

UNIVERSIDAD DE CÁDIZ

FACULTAD DE CIENCIAS

GRADO EN MATEMÁTICAS

SIMULACIÓN NUMÉRICA DE ECUACIONES DE ONDA.

Trabajo de fin de grado presentado por

Ana Cuevas de Cózar

Tutores: Dr. José Rafael Rodríguez Galvan

D. Daniel Acosta Soba

Firma de la alumna

Firma de los tutores

Puerto Real, Cádiz, Junio de 2.023

Abstract

In this work we will ~~adress~~ the study of the partial differential equation for wave phenomena, from a numerical perspective. We will establish a theoretical physics foundation to understand the origin of the wave equation and its derivation.

To proceed further, we are going to tackle the conservation law of the wave equation and the traveling wave type solutions.

Afterwards, using the finite differences method, we aim to obtain a numerical solution, by doing a discretization in space and time to approximate it. We will reintroduce the former aspects back from this perspective. We will also be conducting an acoustic and ~~grafical~~ comparative, varying the parameters and conditions discussed in the previous chapters to verify their effects on it.

To sum up, we are going to present the remarks obtained throughout the work, and the relevant results.

A mis padres.

Resumen

En este trabajo trataremos el estudio de la ecuación de onda en derivadas parciales, desde una perspectiva numérica, explorando diferentes enfoques. Comenzaremos estableciendo una base física para comprender el origen de la ecuación de onda y como se deriva.

A continuación, abordaremos aspectos como la ley de la conservación de la energía o las soluciones de tipo onda viajera.

Luego, empleando el método de las diferencias finitas obtendremos una solución numérica de la ecuación de onda, haciendo una discretización en espacio y tiempo para aproximarla. También retomaremos los aspectos anteriormente nombrados desde este punto de vista. Seguiremos realizando una comparativa, gráfica y sonora de las aproximaciones obtenidas, cambiando los parámetros y condiciones obtenidas en los capítulos anteriores para comprobar su efecto en la misma.

Por último, presentaremos las conclusiones obtenidas a lo largo de este trabajo, resumiendo las observaciones y los resultados relevantes.

Agradecimientos

Cuando comencé el grado en matemáticas con todos mis compañeros, creo que ninguno sabíamos como era esta carrera. A lo largo de los años me enorgullezco en decir que hemos construido amistades duraderas, pasado juntos momentos complicados y aprendido mucho más de lo que podíamos imaginar.

Por todo esto quiero agradecerles a ellos y sobre todo a mi compañera y amiga Marina por estar a mi lado en cada paso del camino, porque gracias a eso hemos conseguido avanzar juntas.

Creo que todos tenemos que agradecer a nuestros profesores de la carrera, por intentar que aprendamos y seamos críticos. Sobre todo a mis directores de este trabajo por su paciencia infinita y su ayuda en todo momento.

También debo agradecer a mis padres, por creer siempre en mi y escucharme cada día, aún cuando ellos no comprendían lo que les contaba. Gracias, por construirme como persona y hacerme llegar a donde estoy.

Por último quiero agradecer a mi novio, porque estos últimos años, tan duros a la par que gratificantes, ha sabido llenarme de confianza.

Gracias a todos y espero poder devolveros algún día todo lo que me habéis dado.

Ana Cuevas de Cózar

junio 2023

Índice general

1	Introducción	1
2	Conceptos previos	3
2.1	Espacios de Lebesgue	3
2.2	Producto interior y suma por partes en L^2	4
2.3	Dominio de dependencia	8
3	Definición de la ecuación de onda	11
3.1	La ecuación de onda unidimensional	12
3.2	Clasificación de la ecuación de onda	14
3.3	Soluciones de tipo onda viajera	15
3.4	La energía en la ecuación de onda	17
4	Resolución numérica de la ecuación de onda	21
4.1	Métodos principales de resolución	21
4.2	Método de diferencias finitas	21
4.3	Consistencia	23
4.4	Análisis de Von Neumann	25
4.5	Resolución numérica para las soluciones de tipo viajero	28
4.6	Ley de energía discreta	29
5	Modelado en Python y comparativa	33
5.1	Soluciones	33
5.1.1	$f(x) = \sin(\pi x)$, $g(x) = 0$	33
5.1.2	$f(x)$ tipo Struck, $g(x) = 0$	35
5.1.3	$f(x) = 0$, $g(x)$ tipo Struck	36

ÍNDICE GENERAL

5.1.4	$f(x) = \sin(2\pi x), g(x) = 0$	37
5.2	Soluciones tipo onda viajera	38
5.3	Comparación variando el valor de μ	40
5.3.1	$\mu \ll 1$	40
5.3.2	$\mu > 1$	41
5.4	Comparar cambiando la energía	42
6	Conclusiones y proyectos futuros	43
A	Anexo	45
A.1	Resolución numérica	45
A.2	Wave	52
A.3	Imágenes	54
	Bibliografía	59

Introducción

Los grandes avances en la computación y la tecnología en los últimos años, han logrado que las matemáticas, y en particular el campo de las ecuaciones diferenciales, hayan cambiado notablemente.

En la actualidad, los métodos numéricos nos brindan la oportunidad de obtener aproximaciones cada vez más precisas de la realidad en situaciones en las que antes teníamos que conformarnos con soluciones parciales o marcos muy específicos.

A lo largo del trabajo, utilizaremos estas técnicas para observar la resolución de la ecuación de onda en derivadas parciales.

La belleza de obtener numéricamente estas soluciones, es que, conforme la potencia de los dispositivos va incrementando, las gráficas y distintos tipos de archivos que podemos crear a partir de las mismas, mejorará de forma significativa.

Nosotros nos centraremos en dar una visión general de la resolución de la ecuación de onda, en el capítulo 2 nos adentraremos en los conceptos previos necesarios para la comprensión del trabajo, como los Espacios de Lebesgue, el concepto de producto interior, o el mallado que utilizaremos para definir nuestro esquema numérico.

1. INTRODUCCIÓN

En el capítulo siguiente, 3, introduciremos la ecuación de onda desde un marco físico y realizaremos una clasificación de la misma. Posteriormente nos centraremos en las soluciones de tipo onda viajera. Por último, analizaremos la energía de la ecuación y veremos en qué condiciones se cumplirá la Ley de la conservación de la energía.

Una vez tenemos todos los cimientos contruidos, en el capítulo 5, haremos uso de uno de los métodos más utilizados para la resolución numérica, el método de las diferencias finitas. Continuando el capítulo, analizaremos la consistencia y estabilidad de las soluciones obtenidas. Acabaremos trasladando tanto las soluciones de tipo onda viajera como la conservación de la energía al ámbito discreto.

Concluiremos el grueso del trabajo, llevando a la práctica la teoría de los capítulos anteriores mediante un script de Python. Este se puede encontrar en el Anexo A y en el Github del trabajo **Simulación numérica de ecuaciones de onda**. En el capítulo 5, comparamos las soluciones obtenidas en varios sentidos, desde cambiando las condiciones iniciales del sistema, hasta variar las variables obtenidas en los capítulos anteriores.

Como cierre, en el apartado de conclusiones 6, sintetizaremos los hallazgos obtenidos y presentaremos sugerencias para seguir explorando en esta área de estudio.

Conceptos previos

Para facilitar la comprensión del tema que vamos a tratar hay varios conceptos que debemos explicar con anterioridad, en ello nos centraremos en este capítulo.

2.1 Espacios de Lebesgue

Comenzaremos definiendo los espacios de Lebesgue. Un concepto clásico que se utiliza de manera recurrente en varios campos de las matemáticas [1].

DEFINICIÓN 2.1.1: Espacio de Lebesgue

Para $E \subset \mathbb{R}$ medible y $p \in \mathbb{N}$ tal que $1 \leq p \leq \infty$ definimos, el Espacio de Lebesgue $L^p(E)$, como la colección de funciones medibles, iguales en casi todo que son p -integrables, es decir, las funciones f para las cuales

$$\|f\|_{L^p(E)} = \left(\int_E |f|^p \right)^{\frac{1}{p}} < \infty.$$



En lo que sigue del trabajo trabajaremos sobre este tipo de espacios. Sigamos viendo que características tendrán nuestros espacios.

2. CONCEPTOS PREVIOS

DEFINICIÓN 2.1.2: Espacio de Banach

Un espacio de Banach $(E, \|\cdot\|)$ es un espacio vectorial normado, en el que E es completo bajo la métrica inducida por la norma $\|\cdot\|$. Es decir, cada serie de Cauchy en E es una serie convergente.

Por lo tanto, tomando $\|f\|_{L^p(E)}$ como la norma asociada a $L^p(E)$, este será un espacio de Banach siempre que E sea completo.

DEFINICIÓN 2.1.3: Espacio de Hilbert

Un espacio vectorial E se llama de Hilbert si tiene un producto interior asociado y es completo bajo su métrica.



El producto interior elegido podría tomar muchas formas, en nuestro caso, trabajaremos principalmente sobre L^2 y por lo tanto vamos a definir un producto interior en el mismo.

2.2 Producto interior y suma por partes en L^2

DEFINICIÓN 2.2.1: Producto interior en L^2

Sean f y g dos funciones definidas en E , definimos el producto interior en $L^2(E)$, de ellas como:

$$\langle f, g \rangle_E = \int_E f(x)g(x)dx.$$

La norma inducida por este producto interior, será la que utilizaremos. Por lo tanto $L^2(E)$, suponiendo E completo, será un espacio de Hilbert 2.1.3.



Para funciones que dependen de más de una variable, como por ejemplo del espacio y el tiempo, $u(x, t)$, fijando un instante t , quedará:

$$\langle u_1(\cdot, t), u_2(\cdot, t) \rangle_E = \int_E u_1(x, t)u_2(x, t)dx, \quad \|u(\cdot, t)\|_E = \sqrt{\langle u(x, t), u(x, t) \rangle_E}. \quad (2.1)$$

Esto a partir de ahora lo denotaremos como $\langle u_1, u_2 \rangle_E$ y $\|u\|_E$ respectivamente.

En este punto nuestro objetivo será llegar a una generalización discreta de este concepto, ya que lo usaremos en la sección 4.6.

DEFINICIÓN 2.2.2: Producto interior en dominios discretos

Sean f y g dos funciones definidas en un conjunto de puntos $D = \{x_1, x_2, \dots, x_m\}$ equidistantes, donde $h = x_{j+1} - x_j$. Definiremos el producto interior sobre D como:

$$(f, g)_{L^2(D)} = \sum_{j=1}^m h f(x_j) g(x_j).$$



La norma asociada a este producto escalar es: $\|f\|_{L^2(D)} = \sqrt{\langle f, f \rangle_D} \geq 0$.

Ahora, necesitaremos extrapolar esta definición 2.2.2 a un retículo o mallado. Para ello vamos a definirlo, tal y como lo usaremos.

DEFINICIÓN 2.2.3: Mallado

Sea $[a, b] \times [c, d]$ con $b \geq a, d \geq c$ y sean $n, m \in \mathbb{N}$. Definimos h como tamaño del mallado en espacio y el paso de tiempo k , de forma que los puntos de nuestro mallado serán:

$$(x, t)_{j,r} = (x_j, t_r), \quad \text{donde}$$

$$x_j = jh \quad \text{para} \quad j = 0, 1, \dots, m,$$

$$t_r = rk \quad \text{para} \quad r = 0, 1, \dots, n.$$

Vemos que nuestro mallado sería del tipo:

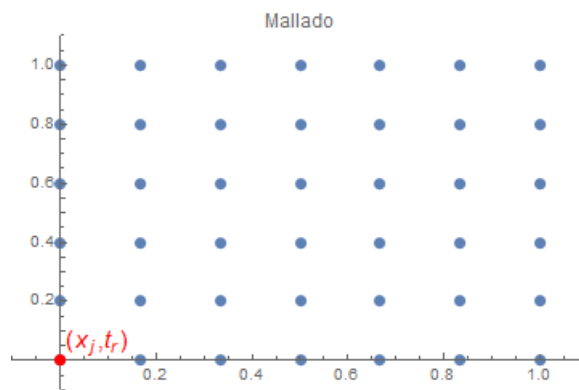


Figura 2.1: Figura 1.

Una vez ya definido este mallado veamos como se extendería la definición 2.2.2 para unas funciones u, w definidas en los puntos del mismo. Antes que nada tendremos que fijar

2. CONCEPTOS PREVIOS

el dominio en el que nos estaremos moviendo, ya que la definición es para un conjunto de puntos en una sola dimensión. Supongamos que nuestro mallado es el de la definición 2.2.3 y que necesitamos aplicar el producto interior en la posición, x . En este caso, fijando el instante de tiempo t_r , nos quedará de la forma:

$$(u(\cdot, t_r), w(\cdot, t_r))_{L^2(D)} = \sum_{j=1}^m h u(x_j, t_r) w(x_j, t_r), \quad (2.2)$$

donde $D = \{x_1, \dots, x_m\}$ definido en nuestro mallado.

Para este tipo de casos usaremos la siguiente notación:

$$(u_{\cdot, r}, w_{\cdot, r})_{L^2(D)} = \sum_{j=1}^m h u_{j, r} w_{j, r}. \quad (2.3)$$

En este punto, una vez definidas las operaciones que nos harán falta a lo largo del trabajo, y para simplificar la notación que usaremos, necesitaremos los siguientes conceptos:

DEFINICIÓN 2.2.4: Desplazamiento progresivo y regresivo

Sean u una función discretizada en el mallado antes descrito. Definiremos el desplazamiento hacia delante (resp. desplazamiento hacia atrás) como:

$$e_{x+} u_{j, r} = u_{j+1, r}, \quad (e_{x-} u_{j, r} = u_{j-1, r} \text{ respectivamente.})$$

Siguiendo de esto podríamos aproximar la derivada de primer orden de la siguiente forma.

DEFINICIÓN 2.2.5: Aproximaciones progresivas, regresivas y centradas

Sea u una función discretizada en un mallado como el la definición 2.2.3. Definiremos las diferencias aproximadas hacia adelante, atrás o centradas (más conocidas como *Forward*, *backward* y *centered difference approximations*) respectivamente, como:

- $\delta_{x+} u_{j, r} = \frac{1}{h} (u_{j+1, r} - u_{j, r}),$
- $\delta_{x-} u_{j, r} = \frac{1}{h} (u_{j, r} - u_{j-1, r}),$
- $\delta_x u_{j, r} = \frac{1}{2h} (u_{j+1, r} - u_{j-1, r}).$

Podemos hacer lo mismo para el eje temporal, denotándolas como $\delta_{t+}, \delta_{t-}, \delta_t$

Para segundo orden definimos,

DEFINICIÓN 2.2.6: Aproximación centrada de la derivada de segundo orden

Sea u una función discretizada en un mallado como el de la definición 2.2.3. Definimos la aproximación centrada de la derivada de segundo orden como:

$$\begin{aligned}\delta_x u_{j,r} &= \delta_{x^+} \delta_{x^-} (u_{j,r}) = \frac{1}{h^2} (e_{x^+} u_{j,r} - 2u_{j,r} + e_{x^-} u_{j,r}) = \\ &= \frac{1}{h^2} (u_{j+1,r} - 2u_{j,r} + u_{j-1,r}) \approx \frac{\partial^2}{\partial x^2} u_{j,r}.\end{aligned}$$

Al igual que la definición anterior esto será análogo en el eje temporal.

Para finalizar la sección, hay otro concepto que tendremos que introducir. Este nos ayudará a hacer una representación discreta de la integración por partes.

LEMA 2.2.1: Representación de la suma por partes

Sean u, w funciones definidas en el mallado 2.2.3 y sea $D = \{x_1, \dots, x_m\}$, la representación de la suma por partes utilizando el producto interior que usaremos en lo que sigue es, fijando un instante temporal r :

$$(u_{\cdot,r}, \delta_{x^+} w_{\cdot,r})_{L^2(D)} = -(\delta_{x^-} u_{\cdot,r}, w_{\cdot,r})_{L^2(D)} \quad (2.4)$$



$$(u_{\cdot,r}, \delta_{x^-} w_{\cdot,r})_{L^2(D)} = -(\delta_{x^+} u_{\cdot,r}, w_{\cdot,r})_{L^2(D)} - u_{0,r} w_{-1,r} + u_{m+1,r} w_{m,r}. \quad (2.5)$$



Demostremos las anteriores igualdades:



$$\begin{aligned}(u_{\cdot,r}, \delta_{x^+} w_{\cdot,r})_{L^2(D)} &= \sum_{j=0}^m h u_{j,r} \frac{1}{h} (w_{j+1,r} - w_{j,r}) = \\ &= - \sum_{j=0}^m h \frac{1}{h} (u_{j,r} - u_{j-1,r}) w_{j,r} + \underline{u_{m,r} w_{m+1,r}} - \underline{u_{-1,r} w_{0,r}} = \\ &= -(\delta_{x^-} u_{\cdot,r}, w_{\cdot,r})_{L^2(D)}\end{aligned}$$

2. CONCEPTOS PREVIOS

$$\begin{aligned}(u_{\cdot,r}, \delta_x w_{\cdot,r})_{L^2(D)} &= \sum_{j=0}^m h u_{j,r} \frac{1}{h} (w_{j,r} - w_{j-1,r}) = \\ &= - \sum_{j=0}^m h \frac{1}{h} (u_{j+1,r} - u_{j,r}) w_{j,r} - u_{0,r} w_{-1,r} + u_{m+1,r} w_{m,r} = \\ &= - (\delta_x u_{\cdot,r}, w_{\cdot,r})_{L^2(D)} - u_{0,r} w_{-1,r} + u_{m+1,r} w_{m,r}.\end{aligned}$$

Vemos que estos últimos términos se salen del mallado que hemos definido, estos serán puntos virtuales del retículo.



2.3 Dominio de dependencia

Como en la sección anterior hemos definido el mallado 2.2.3 que usaremos, vamos ahora a definir el concepto de dominio de dependencia, que estará estrechamente relacionado con el mismo.

DEFINICIÓN 2.3.1: Dominio de dependencia

Sea una función $u(x, t)$ y un conjunto $[a, b] \times [c, d]$. El dominio de dependencia de u en (x_1, t_1) , es el conjunto de puntos de $[a, b]$ necesarios para dar la solución de u en (x_1, t_1) .

Como vemos, este concepto no solo estará definido para entornos discretos como lo que hemos estado tratando hasta ahora. Véase que para cada punto del mallado (x_j, t_r) tendremos un dominio de dependencia asociado.

Ya que en el capítulo 4, daremos solución a la ecuación de onda de forma numérica, necesitaremos que los puntos del mallado que usemos para aproximar la solución en cada (x_j, t_r) contengan al completo al dominio de dependencia. Vamos a ilustrarlo con las siguiente figuras, centrándonos en un ejemplo concreto.



2.3 Dominio de dependencia

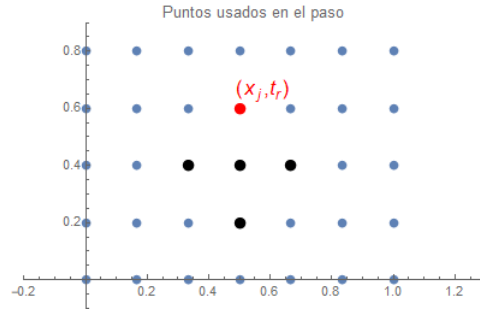


Figura 2.2: Ejemplo de los puntos que podríamos usar para aproximar la solución en (x_j, t_r) .

A su vez para aproximar la solución de cada uno de los puntos coloreados de negro, necesitaremos de los anteriores, por lo tanto:

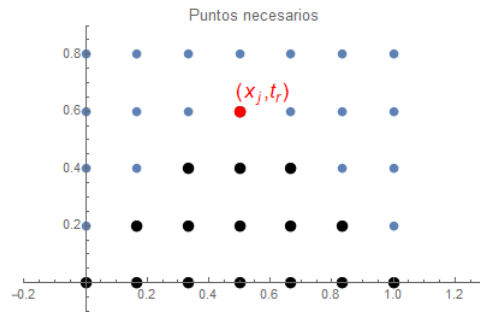


Figura 2.3: Todos los puntos que se usan para aproximar la solución de (x_j, t_r) .

Siendo el triángulo interior de las líneas coloreadas de negro el dominio de dependencia del problema, podrían darse dos casos:

Que el dominio de dependencia quedase en el interior de los puntos utilizados (Coloreados de negro en la Figura 2.3):

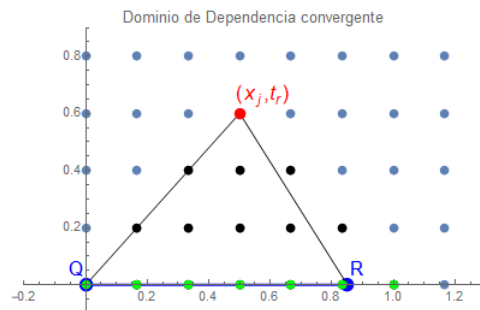


Figura 2.4: En este caso si hay un cambio en cualquier punto de la recta QR coloreada de azul, los puntos verdes lo captarían, al contener íntegramente a la misma.

2. CONCEPTOS PREVIOS

Que el dominio de dependencia quedase fuera de los puntos utilizados (Coloreados de negro en la Figura 2.3):

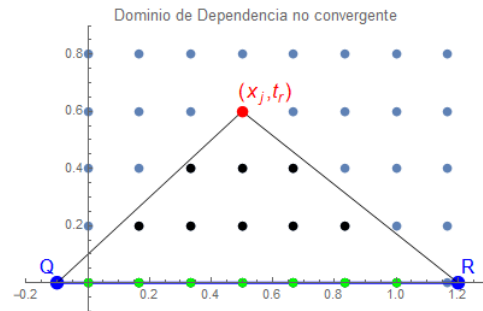


Figura 2.5: En este caso, si se da un cambio en recta azul fuera de los puntos verdes, estos no lo captarían y por lo tanto la aproximación de la solución no reflejaría el mismo.

Querremos estar siempre en la primera situación 2.4, para asegurar la convergencia del método de aproximación que utilizemos.

Definición de la ecuación de onda

Comenzaremos dando una introducción a la ecuación de onda en \mathbb{R}^n , con $n \geq 1$, siguiendo [2, 3]. Para simplificar la notación denotaremos a la función $u(x, t)$, como u , ahora la ecuación general de la ecuación de onda será,

$$u_{tt} - c^2 \Delta u = f, \quad \Omega \times [0, +\infty), \quad (3.1)$$

donde $\Omega \subset \mathbb{R}^n$, f es una función de valores reales, $c > 0$ es una constante de propagación y el operador

$$\Delta u = \sum_{j=1}^N \frac{\partial^2 u}{\partial x_j^2}, \quad (3.2)$$

es conocido como el laplaciano. En esta ecuación la variable t denota el tiempo transcurrido, x_i , $i = 1, \dots, n$ denotan las variables ~~espaciales~~ en cada dirección de \mathbb{R}^n y $u(x, t)$ será la posición la posición de la onda en cada punto del dominio espacio-temporal.

Podemos observar que se trata de una ecuación lineal y, siempre que $f \not\equiv 0$, la ecuación será no homogénea, es decir, tendrá termino independiente no nulo.

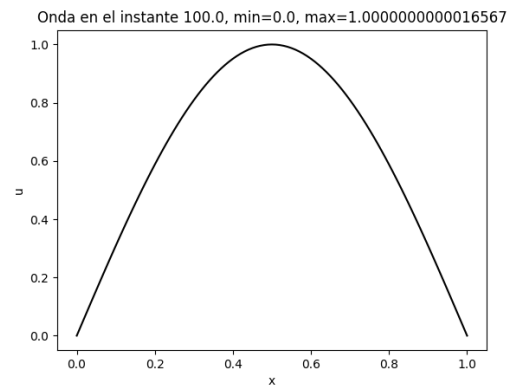
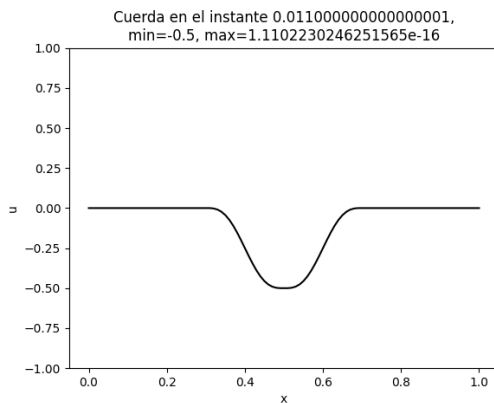
3. DEFINICIÓN DE LA ECUACIÓN DE ONDA

3.1 La ecuación de onda unidimensional

Ya que habiéndonos situado en la ecuación de onda, a partir de esta sección nos centraremos en el caso unidimensional, es decir en el que dominio $\Omega \subset \mathbb{R}^1$. Comenzaremos explicando el modelado de ~~dicha función~~, apoyándonos en [4, 5].

Supongamos que tenemos una cuerda uniforme y tensa, sujeta por los extremos, tal como lo estaría, por ejemplo, en una guitarra. Esta cuerda estaría colocada en nuestro esquema en el eje X entre 0 y L , siendo L el largo de la misma. Así vemos que cada punto de la cuerda se desplazará en el tiempo. Por lo tanto, la función que describa el movimiento de la cuerda será del tipo $u(x, t)$.

Nuestra cuerda en un instante t podría tener una forma parecida a alguna de las siguientes:



Supongamos que la cuerda no está en reposo, habrá una fuerza tensora T , producida por los puntos en los que está sujeta. Nos apoyaremos en la siguiente figura para ver como se describiría la ecuación:

3.1 La ecuación de onda unidimensional

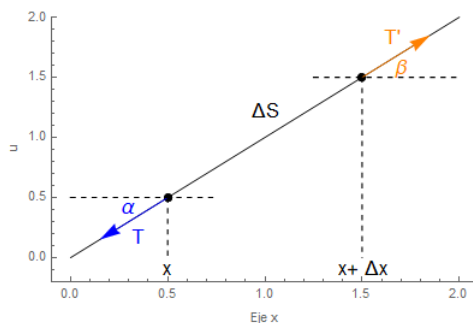


Figura 3.1: Representación de una sección diferencial de la cuerda y las fuerzas asociadas a la misma.

Elegido un segmento diferencial del arco de la cuerda ΔS , nos damos cuenta de que la cuerda se habrá desplazado perpendicularmente una cantidad u . Por lo tanto la fuerza tensora que produce cada punto en el que está fijado, no tendrán la misma dirección y sentido opuesto, sino que tendrán componente vertical. Las componentes verticales son:

$$T_y = -F_T \text{sen}(\alpha),$$

$$T'_y = F_T \text{sen}(\beta).$$

Luego la fuerza vertical quedaría $F = F_T(\text{sen}(\beta) - \text{sen}(\alpha))$, donde F_T es la fuerza generada por la tensión, por otro lado, al encontrarnos en una sección diferencial los ángulos son tan pequeños que podemos decir $\text{sen}(\alpha) \cong \tan(\alpha)$ y $\text{sen}(\beta) \cong \tan(\beta)$ quedando

$$F = F_T(\tan(\beta) - \tan(\alpha)).$$

Además $\tan(\alpha)$ y $\tan(\beta)$ son las pendientes de la curva en los puntos x y $x + \Delta x$ respectivamente, entonces podemos expresarlos como la primera derivada en los puntos correspondientes $\tan(\alpha) = u_x(x, t)$ y $\tan(\beta) = u_x(x + \Delta x, t)$. Obtenemos $F = F_T(u_x(x + \Delta x, t) - u_x(x, t))$.

Aplicando ahora la segunda Ley de Newton, la fuerza debe ser la masa por la aceleración. Luego será la masa $\rho_l \Delta S$, donde ρ_l es la densidad lineal de masa de la cuerda y como estamos en un ambiente diferencial podemos aproximar $\Delta S \cong \Delta x$, y la aceleración $u_{tt}(x, t)$. Reestructurando la ecuación queda:

3. DEFINICIÓN DE LA ECUACIÓN DE ONDA

$$u_{tt}(x, t) = \frac{F_T}{\rho l} \frac{u_x(x + \Delta x, t) - u_x(x, t)}{\Delta x}.$$

Tomando limite cuando $\Delta x \rightarrow 0$ y $c^2 = \frac{F_T}{\rho l}$ la ecuación queda de la siguiente forma:

$$u_{tt} = c^2 u_{xx}, \quad (0, L) \times [0, T]. \quad (3.3)$$

En este caso podemos tomamos el dominio como $\Omega^1 \subset \mathbb{R}$, donde T es un parámetro dado que representa el instante final de la observación. Añadiendo las condiciones de contorno usuales (valores conocidos de la solución sobre la frontera del dominio, $x = 0$ y $x = L$), para más información ver [6], ~~nuestro esquema nos queda.~~

$$u_{tt} - c^2 u_{xx} = 0, \quad (x, t) \in (0, L) \times (0, T), \quad (3.4)$$

$$u(x, 0) = f(x), \quad x \in (0, L), \quad (3.5)$$

$$u_t(x, 0) = g(x), \quad x \in (0, L), \quad (3.6)$$

$$u(0, t) = u(L, t) = 0, \quad t \in (0, T). \quad (3.7)$$

Estas última condiciones de contorno, (3.7), se llaman de tipo Dirichlet ~~homogenea~~. En lo que sigue, supondremos que f y g son funciones suficientemente regulares para que el problema anterior esté bien planteado y tenga solución, es decir, exista una función u dos veces derivable con respecto a t y x que verifica (3.4) a (3.7)

3.2 Clasificación de la ecuación de onda

En esta sección clasificaremos las EDP lineales, estas se clasifican en: elípticas, hiperbólicas y parabólicas. Para poder realizar la clasificación primero tendremos que estandarizarla. Aunque las ecuaciones se pueden clasificar y estandarizar siempre, nosotros por simplicidad lo haremos solo para las de segundo orden.

Una ecuación lineal de segundo orden en forma estándar sería como sigue:

$$Au_{tt} + Bu_{tx} + Cu_{xx} + Du_t + Eu_x + Fu = G, \quad (3.8)$$

donde A, B, C, D, E, F, G son funciones de variables x y t .

Clasificaremos (3.8) dependiendo de las ecuaciones características relativas a la ecuación en derivadas parciales (EDP):

DEFINICIÓN 3.2.1: Clasificación de las EDP de segundo orden

Si en todos los puntos (x, t) de una región $W \subset \mathbb{R}^2$ se cumple que:

- $B^2(x, t) - 4A(x, t)C(x, t) > 0$, entonces la EDP (3.8) se dice hiperbólica.
- $B^2(x, t) - 4A(x, t)C(x, t) = 0$, entonces la EDP (3.8) se dice parabólica.
- $B^2(x, t) - 4A(x, t)C(x, t) < 0$, entonces la EDP (3.8) se dice elíptica.

Vemos que siguiendo la notación de (3.3), $A = 1$, $C = -c^2$, $G = f$ y cada una de las demás funciones es 0. En el caso de la ecuación de onda, tenemos $B^2 - 4AC = 4c^2 > 0$ y por lo tanto nuestra ecuación es hiperbólica.

Este tipo de clasificación de las ecuaciones la usaremos para separar los distintos métodos numéricos que se pueden usar, ya que estos suelen diferir dependiendo de la misma.

3.3 Soluciones de tipo onda viajera

Una vez definida nuestra ecuación de onda, es interesante ver que esta admite dos soluciones de tipo onda viajera para cada uno de los dos sentidos del eje X . Para ello comenzaremos suponiendo que la solución $u(x, t)$ se puede descomponer en dos funciones de propagación, donde c sabemos que es la velocidad de la onda,

$$u(x, t) = p_+(x - ct) + p_-(x + ct), \quad (3.9)$$

aquí llamamos p_+ a la propagación hacia la derecha y p_- hacia la izquierda. Ahora siguiendo [7], veamos como sería la resolución del problema de valores iniciales, (3.4) a (3.7) teniendo en cuenta (3.9). Para ello necesitaremos suponer la derivabilidad de f .

Comenzamos utilizando (3.5), para $t = 0$ tendremos,

$$f(x) = u(x, 0) = p_+(x) + p_-(x), \quad (3.10)$$

derivando (3.10) en x ,

$$f'(x) = p_+'(x) + p_-'(x). \quad (3.11)$$

Ahora queremos sustituir el valor de $u_t(x, 0)$ en la condición (3.6), para ello derivemos antes (3.9) respecto a t ,

3. DEFINICIÓN DE LA ECUACIÓN DE ONDA

$$u_t = \frac{\partial}{\partial t}(p_+(x - ct) + p_-(x + ct)) = cp'_+(x - ct) - cp'_-(x + ct), \quad (3.12)$$

entonces,

$$g(x) = u_t(x, 0) = -cp'_+(x) + cp'_-(x), \quad (3.13)$$

dividiendo la igualdad por c ,

$$\frac{1}{c}g(x) = -p'_+(x) + p'_-(x). \quad (3.14)$$

Haciendo (3.10) + (3.13) y (3.10) - (3.13):

$$f'(x) + \frac{1}{c}g(x) = 2p'_-(x), \quad \text{y} \quad f'(x) - \frac{1}{c}g(x) = 2p'_+(x).$$

Luego despejando,

$$p'_-(x) = \frac{1}{2}f'(x) + \frac{1}{2c}g(x), \quad \text{y} \quad p'_+(x) = \frac{1}{2}f'(x) - \frac{1}{2c}g(x).$$

Como todo este proceso podría hacerse para todo $x \in \mathbb{R}$, integrando respecto a x y sustituyéndolo por $x + ct$ y $x - ct$, respectivamente, obtendremos las representaciones siguientes de p_- y p_+ .

$$p_-(x + ct) = \frac{1}{2}f(x + ct) + \frac{1}{2c} \int_0^{x+ct} g(s)ds + A, \quad (3.15)$$

$$p_+(x + ct) = \frac{1}{2}f(x - ct) - \frac{1}{2c} \int_0^{x-ct} g(s)ds + B, \quad (3.16)$$

donde A, B son constantes que tomaremos como 0 al saber que $A + B = 0$ gracias a 3.9. Nuestras integrales comienzan en 0 por el dominio de 3.4.

Finalmente llegamos a que la solución del problema de valores iniciales, será:

$$u(x, t) = \frac{1}{2} (f(x + ct) + f(x - ct)) + \frac{1}{2c} \int_{x-ct}^{x+ct} g(s)ds. \quad \text{🗨️}$$

3.4 La energía en la ecuación de onda

En esta sección nos centraremos en analizar la ley de energía en la ecuación de onda. Para poder obtener la energía de la ecuación haremos para todo $t \in [0, T]$ el producto interior 2.2.1 de la misma con u_t :

$$\langle u_t(\cdot, t), u_{tt}(\cdot, t) \rangle_{\Omega} = c^2 \langle u_t(\cdot, t), u_{xx}(\cdot, t) \rangle_{\Omega}.$$

Para facilitar el entendimiento de la sección simplificaremos la notación de la siguiente forma:

$$\langle u_t, u_{tt} \rangle_{\Omega} = c^2 \langle u_t, u_{xx} \rangle_{\Omega}. \quad (3.17)$$

Cada uno de los términos de la energía tendremos que ponerlo de la forma $\frac{\partial}{\partial t}(\cdot)$, comenzaremos por el término de la izquierda. Para desarrollarlo necesitaremos la siguiente igualdad, derivada de la integración por partes,

$$\langle u_t, u_{tt} \rangle_{\Omega} = \int_{\Omega} u_t u_{tt} dx = u_t u_t|_{\Omega} - \int_{\Omega} u_{tt} u_t dx = u_t u_t|_{\Omega} - \langle u_{tt}, u_t \rangle_{\Omega}. \quad (3.18)$$

Por otro lado necesitaremos ver como se puede descomponer $\frac{\partial}{\partial t} \|u_t\|_S^2$, utilizando la definición de norma 2.2.1,

$$\frac{\partial}{\partial t} \|u_t\|_{\Omega}^2 = \frac{\partial}{\partial t} \int_{\Omega} u_t u_t dx.$$

Ya que, $\frac{\partial}{\partial t}$ independiente de x , y suponiendo que se dan en u y en u_t condiciones de continuidad adecuadas, podremos permutar de sitio la integral y la derivada, y por tanto,

$$\frac{\partial}{\partial t} \|u_t\|_{\Omega}^2 = \int_{\Omega} \frac{\partial}{\partial t} u_t u_t dx,$$

derivando, usando la regla de la cadena, y separando en dos integrales obtenemos la igualdad

$$\begin{aligned} \frac{\partial}{\partial t} \|u_t\|_{\Omega}^2 &= \int_{\Omega} u_{tt} u_t + u_t u_{tt} dx \\ &= \int_{\Omega} u_{tt} u_t dx + \int_S u_t u_{tt} dx \\ &= \langle u_{tt}, u_t \rangle_{\Omega} + \langle u_t, u_{tt} \rangle_{\Omega}. \end{aligned} \quad (3.19)$$

3. DEFINICIÓN DE LA ECUACIÓN DE ONDA

Continuando ahora desde nuestro factor inicial $\langle u_t, u_{tt} \rangle_\Omega$, usando la igualdad (3.19),

$$\langle u_t, u_{tt} \rangle_\Omega = \int_\Omega u_t u_{tt} dx = \frac{\partial}{\partial t} \|u_t\|_\Omega^2 - \int_\Omega u_{tt} u_t dx.$$

Utilizando (3.18) en el último sumando,

$$\langle u_t, u_{tt} \rangle_\Omega = \frac{\partial}{\partial t} \|u_t\|_\Omega^2 - \left(u_t u_t|_\Omega - \int_\Omega u_t u_{tt} dx \right).$$

Agrupando los términos tendremos,

$$\begin{aligned} 2 \langle u_t, u_{tt} \rangle_\Omega &= \frac{\partial}{\partial t} \|u_t\|_\Omega^2 - u_t u_t|_\Omega, \\ \langle u_t, u_{tt} \rangle_\Omega &= \frac{1}{2} \frac{\partial}{\partial t} \|u_t\|_\Omega^2 - \frac{1}{2} u_t u_t|_\Omega. \end{aligned} \quad (3.20)$$

Ahora nos centraremos en el termino de la derecha, comenzamos utilizando la igualdad (3.18), y pasamos el producto interior hacia la izquierda en (3.17), quedando,

$$\langle u_t, u_{tt} \rangle_\Omega - c^2 \langle u_{tx}, u_x \rangle_\Omega = u_t u_x|_\Omega \quad (3.21)$$

Siguiendo con el análisis de $\langle u_{tx}, u_x \rangle_\Omega$, análogamente a como lo hicimos con $\langle u_t, u_{tt} \rangle_\Omega$, obtendremos

$$\langle u_{tx}, u_x \rangle_\Omega = \frac{1}{2} \frac{\partial}{\partial t} \int_{\mathbb{R}} u_x u_x dx - \frac{1}{2} u_x u_x|_\Omega = \frac{1}{2} \frac{\partial}{\partial t} \|u_x\|_\Omega^2 - \frac{1}{2} u_x u_x|_\Omega. \quad (3.22)$$

Introduciendo (3.20) y (3.22) en la ecuación (3.21) queda:

$$\frac{\partial}{\partial t} \left(\frac{1}{2} \|u_t\|_\Omega^2 + \frac{c^2}{2} \|u_x\|_\Omega^2 \right) = c^2 u_t u_x|_\Omega - \frac{c^2}{2} u_x u_x|_\Omega - \frac{1}{2} u_t u_t|_\Omega.$$

Definiremos $\mathfrak{B} \triangleq c^2 u_t u_x|_\Omega + \frac{c^2}{2} u_x u_x|_\Omega + \frac{1}{2} u_t u_t|_\Omega$.

Ahora podremos separar en energía cinética $K = \frac{1}{2} \|u_t\|_\Omega^2$ y la energía potencial $P = \frac{c^2}{2} \|u_x\|_\Omega^2$ de donde se sigue que la energía total será

$$\mathfrak{H} = K + P = \frac{1}{2} \|u_t\|_\Omega^2 + \frac{\mu^2}{2} \|u_x\|_\Omega^2.$$

Por otro lado, tenemos que $\frac{\partial}{\partial t} \mathfrak{H} = \mathfrak{B}$, para que se diese conservación de la energía tendrá que ser $\mathfrak{B} = 0$. Desarrollándolo,

$$0 = \mathfrak{B} = c^2(u_t(l, t)u_x(l, t) - u_t(0, t)u_x(0, t)) + \frac{c^2}{2}(u_x(l, t)u_x(l, t) - u_x(0, t)u_x(0, t)) \\ + \frac{1}{2}(u_t(l, t)u_t(l, t) - u_t(0, t)u_t(0, t)).$$

Vemos que esto nos impone ciertas restricciones a nuestra ecuación, estas se traducirán a condiciones de contorno en nuestro esquema.

Resolución numérica de la ecuación de onda

4.1 Métodos principales de resolución

Las ecuaciones en derivadas parciales, en general, no se pueden resolver con métodos exactos, como separación de variables, transformada de Fourier, etc. Con la excepción de ecuaciones muy sencillas sobre dominios espaciales no acotados o muy estructurados, como intervalos, rectángulos o esferas, que en la práctica pueden no resultar útiles para capturar todos los matices de la realidad. Esto nos lleva a la creación de métodos numéricos que consiguen resolverlas de una manera aproximada, pero con errores tan pequeños como sea preciso siempre que podamos asumir el coste computacional.

En este apartado nos centraremos en el método de diferencias finitas, y veremos las características principales que nos atañen del mismo. Más tarde, en el capítulo 5, implementaremos este método en Python.

4.2 Método de diferencias finitas

Este método consiste principalmente en el uso de cocientes incrementales para la aproximación en tiempo y en espacio de nuestro problema de contorno con valores iniciales para

4. RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE ONDA

la ecuación de onda (3.4),(3.5),(3.6),(3.7). Vamos a crear un mallado tal y como en 2.2.3, para hacer una discretización del problema.

$$x_j = jh \quad \text{para} \quad j = 1, \dots, m,$$

$$t_r = rk \quad \text{para} \quad r = 0, 1, \dots, n.$$

En este caso elegimos un entero m y el tamaño de paso de tiempo correspondiente $k > 0$. El tamaño de paso del espacio será $h = \frac{l}{m}$ y con esto haremos un mallado del tipo (x_j, t_r) .

Vemos que los puntos (x_j, t_r) del interior del dominio espacio temporal podrían escribirse como:

$$u_{tt}(x_j, t_r) = c^2 u_{xx}(x_j, t_r). \quad (4.1)$$

En lo que sigue, nos propondremos calcular un conjunto de valores, denotados $u_{j,r} \in \mathbb{R}$ que verifiquen de forma aproximada la igualdad anterior, $u(x_j, t_r) \approx u_{j,r}$.

Para ello el método se construirá optando por un esquema del tipo explícito en espacio, utilizando una aproximación centrada tanto en espacio como en tiempo.

$$u_{xx}(x_j, t_r) \approx \frac{u_{j+1,r} - 2u_{j,r} + u_{j-1,r}}{h^2}, \quad (4.2)$$

$$u_{tt}(x_j, t_r) \approx \frac{u_{j,r+1} - 2u_{j,r} + u_{j,r-1}}{k^2}. \quad (4.3)$$

Ahora sustituyendo las expresiones (4.2),(4.3) en la ecuación de onda (4.1) nos queda el esquema.

$$\frac{u_{j,r+1} - 2u_{j,r} + u_{j,r-1}}{k^2} = c^2 \frac{u_{j+1,r} - 2u_{j,r} + u_{j-1,r}}{h^2}. \quad (4.4)$$

Es usual definir $\mu = \frac{ck}{h}$, con lo que se tendría:

$$u_{j,r+1} - 2u_{j,r} + u_{j,r-1} = \mu^2 u_{j+1,r} - 2\mu^2 u_{j,r} + \mu^2 u_{j-1,r},$$

$$\text{para } r = 1, 2, \dots, n, \quad j = 1, 2, \dots, m-1.$$

Despejando la aproximación con mayor nivel en tiempo, obtenemos:

$$u_{j,r+1} = \mu^2(u_{j+1,r} + u_{j-1,r}) + 2(1 - \mu^2)u_{j,r} - u_{j,r-1}. \quad (4.5)$$

Esta expresión define el método que utilizaremos, dejando a un lado las condiciones de contorno (3.5),(3.6) y (3.7).

A continuación nos centraremos en dichas condiciones, siguiendo [7]. En primer lugar la condición inicial (3.5) nos dará los valores de los nodos

$$u_{j,0} = f(x_j), \quad \text{para } j = 0, 1, \dots, m.$$

Por otra parte, las condiciones de tipo Dirichlet (3.7) quedarán como $u_{0,r} = u_{m,r} = 0$ para $r = 0, 1, \dots, n$.

Para la condición que falta 3.6 utilizaremos, haciendo una aproximación por cociente incremental en los dos primeros instantes, $u_{j,1} = u_{j,0} + kg(x_j)$, que servirá ya sean las f y g derivables o no. Para esta podría haber muchas mas opciones. En nuestro caso no ahondaremos mas en este tema y utilizaremos la expresión mas sencilla descrita anteriormente.

Nuestro esquema al completo quedaría:

$$u_{j,r+1} = \mu^2(u_{j+1,r} + u_{j-1,r}) + 2(1 - \mu^2)u_{j,r} - u_{j,r-1}, \quad (4.6)$$

$$u_{j,0} = f(x_j), \quad j = 0, 1, \dots, m, \quad (4.7)$$

$$u_{j,1} = u_{j,0} + kg(x_j), \quad j = 0, 1, \dots, m, \quad (4.8)$$

$$u_{0,r} = u_{m,r} = 0, \quad r = 0, 1, \dots, n. \quad (4.9)$$

4.3 Consistencia

Para poder comprobar el orden del esquema que hemos elegido (4.5) vamos a hacer un análisis de la consistencia, que comenzaremos dividiendo la ecuación (4.4) de la siguiente forma,

$$\underbrace{\frac{u(j, r+1) - 2u(j, r) + u(j, r-1))}{k^2}}_{(1)} - c^2 \underbrace{\frac{u(j+1, r) - 2u(j, r) + u(j-1, r))}{h^2}}_{(2)} = 0.$$

4. RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE ONDA

Haremos ahora el desarrollo de Taylor para cada una de las componentes anteriores, denotando para las derivadas a partir de orden dos $u_{xxx} = u_{3x}$ y $u_{ttt} = u_{3t}$ respectivamente. Suponiendo que nuestra solución u sea suficientemente regular, es decir, que se pueda derivar tantas veces como haga falta obtendremos.

$$\begin{aligned} u(j+1, r) &= u + hu_x + h^2 \frac{u_{xx}}{2} + h^3 \frac{u_{3x}}{3!} + h^4 \frac{u_{4x}}{4!} + h^5 \frac{u_{5x}}{5!} + O(h^6), \\ u(j-1, r) &= u - hu_x + h^2 \frac{u_{xx}}{2} - h^3 \frac{u_{3x}}{3!} + h^4 \frac{u_{4x}}{4!} - h^5 \frac{u_{5x}}{5!} + O(h^6), \\ u(j, r+1) &= u + ku_t + k^2 \frac{u_{tt}}{2} + k^3 \frac{u_{3t}}{3!} + k^4 \frac{u_{4t}}{4!} + k^5 \frac{u_{5t}}{5!} + O(k^6), \\ u(j, r-1) &= u - ku_t + k^2 \frac{u_{tt}}{2} - k^3 \frac{u_{3t}}{3!} + k^4 \frac{u_{4t}}{4!} - k^5 \frac{u_{5t}}{5!} + O(k^6), \\ u(j, r) &= u. \end{aligned}$$

Sustituyendo en (1) y (2), obtendremos:

$$\begin{aligned} (1) &= u_{tt} + 2 \frac{k^2 u_{4t}}{4!} + 2 \frac{k^4 u_{6t}}{6!} + O(h^6), \\ (2) &= u_{xx} + 2 \frac{h^2 u_{4x}}{4!} + 2 \frac{h^4 u_{6x}}{6!} + O(k^6). \end{aligned}$$

Sustituyendo en nuestro esquema nos quedará:

$$\begin{aligned} T_{h,k}u &:= u_{tt} + 2 \frac{k^2 u_{4t}}{4!} + 2 \frac{k^4 u_{6t}}{6!} - c^2 u_{xx} - 2c^2 \frac{h^2 u_{4x}}{4!} - 2c^2 \frac{h^4 u_{6x}}{6!} \dots = \\ &= u_{tt} + 2 \frac{k^2 u_{4t}}{4!} - c^2 u_{xx} - 2c^2 \frac{h^2 u_{4x}}{4!} + O(k^4) + O(h^4). \end{aligned}$$

Como estamos evaluando la solución exacta en cada punto tendremos $u_{tt} = c^2 u_{xx}$, entonces

$$\begin{aligned} u_{ttt} &= c^2 u_{xxt}, \\ u_{tttt} &= c^2 u_{xxtt}. \end{aligned}$$

Por otro lado,

$$\begin{aligned} c^2 u_{xxx} &= u_{ttt}, \\ c^2 u_{xxx} &= u_{tttt}. \end{aligned}$$

De aquí deducimos, $u_{4x} = \frac{u_{4t}}{c^4}$ y por tanto:

$$\begin{aligned}
T_{h,k}u &= u_{tt} + 2\frac{k^2 u_{4t}}{4!} - c^2 u_{xx} - 2c^2 \frac{h^2 u_{4x}}{4!} + O(k^4) + O(h^4) = \\
&= 2\frac{k^2 u_{4t}}{4!} - 2c^2 \frac{h^2 u_{4t}}{4!c^4} + O(k^4) + O(h^4) = \\
&= \frac{2}{4!}(k^2 - \frac{h^2}{c^2})u_{4t} + O(k^4) + O(h^4) = \\
&= O(k^2) + O(h^2).
\end{aligned}$$

Así hemos comprobado que el esquema tiene orden de consistencia dos en espacio y en tiempo.

4.4 Análisis de Von Neumann

En la utilización de métodos multipaso, como el esquema que nos centramos en este trabajo, los errores globales finalmente observados pueden ser mucho mayores de lo que podría esperarse a partir del orden de consistencia local estudiado anteriormente. Esto es debido a que en el error global interviene un aumento del error que se propaga y que, a menudo, incluso se incrementa al disminuir la longitud de paso. Puede estar relacionado con varios factores como el método de inicialización, el truncamiento local y errores de redondeo. Esto se conoce como **inestabilidad**.

En esta sección nos centraremos en el análisis de Estabilidad de Fourier o Von Neuman. Este aunque no garantizará la estabilidad total del esquema, suele dar muy buenos resultados en la práctica, y es el más empleado.

Podemos encontrar diversas notaciones y formas de hacer este análisis en libros clásicos y modernos, véase [8, 9].

Comenzamos relacionando cada solución numérica del esquema con la solución exacta y el error del mismo, de la forma:

$$u_{j,r} = u(x_j, t_r) + \varepsilon_{j,r} \quad (4.10)$$

donde $\varepsilon_{j,r}$ denotará el error en cada uno de los puntos del retículo que hemos formado.

4. RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE ONDA

Ahora vamos a aplicarlo a la ecuación de onda, para profundizar en la estabilidad del esquema (4.6). Haciendo uso del termino del error como hemos visto en (4.10), lo sustituimos en el esquema, esta vez utilizando (4.4).

$$\left(\frac{u(x_j, t_{r+1}) - 2u(x_j, t_r) + u(x_j, t_{r-1}))}{k^2} - c^2 \frac{u(x_{j+1}, t_r) - 2u(x_j, t_r) + u(x_{j-1}, t_r)}{h^2} \right) + \left(\frac{\varepsilon_{j,r+1} - 2\varepsilon_{j,r} + \varepsilon_{j,r-1}}{k^2} - c^2 \frac{\varepsilon_{j+1,r} - 2\varepsilon_{j,r} + \varepsilon_{j-1,r}}{h^2} \right) = 0.$$

Como cualquier esquema numérico lineal se satisface de manera exacta por $u(x_j, t_r)$, los errores también serán soluciones de la ecuación discretizada, obteniendo.

$$\frac{\varepsilon_{j,r+1} - 2\varepsilon_{j,r} + \varepsilon_{j,r-1}}{k^2} - c^2 \frac{\varepsilon_{j+1,r} - 2\varepsilon_{j,r} + \varepsilon_{j-1,r}}{h^2} = 0.$$

Tomando la solución de tipo onda $\varepsilon_{j,r} = g^r e^{ij\beta h}$ donde i es la unidad imaginaria, g denota el factor de amplitud y β se corresponderá con el número de onda de la ecuación de Fourier. Vamos a sustituirlo en la ecuación anterior, quedando:

$$g^{r+1} e^{ij\beta h} = \mu^2 (g^r e^{i(j+1)\beta h} + g^r e^{i(j-1)\beta h}) + 2(1 - \mu^2) g^r e^{ij\beta h} - g^{r-1} e^{ij\beta h}.$$

Dividiendo entre $g^{r-1} e^{ij\beta h}$:

$$\begin{aligned} g^2 &= \mu^2 (g e^{i\beta h} + g e^{-i\beta h}) + 2(1 - \mu^2) g - 1, \\ 0 &= g^2 - \mu^2 (g e^{i\beta h} + g e^{-i\beta h}) - 2(1 - \mu^2) g + 1. \end{aligned}$$

Usando las identidades trigonométricas $e^{ix} + e^{-ix} = 2\cos(x)$ y $1 - \cos(x) = 2\sin^2(\frac{x}{2})$ obtenemos,

$$\begin{aligned} 0 &= g^2 - 2\mu^2 \cos(\beta h) g - 2(1 - \mu^2) g + 1, \\ &= g^2 - 2\mu^2 \cos(\beta h) g + 2\mu^2 g - 2g + 1, \\ &= g^2 + 2g(1 + \mu^2(1 - \cos(\beta h))) + 1, \\ &= g^2 + 2 \left(2\mu^2 \sin^2 \left(\frac{\beta h}{2} \right) - 1 \right) g + 1. \end{aligned}$$

Podemos ahora tratarlo como una ecuación de segundo grado, es decir, tomando $A = \left(1 - 2\mu^2 \sin^2 \left(\frac{\beta h}{2}\right)\right)$ vemos que nos quedaría,

$$\begin{aligned} g &= \frac{2A \pm \sqrt{(-2A)^2 - 4}}{2} = A \pm \sqrt{A^2 - 1} = \\ &= 1 - 2\mu^2 \sin^2 \left(\frac{\beta h}{2}\right) \pm \sqrt{\left(1 - 2\mu^2 \sin^2 \left(\frac{\beta h}{2}\right)\right)^2 - 1}. \end{aligned}$$

Al ser g el factor de ampliación, es una relación del espacio y el tiempo y si se alejase demasiado de 0, las perturbaciones se amplificarán con el tiempo. Por lo tanto necesitaremos $|g| \leq 1$, es decir, vamos a estudiar una condición suficiente (llamada condición CFL) para que

$$\left|A \pm \sqrt{A^2 - 1}\right| \leq 1.$$

Vamos a ver que pasará dependiendo de los valores posibles de A . Como $2\mu^2 \sin^2 \left(\frac{\beta h}{2}\right) \geq 0$ tendremos que $A \leq 1$.

- Si $A < -1$:

La solución $g = A - \sqrt{A^2 - 1} < -1$ y por lo tanto siempre se da $|g| > 1$ y el esquema será en cualquier caso inestable.

- Si $-1 \leq A \leq 1$:

Definimos $s^2 \triangleq \sin^2 \left(\frac{\beta h}{2}\right)$, con esto tenemos

$$0 \leq s^2 \leq 1,$$

luego,

$$\begin{aligned} 0 &\leq 2\mu^2 s^2 \leq 2\mu^2, \\ 0 &\geq -2\mu^2 s^2 \geq -2\mu^2, \\ 1 &\geq 1 - 2\mu^2 s^2 \leq \underbrace{1 - 2\mu^2}_{(1)} \geq -1. \end{aligned}$$

Veamos cual es la condición necesaria para que se de (1),

$$1 - 2\mu^2 \geq -1,$$

$$2\mu^2 \leq 2,$$

4. RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE ONDA

$$\mu \leq 1, \quad (4.11)$$

llegamos a que para que el esquema sea estable necesitamos que se de esta última condición $\mu \leq 1$, la condición CFL.

Por lo tanto nuestro esquema será condicionalmente estable. En este caso, tal y como podemos extraer de [3], mirando el dominio de dependencia (2.3.1) nos damos cuenta de que, cuando se de la igualdad a uno, este coincidirá con los puntos de nuestra discretización, es decir, dependerá del menor número de puntos posibles, mientras cumple la condición CFL.

En la comparativa del capítulo siguiente cambiaremos μ para ilustrar como afecta de manera muy positiva a la estabilidad de las soluciones el análisis anteriormente realizado.

4.5 Resolución numérica para las soluciones de tipo viajero

Tal y como comentamos en la sección 3.3 de la ecuación de onda se pueden obtener dos soluciones de tipo onda viajera (3.9). Ahora nos centraremos en implementar numéricamente dichas soluciones.

Querremos ver dichas soluciones en cada uno de nuestro puntos del mallado, para ello llamaremos a las aproximaciones de las mismas en cada uno de los puntos como

$$\begin{aligned} w^+(x_j, t_r) &= p_+(x_j - ct_r), \\ w^-(x_j, t_r) &= p_-(x_j + ct_r). \end{aligned}$$

Ahora tendremos que la aproximación de la solución de u será,

$$u_{j,r} = w_{j,r}^+ + w_{j,r}^-. \quad (4.12)$$

Partiendo de las representaciones de las soluciones (3.15) y (3.16), aproximaremos $w_{j,r}^+$ y $w_{j,r}^-$. Esto lo haremos primeramente separando en dos la integral, trabajemos con $p_-(x_j + ct_r)$:

$$\begin{aligned} p_-(x_j + ct_r) &= \frac{1}{2}f(x_j + ct_r) + \frac{1}{2c} \int_0^{x_j + ct_r} g(s) \partial s = \\ &= \frac{1}{2}f(x_j + ct_r) + \frac{1}{2c} \int_0^{x_j} g(s) \partial s + \frac{1}{2c} \int_{x_j}^{x_j + ct_r} g(s) \partial s. \end{aligned}$$

Realizamos una aproximación de cada una de las integrales utilizando el método del trapecio y los puntos ya definidos en el mallado. El tamaño de paso en la primera integral será h , y en la segunda será ck , por tanto la aproximación es.

$$w_{j,r}^- = \frac{1}{2}f(x_j + ct_r) + \frac{h}{4c} \left(g(x_0) + 2 \sum_{i=1}^{j-1} g(x_i) + g(x_j) \right) + \frac{ck}{4c} \left(g(x_j + ct_0) + 2 \sum_{i=1}^{r-1} g(x_j + ct_i) + g(x_j + ct_r) \right).$$

Por otro lado trabajando con $p_+(x_j + ct_r)$, por como hemos definido las aproximaciones (4.12) podemos simplemente utilizar:

$$w_{j,r}^+ = u_{j,r} - w_{j,r}^-$$

Con esto ya podríamos implementar cada una de las soluciones, para ver como se comportan las mismas a comparación de u .

4.6 Ley de energía discreta

En este punto nos preguntamos como se traducirían esta ley de energía a nuestro esquema. En este caso al ser el esquema discreto, tendremos que usar la definición de suma por partes 2.2.1, en lugar de la del producto interior anteriormente utilizada en la sección 3.4.

Vamos a comenzar tomando como dominio mallado 2.2.3, sabiendo que queremos hacer la suma por partes en el eje de la posición definimos D los puntos del mismo. Veremos cuales son las condiciones que deberán cumplir nuestro esquema, utilizando la función δ definida en 2.2.5, será.

$$(\delta_t u, \delta_{tt} u)'_D = c^2 \langle \delta_t u, \delta_{xx} u \rangle_{L^2(D)}.$$

Siguiendo la notación que tomabamos en 2.2.1, para cada $r = 1, \dots, n$ tendremos:

$$(\delta_t u_{\cdot,r}, \delta_{tt} u_{\cdot,r})_{L^2(D)} = c^2 (\delta_t u_{\cdot,r}, \delta_{xx} u_{\cdot,r})_{L^2(D)}. \quad (4.13)$$

Ahora para desarrollar la parte derecha de la igualdad usaremos la observación 2.2.6 y posteriormente la definición 2.2.1.

4. RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE ONDA

$$\begin{aligned} (\delta_t u_{\cdot,r}, \delta_{xx} u_{\cdot,r})_{L^2(D)} &\stackrel{2,2,6}{=} (\delta_t u_{\cdot,r}, \delta_{x+x^-} u_{\cdot,r})_{L^2(D)}, \\ &\stackrel{(2.5)}{=} -(\delta_{x^+} \delta_t u_{\cdot,r}, \delta_{x^+} u_{\cdot,r})_{L^2(D)} - (\delta_t u_{0,r})(\delta_{x^-} u_{-1,r}) - (\delta_t u_{m+1,r})(\delta_{x^-} u_{m,r}). \end{aligned} \quad (4.14)$$

Insertando esto en la ecuación (4.13) nos quedaría:

$$(\delta_t u_{\cdot,r}, \delta_{tt} u_{\cdot,r})_{L^2(D)} + c^2 (\delta_t u_{\cdot,r}, \delta_{x^+} \delta_t u_{\cdot,r})_{L^2(D)} = -c^2 (\delta_t u_{0,r})(\delta_{x^-} u_{-1,r}) + c^2 (\delta_t u_{m+1,r})(\delta_{x^-} u_{m,r}).$$

Denotando al termino de la derecha como \mathfrak{b} tendremos que

$$\delta_{t^+} \mathfrak{h} = \mathfrak{b}.$$

Identificamos los términos, de la energía total \mathfrak{h} , energía cinética \mathfrak{t} y energía potencial \mathfrak{v} :

$$\mathfrak{h} = \mathfrak{t} + \mathfrak{v}, \quad \mathfrak{t} = \frac{1}{2} \|\delta_{t^-} u_{\cdot,r}\|_{L^2(D)}^2 \quad \text{y} \quad \mathfrak{v} = \frac{\mu^2}{2} (\delta_{x^+} u_{\cdot,r}, e_{t^-} \delta_{x^+} u_{\cdot,r})_{L^2(D)}.$$

Y por lo tanto se dará la conservación de energía cuando $\mathfrak{b} = 0$, es decir cuando

$$(\delta_t u_{0,r})(\delta_{x^-} u_{-1,r}) = (\delta_t u_{m+1,r})(\delta_{x^-} u_{m,r}). \quad (4.15)$$

Esto se puede dar de muchas maneras, pero las mas sencillas y estudiadas son que ambas sean cero. Para el lado izquierdo de (4.15) se traduciría en:

$$u_{0,r} = 0, \quad r = 0, 1, \dots, n, \quad (4.16)$$

o

$$\delta_{x^-} u_{0,r} \quad r = 0, 1, \dots, n. \quad (4.17)$$

Vemos que 4.16 se corresponde con la condición de Dirichlet en el borde izquierdo del mallado, esta será la misma condición que dábamos en 4.9. De la segunda, 4.17, de la definición de 2.2.5 sabemos, $\delta_{x^-} u_{0,r} = \frac{1}{h} (u_{0,r} - u_{-1,r})$ y por lo tanto $u_{j,0} = u_{j,-1}$ para $j = 0, 1, \dots, m$.

En nuestro caso $j = -1$ no se encuentra en el esquema, pero gracias a la igualdad que acabamos de obtener podremos inicializar los puntos de la primera posición en espacio $j = 0$ con nuestra expresión 4.5, quedando:

$$\begin{aligned} u_{0,r+1} &= \mu^2(u_{1,r} + u_{-1,r}) + 2(1 - \mu^2)u_{0,r} - u_{0,r-1} \\ &= \mu^2(u_{1,r} + u_{0,r}) + 2(1 - \mu^2)u_{0,r} - u_{0,r-1} \\ &= \mu^2 u_{1,r} + (2 - \mu^2)u_{0,r} - u_{0,r-1} \end{aligned}$$

Miramos ahora el lado derecho de la igualdad (4.15) e igualando a cero tendremos

$$u_{m+1,r} = 0, \quad r = 0, 1, \dots, n, \quad (4.18)$$

o

$$\delta_{x-u_{m,r}} \quad r = 0, 1, \dots, n, \quad (4.19)$$

(AQUÍ EL PROBLEMA ES LA INTERPRETACIÓN DE ESTO. LA PRIMERA CONDICIÓN SE REFIERE A UNA RECTA DE PUNTOS VIRTUALES (queda $u_{m+1,r} = 0$) Y LA SEGUNDA UNOS PUNTOS QUE YA TENEMOS DEFINIDOS POR NUESTRO ESQUEMA(4.6) ($u_{m,r} = u_{m-1,r}$ LO QUE AL METERLO EN EL ESQUEMA (4.5),ME QUEDA $u_{m,r+1} = (2 - \mu^2)u_{m,r} + \mu^2 u_{m+1,r} - u_{m,r-1}$) Y AL TENER NODOS VIRTUALES TAMPOCO SE COMO SE COMO CONTINUAR)

En el capítulo siguiente modificaremos estas expresiones obtenidas, para ver que pasa cuando no se da conservación de la energía.

Modelado en Python y comparativa

En este capítulo nos centraremos en ver el modelado en Python de nuestro esquema resuelto por el método de diferencias finitas, (4.6). Veremos como varía la solución en varios instantes de tiempo, la forma que presenta la onda cambiando las condiciones iniciales (4.7) y (4.8), y haremos una comparativa de como afectan las condiciones de estabilidad (4.11), y de conservación de la energía (4.15).

5.1 Soluciones

En esta sección comenzaremos señalando varias soluciones dependiendo de las condiciones iniciales. En toda la sección establecemos $c = 440$.

5.1.1 $f(x) = \sin(\pi x)$, $g(x) = 0$

Cuando la posición inicial depende del seno y la velocidad inicial es 0, se puede observar como nuestra solución oscila sin llegar a parar, al no tener en cuenta nuestro sistema (4.6), fuerzas externas como el rozamiento.

5. MODELADO EN PYTHON Y COMPARATIVA

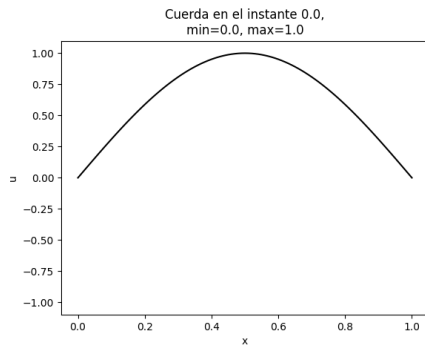


Figura 5.1: Posición inicial de la cuerda.

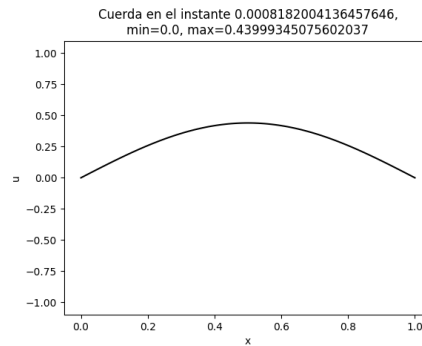


Figura 5.2: La cuerda comienza a bajar.

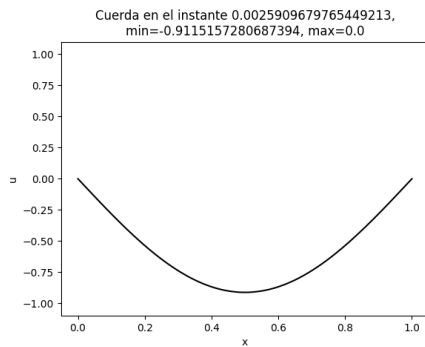


Figura 5.3: Sigue bajando hasta llegar a la posición de máxima amplitud.

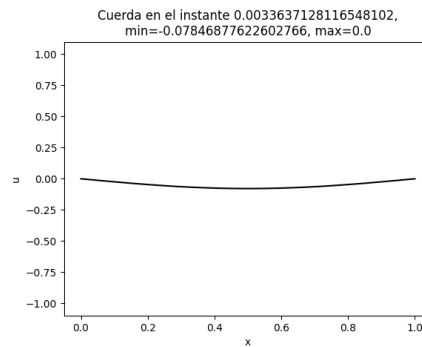


Figura 5.4: La cuerda comienza a subir de nuevo.

Este tipo de movimiento es muy regular, podemos observarlo, por ejemplo, en el movimiento del punto el punto $x = 0,5$ respecto al tiempo.

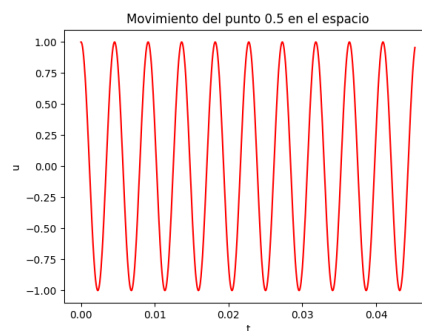


Figura 5.5: Se puede escuchar este sonido en el archivo figura5-5.wav adjunto.

5.1.2 $f(x)$ tipo Struck, $g(x) = 0$

Cuando la posición inicial es de tipo Struck, es decir, un pico en la zona central de la cuerda, y la velocidad inicial es 0, la solución no es tan suave, al ser f no derivable.

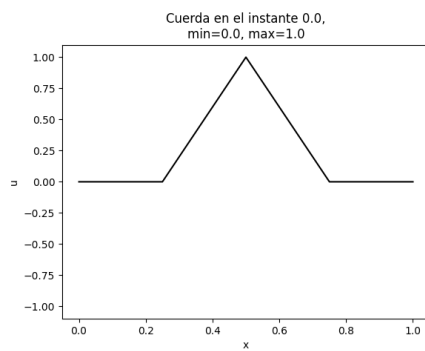


Figura 5.6: Posición inicial de la cuerda.

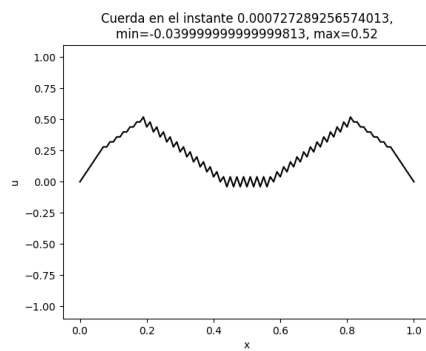


Figura 5.7: Se crean picos en la cuerda.

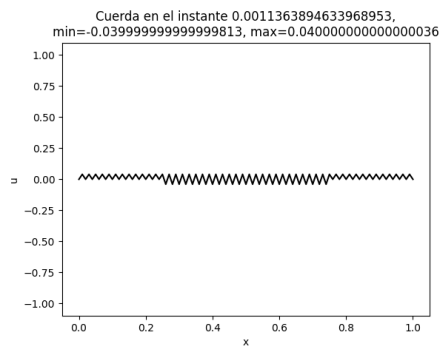


Figura 5.8

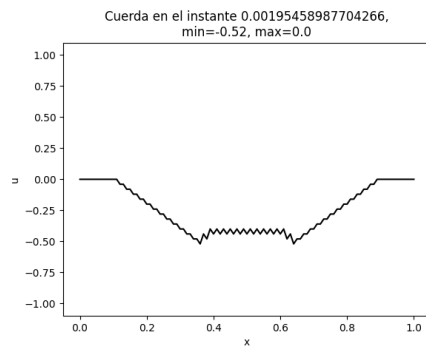


Figura 5.9

5. MODELADO EN PYTHON Y COMPARATIVA

En este caso la forma de la onda en cada punto no es tan regular, y cambia mucho dependiendo del punto que observemos, como ejemplo visualizaremos el movimiento de los puntos $x = 0,07$ y $x = 0,5$.

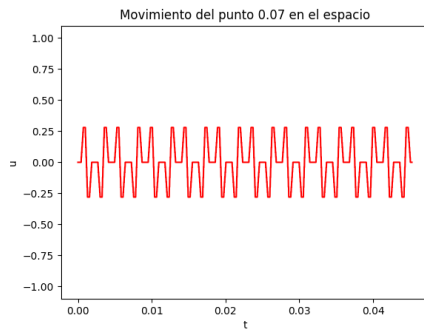


Figura 5.10: Archivo figura5-10.wav

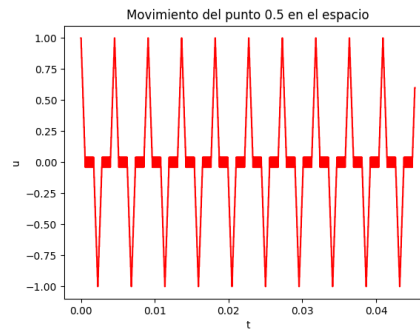


Figura 5.11: Archivo figura5-11.wav

Se observa que la onda varía dependiendo de la posición de la cuerda en las que se haga la observación, por ende, también el sonido producido.

5.1.3 $f(x) = 0, g(x)$ tipo Struck

En este caso partimos de una onda en reposo, con una velocidad inicial tipo Struck. A la derecha vemos una gráfica de la forma que tiene esta velocidad inicial.

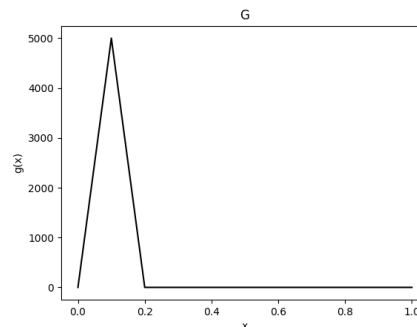


Figura 5.12: Velocidad inicial.

En esta imagen vemos la posición inicial. Dependiendo del número de puntos que elijamos en nuestro mallado, podremos ver que la solución se hace más suave, lo ilustraremos eligiendo 50 y 200 puntos en el eje espacial.

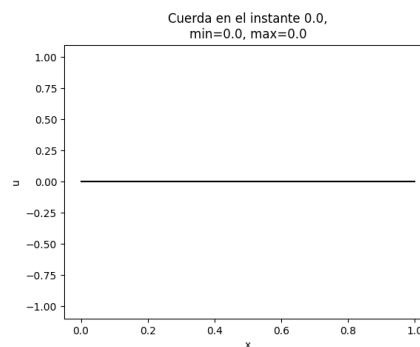


Figura 5.13: Posición inicial de la cuerda.

Muestra de la solución con $m = 50$.

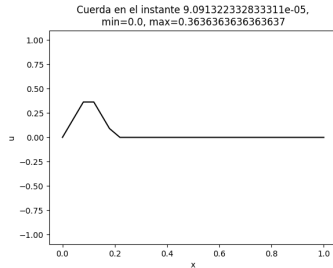


Figura 5.14

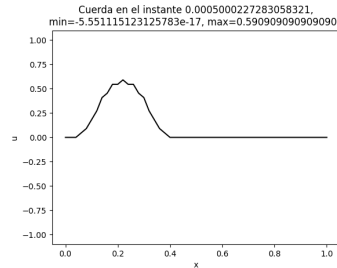


Figura 5.15

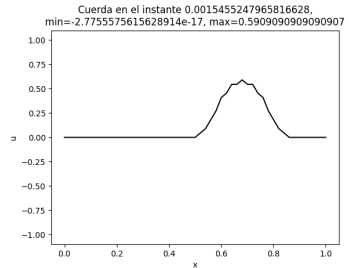


Figura 5.16

Muestra de la solución con $m = 200$:

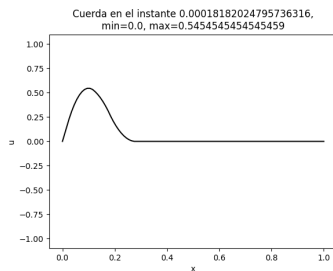


Figura 5.17

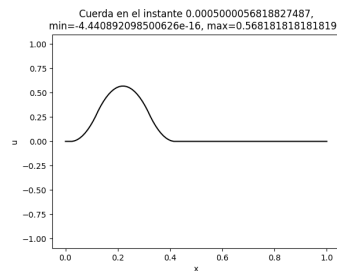


Figura 5.18

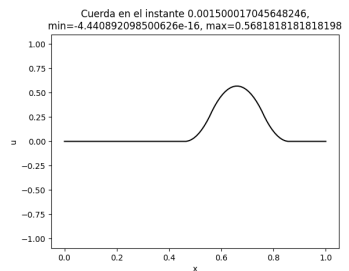


Figura 5.19

Aún así, para los puntos en los que hacemos la resolución, el movimiento de la onda será prácticamente idéntico y la diferencia en el audio indistinguible. Un ejemplo es el esquema de la derecha.

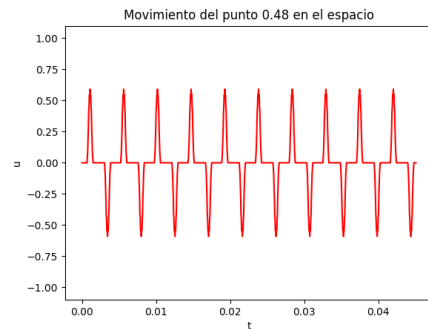


Figura 5.20: Archivo figura5-20.wav

5.1.4 $f(x) = \sin(2\pi x)$, $g(x) = 0$

En cada uno de los casos anteriores, tal y como hemos dicho al principio de la sección, $c = 440$ y como solo hay dos nodos, puntos en los que la amplitud es siempre 0, en todos los casos la frecuencia será 440Hz . Esta frecuencia corresponde a un La central (4ª octava), la hemos elegido al ser la más usual para ejemplificar este tipo de modelado.

5. MODELADO EN PYTHON Y COMPARATIVA

Ahora, elijamos un caso en el que tengamos 3 nodos, por ejemplo $f(x) = \text{sen}(2\pi x)$, La frecuencia se corresponderá a un La de 5ª octava.

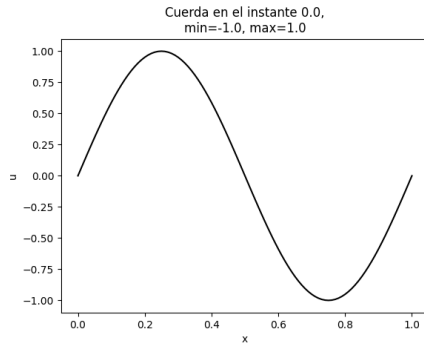


Figura 5.21: Posición inicial.

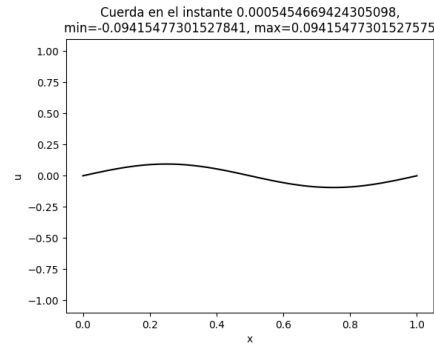


Figura 5.22

Podemos observar esto eligiendo los puntos adecuados, $x = 0,5$ y $x = 0,25$, para seno de πx , 5.1.1, y seno de $2\pi x$. Vemos claramente que llega a su máxima amplitud el doble de veces.

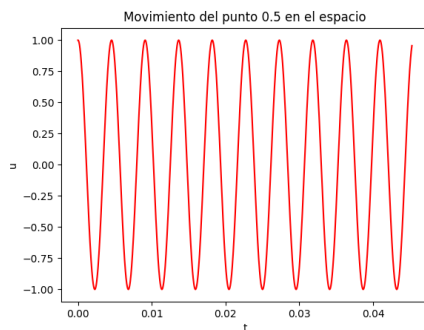


Figura 5.23: $\sin(x)$ en $x = 0,5$ Archivo figura5-5.wav

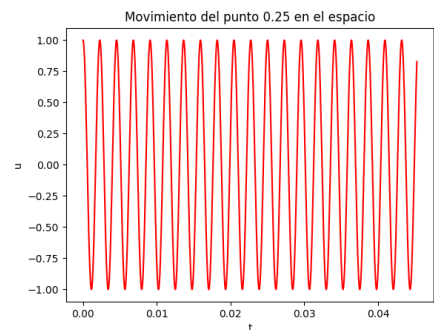


Figura 5.24: $\sin(2x)$ en $x = 0,25$ Archivo figura5-24.wav

5.2 Soluciones tipo onda viajera

En esta sección nuestro objetivo será ejemplificar lo explicado en la sección 4.5. Utilizando de nuevo la función seno, las soluciones de tipo onda viajera en los siguientes casos serían:

5.2 Soluciones tipo onda viajera

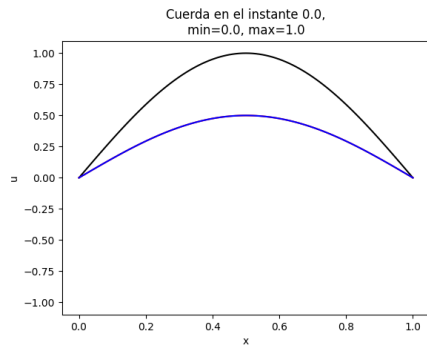


Figura 5.25: Superpuestas ambas soluciones.

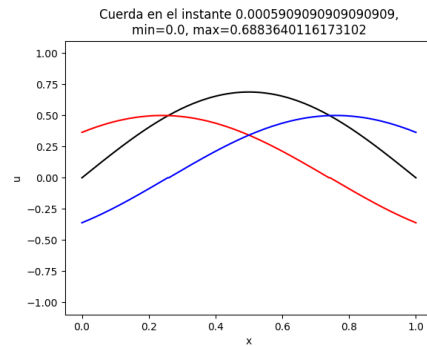


Figura 5.26: Coloreada de azul la solución que se desplaza a la derecha y de rojo la que se desplaza hacia la izquierda.

Comprobamos que la suma de ambas soluciones nos da la posición de la cuerda en cada instante, coloreada de negro.

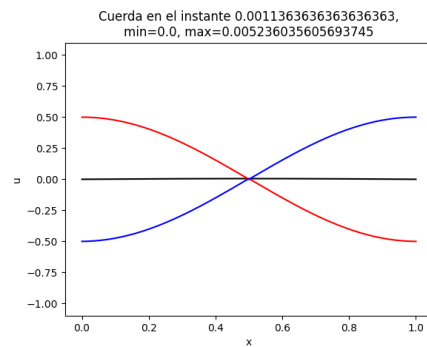


Figura 5.27: Movimiento de las soluciones de tipo onda viajera.

Esto se comprueba de manera incluso más clara con la función $\text{sen}(2\pi x)$.

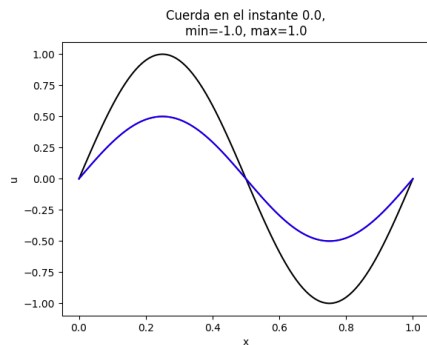


Figura 5.28

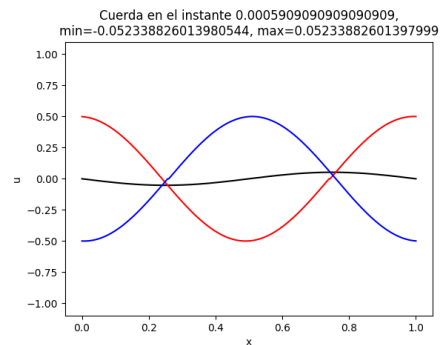


Figura 5.29

5. MODELADO EN PYTHON Y COMPARATIVA

5.3 Comparación variando el valor de μ

Ahora veremos como cambiando el parámetro μ , las soluciones obtenidas propagarán inestabilidades.

5.3.1 $\mu \ll 1$

Cuando elegimos μ mucho menor que 1, es decir, alejándonos de la solución óptima, poco a poco comienzan a generarse inestabilidades. Comparemos la solución con $\mu = 1$, para $f(x)$ de tipo Struck.

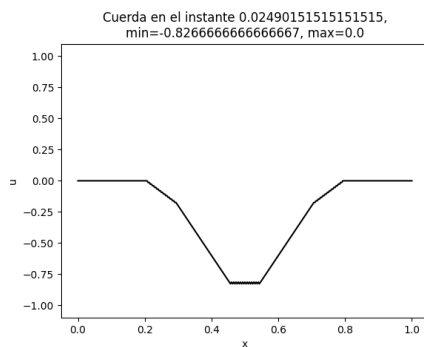


Figura 5.30: Solución para $\mu = 1$.

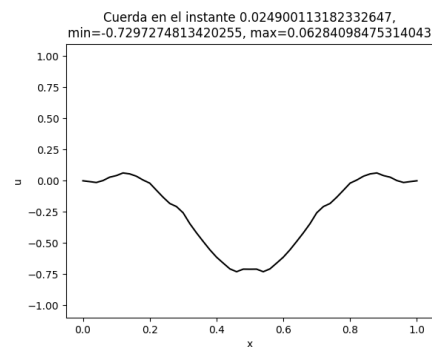


Figura 5.31: Solución para $\mu = 0,1$.

En la imagen de la derecha podemos observar las inestabilidades en forma de ondas por toda la cuerda. Por otro lado, vemos claramente las diferencias en las imágenes del movimiento de la onda dependiendo del tiempo, con el punto $x = 0,5$ fijado.

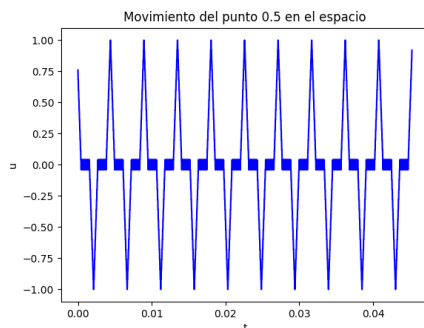


Figura 5.32: Movimiento de la onda para $\mu = 1$. Archivo figura5-32.wav

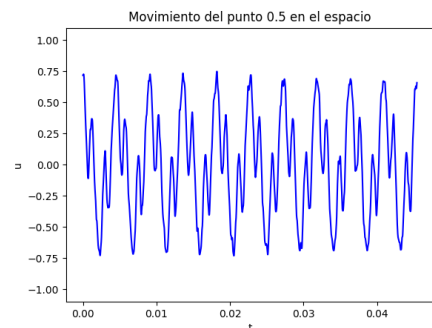


Figura 5.33: Movimiento de la onda para $\mu = 0,1$. Archivo figura5-33.wav

5.3 Comparación variando el valor de μ

Aún viendo claramente las inestabilidades de forma gráfica, no es tan fácil escucharlas en los archivos de audio adjuntos.

5.3.2 $\mu > 1$

Cuando violamos la condición CFL, las inestabilidades se propagarán aún más rápido. Utilizando el mismo ejemplo que en el apartado anterior, podremos comprobar como se propagan a todos los puntos de la cuerda antes de llegar a los 0,002 segundos.

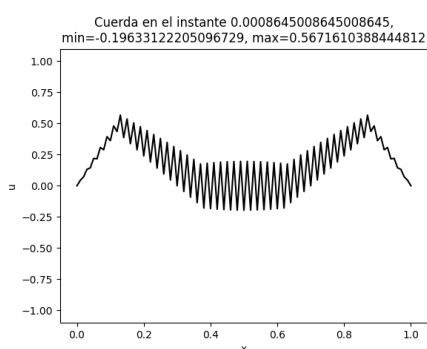


Figura 5.34: Comienzo de las inestabilidades.

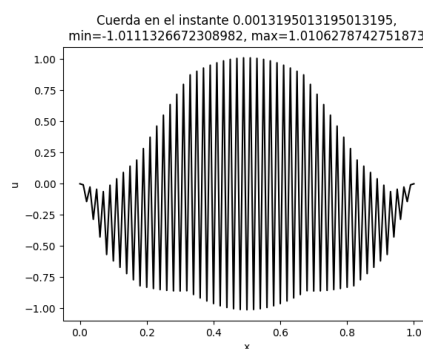


Figura 5.35: Solución para $\mu = 0,1$.

En este caso, a mayor sea m antes se propagarán las inestabilidades, por ello hemos elegido $m = 40$, para observar como cambia la onda dependiendo del tiempo.

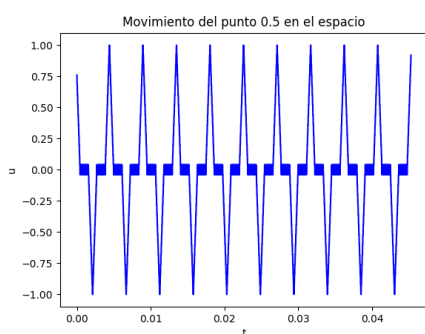


Figura 5.36: Movimiento de la onda para $\mu = 1$.

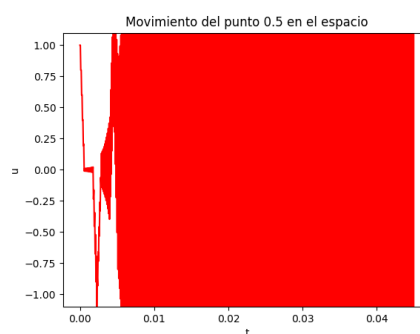


Figura 5.37: Movimiento de la onda para $\mu = 0,1$. Archivo fig5-38.wav

5. MODELADO EN PYTHON Y COMPARATIVA

Si intentamos reproducir el sonido de esta última situación no se escuchará nada, puesto que las inestabilidades son tan grandes que hacen imposible su escucha, dos causas posibles son:

- El guardado en el formato .wav, puesto que cuando la amplitud es mayor que 1, no lo capta de manera adecuada.
- La frecuencia, al ser tan elevada, es imposible de escuchar para el oído humano.

5.4 Comparar cambiando la energía

Cambio la energía y comparo

Conclusiones y proyectos futuros

A lo largo de este trabajo, hemos examinado la solución numérica de la ecuación de onda mediante el método de las diferencias finitas, observando en 5.1, como se obtienen buenos resultados para la misma. Hemos comparado distintas condiciones iniciales que cumplen con los parámetros obtenidos mediante el análisis de Von Neumann 4.4 y el análisis energético 4.6, y hemos conseguido archivos de audio que lo corroboran.

Por otro lado, hemos logrado la implementación de las soluciones de tipo onda viajera 5.2, de las que derivamos que la suma de las mismas describe inequívocamente el movimiento de la cuerda de nuestro esquema.

A continuación hemos representado la relación entre la condición CFL y la inestabilidad de las soluciones, observando como pequeñas variaciones en ella generan dicha inestabilidad de forma acelerada.

Por último se ha visualizado como si no se respeta la ley de la conservación de la energía, es decir, se incumplen las condiciones de contorno dadas en la sección 4.6, la solución numérica va perdiendo energía hasta pararse. (NO ESTOY SEGURA TENGO QUE COMPROBARLO CUANDO TERMINE LA SECCIÓN) Esto puede comprobarse en la sección 5.4 (COMPLETAR CUANDO ESTÉ HECHO)

6. CONCLUSIONES Y PROYECTOS FUTUROS

En definitiva, hemos podido realizar un estudio general de la ecuación de onda en derivadas parciales, abordando desde los conceptos fundamentales hasta la comprobación de como las condiciones teóricas encontradas afectan la solución numérica de la misma.

Este estudio sería fácilmente ampliable tomando varios enfoques,

- Utilizando argumentos similares a los encontrados en [15], podríamos hablar de la controlabilidad y observabilidad de nuestro esquema, en lugar de en un ambiente semi-discreto como el que propone Enrique Zazua.
- Se podría enriquecer el estudio comparando diferentes análisis usados asiduamente en el método de las diferencias finitas.
- También sería interesante el uso de métodos de resolución diferentes, como por ejemplo los métodos espectrales.
- Un enfoque menos teórico que se podría dar es, la utilización de los archivos .wav sintetizados para la creación de música, ya que hemos visto, que dependiendo de las condiciones iniciales, la forma de la onda obtenida cambia, de manera similar a como lo hace al tocar la misma nota en dos instrumentos diferentes.

APPENDIX



Anexo

En este capítulo anexaremos el código realizado en Python.

A.1 Resolución numérica

Comenzamos implementando el esquema (4.6) y así resolviendo la ecuación de onda en ese contexto.

```
import numpy as np
import time
import os as os
from shutil import rmtree
import matplotlib.pyplot as plt

#Funciones necesarias

# Para inicializar los valores iniciales que dependen de la
# derivada lo haremos con una aproximación progresiva del
# tiempo definimos la funcion g del esquema.

def g(x):
    return 0
```

A. ANEXO

```
'''
# En este caso tomamos una condición tipo struck en 0.4,
0.5 max
def g(x):
    if( x<=0.4):
        return 2.5*x
    else:
        return -(10/6)*x
'''
```

```
'''
# En este caso tomamos una condición tipo struck en 0.1,
1 max
```

```
def g(x):
    if( x<=0.1):
        return 50000*x
    if (x <= 0.2):
        return 50000*(0.2-x)
    else:
        return 0
'''
```

```
'''
# En este caso tomamos una condición tipo crc c0=1000,
x0 = 0.5
```

```
def g(x):
    if (0.4<=x <= 0.6):
        return 1000/2 * (1+np.cos(np.pi*(x-0.4)/0.1))
    else:
        return 0
'''
```

```
# Condición de contorno en el eje de posición tipo seno
```

```
def f(x):
```

```
aux = []
for i in x:
    aux.append(np.sin(2*np.pi*i))
return aux # Tener cuidado con la función por si se sale
           del rango

'''
#Condición de contorno en el eje de posición tipo pico
def f(x):
    aux = []
    for i in x:
        if( i<=0.25):
            aux.append(0)
        elif ( i<=0.5):
            aux.append(4*(i-0.25))
        elif ( i<=0.75):
            aux.append(4*(0.75-i))
        else:
            aux.append(0)
    return aux
'''

'''
#Condición de contorno en el eje de posición tipo trapecio
def f(x):
    aux = []
    for i in x:
        if( i<=0.1):
            aux.append(0)
        elif ( i<=0.3):
            aux.append(5*i-0.5)
        elif ( i<=0.7):
            aux.append(1)
        elif ( i<=0.9):
            aux.append(-5*i+4.5)
        else:
            aux.append(0)
    return aux
```

A. ANEXO

```
'''

'''
#condicioón de tipo c_rc con x_hw= 0.2 x_0 = 0.5 c_0= 1
def f(x):
    aux = []
    for i in x:
        if( i<=0.3):
            aux.append(0)
        elif ( i<=0.7):
            aux.append((1+np.cos(np.pi * (i-0.5)/0.2))/2)

    else:
        aux.append(0)
    return aux
'''

'''
# Condición de contorno en el eje de posición tipo plano
def f(x):
    lon = len(x)
    return np.zeros(lon)
'''

# Definimos una función para usar en el método explícito y
poder cambiarlo de manera rápida si hace falta
def Explicito(x1, x2, x3, x4,mu): # Si queremos hacerlo
en un file independiente
tendremos que añadir el mu
return mu ** 2 * (x1 + x2) + 2 * (1- mu ** 2) * x3 - x4

start = time.time()
print("Vamos a hacer una resolución numérica de la ecuación de
    onda  $u_{tt}-c^2u_{xx}=0$ ")
print("Introduce c:")
c= float(input())
print("Introduce el nombre del directorio donde quieres guardar
los archivos:")
```

```
directorio=input()

#Creamos una carpeta en la que se guardarán los datos
home = 'C:\\Users\\Ana Cuevas de Cózar\\PycharmProjects\\
pythonProject2'
directoriofinal=home+'\\'+directorio+'\\'

#Si existe el directorio lo borra
try:
    rmtree(directoriofinal)
except:
    print('No existía el directorio, lo creamos')
    os.mkdir(directoriofinal)

#Constantes necesarias
l=1
tmax= 1
m = int(l*300) #Constante representativa del número de trozos
en los que separamos el espacio
h=l/m #Cambiamos la h para que sea entera

k =h/c #Lo tomamos de esta forma para ganar en exactitud pagina
133 numerical sound synthesis

#Cambiamos k para hacer las comparaciones
'''
k =h/c * 0.9
k =h/c * 0.5
k =h/c * 0.1
k =h/c * 0.01
k =1.001*h/c
k =1.005*h/c
'''

n= int(tmax/k)
mu = c * k / h
```

A. ANEXO

```
if mu>1 :
    raise Exception('Mu es mayor que 1')
print(f'n{n}, c{c}, k{k}, h{h}, m{m}, mu{mu}')

# Creamos un mallado de puntos en los que aproximaremos la
solución
x= np.linspace(0,l,num=m +1)
t= np.linspace(0,tmax,num=n +1)

# Creamos la matriz u de soluciones u(x,t)
u = np.empty((m + 1, n + 1), float)

# Añadimos antes que nada los nodos conocidos por las
condiciones de contorno
u[:, 0] = f(x)

u[0, :] = np.zeros(n + 1)

u[m, :] = np.zeros(n + 1)
aux=[0]

# Para inicializar los valores iniciales que dependen
de la derivada lo haremos con una aproximación progresiva
del tiempo

for i in range(1, m):
    u[i, 1] = u[i, 0] + k * g(x[i])
    aux.append(g(x[i]))

aux.append(g(x[m]))

for j in range(2, n + 1):
    u[1:m, j] = Explicito(u[2:m+1,j-1], u[0:m-1,j-1],
```

```

u[1:m, j-1], u[1:m, j-2], mu)

#Implementamos la solución de tipo onda viajera.
Solo para mu=1

# Creamos la matriz wl la solución viajera hacia la
izquierda y wr hacia la derecha

wl = np.empty((m + 1, n + 1), float) #Solo para mu=1
wr = np.empty((m + 1, n + 1), float) #Solo para mu=1

#Inicializamos la solución
for i in range(0, m+1):
    if i==0:
        sumx =2* g(x[0])
    elif i==1:
        sumx = g(x[0])+g(x[1])
    else:
        sumx += g(x[i-1])+g(x[i])
    wl[i, 0] = 1 / 2 * f([x[i]])[0] + h / (4 * c) * (sumx)
        + k / 4 * 2*g(x[i]+c*t[0]) #Prueba apendice libro Bilbao

wr[0:m+1,0]=wl[0:m+1,0]

for j in range(1, n + 1):
    temp1 = wr[m, j-1]
    temp2 = wl[0, j-1]
    wr[1:m+1, j] = wr[0:m, j-1]
    wl[0:m, j] = wl[1:m+1, j-1]
    wr[0, j] = -temp2
    wl[m, j] = -temp1

#Termina la parte de la solución de tipo onda viajera

end = time.time()

```

A. ANEXO

```
plt.plot(x,aux,'k')
plt.xlabel('x')
plt.ylabel('g(x)')
plt.title(f'G')
plt.savefig(directoriofinal+'prueba g.png')
plt.close()

print(f'Ha tardado {np.floor(end-start)} segundos.')
np.save(directoriofinal+'u', u)
np.save(directoriofinal+'x', x)
np.save(directoriofinal+'t', t)

np.save(directoriofinal+'wl', wl) #Solo para mu=1
np.save(directoriofinal + 'wr', wr) #Solo para mu=1

np.save(directoriofinal+'constantes', [c, m,h,n,k,mu,tmax,l] )
#Guardar en cons las constantes que necesitemos para usarlas luego
en el guardado del sonido
```

Al final guardamos los datos obtenidos para su posterior procesado.

A.2 Wave

Creo ahora un archivo de audio a partir de los datos anteriores y lo guardo en un archivo de sonido .wav

```
import numpy as np
from scipy.io.wavfile import write

print("Vamos a guardar nuestros datos en formato wav")
print("Introduce el nombre de la carpeta:")
dir= str(input())

#Creamos el directorio
```



```
directorio='C:\\Users\\Ana Cuevas de Cózar\\PycharmProjects\\pythonProject2\\'+dir+'\\'

#Añadimos los datos
datos = np.load(directorio+'u.npy', mmap_mode='r')
t = np.load(directorio+'t.npy', mmap_mode='r')
x = np.load(directorio+'x.npy', mmap_mode='r')
[f, m1,h1,n1,k1,mu,tmax]= np.load(directorio+'constantes.npy',
    mmap_mode='r')

m1=int(m1)
n1=int(n1)

#Constantes necesarias
sample_rate = 44100
n2 = int(tmax*sample_rate)
t_necesario= np.linspace(0,tmax,n2 +1)

# Escogiendo un punto en particular
Comprobamos donde está el punto de mayor amplitud en el
instante inicial y en ese punto es donde veremos como
se mueve la onda.
pos_readout= np.where(datos[:,0]==max(datos[:,0]))[0][0]

if pos_readout==0:
pos_readout= 50
print('Cambio la posicion a 0.5')

wave_table = datos[pos_readout] # Elijo un punto en el
espacio en el que veremos el movimiento de la cuerda

wave_table2 = datos[pos_readout+1] # Elijo un punto en el
espacio en el que veremos el movimiento de la cuerda
output = np.interp(t_necesario, t, wave_table)

# Sumando todos los puntos en espacio que tenemos.
```

A. ANEXO

```
#wave_table = np.zeros(n1 + 1)
#for i in range(0, m1+1):
#wave_table += datos[i, :]

#wave_table= wave_table/(m1+1)
#output = np.interp(t_necesario, t, wave_table)

#Vamos a modificar los limites para que suene a un nivel
# de volumen adecuado, para ello teben ser en torno al
# orden e-1 o e-2
#gain= 0
#amplitud = 10** (gain/20)
#output *=amplitud

# Para que al principio y final se escuche mas suave
#output = fade(output)

#Guardamos
write(directorio+f'Sonido.wav', sample_rate, output)
```

A.3 Imágenes

Creamos y guardamos las imágenes de la posición de la cuerda en varios instantes de tiempo y de los cambios de posición de la onda en un punto del eje X durante un periodo de tiempo.

```
import numpy as np
import matplotlib.pyplot as plt
from os import mkdir
from shutil import rmtree

print("Vamos a guardar nuestras imagenes")
print("Introduce el nombre de la carpeta:")
dir= str(input())

#Creamos los directorios
```

```
directorio='C:\\Users\\Ana Cuevas de Cózar\\PycharmProjects\\
pythonProject2\\'+dir+'\\'
cuerda=directorio + 'Imagenes Cuerda\\'
onda= directorio+ 'Imagenes Onda\\'

try:
    rmtree(cuerda)
    rmtree(onda)
except:
    print('No existían las imagenes, las creamos')
    mkdir(onda)
    mkdir(cuerda)

#Añadimos los datos
datos = np.load(directorio+'u.npy', mmap_mode='r')

#Solo para ondas viajeras, para mu=1
datoswl = np.load(directorio+'wl.npy', mmap_mode='r') #Solo para mu=1
datoswr = np.load(directorio+'wr.npy', mmap_mode='r') #Solo para mu=1

t = np.load(directorio+'t.npy', mmap_mode='r')
x = np.load(directorio+'x.npy', mmap_mode='r')
[f, m,h,n,k,mu,tmax]= np.load(directorio+'constantes.npy',
mmap_mode='r')

n=int(n)
m=int(m)

#Vamos a pintar las imagenes de a posición de la cuerda en
    instantes seleccionados
plt.plot(x, datos[:, 0], 'k')

plt.plot(x, datoswl[:, 0], 'r') #Solo para mu=1
plt.plot(x, datoswr[:, 0], 'b') #Solo para mu=1

plt.ylim(-1,1)
```

A. ANEXO

```
plt.xlabel('x')
plt.ylabel('u')
plt.title(f'Cuerda en el instante {t[0]}, \nmin={min(datos[:,0])},
        max={max(datos[:,0])}')
plt.savefig(cuerda+f'Cuerda_instante_{0}.png')
plt.close()
```

```
#Elegimos cuantas imagenes pintar, al menos un ciclo completo.
if (int(2*n/f)+2)/100 > 1:
    aux=int((int(2*n/f)+2)/100)
else:
    aux=1
```

```
for j in range(1, 2*int(2*n/f)+2):
    if j %aux ==0 :
        plt.plot(x, datos[:, j], 'k')
        plt.ylim(-1,1)
        plt.xlabel('x')
        plt.ylabel('u')
        plt.title(f'Cuerda en el instante {t[j]}, \nmin={min(datos[:,j])},
                max={max(datos[:,j])}')
        plt.savefig(cuerda+f'Cuerda_instante_{j}.png')
        plt.close()
```

```
#Pintamos la imagen de la onda en los 1000 primeros instantes
en cada punto
for j in range(0, m + 1):
    plt.plot(t[0:10*int(2*n/f)+2], datos[j,0:10*int(2*n/f)+2], 'r')
    plt.ylim(-1.1, 1.1)
    plt.xlabel('t')
    plt.ylabel('u')
    plt.title(f'Movimiento del punto {x[j]} en el espacio') #axis tight
    plt.savefig(onda+f'PrimerosmilPunto_{j}.png')
    plt.close()
```

```
plt.plot(t[0:10*int(2*n/f)+2], datos[j, n - (10*int(2*n/f)+2):n], 'b')
plt.ylim(-1.1, 1.1)
plt.xlabel('t')
plt.ylabel('u')
plt.title(f'Movimiento del punto {x[j]} en el espacio') # axis tight
plt.savefig(onda + f'UltimosmilPunto_{j}.png')
plt.close()
```


Bibliografía

- [1] H.L. Royden and P. Fitzpatrick. *Real Analysis*. Prentice Hall, 2010. 3
- [2] Abdul-Majid Wazwaz. *Partial Differential Equations and Solitary Waves Theory*. 01 2009. 11
- [3] K. W. Morton, D. F. Mayers, and D. F. Mayers. *Numerical solution of partial differential equations: an introduction*. Cambridge University Press, Cambridge, 2nd edition, 2005. 11, 28
- [4] Patricia Camarena. La transferencia del conocimiento: ecuaciones diferenciales parciales hacia una cuerda que vibra. 2004. 12
- [5] Agustín. Apuntes de vibraciones y ondas. 12
- [6] Victor G. Ganzha and Evgenii V. Vorozhtsov. *Numerical solutions for partial differential equations: problem solving using Mathematica*. CRC Press, Boca Raton (Florida) [etc.], 1996. 14
- [7] W.A. Strauss. *Partial Differential Equations: An Introduction*. Wiley, 2008. 15, 23
- [8] Shoichiro Nakamura. *MATLAB todos numéricos aplicados con software*. Prentice-Hall Hispanoamericana, México [etc.], 1992. 25
- [9] W.A. Smith and F.J.S. Bernabe. *Análisis numérico*. Prentice Hall, 1988. 25
- [10] Enrique Zuazua. Optimal and approximate control of infinite — difference approximation schemes for the 1d wave equation. *Rendiconti di Matematica ed delle sue Appl.*
- [11] Stefan Bilbao. *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Wiley Publishing, 2009.

BIBLIOGRAFÍA

- [12] Juan Carlos Solares Carreño and Rosalba Mendoza Suárez. Acerca de algunas soluciones de ciertas ecuaciones de onda. 2012.
- [13] J. Vankka. *Digital Synthesizers and Transmitters for Software Radio*. Springer US, 2005.
- [14] M.E. Van Valkenburg. *Network Analysis 3rd Edition*. Pearson Education, 2006.
- [15] Enrique Zuazua. Optimal and approximate control of finite-difference approximation schemes for the 1d wave equation. *Rendiconti di Matematica e delle sue Applicazioni. Serie VII*, 2004. 44
- [16] Bo Strand. Summation by parts for finite difference approximations for d/dx . *Journal of Computational Physics*, 110(1):47–67, 1994.
- [17] David L. Russell. Controllability and stabilizability theory for linear partial differential equations: Recent progress and open questions. *SIAM Review*, 20(4):639–739, 1978.
- [18] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, Second Edition*. Society for Industrial and Applied Mathematics, 2004.
- [19] Stig Larsson and Vidar Thomée. *Partial differential equations with numerical methods*, volume 45. Springer, Berlin, 1st softcover pr. edition, 2009.
- [20] E.W. Chen, Q. Luo, N.S. Ferguson, and Y.M. Lu. A reflected wave superposition method for vibration and energy of a travelling string. *Journal of Sound and Vibration*, 400:40–57, 2017.