Trabajo de Fin de Grado

Comparación de métodos de integración temporal para problemas transitorios en Mecánica de Fluidos

Grado en Física

Autor: Carlos Aguilar Abad

Directores: Pilar García-Navarro y Mario Morales-Hernández

Departamento de Ciencia y Tecnología de Materiales y Fluidos

Julio de 2023

Facultad de Ciencias
Universidad Zaragoza

Universidad Zaragoza
1542

Índice

1 Introducción **1**

2 Leyes hiperbólicas de conservación **2**
2.1 Ecuación de convección lineal 2
2.2 Ecuación de Burgers no viscosa 3
2.3 Ecuaciones *shallow water* 1D 5

3 Discretización y métodos numéricos **6**
3.1 Métodos numéricos en diferencias finitas 6
3.2 Condiciones de estabilidad 8
3.3 Métodos numéricos de volúmenes finitos 10

4 Métodos *Local Time Step* **11**
4.1 LTS1: *Frozen Flux* 12
4.2 LTS2: *Full Integration* 14

5 Resultados **15**
5.1 Ecuación de convección lineal 16
5.2 Ecuación de Burgers no viscosa 18
5.3 Ecuaciones *shallow water* 1D 20

6 Conclusiones **24**

7 Referencias **25**

Apéndice A Análisis de von Neumann **a**

Apéndice B Generación de mallas irregulares **d**

1 Introducción

En Mecánica de Fluidos, es habitual la formulación de problemas a través de leyes de conservación. Estos modelos matemáticos están constituidos por sistemas de ecuaciones en derivadas parciales con, en general, un alto grado de no linealidad, que en multitud de problemas impide su resolución analítica. Es por ello que se hace indispensable el uso de métodos numéricos para obtener soluciones de estos problemas. En particular, la propiedad de conservación de los modelos matemáticos empleados hace óptimo el uso de métodos de volúmenes finitos, que por construcción respetan esta propiedad, frente a los más extendidos métodos en diferencias finitas, empleados en otros campos. No obstante, actualmente existen limitaciones que reducen significativamente la eficiencia de los algoritmos. La restricción más significativa en los métodos numéricos empleados en este campo está asociada a la estabilidad numérica, una propiedad de los mismos que asegura que los errores de redondeo no se amplifican en cada iteración, dando lugar a soluciones incorrectas. Esta condición se traduce en restricciones en el paso de tiempo empleado en la simulación, que en la mayor parte de métodos clásicos tiene carácter global; es decir, todas las celdas avanzan en el tiempo con el mismo incremento temporal. Sin embargo, en muchos casos son únicamente unas pocas celdas las que requieren este paso temporal global tan reducido, lo que ralentiza enormemente la simulación en casi todo el dominio, además de introducir otros artefactos asociados al método numérico como la difusión numérica.

Según la discretización temporal empleada, existen dos grandes clases de métodos numéricos con propiedades diferentes en relación a la estabilidad numérica. Por una parte, los métodos implícitos, a pesar de sus buenas condiciones de estabilidad, requieren un alto coste computacional por iteración y presentan difusión numérica, por lo que sólo son adecuados para problemas estacionarios. En contraposición, los métodos explícitos son fáciles de implementar y paralelizar, motivo por el que son los más implementados en diferentes aplicaciones. Sin embargo, presentan estabilidad condicional, siendo necesario reducir enormemente el paso temporal para su correcto funcionamiento, incrementando el tiempo de cálculo.

Para sobrepasar la condición de estabilidad, se han planteado diferentes modificaciones a los métodos clásicos con el objetivo de aumentar su eficiencia [7–9]. En este trabajo, el objetivo es estudiar los métodos numéricos explícitos basados en un paso de tiempo local (LTS) [1, 6, 12] para la resolución de problemas transitorios en Mecánica de Fluidos, frente a los métodos de paso temporal global (GTS) clásicos. Las tareas necesarias para ello son:

1. Describir el funcionamiento de los métodos LTS y su implementación sobre leyes de conservación.
2. Implementar estos métodos numéricos para resolver los modelos propuestos en programas escritos en lenguaje C.
3. Comparar cuantitativamente el rendimiento de los métodos según diferentes criterios: precisión de los resultados, tiempo de cálculo y preservación de las propiedades de la solución exacta, si la hubiera.

2 Leyes hiperbólicas de conservación

En primer lugar, presentaremos los modelos matemáticos sobre los que se implementarán los métodos numéricos para comparar su rendimiento. Todos ellos corresponden a leyes hiperbólicas de conservación, cuya forma general es:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S}, \tag{2.1}$$

donde $\mathbf{U}$ es el vector de cantidades conservadas, $\mathbf{F}$ es el vector de flujos (en general, dependerá de las cantidades conservadas y la posición) y $\mathbf{S}$ es el vector de términos fuente/sumidero. Se puede probar que las ecuaciones de Euler de la Mecánica de Fluidos admiten esta forma, por lo que es razonable emplear ecuaciones de este tipo, aunque sean más simples, para estudiar el comportamiento de los métodos numéricos.

Un objeto matemático fundamental para el estudio de estos sistemas es el **jacobiano**, definido como

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}, \tag{2.2}$$

cuyos autovalores λ_j están asociados a las velocidades de propagación de la información en cada punto. El carácter hiperbólico del sistema garantiza que son reales y diferentes entre sí, por lo que se puede diagonalizar a través de la matriz de autovectores, $\mathbf{P}$:

$$\mathbf{J} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}, \tag{2.3}$$

con $\mathbf{\Lambda}$ diagonal conteniendo los valores propios de $\mathbf{J}$.

Las leyes hiperbólicas empleadas en este trabajo presentan un grado de complejidad creciente: comenzaremos por la ecuación escalar de transporte lineal, que tiene solución analítica; seguiremos por la ecuación de Burgers 1D no viscosa, que introduce la no linealidad al modelo, dando lugar a fenómenos como ondas de rarefacción y ondas de choque; y por último, las ecuaciones *shallow water* en una dimensión, como ejemplo de sistema de ecuaciones hiperbólico no lineal.

2.1 Ecuación de convección lineal

El caso más sencillo para una ley de conservación del tipo (2.1) es la ecuación escalar de convección lineal, donde u es la cantidad conservada, $f = cu$, $s = 0$ y c es una constante.

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0. \tag{2.4}$$

Se ha introducido la notación en minúsculas para remarcar que es una ecuación escalar. Dada una condición inicial $u(x, 0) = g(x)$, es fácil ver que la solución del problema es:

$$u(x, t) = g(x - ct). \tag{2.5}$$

Es decir, la información se propaga en el plano $x - t$ a través de las rectas $x - ct = cte$, denominadas *rectas características*. Cualitativamente, el resultado es que la condición inicial se desplaza hacia la derecha o izquierda, según el signo de c, a velocidad constante. Una representación usual que permite visualizar la propagación de la información en este tipo de ecuaciones es a través de la representación de las rectas características en el plano $x - t$. En el caso de la ecuación de transporte lineal, podemos ver, para un valor de $c > 0$ cualquiera, dicha representación en la figura 2.1.

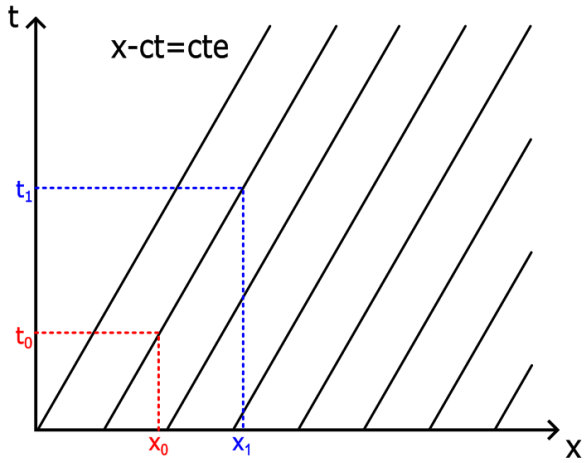


Figura 2.1: Representación en el plano $x-t$ de las rectas características en la ecuación de transporte lineal. El valor de la solución en el punto (x_1, t_1) coincide con el valor en (x_0, t_0) ya que pertenecen a la misma recta característica.

Dado que los puntos (x_0, t_0) y (x_1, t_1) cumplen $x_0 - ct_0 = x_1 - ct_1$, la solución tiene el mismo valor en ellos, $u(x_0, t_0) = u(x_1, t_1)$. Tomando un caso más concreto, pongamos $g(x) = e^{-\frac{(x-2)^2}{8}} + e^{-\frac{(x+2)^2}{4}}$ y $c = 1$. Si observamos en la figura 2.2, en $t = 5$ la condición inicial se desplaza 5 unidades a la derecha en x, y el valor en $x = 5$ es el que tenía la función para $t = 0$ en $x = 0$.

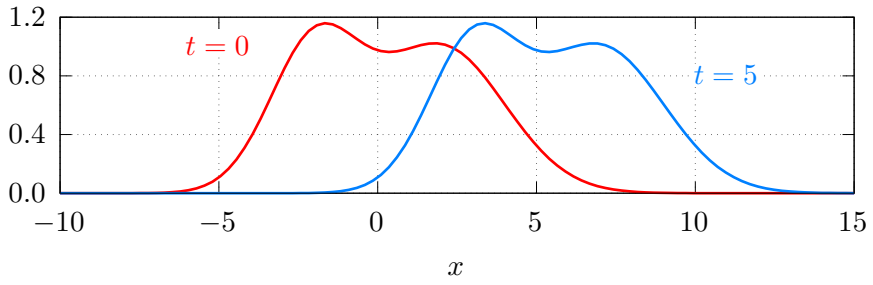


Figura 2.2: Ejemplo de transporte lineal para una distribución inicial

2.2 Ecuación de Burgers no viscosa

La ecuación de Burgers es un ejemplo clásico a la hora de probar nuevos métodos numéricos para este tipo de problemas, ya que introduce de forma sencilla la no linealidad. En este caso, la velocidad

de transporte deja de ser constante, y viene dado por el propio valor de la cantidad conservada u, de modo que el flujo es $f = \frac{1}{2}u^2$, y de nuevo no hay términos fuente. La ecuación es, por tanto:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0. \tag{2.6}$$

Cabe destacar que esta ecuación contiene la no linealidad de la derivada sustancial, generalmente presente en las leyes de conservación de la Mecánica de Fluidos, por lo que tiene un significado físico. Además, con tan sólo una modificación respecto al caso anterior se observan comportamientos cualitativos muy diferentes, como la formación de ondas de choque y ondas de rarefacción al no ser las líneas características paralelas.

Las ondas de choque se forman al converger las líneas características en el plano $x-t$. Esto puede producirse incluso desde una condición inicial regular, como por ejemplo una función gaussiana, para la que en un tiempo finito se observa la formación de la discontinuidad asociada a la onda de choque. La velocidad de propagación S con que se propaga la discontinuidad viene dada por las condiciones de Rankine-Hugoniot [11]:

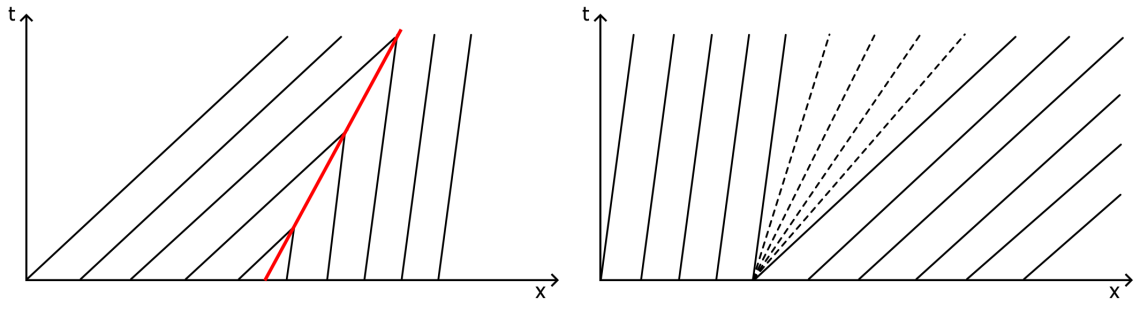
$$f_R - f_L = S(u_R - u_L), \tag{2.7}$$

donde los subíndices L y R denotan los estados a izquierda y derecha de la discontinuidad, respectivamente. En particular, para la ecuación de Burgers, es fácil ver que esta velocidad es la media aritmética entre las velocidades a izquierda y derecha de la discontinuidad:

$$S = \frac{1}{2}\left(u_R + u_L\right). \tag{2.8}$$

En la figura 2.3a se marca la recta asociada a esta velocidad con línea roja.

Las ondas de rarefacción se forman al divergir las líneas características, produciendo un abanico de líneas que recorre todos los valores intermedios entre los extremos. En particular, esto produce una solución continua, aunque provenga de una discontinuidad. El abanico se representa en la figura 2.3b con líneas discontinuas.



(a) Onda de choque (b) Onda de rarefacción

Figura 2.3: Onda de choque y de rarefacción en el plano x-t

2.3 Ecuaciones *shallow water* 1D

Las ecuaciones *shallow water* o de aguas poco profundas son un sistema de dos ecuaciones en una dimensión que se emplea para modelar flujos en lámina libre, tales como canales o ríos. Físicamente, representan la conservación de la masa y la cantidad de movimiento a lo largo del cauce. Existen diferentes formas de presentar estas ecuaciones, y en este trabajo emplearemos la siguiente:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0, \tag{2.9}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A} + gI_1\right) = gA(S_0 - S_f) + gI_2, \tag{2.10}$$

donde A es la sección transversal mojada, Q el caudal y $g = 9.81 m/s^2$ es la aceleración de la gravedad. I_1 es un término de presión hidrostática definida según:

$$I_1 = \int_0^{h(x)} [h(x) - \eta]\, \sigma(x, \eta) d\eta, \tag{2.11}$$

donde h es el nivel de la superficie libre sobre el fondo, z, y $\sigma(x, \eta)$ es la anchura en una posición x y a una altura η dadas. Por último, S_0 es la pendiente del fondo, S_f el término de pendiente de fricción que modelaremos por la fórmula semiempírica de Manning e I_2 es otro término de presión hidrostática que modela el cambio de anchura en la sección transversal:

$$S_0 = -\frac{\partial z}{\partial x}, \tag{2.12}$$

$$S_f = \frac{n^2 Q^2}{A^2 R_h^{4/3}}, \tag{2.13}$$

$$I_2 = \int_0^{h(x)} [h(x) - \eta]\, \frac{\partial \sigma(x, \eta)}{\partial x} d\eta, \tag{2.14}$$

donde n es el coeficiente de rugosidad de Manning, característico de cada superficie, y R_h es el radio hidráulico, definido como el cociente del área transversal mojada entre el perímetro mojado.

Pasando a la notación vectorial de la ecuación (2.1), las cantidades conservadas, flujos y términos fuente son

$$\mathbf{U} = \begin{pmatrix} A \\ Q \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} Q \\ \frac{Q^2}{A} + gI_1 \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ gA(S_0 - S_f) + gI_2 \end{pmatrix}, \tag{2.15}$$

y, empleando que $\frac{\partial I_1}{\partial A} = \frac{A}{b}$, con $b(x) = \sigma(x, h(x))$ la anchura en la superficie, el jacobiano del sistema es,

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{pmatrix} 0 & 1 \\ g\frac{A}{b} - \frac{Q^2}{A^2} & 2\frac{Q}{A} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{pmatrix}, \tag{2.16}$$

donde hemos definido la velocidad media del agua en la sección, $u = Q/A$; y la celeridad de las ondas infinitesimales de lasuperficie, $c = \sqrt{gA/b}$. Su cociente define el número adimensional de

Froude, determinante para este tipo de flujos:

$$Fr = \frac{u}{c} \tag{2.17}$$

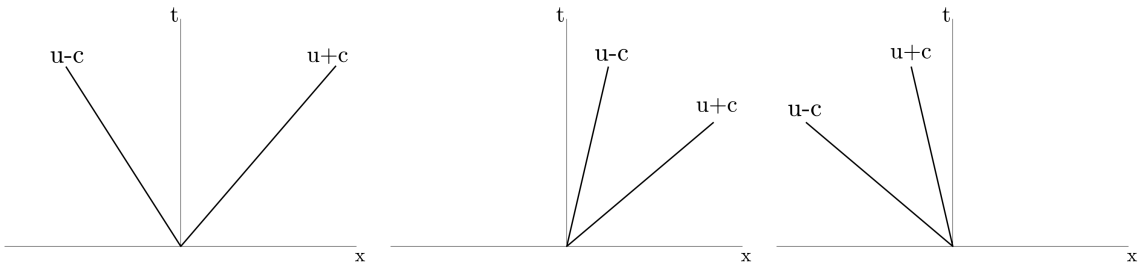
Desarrollando el polinomio característico de (2.16),

$$0 = \lambda(\lambda - 2u) + (u^2 - c^2) = \lambda^2 - 2u\lambda + u^2 - c^2 = (\lambda - u)^2 - c^2. \tag{2.18}$$

Se obtiene que los autovalores del sistema son $\lambda^{1,2} = u \pm c$, que son claramente distintos. Según los valores relativos de $|u|$ y c, se tienen dos condiciones distintas de flujo:

- $Fr < 1$**, flujo subcrítico:** los autovalores tienen signo contrario y hay propagación tanto hacia aguas arriba como aguas abajo.
- $Fr > 1$**, flujo supercrítico:** ambos autovalores tienen el mismo signo y solo se propaga la información en un sentido, que puede ser tanto positivo como negativo.

En la figura 2.4 se representan estas tres posibilidades por medio de las líneas características.



(a) Flujo subcrítico (b) Flujo supercrítico, $u > 0$ (c) Flujo supercrítico, $u < 0$

Figura 2.4: Condiciones de flujo según los valores de las velocidades características

3 Discretización y métodos numéricos

Debido a la complejidad que introduce la no linealidad en las ecuaciones de este tipo de problemas, existen pocos casos en los que se puedan obtener soluciones analíticas para condiciones concretas, que en su mayor parte no tienen aplicación fuera de fines académicos. Es por ello que se hacen indispensables los métodos numéricos para la resolución de las ecuaciones. Para ello, es necesario establecer una discretización tanto temporal como espacial para las ecuaciones a resolver.

3.1 Métodos numéricos en diferencias finitas

En este tipo de métodos numéricos, el dominio se divide en un conjunto finito de N puntos y se aproximan las derivadas por cocientes de incrementos; en particular, por combinaciones lineales de los valores de las variables en puntos del mallado.

- **Discretización espacial:** Podemos aplicar una discretización descentrada hacia delante (ecuación (3.1)) o una centrada (ecuación (3.2)) para la derivada espacial:

$$\frac{\partial u}{\partial x} \approx \frac{u_i^n - u_{i-1}^n}{\Delta x}, \tag{3.1}$$

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}, \tag{3.2}$$

  donde u_i^n es el valor de la variable en el punto x_i del mallado, Δx es la distancia entre puntos adyacentes (supuesta uniforme), evaluado en el nivel temporal t^n.

- **Discretización temporal:** Las derivadas espaciales pueden evaluarse en el nivel temporal actual, t^n, que es conocido, o en el nivel temporal siguiente, t^{n+1}, donde se quiere calcular la solución. Esto da lugar a dos tipos de métodos numéricos: explícitos e implícitos.

Los métodos numéricos **explícitos** calculan directamente los valores de la solución en el siguiente nivel temporal a partir del nivel actual. Por ejemplo, tomando la ecuación de transporte lineal (2.4) y aplicando el método de diferencias centradas a la parte espacial, podemos obtener el valor de la solución en un punto i del mallado en el tiempo t^{n+1} como:

$$0 = \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \tag{3.3}$$

$$\implies u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\frac{u_{i+1}^n - u_{i-1}^n}{2}, \tag{3.4}$$

donde los superíndices denotan el nivel temporal en que se calcula la solución y los subíndices el punto del mallado. Vemos que u_i^{n+1} depende únicamente de los valores en el nivel temporal anterior.

En cambio, los métodos numéricos **implícitos** calculan el siguiente nivel temporal empleando la información de ese mismo tiempo. Esto resulta en la resolución para cada paso temporal de un sistema lineal, al estar acoplados los valores de los diferentes puntos en las ecuaciones. Tomando el mismo caso que el ejemplo anterior:

$$0 = \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} \tag{3.5}$$

$$\implies u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2}. \tag{3.6}$$

Ambos tipos de métodos se pueden emplear para la resolución, sin embargo cada uno presenta ventajas e inconvenientes que son necesarios considerar a la hora de elegir el esquema numérico:

- Los métodos implícitos requieren la resolución de un sistema lineal en cada iteración, lo que supone el uso de algoritmos complejos que, salvo en casos particulares, se vuelven muy costosos computacionalmente al incrementar el número de celdas o nodos.

- Además, suelen presentar difusión numérica, esto es, la tendencia a suavizar las funciones. Esto puede llevar a la pérdida de propiedades de la solución, como en el caso de discontinuidades.

- Los métodos explícitos son menos difusivos y no requieren la resolución de sistemas lineales, pero suelen estar limitados a pasos temporales pequeños para garantizar la estabilidad numérica del método, como veremos en la sección 3.2.

3.2 Condiciones de estabilidad

La estabilidad del método numérico es una propiedad del mismo que asegura que los errores de redondeo producidos en el cálculo no se van a amplificar. Esta propiedad es fundamental, puesto que para cualquier método consistente (es decir, que sea una aproximación de la ecuación diferencial) la estabilidad es equivalente a la convergencia [11].

Un criterio empleado para estudiar la estabilidad es el análisis de von Neumann, que descompone el error numérico en sus componentes de Fourier asociadas a la malla de la discretización espacial y estudia el factor de amplificación de cada una en pasos temporales sucesivos. Si a_j^n es la amplitud asociada al modo del nodo j en el nivel temporal t^n, el método será estable siempre que

$$|G_j| = \left| \frac{a_j^{n+1}}{a_j^n} \right| \leq 1 \quad \forall\, n. \tag{3.7}$$

Si consideramos la ecuación de convección lineal con discretización espacial explícita de diferencias descentradas hacia atrás:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0, \tag{3.8}$$

y sustituimos la componente j del error, $e_j^n = a_j^n e^{i\theta j}$, con θ la fase del sistema, se tiene que

$$(a^{n+1} - a^n) + \frac{c\Delta t}{\Delta x} a^n (1 - e^{-i\theta}) = 0 \implies G = 1 - \frac{c\Delta t}{\Delta x} + \frac{c\Delta t}{\Delta x} e^{-i\theta}. \tag{3.9}$$

Para que $|G| \leq 1$, es necesario que el factor $\nu \equiv \frac{c\Delta t}{\Delta x}$ cumpla $0 \leq \nu \leq 1$. Esto significa que si $c < 0$, este esquema es incondicionalmente inestable y no es capaz de resolver la ecuación de transporte.

Una interpretación física a este resultado es que, si se tiene un sistema con $c < 0$, en el que las ondas de información se propagan de derecha a izquierda, no se puede emplear un método que calcule las variaciones con la información desde la izquierda, pues es contrario al sentido de propagación. En el caso de haber tomado una discretización de diferencias descentradas hacia delante, dada por

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_i}{\Delta x}, \tag{3.10}$$

hubiéramos llegado al resultado opuesto: para $c > 0$ el esquema numérico es inestable. Un método numérico que tiene en cuenta el sentido de propagación es el *upwind*, que en la ecuación de convección lineal se puede expresar como:

$$u_i^{n+1} = u_i^n - \frac{c^+\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) - \frac{c^-\Delta t}{\Delta x}(u_{i+1}^n - u_i^n), \tag{3.11}$$

donde $c^+ = \max\{c, 0\}, c^- = \min\{c, 0\}$ son las partes positiva y negativa de c, respectivamente. Por

tanto, este método no está restringido a un signo de la velocidad de transporte, y será estable si

$$\frac{|c|\Delta t}{\Delta x} \leq 1, \tag{3.12}$$

o, equivalentemente, el paso temporal empleado debe cumplir $\Delta t \leq \frac{\Delta x}{|c|}$, con Δx el espaciado del mallado empleado, supuesto uniforme. En el apéndice A se desarrolla en más profundidad la base de este método y se aplica a otros esquemas numéricos. En particular, se demuestra que el método de diferencias finitas centradas es inestable en la ecuación de transporte, el método *upwind* explícito es condicionalmente estable si cumple la condición dada por la ecuación (3.12), y el método *upwind* implícito es incondicionalmente estable.

Desde el punto de vista de las líneas características, esta condición garantiza que ninguna de ellas recorra más distancia que el espaciado de la malla en un solo paso de tiempo; en el caso de los métodos de volúmenes finitos, en los que se toman las líneas características desde las fronteras de las celdas, es equivalente a que la información no salga de las celdas adyacentes. Intuitivamente, se puede ver que esto está conectado con el hecho de que en la discretización se está tomando únicamente el valor en las celdas adyacentes para calcular el siguiente paso temporal, por lo que la información que viaja entre ellas debería quedar confinada en esa misma región en cada paso temporal.

Podemos generalizar este resultado para el caso en que la velocidad de propagación de la información, λ_i, no sea uniforme para todas las celdas, sino que dependa de las variables conservadas. Este es el caso de la ecuación de Burgers, en la que $\lambda_i = u_i$, o en las ecuaciones *shallow water*, donde $\lambda_i^{1,2} = u_i \pm c_i$. Con estas velocidades de propagación, podemos definir un paso temporal local, $\Delta t_i = \Delta x/|\lambda_i|$, que establece la condición de estabilidad del método explícito a nivel local para cada celda. Para asegurar la estabilidad global, el paso temporal empleado no debe ser mayor al de cada celda, por lo que se puede tomar el mínimo de todos ellos, modulado según el número CFL, que debe estar entre 0 y 1:

$$\Delta t = \mathrm{CFL} \cdot \min_i \Delta t_i. \tag{3.13}$$

Los métodos clásicos emplean un paso temporal global para todo el dominio que, para garantizar la estabilidad, viene dado por el menor de los pasos temporales locales, asociados a las celdas. En general, dado que cada una va a tener valores dispares de velocidades de propagación, la mayor parte de ellas va a emplear un paso temporal en la integración menor del que le permite la condición de estabilidad a nivel local, ya que en la mayor parte de problemas son únicamente unas pocas celdas las que requieren un paso temporal tan reducido. Esta necesidad de limitar el paso temporal en un gran número de celdas para garantizar la estabilidad global es el principal inconveniente de los métodos explícitos que, pese a que cada iteración sea poco costosa computacionalmente comparado con los métodos implícitos, requiere de un gran número de ellas para poder alcanzar los tiempos de simulación requeridos, pues en general presentan estabilidad condicional. Es por ello que se plantearon métodos de paso temporal local, que buscan aplicar en cada celda el mayor paso temporal posible para acelerar la simulación evitando dar más pasos de los necesarios.

3.3 Métodos numéricos de volúmenes finitos

A partir de una partición del dominio en N celdas se aproxima la ecuación integral, obtenida integrando en cada celda de anchura Δx, definida en el segmento $\left[x_{i-1/2}, x_{i+1/2}\right]$ (donde $i \pm 1/2$ denotan las paredes derecha e izquierda de la celda, respectivamente), y en un intervalo temporal $[0, \Delta t]$. Por ejemplo, para una ecuación escalar sin términos fuente, la ecuación integral a aproximar es:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \int_0^{\Delta t} \frac{\partial u}{\partial t} dt dx + \int_{x_{i-1/2}}^{x_{i+1/2}} \int_0^{\Delta t} \frac{\partial f}{\partial x} dt dx = 0. \tag{3.14}$$

Los diferentes métodos se obtienen según la forma de aproximar las integrales. Estos son los métodos comúnmente empleados en este campo por mantener la propiedad de conservación. En particular, podemos construir el método de Roe generalizando a un sistema de ecuaciones de este tipo en una dimensión,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S}, \tag{3.15}$$

y considerando la integral de la misma sobre la celda i-ésima del dominio, definida en el segmento $\left[x_{i-1/2}, x_{i+1/2}\right]$, de longitud Δx; y sobre un intervalo temporal $\left[t^n, t^{n+1}\right]$, de anchura Δt:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} \frac{\partial \mathbf{U}}{\partial t} dt dx + \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} \frac{\partial \mathbf{F}}{\partial x} dt dx = \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} \mathbf{S} dt dx. \tag{3.16}$$

Aproximando las variables conservadas por constantes en toda la celda, y los términos fuente como independientes del tiempo, se tiene que, empleando la regla de Barrow,

$$\Delta x(\mathbf{U}_i^{n+1} - \mathbf{U}_i^n) + \Delta t(\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}) = \Delta t \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{S} dx. \tag{3.17}$$

Los valores $\mathbf{F}_{i\pm 1/2}$ se denominan flujo numérico en la pared correspondiente. Aproximando la integral del término fuente por un valor promedio sobre el volumen, $\Delta x\, \mathbf{S}_i$, se tiene entonces que

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x} \left[\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}\right] + \Delta t\, \mathbf{S}_i. \tag{3.18}$$

Físicamente, la interpretación de esta ecuación de actualización de las celdas es que la variación de las cantidades conservadas se debe, por una parte, a los flujos en las paredes de la misma, y por otra a los términos fuente que actúan en la celda. Cabe señalar que los flujos se cancelan dos a dos en cada pared, de forma que la variación total de las variables conservadas en todo el dominio viene dada por los contornos y los términos fuente. Existen múltiples formas de modelar estos últimos; en este trabajo, el único caso en que los empleamos es en las ecuaciones *shallow water*, y se ha seguido como referencia [10] para su tratamiento.

A partir de esta formulación, podemos separar la diferencia de flujos numéricos entre ambas paredes como una suma de contribuciones de cada pared. Esto corresponde a una descripción del método como ondas que se propagan por el dominio, transportando información con la filosofía

upwind para actualizar las celdas. Matemáticamente,

$$\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2} = \delta\mathbf{F}^{-}_{i+1/2} + \delta\mathbf{F}^{+}_{i-1/2}. \tag{3.19}$$

Los superíndices $\pm$ indican que la pared correspondiente únicamente contribuye a la actualización de la celda si la propagación de información es en el sentido indicado. En base a esto, se puede aplicar una linealización a los valores $\delta\mathbf{F}_{i+1/2}$ empleando un jacobiano aproximado en la pared, de tal forma que

$$\delta\mathbf{F}_{i+1/2} = \tilde{\mathbf{J}}_{i+1/2}\delta\mathbf{U}_{i+1/2}, \tag{3.20}$$

donde la tilde indica que es una aproximación que emplea únicamente los valores a cada lado de la pared para su construcción. Este jacobiano puede diagonalizarse según una matriz $\tilde{\mathbf{P}}$ y su inversa, donde las columnas de $\tilde{\mathbf{P}}$ forman un base de autovectores $\tilde{\mathbf{e}}_j$ asociados a cada autovalor aproximado $\tilde{\lambda}_j$. Descomponiendo $\delta\mathbf{U}_{i+1/2}$ en la base de autovectores según unos coeficientes α_j, se tiene que la ecuación (3.20) se puede expresar como

$$\delta\mathbf{F}_{i+1/2} = \sum_j \tilde{\lambda}_j\alpha_j\tilde{\mathbf{e}}_j\Big|_{i+1/2}, \tag{3.21}$$

y sustituyendo en (3.18), se obtiene el método de Roe:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left[\sum_j \tilde{\lambda}_j^{-}\alpha_j\tilde{\mathbf{e}}_j\Big|_{i+1/2} + \sum_j \tilde{\lambda}_j^{+}\alpha_j\tilde{\mathbf{e}}_j\Big|_{i-1/2}\right] + \Delta t\mathbf{S}_i. \tag{3.22}$$

Este método muestra de forma explícita la naturaleza física de la propagación de la información en medios continuos modelados según leyes hiperbólicas de conservación a través de las velocidades λ_j, que sólo contribuyen a la actualización de las celdas si su sentido de propagación es hacia la celda considerada. El término fuente también puede introducirse en la formulación *upwind* descomponiéndolo sobre los vectores de la base, según lo propuesto en [5].

Una característica fundamental de este método numérico es que el paso temporal Δt, tomado como el más restrictivo, tiene carácter **global**, en el sentido de que todas las celdas emplean este valor para avanzar temporalmente en la simulación. Por este motivo, denominaremos a este método *Global Time Step* (GTS) en lo sucesivo, en contraposición a los métodos de paso temporal local, que buscan mejorar la eficiencia de los GTS sorteando la condición de estabilidad con diferentes estrategias.

4 Métodos *Local Time Step*

Los métodos explícitos *Local Time Step* (LTS) buscan mejorar la eficiencia de los métodos convencionales empleando el mayor paso temporal posible para cada celda dado por la condición de estabilidad. El objetivo en esta sección es introducir dos métodos de este tipo según su formulación en [1–3].

A pesar de tener en común esta filosofía, la implementación de la misma es bastante diferente. En el primer caso (LTS1), en cada iteración no se recalculan los valores de flujo para aquellas celdas que todavía estén dentro de la condición de estabilidad para el último paso temporal en que se haya calculado este valor, aunque la integración se produce de forma sincronizada en todas las celdas (en cada paso temporal se actualizan todas ellas). En cambio, el algoritmo LTS2 integra de forma independiente cada celda hasta un nivel temporal lo más cercano posible al establecido por la condición de estabilidad, trabajando así de forma asíncrona sobre cada una (en cada paso temporal no tienen por qué actualizarse todas las celdas).

Cabe destacar que este procedimiento no es el único para sobrepasar la condición de estabilidad local con el fin de mejorar el rendimiento. Otro tipo de algoritmos son los llamados métodos de *large time step*, que permiten el uso de CFL > 1 al enviar información más allá de las celdas adyacentes a la pared. Esta posibilidad ha sido explorada en [7–9].

4.1 LTS1: *Frozen Flux*

Este algoritmo, propuesto en [12], plantea la implementación del *local time stepping* recalculando el valor de actualización de la celda después de uno o más pasos temporales, según la condición de estabilidad local. De esta forma, el objetivo es acelerar el algoritmo reduciendo el número de cálculos por iteración, que es en general la parte más costosa computacionalmente.

Para pasar del nivel temporal t^n al siguiente, las variables conservadas se actualizan según

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\delta\mathbf{F}^-_{i+1/2} + \delta\mathbf{F}^+_{i-1/2}\right). \tag{4.1}$$

Si suponemos que la condición de estabilidad local en la celda i permite alcanzar un cierto paso temporal $k\Delta t$, con k entero, se podría actualizar la celda los siguientes $k-1$ niveles temporales con el mismo valor de flujo en las paredes sin provocar error numérico de tal forma que podemos expresar el valor en cada nivel $j = 1, \ldots, k$ según

$$\mathbf{U}_i^{n+j} = \mathbf{U}_i^{n+j-1} - \frac{\Delta t}{\Delta x}\left(\delta\mathbf{F}^-_{i+1/2} + \delta\mathbf{F}^+_{i-1/2}\right) = \mathbf{U}_i^{n} - \frac{j\Delta t}{\Delta x}\left(\delta\mathbf{F}^-_{i+1/2} + \delta\mathbf{F}^+_{i-1/2}\right). \tag{4.2}$$

De esta forma, podemos emplear el valor del paréntesis calculado en el nivel temporal t^n durante los k siguientes pasos, sin necesidad de reevaluarlo en cada una; es decir, el flujo permanece *congelado* mientras lo permita la estabilidad local. Para aquellas celdas cuyo paso temporal local sea cercano al paso global, será necesario calcular este valor en cada paso, pero en general estas serán tan solo unas pocas, lo que garantiza, a priori, la eficiencia del algoritmo.

Para una descripción del funcionamiento, nos apoyaremos en la figura 4.1. En primer lugar,

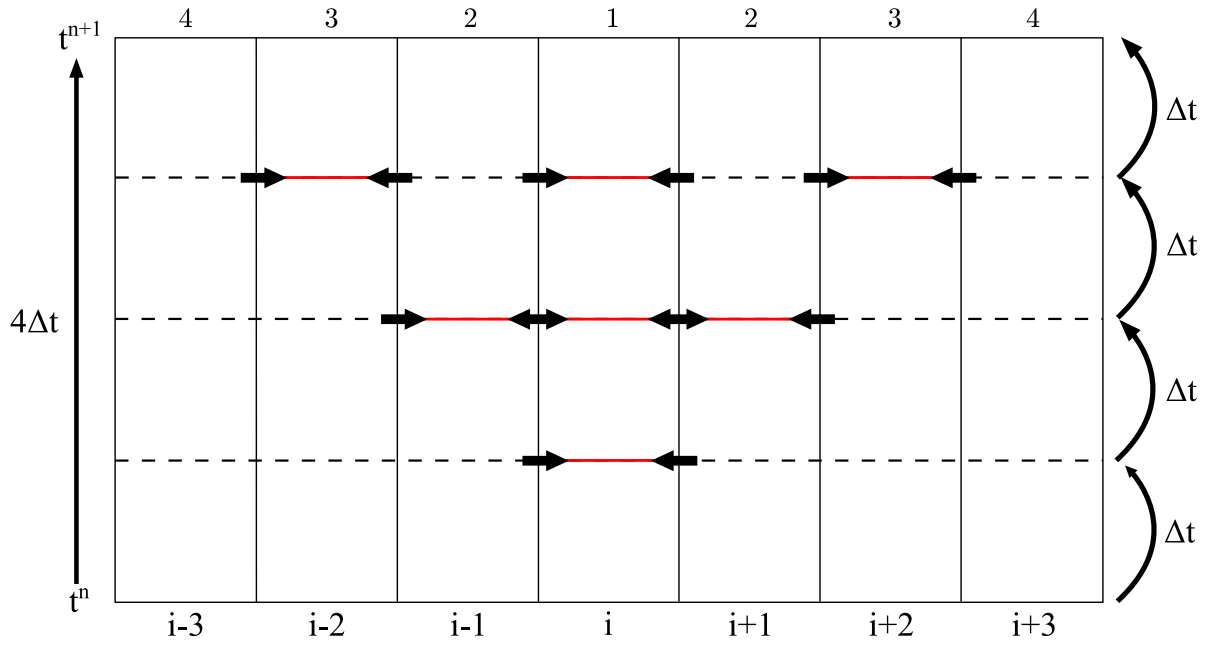


Figura 4.1: Esquema del funcionamiento del algoritmo LTS1. Las celdas adyacentes a la i-ésima avanzan de t^n a t^{n+1} mediante 4 pasos temporales locales. Los números sobre las celdas indican el valor de k_j de cada una, y las líneas rojas el máximo tiempo que puede alcanzar el valor de flujo calculado anteriormente para cada una. Las flechas en las paredes entre celdas indican el cálculo de la contribución de cada una a la celda en los pasos temporales que correspondan.

para cada celda se toma el paso temporal local dado por la condición de estabilidad:

$$\Delta t_i = \frac{\Delta x_i}{\lambda_i}, \tag{4.3}$$

donde Δx_i es la anchura de la celda, que en general será no uniforme, y λ_i es el módulo del mayor de los autovalores del jacobiano evaluado en la celda. Para garantizar la estabilidad local, elegimos como paso temporal global Δt el menor de todos ellos, modulado por el número CFL:

$$\Delta t = \mathrm{CFL} \cdot \min_i \Delta t_i. \tag{4.4}$$

A continuación, asignamos a cada celda un valor entero de resolución temporal, k_i, de tal forma que $k_i \Delta t \leq \Delta t_i < (k_i + 1)\Delta t$, luego podemos tomar

$$k_i = \left\lfloor \frac{\Delta t_i}{\Delta t} \right\rfloor. \tag{4.5}$$

Este valor nos permite definir un nuevo paso temporal local $\Delta t_i^* = k_i \Delta t$ que verifica la condición de estabilidad local y corresponde al intervalo de tiempo en que el algoritmo va a mantener el valor de flujo congelado en esa celda, en su caso. Estos son los valores situados encima de cada celda en la figura 4.1, que en este caso han sido asignados arbitrariamente para mostrar el funcionamiento.

Tomamos $k = \max_i k_i$ el mayor valor de los asignados, que corresponde con el tiempo máximo que puede avanzar el algoritmo desde la primera evaluación de flujos en todas las celdas. De esta forma, el paso global terminará tras $k\Delta t$, y el algoritmo se reiniciará desde ese punto.

El último paso en la inicialización del algoritmo es la asignación de las celdas a dos grupos, $G1$ y $G2$, según si admiten mantener el valor de flujo congelado durante varias iteraciones o necesita actualizarse. El criterio empleado es

$$\begin{cases} k_i \leq 2 \implies i \in G1, \\ k_i > 2 \implies i \in G2. \end{cases} \tag{4.6}$$

La inclusión de las celdas con $k_i = 2$ en $G1$ es necesaria para garantizar el buen funcionamiento, ya que a priori no sería necesario incluirlas en este grupo. Sin embargo, en varias pruebas sencillas de propagación en la ecuación de convección lineal se vio que era necesario que actualizaran el valor de flujo en cada iteración para evitar las inestabilidades numéricas.

Por último, es necesario considerar las fronteras entre bloques de celdas que pertenezcan a distinto grupo para asegurar que la información se propaga correctamente de uno al otro. Para ello, se reasignan las dos primeras celdas del grupo $G2$ en la frontera al grupo $G1$, lo que en los casos prácticos se probó suficiente.

Después de esta inicialización, se procede a la actualización de las celdas según la ecuación (4.2). Cabe destacar que, aunque en el desarrollo hayamos tomado Δt constante por claridad, tras cada actualización es necesario recalcular este valor según los nuevos valores autovalores del jacobiano del sistema, pues es posible que este paso temporal se vuelva demasiado grande para garantizar la estabilidad local en la siguiente iteración si cambian las condiciones del flujo. Además, también es necesario reevaluar si las celdas en $G2$ pueden mantenerse en este grupo o tienen que pasar al otro, lo que también se realiza en cada iteración.

Una vez se alcanza el siguiente nivel temporal global, dado por $t^{n+1} = t^n + k\Delta t$, finaliza el algoritmo y se reinicializa con los valores calculados en este punto.

4.2 LTS2: *Full Integration*

El segundo algoritmo de *local time stepping* implementado en este trabajo fue propuesto en [6]. De nuevo, tomaremos la formulación mostrada en [1–3], y el trabajo de S. Dazzi en [4], cuyo pseudocódigo de los algoritmos ayudó para implementarlo correctamente, aunque su aplicación sea para un modelo más avanzado.

El funcionamiento es similar al caso anterior, asignando a cada celda un nivel temporal según el paso temporal permitido, aunque en este caso se toman en general más de dos. El primer paso al iniciarse el algoritmo es calcular el paso temporal local y establecer el paso global como el mínimo

de entre todas las celdas según la ecuación (4.4). Con estos valores, se asigna a cada celda un valor entero m_i de resolución temporal, que en este caso se toma sobre potencias de 2, de tal forma que

$$2^{m_i}\Delta t \leq \Delta t_i < 2^{m_i+1}\Delta t. \tag{4.7}$$

Por tanto, podemos obtener m_i según

$$m_i = \left\lfloor \frac{\log\left(\frac{\Delta t_i}{\Delta t}\right)}{\log 2} \right\rfloor. \tag{4.8}$$

Definimos $M = \max_i m_i$, que corresponde al mayor nivel temporal alcanzado, y al que está asociado el tiempo máximo que puede alcanzar el algoritmo. A diferencia del LTS1, la integración de las celdas no se realiza de forma simultánea, sino que cada una avanza, cada vez que se integra sobre ella, un intervalo temporal dado por $2^{m_i}\Delta t$. Por tanto, la siguiente iteración del algoritmo será en $t^{n+1} = t^n + 2^M\Delta t$.

La integración se realiza a través de 2^M barridos, donde para cada barrido $p = 0, \ldots, 2^M - 1$ se actualizan las celdas tales que p sea un múltiplo entero de 2^{m_i}. Por ejemplo, para una celda con $m_i = 0$, su valor se actualizará en todos los barridos con un paso temporal Δt, mientras que una celda con $m_i = M$ sólo se integrará en el primer barrido con un paso temporal $2^M\Delta t$ y permanecerá a la espera hasta que finalice el proceso. Un ejemplo más detallado puede verse en la figura 4.2.

En este algoritmo, el valor Δt permanece invariante durante todo este proceso, por lo que es posible que haya algún momento durante la integración en que sea demasiado grande y se produzca inestabilidad en algunas celdas. Por ello, en algunos casos será necesario limitar el valor máximo de m_i, con el objetivo de garantizar la estabilidad.

Por último, respecto a las fronteras entre grupos de celdas con diferentes valores de m_i, se ha seguido el mecanismo propuesto en [2] para garantizar una correcta propagación de la información de una a otra. El proceso empleado consiste en reasignar los valores de m en las celdas con mayor valor de forma progresiva, de tal manera que la máxima diferencia entre celdas adyacentes no sea mayor que 1.

5 Resultados

En esta sección, presentaremos los resultados obtenidos en diferentes casos test para cada ecuación. Es importante señalar que el principal objetivo es reducir el tiempo computacional de los métodos clásicos de *global time step*, o GTS (en nuestro caso, el método de Roe), con la menor pérdida de precisión posible. Para la cuantificación del rendimiento de cada algoritmo, tomaremos como parámetro de control fundamental el tiempo de cálculo relativo al GTS, recogido en la figura 5.1 para todos los casos test. Este se obtiene dividiendo el tiempo de cálculo del algoritmo LTS correspondiente entre el tiempo de cálculo del método GTS, por lo que se obtiene una ganancia siempre

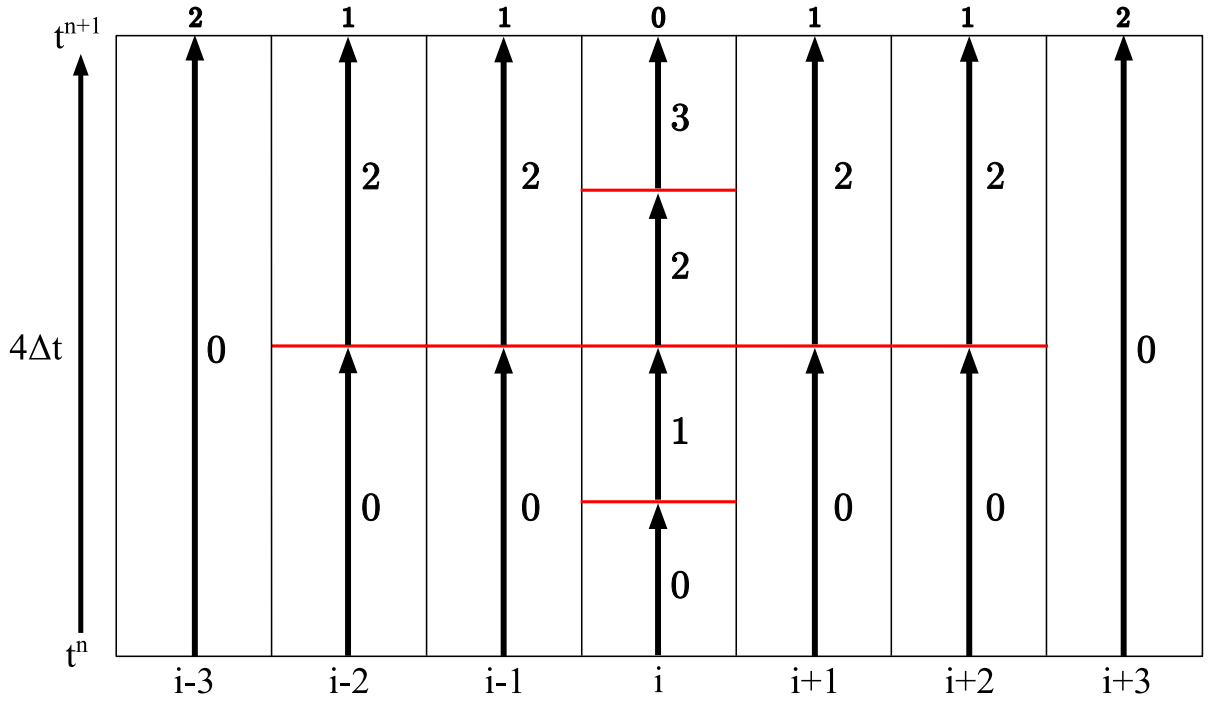


Figura 4.2: Esquema del funcionamiento del algoritmo LTS2. Las celdas adyacentes a la i-ésima avanzan de t^n a t^{n+1} con 4 barridos. Los números superiores indican el valor de m_j en cada celda y los que acompañan a cada flecha el número de barrido en que se realiza la integración.

que el resultado sea menor que el 100%. Las normas de error L^1, L^2 y L^∞, que se muestran en las tablas 5.1 y 5.3, se han calculado respecto a las soluciones exactas y permiten evaluar la precisión de los métodos, para comprobar que la ganancia en tiempo de cálculo no es a costa de empeorar la precisión de la solución obtenida.

5.1 Ecuación de convección lineal

Comenzamos mostrando los resultados para el caso más sencillo. Para poder analizar el rendimiento de los algoritmos, es necesario el uso de una malla irregular para la discretización espacial del dominio, pues en una regular, dado que la velocidad de propagación $\lambda_i \equiv c$ es constante en todas las celdas, se tendría que el paso local, dado por la ecuación (4.3), es el mismo para todas ellas. Por tanto, tanto el LTS1 como el LTS2 operarían como el método GTS, pero ralentizados por la lógica de control implementada en el algoritmo.

Además, para forzar a la existencia de una gran variedad de pasos temporales locales, tomamos una malla con anchura aleatoria según una distribución plana. Para que el tiempo de cálculo sea relevante, fijamos el número de celdas en $N = 10000$. Los resultados se representan en la figura 5.2 para una condición inicial gaussiana centrada en $x = 40$ (CL1) y otra cuadrada de anchura 30, e inicialmente entre $x = 25$ y $x = 55$ (CL2).

Se puede observar en la figura 5.2a que para la condición inicial CL1, los tres métodos (GTS, LTS1 y LTS2) resultan indistinguibles de la solución exacta, obtenida a través de la solución general

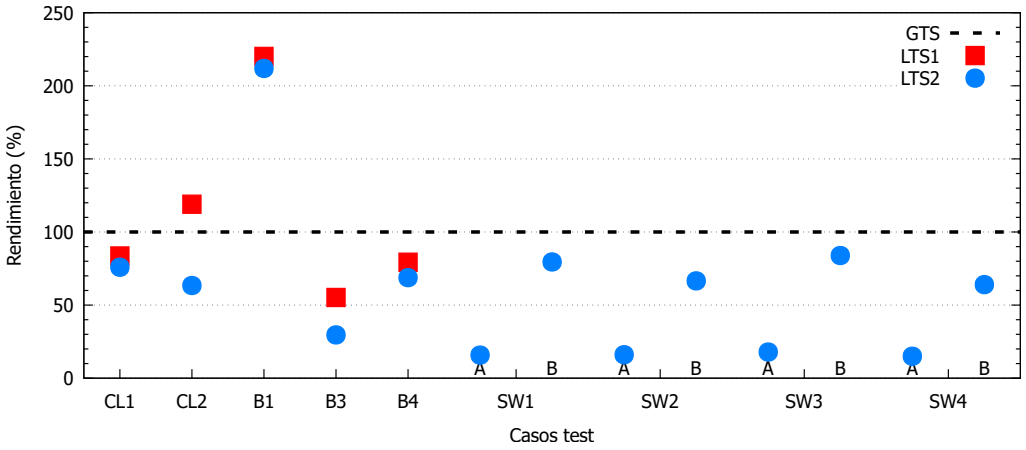


Figura 5.1: Rendimiento relativo respecto al algoritmo GTS para cada caso test

		CL1	CL2	B1	B2	B3	B4
L_1	GTS	0.162	2.896	0.0390	0.0000	0.1912	0.2623
	LTS1	0.036	1.206	0.0299	0.0000	0.0972	0.1471
	LTS2	0.282	3.156	0.0355	0.0000	0.1764	0.1245
L_2	GTS	0.035	2.059	0.0279	0.0000	0.0999	0.1886
	LTS1	0.008	1.332	0.0271	0.0000	0.0904	0.1818
	LTS2	0.069	2.025	0.0421	0.0000	0.0912	0.1862
L_∞	GTS	0.014	2.501	0.2963	0.0000	0.6500	1.603
	LTS1	0.004	2.506	0.2963	0.0000	0.6500	1.603
	LTS2	0.030	2.589	0.4646	0.0000	0.5907	1.630

Tabla 5.1: Normas de error para los diferentes algoritmos y casos test. Convección lineal y ecuación de Burgers

dada por la ecuación (2.5). Para la condición CL2, en la figura 5.2b existe un ligero efecto de difusión en los puntos de discontinuidad, aunque es similar para los tres esquemas. En ambos casos, la precisión de los algoritmos es similar a la ofrecida por el método global, como se puede ver en la tabla 5.1 para los casos CL1 y CL2. Sin embargo, en cuanto al tiempo de cálculo (ver figura 5.1), vemos que el LTS1 es más eficiente que el GTS para la condición inicial gaussiana pero no para la onda cuadrada. En contraposición, el LTS2 no muestra problemas con ninguna de ellas y se desmarca como la opción más eficiente en este aspecto.

A nivel de código, ambas condiciones iniciales se ejecutaron sobre el mismo programa de forma consecutiva, luego es razonable descartar un fallo en la implementación. Es posible que la causa de este problema se deba a un mal tratamiento por parte del algoritmo de la discontinuidad; sin embargo, en este caso los valores que toma la variable conservada no juegan un papel relevante para el funcionamiento del mismo, a diferencia de las ecuaciones siguientes en las que sí se emplea u para calcular las velocidades de propagación.

5.2 Ecuación de Burgers no viscosa

En el caso de la ecuación de Burgers, la no linealidad da lugar a diferentes velocidades de propagación en cada punto según el valor de la variable conservada u. Por ello, tomaremos en primer lugar una malla regular y después dos irregulares, A y B, construidas según el procedimiento detallado en el apéndice B. La primera es una malla regular salvo una celda central, de anchura 1000 veces menor al resto. Esta celda es la que gobierna el paso temporal global independientemente de las condiciones del flujo, permitiendo explorar el máximo rendimiento de los algoritmos LTS. La segunda es una malla irregular con un grupo de $N/10$ celdas centrales de tamaño fijo (pequeño) y que progresivamente se van haciendo más grandes según un factor multiplicativo fijo. Las celdas de los extremos son 128 veces mayores que las interiores.

El primer caso test (B1) emplea una malla regular de $N = 25000$ celdas con una condición inicial cuadrada dada por:

$$u_1(x) = \begin{cases} -1, & x < 30, \\ 3, & 30 < x < 55, \\ -1, & x > 55. \end{cases} \tag{5.1}$$

Esta condición inicial da lugar a la formación de una onda de rarefacción en la parte izquierda y una onda de choque en la parte derecha, pudiendo verse simultáneamente el comportamiento de los algoritmos en ambos casos. Dado que la condición inicial toma valores positivos y negativos, es necesario introducir una corrección de entropía para evitar soluciones no físicas. En este caso, se ha tomado la corrección de entropía de Harten-Hyman, formulada siguiendo [11].

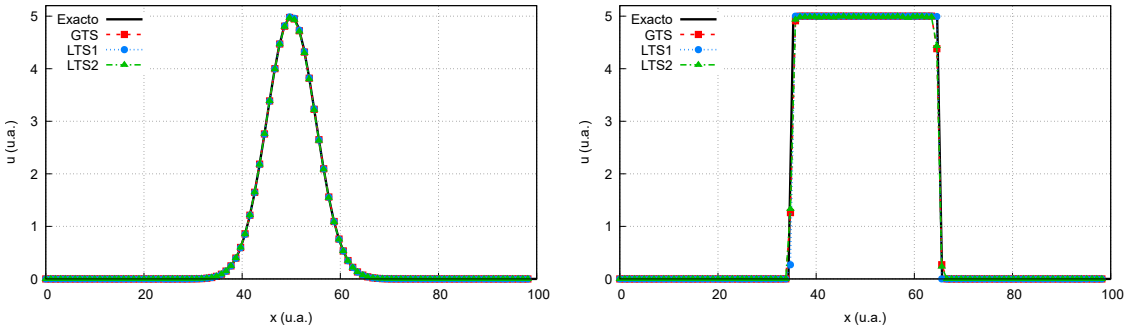
El segundo caso test (B2) también emplea una malla regular de $N = 100$ celdas y la condición inicial dada por:

$$u_2(x) = \begin{cases} 3, & x < 50, \\ -3, & x > 50. \end{cases} \tag{5.2}$$

Es fácil ver que esta condición inicial es estacionaria, pues ${u_2}^2$ es constante. El objetivo de este caso test es comprobar si los algoritmos conservan esta propiedad, lo que es esencial para comprobar la buena implementación de los mismos. Los casos test B3 y B4 emplean mallas irregulares A y B, respectivamente, junto con la condición inicial dada por 5.1 y $N = 25000$ celdas. En ambos casos, la no regularidad debería favorecer el rendimiento tanto de LTS1 como de LTS2 frente al GTS.

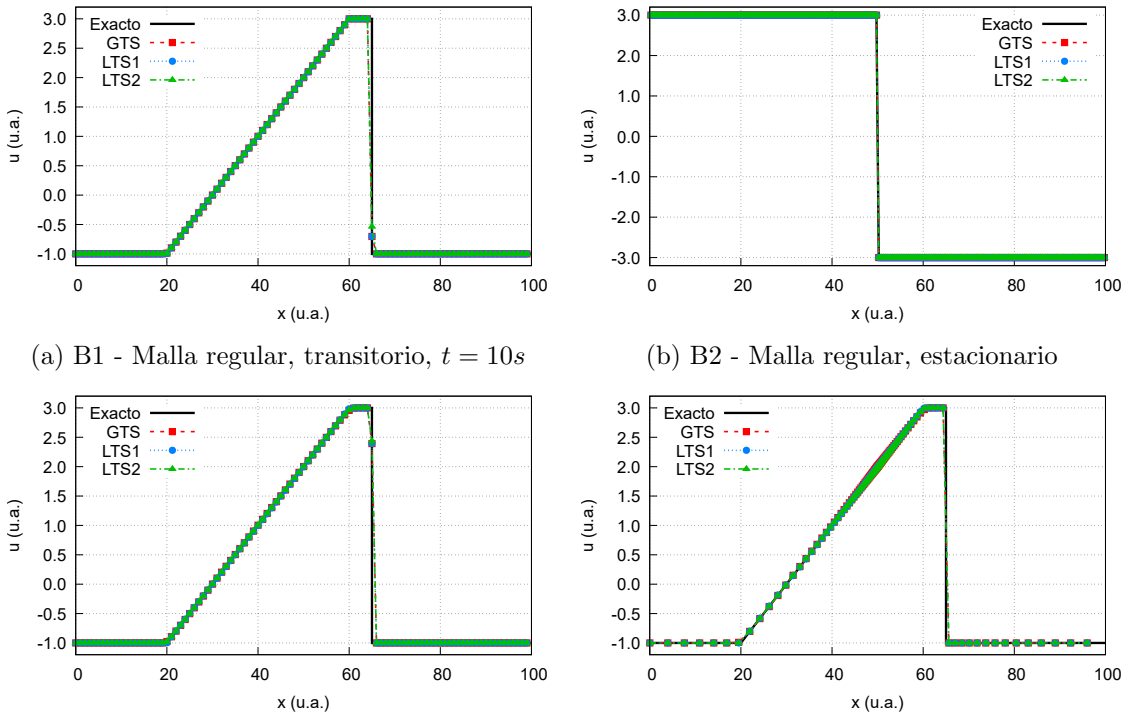
En primer lugar, vemos que la condición inicial estacionaria mostrada en 5.3b no cambia a lo largo del tiempo, por lo que todos los esquemas la resuelven correctamente, como se puede comprobar en las normas de error de la tabla 5.1. Si comparamos los resultados para el problema transitorio entre las diferentes mallas, las soluciones mostradas en las figuras 5.3a, c y d resultan totalmente indistinguibles entre sí y de la solución exacta, obtenida según lo propuesto en [10]. Atendiendo a las normas de error en la tabla 5.1, todas ellas presentan una precisión similar, aunque en el caso de los tiempos de cálculo se presentan más discrepancias que en la ecuación de transporte lineal.

Comparando los resultados de la figura 5.1, vemos que como cabría esperar, en la malla regular



(a) CL1 - Condición inicial gaussiana (b) CL2 - Condición inicial cuadrada

Figura 5.2: Resultados para la ecuación de transporte en malla irregular aleatoria, $c = 1$, $t = 10$. Se muestra 1 de cada 100 puntos por claridad.



(a) B1 - Malla regular, transitorio, $t = 10s$ (b) B2 - Malla regular, estacionario

(c) B3 - Malla irregular A, transitorio, $t = 10s$ (d) B4 - Malla irregular B, transitorio, $t = 10s$

Figura 5.3: Resultados para la ecuación de Burgers en diferentes mallas. Se muestra 1 de cada 250 puntos por claridad en los casos transitorios.

Caso test	h_L	h_R	q_L	q_R	z_L	z_R
SW1	1.0	0.30179953	0.00	0.0	0.0	0.05
SW2	4.0	0.50537954	0.00	0.0	0.0	1.50
SW3	2.5	2.49977310	3.75	0.0	0.0	0.25
SW4	1.5	0.16664757	3.00	0.0	0.0	2.00

Tabla 5.2: Valores iniciales para los casos test de rotura de presa

los algoritmos LTS son menos eficientes que el GTS pues los valores de tiempo local no son suficientemente dispares con la condición inicial empleada. Sin embargo, en las mallas irregulares se aprecian aceleraciones en el tiempo de cálculo notables, siendo mayor para el LTS2 que el LTS1. La malla A magnifica este factor de aceleración ya que casi todas las celdas permiten un paso mucho mayor que el mínimo, mientras que la malla B da una idea más realista de los valores que se pueden alcanzar con estos algoritmos.

5.3 Ecuaciones *shallow water* 1D

Para estudiar los algoritmos en las ecuaciones *shallow water*, emplearemos un canal prismático con sección rectangular, de tal forma que la anchura en cualquier punto de la longitud y a cualquier altura es constante, b. De esta forma, podemos dividir las ecuaciones (2.9) y (2.10) por este valor para trabajar con el calado $h = A/b$ y el caudal por unidad de profundidad $q = Q/b$. Esta es la geometría más simple posible, pero es suficiente en primera instancia para estudiar los métodos numéricos.

Comparando los resultados obtenidos hasta este momento en las ecuaciones más sencillas, se considera razonable implementar únicamente el *solver* LTS2 para las ecuaciones de aguas poco profundas. Los motivos para esta elección son:

1. El algoritmo LTS2 es más eficiente que el LTS1, ya que en todos los casos probados reduce más el tiempo de cálculo, como se puede ver en la figura 5.1, para todos los casos transitorios (CL1, CL2, B1, B3, B4).

2. Aunque la precisión del LTS1 sea algo mejor que la ofrecida por el LTS2, éste es únicamente un parámetro de control para verificar que la solución obtenida no es, en ningún caso, peor que la ofrecida por el GTS. Todos ellos son métodos de primer orden, por lo que el error en la solución tiende a 0 al reducir el tamaño de la malla como $\mathcal{O}(\Delta x)$, y si se quisiera mejorar la precisión es más conveniente recurrir a esquemas numéricos de órdenes superiores.

3. La implementación del LTS2 es, en la experiencia de los sistemas anteriores, más sencilla de generalizar a casos más complejos. Además, la detección y resolución de los errores encontrados a lo largo del desarrollo del código es más directa que en el LTS1, donde la implementación de la lógica de control de las diferentes subetapas que lo componen es menos clara en base a la bibliografía consultada para su desarrollo.

Los casos test propuestos en esta sección corresponden al problema de rotura de presa con diferentes condiciones iniciales. Este viene dado por condiciones iniciales discontinuas del siguiente tipo:

$$h(x) = \begin{cases} h_L, & x < 0, \\ h_R, & x > 0. \end{cases} \quad , \quad q(x) = \begin{cases} q_L, & x < 0, \\ q_R, & x > 0. \end{cases} \quad , \quad z(x) = \begin{cases} z_L, & x < 0, \\ z_R, & x > 0. \end{cases} \tag{5.3}$$

En particular, se han tomado los casos propuestos en [10], donde además se dan las soluciones débiles (denominadas 'Exact' en las figuras) correspondientes a cada uno. Los valores de los parámetros para cada uno se recogen en la tabla 5.2. Para cada caso, además, se comparan los

		SW1		SW2		SW3		SW4	
		A	B	A	B	A	B	A	B
$L_1(h)$	GTS	0.2694	1.748	0.8556	3.421	0.4294	2.670	0.7647	2.577
	LTS2	0.2687	1.299	0.8528	3.163	0.4275	2.471	0.6541	2.455
$L_2(h)$	GTS	0.0826	0.1890	0.2621	0.5764	0.2180	0.4997	0.3549	0.5349
	LTS2	0.0824	0.1814	0.2616	0.5256	0.2175	0.4755	0.2632	0.5030
$L_\infty(h)$	GTS	0.1221	0.1145	0.4509	0.3850	0.2454	0.2605	0.6788	0.4045
	LTS2	0.1221	0.1145	0.4509	0.3850	0.2454	0.2605	0.6788	0.4045

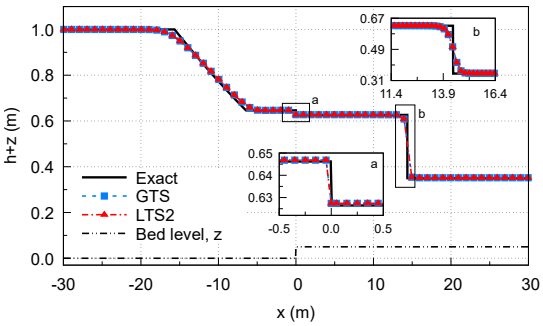
Tabla 5.3: Normas de error para los diferentes algoritmos y casos test. Ecuaciones *shallow water*.

resultados obtenidos empleando mallas irregulares A y B, ambas constituidas por $N = 1000$ celdas. Los resultados se representan en las figuras 5.4 a 5.7.

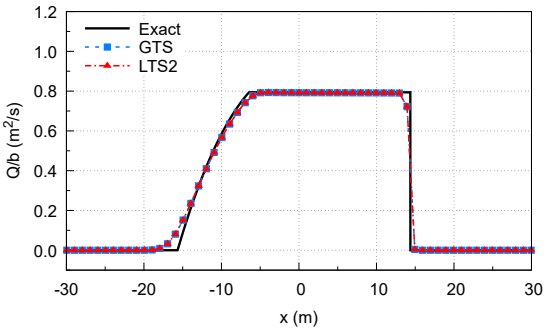
En todos los casos, las soluciones obtenidas por ambos algoritmos resultan indistinguibles a simple vista aunque, comparando los errores obtenidos en la tabla 5.3, se puede comprobar que no son del todo idénticas. No obstante, las diferencias entre ellas son despreciables y permiten asegurar que, independientemente del rendimiento obtenido, en ningún caso el LTS2 empeora la solución obtenida por el GTS.

Respecto al rendimiento, si comparamos en la figura 5.1 observamos que, con un valor permitido máximo de $M = 3$ para el LTS2 en todos los casos, este algoritmo reduce el tiempo de cálculo para todas las condiciones iniciales. Este efecto es significativamente más notorio en la malla A, costando aproximadamente la sexta parte de tiempo completar la simulación. Si bien es cierto que es un resultado bastante prometedor, esta malla favorece enormemente al LTS2 ya que casi todas las celdas van a emplear el valor de M máximo, que sigue siendo bastante limitante respecto al factor 1000 en la anchura de las celdas. Sin embargo, dado que casi todas las celdas están empleando el valor máximo $M = 3$ permitido, cabría esperar un factor de aceleración cercano a $8 = 2^3$, por lo que al haber obtenido un valor menor se hace patente el efecto de las etapas adicionales del algoritmo en cada paso global sobre el tiempo de cálculo.

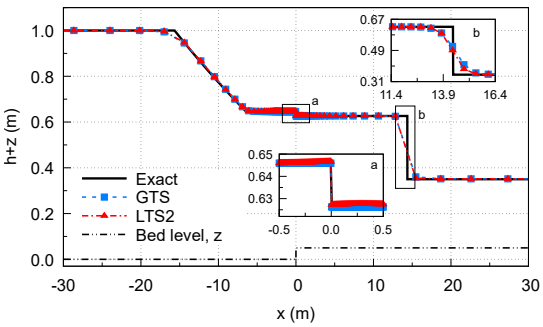
Si nos fijamos en los resultados para la malla B, en la figura 5.1, la ganancia en tiempo de cálculo toma valores más modestos. Este caso modela el comportamiento en una malla más realista, donde el tamaño de las celdas varía de forma más suave a lo largo del dominio, con refinamientos en las zonas en las se quieren estudiar con más precisión los detalles del flujo. Por tanto, podemos esperar que implementaciones sobre modelos bidimensionales presenten factores de aceleración similares, con ganancias en tiempo del cálculo del 20% o 30%. Esto es consistente con los resultados obtenidos en [3] y [4]. Los errores obtenidos al emplear la malla B son mayores que en el caso anterior, como se puede ver en la tabla 5.3; esto se debe a las regiones de los extremos, donde la discretización no es tan fina, y donde además se sitúan las ondas de choque y rarefacción generadas en la rotura de presa. Esto hace que las discontinuidades no se resuelvan con demasiada precisión, incrementando el error. Como se aprecia en las figuras, las soluciones para ambos métodos coinciden en estas regiones y podemos concluir que este efecto se debe únicamente a la malla. Como excepción, en el caso SW4, para la malla A ambos métodos producen una sobreestimación en la propagación del choque que avanza hacia la derecha, como se observa en las figuras 5.7a y 5.7b. El algoritmo LTS2, sin embargo,



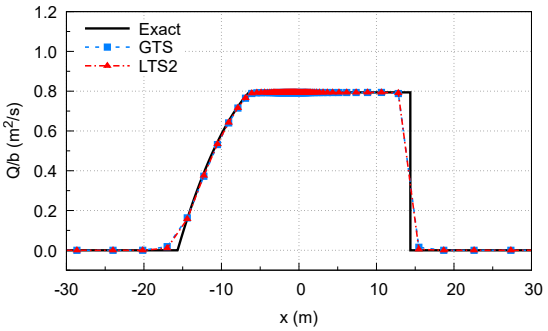
(a) SW1A - Nivel de agua



(b) SW1A - Caudal

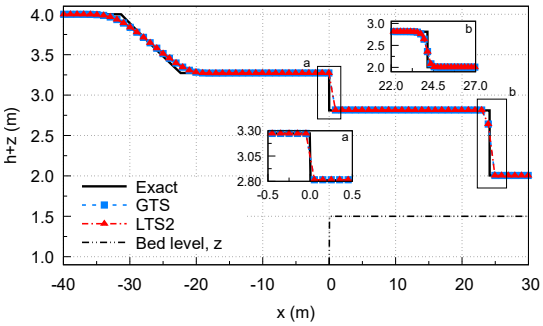


(c) SW1B - Nivel de agua

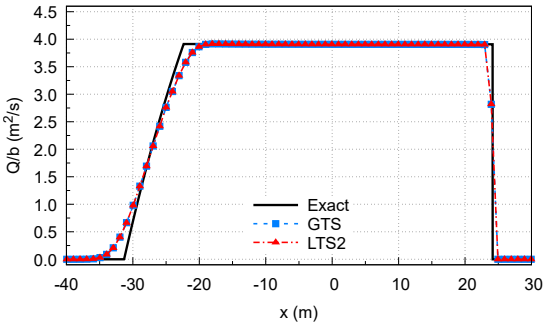


(d) SW1B - Caudal

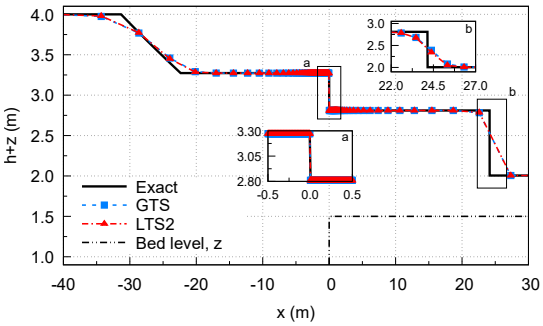
Figura 5.4: Caso test SW1, $t = 5s$. Se muestra 1 de cada 10 puntos por claridad.



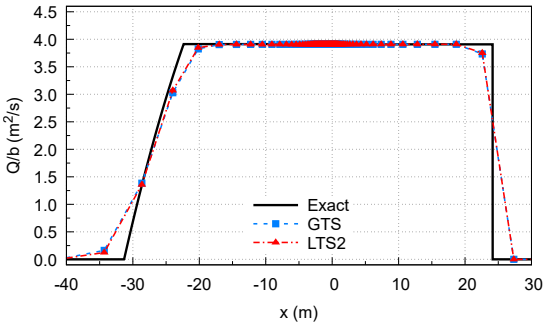
(a) SW2A - Nivel de agua



(b) SW2A - Caudal

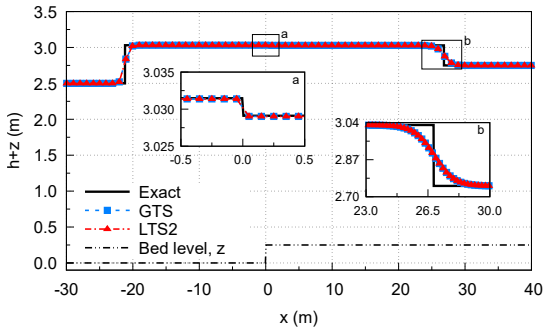


(c) SW2B - Nivel de agua

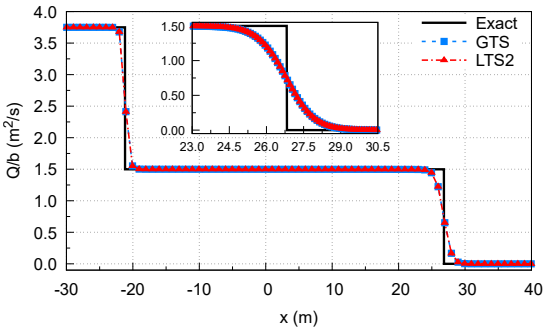


(d) SW2B - Caudal

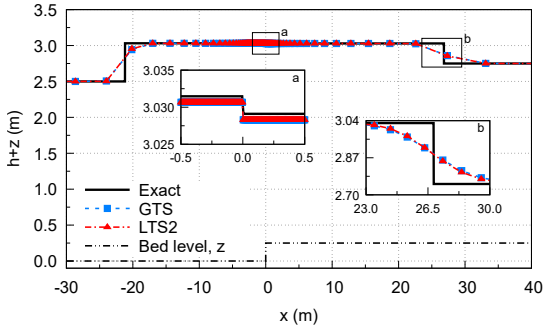
Figura 5.5: Caso test SW2, $t = 5s$. Se muestra 1 de cada 10 puntos por claridad.



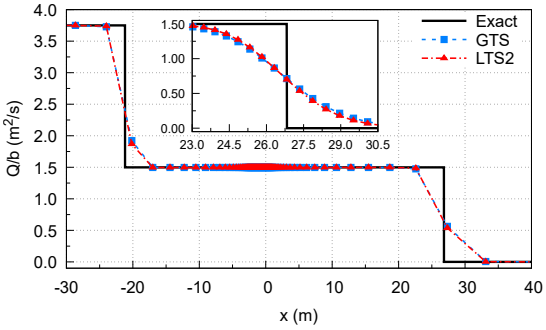
(a) SW3A - Nivel de agua



(b) SW3A - Caudal

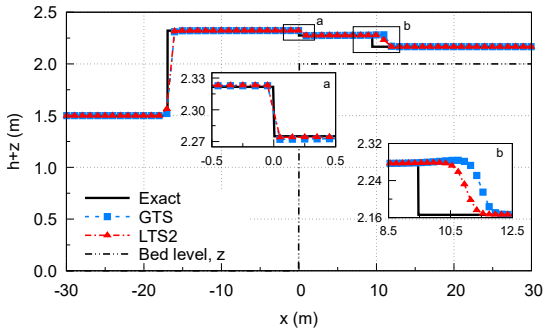


(c) SW3B - Nivel de agua

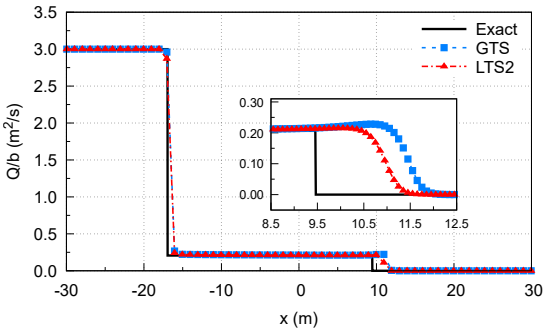


(d) SW3B - Caudal

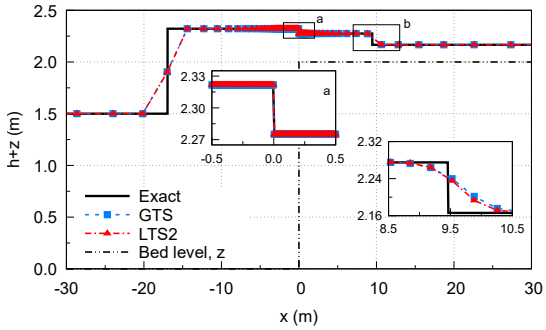
Figura 5.6: Caso test SW3, $t = 5s$. Se muestra 1 de cada 10 puntos por claridad.



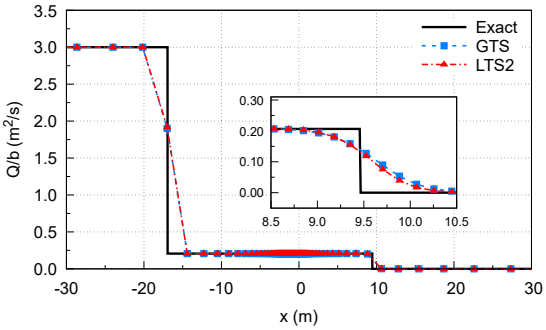
(a) SW4A - Nivel de agua



(b) SW4A - Caudal



(c) SW4B - Nivel de agua



(d) SW4B - Caudal

Figura 5.7: Caso test SW4, $t = 5s$. Se muestra 1 de cada 10 puntos por claridad.

reduce esta sobreestimación respecto al GTS, probablemente debido a una menor difusión numérica al emplearse un paso temporal más cercano al permitido a nivel local en cada celda. Comparando este resultado con la solución obtenida en [10], se observa el mismo comportamiento, por lo que se puede descartar un error de implementación o de tratamiento de los términos fuente. En la malla B ambos algoritmos resuelven correctamente la onda de choque gracias al refinamiento en la región central, que ayuda a la correcta propagación de la información. Esto muestra la ventaja de emplear mallas irregulares con refinamientos en zonas de interés, cuyo uso conjunto con los métodos LTS produce tanto ganancias en tiempo de cálculo como de precisión.

6 Conclusiones

En este trabajo se han desarrollado los esquemas numéricos explícitos empleados en la resolución de problemas transitorios de Mecánica de Fluidos, modelados según leyes hiperbólicas de conservación tales como las ecuaciones *shallow water*. Merece la pena destacar que se han implementado "desde cero"; es decir, partiendo desde su formulación en artículos científicos e implementando los algoritmos en el lenguaje de programación C. Todos ellos se pueden consultar en el siguiente repositorio de GitHub, siguiendo una filosofía de código abierto.

Además, se han expuesto los inconvenientes que presenta el uso de estos algoritmos, centrándonos en el elevado tiempo de cálculo derivado de la gran limitación del paso temporal, sujeto a condiciones de estabilidad numérica necesarias para garantizar la convergencia del mismo. A continuación, se han propuesto dos algoritmos que emplean la filosofía de *local time stepping*, cuyo objetivo es permitir que cada celda pueda avanzar en el tiempo tanto como le permita su condición de estabilidad local, sin restringirse al valor de paso temporal global. Por último, se han implementado para diferentes ecuaciones de complejidad creciente, donde se han simulado diferentes condiciones iniciales y se han comparado los resultados obtenidos para cada uno. Las conclusiones obtenidas de este estudio son:

1. Los algoritmos LTS permiten acelerar el tiempo de cálculo necesario frente a los algoritmos convencionales, con ganancias del orden del 20% al 30% en promedio. El uso conjunto con mallas irregulares permite explotar al máximo el rendimiento de los algoritmos, como se ha visto en los casos prácticos.

2. Los algoritmos LTS no pierden precisión frente a los métodos clásicos, llegando incluso a ganar en ciertos casos al reducirse la difusión numérica a nivel local por emplearse un paso temporal cercano al máximo permitido.

3. Aunque la complejidad en la programación sea bastante mayor, los beneficios comentados en los puntos anteriores superan con creces esta mayor dificultad. En particular, el LTS2 presenta una estructura fácil de adaptar a casos más complejos y mayor facilidad para el depurado de errores que el LTS1.

En suma, los algoritmos LTS son aptos para su implementación a gran escala en simuladores de diferentes ámbitos y aportan ventajas significativas que hacen interesante su estudio y desarrollo. Esto se ha mostrado ya en trabajos como [4], que implementa el LTS2 en dos dimensiones sobre

GPU, obteniendo resultados en el rendimiento comparables a los aquí presentados. Como trabajo a futuro, se podría intentar optimizar el LTS1 para comparar los resultados dados por el LTS2 en las ecuaciones *shallow water*. El siguiente paso es la implementación en modelos bidimensionales, donde los tiempos de cálculo aumentan enormemente y se suelen recurrir a mallas no estructuradas con grandes diferencias de pasos temporales locales a lo largo del dominio.

7 Referencias

[1] A. J. Crossley. "Accurate and efficient numerical solutions for the Saint Venant equations of open channel flow". PhD thesis. University of Nottingham, 1999.

[2] A. J. Crossley and N. G. Wright. "Time accurate local time stepping for the unsteady shallow water equations". In: *International Journal for Numerical Methods in Fluids* 48.7 (2005), pp. 775–799.

[3] A. J. Crossley, N. G. Wright, and C. D. Whitlow. "Local time stepping for modeling open channel flows". In: *Journal of Hydraulic Engineering* 129.6 (2003), pp. 455–462.

[4] S. Dazzi et al. "A local time stepping algorithm for GPU-accelerated 2D shallow water models". In: *Advances in Water Resources* 111 (2018), pp. 274–288.

[5] P. García-Navarro and M. E. Vázquez-Cendón. "On numerical treatment of the source terms in the shallow water equations". In: *Computers & Fluids* 29.8 (2000), pp. 951–979.

[6] W. L. Kleb, J. T. Batina, and M. H. Williams. "Temporal adaptive Euler/Navier-Stokes algorithm involving unstructured dynamic meshes". In: *AIAA Journal* 30.8 (1992), pp. 1980–1985.

[7] M. Morales-Hernández. "Efficient Explicit Finite Volume Schemes for the shallow water equations with solute transport". PhD thesis. Universidad Zaragoza, 2014.

[8] M. Morales-Hernández, P. García-Navarro, and J. Murillo. "A large time step 1D upwind explicit scheme ($CFL > 1$): Application to shallow water equations". In: *Journal of Computational Physics* 231.19 (2012), pp. 6532–6557.

[9] M. Morales-Hernández, M. E. Hubbard, and P. García-Navarro. "A 2D extension of a Large Time Step explicit scheme ($CFL > 1$) for unsteady problems with wet/dry boundaries". In: *Journal of computational physics* 263 (2014), pp. 303–327.

[10] J. Murillo and P. García-Navarro. "Weak solutions for partial differential equations with source terms: Application to the shallow water equations". In: *Journal of Computational Physics* 229.11 (2010), pp. 4327–4368.

[11] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[12] X. D. Zhang et al. "Time-accurate local time stepping method based on flux updating". In: *AIAA Journal* 32.9 (1994), pp. 1926–1929.

Apéndices

Apéndice A Análisis de von Neumann

El análisis de von Neumann es un método empleado para estudiar la estabilidad de métodos numéricos para ecuaciones en derivadas parciales (EDPs). Se basa en la descomposición del error numérico como serie de Fourier de los modos normales asociados a la malla espacial empleada en el cálculo. Sin embargo, no puede aplicarse a todos los casos. Las condiciones necesarias para poder aplicar este procedimiento son:

- Ecuación lineal en derivadas parciales
- Coeficientes constantes
- Condiciones de contorno periódicas

Si consideramos una malla uniforme con espaciado $\Delta x = L/N$, con L la longitud del dominio y N el número de nodos, podemos descomponer el error numérico como una serie sobre los modos de los armónicos asociados a la malla:

$$e(x,t) = \sum_m a_m(t) e^{ik_m x}, \tag{A.1}$$

donde $a_m(t)$ es la amplitud del modo m y k_m es el número de onda asociado al mismo, que en con la discretización empleada es $k_m = \frac{m\pi}{L} = \frac{m\pi}{N\Delta x}$. Podemos definir la fase asociada a cada modo como $\theta_m = k_m \Delta x$.

Para que el método sea estable, las amplitudes asociadas a todos los modos no pueden crecer, es decir:

$$G_m = \frac{a_m^{n+1}}{a_m^n}, \tag{A.2}$$

debe ser menor que 1 en módulo $\forall\, m$. Como hemos exigido que la EDP sea lineal, el error numérico debe cumplirla y cada uno de sus modos por separado también. Por tanto, basta analizar por separado la discretización de cada modo del error y ver que $|G_m| \leq 1$ para todos ellos. A continuación, desarrollaremos como ejemplo la ecuación de transporte lineal, que cumple las hipótesis anteriores, con diferentes discretizaciones. Es decir, la EDP es:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \tag{A.3}$$

con c una constante que representa la velocidad de propagación.

Método de diferencias finitas centradas

Esta discretización para la ecuación A.3 da lugar al siguiente esquema:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0. \tag{A.4}$$

Sustituyendo una componente del error en el punto j, $e_j^n = a^n e^{i\theta j}$, omitiendo la notación del modo por claridad (pues debe cumplirse para todos ellos), se tiene que

$$\frac{a^{n+1} - a^n}{\Delta t} e^{i\theta j} + c a^n \frac{e^{i\theta(j+1)} - e^{i\theta(j-1)}}{2\Delta x} = 0, \tag{A.5}$$

luego, dividiendo por $e^{i\theta j}$ y despejando, se llega a que la expresión de G es

$$G = \frac{a^{n+1}}{a^n} = 1 - i\frac{c\Delta t}{\Delta x}\sin\theta, \tag{A.6}$$

cuyo módulo es

$$|G|^2 = 1 + \left(\frac{c\Delta t}{\Delta x}\right)^2 \sin\theta^2 \geq 1. \tag{A.7}$$

Por lo que el esquema es incondicionalmente inestable, y no puede emplearse para la resolución de esta ecuación numéricamente.

Método *upwind* explícito

La discretización en este caso es descentrada separando la parte positiva y negativa de c para tener en cuenta ambos sentidos de propagación:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c^- \frac{u_{j+1}^n - u_j^n}{\Delta x} + c^+ \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0. \tag{A.8}$$

Sustituyendo $e_j^n = a^n e^{i\theta j}$, se tiene que

$$\frac{a^{n+1} - a^n}{\Delta t} e^{i\theta j} + c^- a^n \frac{e^{i\theta(j+1)} - e^{i\theta j}}{\Delta x} + c^+ a^n \frac{e^{i\theta j} - e^{i\theta(j-1)}}{\Delta x} = 0, \tag{A.9}$$

y por tanto la expresión del factor de amplificación es

$$G = \frac{a^{n+1}}{a^n} = 1 - \frac{\Delta t}{\Delta x}\left(c^-(1 - e^{i\theta}) + c^+(e^{i\theta} - 1)\right) = 1 - \frac{|c|\Delta t}{\Delta x}(1 - e^{i\theta}), \tag{A.10}$$

donde hemos empleado que $|c| = c^+ - c^-$. Tomando módulos, se tiene que, denotando $\nu = |c|\Delta t/\Delta x$,

$$|G| = |1 - \nu + \nu e^{i\theta}| \leq |1 - \nu| + |\nu|. \tag{A.11}$$

Y para que $|G| \leq 1$ es necesario que $0 \leq \nu \leq 1$. Por tanto, el esquema será estable si

$$\frac{|c|\Delta t}{\Delta x} \leq 1. \tag{A.12}$$

Método *upwind* implícito

En este caso, la discretización es igual a la anterior, salvo que la derivada espacial se evalúa en t^{n+1} en vez de t^n.

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+c^+\frac{u_{j+1}^{n+1}-u_j^{n+1}}{\Delta x}+c^-\frac{u_j^{n+1}-u_{j-1}^{n+1}}{\Delta x}=0. \tag{A.13}$$

Análogamente, sustituyendo la componente del error,

$$\frac{a^{n+1}-a^n}{\Delta t}e^{i\theta j}+c^+a^{n+1}\frac{e^{i\theta(j+1)}-e^{i\theta j}}{\Delta x}+c^-a^{n+1}\frac{e^{i\theta j}-e^{i\theta(j-1)}}{\Delta x}=0, \tag{A.14}$$

y, despejando, se llega a

$$G=\frac{a^{n+1}}{a^n}=\frac{1}{1+\frac{|c|\Delta t}{\Delta x}(1-e^{i\theta})}. \tag{A.15}$$

En este caso, es claro que al tomar módulos se cumple $|G|\leq 1$ bajo cualquier condición, luego el esquema implícito es incondicionalmente estable.

Apéndice B Generación de mallas irregulares

En este apéndice se detalla la generación de mallas irregulares empleadas en los casos test.

Malla aleatoria

La malla aleatoria empleada en la ecuación de convección lineal se genera mediante números aleatorios según una distribución uniforme obtenidos por el algortimo de Parisi-Rapuano. Este genera valores entre 0 y 1, que se multliplican por el factor correspondiente para mantener la longitud total en torno a 100. El código es heredado de la asignatura Física Computacional y su autor original es el profesor de la misma, Alfonso Tarancón. La semilla para la generación de la figura 5.2 es 32207.

```
#define NormRANu (2.3283063671E-10F)
#define Frec_Max 100
unsigned int irr[256];
unsigned int ir1;
unsigned char ind_ran,ig1,ig2,ig3;
int frec[Frec_Max];
extern float Random(void);
extern void ini_ran(int SEMILLA);

float Random(void){
float r;
ig1=ind_ran-24;
ig2=ind_ran-55;
ig3=ind_ran-61;
irr[ind_ran]=irr[ig1]+irr[ig2];
ir1=(irr[ind_ran]^irr[ig3]);
ind_ran++;
r=ir1*NormRANu;
//printf("r=%f\n",r);
return r;
}

void ini_ran(int SEMILLA){
printf("%d",SEMILLA);
int INI,FACTOR,SUM,i;
srand(SEMILLA);
INI=SEMILLA;
FACTOR=67397;
SUM=7364893;
for(i=0;i<256;i++)
{
INI=(INI*FACTOR+SUM);
irr[i]=INI;
}
ind_ran=ig1=ig2=ig3=0;
}
```

Código 1: Generador de números aleatorios de Parisi-Rapuano

Malla irregular A

La malla irregular A consta de N celdas de anchura uniforme Δx, salvo la celda central, que tiene una anchura mucho menor. En este caso, empleamos un factor 1000 de reducción del tamaño para asegurar que sea la celda que gobierne el paso temporal para cualquier condición del flujo.

Malla irregular B

La malla irregular B se construye siguiendo el procedimiento detallado en el apéndice de [1]. El objetivo es crear una malla irregular con l celdas centrales de longitud d, simétrica respecto a la celda central, de tal forma que las celdas aumentan su tamaño según un factor f, hasta la última celda de tamaño nd.

```
int nCells; // Numero de celdas
double L; // Longitud del canal
int k = nCells/4-1;
int n_factor = 128; // Factor entre celdas interiores y exteriores
double f = pow(n_factor,1./(k));
double d = L/2./(n_factor+f*(1-pow(f,k))/(1-f)+(nCells+2)/4);

dx[0] = n_factor*d;
x[0] = dx[0]/2.;
for (i=1; i<=k; i++)
{
    dx[i] = dx[i-1]/f;
    x[i] = x[i-1] + 0.5*(dx[i]+dx[i-1]);
}
for (   ; i<=k+1+nCells/2; i++)
{
    dx [i] = d;
    x[i] = x[i-1] + 0.5*(dx[i]+dx[i-1]);
}
for (   ; i<nCells-1; i++)
{
    dx[i] = dx[i-1]*f;
    x[i] = x[i-1] + 0.5*(dx[i]+dx[i-1]);
}
dx[i] = n_factor*d;
x[i] = x[i-1] + 0.5*(dx[i]+dx[i-1]);
```

Código 2: Ejemplo del código en C para la generación de esta malla irregular