

**DETERMINACIÓN DEL CAMPO DE FLUJO Y DE
CONCENTRACIONES DE LA INTERFASE DE DOS FLUIDOS
INCOMPRESIBLES QUE SE DESPLAZAN A TRAVÉS DE UNA
TUBERÍA USANDO EL MÉTODO DE VOLÚMENES FINITOS**

CARLOS ALBERTO FLÓREZ ARIAS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FÍSICO MECÁNICAS
MAESTRÍA EN INGENIERÍA MECÁNICA
BUCARAMANGA**

2015

**DETERMINACIÓN DEL CAMPO DE FLUJO Y DE
CONCENTRACIONES DE LA INTERFASE DE DOS FLUIDOS
INCOMPRESIBLES QUE SE DESPLAZAN A TRAVÉS DE UNA
TUBERÍA USANDO EL MÉTODO DE VOLÚMENES FINITOS**

CARLOS ALBERTO FLÓREZ ARIAS

**Trabajo de Grado para optar al título de
Magister en Ingeniería Mecánica**

Director
Dr. DAVID ALFREDO FUENTES DÍAZ
Doctor en tecnología energética

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FÍSICO MECÁNICAS
MAESTRÍA EN INGENIERÍA MECÁNICA
BUCARAMANGA**

2015

AGRADECIMIENTOS

Al Doctor David Alfredo Fuentes Díaz, por su apoyo y ayuda oportuna, pues sin su guía no habría sido posible la realización de este proyecto.

A la maestría en Ingeniería Mecánica por la formación investigativa y crítica recibida, su colaboración y asistencia fueron cruciales en el desarrollo de este proyecto de investigación.

A mis padres y hermanos, por su comprensión y ayuda incondicional y desinteresada, a mi esposa Gina, porque gracias a su soporte y consejo fue posible la culminación de este trabajo. Gracias mi amor.

Un agradecimiento muy especial a toda la comunidad de software libre, particularmente a la comunidad de los proyectos Ubuntu, OpenFOAM, Python, SALOME y \LaTeX , también a cfd-online.com por compartir el conocimiento libremente.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1 PLANTEAMIENTO DEL PROBLEMA	18
2 JUSTIFICACIÓN	20
3 OBJETIVOS DEL TRABAJO DE INVESTIGACIÓN	22
3.1 OBJETIVO GENERAL	22
3.2 OBJETIVOS ESPECÍFICOS	22
4 MARCO TEÓRICO	23
4.1 CONCEPTOS FUNDAMENTALES	23
4.1.1 Tendencia de la producción petrolera.	23
4.1.2 Características del método de transporte	24
4.1.2.1 Transporte secuencial o “batch”	24
4.1.3 Fenómenos de transferencia de masa en transporte secuencial de fluidos. .	24
4.1.3.1 Difusión molecular	25
4.1.3.2 Difusión laminar	26
4.1.3.3 Difusión turbulenta	27
4.1.3.4 Estratificación	27
4.2 ESTADO DEL ARTE	28
4.3 MÉTODOS DE SOLUCIÓN	31
4.3.1 Método de los volúmenes finitos (FVM)	32
5 MODELAMIENTO DE LA MEZCLA DE DOS FLUIDOS EN TRANSPORTE BATCH	33

CONTENIDO

5.1 ECUACIONES GOBERNANTES	33
5.1.1 Introducción	33
5.1.2 Ecuación de conservación de la masa	33
5.1.3 Ecuación de conservación de la cantidad de movimiento	38
5.1.4 Ecuación de conservación de la energía	42
5.1.5 Ecuaciones del modelo turbulento	47
5.1.5.1 Modelo RANS $k - \epsilon$	48
5.1.6 Ecuación de energía cinética turbulenta k	52
5.1.7 Ecuación de la Razón de disipación ϵ	53
5.1.8 Funciones de pared	53
5.2 ECUACIONES DEL MODELO	53
5.3 DISCRETIZACIÓN	55
5.3.1 Dominios complejos	55
5.3.2 Malla estructurada	56
5.3.3 Discretización del dominio de la solución	56
5.3.4 Discretización de las ecuaciones de transporte	58
5.3.4.1 Discretización de términos espaciales	60
5.3.4.2 Discretización temporal	63
5.3.5 Condiciones de frontera	66
5.3.5.1 Condiciones de frontera numéricas	67
5.3.5.2 Condiciones de frontera física	68
5.3.6 Flujos turbulentos	69
5.3.7 Procedimiento de discretización	69
5.3.7.1 Ecuación de presión	70
5.3.7.2 Acoplamiento de presión y velocidad	71
5.3.7.3 Algoritmo PISO	72
5.3.8 Solución del sistema	73
6 METODOLOGÍA	74
6.1 INSTRUMENTOS	74
6.1.1 OpenFOAM®	75
6.1.1.1 Generalidades	75
6.1.1.2 Estructura de los casos en OpenFOAM	76
6.1.1.3 Geometría y mallado	78
6.1.1.4 Solver	78

CONTENIDO

6.1.1.5 twoLiquidMixingFoam.C	79
6.1.2 Uso del solucionador	80
6.1.2.1 Simulación de un caso laminar	80
6.1.2.2 Simulación de un caso turbulento	81
6.1.2.3 Mapeo de campos y solución en paralelo (MPI)	81
6.2 SOLUCIONADOR DESARROLLADO	82
6.2.1 Malla móvil	82
6.2.1.1 MVF en mallas poliédricas móviles	83
6.2.2 Manipulación de mallas en OpenFOAM	84
6.2.3 Modificaciones al solver twoLiquidMixingFoam	84
7 RESULTADOS	91
7.1 SIMULACIONES DE MEZCLADO AXIAL	91
7.1.1 Modelos y métodos numéricos empleados	91
7.1.2 Modelo	92
7.2 SIMULACIÓN DE MEZCLADO AXIAL CON OpenFOAM	93
7.2.1 Modelos y métodos numéricos empleados	93
7.2.2 Modelo	94
7.2.2.1 Condiciones de frontera	95
7.2.2.2 Esquemas numéricos	95
7.2.2.3 Resultados twoLiquidMixingFoam	95
7.3 ANÁLISIS DE RESULTADOS twoLiquidMixingFoam	98
7.3.1 Relación Δt -longitud de mezcla	98
7.3.2 Variables de flujo	101
7.4 RESULTADOS CON <i>twoLiquidMixingDyMFoam</i>	101
7.4.1 Esquemas numéricos	101
7.4.2 Caso 2D 15 metros estático	102
7.4.3 Caso 2D 10 metros dinámico	103
7.5 ANÁLISIS DE RESULTADOS twoLiquidMixingDyMFoam	104
7.5.1 Diferencia de longitud de mezcla en el tiempo	104
8 CONCLUSIONES Y RECOMENDACIONES	108
8.1 CONCLUSIONES	108
8.2 RECOMENDACIONES	109
REFERENCIAS BIBLIOGRÁFICAS	111

CONTENIDO

ANEXOS

115

LISTA DE FIGURAS

		Pág.
1	Secuencia de bombeo típica	19
2	Instalación esquemática de transporte secuencial o batch	24
3	Perfil de concentración en la interfase a una distancia L y $t > 0$	25
4	Perfil de concentración en $t = 0$ y $x = 0$	25
5	Evolución de la concentración en la distancia y el tiempo	25
6	Difusión molecular	26
7	Estratificación de dos fluidos A y B , con A mas denso que B	28
8	Balance de masa en el elemento	35
9	Balance de fuerzas en dirección x	39
10	Trabajo de las fuerzas de superficie x	43
11	Escalas de tiempo en turbulencia	50
12	Malla estructurada	56
13	Discretización Espacio y tiempo	57
14	Volumen de control	58
15	Vectores \mathbf{d} y \mathbf{S} no ortogonales	62
16	Algoritmo PISO	73
17	Estructura de OpenFOAM	76
18	Estructura de caso	77
19	Algoritmo de twoLiquidMixingFoam	79
20	Modificaciones a twoLiquidMixingFoam	85
21	Resultados malla Zhao, $20 \times 4 \text{ mm}$	96
22	k a 4 metros de la entrada	96

LISTA DE FIGURAS

23	épsilon a 4m de la entrada	97
24	alpha1 a 4m de la entrada	97
25	Perfil de velocidad a 4m	97
26	Longitud de mezcla 2mm	98
27	Longitud de mezcla 4mm	99
28	Δt Vs longitud para 1,2 y 4mm	99
29	Inicio y fin de mezcla 2 mm	100
30	Inicio y fin de mezcla 4mm	100
31	Perfiles 5 y 6 segundos	102
32	Perfiles 7 y 8 segundos	103
33	Perfiles a 9 y 10 segundos	103
34	Perfiles 5 y 6 segundos	103
35	Perfiles 7 y 8 segundos	104
36	Perfiles 9 y 10 segundos	104
37	Crecimiento de la variación de longitud	105
38	Residuales malla Dinámica	106
39	Residuales de malla estática	107

LISTA DE TABLAS

		Pág.
1	Ecuaciones del modelo	54
2	Valores para Γ_{eff}^ϕ y S_ϕ	58
3	Propiedades de los fluidos	93
4	Características del dominio computacional	93
5	Condiciones de frontera	94
6	Resultados Zhao para 4 segundos	94
7	Dominio computacional	94
8	Condiciones de frontera	95
9	Esquemas numéricos	95
10	Esquemas numéricos términos convectivos	95
11	Resumen de resultados	96
12	Esquemas numéricos	102
13	Esquemas numéricos términos convectivos	102
14	Resumen de resultados malla dinámica	105

LISTA DE ANEXOS

	Pág.
Anexo A. DESCRIPCIÓN DE twoLiquidMixingFoam.C	115
Anexo B. ARCHIVOS NECESARIOS PARA LA SIMULACIÓN	132
Anexo C. ARCHIVO twoLiquidMixingDyMFoam.C	152

RESUMEN

Título: Determinación del campo de flujo y de concentraciones de la interfase de dos fluidos incompresibles que se desplazan a través de una tubería usando el método de volúmenes finitos.¹

Autor: Carlos Alberto Flórez Arias²

Palabras claves:

CFD, FVM, mezcla, programación orientada a objetos, software libre, malla dinámica, Open-FOAM.

Descripción:

La industria petroquímica moviliza fluidos mediante redes de tuberías empleando el método de transporte por lotes. Cuando no existe ninguna separación rígida entre cada dos lotes, los productos se mezclan en cierta medida mientras se desplazan, dando lugar a un volumen de mezcla o interfase contaminada. El volumen de mezcla aumenta a lo largo de la tubería como resultado de la influencia del trazado topográfico, efectos de la gravedad y las propiedades de los fluidos: viscosidad, densidad, coeficiente de difusión, etc. En esta Trabajo se muestran los resultados obtenidos en la determinación de dicha interfase para un flujo en régimen isotérmico turbulento totalmente desarrollado por medio de la Dinámica de Fluidos Computacional (CFD) usando el método de volúmenes finitos (FVM) y empleando el software libre OpenFOAM®, con el cual se simula un caso. también se muestra el desarrollo de un solucionador con la adición de funcionalidad de malla Dinámica, este solucionador fue utilizado para simular este proceso de transporte y posteriormente se analizaron los resultados obtenidos. En esta investigación también fueron usados los software SALOME en la etapa de preproceso y PyFoam en las etapas de preproceso y posproceso, así como el lenguaje de programación Python™. Este proyecto empleo exclusivamente software libre en todas sus etapas de desarrollo.

¹Trabajo de grado

²Facultad de Ingenierías Físico-Mecánicas. Escuela de ingeniería mecánica. Director: David Fuentes Díaz, Doctor en tecnología energética.

ABSTRACT

Title: Flow and concentration fields determination in the interface of two incompressible fluids when they are moving through a pipe using the finite volume method.³

Author: Carlos Alberto Flórez Arias⁴

Keywords:

CFD, FVM, mixing, object oriented programing, free software, Dynamic mesh, OpenFOAM.

Description:

The petrochemical industry mobilizes fluids through pipe networks using the batch transportation method. When there is no rigid separation between any two batches, the products are mixed to some extent while they are moving, resulting in a mixed volume or contaminated interface. The mixing volume increases along the pipe as a result of the influence of topography path, effects of gravity and fluid properties: viscosity, density, diffusion coefficient, etc. This work shows the results obtained in the determination of that interface for a flow in fully developed turbulent and isothermal regime using Computational Fluid Dynamics (CFD), using the finite volume method (FVM) and the free software OpenFOAM®, with which a case was simulated. The development of a solver with adding dynamic mesh functionality which this software was used to simulate the process of transport and subsequently analyzed results are also shown. In this study was also used in the preprocessing step SALOME software and in preprocessing and post-processing stages PyFoam software and the Python™ programming language. This project use exclusively free software at all stages of development.

³Degree work

⁴Facultad de Ingenierías Físico-Mecánicas. Escuela de ingeniería mecánica. Advisor: David Alfredo Fuentes Díaz, PhD. energetic technology

INTRODUCCIÓN

La industria petrolera utiliza el método de transporte por tuberías para transportar gran cantidad de productos a través de kilómetros de poliductos. En esta aplicación es de uso común el transporte secuencial por baches o lotes en el mismo conducto, separando o no los baches por medio de un dispositivo sólido. Sin embargo, este método es poco utilizado debido a sus costos y a problemas de seguridad asociados a la inserción y extracción del dispositivo sólido en el conducto. Por esta razón se utilizan fluidos separadores (lotes buffer o cuñas), aunque la mayoría de las veces no se utiliza ninguna separación. En una secuencia de bombeo típica se organizan los productos con densidades cercanas y entre ellos se insertan los llamados lotes “buffer” o cuñas. Al ser transportados, y debido a la acción del campo de flujo, la diferencia de densidades y la difusión molecular, tiene lugar la dilución entre productos o de los productos con la cuña, generando un volumen de mezcla en el que se ven alteradas las propiedades fisicoquímicas de estos.

Debido a este fenómeno las políticas de calidad de la industria petrolera consideran de vital importancia determinar la longitud de volumen de mezcla en cada parte de la línea de transporte en un determinado instante de tiempo, dado que esta zona debe ser extraída para ser reprocesada, ya sea mezclándola en baja concentración con un lote donde resulte aceptable en términos de calidad o para ser llevada nuevamente a refinería[1] .

Con el propósito de garantizar la calidad y de evitar incurrir en sobrecostos durante el transporte de los productos derivados del petróleo, para la industria petrolera es fundamental la investigación sobre el comportamiento de los fluidos en la tubería, para esto es necesario determinar el campo de flujo en la zona de mezcla y definir el campo de concentraciones de cada producto. Adicionalmente, para la academia es importante promover la apropiación de conocimiento sobre el comportamiento de los fluidos con el objeto de aportar herramientas eficaces y eficientes que ayuden a jalonar los sectores estratégicos del país. En este trabajo de investigación se busca establecer el campo de flujo y de concentraciones para dos fluidos incompresibles miscibles que se desplazan por una tubería, usando la dinámica de fluidos computacional (CFD). Con este fin se

planteo el uso una herramienta computacional que empleara las librerías libres escritas en C++ OpenFOAM.

1 PLANTEAMIENTO DEL PROBLEMA

Los principales medios de transporte de crudo y sus derivados son los oleoductos, poliductos, gasoductos, carrotanques, transporte fluvial (barcazas), además del cabotaje. De estos medios los más usados son el transporte por oleoductos, poliductos y gasoductos debido a las cantidades de producto, alto grado de confiabilidad y costos asociados a las otras formas de transporte.

En el transporte de hidrocarburos por poliductos se emplea el método batch en el cual cada producto que es introducido a la tubería empuja al producto que se introdujo antes y así sucesivamente. Cuando no existe ninguna separación rígida entre cada dos batches, los productos se mezclan en cierta medida mientras se desplazan, dando lugar a una zona de mezcla o interfase contaminada. La determinación del volumen de esta interfase en determinado lugar en un momento específico representa un punto crítico en las políticas de calidad de la industria del transporte de crudo y destilados de petróleo, debido a que este volumen de mezcla debe ser extraído para ser reprocesado antes de suministrar los productos a los clientes. Este reproceso en Colombia consiste en diluirlo en algún lote donde no altere sus propiedades como el punto de encendido para diésel o punto de chispa para gasolinas en los destilados y el grado API en crudos. En la Figura 1 se muestra la secuencia típica de bombeo de combustibles en los poliductos Colombianos.

Las estadísticas volumétricas de la industria petrolera en Colombia muestran para el año 2007, según Ecopetrol [3], que el promedio de refinados transportados tanto en redes primarias como en redes secundarias fue de 537, 2 Kbpd, también se muestra que el promedio de crudo transportado en redes primarias y secundarias fue de 1151,21 Kbpd mas 14,28 Kbpd en el oleoducto transandino, y para combustóleo incluyendo gasóleo se tiene 67,6 Kbpd. Para 2009 se transportaron 576,2 Kbpd de crudos, 223,3 Kbpd de refinados. Todo esto a través de su red de 8.500 kilómetros de poliductos y oleoductos, que van desde los centros de producción hasta las refinerías y puertos en los océanos Atlántico y Pacífico.

El volumen contaminado está influenciado por varios factores como la difusión molec-

Figura 1: Secuencia de bombeo típica



Fuente: Tomado de CONCHA , QUIROGA, BENJUMEA 2007 [2]

ular, la flotación debida a la diferencia de densidades, la diferencia de viscosidades y el efecto del campo de velocidades del flujo (laminar o turbulento) de los productos que se desplazan.

El fenómeno de crecimiento del volumen de mezcla comienza con la entrada del segundo fluido en la tubería, desde este instante empieza a actuar la difusión molecular causando la aparición de un volumen contaminado, este volumen también se ve influenciado por el campo de velocidades que puede ser estanco, laminar, de transición o turbulento. Además, si existe una diferencia de densidades entre los fluidos y está presente un campo gravitatorio, aumenta la magnitud del área de exposición entre los fluidos debido al efecto de flotación que se induce y por lo tanto también aumenta la cantidad mezclada. Debido a que el campo de velocidades depende de la densidad y de la viscosidad, la densidad y viscosidad dependen del campo de concentraciones y adicionalmente el campo de concentraciones depende del campo de velocidades, se presenta un problema acoplado en densidad, viscosidad, concentración y velocidades.

Este trabajo de investigación se determinó el campo de flujo y de concentraciones de dos fluidos incompresibles miscibles que se desplazan en una tubería con propiedades constantes bajo condiciones de régimen de flujo turbulento totalmente desarrollado, mediante el uso de una herramienta computacional CFD.

2 JUSTIFICACIÓN

La tendencia de crecimiento que registra el sector hidrocarburos en los últimos años y su importancia para la economía Colombiana, hace necesario que se investiguen los fenómenos directamente relacionados con la calidad de los productos y costos de operación con el fin de desarrollar herramientas que permitan aumentar la eficacia y la eficiencia del sector.

Teniendo en cuenta que el medio más empleado para transportar hidrocarburos es el poliducto, es importante conocer a fondo este medio de transporte y sus características, Un fenómeno característico de este medio de transporte es la aparición de un volumen de mezcla entre los productos transportados de manera secuencial y la determinación de este volumen contaminado depende del comportamiento del campo de flujo.

Muchos investigadores[4][5][6][7] han desarrollado modelos a partir de datos experimentales para determinar la longitud de la zona de mezcla pero estos modelos desestiman los efectos de algunos factores o solo los consideran parcialmente, como en el modelo dispersivo de régimen turbulento de Taylor [5] donde se elimina la difusión molecular en la dirección radial y axial y se considera la difusión turbulenta solo en dirección radial, debido a la complejidad que involucra encontrar una solución que relacione la combinación de estos factores y su contribución al aumento de la longitud de la zona de mezcla dado el acoplamiento que se presenta.

Para la industria del transporte de hidrocarburos es importante la solución de problemas de volúmenes de mezcla en transporte batch, debido a que las simulaciones numéricas requeridas para resolver este tipo de problemas contribuyen al desarrollo y apropiación de conocimiento sobre el comportamiento de los fluidos, siendo más significativas cuando están orientadas a resolver problemas de calidad que se presentan en uno de los sectores vertebrales de la economía de un país. En Colombia, Ecopetrol y su filial Cenit, son las empresas responsables del transporte y disponibilidad oportuna de los diferentes hidrocarburos para refinación, exportación o consumo y para éstas es fundamental contar con una herramienta útil en la toma de decisiones encaminadas al mejoramiento continuo de sus procesos y aprovechamiento máximo de los recursos con los que cuenta

nuestro país.

La complejidad agregada al problema por las contribuciones potenciales de dichos factores y la posibilidad de extender su uso a geometrías variables son las principales razones para utilizar CFD al resolver el modelo matemático resultante y no implementar un modelo dispersivo de un solo producto, como los desarrollados por Taylor, Aris y otros investigadores. Actualmente diferentes grupos de investigación[8][9][10][11] trabajan en el desarrollo de modelos para realizar simulaciones por medio de las cuales se pueda determinar el volumen de mezcla de dos productos transportados secuencialmente en tuberías. La herramienta usada está orientada a determinar el campo de flujo de dos fluidos incompresibles miscibles que se desplazan a través de una tubería, considerando las propiedades de cada uno de los fluidos constantes y flujo isotérmico, teniendo en cuenta la interacción de factores que no han sido incorporados en modelos dispersivos donde se asume flujo axisimétrico, como la influencia en la mezcla axial de la diferencia de densidades, la difusión molecular, el campo de velocidades y viscosidades de los fluidos involucrados.

También es importante resaltar la apropiación de conocimiento en el uso de software libre especializado en investigación, mas concretamente en CFD, debido a los grandes avances alcanzados en esta materia, así como el ahorro económico, dado el alto costo de las licencias y limitaciones de software especializado.

3 OBJETIVOS DEL TRABAJO DE INVESTIGACIÓN

3.1. OBJETIVO GENERAL

Determinar el campo de flujo y de concentraciones que se establece en la interfase de dos fluidos incompresibles miscibles transportados a través una tubería bajo régimen isotérmico turbulento totalmente desarrollado.

3.2. OBJETIVOS ESPECÍFICOS

- Realizar una revisión bibliográfica del estado del arte del modelado de flujo de fluidos incompresibles miscibles dentro de una tubería.
- Elaborar un modelo matemático que describa el comportamiento del flujo en la interfase de dos fluidos incompresibles miscibles al desplazarse por una tubería.
- Discretizar el sistema de ecuaciones resultante del modelo de flujo utilizando la técnica de los volúmenes finitos.
- Implementar mediante código de computador la metodología de solución del sistema de ecuaciones discretizadas utilizando las librerías de código abierto OpenFOAM®.
- Simular el campo de flujo en la interfase de dos fluidos impulsados secuencialmente por una tubería.
- Verificar los resultados obtenidos mediante la comparación con resultados reportados en la literatura.

4 MARCO TEÓRICO

4.1. CONCEPTOS FUNDAMENTALES

4.1.1. Tendencia de la producción petrolera.

La producción mundial de petróleo se podría dividir en la producción de los países miembros de la OPEP, quienes extraen 40% del petróleo mundial, con 24,84 millones de barriles diarios (mbd), y el 60 % restante producido por los demás países. Colombia, quien no es miembro de la OPEP, ha venido experimentando un incremento en la producción de petróleo, después de observarse una caída en la producción desde el año 1999 hasta el año 2004, con una tasa de -7,9%. Luego de 2005 se observó un aumento en la producción, incrementándose con porcentajes cercanos al 1% anual, para que después, en 2008 y 2009 se dieran unos porcentajes de crecimiento del 10,1% y 4,7% respectivamente, coincidiendo con la creación de la agencia nacional de hidrocarburos [12]. En 2010 se presenta un incremento en la producción de 114.000 barriles diarios, con un promedio de 785 kbpd [13], La producción subió 16,3% interanual en el 2011, a 913.583 barriles por día (bpd), en 2012 se registró un promedio de 944.000 barriles de petróleo por día (bpd), es decir 3,16% más frente al año anterior. El Ministerio de Minas y Energía informó que en noviembre de 2013 la producción promedio de petróleo alcanzó 1.012.000 barriles por día, de esta forma, la producción diaria estimada promedio anual de crudo alcanza el 1.007.000 BPD.

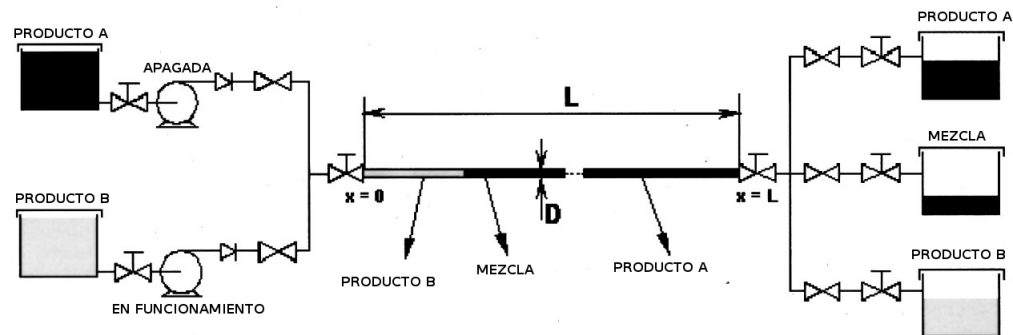
Actualmente Ecopetrol tiene dos refinerías, Cartagena y Barrancabermeja con una producción a 2004 de 78.000 barriles por día y 240.000 barriles por día respectivamente. El 71 % abastece mercado nacional y 29% exportaciones, dos minirefinerías: Apiay 2.500 barriles por día, destinada básicamente a producir Asfalto y Orito 2.000 barriles por día, a abastecer las necesidades internas, también se cuenta con una refinería privada de 14.000 barriles por día en el Magdalena Medio (Refinare).

4.1.2. Características del método de transporte

4.1.2.1. Transporte secuencial o “batch”

El sistema de transporte tipo batch consiste en enviar de manera secuencial dos o mas fluidos por una tubería[14].

Figura 2: Instalación esquemática de transporte secuencial o batch



Fuente:[14]

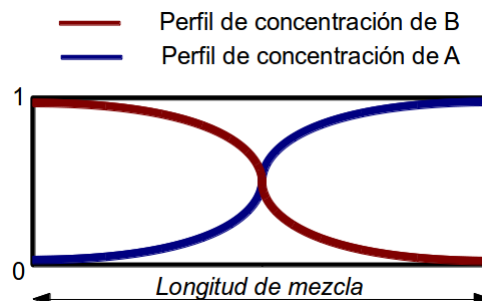
En la Figura 2 se puede observar un esquema de una instalación para el transporte tipo batch de dos productos A y B por una tubería de longitud L y diámetro D . Inicialmente se está bombeando el producto A, en el instante 0 se detiene el suministro de A mediante el accionamiento de una válvula, e inmediatamente se inicia el bombeo del producto B, de tal forma que el producto B empuja al producto A. Desde el momento en que se introduce el fluido B y si los fluidos A y B son miscibles, comienza a diluirse uno en el otro, dando origen a un volumen contaminado que aumenta a medida que los productos se desplazan una distancia L dentro de la tubería, bajo la influencia de los fenómenos de transferencia de masa y las características del flujo que tienen lugar.

4.1.3. Fenómenos de transferencia de masa en transporte secuencial de fluidos.

Cuando dos fluidos son transportados de manera secuencial por una tubería, se presentan un conjunto de fenómenos que van transformando el perfil de concentración a lo largo de la tubería. Dentro de los fenómenos de transferencia de masa se encuentran la difusión molecular, la difusión laminar y la difusión turbulenta.

Inicialmente el perfil de concentraciones tiene la forma descrita en la Figura 4, donde se muestra el instante en el que se empieza a alimentar el fluido B dentro de la tubería.

Figura 3: Perfil de concentración en la interfase a una distancia L y $t > 0$



La Figura 5 muestra la evolución de la concentración de una de la sustancias a lo largo de la línea. Luego de recorrer una distancia L el perfil de concentraciones presenta el comportamiento mostrado en la Figura 3.

Figura 4: Perfil de concentración en $t=0$ y $x=0$

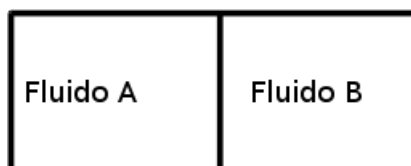
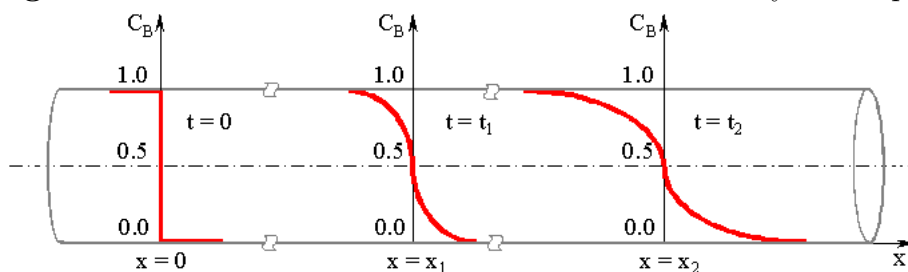


Figura 5: Evolución de la concentración en la distancia y el tiempo



Fuente: Tomado de Baptista, Freitas, Araujo, 2001[11]

4.1.3.1. Difusión molecular

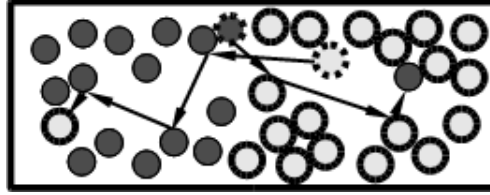
Es la transferencia de moléculas a través de un fluido gracias a los desplazamientos individuales y desordenados de estas, aparece debido a un gradiente de concentración. Se puede pensar en las moléculas desplazándose en línea recta y cambiando su dirección al rebotar contra otras cuando chocan (ver Figura 6). Debido a que las moléculas se desplazan en trayectorias al azar, la difusión molecular también es llamada proceso con

trayectoria aleatoria. Este proceso tiende a uniformar la concentración de las sustancias en contacto y es particularmente importante en el análisis de la mezcla de fluidos en reposo durante un largo tiempo. Si se observa una partícula en especial, se encontrará que, en promedio, después de un cierto tiempo t habrá avanzado una distancia neta que es proporcional a la raíz cuadrada de ese tiempo, tenemos:

$$x_{cm} = \sqrt{2Dt} \quad (4.1)$$

Donde el subíndice cm se usa para cuadrático medio, D es el coeficiente de difusión molecular, el cual tiene las mismas unidades de la viscosidad cinemática. De la Ecuación 4.1 se puede inferir que la difusión molecular es un proceso bastante lento.

Figura 6: Difusión molecular



Debido a los resultados poco satisfactorios de la teoría de difusión en líquidos, se utilizan relaciones empíricas en la estimación de su coeficiente de difusión, aunque su determinación se hace mediante mediciones experimentales. La Ecuación 4.2 es llamada la ecuación de Wilke-Chang, valida para pequeñas concentraciones de A en B.

$$D_{AB} = 7.4 * 10^{-8} \frac{\sqrt{\psi_B M_B T}}{\mu_A \tilde{V}_A^{0.6}} \quad (4.2)$$

Donde \tilde{V}_A es el volumen molar del soluto en $cm^3/g-mol$ como líquido en su temperatura de ebullición, ψ_B es un parámetro de asociación, y μ es la viscosidad en centipoises. La temperatura T en kelvin.

4.1.3.2. Difusión laminar

Se presenta debido al régimen de flujo del mismo nombre, en este fenómeno la energía suministrada por el movimiento aumenta el volumen de mezcla, por ello en el flujo laminar de fluidos por una tubería tiene una mayor incidencia sobre la cantidad mezclada

que la difusión molecular. Este coeficiente ha sido calculado a partir de mediciones de la difusividad longitudinal, la dimensión característica del ducto y la velocidad media del flujo, con los cuales se obtiene la difusividad laminar a partir de una ecuación que describa la dispersión longitudinal como la Ecuación 4.3.

$$D = \frac{a^2 U^2}{48 D_{AB}} \quad (4.3)$$

Donde a es la dimensión característica del ducto, U es la velocidad promedio y D_{AB} es el coeficiente de difusividad molecular.

4.1.3.3. Difusión turbulenta

Este fenómeno es mas efectivo en el transporte de masa que los dos fenómenos anteriores, se presenta debido a los vórtices generados en el flujo, es esencialmente un fenómeno anisotrópico, donde la energía suministrada por los vórtices presentes en el flujo inducen un mayor volumen de mezcla. El cálculo de la difusión turbulenta se consigue a partir de el numero de Schmidt turbulento

$$D_{tAB} = \frac{\nu_t}{Sc_t} \quad (4.4)$$

Donde ν_t es la viscosidad cinemática turbulenta, D_{tAB} es la difusividad turbulenta, en está ecuación la viscosidad cinemática se calcula por medio de la Ecuación 4.5 y la difusividad se determina conociendo el numero de Schmidt característico del flujo.

$$\nu^{1/3} = C_A \nu_A^{1/3} + C_B \nu_B^{1/3} \quad (4.5)$$

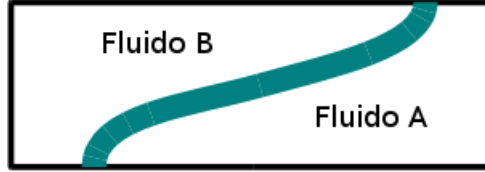
Donde C_A y C_B son las concentraciones de cada una de las sustancias y ν_A y ν_B son las viscosidades cinemáticas del fluido A y B.

4.1.3.4. Estratificación

Una diferencia en las densidades de los dos fluidos en contacto puede generar un gradiente apreciable entre las fuerzas de inercia de los productos transportados, propiciando que el fluido más denso tienda a posicionarse bajo el menos denso Figura 7. Este proceso tiene una mayor importancia en fluidos que se mantienen en reposo durante largos

periodos de tiempo, como una parada en una línea de bombeo, también es relevante en casos donde los perfiles topográficos muestran cambios de nivel importantes. Este fenómeno aumenta el área de contacto entre los dos fluidos y por lo tanto aumenta la cantidad mezclada.

Figura 7: Estratificación de dos fluidos A y B, con A mas denso que B



4.2. ESTADO DEL ARTE

La teoría de dispersión axial en fluidos Newtonianos para fluidos con flujo laminar en tubos diámetro pequeño inicialmente fue propuesta por Taylor[4] en 1953, quien desestima la contribución de la difusión molecular en dirección axial y radial, además considera difusión turbulenta solamente en la dirección radial. Asume velocidad promedio y propone el coeficiente de difusión virtual K definido como:

$$K = \frac{a^2 U^2}{48 D_{AB}} \quad (4.6)$$

Donde a es el radio de la tubería, U es la velocidad media del flujo y D_{AB} es el coeficiente de difusión molecular entre el soluto y el solvente o entre los dos fluidos A y B (en el caso de dos fluidos miscibles). En este trabajo, Taylor usa la ecuación que gobierna la dispersión longitudinal:

$$K \frac{\partial^2 C_\alpha}{\partial x_1^2} = \frac{\partial C_\alpha}{\partial t} \quad (4.7)$$

la cual representa el efecto combinado de la variación de la velocidad en la sección transversal y la difusión molecular en la dirección axial en un flujo laminar, realiza experimentos y compara los resultados con los esperados por el modelo. En 1954 Taylor [5] propone la descripción de la dispersión axial en un flujo turbulento que fluye por

una tubería circular definiendo:

$$K = 10,1aV^* \quad (4.8)$$

ó

$$K = 7,14aU\sqrt{\gamma} \quad (4.9)$$

Donde V^* es la “velocidad de fricción” y γ es el coeficiente de resistencia, pero sus resultados esta restringidos a números de Reynolds mayores a 20,000 y a fluidos que tengan un numero de Schmidt igual a 1. También establece una ecuación para la longitud de mezcla en líquidos miscibles que se desplazan por una tubería como:

$$S^2 = 437 a X \left(\frac{V^*}{U} \right) \quad (4.10)$$

Donde S es la longitud de mezcla para los dos líquidos miscibles.

Luego compara sus resultados con los obtenidos para oleoductos en experimentos realizados en Estados Unidos. Posteriormente R. Aris en 1956 [6] propone una modificación al coeficiente de dispersión para el caso de un flujo en un tubo circular como

$$K = D_{AB} + \frac{a^2 U^2}{48 D_{AB}} \quad (4.11)$$

donde K es la suma de el coeficiente de difusión molecular y el coeficiente de difusión aparente o “virtual” descrito por Taylor en Ecuación 4.6. también propone un modelo general para K con variación en el coeficiente de difusión molecular en un ducto con sección transversal arbitraria.

$$K = D_{AB} + k \frac{a^2 U^2}{48 D_{AB}} \quad (4.12)$$

donde k es constante. Otros autores [15][9] hacen modificaciones al coeficiente K .

Muchos investigadores emplean la nomenclatura usada por Taylor [5] en la solución a la ecuación de conservación de las especies químicas para los modelos dispersivos en una malla que se desplaza con la misma velocidad promedio del fluido, u es la velocidad

promedio en el tiempo definida como :

$$u(t) = \frac{1}{A} \int_A \bar{u}(r, t) r dr d\theta = \frac{1}{A} \int_A \left[\frac{1}{2\Delta t_{t-\Delta t}} \int_{t-\Delta t}^{t+\Delta t} \hat{u}(r, \tau) d\tau \right] r dr d\theta \quad (4.13)$$

También tiene que $K = K(u, C_\alpha)$ es el coeficiente efectivo de dispersión axial definido como:

$$K(u, C_\alpha) = \nu Re \left[\int_0^1 \frac{\left[\int_\xi^1 \left(\frac{\bar{u}}{u} - 1 \right) \xi' d\xi' \right]^2}{2\xi \left\{ \frac{\xi f}{16} \left[\left| \frac{\delta}{\delta \xi} \left(\frac{\bar{u}}{u} \right) \right| \right]^{-1} + \left(\frac{1}{Sc} - 1 \right) \frac{1}{Re} \right\}} \right] \\ + \nu Re \left[\int_0^1 2\xi \left\{ \frac{\xi f}{16} \left[\left| \frac{\delta}{\delta \xi} \left(\frac{\bar{u}}{u} \right) \right| \right]^{-1} + \left(\frac{1}{Sc} - 1 \right) \frac{1}{Re} \right\} \right] \quad (4.14)$$

Donde f es el coeficiente de fricción de Darcy-Weisbach y $\xi = \frac{2r}{\varphi}$, $Re = Re(u, C_u) = \frac{\varphi u}{\nu}$ y $Sc = Sc(C_\alpha) = \frac{\nu}{D}$, donde φ es el diametro del ducto, D es la difusividad molecular, además se define $\nu = \nu(C_\alpha)$ como la viscosidad cinemática, para la cual se tiene:

$$\nu^{1/3} = C_A \nu_A^{1/3} + C_B \nu_B^{1/3}$$

En 2002 Freitas , Carneiro y Baptista [14] plantean un modelo para predecir el volumen mezclado presente en poliductos para transporte batch sin separadores físicos para la zona de transición y turbulenta, ese modelo es resuelto por elementos finitos acoplados por el método de newton. En 2003 Ekambara y Joshi [8] simulan el patrón de flujo del flujo de fluidos por una tubería usando el modelo de bajo número de Reynolds en régimen de transición y turbulento, luego comparan los resultados del coeficiente de dispersión obtenido en las simulaciones con los publicados en la literatura.

En 2003, Freitas , Carneiro y Baptista [11] hacen una publicación en la cual evalúan el volumen de mezcla en procedimientos de parada o arranque en la línea de bombeo, asumen que el coeficiente de dispersión depende de la concentración, también asumen el perfil de velocidades en régimen de transición igual al perfil estable para el mismo número de Reynolds y toman dos sustancias con pesos específicos similares. La solución numérica aproximada de este problema es obtenida por medio del método Runge-Kutta de cuarto orden para la ecuación de momento y un esquema predictivo-corrector de

diferencias finitas para la ecuación de concentración. Por medio de este modelo se pudo concluir que si la línea se detiene mas de tres veces durante la transferencia, el volumen de mezcla se puede incrementar hasta un 9% con relación a la cantidad mezclada cuando la línea no tiene paradas.

En 2010 Zhao, Derksen, Gupta [10] hacen una simulación numérica basada en CFD para determinar el comportamiento de la mezcla de dos fluidos miscibles para la gasificación subterránea de carbón. Primero se verificó el modelo realizando las simulaciones para una tubería de 10 m de largo, luego se hicieron las simulaciones para un tramo de 100 m y con estos resultados se extrapola para longitudes mayores.

En Colombia, Zárate Neira en 2003 [16] publicó un método empírico para predecir el tamaño del volumen de mezcla, allí también se menciona las variables involucradas en el proceso de mezcla. En 2007, Concha, Quiroga, Benjumea [2] proponen un modelo matemático para el crecimiento del volumen de mezcla entre combustibles derivados del petroleo como consecuencia del transporte secuencial de los mismos, trabajan con el coeficiente efectivo de transferencia de masa, luego este modelo es implementado en un tramo del poliducto Sebastopol – Medellín, usando un combustible liviano (gasolina) y uno mediano (diesel), separados con cuñas de virgin oil, se analizaron las variables tasa de flujo, diámetro y longitud del poliducto y se concluyó que la variable que presenta mas incidencia en el crecimiento del volumen de mezcla es la longitud de la tubería. En 2011 Díaz, Corzo, Flórez, Carrillo, Barajas, Cárdenas, García [17] usando la ecuación empírica propuesta por Sjenitzer [15], obtienen expresiones para estimar el tiempo de corte en función de la concentración admisible y finalmente mediante un análisis estadístico establecen que el cambio de diámetro de la tubería ejerce un mayor efecto sobre el volumen de contaminación que el cambio de la longitud del ducto o la variación del caudal.

4.3. MÉTODOS DE SOLUCIÓN

La simulación de flujos incompresibles es un tópico muy importante en el desarrollo de la dinámica de fluidos computacional (CFD). Para resolver los problemas de flujos se emplean métodos basados en la malla como el método de los elementos finitos (FEM), el cual emplea una función de interpolación seleccionada debidamente, usualmente en una malla no estructurada para generar la matriz del sistema de ecuaciones obtenidas a través de la aplicación del principio variacional o algún método de residuos ponderados

(WRM) , también se usa el método de las diferencias finitas (FDM) y el método de los volúmenes finitos (FVM), estos dos métodos involucran la solución del acoplamiento de la presión y la velocidad en una malla escalonada o traslapada. También se utilizan métodos sin malla como el método de la partícula en la celda (PIC), y otro similar, el método de marco y celda (MAC), estos últimos a pesar de ser usados para resolver problemas que incluyen flujos reactivos, flujos multimaterial, flujos multifase y flujos con discontinuidades espaciales, son bastante complicados de programar y representan un costo computacional bastante grande, especialmente para flujos tridimensionales [18]. Recientemente se han desarrollado el método de volumen de fluido (VOF) y el método level-set. La simulación de interfaces en movimiento o cualquier tipo de frontera en movimiento en una malla dispuesta en el espacio del problema, como es el caso de este trabajo de investigación, son comúnmente modelados por FDM y FVM. En este trabajo de investigación se usó el método de los volúmenes finitos (FVM).

4.3.1. Método de los volúmenes finitos (FVM)

El método de discretización de los volúmenes finitos tiene las siguientes características[19]:

- Este método está basado en la discretización de la forma integral de las ecuaciones gobernantes sobre un volumen de control, en el cual, las cantidades tales como masa y cantidad de movimiento se deben conservar a nivel discreto.
- Las ecuaciones son resueltas en un sistema de coordenadas cartesiano fijo en una malla que no cambia en el tiempo. Este método es aplicable a problemas estacionarios y transitorios.
- Los volúmenes de control pueden tener una forma general poliédrica con un número variable de vecinos, por tanto pueden crear una malla estructurada. Todas las variables dependientes comparten el mismo volumen de control, con arreglos comúnmente conocidos como de variables cúbicos o no escalonados.
- Los sistemas de ecuaciones diferenciales parciales son tratados de forma segregada[20] , lo que quiere decir que son resueltas una a la vez, con el acoplamiento entre las ecuaciones tratado de forma explícita. Las Ecuaciones diferenciales no lineales son linealizadas antes de la discretización y los términos no lineales son pospuestos.

5 MODELAMIENTO DE LA MEZCLA DE DOS FLUIDOS EN TRANSPORTE BATCH

5.1. ECUACIONES GOBERNANTES

5.1.1. Introducción

En el presente trabajo se aborda el enfoque Euleriano para la solución del problema de mezcla de dos fluidos incompresibles miscibles que se desplazan por una tubería en tres dimensiones, posteriormente es usado el enfoque Lagrangiano-Euleriano para la solución del mismo problema en dos dimensiones. En el enfoque Euleriano las dos sustancias son tratadas como un medio continuo, cada uno de los cuales se interpenetra con la otra y se representa por un conjunto de ecuaciones de conservación que son válidas a lo largo de todo el dominio. Este enfoque es conocido como enfoque multi-fluido (generalizando para el caso en que se consideren mas de dos sustancias). En el enfoque Lagrangiano-Euleriano las dos sustancias tienen el mismo tratamiento como un medio continuo interpenetrandose cada una con la otra pero en este caso el dominio se desplaza con la velocidad media de las sustancias.

El dominio del problema inicial está dado en tres dimensiones, pero por claridad en su definición, las ecuaciones se deducen en dos dimensiones. Al final de cada deducción se plantea la ecuación de forma vectorial.

5.1.2. Ecuación de conservación de la masa

Esta ecuación se desarrolla a partir de un elemento de dimensiones $\Delta x \Delta y$ con espesor unitario y fijo en el espacio. Para este elemento se tiene que la masa contenida en él está dada por la suma de la contribución de cada una de las especies, de forma que:

$$\delta m = \sum_{i=1}^n \delta m_i \quad (5.1)$$

con

$$\delta m_i = \rho \alpha_i \delta \forall \quad (5.2)$$

donde

$$\alpha_i = \frac{\rho_i}{\rho} \quad (5.3)$$

Donde α_i se conoce como la concentración másica de la especie i , además la fracción másica ρ_i es la densidad de la especie i en el elemento, $\rho_i = \delta m_i / \delta \forall$, si se define la masa total en el volumen de control m como

$$m = m_A + m_B \quad (5.4)$$

con ρ como la densidad de la mezcla entonces

$$\rho \delta \forall = \rho_A \delta \forall + \rho_B \delta \forall \quad (5.5)$$

de lo cual se deduce que

$$\rho = \sum_{i=1}^n \rho_i \quad (5.6)$$

y sumando los aportes de las especies en la Ecuación 5.3 queda:

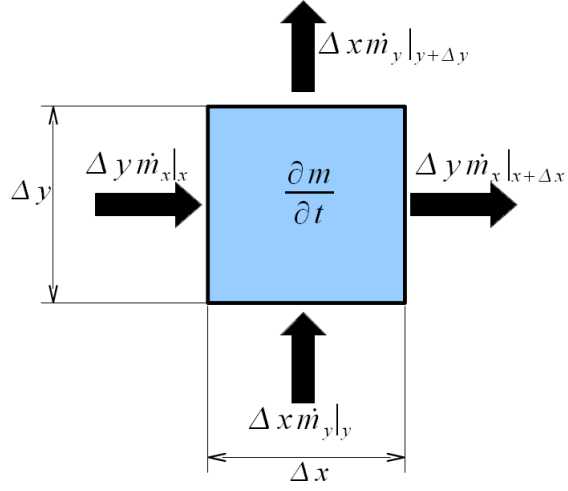
$$\sum_{i=1}^n \alpha_i = 1 \quad (5.7)$$

Dado que la masa del elemento infinitesimal puede variar con el tiempo y ésta no se crea en el elemento, se tiene:

$$\left\{ \begin{array}{c} \text{Razón de} \\ \text{incremento de} \\ \text{la masa} \end{array} \right\} = \left\{ \begin{array}{c} \text{Flujo de} \\ \text{entrada de} \\ \text{la masa} \end{array} \right\} - \left\{ \begin{array}{c} \text{Flujo de} \\ \text{salida de} \\ \text{la masa} \end{array} \right\}$$

En la Figura 8 se llama \dot{m} al flujo neto de masa que entra o sale del elemento, el cual combina los flujos convectivos y moleculares a través de la frontera del volumen, \dot{m}_i se

Figura 8: Balance de masa en el elemento



define como el flujo combinado de masa de la especie i , donde

$$\dot{m}_i = \rho_i \vec{A} \cdot \vec{V} + \vec{J}_i \cdot \vec{A} \quad (5.8)$$

definiendo el flujo molecular para mezclas binarias como

$$\vec{J}_i = -\rho D_{AB} \nabla \alpha_i \quad (5.9)$$

La ecuación de conservación de la masa para cada especie i se puede escribir como

$$\frac{\partial m_i}{\partial t} = (\dot{m}_{ix}|_x + \dot{m}_{iy}|_y - \dot{m}_{iy}|_{y+\Delta y} - \dot{m}_{ix}|_{x+\Delta x}) \quad (5.10)$$

de acuerdo con la serie de expansión de Taylor

$$\dot{m}_{ix}|_{x+\Delta x} = \dot{m}_{ix} + \frac{\partial \dot{m}_{ix}|_x}{\partial x} \Delta x + \frac{\partial^2 \dot{m}_{ix}|_x}{\partial x^2} \frac{(\Delta x)^2}{2} + \dots \quad (5.11)$$

despreciando los términos de orden superior a 1

$$\dot{m}_{ix} - \dot{m}_{ix}|_{x+\Delta x} = -\frac{\partial \dot{m}_{ix}|_x}{\partial x} \Delta x \quad (5.12)$$

y sumando la contribución de cada componente x y y se obtiene

$$\frac{\partial m}{\partial t} = -\frac{\partial \dot{m}_x}{\partial x} \Delta x - \frac{\partial \dot{m}_y}{\partial y} \Delta y \quad (5.13)$$

Definiendo el flujo másico por unidad de área como \vec{n} se puede reescribir la Ecuación 5.13 de la siguiente forma:

$$\frac{\partial m}{\partial t} = -\frac{\partial \vec{n}_x}{\partial x} \Delta x \Delta y - \frac{\partial \vec{n}_y}{\partial y} \Delta x \Delta y \quad (5.14)$$

y si $m_i = \rho_i \Delta x \Delta y$, dividiendo en $\Delta x \Delta y$ queda

$$\frac{\partial \rho_i}{\partial t} = -\left(\frac{\partial \vec{n}_{xi}}{\partial x} + \frac{\partial \vec{n}_{yi}}{\partial y} \right) \quad (5.15)$$

Escribiendo la Ecuación 5.15 en notación vectorial se tiene:

$$\frac{\partial \rho_i}{\partial t} = -\nabla \cdot \vec{n}_i \quad (5.16)$$

donde $\vec{n}_i = \rho_i \vec{V} + \vec{J}_i$, por lo tanto

$$\frac{\partial \rho_i}{\partial t} = -\left(\nabla \cdot \rho_i \vec{V} \right) - \left(\nabla \cdot \vec{J}_i \right) \quad (5.17)$$

Sumando cada una de las ecuaciones representadas en la Ecuación 5.17 para cada especie y usando la definición de densidad total de la mezcla, Ecuación 5.6, se deduce que

$$\frac{\partial \rho}{\partial t} = -\left(\nabla \cdot \rho \vec{V} \right) \quad (5.18)$$

dado que son complementarias, la sumatoria de las difusiones de cada especie es cero y se obtiene la ecuación de conservación de la masa en su forma conservativa.

Sabiendo que la derivada sustancial esta definida por Anderson [21] como

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \vec{V} \cdot \nabla \phi \quad (5.19)$$

teniendo en cuenta la identidad vectorial para la divergencia de un vector por un escalar

(Fuentes[22]):

$$\nabla \cdot (\rho \vec{V}) = (\rho \nabla \cdot \vec{V}) + (\vec{V} \cdot \nabla \rho) \quad (5.20)$$

y reemplazando la Ecuación 5.20 en la Ecuación 5.18 se tiene

$$\frac{\partial \rho_i}{\partial t} = - \left[(\rho_i \nabla \cdot \vec{V}) + (\vec{V} \cdot \nabla \rho_i) - (\nabla \cdot \vec{J}_i) \right] \quad (5.21)$$

reordenando los términos

$$\frac{\partial \rho_i}{\partial t} + (\vec{V} \cdot \nabla \rho_i) = - (\rho_i \nabla \cdot \vec{V}) - (\nabla \cdot \vec{J}_i) \quad (5.22)$$

y empleando la definición de derivada sustancial, Ecuación 5.19, se puede escribir la Ecuación 5.22 como:

$$\frac{D \rho_i}{Dt} = - (\rho_i \nabla \cdot \vec{V}) - (\nabla \cdot \vec{J}_i) \quad (5.23)$$

que representa la forma no conservativa de la ecuación de continuidad para una mezcla de especies. Para una mezcla binaria la Ecuación 5.22 se puede expresar como

$$\frac{\partial \rho_A}{\partial t} + (\vec{V} \cdot \nabla \rho_A) = - (\rho_A \nabla \cdot \vec{V}) + (\nabla \cdot \rho \nabla D_{AB} \alpha_A) \quad (5.24)$$

ó

$$\frac{\partial \rho_B}{\partial t} + (\vec{V} \cdot \nabla \rho_B) = - (\rho_B \nabla \cdot \vec{V}) + (\nabla \cdot \rho \nabla D_{AB} \alpha_B) \quad (5.25)$$

escritas en la forma conservativa de la ecuación de continuidad para cada especie química presente en la mezcla, solo se puede usar una ecuación, dado que si se suman las ecuaciones para las dos especies se obtiene[22]

$$\frac{\partial \rho}{\partial t} + (\vec{V} \cdot \nabla \rho) = - (\rho \nabla \cdot \vec{V}) \quad (5.26)$$

la cual es la ecuación de conservación de la masa total en su forma no conservativa. Para fluidos incompresibles se tiene que $\frac{D \rho}{Dt} = 0$, conocida como la condición de densidad

constante, debido a esto se puede escribir (Fuentes [22])

$$\nabla \cdot \vec{V} = 0 \quad (5.27)$$

por lo tanto las ecuaciones de conservación de las especies en una mezcla binaria de fluidos incompresibles se pueden escribir como

$$\frac{\partial \rho_A}{\partial t} + (\vec{V} \cdot \nabla \rho_A) = (\nabla \cdot \rho D_{AB} \nabla \alpha_A) \quad (5.28)$$

y

$$\frac{\partial \rho_B}{\partial t} + (\vec{V} \cdot \nabla \rho_B) = (\nabla \cdot \rho D_{AB} \nabla \alpha_B) \quad (5.29)$$

si estas ecuaciones se dividen por la densidad total, se obtienen las ecuaciones de la forma:

$$\frac{\partial \alpha_A}{\partial t} + (\vec{V} \cdot \nabla \alpha_A) = (\nabla \cdot D_{AB} \nabla \alpha_A) \quad (5.30)$$

y

$$\frac{\partial \alpha_B}{\partial t} + (\vec{V} \cdot \nabla \alpha_B) = (\nabla \cdot D_{AB} \nabla \alpha_B) \quad (5.31)$$

5.1.3. Ecuación de conservación de la cantidad de movimiento

En el desarrollo de la ecuación de conservación de la cantidad de movimiento se parte de un elemento de fluido que se mueve con el flujo para encontrar la forma no conservativa de dicha ecuación. Se aplica la segunda ley de Newton para un elemento que se mueve con el flujo, $\vec{F} = m\vec{a}$, esta ecuación establece que la fuerza neta actuando sobre un elemento de fluido es igual a su masa multiplicada por la aceleración. Dado que esta es una relación vectorial, se cuenta, para dos dimensiones, con dos relaciones escalares, una en cada eje coordenado x y y ,

$$\vec{F}_x = m\vec{a}_x \quad (5.32)$$

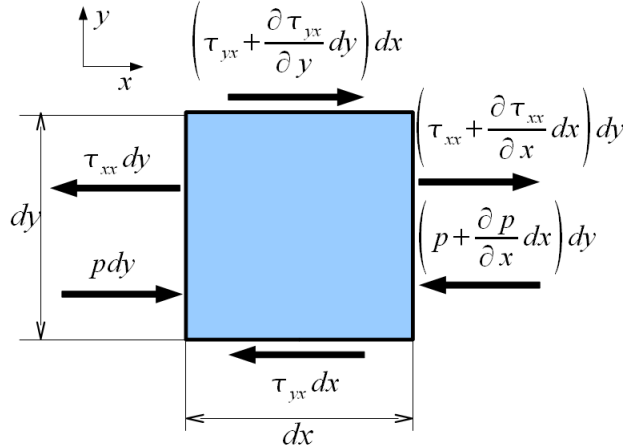
donde F_x y a_x son las componentes escalares en dirección x de la fuerza y la aceleración respectivamente.

Un elemento de fluido en movimiento se encuentra bajo el efecto de dos tipos de fuerzas, las de cuerpo, que actúan directamente sobre el volumen de masa del elemento del fluido (campos externos), y las de superficie, que actúan sobre las superficies del elemento (distribución de presión y distribución de esfuerzos).

Se define una fuerza de cuerpo por unidad de masa que actúa sobre el elemento como \vec{f} , siendo f_x su componente en el eje x. Para un elemento de fluido de dimensiones Δx Δy , con profundidad unitaria, el volumen del fluido es $\Delta x \Delta y$, de forma que:

$$\left\{ \begin{array}{l} \text{Fuerzas de cuerpo sobre} \\ \text{el elemento que actúan} \\ \text{en dirección x} \end{array} \right\} = \rho f_x (\Delta x \Delta y) \quad (5.33)$$

Figura 9: Balance de fuerzas en dirección x



Realizando un balance de fuerzas superficiales en dirección x en el elemento de fluido mostrado en Figura 9

$$\left\{ \begin{array}{l} \text{Fuerzas de superficie} \\ \text{sobre el elemento} \\ \text{actuando en dirección x} \end{array} \right\} = C \quad (5.34)$$

$$C = \left[p - \left(p + \frac{\partial p}{\partial x} dx \right) \right] \Delta y + \left[\left(\tau_{xx} + \frac{\partial \tau_{xx}}{\partial x} dx \right) - \tau_{xx} \right] \Delta y + \left[\left(\tau_{yx} + \frac{\partial \tau_{yx}}{\partial x} dx \right) - \tau_{yx} \right] \Delta x \quad (5.35)$$

Los esfuerzos están relacionados con la deformación del fluido con el tiempo, el esfuerzo cortante se denota por τ_{yx} y está relacionado con el cambio de forma del elemento, en tanto que el esfuerzo normal τ_{xx} está relacionado con el cambio de volumen del elemento. Por lo tanto los esfuerzos cortante y normal dependen de los gradientes de velocidad del fluido.

Empleando las definiciones dadas en la Ecuación 5.33 se deduce

$$F_x = \left[p - \left(p + \frac{\partial p}{\partial x} dx \right) \right] \Delta y + \left[\left(\tau_{xx} + \frac{\partial \tau_{xx}}{\partial x} dx \right) - \tau_{xx} \right] \Delta y + \left[\left(\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy \right) - \tau_{yx} \right] \Delta x + \rho f_x \Delta x \Delta y \quad (5.36)$$

Sumando y cancelando los términos iguales se obtiene

$$F_x = \left[-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} \right] \Delta y \Delta x + \rho f_x \Delta x \Delta y \quad (5.37)$$

Debido a que la masa de un elemento de dimensiones $\Delta x \Delta y$ está dada por

$$m = \rho \Delta x \Delta y \quad (5.38)$$

y a que la aceleración es el cambio de la velocidad en el tiempo, a_x es el cambio de u en el tiempo, además, considerando un elemento que se mueve con el fluido, el cambio de velocidad se define por la derivada sustancial [22], Ecuación 5.19, entonces

$$a_x = \frac{Du}{Dt} \quad (5.39)$$

combinando la Ecuación 5.32 con la Ecuación 5.37, Ecuación 5.38 y Ecuación 5.39, y dividiendo en $\Delta x \Delta y$ se obtiene

$$\rho \frac{Du}{Dt} = \left[-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} \right] + \rho f_x \quad (5.40)$$

de forma similar la componente en y

$$\rho \frac{Dv}{Dt} = \left[-\frac{\partial p}{\partial y} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} \right] + \rho f_y \quad (5.41)$$

En notación vectorial

$$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \nabla \cdot \tau + \rho \vec{f} \quad (5.42)$$

Conocida como la ecuación de conservación de la cantidad de movimiento en forma no conservativa, para obtener la forma conservativa se obtiene la definición de derivada sustancial dada por [21], entonces:

$$\rho \frac{Du}{Dt} = \rho \frac{\partial u}{\partial t} + \rho \vec{V} \cdot \nabla u \quad (5.43)$$

si expandimos $\frac{\partial \rho u}{\partial t} = \rho \frac{\partial u}{\partial t} + u \frac{\partial \rho}{\partial t}$ y reagrupando

$$\rho \frac{\partial u}{\partial t} = \frac{\partial \rho u}{\partial t} - u \frac{\partial \rho}{\partial t} \quad (5.44)$$

usando la identidad para la divergencia de un escalar por un vector

$$\nabla \cdot (\rho u \vec{V}) = u \nabla \cdot (\rho \vec{V}) + (\rho \vec{V}) \cdot \nabla u$$

ó

$$(\rho \vec{V}) \cdot \nabla u = \nabla \cdot (\rho u \vec{V}) - u \nabla \cdot (\rho \vec{V}) \quad (5.45)$$

sustituyendo la Ecuación 5.45 y la Ecuación 5.44 en la Ecuación 5.43 se obtiene

$$\begin{aligned} \rho \frac{Du}{Dt} &= \frac{\partial \rho u}{\partial t} - u \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u \vec{V}) - u \nabla \cdot (\rho \vec{V}) \\ &= \frac{\partial \rho u}{\partial t} - u \left[\frac{\partial \rho}{\partial t} - \nabla \cdot (\rho \vec{V}) \right] + \nabla \cdot (\rho u \vec{V}) \end{aligned} \quad (5.46)$$

donde la expresión $\left[\frac{\partial \rho}{\partial t} - \nabla \cdot (\rho \vec{V}) \right]$ corresponde a la ecuación de conservación de la masa y es igual a cero, por lo tanto se puede reescribir la Ecuación 5.42 en forma conservativa como:

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \vec{V}) = -\nabla p + \nabla \cdot \tau + \rho \vec{f} \quad (5.47)$$

La ecuación constitutiva de un fluido newtoniano conocida como ley de Stokes plantea

$$\boldsymbol{\tau} = 2\mu \left[\mathbf{D} - \frac{1}{3} \text{Tr}(\mathbf{D}) \cdot \mathbf{I} \right] \quad (5.48)$$

siendo \mathbf{I} y \mathbf{D} respectivamente la matriz identidad y la matriz de velocidad de deformación, esta última es la parte simétrica de la matriz de gradiente de velocidad $\nabla \vec{V}$:

$$\mathbf{D} = \text{Sim}(\nabla \vec{V}) = \frac{1}{2} \left[\nabla \vec{V} + (\nabla \vec{V})^T \right] \quad (5.49)$$

cumpléndose que la traza de la matriz \mathbf{D} :

$$\text{Tr}(\mathbf{D}) = \nabla \cdot \vec{V} \quad (5.50)$$

entonces la Ecuación 5.47 se puede reescribir

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \vec{V}) = -\nabla p + \nabla \cdot 2\mu \mathbf{D} - \frac{2}{3} \left[\mu (\nabla \cdot \vec{V}) \right] + \rho \vec{f} \quad (5.51)$$

pero debido a que la divergencia de la velocidad es cero y que la matriz del gradiente de velocidad $\nabla \vec{V}$ es simétrica para fluidos incompresibles, [22] y asumiendo viscosidad constante, la Ecuación 5.51 se puede reescribir como

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \vec{V}) = -\nabla p + \mu \nabla^2 \vec{V} + \rho \vec{f} \quad (5.52)$$

5.1.4. Ecuación de conservación de la energía

Para desarrollar esta ecuación se aplica la primera ley de la termodinámica, según la cual la energía se conserva, para esto se usa el modelo de flujo de un elemento infinitesimal de fluido que se mueve con el flujo para el cual la primera ley establece [22]

$$\underbrace{A}_{\text{Cambio de energía dentro del elemento}} = \underbrace{B}_{\text{Flujo neto de calor dentro del elemento}} + \underbrace{C}_{\text{Trabajo realizado sobre el elemento debido a las fuerzas}}$$

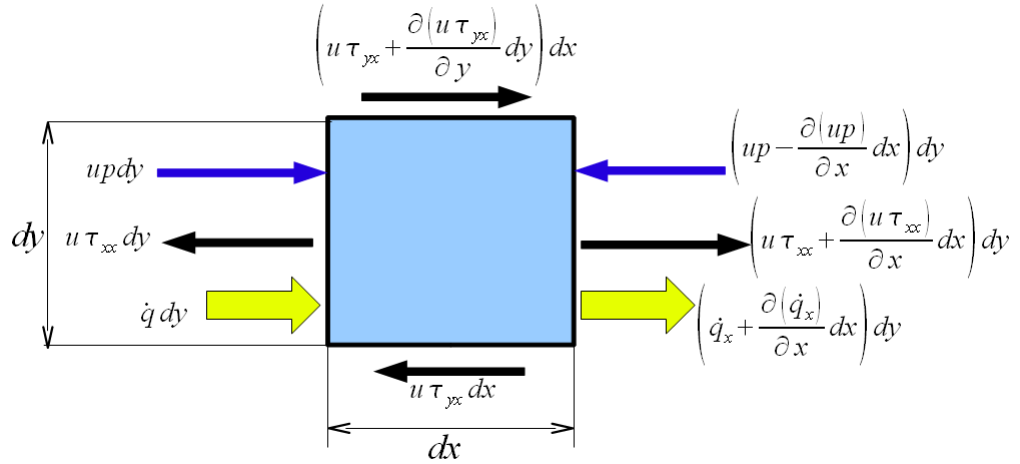
El trabajo realizado por las fuerzas de cuerpo que actúan sobre el elemento de dimensiones $dxdy$ con espesor unitario, que se mueve con el flujo con velocidad \vec{V} está dado por

$$\rho \vec{f} \cdot \vec{V} (dxdy) \quad (5.53)$$

El trabajo hecho por las fuerzas de superficie es mostrado en la Figura 10, donde las fuerzas en dirección x positiva hacen trabajo positivo y las que tienen dirección negativa en x , hacen trabajo negativo. Según la Figura 10 el trabajo neto por la presión en dirección x es:

$$\left[up - \left(up + \frac{\partial (up)}{\partial x} dx \right) \right] dy = -\frac{\partial (up)}{\partial x} dxdy \quad (5.54)$$

Figura 10: Trabajo de las fuerzas de superficie x



Similarmente, para el esfuerzo cortante en dirección x se tiene

$$\left[-u \tau_{yx} + \left(u \tau_{yx} + \frac{\partial (u \tau_{yx})}{\partial y} dy \right) \right] dx = \frac{\partial (u \tau_{yx})}{\partial y} dxdy \quad (5.55)$$

y para el esfuerzo normal en dirección x

$$\left[-u \tau_{xx} + \left(u \tau_{xx} + \frac{\partial (u \tau_{xx})}{\partial x} dx \right) \right] dy = \frac{\partial (u \tau_{xx})}{\partial x} dxdy \quad (5.56)$$

El trabajo neto de las fuerzas de superficie en x sobre el elemento de fluido en movimiento es

$$\left[-\frac{\partial (up)}{\partial x} + \frac{\partial (u\tau_{xx})}{\partial x} + \frac{\partial (u\tau_{yx})}{\partial y} \right] dxdy \quad (5.57)$$

si se consideran las fuerzas en dirección y se obtiene una expresión similar

$$\left[-\frac{\partial (vp)}{\partial y} + \frac{\partial (v\tau_{xy})}{\partial x} + \frac{\partial (v\tau_{yy})}{\partial y} \right] dxdy \quad (5.58)$$

sumando el trabajo de las fuerzas de cuerpo y de superficie se puede escribir

$$C = \left[-\left(\frac{\partial (up)}{\partial x} + \frac{\partial (vp)}{\partial y} \right) + \frac{\partial (u\tau_{xx})}{\partial x} + \frac{\partial (u\tau_{yx})}{\partial y} + \frac{\partial (v\tau_{xy})}{\partial x} + \frac{\partial (v\tau_{yy})}{\partial y} \right] dxdy + \rho \vec{f} \cdot \vec{V} (dxdy) \quad (5.59)$$

El flujo neto de calor dentro del elemento se define como la suma de los aportes del calentamiento volumétrico tales como la absorción o emisión de radiación y la transferencia de calor a través de las superficies debido a gradientes de temperatura. Debido a que el fenómeno estudiado se supone isotérmico, no existen gradientes de temperatura y por lo tanto no existe flujo de calor por conducción, así como tampoco existe calentamiento volumétrico, entonces

$$B = 0 \quad (5.60)$$

Un elemento de fluido en movimiento tiene energía cinética, energía interna y energía potencial, de forma tal que la suma de las tres representa la energía total. Debido a que se está analizando un elemento de fluido que se mueve con el flujo, el cambio de energía por unidad de masa esta determinado por la derivada sustancial de la energía interna e mas la energía cinética $V^2/2$. Con la masa del elemento de fluido dada por $\rho dxdy$ se tiene

$$A = \rho \frac{D}{Dt} \left(e + \frac{V^2}{2} + z \right) dxdy \quad (5.61)$$

de manera que si se lleva a la forma $A = B + C$ y se divide en $dx dy$ se obtiene

$$\rho \frac{D}{Dt} \left(e + \frac{V^2}{2} + z \right) = - \left(\frac{\partial (up)}{\partial x} + \frac{\partial (vp)}{\partial y} \right) + \left[\frac{\partial (u\tau_{xx})}{\partial x} + \frac{\partial (u\tau_{yx})}{\partial y} + \frac{\partial (v\tau_{xy})}{\partial x} + \frac{\partial (v\tau_{yy})}{\partial y} \right] + \rho \vec{f} \cdot \vec{V} \quad (5.62)$$

en forma vectorial

$$\rho \frac{D}{Dt} \left(e + \frac{V^2}{2} + z \right) = -\nabla \cdot (p\vec{V}) + \nabla \cdot (\tau \cdot \vec{V}) + \rho \vec{f} \cdot \vec{V} \quad (5.63)$$

La Ecuación 5.63 es la ecuación de conservación de la energía en función de la energía interna en su forma conservativa, dado que la entalpía esta definida como $h = e + \frac{p}{\rho}$ también se puede escribir esta ecuación en función de ésta así:

$$\rho \frac{D}{Dt} \left(h + \frac{V^2}{2} + z \right) = -\frac{\partial p}{\partial t} + \tau \cdot \nabla \vec{V} \quad (5.64)$$

siendo h la entalpía por unidad de masa, con

$$dh = \left(\frac{\partial h}{\partial T} \right)_p dT + \left(\frac{\partial h}{\partial p} \right)_T dp = C_p dT + \left[v - T \left(\frac{\partial v}{\partial T} \right)_p \right] dp \quad (5.65)$$

integrando temperaturas y presiones entre un estado de referencia a p^o T^o y la presión p y temperatura T se obtiene

$$h = h^o + \int_{T^o}^T C_p dT + \int_{p^o}^p \left[v - T \left(\frac{\partial v}{\partial T} \right)_p \right] dp \quad (5.66)$$

para fluidos incompresibles la segunda integral de la Ecuación 5.66 es igual a $\left(\frac{1}{\rho} \right) (p - p^o)$, ver [23], de modo que la entalpía está dada por:

$$h = h^o + \int_{T^o}^T C_p dT + \left(\frac{1}{\rho} \right) (p - p^o) \quad (5.67)$$

para obtener la forma conservativa de estas ecuaciones se parte de la definición de

derivada material

$$\rho \frac{D\left(e + \frac{V^2}{2} + z\right)}{Dt} = \rho \frac{\partial\left(e + \frac{V^2}{2} + z\right)}{\partial t} + \rho \vec{V} \cdot \nabla \left(e + \frac{V^2}{2} + z\right) \quad (5.68)$$

si se expande $\frac{\partial \rho \left(e + \frac{V^2}{2} + z\right)}{\partial t} = \rho \frac{\partial \left(e + \frac{V^2}{2} + z\right)}{\partial t} + \left(e + \frac{V^2}{2} + z\right) \frac{\partial \rho}{\partial t}$ y se reagrupa

$$\rho \frac{\partial \left(e + \frac{V^2}{2} + z\right)}{\partial t} = \frac{\partial \rho \left(e + \frac{V^2}{2} + z\right)}{\partial t} - \left(e + \frac{V^2}{2} + z\right) \frac{\partial \rho}{\partial t} \quad (5.69)$$

usando la identidad para la divergencia de un escalar por un vector

$$\nabla \cdot \left(\rho \left(e + \frac{V^2}{2} + z \right) \vec{V} \right) = \left(e + \frac{V^2}{2} + z \right) \nabla \cdot (\rho \vec{V}) + (\rho \vec{V}) \cdot \nabla \left(e + \frac{V^2}{2} + z \right)$$

ó

$$(\rho \vec{V}) \cdot \nabla \left(e + \frac{V^2}{2} + z \right) = \nabla \cdot \left(\rho \left(e + \frac{V^2}{2} + z \right) \vec{V} \right) - \left(e + \frac{V^2}{2} + z \right) \nabla \cdot (\rho \vec{V}) \quad (5.70)$$

sustituyendo la Ecuación 5.70 y la Ecuación 5.69 en la Ecuación 5.68, se obtiene

$$\begin{aligned} \rho \frac{D\left(e + \frac{V^2}{2} + z\right)}{Dt} &= \frac{\partial \rho \left(e + \frac{V^2}{2} + z\right)}{\partial t} - \left(e + \frac{V^2}{2} + z\right) \frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \left(e + \frac{V^2}{2} + z \right) \vec{V} \right) \\ &\quad - \left(e + \frac{V^2}{2} + z \right) \nabla \cdot (\rho \vec{V}) \end{aligned} \quad (5.71)$$

agrupando

$$\begin{aligned} \rho \frac{D\left(e + \frac{V^2}{2} + z\right)}{Dt} &= \frac{\partial \rho \left(e + \frac{V^2}{2} + z\right)}{\partial t} - \left(e + \frac{V^2}{2} + z\right) \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) \right] \\ &\quad + \nabla \cdot \left(\rho \left(e + \frac{V^2}{2} + z \right) \vec{V} \right) \end{aligned} \quad (5.72)$$

y aplicando la ecuación de continuidad

$$\rho \frac{D\left(e + \frac{V^2}{2} + z\right)}{Dt} = \frac{\partial \rho \left(e + \frac{V^2}{2} + z\right)}{\partial t} + \nabla \cdot \left(\rho \left(e + \frac{V^2}{2} + z \right) \vec{V} \right) \quad (5.73)$$

por lo tanto la Ecuación 5.63 se puede escribir en su forma conservativa como

$$\frac{\partial \rho \left(e + \frac{V^2}{2} + z \right)}{\partial t} + \nabla \cdot \left(\rho \left(e + \frac{V^2}{2} + z \right) \vec{V} \right) = -\nabla \cdot (p \vec{V}) + \nabla \cdot (\tau \cdot \vec{V}) + \rho \vec{f} \cdot \vec{V} \quad (5.74)$$

de forma análoga para la Ecuación 5.64 se puede escribir su forma conservativa como

$$\frac{\partial \rho \left(h + \frac{V^2}{2} + z \right)}{\partial t} + \nabla \cdot \left(\rho \left(h + \frac{V^2}{2} + z \right) \vec{V} \right) = -\frac{\partial p}{\partial t} + \tau \cdot \nabla \vec{V} \quad (5.75)$$

5.1.5. Ecuaciones del modelo turbulento

La mayor parte de los flujos en la naturaleza son turbulentos, la turbulencia puede ser descrita como un estado de inestabilidad continua en el flujo, donde es posible separar las fluctuaciones de las propiedades medias del fluido, está caracterizada por irregularidad en el flujo, lo cual incrementa la difusividad y la disipación de energía. Los flujos turbulentos siempre son tridimensionales y dependientes del tiempo aun cuando las condiciones de frontera no cambien en el tiempo. El rango de escalas de dichos flujos es bastante amplio, desde las mas pequeñas, caracterizadas por las microescalas de Kolmogorov, hasta las mas grandes, del tamaño de la geometría estudiada.

Existen varios enfoques posibles para simular flujos turbulentos[19]:

1. Simulación numérica directa (DNS por sus siglas en inglés): en este enfoque se integra numéricamente las ecuaciones gobernantes sobre todo el rango de escalas. Para este enfoque se requieren vastos recursos computacionales debido a la resolución de la malla y el tamaño del paso del tiempo, lo cual lo hace inadecuado en el ámbito ingenieril.
2. Simulación de grandes remolinos (LES): en este enfoque se aplica un filtro espacial para separar las diferentes escalas del campo de flujo turbulento. Las grandes estructuras de turbulencia que son resueltas mediante el método numérico en una malla dada son conocidas como las escalas de supermalla. Se modela la influencia de otras escalas (submalla) al comportamiento de la supermalla. Este enfoque se basa en el hecho de que las escalas pequeñas de turbulencia son mas homogéneas e isotrópicas, por lo tanto mas fáciles de modelar, aproximándose así al enfoque DNS.
3. Enfoque de simulación estadístico: este enfoque funciona separando el valor local

de la variable en valor medio y las fluctuaciones alrededor de la media, esto hace posible derivar las ecuaciones para las propiedades medias. Es el enfoque más usado actualmente en la investigación y la industria. La selección del método de promediado depende de las características del flujo turbulento:

- Promediado del tiempo en un punto fijo del espacio, usado para turbulencia estacionaria.
- Promediado en el espacio en un momento fijo en el tiempo, en el caso de turbulencia homogénea
- Promediado ensamblado de una serie de experimentos idénticos. Este es el caso más general de promediado.

Estos métodos promediados se encuentran en dos versiones, no ponderados (Reynolds) y ponderados. En este estudio se usó la técnica de promediado de Reynolds.

5.1.5.1. Modelo RANS $k - \epsilon$

Este modelo de turbulencia se deduce a partir de las ecuaciones de Navier-Stokes promediadas en el tiempo con ayuda de la descomposición de Reynolds (Reynolds averaged Navier Stokes-RANS). Debido a esta descomposición aplicada a la velocidad en las ecuaciones de conservación, aparecen más variables, llamadas los esfuerzos de Reynolds, y dada la naturaleza no lineal de las ecuaciones, al plantear una ecuación de continuidad para los esfuerzos de Reynolds solo se obtienen más variables y no se ganan ecuaciones. La función de modelado de la turbulencia consiste en idear aproximaciones de las correlaciones desconocidas en términos de propiedades de flujo que se conocen de modo que exista un número suficiente de ecuaciones. Al realizar estas aproximaciones se cierra el sistema.

Descomposición de Reynolds Según la descomposición de Reynolds, una propiedad instantánea es calculada como el valor medio más un valor fluctuante

$$\vec{V}_i(x, y, t) = \overline{\vec{V}_i}(x, y) + \vec{V}'_i(x, y, t) \quad (5.76)$$

donde $V_i(x, y, t)$ es la velocidad instantánea, que es la suma de la velocidad media $\overline{\vec{V}_i}(x, y)$ y las fluctuaciones $\vec{V}'_i(x, y, t)$. Dado que en presencia de turbulencia no homogénea, la evaluación en el tiempo es la más adecuada para realizar el promedio, se

puede decir

$$\overline{\vec{V}}_i(x, y) = \lim_{t' \rightarrow \infty} \frac{1}{t'} \int_t^{t+t'} \vec{V}_i(x, y, t) dt \quad (5.77)$$

el promedio en el tiempo de la velocidad media es nuevamente la velocidad media

$$\overline{\overline{\vec{V}}_i(x, y)} = \lim_{t' \rightarrow \infty} \frac{1}{t'} \int_t^{t+t'} \overline{\vec{V}}_i(x, y) dt = \overline{\vec{V}}_i(x, y) \quad (5.78)$$

el valor promedio en el tiempo de la parte fluctuante de la velocidad es cero

$$\overline{\vec{V}'_i(x, y)} = \lim_{t' \rightarrow \infty} \frac{1}{t'} \int_t^{t+t'} \left(\vec{V}_i(x, y) - \overline{\vec{V}}_i(x, y) \right) dt = \overline{\vec{V}}_i(x, y) - \overline{\overline{\vec{V}}_i(x, y)} = 0 \quad (5.79)$$

Como no es posible realizar el límite cuando el tiempo t' tiende a infinito se escoge un valor de t' lo suficientemente grande con relación al máximo valor de duración de las fluctuaciones t_1 , es decir $t' \gg t_1$.

Para flujos que están influenciados por fluctuaciones lentas de la velocidad media en el tiempo pero que no se pueden considerar turbulencia, se tiene

$$\vec{V}_i(x, y, t) = \overline{\vec{V}}_i(x, y, t) + \vec{V}'_i(x, y, t) \quad (5.80)$$

y por lo tanto

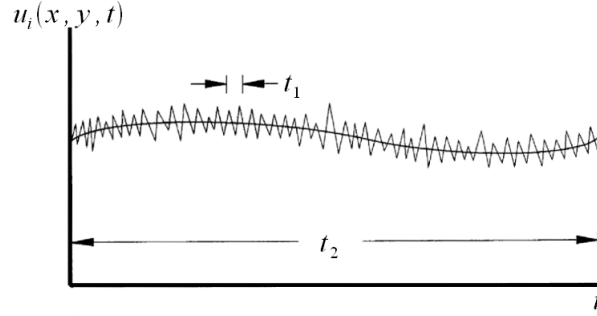
$$\overline{\vec{V}}_i(x, y, t) = \lim_{t' \rightarrow \infty} \frac{1}{t'} \int_t^{t+t'} \vec{V}_i(x, y, t) dt, \quad t_1 \lll t' \lll t_2 \quad (5.81)$$

donde t_2 es la escala de tiempo característica de las variaciones lentas que no se deben a la turbulencia, ver la Figura 11.

Ecuaciones de Navier-Stokes promediadas en el tiempo RANS Para la conservación de la masa se tiene que la Ecuación 5.26 y según la Ecuación 5.76 se puede escribir como

$$\frac{D\rho}{Dt} = - \left(\rho \nabla \cdot \left(\overline{\vec{V}} + \vec{V}' \right) \right) \quad (5.82)$$

Figura 11: Escalas de tiempo en turbulencia



según la Ecuación 5.79 el promedio en el tiempo de la componente fluctuante es 0, por lo tanto la Ecuación 5.82 queda

$$\frac{D\rho}{Dt} = - \left(\rho \nabla \cdot \vec{V} \right) \quad (5.83)$$

Para cada especie, Ecuación 5.24, se sabe que el flujo difusivo J es

$$J_i = \rho_i \left(\vec{V}_i - \vec{V} \right) = -\rho D_{AB} \nabla \alpha_i \quad (5.84)$$

donde \vec{V}_i es la velocidad de la especie i y \vec{V} es la velocidad media del flujo, mediante la descomposición de Reynolds para la velocidad se puede escribir como

$$J_i = \rho_i \left(\vec{V}_i - \vec{V} \right) + \rho_i \left(\vec{V}_i' - \vec{V}' \right) = -\rho D_{ABefect} \nabla \alpha_i \quad (5.85)$$

donde $D_{ABefect} = D_{ABmolecular} + D_{ABturbulenta}$, para cada especie la Ecuación 5.24 entonces tiene la forma

$$\frac{D\alpha_i}{Dt} = - \left(\alpha_i \nabla \cdot \vec{V} \right) + (\nabla \cdot D_{ABefect} \nabla \alpha_i) \quad (5.86)$$

para fluidos incompresibles y usando la definición de derivada sustancial queda:

$$\frac{\partial \alpha_i}{\partial t} + \left(\vec{V} \cdot \nabla \alpha_i \right) = (\nabla \cdot D_{ABefect} \nabla \alpha_i) \quad (5.87)$$

en el caso de la conservación de la cantidad de movimiento, la Ecuación 5.47 en presencia

solamente del campo gravitatorio, se puede escribir en forma conservativa

$$\rho \frac{\partial \vec{V}}{\partial t} + \rho \nabla \cdot (\vec{V} \vec{V}) = -\nabla p + \mu \nabla^2 \vec{V} + \rho \vec{g} \quad (5.88)$$

sustituyendo la Ecuación 5.76 en la Ecuación 5.88 y dividiendo por ρ se obtiene

$$\frac{\partial (\vec{V} + \vec{V}')}{\partial t} + \nabla \cdot ((\vec{V} + \vec{V}') (\vec{V} + \vec{V}')) = -\frac{\nabla (\bar{p} + p')}{\rho} + \nu \nabla^2 (\vec{V} + \vec{V}') + \vec{g} \quad (5.89)$$

promediando se tiene

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V} \vec{V} + \vec{V}' \vec{V}') = -\frac{\nabla (\bar{p})}{\rho} + \nu \nabla^2 \vec{V} + \vec{g} \quad (5.90)$$

reorganizando

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V} \vec{V}) = -\frac{\nabla (\bar{p})}{\rho} + \nu \nabla^2 \vec{V} + \vec{g} - \frac{1}{\rho} \nabla \cdot \rho \vec{V}' \vec{V}' \quad (5.91)$$

reescribiendo

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V} \vec{V}) = -\frac{\nabla (\bar{p})}{\rho} + \nu \nabla^2 \vec{V} + \vec{g} - \frac{1}{\rho} \nabla \cdot \tau^R \quad (5.92)$$

donde el término τ^R es conocido como el tensor de esfuerzos de Reynolds, en 3 dimensiones se tiene

$$\tau^R = -\rho (\vec{V}' \vec{V}') = \begin{pmatrix} \rho \overline{u' u'} & \rho \overline{u' v'} & \rho \overline{u' w'} \\ \rho \overline{v' u'} & \rho \overline{v' v'} & \rho \overline{v' w'} \\ \rho \overline{w' u'} & \rho \overline{w' v'} & \rho \overline{w' w'} \end{pmatrix} \quad (5.93)$$

Aprximación de Boussinesq El tensor de esfuerzos de Reynolds en tres dimensiones forma un tensor simétrico de segundo orden que contiene 6 términos independientes, y analizando el conjunto de ecuaciones deducido y el problema de estudio en tres dimensiones se puede contar el numero de incógnitas, a saber, 3 componentes de velocidad

(u, v, w) , la concentración (α) , la presión (p) , y seis componentes del tensor de esfuerzos de Reynolds (τ^R) ; ya que solo se tienen 5 ecuaciones contando la ecuación de concentraciones, el sistema no es cerrado, entonces serían necesarias 6 relaciones más para cerrar el sistema, para lo cual podríamos derivar las ecuaciones gobernantes del término esfuerzos de Reynolds, pero es más sencillo modelarlo. Una forma común de modelar este término es usando la hipótesis de Bossinesq para relacionar los esfuerzos de Reynolds a los gradientes de la velocidad media:

$$\tau^R = -\rho \left(\overline{\vec{V}'\vec{V}'} \right) = 2\mu_T \bar{\mathbf{D}}^R - \frac{2}{3}\rho k \mathbf{I} = \mu_T \left[\nabla \vec{V} + \left(\nabla \vec{V} \right)^T \right] - \frac{2}{3}\rho k \mathbf{I} \quad (5.94)$$

donde $\bar{\mathbf{D}}^R$ denota el tensor de razón de esfuerzos de Reynolds promediados, \mathbf{I} es la matriz identidad y μ_T es conocida como la viscosidad turbulenta o efectiva, k se conoce como la energía cinética turbulenta para la cual se tiene

$$k = \frac{1}{2} \overline{\vec{V}'\vec{V}'} \quad (5.95)$$

μ_T puede ser evaluada de muchas formas, la usada en este trabajo, la expresa como una función de la energía cinética turbulenta k y su razón de disipación ϵ , definiendo un modelo de turbulencia de dos ecuaciones, donde

$$\mu_T = \rho C_\mu \frac{k^2}{\epsilon} \quad (5.96)$$

donde C_μ es una constante.

5.1.6. Ecuación de energía cinética turbulenta k

Para la ecuación de energía cinética turbulenta se tiene, según [24]:

$$\frac{\partial k}{\partial t} + \nabla \cdot (\vec{V}k) - \nabla \cdot \frac{\nu_T}{\sigma_k} \nabla k = P_k - \epsilon \quad (5.97)$$

donde P_k es la razón de producción de k con

$$P_k = 2\nu_T \left| \nabla \vec{V} \right|^2$$

5.1.7. Ecuación de la Razón de disipación ϵ

En cuanto a la razón de disipación según [24]:

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\vec{V} \epsilon) - \nabla \cdot \frac{\nu_T}{\sigma_\epsilon} \nabla \epsilon = C_{1\epsilon} \frac{\epsilon}{k} P_k - C_{2\epsilon} \frac{\epsilon^2}{k} \quad (5.98)$$

y $C_{1\epsilon}, C_{2\epsilon}, \sigma_\epsilon, \sigma_k$ son coeficientes del modelo.

5.1.8. Funciones de pared

La física de la turbulencia en la vecindad de paredes impermeables sin deslizamiento es considerablemente diferente de otras partes del flujo. En este caso se hace necesario el uso de modelos apropiados en la región cercana a la pared, si se quisiera un tratamiento detallado, debería usarse el modelo de bajo Reynolds, pero para poder resolver bien los detalles cerca a la pared, sería necesaria una malla computacional muy fina en esa región.

Es posible compensar la existencia de la pared sin resolver la zona cercana a ella, esto a expensas de exactitud en la solución. Las funciones de pared representan un modelo simplificado de turbulencia, el cual remeda el comportamiento cercano a la pared de la velocidad \vec{V} , la energía cinética turbulenta k y la disipación ϵ . Estas funciones asumen que el flujo cerca a las paredes se comporta como una capa límite turbulenta totalmente desarrollada. Este modelo se usa para evitar las regiones con grandes gradientes cerca a las paredes y se acopla con el modelo de alto numero de Reynolds en las demás partes del dominio.

5.2. ECUACIONES DEL MODELO

Las ecuaciones que conforman el modelo empleado en este trabajo de investigación se puede resumir en la Tabla 1, estas son las ecuaciones para un flujo incompresible isotérmico completamente desarrollado de dos fluidos que se transportan por tubería:

Tabla 1: Ecuaciones del modelo

Ecuaciones	Forma
Conservación de la masa	$\nabla \cdot \vec{V} = 0$
Conservación de las especies químicas	$\frac{\partial \rho_A}{\partial t} + (\vec{V} \cdot \nabla \rho_A) = (\nabla \cdot \rho D_{AB} \nabla \alpha_A)$
Conservación de cantidad de movimiento	$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V} \vec{V}) = -\frac{\nabla(\bar{p})}{\rho} + \nu \nabla^2 \vec{V} + \vec{g} - \frac{1}{\rho} \nabla \cdot \tau^R$
Conservación de k	$\frac{\partial k}{\partial t} + \nabla \cdot (\vec{V} k) - \nabla \cdot \frac{\nu_T}{\sigma_k} \nabla k = P_k - \epsilon$
Conservación de ϵ	$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\vec{V} \epsilon) - \nabla \cdot \frac{\nu_T}{\sigma_\epsilon} \nabla \epsilon = C_{1\epsilon} \frac{\epsilon}{k} P_k - C_{2\epsilon} \frac{\epsilon^2}{k}$
Modelo turbulento	$P_k = 2\nu_T \nabla \vec{V} ^2, C_{1\epsilon}, C_{2\epsilon}, \sigma_\epsilon, \sigma_k$
Frontera de entrada	$\alpha_1 = 1, \vec{V} = (1, 0, 0)[m/s], p = \text{grad. cero}, k = \text{cte}, \epsilon = \text{cte.}$
Frontera de salida	$p = \text{cte}, \vec{V} = \text{grad. cero}, k = \text{grad. cero}, \epsilon = \text{grad. cero.}$
Paredes	$\vec{V} = 0, p = \text{grad. cero.}$

5.3. DISCRETIZACIÓN

El propósito de cualquier práctica de discretización es transformar una o mas ecuaciones diferenciales parciales en un sistema correspondiente de ecuaciones algebraicas. La solución de estos sistemas produce un conjunto de valores que corresponden a la solución de las ecuaciones originales en algunos lugares predeterminados en el espacio y tiempo, suministrando ciertas condiciones que se deben cumplir. El proceso de discretización se puede dividir en dos pasos: la discretización del dominio de la solución y la discretización de las ecuaciones.

Para la discretización del dominio se produce una descripción numérica en el dominio computacional, que incluye la información de los puntos en los cuales se va a hallar la solución y la descripción de la frontera. Esta descripción se realiza mediante la división del espacio en un número discreto de regiones, llamadas volúmenes de control o celdas, la cual generalmente es escrita por un programa generador de malla en un archivo donde las celdas son serialmente identificadas junto a la información de sus vértices. En el caso de simulaciones transitorias, el intervalo de tiempo considerado es también dividido en un número finito de pasos de tiempo. Por otro lado, la discretización de las ecuaciones nos permite una apropiada transformación de los términos de las ecuaciones gobernantes en expresiones algebraicas.

5.3.1. Dominios complejos

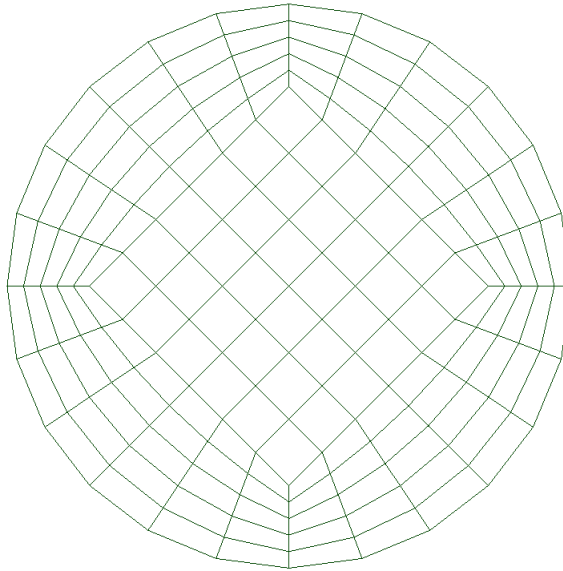
En esta investigación se estudia el modelado de una tubería de diámetro constante a lo largo de un trayecto. En el caso de presentarse variaciones en la geometría del conducto, se hace claro que una malla cartesiana no representaría bien el dominio debido a que las líneas constantes x y y podrían intersectarse con la frontera de forma arbitraria. Para afrontar este inconveniente al representar el dominio se cuenta con alternativas, las mallas estructuradas no cartesianas, las mallas curvas y las mallas no estructuradas usadas en elementos finitos.

En este trabajo se representa el dominio mediante el uso de una malla estructurada no cartesiana.

5.3.2. Malla estructurada

La malla estructurada está formada por la conexión de puntos adecuadamente escogidos acorde con la geometría a describir, llamados vértices por medio de líneas que no se corten en un dominio dado [25], dando como resultado elementos con un orden fijo, como los cuadrilateros mostrados en laFigura 12

Figura 12: Malla estructurada



Para realizar la discretización de volúmenes finitos es necesario definir un nodo y un volumen de control. Existe gran variedad de enfoques posibles para definirlos, los principales son centrado en los vértices y centrado en el elemento. En este trabajo se discretizan las ecuaciones usando el enfoque centrado en el elemento.

5.3.3. Discretización del dominio de la solución

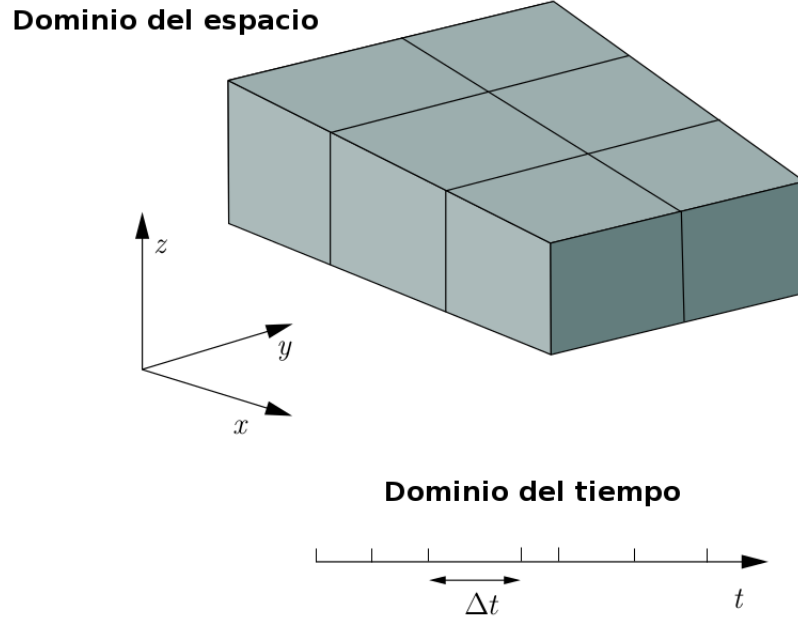
La discretización del dominio de la solución produce una malla en la cual se resuelve el sistema de ecuaciones establecido. En esta etapa se determinan los puntos en el espacio para los cuales se encuentra la solución, pero este procedimiento se debe dividir en dos partes, discretización temporal y discretización espacial.

En un problema transitorio, debido a que el tiempo es una coordenada parabólica, la solución se debe encontrar avanzando en el tiempo desde una condición inicial, entonces

para la discretización temporal es suficiente con definir el tamaño del paso del tiempo que será usado durante el cálculo.

Para la discretización espacial en el método de los volúmenes finitos usado en este estudio se requiere de la subdivisión del dominio en volúmenes de control. Estos volúmenes de control no se deben traslapar y deben llenar por completo el dominio. En este estudio todas las variables comparten el mismo volumen de control.

Figura 13: Discretización Espacio y tiempo



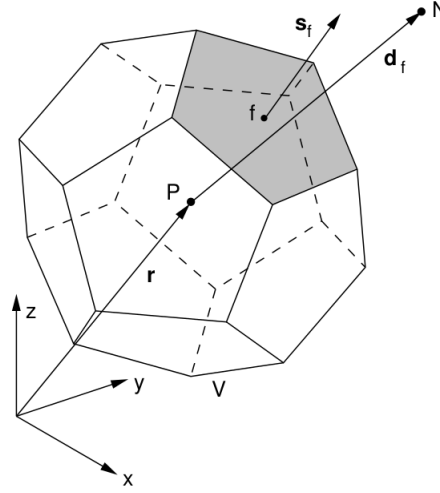
Fuente: Modificado de Rusche 2002 [26]

Un volumen de control típico se muestra en la Figura 14, donde el punto computacional P es ubicado en su centroide, el punto N en el centroide del vecino con el cual comparte la cara sombreada. de forma que

$$\int_{\Omega_P} (x - x_P) d\Omega = 0 \quad (5.99)$$

el volumen de control esta limitado por un grupo de caras planas y cada cara es compartida por solo un volumen de control vecino. Las caras de las celdas se pueden dividir en dos grupos, caras internas, las cuales son compartidas por dos volúmenes de control, y las caras de frontera, que coinciden con las fronteras del dominio. El vector de área \mathbf{S}_f se construye en cada cara de forma tal que apunta hacia afuera de la celda, es normal

Figura 14: Volumen de control



Fuente: Tomado de Jasak 2008[27].

a la cara y tiene una magnitud igual al área de la cara. los vectores de área de frontera apuntan hacia afuera del dominio computacional. f es el centro de la cara.

5.3.4. Discretización de las ecuaciones de transporte

Las ecuaciones de transporte pueden ser escritas en notación compacta para una sola variable ϕ , como

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot \vec{q} = S_\phi \quad (5.100)$$

donde $\vec{q} = \rho u_j \phi - \Gamma_{eff}^\phi \frac{\partial \phi}{\partial x_j}$, los significados para Γ_{eff}^ϕ y S_ϕ para cada ϕ se presentan en la Tabla 2.

Tabla 2: Valores para Γ_{eff}^ϕ y S_ϕ

ϕ	Γ_{eff}^ϕ	S_ϕ
1	0	0
α_i	$\rho(D + D_t)$	0
u_i	$\mu + \mu_t$	$-\frac{\partial p}{\partial x_i} + \rho f_i$
$h + \frac{V^2}{2}$	0	$-\frac{\partial p}{\partial t} + \tau \cdot \nabla \vec{V}$

Fuente: Modificado de Fuentes 2011[28]

para la forma conservativa se tiene en notación vectorial:

$$\underbrace{\frac{\partial \rho \phi}{\partial t}}_{\text{derivada temporal}} + \underbrace{\nabla \cdot \rho \vec{V} \phi}_{\text{término convectivo}} - \underbrace{\nabla \cdot (\rho \Gamma_{eff}^\phi \nabla \phi)}_{\text{término difusivo}} = \underbrace{S_\phi(\phi)}_{\text{término fuente}} \quad (5.101)$$

Donde \vec{V} es el vector velocidad y los valores de S_ϕ y Γ_{eff} se presentan en la Tabla 2. Esta es una ecuación de segundo orden, debido a que el término difusivo incluye la segunda derivada en el espacio. Para una buena exactitud, es necesario que el orden de discretización sea por lo menos igual o superior al orden de la ecuación que va a ser discretizada[19].

la exactitud del método de discretización depende de la variación asumida en el espacio y tiempo alrededor del punto P para la función $\phi = \phi(x, t)$. Si se quiere una exactitud de segundo orden, esta variación debe ser lineal en espacio y tiempo

$$\phi(x) = \phi_P + (x - x_P) \cdot (\nabla \phi)_P \quad (5.102)$$

$$\phi(t + \Delta t) = \phi^t + (\Delta t) \cdot \left(\frac{\partial \phi}{\partial t} \right)^t \quad (5.103)$$

donde $\phi_P = \phi(x_P)$ y $\phi^t = \phi(t)$. Ahora, para la expansión de Taylor alrededor del punto x , se tiene[19]:

$$\begin{aligned} \phi(x) = \phi_P + (x - x_P) \cdot (\nabla \phi)_P + \frac{1}{2}(x - x_P)^2 : (\nabla \nabla \phi)_P + \frac{1}{3!}(x - x_P)^3 :: (\nabla \nabla \nabla \phi)_P \\ + \dots + \frac{1}{n!}(x - x_P)^n \underbrace{\vdots}_n (\underbrace{\nabla \nabla \dots \nabla}_n \phi)_P + \end{aligned} \quad (5.104)$$

donde la expresión $(x - x_P)^n$ representa el enésimo producto tensorial del vector $(x - x_P)$ por él mismo, lo cual produce un tensor de rango n y el operador $\underbrace{\vdots}_n$ representa el producto interno de dos tensores de rango n , el cual crea un escalar.

Si se compara la suposición de variación espacial con la serie de Taylor, se puede ver que el error de truncamiento arranca en el tercer término, por lo cual la variación tiene una exactitud de segundo orden. De la misma forma se puede extender el análisis a la variación temporal, obteniéndose una exactitud también de segundo orden.

El método de los volúmenes finitos requiere que la Ecuación 5.101 se cumpla sobre todo el volumen de control Ω_P alrededor del punto P en su forma integral:

$$\int_t^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_{\Omega_P} \rho \phi d\Omega + \int_{\Omega_P} \nabla \cdot (\rho \vec{V} \phi) d\Omega - \int_{\Omega_P} \nabla \cdot (\rho \Gamma_{eff}^\phi \nabla \phi) d\Omega \right] dt = \int_t^{t+\Delta t} \left(\int_{\Omega_P} S_\phi(\phi) d\Omega \right) dt \quad (5.105)$$

5.3.4.1. Discretización de términos espaciales

Con base en Jasak p.78-80[19], la forma discretizada con segundo orden de exactitud del teorema de Gauss es

$$(\nabla \cdot \mathbf{a}) \Omega_P = \sum_f \mathbf{S} \cdot \mathbf{a}_f \quad (5.106)$$

donde el subíndice f indica el uso del valor de la variable (en este caso \mathbf{a}) en el centro de la cara y \mathbf{S} es el vector de área que apunta hacia afuera de volumen. los términos espaciales se pueden discretizar entonces de la siguiente forma:

Término convectivo La discretización del término convectivo es:

$$\int_{\Omega_P} \nabla \cdot (\rho \vec{V} \phi) d\Omega = \sum_f \mathbf{S} \cdot (\rho \vec{V} \phi)_f = \sum_f \mathbf{S} \cdot (\rho \vec{V})_f \phi_f = \sum_f F \phi_f \quad (5.107)$$

donde F representa el flujo másico en la cara del elemento. Este flujo se calcula a partir de valores interpolados de ρ y \vec{V} .

La Ecuación 5.107 usa el valor de la variable ϕ en la cara calculada de los valores en el centro de las celdas, el cual se obtiene de los esquemas de diferenciación de la convección, los cuales pueden ser:

Central (CD) Tiene exactitud de segundo orden aun en mallas no uniformes, sin embargo causa oscilaciones en los problemas dominados por convección, lo cual no

permite una solución acotada. se puede escribir como

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N \quad (5.108)$$

donde $f_x = \overline{fN}/\overline{PN}$ donde \overline{fN} es la distancia de la cara f al centro de la celda vecina N , y \overline{PN} es la distancia del centro de la celda al centro de la celda vecina.

esta ecuación se puede representar también como

$$\phi_f = f_{xCD}(\phi_P - \phi_N) + \phi_N \quad (5.109)$$

donde f_{xCD} es el factor para la interpolación lineal centrada.

Upwind (UD) el valor de ϕ es determinado a partir de la dirección del flujo, este esquema es acotado a expensas de la exactitud, pues introduce difusión numérica [19]. Se define como:

$$\phi_f = \begin{cases} \phi_f = \phi_P & F \geq 0 \\ \phi_f = \phi_N & F < 0 \end{cases} \quad (5.110)$$

ó

$$\phi_f = f_{xUD}(\phi_P - \phi_N) + \phi_N \quad (5.111)$$

donde f_{xUD} es 1 si $F = \vec{V} \cdot \mathbf{S}_f > 0$ y si no lo es, es igual a 0.

Mezclada (BD) Con este esquema se pretende mantener la exactitud y la solución acotada, es una combinación lineal de CD y UD, se representa por:

$$\phi_f = (1 - \gamma) (\phi_f)_{UD} + \gamma (\phi_f)_{CD} \quad (5.112)$$

ó

$$\begin{aligned} \phi_f = & [(1 - \gamma) \max(\text{sgn}(F), 0) + \gamma f_x] \phi_P \\ & + [(1 - \gamma) \min(\text{sgn}(F), 0) + \gamma (1 - f_x)] \phi_N \end{aligned} \quad (5.113)$$

donde el término γ determina cuanta difusión numérica se introducirá. también se

puede escribir:

$$\phi_f = [(1 - \gamma) f_{xUD} + \gamma f_{xCD}] (\phi_P - \phi_N) + \phi_N \quad (5.114)$$

ó

$$\phi_f = \lambda(\phi_P - \phi_N) + \phi_N \quad (5.115)$$

la cual sería la forma compacta del esquema mezclado teniendo $\lambda = (1 - \gamma) f_{xUD} + \gamma f_{xCD}$, el cual es un factor de peso limitado, conocido como el factor de “mezcla” γ , siendo $0 \leq \gamma \leq 1$, el cual puede ser constante o evaluado con base “local” como se hace en los esquemas de alta resolución basados en **disminución de variación total** (TDV por sus siglas en inglés) o **diagrama de variables normalizado** (NVD por sus siglas en inglés).

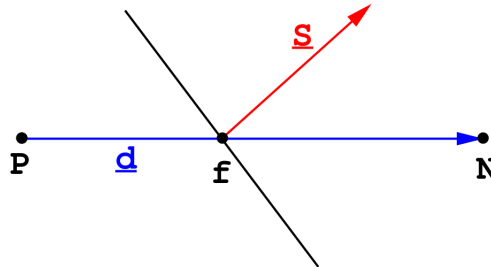
Término difusivo Si se usa la suposición de variación lineal de ϕ y la Ecuación 5.106, se puede decir

$$\int_{\Omega_P} \nabla \cdot (\rho \Gamma_{eff}^\phi \nabla \phi) d\Omega = \sum_f \mathbf{S} \cdot (\rho \Gamma_{eff}^\phi \nabla \phi)_f = \sum_f (\rho \Gamma_{eff}^\phi) \mathbf{S} \cdot (\nabla \phi)_f \quad (5.116)$$

Si la malla es ortogonal, es decir, los vectores \mathbf{S} y \mathbf{d} en la Figura 15 son paralelos es posible usar la expresión:

$$\mathbf{S} \cdot (\nabla \phi)_f = |\mathbf{S}| \frac{\phi_N - \phi_P}{|\mathbf{d}|} \quad (5.117)$$

Figura 15: Vectores \mathbf{d} y \mathbf{S} no ortogonales



Fuente: Tomado de Jasak 1996 [19].

usando la Ecuación 5.117, el valor del gradiente de ϕ en la cara puede evaluarse a partir de los valores en los dos lados de la cara, o se pueden calcular los gradientes centrados en la celda de las celdas que comparten la cara e interpolarlos a la cara [19]:

$$(\nabla\phi_f) = f_x (\nabla\phi)_P + (1 - f_x) (\nabla\phi)_N \quad (5.118)$$

y luego hacer el producto punto con \mathbf{S} .

Si la malla no es ortogonal, como en la mayoría de casos, el producto $\mathbf{S} \cdot (\nabla\phi)_f$ se descompone en dos partes:

$$\mathbf{S} \cdot (\nabla\phi)_f = \underbrace{\Delta \cdot (\nabla\phi)_f}_{\text{contribución ortogonal}} + \underbrace{\mathbf{k} \cdot (\nabla\phi)_f}_{\text{corrección no ortogonal}} \quad (5.119)$$

donde los vectores Δ y \mathbf{k} satisfacen la condición $\mathbf{S} = \Delta + \mathbf{k}$. De esta forma se usa la Ecuación 5.117 en la contribución ortogonal y descomponiendo los vectores \mathbf{S} y \mathbf{d} se puede hallar Δ de muchas formas posibles[19].

Término fuente Los términos en la ecuación de transporte original que no pueden ser escritos como difusivos, convectivos o temporales se tratan como fuentes. el término $S_\phi(\phi)$ antes de ser discretizado debe ser linealizado

$$S_\phi(\phi) = S_u + S_p(\phi) \quad (5.120)$$

donde S_u y S_p pueden depender de ϕ .

La integral de volumen se calcula con[19]:

$$\int_{\Omega_P} S_\phi(\phi) d\Omega = S_u \Omega_P + S_p \Omega_P \phi_P \quad (5.121)$$

5.3.4.2. Discretización temporal

Si se usa la forma integral de la ecuación de transporte y se sustituye la discretización del término difusivo, término convectivo y término fuente (Ecuación 5.105 ,Ecuación 5.107,

Ecuación 5.116, Ecuación 5.121) se obtiene:

$$\begin{aligned} \int_t^{t+\Delta t} \left[\left(\frac{\partial \rho \phi}{\partial t} \right) \Omega_P + \sum_f F \phi_f - \sum_f \left(\rho \Gamma_{eff}^\phi \right) \mathbf{S} \cdot (\nabla \phi)_f \right] dt \\ = \int_t^{t+\Delta t} (S_u \Omega_P + S_p \Omega_P \phi_P) dt \quad (5.122) \end{aligned}$$

Discretización de Crank-Nicholson Si se tiene variación lineal de la función en el tiempo, Ecuación 5.102, se pueden calcular directamente las integrales temporales y la derivada en el tiempo como

$$\left(\frac{\partial \rho \phi}{\partial t} \right)_P = \frac{\rho_P^n \phi_P^n - \rho_P^o \phi_P^o}{\Delta t} \quad (5.123)$$

y

$$\int_t^{t+\Delta t} \phi(t) dt = \frac{1}{2} (\phi^o + \phi^n) \Delta t \quad (5.124)$$

donde $\phi^n = \phi(t + \Delta t)$ y $\phi^o = \phi(t)$.

Asumiendo que la densidad y la difusividad no cambian en el tiempo, podemos reescribir la Ecuación 5.122 así:

$$\begin{aligned} \frac{\rho_P^n \phi_P^n - \rho_P^o \phi_P^o}{\Delta t} \Omega_P + \frac{1}{2} \sum_f F \phi_f^n - \frac{1}{2} \sum_f \left(\rho \Gamma_{eff}^\phi \right)_f \mathbf{S} \cdot (\nabla \phi)_f^n dt \\ + \frac{1}{2} \sum_f F \phi_f^o - \frac{1}{2} \sum_f \left(\rho \Gamma_{eff}^\phi \right)_f \mathbf{S} \cdot (\nabla \phi)_f^o dt \\ = S_u \Omega_P + \frac{1}{2} S_p \Omega_P \phi_P^n + \frac{1}{2} S_p \Omega_P \phi_P^o \quad (5.125) \end{aligned}$$

Esta forma de discretización temporal es conocida como el método de **Crank-Nicholson**, el cual es de segundo orden de exactitud en el tiempo[19]. Debido a que se quiere determinar el nuevo valor de ϕ_P y dado que ϕ_f y $(\nabla \phi)_f$ dependen de los valores en las celdas vecinas, se produce una ecuación algebraica de la forma:

$$a_P \phi_P^n + \sum_N a_N \phi_N^n = R_P \quad (5.126)$$

entonces se construye una ecuación de esta forma para cada volumen de control, y como ϕ_P^n depende de los valores en las celdas vecinas, se crea el sistema de ecuaciones algebraicas:

$$[A] [\phi] = [R] \quad (5.127)$$

En el cual $[A]$ es una matriz dispersa, con coeficientes a_P en la diagonal principal y a_n por fuera de la diagonal principal, $[\phi]$ es el vector de los ϕ de cada volumen de control y $[R]$ es el vector de términos fuente. El patrón de dispersión depende de la forma como se hayan numerado los volúmenes de control y cada coeficiente por fuera de la diagonal principal corresponde a una de las caras del volumen de control. Cuando este sistema se resuelve, se encuentra un nuevo conjunto de valores de ϕ para el paso nuevo del tiempo.

El método de Crank-Nicholson de discretización temporal es incondicionalmente estable pero no garantiza una solución acotada, pero como en el caso del término convectivo, se puede conseguir una solución acotada si se discretiza la ecuación hasta el primer orden de exactitud temporal.

Es de uso común desestimar la variación de los valores en las caras de ϕ y $\Delta\phi$ en el tiempo, esto conduce a varios métodos de discretización temporal. La nueva forma de la ecuación que combina los nuevos y viejos pasos del tiempo en convección, difusión y términos fuente, pero dejando la derivada temporal intacta, queda:

$$\begin{aligned} \frac{\rho_P^n \phi_P^n - \rho_P^o \phi_P^o}{\Delta t} V_P + \sum_f F \phi_f - \sum_f \left(\rho \Gamma_{eff}^\phi \right)_f \mathbf{S} \cdot (\nabla \phi)_f dt \\ = S_u V_P + S_p V_P \phi_P \end{aligned} \quad (5.128)$$

Esta ecuación resultante es solamente de primer orden de exactitud en el tiempo y se debe escoger la forma como se evalúan los valores de ϕ y $\nabla\phi$.

Discretización explícita Los valores en la cara de ϕ y $\nabla\phi$ se determinan a partir del campo del tiempo anterior así:

$$\phi_f = f_x \phi_P^o + (1 - f_x) \phi_N^o \quad (5.129)$$

$$\mathbf{S} \cdot (\nabla \phi)_f = |\Delta| m \frac{\phi_N^o - \phi_P^o}{|\mathbf{d}|} + \mathbf{k} \cdot (\nabla \phi)_f^o \quad (5.130)$$

la parte lineal del término fuente también se determina a partir del tiempo anterior, entonces se puede escribir:

$$\phi_P^n = \phi_P^o + \frac{\Delta t}{\rho_P \Omega_P} \left[- \sum_f F \phi_f + \sum_f \left(\rho \Gamma_{eff}^\phi \right)_f \mathbf{S} \cdot (\nabla \phi)_f dt + S_u \Omega_P + S_p \Omega_P \phi_P^o \right] \quad (5.131)$$

de forma que el lado derecho de la ecuación solo depende de valores de campo anteriores y el campo se puede calcular directamente, sin resolver el sistema de ecuaciones, el inconveniente que presenta este método es el límite en el número de Courant, que se define como

$$Co = \frac{\vec{V}_f \cdot \mathbf{d}}{\Delta t} \quad (5.132)$$

Si el número de Courant es mayor a 1 el sistema explícito es inestable

Discretización Euler implícita Este tipo de discretización expresa los valores en las caras en términos de los nuevos valores en la celda, así:

$$\phi_f = f_x \phi_P^n + (1 - f_x) \phi_N^n \quad (5.133)$$

$$\mathbf{S} \cdot (\nabla \phi)_f = |\Delta| m \frac{\phi_N^n - \phi_P^n}{|\mathbf{d}|} + \mathbf{k} \cdot (\nabla \phi)_f \quad (5.134)$$

Este esquema tiene exactitud de primer orden solamente, se necesita el sistema de ecuaciones para ser resuelto, es estable y produce soluciones acotadas.

5.3.5. Condiciones de frontera

La malla computacional incluye una serie de caras que coinciden con las fronteras del dominio físico considerado, las condiciones allí presentes se determinan a partir de las

condiciones de frontera. Las condiciones de frontera se pueden dividir en dos tipos, numéricas y físicas.

Las condiciones de frontera numéricas se pueden dividir en dos tipos básicos, Dirichlet (o valor fijo) que indica el valor de la variable en la frontera y Newmann que indica el valor del gradiente de la variable en la frontera. Estos dos se incorporan al sistema de ecuaciones algebraicas antes de la solución.

Las condiciones de frontera físicas para problemas de flujo de fluidos son planos de simetría, paredes, entradas y salidas.

5.3.5.1. Condiciones de frontera numéricas

Condición de frontera de valor fijo (Dirichlet) Esta condición de frontera determina el valor de ϕ en la cara b , ϕ_b , y debe tomarse en cuenta en la discretización de los términos convectivos y difusivos.

Término convectivo De acuerdo con la Ecuación 5.107, y si se sabe que ϕ en la cara b es ϕ_b , este término para la cara en la frontera es:

$$F_b \phi_b \quad (5.135)$$

con F_b como el flujo en la cara.

Término difusivo De acuerdo con la Ecuación 5.116, según [19]:

$$\mathbf{S} \cdot (\nabla \phi)_b = |\mathbf{S}| \frac{\phi_b - \phi_P}{|\mathbf{d}_n|} \quad (5.136)$$

debido a que \mathbf{S} y \mathbf{d}_n son paralelos.

Condición de frontera de gradiente fijo (Newmann) Aquí el producto punto de el gradiente y el vector de área unitario está dado en la frontera:

$$\left(\frac{\mathbf{S}}{|\mathbf{S}|} \cdot \nabla \phi \right)_b = g_b \quad (5.137)$$

Término convectivo El valor de ϕ se calcula del valor del centro de la celda y el gradiente establecido:

$$\phi_b = \phi_P + \mathbf{d}_n \cdot (\nabla \phi)_b$$

$$\phi_b = \phi_P + |\mathbf{d}| \cdot g_b \quad (5.138)$$

Término difusivo El producto punto entre el vector de área de la cara y $(\nabla \phi)_b$ es conocido como

$$|\mathbf{S}|g_b \quad (5.139)$$

y se obtiene

$$\left(\rho \Gamma_{eff}^\phi\right)_b |\mathbf{S}|g_b \quad (5.140)$$

5.3.5.2. Condiciones de frontera física

Para el problema de estudio, se tienen las siguientes condiciones de frontera:

Frontera de entrada. La velocidad a la entrada es dada y la condición es de presión es de gradiente 0.

Frontera de salida. Existen dos posibles enfoques:

- La distribución de la velocidad en la frontera se proyecta desde adentro del dominio. Estas velocidades se escalan para satisfacer la continuidad. Es inestable si ocurre flujo entrante en una frontera definida de salida. En esta condición de frontera la condición de frontera de la presión se debe ajustar a gradiente cero.
- Se especifica la distribución de presión, por tanto se usa valor fijo para la presión con gradiente cero para la condición de frontera de velocidad.

Paredes impermeables no deslizantes. La velocidad del fluido en la pared es igual a la velocidad de la pared por la condición de no deslizamiento, entonces la velocidad tiene condición de valor fijo y como se sabe que el flujo a través de la pared es 0 la condición de frontera para la presión es gradiente cero.

5.3.6. Flujos turbulentos

Las condiciones de entrada y salida para variables de turbulencia (en este caso k y ϵ) son típicamente fijadas como valor fijo (entrada) y gradiente cero (salida). En la pared las condiciones de frontera dependen del tratamiento que se dé cerca de esta.

5.3.7. Procedimiento de discretización

Como primer paso se discretiza la ecuación de concentraciones de forma que:

$$\frac{\rho\alpha_{1P}^n - \rho\alpha_{1P}^o}{\Delta t}\Omega_P + \sum_f \mathbf{S}_f \cdot \vec{V} (\lambda(\alpha_{1P} - \alpha_{1N}) + \alpha_{1N})_f - \sum_f (D_{eff})_f |S_f| \frac{\rho_{1N} - \rho_{1P}}{|\mathbf{d}|} = 0 \quad (5.141)$$

donde λ en el término convectivo es determinado a partir del esquema numérico escogido y en el caso de mallas no ortogonales se introduce un término adicional al término difusivo, ver la Ecuación 5.119. Luego se pasa a la discretización de la ecuación de continuidad y por último, a la ecuación de conservación de la cantidad de movimiento. en este proceso hay dos detalles que requieren atención especial; la ecuación de conservación de la cantidad de movimiento no es lineal y el acoplamiento que se da entre la velocidad y la presión.

La no linealidad de la ecuación de conservación de la cantidad de movimiento se debe al término $\nabla \cdot \vec{V} \vec{V}$ o, en otras palabras, al hecho de que la velocidad es transportada por ella misma. La discretización de este término resultaría cuadrática en velocidad y el sistema algebraico asociado por lo tanto sería no lineal. Este problema se puede solucionar de dos formas, usar un solucionador para sistemas no lineales o linealizar el término convectivo. Teniendo en cuenta el esfuerzo computacional requerido para la solución de un sistema no lineal, se escogió la linealización, como se vió anteriormente, implica que el campo de flujo de velocidad existente, que satisface la ecuación de cantidad de movimiento, se usará para calcular a_P y a_N .

Para este trabajo de investigación se usó el algoritmo PIMPLE, una mezcla del algoritmo SIMPLE propuesto por Patankar [20] y PISO propuesto por Issa [29], Este algoritmo (PIMPLE) se usó en modo PISO para resolver el acoplamiento de la velocidad y la presión.

5.3.7.1. Ecuación de presión

Para derivar la ecuación de la presión se usa una forma semi-discretizada de la ecuación de la presión, así:

$$a_P \vec{V}_P = \mathbf{H}(\vec{V}) - \nabla p \quad (5.142)$$

en esta etapa el gradiente de presión no sera discretizado. Esta ecuación se obtiene de la forma integral usando las técnicas de discretización anteriormente descritas y dividiendo en el volumen. El termino $\mathbf{H}(\vec{V})$ está compuesto por dos partes, la parte “transportada”, la cual incluye la matriz de coeficientes de todos los vecinos multiplicados por su correspondiente velocidad, y la parte “fuente” que incluye la parte fuente del término transitorio, ó:

$$\mathbf{H}(\vec{V}) = - \sum_N a_N \vec{V}_N + \frac{\vec{V}^o}{\Delta t} \quad (5.143)$$

como ya se vio la forma discretizada de la ecuación de continuidad es:

$$\nabla \cdot \vec{V} = \sum_f \mathbf{S}_f \cdot \vec{V}_f = 0 \quad (5.144)$$

para expresar \vec{V} :

$$\vec{V}_P = \frac{\mathbf{H}(\vec{V})}{a_P} - \frac{1}{a_P} \nabla p \quad (5.145)$$

y como se necesitan las velocidades interpoladas en la cara:

$$\vec{V}_f = \left(\frac{\mathbf{H}(\vec{V})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\nabla p)_f \quad (5.146)$$

Esto se usará para calcular los flujos en las caras. Entonces se reemplaza la Ecuación 5.146 en la Ecuación 5.144 así:

$$\nabla \cdot \left(\frac{1}{a_P} \nabla p \right)_f = \nabla \cdot \left(\frac{\mathbf{H}(\vec{V})}{a_P} \right)_f = \sum_f \mathbf{S} \cdot \left(\frac{\mathbf{H}(\vec{V})}{a_P} \right)_f \quad (5.147)$$

de forma general el sistema quedaría

$$a_P \vec{V}_P = \mathbf{H}(\vec{V}) - \sum_f \mathbf{S}(p)_f \quad (5.148)$$

$$\begin{aligned} \sum_f \mathbf{S} \cdot \left[\left(\frac{1}{a_P} \right)_f (\nabla p)_f \right] &= \sum_f \mathbf{S} \cdot \left(\frac{\mathbf{H}(\vec{V})}{a_P} \right)_f \\ \left[\sum_f \Delta \cdot \left(\frac{1}{a_P} \right)_f \left(\frac{p_N - p_P}{|\mathbf{d}|} \right)_f + \sum_f \mathbf{k} \cdot \left(\frac{1}{a_P} \right)_f \left(\frac{p_N - p_P}{|\mathbf{d}|} \right)_f \right] &= \sum_f \mathbf{S}_f \cdot \left(\frac{\mathbf{H}(\vec{V})}{a_P} \right)_f \end{aligned} \quad (5.149)$$

donde Δ es la contribución ortogonal y \mathbf{k} es la contribución no ortogonal en caso de malla no ortogonal, Ecuación 5.119. Enseguida se calcula el flujo F usando Ecuación 5.146:

$$F = \mathbf{S} \cdot \vec{V}_f = \mathbf{S} \cdot \left[\left(\frac{\mathbf{H}(\vec{V})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\nabla p)_f \right] \quad (5.150)$$

cuando la Ecuación 5.147 se cumple, los flujos en la cara son conservativos.

5.3.7.2. Acoplamiento de presión y velocidad

Si analizamos el sistema formado por la Ecuación 5.148 y Ecuación 5.147 se puede apreciar una dependencia lineal de la velocidad y la presión y viceversa. Este acoplamiento requiere un tratamiento especial. Existen dos formas de afrontar este problema, la solución simultanea y el abordaje segregado, el primero es usado cuando el número de puntos es pequeño y por lo tanto la cantidad de ecuaciones no es tan grande debido a que la matriz resultante es varias veces mas grande que el numero de puntos computacionales, por lo tanto su costo computacional es mayor. Como segunda opción se cuenta con el abordaje segregado, en el cual las ecuaciones son resueltas de forma secuencial, pero se requiere un tratamiento especial para definir el acoplamiento entre ecuaciones. Los algoritmos PISO y SIMPLE y sus derivados son los métodos mas usados para resolver el acoplamiento entre presión y velocidad.

5.3.7.3. Algoritmo PISO

Fue originalmente propuesto por Issa [29], si se retoma el sistema formado por la Ecuación 5.147 y la Ecuación 5.148, el algoritmo PISO se puede describir de la siguiente forma:

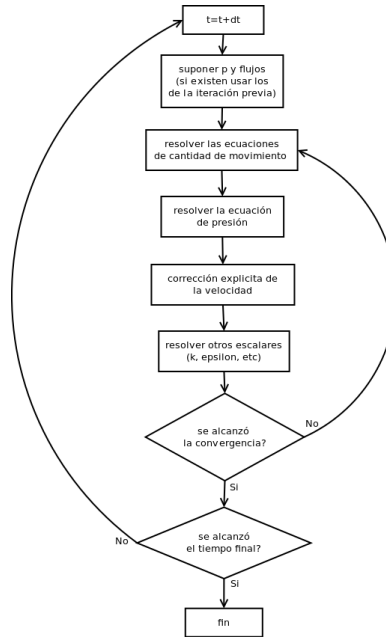
- Se resuelve la ecuación de cantidad de movimiento. En este momento no se conoce exactamente el término fuente de gradiente de presión (se usa el campo de presión del paso anterior, si se tiene). Esta etapa se conoce como el paso **predictor de cantidad de movimiento**, en este la solución de la cantidad de movimiento, Ecuación 5.148, da una aproximación del nuevo campo de velocidades.
- Con este campo de velocidades pronosticado se construye el operador $\mathbf{H}(\vec{V})$ y se puede formular la ecuación de la presión, donde su solución entrega la primera estimación del campo de presiones. Este paso se conoce como la **solución de presiones**.
- Tomando la Ecuación 5.150, se obtiene un conjunto de flujos conservativos que concuerdan con el nuevo campo de presiones, entonces se debe corregir el campo de velocidades como consecuencia del nuevo campo de presiones usando la Ecuación 5.145. este es el paso llamado **corrección explícita de la velocidad**.

Si analizamos la Ecuación 5.145, se puede observar que la corrección de la velocidad está compuesta por dos partes, una debida al cambio en el gradiente de presión y otra debida a la influencia transportada de la corrección de las velocidades de los vecinos, entonces lo que se quiere decir con corrección explícita es que se ignora esta última parte y se asume que todo el error en la velocidad se debe a el error en el termino de presión. Es evidente que esto no es cierto, por lo tanto es necesario corregir el termino $\mathbf{H}(\vec{V})$, formular la nueva ecuación de presión y se repite el procedimiento. Entonces el bucle PISO consiste en un paso predictor de cantidad de movimiento seguido de soluciones de presión y correcciones explícitas de velocidad. Este bucle se repite hasta alcanzar una tolerancia determinada.

También se puede observar que a pesar de la dependencia que existe de los coeficientes de los flujos con $\mathbf{H}(\vec{V})$ y que después de cada solución de presiones es posible recalcular los coeficientes de $\mathbf{H}(\vec{V})$, esto no se hace: lo que sucede es que se asume es que el acoplamiento no lineal es menos importante que el acoplamiento entre presión y velocidad, esto está de acuerdo con la linealización de la ecuación de cantidad de movimiento,

por lo tanto los coeficientes de $\mathbf{H}(\vec{V})$ no se cambian hasta el siguiente paso predictor de cantidad de movimiento.

Figura 16: Algoritmo PISO



5.3.8. Solución del sistema

Para solucionar el sistema de flujo incompresible de dos fluidos que se mezclan en una tubería se debe:

1. Establecer las condiciones iniciales de todos los campos $(\vec{V}, p, \alpha_1, k, \epsilon)$
2. Se ensambla y resuelven los flujos convectivos de α , luego se ensambla y se resuelve la ecuación de difusión de α .
3. Ensamblar y resolver la ecuación de predicción de cantidad de movimiento con los flujos disponibles en las caras.
4. Hacer el bucle PISO hasta que la tolerancia del sistema presión-velocidad se alcance. Se obtienen los campos de presión y velocidad para el paso de tiempo actual y también un nuevo conjunto de flujos conservativos.
5. Si el flujo es turbulento se calcula la viscosidad efectiva a partir de las variables turbulentas.
6. Si no se alcanza el tiempo final, se retoma el paso 2.

6 METODOLOGÍA

El desarrollo de este trabajo de investigación se dividió en cuatro etapas, indagación del estado del arte de la determinación del volumen de mezcla de la interfase de dos fluidos incompresibles miscibles que se desplazan por una tubería, luego se realizó el estudio de la programación en C++ aplicada al método de los volúmenes finitos, después se desarrolló el estudio de los instrumentos necesarios en el trabajo (mallado, solución y visualización de resultados), se estudió el solucionador para dos fluidos incompresibles miscibles, se usó y se compararon los resultados obtenidos con los encontrados en la literatura, se desarrolló el solucionador para dos fluidos incompresibles miscibles con malla móvil concluyendo con la simulación el problema de investigación.

6.1. INSTRUMENTOS

Para la determinación de los campos de flujo y de concentraciones fue usado el software de código abierto OpenFOAM, la primera razón para su uso fue precisamente que sea software de código abierto, dado que las herramientas comerciales son costosas. Además de esto, OpenFOAM se emplea en una gran cantidad de aplicaciones para diferentes áreas de la ingeniería y la ciencia, está escrito en lenguaje de programación C++, lenguaje moderno y eficiente para cálculos numéricos además de muy popular. OpenFOAM se ha usado para resolver problemas que involucran flujos complejos de fluidos con reacciones químicas, transferencia de calor y turbulencia, también funciona con mecánica de sólidos y electromagnetismo. OpenFOAM incluye herramientas de preproceso y post-proceso, además ofrece la posibilidad de trabajar en paralelo, para que de esta forma el usuario pueda usar todo el poder de su computador cuando trabaja. Para el uso de OpenFOAM fue necesario emplear el sistema operativo libre Linux, específicamente la distribución Ubuntu.

Otros software y lenguajes de programación utilizados fueron: Python™, el cual es un renombrado y poderoso lenguaje de programación dinámica usado en muchas aplicaciones, comparable a Tcl, Perl, Ruby o Java. Python™ es software libre, multi-plataforma y maneja una licencia de código abierto. En la etapa de preproceso se

usó la plataforma de software SALOME , este es un software libre multiplataforma escrito en C++ y Python™, fue escogido debido a la facilidad de parametrizar la geometría de este estudio ya que cuenta con dos modos de interacción , una interfaz gráfica acoplada con interacción gráfica 3D usando Qt4 y VTK y una interfaz de texto basada en Python™. Para la generación de la geometría y la construcción de la malla se emplearon scripts escritos en Python™[30][31, 32] (ver Anexo B). En la etapa de post-proceso se usó el software para análisis y visualización de datos de código abierto ParaView, en esta etapa de proceso también se usó la librería PyFoam[33], esta es una librería escrita en Python™ usada para analizar los archivos log producidos por los solucionadores de OpenFOAM, ejecutar , analizar sus salidas simultáneamente, manipular los archivos de parámetros y las condiciones iniciales así como para dibujar los residuales de los solucionadores.

6.1.1. OpenFOAM®

6.1.1.1. Generalidades

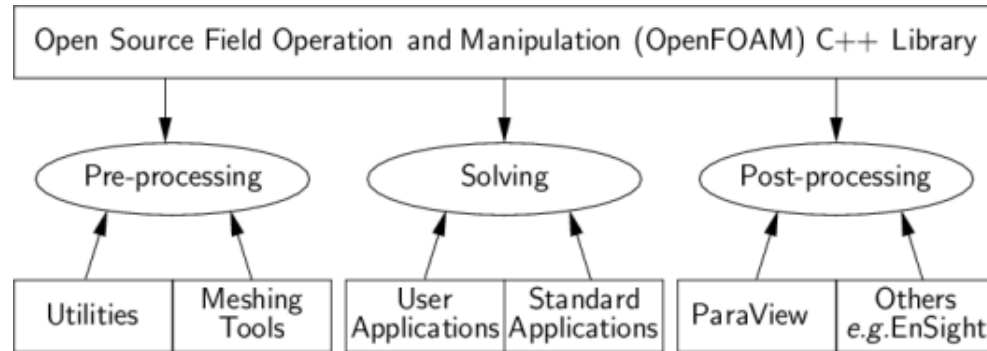
OpenFOAM es principalmente una librería escrita en C++, usada para crear ejecutables conocidos como aplicaciones. Las aplicaciones se dividen en dos categorías: *solvers*, cada uno diseñado para resolver un problema específico en mecánica del continuo; y *utilities* que están diseñadas para ejecutar tareas que involucren manipulación de datos. OpenFOAM contiene numerosos *solvers* y *utilities* que cubren un gran rango de problemas.

Una de las grandes fortalezas de OpenFOAM consiste en que se pueden crear nuevos *solvers* y *utilities* por usuarios con algunos conocimientos del método involucrado, la física y las técnicas de programación necesarias[24, 34]. OpenFOAM viene con entornos de pre y post procesado y esto garantiza un manejo consistente de datos entre los entornos. La estructura general de OpenFOAM se muestra en (Figura 17).

Un punto importante del diseño de OpenFOAM es que las aplicaciones solver, escritas usando las clases OpenFOAM, tienen una sintaxis muy similar a las ecuaciones diferenciales parciales que están siendo resueltas. Por ejemplo

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \varphi \mathbf{U} - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p \quad (6.1)$$

Figura 17: Estructura de OpenFOAM



Fuente: [35]

donde \mathbf{U} representa el vector velocidad \vec{V} , se representa por el Código 6.1.

Código 6.1: Representación de la Ecuación 6.1

```
1  solve
2  (
3      fvm::ddt(rho, U)
4      + fvm::div(phi, U)
5      - fvm::laplacian(mu, U)
6      ==
7      - fvc::grad(p)
8  );
```

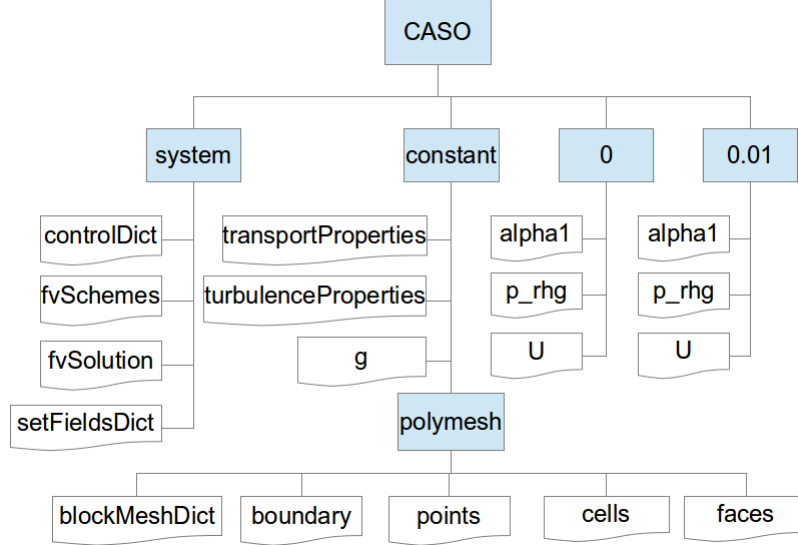
Esto implica que el lenguaje principal de programación de OpenFOAM posee características de programación orientada a objetos como herencia, plantillas de clases, funciones virtuales y sobrecarga de operadores. Es por esto que el lenguaje principal de OpenFOAM es C++[35].

6.1.1.2. Estructura de los casos en OpenFOAM

Los casos en OpenFOAM están constituidos por archivos organizados en una estructura de carpetas (Figura 18). En esta estructura se encuentran archivos separados para la malla, los campos de velocidad, presión y turbulencia, así como para las constantes y ajustes necesarios. Los archivos necesarios para correr una simulación son comúnmente llamados diccionarios. Dependiendo del caso y de los ajustes especificados, se necesitarán más archivos, por ejemplo, cuando se mapean campos de un caso a otro. Estos

archivos se mencionarán en la medida que sean relevantes.

Figura 18: Estructura de caso



Directorio constant Esta carpeta contiene las constantes físicas necesarias para la simulación *transportProperties*, definición del modelo de turbulencia *turbulenceProperties*, también contiene la descripción de la malla dentro de la carpeta *polyMesh*.

Directorio system Contiene el control de la simulación, *controlDict*, esquemas de discretización, *fvSchemes*, parámetros de ajuste para el solucionador, *fvSolution*, ajuste inicial de campos *setFieldsDict*, parámetros para el mapeo de campos desde otro caso, *mapFieldsDict*; También incluye *decomposeParDict*, que contiene los ajustes para la descomposición del caso para ejecutarlo en paralelo.

Directorios de tiempo Contiene los parámetros iniciales de los campos que se encuentran en el directorio *0*, en tanto que los resultados son escritos por OpenFOAM en directorios nombrados para cada instante de tiempo, por ejemplo *0.01*. Dentro de estos directorios se encuentran archivos separados para los campos de la velocidad \vec{V} (*U*), presión (*p_rhg*, el cual incluye el término *pgh*), cantidades escalares como concentración de una sustancia α_1 (*alpha1*) o cantidades turbulentas como ϵ (*epsilon*).

6.1.1.3. Geometría y mallado

OpenFOAM trae herramientas para la generación de la malla y para su manipulación.

blockMesh la herramienta *blockMesh* es un generador nativo de malla que se usa para generación de geometrías y mallas simples. Si se usa es necesario el diccionario *constant/polyMesh/blockMeshDict*.

snappyHexMesh Para geometrías mas complejas se puede usar *snappyHexMesh*. Si se usa, los archivos de geometría se deben suministrar en el formato de *estereolitografía* (*.STL) y se define un dominio que la contenga con la herramienta *blockMesh*. *snappyHexMesh* malla la superficie de la geometría en el archivo *.STL usando el dominio “contenedor” como malla base. Esta herramienta posee muchas opciones para la generación y permite un gran control sobre el resultado. Si es usada, es necesario el diccionario *system/snappyHexMeshDict*.

ideasUnvToFoam Existen muchas herramientas para convertir formatos de malla a mallas que pueda leer OpenFOAM. *ideasUnvToFoam* es una de ellas y es usada para convertir mallas generadas en formato *.unv a formato OpenFOAM.

Inicialmente se seleccionó la herramienta *blockMesh*, pero debido a su rigidez se optó por usar la plataforma SALOME, que es mas flexible y permite parametrización, editar mallas, verificar la calidad de las mismas e importar y exportar en formatos *MED*, *UNV*, *DAT* y *STL*.

6.1.1.4. Solver

Los códigos de los solvers son en gran parte procedimentales, debido a que son una representación bastante cercana de las ecuaciones y los algoritmos de solución, los cuales son procedimentales *per se*. Los solucionadores estándar que vienen con OpenFOAM están almacenados en un directorio al que se accede desde la terminal con la variable de entorno \$FOAM_SOLVERS o con los alias “foamSol” o “sol”. Este directorio está dividido en varios directorios por categorías de la mecánica del continuo: Básicos, flujos incompresibles, flujos compresibles, flujos multifásicos, simulación numérica directa (DNS), combustión, transferencia de calor y flujos por flotación, flujos de rastreo de

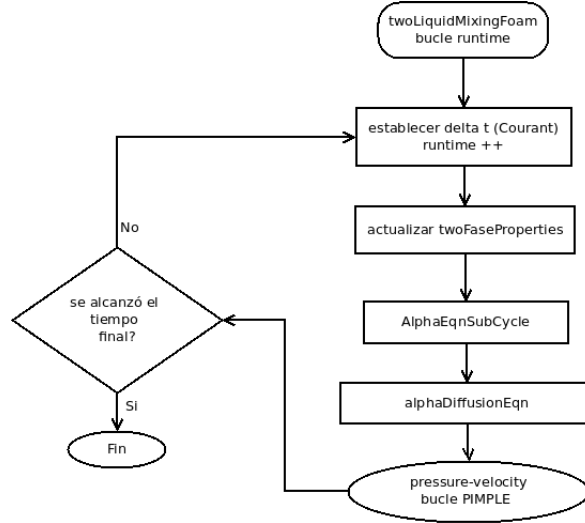
partículas, métodos de dinámica molecular, simulación directa con el método de monte-carlo (DSMC), electromagnetismo, análisis de esfuerzo en sólidos y finanzas. La lista completa de los solucionadores distribuidos con OpenFOAM se pueden ver en la página U-83 de [35].

En este trabajo de investigación se usó el solucionador multifásico *twoLiquidMixingFoam*, solucionador para la mezcla de 2 fluidos incompresibles.

6.1.1.5. twoLiquidMixingFoam.C

Este solucionador está estructurado de la siguiente forma, Figura 19:

Figura 19: Algoritmo de twoLiquidMixingFoam



Luego de iniciar los archivos de encabezado, se procede a dar inicio al bucle *runtime*, en el cual se empieza estableciendo un Δt dependiendo del numero de Courant, luego se da inicio al bucle temporal, enseguida se actualizan las propiedades de las fases en el ducto, se pasa a resolver la parte convectiva de la ecuación de concentraciones un determinado número de subciclos (ver Anexo A) para pasar a resolver la parte difusiva de la ecuación de concentraciones y por último resolver el sistema acoplado presión velocidad en el bucle PIMPLE en su modo PISO (ver Anexo A) en cada paso del tiempo, si el tiempo final se alcanza, se termina el proceso, de no ser así se avanza un paso en el tiempo. Para una descripción más detallada del solver ver Anexo A.

6.1.2. Uso del solucionador

6.1.2.1. Simulación de un caso laminar

En esta sección se asume que el lector tiene un conocimiento básico acerca de cómo se configura un caso de simulación en OpenFOAM y solo se mencionarán los pasos más importantes para la configuración de un caso para el solucionador que se explicó en la sección anterior. En caso de necesitar mas información acerca de cómo preparar casos, referirse a el manual del usuario de OpenFOAM [35] capítulos 1 y 2.

Para correr un caso laminar es necesario ajustar algunos archivos (ver anexo B); el switch *simulationType* del diccionario *constant/turbulenceProperties* a laminar, ajustar la difusividad entre los fluidos y las propiedades de los mismos dentro del diccionario *constant/transportProperties*, ajustar los parámetros de la simulación dentro del directorio *system/controlDict*, escoger los esquemas numéricos adecuados dentro del diccionario *system/fvSchemes*, definir el campo de concentraciones inicial dentro del directorio *system/setFieldsDict* y configurar las entradas *pRefValue* y *pRefPoint* dentro del subdiccionario PIMPLE del diccionario *system/fvSolution*. También es necesario ajustar los archivos en los que se encuentran los campos iniciales: concentraciones (*alpha1*), presiones (*p_rgh*) y velocidades (*U*), ubicados dentro del directorio *0.org* (ver Anexo B). Dentro de esos archivos también se ajustan las condiciones de frontera para cada uno de estos campos, ver (sec.8.2). Luego se debe crear la carpeta 0 y copiar la carpeta 0.org en su interior.

Los casos en OpenFOAM se ejecutan desde la terminal de linux, si la geometría se definió en el diccionario *blockMeshDict* , se debe escribir en la terminal, ubicado en el directorio del caso

```
$ blockMesh
```

inmediatamente se debe ajustar los campos, en este caso *alpha1* introduciendo la orden

```
$ setFields
```

y luego dar comienzo a la simulación con la orden

```
$ twoLiquidMixingFoam
```

y empezará a ejecutarse el solucionador. Cabe anotar que esta es la forma más sencilla de correr el caso, en la cual se verán las salidas que envía el solucionador en la terminal hasta que finalice el tiempo estipulado en la simulación, si no hay ningún problema.

6.1.2.2. Simulación de un caso turbulento

En el caso de la simulación de un flujo turbulento se debe proceder de forma similar al caso laminar pero con algunas variaciones, también se debe ajustar el switch *simulation-Type* del diccionario *constant/turbulenceProperties* pero en este caso se debe escoger el modelo turbulento a desarrollar, RASmodel o LESmodel, dependiendo de esta elección se debe incluir el diccionario RASProperties o LESproperties dentro del directorio *constant*, donde se especifican las propiedades del modelo turbulento escogido. Se deben elegir los esquemas numéricos (*system/fvSchemes*), definir el campo de concentraciones inicial (*system/setFieldsDict*) y configurar los valores del diccionario *system/fvSolution* de la misma forma que se hizo en el caso laminar, pero con los parámetros adecuados para el caso. Otros archivos a modificar, como se hizo en el caso laminar, son los archivos *alpha1*, *p_rgh* y *U* contenidos en el directorio *0.org*, crear y copiar la carpeta temporal 0 y copiar el contenido de 0.org en esta.

El procedimiento para ejecutar el caso turbulento es básicamente el mismo que en el caso laminar, se deben ejecutar las mismas ordenes en la terminal.

6.1.2.3. Mapeo de campos y solución en paralelo (MPI)

Mapeo de campos. En el caso de querer mapear la información de un campo de un caso anteriormente ejecutado, son necesarias algunas modificaciones, se debe modificar el diccionario *controlDict* para especificar el paso del tiempo en el que se debe empezar la simulación, así como es necesaria la inclusión y configuración del archivo *mapFieldsDict* con los datos de fronteras coincidentes que se desean copiar y las fronteras no coincidentes que se quieren usar como “planos de corte” en el directorio *system*. También se debe hacer la inclusión de la carpeta temporal incluyendo el archivo del campo que se desea mapear, p.ej., *U* (campo de velocidades). Cuando se ha hecho esto, se puede ejecutar en la terminal, ubicándose en la carpeta del caso

```
$ mapFields
```

y los campos serán mapeados en el archivo adecuado.

Solución en paralelo (MPI). A diferencia con otros software privativos, OpenFOAM permite la solución de sus casos en paralelo, lo cual disminuye el tiempo máquina empleado en su solución. En una simulación FVM paralelizada en OpenFOAM, el dominio es descompuesto en mallas, cada una asociada a un procesador. Este primer paso es conocido como **descomposición del dominio**, luego se establece un mecanismo de transferencia de datos ente procesadores, se analiza la interdependencia entre puntos y se establecen las restricciones de la malla y se resuelve los cálculos locales de cada procesador, por último se reconstruyen los datos reconstruyendo malla y datos.

Para poder paralelizar un solver en OpenFOAM es necesario crear el diccionario `decomposeParDict` el cual comprende la elección del tipo de descomposición y datos auxiliares necesarios. Para ejecutar esta descomposición, basta con usar la herramienta escribiendo la orden en la terminal

```
$ decomposePar
```

y se crearán los n directorios donde se descompondrán la malla y los campos.

Para la ejecución en paralelo simplemente se lanza el ejecutable usando `mpirun` con la opción `-parallel` y los datos se crearán en la carpeta *processor* correspondiente con base en los datos ingresados en la etapa de preproceso. El caso se corre en paralelo escribiendo en la terminal la orden

```
$ mpirun -np (numero_de_procesadores) twoLiquidMixingFoam -parallel
```

Por último, para poder visualizar los datos de todo el dominio, basta con ejecutar la orden

```
$ reconstrucPar
```

y se reconstruirán las carpetas temporales con los datos completos de la simulación para una sola CPU.

6.2. SOLUCIONADOR DESARROLLADO

6.2.1. Malla móvil

Con el propósito de resolver el problema de la mezcla de dos fluidos incompresibles miscibles transportados en una tubería de gran longitud y dado que el dominio com-

putacional tan extenso exigiría una capacidad de computo muy grande, se procede a implementar la funcionalidad de malla móvil al solver **twoLiquidMixingFoam**, para ser nombrado, haciendo uso de las convenciones empleadas en OpenFOAM como **twoLiquidMixingDyMFoam** donde el agregado **DyM** hace referencia a *Dynamic Mesh* o funcionalidad de malla dinámica.

6.2.1.1. MVF en mallas poliédricas móviles

El método de los volúmenes finitos para mallas móviles se basan en la forma integral de la ecuación gobernante aplicada en un volumen arbitrario limitado por una superficie cerrada S . De forma general para una propiedad tensorial ϕ se tiene según [27]:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \phi \partial \Omega + \oint_S \rho \mathbf{n} \cdot (\mathbf{V} - \mathbf{V}_s) \phi \partial S - \oint_S \rho \gamma_{\phi} \mathbf{n} \cdot \nabla \phi \partial S = \int_{\Omega} s_{\phi} \partial \Omega \quad (6.2)$$

en la cual ρ es la densidad, \mathbf{n} es el vector unitario normal apuntando hacia afuera de la superficie de frontera, \mathbf{V} es la velocidad del fluido, \mathbf{V}_s es la velocidad de la superficie de frontera, γ_{ϕ} es el coeficiente de difusión y s_{ϕ} es el termino fuente volumétrico de ϕ . También existe una relación entre la razón de cambio del volumen Ω y la velocidad \mathbf{u}_s de la superficie de frontera S definida por la ley de conservación del espacio *SCL* (Trulio y Trigger 1961, citado por Demirdzic y Peric [36]; Thomas y Lombard [37]):

$$\frac{\partial}{\partial t} \int_{\Omega} \partial \Omega - \oint_S \mathbf{n} \cdot \mathbf{V}_s \partial S = 0 \quad (6.3)$$

Esta ecuación recibe su nombre debido a que se ocupa de las propiedades geométricas y tiene la misma forma general (Ecuación 5.100) de la ecuación de conservación de la masa (Tabla 2), relaciona el cambio del volumen de control elemental y la velocidad del sistema coordenado y se debe satisfacer junto con las otras ecuaciones de conservación, ya que si no se cumple se introducen errores en forma de fuentes artificiales de masa [36].

Usando la Ecuación 6.2, el término convectivo cambiaría a la siguiente forma discretizada

$$\sum_f (\dot{m}_f - \rho_f^n \dot{\Omega}_f^n) \phi_f^n \quad (6.4)$$

donde es sufijo f representa los valores en la cara y el superíndice n representa el nivel “nuevo” del tiempo, $\dot{m}_f = \rho \mathbf{n}_f \cdot \mathbf{V}_f S_f$ es el flujo másico y $\dot{\Omega}_f = \mathbf{n}_f \cdot \mathbf{V}_{sf} S_f$ es el flujo volumétrico específico de la cara, el cual es calculado directamente de consideraciones geométricas y satisface la forma discreta de la Ecuación 6.3.

6.2.2. Manipulación de mallas en OpenFOAM

Existen dos formas generales de manipular mallas mediante OpenFOAM. Una de ellas se ocupa únicamente de calcular el movimiento de los puntos internos para reaccionar lo mejor posible al desplazamiento de alguna o todas sus fronteras, esta recibe el nombre de movimiento de malla automático, o simplemente deformación de malla, en el cual no se modifica el número de puntos internos y se calcula su nueva posición. Por otro lado, hay otra forma de manipular mallas en las que se modifica su topología. A diferencia del cálculo de los puntos internos, los cambios topológicos modifican el número de elementos de la malla o su conectividad, por ejemplo, abriendo o cerrando un conducto durante la simulación.

6.2.3. Modificaciones al solver twoLiquidMixingFoam

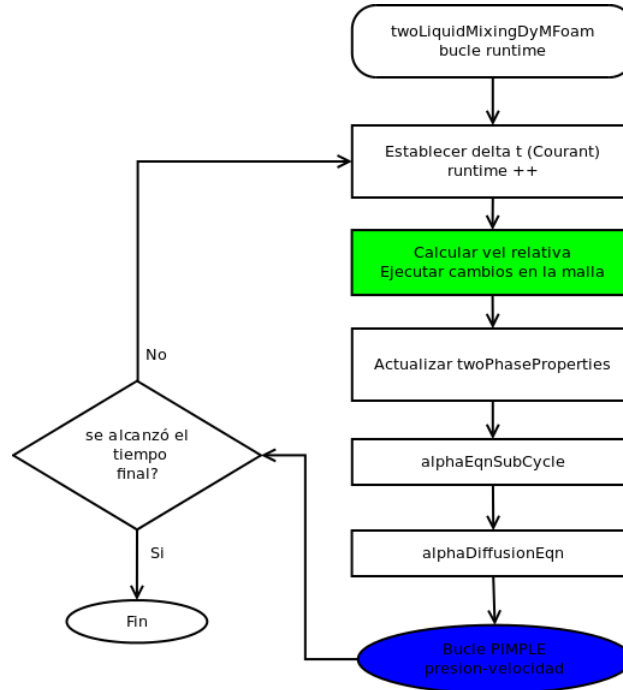
En la Figura 20 se muestran con colores las modificaciones hechas al algoritmo de solución, el fragmento del código modificado **twoLiquidMixingDyMFoam** se muestra en el Código 6.2

Código 6.2: twoLiquidMixingDyMFoam

```

1  #include "fvCFD.H"
2  #include "MULES.H"
3  #include "subCycle.H"
4  #include "incompressibleTwoPhaseMixture.H"
5  #include "turbulenceModel.H"
6  #include "dynamicFvMesh.H"
7  #include "pimpleControl.H"
8
9  // * * * * *
   * * * * * //
10
```

Figura 20: Modificaciones a twoLiquidMixingFoam



```

11  int main(int argc, char *argv[])
12  {
13      #include "setRootCase.H"
14      #include "createTime.H"
15      #include "createDynamicFvMesh.H"
16      #include "readGravitationalAcceleration.H"
17      #include "initContinuityErrs.H"
18      #include "createFields.H"
19      #include "readTimeControls.H"
20
21      pimpleControl pimple(mesh);
22
23      surfaceScalarField phiAbs("phiAbs", phi);
24      fvc::makeAbsolute(phiAbs, U);
25
26      #include "correctPhi.H"
27      #include "CourantNo.H"
28      #include "setInitialDeltaT.H"
29

```

```

30      // * * * * *
      * * * * * //
31
32      Info<< "\nStarting time loop\n" << endl;
33
34      while (runTime.run())
35      {
36          #include "readControls.H"
37          #include "CourantNo.H"
38          #include "alphaCourantNo.H"
39          #include "setDeltaT.H"
40
41          runTime++;
42
43          Info<< "Time = " << runTime.timeName() << nl << endl
              ;
44
45          scalar timeBeforeMeshUpdate = runTime.elapsedCpuTime
              ();
46
47          {
48              // Calculate the relative velocity used to map
              the relative flux phi
49              volVectorField Urel("Urel", U);
50
51              if (mesh.moving())
52              {
53                  Urel -= fvc::reconstruct(fvc::meshPhi(U));
54              }
55
56              // Do any mesh changes
57              mesh.update();
58          }
59
60          if (mesh.changing())
61          {
62              Info<< "Execution time for mesh.update() = "

```

```

63         << runTime.elapsedCpuTime() -
           timeBeforeMeshUpdate
64         << " s" << endl;
65
66         gh = g & mesh.C();
67         ghf = g & mesh.Cf();
68     }
69
70     if (mesh.changing() && correctPhi)
71     {
72         #include "correctPhi.H"
73     }
74
75     if (mesh.changing() && checkMeshCourantNo)
76     {
77         #include "meshCourantNo.H"
78     }

```

El primer cambio se observa en la línea 6 donde se incluye el archivo de cabecera "dynamicFvMesh.H" el cual contiene las funcionalidades necesarias para la manipulación de la malla dinámica en volúmenes finitos. En la línea 15 se sustituye el archivo "createMesh.H" por "createDynamicFvMesh.H" creando una entidad malla manipulable por el solucionador.

Otra modificación es la reubicación de la línea `pimpleControl pimple(mesh)`, la cual se ubica sobre las líneas que leen el número de Courant y determinan el tamaño del paso del tiempo, adicionalmente se agregan las líneas 23 y 24. En la línea 23 (`surfaceScalarField phiAbs ("phiAbs",phi)`) se crea el campo escalar en las caras `phiAbs`, el cual será el flujo absoluto en éstas y en la línea 24, `fvc::makeAbsolute (phiAbs,U)`, se usa la función de malla dinámica "*makeAbsolute*", la cual asigna el valor absoluto al flujo en las caras a partir del valor de la velocidad U y el valor de la velocidad de la frontera de la malla, es decir $U_a = U_r + U_b$, donde U_a representa la velocidad absoluta del flujo, U_r la velocidad relativa, y U_b es la velocidad de las fronteras del volumen de control. Después es necesario agregar la línea 26 "correctPhi.H" que incluye el archivo que asegura que los flujos máscicos sean conservativos, tiene una forma similar a "*pEqn.H*" (ver Anexo A). En *correctPhi.H* se crea la variable *pcorr*, con la cual se

ajustan los flujos en las fronteras que sean ajustables (ver Anexo A) se convierte en absoluto el flujo ϕ a partir de la velocidad U y cuando se inicie el ciclo PIMPLE se efectúa la corrección del flujo másico mediante la solución de la ecuación de Poisson en cada volumen de control con el uso de la variable *pcorr* mientras se realice la corrección de no ortogonalidad en la malla (ver Anexo A).

Luego del inicio del bucle temporal la línea 36 sustituye el archivo `"readTimeControls.H"` por `"readControls.H"`, donde se activan las banderas *correctPhi* como *true* porque se hará corrección de flujos y *checkMeshCourantNo* como *false*, dado que en este caso se trata con una malla con movimiento lineal de todos sus puntos, es decir, sin cambios topológicos, por lo tanto el número de Courant se mantiene invariable debido a que cada volumen de control mantiene su tamaño.

Se adiciona de la línea 45 a la línea 78, en estas líneas se toma el tiempo antes del cambio de la malla (línea 45), en la línea 49 se crea el campo vectorial en los volúmenes (**Urel**) para mapear en él el flujo relativo. Siempre que la malla se mueva (línea 51), se reconstruye el campo de velocidades a partir de los flujos en las caras (línea 53), en la línea 57 se actualiza la malla. Si hay cambios en la malla (línea 60), se calcula el tiempo involucrado en la actualización de esta y se presenta este tiempo en la pantalla (líneas 92-64), se actualiza el valor de la energía potencial del elemento en su centro (línea 66) y en sus caras (línea 67). Si la malla se mueve y la bandera *correctPhi* es *true* se efectúa la corrección de los flujos (línea 72). Si la malla cambia y la bandera *checkMeshCourantNo* es *true* se ejecuta verificación del número de Courant en la malla (línea 77), estas modificaciones garantizan la posibilidad de uso de la malla para movimiento automático y para cambios topológicos de la misma.

Por último se hacen los cambios necesarios en el archivo *pEqn.H*, como se puede ver en el Código 6.3:

Código 6.3: *pEqn.H*

```

1  {
2      volScalarField rAU("rAU", 1.0/UEqn.A());
3      surfaceScalarField rAUf("Dp", fvc::interpolate(rAU));
4
5      volVectorField HbyA("HbyA", U);
6      HbyA = rAU*UEqn.H();
7  }
```

```

8
9  //////////////////////////////////////
10
11  surfaceScalarField phiHbyA
12  (
13      "phiHbyA",
14      (fvc::interpolate(HbyA) & mesh.Sf())
15      + fvc::ddtPhiCorr(rAU, rho, U, phiAbs)
16  );
17  fvc::makeRelative(phiHbyA, U);
18  adjustPhi(phiHbyA, U, p_rgh);
19  fvc::makeAbsolute(phiHbyA, U);
20
21  phiAbs = phiHbyA;
22
23  surfaceScalarField phig
24  (
25      - ghf*fvc::snGrad(rho)*rAUf*mesh.magSf()
26  );
27
28  phiHbyA += phig;
29
30
31  //////////////////////////////////////
32  while (pimple.correctNonOrthogonal())
33  {
34      fvScalarMatrix p_rghEqn
35      (
36          fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
37      );
38
39      p_rghEqn.setReference(pRefCell, getRefCellValue(
40          p_rgh, pRefCell));
41
42      p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.
43          finalInnerIter())));

```

```

43         if (pimple.finalNonOrthogonalIter())
44         {
45             phi = phiHbyA - p_rghEqn.flux();
46
47             U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn
48                 .flux())/rAUf);
49             U.correctBoundaryConditions();
50         }
51     }
52
53     #include "continuityErrs.H"
54     ////////////////////////////////////////////para malla movil
55     phiAbs = phi;
56
57     // Hace los flujos relativos al movimiento de la malla
58     fvc::makeRelative(phi, U);
59     ////////////////////////////////////////////
60
61     p == p_rgh + rho*gh;
62
63     if (p_rgh.needReference())
64     {
65         p += dimensionedScalar
66         (
67             "p",
68             p.dimensions(),
69             pRefValue - getRefCellValue(p, pRefCell)
70         );
71         p_rgh = p - rho*gh;
72     }
73 }

```

Se introducen las siguientes modificaciones: línea 17, se hace el flujo **phiHbyA** relativo al movimiento de la malla, para su ajuste y nuevamente se convierte en Absoluto en la línea 19. En la línea 55 se le asigna al campo **phiAbs** el valor del campo **phi** y luego en la línea 58 se vuelve relativo el campo **phi** al movimiento de la malla.

7 RESULTADOS

En este capítulo se presentan los resultados de la comprobación del solver *twoLiquidMixingFoam* al compararlos con los resultados encontrados en el artículo Simulations of Axial Mixing of Liquids in a Long Horizontal Pipe for Industrial Applications [10], también se presentan los resultados arrojados por el solver desarrollado para malla móvil *twoLiquidMixingDyMFoam* y son comparados.

7.1. SIMULACIONES DE MEZCLADO AXIAL

Se escogió el artículo de Zhao, Derksen, y Gupta [10] para verificar el correcto funcionamiento del solver *twoLiquidMixingFoam* debido a que en este artículo se plantea la simulación numérica basada en CFD, usando el método de los volúmenes finitos, del comportamiento de mezclado de dos fluidos miscibles. Primero describen el modelo numérico y lo examinan con diferentes parámetros usando un tubo de 10 metros de largo y luego presentan los resultados de la simulación de un tubo de 100 metros, estiman la longitud de mezcla en la dirección axial y estos resultados son extrapolados a longitudes mayores.

7.1.1. Modelos y métodos numéricos empleados

En el artículo [10] las ecuaciones son presentadas en notación indicial, en él es usada la ecuación de convección difusión para un escalar α con una concentración Y_α , equivalente a la Ecuación 5.30,

$$\frac{\partial Y_\alpha}{\partial t} + u_i \frac{\partial Y_\alpha}{\partial x_i} = \frac{\partial}{\partial x_i} \left(D \frac{\partial Y_\alpha}{\partial x_i} \right)$$

La ecuación de Navier-Stokes para flujo incompresible usada en [10]

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial u_i}{\partial x_j} \right)$$

que corresponde a la Ecuación 5.52, y el flujo difusivo

$$J_{\alpha,i} = -\rho \left(D_\alpha + \frac{\nu_t}{Sc_t} \right) \frac{\partial Y_\alpha}{\partial x_i}$$

correspondiente con Ecuación 5.85, donde se usan $Sc_t = 0,7$, $D_\alpha = 0$. También es usado el modelo de turbulencia $k - \epsilon$ con una función de pared no especificada, donde $\nu_t = C_\mu \frac{k^2}{\epsilon}$, con las constantes $C_{1\epsilon} = 1,44$, $C_{2\epsilon} = 1,92$, $C_\mu = 0,09$, $\sigma_k = 1,0$, $\sigma_\epsilon = 1,3$. No se tiene en cuenta el término de generación de energía cinética turbulenta k , de forma que $G_b = g_i \frac{\nu_t}{Sc_t} \frac{\partial p}{\partial x_i} = 0$, en notación vectorial se tiene:

$$G_b = g \frac{\nu_t}{Sc_t} \nabla p = 0. \quad (7.1)$$

para la extrapolación a mayores longitudes usan la ecuación de la teoría de penetración para la longitud de mezcla de diferentes especies

$$\delta = \sqrt{\pi t D_t}$$

En esta simulación es usado el software privativo Fluent en su versión 6.3.

7.1.2. Modelo

Primero simulan el flujo totalmente turbulento en la tubería para un solo fluido, luego inician la simulación inyectando el segundo fluido y se corre la simulación por 4 segundos. Las condiciones del modelo se presentan en la Tabla 3, la Tabla 4 y la Tabla 5.

Los resultados para la simulación después de 4 segundos donde se muestra el inicio de la mezcla (90% de concentración) y el fin de la misma (10% de concentración) son presentados en la Tabla 6.

Tabla 3: Propiedades de los fluidos

propiedad	fluido A	fluido B
densidad	750 kg/m^3	523 kg/m^3
viscosidad	$0,3 \times 10^{-3} \frac{\text{kg}}{\text{m.s}}$	$0,048 \times 10^{-3} \frac{\text{kg}}{\text{m.s}}$
Re	30000	130000

Tabla 4: Características del dominio computacional

diámetro	longitud	velocidad	mallla transversal	mallla axial
12 mm	10m	1 m/s	20	1-2-4 mm

También listan resultados para diferentes temperaturas, variación de diámetro y variación de número de Reynolds.

Seguidamente presentan gráficas para fracción de masa, energía cinética turbulenta, epsilon y viscosidad cinética turbulenta.

7.2. SIMULACIÓN DE MEZCLADO AXIAL CON OpenFOAM

La siguiente simulación se realizó haciendo uso de la plataforma GRIDUIS-2 usando el cluster Guane de manera remota mediante conexión *ssh*. Inicialmente se resolvió el problema en una malla más gruesa y esta solución aproximada se mapea en la malla definitiva para poder así iniciar la simulación para 4 segundos de flujo.

7.2.1. Modelos y métodos numéricos empleados

Debido al tamaño computacional, y a su respectivo costo, en el experimento replicado con la malla estática se utilizó un dominio igual a la mitad del dominio, para verificar la malla usada en la réplica del caso, se usó una herramienta de integridad de malla de OpenFOAM y se observó que ésta presenta una no ortogonalidad del 26%, valor bastante bajo para el cual solo fue necesario aumentar el numero de pasos correctores de no ortogonalidad a 1 y así garantizar la convergencia de la solución.

En la simulación en OpenFOAM se usan las mismas ecuaciones usadas en el modelo de Zhao [10], convección difusión para un escalar y Navier-Stokes. Para el modelo turbulento también es usado $k - \epsilon$ con funciones de pared `nutWallFunction` la cual suministra condición de viscosidad cinemática turbulenta, `epsilonWallFunction` esta condición de

Tabla 5: Condiciones de frontera

campo	entrada	salida
velocidad	uniforme	sin variación axial

Tabla 6: Resultados Zhao para 4 segundos

caso número	longitud de malla(mm)	Δt	0.9 (m)	0.1 (m)	longitud de mezcla
1	1	0.01	3.68	4.25	0.57
2	1	0.02	3.59	4.33	0.74
3	1	0.03	3.51	4.42	0.91
4	2	0.005	3.74	4.21	0.47
5	2	0.01	3.68	4.28	0.60
6	2	0.015	3.63	4.32	0.69
7	2	0.02	3.58	4.33	0.75
8	2	0.03	3.5	4.43	0.93
9	4	0.01	3.74	4.39	0.65
10	4	0.02	3.55	4.36	0.81
11	4	0.03	3.5	4.45	0.95

frontera entrega una función para disipación en la frontera, aplicada a las fronteras tipo pared, la cual calcula ϵ y G e inserta los valores directamente en la ecuación de ϵ actuando como una restricción, `kqRWallFunction`, da gradiente cero en la pared para la energía cinética turbulenta k . Se usan las mismas constantes del modelo turbulento, pero se incluye el cálculo de G .

7.2.2. Modelo

Son usadas las mismas propiedades de los fluidos listadas en la Tabla 3. La Tabla 7 describe el modelo computacional resuelto en Guane.

Tabla 7: Dominio computacional

diámetro	longitud	velocidad	mallla transversal	mallla axial
12 mm	5m	1 m/s	20	4 mm

7.2.2.1. Condiciones de frontera

En la tabla se resumen las condiciones de frontera usadas en la verificación

Tabla 8: Condiciones de frontera

campo	entrada	salida
velocidad	uniforme	sin variación axial
p_rgh	zeroGradient	101325 kPa
k	fixedValue	inletOutlet (valor calculado)
épsilon	fixedValue	inletOutlet (valor calculado)

7.2.2.2. Esquemas numéricos

En la Tabla 9 y la Tabla 10, se resumen los esquemas numéricos usados

Tabla 9: Esquemas numéricos

término	nombre	tipo
derivada temporal	Euler	discretización implícita
gradiente	Gauss linear	integración gaussiana con interpolación lineal (IGIL)
laplaciano	Gauss linear corrected	IGIL con corrección explícita de no ortogonalidad
interpolaciones	linear	diferencia centrada

Tabla 10: Esquemas numéricos términos convectivos

término	nombre	tipo
$\phi \cdot u$	Gauss linear	integración gaussiana con diferencia centrada
α_1	Gauss linear	integración gaussiana diferencia centrada
$\epsilon - k$	Gauss limitedlinear 1	diferencia centrada limitada
$\text{div}((\mu_{\text{eff}} \cdot \text{dev}(\nabla(U))))$	Gauss linear	integración gaussiana con diferencia centrada

7.2.2.3. Resultados twoLiquidMixingFoam

Una vez corrido el caso para malla estática con el solver *twoLiquidMixingFoam* se obtuvo la solución y se generaron las gráficas que permiten observar el comportamiento de las variables del flujo. La extensión y el perfil de concentraciones obtenido para una línea a lo largo del eje de la tubería se puede observar en la Figura 21, la cual muestra en el

eje de las abscisas la distancia en metros, mientras que el eje de las ordenadas muestra la fracción másica de la sustancia (α_1).

Figura 21: Resultados malla Zhao, 20*4mm

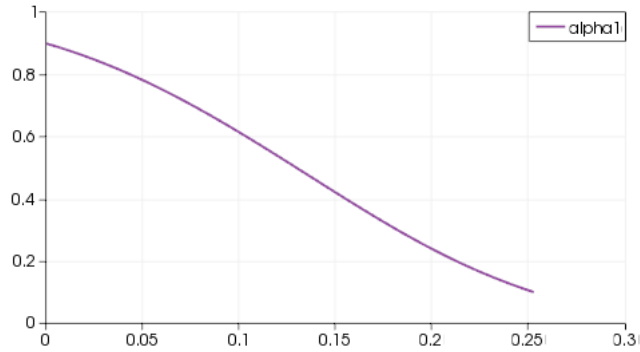


Tabla 11: Resumen de resultados

caso	tiempo	delta t (s)	0.9 (m)	0.15 (m)	longitud
1	4 segundos	0.0015873	3.885	4.1375	0.2525

Al igual que en el trabajo de Zhao, también se muestran los perfiles en una línea transversal al conducto para las cantidades turbulentas energía cinética turbulenta k , velocidad U , épsilon ϵ , y fracción másica a 4m de la entrada y a 4 segundos de simulación en la Figura 22, la Figura 25, la Figura 23 y la Figura 24

Figura 22: k a 4 metros de la entrada

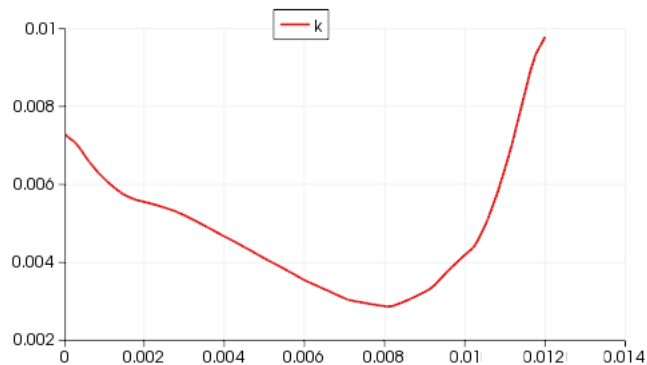


Figura 23: ϵ a 4m de la entrada

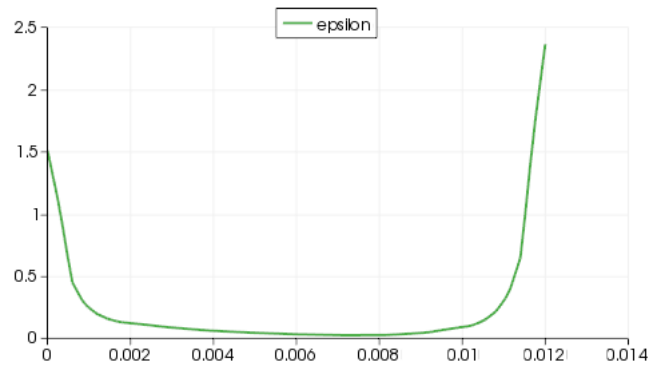


Figura 24: α_1 a 4m de la entrada

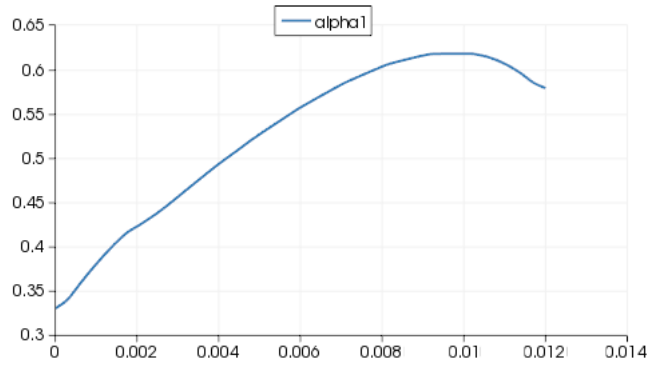
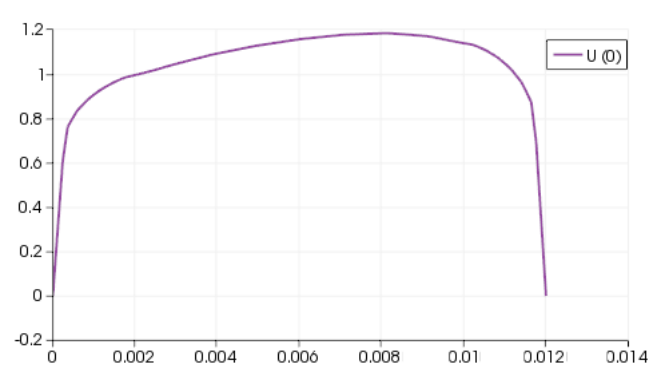


Figura 25: Perfil de velocidad a 4m

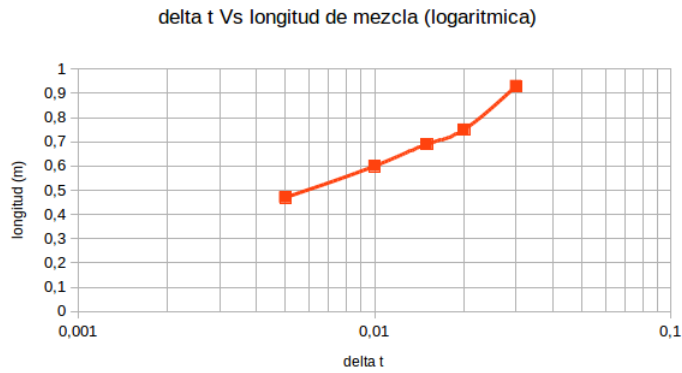


7.3. ANÁLISIS DE RESULTADOS twoLiquidMixingFoam

7.3.1. Relación Δt -longitud de mezcla

A partir de los resultados obtenidos se puede encontrar que el comportamiento del tamaño de mezcla para esta malla es dependiente del valor de delta de tiempo empleado, cabe anotar que este valor es restrictivo en la simulación realizada con OpenFOAM, debido a la dependencia entre delta t y la malla a través del número de Courant. En los datos obtenidos en [10] .

Figura 26: Longitud de mezcla 2mm



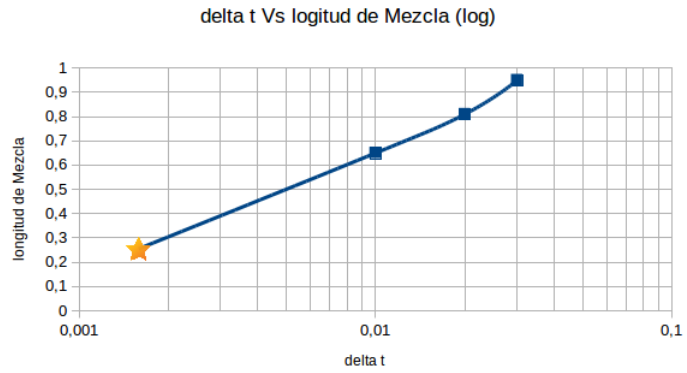
En la Figura 26 se puede ver el aumento que presenta la longitud de mezcla respecto a un aumento del delta de tiempo en segundos usado en la simulación, de forma que si se toman los datos en una escala lineal-logarítmica, se muestra que el crecimiento de longitud de mezcla corresponde con una función potencial para tiempos menores a 0,02 segundos.

Para comparar el resultado obtenido se toman los valores encontrados en [10] para 4 mm y el resultado obtenido con OpenFOAM para el mismo tamaño, el cual se marca con una estrella, en una escala lineal logarítmica, ver la Figura 27.

En la Figura 27 se puede observar el mismo comportamiento observado en la Figura 26, donde se puede apreciar el crecimiento con una función potencial en los primeros 3 puntos, los cuales corresponden a delta t de 0,005 s (OpenFOAM) , 0,01s y 0,02 s respectivamente.

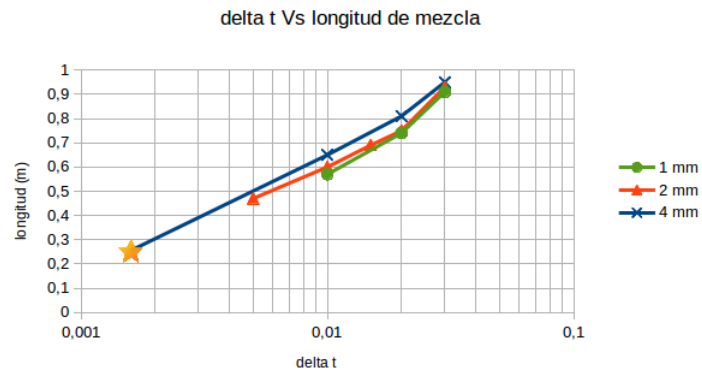
Si se compara el comportamiento del Δt versus la longitud de mezcla para 1,2 y 4 mm se puede observar en la Figura 28 que existe una relación potencial entre la variación

Figura 27: Longitud de mezcla 4mm



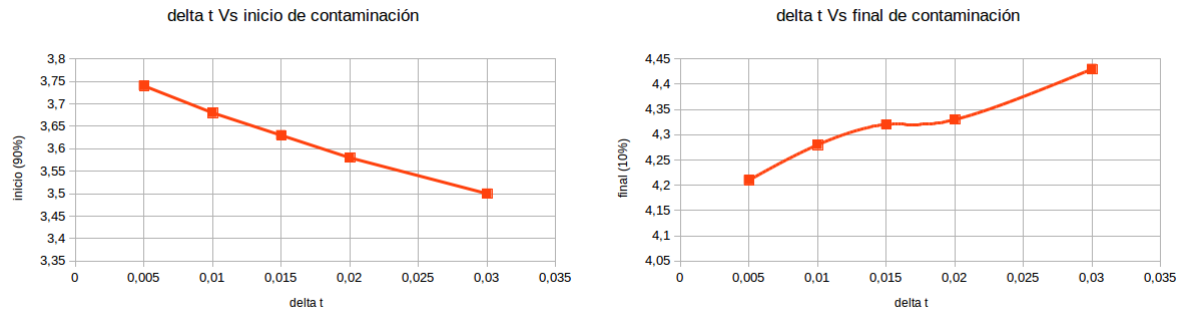
de el Δt y el decrecimiento de la longitud de mezcla en tiempos menores a 0,02 s en las simulaciones.

Figura 28: Δt Vs longitud para 1,2 y 4mm



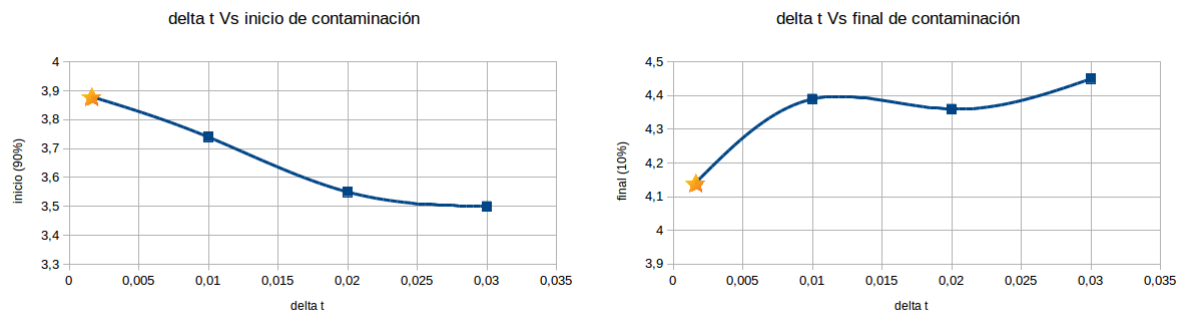
Para el inicio y el fin de la longitud de mezcla para un tamaño de malla de 2mm en la simulación hecha en [10], se tiene lo presentado en la Figura 29.

Figura 29: Inicio y fin de mezcla 2 mm



Se hace lo mismo para 4 mm, se toman los datos obtenidos por Zhao y se agrega el resultado obtenido con OpenFOAM, ver la Figura 30.

Figura 30: Inicio y fin de mezcla 4mm



Una comparación de la Figura 29 y la Figura 30 muestra un comportamiento similar en el inicio de la longitud de mezcla y el final de la misma para ambos casos, aunque la curvatura para las gráficas de 4 mm es mayor, ambas presentan tendencias similares.

Mediante la comparación de los resultados que se encuentran en la literatura y los resultados obtenidos con OpenFOAM se puede concluir que la malla, el solucionador, los esquemas numéricos y los parámetros de solución son los adecuados para la simulación del problema involucrado en el transporte tipo batch de dos fluidos, en donde el fenómeno analizado involucra la mezcla de dos fluidos incompresibles miscibles que se desplazan por una tubería, simulación en la cual se puede apreciar la concordancia con las tendencias y comportamientos de las concentraciones con los descritos en la literatura citada.

7.3.2. Variables de flujo

Los perfiles de las variables de flujo mostrados de la Figura 22 a la Figura 24, ponen en evidencia el efecto apreciable de la gravedad al modificar la simetría de los mismos, contrario a lo presentado en [10], donde no se aprecia variación significativa en los perfiles de las mismas cantidades, esto se puede deber al efecto de la generación de energía cinética turbulenta G , ya que en la simulación hecha en [10] este término se eliminó, *Ecuación 7.1*. La función de pared escogida en la réplica del experimento de Zhao et al., satisface los requerimientos del flujo, ya que el perfil de velocidades presenta altos gradientes en los elementos cercanos a la pared, donde las ecuaciones del flujo son emuladas y son resueltas en las demás zonas del dominio, de manera que la forma del perfil es congruente con un perfil turbulento.

7.4. RESULTADOS CON *twoLiquidMixingDyMFoam*

Para probar el solver *twoLiquidMixingDyMFoam* primero se resolvió el caso para la mezcla de dos fluidos en dos dimensiones para un dominio de 15 metros y 12mm con una malla de 20 elementos transversales y 10 mm de longitud axial durante 5 segundos, la solución de los campos luego de 5 segundos es mapeada en un dominio de 10 metros y 12 mm con una malla de 20 elementos transversales y 10 mm de longitud, mapeando los campos los 10 metros desde la entrada hasta la salida y solucionando el caso con la misma configuración de esquemas numéricos con el solver para malla dinámica y con el solver para malla estática por 9 segundos.

En el caso de malla dinámica la inestabilidad de la solución aumenta, por ello para ambos casos se usó un número de Courant más bajo (0.1) y se empleó el bucle corrector externo PIMPLE `nOuterCorrectors` para mejorar la convergencia de la solución y aumentar `nAlphaSubCycles` a 2, además de emplear factores de relajación para las soluciones de las ecuaciones y los campos (Ver Anexo).

7.4.1. Esquemas numéricos

Para las simulaciones estáticas se usaron los mismos esquemas numéricos usados la simulación con malla móvil, estos esquemas numéricos son listados en la Tabla 12 y la Tabla 13.

Tabla 12: Esquemas numéricos

término	nombre	tipo
derivada temporal	CrankNicolson 0.8	CrankNicolson y backward mezclados
gradiente	Gauss linear	integración gaussiana con interpolación lineal
laplaciano	Gauss linear corrected	IGIL con corrección explícita de no ortogonalidad
interpolaciones	linear	diferencia centrada

Tabla 13: Esquemas numéricos términos convectivos

término	nombre	tipo
$\phi \cdot u$	Gauss QUICK	integración gaussiana con diferencia QUICK
α_1	Gauss linear	integración gaussiana diferencia centrada
$\epsilon - k$	Gauss QUICK	integración gaussiana con diferencia QUICK
$\text{div}((\mu_{\text{eff}} \cdot \text{dev}(\nabla(U))))$	Gauss linear	integración gaussiana con diferencia centrada

7.4.2. Caso 2D 15 metros estático

Los resultados obtenidos después de la simulación estática se presentan en gráficas que facilitan su interpretación, los perfiles de concentración para 5, 6, 7, 8, 9 y 10 segundos a lo largo de una línea en el eje de la tubería en la zona de mezcla se muestran en la Figura 31, la Figura 32 y la Figura 33.

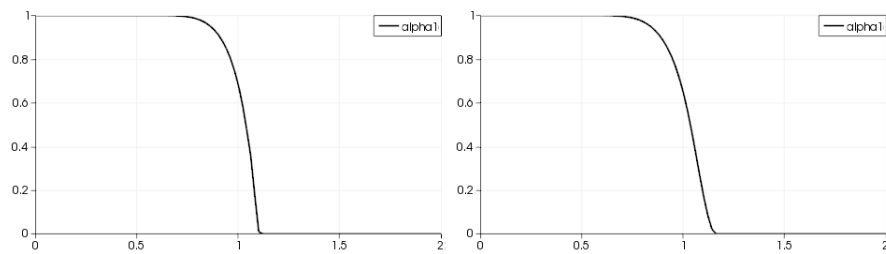
Figura 31: Perfiles 5 y 6 segundos

Figura 32: Perfiles 7 y 8 segundos

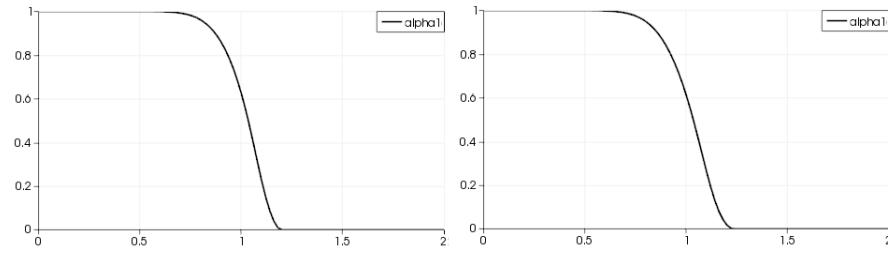
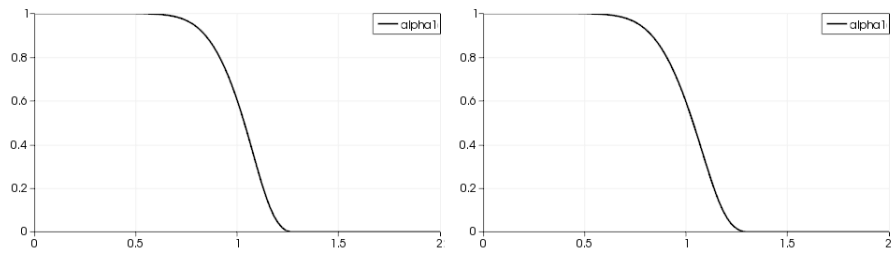


Figura 33: Perfiles a 9 y 10 segundos



7.4.3. Caso 2D 10 metros dinámico

Luego de simular el caso con malla dinámica se obtuvieron los resultados para los perfiles de concentración en 5, 6, 7, 8, 9 y 10 segundos a lo largo de una línea de 2 metros en el eje de la tubería en la zona de mezcla que se muestran en la Figura 34, la Figura 35 y la Figura 36.

Figura 34: Perfiles 5 y 6 segundos

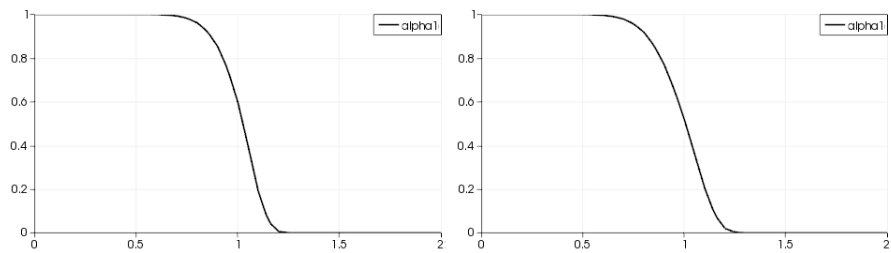


Figura 35: Perfiles 7 y 8 segundos

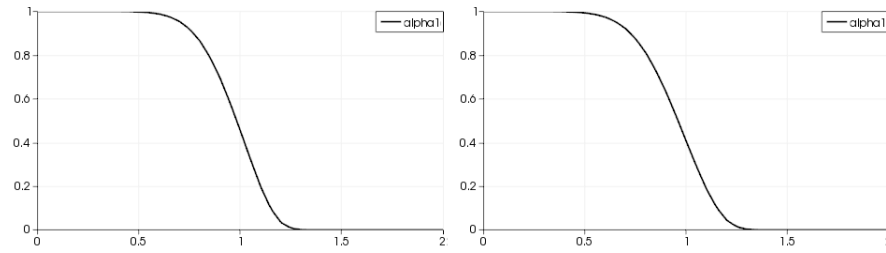
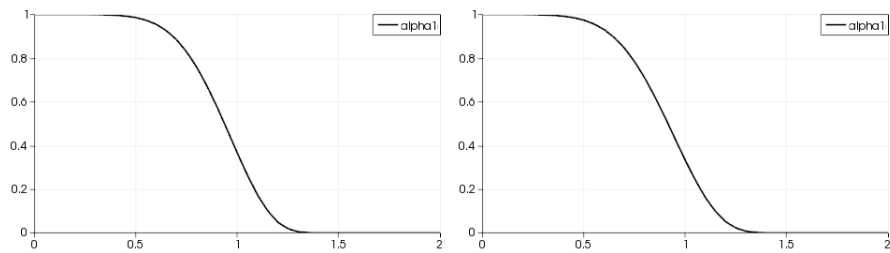


Figura 36: Perfiles 9 y 10 segundos



El resumen de resultados se presenta en la Tabla 14.

7.5. ANÁLISIS DE RESULTADOS twoLiquidMixingDyMFoam

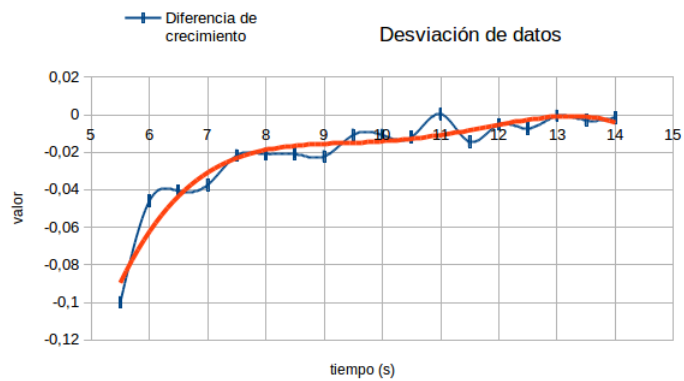
7.5.1. Diferencia de longitud de mezcla en el tiempo

Se puede observar una tendencia decreciente en el aumento de la diferencia de la longitud de mezcla a medida que avanza la solución ver Figura 37, la diferencia de este valor entre pasos del tiempo tiene un aumento drástico al inicio pero este aumento se hace menor a medida que avanza la solución (Tabla 14).

Tabla 14: Resumen de resultados malla dinámica

tiempo (s)	inicio estático	inicio dinámico	fin estático	fin dinámico	distancia estática	distancia dinámica	desviación %
5	4,865	4,865	5,058	5,058	0,193	0,193	0
5,5	5,36	5,395	5,57	5,626	0,21	0,231	10
6	5,85	5,865	6,083	6,132	0,233	0,267	14,592
6,5	6,34	6,34	6,592	6,639	0,252	0,299	18,650
7	6,832	6,815	7,1	7,143	0,268	0,328	22,388
7,5	7,325	7,292	7,61	7,647	0,285	0,355	24,561
8	7,815	7,768	8,115	8,148	0,3	0,38	26,666
8,5	8,31	8,247	8,623	8,65	0,313	0,403	28,753
9	8,803	8,723	9,129	9,15	0,326	0,427	30,981
9,5	9,295	9,203	9,635	9,652	0,34	0,449	32,0588
10	9,79	9,681	10,143	10,151	0,353	0,47	33,144
10,5	10,322	10,161	10,686	10,65	0,364	0,489	34,340
11	10,815	10,641	11,191	11,146	0,376	0,505	34,308
11,5	11,312	11,12	11,698	11,644	0,386	0,524	35,751
12	11,768	11,6	12,165	12,141	0,397	0,541	36,272
12,5	12,3	12,08	12,708	12,639	0,408	0,559	37,009
13	12,757	12,563	13,175	13,136	0,418	0,573	37,081
13,5	13,252	13,045	13,68	13,633	0,428	0,588	37,3831
14	13,748	13,526	14,185	14,127	0,437	0,601	37,528

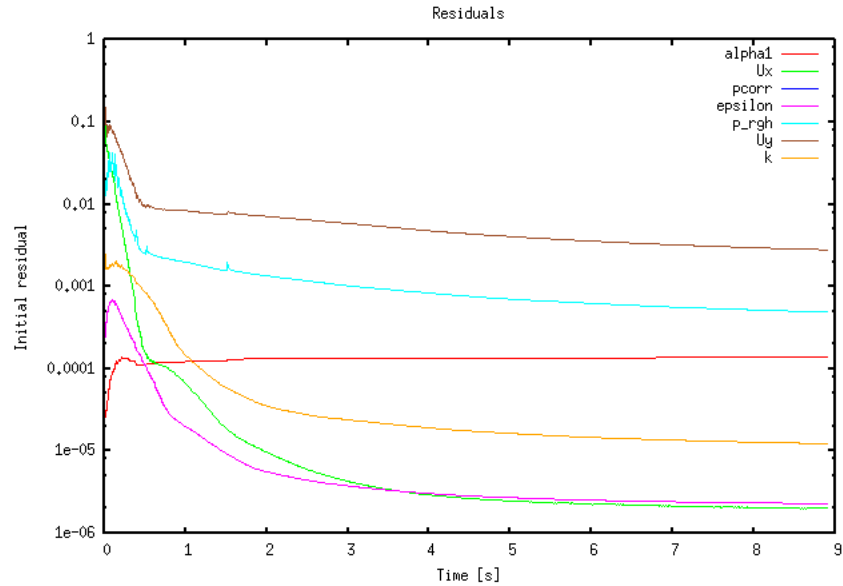
Figura 37: Crecimiento de la variación de longitud



También se puede apreciar la relación existente entre esta disminución y la disminución del valor de los residuales obtenidos en cada Δt en la simulación de malla Dinámica, el aumento inicial de los residuales se da debido a la forma como se ejecuta la solución

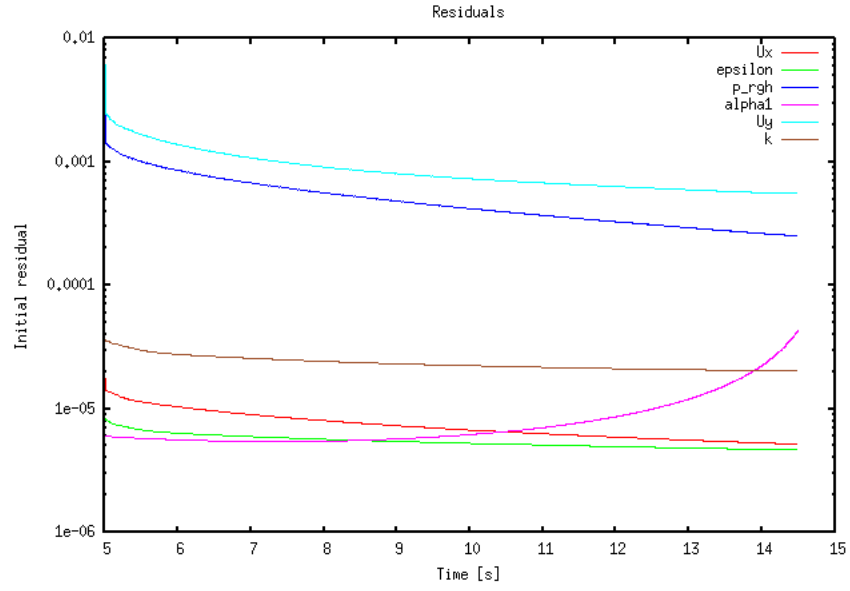
con malla dinámica, en la cual se mapean los campos de una solución estática para garantizar la estabilidad de la solución con malla dinámica. Para ejecutar la solución del modelo de turbulencia OpenFOAM lee la carpeta de tiempo de inicio establecido en *controlDict* y reemplaza los archivos que se encuentran para las variables turbulentas asignándoles a los archivos antiguos la extensión *.old* para así poder crear unos nuevos archivos con las valores asignados a estas variables en la carpeta *0.org* y las constantes del modelo turbulento, y de esta forma poder calcular su valor como si fuera el instante 0. Debido a lo anteriormente expuesto aparecen residuales altos en las iteraciones de los segundos iniciales (Ver la Figura 38).

Figura 38: Residuales malla Dinámica



A diferencia de lo visto anteriormente, durante el proceso de solución en la solución con malla estática, OpenFOAM toma los valores que aparecen en la carpeta temporal y por tanto los valores del campo de flujo no se ven alterados, ver Figura 39.

Figura 39: Residuales de malla estática



El aumento que se aprecia en el valor de α_1 (α_1) (ver Figura 39) en la solución con malla estática se da debido a que la fracción volumétrica varía en la malla estática porque el fluido impulsado está abandonando la tubería empujado por la entrada del fluido impulsor, situación que no se presenta en la malla dinámica, donde los residuales de α_1 (α_1) se mantienen constantes a lo largo del tiempo debido a que la variación de ésta es más o menos uniforme en el tiempo.

8 CONCLUSIONES Y RECOMENDACIONES

8.1. CONCLUSIONES

- Se realizó la recopilación sistemática de los pilares teóricos que enmarcan el problema de investigación, permitiendo contextualizarlo y escoger los modelos matemáticos y físicos usados en su implementación, también se hizo posible la escogencia del experimento usado para comparar los resultados obtenidos con el modelo desarrollado.
- El modelo matemático desarrollado involucró las ecuaciones de conservación de las especies químicas y la conservación de la cantidad de movimiento, Ecuaciones de Navier-Stokes, en tanto que no fue necesario el uso de la ecuación de conservación de la energía. Esto se debió a las condiciones del modelo, el cual se consideró isotérmico, lo que implicó que no existiera acoplamiento entre la temperatura y la densidad.
- En la solución para la malla dinámica se usaron esquemas de interpolación de mínimo segundo orden acotados, ya que los términos convectivos de variables turbulentas presentaban inestabilidad en esquemas de orden inferior. Para este caso se usó el esquema QUICK que puede llegar a ser de tercer orden de exactitud en mallas uniformes como es el caso en la dirección del flujo, este esquema es condicionalmente estable, en esencia no acotado, aunque en la implementación del mismo en OpenFOAM se describe acotado, además debido a inestabilidades en la solución fue necesario el uso de factores de relajación en la solución de las variables para mantener su acotación. En el caso de malla estática no se presentaba esta inestabilidad y bastó con esquemas TDV con interpolación centrada limitada *limitedLinear* para las variables turbulentas, en tanto que para las variables del flujo solo fue necesario el uso de interpolación *linear*, esta equivale a interpolación centrada, que es de segundo orden de exactitud pero no acotada, con la cual se lograron buenos resultados.
- El algoritmo PISO presentó bastante estabilidad para el caso de simulación con malla estática, pero cuando se trabajo en la implementación de la solución con

el uso la funcionalidad de malla dinámica, se debió recurrir al algoritmo PIMPLE, ya que este cuenta con un bucle corrector que permite la verificación de la conservación de las variables para cada paso del tiempo y así aumenta la convergencia de la solución, además fue necesaria la disminución del paso del tiempo para garantizar la estabilidad de la solución disminuyendo el número de Courant a 0,1.

- Se encontró un aumento en el desempeño en la solución al usar hasta 4 procesadores en la solución multiprocesador, al usar más, el tiempo de solución aumenta, llegando a duplicarse en algunos casos.
- La desviación entre los datos obtenidos por el solucionador de malla dinámica y el solucionador de malla estática esta directamente influenciada por la forma como OpenFOAM resuelve el modelo turbulento cuando empieza a resolver un caso, creando nuevos archivos para las variables turbulentas, lo cual produce variaciones iniciales en el flujo debido a altos residuales que se presentan en la iteraciones iniciales de la solución de las ecuaciones, que inducen cambios notables en los campos.

8.2. RECOMENDACIONES

- Es necesario ahondar en el estudio de la influencia de los esquemas numéricos y los parámetros de solución (factores de relajación, pasos correctores, algoritmos de solución) para su uso en la malla dinámica, pues esta funcionalidad demostró ser muy sensible a la modificación de los mismos, además es necesario profundizar en la investigación sobre la variación inicial de los residuales de la solución con malla dinámica y reducir su efecto para así conseguir soluciones mas exactas que permitan modelar longitudes mayores.
- Se consiguió simular malla dinámica para un caso 2D, pero es conveniente estudiar su implementación en casos 3D, valiéndose de la investigación en los métodos, parámetros y esquemas usados, ya que se intentó resolver casos en 3D con la misma configuración numérica y no fue posible encontrar una solución debido principalmente a la inestabilidad de las variables turbulentas.
- La solución en el cluster Guane se debió hacer de manera interactiva, ya que las variables de entorno no se cargan en el ambiente cuando el despliegue se hace de

forma pasiva, esto implica inconvenientes debido a los tiempos límite de uso interactivo del cluster, los cuales son muy inferiores al tiempo de solución necesario. Se recomienda indagar sobre la posibilidad de la modificación del despliegue pasivo en Guane para incluir las variables de entorno de OpenFOAM y así aumentar los tiempos de solución para que los cálculos no se vean interrumpidos y no se pierda tiempo de cómputo.

- La determinación del número óptimo de procesadores involucrados en la solución del problema estudiado podría reducir el tiempo de cálculo de forma conveniente.
- Profundizar el conocimiento sobre las funciones de pared implementadas en OpenFOAM y de los modelos de turbulencia disponibles, para su aprovechamiento y uso óptimo en el caso estudiado y aplicaciones futuras.

REFERENCIAS BIBLIOGRÁFICAS

- [1] ECOPETROL, Empresa Colombiana de Petroleos. Programa de control de calidad para combustibles transportados por poliducto. Bogota D.C.: ECOPETROL, 1990.
- [2] CONCHA DEMOYA, Alexander, QUIROGA CRUZ, Hugo y BENJUMEA HERNANDEZ, Pedro Nel. Modelamiento de la propagación de los frentes de contaminación generados por el transporte de combustibles por poliducto. En: Dyna. 2007, vol. 74, p. 89-96.
- [3] ECOPETROL, Empresa Colombiana de Petroleos. ECOPETROL en datos Operativos historicos 2000-2007 [online]. Disponible desde: http://www.ecopetrol.com.co/documentos/41153_08-03-11-_MIS-ENEDIC-2007.xls.
- [4] TAYLOR, Geoffrey. Dispersion of soluble matter in solvent flowing slowly through a tube. En: Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences. 1953, vol. 219, no. 1137, p. 186-203.
- [5] TAYLOR, Geoffrey. The dispersion of matter in turbulent flow through a pipe. En: Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences. 1954, vol. 223, no. 1155, p. 446-468.
- [6] ARIS, Rutherford. On the dispersion of a solute in a fluid flowing through a tube. En: Proceedings of the Royal Society of London. Series A. 1956, vol. 235, no. 1200, p. 67-77.
- [7] SJENITZER, F. How much do product Mix in a pipeline. Royal Dutch/ SHELL group, 1958.
- [8] EKAMBARA, Kalekudithi y OSHI, Jyeshtharaj Bhalchandra. Axial mixing in pipe flows: turbulent and transition regions. En: Chemical engineering science. 2003, vol. 58, no. 12, p. 2715-2724.
- [9] KRANTZ, William B y WASAN, Darsh T. Axial dispersion in the turbulent flow of power-law fluids in straight tubes. En: Industrial & Engineering Chemistry Fundamentals. 1974, vol. 13, no. 1, p. 56-62.

- [10] ZHAO, Lingling, DERKSEN, Jos y GUPTA, Rajender. Simulations of axial mixing of liquids in a long horizontal pipe for industrial applications. En: Energy & Fuels. 2010, vol. 24, no. 11, p. 5844–5850.
- [11] MARTINS BAPTISTA, Renan, BASTOS DE FREITAS RACHID, Felipe y CARNEIRO DE ARAUJO, Jose Henrique. Mixing Volume Evaluation Under Pumping Shut-down Conditions. Salt Lake City, Utah: PSIG Annual Meeting, 2001. p. 13.
- [12] Cámara Colombiana de Servicios Petroleros, CAMPETROL, . La Producción de los Distintos Tipos de Crudo en Colombia [online]. Septiembre 23.2009 [fecha de citacion]. Disponible desde: <http://campetrol.org/el-sector/estadisticas/cifras-y-datos/>.
- [13] Agencia Nacional de Hidrocarburos, . Indicadores a enero 30 de 2011 [online] [online]. Disponible desde: http://www.anh.gov.co/media/indicadores/indicadores_ene_11_web.xls.
- [14] BASTOS DE FREITAS RACHID, Felipe, CARNEIRO ARAUJO, Jose Henrique y MARTINS BAPTISTA, Renan. Predicting Mixing Volumes in Serial Transport in Pipelines. En: transactions of the ASME, Journal of Fluids Engineering. Junio, 2002, vol. 124, p. 528-534..
- [15] SJENITZER, F. How much do products mix in a pipeline?. En: The Pipeline Engineer. Diciembre, 1958, p. D31-D34.
- [16] ZARATE, Neira y EDDY, Ricardo. Análisis de frentes de contaminación en el transporte de hidrocarburos líquidos por tubería. En: Revista ingeniería e investigación. 2003, vol. 52, p. 31-36.
- [17] ÁLVAREZ, Juan DÍAZ, ÁLVAREZ, Fernando CORZO, GONZALEZ, Ariel FLÓREZ et al.. Análisis estadístico en la predicción del volumen de contaminación generado en el transporte de lotes sucesivos de hidrocarburos a través de poliductos. En: Revista ION. 2011, vol. 24, no. 1, p. 23-33.
- [18] LIU, Mou B, XIE, W.P. y Liu, G.R.. Modeling incompressible flows using a finite particle method. En: Applied mathematical modelling. 2005, vol. 29, no. 12, p. 1252-1270.
- [19] JASAK, Hovre. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flow. Tesis Doctoral. Londres: Department of Mechanical Engineering, Imperial College of Science, Technology, and Medicine. 1996.

- [20] PATANKAR, Suhas V. y SPALDING, D. Brian. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. En: journal of heat and mass transfer. 1972, vol. 15, p. 1787-1806.
- [21] ANDERSON, John D. Jr.. Computational Fluid Dynamics The Basics with Applications. 1221 Avenue of the americas, New York, NY 10020: McGraw-Hill, Inc., 1995. 547 p.
- [22] FUENTES DÍAZ, David Alfredo. Mecánica de fluidos computacional. Bucaramanga: Universiad Industrial de Santander, Escuela de ingeniería mecánica, 2009. 352 p.
- [23] BIRD, Warren R. y BYRON, E. Stewart. Transport Phenomena. Nueva York: John Wiley & Sons, Inc., 2005. 905 p.
- [24] WELLER, Henry G., TABOR, Gavin, JASAK, Hrvoje et al.. A tensorial approach to computational continuum mechanics using object-oriented techniques. En: Computer in phisics. Noviembre-Diciembre, 1998, vol. 12, no. 6, p. 620-631.
- [25] ANIL, Date W. Solution of transport equations on unstructured meshes with cell-centered colocated variables. Part I: Discretization. En: International Journal of Heat and Mass Transfer. 2005, vol. 48, p. 1117-1127.
- [26] RUSCHE, Henrik. Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions. Tesis Doctoral. Londres: Imperial College of Science, Technology & Medicine, Department of Mechanical Engineering. 2002.
- [27] JASAK, Hrvoje. Dynamic Mesh Handling in OpenFOAM. Orlando, Florida: 2008 (AIAA 2009-341).
- [28] FUENTES, David, SÁNCHEZ, Ronald y DÍAZ, Pedro. Cálculo del flujo difusivo en dominios complejos mediante el metodo de volúmenes finitos. En: UIS Ingenierías. junio, 2012, vol. 11, no. 1, p. 45-54.
- [29] ISSA, Raad. Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting. En: Journal of Computational Physics. 1986, vol. 62, p. 40-65.
- [30] Python Software Foundation, . Documentación para el lenguaje Python [online]. 09.2013 [fecha de citacion]. Disponible desde: <http://www.python.org/doc>.
- [31] OPEN CASCADE, . Documentación modulo malla Plataforma SALOME, OPEN CASCADE, 1 Place des FrÃšres Montgolfier, 78280 Guyancourt, Francia [online].

- 2013 [fecha de citacion]. Disponible desde: http://docs.salome-platform.org/latest/SMESH_index.html.
- [32] OPEN CASCADE, . Documentación modulo geometria plataforma SALOME. OPEN CASCADE, 1 Place des Frères Montgolfier, 78280 Guyancourt, Francia [online]. 2013 [fecha de citacion]. Disponible desde: http://docs.salome-platform.org/latest/GEOM_index.html.
- [33] GSCHAIER, Bernhard y Comunidad, . PyFoam, libreria de soporte para trabajo con OpenFOAM y sus forks [online]. 2013 [fecha de citacion]. Disponible desde: <http://openfoamwiki.net/index.php/Contrib/PyFoam>.
- [34] JASAK, Hrvoje, WELLER, Henry G. y NORDIN, Niklas. In-cylinder CFD simulation using a C++ object-oriented toolkit. En: Society of automotive engineers SAE SP. 2004, p. 81–188.
- [35] OpenFOAM Foundation, . OpenFOAM The Open Source CFD Toolbox User Guide [online]. 2013 [fecha de citacion]. Disponible desde: <http://cfd.direct/openfoam/user-guide/>.
- [36] DEMIRDZIC, I. y PERIC, Milovan. Space conservation law in finite volume calculations of fluid flow. En: International journal for numerical methods in fluids. 1988, vol. 8, p. 1037-1050.
- [37] THOMAS, Patrick D. y LOMBARD, Charles K.. Geometric Conservation Law and Its Application to Flow Computations on Moving Grids. En: AIAA Journal. October, 1979, vol. 17, no. 78-1208R, p. 1030-1037.
- [38] FERZIGER, Joel H. y PERIC, Milovan. Computational Methods for Fluid Dynamics. Berlin: Springer-Verlag, 2002. 430 p.
- [39] PATANKAR, Suhas V.. Numerical Heat Transfer and Fluid Flow. New york: McGraw-Hill Book Company, 1980. 197 p.

ANEXO A. DESCRIPCIÓN DE twoLiquidMixingFoam.C

Encabezado:

Luego de información referente a la licencia y una descripción breve del solucionador se incluyen los archivos de encabezado, estos archivos contienen código fuente con funciones fundamentales

Código 1: Encabezado

```
1  /*-----*\
2  =====|
3  \\      /Field      | OpenFOAM: The Open Source CFD Toolbox
4  \\      / Operation |
5  \\      / And       | Copyright (C) 2011–2013 OpenFOAM Foundation
6  \\      / Manipulation|
7  -----|
8  License
9      This file is part of OpenFOAM.
10
11     OpenFOAM is free software: you can redistribute it and/or modify
12     it
13     under the terms of the GNU General Public License as published by
14     the Free Software Foundation, either version 3 of the License, or
15     (at your option) any later version.
16
17     OpenFOAM is distributed in the hope that it will be useful, but
18     WITHOUT
19     ANY WARRANTY; without even the implied warranty of
20     MERCHANTABILITY or
21     FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
22     License
23     for more details.
24
25     You should have received a copy of the GNU General Public License
26     along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
```

```

24 Application
25     twoLiquidMixingFoam
26
27 Description
28     Solver for mixing 2 incompressible fluids.
29
30     Turbulence modelling is generic , i.e. laminar , RAS or LES may be
31     selected .
32
33 \*-----*/
34 #include "fvCFD.H"
35 #include "MULES.H"
36 #include "subCycle.H"
37 #include "incompressibleTwoPhaseMixture.H"
38 #include "turbulenceModel.H"
39 #include "pimpleControl.H"
40
41 // * * * * *
42     * * * * * //

```

por ejemplo:

```

1  #include "fvCFD.H"

```

el archivo "fvCFD.H" contiene herramientas fundamentales para desarrollar cualquier calculo de volúmenes finitos. Este archivo enmarca un grupo de archivos, cada uno de los cuales es un elemento fundamental de la técnica de los volúmenes finitos. En este archivo se lee:

Código 2: fvCFD.H

```

1  #include "parRun.H"
2
3  #include "Time.H"
4  #include "fvMesh.H"
5  #include "fvc.H"
6  #include "fvMatrices.H"
7  #include "fvm.H"
8  #include "linear.H"
9  #include "uniformDimensionedFields.H"

```

```

10  #include "calculatedFvPatchFields.H"
11  #include "fixedValueFvPatchFields.H"
12  #include "adjustPhi.H"
13  #include "findRefCell.H"
14  #include "constants.H"
15
16  #include "OSSpecific.H"
17  #include "argList.H"
18  #include "timeSelector.H"
19
20  #ifndef namespaceFoam
21  #define namespaceFoam
22      using namespace Foam;
23  #endif
24
25  #endif

```

las funciones de algunos de los archivos anteriores son

parRun.H

Provee rutinas para inicializar ejecución en paralelo y su finalización. Esta instancia se declara en el archivo "argList.H" .

Time.H

Este controla la información del tiempo durante la simulación.

fvc.H

Este archivo, finite volume calculus, contiene rutinas para desarrollar operaciones como interpolación, integración y encontrar las derivadas de escalares, vectores y tensores. Con *fvc.H* todas las operaciones son explícitas mientras que con *fvm.H* todas las operaciones son implícitas.

fvMatrices.H

Tipo de solucionador especial de matrices, diseñado para la soluciones de volúmenes finitos de ecuaciones escalares.

linear.H

Clase de esquema de interpolación de diferencia centrada.

calculatedFvPatchFields.H

Provee condiciones de frontera tipo Neumann.

adjustPhi.H

Para los casos en que no se tenga un limite para la presión, ajusta el balance de los flujos para cumplir la continuidad.

findRefCell.H

Ajusta la celda para la referencia de la presión.

constants.H

Suministra las constantes matemáticas (π, e).

OSspecific.H

Suministra funciones como escribir, borrar, etc.

En el caso de la línea 35 del Código 1 `#include "MULES.H"` es descrito como limitador universal multidimensional con solución explícita, resuelve ecuaciones de transporte meramente convectivas con límites superiores e inferiores.

Debido a que la mezcla de dos fluidos es un flujo transitorio el solucionador utiliza una fusión del algoritmo SIMPLE[38, 19, 22, 39] y el algoritmo PISO[29, 38, 19], llamada

PIMPLE, lo que hace este algoritmo es ejecutar un bucle sobre el algoritmo PISO dentro de cada paso del tiempo y esto permite una subrelajación entre bucles, cosa que no se puede realizar con PISO. El algoritmo PIMPLE permite utilizar incrementos de tiempo mayores sin que se pierda la convergencia de los resultados y se puede reducir a PISO puro utilizando solo un bucle “externo” indicándolo en el archivo *system/fvSolution* en el diccionario PIMPLE en la entrada `nOuterCorrectors`, pero no se puede cambiar a SIMPLE debido a que es un solucionador transitorio.

```
1  #include "pimpleControl.H"
```

"pimpleControl.H" es la clase de control para el algoritmo PIMPLE que verifica y suministra información de la convergencia del bucle PIMPLE.

Luego del encabezado comienza el cuerpo del solucionador

Código 3: main

```
1  int main(int argc, char *argv[])
2  {
3      #include "setRootCase.H"
4      #include "createTime.H"
5      #include "createMesh.H"
6      #include "readGravitationalAcceleration.H"
7      #include "initContinuityErrs.H"
8      #include "createFields.H"
9      #include "readTimeControls.H"
10     #include "CourantNo.H"
11     #include "setInitialDeltaT.H"
12
13     pimpleControl pimple(mesh);
14
15     // * * * * *
16     * * * * //
17
18     Info<< "\nStarting time loop\n" << endl;
19
20     while (runTime.run())
21     {
22         #include "readTimeControls.H"
23         #include "CourantNo.H"
24         #include "alphaCourantNo.H"
```

```

24     #include "setDeltaT.H"
25
26     runTime++;
27
28     Info<< "Time = " << runTime.timeName() << nl << endl;
29
30     twoPhaseProperties.correct();
31
32     #include "alphaEqnSubCycle.H"
33     #include "alphaDiffusionEqn.H"
34
35     // — Pressure-velocity PIMPLE corrector loop
36     while (pimple.loop())
37     {
38         #include "UEqn.H"
39
40         // — Pressure corrector loop
41         while (pimple.correct())
42         {
43             #include "pEqn.H"
44         }
45
46         if (pimple.turbCorr())
47         {
48             turbulence->correct();
49         }
50     }
51
52     runTime.write();
53
54     Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
55         << "   ClockTime = " << runTime.elapsedClockTime() << " s"
56         << nl << endl;
57 }
58
59 Info<< "End\n" << endl;
60
61 return 0;
62 }

```

Cuerpo del código

El cuerpo del código esta estructurado así:

En la línea uno del Código 3 indica que main en este caso está definido de tipo entero, por lo cual devuelve 0 en la línea 61. De la línea 3 a la 11 se incluyen archivos necesarios para correr el caso, tales como "createTime.H", "createMesh.H", "createMesh.H", "readGravitationalAcceleration.H", "initContinuityErrs.H", "createFields.H", "readTimeControls.H", "CourantNo.H", "setInitialDeltaT.H", cada uno con funciones que indica su nombre respectivo.

En la línea 17 del Código 3 se ordena que se muestre un mensaje en la pantalla que indique el inicio del bucle temporal, luego en la línea 19 empieza el bucle temporal hasta que se llegue al tiempo final de la simulación, indicado en el archivo *system/controlDict*, si no ocurre ningún error en el proceso.

Entre las líneas 21 a 24 del Código 3 se incluyen funciones que se ejecutan en cada paso del tiempo: leer los parámetros de control del tiempo necesarios para determinar Δt , calcular el número de Courant global, calcular el número de Courant de la mezcla, y determinar el paso del tiempo (Δt).

La línea 26 indica un avance en el tiempo y la línea 28 imprime en la pantalla el valor del tiempo actual en la simulación.

En la línea 30 se calcula la viscosidad laminar llamando a la función *calcNu()*, definida en *twoPhaseMixture.C*. *calcNu()* calcula la viscosidad cinemática ν definida como la viscosidad dinámica μ promediada en el volumen dividida en la densidad ρ promediada en el volumen.

El archivo incluido en la línea 32, Código 4, lee los parámetros dentro del diccionario PIMPLE **nAlphaCorr** y **nAlphaSubCycles** los subCiclos subdividen el paso de tiempo en n lapsos, después la ecuación convectiva de la concentración **alphaEqn.H** se integra esos n lapsos y se divide entre todo el paso del tiempo, actuando como un filtro para mejorar la estabilidad. Por su parte Alpha Corrections se utilizan en una iteración de punto fijo sobre alpha para mejorar la calidad de la solución, si **nAlphaSubCycles**>1, se encarga de crear el campo para el flujo másico y lo establece igual a cero para cada paso del tiempo y para cada subciclo, usando el archivo **alphaEqn.H** mostrado en Código 5, crea el campo escalar de flujos en las caras y calcula el flujo debido a la fase α_1 , **phiAlpha**, en cada cara de cada volumen en cada instante y con MULES soluciona α_1 de forma

explícita, en este caso la solución está limitada entre 1 y 0, (línea 14 del Código 5) y luego calcula el valor del flujo másico así:

$$\rho \mathbf{S}_f \cdot \vec{V}_f = \mathbf{S}_f \cdot \vec{V}_f \alpha_1 (\rho_1 - \rho_2) + \mathbf{S}_f \cdot \vec{V}_f * \rho_2 \quad (.1)$$

simplificando, tomando $\phi_f = \mathbf{S}_f \cdot \vec{V}_f$ y reordenando queda

$$\rho \phi_f = \alpha_1 \rho_1 \phi_f + \alpha_2 \rho_1 \phi_f \quad (.2)$$

posteriormente se suma la posible diferencia de flujos promediada en el paso del tiempo cuando termina el número de ciclos y asigna el valor al flujo másico. Si `nAlphaSubCycles` =1, solamente calcula el flujo debido a la fase α_1 en cada volumen, con MULES soluciona α_1 de forma explícita y después de esto calcula el valor de la densidad en cada volumen.

Código 4: alphaEqnSubCycle.H

```

1  label nAlphaCorr(readLabel(pimple.dict().lookup("nAlphaCorr"
2      ))) ;
3
4  label nAlphaSubCycles(readLabel(pimple.dict().lookup("
5      nAlphaSubCycles")));
6
7  if (nAlphaSubCycles > 1)
8  {
9      dimensionedScalar totalDeltaT = runTime.deltaT();
10     surfaceScalarField rhoPhiSum(0.0*rhoPhi);
11
12     for
13     (
14         subCycle<volScalarField> alphaSubCycle(alpha1,
15             nAlphaSubCycles);
16         !(++alphaSubCycle).end();
17     )
18     {
19         #include "alphaEqn.H"
20         rhoPhiSum += (runTime.deltaT()/totalDeltaT)*rhoPhi;
21     }
22
23     rhoPhi = rhoPhiSum;

```



```

20 }
21 else
22 {
23     #include "alphaEqn.H"
24 }
25
26 rho == alpha1*rho1 + alpha2*rho2;

```

Código 5: alphaEqn.H

```

1 {
2     word alphaScheme("div(phi,alpha1)");
3
4     surfaceScalarField phiAlpha
5     (
6         fvc::flux
7         (
8             phi,
9             alpha1,
10            alphaScheme
11        )
12    );
13
14    MULES::explicitSolve(alpha1, phi, phiAlpha, 1, 0);
15
16    rhoPhi = phiAlpha*(rho1 - rho2) + phi*rho2;
17
18    Info<< "Phase-1 volume fraction = "
19          << alpha1.weightedAverage(mesh.Vsc()).value()
20          << "   Min(alpha1) = " << min(alpha1).value()
21          << "   Max(alpha1) = " << max(alpha1).value()
22          << endl;
23 }

```

En la línea 33 del Código 3, se incluye el archivo `alphaDiffusionEqn.H` mostrado en el Código 6, en el cual se resuelve la matriz escalar de α_1 debida a la difusión, se determina el valor de α_2 , se agrega el valor del flujo de la sustancia 1 menos el flujo de la sustancia

2 debido a la difusión al valor del flujo másico y se actualiza el valor de la densidad en cada volumen con el valor apropiado de α .

Código 6: alphaDiffusionEqn.H

```

1  {
2      fvScalarMatrix alpha1Eqn
3      (
4          fvm::ddt(alpha1)
5          - fvc::ddt(alpha1)
6          - fvm::laplacian
7          (
8              volScalarField("Dab", Dab + alphatab*turbulence
9                  ->nut()),
10             alpha1
11         )
12     );
13
14     alpha1Eqn.solve();
15
16     alpha2 = 1.0 - alpha1;
17     rhoPhi += alpha1Eqn.flux()*(rho1 - rho2);
18 }
19 rho = alpha1*rho1 + alpha2*rho2;

```

Algoritmo PIMPLE

Con la sentencia while de la línea 36 del Código 3, se inicia el ciclo corrector del algoritmo PIMPLE, el cual empieza incluyendo el archivo "UEqn.H", Código 7. En este se crea el tensor para la solución de U ; si el predictor de momento se establece, se resuelve la ecuación de velocidad (como sucede en este caso). En este archivo la velocidad se predice implícitamente debido a la gran estabilidad de los métodos implícitos, lo cual quiere decir que un conjunto de ecuaciones lineales acopladas, expresadas en forma matricial vectorial como

$$\mathbf{Ax} = \mathbf{b} \quad (.3)$$

son resueltas. De forma que, el lado izquierdo de esta ecuación se ensambla en las líneas 1 a 6 del Código 7.

Código 7: UEqn.H

```

1      fvVectorMatrix UEqn
2      (
3          fvm::ddt(rho, U)
4          + fvm::div(rhoPhi, U)
5          + turbulence->divDevRhoReff(rho, U)
6      );
7
8      UEqn.relax();
9
10     if (pimple.momentumPredictor())
11     {
12         solve
13         (
14             UEqn
15             ==
16             fvc::reconstruct
17             (
18                 (
19                     - ghf*fvc::snGrad(rho)
20                     - fvc::snGrad(p_rgh)
21                 ) * mesh.magSf()
22             )
23         );
24     }

```

En las líneas 1 a 6 se usa la clase `fvm` establecida para matriz de volúmenes finitos, y es usada cuando las operaciones van a ser implícitas, aquí se se forma la matriz del lado izquierdo. El término `fvm::ddt(rho, U)` corresponde a la primera derivada de la velocidad por la densidad, $\frac{\partial \rho \vec{V}}{\partial t}$, el término `+ fvm::div(rhoPhi, U)`, a la divergencia del flujo másico $\rho \vec{V}$, multiplicado por la velocidad, $\nabla \cdot \rho \vec{V} \vec{V}$, ó, en otras palabras, la convección del flujo másico. Para el término `+ turbulence->divDevRhoReff(rho, U)` se debe examinar el código para los modelos RANS o LES.

significado de `divDevRhoReff(rho, U)`

si se examina la implementación incompresible $k - \varepsilon$ en `src/turbulenceModels/incompressible/RAS/kEpsilon/kEpsilon.C`, se puede ver que `divDevRhoReff(rho, U)` se define así:

Código 8: `divDevRhoReff(rho,U)`

```
1  tmp<fvVectorMatrix> kEpsilon::divDevRhoReff
2  (
3      const volScalarField& rho,
4      volVectorField& U
5  ) const
6  {
7      volScalarField muEff("muEff", rho*nuEff());
8
9      return
10     (
11         - fvm::laplacian(muEff, U)
12         - fvc::div(muEff*dev(T(fvc::grad(U))))
13     );
14 }
```

En la línea 7 del Código 8 se crea el campo de viscosidad efectiva y de la línea 9 a la 13 se puede traducir la Ecuación .4:

$$\text{divDevRhoReff}(\rho, \vec{V}) = -\nabla^2(\mu_{eff} \vec{V}) - \nabla \cdot \left\{ \mu_{eff} \text{dev} \left[(\nabla \vec{V})^T \right] \right\} \quad (.4)$$

donde $\text{dev}(\mathbf{A}) = \mathbf{A} - \frac{1}{3}\text{trace}(\mathbf{A})\mathbf{I}$, donde \mathbf{I} es la matriz identidad, por lo tanto la Ecuación .4 quedaría:

$$\text{divDevRhoReff}(\rho, \vec{V}) = -\nabla^2(\mu_{eff} \vec{V}) - \nabla \cdot \left\{ \mu_{eff} \left(\nabla \vec{V} \right)^T - \frac{1}{3}\text{trace}(\nabla \vec{V})^T \mathbf{I} \right\} \quad (.5)$$

en notación tensorial

$$\text{divDevRhoReff}(\rho, \vec{V}) = -\frac{\partial}{\partial x_j} \frac{\partial}{\partial x_j} (\mu_{eff} \bar{V}_i) - \frac{\partial}{\partial x_j} \left\{ \mu_{eff} \left[\frac{\partial \bar{V}_j}{\partial x_i} - \frac{1}{3} \left(\frac{\partial \bar{V}_k}{\partial x_k} \right) \delta_{ij} \right] \right\} \quad (.6)$$

reorganizando

$$\text{divDevRhoReff} \left(\rho, \vec{V} \right) = -\frac{\partial}{\partial x_j} \left\{ \mu_{eff} \left(\frac{\partial \bar{V}_i}{\partial x_j} + \frac{\partial \bar{V}_j}{\partial x_i} \right) - \frac{1}{3} \left(\frac{\partial \bar{V}_k}{\partial x_k} \right) \delta_{ij} \right\} \quad (.7)$$

Este término difiere del término de esfuerzos viscosos en la ecuación de cantidad de movimiento debido a que se resta solo $1/3$ de la divergencia de la velocidad en vez de $2/3$, los otros modelos RANS y LES presentan esta misma formulación, por otra parte los modelos compresibles RANS y LES incluyen la fracción correcta $2/3$, usando el operador *dev2*. Parecería que se cometió un error en la programación de estos modelos, pero para fluidos incompresibles el término $\frac{\partial \bar{V}_k}{\partial x_k} = 0$, entonces se puede concluir que este término se mantiene solo para corrección, es decir, $1/3$ es solo un factor de peso.

En la línea se relajan los términos incluidos en la variable *UEqn* (líneas 9 a 13 del Código 7) y se procede a formar los términos de la derecha de la Ecuación .3 en las líneas 10 a 22 del Código 7. en ésta parte del código se puede ver el lado izquierdo de la ecuación almacenados en *UEqn* multiplicado por el vector solución es igual a dos términos. El primer término es el negativo de la energía potencial específica calculada en la cara del elemento, multiplicada por el gradiente normal a la superficie de la densidad. El segundo término es el negativo del gradiente normal a la cara de la presión ($p_rgh = p - \rho gh$, con $gh = \mathbf{g} \cdot \mathbf{mesh.C()}$ en otras palabras gh es el producto punto del vector \mathbf{g} y el vector de posición de la celda) . Los dos términos anteriores multiplicados por el área de la frontera. El comando `fvc::reconstruct`, línea 16 del Código 7, reconstruye un campo de volúmenes desde un campo de flujos en las caras.

El bucle corrector de la presión inicia de la línea 41 a 44 del Código 3, con un bucle while que se repite mientras se esté corrigiendo la presión. En este bucle se incluye el archivo "pEqn.H" Código 9, en el cual en las primeras 6 líneas se predice la velocidad, debido a que es necesaria para ajustar el flujo para la ecuación de la presión.

Código 9: pEqn.H

```

1 {
2     volScalarField rAU("rAU", 1.0/UEqn.A());
3     surfaceScalarField rAUf("Dp", fvc::interpolate(rAU));
4
5     volVectorField HbyA("HbyA", U);
6     HbyA = rAU*UEqn.H();

```

```

7
8     surfaceScalarField phiHbyA
9     (
10         "phiHbyA",
11         (fvc::interpolate(HbyA) & mesh.Sf())
12         + fvc::ddtPhiCorr(rAU, rho, U, phi)
13     );
14     adjustPhi(phiHbyA, U, p_rgh);
15     phi = phiHbyA;
16
17     surfaceScalarField phig
18     (
19         - ghf*fvc::snGrad(rho)*rAUf*mesh.magSf()
20     );
21
22     phiHbyA += phig;
23
24     while (pimple.correctNonOrthogonal())
25     {
26         fvScalarMatrix p_rghEqn
27         (
28             fvm::laplacian(rAUf, p_rgh) == fvc::div(phiHbyA)
29         );
30
31         p_rghEqn.setReference(pRefCell, getRefCellValue(
32             p_rgh, pRefCell));
33
34         p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.
35             finalInnerIter())));
36
37         if (pimple.finalNonOrthogonalIter())
38         {
39             phi = phiHbyA - p_rghEqn.flux();
40
41             U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn
42                 .flux())/rAUf);
43             U.correctBoundaryConditions();

```

```

41     }
42 }
43
44 #include "continuityErrs.H"
45
46 p == p_rgh + rho*gh;
47
48 if (p_rgh.needReference())
49 {
50     p += dimensionedScalar
51     (
52         "p",
53         p.dimensions(),
54         pRefValue - getRefCellValue(p, pRefCell)
55     );
56     p_rgh = p - rho*gh;
57 }
58 }

```

El nombre *de la velocidad*, *HbyA*, se da partiendo de la forma como se resuelven los algoritmos PISO y PIMPLE, donde la forma semi-discretizada de la ecuación de cantidad de movimiento es

$$a_p \vec{V}_P = \mathbf{H}(\vec{V}) - \nabla p \quad (.8)$$

donde a_p son los coeficientes diagonales de la matriz resultante de la discretización de la ecuación de cantidad de movimiento. $\mathbf{H}(\vec{V})$ representa los coeficientes no diagonales (principalmente compuestos por términos convectivos y difusivos) y los términos fuente (la parte fuente del término transitorio y los otros términos fuente que aparezcan en *UEqn*) aparte del gradiente de presión. Asimismo en la Ecuación .8 el campo de presiones resulta de la etapa del tiempo anterior, por lo tanto se busca \vec{V}_P que se encuentra con

$$\vec{V}_P = \frac{\mathbf{H}(\vec{V})}{a_P} - \frac{1}{a_P} \nabla p \quad (.9)$$

y como se sabe, Ecuación 5.144, se puede construir la ecuación de presión:

$$\nabla \cdot \left(\frac{1}{a_P} \nabla p \right) = \nabla \cdot \left(\frac{\mathbf{H}(\vec{V})}{a_P} \right) \quad (.10)$$

de donde se puede reconstruir la ecuación de velocidad con la Ecuación .9. Dado que en el solver estudiado el sistema de ecuaciones tiene la forma $\mathbf{CU}=\mathbf{R}$ donde \mathbf{C} es una matriz, \mathbf{U} es el vector solución, y \mathbf{R} es el lado derecho la matriz \mathbf{C} se puede dividir en dos, una con solo los términos diagonales (llamada \mathbf{A}) y una matriz con los términos fuera de la diagonal (llamada \mathbf{H}'), de modo que $\mathbf{C} = \mathbf{A} + \mathbf{H}'$, conformando el sistema lineal $(\mathbf{A} + \mathbf{H}')\mathbf{U} = \mathbf{R}$, ó $\mathbf{AU} = \mathbf{R} - \mathbf{H}'\mathbf{U}$. En este solver el lado derecho es llamado \mathbf{H} , con $\mathbf{H} = \mathbf{R} - \mathbf{H}'\mathbf{U}$, lo que nos lleva a $\mathbf{AU}=\mathbf{H}$. En este procedimiento se hace obvio el nombre dado a la velocidad por su sigla en ingles \mathbf{H} divided by \mathbf{A} o \mathbf{HbyA} .

Como primer paso el coeficiente central de la matriz de la ecuación de velocidad es extraído e invertido, línea 2 del Código 9, donde `UEqn.A()` extrae los coeficientes centrales (clase *fvMatrix*). En la línea 6 se predice la \mathbf{HbyA} multiplicando el recíproco de los coeficientes centrales con la operación `UEqn.H()` en la misma matriz, Ecuación .9,

Con esta velocidad anticipada es creado el flujo que depende solamente de la velocidad, $\phi_{\mathbf{HbyA}}$, línea 8 a 13 del Código 9, con `(fvc::interpolate(HbyA)&mesh.Sf())` se interpola el campo vectorial de \mathbf{HbyA} a la cara y se efectúa el producto punto por el vector de área de la cara, y se suma la corrección de ϕ_c obtenida con `fvc::ddtPhiCorr(rAU,rho,U,phi)` (`/src/finiteVolume/finiteVolume/fvc/fvcDdt.C`) que llama a una versión apropiada de `fvcDdtphicorr` (`/src/finiteVolume/finiteVolume/ddtSchemes/EulerDdtScheme/EulerDdtScheme.C`), evalúa la diferencia entre el valor antiguo del flujo y el flujo calculado con los valores antiguos de la velocidad, con $\frac{1}{A}$, Δt y ϕ mediante la expresión

$$\phi_{dif} = (\phi_{antigua} - (V_{antigua} \cdot S)) \quad (.11)$$

luego se llama a la función `fvcDdtPhiCoeff` (`/src/finiteVolume/finiteVolume/ddtSchemes/d-dtScheme/ddtScheme.C`), que calcula el valor de K_c

$$K_c = 1 - \min \left[\frac{|\phi_{dif}|}{|\phi_{antigua}| + \epsilon_{peq}}, 1 \right] \quad (.12)$$

la proporción debe estar entre un pequeño ϵ y 1, entonces este campo evaluado es retornado a la función *fvDdtphicorr* donde se calcula el valor de corrección de ϕ

$$\phi_c = \phi_{dif} K_c \frac{1}{A} \frac{1}{\Delta t} \quad (.13)$$

luego es retornado el valor ϕ_c .

En la línea 14 de Código 9 se ajustan los flujos en las fronteras que sean ajustables, en la línea 15 asigna el valor de *phiHbyA* a *phi*, entonces de la línea 17 a la 20 calcula el campo de escalar de flujo debido a la gravedad en la superficie de la celda, para sumárselo a *phi* en la línea 22. Se entra en el ciclo while de la línea 24 y se ejecuta este bucle hasta el número de correctores de no ortogonalidad establecidos en *fvSolution*. La ecuación de la presión *p_rghEqn*, Ecuación .10, es ensamblada en la línea 24, se establece la referencia de presión determinada en el archivo *fvSolution* donde sea necesaria (línea 31) . En la línea 33 se resuelve con la tolerancia determinada para *p_rghEqn* y escoge la tolerancia relativa final para la ultima iteración del archivo *fvSolution*. Cuando se llega a la ultima iteración, se pasa a la línea 37 donde se corrige el flujo con el valor de la presión encontrado, luego se actualiza el valor del campo de velocidades \mathbf{U} solamente teniendo en cuenta el flujo debido a la presión *p_rgh* (línea 39), y por último se corrigen las condiciones de frontera que lo permitan.

En línea 46 de Ecuación .10 se establece el valor del campo de presiones p si es necesaria una referencia, se suma el valor de referencia establecido y se asigna el valor a *p_rgh* en la línea 56 para retornar al Código 3 donde se procede a actualizar el modelo de turbulencia (si está activado) basado en el campo de velocidad resuelto, esto se hace hasta que se termina el bucle corrector de presión velocidad del algoritmo PIMPLE. Las líneas 52 a 56 escriben en la pantalla el tiempo de ejecución y el tiempo empleado y continua el siguiente paso del tiempo

ANEXO B. ARCHIVOS NECESARIOS PARA LA SIMULACIÓN

Simulación laminar

Directorio constant

en este directorio se encuentran los valores constantes que serán usados en la simulación ubicados dentro de sus directorios respectivos.

transportProperties las propiedades de transporte para este caso involucran la difusividad molecular (D_{AB}), el inverso del numero de Schmith ($alphatab$), los valores de densidad (ρ) y viscosidad cinemática (ν).

```
1  /*-----*-- C++  *-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation |Version:2.2.1
5  |    \ \ / And          |Web:      www.OpenFOAM.org
6  |      \ \ / Manipulation|
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object        transportProperties;
15  }
16  // * * * * *
17
18  Dab              Dab [0 2 -1 0 0 0 0] 0;
19  alphatab         alphatab [0 0 0 0 0 0 0] 1.428571429;
```

```

20
21
22 phase1
23 {
24     transportModel    Newtonian;
25     nu                 nu [0 2 -1 0 0 0 0] 4e-07;
26     rho                rho [1 -3 0 0 0 0 0] 750;
27 }
28
29 phase2
30 {
31     transportModel    Newtonian;
32     nu                 nu [0 2 -1 0 0 0 0] 9.1778e-08;
33     rho                rho [1 -3 0 0 0 0 0] 523;
34 }
35
36 // *****

```

g en este archivo se encuentra las unidades de la gravedad, así como su valor y dirección, lo que permite diferentes inclinaciones posibles para el caso.

```

1  /*----- C++ -----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation  |Version:2.2.1
5  |   \ \    / And         |Web:      www.OpenFOAM.org
6  |    \ \ /  Manipulation|
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         uniformDimensionedVectorField;
13     location      "constant";
14     object        g;
15 }

```

```

16 // * * * * * * * * * * * * * * * * * * * * * * * * //
17
18 dimensions      [0 1 -2 0 0 0 0];
19 value           (0 -9.81 0);
20
21 // * * * * * * * * * * * * * * * * * * * * * * * * //

```

turbulenceProperties En este archivo se define el modelo turbulento a resolver, para este caso es laminar.

```

1 /*----- C++ -----*\
2 | ===== |
3 | \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4 | \ \      / Operation |Version:2.2.1
5 |  \ \    / And        |Web:      www.OpenFOAM.org
6 |   \ \ / Manipulation|
7 \*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object         turbulenceProperties;
15 }
16 // * * * * * * * * * * * * * * * * * * * * * * * * //
17
18 simulationType   laminar;
19
20 // * * * * * * * * * * * * * * * * * * * * * * * * //

```

Directorio system

En este directorio están contenidos todos los parámetros que manejan el desarrollo de la simulación

controlDict

```
1  /*-----*- C++ -*-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation |Version:2.2.1
5  |    \ \    / And       |Web:      www.OpenFOAM.org
6  |      \ \ / Manipulation|
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object         controlDict;
15 }
16 // * * * * *
17
18 application      twoLiquidMixingFoam;
19
20 startFrom         startTime;
21
22 startTime         4.4;
23
24 stopAt            endTime;
25
26 endTime           5.2;
27
28 deltaT            0.05;//0.05
29
30 writeControl      adjustableRunTime;
31
32 writeInterval     0.2;
33
34 purgeWrite        0;
35
36 writeFormat       ascii;
```

```

37
38 writePrecision 6;
39
40 writeCompression uncompressed;
41
42 timeFormat      general;
43
44 timePrecision 6;
45
46 runTimeModifiable yes;
47
48 adjustTimeStep on;
49
50 maxCo           0.5;
51 maxAlphaCo      0.5;
52 maxDeltaT       1;
53
54 // *****

```

fvSchemes En este archivo se determinan los esquemas numéricos a usar

```

1  /*-----*-- C++  *-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation  |Version:2.2.1
5  |   \ \    / And         |Web:      www.OpenFOAM.org
6  |    \ \ /  Manipulation|
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object         fvSchemes;
15 }

```

```

16
17 // * * * * * * * * * * * * * * * * * * * * * * * * //
18
19 ddtSchemes
20 {
21     default          Euler;
22 }
23
24 gradSchemes
25 {
26     default          Gauss linear;
27 }
28
29 divSchemes
30 {
31     default          none;
32
33     div(rho*phi,U)   Gauss linear;
34     div(phi,alpha1)  Gauss vanLeer;
35     div(phi,epsilon) Gauss limitedLinear 1;
36     div(phi,k)       Gauss limitedLinear 1;
37     div((muEff*dev(T(grad(U)))) Gauss linear;
38 }
39
40 laplacianSchemes
41 {
42     default          Gauss linear corrected;
43 }
44
45 interpolationSchemes
46 {
47     default          linear;
48 }
49
50 snGradSchemes
51 {
52     default          corrected;

```

```

53 }
54
55 fluxRequired
56 {
57     default          no;
58     p_rgh;
59     alpha1;
60 }
61
62
63 // *****

```

fvSolution En este archivo se introducen los parámetros usados por los métodos de solución de las ecuaciones

```

1  /*----- C++ -----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation  |Version:2.2.1
5  |   \ \    /  And        |Web:      www.OpenFOAM.org
6  |    \ \ /  Manipulation|
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        fvSolution;
15 }
16 // * * * * *
17
18 solvers
19 {
20     "alpha1.*"
21     {

```



```

22         solver          smoothSolver;
23         smoother        GaussSeidel;
24         tolerance       1e-9;
25         relTol          0;
26         nSweeps         1;
27     }
28
29     p_rgh
30     {
31         solver          GAMG;
32         tolerance       1e-7;
33         relTol          0.01;
34         smoother        GaussSeidel;
35         cacheAgglomeration true;
36         nCellsInCoarsestLevel 10;
37         agglomerator     faceAreaPair;
38         mergeLevels      1;
39     }
40
41     p_rghFinal
42     {
43         $p_rgh;
44         relTol          0;
45     }
46
47     U
48     {
49         solver          smoothSolver;
50         smoother        GaussSeidel;
51         tolerance       1e-8;
52         relTol          0.1;
53         nSweeps         1;
54     }
55
56     UFinal
57     {
58         $U;

```

```

59         relTol            0;
60     }
61     epsilonFinal
62     {
63         solver              PBiCG;
64         preconditioner      DILU;
65
66         tolerance           1e-05;
67         relTol              0;
68     }
69     kFinal
70     {
71         solver              PBiCG;
72         preconditioner      DILU;
73
74         tolerance           1e-05;
75         relTol              0;
76     }
77 }
78
79 PIMPLE
80 {
81     momentumPredictor      yes;
82     nOuterCorrectors        1;
83     nCorrectors              2;
84     nNonOrthogonalCorrectors 1;
85     nAlphaSubCycles          2;
86     nAlphaCorr                1;
87     pRefValue                 0;
88     pRefPoint                 (0.1 0.1 1);
89 }
90
91
92 // *****

```

setFieldsDict en este archivo se indica la extensión inicial del campo de concentraciones `alpha1`

```
1  /*-----*-- C++ -*-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation  |Version:2.2.1
5  |    \ \    / And        |Web:      www.OpenFOAM.org
6  |      \ \ / Manipulation|
7  \*-----*--*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        setFieldsDict;
14 }
15 // * * * * *
16
17 defaultFieldValues
18 (
19     volScalarFieldValue alpha1 0
20     volVectorFieldValue U (0 0 0)
21 );
22
23 regions
24 (
25     boxToCell
26     {
27         box (0 -2 -2) (0.1 2 2);
28
29         fieldValues
30         (
31             volScalarFieldValue alpha1 1
32         );
33     }
34 );
```

35

36

```
// ***** /
```

Directorio 0.org En este directorio se encuentran los campos iniciales que son copiados a la carpeta temporal del inicio de la simulación. A continuación se muestran los archivos correspondientes al caso de malla 2D.

U

```
1  /*-----*- C++ -*-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation  |Version:2.2.1
5  |    \ \    / And        |Web:      www.OpenFOAM.org
6  |      \ \ / Manipulation|
7  \*-----*-*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object         U;
15 }
16 // * * * * *
17 //
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField     uniform (0 0 0);
21
22 boundaryField
23 {
24     left
25     {
26         type          fixedValue;
27         value          uniform (1 0 0);
```

```

28     }
29     right
30     {
31         type            zeroGradient;
32     }
33     bottom
34     {
35         type            fixedValue;
36         value           uniform (0 0 0);
37     }
38     top
39     {
40         type            fixedValue;
41         value           uniform (0 0 0);
42     }
43     frontBack
44     {
45         type            empty;
46     }
47 }
48
49
50
51 // *****

```

p_rgh

```

1  /*-----*-- C++ -*-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation  |Version:2.2.1
5  |   \ \    /  And        |Web:      www.OpenFOAM.org
6  |    \ \ /  Manipulation|
7  \*-----*-/
8  FoamFile
9  {
10     version            2.0;

```

```

11     format      ascii;
12     class       volScalarField;
13     location    "0";
14     object      p_rgh;
15 }
16 // * * * * *
17
18 dimensions      [1 -1 -2 0 0 0 0];
19
20 internalField   uniform 0;
21
22 boundaryField
23 {
24     left
25     {
26         type      zeroGradient;
27     }
28     right
29     {
30         type      fixedValue;
31         value      uniform 0;
32     }
33     bottom
34     {
35         type      fixedFluxPressure;
36     }
37     top
38     {
39         type      fixedFluxPressure;
40     }
41     frontBack
42     {
43         type      empty;
44     }
45 }
46
47

```

```
48 // ***** //
```

alpha1

```
1  /*-----* C++ *-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation |Version:2.2.1
5  |    \ \ / And          |Web:      www.OpenFOAM.org
6  |      \ \ / Manipulation|
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object         alpha1;
15 }
16 // * * * * *
17
18 dimensions      [0 0 0 0 0 0 0];
19
20 internalField    uniform 0;
21
22 boundaryField
23 {
24     left
25     {
26         type      fixedValue;
27         value      uniform 1;
28     }
29     right
30     {
31         type      zeroGradient;
32     }
33     bottom
```

```

34     {
35         type                zeroGradient;
36     }
37     top
38     {
39         type                zeroGradient;
40     }
41     frontBack
42     {
43         type                empty;
44     }
45 }
46
47
48 // *****

```

Simulación turbulenta

los archivos aquí incluidos son similares a los incluidos en el caso laminar pero adicionando los parámetros turbulentos y cambiando los esquemas numéricos y demás datos necesarios para su solución. Ver manual OpenFOAM.

Scripts usados en las simulaciones

Geometría

Para obtener la geometría se usó el software SALOME-MECA, y para facilitar su parametrización se usaron scripts que además automatizaron el proceso. El script usado en el dibujo y mallado de la tubería se encuentra a continuación.

tubomallado_x

```

1  import math
2  import geumpyDC
3  import geumpy

```



```

4  import smesh
5  import salome
6  geo = geompy
7
8  # parametros
9  # -----
10
11  radius = 0.006
12  height = 10
13
14  # construire el tubo
15  # -----
16
17  base = geo.MakeVertex(0, 0, 0)
18  direction = geo.MakeVectorDXDYDZ(1, 0, 0)
19
20  cylinder = geo.MakeCylinder(base, direction, radius, height)
21
22  geo.addToStudy(cylinder, "tubo")
23
24  # construire bloques
25  # -----
26
27  size = radius/2
28
29  box_rot = geo.MakeBox(0, -size, -size, height, +size, +size)
30  box_axis = geo.MakeLine(base, direction)
31  box = geo.MakeRotation(box_rot, box_axis, math.pi/4)
32
33  hole = geo.MakeCut(cylinder, box)
34
35  plane_trim = 2.0*height
36
37  plane_a = geo.MakePlane(base, geo.MakeVectorDXDYDZ(0, 0, 1),
38                          plane_trim)
39  plane_b = geo.MakePlane(base, geo.MakeVectorDXDYDZ(0, 1, 0),
40                          plane_trim)

```

```

39
40 blocks_part = geo.MakePartition([hole], [plane_a, plane_b],
    [], [], geo.ShapeType["SOLID"])
41 blocks_list = [box] + geo.SubShapeAll(blocks_part, geo.
    ShapeType["SOLID"])
42 blocks_all = geo.MakeCompound(blocks_list)
43 blocks = geo.MakeGlueFaces(blocks_all, 0.0001)
44
45 geo.addToStudy(blocks, "tuberia")
46
47 # construir objetos geometricos
48 # -----
49
50 # crear los grupos
51
52 def group(name, shape, type, base=None, direction=None):
53     t = geo.ShapeType[type]
54     g = geo.CreateGroup(shape, t)
55
56     geo.addToStudy(g, name)
57     g.SetName(name)
58
59     if base!=None:
60         l = geo.GetShapesOnPlaneWithLocationIDs(shape, t,
            direction, base, geo.GEOM.ST_ON)
61         geo.UnionIDs(g, l)
62
63     return g
64
65 group_a = group("entrada", blocks, "FACE", base, direction)
66
67 base_b = geo.MakeVertex(height, 0, 0)
68 group_b = group("salida", blocks, "FACE", base_b, direction)
69
70
71 #crear el grupo pared

```

```

72 group_2 = geompy.CreateGroup(blocks, geompy.ShapeType["FACE"
73 ])
74 # adicionar objetos al grupo pared
75 SubFaceList1 = geompy.SubShapeAllSortedCentres(blocks,
76 geompy.ShapeType["FACE"])
77
78 for i in [6,7,14,15] :
79     FaceID_1 = geompy.GetSubShapeID(blocks, SubFaceList1[i
80 ])
81     geompy.AddObject(group_2, FaceID_1)
82 id_group_2 = geompy.addToStudy(group_2, "pared")
83
84 #Mallar los bloques con hexahedros
85 # -----
86
87 smesh.SetCurrentStudy(salome.myStudy)
88
89 def discretize(x, y, z, n, s=blocks):
90     p = geo.MakeVertex(x, y, z)
91     e = geo.GetEdgeNearPoint(s, p)
92     a = hexa.Segment(e)
93     a.NumberOfSegments(n)
94     a.Propagation()
95
96 hexa = smesh.Mesh(blocks)
97
98 hexa_1d = hexa.Segment()
99 hexa_1d.NumberOfSegments(1)
100
101 discretize(0, +radius, +radius, 5)
102 discretize(0, -radius, +radius, 5)
103 discretize(0, -radius, -radius, 5)
104 discretize(0, +radius, -radius, 5)
105 discretize(0, (radius+size)/2, 0, 1)
106 discretize(height/2, +radius, 0, height*250)

```

```

106  hexa.Quadrangle()
107  hexa.Hexahedron()
108
109  hexa.Compute()
110
111  hexa.Group(group_a)
112  hexa.Group(group_b)
113  hexa.Group(group_2)

```

Shell scripts

Estos scripts están diseñados para automatizar la solución del caso y se ejecutan desde la terminal ubicándose en la ruta del caso por medio de la orden

```

1  $ ./nombre_del_script

```

en ellos se llaman las funciones y aplicaciones necesarias en el respectivo caso.

Caso turbulento mapeado

```

1  #!/bin/sh
2  cd ${0%/*} || exit 1      # run from this directory
3
4  # Source tutorial run functions
5  . $WM_PROJECT_DIR/bin/tools/RunFunctions
6
7  # Set application name
8  application='getApplication'
9
10
11
12  runApplication ideasUnvToFoam tubo10m2mmyZhao.unv
13  pyFoamChangeBoundaryType.py . pared wall
14  runApplication setFields
15
16  mkdir 1

```

```
17 cp -r 0.org/U 1
18
19 runApplication mapFields ~/Escritorio/10m12mm/12/
    caso10m12mm_4mmy
20
21 cp -r 0.org/p_rgh 1
22 cp -r 0.org/k 1
23 cp -r 0.org/epsilon 1
24 cp -r 0.org/alpha1 1
25
26 runApplication decomposePar
27 runApplication mpirun -np 4 $application -parallel
28 runApplication reconstructPar
29 #
    -----
    end-of-file
```

ANEXO C. ARCHIVO twoLiquidMixingDyMFoam.C

Código 10: twoLiquidMixingDyMFoam

```
1  /*-----*-- C++  *-----*\
2  | ===== |
3  |  \ \      /Field      |OpenFOAM:The Open Source CFD Toolbox
4  |  \ \      / Operation |Version:2.2.1
5  |    \ \ / And          |Web:      www.OpenFOAM.org
6  |      \ \ / Manipulation|
7  \*-----*/
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/
12      or modify it
13      under the terms of the GNU General Public License as
14      published by
15      the Free Software Foundation, either version 3 of the
16      License, or
17      (at your option) any later version.
18
19      OpenFOAM is distributed in the hope that it will be
20      useful, but WITHOUT
21      ANY WARRANTY; without even the implied warranty of
22      MERCHANTABILITY or
23      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
24      Public License
25      for more details.
26
27      You should have received a copy of the GNU General
28      Public License
```

```

22     along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.
23
24 Application
25     twoLiquidMixingDyMFoam
26
27 Description
28     Solucionador para mezclado de 2 fluidos
29     incompresibles
30     añadiendo utilidad de malla móvil
31
32 Autor
33     Carlos Alberto Flórez Arias
34
35     Turbulence modelling is generic, i.e. laminar, RAS or
36     LES may be selected.
37
38 \*-----*/
39
40 #include "fvCFD.H"
41 #include "MULES.H"
42 #include "subCycle.H"
43 #include "incompressibleTwoPhaseMixture.H"
44 #include "turbulenceModel.H"
45 #include "dynamicFvMesh.H"
46 #include "pimpleControl.H"
47
48 // * * * * *
49
50 int main(int argc, char *argv[])
51 {
52     #include "setRootCase.H"
53     #include "createTime.H"
54     #include "createDynamicFvMesh.H"
55     #include "readGravitationalAcceleration.H"
56     #include "initContinuityErrs.H"
57     #include "createFields.H"

```

```

56     #include "readTimeControls.H"
57
58     pimpleControl pimple(mesh);
59
60     surfaceScalarField phiAbs("phiAbs", phi);
61     fvc::makeAbsolute(phiAbs, U);
62
63     #include "correctPhi.H"
64     #include "CourantNo.H"
65     #include "setInitialDeltaT.H"
66
67     // * * * * *
68
69     Info<< "\nStarting time loop\n" << endl;
70
71     while (runTime.run())
72     {
73         #include "readControls.H"
74         #include "CourantNo.H"
75         #include "alphaCourantNo.H"
76         #include "setDeltaT.H"
77
78         runTime++;
79
80         Info<< "Time = " << runTime.timeName() << nl << endl
81             ;
82
83         scalar timeBeforeMeshUpdate = runTime.elapsedCpuTime
84             ();
85
86         {
87             // Calculate the relative velocity used to map
88             // the relative flux phi
89             volVectorField Urel("Urel", U);
90
91             if (mesh.moving())
92             {

```



```

90         Urel -= fvc::reconstruct(fvc::meshPhi(U));
91     }
92
93     // Do any mesh changes
94     mesh.update();
95 }
96
97 if (mesh.changing())
98 {
99     Info<< "Execution time for mesh.update() = "
100         << runTime.elapsedCpuTime() -
101         timeBeforeMeshUpdate
102         << " s" << endl;
103
104     gh = g & mesh.C();
105     ghf = g & mesh.Cf();
106 }
107
108 if (mesh.changing() && correctPhi)
109 {
110     #include "correctPhi.H"
111 }
112
113 if (mesh.changing() && checkMeshCourantNo)
114 {
115     #include "meshCourantNo.H"
116 }
117
118 twoPhaseProperties.correct();
119
120 #include "alphaEqnSubCycle.H"
121 #include "alphaDiffusionEqn.H"
122
123 // --- Pressure-velocity PIMPLE corrector loop
124 while (pimple.loop())
125 {
126     #include "UEqn.H"

```

```

126
127         // --- Pressure corrector loop
128         while (pimple.correct())
129         {
130             #include "pEqn.H"
131         }
132
133         if (pimple.turbCorr())
134         {
135             turbulence->correct();
136         }
137     }
138
139     runTime.write();
140
141     Info<< "ExecutionTime = " << runTime.elapsedCpuTime
142           () << " s"
143           << "   ClockTime = " << runTime.elapsedClockTime
144           () << " s"
145           << nl << endl;
146 }
147
148 Info<< "End\n" << endl;
149
150 return 0;
151 }
152 // *****

```