

Trabajo Fin de Máster



VNIVERSITAT ID VALÈNCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Máster Universitario en
Investigación Matemática

Curso 2019/2020

**Métodos numéricos de alto orden
para un modelo de cromatografía**

Autor: **Alexandre Muñoz Gutiérrez**

Tutora: MCARMEN MARTÍ RAGA

Índice general

1. Introducción	1
2. Teoría de leyes de conservación escalares	3
2.1. Introducción a las leyes de conservación	3
2.2. La ecuación escalar lineal	4
2.3. La ecuación escalar no lineal	6
2.3.1. Método de las características	6
2.3.2. Concepto de solución débil	8
2.3.3. La condición de Rankine-Hugoniot	9
2.3.4. Cálculo de algunas soluciones débiles	10
2.3.5. Condición de entropía	13
2.3.6. Soluciones de entropía de los problemas de Riemann	15
3. Métodos numéricos para la ecuación lineal	17
3.1. Introducción	17
3.2. Conceptos básicos	18
3.3. La condición CFL	19
3.4. El método upwind	21
3.5. El método de Lax-Wendroff	24
4. Métodos de volúmenes finitos	29
4.1. Introducción	29
4.2. Ecuación general	30
4.3. Deducción de algunos métodos numéricos	30
4.3.1. Método de Godunov	31
4.3.2. Método de Roe	33
4.3.3. Método de Rusanov	34
4.4. Análisis de la convergencia	35
4.5. Reconstrucción espacial de segundo orden	38
4.6. Formulación semidiscreta	40
5. Métodos de reconstrucción de orden alto	43
5.1. Reconstrucción de funciones suaves	43
5.2. Método ENO	45
5.3. Método WENO	46
5.4. Aplicación a los métodos de volúmenes finitos	48
5.5. Método de diferencias finitas de Shu y Osher	50
6. Extensión a sistemas de ecuaciones	53
6.1. Introducción: sistemas hiperbólicos	53
6.2. Teoría de sistemas lineales	54
6.3. Métodos numéricos para sistemas lineales	56
6.4. Teoría de sistemas no lineales	57
6.5. Métodos numéricos para sistemas no lineales	59

7. Aplicación a un modelo de cromatografía	63
7.1. Introducción	63
7.2. Análisis matemático del modelo	66
7.3. Métodos numéricos para la resolución del modelo	70
7.4. Experimentos numéricos	75
8. Códigos de MatLab	77
9. Conclusiones	107

Capítulo 1

Introducción

La cromatografía es una herramienta empleada en Química Analítica para la separación e identificación de cada uno de los componentes de una mezcla. Al introducir la mezcla junto con un disolvente en una columna, los distintos componentes precipitan a diferentes velocidades según su interacción con el medio poroso situado en el interior, de modo que cada componente puede recogerse al final de la columna de manera separada.

Este proceso de separación de sustancias, como multitud de problemas en dinámica de fluidos, se modeliza mediante ecuaciones hiperbólicas de leyes de conservación, las cuales constituyen el tema central del trabajo. Resulta de gran utilidad resolverlas numéricamente, para simular las situaciones que representan y evitar, así, experimentos de prueba y error. Cada vez se requieren soluciones numéricas de mayor precisión, por lo que deben desarrollarse métodos más y más potentes.

Estas ecuaciones hiperbólicas de leyes de conservación difieren de las estudiadas en los primeros cursos de ecuaciones diferenciales, ya que sus soluciones pueden generar singularidades a tiempo finito. Por esta razón, no pueden diseñarse métodos numéricos basados en sustituir las derivadas por diferencias finitas, sino que éstos deben construirse desde una perspectiva distinta que permita la convergencia a la solución.

En el capítulo 2 se realiza un estudio teórico de la ecuación hiperbólica escalar, que toma la forma $u_t + f(u)_x = 0$. Tras deducir esta expresión a partir del principio de conservación en la sección 2.1, en el apartado 2.2 se resuelve analíticamente la ecuación lineal $u_t + au_x = 0$ por el conocido método de las características. Por su parte, la sección 2.3 se dedica a analizar la ecuación anterior cuando $f(u)$ es una función no lineal. Se comprueba que, en algunas ocasiones, tan solo es posible definir una solución clásica en un breve intervalo de tiempo. Para su extensión, se generaliza el concepto de solución, dando cabida de esta forma a funciones de clase C^1 a trozos que satisfacen una cierta propiedad en las discontinuidades. Finalmente, para seleccionar la solución físicamente correcta entre todas las anteriores, se impone una condición adicional conocida como condición de entropía.

Posteriormente, en el capítulo 3 se repasan los métodos numéricos empleados para resolver la ecuación escalar lineal. En la sección 3.1 se definen los mallados en el dominio espacial y temporal, y en la sección 3.2 se introducen los conceptos de convergencia, consistencia y estabilidad. Posteriormente, en el apartado 3.3 se estudia una propiedad exclusiva de estos métodos, la condición CFL, que resulta ser una condición necesaria para la convergencia de los esquemas numéricos. Finalmente, en las secciones 3.4 y 3.5 se presentan los métodos *upwind* y *Lax-Wendroff*, se prueba su convergencia y se muestra un estudio basado en series de Fourier para analizar su comportamiento ante soluciones con discontinuidades. Estas propiedades se ponen de manifiesto en los experimentos numéricos realizados a lo largo del capítulo.

Por otro lado, el capítulo 4 se dedica a los métodos numéricos clásicos para la resolución de la ecuación escalar no lineal: los métodos de volúmenes finitos. En las secciones 4.1 y 4.2 se introducen

los mallados y la ecuación general de estos esquemas, en los que no se estima el valor de la solución en un punto sino su promedio en subintervalos del dominio, En el apartado 4.3 se presentan los métodos de *Godunov*, *Roe* y *Rusanov*, y se muestran pruebas numéricas acerca de éstos. En la sección 4.4 se definen propiedades que todo esquema robusto debe cumplir, hecho que justifica alguna anomalía observada en los experimentos previos. En la sección 4.5 se propone una reconstrucción espacial de orden dos que mejora la estimación de orden uno asociada a los métodos anteriores. Finalmente, para aumentar el orden de convergencia en la variable temporal, en el apartado 4.6 se presenta una formulación semidiscreta que permite aplicar un método de Runge-Kutta de orden superior para resolver el sistema resultante.

El capítulo 5 se centra en las técnicas de reconstrucción de orden alto *ENO* y *WENO*. Dada una partición \mathcal{P} de un cierto intervalo, conocidos los promedios de una función en los subintervalos que \mathcal{P} genera, mediante estas técnicas se consigue hallar estimaciones de dicha función en cada nodo de \mathcal{P} . Las herramientas previas, expuestas en las tres primeras secciones, pueden emplearse para sustituir la estimación espacial de orden dos tratada en el capítulo anterior, como se muestra en el apartado 5.4. Por último, en la sección 5.5 se presenta una manera alternativa de aplicar los métodos *ENO* y *WENO* para resolver estas leyes de conservación: el esquema de diferencias finitas de Shu y Osher.

El capítulo 6, por su parte, está dedicado a extender los contenidos estudiados en el caso escalar para sistemas de ecuaciones. Como se muestra en la sección 6.1, es ahora cuando adquiere mayor sentido el concepto de hiperbolicidad. En la sección 6.2 se presenta la solución analítica de los sistemas lineales, y en el siguiente apartado se generalizan los métodos *upwind* y *Lax-Wendroff* para hallarla numéricamente. En la sección 6.4 se estudian los sistemas no lineales desde el punto de vista teórico. En el último apartado del capítulo se comprueba que resulta inviable aplicar los métodos de volúmenes finitos para sistemas de ecuaciones, mientras que es sencillo y eficaz generalizar el método de Shu y Osher introducido en la sección 5.5.

Como aplicación, en el capítulo 7 se reproducen los resultados del artículo [8] acerca de un modelo de cromatografía. En la sección 7.1 se explica con detalle este procedimiento y se deducen las ecuaciones del modelo. A continuación, se razona que estas ecuaciones pueden reescribirse como un sistema hiperbólico de leyes de conservación junto con un término parabólico, en el apartado 7.2. En la sección 7.3 se resuelve el modelo planteado, incluyendo la técnica de reconstrucción *WENO* y el método de diferencias finitas de Shu y Osher. En el apartado final del capítulo se llevan a cabo distintos experimentos numéricos siguiendo las indicaciones del artículo.

En el capítulo 8 se incluyen los códigos de MatLab implementados para generar las tablas y gráficas mostradas a lo largo del trabajo. Por último, en el capítulo 9 se recogen las conclusiones del mismo.

Capítulo 2

Teoría de leyes de conservación escalares

En este capítulo se introducen las ecuaciones hiperbólicas de leyes de conservación. Se repasan las principales propiedades de las ecuaciones escalares, tanto lineales como no lineales, con el objetivo de obtener cierta información que permita diseñar métodos numéricos adecuados para su resolución.

2.1. Introducción a las leyes de conservación

En multitud de problemas de física, biología o ingeniería están presentes las leyes de conservación, según las cuales la cantidad de una cierta sustancia en un sistema no varía en el tiempo. Esta conservación implica que la variación de sustancia contenida en un cierto volumen tan solo se debe por los flujos de materia a través de sus fronteras.

Tómese un intervalo $I = (x_1, x_2) \subset \mathbb{R}$ y una variable de interés $u = u(x, t) = (u_1, \dots, u_m) \in \mathbb{R}^m$ definida para cada $x \in I$ e instante de tiempo $t \geq 0$. Esta variable puede representar, por ejemplo, la temperatura de una barra, la presión de un fluido o la concentración de una sustancia química. El principio de conservación se expresa matemáticamente como

$$\frac{d}{dt} \int_{x_1}^{x_2} u(x, t) dx = f(x_1, t) - f(x_2, t), \quad (2.1)$$

donde $f = f(x, t) = (f_1, \dots, f_m) \in \mathbb{R}^m$ corresponde al flujo de u que entra o sale de I por sus extremos.

Integrando (2.1) respecto del tiempo en un intervalo (t_1, t_2) arbitrario, se obtiene

$$\int_{x_1}^{x_2} u(x, t_2) dx - \int_{x_1}^{x_2} u(x, t_1) dx = \int_{t_1}^{t_2} f(x_1, t) dt - \int_{t_1}^{t_2} f(x_2, t) dt. \quad (2.2)$$

Suponiendo que u y f son funciones diferenciables, mediante las relaciones

$$u(x, t_2) - u(x, t_1) = \int_{t_1}^{t_2} u_t(x, t) dt, \quad f(x_2, t) - f(x_1, t) = \int_{x_1}^{x_2} f_x(x, t) dx$$

se concluye

$$\int_{t_1}^{t_2} \int_{x_1}^{x_2} (u_t + f_x)(x, t) dx dt = 0. \quad (2.3)$$

En estas expresiones, u_t denota la función derivada de u respecto del tiempo y f_x , la derivada de f respecto de x .

Como se cumple (2.3) para intervalos (x_1, x_2) y (t_1, t_2) cualesquiera, el integrando debe ser idénticamente cero. Se deduce, así, la forma diferencial de la ley de conservación:

$$u_t + f_x = 0, \quad (x, t) \in \mathbb{R} \times (0, +\infty). \quad (2.4)$$

Cabe notar que (2.4) incluye una gran cantidad de modelos, como la ecuación del calor si $f = -u_x$. No obstante, en este trabajo se fija $f = f(u)$, de modo que (2.4) se reduce a una ecuación en derivadas parciales de primer orden:

$$u_t + f(u)_x = 0, \quad (x, t) \in \mathbb{R} \times (0, \infty). \quad (2.5)$$

En los siguientes apartados se analiza (2.5) en su versión escalar, esto es, para $m = 1$. En primer lugar se toma $f(u) = au$, y posteriormente se estudia la estructura de las soluciones cuando $f(u)$ es una función no lineal. En ambos casos, (2.5) recibe el nombre de ecuación hiperbólica, aunque este concepto adquiere mayor relevancia cuando $m > 1$, como se muestra en el capítulo 6.

2.2. La ecuación escalar lineal

Para comenzar, se trata la ecuación de advección lineal, el caso más sencillo de ley de conservación. Esto permite familiarizarse con el método de las características, que luego constituye el punto de partida para resolver el modelo no lineal.

Sea $a \in \mathbb{R}$ y $u_0 : \mathbb{R} \rightarrow \mathbb{R}$ una función de clase \mathcal{C}^1 , se considera el problema escalar

$$\begin{aligned} u_t + au_x &= 0, & (x, t) \in \Omega &= \mathbb{R} \times (0, +\infty), \\ u(x, 0) &= u_0(x), & x &\in \mathbb{R}. \end{aligned} \quad (2.6)$$

A continuación se deduce una solución de (2.6) y se muestra que es única.

Defínase la función $t \rightsquigarrow x(t; x_0) = at + x_0$, que verifica la ecuación diferencial $x'(t) = a$ y la condición $x(0) = x_0$, siendo x_0 un número real arbitrario. Suponiendo que existe una solución $u = u(x, t)$ de (2.6), ésta debe satisfacer

$$\begin{aligned} \frac{d}{dt} u(x(t; x_0), t) &= u_x(x(t; x_0), t) \cdot x'(t; x_0) + u_t(x(t; x_0), t) \\ &= au_x(x(t; x_0), t) + u_t(x(t; x_0), t) \\ &= 0. \end{aligned}$$

Así, u es constante a lo largo de cada una de las rectas

$$\mathcal{C}_{x_0} = \{(at + x_0, t) : t \geq 0\}, \quad x_0 \in \mathbb{R}.$$

Estas rectas paralelas, que reciben el nombre de rectas características del problema, cubren todo el semiplano Ω : dado $(x, t) \in \Omega$, existe (un único) $x_0 = x - at$ de modo que $(x, t) \in \mathcal{C}_{x_0}$, por lo que

$$u(x, t) = u(x_0, 0) = u_0(x_0) = u_0(x - at). \quad (2.7)$$

De esta manera se ha demostrado que, en caso de que exista, la solución debe ser (2.7). Para ver que, en efecto, lo es, se comprueba que

$$u_t(x, t) + au_x(x, t) = u'_0(x - at)(-a) + au'_0(x - at) = 0$$

para todo par $(x, t) \in \Omega$ y $u(x, 0) = u_0(x)$ para cualquier $x \in \mathbb{R}$.

Obviamente, (2.7) puede definirse independientemente de que u_0 sea suave o no. Además, al tratarse de una simple traslación, conserva en \mathcal{C}_{x_0} las singularidades que la función inicial presenta en el punto x_0 asociado. De esta forma, si u_0 no es de clase \mathcal{C}^1 en algún punto, (2.7) tampoco lo

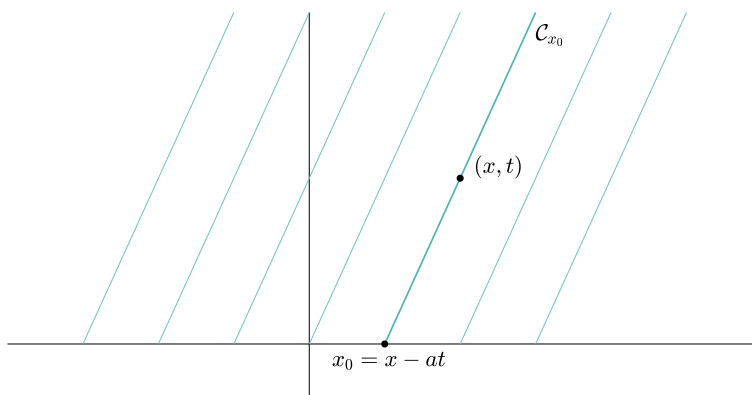


Figura 2.1: Rectas características asociadas al problema escalar lineal (2.6).

es, y por tanto no puede ser solución de la ecuación en derivadas de (2.6).

No obstante, puede comprobarse que, aunque u_0 no sea suave, (2.7) sí satisface la forma integral de la ley de conservación, es decir,

$$\int_{x_1}^{x_2} u(x, t_2) dx - \int_{x_1}^{x_2} u(x, t_1) dx = a \left(\int_{t_1}^{t_2} u(x_1, t) dt - \int_{t_1}^{t_2} u(x_2, t) dt \right) \quad (2.8)$$

para valores $x_1 < x_2$ y $t_1 < t_2$ cualesquiera. Cabe notar que la forma integral (2.8) modeliza la realidad mejor que la ecuación diferencial, ya que ésta se obtiene a partir de la primera añadiendo la condición de suavidad de los integrandos. Así pues, parece lógico tomar la función

$$u(x, t) = u_0(x - at), \quad (x, t) \in \mathbb{R} \times [0, +\infty) \quad (2.9)$$

como solución del proceso físico estudiado.

Por último se muestra otra manera de definir la solución generalizada de (2.6), que puede extenderse al estudio de la ecuación no lineal. Esta alternativa consiste en mantener invariante el dato inicial u_0 y añadir un término de difusión a la ecuación en derivadas parciales:

$$u_t + au_x = \varepsilon u_{xx}, \quad (2.10)$$

para $\varepsilon > 0$ pequeño arbitrario. Como la ecuación (2.10) es parabólica, su solución, denotada por u^ε , pertenece al espacio $C^\infty(\mathbb{R} \times (0, \infty))$ aunque u_0 no sea suave. Puede obtenerse mediante el cambio de variable

$$v^\varepsilon(x, t) = u^\varepsilon(x + at, t),$$

ya que v^ε verifica la ecuación del calor $v_t^\varepsilon = \varepsilon v_{xx}^\varepsilon$. Finalmente se comprueba que el límite de $u^\varepsilon(x, t)$ cuando ε tiende a 0 coincide con (2.9).

Una vez analizado el caso más sencillo, se estudia la ecuación lineal general:

$$\begin{aligned} u_t + a(x, t)u_x &= 0, & (x, t) \in \Omega = \mathbb{R} \times (0, \infty), \\ u(x, 0) &= u_0(x), & x \in \mathbb{R}, \end{aligned} \quad (2.11)$$

siendo u_0 de clase C^1 y $a : \Omega \rightarrow \mathbb{R}$, Lipschitz continua respecto de x y t . Como en el caso anterior, y para un número real x_0 arbitrario, se considera el problema

$$x' = a(x, t), \quad x(0) = x_0, \quad (2.12)$$

el cual admite una única solución, $x(t; x_0)$, por las condiciones impuestas sobre la función a . Esto implica que, para valores x_0 distintos, las gráficas de $x(t; x_0)$ son disjuntas.

Análogamente se comprueba que la solución de (2.11) debe ser constante en cada una de las curvas características $\mathcal{C}_{x_0} = \{(x(t; x_0), t) : t \geq 0\}$, de modo que ésta cumple

$$u(x(t; x_0), t) = u_0(x_0), \quad t \geq 0, \quad x_0 \in \mathbb{R}. \quad (2.13)$$

Así, si estas curvas cubren todo el semiplano Ω , se habrá hallado la solución del problema. De nuevo, (2.13) puede obtenerse aunque u_0 no sea suave, y en ese caso verificará la forma integral de la ley de conservación.

2.3. La ecuación escalar no lineal

El siguiente paso del estudio es la ley de conservación escalar no lineal, con un comportamiento totalmente distinto al que presenta el modelo lineal. Se muestra que las curvas características tan solo permiten resolver el problema en algunos casos particulares, por lo que es necesario introducir nuevos conceptos que ayuden a conseguir el objetivo.

2.3.1. Método de las características

Así, esta sección se centra en la ecuación en derivadas parciales no lineal

$$u_t + f(u)_x = 0, \quad (2.14)$$

siendo f suficientemente suave. Denotando por $a(u)$ a la función derivada de $f(u)$, es claro que (2.14) equivale a

$$u_t + a(u)u_x = 0. \quad (2.15)$$

De ahora en adelante se usa la ecuación (2.14) o (2.15) indistintamente, según interese en cada momento. A continuación se trata el problema de Cauchy

$$\begin{aligned} u_t + f(u)_x = u_t + a(u)u_x = 0, \quad (x, t) \in \Omega = \mathbb{R} \times (0, +\infty), \\ u(x, 0) = u_0(x), \quad x \in \mathbb{R}, \end{aligned} \quad (2.16)$$

para un dato inicial $u_0 : \mathbb{R} \rightarrow \mathbb{R}$ de clase \mathcal{C}^1 .

Suponiendo que existe $u = u(x, t)$ solución de (2.16), se considera el problema

$$x'(t) = a(u), \quad x(0) = x_0,$$

y se denota por $x(t; x_0)$ a su solución, en caso de que exista. Como en los casos anteriores, se tiene

$$\frac{d}{dt}(u(x(t; x_0), t)) = u_x(x(t; x_0), t) \cdot x'(t; x_0) + u_t(x(t; x_0), t) = 0,$$

pues, de nuevo, u es constante en las curvas $\{(x(t; x_0), t) : t \geq 0\}$. Ahora bien, cabe notar que

$$x'(t; x_0) = a(u(x(t; x_0), t)) = a(u_0(x_0)),$$

de modo que $x(t; x_0) = a(u_0(x_0))t + x_0$. Así, las curvas características son ahora las rectas

$$\mathcal{C}_{x_0} = \{(a(u_0(x_0))t + x_0, t) : t \geq 0\}, \quad x_0 \in \mathbb{R}.$$

A diferencia de la ecuación lineal, no puede asegurarse que estas rectas no se crucen, y tampoco se sabe si cubren todo el semiplano Ω . Estas cuestiones se resumen en la pregunta que sigue:

Dado $(x, t) \in \Omega$, ¿existe un único $x_0 = x_0(x, t) \in \mathbb{R}$ tal que $(x, t) \in \mathcal{C}_{x_0}$?

En caso afirmativo, la solución de (2.16) sería

$$u(x, t) = u_0(x_0(x, t)), \quad (x, t) \in \Omega, \quad (2.17)$$

de clase \mathcal{C}^1 . Pero... ¿esto es cierto?

Por el teorema de la función implícita, la pregunta previa es afirmativa siempre y cuando

$$g(x_0, x, t) = a_0(x_0)t + x_0 - x, \quad a_0 = a \circ u$$

cumpla

$$\frac{\partial g}{\partial x_0}(x_0, x, t) = a'_0(x_0)t + 1 \neq 0, \quad \forall x_0, x \in \mathbb{R}, t \geq 0. \quad (2.18)$$

Es claro que esta condición se verifica si a_0 es creciente. En caso contrario, no obstante, sí puede asegurarse (2.18) para valores de t suficientemente pequeños. El menor valor de t para el cual (2.18) no se cumple se denota por t_c y recibe el nombre de tiempo crítico. De esta forma, se tiene solución del problema (2.16) en la franja $\mathbb{R} \times [0, t_c)$. Se muestra a continuación la expresión que toma t_c cuando a_0 no es creciente.

De esta forma, existe $\xi \in \mathbb{R}$ tal que $a'_0(\xi) < 0$, de modo que

$$a_0(\xi) > a_0(\xi + h), \quad h \in (0, \delta) \quad (2.19)$$

para cierto $\delta > 0$. Por (2.19), las rectas características asociadas a ξ y $\xi + h$ se cruzan, en el instante

$$t_h = -\frac{h}{a_0(\xi + h) - a_0(\xi)}.$$

Tomando límites cuando $h \rightarrow 0$ se deduce $t_h \rightarrow -\frac{1}{a'_0(\xi)}$. Como consecuencia, t_c viene dado por

$$t_c = \inf_{\xi \in \mathbb{R}} \left(-\frac{1}{a'_0(\xi)} \right) = -\frac{1}{\inf_{\xi \in \mathbb{R}} a'_0(\xi)}. \quad (2.20)$$

Para $t = t_c$ y quizás para instantes de tiempo posteriores, las rectas características pueden dejar una zona del semiplano sin cubrir, por lo que no sería posible definir la solución en cualquier punto. La otra posibilidad es que dos de estas rectas intersecten: si \mathcal{C}_{x_1} y \mathcal{C}_{x_2} se cortan, sus pendientes son distintas, lo que implica $u_0(x_1) \neq u_0(x_2)$. De esta forma, habría dos posibilidades para definir el valor de la solución en el punto de intersección. Es claro que funciones multievaluadas no tienen sentido físico normalmente, de modo que no puede aceptarse esta situación para $t > t_c$. La discusión previa se refleja en el siguiente ejemplo.

Ejemplo 1. Para $a(u) = u$, (2.15) recibe el nombre de ecuación de Burgers:

$$u_t + \left(\frac{u^2}{2} \right)_x = 0 \quad \equiv \quad u_t + uu_x = 0. \quad (2.21)$$

En este ejemplo se combina (2.21) con un dato inicial de clase \mathcal{C}^1 :

$$u_0(x) = \begin{cases} 2, & x \leq \pi/2, \\ \sin(x) + 1, & \pi/2 \leq x \leq 3\pi/2, \\ 0, & x \geq 3\pi/2. \end{cases} \quad (2.22)$$

Su función derivada es $u'_0(x) = \cos(x)\mathbb{1}_{(\frac{\pi}{2}, \frac{3\pi}{2})}(x)$, de modo que, siguiendo la expresión (2.20), el tiempo crítico toma el valor $t_c = 1$. Se muestran algunas rectas características en la Figura 2.2.

En ella se observa que estas rectas no intersectan cuando $t < 1$, de modo que puede definirse la solución suave (2.17) en esta región. No obstante, a partir de $t_c = 1$, las características se cortan en una determinada zona, donde (2.17) pasa a ser multievaluada.

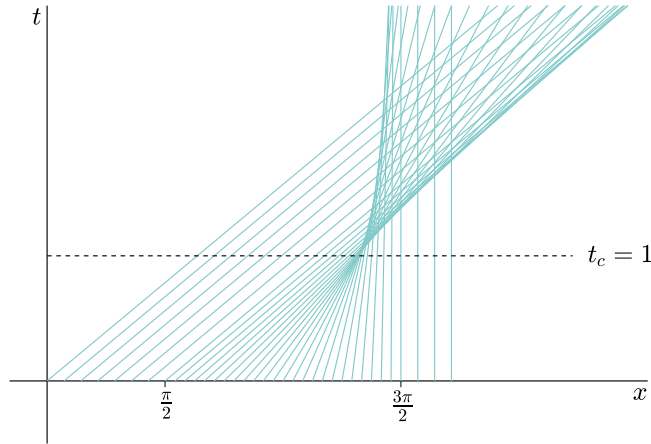


Figura 2.2: Rectas características asociadas a la ecuación de Burgers con valor inicial (2.22).

2.3.2. Concepto de solución débil

Cabe recordar que el método de las características permite resolver la ley de conservación lineal $u_t + au_x = 0$ en todo su dominio. La solución verifica la ecuación diferencial si el dato inicial es suave y (2.8) si éste presenta alguna singularidad.

Por otro lado, para el caso no lineal con u_0 de clase C^1 , en algunas ocasiones tan solo es posible hallar la solución para valores de t menores que un cierto tiempo crítico. Con el fin de prolongar esta función, debe generalizarse el concepto de solución. Así, análogamente a la sección 2.2, se podría regresar a la forma integral de la ley de conservación y afirmar que $u(x, t)$ es solución generalizada de (2.14) si verifica

$$\int_{x_1}^{x_2} u(x, t_2) dx - \int_{x_1}^{x_2} u(x, t_1) dx = \int_{t_1}^{t_2} f(u(x_1, t)) dt - \int_{t_1}^{t_2} f(u(x_2, t)) dt \quad (2.23)$$

para valores $x_1 < x_2$ y $t_1 < t_2$ arbitrarios.

No obstante, suele emplearse una formulación integral alternativa. Esta técnica consiste en multiplicar (2.14) por una función test $\varphi \in C_0^1(\mathbb{R} \times [0, \infty))$, de clase C^1 con soporte compacto, e integrar sobre el dominio de la ecuación:

$$\int_0^\infty \int_{-\infty}^\infty (\varphi u_t + \varphi f(u)_x) dx dt = 0.$$

A continuación, se utiliza integración por partes para eliminar derivadas de u y añadirlas a φ :

$$\int_0^\infty \int_{-\infty}^\infty (\varphi_t u + \varphi_x f(u)) dx dt = - \int_{-\infty}^\infty \varphi(x, 0) u_0(x) dx. \quad (2.24)$$

Se llega de esta forma a la siguiente definición.

Definición 1. Sea $u_0 \in L_{loc}^\infty(\mathbb{R})^1$, se dice que $u \in L_{loc}^\infty(\mathbb{R} \times [0, \infty))^1$ es una solución débil de la ley de conservación (2.16) si cumple (2.24) para toda $\varphi \in C_0^1(\mathbb{R} \times [0, \infty))$.

¹Sea $\Omega \subset \mathbb{R}^n$ abierto, $L_{loc}^\infty(\Omega) = \{u : \Omega \rightarrow \mathbb{R} \text{ medible} : u \in L^\infty(V) \ \forall V \text{ compacto } \subset \Omega\}$. Si Ω es cerrado y $f \in L_{loc}^\infty(\Omega)$, esto quiere decir que f es la restricción a Ω de una función $F \in L_{loc}^\infty(\mathcal{U})$ con \mathcal{U} abierto y $\Omega \subset \mathcal{U}$.

Cabe notar que si u es solución débil de un problema, también lo es cualquier función igual casi por todas partes (cpp) a ella. Además, por construcción es claro que toda solución clásica es solución débil de (2.16). Recíprocamente, una función de clase \mathcal{C}^1 que satisface (2.24) para cualquier φ , también verifica (2.16).

2.3.3. La condición de Rankine-Hugoniot

Obviamente, el concepto de solución débil es mucho más general que el concepto habitual de solución, ya que incluye funciones que cumplen (2.24) pero no son diferenciables o incluso continuas. No obstante, estas funciones deben satisfacer alguna restricción acerca de sus discontinuidades, como se muestra en el presente apartado.

El estudio se restringe a soluciones débiles suaves a trozos. Se dice que una función u definida en $\Omega = \mathbb{R} \times [0, \infty)$ es de clase \mathcal{C}^1 a trozos si existe un número finito de curvas

$$\Gamma_i = \{(\sigma_i(t), t) : t \in I_i\}, \quad i = 1, \dots, n,$$

con $\sigma_i : I_i \rightarrow \mathbb{R}$ de clase \mathcal{C}^1 , tal que u es \mathcal{C}^1 en las componentes conexas de $\Omega - (\cup_{i=1}^n \Gamma_i)$ y presenta discontinuidades de salto en cada una de estas curvas.

Teorema 1. *Sea u una solución débil de (2.16) de clase \mathcal{C}^1 a trozos, sea $\Gamma \equiv x = \sigma(t)$ una de sus curvas de discontinuidad. Dado $P = (x_0, t_0)$ un punto de la curva con $t_0 > 0$, se definen*

$$u^- = \lim_{\varepsilon \rightarrow 0^+} u(x_0 - \varepsilon, t_0), \quad u^+ = \lim_{\varepsilon \rightarrow 0^+} u(x_0 + \varepsilon, t_0), \quad (2.25)$$

los valores de u a la izquierda y derecha de P respectivamente. Entonces, se cumple

$$\frac{f(u^-) - f(u^+)}{u^- - u^+} = \sigma'(t_0), \quad (2.26)$$

expresión conocida como condición de Rankine-Hugoniot o condición RH.

Demostración. Considérese una bola \mathcal{B} centrada en el punto $P = (x_0, t_0)$ suficientemente pequeña de modo que $\mathcal{B} \cap (\mathbb{R} \times \{0\}) = \emptyset$. Sean \mathcal{B}_1 y \mathcal{B}_2 los subconjuntos de \mathcal{B} situados a cada lado de la curva \mathcal{C} . Los puntos intersección de $\partial\mathcal{B}$ con \mathcal{C} se denotan por

$$Q_1 = u(\sigma(t_1), t_1), \quad Q_2 = u(\sigma(t_2), t_2), \quad t_1 < t_2,$$

como se observa en la figura siguiente.

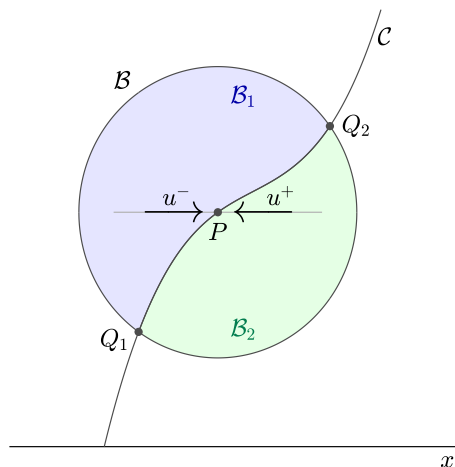


Figura 2.3: Apoyo para la demostración del teorema.

Sea φ una función de clase \mathcal{C}^1 con soporte contenido en \mathcal{B} , como u es solución débil de (2.16) y φ se anula en el eje $\mathbb{R} \times \{0\}$, se tiene

$$\begin{aligned} 0 &= \int_0^\infty \int_{-\infty}^\infty (\varphi_t u + \varphi_x f(u)) dx dt \\ &= \iint_{\mathcal{B}_1} (\varphi_t u + \varphi_x f(u)) dx dt + \iint_{\mathcal{B}_2} (\varphi_t u + \varphi_x f(u)) dx dt. \end{aligned} \quad (2.27)$$

Por otro lado, u es de clase \mathcal{C}^1 en \mathcal{B}_1 y \mathcal{B}_2 , de modo que verifica la ecuación (2.14) en ambos conjuntos. Así, (2.27) equivale a

$$0 = \iint_{\mathcal{B}_1} [(\varphi u)_t + (\varphi f(u))_x] dx dt + \iint_{\mathcal{B}_2} [(\varphi u)_t + (\varphi f(u))_x] dx dt.$$

Aplicando el teorema de Green, usando que $\varphi \equiv 0$ en $\partial\mathcal{B}$ y definiendo

$$u_L(t) = \lim_{\varepsilon \rightarrow 0^+} u(\xi(t) - \varepsilon, t), \quad u_R(t) = \lim_{\varepsilon \rightarrow 0^+} u(\xi(t) + \varepsilon, t),$$

se obtiene

$$\begin{aligned} 0 &= \int_{\partial\mathcal{B}_1^+} \varphi(-u dx + f(u) dt) + \int_{\partial\mathcal{B}_2^+} \varphi(-u dx + f(u) dt) \\ &= \int_{Q_1}^{Q_2} \varphi[-u_L(t) dx + f(u_L(t)) dt] - \int_{Q_1}^{Q_2} \varphi[-u_R(t) dx + f(u_R(t)) dt] \\ &= \int_{Q_1}^{Q_2} \varphi\{-[u_L(t) - u_R(t)] dx + [f(u_L(t)) - f(u_R(t))]\} dt \\ &= \int_{t_1}^{t_2} \varphi(\xi(t), t) \{-[u_L(t) - u_R(t)] \xi'(t) + [f(u_L(t)) - f(u_R(t))]\} dt. \end{aligned}$$

Como la expresión anterior se cumple para cualquier función en el espacio $\mathcal{C}_0^1(\mathcal{B})$, se deduce

$$(u_L(t) - u_R(t)) \xi'(t) = f(u_L(t)) - f(u_R(t))$$

para todo $t \in (t_1, t_2)$, y en particular para $t = t_0$. Así, queda demostrada la relación

$$\frac{f(u^-) - f(u^+)}{u^- - u^+} = \xi'(t_0).$$

□

2.3.4. Cálculo de algunas soluciones débiles

Una vez presentada la condición RH, se enuncia un resultado de caracterización de soluciones débiles que permite obtener algunas de manera sencilla, como se muestra en los ejemplos que le siguen.

Teorema 2. *Sea u una función de clase \mathcal{C}^1 a trozos, es solución débil de (2.16) si y solo si son ciertas las tres afirmaciones siguientes:*

- u verifica la ecuación $u_t + f(u)_x = 0$ en los puntos donde es \mathcal{C}^1 .
- u verifica la condición de Rankine-Hugoniot en los puntos de discontinuidad.
- u cumple la condición inicial $u(x, 0) = u_0(x)$.

Ejemplo 2. Se considera el problema de valor inicial

$$u_t + uu_x = 0 \quad (2.28)$$

$$u_0(x) = \begin{cases} 1, & x < 0, \\ 0, & x > 0. \end{cases} \quad (2.29)$$

Las rectas características vienen dadas por la expresión

$$x(t; x_0) = \begin{cases} t + x_0, & x_0 < 0, \\ x_0, & x_0 > 0. \end{cases}$$

Se representan en la Figura 2.4, en la que también se especifica que la solución debe cumplir $u(x, t) = 1$ si $x < 0$ y $u(x, t) = 0$ si $x > t$.

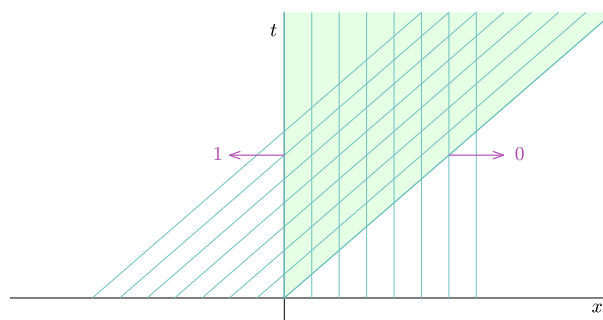


Figura 2.4: Rectas características asociadas a (2.28)-(2.29).

Para definir la solución débil en el área de intersección, es suficiente encontrar una curva $x = \sigma(t)$ compatible con la discontinuidad $u^- = 1$ y $u^+ = 0$ mediante la condición RH, esto es,

$$\sigma'(t) = \frac{f(u^-) - f(u^+)}{u^- - u^+} = \frac{1}{2}.$$

Imponiendo que esta curva pase por el origen, se deduce $\sigma(t) = t/2$. Así, por el Teorema 2, la siguiente función resuelve el problema propuesto:

$$u(x, t) = \begin{cases} 1, & x < t/2, \\ 0, & x > t/2. \end{cases} \quad (2.30)$$

Puede comprobarse, además, que se trata de la única solución débil de (2.28)-(2.29). En la Figura 2.5 se muestran las rectas características asociadas a ella.

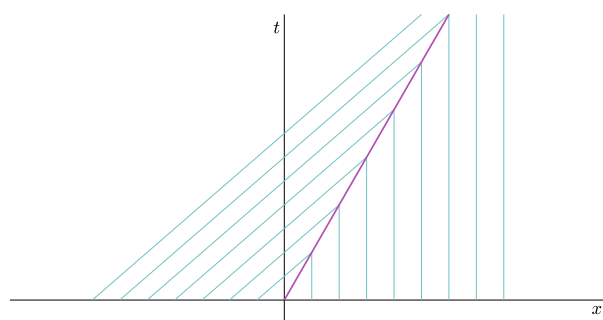


Figura 2.5: Rectas características de la solución (2.30).

Ejemplo 3. Se toma de nuevo la ecuación de Burgers, (2.21), ahora con dato inicial

$$u_0(x) = \begin{cases} 0, & x < 0, \\ 1, & x > 0. \end{cases} \quad (2.31)$$

Las rectas características son de la forma

$$x(t; x_0) = \begin{cases} x_0, & x_0 < 0, \\ t + x_0, & x_0 > 0, \end{cases}$$

algunas de ellas mostradas en la Figura 2.6. En este caso se tiene $u(x, t) = 0$ si $x < 0$, $u(x, t) = 1$ para $x > t$ y una región del semiplano sin cubrir, por lo que debe definirse la solución en este recinto.

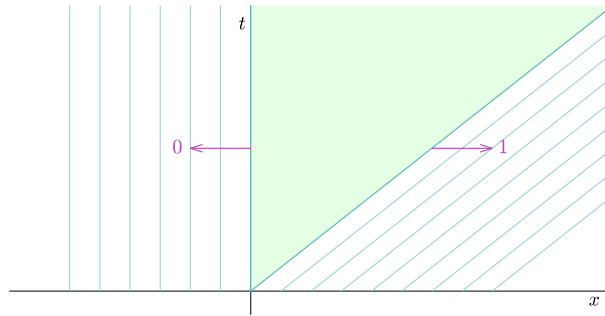


Figura 2.6: Rectas características asociadas a $u_t + uu_x = 0$ con dato inicial (2.31).

Razonando de manera análoga al Ejemplo 2 se obtiene la solución débil

$$u_1(x, t) = \begin{cases} 0, & x < t/2, \\ 1, & x > t/2, \end{cases} \quad (2.32)$$

cuyas características asociadas se muestran en la Figura 2.7.

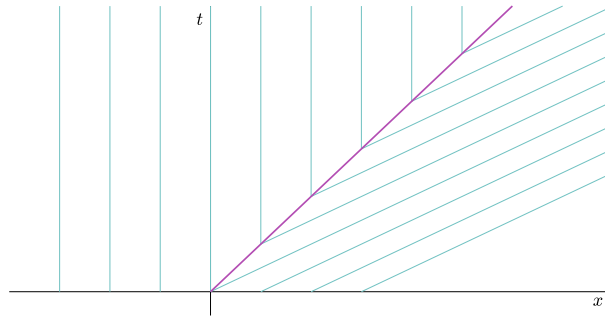


Figura 2.7: Rectas características asociadas a (2.32).

No obstante, (2.32) no es la única solución posible. Por el teorema de caracterización se deduce que la función continua

$$u_2(x, t) = \begin{cases} 0, & x < 0, \\ x/t, & 0 < x < t, \\ 1, & x > t \end{cases}$$

es también solución débil del problema.

En este ejemplo se ha comprobado que algunas leyes de conservación admiten más de una solución débil. Así, para seleccionar la solución físicamente correcta debe imponerse una condición adicional, conocida como condición de entropía, estudiada en el siguiente apartado.

2.3.5. Condición de entropía

Como su nombre indica, este requisito es una versión matemática de un principio fundamental de la termodinámica, según el cual, la entropía de cualquier sistema no decrece.

En la Figura 2.5 se observa que las características asociadas a la solución (2.30) *mueren* en la discontinuidad, mientras que las rectas de (2.32) *nacen* de ésta, en la Figura 2.7. La información, representada por las características, debe crearse en el dato inicial y no en la curva de discontinuidad. Esta discusión heurística toma una expresión matemática en la siguiente definición.

Definición 2. Sea $u = u(x, t)$ una solución débil del problema (2.16), de clase C^1 a trozos. Se dice que u cumple la condición de entropía de Lax si para cada curva de discontinuidad $\mathcal{C} \equiv x = \sigma(t)$ y para cada punto $P = (x_0, t_0)$ de \mathcal{C} , se verifica la relación

$$f'(u^-) > \sigma'(t_0) > f'(u^+), \quad (2.33)$$

siendo u^- y u^+ los valores que toma u a izquierda y derecha de P respectivamente, definidos en (2.25).

Para la ecuación de Burgers, la expresión (2.33) se reduce a $u^- > u^+$. Así, como se avanzaba anteriormente, la solución (2.30) satisface el requisito de entropía, pero se excluye (2.32). A continuación se muestra una definición alternativa de este concepto.

Definición 3. Asímanse las mismas hipótesis que en la definición anterior. Se dice que la solución verifica la condición de entropía de Oleinik si para cada punto $P = (x_0, t_0)$ de cada curva \mathcal{C} , se cumple

$$\frac{f(v) - f(u^-)}{v - u^-} \geq \sigma'(t_0) \geq \frac{f(v) - f(u^+)}{v - u^+} \quad (2.34)$$

para todo v entre u^- y u^+ .

Las condiciones de entropía mencionadas permiten admitir ciertas soluciones débiles y excluir otras. No obstante, se trata de restricciones locales, que solo hacen referencia al comportamiento de las funciones en las discontinuidades. Por esta razón, y con el objetivo de deducir propiedades en todo el dominio, se plantea una condición global equivalente a las estudiadas hasta el momento.

Dado $\varepsilon > 0$ arbitrario, se considera, análogamente al caso lineal, el problema

$$\begin{aligned} u_t + f(u)_x &= \varepsilon u_{xx}, & (x, t) \in \Omega, \\ u(x, 0) &= u_0(x), & x \in \mathbb{R}. \end{aligned} \quad (2.35)$$

Se deduce que la solución clásica de (2.35), denotada por u^ε , equivale a

$$u^\varepsilon = -2\varepsilon \frac{\varphi_x^\varepsilon}{\varphi^\varepsilon},$$

siendo φ^ε solución de

$$\begin{aligned} \varphi_t &= \varepsilon \varphi_{xx}, & (x, t) \in \Omega, \\ \varphi(x, 0) &= \varphi_0^\varepsilon(x), & x \in \mathbb{R}, \end{aligned}$$

para un cierto dato inicial φ_0^ε que depende de u_0 y ε .

Proposición 1. Con la notación anterior, la función $u(x, t) = \lim_{\varepsilon \rightarrow 0^+} u^\varepsilon(x, t)$ es solución débil del problema (2.16). Recibe el nombre de solución *VV*, por sus siglas en inglés, *vanishing viscosity*.

Fijada $\eta : \mathbb{R} \rightarrow \mathbb{R}$ una función convexa suave, se define $q : \mathbb{R} \rightarrow \mathbb{R}$ de la forma

$$q(u) = \int_0^u \eta'(s) f'(s) ds, \quad u \in \mathbb{R}.$$

Es claro que estas funciones verifican la relación $q'(u) = \eta'(u) f'(u)$. Multiplicando la ecuación con viscosidad por $\eta'(u)$, mediante la regla de la cadena se deduce

$$\eta(u^\varepsilon)_t + q(u^\varepsilon)_x = \varepsilon \eta(u^\varepsilon)_{xx} - \varepsilon \eta''(u^\varepsilon) (u_x^\varepsilon)^2.$$

Como $\eta(u)$ es convexa, se obtiene

$$\eta(u^\varepsilon)_t + q(u^\varepsilon)_x \leq \varepsilon \eta(u^\varepsilon)_{xx}.$$

Así, la solución VV cumple

$$\eta(u)_t + q(u)_x \leq 0. \quad (2.36)$$

Esta expresión debe interpretarse en sentido distribucional: para toda función test $\varphi \in C_0^1(\overline{\Omega})$ con $\varphi \geq 0$,

$$\int_0^\infty \int_{-\infty}^\infty [\eta(u)\varphi_t + q(u)\varphi_x] dx dt + \int_{-\infty}^\infty \eta(u_0(x)) \varphi(x, 0) dx \geq 0.$$

El par formado por las funciones η y q recibe el nombre de par de entropía.

Definición 4. Una función $u \in L^\infty(\overline{\Omega})$ recibe el nombre de solución de entropía de (2.16) si:

- u es solución débil de (2.16).
- u verifica (2.36) para todo par de entropía (η, q) .

De nuevo, toda función igual cpp a una solución de entropía es también solución de entropía del problema. Por otro lado, puede comprobarse que una solución débil u es solución de entropía si y solo si

$$|u - c|_t + [\text{sign}(u - c) (f(u) - f(c))]_x \leq 0 \quad (2.37)$$

en sentido distribucional para cualquier constante real c .

Se muestra ahora un resultado de equivalencia entre (2.36) y las condiciones de Lax y Oleinik introducidas en las definiciones 2 y 3 respectivamente.

Teorema 3. Sea $u \in L^\infty(\overline{\Omega})$ una solución débil de (2.16) de clase C^1 a trozos. Las siguientes afirmaciones son equivalentes:

- u es una solución de entropía de (2.16).
- En cada curva de discontinuidad $\mathcal{C} \equiv x = \sigma(t)$ y en cada punto $P = (x_0, t_0)$,

$$q(u^+) - q(u^-) \leq \sigma'(t_0)(u^+ - u^-)$$

para todo par de entropía (η, q) .

- u verifica la condición de entropía de Oleinik.
- Si f es cóncava o convexa, u cumple la condición de entropía de Lax.

Finalmente, se enuncia un resultado de unicidad en el que también se incluyen algunas propiedades que la solución satisface.

Teorema 4. *Considérese el problema (2.16) para una función $f \in \mathcal{C}^1$ y $u_0 \in L^1(\mathbb{R}) \cap L^\infty(\mathbb{R})$. Entonces, la solución VV es la única (excepto variaciones en un conjunto de medida nula) solución de entropía de (2.16) y verifica*

- $\|u(\cdot, t)\|_{L^\infty} \leq \|u_0\|_{L^\infty} \quad \forall t > 0.$
- Si $\|u_0\|_{TV} < \infty$, entonces

$$\|u(\cdot, t)\|_{TV} \leq \|u_0\|_{TV} \quad \forall t > 0.^1 \quad (2.38)$$

- Si u y v son las soluciones de (2.16) con datos iniciales u_0 y v_0 , entonces

$$u_0(x) \leq v_0(x) \quad \forall x \in \mathbb{R} \quad \longrightarrow \quad u(x, t) \leq v(x, t) \quad \forall x, t.$$

2.3.6. Soluciones de entropía de los problemas de Riemann

En este último apartado se muestran las soluciones VV del problema de Riemann

$$\begin{aligned} u_t + f(u)_x &= 0, \quad (x, t) \in \Omega, \\ u_0(x) &= \begin{cases} u_l, & x < 0, \\ u_r, & x > 0, \end{cases} \end{aligned} \quad (2.39)$$

con $u_l \neq u_r$, ya que serán de gran utilidad a la hora de diseñar métodos numéricos para resolver (2.16).

Si $u_l > u_r$, como se ha estudiado para la ecuación de Burgers, la solución de entropía es

$$u(x, t) = \begin{cases} u_l, & x < st, \\ u_r, & x > st, \end{cases} \quad (2.40)$$

donde el valor del parámetro s viene dado por la condición de Rankine-Hugoniot.

Para el caso $u_l < u_r$, no obstante, deben distinguirse dos casos. En primer lugar, si f es una función estrictamente convexa, esta solución es

$$u(x, t) = \begin{cases} u_l, & x < f'(u_l)t, \\ (f')^{-1}(x/t), & f'(u_l)t < x < f'(u_r)t, \\ u_r, & x > f'(u_r)t. \end{cases} \quad (2.41)$$

Por otro lado, para una función f arbitraria, la solución de entropía debe ser (2.41) cambiando f por la mayor función convexa menor o igual que ésta.

¹Dada una función real g definida en $[a, b]$ y una partición $\mathcal{P} = \{a = x_0 < x_1 < \dots < x_{N_{\mathcal{P}}} = b\}$ de este intervalo, se toma

$$g_{\mathcal{P}} = \sum_{j=0}^{N_{\mathcal{P}}} |g(x_{j+1}) - g(x_j)|.$$

Así, la variación total de g se define como la seminorma

$$\|g\|_{TV} = \sup_{\mathcal{P}} (g_{\mathcal{P}}).$$

Capítulo 3

Métodos numéricos para la ecuación lineal

Tras estudiar las ecuaciones hiperbólicas escalares desde el punto de vista teórico, se ha comprobado que tan solo pueden resolverse analíticamente en casos muy concretos. Por esta razón, a lo largo del trabajo se muestran distintos métodos numéricos con el fin de aproximar su solución.

Como punto de partida, no obstante, en este capítulo se presentan y analizan métodos para la ecuación lineal, cuya solución exacta sí se conoce. Así, se considera el problema

$$\begin{aligned} u_t + au_x &= 0, & (x, t) \in \Omega = \mathbb{R} \times (0, \infty), \\ u(x, 0) &= u_0(x), & x \in \mathbb{R}, \end{aligned} \quad (3.1)$$

siendo a una constante distinta de cero. El estudio podría generalizarse para una función $a = a(x, t)$ cualquiera.

3.1. Introducción

Aunque el dominio espacial de (3.1) es la recta real, para diseñar estos métodos es preciso restringirlo a un intervalo acotado. En este intervalo, denotado por $[x_L, x_R]$, se toma una partición uniforme:

$$\mathcal{P}_x = \{x_L = x_0, x_1, \dots, x_{J-1}, x_J = x_R\}, \quad (3.2)$$

siendo $x_j = x_L + j\Delta x$ con $\Delta x = (x_R - x_L)/J$, representada en la Figura 3.1.

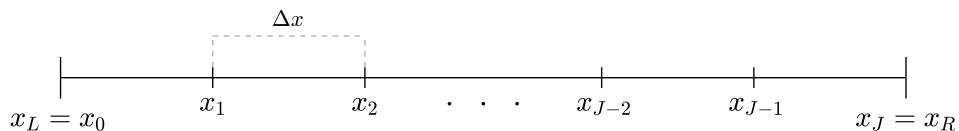


Figura 3.1: Partición del dominio espacial.

Por otro lado, fijado un tiempo final T , se divide $[0, T]$ en N subintervalos iguales, dando lugar a la partición

$$\mathcal{P}_t = \{0 = t_0, t_1, \dots, t_{N-1}, t_N = T\}, \quad (3.3)$$

con $t_n = n\Delta t$ y $\Delta t = T/N$.

Sea $u = u(x, t) = u_0(x - at)$ la solución de (3.1), el objetivo reside en obtener una aproximación del valor que toma u en cada punto (x_j, t_n) del malla, denotada por u_j^n . Mediante la condición inicial se fija

$$u_j^0 = u_0(x_j), \quad j = 0, \dots, J.$$

A lo largo de la sección se plantean distintas expresiones que permiten calcular los valores u_j^{n+1} para $j = 0, \dots, J$ a partir de las estimaciones u_j^n a tiempo t_n .

3.2. Conceptos básicos

Antes de presentar métodos numéricos que aproximen la solución del problema, se muestran algunas propiedades que éstos deben cumplir. En primer lugar, se dice que un método es convergente si el error global

$$e_j^n = u_j^n - u(x_j, t_n)$$

se anula cuando Δx y Δt tienden a 0. Aunque se trata de un concepto sencillo, es muy costoso comprobar por definición si los métodos son convergentes, ya que vienen dados por una relación de recurrencia y normalmente no se tiene la expresión explícita de u_j^n .

Por otro lado, para un método numérico

$$u_j^{n+1} + G(u_j^n, \dots) = 0, \quad (3.4)$$

el error local T_j^n se define como el resultado de sustituir en (3.4) cada estimación u_j^n por el valor exacto $u(x_j, t_n)$, dividiendo por Δt :

$$T_j^n = \frac{1}{\Delta t} [u(x_j, t_{n+1}) + G(u(x_j, t_n), \dots)].$$

Análogamente, un método es consistente si el error local se anula cuando Δx y Δt tienden a 0, hecho que garantiza que el método estime la ecuación diferencial correcta. Finalmente se trata el concepto de estabilidad.

Intuitivamente, se dice que un método es estable si las soluciones numéricas son próximas cuando los datos iniciales lo son. Para la definición formal, suponiendo que el dato inicial puede escribirse como la suma de su serie de Fourier,

$$u_0(x) = \sum_{m=-\infty}^{\infty} a_m e^{im\pi x}, \quad (3.5)$$

la solución de (3.1) viene dada por

$$u(x, t) = \sum_{m=-\infty}^{\infty} a_m e^{im\pi(x-at)} = \sum_{m=-\infty}^{\infty} a_m e^{-im\pi at} e^{im\pi x},$$

y en los puntos del mallado:

$$u(x_j, t_n) = \sum_{m=-\infty}^{\infty} a_m (e^{-im\pi a \Delta t})^n e^{im\pi j \Delta x}.$$

Para obtener la solución de la ecuación del método, se toma $\gamma_j^n = \lambda^n e^{im\pi j \Delta x}$ y se impone que esta expresión verifique dicha ecuación. Una vez hallado el valor $\lambda = \lambda(m)$, se deduce

$$u_j^n = \sum_{m=-\infty}^{\infty} a_m \lambda(m)^n e^{im\pi j \Delta x}.$$

Finalmente, el método propuesto es estable si

$$|\lambda(m)| \leq 1$$

para cualquier valor de m . De esta forma, si

$$(u_j^n)^1 = \sum_{m=-\infty}^{\infty} a_m^1 \lambda(m)^n e^{im\pi j \Delta x}, \quad (u_j^n)^2 = \sum_{m=-\infty}^{\infty} a_m^2 \lambda(m)^n e^{im\pi j \Delta x}$$

son las soluciones que proporciona el método para el problema (3.1) con datos iniciales distintos, se verifica

$$\left| (u_j^n)^1 - (u_j^n)^2 \right| \leq \sum_{m=-\infty}^{\infty} |a_m^1 - a_m^2|.$$

Cabe recordar que los coeficientes a_m^i solo dependen de la función inicial $u_0(x)$, como se observa en (3.5).

Los tres conceptos se relacionan mediante el teorema de equivalencia de Lax. Este resultado afirma que en todo método consistente, la estabilidad es una condición necesaria y suficiente para su convergencia. De esta forma, puede verificarse esta última propiedad de manera sencilla, como se muestra en los apartados 3.4 y 3.5.

3.3. La condición CFL

En esta sección se estudia una propiedad exclusiva de los métodos numéricos para las ecuaciones hiperbólicas de leyes de conservación: la condición CFL. Para ello es necesario introducir dos definiciones acerca de la relación que guarda la condición inicial con la solución de la ecuación y del método.

Es conocido que la solución del problema (3.1) viene dada por la función

$$u(x, t) = u_0(x - at). \quad (3.6)$$

Cabe notar que la imagen de (3.6) en un punto del mallado $P = (x_j, t_n)$ solo depende del valor de u_0 en $Q = (x_j - at_n, 0)$. De esta forma, el dominio de dependencia de la ecuación se define como

$$\mathcal{D}_e(x_j, t_n) = \{x_j - at_n\}.$$

Por otro lado, el valor u_j^n que proporciona el método se calcula a partir de dos o tres estimaciones a tiempo $t = t_{n-1}$, como u_{j-1}^{n-1} , u_j^{n-1} , u_{j+1}^{n-1} , ... Recursivamente, u_j^n depende del valor que toma la función inicial u_0 en distintos puntos del mallado espacial: $\mathcal{M}_j = \{x_{j-k}, \dots, x_j, \dots, x_{j+r}\}$. Con esta notación, el dominio de dependencia del método viene dado por la envoltura convexa del conjunto \mathcal{M}_j , es decir, el intervalo

$$\mathcal{D}_m(x_j, t_n) = [x_{j-k}, x_{j+r}].$$

Definición 5. *Un método numérico para resolver (3.1) cumple la condición de Courant-Friedrichs-Lax o condición CFL si*

$$\mathcal{D}_e(x_j, t_n) \subset \mathcal{D}_m(x_j, t_n), \quad (3.7)$$

es decir, si el dominio de dependencia del método contiene al dominio de dependencia de la ecuación.

A continuación se construye un esquema numérico sencillo, se calcula su dominio de dependencia y se comprueba si éste verifica (3.7).

Ejemplo 4. *Siguiendo el procedimiento comúnmente utilizado para resolver ecuaciones diferenciales numéricamente, se propone aproximar las derivadas parciales de la ecuación $u_t + au_x = 0$ por diferencias finitas. Sustituyendo u_t por una diferencia hacia delante y la derivada espacial por una diferencia hacia atrás, se obtiene el método*

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0.$$

Denotando $\nu = a\Delta x/\Delta t$, la expresión anterior se reescribe como

$$u_j^{n+1} = (1 - \nu)u_j^n + \nu u_{j-1}^n. \quad (3.8)$$

El valor u_j^n se calcula a partir de u_{j-1}^{n-1} y u_j^{n-1} , por lo que depende del valor de u_0 en los puntos

$$x_{j-n}, x_{j-n+1}, \dots, x_{j-1}, x_j$$

siempre y cuando $x_L \leq x_{j-n}$. El dominio de dependencia de este método es, por tanto, el intervalo $[x_{j-n}, x_j]$.

Es claro que la condición CFL no se cumple para valores de a negativos. Para $a > 0$, siguiendo la Figura 3.2, esta condición se verifica si y solo si

$$\text{pendiente } m \leq \text{pendiente } r \quad \equiv \quad a \frac{\Delta t}{\Delta x} \leq 1. \quad (3.9)$$

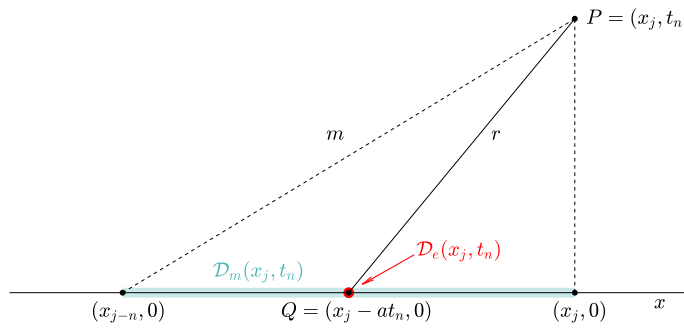


Figura 3.2: Dominios de dependencia para el método (3.8). Los términos m y r hacen referencia a las rectas que pasan por P y los puntos $(x_{j-n}, 0)$ y Q respectivamente.

Dado un esquema que no verifica esta condición y un punto $P = (x_j, t_n)$ del mallado, se modifica el valor de u_0 solamente en $x_0 = x_j - at_n$. Por tanto, también se modifica la solución en la recta $r \equiv x = at + x_0$, y en particular en el punto P . No obstante, como x_0 no pertenece a $\mathcal{D}_m(x_j, t_n)$, el esquema seguiría proporcionando la misma estimación u_j^n , de modo que no sería convergente. Así, se ha comprobado que (3.7) es una condición necesaria para la convergencia de un método.

Por último se muestra que no se trata de una condición suficiente. Si se aproxima ahora u_x por una diferencia centrada se obtiene

$$u_j^{n+1} = u_j^n - \frac{\nu}{2}(u_{j+1}^n - u_{j-1}^n), \quad \nu = a \frac{\Delta t}{\Delta x}. \quad (3.10)$$

Razonando de manera análoga al ejemplo 4, se verifica (3.7) si y solo si $|a|\Delta t/\Delta x \leq 1$. Por otro lado, sustituyendo $\gamma_j^n = \lambda^n e^{im\pi j\Delta x}$ en (3.10) se deduce $\lambda(m) = 1 - \nu i \sin(m\pi\Delta x)$ y, por tanto,

$$|\lambda(m)|^2 = 1 + \nu^2 \sin^2(m\pi\Delta x), \quad m \in \mathbb{Z}.$$

Esto implica que el método propuesto no es estable y, por el teorema de equivalencia de Lax, tampoco convergente.

En los apartados restantes del capítulo se presentan dos esquemas numéricos convergentes para la ecuación de advección lineal: el método upwind y el método de Lax-Wendroff. Se analizan con detalle sus propiedades y se llevan a cabo distintos experimentos numéricos para comprobar su precisión.

3.4. El método upwind

Cabe notar que el método estudiado en el ejemplo 4 cumple la condición CFL para $a > 0$ si Δx y Δt verifican la relación $a\Delta t/\Delta x \leq 1$. De esta forma, puede aprovecharse (3.8) para $a > 0$, y debe generarse una nueva ecuación para valores negativos de este parámetro. Aproximando u_x por una diferencia hacia delante en este caso, se deduce la expresión del método upwind:

$$u_j^{n+1} = (1 + \nu)u_j^n - \nu u_{j+1}^n, \quad a < 0, \quad (3.11)$$

$$u_j^{n+1} = (1 - \nu)u_j^n + \nu u_{j-1}^n, \quad a > 0, \quad (3.12)$$

con $\nu = a\Delta t/\Delta x$. Denotando $a^+ = \max(a, 0)$ y $a^- = \min(a, 0)$, estas dos ecuaciones pueden agruparse en

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} a^+ (u_j^n - u_{j-1}^n) - \frac{\Delta t}{\Delta x} a^- (u_{j+1}^n - u_j^n), \quad a \in \mathbb{R}.$$

Nota 1. Para el caso $a > 0$, conocidos los valores u_j^n para $j = 0, \dots, J$, mediante (3.12) pueden hallarse las estimaciones u_j^{n+1} para $j = 1, \dots, J$. El valor u_0^{n+1} , por su parte, puede tomarse como u_j^{n+1} (condición de frontera artificial) o u_1^{n+1} (condición periódica), según el ejemplo considerado.

Si $a < 0$, se calcularían u_j^{n+1} , $j = 0, \dots, J-1$, usando (3.11), y se fijaría u_j^{n+1} como u_{j-1}^{n+1} (condición artificial) o bien u_0^{n+1} (condición periódica).

En cuanto a la condición CFL, para $a < 0$ puede razonarse como en el ejemplo 4, y se deduce que ésta se satisface si $-a\Delta t/\Delta x \leq 1$. Así, el método upwind cumple la condición CFL si y solo si $|\nu| \leq 1$.

Por otro lado, sustituyendo u_j^n por $\gamma_j^n = \lambda^n e^{im\pi j \Delta x}$ en las ecuaciones (3.11)-(3.12), se obtiene

$$|\lambda(m)|^2 = \begin{cases} 1 + 4\nu(1 + \nu) \sin^2\left(\frac{m\pi\Delta x}{2}\right), & a < 0, \\ 1 - 4\nu(1 - \nu) \sin^2\left(\frac{m\pi\Delta x}{2}\right), & a > 0. \end{cases}$$

Se concluye que el método es estable si $|\nu| \leq 1$, idéntica restricción a la impuesta en la condición CFL. La discusión mostrada a continuación se realiza para $a > 0$, se obtienen resultados análogos si $a < 0$.

En primer lugar se considera un dato inicial suficientemente suave. En ese caso, la función $u(x, t) = u_0(x - at)$ es solución clásica del problema (3.1), de la misma clase que u_0 . Así pues, mediante el desarrollo en serie de Taylor se comprueba que el error local viene dado por

$$T_j^n = -\frac{1}{2}a(1 - \nu)u_{xx}\Delta x + \mathcal{O}(\Delta x^2), \quad (3.13)$$

de orden 1 respecto de Δx . Aplicando el teorema de equivalencia de Lax se deduce que el método es convergente si $|\nu| \leq 1$. Además, siguiendo (3.12) y la definición de error global,

$$e_j^{n+1} = (1 - \nu)e_j^n + \nu e_{j-1}^n - \Delta t \cdot T_j^n.$$

Finalmente, denotando $E^n = \max_j |e_j^n|$ y $\bar{T} = \max_{j,n} |T_j^n|$ se tiene

$$E^{n+1} \leq E^n + \Delta t \cdot \bar{T} \quad \longrightarrow \quad E^N \leq T \cdot \bar{T}. \quad (3.14)$$

Por otro lado, si u_0 presenta alguna singularidad, se sigue tomando $u(x, t) = u_0(x - at)$ como solución de la ley de conservación. Esta función tan solo verifica la forma integral (2.24) y no admite

derivadas en todo su dominio. Por esta razón, no es posible emplear las técnicas anteriores para deducir el orden de convergencia, por lo que se propone un estudio basado en análisis de Fourier.

Cabe recordar que la solución de (3.1) en un punto (x_j, t_n) se expresa como

$$u(x_j, t_n) = \sum_{m=-\infty}^{\infty} a_m \xi(m)^n e^{im\pi j \Delta x},$$

siendo $\xi(m) = e^{-im\pi a \Delta t}$, con $|\xi(m)|^2 = 1$ y fase o argumento $\arg(\xi(m)) = -m\pi a \Delta t$. Por otro lado, la solución de (3.12) viene dada por

$$u_j^n = \sum_{m=-\infty}^{\infty} a_m \lambda(m)^n e^{im\pi j \Delta x},$$

con $\lambda(m) = 1 - \nu + \nu \cos(m\pi \Delta x) - i\nu \sin(m\pi \Delta x)$. Su módulo cumple

$$|\lambda(m)|^2 = |\xi(m)|^2 - \nu(1 - \nu)m^2\pi^2\Delta x^2 + \dots, \quad (3.15)$$

cometiendo de esta forma un error de orden 2 en $\varepsilon = m\pi \Delta x$, que será significativo cuando m sea suficientemente grande.

Por su parte, el argumento se estima como

$$\arg(\lambda(m)) = -\arctan\left(\frac{\nu \sin(m\pi \Delta x)}{1 - \nu + \nu \cos(m\pi \Delta x)}\right) = -\nu m\pi \Delta x \left(1 - \frac{1}{6}(1 - \nu)(1 - 2\nu)m^2\pi^2\Delta x^2 + \dots\right),$$

pues se tiene un error relativo también de orden 2 en ε :

$$\frac{\arg(\lambda(m)) - \arg(\xi(m))}{\arg(\xi(m))} = -\frac{1}{6}(1 - \nu)(1 - 2\nu)m^2\pi^2\Delta x^2 + \dots \quad (3.16)$$

Cabe notar que los errores calculados anteriormente se anulan para $\nu = 1$. Así, en ese caso se obtiene la solución exacta, siempre y cuando la elección de las condiciones de frontera no altere el resultado.

A continuación se aplica el método upwind para resolver el problema (3.1) con $a = 1$ y distintos datos iniciales.

Ejemplo 5. Se toma la función 1-periódica $u_0(x) = \sin(2\pi x)$. Así, se considera el intervalo $[0, 1]$ como dominio espacial y se fijan condiciones de contorno periódicas. Se elige $\Delta x = 0.01$ y, para garantizar la estabilidad del método, $\nu = \Delta t / \Delta x = 0.8$.

Se muestra el resultado obtenido para $T = 1$ en la Figura 3.3. Se observa que la aproximación reproduce perfectamente la velocidad de la onda, por (3.16), mientras que comete un ligero error en su amplitud, véase (3.15), fenómeno conocido como amortiguación.

Para cada $\Delta x > 0$ se define el error

$$E^{\Delta x} = \Delta x \sum_{j=0}^J |u_j^N - u(x_j, T)|, \quad (3.17)$$

cuyo orden se aproxima mediante los valores

$$p^{\Delta x} = \log_2(E^{2\Delta x} / E^{\Delta x}). \quad (3.18)$$

Los resultados obtenidos para este problema se muestran en la Tabla 3.1, donde se observa que el orden coincide con la unidad, al menos para J pequeños, como cabía esperar tras (3.13).

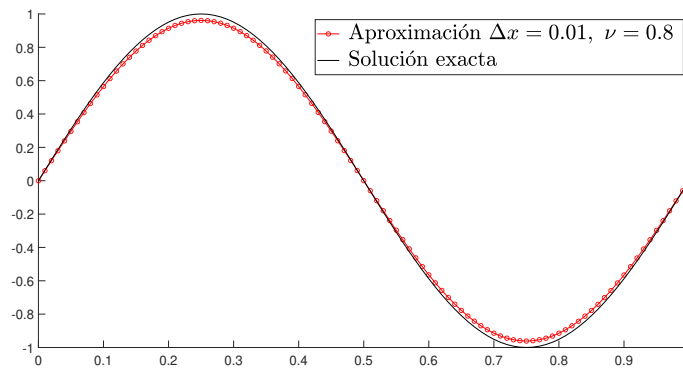


Figura 3.3: Aproximación numérica obtenida con el método upwind y solución exacta de $u_t + u_x = 0$ con $u_0(x) = \sin(2\pi x)$.

Δx	$E^{\Delta x}$	$p^{\Delta x}$
1/5	$6.15 \cdot 10^{-1}$	—
1/10	$2.85 \cdot 10^{-1}$	1.1081
1/20	$1.15 \cdot 10^{-1}$	1.3153
1/40	$6 \cdot 10^{-2}$	0.9361
1/80	$3.07 \cdot 10^{-2}$	0.9664
1/160	$1.55 \cdot 10^{-2}$	0.9827

Tabla 3.1: Error y orden aproximado del método upwind para $u_t + u_x = 0$, $u_0(x) = \sin(2\pi x)$.

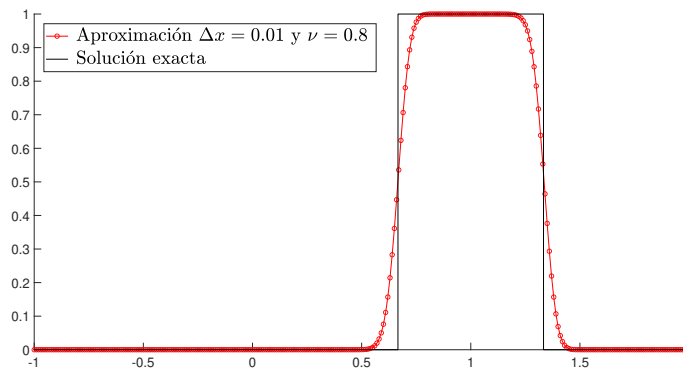


Figura 3.4: Aproximación numérica obtenida con el método upwind y solución exacta del problema $u_t + u_x = 0$, $u_0 = \mathbb{1}_{(-\frac{1}{3}, \frac{1}{3})}$.

Ejemplo 6. Para el siguiente ejemplo se toma $u_0 = \mathbb{1}_{(-1/3, 1/3)}$, dominio espacial $[-1, 2]$, condiciones de frontera artificiales, $\Delta x = 0.01$ y de nuevo $\nu = 0.8$. Para $T = 1$ se obtiene la aproximación de la Figura 3.4.

El error de orden 2 cometido por $|\lambda(m)|^2$ provoca que la aproximación sea redondeada en las esquinas, alrededor de las discontinuidades de salto. Por otro lado, el soporte de la estimación se desplaza a una velocidad muy similar a la de la solución, y no presenta oscilación alguna debido a la cota (3.14). En la Tabla 3.2 se muestra que el orden del error $E^{\Delta x}$ conseguido en este caso es 0.5.

Por último se presenta una manera alternativa de deducir (3.12), que será de gran utilidad a la hora de diseñar el método numérico del siguiente apartado. Una vez calculadas las aproximaciones

Δx	$E^{\Delta x}$	$p^{\Delta x}$
1/40	$1.95 \cdot 10^{-1}$	—
1/80	$1.39 \cdot 10^{-1}$	0.4909
1/160	$9.95 \cdot 10^{-2}$	0.4803
1/320	$7.17 \cdot 10^{-2}$	0.4722
1/640	$4.91 \cdot 10^{-2}$	0.5475
1/1280	$3.46 \cdot 10^{-2}$	0.5062
1/2560	$2.44 \cdot 10^{-2}$	0.5004

Tabla 3.2: Error y orden aproximado del método upwind para $u_t + u_x = 0$, $u_0(x) = \mathbb{1}_{(-\frac{1}{3}, \frac{1}{3})}$.

para $t = t_n$, se considera la función lineal $p = p(x)$ que verifica $p(x_{j-1}) = u_{j-1}^n$ y $p(x_j) = u_j^n$:

$$p(x) = u_{j-1}^n \frac{x_j - x}{x_j - x_{j-1}} + u_j^n \frac{x - x_{j-1}}{x_j - x_{j-1}}.$$

Como se refleja en la Figura 3.5, se toma el punto $Q = (q, t_n)$ perteneciente a la misma recta característica que $P = (x_j, t_{n+1})$. Como $u(P) = u(Q)$, se propone escoger

$$u_j^{n+1} = p(q) = u_{j-1}^n \frac{\overline{AQ}}{\Delta x} + u_j^n \frac{\overline{BQ}}{\Delta x} = u_{j-1}^n \frac{\overline{AQ}}{\Delta x} + u_j^n \left(1 - \frac{\overline{AQ}}{\Delta x}\right), \quad (3.19)$$

donde A y B se definen en la Figura 3.5, y la notación \overline{MN} hace referencia a la longitud del segmento de extremos M y N .

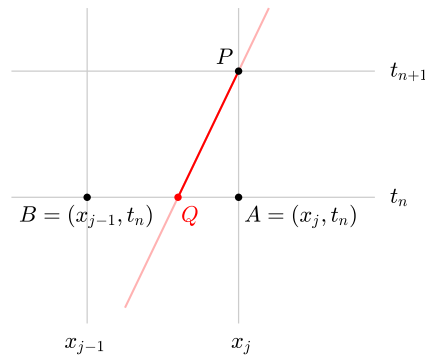


Figura 3.5: Manera alternativa de deducir el método upwind.

Teniendo en cuenta que la pendiente de la recta característica coincide con $1/a$, se deduce fácilmente que $\overline{AQ}/\Delta x = \nu$. Así, (3.19) equivale a

$$u_j^{n+1} = (1 - \nu)u_j^n + \nu u_{j-1}^n,$$

la expresión del método upwind correspondiente a $a > 0$.

3.5. El método de Lax-Wendroff

Para soluciones clásicas, el método upwind permite obtener aproximaciones de primer orden, como se ha estudiado en el análisis de la consistencia y comprobado en los ejemplos numéricos. Por otro lado, aunque el error (3.16) en la fase es relativamente pequeño, la expresión (3.15) puede provocar estimaciones poco precisas cerca de las singularidades o que presentan la amortiguación observada en la Figura 3.3. Con el objetivo de evitar estos resultados y aumentar el orden de

convergencia, en este apartado se presenta el esquema de Lax-Wendroff.

Se deduce su ecuación para el caso $a > 0$, pero se obtiene el mismo resultado si $a < 0$. Se toma como referencia la discusión mostrada al final de la sección anterior para obtener la expresión del método upwind. Así, conocidas las estimaciones a tiempo $t = t_n$, se considera el polinomio $p = p(x)$ de segundo grado que verifica

$$p(x_{j-1}) = u_{j-1}^n, \quad p(x_j) = u_j^n, \quad p(x_{j+1}) = u_{j+1}^n.$$

Éste polinomio puede escribirse de la forma

$$p(x) = u_{j-1}^n \frac{(x_j - x)(x_{j+1} - x)}{(x_j - x_{j-1})(x_{j+1} - x_{j-1})} + u_j^n \frac{(x - x_{j-1})(x - x_{j+1})}{(x_j - x_{j-1})(x_j - x_{j+1})} + u_{j+1}^n \frac{(x - x_{j-1})(x - x_j)}{(x_{j+1} - x_{j-1})(x_{j+1} - x_j)}.$$

De nuevo se fija el punto $Q = (q, t_n)$ que pertenece a la misma recta característica que $P = (x_j, t_{n+1})$ y se escoge $u_j^{n+1} = p(q)$. Según la Figura 3.6,

$$u_j^{n+1} = u_{j-1}^n \frac{\overline{AQ} \cdot \overline{CQ}}{2\Delta x^2} + u_j^n \frac{\overline{QB} \cdot \overline{CQ}}{\Delta x^2} - u_{j+1}^n \frac{\overline{QB} \cdot \overline{AQ}}{2\Delta x^2}.$$

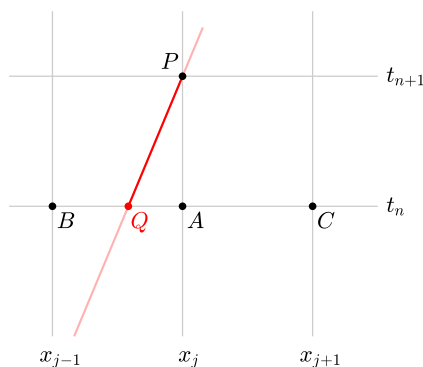


Figura 3.6: Deducción de la ecuación del método de Lax-Wendroff.

Como $\overline{AQ} = a\Delta t = \nu\Delta x$, $\overline{QB} = (1 - \nu)\Delta x$ y $\overline{CQ} = (1 + \nu)\Delta x$, se concluye

$$u_j^{n+1} = \frac{\nu(1 + \nu)}{2} u_{j-1}^n + (1 - \nu^2) u_j^n + \frac{\nu(\nu - 1)}{2} u_{j+1}^n. \quad (3.20)$$

Esta ecuación puede reescribirse como

$$u_j^{n+1} = u_j^n - \frac{\nu}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{\nu^2}{2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \quad \nu = a \frac{\Delta t}{\Delta x}, \quad a \in \mathbb{R}. \quad (3.21)$$

Nota 2. Para este método una vez calculados los valores $u_0^n, u_1^n, \dots, u_{j-1}^n, u_j^n$, se fijan u_{-1}^n y u_{j+1}^n siguiendo una de estas estrategias:

- Condiciones de frontera artificiales: $u_{-1}^n = u_1^n, u_{j+1}^n = u_{j-1}^n$.
- Condiciones de frontera periódicas: $u_{-1}^n = u_{j-1}^n, u_{j+1}^n = u_1^n$.

A continuación se aplica (3.20) para hallar $u_0^{n+1}, \dots, u_j^{n+1}$.

Es claro que la condición CFL se cumple si y solo si $|\nu| \leq 1$. Por otro lado, sustituyendo u_j^n por $\gamma_j^n = \lambda^n e^{im\pi j \Delta x}$ en (3.20) se obtiene

$$\lambda(m) = 1 - 2\nu^2 \sin^2\left(\frac{m\pi\Delta x}{2}\right) - i\nu \sin(m\pi\Delta x). \quad (3.22)$$

Este factor cumple

$$|\lambda(m)|^2 = 1 - 4\nu^2(1 - \nu^2) \sin^4\left(\frac{m\pi\Delta x}{2}\right), \quad (3.23)$$

pues el método es estable sii, de nuevo, $|\nu| \leq 1$.

Si u_0 es suficientemente suave, mediante el desarrollo en serie de Taylor se deduce que el error local es de orden 2 en Δx :

$$T_j^n = \frac{1}{6}a(1 - \nu^2)u_{xxx}\Delta x^2 + \mathcal{O}(\Delta x^2). \quad (3.24)$$

Por estabilidad y el teorema de Lax, se tiene la convergencia del método para $|\nu| \leq 1$. No obstante, en esta ocasión no es posible hallar una cota para el error global similar a (3.14).

Si, por el contrario, el dato inicial presenta alguna singularidad, se recurre al análisis de Fourier. Por (3.22) y el desarrollo de la función seno,

$$|\lambda(m)|^2 = |\xi(m)|^2 - \frac{1}{4}\nu^2(1 - \nu^2)(m\pi\Delta x)^4 + \dots$$

Así, se comete un error de orden 4 en $\varepsilon = m\pi\Delta x$, frente al orden 2 del método upwind. Puede comprobarse que el argumento de $\lambda(m)$, por su parte, verifica

$$\frac{\arg(\lambda(m)) - \arg(\xi(m))}{\arg(\xi(m))} = -\frac{1}{6}(1 - \nu^2)(m\pi\Delta x)^2 + \dots,$$

del mismo orden en ε pero ligeramente superior al del caso anterior.

Ejemplo 7. Por último, se aplica el método a los mismos problemas planteados en el apartado anterior. Para $u_0(x) = \sin(2\pi x)$ y parámetros $\nu = 0.8$ y $\Delta x = 0.02$, se obtiene la aproximación de la Figura 3.7. Los valores $E^{\Delta x}$ y $p^{\Delta x}$ para distintos Δx se muestran en la Tabla 3.3. Cabe notar que, en este caso, el orden estimado es 2, como en (3.24), al menos para valores de J pequeños.

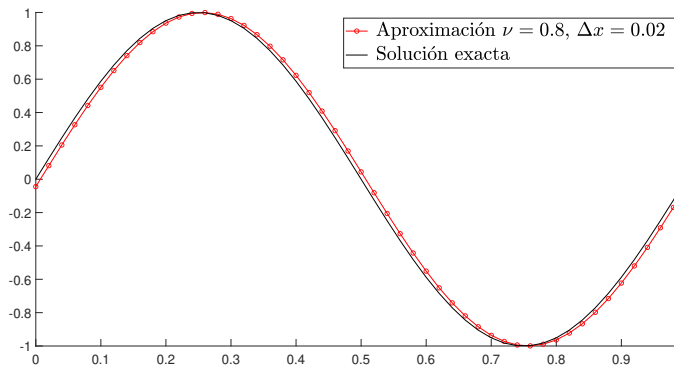


Figura 3.7: Aproximación numérica obtenida con el método de Lax-Wendroff y solución exacta del problema $u_t + u_x = 0$, $u_0(x) = \sin(2\pi x)$.

Al reducirse el error en el módulo, la amplitud de la estimación obtenida es casi idéntica a la de la solución. No obstante, como el error en el argumento ha aumentado ligeramente, no se logra

Δx	$E^{\Delta x}$	$p^{\Delta x}$
1/5	$2.98 \cdot 10^{-1}$	—
1/10	$8.75 \cdot 10^{-2}$	1.7649
1/20	$2.53 \cdot 10^{-2}$	1.7873
1/40	$6.15 \cdot 10^{-3}$	2.0435
1/80	$1.51 \cdot 10^{-3}$	2.0259
1/160	$3.74 \cdot 10^{-4}$	2.0137

Tabla 3.3: Error y orden aproximado del método de Lax-Wendroff para $u_t + u_x = 0$, $u_0(x) = \sin(2\pi x)$.

reproducir demasiado bien la velocidad de la onda.

Por otro lado, para $u_0 = \mathbb{1}_{(-1/3, 1/3)}$, la aproximación presenta oscilaciones cerca de los puntos de discontinuidad, como se muestra en la Figura 3.8. Esto se debe a que no se puede aplicar el teorema de Lax y no se tiene una cota como (3.14) para este método. A pesar de ello, parece que la estimación no es tan redondeada en las esquinas como el método upwind.

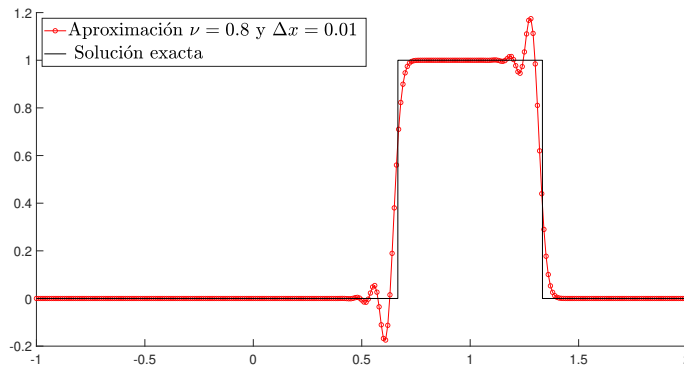


Figura 3.8: Aproximación mediante el método de Lax-Wendroff y solución exacta del problema $u_t + uu_x = 0$, $u_0 = \mathbb{1}_{(-1/3, 1/3)}$.

Capítulo 4

Métodos de volúmenes finitos

Una vez tratado el caso lineal, durante los dos capítulos siguientes se estudian métodos numéricos para la ecuación escalar no lineal

$$u_t + f(u)_x = 0. \quad (4.1)$$

En el capítulo 2 se muestra que las soluciones de las leyes de conservación hiperbólicas pueden presentar singularidades en tiempo finito. De esta forma, no puede aplicarse el desarrollo en serie de Taylor para sustituir las derivadas de (4.1) por diferencias finitas. Por el contrario, los esquemas deben diseñarse desde una perspectiva distinta, a partir de la estructura de estas ecuaciones y sus soluciones.

4.1. Introducción

El objetivo del capítulo es obtener una aproximación de la solución de entropía de

$$\begin{aligned} u_t + f(u)_x &= 0, & (x, t) &\in \mathbb{R} \times (0, \infty), \\ u(x, 0) &= u_0(x), & x &\in \mathbb{R}. \end{aligned} \quad (4.2)$$

Como se ha mencionado anteriormente, para la resolución numérica es preciso restringir (4.2) a un dominio acotado. Así, se asume $x \in [x_L, x_R]$ y $t \in [0, T]$ para ciertos valores x_L , x_R y T fijados. De nuevo, deben imponerse condiciones adicionales adecuadas en las rectas $x = x_L$ y $x = x_R$.

Sea $J \in \mathbb{N}$ arbitrario, se definen los valores

$$x_j = x_L + \left(j + \frac{1}{2}\right) \Delta x, \quad j = 0, \dots, J$$

con $\Delta x = (x_R - x_L) / (J + 1)$. Por otro lado, se fija

$$x_{-\frac{1}{2}} = x_L, \quad x_{j+\frac{1}{2}} = x_j + \frac{\Delta x}{2}, \quad j = 0, \dots, J.$$

Estos puntos dividen el intervalo $[x_L, x_R]$ en subintervalos de longitud Δx :

$$c_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}], \quad j = 0, \dots, J, \quad (4.3)$$

conocidos como celdas o volúmenes de control. En la Figura 4.1 se representa el intervalo $[x_L, x_R]$, los distintos puntos en él considerados y las celdas c_j definidas en (4.3). En ocasiones es necesario definir celdas c_k o nodos x_k para valores de k no contemplados en la figura, pero éstos pueden definirse de manera análoga.

En cuanto al mallado en la variable temporal, se toma $t_0 = 0$ y $t_{n+1} = t_n + \Delta t_n$ hasta obtener $t_N = T$ para cierto natural N . El parámetro Δt_n es una cantidad fija o variable dependiendo del método numérico en cuestión, como se indica posteriormente.

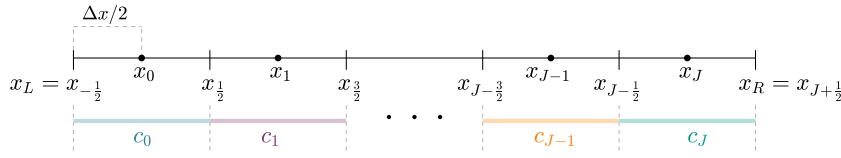


Figura 4.1: Construcción de los volúmenes de control en el intervalo $[x_L, x_R]$.

4.2. Ecuación general

Como se ha avanzado en el inicio del capítulo, deben aproximarse soluciones no necesariamente suaves, funciones para las que no tiene sentido su evaluación en los puntos de discontinuidad, por ejemplo. Por esta razón, no se trata de estimar el valor de la solución en los puntos (x_j, t_n) del mallado, sino de obtener aproximaciones de los promedios

$$I_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_n) dx,$$

denotadas por u_j^n .

Mediante la condición inicial se define

$$u_j^0 = I_j^0 = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u_0(x) dx, \quad j = 0, \dots, J.$$

A continuación se discute cómo generar las estimaciones u_j^{n+1} a partir de las obtenidas para cierto instante $t = t_n$.

En primer lugar se integra la ecuación (4.1) sobre el dominio $c_j \times [t_n, t_{n+1}]$:

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_{n+1}) dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_n) dx = - \int_{t_n}^{t_{n+1}} f(u(x_{j+\frac{1}{2}}, t)) dt + \int_{t_n}^{t_{n+1}} f(u(x_{j-\frac{1}{2}}, t)) dt. \quad (4.4)$$

Esta expresión, la forma integral (2.23), está bien definida porque no puede haber discontinuidad en una recta vertical, según la condición de Rankine-Hugoniot. Definiendo

$$\bar{f}_{j+\frac{1}{2}}^n = \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} f(u(x_{j+\frac{1}{2}}, t)) dt \quad (4.5)$$

y dividiendo (4.4) por Δx , se deduce

$$I_j^{n+1} = I_j^n - \frac{\Delta t_n}{\Delta x} (\bar{f}_{j+\frac{1}{2}}^n - \bar{f}_{j-\frac{1}{2}}^n). \quad (4.6)$$

Cabe notar que los términos \bar{f} no son conocidos, pues dependen de la propia solución a calcular. Por esta razón, deben hallarse unas aproximaciones de $\bar{f}_{j+\frac{1}{2}}^n$, denotadas por $\hat{f}_{j+\frac{1}{2}}^n$, y definir los valores u_j^{n+1} como

$$u_j^{n+1} = u_j^n - \frac{\Delta t_n}{\Delta x} (\hat{f}_{j+\frac{1}{2}}^n - \hat{f}_{j-\frac{1}{2}}^n), \quad j = 0, \dots, J. \quad (4.7)$$

4.3. Deducción de algunos métodos numéricos

Seguindo el desarrollo de la sección anterior, la clave para diseñar los distintos métodos reside en estimar de manera adecuada las expresiones (4.5), como se estudia en el presente apartado.

4.3.1. Método de Godunov

Fijado $t = t_n$ para cierto $n \in \mathbb{N}$, se suponen calculadas las aproximaciones u_j^n en cada celda c_j , $j = 0, \dots, J$.

Nota 3. En primer lugar se determinan los valores u_{-1}^n y u_{J+1}^n siguiendo una de las dos estrategias siguientes:

- Condiciones periódicas: $u_{-1}^n = u_J^n$, $u_{J+1}^n = u_0^n$.
- Condiciones artificiales: $u_{-1}^n = u_0^n$, $u_{J+1}^n = u_J^n$.

Así, para cada $j \in \{-1, \dots, J\}$ se define el problema de Riemann

$$\begin{aligned} u_t + f(u)_x &= 0, \\ u(x, t_n) &= \begin{cases} u_j^n, & x < x_{j+\frac{1}{2}}, \\ u_{j+1}^n, & x > x_{j+\frac{1}{2}}, \end{cases} \end{aligned} \quad (4.8)$$

cuya solución se detalla en el apartado 2.3.6. Debe tenerse en cuenta que las rectas de discontinuidad de las soluciones de problemas vecinos no pueden intersectar para $t < t_{n+1} = t_n + \Delta t_n$. Como el módulo de las pendientes de estas rectas está acotado por $\max_j |f'(u_j^n)|$, se toma Δt_n tal que

$$\max_j |f'(u_j^n)| \frac{\Delta t_n}{\Delta x} \leq \frac{1}{2}. \quad (4.9)$$

Como en el caso lineal, la expresión (4.9) recibe el nombre de condición CFL.

Por otro lado, cabe notar que la solución del problema (4.8), denotada por $\bar{u}_{j+\frac{1}{2}}^n$, verifica

$$\bar{u}_{j+\frac{1}{2}}^n(x, t) = \bar{v}_{j+\frac{1}{2}}^n \left(\frac{x - x_{j+\frac{1}{2}}}{t - t_n} \right)$$

para cierta función $\bar{v}_{j+\frac{1}{2}}^n$. Así, el flujo $\bar{f}_{j+\frac{1}{2}}^n$ puede estimarse de la siguiente manera:

$$\begin{aligned} \bar{f}_{j+\frac{1}{2}}^n &= \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} f(u(x_{j+\frac{1}{2}}, t)) dt \approx \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} f(\bar{u}_j^n(x_{j+\frac{1}{2}}, t)) dt = \\ &= \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} f(\bar{v}_{j+\frac{1}{2}}^n(0)) dt = f(\bar{v}_{j+\frac{1}{2}}^n(0)). \end{aligned}$$

Usando $\hat{f}_{j+\frac{1}{2}}^n = f(v_{j+\frac{1}{2}}(0))$ en la ecuación (4.7) se obtiene el método de Godunov.

Una fórmula explícita para esta aproximación de $\bar{f}_{j+\frac{1}{2}}^n$ viene dada por

$$f(\bar{v}_{j+\frac{1}{2}}^n(0)) = \hat{f}^{God}(u_j^n, u_{j+1}^n) = \begin{cases} \min_{u_j^n \leq \theta \leq u_{j+1}^n} f(\theta), & u_j^n \leq u_{j+1}^n, \\ \max_{u_{j+1}^n \leq \theta \leq u_j^n} f(\theta), & u_{j+1}^n < u_j^n. \end{cases} \quad (4.10)$$

Según esta expresión, el método de Godunov coincide con el método upwind cuando $f(u) = au$. Además, si f no presenta máximos locales y un único mínimo en ω , en particular si f es estrictamente convexa, esta fórmula se reduce a

$$\hat{f}^{God}(u_j^n, u_{j+1}^n) = \max [f(\max(u_j^n, \omega)), f(\min(u_{j+1}^n, \omega))].$$

A continuación se muestran dos ejemplos de aplicación de este método.

Ejemplo 8. Se propone resolver la ecuación de Burgers:

$$u_t + uu_x = 0 \quad (4.11)$$

con dato inicial

$$u_0(x) = \begin{cases} 1, & x < 0, \\ 0, & x > 0. \end{cases} \quad (4.12)$$

Se fija $[-1, 1]$ como dominio espacial, tiempo final $T = 1$ y condiciones de frontera artificiales. Se consideran 50 celdas c_j y, para cada $n \in \mathbb{N}$, se toma, según (4.9), $t_{n+1} = t_n + \Delta t_n$ con

$$\Delta t_n = \frac{\Delta x}{2 \max_j |u_j^n|}.$$

La aproximación obtenida se muestra en la Figura 4.2 junto con la solución exacta del problema:

$$u(x, 1) = \begin{cases} 1, & x < 1/2, \\ 0, & x > 1/2. \end{cases}$$

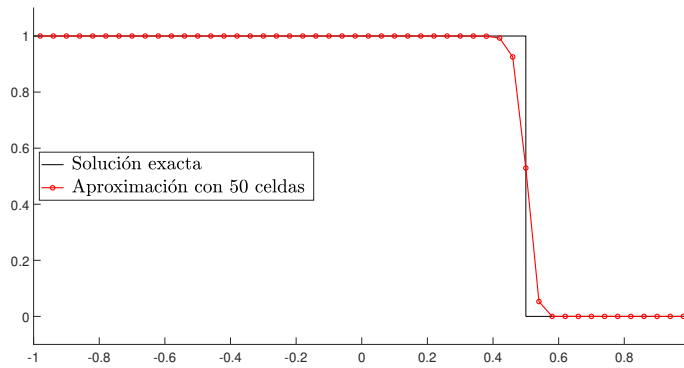


Figura 4.2: Aproximación obtenida con el método de Godunov con 50 celdas y solución exacta de (4.11)-(4.12).

Ejemplo 9. En este caso se pretende hallar la solución de (4.11) con la función inicial

$$u_0(x) = \begin{cases} -1, & x < 0, \\ 1, & x > 0, \end{cases} \quad (4.13)$$

en el mismo intervalo $[-1, 1]$, con $T = 0.5$ e idénticas condiciones de frontera. De nuevo con 50 celdas y Δt_n según (4.9), la aproximación se representa en la Figura 4.3. Esta vez, la solución exacta en $T = 0.5$ viene dada por

$$u(x, 0.5) = \begin{cases} -1, & x < -0.5, \\ 2x, & -0.5 < x < 0.5, \\ 1, & x > 0.5. \end{cases}$$

En ambos casos se obtienen estimaciones de las soluciones de entropía, cuyas discontinuidades se aproximan bastante bien. No obstante, este esquema no puede generalizarse para sistemas de ecuaciones ya que, como se muestra en el capítulo 6, no se tiene una expresión explícita de la solución de los problemas de Riemann. Además, la fórmula (4.10) para calcular $\hat{f}_{j+\frac{1}{2}}^n$ conlleva un coste computacional considerable. Finalmente, cabe notar que resolver (4.8) para emplear solamente el valor de la solución en $\{x_{j+\frac{1}{2}}\} \times (t_n, t_{n+1})$ parece un tanto innecesario. Por estos motivos, se trata de hallar una alternativa para estimar (4.5).

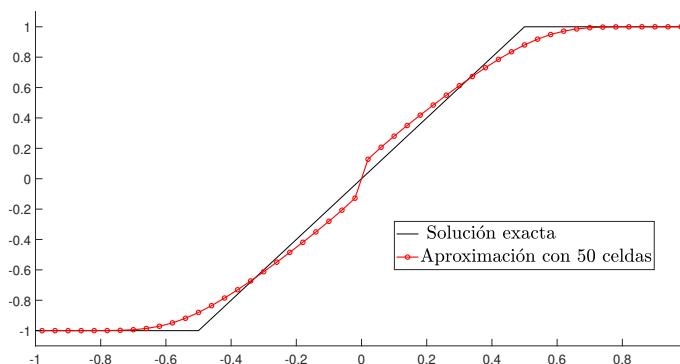


Figura 4.3: Aproximación obtenida con el método de Godunov con 50 celdas y solución exacta de (4.11)-(4.13).

4.3.2. Método de Roe

En primer lugar se propone linealizar la ecuación diferencial en los problemas de Riemann (4.8). De esta forma, se sustituyen por

$$\begin{aligned} u_t + a_{j+\frac{1}{2}}^n u_x &= 0 \\ u(x, t_n) &= \begin{cases} u_j^n, & x < x_{j+\frac{1}{2}}, \\ u_{j+1}^n, & x > x_{j+\frac{1}{2}}, \end{cases} \end{aligned} \quad (4.14)$$

siendo

$$a_{j+\frac{1}{2}}^n = \begin{cases} \frac{f(u_{j+1}^n) - f(u_j^n)}{u_{j+1}^n - u_j^n}, & u_j^n \neq u_{j+1}^n, \\ f'(u_j^n), & u_j^n = u_{j+1}^n. \end{cases}$$

La solución de (4.14) se obtiene mediante el método de las características, pero de nuevo es preciso asegurarse de que las rectas de discontinuidad de problemas vecinos no intersecan. Así, fijado Δx es suficiente tomar Δt_n de modo que se cumpla la condición

$$\max_j |a_{j+\frac{1}{2}}^n| \frac{\Delta t_n}{\Delta x} \leq \frac{1}{2}. \quad (4.15)$$

Análogamente al método de Godunov, la aproximación de (4.5) será la imagen por f del valor de la solución de (4.14) en el segmento $\{x_{j+\frac{1}{2}}\} \times (t_n, t_{n+1})$:

$$\hat{f}_{j+\frac{1}{2}}^n = f^{Roe}(u_j^n, u_{j+1}^n) = \begin{cases} f(u_j^n), & a_{j+\frac{1}{2}}^n \geq 0, \\ f(u_{j+1}^n), & a_{j+\frac{1}{2}}^n < 0. \end{cases} \quad (4.16)$$

Por tanto, la ecuación del método de Roe viene dada por

$$u_j^{n+1} = u_j^n - \frac{\Delta t_n}{\Delta x} (\hat{f}_{j+\frac{1}{2}}^n - \hat{f}_{j-\frac{1}{2}}^n), \quad j = 0, \dots, J, \quad (4.17)$$

calculando los flujos $\hat{f}_{j+\frac{1}{2}}^n$ según (4.16) y teniendo en cuenta la nota 3. El coste computacional asociado a este método es muy bajo, pero falta comprobar si permite resolver correctamente las leyes de conservación, como se analiza a continuación.

Ejemplo 10. Se consideran los problemas de los ejemplos 8 y 9 del apartado anterior. Los resultados se incluyen en la Figura 4.4.

Cabe notar que en el primer caso se genera la misma estimación que con el método de Godunov. No obstante, para el dato inicial (4.13) no se calcula una aproximación de la solución de entropía,

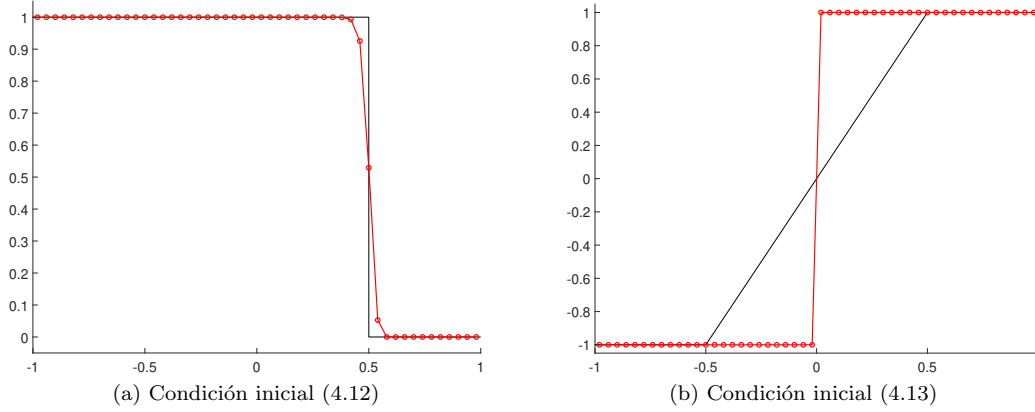


Figura 4.4: Aproximaciones con el método de Roe con 50 celdas y soluciones exactas de la ecuación de Burgers.

y ésta no se consigue aunque se tome un mallado más fino. Este hecho ocurre porque la solución de la linealización de (4.8) solo admite una recta de discontinuidad, mientras que la solución del problema tiene dos.

Tras este intento fallido se propone un método que aproxima la solución VV con un coste computacional inferior al del método de Godunov.

4.3.3. Método de Rusanov

La clave para diseñar un método robusto y eficiente reside en encontrar una buena aproximación de la solución de los problemas de Riemann (4.8). A diferencia del método de Roe, ésta se sustituye directamente por

$$\bar{w}_{j+\frac{1}{2}}^n(x, t) = \begin{cases} u_j^n, & x - x_{j+\frac{1}{2}} < -s_{j+\frac{1}{2}}(t - t_n), \\ u_{j+\frac{1}{2}}^*, & -s_{j+\frac{1}{2}}(t - t_n) < x - x_{j+\frac{1}{2}} < s_{j+\frac{1}{2}}(t - t_n), \\ u_{j+1}^n, & x - x_{j+\frac{1}{2}} > s_{j+\frac{1}{2}}(t - t_n), \end{cases}$$

una función con dos discontinuidades que se desplazan a velocidades $-s_{j+\frac{1}{2}}$ y $s_{j+\frac{1}{2}}$.

Aplicando la condición de Rankine-Hugoniot en cada una de ellas, usando $f_{j+\frac{1}{2}}^*$ en lugar de $f(u_{j+\frac{1}{2}}^*)$, se deduce

$$f_{j+\frac{1}{2}}^* = \frac{f(u_j^n) + f(u_{j+1}^n)}{2} - \frac{s_{j+\frac{1}{2}}}{2}(u_{j+1}^n - u_j^n). \quad (4.18)$$

En este caso se escoge $\hat{f}_{j+\frac{1}{2}}^n = f_{j+\frac{1}{2}}^*$, ya que en el método de Godunov se ha fijado $\hat{f}_{j+\frac{1}{2}}^n$ como la imagen por f del valor de la solución de (4.8) en $\{x_{j+\frac{1}{2}}\} \times (t_n, t_{n+1})$.

Es suficiente determinar el valor de la velocidad $s_{j+\frac{1}{2}}$. Una primera opción sería considerar el máximo que ésta puede tomar, $\Delta x / (2\Delta t)$, pero no se obtendrían resultados muy satisfactorios, al no tener en cuenta el problema tratado. Por otro lado, en el método de Rusanov se propone

$$s_{j+\frac{1}{2}} = \text{máx}(|f'(u_j^n)|, |f'(u_{j+1}^n)|).$$

Así, este método consiste en calcular u_j^{n+1} por medio de la ecuación (4.17) con

$$\hat{f}_{j+\frac{1}{2}}^n = \hat{f}^{Rus}(u_j^n, u_{j+1}^n) = \frac{1}{2} [f(u_j^n) + f(u_{j+1}^n) - \text{máx}(|f'(u_j^n)|, |f'(u_{j+1}^n)|) \cdot (u_{j+1}^n - u_j^n)]. \quad (4.19)$$

Como en los casos anteriores, para establecer el valor de u_{-1}^n y u_{j+1}^n se sigue la nota 3. También se utiliza la condición (4.9) para determinar Δt_n una vez fijado Δx .

Ejemplo 11. *En este ejemplo se aplica el método de Rusanov para resolver los problemas tratados anteriormente.*

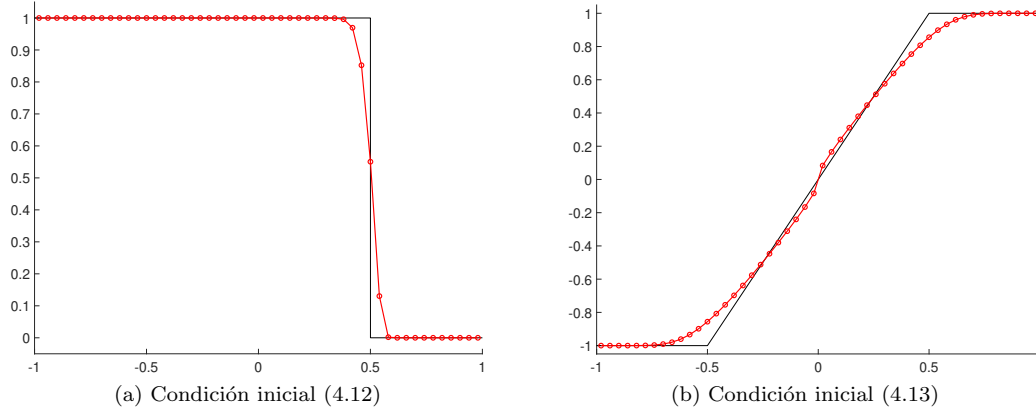


Figura 4.5: Aproximación obtenida con el método de Rusanov con 50 celdas y solución exacta de $u_t + uu_x = 0$.

En la Figura 4.5 se observa que este método permite estimar de nuevo las soluciones de entropía de ambos problemas. Cabe notar que las aproximaciones cerca de las discontinuidades son casi tan buenas como las obtenidas con el método de Godunov. Además, el coste computacional sería notablemente inferior si f fuese una función con algún máximo local, ya que se calcularía (4.19) en lugar de (4.10).

4.4. Análisis de la convergencia

En la sección anterior se han presentado tres métodos numéricos para tratar de resolver el problema (4.2). Dos de ellos, Godunov y Rusanov, permiten hallar estimaciones de la solución VV, mientras que la linealización de Roe puede proporcionar soluciones débiles físicamente incorrectas. A continuación se estudian las propiedades que un método debe cumplir para que esto último no ocurra.

De ahora en adelante se supone que las aproximaciones u_j^n se calculan siguiendo una expresión de la forma

$$u_j^{n+1} = H(u_{j-p}^n, \dots, u_j^n, \dots, u_{j+q}^n) \quad (4.20)$$

con $H(0, \dots, 0) = 0$.

Definición 6. *Sin tener en cuenta las condiciones de frontera, se dice que un método (4.20) es conservativo si*

$$\sum_j u_j^{n+1} = \sum_j u_j^n$$

para todo n .

Puede comprobarse que esta propiedad se cumple si y solo si existe una función \hat{f} tal que (4.20) puede escribirse como

$$\begin{aligned}
u_j^{n+1} &= H(u_{j-p}^n, \dots, u_j^n, \dots, u_{j+q}^n) \\
&= u_j^n - \frac{\Delta t}{\Delta x} \left[\hat{f}(u_{j-r}^n, \dots, u_{j+q}^n) - \hat{f}(u_{j-p}^n, \dots, u_{j+s}^n) \right]
\end{aligned} \tag{4.21}$$

con $r \leq p$, $s \leq q$ y $q + r = p + s$, como ocurre en los casos tratados anteriormente.

Definición 7. Dado un método conservativo, se dice que es consistente si verifica

$$\hat{f}(u, \dots, u) = f(u) \tag{4.22}$$

para cualquier número real u .

Las funciones \hat{f} asociadas a los métodos estudiados verifican la condición (4.22), incluyendo al método de Roe. Así pues, debe definirse otro criterio de estabilidad que, al menos, diferencie la linealización de Roe de las otras dos estrategias.

Para ello, se recuerda que dos soluciones u y v de (4.2) con datos iniciales u_0 y v_0 respectivamente satisfacen

$$u(x, t) \leq v(x, t) \quad \forall x, t \quad \text{si} \quad u_0(x) \leq v_0(x) \quad \forall x \in \mathbb{R},$$

propiedad presentada en el teorema 4. De esta forma, los esquemas numéricos deben cumplir una cierta propiedad de monotonía que garantice la obtención de la solución correcta.

Definición 8. Un método (4.20) es monótono si la función H es creciente en cada una de sus componentes.

Mediante una caracterización de este concepto en términos de \hat{f} , se comprueba que los métodos de Godunov y Rusanov son monótonos, no así el de Roe. Así, se ha hallado una característica que separa los métodos robustos de aquellos que no lo son.

Cabe recordar que, de nuevo por el teorema 4, la solución VV de (4.2), denotada por u , verifica

$$\|u(\cdot, t)\|_{L^\infty} \leq \|u_0\|_{L^\infty} \quad \forall t > 0.$$

Análogamente puede comprobarse que las estimaciones obtenidas mediante un método consistente y monótono cumplen

$$\min_k u_k^0 \leq u_j^n \leq \max_k u_k^0 \quad \forall n, j.$$

Por otro lado, estas aproximaciones también satisfacen una discretización de (2.38),

$$\sum_j |u_{j+1}^{n+1} - u_j^{n+1}| \leq \sum_j |u_{j+1}^n - u_j^n|, \tag{4.23}$$

y de la condición de entropía (2.37).

Estas propiedades refuerzan la hipótesis de que los esquemas consistentes y monótonos permiten calcular la solución de entropía de la ley de conservación, como se enuncia en el siguiente resultado.

Teorema 5. Considérese el problema (4.2) con dato inicial $u_0 \in L^\infty(\mathbb{R})$, y un método numérico con ecuación (4.21) consistente y monótono, siendo \hat{f} diferenciable o con continuidad Lipschitz local. Sean u_j^n las estimaciones calculadas para $\Delta x > 0$ arbitrario, se define la función

$$u^{\Delta x}(x, t) = u_j^n, \quad (x, t) \in [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}) \times [t_n, t_{n+1}).$$

Asúmase que las funciones $u^{\Delta x}$

- son acotadas: para cada $\Delta x > 0$ existe $C > 0$ tal que

$$\|u^{\Delta x}\|_{L^\infty(\mathbb{R} \times \mathbb{R}^+)} \leq C.$$

- convergen en $L^1_{loc}(\mathbb{R} \times \mathbb{R}^+)$ a una cierta función u cuando $\Delta x, \Delta t \rightarrow 0$.

Entonces, la función u es la solución VV de (4.2). Cabe notar que la hipótesis de monotonía puede sustituirse por la relación (4.23).

Una vez conocidos qué métodos permiten aproximar la solución débil correcta, es importante saber a qué velocidad las distintas estimaciones convergen a ella. Así, fijado $t > 0$, con las hipótesis del teorema anterior se tiene

$$\|u^{\Delta x} - u\|_{L^1(\mathbb{R})} \leq C\Delta x, \quad C > 0$$

si la solución es suave. No obstante, para el resto de casos se ha probado la cota

$$\|u^{\Delta x} - u\|_{L^1(\mathbb{R})} \leq C\sqrt{\Delta x}, \quad C > 0.$$

Estos resultados se ven reflejados en los experimentos numéricos realizados para el método upwind, ya que éste equivale al método de Godunov cuando la ecuación es lineal. Así, para la solución suave, en la Tabla 3.1 se observa que el orden de convergencia es 1. Por otro lado, en la solución que presenta discontinuidades, la Tabla 3.2 recoge que el orden es 0.5.

A continuación se estima el orden de convergencia del método de Rusanov aplicado a la ecuación de Burgers con dos funciones iniciales distintas.

Ejemplo 12. En primer lugar se considera la ecuación $u_t + uu_x = 0$ en el intervalo $[0, 5\pi/2]$, y se toma la función inicial

$$u_0(x) = \begin{cases} 2, & x \leq \pi/2, \\ \sin(x) + 1, & \pi/2 < x < 3\pi/2, \\ 0, & x \geq 3\pi/2. \end{cases} \quad (4.24)$$

Seguindo la fórmula (2.20) se deduce que el tiempo crítico asociado al problema es $t_c = 1$. Así, para $t = 0.1$, la solución será una función de clase C^1 , por lo que se espera que el orden coincida con la unidad.

Como no es posible calcular la solución exacta, para hallar el error asociado a un determinado número de celdas $J+1$, y su correspondiente Δx , se toma J_{ref} grande tal que $J_{ref}+1$ sea múltiplo de $J+1$. Sea $u^{ref} = (u_0^{ref}, \dots, u_{J_{ref}}^{ref})$ el vector de aproximaciones con $J_{ref}+1$ celdas, y u , el vector obtenido con $J+1$ celdas, se toma:

$$E^{\Delta x} = \Delta x \|u - \bar{u}^{ref}\|_1 = \Delta x \sum_{i=0}^J |u_i - \bar{u}_i^{ref}|, \quad \bar{u}_j^{ref} = \frac{1}{R} \sum_{i=1}^R u_{Rj+i-1}^{ref}, \quad R = \frac{J_{ref}+1}{J+1}.$$

El orden, por su parte, se obtiene siguiendo la fórmula (3.18).

Así, fijando $J_{ref} = 25599$, se obtienen los resultados de la Tabla 4.1, donde se observa que el orden estimado es próximo a 1.

Ejemplo 13. Finalmente, se considera la ecuación de Burgers con dato inicial (4.13). El orden obtenido mediante el método de Godunov, en la Tabla 4.2, resulta cercano a 0.8.

J	$E^{\Delta x}$	$p^{\Delta x}$
49	$1.53 \cdot 10^{-2}$	—
99	$7.15 \cdot 10^{-3}$	1.1006
199	$3.51 \cdot 10^{-3}$	1.0274
399	$1.72 \cdot 10^{-3}$	1.0246
799	$8.43 \cdot 10^{-4}$	1.0335
1599	$4.07 \cdot 10^{-4}$	1.0512
3199	$1.89 \cdot 10^{-4}$	1.1032

Tabla 4.1: Error y orden aproximado del método de Rusanov para $u_t + uu_x = 0$ y condición inicial (4.24).

J	$E^{\Delta x}$	$p^{\Delta x}$
49	$7.27 \cdot 10^{-2}$	—
99	$3.33 \cdot 10^{-2}$	0.7111
199	$2.7 \cdot 10^{-2}$	0.7191
399	$1.61 \cdot 10^{-2}$	0.7456
799	$9.42 \cdot 10^{-3}$	0.7719
1599	$5.43 \cdot 10^{-3}$	0.7957
3199	$3.08 \cdot 10^{-3}$	0.8163

Tabla 4.2: Error y orden aproximado del método de Rusanov para $u_t + uu_x = 0$ y condición inicial (4.13).

4.5. Reconstrucción espacial de segundo orden

En las secciones restantes del capítulo se presentan distintas estrategias con el fin de aumentar el orden de convergencia de los métodos estudiados hasta el momento. Dichos métodos pueden reescribirse de la siguiente manera:

1. Una vez calculadas las estimaciones u_j^n a tiempo $t = t_n$, se define la función constante a trozos

$$v_n(x) = u_j^n, \quad x \in c_j, \quad j = -1, \dots, J+1. \quad (4.25)$$

2. A continuación, se evalúa esta función en los extremos de cada celda c_j , para obtener

$$u_{j+\frac{1}{2}}^{n-} = v_n(x_{j+\frac{1}{2}}), \quad u_{j-\frac{1}{2}}^{n+} = v_n(x_{j-\frac{1}{2}}).$$

3. Finalmente, se toma

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} \left[\hat{f}(u_{j+\frac{1}{2}}^{n-}, u_{j+\frac{1}{2}}^{n+}) - \hat{f}(u_{j-\frac{1}{2}}^{n-}, u_{j-\frac{1}{2}}^{n+}) \right], \quad j = 0, \dots, J, \quad (4.26)$$

siendo $\hat{f} = \hat{f}^{God}$ o $\hat{f} = \hat{f}^{Rus}$, funciones definidas en (4.10) y (4.19) respectivamente.

Para soluciones suaves de (4.2), denotadas por $u = u(x, t)$, cabe notar que la función definida en (4.25) verifica

$$v_n - u(\cdot, t_n) = \mathcal{O}(\Delta x).$$

Con el objetivo de hallar aproximaciones de orden superior, en este paso se propone tomar funciones lineales a trozos p_n que cumplan puntualmente

$$p_n - u(\cdot, t_n) = \mathcal{O}(\Delta x^2).$$

Denotando

$$p_n(x) = p_{n,j}(x), \quad x \in c_j$$

y teniendo en cuenta que u_j^n es una estimación del promedio

$$\frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_n) dx,$$

se impone que $p_{n,j}$ verifique la relación

$$\frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p_{n,j}(x) dx = u_j^n.$$

Así, se deduce que $p_{n,j}$ debe tomar la forma

$$p_{n,j}(x) = u_j^n + \sigma_j^n (x - x_j),$$

para un cierto parámetro σ_j^n por determinar. La clave para diseñar métodos numéricos robustos reside en precisar el valor de dicha pendiente.

Imponiendo además la condición

$$\frac{1}{\Delta x} \int_{x_{j+\frac{1}{2}}}^{x_{j+\frac{3}{2}}} p_{n,j}(x) dx = u_{j+1}^n, \quad (4.27)$$

se obtendría

$$\sigma_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x}. \quad (4.28)$$

No obstante, para la ecuación lineal, el esquema resultante coincidiría con el método de Lax-Wendroff, presentado en el capítulo anterior. Cabe recordar que las estimaciones que este método proporciona pueden sufrir oscilaciones cuando las soluciones presentan alguna discontinuidad, como se muestra en la Figura 3.8. Por otro lado, se obtienen resultados análogos si se toma

$$\sigma_j^n = \frac{u_j^n - u_{j-1}^n}{\Delta x} \quad \text{o} \quad \sigma_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{\Delta x}.$$

Con el fin de evitar estas oscilaciones, teniendo en cuenta que la solución de entropía del problema satisface

$$\|u(\cdot, t)\|_{TV} \leq \|u_0\|_{TV} \quad t > 0, \quad (4.29)$$

los métodos numéricos deberían cumplir una propiedad similar a (4.29) a nivel discreto. Así, recordando la función v_n definida en (4.25), se elige σ_j^n tal que la función lineal a trozos p_n verifique

$$\|p_n\|_{TV} \leq \|v_n\|_{TV} = \sum_{j=0}^{J-1} |u_{j+1}^n - u_j^n|. \quad (4.30)$$

Un posible valor para el parámetro σ_j^n sería

$$\sigma_j^n = \min\text{mod} \left(\frac{u_{j+1}^n - u_j^n}{\Delta x}, \frac{u_j^n - u_{j-1}^n}{\Delta x} \right), \quad (4.31)$$

con

$$\min\text{mod}(a_1, a_2) = \begin{cases} \text{sign}(a_1) \min(|a_1|, |a_2|), & \text{si } \text{sign}(a_1) = \text{sign}(a_2), \\ 0, & \text{en otro caso.} \end{cases}$$

4.6. Formulación semidiscreta

En el apartado anterior se ha estudiado una técnica de reconstrucción que permite obtener una aproximación de orden dos a nivel espacial, más precisa que la estimación de orden uno asociada al método de Godunov o Rusanov. No obstante, cabe notar que en (4.26) está presente la discretización

$$g_t(t_n) = \frac{g(t_{n+1}) - g(t_n)}{\Delta t_n} + \mathcal{O}(\Delta t_n),$$

por lo que el esquema resultante sigue siendo de primer orden.

Para resolver esta problemática, en esta sección se presentan los métodos semidiscretos, basados en aplicar una discretización espacial y mantener intacta la derivada temporal. De esta forma, la ecuación en derivadas se reduce a un sistema de ecuaciones diferenciales que puede resolverse mediante un método de Runge-Kutta de orden superior.

Así, sea $u = u(x, t)$ la solución de entropía de (4.2), para $t \geq 0$ se considera

$$I_j(t) = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t) dx, \quad j = 0, \dots, J,$$

siendo $I_j(0)$ valores conocidos por la condición inicial. Integrando la ecuación $u_t + f(u)_x = 0$ respecto de x en cada celda c_j se obtiene

$$\frac{d}{dt} I_j(t) + \frac{1}{\Delta x} \left(f(u(x_{j+\frac{1}{2}}^-, t)) - f(u(x_{j-\frac{1}{2}}^+, t)) \right) = 0, \quad t > 0,$$

con $u(x_{j+\frac{1}{2}}^\pm, t) = \lim_{x \rightarrow x_{j+\frac{1}{2}}^\pm} u(x, t)$.

Sea $\hat{f}_{j+\frac{1}{2}}(t)$ una aproximación de $f(u(x_{j+\frac{1}{2}}^-, t))$ y $f(u(x_{j+\frac{1}{2}}^+, t))$, se define $(u_0(t), \dots, u_J(t))$ como la solución del sistema

$$\begin{aligned} \frac{d}{dt} u_j(t) &= \mathcal{L}_j(t) = -\frac{1}{\Delta x} \left(\hat{f}_{j+\frac{1}{2}}(t) - \hat{f}_{j-\frac{1}{2}}(t) \right) = 0, \quad t > 0 \\ u_j(0) &= I_j(0), \end{aligned} \quad j = 0, \dots, J. \quad (4.32)$$

Fijado $t_0 = 0$ y $u_j^0 = u_j(0)$ para cada $j = 0, \dots, J$, suponiendo calculadas las estimaciones u_j^n de $u_j(t_n)$ para cierto t_n , se define $t_{n+1} = t_n + \Delta t_n$ según la condición

$$\max_j |f'(u_j^n)| \frac{\Delta t_n}{\Delta x} \leq \frac{1}{2}.$$

Los valores u_j^{n+1} pueden hallarse mediante el método de Runge-Kutta de segundo orden siguiente:

$$\begin{aligned} u_j^{(1)} &= u_j^n + \frac{\Delta t_n}{2} \mathcal{L}_j^n, \\ u_j^{(2)} &= u_j^{(1)} + \Delta t_n \mathcal{L}_j^{(1)}, \\ u_j^{n+1} &= \frac{1}{2}(u_j^n + u_j^{(2)}), \quad j = 0, \dots, J. \end{aligned} \quad (4.33)$$

Se utiliza este método porque, a diferencia de otros esquemas de Runge-Kutta, satisface una discretización de (4.29):

$$\sum_{j=0}^{J-1} |u_{j+1}^{n+1} - u_j^{n+1}| \leq \sum_{j=0}^{J-1} |u_{j+1}^n - u_j^n|. \quad (4.34)$$

Este proceso se repetiría hasta obtener $t_n = T$ para cierto n .

En (4.33), los términos \mathcal{L}_j^* corresponden a

$$\mathcal{L}_j^* = -\frac{1}{\Delta x} \left(\hat{f}_{j+\frac{1}{2}}^* - \hat{f}_{j-\frac{1}{2}}^* \right).$$

Para los flujos $\hat{f}_{j+\frac{1}{2}}^*$, se toma la función lineal

$$p_{j,n}(x) = u_j^n + \sigma_j^n(x - x_j),$$

con el parámetro σ_j^n definido según (4.31). A continuación, se denota

$$u_{j+\frac{1}{2}}^- = p_{j,n}(x_{j+\frac{1}{2}}), \quad u_{j-\frac{1}{2}}^+ = p_{j,n}(x_{j-\frac{1}{2}})$$

y se escoge

$$\hat{f}_{j+\frac{1}{2}}^* = \hat{f}(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+),$$

con $\hat{f} = \hat{f}^{God}$ o $\hat{f} = \hat{f}^{Rus}$. De esta forma, además de verificar la condición de estabilidad (4.34), el método resultante es conservativo y consistente según las definiciones de la sección 4.4, lo que, por el teorema 5 garantiza la aproximación de la solución correcta.

Cabe notar que, una vez calculados u_0^n, \dots, u_J^n , para obtener las estimaciones del instante $t = t_{n+1}$, es preciso fijar un valor para u_{-1}^n y u_{-2}^n , y también para u_{J+1}^n y u_{J+2}^n . Para ello, se siguen una de las dos estrategias siguientes:

- Condiciones de frontera artificiales: $u_{-1}^n = u_0^n$, $u_{-2}^n = u_{-1}^n$, $u_{J+1}^n = u_J^n$ y $u_{J+2}^n = u_{J+1}^n$.
- Condiciones de frontera periódicas: $u_{-1}^n = u_J^n$, $u_{-2}^n = u_{J-1}^n$, $u_{J+1}^n = u_0^n$ y $u_{J+2}^n = u_1^n$.

Para finalizar el capítulo se muestran dos ejemplos de aplicación del método de orden 2 explicado en este último apartado.

Ejemplo 14. *En primer lugar se trata de resolver el problema del ejemplo 12 con el método presentado en este apartado, tomando $\hat{f} = \hat{f}^{Rus}$, tomando condiciones de frontera artificiales y tiempo final $T = 0.1$. En la Tabla 4.3 se muestran los errores cometidos para distintos valores de J , calculados según la explicación del ejemplo 12, y se estima el orden de convergencia, donde se observa que se aproxima a 2.*

J	$E^{\Delta x}$	$p^{\Delta x}$
49	$3.76 \cdot 10^{-3}$	—
99	$1.03 \cdot 10^{-3}$	1.8661
199	$2.68 \cdot 10^{-4}$	1.9447
399	$6.88 \cdot 10^{-5}$	1.9626
799	$1.76 \cdot 10^{-5}$	1.9628
1599	$4.5 \cdot 10^{-6}$	1.9729
3199	$1.13 \cdot 10^{-6}$	1.9883

Tabla 4.3: Error y orden aproximado de la versión del método de Godunov de orden 2 para $u_t + uu_x = 0$ y condición inicial (4.24).

Ejemplo 15. *En este segundo ejemplo se considera de nuevo la ecuación de Burgers, ahora con dato inicial (4.13). Se aplica el método de Godunov y su versión de orden 2 explicada en este apartado, con condiciones de frontera periódicas. Las soluciones para $T = 0.5$ se representan en la Figura 4.6. Cabe notar que el método de orden 2 aproxima la solución de forma más precisa en las discontinuidades.*

Por otro lado, en la Tabla 4.4 se muestra el error cometido para distintos valores de J mediante el método de orden 2, y la estimación del orden. En este caso, el orden es prácticamente 1, dos décimas superior al obtenido mediante el método de Godunov.

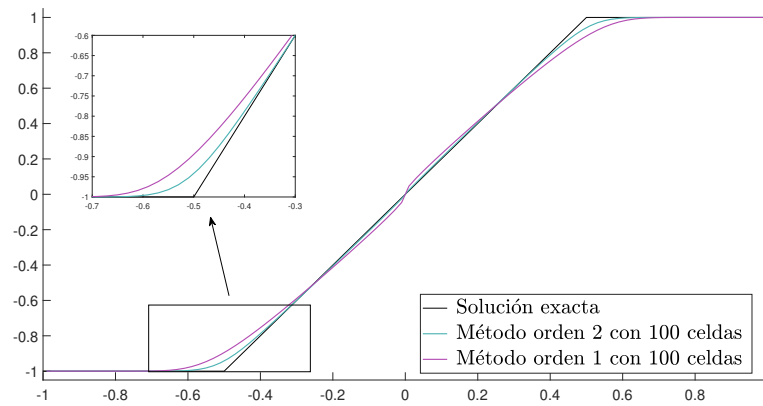


Figura 4.6: Aproximación obtenida con el método de Rusanov y su versión de orden 2, junto con la solución exacta del problema $u_t + uu_x = 0$ con (4.13).

J	$E^{\Delta x}$	$p^{\Delta x}$
49	$2.96 \cdot 10^{-2}$	—
99	$1.45 \cdot 10^{-2}$	1.0258
199	$7.29 \cdot 10^{-3}$	0.9954
399	$3.65 \cdot 10^{-3}$	0.9980
799	$1.83 \cdot 10^{-3}$	0.9991
1599	$9.14 \cdot 10^{-4}$	0.9996
3199	$4.57 \cdot 10^{-4}$	0.9998

Tabla 4.4: Error y orden aproximado del método de orden 2 para $u_t + uu_x = 0$ y condición inicial (4.13).

Capítulo 5

Métodos de reconstrucción de orden alto

Aplicación a la resolución de leyes de conservación

En el capítulo anterior se ha utilizado una estimación del promedio de la solución en cada celda para hallar una aproximación puntual de dicha función. Para esta reconstrucción se han empleado funciones constantes y lineales a trozos, obteniendo, así, orden 1 y 2 respectivamente. En las siguientes páginas se presentan distintas estrategias para aumentar el orden de la estimación puntual, que luego se emplea para resolver los problemas de Riemann adecuados. Estas herramientas, junto con un método de Runge-Kutta estable y de orden alto, permiten obtener una aproximación bastante más precisa de la solución, tanto en los intervalos de suavidad como en los puntos donde ésta presenta alguna singularidad.

En la sección 4.5 se ha estudiado un método de reconstrucción con la propiedad TVD (4.30), mediante el limitador de pendiente *minmod*. No obstante, es conocido que no puede obtenerse orden superior a dos si se pretende conservar la disminución de la variación. Así, para alcanzar mayor precisión se han desarrollado algunas técnicas basadas en ideas similares, pero sin asegurar de manera estricta esta condición TVD.

Algunos ejemplos son los métodos ENO (*essentially non-oscillatory*) y WENO (*weighted essentially non-oscillatory*), explicados en las tres secciones siguientes tomando como referencia [2, 11]. En el apartado 5.4 se estudia cómo aplicar estos procedimientos para la resolución de leyes de conservación, siguiendo las estrategias del capítulo 4. Por último, en la sección 5.5 se propone resolver las ecuaciones planteadas desde una perspectiva distinta [2], basada en estas técnicas de reconstrucción de orden alto y que se usan en el capítulo final del trabajo.

5.1. Reconstrucción de funciones suaves

Un primer paso para estudiar los métodos ENO y WENO es el problema expuesto de manera formal a continuación:

Sea $r > 0$, dada la partición de la Figura 4.1 y los valores promedio de una función $h : [x_L, x_R] \rightarrow \mathbb{R}$ suficientemente suave:

$$\bar{h}_i = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h(\xi) d\xi, \quad (5.1)$$

se pide hallar, para cada celda c_j , un polinomio $q_j^r(x)$ de grado menor que r

que aproxime h en c_j con orden r , es decir,

$$q_j^r(x) - h(x) = \mathcal{O}(\Delta x^r), \quad \forall x \in c_j.$$

En primer lugar, fijada una celda c_j arbitraria, deben seleccionarse r celdas consecutivas conteniendo a c_j , esto es,

$$\bar{S}_{j,k}^r = \{c_{j-k}, \dots, c_{j-k+r-1}\}, \quad k \in \{0, \dots, r-1\}. \quad (5.2)$$

Denótese por $S_{j,k}$ al conjunto formado por los extremos de las celdas en $\bar{S}_{j,k}$:

$$S_{j,k}^r = \{x_{j-k-\frac{1}{2}}, \dots, x_{j-k+r-\frac{1}{2}}\}.$$

Si alguna celda c_* de $\bar{S}_{j,k}^r$ no pertenece a la partición de $[x_L, x_R]$ considerada anteriormente, debe definirse el promedio \bar{h}_* siguiendo una de las estrategias mostradas a continuación:

- Condiciones de frontera artificiales: $\bar{h}_{-r} = \bar{h}_{r-1}$, $\bar{h}_{J+r} = \bar{h}_{J-r+1}$.
- Condiciones de frontera periódicas: $\bar{h}_{-r} = \bar{h}_{J-r+1}$, $\bar{h}_{J+r} = \bar{h}_{r-1}$.

Con el fin de resolver esta cuestión, se toma la función primitiva de h siguiente:

$$H(x) = \int_{x_L}^x h(\xi) d\xi.$$

Cabe notar que las diferencias divididas¹ de primer orden de H en los extremos de las celdas coinciden con los promedios (5.1):

$$H[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] = \frac{H(x_{i+\frac{1}{2}}) - H(x_{i-\frac{1}{2}})}{\Delta x} = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h(\xi) d\xi = \bar{h}_i.$$

Esto implica que las diferencias divididas de H de cualquier orden pueden obtenerse también a partir de la información conocida. En particular, si se define

$$\begin{aligned} \bar{h}[x_i] &= \bar{h}_i, \\ \bar{h}[x_i, \dots, x_{i+l}] &= \frac{\bar{h}[x_{i+1}, \dots, x_{i+l}] - \bar{h}[x_i, \dots, x_{i+l-1}]}{x_{i+l} - x_i}, \quad l > 0, \end{aligned}$$

puede comprobarse la relación

$$H[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, \dots, x_{i+l+\frac{1}{2}}] = \frac{1}{l+1} \bar{h}[x_i, \dots, x_{i+l}], \quad l \geq 0, \quad (5.3)$$

por inducción sobre el parámetro l .

A continuación se toma el (único) polinomio $Q_{j,k}^r$ de grado menor o igual que r que interpola a la función primitiva H en los puntos de $S_{j,k}^r$:

$$Q_{j,k}^r(x) = \sum_{i=0}^r H[x_{j-k-\frac{1}{2}}, \dots, x_{j-k+l-\frac{1}{2}}] \prod_{m=0}^{i-1} (x - x_{j-k+m-\frac{1}{2}}). \quad (5.4)$$

Debe notarse que el polinomio $Q_{j,k}^r$ es una aproximación de H de orden $r+1$ en la envoltura convexa de $S_{j,k}^r$, $\text{conv}(S_{j,k}^r) = [x_{j-k-\frac{1}{2}}, x_{j-k+r-\frac{1}{2}}]$, siempre que esta función sea suficientemente

¹Dada una función $f: D \subset \mathbb{R} \rightarrow \mathbb{R}$, $\{x_i\}_{i=1}^n \subset D$ con $x_i < x_{i+1}$, las **diferencias divididas** de f en estos puntos se definen como

$$\begin{aligned} f[x_i] &= f(x_i), \\ f[x_i, \dots, x_{i+j}] &= \frac{f[x_{i+1}, \dots, x_{i+j}] - f[x_i, \dots, x_{i+j-1}]}{x_{i+j} - x_i}, \quad j = 1, 2, \dots \end{aligned}$$

suave en dicho intervalo.

Por teoría básica de interpolación se deduce que la función derivada de este polinomio, denotada por $q_{j,k}^r$, es una estimación puntual de h de orden r , en particular en c_j :

$$q_{j,k}^r(x) - h(x) = \mathcal{O}(\Delta x^r), \quad \forall x \in c_j.$$

Cabe destacar que este polinomio $q_{j,k}^r$ interpola los promedios de h en las celdas de $\bar{S}_{j,k}^r$:

$$\begin{aligned} \frac{1}{\Delta x} \int_{c_{j-k+i}} q_{j,k}^r(x) dx &= \frac{Q_{j,k}^r(x_{j-k+i+\frac{1}{2}}) - Q_{j,k}^r(x_{j-k+i-\frac{1}{2}})}{\Delta x} \\ &= \frac{H(x_{j-k+i+\frac{1}{2}}) - H(x_{j-k+i-\frac{1}{2}})}{\Delta x} \quad i = 0, \dots, r-1. \\ &= \bar{h}_{j-k+i}. \end{aligned}$$

La segunda igualdad se cumple porque $Q_{j,k}^r$ interpola a H en los puntos de $S_{j,k}^r$.

Al derivar (5.4), el polinomio $q_{j,k}^r$ no incluye el término $H(x_{j-k-\frac{1}{2}})$, de modo que, por la propiedad (5.3), éste solo depende de los promedios conocidos $\bar{h}_{j-k}, \dots, \bar{h}_{j-k+r-1}$. Así pues, aunque se usa la función primitiva H para deducir $q_{j,k}^r$, no es necesario conocer ninguna información acerca de ella. Se ha hallado la expresión explícita de dicho polinomio para $r = 2$, $r = 3$ y $k \in \{-1, 0, \dots, r-1\}$ (se añade $k = -1$ por conveniencia, para los cálculos que siguen).

No obstante, en nuestro caso tan solo se utilizan los valores que toma $q_{j,k}^r$ en los extremos de la celda c_j , los puntos $x_{j-\frac{1}{2}}$ y $x_{j+\frac{1}{2}}$. Evaluando los polinomios calculados, $q_{j,k}^2$ y $q_{j,k}^3$, en estos nodos, se obtiene

$$q_{j,k}^r(x_{j+\frac{1}{2}}) = \sum_{i=0}^{r-1} C_{k,i} \bar{h}_{j-k+i}, \quad q_{j,k}^r(x_{j-\frac{1}{2}}) = \sum_{i=0}^{r-1} C_{k-1,i} \bar{h}_{j-k+i}, \quad r = 2, 3. \quad (5.5)$$

El valor de los parámetros $C_{k,i}$ se especifica en la Tabla 5.1.

r	k	$i = 0$	$i = 1$	$i = 2$
2	-1	$\frac{3}{2}$	$-\frac{1}{2}$	
	0	$\frac{1}{2}$	$\frac{1}{2}$	
	1	$-\frac{1}{2}$	$\frac{3}{2}$	
3	-1	$\frac{11}{6}$	$-\frac{7}{6}$	$\frac{1}{3}$
	0	$\frac{1}{3}$	$\frac{5}{6}$	$-\frac{1}{6}$
	1	$-\frac{1}{6}$	$\frac{5}{6}$	$\frac{1}{3}$
	2	$\frac{1}{3}$	$-\frac{7}{6}$	$\frac{11}{6}$

Tabla 5.1: Coeficientes $C_{k,i}$ para el cálculo de los extremos en (5.5).

5.2. Método ENO

En la sección anterior, dados los valores promedio (5.1) de una función suficientemente suave, se ha obtenido una estimación puntual de ésta en cada celda. Para ello, fijada c_j , pueden emplearse distintos conjuntos de celdas, mostrados en (5.2).

No obstante, las soluciones de las ecuaciones hiperbólicas no lineales pueden presentar singularidades en tiempo finito, como se ha comprobado en algunos ejemplos a lo largo del trabajo.

Es por esto que, si utilizamos cualquier $\overline{S}_{j,k}^r$ para el proceso de reconstrucción, pueden obtenerse aproximaciones que no verifican la condición del orden e incluso presentan alguna oscilación.

La idea principal del método ENO reside en escoger, entre los distintos conjuntos $S_{j,k}^r$ posibles, aquel que proporcione el polinomio interpolador *más suave*. Este algoritmo selecciona $S_{j,k}^r$ usando diferencias divididas como indicadores de suavidad, eligiendo el conjunto que genera menores diferencias, en un intento por evitar oscilaciones en la interpolación y, posteriormente, alrededor de las discontinuidades de la solución. A continuación se detalla el procedimiento seguido para llevar a cabo esta elección.

Para hallar una aproximación de h en una determinada celda c_j , debe incluirse en $S_{j,k}^r$ los extremos de esta celda: $x_{j-\frac{1}{2}}$ y $x_{j+\frac{1}{2}}$. El polinomio de primer grado que interpola a la función primitiva H en estos puntos viene dado por

$$P_j^1(x) = H(x_{j-\frac{1}{2}}) + H[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}](x - x_{j-\frac{1}{2}}).$$

Los términos $H[\cdot, \dots, \cdot]$ corresponden a las diferencias divididas de H , definidas en la página 44.

A continuación, el tercer nodo escogido puede ser $x_{j-\frac{3}{2}}$ o bien $x_{j+\frac{3}{2}}$. Si se toma $x_{j-\frac{3}{2}}$, el polinomio que interpola a H en $\{x_{j-\frac{3}{2}}, x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}\}$ será

$$P_j^{2,1}(x) = P_j^1(x) + H[x_{j-\frac{3}{2}}, x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}](x - x_{j-\frac{1}{2}})(x - x_{j+\frac{1}{2}}).$$

En cambio, si consideramos $x_{j+\frac{3}{2}}$, el polinomio de segundo grado vendrá dado por

$$P_j^{2,2}(x) = P_j^1(x) + H[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}, x_{j+\frac{3}{2}}](x - x_{j-\frac{1}{2}})(x - x_{j+\frac{1}{2}}).$$

Los polinomios $P_j^{2,1}$ y $P_j^{2,2}$ coinciden salvo en la diferencia dividida de segundo orden que acompaña al término $(x - x_{j-\frac{1}{2}})(x - x_{j+\frac{1}{2}})$. Se añade el nodo $x_{j-\frac{3}{2}}$ si se cumple

$$|H[x_{j-\frac{3}{2}}, x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]| \leq |H[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}, x_{j+\frac{3}{2}}]|,$$

o el punto $x_{j+\frac{3}{2}}$ en caso contrario. Cabe notar que las diferencias divididas de H pueden obtenerse a partir de las medias \bar{h}_j conocidas, mediante la relación (5.3).

Debe seguirse este algoritmo hasta obtener $r + 1$ puntos. De esta forma se obtiene el mallado $S_{j,k}^r$ para cierto $k = k_0 \in \{0, \dots, r - 1\}$. La estimación buscada es el polinomio q_{j,k_0}^r presentado en la sección anterior, una aproximación de h de orden r si esta función es suave en las celdas de \overline{S}_{j,k_0}^r . Los valores que toma en los extremos de c_j se calculan mediante (5.5).

5.3. Método WENO

El procedimiento explicado en el apartado anterior permite obtener aproximaciones bastante precisas de las soluciones de las leyes de conservación, sin presentar oscilaciones, como se muestra en la sección 5.4. No obstante, para cada intervalo c_j , los r conjuntos $S_{j,k}^r$ considerados contienen, entre todos, $2r - 1$ celdas: $c_{j-r+1}, \dots, c_{j+r-1}$. Si h es suficientemente suave en cada una de ellas, puede tomarse el conjunto

$$\overline{S}_j^{2r-1} = \{c_{j-r+1}, \dots, c_{j+r-1}\}$$

para hallar una estimación de orden $2r - 1$, denotada por q_j^{2r-1} , siguiendo la explicación de la sección 5.1. Esta aproximación mejoraría la obtenida mediante el algoritmo previo, de orden r . La limitación expuesta motiva el desarrollo de una nueva técnica de reconstrucción, el método WENO, estudiado a continuación.

En primer lugar, cabe notar que, para resolver las ecuaciones hiperbólicas, solo se utiliza la aproximación de h en los extremos de las celdas. Siguiendo [2], los valores $q_j^{2r-1}(x_{j\pm\frac{1}{2}})$ pueden expresarse de la forma

$$q_j^{2r-1}(x_{j+\frac{1}{2}}) = \sum_{k=0}^{r-1} d_k^r q_{j,k}^r(x_{j+\frac{1}{2}}), \quad q_j^{2r-1}(x_{j-\frac{1}{2}}) = \sum_{k=0}^{r-1} d_{r-k-1}^r q_{j,k}^r(x_{j-\frac{1}{2}}). \quad (5.6)$$

Los polinomios $q_{j,k}^r$ son las estimaciones de h en los intervalos $[x_{j-k-\frac{1}{2}}, x_{j-k+r-\frac{1}{2}}]$, cuyas imágenes en $x = x_{j\pm\frac{1}{2}}$ pueden calcularse mediante (5.5).

Los coeficientes d_k^r reciben el nombre de *pesos óptimos*. Estos pesos verifican la relación

$$d_0^r + d_1^r + \cdots + d_{r-2}^r + d_{r-1}^r = 1$$

y su valor para $r = 2$ y $r = 3$ se especifica a continuación:

$$\begin{aligned} r = 2 & : d_0^2 = \frac{2}{3}, \quad d_1^2 = \frac{1}{3} \\ r = 3 & : d_0^3 = \frac{3}{10}, \quad d_1^3 = \frac{3}{5}, \quad d_2^3 = \frac{1}{10}. \end{aligned}$$

Para el método WENO, en lugar de considerar una única estimación $q_{j,k}^r$ y descartar el resto, se propone tomar una combinación convexa de todas ellas, como en (5.6). De esta forma, se asignan unos pesos a los polinomios $q_{j,k}^r$, en función de su *suavidad*, que determinan la contribución de cada uno de ellos a la aproximación final.

Así, las estimaciones de $h(x_{j\pm\frac{1}{2}})$ vendrán dadas por

$$\hat{h}(x_{j+\frac{1}{2}}) = \sum_{k=0}^{r-1} \omega_{j,k}^r q_{j,k}^r(x_{j+\frac{1}{2}}) \approx h(x_{j+\frac{1}{2}}), \quad (5.7)$$

$$\hat{h}(x_{j-\frac{1}{2}}) = \sum_{k=0}^{r-1} \tilde{\omega}_{j,k}^r q_{j,k}^r(x_{j-\frac{1}{2}}) \approx h(x_{j-\frac{1}{2}}), \quad (5.8)$$

para ciertos $\omega_{j,k}^r$ y $\tilde{\omega}_{j,k}^r$, $k = 0, \dots, r-1$. La clave del método reside en la manera de obtener estos pesos.

Como se ha comentado anteriormente, el valor del peso $\omega_{j,k}^r$ se escoge según la suavidad de la función h en las celdas del conjunto $\overline{S}_{j,k}^r$. Si, en ellas, la función presenta alguna discontinuidad, este valor debería ser prácticamente cero. De este modo, $\hat{h}(x_{j+\frac{1}{2}})$ solo se calcularía a partir de las celdas donde no hay singularidades, obteniendo una estimación al menos de orden r . En cambio, si la función es suave en todos los conjuntos $\overline{S}_{j,k}^r$, los pesos $\omega_{j,k}^r$ deben tomar valores cercanos a los óptimos, d_k^r , para alcanzar orden superior.

Como en [2], se propone considerar

$$\begin{aligned} \omega_{j,k}^r &= \frac{\alpha_{j,k}^r}{\alpha_{j,0}^r + \cdots + \alpha_{j,r-1}^r}, & \alpha_{j,k}^r &= \frac{d_k^r}{(\varepsilon + IS_{j,k}^r)^2} \\ \tilde{\omega}_{j,k}^r &= \frac{\tilde{\alpha}_{j,k}^r}{\tilde{\alpha}_{j,0}^r + \cdots + \tilde{\alpha}_{j,r-1}^r}, & \tilde{\alpha}_{j,k}^r &= \frac{d_{r-k-1}^r}{(\varepsilon + IS_{j,k}^r)^2} \end{aligned} \quad k = 0, \dots, r-1. \quad (5.9)$$

En (5.9), el parámetro ε hace referencia a un número positivo muy pequeño, empleado para evitar que el denominador se anule. Por otro lado, el término $IS_{j,k}^r$ corresponde a una medida de la suavidad de la función h en las celdas de $\overline{S}_{j,k}^r$, que puede calcularse según

$$IS_{j,k}^r = \sum_{m=0}^{r-1} \Delta x^{2m-1} \int_{c_j} \left(\frac{d^m}{dx^m} q_{j,k}^r(x) \right)^2 dx. \quad (5.10)$$

Cuanto menor sea el valor $IS_{j,k}^r$, más suave será el polinomio $q_{j,k}^r$ y, por tanto, éste deberá tener mayor peso al calcular la aproximación (5.7). Este mismo hecho se refleja en la definición (5.9).

Finalmente, para $r = 2$ la expresión (5.10) equivale a

$$IS_{j,0}^2 = (\bar{h}_{j+1} - \bar{h}_j)^2, \quad IS_{j,1}^2 = (\bar{h}_j - \bar{h}_{j-1})^2.$$

Y para $r = 3$,

$$\begin{aligned} IS_{j,0}^3 &= \frac{13}{12}(\bar{h}_j - 2\bar{h}_{j+1} + \bar{h}_{j+2})^2 + \frac{1}{4}(3\bar{h}_j - 4\bar{h}_{j+1} + \bar{h}_{j+2})^2 \\ IS_{j,1}^3 &= \frac{13}{12}(\bar{h}_{j-1} - 2\bar{h}_j + \bar{h}_{j+1})^2 + \frac{1}{4}(\bar{h}_{j-1} - \bar{h}_{j+1})^2 \\ IS_{j,2}^3 &= \frac{13}{12}(\bar{h}_{j-2} - 2\bar{h}_{j-1} + \bar{h}_j)^2 + \frac{1}{4}(\bar{h}_{j-2} - 4\bar{h}_{j-1} + 3\bar{h}_j)^2. \end{aligned}$$

5.4. Aplicación a los métodos de volúmenes finitos

Cabe recordar que el objetivo es resolver el problema de valor inicial

$$\begin{aligned} u_t + f(u)_x &= 0, \quad x \in [x_L, x_R], \quad t > 0 \\ u(x, 0) &= u_0(x). \end{aligned} \tag{5.11}$$

Sea $u = u(x, t)$ su solución, en los métodos de volúmenes finitos tratados en el capítulo 4, una vez calculadas las estimaciones u_j^n de los promedios de $u(\cdot, t_n)$ en las celdas, debe hallarse una aproximación puntual de esta función en cada intervalo c_j . Para ello, en las secciones 4.3 y 4.5 se usan funciones constantes

$$u(x, t_n) \approx u_j^n, \quad x \in c_j$$

o lineales

$$u(x, t_n) \approx u_j^n + \sigma_j^n(x - x_j), \quad x \in c_j.$$

Estas sencillas estrategias pueden sustituirse por los métodos de reconstrucción de orden alto estudiados, como se muestra en este apartado.

La solución de (5.11) se obtendrá mediante la formulación semidiscreta presentada en la sección 4.6. Ésta consiste en resolver el sistema de ecuaciones diferenciales

$$u_j'(t) = \mathcal{L}_j(t) = -\frac{1}{\Delta x} \left(\hat{f}_{j+\frac{1}{2}}(t) - \hat{f}_{j-\frac{1}{2}}(t) \right), \quad j = 0, \dots, J \tag{5.12}$$

con condición inicial

$$u_j(0) = \frac{1}{\Delta x} \int_{c_j} u_0(x) dx.$$

En la expresión (5.12), $u_j(t)$ corresponde a una aproximación del promedio

$$\frac{1}{\Delta x} \int_{c_j} u(x, t) dx,$$

mientras que $\hat{f}_{j\pm\frac{1}{2}}(t)$ es una estimación del valor $f(u(x_{j\pm\frac{1}{2}}, t))$.

En el capítulo anterior, para resolver el sistema (5.12), se utiliza un método de Runge-Kutta de orden dos que cumple la propiedad TVD, en (4.34). Cabe notar que las técnicas ENO y WENO permiten hallar una aproximación espacial de orden superior. No obstante, si en este caso se aplica un método de orden dos para la evolución temporal, la estimación final, en las regiones donde la

solución sea suave, será de nuevo de orden dos. Para aumentar esta precisión, se propone emplear un método de Runge-Kutta de orden tres que conserva la condición TVD, dado por

$$\begin{aligned} u_j^{(1)} &= u_j^n + \Delta t_n \mathcal{L}_j^n, \\ u_j^{(2)} &= \frac{3}{4} u_j^n + \frac{1}{4} u_j^{(1)} + \frac{1}{4} \Delta t_n \mathcal{L}_j^{(1)}, \\ u_j^{n+1} &= \frac{1}{3} u_j^n + \frac{2}{3} u_j^{(2)} + \frac{2}{3} \Delta t_n \mathcal{L}_j^{(2)}, \quad j = 0, \dots, J. \end{aligned} \quad (5.13)$$

Para aplicar (5.13), se toma $u_j^0 = u_j(0)$ para cualquier $j \in \{0 \dots, J\}$. Posteriormente, dado un cierto instante t_n , calculados los valores $u_j^n \approx u_j(t_n)$, se determinan las aproximaciones a tiempo $t_{n+1} = t_n + \Delta t_n$, donde Δt_n debe verificar la relación

$$\max_j |f'(u_j^n)| \frac{\Delta t_n}{\Delta x} \leq 1.$$

A continuación se especifica cómo aplicar las nuevas técnicas de reconstrucción para obtener \mathcal{L}_j^* a partir de u_j^* , donde \star hace referencia a "n", "(1)" o "(2)".

En primer lugar se aplica el método ENO o WENO para cierto r , en las celdas

$$c_{-1}, c_0, c_1, \dots, c_{J-1}, c_J, c_{J+1},$$

tomando los promedios $\bar{h}_j = u_j^*$. Deben añadirse los valores u_j^* correspondientes a los subíndices $j = -1, -2, \dots, -r$ y $j = J+1, J+2, \dots, J+r$, mediante alguna de las estrategias explicadas en la sección 5.1. Las aproximaciones de los valores $h(x_{j+\frac{1}{2}})$ y $h(x_{j-\frac{1}{2}})$ en cada celda c_j se denotarán por $u_{j+\frac{1}{2}}^{*-}$ y $u_{j-\frac{1}{2}}^{*+}$ respectivamente. Una vez llevado a cabo este procedimiento, los flujos $\hat{f}_{j+\frac{1}{2}}^*$ se obtienen de la forma

$$\hat{f}_{j+\frac{1}{2}}^* = \hat{f}(u_{j+\frac{1}{2}}^{*-}, u_{j+\frac{1}{2}}^{*+}), \quad j = -1, \dots, J,$$

donde \hat{f} corresponde a \hat{f}^{God} o \hat{f}^{Rus} , funciones presentadas en (4.10) y (4.19) respectivamente. Por último, el término \mathcal{L}_j^* equivale a la expresión siguiente:

$$\mathcal{L}_j^* = -\frac{1}{\Delta x} \left(\hat{f}_{j+\frac{1}{2}}^* - \hat{f}_{j-\frac{1}{2}}^* \right), \quad j = 0, \dots, J.$$

De nuevo puede justificarse que el método es conservativo y consistente. Junto con la propiedad TVD, este hecho prueba que el resultado obtenido es una estimación de la solución de entropía del problema.

A continuación se presenta un experimento numérico con el objetivo de mostrar que las reconstrucciones ENO y WENO permiten obtener estimaciones más precisas de las soluciones de las leyes de conservación.

Se denota por M1 al método de volúmenes finitos que usa reconstrucción lineal *minmod* y método de Runge-Kutta de orden 2. El término M2 hace referencia al método de volúmenes finitos con reconstrucción ENO con $r = 3$ y método de Runge-Kutta de orden 2. Finalmente, M3 corresponde a la reconstrucción WENO con $r = 3$ y método de Runge-Kutta de orden 3. Así, los métodos M1 y M2 serían de orden dos para soluciones suaves, mientras que M3 sería de orden tres.

En la Figura 5.1 se muestra la solución del problema planteado en el ejemplo 15 junto con las estimaciones de los métodos M1 y M2. En ella se observa que mediante la reconstrucción ENO se obtiene una aproximación más precisa en todo el dominio, y principalmente en los puntos de discontinuidad.

Por otro lado, en la Figura 5.2, más concretamente en el recuadro ampliado, se muestra que el método de orden tres logra una estimación que, en todo momento, se encuentra más cercana a la gráfica de la solución exacta que la aproximación del método M2.

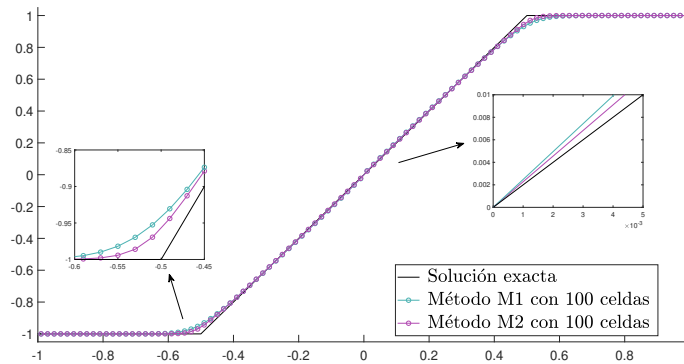


Figura 5.1: Estimaciones con los métodos M1 y M2 y solución exacta de la ecuación de Burgers con condición inicial (4.13).

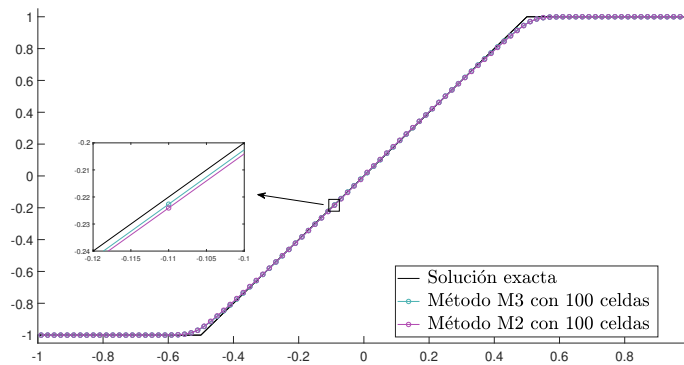


Figura 5.2: Estimaciones con los métodos M2 y M3 y solución exacta de la ecuación de Burgers con condición inicial (4.13).

5.5. Método de diferencias finitas de Shu y Osher

Hasta ahora se han estudiado métodos de volúmenes finitos para la ecuación escalar unidimensional, en los que se busca, para cada instante t_n , aproximar los promedios de la solución en las celdas:

$$u_j^n \approx \frac{1}{\Delta x} \int_{c_j} u(x, t_n) dx.$$

No obstante, generalizar este tipo de métodos para sistemas de ecuaciones o para el caso multidimensional no resulta una tarea sencilla.

Por esta razón, Shu y Osher desarrollaron un método de diferencias finitas, el cual, en lugar de emplear promedios en intervalos, calcula estimaciones de la solución en cada punto del mallado:

$$u_j^n \approx u(x_j, t_n).$$

Esta nueva estrategia para resolver ecuaciones hiperbólicas permite reducir de manera considerable el coste computacional respecto de los métodos de volúmenes finitos del mismo orden. La presentamos a continuación para la ecuación escalar $1D$, tomando como referencia [2].

Este método de diferencias finitas tiene su base en el lema enunciado a continuación.

Lema 1. Si una función $h(x)$ verifica

$$f(u(x)) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi) d\xi,$$

entonces

$$f(u(x))_x = \frac{h\left(x + \frac{\Delta x}{2}\right) - h\left(x - \frac{\Delta x}{2}\right)}{\Delta x}.$$

Con el fin de obtener un método conservativo, el término $f(u(x))_x$ en $x = x_j$ debe estimarse mediante una expresión de la forma

$$\mathcal{L}_j^* = \frac{1}{\Delta x} \left(\hat{f}_{j+\frac{1}{2}}^* - \hat{f}_{j-\frac{1}{2}}^* \right), \quad j = 0, \dots, J. \quad (5.14)$$

Así, el lema anterior sugiere que los flujos $\hat{f}_{j+\frac{1}{2}}^*$ sean una aproximación de los valores $h^*(x_{j+\frac{1}{2}})$, donde $h^*(x)$ es una función con promedios

$$\bar{h}_j^* = \frac{1}{\Delta x} \int_{c_j} h^*(x) dx = f(u_j^*) \quad (5.15)$$

para cada j . Así, deben hallarse estimaciones de una función en los extremos de las celdas a partir de sus medias en ellas, hecho que invita a emplear las técnicas de reconstrucción ENO y WENO estudiadas a lo largo del capítulo. Cabe notar que no se requiere ninguna información acerca de h^* , más allá de las medias en (5.15), obtenidas anteriormente. Por otro lado, una vez calculados los flujos y (5.14), se resuelve el problema aplicando directamente un método de Runge-Kutta adecuado.

De esta forma, en principio solo queda por determinar qué promedios $\bar{h}_j^* = f(u_j^*)$ se usarán a la hora de calcular el término $\hat{f}_{j+\frac{1}{2}}^*$. Para ello, deben tenerse en cuenta las direcciones en las que se mueve la solución. En la ecuación escalar, estas direcciones vienen dadas por el signo de $f'(u)$. Así, una primera opción sería aplicar el siguiente algoritmo, donde \mathcal{R} representa la reconstrucción ENO o WENO de un cierto orden r .

Algoritmo 1

1. Calcular $\kappa_{j+\frac{1}{2}}^* = \begin{cases} \frac{f(u_{j+1}^*) - f(u_j^*)}{u_{j+1}^* - u_j^*}, & u_j^* \neq u_{j+1}^*, \\ f'(u_j^*), & u_j^* = u_{j+1}^*. \end{cases}$
2. Si $\kappa_{j+\frac{1}{2}}^* > 0$, tomar $\hat{f}_{j+\frac{1}{2}}^* = \mathcal{R}(f(u_{j-r+1}^*), \dots, f(u_{j+r-1}^*))$.
3. En caso contrario, tomar $\hat{f}_{j+\frac{1}{2}}^* = \mathcal{R}(f(u_{j-r+2}^*), \dots, f(u_{j+r}^*))$.

Cabe destacar que, para $r = 1$, los flujos coinciden con los del método de Roe, el cual no permite aproximar la solución VV en determinadas ocasiones, como se muestra en la Figura 4.4. Para evitar este hecho, se propone modificar el algoritmo de la forma que sigue.

Algoritmo 2

1. Tomar $\beta_{j+\frac{1}{2}} \geq \max_{u \in [u_j^*, u_{j+1}^*]} |f'(u)|$.
2. Calcular $\hat{f}_{j+\frac{1}{2}}^{*+}$ mediante el paso 2 del algoritmo 1, usando $\frac{1}{2}(f(u) + \beta_{j+\frac{1}{2}}^* u)$ en lugar de $f(u)$.
3. Calcular $\hat{f}_{j+\frac{1}{2}}^{*-}$ mediante el paso 3 del algoritmo 1, usando ahora $\frac{1}{2}(f(u) - \beta_{j+\frac{1}{2}}^* u)$ en vez de $f(u)$.
4. Tomar $\hat{f}_{j+\frac{1}{2}}^* = \hat{f}_{j+\frac{1}{2}}^{*+} + \hat{f}_{j+\frac{1}{2}}^{*-}$.

Este método puede extenderse de manera sencilla para hallar la solución de sistemas de ecuaciones, como se muestra en el capítulo 6. Posteriormente, en el capítulo 7, se usará para resolver el modelo de cromatografía planteado.

Capítulo 6

Extensión a sistemas de ecuaciones

En la primera parte del trabajo se han estudiado las leyes de conservación escalares desde el punto de vista teórico, y se han presentado algunos métodos numéricos que permiten estimar su solución. En el presente capítulo se generaliza, para sistemas de ecuaciones, el contenido tratado hasta el momento para la ecuación escalar. Así, se introduce el concepto de sistema hiperbólico y se plantea un estudio teórico y numérico de estos sistemas, tanto lineales como no lineales.

6.1. Introducción: sistemas hiperbólicos

En general, las leyes de conservación se representan mediante un sistema de ecuaciones en derivadas parciales, de la forma

$$\frac{\partial u_i}{\partial t} + \frac{\partial f_i(u)}{\partial x} = 0, \quad i = 1, \dots, m, \quad x \in \mathbb{R}, \quad t > 0. \quad (6.1)$$

En esta expresión, $u = (u_1, \dots, u_m) : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}^m$ representa la solución de la ley de conservación, formada por las variables conservadas u_i . Por su lado, las componentes de $f = (f_1, \dots, f_m) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ reciben el nombre de funciones de flujo. Cabe notar que este sistema se deduce a partir de una ecuación integral, más general, similar a (2.23).

Como en el caso escalar, se pretende hallar la solución de (6.1) que verifica una condición inicial:

$$u(x, 0) = u^0(x), \quad x \in \mathbb{R};$$

dada una función $u^0 : \mathbb{R} \rightarrow \mathbb{R}^m$ conocida.

El sistema puede reescribirse de la forma

$$u_t + f(u)_x = 0,$$

una expresión equivalente a

$$u_t + A(u)u_x = 0, \quad (6.2)$$

donde $A(u)$ corresponde a la matriz jacobiana de f :

$$A : \mathbb{R}^m \rightarrow \mathbb{R}^{m \times m}, \quad (A(u))_{i,j} = \frac{\partial f_i}{\partial u_j}(u), \quad i, j = 1, \dots, m.$$

Se dice que el sistema (6.2) es hiperbólico si, para cualquier vector $u \in \mathbb{R}^m$, la matriz $A(u)$ es diagonalizable con valores propios reales. Es decir, si $A(u)$ tiene m valores propios reales

$$\lambda_1(u) \leq \lambda_2(u) \leq \dots \leq \lambda_m(u)^2$$

²Cada valor propio se repite tantas veces como indique su multiplicidad.

y existe una base de \mathbb{R}^m formada por vectores propios $r_i(u)$ asociados. De esta forma, sea $R(u) = (r_1(u) | \dots | r_m(u))$, se satisface la relación

$$A(u) = R(u)\Lambda(u)R(u)^{-1}, \quad \Lambda(u) = \text{diag}(\lambda_1(u), \dots, \lambda_m(u)). \quad (6.3)$$

Por otro lado, el sistema será estrictamente hiperbólico si, además, los valores propios de la matriz jacobiana de f son distintos:

$$\lambda_1(u) < \lambda_2(u) < \dots < \lambda_m(u). \quad (6.4)$$

De ahora en adelante se asume que (6.2) es estrictamente hiperbólico. La condición de hiperbolicidad es necesaria para demostrar la existencia de algunas soluciones, mientras que la hipótesis (6.4) se asume para facilitar el estudio a realizar. No obstante, estas restricciones no suponen un impedimento para resolver el modelo del capítulo final, ya que éste, como se muestra más adelante, cumple ambas propiedades.

6.2. Teoría de sistemas lineales

Los sistemas lineales constituyen una generalización en varias variables de la ecuación escalar $u_t + au_x = 0$ estudiada en la sección 2.2. A continuación se deduce teóricamente la solución de los sistemas lineales estrictamente hiperbólicos, para una problema de Cauchy general y, posteriormente, para los problemas de Riemann en particular.

Un sistema lineal de leyes de conservación está formado por las ecuaciones (6.1), si cada función f_i depende de u de manera lineal, es decir, puede escribirse $f(u) = Au$ para una matriz $A \in \mathbb{R}^{m \times m}$ de coeficientes constantes. Como consecuencia, la ecuación (6.2) toma la expresión

$$u_t + Au_x = 0. \quad (6.5)$$

En este caso, la hiperbolicidad estricta implica que A tiene m valores propios reales y distintos $\lambda_1 < \dots < \lambda_m$ y, por tanto, m vectores propios linealmente independientes: r_1, \dots, r_m . Así, esta matriz puede expresarse de la forma

$$A = R\Lambda R^{-1}, \quad (6.6)$$

con $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$ y $R = (r_1 | \dots | r_m)$.

A partir de esta información, la ecuación (6.5) se reescribe de la siguiente forma:

$$u_t + Au_x = 0 \iff u_t + R\Lambda R^{-1}u_x = 0 \iff R^{-1}u_t + \Lambda R^{-1}u_x = 0.$$

Definiendo $w = R^{-1}u$, vector correspondiente a las coordenadas de u en la base de vectores propios, se deduce la expresión

$$w_t + \Lambda w_x = 0, \quad (6.7)$$

un nuevo sistema lineal, ahora con matriz diagonal. Este sistema equivale a m ecuaciones escalares lineales:

$$\frac{\partial w_i}{\partial t} + \lambda_i \frac{\partial w_i}{\partial x} = 0, \quad i = 1, \dots, m, \quad (6.8)$$

estudiadas en el capítulo 2.

Así, dado un dato inicial $u(x, 0) = u^0(x)$ para (6.5), se toma

$$w^0(x) = R^{-1}u^0(x)$$

y se resuelven los problemas

$$\begin{aligned} \frac{\partial w_i}{\partial t} + \lambda_i \frac{\partial w_i}{\partial x} &= 0, \\ w_i(x, 0) &= w_i^0(x), \end{aligned} \quad i = 1, \dots, m, \quad (6.9)$$

donde $w_i^0(x)$ es la componente i -ésima de $w^0(x)$. Es conocido que la solución de (6.9) viene dada por

$$w_i(x, t) = w_i^0(x - \lambda_i t), \quad i = 1, \dots, m. \quad (6.10)$$

Deshaciendo el cambio de variable, se obtiene la solución del sistema (6.5):

$$u(x, t) = R w(x, t) = \sum_{i=1}^m w_i(x, t) r_i = \sum_{i=1}^m w_i^0(x - \lambda_i t) r_i. \quad (6.11)$$

Como en el caso escalar, el procedimiento anterior puede llevarse a cabo para funciones u^0 que presenten alguna singularidad. En ese caso, la función (6.11) obtenida puede que no sea de clase \mathcal{C}^1 y no cumpla, de esta forma, la ecuación diferencial puntualmente. No obstante, se acepta como solución del problema, ya que sí verifica la ecuación integral que modeliza la ley de conservación.

Por último, se muestra con detalle la solución del problema de Riemann asociado a (6.5):

$$\begin{aligned} u_t + A u_x &= 0, \quad x \in \mathbb{R}, \quad t > 0, \\ u(x, 0) &= \begin{cases} u^-, & x < 0, \\ u^+, & x > 0. \end{cases} \end{aligned} \quad (6.12)$$

Siguiendo los pasos anteriores, se toma

$$w^0(x) = R^{-1} u^0(x) = \begin{cases} w^- = R^{-1} u^-, & x < 0, \\ w^+ = R^{-1} u^+, & x > 0, \end{cases}$$

de modo que las soluciones (6.10) son

$$w_i(x, t) = \begin{cases} w_i^-, & x < \lambda_i t, \\ w_i^+, & x > \lambda_i t, \end{cases} \quad i = 1, \dots, m.$$

Así pues, se tiene

$$w(x, t) = \begin{cases} w^{(0)} = w^-, & x < \lambda_1 t, \\ \dots & \\ w^{(i)}, & \lambda_i t < x < \lambda_{i+1} t, \\ \dots & \\ w^{(m)} = w^+, & x > \lambda_m t, \end{cases} \quad w^{(i)} = \begin{pmatrix} w_1^+ \\ \dots \\ w_i^+ \\ w_{i+1}^- \\ \dots \\ w_m^- \end{pmatrix},$$

una función constante a trozos con discontinuidades en las rectas $s_i \equiv x = \lambda_i t$. A través de la recta s_i , se produce una discontinuidad de salto tan solo en la componente i -ésima. Finalmente, la solución de (6.12) viene dada por

$$u(x, t) = R w(x, t) = \begin{cases} u^{(0)} = u^-, & x < \lambda_1 t, \\ \dots & \\ u^{(i)} = R w^{(i)}, & \lambda_i t < x < \lambda_{i+1} t, \\ \dots & \\ u^{(m)} = u^+, & x > \lambda_m t, \end{cases}$$

representada en la Figura 6.1. En esta ocasión, el salto en la recta s_i resulta proporcional al vector propio r_i :

$$u^{(i)} - u^{(i-1)} = R(w^{(i)} - w^{(i-1)}) = (w_i^+ - w_i^-) r_i = \alpha_i r_i,$$

con $\alpha_i = w_i^+ - w_i^-$. Análogamente se comprueba

$$u^{(i)} = u^- + \sum_{p=1}^i \alpha_p r_p, \quad u^{(i)} = u^+ - \sum_{p=i+1}^m \alpha_p r_p.$$

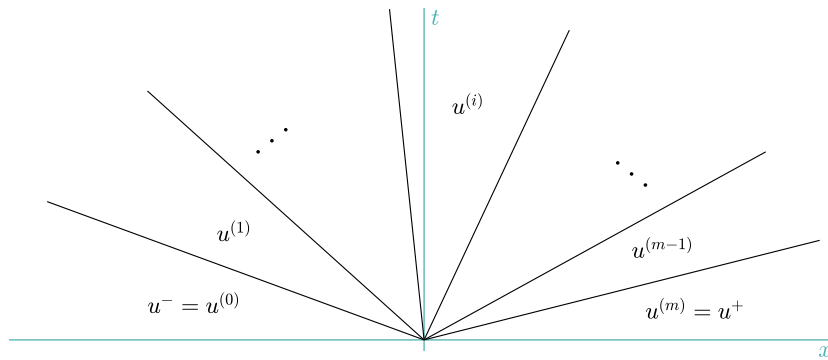


Figura 6.1: Solución del problema de Riemann para la ecuación vectorial lineal.

6.3. Métodos numéricos para sistemas lineales

Una vez analizados los sistemas lineales desde el punto de vista teórico, se presentan métodos numéricos que permiten estimar sus soluciones. Así, se considera la ecuación vectorial

$$u_t + Au_x = 0, \quad x \in [x_L, x_R], \quad t > 0, \quad (6.13)$$

donde A representa una matriz de orden m con valores propios reales y distintos, verificando la relación (6.6).

Fijado un tiempo final $T > 0$, se toman las particiones \mathcal{P}_x y \mathcal{P}_t definidas en (3.2) y (3.3), con espaciados Δx y Δt por determinar, dando lugar a un mallado

$$\{(x_j, t_n) : j = 0, \dots, J; n = 0, \dots, N\}$$

del dominio $[x_L, x_R] \times [0, T]$. Se pretende hallar estimaciones de $u(x_j, t_n) = (u_1(x_j, t_n), \dots, u_m(x_j, t_n))^T$, denotadas por

$$u_j^n = ((u_1)_j^n, \dots, (u_m)_j^n)^T. \quad (6.14)$$

Conocidos los valores u_j^0 a partir de una condición inicial dada, y fijada la condición de contorno a emplear, ver nota 1 o 2, se busca alguna estrategia para obtener los valores u_j^{n+1} a partir de las estimaciones para $t = t_n$.

Estudiado en el apartado anterior, mediante el cambio de variable $w = R^{-1}u$, el sistema (6.13) equivale a m ecuaciones escalares y lineales:

$$\frac{\partial w_i}{\partial t} + \lambda_i \frac{\partial w_i}{\partial x} = 0, \quad i = 1, \dots, m. \quad (6.15)$$

Para estas ecuaciones pueden aplicarse los métodos del caso escalar, tratados en el capítulo 3, como se muestra a continuación.

Usando el método upwind, sea $(w_i)_j^n \approx w_i(x_j, t_n)$, cada ecuación de (6.15) se discretiza como

$$(w_i)_j^{n+1} = (w_i)_j^n - \frac{\Delta t}{\Delta x} \lambda_i^+ [(w_i)_j^n - (w_i)_{j-1}^n] - \frac{\Delta t}{\Delta x} \lambda_i^- [(w_i)_{j+1}^n - (w_i)_j^n], \quad (6.16)$$

bajo la condición de estabilidad

$$|\lambda_i| \frac{\Delta t}{\Delta x} \leq 1. \quad (6.17)$$

Cabe recordar la definición de la parte positiva y negativa de un número real: $\lambda_i^+ = \max(\lambda_i, 0)$ y $\lambda_i^- = \min(\lambda_i, 0)$. Tomando las matrices diagonales $\Lambda^+ = \text{diag}(\lambda_1^+, \dots, \lambda_m^+)$ y $\Lambda^- = \text{diag}(\lambda_1^-, \dots, \lambda_m^-)$, la expresión (6.16) equivale a

$$w_j^{n+1} = w_j^n - \frac{\Delta t}{\Delta x} \Lambda^+ (w_j^n - w_{j-1}^n) - \frac{\Delta t}{\Delta x} \Lambda^- (w_{j+1}^n - w_j^n), \quad (6.18)$$

con $w_j^n = ((w_1)_j^n, \dots, (w_m)_j^n)^T \approx w(x_j, t_n)$. Finalmente, si $A^+ = R\Lambda^+R^{-1}$, $A^- = R\Lambda^-R^{-1}$ y $u_j^n = R w_j^n$, multiplicando (6.18) por R se deduce que el método upwind para sistemas viene dado por

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} A^+(u_j^n - u_{j-1}^n) - \frac{\Delta t}{\Delta x} A^-(u_{j+1}^n - u_j^n),$$

con Δt y Δx tales que

$$\left(\max_i |\lambda_i|\right) \frac{\Delta t}{\Delta x} = \rho(A) \frac{\Delta t}{\Delta x} \leq 1. \quad (6.19)$$

En cambio, si se resuelve (6.15) mediante el método de Lax-Wendroff escalar, se obtiene

$$(w_i)_j^{n+1} = (w_i)_j^n - \frac{\Delta t}{2\Delta x} \lambda_i [(w_i)_{j+1}^n - (w_i)_{j-1}^n] + \frac{\Delta t^2}{2\Delta x^2} \lambda_i [(w_i)_{j+1}^n - 2(w_i)_j^n + (w_i)_{j-1}^n], \quad (6.20)$$

donde los espaciados del mallado deben verificar de nuevo la relación (6.17). En forma vectorial, (6.20) se reduce a

$$w_j^{n+1} = w_j^n - \frac{\Delta t}{2\Delta x} \Lambda(w_{j+1}^n - w_{j-1}^n) + \frac{\Delta t^2}{2\Delta x^2} \Lambda^2(w_{j+1}^n - 2w_j^n + w_{j-1}^n).$$

De esta forma, el método de Lax-Wendroff vectorial es el siguiente:

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x} A(u_{j+1}^n - u_{j-1}^n) + \frac{\Delta t^2}{2\Delta x^2} A^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n),$$

bajo la restricción (6.19).

6.4. Teoría de sistemas no lineales

Tras el estudio de las ecuaciones lineales, se dedica esta sección a analizar las propiedades de los sistemas no lineales de leyes de conservación. Con ello, se pretende obtener cierta información que permita diseñar métodos numéricos adecuados para su resolución, mostrados en el apartado final del capítulo. De nuevo, se consideran problemas

$$\begin{aligned} u_t + f(u)_x &= 0, \\ u(x, 0) &= u^0(x), \end{aligned} \quad x \in \mathbb{R}, \quad t > 0 \quad (6.21)$$

estrictamente hiperbólicos, en los que, para cualquier vector u , la matriz $A(u) = f'(u) \in \mathbb{R}^{m \times m}$ tiene m valores propios reales y distintos; véanse (6.3) y (6.4).

Como en el caso escalar, para funciones suaves u^0 , la no linealidad de estas ecuaciones puede provocar discontinuidades en sus soluciones. Así, debe generalizarse el concepto clásico de solución (función de clase \mathcal{C}^1 que verifica la ecuación diferencial en todo punto), como se muestra a continuación.

Definición 9. Sea $u^0 \in L_{loc}^\infty(\mathbb{R})^m$, se dice que una función $u \in L_{loc}^\infty(\mathbb{R} \times [0, \infty))^m$ es solución suave del problema (6.21) si verifica la relación

$$\int_0^\infty \int_{\mathbb{R}} (u \cdot \varphi_t + f(u) \cdot \varphi_x) dx dt + \int_{\mathbb{R}} u_0(x) \cdot \varphi(x, 0) dx = 0$$

para toda función $\varphi \in \mathcal{C}_0^1(\mathbb{R} \times [0, \infty))^m$.

En esta definición, el punto \cdot denota el producto escalar euclídeo en \mathbb{R}^m . Por otro lado, los espacios L_{loc}^∞ y \mathcal{C}_0^1 se definen en la página 8.

El estudio se restringe a soluciones débiles de clase \mathcal{C}^1 a trozos: existe una cantidad finita de curvas $\Gamma_i \equiv x = \sigma_i(t)$ de clase \mathcal{C}^1 , para $i = 1, \dots, n$, de modo que la solución presenta

discontinuidades de salto en ellas, y es de clase \mathcal{C}^1 en el resto del dominio. Dado un punto (x_0, t_0) perteneciente a alguna curva Γ_{i_0} , con $s = \sigma'_{i_0}(t_0)$, la discontinuidad de salto debe verificar la condición de Rankine-Hugoniot:

$$f(u^+) - f(u^-) = s(u^+ - u^-), \quad u^\pm = \lim_{\delta \rightarrow 0^\pm} u(x_0 + \delta, t_0). \quad (6.22)$$

Además, puede comprobarse que una función \mathcal{C}^1 a trozos es solución débil de

$$u_t + f(u)_x = 0 \quad (6.23)$$

si verifica la ecuación diferencial donde es suave y la relación (6.22) en los puntos de discontinuidad.

Como las leyes de conservación admiten más de una solución débil (ver ejemplo 3), deben imponerse condiciones adicionales para obtener la solución de entropía, aquella que refleja el proceso físico estudiado. Un par de funciones $\eta : \mathbb{R}^m \rightarrow \mathbb{R}$ y $q : \mathbb{R}^m \rightarrow \mathbb{R}$ recibe el nombre de par de entropía para (6.23) si η es convexa y

$$\eta'(u) f'(u) = q'(u) \quad \forall u \in \mathbb{R}^m.$$

Análogamente al caso escalar, se acepta una solución débil de (6.23) si se cumple

$$\eta(u)_t + q(u)_x \leq 0 \quad (6.24)$$

para todo par de entropía (η, q) , en sentido distribucional. Para una función \mathcal{C}^1 a trozos, con la notación de (6.22), la relación (6.24) equivale a

$$s(\eta(u^+) - \eta(u^-)) \geq q(u^+) - q(u^-).$$

Una restricción alternativa, que generaliza la condición de entropía (2.33) de Lax, consiste en admitir tan solo discontinuidades que verifican

$$\lambda_i(u^+) < s < \lambda_i(u^-), \quad \lambda_{i-1}(u^-) < s < \lambda_{i+1}(u^+)$$

para cierto $i \in \{1, \dots, m\}$.

En la parte final del apartado se construye la solución del problema de Riemann asociado a (6.23): el problema de valor inicial

$$\begin{aligned} u_t + f(u)_x &= 0 \\ u(x, 0) &= \begin{cases} u^- & x < 0 \\ u^+ & x > 0 \end{cases} \end{aligned} \quad (6.25)$$

con vectores constantes $u^-, u^+ \in \mathbb{R}^m$.

Para cada $i \in \{1, \dots, m\}$, se asume la condición

$$\lambda'_i(u) \cdot r_i(u) > 0 \quad \forall u \quad (6.26)$$

o bien

$$\lambda'_i(u) \cdot r_i(u) = 0 \quad \forall u. \quad (6.27)$$

En el caso de sistemas estrictamente hiperbólicos que no verifican ninguna de estas dos relaciones, el estudio a realizar requiere de una mayor complejidad, fuera del alcance de este trabajo.

Fijado $\bar{u} \in \mathbb{R}^m$ y un índice i , se define $s \rightsquigarrow R_i(s)(\bar{u})$ como la solución del problema

$$u'(s) = r_i(u(s)), \quad u(0) = \bar{u}.$$

Por (6.26) y (6.27), el campo $\lambda_i : \mathbb{R}^m \rightarrow \mathbb{R}$ es estrictamente creciente o bien constante a lo largo de esta curva. Por otro lado, se denota por $s \rightsquigarrow S_i(s)(\bar{u})$ a la curva formada por aquellos vectores $u \in \mathbb{R}^m$ que cumplen la condición de Rankine-Hugoniot

$$f(u) - f(\bar{u}) = \lambda(u - \bar{u})$$

para cierto escalar λ . Antes de resolver (6.25) con vectores u^- y u^+ arbitrarios, se consideran tres casos iniciales, fijado $i \in \{1, \dots, m\}$:

1. Si este índice i verifica (6.26) y existe $\bar{s} > 0$ tal que $u^+ = R_i(\bar{s})(u^-)$. Para $s \in [0, \bar{s}]$, se define la aplicación

$$\gamma_i(s) = \lambda_i(R_i(s)(u^-)),$$

estrictamente creciente por hipótesis. Así, dado $\lambda \in [\gamma_i(0), \gamma_i(\bar{s})]$, existe un único $s \in [0, \bar{s}]$ tal que $\lambda = \gamma_i(s)$. La solución del problema de Riemann viene dada por

$$u(x, t) = \begin{cases} u^-, & x < \gamma_i(0)t, \\ R_i(s)(u^-), & x = \gamma_i(s)t, \quad s \in (0, \bar{s}), \\ u^+, & x > \gamma_i(\bar{s})t. \end{cases} \quad (6.28)$$

2. Si se cumple (6.26) y existe $\bar{s} < 0$ tal que $u^+ = S_i(\bar{s})(u^-)$, la solución será

$$u(x, t) = \begin{cases} u^-, & x < \lambda t \\ u^+, & x > \lambda t \end{cases}, \quad (6.29)$$

donde λ se fija a partir de la condición RH. Se toma \bar{s} porque la condición de entropía no se verifica para $\bar{s} > 0$.

3. Si el índice i satisface (6.27) y $u^+ = R_i(\bar{s})(u^-) = S_i(\bar{s})(u^-)$ para cierto \bar{s} real (las curvas R_i y S_i coinciden en este caso, para vectores r_i adecuados), la solución de entropía será de nuevo de la forma (6.29).

Fijados $u^- \in \mathbb{R}^m$ e $i \in \{1, \dots, m\}$, se define

$$\psi_i(s)(u^-) = \begin{cases} R_i(s)(u^-), & s \geq 0 \\ S_i(s)(u^+), & s < 0. \end{cases}$$

Como conclusión de los tres puntos anteriores, si u^+ pertenece a esta curva, el problema de Riemann (6.25) se resuelve mediante (6.28) o (6.29).

En general, si los vectores u^- y u^+ están suficientemente próximos, existen únicos valores s_1, \dots, s_m tales que

$$u^+ = \psi_m(s_m) \circ \dots \circ \psi_1(s_1)(u^-).$$

Definiendo

$$\omega_0 = u^-, \quad \omega_i = \psi_i(s_i)(\omega_{i-1}), \quad i = 1, \dots, m,$$

la solución del problema de Riemann se obtiene yuxtaponiendo las soluciones de los m problemas

$$\begin{aligned} u_t + f(u)_x &= 0 \\ u(x, 0) &= \begin{cases} \omega_{i-1}, & x < 0 \\ \omega_i, & x > 0 \end{cases}, \quad i = 1, \dots, m \end{aligned}$$

en distintos sectores del plano $x - t$, halladas siguiendo la explicación anterior.

6.5. Métodos numéricos para sistemas no lineales

Tras analizar los sistemas no lineales desde el punto de vista teórico, dedicamos esta última sección a presentar métodos numéricos que permiten resolverlos. El diseño de métodos para estos sistemas supone un reto mayúsculo, ya que sus soluciones resultan mucho más complejas que en el caso escalar o lineal.

En primer lugar se propone generalizar los métodos de volúmenes finitos estudiados para la ecuación escalar no lineal, en la sección 4.3. Para ello, dado el problema (6.21) se consideran los mallados en dominio espacial y temporal mostrados en la sección 4.1, y se pretende obtener aproximaciones de los promedios de la solución en cada celda:

$$u_j^n \approx \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_n) dx \in \mathbb{R}^m.$$

Fijado el espaciado Δx , una vez halladas las estimaciones u_j^n para cierto n , se toma Δt_n de modo que se cumpla la condición

$$\max_{i,j} |\lambda_i(u_j^n)| \frac{\Delta t}{\Delta x} \leq \frac{1}{2},$$

la extensión de (4.9) para sistemas de ecuaciones.

Posteriormente, siguiendo el método de Godunov, los valores u_j^{n+1} se calculan según la expresión

$$u_j^{n+1} = u_j^n - \frac{\Delta t_n}{\Delta x} (\hat{f}_{j+\frac{1}{2}}^n - \hat{f}_{j-\frac{1}{2}}^n),$$

donde $\hat{f}_{j+\frac{1}{2}}^n$ corresponde al valor de la solución del problema de Riemann

$$\begin{aligned} u_t + f'(u)u_x &= 0, \\ u(x, t_n) &= \begin{cases} u_j^n, & x < x_{j+\frac{1}{2}}, \\ u_{j+1}^n, & x > x_{j+\frac{1}{2}} \end{cases} \end{aligned} \quad (6.30)$$

en los puntos del segmento $\{x_{j+\frac{1}{2}}\} \times (t_n, t_{n+1})$. A diferencia de la ecuación escalar, en esta ocasión no es posible deducir la expresión general de los flujos $\hat{f}_{j+\frac{1}{2}}^n$, lo cual dificulta la aplicación de este método. Para los sistemas lineales, no obstante, sí es posible obtener estos valores, ya que en este caso el problema (6.30) puede resolverse de manera exacta, véase la sección 6.2. Además, se comprueba que este método aplicado a un problema lineal coincide con el método upwind de la sección 6.3.

Como extensión del método de Roe, se propone sustituir el término no lineal $f'(u)$ en (6.30) por una matriz $A_{j+\frac{1}{2}}^n$ que dependa de u_j^n y u_{j+1}^n . En [9] se argumenta que $A_{j+\frac{1}{2}}^n$ debe cumplir las siguientes propiedades:

- $A_{j+\frac{1}{2}}^n (u_{j+1}^n - u_j^n) = f(u_{j+1}^n) - f(u_j^n)$.
- $A_{j+\frac{1}{2}}^n$ sea diagonalizable con valores propios reales.
- $A_{j+\frac{1}{2}}^n \rightarrow f'(\bar{u})$ suavemente si $u_j^n, u_{j+1}^n \rightarrow \bar{u}$.

Aunque pueden obtenerse matrices que verifican estas condiciones, sus expresiones tan complejas hacen inviable la puesta en práctica de esta linealización.

Por otro lado, generalizando el método de Rusanov, se propone aproximar la solución de (6.30) por una función a trozos de la forma

$$\bar{w}_j^n(x, t) = \begin{cases} u_j^n, & x - x_{j+\frac{1}{2}} < -s_{j+\frac{1}{2}}(t - t_n), \\ u_{j+\frac{1}{2}}^{n*}, & -s_{j+\frac{1}{2}}(t - t_n) < x - x_{j+\frac{1}{2}} < s_{j+\frac{1}{2}}(t - t_n), \\ u_{j+1}^n, & x - x_{j+\frac{1}{2}} < s_{j+\frac{1}{2}}(t - t_n), \end{cases}$$

con $u_{j+\frac{1}{2}}^{n*}$ y $s_{j+\frac{1}{2}}$ por determinar. Mediante la condición de Rankine y Hugoniot se considera

$$\hat{f}_{j+\frac{1}{2}}^n = \hat{f}^{Rus}(u_j^n, u_{j+1}^n) = \frac{f(u_j^n) + f(u_{j+1}^n)}{2} - \frac{s_{j+\frac{1}{2}}}{2} (u_{j+1}^n - u_j^n), \quad (6.31)$$

donde el término $s_{j+\frac{1}{2}}$ puede tomarse como

$$s_{j+\frac{1}{2}} = \max_i (|\lambda_i(u_j^n)|, |\lambda_i(u_{j+1}^n)|). \quad (6.32)$$

Finalmente, en cada celda $c_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ puede definirse una función lineal

$$p_{n,j}(x) = u_j^n + \sigma_j^n(x - x_j)$$

con pendiente $\sigma_j^n \in \mathbb{R}^m$ adecuada. A partir de ésta, se sustituyen las estimaciones u_j^n y u_{j+1}^n en (6.31) y (6.32) por los valores $p_{n,j}(x_{j+\frac{1}{2}})$ y $p_{n,j+1}(x_{j+\frac{1}{2}})$ respectivamente. Componente a componente, también pueden aplicarse las técnicas de reconstrucción ENO y WENO presentadas en el capítulo 5.

Cabe notar que las soluciones de los problemas de Riemann, con expresiones complejas mostradas en la sección 6.4, se han estimado mediante funciones constantes a trozos demasiado sencillas. Así, los resultados obtenidos a través de estos métodos no aproximan con demasiada precisión algunas discontinuidades presentes en los sistemas de leyes de conservación.

Por esta razón, los métodos de volúmenes finitos no resultan ser una buena opción a la hora de resolver numéricamente estos sistemas no lineales. Para ello, se emplea el método de diferencias finitas de Shu y Osher, explicado en la sección 5.5, en su versión vectorial, en la que es suficiente aplicar la reconstrucción de alto orden \mathcal{R} componente a componente.

Capítulo 7

Aplicación a un modelo de cromatografía

En los capítulos previos se ha realizado una introducción teórica y un estudio numérico de las ecuaciones y sistemas hiperbólicos de leyes de conservación. Como aplicación, en este apartado se muestra la resolución de un sistema que modeliza el proceso de cromatografía. Para ello se ha tomado el artículo [8] como referencia.

En la primera sección se describe este proceso químico y se deducen las ecuaciones del modelo a tratar. Posteriormente se analiza la estructura matemática de dicho modelo, con el objetivo de obtener cierta información relevante que permita resolverlo. Por último, se presenta el método numérico empleado en [8] y se llevan a cabo distintos experimentos acerca de éste.

7.1. Introducción

La cromatografía es una herramienta potente para la separación de gases o líquidos de una mezcla. Surgió a principios del siglo pasado con el objetivo de extraer ciertos pigmentos a partir de compuestos de origen vegetal. La industria farmacéutica, en la que se requieren altos índices de pureza en los productos, ha mostrado gran interés por esta técnica en los últimos años.

Para el proceso de cromatografía estudiado en este trabajo, se introduce un medio poroso (fase estacionaria o sólida) en una columna cilíndrica. A continuación se inyecta la mezcla (soluto) a descomponer, seguida de un flujo continuo de líquido (solvente o fase móvil) que la conduce a lo largo de la columna. La mezcla entra así en contacto con el medio poroso, y los distintos componentes se separan según el grado de retención (adsorción) que éste ejerce sobre ellos.

Si no hay reacciones en el sistema, la cantidad de cada componente de la mezcla se conserva, de modo que pueden aplicarse las ecuaciones del balance de masas. Para deducir estas ecuaciones deben asumirse ciertas hipótesis que ahora se enuncian, y que en ningún caso limitan la validez del modelo [3].

1. La columna es radialmente homogénea, es decir, sus propiedades se conservan en cualquier sección transversal, y lo mismo ocurre con las concentraciones de cada sustancia. De esta forma, la columna puede considerarse unidimensional.
2. El solvente es un fluido incompresible.
3. No existe adsorción entre el solvente y el medio poroso.
4. El medio poroso es homogéneo en la columna. Este hecho implica que el proceso de adsorción sea uniforme en el tiempo.
5. Hay un equilibrio constante entre la fase móvil y la fase sólida a lo largo de la columna.

6. Las bandas de los distintos componentes de la mezcla tan solo se forman por dispersión. Por ejemplo, no se tiene en cuenta el efecto del calor que se desprende en el proceso de adsorción.
7. El procedimiento se lleva a cabo a temperatura y presión constantes. También se asume constante la velocidad de la fase móvil.

Las ecuaciones que modelizan el proceso tienen, así, dos variables independientes, el tiempo t y la posición o altura en la columna z . Se toma una columna de altura unidad, de modo que $z = 0$ representa la parte superior y $z = 1$, la parte inferior. Dada una mezcla de m componentes, se denota por $c_i = c_i(z, t)$ y $q_i = q_i(z, t)$ a la concentración del componente i -ésimo en la fase móvil y sólida respectivamente, a tiempo t y posición z . A continuación se deduce la ecuación que deben verificar las funciones c_i y q_i .

Para aplicar el balance de masas en la fase móvil, se considera una porción de columna, un cilindro de altura Δz . De esta forma, en un cierto intervalo de tiempo, se afirma que la diferencia entre la cantidad de componente que entra y sale de la porción equivale a la cantidad acumulada en ella.

En primer lugar, el flujo que entra en la porción viene dado por

$$E_i(z, t) = \varepsilon S \left(uc_i - D_a \frac{\partial c_i}{\partial z} \right) (z, t).$$

En esta expresión, $\varepsilon \in (0, 1)$ es un índice de la porosidad de la fase sólida y S , el área de la sección transversal del cilindro. Por otro lado, u hace referencia a la velocidad del solvente y $D_a = u/(2N_t) > 0$, al coeficiente de dispersión, siendo N_t el número de platos que forman la columna, ver [3]. Por su parte, el flujo que sale de la porción es

$$S_i(z, t) = E_i(z + \Delta z, t) = \varepsilon S \left(uc_i - D_a \frac{\partial c_i}{\partial z} \right) (z + \Delta z, t).$$

Finalmente,

$$A_i(z, t) = S \Delta z \left(\varepsilon \frac{\partial c_i}{\partial t} + (1 - \varepsilon) \frac{\partial q_i}{\partial t} \right) (\bar{z}, t)$$

corresponde a la tasa de acumulación de la sustancia en el cilindro de volumen $S \Delta z$, donde \bar{z} es el valor medio de z en esta porción.

Así, la ley de conservación

$$E_i(z, t) - S_i(z, t) = A_i(z, t)$$

equivale a

$$\varepsilon S \left(uc_i - D_a \frac{\partial c_i}{\partial z} \right) (z, t) - \varepsilon S \left(uc_i - D_a \frac{\partial c_i}{\partial z} \right) (z + \Delta z, t) = S \Delta z \left(\varepsilon \frac{\partial c_i}{\partial t} + (1 - \varepsilon) \frac{\partial q_i}{\partial t} \right) (\bar{z}, t).$$

A su vez, esta expresión puede reescribirse como

$$\frac{\partial c_i}{\partial t} (\bar{z}, t) + \frac{1 - \varepsilon}{\varepsilon} \frac{\partial q_i}{\partial t} (\bar{z}, t) = \frac{D_a}{\Delta z} \left(\frac{\partial c_i}{\partial z} (z + \Delta z, t) - \frac{\partial c_i}{\partial z} (z, t) \right) - \frac{u}{\Delta z} (c_i(z + \Delta z, t) - c_i(z, t)).$$

Tomando límites cuando $\Delta z \rightarrow 0$ se obtiene la ecuación

$$\frac{\partial c_i}{\partial t} + \frac{1 - \varepsilon}{\varepsilon} \frac{\partial q_i}{\partial t} + u \frac{\partial c_i}{\partial z} = D_a \frac{\partial^2 c_i}{\partial z^2}, \quad i = 1, \dots, m. \quad (7.1)$$

Sea $c = (c_1, \dots, c_m)^T$ y $q = (q_1, \dots, q_m)^T$, las ecuaciones en (7.1) equivalen a

$$\frac{\partial c}{\partial t} + \frac{1 - \varepsilon}{\varepsilon} \frac{\partial q}{\partial t} + u \frac{\partial c}{\partial z} = D_a \frac{\partial^2 c}{\partial z^2}. \quad (7.2)$$

Es necesario añadir condiciones de frontera adecuadas a (7.2). Siguiendo [8], dada una función conocida c_{inj} , se impone

$$uc - D_a \frac{\partial c}{\partial z} \Big|_{z=0} = uc_{inj}(t), \quad \frac{\partial c}{\partial z} \Big|_{z=1} = 0. \quad (7.3)$$

Cabe notar que la ecuación (7.1) contiene dos funciones, c_i y q_i , pues es necesario establecer una relación entre ellas para su resolución. Como en el proceso de cromatografía se alcanza un equilibrio rápidamente [3], para cada $i \in \{1, \dots, m\}$ se asume la condición

$$q_i = q_i(c_1, \dots, c_m),$$

que ahora se especifica.

En equilibrio, una parte de la superficie está ocupada por el solvente y el resto, por cada uno de los componentes de la mezcla. Denotando por S_0 a la fracción de superficie ocupada por el solvente y, por S_i a la fracción cubierta por el componente i -ésimo, se tiene

$$S_0 + S_1 + \dots + S_m = 1. \quad (7.4)$$

En este estado, la tasa de adsorción de cada componente por el medio poroso coincide con la tasa de eliminación o desorción. La tasa de adsorción es proporcional a la concentración del componente y de la fracción S_0 . Por otro lado, la tasa de desorción es proporcional a la fracción de superficie ocupada por el componente. Así, se tiene

$$k_{a,i} S_0 c_i = k_{d,i} S_i, \quad (7.5)$$

donde $k_{a,i}$ y $k_{d,i}$ corresponden a las constantes de adsorción y desorción respectivamente del componente i -ésimo. Mediante (7.4), la expresión (7.5) equivale a

$$k_{a,i} c_i (1 - S_1 - \dots - S_m) = k_{d,i} S_i, \quad i = 1, \dots, m.$$

Resolviendo el sistema de ecuaciones para S_i se deduce

$$S_i = \frac{\frac{k_{a,i}}{k_{d,i}} c_i}{1 + \sum_{j=1}^m \frac{k_{a,j}}{k_{d,j}} c_j}.$$

Finalmente, la cantidad del componente adsorbida es proporcional a la fracción S_i de superficie ocupada por éste, de modo que

$$q_i = q_i(c) = \frac{a_i c_i}{1 + \sum_{j=1}^m b_j c_j}, \quad i = 1, \dots, m \quad (7.6)$$

para ciertos coeficientes constantes $a_i, b_i > 0$.

El modelo (7.2)–(7.6) puede reescribirse como

$$\frac{\partial w}{\partial t} + \frac{\partial(uc)}{\partial z} = D_a \frac{\partial^2 c}{\partial z^2}, \quad w = W(c) = c + \frac{1-\varepsilon}{\varepsilon} q(c). \quad (7.7)$$

En la siguiente sección del capítulo, se demuestra que $W(c)$ admite función inversa de clase \mathcal{C}^1 , denotada por $C(w)$. Así, la ecuación (7.7) equivale a

$$\frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial z} = D_a \frac{\partial^2 C(w)}{\partial z^2}, \quad f(w) = uC(w), \quad (7.8)$$

un sistema de leyes de conservación cuando D_a se anula. Posteriormente se demuestra que este sistema es estrictamente hiperbólico, de modo que pueden aplicarse las técnicas estudiadas a lo largo del trabajo, junto con una discretización estándar del término parabólico, como se muestra en la sección 7.3.

7.2. Análisis matemático del modelo

Como se ha indicado anteriormente, en este capítulo se prueba que $W(c) = (W_i(c))_{i=1}^m : \mathbb{R}_+^m \rightarrow \mathbb{R}_+^m$, función de clase \mathcal{C}^1 definida por

$$W_i(c) = c_i \left(1 + \frac{\eta_i}{1 + \sum_{j=1}^m b_j c_j} \right), \quad \eta_i = \frac{1 - \varepsilon}{\varepsilon} a_i, \quad i = 1, \dots, m \quad (7.9)$$

tiene inversa $C(w)$ también de clase \mathcal{C}^1 , siendo $\mathbb{R}_+ = [0, \infty)$. Aunque no es posible hallar su expresión explícita, para cualquier $w \in (0, \infty)^m$, el valor $C(w)$ puede obtenerse mediante el cálculo de la única raíz positiva de una cierta función racional. Como consecuencia de este resultado se deducirá que el sistema (7.8) con $D_a = 0$ es estrictamente hiperbólico.

Dada la función continua $\varphi(c) = 1 + b^T c = 1 + b_1 c_1 + \dots + b_m c_m$, se considera el conjunto abierto

$$\mathcal{D}_W = \{c \in \mathbb{R}^m : \varphi(c) > 0\} = \{c \in \mathbb{R}^m : 1 + b^T c > 0\},$$

que verifica $\mathbb{R}_+^m \subset \mathcal{D}_W$. Así, para demostrar la existencia de inversa local en cada punto de \mathbb{R}_+^m , es suficiente comprobar que la matriz jacobiana $W'(c)$ es regular para todo $c \in \mathbb{R}_+^m$, como se muestra a continuación. De ahora en adelante se supone que los componentes de la mezcla están ordenados de forma que $0 < a_1 < a_2 < \dots < a_m$.

Teorema 6. *Fijado $c \in (0, \infty)^m$ arbitrario, la matriz jacobiana $W'(c)$ es diagonalizable con valores propios $\lambda_1, \dots, \lambda_m$ distintos, reales, estrictamente positivos y que verifican*

$$1 < \lambda_1 < d_1 < \lambda_2 < \dots < d_{m-1} < \lambda_m < d_m, \quad d_i = 1 + \frac{\eta_i}{1 + \sum_{j=1}^m b_j c_j}. \quad (7.10)$$

Demostración. En primer lugar, recordando que $\varphi(c) = 1 + b^T c$, se halla la matriz jacobiana de $W(c)$ para cualquier $c \in \mathbb{R}_+^m$, denotada por $J(c)$.

$$\begin{aligned} J_{ij}(c) &= \frac{\partial W_i(c)}{\partial c_j} = \frac{\partial c_i}{\partial c_j} \left(1 + \frac{\eta_i}{\varphi(c)} \right) - \frac{c_i \eta_i}{\varphi(c)^2} \frac{\partial \varphi(c)}{\partial c_j} \\ &= \delta_{ij} \left(1 + \frac{\eta_i}{\varphi(c)} \right) - \frac{c_i \eta_i b_j}{\varphi(c)^2} \end{aligned} \quad i, j = 1, \dots, m.$$

En la expresión anterior, δ_{ij} hace referencia al delta de Kronecker. De esta forma, definiendo

$$d_i = d_i(c) = 1 + \frac{\eta_i}{\varphi(c)}, \quad \tau_i = \tau_i(c) = -\frac{\eta_i c_i}{\varphi(c)^2}, \quad i = 1, \dots, m,$$

la matriz jacobiana equivale a

$$J = D + \tau b^T, \quad D = \text{diag}(d_i)_{i=1}^m.$$

Fijado $c \in (0, \infty)^m$, se considera la función racional

$$S_c(\lambda) = 1 + b^T (D - \lambda I)^{-1} \tau = 1 + \frac{1}{\varphi(c)^2} \sum_{i=1}^m \frac{\eta_i b_i c_i}{\lambda - d_i}.$$

Los polos de esta función, d_i , verifican la relación $d_1 < d_2 < \dots < d_m$, ya que $a_1 < \dots < a_m$ por hipótesis. En efecto, para $i = 1, \dots, m-1$ se tiene

$$d_i = 1 + \frac{\eta_i}{\varphi(c)} = 1 + \frac{1}{\varphi(c)} \frac{1 - \varepsilon}{\varepsilon} a_i < 1 + \frac{1}{\varphi(c)} \frac{1 - \varepsilon}{\varepsilon} a_{i+1} = d_{i+1}.$$

Por otro lado, se verifica:

- $S_c(\pm\infty) = 1$
- $\lim_{\lambda \rightarrow d_i^\pm} S_c(\lambda) = \lim_{\lambda \rightarrow d_i^\pm} \left(1 + \frac{1}{\varphi(c)^2} \left[\sum_{\substack{j=1 \\ j \neq i}}^m \frac{\eta_j b_j c_j}{\lambda - d_j} + \frac{\eta_i b_i c_i}{\lambda - d_i} \right] \right) = \pm\infty$
- $S_c(1) = 1 + \frac{1}{\varphi(c)^2} \sum_{i=1}^m \frac{\eta_i b_i c_i}{1 - d_i} = 1 - \frac{1}{\varphi(c)} \sum_{i=1}^m b_i c_i = \frac{1}{\varphi(c)} > 0.$

Así, existen al menos m raíces de S_c , denotadas por $\lambda_1, \dots, \lambda_m$, que satisfacen

$$1 < \lambda_1 < d_1 < \lambda_2 < d_2 < \dots < d_{m-1} < \lambda_m < d_m. \quad (7.11)$$

En la Figura 7.1 se representa un esbozo de la gráfica de esta función.

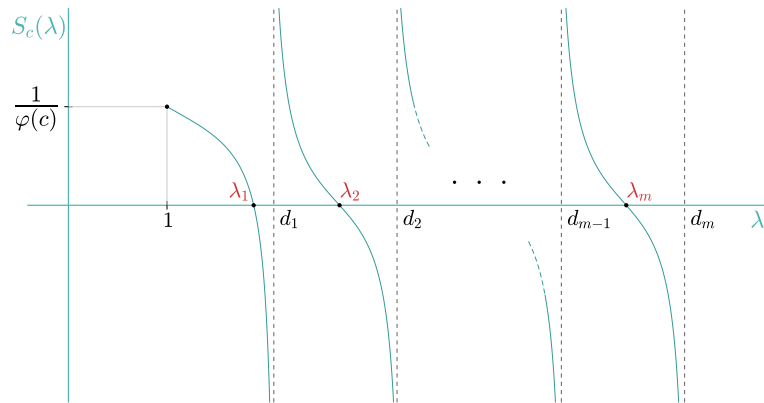


Figura 7.1: Gráfica de la función racional $S_c(\lambda)$.

Además, éstas son las únicas raíces de $S_c(\lambda)$, ya que todo cero de S_c también lo es del polinomio de grado m siguiente:

$$Q(\lambda) = S_c(\lambda) \prod_{i=1}^m (\lambda - d_i).$$

El argumento previo demuestra que, para todo $c \in (0, \infty)^m$, la función racional S_c tiene m raíces $\lambda_1, \dots, \lambda_m$ reales, distintas, estrictamente positivas que verifican la condición (7.10). De esta forma, es suficiente comprobar que cada λ_i es valor propio de la matriz jacobiana $J(c) = W'(c)$.

Fijado $i \in \{1, \dots, m\}$ y $c \in (0, \infty)^m$, se considera la matriz $D - \lambda_i I$. Se trata de una matriz invertible porque su determinante es distinto de cero:

$$\det(D - \lambda_i I) = \prod_{j=1}^m (d_j - \lambda_i) \neq 0.$$

Tomando $x^i = -(D - \lambda_i I)^{-1} \tau \neq 0$, se deduce

$$0 = S_c(\lambda_i) = 1 + b^T (D - \lambda_i I)^{-1} \tau = 1 - b^T x^i \quad \longrightarrow \quad b^T x^i = 1,$$

y como consecuencia,

$$Jx^i = (D + \tau b^T)x^i = Dx^i + \tau b^T x^i = Dx^i + \tau = \lambda_i x^i.$$

Así, λ_i es valor propio de $J = J(c)$, lo que concluye la prueba. \square

Nota 4. Puede comprobarse de manera sencilla que $W'(c)$ es regular si $c \in \mathbb{R}_+^m = [0, \infty)^m$, no solo para aquellos vectores con cada componente estrictamente positiva. Por el teorema de la función inversa, existe una inversa de W de clase C^1 en un entorno de cada punto de \mathbb{R}_+^m .

Para demostrar que $W(c)$ admite función inversa global $C : \mathbb{R}_+^m \rightarrow \mathbb{R}_+^m$, se muestra el siguiente resultado previo.

Lema 2. Sea $w \in \mathbb{R}_+^m$, la función racional

$$R_w(y) = 1 - y + \sum_{i=1}^m \frac{y}{y + \eta_i} b_i w_i \quad (7.12)$$

tiene una única raíz positiva, denotada por $\rho_0(w)$. Esta raíz cumple $1 \leq \rho_0(w) \leq \varphi(w) = 1 + b^T w$.

Demostración. Para $w = 0$ el resultado es obvio. Por otro lado, si existe $i \in \{1, \dots, m\}$ tal que $w_i = 0$, el resultado se prueba de manera muy similar al caso $w_i > 0 \forall i$, el que se trata a continuación. La función definida en (7.12) satisface:

- $\lim_{y \rightarrow -\eta_i^\pm} R_w(y) = 1 + \eta_i - \eta_i \lim_{y \rightarrow -\eta_i^\pm} \left(\sum_{\substack{j=1 \\ j \neq i}}^m \frac{b_j w_j}{y + \eta_j} + \frac{b_i w_i}{y + \eta_i} \right) = \mp \infty, \quad i = 1, \dots, m$
- $\lim_{y \rightarrow \pm \infty} R_w(y) = 1 \mp \infty + b^T w = \mp \infty$
- $R_w(0) = 1$

Como R_w es continua en su dominio, tiene al menos $m + 1$ raíces reales ρ_0, \dots, ρ_m , que cumplen

$$-\eta_m < \rho_m < \dots < -\eta_1 < \rho_1 < 0 < \rho_0.$$

Se muestra un esbozo de la gráfica en la Figura 7.2.

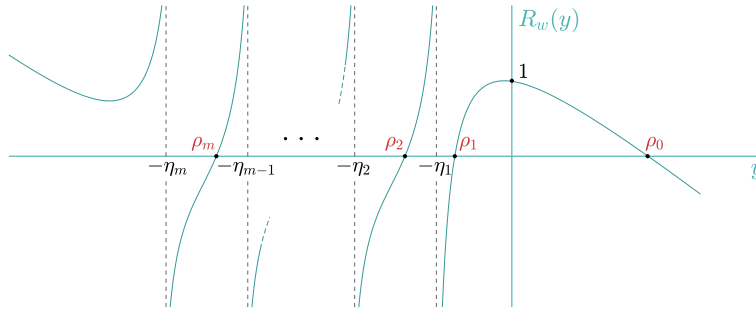


Figura 7.2: Gráfica de la función racional R_w .

Siguiendo el mismo razonamiento de la prueba anterior, como toda raíz de R_w también lo es del polinomio

$$Q(y) = R_w(y) \prod_{i=1}^m (y + \eta_i),$$

de grado $m + 1$, los valores ρ_0, \dots, ρ_m son todos los ceros de R_w . Así, se deduce que la función racional (7.12) tiene una única raíz positiva, denotada por ρ_0 . Se comprueba ahora que ésta verifica la relación $1 \leq \rho_0 \leq 1 + b^T w$.

Para $\varphi(w) = 1 + b^T w > 0$ se tiene

$$R_w(\varphi(w)) = -b^T w + \sum_{i=1}^m \frac{\varphi(w)}{\varphi(w) + \eta_i} b_i w_i = \sum_{i=1}^m \left(\frac{\varphi(w)}{\varphi(w) + \eta_i} - 1 \right) b_i w_i = - \sum_{i=1}^m \frac{\eta_i b_i w_i}{\varphi(w) + \eta_i} < 0,$$

pues $\rho_0 \leq 1 + b^T w$. Por otro lado, como

$$R_w(1) = \sum_{i=1}^m \frac{b_i w_i}{1 + \eta_i} > 0,$$

se concluye $1 \leq \rho_0$. □

Teorema 7. La función $W : \mathbb{R}_+^m \rightarrow \mathbb{R}_+^m$ definida en (7.9) es invertible. Su función inversa $C = W^{-1} : \mathbb{R}_+^m \rightarrow \mathbb{R}_+^m$ es de clase C^1 y viene dada por la expresión

$$C_i(w) = \frac{w_i}{1 + \eta_i/\rho_0(w)}, \quad i = 1, \dots, m,$$

donde $\rho_0(w)$ es la única raíz positiva de la función racional R_w en (7.12).

Demostración. Considérese un vector $c = (c_1, \dots, c_m)^T \in \mathbb{R}_+^m$, con $\varphi(c) = 1 + b^T c$. Por 7.9,

$$W_i(c) = c_i \left(1 + \frac{\eta_i}{\varphi(c)} \right) \equiv c_i = \frac{W_i(c)}{1 + \eta_i/\varphi(c)}.$$

Así,

$$\varphi(c) - 1 = \sum_{i=1}^m b_i c_i = \sum_{i=1}^m \frac{b_i W_i(c)}{1 + \eta_i/\varphi(c)},$$

pues

$$0 = 1 - \varphi(c) + \varphi(c) \sum_{i=1}^m \frac{b_i W_i(c)}{\varphi(c) + \eta_i} = R_{W(c)}(\varphi(c)).$$

Como $\varphi(c) > 0$, ésta debe ser la única raíz positiva de $R_{W(c)}$, es decir,

$$\varphi(c) = \rho_0(W(c)).$$

De esta forma, se define la función $C : \mathbb{R}_+^m \rightarrow \mathbb{R}_+^m$ como

$$C_i(w) = \frac{w_i}{1 + \eta_i/\rho_0(w)}, \quad i = 1, \dots, m, \quad (7.13)$$

que verifica

$$C_i(W(c)) = \frac{W_i(c)}{1 + \eta_i/\rho_0(W(c))} = \frac{W_i(c)}{1 + \eta_i/\varphi(c)} = c_i, \quad i = 1, \dots, m, \quad c \in \mathbb{R}_+^m.$$

Finalmente, dado $w \in \mathbb{R}_+^m$,

$$\begin{aligned} \varphi(C(w)) &= 1 + \sum_{i=1}^m b_i C_i(w) = 1 + \sum_{i=1}^m b_i \frac{w_i}{1 + \eta_i/\rho_0(w)} \\ &= 1 + \rho_0(w) \sum_{i=1}^m \frac{b_i w_i}{\rho_0(w) + \eta_i} = R_w(\rho_0(w)) + \rho_0(w) = \rho_0(w). \end{aligned}$$

Esto implica

$$W_i(C(w)) = C_i(w) \left(1 + \frac{\eta_i}{\varphi(C(w))} \right) = C_i(w) \left(1 + \frac{\eta_i}{\rho_0(w)} \right) = w_i, \quad i = 1, \dots, m.$$

Se ha comprobado, así, que la función C definida en 7.13 es la inversa de $W = W(c)$. Por la nota 4 se deduce que esta función inversa es de clase C^1 . \square

Nota 5. Aunque no se tiene la expresión explícita de la raíz $\rho_0(w)$, dado un vector w , su valor puede aproximarse mediante el método de Newton a partir de la acotación $1 \leq \rho_0(w) \leq 1 + b^T w$ deducida en el lema 2. De esta forma, es posible obtener $C(w)$ para cualquier $w \in \mathbb{R}_+^m$.

Corolario 1. El sistema de leyes de conservación

$$\frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial z} = 0, \quad f(w) = uC(w), \quad u > 0$$

es estrictamente hiperbólico para $w \in (0, \infty)^m$. Además de reales y distintos, los valores propios de $f'(w)$ son positivos y están acotados superiormente por u .

Demostración. Por el teorema 7, la matriz jacobiana de $C(w)$ es la inversa de $W'(c)$. Entonces, para $w \in (0, \infty)^m$, su espectro está formado por $1/\lambda_1, \dots, 1/\lambda_m$, siendo $\lambda_1, \dots, \lambda_m$ los valores propios de $W'(c)$. Por (7.11) se cumple

$$1 > \frac{1}{\lambda_1} > \frac{1}{d_1} > \frac{1}{\lambda_2} > \dots > \frac{1}{d_{m-1}} > \frac{1}{\lambda_m} > \frac{1}{d_m} > 0.$$

Finalmente, los valores propios de $f(w) = uC(w)$ son de la forma $\mu_i = u/\lambda_i$, y verifican

$$u > \mu_1 > \mu_2 > \dots > \mu_m > 0.$$

□

7.3. Métodos numéricos para la resolución del modelo

En la sección anterior se ha comprobado que el sistema de ecuaciones que modeliza el proceso de cromatografía equivale a

$$\frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial z} = D_a \frac{\partial^2 C(w)}{\partial z^2}, \quad f(w) = uC(w), \quad (7.14)$$

donde $C(w)$ puede obtenerse a partir del cálculo numérico de una raíz de una cierta función racional. Además, se ha demostrado que, para $D_a = 0$, el sistema (7.14) es estrictamente hiperbólico.

Este modelo tan solo puede resolverse analíticamente bajo condiciones muy particulares. Por ello, se han desarrollado métodos numéricos que aproximan su solución de manera bastante precisa, mostrados a continuación. En la práctica, estos métodos permiten llevar a cabo simulaciones que sustituyen los experimentos de prueba y error.

Para esta sección, es conveniente reescribir (7.14) como

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial z}(f(w) - g(w)) = 0, \quad g(w) = D_a \frac{\partial C(w)}{\partial z}. \quad (7.15)$$

Con esta notación, las condiciones de frontera definidas en (7.3) toman la forma

$$(f(w) - g(w))(0, t) = uc_{in,j}(t), \quad g(1, t) = 0. \quad (7.16)$$

Por último, cabe destacar que (7.15) y (7.16) deben completarse con una condición inicial $w(z, 0) = w_0(z)$.

Dado J natural, en el dominio espacial, el intervalo $[0, 1]$, se considera el mallado formado por los puntos

$$z_j = \left(j - \frac{1}{2}\right)\Delta z, \quad \Delta z = \frac{1}{J}, \quad j = 1, \dots, J.$$

Fijado $z = z_j$, por (7.15),

$$\begin{aligned} \frac{\partial}{\partial t} w(z_j, t) &= -\frac{\partial f(w)}{\partial z}(z_j, t) + \frac{\partial g(w)}{\partial z}(z_j, t) \\ &\approx -\frac{1}{\Delta z} \left[f(w(z_{j+\frac{1}{2}}, t)) - f(w(z_{j-\frac{1}{2}}, t)) \right] + \frac{1}{\Delta z} \left[g(w(z_{j+\frac{1}{2}}, t)) - g(w(z_{j-\frac{1}{2}}, t)) \right], \end{aligned}$$

siendo $z_{j+\frac{1}{2}} = z_j + \Delta z/2$. Así, se define $w_j(t) \approx w(z_j, t) \in \mathbb{R}^m$ como la solución de

$$w_j'(t) = -\frac{1}{\Delta z} \left[\hat{f}_{j+\frac{1}{2}}(w(t)) - \hat{f}_{j-\frac{1}{2}}(w(t)) \right] + \frac{1}{\Delta z} \left[\hat{g}_{j+\frac{1}{2}}(w(t)) - \hat{g}_{j-\frac{1}{2}}(w(t)) \right] \quad (7.17)$$

con $w(t) = (w_1(t); \dots; w_J(t)) \in \mathbb{R}^{m \times J}$. Denotando

$$\mathcal{L}_j(w(t)) = -\frac{1}{\Delta z} \left[\hat{f}_{j+\frac{1}{2}}(w(t)) - \hat{f}_{j-\frac{1}{2}}(w(t)) \right], \quad \mathcal{D}_j(w(t)) = \frac{1}{\Delta z} \left[\hat{g}_{j+\frac{1}{2}}(w(t)) - \hat{g}_{j-\frac{1}{2}}(w(t)) \right], \quad (7.18)$$

la expresión (7.17) se reduce a

$$w'_j(t) = \mathcal{L}_j(w(t)) + \mathcal{D}_j(w(t)), \quad j = 1, \dots, J.$$

Los términos $\hat{f}_{j+\frac{1}{2}}(w(t))$ y $\hat{g}_{j+\frac{1}{2}}(w(t))$, que ahora se especifican, hacen referencia a aproximaciones de $f(w(z_{j+\frac{1}{2}}, t))$ y $g(w(z_{j+\frac{1}{2}}, t))$ respectivamente.

En primer lugar, en este contexto las condiciones de frontera (7.16) se traducen en

$$\hat{f}_{\frac{1}{2}} - \hat{g}_{\frac{1}{2}} = uc_{inj}, \quad \hat{g}_{m+\frac{1}{2}} = 0.$$

De este modo se fija, obviamente, $\hat{g}_{m+\frac{1}{2}} = 0$ y, por otro lado, $\hat{f}_{\frac{1}{2}} = uc_{inj}$ y $\hat{g}_{\frac{1}{2}} = 0$. El resto de flujos $\hat{f}_{j+\frac{1}{2}}$ pueden obtenerse mediante una técnica de reconstrucción de orden alto, siguiendo el algoritmo 2 de la sección 5.5. Como los valores propios de $f(w)$ están acotados por u según el corolario 1, omitiendo la dependencia temporal $w_j = w_j(t)$, se toma

$$\begin{aligned} \hat{f}_{j+\frac{1}{2}}^+ &= \mathcal{R} \left(\frac{1}{2}(f(w_{j-2}) + uw_{j-2}), \dots, \frac{1}{2}(f(w_{j+2}) + uw_{j+2}) \right) \\ \hat{f}_{j+\frac{1}{2}}^- &= \mathcal{R} \left(\frac{1}{2}(f(w_{j-1}) - uw_{j-1}), \dots, \frac{1}{2}(f(w_{j+3}) - uw_{j+3}) \right), \quad j = 1, \dots, J, \\ \hat{f}_{j+\frac{1}{2}} &= \hat{f}_{j+\frac{1}{2}}^+ + \hat{f}_{j+\frac{1}{2}}^- \end{aligned}$$

donde, en este caso, \mathcal{R} corresponde al método WENO de orden 5. Finalmente, teniendo en cuenta la definición de g en (7.15), los flujos $\hat{g}_{j+\frac{1}{2}}$ para $j = 1, \dots, J-1$ se calculan según la expresión

$$\hat{g}_{j+\frac{1}{2}} = D_a \frac{C(w_{j+1}) - C(w_j)}{\Delta z}.$$

A la hora de calcular los flujos $\hat{f}_{j+\frac{1}{2}}$ es necesario conocer las estimaciones w_j para $j = -1, 0, J+1, J+2, J+3$. A continuación se muestra un procedimiento que permite hallar los valores c_j asociados, a partir de los cuales puede obtenerse cada w_j como $W(c_j)$.

Dado uno de los cinco índices j anteriores, este procedimiento consiste en encontrar un polinomio de primer grado que verifique una condición de frontera de (7.3) (la condición en $z = 0$ para $j = -1, 0$ y en $z = 1$ si $j = J+1, J+2, J+3$) e interpole a c en el simétrico de $z_j = (j+1/2)\Delta z$ respecto del extremo correspondiente ($z = 0$ o bien $z = 1$). El valor c_j será el resultado de evaluar dicho polinomio en el punto $z = z_j$.

Así, por ejemplo, para hallar el valor c_0 , se usa la condición de frontera

$$uc_{\frac{1}{2}} - D_a \frac{c_1 - c_{\frac{1}{2}}}{\Delta z/2} = uc_{inj},$$

que determina $c_{\frac{1}{2}}$. Evaluando el polinomio que pasa por $(0, c_{\frac{1}{2}})$ y $(\Delta z/2, c_1)$ en el punto $z = -\Delta z/2$, se obtiene

$$c_0 = \frac{(D_a/u - \Delta z/2)c_1 + \Delta z c_{inj}}{D_a/u + \Delta z/2}.$$

De manera similar, si se aplica la condición

$$uc_{\frac{1}{2}} - D_a \frac{c_2 - c_{\frac{1}{2}}}{3\Delta z/2} = uc_{inj}$$

para deducir $c_{\frac{1}{2}}$, la recta que interpola los puntos $(0, c_{\frac{1}{2}})$ y $(3\Delta z/2, c_2)$, en $z = -3\Delta z/2$ vale

$$c_{-1} = \frac{(D_a/u - 3\Delta z/2)c_2 + 3\Delta z c_{inj}}{D_a/u + 3\Delta z/2}.$$

Por último, como la condición de frontera en $z = 1$ es $\partial c/\partial z = 0$, el polinomio del razonamiento anterior es constante, pues se toma $c_{J+1} = c_J$, $c_{J+2} = c_{J-1}$ y $c_{J+3} = c_{J-2}$.

Finalmente, denotando $\mathcal{D}(w(t)) = (\mathcal{D}_1(w(t)); \dots; \mathcal{D}_J(w(t)))$ y $\mathcal{L}(w(t)) = (\mathcal{L}_1(w(t)); \dots; \mathcal{L}_J(w(t)))$, las ecuaciones (7.17) equivalen al sistema matricial

$$w'(t) = \mathcal{D}(w(t)) + \mathcal{L}(w(t)). \quad (7.19)$$

Se considera un operador $C^* : \mathbb{R}^{m \times J} \rightarrow \mathbb{R}^{m \times J}$ que extiende la función C de la forma

$$C^*(w(t)) = C^*(w_1(t); \dots; w_J(t)) = (C(w_1(t)); \dots; C(w_J(t))).$$

Análogamente, se define $W^* : \mathbb{R}^{m \times J} \rightarrow \mathbb{R}^{m \times J}$ como

$$W^*(c(t)) = W^*(c_1(t); \dots; c_J(t)) = (W(c_1(t)); \dots; W(c_J(t))).$$

Con esta notación se deduce

$$\mathcal{D}(w(t)) = C^*(w(t))\mathcal{A}, \quad (7.20)$$

siendo \mathcal{A} la matriz tridiagonal de orden J siguiente:

$$\mathcal{A}_{p,q} = \begin{cases} -\mu, & p = q = 1, J \\ -2\mu, & p = q \neq 1, J \\ \mu, & |p - q| = 1 \end{cases}, \quad \mu = D_a/\Delta z^2.$$

Sea $\Delta t > 0$, el sistema (7.19) puede resolverse mediante el método de Runge-Kutta explícito de segundo orden

$$\begin{aligned} w^{n+1/2} &= w^n + \frac{\Delta t}{2} (\mathcal{L}^n + \mathcal{D}^n) \\ w^{n+1} &= w^n + \Delta t (\mathcal{L}^{n+1/2} + \mathcal{D}^{n+1/2}). \end{aligned} \quad (7.21)$$

En esta expresión, w^k es una aproximación de $w(t_k)$, en el instante $t_k = k\Delta t$. Por otro lado, se toma $\mathcal{D}^k = \mathcal{D}(w^k)$, $\mathcal{L}^k = \mathcal{L}(w^k)$ y la matriz $w^0 = w(0)$ a partir de la condición inicial. El valor de Δt queda determinado al imponer una cierta condición que garantice soluciones libres de oscilaciones.

Según [8], esta condición de estabilidad viene dada por

$$\frac{u\Delta t}{\Delta z} \max_w \rho(C'(w)) + \frac{2D_a\Delta t}{\Delta z^2} \max_w \rho(C'(w)) \leq C_0 \quad (7.22)$$

para una constante $C_0 < 1$ próxima a la unidad. Como se ha comprobado que los valores propios de $C'(w)$ son menores que uno, (7.22) se verifica si

$$\frac{u\Delta t}{\Delta z} + \frac{2D_a\Delta t}{\Delta z^2} \leq C_0 \quad \equiv \quad \Delta t \leq \frac{C_0}{u\Delta z + 2D_a} \Delta z^2.$$

Notamos que se trata de una restricción demasiado estricta sobre Δt , de la forma $\Delta t \leq \tilde{C}_0 \Delta z^2$ cuando $u\Delta z \ll 2D_a$.

Con tal de evitar este hecho, se propone emplear (7.21) introduciendo una expresión implícita para el término parabólico del primer paso:

$$\begin{aligned} w^{n+1/2} &= w^n + \frac{\Delta t}{2} (\mathcal{L}^n + \mathcal{D}^{n+1/2}) \\ w^{n+1} &= w^n + \Delta t (\mathcal{L}^{n+1/2} + \mathcal{D}^{n+1/2}). \end{aligned} \quad (7.23)$$

En este método se combina el tratamiento implícito y explícito de \mathcal{D}^k , y se sigue aplicando el esquema WENO para el cálculo de \mathcal{L}^k . Por esta razón, de ahora en adelante nos referimos a (7.23) como el método IMEX-WENO.

En el segundo paso, conocidos los vectores $w_j^{n+1/2}$ se obtiene cada $c_j^{n+1/2}$ mediante el cálculo de la única raíz positiva de R_w con $w = w_j^{n+1/2}$. A continuación se hallan $\mathcal{L}^{n+1/2}$ y $\mathcal{D}^{n+1/2}$ siguiendo (7.18), y finalmente, los vectores w_j^{n+1} para ejecutar la siguiente iteración.

No obstante, el primer paso no puede aplicarse directamente, ya que no se conoce la expresión explícita de la función $C(w)$. Este problema puede resolverse a partir de su función inversa $W(c)$ conocida, como se especifica a continuación.

A partir de (7.20), este primer paso del método IMEX-WENO puede reescribirse como

$$w^{n+1/2} - \frac{\Delta t}{2} C^*(w^{n+1/2}) \mathcal{A} = \mathcal{G}^n, \quad \mathcal{G}^n = w^n + \frac{\Delta t}{2} \mathcal{L}(w^n).$$

La expresión anterior en términos de $c^{n+1/2} = C^*(w^{n+1/2})$ equivale a

$$W^*(c^{n+1/2}) - \frac{\Delta t}{2} c^{n+1/2} \mathcal{A} = \mathcal{G}^n.$$

De esta forma, debe resolverse el sistema matricial

$$W^*(c) - \frac{\Delta t}{2} c \mathcal{A} = \mathcal{G}^n \tag{7.24}$$

para $c = (c_1, \dots, c_J) \in \mathbb{R}^{m \times J}$, con

$$W^*(c)_{i,j} = W_i(c_j) = (c_j)_i \left(1 + \frac{\eta_i}{1 + b^T c_j} \right) = c_{i,j} + \eta_i c_{i,j} \frac{1}{\varphi_j}, \quad \varphi_j = 1 + b^T c_j, \quad \begin{array}{l} i = 1, \dots, m \\ j = 1, \dots, J \end{array}.$$

Así, denotando $M = \text{diag}(\eta_1, \dots, \eta_m)$ y $K(c) = \text{diag}(1/\varphi_1, \dots, 1/\varphi_J)$, se tiene

$$W^*(c) = c + M c K(c).$$

Por otro lado se define el operador

$$\mathcal{V} : \mathbb{R}^{m \times J} \longrightarrow \mathbb{R}^{mJ}, \quad \mathcal{V}(\mathcal{A})_{m(j-1)+i} = a_{i,j}, \quad \mathcal{A} = (a_{i,j}),$$

en el que, dada una matriz, se obtiene un vector que resulta de yuxtaponer ordenadamente sus columnas. Este operador lineal verifica la relación

$$\mathcal{V}(B X A^T) = (A \otimes B) \mathcal{V}(X), \tag{7.25}$$

siendo $A \otimes B$ la matriz por bloques

$$A \otimes B = \begin{pmatrix} a_{1,1} B & \dots & a_{1,m} B \\ \vdots & \ddots & \vdots \\ a_{n,1} B & \dots & a_{n,m} B \end{pmatrix}, \quad \mathcal{A} = (a_{i,j}) \in \mathbb{R}^{n \times m}.$$

Por linealidad, aplicando \mathcal{V} a la ecuación (7.24) se obtiene

$$\mathcal{V}(W^*(c)) - \frac{\Delta t}{2} \mathcal{V}(c \mathcal{A}) - \mathcal{V}(\mathcal{G}^n) = 0. \tag{7.26}$$

Sea $\mathcal{C} = \mathcal{V}(c)$, por simetría de las matrices $K(c)$ y \mathcal{A} , y la propiedad (7.25),

$$\begin{aligned} \mathcal{V}(W^*(c)) &= \mathcal{V}(c + M c K(c)) = \mathcal{V}(c) + \mathcal{V}(M c K(c)^T) = \mathcal{C} + (K(c) \otimes M) \mathcal{C}, \\ \mathcal{V}(c \mathcal{A}) &= \mathcal{V}(I c \mathcal{A}^T) = (\mathcal{A} \otimes I) \mathcal{C}. \end{aligned}$$

De esta forma, (7.26) coincide con el sistema de ecuaciones

$$\mathcal{F}(\mathcal{C}) = \mathcal{E}(\mathcal{C})\mathcal{C} - B\mathcal{C} - \mathcal{V}(\mathcal{G}^n) = 0, \quad (7.27)$$

con

$$\mathcal{E}(\mathcal{C}) = I + K(c) \otimes M = \text{diag}(E^1, \dots, E^J), \quad E^j = \text{diag} \left(1 + \frac{\eta_i}{\varphi_j} \right)_{i=1, \dots, m}, \quad j = 1, \dots, J,$$

$$B = \frac{\Delta t}{2} (\mathcal{A} \otimes I) = \theta \begin{pmatrix} -I & I & & & \\ I & -2I & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & -2I & I \\ & & & I & -I \end{pmatrix}, \quad \theta = \frac{D_a \Delta t}{2\Delta z^2}.$$

Para resolver el sistema (7.27) mediante el método de Newton:

$$\mathcal{C}^{(k+1)} = \mathcal{C}^{(k)} - \mathcal{F}'(\mathcal{C}^{(k)})^{-1} \mathcal{F}(\mathcal{C}^{(k)}), \quad k = 0, 1, \dots \quad \mathcal{C}^{(0)} = \mathcal{V}(c^n), \quad (7.28)$$

es necesario conocer la matriz jacobiana de la función $\mathcal{F}(\mathcal{C})$. Por su definición,

$$\mathcal{F}'(\mathcal{C}) = (\mathcal{E}(\mathcal{C})\mathcal{C})' - B,$$

pues es suficiente deducir la expresión que toma la matriz jacobiana de $\mathcal{E}(\mathcal{C})\mathcal{C}$. Notamos que

$$\mathcal{E}(\mathcal{C})\mathcal{C} = \begin{pmatrix} E^1 & & & & \\ & \ddots & & & \\ & & E^j & & \\ & & & \ddots & \\ & & & & E^J \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_j \\ \vdots \\ c_J \end{pmatrix} = \begin{pmatrix} E^1 c_1 \\ \vdots \\ E^j c_j \\ \vdots \\ E^J c_J \end{pmatrix},$$

con

$$(E^j c_j)_i = \sum_{k=1}^m E_{i,k}^j (c_j)_k = E_{i,i}^j (c_j)_i = \left(1 + \frac{\eta_i}{1 + b^T c_j} \right) (c_j)_i, \quad i = 1, \dots, m, \quad j = 1, \dots, J.$$

Así,

$$(\mathcal{E}(\mathcal{C})\mathcal{C})' = \text{diag}(\hat{E}^1, \dots, \hat{E}^J)$$

para

$$\hat{E}_{k,l}^j = \frac{\partial (E^j c_j)_k}{\partial (c_j)_l} = \frac{\partial}{\partial (c_j)_l} \left[\left(1 + \frac{\eta_k}{\varphi_j} \right) (c_j)_k \right] = - \left(1 + \frac{\eta_k}{\varphi_j} \right) \delta_{k,l} - \frac{\eta_k b_l}{\varphi_j^2} c_{k,j}, \quad k, l = 1, \dots, m.$$

De esta forma, en cada iteración del método de Newton es necesario resolver un sistema de ecuaciones con matriz de coeficientes tridiagonal por bloques de orden m . El coste adicional que este hecho supone respecto del método explícito (7.21) se compensa al poder tomar Δt mayor que en el caso anterior. Concretamente, la restricción de estabilidad para el método IMEX propuesto viene dada por

$$\frac{u\Delta t}{\Delta z} \max_w \rho(C'(w)) < C_1 \leq 1.$$

Así, para el modelo estudiado, la condición

$$\Delta t < \frac{1}{u} \Delta z$$

debe ser suficiente para evitar oscilaciones en sus soluciones.

7.4. Experimentos numéricos

En esta sección se realizan distintos experimentos numéricos acerca del esquema IMEX-WENO presentado en el apartado anterior para resolver el modelo de cromatografía planteado.

En primer lugar se considera la condición inicial suave

$$w_i(x, 0) = \rho_i \exp(-100(x - 1/2)^2), \quad i = 1, 2, 3 \quad (7.29)$$

siendo $\rho_1 = 1$, $\rho_2 = 2$, $\rho_3 = 3$. Por otro lado, se fijan los parámetros $a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $b_1 = b_2 = b_3 = 1$, con $c_{inj} = 0$, $\varepsilon = 0.5$, $u = 0.2$ y tiempo final $T = 0.1$. En la Figura 7.3 se muestran las estimaciones que proporciona el método para $m = 1000$, $D_a = 0$ y $D_a = 10^{-4}$. Se observa que la simetría de las funciones iniciales se ha distorsionado un poco hacia la derecha, aunque siguen siendo suficientemente suaves.

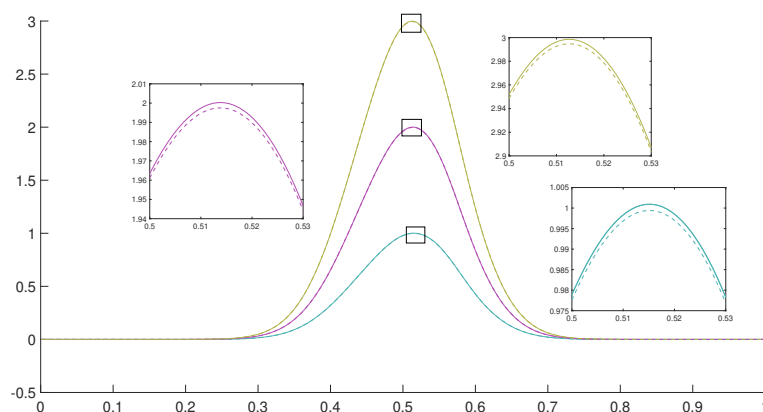


Figura 7.3: Solución del modelo con dato inicial (7.29) para $T = 0.1$ y $m = 1000$. Los resultados para $D_a = 0$ se muestran con línea continua, y para $D_a = 10^{-4}$ con línea discontinua.

Por otro lado se realiza una prueba numérica teniendo en cuenta el concepto de solución débil, un fluido con una afinidad con el medio poroso muy superior a la de los componentes de la mezcla inicial, de modo que puede desplazar a cada uno de ellos a lo largo de la columna. Para una columna suficientemente larga se forman regiones rectangulares, que denotan alta concentración de una sustancia y muy baja de los otros componentes, como se observa en la Figura 7.4.

Los subíndices 1 y 2 hacen referencia a los dos componentes de la mezcla, mientras que el 3 corresponde con la fase móvil o desplazador. Se toman condiciones iniciales nulas y parámetros $a_1 = 4$, $a_2 = 5$, $a_3 = 6$, $b_1 = 4$, $b_2 = 5$ y $b_3 = 1$. Además, se fija $N_t = 10000$, $\varepsilon = 0.5$ y $u = 0.2$, con

$$c_{inj}(t) = \begin{cases} (1 & 1 & 0)^T, & 0 \leq t \leq 0.1, \\ (0 & 0 & 1)^T, & t > 0.1. \end{cases}$$

Por último, se ha considerado $m = 1000$ y la condición CFL $u\Delta t/\Delta z = 0.8$.

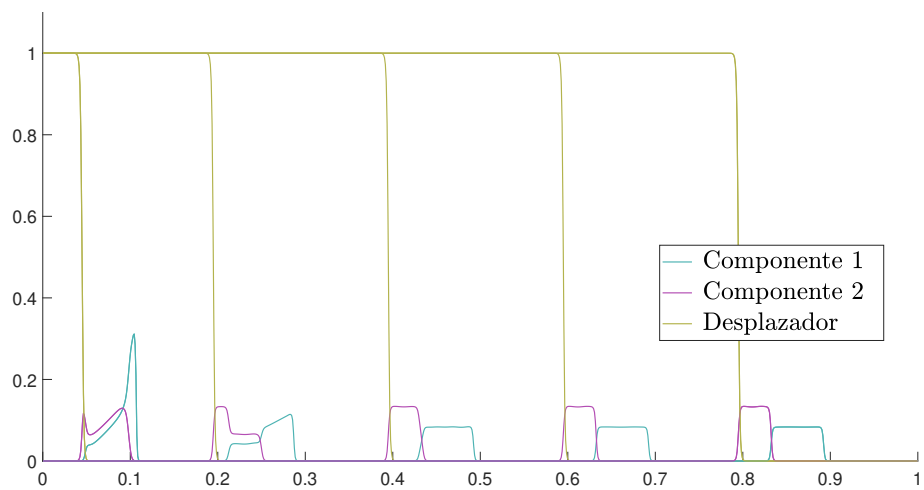


Figura 7.4: Segundo experimento numérico, solución para tiempos $T = 1$, $T = 4$, $T = 8$, $T = 12$ y $T = 16$.

Capítulo 8

Códigos de MatLab

En este capítulo se adjuntan los códigos de MatLab necesarios para generar las tablas y gráficas mostradas en cada uno de los capítulos del trabajo.

Capítulo 2

Figura 2.1

```
hold on
set(gca,'visible','off')
axis ([-2,3,-0.5,3])
set(gca,'XTick',[], 'YTick', [])
plot([-2,3.5],[0,0],'k','LineWidth',1.2)
plot([0,0],[-0.5,5],'k','LineWidth',1.2)
a=linspace(0,5,1000);
p=plot(a,3*a,'b','LineWidth',1.2,'Color',[0.3 0.7 0.7]);
p.Color(4)=0.8;
for i=1:3
    if i==1
        plot(a+0.5*i,3*a,'b','LineWidth',2,'Color',[0.3 0.7 0.7])
    else
        p=plot(a+0.5*i,3*a,'LineWidth',1.2,'Color',[0.3 0.7 0.7]);
        p.Color(4)=0.8;
    end
    p=plot(a-0.5*i,3*a,'LineWidth',1.2,'Color',[0.3 0.7 0.7]);
    p.Color(4)=0.8;
end
scatter(0.5,0.70,'filled','k')
scatter(1,1.5,70,'k','filled')
text(0.15,-0.15,'$x_0=x-at$','interpreter','latex','FontSize',27)
text(1.09,1.6,'$(x,t)$','interpreter','latex','FontSize',27,'color','k');
text(1.5,2.8,'$\mathcal{C}_{x_0}$','interpreter','latex','FontSize',27,'Color','k');
```

Figura 2.2

```
hold on
set(gca,'visible','off')
axis ([-0.5,8,-0.5,3])
set(gca,'DataAspectRatio',[10 6 1])
%axis square
set(gca,'XTick',[], 'YTick', [])
%Ejes coordenados
plot([-0.5,8],[0,0],'k','LineWidth',1.2)
plot([0,0],[-0.5,3],'k','LineWidth',1.2)
text(7.8,-0.1,'$x$','interpreter','latex','FontSize',27)
text(-0.2,2.9,'$t$','interpreter','latex','FontSize',27)
plot([pi/2,pi/2],[-0.05,0],'k')
text(pi/2-0.095,-0.18,'$\frac{\pi}{2}$','interpreter','latex','FontSize',27)
plot([3*pi/2,3*pi/2],[-0.05,0],'k')
text(3*pi/2-0.15,-0.2,'$\frac{3\pi}{2}$','interpreter','latex','FontSize',27)
%Primeras rectas características
a=linspace(0,pi/2,8);
aa=linspace(0,pi/2,13);
b=linspace(0,5,100);
```

```

c=linspace(0,5,100);
for i=1:length(a)
    s1=plot(2*b+a(i),b,'Color',[0.3 0.7 0.7],'LineWidth',1.2);s1.Color(4)=0.7;
    if i<5
        s1=plot(a(i)+3*pi/2*ones(1,100),c,'Color',[0.3 0.7 0.7],'LineWidth',1.2);s1.Color(4)=0.7;
    end
end
for i=1:length(aa)
    if i>1
        s1=plot((sin(aa(i)+pi/2)+1)*b+aa(i)+pi/2,b,'Color',[0.3 0.7 0.7],'LineWidth',1.2);s1.Color(4)=0.7;
    end
    if i<length(aa) && i>1
        s1=plot((sin(aa(i)+pi)+1)*b+aa(i)+pi,b,'Color',[0.3 0.7 0.7],'LineWidth',1.2);s1.Color(4)=0.7;
    end
end
plot([0,7],[1,1],'k--','LineWidth',1.2)
text(7.2,1,'$t_c=1$', 'interpreter','latex','FontSize',27)

```

Figura 2.3

```

hold on
set(gca,'visible','off')
axis([-0.25,3,-0.1,3])
set(gca,'XTick',[ ], 'YTick',[ ])
set(gcf,'Renderer','Painters')
%set(gca,'DataAspectRatio',[8 10 1])
axis square
%Ejes de coordenadas
plot([-0.25,3],[0,0],'k','LineWidth',1.2)
text(2.85,-0.1,'$x$', 'interpreter','latex','FontSize',25,'color','k');
%La circunferencia
x1=linspace(0.25,0.679727,1000);
curve1=1.6094+sqrt(1-(x1-1.25).^2);
curve2=1.6094-sqrt(1-(x1-1.25).^2);
x11=[x1,fliplr(x1)];
inBetween=[curve1,fliplr(curve2)];
p1=fill(x11,inBetween,'b');
set(p1,'EdgeColor','g','FaceAlpha',0.1,'EdgeAlpha',0);
%Otro trozo
x2=linspace(0.679727,2.05651,1000);
curve3=1.6094+sqrt(1-(x2-1.25).^2);
curve4=(x2-1.5).^3+0.5*x2+1;
x22=[x2,fliplr(x2)];inBetween=[curve3,fliplr(curve4)];
p1=fill(x22,inBetween,'b');
set(p1,'EdgeColor','b','FaceAlpha',0.1,'EdgeAlpha',0);
%Otro trozo
curve5=1.6094-sqrt(1-(x2-1.25).^2);
x23=[x2,fliplr(x2)];inBetween=[curve4,fliplr(curve5)];
p1=fill(x23,inBetween,'g');
set(p1,'EdgeColor','b','FaceAlpha',0.1,'EdgeAlpha',0);
%El último trozo
x3=linspace(2.05651,2.25,1000);
curve6=1.6094+sqrt(1-(x3-1.25).^2);
curve7=1.6094-sqrt(1-(x3-1.25).^2);
x33=[x3,fliplr(x3)];inBetween=[curve6,fliplr(curve7)];
p1=fill(x33,inBetween,'g');
set(p1,'EdgeColor','b','FaceAlpha',0.1,'EdgeAlpha',0);
%Los puntos de corte
a=2.05651;scatter(a,(a-1.5)^3+0.5*a+1,70,'k','filled','MarkerFaceAlpha',0.7)
a=0.679727;scatter(a,(a-1.5)^3+0.5*a+1,70,'k','filled','MarkerFaceAlpha',0.7)
%El texto
text(2.4,2.8,'$\mathcal{C}$','interpreter','latex','FontSize',25)
text(2.14,2.2,'$\mathcal{Q}_2$','interpreter','latex','FontSize',25)
text(0.4,0.73,'$\mathcal{Q}_1$','interpreter','latex','FontSize',25)
text(0.5,2.5,'$\mathcal{B}$','interpreter','latex','color','k','FontSize',25)
text(1.34,2.4,'$\mathcal{B}_1$','interpreter','latex','FontSize',25,'color',[0,0,0.65])
text(1.34,0.8,'$\mathcal{B}_2$','interpreter','latex','FontSize',25,'color',[0.01,0.49,0.31])
text(0.85,1.7,'$\mathcal{u}^-$','interpreter','latex','FontSize',25)
text(1.6,1.7,'$\mathcal{u}^+$','interpreter','latex','FontSize',25)
text(1.18,1.46,'$\mathcal{P}$','interpreter','latex','FontSize',25)
%El punto P
s1=plot([0.5,2],[1.6094,1.6094],'k','LineWidth',1.2);s1.Color(4)=0.3;
scatter(1.25,1.6094,70,'k','filled','MarkerFaceAlpha',0.7)
%Curva suave
a=linspace(0.432558,2.5,1000);
s1=plot(a,(a-1.5).^3+0.5*a+1,'k','LineWidth',1.2);s1.Color(4)=0.7;
%Las flechas
text(0.7,1.609,'$\rightarrow$','interpreter','latex','FontSize',40,'color','k')
text(1.35,1.609,'$\leftarrow$','interpreter','latex','FontSize',40,'color','k')
%Las últimos bordes
s1=plot(x1,curve1,'k','LineWidth',1.2);s1.Color(4)=0.7;

```

```

s2=plot(x1,curve2,'k','LineWidth',1.2);s2.Color(4)=0.7;
s1=plot(x2,curve3,'k','LineWidth',1.2);s1.Color(4)=0.7;
s2=plot(x2,curve4,'k','LineWidth',1.2);s2.Color(4)=0.7;
s3=plot(x2,curve5,'k','LineWidth',1.2);s3.Color(4)=0.7;
s4=plot(x3,curve6,'k','LineWidth',1.2);s4.Color(4)=0.7;
s5=plot(x3,curve7,'k','LineWidth',1.2);s5.Color(4)=0.7;

```

Figura 2.4

```

hold on
set(gca,'visible','off')
axis([-2.5,3,-0.25,3])
set(gca,'XTick',[],'YTick',[])
set(gcf,'Renderer','Painters')
%axis square
%Ejes de coordenadas
plot([-3,3],[0,0],'k','LineWidth',1.2)
plot([0,0],[-0.25,3],'k','LineWidth',1.2)
%Zona coloreada
a=linspace(0,5,1000);
curve1=a;
%plot(a,curve1,'b','LineWidth',1.2);
curve2=zeros(1,length(a));
%plot(curve2,a,'b','LineWidth',1.2);
a2=[a,fliplr(curve2)];
inBetween=[curve1,fliplr(a)];
s1=fill(a2,inBetween,'g');
set(s1,'EdgeColor',[0.3 0.7 0.7],'LineWidth',1.8,'FaceAlpha',0.1,'EdgeAlpha',0.9);
%Las rectas características
for i=1:7
    s1=plot(a-0.25*i,a,'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
    s1=plot([0.25*i,0.25*i],[0,3],'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
end
text(-0.13,2.8,'t$','interpreter','latex','FontSize',29,'color','k');
text(2.8,-0.1,'x$','interpreter','latex','FontSize',29,'color','k');
plot([-0.5 0],[1.5*[1 1]','Color',[0.7 0.3 0.7],'LineWidth',1.7)
plot([-0.5 -0.4],[1.5 1.54]','Color',[0.7 0.3 0.7],'LineWidth',1.7)
plot([-0.5 -0.4],[1.5 1.46]','Color',[0.7 0.3 0.7],'LineWidth',1.7)
text(-0.65,1.5,'1$','interpreter','latex','FontSize',29,'color',[0.7 0.3 0.7]);
plot([1.5 2],[1.5*[1 1]','Color',[0.7 0.3 0.7],'LineWidth',1.7)
plot([2 1.9],[1.5 1.54]','Color',[0.7 0.3 0.7],'LineWidth',1.7)
plot([2 1.9],[1.5 1.46]','Color',[0.7 0.3 0.7],'LineWidth',1.7)
text(2.1,1.5,'0$','interpreter','latex','FontSize',29,'color',[0.7 0.3 0.7]);

```

Figura 2.5

```

hold on
set(gca,'visible','off')
axis([-2.5,3,-0.25,3])
set(gca,'XTick',[],'YTick',[])
set(gcf,'Renderer','Painters')
%axis square
%Ejes de coordenadas
plot([-3,3],[0,0],'k','LineWidth',1.2)
plot([0,0],[-0.25,3],'k','LineWidth',1.2)
%Las rectas características
for i=1:8
    if i<8
        s1=plot([-0.25*i,0.25*i],[0,0.5*i]','Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
    end
    s1=plot([0.25*i,0.25*i],[0,0.5*i]','Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
end
%La recta roja
s1=plot([0,3],[0,6]','Color',[0.7 0.3 0.7],'LineWidth',2.5);
text(-0.1,2.85,'t$','interpreter','latex','FontSize',27,'color','k');
text(2.8,-0.1,'x$','interpreter','latex','FontSize',27,'color','k');

```

Figura 2.6

```

hold on
set(gca,'visible','off')
axis([-2,3,-0.25,3])
set(gca,'XTick',[],'YTick',[])
set(gcf,'Renderer','Painters')
%Ejes de coordenadas

```

```

plot([-3,3],[0,0],'k','LineWidth',1.2)
plot([0,0],[-0.25,3],'k','LineWidth',1.2)
text(-0.1,2.8,'$t$', 'interpreter','latex','FontSize',29,'color','k');
text(2.8,-0.1,'$x$', 'interpreter','latex','FontSize',29,'color','k');
%Área coloreada
a=linspace(0,5,1000);
curve1=a;
curve2=zeros(1,length(a));
a2=[a,fliplr(curve2)];
inBetween=[curve1,fliplr(a)];
s1=fill(a2,inBetween,'g');
set(s1,'EdgeColor',[0.3 0.7 0.7],'LineWidth',1.8,'FaceAlpha',0.1,'EdgeAlpha',0.9);
%Las rectas características
for i=1:8
    s1=plot(a+0.25*i,a,'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.6;
    if i<7
        s1=plot([-0.25*i,-0.25*i],[0,3],'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.6;
    end
end
%Las flechas y los números
plot([-0.5 0],1.5*[1 1],'LineWidth',1.7,'Color',[0.7 0.3 0.7])
plot([-0.5 -0.4],[1.5 1.54],'LineWidth',1.7,'Color',[0.7 0.3 0.7])
plot([-0.5 -0.4],[1.5 1.46],'LineWidth',1.7,'Color',[0.7 0.3 0.7])
plot([1.5 2],1.5*[1 1],'LineWidth',1.7,'Color',[0.7 0.3 0.7])
plot([2 1.9],[1.5 1.54],'LineWidth',1.7,'Color',[0.7 0.3 0.7])
plot([2 1.9],[1.5 1.46],'LineWidth',1.7,'Color',[0.7 0.3 0.7])
text(2.05,1.5,'$t$', 'interpreter','latex','FontSize',30,'Color',[0.7 0.3 0.7]);
text(-0.65,1.5,'$x$', 'interpreter','latex','FontSize',30,'Color',[0.7 0.3 0.7]);

```

Figura 2.7

```

hold on
set(gca,'visible','off')
axis ([-1,2,-0.25,3])
set(gca,'XTick',[], 'YTick', [])
set(gcf,'Renderer','Painters')
%Ejes de coordenadas
plot([-3,3],[0,0],'k','LineWidth',1.2)
plot([0,0],[-0.25,3],'k','LineWidth',1.2)
text(-0.07,2.85,'$t$', 'interpreter','latex','FontSize',27,'color','k');
text(1.9,-0.1,'$x$', 'interpreter','latex','FontSize',27,'color','k');
%Las rectas características
a=linspace(0,5,1000);
for i=0:3
    s1=plot(a+0.25*i,a,'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
end
for i=1:6
    s1=plot([0.25*i,3],[0.5*i,3+0.25*i],'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
    s1=plot([0.25*i,0.25*i],[0.5*i,10],'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
end
for i=0:3
    s1=plot([-0.25*i,-0.25*i],[0,3],'Color',[0.3 0.7 0.7],'LineWidth',1.5);s1.Color(4)=0.8;
end
s1=plot(a,2*a,'Color',[0.7 0.3 0.7],'LineWidth',2.5);

```

Capítulo 3

Figura 3.1

```

hold on
set(gca,'visible','off')
axis ([-0.5 5.5 -1 4])
set(gca,'XTick',[], 'YTick', [])
%Línea recta
plot([0 5],[2 2],'k','LineWidth',1.2)
%Los puntos intermedios
plot([0 0],[1.8 2.2],'k','Linewidth',1.2)
plot([5 5],[1.8 2.2],'k','Linewidth',1.2)
plot(0.9*[1 1],[1.9 2.1],'k','Linewidth',1.2)
plot(1.8*[1 1],[1.9 2.1],'k','Linewidth',1.2)
plot(3.2*[1 1],[1.9 2.1],'k','Linewidth',1.2)
plot(4.1*[1 1],[1.9 2.1],'k','Linewidth',1.2)
%Y su texto

```

```

text(-0.3,1.6,'$x_L=x_0$', 'interpreter', 'latex', 'FontSize',25)
text(0.82,1.75,'$x_1$', 'interpreter', 'latex', 'FontSize',25)
text(1.72,1.75,'$x_2$', 'interpreter', 'latex', 'FontSize',25)
text(3.05,1.75,'$x_{J-2}$', 'interpreter', 'latex', 'FontSize',25)
text(3.95,1.75,'$x_{J-1}$', 'interpreter', 'latex', 'FontSize',25)
text(4.7,1.6,'$x_J=x_R$', 'interpreter', 'latex', 'FontSize',25)
%Los puntos suspensivos
scatter(2.3,1.75,10,'filled','k')
scatter(2.5,1.75,10,'filled','k')
scatter(2.7,1.75,10,'filled','k')
%Marcando el espaciado
p=plot(0.9*[1 1],[2.1 2.4],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot([0.9 1.8],[2.4 2.4],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot(1.8*[1 1],[2.1 2.4],'k--','LineWidth',1.2);p.Color(4)=0.3;
text(1.25,2.55,'$\Delta x$', 'interpreter', 'latex', 'FontSize',20)

```

Figura 3.2

```

hold on
set(gca,'visible','off')
axis([0 3 -0.3 3])
set(gca,'XTick',[], 'YTick', [])
%set(gcf,'Renderer','Painters')
%set(gca,'DataAspectRatio',[8 10 1])
%axis square
%Flecha
plot([1.79 1.56],[0.36 0.08],'r','LineWidth',1.4)
plot([1.56 1.6],[0.08 0.15],'r','LineWidth',1.4)
plot([1.56 1.61],[0.08 0.122],'r','LineWidth',1.4)
%Dominio de dependencia del método
p=plot([0.5 2.5],[0 0],'LineWidth',13,'Color',[0.3 0.7 0.7]); p.Color(4)=0.3;
text(0.9,0.2,'$\mathcal{D}_m(x_j,t_n)$', 'interpreter', 'latex', 'FontSize',27,...
'Color',[0.3 0.7 0.7]);
%Dominio de dependencia de la ecuación
scatter(1.5,0,200,'r','filled')
text(1.8,0.4,'$\mathcal{D}_e(x_j,t_n)$', 'interpreter', 'latex', 'FontSize',...,
27,'Color','r','FontWeight','bold');
%Eje de coordenadas
plot([-0.25,3],[0,0],'k','LineWidth',1.2)
text(2.85,-0.1,'$x$', 'interpreter', 'latex', 'FontSize',27,'color','k');
%Punto P
scatter(2.5,2.6,60,'k','filled')
text(2.55,2.6,'$P=\left(x_j,t_n\right)$', 'interpreter', 'latex', 'FontSize',27)
%Recta m
plot([2.5 0.5],[2.6 0],'k--','LineWidth',1.2)
scatter(0.5,0,60,'k','filled')
text(0.36,-0.18,'$(x_{j-n},0)$', 'interpreter', 'latex', 'FontSize',27)
text(1.4,1.5,'$m$', 'interpreter', 'latex', 'FontSize',27)
%Recta r
plot([2.5 1.5],[2.6 0],'k','LineWidth',1.2)
scatter(1.5,0,60,'k','filled')
text(1.2,-0.18,'$Q=(x_j-at_n,0)$', 'interpreter', 'latex', 'FontSize',27)
text(2.13,1.5,'$r$', 'interpreter', 'latex', 'FontSize',27)
%Recta s
plot(2.5*[1 1],[0 2.6],'k--','LineWidth',1.2)
text(2.4,-0.18,'$(x_j,0)$', 'interpreter', 'latex', 'FontSize',27)
scatter(2.5,0,60,'k','filled')

```

Figura 3.3

```

clear u v
%Valor de "a" en la ecuación
a=1; tfin=1;
%Valor de mu
mu=0.8;
%Mallado en las x
xini=0; xfin=1;
J=100; Dx=(xfin-xini)/J;
x=xini:Dx:xfin;
%Valor inicial: u0(x)=sin(2*pi*x);
u=sin(2*pi*x);
%Cálculo de las aproximaciones en los distintos niveles de
t=0;
Dt=min(tfin-t,(Dx*mu)/a);
while t<tfin
    v=zeros(1,J);
    v=(1-mu)*u(2:end)+mu*u(1:end-1);
    v=[v(end) v];

```

```

    u=v;
    clear v
    t=t+Dt;
    Dt=min((Dx*mu)/a,tfin-t);
end
hold on
ax=gca;
ax.FontSize=18;
p1=plot(x,u,'ro-','LineWidth',1.2);
sol=sin(2*pi*(x-tfin));
p2=plot(x,sol,'k','LineWidth',1.2);
leg=legend([p1,p2],{'~Aproximaci\''on $\Delta x=0.01, \nu=0.8^{\nu}$','Soluci\''on exacta'},...
    'interpreter','latex');
leg.FontSize=30;

```

Figura 3.4

```

clear x u v
%Valor de "a" en la ecuación
a=1;
%Valor de mu
mu=0.8;
%Mallado en las x
xini=-1; xfin=2;
J=300; Dx=(xfin-xini)/J;
x=xini:Dx:xfin;
%Mallado en la t
t=0; tfin=1;
Dt=min(tfin-t,(Dx*mu)/a);
%Valor inicial: u0(x)=1 en (-1/3,1/3), 0 en el resto
u=zeros(1,J+1);
for i=1:J+1
    if abs(x(i))<1/3
        u(i)=1;
    else
        u(i)=0;
    end
end
%Cálculo de las aproximaciones en los distintos niveles de t
while t<tfin
    v=zeros(1,J);
    v=(1-mu)*u(2:end)+mu*u(1:end-1);
    v=[v(2) v];
    u=v;
    clear v
    t=t+Dt;
    Dt=min((Dx*mu)/a,tfin-t);
end
hold on
ax=gca;
ax.FontSize=18;
p1=plot(x,u,'ro-','LineWidth',1.2);
%La solución exacta
x=linspace(-1,2,10000);
for i=1:length(x)
    if abs(x(i)-1)<1/3
        sol(i)=1;
    else
        sol(i)=0;
    end
end
p2=plot(x,sol,'k','LineWidth',1.2);
leg=legend([p1,p2],{'~Aproximaci\''on $\Delta x=0.01$ y $\nu=0.8^{\nu}$','Soluci\''on exacta'},...
    'interpreter','latex','Location','best');
leg.FontSize=26;

```

Figura 3.5

```

hold on
axis square
set(gca,'visible','off')
axis ([0 5 1 7])
set(gca,'XTick',[], 'YTick', [])
for h=[4 6]
    p=plot([0 4.8],[h h],'k','LineWidth',1.5);p.Color(4)=0.2;
end
for h=[2 4]
    p=plot([h-1 h-1],[2 7],'k','LineWidth',1.5);p.Color(4)=0.2;
end

```



```

end
text(2.9,1.7,'$x_j$', 'interpreter', 'latex', 'FontSize', 27)
text(0.7,1.7,'$x_{j-1}$', 'interpreter', 'latex', 'FontSize', 27)
text(5,4,'$t_n$', 'interpreter', 'latex', 'FontSize', 27)
text(5,6,'$t_{n+1}$', 'interpreter', 'latex', 'FontSize', 27)
text(3.075,3.7,'$A=(x_j,t_n)$', 'interpreter', 'latex', 'FontSize', 27)
text(0.1,3.7,'$B=(x_{j-1},t_n)$', 'interpreter', 'latex', 'FontSize', 27)
text(2.7,6.2,'$P$', 'interpreter', 'latex', 'FontSize', 27)
p=plot([3.4 1.5],[7 2.25], 'r', 'LineWidth', 2.5);p.Color(4)=0.3;
plot([2.2 3],[4 6], 'r', 'LineWidth', 2.5)
scatter(2.2,4,80, 'r', 'filled')
text(2.24,3.7,'$Q$', 'interpreter', 'latex', 'FontSize', 27, 'Color', 'r')
scatter(3,4,80, 'k', 'filled')
scatter(1,4,80, 'k', 'filled')
scatter(3,6,80, 'k', 'filled')

```

Figura 3.6

```

hold on
axis square
set(gca,'visible','off')
axis ([0 9 1 7])
set(gca,'XTick',[], 'YTick', [])
for h=[4 6]
    p=plot([0 8.5],[h h], 'k', 'LineWidth', 1.5);p.Color(4)=0.2;
end
for h=[2 5 8]
    p=plot([h-1 h-1],[2 7], 'k', 'LineWidth', 1.5);p.Color(4)=0.2;
end
text(3.9,1.7,'$x_j$', 'interpreter', 'latex', 'FontSize', 27)
text(0.7,1.7,'$x_{j-1}$', 'interpreter', 'latex', 'FontSize', 27)
text(6.7,1.7,'$x_{j+1}$', 'interpreter', 'latex', 'FontSize', 27)
text(8.7,4,'$t_n$', 'interpreter', 'latex', 'FontSize', 27)
text(8.7,6,'$t_{n+1}$', 'interpreter', 'latex', 'FontSize', 27)
text(4.1,3.75,'$A$', 'interpreter', 'latex', 'FontSize', 27)
text(1.1,3.75,'$B$', 'interpreter', 'latex', 'FontSize', 27)
text(3.5,6.2,'$P$', 'interpreter', 'latex', 'FontSize', 27)
text(7.1,3.75,'$C$', 'interpreter', 'latex', 'FontSize', 27)
p=plot([4.4688 1.5],[6.75 2], 'r', 'LineWidth', 2.5);p.Color(4)=0.3;
plot([2.75 4],[4 6], 'r', 'LineWidth', 2.5)
scatter(2.75,4,80, 'r', 'filled')
text(2.8,3.75,'$Q$', 'interpreter', 'latex', 'FontSize', 27, 'Color', 'r')
scatter(7,4,80, 'k', 'filled')
scatter(4,4,80, 'k', 'filled')
scatter(1,4,80, 'k', 'filled')
scatter(4,6,80, 'k', 'filled')

```

Figura 3.7

```

clear u v
%Valor de "a" en la ecuación
a=1;
%Valor de mu
mu=0.8;
%Mallado en las x
xini=0; xfin=1;
J=50; Dx=(xfin-xini)/J;
x=xini:Dx:xfin;
%Mallado en la t
Dt=(Dx*mu)/a;
%Valor inicial: u0(x)=sin(2*pi*x);
u=sin(2*pi*x);
%Cálculo de las aproximaciones en los distintos niveles de t
t=0; tfin=1;
Dt=min(tfin-t,(Dx*mu)/a);
while t<tfin
    u=[u(end-1) u u(2)]; %Condiciones de frontera periódicas
    v=zeros(1,J+1);
    v=0.5*mu*(1+mu)*u(1:end-2)+(1-mu^2)*u(2:end-1)+0.5*mu*(mu-1)*u(3:end);
    u=v;
    clear v
    t=t+Dt;
    Dt=min((Dx*mu)/a,tfin-t);
end
hold on
ax=gca;
ax.FontSize=18;
p1=plot(x,u,'ro-', 'LineWidth', 1.2);

```

```

sol=sin(2*pi*(x-tfin));
p2=plot(x,sol,'k','LineWidth',1.2);
leg=legend([p1,p2],{'Aproximaci\''on $\nu=0.8$, $\Delta x=0.02\sim$, '~Soluci\''on exacta'},...
'interpreter','latex');
leg.FontSize=27;
error=100*norm(u-sol,1)/norm(sol,1);

```

Figura 3.8

```

clear sol x u v
%Valor de "a" en la ecuación
a=1;
%Valor de mu
mu=0.8;
%Mallado en las x
xini=-1; xfin=2;
J=300; Dx=(xfin-xini)/J;
x=-1:Dx:2;
%Mallado en la t
t=0; tfin=1;
Dt=min(tfin-t, (Dx*mu)/a);
%Valor inicial
for i=1:J+1
    if abs(x(i))<1/3
        u(i)=1;
    else
        u(i)=0;
    end
end
%Cálculo de las aproximaciones en los distintos niveles de t
while t<tfin
    u=[u(2) u(end-1)]; %Condiciones de frontera periódicas
    v=zeros(1,J+1);
    v=0.5*mu*(1+mu)*u(1:end-2)+(1-mu^2)*u(2:end-1)+0.5*mu*(mu-1)*u(3:end);
    u=v;
    clear v
    t=t+Dt;
    Dt=min((Dx*mu)/a,tfin-t);
end
hold on
ax=gca;
ax.FontSize=18;
p1=plot(x,u,'ro','LineWidth',1.2);
x=linspace(-1,2,10000);
for i=1:length(x)
    if abs(x(i)-1)<1/3
        sol(i)=1;
    else
        sol(i)=0;
    end
end
p2=plot(x,sol,'k','LineWidth',1.2);
leg=legend([p1 p2],{'Aproximaci\''on $\nu=0.8$ y $\Delta x=0.01\sim$, '~Soluci\''on exacta'},...
'interpreter','latex','Location','northwest');
leg.FontSize=26;

```

Tabla 3.1

```

function error = errorUpwindSin(J)
%Valor de "a" en la ecuación
a=1;
%Valor de mu
mu=0.8;
%El vector de x
xini=0; xfin=1;
Dx=(xfin-xini)/J;
x=xini:Dx:xfin;
%Valor inicial: u0(x)=sin(2*pi*x);
u=sin(2*pi*x);
%Cálculo de las aproximaciones en los distintos niveles de t
t=0; tfin=1;
Dt=min(tfin-t, (Dx*mu)/a);
while t<tfin
    v=zeros(1,J);
    v=(1-mu)*u(2:end)+mu*u(1:end-1);
    v=[v(end) v];
    u=v;
    clear v

```

```

    t=t+Dt;
    Dt=min(tfin-t,(Dx*mu)/a);
end
sol=sin(2*pi*(x-tfin));
error=Dx*norm(u-sol,1);
end

clear
J(1)=5;
for i=2:6
    J(i)=2*J(i-1);
end
for i=1:length(J)
    i
    error(i)=errorUpwindSin(J(i));
    if i>1
        orden(i)=log2(error(i-1)/error(i));
    end
end
end

```

Tabla 3.2

```

function error = errorUpwindDos(J)
%Valor de "a" en la ecuación
a=1;
%Valor de mu
mu=0.8;
%Mallado en las x
xini=-1; xfin=2;
Dx=(xfin-xini)/J;
x=xini:Dx:xfin;
%Mallado en la t
t=0; tfin=1;
Dt=min(tfin-t,(Dx*mu)/a);
u=zeros(1,J+1);
for i=1:J+1
    if abs(x(i))<1/3
        u(i)=1;
    else
        u(i)=0;
    end
end
end
%Cálculo de las aproximaciones en los distintos niveles de t
while t<tfin
    v=zeros(1,J);
    v=(1-mu)*u(2:end)+mu*u(1:end-1);
    v=[v(2) v];
    u=v;
    clear v
    t=t+Dt;
    Dt=min((Dx*mu)/a,tfin-t);
end
%La solución exacta
for i=1:J+1
    if abs(x(i)-1)<1/3
        sol(i)=1;
    else
        sol(i)=0;
    end
end
error=Dx*norm(u-sol,1);
end

clear
%Mallado en las x
J(1)=40;
for i=2:7
    J(i)=2*J(i-1);
end
for i=1:length(J)
    error(i)=errorUpwindDos(J(i));
    if i>1
        orden(i)=log2(error(i-1)/error(i));
    end
end
end

```

Tabla 3.3

```

function error = errorLwSin(J)
%Valor de "a" en la ecuación
a=1;
%Valor de mu
mu=0.8;
%El vector de x
xini=0; xfin=1;
Dx=(xfin-xini)/J;
x=xini:Dx:xfin;
%Valor inicial: u0(x)=sin(2*pi*x);
u=sin(2*pi*x);
%Cálculo de las aproximaciones en los distintos niveles de t
t=0; tfin=1;
Dt=min(tfin-t,(Dx*mu)/a);
while t<tfin
    u=[u(end-1) u(2)]; %Condiciones de frontera periódicas
    v=zeros(1,J+1);
    v=0.5*mu*(1+mu)*u(1:end-2)+(1-mu^2)*u(2:end-1)+0.5*mu*(mu-1)*u(3:end);
    u=v;
    clear v
    t=t+Dt;
    Dt=min((Dx*mu)/a,tfin-t);
end
sol=sin(2*pi*(x-tfin));
error=Dx*norm(u-sol,1);
end

clear
%Espaciado en las x
J(1)=5;
for i=2:6
    J(i)=2*J(i-1);
end
for i=1:length(J)
    i
    error(i)=errorLwSin(J(i));
    if i>1
        orden(i)=log2(error(i-1)/error(i));
    end
end
end

```

Capítulo 4

Figura 4.1

```

hold on
set(gca,'visible','off')
axis ([-0.5 8.5 -1 4])
set(gca,'XTick',[], 'YTick', [])
%Línea recta
plot([0 8],[2 2],'k','LineWidth',1.2)
%Los puntos intermedios
plot([0 0],[1.85 2.1],'k','LineWidth',1.2)
plot(1.6*[1 1],[1.85 2.1],'k','LineWidth',1.2)
plot(3.2*[1 1],[1.85 2.1],'k','LineWidth',1.2)
plot(4.8*[1 1],[1.85 2.1],'k','LineWidth',1.2)
plot(6.4*[1 1],[1.85 2.1],'k','LineWidth',1.2)
plot(8*[1 1],[1.85 2.1],'k','LineWidth',1.2)
%Y sus nombres
text(-0.75,1.65,'$x_L=x_{-\frac{1}{2}}$', 'interpreter','latex','FontSize',27);
text(1.5,1.65,'$x_{\frac{1}{2}}$', 'interpreter','latex','FontSize',27);
text(3.1,1.65,'$x_{\frac{3}{2}}$', 'interpreter','latex','FontSize',27);
text(4.5,1.65,'$x_{-\frac{3}{2}}$', 'interpreter','latex','FontSize',27);
text(6.1,1.65,'$x_{\frac{1}{2}}$', 'interpreter','latex','FontSize',27);
text(7.8,1.65,'$x_R=x_{J+\frac{1}{2}}$', 'interpreter','latex','FontSize',27);
%Las líneas discontinuas
p=plot(1.6*[1 1],[1.4 0.8],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot(3.2*[1 1],[1.4 0.8],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot(4.8*[1 1],[1.4 0.8],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot(6.4*[1 1],[1.4 0.8],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot(8*[1 1],[1.4 0.8],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot([0 0],[1.4 0.8],'k--','LineWidth',1.2);p.Color(4)=0.3;
%Lineas para marcar el espaciado
p=plot([0 0],[2.1 2.4],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot([0 0.8],[2.4 2.4],'k--','LineWidth',1.2);p.Color(4)=0.3;
p=plot([0.8 0.8],[2 2.4],'k--','LineWidth',1.2);p.Color(4)=0.3;

```

```

%Y su texto
text(0.1,2.55,'$\Delta x/2$', 'interpreter', 'latex', 'FontSize', 22)
%Las celdas
p=plot([0 1.6],[1.2 1.2], 'LineWidth', 5, 'Color', [0.2 0.5 0.6]); p.Color(4)=0.3;
p=plot([1.6 3.2],[1.2 1.2], 'LineWidth', 5, 'Color', [0.5 0.2 0.4]); p.Color(4)=0.3;
p=plot([4.8 6.4],[1.2 1.2], 'LineWidth', 5, 'Color', [1 0.5 0]); p.Color(4)=0.3;
p=plot([6.4 8],[1.2 1.2], 'LineWidth', 5, 'Color', [0 0.6 0.5]); p.Color(4)=0.3;
%El texto de las celdas
text(0.7,1,'$c_0$', 'interpreter', 'latex', 'Color', [0.2 0.5 0.6], 'FontSize', 27)
text(2.2,1,'$c_1$', 'interpreter', 'latex', 'Color', [0.5 0.2 0.4], 'FontSize', 27)
text(5.4,1,'$c_{J-1}$', 'interpreter', 'latex', 'Color', [1 0.5 0], 'FontSize', 27)
text(7.2,1,'$c_J$', 'interpreter', 'latex', 'Color', [0 0.6 0.5], 'FontSize', 27)
%Los puntos suspensivos
scatter(3.7,1.25,25,'filled','k')
scatter(4,1.25,25,'filled','k')
scatter(4.3,1.25,25,'filled','k')
%Los buenos puntos
scatter(0.8,2,60,'filled','k')
scatter(2.4,2,60,'filled','k')
scatter(5.6,2,60,'filled','k')
scatter(7.2,2,60,'filled','k')
%Y su texto
text(0.7,1.825,'$x_0$', 'interpreter', 'latex', 'FontSize', 27)
text(2.3,1.825,'$x_1$', 'interpreter', 'latex', 'FontSize', 27)
text(5.3,1.825,'$x_{J-1}$', 'interpreter', 'latex', 'FontSize', 27)
text(7.1,1.825,'$x_J$', 'interpreter', 'latex', 'FontSize', 27)

```

Figura 4.2

```

function [x,u]=GodunovEx1(numcel)
J=numcel-1;
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%Valores iniciales
for i=1:length(x)
    if 2*x(i)>Dx
        u(i)=0;
    elseif 2*x(i)<=-Dx
        u(i)=1;
    else
        u(i)=x(i)/Dx+0.5;
    end
end
t=0; tfin=1;
Dt=min(0.5*Dx/max(abs(u)),tfin-t);
%Usaremos condiciones de frontera periódicas, elegidas
%de esta forma por el dato inicial u0.
while t<tfin
    u=[u(1) u u(end)];
    f=max(0.5*max(u(1:end-1),0).^2,0.5*min(u(2:end),0).^2);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
end

clear
%La solución exacta para el plot
xx=linspace(-1,1,10000);
for i=1:length(xx)
    if xx(i)<=0.5
        sol(i)=1;
    else
        sol(i)=0;
    end
end
end
nc=50;
[x,u]=GodunovEx1(nc);
hold on
ax=gca;
ax.FontSize=18;
axis([-1 1 -0.1 1.1])
p1=plot(xx,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x,u,'ro-','LineWidth',1.5);

```

```
leg=legend([p1 p2],{'\Soluci\'on exacta','\Aproximaci\'on con $50$ celdas'},...
    'interpreter','latex','Location','west');
leg.FontSize=26;
```

Figura 4.3

```
function [x,u]=GodunovEx2(numcel)
J=numcel-1;
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%Valores iniciales
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else
        u(i)=2*x(i)/Dx;
    end
end
t=0; tfin=0.5;
Dt=min(0.5*Dx/max(abs(u)),tfin-t);
%Usaremos condiciones de frontera periódicas, elegidas
%de esta forma por el dato inicial u0.
while t<tfin
    u=[u(1) u u(end)];
    f=max(0.5*max(u(1:end-1),0).^2,0.5*min(u(2:end),0).^2);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
end

clear
%La solución exacta para el plot
xx=linspace(-1,1,10000);
for i=1:length(xx)
    if xx(i)<=-0.5
        sol(i)=-1;
    elseif xx(i)>0.5
        sol(i)=1;
    else
        sol(i)=2*xx(i);
    end
end
nc=50;
[x,u]=GodunovEx2(nc);
hold on
ax=gca;
ax.FontSize=18;
axis([-1 1 -1.1 1.1])
p1=plot(xx,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x,u,'ro','LineWidth',1.5);
leg=legend([p1 p2],{'\Soluci\'on exacta','\Aproximaci\'on con $50$ celdas'},...
    'interpreter','latex','Location','best');
leg.FontSize=26;
```

Figura 4.4

```
function [a,f]=fluxRoe(u)
J=length(u)-3;
a=zeros(1,J+2);
f=a;
for i=1:J+2
    if u(i)==u(i+1)
        a(i)=u(i);
    else
        a(i)=0.5*(u(i+1)+u(i));
    end
    if a(i)>0
        f(i)=0.5*u(i)^2;
    end
end
```

```

        else
            f(i)=0.5*u(i+1)^2;
        end
    end
end
end

clear
%La solución exacta
xx=linspace(-1,1,10000);
for i=1:length(xx)
    if xx(i)<=0.5
        sol(i)=1;
    else
        sol(i)=0;
    end
end
nc=50;
%Cálculo de la aproximación (Come on!!!)
J=nc-1;
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%Valores iniciales
for i=1:length(x)
    if 2*x(i)>=Dx
        u(i)=0;
    elseif 2*x(i)<=-Dx
        u(i)=1;
    else
        u(i)=x(i)/Dx+0.5;
    end
end
end
%Mallado temporal
t=0; tfin=1;
while t<tfin
    u=[u(1) u(end)];
    [a,f]=fluxRoe(u);
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
end
hold on
ax=gca;
ax.FontSize=18;
axis([-1 1 -0.05 1.05])
set(gca,'DataAspectRatio',[10 7 10])
p1=plot(xx,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x,u,'ro-','LineWidth',1.5);

clear
%La solución exacta
xx=linspace(-1,1,10000);
for i=1:length(xx)
    if xx(i)<-0.5
        sol(i)=-1;
    elseif xx(i)>0.5;
        sol(i)=1;
    else
        sol(i)=2*xx(i);
    end
end
nc=50;
%Cálculo de la aproximación (Come on!!!)
J=nc-1;
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%Valores iniciales
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else

```

```

        u(i)=2*x(i)/Dx;
    end
end
%Mallado temporal
t=0; tfin=0.5;
while t<tfin
    u=[u(1) u u(end)];
    [a,f]=fluxRoe(u);
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
end
hold on
ax=gca;
ax.FontSize=18;
set(gca,'DataAspectRatio',[7.5 10 10])
axis([-1 1 -1.05 1.05])
p1=plot(xx,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x,u,'ro-','LineWidth',1.5);

```

Figura 4.5

```

function F=fluxRusanov(u)
for i=1:length(u)-1
    a=0.5*(u(i)^2+u(i+1)^2);
    b=max(abs(u(i)),abs(u(i+1)))*(u(i+1)-u(i));
    F(i)=0.5*(a-b);
end
end

clear
%La solución exacta
xx=linspace(-1,1,10000);
sol=zeros(1,10000);
for i=1:length(xx)
    if xx(i)<=0.5
        sol(i)=1;
    else
        sol(i)=0;
    end
end
nc=50;
J=nc-1;
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%Valores iniciales
for i=1:length(x)
    if 2*x(i)>=Dx
        u(i)=0;
    elseif 2*x(i)<=-Dx
        u(i)=1;
    else
        u(i)=x(i)/Dx+0.5;
    end
end
end
t=0; tfin=1;
Dt=min(0.5*Dx/max(abs(u)),tfin);
while t<tfin
    u=[u(1) u u(end)];
    f=fluxRusanov(u);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
hold on
ax=gca;
ax.FontSize=18;
axis([-1 1 -0.05 1.05])
set(gca,'DataAspectRatio',[10 7 10])
p1=plot(xx,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x,u,'ro-','LineWidth',1.5);

```



```

clear
%La solución exacta
xx=linspace(-1,1,10000);
sol=zeros(1,10000);
for i=1:length(xx)
    if xx(i)<-0.5
        sol(i)=-1;
    elseif xx(i)>0.5
        sol(i)=1;
    else
        sol(i)=2*xx(i);
    end
end
nc=50;
J=nc-1;
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%Valores iniciales
u=zeros(1,length(x));
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else
        u(i)=2*x(i)/Dx;
    end
end
t=0; tfin=0.5;
Dt=min(0.5*Dx/max(abs(u)),tfin);
while t<tfin
    u=[u(1) u(end)];
    f=fluxRusanov(u);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
hold on
ax=gca;
ax.FontSize=18;
set(gca,'DataAspectRatio',[7.5 10 10])
axis([-1 1 -1.05 1.05])
p1=plot(xx,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x,u,'ro-','LineWidth',1.5);

```

Figura 4.6

```

J=99;
[x1,u1]=MinmodSing(J);
[x2,u2]=errorRusanov2(J);
y=linspace(-1,1,10000);
sol=SolExacta2(y);
hold on
ax=gca;
ax.FontSize=18;
axis([-1 1 -1.05 1.05])
p1=plot(y,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x1,u1,'LineWidth',1.4,'Color',[0.3 0.7 0.7]);
p3=plot(x2,u2,'LineWidth',1.4,'Color',[0.7 0.3 0.7]);
leg=legend([p1 p2 p3],{'Soluci\''on exacta','M\''etodo orden 2 con $100$ celdas',...
    '\M\''etodo orden 1 con $100$ celdas'},'interpreter','latex','Location','best');
leg.FontSize=26;

```

Tabla 4.1

```

function u = errorRusanov3(J)
%El mallado espacial
xini=0; xfin=5*pi/2;
Dx=(xfin-xini)/(J+1);

```

```

inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%La estimación inicial
u=zeros(1,length(x));
for i=1:length(x)
    if x(i)-Dx/2<=pi/2
        if x(i)+Dx/2<=3*pi/2
            u(i)=2;
        else
            u(i)=1+(pi/2-x(i)+Dx/2-cos(x(i)+Dx/2))/Dx;
        end
    elseif x(i)-Dx/2<=3*pi/2
        if x(i)+Dx/2<=3*pi/2
            u(i)=1-(cos(x(i)+Dx/2)-cos(x(i)-Dx/2))/Dx;
        else
            u(i)=(3*pi/2-x(i)+Dx/2+cos(x(i)-Dx/2))/Dx;
        end
    else
        u(i)=0;
    end
end
end
t=0; tfin=0.1;
Dt=min(0.5*Dx/max(abs(u)),tfin);
while t<tfin
    u=[u(1) u u(end)];
    f=fluxRusanov(u);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
end

```

```

function v = RefOrdenRusanov(u,J)
v=zeros(1,J+1);
R=length(u)/length(v);
for i=1:length(v)
    v(i)=mean(u(R*(i-1)+1:R*i));
end
end

```

```

J=[49 99 199 399 799 1599 3199];
xini=0; xfin=5*pi/2;
Dx=(xfin-xini)/(J+1);
error=zeros(size(J));
Jref=25599;
usol=errorRusanov3(Jref);
for i=1:length(J)
    u=errorRusanov3(J(i));
    usol2=RefOrdenRusanov(usol,J(i));
    error(i)=Dx(i)*norm(u-usol2,1);
end
orden=zeros(1,length(error)-1);
for i=1:length(orden)
    orden(i)=log2(error(i)/error(i+1));
end
end

```

Tabla 4.2

```

function error = errorRusanov2(J)
%El mallado espacial
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%La solución exacta
sol=zeros(1,length(x));
for i=1:length(x)
    if x(i)<=0.5
        sol(i)=-1;
    elseif x(i)>0.5
        sol(i)=1;
    else
        sol(i)=2*x(i);
    end
end
end

```

```

%La aproximación del método de Rusanov
%Valores iniciales
u=zeros(1,length(x));
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else
        u(i)=2*x(i)/Dx;
    end
end
t=0; tfin=0.5;
Dt=min(0.5*Dx/max(abs(u)),tfin);
while t<tfin
    u=[u(1) u u(end)];
    f=fluxRusanov(u);
    v=u(2:end-1)-Dt/Dx*(f(2:end)-f(1:end-1));
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
error=Dx*norm(u-sol,1);
end

clear
%Espaciado en las x
J(1)=49;
for i=2:7
    J(i)=2*(J(i-1)+1)-1;
end
for i=1:length(J)
    i
    error(i)=errorRusanov2(J(i));
    if i>1
        orden(i)=log2(error(i-1)/error(i));
    end
end
end

```

Tabla 4.3

```

function u=errorMinmodSuave(J)
%El mallado espacial
xini=0; xfin=5*pi/2;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%La estimación inicial
u=zeros(1,length(x));
for i=1:length(x)
    if x(i)-Dx/2<=pi/2
        if x(i)+Dx/2<=pi/2
            u(i)=2;
        else
            u(i)=1+(pi/2-x(i)+Dx/2-cos(x(i)+Dx/2))/Dx;
        end
    elseif x(i)-Dx/2<=3*pi/2
        if x(i)+Dx/2<=3*pi/2
            u(i)=1-(cos(x(i)+Dx/2)-cos(x(i)-Dx/2))/Dx;
        else
            u(i)=(3*pi/2-x(i)+Dx/2+cos(x(i)-Dx/2))/Dx;
        end
    else
        u(i)=0;
    end
end
%Mallado temporal
t=0; tfin=0.1;
Dt=min(0.5*Dx/max(abs(u)),tfin-t);
while t<tfin
    u=[u(2) u(1) u u(end) u(end-1)];
    v=evol(u,Dx,Dt);
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
end

```

```

J=[49 99 199 399 799 1599 3199];
xini=0; xfin=5*pi/2;
Dx=(xfin-xini)/(J+1);
error=zeros(size(J));
orden=zeros(1,length(J)-1);
Jref=25599;
usol=errorMinmodSuave(Jref);
for i=1:length(J)
    i
    u=errorMinmodSuave(J(i));
    usol2=RefOrdenRusanov(usol,J(i));
    error(i)=Dx(i)*norm(u-usol2,1);
    if i>1
        orden(i)=log2(error(i-1)/error(i));
    end
end
end

```

Tabla 4.4

```

function sol=SolExacta2(x)
tfin=0.5;
sol=zeros(1,length(x));
for i=1:length(x)
    if x(i)<tfin
        sol(i)=-1;
    elseif x(i)>tfin
        sol(i)=1;
    else
        sol(i)=x(i)/tfin;
    end
end
end

function [x,u]=MinmodSing(J)
%El mallado espacial
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
inix=xini+Dx/2;
finx=xfin-Dx/2;
x=inix:Dx:finx;
%Valores iniciales
u=zeros(1,length(x));
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else
        u(i)=2*x(i)/Dx;
    end
end
end
%Mallado temporal
t=0; tfin=0.5;
Dt=min(0.5*Dx/max(abs(u)),tfin-t);
while t<tfin
    u=[u(2) u(1) u(end) u(end-1)];
    v=evol(u,Dx,Dt);
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
end

function error = errorMinmodSing(J)
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
[x,u]=MinmodSing(J);
sol=SolExacta2(x);
error=Dx*norm(u-sol,1);
end

J=[49 99 199 399 799 1599 3199];
error=zeros(1,length(J));
orden=zeros(1,length(J)-1);
for i=1:length(J)

```

```

    error(i)=errorMinmodSing(J(i));
    if i>1
        orden(i)=log2(error(i-1)/error(i));
    end
end
end

```

Capítulo 5

Figuras 5.1 y 5.2.

```

J=99;
[x1,u1]=MinmodSing(J);
[x1,u1]=WENOSing(J);
[x2,u2]=ENOSing(J);
y=linspace(-1,1,10000);
sol=SolExacta2(y);
hold on
ax=gca;
ax.FontSize=18;
axis([-1 1 -1.05 1.05])
p1=plot(y,sol,'k','LineWidth',1.2);
%Ploteamos la aproximación obtenida
p2=plot(x1,u1,'o-','LineWidth',1.4,'Color',[0.3 0.7 0.7]);
pause(40)
p3=plot(x2,u2,'o-','LineWidth',1.4,'Color',[0.7 0.3 0.7]);
leg=legend([p1 p2 p3],{'~Soluci\''on exacta','~M\''etodo M3 con $100$ celdas',...
    'M\''etodo M2 con $100$ celdas'},'interpreter','latex','Location','best');
leg.FontSize=26;

```

```

function f=fluxRusanovPar(a,b)
s=max(abs(a),abs(b));
f=(a^2+b^2)/4-s/2*(b-a);
end

```

```

function sig=CalculoSigma(u,Dx)
J=length(u)-5;
sig=zeros(1,J+3);
for i=1:J+3
    a=(u(i+2)-u(i+1))/Dx;
    b=(u(i+1)-u(i))/Dx;
    sig(i)=minmod(a,b);
end
end

```

```

function L=vectorL(u,Dx)
sig=CalculoSigma(u,Dx);
J=length(u)-5;
umenos=zeros(1,J+2);
umas=umenos;
f=umas;
for j=1:J+2
    umenos(j)=u(j+1)+Dx*sig(j)/2;
    umas(j)=u(j+2)-Dx*sig(j+1)/2;
    f(j)=fluxRusanovPar(umenos(j),umas(j));
end
L=-(f(2:end)-f(1:end-1))/Dx;
end

```

```

function v=evol(u,Dx,Dt)
L=vectorL(u,Dx);
u1=u(3:end-2)+Dt*L;
u1=[u1(2) u1(1) u1 u1(end) u1(end-1)];
L1=vectorL(u1,Dx);
u2=u1(3:end-2)+Dt*L1;
v=0.5*(u(3:end-2)+u2);
end

```

```

function [x,u]=MinmodSing(J)
%El mallado espacial
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1);
x=xini+Dx/2:Dx:xfin-Dx/2;

```

```

%Valores iniciales
u=zeros(1,length(x));
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else
        u(i)=2*x(i)/Dx;
    end
end
%Mallado temporal
t=0; tfin=0.5;
Dt=min(0.5*Dx/max(abs(u)),tfin-t);
while t<tfin
    u=[u(2) u(1) u u(end) u(end-1)];
    v=evol(u,Dx,Dt);
    u=v;
    clear v
    t=t+Dt;
    Dt=min(0.5*Dx/max(abs(u)),tfin-t);
end
end

function v=PasoRK2(u,Dx,Dt)
%Esta función calcula un paso del método de Runge-Kutta 3 estudiado
%Entradas: u, vector de J+1 elementos
%          L, vector de J+1 elementos
%          Dt, espaciado temporal
%Salida: v, vector de J+1 elementos, correspondiente a la estimación del
%siguiente instante en el mallado, t + Dt.
L=Calculo_L(u,Dx);
u1=u+Dt*L;
L1=Calculo_L(u1,Dx);
u2=u1+Dt*L1;
v=0.5*(u+u2);
end

function L=Calculo_L(u,Dx)
%Calcula el vector L a partir de u
%Entrada: el vector u, de J+1 elementos
%          el mallado espacial, de J+1 elementos
%Salida: el vector L, de J+1 elementos
J=length(u)-1;
u=[u(1)*ones(1,3) u u(end)*ones(1,3)]; %Añadimos los "ghost values" necesarios
[vizq,vder]=ENO(u); %Aproximaciones en los extremos de las celdas
F=CalculoFlujos(vizq,vder); %Flujos F
L=zeros(1,J+1);
for j=1:J+1
    L(j)=-(F(j+1)-F(j))/Dx;
end
end

function f=CalculoFlujos(vizq,vder)
%Cálculo de los flujos F a partir de las aproximaciones en los extremos de las celdas
%Entradas: aproximaciones en los extremos de las celdas, vder y vizq,
%          vectores de J+3 elementos
%Salida: un vector F, de J+2 elementos.
J=length(vder)-3;
f=zeros(1,J+2);
for j=1:J+2
    f(j)=fluxRusanovPar(vder(j),vizq(j+1));
end
end

function [vizq,vder]=ENO(u)
%Esta función calcula la reconstrucción ENO con r=3 en los extremos de las celdas
%Entradas: u vector de promedios, de J+7 elementos
%Salidas: vizq vector de aproximaciones por izquierda, de J+3 eltos
%          vder vector de aproximaciones por derecha, de J+3 eltos
%Definimos la matriz de coeficientes C
C=[11/6 -7/6 1/3; 1/3 5/6 -1/6; -1/6 5/6 1/3; 1/3 -7/6 11/6];
%Elección de "k"
J=length(u)-7;
vizq=zeros(1,J+3); vder=vizq;
for j=-1:J+1
    if abs(u(j+4)-u(j+3))<=abs(u(j+5)-u(j+4))
        if abs(u(j+4)-2*u(j+3)+u(j+2))<=abs(u(j+5)-2*u(j+4)+u(j+3))
            k=2;
        end
    end
end

```

```

        else
            k=1;
        end
    else
        if abs(u(j+6)-2*u(j+5)+u(j+4))<=abs(u(j+5)-2*u(j+4)+u(j+3))
            k=0;
        else
            k=1;
        end
    end
    vder(j+2)=dot(C(k+2,:),u(j-k+4:j-k+6));
    vizq(j+2)=dot(C(k+1,:),u(j-k+4:j-k+6));
end
end

function [x,u] = ENOSing(J)
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1); %espaciado espacial
x=xini+Dx/2:Dx:xfin-Dx/2; %mallado en las x, para la estimación inicial
%Valores iniciales
u=zeros(1,length(x));
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else
        u(i)=2*x(i)/Dx;
    end
end
t=0; tfin=0.5;
Dt=min((0.5*Dx)/max(abs(u)),tfin-t); %espaciado temporal
while t<tfin
    v=PasoRK2(u,Dx,Dt);
    u=v;
    Dt=min((0.5*Dx)/max(abs(u)),tfin-t);
    t=t+Dt;
end
end

function [alpha1,alpha2]=WENOAlphas(v,d,eps)
%Esta función calcula los pesos alpha a partir de u y los pesos óptimos d
%Entradas: u, vector de 5 elementos
%          d, vector de pesos óptimos, de tres elementos
%Salidas: alpha, vector de 3 elementos
IS=zeros(1,3);
IS(1)=13/12*(v(3)-2*v(4)+v(5))^2+1/4*(3*v(3)-4*v(4)+v(5))^2;
IS(2)=13/12*(v(2)-2*v(3)+v(4))^2+1/4*(v(2)-v(4))^2;
IS(3)=13/12*(v(1)-2*v(2)+v(3))^2+1/4*(v(1)-4*v(2)+3*v(3))^2;
alpha1=d./(eps+IS).^2;
alpha2=flip(d)./(eps+IS).^2;
end

function [vizq,vder]=WENO(u,eps)
%Esta función calcula las aproximaciones en los extremos de las celdas,
%usando la reconstrucción WENO de parámetro "r"
%Entradas: u, vector de J+7 elementos, incluyendo ya los "ghost values"
%Salidas: vizq, aproximaciones en el extremo izquierdo, vector de J+3 eltos
%          vder, aproximaciones en el extremo derecho, vector de J+3 eltos
C=[11/6 -7/6 1/3; 1/3 5/6 -1/6; -1/6 5/6 1/3; 1/3 -7/6 11/6]; %Matriz C de coeficientes
d=[3/10 3/5 1/10]; %Pesos óptimos
J=length(u)-7;
vizq=zeros(1,J+3); vder=vizq;
for j=1:J+3
    [alpha1,alpha2]=WENOAlphas(u(j:j+4),d,eps);
    w1=alpha1/sum(alpha1);
    w2=alpha2/sum(alpha2);
    term=zeros(2,3);
    for k=0:2
        term(1,k+1)=dot(C(k+2,:),u(j-k+2:j-k+4));
        term(2,k+1)=dot(C(k+1,:),u(j-k+2:j-k+4));
    end
    vder(j)=dot(w1,term(1,:));
    vizq(j)=dot(w2,term(2,:));
end
end

function L=WENOCalculo_L(u,Dx)

```

```

%Calcula el vector L a partir de u
%Entrada: el vector u, de J+1 elementos
%          el mallado espacial, de J+1 elementos
%Salida: el vector L, de J+1 elementos
eps=1e-30;
u=[u(3) u(2) u(1) u u(end) u(end-1) u(end-2)]; %Condiciones de frontera artificiales
[vizq,vder]=WENO(u,eps); %Aproximaciones en los extremos de las celdas
f=CalculoFlujos(vizq,vder); %Flujos f
L=-(f(2:end)-f(1:end-1))/Dx;
end

```

```

function v=WENOPasoRK3(u,Dx,Dt)
%Esta función ejecuta un paso del método RK3
%Entradas: u, vector de J+1 elementos
%          x, mallado espacial, de J+1 elementos
%          Dt, espaciado temporal
%Salidas: v, vector de J+1 eltos, correspondiente a la aproximación en el
%instante t+Dt.
L=WENOCalculo_L(u,Dx);
u1=u+Dt*L;
L1=WENOCalculo_L(u1,Dx);
u2=3/4*u+1/4*u1+1/4*Dt*L1;
L2=WENOCalculo_L(u2,Dx);
v=1/3*u+2/3*u2+2/3*Dt*L2;
end

```

```

function [x,u] = WENOSing(J)
xini=-1; xfin=1;
Dx=(xfin-xini)/(J+1); %espaciado espacial
x=xini+Dx/2:Dx:xfin-Dx/2; %mallado en las x, para la estimación inicial
%Valores iniciales
u=zeros(1,length(x));
for i=1:length(x)
    if 2*x(i)<=-Dx
        u(i)=-1;
    elseif 2*x(i)>=Dx
        u(i)=1;
    else
        u(i)=2*x(i)/Dx;
    end
end
t=0; tfin=0.5;
Dt=min((0.5*Dx)/max(abs(u)),tfin-t); %espaciado temporal
while t<tfin
    v=WENOPasoRK3(u,Dx,Dt);
    u=v;
    Dt=min((0.5*Dx)/max(abs(u)),tfin-t);
    t=t+Dt;
end
end

```

Capítulo 6

Figura 6.1

```

hold on
set(gca,'visible','off')
axis ([-3,3,-0.1,3])
set(gca,'XTick',[], 'YTick', [])
set(gcf,'Renderer','Painters')
set(gca,'DataAspectRatio',[8 10 1])
%Ejes de coordenadas
plot([-3,3],[0,0],'LineWidth',1.4,'Color',[0.3 0.7 0.7]);
text(2.85,-0.1,'$x$', 'interpreter','latex','FontSize',25,'Color',[0.3 0.7 0.7]);
plot([0 0],[-0.1 3],'Color',[0.3 0.7 0.7],'LineWidth',1.4);
text(0.05,2.9,'$t$', 'interpreter','latex','FontSize',25,'Color',[0.3 0.7 0.7]);
%Líneas de discontinuidad
plot([0 -2.725],[0 1.25],'k','LineWidth',1.2)
plot([0 -2],[0 2.25],'k','LineWidth',1.2);
plot([0 -0.25],[0 2.99],'k','LineWidth',1.2)
plot([0 1.05],[0 2.8],'k','LineWidth',1.2)
plot([0 2.47],[0 1.7],'k','LineWidth',1.2);
plot([0 2.86],[0 0.9],'k','LineWidth',1.2)
%Valores que toma la función

```



```

text(-2,0.3,'$u^{-u^{(0)}}$', 'interpreter', 'latex', 'FontSize', 25)
text(-1.5,0.95,'$u^{(1)}$', 'interpreter', 'latex', 'FontSize', 25)
text(0.2,2.1,'$u^{(i)}$', 'interpreter', 'latex', 'FontSize', 25)
text(1.5,0.8,'$u^{(m-1)}$', 'interpreter', 'latex', 'FontSize', 25)
text(1.6,0.25,'$u^{(m)}=u^{+}$', 'interpreter', 'latex', 'FontSize', 25)
%Puntos suspensivos
scatter(1.5,1.5,15,'k','filled');
scatter(1.6,1.4,15,'k','filled')
scatter(1.4,1.6,15,'k','filled')
scatter(-1.1,1.8,15,'k','filled')
scatter(-1.2,1.7,15,'k','filled')
scatter(-1,1.9,15,'k','filled')

```

Capítulo 7

Figura 7.1

```

hold on
set(gca,'visible','off')
axis([-0.5,8.5,-3,3])
set(gca,'XTick',[], 'YTick', [])
set(gcf,'Renderer','Painters')
set(gca,'DataAspectRatio',[8 10 1])
%Ejes de coordenadas
plot([-0.5,8.5],[0,0],'LineWidth',1.4,'Color',[0.3 0.7 0.7]);
text(8.3,-0.25,'$\lambda$', 'interpreter', 'latex', 'FontSize', 28, 'Color', [0.3 0.7 0.7]);
plot([0 0],[ -3 3], 'Color', [0.3 0.7 0.7], 'LineWidth', 1.4);
text(-0.7,2.7,'$\$c(\lambda)$', 'interpreter', 'latex', 'FontSize', 28, 'Color', [0.3 0.7 0.7]);
%Polos de la función racional
p=plot(2.5*[1 1],[ -3 3], 'k--', 'LineWidth', 1.2); p.Color(4)=0.6;
p=plot(4*[1 1],[ -3 3], 'k--', 'LineWidth', 1.2); p.Color(4)=0.6;
p=plot(6*[1 1],[ -3 3], 'k--', 'LineWidth', 1.2); p.Color(4)=0.6;
p=plot(7.5*[1 1],[ -3 3], 'k--', 'LineWidth', 1.2); p.Color(4)=0.6;
%Marcas de los ejes
plot(1.2*[1 1],0.065*[-1 1], 'k', 'LineWidth', 1.2)
plot(0.05*[-1 1],1.5*[1 1], 'k', 'LineWidth', 1.2)
%Texto
text(-0.7,1.465,'$\frac{1}{\varphi(c)}$', 'interpreter', 'latex', 'FontSize', 38)
text(1.15,-0.25,'$1$', 'interpreter', 'latex', 'FontSize', 25)
text(2.57,-0.22,'$d_1$', 'interpreter', 'latex', 'FontSize', 25)
text(4.07,-0.22,'$d_2$', 'interpreter', 'latex', 'FontSize', 25)
text(6.07,-0.22,'$d_{m-1}$', 'interpreter', 'latex', 'FontSize', 25)
text(7.57,-0.22,'$d_m$', 'interpreter', 'latex', 'FontSize', 25)
%Lineas de la imagen de 1
p=plot(1.2*[1 1],[0 1.5], 'k', 'LineWidth', 1.2); p.Color(4)=0.2;
p=plot([0 1.2],1.5*[1 1], 'k', 'LineWidth', 1.2); p.Color(4)=0.2;
%Gráficas
%La primera, desde 1 a la d1
x=linspace(1.2,2.5,1000);y=(x.^2-5.085)/((x+0.6).*(x-2.55));
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La segunda parte
x=linspace(2.5,4,1000);y=(x-3.3)/((x-3.25).^2-0.72);
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La tercera, primer tramo
x=linspace(4,4.2,1000);y=(x-5.1)/((x-4.85).^2-1);
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
x=linspace(4.2,4.4,1000);y=(x-5.1)/((x-4.85).^2-1);
plot(x,y,'--', 'LineWidth', 1.1, 'Color', [0.2 0.6 0.6])
%La tercera, segundo tramo
x=linspace(5.6,5.8,1000);y=(x-5.29)/((x-5.23).^2-0.7);
plot(x,y,'--', 'LineWidth', 1.1, 'Color', [0.2 0.6 0.6])
x=linspace(5.8,6,1000);y=(x-5.29)/((x-5.23).^2-0.7);
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La cuarta parte
x=linspace(6,7.5,1000);y=(x-6.79)/((x-6.73).^2-0.7);
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%Puntos suspensivos
scatter(4.7,0.35,15,'filled','k')
scatter(5,0.35,15,'filled','k')
scatter(5.3,0.35,15,'filled','k')
%La imagen de 1
scatter(1.2,1.5,25,'filled','k')
%Los lambdas
text(1.95,0.2,'$\lambda_1$', 'interpreter', 'latex', 'FontSize', 25, 'Color', [0.8 0.2 0.2])
scatter(2.255,0,25,'filled','k')
text(3.3,0.2,'$\lambda_2$', 'interpreter', 'latex', 'FontSize', 25, 'Color', [0.8 0.2 0.2])
scatter(3.3,0,25,'filled','k')

```

```
text(6.79,0.2,'$\lambda_m$', 'interpreter', 'latex', 'FontSize', 25, 'Color', [0.8 0.2 0.2])
scatter(6.79,0,25,'filled','k')
```

Figura 7.2

```
hold on
set(gca,'visible','off')
axis([-8,4,-3,3])
set(gca,'XTick',[], 'YTick', [])
set(gcf,'Renderer','Painters')
set(gca,'DataAspectRatio',[8 10 1])
%Ejes de coordenadas
plot([-8,4],[0,0],'LineWidth',1.4,'Color',[0.3 0.7 0.7]);
text(3.8,-0.25,'$y$', 'interpreter', 'latex', 'FontSize', 28, 'Color', [0.3 0.7 0.7]);
plot([0 0],[-3 3],'Color',[0.3 0.7 0.7],'LineWidth',1.4);
text(0.1,2.7,'$R_w(y)$', 'interpreter', 'latex', 'FontSize', 28, 'Color', [0.3 0.7 0.7]);
%Polos de la función racional
p=plot(-5.5*[1 1],[-3 -0.5],'k--','LineWidth',1.2); p.Color(4)=0.6;
p=plot(-5.5*[1 1],[0 3],'k--','LineWidth',1.2); p.Color(4)=0.6;
p=plot(-4.2*[1 1],[-3 -0.5],'k--','LineWidth',1.2); p.Color(4)=0.6;
p=plot(-4.2*[1 1],[0 3],'k--','LineWidth',1.2); p.Color(4)=0.6;
p=plot(-2.5*[1 1],[-3 -0.5],'k--','LineWidth',1.2); p.Color(4)=0.6;
p=plot(-2.5*[1 1],[0 3],'k--','LineWidth',1.2); p.Color(4)=0.6;
p=plot(-1.2*[1 1],[-3 -0.5],'k--','LineWidth',1.2); p.Color(4)=0.6;
p=plot(-1.2*[1 1],[0 3],'k--','LineWidth',1.2); p.Color(4)=0.6;
%Texto
text(0.1,1.66,'$1$', 'interpreter', 'latex', 'FontSize', 25)
text(-5.8,-0.2,'$-\eta_m$', 'interpreter', 'latex', 'FontSize', 25)
text(-4.7,-0.2,'$-\eta_{m-1}$', 'interpreter', 'latex', 'FontSize', 25)
text(-2.8,-0.2,'$-\eta_2$', 'interpreter', 'latex', 'FontSize', 25)
text(-1.5,-0.2,'$-\eta_1$', 'interpreter', 'latex', 'FontSize', 25)
%Gráficas
%La segunda parte
x=linspace(-5.5,-4.2,1000);y=-((x+5.7).^2-1)/((x+5.65).*(x+4));
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La tercera, primer tramo
x=linspace(-4.2,-4,1000);y=-((x+4.4).^2-1)/((x+4.35).*(x+2.7));
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
x=linspace(-4,-3.85,1000);y=-((x+4.4).^2-1)/((x+4.35).*(x+2.7));
plot(x,y,'--','LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La tercera, segundo tramo
x=linspace(-2.65,-2.5,1000);y=-((x+4).^2-1)/((x+3.95).*(x+2.3));
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
x=linspace(-2.75,-2.65,1000);y=-((x+4).^2-1)/((x+3.95).*(x+2.3));
plot(x,y,'--','LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La cuarta parte
x=linspace(-2.5,-1.2,1000);y=-((x+2.7).^2-1)/((x+2.65).*(x+1));
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La primera parte
x=linspace(-8,-5.5,1000);y=-(x+4).^2/(x+5.2)-3.7;
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%La última parte
x=linspace(-1.2,3,1000);y=-(x+2.52).^2/(x+1.32)+6.3;
plot(x,y,'LineWidth',1.1,'Color',[0.2 0.6 0.6])
%El punto (0,1)
scatter(0,1,25,'filled','k')
%Puntos suspensivos
scatter(-3.1,0.35,15,'filled','k')
scatter(-3.35,0.35,15,'filled','k')
scatter(-3.6,0.35,15,'filled','k')
%Los lambdas
text(-5.1,0.3,'$\rho_m$', 'interpreter', 'latex', 'FontSize', 25, 'Color', [0.8 0.2 0.2])
scatter(-4.7,0,25,'filled','k')
text(-2.02,0.3,'$\rho_2$', 'interpreter', 'latex', 'FontSize', 25, 'Color', [0.8 0.2 0.2])
scatter(-1.7,0,25,'filled','k')
text(-0.84,0.3,'$\rho_1$', 'interpreter', 'latex', 'FontSize', 25, 'Color', [0.8 0.2 0.2])
scatter(-0.907,0,25,'filled','k')
text(2.12,0.3,'$\rho_0$', 'interpreter', 'latex', 'FontSize', 25, 'Color', [0.8 0.2 0.2])
scatter(2.16,0,25,'filled','k')
```

Figura 7.3

```
m=1000; tfin=0.1;
Da=0;
[x1,w1]=IMEX_WENO_Exp2(m,tfin,Da);
hold on
ax=gca;
ax.FontSize=18;
```

```

x1=x1(1,:);
plot(x1,w1(1,:), 'LineWidth',1.2, 'Color', [0.3 0.7 0.7])
plot(x1,w1(2,:), 'LineWidth',1.2, 'Color', [0.7 0.3 0.7])
plot(x1,w1(3,:), 'LineWidth',1.2, 'Color', [0.7 0.7 0.3])
pause(150)
Da=1e-4;
[x1,w2]=IMEX_WENO_Exp2(m,tfin, Da);
x1=x1(1,:);
plot(x1,w2(1,:), '--', 'LineWidth',1.2, 'Color', [0.3 0.7 0.7])
plot(x1,w2(2,:), '--', 'LineWidth',1.2, 'Color', [0.7 0.3 0.7])
plot(x1,w2(3,:), '--', 'LineWidth',1.2, 'Color', [0.7 0.7 0.3])

function [x,w0] = IMEX_WENO_Exp2(m,tfin, Da)
%En este script se aplica el método IMEX-WENO explicado en el artículo
%Definición de parámetros
N=3; tol=1e-10; maxiter=100; ep=0.5;
a=(4:6)'; nu=(1-ep)*a/ep; b=ones(N,1);
u=0.2; cinj=zeros(N,1);
ro=(1:3)';
Dz=1/m;
ref=0.9; Dt=Dz*ref/u; t=0;
%Consideramos el mallado espacial
x=Dz/2:Dz:1-Dz/2;
%Condición inicial
w0=ro*exp(-100*(x-0.5).^2);
c0=Funcion_C(w0,b,nu,tol,maxiter);
while t<tfin
    Dt=min(Dt,tfin-t);
    cinj=cinj; %Aquí evaluaríamos cinj en t en caso de que cinj no fuese 0
    w=PasoCompleto(c0,w0,b,nu,tol,maxiter,cinj, Da,u,Dz,Dt);
    w0=w;
    c0=Funcion_C(w0,b,nu,tol,maxiter);
    t=t+Dt;
end
end

function c=Funcion_C(w,b,nu,tol,maxiter)
%Esta función calcula el vector "c" a partir de "w"
%Entradas: matriz w de N filas y m columnas
%           vector b, de parámetros, en columna, de N elementos
%           vector nu, de parámetros, en columna, de N elementos
%           tol: la tolerancia fijada para calcular la raíz con el método de
%           Newton
%           maxiter: el número máximo de iteraciones a la hora de calcular
%           la raíz con el método de Newton
%Salidas: vector c, en columna, de N elementos
a=size(w); N=a(1); m=a(2);
c=zeros(N,m);
for j=1:m
    r=Calculo_raiz(w(:,j),b,nu,tol,maxiter);
    for i=1:N
        c(i,j)=w(i,j)/(1+nu(i)/r);
    end
end
end

function y0=Calculo_raiz(w,b,nu,tol,maxiter)
%Esta función calcula una aproximación de la raíz de la función racional
%R_w, que luego usaremos para hallar el vector "c" a partir de "w"
%Entradas: el vector w, en columna, de N elementos
%           el vector b, de parámetros, en columna, de N elementos
%           el vector nu, de parámetros, en columna, de N elementos
%           tol, la tolerancia fijada al aplicar el método de Newton
%           maxiter, el número máximo de iteraciones para el método de
%           Newton
%Salidas: r, la aproximación de la raíz
y0=1+dot(b,w)/2; %Fijamos la estimación inicial
er1=1; er2=0; %Fijamos estos errores para entrar al bucle por primera vez
iter=0;
[f,df]=Funcion_Racional(w,b,nu,y0);
while er1+er2>tol && iter<maxiter
    y1=y0-f/df;
    er1=abs(y1-y0);
    y0=y1;
    [f,df]=Funcion_Racional(w,b,nu,y0);
    er2=abs(f);
    iter=iter+1;
end
end

```

```

function [fy,dfy]=Funcion_Racional(w,b,nu,y)
    fy=1-y*y*dot(b.*w,1./(y+nu));
    dfy=-1+dot(b.*w,1./(y+nu))-y*dot(b.*w,1./((y+nu).^2));
end

function w=PasoCompleto(c,w0,b,nu,tol,maxiter,cinj,Da,u,Dz,Dt)
%Esta función ejecuta un paso del método RK2 propuesto
%Entradas: w y c, matrices de N filas y m columnas, del paso n
%Salidas: w, matriz de N filas y m columnas, correspondiente al paso n+1
%-----
[N,m]=size(w0);
%Obtenemos c a tiempo t_{n+1/2}
C=SolucionFC(c(:),w0,b,nu,tol,maxiter,cinj,Da,u,Dz,Dt);
c=reshape(C,N,m);
%Pasamos c ---> w, a tiempo t_{n+1/2}
w=Funcion_W(c,b,nu);
%Obtenemos L^{n+1/2}
L=Matriz_L(w,b,nu,tol,maxiter,cinj,Da,u,Dz);
%Obtenemos D^{n+1/2}
D=Matriz_D(c,Da,Dz);
%Obtenemos w a tiempo t_{n+1}
w=w0+Dt*(L+D);
end

function CO=SolucionFC(CO,w,b,nu,tol,maxiter,cinj,Da,u,Dz,Dt)
%Esta función resuelve el sistema de ecuaciones F(C)=0 mediante el método
%de Newton
%Entradas: CO, la iteración inicial
%          tol, la tolerancia fijada
%          maxiter, el número máximo de iteraciones permitidas a la hora de
%          resolver el sistema mediante el método de Newton
%          w, la estimación a tiempo t_n
%Salidas: CO, la estimación de la solución obtenida
er1=1; er2=0; %Valores fijados para entrar en el bucle en la primera iteración
iter=0;
[FC,dFC]=Funcion_Paso1(CO,w,b,nu,tol,maxiter,cinj,Da,u,Dz,Dt);
while er1+er2>=tol && iter<maxiter
    C1=CO-dFC\FC;
    er1=norm(C1-CO);
    CO=C1;
    [FC,dFC]=Funcion_Paso1(CO,w,b,nu,tol,maxiter,cinj,Da,u,Dz,Dt);
    er2=norm(FC);
    iter=iter+1;
end
end

function w=Funcion_W(c,b,nu)
%Esta función calcula "w" a partir de "c".
%Entradas: matriz c, de N filas y m columnas
%          vector b, de parámetros, como columna, de N elementos
%          vector nu, de parámetros, como columna, de N elementos
%Salida: vector w, como una columna, de N elementos
a=size(c); N=a(1); m=a(2);
w=zeros(N,m); %Definimos una matriz de igual tamaño que c
for j=1:m
    for i=1:N
        w(i,j)=c(i,j)*(1+nu(i)/(1+dot(b,c(:,j))));
    end
end
end

function L=Matriz_L(w,b,nu,tol,maxiter,cinj,Da,u,Dz)
%Esta función calcula la matriz L, de N filas y m columnas, a partir de la
%matriz w, del mismo tamaño.
%Entradas: w, matriz de N filas y m columnas
%          cinj, un vector columna de N elementos, correspondiente a la
%          evaluación de la función cinj en el instante de tiempo
%          correspondiente.
%          Dz, espaciado en el dominio espacial
%          u, un parámetro escalar
%Salidas: la matriz L, de N filas y m columnas
a=size(w); m=a(2);
f=Vector_f(w,b,nu,tol,maxiter,cinj,Da,u,Dz);
L=zeros(a);
L(:,1)=(u*cinj-f(:,1))/Dz;
for j=2:m
    L(:,j)=(f(:,j-1)-f(:,j))/Dz;
end
end

```

```

function D=Matriz_D(c, Da, Dz)
%Esta función calcula la matriz D a partir de la matriz c, formada por N
%filas y m columnas
%Entradas: c, matriz de N filas y m columnas
%Salidas: D, matriz de la misma dimensión
a=size(c); m=a(2);
mu=Da/(Dz^2);
D=mu*c*Matriz_A(m);
end

function [FC,dFC]=Funcion_Paso1(C,w,b,nu,tol,maxiter,cinj, Da,u,Dz,Dt)
%Esta función calcula F(C) y F'(C) a partir de un vector columna de Nm
%elementos, que denotamos por C, y las estimaciones de "w" a tiempo t_n,
%una matriz de N filas y m columnas
%Entradas: el vector C, en columna, de Nxm elementos
%           la matriz w, de N filas y m columnas
%Salidas: el vector FC, de Nm elementos
%           la matriz dFC, la matriz jacobiana de F evaluada en C
%Definimos los parámetros N y m
[N,m]=size(w);
G=Matriz_G(w,b,nu,tol,maxiter,cinj, Da,u,Dz,Dt);
E=Matriz_E(C,b,nu,N,m);
theta=(Da*Dt)/(2*Dz^2);
M=E-theta*kron(Matriz_A(m),speye(N));
FC=M*C-G(:);
%Calculamos la matriz jacobiana evaluada en el vector C
c=reshape(C,N,m);
EE=cell(1,m);
for j=1:m
    for k=1:N
        for l=1:N
            EE{j}(k,l)=(1+nu(k)/(1+dot(b,c(:,j))))*(k=1)-nu(k)*b(l)*c(k,j)/(1+dot(b,c(:,j)))^2;
        end
    end
end
H=blkdiag(EE{1:m});
dFC=H-theta*kron(Matriz_A(m),speye(N));
end

function G=Matriz_G(w,b,nu,tol,maxiter,cinj, Da,u,Dz,Dt)
%Esta función calcula la matriz G a partir de la matriz w, de N filas y m
%columnas
%Entradas: w, una matriz de N filas y m columnas
%           cinj, un vector columna de N elementos, correspondiente a la
%           evaluación de la función cinj en el instante de tiempo
%           correspondiente.
%           u, parámetro escalar
%           Dz, espaciado en el dominio espacial
%           Dt, espaciado en el dominio temporal
%Salidas: la matriz G, también de N filas y m columnas
L=Matriz_L(w,b,nu,tol,maxiter,cinj, Da,u,Dz);
G=w+Dt/2*L;
end

function E=Matriz_E(C,b,nu,N,m)
%Esta función calcula la matriz E a partir del vector C (columna, Nm
%elementos) y los parámetros b y nu
%Entradas: vector C, en columna, de Nm elementos
%           vector b, de parámetros, en columna, de N elementos
%           vector nu, de parámetros, en columna, de N elementos
%           N, el número de componentes en la mezcla
%           m, el número de elementos de la partición del dominio espacial
%Salidas: la matriz E
%Reconstruimos la matriz c a partir del vector C
c=reshape(C,N,m);
d=zeros(N*m,1);
for j=1:m
    for i=1:N
        d((j-1)*N+i)=1+nu(i)/(1+dot(b,c(:,j)));
    end
end
E=spdiags(d,0,N*m,N*m);
end

function f=Vector_f(w,b,nu,tol,maxiter,cinj, Da,u,Dz)
%Esta función calcula los flujos f a partir de la matriz w
%Entradas: w, matriz de N filas y m columnas

```

```

%      el vector b, de parámetros, como columna, de N elementos
%      el vector nu, de parámetros, como columna, de N elementos
%      tol, la tolerancia fijada para el método de Newton
%      maxiter, el número máximo de iteraciones fijado para el método
%      de Newton
%      cinj, un vector columna de N elementos, correspondiente al calor
%      de la función c_inj en el instante de tiempo correspondiente
%      Da, una constante escalar
%      u, una constante escalar
%      Dz, el incremento en el eje espacial
%Salidas: f, matriz de N filas y m columnas
%Obtenemos los "ghost values" a partir de la matriz w
a=size(w); N=a(1); m=a(2);
c=Funcion_C(w,b,nu,tol,maxiter);
c=GhostValues_C(c,cinj,Da,u,Dz);
w=Funcion_W(c,b,nu);
fmas=zeros(N,m); fmenos=fmas;
for j=1:m
    for i=1:N
        fmas(i,j)=Rec_Weno(u/2*(c(i,j:j+4)+w(i,j:j+4)),'+');
        fmenos(i,j)=Rec_Weno(u/2*(c(i,j+1:j+5)-w(i,j+1:j+5)),'-');
    end
end
f=fmas+fmenos;
end

```

```

function wref2 = RefOrden(wref1,m)
a=size(wref1);
mref=a(2);
R=mref/m;
wref2=zeros(3,m);
for i=1:m
    wref2(:,i)=mean(wref1(:,R*(i-1)+1:R*i)');
end
end

```

```

function A=Matriz_A(m)
%En esta función, generamos la matriz A, dada su dimensión, m
%Entradas: m, la dimensión de la matriz A
%Salidas: la matriz A
A=spdiags(ones(m,1)*[1 -2 1],[-1:1,m,m]);
A(1,1)=A(1,1)+1;
A(end,end)=A(end,end)+1;
end

```

```

function c=GhostValues_C(c,cinj,Da,u,Dz)
%Esta función calcula los "ghost values" a partir de la matriz c, de N
%filas y m columnas
%Entradas: la matriz c, de N filas y m columnas
%      cinj, un vector columna de N elementos, correspondiente al calor
%      de la función c_inj en el instante de tiempo correspondiente
%      Da, una constante escalar
%      u, una constante escalar
%      Dz, el incremento en el eje espacial
%Salidas: la matriz c, ahora con N filas y (m+5) columnas
%Los tres ghost values a la derecha
a=size(c); m=a(2);
c(:,m+1)=c(:,m);
c(:,m+2)=c(:,m-1);
c(:,m+3)=c(:,m-2);
%Los dos ghost values a la izquierda
col=((Da/u-Dz/2)*c(:,1)+Dz*cinj)/(Da/u+Dz/2);
c=[col c];
col=((Da/u-3*Dz/2)*c(:,3)+3*Dz*cinj)/(Da/u+3*Dz/2);
c=[col c];
end

```

```

function res=Rec_Weno(h,flag)
%Calcula la aproximación de orden 5 a partir de un vector de 5 elementos
%Entradas: h, vector de 5 elementos
%      flag, bandera que nos indica si vamos a calcular fmas o fmenos
%Salidas: res, aproximación de orden alto
%Definimos la matriz de coeficientes C
C=[11/6 -7/6 1/3; 1/3 5/6 -1/6; -1/6 5/6 1/3; 1/3 -7/6 11/6];
%Para fmas
f=zeros(1,3);
for r=0:2
    if flag=='+'
        f(r+1)=dot(C(r+2,:),h(3-r:5-r));
    end
end

```

```

elseif flag=='-'
    f(r+1)=dot(C(r+1,:),h(3-r:5-r));
end
end
%El beta no cambia siga '+' o '-'
beta=zeros(1,3);
beta(1)=13/12*(h(3)-2*h(4)+h(5))^2+1/4*(3*h(3)-4*h(4)+h(5))^2;
beta(2)=13/12*(h(2)-2*h(3)+h(4))^2+1/4*(h(2)-h(4))^2;
beta(3)=13/12*(h(1)-2*h(2)+h(3))^2+1/4*(h(1)-4*h(2)+3*h(3))^2;
d=[3/10 3/5 1/10];
if flag=='+'
    alpha=d./((beta+eps).^2);
elseif flag=='-'
    alpha=flip(d)./((beta+eps).^2);
end
w=alpha./sum(alpha);
res=dot(w,f);
end

```

Figura 7.4

```

clear
%En este script se aplica el método IMEX-WENO explicado en el artículo
%Definición de parámetros
N=3; tol=1e-10; maxiter=100; ep=0.5;
a=(4:6)'; nu=(1-ep)*a/ep; b=[4 5 1]';
u=0.2; Nt=1e4; L=1; Da=(L*u)/(2*Nt);
m=1000; Dz=1/m;
ref=0.8; Dt=Dz*ref/u; t=0; tfin=16;
%Consideramos el mallado espacial
x=Dz/2:Dz:1-Dz/2;
%Condición inicial
w0=zeros(N,m);
c0=Funcion_C(w0,b,nu,tol,maxiter);
hold on
ax=gca;
ax.FontSize=18;
axis ([0 1 0 1.1])
while t<tfin
    Dt=min(Dt,tfin-t);
    cinj=(t>=0 && t<0.1)*[1 1 0]'+(t>=0.1)*[0 0 1]';
    w=PasoCompleto(c0,w0,b,nu,tol,maxiter,cinj,Da,u,Dz,Dt);
    w0=w;
    c0=Funcion_C(w0,b,nu,tol,maxiter);
    t=t+Dt;
if round(1e6*t)*1e-6==1
    p1=plot(x,c0(1,:), 'LineWidth',1.2, 'Color', [0.3 0.7 0.7]);
    p2=plot(x,c0(2,:), 'LineWidth',1.2, 'Color', [0.7 0.3 0.7]);
    p3=plot(x,c0(3,:), 'LineWidth',1.2, 'Color', [0.7 0.7 0.3]);
    leg=legend([p1 p2 p3],{'Componente 1''','Componente 2''','Desplazador'''},...
        'interpreter','latex');
    leg.FontSize=26;
elseif mod(round(1e6*t)*1e-6,4)==0
    plot(x,c0(1,:), 'LineWidth',1.2, 'Color', [0.3 0.7 0.7]);
    plot(x,c0(2,:), 'LineWidth',1.2, 'Color', [0.7 0.3 0.7]);
    plot(x,c0(3,:), 'LineWidth',1.2, 'Color', [0.7 0.7 0.3]);
end
end
t
end

```


Capítulo 9

Conclusiones

El objetivo central de este trabajo ha sido presentar y analizar métodos numéricos de alto orden para la resolución de sistemas hiperbólicos de leyes de conservación, en particular de aquellos presentes en los modelos de cromatografía.

La cromatografía es un proceso de separación de los componentes de una mezcla. Esta técnica consiste en disolver la mezcla en un fluido y hacerla circular a través de una columna cilíndrica que contiene un medio poroso. La diferencia en el grado de retención de los componentes en el medio provoca que éstos se separen y puedan recuperarse de manera independiente en la base del cilindro.

En [3] se razona que este proceso químico puede modelarse mediante un sistema de ecuaciones diferenciales no lineales, un sistema hiperbólico cuando un cierto parámetro se anula. Además, en [8, 7] se argumenta que los métodos para sistemas hiperbólicos pueden modificarse ligeramente para tratar el modelo de cromatografía general.

Con la motivación de poder seguir el artículo [8] y resolver el sistema que en él se plantea, en primer lugar se estudian las leyes de conservación desde el punto de vista teórico. Así, se trata el concepto de solución débil, el cual acepta funciones con alguna singularidad, así como la condición de Rankine-Hugoniot o la condición de entropía, con el objetivo de hallar la solución del sistema físicamente correcta.

Posteriormente se repasan los métodos de volúmenes finitos clásicos, de orden uno, y se presentan algunas estrategias que permiten aumentar el orden de convergencia a dos, tanto en la variable espacial como temporal.

Más cerca del objetivo, la segunda parte del trabajo se centra en los esquemas numéricos de orden alto basados en reconstrucciones polinómicas esencialmente no oscilatorias de tipo WENO y resolvedores de sistemas de ecuaciones diferenciales provenientes de una formulación semidiscreta. De esta forma, se logran estimaciones de orden tres para soluciones suaves, y se consigue estimar de forma más precisa las discontinuidades presentes en algunas soluciones.

Una vez extendidos los contenidos presentados para la ecuación escalar al caso de sistemas, el capítulo final se dedica a reproducir los resultados de [8]. Se han introducido los conceptos básicos de cromatografía para deducir las ecuaciones del modelo, y se ha demostrado que éste se reduce a un sistema hiperbólico de leyes de conservación. Para ello, se ha probado que una cierta función admite una correspondencia inversa, cuyo valor puede obtenerse calculando numéricamente la raíz de una función racional adecuada. Finalmente, se ha introducido un método implícito-explicito que incluye la reconstrucción WENO y un esquema de Runge-Kutta de orden dos, y se han realizado distintos experimentos numéricos acerca de éste.

Bibliografía

- [1] Alberto Bressan. Lecture Notes on Hyperbolic Conservation Laws. https://pdfs.semanticscholar.org/b869/2dedc7fa41f860a26b05f4715d212c86b979.pdf?_ga=2.144307504.469352117.1598885756-326900697.1574454656.
- [2] Antonio Baeza Manzanares. *Adaptive mesh refinement techniques for high order shock capturing schemes for hyperbolic systems of conservation laws*. PhD thesis, Universitat de València, 2010.
- [3] Georges Guiochon, Attila Felinger, Dean G. Shirazi, and Anita M. Katii. *Fundamentals of Preparative and Nonlinear Chromatography*. Elsevier, 2006.
- [4] Helge Holden, and Nils Henrik Risebro. *Front Tracking for Hyperbolic Conservation Laws*. Springer, 2002.
- [5] K. W. Morton, and David Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge, 2005.
- [6] Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, 1998.
- [7] R. Donat, F. Guerrero, and P. Mulet. IMEX WENO Schemes for Two-phase Flow Vertical Equilibrium Processes in a Homogeneous Porous Medium. *Applied Mathematics & Information Sciences*, 7(5):1865–1878, 2013.
- [8] R. Donat, F. Guerrero, and P. Mulet. Implicit–Explicit WENO scheme for the equilibrium dispersive model of chromatography. *Applied Numerical Mathematics*, 123:22–42, 2018.
- [9] Randall J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.
- [10] Randall J. LeVeque. *Finite-Volume Methods for Hyperbolic Problems*. Cambridge, 2004.
- [11] Siddhartha Mishra, Ulrik Skre Fjordholm, and Rémi Abrall. Numerical methods for conservation laws and related equations. https://www2.math.ethz.ch/education/bachelor/lectures/fs2013/math/nhdgl/numcl_notes_HOMEPAGE.pdf.