

UNIVERSIDAD POLITÉCNICA DE CARTAGENA
Escuela Técnica Superior de Ingeniería Industrial



**Simulación numérica del control exacto
en la frontera para la ecuación de ondas**

Titulación: Ingeniero Industrial
Intensificación: Sistemas eléctricos
Alumno/a: Jorge Emilio Villena Lapaz
Director/a/s: Francisco Periago Esparza
Pablo Pedregal

Cartagena, 3 de Octubre de 2008

SIMULACIÓN NUMÉRICA DEL CONTROL EXACTO EN LA FRONTERA PARA LA ECUACIÓN DE ONDAS

Jorge Emilio Villena Lapaz
Escuela Técnica Superior de Ingenieros Industriales
Universidad Politécnica de Cartagena

Cartagena, 2008

Índice general

I	Introducción	1
I.1	Problema de controlabilidad	2
II	El caso unidimensional	5
II.1	Fórmulas explícitas para el control y estado	5
II.2	Simulaciones Numéricas	6
II.2.1	Ejemplo 1.1: datos iniciales regulares	6
II.2.2	Ejemplo 1.2: datos iniciales irregulares	6
III	El caso bidimensional	11
III.1	Decaimiento local de la energía. Principio de Russell	11
III.2	Algoritmo Numérico para el Cálculo del Control	14
III.3	Simulaciones Numéricas	14
IV	Conclusiones y Problemas Abiertos	21
V	Apéndice	23
V.1	Interpretación física de la ecuación de ondas en el caso unidimensional	23
V.2	Solución de D'Alembert de la ecuación de onda	25
V.3	Interpretación física de la ecuación de ondas en el caso bidimensional	26
V.4	Interpretación física de la ecuación de ondas en el caso tridimensional	28
V.5	Solución de la ecuación de ondas para los casos $n = 2$ y $n = 3$	29
V.5.1	Solución para $n = 3$	30
V.5.2	Solución para $n = 2$	31
V.6	Demostración <i>Lema 1</i>	33
V.7	Códigos en MATLAB	34
V.7.1	wave_control_1d.m	34
V.7.2	delta_wave_control_1d.m	35
V.7.3	Simpson_v.m	36
V.7.4	ondas_controld.m	36
	Bibliografía	51

CAPÍTULO I

Introducción

Este proyecto surgió de los trabajos de Pablo Pedregal y Francisco Periago sobre el control en la frontera para sistemas lineales de segundo orden del tipo

$$\begin{cases} u'' + Au = 0, & \text{en } Q = \Omega \times (0, T) \\ B_j u = v_j & \text{en } \Sigma = \Gamma \times [0, T] \text{ para } j = 1, \dots, m \\ u(0) = u^0, u'(0) = u^1 & \text{en } \Omega \end{cases}$$

donde $\Omega \subset \mathbb{R}^n$ es un dominio acotado con frontera regular Γ , A es un operador elíptico de orden $2k$, $k = 1, 2, \dots$, con coeficientes constantes, y B_j , $j = 1, \dots, m$, es una familia de operadores lineales que actúan sobre la variable $x \in \Gamma$ para todo tiempo $0 \leq t \leq T$

En concreto, trabajaremos sobre la ecuación de ondas, que es un caso particular de la anterior si hacemos $A = -c^2 \Delta$, donde Δ es el operador Laplaciano y $c > 0$ es una constante que depende del medio y del significado físico de u , y $B_j = 1$, $j = 1, \dots, m$. Las condiciones de contorno del sistema nuevo significan que en una parte del contorno, Γ_0 , no hay movimiento, mientras que en el resto de la frontera, Γ_1 , actúa una función a la que llamaremos *función de control* f . Queda, entonces:

$$\begin{cases} u_{tt} - c^2 \Delta u = 0, & \text{en } \Omega \times (0, T), (\Omega \subset \mathbb{R}^n, n = 1, 2, 3) \\ u = 0, & \text{en } \Gamma_0 \times [0, T] \\ u = f, & \text{en } \Gamma_1 \times [0, T] \\ u(0) = u^0, u_t(0) = u^1 & \text{en } \Omega \end{cases} \quad (\text{I.1})$$

El sistema anterior modeliza un buen número de problemas físicos de interés en Ingeniería, tales como la vibración transversal de cuerdas y membranas elásticas, la vibración longitudinal y torsional de barras o las vibraciones en un sólido tridimensional.

Una cuestión de gran importancia consiste en tratar de eliminar o reducir dichas vibraciones. Este es precisamente el problema básico que abordamos en este proyecto.

I.1. Problema de controlabilidad

Dado un tiempo fijo $T > 0$ y unos datos iniciales (u^0, u^1) en un espacio apropiado, el problema de controlabilidad exacta para el sistema (I.1) consiste en encontrar una función de control $f : \Gamma_1 \times [0, T]$ tal que en el tiempo T el sistema se estabilice al estado nulo, es decir, que para el instante T se cumpla:

$$u(\cdot, T) = u_t(\cdot, T) = 0 \quad \text{en } \Omega. \quad (\text{I.2})$$

Este problema, desde una perspectiva teórica, fue estudiado y resuelto algunas décadas atrás por distintos caminos. Resulta especialmente interesante mencionar dos de ellos. Por un lado, a principios de la década de los 70 Russell [5] desarrolló un método general válido para ecuaciones hiperbólicas y parabólicas, basado en la disipación exponencial de la energía. Por otro lado, a finales de los 80, Lions [2] introdujo el Método de Unicidad de Hilbert (HUM) en el que la controlabilidad exacta se deduce a partir de las soluciones del sistema homogéneo (es decir, no controlado) asociado.

Desde el punto de vista numérico, aunque se han hecho algunos progresos recientemente, el problema del cálculo numérico del control en la frontera para este tipo de problemas aún es un reto. La principal dificultad proviene del hecho de que algunos métodos numéricos que son estables al resolver problemas de valor inicial no convergen cuando se trata de controlar el sistema.

Según el método de Russell, la principal idea para el caso de la ecuación de ondas es asociar con el sistema de control (I.1)-(I.2) un problema de Cauchy en todo el espacio \mathbb{R}^n en el que los nuevos datos iniciales se definen a partir de los originales extendiéndolos de forma apropiada, como se verá más adelante con detenimiento. Llamemos U a la solución del nuevo problema del valor inicial.

$$\begin{cases} U_{tt}(x, t) - c^2 \Delta U(x, t) = 0, & x \in \mathbb{R}^n, t \in (0, T) \\ U(x, 0) = U^0(x), U_t(x, 0) = U^1(x), & x \in \mathbb{R}^n \end{cases} \quad (\text{I.3})$$

La extensión de los datos iniciales debe ser tal que se cumpla la condición de contorno en Γ_0

$$U(y, t) = 0, \quad y \in \Gamma_0, t \in [0, T]$$

La solución del problema (I.1) será la restricción a Ω de $U(x, t)$

$$u(x, t) = U(x, t)|_{\Omega}$$

La función $f(t) = U(y, t)$, $y \in \Gamma_1$ produce una disipación de energía en la zona donde estaba definido el problema original Ω . A partir de aquí, usando el principio de superposición, la condición de controlabilidad exacta (I.2) se deduce fácilmente.

El objetivo de este trabajo es aplicar el método de Russell al cálculo numérico del control en la frontera para el problema de controlabilidad exacta (I.1)-(I.2). Como veremos más adelante, el esquema numérico que se propone es muy sencillo y fácil de implementar en un ordenador. La convergencia del método se basa en el hecho de que el operador que conecta los datos iniciales originales con la restricción a Ω de la solución del problema de Cauchy asociado, es contractivo. Más aún, como la función de control f se obtiene como la suma de las restricciones a Γ de las soluciones de un

par de problemas de Cauchy, todo esquema numérico estable para resolver problemas del valor inicial lo será también para computar los controles en la frontera. A este respecto, el algoritmo propuesto no genera soluciones numéricas ficticias.

Como resultado de este método podemos usar la fórmula de D'Alembert ($n = 1$) (V.12) y la fórmula de Kirchoff ($n = 3$) (V.22) para dar una forma completamente explícita del control f . Para $n = 2$ no podemos obtener una fórmula explícita pero, usando la fórmula de Poisson (V.28) podemos obtener una aproximación numérica del control frontera.

Esta estrategia no es nueva para el caso $n = 1$; pero para el caso $n = 2$ no es tan evidente que el método funcione. De hecho, el uso del principio de Russell juega un papel clave para probar que la condición de controlabilidad exacta (I.2) se cumple. Las cosas se vuelven de nuevo fáciles para el caso $n = 3$. Por estos motivos, en este trabajo nos centraremos en el caso bidimensional. Pese a todo, dedicaremos el primer capítulo al caso unidimensional con el fin de ilustrar algunas de las ideas básicas subyacentes.

CAPÍTULO II

El caso unidimensional

En este primer capítulo abordaremos el caso unidimensional. Estudiaremos en concreto el caso de una cuerda situada a lo largo del eje OX , de longitud $L = 1$, fija en un extremo $x = 0$ y actuando sobre el otro extremo $x = 1$ la función de control f . Así pues, considerando el sistema (I.1), hemos tomado $\Omega = (0, L)$, Γ_0 el extremo $x = 0$ y $\Gamma_1 = \{L\}$. En tal caso, el sistema a resolver es el siguiente:

$$\begin{cases} u_{tt} - c^2 u_{xx} = 0, & \text{en } [0, 1] \times (0, T) \\ u = 0, & \text{en } \{x = 0\} \times [0, T] \\ u = f, & \text{en } \{x = 1\} \times [0, T] \\ u(x, 0) = u^0, \quad u_t(x, 0) = u^1 & \text{en } x \in [0, 1] \end{cases} \quad (\text{II.1})$$

La función $u(x, t)$ representa el desplazamiento transversal del punto x en el instante t , es decir, el estado en cada instante.

II.1. Fórmulas explícitas para el control y estado

Para obtener una representación explícita del control f , seguimos la estrategia descrita en la Introducción. El primer paso consiste en extender los datos iniciales u^0 y u^1 a todo \mathbb{R} de forma impar en $x \in (-1, 0)$ y con valor nulo en el resto:

$$U^0(x) = \begin{cases} u^0(x), & 0 < x < L \\ -u^0(-x), & -L < x < 0 \\ 0, & \text{c.c.} \end{cases} \quad y \quad U^1(x) = \begin{cases} u^1(x), & 0 < x < L \\ -u^1(-x), & -L < x < 0 \\ 0, & \text{c.c} \end{cases}$$

Con estos datos iniciales extendidos, hemos de resolver el sistema

$$\begin{cases} U_{tt} - c^2 \Delta U = 0, & \text{en } \mathbb{R} \times (0, T) \\ U(x, 0) = U^0(x), \quad U_t(x, 0) = U^1(x) & \text{en } x \in \mathbb{R} \end{cases} \quad (\text{II.2})$$

Como vemos, la extensión de los datos iniciales cumple con las condiciones de contorno del sistema (II.1).

La solución de la ecuación de ondas para el caso unidimensional viene dada por la fórmula de D'Alembert (ecuación (V.12)), que para el problema extendido, es

$$U(x, t) = \frac{1}{2} [U^0(x - ct) + U^0(x + ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} U^1(s) ds \quad (\text{II.3})$$

Además, la restricción de (II.3) a $[0, L]$ es solución de (I.1). La expresión (II.3) refleja la superposición de dos ondas que viajan en sentidos opuestos a lo largo de la recta real. El tiempo que necesitan para atravesar una longitud $2L$ viajando a velocidad c , es decir, la de la cuerda más la zona a la que hemos extendido los datos iniciales, es exactamente $T = \frac{2L}{c}$. Se puede comprobar que el sistema se estabiliza, efectivamente, para ese tiempo. Por esa razón lo llamaremos *tiempo mínimo de controlabilidad* $T_0 = \frac{2L}{c}$.

El control en la frontera $f(t)$ viene dado explícitamente por

$$f(t) = U(L, t) = \begin{cases} \frac{1}{2}u^0(L - ct) + \frac{1}{2c} \int_{L-ct}^L u^1(s) ds, & 0 < t < L/c \\ -\frac{1}{2}u^0(ct - L) + \frac{1}{2c} \int_{L-ct}^L u^1(s) ds, & L/c < t < 2L/c. \end{cases}$$

Nótese que estas representaciones para el estado y el control del sistema (I.1) no sólo se cumplen cuando los datos iniciales (u^0, u^1) son regulares, sino también para datos más irregulares, incluso discontinuos.

A continuación, mostramos un par de simulaciones numéricas. En ambos casos, el cálculo numérico de las integrales que aparecen en las fórmulas explícitas para el estado y el control ha sido llevada a cabo mediante la regla de Simpson (ver Apéndice, *simpson_v.m*).

II.2. Simulaciones Numéricas

II.2.1. Ejemplo 1.1: datos iniciales regulares

Consideremos primero para $L = 1$ y $c = 1$ los datos iniciales

$$u^0(x) = \begin{cases} 2x, & 0 \leq x \leq 0,5 \\ 2(1-x), & 0,5 < x \leq 1 \end{cases}, \quad u^1(x) = \begin{cases} 2x, & 0 < x \leq 0,5 \\ 0, & 0,5 < x \leq 1 \end{cases}.$$

La figura (II.1) muestra los resultados para el estado en diferentes tiempos y el control $f(t)$ (ver Apéndice, programas *wave_control_1d.m* y *simpson_v.m*).

II.2.2. Ejemplo 1.2: datos iniciales irregulares

Consideremos para $L = 1$ y $c = 1$ los datos iniciales irregulares

$$u^0 = 0, \quad u^1 = \delta_{0,5},$$

donde $\delta_{0,5}$ es la delta de Dirac centrada en $x = 0,5$. Las simulaciones numéricas muestran, como era de esperar, que tanto el estado como el control son discontinuos. (ver Apéndice, programa *delta_wave_control_1d.m*).

El tiempo mínimo de controlabilidad para cualquier par de datos iniciales (u^0, u^1) es $T_0 = 2s$. Sin embargo, como se ve en la figura (II.2), abajo a la derecha, el control

está inactivo para $t > 1,5s$. El hecho de que el sistema se estabilice para un tiempo menor que el tiempo mínimo de controlabilidad se debe a que, por la naturaleza de los datos iniciales del ejemplo, las ondas necesitan menos tiempo para salir de la región de la cuerda. Dicho de otra manera, al tener los datos iniciales soporte sólo en parte de la longitud de la cuerda (es decir, la zona en la que son distintos de cero está acotada a un intervalo más pequeño que el intervalo $[0, 1]$), el tiempo de control se reduce.

Se puede dar un sentido matemáticamente más riguroso al concepto de solución de la ecuación de ondas para este tipo de datos iniciales no regulares. Pero dado que ese no es el objetivo de este trabajo, no entraremos en más detalles sobre estas cuestiones más técnicas.

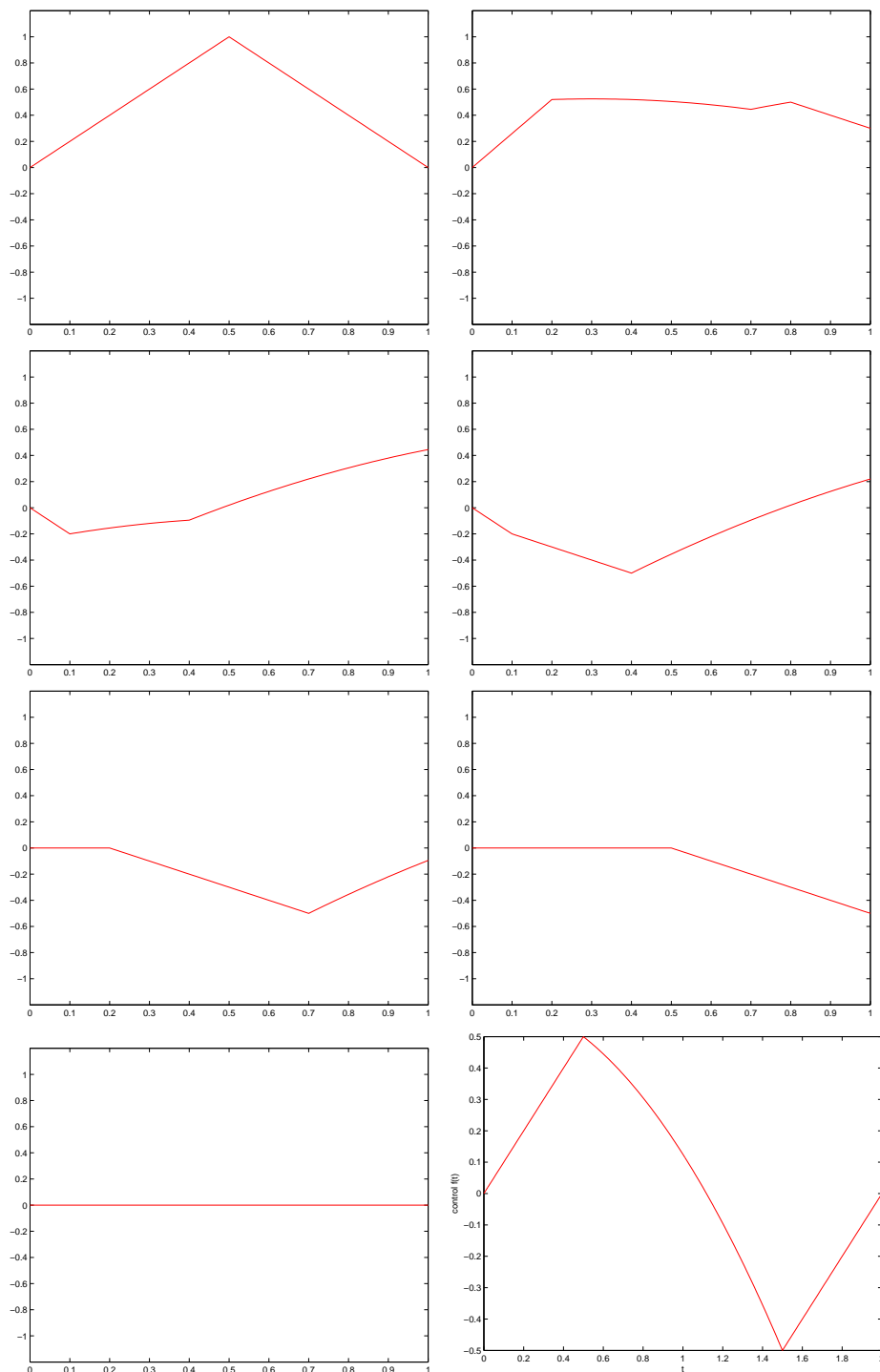


Figura II.1: animación del estado $u(x,t)$ de izquierda a derecha y de arriba hacia abajo para $t = 0, 0.3, 0.6, 0.9, 1.2, 1.5$ y 2 y representación del control $f(t)$ en el extremo $x = 1$.

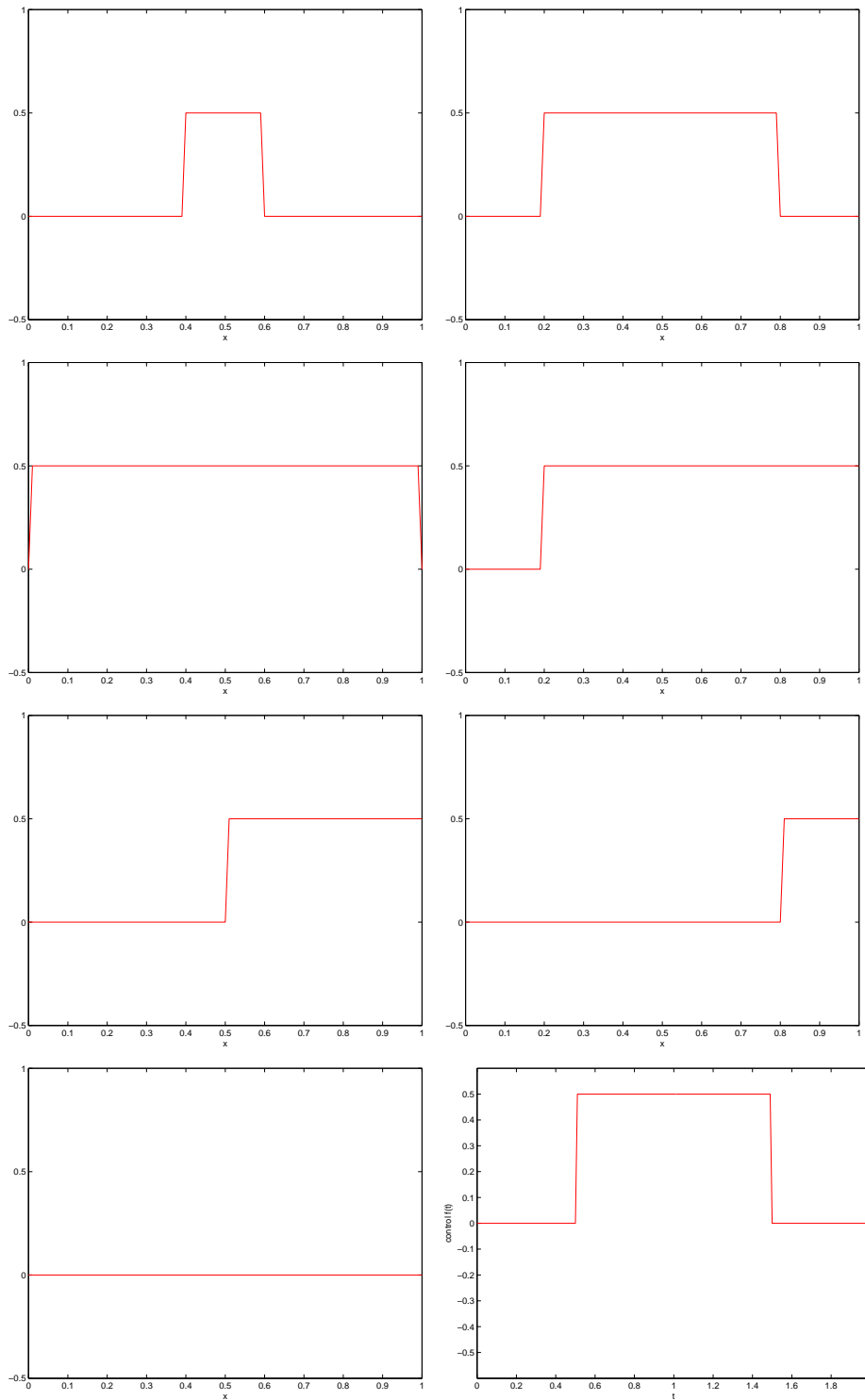


Figura II.2: animación del estado $u(x,t)$ de izquierda a derecha y de arriba hacia abajo para $t = 0.3, 0.6, 0.9, 1.2, 1.5$ y 2 , y representación del control $f(t)$ en el extremo $x = 1$.

CAPÍTULO III

El caso bidimensional

Como vimos en el capítulo anterior, la solución del sistema resultante de la extensión de los datos iniciales (II.2) conducía directamente a la solución del sistema original (II.1) y satisfacía la condición de controlabilidad exacta (I.2). En el caso que nos ocupa en el presente capítulo, desafortunadamente, esto no ocurre. Después de proponer una extensión de los datos iniciales adecuada y una solución al problema de Cauchy asociado correspondiente (V.28), veremos que la condición de controlabilidad no se cumple. En este momento entra en juego el principio de Russell, piedra angular de este trabajo, en virtud del cual podremos proponer un algoritmo que nos conduzca a la estabilización del sistema.

III.1. Decaimiento local de la energía. Principio de Russell

En esta sección consideramos el sistema (I.1) en un cuadrado unitario $\Omega \equiv R_1 = (0, 1)^2$. Dividimos la frontera Γ en dos partes

$$\Gamma_0 = \{(0, s) \in \mathbb{R}^n, 0 \leq s \leq 1\} \cup \{(s, 0) \in \mathbb{R}, 0 \leq s \leq 1\}$$

$$\Gamma_1 = \{(1, s) \in \mathbb{R}^n, 0 \leq s \leq 1\} \cup \{(s, 1) \in \mathbb{R}, 0 \leq s \leq 1\}.$$

Como en el caso unidimensional, nuestro objetivo principal en esta sección será calcular numéricamente el control en la frontera $f : \Gamma_1 \rightarrow \mathbb{R}$ de forma que para un tiempo dado T la solución de (I.1) satisfaga la condición de controlabilidad exacta (I.2).

Empezamos, como en el capítulo anterior, extendiendo los datos iniciales (u^0, u^1) a todo \mathbb{R}^n . Para ello consideramos los rectángulos

$$R_2 = (-1, 0) \times (0, 1),$$

$$R_3 = (-1, 0) \times (-1, 0),$$

$$R_4 = (0, 1) \times (-1, 0),$$

y denotamos

$$R = \bigcup_{i=1}^4 R_i.$$

Extendemos los datos (u^0, u^1) al rectángulo más grande R de forma impar en R_2 y R_4 y de forma par en R_3 . En el resto del plano extendemos los datos con valor nulo. Con esta extensión de los datos, el tiempo que necesita una onda que viaja a velocidad c para recorrer la máxima distancia posible¹ (una diagonal del rectángulo R , cuya longitud es $2\sqrt{2}$) es $T_0 = \frac{2\sqrt{2}}{c}$. Este será el tiempo mínimo de controlabilidad para el caso bidimensional.

Llamamos a los nuevos datos (U^0, U^1) y consideramos el problema auxiliar para $n = 2$:

$$\begin{cases} U_{tt} - c^2 \Delta U = 0, & \text{en } \mathbb{R}^2 \times (0, T) \\ U(x, 0) = U^0(x), U_t(x, 0) = U^1(x) & \text{en } x \in \mathbb{R}^2 \times \mathbb{R}^2 \end{cases} \quad (\text{III.1})$$

La solución de este sistema está dada explícitamente por la fórmula de Poisson

$$U(x, t) = \frac{\partial}{\partial t} \frac{1}{2\pi c} \int_{B(x; ct)} \frac{U^0(y) dy}{\sqrt{c^2 t^2 - |y - x|^2}} + \frac{1}{2\pi c} \int_{B(x; ct)} \frac{U^1(y) dy}{\sqrt{c^2 t^2 - |y - x|^2}}, \quad (\text{III.2})$$

donde $B(x; ct)$ es una bola centrada en x y de radio ct , es decir,

$$B(x; ct) = \left\{ y \in \mathbb{R}^2 : |y - x| = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} < ct \right\}$$

La restricción de U a $R_1 \times \mathbb{R}$ resuelve el sistema (I.1) con $f(y, t) = U(y, t)$, para $y \in \Gamma_1$ y $t > 0$, o lo que es lo mismo,

$$u \equiv U|_{R_1 \times \mathbb{R}} \quad (\text{III.3})$$

Pero si se calcula $u(x, T)$ y $u_t(x, T)$ se puede comprobar que no se satisface la condición de controlabilidad exacta (I.2).

A continuación enunciaremos un par de resultados² que mostrarán que la extensión de los datos iniciales propuesta lleva a la estabilización uniforme del sistema (I.1).

Lema 1.- Sean $(u_0, u_1) \in L^2 \times H^{-1}$ y $T > T_0 = \frac{2\sqrt{2}}{c}$. Existe una constante positiva $C(T)$, con $C(T) \rightarrow 0$ cuando $T \rightarrow \infty$, tal que la solución del sistema (I.1) definida por (III.3) satisface

$$\|(u(T), u_t(T))\|_H \leq C(T) \|(u^0, u^1)\|_H \quad (\text{III.4})$$

(Puede consultarse su demostración en el Apéndice, en [3] o en [5]).

Este resultado significa que con los datos iniciales elegidos, la energía inicial del sistema, dada por el segundo miembro de la desigualdad salvo la constante $C(T)$, tiende a dispersarse a medida que el control actúa, atenuando el movimiento.

¹Tener en cuenta que fuera del rectángulo no hay movimiento.

²En relación con estos resultados aparecen espacios funcionales que son necesarios para llevar a cabo las demostraciones. Profundizar en ello escapa del todo al objetivo de este proyecto, aunque se mencionan por rigor matemático. Puede consultarse cualquier información acerca de estos espacios en cualquier libro clásico (ver, por ejemplo [1]).

Ahora estamos en disposición de demostrar nuestro principal resultado relativo a la controlabilidad. Se incluye además su demostración en esta sección, aunque puede encontrarse en [5], porque resulta constructiva e ilustra el algoritmo numérico que se propone en consecuencia.

Teorema 1 - Sea $T > T_0 = \frac{2\sqrt{2}}{c}$ tal que $C(T) < 1$. Entonces, para cualquier par de datos iniciales $(u^0, u^1) \in H$ existe una función de control $f \in L^2(\Gamma_1 \times (0, T))$ tal que la solución del sistema (I.1) satisface la condición de controlabilidad exacta (I.2).

Dem.- Sean $(v^0, v^1) \in L^2(R_1) \times H^{-1}(R_1)$ y consideremos los sistemas

$$\begin{cases} v_{tt} - c^2 \Delta v = 0, & \text{en } \Omega \times (0, T) \\ v = 0, & \text{en } \Gamma_0 \times [0, T] \\ v = g, & \text{en } \Gamma_1 \times [0, T] \\ v(0) = v^0, v_t(0) = v^1 & \text{en } \Omega, \end{cases} \quad (\text{III.5})$$

y

$$\begin{cases} w_{tt} - c^2 \Delta w = 0, & \text{en } \Omega \times (0, T) \\ w = 0, & \text{en } \Gamma_0 \times [0, T] \\ w = h, & \text{en } \Gamma_1 \times [0, T] \\ w(0) = -v(T), w_t(0) = v_t(T) & \text{en } \Omega \end{cases} \quad (\text{III.6})$$

donde las funciones g y h se obtienen, respectivamente, como las restricciones a $\Gamma_1 \times [0, T]$ de las correspondientes soluciones del sistema (II.2) con la apropiada extensión de sus datos iniciales como se describió anteriormente.

Entonces, está claro que la función $u(x, t) = v(x, t) + w(x, T - t)$ resuelve el sistema

$$\begin{cases} u_{tt} - c^2 \Delta u = 0, & \text{en } \Omega \times (0, T) \\ u = 0, & \text{en } \Gamma_0 \times [0, T] \\ u = f, & \text{en } \Gamma_1 \times [0, T] \\ u(0) = v^0 + w(T), u_t(0) = v^1 - w_t(T) & \text{en } \Omega, \end{cases} \quad (\text{III.7})$$

con

$$f(y, t) = g(y, t) + h(y, T - t), \quad (y, t) \in \Gamma_1 \times [0, T],$$

y satisface la condición de controlabilidad exacta (I.2). Esto nos da la condición de controlabilidad exacta deseada para el sistema (I.1), pero con los datos iniciales (u^0, u^1) en la forma

$$(u^0, u^1) = (v^0 + w(T), v^1 - w_t(T)). \quad (\text{III.8})$$

Así pues, sólo queda mostrar que para cualquier condición inicial dada $(u^0, u^1) \in L^2 \times H^{-1}$ existe $(v^0, v^1) \in L^2 \times H^{-1}$ tal que se satisface (III.8), o equivalentemente, que el operador lineal

$$\begin{aligned} L : L^2 \times H^{-1} &\rightarrow L^2 \times H^{-1} \\ (v^0, v^1) &\mapsto (v^0 + w(T), v^1 - w_t(T)) \end{aligned}$$

es sobreyectivo. Debemos descomponer $L = I - K$, donde el operador $K : L^2 \times H^{-1} \rightarrow L^2 \times H^{-1}$ está definido por

$$K(v^0, v^1) = (-w(T), w_t(T)).$$

Basta entonces con probar que $\|K\| < 1$ ya que en este caso L es invertible y su inverso está dado por

$$L^{-1} = I + K + K^2 + K^3 + \dots$$

Una doble aplicación del *lema 1* nos lleva a

$$\begin{aligned} \|K(v^0, v^1)\|_{L^2 \times H^{-1}} &= \|(-w(T), w_t(T))\|_{L^2 \times H^{-1}} \\ &\leq C(T) \|(-v(T), v_t(T))\|_{L^2 \times H^{-1}} \\ &\leq C(T)^2 \|(v^0, v^1)\|_{L^2 \times H^{-1}}, \end{aligned}$$

lo que completa nuestra demostración.

III.2. Algoritmo Numérico para el Cálculo del Control

Desde un punto de vista computacional, debemos aproximar el operador L^{-1} a cierto orden en K . Para el caso de una aproximación de primer orden de L^{-1} el algoritmo numérico queda estructurado como sigue:

Sean $(u^0, u^1) \in L^2 \times H^{-1}$ los datos iniciales dados y tomemos $T > T_0$ tal que $C(T) < 1$. Entonces:

a) Resolver el sistema (III.5) con los datos iniciales (u^0, u^1) , y después (III.6).

b) Definir $(v^0, v^1) = (u^0 - w(T), u^1 + w_t(T))$, y con estos datos iniciales, calcular $g(y, t)$ en el sistema (III.5) y $h(y, t)$ en (III.6). Una aproximación al control deseado es, entonces, $f(y, t) = g(y, t) + h(y, T - t)$. Y el estado se debe calcular numéricamente también como $u(x, t) = v(x, t) + w(x, T - t)$

Consideremos un ejemplo sencillo:

III.3. Simulaciones Numéricas

Como ejemplo para el caso bidimensional, resolvemos el problema de controlabilidad en la frontera de una membrana elástica plana de dimensiones 1×1 .

Sean

$$\begin{aligned} \Omega &\subset \mathbb{R}^2, \Omega = [0, 1] \times [0, 1] \\ \Gamma_1 &= \{x = (x_1, x_2) \in \mathbb{R}^2 / x_1 = 0 \text{ ó } x_2 = 0\} \\ \Gamma_2 &= \{x = (x_1, x_2) \in \mathbb{R}^2 / x_1 = 1 \text{ ó } x_2 = 1\} \end{aligned}$$

$$\begin{cases} u_{tt} - \Delta u = 0, & \text{en } \Omega \times (0, T) \\ u(x, 0) = 10 \sin(\pi x_1) \sin(\pi x_2), & \text{en } \Omega \times (0, T) \\ u(x, t) = 0, & \text{en } \Gamma_1 \times (0, T) \\ u(x, t) = f(x, t), & \text{en } \Gamma_2 \times (0, T) \end{cases} \quad (\text{III.9})$$

Habrá controlabilidad exacta al estado nulo en la frontera si, transcurrido un tiempo T , se cumple que la posición y la velocidad se anulan, es decir:

$$u(x, T) = u_t(x, T) = 0 \quad \text{en } \Omega$$

Para $T = 3,81$ la constante $C(T)$ del *lema 1* satisface la condición $C(T) < 1$.

Según vimos, la solución del sistema (III.9) viene dada por la fórmula de Poisson:

$$u(x, t) = \frac{1}{2\pi t} \int_{B(x,t)} \frac{g(y) + \nabla g(y)(y-x) + th(y)}{(t^2 - |y-x|^2)^{1/2}} dy \quad (\text{III.10})$$

En el ejemplo que nos ocupa

$$g(y) = 10 \sin(\pi y_1) \sin(\pi y_2) \quad (\text{III.11})$$

$$h(y) = 0.$$

Por tanto, se trata de calcular dos integrales:

$$u(x, t) = I_1 + I_2$$

$$I_1 = \frac{1}{2\pi t} \int_{B(x,t)} \frac{g(y)}{(t^2 - |y-x|^2)^{1/2}} dy \quad (\text{III.12})$$

$$I_2 = \frac{1}{2\pi t} \int_{B(x,t)} \frac{\nabla g(y)(y-x)}{(t^2 - |y-x|^2)^{1/2}} dy \quad (\text{III.13})$$

En las siguientes líneas simplificaremos lo máximo posible estas expresiones para introducirlas en el código. (ver Apéndice, programa *ondas_control2d_1C.m*).

Comencemos con la primera integral. Sustituimos $g(y)$ por (III.11) y efectuamos el cambio de variable (V.27), quedando

$$\begin{aligned} I_1 &= \frac{1}{2\pi t} \int_{B(0,1)} \frac{10 \sin(\pi(x_1 + tz_1)) \sin(\pi(x_2 + tz_2))}{t(1 - |z|^2)^{1/2}} t^2 dz \\ &= \frac{10}{2\pi} \int_{B(0,1)} \frac{\sin(\pi(x_1 + tz_1)) \sin(\pi(x_2 + tz_2))}{(1 - |z|^2)^{1/2}} dz \end{aligned}$$

Cambiamos ahora a coordenadas polares, mediante el cambio de variable

$$\begin{aligned} z_1 &= \rho \cos \theta \\ z_2 &= \rho \sin \theta \end{aligned} \quad (\text{III.14})$$

Sustituyendo,

$$I_1 = \frac{5}{\pi} \int_0^{2\pi} \int_0^1 \frac{\sin(\pi(x_1 + t\rho \cos \theta)) \sin(\pi(x_2 + t\rho \sin \theta))}{(1 - \rho^2)^{1/2}} \rho d\rho d\theta$$

Por último, un nuevo cambio de variable

$$\begin{aligned} 1 - \rho^2 &= \frac{1}{r}; \quad \rho = \sqrt{1 - \frac{1}{r}} \\ 2\rho d\rho &= \frac{1}{r^2} dr \end{aligned} \quad (\text{III.15})$$

nos lleva a

$$\begin{aligned}
I_1 &= \frac{10}{2\pi} \int_0^{2\pi} \int_1^\infty \frac{\sin(\pi s_1) \sin(\pi s_2)}{(r)^{-1/2}} \frac{1}{2r^2} dr d\theta = \\
&= \frac{1}{4\pi} \int_0^{2\pi} \int_1^\infty \frac{10 \sin(\pi(s_1)) \sin(\pi(s_2))}{(r)^{3/2}} dr d\theta = \\
&= \frac{1}{4\pi} \int_0^{2\pi} \int_1^\infty \frac{g(s_1, s_2)}{(r)^{3/2}} dr d\theta, \quad \text{con } \begin{cases} s_1 = x_1 + t\sqrt{1 - \frac{1}{r}} \cos \theta \\ s_2 = x_2 + t\sqrt{1 - \frac{1}{r}} \sin \theta \end{cases}
\end{aligned}$$

Para calcular la segunda integral, desarrollamos primero el interior del integrando:

$$\begin{aligned}
\nabla g(y) &= (10\pi \cos(\pi y_1) \sin(\pi y_2), 10\pi \sin(\pi y_1) \cos(\pi y_2)) \\
\nabla g(y)(y-x) &= (y_1 - x_1)10\pi \cos(\pi y_1) \sin(\pi y_2) + (y_2 - x_2)10\pi \sin(\pi y_1) \cos(\pi y_2) \\
&= 10\pi [(y_1 - x_1) \cos(\pi y_1) \sin(\pi y_2) + (y_2 - x_2) \sin(\pi y_1) \cos(\pi y_2)]
\end{aligned}$$

Sustituyendo en (III.13),

$$I_2 = \frac{1}{2\pi t} \int_{B(x,t)} \frac{10\pi ((y_1 - x_1) \cos(\pi y_1) \sin(\pi y_2) + (y_2 - x_2) \sin(\pi y_1) \cos(\pi y_2))}{(t^2 - |y - x|^2)^{1/2}} dy$$

Hacemos, igual que antes, el cambio (V.27),

$$\begin{aligned}
I_2 &= \frac{5}{t} \int_{B(0,1)} \frac{z_1 t \cos(\pi(x_1 + tz_1)) \sin(\pi(x_2 + tz_2))}{t(1 - |z|^2)^{1/2}} t^2 dz + \dots \\
&\dots + \frac{5}{t} \int_{B(0,1)} \frac{z_2 t \sin(\pi(x_1 + tz_1)) \cos(\pi(x_2 + tz_2))}{t(1 - |z|^2)^{1/2}} t^2 dz \\
&= 5t \int_{B(0,1)} \frac{z_1 \cos(\pi(x_1 + tz_1)) \sin(\pi(x_2 + tz_2))}{(1 - |z|^2)^{1/2}} dz + \dots \\
&\dots + 5t \int_{B(0,1)} \frac{z_2 \sin(\pi(x_1 + tz_1)) \cos(\pi(x_2 + tz_2))}{(1 - |z|^2)^{1/2}} dz
\end{aligned}$$

De la misma forma que para la primera integral, realizamos el cambio (III.14), para que quede

$$\begin{aligned}
I_2 &= 5t \int_0^{2\pi} \int_0^1 \frac{\rho \cos \theta [\cos(\pi s_1) \sin(\pi s_2)] + \rho \sin \theta [\sin(\pi s_1) \cos(\pi s_2)]}{(1 - \rho^2)^{1/2}} \rho d\rho d\theta \\
&\quad \text{con } \begin{cases} s_1 = x_1 + t\sqrt{1 - \frac{1}{r}} \cos \theta \\ s_2 = x_2 + t\sqrt{1 - \frac{1}{r}} \sin \theta \end{cases}
\end{aligned}$$

Y, por último, el cambio (III.15) nos lleva a

$$I_2 = \frac{5}{2}t \int_0^{2\pi} \int_0^1 \frac{\cos \theta [\cos(\pi s_1) \sin(\pi s_2)] + \sin \theta [\sin(\pi s_1) \cos(\pi s_2)]}{r^{3/2}} \sqrt{1 - \frac{1}{r}} dr d\theta$$

Introducimos en el código de MATLAB las integrales en esta forma, la más simplificada posible, para implementar el algoritmo y obtener el estado y el control que ilustramos a continuación. La resolución de las integrales se ha llevado a cabo mediante la regla de Simpson (ver Apéndice, programas *ondas_control2d.m* y *simpson_v.m*). La figura (III.1) muestra la animación de la solución y la figura (III.2) la animación del control en la arista $x = 1$.

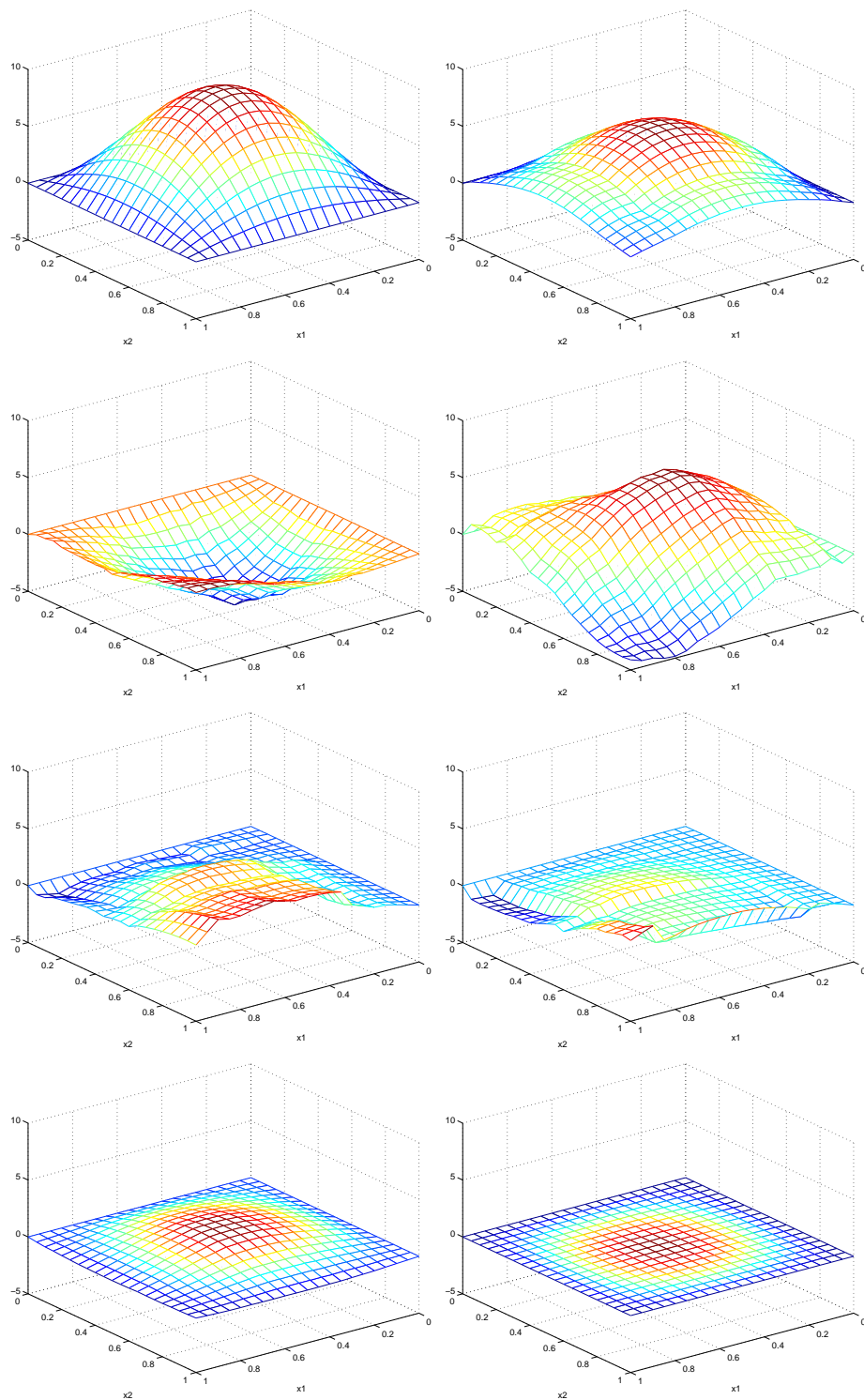


Figura III.1: animación del estado $u(x,t)$ de izquierda a derecha y de arriba hacia abajo para $t = 0, 0.2, 0.6, 1.4, 2, 2.8, 3.6$ y 4 .

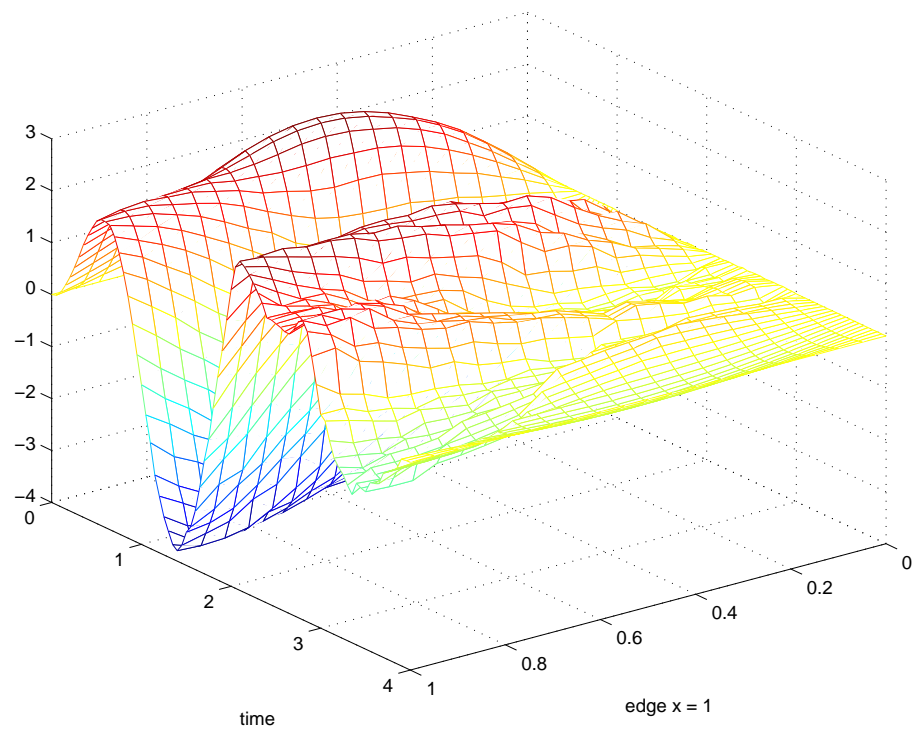


Figura III.2: representación del control en la arista $x = 1$ durante el intervalo de tiempo $0 \leq T \leq 4$.

CAPÍTULO IV

Conclusiones y Problemas Abiertos

Es de sobra conocido que el problema del cálculo numérico del control en la frontera para la ecuación de ondas es extremadamente sensible al esquema numérico usado. De hecho, para los esquemas de diferencias finitas y elementos finitos usados habitualmente, la convergencia falla.

En este trabajo hemos propuesto un algoritmo muy simple para resolver este problema. La convergencia del algoritmo es consecuencia del decaimiento local de energía de las soluciones de un problema de valor inicial asociado con el sistema original.

Aunque el ejemplo para el caso bidimensional estudiado en el capítulo 2 ha sido desarrollado con una aproximación de primer orden del operador L_T^{-1} , un incremento en una unidad, por ejemplo, en dicho orden de aproximación, sólo requiere una recurrencia en los pasos (a) y (b) del algoritmo numérico, que no incrementa significativamente el coste computacional.

La ecuación de ondas modeliza, además del movimiento de cuerdas y membranas elásticas, otros sistemas físicos como los movimientos torsionales y longitudinales de barras y de sólidos tridimensionales (acústica), por citar sólo algunos ejemplos. En el caso del movimiento de barras, en ingeniería, el problema de las vibraciones está siempre presente en ejes, vigas...El control de éstas podría ser de gran utilidad para, por ejemplo, alargar la vida de un eje que esté sometido a cargas torsionales o axiales.

Para finalizar, enumeraremos las principales ventajas e inconvenientes del método propuesto en este proyecto:

Ventajas:

- La principal ventaja del esquema numérico propuesto es su simplicidad. El problema de controlabilidad se resuelve con las mismas técnicas que se usan para problemas del valor inicial. De hecho, como vimos en el Capítulo 1, en algunos casos sencillos se pueden obtener fórmulas explícitas para el control y el estado. En estos casos, la convergencia del método está asegurada.
- El método parece ser muy general en el sentido de que se puede aplicar para

sistemas lineales de segundo orden y reversibles en el tiempo, para cualquier dimensión.

- La implementación numérica del algoritmo es sencilla y proporciona al mismo tiempo aproximaciones numéricas tanto para el control como para el estado.

Inconvenientes:

- Aunque para problemas de geometría sencilla se puede aplicar el método con controles actuando en una pequeña parte del contorno, para geometrías más complejas el método lleva a controles actuando en toda la frontera..
- En el método propuesto, la convergencia del algoritmo propuesto depende del tiempo para el cual se cumple la desigualdad (V.29) a través de la constante $C(T) < 1$. En algunos casos es difícil hallar esta constante y, en consecuencia, el tiempo mínimo de controlabilidad. Aún así, a nivel práctico su valor podría ser estimado numéricamente.

CAPÍTULO V

Apéndice

V.1. Interpretación física de la ecuación de ondas en el caso unidimensional

La ecuación de ondas para el caso $n = 1$ es la que rige las pequeñas vibraciones transversales de una cuerda vibrante de longitud l . Supóngase que la cuerda se deforma y entonces, en un determinado momento que consideramos $t = 0$, se suelta y se deja vibrar. El problema es determinar las vibraciones de la cuerda, es decir, hallar su deflexión $u(x, t)$ en cualquier punto x y en cualquier instante $t > 0$.

Para deducir la ecuación diferencial correspondiente al problema descrito, efectuaremos las siguientes simplificaciones:

- 1.- La masa de la cuerda por unidad de longitud es constante (cuerda homogénea). La cuerda es perfectamente elástica y no ofrece resistencia alguna a la flexión.
- 2.- La tensión causada al estirar la cuerda es tan grande que podemos despreciar la acción de la fuerza de la gravedad sobre la cuerda.
- 3.- La cuerda realiza un movimiento transversal pequeño en un plano vertical, es decir, toda partícula de la cuerda se mueve estrictamente en la dirección vertical, de modo que la deflexión y la pendiente en todo punto de ella toman valores pequeños.

Con el fin de obtener la ecuación diferencial se consideran las fuerzas que actúan sobre una porción de cuerda (V.1). Como esta no ofrece resistencia alguna a la flexión, la tensión es tangencial a la curva que describe la cuerda en cada punto. Sean T_1 y T_2 las tensiones en los puntos P y Q de esa porción. Puesto que no existe movimiento en la dirección horizontal, las componentes horizontales de la tensión deben ser constantes. Así, si se aplica la notación ilustrada en la figura se obtiene

$$T_1 \cos \alpha = T_2 \cos \beta = T = \text{const.} \quad (\text{V.1})$$

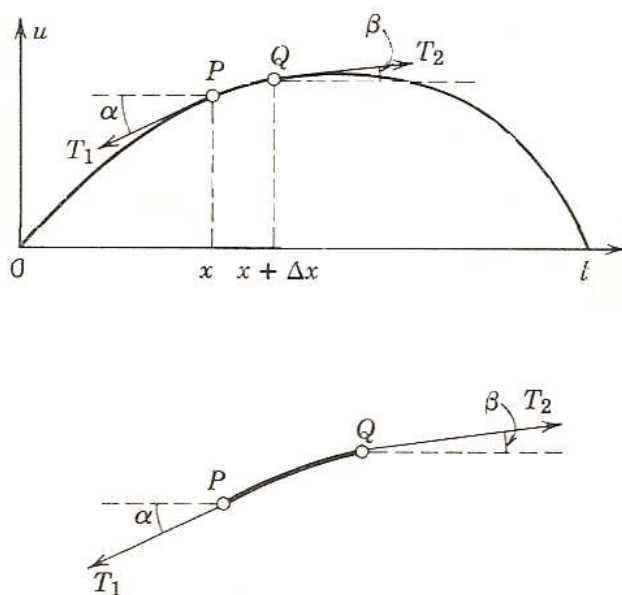


Figura V.1: cuerda vibrante

En la dirección vertical se tienen dos fuerzas, a saber, las componentes verticales de T_1 y T_2 : $-T_1 \sin \alpha$, $T_2 \sin \beta$; el signo negativo se debe a que la componente en P está dirigida hacia abajo. De acuerdo con la segunda ley de Newton, la resultante de estas dos fuerzas es igual a la masa $\rho \Delta x$ de la porción, multiplicada por la aceleración $\partial^2 u / \partial t^2$ que adquiere debido a esa fuerza, evaluada en algún punto entre x y $x + \Delta x$; aquí, ρ es la masa de la cuerda no desviada por unidad de longitud y Δx es la longitud de la cuerda no desviada. De donde

$$T_2 \sin \beta - T_1 \sin \alpha = \rho \Delta x \frac{\partial^2 u}{\partial t^2}$$

Al aplicar (V.1), se obtiene:

$$\frac{T_2 \sin \beta}{T_2 \cos \beta} - \frac{T_1 \sin \alpha}{T_1 \cos \alpha} = \tan \beta - \tan \alpha = \frac{\rho \Delta x}{T} \frac{\partial^2 u}{\partial t^2} \quad (\text{V.2})$$

Ahora bien, $\tan \alpha$ y $\tan \beta$ son las pendientes de la curva descrita por la cuerda x y $x + \Delta x$:

$$\tan \alpha = \left(\frac{\partial u}{\partial x} \right) \quad \text{y} \quad \tan \beta = \left(\frac{\partial u}{\partial x} \right)_{x+\Delta x}$$

Dividiendo (V.2) entre Δx se tiene:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad \text{con } c^2 = \frac{T}{\rho} \quad (\text{V.3})$$

Esta es la *ecuación unidimensional de onda*, la cual rige el problema planteado. Se ve que es homogénea de segundo orden. Se ha elegido la constante c^2 con el fin de indicar que tal constante es positiva.

V.2. Solución de D'Alembert de la ecuación de onda

Buscamos a continuación la solución de la ecuación (V.3) que propuso D'Alembert. Se trata de encontrar una solución para el problema siguiente:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0, & \text{en } \mathbb{R} \times (0, \infty) \\ u(x, 0) = f(x), & \text{en } \mathbb{R} \\ u_t(x, 0) = g(x), & \text{en } \mathbb{R} \end{cases} \quad (\text{V.4})$$

En primer lugar, transformamos la ecuación (V.3) realizando el siguiente cambio de variable:

$$v = x + ct, \quad z = x - ct \quad (\text{V.5})$$

Con este cambio, u se transforma en una función de v y z , y es posible expresar las derivadas que aparecen en (V.4) a través de derivadas respecto a v y z , aplicando la regla de la cadena:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial v} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial z} \frac{\partial z}{\partial x} = \frac{\partial u}{\partial v} + \frac{\partial u}{\partial z} = u_v + u_z$$

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= \frac{\partial^2 u}{\partial v^2} \frac{\partial v}{\partial x} + \frac{\partial^2 u}{\partial v \partial z} \frac{\partial z}{\partial x} + \frac{\partial^2 u}{\partial z \partial v} \frac{\partial v}{\partial x} + \frac{\partial^2 u}{\partial z^2} \frac{\partial z}{\partial x} = \\ &= \frac{\partial^2 u}{\partial v^2} + 2 \frac{\partial^2 u}{\partial v \partial z} + \frac{\partial^2 u}{\partial z^2} = u_{vv} + 2u_{vz} + u_{zz} \end{aligned}$$

Análogamente para la otra derivada:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial v^2} - 2 \frac{\partial^2 u}{\partial v \partial z} + \frac{\partial^2 u}{\partial z^2} \right) = c^2 (u_{vv} - 2u_{vz} + u_{zz})$$

Introduciendo estos resultados en (V.3) se obtiene

$$\frac{\partial^2 u}{\partial v \partial z} = u_{vz} = 0 \quad (\text{V.6})$$

Integrando respecto de z , queda:

$$\frac{\partial u}{\partial v} = h(v)$$

donde $h(v)$ es una función arbitraria de v . Integrando ahora respecto de v

$$u = \int h(v) dv + \psi(z)$$

donde $\psi(z)$ es una función arbitraria de z . Como la integral es una función de v , por ejemplo $\phi(v)$, podemos escribir la solución como

$$u(x, t) = \phi(x + ct) + \psi(x - ct) \quad (\text{V.7})$$

Para determinar las funciones $\phi(v)$ y $\psi(z)$, aplicamos las condiciones iniciales:

$$u(x, 0) = \phi(x) + \psi(x) = f(x) \quad (\text{V.8})$$

$$u_t(x, 0) = c\phi'(x) - c\psi'(x) = g(x)$$

$$\phi(x) - \psi(x) = \frac{1}{c} \int_a^x g(s) ds \quad (\text{V.9})$$

Sumando (V.8) y (V.9), y despejando $\phi(x)$ tenemos:

$$\phi(x) = \frac{1}{2} \left[f(x) + \frac{1}{c} \int_a^x g(s) ds \right] \quad (\text{V.10})$$

Sustituyendo la expresión (V.10) en (V.8) y despejando $\psi(x)$, queda:

$$\psi(x) = \frac{1}{2} \left[f(x) - \frac{1}{c} \int_a^x g(s) ds \right] \quad (\text{V.11})$$

Finalmente, sustituyendo las dos últimas expresiones en (V.7), obtenemos la solución

$$u(x, t) = \frac{1}{2} \left[f(x + ct) + f(x - ct) + \frac{1}{c} \int_{x-ct}^{x+ct} g(s) ds \right] \quad (\text{V.12})$$

que es la *solución de D'Alembert* del problema (V.4).

V.3. Interpretación física de la ecuación de ondas en el caso bidimensional

Otro problema importante dentro del campo de las vibraciones es el del movimiento de una membrana elástica tensa, como el parche de un tambor.

Para obtener la ecuación que modeliza este problema, se harán las suposiciones siguientes:

- 1.- La masa de la membrana por unidad de área es constante ("membrana homogénea"). La membrana es perfectamente delgada y tan flexible que no ofrece resistencia alguna a la flexión.
- 2.- La membrana se tensa y a continuación de fija a lo largo de toda su frontera, en el plano xy . La tensión por unidad de longitud, T , provocada al estirar la membrana, es la misma en todos los puntos y en todas direcciones, y no cambia durante el movimiento.
- 3.- La deflexión $u(x, y, t)$ de la membrana durante el movimiento es pequeña comparada con el tamaño de ésta y todos los ángulos de inclinación son pequeños.

Aún cuando en la práctica no se pueden cumplir esas hipótesis, las vibraciones transversales pequeñas de una membrana física delgada las satisfarán con relativa exactitud.

Para obtener la ecuación diferencial que rige el movimiento de la membrana, en la figura se consideran las fuerzas que actúan sobre una pequeña porción de la misma. Puesto que las deflexiones de la membrana y los ángulos de inclinación son pequeños, los lados de la porción son iguales a Δx y Δy . La tensión T es la

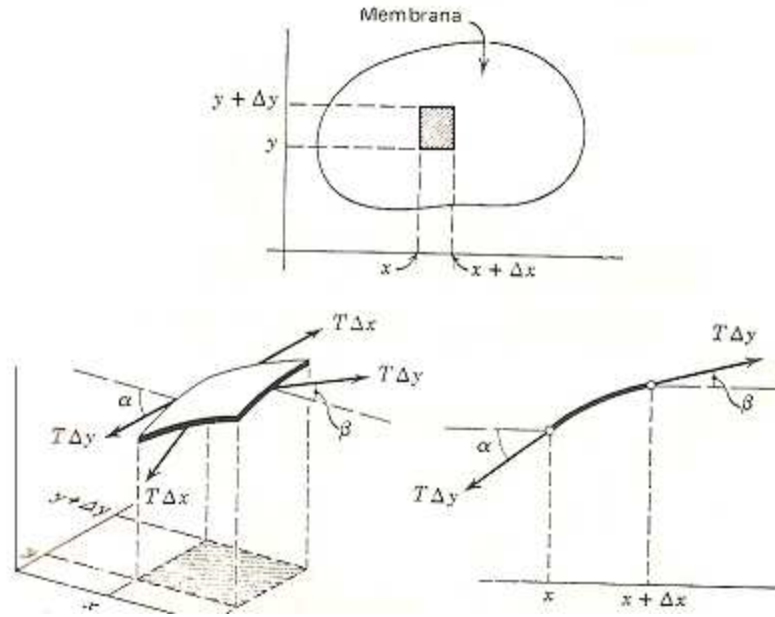


Figura V.2: membrana vibrante

fuerza por unidad de longitud, de donde las fuerzas que actúan sobre los bordes son aproximadamente $T\Delta x$ y $T\Delta y$. Dado que la membrana es perfectamente flexible, esas fuerzas son tangentes a ella.

Primero se consideran las componentes horizontales de las fuerzas. Se obtienen estas componentes al multiplicar las fuerzas por los cosenos de los ángulos de inclinación. Como estos ángulos son pequeños, sus cosenos son aproximadamente iguales a la unidad, y las componentes horizontales de las fuerzas en bordes opuestos son aproximadamente iguales. Por tanto, el movimiento de las partículas de la membrana en dirección horizontal será prácticamente despreciable, y consiguientemente, se puede considerar como un movimiento transversal.

Observando la figura (V.2), en la que miramos perpendicularmente al plano xz , vemos que las componentes verticales de las fuerzas a lo largo de los bordes son:

$$T\Delta y \sin \beta \text{ y } -T\Delta y \sin \alpha$$

Aquí aparece el signo menos porque la fuerza que actúa sobre el borde izquierdo está dirigida hacia abajo. Como los ángulos son pequeños, se pueden reemplazar sus senos por sus tangentes, y estas se pueden expresar a través de la derivada parcial de la posición, $u(x, y)$, respecto de x . Así, la resultante de esas dos componentes verticales es

$$T\Delta y(\sin \beta - \sin \alpha) \approx T\Delta y(\tan \beta - \tan \alpha) = T\Delta y [u_x(x + \Delta x, y_1) - u_x(x, y_2)]$$

donde el subíndice x denota la derivada parcial respecto de la variable x , e y_1 e y_2 son valores entre y e $y + \Delta y$.

De manera análoga, la resultante de las fuerzas que actúan sobre los otros dos bordes de la porción es

$$T\Delta x [u_y(x_1, y + \Delta y) - u_y(x_2, y)]$$

donde el subíndice y denota la derivada parcial respecto de la variable y , y x_1 y x_2 son valores comprendidos entre x y $x + \Delta x$.

De acuerdo con la segunda ley de Newton, la suma de las fuerzas que actúan sobre la porción es igual a su masa, $\rho\Delta A$, por la aceleración que adquiere, $\partial^2 u / \partial t^2$. Aquí, ρ es la masa de la porción no desviada por unidad de área y $\Delta A = \Delta x \Delta y$ es el área de esa misma porción cuando no está desviada. Por tanto,

$$\rho\Delta x\Delta y \frac{\partial^2 u}{\partial t^2} = T\Delta y [u_x(x + \Delta x, y_1) - u_x(x, y_2)] + T\Delta x [u_y(x_1, y + \Delta y) - u_y(x_2, y)]$$

donde la derivada de la izquierda se evalúa en algún punto apropiado (x, y) correspondiente a esa porción. Dividiendo por $\rho\Delta x\Delta y$,

$$\frac{\partial^2 u}{\partial t^2} = \frac{T}{\rho} \left[\frac{u_x(x + \Delta x, y_1) - u_x(x, y_2)}{\Delta x} + \frac{u_y(x_1, y + \Delta y) - u_y(x_2, y)}{\Delta y} \right]$$

Si hacemos tender Δx y Δy hacia cero, se obtiene la ecuación en derivadas parciales:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right]$$

o lo que es lo mismo

$$u_{tt} - c^2 \Delta u = 0, \quad \text{con } c^2 = \frac{T}{\rho} \quad (\text{V.13})$$

Ecuación que se conoce como **ecuación bidimensional de onda**.

V.4. Interpretación física de la ecuación de ondas en el caso tridimensional

Sea $U \subset \mathbb{R}^n$, un conjunto abierto. La incógnita $u : U \times [0, \infty) \rightarrow \mathbb{R}$, $u = u(x, t)$, con $x = (x_1, \dots, x_n)$.

Sea V una región contenida en U . La aceleración de esa región es

$$\frac{d^2}{dt^2} \int_V u dx = \int_V u_{tt} dx$$

Y la fuerza de contacto de la red es

$$- \int_{\partial V} F \cdot \nu dS$$

Donde F denota la fuerza que actúa sobre V a través de ∂V y la densidad de masa se ha tomado igual a la unidad. La ley de Newton asegura que dicha fuerza es igual a la masa de la región V por la aceleración que adquiere:

$$\int_V u_{tt} dx = - \int_{\partial V} F \cdot \nu dS$$

Esta identidad se obtiene para cada subregión V , y así

$$u_{tt} = -\operatorname{div} F$$

Para cuerpos elásticos, F es una función del gradiente del desplazamiento ∇u ; así,

$$u_{tt} + \operatorname{div} (\nabla u) = 0$$

Para pequeños gradientes de desplazamiento, la linealización $F(\nabla u) \approx -a\nabla u$ es a menudo apropiada, deforma que podemos escribir:

$$u_{tt} - a\nabla u = 0$$

Que es la ecuación de ondas para $a = c^2$

V.5. Solución de la ecuación de ondas para los casos $n = 2$ y $n = 3$

Tenemos el problema

$$\begin{cases} u_{tt} - \Delta u = 0 & \text{en } \mathbb{R}^n \times (0, \infty) \\ u = g, u_t = h & \text{en } \mathbb{R}^n \times \{t = 0\} \end{cases} \quad (\text{V.14})$$

El objetivo es encontrar una formula explícita para u en términos de g, h . El proceso será primero estudiar el valor medio de u en ciertas esferas. Esos valores, tomados como funciones del tiempo t y el radio r , resultan ser soluciones de la ecuación de Euler-Poisson-Darboux, una ecuación en derivadas parciales que se puede transformar en la ecuación de ondas unidimensional, cuya solución viene dada por la fórmula de D'Alembert.

Notación.- (i) Sea $x \in \mathbb{R}^n, t > 0, r > 0$. Definimos

$$U(x; r, t) = \overline{\int_{\partial B(x,r)} u(y, t) dS(y)} = \frac{1}{\text{Area de la superficie}} \int_{\partial B(x,r)} u(y, t) dS(y), \quad (\text{V.15})$$

como el valor medio de $u(\cdot, t)$ sobre la esfera $\partial B(x, r)$

(ii) De forma similar,

$$\begin{cases} G(x, r) = \overline{\int_{\partial B(x,r)} g(y) dS(y)} \\ H(x, r) = \overline{\int_{\partial B(x,r)} h(y) dS(y)} \end{cases} \quad (\text{V.16})$$

Para un punto x dado, en adelante consideraremos U como una función de r y t . Esta función resuelve la ecuación de Euler-Poisson-Darboux:

Lema 2.- (Ecuación de Euler-Poisson-Darboux) (ver [1]). Sea $x \in \mathbb{R}^n$, y u la solución del problema (V.14). Entonces, $U \in \mathbb{R}^m(\mathbb{R}_+ \times [0, \infty))$ y

$$\begin{cases} U_{tt} - U_{rr} - \frac{n-1}{r}U_r = o & \text{en } \mathbb{R}_+ \times (0, \infty) \\ U = G, U_t = H & \text{en } \mathbb{R}_+ \times \{t = 0\} \end{cases} \quad (\text{V.17})$$

La ecuación en derivadas parciales que aparece en (V.17) es la ecuación de Euler-Poisson-Darboux (nótese que el término $U_{rr} + \frac{n-1}{r}U_r$ es la parte radial del laplaciano Δ en coordenadas polares).

El plan global en las secciones subsiguientes será transformar la ecuación de Euler-Poisson-Darboux en la ecuación unidimensional de ondas, para los casos $n = 3$ y $n = 2$.

V.5.1. Solución para $n = 3$

Tomemos por tanto de ahora en adelante $n = 3$ y supongamos que $u \in C^2(\mathbb{R}^3 \times [0, \infty))$ resuelve el problema del valor inicial (V.14). Hacemos uso de las definiciones (V.15) y (V.16) de U , G y H hechas anteriormente, y definimos

$$\tilde{U} = rU, \quad \tilde{G} = rG, \quad \tilde{H} = rH. \quad (\text{V.18})$$

Podemos afirmar, por el lema 2 que

$$\begin{cases} \tilde{U}_{tt} - \tilde{U}_{rr} = 0 & \text{en } \mathbb{R}_+ \times (0, \infty) \\ \tilde{U} = \tilde{G}, \tilde{U}_t = \tilde{H} & \text{en } \mathbb{R}_+ \times (0, \infty) \\ \tilde{U} = o & \text{en } \{r = 0\} \times (0, \infty) \end{cases} \quad (\text{V.19})$$

Así,

$$\begin{aligned} \tilde{U}_{tt} &= rU_{tt} \\ &= r \left[U_{rr} + \frac{2}{r}U_r \right] \quad (\text{por (V.17), con } n = 3) \\ &= rU_{rr} + 2U_r = (U + rU_r)_r \\ &= \tilde{U}_{rr}. \end{aligned}$$

Aplicando la fórmula de D'Alembert (V.12) al sistema (V.19) encontramos, para $0 < r < t$

$$\tilde{U}(x; r, t) = \frac{1}{2} \left[\tilde{G}(t+r) - \tilde{G}(t-r) \right] + \frac{1}{2} \int_{t-r}^{t+r} \tilde{H}(y) dy. \quad (\text{V.20})$$

Puesto que (V.15) implica que $u(x, t) = \lim_{r \rightarrow 0^+} U(x; r, t)$, concluimos, de (V.18) y (V.20) que

$$\begin{aligned} u(x, t) &= \lim_{r \rightarrow 0^+} \frac{U(x; r, t)}{r} \\ &= \lim_{r \rightarrow 0^+} \left[\frac{\tilde{G}(t+r) - \tilde{G}(t-r)}{2r} + \frac{1}{2r} \int_{t-r}^{t+r} \tilde{H}(y) dy \right] \\ &= \tilde{G}'(t) + \tilde{H}(t). \end{aligned}$$

Usando las expresiones (V.16), se deduce

$$u(x, t) = \frac{\partial}{\partial t} \left(t \int_{\partial B(x,t)} g dS \right) + t \int_{\partial B(x,t)} h dS \quad (\text{V.21})$$

Pero

$$\int_{\partial B(x,t)} g(y) dS(y) = \int_{\partial B(0,1)} g(x + tz) dS(z);$$

y, por tanto,

$$\frac{\partial}{\partial t} \int_{\partial B(x,t)} g dS = \int_{\partial B(0,1)} \nabla g(x + tz) \cdot z dS(z) = \int_{\partial B(x,t)} \nabla g(y) \left(\frac{y - x}{t} \right) dS(y).$$

Volviendo a (V.21), concluimos:

$$u(x, t) = \int_{\partial B(x,t)} [th(y) + g(y) + \nabla g(y)(y - x)] dS(y) \quad (x \in \mathbb{R}^3, t > 0) \quad (\text{V.22})$$

Esta es la *fórmula de Kirchoff* que da la solución del problema del valor inicial (V.14) en dimensión 3.

V.5.2. Solución para $n = 2$

Para el caso bidimensional no podemos hacer ninguna transformación como la efetuada en (V.18) que convierta la ecuación de Euler-Poisson-Darboux en la ecuación de ondas. En lugar de ello, usaremos el *método de descenso*, es decir, tomaremos el problema del valor inicial (V.14) para $n = 3$ y lo trataremos como tal, pero con la tercera variable espacial igual a cero ($x_3 = 0$).

De esta forma, asumiendo que $u \in C^2(\mathbb{R}^2 \times [0, \infty))$ es solución de (V.14) para $n = 2$, escribamos

$$\bar{u}(x_1, x_2, x_3, t) = u(x_1, x_2, t) \quad (\text{V.23})$$

Entonces, (V.14) implica

$$\begin{cases} \bar{u}_{tt} - \Delta \bar{u} = 0 & \text{en } \mathbb{R}^3 \times (0, \infty) \\ \bar{u} = \bar{g}, \quad \bar{u}_t = \bar{h} & \text{en } \mathbb{R}^3 \times \{t = 0\} \end{cases} \quad (\text{V.24})$$

Para

$$\bar{g}(x_1, x_2, x_3) = g(x_1, x_2), \quad \bar{h}(x_1, x_2, x_3) = h(x_1, x_2)$$

Si escribimos $x = (x_1, x_2) \in \mathbb{R}^2$ y $\bar{x} = (x_1, x_2, 0) \in \mathbb{R}^3$, entonces (V.24) y la fórmula de Kirchoff (en la forma (V.21)) implican:

$$u(x, t) = \bar{u}(\bar{x}, t) = \frac{\partial}{\partial t} \left(t \int_{\partial \bar{B}(\bar{x}, t)} \bar{g} d\bar{S} \right) + t \int_{\partial \bar{B}(\bar{x}, t)} \bar{h} d\bar{S}, \quad (\text{V.25})$$

donde $\bar{B}(\bar{x}, t)$ denota la bola en \mathbb{R}^3 con centro en \bar{x} y radio $t > 0$. Podemos simplificar la expresión (V.25) aplicando la definición de integral de superficie:

$$\overline{\int}_{\partial\overline{B}(\overline{x},t)} \overline{g}d\overline{S} = \frac{1}{4\pi t^2} \int_{\partial\overline{B}(\overline{x},t)} \overline{g}d\overline{S} = \frac{2}{4\pi t^2} \int_{B(x,t)} g(y)(1 + |\nabla\gamma(y)|^2)^{1/2} dy,$$

donde $\gamma(y) = (t^2 - |y - x|^2)^{1/2}$ para $y \in B(x,t)$. El factor "2" aparece porque $\partial\overline{B}(\overline{x},t)$ consiste en 2 semiesferas. Si desarrollamos la expresión $(1 + |\nabla\gamma(y)|^2)^{1/2}$ a la vista de la definición que hemos dado de la función $\gamma(y)$, llegamos a la siguiente igualdad:

$$\overline{\int}_{\partial\overline{B}(\overline{x},t)} \overline{g}d\overline{S} = \frac{1}{2\pi t} \int_{B(x,t)} \frac{g(y)}{(t^2 - |y - x|^2)^{1/2}} dy = \frac{t}{2} \overline{\int}_{B(x,t)} \frac{g(y)}{(t^2 - |y - x|^2)^{1/2}} dy$$

En consecuencia, la fórmula (V.25) queda:

$$u(x,t) = \frac{1}{2} \frac{\partial}{\partial t} \left(t^2 \overline{\int}_{B(x,t)} \frac{g(y)}{(t^2 - |y - x|^2)^{1/2}} dy \right) + \frac{t^2}{2} \overline{\int}_{B(x,t)} \frac{h(y)}{(t^2 - |y - x|^2)^{1/2}} dy \quad (\text{V.26})$$

Para derivar respecto del tiempo, hacemos primero un cambio de variable que simplifique el cálculo:

$$\begin{aligned} y &= x + tz \\ dy &= t dz \end{aligned} \quad (\text{V.27})$$

Así,

$$\begin{cases} y = x \Rightarrow z = 0 \\ y = x + t \Rightarrow z = 1 \end{cases}$$

Haciendo ese cambio, tenemos:

$$t^2 \overline{\int}_{B(x,t)} \frac{g(y)}{(t^2 - |y - x|^2)^{1/2}} dy = t \overline{\int}_{B(0,1)} \frac{g(x + tz)}{(1 - |z|^2)^{1/2}} dz$$

Derivando,

$$\begin{aligned} \frac{1}{2} \frac{\partial}{\partial t} \left(t \overline{\int}_{B(0,1)} \frac{g(x + tz)}{(1 - |z|^2)^{1/2}} dz \right) &= \frac{1}{2} \left(\overline{\int}_{B(0,1)} \frac{g(x + tz)}{(1 - |z|^2)^{1/2}} dz + t \overline{\int}_{B(0,1)} \frac{z \nabla g(x + tz)}{(1 - |z|^2)^{1/2}} dz \right) dy \\ &= \frac{t}{2} \overline{\int}_{B(0,1)} \left(\frac{g(y)}{(t^2 - |y - x|^2)^{1/2}} + \frac{(y - x) \nabla g(y)}{(t^2 - |y - x|^2)^{1/2}} \right) dy \end{aligned}$$

Por último, sustituyendo en (V.26), obtenemos la relación

$$\begin{aligned}
u(x, t) &= \frac{1}{2} \int_{B(x, t)} \frac{tg(y) + t\nabla g(y)(y-x) + t^2h(y)}{(t^2 - |y-x|^2)^{1/2}} dy \\
&= \frac{1}{2\pi t} \int_{B(x, t)} \frac{g(y) + \nabla g(y)(y-x) + th(y)}{(t^2 - |y-x|^2)^{1/2}} dy, \tag{V.28}
\end{aligned}$$

para $x \in \mathbb{R}^2, t > 0$

que es la *Fórmula de Poisson*, solución del problema del valor inicial (V.14).

V.6. Demostración *Lema 1*

Lema 1.- Sean $(u_0, u_1) \in H$ y $T > T_0 = \frac{2\sqrt{2}}{c}$. Existe una constante positiva $C(T)$, con $C(T) \rightarrow 0$ cuando $T \rightarrow \infty$, tal que la solución del sistema (I.1) definida por (III.3) satisface

$$\|(u(T), u_t(T))\|_H \leq C(T) \|(u^0, u^1)\|_H \tag{V.29}$$

Dem.- Dado que el espacio $C_0^\infty(R_1)$ de funciones C^∞ que tienen soporte compacto en R_1 es denso tanto en L^2 como en H^1 , y gracias a la continuidad de la ecuación de ondas respecto a los datos iniciales, basta con probar el resultado para $(u^0, u^1) \in C_0^\infty(R_1) \times C_0^\infty(R_1)$.

Para $T > T_0$ y $x \in R_1$, puesto que (U^0, U^1) son regulares y se anulan fuera de R , usamos la fórmula (III.2) para escribir

$$u(x, T) = \frac{1}{2\pi c} \frac{\partial}{\partial T} \int_R \frac{U^0(y) dy}{\sqrt{c^2 T^2 - |y-x|^2}} + \frac{1}{2\pi c} \int_R \frac{U^1(y) dy}{\sqrt{c^2 T^2 - |y-x|^2}}$$

y, derivando el primer término

$$\begin{aligned}
u(x, T) &= -\frac{c^2}{2\pi c} \int_R T (c^2 T^2 - |y-x|^2)^{-3/2} U^0(y) dy \\
&\quad + \frac{1}{2\pi c} \int_R (c^2 T^2 - |y-x|^2)^{-1/2} U^1(y) dy. \tag{V.30}
\end{aligned}$$

De la desigualdad

$$\frac{1}{c^2 T^2 - |y-x|^2} \leq \frac{1}{c^2 (T^2 - T_0^2)}, \quad x \in R_1, \quad y \in R,$$

y de la desigualdad de Cauchy-Schwartz, se deduce que

$$|u(x, T)| \leq \frac{2}{\pi c^2} T (T^2 - T_0^2)^{-3/2} \|u^0\|_{L^2(R_1)} + \frac{2}{\pi c^2} (T^2 - T_0^2)^{-1/2} \|u^1\|_{L^2(R_1)}$$

Por tanto,

$$\|u(\cdot, T)\|_{L^2(R_1)}^2 \leq \frac{8}{\pi^2 c^4} T^2 (T^2 - T_0^2)^{-3} \|u^0\|_{L^2(R_1)}^2 + \frac{8}{\pi^2 c^4} (T^2 - T_0^2)^{-1} \|u^1\|_{L^2(R_1)}^2.$$

Por otra parte,

$$u_t(x, T) = -\frac{c^2}{2\pi c} \int_R \left[(c^2 T^2 - |y-x|^2)^{-3/2} - 3c^2 T^2 (c^2 T^2 - |y-x|^2)^{-5/2} \right] U^0(y) dy \\ - \frac{c}{2\pi} \int_R T (c^2 T^2 - |y-x|^2)^{-3/2} U^1(y) dy.$$

y de ahí,

$$|u_t(x, T)| \leq \frac{2}{\pi c^2} \left((T^2 - T_0^2)^{-3/2} + 3T^2 (T^2 - T_0^2)^{-5/2} \right) \|u^0\|_{L^2(R_1)} \\ + \frac{2}{\pi c^2} T (T^2 - T_0^2)^{-3/2} \|u^1\|_{L^2(R_1)}.$$

Teniendo en cuenta que, puesto que $u_t(\cdot, T) \in L^2$, podemos identificar su norma en H^{-1} con la norma en L^2 , y razonando igual que antes, tenemos

$$\|u_t(\cdot, T)\|_{H^{-1}(R_1)}^2 \leq \frac{8}{\pi^2 c^4} \left((T^2 - T_0^2)^{-3} + 9T^4 (T^2 - T_0^2)^{-5} + 6T^2 (T^2 - T_0^2)^{-4} \right) \|u^0\|_{L^2(R_1)}^2 \\ + \frac{8}{\pi^2 c^4} T^2 (T^2 - T_0^2)^{-3} \|u^1\|_{H^{-1}(R_1)}^2.$$

Finalmente,

$$\|(u(T), u_t(T))\|_{L^2 \times H^{-1}}^2 = \|u(\cdot, T)\|_{L^2(R_1)}^2 + \|u_t(\cdot, T)\|_{H^{-1}(R_1)}^2 \\ \leq \max\{C_1(T), C_2(T)\} \left(\|u^0\|_{L^2(R_1)}^2 + \|u^1\|_{H^{-1}(R_1)}^2 \right)$$

donde

$$C_1(T) = \frac{8}{\pi^2 c^4} \left(\frac{T^2 + 1}{(T^2 - T_0^2)^3} + \frac{9T^4}{(T^2 - T_0^2)^5} + \frac{6T^2}{(T^2 - T_0^2)^4} \right) \\ C_2(T) = \frac{8}{\pi^2 c^4} \left(\frac{1}{T^2 - T_0^2} + \frac{T^2}{(T^2 - T_0^2)^3} \right)$$

La constante $C(T)$ viene dada por

$$C(T) = \sqrt{\max\{C_1(T), C_2(T)\}}.$$

V.7. Códigos en MATLAB

V.7.1. wave_control_1d.m

```
*****
%wave_control_1d.m calcula (usando la formula de D'Alembert)
%el estado u(x,t) del problema de controlabilidad del ejemplo II.1
%Sintaxis: C=wave_control_1d(c,L,'pos_inicial','velo_inicial',n)
%c,L=parametros de la ecuacion
%pos_inicial,velo_inicial=posicion y velocidad iniciales
%n=numero de subdivisiones del intervalo (0,L) en la integracion numerica
*****
```

```

function C=wave_control_1d(c,L,n)
T=2*L/c;
pt=(10*T)+1;
px=(100*L)+1;
x=0:0.01:L;
t=0:0.1:T;
%Extension de los datos
for j=1:pt
for i=1:px
r(i,j)=x(i)-c*t(j); s(i,j)=x(i)+c*t(j);
p(i,j)=(s(i,j)-r(i,j))/n;
for k=1:n+1
w(i,j,k)=r(i,j)+(k-1)*p(i,j);
g(i,j,k)=feval('veloinieje4',L,w(i,j,k));
end
y(i,j)=0.5*(posinieje4(L,r(i,j))+posinieje4(L,s(i,j)));
z(i,j)=(1/(2*c))*simps_v(g(i,j,:),p(i,j));
u(i,j)=y(i,j)+z(i,j);
end
h=plot(x,u(:,j),'r-');
axis([0 1 -1.2 1.2]);
M(:,j) = getframe;
end
movie2avi(M,'estado_eje11', 'compression', 'Cinepak');%, 'colormap', cmap);

```

V.7.2. delta_wave_control_1d.m

```

*****
%wave_control_1d.m calcula (usando la formula de D'Alembert) el estado u(x,t)
del problema de controlabilidad del ejemplo II.2
%Sintaxis: C=wave_control_1d(c,L,'pos_inicial','velo_inicial',n)
%c,L=parametros de la ecuacion
*****
function C=delta_wave_control_1d(c,L)
T=2*L/c;
pt=(10*T)+1;
px=(100*L)+1;
x=0:0.01:L;
t=0:0.1:T;
t=2/7.;
for j=1:pt
for i=1:px
r(i,j)=x(i)-c*t(j); s(i,j)=x(i)+c*t(j);
y(i,j)=(1/(2*c))*(veloinidelta(L,s(i,j))-veloinidelta(L,r(i,j)));
end
h=plot(x,y(:,j),'r-');
axis([0 1 -0.5 1]);
xlabel('x');

```



```
M(:,j) = getframe;
end
movie(M,1,1);
```

V.7.3. Simpson_v.m

```
*****
%simps_v.m integra una funcion mediante la regla de Simpson extendida
% Sintaxis: I=simps_v(f,h)
%
%f= datos funcionales de abscisas equiespaciadas
%h= tamaño de intervalo de las abscisas
*****

function I=simps_v(f,h)
n=length(f)-1;
if n==1,
fprintf('Los datos tienen solo un intervalo'), return;
end
if n==2,...
I=h/3*(f(1)+4*f(2)+f(3));
return;end
if n==3,...
I=3/8*h*(f(1)+3*f(2)+3*f(3)+f(4));
return;end
I=0;
if 2*floor(n/2)~=n,
I=3/8*h*(f(n-2)+3*f(n-1)+3*f(n)+f(n+1));
m=n-3
else
m=n;
end
I=I+(h/3)*(f(1)+4*sum(f(2:2:m))+f(m+1));
if m>2, I=I+(h/3)*2*sum(f(3:2:m));
end
end
```

V.7.4. ondas_control.m

```
*****
%ondas_control2d.m anima la solución del problema
%de control2d y calcula el control en la arista x=1 para la
%ecuación de ondas % mediante Fubini y la regla de Simpson extendida
%y utiliza la fórmula de Poisson para el control de las ondas 2-d
% Sintaxis: S=ondas_control2d('f_name','g_name',T,f,a,b,c,d,n,m)
%f_name= nombre del archivo m del integrando no gradiente (pos_integra1)
%g_name=nombre del archivo m del integrando con gradiente (pos_integra2)
%hemos tomado velocidad inicial nula
%T=tiempo del proceso de estabilización
```

```

%f=numero de frames por animacion
%a,b=intervalo de integracion para la variable polar de integracion 's'
%c,d=intervalo de integracion para la variable polar de integracion 'z'
%n=numero de subdivisiones del intervalo (a,b)
%m=numero de subdivisiones del intervalo (c,d)
*****
function [control,sol,sol_]=ondas_control2d(f_name,g_name,T,f,a,b,c,d,n,m)
npt = n+1; mpt = m+1;
hs = (b-a)/n; %Tamaño de paso para s
s = a+(0:n)*hs;
hz = (d-c)/m; %Tamaño de paso para z
z = c+(0:m)*hz;
h1 = 0.05; %tamaño de paso para x1, x2
h2 = T/f; %tamaño de paso del tiempo
x = [0:h1:1]; y = x; %malla en el rectangulo grande
paso = 0.01; %paso para el calculo de las derivadas en tiempo finales
nxy = 1/h1+1;
%Resolvemos el primer sistema
for k=1:length(x)
for l=1:length(y)
for i=1:npt
for j=1:mpt
f1(k,l,i,j) = feval(f_name,T,x(k),y(l),s(i),z(j));
g1(k,l,i,j) = feval(g_name,T,x(k),y(l),s(i),z(j));
f2(k,l,i,j) = feval(f_name,T-paso,x(k),y(l),s(i),z(j));
g2(k,l,i,j) = feval(g_name,T-paso,x(k),y(l),s(i),z(j));
end
F1(k,l,i) = simps_v(f1(k,l,i,:),hz);
G1(k,l,i) = simps_v(g1(k,l,i,:),hz);
F2(k,l,i) = simps_v(f2(k,l,i,:),hz);
G2(k,l,i) = simps_v(g2(k,l,i,:),hz);
end
I1(k,l) = simps_v(F1(k,l,:),hs);
J1(k,l) = simps_v(G1(k,l,:),hs);
S1(k,l) = I1(k,l)+J1(k,l); %posicion en el instante T
I2(k,l) = simps_v(F2(k,l,:),hs);
J2(k,l) = simps_v(G2(k,l,:),hs);
S2(k,l) = I2(k,l)+J2(k,l); %posicion en el instante T-paso para calcular la veloci-
dad en T
DS(k,l) = (S1(k,l)-S2(k,l))/paso; %velocidad en el instante T
end
end
%Gradiente de la posicion
[z0X_1C,z0Y_1C] = gradient(S1,h1);
%Extension par e impar de la posicion ,la velocidad y el gradiente
for k=1:length(x)
for l=1:length(y)
z0(nxy-1+k,nxy-1+l) = -S1(k,l); primer cuadrante 11x11 en adelante

```

```

z0(nxy+1-k,nxy+1-l) = -S1(k,l); tercer cuadrante 1x1 en adelante
z0(nxy+1-k,nxy-1+l) = S1(k,l); segundo cuadrante 1x1 en adelante
z0(nxy-1+k,nxy+1-l) = S1(k,l); cuarto cuadrante 1x1 en adelante
zt0(nxy-1+k,nxy-1+l) = DS(k,l);
zt0(nxy+1-k,nxy+1-l) = DS(k,l);
zt0(nxy+1-k,nxy-1+l) = -DS(k,l);
zt0(nxy-1+k,nxy+1-l) = -DS(k,l);
z0X(nxy-1+k,nxy-1+l) = z0X_1C(k,l);
z0X(nxy+1-k,nxy+1-l) = -z0X_1C(k,l);
z0X(nxy+1-k,nxy-1+l) = -z0X_1C(k,l);
z0X(nxy-1+k,nxy+1-l) = z0X_1C(k,l);
z0Y(nxy-1+k,nxy-1+l) = z0Y_1C(k,l);
z0Y(nxy+1-k,nxy+1-l) = -z0Y_1C(k,l);
z0Y(nxy+1-k,nxy-1+l) = z0Y_1C(k,l);
z0Y(nxy-1+k,nxy+1-l) = -z0Y_1C(k,l);
end
end
z0(nxy,:) = 0;
z0(:,nxy) = 0;
zt0(nxy,:) = 0;
zt0(:,nxy) = 0;
z0X(nxy,:) = 0;
z0X(:,nxy) = 0;
z0Y(nxy,:) = 0;
z0Y(:,nxy) = 0;
*****
X=[0:h1:1];
Y=[0:h1:1];
[X,Y]=meshgrid(X,Y);
figure(1);
mesh(X,Y,S1);
axis([0,1,0,1,-5,10]);
view(-217,30);
figure(2)
mesh(X,Y,DS);
axis([0,1,0,1,-5,10]);
view(-217,30);
%borramos variables que no necesitamos para liberar memoria
clear x,y,f1,g1,f2,g2,F1,G1,F2,G2,I1,J1,I2,J2,S2;
%Resolvemos el segundo sistema%
x = [0:h1:1]; y = x; x_ = [-1:h1:1]; y_ = x_;
for k=1:length(x)
for l=1:length(y)
for i=1:npt
for j=1:mpt
%fuera del rectangulo [-1,1]x[-1,1]
if (x(k)+T*sqrt(1-1/s(i))*cos(z(j)) < -1 | x(k)+T*sqrt(1-1/s(i))*cos(z(j)) > 1 ...
| y(l)+T*sqrt(1-1/s(i))*sin(z(j)) < -1 | y(l)+T*sqrt(1-1/s(i))*sin(z(j)) > 1 )

```

```

f11(k,l,i,j) = 0;
g11(k,l,i,j) = 0;
h11(k,l,i,j) = 0;
%en el primer cuadrante
elseif (x(k)+T*sqrt(1-1/s(i))*cos(z(j))> 0 & x(k)+T*sqrt(1-1/s(i))*cos(z(j)) <
1 ...
& y(l)+T*sqrt(1-1/s(i))*sin(z(j)) > 0 & y(l)+T*sqrt(1-1/s(i))*sin(z(j)) < 1 )
f11(k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x, y, -S1,...
x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)));
g11(k,l,i,j) = (T/(4*pi*s(i)^1.5))*(interp2(x, y, z0X_1C, ...
x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x, y, z0Y_1C, x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
h11(k,l,i,j) = (T/(4*pi*s(i)^1.5))*interp2(x, y, DS,...
x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)));
%en el resto de cuadrantes
else
f11(k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x_, y_, z0,...
x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)));
g11(k,l,i,j) = (T/(4*pi*s(i)^1.5))*(interp2(x_, y_, z0X, ...
x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x_, y_, z0Y, x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
h11(k,l,i,j) = (T/(4*pi*s(i)^1.5))*interp2(x_, y_, zt0,...
x(k)+T*sqrt(1-1/s(i))*cos(z(j)), y(l)+T*sqrt(1-1/s(i))*sin(z(j)));
end
%Ahora lo mismo para el instante T-paso
%fuera del resctangulo [-1,1]x[-1,1]
if (x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)) < -1 | x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j))>
1 ...
| y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)) < -1 | y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j))
> 1 )
f12(k,l,i,j) = 0;
g12(k,l,i,j) = 0;
h12(k,l,i,j) = 0;
%en el primer cuadrante
elseif (x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)) > 0 & x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j))
< 1 ...
& y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)) > 0 & y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j))
< 1 )
f12(k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x, y, -S1,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
g12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*(interp2(x, y, z0X_1C, ...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...

```

```

    interp2(x, y, z0Y_1C, x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-
1/s(i))*sin(z(j)))*...
    sqrt(1-1/s(i))*sin(z(j));
    h12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*interp2(x, y, DS,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
    %en el resto de cuadrantes
    else
    f12(k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x_, y_, z0,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
    g12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*(interp2(x_, y_, z0X, ...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
    interp2(x_, y_, z0Y, x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-
1/s(i))*sin(z(j)))*...
    sqrt(1-1/s(i))*sin(z(j));
    h12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*interp2(x_, y_, zt0,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
    end
    end
    F11(k,l,i) = simps_v(f11(k,l,i,:),hz);
    G11(k,l,i) = simps_v(g11(k,l,i,:),hz);
    H11(k,l,i) = simps_v(h11(k,l,i,:),hz);
    F12(k,l,i) = simps_v(f12(k,l,i,:),hz);
    G12(k,l,i) = simps_v(g12(k,l,i,:),hz);
    H12(k,l,i) = simps_v(h12(k,l,i,:),hz);
    end
    I11(k,l) = simps_v(F11(k,l,:),hs);
    J11(k,l) = simps_v(G11(k,l,:),hs);
    K11(k,l) = simps_v(H11(k,l,:),hs);
    S11(k,l) = I11(k,l) + J11(k,l) + K11(k,l); % posicion en el instante T
    I12(k,l) = simps_v(F12(k,l,:),hs);
    J12(k,l) = simps_v(G12(k,l,:),hs);
    K12(k,l) = simps_v(H12(k,l,:),hs);
    S12(k,l) = I12(k,l) + J12(k,l) + K12(k,l); %posicion en el instante T-paso, para
calcular la velocidad en T
    DS2(k,l) = (S11(k,l)-S12(k,l))/paso; %velocidad en el instante T
    end
    end
    %borramos variables que no necesitamos para liberar memoria
    clear f11,g11,h11,f12,g12,h12,F11,G11,H11,F12,G12,H12,I11,J11,K11,I12,J12,K12,S12;
    figure(3);
    mesh(X,Y,S11);
    axis([0,1,0,1,-5,10]);
    view(-217,30);
    figure(4);
    mesh(X,Y,DS2);
    axis([0,1,0,1,-5,10]);
    view(-217,30);

```

```

%Definimos los datos iniciales para resolver los ultimos sistemas
for k=1:length(x)
for l=1:length(y)
y00(k,l)=10*sin(pi*x(k))*sin(pi*y(l));
end
end
y0 = y00-S11;
y1 = DS2;
[y0X_1C,y0Y_1C] = gradient(y0,h1);
%Extension par e impar de la posicion ,la velocidad y el gradiente
for k=1:length(x)
for l=1:length(y)
Y0(nxy-1+k,nxy-1+l) = y0(k,l);
Y0(nxy+1-k,nxy+1-l) = y0(k,l);
Y0(nxy+1-k,nxy-1+l) = -y0(k,l);
Y0(nxy-1+k,nxy+1-l) = -y0(k,l);
Y1(nxy-1+k,nxy-1+l) = y1(k,l);
Y1(nxy+1-k,nxy+1-l) = y1(k,l);
Y1(nxy+1-k,nxy-1+l) = -y1(k,l);
Y1(nxy-1+k,nxy+1-l) = -y1(k,l);
Y0X(nxy-1+k,nxy-1+l) = y0X_1C(k,l);
Y0X(nxy+1-k,nxy+1-l) = -y0X_1C(k,l);
Y0X(nxy+1-k,nxy-1+l) = -y0X_1C(k,l);
Y0X(nxy-1+k,nxy+1-l) = y0X_1C(k,l);
Y0Y(nxy-1+k,nxy-1+l) = y0Y_1C(k,l);
Y0Y(nxy+1-k,nxy+1-l) = -y0Y_1C(k,l);
Y0Y(nxy+1-k,nxy-1+l) = y0Y_1C(k,l);
Y0Y(nxy-1+k,nxy+1-l) = -y0Y_1C(k,l);
end
end
Y0(nxy,:) = 0;
Y0(:,nxy) = 0;
Y1(nxy,:) = 0;
Y1(:,nxy) = 0;
Y0X(nxy,:) = 0;
Y0X(:,nxy) = 0;
Y0Y(nxy,:) = 0;
Y0Y(:,nxy) = 0;
%Resolvemos de nuevo el primer sistema para estos datos
h2 = T/f; %paso de tiempo
t=0:h2:T;
for r=1:length(t) %animacion en tiempo
for k=1:length(x)
for l=1:length(y)
for i=1:npt
for j=1:mpt
%Fuera del retangulo [-1,1]x[-1,1]

```

```

    if (x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)) < -1 | x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)) >
1 ...
    | y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < -1 | y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) > 1 )
    ff11(r,k,l,i,j) = 0;
    gg11(r,k,l,i,j) = 0;
    hh11(r,k,l,i,j) = 0;
    %En el primer cuadrante
    elseif (x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)) > 0 & x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j))
< 1 ...
    & y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) > 0 & y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < 1
)
    ff11(r,k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x, y, y0,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
    gg11(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x, y, y0X_1C, ...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x, y, y0Y_1C, x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
    hh11(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x, y, y1,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
    % En el resto de cuadrantes
    else
    ff11(r,k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x_, y_, Y0,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
    gg11(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x_, y_, Y0X, ...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x_, y_, Y0Y, x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
    hh11(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x_, y_, Y1,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
    end
    end
    FF11(r,k,l,i) = simps_v(ff11(r,k,l,i,:),hz);
    GG11(r,k,l,i) = simps_v(gg11(r,k,l,i,:),hz);
    HH11(r,k,l,i) = simps_v(hh11(r,k,l,i,:),hz);
    end
    II11(r,k,l) = simps_v(FF11(r,k,l,:),hs);
    JJ11(r,k,l) = simps_v(GG11(r,k,l,:),hs);
    KK11(r,k,l) = simps_v(HH11(r,k,l,:),hs);
    SS11(r,k,l) = II11(r,k,l) + JJ11(r,k,l) + KK11(r,k,l); posicion en cada instante t
    end
    c1(r,k) = SS11(r,k,length(y)); control en la arista y = 1
    end
    end
    %calculo de la poscion en el instante T-paso
    for k=1:length(x)
    for l=1:length(y)

```

```

    for i=1:npt
    for j=1:mpt
    %Fuera del rectangulo [-1,1]x[-1,1]
    if (x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)) < -1 | x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j))
> 1 ...
| y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)) < -1 | y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j))
> 1 )
    ff12(k,l,i,j) = 0;
    gg12(k,l,i,j) = 0;
    hh12(k,l,i,j) = 0;
    %En el primer cuadrante
    elseif (x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j))> 0 & x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j))<
1 ...
& y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j))> 0 & y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j))<1
)
    ff12(k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x, y, y0,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
    gg12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*(interp2(x, y, y0X_1C, ...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x, y, y0Y_1C, x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-
1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
    hh12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*interp2(x, y, y1,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
    %En el resto de cuadrantes
    else
    ff12(k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x_, y_, Y0,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
    gg12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*(interp2(x_, y_, Y0X, ...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x_, y_, Y0Y, x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-
1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
    hh12(k,l,i,j) = ((T-paso)/(4*pi*s(i)^1.5))*interp2(x_, y_, Y1,...
x(k)+(T-paso)*sqrt(1-1/s(i))*cos(z(j)), y(l)+(T-paso)*sqrt(1-1/s(i))*sin(z(j)));
    end
    end
    FF12(k,l,i) = simps_v(ff12(k,l,i,:),hz);
    GG12(k,l,i) = simps_v(gg12(k,l,i,:),hz);
    HH12(k,l,i) = simps_v(hh12(k,l,i,:),hz);
    end
    II12(k,l) = simps_v(FF12(k,l,:),hs);
    JJ12(k,l) = simps_v(GG12(k,l,:),hs);
    KK12(k,l) = simps_v(HH12(k,l,:),hs);
    SS12(k,l) = II12(k,l) + JJ12(k,l) + KK12(k,l); %posicion en el instante T-paso
para calcular la velocidad en T

```



```

end
end
for k=1:length(x)
for l=1:length(y)
DDS1(k,l) = (SS11(length(t),k,l)-SS12(k,l))/paso; %velocidad final
end
end
%borramos variables que no necesitamos para liberar memoria
clear ff11 gg11 hh11 FF11 GG11 HH11 II11 JJ11 KK11 ff12 gg12 hh12 FF12
GG12 HH12 II12 JJ12 KK12;
%Datos iniciales para el ultimo sistema
for k=1:length(x)
for l=1:length(y)
v0(k,l) = -SS11(length(t),k,l);
end
end
v1 = DDS1;
[v0X_1C,v0Y_1C] = gradient(v0,h1);
%Extension par e impar de la posicion ,la velocidad y el gradiente
for k=1:length(x)
for l=1:length(y)
V0(nxy-1+k,nxy-1+l) = v0(k,l);
V0(nxy+1-k,nxy+1-l) = v0(k,l);
V0(nxy+1-k,nxy-1+l) = -v0(k,l);
V0(nxy-1+k,nxy+1-l) = -v0(k,l);
V1(nxy-1+k,nxy-1+l) = v1(k,l);
V1(nxy+1-k,nxy+1-l) = v1(k,l);
V1(nxy+1-k,nxy-1+l) = -v1(k,l);
V1(nxy-1+k,nxy+1-l) = -v1(k,l);
V0X(nxy-1+k,nxy-1+l) = v0X_1C(k,l);
V0X(nxy+1-k,nxy+1-l) = -v0X_1C(k,l);
V0X(nxy+1-k,nxy-1+l) = -v0X_1C(k,l);
V0X(nxy-1+k,nxy+1-l) = v0X_1C(k,l);
V0Y(nxy-1+k,nxy-1+l) = v0Y_1C(k,l);
V0Y(nxy+1-k,nxy+1-l) = -v0Y_1C(k,l);
V0Y(nxy+1-k,nxy-1+l) = v0Y_1C(k,l);
V0Y(nxy-1+k,nxy+1-l) = -v0Y_1C(k,l);
end
end
V0(nxy,:) = 0;
V0(:,nxy) = 0;
V1(nxy,:) = 0;
V1(:,nxy) = 0;
V0X(nxy,:) = 0;
V0X(:,nxy) = 0;
V0Y(nxy,:) = 0;
V0Y(:,nxy) = 0;
%calculamos ahora el control en la arista x = 1

```

```

h2 = T/f; paso de tiempo
t=0:h2:T;
for r=1:length(t) %animacion en tiempo
for k=1:length(x)
for l=1:length(y)
for i=1:npt
for j=1:mpt
%Fuera del rectangulo [-1,1]x[-1,1]
if (1+t(r)*sqrt(1-1/s(i))*cos(z(j)) < -1 | 1+t(r)*sqrt(1-1/s(i))*cos(z(j)) > 1 ...
| y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < -1 | y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) > 1 )
ff11(r,l,i,j) = 0;
gg11(r,l,i,j) = 0;
hh11(r,l,i,j) = 0;
%En el primer cuadrante
elseif (1+t(r)*sqrt(1-1/s(i))*cos(z(j)) > 0 & 1+t(r)*sqrt(1-1/s(i))*cos(z(j))< 1
...
& y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) > 0 & y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < 1
)
ff11(r,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x, y, y0,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
gg11(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x, y, y0X_1C, ...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x, y, y0Y_1C, 1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
hh11(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x, y, y1,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
%En el resto de cuadrantes
else
ff11(r,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x_, y_, Y0,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
gg11(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x_, y_, Y0X, ...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x_, y_, Y0Y, 1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
hh11(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x_, y_, Y1,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
end
end
end
FF11(r,l,i) = simps_v(ff11(r,l,i,:),hz);
GG11(r,l,i) = simps_v(gg11(r,l,i,:),hz);
HH11(r,l,i) = simps_v(hh11(r,l,i,:),hz);
end
II11(r,l) = simps_v(FF11(r,l,:),hs);
JJ11(r,l) = simps_v(GG11(r,l,:),hs);
KK11(r,l) = simps_v(HH11(r,l,:),hs);

```

```

    c1(r,l) = III1(r,l) + JJ11(r,l) + KK11(r,l); %control en cada instante t y posicion
0<y<1, x = 1
    end
    end
    %borramos variables que no necesitamos para liberar memoria
clear ff11 gg11 hh11 FF11 GG11 HH11 III1 JJ11 KK11;
    %Calculamos finalmente el control para el segundo sistema para los datos ade-
cuados
    for r=1:length(t) % animacion en tiempo
        %for k=1:length(x)
        for l=1:length(y)
            for i=1:npt
                for j=1:mpt
                    % Fuera del rectangulo [-1,1]x[-1,1]
                    if (1+t(r)*sqrt(1-1/s(i))*cos(z(j)) < -1 | 1+t(r)*sqrt(1-1/s(i))*cos(z(j)) > 1 ...
| y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < -1 | y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) > 1 )
                        ff21(r,l,i,j) = 0;
                        gg21(r,l,i,j) = 0;
                        hh21(r,l,i,j) = 0;
                        %En el primer cuadrante
                    elseif (1+t(r)*sqrt(1-1/s(i))*cos(z(j)) > 0 & 1+t(r)*sqrt(1-1/s(i))*cos(z(j)) < 1
...
& y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) >0 & y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < 1
)
                        ff21(r,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x, y, v0,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
                        gg21(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x, y, v0X_1C, ...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x, y, v0Y_1C, 1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
                        hh21(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x, y, v1,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
                        %En el resto de cuadrantes
                    else
                        ff21(r,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x_, y_, V0,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
                        gg21(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x_, y_, V0X, ...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x_, y_, V0Y, 1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
                        hh21(r,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x_, y_, V1,...
1+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
                    end
                end
            end
        end
        FF21(r,l,i) = simps_v(ff21(r,l,i,:),hz);
        GG21(r,l,i) = simps_v(gg21(r,l,i,:),hz);

```

```

HH21(r,l,i) = simps_v(hh21(r,l,i,:),hz);
end
II21(r,l) = simps_v(FF21(r,l,:),hs);
JJ21(r,l) = simps_v(GG21(r,l,:),hs);
KK21(r,l) = simps_v(HH21(r,l,:),hs);
c2(r,l) = II21(r,l) + JJ21(r,l) + KK21(r,l); %control en cada instante t, 0 < y < 1,
x = 1
end
end
%sumamos ahora los controles
for r=1:length(t)
for k=1:length(x)
control(r,k) = c1(r,k)+c2(length(t)+1-r,k);
end
end
figure(6);
mesh(x,t,control);
view(-217,30);
xlabel('edge x = 1'); ylabel('time');
print -depsc control_80_20_40_005;
%Resolvemos finalmente el segundo sistema para estos datos
for r=1:length(t) %animacion en tiempo
for k=1:length(x)
for l=1:length(y)
for i=1:npt
for j=1:mpt
%Fuera del rectangulo [-1,1]x[-1,1]
if (x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)) < -1 | x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)) >
1 ...
| y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < -1 | y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) > 1 )
ff21(r,k,l,i,j) = 0;
gg21(r,k,l,i,j) = 0;
hh21(r,k,l,i,j) = 0;
%En el primer cuadrante
elseif (x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)) > 0 & x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j))
< 1 ...
& y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) > 0 & y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)) < 1
)
ff21(r,k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x, y, v0,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
gg21(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x, y, v0X_1C, ...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x, y, v0Y_1C, x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
hh21(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x, y, v1,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
%En el resto de cuadrantes

```

```

else
ff21(r,k,l,i,j) = (1/(4*pi*s(i)^1.5))*interp2(x_, y_, V0,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
gg21(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*(interp2(x_, y_, V0X, ...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*cos(z(j))+...
interp2(x_, y_, V0Y, x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)))*...
sqrt(1-1/s(i))*sin(z(j)));
hh21(r,k,l,i,j) = (t(r)/(4*pi*s(i)^1.5))*interp2(x_, y_, V1,...
x(k)+t(r)*sqrt(1-1/s(i))*cos(z(j)), y(l)+t(r)*sqrt(1-1/s(i))*sin(z(j)));
end
end
FF21(r,k,l,i) = simps_v(ff21(r,k,l,i,:),hz);
GG21(r,k,l,i) = simps_v(gg21(r,k,l,i,:),hz);
HH21(r,k,l,i) = simps_v(hh21(r,k,l,i,:),hz);
end
II21(r,k,l) = simps_v(FF21(r,k,l,:),hs);
JJ21(r,k,l) = simps_v(GG21(r,k,l,:),hs);
KK21(r,k,l) = simps_v(HH21(r,k,l,:),hs);
SS21(r,k,l) = II21(r,k,l) + JJ21(r,k,l) + KK21(r,k,l); %posicion en cada instante
t
end
c2(r,k) = SS21(r,k,length(y)); %control en la arista y = 1
end
end
%sumamos ahora los estados y los controles
for r=1:length(t)
for k=1:length(x)
for l=1:length(y)
sol(r,k,l) = SS11(r,k,l)+SS21(length(t)+1-r,k,l);
end
control(r,k) = c1(r,k)+c2(length(t)+1-r,k);
end
end
fnames = {'wave_state_t0', 'wave_state_t02', 'wave_state_t04', 'wave_state_t06',
'wave_state_t08', 'wave_state_t1',...
'wave_state_t12', 'wave_state_t14', 'wave_state_t16', 'wave_state_t18', 'wave_state_t2',
'wave_state_t22', ...
'wave_state_t24', 'wave_state_t26', 'wave_state_t28', 'wave_state_t3', 'wave_state_t32',
'wave_state_t34',...
'wave_state_t36', 'wave_state_t38', 'wave_state_t4'};
for r=1:length(t)
for k=1:length(x)
for l=1:length(y)
sol_(k,l) = sol(r,k,l);
end
end
end
figure(7);

```

```
mesh(x,y,soL);
axis([0,1,0,1,-5,10]);
view(-217,30);
xlabel('x1');ylabel('x2');
print('-depsec',fnames{r});
M(r) = getframe;
end
movie2avi(M,'ondas2d', 'compression', 'Cinepak');%, 'colormap', cmap);
figure(8);
mesh(x,t,control);
xlabel('edge y = 1'); ylabel('time');
print -depsec control_wave;
```

Bibliografía

- [1] Evans, Lawrence C.: Partial Differential Equations, American Mathematical Society, 1998.
- [2] Lions, J. L.: Contrôllability exacte, perturbations et stabilization de systèmes distribués, Tome I, Masson, Paris, 1988.
- [3] Pedregal, P. and Periago, F.: Some remarks on homogenization and exact boundary controllability for the one-dimensional wave equation, Quart. Appl. Math. 64 (2006), 529-546.
- [4] Pedregal, P., Periago, F. y Villena, J.: A numerical method of local energy decay for the boundary controllability of time-reversible distributed parameter systems, Studies in applied mathematics 121:27-47, 2008.
- [5] Russell, D. L.: A unified boundary controllability theory for hyperbolic and parabolic partial differential equations, Studies in Applied Mathematics, Vol. LII, N° 3, 1973.