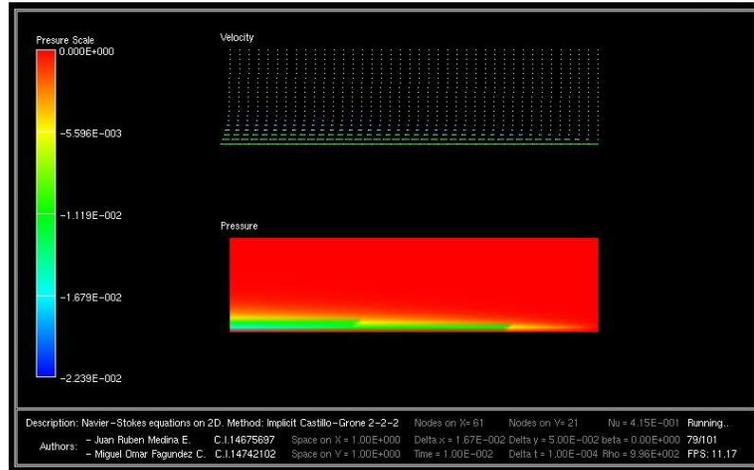


Figura 4.9: Prueba 1, Castillo-Grone 2-2-2

porosos,

- Desarrollar aproximaciones miméticas para fluidos compresibles.

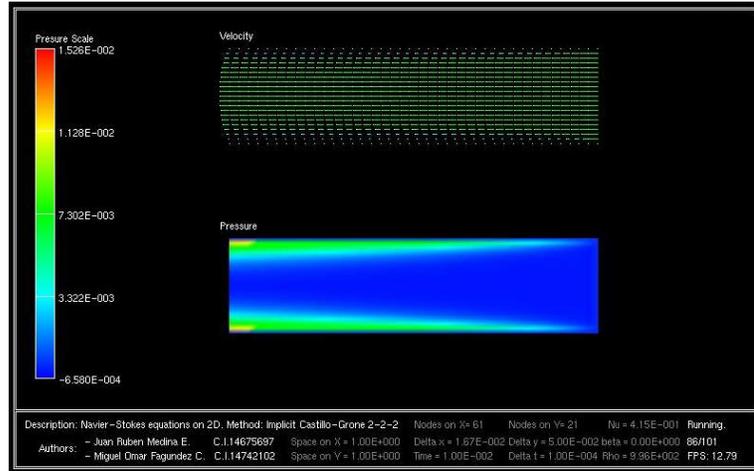


Figura 4.10: Prueba 2, Castillo-Grone 2-2-2 Implicito

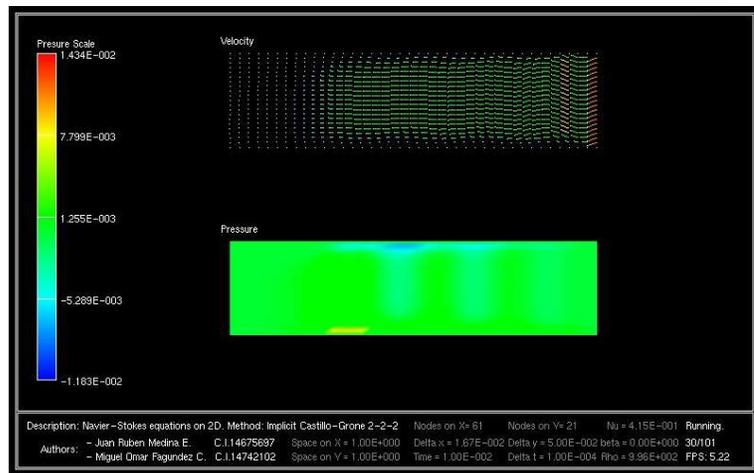


Figura 4.11: Prueba 3, Castillo-Grone 2-2-2 Implicito

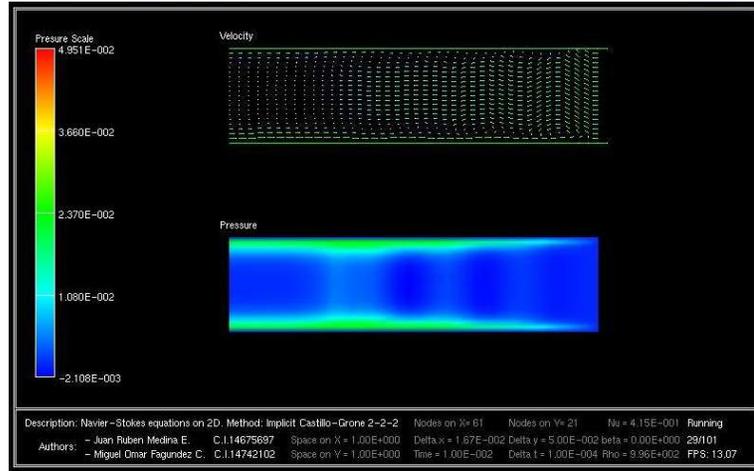


Figura 4.12: Prueba 4, Castillo-Grone 2-2-2 Implicito

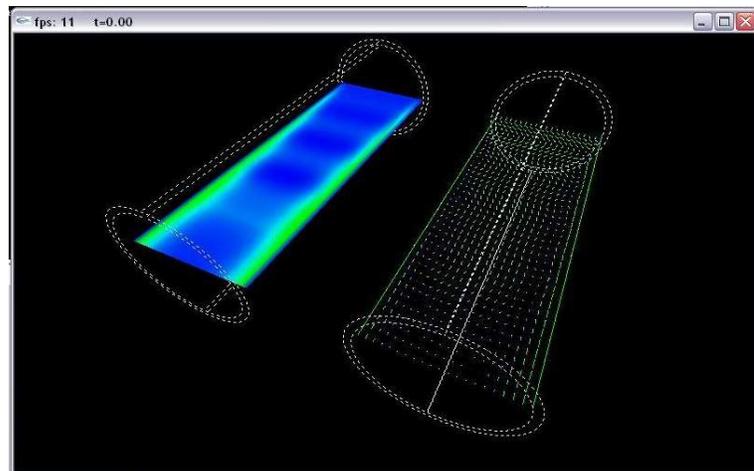


Figura 4.13: Vista 3-D

Apéndice A

Código Fuente de la Implementación

El código desarrollado siguiendo los lineamientos y estándares definidos en UCSparseLib, representa un total aproximado de 10000 líneas. Por esta razón, solo es incluido uno de los cuerpos principales de las simulaciones en dos dimensiones. El resto del código fuente y los programas junto con las bibliotecas que se incluyen, son entregadas en formato digital.

```
#ifndef SOURCE
# include <stdio.h>
# include <stdlib.h>
# include <math.h>
# include "constantes.h"
# include "Ifunc.h"
# include "Dfunc.h"
```

```

# include "TDMatrix.h"
# include "TDMatrix_IO.h"
# include "TDMatrix_OPE.h"
# include "TDMatrix_ITERATIVE.h"
# include "Gen121-2D.h"
# include "Macros.h"
# include "TDGrids.h"

int      iparIM[5];
double   dparIM[3];
double   SpaceX, SpaceY, Rho, Nu, beta, At, TTime, TOL;
int mm,nn;
int SaveEach = 1;

double mem[15] = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
double opf[15] = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
double opm[15] = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};

void ReadParam()
{
    FILE *fp;

    fp = fopen("SolverParam.dat", "r");

```

```

if (fp == NULL) ERROR( "Opening SolverParam.dat" );
fscanf( fp, "%d%*[\n]*c", &iparIM[0] );
fscanf( fp, "%d%*[\n]*c", &iparIM[1] );
fscanf( fp, "%d%*[\n]*c", &iparIM[2] );
fscanf( fp, "%d%*[\n]*c", &iparIM[3] );
fscanf( fp, "%d%*[\n]*c", &iparIM[4] );
fscanf( fp, "%lf%*[\n]*c", &dparIM[0] );
fscanf( fp, "%lf%*[\n]*c", &dparIM[1] );
fscanf( fp, "%lf%*[\n]*c", &dparIM[2] );
fclose( fp );

fp = fopen("Fluid2DParam.dat", "r");
if (fp == NULL) ERROR( "Opening FluidParam.dat" );
fscanf( fp, "%d%*[\n]*c", &mm );
fscanf( fp, "%d%*[\n]*c", &nn );
fscanf( fp, "%lf%*[\n]*c", &SpaceX );
fscanf( fp, "%lf%*[\n]*c", &SpaceY );
fscanf( fp, "%lf%*[\n]*c", &TTime );
fscanf( fp, "%lf%*[\n]*c", &At );
fscanf( fp, "%lf%*[\n]*c", &Rho );
fscanf( fp, "%lf%*[\n]*c", &Nu );
fscanf( fp, "%lf%*[\n]*c", &TOL );
fclose( fp );
}

```

```
//-----
```

```

int main(int argc, char** argv)
{
/* ----- Local Variables */
FILE *outVelocityX, *outVelocityY, *outPressure, *outValues, *fp;
FILE *fMyMatriz;
int      ii, jj, nbrows, totiter, ierr, rr, nbrowsRef;
int      iterRef,save=0, iter=0, *iwa;
double   *SOLRef, *RHS, *RHSRef, *SOL, *plot, *plotRef;
double   ff, minVelx, minVely, maxVelx, maxVely;
double   minPres, maxPres, auxmin, auxmax, ErrorV, ErrorP;
TDMatrix AA,AARef;
TDSparseVec row, rowRef;
TDprecond PM, PMRef;
TDGrid VxAct, VyAct, PPAct, VxActRef, VyActRef;
TDGrid PPActRef, VxNext, VyNext, PPNext, VxNextRef;
TDGrid VyNextRef, PPNextRef, VxAux, VyAux, PPAux;

/* ----- */ BEGIN( main );

outVelocityX = fopen("velocitiesX-121I2D.dat", "w");
if (outVelocityX == NULL) ERROR( "Abriendo fichero" );
outVelocityY = fopen("velocitiesY-121I2D.dat", "w");
if (outVelocityY == NULL) ERROR( "Abriendo fichero" );
outPressure = fopen("pressures-121I2D.dat", "w");
if (outPressure == NULL) ERROR( "Abriendo fichero" );

```

```

outValues = fopen("values-121I2D.dat", "w");
if (outValues == NULL) ERROR( "Abriendo fichero" );

ReadParam();

nbrows = 3*(mm)*(nn);

CreateGrid2DUniform(&VxAct, mm+1, SpaceX, nn+1, SpaceY);
CreateGrid2DUniform(&VyAct, mm+1, SpaceX, nn+1, SpaceY);
CreateGrid2DUniform(&PPAct, mm+1, SpaceX, nn+1, SpaceY);

CreateGrid2DUniform(&VxNext, mm+1, SpaceX, nn+1, SpaceY);
CreateGrid2DUniform(&VyNext, mm+1, SpaceX, nn+1, SpaceY);
CreateGrid2DUniform(&PPNext, mm+1, SpaceX, nn+1, SpaceY);

beta = Nu*At/(VxAct->X[0]*VxAct->X[0]);

InitGrid2D_1( VxAct, 0.0 );
InitGrid2D_1( VyAct, 0.0 );
InitGrid2D_1( PPAct, 0.0 );

SetBoundary2D_1(VxAct, -3.0, -3.0, -3.0, -1.0, -1.0, 0.0, 0.0, 0.0);
SetBoundary2D_1(VyAct, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0);
SetBoundary2D_1(PPAct, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0);

getMinMaxValues( VxAct, &minVelx, &maxVelx );

```

```

getMinMaxValues( VyAct, &minVely, &maxVely );
getMinMaxValues( PPAct, &minPres, &maxPres );

WriteGridValues(VxAct, outVelocityX);
WriteGridValues(VyAct, outVelocityY);
WriteGridValues(PPAct, outPressure);

GenMatrix121_2D( &AA, VxAct, VyAct, PPAct, Nu, Rho, At, beta, 0 );

fMyMatriz = fopen("MyMatriz.txt","w");
TDMatrixWrite(AA,fMyMatriz);
fclose(fMyMatriz);

RHS = ALLOC( nbrows,double,"RHS" );
SOL = ALLOC( nbrows,double,"SOL" );
dSet( nbrows, SOL, 0.0 );

for (ff= At; ff< TTime;ff+= At) //----- TIME LOOP
{
    save++;

    SetBoundary2D_1(VxNext, 2.0, 2.0, 2.0, -1.0, 2.0, 2.0, 2.0, 0.0);
    SetBoundary2D_1(VyNext, 0.0, 0.0, sin(save*PI/4),
                    sin(save*PI/4), sin(save*PI/4), 0.0, 0.0, 0.0);
    SetBoundary2D_1(PPNext, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0);

```

```

GenMatrix121_2D( &AA, VxAct, VyAct, PPAct, Nu, Rho, At, beta, 1 );
GenRHS121_2D(RHS, VxAct, VyAct, PPAct, VxNext, VyNext, PPNext,
             At, beta, Nu, Rho );

if (iparIM[4]) // traza
    plot = ALLOC( iparIM[1]+1, double, "plot" );
else
    plot = (double *)NULL;

computePrecond( iparIM, dparIM, AA, &PM );
BiCGstab( AA, SOL, RHS, &PM, iparIM, dparIM, plot,
          &totiter, &ierr );

Fill3GridsWithSolution(VxNext, VyNext, PPNext, SOL);

rr = 2*mm-1;

CopyGrid( PPNext, PPAct );
CopyGrid( VxNext, VxAct );
CopyGrid( VyNext, VyAct );

getMinMaxValues( VxAct, &auxmin, &auxmax );
if (auxmin<minVelx)
    minVelx=auxmin;
if (auxmax>maxVelx)
    maxVelx=auxmax;

```

```

getMinMaxValues( VyAct, &auxmin, &auxmax );
if (auxmin<minVely)
    minVely=auxmin;
if (auxmax>maxVely)
    maxVely=auxmax;

getMinMaxValues( PPAct, &auxmin, &auxmax );
if (auxmin<minPres)
    minPres=auxmin;
if (auxmax>maxPres)
    maxPres=auxmax;

if (save%SaveEach==0)
{
    iter++;
    WriteGridValues(VxAct, outVelocityX);
    WriteGridValues(VyAct, outVelocityY);
    WriteGridValues(PPAct, outPressure);

    printf("Time %f processed\n",ff);
}
} //----- END TIME LOOP

FREE( SOL );
FREE( RHS );

```

```

if (iparIM[4])
    FREE( plot );

TDMatrixDelete( AA );

fprintf(outValues,"%d      | Pasos de Tiempo\n",iter+1);
fprintf(outValues,"%d      | Cantidad de nodos en X\n", mm+1);
fprintf(outValues,"%d      | Cantidad de nodos en Y\n", nn+1);
fprintf(outValues,"%d      | tipo de mallado \n", getType(VxAct));
fprintf(outValues,"%0.51E   | Espacio en X\n", SpaceX );
fprintf(outValues,"%0.51E   | Espacio en Y\n", SpaceY );
fprintf(outValues,"%0.51E   | Tiempo de simulacion\n", TTime );
fprintf(outValues,"%0.51E   | Delta t\n", At );
fprintf(outValues,"%0.51E   | Velocidad minima en X\n",minVelx);
fprintf(outValues,"%0.51E   | Velocidad maxima en X\n",maxVelx);
fprintf(outValues,"%0.51E   | Velocidad minima en Y\n",minVely);
fprintf(outValues,"%0.51E   | Velocidad maxima en Y\n",maxVely);
fprintf(outValues,"%0.51E   | Presion minima  \n",minPres);
fprintf(outValues,"%0.51E   | Presion maxima  \n",maxPres);
fprintf(outValues,"%0.51E   | Rho\n", Rho );
fprintf(outValues,"%0.51E   | Nu\n", Nu );
fprintf(outValues,"%0.51E   | Tolerancia del error\n", TOL );

fprintf(outValues,"Navier-Stokes equations on 2D.
                    Method: Implicit Support-Operators 1-2-1\n");

```

```
fclose(outVelocityX);
fclose(outVelocityY);
fclose(outPressure);
fclose(outValues);

DeleteGrid(PPAct);
DeleteGrid(VxAct);
DeleteGrid(VyAct);
DeleteGrid(PPNext);
DeleteGrid(VxNext);
DeleteGrid(VyNext);
/* ----- */ END( main );
return 0;
}

#endif /* SOURCE */
```

Bibliografía

- [1] M. Shashkov (1996). *Conservative Finite-Difference Methods on General Grids*, pp 47-64
- [2] J.M. Hyman and M. Shashkov (1998). *Approximation of Boundary Conditions for Mimetic Finite-Difference Methods*. *Computer Math. Applic.*, **36**, 5, pp 79-99.
- [3] J. Castillo and M. Yasuda, (2002). *Comparison of two Matrix Operators Formulation for Mimetic Divergence and Gradient Discretizations*
- [4] J. Castillo and D. Grone, (2003). *A Matrix Analysis Approach to Higher Order Approximations for Divergence and Gradients Satisfying a Global Conservation Law*. *SIAM J. Matrix Anal. Appl.* **25**, 1, pp 128-142
- [5] B. Jiang (1998). *The Least-Square Finite Element Method. Theory and Applications in Computational Fluid Dynamics and Electromagnetics*, pp 128-152.
- [6] J. Stam (1999). *Stable Fluids, SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, pp 121-128.

- [7] J. Stam (2001). *A Simple Fluid Solver based on the FFT*, *Journal of Graphics Tools*, **6**, 2, pp 43-52.
- [8] P. Valera (2002). *Simulación de Tejido Blando basada en el Método de los Elementos Finitos*, Facultad de Ciencia y Tecnología, Departamento de Computación, Universidad de Carabobo.
- [9] G. Larrazabal (2002). *UCSparseLib: Una Librería Numérica para la resolución de sistemas Lineales Dispersas*, Facultad de Ciencia y Tecnología, Departamento de Computación, Universidad de Carabobo
- [10] G. Larrazabal, C. Torres and J. Castillo (2003). *An eficiente and robust algorithms for 2D stratified fluid flow calculations*, To appear in special issue of Journal Applied Numerical Mathematics.
- [11] L.M. de la Cruz Salas, (1999). *Generación de Mallas usando Mapeo Ortogonal y algunas Aplicaciones*, Universidad Nacional Autónoma de México. Página Web: <http://www.mcc.unam.mx/lmd/Proyectos/gema/tesis/tesis.html>.
- [12] R. Schneiders, *Mesh Generation and Grid Generation on the Web*. Página Web: <http://www-users.informatik.rwth-aachen.de/roberts/meshgeneration.html>.
- [13] S. Steinberg, and P.J. Roache, (1986). *Variational Grid Generation*, Num. Meth. for P.D.E.'s., **2**, pp 71-96.
- [14] Kent Beck, (2000). *Extreme Programming Explained: Embrace Change*. Addison Wesley.
- [15] Kent Beck, (1999). *Embracing Change with Extreme Programming*. *Computer, IEEE Computer Society*, **32**, 10, pp 70-77.

- [16] P. BOCHEV, Z. CAI, T. A. MANTEUFFEL y S. F. MCCORMICK, (1998) *Analysis of velocity-flux first-order system least-squares principles for the Navier-Stokes equations: Part I*, SIAM J. Numer. Anal., v. 35, n. 3, pp. 990-1009.
- [17] J. FERZINGER, M. PERIC, (2002) *Computational Methods for Fluid Dynamics*. Springer-Verlag Berlin. 3ra edición.
- [18] H. K. VERSTEEG, W. MALALASEKERA, (1995) *An Introduction to Computational Fluid Dynamics the Finite Volume Methods*. Prentice Hall.
- [19] M. POTTER, D. WIGGERT, (2002) *Mecánica de Fluidos*. Thompson. 3ra edición.
- [20] NEIDER J., Davis T., WOO M., (1993) *OpenGL Programming Guide: The official guide to learning OpenGL*. Release 1. AddisonWesley.
- [21] KILGARD M. (1993) *The OpenGL Utility Toolkit (GLUT) Programming Interface*. API Version 3. Silicon Graphics, Inc.
- [22] RADEMACHER P. (1999) *GLUI. A GLUT-Based User Interface Library* Version 2.0.