

Trabajo Fin de Grado  
Grado en Ingeniería Aeroespacial. Mención en  
Vehículos Aeroespaciales

Análisis y valoración del FVM en problemas  
de convección-difusión térmica

Autor: Alejandro Cumplido Díaz

Tutor: Miguel Pérez-Saborid Sánchez-Pastor

Dpto. Ingeniería Aeroespacial y Mecánica de Fluidos  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2022





Trabajo Fin de Grado  
Grado en Ingeniería Aeroespacial. Mención en Vehículos Aeroespaciales

# **Análisis y valoración del FVM en problemas de convección-difusión térmica**

Autor:

Alejandro Cumplido Díaz

Tutor:

Miguel Pérez-Saborid Sánchez-Pastor

Profesor Titular

Dpto. Ingeniería Aeroespacial y Mecánica de Fluidos  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2022



Trabajo Fin de Grado: Análisis y valoración del FVM en problemas de convección-difusión térmica

Autor: Alejandro Cumplido Díaz

Tutor: Miguel Pérez-Saborid Sánchez-Pastor

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

En primer lugar, me gustaría agradecer a D. Miguel Pérez-Saborid Sánchez-Pastor por haber tutorizado de manera soberbia este Trabajo Fin de Grado. Su gran implicación, su ejemplar docencia y su extenso conocimiento sobre la materia han promovido enormemente mi motivación por el proyecto. Ha sabido ofrecerme las referencias literarias idóneas, que me han hecho sentir guiado en todo momento, al igual que ha sabido adaptar el alcance del proyecto según los obstáculos y limitaciones que he ido encontrando en el camino.

En segundo lugar, también he de agradecer a toda la población de la Línea de Vuelo del A400M por la oportunidad brindada, pues éste ha sido el lugar donde me he iniciado en el mundo laboral. Aunque no tienen una relación directa con el proyecto, debido a que el desempeño de dicho trabajo ha tenido lugar en simultaneidad con la elaboración de este documento, veo necesario dedicar este espacio a todas esas personas que tanto me han enseñado, tanto en lo profesional como en lo personal. Especial mención merece mi equipo de *Coordination Support*, que me acogieron con los brazos abiertos y me formaron excelentemente.

Por último, pero no menos importante, tengo que agradecer a todas esas personas que de una manera o de otra, directa o indirectamente, me han ofrecido su ayuda durante el trascurso de este trabajo. A toda mi familia, amigos y pareja, daros las gracias de corazón por haber estado siempre ahí, ofreciéndome un apoyo incondicional y resultando ser una vía de escape en los momentos más difíciles. Estoy seguro de que sin vosotros jamás lo hubiera conseguido.

*Alejandro Cumplido Díaz*  
*Escuela Técnica Superior de Ingeniería*

*Sevilla, 2022*



# Resumen

---

El FVM (*Finite Volume Method*) es un método extendidamente utilizado en CFD (*Computational Fluid Dynamics*) para la resolución de problemas en Mecánica de Fluidos. Este proyecto pretende constituir un primer paso para el desarrollo de un código CFD en MATLAB que, aunque simplificado, contenga los ingredientes básicos para permitir entender a los estudiantes el funcionamiento interno de códigos CFD comerciales mucho más complejos (Fluent, Star CCM, etc.), que tienen que ser utilizados como “cajas negras” por el usuario. Con este objetivo, en este Trabajo Fin de Grado se analizará y valorará el método FVM para problemas de convección-difusión térmica en geometrías bidimensionales, usando mallas tanto estructuradas como no estructuradas. De particular interés resultará la implementación del mallado y la subsecuente obtención a partir del mismo de la información geométrica necesaria para resolver el problema, así como de los diferentes métodos *upwind* de diferentes órdenes necesarios para tratar el término convectivo de forma estable.



# Abstract

---

The FVM (Finite Volume Method) is a widely used method in CFD (Computational Fluid Dynamics) for solving problems in Fluid Mechanics. This project aims to constitute a first step towards the development of a preliminary CFD code in MATLAB that contains basic notions to allow students understanding the internal workings of much more complex commercial CFD codes (Fluent, Star CCM, etc.), which have to be used as “black boxes” by the user. Focusing on this purpose, this Final Degree Project will analyze and evaluate the FVM for thermal convection-diffusion problems in two-dimensional geometries, using both structured and unstructured meshes. Particularly interesting will be the implementation of the mesh and the subsequent geometrical information obtained thereof needed to solve the problem, as well as the different upwind methods of different orders necessary to treat the convective term in a stable manner.



# Notación

---

$\rho$	Densidad [ $kg/m^3$ ]
$T$	Temperatura [ $K$ o $^{\circ}C$ ]
$\vec{v}$	Vector velocidad [ $m/s$ ]
$p$	Presión [ $Pa$ ]
$\mu$	Viscosidad cinemática [ $m^2/s$ ]
$\vec{f}_m$	Vector de fuerzas másicas [ $N/m^3$ ]
$e$	Energía interna específica [ $J/kg$ ]
$\underline{\underline{\tau'}} : \nabla \vec{v}$	Término de disipación de energía interna
$Q_r$	Potencia calorífica (radiación) por ud. de vol. [ $W/m^3$ ]
$Q_q$	Potencia calorífica (química) por ud. de vol. [ $W/m^3$ ]
$R_g$	Constante universal de los gases [ $J/Kg K$ ]
$\gamma$	Coefficiente de dilatación adiabática [–]
$c_p$	Calor específico a presión constante [ $J/kg K$ ]
$c_v$	Calor específico a volumen constante [ $J/kg K$ ]
$k$	Conductividad térmica [ $W/m K$ ]
$\phi$	Campo escalar genérico
$\nabla \cdot \phi$	Divergencia de $\phi$
$\nabla \phi$	Gradiente de $\phi$
$\vec{n}$	Vector normal
$\vec{t}$	Vector tangente
$\vec{k}$	Vector saliente del plano $x - y$ ( $\vec{k} = \vec{n} \times \vec{t}$ )
$A$	Área [ $m^2$ ]
$VC$	Volumen de control
$S_\phi$	Término “fuente” relativo a generación de $\phi$
$P$	Nodo central
$N$	Nodo norte
$S$	Nodo sur
$E$	Nodo este
$W$	Nodo oeste
$T$	Nodo superior
$B$	Nodo inferior
$n$	Cara norte
$s$	Cara sur
$e$	Cara este
$w$	Cara oeste

$t$	Cara superior
$b$	Cara inferior
$WW$	Nodo oeste con respecto a W
$EE$	Nodo este con respecto a E
$\delta x$	Longitud de celda [ $m$ ]
$\delta x_{Ww}$	Distancia entre el nodo $W$ y la cara $w$ [ $m$ ]
$\delta x_{wP}$	Distancia entre la cara $w$ y el nodo $P$ [ $m$ ]
$\delta x_{Pe}$	Distancia entre el nodo $P$ y la cara $e$ [ $m$ ]
$\delta x_{eE}$	Distancia entre la cara $e$ y el nodo $E$ [ $m$ ]
$\delta x_{WP}$	Distancia entre el nodo $W$ y el nodo $P$ [ $m$ ]
$\delta x_{PE}$	Distancia entre el nodo $P$ y el nodo $E$ [ $m$ ]
$\Gamma$	Coefficiente de difusión genérico
$\phi_W$	Propiedad $\phi$ evaluada en el nodo W
$\phi_P$	Propiedad $\phi$ evaluada en el nodo P
$\phi_E$	Propiedad $\phi$ evaluada en el nodo E
$\phi_{WW}$	Propiedad $\phi$ evaluada en el nodo WW
$\phi_{EE}$	Propiedad $\phi$ evaluada en el nodo EE
$a_W$	Coefficiente de $\phi_W$
$a_P$	Coefficiente de $\phi_P$
$a_E$	Coefficiente de $\phi_E$
$a_{EE}$	Coefficiente de $\phi_{EE}$
$a_{WW}$	Coefficiente de $\phi_{WW}$
$a_{SP}$	Coefficiente independiente referido a fuentes
$\bar{S}$	Generación media
$\Delta V$	Volumen de la celda [ $m^3$ ]
$D_w$	Término difusivo en la cara $w$
$D_e$	Término difusivo en la cara $e$
$F_w$	Término convectivo en la cara $w$
$F_e$	Término convectivo en la cara $e$
$\backslash$	<i>Backslash</i> (comando MATLAB)
$u$	Componente $x$ de la velocidad [ $m/s$ ]
$v$	Componente $y$ de la velocidad [ $m/s$ ]
$Pe$	Número de Peclet [–]
$\Delta t$	Paso de integración [ $s$ ]
$\theta$	Parámetro de ponderación [–]
$\phi_W^0$	Propiedad $\phi$ evaluada en el nodo W en el instante anterior
$\phi_P^0$	Propiedad $\phi$ evaluada en el nodo P en el instante anterior
$\phi_E^0$	Propiedad $\phi$ evaluada en el nodo E en el instante anterior
$L$	Longitud característica en dirección $x$ [ $m$ ]
$H$	Longitud característica en dirección $y$ [ $m$ ]
$M$	Número de nodos (en dirección $x$ )
$N$	Número de nodos (en dirección $y$ )
$\Delta x$	Longitud de celda en dirección $x$ (2D) [ $m$ ]
$\Delta y$	Longitud de celda en dirección $y$ (2D) [ $m$ ]
$\phi_S$	Propiedad $\phi$ evaluada en el nodo S
$\phi_N$	Propiedad $\phi$ evaluada en el nodo N
$a_S$	Coefficiente de $\phi_S$
$a_N$	Coefficiente de $\phi_N$
$D_s$	Término difusivo en la cara $s$
$D_n$	Término difusivo en la cara $n$

$F_s$	Término convectivo en la cara $s$
$F_n$	Término convectivo en la cara $n$
$u_m$	Velocidad media en dirección $x$ [ $m/s$ ]
$N_{f,P}$	Número de caras ( $f$ ) de la celda $P$
$n_c$	Número total de celdas
$n_f$	Número total de caras
$n_v$	Número total de vértices
$n_{fb}$	Número total de caras en la frontera
$(x_c, y_c)$	Coordenadas de los nodos [ $m$ ]
$(x_f, y_f)$	Coordenadas de los centros de las caras [ $m$ ]
$(x_v, y_v)$	Coordenadas de los vértices [ $m$ ]
$n_f(i_c)$	Número total de caras de cada celda
$n_v(i_c)$	Número total de vértices de cada celda
$n_v(i_f)$	Número total de vértices de cada cara
$Lcf(i_c, k)$	Matriz de conectividad Celda-A-Cara [ $n_c, 3$ ]
$Lfc(i_f, k)$	Matriz de conectividad Cara-A-Celda [ $n_f, 2$ ]
$Lcv(i_c, k)$	Matriz de conectividad Celda-A-Vértice [ $n_c, 3$ ]
$Lfv(i_f, k)$	Matriz de conectividad Cara-A-Vértice [ $n_f, 2$ ]
$Lf - bf(i_f)$	Vector de conectividad Cara-A-CaraFrontera [ $1, n_f$ ]
$Lbf - f(i_{fb})$	Vector de conectividad CaraFrontera-A-Cara [ $1, n_{fb}$ ]
$n_{x,f}$	Componente $x$ del vector normal a la cara $f$
$n_{y,f}$	Componente $y$ del vector normal a la cara $f$
$t_{x,f}$	Componente $x$ del vector tangente a la cara $f$
$t_{y,f}$	Componente $y$ del vector tangente a la cara $f$
$w_f$	Función interpolante para las caras
$w_{v,i}$	Función interpolante para los vértices
$\delta_f$	Distancia normal entre dos nodos adyacentes [ $m$ ]
$\vec{l}_f$	Vector que une dos nodos adyacentes
$J_{t,f}$	Flujo tangencial
$G_f$	Gasto másico [ $kg/s$ ]
CFD	<i>Computational Fluid Dynamics</i>
PDE	<i>Partial Differential Equation</i>
CVM	<i>Control Volume Method</i>
FDM	<i>Finite Difference Method</i>
FEM	<i>Finite Element Method</i>
FVM	<i>Finite Volume Method</i>
FOU	<i>First Order Upwind</i>
SOU	<i>Second Order Upwind</i>
SIMPLE	<i>Semi-Implicit Method for Pressure Linked Equations</i>
QUICK	<i>Quadratic Upstream Interpolation for Convective Kinematics</i>



# Índice de Figuras

---

1.1	Ejemplo de una simulación CFD para el análisis de una aeronave (imagen extraída de [7])	5
1.2	Ejemplo de mallado estructurado para un perfil aerodinámico (imagen tomada de [3])	6
1.3	Ejemplo de mallado no estructurado para un perfil aerodinámico (imagen tomada de [4])	7
1.4	Definición del volumen de control	11
2.1	Mallado no homogéneo según el procedimiento A	14
2.2	Mallado no homogéneo según el procedimiento B	14
2.3	Representación del problema descrito	17
2.4	Solución proporcionada por el FVM para el problema descrito	19
2.5	Comparación entre la solución exacta y la obtenida mediante el FVM empleando 5 nodos equiespaciados	20
2.6	Comparación entre la solución exacta y la obtenida mediante el FVM empleando 25 nodos equiespaciados	20
2.7	Comparación entre la solución exacta y la obtenida mediante el FVM empleando 5 nodos ubicados aleatoriamente	21
2.8	Comparación entre la solución exacta y la obtenida mediante el FVM empleando 50 nodos con mayor concentración en los extremos	22
2.9	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 0.1 \text{ m/s}$ empleando 5 nodos equiespaciados y usando el método de las diferencias centradas	26
2.10	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 3 \text{ m/s}$ empleando 5 nodos equiespaciados y usando el método de las diferencias centradas	27
2.11	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 3 \text{ m/s}$ empleando 20 nodos equiespaciados y usando el método de las diferencias centradas	27
2.12	Problema paradigmático para ejemplificar la propiedad de conservatividad	28
2.13	Distribución de $\phi$ en la vecindad de dos fuentes. En la figura superior, el caso límite $Pe \rightarrow 0$ . En la figura inferior, el caso límite $Pe \rightarrow \infty$	30
2.14	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 0.1 \text{ m/s}$ empleando 5 nodos equiespaciados y usando el método <i>upwind</i> de primer orden	32
2.15	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 3 \text{ m/s}$ empleando 5 nodos equiespaciados y usando el método <i>upwind</i> de primer orden	32
2.16	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 3 \text{ m/s}$ empleando 25 nodos (con mayor densidad en los extremos) y usando el método <i>upwind</i> de primer orden	33
2.17	Nomenclatura de nodos en el método <i>upwind</i> de segundo orden	34
2.18	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 0.1 \text{ m/s}$ empleando 5 nodos equiespaciados y usando el método <i>upwind</i> de segundo orden	36

2.19	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 3\text{ m/s}$ empleando 20 nodos equiespaciados y usando el método <i>upwind</i> de segundo orden	36
2.20	Comparación entre la solución exacta y la aproximación polinómica propuesta por Patankar	38
2.21	Comparación entre la solución exacta y la obtenida mediante el FVM para $u = 3\text{ m/s}$ empleando 20 nodos equiespaciados y usando el esquema de ley de potencia de Patankar	38
2.22	Comparación entre la solución exacta y los distintos métodos de discretización para $u = 0.1\text{ m/s}$ empleando 5 nodos equiespaciados	39
2.23	Comparación entre la solución exacta y los distintos métodos de discretización para $u = 3\text{ m/s}$ empleando 5 nodos equiespaciados	39
2.24	Comparación entre la solución exacta y los distintos métodos de discretización para $u = 3\text{ m/s}$ empleando 20 nodos equiespaciados	40
2.25	Geometría del problema que ejemplifica el caso no estacionario para $t > 0$	42
2.26	Comparación entre la solución analítica y la solución mediante el FVM empleando 5 nodos, usando el esquema explícito para el término no estacionario y con un paso $\Delta t = 2\text{ s}$	43
2.27	Comparación entre la solución analítica y la solución mediante el FVM empleando 5 nodos equiespaciados, usando el esquema explícito para el término no estacionario y con un paso $\Delta t = 10\text{ s}$ en $t = 40\text{ s}$	44
2.28	Comparación entre la solución analítica y la solución mediante el FVM empleando 20 nodos equiespaciados, usando el esquema explícito para el término no estacionario y con un paso $\Delta t = 2\text{ s}$ en $t = 40\text{ s}$	44
2.29	Comparación entre la solución analítica y la solución mediante el FVM empleando 20 nodos equiespaciados, usando el esquema explícito para el término no estacionario y con un paso $\Delta t = 0.25\text{ s}$ en $t = 40\text{ s}$	45
2.30	Comparación entre la solución analítica y la solución mediante el FVM empleando 5 nodos equiespaciados, usando el esquema implícito para el término no estacionario y con un paso $\Delta t = 2\text{ s}$	46
2.31	Comparación entre la solución analítica y la solución mediante el FVM empleando 20 nodos equiespaciados, usando el esquema implícito para el término no estacionario y con un paso $\Delta t = 2\text{ s}$	46
3.1	Mallado 2D cartesiano	50
3.2	Representación del problema descrito	52
3.3	Mallado del dominio de computación representado en MATLAB. Los puntos rojos son los nodos donde se calcularán los valores de temperatura	55
3.4	Valores del campo de temperaturas en los nodos para un mallado uniforme $M \times N$ con $M = 4$ y $N = 4$	55
3.5	Valores del campo de temperaturas en los nodos para un mallado uniforme $M \times N$ con $M = 50$ y $N = 50$	56
3.6	Curvas de nivel (isotermas) para un mallado uniforme $M \times N$ con $M = 50$ y $N = 50$	57
3.7	Representación del problema descrito	58
3.8	Valores del campo de temperaturas en los nodos para un mallado $M \times N$ con $M = 50$ y $N = 10$	60
3.9	Curvas de nivel (isotermas) para un mallado $M \times N$ con $M = 50$ y $N = 10$	61
4.1	Malla ortogonal cartesiana para aproximar un problema de flujo en un semicilindro	63
4.2	Indexación global en malla no estructurada	66
4.3	Representación de los vectores unitarios tangente y normal en una cara localmente con respecto a una celda	71
4.4	Esquema paradigmático para la obtención de las funciones interpoladoras $w_f$ y $w_{v,i}$	74

---

4.5	Valores del campo de temperaturas en los nodos para un mallado no estructurado $M \times N$ con $M = 3$ y $N = 3$	84
4.6	Valores del campo de temperaturas en los nodos para un mallado no estructurado $M \times N$ con $M = 50$ y $N = 50$	85
4.7	Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado $M \times N$ con $M = 50$ y $N = 50$	86
4.8	Valores del campo de temperaturas en los nodos para un mallado no estructurado $M \times N$ con $M = 30$ y $N = 30$ en el problema analizado en [6] y [17]-[20]	87
4.9	Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado $M \times N$ con $M = 30$ y $N = 30$ en el problema analizado en [6] y [17]-[20]	89
4.10	Representación del problema descrito en [4]	89
4.11	Valores del campo de temperaturas en los nodos para un mallado no estructurado $M \times N$ con $M = 30$ y $N = 40$ en el problema analizado en [4]	90
4.12	Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado $M \times N$ con $M = 30$ y $N = 40$ en el problema analizado en [4]	91
4.13	Valores del campo de temperaturas en los nodos para un mallado no estructurado $M \times N$ con $M = 50$ y $N = 10$ utilizando el esquema <i>upwind</i>	94
4.14	Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado $M \times N$ con $M = 50$ y $N = 10$	95
4.15	Representación del problema descrito en [6]	96
4.16	Valores del campo de temperaturas en los nodos para un mallado no estructurado $M \times N$ con $M = 50$ y $N = 10$ utilizando el esquema <i>upwind</i> en el problema analizado en [6]	97
4.17	Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado $M \times N$ con $M = 50$ y $N = 10$ en el problema analizado en [6]	98



# Índice de Tablas

---

2.1	Valores del campo de temperaturas y error relativo cometido por el FVM empleando 5 nodos equiespaciados	19
2.2	Valores del campo de temperaturas y error relativo cometido por el FVM empleando 5 nodos ubicados aleatoriamente	21
2.3	Valores de $\phi$ y comparación del error relativo cometido por el FVM usando el método FOU y SOU, para $u = 3\text{ m/s}$ y empleando 7 nodos equiespaciados	36
2.4	Valores del campo de temperaturas y error relativo cometido por el FVM usando el esquema explícito, para $\Delta t = 2\text{ s}$ y empleando 5 nodos equiespaciados en $t = 40\text{ s}$	43



# Índice Abreviado

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Notación</i>	VII
<i>Índice de Figuras</i>	XI
<i>Índice de Tablas</i>	XV
<i>Índice Abreviado</i>	XVII
<b>1 Introducción</b>	<b>1</b>
1.1 Fundamentos del CFD	1
1.2 Historia del CFD	3
1.3 Estado del arte en CFD	4
1.4 Estructura de los códigos CFD	5
1.5 Ecuación de transporte	9
<b>2 FVM para problemas convectivo-difusivos en 1D</b>	<b>13</b>
2.1 Problema difusivo, estacionario y unidimensional	13
2.2 Problema convectivo-difusivo, estacionario y unidimensional. Esquemas de discretización	24
2.3 Problema difusivo, no estacionario y unidimensional	40
<b>3 FVM para problemas convectivo-difusivos en 2D. Mallado estructurado</b>	<b>49</b>
3.1 Problema difusivo, estacionario y bidimensional	49
3.2 Problema convectivo-difusivo, estacionario y bidimensional	57
<b>4 FVM para problemas convectivo-difusivos en 2D. Mallado no estructurado</b>	<b>63</b>
4.1 Problema difusivo, estacionario y unidimensional	64
4.2 Problema convectivo-difusivo, estacionario y unidimensional	91
<b>5 Conclusión y trabajos futuros</b>	<b>101</b>
<b>Apéndice A Códigos de MATLAB</b>	<b>103</b>
A.1 Convección-difusión estacionaria 1D	103
A.2 Difusión no estacionaria 1D	110
A.3 Difusión estacionaria 2D con mallado estructurado	115
A.4 Convección-difusión estacionaria 2D con mallado estructurado	120

*Bibliografía*

127

# Índice

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Notación</i>	VII
<i>Índice de Figuras</i>	XI
<i>Índice de Tablas</i>	XV
<i>Índice Abreviado</i>	XVII
<b>1 Introducción</b>	<b>1</b>
1.1 Fundamentos del CFD	1
1.1.1 Método de las Diferencias Finitas	2
1.1.2 Método de los Volúmenes Finitos	2
1.1.3 Método de los Elementos Finitos	3
1.2 Historia del CFD	3
1.3 Estado del arte en CFD	4
1.4 Estructura de los códigos CFD	5
1.4.1 Pre-procesador	5
1.4.2 Solucionador	7
1.4.3 Post-procesador	8
1.5 Ecuación de transporte	9
<b>2 FVM para problemas convectivo-difusivos en 1D</b>	<b>13</b>
2.1 Problema difusivo, estacionario y unidimensional	13
2.1.1 Mallado del dominio unidimensional	13
2.1.2 Discretización de las ecuaciones	14
2.1.3 Resolución del sistema	16
2.1.4 Aplicación práctica 1D	17
2.2 Problema convectivo-difusivo, estacionario y unidimensional. Esquemas de discretización	24
2.2.1 Esquema de las diferencias centradas	25
2.2.2 Propiedades de los esquemas de discretización	28
2.2.3 Esquema <i>upwind</i> de primer orden	31
2.2.4 Esquema <i>upwind</i> de segundo orden	34
2.2.5 Esquema de ley de potencia de Patankar	37
2.3 Problema difusivo, no estacionario y unidimensional	40
2.3.1 Esquema explícito	42
2.3.2 Esquema completamente implícito	45

---

<b>3</b>	<b>FVM para problemas convectivo-difusivos en 2D. Mallado estructurado</b>	<b>49</b>
3.1	Problema difusivo, estacionario y bidimensional	49
3.1.1	Mallado del domino bidimensional	50
3.1.2	Discretización de las ecuaciones	51
3.1.3	Resolución del sistema	51
3.1.4	Aplicación práctica 2D. Mallado estructurado	52
3.2	Problema convectivo-difusivo, estacionario y bidimensional	57
<b>4</b>	<b>FVM para problemas convectivo-difusivos en 2D. Mallado no estructurado</b>	<b>63</b>
4.1	Problema difusivo, estacionario y unidimensional	64
4.1.1	Mallado del dominio. Procesamiento y almacenamiento de la información geométrica	65
4.1.2	Discretización de las ecuaciones. Flujos normales y tangenciales	76
4.1.3	Resolución del sistema. Ensamblaje de las matrices	80
4.1.4	Aplicación práctica 2D. Mallado no estructurado	84
4.2	Problema convectivo-difusivo, estacionario y unidimensional	91
<b>5</b>	<b>Conclusión y trabajos futuros</b>	<b>101</b>
	<b>Apéndice A Códigos de MATLAB</b>	<b>103</b>
A.1	Convección-difusión estacionaria 1D	103
A.2	Difusión no estacionaria 1D	110
A.3	Difusión estacionaria 2D con mallado estructurado	115
A.4	Convección-difusión estacionaria 2D con mallado estructurado	120
	<i>Bibliografía</i>	127

# 1 Introducción

---

La Dinámica de Fluidos Computacional, extendidamente conocida bajo el nombre de CFD (siglas de su voz inglesa, *Computational Fluid Dynamics*) es una rama de la Mecánica de Fluidos que aborda el análisis numérico de problemas relacionados con el flujo de fluidos y la transferencia de calor, así como otros fenómenos intrínsecamente relacionados con éstos y de mayor complejidad como pueden ser la turbulencia, la radiación o las reacciones químicas en motores de combustión. Mediante potentes algoritmos y herramientas de cálculo numérico embebidas en robustos softwares es posible resolver problemas fluidodinámicos de diversa índole, abarcando campos como la aerodinámica, la hidrodinámica, el estudio de turbomáquinas y motores o la ingeniería eléctrica, electrónica e incluso biomédica. Así, con el paso de los años los códigos CFD se han convertido en una herramienta de vital importancia en el desarrollo de numerosos productos industriales. Tanto es así, que el diseño actual de aeronaves, automóviles, cohetes, submarinos y, en general, casi cualquier vehículo móvil sería impensable sin el uso de CFD.

En este capítulo se detallarán los pilares sobre los que se sustenta todo código CFD, se describirá su evolución a lo largo de la historia y se expondrán sus ventajas y desventajas frente a otras técnicas de análisis.

## 1.1 Fundamentos del CFD

Lo que hoy en día se entiende por códigos comerciales CFD no es más que un conjunto de algoritmos diseñados para resolver complejas expresiones matemáticas no lineales que definen las ecuaciones fundamentales del flujo de fluidos y el transporte de calor: las ecuaciones de Navier-Stokes. Estas ecuaciones se resuelven iterativamente usando dichos algoritmos embebidos en softwares de CFD. De esta forma, se obtiene la solución numérica de una expresión abstracta cuyo significado en papel solo puede ser interpretado por unos pocos expertos en la materia. Así, con una representación de la solución apropiada a través de gráficas, campos vectoriales, códigos de colores o animaciones, se obtiene un resultado mucho más revelador, intuitivo y fácil de interpretar.

Para comprender las bases del CFD hay que entender que este tipo de análisis tiene su raíz en el uso de ciertos principios físico-matemáticos que establecen una relación de causa-efecto en el sistema objeto de estudio. Esta relación de causalidad, que se materializa en un modelo matemático (que por definición relaciona entradas y salidas), puede manifestarse en forma de una simple ecuación o en forma de algo mucho más complejo. El pináculo de complejidad sería un conjunto de ecuaciones diferenciales en derivadas parciales (conocidas como PDEs, del inglés *Partial Differential Equations*) como son las ecuaciones de Navier-Stokes, que describen el comportamiento de los fluidos y que se

sustentan en leyes dogmáticas como las de conservación de masa, cantidad de movimiento o energía.

Estas PDEs, que normalmente surgen de modelos basados en la física, son capaces de predecir la evolución espacio-temporal de una cantidad física de interés dentro de un sistema. Sin embargo, la capacidad de encontrar soluciones analíticas está limitada por las restricciones que subyacen a los métodos conocidos para su obtención (regularidad de la geometría, linealidad de las ecuaciones, coeficientes constantes...). Esto restringe la aplicabilidad de los métodos analíticos a un pequeño conjunto de problemas casi irrelevante, sin permitir abordar la gran mayoría de situaciones de interés que se dan en la práctica. Es por esto que surge la necesidad de elaborar métodos numéricos que posibiliten afrontar satisfactoriamente este amplio rango de problemas.

Los métodos que se emplean para obtener soluciones numéricas en CFD se clasifican principalmente en dos: **deterministas** y **estocásticos**. Los métodos deterministas son aquellos en los que dada una entrada la salida siempre es la misma, independientemente de cuántas veces se realice la operación o en qué condiciones. Por contra, los métodos estocásticos se basan en principios estadísticos y dependen del número de ensayos realizado, es decir, del tamaño de la muestra. Debido a la presencia del error inherente a la naturaleza de los métodos estocásticos, en este documento se estará interesado únicamente en el análisis numérico con métodos deterministas.

Dentro de las técnicas de resolución numérica deterministas de PDEs, los métodos más populares son el **método de las diferencias finitas** (FDM), el **método de los volúmenes finitos** (FVM) y el **método de los elementos finitos** (FEM). A continuación se resumen sucintamente los fundamentos de cada uno, aunque en este Trabajo Fin de Grado se estará únicamente interesado en el **FVM**.

### 1.1.1 Método de las Diferencias Finitas

En este método se impone que el conjunto de PDEs que gobierna el sistema se satisfaga en un conjunto de puntos establecidos en el dominio fluido denominados **nodos**. La unión de dichos puntos entre sí genera una **mall**a, y se dice por ende que en este método los nodos se localizan en los vértices de la misma. Las condiciones de contorno se satisfarán por tanto en nodos localizados en el contorno del dominio. El nombre del método se debe a que las derivadas parciales de la ecuación diferencial se estiman mediante aproximaciones diferenciales, derivadas normalmente de los desarrollos en serie de Taylor. En [6] se analiza este método con exhaustiva profundidad.

### 1.1.2 Método de los Volúmenes Finitos

Este método toma su nombre de la particularidad de que las ecuaciones que gobiernan el sistema se satisfacen en **volúmenes de control** de tamaño finito en lugar de en puntos. Así, el dominio de computación queda dividido en celdas o células, normalmente de forma poligonal y cuyas aristas reciben el nombre de caras.

El primer paso a realizar en este método consiste en integrar la PDE en cada celda del dominio, obteniendo un conjunto de ecuaciones discretas. Esto establece una clara diferencia entre este método y el anterior, pues mientras en el método de las diferencias finitas la PDE se calculaba directamente, en este método primero se integra, después se aproxima y por último se resuelve. Además, como usualmente se elige el centroide del volumen de control como punto donde almacenar la información de la celda, es evidente que ninguno de estos centroides ocupará el contorno del dominio, por lo que las condiciones de contorno no podrán satisfacerse exactamente como sí ocurría en el método anterior. Se dice por tanto que la solución obtenida está en su **forma débil**, mientras que antes se obtuvo en su **forma fuerte**.

Pero sin duda la mayor diferencia que presenta este método frente al resto es la **conservatividad** inherente su naturaleza. Sin entrar en detalle, esta característica viene a decir que la ecuación que gobierna el método de los volúmenes finitos es, en esencia, una ecuación de balance. En capítulos posteriores se detallarán las repercusiones que este hecho tiene y por qué lo convierte en el método predilecto cuando se habla de CFD.

### 1.1.3 Método de los Elementos Finitos

Al igual que el método de los volúmenes finitos, el método de los elementos finitos determina la forma débil de la solución al PDE. Sin entrar en mucho detalle, pueden distinguirse dos pasos en la aplicación de este método. En primer lugar, se transforma la forma fuerte de la PDE en la forma débil. Para ello, se multiplican las ecuaciones por una función de prueba y se integra en el dominio. La elección de los valores de la función de prueba vendrá determinada por las condiciones de contorno. En segundo lugar y previo a la aplicación del método, se discretiza el dominio de computación en elementos convexos de tamaño finito (normalmente, polígonos en el caso 2D o poliedros en el caso 3D). Una vez llegados a este punto, se define la solución y la función de prueba como una combinación lineal de funciones de forma (tantas como nodos haya). Sustituyendo en la PDE pueden obtenerse los coeficientes de la combinación lineal y, por tanto, la solución (para una función de forma escogida).

Históricamente, el método de los elementos finitos se ha utilizado en la resolución de problemas de la mecánica de sólidos, mientras que el método de las diferencias finitas y el método de los volúmenes finitos se han usado tradicionalmente para resolver problemas de flujo y transmisión de calor. No obstante, la flexibilidad que presenta el método de los elementos finitos a la hora de elegir las funciones de forma y su grado permite refinar la precisión de la solución hasta en las geometrías más complejas. Esta atractiva característica ha suscitado la tendencia de buscar nuevos métodos de elementos finitos que pudiesen ser extrapolados a los problemas de dinámica de fluidos y transmisión de calor, en los que la propiedad de conservatividad es la prioridad. La inexistencia de mecanismos que garantizaran la conservación (local y global) en los métodos de elementos finitos motivó el desarrollo de métodos como el CVFEM, un método híbrido entre el FVM y el FEM capaz de asegurar conservación y altamente empleado en los códigos comerciales actuales. Sin embargo, su estudio queda fuera del alcance de este proyecto, centrándose los esfuerzos únicamente en el FVM.

## 1.2 Historia del CFD

Desde las civilizaciones más primigenias, el movimiento de los fluidos siempre ha suscitado gran interés en la humanidad. El flujo del metal fundido, la corriente de agua en un río o el azote del viento han sido siempre cuna de fascinación. En la Antigua Grecia, célebres científicos como Arquímedes comenzaron a sentar las bases de la hidrostática y desarrollaron los primeros métodos para calcular densidades y volúmenes. Durante el Renacimiento, Leonardo Da Vinci hizo grandes aportaciones a la mecánica de fluidos estudiando la superficie del agua y las olas. Ya en el S. XVII con la llegada de Sir Isaac Newton se produjo un gran salto cualitativo en el entendimiento del movimiento fluídico, gracias a aportaciones como las Leyes de Newton y el concepto de viscosidad newtoniana.

Los siglos posteriores supusieron un avance mayúsculo en este campo gracias al esfuerzo titánico de intentar describir matemáticamente el movimiento de los fluidos. En el S. XVIII Bernoulli formuló su célebre ecuación, y Euler aportó las ecuaciones de flujo ideal (sin viscosidad) y sentó las bases de la teoría potencial. Durante el S. XIX, el francés H. Navier y el irlandés G. Stokes postularon las famosas **ecuaciones de Navier-Stokes** al introducir los términos viscosos en las ecuaciones de Euler.

Estas expresiones en forma diferencial son las bases de la industria CFD moderna y conforman una caracterización completa del movimiento fluídico, incluyendo la turbulencia. Este sistema altamente no lineal y cuyas ecuaciones se encuentran fuertemente acopladas es extraordinariamente difícil de resolver, y no ha sido hasta los años 60 con el avance de las tecnologías digitales que ha sido posible obtener soluciones válidas a problemas de flujo en un tiempo computacional razonable. También en el S. XX se hicieron grandes progresos en la dinámica de fluidos, con aportaciones como la teoría de capa límite de L. Prandtl o la formulación de distintos modelos turbulentos.

Con todo esto, la primera aplicación CFD (en el sentido moderno) fue a manos del inglés L. Richardson, quien desarrolló en 1922 un sistema numérico de predicción climatológica basado en celdas y diferencias finitas (que, desgraciadamente, resultó siendo un fracaso). Durante los años 60 la NASA contribuyó con numerosos métodos de cálculo numérico que aún se usan actualmente en los códigos CFD. En los años 70 se propusieron métodos de diferenciación como el *upwind*, así como ciertos modelos de combustión. En 1980 se publicó el libro *Numerical Heat Transfer and Fluid Flow* de V. Patankar, posiblemente la obra más influyente que existe a día de hoy en lo que a CFD se refiere.

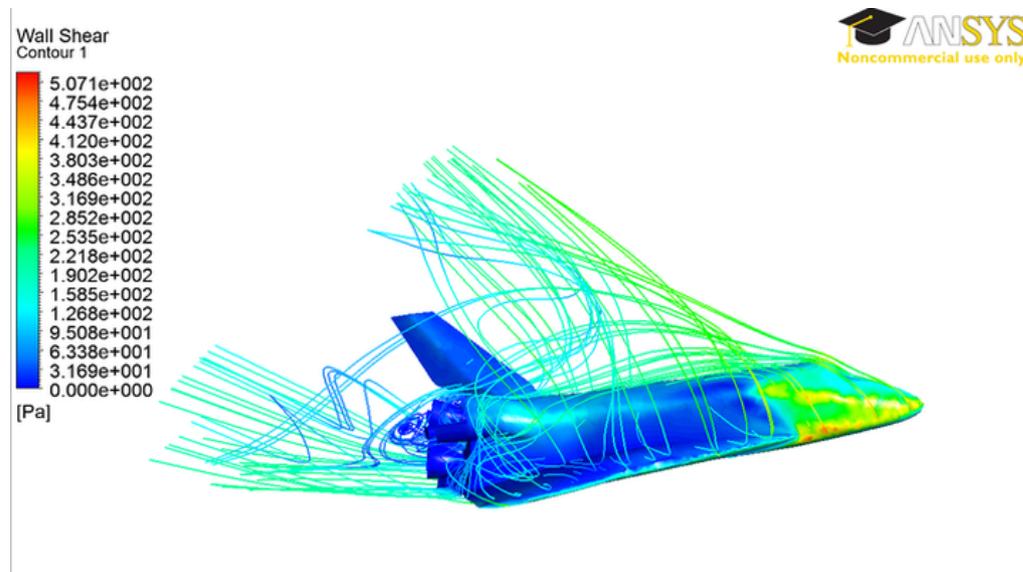
Ya entrados los años 80, los códigos comerciales CFD comenzaron a lanzarse al mercado de forma masiva. Las grandes empresas adquirieron las licencias de ciertos softwares CFD y comenzaron a incluir esta práctica en sus estrategias de diseño. Así ha continuado siendo hasta día de hoy, donde el CFD forma parte del conjunto de herramientas esenciales dentro del marco de la ingeniería asistida por ordenador. El desarrollo acelerado de los microprocesadores, a la par que la compactación del hardware y el abaratamiento de su fabricación, han mejorado la velocidad de computación de manera exponencial, permitiendo obtener en segundos resultados análogos a los que se obtendrían en un túnel de viento y empleando únicamente un dispositivo versátil y al alcance de la mayoría.

### **1.3 Estado del arte en CFD**

Para acometer la tarea de resolución numérica de problemas relacionados con la dinámica de fluidos, se necesitan potentes algoritmos implementados en softwares que permitan transformar las leyes físicas constituidas por las complejas ecuaciones de flujo en derivadas parciales en ecuaciones algebraicas más sencillas y resolverlas de manera significativamente eficiente y rápida. Como ya se ha comentado, el alto potencial de esta herramienta hace que su uso se extienda por un amplio rango de áreas de aplicación, en particular aquéllas relacionadas con la industria aeroespacial, siendo primordial en el estudio aerodinámico de aeronaves y turbomáquinas. El objetivo final de los desarrollos en CFD es proporcionar una competencia comparable a la de otras herramientas de ingeniería asistida por ordenador, como los códigos de análisis de tensiones. No obstante, la tremenda complejidad de la física que subyace al análisis CFD impide obtener resultados lo suficientemente completos en un tiempo aceptable, aunque dicha tendencia está cambiando en las últimas décadas con el desarrollo de supercomputadores de última tecnología a precio asequible y la implementación de interfaces más sencillas para el usuario. Estas nuevas características han provocado un reciente aumento del interés en este campo, introduciendo de lleno el CFD en la comunidad industrial actual.

El alto potencial de este tipo de softwares comerciales justifica el elevado coste que supone adquirir las licencias que permiten su uso, típicamente rondando los \$30.000. Sin embargo, este precio no es comparable al que supondría reproducir experimentalmente muchos de los estudios o análisis que los códigos CFD permiten realizar. La capacidad de estudiar sistemas más allá de sus límites nominales de operación, reproducir condiciones peligrosas sin riesgos, el significativo decremento del tiempo de análisis o la gran facilidad para realizar estudios paramétricos y de optimización de alta calidad sin costes adicionales son algunas de las ventajas únicas que convierten a los códigos

CFD en una herramienta muy atractiva en detrimento de la experimentación. No obstante, y como se verá más adelante, la complejidad de este campo requiere que los usuarios de estos softwares estén altamente cualificados y sean expertos en la materia para modelar correctamente el problema objeto de estudio, y evaluar y discutir correctamente los resultados.



**Figura 1.1** Ejemplo de una simulación CFD para el análisis de una aeronave (imagen extraída de [7]).

## 1.4 Estructura de los códigos CFD

Para comprender cómo trabaja un software de CFD es necesario analizar internamente la estructura que lo forma. Cuando se realiza una simulación, se han de llevar a cabo una serie de pasos secuenciados que guardan relación entre sí. Típicamente, pueden distinguirse tres elementos bien diferenciados: pre-procesador, *solver* y post-procesador. A continuación se describen detalladamente cada una de las fases asociadas a estos elementos cuando se trata del FVM. Importante destacar que la exposición de las mismas se ha realizado fiel a [4].

### 1.4.1 Pre-procesador

En esta fase de la simulación han de establecerse las bases del problema que se quiere resolver. En primer lugar, es necesario definir el dominio computacional que impone la geometría del problema. Seguidamente, se discretiza espacialmente dicho dominio en volúmenes de control, que no son más que celdas en las que las ecuaciones del sistema deben satisfacerse. Por último, habrá que caracterizar físicamente el problema estableciendo las propiedades del fluido en cuestión, imponiendo las condiciones iniciales y de contorno y teniendo en cuenta los fenómenos físicos que han de modelarse (turbulencia, radiación, reacciones químicas, etc.). Resulta interesante destacar que esta tarea consume la mayor parte del tiempo de la simulación. Todo paquete comercial debe incluir una interfaz sofisticada que permita aprovechar al máximo el potencial de los códigos CFD. Simultáneamente, debe ser una interfaz intuitiva para que el usuario pueda caracterizar el problema de la forma más sencilla posible, definiendo correctamente todo lo anterior. El pre-procesador debe poder transformar esta entrada de datos en una forma apropiada y legible para el solucionador.

Es importante destacar que la variable en cuestión que se pretenda calcular como solución al problema de flujo objeto de estudio (temperatura, velocidad...) se obtiene en nodos ubicados en cada volumen de control definido. Así, la precisión del método estará supeditada al número de celdas en que se divide el dominio, lo que es equivalente a decir que la solución será generalmente más precisa cuanto más fino sea el mallado propuesto. No obstante, un mallado más refinado requiere un mayor número de operaciones y por tanto un mayor coste computacional. Para llegar a un compromiso entre tiempo de computación y precisión de la solución será necesario definir un mallado adecuado.

De forma general, existen dos maneras de realizar el mallado: de forma estructurada y de forma no estructurada.

- **Mallado estructurado.** Consiste en emplear elementos típicamente cuadriláteros que conformen un diseño simple y adaptado a la geometría del problema en la división del dominio computacional. Normalmente, las celdas de los mallados estructurados pueden ser fácilmente identificadas por la terna  $\{i,j,k\}$ . Esto facilita enormemente los algoritmos al estar las celdas vecinas igualmente identificadas de forma sencilla. Otra ventaja es que la manera en que se guarda la información suele ser más directa y sencilla que en el caso no estructurado. Además, el hecho de disponer los elementos de forma alineada con la dirección del flujo facilita la convergencia del *solver*. El principal problema radica en que un mallado estructurado puede no adaptarse correctamente a la geometría cuando esta es compleja. En la Figura 1.2 puede encontrarse un ejemplo de un mallado estructurado, fácilmente reconocible por presentar un patrón que se repite en toda la malla.

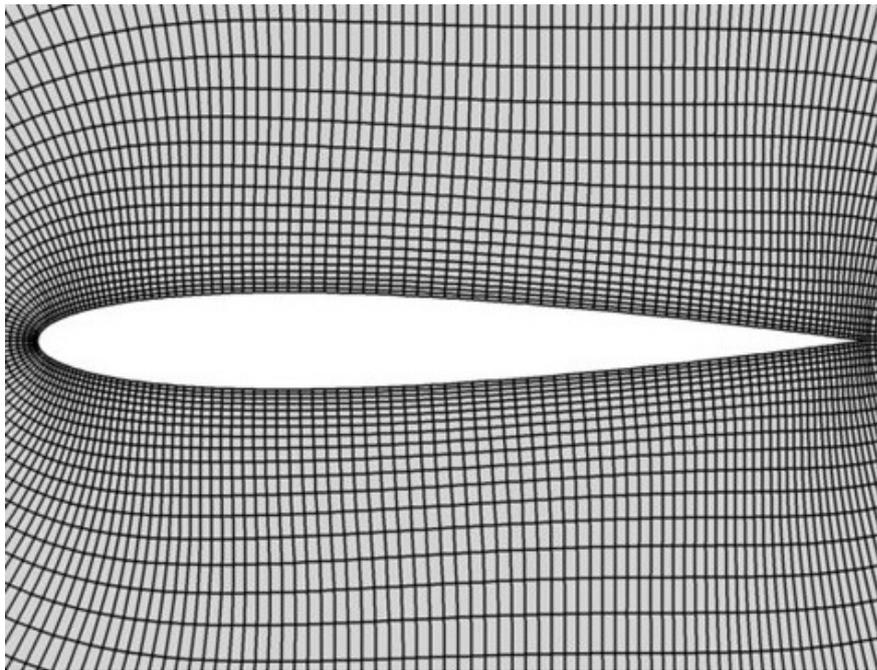
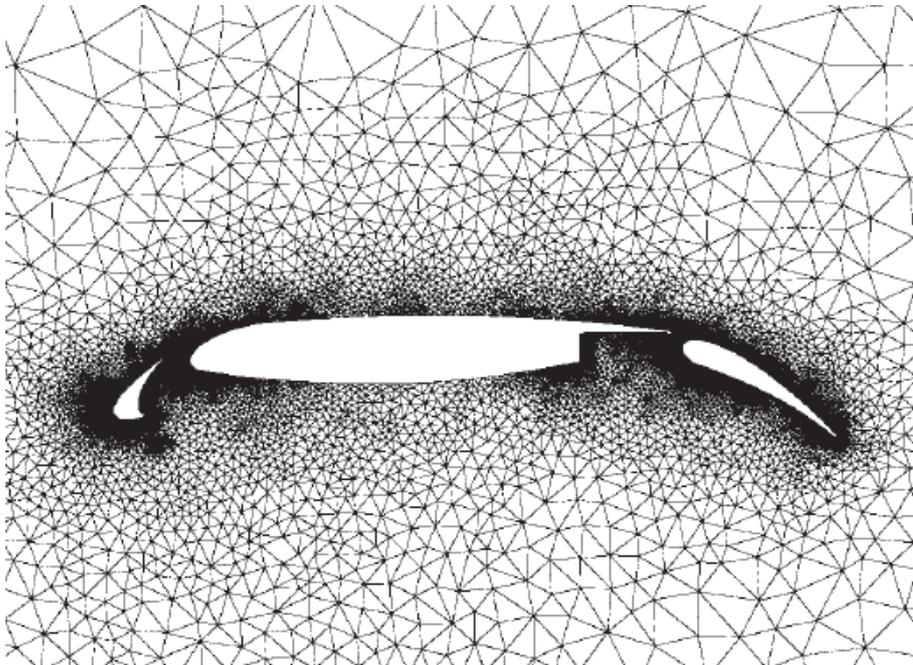


Figura 1.2 Ejemplo de mallado estructurado para un perfil aerodinámico (imagen tomada de [3]).

- **Mallado no estructurado.** Cuando resulta difícil realizar un mallado estructurado debido a la complejidad de la geometría, es necesario recurrir a mallados no estructurados. Típicamente están conformados por celdas triangulares que no siguen un patrón fijo, y cuyas vecindades locales se disponen de manera arbitraria, inhabilitando la posibilidad de definir las mediante la terna  $\{i,j,k\}$ . Esto provoca que el almacenamiento de la información sea más costoso, pues aparece la necesidad de guardar en la memoria la conectividad de las celdas. Además, al no

estar alineadas con la dirección del flujo, la convergencia suele ser más lenta. En la Figura 1.3 puede observarse un mallado no estructurado. Son los más usuales en las aplicaciones CFD gracias a su adaptabilidad frente a geometrías complejas.



**Figura 1.3** Ejemplo de mallado no estructurado para un perfil aerodinámico (imagen tomada de [4]).

Existe otro tipo de mallado que busca el compromiso entre los estructurados y los no estructurados. Son los conocidos como **mallados estructurados por bloques**, en los que el dominio se divide arbitrariamente en sub-regiones que, internamente, se comportan como mallados estructurados. Aunque se trata de una disposición de gran utilidad, queda fuera del alcance de esta memoria.

En definitiva, algo común a cualquier estructura es que el mallado óptimo debe ser no uniforme, con una mayor densidad de nodos en las regiones donde la variable fluida de interés tiene variaciones grandes, y más espaciados cuando dichas variaciones son pequeñas.

La tendencia actual está enfocada al desarrollo de códigos con una capacidad de adaptabilidad automática en función de las condiciones del problema, delegando en el propio código algunas de las decisiones que el usuario debe tomar en la definición del sistema, y que normalmente abarcan la mayor parte del tiempo en la industria del CFD.

#### 1.4.2 Solucionador

Usualmente conocido por su voz inglesa, *solver*, es el núcleo de la estructura de los códigos CFD y donde se encuentran embebidos los métodos numéricos que permiten lidiar con los problemas de flujo. Dentro de las técnicas de resolución numérica pueden distinguirse tres vertientes, ya descritas en el Capítulo 1.1: método de las diferencias finitas, método de los volúmenes finitos y método de los elementos finitos. Como ya se ha comentado, se centrarán los esfuerzos en el método de los volúmenes finitos (FVM), también denominado método de los volúmenes de control (CVM), un tipo de formulación en diferencias finitas de extendido uso en los códigos CFD habituales y cuyo algoritmo consta de los siguientes pasos:

- **Integración** de las ecuaciones que gobiernan el problema en todos los volúmenes de control en los que se ha dividido el dominio fluido. Es este paso el que distingue al método de los volúmenes finitos de otras técnicas empleadas en CFD. La diferencia radica, como se ha comentado en otras ocasiones, en la relación entre el algoritmo numérico y los principios de conservación de la física del problema.
- **Discretización** de las ecuaciones integrales para generar un sistema de ecuaciones algebraico. Como se verá más adelante, existen numerosas técnicas que implementan los códigos CFD para el tratamiento de los términos convectivos, difusivos y transitorios.
- **Resolución** del sistema de ecuaciones algebraico mediante el uso de métodos iterativos. Existe toda una rama de las matemáticas dedicada al estudio de algoritmos que optimizan la resolución de sistemas de ecuaciones algebraicos. No obstante, este análisis queda fuera del alcance de este documento.

En efecto, la prevalencia de este método frente a otras técnicas CFD estriba en la integración sobre los volúmenes de control, estableciendo la conservación de las magnitudes de interés en cada célula que compone el mallado. Así, el algoritmo empleado no es más que una manifestación de los principios físicos de conservación subyacentes, lo que facilita la comprensión de este método en comparación con los métodos diferenciales o de elementos finitos. La conservación de una variable fluidica genérica (que en adelante se denominará por la letra  $\phi$ ) en un volumen de control puede entenderse como un balance neto entre los procesos que tienden a incrementar la variable y los que tienden a disminuirla. Sin pérdida de generalidad, este balance puede expresarse (con palabras) de la siguiente forma (véase [4]):

$$\begin{array}{ccccccc} \text{Tasa de cambio} & & \text{Incremento neto} & & \text{Incremento neto} & & \text{Incremento neto} \\ \text{de } \phi \text{ con respecto} & & \text{de } \phi \text{ debido a} & & \text{de } \phi \text{ debido a} & & \text{de } \phi \text{ debido a} \\ \text{al tiempo en el} & = & \text{convección en el} & + & \text{difusión en el} & + & \text{generación en el} \\ \text{volumen de control} & & \text{volumen de control} & & \text{volumen de control} & & \text{volumen de control} \end{array}$$

Dentro de los códigos CFD se encuentran embebidos las técnicas y modelos apropiados para el tratamiento de los **fenómenos de transporte**, a decir: **convección** (transporte asociado al flujo), **difusión** (transporte asociado a las variaciones de  $\phi$  entre diferentes puntos), **generación** (transporte asociado a la creación o destrucción de la magnitud  $\phi$ ) y el transporte asociado a la **tasa de cambio con respecto al tiempo**. Estos modelos usualmente requieren resolver sistemas lineales mediante métodos iterativos. Son muchos y variados los algoritmos que proporcionan resultados adecuados a la hora de resolver este tipo de sistemas, y su desarrollo abarcaría un capítulo completo. Por ello, no se ahondará en el presente informe en el estudio de dichos algoritmos, sino que se empleará el comando `\` de MATLAB para resolver los sistemas del tipo  $Ax = b \Rightarrow x = A \backslash b$ . Esto puede hacerse sin cavilaciones puesto que MATLAB posee algoritmos internos para detectar automáticamente la estructura de la matriz  $A$  y aplicar el método más apropiado.

### 1.4.3 Post-procesador

Una vez llevada a cabo la simulación, el paso último del análisis CFD consiste en la adquisición, tratamiento e interpretación de la información obtenida. Para poder extraer conclusiones relevantes en relación con el estudio llevado a cabo es de vital importancia la elección de las variables que se quieren graficar. De forma parecida a lo comentado en el pre-procesamiento, el post-procesador deberá contar con aptitudes similares, disponiendo de potentes herramientas gráficas de visualización

de datos que faciliten la interpretación de los resultados y la exportación de los mismos para su uso más allá de los códigos. Así, es imprescindible que el post-procesador cuente con la capacidad de manipular las superficies 2D o 3D generadas y mostrar su mallado o representar los resultados mediante campos vectoriales apoyados en códigos de colores o isolíneas, entre otras funciones.

Aunque, como ya se ha comentado, más del 50% del tiempo empleado en la industria de los códigos CFD se consume en la generación del mallado y la definición de la geometría, esta actividad compete enteramente al usuario y estará condicionada por la experiencia y habilidad que éste presente en el manejo de pre-procesadores y post-procesadores CFD. Como este asunto queda fuera de alcance, en este documento se estará particularmente interesado en el *solver*, por ser el elemento en el que se encuentran embebidos los algoritmos que permiten la resolución numérica del problema.

## 1.5 Ecuación de transporte

Las ecuaciones que gobiernan el flujo de fluidos se infieren de las ecuaciones de conservación paradigmáticas de la física:

- **Conservación de la masa.** La masa de fluido se conserva.
- **Conservación de la cantidad de movimiento.** La variación de la cantidad de movimiento es igual a la suma de todas las fuerzas en la partícula fluida (Segunda Ley de Newton).
- **Conservación de la energía.** La variación de la energía es igual a la suma de las variaciones de calor y trabajo en la partícula fluida (Primera Ley de la Termodinámica).

El desarrollo físico-matemático de dichas expresiones aplicado a una partícula fluida lleva a la formulación del sistema de ecuaciones que gobierna el problema tridimensional de transferencia de flujo y calor en fluidos newtonianos compresibles, las extendidamente conocidas ecuaciones de Navier-Stokes. Dicha formulación es válida siempre que se conciba el fluido como continuo. Esto es, la descripción del comportamiento del fluido según sus propiedades macroscópicas (densidad, velocidad, presión, temperatura...), ignorando los efectos de estructura y movimiento molecular. Así, una partícula fluida será el elemento de fluido más pequeño posible cuyas propiedades macroscópicas no se ven influenciadas por el comportamiento molecular.

Sin entrar en su demostración, las ecuaciones de Navier-Stokes presentan la siguiente forma para un fluido newtoniano compresible (véase [2]):

**Ecuación de continuidad:**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \quad (1.1)$$

**Ecuación de cantidad de movimiento:**

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho \vec{v} \cdot \nabla \vec{v} = -\nabla p + \nabla \cdot (\mu \nabla \vec{v}) + \rho \vec{f}_m \quad (1.2)$$

**Ecuación de la energía:**

$$\rho \frac{\partial e}{\partial t} + \rho \vec{v} \cdot \nabla e = -p \nabla \cdot \vec{v} + \underline{\underline{\tau}}' : \nabla \vec{v} + Q_r + Q_q + \nabla \cdot (k \nabla T) \quad (1.3)$$

Adicionalmente bajo la suposición de fluido ideal y equilibrio termodinámico, se tiene:

**Ecuación de los gases ideales:**

$$p = p(\rho, T) \quad e = e(\rho, T) \quad \rightarrow \quad \frac{p}{\rho} = R_g T \quad e = c_v T \quad (1.4)$$

En estas expresiones,  $t$  denota el tiempo,  $\vec{f}_m$  es el vector de fuerzas másicas,  $c_v$  es la capacidad calorífica del gas a volumen constante,  $k$  su conductividad térmica,  $\underline{\underline{\tau}}': \nabla \vec{v}$  es el término de disipación de energía cinética interna,  $Q_r$  y  $Q_q$  son las potencias caloríficas que, por unidad de volumen, recibe el fluido por radiación y reacción química, respectivamente,  $R_g = \frac{R}{M_m} = c_p - c_v = c_v(\gamma - 1)$ ,  $R$  es la constante universal de los gases,  $c_p$  la capacidad calorífica del gas a presión constante,  $e$  es la energía interna del fluido,  $M_m$  la masa molecular del gas y  $\gamma$  el coeficiente de dilatación adiabática (para el aire,  $R_g \approx 287 \frac{J}{KgK}$  y  $\gamma = 1.4$ ).

Así, se conforma un sistema de siete ecuaciones (debido a que la ecuación de cantidad de movimiento es una ecuación vectorial que equivale a tres ecuaciones escalares) y siete incógnitas ( $p, T, \rho, \vec{v}, e$ ) a las que, adicionalmente, habrá que imponer unas condiciones iniciales y de contorno adecuadas.

Debido a la similitud que presentan entre sí las ecuaciones de Navier-Stokes, es posible reescribirlas en una única ecuación que representa la evolución de una magnitud o propiedad genérica, denotada por  $\phi$ . Se llega así a lo que se conoce como **ecuación de transporte** para la propiedad  $\phi$ :

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\phi\vec{v}) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi \quad (1.5)$$

En el primer miembro aparecen los términos de variación temporal y transporte convectivo, y en el segundo los términos de transporte difusivo y generación, respectivamente. Esta ecuación constituye un punto de partida para los algoritmos computacionales convencionales en el FVM. El paso clave de este método consiste en integrar dicha expresión en un volumen de control tridimensional (en el caso más genérico), resultando:

$$\int_{VC} \frac{\partial(\rho\phi)}{\partial t} dV + \int_{VC} \nabla \cdot (\rho\phi\vec{v}) dV = \int_{VC} \nabla \cdot (\Gamma\nabla\phi) dV + \int_{VC} S_\phi dV \quad (1.6)$$

Reescribiendo los términos convectivo y difusivo como integrales de superficie cerrada (aquella que contiene al volumen de control) mediante la aplicación del Teorema de la Divergencia de Gauss, la ecuación (1.6) puede reescribirse como:

$$\frac{\partial}{\partial t} \left( \int_{VC} \rho\phi dV \right) + \int_A \vec{n} \cdot (\rho\phi\vec{v}) dA = \int_A \vec{n} \cdot (\Gamma\nabla\phi) dA + \int_{VC} S_\phi dV \quad (1.7)$$

Esta expresión tiene un claro significado físico que puede ser fácilmente interpretado analizando individualmente cada uno de los términos. El primero de ellos representa la tasa de cambio total de la cantidad de propiedad  $\phi$  que hay en el volumen de control. El producto escalar  $\vec{n} \cdot (\rho\phi\vec{v})$  expresa la componente de flujo de  $\phi$  en la dirección de la normal exterior de la superficie que encierra al volumen de control. Así, el segundo término está asociado al decremento neto de  $\phi$  en el elemento fluido debido a convección. El flujo difusivo es por definición positivo en la dirección negativa del gradiente, por lo que el producto  $\vec{n} \cdot (\Gamma\nabla\phi) = -\vec{n} \cdot (-\Gamma\nabla\phi)$  puede interpretarse como la componente del flujo difusivo positivo en la dirección del vector normal interior de la superficie del volumen de control. Por tanto, el primer término del segundo miembro representa el incremento neto de  $\phi$  en el elemento fluido debido a difusión. El último término de la expresión simboliza el incremento de  $\phi$  debido a las fuentes de generación de  $\phi$  internas en el elemento fluido. Esto pone de manifiesto con mayor claridad lo que ya se había declarado con anterioridad: la integración de la ecuación de transporte equivale a una ecuación de conservación de la propiedad fluida  $\phi$  en

un volumen de control macroscópico y finito. Este elemento fluido puede caracterizarse según la siguiente figura:

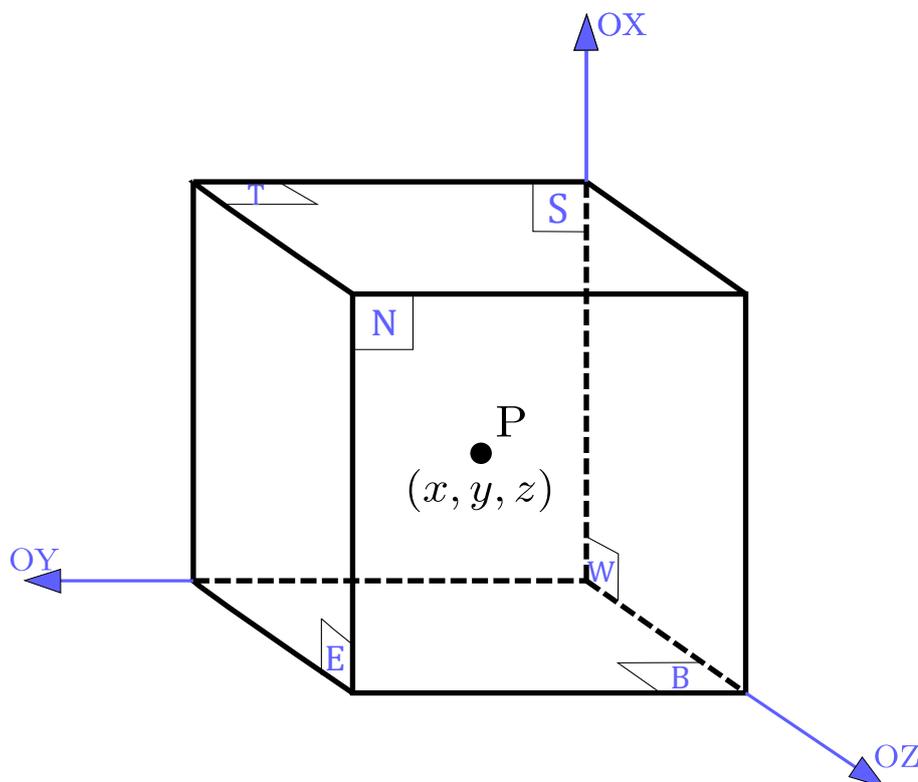


Figura 1.4 Definición del volumen de control.

El caso más general posible es un volumen de control cúbico acotado por seis caras. Siguiendo la nomenclatura de obras como [1] o [4], se denotarán estas caras por  $N$ ,  $S$ ,  $E$ ,  $W$ ,  $T$  y  $B$ , haciendo referencia a *North*, *South*, *East*, *West*, *Top* y *Bottom*, respectivamente. Esta denominación se hace según la dirección positiva de los ejes cartesianos, también representada en la Figura 1.4. Cuando se haga referencia a una celda dentro de un mallado, se usará  $P$  para nombrar al nodo de la celda en cuestión,  $N$ ,  $S$ ,  $E$ ,  $W$ ,  $T$  y  $B$  para denotar a los nodos de las celdas vecinas, y  $n$ ,  $s$ ,  $e$ ,  $w$ ,  $t$  y  $b$  para las caras entre ellas. Es interesante resaltar que  $P = (x, y, z)$  estará ubicado normalmente en el **baricentro** del volumen de control de forma que la magnitud de dicha celda sea lo más representativa posible. Cabe destacar además que todas las propiedades fluidas son función del tiempo y del espacio, por lo que estrictamente hablando habría que escribir  $\phi = \phi(x, y, z, t)$  en todo momento. No obstante, por economía de la escritura, se dará por hecho dicha dependencia funcional sin necesidad de expresarla. En los capítulos venideros se aplicarán las expresiones desarrolladas a problemas convectivo-difusivos prácticos y se obtendrán y analizarán los resultados.

Es necesario destacar la fuerte influencia que en adelante tendrán las obras de Versteeg ([4]) y Mazumder ([6]) a la hora de explicar los principales capítulos de este proyecto. Sin embargo, una aportación importante de este TFG ha sido el desarrollo de programas en MATLAB propios que permiten refinar arbitrariamente el mallado en los problemas y ejemplos mostrados. Los códigos más representativos se han incluido en la memoria, mientras que los que pueden resultar redundantes o captan menos fielmente la esencia del FVM están contenidos en el Apéndice ???. Además, para poder comparar los resultados de los programas propios con los de la literatura, se han reproducido en ocasiones algunos de los problemas (con los mismos datos) que se incluyen en estas obras.



## 2 FVM para problemas convectivo-difusivos en 1D

---

Una vez definida la naturaleza de la ecuación de transporte se va a desarrollar el método numérico basado en su integración, a decir, el FVM. Primeramente va a considerarse el proceso de transporte más simple posible: difusión estacionaria pura en el caso unidimensional. Posteriormente y de forma progresiva, se irán introduciendo los términos convectivos y no estacionarios hasta llegar al estudio de un problema de carácter general para el caso unidimensional. Es de vital importancia resaltar la fuerte influencia que ha tenido la obra de Versteeg y Malalasekera ([4]) en el orden de exposición del capítulo, las ilustraciones mostradas y los ejemplos exhibidos.

### 2.1 Problema difusivo, estacionario y unidimensional

En el caso estacionario y en ausencia de convección, la integración de la ecuación de transporte queda:

$$\int_A \vec{n} \cdot \left( \Gamma \frac{d\phi}{dx} \vec{e}_x \right) dA + \int_{VC} S_\phi dV = 0 \quad (2.1)$$

Para caracterizar y resolver el problema hay que realizar fundamentalmente tres pasos: generación del mallado, discretización de las ecuaciones y resolución del sistema. A continuación se describirán sucintamente cada una de estas etapas. Aunque para ilustrar esta explicación servirá de ejemplo el caso estacionario sin convección, este desarrollo es extrapolable al resto de problemas. La razón es que la estructura, que no es más que un mero reflejo de la que se expuso en el Capítulo 1.4, es idéntica en todos los problemas.

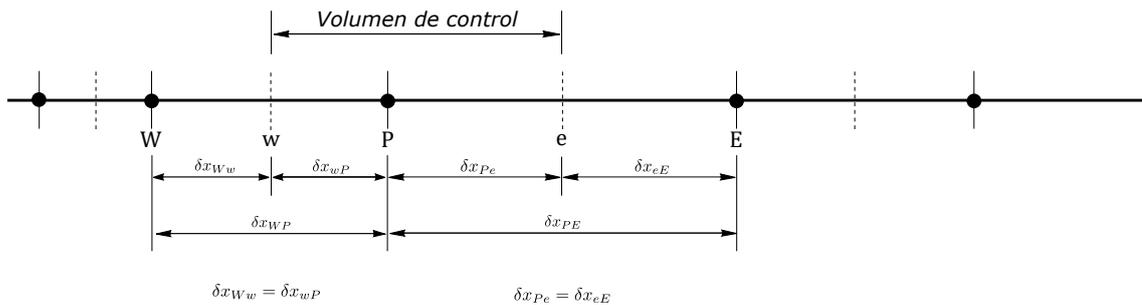
#### 2.1.1 Mallado del dominio unidimensional

En primer lugar, es necesario dividir el dominio en volúmenes de control finitos en los cuales discretizar las ecuaciones. En el caso unidimensional, cada uno de ellos quedará completamente caracterizado por un nodo y dos caras laterales. En casos multidimensionales, de forma tradicional bastará con un nodo y cuatro o seis caras laterales si se utilizan cuadriláteros o hexaedros, respectivamente. Es habitual definir los volúmenes de control ubicados en los extremos del dominio de forma que sus caras laterales coincidan con los límites físicos del dominio, facilitando así la implementación de las condiciones de contorno.

Existen distintas prácticas para disponer estos elementos según se quiera generar un mallado uniforme o no uniforme. Como ya se ha comentado anteriormente, lo más usual en la resolución

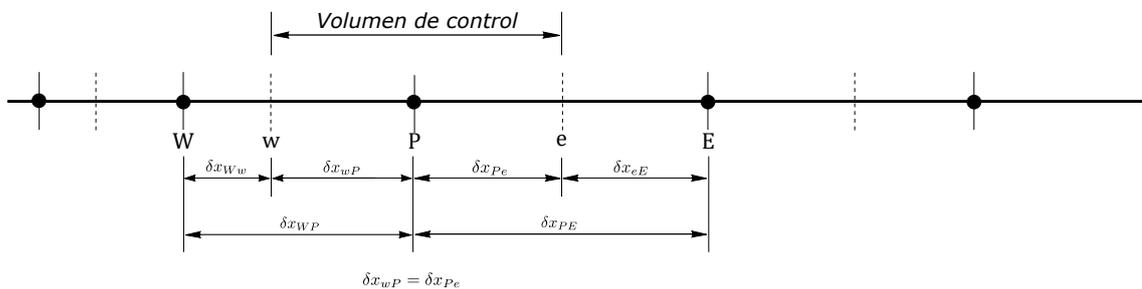
de la mayoría de problemas en ingeniería es el empleo de mallas no estructuradas, con mayor refinamiento en regiones donde las variaciones de la variable en estudio son más acusadas. Aún en mallas estructuradas, la disposición de las caras y los nodos puede prescindir de homogeneidad en busca de una malla más eficiente, refinada en zonas con gradientes elevados. Citando [1], son dos los procedimientos más habituales para la generación de mallas no homogéneas:

- **Procedimiento A.** Se definen primero los nodos y después se localizan las caras de los volúmenes de control en los puntos medios entre nodos. Este procedimiento se ilustra en la Figura 2.1.



**Figura 2.1** Mallado no homogéneo según el procedimiento A.

- **Procedimiento B.** Se ubican primero las caras de los volúmenes de control y después se colocan los nodos en los puntos medios entre caras. Este procedimiento se ilustra en la Figura 2.2.



**Figura 2.2** Mallado no homogéneo según el procedimiento B.

Es más recomendable optar por la práctica B en detrimento del primer procedimiento cuando se quiere trabajar con mallas no uniformes. La justificación de dicha elección, junto con un análisis detallado de las ventajas y desventajas de ambos métodos, puede encontrarse en [1]. Sin pormenorizar los detalles, puede decirse que en la práctica A los métodos de aproximación mediante diferencias centradas para computar los valores de una propiedad en las caras son más precisos que en la práctica B, pues las caras se encuentran en los puntos medios entre nodos. Por otro lado, en la práctica B los valores de  $\phi$  en los nodos son más representativos al localizarse el nodo en el centro de la celda, cosa que no sucede en la práctica A. No obstante, por simplicidad, los capítulos venideros se desarrollarán en base a una malla uniforme (nodos equiespaciados).

### 2.1.2 Discretización de las ecuaciones

El siguiente paso consiste en particularizar la ecuación integral de transporte para cada uno de los volúmenes de control, resultando:

$$\int_{\Delta V} \frac{d}{dx} \left( \Gamma \frac{d\phi}{dx} \right) dV + \int_{\Delta V} S dV = \Gamma_e A_e n_e \left. \frac{d\phi}{dx} \right|_e + \Gamma_w A_w n_w \left. \frac{d\phi}{dx} \right|_w + \bar{S} \Delta V = 0 \quad (2.2)$$

Esta expresión pone de manifiesto la característica antológica y de fácil interpretación física que diferencia al método de los volúmenes finitos del resto de métodos numéricos y que ya ha sido citada en ocasiones anteriores; la expresión (2.2) representa una **ecuación de balance** en el volumen de control.

Nótese que, aunque la solución del problema se obtiene en los nodos, éstas ecuaciones discretizadas requieren evaluar los parámetros característicos del sistema y los gradientes en las caras de los volúmenes de control. Para ello, es necesario desarrollar un modelo que permita aproximar correctamente estos valores. La forma más simple de acometer esta tarea pasa por utilizar una **interpolación lineal** entre nodos para determinar los valores en las caras. Definiendo las funciones de interpolación  $g_w$  y  $g_e$ , estos valores pueden expresarse de la siguiente manera:

$$\Gamma_w = g_w \Gamma_P + (1 - g_w) \Gamma_W \quad \text{donde} \quad g_w = \frac{x_w - x_W}{x_P - x_W} \quad (2.3)$$

$$\Gamma_e = g_e \Gamma_P + (1 - g_e) \Gamma_E \quad \text{donde} \quad g_e = \frac{x_E - x_e}{x_E - x_P} \quad (2.4)$$

En el caso de trabajar con una malla uniforme,  $g_w = g_e = 0.5$  y el valor en las caras no será más que la semisuma de los valores en los nodos adyacentes a la cara. Es decir:

$$\Gamma_w = \frac{\Gamma_P + \Gamma_W}{2} \quad \Gamma_e = \frac{\Gamma_P + \Gamma_E}{2} \quad (2.5)$$

Asimismo, para el caso de los términos difusivos se empleará lo que se conoce como **método de las diferencias centradas**, una aproximación de segundo orden (solo si la malla es uniforme) que da buenos resultados gracias a la propia naturaleza de la difusión, que se transporta homogéneamente en todas direcciones. Así, estos términos pueden expresarse como:

$$\left. \frac{d\phi}{dx} \right|_w = \frac{\phi_P - \phi_W}{x_P - x_W} \quad \left. \frac{d\phi}{dx} \right|_e = \frac{\phi_E - \phi_P}{x_E - x_P} \quad (2.6)$$

Introduciendo estas expresiones en la ecuación (2.2) se obtiene:

$$-\Gamma_e A_e n_e \frac{\phi_E - \phi_P}{x_E - x_P} - \Gamma_w A_w n_w \frac{\phi_P - \phi_W}{x_P - x_W} = \bar{S} \Delta V \quad (2.7)$$

Definiendo  $D_e = \Gamma_e A_e n_e / (x_E - x_P)$  y  $D_w = \Gamma_w A_w n_w / (x_P - x_W)$  como términos asociados al flujo difusivo y reordenando se llega a:

$$D_w \phi_W + (D_e - D_w) \phi_P - D_e \phi_E = \bar{S} \Delta V \quad (2.8)$$

Que puede expresarse como:

$$a_W \phi_W + a_P \phi_P + a_E \phi_E = a_{SP} \quad (2.9)$$

El término  $a_{SP}$  depende del modelo de fuente que se utilice. Es importante resaltar que esta ecuación discretizada representa la expresión típica alrededor de la cual girarán todos los modelos que se desarrollarán a lo largo de la memoria.

### 2.1.3 Resolución del sistema

La expresión genérica a la que se ha llegado en el subapartado anterior es aplicable a todos los nodos internos (aquéllos cuyas caras no coinciden con los límites físicos del problema). Para los volúmenes de control limítrofes, la ecuación general debe modificarse para incluir las condiciones de contorno del problema en cuestión. Así, se llega a un sistema de ecuaciones algebraico lineal de la forma  $Ax = b$  ó  $A\phi = b$  cuya solución será el valor de la propiedad  $\phi$  en cada uno de los nodos. A continuación se describe el ensamblaje de la matriz de coeficientes  $A$  y el vector de términos independientes  $b$ .

En primer lugar, se define el índice  $I$  para recorrer las celdas del dominio, de forma que para las celdas interiores ( $I = 2 : N - 1$ ) debe verificarse la siguiente expresión:

$$a_W(I) \phi_W(I) + a_P(I) \phi_P(I) + a_E(I) \phi_E(I) = a_{SP}(I) \quad (2.10)$$

Esto puede representarse en términos de la matriz  $A$  y del vector de incógnitas  $\phi$  de la siguiente forma:

$$A(I, I-1) \phi(I-1) + A(I, I) \phi(I) + A(I, I+1) \phi(I+1) = b(I) \quad (2.11)$$

En la ecuación (2.11) se ha tenido en cuenta que  $a_W(I) = A(I, I-1)$ ,  $a_P(I) = A(I, I)$ ,  $a_E(I) = A(I, I+1)$ ,  $\phi_W(I) = \phi(I-1)$ ,  $\phi_P(I) = \phi(I)$ ,  $\phi_E(I) = \phi(I+1)$  y  $a_{SP}(I) = b(I)$ . Así, puede decirse que cada fila de la matriz  $A$  representa una celda, y los valores no nulos de cada fila representan o la influencia de dicha celda sobre sí misma (elementos diagonales de  $A$ ) o de las celdas adyacentes sobre la celda en cuestión (elementos extradiagonales de  $A$ ).

Las celdas fronterizas del dominio requieren un tratamiento particular para incluir las condiciones de contorno. A lo largo de esta memoria únicamente se van a considerar dos tipos de condiciones de contorno: la **condición de contorno de Dirichlet**, que se corresponde con la situación de  $\phi$  impuesto, y la **condición de contorno de Neumann**, que equivale a imponer un flujo ( $\partial\phi/\partial x$ ). Existe una tercera denominada **condición de contorno de Robin**, que no es más que una composición de las dos anteriores, y que no se incluirá en esta memoria por simplicidad (hace engorrosas las ecuaciones).

Por tanto, para la condición de contorno izquierda ( $I = 1$ ), se tiene la siguiente expresión:

$$A(1, 1) \phi(1) + A(1, 2) \phi(2) = b(1) \quad (2.12)$$

Según el escenario, los coeficientes simbolizarán unas expresiones u otras. Para el caso de temperatura impuesta, se tiene que  $A(1, 1) = a_P(1)$ ,  $A(1, 2) = a_E(1)$  y  $b(1) = a_{SP}(1) - a_W(1) \phi_L$ . Por contra, para la situación de flujo impuesto se tendría que  $A(1, 1) = a_W(1) + a_P(1)$ ,  $A(1, 2) = a_E(1)$  y  $b(1) = a_{SP}(1) + a_W(1) q_L/D_w(1)$ .

Por otro lado, para la condición de contorno derecha ( $I = N$ ), se tiene la siguiente expresión:

$$A(N, N-1) \phi(N-1) + A(N, N) \phi(N) = b(N) \quad (2.13)$$

De nuevo, los coeficientes adoptarán unas expresiones u otras según el escenario. Para el caso de temperatura impuesta, se tiene que  $A(N, N-1) = a_W(N)$ ,  $A(N, N) = a_P(N)$  y  $b(N) = a_{SP}(N) - a_E(N) \phi_R$ . Por contra, para la situación de flujo impuesto se tendría que  $A(N, N-1) = a_W(N)$ ,  $A(N, N) = a_P(N) + a_E(N)$  y  $b(N) = a_{SP}(N) - a_E(N) q_R/D_e(N)$ .

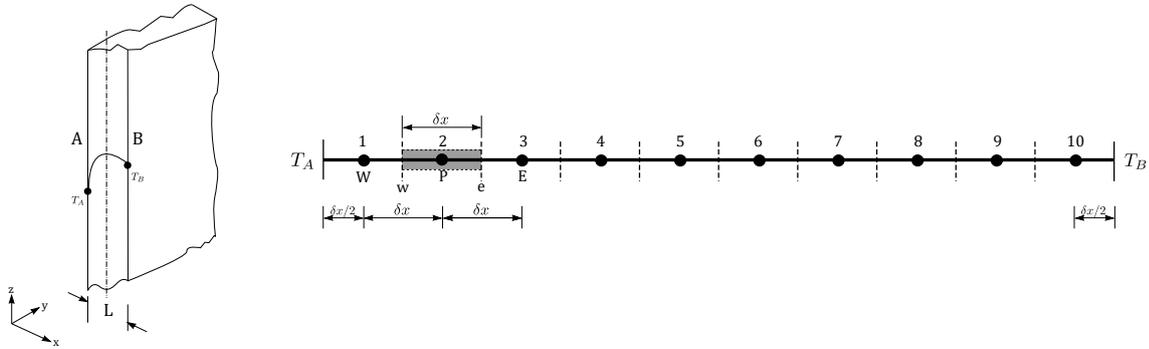
Las expresiones (2.11)-(2.13) tomadas de forma conjunta dan lugar a un sistema de ecuaciones lineales tridiagonal cuya solución son los valores  $\phi(1), \phi(2), \dots, \phi(N)$ , que denotan los valores de

$\phi$  en los nodos de las celdas  $I = 1 : N$ . Existe una rama completa de las matemáticas dedicada específicamente a la resolución de un sistema matricial de esta naturaleza. No obstante, las diferentes técnicas aplicables quedan fuera del alcance de este proyecto. Se empleará el comando \ de MATLAB para resolver este tipo de sistemas.

**2.1.4 Aplicación práctica 1D**

Como aplicación ilustrativa del FVM en este capítulo, se considera como problema paradigmático (basado en [4]) uno de los casos más sencillos posible, y que se describe a continuación.

Considérese una placa de espesor  $L = 3\text{ cm}$  como la mostrada en la Figura 2.3, donde se asume que sus dimensiones a lo largo de las coordenadas  $y$  y  $z$  son lo suficientemente grandes como para tener en cuenta únicamente los gradientes en la dirección  $x$ . Se supondrá que la placa está fabricada en un material cuya conductividad térmica es  $k = 0.75\text{ W/mK}$ , que una resistencia interna proporciona una generación volumétrica de calor de valor  $q = 1500\text{ kW/m}^3$  y que las paredes  $A$  y  $B$  se encuentran a un temperatura de  $T_A = 50\text{ C}^\circ$  y  $T_B = 250\text{ C}^\circ$ , respectivamente.



**Figura 2.3** Representación del problema descrito.

En estas condiciones, se sabe que la ecuación que gobierna este problema es la siguiente:

$$\frac{d}{dx} \left( k \frac{dT}{dx} \right) + q = 0 \tag{2.14}$$

Considérese el dominio dividido en diez volúmenes de control uniformemente distribuidos. Integrando sobre cada volumen de control se obtiene:

$$\int_{\Delta V_i} \frac{d}{dx} \left( k \frac{dT}{dx} \right) dV + \int_{\Delta V_i} S dV = k_e A_e n_e \frac{dT}{dx} \Big|_e + k_w A_w n_w \frac{dT}{dx} \Big|_w + q \Delta V_i = 0 \tag{2.15}$$

Nótese que el resultado es idéntico al que se obtuvo en la formulación general con la salvedad de que el coeficiente de difusión ( $\Gamma$ ), la propiedad escalar ( $\phi$ ) y el término de generación media ( $\bar{S}$ ) genéricos han sido sustituidos por el coeficiente de conductividad térmica ( $k$ ), la temperatura ( $T$ ) y la generación volumétrica ( $q$ ), respectivamente. Aproximando los gradientes mediante interpolaciones lineales se llega a la siguiente expresión:

$$-k_e A_e n_e \frac{T_E - T_P}{x_E - x_P} - k_w A_w n_w \frac{T_P - T_W}{x_P - x_W} = q \Delta V_i \tag{2.16}$$

Para este problema en particular se tiene que  $k_e = k_w = k$ ,  $A_e = A_w = A$ ,  $n_e = 1$  y  $n_w = -1$  por lo que reordenando y definiendo  $D_e = kA/(x_E - x_P)$ ,  $D_w = -kA/(x_P - x_W)$  y  $\Delta V_i = A(x_e - x_w)$  de forma equivalente al caso genérico presentado en el subapartado anterior, se obtiene:

$$D_w T_W + (D_e - D_w) T_P - D_e T_E = q \Delta V_i \quad (2.17)$$

Que puede expresarse como:

$$a_W T_W + a_P T_P + a_E T_E = a_{SP} \quad (2.18)$$

Esta ecuación será válida para los nodos intermedios ( $i = 2, 3, \dots, 9$ ). Para los nodos  $i = 1$  e  $i = 10$  será necesario implementar las condiciones de contorno. Para la frontera izquierda ( $i = 1$ ), se tiene que:

$$-k_e A_e n_e \frac{T_E - T_P}{x_E - x_P} - k_A A_A n_A \frac{T_P - T_A}{x_P - x_A} = q \Delta V_1 \quad (2.19)$$

Reordenando términos y simplificando según las condiciones del problema:

$$\left( \frac{kA}{x_E - x_P} + \frac{kA}{x_P - x_A} \right) T_P - \frac{kA}{x_E - x_P} T_E = q \Delta V_1 + \frac{kA}{x_P - x_A} T_A \quad (2.20)$$

Que puede expresarse de forma equivalente al caso de nodos intermedios, dando lugar a la ecuación discretizada que permite calcular las variables en el primer nodo:

$$a_{P_1} T_P + a_{E_1} T_E = a_{SP_1} - a_{W_1} T_A \quad (2.21)$$

De forma similar, para la frontera derecha ( $i = 10$ ), se tiene que:

$$-k_B A_B n_B \frac{T_B - T_P}{x_B - x_P} - k_w A_w n_w \frac{T_P - T_W}{x_P - x_W} = q \Delta V_{10} \quad (2.22)$$

Reordenando términos y simplificando según las condiciones del problema:

$$\left( \frac{kA}{x_B - x_P} + \frac{kA}{x_P - x_W} \right) T_P - \frac{kA}{x_P - x_W} T_W = q \Delta V_{10} + \frac{kA}{x_B - x_P} T_B \quad (2.23)$$

Que puede expresarse de forma equivalente al caso de nodos intermedios, dando lugar a la ecuación discretizada que permite calcular las variables en el último nodo:

$$a_{W_{10}} T_W + a_{P_{10}} T_P = a_{SP_{10}} - a_{E_{10}} T_B \quad (2.24)$$

Una vez obtenido el conjunto de ecuaciones discretizadas que caracterizan cada uno de los nodos, se está en condiciones de resolver el sistema lineal por cualquiera de los métodos convencionales y, en el caso particular de esta memoria, a través del comando \ de MATLAB concebido para dicha finalidad.

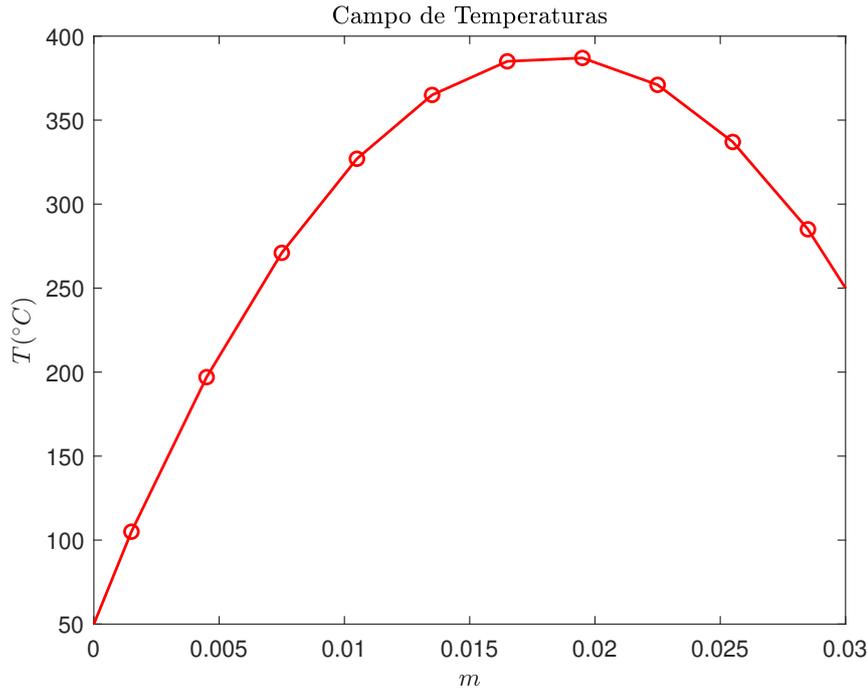


Figura 2.4 Solución proporcionada por el FVM para el problema descrito.

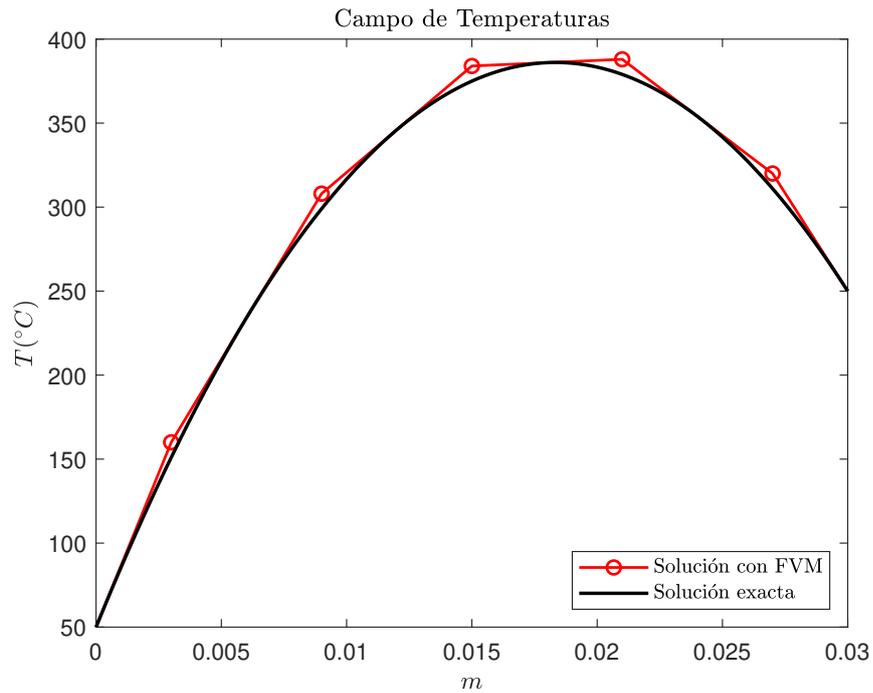
Para evaluar la precisión del método desarrollado lo ideal es comparar los resultados obtenidos numéricamente con la solución analítica del problema. Esto no siempre es posible, pues según la naturaleza del problema puede existir o no dicha solución. No obstante, la ecuación diferencial (2.14) que gobierna el problema planteado es sencilla de resolver por los métodos tradicionales. Integrando dos veces con respecto de  $x$  y aplicando las condiciones de contorno, se llega a:

$$T(x) = \left[ \frac{T_B - T_A}{L} + \frac{q}{2k} (L - x) \right] x + T_A \tag{2.25}$$

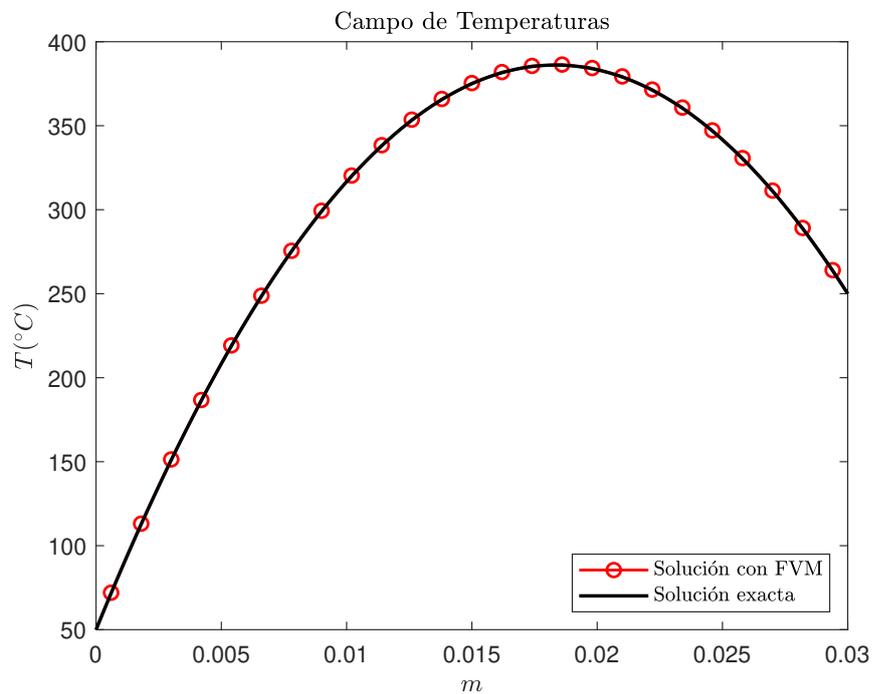
Se muestra a continuación una tabla que contiene el error relativo cometido con la solución numérica que proporciona el FVM (discretizando el dominio en 5 volúmenes fluidos), así como una comparación gráfica. Puede observarse que, incluso empleando un mallado poco refinado la precisión que se obtiene es más que aceptable. Si se quisiese mejorar la precisión de los resultados, bastaría con aumentar el número de volúmenes de control, tal y como muestra la Figura 2.6.

Tabla 2.1 Valores del campo de temperaturas y error relativo cometido por el FVM empleando 5 nodos equiespaciados.

Nodo	1	2	3	4	5
Solución FVM (°C)	160	308	384	388	320
Solución Exacta (°C)	151	299	375	379	311
Error Relativo (%)	5.9603	3.0100	2.4000	2.3747	2.8939



**Figura 2.5** Comparación entre la solución exacta y la obtenida mediante el FVM empleando 5 nodos equiespaciados.

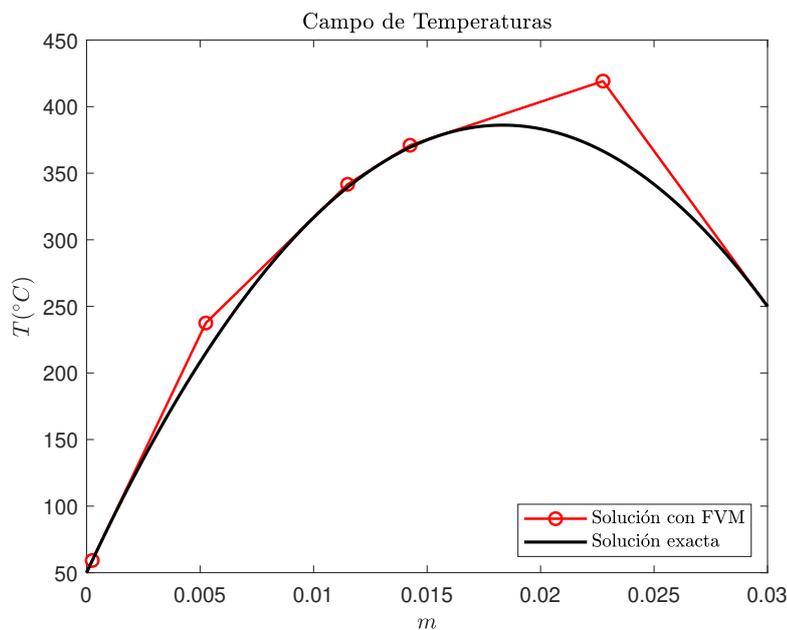


**Figura 2.6** Comparación entre la solución exacta y la obtenida mediante el FVM empleando 25 nodos equiespaciados.

Estos resultados han sido obtenidos con un número  $N$  de nodos distribuidos de forma equiespaciada. No obstante, para ilustrar el concepto de mallas no homogéneas presentado en la Sección 2.1.1, se exponen a continuación dos generaciones de malla alternativas para resolver el problema. Ambas serán obtenidas mediante el procedimiento B descrito en la Figura 2.2. En primer lugar, considérese el caso en que la ubicación de las caras se establece de forma aleatoria a lo largo del dominio. En tal escenario, se obtienen los siguientes resultados:

**Tabla 2.2** Valores del campo de temperaturas y error relativo cometido por el FVM empleando 5 nodos ubicados aleatoriamente.

Nodo	1	2	3	4	5
Solución FVM (°C)	56.55	238.35	351.16	380.72	400.91
Solución Exacta (°C)	56.51	214.55	347.23	378.49	359.08
Error Relativo (%)	0.0564	11.0918	1.1336	0.5877	11.6493

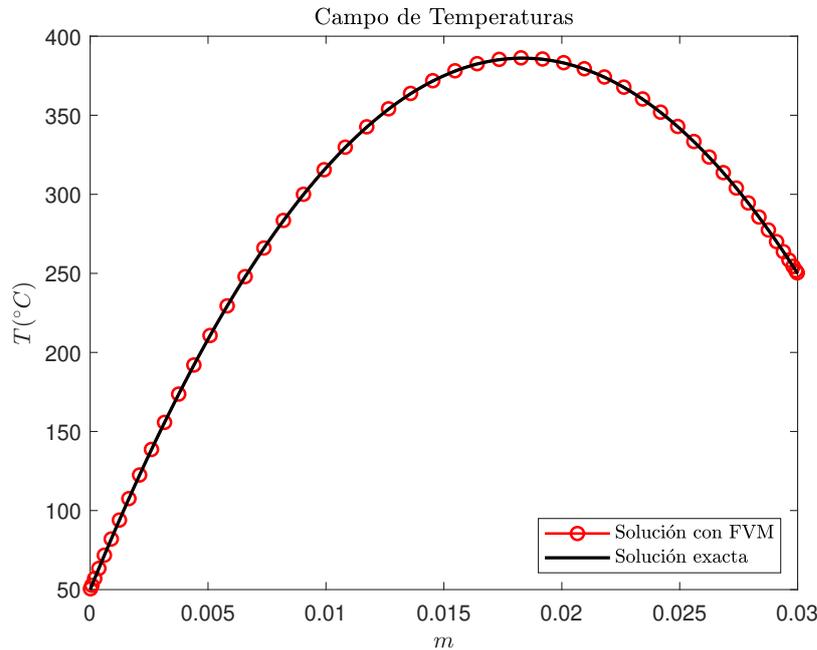


**Figura 2.7** Comparación entre la solución exacta y la obtenida mediante el FVM empleando 5 nodos ubicados aleatoriamente.

Si se comparan los valores presentados en la Tabla 2.1 y la Tabla 2.2 puede comprobarse que una distribución arbitraria dará como resultado, por lo general, un error absoluto máximo mayor que en el caso homogéneo, siendo por tanto preferible elegir esta distribución frente a la aleatoriamente inhomogénea.

Otra opción para generar un mallado no uniforme consiste en realizar el cambio de variable  $x = L/2(1 - \cos \theta)$ , donde  $x$  denota la posición de las caras y  $\theta$  se malla uniformemente en el intervalo  $[0, \pi]$ . Este cambio de variable tiene como consecuencia la generación de una malla con mayor densidad de nodos en los extremos del dominio, lo cual puede ser beneficioso para

reproducir correctamente posibles gradientes elevados de  $\phi$  en dichos límites (por ejemplo, su uso es fundamental en el estudio de la **capa límite**). Esto se ejemplifica a continuación:



**Figura 2.8** Comparación entre la solución exacta y la obtenida mediante el FVM empleando 50 nodos con mayor concentración en los extremos.

Para que sirva de ejemplo al lector, se introduce a continuación el código MATLAB desarrollado para resolver el problema descrito. En dichas líneas puede apreciarse fácilmente la construcción de la matriz  $A$  según lo establecido en la Sección 2.1.3, así como la generación de la malla, el post-procesamiento y la gestión de la información.

```

1 %% PROBLEMA DE DIFUSION PURA
2
3 clear all; close all; clc
4
5 TablaErrores = [];
6
7 % MALLADO
8
9 L = 0.03; % Longitud placa (m)
10 N = 5; % Número de Nodos
11
12 caras = linspace(0,L,N+1); % Nodos Equiespaciados
13
14 % caras_theta = linspace(0,pi,N+1); % Nodos Concentrados en Extremos
15 % caras = L/2*(1 - cos(caras_theta));
16
17 % rng = [0,L*rand(1,N-1),L]; % Nodos Aleatorios
18 % caras = sort(rng);
19
20 for i = 2:length(caras)
21     nodos(i-1) = (caras(i) - caras(i-1))/2 + caras(i-1);
22 end
23

```

```

24 % DATOS
25
26 Gamma_w = 0.75*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
27 Gamma_e = 0.75*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
28 Aw = 1*ones(N,1); % Área Transversal (m^2) -> CTE
29 Ae = 1*ones(N,1); % Área Transversal (m^2) -> CTE
30 nw = -ones(N,1); % Vector Normal (nw=-1 cte para caras west)
31 ne = ones(N,1); % Vector Normal (ne=1 cte para caras east)
32 Fw = zeros(N,1); % Flujo (F=0 por ser problema de difusión pura)
33 Fe = zeros(N,1); % Flujo (F=0 por ser problema de difusión pura)
34 q = 1500e3*ones(N,1); % Generación Volumétrica de Calor (W/m^3)
35 TA = 50+273; % CdC Izquierda: phi impuesto (K)
36 TB = 250+273; % CdC Derecha: phi impuesto (K)
37
38 % COEFICIENTES
39
40 for I = 2:N-1
41     gw(I) = (caras(I) - nodos(I-1))/(nodos(I) - nodos(I-1)); % Factor Interpolación
42     ge(I) = (caras(I+1) - nodos(I+1))/(nodos(I) - nodos(I+1)); % Factor Interpolación
43     Dw(I) = Gamma_w(I)*nw(I)*Aw(I)/(nodos(I) - nodos(I-1));
44     De(I) = Gamma_e(I)*ne(I)*Ae(I)/(nodos(I+1) - nodos(I));
45     aW(I) = Fw(I)*(1 - gw(I)) + Dw(I); % Coeficiente phi_W
46     aP(I) = Fw(I)*gw(I) + Fe(I)*ge(I) - Dw(I) + De(I); % Coeficiente phi_P
47     aE(I) = Fe(I)*(1 - ge(I)) - De(I); % Coeficiente phi_E
48     aSP(I) = q(I)*Ae(I)*(caras(I+1) - caras(I)); % Término Asociado a la Generación
49     A(I,I-1) = aW(I);
50     A(I,I) = aP(I);
51     A(I,I+1) = aE(I);
52     b(I) = aSP(I);
53 end
54
55 % CONDICIONES DE CONTORNO
56
57 % LEFT
58
59 phiL = TA; % Temperatura en Cara Izquierda (K)
60 gw(1) = 0;
61 ge(1) = (caras(2) - nodos(2))/(nodos(1) - nodos(2));
62 Dw(1) = Gamma_w(1)*nw(1)*Aw(1)/(nodos(1) - caras(1));
63 De(1) = Gamma_e(1)*ne(1)*Ae(1)/(nodos(2) - nodos(1));
64 aW(1) = Fw(1)*(1 - gw(1)) + Dw(1);
65 aP(1) = Fw(1)*gw(1) + Fe(1)*ge(1) - Dw(1) + De(1);
66 aE(1) = Fe(1)*(1 - ge(1)) - De(1);
67 aSP(1) = q(1)*Ae(1)*(caras(2) - caras(1));
68 A(1,1) = aP(1);
69 A(1,2) = aE(1);
70 b(1) = aSP(1) - aW(1)*phiL;
71
72 % RIGHT
73
74 phiR = TB; % Temperatura en Cara Derecha (K)
75 gw(N) = (caras(N) - nodos(N-1))/(nodos(N) - nodos(N-1));
76 ge(N) = 0;
77 Dw(N) = Gamma_w(N)*nw(N)*Aw(N)/(nodos(N) - nodos(N-1));
78 De(N) = Gamma_e(N)*ne(N)*Ae(N)/(caras(end) - nodos(N));
79 aW(N) = Fw(N)*(1 - gw(N)) + Dw(N);
80 aP(N) = Fw(N)*gw(N) + Fe(N)*ge(N) - Dw(N) + De(N);
81 aE(N) = Fe(N)*(1 - ge(N)) - De(N);
82 aSP(N) = q(N)*Ae(N)*(caras(end) - caras(end-1));
83 A(N,N-1) = aW(N);
84 A(N,N) = aP(N);
85 b(N) = aSP(N) - aE(N)*phiR;

```

```

86
87 % SOLUCION
88
89 phi = A\b' - 273
90 phiexact = @(x)((TB - TA)/L + q(1)/2/Gamma_e(1)*(L-x)).*x+TA - 273);
91 plot([0,nodos,L],[TA - 273;phi;TB - 273],'ro-', 'MarkerIndices',2:length(nodos) + 1, '
    linewidth',1.2)
92 hold on
93 x = 0:0.0002:L;
94 plot(x,phiexact(x),'k','linewidth',1.5)
95 hold on
96 xlabel('$m$', 'Interpreter','latex')
97 ylabel('$T(\circ C)$', 'Interpreter','latex')
98 title('Campo de Temperaturas', 'Interpreter','latex')
99 legend({'Soluci\'on con FVM', 'Soluci\'on exacta'}, 'Interpreter','latex', 'Location','
    Southeast')
100
101 for i = 1:N
102     TablaErrores = [TablaErrores;i,phi(i),phiexact(nodos(i)),...
103         abs((phi(i) - phiexact(nodos(i)))/phiexact(nodos(i)))*100];
104 end
105
106 disp('     Nodo   Sol. FVM Sol. Exacta  Error(%)')
107 disp(' ')
108 disp(TablaErrores)

```

En el siguiente capítulo, se introducirán los efectos de advección (más comúnmente conocido como convección) a los problemas de flujo y se implementarán y discutirán las diferentes técnicas de modelización que existen a la hora de implementarlos en el FVM.

## 2.2 Problema convectivo-difusivo, estacionario y unidimensional. Esquemas de discretización

Hasta ahora se han estudiado únicamente los problemas difusivos, aquellos en los que no existe flujo de fluido. Sin embargo, en la naturaleza son de mucha mayor abundancia los sistemas en los que difusión y convección ocurren simultáneamente, por lo que surge la necesidad de desarrollar nuevos métodos que permitan predecir correctamente ambos efectos en consonancia y aumentar la aplicabilidad del FVM. Por ello, en adelante se analizarán sistemas en los que el flujo de fluido juega un papel significativo y se describirán las distintas técnicas que pueden emplearse para tratar los términos convectivos que, como se verá, dan lugar en ocasiones a inestabilidades en la solución.

La integración de la ecuación de transporte para el caso convectivo-difusivo estacionario y unidimensional sobre el volumen de control (que representa un balance de flujo sobre éste) resulta:

$$\int_A \vec{n} \cdot (\rho \phi \vec{v}) dA = \int_A \vec{n} \cdot \left( \Gamma \frac{d\phi}{dx} \vec{e}_x \right) dA + \int_{VC} S_\phi dV \quad (2.26)$$

El principal problema en la discretización de los términos convectivos de la ecuación radica en la determinación de la propiedad  $\phi$  en las caras de los volúmenes de control y en el cálculo de los flujos a través de éstas. En el análisis del problema difusivo se introdujo el método de las diferencias centradas para evaluar los términos difusivos en las caras y se demostró que funcionaba bien incluso para un bajo número de celdas, por lo que puede resultar lógico extender su uso a los términos convectivos. No obstante, existe una diferencia importante que limita la extensión de este método y que se describe a continuación.

Mientras que en los procesos difusivos la distribución de la propiedad  $\phi$  es transportada de forma homogénea en todas las direcciones, en los procesos de convección la influencia se limita a la dirección del flujo. Como se comprobará más adelante, esta disimilitud se materializa en una restricción en el número de celdas necesarias para obtener una solución válida si se usa el método de las diferencias centradas para modelar la convección. Dicho número está sujeto a la importancia relativa que tengan los términos convectivos frente a los difusivos, siendo necesario un mallado más fino para tratar fuertes convecciones. Es por ello que se presentarán métodos alternativos para obtener soluciones estables sin necesidad de un mallado tan restrictivo, que no siempre es posible. Adicionalmente, se supondrán conocidos los campos de velocidades en las caras de los volúmenes de control y, por tanto, los flujos.

La forma de resolver el problema es conceptualmente idéntica al caso de difusión pura, por lo que se omitirá el desarrollo exhaustivo de las fases para evitar ser redundantes. Se compone de los mismos tres pasos: mallado, discretización y resolución. Una vez definidos los volúmenes de control, integrando sobre cada uno de ellos se llega a la siguiente ecuación:

$$\rho_w u_w A_w n_w \phi_w + \rho_e u_e A_e n_e \phi_e - \Gamma_e A_e n_e \frac{\phi_E - \phi_P}{x_E - x_P} - \Gamma_w A_w n_w \frac{\phi_P - \phi_W}{x_P - x_W} = \bar{S} \Delta V \quad (2.27)$$

Donde se ha utilizado la diferenciación central para tratar los gradientes difusivos. Además, el flujo debe cumplir la ecuación de continuidad (ecuación de conservación). Es decir:

$$\int_A \vec{n} \cdot (\rho \vec{v}) dA = 0 \quad \Rightarrow \quad F_w + F_e = 0 \quad (2.28)$$

Para obtener el sistema de ecuaciones discretizadas que gobierna el problema de forma más compacta resulta conveniente definir  $F_w = \rho_w u_w A_w n_w$  y  $F_e = \rho_e u_e A_e n_e$  como términos asociados al flujo convectivo y  $D_w = \Gamma_w A_w n_w / (x_P - x_W)$  y  $D_e = \Gamma_e A_e n_e / (x_E - x_P)$  asociados al transporte difusivo. Así, la ecuación que debe satisfacerse en cada nodo será:

$$F_e \phi_e + F_w \phi_w - D_e (\phi_E - \phi_P) - D_w (\phi_P - \phi_W) = \bar{S} \Delta V \quad (2.29)$$

Que puede expresarse fácilmente como la ecuación (2.18) sin más que reagrupar términos.

Las próximas secciones se dedicarán al desarrollo, análisis, implementación y valoración de algunos de los distintos métodos que eminentemente se utilizan en los códigos comerciales para evaluar la propiedad  $\phi$  en las paredes de los volúmenes de control. Se señalarán las ventajas y desventajas de cada uno de ellos, y las limitaciones que han llevado a la necesidad de recurrir a nuevos métodos. Para dar cierta linealidad a la exposición y siendo fiel a [4], éstos serán presentados en el orden lógico en que fueron surgiendo.

### 2.2.1 Esquema de las diferencias centradas

Este método ya ha sido presentado anteriormente, pues es el que se está usando por defecto para evaluar los términos difusivos. Su fundamento no es otro que el de interpolar linealmente los valores de los nodos adyacentes a la cara que se quiere evaluar. Para una malla no uniforme, las expresiones serían análogas a las expuestas en (2.3) y (2.4), siendo necesario el cálculo de las funciones de interpolación  $g_w$  y  $g_e$ . Para un mallado uniforme, las expresiones que resultan son nuevamente la semisuma de los valores:

$$\phi_w = \frac{\phi_W + \phi_P}{2} \quad \phi_e = \frac{\phi_P + \phi_E}{2} \quad (2.30)$$

Sustituyendo dichas expresiones en la ecuación (2.29), se obtiene:

$$\left(\frac{F_w}{2} + D_w\right)\phi_W + \left(\frac{F_w + F_e}{2} + D_e - D_w\right)\phi_P + \left(\frac{F_e}{2} - D_e\right)\phi_E = \bar{S}\Delta V \quad (2.31)$$

Puede reconocerse fácilmente la misma estructura general que en (2.18).

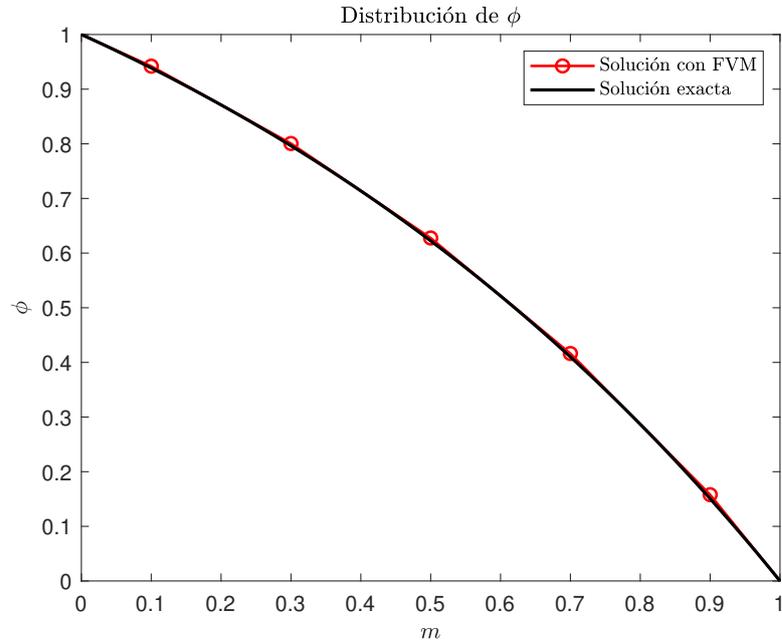
A modo ilustrativo y para que sirva de comparación entre los diferentes métodos, en adelante se tomará como problema de convección-difusión paradigmático el que se describe a continuación, esencialmente idéntico al presentado en [4]. Sea una propiedad  $\phi$  transportada por medio de difusión de cantidad de movimiento (difusión viscosa) y convección a través de un dominio unidimensional de longitud  $L = 1 m$ . La ecuación que gobierna este sistema será:

$$\frac{d}{dx}(\rho u \phi) = \frac{d}{dx}\left(\Gamma \frac{d\phi}{dx}\right) \quad (2.32)$$

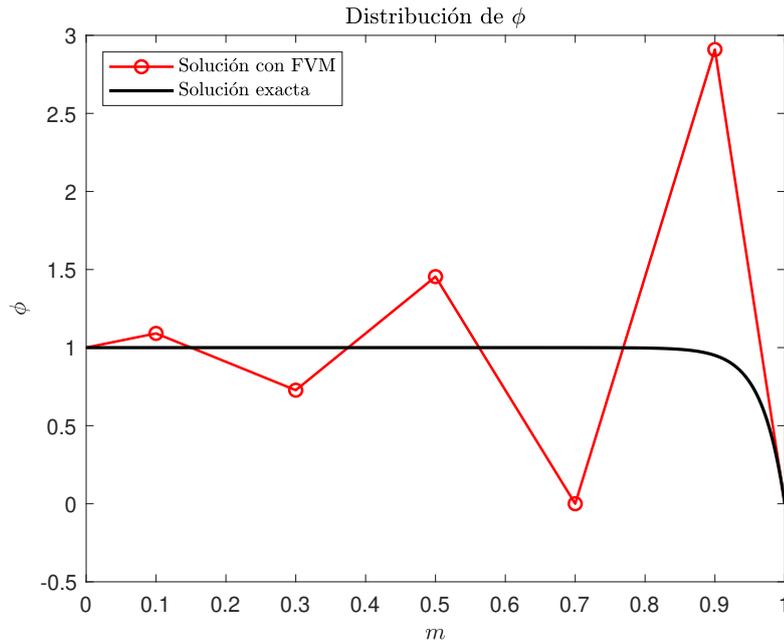
Sujeto a las condiciones de contorno  $\phi_0 = 1$  en  $x = 0$  y  $\phi_L = 0$  en  $x = L$ . La velocidad se considerará uniforme y de valor conocido  $u \in [0.1, 3] (m/s)$ . Como se verá más adelante, el valor de  $u$  será determinante para caracterizar la aplicabilidad de los distintos métodos, pues a mayor velocidad, mayor será la convección. La difusión vendrá caracterizada por  $\Gamma = 0.1 kg/(ms)$ . Además, se considerará que el fluido en cuestión es agua ( $\rho = 1 kg/m^3$ ). Utilizando las herramientas adecuadas de la teoría de resolución de ecuaciones diferenciales ordinarias, puede demostrarse que la solución analítica de dicha ecuación es:

$$\phi(x) = \phi_0 + (\phi_L - \phi_0) \frac{e^{\rho u x / \Gamma} - 1}{e^{\rho u L / \Gamma} - 1} \quad (2.33)$$

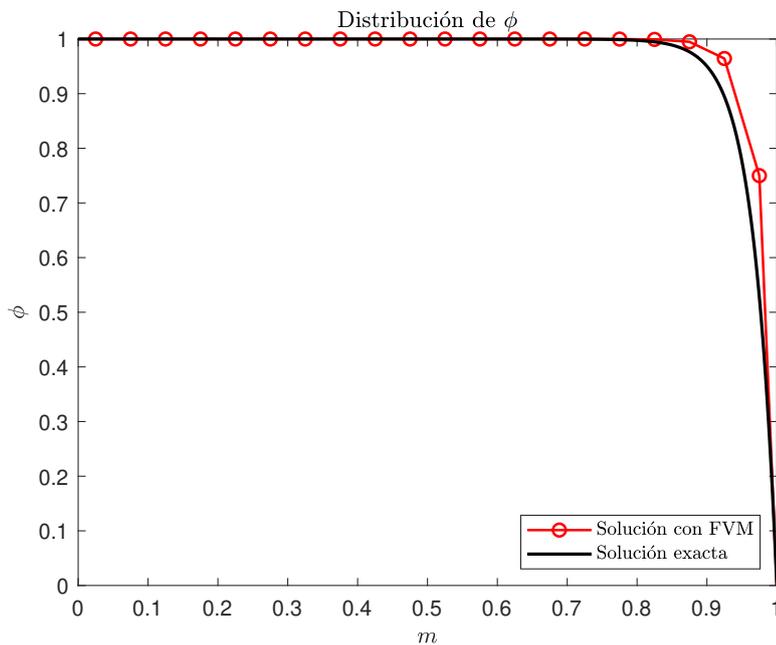
Se muestra a continuación una comparación entre la solución exacta y la solución que proporciona el FVM (utilizando diferenciación central para tratar los términos convectivos y difusivos en las caras) bajo distintas condiciones.



**Figura 2.9** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 0.1 m/s$  empleando 5 nodos equiespaciados y usando el método de las diferencias centradas.



**Figura 2.10** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 3 \text{ m/s}$  empleando 5 nodos equiespaciados y usando el método de las diferencias centradas.



**Figura 2.11** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 3 \text{ m/s}$  empleando 20 nodos equiespaciados y usando el método de las diferencias centradas.

Son varios los comentarios interesantes que pueden hacerse acerca de estas representaciones. En primer lugar nótese cómo, para un valor bajo de la velocidad del campo fluido, un mallado poco refinado es suficiente para obtener una buena aproximación de la solución. A medida que la velocidad se incrementa, el término convectivo prepondera sobre el difusivo, apareciendo unas

**oscilaciones** en la solución aproximada que la hacen diferir notablemente de la solución analítica. Para resolver esta problemática, puede comprobarse cómo un aumento en el número de celdas es suficiente para que la solución numérica sea razonablemente parecida a la exacta.

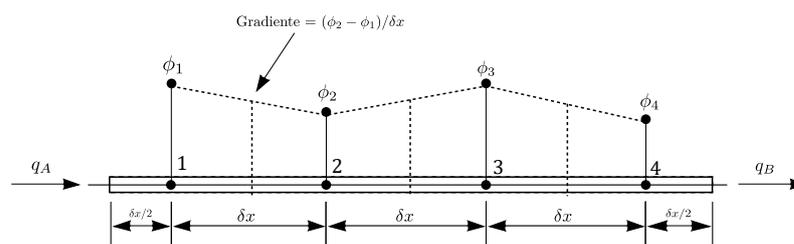
La conclusión que puede inferirse de este análisis es que a medida que el flujo convectivo cobra importancia sobre el difusivo, que es equivalente a decir que la relación  $F/D$  aumenta, la diferenciación centrada produce resultados cada vez menos precisos. Esta disparidad se materializa en oscilaciones sobre la solución exacta, que pueden atenuarse si se disminuye el ratio  $F/D$ , por ejemplo, refinando la malla. La razón de que aparezcan estas oscilaciones y la influencia que tiene en ello el factor  $F/D$  será tratado a continuación.

### 2.2.2 Propiedades de los esquemas de discretización

Para poder desarrollar el resto de métodos de discretización de forma inteligible, es necesario estudiar en profundidad las razones por las que la diferenciación centrada no proporciona buenos resultados bajo determinadas condiciones en problemas convectivo-difusivos. Más aún, aunque los resultados numéricos deben coincidir exactamente con la solución analítica para un número de celdas infinitamente grande (independientemente del método de discretización utilizado), esto no es posible en la práctica. La necesidad de obtener resultados físicamente realistas incluso con un número limitado de celdas induce a analizar ciertas propiedades que subyacen a los métodos de discretización y que éstos deben tener. Algunas de estas propiedades fundamentales son: **conservatividad, delimitación y transportividad**. Estas características serán explicadas a continuación. Destacar que la exposición y las ilustraciones han sido fuertemente influenciadas por [4].

#### Conservatividad

Tal y como se ha comentado en ocasiones anteriores, la conservatividad es uno de los pilares que sustenta la formulación del FVM. La integración de la ecuación de transporte sobre cada volumen de control lleva a un conjunto de ecuaciones de conservación de la propiedad  $\phi$ . Para garantizar dicha conservación, tanto global como localmente, el flujo de  $\phi$  saliente a través de una determinada cara del volumen de control debe ser idéntico al flujo entrante en el volumen de control adyacente a través de la cara fronteriza entre ambos. Esto se consigue representando los valores de flujo de manera consistente y unívoca. Para ejemplificar esto, considérese el siguiente problema de difusión pura sin generación, estacionario y unidimensional:



**Figura 2.12** Problema paradigmático para ejemplificar la propiedad de conservatividad.

Empleando las diferencias centradas para tratar los gradientes difusivos y sumando las ecuaciones discretizadas en cada volumen de control, se llega a la ecuación de balance neto global:

$$\left[ \Gamma_{e_1} \frac{\phi_2 - \phi_1}{\delta x} - q_A \right] + \left[ \Gamma_{e_2} \frac{\phi_3 - \phi_2}{\delta x} - \Gamma_{w_2} \frac{\phi_2 - \phi_1}{\delta x} \right] + \left[ \Gamma_{e_3} \frac{\phi_4 - \phi_3}{\delta x} - \Gamma_{w_3} \frac{\phi_3 - \phi_2}{\delta x} \right] + \left[ q_B - \Gamma_{w_4} \frac{\phi_4 - \phi_3}{\delta x} \right] = q_B - q_A \quad (2.34)$$

Teniendo en cuenta que  $\Gamma_{e_1} = \Gamma_{w_2}$ ,  $\Gamma_{e_2} = \Gamma_{w_3}$  y  $\Gamma_{e_3} = \Gamma_{w_4}$ , los flujos a través de las caras se cancelan dos a dos en el balance neto global, quedando únicamente los términos de flujo en los límites del dominio fluido ( $q_A$  y  $q_B$ ). Esta consistencia en la formulación asegura la conservación de la propiedad  $\phi$  cuando se usan las diferencias centradas. Puede ocurrir que una formulación con otro tipo de interpolación (por ejemplo, cuadrática) lleve a inconsistencias en la conservación. Los flujos en las caras pueden ser desiguales si los gradientes de las parábolas en dichas caras son distintos entre sí, de forma que los términos no se cancelarían y la conservación global no se satisfaría. No obstante, existen esquemas cuadráticos perfectamente consistentes y que se usan en la actualidad, como el *QUICK scheme*, aunque su estudio queda fuera del alcance de este proyecto.

### Delimitación

La resolución del sistema de ecuaciones algebraico se lleva a cabo por métodos iterativos que persiguen la convergencia de la solución. En el marco de este proyecto, solo interesa saber que dichos métodos están embebidos en el comando \ de MATLAB. Puede demostrarse que es condición suficiente para la convergencia del método iterativo en cuestión que se cumpla el **criterio de Scarborough**. Éste puede expresarse de la siguiente forma según el valor de los coeficientes del sistema:

$$\frac{\sum |a_{W,E}|}{|a_P|} \begin{cases} \leq 1 & \text{en todos los nodos} \\ < 1 & \text{en al menos un nodo} \end{cases} \quad (2.35)$$

En esta condición,  $a_P$  representa el valor del coeficiente que multiplica al nodo central, y  $a_{W,E}$  el del coeficiente que multiplica a todos los nodos de la vecindad (en unidimensional, éstos son únicamente el nodo Este y el nodo Oeste). Si dicha condición se verifica, la matriz de coeficientes resultante estará bien condicionada y será diagonal dominante, de forma que los métodos iterativos convergerán a la solución del sistema. Físicamente, esta condición establece que, en ausencia de fuentes, los valores de la variable  $\phi$  están delimitados por sus valores en el contorno.

Otra condición que debe cumplirse para garantizar una buena delimitación es que todos los coeficientes del sistema tengan el mismo signo (cuando se expresa de la forma  $a_P\phi_P = a_E\phi_E + a_W\phi_W$ ). Físicamente esto implica en que un aumento de  $\phi$  en un nodo se traduce en un aumento de  $\phi$  en los nodos vecinos.

Si el esquema de discretización en cuestión no cumple alguna de estas dos condiciones, la solución divergirá o presentará oscilaciones, proporcionando resultados poco precisos.

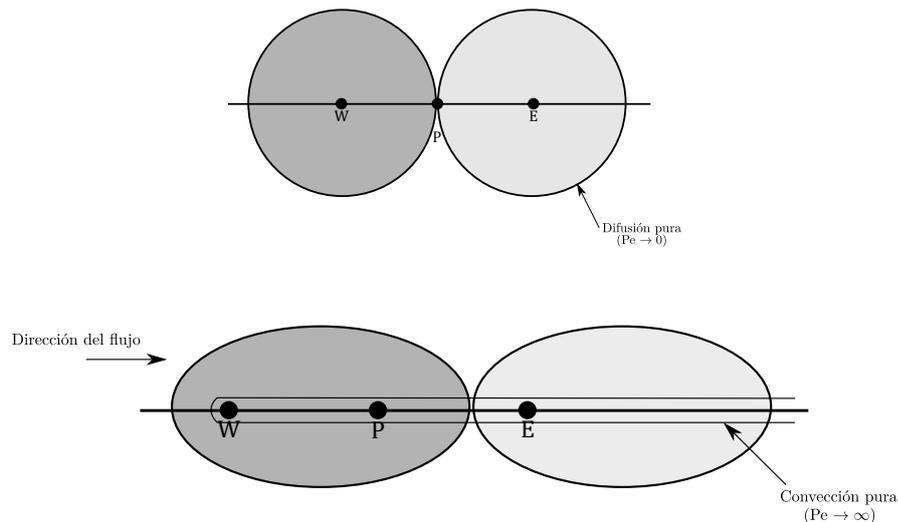
### Transportividad

Para analizar el concepto de transportividad de  $\phi$ , resulta interesante definir el conocido como **número de Peclet**, un número adimensional que mide la importancia relativa de la convección frente a la difusión en un determinado dominio fluido. Así:

$$Pe = \frac{F}{D} = \frac{\rho u A}{\Gamma A / \delta x} \quad (2.36)$$

Donde  $\delta x$  representa la longitud característica (ancho de la celda).

A continuación se representan esquemáticamente las líneas de  $\phi$  constante según diversos valores de Peclet. Para ilustrar este concepto, considérense los casos extremos: difusión pura ( $Pe \rightarrow 0$ ) y convección pura ( $Pe \rightarrow \infty$ ).



**Figura 2.13** Distribución de  $\phi$  en la vecindad de dos fuentes. En la figura superior, el caso límite  $Pe \rightarrow 0$ . En la figura inferior, el caso límite  $Pe \rightarrow \infty$ .

Para el caso en el que solo hay difusión ( $Pe \rightarrow 0$ ), el fluido estará remansado y las líneas de  $\phi$  constante describirán circunferencias centradas en los nodos, ya que el proceso de difusión se extiende equitativamente de forma radial. Así, puede afirmarse que la influencia que ejercen los nodos  $E$  y  $W$  sobre  $P$  es la misma. A medida que el número de  $Pe$  se incrementa, las líneas de  $\phi$  constante tienden a adoptar una forma elíptica y con centro arrastrado aguas abajo del nodo, tal y como puede observarse en la figura inferior. Así, cada nodo  $P$  se verá mayoritariamente influenciado por el nodo que se encuentre aguas arriba ( $W$  o  $E$  según el sentido de la corriente). En el límite de  $Pe \rightarrow \infty$  (sin difusión), toda la propiedad  $\phi$  que emane de los nodos  $W$  o  $E$  se verá inmediatamente convectada aguas abajo. De esta forma, en el caso de un flujo en sentido positivo, el nodo  $P$  no se verá afectado por el nodo  $E$ , sino que estará completamente condicionado por el nodo  $W$ . Tanto es así que, en ausencia de difusión, puede declararse que  $\phi_P = \phi_W$  y  $\phi_E = \phi_P$  para un flujo positivo. La consideración conjunta de la direccionalidad del flujo, y por tanto de la influencia, y la magnitud del número de Peclet es lo que se conoce como **transportividad**, y será una característica de vital importancia en la caracterización de los esquemas de discretización.

Una vez presentadas las propiedades que todo esquema de discretización debe tener para garantizar una convergencia adecuada, se está en condiciones de establecer una valoración objetiva de la precisión de cualquiera de los métodos. Comenzando por el método de las diferencias centradas, el único presentado hasta ahora, puede decirse que:

**a) Conservatividad.** El método de las diferencias centradas emplea expresiones consistentes para evaluar los flujos en las caras y asegurar que el balance neto de flujo se cumple, por lo que esta propiedad se satisface.

**b) Delimitación.** Si la expresión (2.31) se reordena de la forma  $a_P \phi_P = a_W \phi_W + a_E \phi_E$  y se tiene en cuenta que en el caso unidimensional  $n_e = 1$  y  $n_w = -1$ , puede comprobarse que  $a_P = a_E + a_W$ , por lo que el criterio de Scarborough se cumple.

La segunda condición para garantizar la delimitación es que todos los coeficientes de la ecuación discretizada tengan el mismo signo. Teniendo en cuenta las consideraciones anteriores, el valor de los coeficientes es el que sigue:

$$a_W = D_w + \frac{F_w}{2} \quad a_P = D_w + \frac{F_w}{2} + D_e - \frac{F_e}{2} + (F_e - F_w) \quad a_E = D_e - \frac{F_e}{2} \quad (2.37)$$

Por ser un método conservativo,  $F_e - F_w = 0$ , luego el único coeficiente que corre riesgo de ser negativo será  $a_E$ . Esto dependerá de cuán grande sea la contribución convectiva en dicho coeficiente. Como  $D_e > 0$  y  $F_e > 0$ , para que  $a_E$  sea positivo debe satisfacerse la siguiente condición:

$$\frac{F_e}{D_e} = \frac{\rho_e u_e A_e}{\Gamma_e / \delta x} = Pe_e < 2 \quad (2.38)$$

Así, valores de  $Pe_e$  mayores que 2 pueden llevar a soluciones que carezcan de sentido físico. Sujeto a los parámetros que caracterizan el número de Peclet de la celda, puede afirmarse que la condición (2.38) se cumple solo si se trata de un flujo difusivo-dominante a bajo número de  $Re$  (velocidad  $u$  pequeña) o si el mallado es suficientemente fino ( $\delta x$  pequeño). Esto último no siempre será posible, lo que limita la aplicación de este método a un número reducido de problemas.

**c) Transportividad.** En la formulación del método de las diferencias centradas la influencia que ejercen sobre  $P$  los flujos convectivo-difusivos es la misma para todos los nodos vecinos. De esta forma, no se está considerando la dirección del flujo ni la importancia relativa de la convección frente a la difusión. Puede concluirse por tanto que la propiedad de transportividad no es una característica del método de las diferencias centradas para altos  $Pe$ .

En compendio, este análisis pone de manifiesto que las propiedades de delimitación y transportividad pueden no cumplirse para números de  $Pe$  lo suficientemente elevados en el método de las diferencias centradas. A pesar de ser un método de segundo orden, no es un esquema de discretización favorable para numerosos problemas convectivo-difusivos de carácter general. Así, surge la necesidad de desarrollar nuevos métodos que permitan superar estas limitaciones. A continuación se presentan algunos de los más conocidos, y que están implementados en softwares comerciales de extendido uso como OpenFOAM o Ansys Fluent (véase [7]).

### 2.2.3 Esquema *upwind* de primer orden

El esquema *upwind* o FOU (acrónimo de su nombre en inglés, *First Order Upwind*) es un método de discretización que nace como solución a una de las mayores incompetencias del método de las diferencias centradas: su incapacidad de identificar la dirección del flujo. En dicho método el valor de la propiedad  $\phi$  en la cara de una celda está igualmente influenciado por el valor en los nodos colindantes, independientemente de las características del flujo. Por contra, el método *upwind* establece el valor convectivo de  $\phi$  en las caras igual al valor de  $\phi$  en el nodo aguas arriba. Así, se rechaza la influencia equitativa de ambos nodos y se tiene en cuenta la influencia dominante del nodo aguas arriba de la corriente. Esta forma de proceder es precisamente la que da nombre al método.

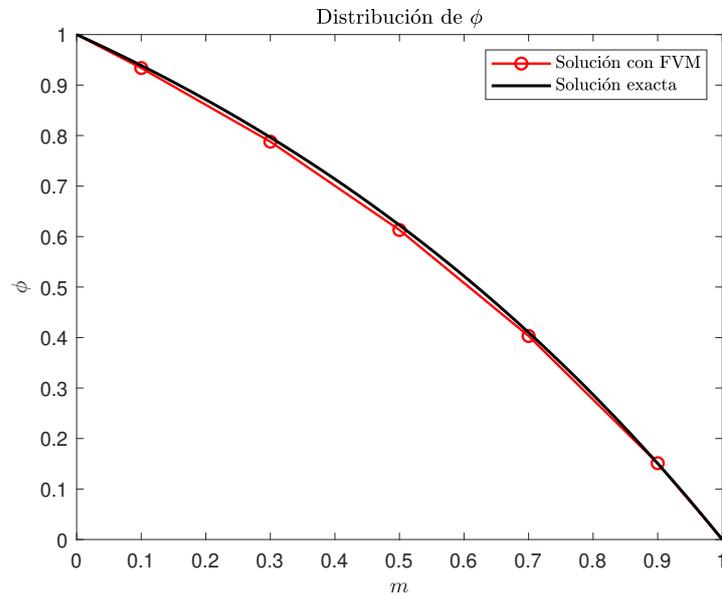
Matemáticamente, este modelo se expresa de la siguiente manera (destacar que para las condiciones en los flujos se considera  $F_w = \rho_w u_w A_w$  y  $F_e = \rho_e u_e A_e$ , sin tener en cuenta las normales):

$$\begin{cases} \phi_w = \phi_W, & \phi_e = \phi_P & \text{si } F_w > 0, & F_e > 0 \\ \phi_w = \phi_P, & \phi_e = \phi_E & \text{si } F_w < 0, & F_e < 0 \end{cases} \quad (2.39)$$

Esto lleva a la siguiente ecuación discretizada (para un flujo positivo):

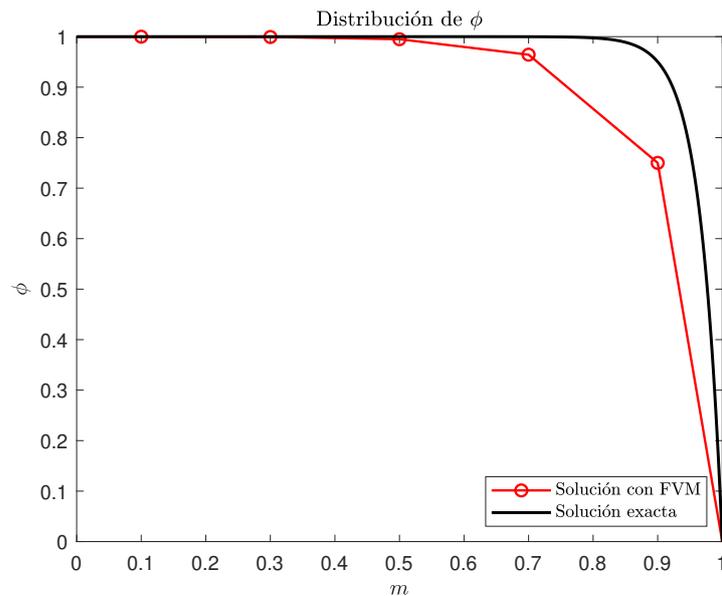
$$(F_w + D_w) \phi_W + (F_e + D_e - D_w) \phi_P + (-D_e) \phi_E = \bar{S} \Delta V \quad (2.40)$$

Para ilustrar este concepto, se aplicará el método al problema descrito en (2.32) y se contrastarán los resultados con aquéllos obtenidos mediante el método de las diferencias centradas. En primer lugar, para un flujo a baja velocidad se tiene lo siguiente:

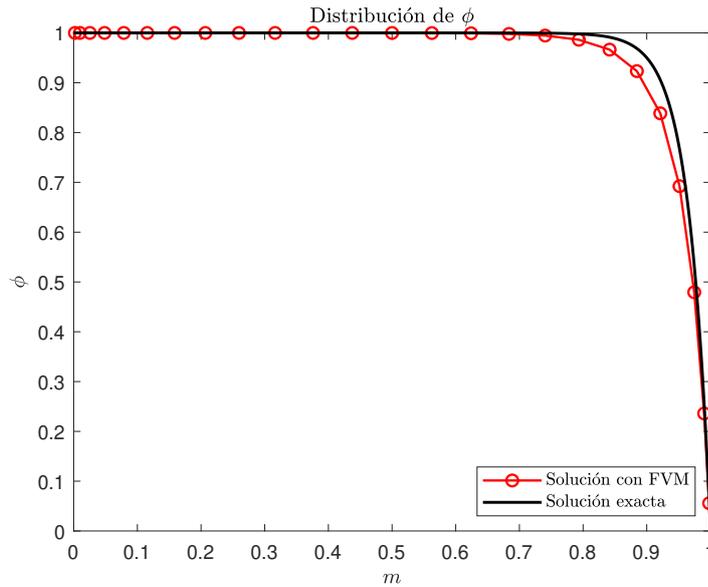


**Figura 2.14** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 0.1 \text{ m/s}$  empleando 5 nodos equiespaciados y usando el método *upwind* de primer orden.

Puede observarse cómo para una velocidad baja ambos métodos proporcionan resultados que se asemejan en gran medida a la solución analítica. Esto se debe a que la convección no juega un papel importante en el transporte de la propiedad  $\phi$ . Aumentando ahora la velocidad:



**Figura 2.15** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 3 \text{ m/s}$  empleando 5 nodos equiespaciados y usando el método *upwind* de primer orden.



**Figura 2.16** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 3 \text{ m/s}$  empleando 25 nodos (con mayor densidad en los extremos) y usando el método *upwind* de primer orden.

En este caso, el aumento sustancial de la velocidad tiene una clara consecuencia en los resultados obtenidos. Mientras que el método de las diferencias centradas presentaba oscilaciones en torno a la solución (véase la Figura 2.10), el método *upwind* se adapta mucho mejor a la solución, aunque está lejos de ésta en la frontera derecha. Esto puede resolverse, por ejemplo, refinando el mallado y concentrando los nodos en los extremos con el cambio de variable  $x = L/2(1 - \cos \theta)$ , tal y como se observa en la Figura 2.16.

Para analizar la evidente mejora que presenta este método frente a las diferencias centradas a elevados números de Peclet, se procede a valorar las características que el método *upwind* presenta como esquema de discretización. Nótese que dicha mejora resulta clara para el caso de 5 nodos, mientras que para un número elevado de nodos puede parecer *a priori* que el método de las diferencias centradas presenta una mayor precisión. No obstante, hay que tener en cuenta que no siempre será posible un refinamiento del mallado, de ahí la eminente ventaja del método *upwind*.

**a) Conservatividad.** Este método emplea las mismas expresiones consistentes que el método de las diferencias centradas para calcular los flujos en las caras. Por tanto, la formulación es conservativa.

**b) Delimitación.** Si para ejemplificar se toma el caso  $F_w > 0$  y  $F_e > 0$ , se reordena la expresión (2.40) de la forma  $a_P \phi_P = a_W \phi_W + a_E \phi_E$  y se tiene en cuenta que  $n_e = 1$  y  $n_w = -1$  en el caso unidimensional, el valor de los coeficientes es el siguiente:

$$a_W = D_w + F_w \quad a_P = D_e + D_w + F_e \quad a_E = D_e \quad (2.41)$$

Obsérvese que se verifica la relación  $a_P = a_W + a_E + (F_e - F_w)$ , y como por continuidad  $F_e - F_w = 0$ , puede afirmarse que el criterio de Scarborough se cumple. Además, los coeficientes son siempre positivos, por lo que todos los requerimientos necesarios para la delimitación se satisfacen. La matriz es diagonal dominante con todos los coeficientes positivos y por tanto la solución no presentará oscilaciones.

c) **Transportividad.** La formulación del método se fundamenta en la dirección de la corriente, definiendo la propiedad  $\phi$  en las caras como funciones a trozos que dependen directamente del sentido del flujo. Por tanto, a diferencia del método de las diferencias finitas, la transportividad sí se satisface.

En resumen, el método *upwind* es una práctica de uso generalizado en códigos CFD por su alta simplicidad, sus propiedades estabilizadoras y los buenos resultados que proporciona. No obstante, en problemas multidimensionales puede llevar a un concepto conocido como **falsa difusión**, que aparece en problemas puramente convectivos cuando la dirección del flujo no es paralela a los ejes coordenados. Este es uno de los mayores problemas en la resolución de ecuaciones convectivo-difusivas, y es un tema de actualidad por la controversia y la confusión que despierta en los expertos en la materia. Aunque este es un problema que puede resolverse con un mallado fino, el refinamiento necesario puede llegar a ser excesivamente caro. Además, para valores altos del número de Reynolds, la falsa difusión puede llevar a resultados sustancialmente incorrectos. También tiene la limitación de ser un método de primer orden, por lo que puede no ser lo suficientemente exacto para determinados problemas. Es por todo esto que surge la necesidad de centrar los esfuerzos en desarrollar nuevos métodos mejorados. Se presenta en el próximo capítulo el método *upwind* de segundo orden, más preciso y con capacidad de atenuar los efectos negativos de la falsa difusión.

Especial mención merece el **esquema híbrido**, un método de discretización que recoge conjuntamente los aspectos positivos del método de las diferencias centradas y el *upwind* de primer orden. Su formulación permite emplear unas expresiones u otras según el número de Peclet en las caras de las celdas, usando las diferencias centradas cuando  $Pe < 2$  y el *upwind* cuando  $Pe > 2$ . Esto lo convierte en una buena aproximación cuando el número de Peclet es muy grande o cercano a cero. No obstante, queda fuera del alcance de este proyecto desarrollar las expresiones en que se sustenta, pues son una combinación de las que ya se han visto y no aporta ninguna diferencia sustancial. Además, no es un método con un uso tan extendido como el resto.

#### 2.2.4 Esquema *upwind* de segundo orden

El esquema *upwind* de segundo orden o SOU (acrónimo de su nombre en inglés, *Second Order Upwind*) nace para combatir las limitaciones del *upwind* de primer orden. Con un orden más elevado, pretende ser un método más preciso y con una falsa difusión reducida, permitiendo obtener resultados satisfactorios sin necesidad de un mallado excesivamente pequeño. En este método, también capaz de identificar la dirección del flujo, no solo se tiene en cuenta la influencia de los nodos colindantes a una cara, sino la de nodos más lejanos. Así, se considerarán los nodos *WW* y *EE* para denotar a los nodos que se encuentran a izquierda del *W* y a derecha del *E*, respectivamente. Esta idea, extraída de [6], se ilustra en la siguiente figura:

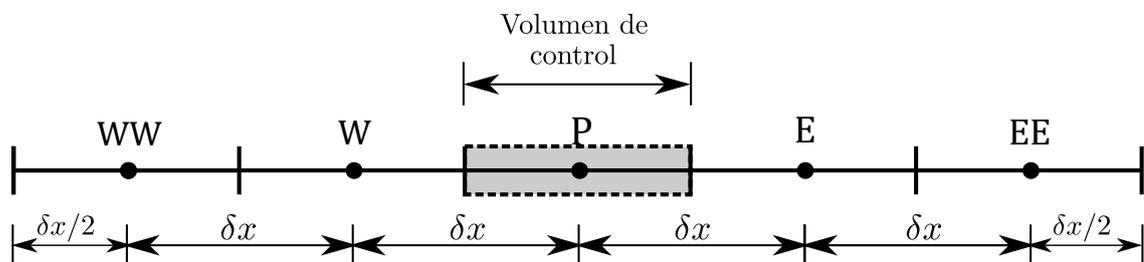


Figura 2.17 Nomenclatura de nodos en el método *upwind* de segundo orden.

Las bases de este método son eminentemente las mismas que en el esquema *upwind* de primer orden, pues también se sustenta en la identificación de la dirección del flujo y la consideración de

la influencia dominante de los nodos aguas arriba. En este caso, se emplean dos expansiones en serie de Taylor para aproximar el valor de  $\phi$  en las caras. La primera de ellas es para la primera celda aguas arriba, y la segunda para la inmediatamente posterior. En primer lugar, recuérdese la expresión para un polinomio de Taylor de orden  $n$  de la función  $f$  alrededor del punto  $a$ :

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n \quad (2.42)$$

Así, si se tiene por ejemplo un flujo positivo, se tendrían las siguientes expansiones para  $\phi_w$ :

$$\phi_w = \phi_w - \frac{\Delta x}{2} \left. \frac{d\phi}{dx} \right|_w + \frac{1}{2} \left( \frac{\Delta x}{2} \right)^2 \left. \frac{d^2\phi}{dx^2} \right|_w - \dots \quad (2.43)$$

$$\phi_{ww} = \phi_w - \frac{3\Delta x}{2} \left. \frac{d\phi}{dx} \right|_w + \frac{1}{2} \left( \frac{3\Delta x}{2} \right)^2 \left. \frac{d^2\phi}{dx^2} \right|_w - \dots \quad (2.44)$$

Multiplicando (2.43) por tres y restando (2.44) se llega a:

$$3\phi_w - \phi_{ww} = 2\phi_w - \frac{3\Delta x^2}{4} \left. \frac{d\phi}{dx} \right|_w + \dots \quad (2.45)$$

Expresando el resto de expansiones de forma similar y manipulando convenientemente las expresiones, se llega al siguiente modelo matemático:

$$\begin{cases} \phi_w = \frac{3\phi_w - \phi_{ww}}{2}, & \phi_e = \frac{3\phi_p - \phi_w}{2} & \text{si } F_w > 0, \quad F_e > 0 \\ \phi_w = \frac{3\phi_p - \phi_e}{2}, & \phi_e = \frac{3\phi_e - \phi_{ee}}{2} & \text{si } F_w < 0, \quad F_e < 0 \end{cases} \quad (2.46)$$

Así, para un flujo en la dirección positiva, se llega a la siguiente ecuación discretizada:

$$\left( -\frac{F_w}{2} \right) \phi_{ww} + \left( D_w + \frac{3F_w}{2} - \frac{F_e}{2} \right) \phi_w + \left( D_e - D_w + \frac{3F_e}{2} \right) \phi_p + (-D_e) \phi_e = \bar{S} \Delta V \quad (2.47)$$

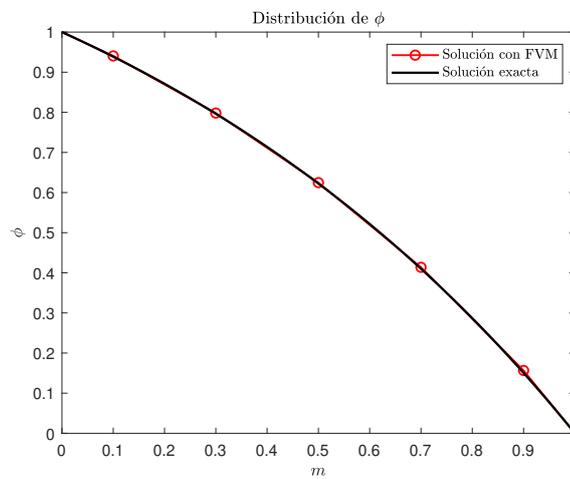
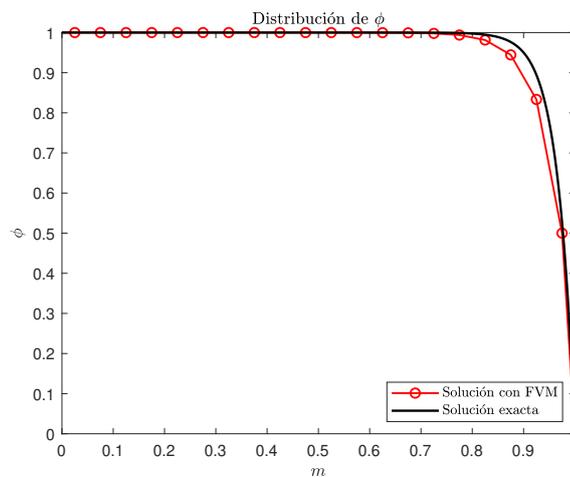
Si el problema en cuestión presentase un flujo en la dirección negativa, se llegaría a la siguiente ecuación modificada:

$$D_w \phi_w + \left( D_e - D_w + \frac{3F_w}{2} \right) \phi_p + \left( -D_e + \frac{3F_e}{2} - \frac{F_w}{2} \right) \phi_e - \left( \frac{F_e}{2} \right) \phi_{ee} = \bar{S} \Delta V \quad (2.48)$$

Procediendo de la misma manera que en el capítulo anterior, se aplicará este método al problema (2.32) y se compararán los resultados con los obtenidos mediante el *upwind* de primer orden. En primer lugar, para una convección pequeña se obtiene la representación de la Figura 2.18. Puede observarse cómo los resultados obtenidos prácticamente coinciden con la solución exacta. Incrementando la velocidad del flujo y el número de nodos, se llega a la situación de la Figura 2.19. Si se compara con los resultados proporcionados por el método *upwind* de primer orden, se aprecia un claro aumento de la precisión. Para cuantificar esta diferencia, véase la Tabla 2.3. En efecto, el error relativo cometido mediante la aplicación del SOU es significativamente menor.

**Tabla 2.3** Valores de  $\phi$  y comparación del error relativo cometido por el FVM usando el método FOU y SOU, para  $u = 3\text{ m/s}$  y empleando 7 nodos equiespaciados.

Nodo	1	2	3	4	5	6	7
Solución FOU ( $\phi$ )	1.0000	0.9999	0.9996	0.9979	0.9886	0.9398	0.6818
Solución SOU ( $\phi$ )	1.0000	1.0000	0.9999	0.9993	0.9952	0.9655	0.7539
Solución Exacta ( $\phi$ )	1.0000	1.0000	1.0000	1.0000	1.0000	0.9984	0.8827
Error Relativo FOU (%)	0.0010	0.0072	0.0403	0.2150	1.1362	5.8672	22.7556
Error Relativo SOU (%)	0.0001	0.0013	0.0095	0.0679	0.4820	3.2959	14.5857

**Figura 2.18** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 0.1\text{ m/s}$  empleando 5 nodos equiespaciados y usando el método *upwind* de segundo orden.**Figura 2.19** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 3\text{ m/s}$  empleando 20 nodos equiespaciados y usando el método *upwind* de segundo orden.

Se analizan ahora las propiedades descritas en la Sección 2.2.2 para comprobar si las expresiones sobre las que se sustenta este método dan lugar a un modelo consistente con las propiedades que deben poseer los esquemas de discretización:

**a) Conservatividad.** Este método emplea las mismas expresiones consistentes que los anteriores para calcular los flujos en las caras. Por tanto, la formulación es conservativa.

**b) Delimitación.** Si para ejemplificar se toma el caso  $F_w > 0$  y  $F_e > 0$ , se reordena la expresión (2.48) de la forma  $a_P \phi_P = a_W \phi_W + a_E \phi_E + a_{WW} \phi_{WW}$  y se tiene en cuenta que  $n_e = 1$  y  $n_w = -1$  en el caso unidimensional, el valor de los coeficientes es el siguiente:

$$a_{WW} = -\frac{F_w}{2} \quad a_W = D_w + \frac{3F_w}{2} + \frac{F_e}{2} \quad a_E = D_e \quad a_P = D_e + D_w + \frac{3F_e}{2} \quad (2.49)$$

Obsérvese que se verifica la relación  $a_P = a_{WW} + a_W + a_E + (F_e - F_w)$ , y como por continuidad  $F_e - F_w = 0$ , puede afirmarse que el criterio de Scarborough se cumple. Los coeficientes son siempre positivos a excepción de  $a_{WW}$ , lo cual se traduce en un método con una convergencia peor que en el caso anterior. Otra diferencia es que la matriz pasa a ser **pentadiagonal**, lo cual restringe la aplicabilidad de ciertos métodos algebraicos concebidos únicamente para matrices tridiagonales.

**c) Transportividad.** La formulación del método se fundamenta en la dirección de la corriente, definiendo la propiedad  $\phi$  en las caras como funciones a trozos que dependen directamente del sentido del flujo. Por tanto, la transportividad sí se satisface.

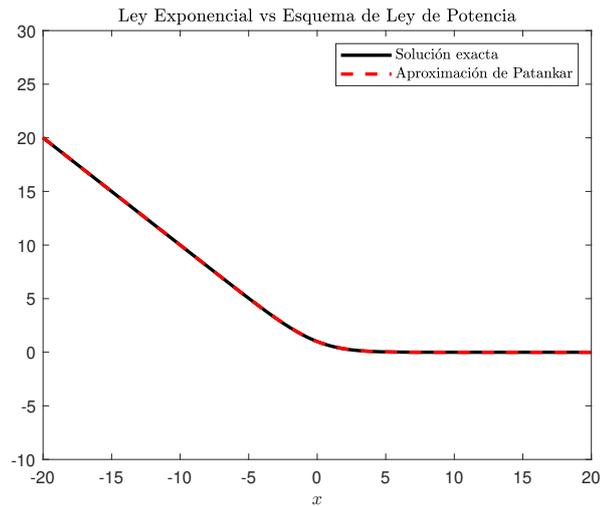
A modo de síntesis, el esquema *upwind* de segundo orden es un modelo más preciso que el *upwind* de primer orden, pero resulta más difícil alcanzar la convergencia y es computacionalmente más caro. Su implementación presenta una mayor complejidad frente a la enorme simplicidad del método de primer orden, por lo que ambos métodos se usan alternativamente según el problema en cuestión. En ocasiones se emplea el método de primer orden hasta que uno se asegura de que está convergiendo hacia la solución, y después se cambia al de segundo orden para aumentar la precisión.

### 2.2.5 Esquema de ley de potencia de Patankar

El esquema de ley de potencia (extendidamente conocido bajo su voz inglesa *Power-Law Scheme*) es un método que proporciona resultados muy precisos y cercanos a la solución analítica (véase [14]). La razón de esto es que las expresiones que caracterizan su formulación tienen su origen en el **esquema exponencial**, un método que se desarrolla directamente a partir de la solución exacta. Esta no es más que la solución presentada en (2.33). La aproximación del esquema de ley de potencia propuesto por el ingeniero Patankar en 1980 es extraordinariamente buena a un coste mucho menor que las leyes exponenciales. Tanto es así que las diferencias entre ambas pueden despreciarse, como se ilustra en la Figura 2.20, dando lugar a un método capaz de reproducir casi por completo la solución real del problema. Además elimina casi por completo los problemas de falsa difusión, convirtiéndose en la elección más favorable de entre todos los métodos numéricos desarrollados.

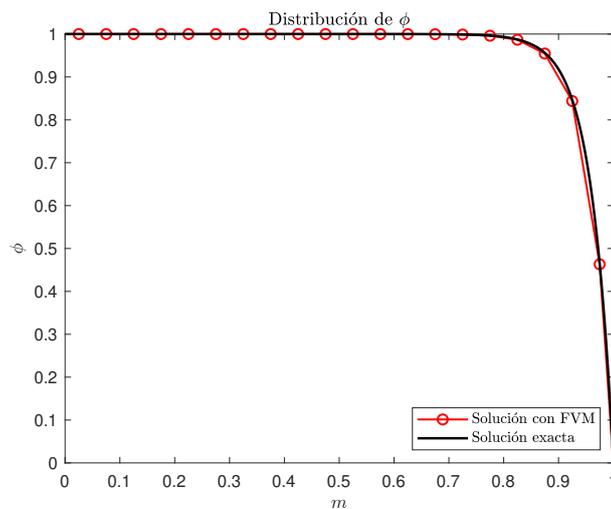
A continuación, se muestran los valores que dicta el esquema de ley de potencia de Patankar para la propiedad  $\phi$  en las caras:

$$\left\{ \begin{array}{ll} \phi_w = \phi_W - \frac{(1 - 0.1 Pe_w)^5}{Pe_w} (\phi_P - \phi_W) & \text{si } 0 < Pe_w < 10 \\ \phi_w = \phi_W & \text{si } Pe_w > 10 \\ \phi_e = \phi_P - \frac{(1 - 0.1 Pe_e)^5}{Pe_e} (\phi_E - \phi_P) & \text{si } 0 < Pe_e < 10 \\ \phi_e = \phi_P & \text{si } Pe_e > 10 \end{array} \right. \quad (2.50)$$



**Figura 2.20** Comparación entre la solución exacta y la aproximación polinómica propuesta por Patankar.

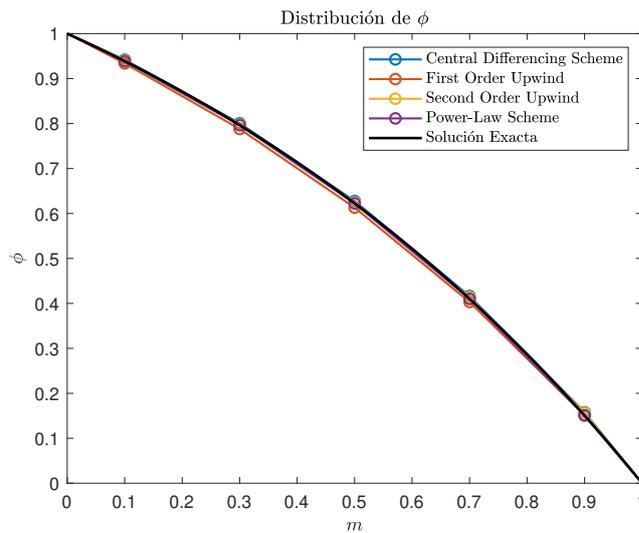
Si se aplica este método al problema (2.32) puede observarse que los errores obtenidos son con diferencia los más reducidos frente al resto de métodos. En efecto, obsérvese la siguiente figura:



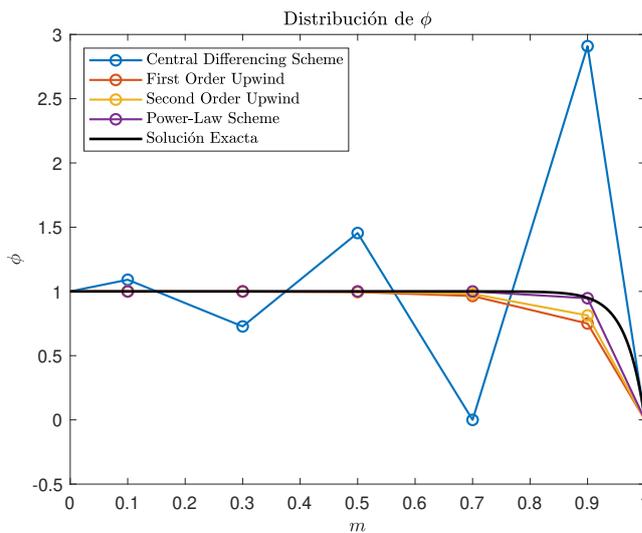
**Figura 2.21** Comparación entre la solución exacta y la obtenida mediante el FVM para  $u = 3 \text{ m/s}$  empleando 20 nodos equiespaciados y usando el esquema de ley de potencia de Patankar.

En conclusión, las propiedades del esquema de ley de potencia lo hacen sobresalir frente a los demás, pues pretende representar la solución exacta a un coste computacional reducido. Es una alternativa de extendido uso y códigos comerciales como Ansys Fluent lo implementan como opción de discretización para el usuario (véase [7]).

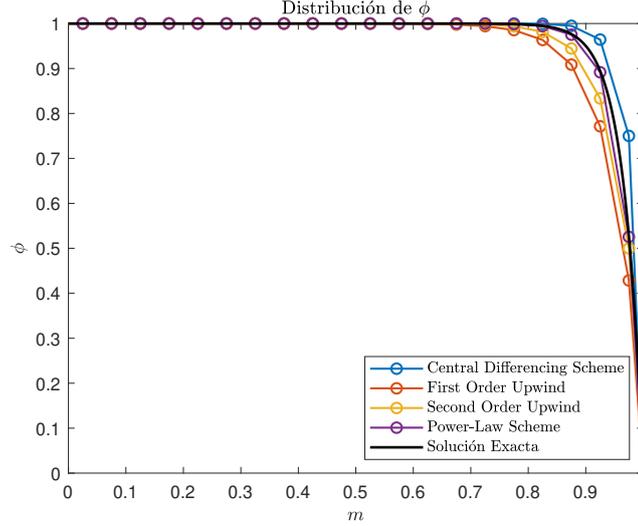
En último lugar, resulta bastante representativo graficar en una misma figura todos los métodos empleados para varios valores del número de nodos y número de Peclet. Los resultados son congruentes con lo que se ha ido comentando a lo largo del capítulo, quedando claramente representadas las limitaciones de cada método desarrollado y la mejora que progresivamente se ha ido consiguiendo con cada uno de ellos.



**Figura 2.22** Comparación entre la solución exacta y los distintos métodos de discretización para  $u = 0.1 \text{ m/s}$  empleando 5 nodos equiespaciados.



**Figura 2.23** Comparación entre la solución exacta y los distintos métodos de discretización para  $u = 3 \text{ m/s}$  empleando 5 nodos equiespaciados.



**Figura 2.24** Comparación entre la solución exacta y los distintos métodos de discretización para  $u = 3 \text{ m/s}$  empleando 20 nodos equiespaciados.

En el próximo capítulo se dará un paso más en el estudio de los problemas convectivo-difusivos relajando la hipótesis de estacionariedad. Se expondrán y analizarán varios métodos para tratar numéricamente el término de variación temporal de la ecuación general, así como ventajas y limitaciones de los mismos, y se aplicarán a un problema característico.

### 2.3 Problema difusivo, no estacionario y unidimensional

Una vez estudiados en detalle los diferentes métodos para resolver analíticamente problemas convectivo-difusivos en régimen estacionario, se está en condiciones de abordar el régimen transitorio o no estacionario. Para ello y siguiendo la exposición de [4], se recuperará el término de variación temporal de la ecuación general (1.5), que representa la tasa de cambio de la propiedad  $\phi$  con respecto del tiempo. Así, retener la variable temporal requiere una nueva integración respecto al tiempo en la ecuación (1.6), resultando la siguiente expresión general:

$$\int_{CV} \int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho \phi) dt dV + \int_t^{t+\Delta t} \int_A \vec{n} \cdot (\rho \phi \vec{v}) dA dt = \int_t^{t+\Delta t} \int_A \vec{n} \cdot (\Gamma \nabla \phi) dA dt + \int_t^{t+\Delta t} \int_{CV} S_\phi dV dt \quad (2.51)$$

La integración a lo largo del volumen de control es esencialmente idéntica a lo que se ha desarrollado en el caso estacionario, así como los métodos expuestos para el tratamiento de los términos convectivo-difusivos. Por ello, en este capítulo se centrarán los esfuerzos en la integración temporal a lo largo de un paso  $\Delta t$ . La forma más sencilla de abordar esta tarea es partiendo nuevamente de un problema paradigmático que permita focalizar la atención en lo que verdaderamente se está interesado. Es buen ejemplo la transferencia de calor no estacionaria en una barra unidimensional. Este es un problema puramente difusivo y está gobernado por la siguiente ecuación:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + S_\phi \quad (2.52)$$

Donde  $c$  es el calor específico del fluido y se mide en  $J/(KgK)$ . Integrando con respecto al tiempo y al volumen de control, se tiene:

$$\int_t^{t+\Delta t} \int_{CV} \rho c \frac{\partial T}{\partial t} dV dt = \int_t^{t+\Delta t} \int_{CV} \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) dV dt + \int_t^{t+\Delta t} \int_{VC} S dV dt \quad (2.53)$$

Si se considera la temperatura en los nodos como valor representativo del volumen de control, el primer miembro de la ecuación (2.53) puede aproximarse por:

$$\int_{CV} \left[ \int_t^{t+\Delta t} \rho c \frac{\partial T}{\partial t} dt \right] dV = \rho c (T_P - T_P^0) \Delta V \quad (2.54)$$

El valor  $T_P^0$  hace referencia a la temperatura en el instante  $t$ , mientras que  $T_P$  se refiere al valor de la temperatura en el instante  $t + \Delta t$ . En adelante, se mantendrá esta notación para distinguir los diferentes instantes temporales. Nótese que para llegar a este resultado se ha discretizado el término  $\partial T / \partial t$  usando la diferenciación central ( $\partial T / \partial t = (T_P - T_P^0) / \Delta t$ ). De igual forma, si se aplica el método de las diferencias centradas para modelar los términos difusivos tal y como se ha venido haciendo a lo largo de la memoria, se llega a la siguiente expresión:

$$\rho c (T_P - T_P^0) \Delta V = \int_t^{t+\Delta t} \left[ \left( k_e A_e n_e \frac{T_E - T_P}{x_E - x_P} \right) + \left( k_w A_w n_w \frac{T_P - T_W}{x_P - x_W} \right) \right] dt + \int_t^{t+\Delta t} \bar{S} \Delta V dt \quad (2.55)$$

Para evaluar las integrales del segundo miembro de la ecuación (2.55) debe modelarse la variación de la temperatura con respecto al tiempo. Puede hacerse la suposición de que el valor que prevalece es el que toma en el instante  $t$ , aunque también podría suponerse que el valor representativo es el del instante  $t + \Delta t$ . Para resolver esta problemática sin pérdida de generalidad, lo más útil es aproximar el valor de la integral mediante una **ponderación** de ambos valores, caracterizada por el parámetro  $\theta \in [0, 1]$ . Así:

$$I_T = \int_t^{t+\Delta t} T_P dt = [\theta T_P + (1 - \theta) T_P^0] \Delta t \quad (2.56)$$

Aplicando la discretización de (2.56) en (2.55) y teniendo en cuenta que  $D_e = k_e A_e n_e / (x_E - x_P)$  y  $D_w = k_w A_w n_w / (x_P - x_W)$ , se obtiene:

$$\begin{aligned} -\theta D_e T_E + [\theta (D_e - D_w) + \frac{\rho c \Delta V}{\Delta t} T_P + \theta D_w T_W] &= (1 - \theta) D_e T_E^0 + \\ \left[ (1 - \theta) (D_w - D_e) + \frac{\rho c \Delta V}{\Delta t} \right] T_P^0 - (1 - \theta) D_w T_W^0 &+ \bar{S} \Delta V \end{aligned} \quad (2.57)$$

Que puede escribirse en la forma común:

$$a_E T_E + a_P T_P + a_W T_W = a_E^0 T_E^0 + a_P^0 T_P^0 + a_W^0 T_W^0 + \bar{S} \Delta V = a_{SP} \quad (2.58)$$

La forma final de la ecuación discretizada dependerá del valor que se asigne a  $\theta$ . Para  $\theta = 0$ ,  $I_T = T_P^0 \Delta t$ , por lo que en la expresión únicamente se usarán las temperaturas en el instante anterior  $t$  para obtener el valor de  $T_P$ . El esquema de discretización resultante se denomina **explícito**. Cuando  $0 < \theta \leq 1$ , aparecen las temperaturas en el instante nuevo  $t + \Delta t$  en ambos miembros de la ecuación, y los esquemas resultantes se conocen como implícitos. El caso límite de  $\theta = 1$ , donde  $I_T = T_P \Delta t$ , se designa como esquema **completamente implícito**. El caso correspondiente a  $\theta = 1/2$  se conoce como esquema de **Crank-Nicolson**, y se verifica que  $I_T = 1/2 (T_P + T_P^0) \Delta t$ .

En los capítulos venideros se desarrollarán minuciosamente los esquemas explícito y completamente implícito y se ilustrará su aplicación tomando como ejemplo el problema descrito en la Figura 2.25 y gobernado por (2.52). El esquema de Crank-Nicolson queda fuera del alcance de este documento por ser su uso menos extendido que los casos límite de  $\theta = 0$  y  $\theta = 1$ .

### 2.3.1 Esquema explícito

Para analizar y valorar en profundidad las propiedades de los esquemas de discretización explícito y completamente implícito y poder establecer una comparación entre ambos, considérese el siguiente problema descrito en [4]. Sea una placa metálica y delgada de espesor  $L = 2\text{ cm}$  que se encuentra inicialmente a una temperatura constante de  $T = 200^\circ\text{C}$ . En un determinado instante ( $t = 0$ ), la temperatura de la cara este se reduce instantáneamente a  $T_B = 0^\circ\text{C}$ . La cara oeste permanece aislada térmicamente ( $\partial T/\partial x|_A = 0$ ). Además, la placa presenta una conductividad térmica  $k = 10\text{ W}/(\text{m K})$  y un factor  $\rho c = 10 \cdot 10^6\text{ J}/(\text{m}^3\text{ K})$ . Puede demostrarse que la solución analítica de este problema es la que se muestra en (2.59).

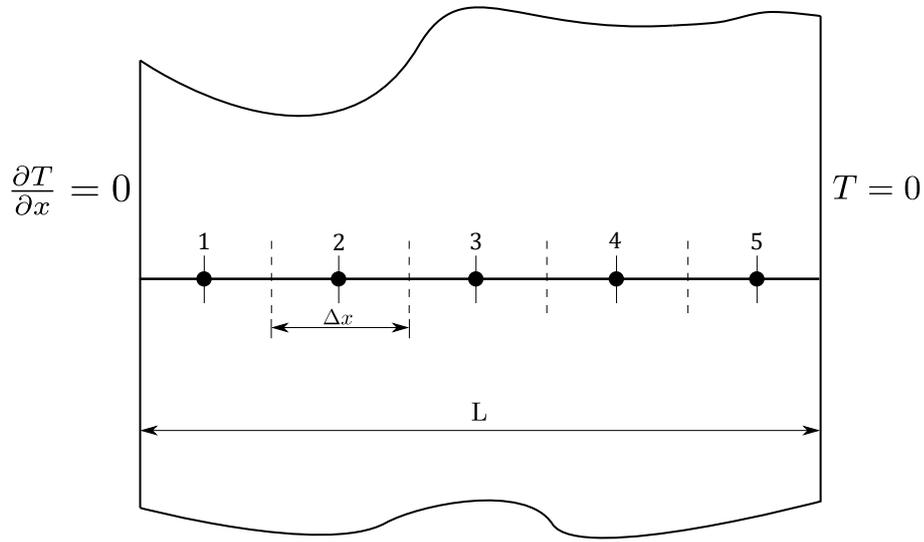


Figura 2.25 Geometría del problema que ejemplifica el caso no estacionario para  $t > 0$ .

$$\frac{T(x,t)}{T_0} = \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1} e^{-\alpha \lambda_n^2 t} \cos(\lambda_n x) \quad (2.59)$$

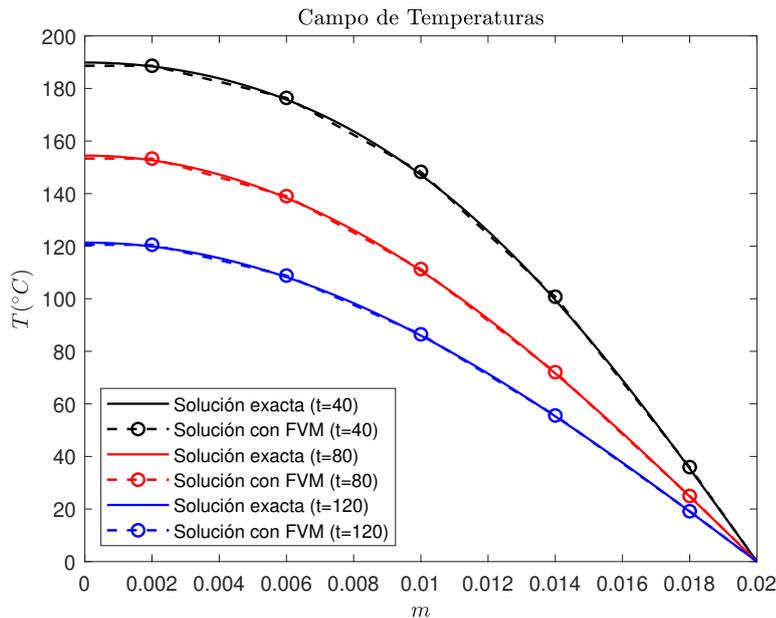
Donde  $\lambda_n = \frac{(2n-1)\pi}{2L}$  y  $\alpha = k/\rho c$ . Bajo estas condiciones, el esquema explícito lleva a la siguiente expresión:

$$\left( \frac{\rho c \Delta V}{\Delta t} \right) T_P = D_e T_E^0 + \left( D_w - D_e + \frac{\rho c \Delta V}{\Delta t} \right) T_P^0 - D_w T_W^0 \quad (2.60)$$

Tal y como se explicó con detalle en el Capítulo 2.2.2, para que la solución tenga una convergencia adecuada todos los coeficientes del sistema deben tener el mismo signo. El coeficiente  $a_P^0$  puede entenderse como un coeficiente del nodo vecino  $T_P^0$  (en este caso un vecino temporal), pues conecta los valores en  $P$  del instante  $t$  con los del instante  $t + \Delta t$ . En tal caso, reescribiendo (2.60) como  $a_P T_P = a_E T_E^0 + a_P T_P^0 + a_W T_W^0$ , considerando un malla uniforme ( $x_E - x_P = x_P - x_W = \Delta x$ ), área constante y teniendo en cuenta que  $n_e = 1$  y  $n_w = -1$  en el caso unidimensional, deberá cumplirse que:

$$a_p^0 > 0 \Rightarrow \frac{\rho c A \Delta x}{\Delta t} - \frac{kA}{\Delta x} - \frac{kA}{\Delta x} > 0 \Rightarrow \Delta t < \rho c \frac{\Delta x^2}{2k} \quad (2.61)$$

Esta condición establece una cota superior en el paso de integración y representa una **limitación** considerable para el esquema explícito. Un paso que no verifique la desigualdad puede llevar a soluciones completamente alejadas de la solución real, mientras que escoger un paso dentro del intervalo impuesto puede ser computacionalmente muy costoso. En particular, el paso se reduce como  $\Delta x^2$ , por lo que el refinamiento del mallado necesario puede llegar a ser inabordable. Es por todo esto que el método explícito no suele ser una buena recomendación para resolver problemas transitorios de carácter general. Todo esto se ilustra a continuación:

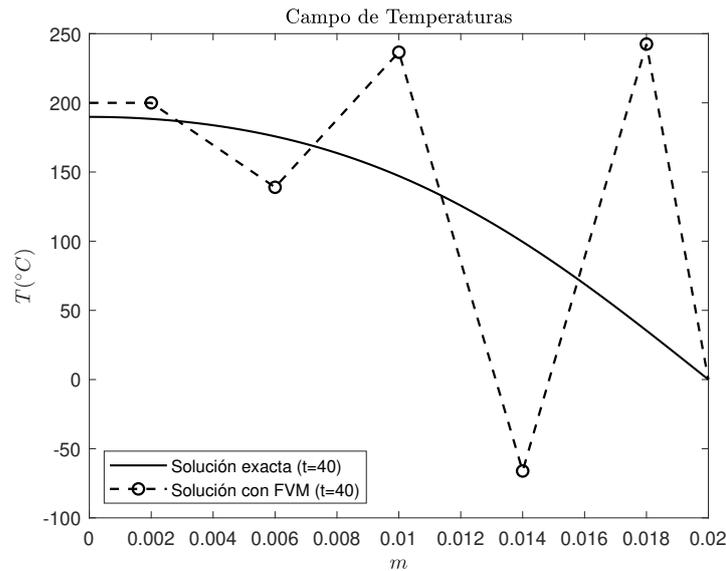


**Figura 2.26** Comparación entre la solución analítica y la solución mediante el FVM empleando 5 nodos, usando el esquema explícito para el término no estacionario y con un paso  $\Delta t = 2s$ .

**Tabla 2.4** Valores del campo de temperaturas y error relativo cometido por el FVM usando el esquema explícito, para  $\Delta t = 2s$  y empleando 5 nodos equiespaciados en  $t = 40s$ .

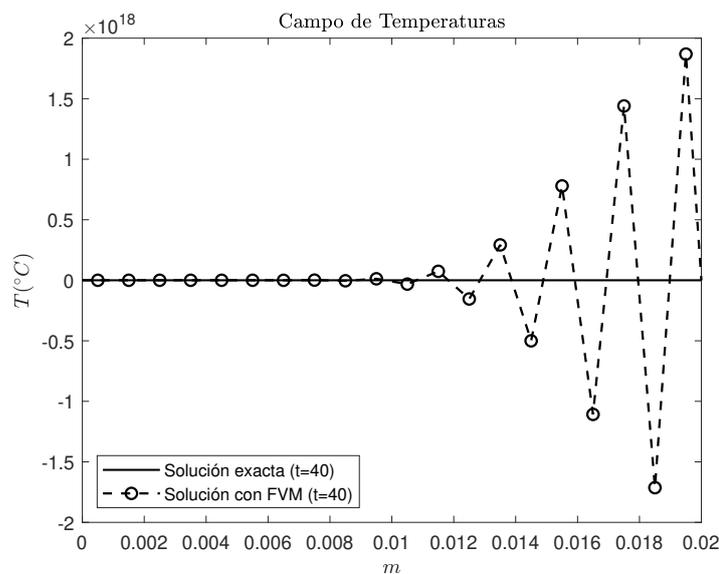
Nodo	1	2	3	4	5
<b>Solución FVM (°C)</b>	188.6386	176.4132	148.2926	100.7597	35.9418
<b>Solución Exacta (°C)</b>	188.3845	175.7649	147.1303	99.5043	35.3836
<b>Error Relativo (%)</b>	0.1349	0.3688	0.7900	1.2616	1.5777

Nótese cómo a medida que transcurre el tiempo la temperatura disminuye, como era de esperar. Para el caso particular del problema en cuestión y con una discretización de 5 nodos, se tiene que la condición a cumplir es  $\Delta t < 8s$ . Los resultados obtenidos en la Figura 2.26 son extraordinariamente precisos debido a que  $\Delta t = 2s$ . Escogiendo un paso fuera del intervalo:

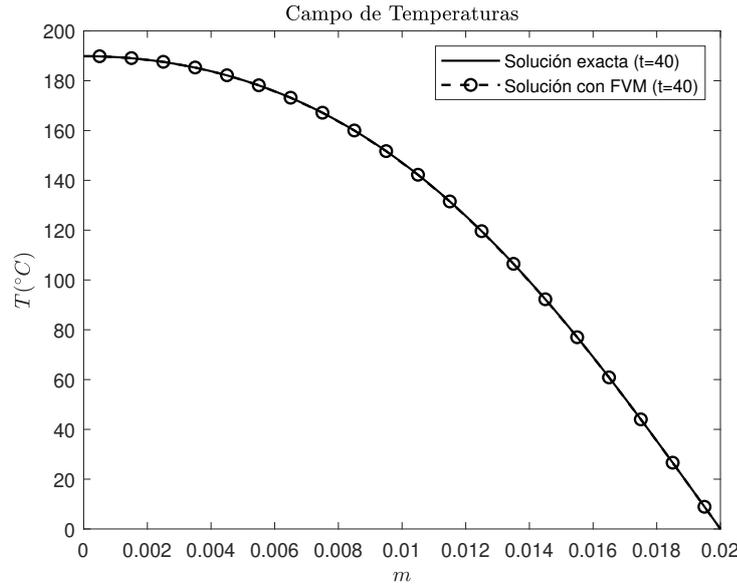


**Figura 2.27** Comparación entre la solución analítica y la solución mediante el FVM empleando 5 nodos equiespaciados, usando el esquema explícito para el término no estacionario y con un paso  $\Delta t = 10 s$  en  $t = 40 s$ .

En efecto, con un paso  $\Delta t = 10 s$  la solución oscila violentamente en torno a la solución analítica y claramente diverge de ésta. El problema de este método radica en que para mejorar la precisión de la solución es necesario aumentar el número de nodos, lo que disminuye el paso cuadráticamente. La exactitud de la Figura 2.26 se debe a la simplicidad del problema descrito, pero imagínese el escenario en que la solución mostrada en la Tabla 2.4 no es válida porque el máximo error relativo permitido sea del 0.1%. En tal caso, el único camino sería aumentar el número de nodos. Por ejemplo, usando 20 nodos:



**Figura 2.28** Comparación entre la solución analítica y la solución mediante el FVM empleando 20 nodos equiespaciados, usando el esquema explícito para el término no estacionario y con un paso  $\Delta t = 2 s$  en  $t = 40 s$ .



**Figura 2.29** Comparación entre la solución analítica y la solución mediante el FVM empleando 20 nodos equiespaciados, usando el esquema explícito para el término no estacionario y con un paso  $\Delta t = 0.25 s$  en  $t = 40 s$ .

La Figura 2.28 ilustra a la perfección la limitación del esquema explícito. Con 20 nodos y un mallado homogéneo debe cumplirse que  $\Delta t < 0.5 s$  y, efectivamente, con  $\Delta t = 2 s$  los resultados obtenidos son completamente desorbitados y no se acercan para nada a la solución. Bajando el paso a  $\Delta t = 0.25 s$ , la solución numérica se adapta perfectamente a la analítica, logrando un error relativo máximo menor al 0.1 % tal y como se muestra en la Figura 2.29.

### 2.3.2 Esquema completamente implícito

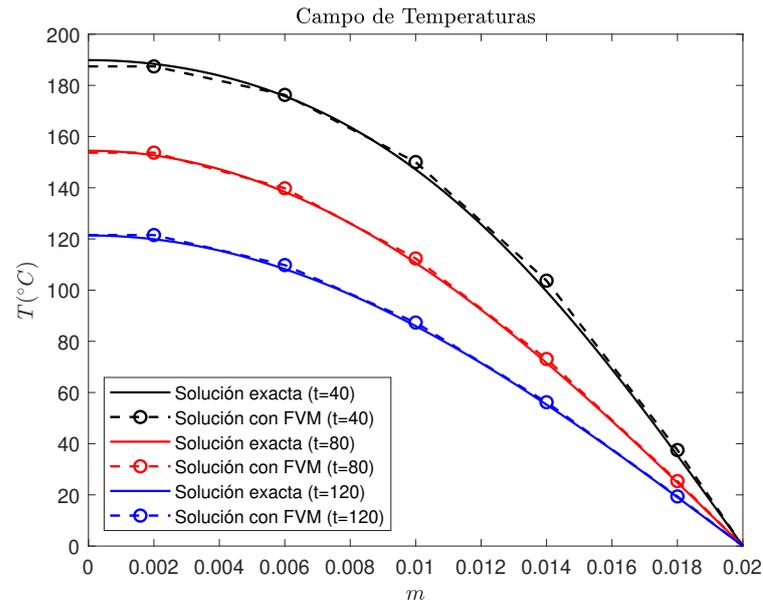
Para solucionar la problemática computacional que presenta el esquema explícito, se desarrollan a continuación las expresiones sobre las que se sustenta el esquema completamente implícito aplicado al problema descrito. Haciendo  $\theta = 1$  en la ecuación (2.57), se llega a la siguiente expresión:

$$-D_e T_E + \left( D_e - D_w + \frac{\rho c \Delta V}{\Delta t} \right) T_P + D_w T_W = \left( \frac{\rho c \Delta V}{\Delta t} \right) T_P^0 \quad (2.62)$$

Si ahora se reescribe la ecuación (2.62) como  $a_P T_P = a_E T_E + a_W T_W + a_P^0 T_P^0$ , considerando un mallado uniforme ( $x_E - x_P = x_P - x_W = \Delta x$ ), área constante y teniendo en cuenta que  $n_e = 1$  y  $n_w = -1$  en el caso unidimensional, se tiene:

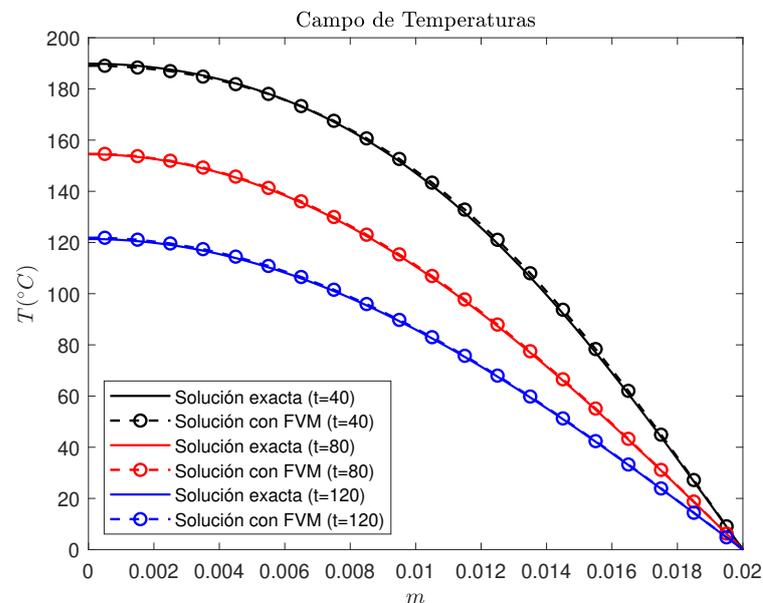
$$\left( D_e + D_w + \frac{\rho c \Delta V}{\Delta t} \right) T_P = D_w T_W + D_e T_E + \left( \frac{\rho c \Delta V}{\Delta t} \right) T_P^0 \quad (2.63)$$

Puede comprobarse que este esquema tiene la particularidad de que todos los coeficientes del sistema son positivos independientemente de su valor, lo que convierte a esta práctica en un método **incondicionalmente estable** para cualquier paso. Son su robustez y estabilidad los que hacen que este método sea el recomendado para cálculos transitorios de carácter general. A continuación se ilustra este esquema mediante su aplicación al problema descrito en (2.59).



**Figura 2.30** Comparación entre la solución analítica y la solución mediante el FVM empleando 5 nodos equiespaciados, usando el esquema implícito para el término no estacionario y con un paso  $\Delta t = 2 s$ .

Para el mismo paso y número de nodos, puede comprobarse como la solución, aunque ligeramente menos precisa que en el caso explícito, se aproxima con bastante exactitud a la solución analítica. Si se quiere mejorar la precisión puede aumentarse el número de nodos, esta vez sin necesidad de disminuir el paso.



**Figura 2.31** Comparación entre la solución analítica y la solución mediante el FVM empleando 20 nodos equiespaciados, usando el esquema implícito para el término no estacionario y con un paso  $\Delta t = 2 s$ .

Llegados a este punto se han desarrollado todos y cada uno de los esquemas de discretización que se emplearán a lo largo de la memoria para modelar los términos difusivos, convectivos y transitorios. Todos ellos, salvando las distancias con ciertas variaciones, son extrapolables al caso bidimensional. Su estudio será abordado en el próximo capítulo.



# 3 FVM para problemas convectivo-difusivos en 2D. Mallado estructurado

---

En los capítulos precedentes se han desarrollado los fundamentos del FVM y se han puesto en práctica algunos de los métodos existentes para resolver problemas convectivo-difusivos y no estacionarios de una sola dimensión. En este capítulo, se extenderán estos conceptos a problemas bidimensionales con **mallados estructurados**, en particular mallas ortogonales en un sistema de coordenadas cartesianas. Las mallas curvilíneas, aunque cuentan con una gran aplicación en la industria a la hora de modelar geometrías complejas, quedan fuera del alcance de este trabajo. La resolución de problemas 2D con mallas no estructuradas será tratada en profundidad en el próximo capítulo.

Al igual que se expuso en el caso 1D, se comenzará desarrollando un problema simplificado de difusión pura estacionaria. Posteriormente, se añadirán los términos advectivos y transitorios para finalizar el capítulo abordando un problema completo. Este desarrollo se hará en menor detalle, pues los métodos y esquemas de discretización que se implementan para tratar dichos términos son esencialmente los mismos que en el caso unidimensional, siendo no más que una mera extensión al problema multidimensional. Será clave la influencia de [4] y [6] a la hora de desarrollar este capítulo.

## 3.1 Problema difusivo, estacionario y bidimensional

La integración de la ecuación de transporte presentada en (1.5) en ausencia de convección para el caso estacionario y bidimensional queda:

$$\int_A \vec{n} \cdot \Gamma \left( \frac{\partial \phi}{\partial x} \vec{e}_x + \frac{\partial \phi}{\partial y} \vec{e}_y \right) dA + \int_{VC} S_\phi dV = 0 \quad (3.1)$$

La secuenciación a seguir para resolver el problema presenta los mismos pasos que en el caso unidimensional: mallado, discretización y resolución. A continuación se expondrán las diferencias que pueden encontrarse en cada uno con respecto al 1D. De igual forma, estos pasos serán equivalentes en cualquier problema de dos dimensiones.

### 3.1.1 Mallado del dominio bidimensional

De forma similar al caso 1D, el dominio computacional se divide en celdas o volúmenes de control. Éstas tienen ahora la particularidad de contar con cuatro caras denotadas por  $n$ ,  $s$ ,  $e$  y  $w$ , al igual que generalmente pueden definirse cuatro celdas colindantes o vecinas, denotadas por  $N$ ,  $S$ ,  $E$  y  $W$ . Análogamente al caso 1D, la información se almacena en el centroide de las celdas, mientras que para evaluar los flujos habrá que emplear los modelos ya conocidos que aproximan los valores en las caras. Para ilustrar un mallado estructurado ortogonal y cartesiano típico, obsérvese la Figura 3.1.

Resulta interesante destacar que la forma habitual de caracterizar los nodos es con una notación de doble índice  $(i, j)$ , donde  $i = 1, 2, \dots, M$  hace alusión al mallado a lo largo de la dirección del eje  $x$ , y  $j = 1, 2, \dots, N$  al mallado en el eje  $y$ , siendo  $M$  y  $N$  el número de nodos escogido en cada dirección, respectivamente. Sin embargo y como se verá más adelante, para obtener la solución del problema habrá que resolver nuevamente un conjunto de ecuaciones discretizadas, para lo cual se usará el comando  $\phi = A \setminus b$  de MATLAB. Esto quiere decir que  $\phi$  será un vector de longitud  $M \times N$  y no una matriz, por lo que para identificar correctamente los nodos habrá que pasar a notación de un solo índice  $(k)$ . Una vez resuelto el sistema, será necesario recuperar la notación de doble índice para realizar una correcta representación de los resultados. Esta transformación se hará según la siguiente expresión, que permite recorrer los nodos por filas:

$$k = (j - 1)M + i \quad (3.2)$$

Es invariante para el resultado final el sentido en el que se recorran los nodos. Si por ejemplo se quisiesen recorrer por columnas, habría que realizar el siguiente cambio:

$$k = (i - 1)N + j \quad (3.3)$$

De nuevo existen dos procedimientos para generar una malla de computación no homogénea, pero por simplicidad se supondrá un mallado homogéneo. En caso de necesitarse, se optará por aquél en que primero se ubican las caras de los volúmenes de control y posteriormente los nodos en los puntos medios entre las caras. La justificación de esto es nuevamente la presentada en la Sección 2.1.1.

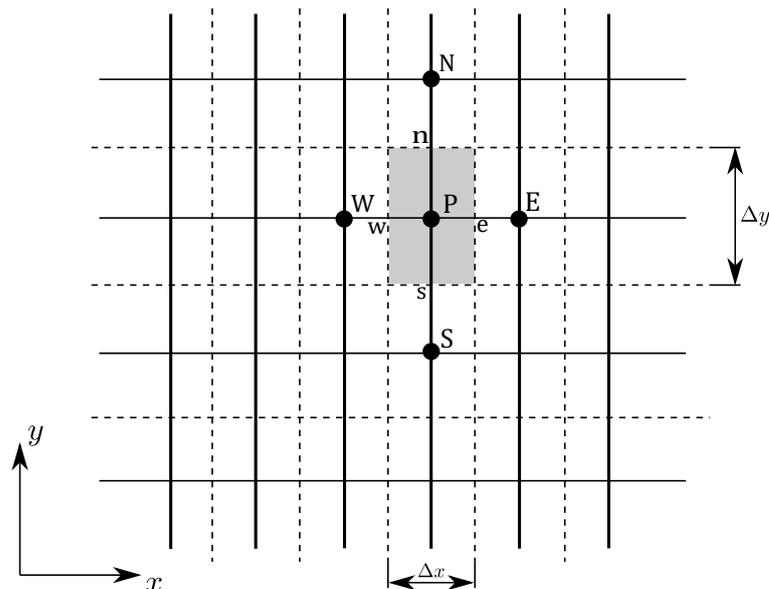


Figura 3.1 Mallado 2D cartesiano.

### 3.1.2 Discretización de las ecuaciones

El segundo paso a seguir consiste en desarrollar la ecuación de transporte integral (3.1) y particularizarla para cada volumen de control:

$$\int_A \Gamma(\nabla\phi) \cdot \vec{n} dA + \int_{\Delta V} S dV = \Gamma_e A_e n_e \left. \frac{\partial\phi}{\partial x} \right|_e + \Gamma_w A_w n_w \left. \frac{\partial\phi}{\partial x} \right|_w + \Gamma_n A_n n_n \left. \frac{\partial\phi}{\partial y} \right|_n + \Gamma_s A_s n_s \left. \frac{\partial\phi}{\partial y} \right|_s + \bar{S} \Delta V = 0 \quad (3.4)$$

Nuevamente se usará el método de las diferencias centradas para aproximar los gradientes de los términos difusivos. Esta aproximación, de segundo orden cuando la malla empleada sea uniforme, proporciona resultados válidos debido a que la difusión ocurre uniformemente en todas las direcciones.

$$\left. \frac{\partial\phi}{\partial x} \right|_w = \frac{\phi_P - \phi_W}{x_P - x_W} \quad \left. \frac{\partial\phi}{\partial x} \right|_e = \frac{\phi_E - \phi_P}{x_E - x_P} \quad \left. \frac{\partial\phi}{\partial y} \right|_s = \frac{\phi_P - \phi_S}{y_P - y_S} \quad \left. \frac{\partial\phi}{\partial y} \right|_n = \frac{\phi_N - \phi_P}{y_N - y_P} \quad (3.5)$$

Para los valores del coeficiente de difusión en las caras, puede usarse una interpolación lineal. Siempre que el mallado sea homogéneo, será la semisuma de los valores en los nodos colindantes:

$$\Gamma_w = \frac{\Gamma_P + \Gamma_W}{2} \quad \Gamma_e = \frac{\Gamma_P + \Gamma_E}{2} \quad \Gamma_s = \frac{\Gamma_P + \Gamma_S}{2} \quad \Gamma_n = \frac{\Gamma_P + \Gamma_N}{2} \quad (3.6)$$

Introduciendo en la ecuación (3.4) las expresiones derivadas para los gradientes, se obtiene:

$$-\Gamma_e A_e n_e \frac{\phi_E - \phi_P}{x_E - x_P} - \Gamma_w A_w n_w \frac{\phi_P - \phi_W}{x_P - x_W} - \Gamma_n A_n n_n \frac{\phi_N - \phi_P}{y_N - y_P} - \Gamma_s A_s n_s \frac{\phi_P - \phi_S}{y_P - y_S} = \bar{S} \Delta V \quad (3.7)$$

Definiendo  $D_e = \Gamma_e A_e n_e / (x_E - x_P)$ ,  $D_w = \Gamma_w A_w n_w / (x_P - x_W)$ ,  $D_n = \Gamma_n A_n n_n / (y_N - y_P)$  y  $D_s = \Gamma_s A_s n_s / (y_P - y_S)$  como términos asociados al flujo difusivo y reordenando se llega a:

$$[D_e + D_n - D_w - D_s] \phi_P - D_e \phi_E + D_w \phi_W - D_n \phi_N + D_s \phi_S = \bar{S} \Delta V \quad (3.8)$$

Que puede agruparse según los coeficientes  $a_i$  ya utilizados en el caso 1D:

$$a_P \phi_P + a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S = a_{SP} \quad (3.9)$$

Esta expresión primordial, formalmente idéntica en cada nodo, conformará cada una de las ecuaciones que componen el sistema a resolver. Esto se discutirá en la próxima sección.

### 3.1.3 Resolución del sistema

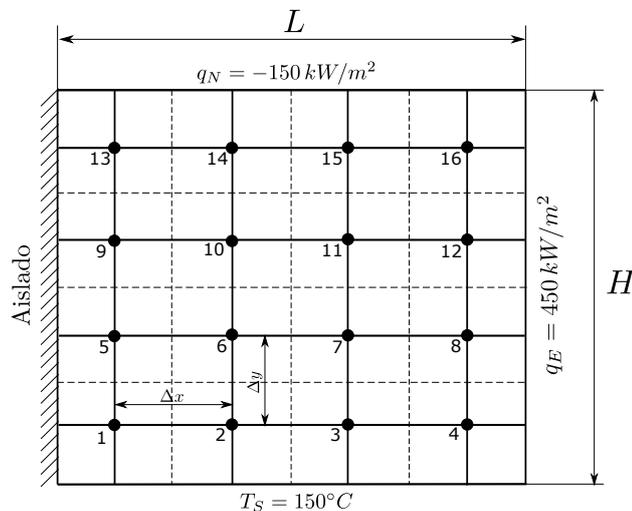
El desarrollo realizado para llegar a la ecuación (3.9) es válido para cada uno de los nodos internos que conforman el mallado. Sin embargo, el valor de los coeficientes variará cuando se trate de nodos pertenecientes a celdas fronterizas, pues deben incluirse las condiciones de contorno. Mientras en el caso unidimensional solo aparecían dos celdas limítrofes, el mallado 2D necesitará todo un *array* de celdas por cada cara de la malla donde aplicar las condiciones de contorno, siempre y cuando se considere el caso de malla ortogonal y cartesiana. Especial atención merecen las esquinas del dominio, donde deberán de verificarse dos condiciones simultáneamente.

Como se ha comentado en otras ocasiones, el sistema algébrico al que se llega será resuelto directamente con la función de MATLAB predeterminada (`\`). Para ello será necesario la transformación  $(i, j) \rightarrow k$ , de forma que la solución sea un vector cuyas componentes se refieran al valor de la propiedad  $\phi$  en cada nodo. Esta nueva nomenclatura supone que la matriz de coeficientes no sea pentadiagonal. Aunque los valores no nulos se agrupan en bandas, estas bandas estarán separadas entre sí. En particular, la diagonal principal, su primera subdiagonal y su primera superdiagonal serán no nulas (haciendo referencia a los nodos  $W$ ,  $P$  y  $E$ , respectivamente). Las otras dos bandas estarán separadas de la diagonal principal  $M$  diagonales, y harán referencia a los nodos  $S$  y  $N$ . Aunque las matrices sean dispersas, esta característica limita los métodos matemáticos que pueden utilizarse para resolver el sistema de forma eficiente, aunque dicho análisis queda fuera del alcance del proyecto.

### 3.1.4 Aplicación práctica 2D. Mallado estructurado

Como aplicación ilustrativa del FVM en dos dimensiones se considera como ejemplo paradigmático un problema sencillo que se describe a continuación. Se ha tomado como referencia uno de los problemas presentados en [4], al que se han cambiado las condiciones de contorno y los datos numéricos.

Considérese una placa bidimensional cuadrada de dimensiones  $L \times H$ , donde  $L = 0.5\text{ m}$  y  $H = 0.5\text{ m}$ , y con un espesor  $t = 0.01\text{ m}$ . El material de la placa es tal que su conductividad térmica es  $k = 1000\text{ W/mK}$ . La cara este de la placa está expuesta a un flujo de calor estacionario entrante de valor  $q_W = 450\text{ kW/m}^2$ . En la cara norte se tiene otro flujo de calor estacionario, esta vez saliente, de valor constante  $q_N = -150\text{ kW/m}^2$ . La cara sur está expuesta a una temperatura constante de  $T_S = 150^\circ\text{C}$ . Por último, la cara oeste se encuentra aislada térmicamente, por lo que el flujo de calor a través de ella es nulo. El problema descrito presenta el siguiente esquema:



**Figura 3.2** Representación del problema descrito.

Bajo estas condiciones, la ecuación de transferencia de calor que gobierna este problema estacionario bidimensional es:

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) = 0 \quad (3.10)$$

En primer lugar, se considerará un mallado uniforme que dividirá el dominio en dieciséis volúmenes de control. Esto equivale a decir que la placa se ha dividido en  $M = 4$  celdas en la dirección  $x$  y  $N = 4$  celdas en dirección  $y$ . Así, cada celda tendrá sus caras con una longitud constante de

$\Delta x = L/M = 0.125 \text{ m}$  y  $\Delta y = H/N = 0.125 \text{ m}$ . En cada volumen de control deberá satisfacerse la siguiente expresión:

$$k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e + k_w A_w n_w \left. \frac{\partial T}{\partial x} \right|_w + k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n + k_s A_s n_s \left. \frac{\partial T}{\partial y} \right|_s = 0 \quad (3.11)$$

Teniendo en consideración que para este problema en particular se verifica que  $k_w = k_e = k_n = k_s = k$ ,  $A_w = A_e = t \Delta y$ ,  $A_s = A_n = t \Delta x$ ,  $n_e = n_n = 1$  y  $n_w = n_s = -1$ , pueden definirse los siguientes términos difusivos:  $D_w = -kt \Delta y / \Delta x$ ,  $D_s = -kt \Delta x / \Delta y$ ,  $D_e = kt \Delta y / \Delta x$  y  $D_n = kt \Delta x / \Delta y$ . Así, se llega a una expresión similar a la presentada en (3.8):

$$[D_e + D_n - D_w - D_s] \phi_P - D_e \phi_E + D_w \phi_W - D_n \phi_N + D_s \phi_S = 0 \quad (3.12)$$

Esta ecuación discreta es válida únicamente para los nodos intermedios ( $k = 6, k = 7, k = 10, k = 11$ ). El resto de nodos están sujetos a condiciones de contorno de diferente naturaleza. A continuación, se detallarán las ecuaciones que deben satisfacerse en cada celda fronteriza. A modo ilustrativo y con la intención de generalizar la formulación, se añadirán también las expresiones a las que se llegaría en caso de tener la condición de contorno de temperatura impuesta en las caras en las que se impone el flujo, y la condición de contorno de flujo impuesto en las caras en las que se impone la temperatura. La generalización para la esquinas pasaría por combinar toda la casuística posible de las expresiones (3.13)-(3.20), la cual no se va a detallar.

**Cara oeste** ( $k = 1, k = 5, k = 9, k = 13$ )

La cara oeste se encuentra aislada térmicamente. Como consecuencia de esto, en la ecuación (3.11) habrá que imponer  $k_w A_w n_w \left. \frac{\partial T}{\partial x} \right|_w = 0$ . La ecuación resultante será:

$$[D_e + D_n - D_s] \phi_P - D_e \phi_E - D_n \phi_N + D_s \phi_S = 0 \quad (3.13)$$

Si por el contrario se impusiese una temperatura constante de valor  $T_W = \phi_{left}$ , se tendría que  $\left. \frac{\partial T}{\partial x} \right|_w = \frac{T_P - \phi_{left}}{\Delta x / 2}$  y se llegaría a la siguiente expresión:

$$[D_e + D_n - 2D_w - D_s] \phi_P - D_e \phi_E - D_n \phi_N + D_s \phi_S = -2D_w \phi_{left} \quad (3.14)$$

**Cara sur** ( $k = 2, k = 3$ )

En la cara sur existe una resistencia que impone una temperatura constante en el tiempo y de valor  $T_S = \phi_{down} = 150^\circ \text{C}$ . Así, en la ecuación (3.11) habrá que imponer  $\left. \frac{\partial T}{\partial y} \right|_s = \frac{T_P - \phi_{down}}{\Delta y / 2}$ . La ecuación resultante será:

$$[D_e + D_n - D_w - 2D_s] \phi_P - D_e \phi_E + D_w \phi_W - D_n \phi_N = -2D_s \phi_{down} \quad (3.15)$$

En el caso en que se impusiese el flujo  $q_S$  en esta cara, debe verificarse la condición  $k_s A_s n_s \left. \frac{\partial T}{\partial y} \right|_s = q_S A_s$ , lo cual lleva a la siguiente expresión:

$$[D_e + D_n - D_w] \phi_P - D_e \phi_E + D_w \phi_W - D_n \phi_N = q_S A_s \quad (3.16)$$

**Cara este** ( $k = 4, k = 8, k = 12, k = 16$ )

En la cara este se impone un flujo entrante de calor por unidad de superficie de valor  $q_E = 450 \text{ kW/m}^2$ . El área que atraviesa dicho flujo será  $A_e = t \Delta y$  para todos los volúmenes de control. Por lo tanto, en la ecuación (3.11) habrá que imponer  $k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e = q_E A_e$ . La ecuación resultante será:

$$[D_n - D_w - D_s] \phi_P + D_w \phi_W - D_n \phi_N + D_s \phi_S = q_E A_e \quad (3.17)$$

En el caso en que se impusiese una temperatura constante de valor  $T_E = \phi_{right}$ , se tendría que  $\left. \frac{\partial T}{\partial x} \right|_w = \frac{\phi_{right} - T_P}{\Delta x/2}$  y se llegaría a la siguiente expresión:

$$[2D_e + D_n - D_w - D_s] \phi_P + D_w \phi_W - D_n \phi_N + D_s \phi_S = 2D_e \phi_{right} \quad (3.18)$$

#### **Cara norte** ( $k = 14, k = 15$ )

En la cara norte se impone un flujo saliente de calor por unidad de superficie de valor  $q_N = -150 \text{ kW/m}^2$ . El área que atraviesa dicho flujo será  $A_n = t \Delta x$  para ambos volúmenes de control. Por lo tanto, en la ecuación (3.11) habrá que imponer  $k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n = q_N A_n$ . Así:

$$[D_e - D_w - D_s] \phi_P - D_e \phi_E + D_w \phi_W + D_s \phi_S = q_N A_n \quad (3.19)$$

Por último, en la situación en la que se impusiese una temperatura constante de valor  $T_N = \phi_{up}$ , se tendría que  $\left. \frac{\partial T}{\partial y} \right|_n = \frac{\phi_{up} - T_P}{\Delta y/2}$  y se llegaría a la siguiente expresión:

$$[D_e + 2D_n - D_w - D_s] \phi_P - D_e \phi_E + D_w \phi_W + D_s \phi_S = 2D_n \phi_{up} \quad (3.20)$$

Una vez caracterizadas todas las caras limítrofes, se pasa a modelar las celdas ubicadas en las esquinas del dominio computacional. En ellas deberán verificarse simultáneamente dos condiciones de contorno.

#### **Esquina sudoeste** ( $k = 1$ )

En la esquina sudoeste deben imponerse las condiciones  $\left. \frac{\partial T}{\partial y} \right|_s = \frac{T_P - T_S}{\Delta y/2}$  y  $k_w A_w n_w \left. \frac{\partial T}{\partial x} \right|_w = 0$ . La ecuación resultante será:

$$[D_e + D_n - 2D_s] \phi_P - D_e \phi_E - D_n \phi_N = -2D_s T_S \quad (3.21)$$

#### **Esquina sudeste** ( $k = 4$ )

En la esquina sudeste deben imponerse las condiciones  $\left. \frac{\partial T}{\partial y} \right|_s = \frac{T_P - T_S}{\Delta y/2}$  y  $k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e = q_E A_e$ . La ecuación resultante será:

$$[D_n - D_w - 2D_s] \phi_P + D_w \phi_W - D_n \phi_N = -2D_s T_S + q_E A_e \quad (3.22)$$

#### **Esquina noreste** ( $k = 16$ )

En la esquina noreste deben imponerse las condiciones  $k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n = q_N A_n$  y  $k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e = q_E A_e$ . La ecuación resultante será:

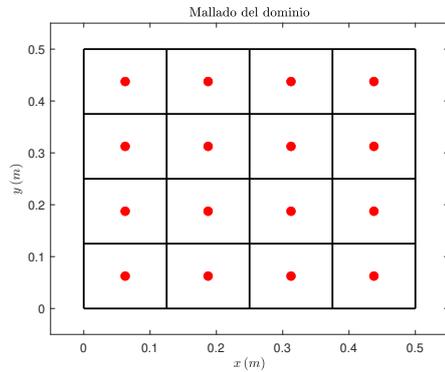
$$[-D_w - D_s] \phi_P + D_w \phi_W + D_s \phi_S = q_E A_e + q_N A_n \quad (3.23)$$

#### **Esquina noroeste** ( $k = 13$ )

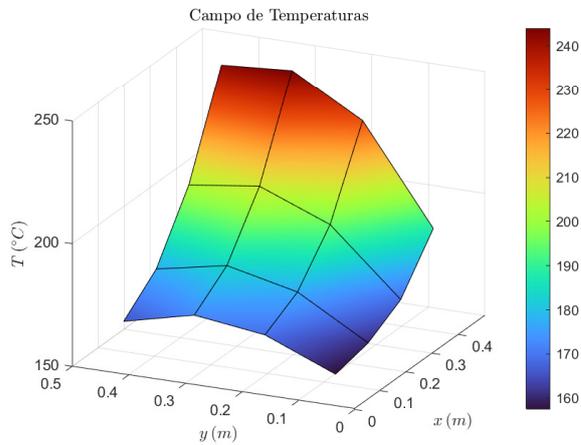
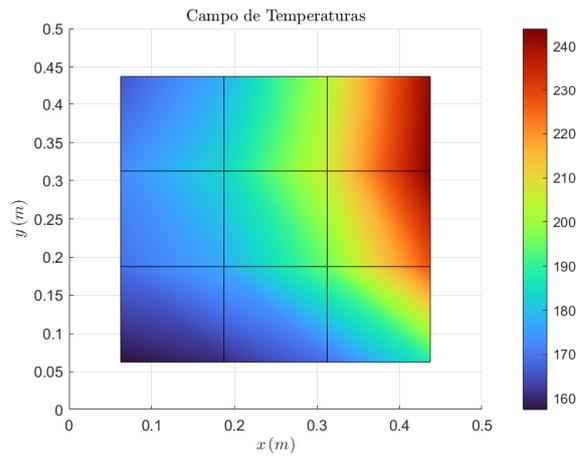
En la esquina noroeste deben imponerse las condiciones  $k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n = q_N A_n$  y  $k_w A_w n_w \left. \frac{\partial T}{\partial x} \right|_w = 0$ . La ecuación resultante será:

$$[D_e - D_s] \phi_P - D_e \phi_E + D_s \phi_S = q_N A_n \quad (3.24)$$

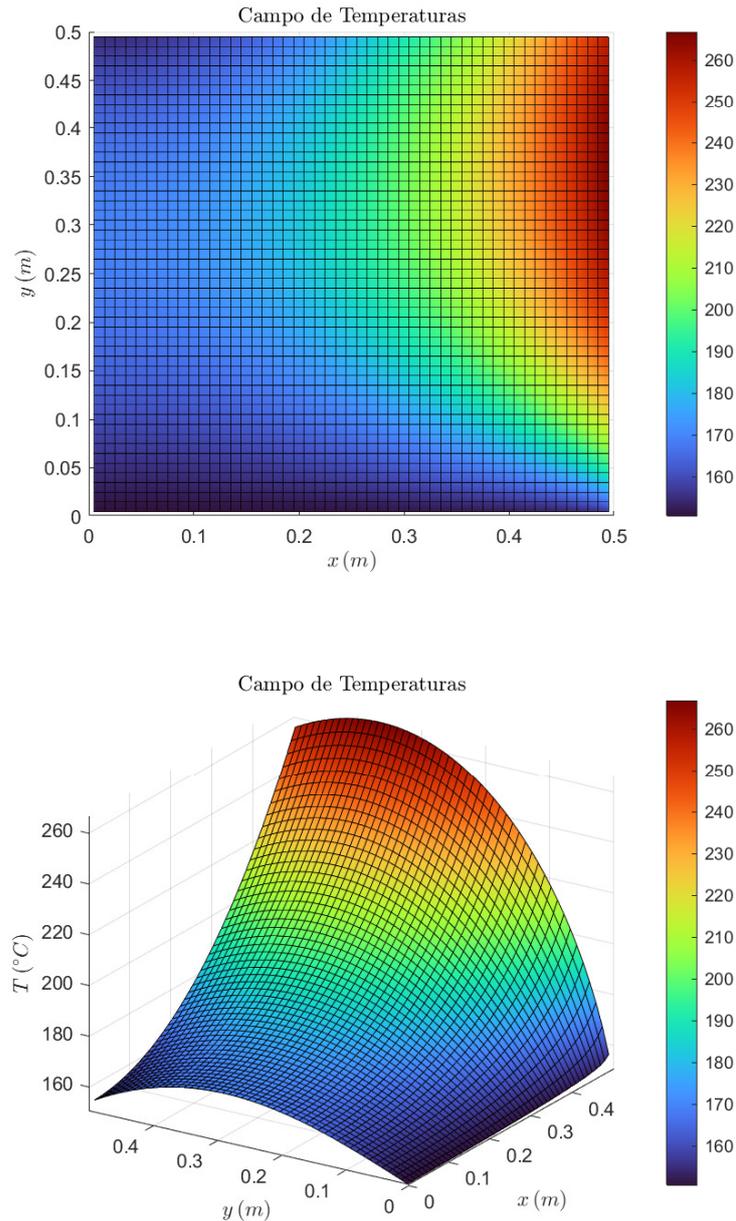
Una vez obtenido el conjunto de ecuaciones discretizadas que deben satisfacerse en cada uno de los nodos de la malla, se resuelve el sistema lineal mediante el comando `\` de MATLAB. A continuación, se presenta la solución calculada.



**Figura 3.3** Mallado del dominio de computación representado en MATLAB. Los puntos rojos son los nodos donde se calcularán los valores de temperatura.



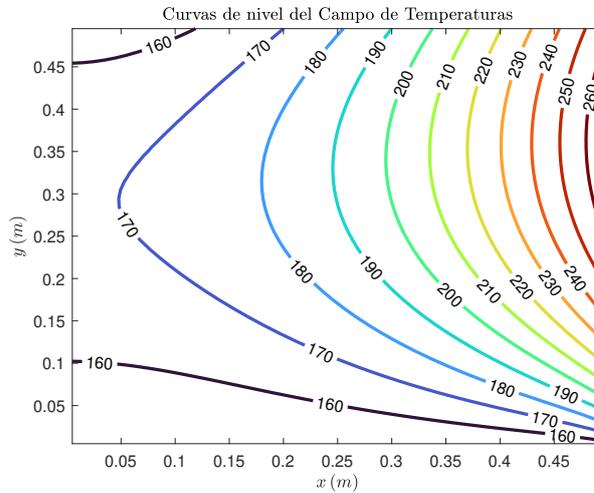
**Figura 3.4** Valores del campo de temperaturas en los nodos para un mallado uniforme  $M \times N$  con  $M = 4$  y  $N = 4$ .



**Figura 3.5** Valores del campo de temperaturas en los nodos para un mallado uniforme  $M \times N$  con  $M = 50$  y  $N = 50$ .

Son varios los comentarios que pueden hacerse acerca de las representaciones anteriores. En primer lugar, el problema que se ha resuelto mediante el FVM no cuenta con una solución analítica, por lo que los resultados no pueden ser contrastados contra una solución exacta. No obstante, éstos han sido contrastados con los que se obtienen en la literatura y han demostrado ser perfectamente válidos. En cualquier caso, para dar mayor credibilidad a los resultados, puede realizarse una valoración cualitativa de lo obtenido. La solución es congruente con lo que cabría esperar por varias razones: primeramente, cabía esperar que el valor máximo de temperatura se obtuviese cerca de la mitad de la cara este, pues existe un flujo de calor entrante por dicha cara, mientras que en las caras adyacentes se tiene un flujo de calor saliente (cara norte) y una temperatura impuesta (cara sur) menor a la temperatura que produce  $q_E$ . Por otro lado, la región más fría de la placa es la suroeste, pues la cara sur se mantiene a una temperatura constante de  $150^\circ\text{C}$  y la cara oeste está aislada. La

temperatura aumenta gradualmente a medida que se avanza hacia la derecha en la placa, como es lógico. Además, las caras aisladas térmicamente se caracterizan porque las curvas **isotermas** son **perpendiculares** a ellas. Esto se aprecia claramente en la siguiente figura.



**Figura 3.6** Curvas de nivel (isotermas) para un mallado uniforme  $M \times N$  con  $M = 50$  y  $N = 50$ .

En segundo lugar, resulta interesante comentar que con un mayor número de nodos se obtiene una representación mucho más realista debido a que la ubicación de los mismos se aproxima a los límites, logrando que la implementación de las condiciones de contorno sea más fidedigna.

En la siguiente sección, se resolverá y analizará un problema bidimensional convectivo-difusivo estacionario. Se recuperarán en las ecuaciones los términos de convección y se aplicará alguno de los modelos estudiados en el Capítulo 2 para su tratamiento.

### 3.2 Problema convectivo-difusivo, estacionario y bidimensional

Para continuar con el análisis de problemas convectivo-difusivos en 2D se procede a resolver un problema típico que incluye ambos fenómenos de transporte, similar a uno de los problemas que se encuentran en [6]. El tratamiento del término referente al flujo convectivo permanece inalterado en problemas multidimensionales, siempre y cuando se haga un uso correcto de las componentes de la velocidad, que ahora puede no estar alineada con los ejes coordenados (dando lugar al fenómeno de falsa difusión). Sirva de ejemplo el problema que se considera a continuación, una situación típica dentro del campo de la transferencia de calor. Para simplificar, el problema se resolverá haciendo uso únicamente del esquema *upwind* de primer orden para el término advectivo.

Considérese un conducto plano bidimensional por el que circula agua, de dimensiones  $L = 0.5\text{ m}$  y  $H = 0.1\text{ m}$ . El agua entra en el canal por la cara izquierda, que se encuentra a una temperatura constante de  $T_w = 30^\circ\text{C}$ . Una resistencia calienta el conducto por la cara superior con un flujo de calor por unidad de superficie de valor constante  $q_N = -k \frac{\partial T}{\partial y} \Big|_N = 15\text{ W/m}^2$ . Por otro lado, la cara inferior está completamente aislada ( $q_S = -k \frac{\partial T}{\partial y} \Big|_S = 0$ ). En la cara de salida se tiene un flujo de calor por unidad de superficie negativo (sale del dominio) de valor ( $q_E = -k \frac{\partial T}{\partial x} \Big|_E = -20\text{ W/m}^2$ ). Bajo estas condiciones, las propiedades termodinámicas del agua presentan los siguientes valores constantes:  $\rho = 1000\text{ kg/m}^3$ ,  $c_p = 4200\text{ J/kg K}$  y  $k = 0.613\text{ W/m K}$ . También se asumirá que el flujo

es laminar y está completamente desarrollado. En este escenario, las componentes de la velocidad presentan las siguientes expresiones:

$$u = u(y) = 6u_m \left[ \left( \frac{y}{H} \right) - \left( \frac{y}{H} \right)^2 \right] \quad v = 0 \quad (3.25)$$

Donde  $u_m$  es la velocidad media en dirección  $x$  a la entrada del canal. Su valor será tal que el  $Pe = \rho u_m H / \Gamma = 200$ , siendo  $\Gamma = k / \rho c_p$ . Así, se obtiene que  $u_m = 2.92 \cdot 10^{-7} \text{ m/s}$ . El objetivo es obtener la distribución de temperaturas dentro del conducto. Esta situación se describe en la Figura 3.7. La ecuación que gobierna este problema de convección-difusión estacionaria es la siguiente:

$$\frac{\partial}{\partial x} (\rho u c_p T) + \frac{\partial}{\partial y} (\rho v c_p T) = \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) \quad (3.26)$$

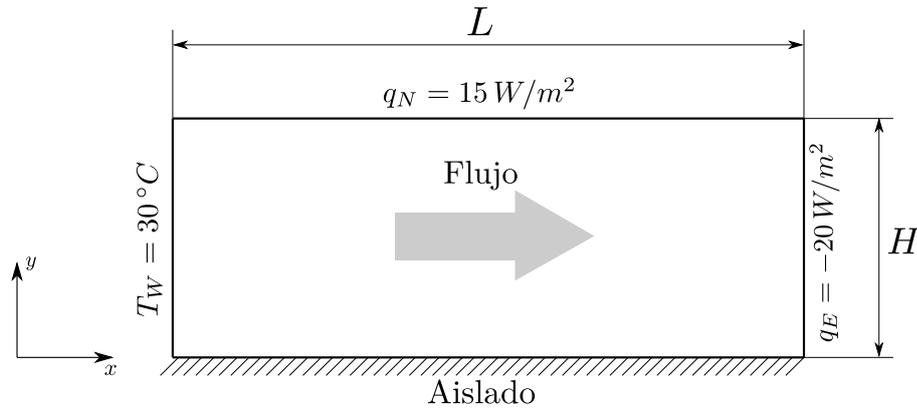


Figura 3.7 Representación del problema descrito.

Para una malla uniforme de tamaño genérico  $M \times N$  se tiene que en cada volumen de control debe satisfacerse la siguiente expresión:

$$\rho c_p u_e n_e A_e \phi_e + \rho c_p u_w n_w A_w \phi_w + \rho c_p v_n n_n A_n \phi_n + \rho c_p v_s n_s A_s \phi_s = k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e + k_w A_w n_w \left. \frac{\partial T}{\partial x} \right|_w + k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n + k_s A_s n_s \left. \frac{\partial T}{\partial y} \right|_s \quad (3.27)$$

Teniendo en cuenta que para este problema en particular se verifica que  $k_w = k_e = k_n = k_s = k$ ,  $A_w = A_e = \Delta y$ ,  $A_s = A_n = \Delta x$ ,  $n_e = n_n = 1$ ,  $n_w = n_s = -1$ ,  $u = u(y)$  y  $v = 0$ , pueden definirse los siguientes términos difusivos:  $D_w = -k \Delta y / \Delta x$ ,  $D_s = -k \Delta x / \Delta y$ ,  $D_e = k \Delta y / \Delta x$  y  $D_n = k \Delta x / \Delta y$ . Por otro lado, también pueden definirse los siguientes términos convectivos:  $F_w = -\rho c_p u(y) \Delta y$ ,  $F_e = \rho c_p u(y) \Delta y$ ,  $F_n = 0$  y  $F_s = 0$ . Así, usando el método de las diferencias centradas para aproximar los gradientes, se llega a la siguiente expresión:

$$[F_e + D_e + D_n - D_w - D_s] \phi_P - D_e \phi_E + (F_w + D_w) \phi_W - D_n \phi_N + D_s \phi_S = 0 \quad (3.28)$$

Esta ecuación discretizada servirá únicamente para los nodos intermedios ( $i = 2 : M - 1$ ,  $j = 2 : N - 1$ ). El resto de celdas estará sujeta a las condiciones de contorno que presenta el problema descrito. Se exponen a continuación de manera sucinta las expresiones que deben satisfacerse en cada volumen de control fronterizo y en las celdas ubicadas en las esquinas.

### Cara oeste

En la cara oeste se impone la condición de temperatura conocida  $T_W = 30^\circ\text{C}$ . Debe verificarse que  $\left. \frac{\partial T}{\partial x} \right|_w = \frac{T_P - T_W}{\Delta x/2}$ . La ecuación resultante será:

$$[F_e + D_e + D_n - 2D_w - D_s] \phi_P - D_e \phi_E - D_n \phi_N + D_s \phi_S = -[2D_w + F_w] T_W \quad (3.29)$$

### Cara norte

En la cara norte se impone la condición de flujo conocido, de valor  $q_N = 15 \text{ W/m}^2$ . El área que atraviesa dicho flujo es  $A_n = \Delta x$ . Debe verificarse que  $k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n = q_N A_n$ . La ecuación resultante será:

$$[F_e + D_e + D_n - D_w] \phi_P - D_e \phi_E + [F_w + D_w] \phi_W - D_n \phi_N = q_N A_n \quad (3.30)$$

### Cara este

En la cara este se impone un flujo saliente de valor  $q_E = -20 \text{ W/m}^2$ . El área que atraviesa dicho flujo es  $A_e = \Delta y$ . Como consecuencia habrá que imponer  $k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e = q_E A_e$ . La ecuación resultante será:

$$[F_e + D_n - D_w - D_s] \phi_P + [F_w + D_w] \phi_W - D_n \phi_N + D_s \phi_S = q_E A_e \quad (3.31)$$

### Cara sur

La cara sur se encuentra aislada térmicamente. Como consecuencia habrá que imponer  $k_s A_s n_s \left. \frac{\partial T}{\partial y} \right|_s = 0$ . La ecuación resultante será:

$$[F_e + D_e + D_n - D_w] \phi_P - D_e \phi_E + [F_w + D_w] \phi_W - D_n \phi_N = 0 \quad (3.32)$$

### Esquina sudoeste

En la esquina sudoeste deben imponerse las condiciones  $k_s A_s n_s \left. \frac{\partial T}{\partial y} \right|_s = 0$  y  $\left. \frac{\partial T}{\partial x} \right|_w = \frac{T_P - T_W}{\Delta x/2}$ . La ecuación resultante será:

$$[F_e + D_e + D_n - 2D_w] \phi_P - D_e \phi_E - D_n \phi_N = -[2D_w + F_w] T_W \quad (3.33)$$

### Esquina sudeste

En la esquina sudeste deben imponerse las condiciones  $k_s A_s n_s \left. \frac{\partial T}{\partial y} \right|_s = 0$  y  $k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e = q_E A_e$ . La ecuación resultante será:

$$[F_e + D_n - D_w] \phi_P + [F_w + D_w] \phi_W - D_n \phi_N = q_E A_e \quad (3.34)$$

### Esquina noreste

En la esquina noreste deben imponerse las condiciones  $k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n = q_N A_n$  y  $k_e A_e n_e \left. \frac{\partial T}{\partial x} \right|_e = q_E A_e$ . La ecuación resultante será:

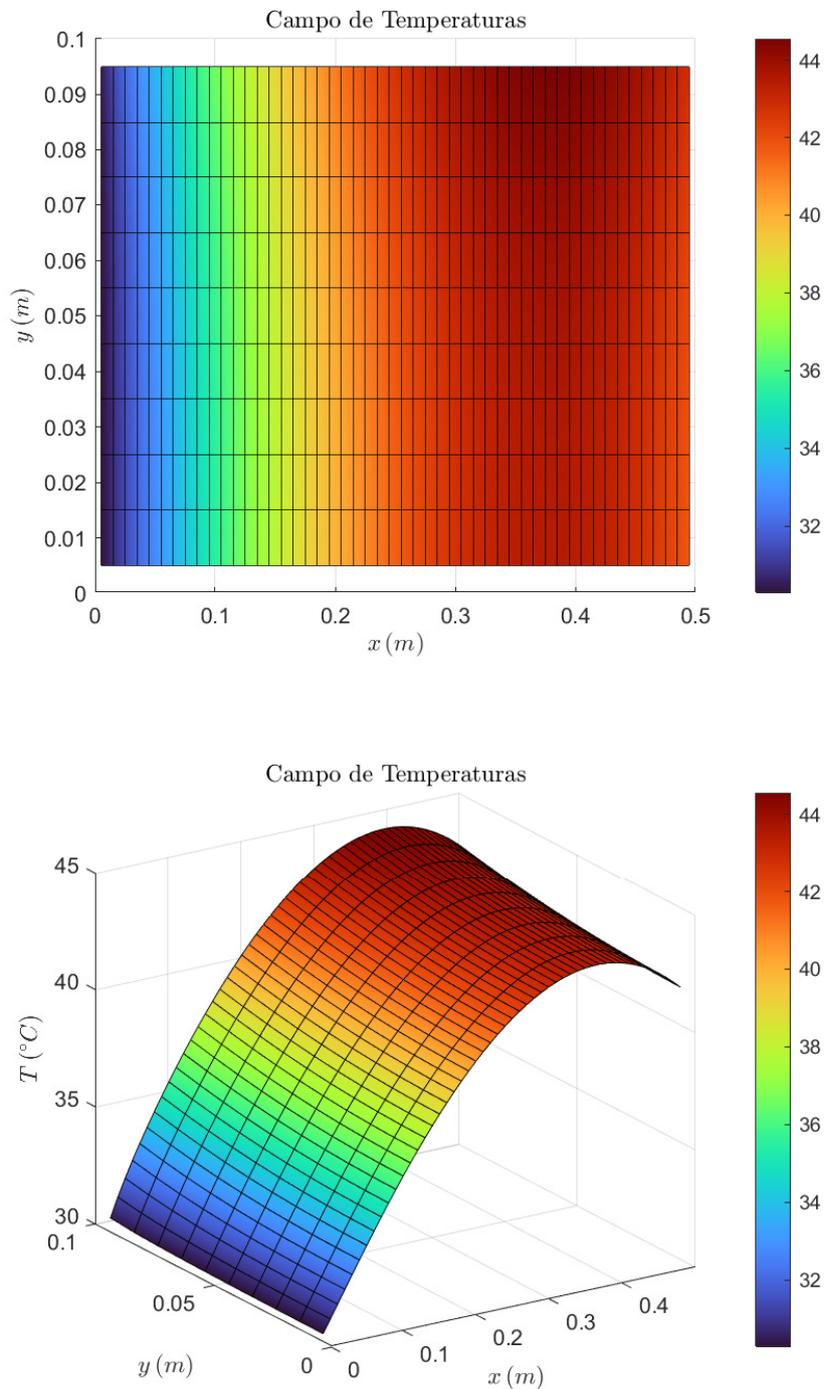
$$[F_e - D_w - D_s] \phi_P + [F_w + D_w] \phi_W + D_s \phi_S = q_N A_n + q_E A_e \quad (3.35)$$

### Esquina noroeste

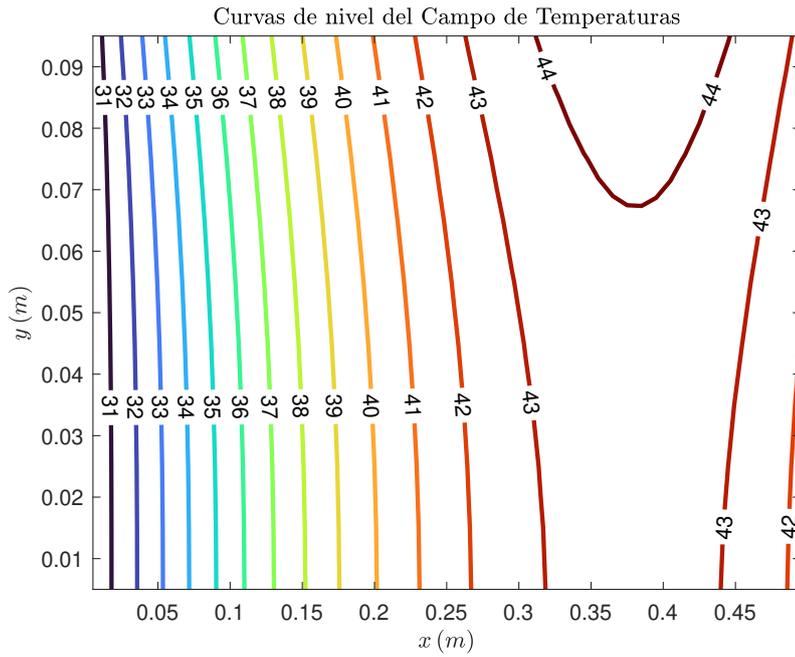
En la esquina noroeste deben imponerse las condiciones  $k_n A_n n_n \left. \frac{\partial T}{\partial y} \right|_n = q_N A_n$  y  $\left. \frac{\partial T}{\partial x} \right|_w = \frac{T_P - T_W}{\Delta x/2}$ . La ecuación resultante será:

$$[F_e + D_e - 2D_w - D_s] \phi_P - D_e \phi_E + D_s \phi_S = -[2D_w + F_w] T_W + q_N A_n \quad (3.36)$$

Una vez obtenido el conjunto de ecuaciones discretizadas que deben satisfacerse en cada uno de los nodos de la malla, se resuelve el sistema lineal mediante el comando `\` de MATLAB. A continuación, se presenta la solución calculada.



**Figura 3.8** Valores del campo de temperaturas en los nodos para un mallado  $M \times N$  con  $M = 50$  y  $N = 10$ .



**Figura 3.9** Curvas de nivel (isotermas) para un mallado  $M \times N$  con  $M = 50$  y  $N = 10$ .

Puede observarse cómo los resultados obtenidos son congruentes con lo esperado. En efecto, como el canal se calienta por la parte superior, la mayor temperatura se alcanza en dicha zona. La convección de izquierda a derecha se manifiesta con claridad en la curvatura que presentan las isotermas representadas en la Figura 3.9. Por último, la fronteras aislada térmicamente se identifica fácilmente al ser las curvas de nivel ortogonales a dicha pared.

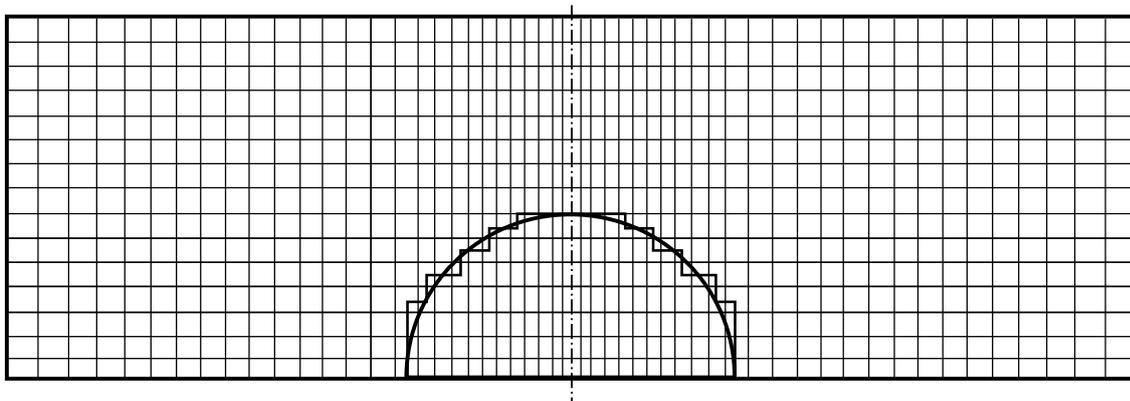
Una vez analizado el caso bidimensional con mallas estructuradas para problemas de convección-difusión térmica, se procede en el próximo capítulo a desarrollar el FVM para geometrías 2D con mallas no estructuradas. El estudio del término no estacionario se ha omitido para el caso bidimensional por ser su implementación y los modelos en los que se sustenta esencialmente idénticos a los presentados en la Sección 2.3.



## 4 FVM para problemas convectivo-difusivos en 2D. Mallado no estructurado

---

Hasta ahora las técnicas de resolución para las ecuaciones de flujo estaban basadas en esquemas de discretización sustentados en mallas estructuradas cartesianas. Este escenario es el más simple de todos, y ha sido particularmente útil para introducir y desarrollar los fundamentos del método de los volúmenes finitos de una forma fácil de entender. No obstante, la mayor parte de problemas que aparecen en ingeniería presentan complejas geometrías que difícilmente encajan con este tipo de mallas. Una posible forma de abordar situaciones en las que las fronteras del dominio fluido no coinciden con el sistema de coordenadas de la malla sería aproximar la geometría, por ejemplo usando una aproximación escalonada, tal y como se muestra en la siguiente figura extraída de [4].



**Figura 4.1** Malla ortogonal cartesiana para aproximar un problema de flujo en un semicilindro.

Sin embargo, una aproximación así es laboriosa e ineficiente. Las celdas que quedan fuera del dominio objeto de estudio contienen una información innecesaria, lo que se traduce en una pérdida de almacenamiento y un derroche de recursos, computacionalmente hablando. Además, la aproximación introduce errores al método de resolución que solo pueden ser salvados minimizando el tamaño de la malla, lo que implica un mayor refinamiento también en las zonas que no son de interés y por tanto una mayor ineficiencia. Esto pone de manifiesto la clara limitación que tienen este tipo de mallas y la necesidad de desarrollar nuevas estructuras que puedan lidiar con complejas geometrías de forma natural.

Así, aparecen las **estructuras de malla curvilíneas**, capaces de adaptarse a la geometría del cuerpo y mapear el dominio computacional en una forma relativamente sencilla de resolver (mediante cambios de variable). Esta nueva técnica resuelve geometrías con cierta complejidad, pero no es válida cuando la complejidad es considerable. Además, la formulación de las ecuaciones que gobiernan el problema es mucho más engorrosa. En estos casos, una **estructura de malla por bloques** resulta mucho más prometedora, pues establece sub-regiones que se mullan independientemente unas de otras según convenga. De esta forma, este tipo de estructura combina la sencillez en la generación, formulación y resolución que presentan las mallas cartesianas con la capacidad de adaptabilidad de las mallas por bloques.

Sería lógico pensar que para las geometrías más difíciles será necesario un número elevado de bloques. La extensión de esta idea lleva a las **mallas no estructuradas**, donde cada celda es propiamente un bloque. Esto da una flexibilidad ilimitada a la hora de adaptarse a la geometría del dominio y permite emplear los recursos de computación de manera óptima. La razón principal de ello es que no existe una malla de líneas coordenadas que implícitamente imponga una estructura determinada, por lo que las celdas pueden concentrarse libremente en las zonas donde sea necesario sin desperdiciar almacenamiento. Además, los volúmenes de control pueden adoptar cualquier forma, incluso combinarse (lo que se conoce como **mallado no estructurado híbrido**), y sin restricciones en el número de celdas colindantes.

La gran capacidad de adaptabilidad de las mallas no estructuradas a las geometrías más complejas no es el único de los grandes beneficios por los que esta configuración creció rápidamente en popularidad a comienzos de los años 90. Otro de los principales puntos a favor es que se sustenta en el uso de tecnología de generación automática de mallas no estructuradas, que por entonces ya estaba lo suficientemente madura gracias a que forma parte de las bases del FEM. No es de extrañar entonces que los expertos en este campo centrasen sus esfuerzos en adaptar esta topología al FVM para que las propiedades de conservatividad inherentes a este método se combinaran con la habilidad de tratar complejas geometrías para crear esta poderosa infraestructura de modelización. Es por todo ello que este método es reconocido como la técnica más utilizada a día de hoy en la industria del CFD.

A lo largo de este capítulo van a establecerse las bases del Método de los Volúmenes Finitos en mallas no estructuradas. Se caracterizarán exhaustivamente los tres pasos ya conocidos para obtener la solución del problema (mallado, discretización y resolución), donde cobrará vital importancia la gestión y almacenamiento de la información geométrica necesaria para resolver el problema. La obra de Mazumder ([6] y [17]-[20]) ha sido clave a la hora de exponer ordenadamente las ideas que subyacen al mallado no estructurado, resultando de gran ayuda en la elaboración de los códigos.

## 4.1 Problema difusivo, estacionario y unidimensional

Para sentar los pilares de este método de una manera progresiva, se comienza nuevamente con la situación más sencilla posible, a decir, un proceso de transporte puramente difusivo (en ausencia de convección y términos dependientes del tiempo). Servirá de ejemplo para los próximos capítulos la situación descrita en la Figura 3.2. Se sabe que bajo esas condiciones el problema está gobernado por la siguiente expresión:

$$\nabla \cdot (\Gamma \nabla \phi) + S_\phi = 0 \quad (4.1)$$

Como ya se ha venido viendo a lo largo del documento, el primer paso del FVM consiste en integrar la ecuación anterior a lo largo de un volumen de control genérico. La particularidad en este

caso es que el volumen de control puede tener una forma arbitraria cualquiera. No obstante, casi la totalidad de los generadores de malla 2D que se usan en la práctica emplean celdas poligonales (limitadas por un número finito de segmentos rectos) en lugar de curvas. De forma similar, los generadores de malla 3D reniegan de superficies curvas y mallan el dominio de computación con celdas poliédricas (delimitadas por superficies planas). La integración de la expresión (4.1) queda como sigue:

$$\int_A \vec{n} \cdot (\Gamma \nabla \phi) dA + \int_{VC} S_\phi dV = \int_A \vec{n} \cdot (\Gamma \nabla \phi) dA + \bar{S}_\phi \Delta V = 0 \quad (4.2)$$

Para llegar a (4.2) se ha aplicado el Teorema de la Divergencia de Gauss en el primer sumando y se ha tenido en cuenta que el término relativo a la fuente puede aproximarse por su valor medio localizado en el centroide del volumen de control, de forma idéntica al resto de problemas. Como el volumen de control está delimitado por segmentos rectos, la integral de área puede sustituirse por un sumatorio discreto de las integrales sobre cada cara. Esto es:

$$\sum_{f=1}^{N_{f,P}} \int_A \vec{n} \cdot (\Gamma \nabla \phi) dA + \bar{S}_\phi \Delta V = \sum_{f=1}^{N_{f,P}} \Gamma_f [(\nabla \phi)_f \cdot \vec{n}_f] A_f + \bar{S}_\phi \Delta V = 0 \quad (4.3)$$

Donde  $N_{f,P}$  es el número de caras de la celda  $P$  y  $\Gamma_f$ ,  $(\nabla \phi)_f$ ,  $\vec{n}_f$  y  $A_f$  hacen referencia a los valores de dichas magnitudes en cada una de las caras (el subíndice  $f$  denota *face*). De forma similar al término fuente, se asume que los valores medios en las caras coinciden exactamente con el valor en el centro de la cara. Puede comprobarse como, particularizando la expresión anterior para un mallado cartesiano, se llegaría a la ecuación obtenida en (3.4).

El siguiente paso a seguir en la formulación del FVM es caracterizar los valores de las variables de interés en el centroide de las celdas, que como ya se sabe son los nodos donde la solución será obtenida. Sin embargo, antes de acometer dicha tarea, es imperativo obtener y almacenar toda la información geométrica necesaria. Esta labor, que resultaba trivial en el caso de mallas estructuradas, requiere en el caso de mallas no estructuradas un pre-procesamiento mucho más extenso y dedicado.

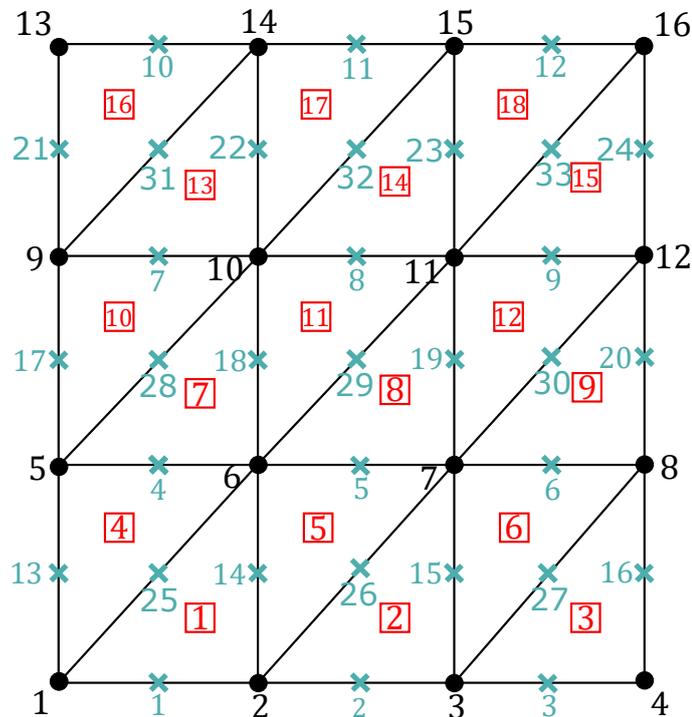
#### 4.1.1 Mallado del dominio. Procesamiento y almacenamiento de la información geométrica

La definición más sencilla y representativa de mallado no estructurado es aquél que no tiene un orden  $\{i, j, k\}$  predeterminado ni un patrón numérico definido. Como no está sustentado en esta ordenación, no se está obligado a definir los volúmenes de control de forma que las caras opuestas de dos celdas contiguas estén unidas. Esta es la razón por la que las celdas puedan tener una forma arbitraria (mezcla de triángulos, cuadriláteros, hexaedros y otros polígonos 2D).

Así, puede decirse que un mallado estructurado no es más que una particularización del mallado no estructurado en el que la numeración sigue un determinado orden. De esta forma, si un solucionador de PDE en CFD está escrito para una malla no estructurada, también funcionará automáticamente para una estructurada, pues la información geométrica estará escrita y procesada en formato no estructurado. Esta es una de las razones por las que este capítulo cobra especial importancia, ya que engloba y generaliza todo lo que se ha visto hasta ahora.

El primer paso para desarrollar un mallado no estructurado es el procesamiento de la geometría y la malla. Puesto que las celdas no están numeradas de forma ordenada, es necesario en primer lugar procesar y almacenar correctamente cómo están dispuestas espacialmente unas con respecto a otras. A este concepto se le conoce como **conectividad**, y es un trabajo previo vital para conocer cómo se comunican las celdas, caras y vértices entre sí. El número identificativo de cada nodo o celda, asignado por el algoritmo interno del generador de malla, se conoce como índice de celda

global. De forma similar, las caras y vértices también contarán con su propia asignación global que los identifique. Todo esto se ilustra en la siguiente figura, supuesto el dominio de computación del problema resuelto en la Sección 3.1.4.



**Figura 4.2** Indexación global en malla no estructurada.

Es importante notar que, aunque en el esquema de la Figura 4.2 parezca que existe cierto orden, a la hora de desarrollar el código hay que asumir que no se tiene dicho orden. Con todo esto, la primera información clave que es necesario calcular y almacenar es la siguiente:

- Número total de celdas:  $n_c$
- Número total de caras:  $n_f$
- Número total de vértices:  $n_v$
- Coordenadas de todos los nodos:  $(x_c, y_c)$
- Coordenadas de todos los centros de cara:  $(x_f, y_f)$
- Coordenadas de todos los vértices:  $(x_v, y_v)$

Esta información se concibe como parámetros del mallado a nivel global. Adicionalmente a la numeración global, toda celda necesitará también un sistema de numeración local (1, 2, 3, etc.). Esto servirá para identificar localmente a las caras y vértices que limitan cada celda. Por ello, también se necesita obtener la siguiente información:

- Número total de caras de cada celda:  $n_f(i_c)$
- Número total de vértices de cada celda:  $n_v(i_c)$
- Número total de vértices de cada cara:  $n_v(i_f)$

En el caso particular que se está estudiando, puesto que todas las celdas son triángulos, se tiene que  $n_f(i_c) = 3 (\forall i_c)$ . Además, por la composición del mallado en cuestión, se verifica también que  $n_v(i_c) = 3 (\forall i_c)$  y  $n_v(i_f) = 2 (\forall i_f)$ . A continuación se muestran los códigos de MATLAB que generan esta información. Puede observarse cómo el cálculo de las coordenadas de los centros de las caras no se ha incluido. El motivo de esto es que dicho cálculo requiere un procesamiento previo de la conectividad entre caras y vértices, tal y como se comprobará posteriormente.

```

1 %% MALLADO
2
3 M = 3;
4 N = 3;
5 L = 1;
6 H = 1;
7 dx = L/M;
8 dy = H/N;
9
10 % Número Total de Nodos
11
12 ic = 0;
13 for j = 1:N
14     for i = 1:M
15         ic = ic + 1;
16     end
17     for i = 1:M
18         ic = ic + 1;
19     end
20 end
21 n_c = ic;
22
23 % Número Total de Vértices
24
25 iv = 0;
26 for j = 1:N+1
27     for i = 1:M+1
28         iv = iv + 1;
29     end
30 end
31 n_v = iv;
32
33 % Coordenadas de Vértices
34
35 iv = 0;
36 xv = zeros(1,n_v);
37 yv = zeros(1,n_v);
38 for j = 1:N+1
39     for i = 1:M+1
40         iv = iv + 1;
41         xv(iv) = (i-1)*dx;
42         yv(iv) = (j-1)*dy;
43     end
44 end
45
46 % Coordenadas de Nodos
47
48 ic = 0;
49 xc = zeros(1,n_c);
50 yc = zeros(1,n_c);
51 for j = 1:N
52     for i = 1:M
53         ic = ic + 1;

```

```

54     xc(ic) = (xv(i) + xv(i+1) + xv(i+1+M+1))/3;
55     yc(ic) = (j-1)*H/(N) + (yv(i) + yv(i+1) + yv(i+1+M+1)) / 3;
56     end
57     for i = 1:M
58         ic = ic + 1;
59         xc(ic) = (xv(i) + xv(i+M+1) + xv(i+1+M+1))/3;
60         yc(ic) = (j-1)*H/(N) + (yv(i) + yv(i+M+1) + yv(i+1+M+1)) / 3;
61     end
62 end

```

Téngase en cuenta que para ubicar los nodos en cada celda triangular se ha elegido su **baricentro**, cuya posición puede obtenerse como la media aritmética de las coordenadas de los tres vértices. Para terminar de dar sentido a este mallado no estructurado, puesto que no sigue ningún patrón (véase por ejemplo la celda 1, rodeada de las caras 1, 14 y 25), es necesario calcular y almacenar las matrices de conectividad. Esto incluye:

- Conectividad *Celda-A-Cara*:  $Lcf(i_c, k)$
- Conectividad *Cara-A-Celda*:  $Lfc(i_f, k)$
- Conectividad *Celda-A-Vértice*:  $Lcv(i_c, k)$
- Conectividad *Cara-A-Vértice*:  $Lfv(i_f, k)$

Donde  $k$  hace referencia al índice local y podrá tomar los valores  $k = 1, 2$  si se habla de vértices asociados a una cara ó  $k = 1, 2, 3$  si se habla de caras o vértices asociados a una celda. Esta información es proporcionada por el generador de malla. En los softwares CFD esta tarea compete al pre-procesador y se hace de forma automática. No obstante, debido a que los códigos de este proyecto se están elaborando en MATLAB, la generación de esta información deberá ser proporcionada por el usuario, simulando lo que un software convencional haría automáticamente. A continuación se exponen los códigos que se han desarrollado para el cálculo de dichas matrices de conectividad.

```

1 %% CONECTIVIDAD CELDA-A-CARA
2
3 ic = 0; % Indice para recorrer las celdas
4 Lcf = zeros(n_c,3); % Matriz de Conectividad Celda-A-Cara
5 for j = 1:N
6     for i = 1:M
7         ic = ic + 1;
8         Lcf(ic,1) = ic - (j-1)*M;
9         Lcf(ic,2) = ic + M*(N+1) + 1 - (j-1)*(M-1);
10        Lcf(ic,3) = ic + M*(N+1) + N*(M+1) - (j-1)*M;
11    end
12    for i = 1:M
13        ic = ic + 1;
14        Lcf(ic,1) = ic - M + M*(N+1) + N*(M+1) - (j-1)*M;
15        Lcf(ic,2) = ic - (j-1)*M;
16        Lcf(ic,3) = ic - M + M*(N+1) - (j-1)*(M-1);
17    end
18 end
19
20 n_f = Lcf(n_c,1); % Número Total de Caras

```

```

1 %% CONECTIVIDAD CARA-A-CELDA
2
3 Lfc = zeros(n_f,2); % Matriz de Conectividad Cara-A-Celda
4 for iface = 1:n_f
5     c1 = 0; c2 = 0;
6     for i = 1:n_c

```

```

7     for k = 1:3
8         if Lcf(i,k) == iface
9             if c1 == 0
10                c1 = i;
11            else
12                c2 = i;
13            end
14        end
15    end
16 end
17 Lfc(iface,1) = c1;
18 Lfc(iface,2) = c2;
19 if Lfc(iface,1) == 0
20     Lfc(iface,1) = Lfc(iface,2);
21 end
22 if Lfc(iface,2) == 0
23     Lfc(iface,2) = Lfc(iface,1);
24 end
25 end

```

```

1 %% CONECTIVIDAD CELDA-A-VERTICE
2
3 ic = 0; % Indice para recorrer las celdas
4 Lcv = zeros(n_c,3); % Matriz de Conectividad Celda-A-Vértice
5 for j = 1:N
6     for i = 1:M
7         ic = ic + 1;
8         Lcv(ic,1) = ic - (j-1)*(M-1);
9         Lcv(ic,2) = ic + 1 - (j-1)*(M-1);
10        Lcv(ic,3) = ic + 1 + (M+1) - (j-1)*(M-1);
11    end
12    for i = 1:M
13        ic = ic + 1;
14        Lcv(ic,1) = ic - M - (j-1)*(M-1);
15        Lcv(ic,2) = ic + (M+1) - j*(M-1);
16        Lcv(ic,3) = ic - 1 + (M+1) - j*(M-1);
17    end
18 end

```

```

1 %% CONECTIVIDAD CARA-A-VERTICE
2
3 Lfv = zeros(n_f,2); % Matriz de Conectividad Celda-A-Vértice
4 for ic = 1:n_c
5     iface1 = Lcf(ic,1);
6     Lfv(iface1,1) = Lcv(ic,1);
7     Lfv(iface1,2) = Lcv(ic,2);
8     ifc2 = Lcf(ic,2);
9     Lfv(iface2,1) = Lcv(ic,2);
10    Lfv(iface2,2) = Lcv(ic,3);
11    ifc3 = Lcf(ic,3);
12    Lfv(iface3,1) = Lcv(ic,3);
13    Lfv(iface3,2) = Lcv(ic,1);
14
15 end
16
17 % Coordenadas de Centros de las Caras
18
19 xf = zeros(n_f,1);
20 yf = zeros(n_f,1);
21 for iface = 1:n_f

```

```

22 v1 = Lfv(iface,1);
23 v2 = Lfv(iface,2);
24 xf(iface) = 0.5*(xv(v1) + xv(v2));
25 yf(iface) = 0.5*(yv(v1) + yv(v2));
26 end

```

Son varios los comentarios que pueden hacerse en relación con el almacenamiento de la conectividad que se ha expuesto en las líneas de código anteriores. En primer lugar, uno podría preguntarse sobre la necesidad de este procesamiento. La respuesta a dicha pregunta se ejemplificará más adelante, pero fundamentalmente la exigencia de esto radica en la propia naturaleza del método. En los cálculos posteriores se requerirá la evaluación de ciertas magnitudes en nodos, caras y vértices de forma continuada, y la única forma eficiente de realizar esta tarea es disponer de un almacenamiento compacto de números enteros (ocupan poco espacio en la memoria) que sea fácilmente accesible. En segundo lugar, si uno computa los códigos adjuntos comprobará que la asignación local de caras y vértices se ha realizado siguiendo siempre un sentido antihorario. Esto se hace con la intención de generar matrices claras y fácilmente depurables en caso de encontrarse fallos. Por último, en la matriz de conectividad *Cara-A-Celda*, aunque la asignación  $k = 1$  ó  $k = 2$  se ha hecho de forma completamente arbitraria, se ha establecido por norma que aquellas caras limítrofes (que solo tienen una celda colindante) presenten el mismo índice global para cada índice local; es decir, si  $f_b$  es una cara frontera, entonces  $Lfc(f_b,1) = Lfc(f_b,2)$ . El interés de esto se justificará posteriormente.

A modo ilustrativo y tomando como referencia la malla representada en la Figura 4.2, puede decirse que  $Lcf(9,1) = 6$ ,  $Lcf(9,2) = 20$  y  $Lcf(9,3) = 30$  para ejemplificar la conectividad *Celda-A-Cara*,  $Lfc(32,1) = 14$  y  $Lfc(32,2) = 17$  para la conectividad *Cara-A-Celda*,  $Lcv(7,1) = 5$ ,  $Lcv(7,2) = 6$  y  $Lcv(7,3) = 10$  para la conectividad *Celda-A-Vértice* y  $Lfv(20,1) = 8$  y  $Lfv(20,2) = 12$  para la conectividad *Cara-A-Vértice*. Todo esto puede comprobarse ejecutando secuencialmente los códigos.

Prosiguiendo con el procesamiento de la geometría, se procede a continuación a desarrollar los algoritmos que permiten la obtención de los vectores tangentes y normales a cada cara, así como el área de las mismas (que en 2D no es más que la longitud del segmento). Particular interés tiene el cálculo de los vectores normales, pues debe asegurarse que éstos apuntan en la dirección exterior con respecto a cada celda para que el Teorema de la Divergencia de Gauss aplicado a la expresión (4.1) sea válido.

El cálculo de las áreas es trivial una vez obtenida la matriz de conectividad que relaciona las caras y los vértices. Matemáticamente, puede expresarse como sigue:

$$A_f = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4.4)$$

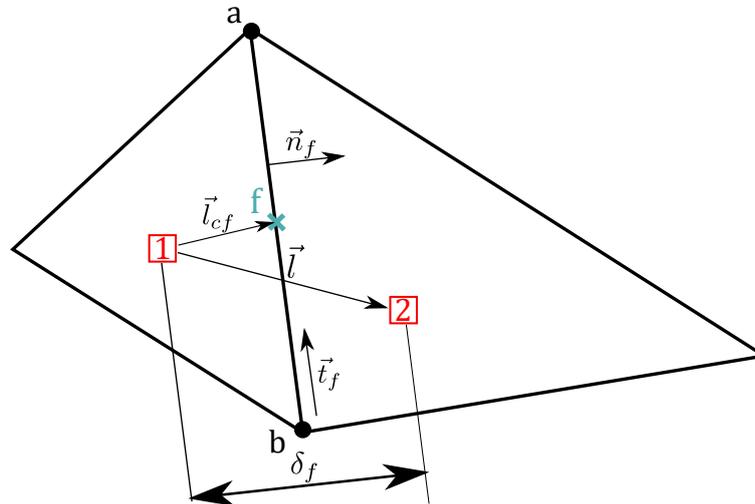
Donde los subíndices 1 y 2 hacen referencia a cada vértice de la cara. Una vez obtenida la longitud del segmento, el cálculo del vector tangente unitario es inmediato:

$$t_{x,f} = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} = \frac{x_2 - x_1}{A_f} \quad t_{y,f} = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} = \frac{y_2 - y_1}{A_f} \quad (4.5)$$

Las componentes del vector normal unitario pueden obtenerse a partir de éste resolviendo el siguiente sistema de ecuaciones:

$$\begin{cases} \vec{n} \cdot \vec{t} = 0 \\ \vec{n} \times \vec{t} = \vec{k} \end{cases} \Rightarrow \begin{cases} n_{x,f} t_{x,f} + n_{y,f} t_{y,f} = 0 \\ n_{x,f} t_{y,f} - n_{y,f} t_{x,f} = 1 \end{cases} \Rightarrow \begin{cases} n_{x,f} = t_{y,f} \\ n_{y,f} = -t_{x,f} \end{cases} \quad (4.6)$$

Una vez determinadas las componentes del vector normal unitario, es necesario desarrollar un método que discrimine la normal exterior en detrimento de la interior. Una posible solución es imponer que el vector normal calculado verifique la condición  $\vec{l}_{cf} \cdot \vec{n} > 0$ , donde  $\vec{l}_{cf}$  es el vector que une el nodo de la celda con el centro de la cara en cuestión, tal y como se ilustra en la Figura 4.3. Si dicho producto escalar es positivo, entonces el vector normal es exterior; si es negativo, bastaría con cambiar de signo las componentes de ambos vectores tangente y normal, de forma que la expresión  $\vec{n} \times \vec{t} = \vec{k}$  se siga cumpliendo. A continuación se expone el código que realiza los cálculos descritos.



**Figura 4.3** Representación de los vectores unitarios tangente y normal en una cara localmente con respecto a una celda.

```

1 %% CALCULO DE AREAS Y NORMALES Y TANGENTES UNITARIAS
2
3 lfcx = zeros(n_c,3);
4 lfcy = zeros(n_c,3);
5 area = zeros(1,n_f);
6 tx = zeros(n_c,3);
7 ty = zeros(n_c,3);
8 nx = zeros(n_c,3);
9 ny = zeros(n_c,3);
10 for ic = 1:n_c
11     for k = 1:3
12         iface = lcf(ic,k);
13         lfcx(ic,k) = xf(iface) - xc(ic);
14         lfcy(ic,k) = yf(iface) - yc(ic);
15         v1 = Lfv(iface,1);
16         v2 = Lfv(iface,2);
17         dx1 = xv(v2) - xv(v1);
18         dy1 = yv(v2) - yv(v1);
19         area(iface) = sqrt(dx1^2 + dy1^2);
20         tx(ic,k) = dx1/area(iface);
21         ty(ic,k) = dy1/area(iface);
22         nx(ic,k) = ty(ic,k);
23         ny(ic,k) = -tx(ic,k);
24         scalar_n1 = nx(ic,k)*lfcx(ic,k) + ny(ic,k)*lfcy(ic,k);
25         if scalar_n1 < 0
26             tx(ic,k) = -tx(ic,k);
27             ty(ic,k) = -ty(ic,k);
28             nx(ic,k) = -nx(ic,k);
29             ny(ic,k) = -ny(ic,k);

```

```

30     end
31   end
32 end

```

En la Figura 4.3 también aparece indicada la magnitud  $\delta = \vec{n} \cdot \vec{l}$ , donde  $\vec{l}$  es un vector que une los nodos de dos celdas contiguas. Por tanto, la magnitud escalar  $\delta_f$  representa la distancia que existe entre los centros de dos celdas colindantes en la dirección normal de la cara que las separa. Este término aparece en la formulación del FVM por lo que también es necesario calcularlo y almacenarlo.

```

1  %% CALCULO DE DELTA
2
3  delta = zeros(1,n_f);
4  for iface = 1:n_f
5      c1 = Lfc(iface,1);
6      c2 = Lfc(iface,2);
7      for ic = 1:n_c
8          for k = 1:3
9              if Lcf(ic,k) == iface
10                 nx1 = nx(ic,k);
11                 ny1 = ny(ic,k);
12             end
13         end
14     end
15     if c1 == c2 % Cara Frontera
16         dx1 = xf(iface) - xc(c1);
17         dy1 = yf(iface) - yc(c1);
18         delta(iface) = abs(dx1*nx1 + dy1*ny1);
19     else
20         dx1 = xc(c2) - xc(c1);
21         dy1 = yc(c2) - yc(c1);
22         delta(iface) = abs(dx1*nx1 + dy1*ny1);
23     end
24 end

```

El cálculo del volumen de la celda es otro dato geométrico necesario para la resolución de las ecuaciones que gobiernan el problema. En el caso 2D, este valor coincide con el área de la celda. Para el mallado bidimensional triangular en cuestión, puede hacerse uso de la definición geométrica del área de un paralelogramo para realizar éste cálculo. Así, el área del triángulo será la mitad del área del paralelogramo formado por dos vectores directores cualesquiera. Es decir:

$$V_c = \frac{1}{2} |\vec{t}_1 \times \vec{t}_2| = \frac{1}{2} (t_{x,1} t_{y,2} - t_{y,1} t_{x,2}) \quad (4.7)$$

Es importante puntualizar que en este caso los vectores  $\vec{t}_1$  y  $\vec{t}_2$  no son unitarios. El código que computa estos cálculos se expone a continuación:

```

1  %% CALCULO DEL VOLUMEN
2
3  vol(:) = 0;
4  for ic = 1:n_c
5      v1 = Lcv(ic,1);
6      v2 = Lcv(ic,2);
7      v3 = Lcv(ic,3);
8      dx1 = xv(v2) - xv(v1);
9      dy1 = yv(v2) - yv(v1);
10     dx2 = xv(v3) - xv(v1);
11     dy2 = yv(v3) - yv(v1);
12     vol(ic) = 0.5*(dx1*dy2 - dy1*dx2); % A = 1/2*|t1 x t2|

```

13 end

En la formulación del FVM no estructurado es frecuente la necesidad de calcular los valores de alguna propiedad  $\phi$  en las caras y vértices a partir de sus valores en los nodos. Por ejemplo, véase la expresión (4.3), donde se requiere conocer  $\Gamma_f$ . La forma más usual de llevar a cabo este cálculo es mediante interpolaciones lineales. Este interpolante se obtiene normalmente como una función que pondera la distancia. Como la obtención de estas funciones interpoladoras es puramente geométrica, es recomendable realizar este cómputo como parte del pre-procesamiento y almacenar la información para cuando se necesite. Así, el valor de  $\phi$  en las caras puede expresarse de la siguiente manera:

$$\phi_f = w_f \phi_1 + (1 - w_f) \phi_2 \quad \text{donde} \quad w_f = \frac{1/d_1}{1/d_1 + 1/d_2} \quad (4.8)$$

En la expresión de la función interpolante ( $w_f$ ),  $d_1$  y  $d_2$  son las distancias entre los nodos de las celdas 1 y 2 y el centro de la cara que las separa, respectivamente. Esto se ilustra en la Figura 4.4. De esta forma, cuando  $d_1 \rightarrow 0$  ó  $d_2 \rightarrow \infty$ , entonces  $\phi_f \rightarrow \phi_1$ . Contrariamente, si  $d_1 \rightarrow \infty$  ó  $d_2 \rightarrow 0$ , entonces  $\phi_f \rightarrow \phi_2$ . En esencia, la ecuación (4.8) pretende expresar cómo la influencia de un nodo contiguo a la cara decrece con la distancia a dicha cara.

Esta interpolación ponderada inversamente proporcional a la distancia no es la única opción que puede tomarse para realizar este cálculo. Existen ciertos generadores de malla que utilizan funciones interpoladoras inversamente proporcionales al cuadrado de la distancia. Este procedimiento alternativo, derivado de ciertas leyes de la física cuyas magnitudes decaen con el cuadrado de la distancia, tiene el mismo comportamiento que el detallado en el párrafo anterior, con la salvedad de que ahora el decaimiento de la influencia con la distancia es mucho más acusado. No obstante, se usará la primera opción por ser la elección mayoritariamente utilizada por los softwares CFD actuales. Se añade subsiguientemente el código que computa los valores de  $\Gamma$  en las caras.

```

1 %% FUNCIONES DE INTERPOLACION NODO-A-CARA
2
3 wfun = zeros(1,n_f);
4 gammaf = zeros(1,n_f);
5 for iface = 1:n_f
6     c1 = Lfc(iface,1);
7     c2 = Lfc(iface,2);
8     dx1 = xf(iface) - xc(c1);
9     dy1 = yf(iface) - yc(c1);
10    d1 = sqrt(dx1^2 + dy1^2);
11    dx2 = xf(iface) - xc(c2);
12    dy2 = yf(iface) - yc(c2);
13    d2 = sqrt(dx2^2 + dy2^2);
14    wfun(iface) = d2 / (d1+d2);
15    gammaf(iface) = wfun(iface)*gamma(ic1) + (1-wfun(iface))*gamma(ic2);
16 end

```

Por otro lado, para el cálculo de una propiedad  $\phi$  en los vértices, éste puede computarse empleando funciones interpoladoras con un procedimiento similar. La disimilitud con el caso anterior es que ahora un vértice puede estar influenciado por  $N$  celdas, y no únicamente por dos como en el caso de las caras. Por tanto, los interpolantes tienen la siguiente forma:

$$\phi_v = \frac{\phi_i/d_i}{\sum_{i=1}^N 1/d_i} = \sum_{i=1}^N w_{v,i} \phi_i \quad \text{donde} \quad w_{v,i} = \frac{1/d_i}{\sum_{i=1}^N 1/d_i} \quad (4.9)$$

Se expone a continuación el código MATLAB que calcula el valor de  $\phi$  en los vértices.

```

1 %% FUNCIONES DE INTERPOLACION NODO-A-VERTICE
2
3 cwfun = zeros(n_c,3);
4 sumwt = zeros(1,n_v);
5 for ic = 1:n_c
6     for iv = 1:3
7         v1 = Lcv(ic,iv);
8         d1 = sqrt((xv(v1) - xc(ic))^2 + (yv(v1) - yc(ic))^2);
9         cwfun(ic,iv) = 1 / d1;
10        sumwt(v1) = sumwt(v1) + 1/d1;
11    end
12 end
13
14 for ic = 1:n_c
15     for iv = 1:3
16         v1 = Lcv(ic,iv);
17         cwfun(ic,iv) = cwfun(ic,iv) / sumwt(v1);
18     end
19 end
20
21 % Cálculo de \phi en Vértices
22
23 phiv = zeros(1,n_v);
24 for ic = 1:n_c
25     for k = 1:3
26         iv = Lcv(ic,k);
27         if bvert(iv) == 0
28             phiv(iv) = phiv(iv) + phi_ini(ic)*cwfun(ic,k);
29         end
30     end
31 end

```

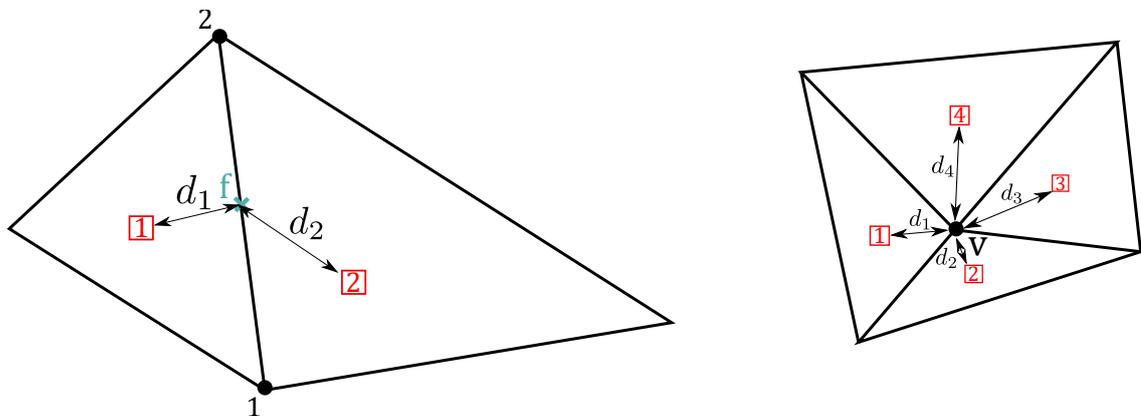


Figura 4.4 Esquema paradigmático para la obtención de las funciones interpoladoras  $w_f$  y  $w_{v,i}$ .

Por último y para terminar el procesamiento de la información relativa a la geometría se detallará el tratamiento de las condiciones de contorno, un concepto inherentemente relacionado con ésta. A la hora de almacenar la información en una formulación no estructurada las condiciones de contorno son tratadas de forma separada e independiente al resto, y también lo es su almacenamiento. Existen esencialmente tres razones de peso para hacer esto.

En primer lugar, las condiciones de contorno solo se aplican en las caras fronterizas, y por tanto

solo deben ser almacenadas en dichas caras. Debido a que el número total de celdas normalmente excede significativamente el número de celdas frontera, una formulación en la que se tengan en cuenta todas las celdas implicaría un derroche de recursos computacionales considerable.

En segundo lugar, aislar la información relativa a las condiciones de contorno permite que el almacenamiento sea más compacto y rápido de acceder para la máquina, pues a la hora de barrer la memoria siempre es más eficiente especificar un punto de partida y la longitud de la información que se requiere tratar. Así se evita que el puntero pierda tiempo barriendo celdas interiores. Esta ventaja está directamente relacionada con la arquitectura interna de la máquina.

En tercer lugar, se evita la necesidad de comprobar constantemente si la cara en cuestión es frontera o no. La ventaja de esto radica en que en la generación de códigos a gran escala el uso de comprobaciones *IF* dentro de bucles es prohibitivamente caro.

Para resolver toda esta problemática, es necesario almacenar los siguientes *arrays* de información:

- Número total de caras en la frontera:  $n_{fb}$
- Conectividad *Cara-A-CaraFrontera*:  $Lf - bf(i_f)$
- Conectividad *CaraFrontera-A-Cara*:  $Lbf - f(i_{fb})$

Además, para identificar si una cara está en la frontera o no será necesario un vector de valores binarios que al recorrer las caras devuelva 1 si la cara está en la frontera y 0 en caso contrario. Se expone seguidamente el código que lleva a cabo la numeración de las caras limítrofes y su almacenamiento. Esto será clave para la implementación de las condiciones de contorno del problema, en las que se entrará en detalle más adelante.

```

1  %% INDEXACION DE CARAS FRONTERA
2
3  n_f_b = 2*(M+N); % Total de Caras en Contorno
4  bface = zeros(1,n_f); % 1 SI Cara en Contorno, 0 SI Cara Interior
5  f_to_bf = zeros(1,n_f);
6  bf_to_f = zeros(1,n_f_b);
7  iface_b = 0;
8
9  for i = 1:M
10     ic = i;
11     iface_b = iface_b + 1;
12     bf_to_f(iface_b) = Lcf(ic,1);
13     bface(Lcf(ic,1)) = 1; % Cara en Contorno
14     f_to_bf(Lcf(ic,1)) = iface_b;
15 end
16 for j = 1:N
17     ic = M + 2*(j-1)*M;
18     iface_b = iface_b + 1;
19     bf_to_f(iface_b) = Lcf(ic,2);
20     bface(Lcf(ic,2)) = 1; % Cara en Contorno
21     f_to_bf(Lcf(ic,2)) = iface_b;
22 end
23 for i = 1:M
24     ic = n_c - (i-1);
25     iface_b = iface_b + 1;
26     bf_to_f(iface_b) = Lcf(ic,2);
27     bface(Lcf(ic,2)) = 1; % Cara en Contorno
28     f_to_bf(Lcf(ic,2)) = iface_b;

```

```

29 end
30 for j = 1:N
31     ic = n_c - (M-1) - 2*(j-1)*M;
32     iface_b = iface_b + 1;
33     bf_to_f(iface_b) = Lcf(ic,3);
34     bface(Lcf(ic,3)) = 1; % Cara en Contorno
35     f_to_bf(Lcf(ic,3)) = iface_b;
36 end

```

Llegados a este punto, el procesamiento y almacenamiento de las características geométricas está completo. En la próxima sección se retomará la ecuación principal del FVM en mallas no estructuradas presentada en (4.2) y se buscará la manera de expresar todas las magnitudes en las caras en términos de los valores en los nodos con el fin de completar la formulación.

#### 4.1.2 Discretización de las ecuaciones. Flujos normales y tangenciales

Hasta ahora se han centrado los esfuerzos en procesar y almacenar toda la información inherente al mallado del dominio computacional, de forma que tenga que computarse una sola vez y sea accesible para su fin último: aproximar con la mayor rigurosidad y eficiencia posible la ecuación que gobierna el problema objeto de estudio, representada en (4.2). De todos los términos que componen la ecuación, ya se han desarrollado expresiones robustas para la obtención de  $\Gamma_f$ ,  $A_f$  y  $\Delta V$ . Puesto que el término relativo a la fuente depende del problema (y en el caso particular del que se está estudiando su valor es nulo), restaría obtener una expresión para  $(\nabla\phi)_f \cdot \vec{n}_f$  en función de los valores en los nodos. El desarrollo que se lleva a cabo a continuación se hará en pos de dicha finalidad.

El primer paso a tomar es tener en cuenta que el gradiente de un campo escalar  $\phi$  puede reescribirse en términos de cualquier sistema de coordenadas ortogonal sin pérdida de generalidad. Así, la siguiente igualdad se cumple para el caso 2D:

$$\nabla\phi = \frac{\partial\phi}{\partial x} \vec{i} + \frac{\partial\phi}{\partial y} \vec{j} = \frac{\partial\phi}{\partial t} \vec{t} + \frac{\partial\phi}{\partial n} \vec{n} \quad (4.10)$$

Donde  $\vec{t}$  y  $\vec{n}$  son los vectores tangentes unitarios a la cara en cuestión, según el criterio especificado en la sección anterior ( $\vec{n}$  es la normal exterior a la celda y  $\vec{t}$  es el tangente por el cual se verifica que  $\vec{n} \times \vec{t} = \vec{k}$ ). Además, nótese que  $\partial\phi/\partial n = (\nabla\phi) \cdot \vec{n}$  y  $\partial\phi/\partial t = (\nabla\phi) \cdot \vec{t}$ , luego:

$$(\nabla\phi)_f = [(\nabla\phi)_f \cdot \vec{n}_f] \vec{n}_f + [(\nabla\phi)_f \cdot \vec{t}_f] \vec{t}_f \quad (4.11)$$

Si se multiplica la expresión (4.11) escalarmente por  $\vec{l}$  (donde  $\vec{l}$  es el vector representado en la Figura 4.3), se llega a:

$$(\nabla\phi)_f \cdot \vec{l} = [(\nabla\phi)_f \cdot \vec{n}_f] \delta + [(\nabla\phi)_f \cdot \vec{t}_f] \vec{t}_f \cdot \vec{l} \quad (4.12)$$

Para expresar  $(\nabla\phi)_f \cdot \vec{l}_f$  en términos de los valores en los nodos es necesario realizar una **expansión en serie de Taylor de dos variables** (debido a que la malla no es ortogonal). Esto es:

$$\left\{ \begin{array}{l} \phi_2 = \phi_f + \frac{\partial \phi}{\partial x} \Big|_f (x_2 - x_f) + \frac{\partial \phi}{\partial y} \Big|_f (y_2 - y_f) + \\ \quad \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_f (x_2 - x_f)^2 + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_f (y_2 - y_f)^2 + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_f (x_2 - x_f)(y_2 - y_f) + \dots \\ \phi_1 = \phi_f + \frac{\partial \phi}{\partial x} \Big|_f (x_1 - x_f) + \frac{\partial \phi}{\partial y} \Big|_f (y_1 - y_f) + \\ \quad \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_f (x_1 - x_f)^2 + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_f (y_1 - y_f)^2 + \frac{\partial^2 \phi}{\partial x \partial y} \Big|_f (x_1 - x_f)(y_1 - y_f) + \dots \end{array} \right. \quad (4.13)$$

Si se sustraen las dos expresiones de (4.32) se llega a lo siguiente:

$$\begin{aligned} \phi_2 - \phi_1 &= \frac{\partial \phi}{\partial x} \Big|_f (x_2 - x_1) + \frac{\partial \phi}{\partial y} \Big|_f (y_2 - y_1) + \\ &\quad \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_f [(x_2 - x_f)^2 - (x_1 - x_f)^2] + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} \Big|_f [(y_2 - y_f)^2 - (y_1 - y_f)^2] + \\ &\quad \frac{\partial^2 \phi}{\partial x \partial y} \Big|_f [(x_2 - x_f)(y_2 - y_f) - (x_1 - x_f)(y_1 - y_f)] + \dots \end{aligned} \quad (4.14)$$

Si ahora se tiene en cuenta que  $\vec{l}_f = (x_2 - x_1)\vec{i} + (y_2 - y_1)\vec{j}$ , entonces puede decirse que:

$$\phi_2 - \phi_1 \approx (\nabla \phi)_f \cdot \vec{l}_f \quad (4.15)$$

En dicha aproximación se comete un error truncado de valor  $\varepsilon$  que es difícil de estimar, a diferencia de lo que sucedía en mallas cartesianas donde  $\varepsilon \sim O(\Delta x^2)$  ó  $\varepsilon \sim O(\Delta y^2)$ . Sustituyendo (4.15) en (4.12) se obtiene finalmente:

$$(\nabla \phi)_f \cdot \vec{n}_f = \frac{\phi_2 - \phi_1}{\delta_f} - \frac{[(\nabla \phi)_f \cdot \vec{l}_f] \vec{l}_f \cdot \vec{l}_f}{\delta_f} = \frac{\phi_2 - \phi_1}{\delta_f} - \frac{J_{t,f}}{\delta_f} \quad (4.16)$$

La expresión (4.16) representa en esencia una descomposición del flujo normal a la cara  $f$  en dos componentes: una en la dirección del vector  $\vec{l}$  y otra en dirección del vector  $\vec{l}$ . En el caso particular en el que  $\vec{l}$  y  $\vec{n}$  sean colineales, entonces  $\vec{l} \cdot \vec{l} = 0$  y la formulación pasa a ser formalmente idéntica al caso de mallas estructuradas cartesianas ya estudiado. Es importante destacar que una malla cartesiana es **condición suficiente pero no necesaria** para cancelar el término tangencial; pueden existir mallas no estructuradas donde dicho término se anule (por ejemplo, una malla formada en su totalidad por triángulos equiláteros), la única condición es que los vectores  $\vec{l}$  y  $\vec{n}$  estén alineados. Esto interesa desde el punto de vista computacional, pues puede demostrarse que la existencia de dicho término en las ecuaciones dificulta la convergencia de la solución. Esto ejemplifica el enorme interés que se tiene en una correcta generación de malla y da sentido a la existencia de campos completos de las matemáticas dedicados a la generación óptima de mallas (por ejemplo, las triangulaciones de Delaunay).

Para completar el desarrollo de las ecuaciones sin pérdida de generalidad, va a suponerse el escenario en que el término asociado al flujo tangencial es no nulo. En tal caso, su cálculo requiere de un procedimiento completamente análogo al efectuado en (4.32). Así, se llega a:

$$(\nabla \phi)_f \cdot \vec{l}_f \approx \frac{\phi_a - \phi_b}{|\vec{l}_f|} \quad (4.17)$$

Donde  $\phi_a$  y  $\phi_b$  son los valores de la propiedad  $\phi$  en los vértices de la cara  $f$  (véase la Figura 4.3) y  $|\vec{l}_f| = A_f$  es la longitud de la cara (por ser el problema 2D). De esta forma, y disponiendo de la información geométrica necesaria para computar el producto escalar  $\vec{t} \cdot \vec{l}$ , el término relativo al flujo tangencial tendrá la siguiente expresión:

$$J_{t,f} = [(\nabla\phi)_f \cdot \vec{t}] \vec{t} \cdot \vec{l} = \frac{\phi_a - \phi_b}{A_f} \frac{(x_a - x_b)(x_2 - x_1) + (y_a - y_b)(y_2 - y_1)}{A_f} \quad (4.18)$$

Sustituyendo la ecuación (4.18) en (4.16), y ésta a su vez en (4.3), se llega a la expresión general que gobierna el problema de difusión pura con un mallado no estructurado:

$$\sum_{f=1}^{N_{f,p}} \Gamma_f \left[ \frac{\phi_{nb(f)} - \phi_P}{\delta_f} - \left( \frac{\phi_{a(f)} - \phi_{b(f)}}{\delta_f A_f} \right) \vec{t}_f \cdot \vec{l}_f \right] A_f + \bar{S}_\phi \Delta V = 0 \quad (4.19)$$

Donde, para el caso en cuestión,  $N_{f,p} = 3$  en todas las celdas. Además, el subíndice  $nb$  hace referencia a los nodos vecinos (del inglés *neighbour*), especificando  $nb(f)$  para hacer referencia al nodo adyacente a la cara  $f$  que se está considerando en el sumatorio.

Para completar la discretización del problema es necesario expresar de algún modo las variables en los vértices, pues son *a priori* desconocidas. Existen dos formas de acometer esta tarea. La primera de ellas consiste en expresar los valores en los vértices en términos de los valores en los nodos haciendo uso de las funciones interpoladoras  $w_{v,i}$ , descritas en (4.9). Así, las únicas incógnitas que presentarían las ecuaciones discretas serían los valores en los nodos, que es la situación que se andaba buscando. Esto se conoce como procedimiento **implícito**. La segunda forma de abordar el tratamiento de los valores en los vértices es computarlos según los valores de la iteración anterior, de forma que pueden pasarse al segundo miembro de la ecuación como términos conocidos. Este es el llamado tratamiento **explícito**.

Son varias las ventajas y desventajas de cada uno de estos métodos. El implícito tiene la ventaja de presentar una convergencia más rápida a la solución, pero la matriz de coeficiente perderá parcialmente su cualidad como **matriz dispersa** (cuanto más dispersa es la matriz, más eficientes son los algoritmos que se emplean en este tipo de matrices y menos memoria se requiere) al aparecer términos relativos a vecinos más lejanos en cada ecuación discreta. Además, dificulta la implementación de las condiciones de contorno. Por otro lado, el método explícito mantiene la matriz mayoritariamente dispersa al incluir únicamente las celdas inmediatamente vecinas, pero tiene una convergencia más lenta. Por las razones expuestas, en la industria del CFD el tratamiento explícito es de los dos el que tiene un uso más extendido, y será por tanto el que se utilice en adelante.

Resulta lógico pensar que la convergencia del método explícito estará condicionada por la importancia relativa del término tangencial frente al normal. Esto es así porque la diferencia entre dos iteraciones sucesivas será más notable si el valor de  $J_{t,f}$  juega un papel determinante. De este sencillo análisis puede inferirse que cuanto menos alineados estén los vectores  $\vec{n}$  y  $\vec{l}$ , mayor será  $J_{t,f}$  y más pobre será la convergencia. En la literatura se emplea el término *skewness* para hacer referencia a la desalineación de estos vectores. Como ya se ha comentado en otras ocasiones, es de vital importancia que los generadores de malla tengan en cuenta este concepto durante el pre-procesamiento para que la computación posterior tenga mejores resultados. Así, el quid de la cuestión consistirá en evitar las celdas con ángulos extremadamente agudos u obtusos y tratar de que los  $60^\circ$  en mallas triangulares y los  $90^\circ$  en mallas cuadrangulares prevalezcan.

Para terminar con el planteamiento de las ecuaciones, es necesario puntualizar que la expresión general (4.19) se verá modificada por las condiciones de contorno en las celdas limítrofes. Para

ilustrar este tema, se ejemplificarán los tipos de condiciones de contorno analizando directamente el problema de la Figura 3.2.

### Cara norte. Flujo de calor impuesto

La cara norte del dominio computacional está sujeta a la condición de flujo de calor impuesto de valor conocido (saliente). Matemáticamente, esto se expresa como  $(\nabla\phi)_N \cdot \vec{n}_N = J_N$ . Así, de acuerdo con la Figura 4.2, la ecuación en las celdas 17 y 18 queda como sigue:

$$\sum_{\substack{f=1 \\ f \neq N}}^{N_{f,P}} \Gamma_f \left[ \frac{\phi_{nb(f)} - \phi_P}{\delta_f} - \left( \frac{\phi_{a(f)}^0 - \phi_{b(f)}^0}{\delta_f A_f} \right) \vec{t}_f \cdot \vec{l}_f \right] A_f + \Gamma_N J_N A_N + \bar{S}_\phi \Delta V = 0 \quad (4.20)$$

### Cara oeste. Aislada térmicamente

La cara oeste tiene la condición de aislamiento térmico (flujo difusivo nulo a través de esta cara). Este caso es similar al anterior, salvo que ahora  $(\nabla\phi)_W \cdot \vec{n}_W = 0$ . Esto afecta a las celdas 4 y 10 según la Figura 4.2, y la expresión final a la que se llega es:

$$\sum_{\substack{f=1 \\ f \neq W}}^{N_{f,P}} \Gamma_f \left[ \frac{\phi_{nb(f)} - \phi_P}{\delta_f} - \left( \frac{\phi_{a(f)}^0 - \phi_{b(f)}^0}{\delta_f A_f} \right) \vec{t}_f \cdot \vec{l}_f \right] A_f + \bar{S}_\phi \Delta V = 0 \quad (4.21)$$

### Cara este. Flujo de calor impuesto

Caso formalmente idéntico al que se tiene en la cara norte, aunque aquí el flujo es entrante. La condición se expresa como  $(\nabla\phi)_E \cdot \vec{n}_E = J_E$  y la ecuación matemática que se verifica en las celdas 9 y 15 es:

$$\sum_{\substack{f=1 \\ f \neq E}}^{N_{f,P}} \Gamma_f \left[ \frac{\phi_{nb(f)} - \phi_P}{\delta_f} - \left( \frac{\phi_{a(f)}^0 - \phi_{b(f)}^0}{\delta_f A_f} \right) \vec{t}_f \cdot \vec{l}_f \right] A_f + \Gamma_E J_E A_E + \bar{S}_\phi \Delta V = 0 \quad (4.22)$$

### Cara sur. Temperatura impuesta

En la cara sur se cumple la condición  $\phi = \phi_S$ . Este caso difiere ligeramente de los presentados anteriormente. De nuevo, se divide el sumatorio entre las caras interiores y la cara frontera, pero esta vez el valor  $(\nabla\phi)_S \cdot \vec{n}_S$  no es conocido, sino que tiene que ser reescrito en términos del flujo normal y tangencial, tal y como se detalló en el desarrollo de la expresión general. Finalmente, se llega a la siguiente expresión, válida para las celdas 1 y 2:

$$\sum_{\substack{f=1 \\ f \neq S}}^{N_{f,P}} \Gamma_f \left[ \frac{\phi_{nb(f)} - \phi_P}{\delta_f} - \left( \frac{\phi_{a(f)}^0 - \phi_{b(f)}^0}{\delta_f A_f} \right) \vec{t}_f \cdot \vec{l}_f \right] A_f + \Gamma_S \left[ \frac{\phi_S - \phi_P}{\delta_S} - \left( \frac{\phi_{a(S)}^0 - \phi_{b(S)}^0}{\delta_S A_S} \right) \vec{t}_S \cdot \vec{l}_S \right] A_S + \bar{S}_\phi \Delta V = 0 \quad (4.23)$$

Es importante resaltar que en esta expresión el vector  $\vec{l}_S$  va desde el nodo de la celda  $P$  hasta el centro de la cara frontera y no hasta el nodo de la celda vecina, pues no existe una celda colindante en esa dirección. Puesto que el segundo término del primer miembro de la ecuación es conocido, deberá ser tratado dentro del vector de términos independientes. Especial mención merece el caso en que la condición de contorno es constante a lo largo de toda la frontera. En dicha situación,

se verifica  $\phi_{a(s)} = \phi_{b(s)}$ , luego el término tangencial se cancela al no existir gradiente en esa dirección.

El resto de celdas frontera (a decir, 3 y 16) presentan una doble condición de contorno por pertenecer a esquinas del dominio computacional. La discretización en dichas celdas es bastante directa y no se detallará para evitar redundancia. Para su obtención no habría más que tener en cuenta simultáneamente las dos condiciones de contorno que apliquen en cada caso. Una vez desarrolladas las ecuaciones que han de cumplirse de forma discretizada para cada celda y analizada la implementación de las condiciones de contorno, se está en condiciones de ensamblar y resolver el sistema. Esto se discutirá en la siguiente sección.

#### 4.1.3 Resolución del sistema. Ensamblaje de las matrices

Las ecuaciones discretizadas obtenidas en la sección anterior requieren una reestructuración de forma que el sistema pueda resolverse sencillamente mediante el comando \ de MATLAB ( $\phi = A \backslash b$ ). Para ello, se comienza reescribiendo la expresión (4.19) como:

$$A_{p,P} \phi_P + \sum_{k=1}^{N_{f,P}} A_{k,P} \phi_{k(P)} = \bar{S}_\phi \Delta V + S_{skew,P} = b_P \quad (4.24)$$

En esta expresión, válida para las celdas interiores,  $A_{p,P}$  denota los coeficientes de la matriz  $A$  ubicados en la diagonal, mientras que  $A_{k,P}$  hace referencia a los elementos no nulos de  $A$  fuera de la diagonal pero dentro de esa misma fila. Es decir, la posición de los términos  $A_{k,P}$  dentro de la fila  $P$  estarán ubicados en las columnas pertenecientes a las celdas vecinas a  $P$ , bajo la indexación global detallada en la Figura 4.2. El término perteneciente a la fuente ha pasado al segundo miembro por ser una cantidad conocida. El signo que lo precede continúa siendo positivo porque, como se verá a continuación, se ha cambiado el signo a toda la ecuación para que los términos diagonales de  $A$  sean positivos de acuerdo con el convenio establecido. Por último, el término  $S_{skew,P}$  denota la contribución de los flujos tangenciales. Está en el segundo miembro por ser una cantidad conocida debido al tratamiento explícito de los valores en los vértices, tal y como se detalló en la sección precedente. Con todo, los valores que conforman el sistema pueden obtenerse mediante las siguientes expresiones:

$$A_{p,P} = \sum_{f=1}^{N_{f,P}} \frac{\Gamma_f A_f}{\delta_f} \quad (4.25)$$

$$A_{k,P} = - \frac{\Gamma_{k,P} A_{k,P}}{\delta_{k,P}} \quad (4.26)$$

$$A_{skew,P} = - \sum_{f=1}^{N_{f,P}} \Gamma_f \left( \frac{\phi_{a(f)}^0 - \phi_{b(f)}^0}{\delta_f} \right) \vec{t}_f \cdot \vec{l}_f \quad (4.27)$$

Es importante recordar que las expresiones anteriores pueden sufrir modificaciones debido a las condiciones de contorno cuando  $P$  haga referencia a una celda limítrofe. Para computar estas fórmulas, es necesario ensamblar la matriz  $A$  y el vector  $b$ , siendo precavidos con la inclusión de las condiciones de contorno. A continuación se exponen los códigos que llevan a cabo esta tarea.

```

1 %% CALCULO DE LA SOLUCION
2
3 err = 10^-3; % Error inicial > TOL para entrar en el bucle
4 TOL = 10^-3; % Tolerancia
5 gamma = 1000*ones(1,n_c);
6

```

```

7  gammaf = zeros(1,n_f);
8  for iface = 1:n_f
9      ic1 = Lfc(iface,1);
10     ic2 = Lfc(iface,2);
11     gammaf(iface) = wfun(iface)*gamma(ic1) + (1-wfun(iface))*gamma(ic2);
12 end
13
14 flag = 0;
15 while max(err,[],'all') > TOL
16
17     A = zeros(n_c,n_c); % Matriz de coeficientes
18     b = zeros(1,n_c); % Vector de términos independientes
19     A0 = zeros(1,n_c); % Elementos diagonales de A
20     Anb = zeros(n_c,3); % Elementos extra-diagonales de A
21     skew = zeros(1,n_c); % Término tangencial (skewness)
22     sc = zeros(1,n_c); % Término fuente
23     phi_top = 100; % Condición de Contorno en Cara Norte
24     q_west = 500*10^-3; % Condición de Contorno en Cara Oeste
25
26     % Condiciones de Contorno (Norte: \phi impuesto; Resto: flujo impuesto)
27
28     phib = zeros(1,n_f_b);
29     Jb = zeros(1,n_f_b);
30     for iface_b = 1:n_f_b
31         iface = bf_to_f(iface_b);
32         if yf(iface) > 0.99*H
33             phib(iface_b) = phi_top;
34         end
35         if xf(iface) < 0.01*L
36             Jb(iface_b) = q_west;
37         end
38     end
39
40     % Coeficientes de la Matriz A
41
42     for ic = 1:n_c
43         for k = 1:3
44             iface = Lcf(ic,k);
45             if bface(iface) == 0
46                 A0(ic) = A0(ic) + gammaf(iface)*area(iface)/delta(iface);
47                 Anb(ic,k) = - gammaf(iface)*area(iface)/delta(iface);
48                 ic1 = Lfc(iface,1);
49                 ic2 = Lfc(iface,2);
50                 if ic1 == ic
51                     A(ic,ic2) = Anb(ic,k);
52                 else
53                     A(ic,ic1) = Anb(ic,k);
54                 end
55             elseif bface(iface) == 1 && phib(f_to_bf(iface)) ~= 0
56                 ifb = f_to_bf(iface);
57                 A0(ic) = A0(ic) + gammaf(iface)*area(iface)/delta(iface);
58                 Anb(ic,k) = 0;
59             else
60                 A0(ic) = A0(ic) + 0;
61                 Anb(ic,k) = 0;
62             end
63             A(ic,ic) = A0(ic);
64         end
65     end
66
67     if flag == 1
68

```

```

69 % Cálculo de \phi en Vértices
70
71 phiv = zeros(1,n_v);
72 for ic = 1:n_c
73     for k = 1:3
74         iv = Lcv(ic,k);
75         if bvert(iv) == 0
76             phiv(iv) = phiv(iv) + phi_ini(ic)*cwfun(ic,k);
77         else
78             if yv(iv) > 0.99*H
79                 phiv(iv) = phi_top;
80             else
81                 phiv(iv) = phiv(iv) + phi_ini(ic)*cwfun(ic,k);
82             end
83         end
84     end
85 end
86
87 % Término Independiente (Fuente + Skew)
88
89 sign = zeros(n_c,3);
90 for ic = 1:n_c
91     skew(ic) = 0;
92     for k = 1:3
93         iface = Lcf(ic,k);
94         if bface(iface) == 0
95             va = Lfv(iface,1);
96             vb = Lfv(iface,2);
97             ic1 = Lfc(iface,1);
98             ic2 = Lfc(iface,2);
99             dx1 = xc(ic2)-xc(ic1);
100            dy1 = yc(ic2)-yc(ic1);
101            tx1 = xv(vb) - xv(va);
102            ty1 = yv(vb) - yv(va);
103            t_unit = [tx1,ty1]./area(iface);
104            scalar_t1 = t_unit(1)*dx1 + t_unit(2)*dy1;
105            scalar_n1 = nx(ic,k)*dx1 + ny(ic,k)*dy1;
106            if scalar_n1 > 0
107                sign(ic,k) = 1;
108            else
109                sign(ic,k) = -1;
110            end
111            skew(ic) = skew(ic) - gammaf(iface)*scalar_t1*...
112                (phiv(vb) - phiv(va))*sign(ic,k)/delta(iface);
113        end
114    end
115    b(ic) = skew(ic) - sc(ic)*vol(ic);
116 end
117 end
118
119 for ibf = 1:n_f_b
120     iface = bf_to_f(ibf);
121     ic1 = Lfc(iface,1);
122     b(ic1) = b(ic1) + gammaf(iface)*area(iface)*phib(ibf)/delta(iface) + ...
123         + area(iface)*Jb(ibf);
124 end
125
126 phi = A\b';
127 if flag == 1
128     error = abs(phi - phi_ini);
129 end
130 phi_ini = phi;

```

```

131     flag = 1;
132 end

```

Es necesario resaltar que los cálculos se hacen de forma iterativa hasta que la diferencia entre dos soluciones consecutivas sea menor que una tolerancia establecida. Será entonces cuando pueda afirmarse que el método ha convergido a la solución deseada. Nótese además cómo, para comenzar los cálculos, hay que establecer un valor inicial de  $\phi$  en los nodos para el primer cómputo de  $\phi$  en los vértices, y así tener un punto de partida para el proceso iterativo. Elegir un valor inicial adecuado puede ser determinante a la hora de mejorar la eficiencia del método. La solución que aquí se propone consiste en tomar los valores de  $\phi$  que se obtienen resolviendo el problema sin considerar el término de *skewness*, lo cual resultará una buena aproximación si en la geometría de la malla en cuestión predominan los valores bajos del producto escalar  $\vec{\tau} \cdot \vec{l}$ . La definición de la variable binaria *flag* que se observa en los códigos arriba descritos viene motivada por esta decisión.

Existen otras formas de optimizar este método para lograr una convergencia más rápida, y una de ellas es utilizar el comando `\` de MATLAB, tal y como está haciendo. Esta herramienta funciona como una “caja negra” a la hora de resolver el sistema, y el algoritmo interno trata de buscar la convergencia de la solución para los valores de  $\phi$  en los vértices de la iteración en cuestión. Tras ello, éstos valores se actualizan y el algoritmo interno del comando `\` vuelve a funcionar tratando de conseguir una **convergencia parcial** antes de actualizar los nuevos valores. Computacionalmente, esto es mucho más eficiente que tratar de alcanzar la convergencia total en cada iteración.

Por si resultara de interés al lector, se adjunta a continuación un código que muestra una de las muchas formas posibles que existen para post-procesar la solución mediante representaciones gráficas ejemplares.

```

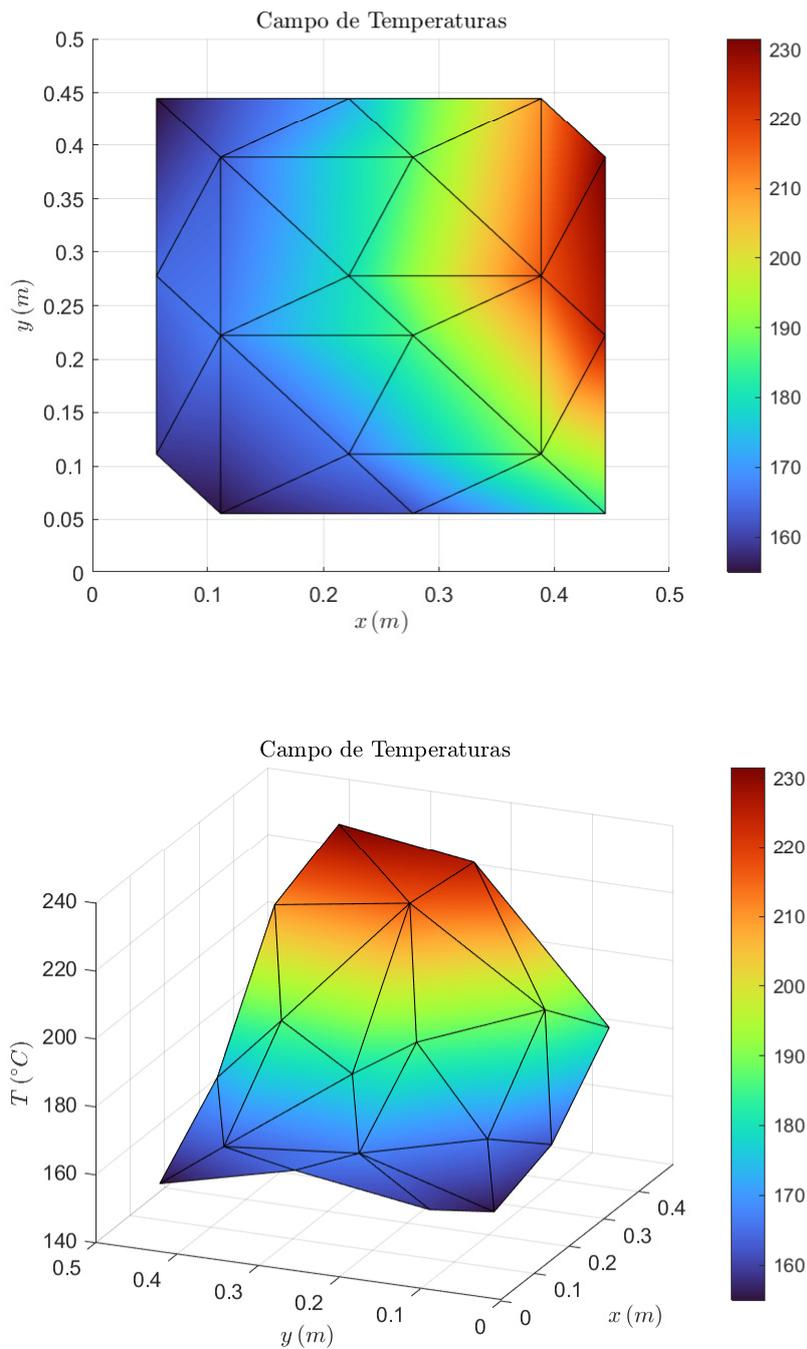
1 %% REPRESENTACION GRAFICA
2
3 figure(1)
4 tri = delaunay(xc, yc);
5 h=trisurf(tri, xc, yc, phi);
6 colormap(turbo)
7 colorbar
8 shading interp
9 set(h,'EdgeColor','k')
10 xlabel('$x$, (m)$', 'Interpreter', 'latex')
11 ylabel('$y$, (m)$', 'Interpreter', 'latex')
12 zlabel('$T$, (~{\circ}C)$', 'Interpreter', 'latex')
13 title('Campo de Temperaturas', 'Interpreter', 'latex')
14
15 figure(2)
16 xq=min(xc):(max(xc)-min(xc))/200:max(xc);
17 yq=min(yc):(max(yc)-min(yc))/200:max(yc);
18 [Xq, Yq] = meshgrid(xq,yq);
19 Zq =griddata(xc,yc,phi,Xq,Yq);
20 contour(Xq,Yq,Zq,110:15:260,'Linewidth',2,'ShowText','on');
21 shading interp
22 colormap(turbo)
23 xlabel('$x$, (m)$', 'Interpreter', 'latex')
24 ylabel('$y$, (m)$', 'Interpreter', 'latex')
25 zlabel('$T$, (~{\circ}C)$', 'Interpreter', 'latex')
26 title('Curvas de nivel del Campo de Temperaturas', 'Interpreter', 'latex')

```

Para finalizar con este capítulo, se dedica la siguiente sección a obtener y valorar los resultados obtenidos mediante la formulación no estructurada del FVM aplicada al problema de la Figura 3.2.

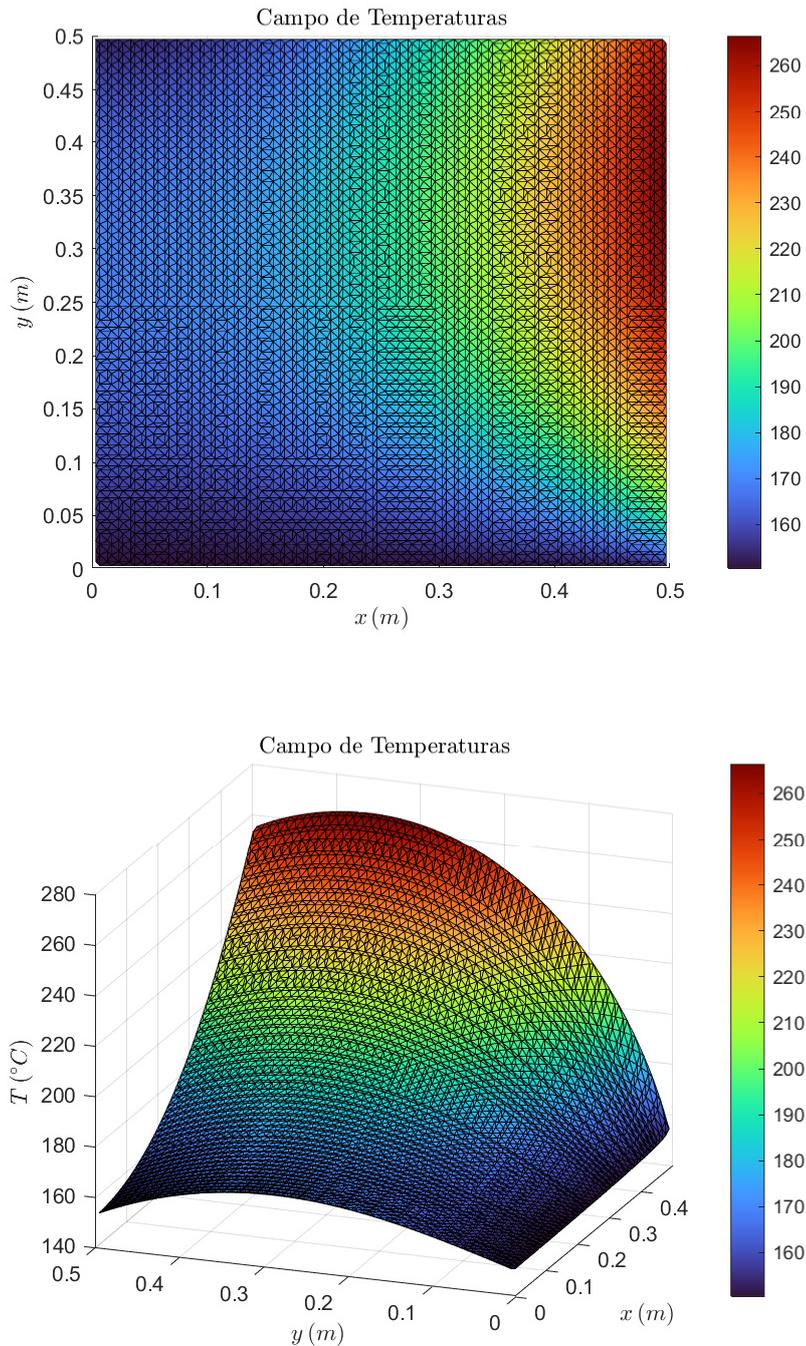
## 4.1.4 Aplicación práctica 2D. Mallado no estructurado

Para ilustrar la formulación del FVM en mallas no estructuradas desarrollada en los capítulos previos, considérese el problema paradigmático descrito en la Sección 3.1.4. Ejecutando de manera secuencial los códigos expuestos, se llega a la solución mostrada en la Figura 4.5. Es importante destacar que la malla triangular representada no coincide con la que verdaderamente se está considerando, sino que se está haciendo uso de la generada por el comando *delaunay*. Esta es una forma sencilla de obtener una ilustración característica de la superficie.



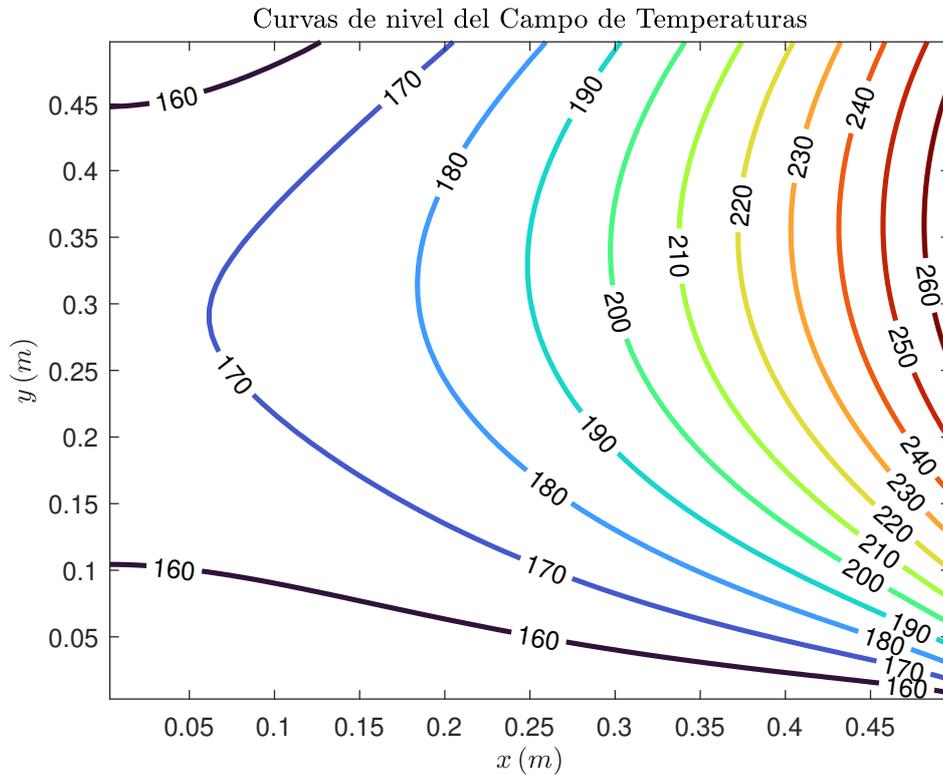
**Figura 4.5** Valores del campo de temperaturas en los nodos para un mallado no estructurado  $M \times N$  con  $M = 3$  y  $N = 3$ .

Como puede observarse, los resultados obtenidos son esencialmente idénticos a los que se lograron con una malla estructurada cartesiana, representados en la Figura 3.4. Nótese además cómo en el caso de mallas no estructuradas, debido al algoritmo desarrollado, una malla  $M \times N$  no cuenta con  $M \times N$  celdas, sino con el doble ( $M \times N$  cuadriláteros divididos en dos triángulos según una de sus diagonales, tal y como se observa en la Figura 4.2). Además, como el código se ha desarrollado sin pérdida de generalidad, la malla puede refinarse cuanto se quiera. Por ejemplo, a modo comparativo, con un mallado equivalente al empleado en la Figura 4.15 se obtienen las siguientes ilustraciones.



**Figura 4.6** Valores del campo de temperaturas en los nodos para un mallado no estructurado  $M \times N$  con  $M = 50$  y  $N = 50$ .

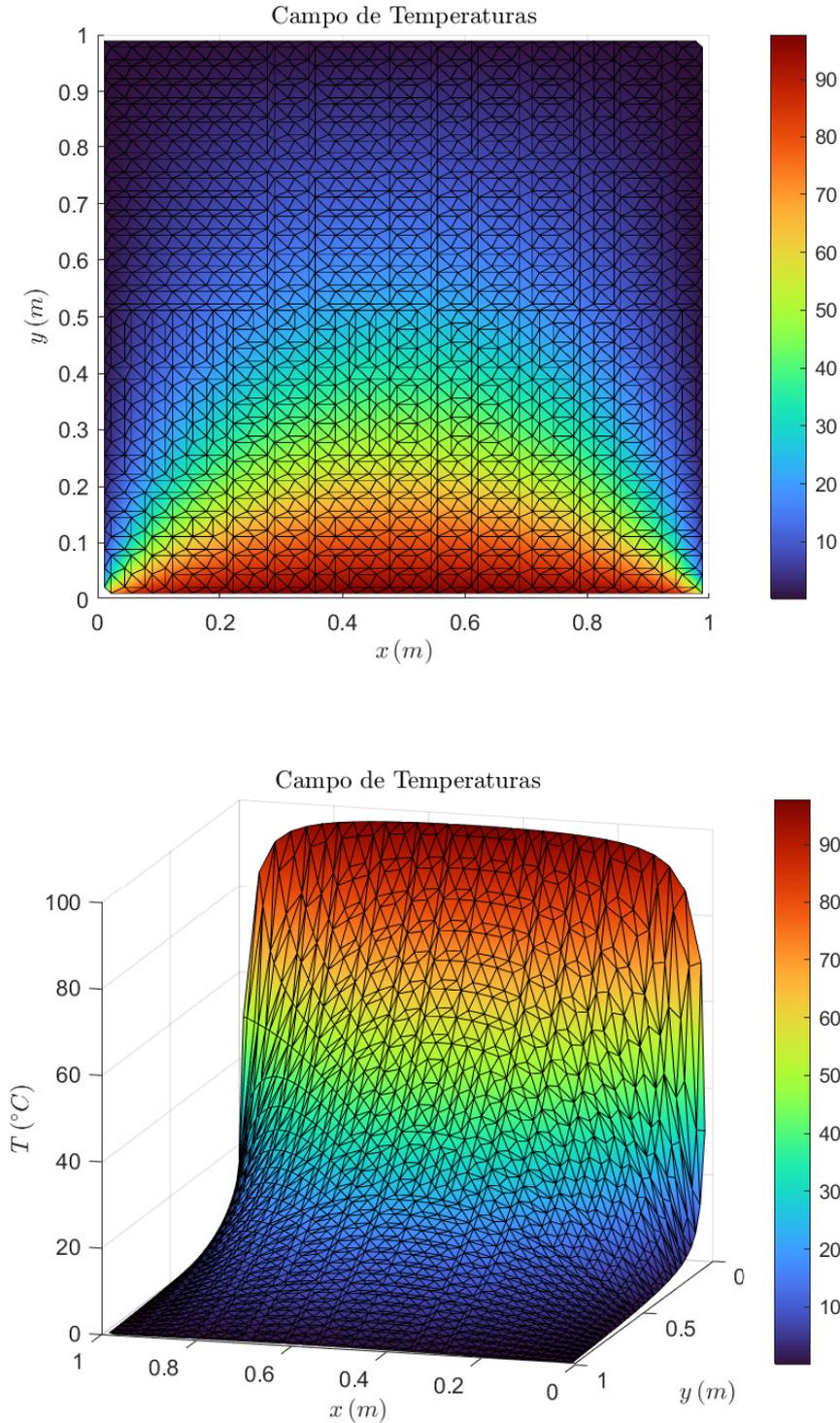
La representación de las curvas de nivel, que simbolizan las líneas de temperatura constante, es muy buen indicativo de la validez de la solución. En efecto, se observa cómo la temperatura en la pared inferior anda rondando los  $150^{\circ}\text{C}$ , tal y como debería ser por la condición de contorno impuesta en dicha cara. Por otro lado, la condición de aislamiento térmico (que matemáticamente se expresa como  $\nabla\phi \cdot \vec{n} = 0$ ) se advierte fácilmente en la cara oeste, pues las isotermas entran en ella de forma perpendicular. Estos comentarios pueden confirmarse con la Figura 4.7.



**Figura 4.7** Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado  $M \times N$  con  $M = 50$  y  $N = 50$ .

Para aumentar la aplicabilidad del código desarrollado y validar su robustez, se pasa a continuación a analizar dos problemas similares al recién estudiado pero modificando las condiciones de contorno. Resulta interesante destacar que en el capítulo de mallas estructuradas las ecuaciones impuestas por las condiciones de contorno debían cambiarse manualmente en cada celda. Al contrario de esto, la gestión y almacenamiento de la información elaborada en el código de mallas no estructuradas permite estudiar problemas fundamentalmente diferentes alterando únicamente unas pocas líneas de código, tal y como se verá a continuación.

Por ejemplo, considérese el siguiente problema similar al descrito en la Sección 3.1.4. Se tiene una placa plana de  $k = 1000\text{W}/(\text{mK})$  y dimensiones  $L \times H$ , con  $L = 1\text{m}$  y  $H = 1\text{m}$  y con las condiciones de contorno siguientes:  $T_W = T_N = T_E = 0^{\circ}\text{C}$  y  $T_S = 100^{\circ}\text{C}$ . Este escenario pretende imitar el problema descrito por [6] y [17]-[20]. En este caso, la solución que se obtiene es la que se observa en las Figuras 4.8 y 4.9. En efecto, los resultados son los que cabía esperar según las condiciones de contorno impuestas, y están en perfecta consonancia con los que proporciona la literatura.



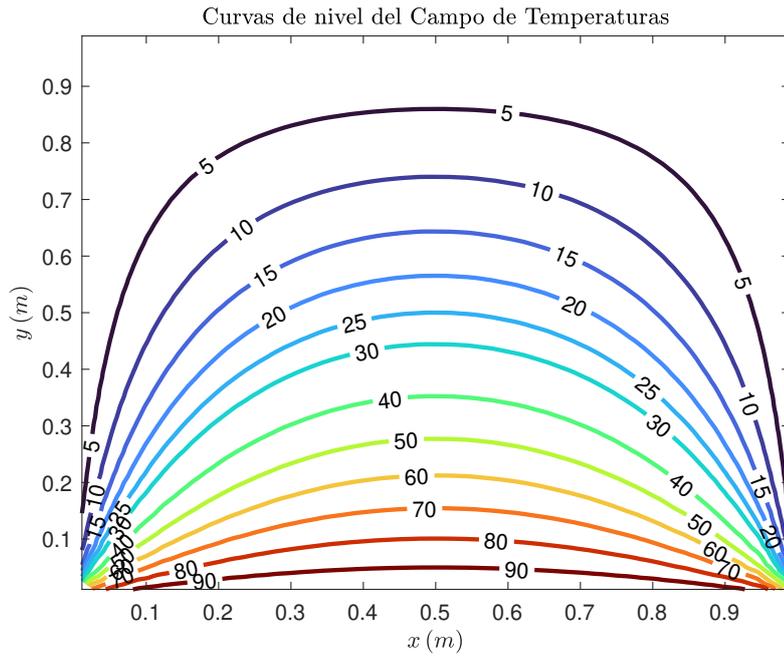
**Figura 4.8** Valores del campo de temperaturas en los nodos para un mallado no estructurado  $M \times N$  con  $M = 30$  y  $N = 30$  en el problema analizado en [6] y [17]-[20].

Obsérvese a continuación el único cambio a realizar dentro del código para obtener los resultados expuestos. El resto de líneas del programa permanecen inalteradas. Notar que no se ha asignado un valor idénticamente igual a cero (sino un valor ínfimo) en las condiciones de contorno de  $\phi = 0$  para ser consistentes con la arquitectura del código.

```

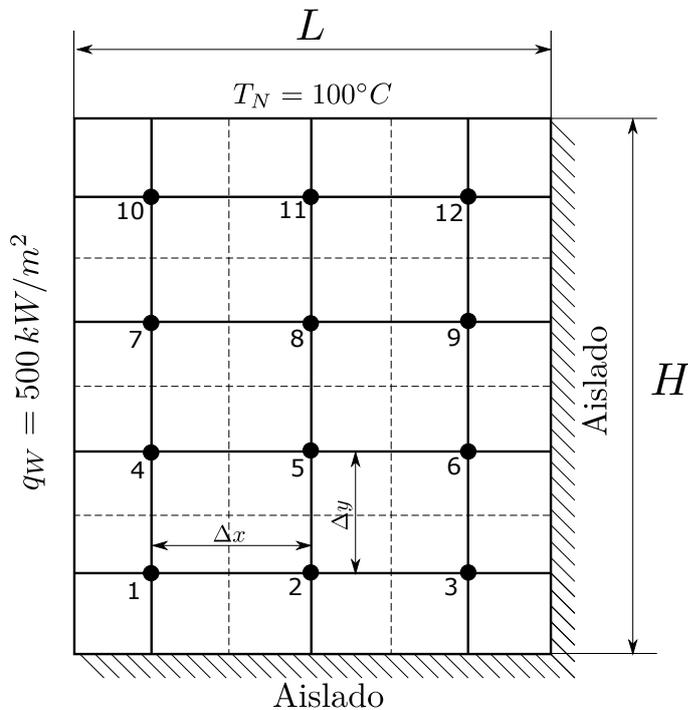
1  %% CAMBIOS PARA MODIFICACION DE CONDICIONES DE CONTORNO
2
3  % Condiciones de Contorno (phi impuesto en todas las caras)
4
5  phi_south = 100;
6  phi_east = 0.001;
7  phi_west = 0.001;
8  phi_north = 0.001;
9  phib = zeros(1,n_f_b);
10 for iface_b = 1:n_f_b
11     iface = bf_to_f(iface_b);
12     if xf(iface) < 0.01*L
13         phib(iface_b) = phi_west;
14     end
15     if xf(iface) > 0.99*L
16         phib(iface_b) = phi_east;
17     end
18     if yf(iface) < 0.01*H
19         phib(iface_b) = phi_south;
20     end
21     if yf(iface) > 0.99*H
22         phib(iface_b) = phi_north;
23     end
24 end
25
26 % Cálculo de phi en Vértices
27
28 if flag == 1
29     phiv = zeros(1,n_v);
30     for ic = 1:n_c
31         for k = 1:3
32             iv = Lcv(ic,k);
33             if bvert(iv) == 0
34                 phiv(iv) = phiv(iv) + phi_ini(ic)*cwfun(ic,k);
35             else
36                 if xv(iv) < 0.01*L
37                     phiv(iv) = phi_west;
38                 elseif xv(iv) > 0.99*L
39                     phiv(iv) = phi_east;
40                 elseif yv(iv) < 0.01*H
41                     phiv(iv) = phi_south;
42                 elseif yv(iv) > 0.99*H
43                     phiv(iv) = phi_north;
44                 else
45                     phiv(iv) = phiv(iv) + phi_ini(ic)*cwfun(ic,k);
46                 end
47             end
48         end
49     end
50 end

```



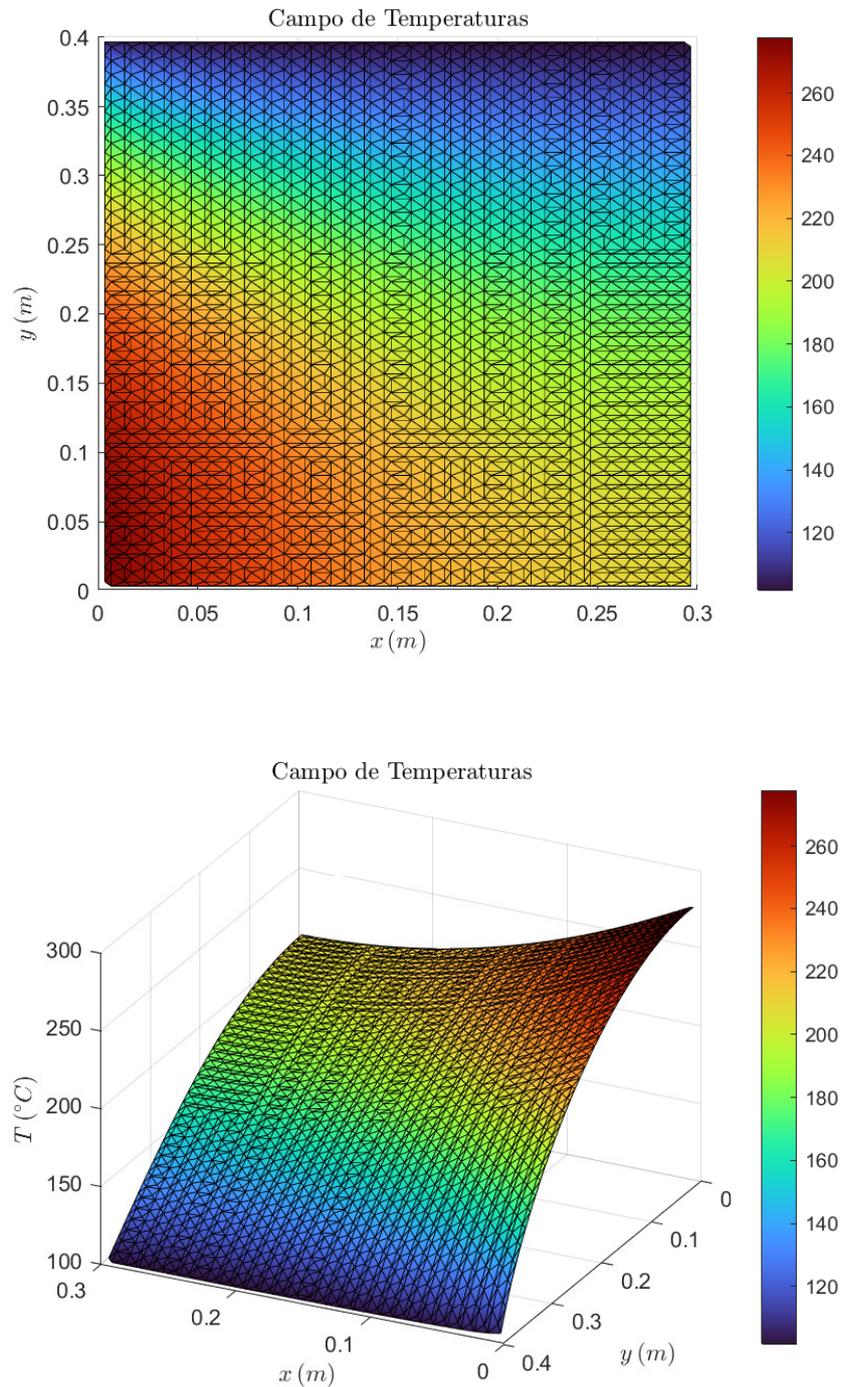
**Figura 4.9** Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado  $M \times N$  con  $M = 30$  y  $N = 30$  en el problema analizado en [6] y [17]-[20].

Por si esto no confiriase suficiente validez al algoritmo desarrollado, se estudiará subsiguientemente uno de los problemas propuestos en [4] para la exposición del FVM en geometrías bidimensionales. Dicho problema es nuevamente similar al descrito en la Sección 3.1.4. Consiste en una placa bi-dimensional con  $k = 1000 W/(mK)$  y de dimensiones  $L \times H$ , con  $L = 0.3 m$  y  $H = 0.4 m$ . Las condiciones de contorno se especifican en la siguiente figura:

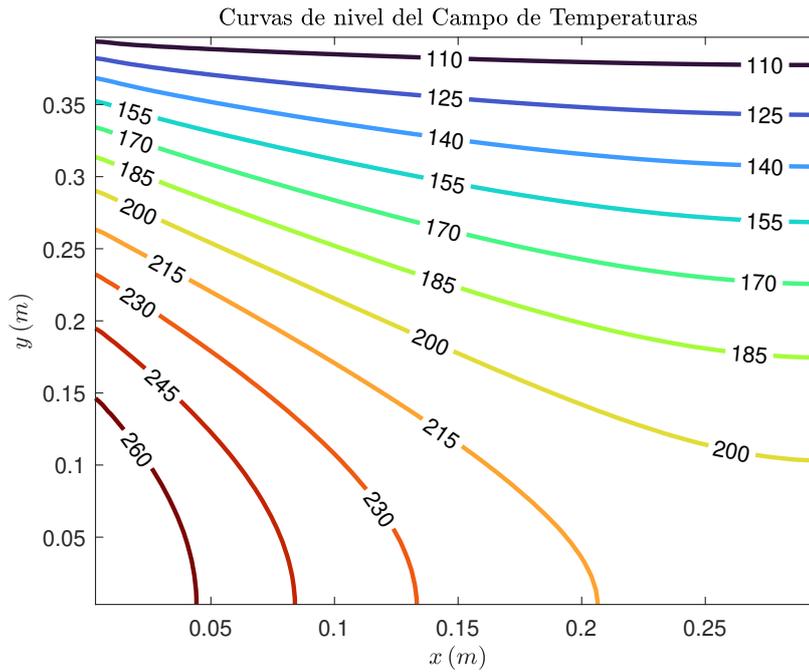


**Figura 4.10** Representación del problema descrito en [4].

Nuevamente, modificando únicamente las condiciones de contorno, se obtienen los siguientes resultados:



**Figura 4.11** Valores del campo de temperaturas en los nodos para un mallado no estructurado  $M \times N$  con  $M = 30$  y  $N = 40$  en el problema analizado en [4].



**Figura 4.12** Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado  $M \times N$  con  $M = 30$  y  $N = 40$  en el problema analizado en [4].

Los resultados son congruentes con la física del problema (véase la perpendicularidad de las isotermas en las caras aisladas y el cumplimiento de la condición de temperatura impuesta) y coinciden exactamente con los que reporta la literatura en cuestión. Con todo esto, se ha evidenciado que la formulación del FVM con mallas no estructuradas para problemas difusivos es totalmente correcta. Se dará un paso más en la formulación añadiendo el término advectivo en la próxima sección.

## 4.2 Problema convectivo-difusivo, estacionario y unidimensional

En busca de la completitud de la caracterización del FVM en mallas no estructuradas, se dedica esta sección a la generalización de la formulación para problemas convectivo-difusivos. Como ya se ha visto en capítulos anteriores, se retoma a continuación la ecuación general que gobierna un problema convectivo-difusivo no estacionario en forma diferencial, que ya fue expuesta en (1.5):

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\vec{v}\phi) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi \tag{4.28}$$

Como ya se sabe, el FVM consiste en integrar la ecuación (4.28) sobre un volumen de control genérico, llegando a la ecuación (1.6). Puesto que ya se han analizado con minuciosidad cada uno de los términos de esta expresión, se tratará a continuación de obtener únicamente una expresión útil para el flujo advectivo, evitando así ser redundantes. El término transitorio se supondrá nulo en este capítulo. No obstante, su caracterización fue ya estudiada en profundidad en la Sección 2.3. A modo de sucinto resumen, puede decirse que su aproximación se sustenta en tomar el valor medio y adopta la siguiente expresión:

$$\int_{VC} \frac{\partial}{\partial t} (\rho\phi) dV = \frac{\partial(\rho\phi)}{\partial t} \Big|_P \Delta V = \left[ \frac{(\rho\phi)_P - (\rho\phi)_P^0}{\partial t} \right] \Delta V \tag{4.29}$$

Integrando con respecto al tiempo y aplicando alguno de los esquemas conocidos (implícito, explícito o Crank-Nicolson) se tendría completamente caracterizado este término. Volviendo al término convectivo, su integración lleva a la siguiente expresión:

$$\int_{VC} \nabla \cdot (\rho \vec{v} \phi) = \int_S (\rho \vec{v} \phi) \cdot \vec{n} dA = \sum_{f=1}^{N_{f,P}} (\rho \vec{v} \cdot \vec{n} A)_f \phi_f = \sum_{f=1}^{N_{f,P}} G_f \phi_f \quad (4.30)$$

Donde se ha aplicado el Teorema de la Divergencia de Gauss y se ha tenido en cuenta que la malla considerada está constituida por celdas delimitadas por segmentos rectos. Además, se ha definido el gasto másico que atraviesa cada cara como  $G_f = (\rho \vec{v} \cdot \vec{n} A)_f$ , donde  $\vec{n}$  es la normal externa. Notar también que nuevamente en el caso de malla triangular se tiene que  $N_{f,P} = 3$  para cualquier celda.

El siguiente paso a seguir consiste en computar el valor del flujo  $G_f$ . Suponiendo que el campo de velocidades es un dato del problema (si no lo fuera, habría que resolver primero las ecuaciones del flujo), los términos que influyen en el cálculo del gasto son magnitudes conocidas. Por tanto, pueden obtenerse como parte del pre-procesamiento. El código es considerablemente sencillo y se adjunta a continuación.

```

1 %% GASTO MASICO
2
3 rho = 1000;
4 cp = 4200;
5 um = 2.92*10^-7;
6 u = zeros(1,n_f);
7 G = zeros(n_c,3);
8 for ic = 1:n_c
9     for k = 1:3
10        iface = Lcf(ic,k);
11        u(iface) = 6*um*((yf(iface)/H)-(yf(iface)/H)^2);
12        G(ic,k) = rho*cp*u(iface)*area(iface)*nx(ic,k);
13    end
14 end

```

Téngase en cuenta que se están tomando los mismos datos de  $u_m$ ,  $\rho$  y  $c_p$  que los que se usaron en el problema descrito en la Sección 3.2, así como la expresión (3.25) para el cómputo de la velocidad. Una vez calculado el gasto másico, la siguiente y última tarea consiste en expresar el valor de  $\phi_f$  en términos de los valores en los nodos. La Sección 2.2 estuvo dedicada específicamente a presentar y desarrollar algunos de los métodos que comúnmente se utilizan en la industria para llevar a cabo esta tarea, propia del término advectivo. A modo ilustrativo, en esta sección se discutirán únicamente el método de las diferencias centradas y el esquema *upwind*, dos de los más utilizados y a la vez más sencillos de implementar, sin descuidar la precisión de los resultados.

En el método de las diferencias centradas, el valor de  $\phi_f$  se obtiene ponderando los valores de  $\phi$  en los nodos de las celdas adyacentes a la cara con la función de interpolación  $w_f$ , cuyo cálculo ya se presentó en la Sección 4.1.1. Así, se verifica la siguiente igualdad:

$$\sum_{f=1}^{N_{f,P}} G_f \phi_f = \sum_{f=1}^{N_{f,P}} G_f [w_f \phi_P + (1 - w_f) \phi_{nb(f)}] \quad (4.31)$$

Las ventajas y desventajas de este método ya han sido descritas en numerosas ocasiones. Por un lado, el error de truncamiento decae cuadráticamente con la distancia entre el nodo y la cara, lo cual es un aspecto positivo; por otro lado, no se tiene en cuenta que el proceso de convección transporta la información aguas abajo acentuando la influencia en uno de los nodos según la dirección del flujo,

lo cual puede llevar a resultados incorrectos u oscilaciones alrededor de la solución. No obstante, si la convección en el problema objeto de estudio no es lo suficientemente fuerte como para penalizar este método frente al *upwind*, ambos serán buenas opciones a elegir.

En el esquema *upwind*,  $\phi_f$  tomará el valor de una de las celdas colindantes según la dirección del flujo. Una forma de expresar  $\phi_f$  es como una función a trozos:

$$\phi_f = \begin{cases} \phi_P & \text{si } G_f > 0 \\ \phi_{nb} & \text{si } G_f < 0 \end{cases} \quad (4.32)$$

Sin embargo, esta opción obliga al código a realizar numerosas comprobaciones *IF* para discernir si se está en un caso o en otro, lo cual es, como ya se sabe, una operación computacionalmente costosa. Una alternativa es definir la función de forma que la lógica anteriormente descrita se cumpla automáticamente. Esto se consigue de la siguiente forma:

$$\sum_{f=1}^{N_{f,P}} G_f \phi_f = \sum_{f=1}^{N_{f,P}} \left[ \frac{G_f + |G_f|}{2} \phi_P + \frac{G_f - |G_f|}{2} \phi_{nb(f)} \right] \quad (4.33)$$

Nuevamente, este método tiene sus pros y sus contras. Mientras que la física que subyace al método logra ser más realista y su implementación es tremendamente sencilla, la precisión con la que este esquema resuelve el problema es numéricamente peor. La razón de ello es que el error de truncamiento cae con la distancia entre el nodo y la cara, y no con su cuadrado como ocurría en el método de las diferencias centradas. Además, puede llevar a problemas de falsa difusión, como ya se ha comentado en anteriores ocasiones. No obstante, este método resulta perfectamente válido para el caso que nos ocupa, como se verá inmediatamente.

Por último, si a la ecuación (4.19) se le añade el término convectivo, se llega a la siguiente ecuación general para problemas convectivo-difusivos estacionarios en mallas no estructuradas:

$$\begin{aligned} \sum_{f=1}^{N_{f,P}} \left[ \frac{G_f + |G_f|}{2} \phi_P + \frac{G_f - |G_f|}{2} \phi_{nb(f)} \right] = \\ \sum_{f=1}^{N_{f,P}} \Gamma_f \left[ \frac{\phi_{nb(f)} - \phi_P}{\delta_f} - \left( \frac{\phi_{a(f)} - \phi_{b(f)}}{\delta_f A_f} \right) \vec{t}_f \cdot \vec{l}_f \right] A_f + \bar{S}_\phi \Delta V \end{aligned} \quad (4.34)$$

El ensamblaje del sistema es muy similar al que se expuso en (4.24). No obstante, el valor de los coeficientes necesita ser modificado para incluir el término relativo a la convección. Así, la forma final de los coeficientes de la matriz será:

$$A_{p,p} = \sum_{f=1}^{N_{f,P}} \left[ \frac{\Gamma_f A_f}{\delta_f} + \frac{G_f + |G_f|}{2} \right] \quad (4.35)$$

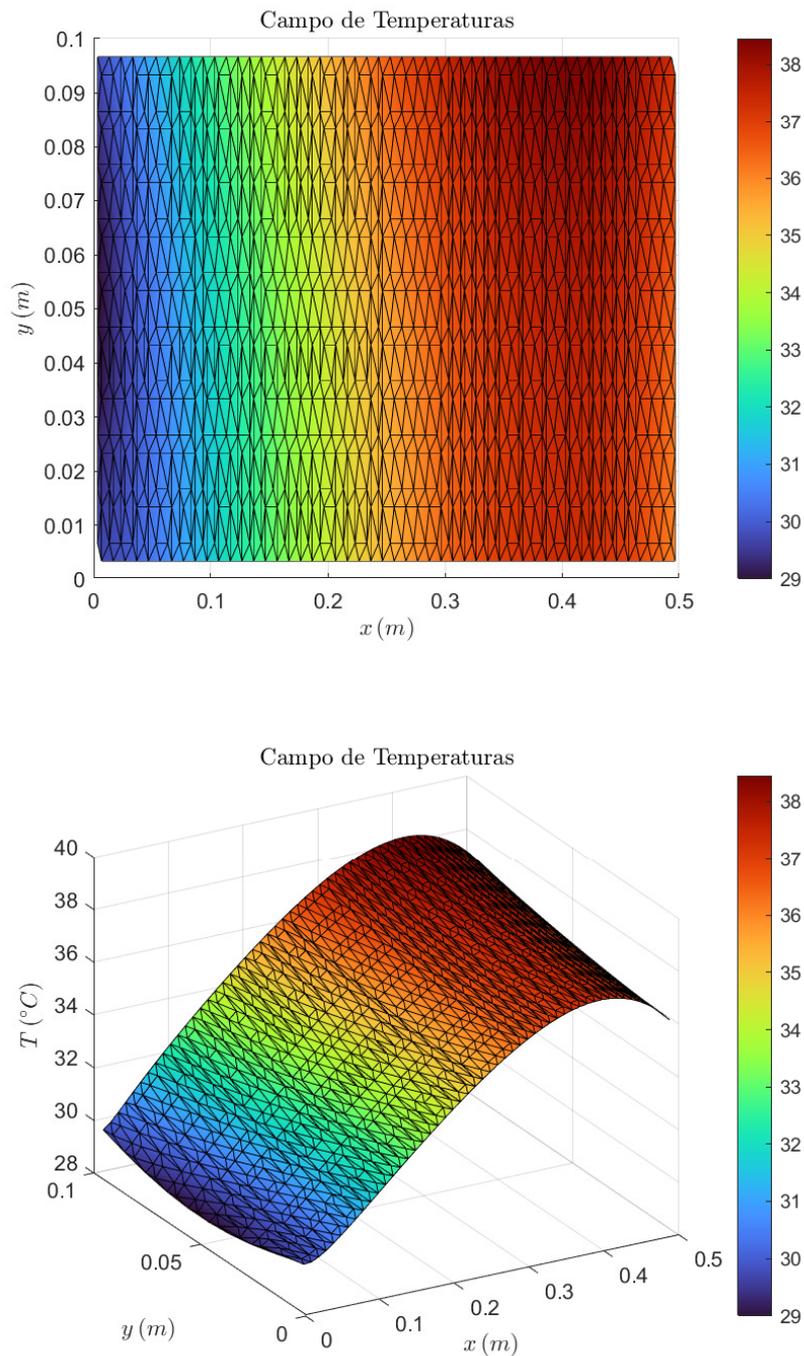
$$A_{k,p} = -\frac{\Gamma_{k,p} A_{k,p}}{\delta_{k,p}} + \frac{G_{k,p} - |G_{k,p}|}{2} \quad (4.36)$$

$$A_{skew,p} = -\sum_{f=1}^{N_{f,P}} \Gamma_f \left( \frac{\phi_{a(f)}^0 - \phi_{b(f)}^0}{\delta_f} \right) \vec{t}_f \cdot \vec{l}_f \quad (4.37)$$

Es interesante notar como el término  $A_{skew,p}$  permanece inalterado. Esto es porque el proceso de convección no lleva asociado un flujo tangencial; éste es único del proceso de difusión. El motivo

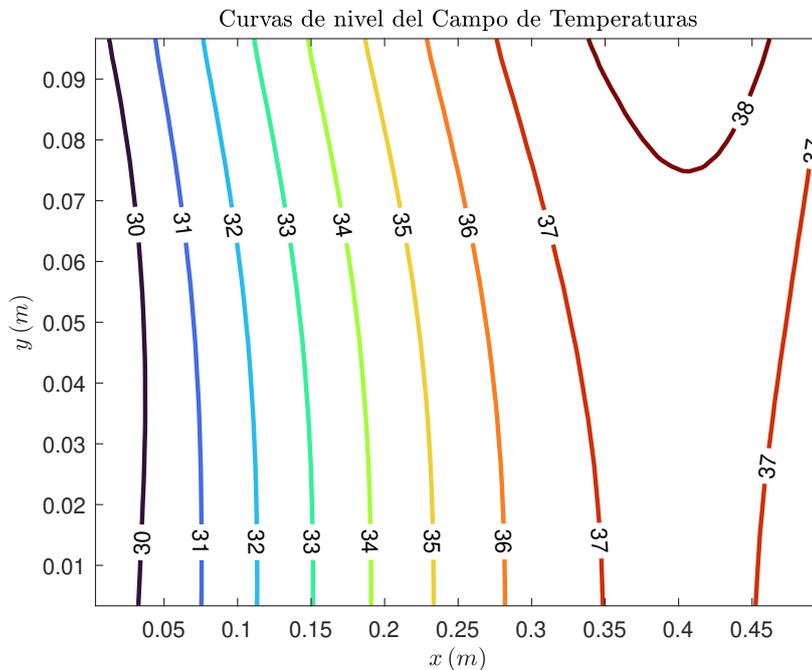
de esto radica en que el flujo difusivo depende del gradiente de  $\phi$ , mientras que el flujo convectivo no lo hace.

Llegados a este punto, se está en condiciones de resolver un problema de convección-difusión empleando esta formulación. Para validar los resultados, se resolverá el mismo problema presentado en la Sección 3.2 y se compararán las soluciones. Puede encontrarse una representación esquemática de este problema en la Figura 3.7.



**Figura 4.13** Valores del campo de temperaturas en los nodos para un mallado no estructurado  $M \times N$  con  $M = 50$  y  $N = 10$  utilizando el esquema *upwind*.

De forma equivalente al caso puramente difusivo, se añade a continuación una representación de las curvas de nivel como herramienta para validar los resultados. Igual que se comentó en el caso de malla estructurada, esta ilustración muestra lo que era de esperar: las isoterms entran perpendicularmente en las caras donde el flujo difusivo es nulo, y presentan cierta curvatura cóncava con respecto al flujo incidente como manifestación del proceso de convección. Además, la temperatura en la pared oeste roza los  $30^{\circ}\text{C}$ , cumpliendo con la condición de contorno.



**Figura 4.14** Curvas de nivel (isoterms) del campo de temperaturas para un mallado no estructurado  $M \times N$  con  $M = 50$  y  $N = 10$ .

El código ejecutado para obtener estos resultados es prácticamente idéntico al caso anterior. Como el pre-procesamiento es el mismo (añadiendo el cálculo del gasto  $G_f$ , que ya ha sido detallado), se presentan a continuación únicamente las líneas que difieren con respecto a la situación puramente difusiva. Destacar que los cálculos se han hecho empleando el esquema *upwind* para aproximar el término convectivo.

```

1  %% CAMBIOS PARA INTRODUCCION DE TERMINO CONVECTIVO
2
3  % Coeficientes de la Matriz A
4
5  for ic = 1:n_c
6      for k = 1:3
7          iface = Lcf(ic,k);
8          if bface(iface) == 0
9              A0(ic) = A0(ic) + gammaf(iface)*area(iface)/delta(iface) + ...
10                 (G(ic,k) + abs(G(ic,k)))/2;
11              Anb(ic,k) = -gammaf(iface)*area(iface)/delta(iface) + ...
12                 (G(ic,k) - abs(G(ic,k)))/2;
13              ic1 = Lfc(iface,1);
14              ic2 = Lfc(iface,2);
15              if ic1 == ic
16                  A(ic,ic2) = Anb(ic,k);
17              else

```

```

18     A(ic,ic1) = Anb(ic,k);
19     end
20     elseif bface(iface) == 1 && phib(f_to_bf(iface)) ~= 0
21         A0(ic) = A0(ic) + gammaf(iface)*area(iface)/delta(iface) + ...
22             (G(ic,k) + abs(G(ic,k)))/2;
23         Anb(ic,k) = 0;
24     else
25         A0(ic) = A0(ic) + (G(ic,k) + abs(G(ic,k)))/2;
26         Anb(ic,k) = 0;
27     end
28     A(ic,ic) = A0(ic);
29 end
30 end
31
32 for ibf = 1:n_f_b
33     iface = bf_to_f(ibf);
34     ic1 = Lfc(iface,1);
35     b(ic1) = b(ic1) + (gammaf(iface)*area(iface)/delta(iface) + ...
36         (G(iface) - abs(G(iface)))/2)*phib(ibf) + area(iface)*Jb(ibf);
37 end

```

De nuevo, para aumentar la aplicabilidad del método y comparar los resultados directamente con la literatura se estudia a continuación uno de los problemas presentados en [6]. En esencia es similar al presentado en la Figura 3.7, pero con las condiciones de contorno modificadas y con un flujo convectivo caracterizado por la expresión (3.25), siendo  $Pe = \rho u_m H / \Gamma = 200$ , lo cual implica que  $u_m = 2.92 \cdot 10^{-7} m/s$ . El esquema de este problema se representa en la siguiente figura:

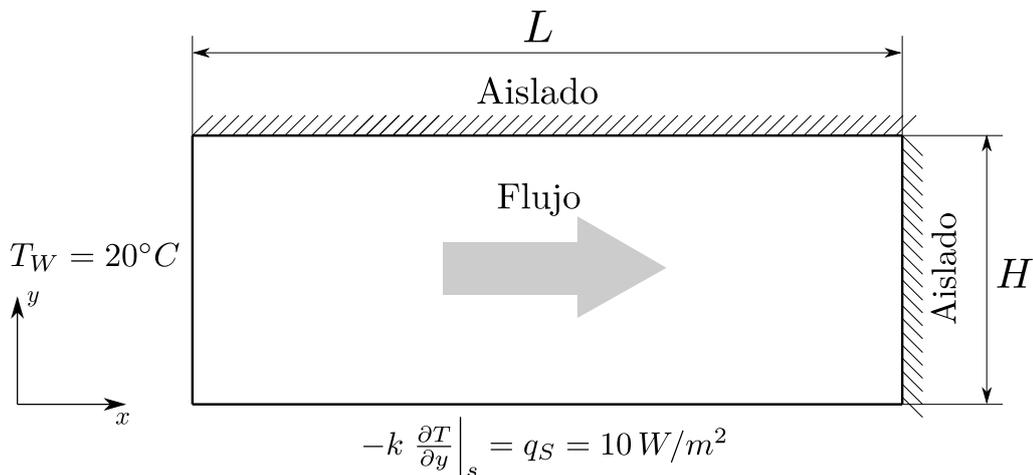
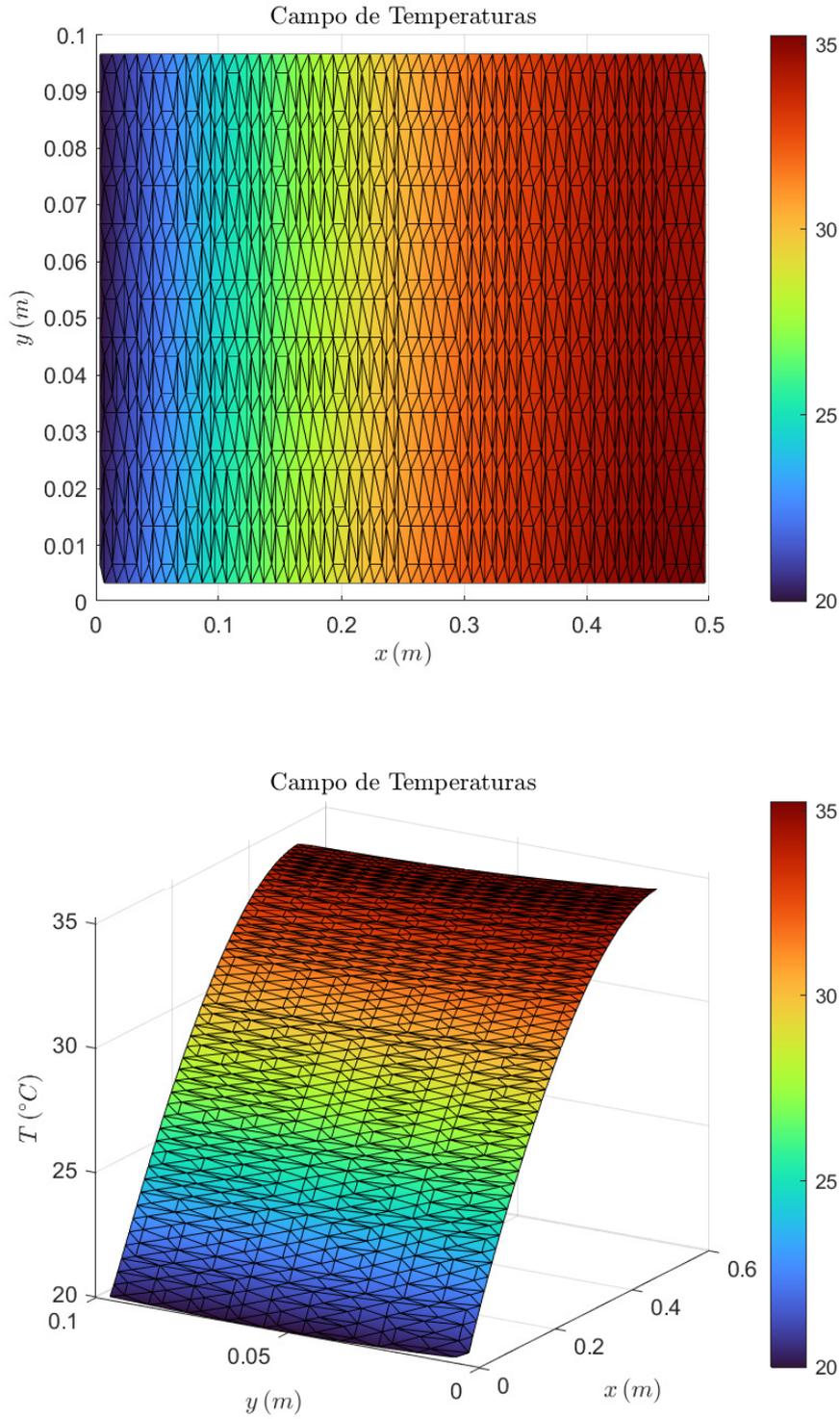
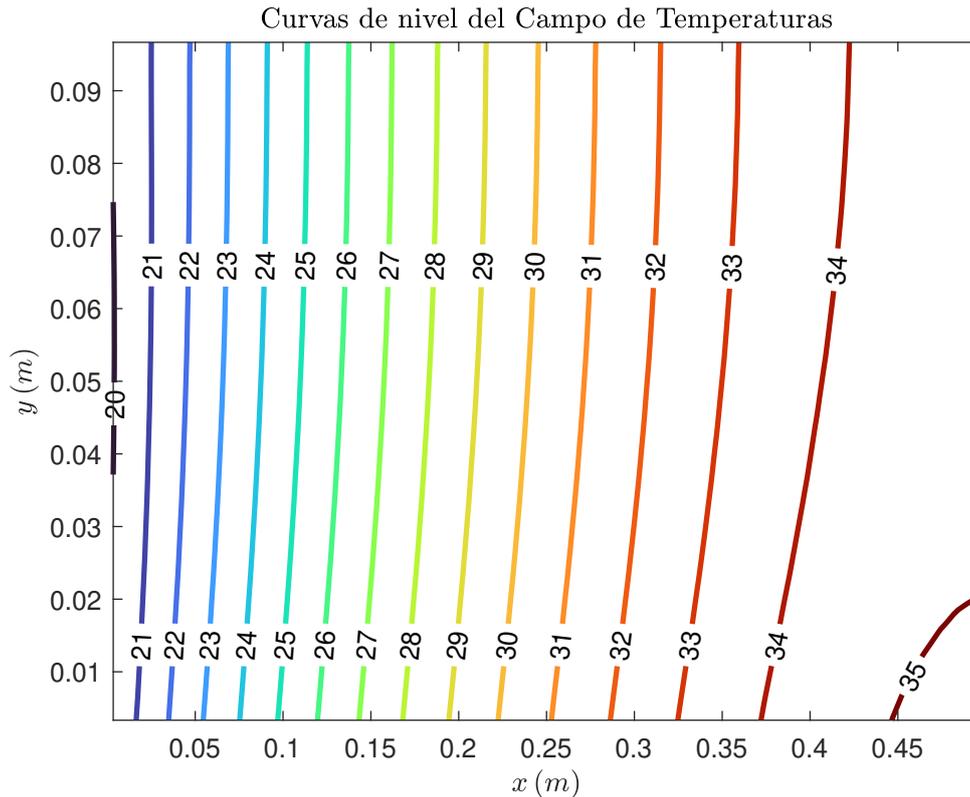


Figura 4.15 Representación del problema descrito en [6].

Resolviendo el problema con la formulación no estructurada del FVM se obtiene la siguiente solución. Nuevamente puede observarse la concordancia entre los resultados obtenidos y los que se reportan en la literatura, poniendo de manifiesto que la modelización del término convectivo en la formulación desarrollada es totalmente consistente.



**Figura 4.16** Valores del campo de temperaturas en los nodos para un mallado no estructurado  $M \times N$  con  $M = 50$  y  $N = 10$  utilizando el esquema *upwind* en el problema analizado en [6].



**Figura 4.17** Curvas de nivel (isotermas) del campo de temperaturas para un mallado no estructurado  $M \times N$  con  $M = 50$  y  $N = 10$  en el problema analizado en [6].

Llegados a este punto, puede concluirse que la formulación del FVM para el estudio de problemas de convección-difusión térmica en geometrías bidimensionales no estructuradas ha originado resultados satisfactorios en todos los casos estudiados. Esta afirmación se refuerza comparando dichos resultados con algunos de los problemas que se analizan en la literatura, especialmente en [4] y [6], que además de servir de gran ayuda y referencia para la elaboración del proyecto son la prueba perfecta para la validación de los códigos propios. Queda así probada su autenticidad y versatilidad para resolver problemas de distinta índole obteniendo resultados muy precisos en poco tiempo.

La relajación de la hipótesis de estacionariedad para estudiar un problema con el término de variación temporal no nulo no se contempla en este capítulo. Los esquemas de discretización expuestos en la Sección 2.3 son igualmente válidos para el caso 2D, y su implementación es fundamentalmente idéntica.

Se dedica un último capítulo a la recapitulación de todo el trabajo desarrollado a lo largo de la memoria, así como un listado de posibles estudios futuros que complementarían en gran medida al contenido expuesto en este proyecto.

Por último, anexo a la memoria, puede encontrarse un Apéndice con algunos de los códigos propios cuya ejecución proporciona ciertos resultados que se han ido exponiendo a lo largo del texto. Puesto que los más relevantes se han integrado en la memoria a medida que se ha ido desarrollando, aquí se incluirán los menos representativos, con la única finalidad de que el lector pueda acudir a ellos con facilidad para una posible necesidad de uso individual. Para un mayor detalle y si el lector estuviera interesado, pueden consultarse adicionalmente las obras de Patankar ([1]), Date ([3]) y

Voller ([5]).



## 5 Conclusión y trabajos futuros

---

A lo largo de la presente memoria se ha profundizado en el Método de los Volúmenes Finitos, una práctica de resolución numérica para problemas fluido-dinámicos generales y cuyo extendido uso dentro de la industria del CFD está intrínsecamente relacionado con la singular característica que lo diferencia del resto: su naturaleza conservativa. Así, se ha acometido de forma satisfactoria el análisis exhaustivo de este método aplicado a problemas convectivo-difusivos unidimensionales y bidimensionales, objetivo principal de este proyecto.

Para ello, y tras contextualizar el trabajo dentro del marco de la industria del CFD, se han establecido los fundamentos del FVM de una manera escalonada en complejidad, comenzando por la aplicación del método a problemas unidimensionales, estacionarios y de difusión pura y finalizando con problemas bidimensionales, convectivo-difusivos y no estacionarios en mallas no estructuradas.

A lo largo de dicho desarrollo se ha ahondado en la discretización de las ecuaciones y la modelización de los términos difusivos y advectivos, resolviendo en el caso de éstos últimos los problemas de estabilidad asociados a la propia naturaleza de la convección, cuya influencia se arrastra aguas abajo del flujo. Así aparecen lo que se conoce como métodos *upwind*, que captan correctamente esta influencia recuperando la información aguas arriba, lo que da sentido a su nombre.

Los resultados proporcionados por estos esquemas de discretización, aplicados a problemas generales que usualmente se encuentran en la práctica, han sido comparados con aquéllos reportados por la literatura, confirmando validez a los mismos. Así, se ha evidenciado tener una sencilla pero potente herramienta para la resolución de las ecuaciones de flujo y el cálculo de las variables de interés en problemas donde convección y difusión actúan simultáneamente, situación que aparece con suma frecuencia en la naturaleza.

Especial interés cobra el capítulo dedicado a mallas no estructuradas, donde el pre-procesamiento realizado para cálculo y almacenamiento de la información geométrica necesaria en la resolución del problema ha resultado ser tremendamente eficaz. La aplicación de esta formulación a problemas de distinta índole ha constituido una visión primitiva de algunas de las grandes ventajas de este tipo de mallas, poniendo de manifiesto su popular uso dentro de la industria para resolver geometrías de complejidad ilimitada.

No obstante, hay que tener en consideración que la herramienta utilizada no está exenta de los errores que conlleva toda modelización de un sistema real, así como errores numéricos asociados a la implementación del modelo en un programa de cómputo numérico (MATLAB), en el que la resolución de los sistemas de ecuaciones no está exenta de las limitaciones propias de un programa

de cálculo numérico, como son la propagación de errores o el tiempo computacional.

Adicionalmente, se listan a continuación una serie de potenciales trabajos futuros que podrían extender el alcance de este proyecto:

- Extensión de los métodos descritos a **problemas tridimensionales**. Generación de mallas hexaédricas estructuradas y tetraédricas no estructuradas, formulación de las ecuaciones del problema, resolución y valoración de los resultados.
- Análisis y **valoración de problemas convectivo-difusivos no estacionarios** de carácter general en geometrías unidimensionales y bidimensionales.
- Análisis del FVM en **geometrías axilsimétricas**. Formulación del problema convectivo-difusivo en coordenadas cilíndricas, discretización y resolución.
- Estudio de problemas bidimensionales con **mallas curvilíneas**. Formulación de las ecuaciones en las variables  $\xi$  y  $\eta$  y cálculo de la solución.
- **Cómputo del campo de velocidades** ( $\vec{v}$ ) como paso previo a la resolución del problema. Puede calcularse de forma autocontenida a través de la ecuación de cantidad de movimiento con las presiones y velocidades (se conoce como método SIMPLE) o mediante la Teoría Potencial resolviendo  $\nabla^2\psi = 0$ , donde  $u = \partial\psi/\partial x$  y  $v = \partial\psi/\partial y$ .
- Comparación del FVM con otros métodos similares en relación a determinados problemas convectivo-difusivos. Especial interés cobraría la comparación con el **CVFEM**, variante del método de los elementos finitos basado en la naturaleza conservativa de los métodos de volúmenes finitos.
- Proceso de optimización de mallas. **Triangulación de Delaunay**. Permite obtener mallas triangulares en las que se maximiza el número de triángulos equiláteros, evitando algunas propiedades no deseadas que presentan los triángulos excesivamente esbeltos.
- Implementación de los problemas analizados en algún software CFD usado actualmente en la industria (Ansys Fluent, OpenFOAM, STAR-CCM+, etc.). Comparación de la solución obtenida con los resultados extraídos de MATLAB.

# Apéndice A

## Códigos de MATLAB

---

A lo largo del documento se han introducido ciertas líneas de código que pretendían servir como elemento de ayuda en la exposición del tema en cuestión. La intención es que dichos códigos tuvieran una estructura especialmente clara y compacta para mantener la continuidad de la presentación. Sin embargo, muchos de los resultados que se muestran no aparecen respaldados por ningún código.

Por ello, en este apéndice se adjuntan otros códigos MATLAB desarrollados por el autor que también resuelven ciertos problemas térmicos presentados a lo largo de la memoria. En particular, aquí se incluyen aquellos menos representativos, bien porque resultan redundantes o bien porque la claridad de su exposición puede no resultar tan representativa. Así, el lector dispone de una herramienta para verificar que los resultados que en este proyecto se presentan son acordes a los obtenidos. Además, pueden ser un buen punto de partida para lectores interesados en profundizar más en el tema principal de este proyecto.

### A.1 Convección-difusión estacionaria 1D

```
1 %% COMPARACION DE TODOS LOS METODOS
2
3 clear all; close all; clc
4
5 % MALLADO
6
7 L = 1; % Longitud Barra (m)
8 N = 20; % Número de Nodos
9
10 caras = linspace(0,L,N+1); % Nodos Equiespaciados
11
12 % caras_theta = linspace(0,pi,N+1); % Nodos Concentrados en Extremos
13 % caras = L/2*(1 - cos(caras_theta));
14
15 % rng = [0,L*rand(1,N-1),L]; % Nodos Aleatorios
16 % caras = sort(rng);
17
18 for i = 2:length(caras)
19     nodos(i-1) = (caras(i)-caras(i-1))/2 + caras(i-1);
20 end
21
22 % DATOS
23
24 Gamma_w = 0.1*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
```

```

25 Gamma_e = 0.1*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
26 Aw = 1*ones(N,1);      % Área Transversal (m^2) -> CTE
27 Ae = 1*ones(N,1);      % Área Transversal (m^2) -> CTE
28 nw = -ones(N,1);       % Vector Normal (nw=-1 cte para caras west)
29 ne = ones(N,1);        % Vector Normal (ne=1 cte para caras east)
30 rho = 1*ones(N,1);     % Densidad del Fluido (kg/m^3)
31 u = 3*ones(N,1);       % Velocidad Unidireccional (m/s)
32 Fw = rho.*Aw.*nw.*u;   % Flujo de Fluido West
33 Fe = rho.*Ae.*ne.*u;   % Flujo de Fluido East
34
35 %%% DIFERENCIAS CENTRADAS %%%
36
37 % COEFICIENTES
38
39 for I = 2:N-1
40     gw(I) = (caras(I)-nodos(I-1))/(nodos(I)-nodos(I-1)); % Factor Interpolación
41     ge(I) = (caras(I+1)-nodos(I+1))/(nodos(I)-nodos(I+1)); % Factor Interpolación
42     Dw(I) = Gamma_w(I)*nw(I)*Aw(I)/(nodos(I)-nodos(I-1));
43     De(I) = Gamma_e(I)*ne(I)*Ae(I)/(nodos(I+1)-nodos(I));
44     aW(I) = Fw(I)*(1-gw(I))+Dw(I); % Coeficiente phi_W
45     aP(I) = Fw(I)*gw(I)+Fe(I)*ge(I)-Dw(I)+De(I); % Coeficiente phi_P
46     aE(I) = Fe(I)*(1-ge(I))-De(I); % Coeficiente phi_E
47     aSP(I) = 0; % Término Asociado a la Generación
48     A(I,I-1) = aW(I);
49     A(I,I) = aP(I);
50     A(I,I+1) = aE(I);
51     b(I) = aSP(I);
52 end
53
54 % CONDICIONES DE CONTORNO
55
56 % LEFT
57
58 phi_0 = 1; % CdC Izquierda: phi impuesto (K);
59 gw(1) = 0;
60 ge(1) = (caras(2)-nodos(2))/(nodos(1)-nodos(2));
61 Dw(1) = Gamma_w(1)*nw(1)*Aw(1)/(nodos(1)-caras(1));
62 De(1) = Gamma_e(1)*ne(1)*Ae(1)/(nodos(2)-nodos(1));
63 aW(1) = Fw(1)*(1-gw(1))+Dw(1);
64 aP(1) = Fw(1)*gw(1)+Fe(1)*ge(1)-Dw(1)+De(1);
65 aE(1) = Fe(1)*(1-ge(1))-De(1);
66 aSP(1) = 0;
67 A(1,1) = aP(1);
68 A(1,2) = aE(1);
69 b(1) = aSP(1)-aW(1)*phi_0;
70
71 % RIGHT
72
73 phi_L = 0; % CdC Izquierda: phi impuesto (K);
74 gw(N) = (caras(N)-nodos(N-1))/(nodos(N)-nodos(N-1));
75 ge(N) = 0;
76 Dw(N) = Gamma_w(N)*nw(N)*Aw(N)/(nodos(N)-nodos(N-1));
77 De(N) = Gamma_e(N)*ne(N)*Ae(N)/(caras(end)-nodos(N));
78 aW(N) = Fw(N)*(1-gw(N))+Dw(N);
79 aP(N) = Fw(N)*gw(N)+Fe(N)*ge(N)-Dw(N)+De(N);
80 aE(N) = Fe(N)*(1-ge(N))-De(N);
81 aSP(N) = 0;
82 A(N,N-1) = aW(N);
83 A(N,N) = aP(N);
84 b(N) = aSP(N)-aE(N)*phi_L;
85
86 % SOLUCION

```

```

87
88 phi = A\b';
89 plot([0,nodos,L],[phi_0;phi;phi_L],'o-', 'MarkerIndices',2:length(nodos)+1, 'linewidth'
90 ,1.2)
91 hold on
92
93 clearvars -except L N caras nodos Gamma_w Gamma_e Aw Ae nw ne rho u Fw Fe
94
95 %%% FIRST ORDER UPWIND %%%
96
97 % COEFICIENTES
98
99 for I = 2:N-1
100     gw(I) = 1-heaviside(u(I)); % Método Upwind
101     ge(I) = heaviside(u(I)); % Método Upwind
102     Dw(I) = Gamma_w(I)*nw(I)*Aw(I)/(nodos(I)-nodos(I-1));
103     De(I) = Gamma_e(I)*ne(I)*Ae(I)/(nodos(I+1)-nodos(I));
104     aW(I) = Fw(I)*(1-gw(I))+Dw(I); % Coeficiente phi_W
105     aP(I) = Fw(I)*gw(I)+Fe(I)*ge(I)-Dw(I)+De(I); % Coeficiente phi_P
106     aE(I) = Fe(I)*(1-ge(I))-De(I); % Coeficiente phi_E
107     aSP(I) = 0; % Término Asociado a la Generación
108     A(I,I-1) = aW(I);
109     A(I,I) = aP(I);
110     A(I,I+1) = aE(I);
111     b(I) = aSP(I);
112 end
113
114 % CONDICIONES DE CONTORNO
115
116 % LEFT
117
118 phi_0 = 1; % CdC Izquierda: phi impuesto (K)
119 gw(1) = 1-heaviside(u(1));
120 ge(1) = heaviside(u(1));
121 Dw(1) = Gamma_w(1)*nw(1)*Aw(1)/(nodos(1)-caras(1));
122 De(1) = Gamma_e(1)*ne(1)*Ae(1)/(nodos(2)-nodos(1));
123 aW(1) = Fw(1)*(1-gw(1))+Dw(1);
124 aP(1) = Fw(1)*gw(1)+Fe(1)*ge(1)-Dw(1)+De(1);
125 aE(1) = Fe(1)*(1-ge(1))-De(1);
126 aSP(1) = 0;
127 A(1,1) = aP(1);
128 A(1,2) = aE(1);
129 b(1) = aSP(1)-aW(1)*phi_0;
130
131 % RIGHT
132
133 phi_L = 0; % CdC Derecha: phi impuesto (K)
134 gw(N) = 1-heaviside(u(N));
135 ge(N) = heaviside(u(N));
136 Dw(N) = Gamma_w(N)*nw(N)*Aw(N)/(nodos(N)-nodos(N-1));
137 De(N) = Gamma_e(N)*ne(N)*Ae(N)/(caras(end)-nodos(N));
138 aW(N) = Fw(N)*(1-gw(N))+Dw(N);
139 aP(N) = Fw(N)*gw(N)+Fe(N)*ge(N)-Dw(N)+De(N);
140 aE(N) = Fe(N)*(1-ge(N))-De(N);
141 aSP(N) = 0;
142 A(N,N-1) = aW(N);
143 A(N,N) = aP(N);
144 b(N) = aSP(N)-aE(N)*phi_L;
145
146 % SOLUCION
147 phi = A\b';

```

```

148 plot([0,nodos,L],[phi_0;phi;phi_L],'o-','MarkerIndices',2:length(nodos)+1,'linewidth'
      ,1.2)
149 hold on
150
151 clearvars -except L N caras nodos Gamma_w Gamma_e Aw Ae nw ne rho u Fw Fe
152
153 %% SECOND ORDER UPWIND %%
154
155 % COEFICIENTES
156
157 for I = 3:N-2
158     gwww(I) = -1/2*heaviside(u(I));
159     gww(I) = 3/2*heaviside(u(I));
160     gew(I) = -1/2*heaviside(u(I));
161     gep(I) = 3/2*heaviside(u(I));
162     gwp(I) = 3/2*(1-heaviside(u(I)));
163     gwe(I) = -1/2*(1-heaviside(u(I)));
164     gee(I) = 3/2*(1-heaviside(u(I)));
165     geee(I) = -1/2*(1-heaviside(u(I)));
166     Dw(I) = Gamma_w(I)*nw(I)*Aw(I)/(nodos(I)-nodos(I-1));
167     De(I) = Gamma_e(I)*ne(I)*Ae(I)/(nodos(I+1)-nodos(I));
168     aWW(I) = Fw(I)*gwww(I); % Coeficiente phi_WW
169     aW(I) = Fw(I)*gww(I)+Fe(I)*gew(I)+Dw(I); % Coeficiente phi_W
170     aP(I) = Fw(I)*gwp(I)+Fe(I)*gep(I)-Dw(I)+De(I); % Coeficiente phi_P
171     aE(I) = Fe(I)*gee(I)+Fw(I)*gwe(I)-De(I); % Coeficiente phi_E
172     aEE(I) = Fe(I)*geee(I); % Coeficiente phi_EE
173     aSP(I) = 0; % Término Asociado a la Generación
174     A(I,I-2) = aWW(I);
175     A(I,I-1) = aW(I);
176     A(I,I) = aP(I);
177     A(I,I+1) = aE(I);
178     A(I,I+2) = aEE(I);
179     b(I) = aSP(I);
180 end
181
182 % CONDICIONES DE CONTORNO
183
184 % LEFT
185
186 phi_0 = 1; % CdC Izquierda: phi impuesto (K)
187
188 gwww(1) = 0;
189 gww(1) = 1;
190 gew(1) = -1;
191 gep(1) = 2;
192 gwp(1) = 0;
193 gwe(1) = 0;
194 gee(1) = 0;
195 geee(1) = 0;
196 Dw(1) = Gamma_w(1)*nw(1)*Aw(1)/(nodos(1)-caras(1));
197 De(1) = Gamma_e(1)*ne(1)*Ae(1)/(nodos(2)-nodos(1));
198 aWW(1) = Fw(1)*gwww(1); % Coeficiente phi_WW
199 aW(1) = Fw(1)*gww(1)+Fe(1)*gew(1)+Dw(1); % Coeficiente phi_W
200 aP(1) = Fw(1)*gwp(1)+Fe(1)*gep(1)-Dw(1)+De(1); % Coeficiente phi_P
201 aE(1) = Fe(1)*gee(1)+Fw(1)*gwe(1)-De(1); % Coeficiente phi_E
202 aEE(1) = Fe(1)*geee(1); % Coeficiente phi_EE
203 aSP(1) = 0; % Término Asociado a la Generación
204 A(1,1) = aP(1);
205 A(1,2) = aE(1);
206 A(1,3) = aEE(1);
207 b(1) = aSP(1)-aW(1)*phi_0;
208

```

```

209 gwww(2) = -1;
210 gww(2) = 2;
211 gew(2) = -1/2;
212 gep(2) = 3/2;
213 gwp(2) = 0;
214 gwe(2) = 0;
215 gee(2) = 0;
216 geee(2) = 0;
217 Dw(2) = Gamma_w(2)*nw(2)*Aw(2)/(nodos(2)-nodos(1));
218 De(2) = Gamma_e(2)*ne(2)*Ae(2)/(nodos(3)-nodos(2));
219 aWW(2) = Fw(2)*gwww(2); % Coeficiente phi_WW
220 aW(2) = Fw(2)*gww(2)+Fe(2)*gew(2)+Dw(2); % Coeficiente phi_W
221 aP(2) = Fw(2)*gwp(2)+Fe(2)*gep(2)-Dw(2)+De(2); % Coeficiente phi_P
222 aE(2) = Fe(2)*gee(2)+Fw(2)*gwe(2)-De(2); % Coeficiente phi_E
223 aEE(2) = Fe(2)*geee(2); % Coeficiente phi_EE
224 aSP(2) = 0; % Término Asociado a la Generación
225 A(2,1) = aW(2);
226 A(2,2) = aP(2);
227 A(2,3) = aE(2);
228 A(2,4) = aEE(2);
229 b(2) = aSP(2)-aWW(2)*phi_0;
230
231 % RIGHT
232
233 phi_L = 0; % CdC Derecha: phi impuesto (K)
234
235 gwww(N-1) = -1/2*heaviside(u(N-1));
236 gww(N-1) = 3/2*heaviside(u(N-1));
237 gew(N-1) = -1/2*heaviside(u(N-1));
238 gep(N-1) = 3/2*heaviside(u(N-1));
239 gwp(N-1) = 3/2*(1-heaviside(u(N-1)));
240 gwe(N-1) = -1/2*(1-heaviside(u(N-1)));
241 gee(N-1) = 3/2*(1-heaviside(u(N-1)));
242 geee(N-1) = -1/2*(1-heaviside(u(N-1)));
243 Dw(N-1) = Gamma_w(N-1)*nw(N-1)*Aw(N-1)/(nodos(N-1)-nodos(N-2));
244 De(N-1) = Gamma_e(N-1)*ne(N-1)*Ae(N-1)/(nodos(N)-nodos(N-1));
245 aWW(N-1) = Fw(N-1)*gwww(N-1); % Coeficiente phi_WW
246 aW(N-1) = Fw(N-1)*gww(N-1)+Fe(N-1)*gew(N-1)+Dw(N-1); % Coeficiente phi_W
247 aP(N-1) = Fw(N-1)*gwp(N-1)+Fe(N-1)*gep(N-1)-Dw(N-1)+De(N-1); % Coeficiente phi_P
248 aE(N-1) = Fe(N-1)*gee(N-1)+Fw(N-1)*gwe(N-1)-De(N-1); % Coeficiente phi_E
249 aEE(N-1) = Fe(N-1)*geee(N-1); % Coeficiente phi_EE
250 aSP(N-1) = 0; % Término Asociado a la Generación
251 A(N-1,N-3) = aWW(N-1);
252 A(N-1,N-2) = aW(N-1);
253 A(N-1,N-1) = aP(N-1);
254 A(N-1,N) = aE(N-1);
255 b(N-1) = aSP(N-1);
256
257 gwww(N) = -1/2*heaviside(u(N));
258 gww(N) = 3/2*heaviside(u(N));
259 gew(N) = -1/2*heaviside(u(N));
260 gep(N) = 3/2*heaviside(u(N));
261 gwp(N) = 0;
262 gwe(N) = 0;
263 gee(N) = 0;
264 geee(N) = 0;
265 Dw(N) = Gamma_w(N)*nw(N)*Aw(N)/(nodos(N)-nodos(N-1));
266 De(N) = Gamma_e(N)*ne(N)*Ae(N)/(caras(end)-nodos(N));
267 aWW(N) = Fw(N)*gwww(N); % Coeficiente phi_WW
268 aW(N) = Fw(N)*gww(N)+Fe(N)*gew(N)+Dw(N); % Coeficiente phi_W
269 aP(N) = Fw(N)*gwp(N)+Fe(N)*gep(N)-Dw(N)+De(N); % Coeficiente phi_P
270 aE(N) = Fe(N)*gee(N)+Fw(N)*gwe(N)-De(N); % Coeficiente phi_E

```

```

271     aEE(N) = Fe(N)*geee(N); % Coeficiente phi_EE
272     aSP(N) = 0; % Término Asociado a la Generación
273     A(N,N-2) = aW(N);
274     A(N,N-1) = aW(N);
275     A(N,N) = aP(N);
276     b(N) = aSP(N)-aE(N)*phi_L;
277
278 % SOLUCION
279
280 phi = A\b';
281 plot([0,nodos,L],[phi_0;phi;phi_L],'o-','MarkerIndices',2:length(nodos)+1,'linewidth'
,1.2)
282 hold on
283
284 clearvars -except L N caras nodos Gamma_w Gamma_e Aw Ae nw ne rho u Fw Fe
285
286 %%% POWER-LAW SCHEME %%%
287
288 % COEFICIENTES
289
290 for I = 2:N-1
291     Dw(I) = Gamma_w(I)*nw(I)*Aw(I)/(nodos(I)-nodos(I-1));
292     De(I) = Gamma_e(I)*ne(I)*Ae(I)/(nodos(I+1)-nodos(I));
293     Pe_w(I) = Fw(I)/Dw(I);
294     Pe_e(I) = Fe(I)/De(I);
295     beta_w(I) = (1-0.1*Pe_w(I))^5/Pe_w(I);
296     beta_e(I) = (1-0.1*Pe_e(I))^5/Pe_e(I);
297
298     if abs(Pe_w(I)) < 10
299         gwp(I) = -beta_w(I);
300         gww(I) = 1+beta_w(I);
301     elseif Pe_w(I) >= 10
302         gwp(I) = 0-1/Pe_w(I);
303         gww(I) = 1+1/Pe_w(I);
304     else
305         gwp(I) = 1-1/Pe_w(I);
306         gww(I) = 0+1/Pe_w(I);
307     end
308
309     if abs(Pe_e(I)) < 10
310         gep(I) = 1+beta_e(I);
311         gee(I) = -beta_e(I);
312     elseif Pe_e(I) >= 10
313         gep(I) = 1+1/Pe_e(I);
314         gee(I) = 0-1/Pe_e(I);
315     else
316         gep(I) = 0+1/Pe_e(I);
317         gee(I) = 1-1/Pe_e(I);
318     end
319
320     aW(I) = Fw(I)*gww(I); % Coeficiente phi_W
321     aP(I) = Fw(I)*gwp(I)+Fe(I)*gep(I); % Coeficiente phi_P
322     aE(I) = Fe(I)*gee(I); % Coeficiente phi_E
323     aSP(I) = 0; % Término Asociado a la Generación
324     A(I,I-1) = aW(I);
325     A(I,I) = aP(I);
326     A(I,I+1) = aE(I);
327     b(I) = aSP(I);
328 end
329
330 % CONDICIONES DE CONTORNO
331

```

```

332 % LEFT
333
334 phi_0 = 1; % CdC Izquierda: phi impuesto (K)
335 Dw(1) = Gamma_w(1)*nw(1)*Aw(1)/(nodos(1)-caras(1));
336 De(1) = Gamma_e(1)*ne(1)*Ae(1)/(nodos(2)-nodos(1));
337 Pe_w(1) = Fw(1)/Dw(1);
338 Pe_e(1) = Fe(1)/De(1);
339 beta_w(1) = (1-0.1*Pe_w(1))^5/Pe_w(1);
340 beta_e(1) = (1-0.1*Pe_e(1))^5/Pe_e(1);
341
342 if abs(Pe_w(1)) < 10
343     gwp(1) = -beta_w(1);
344     gww(1) = 1+beta_w(1);
345 elseif Pe_w(1) >= 10
346     gwp(1) = 0-1/Pe_w(1);
347     gww(1) = 1+1/Pe_w(1);
348 else
349     gwp(1) = 1-1/Pe_w(1);
350     gww(1) = 0+1/Pe_w(1);
351 end
352
353 if abs(Pe_e(1)) < 10
354     gep(1) = 1+beta_e(1);
355     gee(1) = -beta_e(1);
356 elseif Pe_e(1) >= 10
357     gep(1) = 1+1/Pe_e(1);
358     gee(1) = 0-1/Pe_e(1);
359 else
360     gep(1) = 0+1/Pe_e(1);
361     gee(1) = 1-1/Pe_e(1);
362 end
363
364 aW(1) = Fw(1)*gww(1);
365 aP(1) = Fw(1)*gwp(1)+Fe(1)*gep(1);
366 aE(1) = Fe(1)*gee(1);
367 aSP(1) = 0;
368 A(1,1) = aP(1);
369 A(1,2) = aE(1);
370 b(1) = aSP(1)-aW(1)*phi_0;
371
372 % RIGHT
373
374 phi_L = 0; % CdC Derecha: phi impuesto (K)
375 Dw(N) = Gamma_w(N)*nw(N)*Aw(N)/(nodos(N)-nodos(N-1));
376 De(N) = Gamma_e(N)*ne(N)*Ae(N)/(caras(end)-nodos(N));
377 Pe_w(N) = Fw(N)/Dw(N);
378 Pe_e(N) = Fe(N)/De(N);
379 beta_w(N) = (1-0.1*Pe_w(N))^5/Pe_w(N);
380 beta_e(N) = (1-0.1*Pe_e(N))^5/Pe_e(N);
381
382 if abs(Pe_w(N)) < 10
383     gwp(N) = -beta_w(N);
384     gww(N) = 1+beta_w(N);
385 elseif Pe_w(N) >= 10
386     gwp(N) = 0-1/Pe_w(N);
387     gww(N) = 1+1/Pe_w(N);
388 else
389     gwp(N) = 1-1/Pe_w(N);
390     gww(N) = 0+1/Pe_w(N);
391 end
392
393 if abs(Pe_e(N)) < 10

```

```

394     gep(N) = 1+beta_e(N);
395     gee(N) = -beta_e(N);
396     elseif Pe_e(N) >= 10
397         gep(N) = 1+1/Pe_e(N);
398         gee(N) = 0-1/Pe_e(N);
399     else
400         gep(N) = 0+1/Pe_e(N);
401         gee(N) = 1-1/Pe_e(N);
402     end
403
404     aW(N) = Fw(N)*gww(N);
405     aP(N) = Fw(N)*gwp(N)+Fe(N)*gep(N);
406     aE(N) = Fe(N)*gee(N);
407     aSP(N) = 0;
408     A(N,N-1) = aW(N);
409     A(N,N) = aP(N);
410     b(N) = aSP(N)-aE(N)*phi_L;
411
412     % SOLUCION
413
414     phi = A\b';
415     plot([0,nodos,L],[phi_0;phi;phi_L],'o-','MarkerIndices',2:length(nodos)+1,'linewidth'
         ,1.2)
416     hold on
417
418     phiexact = @(x)(phi_0+(phi_L-phi_0)*(exp(rho(1)*u(1)*x/Gamma_e(1))-1)/...
         ((exp(rho(1)*u(1)*L/Gamma_e(1))-1)));
419
420     x = [0:0.002:L];
421     plot(x,phiexact(x),'k','linewidth',1.5)
422     hold on
423     xlabel('$m$', 'Interpreter', 'latex')
424     ylabel('$\phi$', 'Interpreter', 'latex')
425     title('Distribuci\''on de $\phi$', 'Interpreter', 'latex')
426     legend({'Central Differencing Scheme', 'First Order Upwind', 'Second Order Upwind', 'Power-
         Law Scheme', 'Soluci\''on Exacta'}, 'Interpreter', 'latex', ...
         'Location', 'SouthEast')
427

```

## A.2 Difusión no estacionaria 1D

```

1 % PROBLEMA DE DIFUSION (NO ESTACIONARIO, EXPLICIT SCHEME)
2
3 clear all; close all; clc
4
5 TablaErrores = [];
6
7 % MALLADO
8
9 L = 0.02; % Longitud Barra (m)
10 N = 20; % Número de Nodos
11 phi_ini = 200*ones(N,1); % Valores Iniciales
12 phiexact_value = [];
13
14 caras = linspace(0,L,N+1); % Nodos Equiespaciados
15
16 % caras_theta = linspace(0,pi,N+1); % Nodos Concentrados en Extremos
17 % caras = L/2*(1 - cos(caras_theta));
18
19 % rng = [0,L*rand(1,N-1),L]; % Nodos Aleatorios
20 % caras = sort(rng);

```

```

21
22 for i = 2:length(caras)
23     nodos(i-1) = (caras(i)-caras(i-1))/2 + caras(i-1);
24 end
25
26 % DATOS
27
28 Deltat = 0.25*ones(N,1); % Time Step (s)
29 Gamma_w = 10*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
30 Gamma_e = 10*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
31 Aw = 1*ones(N,1); % Área Transversal (m^2) -> CTE
32 Ae = 1*ones(N,1); % Área Transversal (m^2) -> CTE
33 nw = -ones(N,1); % Vector Normal (nw=-1 cte para caras west)
34 ne = ones(N,1); % Vector Normal (ne=1 cte para caras east)
35 rho = 1*ones(N,1); % Densidad del Fluido (kg/m^3)
36 rho_c = 10*10^6*ones(N,1);
37 u = 0*ones(N,1); % Velocidad Unidireccional (m/s)
38 Fw = rho.*Aw.*nw.*u; % Flujo de Fluido West
39 Fe = rho.*Ae.*ne.*u; % Flujo de Fluido East
40
41 % COEFICIENTES
42
43 for t = 0.25:0.25:120
44
45     phiexact_value = [];
46
47     for I = 2:N-1
48         Dw(I) = Gamma_w(I)*nw(I)*Aw(I)/(nodos(I)-nodos(I-1));
49         De(I) = Gamma_e(I)*ne(I)*Ae(I)/(nodos(I+1)-nodos(I));
50         DeltaV(I) = Ae(I)*(caras(I+1)-caras(I));
51         aW(I) = 0; % Coeficiente phi_W
52         aP(I) = rho_c(I)*DeltaV(I)/Deltat(I); % Coeficiente phi_P
53         aE(I) = 0; % Coeficiente phi_E
54         aSP(I) = De(I)*phi_ini(I+1)-Dw(I)*phi_ini(I-1)+(Dw(I)-De(I)+rho_c(I)*DeltaV(I)/
55             Deltat(I))*phi_ini(I);
56         A(I,I-1) = aW(I);
57         A(I,I) = aP(I);
58         A(I,I+1) = aE(I);
59         b(I) = aSP(I);
60     end
61
62 % CONDICIONES DE CONTORNO
63
64 % LEFT
65 % qI = 0; % CdC Izquierda: flujo impuesto (K);
66 Dw(1) = Gamma_w(1)*nw(1)*Aw(1)/(nodos(1)-caras(1));
67 De(1) = Gamma_e(1)*ne(1)*Ae(1)/(nodos(2)-nodos(1));
68 DeltaV(1) = Ae(1)*(caras(2)-caras(1));
69 aW(1) = 0;
70 aP(1) = rho_c(1)*DeltaV(1)/Deltat(1);
71 aE(1) = 0;
72 aSP(1) = De(1)*phi_ini(2)+(-De(1)+rho_c(1)*DeltaV(1)/Deltat(1))*phi_ini(1);
73 A(1,1) = aP(1);
74 A(1,2) = aE(1);
75 b(1) = aSP(1);
76
77 % RIGHT
78
79 phi_R = 0; % CdC Derecha: phi impuesto (K);
80 Dw(N) = Gamma_w(N)*nw(N)*Aw(N)/(nodos(N)-nodos(N-1));
81 De(N) = Gamma_e(N)*ne(N)*Ae(N)/(caras(end)-nodos(N));

```

```

82   DeltaV(N) = Ae(N)*(caras(end)-caras(end-1));
83   aW(N) = 0;
84   aP(N) = rho_c(N)*DeltaV(N)/Deltat(N);
85   aE(N) = 0;
86   aSP(N) = De(N)*phi_R-Dw(N)*phi_ini(N-1)+(Dw(N)-De(N)+rho_c(N)*DeltaV(N)/Deltat(N))*
      phi_ini(N);
87   A(N,N-1) = aW(N);
88   A(N,N) = aP(N);
89   b(N) = aSP(N);
90
91   % SOLUCION
92
93   phi = A\b';
94
95   if t == 40
96     alpha(I) = Gamma_e(I)/rho_c(I);
97     n = 1:20;
98     lambda_n = (2.*n-1)*pi/(2*L);
99     phiexact = @(x) 800./pi.*sum((-1).^(n+1)./(2.*n-1).*exp(-alpha(I).*lambda_n.^2.*t).*cos
      (lambda_n.*x));
100    for x = 0:0.0002:L
101      phiexact_value = [phiexact_value,phiexact(x)];
102    end
103    x = 0:0.0002:L;
104    if t == 40
105      plot(x,phiexact_value,'k','linewidth',1.1,'DisplayName',['Solución exacta (t=',num2str(
      t),')'])
106      hold on
107      plot([caras(1),nodos,caras(end)],[phi(1);phi;phi_R],'--ko','MarkerIndices',2:length(
      nodos)+1,'linewidth',1.2,'DisplayName',['Solución con FVM (t=',num2str(t),')'])
108      hold on
109      elseif t == 80
110      plot(x,phiexact_value,'r','linewidth',1.1,'DisplayName',['Solución exacta (t=',num2str(
      t),')'])
111      hold on
112      plot([caras(1),nodos,caras(end)],[phi(1);phi;phi_R],'--ro','MarkerIndices',2:length(
      nodos)+1,'linewidth',1.2,'DisplayName',['Solución con FVM (t=',num2str(t),')'])
113      hold on
114      else
115      plot(x,phiexact_value,'b','linewidth',1.1,'DisplayName',['Solución exacta (t=',num2str(
      t),')'])
116      hold on
117      plot([caras(1),nodos,caras(end)],[phi(1);phi;phi_R],'--bo','MarkerIndices',2:length(
      nodos)+1,'linewidth',1.2,'DisplayName',['Solución con FVM (t=',num2str(t),')'])
118      hold on
119      end
120
121      for i=1:N
122        TablaErrores=[TablaErrores;i,phi(i),phiexact(nodos(i)),...
123          abs((phi(i)-phiexact(nodos(i)))/phiexact(nodos(i)))*100];
124      end
125
126      X = ['Tabla de errores para t=',num2str(t)];
127      disp(X)
128      disp(' ')
129      disp('      Nodo      Sol. CVM Sol. Exacta  Error(%)')
130      disp(' ')
131      disp(TablaErrores)
132
133      TablaErrores = [];
134
135      end

```

```

136 phi_ini=phi;
137
138 end
139 legend('show','Location','SouthWest')
140 title('Campo de Temperaturas','Interpreter','latex')
141 xlabel('$m$','Interpreter','latex')
142 ylabel('$T(\textcircled{C}$)','Interpreter','latex')

```

---

```

1 % PROBLEMA DE DIFUSION (CD, NO ESTACIONARIO, IMPLICIT SCHEME)
2
3 clear all; close all; clc
4
5 TablaErrores = [];
6
7 % MALLADO
8
9 L = 0.02; % Longitud Barra (m)
10 N = 20; % Número de Nodos
11 phi_ini = 200*ones(N,1); % Valores Iniciales
12 phiexact_value = [];
13
14 caras = linspace(0,L,N+1); % Nodos Equiespaciados
15
16 % caras_theta = linspace(0,pi,N+1); % Nodos Concentrados en Extremos
17 % caras = L/2*(1 - cos(caras_theta));
18
19 % rng = [0,L*rand(1,N-1),L]; % Nodos Aleatorios
20 % caras = sort(rng);
21
22 for i = 2:length(caras)
23     nodos(i-1) = (caras(i)-caras(i-1))/2 + caras(i-1);
24 end
25
26 % DATOS
27
28 Deltat = 2*ones(N,1); % Time Step (s)
29 Gamma_w = 10*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
30 Gamma_e = 10*ones(N,1); % Conductividad Térmica (W/(m*K)) -> CTE
31 Aw = 1*ones(N,1); % Área Transversal (m^2) -> CTE
32 Ae = 1*ones(N,1); % Área Transversal (m^2) -> CTE
33 nw = -ones(N,1); % Vector Normal (nw=-1 cte para caras west)
34 ne = ones(N,1); % Vector Normal (ne=1 cte para caras east)
35 rho = 1*ones(N,1); % Densidad del Fluido (kg/m^3)
36 rho_c = 10*10^6*ones(N,1);
37 u = 0*ones(N,1); % Velocidad Unidireccional (m/s) - CASO 1
38 Fw = rho.*Aw.*nw.*u; % Flujo de Fluido West - CASO 1
39 Fe = rho.*Ae.*ne.*u; % Flujo de Fluido East - CASO 1
40
41 % COEFICIENTES
42
43 for t = 2:2:120
44
45     phiexact_value = [];
46
47 for I = 2:N-1
48     Dw(I) = Gamma_w(I)*nw(I)*Aw(I)/(nodos(I)-nodos(I-1));
49     De(I) = Gamma_e(I)*ne(I)*Ae(I)/(nodos(I+1)-nodos(I));
50     DeltaV(I) = Ae(I)*(caras(I+1)-caras(I));
51     aW(I) = -Dw(I); % Coeficiente phi_W
52     aP(I) = Dw(I)-De(I)-rho_c(I)*DeltaV(I)/Deltat(I); % Coeficiente phi_P
53     aE(I) = De(I); % Coeficiente phi_E

```

```

54     aSP(I) = -rho_c(I)*DeltaV(I)/Deltat(I)*phi_ini(I);
55     A(I,I-1) = aW(I);
56     A(I,I) = aP(I);
57     A(I,I+1) = aE(I);
58     b(I) = aSP(I);
59 end
60
61 % CONDICIONES DE CONTORNO
62
63     % LEFT
64
65     qI = 0; % CdC Izquierda: flujo impuesto (K);
66     Dw(1) = 0;
67     De(1) = Gamma_e(1)*ne(1)*Ae(1)/(nodos(2)-nodos(1));
68     DeltaV(1) = Ae(1)*(caras(2)-caras(1));
69     aW(1) = -Dw(1);
70     aP(1) = Dw(1)-De(1)-rho_c(1)*DeltaV(1)/Deltat(1);
71     aE(1) = De(1);
72     aSP(1) = -rho_c(1)*DeltaV(1)/Deltat(1)*phi_ini(1);
73     A(1,1) = aP(1);
74     A(1,2) = aE(1);
75     b(1) = aSP(1);
76
77     % RIGHT
78
79     phi_R = 0; % CdC Derecha: phi impuesto (K);
80     Dw(N) = Gamma_w(N)*nw(N)*Aw(N)/(nodos(N)-nodos(N-1));
81     De(N) = Gamma_e(N)*ne(N)*Ae(N)/(caras(end)-nodos(N));
82     DeltaV(N) = Ae(N)*(caras(end)-caras(end-1));
83     aW(N) = -Dw(N);
84     aP(N) = Dw(N)-De(N)-rho_c(N)*DeltaV(N)/Deltat(N);
85     aE(N) = De(N);
86     aSP(N) = -rho_c(N)*DeltaV(N)/Deltat(N)*phi_ini(N);
87     A(N,N-1) = aW(N);
88     A(N,N) = aP(N);
89     b(N) = aSP(N)-aE(N)*phi_R;
90
91 % SOLUCION
92
93 phi = A\b';
94
95 if t == 40 || t == 80 || t == 120
96 alpha(I) = Gamma_e(I)/rho_c(I);
97 n = 1:20;
98 lambda_n = (2.*n-1)*pi/(2*L);
99 phiexact = @(x) 800./pi.*sum((-1).^n+1)./(2.*n-1).*exp(-alpha(I).*lambda_n.^2.*t).*cos
    (lambda_n.*x);
100 for x = 0:0.0002:L
101 phiexact_value = [phiexact_value,phiexact(x)];
102 end
103 x = 0:0.0002:L;
104 if t == 40
105 plot(x,phiexact_value,'k','linewidth',1.1,'DisplayName',['Solución exacta (t=',num2str(
    t),')'])
106 hold on
107 plot([caras(1),nodos,caras(end)],[phi(1);phi;phi_R],'--ko','MarkerIndices',2:length(
    nodos)+1,'linewidth',1.2,'DisplayName',['Solución con FVM (t=',num2str(t),')'])
108 hold on
109 elseif t == 80
110 plot(x,phiexact_value,'r','linewidth',1.1,'DisplayName',['Solución exacta (t=',num2str(
    t),')'])
111 hold on

```

```

112 plot([caras(1),nodos,caras(end)],[phi(1);phi;phi_R], '--ro', 'MarkerIndices', 2:length(
    nodos)+1, 'linewidth', 1.2, 'DisplayName', ['Solución con FVM (t=', num2str(t), ')'])
113 hold on
114 else
115 plot(x, phiexact_value, 'b', 'linewidth', 1.1, 'DisplayName', ['Solución exacta (t=', num2str(
    t), ')'])
116 hold on
117 plot([caras(1),nodos,caras(end)],[phi(1);phi;phi_R], '--bo', 'MarkerIndices', 2:length(
    nodos)+1, 'linewidth', 1.2, 'DisplayName', ['Solución con FVM (t=', num2str(t), ')'])
118 hold on
119 end
120
121 for i=1:N
122     TablaErrores=[TablaErrores;i,phi(i),phiexact(nodos(i)),...
123         abs((phi(i)-phiexact(nodos(i)))/phiexact(nodos(i)))*100];
124 end
125
126 X = ['Tabla de errores para t=', num2str(t)];
127 disp(X)
128 disp(' ')
129 disp('     Nodo     Sol. CVM Sol. Exacta  Error(%)')
130 disp(' ')
131 disp(TablaErrores)
132
133 TablaErrores = [];
134
135 end
136 phi_ini=phi;
137
138 end
139 legend('show','Location','SouthWest')
140 title('Campo de Temperaturas','Interpreter','latex')
141 xlabel('$m$', 'Interpreter','latex')
142 ylabel('$T(\circ C)$', 'Interpreter','latex')

```

### A.3 Difusión estacionaria 2D con mallado estructurado

```

1 %% DIFUSION 2D
2
3 clear all; close all; clc;
4
5 % MALLADO EN X-Y
6
7 M = 50;
8 N = 50;
9 L = 0.5;
10 H = 0.5;
11 Deltax = L/M;
12 Deltay = H/N;
13
14 for j = 1:(N+1)
15     ximat(1:(M+1),j) = linspace(0,L,(M+1));
16     etamat(1:(M+1),j) = H*(j-1)/N;
17 end
18
19 figure (1)
20
21 for j = 1:(N+1)
22     plot(ximat(1:(M+1),j), etamat(1:(M+1),j), 'k', 'linewidth', 1.5)

```

```

23     hold on
24 end
25 for i = 1:(M+1)
26     plot(ximat(i,1:(N+1)), etamat(i,1:(N+1)), 'k', 'linewidth', 1.5)
27     hold on
28 end
29 xlabel('$x\,(m)$', 'Interpreter', 'Latex')
30 ylabel('$y\,(m)$', 'Interpreter', 'Latex')
31 title('Mallado del dominio', 'Interpreter', 'Latex')
32 axis([-0.1*L 1.1*L -0.1*H 1.1*H])
33
34 % COORDENADAS DE LOS VERTICES Y NODOS
35
36 for j = 1:N
37     for i = 1:M
38         k = (j-1)*M+i;
39         xi1(k) = ximat(i,j);     eta1(k) = etamat(i,j);
40         xi2(k) = ximat(i,j+1);   eta2(k) = etamat(i,j+1);
41         xi3(k) = ximat(i+1,j+1); eta3(k) = etamat(i+1,j+1);
42         xi4(k) = ximat(i+1,j);   eta4(k) = etamat(i+1,j);
43         xinodo(k) = xi1(k)+(xi3(k)-xi1(k))/2;
44         etanodo(k) = eta1(k)+(eta2(k)-eta1(k))/2;
45
46     end
47 end
48
49 figure(1)
50 nodos = [xinodo', etanodo'];
51 hold on
52 plot(nodos(:,1), nodos(:,2), '.r', 'MarkerSize', 25)
53
54 % DATOS
55
56 Np = M*N;
57
58 Gamma = 1000; % Conductividad Térmica (W/(m*K)) -> CTE
59 nw = -1;     % Vector Normal (nw=-1 cte para caras west)
60 ne = 1;     % Vector Normal (ne=1 cte para caras east)
61 ns = -1;    % Vector Normal (ns=-1 cte para caras south)
62 nn = 1;    % Vector Normal (nn=1 cte para caras north)
63 rho = 1;   % Densidad del Fluido (kg/m^3)
64 t = 0.01;
65 Dw = Gamma*nw*Deltay*t/Deltax;
66 De = Gamma*ne*Deltay*t/Deltax;
67 Ds = Gamma*ns*Deltax*t/Deltay;
68 Dn = Gamma*nn*Deltax*t/Deltay;
69 qE = 450*10^3;
70 qN = -150*10^3;
71 Ae = Deltay*t;
72 An = Deltax*t;
73 TS = 150;
74 k = [1:1:Np]';
75
76 % NODOS INTERMEDIOS
77
78
79 for j = 2:N-1
80     for i = 2:M-1
81         k = (j-1)*M+i;
82         aW(k) = Dw;
83         aP(k) = De+Dn-Dw-Ds;
84         aE(k) = -De;

```

```

85     aS(k) = Ds;
86     aN(k) = -Dn;
87     aSP(k) = 0;
88     A(k,k-1) = aW(k);
89     A(k,k) = aP(k);
90     A(k,k+1) = aE(k);
91     A(k,k+M) = aN(k);
92     A(k,k-M) = aS(k);
93     b(k) = aSP(k);
94     end
95 end
96
97 %%% CONDICIONES DE CONTORNO %%%
98
99 % SOUTH
100
101 for j = 1
102     for i = 2:M-1
103         k = (j-1)*M+i;
104         aW(k) = Dw;
105         aP(k) = De+Dn-Dw-2*Dd;
106         aE(k) = -De;
107         aS(k) = 0;
108         aN(k) = -Dn;
109         aSP(k) = -2*Dd*TS;
110         A(k,k-1) = aW(k);
111         A(k,k) = aP(k);
112         A(k,k+1) = aE(k);
113         A(k,k+M) = aN(k);
114         b(k) = aSP(k);
115     end
116 end
117
118 % NORTH
119
120 for j = N
121     for i = 2:M-1
122         k = (j-1)*M+i;
123         aW(k) = Dw;
124         aP(k) = De-Dw-Dd;
125         aE(k) = -De;
126         aN(k) = 0;
127         aS(k) = Ds;
128         aSP(k) = qN*An*nn;
129         A(k,k-1) = aW(k);
130         A(k,k) = aP(k);
131         A(k,k+1) = aE(k);
132         A(k,k-M) = aS(k);
133         b(k) = aSP(k);
134     end
135 end
136
137 % WEST
138
139 for j = 2:N-1
140     for i = 1
141         k = (j-1)*M+i;
142         aW(k) = 0;
143         aP(k) = De+Dn-Dd;
144         aE(k) = -De;
145         aS(k) = Ds;
146         aN(k) = -Dn;

```

```

147     aSP(k) = 0;
148     A(k,k) = aP(k);
149     A(k,k+1) = aE(k);
150     A(k,k+M) = aN(k);
151     A(k,k-M) = aS(k);
152     b(k) = aSP(k);
153     end
154 end
155
156     % EAST
157
158 for j = 2:N-1
159     for i = M
160         k = (j-1)*M+i;
161         aW(k) = Dw;
162         aP(k) = Dn-Dw-Ds;
163         aE(k) = 0;
164         aS(k) = Ds;
165         aN(k) = -Dn;
166         aSP(k) = qE*Ae*ne;
167         A(k,k-1) = aW(k);
168         A(k,k) = aP(k);
169         A(k,k+M) = aN(k);
170         A(k,k-M) = aS(k);
171         b(k) = aSP(k);
172     end
173 end
174
175 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
176
177     % SOUTH-WEST CORNER
178
179 for j = 1
180     for i = 1
181         k = (j-1)*M+i;
182         aW(k) = 0;
183         aP(k) = De+Dn-2*Ds;
184         aE(k) = -De;
185         aS(k) = 0;
186         aN(k) = -Dn;
187         aSP(k) = -2*Ds*TS;
188         A(k,k) = aP(k);
189         A(k,k+1) = aE(k);
190         A(k,k+M) = aN(k);
191         b(k) = aSP(k);
192     end
193 end
194
195     % SOUTH-EAST CORNER
196
197 for j = 1
198     for i = M
199         k = (j-1)*M+i;
200         aW(k) = Dw;
201         aP(k) = Dn-Dw-2*Ds;
202         aE(k) = 0;
203         aS(k) = 0;
204         aN(k) = -Dn;
205         aSP(k) = -2*Ds*TS+qE*Ae*ne;
206         A(k,k-1) = aW(k);
207         A(k,k) = aP(k);
208         A(k,k+M) = aN(k);

```

```

209     b(k) = aSP(k);
210     end
211 end
212
213 % NORTH-WEST CORNER
214
215 for j = N
216     for i = 1
217         k = (j-1)*M+i;
218         aW(k) = 0;
219         aP(k) = De-Ds;
220         aE(k) = -De;
221         aS(k) = Ds;
222         aN(k) = 0;
223         aSP(k) = +qN*An*nn;
224         A(k,k) = aP(k);
225         A(k,k+1) = aE(k);
226         A(k,k-M) = aS(k);
227         b(k) = aSP(k);
228     end
229 end
230
231 % NORTH-EAST CORNER
232
233 for j = N
234     for i = M
235         k = (j-1)*M+i;
236         aW(k) = Dw;
237         aP(k) = -Dw-Ds;
238         aE(k) = 0;
239         aS(k) = Ds;
240         aN(k) = 0;
241         aSP(k) = qE*Ae*ne+qN*An*nn;
242         A(k,k-1) = aW(k);
243         A(k,k) = aP(k);
244         A(k,k-M) = aS(k);
245         b(k) = aSP(k);
246     end
247 end
248
249 % SOLUCION
250
251 phi1D = A\b';
252
253 for j = 1:N
254     for i = 1:M
255         k = (j-1)*M+i;
256         phi2D(i,j) = phi1D(k);
257         xiplot(i,j) = nodos(k,1);
258         etaplot(i,j) = nodos(k,2);
259     end
260 end
261 phi2D
262
263 figure(3)
264 h=surf(xiplot,etaplot,phi2D);
265 colormap(turbo)
266 cb = colorbar;
267 shading interp
268 set(h,'EdgeColor','k')
269 axis([0 L 0 H])
270 xlabel('$x$, (m)$', 'Interpreter', 'latex')

```

```

271 ylabel('$y\,(m)$', 'Interpreter', 'latex')
272 xlabel('$T\,(^{\circ}C)$', 'Interpreter', 'latex')
273 title('Campo de Temperaturas', 'Interpreter', 'latex')
274
275 figure(4)
276 contour(xiplot, etaplot, phi2D, 150:10:270, 'Linewidth', 2, 'ShowText', 'on');
277 colormap(turbo)
278 xlabel('$x\,(m)$', 'Interpreter', 'latex')
279 ylabel('$y\,(m)$', 'Interpreter', 'latex')
280 xlabel('$T\,(^{\circ}C)$', 'Interpreter', 'latex')
281 title('Curvas de nivel del Campo de Temperaturas', 'Interpreter', 'latex')

```

## A.4 Convección-difusión estacionaria 2D con mallado estructurado

```

1  %% CONVECCION-DIFUSION 2D
2
3  clear all; clc;
4
5  % MALLADO EN X-Y
6
7  M = 50;
8  N = 10;
9  L = 0.5;
10 H = 0.1;
11 Deltax = L/M;
12 Deltay = H/N;
13
14 for j = 1:(N+1)
15     ximat(1:(M+1),j) = linspace(0,L,(M+1));
16     etamat(1:(M+1),j) = H*(j-1)/N;
17 end
18
19 % figure (1)
20 %
21 % for j = 1:(N+1)
22 %     plot(ximat(1:(M+1),j), etamat(1:(M+1),j), 'r', 'linewidth', 1.5)
23 %     hold on
24 % end
25 % for i = 1:(M+1)
26 %     plot(ximat(i,1:(N+1)), etamat(i,1:(N+1)), 'r', 'linewidth', 1.5)
27 %     hold on
28 % end
29 % xlabel('$x\,(m)$', 'Interpreter', 'Latex')
30 % ylabel('$y\,(m)$', 'Interpreter', 'Latex')
31 % title('Mallado del dominio', 'Interpreter', 'Latex')
32 % axis([-0.1 0.6 -0.1 0.2])
33
34 % COORDENADAS DE LOS VERTICES Y NODOS
35
36 for j = 1:N
37     for i = 1:M
38         k = (j-1)*M+i;
39         xi1(k) = ximat(i,j);     eta1(k) = etamat(i,j);
40         xi2(k) = ximat(i,j+1);   eta2(k) = etamat(i,j+1);
41         xi3(k) = ximat(i+1,j+1); eta3(k) = etamat(i+1,j+1);
42         xi4(k) = ximat(i+1,j);   eta4(k) = etamat(i+1,j);
43         xinodo(k) = xi1(k)+(xi3(k)-xi1(k))/2;
44         etanodo(k) = eta1(k)+(eta2(k)-eta1(k))/2;
45

```

```

46     end
47 end
48
49 % figure(1)
50 nodos = [xinodo',etanodo'];
51 % hold on
52 % plot(nodos(:,1),nodos(:,2),'.','MarkerSize',25)
53
54 % DATOS
55
56 Np = M*N;
57
58 Gamma = 0.613; % Conductividad Térmica (W/(m*K)) -> CTE
59 nw = -1;      % Vector Normal (nw=-1 cte para caras west)
60 ne = 1;      % Vector Normal (ne=1 cte para caras east)
61 ns = -1;     % Vector Normal (ns=-1 cte para caras south)
62 nn = 1;     % Vector Normal (nn=1 cte para caras north)
63 rho = 1000;  % Densidad del Fluido (kg/m^3)
64 cp = 4200;  % Calor Especifico del Agua (J/(kg*K))
65 t = 1;
66 um = 2.92*10^-7; % Velocidad media (m/s)
67 Dw = Gamma*nw*Deltay*t/Deltax;
68 De = Gamma*ne*Deltay*t/Deltax;
69 Ds = Gamma*ns*Deltax*t/Deltay;
70 Dn = Gamma*nn*Deltax*t/Deltay;
71 qN = 15;
72 qE = -20;
73 TW = 303;
74 k = [1:1:Np]';
75
76 % NODOS INTERMEDIOS
77
78
79 for j = 2:N-1
80     for i = 2:M-1
81         k = (j-1)*M+i;
82         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
83         Fe(k) = rho*cp*u(k)*ne*Deltay;
84         Fw(k) = rho*cp*u(k)*nw*Deltay;
85         aW(k) = Dw+Fw(k);
86         aP(k) = Fe(k)+De+Dn-Dw-Ds;
87         aE(k) = -De;
88         aS(k) = Ds;
89         aN(k) = -Dn;
90         aSP(k) = 0;
91         A(k,k-1) = aW(k);
92         A(k,k) = aP(k);
93         A(k,k+1) = aE(k);
94         A(k,k+M) = aN(k);
95         A(k,k-M) = aS(k);
96         b(k) = aSP(k);
97     end
98 end
99
100 %%% CONDICIONES DE CONTORNO %%%
101
102 % SOUTH
103
104 for j = 1
105     for i = 2:M-1
106         k = (j-1)*M+i;
107         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);

```

```

108 Fe(k) = rho*cp*u(k)*ne*Deltay;
109 Fw(k) = rho*cp*u(k)*nw*Deltay;
110 aW(k) = Fw(k)+Dw;
111 aP(k) = Fe(k)+De+Dn-Dw;
112 aE(k) = -De;
113 aS(k) = 0;
114 aN(k) = -Dn;
115 aSP(k) = 0;
116 A(k,k-1) = aW(k);
117 A(k,k) = aP(k);
118 A(k,k+1) = aE(k);
119 A(k,k+M) = aN(k);
120 b(k) = aSP(k);
121 end
122 end
123
124 % NORTH
125
126 for j = N
127     for i = 2:M-1
128         k = (j-1)*M+i;
129         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
130         Fe(k) = rho*cp*u(k)*ne*Deltay;
131         Fw(k) = rho*cp*u(k)*nw*Deltay;
132         aW(k) = Fw(k)+Dw;
133         aP(k) = Fe(k)+De-Dw-Ds;
134         aE(k) = -De;
135         aN(k) = 0;
136         aS(k) = Ds;
137         aSP(k) = qN*Deltax*nn;
138         A(k,k-1) = aW(k);
139         A(k,k) = aP(k);
140         A(k,k+1) = aE(k);
141         A(k,k-M) = aS(k);
142         b(k) = aSP(k);
143     end
144 end
145
146 % WEST
147
148 for j = 2:N-1
149     for i = 1
150         k = (j-1)*M+i;
151         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
152         Fe(k) = rho*cp*u(k)*ne*Deltay;
153         Fw(k) = rho*cp*u(k)*nw*Deltay;
154         aW(k) = 0;
155         aP(k) = Fe(k)+De+Dn-2*Dw-Ds;
156         aE(k) = -De;
157         aS(k) = Ds;
158         aN(k) = -Dn;
159         aSP(k) = -(2*Dw+Fw(k))*TW;
160         A(k,k) = aP(k);
161         A(k,k+1) = aE(k);
162         A(k,k+M) = aN(k);
163         A(k,k-M) = aS(k);
164         b(k) = aSP(k);
165     end
166 end
167
168 % EAST
169

```

```

170 for j = 2:N-1
171     for i = M
172         k = (j-1)*M+i;
173         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
174         Fe(k) = rho*cp*u(k)*ne*Deltay;
175         Fw(k) = rho*cp*u(k)*nw*Deltay;
176         aW(k) = Fw(k)+Dw;
177         aP(k) = Fe(k)+Dn-Dw-Ds;
178         aE(k) = 0;
179         aS(k) = Ds;
180         aN(k) = -Dn;
181         aSP(k) = qE*Deltay*ne;
182         A(k,k-1) = aW(k);
183         A(k,k) = aP(k);
184         A(k,k+M) = aN(k);
185         A(k,k-M) = aS(k);
186         b(k) = aSP(k);
187     end
188 end
189
190 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
191
192 % SOUTH-WEST CORNER
193
194 for j = 1
195     for i = 1
196         k = (j-1)*M+i;
197         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
198         Fe(k) = rho*cp*u(k)*ne*Deltay;
199         Fw(k) = rho*cp*u(k)*nw*Deltay;
200         aW(k) = 0;
201         aP(k) = Fe(k)+De+Dn-2*Dw;
202         aE(k) = -De;
203         aS(k) = 0;
204         aN(k) = -Dn;
205         aSP(k) = -(2*Dw+Fw(k))*TW;
206         A(k,k) = aP(k);
207         A(k,k+1) = aE(k);
208         A(k,k+M) = aN(k);
209         b(k) = aSP(k);
210     end
211 end
212
213 % SOUTH-EAST CORNER
214
215 for j = 1
216     for i = M
217         k = (j-1)*M+i;
218         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
219         Fe(k) = rho*cp*u(k)*ne*Deltay;
220         Fw(k) = rho*cp*u(k)*nw*Deltay;
221         aW(k) = Fw(k)+Dw;
222         aP(k) = Fe(k)+Dn-Dw;
223         aE(k) = 0;
224         aS(k) = 0;
225         aN(k) = -Dn;
226         aSP(k) = qE*Deltay*ne;
227         A(k,k-1) = aW(k);
228         A(k,k) = aP(k);
229         A(k,k+M) = aN(k);
230         b(k) = aSP(k);
231     end

```

```

232 end
233
234 % NORTH-WEST CORNER
235
236 for j = N
237     for i = 1
238         k = (j-1)*M+i;
239         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
240         Fe(k) = rho*cp*u(k)*ne*Deltay;
241         Fw(k) = rho*cp*u(k)*nw*Deltay;
242         aW(k) = 0;
243         aP(k) = Fe(k)+De-2*Dw-Ds;
244         aE(k) = -De;
245         aS(k) = Ds;
246         aN(k) = 0;
247         aSP(k) = -(2*Dw+Fw(k))*TW+qN*Deltax*nn;
248         A(k,k) = aP(k);
249         A(k,k+1) = aE(k);
250         A(k,k-M) = aS(k);
251         b(k) = aSP(k);
252     end
253 end
254
255 % NORTH-EAST CORNER
256
257 for j = N
258     for i = M
259         k = (j-1)*M+i;
260         u(k) = 6*um*(etanodo(k)/H-(etanodo(k)/H)^2);
261         Fe(k) = rho*cp*u(k)*ne*Deltay;
262         Fw(k) = rho*cp*u(k)*nw*Deltay;
263         aW(k) = Fw(k)+Dw;
264         aP(k) = Fe(k)-Dw-Ds;
265         aE(k) = 0;
266         aS(k) = Ds;
267         aN(k) = 0;
268         aSP(k) = qN*Deltax*nn+qE*Deltay*ne;
269         A(k,k-1) = aW(k);
270         A(k,k) = aP(k);
271         A(k,k-M) = aS(k);
272         b(k) = aSP(k);
273     end
274 end
275
276 % SOLUCION
277
278 phi1D = A\b'-273;
279
280 for j = 1:N
281     for i = 1:M
282         k = (j-1)*M+i;
283         phi2D(i,j) = phi1D(k);
284         xiplot(i,j) = nodos(k,1);
285         etaplot(i,j) = nodos(k,2);
286     end
287 end
288 phi2D
289
290 figure(4)
291 h=surf(xiplot,etaplot,phi2D);
292 colormap(turbo)
293 cb = colorbar;

```

```
294 shading interp
295 set(h,'EdgeColor','k')
296 xlabel('$x\,(m)$','Interpreter','latex')
297 ylabel('$y\,(m)$','Interpreter','latex')
298 zlabel('$T\,(^{\circ}C)$','Interpreter','latex')
299 title('Campo de Temperaturas','Interpreter','latex')
300
301 figure(5)
302 contour(xiplot,etaplot,phi2D,[30:1:45],'Linewidth',2,'ShowText','on');
303 colormap(turbo)
304 xlabel('$x\,(m)$','Interpreter','latex')
305 ylabel('$y\,(m)$','Interpreter','latex')
306 zlabel('$T\,(^{\circ}C)$','Interpreter','latex')
307 title('Curvas de nivel del Campo de Temperaturas','Interpreter','latex')
```



# Bibliografía

---

- [1] S. V. PATANKAR (1980): *Numerical Heat Transfer and Fluid Flow*, 1ª Edición, Boca Raton (Florida), CRC Press.
- [2] A. BARRERO Y M. PÉREZ-SABORID (2005): *Fundamentos y Aplicaciones de la Mecánica De Fluidos*, 1ª Edición, España, McGraw Hill.
- [3] A. W. DATE (2005): *Introduction to Computational Fluid Dynamics*, Instituto Tecnológico de India (Bombay), Cambridge University Press.
- [4] H. K. VERSTEEG AND W. MALALASEKERA (2007): *An Introduction to Computational Fluid Dynamics*, 2ª Edición, Inglaterra, Pearson Education.
- [5] V. R. VOLLER (2009): *Basic Control Volume Finite Element Methods for Fluids and Solids*, 1ª Edición, Singapore, IISc Press.
- [6] S. MAZUMDER (2016): *Numerical Methods for Partial Differential Equations*, Universidad Estatal de Ohio, Elsevier Inc.
- [7] ANSYS (2017): *ANSYS Fluent Theory Guide*, Release 18.2, ANSYS, Inc. and ANSYS Europe.
- [8] A. G. GALINDO (2017): *Modelado CFD de la atomización del combustible en quemadores de turbinas de gas*, Trabajo Fin de Máster, Universitat Politècnica de València, Escuela Técnica Superior de Ingeniería del Diseño. Disponible en: <https://riunet.upv.es/handle/10251/88196> [Consulta: 25-03-2022].
- [9] A. BAKKER (2008): *Applied Computational Fluid Dynamics*. CFD Lectures, Lecture 5-Solution Methods. Disponible en: <http://www.bakker.org> [Consulta: 31-03-2022].
- [10] A. KAUSHIK (2016): *Critical Evaluation of Four Differencing Schemes for A Steady Convection-Diffusion Problem*. Research Journal of Mathematical and Statistical Sciences, Volume 4, No. 2, pp 6-12 Disponible en: [http://www.isca.in/MATH\\_SCI/v4i2.php](http://www.isca.in/MATH_SCI/v4i2.php) [Consulta: 12-04-2022].
- [11] Y. NIÑO (2008): *Método de Volúmenes Finitos*. Dpto de Ing. Civil, Universidad de Chile. Disponible en: [https://www.u-cursos.cl/ingenieria/2008/2/CI71D/1/material\\_docente/detalle?id=200975](https://www.u-cursos.cl/ingenieria/2008/2/CI71D/1/material_docente/detalle?id=200975) [Consulta: 20-04-2022].
- [12] A. ATEŞ, O. ALTUN AND A. KILICMAN (2017): *On A Comparison of Numerical Solution Methods for General Transport Equation on Cylindrical Coordinates*. Applied Mathematics & Information Sciences, Volume 11, No. 2, pp 433-439. Disponible en: <https://www.naturalspublishing.com/Article.asp?ArtcID=12846> [Consulta: 20-04-2022].

- [13] W. Y. TEY, R. Y. WYE HONG, Y. ASAKO, H. S. KANG AND K. C. NG (2019): *Analysis on Computational Efficiency of Convection Discretisation Schemes in SIMPLE Algorithm*. Journal of Advanced Research in Fluid Mechanics and Thermal Sciences, Volume 58, No. 1, pp 100-117. Disponible en: <https://www.akademiabaru.com/submit/index.php/arfmts/article/view/100-117> [Consulta: 26-03-2022].
- [14] WIKIWAND: *Power Law Scheme*. Disponible en: [https://www.wikiwand.com/en/Power\\_law\\_scheme](https://www.wikiwand.com/en/Power_law_scheme) [Consulta: 21-04-2022].
- [15] WIKIWAND: *False Diffusion*. Disponible en: [https://www.wikiwand.com/en/False\\_diffusion](https://www.wikiwand.com/en/False_diffusion) [Consulta: 21-04-2022].
- [16] AMWEL: *CFD History*. Disponible en: <http://www.amwel.com/history.html> [Consulta: 12-05-2022].
- [17] S. MAZUMDER (2016). *Finite Volume Method: Unstructured Mesh (Part 1)*. YouTube. <https://www.youtube.com/watch?v=FUyVOMvDMoM>
- [18] S. MAZUMDER (2016). *Finite Volume Method: Unstructured Mesh (Part 2)*. YouTube. <https://www.youtube.com/watch?v=k8O1soqZLz0>
- [19] S. MAZUMDER (2016). *Finite Volume Method: Unstructured Mesh (Part 3)*. YouTube. <https://www.youtube.com/watch?v=n4alY826-Ns>
- [20] S. MAZUMDER (2016). *Finite Volume Method: Unstructured Mesh (Part 4)*. YouTube. [https://www.youtube.com/watch?v=aVb2\\_VBjMgwM](https://www.youtube.com/watch?v=aVb2_VBjMgwM)