



 UNIVERSIDADE DA CORUÑA



Universidade Vigo



Universidad  
Carlos III de Madrid



POLITÉCNICA

## MÁSTER EN MATEMÁTICA INDUSTRIAL

AMPLIACIÓN DE VOLÚMENES FINITOS

---

# Proyecto: Sistemas Lineales

---

ALBERTO CUADRA LARA

*Proyecto presentado en cumplimiento de los requisitos de la asignatura  
Ampliación de Volúmenes Finitos curso 2017-2018 para el Máster en  
Matemática Industrial*

15 de mayo de 2018

# Índice general

<b>1. Sistemas lineales</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Descripción del problema . . . . .	1
1.2.1. Sistema de leyes de conservación . . . . .	1
Sistema hiperbólico . . . . .	2
1.2.2. Problema de Cauchy . . . . .	3
1.2.3. Problema de Riemann . . . . .	4
Caso de estudio . . . . .	5
1.2.4. Método de Godunov . . . . .	6
Observación . . . . .	7
Análisis de estabilidad de von Neumann y la condición CFL . . . . .	7
1.3. Resultados . . . . .	8
1.3.1. Parámetros del problema . . . . .	8
Caso 1 . . . . .	8
1.3.2. Condiciones iniciales . . . . .	8
Problema de Riemann . . . . .	9
Problema de Cauchy . . . . .	9
1.3.3. Análisis . . . . .	9
1.4. Conclusiones y trabajo futuro . . . . .	13
<b>A. Código utilizado en MATLAB</b>	<b>14</b>
<b>Bibliografía</b>	<b>20</b>

# Índice de figuras

1.1. Solución del sistema lineal hiperbólico <i>One-Dimensional Linearized Shallow water equations</i> asociado al problema de Riemann: exacta y mediante el método de Godunov. Rango: $x \in [-3, 3] \text{ m}$ , $t \in [0, 0,3] \text{ s}$ . <i>Courant</i> : (a) $\mu = 0,8$ , (b) $\mu = 1,01$ .	10
1.2. Solución del sistema lineal hiperbólico <i>One-Dimensional Linearized Shallow water equations</i> asociado al problema de Cauchy: exacta y mediante el método de Godunov. Rango: $x \in [-3, 3] \text{ m}$ , $t \in [0, 0,3] \text{ s}$ . <i>Courant</i> : (a) $\mu = 0,8$ , (b) $\mu = 1,01$ .	11
1.3. Solución del sistema lineal hiperbólico <i>One-Dimensional Linearized Shallow water equations</i> asociado al problema de Cauchy: exacta y mediante el método de Godunov. Rango: $x \in [-3, 3] \text{ m}$ , $t \in [0, 0,3] \text{ s}$ . <i>Courant</i> : (a) $\mu = 0,8$ , (b) $\mu = 1,5$ .	12

# Proyecto 1

## Sistemas lineales

### 1.1. Introducción

En el presente proyecto se va a resolver un sistema lineal de ecuaciones unidimensional a partir del método de volúmenes finitos empleando el método de Godunov bajo dos tipo de condiciones iniciales distintas, asociadas al problema de Riemann y al problema de Cauchy, respectivamente. Se comparan los resultados con las soluciones exactas. El sistema de prueba escogido corresponde con el modelo linealizado de las aguas someras unidimensionales, *One-Dimensional Linearized Shallow Water equations*.

El código ha sido diseñado en Matlab R2018a y se adjunta de forma paralela al documento, así mismo, se ha incluido éste en el anexo. Adicionalmente, se ha realizado una aplicación tipo *GUI, Graphic User Interface*, en Matlab, bajo el nombre *Finite Volume Methods - project (FVM\_project)*. El objetivo es poder manipular con facilidad los parámetros del código generado, convirtiéndolo en una locuaz herramienta con pretensiones didácticas.

### 1.2. Descripción del problema

El sistema correspondiente al modelo linealizado de las aguas someras unidimensionales se expresa como

$$\begin{cases} \frac{\partial h}{\partial t} + \tilde{u} \frac{\partial h}{\partial x} + \tilde{h} \frac{\partial u}{\partial x} = 0, \\ \frac{\partial u}{\partial t} + g \frac{\partial h}{\partial x} + \tilde{u} \frac{\partial u}{\partial x} = 0, \end{cases} \quad (1.1)$$

donde  $\tilde{h}$  y  $\tilde{u}$ , son constantes y representan las componentes linealizadas de la altura del agua  $h(x, t)$  y de la velocidad horizontal  $u(x, t)$ , respectivamente.  $g = 9,8 \text{ m/s}^2$  y representa la aceleración de la gravedad.

#### 1.2.1. Sistema de leyes de conservación

Sea  $\Omega \in \mathbb{R}^p$  el dominio del problema y  $f$  una función regular de  $\Omega$ :

$$f : \Omega \rightarrow \mathbb{R}^p. \quad (1.2)$$

Aplicando la definición dada por los autores E. Godlewski y P. A. Raviart en [3] la forma general de un sistema de leyes de conservación es

$$w_t + (f(w))_x = 0, \quad x \in \mathbb{R}, \quad t > 0, \quad (1.3)$$

donde  $w = (w_1, \dots, w_p)$  es el vector de las variables conservativas y  $f = (f_1, \dots, f_p)$  es el flujo físico [1].

Igualando los términos del sistema 1.1 y 1.3 se obtiene

$$\left. \begin{array}{l} w_1 = h \\ w_2 = u \end{array} \right\} \rightarrow w = (w_1, w_2)^T = (h, u)^T, \quad (1.4)$$

$$\frac{\partial f_1(w)}{\partial x} = \tilde{u} \frac{\partial h}{\partial x} + \tilde{h} \frac{\partial u}{\partial x}, \quad (1.5)$$

integrando sobre todo el dominio  $\Omega$

$$\int_{\Omega} \frac{\partial f_1(w)}{\partial x} = \int_{\Omega} \tilde{u} \frac{\partial h}{\partial x} + \int_{\Omega} \tilde{h} \frac{\partial u}{\partial x}, \quad (1.6)$$

$$f_1(w) = \tilde{u}h + \tilde{h}u. \quad (1.7)$$

Análogamente

$$f_2(w) = gh + \tilde{u}u. \quad (1.8)$$

Por tanto,

$$f(w) = (f_1(w), f_2(w))^T = (\tilde{u}h + \tilde{h}u, gh + \tilde{u}u). \quad (1.9)$$

### Sistema hiperbólico

Por definición, en el caso unidimensional, el sistema 1.1 se dice hiperbólico si la matriz  $A(w)$  tiene 2 autovalores reales  $\lambda_1(w) \leq \lambda_2(w)$  y los 2 autovectores correspondientes son linealmente independientes<sup>1</sup>.  $A(w)$  es la matriz Jacobiana asociada a la función de flujo  $f(w)$  y por definición es

$$A(w) = \frac{\partial f(w)}{\partial w}. \quad (1.10)$$

La matriz Jacobiana  $A$  asociada a la función de flujo  $f(w)$  es por tanto

$$A(w) = \frac{\partial f(w)}{\partial w} = \begin{pmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} \end{pmatrix} = \begin{pmatrix} \tilde{u} & \tilde{h} \\ g & \tilde{u} \end{pmatrix}. \quad (1.11)$$

A continuación se van a obtener los autovalores de la matriz  $A$ .

$$\det(A - \lambda I) = 0 \rightarrow (\tilde{u} - \lambda)^2 - g\tilde{h} = 0, \quad (1.12)$$

$$\lambda = \tilde{u} \pm \sqrt{g\tilde{h}}. \quad (1.13)$$

El sistema es hiperbólico para  $\tilde{h} \geq 0$  y estrictamente hiperbólico para  $\tilde{h} > 0$ . Nótese, que no es necesario imponer una condición sobre  $\tilde{u}$ .

---

<sup>1</sup>En caso de que ambos autovalores sean distintos se tendría un sistema estrictamente hiperbólico.

Adicionalmente, se van a obtener los autovectores asociados a cada autovalor:

$$(A - \lambda I)e = 0. \quad (1.14)$$

Para  $\lambda_1 = \tilde{u} + \sqrt{g\tilde{h}}$ :

$$\begin{pmatrix} -\sqrt{g\tilde{h}} & \tilde{h} \\ g & -\sqrt{g\tilde{h}} \end{pmatrix} \begin{pmatrix} e_{1,1} \\ e_{1,2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (1.15)$$

obteniendo que el autovector  $e_1$  asociado a  $\lambda_1$  es

$$e_1 = \left( \frac{\sqrt{g\tilde{h}}}{g}, 1 \right)^T. \quad (1.16)$$

Del mismo modo, se obtiene

$$e_2 = \left( -\frac{\sqrt{g\tilde{h}}}{g}, 1 \right)^T. \quad (1.17)$$

De las expresiones anteriores se tiene la siguiente matriz de paso  $P$

$$P = \begin{pmatrix} \sqrt{g\tilde{h}} & -\sqrt{g\tilde{h}} \\ 1 & 1 \end{pmatrix}, \quad (1.18)$$

donde las columnas corresponden con los autovectores obtenidos.

### 1.2.2. Problema de Cauchy

A continuación se expone como puede hallarse la solución de un problema de Cauchy asociado a la Ec. 1.3, con condiciones iniciales  $w(x, 0) = w_0(x)$ , conociendo la solución de la ecuación de transporte.

Sea  $u = (u_1, \dots, u_p)^T$  el vector de variables características, y teniendo en cuenta la matriz de paso  $P$  definida anteriormente, se tiene

$$w = Pu, \quad (1.19)$$

siendo  $w$  la solución del sistema lineal hiperbólico. Derivando la Ec. 1.19 con respecto al tiempo  $y$  el espacio  $y$  y sustituyendo en 1.1 se obtienen  $p$  ecuaciones de transporte, tales que

$$(u_k)_t + \lambda_k(u_k)_x = 0, \quad k = 1, \dots, p. \quad (1.20)$$

La solución de la ecuación del transporte por el método de las características es

$$u_k(x, t) = u_{k,0}(x - \lambda_k t), \quad k = 1, \dots, p, \quad (1.21)$$

donde las componentes de la condición inicial  $u_{k,0}$  de las variables características vienen dadas por la expresión

$$u_0(x) = P^{-1}w_0(x). \quad (1.22)$$

Por último, deshaciendo el cambio de variable, se obtiene que la solución es

$$w(x, t) = P u(x, t). \quad (1.23)$$

En resumen, el procedimiento para resolver el problema de Cauchy asociado a un sistema lineal de ecuaciones de carácter hiperbólico es:

1. Obtener la matriz  $A$ , sus autovalores, autovectores y su matriz de paso  $P$ .
2. Calcular  $w_0(x)$  para la discretización espacial utilizada<sup>2</sup>.
3. Calcular  $u_0(x)$  dada por la Ec.1.22.
4. Calcular  $u_k(x, t)$  para cada  $k$  dada por la Ec.1.21, obteniendo el vector  $u(x, t)$ .
5. Obtener la solución  $w(x, t)$  dada por la Ec.1.23.

### 1.2.3. Problema de Riemann

El problema de Riemann es un problema de Cauchy donde las condiciones iniciales son discontinuas y vienen dadas de la forma siguiente:

$$w_0(x) = \begin{cases} w_l & x < 0, \\ w_r & x > 0, \end{cases} \quad (1.24)$$

donde  $w_l$  y  $w_r$  son dos constantes. En relación al problema de estudio, estas condiciones iniciales modelan la rotura de una presa.

Atendiendo a la Ec. 1.23, la solución  $w$  puede expresarse como

$$w = \sum_{k=1}^p \alpha_k e_k, \quad (1.25)$$

donde  $e_k$  con  $k = 1, \dots, p$  son los autovectores de  $A$  que conforman las columnas de la matriz de paso  $P$ . Por tanto,  $\alpha_k$  corresponde con las coordenadas de  $w$  en la base de los autovectores.

Para poder obtener la solución del problema de Riemann, es necesario ordenar los autovalores de  $A$  en forma ascendente

$$\lambda_1 < \lambda_2 < \dots < \lambda_p, \quad (1.26)$$

tal que el índice del autovalor  $k$ -ésimo corresponda con la columna  $k$ -ésima de la matriz de paso  $P$ , es decir, es necesario reorganizar del mismo modo la matriz  $P$ .

Dada la ecuación 1.25 se tiene

$$w_l = \sum_{k=1}^p \alpha_{k,l} e_k, \quad w_r = \sum_{k=1}^p \alpha_{k,r} e_k. \quad (1.27)$$

---

<sup>2</sup>En caso de discretizarse el problema.

En este caso se tienen  $p$  problemas de Riemann cuya solución exacta es

$$u_k(x, t) = u_{k,0}(x - \lambda_k t) = \begin{cases} \alpha_{k,l} & x - \lambda_k t < 0, \\ \alpha_{k,r} & x - \lambda_k t > 0, \end{cases} \quad (1.28)$$

de esta forma puede reconstruirse la solución exacta del mismo modo que el Problema de Cauchy, obteniendo

$$w(x, t) = P u(x, t), \quad (1.29)$$

teniendo en cuenta que el vector columna de la solución exacta en las variables conservativas  $u(x, t)$  se puede expresar como

$$u(x, t) = \begin{pmatrix} \alpha_{1,r} \\ \vdots \\ \alpha_{I,r} \\ \vdots \\ \alpha_{p,l} \end{pmatrix}, \quad (1.30)$$

donde  $I$  representa el índice donde el pie de las características  $x - \lambda_I t$  cambia de signo positivo a negativo, es decir,

$$x - \lambda_1 t > \dots > x - \lambda_I t > 0 > x - \lambda_{I+1} t > \dots > x - \lambda_p t. \quad (1.31)$$

En resumen, el procedimiento para resolver el problema de Riemann asociado a un sistema lineal de ecuaciones de carácter hiperbólico es:

1. Obtener la matriz  $A$ , sus autovalores, autovectores y su matriz de paso  $P$ .
2. Reorganizar los autovalores y autovectores.
3. Calcular  $\alpha_{k,l}$  y  $\alpha_{k,r}$  con  $k = 1, \dots, p$ , a partir de la Ec. 1.27.
4. Obtener la solución  $w(x, t)$  a partir de la Ec. 1.29.

### Caso de estudio

Para el problema de estudio, con condiciones iniciales dadas por la Ec. 1.24 con  $w_l = (h_l, u_l)^T$  y  $w_r = (h_r, u_r)^T$  la ecuación 1.29 puede presentarse como

$$w(x, t) = \begin{cases} w_l & x - \lambda_1 t < 0, \\ w_l + (\alpha_{1,r} - \alpha_{1,l})e_1 & (x - \lambda_1 t)(x - \lambda_2 t) < 0, \\ w_r & x - \lambda_2 t > 0, \end{cases} \quad (1.32)$$

donde  $\alpha_{1,l}$  y  $\alpha_{1,r}$  son constantes y vienen dadas por

$$\begin{aligned} u_1(x, 0) = u_{1,0}(x - \lambda_k t) &= \begin{cases} \alpha_{1,l} & x - \lambda_1 t < 0, \\ \alpha_{1,r} & x - \lambda_1 t > 0, \end{cases} = \begin{cases} P^{-1}w_{1,l} & x - \lambda_1 t < 0, \\ P^{-1}w_{1,r} & x - \lambda_1 t > 0, \end{cases} \\ &= \begin{cases} P^{-1}(h_l, u_l)^T & x - \lambda_1 t < 0, \\ P^{-1}(h_r, u_r)^T & x - \lambda_1 t > 0, \end{cases} = \begin{cases} u_l + \sqrt{\frac{g}{h}}h_l & x - \lambda_1 t < 0, \\ u_r + \sqrt{\frac{g}{h}}h_r & x - \lambda_1 t > 0, \end{cases} \end{aligned} \quad (1.33)$$

es decir,

$$\alpha_{1,l} = u_l + \sqrt{\frac{g}{\tilde{h}}} h_l, \quad \alpha_{1,r} = u_r + \sqrt{\frac{g}{\tilde{h}}} h_r. \quad (1.34)$$

### 1.2.4. Método de Godunov

El método de Godunov es un esquema descentrado en tiempo y en espacio, que consiste en resolver dos problemas de Riemann locales para cada celda  $C_i$ , la frontera izquierda involucra a  $w_{i-1}^n, w_i^n$  y la derecha  $w_i^n, w_{i+1}^n$ , tomando como solución  $w_i^{n+1}$  el promedio de las soluciones obtenidas en cada frontera local. Para la ecuación de transporte, el método de Godunov en forma discreta conservativa se define como

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{\Delta x} \left( f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n \right), \quad (1.35)$$

donde  $f_{i\pm\frac{1}{2}}^n$  es el flujo numérico y corresponde con el flujo físico  $f$  evaluado en la frontera de la celda del volumen finito  $C_i$ , expresado en un sistema local de coordenadas tal que la frontera está situada en  $x/t = 0^3$ , que va a ser denotado por  $w_{i\pm\frac{1}{2}}^n(0)$ :

$$\begin{aligned} f_{i+\frac{1}{2}}^n &= \phi(w_i^n, w_{i+1}^n) = f\left(w_{i+\frac{1}{2}}^n(0)\right) = Aw_{i+\frac{1}{2}}^n(0), \\ f_{i-\frac{1}{2}}^n &= \phi\left(w_{i-\frac{1}{2}}^n, w_i^n\right) = f\left(w_{i-\frac{1}{2}}^n(0)\right) = Aw_{i-\frac{1}{2}}^n(0). \end{aligned} \quad (1.36)$$

Tras ciertas consideraciones y aplicando técnicas de descomposición del flujo (véase [1]) se acaba obteniendo

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{\Delta x} \left( A^+(w_i^n - w_{i-1}^n) + A^-(w_{i+1}^n - w_i^n) \right), \quad (1.37)$$

donde  $A^\pm$  corresponden con la parte positiva y negativa de la matriz  $A$ , definidas a partir del signo de sus autovalores correspondientes  $\lambda_k^\pm, k = 1, \dots, p$ :

$$A^\pm = P\Lambda^\pm P^{-1} = P \left[ \frac{1}{2} (\Lambda \pm |\Lambda|) \right] P^{-1}, \quad (1.38)$$

siendo  $\Lambda$  la matriz diagonal cuyas componentes corresponden con los autovalores de la matriz  $A$ .

Desarrollando la expresión del flujo numérico Ec.1.36 se tiene para  $f_{i+\frac{1}{2}}^n$

$$\begin{aligned} f_{i+\frac{1}{2}}^n &= f\left(w_{i+\frac{1}{2}}^n(0)\right) = Aw_{i+\frac{1}{2}}^n(0) \\ &= \frac{1}{2} (Aw_i^n + Aw_{i+1}^n) - \frac{1}{2} \sum_{k=1}^p \text{sign}(\lambda_k)(\alpha_{r,k} - \alpha_{l,k}) Ae_k, \\ &= \{Ae_k = \lambda e_k\} = \\ &= \frac{1}{2} (f(w_i^n) + f(w_{i+1}^n)) - \underbrace{\frac{1}{2} \sum_{k=1}^p |\lambda_k| (\alpha_{r,k} - \alpha_{l,k}) e_k}_{\text{viscosidad numérica}} \end{aligned} \quad (1.39)$$

---

<sup>3</sup>Como el problema de Riemann es autosimilar, más que depender de  $x$  por un lado y de  $t$  por otro, depende de  $x/t$ , porque es constante a lo largo de las características.

donde el segundo miembro de la ecuación representa la viscosidad numérica, siendo éste el encargado de estabilizar el esquema numérico. Del resultado, se puede anticipar que para autovalores nulos, el esquema se vuelve incondicionalmente inestable, del mismo modo que el esquema centrado<sup>4</sup>.

La ecuación 1.37, reorganizando la notación como en la ecuación de ondas, puede presentarse como

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{\Delta x} \left( \delta f_{i-\frac{1}{2}}^+ + \delta f_{i+\frac{1}{2}}^- \right), \quad (1.40)$$

con

$$\delta f_{i\mp\frac{1}{2}}^\pm = A^\pm \delta w_{i\mp\frac{1}{2}} \rightarrow \begin{cases} \delta w_{i-\frac{1}{2}} = w_i^n - w_{i-1}^n, \\ \delta w_{i+\frac{1}{2}} = w_{i+1}^n - w_i^n. \end{cases} \quad (1.41)$$

Lo expuesto, puede interpretarse como la comunicación de una celda con otra donde [2]:

- La información que entra por la izquierda de la celda está asociada con los autovalores positivos,  $\delta f_{i-\frac{1}{2}}^+$ .
- La información que entra por la derecha de la celda está asociada con los autovalores negativos,  $\delta f_{i+\frac{1}{2}}^-$ .

### Observación

Obsérvese, que la  $\lambda_k$  está implícito en la matriz  $A^\pm$  de la Ec. 1.41. Recordando que  $\lambda_k$  representa la velocidad a la que se propaga la información, nos indica que en los esquemas descentrados la información para cada variable característica se obtiene buscando en la dirección en la que la información podría estar viniendo.

### Análisis de estabilidad de von Neumann y la condición CFL

Un aspecto que aún no se ha tratado y es de vital relevancia de cara al análisis del problema, es la estabilidad del esquema en función de la discretización espaciotemporal. Para ello, se aplica el análisis de von Neumann, véase [1], donde se llega a que un esquema en diferencias es un espacio  $L^2$  estable, si y sólo si sus factores de amplificación  $g(k, \Delta x, \Delta t)$  verifican

$$|g(k, \Delta x, \Delta t)| \leq 1, \quad \forall k, \quad (1.42)$$

siendo

$$g(k, \Delta x, \Delta t) = \sum_{l=-1}^{l=1} c_l e^{ikl\Delta x} = \frac{\hat{w}^{n+1}(k)}{\hat{w}^n(k)}. \quad (1.43)$$

Teniendo en cuenta que el problema de estudio es un sistema lineal que se desacopla en  $p$  ecuaciones de transporte lineal, siendo  $p = 2$ , parece razonable, atacar el problema en su versión desacoplada para evitar una notación más engorrosa. Tras ciertas manipulaciones algebraicas, de la definición anterior, se alcanza, que la condición

---

<sup>4</sup>Para autovalores nulos, el flujo numérico del esquema centrado coincide con el flujo numérico del método de Godunov.

de estabilidad para sistemas es

$$\max_{k=1,\dots,p} |\lambda_k| \frac{\Delta t}{\Delta x} \leq 1, \quad (1.44)$$

donde  $\lambda \frac{\Delta t}{\Delta x}$  se conoce como número CFL o de *Courant*. El CFL puede interpretarse como el cociente entre la velocidad a que se propaga la información  $\lambda$  y la velocidad a la que se propaga la discretización espacial  $\frac{\Delta x}{\Delta t}$ , donde para un esquema de primer orden que sólo involucra a las celdas vecinas (izquierda y derecha) de  $C_i$ , parece coherente pensar que para poder capturar correctamente la información es necesario desplazarse a una velocidad igual o mayor a ésta para cada paso de tiempo  $n$ .

Como se verá en los resultados obtenidos, en el caso de tener  $p$  autovalores iguales en valor absoluto y un *Courant* unitario, la discretización espacial se mueve exactamente a la misma velocidad que la información en dirección y sentido, lo que permite retener de forma resolver de forma exacta el problema de Riemann local, obteniendo la solución exacta en todo el dominio  $\Omega$  del problema.

## 1.3. Resultados

### 1.3.1. Parámetros del problema

Se va a realizar un análisis con tres números de *Courant*, concretamente,  $\mu < 1$ ,  $\mu = 1$ , y  $\mu > 1$ . Así mismo, se van a emplear tres mallas espaciales con un número de nodos en la variable espacial  $N_x = 150, 300$  y  $600$ , a las que denotaremos como basta, media y fina, respectivamente.

#### Caso 1

- $\tilde{h} = 7 \text{ m}$ ,
- $\tilde{u} = 0,5 \text{ m/s}$ ,
- $g = 9,8 \text{ m/s}^2$ ,

obteniendo dos autovalores distintos y de signo opuesto dados por la Ec.1.13.

### 1.3.2. Condiciones iniciales

El problema planteado 1.1 se ha resuelto para dos tipos de condiciones iniciales:

- **Problema de Riemann:** Condiciones iniciales con un a discontinuidad de salto en un punto  $x_b$ , tenido lo que se conoce como esta de la izquierda  $w_l$  para  $x < x_b$  y estado de la derecha para  $x > x_b$ .
- **Problema de Cauchy:** Condiciones iniciales dadas por una función  $f(x)$  (vectorial al ser un sistema). En este caso se ha escogido una condición inicial regular, concretamente, una gaussiana, con el objetivo de analizar el comportamiento de los métodos ante condiciones iniciales regulares.

### Problema de Riemann

$$w(x, 0) = w_0(x) = \begin{cases} w_l, & x < 0, \\ w_r, & x > 0, \end{cases} \quad (1.45)$$

donde se ha tomado  $w_l = [h_l, u_l] = [10, 0]$  y  $w_r = [h_r, u_r] = [5, 0]$ .

### Problema de Cauchy

$$w(x, 0) = w_0(x) = [h_0; u_0] = \begin{cases} h_0 = 1 + \frac{2}{5}e^{-5x^2}, \\ u_0 = 0. \end{cases} \quad (1.46)$$

#### 1.3.3. Análisis

En la figura 1.1 se ha representado la variación temporal de la solución  $w(x, t)$  para el problema de Riemann con condiciones iniciales dadas por la Ec.1.45 ante dos *Courant*: (a)  $\mu = 0,8$ , (b)  $\mu = 1,01$ . En la subfigura 1.1a se observa como para un *Courant*, en el rango estable ( $CFL \leq 1$ ), la solución obtenida mediante el método de Godunov captura con una exactitud del orden  $\mathcal{O}(10^{-1})$ , siendo un gran resultado para la simplicidad del método. En contra partida, en la subfigura 1.1b se tiene un *Courant* ligeramente por encima de la condición de estabilidad. Se observa como la inestabilidad crece de manera no acotada, dando lugar a soluciones espurias. Profundizando más sobre la transición mostrada, se observa que la solución de la onda que viaja hacia la izquierda sigue reteniendo en buen grado la solución, siendo la onda que viaja hacia la derecha la inestable. Esto se debe al valor del autovalor  $\lambda^+$  cuyo valor es mayor al valor absoluto de  $\lambda^-$ . Para tener una solución estable el CFL es necesario fijarlo para la onda que viaja a la mayor velocidad con el objetivo de poder capturar de forma correcta la solución del problema. Debido al ligero incremento en el CFL, éste no supera la condición de estabilidad en la onda de la izquierda, de ahí lo observado.

En la figura 1.2 se observa como la solución ante una condición inicial más regular, como es una gaussiana, el método responde de mejor modo en el cálculo de la solución. A pesar de ello, para tiempos suficientemente largos, dado que la velocidad de propagación de la información de la onda es mayor a la velocidad de propagación de la malla,  $Courant > 1$ , genera una inestabilidad que da lugar a un incremento no acotado del error, ocasionando soluciones espurias. Nótese, que debido a la regularidad de la condición inicial, ha sido necesario incrementar el valor del CFL, para obtener un error apreciable en el mismo tiempo.

En la figura 1.3 se ha representado el error cometido en función al número de nodos espaciales y y para un  $CFL=0,8$  y hasta el instante  $t = 0,5 s$  con  $x \in [-8, 8]$ , correspondiendo las subfiguras (a) y (b) con el problema de Riemann y Cauchy, respectivamente. El desarrollo del error es el esperado. En función al número de nodos considerados en la discretización espacial el error cometido varía, tiendiendo a aumentar para mallas más bajas (menor número de nodos). Nótese, que el error cometido se incrementa con el transcurso del tiempo llegando a alcanzar un error del orden  $\mathcal{O}(10^{-2})$  en el cálculo de la altura y del orden  $\mathcal{O}(10^{-1})$  en el cálculo de la velocidad, para ambas condiciones iniciales.

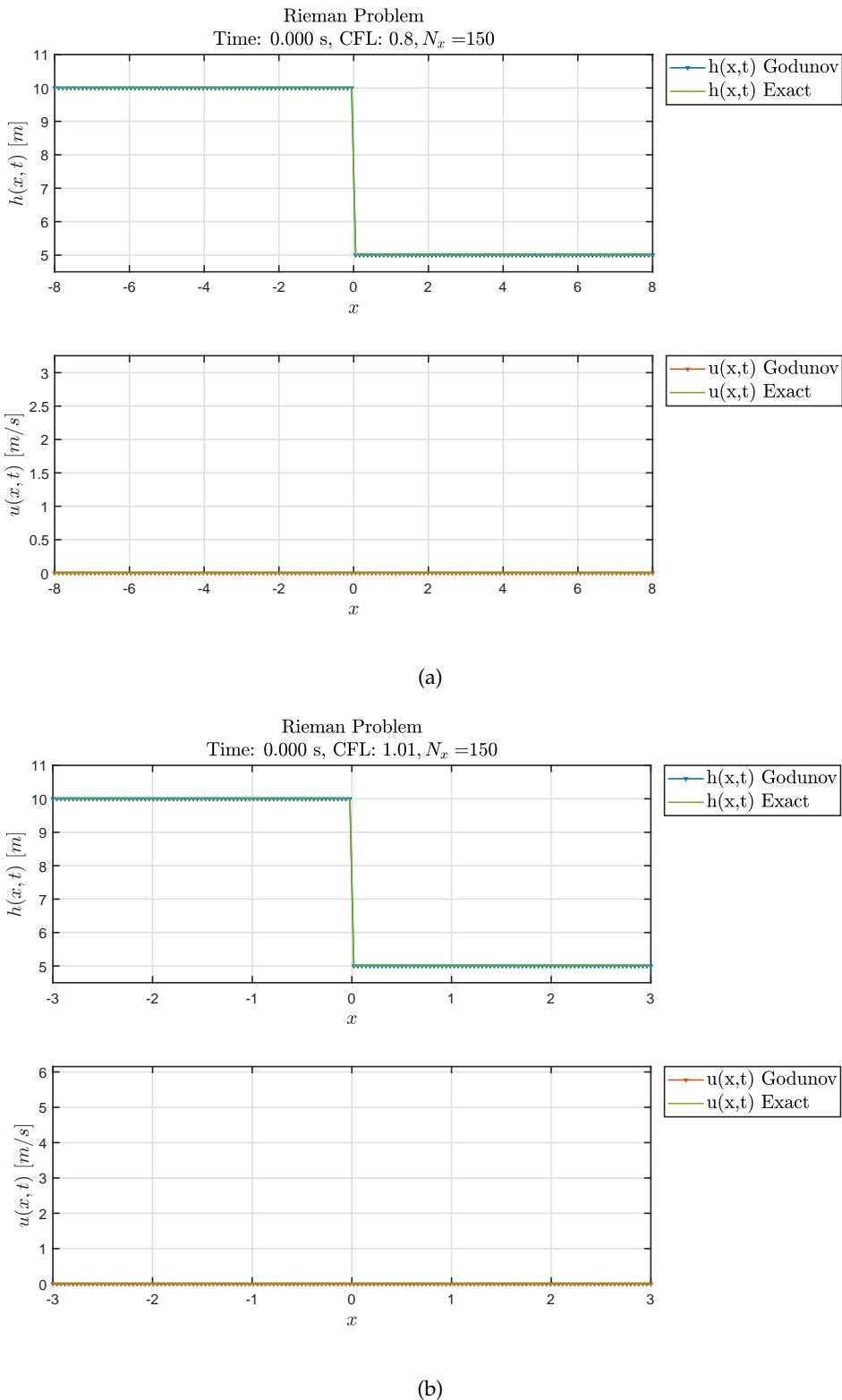
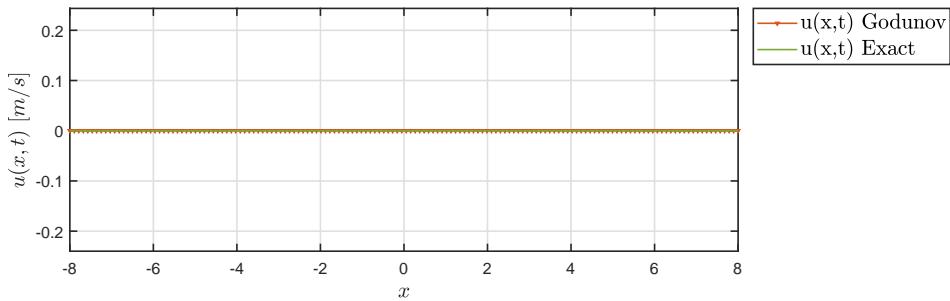
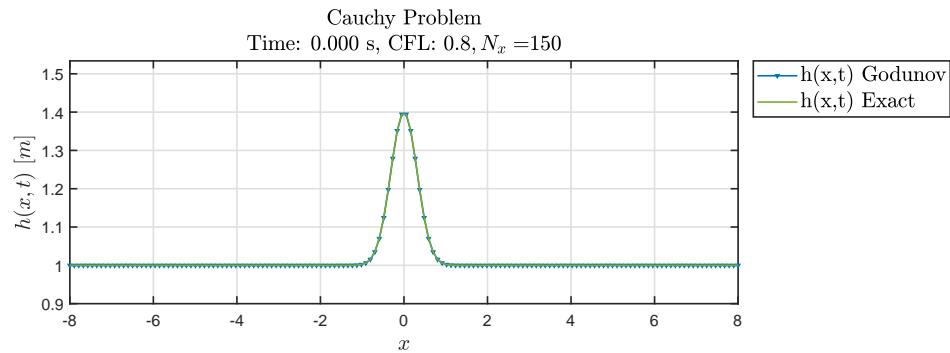
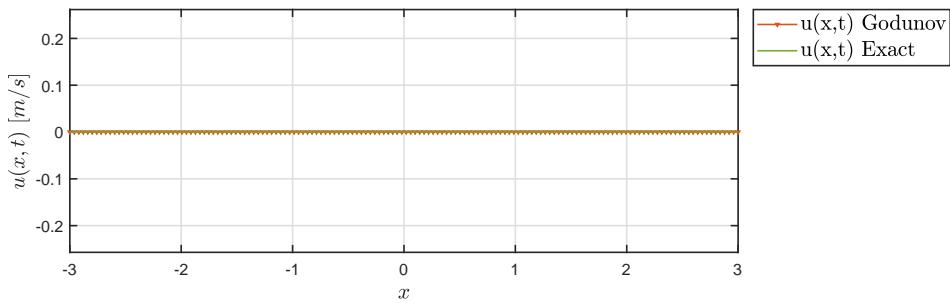
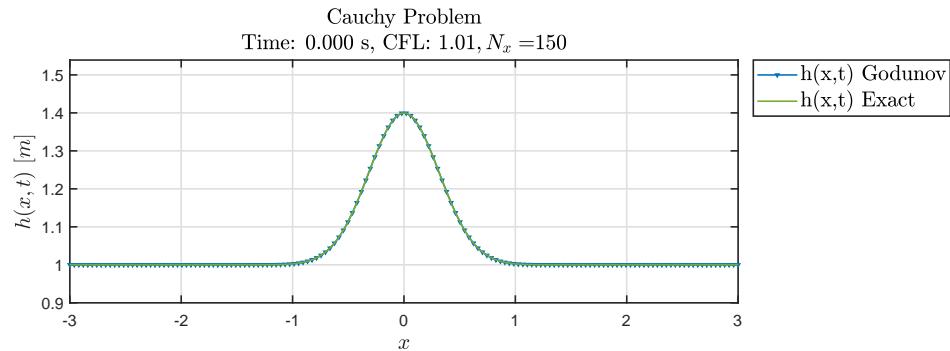


FIGURA 1.1: Solución del sistema lineal hiperbólico *One-Dimensional Linearized Shallow water equations* asociado al problema de Riemann: exacta y mediante el método de Godunov. Rango:  $x \in [-3, 3]$  m,  $t \in [0, 0.3]$  s.  
*Courant:* (a)  $\mu = 0.8$ , (b)  $\mu = 1.01$ .



(a)



(b)

FIGURA 1.2: Solución del sistema lineal hiperbólico *One-Dimensional Linearized Shallow water equations* asociado al problema de Cauchy: exacta y mediante el método de Godunov. Rango:  $x \in [-3, 3]$  m,  $t \in [0, 0,3]$  s.  
*Courant:* (a)  $\mu = 0,8$ , (b)  $\mu = 1,01$ .

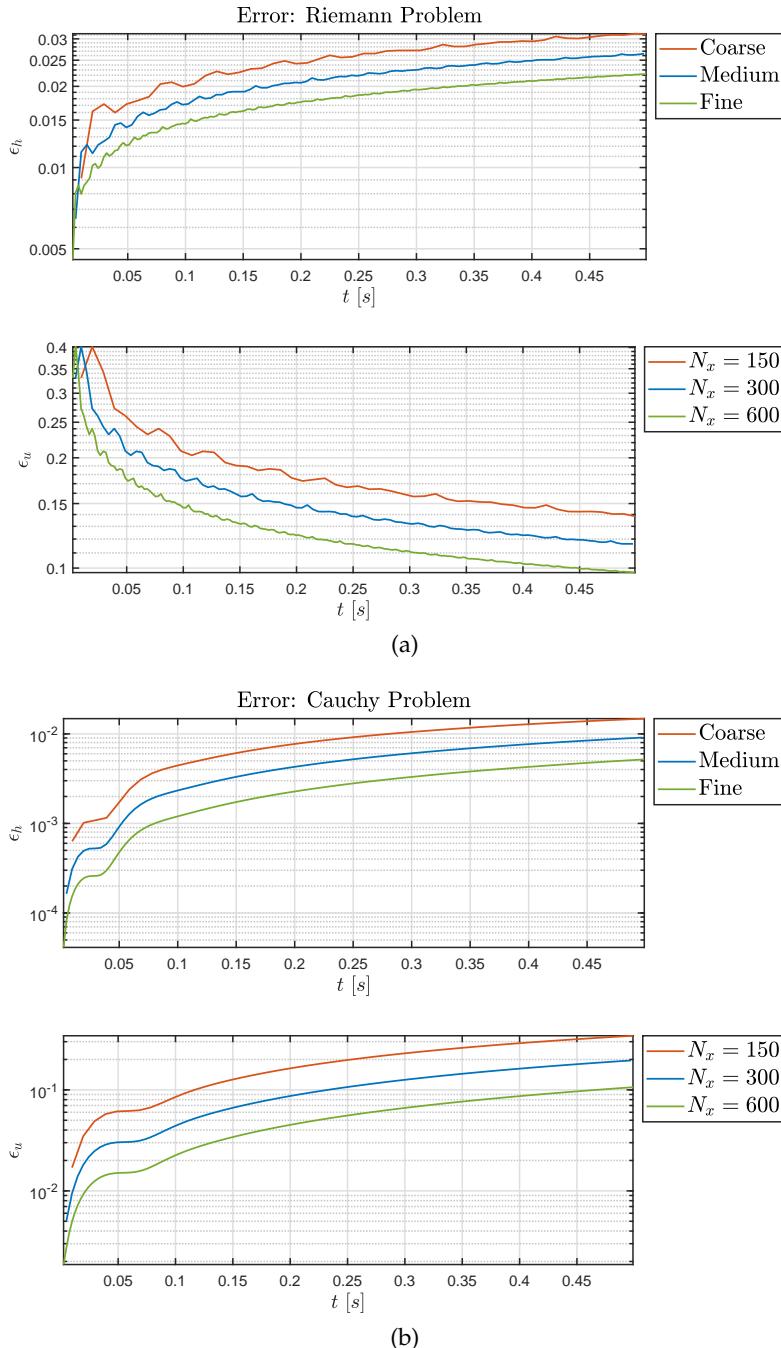


FIGURA 1.3: Solución del sistema lineal hiperbólico *One-Dimensional Linearized Shallow water equations* asociado al problema de Cauchy: exacta y mediante el método de Godunov. Rango:  $x \in [-3, 3]$  m,  $t \in [0, 0,3]$  s.  
*Courant:* (a)  $\mu = 0,8$ , (b)  $\mu = 1,5$ .

## 1.4. Conclusiones y trabajo futuro

Se ha demostrado que a pesar de partir de condiciones iniciales más regulares, si no se cumple la condición de estabilidad  $CFL < 1$  se tienen soluciones espurias. Además, se ha observado como el método de Godunov para un  $CFL = 1$  coincide con la solución exacta para el máximo autovalor en valor absoluto. Esto se debe a que a que el método de Godunov evalúa el flujo físico en la frontera del volumen finito, en caso de que éste se desplace a la misma velocidad que a la velocidad a la que se transmite la información ( $CFL = 1$ ) la solución calculada coincide exactamente con el flujo físico y por tanto con la solución exacta. También se ha demostrado que el número de nodos considerados en la discretización espacial es representativo en el orden del error alcanzado en el problema. Al ser un problema espacio-temporal el error también tiende a aumentar con el transcurso del tiempo.

## Anexo A

## Código utilizado en MATLAB

#### LISTADO A.1: Función principal.

```

1 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
2 % FINITE VOLUME METHOD: Linear Systems
3 % Author: Alberto Cuadra Lara
4 % Email: albertocuadralara@gmail.com
5 % Last update: 23/04/2018
6 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
7 clearvars -except t1 t2 t3 nx1 nx2 nx3 err1 err2 err3;
8 close all; clc; help main_linearsystem;
9 global A framerate auxsave nx nt nl x t cfl solve_problem nx1 nx2 nx3 fpath
10 % -----
11 % Save figures path
12 % -----
13 fpath = strcat('D:\Google Drive\M2i\Ampliacion de volumenes finitos\',...
    'Trabajo individual\Latex\Figures\');
14 auxsave = 0; % Save figures == 1
15 framerate = 0; % Framerate plotting
16 % -----
17 % PROBLEM CONFIGURATION
18 % -----
19 % Matrix A
20 a11 = 7;
21 a12 = 0.5;
22 a21 = 9.8;
23 a22 = 0.5;
24 A = [a12 a11;a21 a12];
25 % Initial condition
26 wl1 = 10; % Left state 1
27 wl2 = 0; % Left state 2
28 wr1 = 5; % Right state 1
29 wr2 = 0; % Right state 2
30 wl = [wl1, wl2];
31 wr = [wr1, wr2];
32 % Spatial discretization
33 a = -3; % Left bound
34 b = 3; % Right bound
35 nx = 150; % Spatial nodes
36 deltax = (b-a)/(nx-1);
37 x = a:deltax:b; % Spatial discretization
38 % Time discretization
39 t0 = 0; % Initial time
40 tmax=0.3; % Final time
41 cfl =0.8; % Courant number
42 [P_aux,D] = eig(A);
43 [lambda,ind] = sort(diag(D));
44 nl = length(lambda); % Number of eigenvalues
45 deltat=cfl*deltax/max(abs(lambda)); % Deltat from Courant number
46 dtdx= deltat/deltax; % dt/dx

```

```

48 t = t0:deltat:tmax; % Time discretization
49 nt = length(t); % Time nodes
50 % -----
51 % Display configuration
52 % -----
53 disp('----- Configuration -----')
54 fprintf('Nx = %d \n',nx)
55 % fprintf('a = %.3f \n',a)
56 % fprintf('b = %.3f \n',b)
57 % fprintf('dx = %.3e \n',deltax)
58 fprintf('Nt = %d \n',nt)
59 fprintf('t0 = %.3f \n',t0)
60 fprintf('tf = %.3f \n',tmax)
61 % fprintf('dt = %.3e \n',deltat)
62 fprintf('CFL = %.3f \n',cfl)
63 fprintf('wl = [% .3f, % .3f] \n',wl)
64 fprintf('wr = [% .3f, % .3f] \n',wr)
65 disp('-----')
66 % % SOLVING THE SYSTEM – RIEMANN PROBLEM
67 riemann;
68 % Plot and Video
69 filename = strcat ('shallow1d1',num2str(nx));
70 % plotfun(filename,w_riemann)
71 disp('End')
72 % % SOLVING THE SYSTEM – CAUCHY PROBLEM
73 cauchy;
74 % PLOT AND VIDEO
75 filename = strcat ('shallow1d2',num2str(nx));
76 % plotfun(filename,w_cauch)
77 disp('End')
78 % % GODUNOV METHOD FOR LINEAR SYSTEMS
79 % Solve problem with initial conditions:
80 % – Riemann problem --> solve_problem == 1
81 % – Cauchy problem --> solve_problem != 1
82 solve_problem = 0;
83 godunov;
84 % PLOT AND VIDEO
85 if solve_problem
86     filename = strcat ('shallow1d3r',num2str(nx));
87 else
88     filename = strcat ('shallow1d3c',num2str(nx));
89 end
90 plotfuncomp(filename,wn,w)
91 disp('End')
92 % -----
93 % % Relative error
94 % -----
95 fprintf('Relative error for Nx= %d \n',nx)
96 disp('ATTENTION! TAKE IN ACCOUNT THE DOMAIN IN')
97 disp('THE CALCULUS OF THE ERROR FOR LONG TIMES')
98 for i=1:nt
99     if nx == 150
100         if solve_problem
101             err1(i,1) = norm(w_riem(:,i,1)-wn(:,i,1))/norm(wn(:,i,1));
102             err1(i,2) = norm(w_riem(:,i,2)-wn(:,i,2))/norm(wn(:,i,2));
103         else
104             err1(i,1) = norm(w_cauch(:,i,1)-wn(:,i,1))/norm(wn(:,i,1));
105             err1(i,2) = norm(w_cauch(:,i,2)-wn(:,i,2))/norm(wn(:,i,2));
106         end
107         nx1 = nx;
108         t1 = t;

```

```

109 elseif nx == 300
110     if solve_problem
111         err2(i,1) = norm(w_riem(:,i,1)-wn(:,i,1))/norm(wn(:,i,1));
112         err2(i,2) = norm(w_riem(:,i,2)-wn(:,i,2))/norm(wn(:,i,2));
113     else
114         err2(i,1) = norm(w_cauch(:,i,1)-wn(:,i,1))/norm(wn(:,i,1));
115         err2(i,2) = norm(w_cauch(:,i,2)-wn(:,i,2))/norm(wn(:,i,2));
116     end
117     nx2 = nx;
118     t2 = t;
119 elseif nx == 600
120     if solve_problem
121         err3(i,1) = norm(w_riem(:,i,1)-wn(:,i,1))/norm(wn(:,i,1));
122         err3(i,2) = norm(w_riem(:,i,2)-wn(:,i,2))/norm(wn(:,i,2));
123     else
124         err3(i,1) = norm(w_cauch(:,i,1)-wn(:,i,1))/norm(wn(:,i,1));
125         err3(i,2) = norm(w_cauch(:,i,2)-wn(:,i,2))/norm(wn(:,i,2));
126     end
127     nx3 = nx;
128     t3 = t;
129 end
130 if exist('t1') && exist('t2') && exist('t3')
131     if solve_problem
132         filename = 'shallow1d_errorr';
133     else
134         filename = 'shallow1d_errorc';
135     end
136 pos1 = 1; % First value is exact accord to the initial condition
137 plot_error(filename,t1,t2,t3,err1,err2,err3,pos1)
138 disp('End')
139
140 end

```

LISTADO A.2: Llamada Problema de Cauchy.

```

1 % SOLVING THE SYSTEM – CAUCHY PROBLEM
2 % Eigenvectors and eigenvalues
3 [P,D] = eig(A);
4 lambda = diag(D);
5 % Summoning w_cauchy
6 for i=1:length(t)
7     for j=1:length(x)
8         w_cauch(j,i,:) = w_cauch(x(j),t(i),P,lambda);
9     end
10 end
11 disp('Results:')
12 fprintf ('A      = [ %.3f %.3f %.3f %.3f]\n',A);
13 fprintf ('P      = [ %.3f %.3f %.3f %.3f]\n',P);
14 fprintf ('lambda = [ %.3f %.3f]\n',lambda);
15 fprintf ('Solution = w_cauch\n');
16 disp('-----')

```

LISTADO A.3: Condiciones iniciales del problema de Cauchy.

```

1 function w0 = w0_cauchy(x)
2     h0 = 1+2/5.*exp(-5.*x.^2);
3     u0 = zeros(1,length(x));
4     % h0 = 0.5.*exp(-(x-20).^2./8);
5     % u0 = 100.*h0;
6     w0 = [h0;u0];
7     % w0 = [cos(x);sin(x)];

```

```
8 end
```

LISTADO A.4: Componentes de la condición inicial  $u_{k,0}$  de las variables características.

```
1 function u0 = u0_cauchy(x,P,i)
2     u0 = P \ w0_cauchy(x);
3     u0 = u0(i ,:); % In order to return the correct value
4 end
```

LISTADO A.5: Cálculo de la solución del problema de Cauchy.

```
1 function [w] = w_cauchy(x,t,P,lambda)
2     u = zeros(length(lambda),1);
3     for i=1:length(lambda)
4         u(i,1) = u0_cauchy(x-lambda(i)*t,P,i);
5     end
6     w = P*u;
7 end
```

LISTADO A.6: Llamada Problema de Riemann.

```
1 % SOLVING THE SYSTEM – RIEMANN PROBLEM
2 disp('--- Solving the system: Riemann Problem ---')
3 % Eigenvectors and eigenvalues
4 [P_aux,D] = eig(A);
5 % Ordering eigenvalues and eigenvectors
6 % lambda1 < lambda2 < ... < lambdaP
7 % x-lambda1*t > 0 > ... > x-lambdaP*t
8 [lambda,ind] = sort(diag(D));
9 P=P_aux(:,ind);
10 alphal = P \ wl; %inv(P)*wl
11 alphar = P \ wr; %inv(P)*wr
12 % Summoning w_riemann
13 w_riem = zeros(length(x),length(t),length(lambda));
14 for i=1:length(t)
15     w_riem(:,i ,:) = w_riemann(x,t(i),lambda,wl,wr,alphal,alphar,P);
16 end
17 disp('Results: ')
18 fprintf ('A      = [ %.3f, %.3f, %.3f, %.3f ]\n',A);
19 fprintf ('P      = [ %.3f, %.3f, %.3f, %.3f ]\n',P);
20 fprintf ('lambda = [ %.3f, %.3f ]\n',lambda);
21 fprintf ('alphal = [ %.3f, %.3f ]\n',alphal);
22 fprintf ('alphar = [ %.3f, %.3f ]\n',alphar);
23 fprintf ('Solution = w_riem\n');
24 disp('-----')
```

LISTADO A.7: Cálculo de la solución del problema de Riemann

```
1 function [w] = w_riemann(x,t,lambda,wl,wr,alphal,alphar,P)
2 w = zeros(length(x),length(lambda));
3 for i=1:length(x)
4     if (x(i)-lambda(1)*t)<=0 % Case 1:
5         w(i ,:) = wl;
6     elseif (x(i)-lambda(1)*t)*(x(i)-lambda(2)*t)<0 % Case 2:
7         w(i ,:) = wl + (alphar(1)-alphal(1))*P (:,1);
8     elseif (x(i)-lambda(2)*t)>=0 % Case 3:
9         w(i ,:) = wr;
10    end
11 end
12 % CHECK
```

```

13 % wll = 0;
14 % for i=1:2
15 % wll = wll + alphal(i)*P(:, i);
16 % end

```

LISTADO A.8: Llamada método de Godunov.

```

1 % GODUNOV METHOD FOR LINEAR SYSTEMS
2 disp('Godunov Method for Linear Systems:')
3 disp('-----')
4 if solve_problem
5   disp('Riemann Problem:')
6   wa = zeros(nl,nx);
7   for i=1:nx
8     if x(i)<0 % Case 1:
9       wa(:,i) = wl;
10    else % Case 2:
11      wa(:,i) = wr;
12    end
13  end
14  w = w_riem;
15 else
16   disp('Cauchy Problem:')
17   wa=w0_cauchy(x);
18   w = w_cauch;
19 end
20 % Summoning Godunov Method
21 wn = zeros(nl,nx,nt);
22 wn (:,:,1) = wa;
23 for i=2:nt
24   wn (:,:, i) = god(x,A,dtdx,wa);
25   wa = wn (:,:, i);
26 end
27 wn = permute(wn,[2 3 1]);

```

LISTADO A.9: Cálculo de la solución aproximada mediante el método de Godunov

```

1 function wn = god(x,A,dtdx,wa)
2 % [ind1,ind2] = ind2sub(size(A),find(A>0));
3 nx = length(x);
4 [P,D] = eig(A);
5 A_plus = P*(0.5*(D+abs(D)))*inv(P); %#ok<MINV>
6 A_minus= P*(0.5*(D-abs(D)))*inv(P); %#ok<MINV>
7 % wn=zeros(length(A(:,1)),length(x));
8 wam(:,1:nx-1)=wa(:,2:nx);
9 % Transmissive boundary conditions
10 wam(:,nx)=wa(:,nx-1);
11 wan(:,2:nx)=wa(:,1:nx-1);
12 % Transmissive boundary conditions
13 wan(:,1)=wa(:,2);
14 % Godunov scheme for Linearized Shallow Water 1D
15 wn(:,1:nx)=wa(:,1:nx)-dtdx*(A_plus*(wa(:,1:nx))-wan(:,1:nx))+A_minus*(wam(:,1:nx)-wa(:,1:nx)));

```

LISTADO A.10: Función *plotfun*

```

1 function plotfun(filename,w)
2 global fpath framerate auxsave nx nt x t
3 color = colours;
4 % Plot configuration
5 fig = figure;

```

```

6 set(fig,'innerposition',[250,200,1150,700],...
7      'outerposition',[250 200 1000 600]);
8 hold on
9 sub1 = subplot(2,1,1);
10 grid on; box on; hold on;
11 set(sub1,'LineWidth',1.2,'FontSize',12,'BoxStyle','full')
12 xlabel('$x$', 'FontSize',16, 'interpreter','latex');
13 ylabel('$h(x,t) \backslash [m]$', 'FontSize',16, 'interpreter','latex');
14 ylim([min(min(w(:,:,1)))-0.2 max(max(w(:,:,1)))+0.2]);
15 sub2 = subplot(2,1,2);
16 grid on; box on; hold on;
17 set(sub2,'LineWidth',1.2,'FontSize',12,'BoxStyle','full')
18 xlabel('$x$', 'FontSize',16, 'interpreter','latex');
19 ylabel('$u(x,t) \backslash [m/s]$', 'FontSize',16, 'interpreter','latex');
20 ylim(sub2,[min(min(w(:,:,2)))-0.2 max(max(w(:,:,2)))+0.2]);
21 % mov=VideoWriter(strcat(fpPath,filename),'MPEG-4');
22 % set(mov,'FrameRate',framerate);
23 % open(mov);
24 % Plot
25 for i=1:nt
26 if exist('dl1')
27     delete(dl1);
28     delete(dl2);
29 end
30 dl1 = plot(x,w(:,i,1), 'Parent',sub1,'color',color(2,:),'LineWidth',1.4);
31 tit = [ 'Time: ',num2str(sprintf('%.3f',round(t(i),3))), ' s '];
32 title({ tit }, 'Parent',sub1,'Interpreter','latex','FontSize',18)
33 dl2 = plot(x,w(:,i,2), 'Parent',sub2,'color',color(1,:),'LineWidth',1.4);
34 pause(framerate)
35 % M(i)=getframe(fig);
36 if auxsave
37     filename2 = strcat(fpPath,filename,'\' ,filename,' -',num2str(i));
38     saveas(fig,filename2,'epsc');
39 end
40 end
41 % writeVideo(mov,M);
42 % movie(M);
43 % close(mov);

```

LISTADO A.11: Función *plotfuncomp*

```

1 function plotfuncomp(filename,wn,w)
2 global fpPath nx x t cfl framerate solve_problem auxsave
3 % Plot configuration
4 color = colours;
5 fig = figure;
6 if solve_problem
7     tit0 = 'Riemann Problem';
8 else
9     tit0 = 'Cauchy Problem';
10 end
11 set(fig,'innerposition',[250,50,1150,800],...
12      'outerposition',[250 50 1150 800]);
13 sub1 = subplot(2,1,1);
14 grid on; box on; hold on;
15 set(sub1,'LineWidth',1.2,'FontSize',12,'BoxStyle','full')
16 xlabel('$x$', 'FontSize',16, 'interpreter','latex');
17 ylabel('$h(x,t) \backslash [m]$', 'FontSize',16, 'interpreter','latex');
18 minwn1 = min(min(wn(:,:,1)));
19 maxwn1 = max(max(wn(:,:,1)));
20 minwn2 = min(min(wn(:,:,2)));

```

```

21 maxwn2 = max(max(wn(:,2)));
22 ylim(sub1,[minwn1-0.1*abs(minwn1) maxwn1+0.1*abs(maxwn1)]);
23 sub2 = subplot(2,1,2);
24 grid on; box on; hold on;
25 set(sub2,'LineWidth',1.2,'FontSize',12,'BoxStyle','full')
26 xlabel('$x$', 'FontSize',16, 'interpreter', 'latex');
27 ylabel('$u(x,t) \ [m/s]$', 'FontSize',16, 'interpreter', 'latex');
28 ylim(sub2,[minwn2-0.1*abs(minwn2) maxwn2+0.1*abs(maxwn2)]);
29 % mov=VideoWriter(strcat(fpname,filename),'MPEG-4');
30 % set(mov,'FrameRate',10);
31 % open(mov);
32 % Plot
33 leg{1} = 'h(x,t) Godunov';
34 leg{2} = 'h(x,t) Exact';
35 leg{3} = 'u(x,t) Godunov';
36 leg{4} = 'u(x,t) Exact';
37 for i=1:length(t)
38 if exist('dl1')
39     delete(dl1);
40     delete(dl12);
41     delete(dl2);
42     delete(dl22);
43 end
44 tit1 = ['Time: ' num2str(sprintf('%.3f',round(t(i),3))) ' s' ...
45 ', CFL: ' num2str(cfl) '$, N_x =' num2str(nx)];
46 title (fig.Children(end),{tit0 ,tit1 },'Interpreter','latex','FontSize',16);
47 dl1 = plot(x,wn(:,i,1), '-v','Parent',sub1,'color',color (2,:',...
48 'LineWidth',1.2,'MarkerSize',2);
49 dl12 = plot(x,w(:,i,1), 'Parent',sub1,'color',color (3,:), 'LineWidth',1.2);
50 dl2 = plot(x,wn(:,i,2), '-v','Parent',sub2,'color',color (1,:',...
51 'LineWidth',1.2,'MarkerSize',2);
52 dl22 = plot(x,w(:,i,2), 'Parent',sub2,'color',color (3,:), 'LineWidth',1.2);
53 legend(sub1,leg (1:2), 'FontSize',16,'Location','northeastoutside','interpreter','latex')
54 legend(sub2,leg (3:4), 'FontSize',16,'Location','northeastoutside','interpreter','latex')
55 % M(i)=getframe(fig);
56 pause(framerate);
57 if auxssave
58     filename2 = strcat(fpname,filename,'_\$,shallow', '-',num2str(i));
59     saveas(fig,filename2,'epsc');
60 end
61 end
62 % writeVideo(mov,M);
63 % movie(M);
64 % close(mov);

```

LISTADO A.12: Función *plot\_error*

```

1 function plot_error(filename,x1,x2,x3,y1,y2,y3,pos1)
2 global fpname auxssave solve_problem nx1 nx2 nx3
3 color = colours;
4 % Plot configuration
5 fig = figure;
6 set(fig,'innerposition',[250,50,1150,800],...
7      'outerposition',[250 50 1000 800]);
8 hold on
9 if solve_problem
10    tit = 'Error: Riemann Problem';
11 else
12    tit = 'Error: Cauchy Problem';
13 end
14 leg{1} = 'Coarse';

```

```
15 leg{2} = 'Medium';
16 leg{3} = 'Fine';
17 leg2{1} = strcat ('$N_x = ',num2str(nx1),'$');
18 leg2{2} = strcat ('$N_x = ',num2str(nx2),'$');
19 leg2{3} = strcat ('$N_x = ',num2str(nx3),'$ ');
20 % SubPlot1 configuration
21 sub1 = subplot (2,1,1);
22 grid on; box on; hold on; axis tight;
23 set(sub1,'LineWidth',1.2,'FontSize',12,'BoxStyle','full')
24 set(sub1,'yscale','log')
25 xlabel('$t^ [s]$', 'FontSize',16, 'interpreter','latex');
26 ylabel('$\epsilon_h$', 'FontSize',16, 'interpreter','latex');
27 % SubPlot2 configuration
28 sub2 = subplot (2,1,2);
29 grid on; box on; hold on; axis tight;
30 set(sub2,'LineWidth',1.2,'FontSize',12,'BoxStyle','full')
31 set(sub2,'yscale','log')
32 xlabel('$t^ [s]$', 'FontSize',16, 'interpreter','latex');
33 ylabel('$\epsilon_u$', 'FontSize',16, 'interpreter','latex');
34 % Plot
35 plot(x1(pos1:end),y1(pos1:end,1),'Parent',sub1,'color',color (1,:),'LineWidth',1.4);
36 plot(x2(pos1:end),y2(pos1:end,1),'Parent',sub1,'color',color (2,:),'LineWidth',1.4);
37 plot(x3(pos1:end),y3(pos1:end,1),'Parent',sub1,'color',color (3,:),'LineWidth',1.4);
38 legend(sub1,leg,'FontSize',18,'Location','northeastoutside','interpreter','latex')
39 title ({ tit },'Parent',sub1,'Interpreter','latex','FontSize',18)
40 plot(x1,y1 (:,2), 'Parent',sub2,'color',color (1,:),'LineWidth',1.4);
41 plot(x2,y2 (:,2), 'Parent',sub2,'color',color (2,:),'LineWidth',1.4);
42 plot(x3,y3 (:,2), 'Parent',sub2,'color',color (3,:),'LineWidth',1.4);
43 legend(sub2,leg2,'FontSize',18,'Location','northeastoutside','interpreter','latex')
44 if auxsave
45     filename2 = strcat (fpath,filename);
46     saveas(fig,filename2,'epsc');
47 end
```

# Bibliografía

- [1] M. E. Vázquez Cendón. *Introducción al método de volúmenes finitos*. Universidad de Santiago de Compostela. Servicio de Publicaciones e Intercambio Científico, 2008.
- [2] M. E. Vázquez Cendón. *Solving Hyperbolic Equations with Finite Volume Methods*. Springer, 2015.
- [3] E. Godlewski y P. A. Raviart. *Numerical Approximation of Hyperbolic systems of conservation laws*. Springer, 1996.