Universidad de San Carlos de Guatemala
Escuela de Ciencias Físicas y Matemáticas
Departamento de Física

SOLUCIÓN NUMÉRICA DE LAS ECUACIONES DE EULER DE LA DINÁMICA DE FLUIDOS MEDIANTE EL ESQUEMA DE ROE

RODRIGO RAFAEL CASTILLO CHONG
Asesorado por DR. ENRIQUE PAZOS ÁVALOS

Guatemala, febrero de 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

ESCUELA DE CIENCIAS FÍSICAS Y MATEMÁTICAS

SOLUCIÓN NUMÉRICA DE LAS ECUACIONES DE EULER DE LA DINÁMICA DE FLUIDOS MEDIANTE EL ESQUEMA DE ROE

TRABAJO DE GRADUACIÓN
PRESENTADO A LA JEFATURA DEL
DEPARTAMENTO DE FÍSICA
POR

RODRIGO RAFAEL CASTILLO CHONG
ASESORADO POR DR. ENRIQUE PAZOS ÁVALOS

AL CONFERÍRSELE EL TÍTULO DE
LICENCIADO EN FÍSICA APLICADA

GUATEMALA, FEBRERO DE 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
ESCUELA DE CIENCIAS FÍSICAS Y MATEMÁTICAS

CONSEJO DIRECTIVO INTERINO

Director	M.Sc. Jorge Marcelo Ixquiac Cabrera
Representante Docente	Arqta. Ana Verónica Carrera Vela
Representante Docente	M.A. Pedro Peláez Reyes
Representante de Egresados	Lic. Urías Amitaí Guzmán García
Representante de Estudiantes	Elvis Enrique Ramírez Mérida
Representante de Estudiantes	Oscar Eduardo García Orantes
Secretario	M.Sc. Freddy Estuardo Rodríguez Quezada

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

Director	M.Sc. Jorge Marcelo Ixquiac Cabrera
Examinador	M.Sc. Osmar Obdulio Hernández Aguilar
Examinador	Dr. José Rodrigo Sacahui Reyes
Examinador	Dr. Juan Adolfo Ponciano Castellanos
Secretario	Ing. Edgar Damián Ochoa Hernández

Este archivo pdf es una muestra

Fecha

datos

cuerpo

despedida

firma

nombre

AGRADECIMIENTOS

DEDICATORIA

A mí mamá, mis primas, mis tías, mis amigos, mis amigas, “El Núcleo”, “Las Amebas” y mis compañeros de trabajo.

“He know, **flow** like interstellar wind...” - MF DOOM

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
F_x	derivada parcial de F respecto a x
$\mathbf{U}$	vector de magnitudes conservadas
U	vector de magnitudes conservadas aproximadas numéricamente
$\mathbf{F}$	vector de flujos de magnitudes conservadas exactas
F	vector de flujos de magnitudes conservadas aproximados numéricamente
$\mathbf{A}$	matriz jacobiana
ε	coeficiente de viscosidad cinemática
u	velocidad del gas sobre el eje x
ρ	densidad del gas
p	presión del gas
e	energía interna específica del gas
E	densidad de energía total del gas
T	temperatura del gas
W	trabajo realizado sobre el gas
k_B	constante de Boltzmann
$\mathcal{R}$	constante específica de los gases
c_v	capacidad calorífica específica a volumen constante
c_p	capacidad calorífica específica a presión constante
γ	coeficiente de dilatación adiabática
α	grados de libertad internos del gas
S	entropía del gas

OBJETIVOS

General

Resolver las ecuaciones de Euler de la dinámica de fluidos para un gas ideal, con el método de volúmenes finitos, utilizando el esquema de Roe.

Específicos

1. Describir el método de volúmenes finitos y la motivación de su uso.
2. Describir el funcionamiento del esquema de Roe y su implementación en el lenguaje `C++`.
3. Comparar las soluciones obtenidas a través del programa implementado en `C++` con las soluciones producidas con la librería PyClaw del lenguaje Python.
4. Analizar la diferencia entre simulaciones considerando gases con distintos grados de libertad, aprovechando la solución numérica obtenida a través del programa escrito en `C++`.

INTRODUCCIÓN

El estudio de las ecuaciones diferenciales es de gran importancia en las ciencias físicas, ya que cada teoría física se sustenta en ecuaciones diferenciales que describen el comportamiento a través del tiempo de cualquier sistema que dicha teoría busque explicar. La motivación del estudio de las ecuaciones diferenciales es encontrar soluciones generales de las mismas, principalmente a través de métodos analíticos que buscan soluciones exactas de las ecuaciones diferenciales. Sin embargo, no todas las ecuaciones diferenciales poseen soluciones exactas, lo cual motiva el estudio y desarrollo de métodos numéricos para la resolución de las mismas.

En el área de estudio del análisis numérico aplicado a ecuaciones diferenciales, existe una gran variedad de métodos y esquemas que se aplican para obtener una solución numérica, esto se debe a la amplia variedad de ecuaciones diferenciales de la física que carecen de solución analítica. Por otro lado, las ecuaciones diferenciales parciales son considerablemente más complejas que las ecuaciones diferenciales ordinarias, por lo que existen métodos más apropiados para resolver ecuaciones diferenciales que involucran funciones de varias variables.

Las ecuaciones de conservación tienen un papel importante en múltiples áreas de la física, de tal manera que se han desarrollado métodos numéricos apropiados para resolver este tipo de ecuaciones diferenciales parciales, siendo el método de volúmenes finitos el más utilizado. Un conjunto en particular de ecuaciones de conservación son las ecuaciones de Euler, que rigen la dinámica de un fluido compresible y no viscoso a partir de su ecuación de estado. Existen pocas soluciones analíticas conocidas a las ecuaciones de Euler, por lo que resolver este conjunto de ecuaciones de conservación con un método numérico apropiado resulta ser un problema interesante.

1. ECUACIONES DE CONSERVACIÓN Y SISTEMAS HIPERBÓLICOS DE PRIMER ORDEN

En este capítulo se introducen los conceptos fundamentales de las ecuaciones de conservación y sistemas hiperbólicos de primer orden. También se introduce el problema de Riemann asociado a una ecuación de conservación. Los desarrollos de los conceptos en este capítulo están basados en la presentación del tema realizada por Randall LeVeque [6], especialmente en los capítulos *The Derivations of Conservation Laws*, *Scalar Conservations Laws*, *Scalar Examples* y *Linear Hyperbolic Systems*.

1.1. Ecuaciones de conservación

En física, una ecuación de conservación es una ecuación diferencial parcial de la siguiente forma

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = 0 \tag{1.1}$$

o utilizando una notación más compacta para las derivadas,

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0 \tag{1.2}$$

donde $\mathbf{U}$ es un vector n-dimensional de variables físicas que se conservan, por ejemplo, la energía, la masa o el momentum de un medio. En este texto, las variables de las que depende $\mathbf{U}$ dependen de x y t, una variable espacial y otra temporal respectivamente. Por tanto, $\mathbf{U}$ se define formalmente como $\mathbf{U} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$, mientras que la i-ésima variable conservada se denomina $\mathbf{u}_i$, de tal manera que $\mathbf{U} = \mathbf{U}(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n)$.

La función $\mathbf{F}$ corresponde al **flujo** de cada una de las variables involucradas en un punto (x, t). Al igual que $\mathbf{U}$, la función $\mathbf{F}$ depende de las mismas variables físicas y por ende, también depende de (x, t). Sin embargo, el flujo de cada variable

conservada puede tener una forma distinta, entonces es conveniente escribir a $\mathbf{F}$ como un vector de n funciones independientes, $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n)$ de tal manera que f_i es la función de flujo de la i-ésima variable conservada, u_i.

Una ecuación de conservación para un sistema definido en un intervalo espacial $D = [a, b]$ necesita de condiciones iniciales para su resolución, el caso más simple a considerar es el de un problema de Cauchy. En dicho caso, se debe especificar una función $\mathbf{U}_0(x)$

$$\mathbf{U}(x, 0) = \mathbf{U}_0(x) \tag{1.3}$$

la cual sea válida para todo x tal que $x \in D$ y condiciones de frontera

$$\mathbf{U}(a, t) = \mathbf{U}_a \tag{1.4}$$

$$\mathbf{U}(b, t) = \mathbf{U}_b \tag{1.5}$$

con $\mathbf{U}_a$ y $\mathbf{U}_b$ fijos.

Otra forma de escribir una ecuación de conservación es utilizando la matriz jacobiana $\mathbf{A}(\mathbf{U})$ definida como

$$\mathbf{A}(\mathbf{U}) \equiv \begin{bmatrix} \frac{\partial \mathbf{f}_1}{\partial u_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_n}{\partial u_1} & \cdots & \frac{\partial \mathbf{f}_n}{\partial u_n} \end{bmatrix} \tag{1.6}$$

de tal manera que la ecuación (1.1) se convierte en

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x = 0. \tag{1.7}$$

Esta forma de escribir una ecuación de conservación es relevante ya que permite definir un **sistema hiperbólico**. Un sistema hiperbólico es una ecuación de conservación de la forma (1.7) tal que los autovalores de la matriz $\mathbf{A}(\mathbf{U})$ para todo $\mathbf{U}$ sean reales y que dicha matriz sea diagonalizable. Esto implica que existen n vectores propios linealmente independientes de $\mathbf{A}(\mathbf{U})$. Una ecuación de conservación depende de una función $\mathbf{F}(\mathbf{U})$ que, por lo general, no es una función lineal de $\mathbf{U}$, lo que implica que las ecuaciones de conservación son regularmente no lineales. Esto también se puede inferir por la dependencia en $\mathbf{U}$ de la matriz $\mathbf{A}$ en la ecuación (1.7).

1.2. Derivación de una ecuación de conservación

El principio físico de una ecuación de conservación es más explícito cuando esta se deriva a través de cantidades expresadas en forma **integral.** Considerando un ejemplo de la mecánica de fluidos, se define $M(x_1, x_2, t)$ como la cantidad de masa de un fluido que se encuentra contenido en un "tubo" unidimensional en un intervalo $[x_1, x_2]$ en un tiempo t. Si a dicho fluido se le asocia una densidad $\rho(x, t)$, entonces esta última se define de tal manera que su integral definida en un intervalo espacial sea igual a la masa contenida en ese mismo intervalo, i.e.,

$$M(x_1, x_2, t) = \int_{x_1}^{x_2} \rho(x, t)\, \mathrm{d}x\,. \tag{1.8}$$

Ahora, asumiendo que el tubo es cerrado e impenetrable, la cantidad de masa en una región arbitraria $[x_1, x_2]$ puede variar solamente a causa de que el fluido se desplace (fluya) a través de los puntos límites de la región, x_1 y x_2. Para cuantificar el flujo que sale o entra en una región se necesita la velocidad del fluido, $v(x, t)$. Cabe destacar que debido a que el fluido se mueve en un espacio unidimensional, su velocidad se limita a dirigirse en el mismo sentido espacial, es decir, su velocidad tiene dirección sobre x. Entonces el flujo del fluido en un punto (x, t), $F(x, t)$, se define como

$$F(x, t) = \rho(x, t)v(x, t) \tag{1.9}$$

. Entonces, como previamente se comentó, se puede escribir la razón instantánea de cambio de masa en la región $[x_1, x_2]$ en términos del flujo entrante y saliente de la misma región

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[M(x_1, x_2, t)\right] = F(x_2, t) - F(x_1, t) \tag{1.10}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2} \rho(x, t)\, \mathrm{d}x = \rho(x_2, t)v(x_2, t) - \rho(x_1, t)v(x_1, t) \tag{1.11}$$

Esta es la forma integral de una ecuación de conservación. En particular, esta ecuación refleja el principio de conservación de la masa y a su vez es conocida como **ecuación de continuidad**. La ecuación (1.11) se puede integrar en el tiempo para conseguir expresarla independientemente de cualquier derivada, obteniendo

$$\int_{t_1}^{t_2} \frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2} \rho(x, t)\, \mathrm{d}x\, \mathrm{d}t = \int_{t_1}^{t_2} \left[\rho(x_2, t)v(x_2, t) - \rho(x_1, t)v(x_1, t)\right] \mathrm{d}t \tag{1.12}$$

$$\int_{x_1}^{x_2} [\rho(x,t_2) - \rho(x,t_1)]\,\mathrm{d}x = \int_{t_1}^{t_2} [\rho(x_2,t)v(x_2,t) - \rho(x_1,t)v(x_1,t)]\,\mathrm{d}t\,. \tag{1.13}$$

Asumiendo que $t_1 < t_2$, esta igualdad ofrece una expresión para la diferencia de masa contenida en la región $[x_1, x_2]$ entre los instantes t_2 y t_1.

Es posible obtener una forma diferencial partiendo de la forma integral de una ecuación de conservación, pero es necesario asumir que las funciones $\rho(x,t)$ y $v(x,t)$ son **diferenciables**. Esta última característica que se exige en las funciones entra en conflicto cuando se estudian soluciones[1] de las ecuaciones de conservación con discontinuidades. Por lo tanto, la forma integral de las ecuaciones es utilizada al estudiar problemas con dichas características. Para convertir la ecuación en forma integral a su forma diferencial, se tienen que usar las siguientes expresiones:

$$\rho(x,t_2) - \rho(x,t_1) = \int_{t_1}^{t_2} \frac{\partial}{\partial t}\rho(x,t)\,\mathrm{d}t \tag{1.14}$$

y

$$\rho(x_2,t)v(x_2,t) - \rho(x_1,t_1)v(x_1,t) = \int_{x_1}^{x_2} \frac{\partial}{\partial x}\rho(x,t)v(x,t)\,\mathrm{d}x \tag{1.15}$$

sustituyendo estas expresiones en (1.13) se obtiene

$$\int_{t_1}^{t_2}\int_{x_1}^{x_2} \left[\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial \rho(x,t)v(x,t)}{\partial x}\right]\mathrm{d}x\,\mathrm{d}t = 0. \tag{1.16}$$

Puesto que la ecuación (1.16) se cumple para cualquier punto (x,t) del dominio, el integrando de la misma debe ser idénticamente cero. Entonces,

$$\rho_t + (\rho v)_x = 0 \tag{1.17}$$

es la forma diferencial de la ecuación de conservación de la masa, así como se definió una ecuación de conservación en (1.2), ya que la función de flujo se definió como $F = \rho v$. Puesto que la ecuación diferencial de conservación de la masa involucra dos cantidades físicas, esta se puede resolver ya sea si se conoce previamente la función $v(x,t)$ o si esta última se puede escribir como una función de ρ, i.e., $v = v(\rho)$. En este último caso la ecuación de conservación de la masa es una ecuación diferencial parcial únicamente para ρ y toma la siguiente forma

$$\rho_t + f(\rho)_x = 0. \tag{1.18}$$

[1]Dichas soluciones se conocen como **soluciones débiles**, tema que se abordará a detalle más adelante.

Esta última es un ejemplo de **ecuación de conservación escalar** ya que solamente interviene una incógnita, ρ. En caso que la velocidad $v(x,t)$ sea una constante α, la ecuación de conservación de la masa se convierte en:

$$\rho_t + \alpha\rho_x = 0, \tag{1.19}$$

esta ecuación se conoce como **ecuación de advección lineal** o como ecuación de onda de primer orden [8].

1.3. Ecuación de advección lineal

La ecuación de advección (1.19) con la siguiente condición inicial:

$$\rho(x,0) = \rho_0(x), \qquad -\infty < x < \infty, \tag{1.20}$$

tiene como solución:

$$\rho(x,t) = \rho_0(x - \alpha t), \tag{1.21}$$

para $t > 0$, asumiendo que ρ_0 es una función diferenciable. Esta solución se puede interpretar como la traslación de la función ρ_0 a lo largo del eje x con una velocidad α, en la misma dirección de la velocidad, es decir, a la derecha si $\alpha > 0$ y a la izquierda si $\alpha < 0$.

1.3.1. Curvas características

Se puede demostrar que la solución (1.21) es constante respecto al tiempo a lo largo de cada curva definida por $x - \alpha t = x_0$, para cualquier x_0 tal que $x_0 \in [-\infty, \infty]$. Dichas curvas de la forma $x = x(t)$ son conocidas como **curvas características** de la ecuación diferencial en cuestión. Las curvas características definen dominios en donde la ecuación diferencial parcial se puede escribir como una ecuación diferencial ordinaria. Dichas curvas se definen a partir de ecuaciones diferenciales ordinarias. En el caso de la ecuación de advección lineal (1.21), las características satisfacen la siguiente ecuación diferencial:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \alpha, \qquad x(0) = x_0, \tag{1.22}$$

cuya solución es: $x - \alpha t = x_0$. Para demostrar que la ecuación de advección lineal se convierte en una ecuación diferencial ordinaria a lo largo de dichas curvas, se deriva

la función incógnita ρ respecto al tiempo,

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \frac{\partial\rho}{\partial t} + \frac{\partial\rho}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.23}$$

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \frac{\partial\rho}{\partial t} + \alpha\frac{\partial\rho}{\partial x} \tag{1.24}$$

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = 0, \tag{1.25}$$

este último paso toma efecto dado que a lo largo de las curvas características, $\frac{\mathrm{d}x}{\mathrm{d}t} = \alpha$. Se puede notar que en este último procedimiento se recuperó la ecuación diferencial en cuestión (1.21), expresada como una ecuación diferencial ordinaria y concluyó al igualar la derivada temporal a cero, consiguiendo demostrar que ρ es constante a lo largo de las características.

1.4. Dominio de dependencia de una ecuación de conservación

En la sección anterior se introdujo el concepto de curvas características de una ecuación de conservación y en el caso de la ecuación de advección, es posible obtener la solución general utilizando la expresión para sus características, esto es $x(t) = x_0 + \alpha t$. Por lo tanto, se puede afirmar que el valor de $\rho(x,t)$ para un instante (x',t') depende de la condición inicial ρ_0 únicamente en un solo punto x_0' ya que ρ es constante a lo largo de la curva característica $x' = x_0^{'} + \alpha t$. Entonces, sin importar de qué manera se cambien los valores que toma $\rho_0(x)$, mientras no se cambie su valor en x_0', el valor $\rho(x',t')$ permanecerá sin cambiar.

En el caso general, para cualquier sistema hiperbólico, el valor de la solución $\mathbf{U}$ en un punto (x',t') depende únicamente de los valores que toma la función inicial $\mathbf{U}_0$ en un dominio $D(x',t')$ que corresponde a un intervalo **cerrado**. Este último se denomina **dominio de dependencia**. El tamaño de este intervalo usualmente crece a medida que avance el tiempo, sin embargo, el tamaño está acotado de la siguiente forma $D(x',t') = \{x : |x - x'| \leq a_{\text{máx}}t\}$. En resumen, las ecuaciones hiperbólicas poseen una velocidad finita de propagación; lo que significa que la información de la solución puede viajar a lo largo del espacio a una velocidad $a_{\text{máx}}$ como máximo.

1.5. Discontinuidades en soluciones

Al derivar la forma diferencial de una ecuación de conservación (1.16) se enfatizó que era necesario exigir la diferenciabilidad de las cantidades conservadas. Sin embargo, si se toman en cuenta los atributos de la solución, proporcionados por las curvas características, de la ecuación de advección, resulta que el valor de la variable involucrada $\rho(x,t)$ depende únicamente del valor de la condición inicial en un punto x_0 tal que $x = x_0 + \alpha t$. Por esta última razón se puede concluir que, al menos para la ecuación de advección, es posible definir una solución sin exigirle que sea una función suave. La definición de una solución con singularidades surge gracias a que la forma integral de una ecuación de conservación sí se satisface con funciones no diferenciables, en contraste con la forma diferencial. También es importante hacer énfasis en que la forma integral de una ecuación de conservación tiene un significado físico más fundamental.

Si la condición inicial de una ecuación de conservación escalar $u_0(x)$ tiene una singularidad (discontinuidad en sí misma o en alguna derivada), en un punto x_0, entonces la solución de la ecuación, $u(x,t)$, también poseerá una singularidad del mismo orden en todos los puntos definidos por la característica que parte de x_0. Para los sistemas hiperbólicos en general, las singularidades se propagan a lo largo de las curvas características.

A continuación se describen dos formas de definir las soluciones con singularidades:

1.5.1. Viscosidad disipada

Como se vio anteriormente algunas soluciones de las ecuaciones de conservación pueden poseer singularidades si la condición inicial de la ecuación tiene una singularidad. Para los sistemas hiperbólicos no lineales, las discontinuidades pueden formarse a lo largo de la evolución temporal de la solución. A las singularidades que se desarrollan en las soluciones se les llama **ondas de choque** o **shocks** en inglés [3]. Para definir cómo estas funciones siguen siendo soluciones, en el sentido estricto de la palabra, se redefine la ecuación que deben satisfacer agregándole un coeficiente de viscosidad ε que va acompañado del laplaciano de la variable involucrada. En el caso de la ecuación de advección para u, se define la siguiente ecuación modificada

$$u_t + \alpha u_x = \varepsilon u_{xx}, \tag{1.26}$$

donde $u_{xx} \equiv \frac{\partial^2 u}{\partial x^2}$. A esta ecuación se le conoce como **ecuación de advección - difusión**, ya que el término viscoso también se denomina como término difusivo y está involucrado en la ecuación de difusión, también conocida como ecuación de calor [8].

Si se denota a $u^\varepsilon(x,t)$ como la solución de la ecuación (1.26) con condición inicial $u_0(x)$, entonces $u^\varepsilon \in C^\infty(\mathbb{R} \times [0,\infty))$ donde $C^\infty(\mathbb{R} \times [0,\infty))$ es el conjunto de funciones infinitamente diferenciables (suaves) definidas en el mismo espacio en donde se resuelve la ecuación. Esto se cumple incluso si $u_0(x)$ no es una función suave, ya que la ecuación (1.26) es de tipo parabólica. Para encontrar la solución a la ecuación original (1.19) se debe hacer el límite $\varepsilon \to 0$ en la solución de la ecuación de advección - difusión (1.26); a esta forma de definir una solución con discontinuidades se le denomina **solución con viscosidad disipada**.

1.5.2. Solución débil

La otra forma de definir una solución con singularidades es definiendo una **solución débil** de la ecuación de conservación, haciendo uso de la forma integral de esta última. Es evidente que la forma diferencial de la ecuación de conservación no se satisface con una función con singularidades ya que sus derivadas no están bien definidas en todo el dominio de la función. La idea del procedimiento de la definición de una solución débil es reescribir la ecuación en forma integral, trasladando las derivadas hacia una **función prueba**, que sí es una función diferenciable y así relajar las condiciones de suavidad para la función incógnita.

Sea $\phi(x,t)$ una función continuamente diferenciable con **soporte compacto**, i.e., $\phi(x,t) \in C_0^1(\mathbb{R}\times[0,\infty))$. La tenencia de soporte compacto de la función significa que $\phi(x,t)$ es exactamente igual a cero fuera de algún conjunto cerrado, por lo que el soporte de la función yace en un subconjunto compacto del espacio $(x,t) = \mathbb{R}\times[0,\infty)$ [3].

Para definir la solución débil de la ecuación de conservación escalar $u_t + f(u)_x = 0$, esta se multiplica por la función ϕ descrita anteriormente:

$$u_t + f(u)_x = 0 \tag{1.27}$$

$$\phi u_t + \phi f(u)_x = 0. \tag{1.28}$$

Ahora, se integra sobre todo el espacio y el tiempo,

$$\int_0^\infty \int_{-\infty}^\infty (\phi u_t + \phi f(u)_x)\,\mathrm{d}x\,\mathrm{d}t = 0, \tag{1.29}$$

luego, integrando por partes cada término de la integral, se obtiene

$$\int_{-\infty}^\infty \int_0^\infty \phi u_t\,\mathrm{d}t\,\mathrm{d}x = \int_{-\infty}^\infty \int_0^\infty ((\phi u)_t - \phi_t u)\,\mathrm{d}t\,\mathrm{d}x \tag{1.30}$$

$$\int_{-\infty}^\infty \int_0^\infty \phi u_t\,\mathrm{d}t\,\mathrm{d}x = \int_{-\infty}^\infty (\phi u)\Big|_0^\infty \mathrm{d}x - \int_{-\infty}^\infty \int_0^\infty \phi_t u\,\mathrm{d}t\,\mathrm{d}x\,. \tag{1.31}$$

Dado que ϕ tiene soporte compacto, es posible asegurarse que $\lim\limits_{t\to\infty} \phi(x,t) = 0$. Por tanto,

$$\int_{-\infty}^\infty \int_0^\infty \phi u_t\,\mathrm{d}t\,\mathrm{d}x = -\int_{-\infty}^\infty \phi(x,0)u(x,0)\,\mathrm{d}x - \int_{-\infty}^\infty \int_0^\infty \phi_t u\,\mathrm{d}t\,\mathrm{d}x\,. \tag{1.32}$$

Trabajando el segundo término, se tiene

$$\int_0^\infty \int_{-\infty}^\infty \phi f(u)_x\,\mathrm{d}x\,\mathrm{d}t = \int_0^\infty \int_{-\infty}^\infty ((\phi f(u))_x - \phi_x f(u))\,\mathrm{d}x\,\mathrm{d}t \tag{1.33}$$

$$\int_0^\infty \int_{-\infty}^\infty \phi f(u)_x\,\mathrm{d}x\,\mathrm{d}t = \int_0^\infty \phi f(u)\Big|_{-\infty}^\infty \mathrm{d}t - \int_0^\infty \int_{-\infty}^\infty \phi_x f(u)\,\mathrm{d}x\,\mathrm{d}t\,, \tag{1.34}$$

de igual manera que con la valuación de la integral del término anterior, el soporte compacto de ϕ permite justificar que $\lim\limits_{x\to-\infty} \phi = 0 = \lim\limits_{x\to\infty} \phi$. Entonces,

$$\int_0^\infty \int_{-\infty}^\infty \phi f(u)_x\,\mathrm{d}x\,\mathrm{d}t = -\int_0^\infty \int_{-\infty}^\infty \phi_x f(u)\,\mathrm{d}x\,\mathrm{d}t\,. \tag{1.35}$$

Por tanto, la integral de la ecuación de conservación multiplicada por la función prueba queda como

$$\begin{aligned}\int_0^\infty \int_{-\infty}^\infty (\phi u_t + \phi f(u)_x)\,\mathrm{d}x\,\mathrm{d}t &= \int_{-\infty}^\infty \phi(x,0)u(x,0)\,\mathrm{d}x \\ &\quad - \int_0^\infty \int_{-\infty}^\infty (\phi_t u + \phi_x f(u))\,\mathrm{d}x\,\mathrm{d}t = 0.\end{aligned} \tag{1.36}$$

Por tanto, se obtiene la igualdad

$$\int_0^\infty \int_{-\infty}^\infty (\phi_t u + \phi_x f(u))\,\mathrm{d}x\,\mathrm{d}t = -\int_{-\infty}^\infty \phi(x,0)u(x,0)\,\mathrm{d}x\,, \tag{1.37}$$

que tal y como se dijo previamente, las derivadas parciales ahora están siendo aplicadas a ϕ y la integral de la nueva ecuación diferencial se iguala a una integral que depende únicamente de las condiciones iniciales del problema a resolver para u. Entonces se dice que u es una **solución débil** de la ecuación (1.27) si y solo si se cumple la igualdad (1.37) para todo ϕ tal que $\phi(x,t) \in C_0^1(\mathbb{R} \times [0,\infty))$.

Es posible demostrar que la igualdad (1.37) es equivalente a la forma integral de la ecuación diferencial en cuestión (ecuación 1.13), mediante la elección de una función de prueba que esta cumpla con la siguiente propiedad:

$$\phi(x,t) = \begin{cases} 1 & \text{para } (x,t) \in [x_1, x_2] \times [t_1, t_2] \\ 0 & \text{para } (x,t) \notin [x_1 - \epsilon, x_2 + \epsilon] \times [t_1 - \epsilon, t_2 + \epsilon]. \end{cases} \tag{1.38}$$

Además ϕ es una función suave en la banda intermedia con ancho ϵ. Entonces al calcular la integral (1.37) se puede notar que esta se reduce a la integral sobre la banda ϵ, a raíz de la propiedad expuesta anteriormente. Al calcular esta integral y haciendo el límite $\epsilon \to 0$, las derivadas ϕ_x y ϕ_t tienden a deltas de Dirac [2], recuperando así los valores de u y $f(u)$ en las fronteras del dominio $[x_1, x_2] \times [t_1, t_2]$ en integrales que equivalen a los términos de la ecuación de conservación en forma integral (1.13). Cabe destacar que este es un esbozo solamente de la demostración, ya que esta requiere más rigor matemático. Sin embargo, lo imprescindible es mostrar la equivalencia entre ambas expresiones.

La solución de viscosidad disipada también califica como una solución débil y físicamente es la solución correcta. Existen muchas soluciones débiles, incluyendo algunas que no tienen un significado físico consistente, para ello es necesario exigir otras condiciones sobre las soluciones débiles de tal manera que se obtenga la solución correcta. Estas condiciones son conocidas como *condiciones de entropía* por su papel en la dinámica de fluidos.

1.6. Ecuación de Burgers

Se pueden destacar las propiedades más importantes de las ecuaciones de conservación no lineales de la mecánica de fluidos a través de un ejemplo escalar conocido como **ecuación de Burgers.** Esta ecuación tiene la siguiente forma:

[2] La delta de Dirac $\delta(x)$ es una función generalizada que cumple con la siguiente propiedad $\int_{-\infty}^{\infty} \delta(x-a) f(x)\,\mathrm{d}x = f(a)$ [2].

$$u_t + uu_x = 0, \tag{1.39}$$

la cual es formalmente conocida como Ecuación de Burgers sin viscosidad, pero en este texto se le referirá solamente como Ecuación de Burgers. Esta ecuación es un modelo muy simple de la velocidad u de un gas [3]. La ecuación originalmente estudiada por Jan Burgers contiene un término de viscosidad cinemática:

$$u_t + uu_x = \varepsilon u_{xx}, \tag{1.40}$$

donde ε es el coeficiente de viscosidad cinemática. A esta última ecuación se le referirá como Ecuación de Burgers viscosa.

De la ecuación (1.39) podemos notar un ligero parecido a la ecuación de advección (1.19), con la diferencia que la velocidad de advección α se reemplaza por la variable que sufre la advección, i.e., u, lo que brinda la naturaleza no lineal de este modelo. La ecuación de Burgers se puede escribir en forma conservativa:

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0. \tag{1.41}$$

A partir de esta forma, se puede concluir que la función de flujo es $f(u) = \frac{1}{2}u^2$.

1.6.1. Curvas características

Las curvas características de la ecuación de Burgers se pueden definir a través de la siguiente ecuación diferencial

$$x'(t) = u(x(t), t), \tag{1.42}$$

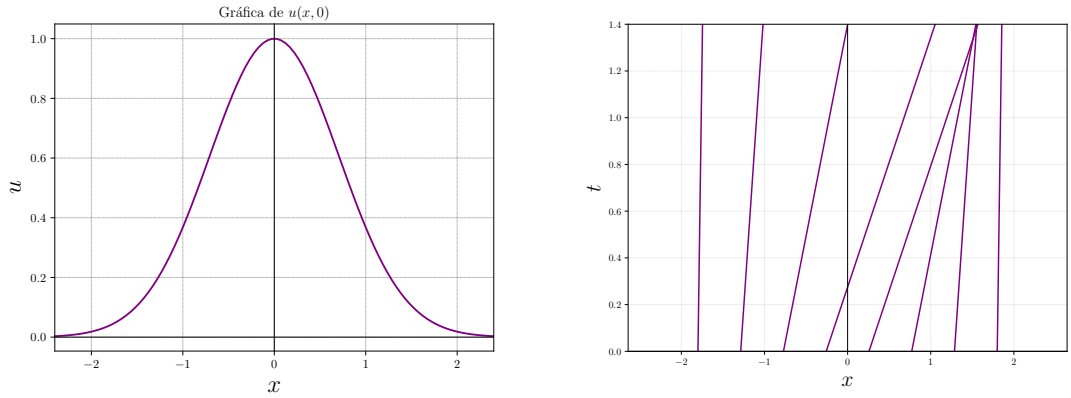
dado que se puede recuperar la ecuación diferencial parcial derivando respecto al tiempo la velocidad:

$$\frac{\mathrm{d}}{\mathrm{d}t}u(x(t), t) = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.43}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u(x(t), t) = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0. \tag{1.44}$$

Por lo tanto, u es constante en el tiempo sobre cualquier característica. Se puede resolver la ecuación (1.42) utilizando este último hecho:

$$x(t) = u(\xi, 0)t + \xi \tag{1.45}$$



(a) Gráfica de $u(x,0) = \exp(-x^2)$. **(b)** Características de la ecuación de Burgers.

Figura 1.1. Gráficas de la condición inicial $u(x,0) = \exp(-x^2)$ para la ecuación de Burgers y de algunas características asociadas a esta condición inicial. **Fuente:** elaboración propia.

y al resolver para ξ, es posible resolver la ecuación diferencial

$$u(x,t) = u(\xi,0). \tag{1.46}$$

Es importante destacar que la expresión (1.46) para $u(x,t)$ no es válida para todo t, dado que las características definidas por la ecuación (1.45) pueden intersectarse en algún momento. Esto se da si para algún x, se tiene que $u_0'(x) < 0$.

1.6.2. Ondas de choque

Como se puede observar en la figura (1.1b), las características asociadas a una condición inicial con forma gaussiana se intersectan. Entonces, si las características se intersectan, la expresión (1.46) no está bien definida, sino que $u(x,t)$ puede tomar varios valores a la vez, cada uno correspondiente a una característica diferente. En el caso de la ecuación de Burgers, cuando las características se intersectan, la solución se convierte en una "función" con tres valores distintos. Esta solución puede tener sentido en algunos contextos, como en el caso del problema de agua superficial. Sin embargo, en la mayoría de los casos esta solución no tiene significado físico. Por ejemplo, en la ecuación de Burgers la variable $u(x,t)$ representa una velocidad y esta debe devolver un solo valor.

La solución con el comportamiento físico correcto cuando las características se intersectan corresponde a una discontinuidad que se propaga con una velocidad determinada. Este efecto se puede demostrar a través del formalismo de la visco-

sidad disipada. Al obtener la solución de la ecuación de Burgers viscosa (1.40), el parámetro εu_{xx} evita que la solución presente una discontinuidad, dado que cuando la onda comienza a presentar el choque, la segunda derivada de u crece más rápido que su primera derivada. Así, al efectuar el límite de $\varepsilon \to 0$ se puede obtener una discontinuidad absoluta que corresponde a la solución de la ecuación de Burgers sin viscosidad. La tendencia de adquirir una discontinuidad de las soluciones con menos viscosidad se puede evidenciar en la figura (1.2).

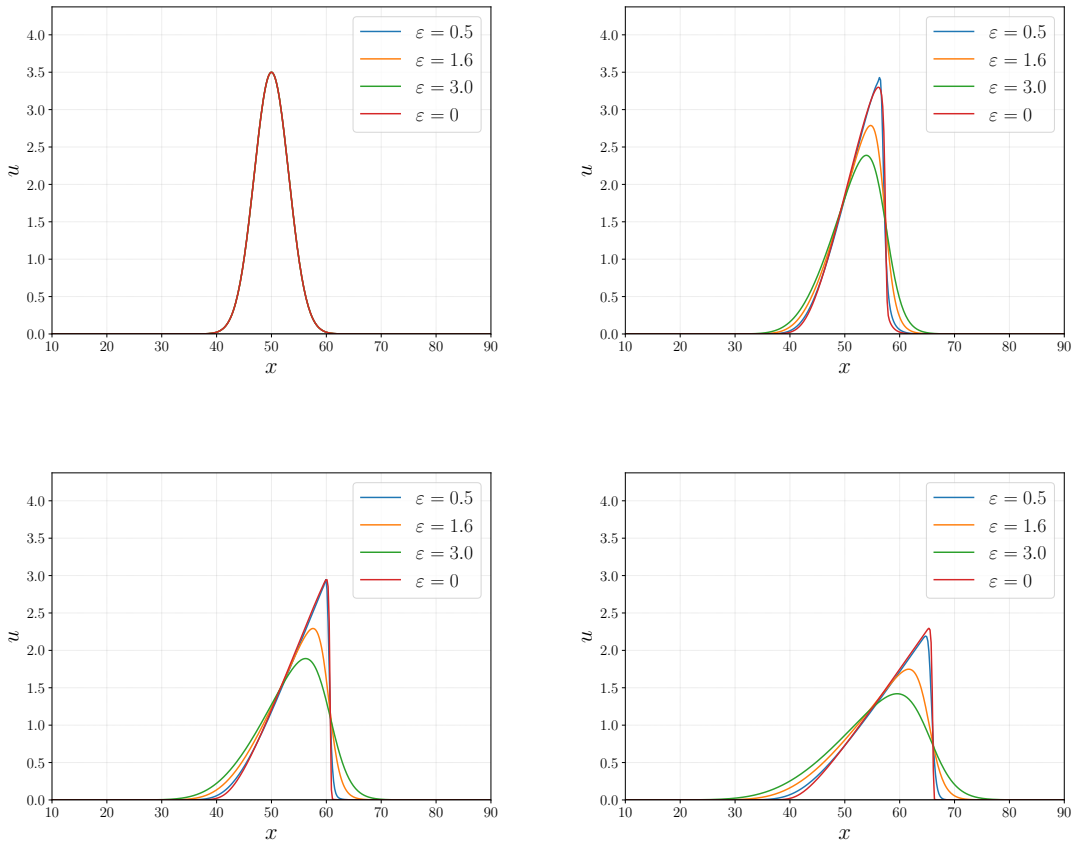


Figura 1.2. Gráficas de las soluciones numéricas, en distintos instantes de tiempo, de la ecuación de Burgers viscosa para distintos valores de coeficiente de viscosidad cinemática (ε), incluyendo el caso donde no hay viscosidad (en rojo). **Fuente:** elaboración propia.

1.7. Problema de Riemann

Como se ha mostrado, las soluciones a las ecuaciones de conservación admiten soluciones débiles que pueden presentar discontinuidades tanto en la condición inicial

como en la evolución temporal de las mismas. Por esta razón, es natural estudiar ecuaciones de conservación con condiciones iniciales discontinuas. Este problema es conocido como **problema de Riemann**. En el caso escalar, el problema de Riemann para una ecuación de conservación de la forma $u_t + f(u)_x = 0$ corresponde a la siguiente condición inicial

$$u(x,0) = \begin{cases} u_L & \text{si } x < 0 \\ u_R & \text{si } x \geq 0, \end{cases} \tag{1.47}$$

en donde la solución es determinada por la relación entre u_L y u_R.

1.7.1. Solución al problema de Riemann para la ecuación de Burgers

Para encontrar la solución al problema de Riemann de la ecuación de Burgers es necesario considerar dos casos.

1.7.1.1. Caso $u_L > u_R$:

La función

$$u(x,t) = \begin{cases} u_L, & x < st \\ u_R, & x > st, \end{cases} \tag{1.48}$$

conocida como **onda de choque**, representa una discontinuidad trasladándose a través del eje x con una velocidad s. Esta última es una solución débil de la ecuación de Burgers, para la condición inicial (1.47), si se cumple la condición general de **Rankine - Hugoniot** [3], que se define como:

$$f(u_L) - f(u_R) = s(u_L - u_R), \tag{1.49}$$

donde $f(u) = \frac{1}{2}u^2$. A continuación se demuestra la expresión general de la condición Rankine - Hugoniot.

Prueba: Sea u una solución débil de la ecuación diferencial $u_t + f(u)_x = 0$, de la forma (1.48). Sea $\mathcal{M} \gg st$. Aplicando la expresión (1.11) a u:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{-\mathcal{M}}^{\mathcal{M}} u(x,t)\, \mathrm{d}x = f(u_L) - f(u_R). \tag{1.50}$$

Integrando u a partir de su definición (1.48), se tiene

$$\int_{-\mathcal{M}}^{\mathcal{M}} u(x,t)\,\mathrm{d}x = (\mathcal{M}+st)u_L + (\mathcal{M}-st)u_R. \tag{1.51}$$

Entonces, sustituyendo en (1.50),

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[(\mathcal{M}+st)u_L + (\mathcal{M}-st)u_R\right] = f(u_L) - f(u_R) \tag{1.52}$$

$$s(u_L - u_R) = f(u_L) - f(u_R), \tag{1.53}$$

se obtiene la condición de salto de Rankine - Hugoniot.

Por lo tanto, para el caso $u_L > u_R$, la solución de la ecuación de Burgers es una discontinuidad que se propaga con una velocidad constante $s = \frac{u_L+u_R}{2}$ [3].

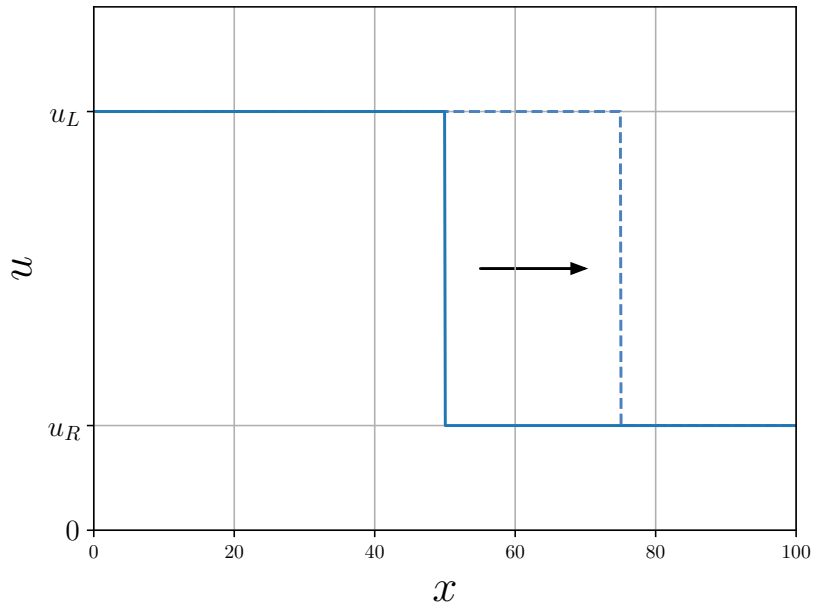


Figura 1.3. Ejemplo gráfico de la solución de la ecuación de Burgers para el caso $u_L > u_R$.
Fuente: elaboración propia, basada en el esquema obtenido de [6].

1.7.1.2. Caso $u_L < u_R$:

Hay múltiples soluciones débiles para este caso, por ejemplo, la solución (1.48) [3]. Sin embargo, esta no es la solución que corresponde a la solución con viscosidad

disipiada [3]. La solución

$$u(x,t) = \begin{cases} u_L, & x < u_L t \\ x/t, & u_L t \leq x \leq u_R t \\ u_R, & x > u_R t, \end{cases} \tag{1.54}$$

es la solución correspondiente a la solución con viscosidad disipada y es conocida como una **onda de rarefacción**. Resulta ser poco práctico tener que recurrir frecuentemente a la solución de viscosidad disipada para encontrar la solución a cualquier caso del problema de Riemann, por lo que en su lugar se suelen imponer condiciones conocidas como **condiciones de entropía** que son equivalentes a exigir que la solución sea consistente con el formalismo de viscosidad disipada.

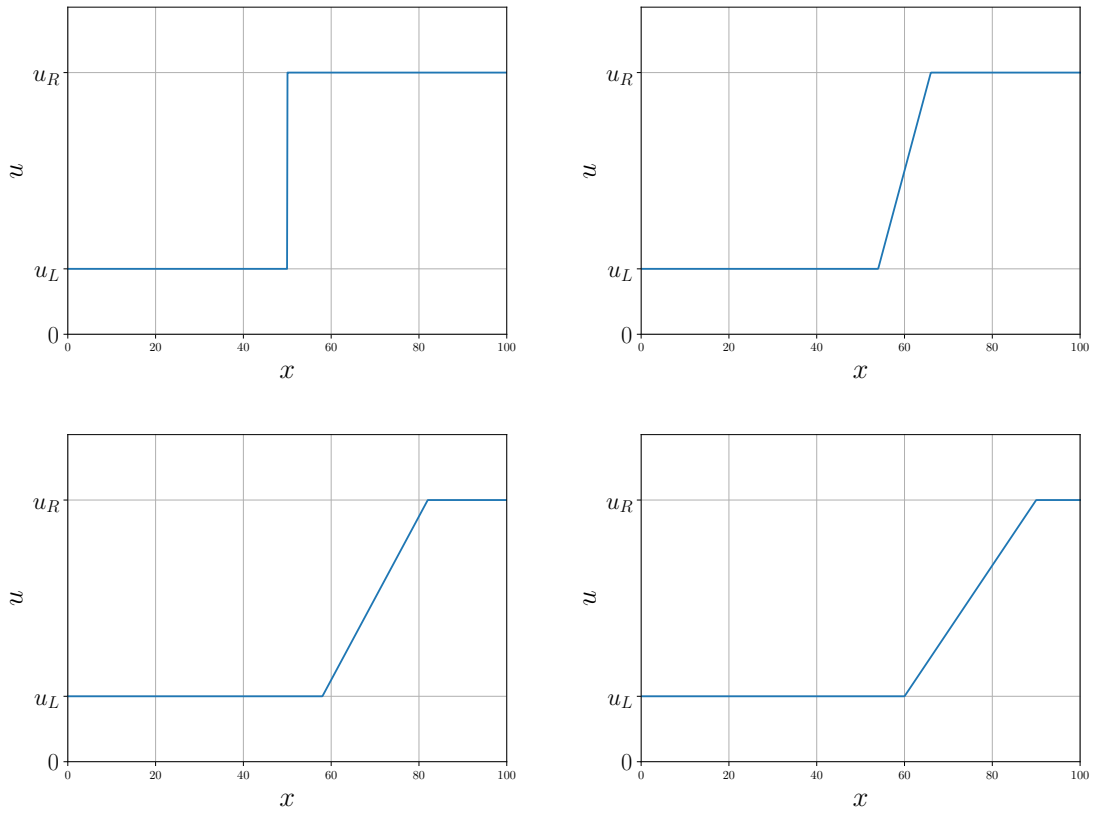


Figura 1.4. Gráficas de las soluciones, en distintos instantes de tiempo, de la ecuación de Burgers para el caso $u_L < u_R$ del problema de Riemann **Fuente:** elaboración propia.

1.8. Condiciones de entropía

Se ha mencionado que la solución con viscosidad disipada es la solución débil correcta que debe considerarse al resolver una ecuación de conservación, sin embargo, resulta ser complicado encontrar la solución para una ecuación de conservación con viscosidad. Por lo tanto, se suelen imponer un conjunto de condiciones que sean equivalentes a aplicar el formalismo de viscosidad disipada. Estas se denominan **condiciones de entropía**, dado que cumplen un papel similar al de la entropía misma en problemas de la dinámica de los gases.

Existen varias versiones de la condición de entropía para una ecuación de conservación. Se mostraran las dos más relevantes con respecto a los métodos numéricos a tratar en este texto.

1.8.1. Condición de entropía, primera versión

La primera versión de la condición de entropía establece que una discontinuidad que se propaga a velocidad s debe cumplir con la siguiente desigualdad

$$f'(u_L) > s > f'(u_R) \tag{1.55}$$

para que se satisfaga la misma condición. Para interpretar esta condición se debe resaltar que la derivada $f'(u)$ corresponde a la velocidad característica de la onda en un punto (x, t). Por lo tanto, se puede notar que las curvas características se intersectan con la velocidad de propagación de la discontinuidad, de tal manera que las características a la izquierda de la onda de choque avanzan hacia el mismo choque y las características a la derecha son alcanzadas por la onda de choque.

Por otro lado, si se tuviese una discontinuidad que se propaga con una velocidad s tal que se cumple la condición contraria, i.e., $f'(u_L) < s < f'(u_R)$, esta solución no sería estable ante alguna perturbación, ya sea al añadir una pequeña viscosidad o suavizando la discontinuidad en la solución. Al aplicar dichas perturbaciones, se generarían ondas de rarefacción en lugar de ondas de choque.

1.8.2. Condición de entropía, segunda versión

Esta condición desarrollada por Oleinik indica que si $u(x, t)$ es una solución débil físicamente válida si existe una constante $E > 0$ tal que para todo $a > 0$,

$t > 0$, sobre cualquier punto x del dominio espacial se cumple que:

$$\frac{u(x+a,t) - u(x,t)}{a} < \frac{E}{t} \tag{1.56}$$

.

1.9. Sistemas hiperbólicos lineales

El objetivo de esta sección es explicar la solución general de un sistema hiperbólico lineal y la solución del problema de Riemann asociado. Estos resultados serán de alta utilidad para construir la solución numérica que se busca en este texto.

El sistema descrito en la ecuación (1.2) con una función de flujo dada por $\mathbf{F}(\mathbf{U}) = \mathbf{A}\mathbf{U}$, siendo $\mathbf{A}$ una matriz con entradas constantes, es un sistema lineal de conservación. El sistema se puede escribir como

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_x = 0 \tag{1.57}$$

donde $\mathbf{A} \in \mathbb{R}^{m\times m}$ y $\mathbf{U} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^m$. Formalmente hablando, este es un sistema lineal de m cantidades conservadas. Naturalmente, también se define una condición inicial para este sistema,

$$\mathbf{U}(x, 0) = \mathbf{U}_0(x). \tag{1.58}$$

En la sección 1.1 se definió un sistema hiperbólico de primer orden como un sistema de la forma (1.7) en donde los autovalores de $\mathbf{A}$ son todos distintos y son reales. De tal manera que si el sistema (1.57) es hiperbólico, se puede diagonalizar $\mathbf{A}$, i.e.,

$$\mathbf{A} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}. \tag{1.59}$$

En esta relación, $\mathbf{\Lambda}$ es la matriz definida como $\text{diag}(\lambda_1, \lambda_2, \ldots \lambda_m)$ donde λ_i es el i-ésimo autovalor de $\mathbf{A}$. Mientras que $\mathbf{R}$ es la matriz formada por los autovectores columna de $\mathbf{A}$, denominados $\mathbf{r}_i$. Por ello, se tiene que

$$\mathbf{A}\mathbf{r}_p = \lambda_p \mathbf{r}_p \tag{1.60}$$

para todo $p \in [1, 2, \ldots, m]$.

1.9.1. Resolución de un sistema hiperbólico lineal

Se asumirá que se tiene un sistema de la forma (1.57), tal que los autovalores de $\mathbf{A}$ son todos distintos y reales, i.e., se trata de un sistema **estrictamente hiperbólico**. Este sistema se puede resolver introduciendo nuevas variables conocidas como **variables características**. Se define $\mathbf{v}$ como

$$\mathbf{v} \equiv \mathbf{R}^{-1}\mathbf{U}. \tag{1.61}$$

Por otro lado, se multiplica la ecuación (1.57) por $\mathbf{R}$

$$\mathbf{R}^{-1}\mathbf{U}_t + \mathbf{R}^{-1}\mathbf{A}\mathbf{U}_x = 0, \tag{1.62}$$

sustituyendo la expresión (1.59) obteniendo,

$$\mathbf{R}^{-1}\mathbf{U}_t + \mathbf{\Lambda}\mathbf{R}^{-1}\mathbf{U}_x = 0 \tag{1.63}$$

al usar el hecho de que $\mathbf{R}^{-1}\mathbf{R} = \mathbf{I}$. Luego, usando la definición de $\mathbf{v}$ (1.61) y dado que este ejemplo corresponde a un sistema lineal, las matrices $\mathbf{R}$ y su inversa son constantes, por tanto se tiene que

$$\mathbf{v}_t + \mathbf{\Lambda}\mathbf{v}_x = 0 \tag{1.64}$$

. Cabe destacar la similitud entre esta última ecuación y la ecuación (1.57). Sin embargo, la principal diferencia entre éstas es que la matriz $\mathbf{\Lambda}$ es una matriz diagonal y la matriz $\mathbf{A}$, en general, no lo es. Por tanto, el sistema definido en (1.64) se puede desacoplar, obteniendo m ecuaciones escalares lineales independientes. Esto es,

$$(v_p)_t + \lambda_p (v_p)_x = 0, \qquad \forall p \in [1, 2, \ldots, m], \tag{1.65}$$

donde v_p es cada componente de $\mathbf{v}$. Este último sistema se resuelve usando el resultado de la ecuación de advección lineal escalar (1.21), obteniendo

$$v_p(x, t) = v_p(x - \lambda_p t, 0) \tag{1.66}$$

dado que $v_p(x, 0)$ es la condición inicial para cada ecuación escalar según (1.58).

Ya que se tiene la solución al sistema lineal en términos de las variables características, es oportuno escribir las soluciones en términos de las variables originales,

i.e., $\mathbf{U}$. De forma que se tiene,

$$\mathbf{v}(x,0) = \mathbf{R}^{-1}\mathbf{U}_0(x) \tag{1.67}$$

y

$$\mathbf{U}(x,t) = \mathbf{R}\mathbf{v}(x,t) \tag{1.68}$$

. Además $\mathbf{U}$ se puede escribir como una combinación lineal de los autovectores de $\mathbf{A}$, tomando en cuenta que el valor $\mathbf{v}_p$ es el coeficiente de cada autovector $\mathbf{r}_p$ en dicha expansión lineal, de tal manera que se tiene lo siguiente

$$\mathbf{U}(x,t) = \sum_{p=1}^{m} v_p(x,t)\mathbf{r}_p \tag{1.69}$$

$$\mathbf{U}(x,t) = \sum_{p=1}^{m} v_p(x-\lambda_p t,0)\mathbf{r}_p \tag{1.70}$$

. Cabe destacar que, de forma similar al caso escalar, la solución $\mathbf{U}$ a este sistema lineal solo depende de la condición inicial $\mathbf{U}_0$ en los m conjuntos de puntos, dados por $x - \lambda_p t$. Formalmente, su dominio de dependencia se escribe como $D(\bar{x},\bar{t}) = \{x = \bar{x} - \lambda_p \bar{t}, \quad p = 1,2,\ldots,m\}$.

De la misma manera que al generalizar el dominio de dependencia de una ecuación de advección lineal, se pueden introducir las curvas características asociadas a este sistema. El conjunto de características de la p familia está dado por la ecuación $x'(t) = \lambda_p$. De tal manera que dichas curvas son de la forma $x = x_0 + \lambda_p t$ y se interpretan como rectas sobre el plano $x - t$, lo cual es consistente con el hecho de que se tratan de las características asociadas a un sistema lineal. El coeficiente $v_p(x,t)$ de cada autovector $\mathbf{r}_p$ es constante a lo largo de cada característica.

1.9.2. Solución al problema de Riemann para un sistema hiperbólico lineal

El problema de Riemann para un sistema hiperbólico lineal corresponde a la ecuación diferencial

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_x = 0 \tag{1.71}$$

junto a la siguiente condición inicial

$$\mathbf{U}_0(x) = \begin{cases} \mathbf{U}_L, & x < 0 \\ \mathbf{U}_R, & x > 0, \end{cases} \tag{1.72}$$

considerando que este sistema es estrictamente hiperbólico.

Para resolver este problema, es conveniente escribir los dos estados $\mathbf{U}_L$ y $\mathbf{U}_R$ en términos de los autovectores de $\mathbf{A}$, $\mathbf{r}_i$. Esta expansión se sustenta en la propiedad de independencia lineal que poseen los vectores $\mathbf{r}_i$ y dado que estos son m distintos vectores, indica que forman una base del espacio $\mathbb{R}^m$. Por tanto,

$$\mathbf{U}_L = \sum_{p=1}^{m} \alpha_p \mathbf{r}_p \tag{1.73}$$

$$\mathbf{U}_R = \sum_{p=1}^{m} \beta_p \mathbf{r}_p \tag{1.74}$$

con α_p y β_p reales y constantes, para todo $p = 1, 2, \ldots, m$. Luego, utilizando la solución (1.69) se obtiene

$$v_p(x,0) = \begin{cases} \alpha_p, & x < 0 \\ \beta_p, & x > 0, \end{cases} \tag{1.75}$$

y

$$v_p(x,t) = \begin{cases} \alpha_p, & \text{si} \quad x - \lambda_p t < 0 \\ \beta_p, & \text{si} \quad x - \lambda_p t > 0, \end{cases} \tag{1.76}$$

utilizando la relación del resultado (1.66). Entonces, la solución al problema de Riemann tiene la forma (1.70), con v_p definido como una función por partes.

Si se define a $P(x,t)$ como el valor máximo de p tal que $x - \lambda_p t > 0$, entonces se puede escribir la solución al problema de Riemann como:

$$\mathbf{U}(x,t) = \sum_{p=1}^{P(x,t)} \beta_p \mathbf{r}_p + \sum_{p=P(x,t)+1}^{m} \alpha_p \mathbf{r}_p \tag{1.77}$$

.

Se puede verificar que esta solución cumple con la condición de salto de Rankine-Hugoniot (1.49). Notando que al atravesar una curva característica p, el coeficiente

de $\mathbf{r}_p$ salta de acuerdo a la siguiente expresión:

$$\mathbf{U}_{Lp} - \mathbf{U}_{Rp} = (\beta_p - \alpha_p)\mathbf{r}_p \tag{1.78}$$

donde $\mathbf{U}_{Lp}$ y $\mathbf{U}_{Rp}$ son los valores de la onda a la izquierda y derecha de la curva característica p, respectivamente.

Tomando en cuenta que $\mathbf{F}(\mathbf{U}) = \mathbf{A}\mathbf{U}$, entonces:

$$\mathbf{F}(\mathbf{U}_{Lp}) - \mathbf{F}(\mathbf{U}_{Rp}) = \mathbf{A}(\mathbf{U}_{Lp} - \mathbf{U}_{Rp}) \tag{1.79}$$

$$\mathbf{F}(\mathbf{U}_{Lp}) - \mathbf{F}(\mathbf{U}_{Rp}) = \mathbf{A}(\beta_p - \alpha_p)\mathbf{r}_p \tag{1.80}$$

$$\mathbf{F}(\mathbf{U}_{Lp}) - \mathbf{F}(\mathbf{U}_{Rp}) = \lambda_p(\mathbf{U}_{Lp} - \mathbf{U}_{Rp}), \tag{1.81}$$

con lo cual se demuestra que se cumple la condición Rankine - Hugoniot, notando que la velocidad de propagación de la discontinuidad es λ_p.

Otra forma útil de escribir la solución (1.77) es a través de los saltos previamente desarrollados, esto es,

$$\mathbf{U}(x,t) = \mathbf{U}_L(x,t) + \sum_{\lambda_p < x/t} (\beta_p - \alpha_p)\mathbf{r}_p, \tag{1.82}$$

o de forma equivalente,

$$\mathbf{U}(x,t) = \mathbf{U}_R(x,t) - \sum_{\lambda_p \geq x/t} (\beta_p - \alpha_p)\mathbf{r}_p. \tag{1.83}$$

2. MÉTODO DE VOLÚMENES FINITOS Y ESQUEMA DE ROE

A continuación se describen el método y los esquemas a utilizar para llevar a cabo una solución numérica de una ecuación de conservación. La idea principal del capítulo es describir el método de volúmenes finitos y la motivación de su uso. Se explicarán los esquemas adecuados para aplicar el mencionado método, un solucionador del problema de Riemann, denominado esquema de Godunov y otro solucionador aproximado del problema de Riemann, denominado esquema de Roe. Este último es el esquema elegido para resolver las ecuaciones de Euler en este texto. El contenido de este capítulo se basa en el capítulo *Numerical Methods in One Dimension* del texto [5] y en el capítulo de *Numerical Methods* del texto [6], ambos escritos por Randall LeVeque.

2.1. Método de volúmenes finitos

El método de volúmenes finitos (MVF) es un método numérico de integración que se especializa en resolver ecuaciones diferenciales escritas en forma conservativa. El MVF destaca por ofrecer una interpretación peculiar de la función a resolver, ya que es un método basado en la forma **integral** de las ecuaciones.

2.1.1. Discretización del dominio

Al aplicar un método numérico para resolver una ecuación diferencial se necesita discretizar el dominio de la función y la función misma, redefiniendo algunas nociones matemáticas y utilizando aproximaciones. Sea $D = [a, b]$ el dominio espacial de la solución de una ecuación de conservación. Para aplicar el MVF, este dominio se divide en N intervalos iguales llamados **celdas**. Cada celda se denomina $\mathcal{C}_i$ y se define como un intervalo, $\mathcal{C}_i = [x_i, x_{i+1}]$. Además, cada celta tiene un ancho

h [1], dado por

$$h = \frac{b-a}{N}. \tag{2.1}$$

Por lo tanto, se tiene una expresión para el valor de cada x_i,

$$x_i = a + (i-1)h. \tag{2.2}$$

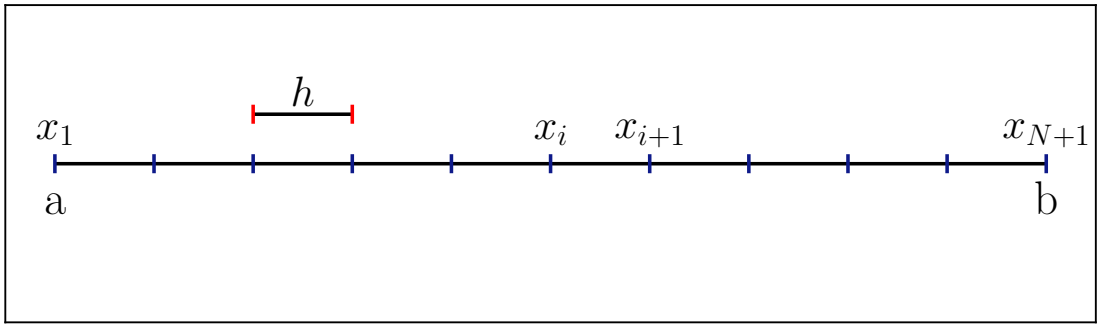


Figura 2.1. Esquema de símbolos utilizados para la discretización del dominio espacial. **Fuente:** elaboración propia.

El dominio temporal, dado por la variable t, se discretiza de forma similar. El instante inicial t_0 corresponde a $t = 0$, de tal manera que cada instante consecuente está separado por un múltiplo entero de una cantidad k denominada *salto temporal*. Por lo tanto, el enésimo instante de tiempo queda expresado como

$$t_n = nk, \qquad \forall n \in \mathbb{N}. \tag{2.3}$$

2.1.2. Aproximaciones numéricas

Para referirse a la función $\mathbf{U}$ discretizada se utilizará una notación especial, esta es,

$$\mathbf{U}(x_i, t_n) \approx U_i^n \tag{2.4}$$

que se interpreta como el valor aproximado numéricamente de la función $\mathbf{U}$ en un instante t_n. Para conseguir dicha aproximación, el MVF utiliza la siguiente expresión

$$U_i^n \approx \frac{1}{h}\int_{x_i}^{x_{i+1}} \mathbf{U}(x, t_n)\,\mathrm{d}x \equiv \frac{1}{h}\int_{\mathcal{C}_i} \mathbf{U}(x, t_n)\,\mathrm{d}x \tag{2.5}$$

[1] La diferencia entre dos puntos sobre el eje x también se denomina Δx, pero se opta por usar h como símbolo, para evitar escribir ecuaciones engorrosas.

de tal manera que, aproximadamente, U_i^n toma el valor promedio de $\mathbf{U}(x, t_n)$ [2] sobre la celda $\mathcal{C}_i$. Vale la pena destacar que si las funciones u_j de las que depende $\mathbf{U}$ son funciones suaves, la expresión para U_i^n coincide en $\mathcal{O}(h^2)$ con el valor exacto de $\mathbf{U}$ en el punto medio de la i-ésima celda.

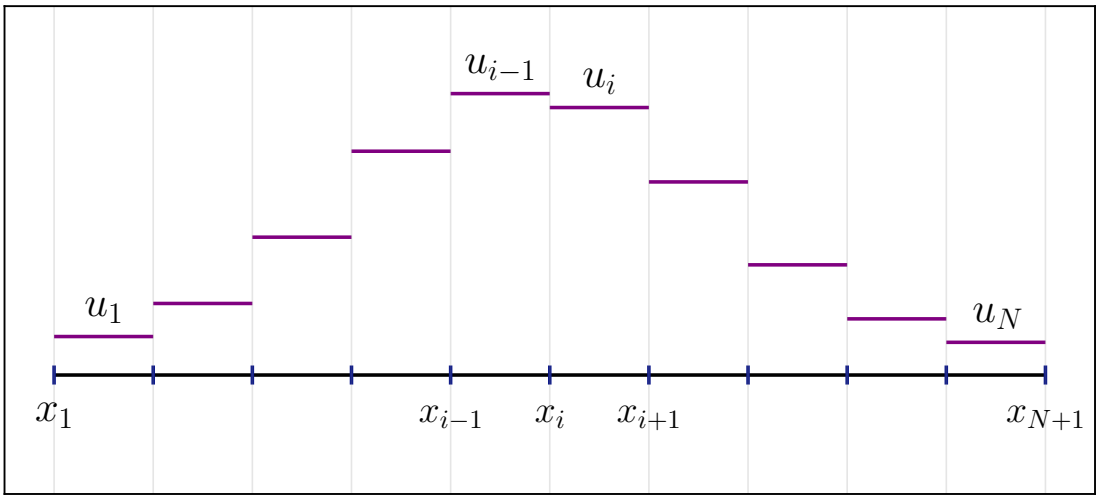


Figura 2.2. Gráfica de la construcción de una función escalar numéricamente aproximada u_i, en un dominio con pocas celdas. Se puede considerar que una función numéricamente aproximada por el MVF es una función definida por partes, en donde cada parte es una constante. **Fuente:** elaboración propia.

La ventaja de utilizar un método aproximado basado en los valores promedio de las funciones en las celdas es que éste puede considerarse un método **conservativo** de tal manera que imite la ley de conservación obtenida a partir de la forma integral de la ecuación de conservación. Esta característica es sumamente importante al considerar ondas de choque como posibles soluciones.

Habiendo considerado la aproximación para obtener U_i^n, para continuar con la discretización del problema se puede integrar la ecuación de conservación sobre el dominio $[x_i, x_{i+1}] \times [t_n, t_{n+1}]$, que corresponde a la integral sobre una celda $\mathcal{C}_i$ del dominio espacial y sobre un instante temporal. Utilizando la expresión (1.13) se obtiene

$$\int_{\mathcal{C}_i} \mathbf{U}(x, t_{n+1})\,\mathrm{d}x - \int_{\mathcal{C}_i} \mathbf{U}(x, t_n)\,\mathrm{d}x = \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{U}(x_i, t))\,\mathrm{d}t - \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{U}(x_{i+1}, t))\,\mathrm{d}t \tag{2.6}$$

[2] Para un sistema de conservación, tanto $\mathbf{U}$ como U son vectores. Entonces, formalmente hablando, cada componente de U es aproximadamente el valor promedio de cada componente de $\mathbf{U}$ en una celda $\mathcal{C}_i$.

$$\frac{1}{h}\int_{\mathcal{C}_i}\mathbf{U}(x,t_{n+1})\,\mathrm{d}x-\frac{1}{h}\int_{\mathcal{C}_i}\mathbf{U}(x,t_n)\,\mathrm{d}x=\frac{1}{h}\int_{t_n}^{t_{n+1}}\mathbf{F}(\mathbf{U}(x_i,t))\,\mathrm{d}t-\frac{1}{h}\int_{t_n}^{t_{n+1}}\mathbf{F}(\mathbf{U}(x_{i+1},t))\,\mathrm{d}t \tag{2.7}$$

En esta expresión se puede obtener la diferencia exacta entre dos estados temporales del valor promedio de $\mathbf{U}$. Las integrales del flujo sobre el tiempo no se pueden calcular con exactitud, dado que se necesitaría saber exactamente cómo evoluciona $\mathbf{U}$ con el tiempo. Entonces, se define el flujo aproximado $F^n_{i-\frac{1}{2}}$,

$$F^n_{i-\frac{1}{2}}\approx\frac{1}{k}\int_{t_n}^{t_n+1}\mathbf{F}(\mathbf{U}(x_i,t))\,\mathrm{d}t\,. \tag{2.8}$$

Además, se puede usar el hecho de que la información se propaga con velocidad finita a través del espacio, usando las conclusiones del tema del domino de dependencia de una ecuación de conservación. Por lo que se puede asumir que el flujo numérico aproximado $F^n_{i-\frac{1}{2}}$ depende únicamente de los valores de U en las celdas adyacentes,

$$F^n_{i-\frac{1}{2}}=F(U^n_{i-1},U^n_i) \tag{2.9}$$

por esta razón, se utiliza el índice fraccionario para indicar dónde se calcula el flujo F.

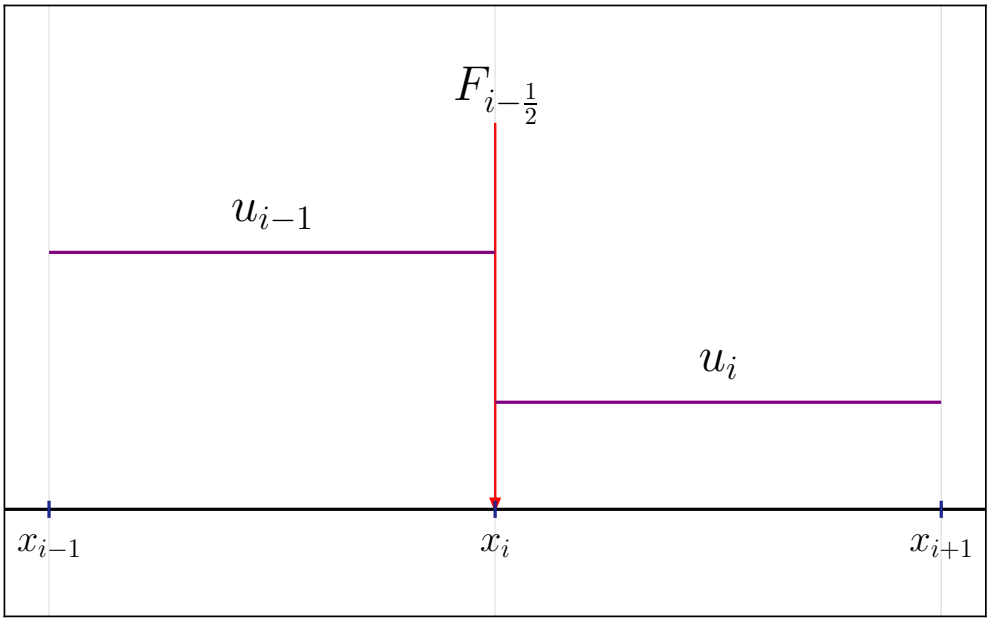


Figura 2.3. Esquema que muestra que el flujo numérico aproximado F se calcula entre las interfaces de las celdas $\mathcal{C}_i$. **Fuente:** elaboración propia.

Si es posible promediar adecuadamente el valor del flujo entre las interfaces de las celdas, se podrá construir un método numérico discreto completo. Sustituyendo las expresiones para las aproximaciones numéricas en (2.7), se obtiene

$$U_i^{n+1} - U_i^n = \frac{k}{h}\left[F_{i-\frac{1}{2}} - F_{i+\frac{1}{2}}\right] \tag{2.10}$$

$$U_i^{n+1} - U_i^n = \frac{k}{h}\left[F(U_{i-1}^n, U_i^n) - F(U_i^n, U_{i+1}^n)\right] \tag{2.11}$$

de tal manera que se obtiene un método iterativo para calcular U en cualquier instante de tiempo t_n.

Como se mencionó previamente, un método numérico de la forma (2.10) es considerado conservativo, dado que imita la propiedad (2.7) de la solución exacta, pero en forma discreta. Si se suman todos los valores de U y F sobre todas las celdas, desde la celda $\mathcal{C}_L$ hasta la celda $\mathcal{C}_R$ a partir de la expresión (2.11), se obtiene

$$\sum_{i=L}^{R}\left[U_i^{n+1} - U_i^n\right] - \frac{k}{h}\sum_{i=L}^{R}\left[F(U_{i-1}^n, U_i^n) - F(U_i^n, U_{i+1}^n)\right] = 0 \tag{2.12}$$

$$\sum_{i=L}^{R}\left[U_i^{n+1} - U_i^n\right] - \frac{k}{h}\left[F(U_{R-1}^n, U_R^n) - F(U_L^n, U_{L+1}^n)\right] = 0 \tag{2.13}$$

o bien, utilizando la notación de medios enteros

$$\sum_{i=L}^{R}\left[U_i^{n+1} - U_i^n\right] - \frac{k}{h}\left[F_{R-\frac{1}{2}}^n - F_{L+\frac{1}{2}}^n\right] = 0. \tag{2.14}$$

A partir de este último resultado se puede concluir que la diferencia entre la suma de los valores de U en un conjunto de celdas consecutivas varía únicamente de acuerdo a los flujos sobre las fronteras de las celdas exteriores. En caso las celdas exteriores fueran las que limitan el dominio completo, se tendrían que invocar las **condiciones de frontera** adecuadas para el problema a resolver.

2.2. Esquemas de flujo numérico

El método iterativo utilizado para resolver la ecuación de conservación depende de cómo esté definida la función de flujo numérico $F_{i+\frac{1}{2}}^n$ en la interfaz entre dos celdas, por lo que es necesario definir un **esquema** que calcule F basándose en los valores de la función U adyacentes. A continuación se presentan algunos esquemas

que proporcionan una función de flujo numérico **escalar**.

2.2.1. Esquema de Lax-Friedrichs

Para definir una función de flujo numérico aproximada, que dependa de los valores de u_i^n en dos celdas vecinas, se puede considerar promediar el valor de la función de flujo exacta valuada en los valores de u en ambas celdas, esto es

$$F(u_{i-1}, u_{i+1}) = \frac{1}{2}\left(f(u_{i-1}) + f(u_{i+1})\right) \tag{2.15}$$

de tal manera que al sustituir en (2.11) se obtiene

$$u_i^{n+1} = u_i^n + \frac{k}{2h}\left[f(u_{i-1}^n) - f(u_{i+1}^n)\right]. \tag{2.16}$$

Sin embargo este esquema no es estable numéricamente para ningún valor de h/k. En cambio al usar el siguiente flujo modificado

$$F(u_{i-1}, u_{i+1}) = \frac{1}{2}\left(f(u_{i-1}) + f(u_{i+1})\right) + \frac{h}{2k}\left(u_{i-1} - u_{i+1}\right) \tag{2.17}$$

se obtiene el esquema de **Lax-Friedrichs** al sustituir en (2.11),

$$u_i^{n+1} = \frac{1}{2}\left(u_{i-1}^n + u_{i+1}^n\right) + \frac{h}{2k}\left(f(u_{i-1}^n) - f(u_{i+1}^n)\right) \tag{2.18}$$

Este esquema produce un flujo correcto a primer orden. El término añadido al flujo en (2.17) es un flujo difusivo que funciona agregando una pequeña viscosidad artificial a la dinámica de la solución.

A continuación se presentan dos esquemas que se basan en la solución del problema de Riemann en forma exacta y aproximada, respectivamente.

2.2.2. Esquema de Godunov

El esquema propuesto por Godunov consiste básicamente en resolver el problema de Riemann de la ecuación de conservación en cuestión para cada celda del dominio con el objetivo de calcular el flujo numérico adecuado en las interfaces de las celdas. La ventaja de este esquema es que al resolver el problema de Riemann en distintos intervalos espaciales, este método resulta ser conservativo, puesto que la solución al problema de Riemann satisface la ecuación de conservación como una

solución débil.

El esquema de Godunov define una función por partes $\tilde{u}^n(x, t_n)$ que toma el valor de u_i^n para cada $x \in \mathcal{C}_i$ como un valor constante (La idea de interpretar a u_i^n como una función por partes fue discutida en la figura (2.2)), y está definida para un intervalo temporal $t_n \leq t \leq t_{n+1}$. De tal manera que $\tilde{u}^n(x, t_n)$ se toma como la condición inicial para resolver la ecuación de conservación en cuestión en dicho intervalo de tiempo, por tanto, resolviendo una secuencia de problemas de Riemann.

Ya que la idea de un esquema numérico el contexto de los métodos de volúmenes finitos es producir un flujo numérico adecuado, el esquema de Godunov redefine la función de flujo aproximado (2.8) como la integral del flujo exacto en la coordenada x_i dado al valuar la la función por partes $\tilde{u}^n(x, t_n)$ en la función de flujo, esto es

$$F^n_{i-\frac{1}{2}} = \frac{1}{k} \int_{t_n}^{t_{n+1}} f(\tilde{u}^n(x_i, t)) \, \mathrm{d}t \, . \tag{2.19}$$

La diferencia entre esta integral y la definida en (2.8) es que la solución de la última es trivial. Esto se justifica con el hecho de que todas las soluciones al problema de Riemann en $x = x_i$ son soluciones de **similitud**, i.e., son constantes en las rectas dadas por $(x - x_i)/t = c$, con c constante. Entonces, $\tilde{u}^n(x_i, t)$ es constante en el tiempo y la integral es trivial.

Sea $u^*(u_L; u_R)$ la solución al problema de Riemann sobre la recta $x/t = 0$, que se obtiene con la siguiente condición inicial

$$u(x, 0) = \begin{cases} u_L & \text{si } x < 0 \\ u_R & \text{si } x > 0 \end{cases} \tag{2.20}$$

entonces, se define a $\tilde{u}^n(x, t)$ como

$$\tilde{u}^n(x_i, t) \equiv u^*(u^n_{i-1}; u^n_i). \tag{2.21}$$

Por tanto, el flujo numérico se reduce a

$$F(u^n_{i-1}, u^n_i) = f(u^*(u^n_{i-1}; u^n_i)) \tag{2.22}$$

y el esquema de Godunov aplicado al MVF resulta en

$$u_i^{n+1} = u_i^n + \frac{h}{k} \left[f(u^*(u^n_{i-1}; u^n_i)) - f(u^*(u^n_i; u^n_{i+1})) \right] . \tag{2.23}$$

2.2.2.1. Esquema de Godunov para el caso escalar

Para definir la función $u^*(u_L; u_R)$ basta con recordar que el problema de Riemann tiene una solución que corresponde a una discontinuidad que se traslada con una velocidad $s = \frac{f(u_L)-f(u_R)}{u_L-u_R}$, de acuerdo a la condición Rankine - Hugoniot. Por tanto, es conveniente proponer una solución de este tipo para u^*;

$$u^*(u_L; u_R) = \begin{cases} u_L & \text{si } s \geq 0 \\ u_R & \text{si } s < 0. \end{cases} \tag{2.24}$$

Sin embargo, esta construcción no garantiza que se satisfaga la condición de entropía. Para corregir esto, se debe considerar usar la solución débil que satisface la condición de entropía, que puede consistir en una onda de choque o en una onda de rarefacción. En caso $f(u)$ sea una función convexa, es decir que $f''(u) > 0$ para todo u, se deben considerar cuatro casos:

1. $f'(u_L), f'(u_R) \geq 0 \;\Rightarrow\; u^* = u_L$ (2.25)
2. $f'(u_L), f'(u_R) \leq 0 \;\Rightarrow\; u^* = u_R$ (2.26)
3. $f'(u_L) \geq 0 \geq f'(u_R) \;\Rightarrow\; u^* = \begin{cases} u_L & \text{si } s > 0 \\ u_R & \text{si } s < 0. \end{cases}$ (2.27)
4. $f'(u_L) < 0 < f'(u_R) \;\Rightarrow\; u^* = u_s.$ (2.28)

Los primeros 3 casos están dados correctamente por (2.24). El caso número 4 corresponde a una solución conocida como *onda de rarefacción transónica* y la función u^* toma el valor intermedio u_s. Este valor se conoce como *punto sónico* y representa el valor de u para el cual la velocidad característica es cero. Esto es,

$$f'(u_s) = 0. \tag{2.29}$$

Entonces, se puede definir el flujo del esquema de Godunov como

$$F(u_L, u_R) = \begin{cases} \text{mín}_{u_L \leq u \leq u_R}\ f(u) & \text{si } u_L \leq u_R \\ \text{máx}_{u_R \leq u \leq u_L}\ f(u) & \text{si } u_L > u_R \end{cases} \tag{2.30}$$

para luego sustituir en (2.11) y concluir con la aplicación de este esquema en el caso escalar.

2.2.2.2. Condición de estabilidad del esquema de Godunov

Ahora bien, es posible que el valor de la solución $\tilde{u}^n(x_i, t)$ no sea constante en x_i por el efecto de otras ondas que se propagan e interferirían con la solución. Sin embargo, las velocidades de las ondas están limitadas a los autovalores de la matriz jacobiana $f'(u)$ y dado que los intervalos que separan cada dominio en que se resuelve cada problema de Riemann independiente están separados por una distancia h, se puede asumir que $\tilde{u}^n(x_i, t)$ será constante en el intervalo de tiempo $[t_n, t_{n+1}]$ siempre que k sea lo suficientemente pequeño. Matemáticamente, esto se describe como

$$\left|\frac{h}{k}\lambda_p(u_i^n)\right| \leq 1 \tag{2.31}$$

para todos los autovalores λ_p en cada u_i^n. En el caso escalar, esta condición equivale a

$$\left|\frac{h}{k}f'(u_i^n)\right| \leq 1. \tag{2.32}$$

2.2.3. Esquema de Roe

El esquema de Roe es un método numérico basado en la resolución aproximada del problema de Riemann, a diferencia del esquema de Godunov, que es un solucionador exacto.

El esquema de Godunov resuelve el problema de Riemann sobre cada celda del dominio discretizado y genera el flujo numérico apropiado entre las interfaces de las celdas del dominio, para cada instante de tiempo. Sin embargo, el valor obtenido para la función en cuestión, en cada celda es el **promedio** de la función exacta en dicho intervalo; y esto produce error numérico. Por otro lado, la resolución del problema de Riemann es usualmente una tarea realizable, pero que puede ser costosa y típicamente se necesita de algún otro método numérico para resolver ecuaciones algebraicas no lineales que complementan alguna iteración del cálculo de la solución numérica. Por estas razones, se considerará utilizar y estudiar a fondo un esquema aproximado como solucionador de Riemann, ya que involucra medios más prácticos para obtener la solución numérica a un sistema de conservación.

2.2.3.1. Idea general de una solución aproximada

Sea $\hat{\mathbf{U}}(x, t)$ una solución aproximada al problema de Riemann para la ecuación de conservación $\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0$, entre dos celdas del dominio discretizado; cada una con valores iniciales $\mathbf{U}_L$ y $\mathbf{U}_R$ para $\mathbf{U}$. Como se mencionó en la sección 2.2.2, toda

solución al problema de Riemann puede escribirse como una función que depende solamente de x/t. Entonces se define la solución aproximada como una función de mencionado tipo, $\hat{\mathbf{U}}(x,t) = \hat{\mathbf{w}}(x/t)$. Para que esta solución sea conservativa, debe cumplirse la siguiente condición para todo $\mathcal{M}$ suficientemente grande:

$$\int_{-\mathcal{M}}^{\mathcal{M}} \hat{\mathbf{w}}(\xi)\,\mathrm{d}\xi = \mathcal{M}(\mathbf{U}_L + \mathbf{U}_R) + \mathbf{F}(\mathbf{U}_L) + \mathbf{F}(\mathbf{U}_R) \tag{2.33}$$

cabe destacar que utilizando los resultados de la sección 1.7.1.1 se puede demostrar que la solución exacta del problema de Riemann satisface esta condición cuando se integra sobre $[\mathcal{M},\mathcal{M}]\times[0,1]$. Si $\hat{\mathbf{w}}$ satisface la igualdad anterior entonces se puede aplicar el MVF para encontrar la solución aproximada. Se puede integrar $\hat{\mathbf{w}}$ de 0 a $\mathcal{M}$, de tal manera que siendo $\mathcal{M}$ suficientemente grande, se obtiene una expresión para el flujo numérico aproximado $F(\mathbf{U}_L,\mathbf{U}_R)$ en términos de $\mathbf{U}_R$

$$\int_{0}^{\mathcal{M}} \hat{\mathbf{w}}(\xi)\,\mathrm{d}\xi = \mathcal{M}\mathbf{U}_R + F(\mathbf{U}_L,\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_R) \tag{2.34}$$

$$F(\mathbf{U}_L,\mathbf{U}_R) = \mathbf{F}(\mathbf{U}_R) - \mathcal{M}\mathbf{U}_R + \int_{0}^{\mathcal{M}} \hat{\mathbf{w}}(\xi)\,\mathrm{d}\xi\,. \tag{2.35}$$

o bien, en términos de $\mathbf{U}_L$

$$F(\mathbf{U}_L,\mathbf{U}_R) = \mathbf{F}(\mathbf{U}_L) + \mathcal{M}\mathbf{U}_L - \int_{-\mathcal{M}}^{0} \hat{\mathbf{w}}(\xi)\,\mathrm{d}\xi\,. \tag{2.36}$$

2.2.3.2. Modificación de la ecuación de conservación

Puesto que se necesita resolver una versión aproximada del problema de Riemann, es natural tener que redefinir la ecuación de conservación a resolver para cada par de celdas como otra ecuación que sea más sencilla de resolver. Sea esta una ecuación de la forma

$$\mathbf{U}_t + \hat{\mathbf{F}}(\mathbf{U})_x = 0 \tag{2.37}$$

donde $\hat{\mathbf{F}}$ es una función de flujo que se aproxima en algún límite a la función original $\mathbf{F}$ y es una función más sencilla de trabajar al momento de resolver el problema de Riemann. Al integrar la ecuación (2.37) sobre el dominio $[-\mathcal{M},\mathcal{M}]\times[0,1]$ se puede concluir que la condición descrita en (2.33) se cumplirá si la siguiente relación entre ambas funciones de flujos también se cumple,

$$\hat{\mathbf{F}}(\mathbf{U}_R) - \hat{\mathbf{F}}(\mathbf{U}_L) = \mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L). \tag{2.38}$$

Entonces, se define a $\hat{\mathbf{w}}(x/t)$ como la solución a la ecuación (2.37). Por tanto, se puede encontrar una expresión para la función de flujo numérico,

$$F(\mathbf{U}_R, \mathbf{U}_L) = \hat{\mathbf{F}}(\hat{\mathbf{w}}(0)) + \mathbf{F}(\mathbf{U}_R) - \hat{\mathbf{F}}(\mathbf{U}_R). \tag{2.39}$$

2.2.3.3. El solucionador de Riemann de Roe

La idea básica del esquema de Roe consiste en resolver el problema de Riemann, sobre cada celda del dominio, para el sistema de ecuaciones de conservación original como un sistema con coeficientes constantes. La resolución de un sistema hiperbólico lineal fue expuesta en la sección 1.9.1. Por tanto, en el esquema de Roe, la función de flujo aproximada $\hat{\mathbf{F}}$ es una función lineal, i.e,

$$\hat{\mathbf{F}}(\mathbf{U}) = \hat{\mathbf{A}}\mathbf{U}. \tag{2.40}$$

Dado que esta definición para la función de flujo depende del problema de Riemann a resolver, es evidente que la matriz $\hat{\mathbf{A}}$ dependerá de los valores de las celdas adyacentes en donde se desea calcular el flujo,

$$\hat{\mathbf{A}} = \hat{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R). \tag{2.41}$$

En general, la definición de la matriz $\hat{\mathbf{A}}$ dependerá del sistema que se busca aproximar. Entonces el sistema lineal que define la solución aproximada $\hat{\mathbf{U}}$ se escribe como

$$\hat{\mathbf{U}}_t + \hat{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R)\hat{\mathbf{U}}_x = 0. \tag{2.42}$$

Como se mostró en la sección 1.9.1 la solución al problema de Riemann de la ecuación (2.42), si $\hat{\mathbf{A}}$ tiene autovalores $\hat{\lambda}_i$ y autovalores $\hat{\mathbf{r}}_i$, se puede expresar utilizando la siguiente expresión

$$\mathbf{U}_R - \mathbf{U}_L = \sum_p \alpha_p \hat{\mathbf{r}}_p. \tag{2.43}$$

Esta última se justifica a través de las expresiones de las soluciones al problema de Riemann para un sistema hiperbólico lineal, según las ecuaciones (1.73) y (1.74). Utilizando las expresiones (1.82) y (1.83) se obtiene

$$\hat{\mathbf{U}}(x,t) = \mathbf{U}_L + \sum_{\hat{\lambda}_p < x/t} \alpha_p \hat{\mathbf{r}}_p \tag{2.44}$$

Haciendo uso de la solución de similitud, $\hat{\mathbf{U}}(x,t) = \hat{\mathbf{w}}(x/t) = \hat{\mathbf{w}}(\xi)$, con $\xi = x/t$, se puede escribir

$$\hat{\mathbf{w}}(\xi) = \mathbf{U}_L + \sum_{\hat{\lambda}_p < \xi} \alpha_p \hat{\mathbf{r}}_p \tag{2.45}$$

donde la suma es sobre todo p tal que $\lambda_p < \xi$. De manera equivalente se puede escribir la solución en términos de $\mathbf{U}_R$,

$$\hat{\mathbf{w}}(\xi) = \mathbf{U}_R - \sum_{\hat{\lambda}_p > \xi} \alpha_p \hat{\mathbf{r}}_p. \tag{2.46}$$

Se comentó que la definición de la matriz aproximada $\hat{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R)$ depende del sistema a resolver. Sin embargo, Roe propone tres condiciones generales que se deben imponer a esta matriz.

La primera condición exige que

$$\text{i)} \quad \hat{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R)(\mathbf{U}_R - \mathbf{U}_L) = \mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L), \tag{2.47}$$

con la cual se garantiza que se cumpla la relación (2.38) que a su vez garantiza la condición que debe cumplir toda solución aproximada, descrita en (2.33). Otro efecto de la anterior condición es asegurar que se cumpla la relación de salto de Rankine-Hugoniot; combinando la ecuación

$$\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L) = s(\mathbf{U}_R - \mathbf{U}_L), \tag{2.48}$$

donde s es la velocidad de la onda de choque o discontinuidad de contacto, con la condición (2.47), se muestra que $\mathbf{U}_R - \mathbf{U}_L$ es un autovector de $\hat{\mathbf{A}}$ con autovalor s. Con este resultado, la solución aproximada $\hat{\mathbf{U}}$ consiste en un salto $\mathbf{U}_R - \mathbf{U}_L$ que se propaga con una velocidad s.

La segunda condición,

$$\text{ii)} \quad \hat{\mathbf{A}} \text{ es diagonalizable con autovalores reales.} \tag{2.49}$$

implica que el sistema (2.42) se puede resolver y es hiperbólico.

La tercera y última condición establece que

$$\text{iii)} \quad \hat{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R) \to \mathbf{F}'(\bar{\mathbf{U}}) \text{ suavemente mientras } \mathbf{U}_L, \mathbf{U}_R \to \bar{\mathbf{U}}. \tag{2.50}$$

La tercera condición garantiza que el método de Roe funcione adecuadamente con

soluciones suaves, en contraste con la primera condición, que garantiza el funcionamiento del método con discontinuidades locales.

2.2.3.4. El flujo numérico de Roe

La función $\hat{\mathbf{w}}(\xi)$ que se encarga de producir el esquema numérico de Roe corresponde a la solución exacta del problema lineal descrito en la ecuación (2.42). Entonces, utilizando la expresión de flujo numérico de (2.39) con el flujo numérico aproximado (2.40), se obtiene

$$F(\mathbf{U}_L, \mathbf{U}_R) = \hat{\mathbf{A}}\hat{\mathbf{w}}(0) + \mathbf{F}(\mathbf{U}_R) - \hat{\mathbf{A}}\mathbf{U}_R \tag{2.51}$$

sustituyendo la solución (2.46) se obtiene,

$$F(\mathbf{U}_L, \mathbf{U}_R) = \mathbf{F}(\mathbf{U}_R) - \hat{\mathbf{A}} \sum_{\hat{\lambda}_p > 0} \alpha_p \hat{\mathbf{r}}_p \tag{2.52}$$

$$F(\mathbf{U}_L, \mathbf{U}_R) = \mathbf{F}(\mathbf{U}_R) - \sum_{p=1}^{m} \hat{\lambda}_p^+ \alpha_p \hat{\mathbf{r}}_p \tag{2.53}$$

donde $\hat{\lambda}_p^+ = \text{máx}\,(\hat{\lambda}_p, 0)$. También es posible escribir el flujo numérico en términos de $\mathbf{U}_L$,

$$F(\mathbf{U}_L, \mathbf{U}_R) = \mathbf{F}(\mathbf{U}_L) + \sum_{p=1}^{m} \hat{\lambda}_p^- \alpha_p \hat{\mathbf{r}}_p, \tag{2.54}$$

con $\hat{\lambda}_p^- = \text{mín}\,(\hat{\lambda}_p, 0)$. Combinando ambas expresiones mediante un promedio, se obtiene el flujo numérico de Roe,

$$F(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_L) + \mathbf{F}(\mathbf{U}_R)\right) - \frac{1}{2}\sum_{p=1}^{m} |\hat{\lambda}_p| \alpha_p \hat{\mathbf{r}}_p. \tag{2.55}$$

2.2.3.5. Corrección de entropía sónica

Una desventaja del método de Roe es su incapacidad de producir las soluciones físicamente correctas en el caso que la solución al problema de Riemann corresponda a una onda de rarefacción sónica, ya que puede producir soluciones que violan la condición de entropía. La onda de rarefacción sónica fue expuesta en la ecuación (2.28), considerada como un caso especial por el esquema de Godunov. Por lo tanto, en el contexto del esquema de Roe, este caso se puede determinar cuando se cumple que $\lambda_{p,L} < 0 < \lambda_{p,R}$, donde $\lambda_{p,K} = \lambda_p(\mathbf{U}_K)$. Si esta condición se da, es necesario

modificar el flujo numérico de Roe. Hay varias maneras de corregir esto. El flujo modificado propuesto por Harten y Hyman [7] es el siguiente.

$$F(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_L) + \mathbf{F}(\mathbf{U}_R)\right) - \frac{1}{2}\sum_{p=1}^{m} H(\hat{\lambda}_p)\alpha_p \hat{\mathbf{r}}_p. \tag{2.56}$$

donde

$$H(\hat{\lambda}_p) = \begin{cases} \delta_p & \text{si } |\hat{\lambda}_p| < \delta_p \\ |\hat{\lambda}_p| & \text{si } |\hat{\lambda}_p| \geq \delta_p \end{cases} \tag{2.57}$$

con

$$\delta_p = \text{máx}(0, \hat{\lambda}_p - \lambda_{p,L}, \lambda_{p,R} - \hat{\lambda}_p). \tag{2.58}$$

3. ECUACIONES DE EULER Y APLICACIÓN DEL ESQUEMA DE ROE

En este capítulo se explican y derivan las ecuaciones de Euler utilizando las variables generales (presión, densidad y velocidad) y se introducen las variables conservadas. Se explican las ligaduras adicionales involucradas para que las ecuaciones de Euler sean aplicadas a un gas ideal poliatómico.

Se describe el esquema de Roe implementado en la solución de las ecuaciones de Euler para un gas ideal poliatómico así como las demás especificaciones requeridas por el método de volúmenes finitos. Se explica la implementación del método numérico en `C++`. Se muestran los resultados obtenidos para un problema de condiciones iniciales específicas.

3.1. Ecuaciones de Euler

Las ecuaciones fundamentales de la dinámica de fluidos se basan en las siguientes leyes de conservación universales:

- Conservación de la masa
- Conservación del momentum
- Conservación de la energía.

La ecuación de conservación de la masa, que se derivó en la sección 1.2, consiste en aplicar la ecuación de continuidad para un fluido con cierta densidad. La ley de conservación del momentum resulta al aplicar la Segunda Ley de Newton en un fluido. Por último, la ley de conservación de la energía es equivalente a la aplicación de la Primera Ley de la Termodinámica. Además de las tres ecuaciones de conservación, es necesario establecer una relación entre las variables físicas del fluido, de tal manera que el sistema de ecuaciones sea resoluble [8]. En el caso de un gas ideal, la

ecuación adicional que relaciona las variables densidad ρ, presión p y temperatura T, es la ecuación de estado.

3.1.1. Derivación de las ecuaciones

La ecuación de continuidad, derivada en la sección 1.2 tiene la siguiente forma

$$\rho_t + (\rho v)_x = 0. \tag{3.1}$$

Ya que el flujo asociado a esta ecuación de conservación es ρv, se puede interpretar que, generalmente, para cualquier cantidad física z que sufra advección, su flujo estará dado por el producto de la cantidad por la velocidad de advección, i.e., $f = zv$.

A partir del último razonamiento, el flujo asociado a la advección de momentum ρv, tiene una contribución al flujo dada por $(\rho v) \cdot v = \rho v^2$. Sin embargo, además de la advección que sufre el momentum, deben considerarse las fuerzas que actúan en el fluido para expresar el flujo total del momentum. En este texto no se considerarán fuerzas externas, por lo que únicamente habría que tomar en cuenta la fuerza interna del fluido, que está dada por el gradiente de **presión**, p_x [6]. De esta manera, se consigue la ecuación de conservación del momentum:

$$(\rho v)_t + (\rho v^2 + p)_x = 0. \tag{3.2}$$

Para derivar la conservación de energía, se debe considerar que ésta se compone por un término cinético y uno correspondiente a la energía interna del fluido. Sea E la densidad de energía total de un fluido. Entonces, se tiene que:

$$E = \frac{1}{2}v^2 + e, \tag{3.3}$$

donde el término $\frac{1}{2}v^2$ corresponde a la densidad de energía cinética por unidad de masa y e es la **energía interna** por unidad de masa del fluido, que también suele denominarse como energía interna específica. La energía interna depende de los grados de libertad internos de las moléculas de los gases, considerando energía de rotación, cinética, de vibración y otras formas más complejas de energía. Las ecuaciones de Euler suponen que la ecuación de estado provee una expresión para la energía interna específica tal que ésta depende de la presión y la densidad del fluido únicamente

$$e = e(p, \rho). \tag{3.4}$$

De igual manera que con el momentum, la energía total se ve afectada por la advección del flujo del fluido. Dado que no se consideran fuerzas externas que afecten al sistema, únicamente la presión del fluido hace trabajo y es proporcional al gradiente de vp. Entonces la ecuación de conservación de la energía se reduce a:

$$(\rho E)_t + [v(\rho E + p)]_x = 0. \tag{3.5}$$

Haciendo notar que ρE corresponde a la energía total del gas.

Las ecuaciones de Euler, (3.1), (3.2) y (3.5) se pueden escribir en forma vectorial, obteniendo:

$$\begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix}_t + \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(\rho E + p) \end{bmatrix}_x = 0, \tag{3.6}$$

cuya forma coincide con la presentada en (1.1). Cabe mencionar que en este texto se considerarán problemas unidimensionales solamente.

3.1.2. Ecuación de estado para un gas politrópico

Para completar el sistema de ecuaciones de Euler, es necesario definir la ecuación de estado que relacione la energía con las variables físicas de presión y densidad. El desarrollo de la ecuación de estado para un gas ideal sigue de cerca la sección *Ideal Gas* del capítulo *Some Linear Systems* en [6].

La energía interna de un gas ideal es únicamente dependiente de la temperatura,

$$e = e(T). \tag{3.7}$$

Mientras que la temperatura se relaciona con la presión y densidad a través de la ley del gas ideal, que define al mismo. Esta es:

$$p = \mathcal{R}\rho T \tag{3.8}$$

donde $\mathcal{R}$ es la constante específica de los gases, que se define como el cociente entre la constante de Boltzmann k_B y la masa de cada molécula del gas m [1]. Por otro lado, la energía interna específica es proporcional a la temperatura,

$$e = c_v T \tag{3.9}$$

donde c_v es la capacidad calorífica específica a volumen constante. Los gases que

cumplen con esta propiedad se conocen como gases **politrópicos**. Entonces, si se cambiase la temperatura de un gas en una cantidad infinitesimal $\mathrm{d}T$, manteniendo el volumen constante, se obtendría

$$\mathrm{d}e = c_v \, \mathrm{d}T \tag{3.10}$$

como cambio infinitesimal de energía interna específica. En cambio, si se permite que el gas se expanda pero manteniendo ahora la presión constante, se obtendría una expresión para el cambio de energía considerando al trabajo, $\mathrm{d}W = -p \, \mathrm{d}V$, realizado sobre el gas, esto es

$$m \, \mathrm{d}e = \mathrm{d}W + đQ \tag{3.11}$$

$$m \, \mathrm{d}e = -p \, \mathrm{d}V + đQ \tag{3.12}$$

$$\mathrm{d}e = -\frac{p}{m} \, \mathrm{d}V + c_p \, \mathrm{d}T \tag{3.13}$$

y puesto que al usar $\rho = \frac{m}{V}$, se obtiene que $\mathrm{d}V = m \, \mathrm{d}\left(\frac{1}{\rho}\right)$, entonces:

$$\mathrm{d}e = -p \, \mathrm{d}\left(\frac{1}{\rho}\right) + c_p \, \mathrm{d}T \tag{3.14}$$

$$\mathrm{d}\left(e + \frac{p}{\rho}\right) = c_p \, \mathrm{d}T \, , \tag{3.15}$$

donde c_p es la capacidad calorífica específica del gas a presión constante. A partir del anterior resultado se define la **entalpía** interna h_i:

$$\mathrm{h}_i \equiv e + \frac{p}{\rho}, \tag{3.16}$$

mientras que la entalpía total h [1] es:

$$\mathrm{h} = E + \frac{p}{\rho}. \tag{3.17}$$

Para un gas politrópico se considera a c_p como constante, por lo que integrando la ecuación (3.15) se obtiene otra expresión para la entalpía interna,

$$\mathrm{h}_i = c_p T. \tag{3.18}$$

[1] No confundir con h definida en 2.1 como el tamaño de cada celda del dominio.

Por otro lado, de acuerdo a la ley del gas ideal, se tiene la siguiente relación

$$c_p - c_v = \mathcal{R}. \tag{3.19}$$

Para continuar con la derivación, es conveniente definir el **coeficiente de dilatación adiabática** γ como:

$$\gamma \equiv \frac{c_p}{c_v}. \tag{3.20}$$

Dicha cantidad está estrechamente relacionada con el número de grados de libertad internos del gas, que depende de la naturaleza del mismo. Según el teorema de equipartición de la energía, el promedio de energía involucrada en cada grado de libertad es el mismo. Específicamente, cada grado de libertad aporta una cantidad promedio de energía por molécula de $\frac{1}{2}k_BT$. De tal manera que si existen α grados de libertad internos en un gas que tiene n moléculas por unidad de masa, se obtiene una expresión para la energía interna específica

$$e = \frac{\alpha}{2} n k_B T \tag{3.21}$$

o bien,

$$e = \frac{\alpha}{2} \mathcal{R} T. \tag{3.22}$$

Comparando con (3.9) se obtiene

$$c_v = \frac{\alpha}{2} \mathcal{R}, \tag{3.23}$$

sustituyendo en (3.19),

$$c_p = \left(1 + \frac{\alpha}{2}\right) \mathcal{R}. \tag{3.24}$$

Y al aplicar la definición de γ en (3.20) se obtiene la expresión

$$\gamma = \frac{\alpha + 2}{\alpha}, \tag{3.25}$$

que, como se expuso previamente, relaciona el número de grados de libertad con el coeficiente de dilatación adiabática. Por ejemplo, para gases monoatómicos se consideran únicamente tres grados de libertad, correspondientes al movimiento traslacional en tres dimensiones, por lo que $\alpha = 3$ y $\gamma = 5/3$. En gases diatómicos (como el aire, compuesto por H_2 y N_2 principalmente) se agregan dos grados libertad correspondientes a dos ejes de rotación posibles para cada molécula, de tal manera que $\alpha = 5$ y $\gamma = 7/5 = 1.4$.

Por último, se escribe la ecuación de estado del gas ideal para la energía utilizando (3.8),

$$e = c_v T = \frac{c_v}{\mathcal{R}} \cdot \frac{p}{\rho} \tag{3.26}$$

$$e = \frac{p}{(\gamma - 1)\rho}. \tag{3.27}$$

De tal manera que la energía total del gas (ρE) queda como

$$\rho E = \frac{1}{2}\rho v^2 + \frac{p}{\gamma - 1}. \tag{3.28}$$

3.2. Aplicación del esquema de Roe

Para aplicar el esquema de Roe en la expresión del método de volúmenes finitos, dada por

$$U_i^{n+1} - U_i^n = \frac{k}{h}\left[F(U_{i-1}^n, U_i^n) - F(U_i^n, U_{i+1}^n)\right], \tag{3.29}$$

es necesario definir el flujo entre cada interfaz, $F_{i\pm\frac{1}{2}}$. La expresión de dicho flujo, en su forma simple, se detalló en (2.55). Mientras que la versión del flujo en conjunto con la corrección de entropía es (2.56). Por tanto, el siguiente paso es construir el flujo de Roe de acuerdo a las aproximaciones correspondientes.

3.2.1. Variables conservadas y propiedades de $\mathbf{A}(\mathbf{U})$

Para implementar el método de volúmenes finitos junto al esquema de Roe en la solución numérica de las ecuaciones de Euler es necesario identificar las **variables conservadas** involucradas en la definición de un sistema general de conservación, definido en (1.2). Comparando esta expresión con el sistema (3.6) se obtiene:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{bmatrix} \equiv \begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix} \tag{3.30}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{bmatrix} \equiv \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(\rho E + p) \end{bmatrix}. \tag{3.31}$$

Por tanto, se define a $\mathbf{u}_i$ como la i-ésima variable conservada cuyo flujo correspondiente es $\mathbf{f}_i$.

Por otro lado, el esquema de Roe depende de los autovalores y autovectores de la matriz jacobiana $\mathbf{A}(\mathbf{U})$ definida en (1.7). Aplicando la definición (1.6) mientras se toma en cuenta la ecuación de estado para la energía (3.8), se obtiene una expresión para esta matriz:

$$\mathbf{A}(\mathbf{U}) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)v^2 & (3-\gamma)v & (\gamma-1) \\ \frac{1}{2}(\gamma-1)v^3 - v(\rho E + p)/\rho & (\rho E + p)/\rho - (\gamma-1)v^2 & \gamma v \end{bmatrix}. \quad (3.32)$$

Seguidamente, se deben calcular los autovalores y autovectores de esta matriz. Siguiendo la notación de los últimos capítulos, al autovalor λ_i le corresponde el autovector $\mathbf{r}_i$, y estos son:

$$\lambda_1 = v - c, \quad \lambda_2 = c, \quad \lambda_3 = v + c, \quad (3.33)$$

$$\mathbf{r}_1 = \begin{bmatrix} 1 \\ v - c \\ \mathrm{h} - vc \end{bmatrix}, \quad \mathbf{r}_2 = \begin{bmatrix} 1 \\ v \\ \frac{1}{2}v^2 \end{bmatrix}, \quad \mathbf{r}_3 = \begin{bmatrix} 1 \\ v + c \\ \mathrm{h} + vc \end{bmatrix}. \quad (3.34)$$

donde c es la velocidad del sonido, dada por:

$$c = \sqrt{\frac{\gamma p}{\rho}}, \quad (3.35)$$

mientras h es la entalpía total definida en (3.17), o bien, equivalentemente:

$$\mathrm{h} = \frac{1}{2}v^2 + \left(\frac{\gamma}{\gamma-1}\right)\frac{p}{\rho}. \quad (3.36)$$

Por lo tanto, es posible escribir la velocidad del sonido en términos de la entalpía y la velocidad:

$$c^2 = (\gamma-1)\left[\mathrm{h} - \tfrac{1}{2}v^2\right]. \quad (3.37)$$

3.2.2. Valores promediados de Roe

Retornando a la expresión para el flujo numérico simple de Roe,

$$F(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_L) + \mathbf{F}(\mathbf{U}_R)\right) - \frac{1}{2}\sum_{p=1}^{m} |\hat{\lambda}_p| \alpha_p \hat{\mathbf{r}}_p, \quad (3.38)$$

se destaca que es necesario encontrar los autovalores $\hat{\lambda}_i$ y autovectores $\hat{r}_i$ correspondientes a la matriz aproximada $\hat{\mathbf{A}}$. Roe [9] propuso expresiones, basadas en promedios específicos de los valores adyacentes, para los mencionados autovectores y autovalores de la matriz aproximada y junto a Pike demostraron que son los únicos promedios que satisfacen las condiciones necesarias impuestas al flujo numérico del esquema de Roe [10]. Los autovalores y autovectores de Roe son:

$$\hat{\lambda}_1 = \tilde{v} - \tilde{c}, \quad \hat{\lambda}_2 = \tilde{c}, \quad \hat{\lambda}_3 = \tilde{v} + \tilde{c}, \tag{3.39}$$

$$\hat{\mathbf{r}}_1 = \begin{bmatrix} 1 \\ \tilde{v} - \tilde{c} \\ \tilde{\mathrm{h}} - \tilde{v}\tilde{c} \end{bmatrix}, \quad \hat{\mathbf{r}}_2 = \begin{bmatrix} 1 \\ \tilde{v} \\ \frac{1}{2}\tilde{v}^2 \end{bmatrix}, \quad \hat{\mathbf{r}}_3 = \begin{bmatrix} 1 \\ \tilde{v} + \tilde{c} \\ \tilde{\mathrm{h}} + \tilde{v}\tilde{c} \end{bmatrix}. \tag{3.40}$$

Mientras que los coeficientes de las características α_i están dados por:

$$\alpha_1 = \frac{1}{2\tilde{c}^2}[\Delta p - \tilde{\rho}\tilde{c}\Delta v], \quad \alpha_2 = \frac{1}{\tilde{c}^2}\left[\tilde{c}^2\Delta\rho - \Delta p\right], \quad \alpha_3 = \frac{1}{2\tilde{c}^2}[\Delta p + \tilde{\rho}\tilde{c}\Delta v], \tag{3.41}$$

con

$$\tilde{\rho} = \sqrt{\rho_L \rho_R} \tag{3.42}$$

$$\tilde{v} = \frac{\sqrt{\rho_L}\cdot v_L + \sqrt{\rho_R}\cdot v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{3.43}$$

$$\tilde{\mathrm{h}} = \frac{\sqrt{\rho_L}\cdot \mathrm{h}_L + \sqrt{\rho_R}\cdot \mathrm{h}_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{3.44}$$

$$\tilde{c}^2 = (\gamma - 1)[\tilde{h} - \tfrac{1}{2}\tilde{v}^2] \tag{3.45}$$

donde $\Delta u = u_R - u_L$, para cualquier variable u, siendo u_R y u_L el valor de la variable a la derecha e izquierda de la interfaz en donde se calcula el flujo, respectivamente.

3.3. Características básicas de la simulación

Para resolver las ecuaciones de Euler es necesario especificar algunas características del medio y del espacio en el que toman lugar las simulaciones a presentar en este texto.

3.3.1. Dimensiones

Las ecuaciones de Euler serán resueltas en una dimensión espacial, que se puede interpretar como la evolución temporal de un gas dentro de un tubo. Naturalmente,

un tubo no es exactamente un objeto unidimensional (un objeto unidimensional es una idealización) pero se aproximará como tal para efectuar la simulación. Este dominio posee una longitud de 10m, de tal manera que definiendo los parámetros de la sección 2.1.1, se tiene:

$$a = 0\text{m}, \tag{3.46}$$

$$b = 10\text{m}. \tag{3.47}$$

Es natural entonces definir una variable para el largo del dominio, $L \equiv b - a$. Mientras que el número de celdas N a utilizar será:

$$N = 500. \tag{3.48}$$

De tal forma que el tamaño de cada celda está dado por:

$$h = \frac{10\text{m}}{500} = 0.02\text{m}. \tag{3.49}$$

El tiempo total por defecto será de 4s, y se producirá la solución para un número determinado de instantes temporales en total, que generalmente no coincidirá con el número de instantes temporales calculados. Esto se hace para optimizar el tiempo de cómputo. Por esta razón, el tamaño de paso temporal k puede estar sujeto a cambios dependiendo de la simulación, ya que en caso no se cumpla la condición de estabilidad (2.32) deberán recalibrarse los parámetros temporales.

A excepción de los resultados mostrados en el capítulo 5, se tomará el coeficiente de dilatación adiabática como el valor aproximado que posee el aire, $\gamma = 1.4$ [6].

3.3.2. Condiciones iniciales

Se experimentará con tres distintos conjuntos de condiciones iniciales para las ecuaciones de Euler.

3.3.2.1. Primer conjunto

Este conjunto corresponde a una solución trivial. Las condiciones iniciales de las variables físicas son las siguientes

$$\rho(x,0) = \left(1.0 + \exp\left[-(x - \tfrac{L}{2})^2\right]\right) \text{ kg/m}^3 \tag{3.50}$$

$$v(x,0) = 1.0 \,\mathrm{m\,s^{-1}} \tag{3.51}$$

$$p(x,0) = 0.5 \text{ Pa} \tag{3.52}$$

Notando que tanto la velocidad como la presión son constantes, se espera que el perfil gaussiano definido para la densidad se traslade a velocidad constante sobre el eje x. Esto se deriva del hecho de que la ecuación de continuidad (3.1) se reduce a una ecuación de advección lineal en este caso.

Los siguientes conjuntos de condiciones iniciales corresponden a funciones definidas por partes. Las variables tendrán un valor constante que depende de la mitad del dominio en donde se valúen. Se adoptará la siguiente notación para escribir de manera compacta las condiciones iniciales:

$$\mathbf{V}(x,0) = \begin{cases} \mathbf{V}_L & \text{si } x \leq L/2 \\ \mathbf{V}_R & \text{si } x > L/2, \end{cases} \tag{3.53}$$

donde

$$\mathbf{V}(x,t) = \begin{bmatrix} \rho \\ v \\ P \end{bmatrix}, \tag{3.54}$$

es el vector de variables físicas independientes.

3.3.2.2. Segundo conjunto

Este conjunto de condiciones iniciales se denomina Test de Sod [11]. Consiste en los siguientes valores para las variables físicas independientes [2]

$$\mathbf{V}_L = \begin{bmatrix} 3.0 \\ 0.0 \\ 3.0 \end{bmatrix}, \quad \mathbf{V}_R = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix}. \tag{3.55}$$

[2] Se omiten las dimensiones de las variables para mantener la legibilidad del texto, sin embargo, todas las unidades utilizadas corresponden a las del Sistema Internacional.

En este conjunto de condiciones iniciales se destaca la diferencia de presión en cada mitad del dominio, así como de la densidad, que va de mayor a menor cantidad. La velocidad inicial del gas es nula.

3.3.2.3. Tercer conjunto

El siguiente conjunto de condiciones iniciales fue propuesto por LeVeque para evaluar el funcionamiento del esquema numérico utilizado cuando se presentan ondas de rarefacción transónica [7]. Es similar al test de Sod, pero con la diferencia que la velocidad inicial del gas no es nula. Los valores iniciales son:

$$\mathbf{V}_L = \begin{bmatrix} 3.0 \\ 0.9 \\ 3.0 \end{bmatrix}, \quad \mathbf{V}_R = \begin{bmatrix} 1.0 \\ 0.9 \\ 1.0 \end{bmatrix}. \tag{3.56}$$

3.3.3. Condiciones de frontera

Previo a poder aplicar el método de volúmenes finitos (MVF) junto al esquema de Roe, es necesario especificar las condiciones de frontera en el dominio espacial. De acuerdo a la expresión (3.29), generalmente el MVF se aplica sobre un dominio dividido en N celdas, de manera que el índice i se recorre de 1 a N. Por lo tanto, es necesario definir los valores de la frontera, siendo estos U_0^n y U_{N+1}^n, para que los flujos $F_{\frac{1}{2}}$ y $F_{N+\frac{1}{2}}$ estén bien definidos y correspondan a flujos físicamente consistentes.

3.3.3.1. Condiciones transmisivas

Si se asume que el gas se encuentra dentro de un tubo con las paredes extremas abiertas, se pueden considerar condiciones que permitan la transmisión del medio al exterior [11]. Las condiciones de frontera transmisivas se puede conseguir con las siguientes igualdades:

$$\rho_0^n = \rho_1^n, \quad \rho_{N+1}^n = \rho_N^n \tag{3.57}$$

$$v_0^n = v_1^n, \quad v_{N+1}^n = v_N^n \tag{3.58}$$

$$p_0^n = p_1^n, \quad p_{N+1}^n = p_N^n. \tag{3.59}$$

3.4. Código implementado

Como fue comentado previamente, la solución numérica de las ecuaciones de Euler se implementó en un programa de `C++`. El código fuente completo de la simulación se encuentra disponible en `https://github.com/highchen147/tesis/blob/main/euler1D/euler1D.cpp`.

3.4.1. Librerías y paquetes utilizados

Se utilizaron las siguientes librerías en el desarrollo del programa integrador:

Librerías

```
#include <iostream>
#include <string>
#include <ctime>
#include <cstdlib>
#include <cmath>
#include <iomanip>
#include <fstream>
#include <vector>
#include <sys/stat.h>
#include <algorithm>
#include "funciones.hpp" // Incluye funciones predefinidas
using namespace std;
```

En el paquete con nombre `funciones.hpp` se definió una función por partes, con nombre `step_neg`, para aplicar las condiciones iniciales de forma práctica.

Listing 3.1. Función `step_neg`

```
double step_neg(double x, double max, double min, double x_0)
{
    if (x <= x_0)
    {
        return max;
    }
    else
    {
```

```
        return min;
    }
}
```

3.4.2. Definición de parámetros principales

Se definió el coeficiente de dilatación adiabática como una variable global, esto con el fin de evitar escribir muchas funciones en donde dicho coeficiente fuera un parámetro de las mismas y así contar con una variable constante y segura para toda la simulación.

Listing 3.2. Definición de γ

```
const double Gamma = 1.4;
```

Se definieron seis parámetros temporales en la función `int main()` del programa:

Listing 3.3. Parámetros temporales

```
const double t_total = 4; // Tiempo total en segundos
const double dt = 0.005; // Tamaño de paso temporal en segundos
int Niter = floor(t_total/dt); // Número total de iteraciones
const int num_outs = 400; // Número de instantes temporales producidos
int out_cada = floor(Niter / num_outs); // Cada out_cada veces se
                                        // imprimen los valores
double tiempo = 0.0; // Variable de tiempo en la simulación
```

Los parámetros `t_total`, `dt` y `num_outs` son modificables de acuerdo a lo necesario para conseguir una simulación satisfactoria. `t_total` es el tiempo total de la simulación, mientras que `dt` es el tamaño de paso temporal que fue definido en el capítulo 2 como k. El parámetro `Niter` se define como el número de iteraciones necesarias temporales para conseguir una simulación que dure lo preestablecido y coincida con el tamaño de paso `dt`; además, se usa la función `floor()` para garantizar un número entero de iteraciones.

La variable `num_outs` almacena el número de soluciones producidas para estados temporales determinados, es decir, se imprimen los datos de las funciones calculadas para `num_outs` instantes temporales, que no necesariamente coincidirá con el número de iteraciones. Luego se define `out_cada`, que calcula cada cuántas iteraciones se deberán imprimir los datos. Por último, la variable `tiempo` almacena el tiempo real de la simulación.

Los parámetros espaciales, que fueron descritos en la sección 3.3.1 se definieron de la siguiente manera en el código:

Listing 3.4. Parámetros espaciales

```
int Nx = 500; // Número de celdas en el eje x
double L = (10); // Largo del dominio en metros
double dx = L/(Nx); // Tamaño de paso en el eje x
```

Por último, se definió una variable de tipo booleana, para activar o desactivar la corrección de entropía. En otras palabras, esta variable decide si se usa el flujo simple de Roe (2.55) o el flujo modificado para corregir el caso de entropía sónica (2.56).

Listing 3.5. Corrección de entropía

```
bool correccion_de_entropia = true;
```

3.4.3. Objetos definidos

Las soluciones numéricas son aproximaciones en forma discreta, por lo que para construir las funciones de magnitudes físicas en el programa, se definieron punteros a arreglos unidimensionales, con un tamaño que coincidiera con el número de celdas definido para la simulación [3].

Listing 3.6. Arreglos de variables físicas

```
// Cantidades físicas
// Densidad
double *rho = new double[Nx];
double *rho_nueva = new double[Nx];
// Velocidad
double *u = new double[Nx];
double *u_nueva = new double[Nx];
// Presión
double *p = new double[Nx];
double *p_nueva = new double[Nx];
```

Las variables `rho`, `u` y `p` corresponden a las funciones de densidad, velocidad y presión valuadas en el instante temporal n, respectivamente. Las variables cuyo

[3] A diferencia del resto del texto, se definió a la velocidad del gas como `u`.

nombre lleva el sufijo `_nueva` corresponden a la cantidad valuada en el instante temporal siguiente $n+1$, i.e,

$$\rho_i^n \leftrightarrow \texttt{rho[i]} \tag{3.60}$$

$$\rho_i^{n+1} \leftrightarrow \texttt{rho\_nueva[i]}. \tag{3.61}$$

También se definieron arreglos para las componentes del vector **U**, siendo `ui` la iésima componente de **U**.

Listing 3.7. Componentes vectoriales

```
// Componentes del vector U
double *u1 = new double[Nx];
double *u2 = new double[Nx];
double *u3 = new double[Nx];
```

El vector en cuestión fue implementado como un objeto de la clase `vector` de tipo `double` con dimensión 3. El uso de objetos de dicha clase fue con el objetivo de utilizar la sobrecarga de los operadores básicos (`+`, `-`, `*`) para realizar operaciones entre los vectores prácticamente. La definición de la acción de los operadores se omitirá en este texto, ya que su funcionamiento e implementación son triviales.

Listing 3.8. Componentes vectoriales

```
// Se declaran los vectores principales de la integración
vector<double> U(3);
```

El dominio se dividió en celdas, cuya ubicación en el espacio se almacenó en un puntero.

Listing 3.9. Arreglo de celdas

```
// Celdas sobre el eje x
double *x = new double[Nx];
```

3.4.4. Inicialización de arreglos

Para inicializar cada arreglo previamente definido se utilizaron ciclos iterativos.

Listing 3.10. Inicialización de variables

```
// Dominio espacial
for (int i = 0; i < Nx; i++)
{
```

```
    x[i] = i*dx;
}
// Cantidades físicas
for (int i = 0; i < Nx; i++)
{
    // Densidad
    rho[i] = rho_inicial(x[i], L);
    // Presión
    p[i] = p_inicial(x[i], L);
    // Velocidad
    u[i] = u_inicial(x[i], L);
}
```

Las funciones `rho_inicial`, `p_inicial`, `u_inicial` son variaciones de la función por partes, definida en el código (3.1), que dependen de la condición inicial por aplicar. Por ejemplo, la condición de la sección 3.3.2.3 para la velocidad, se aplica de la siguiente manera:

Listing 3.11. Condición inicial para la velocidad

```
double u_inicial(double x, double L)
{
    return step_neg(x, -1, 1, L/2);
}
```

La inicialización de las componentes de **U** se realizó con una función de tipo `void` para facilitar la escritura en el código del cálculo de las componentes del mismo vector.

Listing 3.12. Cálcuo de las componentes del vector de cantidades conservadas

```
/**
 * @brief Asignar valores a las componentes del vector U
 *
 * @param u1 Componente 1 de U
 * @param u2 Componente 2 de U
 * @param u3 Componente 3 de U
 * @param rho Densidad
 * @param p Presión
 * @param u Velocidad
```

```
 * @param Nx Tamaño de los arreglos que almacenan las funciones
 */
void calc_componentes_U(double *u1,
                        double *u2,
                        double *u3,
                        double *rho,
                        double *p,
                        double *u,
                        int Nx)
{
    for (int i = 0; i < Nx; i++)
    {
        u1[i] = rho[i];
        u2[i] = rho[i]*u[i];
        u3[i] = p[i]/(Gamma-1) + 0.5*rho[i]*pow(u[i], 2);
    }
}
```

Se puede apreciar que los cálculos de las componentes de **U** coinciden con (3.6) tomando en cuenta la definición de la energía (3.28).

3.4.5. Funciones auxiliares del esquema de Roe

Previo a la descripción del ciclo iterativo de la integración numérica, se muestra la construcción de las funciones que completan la implementación en el programa de la expresión (2.11).

La función `flujo_euler` calcula el flujo exacto del sistema de Euler. En otras palabras, es la implementación directa de **F**, definido en (3.31).

Listing 3.13. Definición de la función que calcula el flujo del sistema de Euler

```
/**
 * @brief Calcula el flujo F de la ecuación de Euler en forma
 * conservativa.
 *
 * @param rho Densidad
 * @param p Presión
 * @param u Velocidad
 * @return vector<double>
```

```
 */
vector<double> flujo_euler(double rho, double p, double u)
{
    vector<double> f_resultante(3);
    double F1 = rho*u;
    double F2 = p + rho*pow(u, 2);
    double F3 = u*(p/(Gamma-1) + 0.5*rho*pow(u, 2) + p);
    f_resultante = {F1, F2, F3};
    return f_resultante;
}
```

Cabe destacar que el flujo de Euler se implementa como un objeto de la clase `vector`, de la misma manera que se define `U`, como se expuso en el código (3.8).

Evidentemente, el flujo exacto de Euler no es igual al flujo numérico de Roe, sino que el último depende del primero. El flujo de Roe se construyó por partes en el programa. A continuación se muestra la implementación de $F(\mathbf{U}_L, \mathbf{U}_R)$, correspondiente a la ecuación (2.55).

Listing 3.14. Definición de la función que devuelve el flujo numérico

```
/**
 * @brief Calcula el flujo entre celdas utilizando el esquema de Roe
 *
 * @param F_L Flujo exacto en la celda izquierda
 * @param F_R Flujo exacto en la celda derecha
 * @param p_L Presión a la izquierda
 * @param p_R Presión a la derecha
 * @param u_L Velocidad a la izquierda
 * @param u_R Velocidad a la derecha
 * @param rho_L Densidad a la izquierda
 * @param rho_R Densidad a la derecha
 * @param entropy_fix Parámetro para decidir el uso de la
 * correción de entropía
 * @return vector<double>
 */
vector<double> Flujo(
    vector<double> F_L,
    vector<double> F_R,
```

```
    double p_L,
    double p_R,
    double u_L,
    double u_R,
    double rho_L,
    double rho_R,
    bool entropy_fix)
{
    vector<double> F_prom = (F_L + F_R)*0.5;
    if (entropy_fix)
    {
        return F_prom - (suma_p_fix(p_L,
                                    p_R,
                                    u_L,
                                    u_R,
                                    rho_L,
                                    rho_R)*0.5);
    }
    else
    {
        return F_prom - (suma_p(p_L,
                                p_R,
                                u_L,
                                u_R,
                                rho_L,
                                rho_R)*0.5);
    }
}
```

Esta función retorna un objeto de la clase `vector`. La función `suma_p` corresponde al cálculo de la suma $\sum_{p=1}^{m} |\hat{\lambda}_p| \alpha_p \hat{\mathbf{r}}_p$. Por otro lado, la función `suma_p_fix` corresponde a la modificación de la anterior implementando la corrección de entropía, es decir, calcula la suma $\sum_{p=1}^{m} H(\hat{\lambda}_p) \alpha_p \hat{\mathbf{r}}_p$, definida en la ecuación (2.56). A continuación se muestra el código implementado de la suma sobre los vectores propios.

Listing 3.15. Definición de `suma_p`

```
/**
```

```
 * @brief Suma sobre p de los autovectores de la matriz A del sistema
 * por sus fuerzas y autovalores.
 *
 * @param p_L Presión a la izquierda
 * @param p_R Presión a la derecha
 * @param u_L Velocidad a la izquierda
 * @param u_R Velocidad a la derecha
 * @param rho_L Densidad a la izquierda
 * @param rho_R Densidad a la derecha
 * @return vector<double>
 */
vector<double> suma_p(double p_L, double p_R,
                      double u_L, double u_R,
                      double rho_L, double rho_R)
{
    // Cálculo de promedios de Roe
    // Velocidad promedio
    double u = u_prom(u_L, u_R, rho_L, rho_R);
    // Densidad promedio
    double rho = rho_prom(rho_L, rho_R);
    // Entalpía promedio
    double h = h_prom(p_L, p_R, u_L, u_R, rho_L, rho_R);
    // Velocidad del sonido promedio
    double c = c_prom(p_L, p_R, rho_L, rho_R, u_L, u_R);
    // Cálculo de diferencias laterales
    double dp = p_R - p_L;
    double du = u_R - u_L;
    double drho = rho_R - rho_L;
    // Coeficientes alfa
    double alfa_1 = 0.5*(dp-rho*c*du)/pow(c, 2);
    double alfa_2 = (pow(c, 2)*drho-dp)/pow(c, 2);
    double alfa_3 = 0.5*(dp+rho*c*du)/pow(c, 2);
    // Vector de coeficientes alfa
    vector<double> alfa = {alfa_1, alfa_2, alfa_3};
    // Vector de autovalores de las ondas centrales
    vector<double> lambda = {u-c, u, u+c};
```

```
        // Autovectores
        vector<double> r_1 = {1, u-c, h-u*c};
        vector<double> r_2 = {1, u, 0.5*pow(u,2)};
        vector<double> r_3 = {1, u+c, h+u*c};
        // vector de vectores propios
        vector<vector<double>> r_vec = {r_1, r_2, r_3};

        // Se declara el vector resultante de la suma,
        // de dimensión 3 y con ceros.
        vector<double> resultado(3, 0);

        // Se realiza la suma
        for (int i = 0; i < 3; i++)
        {
            // Se define la variable que almacena el valor absoluto de
            // cada autovalor
            double lambda_i = abs(lambda[i]);
            resultado += r_vec[i]*(alfa[i]*lambda_i);
        }
        return resultado;
}
```

En esta función se realizó el cálculo de los promedios de Roe, a través de las funciones `rho_prom`, `u_prom` y `p_prom`. La implementación del cálculo de dichos promedios (ecuaciones (3.42) - (3.45)) es trivial y directa, por lo que se omitirá el código fuente de cada función en este texto. Seguidamente, se calculan las diferencias laterales $\Delta\rho$, Δu, Δp; los coeficientes de las ondas, α_p; los autovalores de las ondas, λ_p; y los autovectores $\hat{\mathbf{r}}$.

Por otra parte, en la función `suma_p_fix` se definen las mismas variables que en `suma_p`; ya que, en problemas en donde no se detectan rarefacciones sónicas, ambas funciones son equivalentes. En `suma_p_fix` es necesario definir los autovalores de las ondas laterales: $\lambda_{p,L}$ y $\lambda_{p,R}$.

Listing 3.16. Implementación de $\lambda_{p,L}$ y $\lambda_{p,R}$

```
        // Construcción de autovalores de las ondas laterales
        double c_L = c_prom(p_L, p_L, rho_L, rho_L, u_L, u_L);
        double c_R = c_prom(p_R, p_R, rho_R, rho_R, u_R, u_R);
```

```
vector<double> lambda_L = {u_L - c_L, u, u_L + c_L};
vector<double> lambda_R = {u_R - c_R, u, u_R + c_R};
```

Puesto que los autovalores de una onda dependen de la velocidad del sonido en dicha onda (c), se definen las variables `c_L` y `c_R` que corresponden a la velocidad del sonido de la onda izquierda y derecha respectivamente. Dichas variables se construyen en el código utilizando las siguientes propiedades de los promedios de Roe:

$$\tilde{c}(\mathbf{U}_L, \mathbf{U}_L) = c_L \tag{3.62}$$

$$\tilde{c}(\mathbf{U}_R, \mathbf{U}_R) = c_R, \tag{3.63}$$

que se deducen inmediatamente de (2.50), y se implementan mediante la función `c_prom`. El siguiente proceso que se define para `suma_p_fix` se encarga de implementar la función $H(\lambda_p)$ definida en (2.57).

Listing 3.17. Construcción de la corrección de entropía de Harten y Hyman

```
for (int i = 0; i < 3; i++)
{
    // Se define la variable que almacena el valor absoluto del
    // autovalor iésimo.
    double lambda_i = abs(lambda[i]);
    // Se define delta de la corrección HH.
    double delta_i = max(lambda_i-lambda_L[i],
                         lambda_R[i]-lambda_i);
    delta_i = max(delta_i, 0.0);
    // Aplicación de la función por partes de la correción HH.
    if (lambda_i < delta_i)
    {
        lambda_i = delta_i;
    }
    else
    {
        lambda_i = abs(lambda[i]);
    }
    resultado += r_vec[i]*(alfa[i]*lambda_i);
}
return resultado;
```

3.4.6. Ciclo principal de iteración

Habiendo explicado las funciones auxiliares involucradas en el esquema de Roe, se puede entender el funcionamiento del ciclo principal de iteración. Este ciclo se encarga de aplicar la definición del método de volúmenes finitos,

$$U_i^{n+1} = U_i^n - \frac{k}{h}\left[F(U_i^n, U_{i+1}^n) - F(U_{i-1}^n, U_i^n)\right]. \tag{3.64}$$

Es evidente que para completar la aproximación numérica U se necesita iterar sobre dos ciclos: el espacial, iterando i y el temporal, iterando n. Para iterar sobre los instantes temporales en el código, se utilizó el índice `k`.

Listing 3.18. Ciclo principal de iteración

```
// Comienza a correr el tiempo
tiempo += dt;
// Ciclo principal
for (int k = 0; k < Niter; k++)
{
// Se calculan las componentes del vector U
calc_componentes_U(u1, u2, u3, rho, p, u, Nx);
// Ciclo para integración espacial
for (int i = 1; i < Nx-1; i++)
{
    // Definición del vector U_N que corresponde al vector U en
    // el siguiente instante de tiempo
    vector<double> U_N(3);
    // Definir valores de U
    U = {u1[i], u2[i], u3[i]};
    // Actualizar e integrar U
    U_N = U-((Flujo(flujo_euler(rho[i], p[i], u[i]),
                    flujo_euler(rho[i+1], p[i+1], u[i+1]),
                    p[i], p[i+1],
                    u[i], u[i+1],
                    rho[i], rho[i+1], correccion_de_entropia)-
              Flujo(flujo_euler(rho[i-1], p[i-1], u[i-1]),
                    flujo_euler(rho[i], p[i], u[i]),
                    p[i-1], p[i],
```

```
                u[i-1], u[i],
                rho[i-1], rho[i], correccion_de_entropia))*
                (dt/dx));
```

Se puede notar que el ciclo de las iteraciones temporales se detiene en cuánto se alcanza el número de iteraciones `Niter`, previamente definido. También destaca la expresión para `U_N`, que emula el resultado de la ecuación (3.64), tomando en cuenta las siguientes equivalencias:

$$U^n \leftrightarrow \texttt{U} \tag{3.65}$$

$$U^{n+1} \leftrightarrow \texttt{U\_N}. \tag{3.66}$$

Consecutivamente, en el ciclo de integración espacial, se despejan y actualizan las variables físicas independientes de los valores de `U_N`, ya que dichas variables serán exportadas como los datos definitivos de solución del sistema.

```
    // Despejar variables físicas de U
    rho_nueva[i] = U_N[0];
    u_nueva[i] = U_N[1]/rho_nueva[i];
    p_nueva[i] = (U_N[2]-0.5*rho_nueva[i]*pow(u_nueva[i], 2))*
                 (Gamma-1);
}

// Actualizar variables físicas
for (int i = 1; i < Nx-1; i++)
{
    rho[i] = rho_nueva[i];
    u[i] = u_nueva[i];
    p[i] = p_nueva[i];
}
```

Por último, se vuelven a aplicar las condiciones de frontera en el ciclo principal y se imprimen los datos en archivos de formato `.dat` cuando la iteración corresponda a una salida de datos.

```
// Condiciones de frontera transmisivas
rho[0] = rho[1];
rho[Nx-1] = rho[Nx-2];
u[0] = u[1];
u[Nx-1] = u[Nx-2];
```

```
p[0] = p[1];
p[Nx-1] = p[Nx-2];

// Se evalúa si la iteración corresponde a un instante de
// impresión de datos
if (k % out_cada == 0)
{
    salida(file_densidad, rho, x, tiempo, Nx);
    salida(file_presion, p, x, tiempo, Nx);
    salida(file_velocidad, u, x, tiempo, Nx);
    cout << round(100*tiempo/t_total*100)/100 << endl;
}
// Actualizar el tiempo
tiempo += dt;
}
```

La función `salida` se encarga de imprimir los datos haciendo uso de objetos de tipo `ofstream`.

Listing 3.19. Definición de la función `salida`, que envía los datos

```
/**
 * @brief Función que envía datos a los archivos, con instantes
 * temporales separados por doble enter
 *
 * @param of Archivo de datos
 * @param u Arreglo que almacena los valores de las funciones
 * @param x Arreglo de dimensión espacial
 * @param tiempo Instante temporal en cuestión
 * @param N Tamaño de los arreglos u y x
 */
void salida(ofstream &of, double *u, double *x, double tiempo,
            int N){
    for (int i = 0; i < N; i++)
    {
        of << tiempo << "\t" << x[i] << "\t" << u[i] << endl;
    }
    of << endl << endl;}
```

3.5. Resultados

3.5.1. Simulación con el primer conjunto de condiciones iniciales

Se realizó la simulación de las ecuaciones de Euler con las condiciones iniciales descritas en la sección 3.3.2.1. Como se había mencionado, se esperaba que la función de densidad ρ se comportara como la solución a una ecuación de advección; esto a partir de que la condición inicial para la velocidad era constante y la presión también.

3.5.1.1. Gráficas

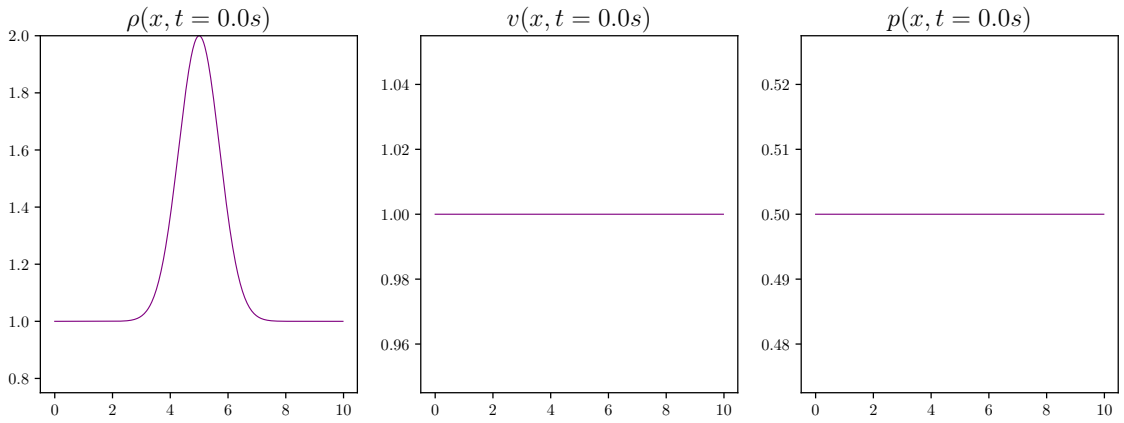


Figura 3.1. Gráficas para $t = 0.0$s

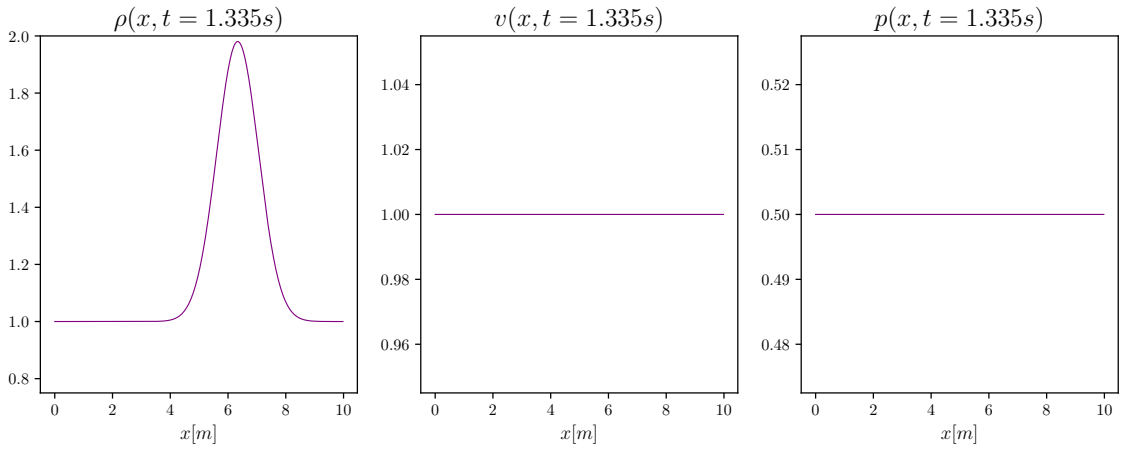


Figura 3.2. Gráficas para $t = 1.335$s

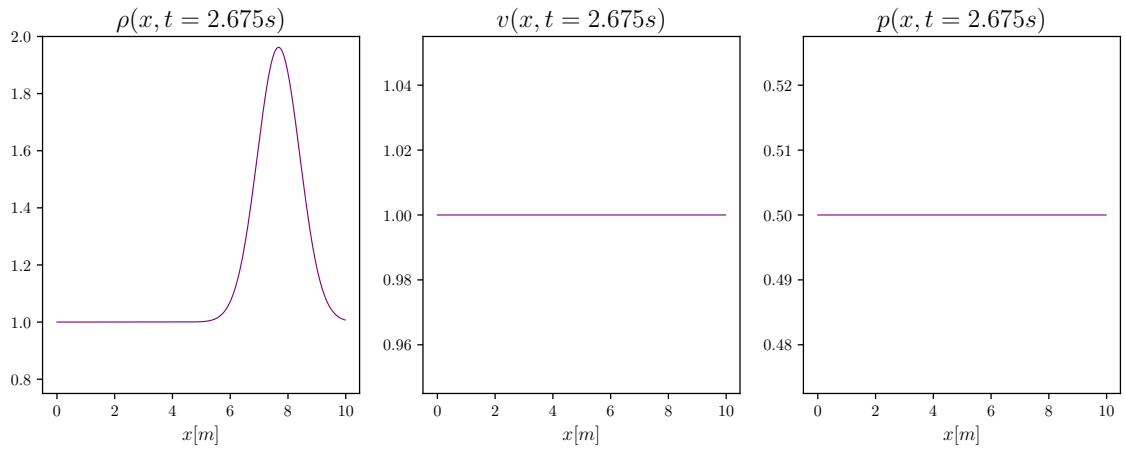


Figura 3.3. Gráficas para $t = 2.675$s

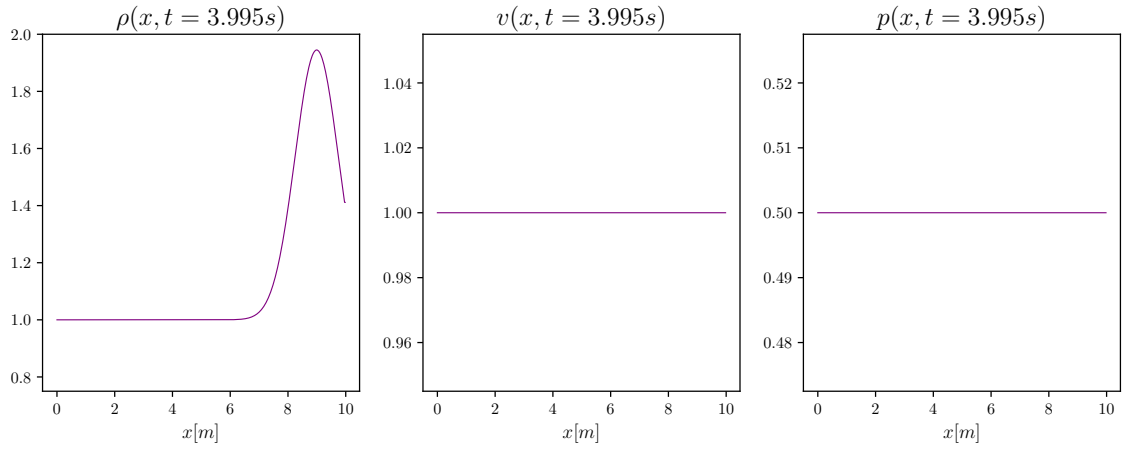


Figura 3.4. Gráficas para $t = 3.995$s

3.5.1.2. Discusión

Es apreciable el comportamiento de advección que sufre el perfil gaussiano de la densidad del gas. En el instante $t = 0.0$s el centro de la curva gaussiana se encuentra en $x = 5$m, mientras que en el instante $t = 3.995$s éste se encuentra aproximadamente en $x = 9$m. Esta última observación coincide con lo esperado, dado que la velocidad del gas es $1.0\text{m}\,\text{s}^{-1}$ y es constante a lo largo de toda la simulación. Por otro lado, destaca que el máximo de la función de densidad no se mantiene constante, sino que se reduce. Esto se debe al error numérico de integración del esquema utilizado.

Esta simulación se realizó sin la corrección de entropía del esquema de Roe ya que no presentó ninguna diferencia cuando se aplicaba la corrección.

3.5.2. Simulación con el segundo conjunto de condiciones iniciales

Las gráficas de las figuras 3.5 - 3.8 corresponden a la simulación realizada con las condiciones iniciales descritas en la sección 3.3.2.2. Este conjunto de condiciones se caracteriza por la diferencia entre las magnitudes de las mitades del dominio. La densidad del lado izquierdo triplica la del lado derecho, al igual que sucede con la presión, mientras que la velocidad inicial del fluido es nula. Entonces este sistema puede ser interpretado como la súbita fusión de dos gases con densidad y presión distinta, o en otras palabras, como un **choque**.

3.5.2.1. Gráficas

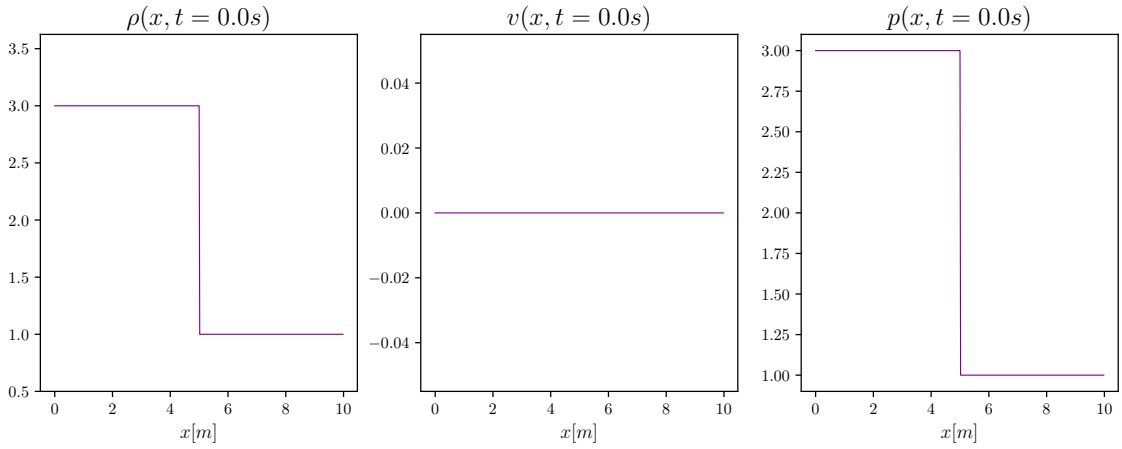


Figura 3.5. Gráficas para $t = 0.0$s

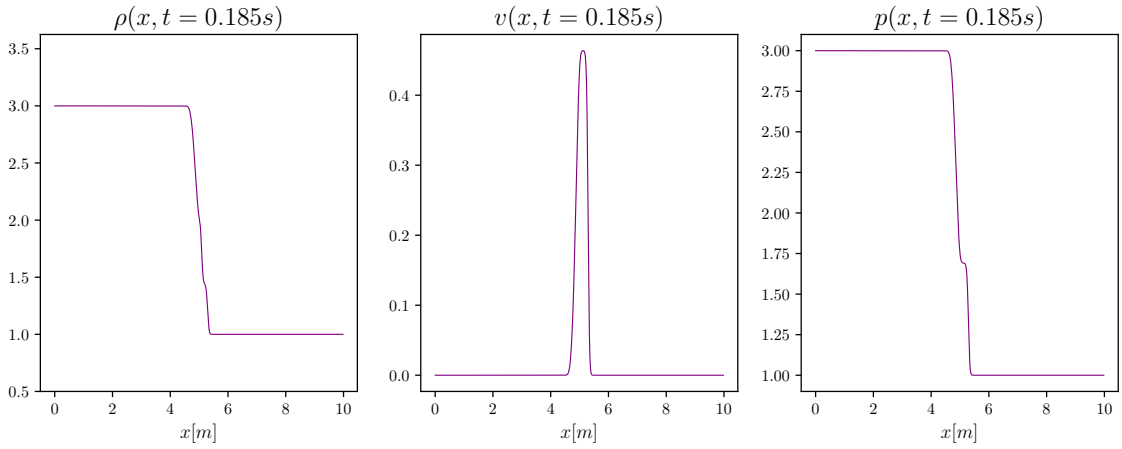


Figura 3.6. Gráficas para $t = 0.185$s

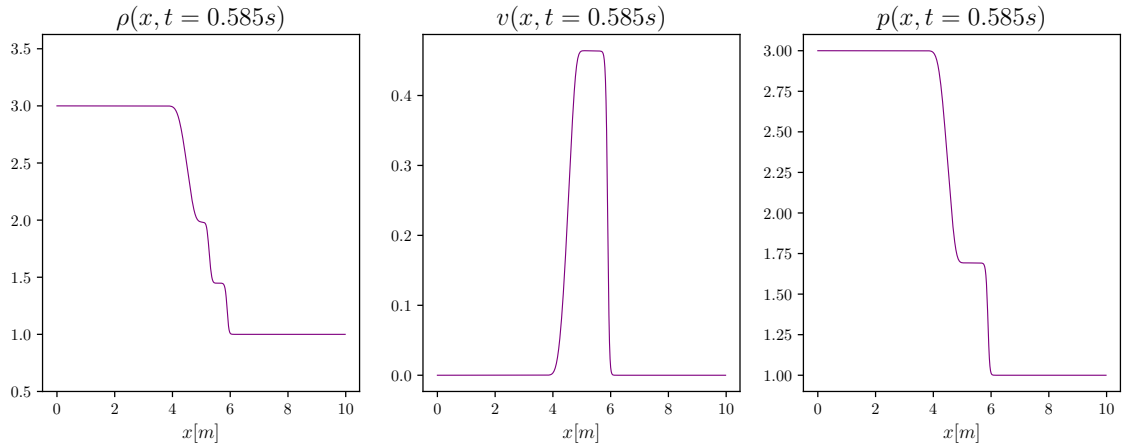


Figura 3.7. Gráficas para $t = 0.585$s

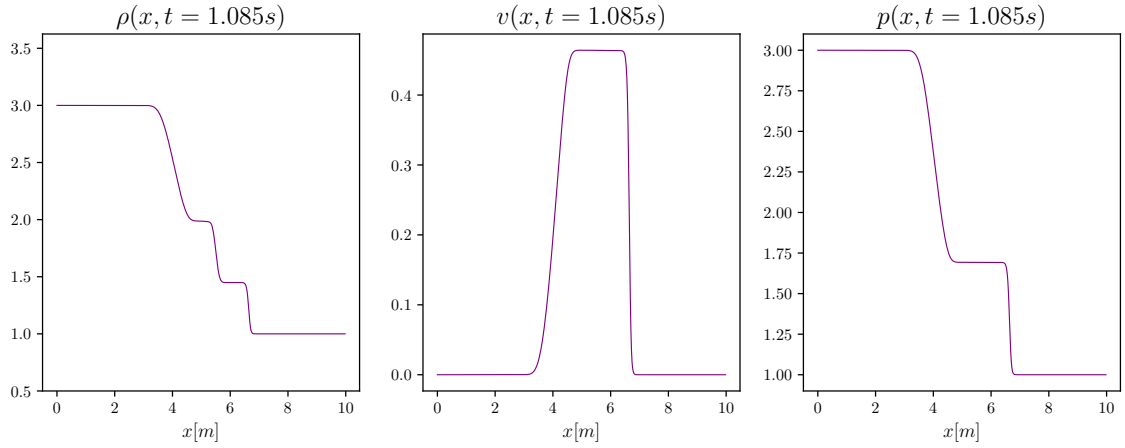


Figura 3.8. Gráficas para $t = 1.085$s

3.5.2.2. Discusión

En esta simulación se destaca principalmente la manifestación de ondas de choque y de rarefacción en todas las variables del gas. La densidad se divide en tres ondas; una de choque, que avanza hacia la derecha, y otra de rarefacción, que avanza a la izquierda. La onda intermedia entre las dos últimas se conoce como **discontinuidad de contacto** [11] y se caracteriza por ser una discontinuidad en la densidad únicamente, ya que la presión y velocidad permanecen constantes. La discontinuidad de contacto también está estrechamente relacionada con la diferencia de temperatura entre las partes del gas que ésta divide [6].

Por otro lado, es notable la eficiencia del esquema de Roe al capturar ondas de choque (discontinuidades) como soluciones.

Esta simulación se realizó sin la corrección de entropía del esquema de Roe ya

que no presentó ninguna diferencia cuando se aplicaba la corrección.

3.5.3. Simulación con el tercer conjunto de condiciones iniciales

Se experimentó con el conjunto de condiciones iniciales descrito en la sección 3.3.2.3. Este conjunto es similar al discutido anteriormente, con la diferencia que la velocidad inicial de este conjunto no es nula, sino que tiene un valor constante en todo el dominio. Este problema surge a partir de una forma de evaluar la aparición de ondas de rarefacción sónica, de tal manera que pone a prueba la función de corrección de entropía utilizada.

3.5.3.1. Gráficas de la simulación sin corrección de entropía

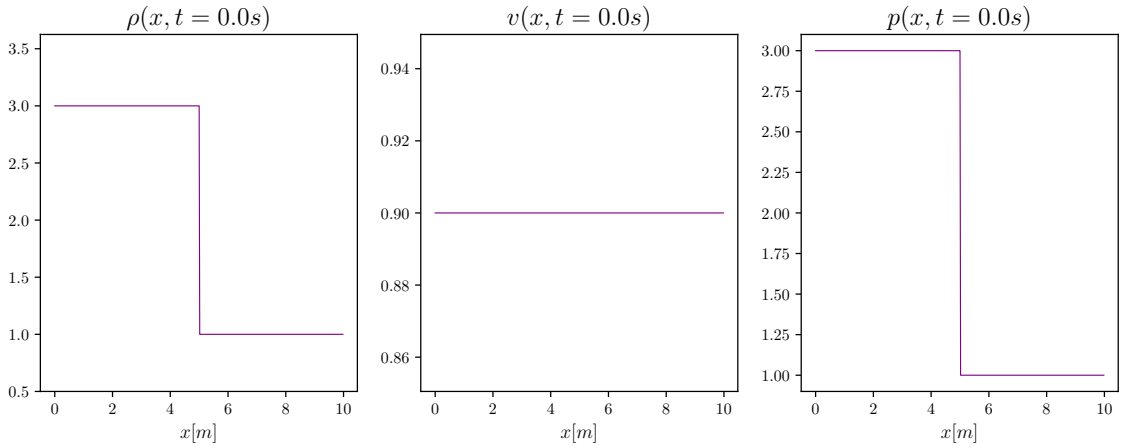


Figura 3.9. Gráficas para $t = 0.0$s

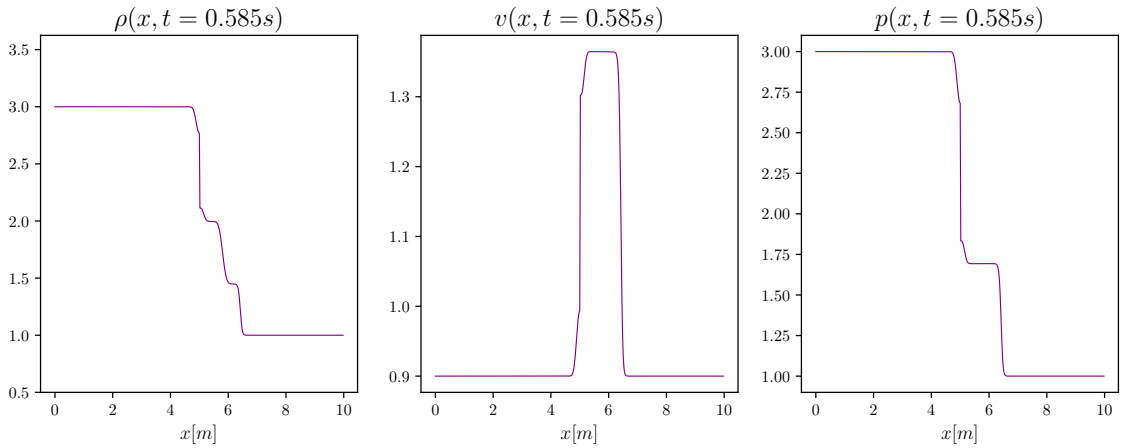


Figura 3.10. Gráficas para $t = 0.585$s

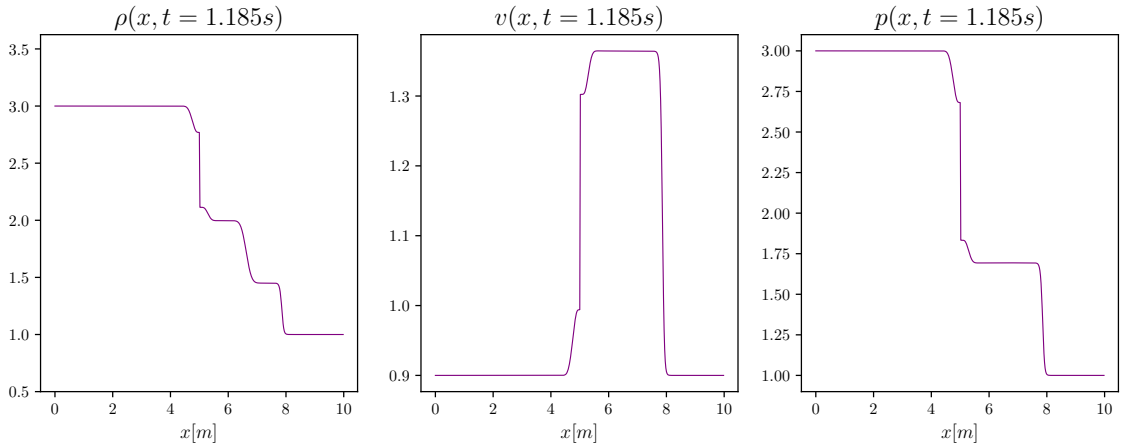


Figura 3.11. Gráficas para $t = 1.185$s

3.5.3.2. Gráficas de la simulación con corrección de entropía

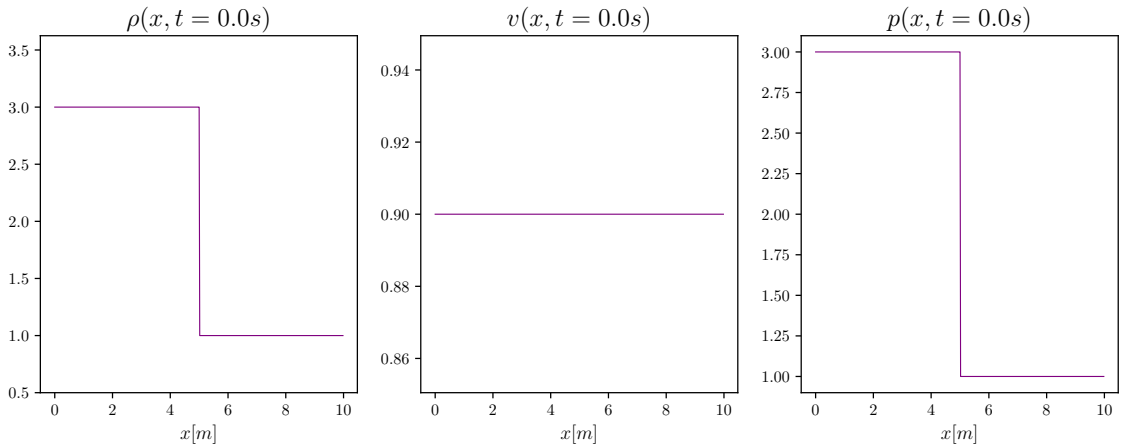


Figura 3.12. Gráficas para $t = 0.0$s

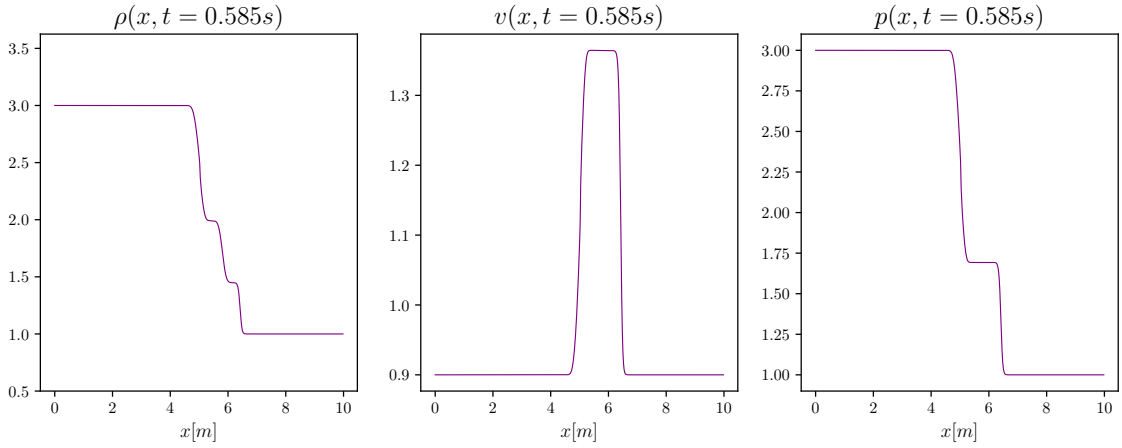


Figura 3.13. Gráficas para $t = 0.585$s

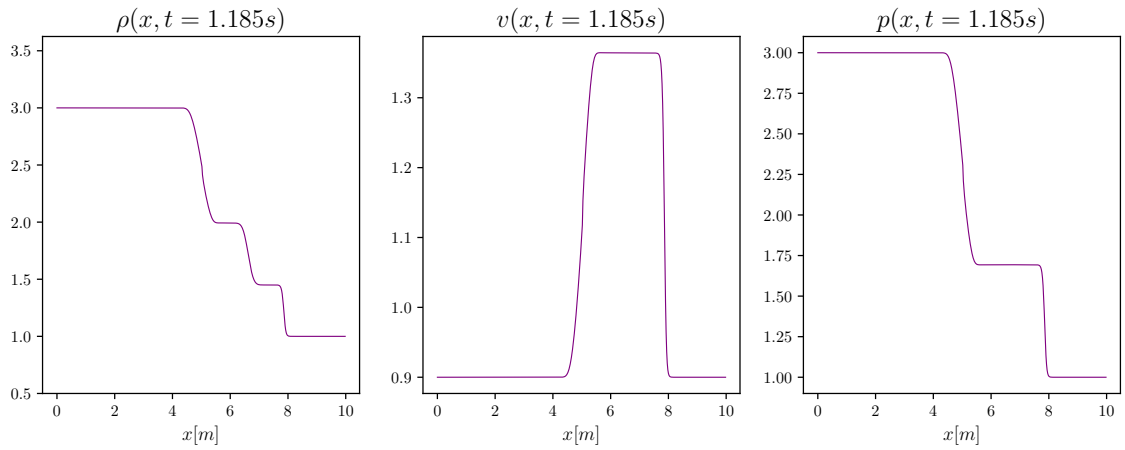


Figura 3.14. Gráficas para $t = 1.185$s

3.5.3.3. Discusión

Se puede notar la similitud entre los resultados de esta simulación y la correspondiente al segundo conjunto. Sin embargo, en este caso el programa sin la condición de entropía no logra capturar las ondas de rarefacción, como puede notarse en la figura 3.14; notando que especialmente la onda de rarefacción que avanza a la izquierda (identificada en los resultados de la sección anterior) aparece dividida por una discontinuidad. En cambio, los resultados producidos al utilizar la corrección de entropía sí son consistentes con las ondas de rarefacción esperadas; se puede notar la similitud entre estos resultados y los de la sección 3.5.2. La onda de rarefacción izquierda no presenta discontinuidades en estos resultados.

4. COMPARACIÓN CON PYCLAW

Mientras se investigaba sobre las ecuaciones de conservación, previo a la realización de este texto, los textos de Randy LeVeque destacaron por su inmenso aporte a la teoría de la resolución numérica de estos sistemas. Al profundizar en los aportes de LeVeque, fue imposible no toparse con el paquete desarrollado para el lenguaje de programación Python, **PyClaw**, donde LeVeque está listado como el principal diseñador del software y de los algoritmos implementados [4]. El paquete PyClaw fue desarrollado para la resolución numérica de ecuaciones de conservación lineales y no lineales, utilizando métodos de alta resolución. Estos métodos se basan en la construcción de la solución del problema de Riemann a través de diversos esquemas, como el de Roe.

A continuación se describen los algoritmos implementados en el paquete **Clawpack**. Esta descripción extrae los conceptos presentados en la documentación oficial del mismo [4].

4.1. Fundamentos de Clawpack

PyClaw forma parte del paquete de solucionadores numéricos **Clawpack** [1]. Éste es capaz de resolver sistemas de ecuaciones diferenciales de la forma estándar de conservación

$$q_t + f(q)_x = 0. \tag{4.1}$$

La solución numérica de estos sistemas se basa en solucionadores de Riemann. Sea $S(Q_{i-1}, Q_i)$ un solucionador de Riemann. Para poder implementar éste último es necesario que retorne un conjunto de M_w ondas denominadas $\mathcal{W}^p_{i-1/2}$, con velocidades $s^p_{i-1/2}$, que correspondan a la solución del problema de Riemann entre las celdas i e

[1] El nombre abrevia la frase en inglés **C**onservation **Law Pack**age.

$i-1$, siempre que satisfaga la siguiente condición

$$\sum_{p=1}^{M_w} \mathcal{W}_{i-1/2}^p = Q_i - Q_{i-1} \equiv \Delta Q_{i-1/2}. \tag{4.2}$$

La construcción, basada en ondas y sus velocidades, de la solución del problema de Riemann se asemeja a la forma en la que se construye la solución para el caso lineal (ver sección 1.9.2). Para calcular la diferencia de flujos que toma lugar en el método de volúmenes finitos, el algoritmo de PyClaw divide la diferencia de flujos en dos fluctuaciones,

$$\mathcal{A}^+ \Delta Q_{i-1/2} = \sum_p (s_{i-1/2}^p)^+ \mathcal{W}_{i-1/2}^p \tag{4.3}$$

$$\mathcal{A}^- \Delta Q_{i-1/2} = \sum_p (s_{i-1/2}^p)^- \mathcal{W}_{i-1/2}^p, \tag{4.4}$$

con $s^- = \text{mín}(s, 0)$ y $s^+ = \text{máx}(s, 0)$. De tal manera que estas fluctuaciones satisfacen lo siguiente

$$\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i-1/2} = f(Q_i) - f(Q_{i-1}). \tag{4.5}$$

Definiendo estos términos, se obtiene una expresión para el esquema general de integración. En la documentación oficial de PyClaw éste se denomina **Método de Godunov**, pero no debe confundirse con el esquema de Godunov. Entonces,

$$Q_i^{n+1} = Q_i^n - \frac{k}{h} \left[\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i-1/2} \right] \tag{4.6}$$

es la expresión utilizada en los algoritmos de integración de PyClaw. Cabe resaltar su similitud con (2.11), y además implica que la diferencia de flujos numéricos $F(U_{i-1}^n, U_i^n) - F(U_i^n, U_{i+1}^n)$ equivale a la suma de las fluctuaciones $\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i-1/2}$ definidas en los algoritmos de PyClaw.

En el software PyClaw se implementa la forma (4.6) para resolver el sistema de conservación ya que al construir la solución a través de las ondas $\mathcal{W}_{i-1/2}^p$ y sus velocidades $s_{i-1/2}^p$ es posible implementar métodos de alta resolución. Los métodos de alta resolución introducen valores limitantes para las ondas, de tal manera que se evitan oscilaciones no-naturales cerca de las discontinuidades o también gradientes muy pronunciados o inexactos. Los métodos de alta resolución son métodos numéricos con *variación total disminuida* o métodos **TVD** por sus siglas en inglés. La

forma general de éstos es

$$Q_i^{n+1} = Q_i^n - \frac{k}{h}\left[\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i-1/2}\right] - \frac{k}{h}\left(\tilde{F}_{i+1/2} - \tilde{F}_{i-1/2}\right), \tag{4.7}$$

donde la corrección de flujos está dada por:

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{M_w}\left|s_{i-1/2}^p\right|\left(1 - \frac{\Delta t}{\Delta x}\left|s_{i-1/2}^p\right|\right)\tilde{\mathcal{W}}_{i-1/2}^p. \tag{4.8}$$

Aquí, $\tilde{\mathcal{W}}_{i-1/2}^p$ representa la onda $\mathcal{W}_{i-1/2}^p$ luego de que se le ha aplicado la limitante, de acuerdo al método de alta resolución.

4.2. Código de PyClaw utilizado

PyClaw ofrece la resolución de las ecuaciones de Euler a través de los algoritmos presentados implementados en Python o incluso mediante Fortran [2]. Para la resolución de las ecuaciones de Euler se decidió utilizar la versión de Python. En PyClaw están disponibles otros esquemas de flujo numérico distintos al esquema de Roe. Sin embargo, se utilizó de nuevo este esquema para reducir las diferencias entre los programas diseñados (C++ y Python) y basar la comparación en los algoritmos de resolución, principalmente.

Listing 4.1. Paquetes e instancias

```
# @title Euler 1D set 1
nombre = "set1"
from clawpack import pyclaw
from clawpack import riemann
from funciones import funcion_paso
# Importar el solucionador
solver = pyclaw.ClawSolver1D(riemann.euler_1D_py.euler_roe_1D)
# Establecer el kernel del algoritmo de integración
solver.kernel_language = "Python"
```

Los paquetes imprescindibles en una simulación de Clawpack son `pyclaw` y `riemann`. El solucionador, que implementa la expresión (4.6), es un objeto de `pyclaw` y debe elegirse dependiendo de la dimensión espacial del problema; por ello se utilizó

[2] Fortran es un lenguaje de programación para cálculos científicos, creado en 1957. Destaca por su eficiencia numérica y sigue siendo útil en aplicaciones científicas.

`ClawSolver1D`. El paquete `riemann` contiene, como objetos, los sistemas que Clawpack puede solucionar numéricamente. Dichos objetos tienen como métodos los solucionadores de Riemann disponibles.

Listing 4.2. Parámetros básicos de la solución

```
# Condiciones de frontera.
solver.bc_upper[0] = pyclaw.BC.extrap
solver.bc_lower[0] = pyclaw.BC.extrap
# Dominio
domain = pyclaw.Domain([0], [10], [500])
# Objeto de solución
solution = pyclaw.Solution(solver.num_eqn, domain)
# Objeto de estado de la simulación
state = solution.state
# Centros de celdas de dominio espacial
xc = state.grid.p_centers[0]
```

Las condiciones de frontera que funcionan como condiciones transmisivas se denominan `extrap` y se pasan como valores de las listas `bc_upper` y `bc_lower`, que son atributos del solucionador. El dominio se divide en 500 celdas, al igual que en la simulación de C++, y va de 0m a 10m horizontalmente. Es posible acceder al punto medio sobre el eje x de cada celda con la lista `state.grid.p_centers[0]`.

Listing 4.3. Condiciones iniciales y parámetros auxiliares

```
from numpy import exp
# Gamma
cda_gamma = 1.4
# Definicion de funciones auxiliares de
# variables independientes y conservadas
densidad = 1 + exp(-((xc-0.01)-5)**2)
velocidad = 1.0
presion = 0.5
energia = presion/(cda_gamma - 1) + 0.5*densidad*velocidad**2
# Asignación de variables conservadas
state.q[0,:] = densidad
state.q[1,:] = densidad*velocidad
state.q[2,:] = energia
# Parámetros del problema
```

```
state.problem_data["gamma"] = cda_gamma
state.problem_data["gamma1"] = cda_gamma - 1
state.problem_data["efix"] = False
```

La simulación de PyClaw implementa la resolución de las ecuaciones de Euler en términos de las variables conservadas, i.e., densidad, ρ; momentum, ρv y energía, ρE. Por esta razón, se definieron las variables físicas independientes y en términos de éstas se asignaron los valores de las variables conservadas. Además, PyClaw exige la definición del coeficiente de dilatación adiabática γ como un parámetro del problema como valor del atributo del estado de la solución, `problem_data`. Curiosamente, el valor para $\gamma - 1$ se asigna por aparte en este mismo atributo.

Por último, el parámetro `efix` de `problem_data` es un booleano que decide si se utilizará la corrección de entropía. Sin embargo, a la fecha no existe ningún algoritmo de corrección de entropía implementado en PyClaw.

Listing 4.4. Ejecución de la simulación

```
# Controlador
controller = pyclaw.Controller()
controller.solution = solution
controller.solver = solver
controller.num_output_times = 5
controller.tfinal = 3
controller.output_format = "ascii"
controller.outdir = nombre
# Correr la simulación
status = controller.run()
```

Para ejecutar la simulación es necesario utilizar un objeto de la clase `Controller` de `pyclaw`. En los atributos del objeto se definen otros parámetros relacionados con el tiempo total de simulación y las veces que se imprimen los valores. Cabe destacar que existen atributos del controlador que sirven para indicar el tamaño de paso temporal máximo, ya que hay algoritmos con tamaño de paso temporal adaptativo implementados en la librería.

4.3. Comparación de resultados

Para visualizar la diferencia entre las simulaciones obtenidas con el programa escrito en C++ y las generadas a través de PyClaw se mostrará la superposición de

las gráficas de ambas soluciones sobre algunos instantes temporales.

Por otro lado, para poder cuantificar la diferencia entre las simulaciones se optó por hacer uso del **error cuadrático medio** entre ambas soluciones. Considerando a $\rho^n_{i,\texttt{c++}}$, $u^n_{i,\texttt{c++}}$ y $p^n_{i,\texttt{c++}}$ como las funciones numéricas producidas por la simulación en C++ y a $\rho^n_{i,\texttt{py}}$, $u^n_{i,\texttt{py}}$ y $p^n_{i,\texttt{py}}$ como las generadas por PyClaw, se define el error cuadrático medio [3] en el enésimo instante de tiempo, RMSE_n, como:

$$\text{RMSE}_n(\rho) = \sqrt{\frac{\sum_i (\rho^n_{i,\texttt{c++}} - \rho^n_{i,\texttt{py}})^2}{N_x}}, \tag{4.9}$$

donde N_x es el número de celdas. La misma fórmula se aplica para las demás variables independientes. Para cuantificar el error total entre simulaciones, se calcula la media cuadrática de los errores cuadráticos previamente definidos. Entonces, se define la cantidad RMSE como

$$\text{RMSE} = \sqrt{\frac{\sum_n^{N_t} (\text{RMSE}_n)^2}{N_t}}, \tag{4.10}$$

donde N_t es el número de instantes temporales. Este cálculo también se realiza por cada variable.

4.3.1. Comparación del primer conjunto de condiciones iniciales

4.3.1.1. Gráficas

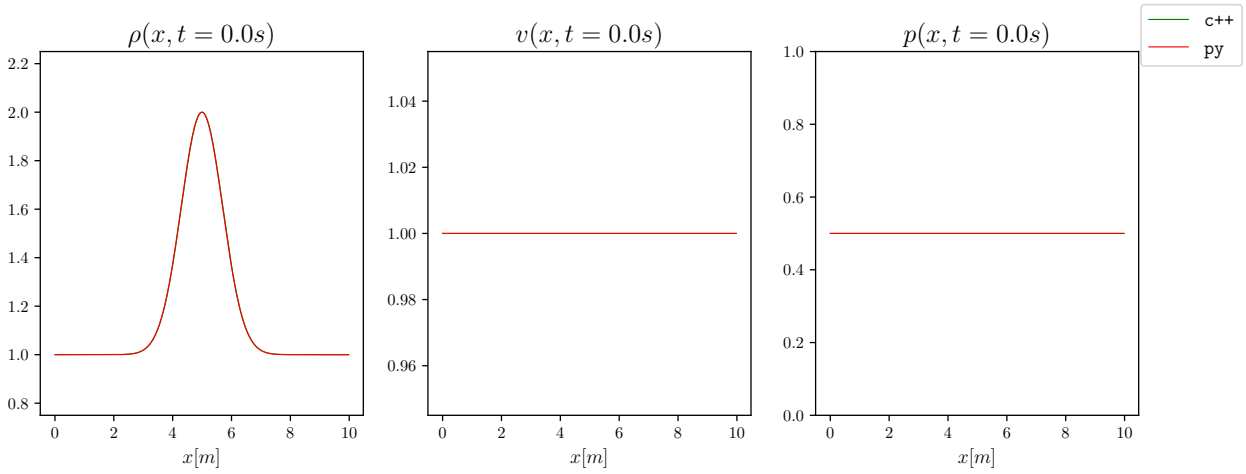


Figura 4.1. Gráficas para $t = 0.0$s

[3] RMSE: Root-Mean-Square Error, en inglés.

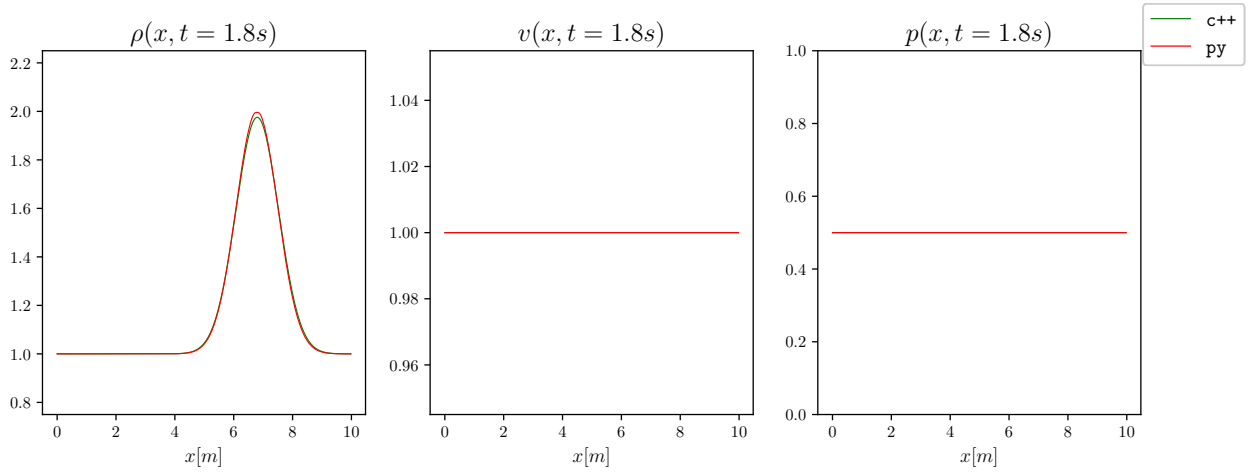


Figura 4.2. Gráficas para $t = 1.8$s

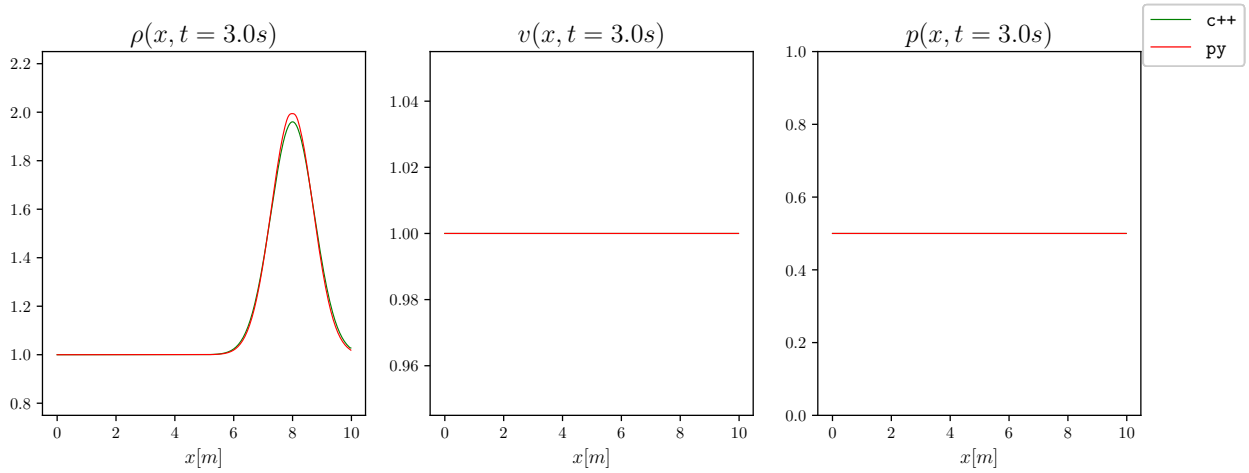


Figura 4.3. Gráficas para $t = 3.0$s

4.3.1.2. Discusión del primer conjunto de condiciones iniciales

Se puede notar cómo la simulación de PyClaw sí mantiene la forma original de la campana gaussiana correspondiente a la densidad. En la figura 4.3 es apreciable que el máximo de las dos funciones difiere. Las funciones de velocidad y presión no presentan diferencias entre las dos simulaciones.

Resultados	ρ	u	p
RMSE	8.2e-03	0.0e+00	0.0e+00
Error máximo	4.1e-02	0.0e+00	0.0e+00

Tabla 4.1. Comparativa de error entre simulaciones, primer conjunto de condiciones iniciales.

4.3.2. Comparación del segundo conjunto de condiciones iniciales

4.3.2.1. Gráficas

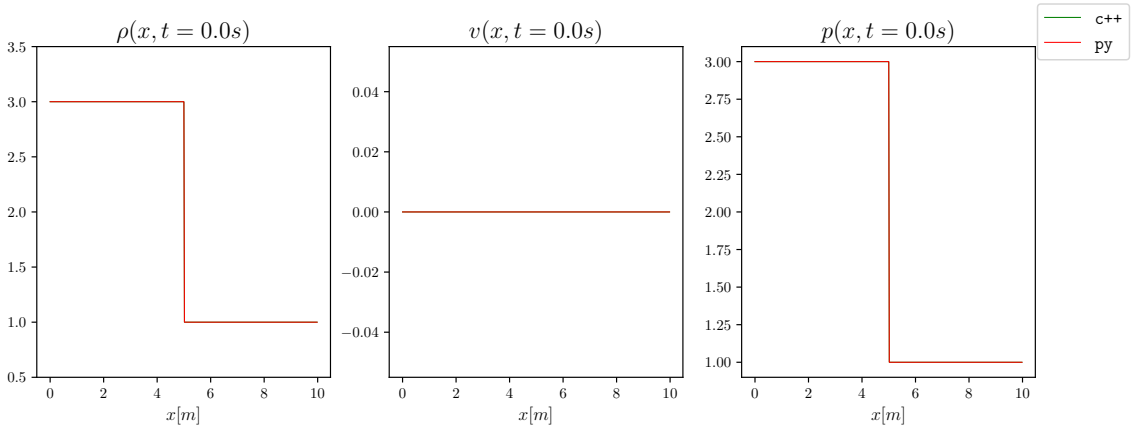


Figura 4.4. Gráficas para $t = 0.0$s

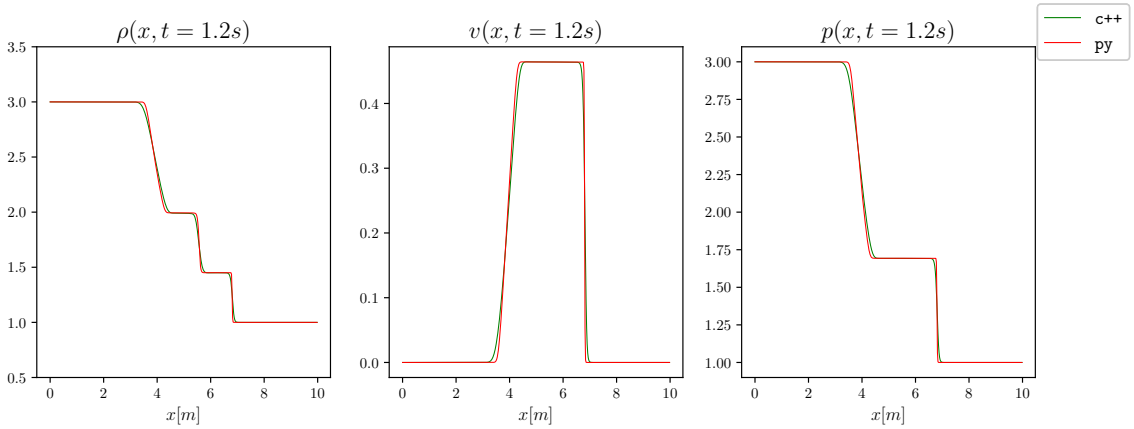


Figura 4.5. Gráficas para $t = 1.2$s

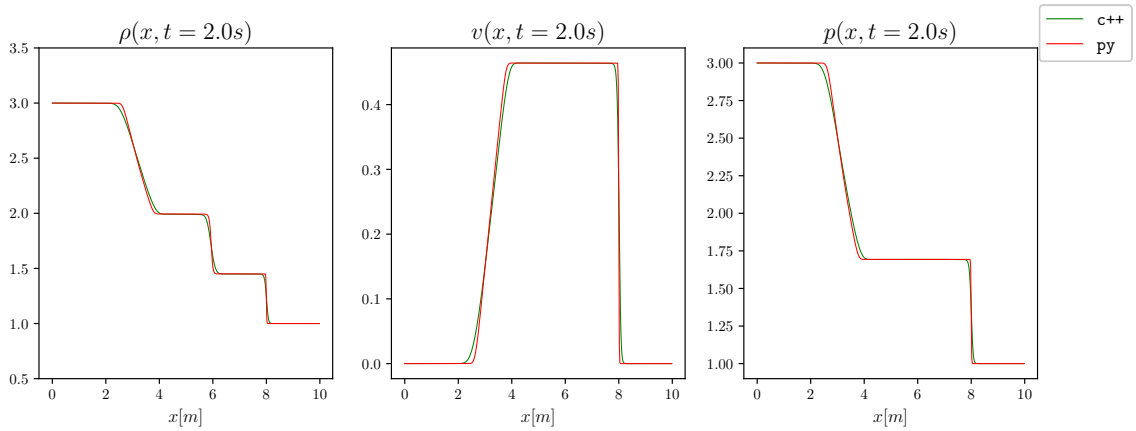


Figura 4.6. Gráficas para $t = 3.0$s

Resultados	ρ	u	p
RMSE	2.2e-02	1.4e-02	2.6e-02
Error máximo	1.6e-01	1.7e-01	2.3e-01

Tabla 4.2. Comparativa de error entre simulaciones, segundo conjunto de condiciones iniciales.

4.3.2.2. Discusión del segundo conjunto de condiciones iniciales

A través de la simulación con este conjunto de condiciones iniciales se destaca que las funciones producidas por PyClaw devuelven ondas de rarefacción distintas a las producidas por la simulación de C++, notando que poseen pendientes más pronunciadas cerca de las discontinuidades. Este hecho es más evidente al observar la evolución de la presión, ya que es la variable con el valor de RMSE más alto. Se puede inferir que esta diferencia se debe a los algoritmos de alta resolución implementados en PyClaw.

A través de esta simulación resalta que tanto el algoritmo implementado en C++ como los métodos de PyClaw capturan adecuadamente la velocidad de propagación de las ondas de choque y de rarefacción. Este hecho es importante dado que garantiza que los métodos implementados son conservativos.

4.3.3. Comparación del tercer conjunto de condiciones iniciales

4.3.3.1. Gráficas

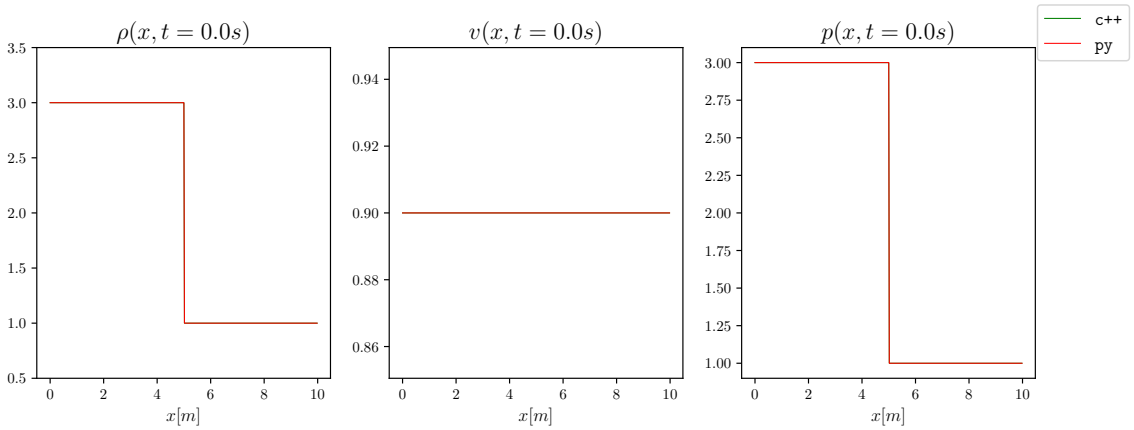


Figura 4.7. Gráficas para $t = 0.0$s

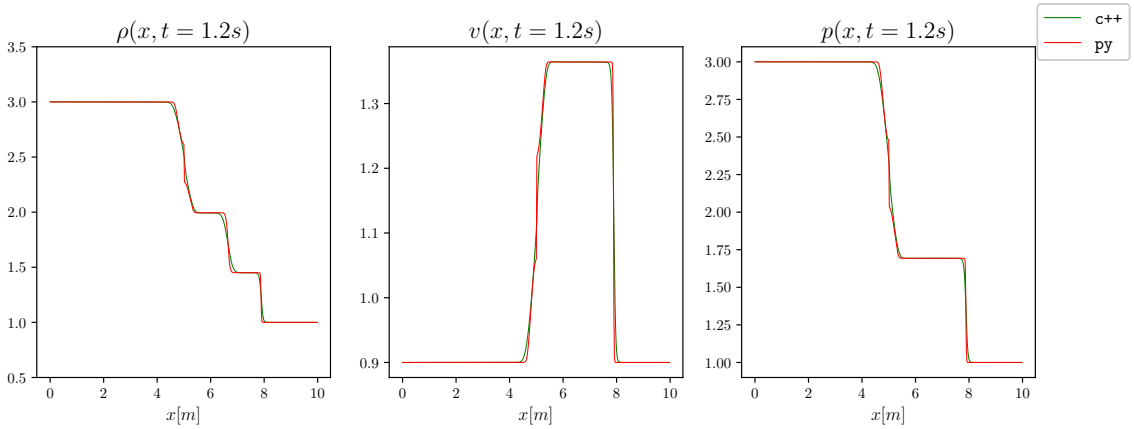


Figura 4.8. Gráficas para $t = 1.2$s

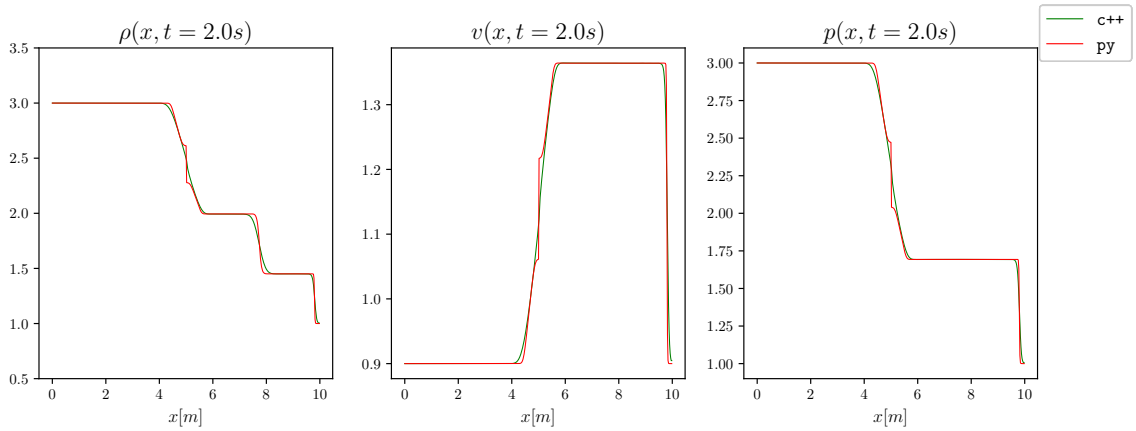


Figura 4.9. Gráficas para $t = 2.0$s

Resultados	ρ	u	p
RMSE	2.5e-02	1.4e-02	2.6e-02
Error máximo	2.5e-01	1.6e-01	3.2e-01

Tabla 4.3. Comparativa de error entre simulaciones, tercer conjunto de condiciones iniciales.

4.3.3.2. Discusión del tercer conjunto de condiciones iniciales

En este conjunto de soluciones se expone una diferencia que ya había sido observada en los resultados de la simulación con C++ con la corrección de entropía y sin ésta (sección 3.5.3). El algoritmo de PyClaw no considera la corrección de entropía en el esquema de Roe, a diferencia del programa escrito en C++, que sí tiene esta funcionalidad. Por estas razones, este conjunto de soluciones son las que presentan los mayores RMSE. Aunque los errores máximos encontrados no parecen alarmantes, si se dejara correr la simulación por más tiempo, se esperaría que el error aumentara considerablemente.

5. SIMULACIONES CON DISTINTOS COEFICIENTES DE DILATACIÓN ADIABÁTICA

En este último capítulo se comparan los resultados obtenidos en simulaciones del mismo problema de condición inicial pero con distinto coeficiente de dilatación adiabática γ, con el fin de obtener una intuición física, a través de la simulación, de cómo varía el comportamiento de un gas cuando el número de grados de libertad interno del mismo cambia.

5.1. Consideraciones preliminares

Dado que el coeficiente de dilatación adiabática está relacionado explícitamente con el número de grados de libertad interno de un gas (α),

$$\gamma = \frac{\alpha + 2}{\alpha}, \tag{5.1}$$

es conveniente definir los valores para γ tal que dependan de un valor entero para α.

En las anteriores simulaciones, se calculó γ para un gas diatómico, como el aire. La justificación del número de grados de libertad de estos gases se dio en la sección 3.1.2. En este caso:

$$\alpha = 5 \implies \gamma = 7/5 = 1.4. \tag{5.2}$$

Entonces, se considerarán experimentos con gases de 16 y 32 grados de libertad internos, que corresponden a $\gamma = 1.125$ y $\gamma = 1.0625$ respectivamente.

5.2. Test de Sod

El problema de condiciones iniciales que se estudiará con simulaciones con distintos valores para γ es el test de Sod (sección 3.3.2.2). En las figuras 5.1 - 5.2 se

encuentran seis instantes temporales de las simulaciones para el test de Sod.

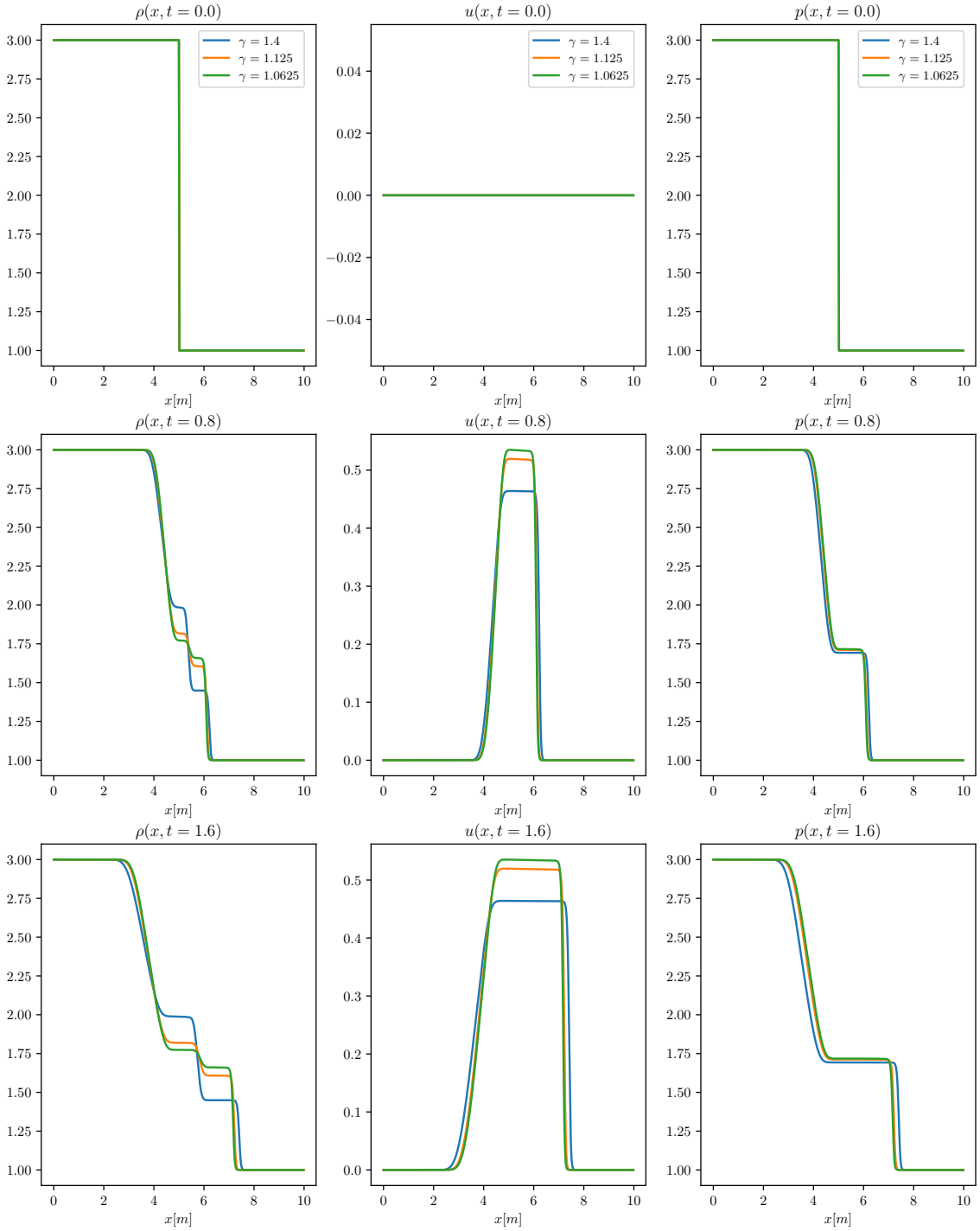


Figura 5.1. Primeros tres instantes de las simulaciones con distintos γ.

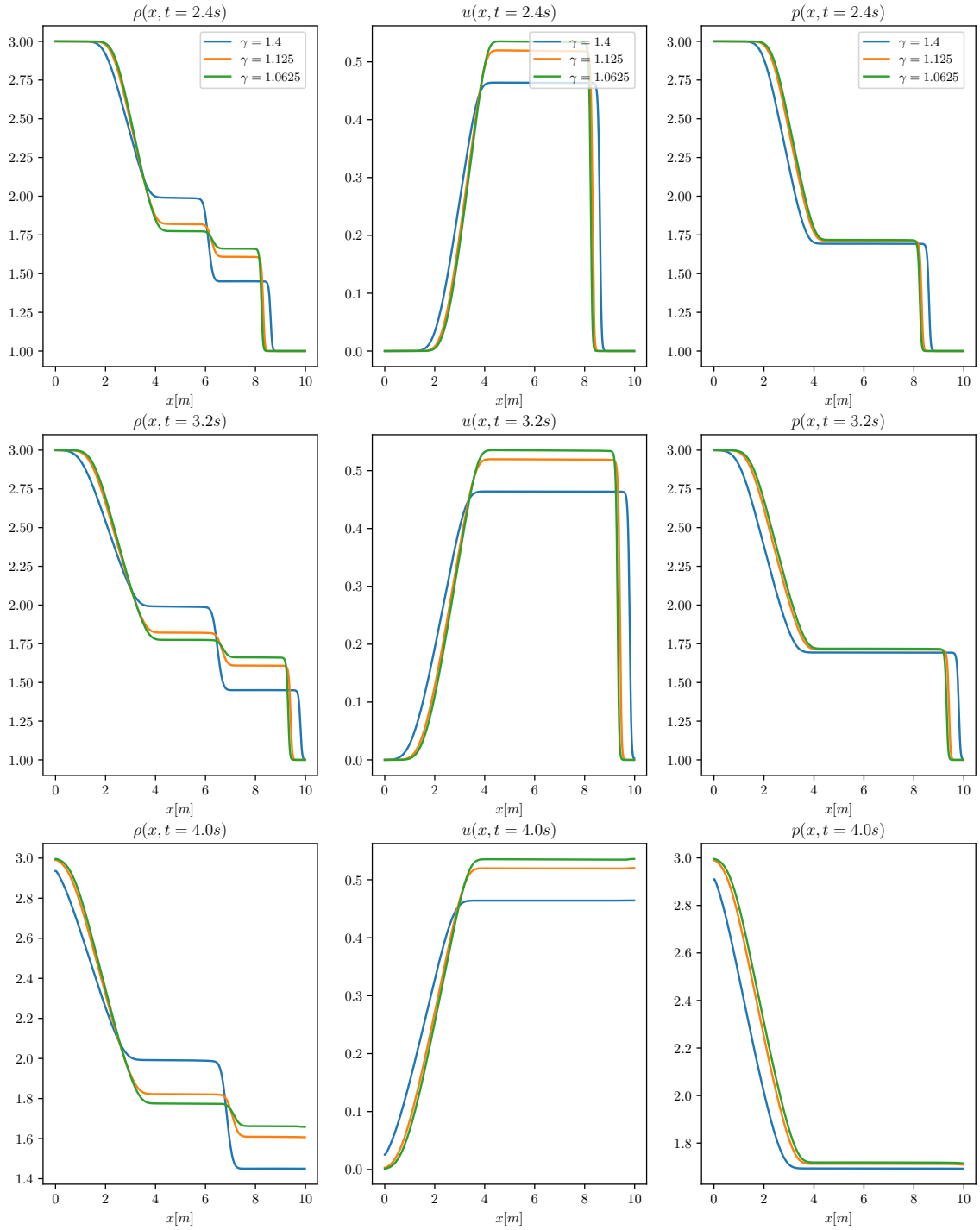


Figura 5.2. Últimos tres instantes de las simulaciones con distintos γ.

5.2.1. Observaciones

- En la simulación para la densidad, destaca cómo la diferencia en la discontinuidad de contacto disminuye al disminuir γ. Se puede hipotetizar que en el

límite $\alpha \to \infty$, la discontinuidad de contacto desaparece completamente. De acuerdo a [6], cuando $\gamma = 1$, la simulación correspondería a la de un gas con **flujo isotérmico**. Esto último es consistente con la desaparición de la discontinuidad de contacto, ya que esta aparece cuando dos partes del gas tienen temperaturas distintas.

- Se destaca que las velocidades de los gases alcanzan distintos valores máximos; mientras más cercano a 1 es el valor de γ mayor es el valor de la velocidad u. Sin embargo, la velocidad de propagación de esta discontinuidad es distinta para cada gas y sucede al contrario que con la última observación. A mayor γ, mayor velocidad de propagación.

- La presión destaca por ser la cantidad que menos varía entre los gases simulados. La principal diferencia entre las presiones es, al igual que con la velocidad, la velocidad de propagación de las ondas. Esto sucede tanto con la onda de choque como con la de rarefacción.

5.2.2. Entropía y energía

La entropía de un gas regido por las ecuaciones de Euler, está definida por

$$S = c_v \log(p/\rho^\gamma) + \text{constante}, \tag{5.3}$$

donde c_v es la capacidad calorífica específica a volumen constante [6]. En las figuras 5.3 - 5.4 se muestran las gráficas correspondientes a la energía total (3.28) y la entropía, para cada gas, en seis instantes de tiempo. La entropía se calculó y graficó en unidades de c_v y con la constante arbitraria igual a 0.

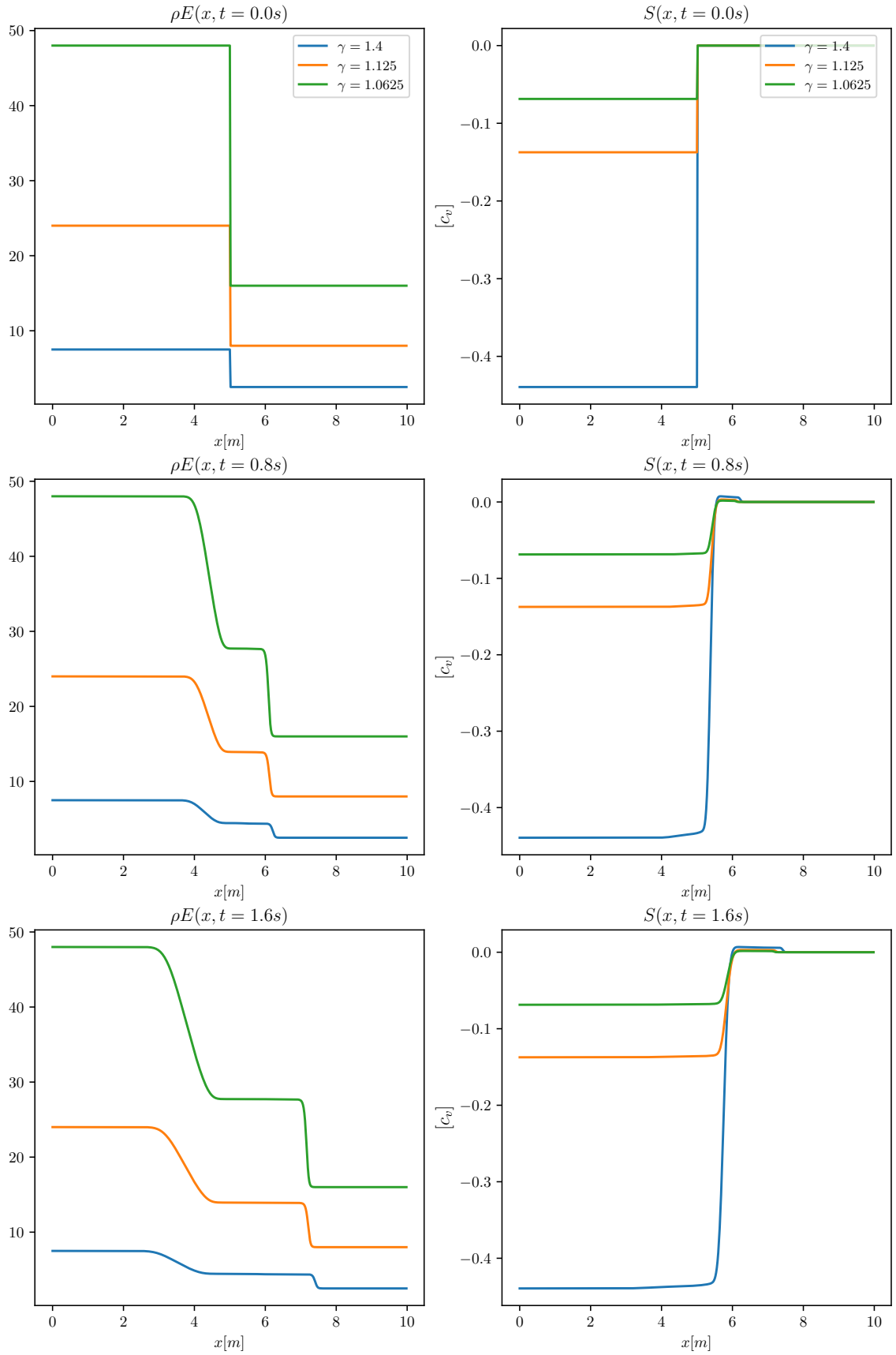


Figura 5.3. Energía y entropía en los primeros tres instantes de las simulaciones con distintos γ.

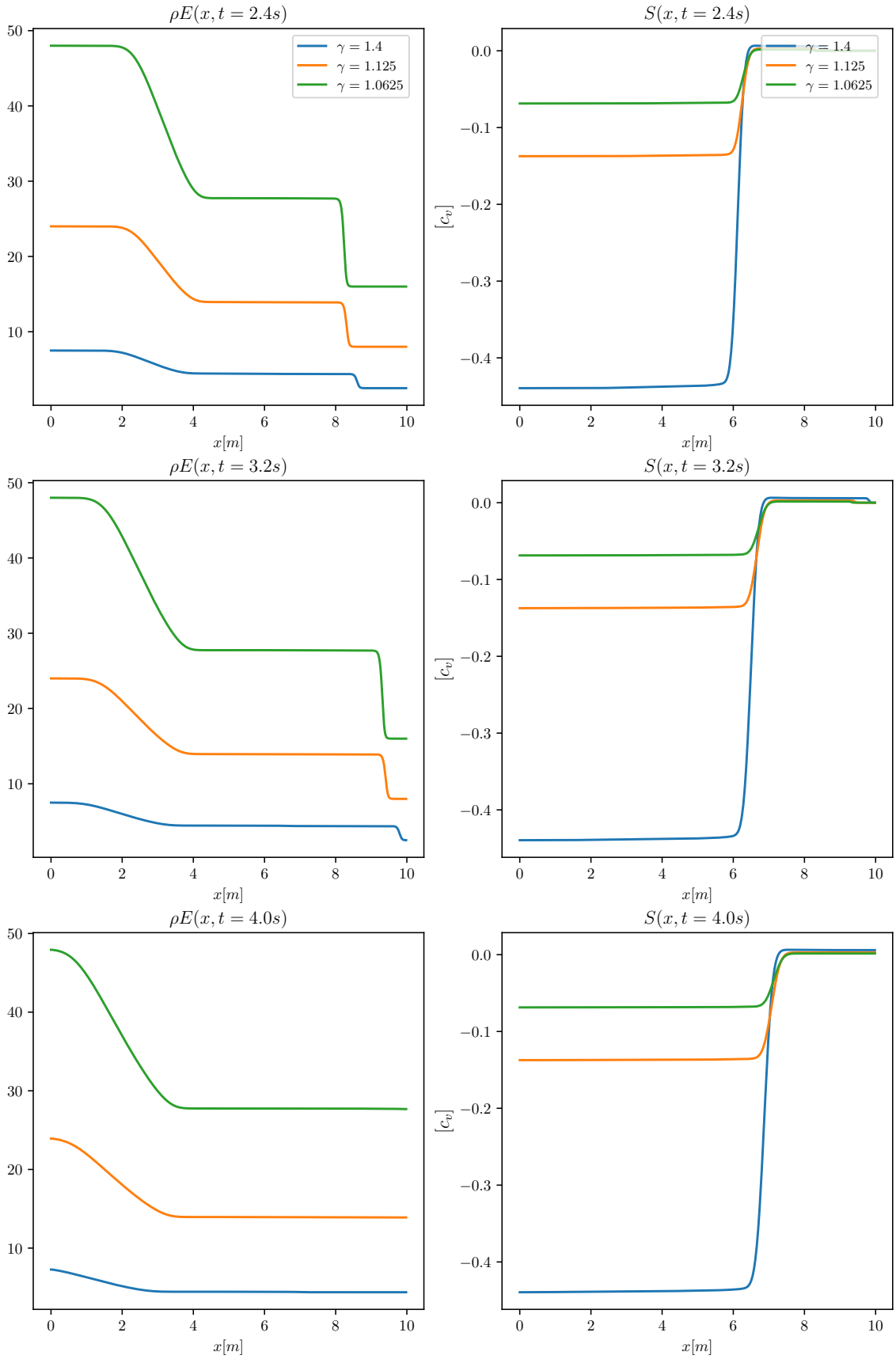


Figura 5.4. Energía y entropía en los últimos tres instantes de las simulaciones con distintos γ.

Se pueden destacar las siguientes observaciones encontradas a través de las gráficas de energía y entropía:

- La energía presenta la mayor diferencia entre los gases simulados, lo cual era esperado ya que la energía total depende explícitamente de γ. La gráfica de la energía se parece a la de presión, en el sentido que presenta una onda de rarefacción, que se traslada hacia la izquierda, y una onda de choque que se traslada hacia la derecha. Esta última característica también era esperada dado que la energía interna depende linealmente de la presión. La energía cinética contribuye principalmente a la onda de choque presentada en la energía total.

- La solución producida para la entropía destaca por coincidir con una onda de choque. La entropía tiene a ser constante mientras avanza la evolución temporal. La diferencia entre la entropía de cada gas es considerable dado que ésta también depende de γ.

- Otro hecho demostrado con la simulación es que la entropía de un gas con mayor α (menor γ) siempre es mayor que la entropía de un gas con menor α. Es sencillo justificar esta observación, puesto que cuando existen más grados de libertad internos disponibles, la energía puede repartirse de más formas; logrando así un incremento en la entropía.

CONCLUSIONES

1. El método de volúmenes finitos es óptimo para resolver ecuaciones diferenciales de conservación. Sin embargo, debe estudiarse adicionalmente un esquema apto para el cálculo de flujos numéricos aproximados entre las celdas definidas por el método.

2. El esquema de Roe es un método práctico para resolver numéricamente las ecuaciones de Euler y a través de los promedios de Roe, su implementación es directa. La parte más difícil de completar para implementar el esquema de Roe en la resolución de un sistema de conservación, es encontrar el promedio adecuado para los autovalores y autovectores de la matriz $\hat{\mathbf{A}}$. Sin embargo, las propiedades que se exigen que cumpla $\hat{\mathbf{A}}$ son sencillas de satisfacer y se pueden usar los promedios de Roe para las ecuaciones de Euler como base.

3. Las diferencias entre las simulaciones de C++ y PyClaw son menores. Sin embargo, la solución de PyClaw se realiza utilizando técnicas de alta resolución; específicamente, produciendo soluciones a segundo orden de precisión. La principal desventaja al utilizar PyClaw es su ineficiente captura de ondas de rarefacción transónica, dado que no posee algoritmos de corrección de entropía implementados.

4. El uso de las simulaciones de gases poliatómicos como experimentos, a través de la solución numérica de las ecuaciones de Euler, resulta ser una tarea que brinda intuición física suficiente para entender mejor los fenómenos de los medios continuos. A través de las simulaciones con distintos coeficientes de dilatación adiabática, se pudieron observar efectos que parecían alcanzables únicamente por medio de la solución analítica de los problemas. Además, las variables más abstractas, como la entropía, son más fáciles de comprender mediante una gráfica de la misma, obtenida a través de una simulación.

5. Las simulaciones numéricas no reemplazan completamente los experimentos de laboratorio. Sin embargo, las primeras pueden ser de alta utilidad al hacer

comparaciones de gases ficticios, tomando como ejemplo las simulaciones de gases con distintos γ. Las soluciones numéricas son un recurso de alta utilidad didáctica, especialmente en cursos básicos de la carrera de física, como termodinámica.

RECOMENDACIONES

1. Se recomienda estudiar a mayor profundidad el método de división de flujos, implementado en PyClaw, con la finalidad de añadir algún algoritmo que realice la corrección de entropía. De preferencia, agregar un conjunto de algoritmos correctores de entropía a disposición del solucionador de Euler. Esta adición es factible de realizar, ya que la librería Clawpack es un software de código abierto.

2. Ya que se estudiaron brevemente algunos esquemas distintos al esquema de Roe, se sugiere adaptar el esquema de Godunov para resolver las ecuaciones de Euler. Esto puede realizarse en un programa de C++ o bien, añadiendo un solucionador exacto en el paquete de Clawpack. Sería apropiado analizar y comparar las soluciones producidos con los esquemas de Roe y Godunov.

3. Es imprescindible considerar la implementación de métodos de impresión de datos más estandarizados, en el software de Clawpack. Esto se expone como una recomendación porque PyClaw resulta ser ligeramente limitante al momento de generar la solución numérica. Es importante reconocer, de todas maneras, que PyClaw se especializa en producir soluciones gráficas a través del paquete de visualización visClaw.

BIBLIOGRAFÍA

[1] Blundell, S.; Blundell, S.J. y Blundell, K.M.: *Concepts in Thermal Physics*. Comprehensive Assessment of Water Management in Agriculture. Oxford University Press, 2006. ISBN 9780198567691.
https://books.google.com.gt/books?id=oaDdswEACAAJ

[2] Boas, M.L.: *Mathematical Methods in the Physical Sciences*. Wiley student edition. Wiley, 2006. ISBN 9788126508105.
https://books.google.com.gt/books?id=1xV0CgAAQBAJ

[3] Cameron, Maria: «Notes on Burger's Equation», 2016.
https://www.math.umd.edu/~mariakc/burgers.pdf

[4] Clawpack Development Team: «Clawpack software», 2020. doi: https://doi.org/10.5281/zenodo.4025432. Version 5.7.1.
http://www.clawpack.org

[5] LeVeque, Randall J.: *Nonlinear Conservation Laws and Finite Volume Methods*. pp. 1–159. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-31632-9, 1998. doi: 10.1007/3-540-31632-9_1.
https://doi.org/10.1007/3-540-31632-9_1

[6] ——: *Numerical Methods for Conservation Laws*. Lectures in mathematics ETH Zürich. Birkhäuser Basel, 2013. ISBN 9783034851169.
https://books.google.com.gt/books?id=UvmpBwAAQBAJ

[7] Pelanti, M; Quartapelle, L y Vigevano, L: «A review of entropy fixes as applied to Roe's linearization». pp. 19–20. Politecnico di Milano, 2000.
https://perso.ensta-paris.fr/~pelanti/ef_PQV.pdf

[8] Pletcher, R.H.; Tannehill, J.C. y Anderson, D.: *Computational Fluid Mechanics and Heat Transfer, Second Edition*. Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Taylor & Francis, 1997.

ISBN 9781560320463.
https://books.google.com.gt/books?id=ZJPbtHeilCgC

[9] Roe, P.L: «Approximate Riemann solvers, parameter vectors, and difference schemes». *Journal of Computational Physics*, 1981, **43(2)**, pp. 357–372. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(81)90128-5.
https://www.sciencedirect.com/science/article/pii/0021999181901285

[10] ——: «Characteristic-Based Schemes for the Euler Equations». *Annual Review of Fluid Mechanics*, 2003, **18**, pp. 337–365. doi: 10.1146/annurev.fl.18.010186.002005.

[11] van Halder, Yous: «Godunov method for the Euler Equations», 2014. Disponible en https://pure.tue.nl/ws/portalfiles/portal/67742456/789123-1.pdf.