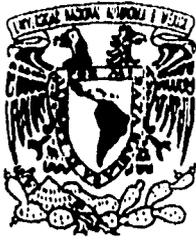


3
24



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

Solución numérica a las ecuaciones de Navier - Stokes en coordenadas generalizadas.

TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE:

F I S I C O
P R E S E N T A

Julián Tercero Becerra Sagredo



Director de Tesis:

Dr. Eduardo Ramos Mora

México D.F. 1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

M. en C. Virginia Abrín Batule
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis: Solución Numérica a las Ecuaciones de Navier-Stokes en Coordenadas Generalizadas.

realizado por Julián Tercero Becerra Sagredo

con número de cuenta 9150723-7 , pasante de la carrera de Física.

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario

DR. EDUARDO RAMOS MORA

Ramos

Propietario

DR. RAMON PERALTA FABI

Ramon Peralta Fabi

Propietario

DR. ROBERTO ALEJANDRO RUELAS MAYORGA

Ruelas

Suplente

DR. ALEJANDRO RAGA RASMUSSEN

A. Raga

Suplente

DR. SERGIO CUEVAS GARCIA

Suevas

Consejo Departamental de Física

DR. ROBERTO ALEJANDRO RUELAS MAYORGA

Ruelas



FACULTAD DE CIENCIAS
DEPARTAMENTO DE FÍSICA

A Margarita, Cecy

y a mi banda pesada.

Agradecimientos

A mi madre por ser una fuente inagotable de amor, libertad y apoyo.

A cada uno de los miembros de la banda pesada: la Paula, el Paco, el Málaga, el Charlee, el Neto, la Chana, el Rodrigo, el Burguer, la Vanesa, el Orzo, la Blanca, el Joaquín y el Nicolás; por haber compartido los mejores días de mi existencia entre tareas, reventones, aventuras y viajes.

Muy especialmente al Dr. Eduardo Ramos Mora por haberme dado la oportunidad de realizar este trabajo de tesis bajo una guía intensa y un ambiente de amistad y equipo que sólo se tiene con los mejores amigos. Cuando la presión del trabajo llegó a su límite, Eduardo estuvo cerca preocupado hasta por mi estado de salud.

A los sinodales: Dr. Ramón Peralta Fabi, Dr. Alejandro Ruelas, Dr. Sergio Cuevas y Dr. Alejandro Raga, por haber revisado este trabajo de tesis.

A todo el grupo de Transferencia de Masa y Energía del Laboratorio de Energía Solar del IIM, UNAM., especialmente a Guillermo, Juan Manuel, Guadalupe, Ramón y Minerva, por su amistad y apoyo.

A mis maestros durante la carrera, en especial al Dr. Jorge Ize, Dr. Ramón Peralta Fabi y Dr. Eugenio Ley Koo.

A mis compañeros de clase, en especial a Wenceslao, Enrique y Ramón por haberme enseñado lo lejos que se puede llegar cuando se ama a la ciencia.

A Celia, Elsa y Noemí, por todo el cuidado y hospitalidad que me brindaron durante casi cinco años en su hermosa casa del andador Jardín.

A la DGAPA, UNAM., por la beca de tesis.

Índice

Resumen	3
1. Introducción.	4
1.1 Conceptos básicos de dinámica de fluidos.	6
1.1.1 Ecuaciones que gobiernan al movimiento de un fluido newtoniano e incompresible.	6
a) Ecuación de continuidad.	7
b) Ecuación de conservación de cantidad de movimiento.	8
1.1.2 Existencia y unicidad de las soluciones a las ecuaciones de Navier-Stokes.	10
1.2 Introducción a la dinámica de fluidos computacional (CFD).	11
1.2.1 Desarrollo histórico.	11
1.2.2 Soluciones numéricas.	12
1.2.3 Discretización.	13
1.2.4 Convergencia de los algoritmos.	15
1.2.5 Métodos de residuos pesados.	16
1.2.6 Método de volumen finito.	18
2. Algoritmo de volumen finito en coordenadas generalizadas.	19
2.1 Geometría.	22
2.1.1 Transformación de coordenadas.	22
2.1.2 Bases covariante y contravariante.	23
2.1.3 El tensor métrico.	24
2.1.4 Interpretación geométrica de la transformación.	25
2.1.5 Cálculo de las áreas y el volumen de cada elemento.	27
2.1.6 Equivalencia de enfoques.	28
2.1.7 Interpolación lineal.	28
2.2 Las ecuaciones de movimiento y su discretización.	30
2.2.1 La ecuación de convección-difusión.	30
2.2.2 Discretización de la ecuación escalar de convección-difusión.	32
2.2.3 Coeficientes de matriz.	34
2.3 Algoritmo de acoplamiento velocidad-presión.	36
2.3.1 Ecuaciones de cantidad de movimiento.	36
2.3.2 Ecuación de corrección de presión.	38
2.3.3 Método de interpolación de Rhie-Chow.	41
3. Codificación.	43
3.1 Ecuaciones de movimiento.	43
3.2 Geometría.	43
3.3 NECRONOM.f: código de volumen finito en coordenadas generalizadas.	44
4. Generación de mallas.	47
4.1 Mapeos algebraicos.	47
4.1.1 Funciones de deformación unidimensionales.	47
4.1.2 Método de multisuperficies.	49
4.2 Aplicación numérica del mapeo algebraico.	51
4.2.1 MALLAS.f: código de generación de malla para un difusor.	51
4.2.2 TUBOTE.f: código de generación de malla para tubos rectos, con codos y en forma de "s".	54

5. Resultados.	55
5.1 Validación del código.	55
5.1.1 Descripción del problema.	55
5.1.2 Solución analítica.	56
5.1.3 Comparación de resultados numéricos y analíticos.	57
5.1.4 Perfil en desarrollo.	59
5.2 Tubo con codo.	60
5.2.1 Descripción del problema.	60
5.2.2 Presentación de resultados.	62
a) Campos de presiones relativas.	62
b) Campos de velocidades.	62
c) Perfiles de velocidad.	62
5.3 Tubo en forma de "s".	68
5.3.1 Descripción del problema.	68
5.3.2 Presentación de resultados.	69
a) Campos de presiones relativas.	69
b) Campos de velocidades.	69
c) Perfiles de velocidad.	72
6. Discusión y conclusiones.	75
Apéndice 1. Diferenciación covariante.	77
Apéndice 2. Ecuaciones dependientes del tiempo y condiciones de frontera.	79
Apéndice 3. Método híbrido.	81
Apéndice 4. Listado del programa NECRONOM.f.	84
A.4.1 Programa principal.	84
A.4.2 Subrutina INIT.	88
A.4.3 Subrutina GEOMGC.	89
A.4.4 Subrutina INITCOND.	95
A.4.5 Subrutina CALCOEF.	95
A.4.6 Subrutina FUENTE.	97
A.4.7 Subrutina PROMOD.	101
A.4.8 Subrutina LISOLV.	103
A.4.9 Subrutina RHIECHOW.	104
A.4.10 Subrutina RHICHW.	106
A.4.11 Subrutina CALP.	109
A.4.12 Subrutina MPGSOUT.	112
A.4.13 Subrutina SUSTITU.	113
A.4.14 Subrutina PRINT.	113
A.4.15 Archivo de parámetros param.f.	114
Apéndice 5. Listado de los programas generadores de mallas.	118
A.5.1 Listado del programa MALLAS.f.	118
A.5.2 Subrutina BOUNDSURF.	120
A.5.3 Subrutina STRECH.	121
A.5.4 Subrutina SURCH.	121
A.5.5 Subrutina ENSIGHT.	122
A.5.6 Programa TUBOTE.f.	123
Bibliografía.	126

Resumen

Se desarrolló un programa de cómputo capaz de resolver las ecuaciones de Navier-Stokes en coordenadas generalizadas. Se expone la transformación de estas ecuaciones de un sistema de coordenadas cartesianas a un sistema arbitrario de coordenadas. Los factores que aparecen como resultado de la transformación de coordenadas son identificados como propiedades geométricas tales como las áreas y el volumen de elementos infinitesimales. Se aplica el método de volumen finito para discretizar estas ecuaciones enfatizando el hecho de que se asegura el cumplimiento de las ecuaciones de conservación en volúmenes de control finitos. Se hace la codificación del método descrito y de la discretización espacial de dominios de estudio tridimensionales. Se hace la validación del código resolviendo numéricamente el flujo dentro de un tubo recto de sección cuadrada y comparando la solución numérica con la solución analítica de flujo totalmente desarrollado. Se aplica el código a dos ejemplos de tubos de sección cuadrada con curvatura. De los resultados se concluye que el código resuelve satisfactoriamente los flujos para pequeñas curvaturas y que deben hacerse mejores aproximaciones en la discretización de los términos convectivos y difusivos para reducir el esfuerzo de cómputo al resolver geometrías más complejas.

El programa de cómputo desarrollado en esta tesis es parte de un proyecto que pretende elaborar un programa general que permita el cálculo de flujos uni-fásicos, laminares o turbulentos con transferencia de masa y energía.

1. Introducción.

El objetivo de esta tesis es desarrollar un programa de cómputo capaz de resolver las ecuaciones de Navier-Stokes en geometrías arbitrarias tridimensionales.

La mayoría de los procesos naturales e industriales en los que algún tipo de fluido está presente toman lugar dentro de geometrías complicadas. La solución numérica de las ecuaciones que gobiernan el movimiento de los fluidos, en especial de las ecuaciones transformadas a sistemas de coordenadas generalizadas, hace posible un análisis rápido y de bajo costo para aquellos problemas donde hasta hace poco los datos experimentales eran la única manera de obtener información.

La técnica de coordenadas generalizadas ha sido desarrollada principalmente por Thompson (1978) y Thompson et al. (1982). Por otro lado, los métodos numéricos de volumen finito para ecuaciones de conservación han sido formulados por Patankar y Spalding (1972) y han sido explicados ampliamente en el libro de texto de Patankar (1980). La combinación de la técnica de coordenadas generalizadas y de volumen finito ha sido estudiada por Demirdzic (1982), Peric (1985), Shyy et al. (1985), Napolitano y Orlandi (1985), y Burns et al. (1987). Los resultados de estos primeros estudios han sido mejorados durante esta década por Melaaen (1992) y Majumdar et al. (1992). La técnica de discretización de volumen finito en geometrías arbitrarias sigue siendo tema actual de investigación en búsqueda de una base aún más sólida para la solución numérica de las ecuaciones que gobiernan a los fluidos dentro de geometrías arbitrarias.

En esta tesis se sigue la formulación teórica de Burns et al. (1987) y se toma como base el programa de cómputo en coordenadas cartesianas del M. en I. Rubén Avila, del Instituto Nacional de Investigaciones Nucleares. Existen diferencias sustanciales entre la formulación de volumen finito para coordenadas cartesianas y la formulación en coordenadas generalizadas. Por lo tanto, la mayor parte del esfuerzo realizado en esta tesis se enfoca en la programación del código.

En la siguiente sección se expone una breve introducción a los conceptos básicos de dinámica de fluidos. Ahí se deducen las ecuaciones de balance de masa y cantidad de movimiento para fluidos newtonianos e incompresibles en sistemas de coordenadas cartesianas. Dentro de esta misma introducción, también se incluyen algunos conceptos básicos de dinámica de fluidos computacional (CFD). Se explica su desarrollo histórico junto con la descripción general de algunas técnicas numéricas aplicadas a la solución de ecuaciones diferenciales.

El segundo capítulo trata la teoría desarrollada por Burns et al. (1987), que aplica la técnica de coordenadas generalizadas a la discretización de volumen finito de las ecuaciones de Navier-Stokes en geometrías arbitrarias. En este capítulo se deducen una serie de fórmulas que forman la base lógica que se aplica en la programación.

En el tercer capítulo se expone la explicación del código final y su diagrama de flujo. El listado del código se expone en el apéndice 4. Este contiene encabezados y comentarios que ayudan al usuario a modificar directamente el código fuente. Al final del apéndice 4 se hace un resumen de la nomenclatura utilizada para las variables de Fortran del programa.

En el cuarto capítulo se exponen un conjunto de técnicas algebraicas que permiten hacer la discretización espacial de los dominios de estudio. La técnica de discretización de los dominios de estudio es conocida como generación de mallas y ha sido desarrollada ampliamente por Thompson (1984). Se otorga especial importancia a la ortogonalidad de estas mallas cerca de las fronteras del dominio y a la capacidad de los programas generadores de malla para poder ajustar el número de nodos en las regiones donde una mayor resolución numérica es necesaria. Como base para la realización de los generadores de malla expuestos en ese capítulo se utiliza el libro de texto de Fletcher (1991).

En el quinto capítulo se presentan tres ejemplos de aplicación del programa desarrollado. Se hace la validación del código comparando la solución numérica obtenida para el flujo dentro de un tubo recto de sección cuadrada con el resultado analítico de flujo totalmente desarrollado. Posteriormente, el código se aplica a dos deformaciones de este tubo. La primera es un tubo con un codo en la mitad y la segunda es un tubo en forma de "s". Diversos autores han hecho estudios tanto experimentales como numéricos de tubos con curvatura. Por ejemplo, hace setenta años Nikuradse (1926) descubrió que dentro de tubos con curvatura, a parte del flujo en la dirección axial, existe un flujo secundario en el plano transversal, que se dirige hacia las paredes laterales del tubo, transportando cantidad de movimiento hacia esas caras. Efectos como éste han sido explicados parcialmente a mediados de siglo (Prandtl 1953, Brundrett y Baines 1964) y sólo recientemente, una combinación de experimentos y cálculos numéricos, han llevado a Humphrey et al. (1977) a explicar el efecto detalladamente.

Por último, en el sexto capítulo se exponen la discusión y las conclusiones generales de la tesis. Para optimizar este tipo de códigos, Melaaen (1992) y Majumdar et al. (1992) proponen una aproximación de segundo orden en los términos convectivos y difusivos de la ecuación de conservación de cantidad de movimiento. Además, según Patankar (1980) es trivial incorporar la ecuación de conservación de energía dada la estructura de la ecuación de convección-difusión estudiada en esta tesis. Ambas modificaciones serán motivo de estudios futuros.

1.1 Conceptos básicos de dinámica de fluidos.

En esta sección se introducen los conceptos de la dinámica de fluidos que llevan a las ecuaciones de Navier-Stokes para fluidos newtonianos e incompresibles para sistemas de coordenadas cartesianas. La transformación de estas ecuaciones a sistemas de coordenadas generalizadas se realiza en la sección 2.2. Se discuten las dificultades matemáticas alrededor de la existencia y unicidad de la solución de las mismas. Una exposición detallada de lo expuesto en esta sección puede encontrarse en libros de texto como Currie (1993) o Batchelor (1967).

1.1.1 Ecuaciones que gobiernan al movimiento de un fluido newtoniano e incompresible.

Los fluidos son materiales que se caracterizan por la facilidad con la que pueden ser deformados. A diferencia de los sólidos, los fluidos ceden ante fuerzas de corte cuando están en reposo, de tal forma que se produce un movimiento relativo entre sus capas que continúa hasta que se deja de aplicar la fuerza.

Los fluidos se clasifican a *grosso modo* en líquidos y gases. Los líquidos tienen un volumen definido que sólo varía con la temperatura y/o la presión. Estos se adaptan a la forma geométrica del recipiente que los contiene ocupando la fracción de volumen que les corresponde. En contraste, los gases se expanden dentro del recipiente que los contiene hasta ocupar su totalidad. En general, los gases son más fáciles de comprimir que los líquidos. Cualquier movimiento del fluido que involucre variaciones apreciables en la presión implica cambios en el volumen por unidad de masa mayores para un gas que para un líquido.

Para estudiar el comportamiento de los fluidos se recurre a la hipótesis del continuo, en la que se aproxima al fluido como una distribución continua de materia, sin tomar en cuenta el comportamiento de las moléculas individuales que lo componen. Esta manera de tratar a los fluidos es válida siempre y cuando el camino libre medio de las moléculas sea muy pequeño comparado con la escala física más pequeña del flujo bajo estudio. En el contexto de la hipótesis del continuo, en vez de moléculas, se habla de elementos o parcelas de fluido. Estos elementos son pequeños volúmenes de fluido que contienen una cantidad suficiente de moléculas como para que se pueda despreciar el efecto individual de ellas, y que las propiedades físicas de la parcela de fluido estén determinadas por el promedio de las propiedades físicas de las moléculas que la componen.

Para deducir las ecuaciones que describen al comportamiento de un fluido, se supone que en cada punto se tiene un valor único para la velocidad (\vec{U}), presión (p), densidad (ρ) y demás variables de campo, y que éstas varían continuamente con el tiempo y la posición. El fluido obedece a las leyes de conservación de masa y de cantidad de movimiento, y de ahí se obtiene un conjunto de ecuaciones diferenciales que involucran a las variables de campo,

de manera que al resolverlas se obtiene el valor de éstas para cada instante en cada punto del fluido.

Es necesario escoger un sistema de referencia en el cual formular las leyes de conservación. El sistema euleriano es aquel donde se suelen resolver la mayoría de los problemas. En él, las variables independientes son las coordenadas espacio-temporales (x,y,z,t) y las ecuaciones se deducen a partir de estudiar lo que ocurre en un volumen de control fijo en el espacio. El otro sistema de referencia es el lagrangiano, el cual se fija en una sola partícula de fluido y sigue su evolución en el tiempo. Aquí las variables independientes son el tiempo inicial y la posición de la partícula en ese instante (x_0, y_0, z_0, t_0) , pues la posición de la partícula para cualquier tiempo posterior se calcula a partir de las componentes de la velocidad.

La relación entre las derivadas lagrangianas y las eulerianas es la siguiente (Currie 1993):

$$\frac{D}{Dt}\alpha = \frac{\partial\alpha}{\partial t} + (\vec{U} \cdot \nabla)\alpha, \quad (1.1)$$

donde α es una variable de campo.

Utilizando el teorema de transporte de Reynolds (Currie 1993) es posible transformar la derivada lagrangiana o material de una integral de volumen a una integral de las derivadas eulerianas:

$$\frac{D}{Dt} \int_V \alpha dV = \int_V \left(\frac{\partial\alpha}{\partial t} + \nabla \cdot (\alpha\vec{U}) \right) dV. \quad (1.2)$$

a) Ecuación de continuidad.

Para un volumen de control arbitrario V , fijo en el espacio y tiempo, la conservación de la masa requiere que el cambio de masa dentro del volumen de control sea igual al flujo de masa que cruza la superficie S de V , y por lo tanto:

$$\frac{\partial}{\partial t} \int_V \rho dV = - \int_S \rho \vec{U} \cdot \vec{n} dS, \quad (1.3)$$

donde \vec{n} es el vector unitario normal a la superficie cuya dirección apunta al exterior del volumen de control. Usando el teorema de Gauss, la integral de superficie puede ser reemplazada por una integral de volumen. Entonces se obtiene:

$$\int_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) \right] dV = 0. \quad (1.4)$$

Y como es válida para cualquier tamaño y forma de V , la ecuación anterior implica que:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = 0, \quad (1.5)$$

que es la ecuación de conservación de masa o ecuación de continuidad.

Cuando las variaciones en la densidad del fluido pueden ignorarse, se dice que el fluido es incompresible, en este caso la ecuación (1.5) se reduce a:

$$\nabla \cdot \vec{U} = 0. \quad (1.6)$$

b) Ecuación de conservación de cantidad de movimiento.

A pesar de no soportar fuerzas de corte en el reposo, los fluidos pueden sostener y transmitir una fuerza de corte cuando se encuentran en movimiento. Este fenómeno introduce una propiedad llamada viscosidad dinámica. Se observa experimentalmente que el esfuerzo de corte viscoso τ (fuerza por unidad de área) entre dos capas adyacentes de fluido es proporcional al gradiente de velocidad normal a las capas. Si consideramos estas capas en el plano xy , con un flujo en la dirección x , obtenemos la siguiente relación:

$$\tau = \mu \frac{\partial U}{\partial y}, \quad (1.7)$$

donde μ es la constante de proporcionalidad conocida como coeficiente de viscosidad dinámica y U es la componente x de la velocidad. Los fluidos que cumplen con este modelo son conocidos como fluidos newtonianos debido a que fué Newton el primero en proponer esta relación.

La ecuación de conservación de cantidad de movimiento resulta de aplicar la segunda ley de movimiento de Newton a un volumen de fluido V en movimiento. Las fuerzas que actúan sobre la masa de fluido se clasifican en fuerzas volumétricas o másicas (como la gravitacional o la electromagnética) y fuerzas superficiales (como las fuerzas de presión y las fuerzas viscosas). Las fuerzas volumétricas dependen básicamente de la masa en cuestión y son fuerzas externas; se representan como fuerzas por unidad de volumen. Las fuerzas de superficie se deben a lo que el resto del fluido ejerce sobre un volumen dado de éste, son de corto alcance y se representan como fuerzas por unidad de área, es decir, esfuerzos. Las componentes de dichos esfuerzos pueden representarse como componentes

de un tensor de rango dos ($\vec{\sigma}$), de manera que σ_{ij} representa a la componente del esfuerzo que actúa en el plano $x_i = \text{cte}$, con $x_i = (x, y, z)$ para $i=1,2,3$, y dirigida en la dirección de x_j . Así se tiene que el esfuerzo que actúa en la dirección x_j es $\sigma_{ij}n_i$, donde n_i es la componente i -ésima del vector normal a la superficie del volumen de fluido. Si las fuerzas volumétricas están dadas por \vec{f} , entonces la conservación de la cantidad de movimiento implica que:

$$\frac{D}{Dt} \int_V \rho U_j dV = \int_S \sigma_{ij} n_i dS + \int_V f_j dV, \quad (1.8)$$

usando la convención de suma sobre índices repetidos.

Para el caso de un fluido newtoniano e incompresible, libre de fuerzas volumétricas, el tensor de esfuerzos σ_{ij} es:

$$\sigma_{ij} = -p\delta_{ij} + \mu \left[\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right], \quad (1.9)$$

donde p es la presión dinámica y δ_{ij} es la delta de Kronecker. Sustituyendo la expresión (1.9) en la ecuación (1.8) se tiene que:

$$\frac{D}{Dt} \int_V \rho U_j dV = \int_S \left(-p\delta_{ij} + \mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right) n_i dS, \quad (1.10)$$

Si la viscosidad y la densidad son constantes en todo el fluido, al aplicar los teoremas de Gauss y la relación entre derivadas lagrangianas y eulerianas (1.1), se puede llevar a la ecuación (1.10) a la forma:

$$\int_V \rho \left[\frac{\partial \vec{U}}{\partial t} + \vec{U} \cdot \nabla \vec{U} \right] dV = \int_V (-\nabla p + \mu \nabla^2 \vec{U}) dV, \quad (1.11)$$

que es válida para cualquier volumen de control y de ahí que se pueda obtener la ecuación de Navier-Stokes para un fluido newtoniano incompresible:

$$\rho \left[\frac{\partial \vec{U}}{\partial t} + \vec{U} \cdot \nabla \vec{U} \right] = -\nabla p + \mu \nabla^2 \vec{U}. \quad (1.12)$$

1.1.2 Existencia y unicidad de las soluciones a las ecuaciones de Navier-Stokes.

Una vez que se han logrado establecer las ecuaciones de balance de masa (1.6) y cantidad de movimiento (1.12), junto con las condiciones iniciales y de frontera (ver apéndice 2 para la descripción de estas condiciones), se cuenta con un esquema matemático completo que en principio describe la dinámica del movimiento de un fluido newtoniano e incompresible. El paso inmediato, concierne a la búsqueda de la solución al sistema de ecuaciones, para un conjunto particular de condiciones iniciales y de frontera.

Como es bien sabido, se conocen soluciones para condiciones geométricas con un alto grado de simetría, por ejemplo el flujo totalmente desarrollado dentro de un tubo de sección circular (flujo de Poiseuille) o el flujo entre dos cilindros rotatorios (flujo de Couette), etc., o bien en condiciones en las que algunos términos en las ecuaciones puedan ser ignorados debido a su tamaño relativamente pequeño, por ejemplo, los flujos estudiados en teoría de capa límite, etc. Los ejemplos en los que se ha podido encontrar la solución analítica a las ecuaciones de balance son clásicos y se discuten en la mayoría de los libros de texto como Batchelor (1967), Currie (1993), Schlichting (1979), etc. Sin embargo, en el presente no se conoce la solución general al sistema de ecuaciones de balance para condiciones arbitrarias. Más aún, la existencia y unicidad de las soluciones a las ecuaciones de balance es aún un problema abierto.

Desde hace más de medio siglo se conoce la existencia de soluciones "débiles" a las ecuaciones de Navier-Stokes (Leray 1934), construidas a partir de aproximaciones de Galerkin, en un esfuerzo por aplicar el método de Picard para demostrar la existencia de soluciones generales. Desafortunadamente, no se ha podido demostrar que las expresiones así obtenidas son efectivamente soluciones de las ecuaciones de balance. Es posible demostrar, para condiciones de frontera periódicas, que las aproximaciones de Galerkin indican que la energía cinética al tiempo t más la energía total disipada hasta ese mismo instante es menor o igual que la energía cinética inicial más el trabajo efectuado por la fuerza de cuerpo, pero no ha sido posible demostrar la igualdad (Doering y Gibbon 1995). Esta propiedad define a las soluciones "débiles".

Más aún, no se ha podido demostrar la unicidad de las soluciones "débiles", aunque se sabe que la posible no-unicidad está íntimamente relacionada con la suavidad de los campos de velocidad. Para una discusión amplia de las propiedades matemáticas de las soluciones a las ecuaciones de Navier-Stokes, consúltese Constantin y Foias (1988), y Doering y Gibbon (1995).

En innumerables casos de interés, no se conoce la solución analítica a las ecuaciones de balance. A pesar de la profunda incertidumbre en cuanto a su naturaleza y propiedades, muchas aplicaciones en campos tan diversos como astronomía, meteorología, ingeniería mecánica y química, etc., requieren de métodos de solución que permitan conocer los campos de velocidad y de presión bajo determinadas condiciones iniciales y de frontera. Esta situación constituye la motivación principal para desarrollar los métodos numéricos para la solución de las ecuaciones de balance.

1.2 Introducción a la dinámica de fluidos computacional (CFD).

En esta sección se introducen brevemente algunos conceptos utilizados en la CFD.

1.2.1 Desarrollo histórico.

Desde el tiempo de Newton, la dinámica de fluidos se estableció como ciencia y comenzó a ser aplicada. Su desarrollo teórico se ha enfocado en la construcción y solución de las ecuaciones que gobiernan las diferentes categorías de fluidos en movimiento y en el estudio de las aproximaciones involucradas.

Las ecuaciones que gobiernan la dinámica de fluidos newtonianos, llamadas de Navier-Stokes, han sido conocidas por más de 150 años. Sin embargo, todavía hay mucho que hacer tanto en el desarrollo de formas reducidas de estas ecuaciones como en su solución.

Los estudios experimentales han validado y delineado los límites de las aproximaciones en las ecuaciones. Desafortunadamente, muchos estudios experimentales suelen ser sumamente costosos y es necesario utilizar un método alternativo de validación.

El constante aumento de la velocidad y de la memoria de las computadoras desde mediados de este siglo, ha hecho nacer a la CFD. Esta rama de la dinámica de fluidos complementa a los estudios teóricos y experimentales con una alternativa económica en la simulación de flujos a escala real, y ha hecho posible avances teóricos para condiciones de flujo todavía inalcanzables experimentalmente.

Tal vez, la razón principal para el desarrollo de la CFD haya sido su bajo costo en comparación con las pruebas de túnel de viento, y será aun menor en el futuro. Hasta ahora han sido una constante tanto el mejoramiento como el abaratamiento de los equipos de cómputo, por lo tanto, para un algoritmo y un problema dados, el costo relativo de la simulación computacional ha bajado significativamente de manera histórica (Fig. 1.1).



Fig. 1.1. Costo relativo de cómputo para un algoritmo y flujo dados (Chapman 1979).

1.2.2 Soluciones numéricas.

El proceso total para determinar la información práctica sobre el comportamiento de fluidos en movimiento puede ser representado como en la Fig. 1.2.

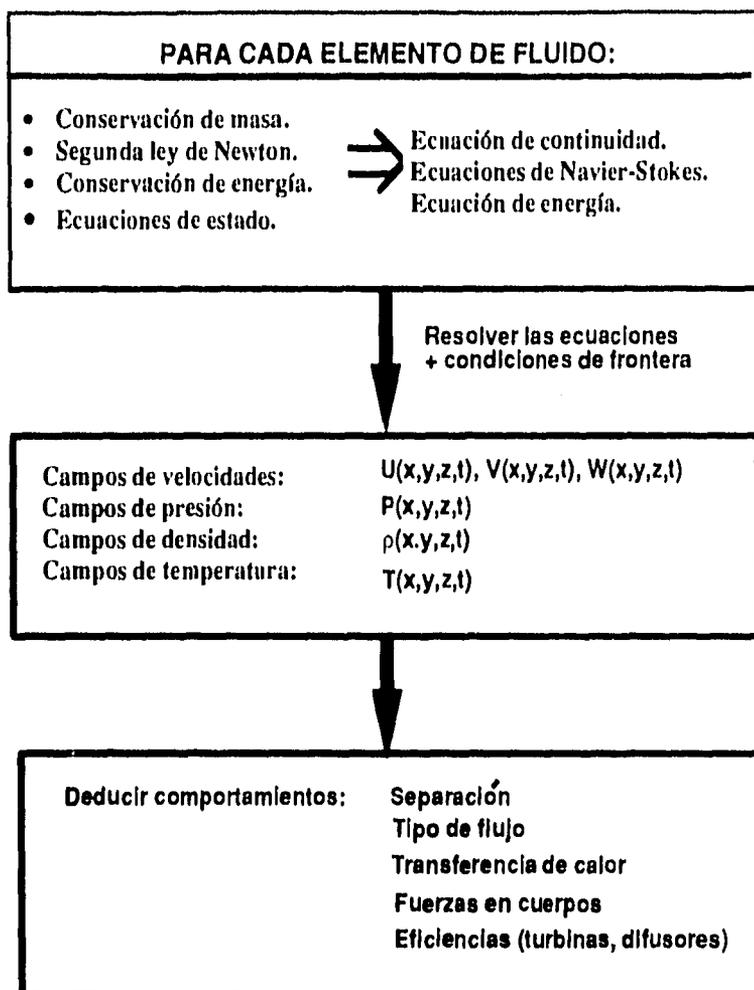


Fig. 1.2. Obtención de información práctica en mecánica de fluidos computacional

Cuando las ecuaciones de conservación son lo suficientemente complicadas como para no tener una solución exacta al alcance de nuestras técnicas matemáticas, es necesario buscar una solución numérica. Las técnicas computacionales reemplazan a las ecuaciones diferenciales parciales con un sistema de ecuaciones algebraicas con el propósito de que la computadora pueda ser utilizada para obtener la solución.

Para métodos locales, como las diferencias finitas, elementos finitos y volúmenes finitos, las ecuaciones algebraicas relacionan los valores de las variables dependientes en puntos adyacentes. Se entiende entonces que una red discreta de puntos, donde estas variables serán calculadas, es distribuida a lo largo del dominio de estudio, tanto espacial como temporalmente.

1.2.2 Soluciones numéricas.

El proceso total para determinar la información práctica sobre el comportamiento de fluidos en movimiento puede ser representado como en la Fig. 1.2.

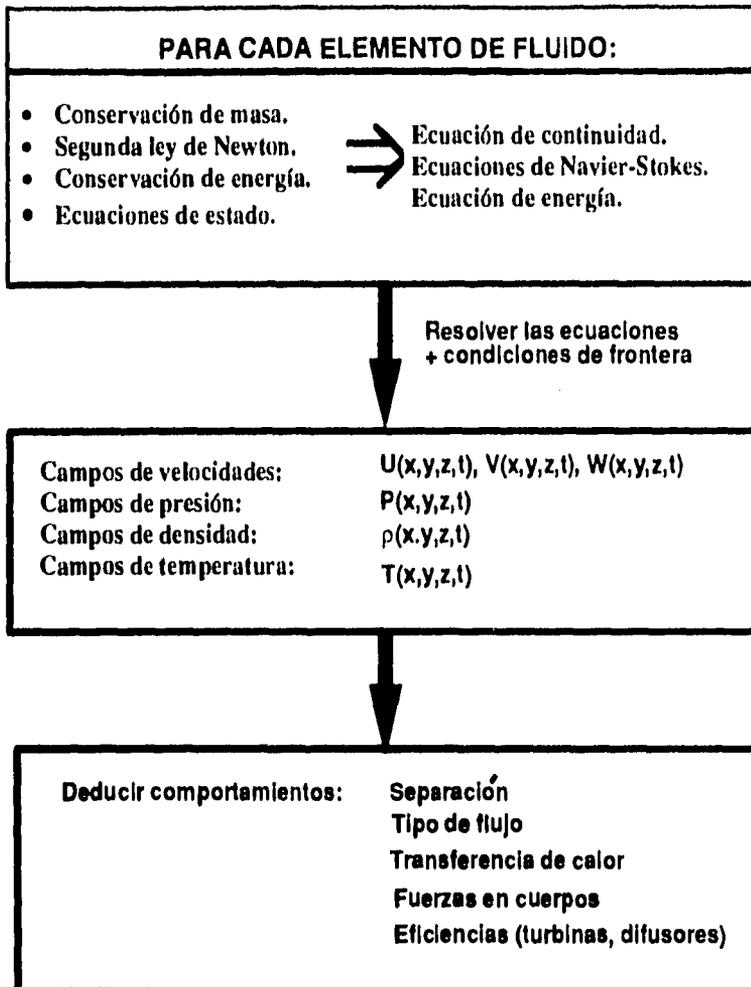


Fig. 1.2. Obtención de información práctica en mecánica de fluidos computacional

Cuando las ecuaciones de conservación son lo suficientemente complicadas como para no tener una solución exacta al alcance de nuestras técnicas matemáticas, es necesario buscar una solución numérica. Las técnicas computacionales reemplazan a las ecuaciones diferenciales parciales con un sistema de ecuaciones algebraicas con el propósito de que la computadora pueda ser utilizada para obtener la solución.

Para métodos locales, como las diferencias finitas, elementos finitos y volúmenes finitos, las ecuaciones algebraicas relacionan los valores de las variables dependientes en puntos adyacentes. Se entiende entonces que una red discreta de puntos, donde estas variables serán calculadas, es distribuida a lo largo del dominio de estudio, tanto espacial como temporalmente.

Consecuentemente, el proceso de convertir a las ecuaciones continuas de conservación a un sistema de ecuaciones algebraicas es conocido como discretización. El número de puntos requeridos en la malla para obtener una solución de buena precisión depende típicamente de la dimensionalidad, de la complejidad geométrica y de la magnitud de los gradientes de las variables dependientes. Por ejemplo, para obtener el flujo alrededor de un avión completo es necesaria una malla de un poco más de diez millones de puntos, y para cada punto es posible tener desde cinco (para flujo laminar) hasta treinta (para flujo turbulento) variables dependientes (Fletcher 1991).

Dado que las ecuaciones gobernantes de la dinámica de fluidos son en general no lineales, el cómputo de la solución es hecha de forma iterativa. Esto es, la solución para cada variable dependiente en cada punto del dominio es corregida en secuencia usando las ecuaciones discretas. El número de iteraciones hasta que se alcance la convergencia, puede variar desde unos cuantos cientos hasta muchos miles.

El proceso de discretización introduce un error que puede ser reducido, en principio, refinando la malla y mientras las ecuaciones discretas sigan siendo una representación confiable de las ecuaciones gobernantes. La solución computacional podrá acercarse arbitrariamente a la solución verdadera refinando la malla, si hay suficientes recursos de cómputo.

1.2.3 Discretización.

La Fig. 1.3. muestra los pasos necesarios para la obtención de una solución numérica en su forma más simple.

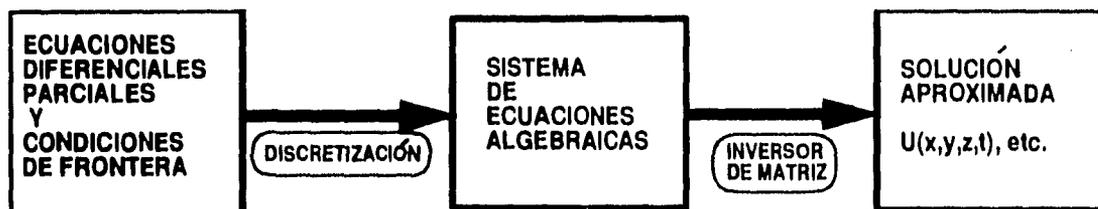


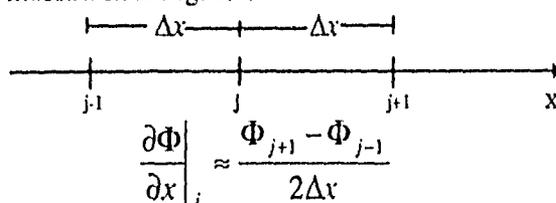
Fig 1.3. Simplificación del proceso computacional de solución.

La primera parte convierte las ecuaciones diferenciales parciales y sus condiciones auxiliares (iniciales y de frontera) en un sistema de ecuaciones algebraicas donde las variables sólo se definen en puntos discretos del espacio. Esta parte es llamada discretización. Este proceso es fácilmente identificable si se trata de los métodos de diferencias finitas pero no es tan obvio en los métodos de elemento finito o de volumen finito. Al reemplazar los términos diferenciales por expresiones algebraicas que conectan a los valores nodales en la malla, introducimos un error que depende del tipo de ecuación y de la expresión algebraica. Una discusión amplia de esto es expuesta en libros de texto (Fletcher 1991).

La segunda parte del proceso de solución requiere de un inversor de matriz para dar solución al sistema de ecuaciones algebraicas. Aquí también se introduce un error que es usualmente despreciable en comparación al error por discretización a menos de que el algoritmo sea inestable.

El cambio de las ecuaciones continuas por sistemas algebraicos involucra el cambio de las derivadas espaciales y temporales por términos que contengan valores nodales de las variables.

Un ejemplo trivial del cambio de las derivadas por expresiones algebraicas son las diferencias finitas centradas como se muestra en la Fig. 1.4.



$$\left. \frac{\partial \Phi}{\partial x} \right|_j \approx \frac{\Phi_{j+1} - \Phi_{j-1}}{2\Delta x}$$

Fig 1.4. Puntos para el cálculo de diferencias centradas.

Con el fin de ayudar a la deducción de fórmulas de diferencias finitas para mayores órdenes de aproximación y para diferentes arreglos de puntos (comúnmente llamados moléculas numéricas), se expone una técnica metódica para construir las (Fletcher 1991). Se empieza con la expresión para tres puntos:

$$\left[\frac{\partial \Phi}{\partial x} \right]_j = a\Phi_{j-1} + b\Phi_j + c\Phi_{j+1} + O(\Delta x^m), \quad (1.13)$$

donde a, b y c serán determinadas y el término $O(\Delta x^m)$ indicará la precisión de dicha aproximación.

Si ahora se hace la expansión en series de Taylor de los términos $j+1$ y $j-1$, como sigue:

$$\Phi_{j+1} = \sum_{m=0}^{\infty} \frac{(\Delta x)^m}{m!} \left(\frac{\partial^m \Phi}{\partial x^m} \right)_j, \quad (1.14)$$

$$\Phi_{j-1} = \sum_{m=0}^{\infty} \frac{(-\Delta x)^m}{m!} \left(\frac{\partial^m \Phi}{\partial x^m} \right)_j. \quad (1.15)$$

Entonces se puede escribir a la expresión (1.13) de la siguiente forma:

$$\begin{aligned} a\Phi_{j-1} + b\Phi_j + c\Phi_{j+1} &= (a+b+c)\Phi_j + (-a+c)\Delta x \left[\frac{\partial \Phi}{\partial x} \right]_j \\ &+ (a+c) \frac{\Delta x^2}{2} \left[\frac{\partial^2 \Phi}{\partial x^2} \right]_j + (-a+c) \frac{\Delta x^3}{6} \left[\frac{\partial^3 \Phi}{\partial x^3} \right]_j + \dots \end{aligned} \quad (1.16)$$

y he aquí donde se decide que tipo de aproximación se quiere.

Por ejemplo, poniendo $a + b + c = 0$ y $(-a + c)\Delta x = 1 \Rightarrow a = c - \frac{1}{\Delta x}$ y $b = -2c + \frac{1}{\Delta x}$ para cualquier c . Y escogiendo c de tal forma que el término cuadrático desaparezca de la expresión (1.16), es decir, $c = -a$, que implica $b = 0$, tenemos:

$$\left. \frac{d\Phi}{dx} \right|_j \approx \frac{\Phi_{j+1} - \Phi_{j-1}}{2\Delta x} - \frac{\Delta x^2}{6} \left[\frac{\partial^3 \Phi}{\partial x^3} \right]_j + \dots,$$

entonces

$$\left. \frac{d\Phi}{dx} \right|_j \approx \frac{\Phi_{j+1} - \Phi_{j-1}}{2\Delta x} \quad (1.17)$$

con un error por truncamiento de $O(\Delta x^2)$, recuperando la fórmula de diferencias centradas.

Esta misma técnica puede ser utilizada para obtener aproximaciones para derivadas de orden más alto, para obtener fórmulas que involucren nodos hacia un solo lado, para fórmulas multidimensionales y hasta para mallas no uniformes.

1.2.4 Convergencia de los algoritmos.

Una pregunta muy importante sobre las soluciones numéricas es qué garantía se tiene de que la solución numérica de las ecuaciones diferenciales parciales esté cercana a la exacta y bajo qué circunstancias la numérica coincidirá con ella. La segunda parte de esta pregunta puede ser contestada superficialmente si se pide que la solución computacional converja a la solución exacta cuando Δt y Δx disminuyan a cero. Sin embargo, no existe prueba directa de la convergencia para las ecuaciones que gobiernan el movimiento de los fluidos. Es por esto que suele tomarse una ruta indirecta (Fig. 1.5).

La ruta indirecta provee de una guía cualitativa para buscar la convergencia pero no ha podido formularse para mallas finitas o para ecuaciones no lineales. Bajo esta base, las condiciones sobre las cuales se justifique la convergencia deberán ser tomadas como necesarias pero no como suficientes.

Asumiendo esto, se invoca el teorema de equivalencia de Lax: "Dado un problema lineal de valores iniciales propiamente definido y dada una aproximación a él por diferencias finitas que satisfaga la condición de consistencia, la estabilidad es la condición necesaria y suficiente para la convergencia". Con esto se tiene una forma alternativa para poder asegurar convergencia si se prueba la estabilidad y la consistencia del algoritmo.

CONSISTENCIA + ESTABILIDAD \longrightarrow CONVERGENCIA

La consistencia implica que el proceso de discretización puede ponerse en reversa por medio de una expansión en series de Taylor y así recuperar las ecuaciones diferenciales.

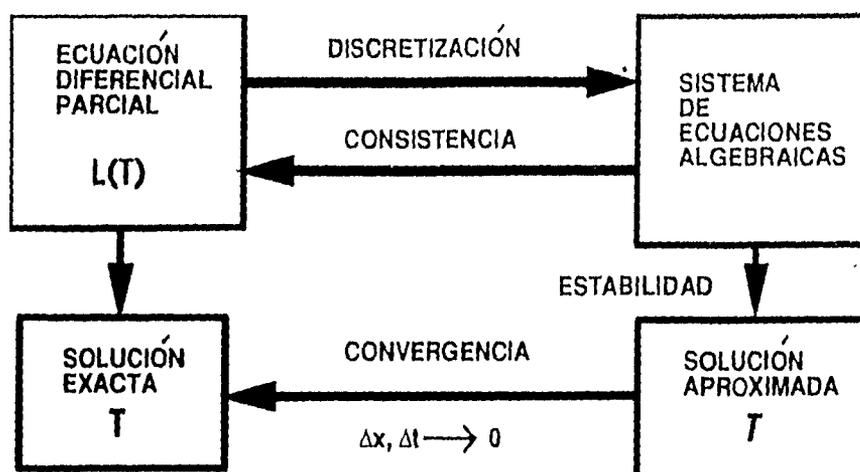


Fig. 1.5. Relación conceptual entre consistencia, estabilidad y convergencia.

El concepto de estabilidad engloba el crecimiento o decaimiento de los errores introducidos en cualquier parte del algoritmo. Estos errores no son debidos a procesos lógicos incorrectos sino a que los números en la memoria de la computadora tienen una aproximación decimal finita. Existen métodos para determinar la estabilidad de un algoritmo pero son sólo estrictamente aplicables a ecuaciones lineales por lo que es común "congelar" la no linealidad localmente para su uso. También existe un método de energía que puede manejar algunas ecuaciones no lineales directamente (Richtmyer y Morton 1967). Aunque cualquiera de estos métodos requiere de una nueva evaluación numérica y se vuelve a las condiciones necesarias pero no suficientes.

Cuando es complicado asegurar la consistencia de un algoritmo es preferible replantearlo bajo la estrategia de correcciones diferidas (Smith 1985). En este método se calcula una solución preliminar y se utiliza para evaluar el término principal en el error de truncamiento. Este término se añade a la ecuación discreta original como término fuente (del lado derecho) y una solución mejorada es obtenida bajo un procedimiento iterativo. Este método es utilizado en el capítulo siguiente en los términos debidos a la no ortogonalidad de la malla.

1.2.5 Métodos de residuos pesados.

Un método poderoso para resolver ecuaciones diferenciales es el método de residuos pesados que es descrito en detalle por Finlayson (1972). Se representa a la ecuación diferencial de la siguiente forma:

$$L(\phi) = 0. \quad (1.18)$$

Se supone una solución aproximada que contiene un número de parámetros indeterminados, por ejemplo:

$$\tilde{\phi} = \phi_0(x, y, z) + \sum_{j=1}^J a_j(t) \phi_j(x, y, z), \quad (1.19)$$

donde las a_j son los parámetros.

La sustitución de la expresión (1.19) en la ecuación (1.18) deja un residuo R de tal forma que:

$$L(\tilde{\phi}) = R, \quad (1.20)$$

y para reducirlo se integra sobre el dominio de estudio

$$\iiint W_m(x, y, z) R dx dy dz = 0. \quad (1.21)$$

Poniendo $m=1, \dots, M$, se genera un sistema de ecuaciones para las a_j .

Diferentes formas de escoger las funciones de peso W_m llevan a diferentes métodos. Algunos de estos son:

i) Método de subdominios.

El dominio de cómputo es partido en M subdominios D_m que pueden intersectarse y donde

$$W_m = \begin{cases} 1 \rightarrow \text{en } D_m, \\ 0 \rightarrow \text{fuera.} \end{cases}$$

ii) Método de colocación.

$$W_m(\bar{x}) = \delta(\bar{x} - \bar{x}_m),$$

donde δ es la delta de Dirac y $\bar{x} = (x, y, z)$.

iii) Método de mínimos cuadrados.

$$W_m = \frac{\partial R}{\partial a_m} \Rightarrow \iiint R^2 dx dy dz \quad \text{sea mínimo.}$$

iv) Método de Galerkin.

$$W_m(x, y, z) = \phi_m(x, y, z).$$

En este último, las funciones de peso se escogen entre la misma familia de funciones que son usadas en la aproximación (1.19). Este método interseca con los métodos de elemento finito usados en dinámica de fluidos y se le puede dar una interpretación variacional si se proponen a los coeficientes a_j como los valores nodales de la variable dependiente y a las ϕ_m como funciones de interpolación de esos valores.

1.2.6 Método de volumen finito.

Este método es similar al método de subdominios, con la particularidad de que éstos no se intersecan entre sí y de que no es necesario proponer una solución aproximada como la expresión (1.19).

La idea básica de la formulación de volumen finito o volumen de control es fácil de entender y lleva por sí misma a una interpretación física. El dominio de cálculo es dividido en un número finito de volúmenes de control que no se intersecan, de tal manera que exista un volumen de control por cada punto en la malla. Las ecuaciones diferenciales parciales son integradas en cada volumen de control, los valores de las variables son considerados constantes o con perfiles previamente definidos dentro del volumen, de tal forma que las integrales pueden ser evaluadas dando como resultado un conjunto de ecuaciones discretas con los valores nodales de las variables como incógnitas.

Las ecuaciones discretas obtenidas de esta manera expresan el principio de conservación para una cantidad ϕ (ver ecuación 1.18), en el volumen de control, justo como la ecuación diferencial lo expresaba para un volumen de control infinitesimal.

Lo mas atractivo de la formulación de volumen finito está en que la convergencia implica la conservación de cantidades como la masa, el momento y la energía, en todo el grupo de volúmenes de control y por tanto en todo el dominio.

1. Algoritmo de volumen finito en coordenadas generalizadas.

Este capítulo presenta la teoría requerida para la descripción del programa NECRONOM.f (ver apéndice 4). NECRONOM.f es parte de una serie de códigos, escritos en lenguaje Fortran 77, cuya finalidad es formar un paquete de propósitos generales que permita el cómputo de flujos laminares o turbulentos (modelo $\kappa - \epsilon$, Fletcher 1991) con transferencia de calor y masa. Estos programas han sido formulados bajo la estrategia de discretización de volumen finito convencional, en variables primitivas (velocidad y presión), y han sido capaces de resolver flujos tanto en dos dimensiones (2D) como en tres dimensiones (3D) para geometrías cúbicas o compuestas por paralelepípedos regulares. La contribución de este trabajo consiste en la formulación y consititución de un código que permita el tratamiento de geometrías arbitrarias en 3D, heredando al mismo tiempo las propiedades de simplicidad y rapidez de las versiones que sólo manejan geometrías cartesianas.

Algunas características distintivas del código son:

a) Sistemas de coordenadas curvilíneas o generalizadas.

Este método ha sido desarrollado recientemente (Thompson et al, 1982) para permitir que los métodos de diferencias finitas trabajen con geometrías complicadas. La idea básica es usar una transformación de coordenadas para mapear el dominio complicado a un dominio simple y regular donde las ecuaciones diferenciales parciales (1.5) y (1.12), puedan ser resueltas de manera más simple (Fig. 2.1).

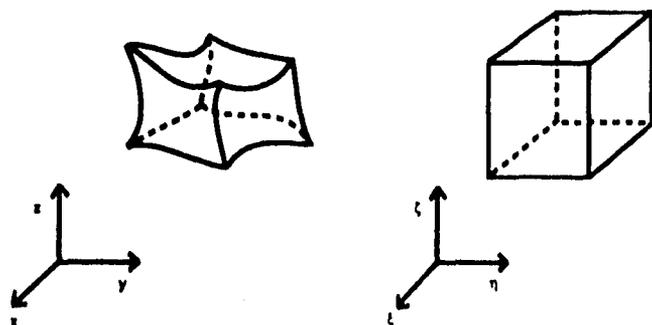


Fig. 2.1. Transformación de un dominio arbitrario a uno simple.

Al reemplazar el sistema de coordenadas cartesiano $(x^i) = (x,y,z)$ por un sistema de coordenadas curvilíneas $(\xi^i) = (\xi,\eta,\zeta)$, tales que las fronteras correspondan a superficies de $\xi^i = \text{cte}$ (Fig. 2.2), se gana una manera sencilla de expresar las condiciones de frontera del problema pero a su vez se requiere trabajar con unas ecuaciones diferenciales parciales más complicadas debido a la naturaleza no lineal de la transformación.

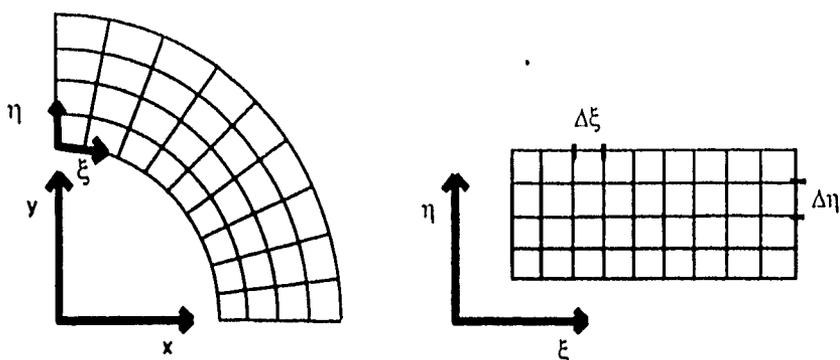


Fig. 2.2. Mapeo de un dominio en el espacio físico al espacio de coordenadas generalizadas.

Algunos problemas con geometrías simples pueden ser resueltos haciendo uso de la transformación analítica entre los sistemas coordenados. Este es el caso de aquellos problemas donde la simetría esférica, cilíndrica, toroidal, etc., es evidente. Pero debido a que las geometrías a resolver son complicadas, la transformación es definida numéricamente.

En este capítulo se supone que una malla de coordenadas curvilíneas, ortogonales en las fronteras (por simplicidad al definir las condiciones de frontera), es conocida y se discuten las estrategias de discretización y solución de las ecuaciones de movimiento de la mecánica de fluidos en esa malla. La estrategia de generación de malla se presenta en detalle en el capítulo 4.

b) Malla no defasada espacialmente.

El defasamiento en los lugares donde las variables primitivas son calculadas es conocido en inglés como *staggered* (Fig. 2.3) y es una técnica común en los códigos de volumen finito. Su virtud es evitar ciertos problemas de inestabilidad debido a la introducción de oscilaciones de tipo tablero de ajedrez dentro del proceso numérico en los valores de la velocidad y de la presión (Patankar 1980). Sin embargo, este defasamiento aplicado a geometrías arbitrarias implica el cálculo laborioso de una gran cantidad de variables por cada punto en la malla. Este problema se evita resolviendo las ecuaciones de cantidad de movimiento para las componentes cartesianas de la velocidad en una malla donde todas las variables primitivas son calculadas en los centroides de los volúmenes de control que componen la geometría (Fig. 2.4), y aplicando un método de interpolación adecuado para obtener los valores de las componentes de la velocidad en las caras de los volúmenes de control. El método de interpolación utilizado se conoce con el nombre de Rhie-Chow (Rhie 1981, Rhie y Chow 1983) y toma en cuenta que la variación de la velocidad depende a su vez de la presión, evitando los problemas de inestabilidad antes mencionados. El método de interpolación de Rhie-Chow se discute en detalle en la sección 2.3.3.

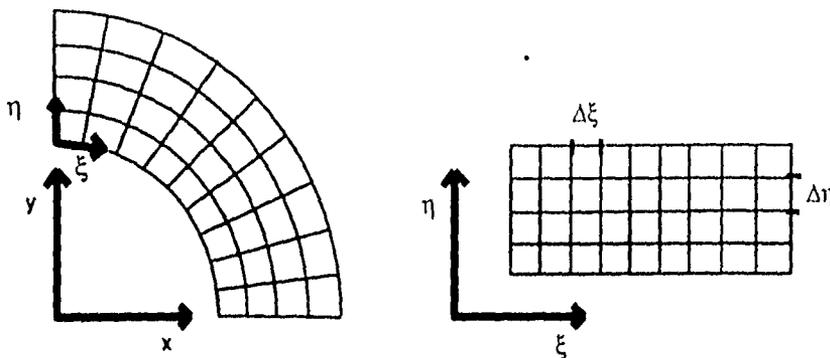


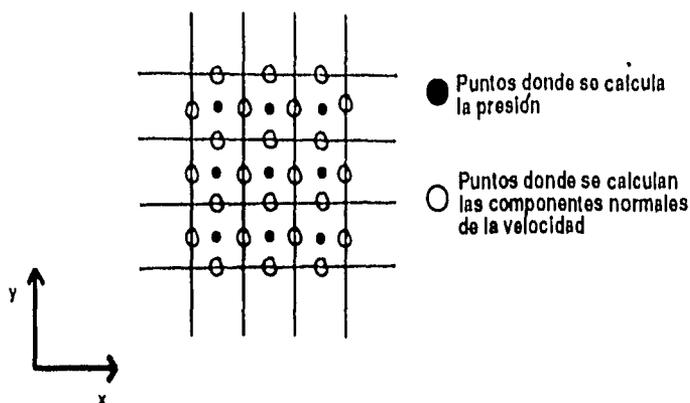
Fig. 2.2. Mapeo de un dominio en el espacio físico al espacio de coordenadas generalizadas.

Algunos problemas con geometrías simples pueden ser resueltos haciendo uso de la transformación analítica entre los sistemas coordenados. Este es el caso de aquellos problemas donde la simetría esférica, cilíndrica, toroidal, etc., es evidente. Pero debido a que las geometrías a resolver son complicadas, la transformación es definida numéricamente.

En este capítulo se supone que una malla de coordenadas curvilíneas, ortogonales en las fronteras (por simplicidad al definir las condiciones de frontera), es conocida y se discuten las estrategias de discretización y solución de las ecuaciones de movimiento de la mecánica de fluidos en esa malla. La estrategia de generación de malla se presenta en detalle en el capítulo 4.

b) Malla no defasada espacialmente.

El defasamiento en los lugares donde las variables primitivas son calculadas es conocido en inglés como *staggered* (Fig. 2.3) y es una técnica común en los códigos de volumen finito. Su virtud es evitar ciertos problemas de inestabilidad debido a la introducción de oscilaciones de tipo tablero de ajedrez dentro del proceso numérico en los valores de la velocidad y de la presión (Patankar 1980). Sin embargo, este defasamiento aplicado a geometrías arbitrarias implica el cálculo laborioso de una gran cantidad de variables por cada punto en la malla. Este problema se evita resolviendo las ecuaciones de cantidad de movimiento para las componentes cartesianas de la velocidad en una malla donde todas las variables primitivas son calculadas en los centroides de los volúmenes de control que componen la geometría (Fig. 2.4), y aplicando un método de interpolación adecuado para obtener los valores de las componentes de la velocidad en las caras de los volúmenes de control. El método de interpolación utilizado se conoce con el nombre de Rhie-Chow (Rhie 1981, Rhie y Chow 1983) y toma en cuenta que la variación de la velocidad depende a su vez de la presión, evitando los problemas de inestabilidad antes mencionados. El método de interpolación de Rhie-Chow se discute en detalle en la sección 2.3.3.

Fig. 2.3. Malla defasada (*staggered*).

Al transformar las ecuaciones de cantidad de movimiento del espacio físico al computacional empleando una sola malla para todas las variables, las componentes cartesianas de la velocidad son tratadas como escalares y no hay coeficientes de conexión (símbolos de Christoffel) en las ecuaciones finales de movimiento. La codificación es más simple y es mucho menor el espacio de memoria requerido para almacenar la información geométrica en comparación al empleado en los métodos con malla defasada.

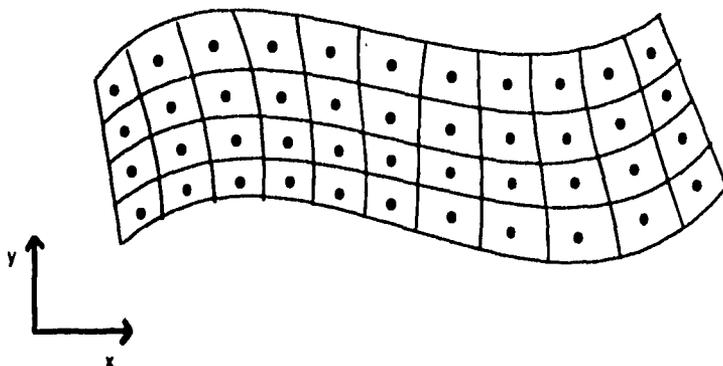


Fig. 2.4. Malla curvilínea no defasada.

A lo largo de este capítulo se explican cada uno de los comentarios presentados en esta breve introducción, haciendo énfasis en la existencia de dos puntos de vista complementarios que, para nuestros propósitos, son equivalentes. En el primero se usan las herramientas del cálculo tensorial para transformar las ecuaciones de movimiento de la geometría complicada en el espacio físico a la geometría regular en el espacio computacional y, una vez ahí, se hace la discretización de las ecuaciones. En el segundo enfoque se trabaja enteramente con la malla distorsionada del espacio físico. Las ecuaciones se discretizan aplicando las leyes de conservación del espacio físico a cada volumen de control, y toda la información geométrica es incorporada en términos de los volúmenes y vectores de área normales de cada volumen de control. Ambos enfoques guardan una estrecha relación y son conceptos diferentes sólo en apariencia. Esto se discute con mayor detalle en la sección 2.1.6.

2.1 Geometría.

En esta sección se discuten las ideas básicas de cálculo tensorial que serán útiles en las secciones siguientes, puntualizando la relación que hay entre ciertos conceptos matemáticos abstractos con la geometría de coordenadas curvilíneas (Aris 1962, Burns y Wilkes 1987).

2.1.1 Transformación de coordenadas.

Utilizaremos la siguiente notación:

$(x^i)_{i=1,2,3} = (x,y,z)$: Coordenadas cartesianas que definen el espacio físico.

$(\xi^i)_{i=1,2,3} = (\xi,\eta,\zeta)$: Coordenadas curvilíneas no ortogonales que definen el espacio computacional.

$J^i_j = \frac{\partial x^i}{\partial \xi^j}$: Matriz jacobiana.

$\bar{J}^j_i = \frac{\partial \xi^j}{\partial x^i} = (J^{-1})^j_i$: Matriz jacobiana inversa.

$|J| = \det(J^i_j)$: Determinante jacobiano.

$A^i_j = |J| \bar{J}^i_j$: Matriz jacobiana adjunta.

Se supone que la transformación de coordenadas es no singular y por lo tanto que las funciones $x^i(\xi^j)$, $\xi^j(x^i)$ son uno a uno. Consecuentemente, la matriz jacobiana tampoco es singular, así que su inversa siempre existe y su determinante es siempre distinto de cero. Más aún, se pide que la transformación esté orientada de manera positiva, es decir, que un sistema de referencia local de mano derecha en el espacio computacional sea mapeado a un sistema de mano derecha en el espacio físico y viceversa. Esto ocurre si y sólo si el determinante jacobiano es positivo en todo el dominio. La importancia principal de la matriz jacobiana y su inversa es que, por la regla de la cadena, relacionan las derivadas en el espacio físico con las derivadas en el espacio computacional. Para una función escalar ϕ dependiente de la posición se tiene que:

$$\frac{\partial \phi}{\partial \xi^i} = \frac{\partial x^j}{\partial \xi^i} \frac{\partial \phi}{\partial x^j} = J^j_i \frac{\partial \phi}{\partial x^j}, \quad (2.1)$$

$$\frac{\partial \phi}{\partial x^i} = \frac{\partial \xi^j}{\partial x^i} \frac{\partial \phi}{\partial \xi^j} = \bar{J}^j_i \frac{\partial \phi}{\partial \xi^j}, \quad (2.2)$$

usando la convención de suma sobre índices repetidos.

De estas ecuaciones, (2.2) requiere especial atención ya que las derivadas dentro de las ecuaciones (1.5) y (1.12), originalmente en coordenadas cartesianas, deben sustituirse en términos de las coordenadas curvilíneas. Con esto se gana una malla rectangular y uniforme en el espacio computacional, donde se aproximan las nuevas derivadas por operadores de diferencias finitas.

2.1.2 Bases covariante y contravariante.

Para cualquier sistema de coordenadas curvilíneas (ξ^i) se pueden asociar a cada punto dos conjuntos distintos de vectores base:

a) Base contravariante, normal a las superficies coordenadas:

$$\bar{e}^{(i)} = \frac{\partial \xi^i}{\partial \bar{x}} = \frac{\partial \xi^i}{\partial x^k} \hat{i}_k = \bar{J}_k^i \hat{i}_k, \quad (2.3)$$

donde \hat{i}_k representa los vectores unitarios de la base cartesiana.

b) Base covariante, tangente a las curvas coordenadas:

$$\bar{e}_{(i)} = \frac{\partial \bar{x}}{\partial \xi^i} = \frac{\partial x^k}{\partial \xi^i} \hat{i}_k = J_i^k \hat{i}_k. \quad (2.4)$$

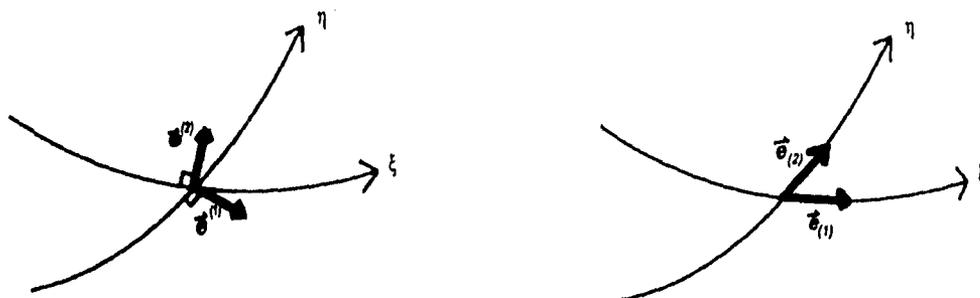


Fig. 2.5. Bases contravariante (izquierda) y covariante (derecha) para un sistema de coordenadas curvilíneas dado.

De esto se puede deducir que:

$$\bar{e}^{(i)} \cdot \bar{e}_{(j)} \Leftrightarrow \text{la transformación es ortogonal y}$$

$$\bar{e}^{(i)} = \bar{e}_{(i)} \Leftrightarrow \text{la transformación es ortonormal.}$$

De las expresiones (2.3) y (2.4) se observa que la base covariante equivale a los vectores columna de la matriz jacobiana, mientras que la base contravariante equivale a los vectores renglón de la matriz jacobiana inversa.

Nótese que por convención, los índices encerrados entre paréntesis denotan vectores que son miembros de una colección, mientras que los índices que no están encerrados entre paréntesis, denotan componentes de un vector.

Bajo estas bases, todo vector cartesiano $\vec{V} = (V_i)$ puede ser expresado de las dos siguientes maneras:

$$\vec{V} = V_i \vec{e}^{(i)} \Rightarrow V_i' = J_i^j V_j, \quad (2.5)$$

$$\vec{V} = V''^i \vec{e}_{(i)} \Rightarrow V''^i = \bar{J}_j^i V_j, \quad (2.6)$$

donde V_i' y V''^i son llamadas las componentes covariantes y contravariantes de \vec{V} respectivamente.

2.1.3 El tensor métrico.

Los incrementos en las coordenadas del espacio físico Δx^k pueden relacionarse a cambios en las coordenadas del espacio computacional $\Delta \xi^i$ por:

$$\Delta x^k = \frac{\partial x^k}{\partial \xi^i} \Delta \xi^i, \quad (2.7)$$

$$\begin{aligned} \Rightarrow \Delta s^2 &= \sum_{k=1}^3 \Delta x^k \Delta x^k = \sum_{k=1}^3 \left(\frac{\partial x^k}{\partial \xi^i} \Delta \xi^i \right) \left(\frac{\partial x^k}{\partial \xi^j} \Delta \xi^j \right) \\ &= g_{ij} \Delta \xi^i \Delta \xi^j, \end{aligned} \quad (2.8)$$

$$\text{donde} \quad g_{ij} = \sum_{k=1}^3 \frac{\partial x^k}{\partial \xi^i} \frac{\partial x^k}{\partial \xi^j}, \quad (2.9)$$

y recibe el nombre de tensor métrico debido a que relaciona la distancia con incrementos infinitesimales en las coordenadas (Aris 1962).

Este tensor, en su forma normal o inversa, puede ser escrito en términos de los vectores base covariantes y contravariantes de la siguiente manera:

$$g_{ij} = \vec{e}_{(i)} \bullet \vec{e}_{(j)} = J_i^k J_j^k, \quad (2.10)$$

$$g^{ij} = \vec{e}^{(i)} \bullet \vec{e}^{(j)} = \bar{J}_k^i \bar{J}_k^j, \quad (2.11)$$

$$g_j^i = \vec{e}^{(i)} \bullet \vec{e}_{(j)} = \bar{J}_k^i J_j^k = \delta_j^i. \quad (2.12)$$

Las normas euclidianas y los productos escalares de vectores respecto a sus componentes contravariantes y covariantes están dadas por:

$$\vec{V} \bullet \vec{V} = g_{ij} V''^i V''^j = g^{ij} V_i V_j'. \quad (2.13)$$

Además, el tensor métrico y su inverso pueden ser utilizados para subir o bajar índices, es decir, para cambiar de base:

$$V'^i = g^{ij}V'_j, \quad V'_j = g_{ij}V'^i. \quad (2.14)$$

2.1.4 Interpretación geométrica de la transformación.

Las relaciones entre el tensor métrico, su inverso y las bases covariante y contravariante, tienen una interpretación geométrica simple e instructiva.

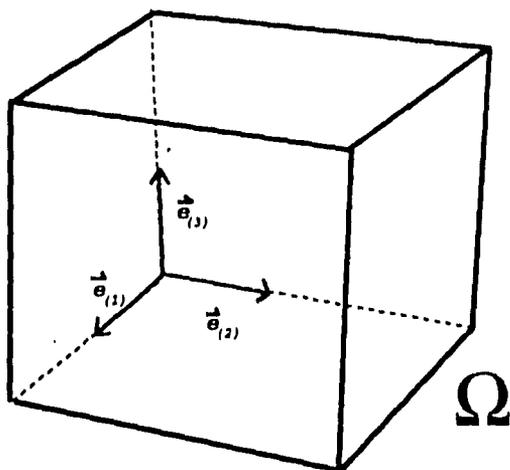


Fig. 2.6. Paralelepípedo generado por la base covariante.

Sea Ω el paralelepípedo generado por la triada $\bar{e}_{(i)}$ (Fig. 2.6), sean $\vec{A}^{(i)}$ ($i = 1, 2, 3$) los vectores de área normales a la superficie de Ω (Fig. 2.7), es decir, los vectores normales a las caras de Ω y con magnitudes iguales a las áreas de las caras, y sea $vol\Omega$ el volumen de Ω . Entonces, tenemos que:

$$\vec{A}^{(1)} = \bar{e}_{(2)} \times \bar{e}_{(3)}, \quad \vec{A}^{(2)} = \bar{e}_{(3)} \times \bar{e}_{(1)}, \quad \vec{A}^{(3)} = \bar{e}_{(1)} \times \bar{e}_{(2)}.$$

$$\bar{e}_{(1)} \cdot \vec{A}^{(1)} = \bar{e}_{(1)} \cdot (\bar{e}_{(2)} \times \bar{e}_{(3)}) = vol\Omega,$$

de manera análoga
$$\vec{A}^{(i)} \cdot \bar{e}_{(j)} = vol\Omega \delta_j^i. \quad (2.15)$$

Por otro lado, el determinante jacobiano también tiene una relación directa con la base covariante:

$$|J| = \begin{vmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} & \frac{\partial x^1}{\partial \xi^3} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} & \frac{\partial x^2}{\partial \xi^3} \\ \frac{\partial x^3}{\partial \xi^1} & \frac{\partial x^3}{\partial \xi^2} & \frac{\partial x^3}{\partial \xi^3} \end{vmatrix} = \bar{e}_{(1)} \cdot (\bar{e}_{(2)} \times \bar{e}_{(3)}) = vol\Omega. \quad (2.16)$$

Y dado que:

$$\vec{e}^{(i)} \cdot \vec{e}_{(j)} = \delta_j^i,$$

$$\vec{A}^{(i)} \cdot \vec{e}_{(j)} = vol\Omega \delta_j^i,$$

entonces

$$\vec{A}^{(i)} = vol\Omega \vec{e}^{(i)} = |J| \vec{e}^{(i)}, \quad (2.17)$$

con las siguientes componentes cartesianas:

$$A_k^{(i)} = |J| \bar{J}_k^i = A_k^i. \quad (2.18)$$

Esto quiere decir que las componentes cartesianas de los vectores de área forman los renglones de la matriz jacobiana adjunta, y que junto con las otras relaciones, el determinante jacobiano y la matriz jacobiana adjunta, evaluados en un punto P , tienen una interpretación simple en términos del volumen y los vectores de área del paralelepípedo generado por los vectores de la base covariante de referencia en P .

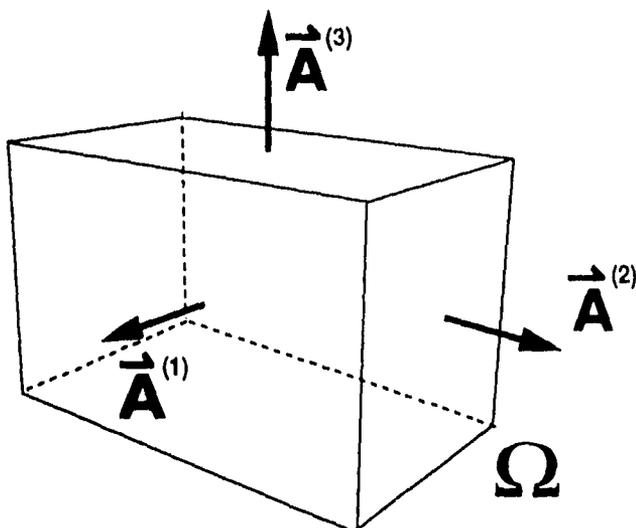


Fig. 2.7. Vectores de área del paralelepípedo generado por la base covariante.

Más aún, es posible interpretar las componentes del tensor métrico y de su inverso en términos de la geometría de Ω :

$$g_{ij} = \vec{e}_{(i)} \cdot \vec{e}_{(j)} = \|\vec{e}_{(i)}\| \|\vec{e}_{(j)}\| \cos\theta_{ij} \quad (2.19)$$

donde $\cos\theta_{ij}$ son los cosenos directores de Ω ,

$$g^{ij} = \vec{e}^{(i)} \cdot \vec{e}^{(j)} = \frac{\vec{A}^{(i)} \cdot \vec{A}^{(j)}}{|J|^2} \quad (2.20)$$

Si denotamos a g como el determinante del tensor métrico, dado que $g = J^T J$, donde las negritas significan que son matrices, se obtiene que $g = |J|^2$, esto es:

$$\sqrt{g} = |J| = vol\Omega \quad (2.21)$$

De hecho, \sqrt{g} es utilizado comúnmente como un elemento de volumen infinitesimal en el punto P .

Usando las relaciones anteriores, es posible conseguir que toda la información que se requiere en la transformación de coordenadas sea relegada al cálculo de los volúmenes y áreas de los elementos que componen el espacio físico, es decir, los volúmenes de control. Es conveniente hacer el mapeo de tal manera que los volúmenes de control se conviertan en **elementos unitarios** ($\Delta\xi^{(i)} = 1$) en el espacio computacional.

Por otro lado, en vez de trabajar con las componentes contravariantes de los campos vectoriales como el de la velocidad, es conveniente utilizar las componentes normales de flujo. Estas están definidas de la siguiente manera:

$$\hat{V}^i = |J|V'^i = \sqrt{g}V'^i = A_j^i V^j \quad (2.22)$$

$$\vec{V} \cdot \vec{A}^{(i)} = V'^j \vec{e}_{(j)} \cdot |J| \vec{e}^{(i)} = |J|V'^j \delta_j^i = |J|V'^i$$

entonces
$$\vec{V} \cdot \vec{A}^{(i)} = \hat{V}^i \quad (2.23)$$

Por lo tanto, y como lo indica su nombre, las componentes normales de flujo son tan solo los productos escalares de \vec{V} con los vectores de área de Ω .

2.1.5 Cálculo de las áreas y el volumen de cada elemento.

Para realizar este cálculo, se supone que cada uno de los elementos del dominio es un hexaedro no necesariamente regular, es decir, que está formado por ocho vértices y seis caras aproximadamente planas pero no necesariamente ortogonales. Para que esto sea cierto, es necesario que la forma de definir la malla en el dominio físico no tenga discontinuidades. Esto es equivalente a pedir que los puntos estén distribuidos sobre curvas suaves.

El volumen de un dominio arbitrario Ω en el espacio físico puede ser expresado enteramente en términos de la integral de área sobre su frontera, usando la ley de Gauss:

$$\int_{\Omega} \nabla \cdot \vec{W} dV = \int_{\partial\Omega} \vec{W} \cdot d\vec{A},$$

donde \vec{W} es un campo vectorial arbitrario. Ahora bien, aplicada al campo vectorial $\vec{W} \equiv \vec{x}$,

se tiene que
$$vol\Omega = \frac{1}{3} \int_{\partial\Omega} \vec{x} \cdot d\vec{A}, \quad (2.24)$$

y haciendo uso de la aproximación de caras planas,

$$vol\Omega = \frac{1}{3} \sum_{nn} \left(\int_{\partial\Omega} \vec{x} \cdot d\vec{A} \right)_{nn} \equiv \frac{1}{3} \sum_{nn} \left(\vec{A} \cdot \frac{\int_{\partial\Omega} \vec{x} dS}{\|\vec{A}\|} \right)_{nn} = \frac{1}{3} \sum_{nn} \vec{A}_{nn} \cdot \vec{x}_{nn} \quad (2.25)$$

donde el subíndice nn denota las seis caras del hexaedro, \vec{x}_{nn} son las componentes cartesianas de los centroides de cada una de las caras y dS es el elemento infinitesimal de superficie.

Cada vector de área se obtiene de la siguiente manera: se separa la cara correspondiente en dos triángulos, luego se calcula el vector de área de cada triángulo haciendo el semi-producto cruz entre los vectores que componen dos de sus lados, y por último se suman ambos vectores para obtener el vector de área de la cara completa.

Una característica interesante de la fórmula aproximada (2.25) es que hace obvia la siguiente propiedad: el volumen total del dominio sólo depende de la distribución de la malla frontera y no de la distribución de los nodos interiores, debido a que las contribuciones de las superficies internas se cancelan por pares.

2.1.6 Equivalencia de enfoques.

La matriz jacobiana y su adjunta se aproximan haciendo uso de los vectores de área y los volúmenes mostrados en las ecuaciones (2.18) y (2.21). En este enfoque hay menos énfasis en la presencia de la transformación de coordenadas y más en el hecho de que se ha llenado el dominio físico con un arreglo de volúmenes de control de forma irregular en cada uno de los cuales impondremos leyes discretas de conservación. De hecho, es posible calcular las derivadas espaciales sin hacer ninguna referencia a la matriz jacobiana inversa aplicando la ley de Gauss a un volumen de control en el espacio físico:

$$\int_{\Omega} \frac{\partial \phi}{\partial x^i} dV = \int_{\partial\Omega} \phi dA_i \Rightarrow V_p \left(\frac{\partial \phi}{\partial x^i} \right)_p \approx \sum_{nn} (A_i \phi)_{nn} .$$

Y es completamente equivalente al cálculo de las derivadas en el espacio físico usando las expresiones para las derivadas del espacio computacional (2.2) y las relaciones (2.18) y (2.21), previamente obtenidas para la matriz jacobiana inversa en términos del volumen y los vectores de área de la célula de control.

2.1.7 Interpolación lineal.

En algunas operaciones, es necesario conocer el valor de las variables en las caras de los volúmenes de control. Las distancias entre nodos en el espacio físico no son regulares y por consiguiente es necesario efectuar interpolaciones. Debido a que en la mayoría de las mallas la distancia entre nodos es pequeña, es suficientemente preciso usar un método de interpolación lineal pesada basada en las distancias relativas en el espacio físico.

$$vol\Omega = \frac{1}{3} \sum_{nn} \left(\int_{\partial\Omega} \vec{x} \cdot d\vec{A} \right)_{nn} \equiv \frac{1}{3} \sum_{nn} \left(\vec{A} \cdot \frac{\int \vec{x} dS}{\|\vec{A}\|} \right)_{nn} = \frac{1}{3} \sum_{nn} \vec{A}_{nn} \cdot \vec{x}_{nn} \quad (2.25)$$

donde el subíndice nn denota las seis caras del hexahedro, \vec{x}_{nn} son las componentes cartesianas de los centroides de cada una de las caras y dS es el elemento infinitesimal de superficie.

Cada vector de área se obtiene de la siguiente manera: se separa la cara correspondiente en dos triángulos, luego se calcula el vector de área de cada triángulo haciendo el semi-producto cruz entre los vectores que componen dos de sus lados, y por último se suman ambos vectores para obtener el vector de área de la cara completa.

Una característica interesante de la fórmula aproximada (2.25) es que hace obvia la siguiente propiedad: el volumen total del dominio sólo depende de la distribución de la malla frontera y no de la distribución de los nodos interiores, debido a que las contribuciones de las superficies internas se cancelan por pares.

2.1.6 Equivalencia de enfoques.

La matriz jacobiana y su adjunta se aproximan haciendo uso de los vectores de área y los volúmenes mostrados en las ecuaciones (2.18) y (2.21). En este enfoque hay menos énfasis en la presencia de la transformación de coordenadas y más en el hecho de que se ha llenado el dominio físico con un arreglo de volúmenes de control de forma irregular en cada uno de los cuales impondremos leyes discretas de conservación. De hecho, es posible calcular las derivadas espaciales sin hacer ninguna referencia a la matriz jacobiana inversa aplicando la ley de Gauss a un volumen de control en el espacio físico:

$$\int_{\Omega} \frac{\partial \phi}{\partial x^i} dV = \int_{\partial\Omega} \phi dA_i \Rightarrow V_p \left(\frac{\partial \phi}{\partial x^i} \right)_p \approx \sum_{nn} (A_i \phi)_{nn} .$$

Y es completamente equivalente al cálculo de las derivadas en el espacio físico usando las expresiones para las derivadas del espacio computacional (2.2) y las relaciones (2.18) y (2.21), previamente obtenidas para la matriz jacobiana inversa en términos del volumen y los vectores de área de la célula de control.

2.1.7 Interpolación lineal.

En algunas operaciones, es necesario conocer el valor de las variables en las caras de los volúmenes de control. Las distancias entre nodos en el espacio físico no son regulares y por consiguiente es necesario efectuar interpolaciones. Debido a que en la mayoría de las mallas la distancia entre nodos es pequeña, es suficientemente preciso usar un método de interpolación lineal pesada basada en las distancias relativas en el espacio físico.

Considérese el punto P como en la Fig. 2.8 en el centro del volumen de control. Delante de cada cara sigue otro volumen de control con otro nodo en su centro, este nodo vecino está representado por N , y el punto centroide de la cara que divide a un volumen de control de su vecino está representado por n . Así pues, la barra \overline{nN} significa la distancia desde el punto n sobre la cara vecina al punto interno N del volumen vecino en el espacio físico. Como hay seis vecinos, deben considerarse los puntos w, e, s, n, b y t , asociados a los puntos denotados por sus mayúsculas respectivas. Estas representan las direcciones locales oeste, este, sur, norte, abajo y arriba respectivamente y se asocian por pares a las direcciones ξ, η, ζ del espacio computacional.

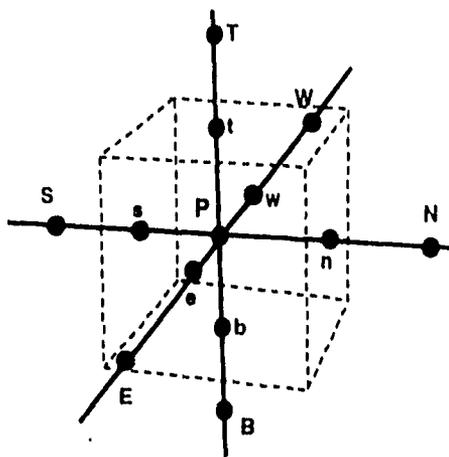


Fig. 2.8. Nodos alrededor de un volumen de control con centroide P .

Como notación se usa una barra en la parte inferior de la variable que es interpolada. De esta forma, la interpolación lineal pesada se expresa, con base en distancias relativas en el espacio físico, y para un escalar Φ , como sigue:

$$\underline{\Phi}_n = \frac{\overline{nN}\Phi_P + \overline{Pn}\Phi_N}{\overline{nN} + \overline{Pn}} \Rightarrow \underline{\Phi}_n = (1 - W_n)\Phi_P + W_n\Phi_N,$$

$$W_n = \frac{\overline{Pn}}{\overline{Pn} + \overline{nN}}. \quad (2.26)$$

Y esto requiere del almacenamiento de tres factores de peso de interpolación lineal W_n por cada nodo de la malla.

2.2 Las ecuaciones de movimiento y su discretización.

El precio que ha de ser pagado por la simplicidad de las expresiones en las condiciones de frontera al usar sistemas de coordenadas curvilíneas, es el incremento de complejidad en las ecuaciones de movimiento transformadas al sistema no ortogonal de coordenadas (ξ^i) . En esta sección se identifica ese aumento de complejidad y se interpreta físicamente como una transformación de un problema de convección-difusión isótropo en el espacio físico, a un problema anisótropo del mismo tipo en el espacio computacional.

La idea principal, en su forma más simple, es tomar las ecuaciones de movimiento en el espacio físico (1.5) y (1.12), y reemplazar todas las derivadas en coordenadas cartesianas por derivadas en el espacio computacional haciendo uso de las ecuaciones (2.1) y (2.2). Hacer esto en tres dimensiones es algebraicamente muy laborioso, y es mucho más conveniente expresar las ecuaciones en su forma tensorial covariante y explotar la notación introducida por el cálculo tensorial.

En este punto es conveniente recordar la siguiente relación para la divergencia covariante de un vector:

$$\nabla \cdot \vec{v} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} (\sqrt{g} V^{ii}) = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} \hat{V}^i, \quad (2.27)$$

donde \hat{V}^i son las componentes normales de flujo del campo \vec{v} como se define en la sección 2.1.4. En el apéndice 1 se presenta la discusión de la derivación covariante requerida para la deducción de este último resultado.

2.2.1 La ecuación de convección-difusión.

Debido a que todas las variables en las ecuaciones (1.5) y (1.12), incluyendo las componentes cartesianas de la velocidad, son tratadas como escalares, la ecuación de convección-difusión es el prototipo de ecuación que se debe resolver. Sin perder generalidad, se trabaja con la ecuación en estado estacionario. Las modificaciones extras debidas al término no estacionario no requieren de análisis tensorial y son discutidas en el apéndice 2.

La ecuación de convección-difusión en su forma cartesiana y en estado estacionario es:

$$\frac{\partial}{\partial x^i} (\rho U^i \Phi) = \frac{\partial}{\partial x^i} \left(\Gamma \frac{\partial \Phi}{\partial x^i} \right) + S, \quad (2.28)$$

donde ρ es la densidad del fluido, U^i es la velocidad del fluido, Γ es la difusividad escalar, S es un término fuente y Φ es un campo escalar. Es conveniente escribir la ecuación (2.28) en la forma equivalente:

$$\nabla \cdot \vec{I} = \frac{\partial}{\partial x^i} I^i = S, \quad (2.29)$$

donde \vec{I} es llamado el flujo total (convectivo + difusivo) y está dado por:

$$I^i = \rho U^i \Phi - \Gamma \frac{\partial \Phi}{\partial x^i} . \quad (2.30)$$

Usando la ley de Gauss, la ecuación (2.29) es equivalente a la ley de balance de ϕ en forma integral para cualquier dominio Ω en el espacio físico:

$$\int_{\Omega} \nabla \cdot \vec{I} dV = \int_{\partial\Omega} \vec{I} \cdot d\vec{A} = \int_{\Omega} S dV . \quad (2.31)$$

Las ecuaciones (2.29) y (2.30) son muy útiles para transformar a las coordenadas curvilíneas. Usando la expresión (2.27) para la divergencia covariante se obtiene de manera inmediata:

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} (\sqrt{g} I'^i) = S \Leftrightarrow \frac{\partial}{\partial \xi^i} \hat{I}^i = \sqrt{g} S , \quad (2.32)$$

y por lo tanto se recupera la misma forma que aparece en la ecuación (2.29), con el flujo total efectivo expresado en componentes contravariantes:

$$\begin{aligned} I'^i &= \rho U'^i \Phi - \Gamma g^{ij} \frac{\partial \Phi}{\partial \xi^j} , \\ \Rightarrow \hat{I}^i &= \sqrt{g} I'^i = \rho \hat{U}^i \Phi - \Gamma \sqrt{g} g^{ij} \frac{\partial \Phi}{\partial \xi^j} . \end{aligned} \quad (2.33)$$

En resumen, las ecuaciones de movimiento en el espacio computacional son:

$$\frac{\partial}{\partial \xi^i} \hat{I}^i = \sqrt{g} S , \quad \hat{I}^i = \rho \hat{U}^i \Phi - \Gamma^{ij} \frac{\partial \Phi}{\partial \xi^j} , \quad (2.34)$$

donde

$$\hat{U}^i = \sqrt{g} U'^i = \vec{U} \cdot \vec{A}^{(i)} \quad (2.35)$$

y

$$\Gamma^{ij} = \sqrt{g} g^{ij} \Gamma . \quad (2.36)$$

Esta es efectivamente otra ecuación de convección-difusión con difusividad anisótropa Γ^{ij} definida por la expresión (2.36). Nótese que Γ^{ij} es un tensor simétrico en virtud de que g^{ij} es simétrico. También, es no diagonal si y sólo si g^{ij} es no diagonal, así que el tensor de difusión es ortótropo si y sólo si la transformación de coordenadas es ortogonal, y es completamente anisótropo si y sólo si la transformación de coordenadas no es ortogonal.

Debido a su importancia en el proceso de discretización subsecuente, el tensor que multiplica a Γ en la expresión (2.36) tiene un nombre especial. Los coeficientes geométricos de difusión se definen como:

$$G^{ij} = \sqrt{g} g^{ij} . \quad (2.37)$$

Haciendo uso de la ecuación (2.20), los coeficientes geométricos de difusión pueden ser expresados en términos de los vectores de área y el volumen de los elementos del espacio físico:

$$G^{ij} = \sqrt{g} g^{ij} = \frac{\vec{A}^{(i)} \cdot \vec{A}^{(j)}}{\text{vol}\Omega} . \quad (2.38)$$

Hay que hacer énfasis en el hecho de que el mismo resultado en la ecuación (2.34) se obtiene si se hace el cambio de variable en la ecuación integral (2.31):

$$\int_{\partial\Omega'} \hat{I} \cdot d\bar{A}' = \int_{\Omega'} \frac{\partial}{\partial \xi^i} \hat{I}^i dV' = \int_{\Omega'} S|J| dV' \Rightarrow \frac{\partial}{\partial \xi^i} \hat{I}^i = \sqrt{g}S. \quad (2.39)$$

2.2.2 Discretización de la ecuación escalar de convección-difusión.

Para discretizar la ecuación de convección-difusión se procede de la siguiente manera. Integrando la ecuación (2.39) sobre un volumen de control con centro P en el espacio computacional, donde los volúmenes de control son cubos unitarios, se obtiene:

$$\left[\hat{I}^1 \right]_{w}^e + \left[\hat{I}^2 \right]_s^n + \left[\hat{I}^3 \right]_b^t = \langle \sqrt{g}S \rangle_P. \quad (2.40)$$

Los corchetes del lado derecho indican un promedio integral sobre el volumen de control.

Es además instructivo recuperar la ecuación (2.40) de la ecuación de conservación (2.31) para un volumen de control en el espacio físico. En este:

$$\bar{I}_w \cdot \bar{A}_w + \bar{I}_e \cdot \bar{A}_e + \bar{I}_s \cdot \bar{A}_s + \bar{I}_n \cdot \bar{A}_n + \bar{I}_b \cdot \bar{A}_b + \bar{I}_t \cdot \bar{A}_t = \text{vol}\Omega_P \langle S \rangle_P. \quad (2.41)$$

La ecuación (2.40) se sigue de la ecuación (2.41) aproximando el volumen por el determinante jacobiano, y aproximando los vectores de área por los renglones de la matriz jacobiana adjunta, como se describe en la sección 2.1.4. Por lo tanto, la ecuación (2.40) describe a las ecuaciones discretas apropiadas para la formulación en el espacio computacional, mientras que la ecuación (2.41) es la discretización asociada naturalmente con la formulación en el espacio físico, donde los volúmenes y las áreas son calculados con mayor precisión.

Trabajando con los volúmenes de control en el espacio físico, se puede volver a deducir la ecuación (2.36) para las componentes normales de flujo. Se tiene que:

$$\begin{aligned} (\bar{I} \cdot \bar{A})_{nn} &= \int_{A_{nn}} \bar{I} \cdot d\bar{A} = \int_{A_{nn}} (\rho \bar{U} \Phi - \Gamma \nabla \Phi) \cdot d\bar{A}, \\ \Rightarrow \hat{I}_{nn}^i &= \left(\rho \bar{U} \cdot \bar{A}^{(i)} \Phi - \Gamma \nabla \Phi \cdot \bar{A}^{(i)} \right)_{nn}, \end{aligned} \quad (2.42)$$

donde: $nn=w, e$ si $i=1$; $nn=s, n$ si $i=2$; $nn=b, t$ si $i=3$.

Pero $\bar{U} \cdot \bar{A}^{(i)} = \hat{U}^i$, y el último término se expresa en términos de las derivadas en el espacio computacional como sigue:

$$\nabla \Phi \cdot \bar{A}^{(i)} = \frac{\partial \Phi}{\partial x^k} A_k^i = \frac{A_k^j}{|J|} \frac{\partial \Phi}{\partial \xi^j} A_k^i = \frac{A_k^j A_k^i}{|J|} \frac{\partial \Phi}{\partial \xi^j}. \quad (2.43)$$

Sustituyendo la expresión (2.43) en la ecuación (2.42) se obtiene:

$$\hat{I}_{nn}^i = \left(\rho \hat{U}^i \Phi - \Gamma \frac{\vec{A}^{(i)} \cdot \vec{A}^{(j)}}{\sqrt{g}} \frac{\partial \Phi}{\partial \xi^j} \right)_{nn}. \quad (2.44)$$

Ahora se debe aproximar el flujo total por medio de diferencias finitas. Las ecuaciones discretizadas (2.40) y (2.41) utilizan los promedios del flujo efectivo total sobre las caras del volumen de control, y dicho flujo se escribe de la siguiente forma:

$$\hat{I}_{nn}^i = C_{nn}^i \Phi_{nn} - D_{nn}^{ij} \left(\frac{\partial \Phi}{\partial \xi^j} \right)_{nn}, \quad (2.45)$$

donde C_{nn}^i , D_{nn}^{ij} son los coeficientes de convección y de difusión anisótropa definidos por:

$$C_{nn}^i = \left(\rho \hat{U}^i \right)_{nn} = \left(\rho \vec{U} \cdot \vec{A}^{(i)} \right)_{nn}, \quad (2.46)$$

$$D_{nn}^{ij} = \Gamma_{nn}^{ij} = \Gamma_{nn} G_{nn}^{ij}. \quad (2.47)$$

Dado que todas las variables se calculan en los mismos lugares (los centroides de los volúmenes de control) de la misma malla, los coeficientes de convección son los mismos para todas las variables, así en cada iteración deben ser calculados una sola vez.

Para el cálculo de los coeficientes de convección, se requiere conocer los valores de la velocidad en las caras del volumen de control. Para obtenerlas, es necesario hacer una interpolación de los valores de la velocidad conocidos en los centroides de los volúmenes. Esta interpolación que, además de tomar en consideración los valores de la velocidad en los centroides, incorpora información del gradiente de presiones, se conoce como el algoritmo de Rhie-Chow y es discutida en la sección 2.3.3.

El cálculo de los coeficientes de difusión requiere los valores de Γ^{ij} en las caras del volumen de control. La difusividad Γ es evaluada en las caras del volumen de control por interpolación lineal desde los centroides de los volúmenes, usando los factores de peso discutidos en la sección 2.1.7. G^{ij} sólo depende de los vectores de área y del volumen del elemento y no necesita interpolarse. Los productos punto entre áreas incluidos en G^{ij} se hacen contra las caras contiguas y del mismo elemento. Por ejemplo, el vector de área oeste (w) sólo puede contraerse con los vectores de área sur (s) y abajo (b), además de sí mismo; de la misma manera quedan relacionadas las áreas este (e), norte (n) y arriba (t).

Por último, se deben discutir las representaciones en diferencias finitas de los valores de Φ en las caras del volumen de control y de las derivadas en el espacio computacional $\frac{\partial \Phi}{\partial \xi^i}$, que ocurren en la expresión (2.45) para el vector de flujo total. Los términos que multiplican a las componentes de la diagonal de D^{ij} son derivadas computacionales normales a las caras del volumen de control y esos son los términos difusivos que aparecen en el caso cartesiano. Los términos que multiplican a lo que está fuera de la diagonal de D^{ij} sólo aparecen si la malla es estrictamente no ortogonal, y estas son derivadas en el espacio computacional paralelas a las

caras del volumen de control. En los programas para coordenadas cartesianas se ha usado un método conocido como diferencias híbridas (Patankar 1980), que consiste en usar diferencias centradas cuando el número de Peclet de la malla (ver apéndice 3, ecuación A.3.5) sea menor que dos, y diferencias aguas arriba (*upwind*) cuando el número de Peclet de la malla sea mayor que dos (Spalding 1972), para los valores de Φ en las caras del volumen de control que aparecen en el término convectivo de la ecuación (2.45). Además de adoptar esta misma estrategia, se emplean diferencias centradas para las derivadas tanto normales como cruzadas que aparecen en el término difusivo de (2.45). Para un mayor detalle del método híbrido de diferencias para el tratamiento de los términos convectivos y difusivos el lector es referido al apéndice 3.

2.2.3 Coeficientes de matriz.

El uso de diferencias centradas para el cálculo de las derivadas cruzadas, descritas en la sección anterior, extiende la molécula de diferencias de una estrella de 7-puntos a una estrella de 19-puntos en tres dimensiones.

Al sustituir la expresión (2.45) en la ecuación (2.40) y reemplazando a Φ y a sus derivadas según la sección anterior, se obtiene la ecuación algebraica:

$$\left(\sum_{nn} a_{nn} + s_m \right) \Phi_P - \sum_{nn} a_{nn} \Phi_{NN} - S' = \langle J|S \rangle_P, \quad (2.48)$$

como en la sección 2.1.7, los subíndices con mayúsculas significan los valores en los centroides de los volúmenes de control vecinos, y las minúsculas se refieren a las caras del volumen centrado en P . Así mismo, a_{nn} son los coeficientes de matriz estándar obtenidos bajo la estrategia de diferencias híbridas como se explica en el apéndice 3, y s_m es un término fuente de masa. Por lo tanto:

$$\begin{aligned} a_e &= \max\left(\frac{1}{2}|C_e|, D_e\right) - \frac{1}{2}C_e, \\ a_w &= \max\left(\frac{1}{2}|C_w|, D_w\right) + \frac{1}{2}C_w, \\ a_n &= \max\left(\frac{1}{2}|C_n|, D_n\right) - \frac{1}{2}C_n, \\ a_s &= \max\left(\frac{1}{2}|C_s|, D_s\right) + \frac{1}{2}C_s, \\ a_t &= \max\left(\frac{1}{2}|C_t|, D_t\right) - \frac{1}{2}C_t, \\ a_b &= \max\left(\frac{1}{2}|C_b|, D_b\right) + \frac{1}{2}C_b, \\ s_m &= C_e - C_w + C_n - C_s + C_t - C_b, \end{aligned} \quad (2.49)$$

S' es el término extra que sale de las derivadas cruzadas debido a la no ortogonalidad de la malla:

$$S' = \left[D^{12} \frac{\partial \Phi}{\partial \xi_2} + D^{13} \frac{\partial \Phi}{\partial \xi_3} \right]'_w + \left[D^{21} \frac{\partial \Phi}{\partial \xi_1} + D^{23} \frac{\partial \Phi}{\partial \xi_3} \right]'_s + \left[D^{31} \frac{\partial \Phi}{\partial \xi_1} + D^{32} \frac{\partial \Phi}{\partial \xi_2} \right]'_b. \quad (2.50)$$

Desarrollando las derivadas y evaluando en las caras correspondientes se obtiene:

$$\begin{aligned} S' = & D_e^{12} \left[\frac{1}{4} (\Phi_N - \Phi_S + \Phi_{NE} - \Phi_{SE}) \right] + D_e^{13} \left[\frac{1}{4} (\Phi_T - \Phi_B + \Phi_{TE} - \Phi_{BE}) \right] \\ & - D_w^{12} \left[\frac{1}{4} (\Phi_N - \Phi_S + \Phi_{NW} - \Phi_{SW}) \right] - D_w^{13} \left[\frac{1}{4} (\Phi_T - \Phi_B + \Phi_{TW} - \Phi_{BW}) \right] \\ & + D_n^{21} \left[\frac{1}{4} (\Phi_E - \Phi_W + \Phi_{EN} - \Phi_{WN}) \right] + D_n^{23} \left[\frac{1}{4} (\Phi_T - \Phi_B + \Phi_{TN} - \Phi_{BN}) \right] \\ & - D_s^{21} \left[\frac{1}{4} (\Phi_E - \Phi_W + \Phi_{ES} - \Phi_{WS}) \right] - D_s^{23} \left[\frac{1}{4} (\Phi_T - \Phi_B + \Phi_{TS} - \Phi_{BS}) \right] \\ & + D_t^{31} \left[\frac{1}{4} (\Phi_E - \Phi_W + \Phi_{ET} - \Phi_{WT}) \right] + D_t^{32} \left[\frac{1}{4} (\Phi_N - \Phi_S + \Phi_{NT} - \Phi_{ST}) \right] \\ & - D_b^{31} \left[\frac{1}{4} (\Phi_E - \Phi_W + \Phi_{EB} - \Phi_{WB}) \right] - D_b^{32} \left[\frac{1}{4} (\Phi_N - \Phi_S + \Phi_{NB} - \Phi_{SB}) \right], \end{aligned}$$

$$\begin{aligned} \Rightarrow 4S' = & (\Phi_E - \Phi_W) (D_n^{21} - D_s^{21} + D_t^{31} - D_b^{31}) \\ & + (\Phi_N - \Phi_S) (D_e^{12} - D_w^{12} + D_t^{32} - D_b^{32}) \\ & + (\Phi_T - \Phi_B) (D_e^{13} - D_w^{13} + D_n^{23} - D_s^{23}) \\ & + \Phi_{NE} (D_n^{21} + D_e^{12}) - \Phi_{SE} (D_e^{12} + D_s^{21}) \\ & + \Phi_{TE} (D_e^{13} + D_t^{31}) - \Phi_{BE} (D_e^{13} + D_b^{31}) \\ & + \Phi_{SW} (D_w^{12} + D_s^{21}) - \Phi_{NW} (D_w^{12} + D_n^{21}) \\ & + \Phi_{BW} (D_w^{13} + D_b^{31}) - \Phi_{TW} (D_w^{13} + D_t^{31}) \\ & + \Phi_{TN} (D_n^{23} + D_t^{32}) - \Phi_{TS} (D_s^{23} + D_t^{32}) \\ & + \Phi_{BS} (D_s^{23} + D_b^{32}) - \Phi_{BN} (D_n^{23} + D_b^{32}), \end{aligned} \quad (2.51)$$

insertando la expresión (2.51) en la ecuación (2.48), se obtienen las ecuaciones lineales requeridas para Φ sobre la molécula de 19-puntos. Sin embargo, se ha preferido usar la aproximación de corrección diferida (*deferred correction*), donde los términos en S' se incluyen

como término fuente en el lado derecho de la ecuación (2.48). Linealizando el término fuente en la forma usual:

$$\langle J|S \rangle_p = vol/\Omega_p \langle S \rangle_p = s_\phi + s_0 \Phi_p,$$

se obtienen las ecuaciones lineales:

$$a_p \Phi_p - \sum_{nn} a_{nn} \Phi_{NN} = s_\phi + S', \quad (2.52)$$

donde

$$a_p = \sum_{nn} a_{nn} - s_p, \quad s_p = s_0 - s_m.$$

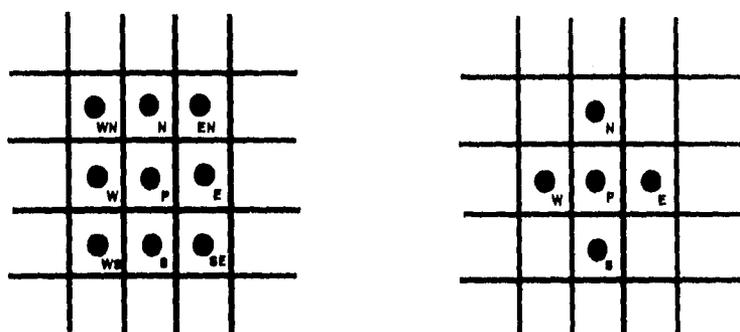


Fig. 2.9. Reducción de la molécula numérica usando corrección diferida en 2D.

Entonces, se ha reducido nuevamente la molécula de 19-puntos a 7-puntos. Además, el uso de diferencias híbridas para el cálculo de los coeficientes de matriz de esta molécula de 7-puntos garantiza recuperar las propiedades de la matriz usada en las versiones cartesianas, y por lo tanto se puede hacer uso del mismo inversor de matriz incluido en esas versiones. Este inversor funciona de manera iterativa bajo el algoritmo de Thomas para matrices tridiagonales (Fletcher 1991).

2.3 Algoritmo de acoplamiento velocidad-presión.

2.3.1 Ecuaciones de cantidad de movimiento.

La forma cartesiana de las ecuaciones de Navier-Stokes en estado permanente y para fluidos incompresibles está dada por la ecuación (1.15) que se escribe en forma tensorial como sigue:

$$\frac{\partial}{\partial x^i} (\rho U^i U^k) = \frac{\partial}{\partial x^i} (\sigma^{ik}),$$

$$\sigma^{ik} = -p \delta_k^i + \mu \left(\frac{\partial U^i}{\partial x^k} + \frac{\partial U^k}{\partial x^i} \right),$$

donde σ^{ik} es el tensor de esfuerzos, p es la presión dinámica y μ es el coeficiente de viscosidad dinámica (no necesariamente constante).

Se puede llevar a estas ecuaciones a la forma general de la ecuación de convección-difusión de la siguiente manera:

$$\begin{aligned} \frac{\partial}{\partial x^i} \left(\rho U^i U^k - \mu \frac{\partial U^k}{\partial x^i} \right) &= -\frac{\partial p}{\partial x^k} + \frac{\partial}{\partial x^i} \left(\mu \frac{\partial U^i}{\partial x^k} \right), \\ \Rightarrow \frac{\partial}{\partial x^i} I_{U^k}^i &= S_{U^k}, \end{aligned} \quad (2.53)$$

donde los vectores del flujo total y los términos fuente están dados por:

$$I_{U^k}^i = \rho U^i U^k - \mu \frac{\partial U^k}{\partial x^i}, \quad (2.54)$$

$$S_{U^k} = -\frac{\partial p}{\partial x^k} + \frac{\partial}{\partial x^i} \left(\mu \frac{\partial U^i}{\partial x^k} \right). \quad (2.55)$$

La ecuación (2.53) está en la forma general de la ecuación de convección-difusión (2.29). Así que, dejando las componentes de la velocidad U^i fijas en las direcciones cartesianas, se transforma al espacio computacional de la siguiente manera:

$$\frac{\partial}{\partial \xi^i} \hat{I}_{U^k}^i = \sqrt{g} S_{U^k}, \quad (2.56)$$

donde

$$\hat{I}_{U^k}^i = \rho \hat{U}^i U^k - \Gamma^{ij} \frac{\partial U^k}{\partial \xi^j}, \quad \Gamma^{ij} = \mu G^{ij}. \quad (2.57)$$

Siguiendo el procedimiento explicado en la sección 2.2.2, se llega a las ecuaciones discretas:

$$\begin{aligned} \left[\hat{I}_{U^k}^1 \right]_w^e + \left[\hat{I}_{U^k}^2 \right]_s^n + \left[\hat{I}_{U^k}^3 \right]_b^t &= \langle \sqrt{g} S_{U^k} \rangle, \\ \Rightarrow a_p U_p^k - \sum_{nn} a_{nn} U_{NN}^k &= vol \Omega_p \langle S_{U^k} \rangle + S'_{U^k}, \end{aligned} \quad (2.58)$$

donde los coeficientes de matriz están dados por las ecuaciones (2.49) usando a la viscosidad como difusividad escalar y el término fuente, debido a la no ortogonalidad, S'_{U^k} está dado por la ecuación (2.51), usando a la componente cartésiana U^k en vez de Φ . De aquí se sigue que los coeficientes de difusión y de convección, y por lo tanto los coeficientes de matriz, son idénticos para las tres componentes cartesianas de la velocidad.

Para realizar el cálculo del término fuente entre corchetes en la ecuación (2.58), se integra en el espacio físico en un volumen de control Ω con centroide P :

$$\int_{\Omega} S_{U^k} dV = -\int_{\Omega} \frac{\partial p}{\partial x^k} dV + \int_{\Omega} \frac{\partial}{\partial x^i} \left(\mu \frac{\partial U^i}{\partial x^k} \right) dV, \quad (2.59)$$

$$\begin{aligned}
\Rightarrow \text{vol}\Omega \langle S_{U^k} \rangle &\approx -\text{vol}\Omega \left(\frac{\partial p}{\partial x^k} \right)_p + \int_{\partial\Omega} \left(\mu \frac{\partial \bar{U}}{\partial x^k} \right) \cdot d\bar{A}, \\
&= -\text{vol}\Omega \left(\frac{A_k^j}{|J|} \frac{\partial p}{\partial \xi^j} \right)_p + \sum_{nn} \left(\mu \bar{A} \cdot \frac{\partial \bar{U}}{\partial x^k} \right)_{nn}, \\
&= - \left(A_k^j \frac{\partial p}{\partial \xi^j} \right)_p + \sum_{nn} \left(\mu \bar{A} \cdot \frac{A_k^j}{|J|} \frac{\partial \bar{U}}{\partial \xi^j} \right)_{nn}, \\
&= - \left(A_k^j \frac{\partial p}{\partial \xi^j} \right)_p + \sum_{nn} \left(\mu \bar{A} \cdot \frac{A_k^j}{\text{vol}\Omega} \frac{\partial \bar{U}}{\partial \xi^j} \right)_{nn}.
\end{aligned} \tag{2.60}$$

Por lo tanto, en coordenadas generalizadas, cada ecuación de balance de cantidad de movimiento contiene las tres derivadas de la presión en el espacio computacional. Estas se calculan usando diferencias centradas en todos los volúmenes de control, excepto en aquellos adyacentes a la frontera del flujo, donde es necesario retrasar o adelantar la molécula de la derivada puesto que la presión no está definida fuera del dominio de flujo.

2.3.2 Ecuación de corrección de presión.

Para resolver simultáneamente las ecuaciones de balance de cantidad de movimiento y de masa, se construye una ecuación aproximada para la presión conocida como ecuación de corrección de presión. En esta sección se discute la ecuación de corrección de presión que se obtiene al aplicar el algoritmo SIMPLE (Patankar y Spalding 1972, Patankar 1980) a las ecuaciones de cantidad de movimiento (2.58), para las componentes cartesianas de la velocidad en una malla no defasada en variables (*non-staggered*).

Sean U^{i*} y p^* la velocidad y la presión obtenidas después de resolver las ecuaciones linealizadas de cantidad de movimiento (2.58). De la discusión previa, se pueden escribir las ecuaciones de cantidad de movimiento de la siguiente manera:

$$U_p^{i*} = \sum_{nn} \frac{a_{nn} U_{NN}^{i*}}{a_p} + S'_{U^i} - \frac{A_i^k}{a_p} \frac{\partial p^*}{\partial \xi^k}, \tag{2.61}$$

donde S'_{U^i} denota a los términos fuente restantes después de quitar los de gradiente de presión y dividirlo por el coeficiente a_p .

Pero la solución U^{i*} de la ecuación (2.61) no necesariamente satisface la ecuación de continuidad; tiene una masa residual:

$$\begin{aligned}
m_p &= [\rho \hat{U}^{1*}]_w + [\rho \hat{U}^{2*}]_s + [\rho \hat{U}^{3*}]_b, \\
&= C_e^* - C_w^* + C_n^* - C_s^* + C_t^* - C_b^*.
\end{aligned} \tag{2.62}$$

Los términos en la ecuación (2.62) incluyen los valores de las componentes de la velocidad normales a las caras del volumen de control, y estas deben ser aproximadas a partir de los valores de la velocidad en los centroides de cada volumen de control. El algoritmo de Rhie-Chow, discutido en la sección siguiente, es el responsable de hacer esto de una manera adecuada. Por ahora se suponen hechas tanto las interpolaciones necesarias, como el cómputo de los términos fuente de masa.

La idea principal del algoritmo SIMPLE es encontrar nuevas velocidades y presiones U^{i**} y p^{**} que satisfagan las ecuaciones discretas de balance de cantidad de movimiento y masa:

$$U_p^{i**} = \sum_{nn} \frac{a_{nn} U_{nn}^{i**}}{a_p} + S'_{U^i} - \frac{A_i^k}{a_p} \frac{\partial p^{**}}{\partial \xi^k}, \quad (2.63)$$

$$[\rho \hat{U}^{1**}]_w^c + [\rho \hat{U}^{2**}]_s^n + [\rho \hat{U}^{3**}]_b^t = 0. \quad (2.64)$$

Para lograrlo, se propone que los valores nuevos (de doble asterisco) sean la suma del valor que cumple la ecuación de cantidad de movimiento (de un solo asterisco) y una corrección, así pues:

$$U^{i**} = U^{i*} + \tilde{u}^i, \quad p^{**} = p^* + p'. \quad (2.65)$$

Además, si se supone que las velocidades en los nodos vecinos son aproximables por los valores no corregidos:

$$U_{NN}^{i**} \approx U_{NN}^{i*}, \quad \Rightarrow \tilde{u}^i = -\frac{A_i^k}{a_p} \frac{\partial p'}{\partial \xi^k}.$$

Entonces, las correcciones para la velocidad y para las componentes de flujo de velocidad normal están dadas por:

$$U^{i**} = U^{i*} - \frac{A_i^k}{a_p} \frac{\partial p'}{\partial \xi^k}, \quad (2.66)$$

$$\hat{U}^i = A_j^i U^j, \quad \Rightarrow \hat{U}^{i**} = \hat{U}^{i*} - \frac{A_j^i A_j^k}{a_p} \frac{\partial p'}{\partial \xi^k}. \quad (2.67)$$

Rhie y Chow señalan que para mallas casi ortogonales, las componentes diagonales del producto entre áreas en la expresión (2.67) son muy pequeñas, y en cualquier caso, las correcciones se anulan para una solución en convergencia. Se recomienda despreciar los términos fuera de la diagonal en la expresión (2.67) para obtener:

$$\hat{U}^{i**} = \hat{U}^{i*} - \frac{A_j^i A_j^i}{a_p} \frac{\partial p'}{\partial \xi^i}. \quad (2.68)$$

Para obtener la ecuación de corrección de presión se sustituye la ecuación aproximada (2.68) en la ecuación de conservación de masa (2.64) y se usan diferencias centradas para aproximar los gradientes de corrección de presión. De esta manera se obtienen las ecuaciones discretas:

$$b_p p'_p = \sum_{nn} b_{nn} p'_{NN} - m_p, \quad b_p = \sum_{nn} b_{nn}, \quad (2.69)$$

donde

$$b_{e,w} = \left(\rho \frac{A_j^1 A_j^1}{a_p} \right)_{e,w}, \quad b_{n,s} = \left(\rho \frac{A_j^2 A_j^2}{a_p} \right)_{n,s}, \quad b_{t,b} = \left(\rho \frac{A_j^3 A_j^3}{a_p} \right)_{t,b}. \quad (2.70)$$

Por lo tanto, como en el caso cartesiano, se obtiene una ecuación lineal simétrica en una molécula de 7-puntos que puede ser resuelta con el mismo inversor de matriz usado para la ecuación de cantidad de movimiento.

Al comparar las expresiones (2.70) con la fórmula (2.68), se observa que los coeficientes en la ecuación de corrección de presión son proporcionales al factor de los gradientes de presión en la corrección de la velocidad. Se debe notar que, para la línea fronteriza del dominio de estudio, no puede haber corrección a la velocidad externa (fija como condición de frontera), y por lo tanto los coeficientes que acoplan nodos internos con nodos externos deben anularse. De hecho, así debe de ser, pues la presión no está definida en nodos externos.

Es interesante recuperar la forma de la ecuación de corrección de presión sin recurrir a la aproximación de Rhie-Chow que elimina los términos fuera de la diagonal en el producto entre áreas en la ecuación (2.67). Al retenerlos, las correcciones a las componentes cartesianas y normales de la velocidad están dadas por las expresiones (2.66) y (2.67) respectivamente, y la ecuación de corrección de presión se obtiene al sustituir la expresión (2.67) en la ecuación de continuidad (2.64). Así es como se obtiene la ecuación discreta:

$$b_p p'_p = \sum_{nn} b_{nn} p'_{NN} + S' - m_p, \quad (2.71)$$

que contiene como único término extra a S' que incluye a los términos nuevos debido a la no ortogonalidad de la malla. Estos están dados por:

$$S' = \left[E^{12} \frac{\partial p'}{\partial \xi_2} + E^{13} \frac{\partial p'}{\partial \xi_3} \right]_w^e + \left[E^{21} \frac{\partial p'}{\partial \xi_1} + E^{23} \frac{\partial p'}{\partial \xi_3} \right]_s^n + \left[E^{31} \frac{\partial p'}{\partial \xi_1} + E^{32} \frac{\partial p'}{\partial \xi_2} \right]_b^t, \quad (2.72)$$

donde

$$E_{nn}^{ij} = \left(\rho \frac{A_k^i A_k^j}{a_p} \right)_{nn} = \left(\rho \frac{\sqrt{g}}{a_p} G^{ij} \right)_{nn}, \quad (2.73)$$

y que contienen a las cantidades previamente definidas en las ecuaciones (2.21) y (2.38).

El desarrollo de la expresión para S' se hace de manera análoga a la fórmula (2.51). Los términos extras en S' extienden la ecuación discretizada a una molécula de 19-puntos. Este problema se resuelve nuevamente haciendo una corrección diferida. Sin embargo debe tomarse en cuenta que si no la hacemos antes de cada llamada al inversor de matriz para p' , estamos repitiendo la aproximación de Rhie-Chow pues comenzaremos con un campo de presiones inicializado a cero. El ganar precisión frente a la aproximación de Rhie-Chow asegura una mejor conservación de masa.

2.3.3 Método de interpolación de Rhie-Chow.

Finalmente, se debe contestar a la pregunta de como obtener las componentes de la velocidad en las caras del volumen de control partiendo de los valores de la velocidad en los centroides de los volúmenes de control, previo al cálculo de los coeficientes de convección y del término fuente de masa en la ecuación (2.62). La manera intuitiva de hacerlo es utilizando una interpolación lineal en el espacio físico como se explica en la sección 2.1.7:

$$U_e^i = \underline{U}_e^i = (1 - W_e)U_P^i + W_e U_E^i. \quad (2.74)$$

Sin embargo, se sabe que esto implica desacoplar la malla al estilo de un tablero de ajedrez, es decir, el algoritmo se vuelve insensible a variaciones de la presión entre nodos vecinos (Patankar 1980). Para lograr una mejor interpolación, se recurre a la ecuación discreta de cantidad de movimiento para las componentes cartesianas de la velocidad (2.61). Esta se escribe en forma corta como:

$$U + B\nabla p = AU + S. \quad (2.75)$$

Usando la ecuación (2.75) en un centroide P y en su vecino al este E , las componentes de la velocidad cumplen la ecuación discreta de balance de cantidad de movimiento:

$$U_P + (B\nabla p)_P = (AU)_P + S_P, \quad (2.76)$$

$$U_E + (B\nabla p)_E = (AU)_E + S_E. \quad (2.77)$$

La forma tradicional de acabar con las oscilaciones en la presión es usar las componentes de la velocidad U_e , etc., obtenidas al discretizar las ecuaciones de momento en una malla defasada (*staggered*) cuyos centros son las caras de los volúmenes de control originales. En ese caso, U_e satisface la ecuación discreta de cantidad de movimiento de la forma:

$$U_e + (B\nabla p)_e = (AU)_e + S_e, \quad (2.78)$$

donde los gradientes de presión se calculan usando diferencias entre vecinos próximos (E y P en este caso). La idea de Rhie y Chow (Rhie 1981, Rhie y Chow 1983) es simplemente aproximar las soluciones de (2.78) de las soluciones de (2.76) y (2.77). Si se supone que los términos en el lado derecho de (2.78) pueden ser aproximados por interpolaciones lineales de los términos correspondientes en las ecuaciones (2.76) y (2.77), entonces se obtiene:

$$U_e + (B\nabla p)_e = \underline{(AU)_e} + \underline{S_e} = \underline{U_e} + \underline{(B\nabla p)_e},$$

$$\Rightarrow U_e = \underline{U_e} + \underline{(B\nabla p)_e} - (B\nabla p)_e.$$

Dado que $B = \frac{A_i^k}{a_p}$, es factible suponer que $B_e \approx \underline{B_e}$ y $\underline{(B\nabla p)_e} \approx \underline{B_e} \nabla p_e$ si las áreas no cambian bruscamente de un elemento a otro, es decir, si la malla es suave. Con esta suposición se obtiene la fórmula de interpolación de Rhie-Chow:

$$U_e = \underline{U_e} + \underline{B_e}(\underline{\nabla p_e} - (\nabla p)_e). \quad (2.79)$$

Escrito en componentes, esto es:

$$\begin{aligned}
 U_e^i &= \underline{U}_e^i + B_{i e}^k \left(\left(\frac{\partial p}{\partial \xi^k} \right)_e - \left(\frac{\partial p}{\partial \xi^k} \right)_c \right), \\
 \hat{U}^i &= A_j^i U^j, \Rightarrow \\
 \hat{U}_e^i &= \underline{\hat{U}}_e^i + \left(\frac{A_k^i A_k^j}{a_p} \right)_e \left(\left(\frac{\partial p}{\partial \xi^j} \right)_e - \left(\frac{\partial p}{\partial \xi^j} \right)_c \right). \quad (2.80)
 \end{aligned}$$

Dado que los gradientes de presión se calculan usando diferencias centradas, las derivadas cruzadas coinciden con sus interpoladas, cancelándose. Por lo tanto:

$$\hat{U}_e^i = \underline{\hat{U}}_e^i + \left(\frac{A_k^i A_k^i}{a_p} \right)_e \left(\left(\frac{\partial p}{\partial \xi^i} \right)_e - \left(\frac{\partial p}{\partial \xi^i} \right)_c \right). \quad (2.81)$$

(sin suma sobre i)

las derivadas se toman sólo en la dirección normal a la cara donde se calcula el flujo. Dado que el coeficiente factor de los gradientes de presión contiene la norma al cuadrado del área en la cara correspondiente, esta no se interpola, pero el coeficiente a_p se interpola linealmente en el espacio físico. La fórmula (2.81) es análoga para cualquiera de las otras caras w, n, s, t, b .

Aunque el método de interpolación de Rhie-Chow ha mostrado buenas propiedades de estabilidad y convergencia, tiene una propiedad desafortunada. Cuando se emplea un factor de atenuación (*underrelaxation*) para facilitar la convergencia, a_p es cambiado por a_p / α . Esto lleva a un método de interpolación dependiente de α . Sin embargo, si α se ignora en la fórmula (2.81), aparece una solución independiente de α (Melaen 1992).

3. Codificación.

En este capítulo se exponen los puntos principales de la codificación del método de volumen finito desarrollado teóricamente en el capítulo 2. Este código, llamado NECRONOM.f (apéndice 4), resuelve numéricamente las ecuaciones de Navier-Stokes (1.12) para las componentes cartesianas de la velocidad (sección 2.3.1), junto con la ecuación de conservación de masa (1.5), escrita en forma de corrección de presión (sección 2.3.2), dentro de dominios en coordenadas generalizadas.

3.1 Ecuaciones de movimiento.

Las ecuaciones que describen el movimiento de fluidos newtonianos e incompresibles son:

$$\frac{\partial U^i}{\partial x^i} = 0 \quad (3.1)$$

$$\frac{\partial}{\partial t}(\rho U^k) + \frac{\partial}{\partial x^i}(\rho U^i U^k) = \frac{\partial}{\partial x^i} \left(-p \delta_k^i + \mu \left(\frac{\partial U^i}{\partial x^k} + \frac{\partial U^k}{\partial x^i} \right) \right) \quad (3.2)$$

siguiendo la notación de la sección 2.3.1. Estas ecuaciones expresan respectivamente la conservación de la masa y de la cantidad de movimiento en cada elemento infinitesimal de fluido. En el capítulo 2 se estudia la forma de discretizar estas ecuaciones para asegurar las mismas leyes de conservación en volúmenes de control finitos, de ahí se obtienen una serie de fórmulas discretas que conforman el conjunto de operaciones codificadas en el programa NECRONOM.f.

Además de las ecuaciones (3.1) y (3.2), es necesario especificar el tipo de condiciones iniciales y de frontera para cada problema en particular. En el apéndice 2 se hace una descripción breve del tipo de condiciones de frontera utilizadas dentro de NECRONOM.f.

3.2 Geometría.

El dominio de estudio debe ser entregado por separado al programa NECRONOM.f en el archivo MALLA3D.DAT, este debe tener el formato Enight (CEI 1994) y puede ser generado por cualquier generador de mallas de hexahedros no necesariamente regulares. En el capítulo 4 se exponen dos códigos generadores de este tipo de mallas.

3.3 NECRONOM.f : código de volumen finito en coordenadas generalizadas.

El diagrama de flujo del programa NECRONOM.f se muestra en la Fig. 3.1. Dentro del programa principal (sección A.4.1) se asignan valores a los parámetros físicos como: la densidad ρ , la viscosidad μ , los pasos e incrementos de tiempo t y Δt , y el campo inicial de velocidades. También se dan valores a los parámetros numéricos tales como: las iteraciones máximas, el parámetro de convergencia, los parámetros de atenuación o relajamiento para velocidades y presión, los parámetros de impresión, las llaves de llamado a subrutinas, y los parámetros de monitoreo de variables, definidos en el listado de la sección A.4.1. Antes de dar inicio al procedimiento, es necesario limpiar las localidades de memoria, esto se hace en la subrutina INIT (sección A.4.2) que inicializa las variables a cero. Las primeras cantidades que se calculan son aquellas asociadas a la geometría, es decir, el cálculo de vectores de área, de volúmenes, de factores de peso de interpolación lineal y de productos internos entre áreas. Esto se hace en la subrutina GEOMGC (sección A.4.3), cuyo funcionamiento se comenta en las secciones 2.1.5 y 2.1.7. Esta subrutina necesita alimentarse del archivo MALLA3D.DAT en formato Ensignt (CEI 1994) y genera un nuevo archivo MALLAXYZ.OUT que contiene las coordenadas cartesianas de los centroides de los elementos de la malla original. Posteriormente, se inicializan los campos de velocidades en el tiempo dentro de la subrutina INITCOND (sección A.4.4) y se dejan fijas las condiciones de frontera de tipo Dirichlet (según sea el caso), que no vuelven a ser modificadas durante el algoritmo, y que engloban a las condiciones de pared y de entrada para las tres componentes cartesianas de la velocidad (ver apéndice 2). En los ejemplos expuestos en esta tesis la pared de entrada está en la cara oeste del arreglo geométrico y se calcula el flujo de masa en la entrada (FLOWIN, línea 250, sección A.4.1) para su uso posterior en el balance de masa global.

El ciclo en el tiempo inicia con el subciclo SIMPLE (Patankar 1980) comentado en la sección 2.3.2, donde una serie de subrutinas son llamadas para integrar las ecuaciones gobernantes. Las ecuaciones de cantidad de movimiento para cada componente cartesiana de la velocidad son las primeras en ser integradas según la ecuación (2.58). Se llama a la subrutina CALCOEF (sección A.4.5) que calcula los coeficientes de matriz expresados en las ecuaciones (2.49), y luego a la subrutina FUENTE (sección A.4.6) que además de calcular los términos fuente (2.60) y (2.51), llama a la subrutina PROMOD (sección A.4.7), donde se prevén modificaciones especiales (en caso de ser necesarias) tales como el bloqueo de celdas y las condiciones de frontera de Neuman. La subrutina FUENTE llama al inversor de matriz LISOLV (sección A.4.8), que es el mismo que se usa en las versiones cartesianas, para la solución de cada conjunto de ecuaciones algebraicas asociadas a cada componente cartesiana de la velocidad. También dentro de la subrutina FUENTE (sección A.4.6), se evalúan los residuos asociados a las ecuaciones de conservación de cantidad de movimiento (RESORU, línea 275, sección A.4.6) para ser comparados posteriormente con el parámetro de convergencia (SORMAX, línea 152, sección A.4.1).

De la subrutina FUENTE (sección A.4.6) se obtiene el campo de velocidades que es interpolado a los centroides de las caras de los volúmenes de control bajo la fórmula de

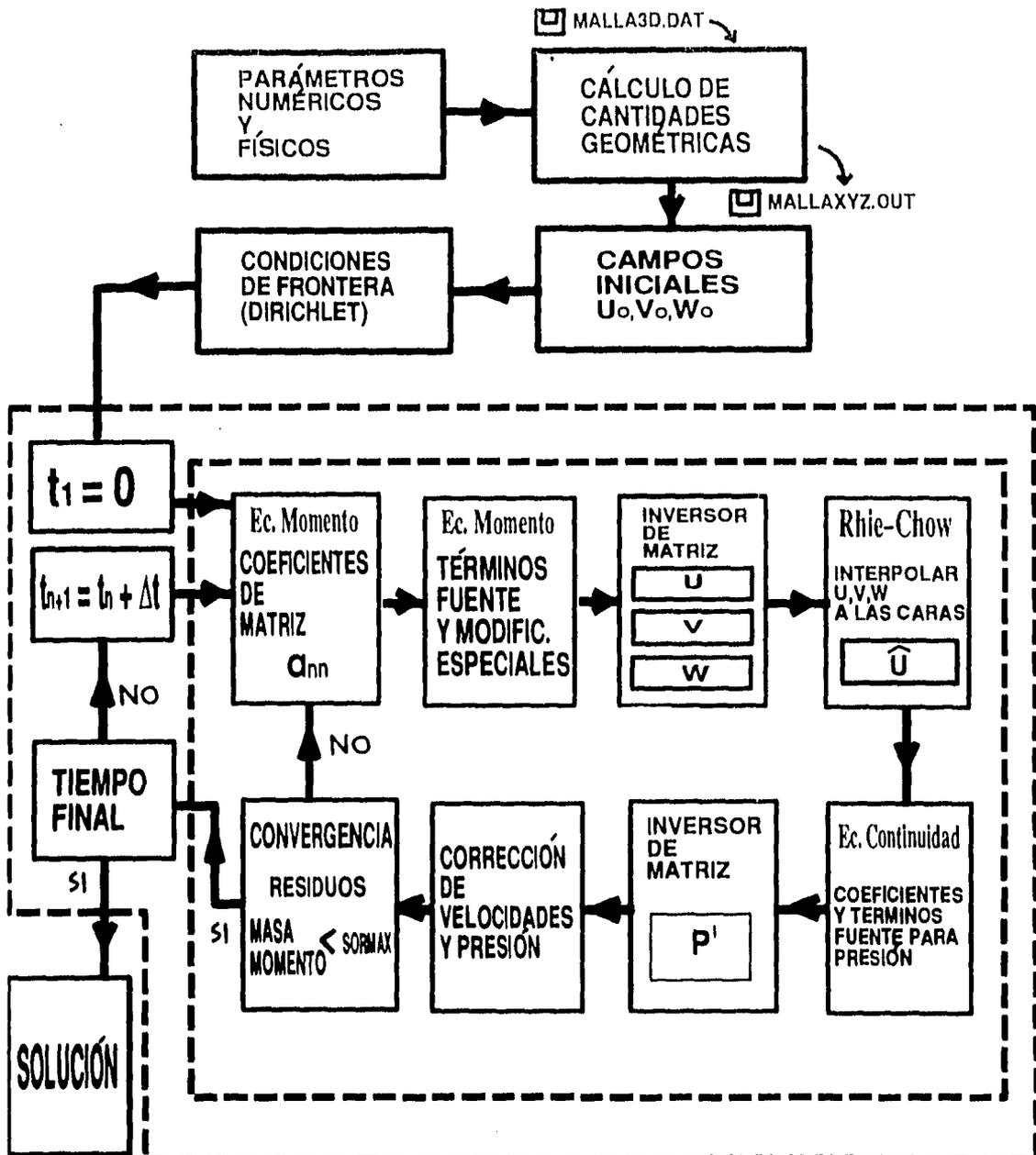


Fig 3.1. Diagrama de flujo del programa NECRONOM.f

Rhie-Chow (2.81) y esto se hace en una serie de subrutinas asociadas con cada una de las coordenadas generalizadas: RHIECHOWW (asociado a ξ , cara oeste), RHIECHOWS (asociado a η , cara sur), y RHIECHOWB (asociado a ζ , cara abajo) (sección A.4.9). Ha sido necesario especializar una serie de subrutinas pequeñas para el cálculo de la velocidad en las caras frontera del dominio, estas subrutinas toman el valor de la velocidad en la frontera y no lo interpolan con el fin de dejar la condición de frontera fija en las caras

frontera del dominio. Estas subrutinas se llaman RHICHWW, RHICHWS y RHICHWB (sección A.4.10).

El siguiente paso es la integración de la ecuación de conservación de masa que, como ha sido explicado en la sección 2.3.2, se cambia por una ecuación de corrección para la presión. Los coeficientes de matriz (2.70), los términos fuente (2.62) y (2.72), el residuo de la ecuación de conservación de masa (RESORM, línea 135, sección A.4.11), las modificaciones especiales y la inversión de matriz para la corrección de presión (subrutina LISOLV, sección A.4.8), son calculados dentro de la subrutina CALP (sección A.4.11). Las modificaciones especiales que se hacen para la corrección de presión se llevan a cabo en la subrutina PROMOD (sección A.4.7) y consisten en bloquear los coeficientes de matriz en las fronteras, como se explica en la sección 2.3.2, asegurar la condición de salida en la presión $\left. \frac{\partial p}{\partial \xi} \right|_{SALIDA} = 0$ para flujo desarrollado, y extrapolar los valores de la presión y de su corrección para asegurar el funcionamiento de las fórmulas que necesitan de los gradientes de presión cerca de las fronteras.

En la parte final de la subrutina CALP (sección A.4.11) se corrigen las componentes cartesianas de la velocidad, las velocidades normales a las caras de los volúmenes de control, y la presión, según las ecuaciones (2.66), (2.67) y (2.65), tomando los valores de la corrección de presión p' obtenidos del inversor de matriz.

De regreso al programa principal NECRONOM.f, se hace la comparación de los residuos de masa (RESORM) y de cantidad de movimiento (RESORU) con el parámetro de convergencia (SORMAX), y si alguno es mayor que este último, se repite el ciclo SIMPLE y sólo se reportan valores de monitoreo en el archivo de salida channel.out. Pero si se alcanza la convergencia requerida, se llama a la subrutina MPGSOUT (sección A.4.12) que llena los archivos de salida (velo****.vec) para los campos de velocidades y (PRES****.scl) para los campos de presión con el formato Enight (CEI 1994), donde los asteriscos significan el paso en el tiempo actual (p. ej: t=1, ****=0001). Posteriormente, se aumenta el paso en el tiempo por Δt , se sustituyen variables viejas por nuevas en la subrutina SUSTITU (sección A.4.13), y se regresa al subciclo SIMPLE hasta llegar nuevamente a la convergencia.

Cuando se ha alcanzado el último paso de tiempo con la convergencia requerida se ha obtenido una solución.

En la tabla A.4.1 del apéndice 4, se describen los parámetros utilizados a lo largo del programa NECRONOM.f (sección A.4.1) y de las subrutinas INIT (sección A.4.2), GEOMGC (sección A.4.3), INITCOND (sección A.4.4), CALCOEF (sección A.4.5), FUENTE (sección A.4.6), PROMOD (sección A.4.7), LISOLV (sección A.4.8), RHIECHOW (sección A.4.9), RHICHW (sección A.4.10), CALP (sección A.4.11), MPGSOUT (sección A.4.12), SUSTITU (sección A.4.13), y PRINT (sección A.4.14).

4. Generación de mallas.

En los capítulos anteriores se ha desarrollado la estrategia teórica y computacional para resolver el flujo laminar e incompresible de un fluido dentro de una geometría descrita en coordenadas generalizadas. Para comenzar el cálculo es necesario proporcionar el dominio de estudio en forma de una malla tridimensional compuesta por hexahedros, no necesariamente regulares. El propósito de este capítulo es explicar y aplicar ciertas técnicas algebraicas que permiten construir, con dos curvas dadas, una malla bidimensional localmente ortogonal en sus fronteras. Con este tipo de mallas se pueden generar dominios tridimensionales con sección constante repitiendo la malla en la dirección z .

4.1 Mapeos algebraicos.

Las técnicas de mapeo algebraico interpolan los datos de las fronteras para generar la malla interior. La interpolación puede obtenerse para una dimensión o para varias dimensiones. Un requerimiento esencial es que la malla sea bien comportada, es decir, que varíe suavemente, casi ortogonal, y que sus elementos tengan razones de aspecto cercanas a la unidad. Es importante construir una malla que sea ortogonal o casi ortogonal cerca de las fronteras pues las soluciones en estas zonas presentan mayor variabilidad y sensibilidad. Los métodos descritos en esta sección tienen esta propiedad además de poder deformar la malla, aumentando el número de nodos en zonas donde una mayor resolución numérica es necesaria.

La distribución de puntos en las fronteras es gobernado por las funciones de deformación (*stretching functions*) (Roberts 1971), mientras que la distribución interior es calculada con la técnica de multisuperficies (Fletcher 1991), que permite asegurar la ortogonalidad local en las fronteras.

4.1.1 Funciones de deformación unidimensionales.

Este tipo de funciones son usadas para reordenar la distribución de puntos a lo largo de una frontera particular. Esto permite aumentar el número de nodos en regiones específicas del dominio. Para flujos viscosos es apropiado introducir este tipo de funciones para poder aumentar el número de puntos perpendiculares a las fronteras sólidas y así poder resolver los gradientes esperados en esas zonas.

Es deseable expresar a las variables en forma normalizada dentro de las funciones de deformación. Sea una curva delimitada por los puntos A y B. Una normalización apropiada para la coordenada curvilínea η sobre esa curva es:

$$\eta^* = \frac{\eta - \eta_A}{\eta_B - \eta_A}, \quad \Rightarrow \quad \begin{aligned} 0 \leq \eta^* \leq 1 \\ \eta_A \leq \eta \leq \eta_B \end{aligned} \quad (4.1)$$

Una función de deformación efectiva (Roberts 1971, Eiseman 1979) es:

$$s = P\eta^* + (1 - P) \left(1 - \frac{\tanh(Q(1 - \eta^*))}{\tanh Q} \right), \quad (4.2)$$

donde P y Q son parámetros que controlan la distribución de puntos, ver Fig. 4.1. P es el responsable de la pendiente inicial de la función ($s = P\eta^*$ cerca de $\eta^* = 0$) cuando Q no es pequeña, ver Fig. 4.1(a). Q es llamada factor de amortiguamiento y controla la separación del comportamiento lineal de s vs. η^* , ver Fig. 4.1(b). Pequeños valores de Q causan pequeñas diferencias con el comportamiento lineal. Sin embargo, si P está cerca de la unidad, la diferencia con el comportamiento lineal es pequeña.

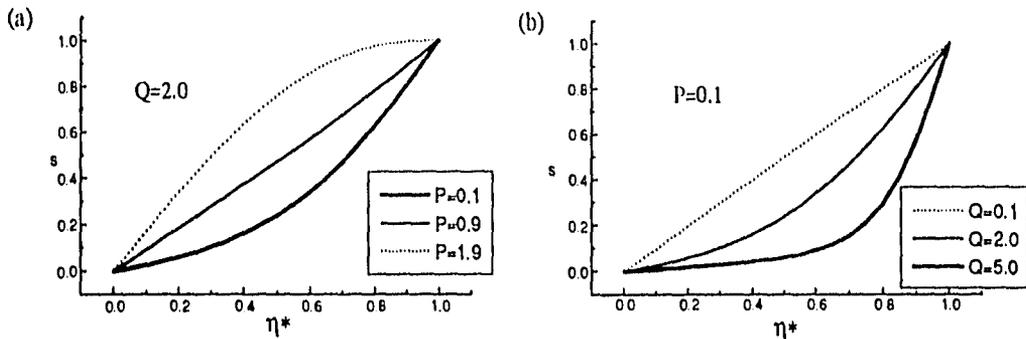


Fig. 4.1. Gráficas de la función de deformación 4.2 para distintos valores de Q y P.

Una vez obtenida s, se utiliza para especificar la distribución de x y y. Por ejemplo, definiendo:

$$\frac{x - x_A}{x_B - x_A} = f(s), \quad \frac{y - y_A}{y_B - y_A} = g(s), \quad (4.3)$$

se generan $x(s)$ y $y(s)$ directamente. Un caso sencillo se encuentra definiendo $f(s) = g(s) = s$, de esta forma se tiene:

$$\begin{aligned} x &= x_A + s(x_B - x_A), \\ y &= y_A + s(y_B - y_A). \end{aligned} \quad (4.4)$$

Que en el caso de una frontera recta, hereda la distribución de puntos directamente del comportamiento de s vs. η^* (Fig. 4.1). De aquí es posible escoger los valores convenientes de Q y P para el efecto deseado. En el caso general de fronteras curvas, una vez que s se ha calculado, se interpola la distancia sobre la curva para obtener las coordenadas (x,y) asociadas a cada s.

4.1.2 Método de multisuperficies.

Para construir una malla bidimensional con este método, es necesario especificar dos curvas que constituyan la frontera y que se unan en su entrada y salida por rectas como si formaran un canal, ver Fig. 4.2.

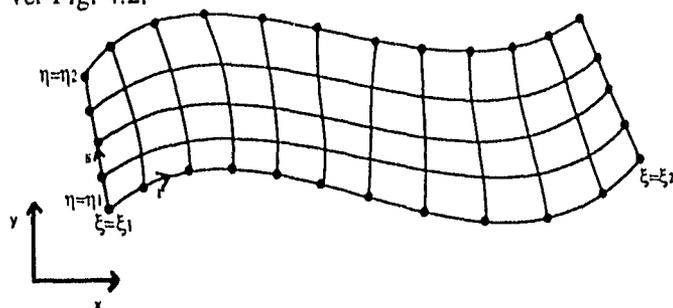


Fig. 4.2. Canal bidimensional generado por dos curvas frontera donde se conoce la distribución de puntos r y s .

Como su nombre lo indica, en esta técnica se introducen otras superficies o curvas intermedias con las que se logra obtener un mayor control sobre la malla interna.

Como paso intermedio para obtener la transformación que permita construir la malla interna, se introducen las coordenadas paramétricas (r,s) (Fig. 4.2) de tal manera que:

$$\begin{aligned} 0 \leq r \leq 1 & \quad \text{para} \quad \xi_1 \leq \xi \leq \xi_2, \\ 0 \leq s \leq 1 & \quad \text{para} \quad \eta_1 \leq \eta \leq \eta_2. \end{aligned}$$

Estas coordenadas contienen la información de la función de deformación (4.2) en cada una de las fronteras y representan a las coordenadas generalizadas normalizadas.

En este método se propone interpolar una serie de vectores que unen a puntos correspondientes (mismos valores de r) en superficies adyacentes, ver Fig. 4.3., esto permite obtener dos ventajas directas. Primero, ajustando la correspondencia puntual entre una superficie frontera y su superficie intermedia vecina, se hace la malla localmente ortogonal a dicha frontera. Segundo, la distribución de puntos en s se obtiene integrando la interpolación de la secuencia de vectores. Esto proporciona una distribución muy suave para s e, incidentalmente, no requiere que la malla interna interpole a las superficies intermedias. En principio no hay límite para el número de superficies intermedias, pero si lo que se busca es una malla localmente ortogonal en las fronteras, basta con dos superficies intermedias.

La distribución de puntos en la i -ésima superficie se incluye en una función vectorial $\vec{Z}_i(r)$ con componentes $x_i(r), y_i(r)$, para el caso de dos dimensiones.

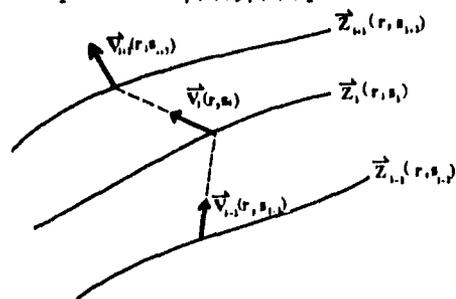


Fig. 4.3. Vectores tangentes \vec{V}_i y superficies intermedias \vec{Z}_i .

Contando a las superficies fronterizas, se tienen N superficies totales, y por lo tanto, $N-2$ superficies intermedias. Definimos los vectores que unen a los puntos en superficies contiguas como:

$$\vec{V}_i(r) = A_i [\vec{Z}_{i+1}(r) - \vec{Z}_i(r)], \quad i = 1, \dots, N-1. \quad (4.5)$$

El parámetro A_i es determinado más tarde y es un factor de normalización de manera que la interpolación interna esté en el intervalo $0 \leq s \leq 1$. Una interpolación sobre la familia de vectores $\vec{V}_i(r)$ tiene como resultado una función vectorial $\vec{V}(r, s)$ continua en r y s .

$$\vec{V}(r, s) = \sum_{i=1}^{N-1} \psi_i(s) \vec{V}_i(r), \quad (4.6)$$

donde $\psi_i(s)$ son funciones de interpolación, a ser escogidas, de tal manera que cumplan $\psi_i(s_k) = 1$ si $i=k$ y cero si $i \neq k$. De cualquier manera, de la forma en la que $\vec{V}_i(r)$ ha sido construido y para que sea vector tangente es claro que:

$$\frac{\partial \vec{Z}(r, s)}{\partial s} = \vec{V}(r, s) = \sum_{i=1}^{N-1} \psi_i(s) \vec{V}_i(r), \quad (4.7)$$

de donde se obtiene $\vec{Z}(r, s)$ que es la función continua que genera la malla para valores dados de r y s (y por lo tanto de ξ y η). $\vec{Z}(r, s)$ se obtiene al integrar (4.7) en el intervalo $0 \leq s \leq 1$. Entonces, con la inclusión de la expresión (4.5) se tiene que:

$$\vec{Z}(r, s) = \vec{Z}_1(r) + \sum_{i=1}^{N-1} A_i G_i(s) [\vec{Z}_{i+1}(r) - \vec{Z}_i(r)], \quad (4.8)$$

donde

$$G_i(s) = \int_0^s \psi_i(s') ds'. \quad (4.9)$$

Los parámetros A_i se escogen de tal manera que $A_i G_i(1) = 1$. Así, la ecuación (4.8) se reduce a $\vec{Z}(r, s) = \vec{Z}_N(r)$ cuando $s=1$, como se requiere. Consecuentemente:

$$\vec{Z}(r, s) = \vec{Z}_1(r) + \sum_{i=1}^{N-1} \frac{G_i(s)}{G_i(1)} [\vec{Z}_{i+1}(r) - \vec{Z}_i(r)], \quad (4.10)$$

que es la forma general de la transformación de multisuperficies.

Las funciones de interpolación $\psi_i(s)$ deben ser continuamente diferenciables un orden mayor que el orden necesario en la malla. Una familia apropiada de funciones de interpolación $\psi_i(s)$ es:

$$\psi_i(s) = \prod_{\substack{l=1 \\ l \neq i}}^{N-1} (s - s_l). \quad (4.11)$$

En la sección siguiente se aplican estas técnicas para obtener un programa generador de mallas bidimensionales capaz de asegurar ortogonalidad local en las fronteras.

4.2 Aplicación numérica del mapeo algebraico.

Las técnicas discutidas en la sección anterior son combinadas para producir un programa de cómputo capaz de generar una malla entre dos curvas frontera.

Para el dominio mostrado en la Fig. 4.4, las superficies frontera consisten de la pared de un difusor, AB, y de la línea recta, CD, que es un eje de simetría. Entre estas dos fronteras se genera la mitad de la malla para un difusor. El resto de la malla se obtiene reflejándola sobre la recta CD.

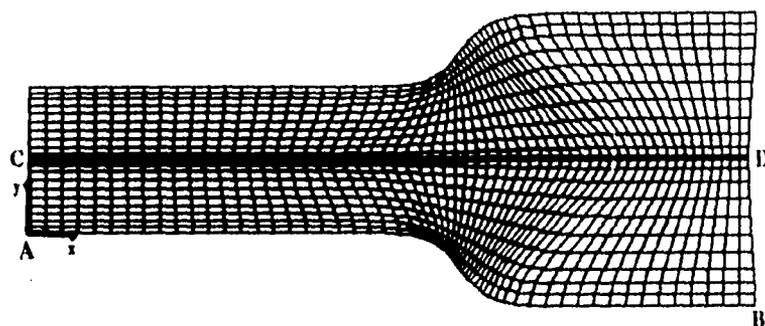


Fig. 4.4. Difusor bidimensional generado con la técnica de multisuperficies.

Como se ha indicado en la primera sección de este capítulo, la generación de la malla consta de dos partes. Primero se determina la distribución de puntos en todas las fronteras, utilizando la función de deformación unidimensional (4.2). Subsecuentemente, se generan los nodos interiores con la técnica de multisuperficies. Dos superficies intermedias, \bar{Z}_2 y \bar{Z}_3 , son introducidas, cada una adyacente a las superficies AB y CD respectivamente. La correspondencia paramétrica (r) de las superficies \bar{Z}_2 y \bar{Z}_3 con sus fronteras vecinas se ajusta de tal forma que las líneas de la malla intersequen ortogonalmente a las fronteras.

4.2.1 MALLAS.f : código de generación de malla para un difusor.

La curva AB de la figura 4.4 representa la pared de un difusor y tiene la siguiente expresión:

$$y = \frac{1}{2}(\tanh(2-10x) - \tanh 2). \quad (4.12)$$

Mientras que la superficie CD es una línea de simetría en $y=1$.

Para relacionar las coordenadas físicas (x, y) de AB con la coordenada computacional ξ es necesario introducir una longitud r_A sobre la curva definida de la siguiente forma:

$$r_A = \int_0^{x_A} \left[1 + \left(\frac{dy}{dx} \right)^2 \right]^{1/2} dx, \quad (4.13)$$

donde $\frac{dy}{dx}$ se calcula de la función (4.12), y es:

$$\frac{dy}{dx} = \frac{-20}{e^{4-20x} + e^{-4+20x} + 2}$$

La ecuación (4.13) se integra numéricamente en la subrutina BOUNDSURF (sección A.5.2) cuando INT=0, y de ahí se obtiene r_A como función de x_A , donde x_A es la coordenada x local $0 \leq x_A \leq x_{MAX}$. La longitud total se define como:

$$r_{1AB} = r_A(x_{MAX}). \quad (4.14)$$

La función de deformación (4.2) produce un parámetro normalizado r_{AB}^J tal que: $0 \leq r_{AB}^J \leq 1$ para $\xi_A \leq \xi \leq \xi_B$, donde $0 \leq J \leq NX$, $J=1,2,3,\dots$, y NX es el número de nodos en la dirección ξ . Consecuentemente, la coordenada superficial $r_{AB,D}$ se obtiene haciendo el siguiente producto:

$$r_{AB,D} = r_{AB}^J r_{1AB}, \quad (4.15)$$

donde r_{AB}^J es evaluada en la subrutina STRECH (sección A.5.3) (Fletcher 1991), y r_{1AB} está definida en la ecuación (4.14).

Las coordenadas físicas (x,y) de los puntos sobre la curva AB se obtienen interpolando r_A y x_A , y usando la ecuación (4.12) para la coordenada y correspondiente. Esto se hace en la subrutina BOUNDSURF (sección A.5.2) cuando INT=1.

Las coordenadas físicas de AB se denotan como XS(1,J), YS(1,J) en el programa MALLAS.f (sección A.5.1) pues son las componentes x y y de la curva \vec{Z}_1 de la sección 4.1.2. La superficie frontera CD se lleva de forma similar al arreglo de puntos XS(4,J), YS(4,J). Debido a que la expresión para las curvas y sus derivadas se utilizan en su forma general, $y=f(x)$, el procedimiento es útil para cualquier par de curvas expresadas de esa manera.

El método de multisuperficies (sección 4.1.2) es aplicado a cuatro curvas. Las curvas AB y CD constituyen a las funciones vectoriales \vec{Z}_1 y \vec{Z}_4 respectivamente. Inicialmente, las curvas intermedias \vec{Z}_2 y \vec{Z}_3 se construyen con una interpolación lineal de las superficies \vec{Z}_1 y \vec{Z}_4 , a una distancia fraccional $s_2=0.1$ y $s_3=0.9$ entre ellas como sigue:

$$\vec{Z}_2^{INICIAL} = \vec{Z}_1 + s_2(\vec{Z}_4 - \vec{Z}_1), \quad \text{y} \quad \vec{Z}_3^{INICIAL} = \vec{Z}_1 + s_3(\vec{Z}_4 - \vec{Z}_1).$$

Estas superficies están en un lugar físicamente correcto pero les hace falta un ajuste del parámetro r para que las líneas que unen a los puntos con la misma r en \vec{Z}_1 y \vec{Z}_2 salgan ortogonales a \vec{Z}_1 . De esta forma, el algoritmo de multisuperficies asegura la ortogonalidad local a \vec{Z}_1 y \vec{Z}_4 .

Como $\vec{Z}_1(r)$ es conocido, es suficiente tomar cada punto XS(1,J), YS(1,J), y proyectarlo en una línea recta normal a \vec{Z}_1 hasta que interseque a \vec{Z}_2 . El punto de intersección se convierte en

$XS(2,J)$, $YS(2,J)$. Este procedimiento es aplicado en la subrutina SURCH (sección A.5.4) (Fletcher 1991). De forma análoga se construyen los puntos en la curva \vec{Z}_3 , $XS(3,J)$, $YS(3,J)$, ortogonales a \vec{Z}_4 .

Una vez calculadas las coordenadas de las cuatro superficies, el algoritmo de multisuperficies para cuatro superficies (Eiseman 1979) se utiliza de la siguiente forma:

$$X(J,K) = \sum_{L=1}^4 SH(L)XS(L,J), \quad Y(J,K) = \sum_{L=1}^4 SH(L)YS(L,J),$$

donde

$$\begin{aligned} SH(1) &= (1-s)^2(1-a_1s), \\ SH(2) &= s(1-s)^2(a_1+2), \\ SH(3) &= s^2(1-s)(a_2+2), \\ SH(4) &= s^2(1-a_2(1-s)), \\ a_1 &= \frac{2}{3a_w-1} \quad \text{y} \quad a_2 = \frac{2}{2-3a_w}, \end{aligned}$$

s es un parámetro normalizado de deformación que se obtiene por interpolación en la dirección ξ

$$s = s_{AB}(K) + \xi(J)[s_{BC}(K) - s_{AD}(K)].$$

El parámetro a_w puede ser movido entre los valores 0.5 y 0.6, y proporciona un control sobre la uniformidad de la malla interior (Eiseman 1979).

Este algoritmo que genera la malla interna dadas dos curvas fronteras de la forma $y=f(x)$, está codificado en el programa MALLAS.f (sección A.5.1) con las subrutinas BOUNDSURF (sección A.5.2), STRECH (sección A.5.3), SURCH (sección A.5.4) y ENSIGHT (sección A.5.5).

Esta última subrutina proporciona el formato de salida necesario para que la malla sea visualizada con el programa comercial Ensignt (CEI 1994).

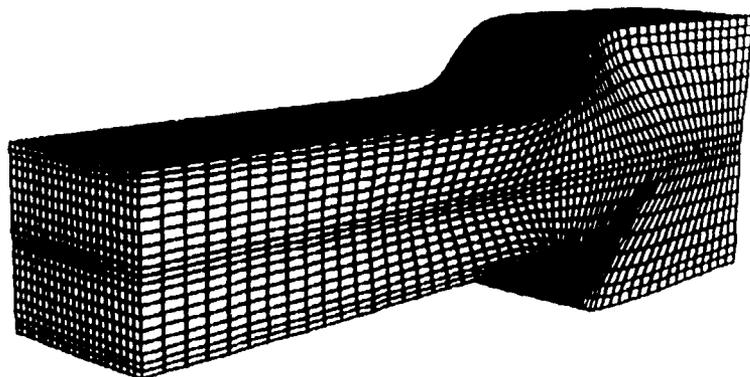


Fig. 4.5. Difusor tridimensional.

Los parámetros utilizados en el programa se presentan en la tabla A.5.1. Un ejemplo de difusor tridimensional se muestra en la figura 4.5 donde se hace evidente la capacidad del código para controlar los puntos en las fronteras y para asegurar ortogonalidad local en las mismas.

4.2.2 TUBOTE.f : código de generación de malla para tubos rectos, con codos y en forma de "s".

Para finalizar este capítulo se muestran las mallas generadas por el código llamado TUBOTE.f, cuyo listado se encuentra en la sección A.5.6. Este código genera una malla tridimensional ya sea en forma de canal recto con sección transversal cuadrada o sus deformaciones a tubos con codos intermedios en forma de arco de círculo y en forma de "s".

Una virtud del algoritmo (sección A.5.6) es el uso de la función de deformación (4.2) para ajustar la densidad de la malla. Estas son calculadas en la subrutina STRECH (sección A.5.3) y el formato de salida de la malla es construido en la subrutina ENSIGHT (sección A.5.5).

El conjunto de parámetros utilizados en el programa TUBOTE.f (sección A.5.6) son descritos en la tabla A.5.2. En la figura 4.6 se muestran tres ejemplos de los tubos resultantes de este generador de mallas.

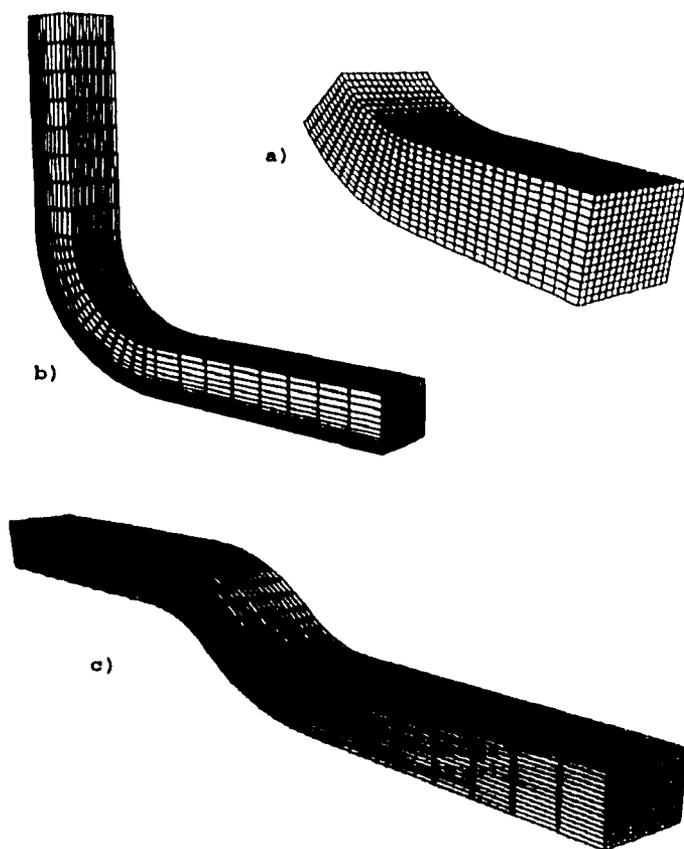


Fig. 4.6. Tubos generados con el programa TUBOTE.f (sección A.5.6). Los tres tipos de tubo están determinados por el parámetro NREFLEX (línea 12, sección A.5.6).
 a) Tubo con inicio curvo, b) Tubo con codo y malla densa en la cara de abajo, c) Tubo en forma de s.

5. Resultados.

En este capítulo se exponen tres ejemplos de resultados obtenidos con el programa NECRONOM.f. En la sección 5.1 se resuelve numéricamente el flujo dentro de un tubo recto de sección transversal cuadrada. Esto permite efectuar una validación simple del código, pues es posible hacer la comparación de los resultados numéricos con la solución analítica de flujo totalmente desarrollado. En la sección 5.2 se expone la solución obtenida para el flujo dentro de un tubo de sección cuadrada y con un codo en su parte media. Esta solución se discute en términos de los campos de presión y velocidad en el plano medio del tubo y de los campos de velocidad en tres planos transversales en diferentes posiciones a lo largo del tubo. En la sección 5.3 se hace un análisis similar para un tubo en forma de "s". Todos los cálculos presentados en este capítulo se efectuaron en una estación de trabajo Indigo 2 Silicon Graphics con precisión sencilla (palabra de 32 bits).

5.1 Validación del código.

5.1.1 Descripción del problema.

Considérese un tubo recto de sección cuadrada como se muestra en la Fig. 5.1. Sus dimensiones son: 10cm de largo, 1cm de ancho y 1cm de alto. El tubo está lleno de un fluido newtoniano, incompresible, con viscosidad dinámica μ y densidad ρ . Inicialmente ($t=0$) se considera un campo de velocidades constante $\vec{U} = (U, 0, 0)$ con $U = 10^3$ m/s. Para $t > 0$ se considera que el flujo es promovido por un suministro constante de fluido a la entrada.

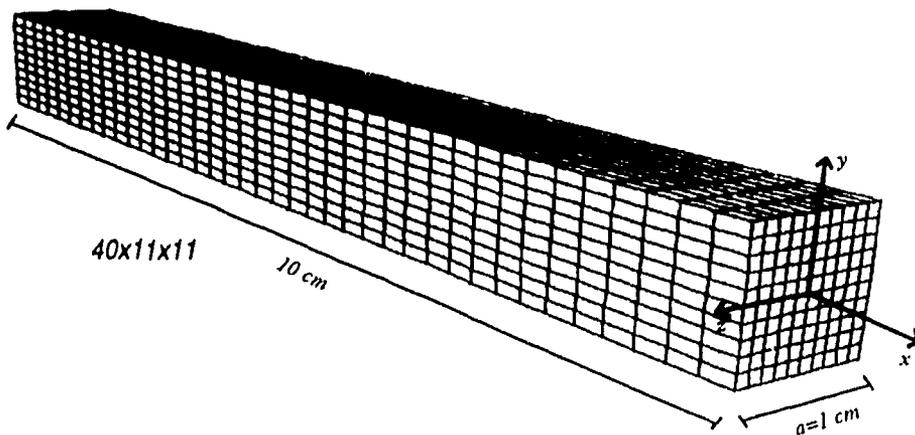


Fig. 5.1. Tubo recto de sección cuadrada.

La descripción del movimiento se obtiene al resolver las ecuaciones que gobiernan el flujo en su interior junto con la condición de pared sólida ($\|\vec{U}\|=0$). Esta condición se aplica a todas las fronteras excepto a las caras de entrada y salida, orientadas con el eje x . En la entrada se impone un perfil de velocidades constante ($\|\vec{U}\|=0.001\text{m/s}$), mientras que en la salida sólo se asegura que el flujo que sale sea igual al que entra.

El flujo másico dentro del tubo es:

$$\int_{\text{entrada}} \rho \vec{U} \cdot d\vec{A} = 10^{-4} \text{kg/s},$$

y el flujo de cantidad de movimiento:

$$\left\| \int_{\text{entrada}} (\rho \vec{U}) \vec{U} \cdot d\vec{A} \right\| = 10^{-7} \text{kgm/s}^2.$$

El fluido que se simula tiene las propiedades físicas del agua en condiciones normales, que son: densidad $\rho = 10^3 \text{kg/m}^3$ y viscosidad $\mu = 8 \times 10^{-4} \text{kg/(m s)}$. Por lo tanto, el número de Reynolds es:

$$\text{Re} = \frac{\rho \|\vec{U}\| a}{\mu} = 12.5.$$

Los resultados numéricos se encuentran usando una malla de $40 \times 11 \times 11$ volúmenes y el cálculo se lleva a cabo durante 20 pasos de tiempo de 1s cada uno. Se alcanza un residuo en la ecuación de conservación de masa de $0.2 \times 10^{-8} \text{kg/s}$ y un residuo en la ecuación de conservación de cantidad de movimiento de $0.1 \times 10^{-10} \text{kg m/s}^2$. Esto corresponde al 0.01% del flujo másico y 0.1% del flujo total de cantidad de movimiento. La computadora tardó seis minutos y dos segundos en hacer el cálculo completo, utilizando 92% de su capacidad en CPU. Esto corresponde a 18.1s de CPU por cada segundo de cálculo.

5.1.2 Solución analítica.

El flujo dentro del canal presenta evolución temporal y evolución espacial que durante el cálculo numérico se desarrollan simultáneamente. La primera se presenta debido a que las condiciones iniciales para los campos de velocidad y presión que se impusieron no corresponden a los campos en estado permanente. La segunda es una consecuencia de la anisotropía impuesta el flujo debido al flujo dominante en la dirección axial. Esto último provoca que el mecanismo de transporte de la vorticidad a lo largo del eje sea la convección mientras que en las direcciones transversales sea la difusión. En consecuencia, las capas límite se comienzan a formar en la entrada del canal y crecen hacia el centro en regiones aguas-abajo. A partir de la zona donde las capas límite se encuentran, el perfil dinámico no se modifica, el gradiente de presión es constante en la dirección axial y se dice que el flujo está totalmente desarrollado. Bajo estas condiciones, los cambios temporales y a lo largo del eje x son cero. También se tiene que la velocidad sólo presenta componente en la dirección axial y ésta sólo es función de las coordenadas transversales.

Considerando el flujo completamente desarrollado, las ecuaciones (3.1) y (3.2) que describen el movimiento se reducen a:

$$\frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = \frac{1}{\mu} \frac{\partial p}{\partial x} = cte,$$

con condiciones de frontera $U(y = \pm 0.5a, z) = U(y, z = \pm 0.5a) = 0$.

La solución de esta ecuación es (Timoshenko y Goodier 1951):

$$U = G \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n^3} (-1)^{\frac{(n-1)}{2}} \left[1 - \frac{\cosh(n\pi y / a)}{\cosh(n\pi / 2)} \right] \cos\left(\frac{n\pi z}{a}\right), \quad (5.1)$$

donde U es la componente x de la velocidad y G es una constante que se usa de tal manera que se obtiene una velocidad normalizada. Para obtener valores numéricos se ha calculado la suma para $n_{(max)}=281$.

5.1.3 Comparación de resultados numéricos y analíticos.

En la Fig. 5.2(a) se presentan las gráficas de los perfiles numérico y analítico para el flujo totalmente desarrollado, así como la resta de ambos para un corte central ($z=0$). La diferencia entre ambos se muestra en la Fig. 5.2(b). La diferencia máxima se alcanza en regiones cercanas a los bordes y es menor al 1%.

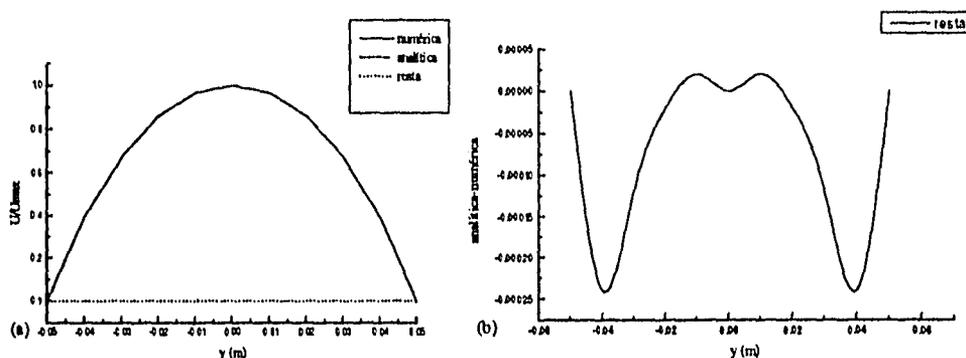


Fig. 5.2. (a) Perfil de velocidades numérico, analítico y resta para un corte central, (b) resta.

En la Fig. 5.3 se presenta el perfil numérico, el analítico y la resta de ambos, en toda la sección transversal. En la gráfica de la resta se observa que la máxima diferencia se alcanza cerca de las esquinas del dominio y también es cercana al 1%.

Con este resultado se valida el programa NECRONOM.f para el perfil totalmente desarrollado en un canal recto.

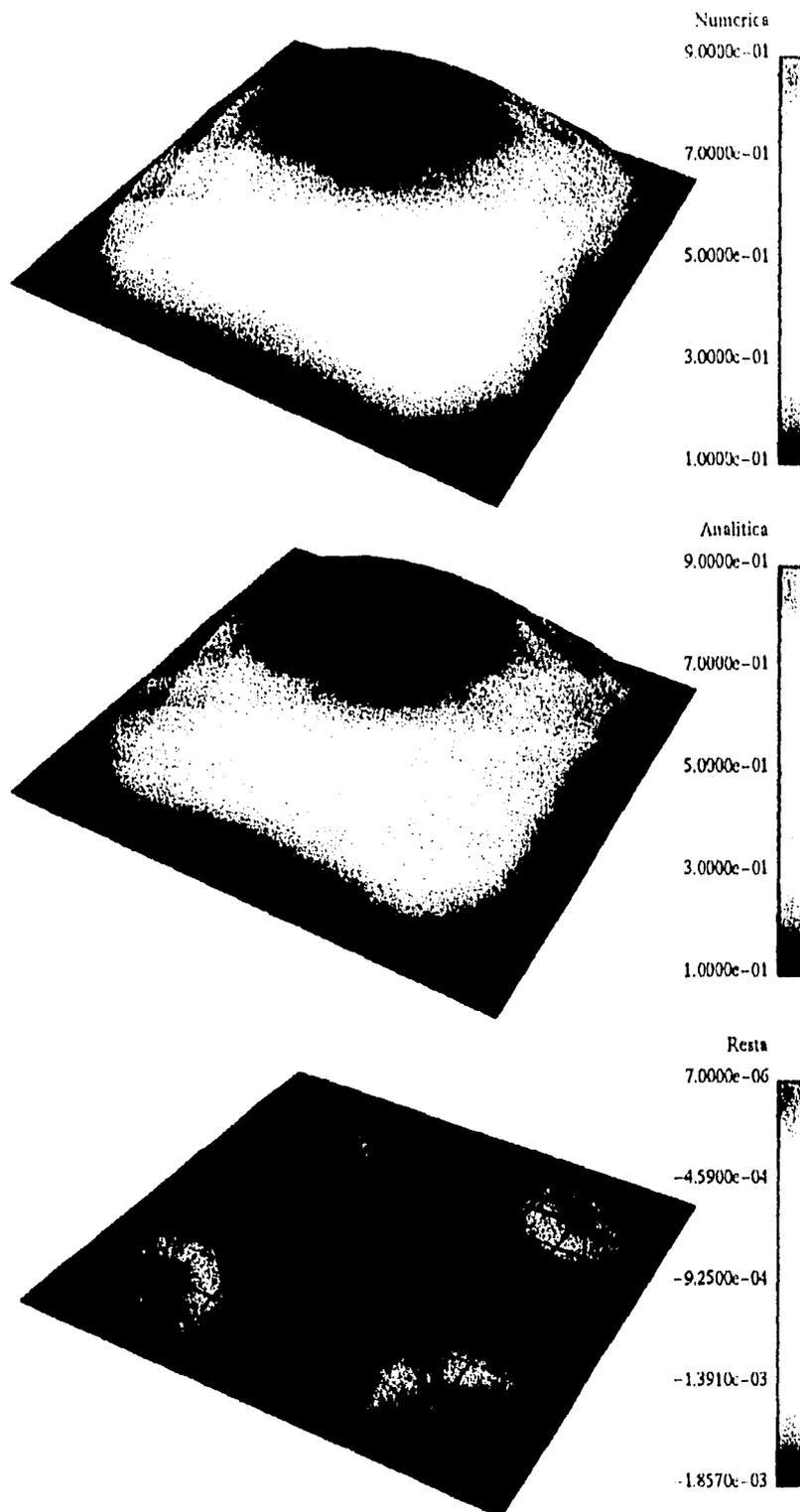


Fig. 5.3. Perfil de velocidad axial numérico, analítico y diferencia entre ambos sobre la sección transversal final del tubo recto de sección cuadrada en $t=20s$.

5.1.4 Perfil en desarrollo.

En la Fig. 5.4 se expone la evolución temporal del flujo hasta alcanzar el desarrollo. La condición de inicio indica que la velocidad es constante en todo el volumen. Aunque la condición inicial sería difícil de conseguir experimentalmente, es un punto de partida para analizar el tiempo y la distancia de desarrollo del perfil.

Se observa que se alcanza el flujo desarrollado cerca de los 8s (Fig. 5.4(e)) y a una distancia de 1 cm aproximadamente.

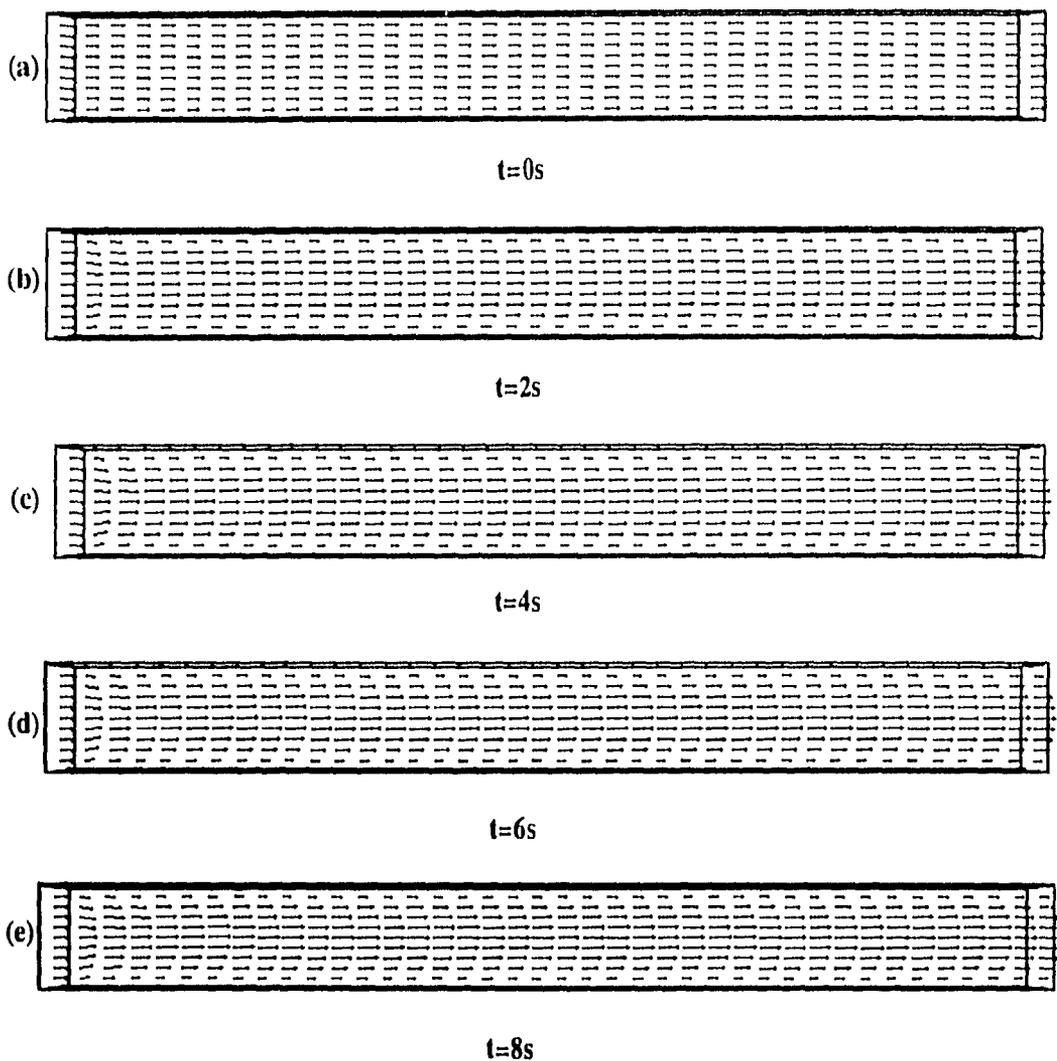


Fig. 5.4. Evolución temporal del flujo dentro de un tubo recto de sección cuadrada. Se toma como referencia la velocidad 0.001m/s en $t=0$.

El tiempo de desarrollo convectivo t_c , el difusivo t_D y la longitud de desarrollo L_d se calculan según las siguientes relaciones (Schlichting 1979):

$$t_c = \frac{X}{U}, \quad t_D = \frac{L^2 \rho}{\mu}, \quad \text{y} \quad L_d = LkRe/2.$$

donde X es la longitud del tubo, L es el lado de la sección, U es la velocidad característica, Re es el número de Reynolds, ρ es la densidad del fluido, μ es la viscosidad efectiva y $k=0.04$.

Los valores obtenidos para este problema son:

$$t_c=100s, \quad t_D=125s \quad \text{y} \quad L_d=0.0025m.$$

Los tiempos de desarrollo son diez veces mas grandes que el observado en el cálculo numérico y la distancia de desarrollo es cuatro veces menor que aquella observada en la Fig. 5.4. Debe tomarse en cuenta que en la estimación teórica de los tiempos de desarrollo se inicia del reposo, mientras que el problema resuelto numéricamente se comienza con un flujo de velocidad uniforme. Esto provoca la diferencia entre los tiempos calculados teóricamente y los calculados numéricamente. Por otro lado, la longitud de desarrollo obtenida numéricamente tiene la desventaja de depender mucho del número de nodos en la dirección x pues hay sólo cuatro nodos por cada centímetro. Dada la distancia estimada numéricamente como un centímetro, esto amplifica el error en la estimación en un 25%. Por lo tanto, es difícil hacer una buena estimación de la distancia de desarrollo a partir del cálculo numérico aquí presentado y está fuera del alcance de este análisis el dar una buena comparación entre la distancia de desarrollo teórica y la numérica.

5.2 Tubo con codo.

5.2.1 Descripción del problema.

En la Fig. 5.5 se muestra un tubo de sección cuadrada, simétrico respecto al plano medio B. Cada mitad del tubo está formada por dos partes: una recta y una sección circular. La parte recta mide 0.25m de largo y su sección transversal tiene 0.10m de lado. La sección circular forma un codo que subtiende un ángulo de 5.625 grados y tiene un radio máximo de 2.50m. El tubo tiene en total 0.99m de longitud.

La localización de los planos A, B y C, corresponde al inicio del codo (a 0.25m de la entrada), la mitad del tubo (a 0.495m de la entrada) y la salida del tubo (0.99m de la entrada) respectivamente. Estos planos se utilizan para monitorear el campo de velocidades.

En el interior del tubo se simula un fluido newtoniano, incompresible con la densidad y viscosidad del agua en condiciones normales como en el tubo recto de la sección 5.1. La velocidad de entrada se aumenta a 0.01 m/s y el campo de velocidades se inicializa con vectores de esta magnitud y en dirección de la coordenada generalizada ξ . Esta última sigue la dirección principal del tubo.

El número de Reynolds es de 1250, el flujo másico es de 0.1kg/s y el flujo total de cantidad de movimiento es de 0.001kgm/s².

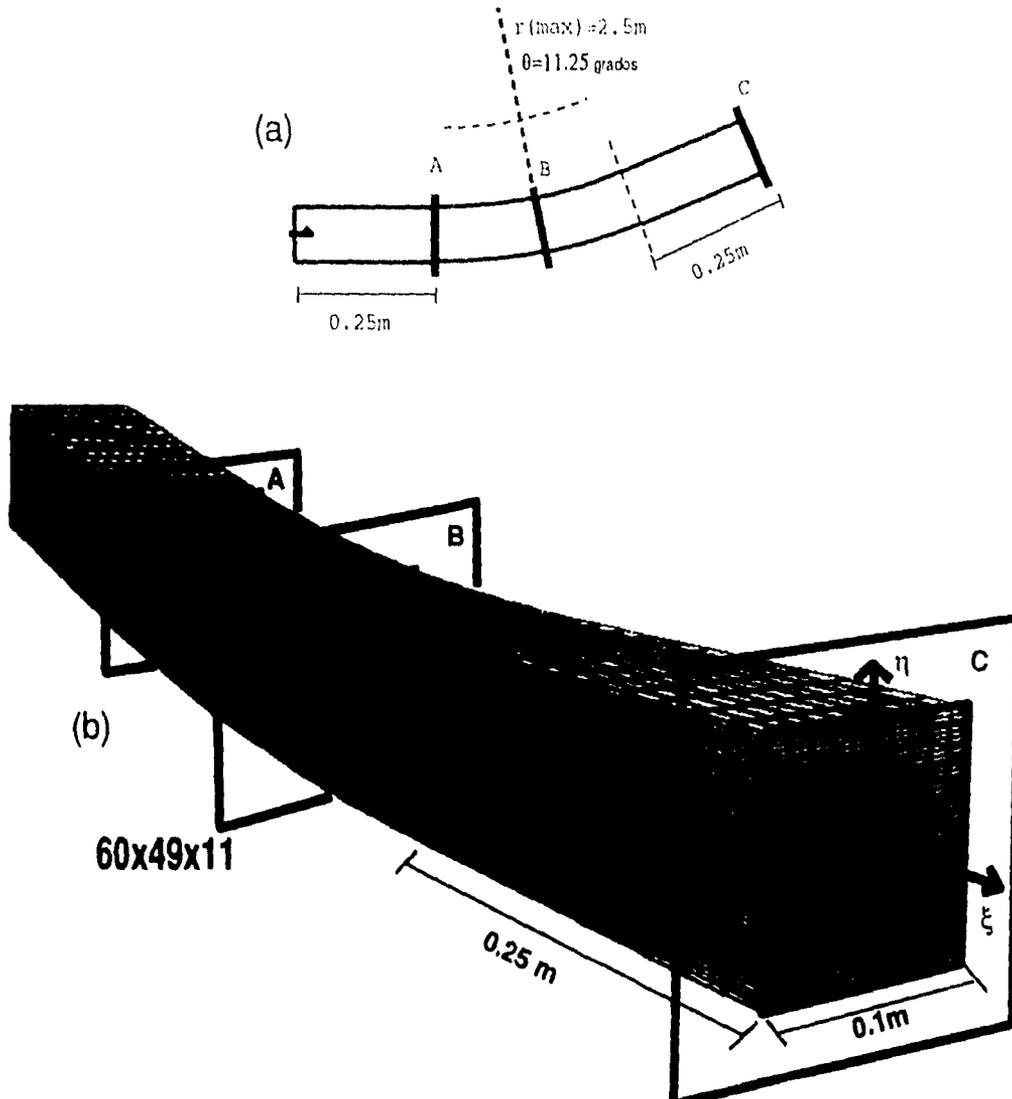


Fig. 5.5. Tubo de sección cuadrada con codo, (a) corte longitudinal, (b) perspectiva.

Los resultados numéricos se encuentran usando una malla de $69 \times 49 \times 11$ volúmenes y el cálculo se lleva a cabo durante 30 pasos de tiempo de 5s cada uno. Se alcanza un residuo en la ecuación de conservación de masa de $0.7 \times 10^{-3}\text{kg/s}$ y un residuo en la ecuación de

conservación de cantidad de movimiento de $0.22 \times 10^{-6} \text{ kg m/s}^2$. Esto corresponde al 0.001% del flujo másico y 2% del flujo total de cantidad de movimiento. El cálculo completo tardó tres horas y cuatro minutos, utilizando 57% de la capacidad en CPU. Esto corresponde aproximadamente a seis minutos de CPU por cada paso en el tiempo. Se eligieron las dimensiones del dominio, el número de nodos en la malla y la velocidad de inicio que permiten una convergencia satisfactoria con el mínimo de esfuerzo computacional. Es conveniente notar que cálculos con mallas compuestas por un menor número de nodos presentan problemas de convergencia.

5.2.2 Presentación de resultados.

a) Campos de presiones relativas.

En la Fig. 5.6 se muestra el campo de presiones relativas en el plano medio en dirección z para cuatro tiempos distintos. El campo se inicializa a cero y evoluciona de tal forma que la presión en la entrada disminuye gradualmente hacia la salida y en el codo genera una depresión en la zona cercana al radio interior.

b) Campos de velocidades.

En la Fig. 5.7 se muestra el campo de velocidades en el plano medio en dirección z para cuatro tiempos distintos. El campo se inicializa constante a 0.01 m/s y en dirección de las líneas coordenadas de ξ . En el tramo recto inicial, el campo de velocidades presenta una solución que tiende hacia el perfil desarrollado, análogo al presentado en la sección 5.1. En la sección de curvatura finita, los perfiles de velocidad denotan un transporte de cantidad de movimiento de la zona cercana al radio interior, hacia las regiones vecinas del radio exterior. Esta característica se hace más notoria a tiempos largos (ver $t=150\text{s}$ de la Fig. 5.7). Las zonas de aceleración y desaceleración del fluido corresponden a las regiones de gradiente de presión favorable y adverso respectivamente (ver la Fig. 5.6). El flujo se expone a la fuerza inducida por la pared que lo hace dar la vuelta y es sólo hacia la salida que empieza a recuperar el perfil de tubo recto aunque sin alcanzar a recuperarlo por completo. La distribución de velocidades aquí descrita corresponde cualitativamente a las observaciones experimentales efectuadas por Humphrey et al. (1977) en un tubo con un codo a 90 grados.

c) Perfiles de velocidad.

En la Fig. 5.8 se muestran tres curvas que representan los perfiles de velocidad axial en la intersección del plano medio en dirección z con los tres planos A, B y C de la Fig. 5.5.

El perfil en el plano A es prácticamente simétrico, mientras que los perfiles de los planos B y C reflejan la fuerza a la que el fluido está sometido en la curva y que lo hace desplazar su máximo de velocidad hacia la cara externa de curvatura. Se puede observar que la

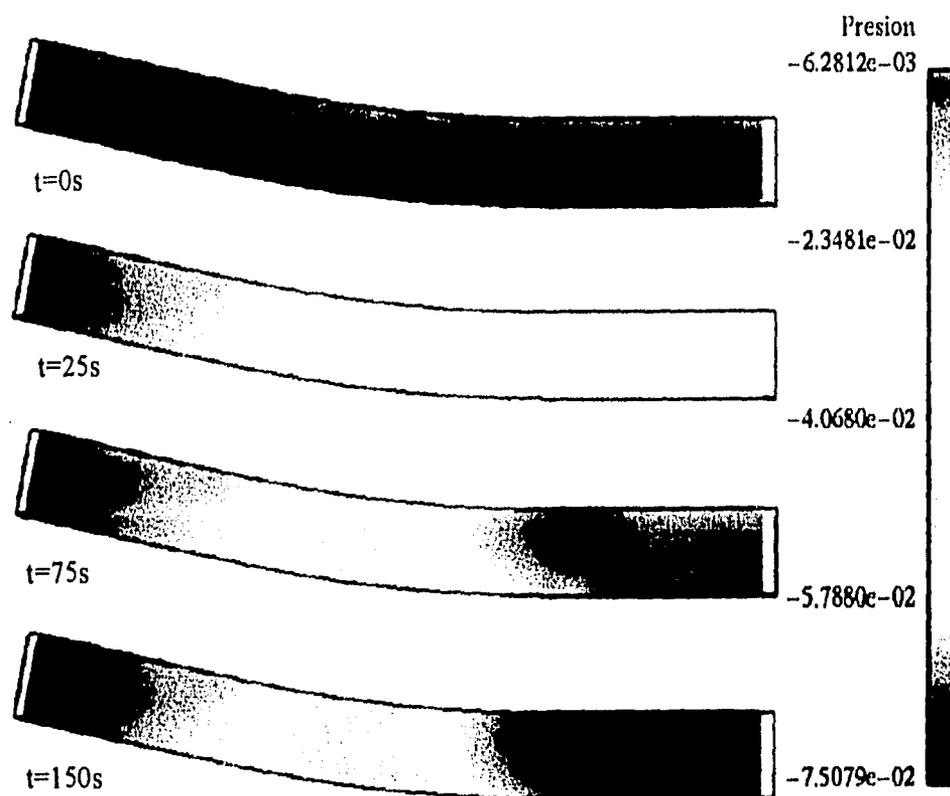


Fig. 5.6. Evolución temporal del campo de presiones relativas en el plano medio en dirección z.

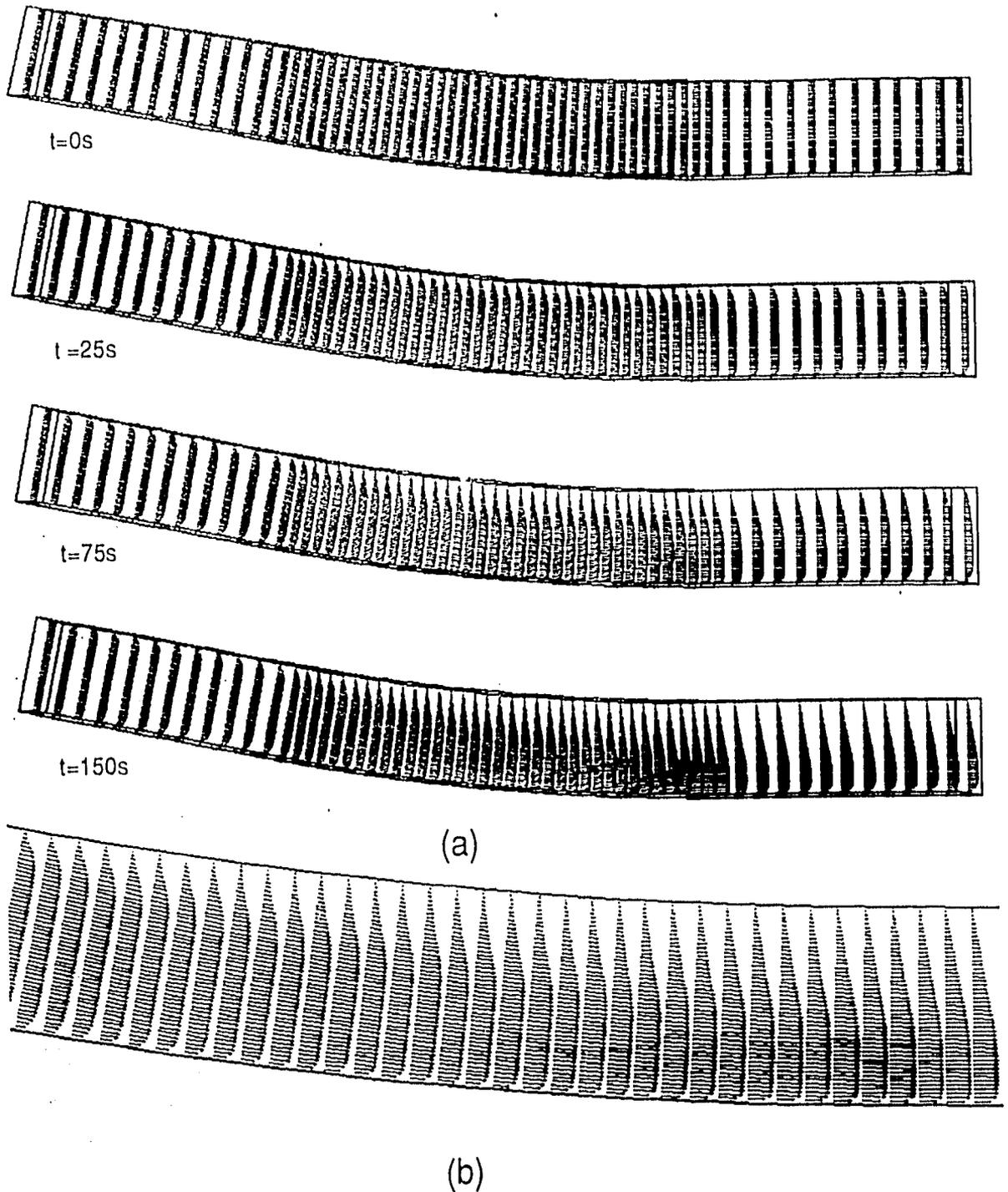


Fig. 5.7. (a) Campos de velocidades en el plano medio en dirección z para cuatro pasos de tiempo distintos, la velocidad de referencia es la de inicio (0.01m/s), (b) detalle de $t=150s$.

velocidad máxima debido a la aceleración provocada por la curvatura alcanza hasta 1.5 veces el valor de la velocidad en la entrada.

La Fig. 5.9 muestra las superficies formadas por la velocidad axial en los planos A, B y C. Las características mostradas son similares a las discutidas en el párrafo anterior. Es claro que el flujo no está totalmente desarrollado pues la capa límite no ha terminado de formarse. En el plano A se tiene un perfil casi simétrico pero muy chato. En el plano B se observa el traslado de la velocidad máxima a la pared externa del codo, y en el plano C, además de este traslado, se observa cierta concavidad en la parte central hacia la pared interna del codo que indica un transporte de cantidad de movimiento hacia las paredes laterales del tubo.

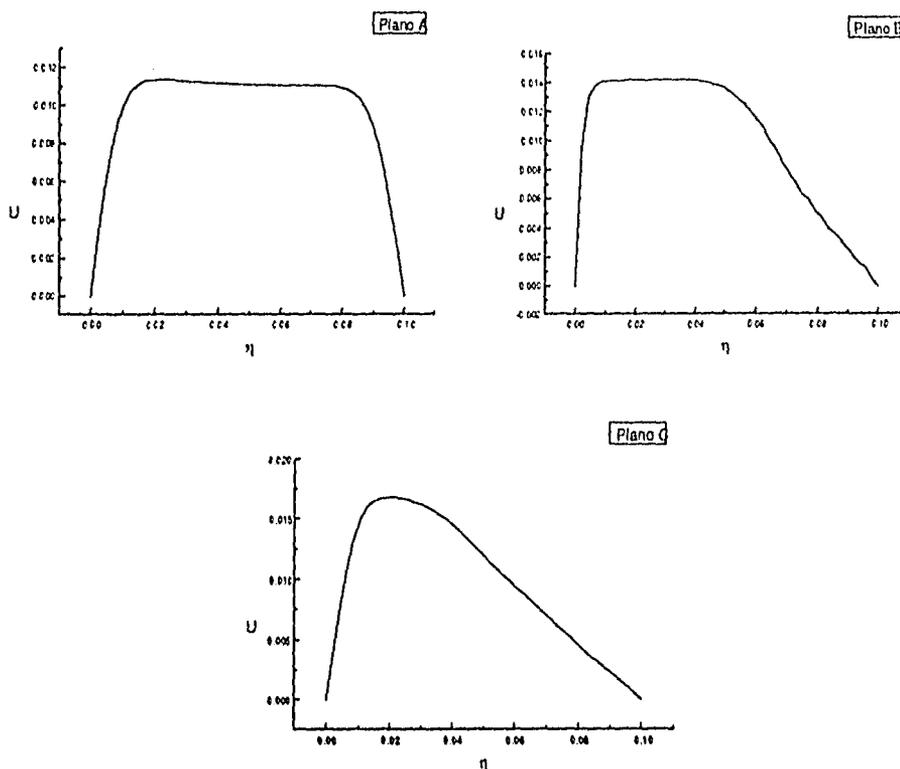


Fig. 5.8. Perfiles de velocidad axial en tres planos para un tubo con codo.

En la Fig. 5.10 se presentan tres pasos de tiempo en la evolución de la superficie de velocidades en el plano de salida C donde se puede observar la formación de la concavidad descrita en el párrafo anterior.

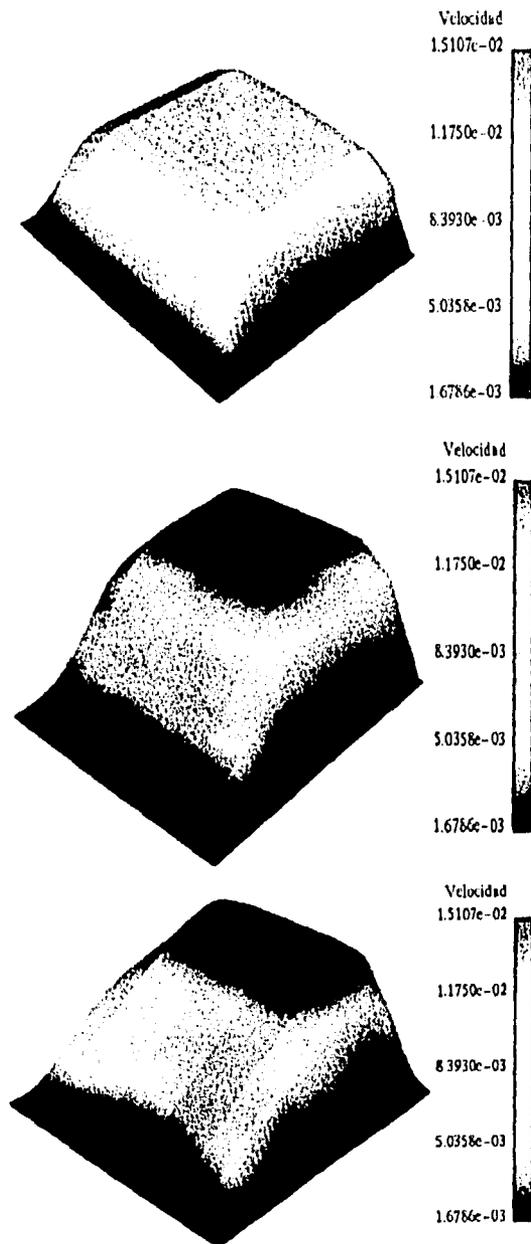


Fig. 5.9. Superficies de velocidad al tiempo final en los planos A, B y C.

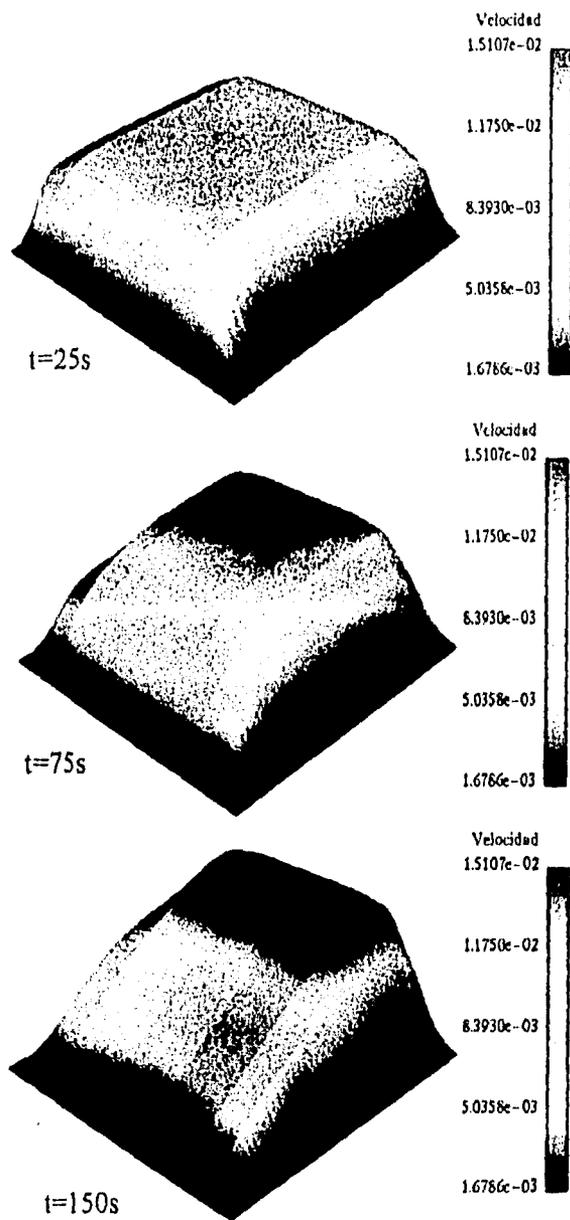


Fig. 5.10. Evolución temporal de la superficie de velocidades en la salida del tubo con codo.

5.3 Tubo en forma de "s".

5.3.1 Descripción del problema.

En la Fig. 5.11 se muestra un tubo en forma de "s" y de sección cuadrada, que ha sido generado haciendo una rotación de 180 grados al tubo con codo de la sección 5.2 sobre el eje del plano B de la Fig. 5.5. De esta manera, el codo del tubo de la Fig. 5.5 se ha deformado de tal forma que ahora se compone de dos partes con direcciones de curvatura contrarias y que empalman suavemente. Cada una de estas partes subtiende el mismo ángulo que tenía la mitad del codo (5.625 grados) y el mismo radio máximo (2.50m) pero ahora cada mitad de la zona que forma la "s" tiene el radio en direcciones opuestas.

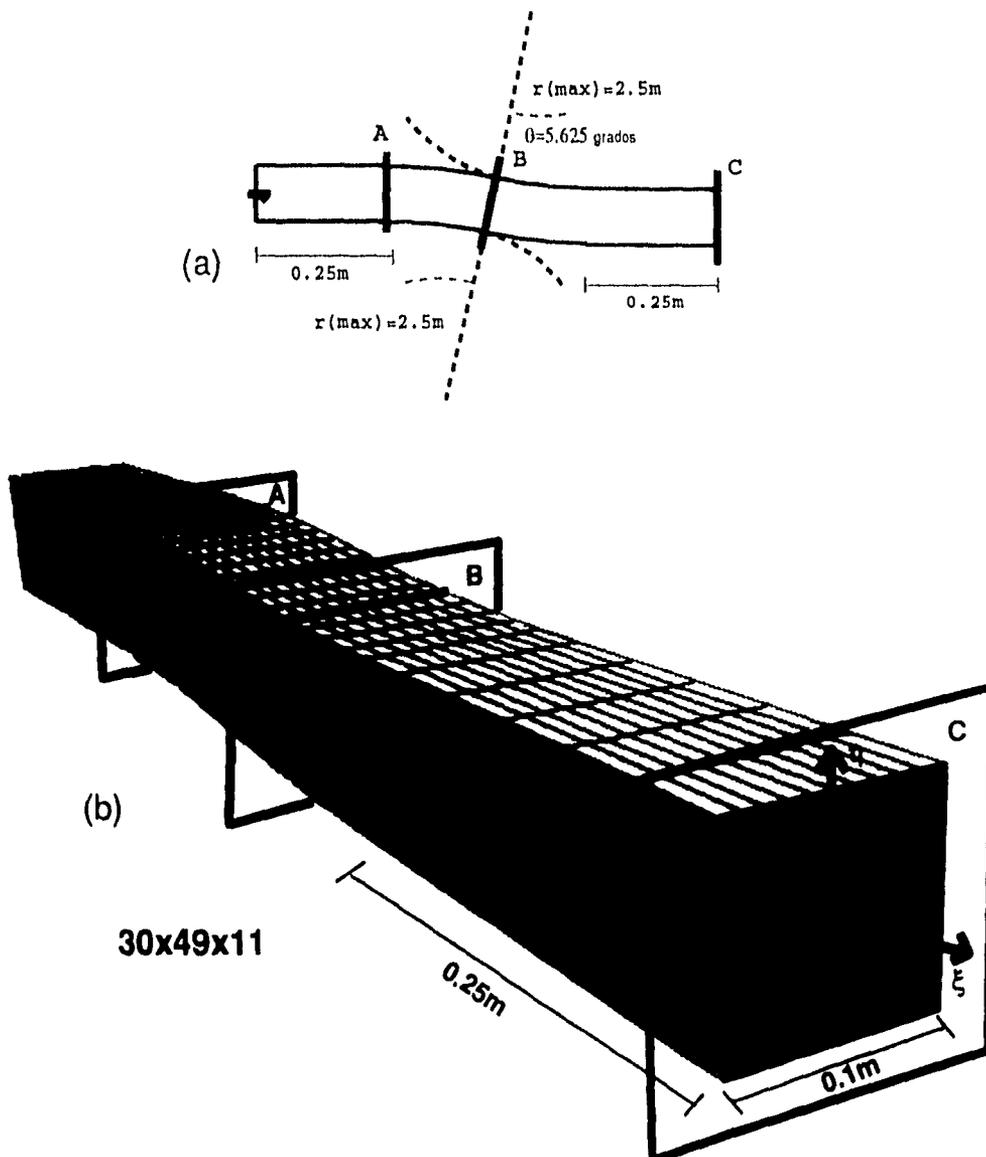


Fig. 5.11. Tubo de sección cuadrada en forma de "s", (a) corte longitudinal, (b) perspectiva.

La localización de los planos A, B y C, corresponde al inicio de la primera curva (a 0.25m de la entrada), la mitad de la "s" (a 0.495m de la entrada) y la salida del tubo (0.99m de la entrada) respectivamente. Estos planos se utilizan para monitorear el perfil de velocidad.

En el interior del tubo se simula un fluido newtoniano, incompresible con la densidad y viscosidad del agua en condiciones normales y con la misma velocidad de entrada e inicio usados en la sección 5.2. La inicialización de la velocidad toma en cuenta las nuevas curvas de dirección de la coordenada generalizada ξ .

Al igual que en la sección 5.2, el número de Reynolds es de 1250, el flujo másico es de 0.1kg/s y el flujo total de cantidad de movimiento es de 0.001kgm/s².

Los resultados numéricos se encuentran con una malla de 30x49x11 volúmenes y el cálculo se lleva a cabo durante 30 pasos de tiempo de 5s cada uno. Se alcanza un residuo en la ecuación de conservación de masa de 0.6×10^{-5} kg/s y un residuo en la ecuación de conservación de cantidad de movimiento de 0.9×10^{-7} kg m/s². Esto corresponde al 0.06% del flujo másico y 0.9% del flujo total de cantidad de movimiento. El cálculo completo tardó cuarenta y dos minutos, utilizando 97% de la capacidad en CPU. Esto corresponde a 1.4 minutos de CPU por cada paso en el tiempo.

5.3.2 Presentación de resultados.

a) Campos de presiones relativas.

En la Fig. 5.12 se muestra el campo de presiones relativas en el plano medio en dirección z para cuatro tiempos distintos. El campo se inicializa a cero y evoluciona de tal forma que la presión en la entrada disminuye gradualmente hacia la salida y en la "s" genera una depresión en las zonas cercanas al radio interior. Sobre la parte media de la "s" existe un punto de máxima presión cerca de la pared superior y un punto de mínima presión cerca de la pared inferior.

b) Campos de velocidades.

En la Fig. 5.13 se muestran los campos de velocidades en el plano medio en dirección z para cuatro tiempos distintos. El campo se inicializa constante a 0.01m/s y en dirección de las líneas coordenadas de ξ . Este evoluciona de tal forma que al comienzo de la "s", el máximo de la velocidad del fluido se desplaza hacia las regiones vecinas al radio exterior. En el plano B, donde la "s" cambia de concavidad, el campo de velocidades tiende a simetrizarse, y al proceder en la zona de concavidad inversa, el efecto de la curvatura se invierte provocando una fuerza hacia la zona cercana al radio exterior que empuja al campo de velocidades en sentido contrario. La salida del tubo es corta por lo que el fluido no alcanza a recuperar una forma simétrica y permanece con el máximo de velocidad desplazado.

Sobre los puntos de presión máxima y mínima puede observarse que la capa límite es muy delgada y se insinúa la generación de un vórtice. Para resolver esta parte del flujo,

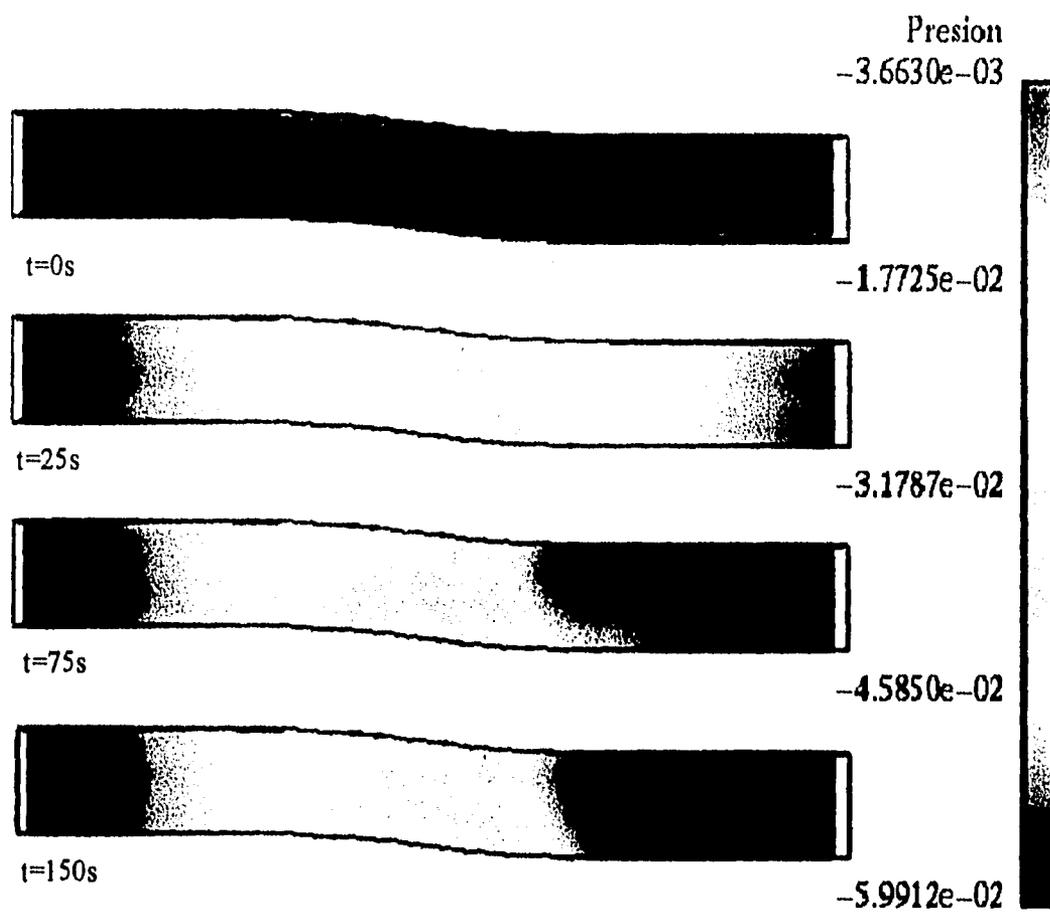


Fig. 5.12. Evolución temporal del campo de presiones relativas en el plano medio en dirección z.

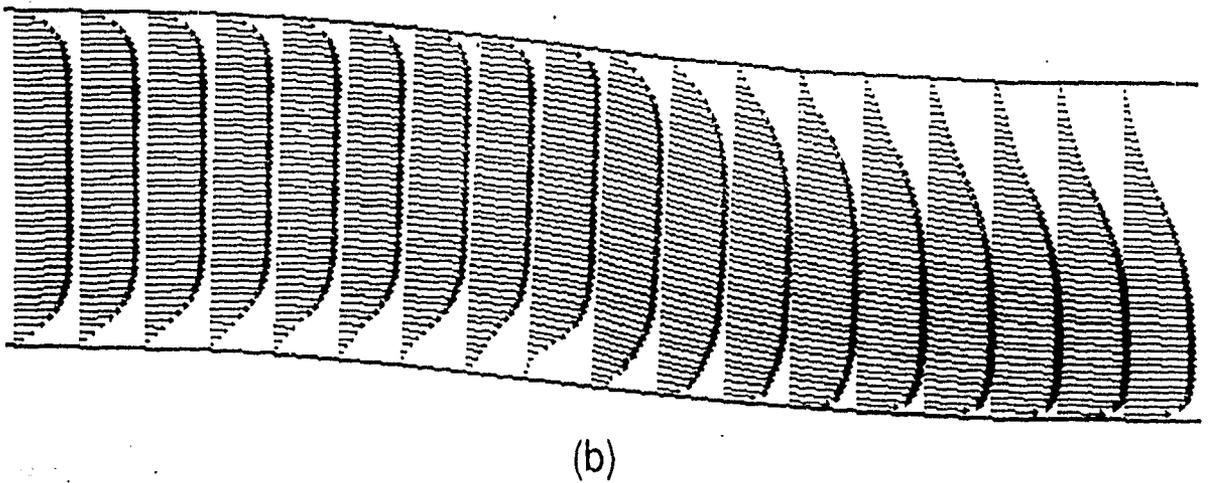
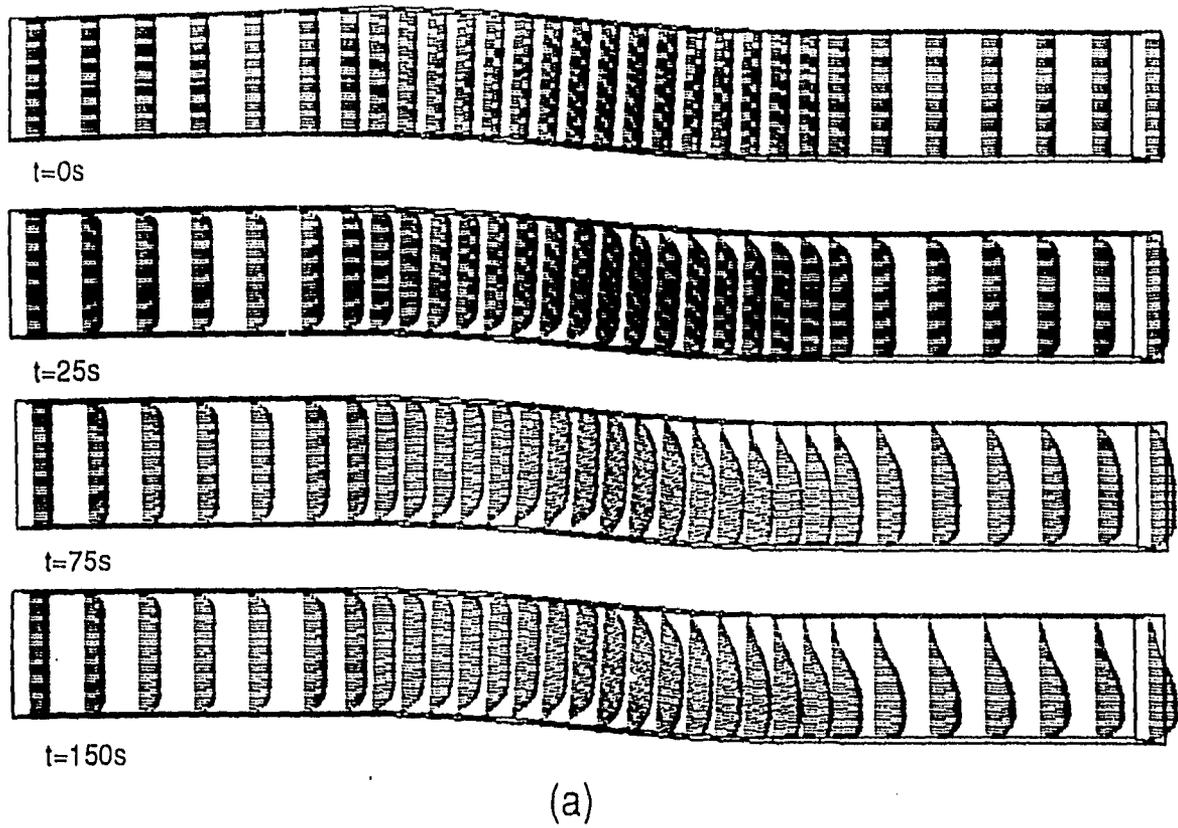


Fig. 5.13. (a) Campos de velocidades en el plano medio en dirección z para cuatro pasos de tiempo distintos, la velocidad de referencia es la de inicio (0.01m/s), (b) detalle de $t=150\text{s}$.

sería necesario un estudio en una malla más fina.

e) Perfiles de velocidad.

En la Fig. 5.14 se muestran tres curvas que representan los perfiles de velocidad axial en la intersección del plano medio en dirección z con los tres planos A, B y C de la Fig. 5.11. En el plano A, el perfil de velocidades tiene una ligera tendencia hacia la cara externa de curvatura en esa zona. En el plano B se puede observar que el campo de velocidades comienza a tomar la tendencia contraria al plano A y se nota una pequeña zona de recirculación cerca de la pared inferior ($\eta = 0$). En el plano de salida C, el perfil de velocidad es muy parecido al obtenido en la salida del tubo con codo pues el efecto de la última curvatura es el que predomina.

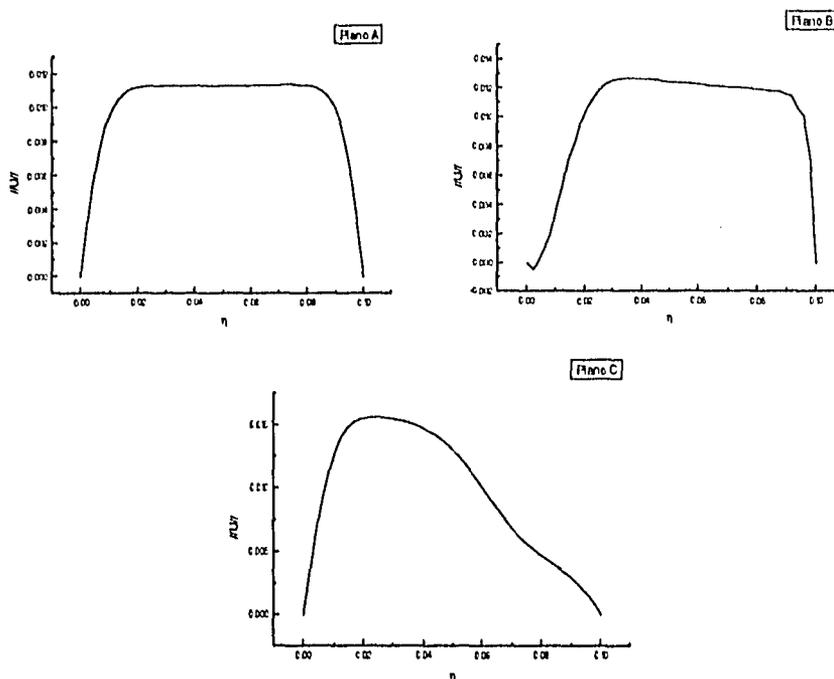


Fig. 5.14. Perfiles de velocidad axial en tres planos para un tubo en "s".

La Fig. 5.15 muestra las superficies formadas por la velocidad en los planos A, B y C. En este ejemplo también es claro que el flujo no está totalmente desarrollado pues la capa límite no ha terminado de formarse. En el plano A se nota que el punto de máxima velocidad comienza a desplazarse hacia la cara exterior de curvatura pero es aún muy chato. En el plano B se observa que el máximo de velocidad vuelve a acercarse al centro para tomar la curva contraria, y en el plano C se observan, de forma exagerada, las características encontradas para la superficie de velocidad de salida en el problema del tubo con codo, es decir, se presenta el traslado del máximo de velocidad hacia las zonas cercanas al radio exterior.

En la Fig. 5.16 se presentan tres pasos de tiempo en la evolución de la superficie de velocidades en el plano de salida C.

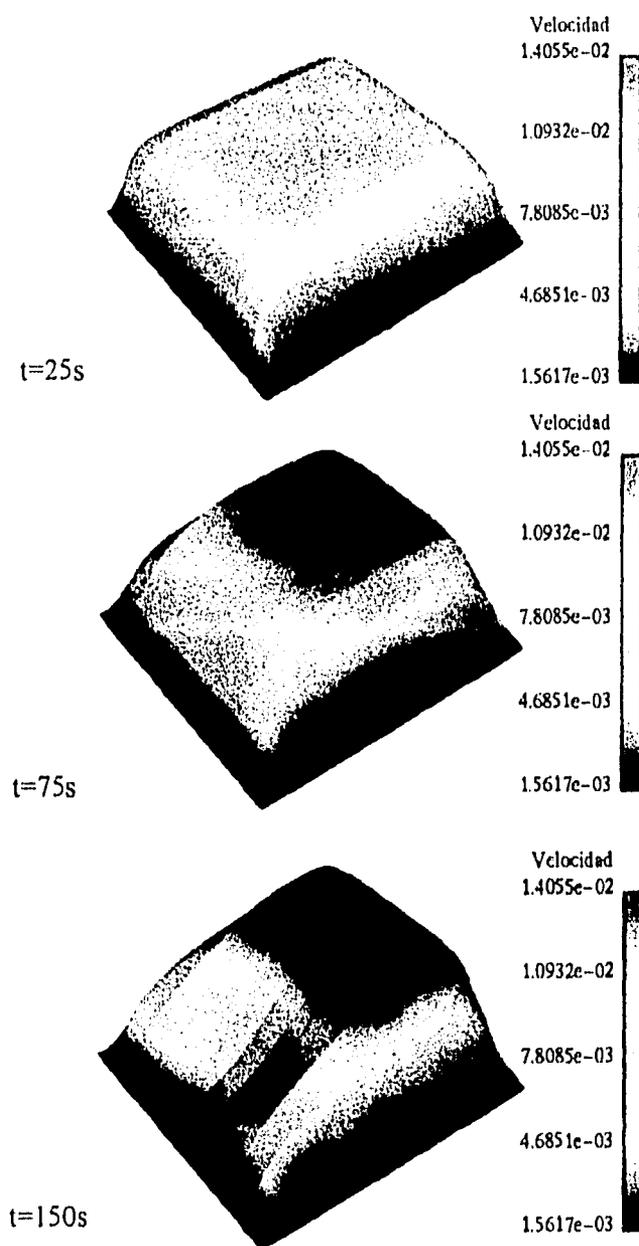


Fig. 5.16. Evolución temporal de la superficie de velocidad en el plano de salida C para un tubo en "s".

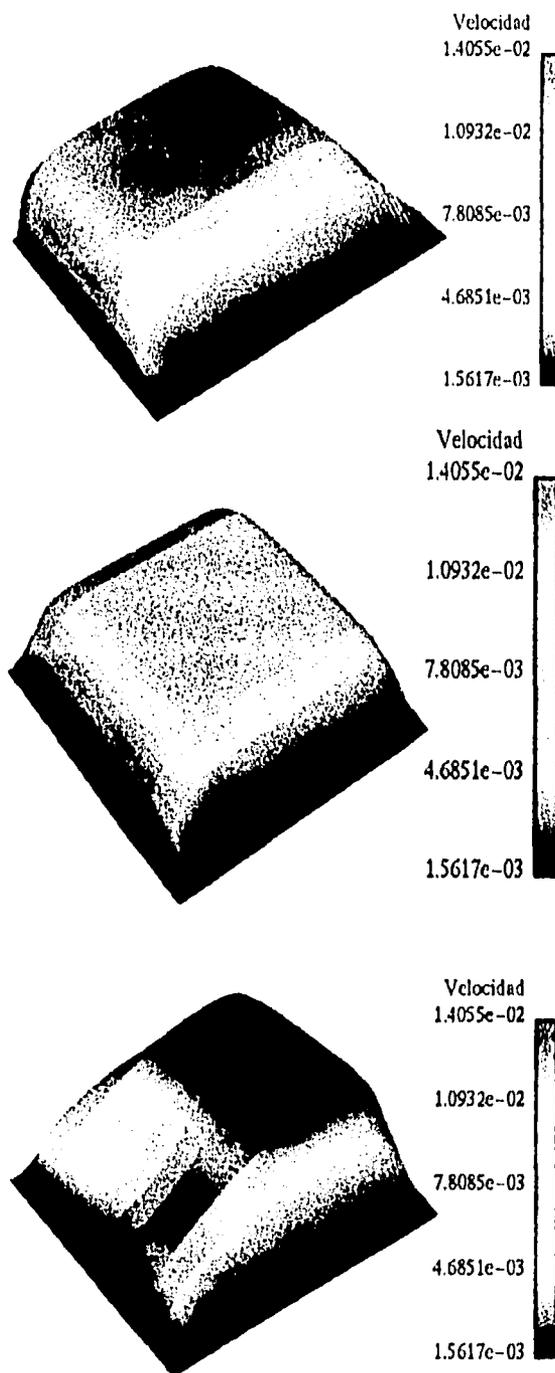


Fig. 5.15. Superficies de velocidad a tiempo final en los planos A,B y C del tubo en "s".

6. Discusión y conclusiones.

Se ha codificado con éxito un programa de cómputo capaz de incorporar la técnica de coordenadas generalizadas al método de discretización de volumen finito y ha sido utilizado para resolver numéricamente las ecuaciones de Navier-Stokes en dominios no regulares. Se han identificado los factores que surgen de la transformación de las ecuaciones de Navier-Stokes a sistemas de coordenadas generalizadas como propiedades geométricas del dominio de estudio. Es equivalente plantear esta transformación en términos de las modificaciones que impone el cambio de coordenadas a las derivadas parciales que aparecen en las ecuaciones, a plantearlo como una integración directa de las ecuaciones en cada elemento del dominio de estudio. Este planteamiento ha permitido escribir las modificaciones en términos de los vectores de área y de los volúmenes de los elementos que componen a la geometría. Dado un dominio tridimensional con fronteras fijas, se ha podido desarrollar un programa de cómputo capaz de resolver el flujo en su interior con condiciones de frontera de no resbalamiento en las paredes, con el flujo de entrada fijo y asegurando la conservación de masa en su salida. Los ejemplos resueltos, muestran la capacidad del código para resolver dominios con curvatura y abren la posibilidad de atacar numéricamente este tipo de problemas con un esfuerzo computacional razonable.

Aún se requieren de muchas pruebas para hacer una evaluación completa del comportamiento del código con el fin de optimizarlo. Sobre este respecto, diversos autores han comentado las virtudes y defectos comúnmente encontrados en este tipo de técnicas computacionales. Ningún comentario adverso ha sido encontrado respecto al manejo de las propiedades geométricas que se incorporan en el código y en este renglón se puede estar satisfecho con el trabajo hecho. Sin embargo, varios autores reconocen que este tipo de códigos presentan inestabilidades cuando se utilizan en dominios con mallas poco densas, lo que aumenta el costo computacional cuando se trata de resolver geometrías de gran curvatura, pues es necesario hacer el cálculo utilizando una malla muy fina. Es un hecho conocido (Melaen 1992) que el método híbrido de primer orden utilizado para la discretización de los términos convectivos y difusivos, descrito en el apéndice 3, produce mucha difusión falsa al enfrentar zonas de gran deformación (Patankar 1980). Recientemente se han propuesto nuevos mecanismos de aproximación de esos términos que ayudan a ganar precisión y estabilidad al enfrentar zonas de gran deformación (Majumdar, Rodi y Zhu 1992) con la consecuente disminución del costo computacional. Esto sería tema de refinamientos futuros.

En los ejemplos presentados en esta tesis, se han evitado los problemas descritos en el párrafo anterior utilizando mallas suficientemente densas en elementos y con curvaturas pequeñas. En principio, todos los errores debidos a las aproximaciones de discretización

disminuyen al refinar la malla. La refinación debe llevarse a cabo siempre y cuando el esfuerzo computacional no sea mayor al requerido si se utilizan mejores aproximaciones en los términos que causan problemas. Por otro lado, cuando se encuentra un problema donde es difícil llegar a la convergencia del método, hay que tomar en consideración que no sólo puede atribuirse a problemas intrínsecos del método numérico, sino que existen algunos factores físicos que también juegan un papel en la desestabilización. La inestabilidad de algunas soluciones puede ser heredada por el método numérico y en este caso le será más difícil resolver el flujo de manera satisfactoria. Por ejemplo, es conocido que el flujo de capa límite cercano a fronteras cóncavas donde el gradiente de presión es contrario a la dirección radial presenta una inestabilidad conocida como vórtices de Görtler (1941), y si la malla utilizada en la solución numérica de este tipo de problemas no es suficiente para resolver los altos gradientes generados por las inestabilidades, el método no llegará a la convergencia. Sin embargo, diversos estudios numéricos con mallas adecuadas han apoyado al conocimiento de este tipo de estructuras (Benmalek 1993).

El trabajo futuro en relación al código presentado en esta tesis deberá concentrarse en buscar una mejor aproximación de los términos difusivos y convectivos, y en la incorporación de la ecuación de conservación de energía. Mucho se ha hecho en este sentido en las versiones cartesianas de este tipo de códigos y será natural la incorporación de estas modificaciones según la forma de tratar los términos no ortogonales planteada en esta tesis.

Apéndice 1.

Diferenciación covariante.

Sea $T_{ij\dots}^{kl\dots}$ un tensor cartesiano, y sean $T'_{ij\dots}{}^{kl\dots}$ sus nuevas componentes en el sistema de coordenadas curvilíneas (ξ^i) . El número de subíndices indica el orden o rango covariante del tensor, mientras que el número de superíndices indica el orden o rango contravariante del tensor (Aris 1962). Siguiendo la notación introducida en la sección 2.1.1 para la transformación de coordenadas, las componentes de los tensores se transforman de un sistema de coordenadas a otro según la siguiente regla:

$$T'_{ij\dots}{}^{kl\dots} = J_i^p J_j^q \dots \bar{J}_r^k \bar{J}_s^l \dots T_{rs\dots}{}^{pq\dots}$$

Es muy útil expresar las ecuaciones de la física en formas tensoriales covariantes y contravariantes, pues así es fácil transformarlas de un sistema coordenado a otro usando la ley de transformación anterior. En otras palabras, es más natural expresar las leyes de la física de manera que no dependan del sistema coordenado que se escoja. La necesidad de la diferenciación covariante nace del hecho de que la derivada ordinaria $\partial T'_{ij\dots}{}^{kl\dots} / \partial \xi^m$ de un tensor general no se transforma como un tensor covariante. La derivada covariante $\nabla_m T'_{ij\dots}{}^{kl\dots}$ de un tensor está definida de manera que se transforme covariantemente, por lo tanto:

$$\nabla_m T'_{ij\dots}{}^{kl\dots} = J_m^n J_i^p J_j^q \dots \bar{J}_r^k \bar{J}_s^l \dots \partial T_{rs\dots}{}^{pq\dots} / \partial x^n$$

De manera explícita, las derivadas covariantes se definen como sigue:

Escalares. La derivada covariante de un escalar es la misma que su derivada ordinaria,

$$\nabla_i \Phi = \frac{\partial \Phi}{\partial \xi^i}. \quad (\text{A.1.1})$$

Vectores. Las derivadas covariantes de un vector varían si se emplean las componentes covariantes o contravariantes del vector. Estas están dadas por:

$$\nabla_i V'^j = \frac{\partial V'^j}{\partial \xi^i} + \Gamma_{ik}^j V'^k \quad (\text{A.1.2})$$

$$\nabla_i V'_j = \frac{\partial V'_j}{\partial \xi^i} - \Gamma_{ij}^k V'_k \quad (\text{A.1.3})$$

donde Γ_{ij}^k son los coeficientes de conexión, o símbolos de Christoffel, y están dados por:

$$\Gamma_{ij}^k = \bar{J}_i^l \frac{\partial J'_j}{\partial \xi^i} = \frac{1}{2} g^{kl} \left(\frac{\partial g_{jl}}{\partial \xi^i} + \frac{\partial g_{il}}{\partial \xi^j} - \frac{\partial g_{ij}}{\partial \xi^l} \right), \quad (\text{A.1.4})$$

donde g_{ij} es el tensor métrico definido en la sección 2.1.3.

Tensores. Las ecuaciones (A.1.2), (A.1.3) se extienden por la regla de Leibnitz para dar las siguientes operaciones sobre tensores de segundo rango:

$$\nabla_i T'^{jk} = \frac{\partial T'^{jk}}{\partial \xi^i} + \Gamma_{ij}^j T'^{ik} + \Gamma_{ij}^k T'^{jl}$$

$$\nabla_i T'_{jk} = \frac{\partial T'_{jk}}{\partial \xi^i} - \Gamma_{ij}^l T'_{lk} - \Gamma_{ik}^l T'_{jl}$$

$$\nabla_i T'_k{}^j = \frac{\partial T'_k{}^j}{\partial \xi^i} - \Gamma_{ik}^l T'_l{}^j + \Gamma_{il}^j T'_k{}^l$$

y así se sigue para tensores de mayor rango.

Cualquier sistema de ecuaciones diferenciales parciales expresadas en forma tensorial cartesiana covariante, puede ahora ser transformado a un sistema arbitrario de coordenadas curvilíneas como sigue:

- Remplace todas las componentes tensoriales cartesianas por sus componentes transformadas al nuevo sistema coordenado.
- Remplace todas las derivadas por derivadas covariantes.

Ahora se deduce la identidad (2.27) para la divergencia covariante de un vector. Debido a que la divergencia de un vector es un escalar, es la misma en todos los sistemas coordenados, así que se puede escribir, usando (A.1.2):

$$\frac{\partial V^i}{\partial x^i} = \nabla_i V'^i = \frac{\partial V'^i}{\partial \xi^i} + \Gamma_{ik}^k V'^i \quad (\text{A.1.5})$$

donde

$$\Gamma_{ik}^k = \bar{J}_i^k \frac{\partial J^i_k}{\partial \xi^i} = \frac{1}{2} g^{kl} \left(\frac{\partial g_{kl}}{\partial \xi^i} + \frac{\partial g_{il}}{\partial \xi^k} - \frac{\partial g_{ik}}{\partial \xi^l} \right)$$

$$\Rightarrow \Gamma_{ik}^k = \frac{1}{2} g^{kl} \frac{\partial g_{kl}}{\partial \xi^i}.$$

Y sin perder generalidad, se escoge un sistema de referencia local en el que g_{ij} es diagonal, esto es: $g_{ij} = \text{diag}(g_1, g_2, g_3)$. Entonces:

$$\Gamma_{ik}^k = \frac{1}{2} \left(g_1^{-1} \frac{\partial g_1}{\partial \xi^i} + g_2^{-1} \frac{\partial g_2}{\partial \xi^i} - g_3^{-1} \frac{\partial g_3}{\partial \xi^i} \right) = \frac{1}{2g_1 g_2 g_3} \frac{\partial (g_1 g_2 g_3)}{\partial \xi^i}$$

y dado que $g_1 g_2 g_3 = \det(g_{ij}) = g$, se obtiene la siguiente identidad:

$$\Gamma_{ik}^k = \frac{1}{2g} \frac{\partial g}{\partial \xi^i} = \frac{\partial}{\partial \xi^i} \ln \sqrt{g} \quad (\text{A.1.6})$$

Sólo falta sustituir (A.1.6) en (A.1.5) para obtener la identidad para la divergencia covariante:

$$\nabla \cdot \bar{V} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} (\sqrt{g} V'^i) = \frac{1}{\sqrt{g}} \frac{\partial \hat{V}^i}{\partial \xi^i} \quad (\text{A.1.7})$$

Apéndice 2.

Ecuaciones dependientes del tiempo y condiciones de frontera.

- Problema transiente.

La ecuación de Navier-Stokes dependiente del tiempo está dada por:

$$\frac{\partial}{\partial t}(\rho U^k) + \frac{\partial}{\partial x^i}(\rho U^i U^k) = \frac{\partial}{\partial x^i}(\sigma^{ik}) \quad (\text{A.2.1})$$

Esta ecuación sólo incorpora el primer término a la ecuación de momento estacionaria de la sección 2.3.1, y como no depende de las coordenadas curvilíneas, es fácil tratar los términos extras que resultan de su discretización.

Así pues, integrando sobre un volumen de control en el dominio computacional, se llega a la ecuación (2.58) con la adición del término transiente:

$$\left\langle \sqrt{g} \frac{\partial}{\partial t}(\rho U^k) \right\rangle + [\hat{I}_{U^k}^1]_w^e + [\hat{I}_{U^k}^2]_s^n + [\hat{I}_{U^k}^3]_b^l = \langle \sqrt{g} S_{U^k} \rangle \quad (\text{A.2.2})$$

El término transiente es discretizado por diferencias finitas quedando de la siguiente manera:

$$\left\langle \sqrt{g} \frac{\partial}{\partial t}(\rho U^k) \right\rangle_p \approx \text{vol} \Omega_p \left(\frac{\rho_p U_p^k - \rho_p^0 U_p^{k0}}{\Delta t} \right) \quad (\text{A.2.3})$$

Como se ve en el lado derecho de (A.2.3), se tiene un término extra multiplicando a U_p^k que se incorpora al coeficiente a_p en (2.58) y el resto se suma al término fuente, por lo tanto, se tienen las siguientes modificaciones en el caso transiente:

$$a_p \mapsto a_p + a_p^0 \quad (\text{A.2.4})$$

y el término fuente $\text{vol} \Omega_p \langle S_{U^k} \rangle \mapsto \text{vol} \Omega_p \langle S_{U^k} \rangle + a_p^0 U_p^{k0}$ (A.2.5)

$$a_p^0 = \frac{\rho_p^0 \text{vol} \Omega_p}{\Delta t} \quad (\text{A.2.6})$$

El mismo tratamiento se hace para la ecuación general de convección-difusión transiente, reemplazando U^k con Φ . Esto lleva a las modificaciones para la ecuación de continuidad que simplemente se reducen a un término extra en (2.62), que contiene la diferencia temporal de densidades como factor del volumen del elemento, y dividido por el intervalo de tiempo, como se puede obtener de reemplazar U^k en (A.2.3) por uno, que es el caso de la ecuación de continuidad.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

- **Condiciones de frontera.**

Las condiciones de frontera utilizadas en los ejercicios presentados en esta tesis son de tres tipos:

a) Pared sólida:

La velocidad se inicializa a cero en las fronteras sólidas y no se modifica durante todo el proceso de cálculo (caras n,s,t,b).

b) Entrada:

El campo de velocidades en la cara de entrada debe ser dado en el programa. Este campo tampoco es modificado durante el proceso de cálculo. Establece el perfil del fluido en la entrada y la cantidad de masa que atraviesa dicha cara de entrada (cara oeste w).

c) Salida:

El campo de velocidades en el plano de salida (cara este) se copia del plano inmediato anterior para asegurar una condición de flujo desarrollado y se corrige la cantidad de masa que sale para asegurar conservación de masa global y acelerar la convergencia.

Otro tipo de condiciones de frontera donde las velocidades en las paredes no sean distintas de cero necesitarán el cálculo de los esfuerzos cortantes en las paredes respectivas. Las condiciones en las derivadas de las variables podrán ser incorporadas de manera natural si se tiene una malla ortogonal hacia las fronteras, sólo hay que repetir el valor de la variable en el sentido de las coordenadas curvilíneas normales a la pared o a la frontera en cuestión.

Apéndice 3. Método híbrido.

La discretización de los términos difusivos y convectivos que da como resultado las fórmulas (2.49) se hace a través del método híbrido. Para explicarlo, se simplifica el problema a un caso unidimensional, estacionario y sin términos fuente. Con esto se pierde precisión pero se gana una manera sencilla de entenderlo.

El problema unidimensional, estacionario y sin términos fuente está gobernado por las ecuaciones:

$$\frac{d}{dx}(\rho U \Phi) = \frac{d}{dx} \left(\Gamma \frac{d\Phi}{dx} \right) \quad (\text{A.3.1})$$

y

$$\frac{d}{dx}(\rho U) = 0 \Rightarrow \rho U = cte. \quad (\text{A.3.2})$$

Siguiendo el resultado (A.3.2) y suponiendo que Γ es constante (por ejemplo la viscosidad), se puede obtener una solución exacta a la ecuación (A.3.1).

Sea el dominio $0 \leq x \leq L$

y las condiciones de frontera $\text{en } x = 0 \quad \Phi = \Phi_0$
 $\text{en } x = L \quad \Phi = \Phi_L$

$$\Rightarrow \frac{\Phi - \Phi_0}{\Phi_L - \Phi_0} = \frac{e^{\frac{Px}{L}} - 1}{e^P - 1} \quad (\text{A.3.3})$$

donde P es un número de Peclet definido como:

$$P = \frac{\rho U L}{\Gamma}$$

que expresa la razón entre los coeficientes convectivos y difusivos, $C = \rho U$ y $D = \Gamma / \Delta x$.

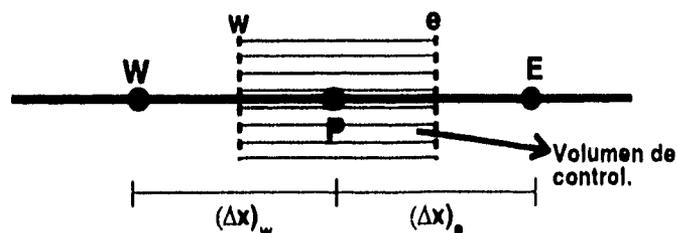


Fig A.3.1. Volumen de control unidimensional.

Al aproximar la derivada en (A.3.1) por diferencias finitas alrededor del punto P según la Fig. A.3.1, se obtiene:

$$\left(\rho U \Phi - \Gamma \frac{d\Phi}{dx}\right)_e - \left(\rho U \Phi - \Gamma \frac{d\Phi}{dx}\right)_w = 0 \quad (\text{A.3.4})$$

Usando a la solución exacta (A.3.3) como perfil entre los puntos P y E, con Φ_p y Φ_E reemplazando a Φ_0 y Φ_L , y $(\Delta x)_e$ reemplazando a L, se obtiene:

$$\begin{aligned} \left(\rho U \Phi - \Gamma \frac{d\Phi}{dx}\right)_e &= (\rho U)_e \left(\Phi_p + \frac{\Phi_p - \Phi_E}{e^{Pe} - 1}\right) \\ Pe &= \frac{(\rho U)_e (\Delta x)_e}{\Gamma_e} = \frac{C_e}{D_e} \end{aligned} \quad (\text{A.3.5})$$

Haciendo lo mismo para el intervalo entre los puntos W y P, se obtiene de (A.3.4):

$$(\rho U)_e \left(\Phi_p + \frac{\Phi_p - \Phi_E}{e^{Pe} - 1}\right) - (\rho U)_w \left(\Phi_w + \frac{\Phi_w - \Phi_p}{e^{Pw} - 1}\right) = 0$$

y al ponerlo en la forma:

$$a_p \Phi_p = a_E \Phi_E + a_w \Phi_w \quad (\text{A.3.6})$$

se tiene que

$$a_E = \frac{C_e}{e^{Pe} - 1} \quad (\text{A.3.7})$$

$$a_w = \frac{C_w e^{Pw}}{e^{Pw} - 1} \quad (\text{A.3.8})$$

$$a_p = a_E + a_w + C_e - C_w \quad (\text{A.3.9})$$

A pesar de tener un comportamiento conveniente, este enfoque no se ha implementado en el programa debido a que las exponenciales consumen mucho tiempo de cómputo, y de cualquier forma no es exacto para el caso tridimensional y con términos fuente. Es más eficiente buscar fórmulas simples que representen un comportamiento parecido.

El esquema híbrido (Spalding 1972), (Patankar 1980), obtiene fórmulas sencillas para los coeficientes en la ecuación (A.3.6), siguiendo el comportamiento asintótico y central de los coeficientes de forma exponencial (A.3.7) y (A.3.8).

$$\frac{a_E}{D_e} = \frac{Pe}{e^{Pe} - 1} \Rightarrow \begin{cases} \frac{a_E}{D_e} \rightarrow 0 & \text{si } Pe \rightarrow \infty \\ \frac{a_E}{D_e} \rightarrow -Pe & \text{si } Pe \rightarrow -\infty \\ \frac{a_E}{D_e} \rightarrow 1 - \frac{Pe}{2} & \text{si } Pe = 0 \end{cases} \quad (\text{A.3.10})$$

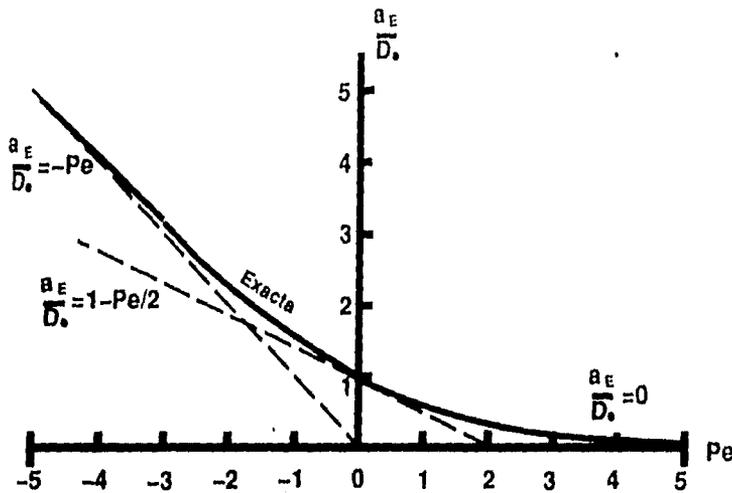


Fig. A.3.2. Variación del coeficiente a_E con el número de Peclet.

El método híbrido, aprovecha estos tres comportamientos y aproxima los tres intervalos por líneas rectas como lo muestra la figura A.3.2.

$$\begin{aligned}
 Pe < -2 & \mapsto \frac{a_E}{D_e} = -Pe \\
 -2 \leq Pe \leq 2 & \mapsto \frac{a_E}{D_e} = 1 - \frac{Pe}{2} \\
 Pe > 2 & \mapsto \frac{a_E}{D_e} = 0
 \end{aligned} \tag{A.3.11}$$

por lo tanto

$$a_E = \max\left(\frac{1}{2}|C_e|, D_e\right) - \frac{1}{2}C_e \tag{A.3.12}$$

Si siguiendo esta misma idea para el coeficiente a_w , se obtiene la discretización de convección-difusión a través del esquema híbrido:

$$a_P \Phi_P = a_E \Phi_E + a_W \Phi_W \tag{A.3.13}$$

$$a_E = \max\left(\frac{1}{2}|C_e|, D_e\right) - \frac{1}{2}C_e \tag{A.3.14}$$

$$a_W = \max\left(\frac{1}{2}|C_w|, D_w\right) + \frac{1}{2}C_w \tag{A.3.15}$$

$$a_P = a_E + a_W + C_e - C_w \tag{A.3.16}$$

Se le ha dado el nombre de método híbrido porque en sus tres partes corresponde a una combinación entre la aproximación de diferencias centradas y la de aguas-arriba (*upwind*), pero es mucho más claro considerarlo como una aproximación de tres rectas a la curva exacta como se muestra en la figura A.3.2.

Apéndice 4.

Listado del programa NECRONOM.f.

A.4.1 Programa principal.

```

1 C
2 C
3 C
4 C
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C

```

```

NECRONOM.f

```

```

A COMPUTER PROGRAM TO SOLVE THREE DIMENSIONAL LAMINAR INCOMPRESSIBLE
FLUID FLOW IN COMPLEX GEOMETRIES

```

```

CODIGO EN FORTRAN 77 PARA RESOLVER, MEDIANTE LA ESTRATEGIA DE
VOLUMEN FINITO, FLUJOS EN TRES DIMENSIONES LAMINARES E
INCOMPRESIBLES EN GEOMETRIAS CON COORDENADAS CURVILINEAS

```

```

BY JULIAN T. BECERRA SAGREDO
LABORATORIO DE ENERGIA SOLAR, U.N.A.M.

```

```

INCLUDE 'param.f'

```

```

CHARACTER*12 HEDU,HEDV,HEDW,HEDP,HEDM,HEDEN
DATA HEDU /'u-vel.'/
DATA HEDV /'v-vel.'/
DATA HEDW /'w-vel.'/
DATA HEDP /'pressure'/
DATA HEDM /'vis. ef.'/
DATA HEDEN /'density'/

```

```

VARIABLE STORAGE IN MEMORY

```

```

LOCACIONES DE MEMORIA PARA LAS VARIABLES

```

```

COMMON
1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
2 VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3),
3 AREAWS(NX,NY,NZ),AREAWB(NX,NY,NZ),AREASB(NX,NY,NZ),
4 AREAET(NX,NY,NZ),AREAEN(NX,NY,NZ),AREANT(NX,NY,NZ)
1/UVEL/RESORU,NSWPU,URFU
1/VVEL/RESORV,NSWPV,URFV
1/WVEL/RESORW,NSWPW,URFW
1/PCOR/RESORM,NSWPP,URFP,
1 IPREF,JPREF,KPREF
1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
2 W(NX,NY,NZ)
3/VARRC/UFLUX(NX,NY,NZ,3)
1/ALL/IT,JT,KT,NI,NJ,NK,NIM1,NJM1,NKM1,NIM2,NJM2,NKM2,GREAT
1/FRONT/YW(NY,NZ),YE(NY,NZ),YS(NX,NZ),YN(NX,NZ),YB(NX,NY),YT(NX,NY)
COMMON
1/FLUPR/VISCOS,DENSIT,DEN(NX,NY,NZ)
2 VIS(NX,NY,NZ)
1/KASE T1/UIN,FLOWIN
1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
1 AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
2 SP(NX,NY,NZ)
1/TRANS/DENO(NX,NY,NZ),APO(NX,NY,NZ)
1/VARTR/UO(NX,NY,NZ),VO(NX,NY,NZ),WO(NX,NY,NZ)
1/TIMEINC/DTIME
1/FLO/FLOW
1/ANALITICA/UALIT(3,NX,NY,NZ)
LOGICAL INCALU,INCALV,INCALW,INCALP,INPRO,INPROVAR,
1 INTRANS,INCALMPGS,IGEOMCC,IGEOMGC

```

```

READ AND WRITE FILES

```

```

69 C   ARCHIVOS DE ENTRADA Y SALIDA
70 C
71     OPEN(UNIT=1, FILE='MALLA3D.DAT')
72     OPEN(UNIT=7, FILE='channel.cut')
73     OPEN(UNIT=12, FILE='MALLAXYZ.OUT')
74     OPEN(UNIT=2, FILE='u-analit.ens')
75 C
76 C
77     WRITE(6,*) 'COMIENZO'
78 C
79 C   PARAMETERS AND CONTROL INDICES
80 C
81 C   PARAMETROS E INDICES DE CONTROL
82 C
83 C-----
84     GREAT=1.E30
85     NITER=0
86     IT=NX
87     JT=NY
88     KT=NZ
89     NSWPU=9
90     NSWPU=9
91     NSWPU=9
92     NSWPU=9
93 C-----
94 C----GRID
95 C----MALLA
96 C
97     NI=NX
98     NJ=NY
99     NK=NZ
100    NIM1=NI-1
101    NJM1=NJ-1
102    NKM1=NK-1
103    NIM2=NI-2
104    NJM2=NJ-2
105    NKM2=NK-2
106 C-----
107 C----DELTA TIME
108 C----PASO EN EL TIEMPO
109     DTIME=2.
110     NTSTEP=20
111 C-----
112 C----VARIABLE SELECTION
113 C----SELECCION DE VARIABLES
114     INCALU=.TRUE.
115     INCALV=.TRUE.
116     INCALW=.TRUE.
117     INCALP=.TRUE.
118     INPRO=.TRUE.
119     INPROVAR=.TRUE.
120     INTRANS=.TRUE.
121     INCALMPGS = .TRUE.
122     IGEOMCC=.TRUE.
123     IGEOMGC=.TRUE.
124 C-----
125 C----FLUID PROPERTIES
126 C----PROPIEDADES DEL FLUIDO
127     DENSIT =1000.0
128     VISCOS=0.0008
129 C-----
130 C----BOUNDARY VALUES
131 C----CONSTANTES DE FRONTERA
132     UIN=0.001
133 C-----
134 C----PRESSURE REFERENCE COORDINATES
135 C----COORDENADAS DE PRESION DE REFERENCIA
136     IPREF=2
137     JPREF=2
138     KPREF=2
139 C-----
140 C----PROGRAM CONTROL AND MONITOR
141 C----CONTROL Y MONITOREO DEL PROGRAMA
142     MAXIT=10000
143     IMON=5
144     JMON=5
145     KMON=5
146     URFU=0.5
147     URFV=0.5
148     URFW=0.5
149     URFP=0.1
150     INDPRI =10000

```

```

151         INDEFGR=1
152         SORMAX =.900E-06
153 C
154 C
155 C         INITIAL OPERATIONS
156 C
157 C         OPERACIONES INICIALES
158 C-----
159 C-----SET VARIABLES TO ZERO
160 C-----INICIALIZACION A CERO DE LAS VARIAELES
161 C
162         IF(IGEOMCC) CALL INIT
163 C-----
164 C-----GEOMETRY QUANTITIES
165 C-----CANTIDADES GEOMETRICAS
166 C
167         IF(IGEOMGC) CALL GEOMGC
168 C-----
169 C-----INITIAL FIELDS
170 C-----CAMPOS INICIALES
171 C
172         READ(2,*)
173         READ(2,951) (((UALIT(IJK, I, J, K), IJK=1, 3), I=1, NIM1), J=1, NJM1)
174         * , K=1, NKM1)
175     951  FORMAT(6E12.5)
176         DO 202 I=1, NIM1
177         DO 202 J=1, NJM1
178         DO 202 K=1, NKM1
179         U(I, J, K)=UALIT(1, NIM2, J, K)*AREA(I, J, K, 1, 1)/SQRT(GAREA(I, J, K, 1))
180         V(I, J, K)=UALIT(1, NIM2, J, K)*AREA(I, J, K, 1, 2)/SQRT(GAREA(I, J, K, 1))
181 C         U(I, J, K)=UIN
182 C         V(I, J, K)=0.0
183     202  CONTINUE
184         IF(INTRANS) CALL INITCOND
185         DO 201 I=1, NI
186         DO 201 J=1, NJ
187         DO 201 K=1, NK
188         DEN(I, J, K)=DENSIT
189     201  CONTINUE
190 C
191 C-----
192 C-----VELOCITY BOUNDARY CONDITIONS
193 C-----CONDICIONES DE FRONTERA PARA LA VELOCIDAD
194 C-----NORTH WALL
195         DO 207 I=1, NIM1
196         DO 207 K=1, NKM1
197         U(I, NJM1, K)=0.
198         V(I, NJM1, K)=0.
199         W(I, NJM1, K)=0.
200 C-----SOUTH WALL
201         U(I, 1, K)=0.
202         V(I, 1, K)=0.
203         W(I, 1, K)=0.
204     207  CONTINUE
205 C-----WEST WALL
206         DO 215 J=2, NJM2
207         DO 215 K=2, NKM2
208         U(1, J, K)=UALIT(1, NIM2, J, K)*AREA(1, J, K, 1, 1)/SQRT(GAREA(1, J, K, 1))
209         V(1, J, K)=UALIT(1, NIM2, J, K)*AREA(1, J, K, 1, 2)/SQRT(GAREA(1, J, K, 1))
210 C         U(1, J, K)=UIN
211 C         V(1, J, K)=0.0
212         W(1, J, K)=0.
213 C-----EAST WALL
214         V(NIM1, J, K)=0.
215         W(NIM1, J, K)=0.
216     215  CONTINUE
217 C-----BOTTOM WALL
218         DO 218 I=1, NIM1
219         DO 218 J=1, NJM1
220         U(I, J, 1)=0.
221         V(I, J, 1)=0.
222         W(I, J, 1)=0.
223 C-----TOP WALL
224         U(I, J, NKM1)=0.
225         V(I, J, NKM1)=0.
226         W(I, J, NKM1)=0.
227     218  CONTINUE
228 C-----
229 C-----INITIAL OUTPUT
230 C-----SALIDA INICIAL
231         IF(INCALMPGS) CALL MPGSOUT(0, 1, 1, 1, NI, NJ, NK, U, V, W, P, 'PRES')
232         WRITE(7, 210)

```

```

233      WRITE(7,211)
234      WRITE(7,260) DENSIT
235      WRITE(7,250) VISCOS
236      IF(INCALU) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,XU,Y,Z,U,HEDU)
237      IF(INCALV) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,X,YV,Z,V,HEDV)
238      IF(INCALW) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,X,Y,ZW,W,HEDW)
239      C      IF(INCALP) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,X,Y,Z,P,HEDP)
240      C      IF(INPROVAR) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,X,Y,Z,DEN,HEDEN)
241      C
242      C-----
243      C-----CALCULATE RESIDUAL SOURCES AND NORMALIZATION FACTORS
244      C-----CALCULO DE FUENTES RESIDUALES Y FACTORES DE NORMALIZACION
245      FLOWIN=0.
246      XMONIN=0.
247      I = 1
248      DO K=2,NKM2
249      DO J=2,NJM2
250      FLOWIN=FLOWIN+DEN(I,J,K)*(AREA(I,J,K,1,1)*U(I,J,K)
251      * +AREA(I,J,K,1,2)*V(I,J,K) + AREA(I,J,K,1,3)*W(I,J,K))
252      ENDDO
253      ENDDO
254      WRITE(7,440) FLOWIN
255      WRITE(7,445) FLOW
256      C-----
257      C-----INITIAL CALCULATION OF VELOCITY IN ELEMENT FACES
258      C-----CALCULO INICIAL DE LA VELOCIDAD EN LAS CARAS DE LOS ELEMENTOS
259      CALL RHICHHW(2,2,2,NIM1,NJM2,NKM2)
260      CALL RHICHS(2,2,2,NIM2,NJM1,NKM2)
261      CALL RHICHWB(2,2,2,NIM2,NJM2,NKM1)
262      C-----
263      C
264      C      TRANSIENT ANALYSIS
265      C
266      C      ANALISIS TRANSIENTE
267      C
268      DO 400 ITT=1,NTSTEP
269      NITER=0
270      C
271      C
272      C      ITERATION LOOP
273      C
274      C      CICLO ITERATIVO (SIMPLE)
275      C
276      TTIME=DTIME*FLOAT(ITT)
277      WRITE(7,310) IMON,JMON,KMON
278      300 NITER=NITER+1
279      WRITE(6,450)RESORM
280      CALL CALCOEF(2,2,2,NIM2,NJM2,NKM2)
281      CALL FUENTE(2,2,2,NIM2,NJM2,NKM2)
282      CALL RHIECHOWW(3,2,2,NIM2,NJM2,NKM2)
283      CALL RHIECHOWS(2,3,2,NIM2,NJM2,NKM2)
284      CALL RHIECHOWB(2,2,3,NIM2,NJM2,NKM2)
285      CALL RHICHHW(2,2,2,NJM2,NKM2)
286      CALL RHICHS(2,2,2,NIM2,2,NKM2)
287      CALL RHICHWB(2,2,2,NIM2,NJM2,2)
288      CALL RHICHWE(NIM1,2,2,NIM1,NJM2,NKM2)
289      CALL RHICHWN(2,NJM1,2,NIM2,NJM1,NKM2)
290      CALL RHICHTW(2,2,NKM1,NIM2,NJM2,NKM1)
291      CALL CALP(2,2,2,NIM2,NJM2,NKM2)
292      C
293      C-----INTERMEDIATE OUTPUT
294      C-----SALIDA DE CADA ITERACION
295      RESORM =RESORM/1.
296      RESORU =RESORU/1.
297      RESORV =RESORV/1.
298      RESORW =RESORW/1.
299      WRITE(7,311) NITER,RESORU,RESORV,RESORW,RESORM,
300      1U(IMON,JMON,KMON),V(IMON,JMON,KMON),W(IMON,JMON,KMON)
301      1,P(IMON,JMON,KMON)
302      1,TTIME
303      IF(MOD(NITER,INDPRI).NE.0) GO TO 301
304      C      IF(INCALU) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,XU,Y,Z,U,HEDU)
305      C      IF(INCALV) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,X,YV,Z,V,HEDV)
306      C      IF(INCALW) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,X,Y,ZW,W,HEDW)
307      C      IF(INCALP) CALL PRINT(1,1,1,NI,NJ,NK,IT,JT,KT,X,Y,Z,P,HEDP)
308      WRITE(7,310) IMON,JMON,KMON
309      301 CONTINUE
310      C
311      C-----TERMINATION TESTS
312      C-----EXAMEN DE CONVERGENCIA
313      C
314      SORCE=AMAX1 (RESORM,RESORU,RESORV,RESORW)

```



```

28      1/VARRC/UFLUX(NX,NY,NZ,3)
29      C
30      C
31      C      -----
32      C      SET VARIABLES TO ZERO
33      C      -----
34      C      INICIALIZAR LAS VARIABLES A CERO
35      RESORU=0.
36      RESORV=0.
37      RESORW=0.
38      RESORM=0.
39      DO 200 I=1,N1
40      DO 200 J=1,NJ
41      DO 200 K=1,NK
42      UFLUX(I,J,K,1)=0.0
43      UFLUX(I,J,K,2)=0.0
44      UFLUX(I,J,K,3)=0.0
45      U(I,J,K)=0.0
46      V(I,J,K)=0.0
47      W(I,J,K)=0.0
48      P(I,J,K)=0.0
49      PP(I,J,K)=0.0
50      DEN(I,J,K)=0.0
51      VIS(I,J,K)=VISCOS
52      SU(I,J,K,1)=0.0
53      SU(I,J,K,2)=0.0
54      SU(I,J,K,3)=0.0
55      SP(I,J,K)=0.0
56      Uo(I,J,K)=0.0
57      VO(I,J,K)=0.0
58      WO(I,J,K)=0.0
59      DENO(I,J,K)=0.0
60      AP(I,J,K,1)=0.0
61      AP(I,J,K,2)=0.0
62      APO(I,J,K)=0.0
63      AW(I,J,K)=0.0
64      AE(I,J,K)=0.0
65      AS(I,J,K)=0.0
66      AN(I,J,K)=0.0
67      AB(I,J,K)=0.0
68      AT(I,J,K)=0.0
69      200 CONTINUE
70      RETURN
71      END

```

A.4.3 Subrutina GEOMGC.

```

1      C-----
2      SUBROUTINE GEOMGC
3      C
4      C      GEOMETRIC FACTORS COMPUTATION
5      C
6      C      EN ESTE PROGRAMA SE CALCULAN LOS CENTROIDES, LOS VECTORES
7      C      DE AREA, LOS FACTORES DE PESO DE INTERPOLACION LINEAL
8      C      Y EL VOLUMEN DE LOS ELEMENTOS DE CONTROL PARA UNA
9      C      MALLA DADA, TRIDIMENSIONAL Y TOPOLOGICAMENTE CARTESIANA NO
10     C      NECESARIAMENTE REGULAR.
11     C
12     C
13     C      INCLUDE 'param.f'
14     C      COMMON
15     C      1/ALL/IT, JT, KT, NI, NJ, NK, NIM1, NJM1, NKM1, NIM2, NJM2, NKM2, GREAT
16     C      1/GGC/AREA(NX,NY,NZ,3,3), GAREA(NX,NY,NZ,3),
17     C      2/VOL(NX,NY,NZ), WFACT(NX,NY,NZ,3),
18     C      3/AREAWS(NX,NY,NZ), AREAWB(NX,NY,NZ), AREASB(NX,NY,NZ),
19     C      4/AREAET(NX,NY,NZ), AREAEN(NX,NY,NZ), AREANT(NX,NY,NZ)
20     C      1/FRONT/YW(NY,NZ), YE(NY,NZ), YS(NX,NZ), YN(NX,NZ), YB(NX,NY), YT(NX,NY)
21     C      DIMENSION XC(NX,NY,NZ), YC(NX,NY,NZ), ZC(NX,NY,NZ),
22     C      *      X(NX,NY,NZ), Y(NX,NY,NZ), Z(NX,NY,NZ),
23     C      *      X1(NX,NY,NZ), Y1(NX,NY,NZ), Z1(NX,NY,NZ),
24     C      *      X2(NX,NY,NZ), Y2(NX,NY,NZ), Z2(NX,NY,NZ),
25     C      *      X3(NX,NY,NZ), Y3(NX,NY,NZ), Z3(NX,NY,NZ)
26     C
27     C
28     C      CORNER COORDINATES GRID
29     C
30     C      -----
31     C      LECTURA DE LA MALLA HECHA CON LAS COORDENADAS DE LAS ESQUINAS
32     C      DE LOS ELEMENTOS
33     C
34     C      READ(1,1)

```

```

34 1 FORMAT(////)
35 DO 5 K=1,NK
36 DO 4 J=1,NJ
37 DO 3 I=1,NI
38 READ(1,2)J1LZ,XC(I,J,K),YC(I,J,K),ZC(I,J,K)
39 2 FORMAT(18,3E12.5)
40 3 CONTINUE
41 4 CONTINUE
42 5 CONTINUE
43 C
44 C
45 C CENTROIDS OF VOLUME AND FACES
46 C
47 C CALCULO DE LOS CENTROIDES
48 C
49 DO 9 K=1,NKM1
50 DO 8 J=1,NJM1
51 DO 7 I=1,NIM1
52 X(I,J,K) = 0.125*(XC(I,J,K)+XC(I+1,J,K)+XC(I,J+1,K)
53 * +XC(I,J,K+1)+XC(I+1,J+1,K)+XC(I+1,J,K+1)
54 * +XC(I,J+1,K+1)+XC(I+1,J+1,K+1))
55 Y(I,J,K) = 0.125*(YC(I,J,K)+YC(I+1,J,K)+YC(I,J+1,K)
56 * +YC(I,J,K+1)+YC(I+1,J+1,K)+YC(I+1,J,K+1)
57 * +YC(I,J+1,K+1)+YC(I+1,J+1,K+1))
58 Z(I,J,K) = 0.125*(ZC(I,J,K)+ZC(I+1,J,K)+ZC(I,J+1,K)
59 * +ZC(I,J,K+1)+ZC(I+1,J+1,K)+ZC(I+1,J,K+1)
60 * +ZC(I,J+1,K+1)+ZC(I+1,J+1,K+1))
61 C
62 X1(I,J,K) = 0.25*(XC(I,J,K)+XC(I,J+1,K)+XC(I,J,K+1)+XC(I,J+1,K+1))
63 Y1(I,J,K) = 0.25*(YC(I,J,K)+YC(I,J+1,K)+YC(I,J,K+1)+YC(I,J+1,K+1))
64 Z1(I,J,K) = 0.25*(ZC(I,J,K)+ZC(I,J+1,K)+ZC(I,J,K+1)+ZC(I,J+1,K+1))
65 X2(I,J,K) = 0.25*(XC(I,J,K)+XC(I+1,J,K)+XC(I,J,K+1)+XC(I+1,J,K+1))
66 Y2(I,J,K) = 0.25*(YC(I,J,K)+YC(I+1,J,K)+YC(I,J,K+1)+YC(I+1,J,K+1))
67 Z2(I,J,K) = 0.25*(ZC(I,J,K)+ZC(I+1,J,K)+ZC(I,J,K+1)+ZC(I+1,J,K+1))
68 X3(I,J,K) = 0.25*(XC(I,J,K)+XC(I+1,J,K)+XC(I,J+1,K)+XC(I+1,J+1,K))
69 Y3(I,J,K) = 0.25*(YC(I,J,K)+YC(I+1,J,K)+YC(I,J+1,K)+YC(I+1,J+1,K))
70 Z3(I,J,K) = 0.25*(ZC(I,J,K)+ZC(I+1,J,K)+ZC(I,J+1,K)+ZC(I+1,J+1,K))
71 C
72 7 CONTINUE
73 8 CONTINUE
74 9 CONTINUE
75 C
76 C
77 C BOUNDARY DISTANCE WALL-CENTROID
78 C
79 C DISTANCIAS ENTRE PUNTOS FRONTERA Y PLANO FRONTERA
80 C
81 C WEST WALL
82 C
83 C PLANO OESTE
84 C
85 I=2
86 DO 198 K=1,NKM1
87 DO 198 J=1,NJM1
88 YW(J,K)= SQRT((X1(I,J,K)-X(I,J,K))**2.+(Y1(I,J,K)-Y(I,J,K))**2.
89 * +(Z1(I,J,K)-Z(I,J,K))**2.)
90 198 CONTINUE
91 C
92 C
93 C EAST WALL
94 C
95 C PLANO ESTE
96 C
97 I=NIM1
98 DO 199 K=1,NKM1
99 DO 199 J=1,NJM1
100 YE(J,K)= SQRT((X1(I,J,K)-X(I-1,J,K))**2.+(Y1(I,J,K)-Y(I-1,J,K))**2.
101 * +(Z1(I,J,K)-Z(I-1,J,K))**2.)
102 199 CONTINUE
103 C
104 C SOUTH WALL
105 C
106 C PLANO SUR
107 C
108 J=2
109 DO 200 K=1,NKM1
110 DO 200 I=1,NIM1
111 YS(I,K)= SQRT((X2(I,J,K)-X(I,J,K))**2.+(Y2(I,J,K)-Y(I,J,K))**2.
112 * +(Z2(I,J,K)-Z(I,J,K))**2.)
113 200 CONTINUE
114 C
115 C NORTH WALL

```

```

116 C
117 C PLANO NORTE
118 C
119 J=NJM1
120 DO 201 K=1,NKM1
121 DO 201 I=1,NIM1
122 YN(I,K) = SQRT((X2(I,J,K)-X(I,J-1,K))**2.+(Y2(I,J,K)-Y(I,J-1,K))**2.
123 * +(Z2(I,J,K)-Z(I,J-1,K))**2.)
124 201 CONTINUE
125 C
126 C BOTTON WALL
127 C
128 C PLANO BAJO
129 C
130 K=2
131 DO 202 J=1,NJM1
132 DO 202 I=1,NIM1
133 YB(I,J) = SQRT((X3(I,J,K)-X(I,J,K))**2.+(Y3(I,J,K)-Y(I,J,K))**2.
134 * +(Z3(I,J,K)-Z(I,J,K))**2.)
135 202 CONTINUE
136 C
137 C TOP WALL
138 C
139 C PLANO TOPE
140 C
141 K=NKM1
142 DO 203 J=1,NJM1
143 DO 203 I=1,NIM1
144 YT(I,J) = SQRT((X3(I,J,K)-X(I,J,K-1))**2.+(Y3(I,J,K)-Y(I,J,K-1))**2.
145 * +(Z3(I,J,K)-Z(I,J,K-1))**2.)
146 203 CONTINUE
147 C
148 C
149 C LINEAR INTERPOLATION WEIGHTS
150 C
151 C CALCULO DE FACTORES DE PESO DE INTERPOLACION LINEAL
152 C
153 DO 12 K=2,NKM1
154 DO 11 J=2,NJM1
155 DO 10 I=2,NIM1
156 PW = SQRT((X1(I,J,K)-X(I,J,K))**2.+(Y1(I,J,K)-Y(I,J,K))**2.
157 * +(Z1(I,J,K)-Z(I,J,K))**2.)
158 WW = SQRT((X1(I,J,K)-X(I-1,J,K))**2.+(Y1(I,J,K)-Y(I-1,J,K))**2.
159 * +(Z1(I,J,K)-Z(I-1,J,K))**2.)
160 PS = SQRT((X2(I,J,K)-X(I,J,K))**2.+(Y2(I,J,K)-Y(I,J,K))**2.
161 * +(Z2(I,J,K)-Z(I,J,K))**2.)
162 SS = SQRT((X2(I,J,K)-X(I,J-1,K))**2.+(Y2(I,J,K)-Y(I,J-1,K))**2.
163 * +(Z2(I,J,K)-Z(I,J-1,K))**2.)
164 PB = SQRT((X3(I,J,K)-X(I,J,K))**2.+(Y3(I,J,K)-Y(I,J,K))**2.
165 * +(Z3(I,J,K)-Z(I,J,K))**2.)
166 BB = SQRT((X3(I,J,K)-X(I,J,K-1))**2.+(Y3(I,J,K)-Y(I,J,K-1))**2.
167 * +(Z3(I,J,K)-Z(I,J,K-1))**2.)
168 C
169 WFACT(I,J,K,1) = PW/(PW+WW)
170 WFACT(I,J,K,2) = PS/(PS+SS)
171 WFACT(I,J,K,3) = PB/(PB+BB)
172 C
173 10 CONTINUE
174 11 CONTINUE
175 12 CONTINUE
176 C
177 C
178 C AREA VECTORS
179 C
180 C CALCULO DE LOS VECTORES DE AREA DE CADA CARA SEGUN EL
181 C SENTIDO POSITIVO DE LAS COORDENADAS CURVILINEAS
182 C
183 DO 15 K=1,NKM1
184 DO 14 J=1,NJM1
185 DO 13 I=1,NIM1
186 C
187 DIAGW1 = XC(I,J+1,K+1) - XC(I,J,K)
188 DIAGW2 = YC(I,J+1,K+1) - YC(I,J,K)
189 DIAGW3 = ZC(I,J+1,K+1) - ZC(I,J,K)
190 DIAGS1 = XC(I+1,J,K+1) - XC(I,J,K)
191 DIAGS2 = YC(I+1,J,K+1) - YC(I,J,K)
192 DIAGS3 = ZC(I+1,J,K+1) - ZC(I,J,K)
193 DIAGB1 = XC(I+1,J+1,K) - XC(I,J,K)
194 DIAGB2 = YC(I+1,J+1,K) - YC(I,J,K)
195 DIAGB3 = ZC(I+1,J+1,K) - ZC(I,J,K)
196 GDIAW1 = XC(I,J+1,K) - XC(I,J,K+1)
197 GDIAW2 = YC(I,J+1,K) - YC(I,J,K+1)

```

```

198 GDIAW3 = ZC(I,J+1,K) - ZC(I,J,K+1)
199 GDIAS1 = XC(I,J,K+1) - XC(I+1,J,K)
200 GDIAS2 = YC(I,J,K+1) - YC(I+1,J,K)
201 GDIAS3 = ZC(I,J,K+1) - ZC(I+1,J,K)
202 GDIAB1 = XC(I+1,J,K) - XC(I,J+1,K)
203 GDIAB2 = YC(I+1,J,K) - YC(I,J+1,K)
204 GDIAB3 = ZC(I+1,J,K) - ZC(I,J+1,K)
205 C
206 AREA(I,J,K,1,1) = 0.5*(DIAGW3*GDIAW2 - DIAGW2*GDIAW3)
207 AREA(I,J,K,1,2) = 0.5*(DIAGW1*GDIAW3 - DIAGW3*GDIAW1)
208 AREA(I,J,K,1,3) = 0.5*(DIAGW2*GDIAW1 - DIAGW1*GDIAW2)
209 AREA(I,J,K,2,1) = 0.5*(DIAGS3*GDIAS2 - DIAGS2*GDIAS3)
210 AREA(I,J,K,2,2) = 0.5*(DIAGS1*GDIAS3 - DIAGS3*GDIAS1)
211 AREA(I,J,K,2,3) = 0.5*(DIAGS2*GDIAS1 - DIAGS1*GDIAS2)
212 AREA(I,J,K,3,1) = 0.5*(DIAGB3*GDIAB2 - DIAGB2*GDIAB3)
213 AREA(I,J,K,3,2) = 0.5*(DIAGB1*GDIAB3 - DIAGB3*GDIAB1)
214 AREA(I,J,K,3,3) = 0.5*(DIAGB2*GDIAB1 - DIAGB1*GDIAB2)
215 C
216 13 CONTINUE
217 14 CONTINUE
218 15 CONTINUE
219 C
220 C
221 C BOUNDARY CALCULATIONS OF AREA, WEIGHTS AND CENTROIDS
222 C
223 C CALCULO DE CENTROIDES, AREAS Y FACTORES DE INTERPOLACION
224 C PARA LAS CARAS EXTERNAS DE LOS ULTIMOS VOLUMENES DE
225 C CONTROL
226 C
227 C
228 C CARA ESTE
229 C
230 DO 100 K=1,NKM1
231 DO 99 J=1,NJM1
232 C
233 I=NI
234 C
235 X1(I,J,K) = 0.25*(XC(I,J,K)+XC(I,J+1,K)+XC(I,J,K+1)+XC(I,J+1,K+1))
236 Y1(I,J,K) = 0.25*(YC(I,J,K)+YC(I,J+1,K)+YC(I,J,K+1)+YC(I,J+1,K+1))
237 Z1(I,J,K) = 0.25*(ZC(I,J,K)+ZC(I,J+1,K)+ZC(I,J,K+1)+ZC(I,J+1,K+1))
238 C
239 PW = SQRT((X1(I,J,K)-X(I,J,K))**2.+(Y1(I,J,K)-Y(I,J,K))**2.
240 * +(Z1(I,J,K)-Z(I,J,K))**2.)
241 WW = SQRT((X1(I,J,K)-X(I-1,J,K))**2.+(Y1(I,J,K)-Y(I-1,J,K))**2.
242 * +(Z1(I,J,K)-Z(I-1,J,K))**2.)
243 WFACT(I,J,K,1) = PW/(PW+WW)
244 C
245 DIAGW1 = XC(I,J+1,K+1) - XC(I,J,K)
246 DIAGW2 = YC(I,J+1,K+1) - YC(I,J,K)
247 DIAGW3 = ZC(I,J+1,K+1) - ZC(I,J,K)
248 GDIAW1 = XC(I,J+1,K) - XC(I,J,K+1)
249 GDIAW2 = YC(I,J+1,K) - YC(I,J,K+1)
250 GDIAW3 = ZC(I,J+1,K) - ZC(I,J,K+1)
251 AREA(I,J,K,1,1) = 0.5*(DIAGW3*GDIAW2 - DIAGW2*GDIAW3)
252 AREA(I,J,K,1,2) = 0.5*(DIAGW1*GDIAW3 - DIAGW3*GDIAW1)
253 AREA(I,J,K,1,3) = 0.5*(DIAGW2*GDIAW1 - DIAGW1*GDIAW2)
254 GAREA(I,J,K,1) = AREA(I,J,K,1,1)*AREA(I,J,K,1,1)
255 * +AREA(I,J,K,1,2)*AREA(I,J,K,1,2)
256 * +AREA(I,J,K,1,3)*AREA(I,J,K,1,3)
257 C
258 99 CONTINUE
259 100 CONTINUE
260 C
261 C
262 C NORTH WALL
263 C
264 C CARA NORTE
265 C
266 DO 98 K=1,NKM1
267 DO 97 I=1,NIM1
268 C
269 J=NJ
270 C
271 X2(I,J,K) = 0.25*(XC(I,J,K)+XC(I+1,J,K)+XC(I,J,K+1)+XC(I+1,J,K+1))
272 Y2(I,J,K) = 0.25*(YC(I,J,K)+YC(I+1,J,K)+YC(I,J,K+1)+YC(I+1,J,K+1))
273 Z2(I,J,K) = 0.25*(ZC(I,J,K)+ZC(I+1,J,K)+ZC(I,J,K+1)+ZC(I+1,J,K+1))
274 C
275 PS = SQRT((X2(I,J,K)-X(I,J,K))**2.+(Y2(I,J,K)-Y(I,J,K))**2.
276 * +(Z2(I,J,K)-Z(I,J,K))**2.)
277 SS = SQRT((X2(I,J,K)-X(I,J-1,K))**2.+(Y2(I,J,K)-Y(I,J-1,K))**2.
278 * +(Z2(I,J,K)-Z(I,J-1,K))**2.)
279 WFACT(I,J,K,2) = PS/(PS+SS)

```

```

280 C
281     DIAGS1 = XC(I+1,J,K+1) - XC(I,J,K)
282     DIAGS2 = YC(I+1,J,K+1) - YC(I,J,K)
283     DIAGS3 = ZC(I+1,J,K+1) - ZC(I,J,K)
284     GDIAS1 = XC(I,J,K+1) - XC(I+1,J,K)
285     GDIAS2 = YC(I,J,K+1) - YC(I+1,J,K)
286     GDIAS3 = ZC(I,J,K+1) - ZC(I+1,J,K)
287     AREA(I,J,K,2,1) = 0.5*(DIAGS3*GDIAS2 - DIAGS2*GDIAS3)
288     AREA(I,J,K,2,2) = 0.5*(DIAGS1*GDIAS3 - DIAGS3*GDIAS1)
289     AREA(I,J,K,2,3) = 0.5*(DIAGS2*GDIAS1 - DIAGS1*GDIAS2)
290     GAREA(I,J,K,2) = AREA(I,J,K,2,1)*AREA(I,J,K,2,1)
291     *                   +AREA(I,J,K,2,2)*AREA(I,J,K,2,2)
292     *                   +AREA(I,J,K,2,3)*AREA(I,J,K,2,3)
293 C
294     97 CONTINUE
295     98 CONTINUE
296 C
297 C
298 C     TOP WALL
299 C
300 C     CARA SUPERIOR
301 C
302     DO 96 J=1,NJM1
303     DO 95 I=1,NIM1
304 C
305     K=NK
306 C
307     X3(I,J,K) = 0.25*(XC(I,J,K)+XC(I+1,J,K)+XC(I,J+1,K)+XC(I+1,J+1,K))
308     Y3(I,J,K) = 0.25*(YC(I,J,K)+YC(I+1,J,K)+YC(I,J+1,K)+YC(I+1,J+1,K))
309     Z3(I,J,K) = 0.25*(ZC(I,J,K)+ZC(I+1,J,K)+ZC(I,J+1,K)+ZC(I+1,J+1,K))
310 C
311     PB = SQRT((X3(I,J,K)-X(I,J,K))**2.+(Y3(I,J,K)-Y(I,J,K))**2.
312     *         +(Z3(I,J,K)-Z(I,J,K))**2.)
313     BB = SQRT((X3(I,J,K)-X(I,J,K-1))**2.+(Y3(I,J,K)-Y(I,J,K-1))**2.
314     *         +(Z3(I,J,K)-Z(I,J,K-1))**2.)
315     WFACT(I,J,K,3) = PB/(PB+BB)
316 C
317     DIAGB1 = XC(I+1,J+1,K) - XC(I,J,K)
318     DIAGB2 = YC(I+1,J+1,K) - YC(I,J,K)
319     DIAGB3 = ZC(I+1,J+1,K) - ZC(I,J,K)
320     GDIAB1 = XC(I+1,J,K) - XC(I,J+1,K)
321     GDIAB2 = YC(I+1,J,K) - YC(I,J+1,K)
322     GDIAB3 = ZC(I+1,J,K) - ZC(I,J+1,K)
323     AREA(I,J,K,3,1) = 0.5*(DIAGB3*GDIAB2 - DIAGB2*GDIAB3)
324     AREA(I,J,K,3,2) = 0.5*(DIAGB1*GDIAB3 - DIAGB3*GDIAB1)
325     AREA(I,J,K,3,3) = 0.5*(DIAGB2*GDIAB1 - DIAGB1*GDIAB2)
326     GAREA(I,J,K,3) = AREA(I,J,K,3,1)*AREA(I,J,K,3,1)
327     *                   +AREA(I,J,K,3,2)*AREA(I,J,K,3,2)
328     *                   +AREA(I,J,K,3,3)*AREA(I,J,K,3,3)
329 C
330     95 CONTINUE
331     96 CONTINUE
332 C
333 C
334 C     VOLUME OF ELEMENTS
335 C
336 C     CALCULO DE LOS VOLUMENES
337 C
338     DO 18 K=1,NKM1
339     DO 17 J=1,NJM1
340     DO 16 I=1,NIM1
341 C
342     VOL(I,J,K) = (1.0/3.0)*(AREA(I+1,J,K,1,1)*X1(I+1,J,K)
343     *                   +AREA(I+1,J,K,1,2)*Y1(I+1,J,K)
344     *                   +AREA(I+1,J,K,1,3)*Z1(I+1,J,K)
345     *                   -AREA(I,J,K,1,1)*X1(I,J,K)
346     *                   -AREA(I,J,K,1,2)*Y1(I,J,K)
347     *                   -AREA(I,J,K,1,3)*Z1(I,J,K)
348     *                   -AREA(I,J,K,2,1)*X2(I,J,K)
349     *                   -AREA(I,J,K,2,2)*Y2(I,J,K)
350     *                   -AREA(I,J,K,2,3)*Z2(I,J,K)
351     *                   +AREA(I,J+1,K,2,1)*X2(I,J+1,K)
352     *                   +AREA(I,J+1,K,2,2)*Y2(I,J+1,K)
353     *                   +AREA(I,J+1,K,2,3)*Z2(I,J+1,K)
354     *                   -AREA(I,J,K,3,1)*X3(I,J,K)
355     *                   -AREA(I,J,K,3,2)*Y3(I,J,K)
356     *                   -AREA(I,J,K,3,3)*Z3(I,J,K)
357     *                   +AREA(I,J,K+1,3,1)*X3(I,J,K+1)
358     *                   +AREA(I,J,K+1,3,2)*Y3(I,J,K+1)
359     *                   +AREA(I,J,K+1,3,3)*Z3(I,J,K+1))
360 C
361     16 CONTINUE

```

```

362 17 CONTINUE
363 18 CONTINUE
364 C
365 C
366 C   SCALAR PRODUCTS BETWEEN AREA VECTORS
367 C
368 C   PRODUCTOS INTERNOS ENTRE AREAS
369 C
370 DO 21 K=1,NKM1
371 DO 20 J=1,NJM1
372 DO 19 I=1,NIM1
373 C
374   GAREA(I,J,K,1) = AREA(I,J,K,1,1)*AREA(I,J,K,1,1)
375   *   +AREA(I,J,K,1,2)*AREA(I,J,K,1,2)
376   *   +AREA(I,J,K,1,3)*AREA(I,J,K,1,3)
377   GAREA(I,J,K,2) = AREA(I,J,K,2,1)*AREA(I,J,K,2,1)
378   *   +AREA(I,J,K,2,2)*AREA(I,J,K,2,2)
379   *   +AREA(I,J,K,2,3)*AREA(I,J,K,2,3)
380   GAREA(I,J,K,3) = AREA(I,J,K,3,1)*AREA(I,J,K,3,1)
381   *   +AREA(I,J,K,3,2)*AREA(I,J,K,3,2)
382   *   +AREA(I,J,K,3,3)*AREA(I,J,K,3,3)
383   AREAWS(I,J,K) = AREA(I,J,K,1,1)*AREA(I,J,K,2,1)
384   *   +AREA(I,J,K,1,2)*AREA(I,J,K,2,2)
385   *   +AREA(I,J,K,1,3)*AREA(I,J,K,2,3)
386   AREAWB(I,J,K) = AREA(I,J,K,1,1)*AREA(I,J,K,3,1)
387   *   +AREA(I,J,K,1,2)*AREA(I,J,K,3,2)
388   *   +AREA(I,J,K,1,3)*AREA(I,J,K,3,3)
389   AREASB(I,J,K) = AREA(I,J,K,2,1)*AREA(I,J,K,3,1)
390   *   +AREA(I,J,K,2,2)*AREA(I,J,K,3,2)
391   *   +AREA(I,J,K,2,3)*AREA(I,J,K,3,3)
392   AREAET(I,J,K) = AREA(I+1,J,K,1,1)*AREA(I,J,K+1,3,1)
393   *   +AREA(I+1,J,K,1,2)*AREA(I,J,K+1,3,2)
394   *   +AREA(I+1,J,K,1,3)*AREA(I,J,K+1,3,3)
395   AREAEN(I,J,K) = AREA(I+1,J,K,1,1)*AREA(I,J+1,K,2,1)
396   *   +AREA(I+1,J,K,1,2)*AREA(I,J+1,K,2,2)
397   *   +AREA(I+1,J,K,1,3)*AREA(I,J+1,K,2,3)
398   AREANT(I,J,K) = AREA(I,J+1,K,2,1)*AREA(I,J,K+1,3,1)
399   *   +AREA(I,J+1,K,2,2)*AREA(I,J,K+1,3,2)
400   *   +AREA(I,J+1,K,2,3)*AREA(I,J,K+1,3,3)
401 C
402 19 CONTINUE
403 20 CONTINUE
404 21 CONTINUE
405 C
406 C
407 C   VISUALIZATION OUTPUT
408 C
409 C   ARCHIVO DE SALIDA X,Y,Z CON FORMATO ENSIGHT PARA SU VISUALIZACION
410 C
411   OPEN(12,FILE='MALLAXYZ.OUT')
412   WRITE(12,31)
413   31 FORMAT('GEOMETRIA DE CENTROIDES EN FORMATO ENSIGHT')
414   WRITE(12,32)
415   32 FORMAT(' ')
416   WRITE(12,33)
417   33 FORMAT('node id given')
418   WRITE(12,34)
419   34 FORMAT('element id given')
420   WRITE(12,35)
421   35 FORMAT('coordinates')
422   KMAX = NJM1
423   JMAX = NIM1
424   LMAX = NKM1
425   NUPO=KMAX*JMAX*LMAX
426   WRITE(12,36)NUPO
427   36 FORMAT(I8)
428   IDN = 0
429   DO 39 L=1,LMAX
430   DO 39 K=1,KMAX
431   DO 38 J=1,JMAX
432   IDN = IDN + 1
433   WRITE(12,37)IDN,X(J,K,L),Y(J,K,L),Z(J,K,L)
434   37 FORMAT(I8,3E12.5)
435   38 CONTINUE
436   39 CONTINUE
437 C
438 C   CONECTIVIDAD
439 C
440   WRITE(12,40)
441   40 FORMAT('part 1',/, 'conectividad',/, 'hexa8')
442   NUPO = (JMAX-1)*(KMAX-1)*(LMAX-1)
443   IDN = 0

```

```

444     IDO = 0
445     WRITE(12,41)NUNO
446 41  FORMAT(16)
447     DO 45 L=1,LMAX-1
448     DO 44 K=1,KMAX
449     DO 43 J=1,JMAX
450     IDM = IDM + 1
451     JMAS = IDM + 1
452     IVEC = IDM + JMAX
453     IVED = JMAS + JMAX
454     IVES = IDM + (JMAX*KMAX)
455     IVET = IVES + 1
456     IVEU = IVES + JMAX
457     IVEV = IVET + JMAX
458     IF(J .EQ. JMAX)GOTO 43
459     IF(K .EQ. KMAX)GOTO 43
460     IDO = IDO + 1
461 C   WRITE(7,42) IDO, IDM, JMAS, IVED, IVEC, IVES, IVET, IVEV, IVEU
462     WRITE(12,42) IDO, IDM, IVEC, IVED, JMAS, IVES, IVEU, IVEV, IVET
463 42  FORMAT(9I8)
464 43  CONTINUE
465 44  CONTINUE
466 45  CONTINUE
467 C
468 C
469     RETURN
470     END

```

A.4.4 Subrutina INITCOND.

```

1 C-----
2     SUBROUTINE INITCOND
3 C
4 C----INITIAL CONDITIONS
5 C----CONDICIONES INICIALES EN EL TIEMPO
6 C
7     INCLUDE 'param.f'
8     COMMON
9     1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
10    2    VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3),
11    3    AREAWS(NX,NY,NZ),AREAWB(NX,NY,NZ),AREASB(NX,NY,NZ),
12    4    AREAET(NX,NY,NZ),AREAEN(NX,NY,NZ),AREANT(NX,NY,NZ)
13    1/ALL/IT,JT,KT,NI,NJ,NK,NIM1,NJM1,NKM1,HIM2,NJM2,NKM2,GREAT
14    1/FLUPR/VISCOS,DENSIT,DEN(NX,NY,NZ)
15    2    ,VIS(NX,NY,NZ)
16    1/KASE T1/UIIN,FLOWIN
17    1/TRANS/DENO(NX,NY,NZ),APO(NX,NY,NZ)
18    1/VARTR/UO(NX,NY,NZ),VO(NX,NY,NZ),WO(NX,NY,NZ)
19    1/TIMEINC/DTIME
20    1/ANALITICA/UALIT(3,NX,NY,NZ)
21    DO 100 I=2,NIM2
22    DO 100 J=2,NJM2
23    DO 100 K=2,NKM2
24    UO(I,J,K)=UALIT(1,NIM2,J,K)*AREA(I,J,K,1,1)/SQRT(GAREA(I,J,K,1))
25    VO(I,J,K)=UALIT(1,NIM2,J,K)*AREA(I,J,K,1,2)/SQRT(GAREA(I,J,K,1))
26    WO(I,J,K)=0.0
27 C   UO(I,J,K)=UIN
28 C   VO(I,J,K)=0.0
29     DENO(I,J,K)=DENSIT
30 100 CONTINUE
31     RETURN
32     END

```

A.4.5 Subrutina CALCOEF.

```

1 C-----
2     SUBROUTINE CALCOEF(IINIT,JINIT,KINIT,IEND,JEND,KEND)
3     INCLUDE 'param.f'
4     COMMON
5     1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
6     2    VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3),
7     3    AREAWS(NX,NY,NZ),AREAWB(NX,NY,NZ),AREASB(NX,NY,NZ),
8     4    AREAET(NX,NY,NZ),AREAEN(NX,NY,NZ),AREANT(NX,NY,NZ)
9     3/VARRC/UFLUX(NX,NY,NZ,3)
10    1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
11    1    AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
12    2    SP(NX,NY,NZ)
13    1/FLUPR/VISCOS,DENSIT,DEN(NX,NY,NZ)
14    2    ,VIS(NX,NY,NZ)

```

```

15 C
16 C ASSEMBLY OF COEFFICIENTS (CONVECTION AND DIFFUSION)
17 C
18 C ENSAMBLE DE COEFICIENTES (CONVECCION Y DIFUSION)
19 C
20 DO 102 K=KINIT, KEND
21 DO 101 J=JINIT, JEND
22 DO 100 I=IINIT, IEND
23 C
24 C DENSITY IN ELEMENT FACES
25 C
26 C CALCULO DE LA DENSIDAD EN CARAS INTERMEDIAS
27 C
28 DENW = (1.-WFACT(I,J,K,1))*DEN(I,J,K)
29 * + WFACT(I,J,K,1)*DEN(I-1,J,K)
30 DENE = (1.-WFACT(I+1,J,K,1))*DEN(I+1,J,K)
31 * + WFACT(I+1,J,K,1)*DEN(I,J,K)
32 DENS = (1.-WFACT(I,J,K,2))*DEN(I,J,K)
33 * + WFACT(I,J,K,2)*DEN(I,J-1,K)
34 DENN = (1.-WFACT(I,J+1,K,2))*DEN(I,J+1,K)
35 * + WFACT(I,J+1,K,2)*DEN(I,J,K)
36 DENB = (1.-WFACT(I,J,K,3))*DEN(I,J,K)
37 * + WFACT(I,J,K,3)*DEN(I,J,K-1)
38 DENP = (1.-WFACT(I,J,K+1,3))*DEN(I,J,K+1)
39 * + WFACT(I,J,K+1,3)*DEN(I,J,K)
40 C
41 C CONVECTION COEFFICIENTS
42 C
43 C COEFICIENTES DE CONVECCION
44 C
45 CW = DENW*UFLUX(I,J,K,1)
46 CE = DENE*UFLUX(I+1,J,K,1)
47 CS = DENS*UFLUX(I,J,K,2)
48 CN = DENN*UFLUX(I,J+1,K,2)
49 CB = DENB*UFLUX(I,J,K,3)
50 CT = DENT*UFLUX(I,J,K+1,3)
51 C
52 C VISCOSITY IN ELEMENT FACES
53 C
54 C CALCULO DE LA VISCOSIDAD EN CARAS INTERMEDIAS
55 C
56 VISW = (1.-WFACT(I,J,K,1)) *VIS(I,J,K)
57 * + WFACT(I,J,K,1) *VIS(I-1,J,K)
58 VISE = (1.-WFACT(I+1,J,K,1))*VIS(I+1,J,K)
59 * + WFACT(I+1,J,K,1) *VIS(I,J,K)
60 VISS = (1.-WFACT(I,J,K,2)) *VIS(I,J,K)
61 * + WFACT(I,J,K,2) *VIS(I,J-1,K)
62 VISN = (1.-WFACT(I,J+1,K,2))*VIS(I,J+1,K)
63 * + WFACT(I,J+1,K,2) *VIS(I,J,K)
64 VISB = (1.-WFACT(I,J,K,3)) *VIS(I,J,K)
65 * + WFACT(I,J,K,3) *VIS(I,J,K-1)
66 VIST = (1.-WFACT(I,J,K+1,3))*VIS(I,J,K+1)
67 * + WFACT(I,J,K+1,3) *VIS(I,J,K)
68 C
69 C DIFUSION COEFFICIENTS
70 C
71 C COEFICIENTES DE DIFUSION
72 C
73 DW = VISW*GAREA(I,J,K,1)/VOL(I,J,K)
74 DE = VISE*GAREA(I+1,J,K,1)/VOL(I,J,K)
75 DS = VISS*GAREA(I,J,K,2)/VOL(I,J,K)
76 DN = VISN*GAREA(I,J+1,K,2)/VOL(I,J,K)
77 DB = VISB*GAREA(I,J,K,3)/VOL(I,J,K)
78 DT = VIST*GAREA(I,J,K+1,3)/VOL(I,J,K)
79 C
80 C MASS SOURCE TERM
81 C
82 C CALCULO DEL TERMINO FUENTE DE MASA
83 C
84 SMP=CN-CS+CE-CW+CT-CB
85 CP=AMAX1(0.0,SMP)
86 C
87 C ASSEMBLY OF MATRIX COEFFICIENTS
88 C
89 C ENSAMBLE DE LOS COEFICIENTES DE MATRIZ
90 C
91 AW(I,J,K)=AMAX1(ABS(0.5*CW),DW)+0.5*CW
92 AE(I,J,K)=AMAX1(ABS(0.5*CE),DE)-0.5*CE
93 AS(I,J,K)=AMAX1(ABS(0.5*CS),DS)+0.5*CS
94 AN(I,J,K)=AMAX1(ABS(0.5*CN),DN)-0.5*CN
95 AB(I,J,K)=AMAX1(ABS(0.5*CB),DB)+0.5*CB
96 AT(I,J,K)=AMAX1(ABS(0.5*CT),DT)-0.5*CT

```

```

97      SP(I,J,K)=-CP
98 C
99      100 CONTINUE
100     101 CONTINUE
101     102 CONTINUE
102 C
103     RETURN
104     END

```

A.4.6 Subrutina FUENTE.

```

1 C-----
2      SUBROUTINE FUENTE(IINIT,JINIT,KINIT,IEND,JEND,KEND)
3 C
4 C      VELOCITY SOURCE TERM COMPUTATION
5 C
6 C      ESTA SUBROUTINA CALCULA LOS TERMINOS FUENTE PARA
7 C      CADA COMPONENTE CARTESIANA DE LA VELOCIDAD
8 C
9 C
10     INCLUDE 'param.f'
11     COMMON
12     1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
13     2   VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3),
14     3   AREAW(NX,NY,NZ),AREAWB(NX,NY,NZ),AREASB(NX,NY,NZ),
15     4   AREAET(NX,NY,NZ),AREAEN(NX,NY,NZ),AREANT(NX,NY,NZ)
16     1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
17     2   ,W(NX,NY,NZ)
18     1/ALL/IT,JT,KT,NI,NJ,NK,NIM1,NJM1,NKM1,NIM2,NJM2,NKM2,GREAT
19     1/KASE T1/UIN,FLOWIN
20     1/FLUPR/VISCOS,DENSIT,DEN(NX,NY,NZ)
21     2   ,VIS(NX,NY,NZ)
22     COMMON
23     1/UVEL/RESORU,NSWPU,URFU
24     1/VVEL/RESORV,NSWPV,URFV
25     1/WVEL/RESORW,NSWPW,URFW
26     1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
27     1   AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
28     2   SP(NX,NY,NZ)
29     1/TRANS/DENO(NX,NY,NZ),APO(NX,NY,NZ)
30     1/VARTR/UC(NX,NY,NZ),VO(NX,NY,NZ),WO(NX,NY,NZ)
31     1/TIMEINC/DTIME
32 C
33     DO 100 K=KINIT,KEND
34     DO 99 J=JINIT,JEND
35     DO 98 I=IINIT,IEND
36 C
37 C
38 C      VELOCITY DERIVATIVES
39 C
40 C      DERIVADAS DE LA VELOCIDAD
41 C
42     UDERW1 = U(I,J,K) - U(I-1,J,K)
43     UDERW2 = 0.25*(U(I,J+1,K)-U(I,J-1,K)+U(I-1,J+1,K)-U(I-1,J-1,K))
44     UDERW3 = 0.25*(U(I,J,K+1)-U(I,J,K-1)+U(I-1,J,K+1)-U(I-1,J,K-1))
45     UDERS1 = 0.25*(U(I+1,J,K)-U(I-1,J,K)+U(I+1,J-1,K)-U(I-1,J-1,K))
46     UDERS2 = U(I,J,K) - U(I,J-1,K)
47     UDERS3 = 0.25*(U(I,J,K+1)-U(I,J,K-1)+U(I,J-1,K+1)-U(I,J-1,K-1))
48     UDERB1 = 0.25*(U(I+1,J,K)-U(I-1,J,K)+U(I+1,J,K-1)-U(I-1,J,K-1))
49     UDERB2 = 0.25*(U(I,J+1,K)-U(I,J-1,K)+U(I,J+1,K-1)-U(I,J-1,K-1))
50     UDERB3 = U(I,J,K) - U(I,J,K-1)
51     UDERE1 = U(I+1,J,K) - U(I,J,K)
52     UDERE2 = 0.25*(U(I,J+1,K)-U(I,J-1,K)+U(I+1,J+1,K)-U(I+1,J-1,K))
53     UDERE3 = 0.25*(U(I,J,K+1)-U(I,J,K-1)+U(I+1,J,K+1)-U(I+1,J,K-1))
54     UDERN1 = 0.25*(U(I+1,J,K)-U(I-1,J,K)+U(I+1,J+1,K)-U(I-1,J+1,K))
55     UDERN2 = U(I,J+1,K) - U(I,J,K)
56     UDERN3 = 0.25*(U(I,J,K+1)-U(I,J,K-1)+U(I,J+1,K+1)-U(I,J+1,K-1))
57     UDERT1 = 0.25*(U(I+1,J,K)-U(I-1,J,K)+U(I+1,J,K+1)-U(I-1,J,K+1))
58     UDERT2 = 0.25*(U(I,J+1,K)-U(I,J-1,K)+U(I,J+1,K+1)-U(I,J-1,K+1))
59     UDERT3 = U(I,J,K+1) - U(I,J,K)
60     VDERW1 = V(I,J,K) - V(I-1,J,K)
61     VDERW2 = 0.25*(V(I,J+1,K)-V(I,J-1,K)+V(I-1,J+1,K)-V(I-1,J-1,K))
62     VDERW3 = 0.25*(V(I,J,K+1)-V(I,J,K-1)+V(I-1,J,K+1)-V(I-1,J,K-1))
63     VDERS1 = 0.25*(V(I+1,J,K)-V(I-1,J,K)+V(I+1,J-1,K)-V(I-1,J-1,K))
64     VDERS2 = V(I,J,K) - V(I,J-1,K)
65     VDERS3 = 0.25*(V(I,J,K+1)-V(I,J,K-1)+V(I,J-1,K+1)-V(I,J-1,K-1))
66     VDERB1 = 0.25*(V(I+1,J,K)-V(I-1,J,K)+V(I+1,J,K-1)-V(I-1,J,K-1))
67     VDERB2 = 0.25*(V(I,J+1,K)-V(I,J-1,K)+V(I,J+1,K-1)-V(I,J-1,K-1))

```

```

68 VDFERR3 = V(I,J,K) - V(I,J,K-1)
69 VDERE1 = V(I+1,J,K) - V(I,J,K)
70 VDERE2 = 0.25*(V(I,J+1,K)-V(I,J-1,K)+W(I+1,J+1,K)-V(I+1,J-1,K))
71 VDERE3 = 0.25*(V(I,J,K+1)-V(I,J,K-1)+V(I+1,J,K+1)-V(I+1,J,K-1))
72 VDERN1 = 0.25*(V(I+1,J,K)-V(I-1,J,K)+V(I+1,J+1,K)-V(I-1,J+1,K))
73 VDERN2 = V(I,J+1,K) - V(I,J,K)
74 VDERN3 = 0.25*(V(I,J,K+1)-V(I,J,K-1)+V(I,J+1,K+1)-V(I,J+1,K-1))
75 VDERP1 = 0.25*(V(I+1,J,K)-V(I-1,J,K)+V(I+1,J,K+1)-V(I-1,J,K+1))
76 VDERT2 = 0.25*(V(I,J+1,K)-V(I,J-1,K)+V(I,J+1,K+1)-V(I,J-1,K+1))
77 VDERT3 = V(I,J,K+1) - V(I,J,K)
78 WDERW1 = W(I,J,K) - W(I-1,J,K)
79 WDERW2 = 0.25*(W(I,J+1,K)-W(I,J-1,K)+W(I-1,J+1,K)-W(I-1,J-1,K))
80 WDERW3 = 0.25*(W(I,J,K+1)-W(I,J,K-1)+W(I-1,J,K+1)-W(I-1,J,K-1))
81 WDERS1 = 0.25*(W(I+1,J,K)-W(I-1,J,K)+W(I+1,J-1,K)-W(I-1,J-1,K))
82 WDERS2 = W(I,J,K) - W(I,J-1,K)
83 WDERS3 = 0.25*(W(I,J,K+1)-W(I,J,K-1)+W(I,J-1,K+1)-W(I,J-1,K-1))
84 WDERB1 = 0.25*(W(I+1,J,K)-W(I-1,J,K)+W(I+1,J,K-1)-W(I-1,J,K-1))
85 WDERB2 = 0.25*(W(I+1,J,K)-W(I-1,J,K)+W(I+1,J+1,K)-W(I-1,J+1,K-1))
86 WDERB3 = W(I,J,K) - W(I,J,K-1)
87 WDERE1 = W(I+1,J,K) - W(I,J,K)
88 WDERE2 = 0.25*(W(I,J+1,K)-W(I,J-1,K)+W(I+1,J+1,K)-W(I+1,J-1,K))
89 WDERE3 = 0.25*(W(I,J,K+1)-W(I,J,K-1)+W(I+1,J,K+1)-W(I+1,J,K-1))
90 WDERN1 = 0.25*(W(I+1,J,K)-W(I-1,J,K)+W(I+1,J+1,K)-W(I-1,J+1,K))
91 WDERN2 = W(I,J+1,K) - W(I,J,K)
92 WDERN3 = 0.25*(W(I,J,K+1)-W(I,J,K-1)+W(I,J+1,K+1)-W(I,J+1,K-1))
93 WDERT1 = 0.25*(W(I+1,J,K)-W(I-1,J,K)+W(I+1,J,K+1)-W(I-1,J,K+1))
94 WDERT2 = 0.25*(W(I,J+1,K)-W(I,J-1,K)+W(I,J+1,K+1)-W(I,J-1,K+1))
95 WDERT3 = W(I,J,K+1) - W(I,J,K)
96 C
97 C
98 C PRESSURE GRADIENTS
99 C
100 C DERIVADAS DE LA PRESION
101 C
102 PPDER1 = 0.5*(P(I+1,J,K)-P(I-1,J,K))
103 PPDER2 = 0.5*(P(I,J+1,K)-P(I,J-1,K))
104 PPDER3 = 0.5*(P(I,J,K+1)-P(I,J,K-1))
105 C
106 C
107 C VISCOSITY IN ELEMENT FACES
108 C
109 C VISCOSIDAD EN CARAS INTERMEDIAS
110 C
111 VISW = (1.-WFACT(I,J,K,1)) *VIS(I,J,K)
112 * WFACT(I,J,K,1) *VIS(I-1,J,K)
113 VISE = (1.-WFACT(I+1,J,K,1)) *VIS(I+1,J,K)
114 * WFACT(I+1,J,K,1) *VIS(I,J,K)
115 VISS = (1.-WFACT(I,J,K,2)) *VIS(I,J,K)
116 * WFACT(I,J,K,2) *VIS(I,J-1,K)
117 VISN = (1.-WFACT(I,J+1,K,2)) *VIS(I,J+1,K)
118 * WFACT(I,J+1,K,2) *VIS(I,J,K)
119 VISB = (1.-WFACT(I,J,K,3)) *VIS(I,J,K)
120 * WFACT(I,J,K,3) *VIS(I,J,K-1)
121 VIST = (1.-WFACT(I,J,K+1,3)) *VIS(I,J,K+1)
122 * WFACT(I,J,K+1,3) *VIS(I,J,K)
123 C
124 C
125 C NON-ORTHOGONALITY SOURCE TERM
126 C
127 C TERMINO FUENTE DEBIDO A LA NO ORTOGONALIDAD DE LA MALLA
128 C
129 SUNO = (0.25/VOL(I,J,K)) * ((U(I+1,J,K)-U(I-1,J,K)) *
130 (AREAEN(I,J,K)-AREAWS(I,J,K)+AREAET(I,J,K)-AREAWB(I,J,K))
131 * (U(I,J+1,K)-U(I,J-1,K)) *
132 (AREAEN(I,J,K)-AREAWS(I,J,K)+AREANT(I,J,K)-AREASB(I,J,K))
133 * (U(I,J,K+1)-U(I,J,K-1)) *
134 (AREAET(I,J,K)-AREAWB(I,J,K)+AREANT(I,J,K)-AREASB(I,J,K))
135 + U(I+1,J+1,K) * 2.*AREAEN(I,J,K)
136 - U(I+1,J-1,K) * (AREAEN(I,J,K)+AREAWS(I,J,K))
137 + U(I+1,J,K+1) * 2.*AREAET(I,J,K)
138 - U(I+1,J,K-1) * (AREAET(I,J,K)+AREAWB(I,J,K))
139 + U(I-1,J-1,K) * 2.*AREAWS(I,J,K)
140 - U(I-1,J+1,K) * (AREAWS(I,J,K)+AREAEN(I,J,K))
141 + U(I-1,J,K-1) * 2.*AREAWB(I,J,K)
142 - U(I-1,J,K+1) * (AREAWB(I,J,K)+AREAET(I,J,K))
143 + U(I,J+1,K+1) * 2.*AREANT(I,J,K)
144 - U(I,J-1,K+1) * (AREASB(I,J,K)+AREANT(I,J,K))
145 + U(I,J-1,K-1) * 2.*AREASB(I,J,K)
146 - U(I,J+1,K-1) * (AREANT(I,J,K)+AREASB(I,J,K))
147 C
148 SVNO = (0.25/VOL(I,J,K)) * ((V(I+1,J,K)-V(I-1,J,K)) *
149 (AREAEN(I,J,K)-AREAWS(I,J,K)+AREAET(I,J,K)-AREAWB(I,J,K))

```

```

150 * * * * * + (V(I,J+1,K) - V(I,J-1,K)) *
151 * * * * * (AREAEN(I,J,K) - AREAWS(I,J,K) + AREANT(I,J,K) - AREASB(I,J,K))
152 * * * * * + (V(I,J,K+1) - V(I,J,K-1)) *
153 * * * * * (AREAET(I,J,K) - AREAWB(I,J,K) + AREANT(I,J,K) - AREASB(I,J,K))
154 * * * * * + V(I+1,J+1,K) * 2. * AREAEN(I,J,K)
155 * * * * * - V(I+1,J-1,K) * (AREAEN(I,J,K) + AREAWS(I,J,K))
156 * * * * * + V(I+1,J,K+1) * 2. * AREAET(I,J,K)
157 * * * * * - V(I+1,J,K-1) * (AREAET(I,J,K) + AREAWB(I,J,K))
158 * * * * * + V(I-1,J-1,K) * 2. * AREAWS(I,J,K)
159 * * * * * - V(I-1,J+1,K) * (AREAWS(I,J,K) + AREAEN(I,J,K))
160 * * * * * + V(I-1,J,K-1) * 2. * AREAWB(I,J,K)
161 * * * * * - V(I-1,J,K+1) * (AREAWB(I,J,K) + AREAET(I,J,K))
162 * * * * * + V(I,J+1,K+1) * 2. * AREANT(I,J,K)
163 * * * * * - V(I,J-1,K+1) * (AREASB(I,J,K) + AREANT(I,J,K))
164 * * * * * + V(I,J-1,K-1) * 2. * AREASB(I,J,K)
165 * * * * * - V(I,J+1,K-1) * (AREANT(I,J,K) + AREASB(I,J,K))
166 C
167 * * * * * SRNO = (0.25/VOL(I,J,K)) * ((W(I+1,J,K) - W(I-1,J,K)) *
168 * * * * * (AREAEN(I,J,K) - AREAWS(I,J,K) + AREAET(I,J,K) - AREAWB(I,J,K))
169 * * * * * + (W(I,J+1,K) - W(I,J-1,K)) *
170 * * * * * (AREAEN(I,J,K) - AREAWS(I,J,K) + AREANT(I,J,K) - AREASB(I,J,K))
171 * * * * * + (W(I,J,K+1) - W(I,J,K-1)) *
172 * * * * * (AREAET(I,J,K) - AREAWB(I,J,K) + AREANT(I,J,K) - AREASB(I,J,K))
173 * * * * * + W(I+1,J+1,K) * 2. * AREAEN(I,J,K)
174 * * * * * - W(I+1,J-1,K) * (AREAEN(I,J,K) + AREAWS(I,J,K))
175 * * * * * + W(I+1,J,K+1) * 2. * AREAET(I,J,K)
176 * * * * * - W(I+1,J,K-1) * (AREAET(I,J,K) + AREAWB(I,J,K))
177 * * * * * + W(I-1,J-1,K) * 2. * AREAWS(I,J,K)
178 * * * * * - W(I-1,J+1,K) * (AREAWS(I,J,K) + AREAEN(I,J,K))
179 * * * * * + W(I-1,J,K-1) * 2. * AREAWB(I,J,K)
180 * * * * * - W(I-1,J,K+1) * (AREAWB(I,J,K) + AREAET(I,J,K))
181 * * * * * + W(I,J+1,K+1) * 2. * AREANT(I,J,K)
182 * * * * * - W(I,J-1,K+1) * (AREASB(I,J,K) + AREANT(I,J,K))
183 * * * * * + W(I,J-1,K-1) * 2. * AREASB(I,J,K)
184 * * * * * - W(I,J+1,K-1) * (AREANT(I,J,K) + AREASB(I,J,K))
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 * * * * * SU(I,J,K,L) = -0.5 * (AREA(I+1,J,K,1,L) + AREA(I,J,K,1,L)) * PFDER1
194 * * * * * -0.5 * (AREA(I,J+1,K,2,L) + AREA(I,J,K,2,L)) * PFDER2
195 * * * * * -0.5 * (AREA(I,J,K+1,3,L) + AREA(I,J,K,3,L)) * PFDER3
196 * * * * * + (1./VOL(I,J,K)) *
197 * * * * * (VISW * (-AREA(I,J,K,1,1) * (AREA(I,J,K,1,L) * UDERW1
198 * * * * * + AREA(I,J,K,2,L) * UDERW2 + AREA(I,J,K,3,L) * UDERW3)
199 * * * * * - AREA(I,J,K,1,2) * (AREA(I,J,K,1,L) * VDERW1
200 * * * * * + AREA(I,J,K,2,L) * VDERW2 + AREA(I,J,K,3,L) * VDERW3)
201 * * * * * - AREA(I,J,K,1,3) * (AREA(I,J,K,1,L) * WDERW1
202 * * * * * + AREA(I,J,K,2,L) * WDERW2 + AREA(I,J,K,3,L) * WDERW3)
203 * * * * * + VISS * (-AREA(I,J,K,2,1) * (AREA(I,J,K,1,L) * UDERS1
204 * * * * * + AREA(I,J,K,2,L) * UDERS2 + AREA(I,J,K,3,L) * UDERS3)
205 * * * * * - AREA(I,J,K,2,2) * (AREA(I,J,K,1,L) * VDERS1
206 * * * * * + AREA(I,J,K,2,L) * VDERS2 + AREA(I,J,K,3,L) * VDERS3)
207 * * * * * - AREA(I,J,K,2,3) * (AREA(I,J,K,1,L) * WDERS1
208 * * * * * + AREA(I,J,K,2,L) * WDERS2 + AREA(I,J,K,3,L) * WDERS3)
209 * * * * * + VISB * (-AREA(I,J,K,3,1) * (AREA(I,J,K,1,L) * UDERB1
210 * * * * * + AREA(I,J,K,2,L) * UDERB2 + AREA(I,J,K,3,L) * UDERB3)
211 * * * * * - AREA(I,J,K,3,2) * (AREA(I,J,K,1,L) * VDERB1
212 * * * * * + AREA(I,J,K,2,L) * VDERB2 + AREA(I,J,K,3,L) * VDERB3)
213 * * * * * - AREA(I,J,K,3,3) * (AREA(I,J,K,1,L) * WDERB1
214 * * * * * + AREA(I,J,K,2,L) * WDERB2 + AREA(I,J,K,3,L) * WDERB3)
215 * * * * * + VISE * ( AREA(I+1,J,K,1,1) * (AREA(I+1,J,K,1,L) * UDERE1
216 * * * * * + AREA(I,J+1,K,2,L) * UDERE2 + AREA(I,J,K+1,3,L) * UDERE3)
217 * * * * * + AREA(I+1,J,K,1,2) * (AREA(I+1,J,K,1,L) * VDERE1
218 * * * * * + AREA(I,J+1,K,2,L) * VDERE2 + AREA(I,J,K+1,3,L) * VDERE3)
219 * * * * * + AREA(I+1,J,K,1,3) * (AREA(I+1,J,K,1,L) * WDERE1
220 * * * * * + AREA(I,J+1,K,2,L) * WDERE2 + AREA(I,J,K+1,3,L) * WDERE3)
221 * * * * * + VISN * ( AREA(I,J+1,K,2,1) * (AREA(I+1,J,K,1,L) * UDERN1
222 * * * * * + AREA(I,J+1,K,2,L) * UDERN2 + AREA(I,J,K+1,3,L) * UDERN3)
223 * * * * * + AREA(I,J+1,K,2,2) * (AREA(I+1,J,K,1,L) * VDERN1
224 * * * * * + AREA(I,J+1,K,2,L) * VDERN2 + AREA(I,J,K+1,3,L) * VDERN3)
225 * * * * * + AREA(I,J+1,K,2,3) * (AREA(I+1,J,K,1,L) * WDERN1
226 * * * * * + AREA(I,J+1,K,2,L) * WDERN2 + AREA(I,J,K+1,3,L) * WDERN3)
227 * * * * * + VIST * ( AREA(I,J,K+1,3,1) * (AREA(I+1,J,K,1,L) * UDERT1
228 * * * * * + AREA(I,J+1,K,2,L) * UDERT2 + AREA(I,J,K+1,3,L) * UDERT3)
229 * * * * * + AREA(I,J,K+1,3,2) * (AREA(I+1,J,K,1,L) * VDERT1
230 * * * * * + AREA(I,J+1,K,2,L) * VDERT2 + AREA(I,J,K+1,3,L) * VDERT3)
231 * * * * * + AREA(I,J,K+1,3,3) * (AREA(I+1,J,K,1,L) * WDERT1

```

```

232      *          +AREA(I,J+1,K,2,L)*WBERT2+AREA(I,J,K+1,3,L)*WBERT3)))
233 50 CONTINUE
234 C
235 APO(I,J,K)=DENO(I,J,K)*VOL(I,J,K)/DTIME
236 SU(I,J,K,1)=SU(I,J,K,1)+SUNO+APO(I,J,K)*UO(I,J,K)
237 *      -SP(I,J,K)*U(I,J,K)
238 SU(I,J,K,2)=SU(I,J,K,2)+SVNO+APO(I,J,K)*VO(I,J,K)
239 *      -SP(I,J,K)*V(I,J,K)
240 SU(I,J,K,3)=SU(I,J,K,3)+SWNO+APO(I,J,K)*WO(I,J,K)
241 *      -SP(I,J,K)*W(I,J,K)
242 C
243 98 CONTINUE
244 99 CONTINUE
245 100 CONTINUE
246 C
247 C      SPECIAL MODIFICATIONS
248 C
249 C      MODIFICACIONES ESPECIALES
250 C
251 C      CALL MODU
252 C
253 C
254 C      FINAL COEFFICIENTS ASSEMBLY AND RESIDUALS
255 C
256 C      ENSAMBLE FINAL DE COEFICIENTES, CALCULO DE RESIDUOS
257 C
258 RESORU = 0.0
259 RESORV = 0.0
260 RESORW = 0.0
261 C
262 DO 103 K=KINIT,KEND
263 DO 102 J=JINIT,JEND
264 DO 101 I=IINIT,IEND
265 C
266 AP(I,J,K,1)=AN(I,J,K)+AS(I,J,K)+AE(I,J,K)+AW(I,J,K)+AT(I,J,K)
267 *      +AB(I,J,K)-SP(I,J,K)+APO(I,J,K)
268 C
269 RESOR=AN(I,J,K)*U(I,J+1,K)+AS(I,J,K)*U(I,J-1,K)+
270 *      AE(I,J,K)*U(I+1,J,K)+AW(I,J,K)*U(I-1,J,K)+
271 *      AT(I,J,K)*U(I,J,K+1)+AB(I,J,K)*U(I,J,K-1)-AP(I,J,K,1)*
272 *      U(I,J,K)+SU(I,J,K,1)
273 SORVOL=GREAT*VOL(I,J,K)
274 IF(-SP(I,J,K).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
275 RESORU=RESORU+ABS(RESOR)
276 C
277 RESOR=AN(I,J,K)*V(I,J+1,K)+AS(I,J,K)*V(I,J-1,K)+AE(I,J,K)*
278 *      V(I+1,J,K)+AW(I,J,K)*V(I-1,J,K)+AT(I,J,K)*V(I,J,K+1)
279 *      +AB(I,J,K)*V(I,J,K-1)-AP(I,J,K,1)*V(I,J,K)+SU(I,J,K,2)
280 IF(-SP(I,J,K).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
281 RESORV=RESORV+ABS(RESOR)
282 C
283 RESOR=AN(I,J,K)*W(I,J+1,K)+AS(I,J,K)*W(I,J-1,K)+AE(I,J,K)*
284 *      W(I+1,J,K)+AW(I,J,K)*W(I-1,J,K)+AT(I,J,K)*W(I,J,K+1)
285 *      +AB(I,J,K)*W(I,J,K-1)-AP(I,J,K,1)*W(I,J,K)+SU(I,J,K,3)
286 IF(-SP(I,J,K).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
287 RESORW=RESORW+ABS(RESOR)
288 C
289 C
290 C      UNDER-RELAXATION OF AP
291 C
292 C      ATENUACION DEL COEFICIENTE AP
293 C
294 AP(I,J,K,1)=AP(I,J,K,1)/URFU
295 SU(I,J,K,1)=SU(I,J,K,1)+(1.-URFU)*AP(I,J,K,1)*U(I,J,K)
296 SU(I,J,K,2)=SU(I,J,K,2)+(1.-URFU)*AP(I,J,K,1)*V(I,J,K)
297 SU(I,J,K,3)=SU(I,J,K,3)+(1.-URFU)*AP(I,J,K,1)*W(I,J,K)
298 C
299 101 CONTINUE
300 102 CONTINUE
301 103 CONTINUE
302 C
303 C      CALL MODAP
304 C
305 C      MATRIX SOLVER CALLING
306 C
307 C      LLAMADO AL INVERSOR DE MATRIZ
308 C
309 DO 104 N=1,NSWPU
310 104 CALL LISOLV(2,2,2,NIM1,NJM1,NKM1,IT,JT,KT,1,U,1)
311 DO 105 N=1,NSWPV
312 105 CALL LISOLV(2,2,2,NIM1,NJM1,NKM1,IT,JT,KT,2,V,1)
313 DO 106 N=1,NSWPW

```

```

314 106 CALL LISOLV(2,2,2,HIM1,NJM1,NFM1,IT,JT,KT,3,W,1)
315 RETURN
316 END

```

A.4.7 Subrutina PROMOD.

```

1  C-----
2  SUBROUTINE PROMOD
3  INCLUDE 'param.f'
4  COMMON
5  1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
6  2 VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3),
7  3 AREAWS(NX,NY,NZ),AREAWB(NX,NY,NZ),AREASE(NX,NY,NZ),
8  4 AREAET(NX,NY,NZ),AREAEN(NX,NY,NZ),AREANT(NX,NY,NZ)
9  1/FRONT/YW(NY,NZ),YE(NY,NZ),YS(NX,NZ),YN(NX,NZ),YB(NX,NY),YT(NX,NY)
10 1/UVEL/RESORU,NSWPU,URFU
11 1/VVEL/RESORV,NSWPV,URFV
12 1/WVEL/RESORW,NSWPW,URFW
13 1/PCOR/RESORM,NSWPP,URFP,
14 1 IPREF,JPREF,KPREF
15 1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
16 2 W(NX,NY,NZ)
17 1/ALL/IT,JT,KT,NI,NJ,NK,NIM1,NJM1,NKM1,NIM2,NJM2,NKM2,GREAT
18 1/FLUPR/VISCOS,DENSIT,DEN(NX,NY,NZ)
19 2 VIS(NX,NY,NZ)
20 1/KASE T1/UIIN,FLOWIN
21 1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
22 1 AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
23 2 SP(NX,NY,NZ)
24 1/FLO/FLOW
25  C
26  C
27  ENTRY MODU
28  C
29  C EAST OUTLET CONDITIONS
30  C
31  C CONDICIONES EN LA SALIDA ESTE
32  C
33  ARDENT=0.
34  FLOW=0.
35  I=NIM2
36  DO K=2,NKM2
37  DO J=2,NJM2
38  ARDEN=DEN(I,J,K)*SQRT(GAREA(I,J,K,1))
39  ARDENT=ARDENT+ARDEN
40  FLOW=FLOW+DEN(I,J,K)*(AREA(I,J,K,1,1)*U(I,J,K)
41  *+AREA(I,J,K,1,2)*V(I,J,K)+AREA(I,J,K,1,3)*W(I,J,K))
42  ENDDO
43  ENDDO
44  UINC=(FLOWIN-FLOW)/ARDENT
45  DO J=2,NJM2
46  DO K=2,NKM2
47  U(NIM1,J,K)=U(NIM2,J,K)+UINC
48  C U(NIM1,J,K)=U(NIM2,J,K)
49  ENDDO
50  ENDDO
51  WRITE(7,200)FLOW
52  200 FORMAT('FLOW=',E15.7)
53  C
54  RETURN
55  C
56  C
57  ENTRY MODP
58  C-----EAST WALL
59  C-----PARED ESTE
60  DO J=1,NJM1
61  DO K=1,NKM1
62  P(NIM1,J,K)=P(NIM2,J,K)
63  ENDDO
64  ENDDO
65  DO K=1,NK
66  DO J=1,NJ
67  DO I=1,NI
68  PP(I,J,K)=0.0
69  ENDDO
70  ENDDO
71  ENDDO
72  C
73  RETURN
74  C

```

```

75     ENTRY MODAP
76     C
77     C-----WEST WALL
78     C-----PARED OESTE
79         I=1
80         DO 250 K=2,NKM2
81         DO 249 J=2,NJM2
82         PP(I,J,K) = PP(I+1,J,K)
83         P(I,J,K) = P(I+1,J,K)+P(I+2,J,K)-P(I+3,J,K)
84         AP(I,J,K,1) = AP(I+1,J,K,1)
85         249 CONTINUE
86         250 CONTINUE
87     C
88     C-----EAST WALL
89     C-----PARED ESTE
90         I=NIM1
91         DO 252 K=2,NKM2
92         DO 251 J=2,NJM2
93         PP(I,J,K) = PP(I-1,J,K)
94         P(I,J,K) = P(I-1,J,K)+P(I-2,J,K)-P(I-3,J,K)
95         AP(I,J,K,1) = AP(I-1,J,K,1)
96         251 CONTINUE
97         252 CONTINUE
98     C
99     C-----SOUTH WALL
100    C-----PARED SUR
101        J=1
102        DO 254 K=2,NKM2
103        DO 253 I=2,NIM2
104        PP(I,J,K) = PP(I,J+1,K)
105        P(I,J,K) = P(I,J+1,K)+P(I,J+2,K)-P(I,J+3,K)
106        AP(I,J,K,1) = AP(I,J+1,K,1)
107        253 CONTINUE
108        254 CONTINUE
109    C
110    C-----NORTH WALL
111    C-----PARED NORTE
112        J=NJM1
113        DO 256 K=2,NKM2
114        DO 255 I=2,NIM2
115        PP(I,J,K) = PP(I,J-1,K)
116        P(I,J,K) = P(I,J-1,K)+P(I,J-2,K)-P(I,J-3,K)
117        AP(I,J,K,1) = AP(I,J-1,K,1)
118        255 CONTINUE
119        256 CONTINUE
120    C
121    C-----BOTTOM WALL
122    C-----PARED ABAJO
123        K=1
124        DO 258 J=2,NJM2
125        DO 257 I=2,NIM2
126        PP(I,J,K) = PP(I,J,K+1)
127        P(I,J,K) = P(I,J,K+1)+P(I,J,K+2)-P(I,J,K+3)
128        AP(I,J,K,1) = AP(I,J,K+1,1)
129        257 CONTINUE
130        258 CONTINUE
131    C
132    C-----TOP WALL
133    C-----PARED ARRIBA
134        K=NKM1
135        DO 260 J=2,NJM2
136        DO 259 I=2,NIM2
137        PP(I,J,K) = PP(I,J,K-1)
138        P(I,J,K) = P(I,J,K-1)+P(I,J,K-2)-P(I,J,K-3)
139        AP(I,J,K,1) = AP(I,J,K-1,1)
140        259 CONTINUE
141        260 CONTINUE
142    RETURN
143    C
144    ENTRY MODAN
145    C-----WEST WALL
146    C-----PARED OESTE
147        I=2
148        DO K=1,NK
149        DO J=1,NJ
150        AW(I,J,K)=0.0
151    ENDDO
152    ENDDO
153    C
154    C-----EAST WALL
155    C-----PARED ESTE
156        I=NIM2

```

```

157      DO K=1,NK
158      DO J=1,NJ
159      AE(I,J,K)=0.0
160      ENDDO
161      ENDDO
162      C
163      C-----SOUTH WALL
164      C-----PARED SUR
165      J=2
166      DO K=1,NK
167      DO I=1,NI
168      AS(I,J,K)=0.0
169      ENDDO
170      ENDDO
171      C
172      C-----NORTH WALL
173      C-----PARED NORTE
174      J=NJM2
175      DO K=1,NK
176      DO I=1,NI
177      AN(I,J,K)=0.0
178      ENDDO
179      ENDDO
180      C
181      C-----BOTTOM WALL
182      C-----PARED ABAJO
183      K=2
184      DO J=1,NJ
185      DO I=1,NI
186      AB(I,J,K)=0.0
187      ENDDO
188      ENDDO
189      C
190      C-----TOP WALL
191      C-----PARED ARRIBA
192      K=NKM2
193      DO J=1,NJ
194      DO I=1,NI
195      AT(I,J,K)=0.0
196      ENDDO
197      ENDDO
198      RETURN
199      END

```

A.4.8 Subrutina LISOLV.

```

1  C-----
2  SUBROUTINE LISOLV(ISTART,JSTART,KSTART,NI,NJ,NK,IT,JT,KT,L,PHI,M)
3  INCLUDE 'param.f'
4  DIMENSION PHI(IT,JT,KT),A(NY),B(NY),C(NY),D(NY)
5  COMMON
6  1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
7  1 AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
8  2 SP(NX,NY,NZ)
9  C
10 C
11 C  MATRIX SOLVER
12 C
13 C  INVERSOR DE MATRIZ
14 C
15 NIM1=NI-1
16 NJM1=NJ-1
17 NKM1=NK-1
18 JSTM1=JSTART-1
19 A(JSTM1)=0.0
20 C-----COMMENCE T-B SWEEP
21 DO 100 K=KSTART,NKM1
22 C-----COMMENCE W-E SWEEP
23 DO 100 I=ISTART,NIM1
24 C(JSTM1)=PHI(I,JSTM1,K)
25 C-----COMMENCE S-N TRAVERSE
26 DO 101 J=JSTART,NJM1
27 C-----ASSEMBLE TDMA COEFFICIENTS
28 A(J)=AN(I,J,K)
29 B(J)=AS(I,J,K)
30 C(J)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)+
31 1AT(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K,L)
32 D(J)=AP(I,J,K,M)
33 C-----CALCULATE COEFFICIENTS OF RECURRENCE FORMULA

```

```

34      TERM=1./ (D(J)-E(J))*A(J-1)
35      A(J)=A(J)*TERM
36      101 C(J)=(C(J)+B(J))*C(J-1))*TERM
37 C-----OBTAIN NEW PHI
38      DO 102 JJ=JSTART, NJM1
39      J=NJ+JSTM1-JJ
40      102 PHI(I,J,K)=A(J)*PHI(I,J+1,K)+C(J)
41      100 CONTINUE
42      RETURN
43      END

```

A.4.9 Subrutina RHIECHOW.

```

1 C-----
2      SUBROUTINE RHIECHOW(IINIT, JINIT, KINIT, IEND, JEND, KEND)
3 C
4 C      INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
5 C      CELLS BY RHIE-CHOW METHOD
6 C
7 C      RHIE-CHOW CALCULA EL FLUJO EN LAS CARAS DE LOS
8 C      VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
9 C      DE RHIE-CHOW
10 C
11      INCLUDE 'param.f'
12      COMMON
13      1/COEF/AP(NX,NY,NZ,2), AN(NX,NY,NZ), AS(NX,NY,NZ), AE(NX,NY,NZ),
14      1      AW(NX,NY,NZ), AT(NX,NY,NZ), AB(NX,NY,NZ), SU(NX,NY,NZ,3),
15      2      SP(NX,NY,NZ)
16      1/VAR/U(NX,NY,NZ), V(NX,NY,NZ), P(NX,NY,NZ), FP(NX,NY,NZ)
17      2      W(NX,NY,NZ)
18      3/VARRC/UFLUX(NX,NY,NZ,3)
19      1/GGC/AREA(NX,NY,NZ,3,3), GAREA(NX,NY,NZ,3),
20      2      VOL(NX,NY,NZ), WFACT(NX,NY,NZ,3)
21      1/UVEL/RESORU, NSWPU, URFU
22      DO 102 K=KINIT, KEND
23      DO 101 J=JINIT, JEND
24      DO 100 I=IINIT, IEND
25 C
26 C
27 C      PRESSURE DERIVATIVES
28 C
29 C      DERIVADAS DE LA PRESION EN LOS NODOS
30 C
31      PPDER1 = 0.5*( P(I+1,J,K) - P(I-1,J,K) )
32      PWDER1 = 0.5*( P(I,J,K) - P(I-2,J,K) )
33 C
34 C
35 C      DERIVADAS DE LA PRESION EN LAS CARAS
36 C
37      FPWDER1 = ( P(I,J,K) - P(I-1,J,K) )
38 C
39 C
40 C      FLUX ASSEMBLY
41 C
42 C      ENSAMBLE DEL FLUJO EN LAS CARAS
43 C
44      POLAW = (1.-WFACT(I,J,K,1))* (1./ (AP(I,J,K,1)*URFU))
45      *      +WFACT(I,J,K,1)* (1./ (AP(I-1,J,K,1)*URFU))
46      POLAW1 = GAREA(I,J,K,1)*POLAW
47 C
48      PRGRAW1 = (1.-WFACT(I,J,K,1))*PPDER1
49      *      + WFACT(I,J,K,1)*PWDER1 - FPWDER1
50 C
51      UINTW1= (1.-WFACT(I,J,K,1))*U(I,J,K) + WFACT(I,J,K,1)*U(I-1,J,K)
52      UINTW2= (1.-WFACT(I,J,K,1))*V(I,J,K) + WFACT(I,J,K,1)*V(I-1,J,K)
53      UINTW3= (1.-WFACT(I,J,K,1))*W(I,J,K) + WFACT(I,J,K,1)*W(I-1,J,K)
54 C
55      UFLUX(I,J,K,1) = AREA(I,J,K,1,1)*UINTW1 + AREA(I,J,K,1,2)*UINTW2
56      *      +AREA(I,J,K,1,3)*UINTW3 + POLAW1*PRGRAW1
57 C
58      100 CONTINUE
59      101 CONTINUE
60      102 CONTINUE
61      RETURN
62      END
63 C
64 C-----
65      SUBROUTINE RHIECHOWS(IINIT, JINIT, KINIT, IEND, JEND, KEND)
66 C
67 C      INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL

```

```

68 C      CELLS BY RHIE-CHOW METHOD
69 C
70 C      RHIE-CHOW CALCULA EL FLUJO EN LAS CARAS DE LOS
71 C      VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
72 C      DE RHIE-CHOW
73 C
74 C      INCLUDE 'param.f'
75 C      COMMON
76 C      1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
77 C      1      AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
78 C      2      SP(NX,NY,NZ)
79 C      1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
80 C      2      W(NX,NY,NZ)
81 C      3/VARRC/UFLUX(NX,NY,NZ,3)
82 C      1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
83 C      2      VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3)
84 C      1/VEL/RESORU,NSWPU,URFU
85 C      DO 102 K=KINIT,KEND
86 C      DO 101 J=JINIT,JEND
87 C      DO 100 I=IINIT,IEND
88 C
89 C
90 C      PRESSURE DERIVATIVES
91 C
92 C      DERIVADAS DE LA PRESION EN LOS NODOS
93 C
94 C      PPDER2 = 0.5*( P(I,J+1,K) - P(I,J-1,K) )
95 C      PSDER2 = 0.5*( P(I,J,K) - P(I,J-2,K) )
96 C
97 C
98 C      DERIVADAS DE LA PRESION EN LAS CARAS
99 C
100 C      FPSDER2 = ( P(I,J,K) - P(I,J-1,K) )
101 C
102 C
103 C      FLUX ASSEMBLY
104 C
105 C      ENSAMBLE DEL FLUJO EN LAS CARAS
106 C
107 C      POLAS = (1.-WFACT(I,J,K,2))* (1./(AP(I,J,K,1)*URFU))
108 C      *      +WFACT(I,J,K,2)* (1./(AP(I,J-1,K,1)*URFU))
109 C      POLAS2 = GAREA(I,J,K,2)*POLAS
110 C
111 C      PRGRAS2 = (1.-WFACT(I,J,K,2))*PPDER2
112 C      *      + WFACT(I,J,K,2)*PSDER2 - FPSDER2
113 C
114 C      UINTS1= (1.-WFACT(I,J,K,2))*U(I,J,K) + WFACT(I,J,K,2)*U(I,J-1,K)
115 C      UINTS2= (1.-WFACT(I,J,K,2))*V(I,J,K) + WFACT(I,J,K,2)*V(I,J-1,K)
116 C      UINTS3= (1.-WFACT(I,J,K,2))*W(I,J,K) + WFACT(I,J,K,2)*W(I,J-1,K)
117 C
118 C      UFLUX(I,J,K,2) = AREA(I,J,K,2,1)*UINTS1 + AREA(I,J,K,2,2)*UINTS2
119 C      *      +AREA(I,J,K,2,3)*UINTS3 + POLAS2*PRGRAS2
120 C
121 C      100 CONTINUE
122 C      101 CONTINUE
123 C      102 CONTINUE
124 C      RETURN
125 C      END
126 C
127 C-----
128 C      SUBROUTINE RHIECHOW(IINIT,JINIT,KINIT,IEND,JEND,KEND)
129 C
130 C      INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
131 C      CELLS BY RHIE-CHOW METHOD
132 C
133 C      RHIE-CHOW CALCULA EL FLUJO EN LAS CARAS DE LOS
134 C      VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
135 C      DE RHIE-CHOW
136 C
137 C      INCLUDE 'param.f'
138 C      COMMON
139 C      1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
140 C      1      AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
141 C      2      SP(NX,NY,NZ)
142 C      1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
143 C      2      W(NX,NY,NZ)
144 C      3/VARRC/UFLUX(NX,NY,NZ,3)
145 C      1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
146 C      2      VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3)
147 C      1/VEL/RESORU,NSWPU,URFU
148 C      DO 102 K=KINIT,KEND
149 C      DO 101 J=JINIT,JEND

```

```

150      DO 100 I=IINIT, IEND
151      C
152      C
153      C      PRESSURE DERIVATIVES
154      C
155      C      DERIVADAS DE LA PRESION EN LOS NODOS
156      C
157      PPDER3 = 0.5*( P(I,J,K+1) - P(I,J,K-1) )
158      PBDER3 = 0.5*( P(I,J,K) - P(I,J,K-2) )
159      C
160      C
161      C      DERIVADAS DE LA PRESION EN LAS CARAS
162      C
163      FPBDER3 = ( P(I,J,K) - P(I,J,K-1) )
164      C
165      C
166      C      FLUX ASSEMBLY
167      C
168      C      ENSAMBLE DEL FLUJO EN LAS CARAS
169      C
170      POLAB = (1.-WFACT(I,J,K,3))*(1./(AP(I,J,K,1)*URFU))
171      *      +WFACT(I,J,K,3)*(1./(AP(I,J,K-1,1)*URFU))
172      POLAB3 = GAREA(I,J,K,3)*POLAB
173      C
174      PRGRAB3 = (1.-WFACT(I,J,K,3))*PPDER3
175      *      + WFACT(I,J,K,3)*PBDER3 - FPBDER3
176      C
177      UINTB1= (1.-WFACT(I,J,K,3))*U(I,J,K) + WFACT(I,J,K,3)*U(I,J,K-1)
178      UINTB2= (1.-WFACT(I,J,K,3))*V(I,J,K) + WFACT(I,J,K,3)*V(I,J,K-1)
179      UINTB3= (1.-WFACT(I,J,K,3))*W(I,J,K) + WFACT(I,J,K,3)*W(I,J,K-1)
180      C
181      UFLUX(I,J,K,3) = AREA(I,J,K,3,1)*UINTB1 + AREA(I,J,K,3,2)*UINTB2
182      *      +AREA(I,J,K,3,3)*UINTB3 + POLAB3*PRGRAB3
183      C
184      100 CONTINUE
185      101 CONTINUE
186      102 CONTINUE
187      RETURN
188      END

```

A.4.10 Subrutina RHICHW.

```

1      C-----
2      SUBROUTINE RHICHWW(IINIT,JINIT,KINIT,IEND,JEND,KEND)
3      C
4      C      INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
5      C      CELLS BY LINEAR INTERPOLATION
6      C
7      C      RHICHW CALCULA EL FLUJO EN LAS CARAS DE LOS
8      C      VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
9      C      LINEAL
10     C
11     INCLUDE 'param.f'
12     COMMON
13     1/COEF/AP(NX,NY,NZ,2), AN(NX,NY,NZ), AS(NX,NY,NZ), AE(NX,NY,NZ),
14     1      AW(NX,NY,NZ), AT(NX,NY,NZ), AB(NX,NY,NZ), SU(NX,NY,NZ,3),
15     2      SP(NX,NY,NZ)
16     1/VAR/U(NX,NY,NZ), V(NX,NY,NZ), P(NX,NY,NZ), PP(NX,NY,NZ)
17     2      ,W(NX,NY,NZ)
18     3/VARRC/UFLUX(NX,NY,NZ,3)
19     1/GCC/AREA(NX,NY,NZ,3,3), GAREA(NX,NY,NZ,3),
20     2      VOL(NX,NY,NZ), WFACT(NX,NY,NZ,3)
21     DO 102 K=KINIT,KEND
22     DO 101 J=JINIT,JEND
23     DO 100 I=IINIT,IEND
24     C
25     C
26     C
27     UINTW1 = U(I-1,J,K)
28     UINTW2 = V(I-1,J,K)
29     UINTW3 = W(I-1,J,K)
30     C
31     UFLUX(I,J,K,1) = AREA(I,J,K,1,1)*UINTW1
32     *      +AREA(I,J,K,1,2)*UINTW2
33     *      +AREA(I,J,K,1,3)*UINTW3
34     C
35     100 CONTINUE
36     101 CONTINUE
37     102 CONTINUE

```

```

38      RETURN
39      END
40  C
41  C-----
42      SUBROUTINE RHICHWE(IINIT,JINIT,KINIT,IEND,JEND,KEND)
43  C
44  C      INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
45  C      CELLS BY LINEAR INTERPOLATION
46  C
47  C      RHICHW CALCULA EL FLUJO EN LAS CARAS DE LOS
48  C      VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
49  C      LINEAL
50  C
51      INCLUDE 'param.f'
52      COMMON
53      1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
54      1      AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
55      2      SP(NX,NY,NZ)
56      1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
57      2      ,W(NX,NY,NZ)
58      3/VARRC/UFLUX(NX,NY,NZ,3)
59      1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
60      2      VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3)
61      DO 102 K=KINIT,KEND
62      DO 101 J=JINIT,JEND
63      DO 100 I=IINIT,IEND
64  C
65  C-----
66  C
67      UINTW1 = U(I,J,K)
68      UINTW2 = V(I,J,K)
69      UINTW3 = W(I,J,K)
70  C
71      UFLUX(I,J,K,1) = AREA(I,J,K,1,1)*UINTW1
72      *                   +AREA(I,J,K,1,2)*UINTW2
73      *                   +AREA(I,J,K,1,3)*UINTW3
74  C
75      100 CONTINUE
76      101 CONTINUE
77      102 CONTINUE
78  C
79      RETURN
80      END
81  C
82  C-----
83      SUBROUTINE RHICHWS(IINIT,JINIT,KINIT,IEND,JEND,KEND)
84  C
85  C      INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
86  C      CELLS BY LINEAR INTERPOLATION
87  C
88  C      RHICHW CALCULA EL FLUJO EN LAS CARAS DE LOS
89  C      VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
90  C      LINEAL
91  C
92      INCLUDE 'param.f'
93      COMMON
94      1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
95      1      AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
96      2      SP(NX,NY,NZ)
97      1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
98      2      ,W(NX,NY,NZ)
99      3/VARRC/UFLUX(NX,NY,NZ,3)
100     1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
101     2      VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3)
102     DO 102 K=KINIT,KEND
103     DO 101 J=JINIT,JEND
104     DO 100 I=IINIT,IEND
105  C
106  C-----
107  C
108     UINTS1 = U(I,J-1,K)
109     UINTS2 = V(I,J-1,K)
110     UINTS3 = W(I,J-1,K)
111  C
112     UFLUX(I,J,K,2) = AREA(I,J,K,2,1)*UINTS1
113     *                   +AREA(I,J,K,2,2)*UINTS2
114     *                   +AREA(I,J,K,2,3)*UINTS3
115  C
116     100 CONTINUE
117     101 CONTINUE
118     102 CONTINUE
119     RETURN

```

```

120      END
121      C
122      C-----
123      SUBROUTINE RHICHRN(IINIT,JINIT,KINIT,IEND,JEND,KEND)
124      C
125      C INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
126      C CELLS BY LINEAR INTERPOLATION
127      C
128      C RHICHRW CALCULA EL FLUJO EN LAS CARAS DE LOS
129      C VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
130      C LINEAL
131      C
132      INCLUDE 'param.f'
133      COMMON
134      1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
135      1 AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
136      2 SP(NX,NY,NZ)
137      1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
138      2 W(NX,NY,NZ)
139      3/VARRC/UFLUX(NX,NY,NZ,3)
140      1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
141      2 VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3)
142      DO 102 K=KINIT,KEND
143      DO 101 J=JINIT,JEND
144      DO 100 I=IINIT,IEND
145      C
146      C -----
147      C
148      UINTS1 = U(I,J,K)
149      UINTS2 = V(I,J,K)
150      UINTS3 = W(I,J,K)
151      C
152      UFLUX(I,J,K,2) = AREA(I,J,K,2,1)*UINTS1
153      * +AREA(I,J,K,2,2)*UINTS2
154      * +AREA(I,J,K,2,3)*UINTS3
155      C
156      100 CONTINUE
157      101 CONTINUE
158      102 CONTINUE
159      C
160      RETURN
161      END
162      C
163      C-----
164      SUBROUTINE RHICHRB(IINIT,JINIT,KINIT,IEND,JEND,KEND)
165      C
166      C INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
167      C CELLS BY LINEAR INTERPOLATION
168      C
169      C RHICHRW CALCULA EL FLUJO EN LAS CARAS DE LOS
170      C VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
171      C LINEAL
172      C
173      INCLUDE 'param.f'
174      COMMON
175      1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
176      1 AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
177      2 SP(NX,NY,NZ)
178      1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
179      2 W(NX,NY,NZ)
180      3/VARRC/UFLUX(NX,NY,NZ,3)
181      1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
182      2 VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3)
183      DO 102 K=KINIT,KEND
184      DO 101 J=JINIT,JEND
185      DO 100 I=IINIT,IEND
186      C
187      C -----
188      C
189      UINTB1 = U(I,J,K-1)
190      UINTB2 = V(I,J,K-1)
191      UINTB3 = W(I,J,K-1)
192      C
193      UFLUX(I,J,K,3) = AREA(I,J,K,3,1)*UINTB1
194      * +AREA(I,J,K,3,2)*UINTB2
195      * +AREA(I,J,K,3,3)*UINTB3
196      C
197      100 CONTINUE
198      101 CONTINUE
199      102 CONTINUE
200      RETURN
201      END

```

```

202 C
203 C-----
204 SUBROUTINE RHICHTW(IINIT,JINIT,KINIT,IEND,JEND,KEND)
205 C
206 C INTERPOLATION OF VELOCITY TO THE FACES OF CONTROL
207 C CELLS BY LINEAR INTERPOLATION
208 C
209 C RHICHTW CALCULA EL FLUJO EN LAS CARAS DE LOS
210 C VOLUMENES DE CONTROL USANDO EL METODO DE INTERPOLACION
211 C LINEAL
212 C
213 C INCLUDE 'param.f'
214 C COMMON
215 1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
216 1 AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
217 2 SP(NX,NY,NZ)
218 1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
219 2 W(NX,NY,NZ)
220 3/VARRC/UFLUX(NX,NY,NZ,3)
221 1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
222 2 VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3)
223 DO 102 K=KINIT,KEND
224 DO 101 J=JINIT,JEND
225 DO 100 I=IINIT,IEND
226 C
227 C-----
228 C
229 UINTB1 = U(I,J,K)
230 UINTB2 = V(I,J,K)
231 UINTB3 = W(I,J,K)
232 C
233 UFLUX(I,J,K,3) = AREA(I,J,K,3,1)*UINTB1
234 * +AREA(I,J,K,3,2)*UINTB2
235 * +AREA(I,J,K,3,3)*UINTB3
236 C
237 100 CONTINUE
238 101 CONTINUE
239 102 CONTINUE
240 C
241 RETURN
242 END

```

A.4.11 Subrutina CALP.

```

1 C-----
2 SUBROUTINE CALP(IINIT,JINIT,KINIT,IEND,JEND,KEND)
3 C
4 C PRESSURE CORRECTION AS CONTINUITY EQUATION
5 C
6 C AQUI SE CALCULA LA CORRECCION AL VALOR DE LA PRESION
7 C PARA OBTENER NUEVOS VALORES DE P,U,V Y W QUE CUMPLAN
8 C LA ECUACION DE CONSERVACION DE MASA
9 C
10 C
11 C INCLUDE 'param.f'
12 C COMMON
13 1/COEF/AP(NX,NY,NZ,2),AN(NX,NY,NZ),AS(NX,NY,NZ),AE(NX,NY,NZ),
14 1 AW(NX,NY,NZ),AT(NX,NY,NZ),AB(NX,NY,NZ),SU(NX,NY,NZ,3),
15 2 SP(NX,NY,NZ)
16 1/ALL/IT,JT,KT,NI,NJ,NK,NIM1,NJM1,NKM1,NIM2,NJM2,NKM2,GREAT
17 1/VAR/U(NX,NY,NZ),V(NX,NY,NZ),P(NX,NY,NZ),PP(NX,NY,NZ)
18 2 W(NX,NY,NZ)
19 1/PCOR/RESORM,NSWPP,URFP,
20 1 IPREF,JPREF,KPREF
21 1/GGC/AREA(NX,NY,NZ,3,3),GAREA(NX,NY,NZ,3),
22 2 VOL(NX,NY,NZ),WFACT(NX,NY,NZ,3),
23 3 AREAWS(NX,NY,NZ),AREAWB(NX,NY,NZ),AREASB(NX,NY,NZ),
24 4 AREAET(NX,NY,NZ),AREAEN(NX,NY,NZ),AREANT(NX,NY,NZ)
25 1/FLUPR/VISCOS,DENSIT,DEN(NX,NY,NZ)
26 2 VIS(NX,NY,NZ)
27 1/VARRC/UFLUX(NX,NY,NZ,3)
28 1/TRANS/DENO(NX,NY,NZ),APO(NX,NY,NZ)
29 1/TIMEINC/DTIME
30 C
31 RESORM = 0.0
32 C
33 DO 100 K=KINIT,KEND
34 DO 99 J=JINIT,JEND
35 DO 98 I=IINIT,IEND

```

```

36 C
37 C DENSITY IN ELEMENT FACES
38 C
39 C DENSIDAD EN CARAS INTERMEDIAS
40 C
41 DENW = (1.-WFACT(I,J,K,1))*DEN(I,J,K)
42 * + WFACT(I,J,K,1)*DEN(I-1,J,K)
43 DENE = (1.-WFACT(I+1,J,K,1))*DEN(I+1,J,K)
44 * + WFACT(I+1,J,K,1)*DEN(I,J,K)
45 DENS = (1.-WFACT(I,J,K,2))*DEN(I,J,K)
46 * + WFACT(I,J,K,2)*DEN(I,J-1,K)
47 DENN = (1.-WFACT(I,J+1,K,2))*DEN(I,J+1,K)
48 * + WFACT(I,J+1,K,2)*DEN(I,J,K)
49 DENB = (1.-WFACT(I,J,K,3))*DEN(I,J,K)
50 * + WFACT(I,J,K,3)*DEN(I,J,K-1)
51 DENT = (1.-WFACT(I,J,K+1,3))*DEN(I,J,K+1)
52 * + WFACT(I,J,K+1,3)*DEN(I,J,K)
53 C
54 C AP COEFFICIENT IN ELEMENT FACES
55 C
56 C COEFICIENTE PRINCIPAL AP EN CARAS INTERMEDIAS
57 C
58 APW = (1.-WFACT(I,J,K,1))*AP(I,J,K,1)
59 * + WFACT(I,J,K,1)*AP(I-1,J,K,1)
60 APE = (1.-WFACT(I+1,J,K,1))*AP(I+1,J,K,1)
61 * + WFACT(I+1,J,K,1)*AP(I,J,K,1)
62 APS = (1.-WFACT(I,J,K,2))*AP(I,J,K,1)
63 * + WFACT(I,J,K,2)*AP(I,J-1,K,1)
64 APN = (1.-WFACT(I,J+1,K,2))*AP(I,J+1,K,1)
65 * + WFACT(I,J+1,K,2)*AP(I,J,K,1)
66 APB = (1.-WFACT(I,J,K,3))*AP(I,J,K,1)
67 * + WFACT(I,J,K,3)*AP(I,J,K-1,1)
68 APT = (1.-WFACT(I,J,K+1,3))*AP(I,J,K+1,1)
69 * + WFACT(I,J,K+1,3)*AP(I,J,K,1)
70 C
71 C ASSEMBLY OF COEFFICIENTS
72 C
73 C ENSAMBLE DE COEFICIENTES
74 C
75 AW(I,J,K) = DENW*GAREA(I,J,K,1)/APW
76 AS(I,J,K) = DENS*GAREA(I,J,K,2)/APS
77 AB(I,J,K) = DENB*GAREA(I,J,K,3)/APB
78 AE(I,J,K) = DENE*GAREA(I+1,J,K,1)/APE
79 AN(I,J,K) = DENN*GAREA(I,J+1,K,2)/APN
80 AT(I,J,K) = DENT*GAREA(I,J,K+1,3)/APT
81 C
82 C
83 C NON-ORTHOGONALITY SOURCE TERM
84 C
85 C TERMINO FUENTE DEBIDO A LA NO-ORTOGONALIDAD
86 C
87 EN21 = DENN*AREAEN(I,J,K)/APN
88 ES21 = DENS*AREAWS(I,J,K)/APS
89 ET31 = DENT*AREAET(I,J,K)/APT
90 EB31 = DENB*AREAWB(I,J,K)/APB
91 EE12 = DENE*AREAEN(I,J,K)/APE
92 EW12 = DENW*AREAWS(I,J,K)/APW
93 ET32 = DENT*AREANT(I,J,K)/APT
94 EB32 = DENB*AREASB(I,J,K)/APB
95 EE13 = DENE*AREAET(I,J,K)/APE
96 EW13 = DENW*AREAWB(I,J,K)/APW
97 EN23 = DENN*AREANT(I,J,K)/APN
98 ES23 = DENS*AREASB(I,J,K)/APS
99 C
100 C
101 SUP = 0.25*{(PP(I+1,J,K)-PP(I-1,J,K))*(EN21-ES21+ET31-EB31)
102 * + (PP(I,J+1,K)-PP(I,J-1,K))*(EE12-EW12+ET32-EB32)
103 * + (PP(I,J,K+1)-PP(I,J,K-1))*(EE13-EW13+EN23-ES23)
104 * + PP(I+1,J+1,K)*(EN21+EE12)-PP(I+1,J-1,K)*(EE12+ES21)
105 * + PP(I+1,J,K+1)*(EE13+ET31)-PP(I+1,J,K-1)*(EB31+EE13)
106 * + PP(I-1,J-1,K)*(ES21+EW12)-PP(I-1,J+1,K)*(EN21+EW12)
107 * + PP(I-1,J,K-1)*(EB31+EW13)-PP(I-1,J,K+1)*(ET31+EW13)
108 * + PP(I,J+1,K+1)*(ET32+EN23)-PP(I,J-1,K+1)*(ET32+ES23)
109 * + PP(I,J-1,K-1)*(EB32+ES23)-PP(I,J+1,K-1)*(EB32+EN23)}
110 C
111 C
112 C CONVECTION COEFFICIENTS
113 C
114 C COEFICIENTES DE CONVECCION
115 C
116 CW = DENW*UFLUX(I,J,K,1)
117 CE = DENE*UFLUX(I+1,J,K,1)

```

```

118      CS = DENS*UFLUX(I,J,K,2)
119      CN = DENN*UFLUX(I,J+1,K,2)
120      CB = DENB*UFLUX(I,J,K,3)
121      CT = DENT*UFLUX(I,J,K+1,3)
122 C
123 C
124 C      MASS SOURCE TERM
125 C
126 C      CALCULO DEL TERMINO FUENTE DE MASA
127 C
128      DIFDEN=(DEN(I,J,K)-DENO(I,J,K))*VOL(I,J,K)/DTIME
129      SMP=CN-CS+CE-CW+CT-CB+DIFDEN
130 C
131      SU(I,J,K,1) = SUP - SMP
132      CP=AMAX1(0.0,SMP)
133      SP(I,J,K)=-CP
134 C      SP(I,J,K)=0.0
135      RESORM = RESORM + ABS(SMP)
136 C
137      98 CONTINUE
138      99 CONTINUE
139      100 CONTINUE
140 C
141 C
142 C      SPECIAL AND BOUNDARY MODIFICATIONS
143 C
144 C      MODIFICACIONES ESPECIALES
145 C
146      CALL MODP
147      CALL MODAN
148 C
149 C      ASSEMBLY OF FINAL COEFFICIENTS
150 C
151 C      ENSAMBLE FINAL DE COEFICIENTES
152 C
153      DO 301 I=IINIT,IEND
154      DO 301 J=JINIT,JEND
155      DO 301 K=KINIT,KEND
156      AP(I,J,K,2)=AN(I,J,K)+AS(I,J,K)+AE(I,J,K)+AW(I,J,K)+AT(I,J,K)
157      *      +AB(I,J,K)-SP(I,J,K)
158      301 CONTINUE
159 C
160 C      CALL TO MATRIX SOLVER
161 C
162 C      LLAMADO AL INVERSOR DE MATRIZ
163 C
164      DO 400 N=1,NSWPP
165      400 CALL LISOLV(2,2,2,NIM1,NJM1,NKM1,IT,JT,KT,1,PP,2)
166 C
167 C
168 C      PRESSURE AND VELOCITY CORRECTION
169 C
170 C      CORRECCION A LOS VALORES DE PRESION Y VELOCIDAD
171 C
172      CALL MODAP
173 C
174      DO 572 I=IINIT,IEND
175      DO 562 J=JINIT,JEND
176      DO 552 K=KINIT,KEND
177 C
178      APW = (1.-WFACT(I,J,K,1))*AP(I,J,K,1)
179      *      + WFACT(I,J,K,1)*AP(I-1,J,K,1)
180      APE = (1.-WFACT(I+1,J,K,1))*AP(I+1,J,K,1)
181      *      + WFACT(I+1,J,K,1)*AP(I,J,K,1)
182      APS = (1.-WFACT(I,J,K,2))*AP(I,J,K,1)
183      *      + WFACT(I,J,K,2)*AP(I,J-1,K,1)
184      APN = (1.-WFACT(I,J+1,K,2))*AP(I,J+1,K,1)
185      *      + WFACT(I,J+1,K,2)*AP(I,J,K,1)
186      APB = (1.-WFACT(I,J,K,3))*AP(I,J,K,1)
187      *      + WFACT(I,J,K,3)*AP(I,J,K-1,1)
188      APT = (1.-WFACT(I,J,K+1,3))*AP(I,J,K+1,1)
189      *      + WFACT(I,J,K+1,3)*AP(I,J,K,1)
190 C
191      PPREF=PP(IPREF,JPREF,KPREF)
192      P(I,J,K)=P(I,J,K)+URFP*(PP(I,J,K)-PPREF)
193 C
194      UFLUX(I,J,K,1) = UFLUX(I,J,K,1) - GAREA(I,J,K,1)
195      **{(PP(I,J,K)-PP(I-1,J,K))/APW}
196      UFLUX(I,J,K,2) = UFLUX(I,J,K,2) - GAREA(I,J,K,2)
197      **{(PP(I,J,K)-PP(I,J-1,K))/APS}
198      UFLUX(I,J,K,3) = UFLUX(I,J,K,3) - GAREA(I,J,K,3)
199      **{(PP(I,J,K)-PP(I,J,K-1))/APB}

```

```

200 C
201 * U(I,J,K)=U(I,J,K)-0.5*
202 * (AREA(I,J,K,1,1)*(PP(I,J,K)-PP(I-1,J,K))/APW
203 * +AREA(I,J,K,2,1)*(PP(I,J,K)-PP(I,J-1,K))/APS
204 * +AREA(I,J,K,3,1)*(PP(I,J,K)-PP(I,J,K-1))/APB
205 * +AREA(I+1,J,K,1,1)*(PP(I+1,J,K)-PP(I,J,K))/APE
206 * +AREA(I,J+1,K,2,1)*(PP(I,J+1,K)-PP(I,J,K))/APN
207 * +AREA(I,J,K+1,3,1)*(PP(I,J,K+1)-PP(I,J,K))/APT)
208 V(I,J,K)=V(I,J,K)-0.5*
209 * (AREA(I,J,K,1,2)*(PP(I,J,K)-PP(I-1,J,K))/APW
210 * +AREA(I,J,K,2,2)*(PP(I,J,K)-PP(I,J-1,K))/APS
211 * +AREA(I,J,K,3,2)*(PP(I,J,K)-PP(I,J,K-1))/APB
212 * +AREA(I+1,J,K,1,2)*(PP(I+1,J,K)-PP(I,J,K))/APE
213 * +AREA(I,J+1,K,2,2)*(PP(I,J+1,K)-PP(I,J,K))/APN
214 * +AREA(I,J,K+1,3,2)*(PP(I,J,K+1)-PP(I,J,K))/APT)
215 W(I,J,K)=W(I,J,K)-0.5*
216 * (AREA(I,J,K,1,3)*(PP(I,J,K)-PP(I-1,J,K))/APW
217 * +AREA(I,J,K,2,3)*(PP(I,J,K)-PP(I,J-1,K))/APS
218 * +AREA(I,J,K,3,3)*(PP(I,J,K)-PP(I,J,K-1))/APB
219 * +AREA(I+1,J,K,1,3)*(PP(I+1,J,K)-PP(I,J,K))/APE
220 * +AREA(I,J+1,K,2,3)*(PP(I,J+1,K)-PP(I,J,K))/APN
221 * +AREA(I,J,K+1,3,3)*(PP(I,J,K+1)-PP(I,J,K))/APT)
222 552 CONTINUE
223 562 CONTINUE
224 572 CONTINUE
225 CALL RHICHHW(2,2,2,NJM2,NKM2)
226 CALL RHICHWS(2,2,2,NIM2,2,NKM2)
227 CALL RHICHWB(2,2,2,NIM2,NJM2,2)
228 CALL RHICHWE(NIM1,2,2,NIM1,NJM2,NKM2)
229 CALL RHICHWN(2,NJM1,2,NIM2,NJM1,NKM2)
230 CALL RHICHT(2,2,NKM1,NIM2,NJM2,NKM1)
231 CALL MODAP
232 C
233 RETURN
234 END

```

A.4.12 Subrutina MPGSOUT.

```

1 C-----
2      subroutine mpgsout(ttime,istart,jstart,kstart,ni,nj,nk,
3      & u,v,w,scl1,scln1)
4 C
5 C      MPGSOUT CREA ARCHIVOS DE SALIDA PARA LAS VELOCIDADES U,V,W Y
6 C      PARA LA PRESION P CON FORMATO ENSIGHT PARA SU VISUALIZACION
7 C
8
9      implicit none
10     include 'param.f'
11 C      real ttime                !current time
12     integer ttime
13     integer istart,jstart,kstart !first i,j,k points to graph
14     integer ni,nj,nk,nim1,njm1,nkm1 !points to graph
15     real u(ni,nj,nk),v(ni,nj,nk), w(ni,nj,nk) ! the velocities
16     character*4 scln1           !name of the 1st scalar
17     real scl1(ni,nj,nk)        !1st scalar being graphed
18
19     integer itime               ! the integer value of ttime
20     character*12 sclf1          !file name for the 1st scalar
21     character*12 velf           !file name for the velocities
22     character*9 sclhead1       !heading placed at top of
23     !output file for scalar 1
24     integer u1,u10,u100,u1000 !the integers in the 1s,10s,100s
25     !and 1000s position of the time
26     real vect(3,nx,ny,nz)      !contains the u,v,w arrays
27     integer itot, i,j,k, ijk   !work variables
28     itime = int(ttime)
29     nim1 = ni - 1
30     njm1 = nj - 1
31     nkm1 = nk - 1
32 C----- create the files
33     u1 = mod(itime,10)
34     itot = (itime-u1)/10
35     u10 = mod(itot,10)
36     itot = (itot-u10)/10
37     u100 = mod(itot,10)
38     itot = (itot-u100)/10
39     u1000 = mod(itot,10)
40     velf(1:4) = 'velo'
41     write(velf(5:5),(' (11)')) u1000

```

```

42     write(velf(6:6),(' (i1)')) u100
43     write(velf(7:7),(' (i1)')) u10
44     write(velf(8:8),(' (i1)')) u1
45     velf(9:12) = '.vec'
46     open(12,file = velf, status = 'unknown')
47     sclfl(1:4) = sclnl
48     write(sclfl(5:5),(' (i1)')) u1000
49     write(sclfl(6:6),(' (i1)')) u100
50     write(sclfl(7:7),(' (i1)')) u10
51     write(sclfl(8:8),(' (i1)')) u1
52     sclfl(9:12) = '.scl'
53     open(13,file = sclfl,status = 'unknown')
54     do k=kstart,nk
55         do j=jstart,nj
56             do i=istart,ni
57                 vect(1,i,j,k)=u(i,j,k)
58                 vect(2,i,j,k)=v(i,j,k)
59                 vect(3,i,j,k)=w(i,j,k)
60             enddo
61         enddo
62     enddo
63     write(12,3010) *VECTORFIELD*
64     write(12,3030) (((vect(1jk,i,j,k),ijk=1,3),
65 & i=istart,nim1),j=jstart,njm1),k=kstart,nkm1)
66     close(12)
67     sclhead1(1:4)= sclnl
68     sclhead1(5:9)= 'FIELD'
69     write(13,3010) sclhead1
70     write(13,3030) (((scl1(i,j,k),
71 & i=istart,nim1),j=jstart,njm1),k=kstart,nkm1)
72     close(13)
73 3010 format(a20)
74 3030 format(6e12.5)
75     end

```

A.4.13 Subrutina SUSTITU.

```

1 C-----
2     SUBROUTINE SUSTITU
3 C
4 C-----SUBSTITUTION OF THE VALUES **OLD TO NEW**
5 C-----SUSTITUCION DE VALORES **VIEJOS POR NUEVOS**
6     INCLUDE 'param.f'
7     COMMON
8     1/ALL/IT, JT, KT, NI, NJ, NK, NIM1, NJM1, NKM1, NIM2, NJM2, NKM2, GREAT
9     1/VAR/U(NX, NY, NZ), V(NX, NY, NZ), P(NX, NY, NZ), PP(NX, NY, NZ)
10    2     W(NX, NY, NZ)
11    1/FLUPR/VISCOS, DENSIT, DEN(NX, NY, NZ)
12    2     VIS(NX, NY, NZ)
13    1/KASE T1/UIIN, FLOWIN
14    1/TRANS/DENO(NX, NY, NZ), APO(NX, NY, NZ)
15    1/VARTR/UO(NX, NY, NZ), VO(NX, NY, NZ), WO(NX, NY, NZ)
16    1/TIMEINC/DTIME
17    DO 100 I=2, NIM2
18    DO 100 J=2, NJM2
19    DO 100 K=2, NKM2
20    UO(I, J, K)=U(I, J, K)
21    VO(I, J, K)=V(I, J, K)
22    WO(I, J, K)=W(I, J, K)
23    DENO(I, J, K)=DEN(I, J, K)
24 100 CONTINUE
25    RETURN
26    END

```

A.4.14 Subrutina PRINT.

```

1 C-----
2     SUBROUTINE PRINT(ISTART, JSTART, KSTART, NI, NJ, NK, IT, JT, KT, X, Y, Z,
3     PHI, HEAD)
4     INCLUDE 'param.f'
5     DIMENSION PHI(IT, JT, KT), X(IT), Y(JT), Z(KT), STORE(NX)
6     DIMENSION F(7), F4(11)
7     character*12 HEAD
8     DATA F/4H(1H ,4H,A6, ,4H)I3, ,4H)I11 ,4H)I0, ,4H)7X, ,
9     14HA6) /
10    DATA F4/4H 1I ,4H 2I ,4H 3I ,4H 4I ,4H 5I ,4H 6I ,
11    1     4H 7I ,4H 8I ,4H 9I ,4H)I0I ,4H)I11 /
12    DO 50 K=KSTART, NK

```

```

13     ISKIP=1
14     JSKIP=1
15     WRITE(7,110) HEAD,K
16     ISTA=ISTART-12
17     100 CONTINUE
18     ISTA=ISTA+12
19     IEND=ISTA+11
20     IEND=MINO(NI,IEND)
21     F(4)=F4(IEND-ISTA)
22     WRITE(7,112)
23     DO 101 JJ=JSTART,NJ,JSKIP
24     J=JSTART+NJ-JJ
25     DO 120 I=ISTA,IEND
26     A=PHI(I,J,K)
27     IF(ABS(A).LT.1.E-20) A=0.0
28     120 STORE(I)=A
29     101 WRITE(7,113) J,(STORE(I),I=ISTA,IEND,ISKIP)
30     WRITE(7,114) (I,I=ISTA,IEND,ISKIP)
31     IF(IEND.LT.NI)GO TO 100
32     50 CONTINUE
33     RETURN
34     110 FORMAT(1H0,10(2H*-),7X,12X,A12,12X,7X,5(2H-*),
35     $      'LA COORDENADA Z=',I10)
36     111 FORMAT(1H0,6H I = ,I3,11I10,7X)
37     112 FORMAT(3H J)
38     113 FORMAT(1H ,I3,1P12E10.2)
39     114 FORMAT(6H0X= ,I7,11I10)
40     END

```

A.4.15 Archivo de parámetros param.f.

```

1     INTEGER NX,NY,NZ
2     PARAMETER (NX = 31, NY = 15, NZ = 15)

      (para una malla de 31x15x15)

```

VARIABLE EN FORTRAN	SIGNIFICADO	(sin estricto orden alfabético)
(I,J,K)	Índices de cada elemento según las direcciones generalizadas ξ, η, ζ .	
AW(I,J,K) AE(I,J,K) AS(I,J,K) AN(I,J,K) AB(I,J,K) AT(I,J,K)	Coefficientes de matriz para cada volumen de control, ec. (2.49).	
AP(I,J,K,M)	Suma de los coeficientes de matriz en cada volumen de control. Se asocia a las velocidades cuando $M=1$ y a las presiones cuando $M=2$. Ec. (2.52).	
APO(I,J,K)	Coefficiente transiente. Ec. (A.3.6)	
APW, APE APS, APN APB, APT	Interpolación del coeficiente a_p ($AP(I,J,K,M)$) en las caras de los volúmenes de control.	
ARDENT	Área total de la sección de salida multiplicada por la densidad.	
AREA(I,J,K,L,M)	Componente cartesiana ($M: 1=x, 2=y, 3=z$) del vector de área asociado a la cara L en dirección generalizada ($1=w$ oeste, $2=s$ sur, $3=b$ abajo).	
AREAWS(I,J,K) AREAWB(I,J,K) AREASB(I,J,K) AREAET(I,J,K) AREAEN(I,J,K) AREANT(I,J,K)	Productos internos entre vectores de área del mismo elemento, las últimas dos letras indican las caras que son contraídas. Estas son utilizadas en la ecuación (2.38).	
CW, CE CS, CN CB, CT	Coefficientes de convección, ec. (2.46).	
DEN(I,J,K)	Densidad en cada elemento.	
DENO(I,J,K)	Densidad en cada elemento al tiempo anterior.	
DENW, DENE DENS, DENN DENB, DENT	Interpolación de la densidad a las caras de los elementos.	
DENSIT	Densidad característica del fluido.	
DTIME	Δt , cambio en el tiempo.	
DW, DE DS, DN DB, DT	Coefficientes de difusión, ec. (2.47).	
FLOW	Masa por unidad de tiempo y de área que pasa por la sección de salida.	
FLOWIN	Masa por unidad de tiempo y de área que pasa por la sección de entrada.	

GAREA(I,J,K,L)	Cuadrado de la magnitud del vector de área asociado a la cara L en dirección generalizada (1=w oeste, 2=s sur, 3=b abajo).
GREAT	Cota superior al término fuente de masa.
INCALU	Variable lógica para activar el cálculo de la componente U de la velocidad.
INCALV	Variable lógica para activar el cálculo de la componente V de la velocidad.
INCALW	Variable lógica para activar el cálculo de la componente W de la velocidad.
INCALP	Variable lógica para activar el cálculo de la presión.
INPRO	Variable lógica que activa la subrutina de propiedades del fluido.
INPROVAR	Variable lógica que activa la subrutina de actualización de propiedades del fluido.
INTRANS	Variable lógica que activa el análisis transiente.
INCALMPGS	Variable lógica que activa los archivos de salida tipo Ensignht (CEI 1994).
IGEOMCC	Variable lógica que activa la limpia de localidades de memoria.
IGEOMGC	Variable lógica que activa el cálculo de las propiedades de la malla.
INDPRI	Control de impresión en cada iteración.
INDPRGR	Control de impresión en cada paso de tiempo.
IPREF	
JPREF	
KPREF	Indíces de las coordenadas de la presión de referencia.
MAXIT	Número máximo de iteraciones en cada paso de tiempo.
NI, NJ, NK	
NX, NY, NZ	Número de puntos en las direcciones ξ, η, ζ .
NIM1, NJM1, NKM1	Número de puntos en las direcciones ξ, η, ζ menos uno.
NIM2, NJM2, NKM2	Número de puntos en las direcciones ξ, η, ζ menos dos.
NITER	Conteo de las iteraciones dentro del ciclo SIMPLE (sección 2.3.2).
NTSTEP	Número de pasos en el tiempo.
NSWPU	
NSWPV	
NSWPW	
NSWPP	Número de iteraciones dentro del inversor de matriz para cada componente de la velocidad y para la corrección de presión.
P(I,J,K)	Presión en cada volumen de control.
PP(I,J,K)	Corrección de presión p' , ec. (2.65).
PPREF	Presión de referencia.
RESORU	
RESORV	
RESORW	Residuo de las ecuaciones de cantidad de movimiento (para cada componente U, V y W).
RESORM	Residuo de la ecuación de conservación de masa.
SORCE	Escoge el máximo de los residuos.
SORMAX	Parámetro de convergencia.

SP(I,J,K)	Término fuente de las ecuaciones (2.52) y (2.62).
SU(I,J,K,M)	Término fuente de las ecuaciones (2.52) y (2.62) para cada componente M de la velocidad.
SUNO SVNO SWNO	Términos fuente no ortogonales, ec. (2.50), para cada componente de la velocidad.
SUP	Término no ortogonal para la ecuación de corrección de presión, ec. (2.72).
TTIME	Tiempo actual de cálculo.
U(I,J,K)	Componente cartesiana x de la velocidad.
V(I,J,K)	Componente cartesiana y de la velocidad.
W(I,J,K)	Componente cartesiana z de la velocidad.
UALIT(I,J,K)	Campo inicial de velocidades.
UFLUX(I,J,K,M)	Componentes normales de flujo, ec. (2.23) y (2.35), para cada cara M del elemento (1=w oeste, 2=s sur, 3=b abajo).
UIN	Velocidad inicial constante.
UO(I,J,K)	Componente x de la velocidad al tiempo anterior.
VO(I,J,K)	Componente y de la velocidad al tiempo anterior.
WO(I,J,K)	Componente z de la velocidad al tiempo anterior.
URFU URFV URFW URFP	Factor de atenuación (<i>under-relaxation</i>) para el coeficiente AP de velocidades y para la corrección de presión.
VIS(I,J,K)	Viscosidad en cada elemento.
VISCOS	Viscosidad general del fluido.
VISW, VISE VISS, VISN VISB, VIST	Interpolación de la viscosidad a las caras del volumen de control.
VOL(I,J,K)	Volumen de cada elemento o célula de control, ec. (2.25).
WFACT(I,J,K,M)	Factor de peso en la interpolación lineal para cada dirección generalizada M, (1=w oeste, 2=s sur, 3=b abajo), ec. (2.26).
X(I,J,K) Y(I,J,K) Z(I,J,K)	Coordenadas cartesianas de los centroides de los elementos.
XC(I,J,K) YC(I,J,K) ZC(I,J,K)	Coordenadas cartesianas de los vértices de los elementos.
YW(J,K), YE(J,K) YS(I,K), YN(I,K) YB(I,J), YT(I,J)	Distancias entre centroide y pared para los elementos en la frontera.

Tabla A.4.1. Variables principales del programa NECRONOM.f.

Apéndice 5.

Listado de los programas generadores de mallas.

A.5.1 Programa MALLAS.f

```

1 C
2 C
3 C MALLAS APLICA UNA TECNICA MODIFICADA DE (4) MULTISUPERFICIES
4 C PARA GENERAR UNA CUADRICULA EN COORDENADAS CURVILINEAS DADAS
5 C DOS CURVAS FRONTERA.
6 C
7 C
8 C NOTA: LA MALLA NECESITARA UN NY IMPAR POR RAZONES DE SIMETRIA
9 C
10 C INCLUDE 'param.f'
11 C DIMENSION X(NX,NY,NZ),Y(NX,NY,NZ),XB(4),YB(4),XS(4,NX),YS(4,NX)
12 C DIMENSION RAB(NX),RDC(NX),SBC(NY),SAD(NY),SH(4),XA(51),RA(51)
13 C * ,XA1(51),RA1(51),Z(NX,NY,NZ)
14 C COMMON /ENSIGH/X,Y,Z
15 C COMMON /RAXAS/RA,XA,RA1,XA1
16 C
17 C OPEN(1,FILE='MALLA.DAT')
18 C OPEN(7,FILE='MALLA3D.DAT')
19 C READ(1,1)S2,S3,AW
20 C 1 FORMAT(3E10.3)
21 C READ(1,2)PAB,QAB,PDC,QDC,PAD,QAD,PBC,QBC
22 C 2 FORMAT(8E10.3)
23 C NXM2 = NX -2
24 C NYM2 = INT(0.5*(NY-1.))
25 C
26 C DEFINE CORNER POINTS OF THE BOUNDARY
27 C
28 C DEFINIR PUNTOS EN LAS ESQUINAS DE LA FRONTERA
29 C
30 C ZMA = 2.0
31 C DATA XB/-1.00,5.00,5.00,-1.00/
32 C DATA YB/-1.00,-2.00,0.00,0.00/
33 C
34 C
35 C GENERATE STRETCHING FUNCTIONS
36 C
37 C GENERAR FUNCIONES DE DEFORMACION
38 C
39 C CALL STRECH(NXM2,PAB,QAB,RAB)
40 C
41 C CALL STRECH(NXM2,PDC,QDC,RDC)
42 C
43 C CALL STRECH(NYM2,PAD,QAD,SAD)
44 C
45 C CALL STRECH(NYM2,PBC,QBC,SBC)
46 C
47 C
48 C OBTAIN SURFACE COORDINATES OF BOUNDARIES
49 C
50 C OBTENER COORDENADAS SUPERFICIALES DE LAS FRONTERAS
51 C
52 C CALL BOUNDSURF(0,R1AB,R1DC,XD,YD,XE,YE)
53 C
54 C
55 C GENERATE BOUNDING SURFACES 1 AND 4
56 C
57 C GENERAR SUPERFICIES FRONTERA 1 Y 4
58 C
59 C DO 17 J = 1,NXM2
60 C
61 C RABD = RAB(J)*R1AB
62 C
63 C CALL BOUNDSURF(1,RABD,RDCD,XD,YD,XE,YE)
64 C XS(1,J) = XD*(XB(2)-XB(1)) + XB(1)
65 C YS(1,J) = YD + YB(1)
66 C XS(1,J) = XB(1) + RABD*(XB(2)-XB(1))/R1AB
67 C YS(1,J) = YB(1)
68 C
69 C RDCD = RDC(J)*R1DC

```

```

70 C
71 CALL BOUNDSURF(2,RABD,RDCD,XD,YD,XE,YE)
72 XS(4,J) = XE*(XB(3)-XB(4)) + XB(4)
73 YS(4,J) = YE + YB(4)
74 C XS(4,J) = XB(4) + RDCD*(XB(3)-XB(4))/R1DC
75 C YS(4,J) = YB(4)
76 17 CONTINUE
77 C
78 C
79 C SURCH GENERATES SURFACES 2 AND 3 SO THAT THE GRID
80 C ADJACENT TO SURFACES 1 AND 4 IS ORTHOGONAL
81 C
82 C SURCH GENERA LAS SUPERFICIES INTERMEDIAS 2 Y 3 DE
83 C TAL FORMA QUE LA MALLA SEA LOCALMENTE ORTOGONAL
84 C EN LAS FRONTERAS
85 C
86 CALL SURCH(NXM2,S2,S3,XS,YS)
87 C
88 C
89 C
90 C GENERATE INTERIOR GRID
91 C
92 C GENERAR MALLA INTERIOR
93 C
94 A1 = 2./(3.*AW-1.)
95 A2 = 2./(3.*(1.-AW) -1.)
96 AJM = NXM2 - 1
97 DZI = 1./AJM
98 DO 24 K = 1,NYM2
99 DO 23 J = 1,NXM2
100 NJ1 = J + 1
101 NK1 = K + 1
102 AJ = J - 1
103 ZI = AJ*DZI
104 S = SAD(K) + ZI*(SBC(K)-SAD(K))
105 SH(1) = (1.-S)**2*(1.-A1*S)
106 SH(2) = (1.-S)**2*S*(A1+2.)
107 SH(3) = (1.-S)*S*S*(A2+2.)
108 SH(4) = S*S*(1.-A2*(1.-S))
109 X(NJ1,NK1,1) = 0.
110 Y(NJ1,NK1,1) = 0.
111 DO 22 L = 1,4
112 X(NJ1,NK1,1) = X(NJ1,NK1,1) + SH(L)*XS(L,J)
113 Y(NJ1,NK1,1) = Y(NJ1,NK1,1) + SH(L)*YS(L,J)
114 22 CONTINUE
115 23 CONTINUE
116 24 CONTINUE
117 NXM1 = NX - 1
118 NYM1 = NY - 1
119 DO 430 J=2,NXM1
120 X(J,1,1) = 2.*X(J,2,1)-X(J,3,1)
121 Y(J,1,1) = 2.*Y(J,2,1)-Y(J,3,1)
122 430 CONTINUE
123 DO 440 K=1,NYM2+1
124 X(1,K,1) = 2.*X(2,K,1)-X(3,K,1)
125 Y(1,K,1) = 2.*Y(2,K,1)-Y(3,K,1)
126 X(NX,K,1) = 2.*X(NXM1,K,1)-X(NXM2,K,1)
127 Y(NX,K,1) = 2.*Y(NXM1,K,1)-Y(NXM2,K,1)
128 440 CONTINUE
129 C
130 C
131 C TRASLATE THE GRID TO THE X-AXIS
132 C
133 C TRASLADO DE LA MALLA AL EJE X
134 C
135 C DO 610 J=1,NX
136 C DO 600 K=1,NYM2+1
137 C Y(J,K,1)=Y(J,K,1)-YB(4)
138 C 600 CONTINUE
139 C 610 CONTINUE
140 C
141 C REFLECT GRID ABOUT X-AXIS
142 C
143 C REFLEJAR LA MALLA SOBRE EL EJE X
144 C
145 C DO 27 K = 1,NYM2+1
146 C DO 25 J = 1,NX
147 C KA = NY + 1 - K
148 C X(J,KA,1) = X(J,K,1)
149 C Y(J,KA,1) = -Y(J,K,1)
150 C 25 CONTINUE
151 C 27 CONTINUE

```

```

152      DO 650 L=1,NZ
153      DO 660 K=1,NY
154      DO 670 J=1,NX
155      X(J,K,L) = X(J,K,1)
156      Y(J,K,L) = Y(J,K,1)
157      Z(J,K,L) = (L-1)*ZMA/(NZ-2)
158      670 CONTINUE
159      660 CONTINUE
160      650 CONTINUE
161  C
162      28 CALL ENSIGHT(NX,NY,NZ)
163      STOP
164      END

```

A.5.2 Subrutina BOUNDSURF.

```

1  C-----
2      SUBROUTINE BOUNDSURF(INT, RAER, RDCR, XD, YD, XE, YE)
3  C
4  C      GENERATES TWO BOUNDARIES WITH CURVES Y=F(X)
5  C      GENERA DOS FRONTERAS DADAS COMO CURVAS Y=F(X)
6  C      IF INT = 0,  NUMERICALLY INTEGRATE TO OBTAIN SURFACE
7  C      COORDINATE
8  C      IF INT = 1,  INTERPOLATE SURFACE 1 COORDINATES TO OBTAIN
9  C      CORRESPONDING (X,Y)
10 C      IF INT =2,  INTERPOLATE SURFACE 2 COORDINATES TO OBTAIN
11 C      CORRESPONDING (X,Y)
12 C
13      DIMENSION XA(51),RA(51),XA1(51),RA1(51)
14      COMMON /RAXAS/RA, XA, RA1, XA1
15      E = 1./(EXP(4.)+1.)
16      IF(INT .EQ. 1)GOTO 2
17      IF(INT .EQ. 2)GOTO 4
18  C
19  C      NUMERICALLY INTEGRATE TO OBTAIN RA(L) AS A FUNCTION OF XA(L)
20  C      INTEGRAR NUMERICAMENTE PARA OBTENER RA(L) COMO FUNCION DE XA(L)
21  C
22      RA(1) = 0.
23      XA(1) = -1.0
24      RA1(1) = 0.
25      XA1(1) = -1.0
26      DUM = -20./(EXP(4.-20.*XA(1))+EXP(-4.+20.*XA(1))+2.)
27      FLP = SQRT(1. + DUM*DUM)
28      FLP1 = 1.
29      DO 1 L = 1,50
30      AL = L
31      LP = L + 1
32      XA(LP) = 0.04*AL + XA(1)
33      XA1(LP) = 0.04*AL + XA1(1)
34      DX = XA(LP) - XA(L)
35      FL = FLP.
36      FL1 = FLP1
37      DUM = -20./(EXP(4.-20.*XA(LP))+EXP(-4.+20.*XA(LP))+2.)
38      FLP = SQRT(1. + DUM*DUM)
39      FLP1 = 1.
40      RA(LP) = RA(L) + 0.5*(FL + FLP)*DX
41      RA1(LP) = RA1(L) + 0.5*(FL1 + FLP1)*DX
42  1 CONTINUE
43      RABR= RA(51)
44      RDCR= RA1(51)
45      RETURN
46  C
47  C      INTERPOLATE RA(L) TO OBTAIN X CORRESPONDING TO RAER
48  C      SUBSEQUENTLY OBTAIN Y FROM ANALYTIC Y=F(X)
49  C
50  C      INTERPOLAR RA(L) OBTENIENDO X PARA RABR DADO Y
51  C      SUBSECUENTEMENTE OBTENER Y DE Y=F(X)
52  C
53  2 DO 3 L = 2,51
54      IF(RABR .GT. RA(L))GOTO 3
55      LM = L - 1
56      XD = XA(LM) + (XA(L)-XA(LM))*(RABR-RA(LM))/(RA(L)-RA(LM))
57  C      IF(XD .LT. 1.0E-06)XD=1.0E-06
58      YD = E - (1./(EXP(4.-20.*XD)+1))
59      RETURN
60  3 CONTINUE
61      RETURN
62  4 DO 5 L = 2,51

```

```

63     IF(RDCR .GT. RA1(L))GOTO 5
64     LM = L - 1
65     XE = XA1(LM)+(XA1(L)-XA1(LM))*(RDCR-RA1(LM))/(RA1(L)-RA1(LM))
66 C   IF(XE .LT. 1.0E-06)XE=1.0E-06
67     YE = 0.0
68     RETURN
69     5 CONTINUE
70     RETURN
71     END

```

A.5.3 Subrutina STRECH.

```

1 C-----
2   SUBROUTINE STRECH(N,P,Q,S)
3 C
4 C   COMPUTES ONE-DIMENSIONAL STRETCHING FUNCTION,
5 C   S = P*ETA + (1.-TANH(Q*(1.-ETA))/TANH(Q)),
6 C   FOR GIVEN CONTROL PARAMETERS, P AND Q.
7 C
8 C   CALCULA LA FUNCION DE DEFORMACION PARA PARAMETROS
9 C   DE CONTROL P Y Q DADOS
10 C
11   INCLUDE 'param.f'
12   DIMENSION S(NX)
13   AN = N-1
14   DELTA = 1./AN
15   TQI = 1./TANH(Q)
16 C
17   DO 1 L = 1,N
18     AL = L - 1
19     ETA = AL*DELTA
20     DUM = Q*(1. - ETA)
21     DUM = 1. - TANH(DUM)*TQI
22     S(L) = P*ETA + (1.-P)*DUM
23   1 CONTINUE
24   RETURN
25   END

```

A.5.4 Subrutina SURCH.

```

1 C-----
2   SUBROUTINE SURCH(JMAX,S2,S3,XS,YS)
3 C
4 C   GENERATES SURFACES 2 AND 3 TO CREATE ORTHOGONAL
5 C   BOUNDARY GRIDS
6 C   GENERA LAS SUPERFICIES INTERMEDIAS 2 Y 3 PARA ASEGURAR
7 C   ORTOGONALIDAD LOCAL EN LAS FRONTERAS
8 C
9   INCLUDE 'param.f'
10  DIMENSION XS2(NX),YS2(NX),XS3(NX),YS3(NX),XS(4,NX),YS(4,NX)
11  JMAP = JMAX - 1
12 C
13 C   PRELIMINARY GENERATION OF SURFACES 2 AND 3
14 C   GENERACION PRELIMINAR DE LAS SUPERFICIES 2 Y 3
15 C
16   DO 1 J = 1,JMAX
17     DXS = XS(4,J) - XS(1,J)
18     DYS = YS(4,J) - YS(1,J)
19     XS(2,J) = XS(1,J) + S2*DXS
20     YS(2,J) = YS(1,J) + S2*DYS
21     XS(3,J) = XS(1,J) + S3*DXS
22     YS(3,J) = YS(1,J) + S3*DYS
23   1 CONTINUE
24 C
25 C   PROJECT ORTHOGONALLY FROM SURFACE 1 ONTO SURFACE 2
26 C   PROYECCION ORTOGONAL DE LA SUPERFICIE 1 A LA 2
27 C
28   DO 9 J = 2,JMAP
29     IF(ABS(XS(1,J+1)-XS(1,J-1)) .GT. 1.0E-06)GOTO 2
30     EM1 = 1.0E06*(YS(1,J+1)-YS(1,J-1))
31     GOTO 3
32   2 EM1 = (YS(1,J+1)-YS(1,J-1))/(XS(1,J+1)-XS(1,J-1))
33   3 IF(ABS(XS(2,J)-XS(2,J-1)) .GT. 1.0E-06)GOTO 4
34     EM2 = 1.0E+06*(YS(2,J)-YS(2,J-1))
35     GOTO 5

```

```

36      4 EM2 = (YS(2,J)-YS(2,J-1))/(XS(2,J)-XS(2,J-1))
37      5 X2 = (EM1*(YS(1,J)-YS(2,J)+EM2*XS(2,J))+XS(1,J))/(1.+EM1*EM2)
38      Y2 = YS(2,J) + EM2*(X2 - XS(2,J))
39      STJM = SQRT((X2-XS(2,J-1))**2 + (Y2-YS(2,J-1))**2)
40      SJJM = SQRT((XS(2,J)-XS(2,J-1))**2 + (YS(2,J)-YS(2,J-1))**2)
41      IF(STJM.LT. SJJM)GOTO 8
42      IF(ABS(XS(2,J+1)-XS(2,J)) .GT. 1.0E-06)GOTO 6
43      EM2 = 1.0E+06*(YS(2,J+1)-YS(2,J))
44      GOTO 7
45      6 EM2 = (YS(2,J+1)-YS(2,J))/(XS(2,J+1)-XS(2,J))
46      7 X2 = (EM1*(YS(1,J)-YS(2,J)+EM2*XS(2,J))+XS(1,J))/(1.+EM1*EM2)
47      Y2 = YS(2,J) + EM2*(X2-XS(2,J))
48      8 XS2(J) = X2
49      YS2(J) = Y2
50      9 CONTINUE
51 C
52 C      PROJECT ORTHOGONALLY FROM SURFACE 4 ONTO SURFACE 3
53 C      PROYECCION ORTOGONAL DE LA SUPERFICIE 4 A LA 3
54 C
55      DO 17 J = 2,JMAP
56      IF(ABS(XS(4,J+1)-XS(4,J-1)) .GT. 1.0E-06)GOTO 10
57      EM4 = 1.0E+06*(YS(4,J+1)-YS(4,J-1))
58      GOTO 11
59      10 EM4 = (YS(4,J+1) - YS(4,J-1))/(XS(4,J+1) - XS(4,J-1))
60      11 IF(ABS(XS(3,J) - XS(3,J-1)) .GT. 1.0E-06)GOTO 12
61      EM3 = 1.0E+06*(YS(3,J)-YS(3,J-1))
62      GOTO 13
63      12 EM3 = (YS(3,J)-YS(3,J-1))/(XS(3,J)-XS(3,J-1))
64      13 X3 = (EM4*(YS(4,J)-YS(3,J)+EM3*XS(3,J))+XS(4,J))/(1.+EM3*EM4)
65      Y3 = YS(3,J) + EM3*(X3 - XS(3,J))
66      STJM = SQRT((X3-XS(3,J-1))**2 + (Y3-YS(3,J-1))**2)
67      SJJM = SQRT((XS(3,J)-XS(3,J-1))**2 + (YS(3,J)-YS(3,J-1))**2)
68      IF(STJM.LT. SJJM)GOTO 16
69      IF(ABS(XS(3,J+1)-XS(3,J)) .GT. 1.0E-06)GOTO 14
70      EM3 = 1.0E+06*(YS(3,J+1)-YS(3,J))
71      GOTO 15
72      14 EM3 = (YS(3,J+1)-YS(3,J))/(XS(3,J+1)-XS(3,J))
73      15 X3 = (EM4*(YS(4,J)-YS(3,J)+EM3*XS(3,J))+XS(4,J))/(1.+EM3*EM4)
74      Y3 = YS(3,J) + EM3*(X3 - XS(3,J))
75      16 XS3(J) = X3
76      YS3(J) = Y3
77      17 CONTINUE
78 C
79 C      STORE SURFACE 2 AND 3 LOCATIONS
80 C      GUARDAR COORDENADAS EN SUPERFICIES 2 Y 3
81 C
82      DO 18 J = 2,JMAP
83      XS(2,J) = XS2(J)
84      YS(2,J) = YS2(J)
85      XS(3,J) = XS3(J)
86      YS(3,J) = YS3(J)
87      18 CONTINUE
88      RETURN
89      END

```

A.5.5 Subrutina ENSIGHT.

```

1 C-----
2      SUBROUTINE ENSIGHT(JMAX,KMAX,LMAX)
3 C      ENSIGHT CREATES FILE WITH ENSIGHT DATA FILE FORMAT
4 C
5 C      ENSIGHT CREA UN ARCHIVO DE COORDENADAS EN FORMATO ENSIGHT
6 C
7      INCLUDE 'param.f'
8      DIMENSION X(NX,NY,NZ), Y(NX,NY,NZ), Z(NX,NY,NZ)
9      COMMON /ENSIGH/X,Y,Z
10 C
11      OPEN(7,FILE='MALLA3D.DAT')
12      WRITE(7,1)
13      1 FORMAT('GENERACION DE UN PARALELEPIPEDO')
14      WRITE(7,2)
15      2 FORMAT(' ')
16      WRITE(7,3)
17      3 FORMAT('node id given')
18      WRITE(7,4)
19      4 FORMAT('element id given')
20      WRITE(7,5)
21      5 FORMAT('coordinates')
22      NUPO=KMAX*JMAX*LMAX

```

```

23      WRITE(7,6)NUP0
24      6 FORMAT(18)
25      IDN = 0
26      DO 9 L=1,LMAX
27      DO 9 K=1,KMAX
28      DO 8 J=1,JMAX
29      IDN = IDN + 1
30      WRITE(7,7)IDN,X(J,K,L),Y(J,K,L),Z(J,K,L)
31      7 FORMAT(18,3E12.5)
32      8 CONTINUE
33      9 CONTINUE
34      C
35      C      CONECTIVIDAD
36      C
37      WRITE(7,10)
38      10 FORMAT('part 1',/, 'conectividad',/, 'hexa8')
39      NUNO = (JMAX-1)*(KMAX-1)*(LMAX-1)
40      IDM = 0
41      IDO = 0
42      WRITE(7,11)NUNO
43      11 FORMAT(18)
44      DO 15 L=1,LMAX-1
45      DO 14 K=1,KMAX
46      DO 13 J=1,JMAX
47      IDM = IDM + 1
48      JMAS = IDM + 1
49      IVEC = IDM + JMAX
50      IVED = JMAS + JMAX
51      IVES = IDM + (JMAX*KMAX)
52      IVET = IVES + 1
53      IVEU = IVES + JMAX
54      IVEV = IVET + JMAX
55      IF(J .EQ. JMAX)GOTO 13
56      IF(K .EQ. KMAX)GOTO 13
57      IDO = IDO + 1
58      WRITE(7,12)IDO, IDM, JMAS, IVED, IVEC, IVES, IVET, IVEV, IVEU
59      12 FORMAT(9I8)
60      13 CONTINUE
61      14 CONTINUE
62      15 CONTINUE
63      RETURN
64      C
65      END

```

PARAMETRO	SIGNIFICADO
NX, NY, NZ	No. de puntos en las direcciones ξ, η, ζ .
PAB, QAB PDC, QDC PAD, QAD PBC, QBC	Parámetros P y Q en la función de deformación (4.2) para cada frontera.

Las funciones que definen a las fronteras AB y CD pueden ser modificadas manualmente en la subrutina BOUNDSURF.

Tabla A.5.1. Parámetros de entrada del programa MALLAS.f.

A.5.6 Programa TUBOTE.f.

```

1  C      *****
2  C      MALLADOR DE TUBOS CON CODOS CIRCULARES O EN FORMA DE S
3  C      *****
4  C
5  C      INCLUDE 'param.f'
6  C
7  C      DIMENSION X(NX,NY,NZ), Y(NX,NY,NZ), Z(NX,NY,NZ),
8  C      *      XA(NX,NY,NZ), YA(NX,NY,NZ)
9  C      DIMENSION SAD(NY)
10 C      COMMON /ENSIGH/X,Y,Z
11 C

```

```

12 C NREFLEX = 1 :TUBO CON INICIO CURVO
13 C NREFLEX = 2 :TUBO CON CODO EN LA MITAD
14 C NREFLEX = 3 :TUBO EN FORMA DE S
15 C
16 C ****IMPORTANTE**** NX DEBE SER IMPAR
17 C
18 NREFLEX = 2
19 PI = 3.1415927
20 RMAX = 0.5
21 RMIN = 0.4
22 TMAX = 3.0*PI/2.0
23 TMIN = 5.0*PI/4.0
24 XMAX = 1.0
25 YMAX = 0.1
26 ZMAX = 0.1
27 PAD = 0.1
28 QAD = 2.0
29 PM = COS(TMAX-TMIN)/SIN(TMAX-TMIN)
30 PN = -SIN(TMAX-TMIN)/COS(TMAX-TMIN)
31 B = ((RMAX-RMIN)/2.0 +RMIN)*(COS(TMIN)/PM +SIN(TMIN))+RMAX
32 C
33 C GENERAR FUNCIONES DE DEFORMACION
34 C
35 CALL STRECH(NY, PAD, QAD, SAD)
36 C
37 IF (NREFLEX .EQ. 2)GOTO 10
38 IF (NREFLEX .EQ. 3)GOTO 10
39 C NXM = INT(NX/2)
40 NXM = 1
41 DO 1 I=1, NXM
42 DO 2 J=1, NY
43 DO 3 K=1, NZ
44 X(I, J, K) = (RMAX - (J-1) * (RMAX-RMIN) / (NY-2)) * COS(TMIN + (I-1)
45 * (TMAX-TMIN) / (NXM-1))
46 Y(I, J, K) = (RMAX - (J-1) * (RMAX-RMIN) / (NY-2)) * SIN(TMIN + (I-1)
47 * (TMAX-TMIN) / (NXM-1)) + RMAX
48 Z(I, J, K) = (K-1) * ZMAX / (NZ-2)
49 3 CONTINUE
50 2 CONTINUE
51 1 CONTINUE
52 DO 4 I=NXM, NX
53 DO 5 J=1, NY
54 DO 6 K=1, NZ
55 X(I, J, K) = (I-NXM) * XMAX / (NX-2)
56 C Y(I, J, K) = (J-1) * YMAX / (NY-2)
57 Y(I, J, K) = SAD(J) * YMAX
58 Z(I, J, K) = (K-1) * ZMAX / (NZ-2)
59 6 CONTINUE
60 5 CONTINUE
61 4 CONTINUE
62 GOTO 100
63 10 NXC = INT(3*NX/4)
64 NXM = INT(NX/2)
65 DO 12 I=NXM+1, NXC
66 DO 14 J=1, NY
67 DO 16 K=1, NZ
68 C X(I, J, K) = (RMAX - (J-1) * (RMAX-RMIN) / (NY-2)) * COS(TMIN + (I-NXM-1)
69 C * (TMAX-TMIN) / (NXC-NXM-1))
70 C Y(I, J, K) = (RMAX - (J-1) * (RMAX-RMIN) / (NY-2)) * SIN(TMIN + (I-NXM-1)
71 C * (TMAX-TMIN) / (NXC-NXM-1)) + RMAX
72 X(I, J, K) = (RMAX - SAD(J) * (RMAX-RMIN)) * COS(TMIN + (I-NXM-1)
73 * (TMAX-TMIN) / (NXC-NXM-1))
74 Y(I, J, K) = (RMAX - SAD(J) * (RMAX-RMIN)) * SIN(TMIN + (I-NXM-1)
75 * (TMAX-TMIN) / (NXC-NXM-1)) + RMAX
76 Z(I, J, K) = (K-1) * ZMAX / (NZ-2)
77 16 CONTINUE
78 14 CONTINUE
79 12 CONTINUE
80 DO 18 I=NXC, NX
81 DO 20 J=1, NY
82 DO 22 K=1, NZ
83 X(I, J, K) = (I-NXC) * XMAX / (NX-1-NXC)
84 Y(I, J, K) = SAD(J) * YMAX
85 Z(I, J, K) = (K-1) * ZMAX / (NZ-2)
86 22 CONTINUE
87 20 CONTINUE
88 18 CONTINUE
89 DO 24 I=1, NXM
90 DO 26 J=1, NY
91 DO 28 K=1, NZ
92 NXF1=NX+1-I
93 X(I, J, K) = X(NXF1, J, K) + (2.0*PM / (1.0 + (PM)**2.0)) * (Y(NXF1, J, K)

```

```

94      *          -PM*X(NXF1,J,K)-RMAX)
95      Y(I,J,K) = Y(NXF1,J,K) +(2.0/(1.0+(PM)**2.0))*(RMAX
96      *          +PM*X(NXF1,J,K)-Y(NXF1,J,K))
97      Z(I,J,K) = Z(NXF1,J,K)
98      28 CONTINUE
99      26 CONTINUE
100     24 CONTINUE
101     IF (NREFLEX .EQ. 2)GOTO 100
102     DO 30 I=1,NXM
103     DO 32 J=1,NY
104     DO 34 K=1,NZ
105     NYF1 = NY+1-J
106     XA(I,J,K) = X(I,NYF1,K) +(2.0*PN/(1.0+(PN)**2.0))*(Y(I,NYF1,K)
107     *          -PN*X(I,NYF1,K)-B)
108     YA(I,J,K) = Y(I,NYF1,K) +(2.0/(1.0+(PN)**2.0))*(B
109     *          +PN*X(I,NYF1,K)-Y(I,NYF1,K))
110     34 CONTINUE
111     32 CONTINUE
112     30 CONTINUE
113     DO 36 I=1,NXM
114     DO 38 J=1,NY
115     DO 40 K=1,NZ
116     X(I,J,K) = XA(I,J,K)
117     Y(I,J,K) = YA(I,J,K)
118     40 CONTINUE
119     38 CONTINUE
120     36 CONTINUE
121     100 CALL ENSIGHT(NX,NY,NZ)
122     STOP
123     END

```

PARÁMETRO SIGNIFICADO

NX, NY, NZ	No. de puntos en las direcciones ξ, η, ζ .
PAD, QAD	Parámetros P y Q en la función de deformación (4.2) para la frontera en dirección η .
NREFLEX	1) Tubo con inicio curvo, 2) Tubo con codo en la mitad, 3) Tubo en forma de s.
RMAX	Radio máximo de curvatura.
RMIN	Radio mínimo de curvatura.
TMAX	Ángulo máximo de sección circular
TMIN	Ángulo mínimo de sección circular
XMAX	Longitud del tramo recto
YMAX	Dimensión de la sección en dirección η .
ZMAX	Dimensión de la sección en dirección ζ (que es lo mismo que la dirección z).

Tabla A.5.2. Parámetros de entrada del programa TUBOTE.f.

Bibliografía.

Aris, R., 1962, "Vector, Tensors, and the Basic Equations of Fluid Mechanics", Prentice-Hall, Englewood Cliffs, New Jersey.

Batchelor, G.K., 1967, "An Introduction to Fluid Dynamics", Cambridge University Press.

Benmalek, A., 1993, "Nonlinear Development of Görtler Vortices over Variable Curvature Walls", Ph.D. thesis, Ariz. State Univ.

Brundrett, E., and Baines, W.D., 1964, *J. Fluid Mech.* **19**, 375-392.

Burns, A.D., and Wilkes, N.S., 1987, "A Finite Difference Method for the Computation of Fluid Flows in Complex Three Dimensional Geometries", AERE - R 12342.

CEI, 1994, "Ensignt User Manual for Version 5.2", Computational Engineering International, Inc., Research Triangle Park, NC.

Constantin, P., and Foias, C., "Navier-Stokes Equations", The University of Chicago Press.

Currie, I.G., 1993, "Fundamental Mechanics of Fluids", Mc Graw Hill, New York.

Chapman, W., 1979, *AIAA J.* **17**, 1293-1313.

Demirdzic, I.A., 1982, "A Finite Volume Method for Computation of Fluid Flow in Complex Geometries", Ph.D. thesis, University of London, Imperial College.

Doering C.R., and Gibbon, J.D., 1995, "Applied Analysis of the Navier-Stokes Equations", Cambridge University Press.

Eiseman, P.R., 1979, *J. Comput. Phys.* **33**, 118-150.

Finlayson, B.A., 1972, "The Method of Weighted Residuals and Variational Principles", Academic, New York.

Fletcher, C.A.J., 1991, "Computational Techniques for Fluid Dynamics", Vol I and Vol II, Springer-Verlag, Berlin Heidelberg.

Görtler, H., 1941, *ZAMM* **21**, 250-252.

Humphrey, A.C., Taylor A.M.K., and Whitelaw, J.H., 1977, *J. Fluid Mech.*, **83**, part 3, 509-527.

Karki, K.C., 1986, "A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries", Ph.D. thesis, University of Minesota.

Launder, B.E., and Ying, W.M., 1972, *J. Fluid Mechanics*, **54**, part 2, 289-295.

Leray, J., 1934, "Essai sur le mouvement d'un liquide visqueux emplissant l'espace", *Acta Math.*, **63**.

Majumdar, S., Rodi, W., and Zhu, J., 1992, *J. of Fluid Engineering*, **114**, 496-503.

Melaan, M.C., 1992, *Num. Heat Transfer J., Part B*, **21**, 1-19.

Napolitano, M., and Orlandi, P., 1985, *Int. J. Num. Meth. Fluids*, **5**, 667.

Nikuradse, J., 1926, *Forschungsh. Ver. dtsh. Ing.*, 281.

Patankar, S.V., 1980, "Numerical Heat Transfer and Fluid Flow", Hemisphere, Washington D.C.

Patankar, S.V., and Spalding, D.B., 1972, *Int. J. Heat Mass Transfer*, **15**, 1787-1806.

Peric, M., 1985, "A Finite Volume Method for the Prediction of Three-Dimensional Fluid Flow in Complex Ducts", Ph.D. thesis, University of London, Imperial College.

Prandtl, L., 1953, "Essentials of Fluid Mechanics", New York Hafner.

Rhie, C.M., 1981, "A Numerical Study of the Flow Past an Isolated Airfoil with Separation", Ph. D. Thesis, Dept. of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign.

Rhie, C.M., and Chow, W.L., 1983, *AIAA J*, **21**, 1527-1532.

Richtmyer, R.D., and Morton, K.W., 1967, "Difference Method for Initial-Value Problems", Interscience New York.

Roberts, G.O., 1971, *Lecture Notes in Physics*, Vol. 8, Springer-Verlag Berlin Heidelberg, 171-176.

Saric, W.S., 1994, *Annu. Rev. Fluid Mech.*, **26**, 379-409.

Schlichting, H., 1979, "Boundary Layer Theory", Mac Graw Hill, New York.

Shyy, W., Tong, S.S., and Correa, S.M., 1985, *Num. Heat Transf.*, **8**, 99-113.

Smith, G.D., 1985, "Nonlinear Solutions of Partial Differential Equations", Oxford University Press.

Spalding, D.B., 1972, *Int. J. Num. Meth. Engng*, **4**, 551-559.

Timoshenko, S.P., and Goodier, J.M., 1951, "Theory of Elasticity", Mc Graw-Hill, New York.

Thompson, J.F., 1978, "Numerical solution of flow problems using body-fitted coordinate systems", *Lecture Series Von Karman Institute*.

Thompson, J.F., 1984, *AIAA Journal*, **22**, No. 11, 1505-1523.

Thompson, J.F., Warsi Z.U., and Mastin C.W., 1982, *J. Comp. Phys.*, **47**, 1-108.