



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Métodos numéricos para la ecuación del transporte en una dimensión

José Ramón Gutiérrez Rodríguez

2019/2020

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Métodos numéricos para la ecuación del transporte en una dimensión

José Ramón Gutiérrez Rodríguez

Julio, 2020

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Coñecemento: Matemática Aplicada
Título: Métodos numéricos para la ecuación del transporte en una dimensión.
Breve descripción do contido
Se trata de introducir, analizar e implementar en el ordenador, métodos numéricos para la resolución de la ecuación en derivadas parciales del transporte, en una dimensión espacial. Se estudiarán aspectos como el orden, la estabilidad, la conservación de ciertas magnitudes físicas, etc.
Recomendacións
Outras observacións

Índice general

Resumen	VIII
Introducción	XI
1. Ecuación del transporte.	1
1.1. Análisis general.	1
1.1.1. Ecuaciones con velocidad constante.	2
1.1.2. Ecuaciones con velocidad variable.	5
2. Aproximación numérica.	9
2.1. Convergencia.	9
2.2. Consistencia.	13
2.3. Estabilidad.	13
2.3.1. Análisis de Fourier.	17
2.3.2. Análisis de Von-Neumann.	19
2.3.3. Estabilidad en esquemas generales.	24
2.4. Orden.	25
3. Esquemas de aproximación.	29
3.1. FTFS	29
3.2. FTBS	32
3.3. FTFS-FTBS	35
3.4. Lax-Wendroff	36
3.5. BTFS	38
3.6. BTBS	39
3.7. BTCS	39
3.8. Método de características.	41

4. Anexo.	45
4.1. Programación esquema BTCS MATLAB.	45
4.2. Programación esquema FTFS MATLAB.	57
Bibliografía	61

Resumen

Supongamos que queremos diseñar la red de gaseoductos de nuestra ciudad, con el objetivo de conocer, entre otras cosas, qué cantidad de gas debemos introducir en la red y por dónde, para conseguir que lleguen las cantidades correctas de gas a cada vivienda de la ciudad. Pues bien, este y otros muchos problemas se modelan matemáticamente mediante la ecuación del transporte. Dicha ecuación, que en breve presentaremos, en muy pocos casos se puede resolver mediante cálculos elementales, por lo que debemos recurrir a la implementación de métodos numéricos que nos faciliten el trabajo. Así, primero consideraremos casos sencillos donde el problema se resuelve de forma analítica y, a continuación, analizaremos varios métodos numéricos y sus diversas características, como son convergencia, consistencia, estabilidad, etc., con el objetivo de comprobar si se adaptan a nuestras necesidades, y los compararemos entre sí para decidir con cual quedarnos en función de lo que necesitemos.

Abstract

Let us suppose that we want to design the gas pipeline network of our city, with the aim of knowing, among other things, how much gas we should introduce into the network and where, in order to get the right amounts of gas to each house in the city. Well, this and many other problems are mathematically modelled using the transport equation. This equation, which we will shortly present, can only be solved in very few cases by means of elementary calculations, so we must resort to the implementation of numerical methods that facilitate our work. Thus, first we will consider simple cases where the problem is solved in an analytical way, and then we will analyze several numerical methods and their diverse characteristics, such as convergence, consistency, stability, etc., with the objective of checking if they adapt to our needs, and we will compare them with each other to decide which one to use depending on what we need.

Introducción

El objetivo será analizar una serie de esquemas para la ecuación del transporte, ya que aparece como modelo matemático en numerosas ocasiones. Para resolver esta ecuación numéricamente se utilizan esquemas en diferencias finitas, donde ambas derivadas (espacial y temporal) se aproximan por diferencias finitas. Dependiendo de cómo se hacen estas aproximaciones, se obtendrán diferentes esquemas de los cuales se tratará de estudiar propiedades como convergencia, estabilidad, orden, etc. La estructura del documento es la siguiente:

1. **Ecuación del transporte.** En este capítulo se hará una revisión general de algunos aspectos de la ecuación, y se mostrará que, para ciertos casos, es posible resolver la ecuación mediante cálculos elementales.
2. **Aproximación numérica.** En este capítulo se procederá con la discretización y las herramientas necesarias para el análisis de la convergencia, la consistencia, la estabilidad y el orden de convergencia de los esquemas.
3. **Esquemas de aproximación.** Este capítulo se dedica a introducir una colección de esquemas de aproximación, para los que se analizarán sus características principales, así como algunas comprobaciones del orden de convergencia de forma experimental. Para ello, se ha llevado a cabo una implementación en ordenador utilizando MATLAB
4. **Anexo.** En él, a modo de ejemplo, se encontrará el código de MATLAB de dos de los esquemas de aproximación del apartado anterior, que nos permitirán resolver la ecuación del transporte en el ordenador. No obstante, cabe señalar que para realizar el trabajo se han implementado varios esquemas más que no se recogen en el anexo para no hacer un documento excesivamente largo.

Capítulo 1

Ecuación del transporte.

1.1. Análisis general.

Consideramos el problema del transporte unidimensional en un dominio acotado

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) + c(x, t) \frac{\partial u}{\partial x}(x, t) = f(x, t), & \forall (x, t) \in [a, b] \times [T_0, T_f], \\ u(x, T_0) = u_0(x), & \forall x \in [a, b] \text{ (condición inicial)}. \end{cases} \quad (1.1)$$

siendo $c(x, t)$ la velocidad de propagación, con condición de contorno

$$\begin{cases} u(a, t) = \alpha(t), & \forall t \in [T_0, T_f], \text{ si } c(a, t) > 0, \\ u(b, t) = \beta(t), & \forall t \in [T_0, T_f], \text{ si } c(b, t) < 0. \end{cases} \quad (1.2)$$

Cuando el dominio espacial es \mathbb{R} , no hay condiciones de contorno y el modelo representa, si c es constante y $f \equiv 0$, el transporte puro de la señal u_0 con velocidad c . El caso más sencillo se da cuando $c \equiv 1$ y $f \equiv 0$, es decir, cuando tenemos la ecuación

$$\frac{\partial u}{\partial t}(x, t) + \frac{\partial u}{\partial x}(x, t) = 0. \quad (1.3)$$

Se tiene que $u = u(x, t)$ es solución de (1.3) si y solo si es constante a lo largo de las *líneas características*

$$x - t = cte. \quad (1.4)$$

De este modo, deducimos que las soluciones de (1.3) son de la forma

$$u(x, t) = u_0(x - t) \quad (1.5)$$

siendo $u_0(x)$ la condición inicial del problema.

La solución (1.5) es entonces una onda simple de transporte en la que la gráfica de u_0 se transporta (avanza) en el eje real a velocidad uno.

1.1.1. Ecuaciones con velocidad constante.

Si c es una **constante arbitraria no nula**, tenemos la ecuación en derivadas parciales (EDP),

$$\frac{\partial u}{\partial t}(x, t) + c \frac{\partial u}{\partial x}(x, t) = 0, \quad c \in \mathbb{R} \setminus \{0\} \quad (1.6)$$

y en este caso, la solución es

$$u(x, t) = u_0(x - ct) \quad (1.7)$$

que de nuevo es una onda simple de transporte en la que la gráfica de u_0 avanza en el eje real a velocidad c . Es claro que (1.7) es una solución de (1.6), veremos más adelante que es la única.

De la fórmula (1.7) podemos deducir que la solución en cualquier instante t_0 es una copia de la función original, pero desplazada hacia la derecha, si c es positivo, o hacia la izquierda, si c es negativo, una distancia $|c|t_0$. Otra forma de expresar esto es que la solución en (x, t) depende exclusivamente del valor de $\xi = x - ct$. Las líneas en el plano (x, t) en las cuales $x - ct$ es constante, son las llamadas *líneas características*. Para analizar de manera más profunda el concepto de líneas características, vamos a considerar una ecuación hiperbólica más general:

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) + c \frac{\partial u}{\partial x}(x, t) + bu = f(x, t) \\ u(x, T_0) = u_0(x) \end{cases} \quad (1.8)$$

donde c y b son constantes. Basándonos en las observaciones anteriores, vamos a cambiar las variables (x, t) por (ξ, τ) , donde τ y ξ se definen como

$$\tau = t, \quad \xi = x - ct. \quad (1.9)$$

El cambio de variable inverso es entonces

$$t = \tau, \quad x = \xi + c\tau \quad (1.10)$$

y definimos $\tilde{u}(\xi, \tau) = u(x, t)$, donde (ξ, τ) y (x, t) están relacionadas de la forma que acabamos de definir. Ambas funciones \tilde{u} y u representan la misma función, pero la tilde es necesaria para distinguir entre los dos sistemas de coordenadas para las variables independientes. La ecuación (1.8) pasa a ser,

$$\frac{d\tilde{u}}{d\tau} = \frac{\partial \tilde{u}}{\partial \tau} = \frac{\partial t}{\partial \tau} u_t + \frac{\partial x}{\partial \tau} u_x = u_t + cu_x = -bu + f(\xi + c\tau, \tau). \quad (1.11)$$

Entonces, tenemos:

$$\frac{d\tilde{u}}{d\tau} = -b\tilde{u} + f(\xi + c\tau, \tau). \quad (1.12)$$

Esta es una ecuación diferencial ordinaria en τ y su solución es

$$\tilde{u}(\xi, \tau) = u_0(\xi)e^{-b\tau} + \int_0^\tau f(\xi + c\sigma, \sigma)e^{-b(\tau-\sigma)} d\sigma. \quad (1.13)$$

Volviendo a las variables originales, obtenemos la solución de la ecuación (1.8):

$$u(x, t) = u_0(x - ct)e^{-bt} + \int_0^t f(x - c(t-s), s)e^{-b(t-s)} ds. \quad (1.14)$$

Vemos entonces que $u(x, t)$ sólo depende de los valores de (x', t') tales que $x' - ct' = x - ct$, es decir, solo de los valores de u y f en la característica a través de (x, t) para $0 \leq t' \leq t$.

Este método de solución de (1.8) se puede extender fácilmente a ecuaciones no lineales de la forma

$$u_t + cu_x = f(x, t, u) \quad (1.15)$$

En nuestro caso, tomando $b = 0$, tenemos que la solución de

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) + c \frac{\partial u}{\partial x}(x, t) = f(x, t) \\ u(x, T_0) = u_0(x) \end{cases} \quad (1.16)$$

es la siguiente:

$$u(x, t) = u_0(x - ct) + \int_0^t f(x - c(t - s), s) ds. \quad (1.17)$$

1.1.2. Ecuaciones con velocidad variable.

Si c es una **función de x y de t** , $c = c(x, t)$, tenemos la ecuación

$$\frac{\partial u}{\partial t}(x, t) + c(x, t) \frac{\partial u}{\partial x}(x, t) = f(x, t). \quad (1.18)$$

Consideremos en primer lugar $f \equiv 0$:

$$\frac{\partial u}{\partial t}(x, t) + c(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad (1.19)$$

con condición inicial $u(x, 0) = u_0(x)$, y velocidad de propagación $c(x, t)$. Volvemos a aplicar el cambio de variable a τ y ξ , donde $\tau = t$ y ξ todavía no está determinado. Tenemos,

$$\frac{d\tilde{u}}{d\tau} = \frac{\partial \tilde{u}}{\partial \tau} u_t + \frac{\partial \tilde{u}}{\partial \xi} u_x = u_t + \frac{\partial \tilde{u}}{\partial \xi} u_x. \quad (1.20)$$

Análogamente al caso $c = cte$, establecemos,

$$\frac{dx}{d\tau} = c(x, t) = c(x, \tau). \quad (1.21)$$

Esta es una ecuación diferencial ordinaria para x , dando la velocidad a lo largo de la característica a través del punto (x, τ) , $c(x, \tau)$. Establecemos el valor inicial para la curva característica a través de (x, τ) para que sea ξ . Así, la ecuación (1.19) es equivalente al

sistema de ecuaciones diferenciales ordinarias siguiente:

$$\begin{cases} \frac{\partial \tilde{u}}{\partial \tau} = 0, & \tilde{u}(\xi, 0) = u_0(\xi) \\ \frac{\partial x}{\partial \tau} = c(x, \tau), & x(0) = \xi. \end{cases} \quad (1.22)$$

Como vemos en la primera ecuación de este sistema, u es constante a lo largo de cada curva característica, pero la característica no es necesariamente una recta. A continuación vemos un ejemplo que ilustra todo esto.

Ejemplo 1.1. Consideremos la ecuación:

$$u_t + xu_x = 0$$

$$u(x, 0) = \begin{cases} 1, & \text{si } 0 \leq x \leq 1, \\ 0 & \text{en otro caso.} \end{cases}$$

Correspondiendo a (1.22), tenemos las ecuaciones

$$\frac{\partial \tilde{u}}{\partial \tau} = 0, \quad \frac{\partial x}{\partial \tau} = x, \quad x(0) = \xi. \quad (1.23)$$

La solución general de la ecuación diferencial para $x(\tau)$ es $x(\tau) = ce^\tau$. Dado que especificamos que ξ es definido por $x(0) = \xi$, tenemos $x(\tau) = \xi e^\tau$, o $\xi = x e^{-t}$. La ecuación para \tilde{u} muestra que \tilde{u} es independiente de τ , entonces por la condición en $\tau = 0$ tenemos que

$$\tilde{u}(\xi, \tau) = u_0(\xi), \quad (1.24)$$

y por tanto,

$$u(x, t) = \tilde{u}(\xi, \tau) = u_0(\xi) = u_0(xe^{-t}). \quad (1.25)$$

Entonces tenemos, para $t > 0$,

$$u(x, t) = \begin{cases} 1, & \text{si } 0 \leq x \leq e^t, \\ 0 & \text{en otro caso. } \square \end{cases}$$

Igual que para las ecuaciones con coeficientes constantes, estos métodos pueden aplicarse a ecuaciones no lineales de la forma

$$u_t + c(x, t)u_x = f(x, t, u). \quad (1.26)$$

Capítulo 2

Aproximación numérica.

En la mayoría de los casos, no podemos resolver la ecuación del transporte mediante métodos elementales, ni siquiera podemos resolverla de forma exacta utilizando medios informáticos. Por ello se introducen los métodos de discretización, con el objetivo de aproximarse lo máximo posible a la solución. Para esta aproximación, analizaremos conceptos como convergencia, consistencia y estabilidad, además del orden de convergencia.

2.1. Convergencia.

La propiedad más básica que debe tener un esquema de aproximación para ser útil es que sus soluciones se aproximen a la solución de la *EDP* correspondiente y que la aproximación mejore a medida que los espaciados de la malla, h y k , tiendan a cero. En tal caso, diremos que el esquema es convergente.

Notación 2.1.

$$U_j^n \approx u(x_j, t_n)$$

Definición 2.2. Un esquema de aproximación en diferencias finitas de un paso de una ecuación en derivadas parciales es un esquema convergente si para cualquier solución de la ecuación en derivadas parciales, $u(x, t)$, y soluciones del esquema en diferencias finitas, U_j^n , tales que U_j^0 converge a $u_0(x)$ si jh converge a x , entonces U_j^n converge a $u(x, t)$ cuando (jh, nk) converge a (x, t) mientras h y k convergen a 0.

Esta definición no está completa hasta que clarifiquemos la naturaleza de la convergencia de U_j^n , definida en la cuadrícula, a $u(x, t)$ definida por continuidad variando (x, t) . Para esquemas multipaso, la definición asume que algún procedimiento de inicialización es usado para computar los primeros niveles de tiempo necesarios para emplear el esquema multipaso. Para el caso donde los datos se especifican en estos pasos de tiempo, la definición se ve alterada para requerir que U_j^n , para $0 \leq n \leq N$, converja a $u_0(x_j)$.

Definimos una malla uniforme en (x, t) :

$$\begin{cases} a = x_1 < x_2 < \cdots < x_{J-1} < x_J = b \\ T_0 = t_1 < t_2 < \cdots < t_{N-1} < t_N = T_f \end{cases} \quad (2.1)$$

con nodos equiespaciados

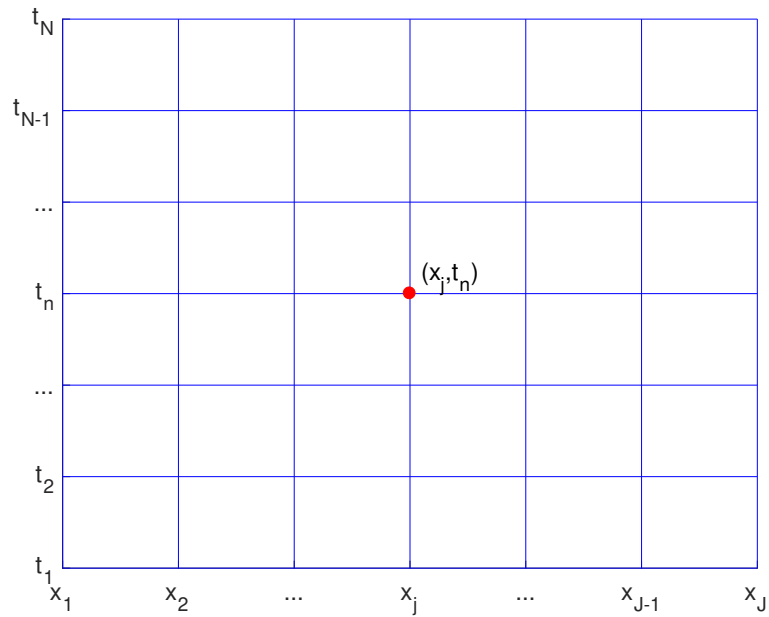
$$\begin{cases} x_j = a + (j-1)h & \text{para } j = 1, \dots, J, \text{ con } h = \frac{b-a}{J-1}, \\ t_n = T_0 + (n-1)k & \text{para } n = 1, \dots, N, \text{ con } k = \frac{T_f - T_0}{N-1}, \end{cases} \quad (2.2)$$

siendo h el paso de discretización de la variable espacial, y k el paso de discretización de la variable temporal. En la Figura (2.1) podemos ver una representación gráfica de la malla que acabamos de definir.

Buscamos en primer lugar una semidiscretización, es decir, que sea discreta en espacio y continua en tiempo, por lo que definimos las siguientes funciones continuas en tiempo:

$$u_j(t) = u(x_j, t) \quad (2.3)$$

El objetivo es reducir la *EDP* a un sistema de ecuaciones diferenciales cuya solución nos permita obtener una aproximación $U_j(t)$ de la solución $u = u(x, t)$ en el punto $x = x_j$. La forma más fácil de construir una semidiscretización es utilizando un desarrollo de Taylor.

Figura 2.1: Malla en (x, t)

Hacemos entonces un desarrollo de Taylor de orden 1 de la función u en el punto $(x_j + h, t) = (x_{j+1}, t)$ y centrada en (x_j, t) . Teniendo en cuenta que $x_{j+1} - x_j = h$, tenemos

$$u(x_{j+1}, t) = u(x_j, t) + u_x(x_j, t)h + O(h^2) \quad (2.4)$$

por lo que si despejamos $u_x(x_j, t)$, obtenemos

$$u_x(x_j, t) = \frac{u(x_{j+1}, t) - u(x_j, t)}{h} + O(h) \quad (2.5)$$

y por tanto, se tiene la siguiente aproximación:

$$u_x(x_i, t) \approx \frac{u(x_{i+1}, t) - u(x_i, t)}{h} \approx \frac{U_{i+1}(t) - U_i(t)}{h}. \quad (2.6)$$

Ahora, haciendo un desarrollo de Taylor de orden 1 de la función u en el punto $(x_j - h, t) = (x_{j-1}, t)$ y centrada en (x_j, t) , y teniendo en cuenta que $x_{j-1} - x_j = -h$, obtenemos:

$$u(x_{j-1}, t) = u(x_j, t) - u_x(x_j, t)h + O(h^2), \quad (2.7)$$

llegando a la aproximación

$$u_x(x_j, t) \approx \frac{u(x_j, t) - u(x_{j-1}, t)}{h} \approx \frac{U_j(t) - U_{j-1}(t)}{h}. \quad (2.8)$$

Por último, haciendo un desarrollo de Taylor de orden 2 de la función u en el punto $(x_j + h, t) = (x_{j+1}, t)$ y centrada en (x_j, t) y haciendo un desarrollo de Taylor de orden 2 de la función u en el punto $(x_j - h, t) = (x_{j-1}, t)$ y centrada en (x_j, t) , conseguimos:

$$\begin{cases} u(x_{j+1}, t) = u(x_j, t) + u_x(x_j, t)h + u_{xx}(x_j, t)\frac{h^2}{2} + O(h^3) \\ u(x_{j-1}, t) = u(x_j, t) - u_x(x_j, t)h + u_{xx}(x_j, t)\frac{h^2}{2} + O(h^3). \end{cases} \quad (2.9)$$

Si ahora restamos ambas ecuaciones y despejamos, se tiene

$$u_x(x_j, t) = \frac{u(x_{j+1}, t) - u(x_{j-1}, t)}{2h} + O(h^2). \quad (2.10)$$

Así, conseguimos las tres siguientes aproximaciones, que utilizaremos habitualmente en lo que resta:

$$u_x(x_j, t) \approx \frac{u(x_{j+1}, t) - u(x_j, t)}{h} \approx \frac{U_{j+1}(t) - U_j(t)}{h} \quad (2.11)$$

$$u_x(x_j, t) \approx \frac{u(x_j, t) - u(x_{j-1}, t)}{h} \approx \frac{U_j(t) - U_{j-1}(t)}{h} \quad (2.12)$$

$$u_x(x_j, t) \approx \frac{u(x_{j+1}, t) - u(x_{j-1}, t)}{2h} \approx \frac{U_{j+1}(t) - U_{j-1}(t)}{2h}. \quad (2.13)$$

Observamos que las aproximaciones (2.11) y (2.12) son de orden 1, mientras que la aproximación (2.13) es de orden dos. Además, corresponden a diferencias progresivas, regresivas y centradas, respectivamente.

2.2. Consistencia.

De forma intuitiva, la consistencia es una característica que nos dice que la solución exacta de una *EDP* satisface las ecuaciones en diferencias finitas de los métodos con un cierto margen de error, dado por la aproximación obtenida generalmente de las series de Taylor.

Definición 2.3. Dada una ecuación en derivadas parciales $P(\partial_t, \partial_x)u = f$ y un esquema en diferencias finitas, $P_{k,h}v = f$, decimos que el esquema en diferencias finitas es consistente con la ecuación en derivadas parciales si para cualquier función diferenciable $\phi(x, t)$, se tiene:

$$P\phi - P_{h,k}\phi \rightarrow 0 \text{ cuando } h, k \rightarrow 0 \quad (2.14)$$

siendo la convergencia puntual en cada punto de la malla.

2.3. Estabilidad.

La estabilidad es una característica que está relacionada con el comportamiento de la solución y su acotación cuando h y k tienden a 0. Veremos primero unos ejemplos muy sencillos en los que analizaremos su estabilidad, y a continuación daremos definiciones más generales de estabilidad. Definimos:

$$\hat{u}(\theta, t) = \sum_{j \in \mathbb{Z}} u_j(t) e^{ij\theta}. \quad (2.15)$$

Así, \hat{u} , en cada una de las aproximaciones (2.11), (2.12) y (2.13), satisface

$$\hat{u}_t(\theta, t) + \left(\frac{e^{-i\theta} - 1}{h} \right) \hat{u}(\theta, t) = 0, \quad t > 0 \quad (2.16)$$

$$\hat{u}_t(\theta, t) + \left(\frac{1 - e^{i\theta}}{h} \right) \hat{u}(\theta, t) = 0, \quad t > 0 \quad (2.17)$$

$$\hat{u}_t(\theta, t) + \left(\frac{e^{-i\theta} - e^{i\theta}}{2h} \right) \hat{u}(\theta, t) = 0, \quad t > 0. \quad (2.18)$$

La transformada de Fourier no sólo nos proporciona las anteriores ecuaciones diferenciales de parámetro θ , que son inmediatas de resolver, sino que, invirtiendo la fórmula (2.15):

$$u_j(t) = \frac{1}{2\pi} \int_0^{2\pi} \hat{u}(\theta, t) e^{-ij\theta} d\theta \quad (2.19)$$

se tiene

$$\frac{1}{2\pi} \int_0^{2\pi} |\hat{u}(\theta, t)|^2 d\theta = \sum_{j \in \mathbb{Z}} |u_j(t)|^2 \quad (2.20)$$

y por tanto, obtenemos

$$\hat{u}(\theta, t) = e^{a_h(\theta)t} \hat{u}(\theta, 0) \quad (2.21)$$

donde $a_h(\theta)$ es una función que mostramos para cada caso:

$$a(\theta) = \begin{cases} \frac{1 - e^{-i\theta}}{h}, & \text{(esquema progresivo),} \\ \frac{e^{i\theta} - 1}{h}, & \text{(esquema regresivo),} \\ \frac{e^{i\theta} - e^{-i\theta}}{2h}, & \text{(esquema centrado).} \end{cases}$$

Como sabemos, para que un método sea convergente es necesario que sea estable, y para ello se necesita que $Re(a_h(\theta))$ esté acotada superiormente si $h \rightarrow 0$ uniformemente en $\theta \in [0, 2\pi)$. Veamos qué pasa en cada esquema.

- Esquema progresivo:

$$a_h(\theta) = \frac{1 - e^{i\theta}}{h} = \frac{1 - \cos(\theta)}{h} - \frac{i \sin(\theta)}{h} \quad (2.22)$$

por lo que

$$Re(a_h(\theta)) = \frac{1 - \cos(\theta)}{h}. \quad (2.23)$$

Entonces, tenemos

$$Re(a_h(\theta)) \rightarrow \infty \text{ cuando } h \rightarrow 0, \forall \theta \in (0, 2\pi) \quad (2.24)$$

y por tanto, el esquema no es estable, y en consecuencia tampoco convergente.

- Esquema regresivo:

$$a_h(\theta) = \frac{e^{-i\theta} - 1}{h} = \frac{\cos(\theta) - 1}{h} - \frac{i \sin(\theta)}{h} \quad (2.25)$$

por lo que

$$Re(a_h(\theta)) = \frac{\cos(\theta) - 1}{h} \leq 0, \forall \theta \in [0, 2\pi). \quad (2.26)$$

Vemos que en este caso el esquema es estable, por lo que también es convergente.

- Esquema centrado:

$$a_h(\theta) = \frac{e^{-i\theta} - e^{i\theta}}{2h} = -\frac{i \sin(\theta)}{h}. \quad (2.27)$$

Claramente

$$\operatorname{Re}(a_h(\theta)) = 0 \quad (2.28)$$

por lo que este esquema es estable y convergente.

La estabilidad es una característica de los esquemas de aproximación numérica que se puede entender en varios sentidos, incluso puede variar su análisis en función del tipo de esquema que queramos utilizar. Así que vamos con una definición de estabilidad para un tipo concreto de esquemas.

Definición 2.4. Un esquema en diferencias finitas $P_{h,k}U_j^n = 0$ para una ecuación de primer orden homogénea es estable en una región de estabilidad Λ si existe un entero K tal que para cualquier tiempo T positivo, existe una constante C_T de forma que

$$h \sum_{j=-\infty}^{\infty} |U_j^n|^2 \leq C_T h \sum_{k=0}^K \sum_{j=-\infty}^{\infty} |U_j^k|^2 \quad (2.29)$$

para $0 \leq nk \leq T$, con $(h, k) \in \Lambda$.

Introducimos ahora la siguiente notación:

$$\|w\|_h = \left(h \sum_{j=-\infty}^{\infty} |w_j|^2 \right)^{\frac{1}{2}} \quad (2.30)$$

para cualquier función w definida en un dominio discretizado. Esta cantidad se conoce como la norma en L^2 de la función w y es una medida del tamaño de la solución. Con esta notación, la desigualdad (2.29) se puede reescribir como

$$\|U^n\|_h \leq \left(C_T \sum_{k=0}^K \|U^k\|^2 \right)^{\frac{1}{2}} \quad (2.31)$$

o equivalentemente:

$$\|U^n\|_h \leq \hat{C}_T \sum_{k=0}^K \|U^k\|^2 \quad (2.32)$$

para alguna constante \hat{C}_T . Las desigualdades (2.29) y (2.32) muestran que la norma de la solución en un instante t , con $0 \leq t \leq T$, está acotada por el crecimiento que puede tener. Este crecimiento es a lo sumo un múltiplo constante de la suma de las normas de la solución en los primeros $K + 1$ pasos, para un cierto k .

2.3.1. Análisis de Fourier.

El análisis de Fourier es la herramienta que nos va a permitir estudiar el concepto de estabilidad de forma muy extensa. Para una función $u(x)$ definida en \mathbb{R} , su transformada de Fourier es:

$$\hat{u}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-i\omega x} u(x) dx. \quad (2.33)$$

Esta transformación de u es una función de la variable real ω y se define de forma única para u , es decir, la función \hat{u} es un tipo de representación de u , y por tanto, algunas propiedades de u se pueden obtener a partir de \hat{u} . Se tiene la siguiente fórmula de inversión de Fourier:

$$u(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{i\omega x} \hat{u}(\omega) d\omega \quad (2.34)$$

y de esta forma u puede ser recuperado teniendo \hat{u} . Ésta última fórmula expresa u como una superposición de ondas dadas por $e^{i\omega x}$, con amplitudes $\hat{u}(\omega)$. Nótese que $\hat{u}(\omega)$ puede

tener valores complejos incluso cuando $u(x)$ toma valores reales.

Supongamos ahora que u es una función definida en el conjunto de los enteros $j \in \mathbb{Z}$. Su transformada de Fourier se define de la forma

$$\hat{u}(\xi) = \frac{1}{\sqrt{2\pi}} \sum_{j=-\infty}^{+\infty} e^{-ij\xi} u_j \quad (2.35)$$

donde $\xi \in [-\pi, \pi]$ y $\hat{u}(-\pi) = \hat{u}(\pi)$. La fórmula de inversión de Forier en este caso es

$$u_j = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{ij\xi} \hat{u}(\xi) d\xi. \quad (2.36)$$

El análisis de Fourier para funciones definidas en \mathbb{Z} que acabamos de hacer, es el mismo que para funciones definidas en un intervalo. En una serie de Fourier, normalmente se empieza con una función $\hat{u}(\xi)$ definida en el intervalo $[-\pi, \pi]$ y se representa en la forma dada por la ecuación (2.35), donde los coeficientes u_j se obtienen de (2.36). Pero en el estudio de esquemas de aproximación en diferencias finitas es más habitual tomar las funciones u_j y tomar la ecuación (2.36) como su representación, ya que ambas formas son equivalentes.

Si ahora suponemos que la distancia entre los puntos del espacio discreto donde está definida u_j es h , tenemos que la transformada de Fourier es

$$\hat{u}(\xi) = \frac{1}{\sqrt{2\pi}} \sum_{j=-\infty}^{+\infty} e^{-ijh\xi} u_j h \quad (2.37)$$

para $\xi \in [-\frac{\pi}{h}, \frac{\pi}{h}]$, y la fórmula de inversión de Fourier pasa a ser

$$u_j = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} e^{ijh\xi} \hat{u}(\xi) d\xi. \quad (2.38)$$

De las definiciones anteriores podemos concluir que la norma de u en L^2 ,

$$\|u\| = \left(\int_{-\infty}^{+\infty} |u(x)|^2 dx \right)^{\frac{1}{2}} \quad (2.39)$$

es la misma que la norma de $\hat{u}(\omega)$ en L^2 , es decir,

$$\int_{-\infty}^{+\infty} |u(x)|^2 dx = \int_{-\infty}^{+\infty} |\hat{u}(\omega)|^2 d\omega. \quad (2.40)$$

Esto tambien ocurre en la transformada discreta, por lo que tenemos,

$$\|\hat{u}\|_h^2 = \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} |\hat{u}(\xi)|^2 d\xi = \sum_{j=-\infty}^{\infty} |u_j|^2 h = \|u\|_h^2. \quad (2.41)$$

Las ecuaciones (2.40) y (2.41) se llaman relaciones de Parseval, de las cuales se deduce que la transformada de Fourier está definida para todas las funciones en $L^2(\mathbb{R})$ y $L^2(h\mathbb{Z})$.

2.3.2. Análisis de Von-Neumann.

Una de las grandes aplicaciones del análisis de Fourier, es el análisis de estabilidad de Von-Neumann de esquemas en diferencias finitas. Las ventajas de este método de análisis son que es fácil de aplicar y que es un método muy general.

Vamos a explicar el análisis de la estabilidad de Von-Neumann mediante el siguiente esquema (FTBS). A continuación, haremos un análisis más general. Utilizando la transformación de Fourier, el análisis de la estabilidad se reduce a hacer una serie de observaciones algebraicas. Se considera el esquema

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_j^n - U_{j-1}^n}{h} = 0 \quad (2.42)$$

de donde, si despejamos U_j^{n+1} y tomamos $\lambda = \frac{k}{h}$, se tiene

$$U_j^{n+1} = (1 - c\lambda)U_j^n + c\lambda U_{j-1}^n. \quad (2.43)$$

Usando la fórmula de inversión de Fourier (2.38) para U^n , tenemos:

$$U_j^n = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} e^{ijh\xi} \hat{U}^n(\xi) d\xi \quad (2.44)$$

y sustituyendo esto en (2.43) para U_j^n y U_{j-1}^n , obtenemos:

$$U_j^{n+1} = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} e^{ijh\xi} [(1 - c\lambda) + c\lambda e^{-ih\xi}] \hat{U}^n(\xi) d\xi. \quad (2.45)$$

Si comparamos esta ecuación con la fórmula de inversión de Fourier

$$U_j^{n+1} = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} e^{ijh\xi} \hat{U}^{n+1}(\xi) d\xi \quad (2.46)$$

y teniendo en cuenta que la transformada de Fourier es única, tenemos que los integrandos son iguales, de donde

$$\hat{U}^{n+1}(\xi) = [(1 - c\lambda) + c\lambda e^{-ih\xi}] \hat{U}^n(\xi) = g(h\xi) \hat{U}^n(\xi). \quad (2.47)$$

El número $g(h\xi) = (1 - c\lambda) + c\lambda e^{-ih\xi}$ se llama *factor de amplificación* que, como podemos ver en la igualdad anterior, al multiplicar este factor por la transformada de Fourier de la solución obtenemos el siguiente paso en tiempo del esquema. El factor de amplificación recibe ese nombre porque su magnitud se corresponde con la cantidad en la que la amplitud de cada frecuencia en la solución, $\hat{U}^n(\xi)$, se incrementa en cuando se avanza en un paso temporal. Obtenemos entonces la siguiente fórmula, muy importante,

$$\hat{U}^n(\xi) = g(h\xi)^n \hat{U}^0(\xi) \quad (2.48)$$

donde cabe destacar que el superíndice del factor de amplificación es una potencia, mientras que en el caso de \hat{U} se corresponde con el índice del nodo temporal.

Gracias al método general que nos proporciona la transformada de Fourier, cualquier esquema de un paso se puede poner de la forma anterior, lo que servirá para analizar su estabilidad y orden de convergencia.

Vamos a continuar con el análisis de la estabilidad del esquema (2.42) utilizando la fórmula (2.48). Por la relación de Parseval (2.41), se tiene:

$$h \sum_{j=-\infty}^{\infty} |U_j^n|^2 = \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} |\hat{U}^n(\xi)|^2 d\xi = \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} |g(h\xi)|^{2n} |\hat{U}^0(\xi)|^2 d\xi. \quad (2.49)$$

Así, vemos que la condición de estabilidad dada por (2.29) se cumple, con $K = 0$, si $|g(h\xi)|^{2n}$ está correctamente acotada. Calculemos $|g(h\xi)|$, y para ello tomamos $\theta = h\xi$:

$$\begin{aligned} |g(\theta)|^2 &= |(1 - c\lambda) + c\lambda e^{-i\theta}|^2 = (1 - c\lambda + c\lambda \cos \theta)^2 + c^2 \lambda^2 \sin^2 \theta \\ &= (1 - 2c\lambda \sin^2 \frac{\theta}{2})^2 + 4c^2 \lambda^2 \sin^2 \frac{\theta}{2} \cos^2 \frac{\theta}{2} \\ &= 1 - 4c\lambda \sin^2 \frac{\theta}{2} + 4c^2 \lambda^2 \sin^4 \frac{\theta}{2} + 4c^2 \lambda^2 \sin^2 \frac{\theta}{2} \cos^2 \frac{\theta}{2} \\ &= 1 - 4c\lambda(1 - c\lambda) \sin^2 \frac{\theta}{2}. \end{aligned}$$

Podemos ver que $|g(\theta)|$ está acotado por 1 si $0 \leq c\lambda \leq 1$. Así, gracias a la ecuación (2.48), se tiene

$$h \sum_{j=-\infty}^{\infty} |U_j^n|^2 \leq \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} |\hat{U}^0(\xi)|^2 d\xi = h \sum_{j=-\infty}^{\infty} |U_j^0|^2 \quad (2.50)$$

y así el esquema es estable por la Definición (2.4). Sin embargo, si $c\lambda \notin [0, 1]$, fijando λ cuando h y k tienden a cero, se tiene que $|g(\theta)|$ es mayor que uno para algún valor de θ , por lo que el esquema será inestable. A continuación hablamos más en profundidad sobre esto, introduciendo una condición de estabilidad más general.

Aunque en el ejemplo anterior hemos considerado que el factor de amplificación g dependía sólo de $\theta = h\xi$, en general éste puede depender de h y de k , por lo que en el siguiente teorema ofrecemos una condición de estabilidad para esquemas con coeficientes constantes donde el factor de amplificación no sólo depende de θ .

Teorema 2.5. *Un esquema en diferencias finitas de un paso con coeficientes constantes es estable si y solo si existe una constante K , independiente de θ , h y k , y existen pasos de discretización h_0 y k_0 tales que*

$$|g(\theta, h, k)| \leq 1 + Kk \quad (2.51)$$

para todo θ , $0 < k \leq k_0$, $0 < h \leq h_0$. Si $g(\theta, h, k)$ no depende de h ni de k , la desigualdad anterior puede ser reemplazada por

$$|g(\theta)| \leq 1. \quad (2.52)$$

Como podemos ver en el anterior teorema, la estabilidad es una característica que se puede analizar simplemente con el factor de amplificación, observación realizada por Von-Neumann, de ahí el nombre de este tipo de análisis.

Cabe destacar que, a la hora de la práctica, no es necesario obtener las integrales de (2.37) para obtener el factor de amplificación. Una forma equivalente de obtenerlo es sustituir U_j^n en el esquema por $g^n e^{ij\theta}$ para cada valor de j y n . Resolviendo la ecuación resultante, obtenemos el factor de amplificación. Vamos a ver esto en un ejemplo:

Ejemplo 2.6. Analicemos la estabilidad del siguiente esquema (FTCS):

$$\frac{U_j^{n+1} - U_j^n}{k} + c \frac{U_{j+1}^n - U_{j-1}^n}{2h} = 0. \quad (2.53)$$

Si reemplazamos U_j^n en el esquema por $g^n e^{ijh\xi}$, obtenemos,

$$\frac{g^{n+1} e^{ij\theta} - g^n e^{ij\theta}}{k} + c \frac{g^n e^{i(j+1)\theta} - g^n e^{i(j-1)\theta}}{2h} = g^n e^{ij\theta} \left(\frac{g-1}{k} + c \frac{e^{i\theta} - e^{-i\theta}}{2h} \right) = 0 \quad (2.54)$$

por lo que si resolvemos la ecuación, se tiene que

$$g = 1 - ic\lambda \sin \theta, \quad (2.55)$$

donde $\lambda = \frac{k}{h}$. Si λ es constante, entonces el factor de amplificación no depende ni de h ni de k , por lo que

$$|g(\theta)|^2 = 1 + c^2 \lambda^2 \sin^2 \theta. \quad (2.56)$$

Ya que $|g(\theta)|$ es mayor que 1 para θ distinto de 0 o de π , deducimos del Teorema (2.5) que el esquema FTCS no es estable.

Ese mismo procedimiento puede ser aplicado a cualquier esquema de un paso para determinar la forma del factor de amplificación.

Corolario 2.7. *Si un esquema en las condiciones del Teorema (2.5) se modifica de forma que solo se ve alterado el factor de amplificación por la suma de una cantidad $O(k)$ uniformemente en ξ , entonces el esquema modificado es estable si y solo si lo es el original.*

Demostración. Sea g el factor de amplificación del esquema. Si g satisface $|g| \leq 1 + Kk$, entonces el factor de amplificación del esquema modificado, g' , satisface

$$|g'| = |g + O(k)| \leq 1 + Kk + Ck = 1 + K'k. \quad (2.57)$$

Por tanto, el esquema modificado es estable si lo es el original, y viceversa. □

El uso del Teorema (2.5) y del Corolario anterior, nos va a permitir analizar la estabilidad de una gran cantidad de esquemas de un paso.

2.3.3. Estabilidad en esquemas generales.

Un esquema es estable para la ecuación $P_{h,k}U = f$ si lo es para la ecuación $P_{h,k}U = 0$. Por tanto, todo el análisis que hemos realizado para el caso $f \equiv 0$ nos sirve para cualquier otra función f . Veamos qué pasa cuando los coeficientes son variables, ya que hasta ahora habíamos considerado que la velocidad de propagación c era constante.

Las condiciones de estabilidad obtenidas para esquemas con coeficientes constantes se pueden usar para obtener las condiciones de estabilidad del mismo esquema con coeficientes variables. Veamos esto a través de un ejemplo, en el que se considera el esquema de *Lax-Friedrichs*:

$$\frac{U_j^{n+1} - \frac{1}{2}(U_{j+1}^n + U_{j-1}^n)}{k} + c \frac{U_{j+1}^n - U_{j-1}^n}{2h} = 0 \quad (2.58)$$

y lo aplicamos a la ecuación

$$u_t + c(x, t)u_x = 0, \quad (2.59)$$

obteniendo

$$U_j^{n+1} = \frac{1}{2}(U_{j+1}^n + U_{j-1}^n) - \frac{1}{2}c(x_j, t_n)\lambda(U_{j+1}^n - U_{j-1}^n). \quad (2.60)$$

La condición de estabilidad para este esquema es que $|c(x_j, t_n)|\lambda \leq 1$ se satisfaga para todos los valores de (x_j, t_n) en el dominio de computación. La explicación de esto es que podemos considerar los *problemas del coeficiente congelado*, es decir, cada problema cuyos coeficientes sean constantes tras fijar el valor de los coeficientes que en un principio eran variables. Los valores que se fijan tienen que ser todos los posibles del dominio de computación, y de esta manera, si cada problema del coeficiente congelado es estable, también lo será el problema de coeficientes variables.

Por último, si la condición de estabilidad obtenida de los problemas del coeficiente congelado no se cumple en una pequeña región, la inestabilidad resultante se originará en dicha región y no crecerá fuera de ese área.

2.4. Orden.

Hemos visto anteriormente que un esquema se considera aceptable si es convergente, mientras que se considera inaceptable si no lo es. Pues bien, dos esquemas convergentes pueden diferir mucho en cómo sus soluciones se aproximan a la solución de la ecuación. Vamos a obtener el esquema *Lax-Wendroff* y posteriormente daremos una definición del orden de convergencia.

Consideramos un desarrollo de Taylor de orden 2 de la función u en el punto $(x, t+k)$ y centrado en (x, t) . Teniendo en cuenta que $(t+k) - t = k$, tenemos:

$$u(x, t+k) = u(x, t) + ku_t(x, t) + \frac{k^2}{2}u_{tt}(x, t) + O(k^3). \quad (2.61)$$

Apoyándonos en la EDP

$$u_t + cu_x = f \quad (2.62)$$

tenemos

$$\begin{aligned}
 u_{tt} &= \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial t} \left(\frac{\partial u}{\partial t} \right) = \frac{\partial}{\partial t} \left(f - c \frac{\partial u}{\partial x} \right) = f_t - c \frac{\partial}{\partial t} \left(\frac{\partial u}{\partial x} \right) \\
 &= f_t - c \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial t} \right) = f_t - c \frac{\partial}{\partial x} \left(f - c \frac{\partial u}{\partial x} \right) \\
 &= f_t - cf_x + c^2 \frac{\partial^2 u}{\partial x^2} = f_t - cf_x + c^2 u_{xx}.
 \end{aligned}$$

Si ahora sustituimos en (2.61), se tiene:

$$u(x, t+k) = u(x, t) - ck u_x(x, t) + \frac{c^2 k^2}{2} u_{xx}(x, t) + kf - \frac{ck^2}{2} f_x + \frac{k^2}{2} f_t + O(k^3). \quad (2.63)$$

Si ahora sustituimos las derivadas con respecto a x por aproximaciones de orden 2, y sustituimos la derivada de f con respecto a t por un esquema progresivo, tenemos,

$$\begin{aligned}
 u(x, t+k) &= u(x, t) - ck \frac{u(x+h, t) - u(x-h, t)}{2h} + \frac{c^2 k^2}{2} \frac{u(x+h, t) - 2u(x, t) + u(x-h, t)}{h^2} \\
 &+ \frac{k}{2} [f(x, t+k) + f(x, t)] - \frac{ck^2}{2} \frac{[f(x+h, t) - f(x-h, t)]}{2h} + O(kh^2) + O(k^3)
 \end{aligned}$$

lo que sugiere el esquema de Lax-Wendroff:

$$\begin{aligned}
 \frac{U_j^{n+1} - U_j^n}{k} + c \frac{U_{j+1}^n - U_{j-1}^n}{2h} - \frac{c^2 k}{2} \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{h^2} \\
 = \frac{1}{2} (f_j^{n+1} + f_j^n) - \frac{ck}{4h} (f_{j+1}^n - f_{j-1}^n)
 \end{aligned} \quad (2.64)$$

donde si despejamos U_j^{n+1} , llegamos a

$$\begin{aligned}
U_j^{n+1} = & U_j^n - \frac{c\lambda}{2}(U_{j+1}^n - U_{j-1}^n) + \frac{c^2\lambda^2}{2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n) \\
& + \frac{k}{2}(f_j^{n+1} + f_j^n) - \frac{ck\lambda}{4}(f_{j+1}^n - f_{j-1}^n)
\end{aligned} \tag{2.65}$$

siendo $f_j^n = f(x_j, t_n)$.

En general, cualquier esquema para la ecuación en derivadas parciales $Pu = f$ puede ser escrito de la forma $P_{h,k}U = R_{h,k}f$. En base a esto, vamos a dar una definición de orden de convergencia de un esquema.

Definición 2.8. Un esquema $P_{h,k}U = R_{h,k}f$ consistente con la ecuación diferencial $Pu = f$ se dice que es de orden p en espacio y de orden q en tiempo si para cualquier función diferenciable $\phi(x, t)$ se tiene:

$$P_{h,k}\phi - R_{h,k}P\phi = O(h^p) + O(k^q). \tag{2.66}$$

A la cantidad $P_{h,k}\phi - R_{h,k}P\phi$ se le conoce como error de truncamiento del esquema.

Ejemplo 2.9. Vamos a utilizar esta definición para analizar el orden de convergencia del esquema de Lax-Wendroff. Utilizando una serie de Taylor evaluada en (x_j, t_n) en el lado izquierdo de la igualdad (2.64), obtenemos

$$\phi_t + \frac{k}{2}\phi_{tt} + c\phi_x - \frac{c^2k}{2}\phi_{xx} + O(h^2) + O(k^2). \tag{2.67}$$

Para una función diferenciable $f(x, t)$, el lado derecho de (2.64) pasa a ser

$$f + \frac{k}{2}f_t - \frac{ck}{2}f_x + O(h^2) + O(k^2). \tag{2.68}$$

Ahora, si $f = \phi_t + c\phi_x = P\phi$, se tiene

$$\phi_t + c\phi_x + \frac{k}{2}\phi_{tt} + \frac{k}{2}c\phi_{xt} - \frac{ck}{2}\phi_{xt} - \frac{c^2k}{2}\phi_{xx} + O(h^2) + O(k^2) \quad (2.69)$$

y teniendo en cuenta que $\phi_{xt} = \phi_{tx}$, llegamos a que

$$P_{h,k}\phi - R_{h,k}P\phi = O(h^2) + O(k^2) \quad (2.70)$$

por lo que el esquema Lax-Wendroff converge con orden 2 en espacio y orden 2 en tiempo.

Capítulo 3

Esquemas de aproximación.

En este capítulo, introduciremos una serie de esquemas de aproximación para la ecuación del transporte, y analizaremos su consistencia, estabilidad, convergencia y orden de convergencia.

3.1. FTFS

El primer esquema que analizamos es el FTFS (Forward in Time, Forward in Space).

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_{j+1}^n - U_j^n}{h} = f(x_j, t_n) \quad \begin{cases} 1 \leq j \leq J-1, \\ 1 \leq n \leq N-1. \end{cases} \quad (3.1)$$

donde $c(x_j, t_n) < 0, \forall(x_j, t_n)$.

- Consistencia.

El operador P de la Definición 2.3 es

$$P(\partial_t, \partial_x) = \frac{\partial}{\partial t} + c \frac{\partial}{\partial x}. \quad (3.2)$$

Así, tenemos

$$P\phi = \phi_t + c\phi_x. \quad (3.3)$$

Para este esquema, se tiene también

$$P_{h,k}\phi = \frac{\phi_j^{n+1} - \phi_j^n}{k} + c \frac{\phi_{j+1}^n - \phi_j^n}{h} \quad (3.4)$$

donde

$$\phi_j^n = \phi(x_j, t_n). \quad (3.5)$$

Si ahora hacemos un desarrollo de Taylor de la función ϕ en el punto $(x_j + h, t_n) = (x_{j+1}, t_n)$ centrado en (x_j, t_n) , y un desarrollo de Taylor de la función ϕ en el punto $(x_j, t_n + k) = (x_j, t_{n+1})$ centrado en (x_j, t_n) , obtenemos

$$\phi_{j+1}^n = \phi_j^n + \phi_x h + \phi_{xx} \frac{h^2}{2} + O(h^3) \quad (3.6)$$

$$\phi_j^{n+1} = \phi_j^n + \phi_t k + \phi_{tt} \frac{k^2}{2} + O(k^3) \quad (3.7)$$

donde las derivadas de los miembros de la derecha en ambas ecuaciones están evaluadas en (x_j, t_n) , y así,

$$P_{h,k}\phi = \phi_t + c\phi_x + \phi_{tt} \frac{k}{2} + c\phi_{xx} \frac{h}{2} + O(k^2) + O(h^2). \quad (3.8)$$

Entonces, se tiene

$$P\phi - P_{h,k}\phi = -\phi_{tt} \frac{k}{2} - c\phi_{xx} \frac{h}{2} + O(k^2) + O(h^2) \rightarrow 0 \text{ cuando } h, k \rightarrow 0 \quad (3.9)$$

lo que indica que el esquema FTFS es consistente.

- Estabilidad.

Vamos a analizar la estabilidad de los esquemas de la forma $U_j^{n+1} = \alpha U_j^n + \beta U_{j+1}^n$, de los cuales el esquema FTFS es un caso particular. Tenemos:

$$\begin{aligned}
\sum_{j=-\infty}^{\infty} |U_j^{n+1}|^2 &= \sum_{j=-\infty}^{\infty} |\alpha U_j^n + \beta U_{j+1}^n|^2 \\
&\leq \sum_{j=-\infty}^{\infty} |\alpha|^2 |U_j^n|^2 + 2|\alpha||\beta| |U_j^n| |U_{j+1}^n| + |\beta|^2 |U_{j+1}^n|^2 \\
&\leq \sum_{j=-\infty}^{\infty} |\alpha|^2 |U_j^n|^2 + |\alpha||\beta| (|U_j^n|^2 + |U_{j+1}^n|^2) + |\beta|^2 |U_{j+1}^n|^2 \\
&= \sum_{j=-\infty}^{\infty} (|\alpha|^2 + 2|\alpha||\beta| + |\beta|^2) |U_j^n|^2 \\
&= (|\alpha| + |\beta|)^2 \sum_{j=-\infty}^{\infty} |U_j^n|^2
\end{aligned}$$

donde hemos utilizado las desigualdades $2xy \leq x^2 + y^2$, $|x+y| \leq |x| + |y|$, y la igualdad $\sum_{n=-\infty}^{\infty} a_n = \sum_{n=-\infty}^{\infty} a_{n+1}$. Así, este tipo de esquemas son estables si $|\alpha| + |\beta| \leq 1$.

En el caso particular del esquema FTFS, si despejamos U_j^{n+1} , tenemos

$$U_j^{n+1} = U_j^n - \frac{ck}{h}(U_{j+1}^n - U_{j-1}^n) = (1 + \frac{ck}{h})U_j^n - \frac{ck}{h}U_{j+1}^n \quad (3.10)$$

por lo que, tomando $\lambda = \frac{k}{h}$, y teniendo en cuenta que $c(x_j, t_n) < 0$, $\forall (x_j, t_n)$, la condición de estabilidad se reduce a

$$|1 + c\lambda| + |c\lambda| \leq 1 \Leftrightarrow -1 \leq c\lambda \leq 0 \Leftrightarrow |c\lambda| \leq 1 \quad (3.11)$$

por lo que el esquema FTFS es estable si

$$|c|k \leq h. \quad (3.12)$$

- Orden de convergencia.

Este esquema se ha obtenido utilizando una aproximación en diferencias finitas progresiva tanto para la derivada espacial como temporal. Utilizando el polinomio de Taylor de u evaluado en (x_j, t_{n+1}) y centrado en (x_j, t_n) , y el polinomio de Taylor de u evaluado en (x_{j+1}, t_n) y centrado en (x_j, t_n)

$$\frac{\partial u}{\partial x}(x_j, t_n) = \frac{u(x_{j+1}, t_n) - u(x_j, t_n)}{h} + O(h) \quad (3.13)$$

$$\frac{\partial u}{\partial t}(x_j, t_n) = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{k} + O(k) \quad (3.14)$$

y, sustituyendo en la EDP, se tiene

$$\frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{k} + c(x_j, t_n) \frac{u(x_{j+1}, t_n) - u(x_j, t_n)}{h} = f(x_j, t_n) + O(h) + O(k), \quad (3.15)$$

por lo que el esquema FTFS es de orden 1 en espacio y de orden 1 en tiempo.

3.2. FTBS

El siguiente esquema que analizamos es el FTBS (Forward in Time, Backward in Space).

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_j^n - U_{j-1}^n}{h} = f(x_j, t_n) \quad \begin{cases} 2 \leq j \leq J, \\ 1 \leq n \leq N - 1. \end{cases} \quad (3.16)$$

donde $c(x_j, t_n) > 0, \forall(x_j, t_n)$.

- Consistencia.

El operador P de la Definición 2.3 es

$$P(\partial_t, \partial_x) = \frac{\partial}{\partial t} + c \frac{\partial}{\partial x}. \quad (3.17)$$

Así, tenemos

$$P\phi = \phi_t + c\phi_x. \quad (3.18)$$

Para este esquema, se tiene tambien

$$P_{h,k}\phi = \frac{\phi_j^{n+1} - \phi_j^n}{k} + c \frac{\phi_j^n - \phi_{j-1}^n}{h}, \quad (3.19)$$

donde

$$\phi_j^n = \phi(x_j, t_n). \quad (3.20)$$

Si ahora hacemos un desarrollo de Taylor de la función ϕ en el punto $(x_j - h, t_n) = (x_{j-1}, t_n)$ centrado en (x_j, t_n) , y un desarrollo de Taylor de la función ϕ en el punto

$(x_j, t_n + k) = (x_j, t_{n+1})$ centrado en (x_j, t_n) , obtenemos

$$\phi_{j-1}^n = \phi_j^n - \phi_x h + \phi_{xx} \frac{h^2}{2} + O(h^3) \quad (3.21)$$

$$\phi_j^{n+1} = \phi_j^n + \phi_t k + \phi_{tt} \frac{k^2}{2} + O(k^3), \quad (3.22)$$

donde las derivadas de los miembros de la derecha en ambas ecuaciones están evaluadas en (x_j, t_n) , y así:

$$P_{h,k}\phi = \phi_t + c\phi_x + \phi_{tt} \frac{k}{2} - c\phi_{xx} \frac{h}{2} + O(k^2) + O(h^2). \quad (3.23)$$

Entonces, se tiene

$$P\phi - P_{h,k}\phi = -\phi_{tt} \frac{k}{2} + c\phi_{xx} \frac{h}{2} + O(k^2) + O(h^2) \rightarrow 0 \text{ cuando } h, k \rightarrow 0, \quad (3.24)$$

lo que indica que el esquema FTBS es consistente.

- Estabilidad.

Teniendo en cuenta que $c(x_j, t_n) > 0, \forall(x_j, t_n)$, y realizando cálculos análogos al esquema anterior, llegamos a que el esquema FTBS es estable si

$$ck \leq h. \quad (3.25)$$

- Orden de convergencia.

Análogamente al esquema FTFS, pero en este caso utilizando una aproximación en diferencias finitas regresiva en la derivada espacial, se tiene que el esquema FTBS es

de orden 1 en espacio y de orden 1 en tiempo.

3.3. FTFS-FTBS

Hasta ahora hemos definido dos esquemas, en el FTFS la velocidad $c(x_j, t_n)$ debe ser siempre negativa y en el FTBS debe ser siempre positiva. Pero en la práctica es posible que la velocidad cambie de signo, por lo que introducimos el siguiente esquema, donde no es necesario que la velocidad tenga signo constante:

$$f(x_j, t_n) = \begin{cases} \frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_{j+1}^n - U_j^n}{h}, & \text{si } c(x_j, t_n) \leq 0, \\ \frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_j^n - U_{j-1}^n}{h}, & \text{si } c(x_j, t_n) > 0. \end{cases}$$

Este esquema mantiene las propiedades de los dos esquemas de los que proviene por separado, por lo que es un esquema consistente, estable si $|c|k \leq h$, independientemente del signo de c , y converge con orden 1 en espacio y orden 1 en tiempo.

Ejecutando en MATLAB el programa correspondiente a este esquema, que está en el Anexo a este documento, obtenemos los resultados dados por la siguiente tabla, donde en la primera columna podemos ver el número de nodos espaciales I , junto con el paso de discretización espacial h , en la segunda columna tenemos el número de nodos temporales N , junto con el paso de discretización temporal k , en la tercera columna se encuentra el error relativo entre la solución exacta y la solución aproximada por el programa, y en la cuarta y última columna tenemos unas breves observaciones para cada caso. El texto en rojo indica un comportamiento inestable, ya que en esos casos no se cumple la condición de estabilidad.

J, h	N, k	Error relativo	Observaciones
101, 10^{-2}	41, $3 \cdot 10^{-2}$	$4.35532 \cdot 10^3$	Comportamiento inestable.
101, 10^{-2}	53, $2.307692 \cdot 10^{-2}$	$2.66180 \cdot 10^{-1}$	Gráfica de $u(\cdot, T_f)$ con oscilaciones espurias.
101, 10^{-2}	151, $8 \cdot 10^{-3}$	$5.21553 \cdot 10^{-4}$	Comportamiento estable.
1001, 10^{-3}	1501, $8 \cdot 10^{-4}$	$5.74262 \cdot 10^{-5}$	Reducción esperada para el orden teórico.

Cuadro 3.1: Resultados FTFS

3.4. Lax-Wendroff

El esquema FTCS (Forward in Time, Centered in Space)

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_{j+1}^n - U_{j-1}^n}{2h} = f(x_j, t_n) \quad (3.26)$$

no converge, ya que no es estable, tal y como hemos visto en el Ejemplo (2.6), por eso introducimos un nuevo término que tiende a 0, obteniendo el esquema de Lax-Wendroff, que sí es estable y convergente:

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_{j+1}^n - U_{j-1}^n}{2h} - \frac{c^2 k}{2} \frac{U_{j-1}^n - 2U_j^n + U_{j+1}^n}{2h} = f(x_j, t_n) \quad (3.27)$$

$$\begin{cases} 2 \leq j \leq J-1, \\ 1 \leq n \leq N-1. \end{cases}$$

- Consistencia.

El operador P de la Definición 2.3 es

$$P(\partial_t, \partial_x) = \frac{\partial}{\partial t} + c \frac{\partial}{\partial x}. \quad (3.28)$$

Así, tenemos

$$P\phi = \phi_t + c\phi_x. \quad (3.29)$$

Para este esquema, se tiene tambien

$$P_{h,k}\phi = \frac{\phi_j^{n+1} - \phi_j^n}{k} + c \frac{\phi_{j+1}^n - \phi_{j-1}^n}{2h} - \frac{c^2 k}{2} \frac{\phi_{j-1}^n - 2\phi_j^n + \phi_{j+1}^n}{h^2}, \quad (3.30)$$

donde

$$\phi_j^n = \phi(x_j, t_n). \quad (3.31)$$

Si ahora hacemos un desarrollo de Taylor de la función ϕ en el punto $(x_j - h, t_n) = (x_{j-1}, t_n)$ centrado en (x_j, t_n) , un desarrollo de Taylor de la función ϕ en el punto $(x_j + h, t_n) = (x_{j+1}, t_n)$ centrado en (x_j, t_n) y un desarrollo de Taylor de la función ϕ en el punto $(x_j, t_n + k) = (x_j, t_{n+1})$ centrado en (x_j, t_n) , obtenemos:

$$\phi_{j-1}^n = \phi_j^n - \phi_x h + \phi_{xx} \frac{h^2}{2} + O(h^3) \quad (3.32)$$

$$\phi_{j+1}^n = \phi_j^n + \phi_x h + \phi_{xx} \frac{h^2}{2} + O(h^3) \quad (3.33)$$

$$\phi_j^{n+1} = \phi_j^n + \phi_t k + \phi_{tt} \frac{k^2}{2} + O(k^3), \quad (3.34)$$

donde las derivadas de los miembros de la derecha en las tres ecuaciones están evaluadas en (x_j, t_n) , y así

$$P_{h,k}\phi = \phi_t + c\phi_x + \phi_{tt} \frac{k}{2} - \frac{c^2 k}{2} \phi_{xx} + O(k^2) + O(h^2). \quad (3.35)$$

Entonces, se tiene

$$P\phi - P_{h,k}\phi = -\phi_{tt}\frac{k}{2} + \frac{c^2k}{2}\phi_{xx} + O(k^2) + O(h^2) \rightarrow 0 \text{ cuando } h, k \rightarrow 0, \quad (3.36)$$

lo que indica que el esquema de Lax-Wendroff es consistente.

- Estabilidad.

Realizando cálculos similares a los esquemas anteriores, sellega a que el esquema de Lax-Wendroff es estable si

$$ck \leq h. \quad (3.37)$$

- Orden de convergencia.

Tal y como hemos demostrado en el Ejemplo (2.9), el esquema de Lax-Wendroff converge con orden 2 en espacio y con orden 2 en tiempo.

3.5. BTFS

El siguiente esquema que analizamos es el BTFS (Backward in Time, Forward in Space), que es la versión implícita del esquema FTFS (3.1).

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_{j+1}^{n+1} - U_j^{n+1}}{h} = f(x_j, t_n) \quad \begin{cases} 1 \leq j \leq J-1, \\ 1 \leq n \leq N-1. \end{cases} \quad (3.38)$$

Este esquema es la versión implícita del FTFS. Es un esquema consistente, pero no siempre estable ni convergente. Diferenciamos dos casos:

- Si $c(x_j, t_n) < 0$, es estable, y converge con orden $O(h) + O(k)$.
- Si $c(x_j, t_n) > 0$, es estable si $ck \geq h$, y converge con orden $O(h) + O(k)$.

3.6. BTBS

El siguiente esquema que analizamos es el BTBS (Backward in Time, Backward in Space). Es un esquema implícito.

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_j^{n+1} - U_{j-1}^{n+1}}{h} = f(x_j, t_n) \quad \begin{cases} 2 \leq j \leq J, \\ 1 \leq n \leq N - 1. \end{cases} \quad (3.39)$$

Es un esquema consistente, pero no siempre estable ni convergente. Diferenciamos dos casos:

- Si $c(x_j, t_n) < 0$, es estable si $|c|k \geq h$, y converge con orden $O(h) + O(k)$.
- Si $c(x_j, t_n) > 0$, es estable, y converge con orden $O(h) + O(k)$.

3.7. BTCS

El siguiente esquema que analizamos es el BTCS (Backward in Time, Centered in Space). Es un esquema implícito.

$$\frac{U_j^{n+1} - U_j^n}{k} + c(x_j, t_n) \frac{U_{j+1}^{n+1} - U_{j-1}^{n+1}}{2h} = f(x_j, t_n) \quad \begin{cases} 2 \leq j \leq J - 1, \\ 1 \leq n \leq N - 1. \end{cases} \quad (3.40)$$

J, h	N, k	Error relativo	Observaciones
11, 10^{-1}	25, $5 \cdot 10^{-2}$	$5.02249 \cdot 10^{-3}$	Comportamiento estable.
33, $3.125 \cdot 10^{-2}$	241, $5 \cdot 10^{-3}$	$5.31401 \cdot 10^{-4}$	Reducción esperada para el orden teórico.
101, 10^{-2}	2401, $5 \cdot 10^{-4}$	$5.40424 \cdot 10^{-5}$	Reducción esperada para el orden teórico.

Cuadro 3.2: Resultados BTCS

Este esquema converge con orden 2 en espacio y con orden 1 en tiempo.

Ejecutando en MATLAB el programa correspondiente al esquema BTCS, que podemos encontrar en el Anexo a este documento, obtenemos los resultados dados por la anterior tabla, donde en la primera columna podemos ver el número de nodos espaciales I , junto con el paso de discretización espacial h , en la segunda columna tenemos el número de nodos temporales N , junto con el paso de discretización temporal k , en la tercera columna se encuentra el error relativo entre la solución exacta y la solución aproximada por el programa, y en la cuarta y última columna tenemos unas breves observaciones para cada caso.

A continuación, mostramos dos gráficas donde se puede apreciar visualmente el orden, obtenidas trabajando en escala logarítmica. Supongamos que discretizamos en la variable espacial, siendo el orden del método p . Para h suficientemente pequeño, se tiene

$$e = O(h^p) = Ch^p, \quad (3.41)$$

de donde

$$\log(e) = \log(C) + p \cdot \log(h), \quad (3.42)$$

que es una recta de pendiente p , es decir, la pendiente tiende al orden del método a medida que h tiende a 0.

En nuestro caso, como estamos discretizando tanto en la variable espacial como temporal, para ver el orden gráficamente necesitaríamos una superficie, por tanto, lo que haremos será fijar primero el paso espacial $h = 10^{-4}$ e ir calculando los errores relativos para varios pasos de discretización temporal k , lo cual representamos en la Figura (3.1). A continuación, fijamos el paso temporal $k = 5 \cdot 10^{-4}$ y calculamos los errores relativos para unos cuantos pasos de discretización espacial h , tal y como podemos observar en la Figura (3.2).

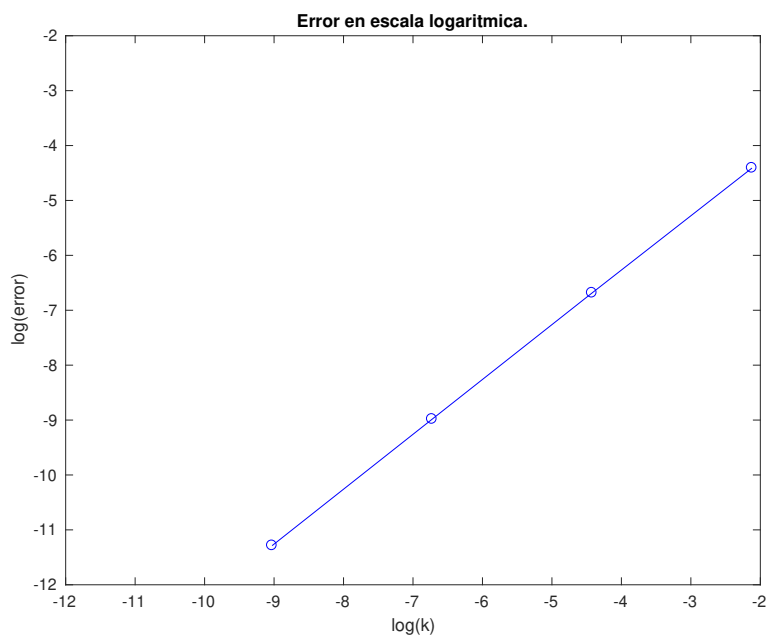


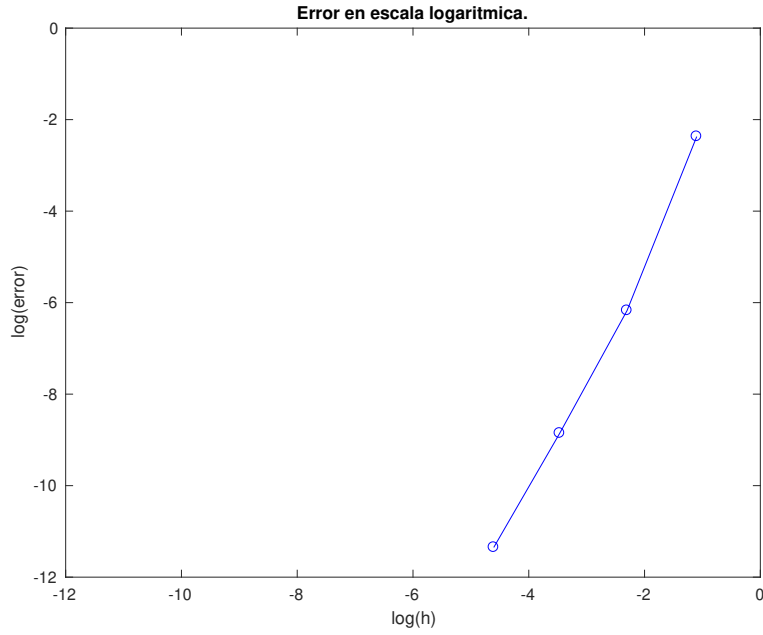
Figura 3.1: Gráfica del error fijando h .

3.8. Método de características.

Para los conceptos y notaciones que se emplean en esta sección puede verse la referencia [2].

Denotamos por X el movimiento para el cual $c(x, t)$ sea la velocidad. Si $u(x, t)$ es un campo escalar espacial, su derivada material con respecto al tiempo se define de la forma

$$\dot{u}(x, t) := \frac{\partial}{\partial t} u(X(p, t), t)|_{p=P(x, t)} \quad (3.43)$$

Figura 3.2: Gráfica del error fijando k .

donde P denota la aplicación de referencia del movimiento X , es decir,

$$p = P(x, t) \Leftrightarrow x = X(p, t).$$

Se puede demostrar que \dot{u} se calcula mediante la fórmula

$$\dot{u}(x, t) = \frac{\partial u}{\partial t}(x, t) + c(x, t) \frac{\partial u}{\partial x}(x, t).$$

Por tanto, la ecuación del transporte

$$\frac{\partial u}{\partial t} + c(x, t) \frac{\partial u}{\partial x} = f, \quad (3.44)$$

se puede reescribir de la forma

$$\dot{u}(x, t) = f(x, t). \quad (3.45)$$

Ahora, la definición (3.43) sugiere utilizar la siguiente discretización descentrada *aguas arribade* \dot{u} :

$$\dot{u}(x, t_{n+1}) = \frac{\partial}{\partial t} u(X(p, t), t)|_{p=P(x, t_{n+1})} \approx \frac{u(X(p, t_{n+1}), t_{n+1}) - u(X(p, t_n), t_n)}{k}. \quad (3.46)$$

(Recordemos que k denota el paso temporal de la discretización).

Ahora hay que escribir esta aproximación en coordenada espacial x . La posición que ocupa en el instante t_{n+1} la partícula que en el instante t_n ocupa la posición x es claramente X , es decir,

$$X(P(x, t_{n+1}), t_{n+1}) = x \quad (3.47)$$

ya que $X(., t)$ y $P(., t)$ son inversas; por ello, el último término de la ecuación (3.46) es igual a

$$\frac{u(x, t_{n+1}) - u(X(P(x, t_{n+1}), t_n), t_n)}{k}, \quad (3.48)$$

mientras que el término

$$X(P(x, t_{n+1}), t_n) \quad (3.49)$$

representa la posición del punto material $P(x, t_{n+1})$, k unidades de tiempo antes. Todo esto sugiere el esquema de discretización para la ecuación del transporte,

$$\frac{U^{n+1}(x) - U^n(X(P(x, t_{n+1}), t_n))}{k} = f^{n+1}(x). \quad (3.50)$$

La dificultad del método de características está en calcular

$$u(X(P(x, t_{n+1}), t_n), t_n). \quad (3.51)$$

Para ello, recordamos en primer lugar que, por definición, se tiene

$$c(x, t) = (\dot{X})_s(x, t) := \frac{\partial X}{\partial t}(p, t)|_{p=P(x, t)}$$

, por lo que

$$\frac{\partial X}{\partial t}(P(x, t), t) = c(x, t). \quad (3.52)$$

Por consiguiente, para calcular $u(X(P(x, t_{n+1}), t_n), t_n)$ debe resolverse un problema de Cauchy para una ecuación diferencial ordinaria (EDO), concretamente,

$$\begin{cases} \frac{\partial X}{\partial t}(p, t) = c(X(p, t), t), \\ X(p, t_{n+1}) = x. \end{cases} \quad (3.53)$$

En general, esta ecuación es no lineal, y se puede resolver, entre otros, mediante los métodos de Euler, Runge-Kutta, etc. Por ejemplo, utilizando el método de Euler implícito, se tiene

$$\frac{X^{n+1}(p) - X^n(p)}{k} = c(X^{n+1}(p), t_{n+1}) \quad (3.54)$$

con $X^{n+1}(p) = X(p, t_{n+1}) = x$, es decir,

$$X^n(p) = x - kc(x, t_{n+1}).$$

Capítulo 4

Anexo.

4.1. Programación esquema BTCS MATLAB.

```
1 % BTCS Scheme
2
3 tic
4
5 clc;
6 clear all;
7 close all;
8 ofi = fopen('output.txt','wt');
9
10 data
11
12 fprintf(ofi,'\n\n\n**** COMPUTATION ****');
13 [x,t,U] = BTCS(a,b,iT,fT,I,N,ofi);
14
15 displayer
16
17 fclose(ofi);
18
19 plotter
20
21 showmovie
22
23 toc
```

```

1  % DATA
2
3  fprintf(ofi, '***** DATA *****');
4
5  % LEFT SPATIAL EXTREMUM:
6  a = 0;
7  fprintf(ofi, '\n\n* Left spatial extremum: a = %-.15E.', a);
8
9  % RIGHT SPATIAL EXTREMUM:
10 b = 1;
11 fprintf(ofi, '\n\n* Right spatial extremum: b = %-.15E.', b);
12
13 % INITIAL TIME:
14 iT = 0;
15 fprintf(ofi, '\n\n* Initial time: iT = %-.15E.', iT);
16
17 % FINAL TIME:
18 fT = 1.2;
19 fprintf(ofi, '\n\n* Final time: fT = %-.15E.', fT);
20
21 % NUMBER OF POINTS (or NODES) OF THE SPATIAL MESH
22 % (= NUMBER OF SUBINTERVALS + 1):
23 I = 101; % x(1) = a < x(2) < ... < x(I - 1) < x(I) = b.
24 fprintf(ofi, '\n\n* Number of nodes of the spatial mesh: I = %-i.', I);
25
26 % NUMBER OF TIME STEPS (THE FIRST ONE, t(1) = iT, COUNTS):
27 N = 2401; % t(1) = iT < t(2) < ... < t(N - 1) < t(N) = fT.
28 fprintf(ofi, '\n\n* Number of time steps: N = %-i.', N);
29
30 % DISPLAY FREQUENCY IN x (the results are displayed only for
31 % the first node, the last node, and when the number of the
32 % node is a multiple of dfreqx):
33 dfreqx = max(1, round(I/5));
34 fprintf(ofi, '\n\n* Display frequency in x: dfreqx = %-i ', dfreqx);
35 fprintf(ofi, '\n(in order not to display too many lines, the results are');
36 fprintf(ofi, '\ndisplayed only for the first node, the last node, and');
37 fprintf(ofi, '\nwhen the number of the node is a multiple of dfreqx).');
38
39 % DISPLAY FREQUENCY IN t (the results are displayed only for
40 % the initial time, the final time, and when the index of the
41 % time step is a multiple of dfreqt):

```

```

42 dfreqt = max(1,round(N/10));
43 fprintf(ofi,'\n\n* Display frequency in t: dfreqt = %-i ', dfreqt);
44 fprintf(ofi,'\n(in order not to display too many lines, the results are');
45 fprintf(ofi,'\ndisplayed only for the initial time, the final time, and');
46 fprintf(ofi,'\nwhen the index of the time step is a multiple of dfreqt).');
47
48 % PLOT FREQUENCY IN x (the results are plotted only for the first node,
49 % the last node, and when the number of the node is a multiple of plfreqx):
50 plfreqx = max(1,round(I/50));
51 fprintf(ofi,'\n\n* Plot frequency in x: plfreqx = %-i ', plfreqx);
52 fprintf(ofi,'\n(in order not to plot too many points, the results are');
53 fprintf(ofi,'\nplotted only for the first node, the last node, and');
54 fprintf(ofi,'\nwhen the number of the node is a multiple of plfreqx).');
55
56 % PLOT FREQUENCY IN t (the results are plotted only for
57 % the initial time, the final time, and when the index of
58 % the time step is a multiple of plfreqt):
59 plfreqt = max(1,round(N/50));
60 fprintf(ofi,'\n\n* Plot frequency in t: plfreqt = %-i ', plfreqt);
61 fprintf(ofi,'\n(in order not to plot too many points, the results are');
62 fprintf(ofi,'\nplotted only for the initial time, the final time, and');
63 fprintf(ofi,'\nwhen the index of the time step is a multiple of plfreqt).');
64
65 % IS THE EXACT SOLUTION KNOWN (ex = 1) OR NOT (otherwise)?
66 ex = 1;
67 fprintf(ofi,'\n\n* Option for exact solution: ex = %-i', ex);
68 if (ex == 1)
69     fprintf(ofi,' (the exact solution is known).');
70 else
71     fprintf(ofi,' (the exact solution is not known).');
72 end

1 % Numerical resolution
2 %
3 %  $u_t + c(x,t)u_x = f(x,t)$ 
4 % for  $a \leq x \leq b$ ,  $iT \leq t \leq fT$ 
5 % [iT: initial time, often iT = 0; fT: final time],

```

```

6 %
7 % u(x,iT) = u0(x) [INITIAL CONDITION] for a <= x <= b,
8 %
9 % u(a,t) = alpha(t) [BOUNDARY CONDITION AT x = a] for iT <= t <= fT, if
10 % c(x,t)>0, for all a <= x <= b, iT <= t <= fT,
11 % u(b,t) = beta(t) [BOUNDARY CONDITION AT x = b] for iT <= t <= fT, if
12 % c(x,t)<0, for all a <= x <= b, iT <= t <= fT.
13 %
14 % By using the explicit scheme with I spatial nodes
15 % x(1) = a, x(2),..., x(I - 1), x(I) = b and N time steps
16 % t(1) = iT < t(2) < ... < t(N - 1) < t(N) = fT.
17 %
18 % x is the vector of spatial nodes, and
19 % t is the vector of time steps.
20 %
21 % U is the matrix of approximations: U(i,n) is the approximation of
22 % u(x(i),t(n)), being u(x,t) the exact solution. In other
23 % words, the column n of U contains the approximations of u(.,t(n))
24 % at the spatial nodes.
25 %
26 % U is a rectangular matrix of dimensions I x N.
27 %
28 function [x,t,U] = BTCS(a,b,iT,fT,I,N,ofi)
29
30 % h is the spatial grid spacing (h = Delta x):
31 h = (b-a)/(I-1);
32 fprintf(ofi,'\n\nSpatial grid spacing: h = (b-a)/(I-1) = %-.15E.', h);
33
34 % k is the time grid spacing (k = Delta t):
35 k = (fT-iT)/(N-1);
36 fprintf(ofi,'\n\nTime grid spacing: k = (fT-iT)/(N-1) = %-.15E.', k);
37
38 % x is the vector of spatial nodes:
39 x = linspace(a,b,I);
40
41 % t is the vector of time steps:
42 t = linspace(iT,fT,N);
43
44 %Stability
45 [X,T] = meshgrid(x,t);
46 CC = c(X,T);

```

```

47
48 cmin=min(min(CC));
49 cmax=max(max(CC));
50 if (cmin>=0)
51     SIGNO=1;
52 elseif (cmax<=0)
53     SIGNO=-1;
54 else
55     error('Function c (in file c.m) must have constant sign')
56 end
57
58 speed = max(max(abs(CC)));
59 fprintf(ofi, '\n\n|speed|*k = %-.6E.', speed*k);
60 fprintf(ofi, '\nh = %-.6E.', h);
61 if (speed*k > h)
62     fprintf(ofi, '\n\nDANGER!!! INSTABILITIES MAY OCCUR!!!');
63 end
64
65 % The following sentence makes the program run faster, since
66 % it fixes the size of the matrix U instead of increasing its
67 % size at every step.
68 U = zeros(I,N);
69
70 % The first column of U is given by the initial condition:
71 U(:,1) = u0(x); % IC
72
73 if SIGNO==1
74     U(1,2:N) = alpha(t(2:N)); % BC at x = a.
75     for n = 1:N-1
76         tnp1=t(n+1);
77         U(2:I,n+1) = BTCScomp1(k,h,tnp1,U(2:I,n),x(2:I));
78     end
79 elseif SIGNO==-1
80     U(I,2:N) = beta(t(2:N)); % BC at x = b.
81     for n = 1:N-1
82         tnp1=t(n+1);
83         U(1:I-1,n+1) = BTCScomp2(k,h,tnp1,U(1:I-1,n),x(1:I-1));
84     end
85 end
86
87 end

```

```

1 % BTCScomp1
2
3 function Unp1 = BTCScomp1(k,h,tnp1,U2,xx)
4
5 % Unp1 stands for "U^{n plus 1} = U^{n+1}".
6
7 % IT IS A COLUMN VECTOR.
8
9 C = c(xx,tnp1).';
10 L = length(xx);
11 d = sparse(1:L,1:L,1,L,L); % diagonal
12 ld = sparse(2:L,1:L-1,-(k/(2*h)).*C(2:L),L,L); % lower diagonal
13 ud = sparse(1:L-1,2:L,(k/(2*h)).*C(1:L-1),L,L); % upper diagonal
14 M = d + ld + ud; % matrix for linear system
15 M(L,L-2)=1;
16 M(L,L-1)=-2;
17 sm = U2 + k*f(xx,tnp1).';
18 sm(1) = sm(1)+(k/(2*h))*C(1)*alpha(tnp1);
19 sm(L) = 0;
20 Unp1 = M\sm;
21
22 end

```

```

1 % BTCScomp2
2
3 function Unp1 = BTCScomp2(k,h,tnp1,U2,xx)
4
5 % Unp1 stands for "U^{n plus 1} = U^{n+1}".
6
7 % IT IS A COLUMN VECTOR.
8
9 C = c(xx,tnp1).';

```

```

10 L = length(xx);
11 d = sparse(1:L,1:L,1,L,L); % diagonal
12 ld = sparse(2:L,1:L-1,-(k/(2*h)).*C(2:L),L,L); % lower diagonal
13 ud = sparse(1:L-1,2:L,(k/(2*h)).*C(1:L-1),L,L); % upper diagonal
14 M = d + ld + ud; % matrix for linear system
15 M(1,3)=1;
16 M(1,2)=-2;
17 sm = U2 + k*f(xx,tnp1).';
18 sm(L) = sm(L) - (k/(2*h))*C(L)*beta(tnp1);
19 sm(1) = 0;
20 Unp1 = M\sm;
21
22 end

```

```

1 % Function c(x,t)
2
3 function vc = c(x,t)
4
5 vc = -x.*t;
6
7 end

```

```

1 % Function f(x,t)
2
3 function vf = f(x,t)
4
5 vf=(1-t^2).*x.*cos(x*t);
6
7 end

```

```
1 % Function defining the initial condition, i. e.,
2 % if iT is the initial time (often, iT = 0), then
3 % u(x,iT) = u0(x) for all x.
4
5 function vu0 = u0(x)
6
7 vu0 = 0.*x;
8
9 end
```

```
1 % Function defining the boundary condition at x = a,
2 % i. e., u(a,t) = alpha(t) for all t.
3
4 function valpha = alpha(t)
5
6 valpha = 0.*t;
7
8 end
```

```
1 % Function defining the boundary condition at x = b,
2 % i. e., u(b,t) = beta(t) for all t.
3
4 function vbeta = beta(t)
5
6 vbeta = sin(t);
7
8 end
```

```
1 % Exact solution
2
```

```

3 function vexact = exact(x,t)
4
5 vexact = sin(x.*t);
6
7 end

1 % Displayer
2
3 % dfreqx: display frequency in x (in order not to display too many lines,
4 % the results are displayed only for the first node, the last node, and
5 % when the number of the node is a multiple of dfreqx).
6 %
7 % dfreqt: display frequency in t (in order not to display too many lines,
8 % the results are displayed only for the initial time, the final time, and
9 % when the index of the time step is a multiple of dfreqt).
10 %
11 % ex: it indicates whether the exact solution
12 % is known (ex=1) or not (otherwise).
13 %
14 % of: it is a file identifier for the output file.
15
16 indexes = 1:N;
17 idxt = find(indexes == 1 | mod(indexes,dfreqt) == 0 | indexes == N);
18 tdisp = t(idxt);
19
20 indexes = 1:I;
21 idxx = find(indexes == 1 | mod(indexes,dfreqx) == 0 | indexes == I);
22 xdisp = x(idxx);
23
24 Udisp = U(idxx,idxt);
25
26 dimt = length(tdisp);
27 dimx = length(xdisp);
28
29 if (ex == 1) % EXACT SOLUTION IS KNOWN
30     [X,T] = meshgrid(x,t);
31     Uex = exact(X,T).'; % COMPLETE MATRIX OF EXACT VALUES

```

```

32     % Component (j1,j2) of exact (X,T) is exact (x(j2),t(j1)).
33     % That's why we transpose exact (X,T).
34
35     Uexdisp = Uex(idxx,idxt);
36     ERR = Uex - U; % COMPLETE MATRIX OF ERRORS
37     ERRdisp = ERR(idxx,idxt);
38     maxerr = max(max(abs(ERR)));
39     maxrelerr = norm(ERR)/norm(Uex);
40     fprintf(ofi,'\n\n\nMAXIMUM ERROR (in absolute value) = %-.5E.',...
41         maxerr);
42     fprintf(ofi,'\n\nMAXIMUM RELATIVE ERROR = %-.5E.',...
43         maxrelerr);
44     [ierrx,ierrt] = find(abs(ERR) == maxerr);
45     fprintf(ofi,'\n\nThis maximum error is found at time %-.5E',t(ierrt(1)));
46     fprintf(ofi,'\n\nand spatial node %-.5E.',x(ierrx(1)));
47
48     if (length(ierrx) > 1 || length(ierrt) > 1)
49         fprintf(ofi,'\n\nThere are other pairs (x_i,t_n) where the same');
50         fprintf(ofi,' error is attained. \nSee file displayer.m.');
```

51 end

52

```

53     for n = 1:dimt
54         % TIME = tdisp(n)
55         step = idxt(n);
56         time = tdisp(n);
57         fprintf(ofi,'\n\n###Step %-i, time = %-.5E:\n',step,time);
58         fprintf(ofi,'\n\n[SPATIAL NODE], EXACT: ');
59         for i = 1:dimx
60             fprintf(ofi,[' %-.5E], %-.5E\t\t',xdisp(i),Uexdisp(i,n));
61         end
62         fprintf(ofi,'\n\n[SPATIAL NODE], APPRO: ');
63         for i = 1:dimx
64             fprintf(ofi,[' %-.5E], %-.5E\t\t',xdisp(i),Udisp(i,n));
65         end
66         fprintf(ofi,'\n\n[SPATIAL NODE], ERROR: ');
67         for i = 1:dimx
68             fprintf(ofi,[' %-.5E], %-.5E\t\t',xdisp(i),ERRdisp(i,n));
69         end
70     end
71 else % EXACT SOLUTION IS NOT KNOWN
72     for n = 1:dimt
```

```

73     % TIME = tdisp(n)
74     step = idxt(n);
75     time = tdisp(n);
76     fprintf(ofi, '\n\n####Step %-i, time = %-.5E:\n', step, time);
77     fprintf(ofi, '\n[SPATIAL NODE], APPROXIMATE: ');
78     for i = 1:dimx
79         fprintf(ofi, '[%-.5E], %-.5E\t\t', xdisp(i), Udisp(i, n));
80     end
81 end
82 end

1 % Plotter
2
3 % plfreqx: plot frequency in x (in order not to plot too many lines,
4 % the results are plotted only for the first node, the last node, and
5 % when the number of the node is a multiple of plfreqx).
6 %
7 % plfreet: plot frequency in t (in order not to plot too many lines,
8 % the results are plotted only for the initial time, the final time, and
9 % when the index of the time step is a multiple of plfreet).
10 %
11 % ex: it indicates whether the exact solution
12 % is known (ex=1) or not (otherwise).
13
14 indexes = 1:N;
15 idxt = find(indexes == 1 | mod(indexes,plfreet) == 0 | indexes == N);
16 tpl = t(idxt);
17
18 indexes = 1:I;
19 idxx = find(indexes == 1 | mod(indexes,plfreqx) == 0 | indexes == I);
20 xpl = x(idxx);
21
22 Upl = U(idxx, idxt);
23
24 mesh(xpl, tpl, Upl.')
25 % surf(xpl, tpl, Upl.')
26 colorbar

```

```

27 title('APPROXIMATE SOLUTION');
28
29 if (ex == 1)
30
31     [X,T] = meshgrid(x,t);
32     Uex = exact(X,T)'; % COMPLETE MATRIX OF EXACT VALUES
33     % Component (j1,j2) of exact(X,T) is exact(x(j2),t(j1)).
34     % That's why we transpose exact(X,T).
35     Uexpl = Uex(idxx,idxt);
36     figure
37     mesh(xpl,tpl,Uexpl.')
38     colorbar
39     title('EXACT SOLUTION');ERR = Uex - U; % COMPLETE MATRIX OF ERRORS
40
41     ERRpl = ERR(idxx,idxt);
42     figure
43     mesh(xpl,tpl,ERRpl.')
44     colorbar
45     title('ERROR');
46
47 end

1 % Showmovie
2
3 indexes = 1:N;
4 idxt = find(indexes == 1 | mod(indexes,plfreqt) == 0 | indexes == N);
5 tpl = t(idxt);
6 last = length(tpl);
7
8 indexes = 1:I;
9 idxx = find(indexes == 1 | mod(indexes,plfreqx) == 0 | indexes == I);
10 xpl = x(idxx);
11
12 Upl = U(idxx,idxt);
13
14 MI = min(min(Upl));
15 MA = max(max(Upl));

```

```

16
17 figure
18
19 for n = 1:last
20     plot(xpl,Upl(:,n),'o-')
21     axis ([x(1) x(I) MI MA])
22     getframe;
23     pause(0.02)
24 end
25
26 legend('Temperature in K')

```

4.2. Programación esquema FTFS MATLAB.

```

1 % FTFS Scheme
2
3 tic
4
5 clc;
6 clear all;
7 close all;
8 ofi = fopen('output.txt','wt');
9
10 data
11
12 fprintf(ofi,'\n\n\n**** COMPUTATION ****');
13 [x,t,U] = FTFS(a,b,iT,fT,I,N,ofi);
14
15 displayer
16
17 fclose(ofi);
18
19 plotter
20
21 showmovie
22
23 toc

```

```

1 % Numerical resolution
2 %
3 %  $u_t + c(x,t)u_x = f(x,t)$ 
4 % for  $a \leq x \leq b$ ,  $iT \leq t \leq fT$ 
5 % [ $iT$ : initial time, often  $iT = 0$ ;  $fT$ : final time],
6 %
7 %  $u(x, iT) = u_0(x)$  [INITIAL CONDITION] for  $a \leq x \leq b$ ,
8 %
9 %  $u(a, t) = \alpha(t)$  [BOUNDARY CONDITION AT  $x = a$ ] for  $iT \leq t \leq fT$ , if
10 %  $c(x,t) > 0$ , for all  $a \leq x \leq b$ ,  $iT \leq t \leq fT$ ,
11 %  $u(b, t) = \beta(t)$  [BOUNDARY CONDITION AT  $x = b$ ] for  $iT \leq t \leq fT$ , if
12 %  $c(x,t) < 0$ , for all  $a \leq x \leq b$ ,  $iT \leq t \leq fT$ .
13 %
14 % By using the explicit scheme with  $I$  spatial nodes
15 %  $x(1) = a$ ,  $x(2), \dots, x(I - 1)$ ,  $x(I) = b$  and  $N$  time steps
16 %  $t(1) = iT < t(2) < \dots < t(N - 1) < t(N) = fT$ .
17 %
18 %  $x$  is the vector of spatial nodes, and
19 %  $t$  is the vector of time steps.
20 %
21 %  $U$  is the matrix of approximations:  $U(i,n)$  is the approximation of
22 %  $u(x(i), t(n))$ , being  $u(x,t)$  the exact solution. In other
23 % words, the column  $n$  of  $U$  contains the approximations of  $u(., t(n))$ 
24 % at the spatial nodes.
25 %
26 %  $U$  is a rectangular matrix of dimensions  $I \times N$ .
27 %
28 function [x,t,U] = FTFS(a,b,iT,fT,I,N,ofi)
29
30 %  $h$  is the spatial grid spacing ( $h = \Delta x$ ):
31 h = (b-a)/(I-1);
32 fprintf(ofi, '\n\nSpatial grid spacing: h = (b-a)/(I-1) = %-.15E.', h);
33
34 %  $k$  is the time grid spacing ( $k = \Delta t$ ):
35 k = (fT-iT)/(N-1);
36 fprintf(ofi, '\n\nTime grid spacing: k = (fT-iT)/(N-1) = %-.15E.', k);
37
38 %  $x$  is the vector of spatial nodes:
39 x = linspace(a,b,I);
40
41 %  $t$  is the vector of time steps:

```

```

42 t = linspace(iT,fT,N);
43
44 %Stability
45 [X,T] = meshgrid(x,t);
46 CC = c(X,T);
47
48 cmax=max(max(CC));
49 if (cmax<=0)
50     SIGNO=-1;
51 else
52     error('Function c (in file c.m) must have constant (negative) sign')
53 end
54
55 speed = max(max(abs(CC)));
56 fprintf(ofi,'\n\n|speed|*k = %-.6E.', speed*k);
57 fprintf(ofi,'\nh = %-.6E.', h);
58 if (speed*k > h)
59     fprintf(ofi,'\n\nDANGER!!! INSTABILITIES MAY OCCUR!!!');
60 end
61
62 % The following sentence makes the program run faster, since
63 % it fixes the size of the matrix U instead of increasing its
64 % size at every step.
65 U = zeros(I,N);
66
67 % The first column of U is given by the initial condition:
68 U(:,1) = u0(x); % IC
69 U(I,2:N) = beta(t(2:N)); % BC at x = b.
70
71 if SIGNO== -1
72     for n = 1:N-1
73         tn = t(n);
74         U1=U(1:I-1,n);
75         U2=U(2:I,n);
76         U(1:I-1,n+1) = FTFScomp(k,h,tn,U1,U2,x(1:I-1));
77     end
78 end
79
80 end

```

```
1 function Unp1 = FTFScomp(k,h,tn,U1,U2,xx)
2
3 % Unp1 stands for "U^{n plus 1} = U^{n+1}".
4
5 % IT IS A COLUMN VECTOR.
6
7 C = c(xx,tn).';
8 A = C.*U1-C.*U2;
9 Unp1 = U1 + (k/h).*A + k.*f(xx,tn).';
10
11 end
```

Bibliografía

- [1] Evans, G., Blackledge, J. and Yardley, P., *Numerical Methods for Partial Differential Equations*, Springer Undergraduate Mathematics Series, Springer-Verlag, London, 2000.
- [2] Gurtin, Morton E., *An introduction to Continuum Mechanics*, Academic Press, New York, 1981.
- [3] LeVeque, Randall J., *Finite Difference Methods for Ordinary and Partial Differential Equations*, Society for Industrial and Applied Mathematics, Seattle, 2007.
- [4] Morton, K. W. and Mayers, D.F., *Numerical Solution of Partial Differential Equations*, 2nd ed., Cambridge University Press, 2005.
- [5] Strikwerda, John C., *Finite Difference Schemes and Partial Differential Equations*, Chapman & Hall, New York, 1989.
- [6] Thomas, J. W., *Numerical Partial Differential Equations: Finite Difference Methods*, Texts in Applied Mathematics, Springer-Verlag, New York, 1995.
- [7] Uwe Lelke, *m-code to LaTeX converter*, <https://www.mathworks.com/matlabcentral/fileexchange/24515-m-code-to-latex-converter>, MATLAB Central File Exchange, MathWorks, 2020.
- [8] Vichnevetsky, Robert and Bowles, John B., *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*, SIAM Studies in Applied Mathematics, Philadelphia, 1982.
- [9] Zuazua, Enrique, *Métodos numéricos de resolución de Ecuaciones en Derivadas Parciales*, http://www.bcamath.org/documentos_public/archivos/personal/comites/1_notas-05_065-complete.pdf, BCAM, Bilbao, 2009.