

Problema sistema hiperbólico lineal

Adrián Sanjurjo García

UDC

Marzo 2021

Índice

1. Introducción al problema	2
2. Apartado a.)	3
3. Apartado b.)	4
4. Apartado c.)	5
5. Apartado d.)	7
6. Apartado e.)	8
7. Apartado f.)	10
8. Apartado g.)	11
9. Anexo: Código en MATLAB.	13

1. Introducción al problema

Se nos plantea un sistema de ecuaciones en derivadas parciales de la forma:

$$\begin{aligned}\frac{\partial u}{\partial t} + \frac{1}{\rho_0} \frac{\partial p}{\partial x} &= 0 \\ \frac{\partial p}{\partial t} + \rho_0 c_0^2 \frac{\partial u}{\partial x} &= 0\end{aligned}\tag{1}$$

Donde ρ_0 y c_0 son la densidad y la compresibilidad del medio, ambos en reposo. Mediante distintos apartados se nos pide plantear varios resultados para este problema, comenzando por transformarlo en un sistema hiperbólico, para así simplificar las cuentas que se realizarán. Una vez obtenido, trataremos de resolver dos casos concretos: Un problema de Riemann, y un problema de Cauchy que no sea de Riemann, suponiendo en ambos casos distintas condiciones iniciales, y teniendo que implementarlos gráficamente en un código en MATLAB, el cual se adjunta con este trabajo y además se visualiza en el anexo de este mismo.

Proseguimos con las aproximaciones numéricas de ambos problemas mediante el método de Godunov, el cual aproxima por volúmenes finitos el resultado. Esto dependerá del número de Courant implementado, que hace que dichos cálculos sean más o menos precisos respecto a la solución exacta. Todo ello estará representado en distintas figuras.

2. Apartado a.)

Representarlo como sistema de leyes de conservación.

Para representarlo como un sistema de leyes de conservación, tenemos que fijarnos en que el sistema se puede describir en forma matricial, de tal forma que la ecuación 1 pasa a tener la forma:

$$\vec{\omega}_t + \vec{f}_x(\omega) = 0 \quad (2)$$

De tal forma que

$$\vec{\omega} = \begin{pmatrix} u \\ p \end{pmatrix} \quad , \quad \vec{f}(\omega) = A \vec{\omega} = \begin{pmatrix} 0 & \frac{1}{\rho_0} \\ \rho_0 c_0^2 & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} \quad (3)$$

Entonces, podemos diagonalizar la matriz A y sacar sus autovalores, lo cual nos permite cambiar la representación del sistema.

$$\det(A - \lambda \mathbb{1}_2) = \begin{vmatrix} -\lambda & \frac{1}{\rho_0} \\ \rho_0 c_0^2 & -\lambda \end{vmatrix} = \lambda^2 - c_0^2 = 0 \Rightarrow \lambda = \pm c_0 \quad (4)$$

La matriz diagonal de nuestro sistema es:

$$\Lambda = \begin{pmatrix} -c_0 & 0 \\ 0 & c_0 \end{pmatrix} \quad (5)$$

Ahora bien, para poder transformar nuestro sistema, necesitamos conocer las matrices de paso P y P^{-1} tales que $P^{-1} A P = \Lambda$, así que procedemos a calcular los autovectores.

$$\text{Ker}[A + c_0 \mathbb{1}_2] = 0:$$

$$\begin{pmatrix} c_0 & \frac{1}{\rho_0} \\ \rho_0 c_0^2 & c_0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \vec{0} \Rightarrow c_2 = -c_0 \rho_0 c_1 \Rightarrow \vec{e}_1 = \begin{pmatrix} 1 \\ -c_0 \rho_0 \end{pmatrix} \quad (6)$$

$$\text{Ker}[A - c_0 \mathbb{1}_2] = 0:$$

$$\begin{pmatrix} -c_0 & \frac{1}{\rho_0} \\ \rho_0 c_0^2 & -c_0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \vec{0} \Rightarrow c_2 = c_0 \rho_0 c_1 \Rightarrow \vec{e}_2 = \begin{pmatrix} 1 \\ c_0 \rho_0 \end{pmatrix} \quad (7)$$

Por lo que la matriz de paso P resulta:

$$P = (\vec{e}_1 \ \vec{e}_2) = \begin{pmatrix} 1 & 1 \\ -c_0 \rho_0 & c_0 \rho_0 \end{pmatrix} \quad (8)$$

Y su inversa P^{-1} :

$$P^{-1} = \frac{1}{2} \begin{pmatrix} 1 & \frac{-1}{c_0 \rho_0} \\ 1 & \frac{1}{c_0 \rho_0} \end{pmatrix} \quad (9)$$

Así pues, en la ecuación 2 hacemos el cambio de variable $\vec{\omega} = P \vec{v}$, resultando:

$$P \vec{v}_t + A P \vec{v}_x = 0 \quad (10)$$

Multiplicando la expresión 10 por P^{-1} :

$$P^{-1}P\vec{v}_t + P^{-1}A P\vec{v}_x = 0 \Rightarrow \vec{v}_t + \Lambda\vec{v}_x = 0 \Rightarrow \boxed{\begin{cases} v_{1t} - c_0 v_{1x} = 0 \\ v_{2t} + c_0 v_{2x} = 0 \end{cases}} \quad (11)$$

Entonces, en el final de la expresión 11 tenemos representado el sistema de ecuaciones en forma de un sistema hiperbólico. Además, las soluciones de dicho sistema son conocidas, siendo $v_1(x + c_0 t) = v_{01}(x + c_0 t)$ y $v_2(x + c_0 t) = v_{02}(x - c_0 t)$, correspondiéndose respectivamente con las soluciones de propagación hacia la izquierda y hacia la derecha, y v_{0i} con $i = 1, 2$ la función correspondiente a la condición inicial.

Deshaciendo la relación $\vec{\omega} = P\vec{v}$, tenemos que:

$$\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -c_0\rho_0 & c_0\rho_0 \end{pmatrix} \begin{pmatrix} v_1(x + c_0 t) \\ v_2(x - c_0 t) \end{pmatrix} = \begin{pmatrix} v_1(x + c_0 t) + v_2(x - c_0 t) \\ (-v_1(x + c_0 t) + v_2(x - c_0 t))\rho_0 c_0^2 \end{pmatrix} \quad (12)$$

3. Apartado b.)

Bajo que hipótesis sobre las constantes ρ_0 y c_0 se tiene un sistema estrictamente hiperbólico.

La condición para que se trate de un sistema estrictamente hiperbólico es que sus autovalores pertenezcan a \mathbb{R} . Como sus autovalores son $\pm c_0$, será suficiente que c_0 sea un número real distinto de cero, para que así el problema además de ser estrictamente hiperbólico, tenga también sentido físico. Puesto que la constante ρ_0 no está presente en el cálculo de los autovalores, no debería tener ninguna restricción adicional, pero también exigiremos que sea real para que el propio problema sea consistente con el sentido físico.

4. Apartado c.)

Diseñar un código informático que proporcione la solución exacta de un problema de Riemann asociado al sistema de leyes de conservación en un intervalo $[a, b]$. Tomando como datos c_0 y las condiciones iniciales del problema de Riemann u_l, p_l, u_r, p_r .

Para la realización de este problema, se ha diseñado un código en MATLAB (adjunto con este informe y en el propio anexo) que resuelve de forma exacta el problema de Riemann asociado a este sistema hiperbólico. Sabemos que la condición inicial en un problema de Riemann es de la forma:

$$\vec{\omega}(x, 0) = \vec{\omega}_0(x) = \begin{cases} \vec{\omega}_l = \begin{pmatrix} u_l \\ p_l \end{pmatrix} & x < 0 \\ \vec{\omega}_r = \begin{pmatrix} u_r \\ p_r \end{pmatrix} & x \geq 0 \end{cases} \quad (13)$$

Ahora bien, sabemos por la teoría que la solución a este problema de Riemann es de la forma:

$$\vec{\omega}(x, t) = \vec{\omega}_0(x, t) = \begin{cases} \vec{\omega}_l & x + c_0 t < 0 \\ \alpha_{1l} \vec{e}_1 + \alpha_{2r} \vec{e}_2 & (x + c_0 t)(x - c_0 t) < 0 \\ \vec{\omega}_r & x - c_0 t \geq 0 \end{cases} \quad (14)$$

Donde las constantes α_{1l} y α_{2r} se hayan a partir de $v_1(x, t)$ y $v_2(x, t)$. Para ello, calculamos $\vec{v} = P^{-1}\vec{\omega}$.

$$\begin{pmatrix} v_1(x + c_0 t) \\ v_2(x - c_0 t) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & \frac{-1}{c_0 \rho_0} \\ 1 & \frac{1}{c_0 \rho_0} \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} \frac{u - \frac{p}{\rho c_0}}{2} \\ \frac{u + \frac{p}{\rho c_0}}{2} \end{pmatrix} \quad (15)$$

Entonces, definimos las constantes α_{1l} y α_{2r} como:

$$\alpha_{1l} = \frac{u_l - \frac{p_l}{\rho c_0}}{2}, \quad \alpha_{2r} = \frac{u_r + \frac{p_r}{\rho c_0}}{2} \quad (16)$$

Para ejecutar esta solución en el código en MATLAB, hemos de seleccionar en la línea 48 del código apartado='c', puesto que esto nos calcula y representa gráficamente el resultado. Como ejemplo, vamos a hacer un caso aproximado del agua. Tomamos $\rho_0 = 1000$ y $c_0 = 1500$, así como las constantes de la condición inicial $u_l = 0$, $p_l = 10^6$, $u_r = 0$, $p_r = 10^3$. En la figura 1 podemos ver la forma que presentará la solución para un tiempo $t > 0$ cualquiera.

En ambos gráficos de 1 podemos apreciar la distinción de tres zonas, concretamente aquellas formuladas en la ecuación 14: Para la u los valores serán cero (correspondiente a u_l y u_r , a excepción de la zona en la que se cumple que $(x + c_0 t)(x - c_0 t) < 0$, mientras que para p , las zonas de la izquierda y derecha son las correspondientes a p_l y p_r , salvo la zona central, también correspondiente a aquella que cumple $(x + c_0 t)(x - c_0 t) < 0$.

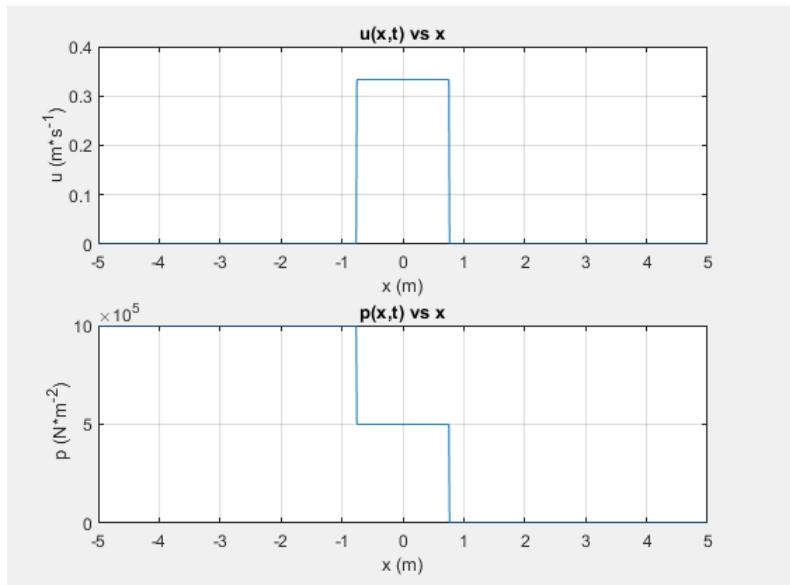


Figura 1: Representación gráfica de la solución del problema de Riemann, correspondiéndose al gráfico superior a la velocidad u y el de abajo a la presión p .

5. Apartado d.)

Diseñar un código informático que proporcione la solución exacta de un problema de Cauchy que no sea un problema de Riemann asociado al sistema de leyes de conservación. Tomando como datos las condiciones iniciales de las variables conservativas.

En este caso, el código realiza los cálculos y la representación gráfica con una condición inicial que no es de tipo Riemann. Esta condición se incorpora en las fórmulas de $v_1(x + c_0 t)$ y $v_2(x - c_0 t)$, siendo sus expresiones:

$$v_1(x + c_0 t) = -e^{-(x+c_0 t)^2} \quad , \quad v_2(x - c_0 t) = e^{-(x-c_0 t)^2} \quad (17)$$

Para ejecutarlo, volvemos a la línea 48 y reescribimos apartado='d'. Una vez se ejecute, veremos que se generan dos ondas que se propagan hacia la izquierda y la derecha, y además, la presión se propagará de forma simétrica hacia ambos sentidos. En la figura 2 se puede apreciar este comportamiento.

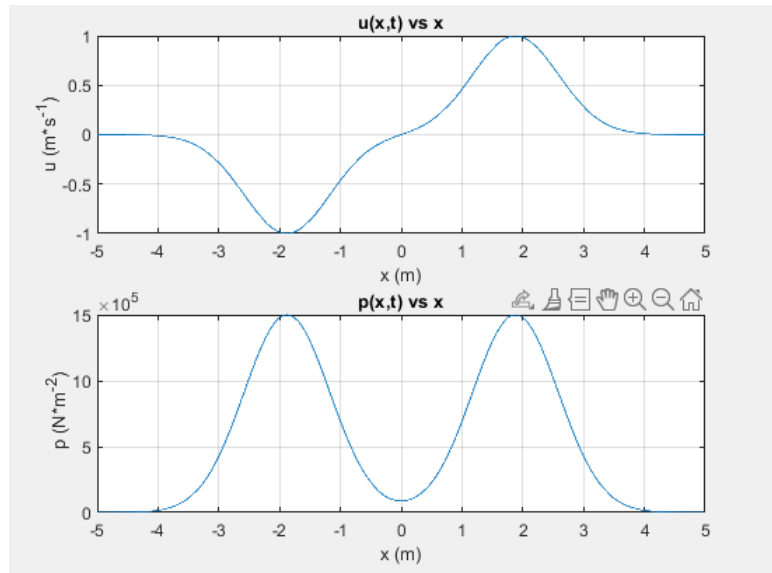


Figura 2: Representación gráfica del problema de Cauchy que no es de Riemann, tomando como condición inicial un pulso exponencial.

6. Apartado e.)

Desarrollar la discretización del problema utilizando el método de Godunov e implementarlo en el código informático. Aplicarlo a la resolución de los dos problemas de los apartados c) y d).

En este caso, vamos a tratar de resolver el problema numérico de ambos apartados anteriores mediante el método de Godunov, el cual presenta la forma:

$$\vec{\omega}_i^{n+1} = \vec{\omega}_i^n - \frac{\Delta t}{\Delta x} \left\{ \vec{f}_{i+\frac{1}{2}}^n - \vec{f}_{i-\frac{1}{2}}^n \right\} \quad (18)$$

Con el superíndice n refiriéndose al instante de tiempo n -ésimo, así como el subíndice i a la posición i -ésima del mallado de puntos de la aproximación. Definiendo además la matriz $|A|$ como $|A| = P \text{abs}(\Lambda) P^{-1}$, siendo $\text{abs}(\Lambda)$ la matriz diagonal de autovalores con todos sus autovalores en valor absoluto, tenemos que:

$$\begin{aligned} \vec{f}_{i+\frac{1}{2}}^n &= \frac{1}{2} \left(\vec{f}_i^n + \vec{f}_{i+1}^n \right) - \frac{1}{2} |A| (\vec{\omega}_{i+1}^n - \vec{\omega}_i^n) \\ \vec{f}_{i-\frac{1}{2}}^n &= \frac{1}{2} \left(\vec{f}_{i-1}^n + \vec{f}_i^n \right) - \frac{1}{2} |A| (\vec{\omega}_i^n - \vec{\omega}_{i-1}^n) \end{aligned} \quad (19)$$

En el caso particular de este problema, tenemos que la matriz $\text{abs}(\Lambda) = c_0 \mathbb{I}_2$, por lo que $|A| = P \text{abs}(\Lambda) P^{-1} = P c_0 \mathbb{I}_2 P^{-1} = c_0 \mathbb{I}_2$, por lo que las cuentas se simplificarán. Operando sobre las expresiones de 19 y sustituyendo el resultado en la expresión 18, llegamos a que la fórmula que nos aproximará el problema mediante el método de Godunov resulta:

$$\begin{pmatrix} u_i^{n+1} \\ p_i^{n+1} \end{pmatrix} = (1 - c_0 \frac{\Delta t}{\Delta x}) \begin{pmatrix} u_i^n \\ p_i^n \end{pmatrix} - \frac{\Delta t}{2\Delta x} \left\{ \begin{pmatrix} \frac{p_{i+1}^n - p_{i-1}^n}{\rho_0} \\ (u_{i+1}^n - u_{i-1}^n) \rho_0 c_0^2 \end{pmatrix} - c_0 \begin{pmatrix} u_{i+1}^n + u_{i-1}^n \\ p_{i+1}^n + p_{i-1}^n \end{pmatrix} \right\} \quad (20)$$

Fijémonos que en los extremos de un mallado de N nodos, tendremos que para los nodos 1 y N aparecerán los términos -1 y $N+1$, los cuales no están incluidos. Para omitir esta contribución, vamos a imponer en cada iteración que $\vec{\omega}_1^n = \vec{\omega}_2^n$ y $\vec{\omega}_N^n = \vec{\omega}_{N-1}^n$.

Teniendo clara la implementación de este método, el programa realiza los cálculos para aproximar los dos apartados anteriores: Si queremos realizar la aproximación del problema de Riemann, hemos de seleccionar en la línea 48 la opción apartado='er', mientras que si lo que nos interesa es el caso de del problema de Cauchy que no es de Riemann, se introduce apartado='ec'. En ambos casos, podemos realizar la comparación con la solución exacta. Esto se consigue en la línea de código 52, introduciendo comparativa='si'.

En las figura 3 y 4 representamos la comparación en ambos métodos, para así visualizar mejor el resultado, donde se puede apreciar en color azul la línea correspondiente a la aproximación por el método de Godunov en ambos casos, mientras que la naranja se refiere al resultado exacto.

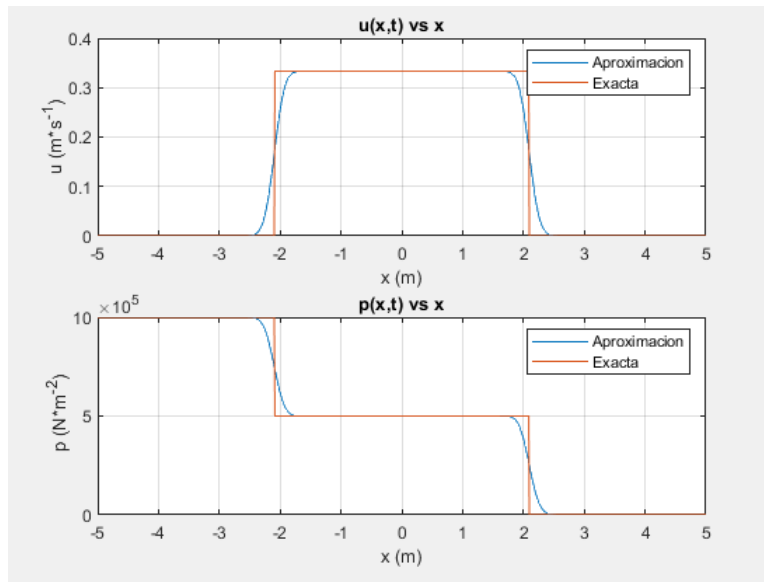


Figura 3: Comparativa entre la solución exacta y la solución aproximada del problema de Riemann

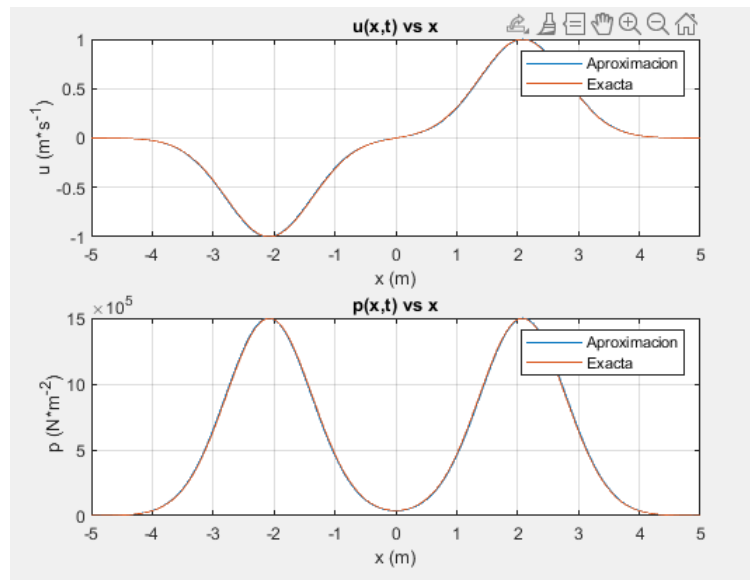


Figura 4: Comparativa entre la solución exacta y la solución aproximada del problema de Cauchy que no es de Riemann

7. Apartado f.)

Hacer un análisis de precisión del método implementado con la solución del apartado d) empleando al menos tres mallas espaciales del intervalo $[a, b]$.

Para realizar un análisis de precisión, calculamos la norma del error con las normas 1 e infinito:

$$\text{Norma 1} \longrightarrow \epsilon_{\Delta x}^1(t^f) = \Delta x \sum_{j=1}^M \|\omega(x_j, t^f) - \omega_j^N\| \quad (21)$$

$$\text{Norma infinito} \longrightarrow \epsilon_{\Delta x}^\infty(t^f) = \max \|\omega(x_j, t^f) - \omega_j^N\| \quad (22)$$

El interior de los valores absolutos se corresponden con la resta entre el valor aproximado $\omega(x_j, t^f)$ y el exacto ω_j^N , donde j se corresponde al nodo j -ésimo, t^f al tiempo final de ejecución, y M el número de nodos del mallado. Veamos que al variar esta cantidad M también varía el valor de Δx . Mediante esta variación, si tenemos dos mallados de interespaciado Δx_k y Δx_{k+1} los cuales satisfacen la relación $\frac{\Delta x_{k+1}}{\Delta x_k} = 2$, podemos definir el orden de precisión como:

$$p_k = \frac{\ln(\mathbb{K}_k)}{\ln(2)} \quad , \quad \mathbb{K}_k = \frac{\epsilon_{\Delta x_k}(t^f)}{\epsilon_{\Delta x_{k+1}}(t^f)} \quad , \quad k = 1, 2, \dots \quad (23)$$

Comenzamos el cálculo de los errores de la velocidad u para un mallado de 1000 nodos y mantenemos fijo el número de Courant en $\mu = 0,9$, y repetimos el proceso para 2000 nodos y 3000 nodos, de acuerdo a la expresión [23](#).

Nodos	$\epsilon_{\Delta x}^1(t^f)$	p_k	$\epsilon_{\Delta x}^\infty(t^f)$	p_k
1000	0.01159121	1.00000124	0.20946289	0
2000	0.00579560	0.42040201	0.20946289	0.048011716
3000	0.00380645	—	0.20260685	—

8. Apartado g.)

Analizar desde el punto de vista computacional la estabilidad del método numérico implementado en función del número de Courant.

Siendo acordes a la teoría, para que el método numérico sea estable se tiene que cumplir que:

$$\mu = c_0 \frac{\Delta t}{\Delta x} \leq 1 \quad (24)$$

Donde μ es el llamado número de Courant, y todas las cantidades que lo definen son positivas. En caso de que se cumpla la igualdad, tendremos que el método se ajusta por completo al resultado exacto, mientras que si el número de Courant es mayor que 1, el método no es convergente.

En el código del programa se implementa en la línea 27 la posibilidad de elegir el número de Courant del método, lo cual cambiará el diferencial de tiempo, mientras que deja intacto el espacial. Vamos a visualizar en este apartado los tres casos posibles para el número de Courant, y así comprobar cuan precisa es dicha aproximación del Método de Godunov. En este caso, vamos a representar el caso del problema de Riemann, pero también se podría visualizar con el problema del apartado d.).

Tomamos pues como números de Courant $\mu = 0,5$, $\mu = 1$ y $\mu = 1,5$ en las siguientes figuras.

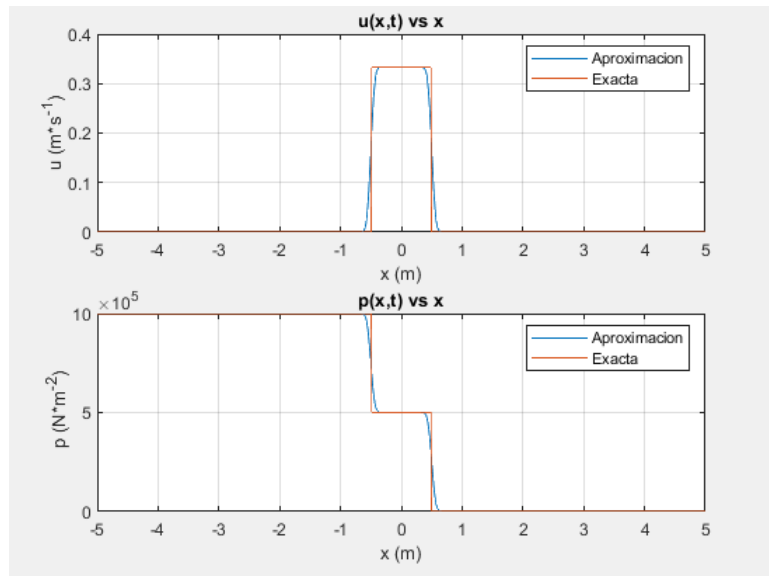


Figura 5: Comparativa del método de Godunov para el problema de Riemann con $\mu = 0,5$

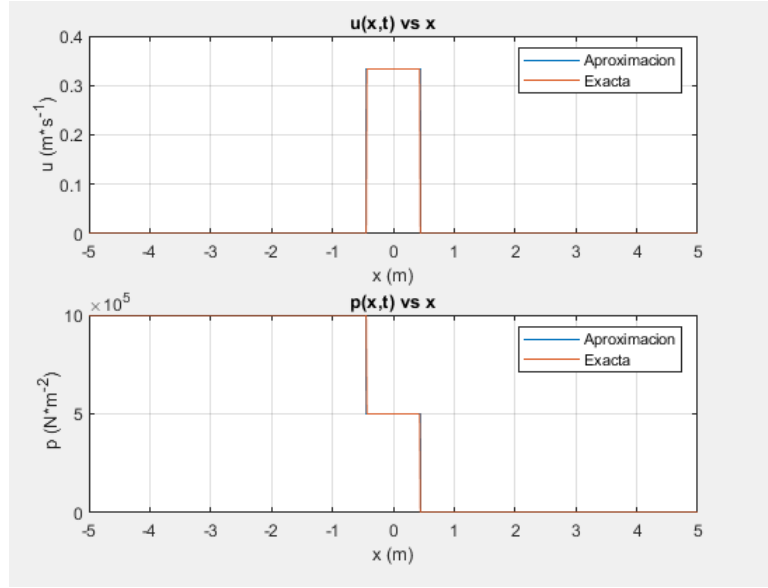


Figura 6: Comparativa del método de Godunov para el problema de Riemann con $\mu = 1$

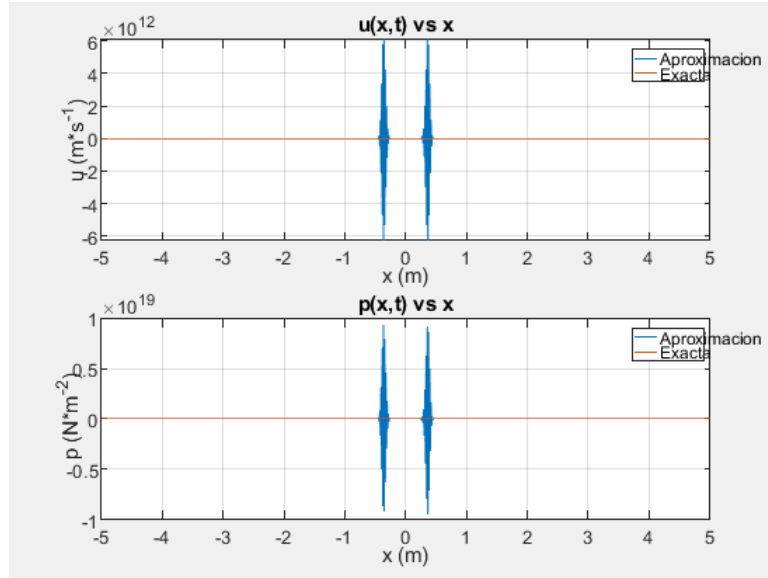


Figura 7: Comparativa del método de Godunov para el problema de Riemann con $\mu = 1,5$

En la figura 5 se puede deducir que la aproximación no es completamente correcta pero se ajusta en la medida de lo posible, de forma similar a la representada en la figura 3, mientras que en 6 podemos ver claramente una aproximación exacta, ajustándose de igual forma ambas curvas. Sin embargo, en el gráfico 7 se ve que la curva no se ajusta nada la solución exacta, obteniendo órdenes de magnitud mucho más grandes de lo correcto, siendo esto acorde a la expresión 24.

9. Anexo: Código en MATLAB.

```
clear
format long

% Instrucciones de uso:
% El programa toma inicialmente las constantes del problema planteado. Se
% pueden modificar al gusto. En cada una viene indicado que magnitud fisica
% representa

% Ecuacion de ondas

c0=1500;           % Velocidad de propagacion de la onda en el medio
rho=1000;          % Densidad del medio

%Condiciones iniciales
ur=0;             % Velocidad a la derecha
ul=0;             % Velocidad a la izquierda
pl=1e6;           % Presion a la izquierda
pr=1e3;           % Presion a la derecha
%P=max([pl,pr]);   % En caso de querer otro tipo de escalado,
%descomentar esta linea junto con las lineas en las cuales los comandos
%axis estan comentados.

%Variables caracteristicas en el centro
alf1R=0.5*(ur-pr/(c0*rho));
alf2L=0.5*(ul+pl/(c0*rho));

N=3000;           % Numero de puntos del mallado
x=linspace(-5,5,N); % Vector de mallado
t=0;              % Instante de tiempo inicial
mu=0.9;
Dx=(x(N)-x(1))/N; % Espacio entre puntos del mallado
Dt=Dx*mu/c0;      % Espacio entre tiempos. Esta no tocarla, depende de mu y Dx
it_temp=40;        % Numero de nodos espaciales que queremos.
tfin=Dt*it_temp;   % Tiempo final de ejecucion

% Ejes modificados de las graficas. Descomentar si se quiere para tener
% otro escalado, junto con lo que ya se ha hablado anteriormente

% xmin=x(1);
% xmax=x(N);
% yminp=-2*P;
% ymaxp=2*P;

% Eleccion del apartado que queremos representar. Posibilidades:
% apartado="c"
% apartado="d"
% apartado="er" se corresponde con el apartado e y problema de riemann
% apartado="ec" apartado e y problema de cauchy del apartado d
% Es importante la eleccion de estas y solo estas opciones para el uso
% correcto del programa. En caso contrario no ejecuta los calculos.

apartado="er";
% Para el apartado e se pueden comparar los resultados de los apartados
% anteriores con los de este, eligiendo la opcion "si". En caso contrario,
% elegir "no". Cualquier otra opcion no imprime las graficas.
comparativa="si";

if apartado=="c" | apartado=="d"
    while t<tfin
        % Bucle correspondiente a la representacion del problema de Riemann
        % con solucion exacta.
        if apartado=="c"
            % Solucion del apartado c en la iteracion i-esima
            [U,P]=riemann(x,t,c0,rho,ur,ul,pr,pl,alf1R,alf2L);
```

```

elseif apartado=="d"
    % Solucion del apartado d en la iteracion i-esima
    [U,P]=cauchy(x,t,c0,rho);
end
% Grafica correspondiente a las velocidades
subplot(2,1,1);
plot(x,U);
%axis equal
title("u(x,t) vs x")
xlabel("x (m)")
ylabel("u (m*s^{-1})")
grid
% Grafica correspondiente a las presiones
subplot(2,1,2);
plot(x,P);
%axis([xmin xmax yminp ymaxpl])
title("p(x,t) vs x")
xlabel("x (m)")
ylabel("p (N*m^{-2})")
grid
t=t+Dt;
pause(Dt)
end
elseif apartado=="er" | apartado=="ec" % Ejecuta esta parte de codigo
    u_i_1=zeros(size(x)); % en el apartado e.
    p_i_1=zeros(size(x));
    if apartado=="er"
        % Apartado e aplicado al problema de riemann
        for i=1:N
            if x(i)<=0
                u_i_1(i)=ul;
                p_i_1(i)=pl;
            elseif x(i)>0
                u_i_1(i)=ur;
                p_i_1(i)=pr;
            end
        end
    end
    elseif apartado=="ec"
        [u_i_1,p_i_1]=cauchy(x,0,c0,rho); % Solucion de Cauchy
    end
    u_i=u_i_1;
    p_i=p_i_1;
    while t<tfin %El tiempo maximo elegido es suficiente para ver el
        comportamiento de las funciones
        for i=2:N-1
            u_i_1(i)=u_i*(1-Dt*c0/Dx)-(0.5*Dt/Dx)*((p_i(i+1)-p_i(i-1))/rho -
            c0*(u_i(i+1)+u_i(i-1)));
            p_i_1(i)=p_i*(1-Dt*c0/Dx)-(0.5*Dt/Dx)*((u_i(i+1)-u_i(i-1))*rho*
            c0^2 -c0*(p_i(i+1)+p_i(i-1)));
        end
        u_i_1(1)=u_i_1(2);
        u_i_1(N)=u_i_1(N-1);
        p_i_1(1)=p_i_1(2);
        p_i_1(N)=p_i_1(N-1);
        if comparativa=="si"
            if apartado=="er"
                [U,P]=riemann(x,t,c0,rho,ur,ul,pr,pl,alf1R,alf2L);
            elseif apartado=="ec"
                [U,P]=cauchy(x,t,c0,rho);
            end
        end
        subplot(2,1,1);
        if comparativa=="si"
            plot(x,u_i_1,x,U);
            legend("Aproximacion","Exacta")
        elseif comparativa=="no"
            plot(x,u_i_1);
        else
            fprintf("No has seleccionado una opcion correcta")
        end
    end
end

```

```

        break
    end
    %axis equal
    title("u(x,t) vs x")
    xlabel("x (m)")
    ylabel("u (m*s^{-1})")
    grid
    % Grafica correspondiente a las presiones
    subplot(2,1,2);
    if comparativa=="si"
        plot(x,p_i_1,x,P);
        legend("Aproximacion","Exacta")
    elseif comparativa=="no"
        plot(x,p_i_1);
    else
        fprintf("No has seleccionado una opcion correcta")
        break
    end
    %axis([xmin xmax yminp ymaxpl])
    title("p(x,t) vs x")
    xlabel("x (m)")
    ylabel("p (N*m^{-2})")
    grid
    t=t+Dt;
    pause(Dt)
    u_i=u_i_1;
    p_i=p_i_1;
end
if comparativa=="si"
    fprintf("Error de aproximacion de la velocidad u")
    err_u_1=error_un(u_i_1,U,Dx) % error en norma 1
    err_u_2=error_inf(u_i_1,U) % error en norma infinito
    fprintf("Error relativo de aproximacion de la presion p") % hacemos ←
        el error relativo para apreciar el error respecto al valor inicial←
        pl, que es del orden de 1e6
    err_p_1=error_un(p_i_1,P,Dx)/pl % error "relativo" en norma 1
    err_p_2=error_inf(p_i_1,P)/pl % error "relativo" en norma ←
        infinito
end
else
    fprintf("No has seleccionado una opcion correcta")
end

% Funciones de calculo del problema
% Funcion del problema de Riemann

function [U,P]=riemann(x,t,c0,rho,ur,ul,pr,pl,alf1R,alf2L)
    %%% Funcion de velocidades del problema de Riemann para el sistema
    %%% hiperbolico de ondas.
    n=size(x,2);
    U=zeros(n,0);
    P=zeros(n,0);
    for i=1:n
        if x(i)+c0*t<=0
            U(i)=ul;
            P(i)=pl;
        elseif (x(i)+c0*t)*(x(i)-c0*t)<0
            U(i)=alf1R+alf2L;
            P(i)=(-alf1R+alf2L)*rho*c0;
        elseif x(i)-c0*t>0
            U(i)=ur;
            P(i)=pr;
        end
    end
end

% Funcion del problema de Cauchy que no es de Riemann para el apartado d.
function [U,P]=cauchy(x,t,c0,rho)

```

```

    % Pulsos correspondientes a los invariantes
    T=zeros(size(x));           % Array de tiempos necesario para las operaciones ←
    posteriores.
    T(:,:)=t
    v1_pulso=-exp(-(x+c0*T).^2);
    v2_pulso=exp(-(x-c0*T).^2);
    U=v1_pulso+v2_pulso;
    P=(-v1_pulso+v2_pulso)*rho*c0;
end

function err= error_un(X,Y,d)
    err=0;                      % d es el interespaciado
    for i=1:size(X,2)
        err=err+abs(X(i)-Y(i));
    end
    err=err*d;
end
function err=error_inf(X,Y)
    err=max(abs(X-Y));
end

```