

# Métodos numéricos en fluidos – Instituto Balseiro – 2021

## Práctica 3

1. Mediante el análisis de número de onda modificado muestre que la utilización del esquema de segundo orden y hacia un lado para la derivada segunda:

$$\frac{\partial^2 u}{\partial x^2} \Big|_j \approx \frac{-u_{j+3} + 4u_{j+2} - 5u_{j+1} + 2u_j}{\Delta x^2},$$

conduce a inestabilidades si se aplica a la ecuación de difusión ( $u_t - \alpha u_{xx} = 0$ ).

2. Suponga que decide avanzar la ecuación de onda de primer orden ( $u_t + c u_x = 0$ ) con el método de Leap-Frog para la parte temporal y con diferencias centradas de segundo orden o Padé de cuarto orden para la parte espacial. Compare para cada caso el máximo número CFL ( $\text{CFL} = c \Delta t / \Delta x$ ) que resulta en un esquema estable. Note que ocurre con el CFL cuando se mejora la resolución espacial.

3. El siguiente esquema numérico ha sido propuesto para resolver la ecuación de onda de primer orden:

$$\frac{1}{\Delta t} [u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)] = \frac{c}{2\Delta x} (u_{j+1}^n - u_{j-1}^n).$$

- (a) Encuentre el rango para el CFL que hace el método estable.
- (b) ¿Es el método consistente?

4. El esquema ADI (o de factorización aproximada) recomendado para la ecuación de difusión en 3D es:

$$\begin{aligned} (I - \alpha \Delta t / 2 A_x) \mathbf{u}^* &= [I + \alpha \Delta t (A_x / 2 + A_y + A_z)] \mathbf{u}^{(n)}, \\ (I - \alpha \Delta t / 2 A_y) \mathbf{u}^{**} &= \mathbf{u}^* - \alpha \Delta t / 2 A_y \mathbf{u}^{(n)}, \\ (I - \alpha \Delta t / 2 A_z) \mathbf{u}^{n+1} &= \mathbf{u}^{**} - \alpha \Delta t / 2 A_z \mathbf{u}^{(n)}. \end{aligned}$$

- (a) Muestre que el método es incondicionalmente estable y halle el orden de aproximación.
- (b) Discuta como aplicaría las condiciones de contorno en esta formulación.

5. Considere la ecuación de Burgers en 1D:

$$u_t + u u_x = \alpha u_{xx}$$

en  $0 \leq x \leq 1$ , con las siguientes condiciones de borde:

$$u(0, t) = 0, \quad u(1, t) = 0.$$

- (a) Si se utiliza un esquema implícito de segundo orden para la parte temporal y esquemas centrados de segundo orden para la parte espacial; escriba la forma discreta de la ecuación, las matrices involucradas, y explique como sería el algoritmo que utilizaría para resolver la ecuación.
  - (b) ¿Cómo podría evitar iteraciones en su algoritmo?
6. **(Problema a entregar - 18/10/2021)** Las imágenes sísmicas son utilizadas en una gran variedad de aplicaciones, desde explotaciones petrolíferas hasta observaciones médicas no intrusivas. Se quiere estudiar numéricamente un modelo unidimensional de un problema de imágenes sísmicas para ver los efectos que tiene una velocidad del sonido variable entre diferentes medios sobre la transmisión y reflexión de una onda acústica. Considere la ecuación de onda escalar, homogénea y unidimensional

$$\frac{\partial^2 u}{\partial t^2} - c^2(x) \frac{\partial^2 u}{\partial x^2} = 0 \quad t \geq 0 \quad -\infty < x < \infty \quad (1)$$

con condiciones iniciales

$$u(x, 0) = u_0(x) \quad u_t(x, 0) = 0$$

donde  $c > 0$  es la velocidad del sonido. El dominio  $x$  para este problema es infinito. Para poder abordarlo numéricamente se trunca el dominio en  $0 \leq x \leq 4$ . Sin embargo, para hacer esto es necesario especificar algunas condiciones en los bordes del dominio,  $x = 4$  y  $x = 0$ , de modo que las ondas calculadas viajen suavemente saliendo del dominio computacional como si éste se extendiera hasta el infinito. Una “condición radiativa” (condición radiativa de Sommerfeld) especificaría que en  $\infty$  todas las ondas son salientes, lo cual es necesario para que el problema esté bien planteado. En casos unidimensionales, esta condición puede ser aplicada en forma exacta en un  $x$  finito: se desea sólo ondas salientes en los bordes del dominio. Es decir, en  $x = 4$  se quiere que la solución numérica mantenga sólo ondas que viajan hacia la derecha y en  $x = 0$  sólo aquellas que lo hacen hacia la izquierda. Si se factorizan los operadores en la ecuación de onda puede verse en forma más explícita qué es lo que debe hacerse (asumiendo  $c$  constante por el momento)

$$\left( \frac{\partial}{\partial t} - c \frac{\partial}{\partial x} \right) \left( \frac{\partial}{\partial t} + c \frac{\partial}{\partial x} \right) u = 0 \quad (2)$$

La parte de la solución que viaja hacia la derecha es

$$\left( \frac{\partial}{\partial t} + c \frac{\partial}{\partial x} \right) u = 0 \quad (3)$$

y la que viaja hacia la izquierda es

$$\left( \frac{\partial}{\partial t} - c \frac{\partial}{\partial x} \right) u = 0 \quad (4)$$

de modo que en  $x = 4$  es necesario resolver la ecuación (3) en lugar de la ecuación (1) para asegurar una solución saliente. Del mismo modo, en  $x = 0$  hay que resolver la ecuación (4) en lugar de la ecuación (1).

Para el avance en el tiempo es recomendable que la ecuación (1) sea descompuesta en dos ecuaciones de primer orden en el tiempo

$$\frac{\partial u_1}{\partial t} = u_2 \quad y \quad \frac{\partial u_2}{\partial t} = c^2(x) \frac{\partial^2 u_1}{\partial x^2}$$

Las condiciones de borde resultan

$$\left. \frac{\partial u_1}{\partial t} \right|_{x=0} = c(0) \left. \frac{\partial u_1}{\partial x} \right|_{x=0} \quad \left. \frac{\partial u_1}{\partial t} \right|_{x=4} = -c(4) \left. \frac{\partial u_1}{\partial x} \right|_{x=4}$$

Se recomienda utilizar esquemas de segundo orden para la derivada espacial. Este problema requiere una elevada precisión para la solución y encontrará que son necesarios al menos  $N = 400$  puntos. Compare una solución con menos puntos con una que considere precisa. Se recomienda utilizar Runge-Kutta de cuarto orden para el avance temporal.

¿Qué valor de  $c$  debería ser usado para estimar el máximo  $\Delta t$  permitido para obtener una solución estable? Estimar el máximo paso de tiempo permitido a través del análisis del número de onda modificado. Tome  $u(x, 0) = \exp[-200(x - 0.25)^2]$  especificando  $c(x)$  según:

- (a) Arenisca porosa:  $c(x) = 1$ .
- (b) Transición a arenisca impermeable:  $c(x) = 1.25 - 0.25 \tanh[40(0.75 - x)]$ .
- (c) Arenisca impermeable:  $c(x) = 1.5$ .
- (d) Nave alienígena sepultada:  $c(x) = 1.5 - \exp[-300(x - 1.75)^2]$ .

Grafique  $u(x)$  para diferentes tiempos ( $\sim 8$ ) a medida que la onda se propaga a través de todo el dominio.

**Instituto Balseiro**

MÉTODOS NUMÉRICOS EN FLUIDOS I



LABORATORIO N°3

Renato Trono Figueras

23 de Octubre de 2021

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Implementación del método</b>	<b>2</b>
2.1. Semidiscretización espacial . . . . .	2
2.2. Condiciones de borde . . . . .	3
2.3. Avance temporal . . . . .	3
<b>3. Resultados</b>	<b>4</b>
3.1. Estabilidad con número de onda modificado . . . . .	4
3.2. Comparación entre soluciones con distinta resolución espacial . . . . .	5
3.3. Soluciones para las distintas velocidades del sonido $c(x)$ . . . . .	6
<b>4. Conclusiones</b>	<b>7</b>
<b>5. Apéndice</b>	<b>8</b>
5.1. Código en Python para la resolución numérica . . . . .	8

## Resumen

Se resolvió numéricamente la ecuación de onda homogénea unidimensional en un dominio espacial finito, representando una onda acústica propagándose en un medio sólido. Se aplicaron las correspondientes condiciones de borde para simular un dominio infinito. Las derivadas espaciales se aproximaron con diferencias finitas centradas de segundo orden, mientras que para el avance temporal se implementó Runge-Kutta de cuarto orden. Se comprobó que el problema necesita de una alta resolución espacial para obtener una solución aceptable y que no aparezcan oscilaciones espúreas. Para ello, se compararon soluciones con distintas resoluciones espaciales. Además, se estudió la estabilidad del método utilizado mediante el análisis del número de onda modificado. Se halló que el máximo paso de tiempo para mantener estabilidad es proporcional al paso espacial e inversamente proporcional a la máxima velocidad de propagación de la onda. Finalmente, se compararon soluciones para distintas velocidades de propagación, simulando distintos medios en los cuales la onda se propagaba.

## 1. Introducción

Se desea resolver numéricamente la ecuación de onda unidimensional y homogénea

$$\frac{\partial^2 u}{\partial t^2} - c^2(x) \frac{\partial^2 u}{\partial x^2} = 0 \quad t \geq 0 \quad -\infty \leq x \leq \infty \quad (1)$$

con condiciones iniciales

$$u(x, 0) = u_0(x) \quad u_t(x, 0) = 0$$

En el caso de este trabajo, la función  $u(x, t)$  representa una onda acústica propagándose a través de un medio sólido, con  $c(x)$  la velocidad del sonido en dicho medio. Si bien el dominio espacial del problema es infinito, para resolver

numéricamente la ecuación hay que truncarlo. Se considera  $x \in [0, 4]$ . Así, deben aplicarse condiciones de borde que simulen un dominio infinito. Las mismas están dadas por [1]

$$\frac{\partial u}{\partial t}(0, t) = c(0) \frac{\partial u}{\partial x}(0, t) \quad (2)$$

$$\frac{\partial u}{\partial t}(4, t) = -c(4) \frac{\partial u}{\partial x}(4, t) \quad (3)$$

Se aplicará un esquema de diferencias centradas de segundo orden para aproximar la derivada espacial, y el avance temporal se hará mediante el método de Runge-Kutta de cuarto orden.

Se desea estudiar la estabilidad del método mediante el análisis del número de onda modificado. Además, se quiere resolver la ecuación para distintas velocidades del sonido, representando diferentes medios. Los mismos son:

- Arenisca porosa:  $c(x) = 1$
- Transición a arenisca impermeable:  $c(x) = 1,25 - 0,25 \tanh [40(0,75 - x)]$
- Arenisca impermeable:  $c(x) = 1,5$
- Nave alienígena sepultada:  $c(x) = 1,5 - e^{-300(x-1,75)^2}$

La condición inicial se considera  $u_0(x) = e^{-200(x-0,25)^2}$

El problema en cuestión requiere de una elevada precisión para la solución, y son necesarios al menos  $N = 400$  puntos [1]. Se compararán soluciones de menos cantidad de puntos con otras con suficiente precisión.

## 2. Implementación del método

En esta sección se detalla la implementación del método numérico utilizado. Como se mencionó antes, se utilizan diferencias centradas de segundo orden para la semidiscretización espacial y Runge-Kutta de cuarto orden para el avance temporal.

### 2.1. Semidiscretización espacial

En primer lugar, se aproxima la segunda derivada espacial en los nodos internos con diferencias finitas centradas de segundo orden. En el nodo  $j$  y tiempo  $t_n = n\Delta t$ , se tiene:

$$\frac{d^2 u_j^n}{dt^2} = \frac{c^2(x_j)}{\Delta x^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \quad (4)$$

donde el subíndice  $j$  denota nodo espacial  $j$  y el superíndice  $n$  denota tiempo  $t_n$ . Luego, se convierte la ecuación diferencial ordinaria de segundo orden (4) en dos ecuaciones diferenciales ordinarias de primer orden, como sigue:

$$\begin{aligned} u_1 &= u \\ u_2 &= \frac{\partial u}{\partial t} \end{aligned}$$

En el nodo  $j$  y tiempo  $t_n$  se obtiene:

$$\frac{du_{1j}^n}{dt} = u_{2j}^n \quad (5)$$

$$\frac{du_{2j}^n}{dt} = \frac{c^2(x_j)}{\Delta x^2} (u_{1j+1}^n - 2u_{1j}^n + u_{1j-1}^n) \quad (6)$$

## 2.2. Condiciones de borde

El método requiere un tratamiento especial en los bordes del dominio espacial. La ecuación a resolver en  $x = 0$  es la ecuación (2). Además, se usan diferencias adelantadas de segundo orden. Así, en el nodo  $j = 0$  se tiene:

$$\frac{du_{10}^n}{dt} = \frac{c(0)}{2\Delta x} (-3u_{10}^n + 4u_{11}^n - u_{12}^n) \quad (7)$$

Por otro lado, la ecuación a resolver en  $x = 4$  es la ecuación (3). Además, se usan diferencias atrasadas de segundo orden. Así, en el nodo  $j = N + 1$  se tiene:

$$\frac{du_{1N+1}^n}{dt} = \frac{-c(4)}{2\Delta x} (3u_{1N+1}^n - 4u_{1N}^n + u_{1N-1}^n) \quad (8)$$

donde  $N$  es la cantidad de nodos internos.

## 2.3. Avance temporal

Con lo expuesto en 2.1 y 2.2, se tiene el siguiente sistema de ecuaciones:

$$\frac{d\bar{u}_1^n}{dt} = \begin{pmatrix} \frac{c(0)}{2\Delta x} (-3u_{10} + 4u_{11} - u_{12}) \\ u_{21} \\ u_{22} \\ \vdots \\ u_{2N} \\ \frac{-c(4)}{2\Delta x} (3u_{1N+1} - 4u_{1N} + u_{1N-1}) \end{pmatrix}^n = \bar{f}(\bar{u}_1^n, \bar{u}_2^n), \quad \bar{f} \in \mathbb{R}^{(N+2) \times 1} \quad (9)$$

$$\frac{d\bar{u}_2^n}{dt} = \frac{1}{\Delta x^2} A \begin{pmatrix} u_{11} \\ u_{12} \\ \vdots \\ u_{1N} \end{pmatrix}^n + \begin{pmatrix} \frac{c^2(x_1)}{\Delta x^2} u_{10} \\ 0 \\ \vdots \\ 0 \\ \frac{c^2(x_N)}{\Delta x^2} u_{1N+1} \end{pmatrix}^n = \bar{g}(\bar{u}_1^n, \bar{u}_2^n), \quad \bar{g} \in \mathbb{R}^{N \times 1} \quad (10)$$

con  $A \in \mathbb{R}^{N \times N}$  tal que:

$$A = \begin{bmatrix} -2c^2(x_1) & c^2(x_1) & 0 & \cdots & \cdots & 0 \\ c^2(x_2) & -2c^2(x_2) & c^2(x_2) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & c^2(x_{N-1}) & -2c^2(x_{N-1}) & c^2(x_{N-1}) \\ 0 & \cdots & \cdots & 0 & c^2(x_N) & -2c^2(x_N) \end{bmatrix}$$

La implementación de Runge-Kutta de cuarto orden para el avance temporal se llevó a cabo como sigue [2]:

$$\bar{u}_1^{n+1} = \bar{u}_1^n + \frac{1}{6}(\bar{k}_1 + 2\bar{k}_2 + 2\bar{k}_3 + \bar{k}_4) \quad (11)$$

$$\bar{u}_2^{n+1} = \bar{u}_2^n + \frac{1}{6}(\bar{l}_1 + 2\bar{l}_2 + 2\bar{l}_3 + \bar{l}_4) \quad (12)$$

donde

$$\bar{k}_1 = \bar{f}(\bar{u}_1^n, \bar{u}_2^n)\Delta t \quad \bar{l}_1 = \bar{g}(\bar{u}_1^n, \bar{u}_2^n)\Delta t$$

$$\bar{k}_2 = \bar{f}(\bar{u}_1^n + \bar{k}_1/2, \bar{u}_2^n + \bar{l}_1/2)\Delta t \quad \bar{l}_2 = \bar{g}(\bar{u}_1^n + \bar{k}_1/2, \bar{u}_2^n + \bar{l}_1/2)\Delta t$$

$$\bar{k}_3 = \bar{f}(\bar{u}_1^n + \bar{k}_2/2, \bar{u}_2^n + \bar{l}_2/2)\Delta t \quad \bar{l}_3 = \bar{g}(\bar{u}_1^n + \bar{k}_2/2, \bar{u}_2^n + \bar{l}_2/2)\Delta t$$

$$\bar{k}_4 = \bar{f}(\bar{u}_1^n + \bar{k}_3, \bar{u}_2^n + \bar{l}_3)\Delta t \quad \bar{l}_4 = \bar{g}(\bar{u}_1^n + \bar{k}_3, \bar{u}_2^n + \bar{l}_3)\Delta t$$

### 3. Resultados

En esta sección se presentan los resultados obtenidos para el análisis de estabilidad, para la comparación entre soluciones con  $N < 400$  y  $N > 400$ , y para las soluciones para las distintas velocidades del sonido  $c(x)$ .

#### 3.1. Estabilidad con número de onda modificado

Para hacer análisis de estabilidad con número de onda modificado, se asume una solución del tipo

$$u(x, t) = \psi e^{ikx} \quad (13)$$

Reemplazando (13) en (1), se obtiene:

$$\psi'' + c^2 k^2 \psi = 0$$

lo cual corresponde al caso del problema continuo.



Para el problema discreto, reemplazamos la derivada espacial continua por el esquema en diferencias utilizado, que en este caso es diferencias centradas de segundo orden. Se obtiene:

$$\psi'' e^{ikx_j} - \frac{c^2(x_j)}{\Delta x^2} (e^{ikx_{j+1}} - 2e^{ikx_j} + e^{ikx_{j-1}}) \psi = 0$$

dividiendo por  $e^{ikx_j}$  y operando se llega a:

$$\psi'' + \frac{2c^2(x_j)\psi}{\Delta x^2} [1 - \cos(k\Delta x)] = \psi'' + c^2(x_j)k'^2\psi = 0 \quad (14)$$

donde  $k' = \sqrt{\frac{2}{\Delta x^2} [1 - \cos(k\Delta x)]}$  es el número de onda modificado.

Definiendo  $\psi_1 = \psi$  y  $\psi_2 = \psi'$ , se obtiene:

$$\begin{pmatrix} \psi_1' \\ \psi_2' \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -k'^2 c^2(x_j) & 0 \end{bmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} = A\bar{\psi} \quad (15)$$

Los autovalores de  $A$  son  $\lambda = \pm ik'c(x_j)$ . De ahí

$$|\lambda| = k'|c(x_j)| \quad (16)$$

Por otro lado, se tiene que para el método de Runge-Kutta de cuarto orden con  $\lambda$  imaginario puro

$$\Delta t \leq \frac{2,8}{|\lambda|} \quad (17)$$

Reemplazando (16) en (17) se llega a

$$\Delta t \leq \frac{2,8\Delta x}{c(x_j)\sqrt{2[1 - \cos(k\Delta x)]}} \quad (18)$$

El caso que restringe más a  $\Delta t$  es  $\cos(k\Delta x) = -1$  y el valor máximo de  $c(x)$ , denominado  $c_{max}$ . Con eso, el método es estable si se cumple:

$$\Delta t \leq \frac{1,4\Delta x}{c_{max}} \quad (19)$$

Así, la relación entre el paso de tiempo máximo y el paso espacial es lineal. Además, el paso de tiempo máximo es inversamente proporcional a la máxima velocidad de propagación de la onda.



### 3.2. Comparación entre soluciones con distinta resolución espacial

A continuación, en la Figura 1, se muestra una comparación entre las soluciones para  $N = 200$  y  $N = 600$ , donde  $N$  es el número de nodos internos de la discretización espacial, con  $c(x) = 1$ .

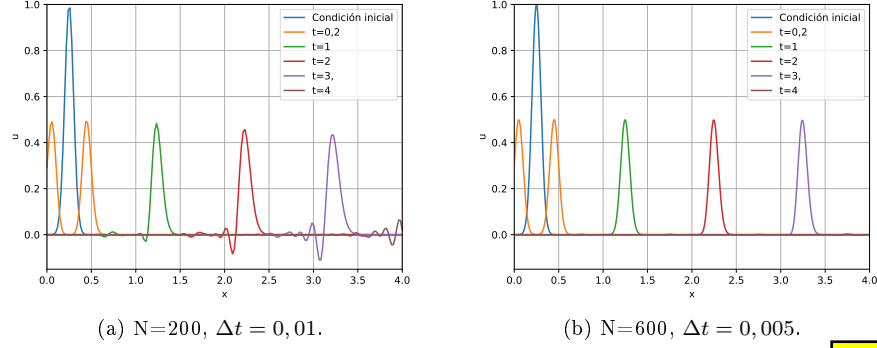


Figura 1: Comparación de dos soluciones con distinta resolución espacial, para  $c(x) = 1$ .

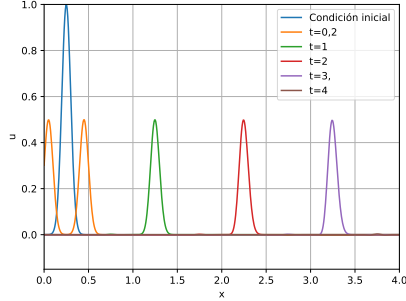
Según la inecuación (19), en ambos casos el método es estable. Aún así, para  $N = 200$  se observan oscilaciones espúreas que deforman la onda. Por el contrario, para  $N = 600$ , la onda se propaga sin deformarse y se obtiene una solución aceptable.

### 3.3. Soluciones para las distintas velocidades del sonido $c(x)$

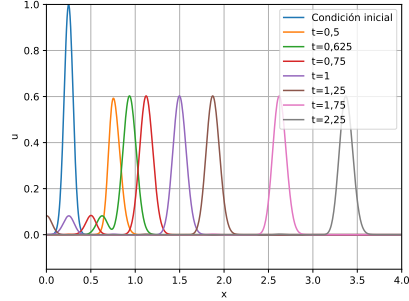
A continuación, se exponen las soluciones obtenidas para las distintas velocidades del sonido  $c(x)$ . En todos los casos, se utilizaron  $N = 600$  nodos internos para la discretización espacial. Con eso, según la inecuación (19) y teniendo en cuenta el  $c_{max}$  en cada caso, los máximos pasos de tiempo para tener estabilidad resultaron:

- Arenisca porosa:  $\Delta t_{max} \simeq 9,3 \times 10^{-3}$
- Transición a arenisca impermeable:  $\Delta t_{max} \simeq 6,2 \times 10^{-3}$
- Arenisca impermeable:  $\Delta t_{max} \simeq 6,2 \times 10^{-3}$
- Nave alienígena sepultada:  $\Delta t_{max} \simeq 6,2 \times 10^{-3}$

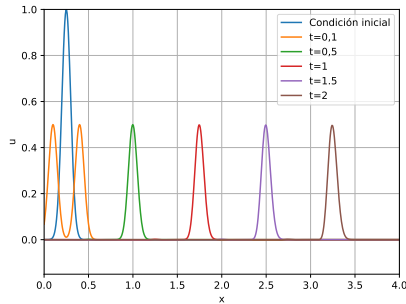
En todos los casos, se utilizó  $\Delta t = 5 \times 10^{-3}$ . En la Figura 2 se muestran las soluciones para las distintas velocidades del sonido  $c(x)$ .



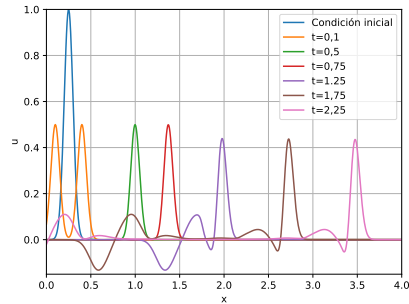
(a) Arenisca porosa



(b) Transición a arenisca impermeable



(c) Arenisca impermeable



(d) Nave alienígena sepultada

Figura 2: Comparación de las soluciones para distintos  $c(x)$ , con  $N = 600$  y  $\Delta t = 0,005$ .

La Figura 2a coincide con la Figura 1b. En la Figura 2b se puede ver que en algún instante entre  $t = 0,5$  y  $t = 0,625$  la onda se encuentra con la transición de arenisca porosa a arenisca impermeable, donde la velocidad pasa de  $c = 1$  a  $c = 1,5$ . Allí, parte de la onda rebota, lo cual puede verse en los tiempos posteriores como una perturbación viajando hacia la izquierda.

En la Figura 2c, que corresponde a arenisca impermeable, se ve algo similar a lo que pasa en arenisca porosa, aunque por los tiempos y posiciones se nota la diferencia en la velocidad de propagación.

Finalmente, en la Figura 2d se ve como en algún instante entre  $t = 0,75$  y  $t = 1,25$  la onda se encuentra con la nave alienígena sepultada y ocurre algo similar al caso de la transición entre los dos tipos de areniscas. En el caso de la nave, la parte de la onda que rebota y vuelve hacia la izquierda es mayor.

## 4. Conclusiones

Se resolvió numéricamente la ecuación de onda homogénea unidimensional en un dominio espacial truncado, representando una onda acústica propagándose en un medio sólido. Se aplicaron las condiciones de borde correspondientes para simular un dominio infinito.

Para la semidiscretización espacial se utilizó un esquema de diferencias finitas centradas de segundo orden mientras que para el avance temporal se implementó Runge-Kutta de cuarto orden.

Se pudo ver que el problema requiere alta resolución en la discretización espacial, comparando dos soluciones, una con 200 nodos internos y otra con 600 nodos internos. Por otro lado, se estudió la estabilidad del método mediante el análisis del número de onda modificado. Se halló que el máximo paso de tiempo para tener estabilidad se relaciona linealmente con el paso espacial y es inversamente proporcional a la máxima velocidad de propagación de la onda. Con eso, se estimó el máximo paso de tiempo que se podía utilizar en los distintos casos estudiados.

Finalmente, se compararon soluciones para distintas velocidades de propagación, simulando distintos medios por los que se propagaba la onda.

## Referencias

- [1] Practica 3, Cátedra de Métodos Numéricos en Fluidos I, Instituto Balseiro, 2021
- [2] "Fundamentals of Engineering Numerical Analysis", P. Moin, Cambridge University Press, 2010

## 5. Apéndice

### 5.1. Código en Python para la resolución numérica

El código para la resolución numérica se encuentra en la página siguiente.

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
```

```
N=600
```

```
delta_x=4/(N+1)
```

```
delta_t=0.005
```

```
T_final=8
```

```
def tridiag (ld,pd,ud,N):
```

```
    #crea matriz de ceros de dimension NxN
```

```
    A=np.zeros((N,N))
```

```
    #Llena la diagonal principal
```

```
    i=0
```

```
    j=0
```

```
    while i<=N-1:
```

```
        while j<=N-1:
```

```
            A[i,j]=pd
```

```
            j=j+1
```

```
            i=i+1
```

```
    #Llena la subdiagonal
```

```
    i=1
```

```
    j=0
```

```
    while i<N-1:
```

```
        while j<N-1:
```

```
            A[i,j]=ld
```

```
            j=j+1
```

```
            i=i+1
```

```
    #Llena la supradiagonal
```

```
    i=0
```

```
    j=1
```

```
    while i<N-1:
```

```
        while j<=N-1:
```

```

        A[i,j]=ud
        j=j+1
        i=i+1
    return A

```

```

def u1_inicial(x):
    u=np.exp(-200*(x-0.25)**2)
    return u

```

```

def vel_c(x):
    #c=1.5-np.exp(-300*(x-1.75)**2)
    c=1
    #c=1.5
    #c=1.25-0.25*np.tanh(40*(0.75-x))
    return c

```

```

indice_t=int(T_final/delta_t)
indice_x=N
u1=np.zeros((indice_t,int(indice_x+2)))
u2=np.zeros((indice_t,indice_x))

```

```

x=np.linspace(0,4,N+2)
c=np.zeros((N+2,1))
for i in range(0,N+2):
    c[i]=vel_c(x[i])
    u1[0,i]=u1_inicial(x[i])

```

```

dc_2= tridiag(1, -2, 1, N)
dc_2=1/(delta_x**2)*dc_2

```

```

for i in range(0,N):
    dc_2[i,:]=c[i]**2*dc_2[i,:]

```

```

for i in range(0,indice_t-1):

    k1=np.transpose(u2[i,:])

    k1=np.append(k1,-c[N+1]/(2*delta_x)*(3*u1[i,N+1] - 4*u1[i,N] + u1[i,N-1]))

    k1=np.insert(k1,0,c[0]/(2*delta_x)*(-3*u1[i,0] + 4*u1[i,1] - u1[i,2]))

    k1=delta_t*k1

    q=np.zeros((N,1))

    q[0]=c[1]**2/delta_x**2*u1[i,0]

    q[N-1]=c[N]**2/delta_x**2*u1[i,N+1]

    l1=dc_2@np.transpose(u1[i,1:-1]) + q[:,0]

    l1=delta_t*l1

    k2=np.transpose(u2[i,:]) + l1/2

    k2=np.append(k2,-c[N+1]/(2*delta_x)*(3*(u1[i,N+1] + k1[N+1]/2) - 4*(u1[i,N] + k1[N]/2) + u1[i,N-1] +
k1[N-1]/2))

    k2=np.insert(k2,0,c[0]/(2*delta_x)*(-3*(u1[i,0] + k1[0]/2) + 4*(u1[i,1] + k1[1]/2) - u1[i,2] - k1[2]/2))

    k2=k2*delta_t

    q=np.zeros((N,1))

    q[0]=c[1]**2/delta_x**2*(u1[i,0] + k1[0]/2)

    q[N-1]=c[N]**2/delta_x**2*(u1[i,N+1] + k1[N+1]/2)

    l2=dc_2@(np.transpose(u1[i,1:-1])+1/2*k1[1:-1]) + q[:,0]

    l2=l2*delta_t

    k3=np.transpose(u2[i,:]) + l2/2

    k3=np.append(k3,-c[N+1]/(2*delta_x)*(3*(u1[i,N+1] + k2[N+1]/2) - 4*(u1[i,N] + k2[N]/2) + u1[i,N-1] +
k2[N-1]/2))

    k3=np.insert(k3,0,c[0]/(2*delta_x)*(-3*(u1[i,0] + k2[0]/2) + 4*(u1[i,1] + k2[1]/2) - u1[i,2] - k2[2]/2))

    k3=k3*delta_t

    q=np.zeros((N,1))

    q[0]=c[1]**2/delta_x**2*(u1[i,0] + k2[0]/2)

    q[N-1]=c[N]**2/delta_x**2*(u1[i,N+1] + k2[N+1]/2)

    l3=dc_2@(np.transpose(u1[i,1:-1])+1/2*k2[1:-1]) + q[:,0]

    l3=l3*delta_t

    k4=np.transpose(u2[i,:]) + l3

    k4=np.append(k4,-c[N+1]/(2*delta_x)*(3*(u1[i,N+1] + k3[N+1]) - 4*(u1[i,N] + k3[N]) + u1[i,N-1] +
k3[N-1]))

    k4=np.insert(k4,0,c[0]/(2*delta_x)*(-3*(u1[i,0] + k3[0]) + 4*(u1[i,1] + k3[1]) - u1[i,2] - k3[2]))

```

```

k4=k4*delta_t
q=np.zeros((N,1))
q[0]=c[1]**2/delta_x**2*(u1[i,0] + k3[0])
q[N-1]=c[N]**2/delta_x**2*(u1[i,N+1] + k3[N+1])
l4=dc_2@(np.transpose(u1[i,1:-1])+k3[1:-1]) + q[:,0]
l4=l4*delta_t
u1[i+1,:]=u1[i,:] + 1/6*(k1 + 2*k2 + 2*k3 + k4)
u2[i+1,:]=u2[i,:] + 1/6*(l1 + 2*l2 + 2*l3 + l4)

```

```

fig1, ax1 =plt.subplots()
ax1.plot(x,u1[0,:], label='Condición inicial')
ax1.plot(x,u1[40,:], label = 't=0,2')
ax1.plot(x,u1[200,:], label = 't=1')
ax1.plot(x,u1[400,:], label = 't=2')
ax1.plot(x,u1[600,:], label = 't=3,')
ax1.plot(x,u1[800,:], label = 't=4')

```

```

ax1.legend()
ax1.grid()
ax1.set_xlabel('x')
ax1.set_ylabel('u')
ax1.set_xlim(0,4)
ax1.set_ylim(-0.15,1)
#fig1.savefig('N200.pdf')

```