UNIVERSIDAD POLITÉCNICA DE CARTAGENA
Escuela Técnica Superior de Ingeniería Industrial

Simulación numérica del control exacto en la frontera para el sistema de la elasticidad lineal en 2D

Titulación: Ingeniero Industrial
Intensificación: –
Alumno/a: Roberto Javier Font Ruiz
Director/a/s: Francisco Periago Esparza

Cartagena, 28 de Junio de 2009

Deseo expresar mi agradecimiento, en primer lugar, a Francisco Periago Esparza, por su confianza, su dedicación y su contagioso entusiasmo. Y a la Universidad Politécnica de Cartagena, por concederme la beca de formación bajo cuya financiación fue realizado este proyecto.

Gracias también a mi familia y amigos por su apoyo constante, tantas veces inmerecido, y sobre todo gracias, María Amelia, por más de lo que podría expresar con palabras.

Índice general

Introducción

El objetivo de este proyecto es utilizar la teoría de la controlabilidad exacta para resolver el problema de control exacto en la frontera para el sistema de la elasticidad lineal. Dicho de otra forma, buscamos calcular el control que hay que aplicar sobre la frontera de un cuerpo elástico de modo que éste, en un cierto tiempo, se estabilice en el estado nulo. Para entender esto, veamos primero algunos conceptos básicos.

Llamamos sistema a un conjunto de elementos interconectados y que se relacionan entre sí de modo diferente a como lo hacen con el exterior del sistema, al que llamamos entorno, de forma que puede considerarse un todo homogéneo. El límite entre el sistema y su entorno se denomina frontera.

Este sistema puede representar desde la economía de un estado hasta una planta industrial, y su comportamiento viene descrito por una serie de ecuaciones (algebraicas, diferenciales, etc.). En el caso que nos ocupa, el sistema será un cuerpo elástico cualquiera, y su comportamiento vendrá modelado por un sistema de ecuaciones en derivadas parciales (dos en el caso bidimensional) que constituyen lo que llamamos el sistema de la elasticidad lineal.

Podemos decir que al aplicar el control sobre un sistema lo que buscamos es forzarlo a que se comporte del modo en que nosotros le dictemos. Así, podemos lograr que nuestro sistema se comporte de modo óptimo, por ejemplo, en forma de productividad máxima, beneficio máximo, costo mínimo o utilización mínima de energía, o controlar de forma precisa el movimiento de un robot industrial.

En las últimas décadas, el control ha cobrado una importancia creciente, hasta el punto de que hoy en día resulta difícil concebir una industria que no lo incorpore en algún punto de su proceso productivo. Un aspecto de particular interés en ingeniería es el control de vibraciones. Las vibraciones suponen una constante fuente de problemas: roturas por fatiga, mal funcionamiento, ruidos... Así, en los últimos tiempos, cada vez más estructuras, como edificios altos, brazos robot o vehículos espaciales incorporan en su diseño algún tipo de control activo de vibraciones.

Puesto que nuestro sistema de la elasticidad lineal modela un cuerpo elástico sometido a vibración, el control exacto de frontera de dicho sistema equivale a calcular cómo hemos de actuar sobre su frontera de forma que eliminemos por completo las vibraciones. El interés de este problema es innegable, por tanto, no sólo desde un punto de vista teórico, sino también del de su posible aplicación en ingeniería.

Matemáticamente, este tipo de problema puede expresarse como un sistema de la forma:

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

donde $\Omega \subset \mathbb{R}^n$ es un dominio acotado con frontera regular Γ, y A un operador elíptico de orden $2k$, $k = 1, 2, \dots$, con coeficientes constantes. Siendo, típicamente, $A = -\Delta$, con Δ el Laplaciano, $A = \Delta^2$, el operador biarmónico, o $A = -\mu\Delta - (\lambda + \mu)\nabla(\nabla\cdot)$ el operador de la elasticidad lineal para materiales isótropos y homogéneos. En este último caso, del que nos ocuparemos en el presente trabajo, la incógnita u es un vector de n componentes. En cuanto a B_j, $1 \leq j \leq m$, supondremos que se trata de una familia de operadores lineales actuando sobre la variable espacial $x \in \Gamma$ para todo tiempo $0 \leq t \leq T$.

Dados unos datos iniciales (u^0, u^1) en espacios apropiados, el problema de control exacto de frontera para el sistema (1) consiste en encontrar una familia de controles de frontera $\{v_j\}_{1 \leq j \leq m}$ tales que en tiempo T la solución de (1) satisfaga la condición de controlabilidad exacta

$$u(\,\cdot\,, T) = u'(\,\cdot\,, T) = 0 \quad \text{en } \Omega \tag{2}$$

Desde un punto de vista teórico, este problema ha sido estudiado y resuelto, en las últimas décadas, por distintos caminos. Véase, por ejemplo, [Russ 73], [Lions 88]. Desde el punto de vista numérico, si bien se han obtenido progresos en los últimos años, 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 y/o frontera no convergen cuando se trata de controlar el sistema.

Volviendo al método de Russell, la principal idea consiste en asociar al sistema de control (1)−(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 con valor nulo fuera de Ω. Si llamamos $\bar{u}$ a la solución de este problema de valores iniciales, entonces los elementos $B_j\bar{u}$, $1 \leq j \leq m$, actuando sobre Γ, producen una disipación de energía en la región de control Ω en la que el problema de controlabilidad exacta original estaba planteado. A partir de aquí, haciendo uso del principio de superposición, la condición de controlabilidad exacta se deduce fácilmente.

En un proyecto fin de carrera anteriormente defendido en esta misma escuela, [Vill 08], se aplicó el método de Russell para simular numéricamente el control exacto de frontera en el caso de la ecuación de ondas; nuestro objetivo en el presente trabajo será emplear el mismo

procedimiento para el caso, algo más complejo, del sistema de la elasticidad lineal. La principal dificultad radica en que, al contrario de lo ocurre con la ecuación de ondas, nuestro sistema no tiene solución explícita. La estrategia que seguiremos, como se verá más adelante, será aplicar la transformada de Fourier al sistema de ecuaciones en derivadas parciales transformándolo en un sistema de ecuaciones diferenciales ordinarias que sí podremos resolver.

En nuestro caso, el sistema (1) tomará la forma:

$$\begin{cases} u_{tt} - \mu\Delta u - (\lambda+\mu)\nabla(\nabla\cdot u) = 0 & \text{en} \quad \Omega\times(0,T) \\ u(x,0) = u^0(x),\ u_t(x,0) = u^1(x) & \text{en} \quad \Omega \\ u(x,t) = 0 & \text{en} \quad \Gamma_0\times(0,T) \\ u(x,t) = v(x,t) & \text{en} \quad \Gamma_1\times(0,T) \end{cases} \tag{3}$$

donde λ y μ son los coeficientes de Lamé, cuyo significado físico se discutirá más adelante. Las condiciones de contorno significan que una parte del contorno, Γ_0, permanece fija, mientras que en el resto de la frontera, Γ_1, actúa una función v a la que llamaremos función de control. Nuestro objetivo, por tanto, será encontrar, para un cierto $T > 0$, el control de frontera $v = v(x,t)$ que haga que se verifique la condición de controlabilidad exacta (2).

En el sistema (3) se ha supuesto, por simplicidad, que el control que actúa sobre la porción de la frontera Γ_1 es de tipo Dirichlet, es decir, la forma en que actuamos sobre esta porción de la frontera es imponiendo un campo de desplazamientos a lo largo de ella. Sin embargo, la principal ventaja del método que emplearemos es que es fácilmente aplicable a problemas con diferentes geometrías (en lo referente al conjunto Ω que define la geometría del cuerpo elástico) y diferentes tipos de control en la frontera. Para ilustrar este punto, en el Capítulo 3 realizaremos simulaciones numéricas para dos diferentes casos: en primer lugar, el cuadrado unidad con controles en forma de desplazamientos actuando sobre dos aristas adyacentes y, en segundo lugar, el círculo unidad con controles tipo Neumann actuando sobre toda la frontera. En este último caso tendremos $v(x,t) = \sigma\cdot n$, lo que representa una densidad de fuerzas de superficie actuando sobre la frontera.

Antes de esto, en el Capítulo 1, nos ocuparemos de las ecuaciones que modelan el comportamiento de un cuerpo elástico con el fin de comprender mejor la realidad física tras el sistema (3). En el Capítulo 2 analizaremos la metodología que posteriormente aplicaremos a los dos casos antes mencionados. El Capítulo 4 presenta unas breves conclusiones y, por último, el Anexo incluye los códigos de MatLab empleados para las simulaciones numéricas.

CAPÍTULO 1

El sistema de la elasticidad lineal

En la introducción se presentó el sistema de la elasticidad lineal y se esbozó el método que seguiremos para la resolución del problema de control exacto en la frontera. Nuestro objetivo en este primer capítulo, antes de abordar la resolución del problema, será profundizar en las ecuaciones de la elasticidad con el fin de comprender mejor la realidad física subyacente. Partiendo de esta realidad física, deduciremos las ecuaciones que modelan el comportamiento del sólido elástico hasta llegar al sistema (3), objeto de nuestro estudio.

El propósito fundamental de la Teoría de la Elasticidad es el cálculo del campo de desplazamientos que aparece en un sólido cuando éste se somete a la acción de cargas exteriores. Este campo de desplazamientos tiene una importante peculiaridad: produce cambio en las posiciones relativas de las partículas del sólido. Para determinar los desplazamientos que sufren los puntos del sólido se requerirá, por tanto, describir dicho cambio de posiciones relativas entre las partículas. Este cambio estará caracterizado por un tensor de segundo orden simétrico, el tensor de deformaciones, del que nos ocuparemos en la sección 1.2. Las ecuaciones que modelizan la deformación de un cuerpo sujeto a la acción de ciertas fuerzas surgirán de leyes de conservación físicas, como la conservación de la cantidad de movimiento, de las que nos ocuparemos en la sección 1.4. Antes de poder aplicar estas leyes, sin embargo, necesitaremos conocer la distribución de tensiones que las fuerzas generan en el cuerpo. En la sección 1.3 introduciremos el tensor de esfuerzos, que describe el estado tensional del solido.

Llegados a este punto, tendremos un total de 9 ecuaciones en derivadas parciales (las tres componentes de la ecuación de conservación de la cantidad de movimiento y seis relaciones deformación-desplazamiento) y 15 incógnitas (las seis componentes del tensor de deformaciones, las seis componentes del tensor de esfuerzos y las tres componentes del desplazamiento). Para

cerrar el problema será necesario, por tanto, introducir un nuevo conjunto de ecuaciones que reciben el nombre de ley de comportamiento o ecuaciones constitutivas del material. Estas ecuaciones, de las que nos ocuparemos en la sección 1.5 serán las encargadas de describir el comportamiento del material, que en nuestro caso será elástico lineal. Una vez que tengamos a nuestra disposición todas las ecuaciones necesarias, en la sección 1.6 veremos cómo reducir este sistema de 15 ecuaciones con 15 incógnitas a un sistema de 3 ecuaciones en derivadas parciales con los desplazamientos como única incógnita, así como las condiciones de contorno a aplicar de forma que nuestro sistema tome su forma final (3). Por último, en la sección 1.7 veremos la forma que tomará el sistema (3) en el caso, bidimensional y en ausencia de fuerzas másicas, del que nos ocuparemos en el presente trabajo.

Se ha tratado de abordar esta tarea con la mayor generalidad y rigor matemático posibles. Un enfoque más *físico*, más usual en ingeniería, puede encontrarse, por ejemplo, en [Par 96].

1.1. El medio continuo

Desde un punto de vista físico, se llama medio continuo a todo líquido, gas o sólido que es considerado desde una perspectiva macroscópica, en contraposición a una descrpción corpuscular. Dado $\Omega_0 \subseteq \mathbb{R}^3$ un abierto acotado y conexo, llamamos medio continuo a toda aplicación

$$\begin{aligned} \Phi : \ \Omega_0 \times [0,T] &\rightarrow \mathbb{R}^3 \\ (X;t) &\mapsto x = \Phi(X;t) \end{aligned}$$

que satisface las siguientes propiedades de regularidad:

(a) $\Phi \in C^2$

(b) para cada t fijo, $\Phi(\,\cdot\,;t) = \Phi_t : \Omega_0 \rightarrow \Omega_t = \Phi(\Omega_0;t)$ es un difeomorfismo

(c) Φ_0 es la identidad

En la definición anterior la variable t representa el tiempo y tanto $X = (X_1, X_2, X_3) \in \Omega_0$ como $x = (x_1, x_2, x_3) \in \Omega_t = \Phi(\Omega_0;t)$ representan las coordenadas espaciales. A partir de ahora denotaremos por $\Omega_0 = \Omega(0)$ la configuración inicial de dicho medio, es decir, Ω_0 representa la región de $\mathbb{R}^3$ ocupada por el sólido en el instante inicial. Por su parte, Ω_t representa la configuración del medio continuo en el instante t, esto es, la región ocupada por dicho medio en el tiempo t.

Es preciso también hacer algunos comentarios sobre las hipótesis (a), (b) y (c) en la definición anterior:

(a) La hipótesis (a) representa físicamente una deformación *suave*. Sin embargo, no es ésa la finalidad de esta hipótesis. Su finalidad es puramente matemática: en lo que sigue tendremos que hacer cálculos en los que nos interesará que se cumpla la propiedad de la igualdad de las derivadas cruzadas; por esa razón hacemos la hipótesis de ser $\Phi \in C^2$.

(b) La hipótesis (b) nos garantiza que para cada t la matriz jacobiana

$$D\Phi(\,\cdot\,;t) = \frac{\partial(x_1, x_2, x_3)}{\partial(X_1, X_2, X_3)}$$

es no singular, es decir, $\det(D\Phi(\,\cdot\,;t)) \neq 0$ para cada t. Además, de la hipótesis (a) concluimos que $D\Phi(X;t)$ es de clase $C^1(\Omega_0 \times [0,T])$.

(c) Dado que Φ_0 es la identidad, $\det(D\Phi(\,\cdot\,;0)) = 1$. De (b) deducimos ahora que

$$\det(D\Phi(\,\cdot\,;t)) > 0 \quad \forall\ 0 \leq t \leq T.$$

Asociado al medio continuo Φ siempre tendremos un campo de velocidades (eulerianas) que se define como

$$\vec{v}(x) = (v_1, v_2, v_3) = \left(\frac{\partial x_1}{\partial t}, \frac{\partial x_2}{\partial t}, \frac{\partial x_3}{\partial t}\right).$$

1.2. Tensor de deformaciones

Desde un punto de vista físico, se dice que un medio continuo sufre una deformación si las distancias relativas de los puntos materiales de que se compone dicho medio varían a lo largo del tiempo. Precisaremos esta definición desde un punto de vista matemático.

Sea $M_0 \equiv X = (X_1, X_2, X_3)$ un punto de la configuración inicial Ω_0, si el medio sufre una deformación, en el instante $t > 0$ el punto M_0 pasa a ocupar la posición $M \in \Omega(t)$ y que denotamos por $M \equiv x = (x_1, x_2, x_3) = \Phi(X;t)$. Por abuso de notación escribiremos $x = x(X;t)$. Por tanto, la deformación (punto a punto) del medio continuo se describe matemáticamente por medio de la ecuación vectorial

$$x = \Phi_t(X) \tag{1.1}$$

donde

$$\Phi_t : \Omega_0 \to \Omega(t).$$

Volvamos ahora a considerar la matriz jacobiana de esta transformación, que a partir de ahora denotaremos por F, esto es,

$$F(M_0) = (F_{ij}(M_0)) = \begin{bmatrix} \dfrac{\partial x_1}{\partial X_1}(M_0) & \dfrac{\partial x_1}{\partial X_2}(M_0) & \dfrac{\partial x_1}{\partial X_3}(M_0) \\ \dfrac{\partial x_2}{\partial X_1}(M_0) & \dfrac{\partial x_2}{\partial X_2}(M_0) & \dfrac{\partial x_2}{\partial X_3}(M_0) \\ \dfrac{\partial x_3}{\partial X_1}(M_0) & \dfrac{\partial x_3}{\partial X_2}(M_0) & \dfrac{\partial x_3}{\partial X_3}(M_0) \end{bmatrix}$$

Si hacemos variar M_0 en Ω_0 obtenemos el campo tensorial

$$\begin{aligned} F \ : \Omega_0 &\to \mathcal{M}_{3\times 3}(\mathbb{R}^3) \\ M_0 &\mapsto F(M_0) \end{aligned}$$

al que suele llamarse campo *gradiente de deformación*.

Si $\overrightarrow{dM_0} = (dX_1, dX_2, dX3)$ es un vector (infinitesimal) en el punto M_0, su transformado por el tensor $F(M_0)$, que denotamos por $\overrightarrow{dM} = (dx_1, dx_2, dx_3)$ satisface la relación

$$\overrightarrow{dM} = F(M_0) \cdot \overrightarrow{dM_0}$$

que escrita en notación tensorial y usando el criterio de sumación de índices (esto es, índices repetidos están sumados) se reescribe como

$$dx_i = F_{ij} dX_j = \frac{\partial x_i}{\partial X_j} dX_j \quad (1 \leq i \leq 3). \tag{1.2}$$

Supongamos ahora que tenemos los vectores $\overrightarrow{dM_0} = (dX_1, dX_2, dX_3)$, $\overrightarrow{\delta M_0} = (\delta X_1, \delta X_2, \delta X_3)$ en el punto $M_0 \in \Omega_0$ y $\overrightarrow{dM} = (dx_1, dx_2, dx_3)$, $\overrightarrow{\delta M_0} = (\delta x_1, \delta x_2, \delta x_3)$ sus correspondientes transformadas por el tensor F en el punto $M \in \Omega(t)$. Una forma de medir la deformación entre las direcciones marcadas por los vectores $\overrightarrow{dM_0}$ y $\overrightarrow{\delta M_0}$ es por medio de la variación de los productos escalares de los dos pares de vectores anteriores, es decir,

$$\overrightarrow{dM} \cdot \overrightarrow{\delta M} - \overrightarrow{dM_0} \cdot \overrightarrow{\delta M_0}$$

que escrito en coordenadas y usando otra vez la notación tensorial y la relación (1.2) se escribe como

$$\begin{aligned} \overrightarrow{dM} \cdot \overrightarrow{\delta M} - \overrightarrow{dM_0} \cdot \overrightarrow{\delta M_0} &= dx_i \delta x_i - dX_j \delta X_j \\ &= F_{ij} dX_j F_{ik} \delta X_k - dX_j \delta X_j \\ &= (F_{ij} F_{ik} - \delta_{jk}) dX_j \delta X_k \\ &= (C_{jk} - \delta_{jk}) dX_j \delta X_k \end{aligned}$$

donde δ_{jk} es la delta de Kronecker (esto es, $\delta_{jk} = 1$ si $j = k$ y $\delta_{jk} = 0$ si $j \neq k$) y

$$C = (C_{jk}) = (F_{ij} F_{ik})$$

se denomina *tensor de dilataciones*. Nótese que $C = F^t F$, y por tanto, C es una matriz simétrica. Sus valores propios se denominan *dilataciones principales* y los vectores propios correspondientes, *direcciones principales de deformación*. Se puede demostrar que estos vectores propios son ortogonales dos a dos y que los valores propios son estrictamente positivos. En la práctica es más usual escribir las ecuaciones anteriores en la forma

$$\begin{aligned} \overrightarrow{dM} \cdot \overrightarrow{\delta M} - \overrightarrow{dM_0} \cdot \overrightarrow{\delta M_0} &= (C_{jk} - \delta_{jk}) dX_j \delta X_k \\ &= 2 D_{jk} dX_j \delta X_k, \end{aligned}$$

donde el tensor

$$D = \frac{1}{2}(C - I) = \frac{1}{2}(C_{jk} - \delta_{jk})$$

se denomina *tensor de deformaciones* y obviamente es también un tensor simétrico.

Consideremos el campo vectorial de desplazamientos del medio continuo, esto es, el campo

$$\begin{aligned} \vec{u} \ : \Omega_0 \ &\to \ \mathbb{R}^3 \\ X \ &\mapsto \ \vec{u}(X;t) = \overrightarrow{M_0M} \end{aligned}$$

siendo $M_0 \equiv X = (X_1, X_2, X_3)$ un punto de Ω_0 y $M \equiv x = (x_1, x_2, x_3)$ la nueva posición que ocupa el punto M_0 en el instante t. De (1.1) se deduce que las coordenadas del campo $\vec{u} = (u_1, u_2, u_3)$ satisfacen la igualdad

$$u_i = x_i - X_i = \Phi_i(X;t) - X_i$$

y por tanto

$$x_i = X_i + u_i(X;t)$$

De esta forma, las componentes del tensor gradiente de deformación vienen dadas por

$$F_{ij} = \frac{\partial x_i}{\partial X_j} = \delta_{ij} + \frac{\partial u_i}{\partial X_j}.$$

En consecuencia, el tensor de dilataciones se escribe como

$$C_{jk} = F_{ij}F_{ik} = \delta_{jk} + \frac{\partial u_j}{\partial X_k} + \frac{\partial u_k}{\partial X_j} + \frac{\partial u_i}{\partial X_j}\frac{\partial u_i}{\partial X_k}$$

y lo que es más importante, el tensor de deformaciones está dado por

$$D_{jk} = \frac{1}{2}\left(\frac{\partial u_j}{\partial X_k} + \frac{\partial u_k}{\partial X_j}\right) + \frac{1}{2}\frac{\partial u_i}{\partial X_j}\frac{\partial u_i}{\partial X_k}$$

Como se puede observar, este tensor de deformaciones es no lineal debido a los productos $\frac{\partial u_i}{\partial X_j}\frac{\partial u_i}{\partial X_k}$. Con el fin de poder hacer mucho más tratable matemáticamente dicho tensor y bajo la hipótesis física de pequeños desplazamientos, es decir,

$$\left|\frac{\partial u_i}{\partial X_j}\right| \ll 1$$

conseguimos linealizar el tensor de deformaciones el cual, una vez linealizado, adopta la forma:

$$\varepsilon = (\varepsilon_{jk})$$

donde

$$\varepsilon_{jk}(\vec{u}) = \frac{1}{2}\left(\frac{\partial u_j}{\partial X_k} + \frac{\partial u_k}{\partial X_j}\right) \tag{1.3}$$

A partir de ahora denotaremos por $\varepsilon = (\varepsilon_{jk})$ al *tensor de deformaciones linealizado* y que está dado por la ecuación (1.3).

1.3. El teorema de Cauchy. Tensor de esfuerzos

Sean $\Omega = \Omega(t)$ un medio continuo y $\omega(t) \subseteq \Omega(t)$ un subconjunto de $\Omega(t)$ que a partir de ahora siempre supondremos es un dominio (conjunto abierto y conexo) acotado cuya frontera $\partial\omega(t)$ es una superficie regular a trozos. Consideremos dos campos vectoriales

$$\begin{aligned} \overrightarrow{b} \ : \Omega(t) \ &\to \ \mathbb{R}^3 \\ x \ &\mapsto \ \overrightarrow{b}(x) \end{aligned}$$

y

$$\begin{aligned} \overrightarrow{\alpha} \ : \Omega(t) \times \mathbb{R}^3 \ &\to \ \mathbb{R}^3 \\ (x; \overrightarrow{n}) \ &\mapsto \ \overrightarrow{\alpha}(x; \overrightarrow{n}) \end{aligned}$$

donde $\overrightarrow{n} = (n_i)$ es cualquier vector unitario situado en el punto x. También se puede decir que $\overrightarrow{n}$ es un vector normal unitario exterior a $\partial\omega(t)$, donde $\partial\omega(t)$ es una "pequeña"superficie (o superficie infinitesimal) sobre la que está situado el punto x. En las aplicaciones, $\overrightarrow{\alpha}$ representa un campo de fuerzas y con la dependencia $\overrightarrow{\alpha} = \overrightarrow{\alpha}(x; \overrightarrow{n})$ sólo se pretende poner de manifiesto que dicho campo de fuerzas depende del punto y también de la dirección, sin olvidarnos claro está del tiempo. A menudo no se suele hacer referencia explícita a la variable temporal t, tal y como acabamos de hacer anteriormente.

Finalmente, consideremos una ley general de conservación que matemáticamente podamos escribir por medio de la identidad

$$\int_\omega \overrightarrow{b} \ dx = \int_{\partial\omega} \overrightarrow{\alpha} \ dS, \quad \forall \ \omega \subseteq \Omega \tag{1.4}$$

Con todos estos elementos estamos en condiciones de enunciar el Teorema de Cauchy. En el teorema, la dependencia en la variable t no juega ningún papel, por lo que evitaremos hacer referencia a ella.

Teorema 1.1 (Cauchy)**.** *Sean Ω, ω, $\overrightarrow{b}$ y $\overrightarrow{\alpha}$ como antes y supongamos que se verifica la ley de conservación* (1.4)*. Supongamos además que se verifican las dos siguientes condiciones:*

1. *$\overrightarrow{b}$ es acotado, es decir, existe una constante $C > 0$ tal que*

$$\left| \overrightarrow{b}(x) \right| \leq C \quad \forall \ x \in \Omega,$$

2. *para cada $\overrightarrow{n}$ fijo la función*

$$\begin{aligned} \overrightarrow{\alpha} \ : \Omega \ &\to \ \mathbb{R}^3 \\ x \ &\mapsto \ \overrightarrow{\alpha}(x; \overrightarrow{n}) \end{aligned}$$

es continua.

Entonces existe un campo tensorial $\sigma = \sigma(x;t)$ (llamado tensor de esfuerzos o de tensiones) tal que

$$\alpha_i = \sigma_{ij}(x;t)n_j,$$

es decir, el campo vectorial $\overrightarrow{\alpha}$ depende linealmente del vector normal $\overrightarrow{n}$.

Demostración: Demostraremos que se verifica la identidad

$$\overrightarrow{\alpha}(x;\overrightarrow{n}) = \overrightarrow{\alpha}(x;\overrightarrow{e_1})n_1 + \overrightarrow{\alpha}(x;\overrightarrow{e_2})n_2 + \overrightarrow{\alpha}(x;\overrightarrow{e_3})n_3$$

Tomemos un punto $x \in \Omega$ y un tetraedro ω centrado en x, contenido en Ω, de vértices x, B, C, D y de forma que $\overrightarrow{xB}$ lleva la dirección de $\overrightarrow{e_1}$ y tiene longitud a_1, $\overrightarrow{xC}$ la de $\overrightarrow{e_2}$ y con longitud a_2, y $\overrightarrow{xD}$ la de $\overrightarrow{e_3}$ y con longitud a_3. Las caras del tetraedro serán denotadas por $\Gamma_1 = xCD$, $\Gamma_2 = xBD$, $\Gamma_3 = xBC$, y finalmente $\Gamma = BCD$, con vectores normales respectivos $-\overrightarrow{e_1}$, $-\overrightarrow{e_2}$, $-\overrightarrow{e_3}$ y $\overrightarrow{n} = (n_1, n_2, n_3)$. En los libros de geometría euclídea se muestra que se tienen las siguientes relaciones

$$n_i = \cos\gamma_i, \quad |\Gamma_i| = |\Gamma|\, n_i \quad (1 \le i \le 3) \tag{1.5}$$

siendo γ_i los ángulos que forman los ejes coordenados con el vector $\overrightarrow{n}$ y $|\Gamma_i|$ el área de cada una de las caras.

Si aplicamos ahora la ley de conservación a esta situación obtenemos

$$\begin{aligned}\int_\omega \overrightarrow{b}(x)\, dx = &\int_\Gamma \overrightarrow{\alpha}(x;\overrightarrow{n})\, dS \\ &+ \int_{\Gamma_1} \overrightarrow{\alpha}(x;-\overrightarrow{e_1})\, dS + \int_{\Gamma_2} \overrightarrow{\alpha}(x;-\overrightarrow{e_2})\, dS + \int_{\Gamma_3} \overrightarrow{\alpha}(x;-\overrightarrow{e_3})\, dS\end{aligned}$$

Dividiendo esta expresión por $|\Gamma|$ y teniendo en cuenta (1.5) y que el campo $\overrightarrow{b}$ es acotado se tiene que

$$\begin{aligned}&\left| \frac{1}{|\Gamma|}\int_\Gamma \overrightarrow{\alpha}(x;\overrightarrow{n})\, dS + \frac{n_1}{|\Gamma_1|}\int_{\Gamma_1} \overrightarrow{\alpha}(x;-\overrightarrow{e_1})\, dS \right. \\ &\left. + \frac{n_1}{|\Gamma_2|}\int_{\Gamma_2} \overrightarrow{\alpha}(x;-\overrightarrow{e_2})\, dS + \frac{n_3}{|\Gamma_3|}\int_{\Gamma_3} \overrightarrow{\alpha}(x;-\overrightarrow{e_3})\, dS \right| \\ &\le \frac{V}{|\Gamma|}\left\|\overrightarrow{b}\right\|_\infty\end{aligned} \tag{1.6}$$

donde V representa el volumen de ω. Si ahora repetimos los cálculos para un tetraedro homotético a escala $\varepsilon > 0$, dado que el nuevo volumen es $V_\varepsilon = \varepsilon^3 V$ y las caras laterales $|\Gamma_i^\varepsilon| = \varepsilon^2|\Gamma_i|$, tomando límites cuando $\varepsilon \to 0$ en la expresión equivalente a (1.6) se obtiene la identidad

$$\overrightarrow{\alpha}(x;\overrightarrow{n}) + \overrightarrow{\alpha}(x;-\overrightarrow{e_1})n_1 + \overrightarrow{\alpha}(x;-\overrightarrow{e_2})n_2 + \overrightarrow{\alpha}(x;-\overrightarrow{e_3})n_3 = 0 \tag{1.7}$$

pues al ser $\overrightarrow{\alpha}(\,\cdot\,;\overrightarrow{n})$ continuo para cada $\overrightarrow{n}$ fijo,

$$\lim_{\varepsilon\to 0}\frac{1}{|\Gamma^\varepsilon|}\int_{\Gamma^\varepsilon}\overrightarrow{\alpha}(x;\overrightarrow{n})\,dS=\overrightarrow{\alpha}(x;\overrightarrow{n})$$

y lo mismo sucede con el resto de las caras del tetraedro.

Todos los cálculos anteriores se han hecho para $n_i>0$ para todo $1\leq i\leq 3$. Si algún n_i fuese negativo bastaría con cambiar la orientación del tetraedro. Si algún $n_i=0$ se podría considerar un paralelepípedo en lugar de un tetraedro. Lo importante es que (1.7) vale para todo vector unitario $\overrightarrow{n}$.

Finalmente veamos que

$$\overrightarrow{\alpha}(x;-\overrightarrow{e_i})=-\overrightarrow{\alpha}(x;\overrightarrow{e_i})\quad\forall\ 1\leq i\leq 3. \tag{1.8}$$

Para ello consideremos una bola centrada en x y de radio $\varepsilon>0$. Dividamos dicha bola en dos semiesferas B_1 y B_2 mediante un plano de normal $\overrightarrow{e_1}$ y con superficies exteriores Γ_1 y Γ_2, con vectores normales $\overrightarrow{n_1}$ y $\overrightarrow{n_2}$, respectivamente. Denotemos por Γ a la superficie plana común a dichas semiesferas y orientada con vector normal exterior $\overrightarrow{e_1}$ en B_1 y $-\overrightarrow{e_1}$ en B_2. Aplicando la ley de conservación a cada una de las superficies anteriores y razonando igual que hemos hecho anteriormente con el tetraedro obtenemos que

$$\int_{\Gamma_1}\overrightarrow{\alpha}(x;\overrightarrow{n_1})\,dS+\int_{\Gamma_2}\overrightarrow{\alpha}(x;\overrightarrow{n_2})\,dS=O(\varepsilon^3),$$

$$\int_{\Gamma_1}\overrightarrow{\alpha}(x;\overrightarrow{n_1})\,dS+\int_{\Gamma}\overrightarrow{\alpha}(x;\overrightarrow{e_1})\,dS=O(\varepsilon^3),$$

y

$$\int_{\Gamma_2}\overrightarrow{\alpha}(x;\overrightarrow{n_2})\,dS+\int_{\Gamma}\overrightarrow{\alpha}(x;-\overrightarrow{e_1})\,dS=O(\varepsilon^3).$$

Si ahora restamos a la primera identidad las dos restantes, dividimos por ε^2 y finalmente tomamos límites cuando $\varepsilon\to 0$ se llega a que

$$\overrightarrow{\alpha}(x;\overrightarrow{e_1})+\overrightarrow{\alpha}(x;-\overrightarrow{e_1})=0$$

tal y como queríamos demostrar. Para las dos direcciones restantes se procede exactamente igual. Nótese que la identidad (1.8) representa el principio físico de acción-reacción. Con todo ello se tiene lo que queríamos probar, ya que en la base $\{\overrightarrow{e_1},\overrightarrow{e_2},\overrightarrow{e_3}\}$ el tensor de esfuerzos $\sigma=(\sigma_{ij})$ es el que tiene por columnas $\{\overrightarrow{\alpha}(x;\overrightarrow{e_1}),\overrightarrow{\alpha}(x;\overrightarrow{e_2}),\overrightarrow{\alpha}(x;\overrightarrow{e_3})\}$. □

Como se dijo anteriormente, $\overrightarrow{\alpha}$ representa a las fuerzas superficiales que actúan sobre cada punto x del medio continuo Ω. Una vez tenemos a nuestra disposición el tensor σ, para determinar la fuerza que actúa sobre el punto x en la dirección $\overrightarrow{n}$ únicamente hemos de multiplicar σ por $\overrightarrow{n}$. Por decirlo de alguna forma, el tensor de esfuerzos proporciona toda la información sobre el estado tensional del medio Ω.

1.4. Leyes de conservación

Nuestro objetivo en esta sección será llegar a deducir las ecuaciones que modelizan la deformación de un medio continuo sujeto a la acción de fuerzas externas y/o internas. Dichas ecuaciones surgen de principios físicos o leyes de conservación tales como el principio de conservación de la masa, del momento, etc. En este proceso jugará un papel fundamental el teorema conocido como *Teorema del transporte de Reynolds*, del que nos ocuparemos a continuación.

1.4.1. El teorema del transporte de Reynolds

Consideremos ahora un conjunto $\omega(t) \subseteq \Omega(t)$, de modo que $\partial\omega(t)$ sea una superficie regular orientable con el vector normal apuntando hacia afuera de $\omega(t)$. Sea ahora $k : \Omega(t) \to \mathbb{R}$ un campo escalar y consideremos finalmente la función

$$K(t) = \int_{\omega(t)} k(x;t)\ dx$$

Nos planteamos ahora el problema siguiente: ¿Cómo calculamos $\frac{dK}{dt}$? Antes de abordar la solución de este problema, es importante tener en cuenta que la dependencia de K en t es no sólo respecto del integrando sino también del dominio de integración. Recordemos que $\overrightarrow{v} = \overrightarrow{v}(x)$ denota el campo de velocidades del medio continuo. El resultado que sigue puede interpretarse como una generalización del concepto de derivada sustancial (o material) de un campo escalar.

Teorema 1.2 (Transporte de Reynolds)**.** *Sean k, $\overrightarrow{v}$ y $\omega(t)$ como antes y supongamos que las funciones k y $\overrightarrow{v}$ son de clase $C^1(\overline{\Omega}\times]0,T[)$ y que sus derivadas parciales son acotadas. Entonces,*

$$\frac{dK}{dt} = \int_{\omega(t)} \left[\frac{\partial k}{\partial t} + \operatorname{div}(k\overrightarrow{v})\right]\ dx \tag{1.9}$$

o también

$$\frac{dK}{dt} = \int_{\omega(t)} \frac{\partial k}{\partial t}\ dx + \int_{\partial\omega(t)} k\overrightarrow{v}\cdot dS \tag{1.10}$$

La demostración de este teorema puede encontrarse en [Malv 69].

1.4.2. Conservación de la masa

Dado un medio continuo $\Omega(t)$, se llama densidad de masa a una función (diferenciable y con derivadas parciales acotadas) $\rho = \rho(x;t)$ que verifica que para cada dominio acotado $\omega(t) \subseteq \Omega(t)$, la masa de $\omega(t)$ está dada por

$$m(\omega(t)) = \int_{\omega(t)} \rho(x;t)\ dx$$

El principio de conservación de la masa establece que

$$\frac{d}{dt}m(\omega(t)) = 0 \quad \forall \omega(t) \subseteq \Omega(t).$$

Del Teorema del transporte de Reynolds se deduce que

$$\frac{d}{dt}m(\omega(t)) = \int_{\omega(t)} \left[\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \overrightarrow{v})\right] \, dx = 0 \quad \forall \omega(t) \subseteq \Omega(t).$$

Si suponemos ahora que la función del integrando anterior es continua, entonces se tiene que

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \overrightarrow{v}) = 0 \tag{1.11}$$

y viceversa, si se cumple (1.11), entonces volviendo hacia atrás también se tiene que

$$\frac{d}{dt}m(\omega(t)) = 0 \quad \forall \omega(t) \subseteq \Omega(t).$$

Por tanto, (1.11) es la expresión analítica del principio de conservación de la masa y se denomina usualmente *ecuación de continuidad*.

1.4.3. Conservación de la cantidad de movimiento

Sean $\Omega = \Omega(t)$ un medio continuo y $\omega(t) \subseteq \Omega(t)$ un dominio acotado. La cantidad de movimiento del sistema material $\omega(t)$ viene dada por

$$\int_{\omega(t)} \rho \overrightarrow{v} \, dx$$

donde $\rho = \rho(x;t)$ es la densidad de masa del sistema $\omega(t)$ y $\overrightarrow{v}$ es el campo de velocidades del medio continuo.

Por otra parte, las fuerzas que actúan sobre $\omega(t)$ son de dos tipos: (a) fuerzas másicas de densidad de volumen $\rho \overrightarrow{f}$, las cuales son bien fuerzas de largo alcance como la fuerza de la gravedad o bien fuerzas de corto alcance que tienen un origen molecular, y (b) fuerzas exteriores que actúan sobre la frontera del sistema $\omega(t)$ y que representamos por medio de la función de densidad $\overrightarrow{g}(x, \overrightarrow{n}; t)$ donde $\overrightarrow{n}$ representa el vector normal unitario exterior a $\partial\omega(t)$. Por tanto, la suma de las fuerzas que actúan sobre el sistema $\omega(t)$ es igual a

$$\int_{\omega(t)} \rho \overrightarrow{f} \, dx + \int_{\partial\omega(t)} \overrightarrow{g}(x, \overrightarrow{n}; t) \, dS$$

El *principio de conservación de la cantidad de movimiento* establece que la variación de la cantidad de movimiento que experimenta el sistema $\omega(t)$ es igual a la suma de las fuerzas que actúan sobre dicho sistema.

De manera precisa y en lenguaje matemático este principio se escribe:

$$\frac{d}{dt}\int_{\omega(t)} \rho\overrightarrow{v}\ dx = \int_{\omega(t)} \rho\overrightarrow{f}\ dx + \int_{\partial\omega(t)} \overrightarrow{g}(x,\overrightarrow{n};t)\ dS \quad \forall\ \omega(t)\subseteq\Omega(t) \tag{1.12}$$

Si suponemos que se verifican las hipótesis del Teorema del transporte de Reynolds, entonces la ecuación (1.12) se transforma en

$$\int_{\omega(t)} \rho\left(\overrightarrow{\gamma} - \overrightarrow{f}\right)\ dx = \int_{\partial\omega(t)} \overrightarrow{g}(x,\overrightarrow{n};t)\ dS \quad \forall\ \omega(t)\subseteq\Omega(t) \tag{1.13}$$

donde

$$\overrightarrow{\gamma} = \frac{D\overrightarrow{v}}{Dt} = \frac{\partial\overrightarrow{v}}{\partial t} + \sum_{j=1}^{3}\frac{\partial\overrightarrow{v}}{\partial x_j}v_j$$

Nótese que para obtener (1.13) se ha utilizado también el principio de conservación de la masa ya que para cada $i = 1, 2, 3$ se tiene que

$$\begin{aligned}\frac{\partial}{\partial t}(\rho v_i) + \sum_{j=1}^{3}\frac{\partial}{\partial x_j}(\rho v_i v_j) = & \\ &= v_i\left(\frac{\partial\rho}{\partial t} + \sum_{j=1}^{3}\frac{\partial}{\partial x_j}(\rho v_j)\right) + \rho\left(\frac{\partial v_i}{\partial t} + \sum_{j=1}^{3}\frac{\partial v_i}{\partial x_j}v_j\right)\\ &= \rho\gamma_i\end{aligned}$$

ya que

$$\frac{\partial\rho}{\partial t} + \sum_{j=1}^{3}\frac{\partial}{\partial x_j}(\rho v_j) = 0$$

por el principio de conservación de la masa.

Si suponemos ahora que los campos vectoriales que aparecen en las dos integrales anteriores satisfacen las hipótesis del Teorema de Cauchy podemos concluir que existe un campo tensorial

$$\begin{aligned}\sigma\ : \Omega(t)\ &\to\ \mathcal{M}_{3\times3}(\mathbb{R}^3)\\ x\ &\mapsto\ \sigma(x)\end{aligned}$$

tal que

$$\overrightarrow{g}(x,\overrightarrow{n};t) = \sigma_{ij}n_j\overrightarrow{e_i}$$

donde $\{\overrightarrow{e_i}\}_{i=1}^{3}$ son los vectores de la base ortonormal de $\mathbb{R}^3$.

De esta forma, el principio de conservación de la cantidad de movimiento (1.13) se reescribe como

$$\int_{\omega(t)} \rho\left(\overrightarrow{\gamma} - \overrightarrow{f}\right)\ dx = \overrightarrow{e_i}\int_{\partial\omega(t)} \sigma_{ij}n_j\ dS \quad \forall\ \omega(t)\subseteq\Omega(t)$$

y aplicando el Teorema de la Divergencia a la integral del término de la derecha,

$$\int_{\omega(t)} \left[\rho(\gamma_i - f_i) - \frac{\partial}{\partial x_j}\sigma_{ij} \right] \, dx = 0 \quad \forall\, \omega(t) \subseteq \Omega(t) \text{ y } \forall\, 1 \leq i \leq 3.$$

Como la igualdad anterior vale para todo $\omega(t) \subseteq \Omega(t)$, si el integrando que aparece en la expresión anterior tiene cierta regularidad (por ejemplo es continuo), entonces no es difícil probar que

$$\rho(\gamma_i - f_i) - \frac{\partial}{\partial x_j}\sigma_{ij} = 0 \quad \forall\, \omega(t) \subseteq \Omega(t) \text{ y } \forall\, 1 \leq i \leq 3$$

es decir

$$\rho\gamma_i = \frac{\partial}{\partial x_j}\sigma_{ij} + \rho f_i \quad \text{en } \Omega(t) \tag{1.14}$$

o también en forma vectorial

$$\rho\overrightarrow{\gamma} = \operatorname{div}(\sigma) + \rho\overrightarrow{f} \tag{1.15}$$

Este sistema de tres ecuaciones en derivadas parciales describe el movimiento del medio continuo $\Omega(t)$ y se denominan *ecuaciones de movimiento*.

1.4.4. Conservación del momento angular

Sea $\omega(t) \subseteq \Omega(t)$, cumpliendo las hipótesis hechas en apartados anteriores, el momento angular del sistema material $\omega(t)$ se escribe en la forma

$$\int_{\omega(t)} x \wedge \rho\overrightarrow{v} \, dx.$$

La ley de conservación del momento para el medio continuo $\Omega(t)$ se escribe como

$$\frac{d}{dt}\int_{\omega(t)} x \wedge \rho\overrightarrow{v} \, dx = \int_{\omega(t)} x \wedge \rho\overrightarrow{f} \, dx + \int_{\partial\omega(t)} x \wedge \overrightarrow{g}(\overrightarrow{n}) \, dS.$$

Derivando en el término de la izquierda llegamos a que

$$\begin{aligned} \int_{\omega(t)} x \wedge \rho\left(\overrightarrow{\gamma} - \overrightarrow{f}\right) \, dx &= \int_{\partial\omega(t)} x \wedge \overrightarrow{g}(\overrightarrow{n}) \, dS \\ &= \int_{\partial\omega(t)} x \wedge \sigma(\overrightarrow{n}) \, dx \end{aligned}$$

donde en la segunda igualdad se ha aplicado el Teorema de Cauchy.

Escribiendo en coordenadas esta última igualdad, una vez aplicado el Teorema de la Divergencia, y teniendo en cuenta que el producto vectorial de dos vectores $\overrightarrow{a}$ y $\overrightarrow{b}$ se puede expresar como

$$\overrightarrow{a} \wedge \overrightarrow{b} = \sum_{i,j,k} \varepsilon_{ijk} a_j b_k \overrightarrow{e}_i,$$

con $\varepsilon_{ijk} = 0$ si alguno de los índices se repite, $\varepsilon_{ijk} = -1$ si la permutación de los índices es impar, y $\varepsilon_{ijk} = 1$ si la permutación es par, se tiene que para cada $i = 1, 2, 3$

$$\int_{\omega(t)} \left\{ \rho \varepsilon_{ijk} x_j (\gamma_k - f_k) - \varepsilon_{ijk} \frac{\partial}{\partial x_l} (x_j \sigma_{kl}) \right\} \, dx$$

$$= \int_{\omega(t)} \left\{ \varepsilon_{ijk} x_j \left[\rho(\gamma_k - f_k) - \frac{\partial}{\partial x_l} \sigma_{kl} \right] - \varepsilon_{ilk} \sigma_{kl} \right\} \, dx = 0 \quad \forall \omega(t).$$

Aplicando ahora el principio de conservación de la cantidad de movimiento obtenemos que

$$\int_{\omega(t)} \varepsilon_{ilk} \sigma_{kl} \, dx = 0 \quad \forall \omega(t)$$

y teniendo en cuenta las propiedades de ε_{ilk} se llega a que $\sigma_{kl} = \sigma_{lk}$, es decir, *el tensor de esfuerzos es simétrico.*

1.5. Ley de comportamiento

Hasta este momento, todas las consideraciones realizadas son válidas para cualquier medio continuo, independientemente de su naturaleza. Las ecuaciones que introduciremos en esta sección relacionan la tensión aplicada sobre el medio con las deformaciones que experimenta; estas ecuaciones caracterizarán, por tanto, el comportamiento específico de un determinado medio, y serán las que permitan modelar el comportamiento de un fluido o de un sólido deformable, por ejemplo. En nuestro caso, trataremos de hallar la ley de comportamiento de un medio homogéneo e isótropo con comportamiento elástico lineal. Comenzaremos considerando la evidencia empírica acerca del comportamiento de este tipo de sólidos, con el fin de comprender mejor el significado físico de las ecuaciones, para a continuación abordar un enfoque más riguroso matemáticamente.

1.5.1. Ley de Hooke. Ecuaciones de Lamé

Experimentalmente, se comprueba que existe un determinado rango de tensiones para el que existe, por un lado, una proporcionalidad entre las tensiones aplicadas y las deformaciones experimentadas, y por otro una reversibilidad total del proceso de carga, teniendo el material la propiedad de recuperar su forma y tamaño una vez que desaparecen las cargas que lo solicitan. Esto es lo que se denomina comportamiento elástico lineal.

Esta relación lineal entre tensión y deformación puede expresarse en la forma:

$$\varepsilon = \frac{\sigma}{E} \tag{1.16}$$

que recibe el nombre de ley de Hooke, quien en 1678 sugirió ('ut tensio sic vis') que el alargamiento iba con la fuerza que actuaba y que permite relacionar en un sólido de comportamiento elástco lineal la deformación en una dirección con la tensión en la misma dirección que la está ocasionando.

La constante E recibe el nombre de módulo de elasticidad longitudinal o módulo de Young y podría definirse como el valor de la tensión que provocaría una deformación longitudinal unitaria. El valor de esta constante representa una medida de la rigidez.

Existe otro fenómeno que puede observarse experimentalmente: acompañando la deformación longitudinal aparece una contracción transversal. Como medida de este fenómeno, Poisson definió el coeficiente:

$$\nu = \frac{\text{contracción transversal unitaria}}{\text{alargamiento longitudinal unitario}}$$

Por tanto, en una dirección perpendicular a aquella en que está aplicada la tensión σ aparecerá una deformación de valor:

$$\varepsilon_{\text{transversal}} = -\nu\varepsilon_n = -\nu\frac{\sigma}{E} \tag{1.17}$$

El coeficiente de Poisson es también una medida de rigidez pero en esta ocasión transversal.

En el caso de tener una tensión actuando en una única dirección, por tanto, la relación entre tensión y deformación vendrá dada por las ecuaciones (1.16) y (1.17). En el caso de un estado tensional cualquiera, vendrá dada por la llamada ley de Hooke generalizada:

$$\varepsilon_{ij} = \frac{1+\nu}{E}\sigma_{ij} - \frac{\nu}{E}\sigma_{kk}\delta_{ij} \tag{1.18}$$

La demostración de esta expresión puede encontarse, por ejemplo, en [Par 96] o [Malv 69].

Si consideramos ahora la deformación tangencial del material, γ_{ij}, podemos introducir una nueva propiedad del material:

$$\gamma_{ij} = 2\varepsilon_{ij} = \frac{2(1+\nu)}{E}\sigma_{ij} = \frac{\sigma_{ij}}{G} \qquad i \neq j$$

donde el valor G introducido, también denotado por μ, representa la proporcionalidad entre la tensión tangencial aplicada σ_{ij} y la deformación tangencial (distorsión) γ_{ij}:

$$\mu = G = \frac{\sigma_{ij}}{\gamma_{ij}} \quad (i \neq j)$$

y que tiene un valor, en función de las propiedades ya definidas del material:

$$\mu = G = \frac{E}{2(1+\nu)} \tag{1.19}$$

G se denomina módulo de elasticidad tangencial o más usualmente módulo de cizalladura. Este valor puede definirse como la tensión tangencial que produciría un valor unidad de la deformación tangencial. G, similarmente a E, representa una medida de la rigidez, en este caso tangencial o a cizalladura.

Como hemos visto, la ley de Hooke generalizada, proporciona las deformaciones en función de las tensiones. Invirtiendo la ecuación (1.18) podemos obtener las tensiones en función de las deformaciones:

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \tag{1.20}$$

ecuaciones que se conocen como ecuaciones de Lamé, donde λ y μ son los coeficientes de Lamé, siendo $\mu = G$, como ya vimos, y

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}.$$

1.5.2. Ley general de comportamiento elástico lineal

Se dice que un medio continuo $\Omega = \Omega(t)$ es elástico si existe un sistema de coordenadas y un tiempo $t = 0$ en el que Ω está libre de tensión, y si después de la deformación el tensor de esfuerzos depende únicamente del tensor de deformaciones calculado en este mismo sistema de referencia. Escrito en lenguaje más matemático, diremos que el medio Ω es elástico si su tensor de esfuerzos asociado se puede escribir en la forma

$$\sigma_{ij} = a_{ijkh} \varepsilon_{kh} \tag{1.21}$$

donde $\varepsilon(\overrightarrow{u}) = (\varepsilon_{ij})$ es el tensor de deformaciones (linealizado) y a_{ijkh} son unos coeficientes que dependen de las propiedades del material y que se denominan coeficientes de elasticidad. Se dice que el material elástico Ω es homogéneo si los coeficientes a_{ijkh} son constantes, es decir, si no dependen del punto $x \in \Omega$. En caso contrario se dice que el medio es no homogéneo. Se dice que un material elástico es isótropo si en cualquier punto $x \in \Omega$ el material tiene las mismas propiedades en todas las direcciones posibles alrededor de dicho punto.

Para el caso de materiales elásticos isótropos y homogéneos puede demostrarse que el número de constantes a_{ijkh} necesario se reduce a dos, y el tensor de esfuerzos se escribe en la forma

$$\sigma = \lambda \text{traza}(\varepsilon(\overrightarrow{u}))I + 2\mu\varepsilon(\overrightarrow{u})) \tag{1.22}$$

expresión análoga a (1.20).

En [Par 96] puede encontrarse el proceso detallado que conduce de (1.21) a (1.22) mediante la aplicación de las sucesivas hipótesis simplificativas.

1.6. El problema elástico

1.6.1. Formulación en desplazamientos. Ecuaciones de Navier

Como se dijo anteriormente, el objetivo de la Teoría de la Elasticidad es la determinación del campo de desplazamientos que aparece en un sólido cuando se le somete a acciones exteriores, particularizadas en fuerzas másicas y de contorno y desplazamientos impuestos. Hemos visto que para obtener la respuesta del sólido elástico es necesario incluir otras variables del problema que afectan a todos los puntos del sólido: los tensores de tensión y deformación. Por consiguiente, para la solución completa del problema elástico es necesario conocer no sólo la configuración deformada del sólido sino también el estado de tensiones y deformaciones en cada punto. A lo largo de las secciones anteriores hemos ido deduciendo las ecuaciones que relacionan estas variables:

La ecuación vectorial del movimiento:

$$\rho\overrightarrow{u_{tt}} = \mathrm{div}\sigma + \rho\overrightarrow{f} \tag{1.23}$$

Relaciones desplazamientos−deformaciones:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right) \tag{1.24}$$

Leyes de comportamiento:

$$\sigma_{ij} = \lambda\varepsilon_{kk}\delta_{ij} + 2\mu\varepsilon_{ij} \tag{1.25}$$

Así pues, el problema elástico está constituido por un sistema de 15 ecuaciones en derivadas parciales (tres ecuaciones del movimiento, seis relaciones desplazamientos−deformaciones y seis ecuaciones de comportamiento) con 15 incógnitas (seis componentes del tensor de tensiones, seis componentes del tensor de deformaciones y tres desplazamientos).

Para expresar este problema en función de los desplazamientos basta con sustituir las relaciones desplazamientos−deformaciones (1.24) en las leyes de comportamiento (1.25) y el resultado en (1.23). Así, sustituyendo las relaciones desplazamientos−deformaciones en las leyes de comportamiento obtenemos:

$$\sigma_{ij} = \mu\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right) + \lambda\frac{\partial u_k}{\partial X_k}\delta_{ij}$$

La divergencia de un tensor es un vector, y $\frac{\partial\sigma_{ij}}{\partial X_j}$ es la componente i del vector divergencia del tensor de tensiones:

$$\frac{\partial\sigma_{ij}}{\partial X_j} = \lambda\frac{\partial^2 u_k}{\partial X_i\partial X_k} + 2\mu\frac{1}{2}\frac{\partial^2 u_i}{\partial X_j^2} + 2\mu\frac{1}{2}\frac{\partial^2 u_j}{\partial X_i\partial X_j}$$

agrupando, queda:

$$\frac{\partial \sigma_{ij}}{\partial X_j} = (\lambda + \mu)\frac{\partial^2 u_k}{\partial X_i \partial X_k} + \mu \frac{\partial^2 u_i}{\partial X_j^2}$$

que, usando los operadores vectoriales, puede expresarse en la forma:

$$\text{div}\sigma = (\lambda + \mu)\nabla(\nabla \cdot \overrightarrow{u}) + \mu \Delta \overrightarrow{u}.$$

Finalmente, sustituyendo en (1.23)

$$\rho \overrightarrow{u_{tt}} - (\lambda + \mu)\nabla(\nabla \cdot \overrightarrow{u}) - \mu \Delta \overrightarrow{u} - \rho \overrightarrow{f} = 0 \tag{1.26}$$

La ecuación (1.26) representa un sistema de 3 ecuaciones en derivadas parciales acopladas con 3 incógnitas: las componentes del campo de desplazamientos. Estas ecuaciones se conocen con el nombre de ecuaciones de Navier, quien las introdujo en 1821, pero no siguiendo el desarrollo anterior sino partiendo del concepto newtoniano de interacción molecular.

1.6.2. Condiciones de contorno

Una vez que tenemos las ecuaciones de Navier, sólo resta añadir las condiciones de contorno para completar nuestro modelo. En mecánica de sólidos suelen aparecer las siguientes tres condiciones:

(a) Frontera fija o empotrada: es aquella en la que no se produce ningún desplazamiento. Matemáticamente se expresa como:

$$\overrightarrow{u} = 0 \qquad \text{sobre } \Gamma.$$

(b) Frontera libre: es aquella sobre la que no actúa ninguna fuerza. Matemáticamente se expresa como

$$\sigma_{ij} n_j = 0 \qquad \text{sobre } \Gamma.$$

(c) Frontera sometida a la acción de fuerzas externas: es aquella sometida a la acción de una densidad de fuerzas superficiales $\overrightarrow{g} = (g_i)$. Matemáticamente se expresa como

$$\sigma_{ij} n_j = g_i \qquad \text{sobre } \Gamma.$$

1.7. Caso bidimensional en ausencia de fuerzas másicas

En caso de que no existan fuerzas másicas, la ecuación de Navier (1.26) tomará la forma:

$$\rho \overrightarrow{u_{tt}} - (\lambda + \mu)\nabla(\nabla \cdot \overrightarrow{u}) - \mu \Delta \overrightarrow{u} = 0 \tag{1.27}$$

Si, finalmente, tomamos $\rho = 1$ en esta ecuación, obtendremos la forma dada en (3).

Esta última hipótesis puede parecer enormemente restrictiva desde un punto de vista físico. Sin embargo, hay que tener en cuenta que para modelar el comportamiento de un material cualquiera bastaría con dividir (1.27) entre ρ de forma que $\lambda' = \frac{\lambda}{\rho}$ y $\mu' = \frac{\mu}{\rho}$ pasarían a desempeñar los papeles de λ y μ. En este caso, μ' dejaría de representar al módulo de elasticidad tangencial, pero la ecuación seguiría siendo totalmente válida. Añadiendo las condiciones iniciales y de contorno, como vimos en (3) y desarrollando (1.27) para el caso bidimensional:

$$\begin{cases} u_{1,tt} - \mu\left(\dfrac{\partial^2 u_1}{\partial x_1^2} + \dfrac{\partial^2 u_1}{\partial x_2^2}\right) - (\lambda+\mu)\dfrac{\partial}{\partial x_1}\left(\dfrac{\partial u_1}{\partial x_1} + \dfrac{\partial u_2}{\partial x_2}\right) = 0 & \text{en} \quad \Omega \times (0,T) \\ u_{2,tt} - \mu\left(\dfrac{\partial^2 u_2}{\partial x_1^2} + \dfrac{\partial^2 u_2}{\partial x_2^2}\right) - (\lambda+\mu)\dfrac{\partial}{\partial x_2}\left(\dfrac{\partial u_1}{\partial x_1} + \dfrac{\partial u_2}{\partial x_2}\right) = 0 & \text{en} \quad \Omega \times (0,T) \\ u_1(x,0) = u_1^0(x),\ u_2(x,0) = u_2^0(x) & \text{en} \quad \Omega \\ u_{1,t}(x,0) = u_1^1(x),\ u_{2,t}(x,0) = u_2^1(x) & \text{en} \quad \Omega \\ (u_1,u_2)(x,t) = 0 & \text{en} \quad \Gamma_0 \times (0,T) \\ (u_1,u_2)(x,t) = (v_1,v_2)(x,t) & \text{en} \quad \Gamma_1 \times (0,T) \end{cases} \tag{1.28}$$

CAPÍTULO 2

Resolución del problema de controlabilidad exacta en la frontera

Una vez formulado el problema y expuesta la realidad física subyacente, el propósito del presente capítulo es la obtención del control exacto en la frontera para el sitema de la elasticidad lineal en 2D. En la primera sección expondremos el método propuesto para la resolución de dicho problema. Una descripción más general y con mayor rigor matemático puede encontrarse en [PPV 08]. Puesto que el método requiere la resolución de varios problemas de Cauchy, la segunda de la secciones abordará la metodología empleada a tal fin. Para ello, introduciremos la transformada rápida de Fourier, de la que nos ocuparemos en la sección 2.2.1.

2.1. Descripción del método

A lo largo de esta sección, al igual que hicimos en (3), supondremos, con el fin de simplificar la notación, que los controles actúan en forma de desplazamientos. Hay que recordar, sin embargo, que el método es aplicable para cualquier tipo de control en la frontera.

Supongamos que los datos iniciales del sistema (3), $(u^0 = (u_1^0, u_2^0), u^1 = (u_1^1, u_2^1))$, pertenecen a espacios adecuados (véase [PPV 08] para una discusión acerca de los requisitos que deben cumplir dichos espacios) $X = X_0 \times X_1$.

El método propuesto se compone de tres pasos:

Paso 1: Extensión a un problema de Cauchy en todo el espacio. Comenzaremos por extender los datos iniciales (u^0, u^1) a todo el espacio $\mathbb{R}^2$. Si llamamos $(\overline{u}^0, \overline{u}^1)$ a estos nuevos datos, tendremos el siguiente problema de Cauchy:

$$\begin{cases} \overline{u}_{tt} - \mu\Delta\overline{u} - (\lambda+\mu)\nabla(\nabla\cdot\overline{u}) = 0 & \text{en} \quad \mathbb{R}^2 \times (0,T) \\ \overline{u}(0) = \overline{u}^0,\ \overline{u}_t(0) = \overline{u}^1 & \text{en} \quad \mathbb{R}^2 \end{cases} \tag{2.1}$$

La restricción de $\overline{u}$ a Ω, $\overline{u}|_\Omega$, es solución del sistema (3) con $v = \overline{u}|_{\Gamma_1}$.

La extensión de los datos iniciales (u^0, u^1) debe ser tal que la solución de (2.1), $\overline{u}$, cumpla las condiciones de frontera. Por razones prácticas, es importante controlar un sistema actuando sólo sobre una pequeña parte de la frontera. Por esta razón, las condiciones de contorno tendrán la forma vista en (3):

$$\begin{cases} u(x,t) = 0 & \text{en} \quad \Gamma_0 \\ u(x,t) = v(x,t) & \text{en} \quad \Gamma_1 \end{cases}$$

Debemos, por tanto, extender los datos de forma que se verifiquen estas condiciones de frontera. En geometrías sencillas, tales como rectángulos, no es difícil de conseguir, como veremos, mientras que si el control actúa sobre toda la frontera bastará con extender (u^0, u^1) con valor nulo fuera de Ω. Ambos casos serán contemplados en las simulaciones numéricas, de modo que en Capítulo 3 entraremos con mayor detalle en la extensión de los datos necesaria en cada caso.

Paso 2: Disipación local de la energía. El punto clave del método es que se cumpla la siguiente propiedad: Existe una constante positiva $C(T) < 1$ tal que

$$\|(u(T), u_t(T))\|_X \leq C(T)\|(u^0, u^1)\|_X \tag{2.2}$$

El significado de (2.2) es que la energía del sistema tiende a disiparse conforme actúa el control, atenuando el movimiento. Es necesario hacer notar que puesto que el control actúa sobre la frontera, y debido a la velocidad finita en que la información puede viajar a través del cuerpo elástico, la condición de controlabilidad exacta (2) no puede lograrse para un tiempo T arbitrariamente pequeño. Existe por lo tanto un tiempo mínimo de controlabilidad, $T^\star > 0$, que depende de Ω, Γ_0 y de los coeficientes de Lamé, para el que el problema (3)-(2) tiene solución. Los detalles acerca de este aspecto pueden consultarse en [Lions 88].

Paso 3: Principio de superposición. A continuación, mostraremos que, si se cumple (2.2), se verificará la condición de controlabilidad exacta (2). Tomemos (ϕ^0, ϕ^1) y extendamos estos datos en todo $\mathbb{R}^2$ como se vio en el paso 1. Si llamamos $(\overline{\phi}^0, \overline{\phi}^1)$ a estos nuevos datos y consideramos el problema de Cauchy

$$\begin{cases} \overline{\phi}_{tt} - \mu\Delta\overline{\phi} - (\lambda+\mu)\nabla(\nabla\cdot\overline{\phi}) = 0 & \text{en} \quad \mathbb{R}^2 \times (0,T) \\ \overline{\phi}(0) = \overline{\phi}^0,\ \overline{\phi}_t(0) = \overline{\phi}^1 & \text{en} \quad \mathbb{R}^2 \end{cases} \tag{2.3}$$

entonces, la función $\phi = \overline{\phi}|_\Omega$ es solución de

$$\begin{cases} \phi_{tt} - \mu\Delta\phi - (\lambda+\mu)\nabla(\nabla\cdot\phi) = 0 & \text{en} \quad \Omega\times(0,T) \\ \phi(0) = \phi^0,\ \phi_t(0) = \phi^1 & \text{en} \quad \Omega \\ \phi = g & \text{en} \quad \Gamma \end{cases} \tag{2.4}$$

donde $g = \overline{\phi}|_\Gamma$.

A continuación consideremos el sistema

$$\begin{cases} \overline{\psi}_{tt} - \mu\Delta\overline{\psi} - (\lambda+\mu)\nabla(\nabla\cdot\overline{\psi}) = 0 & \text{en} \quad \mathbb{R}^2\times(0,T) \\ \overline{\psi}(0) = \overline{\psi}^0,\ \overline{\psi}_t(0) = \overline{\psi}^1 & \text{en} \quad \mathbb{R}^2 \end{cases} \tag{2.5}$$

donde los datos $(\overline{\psi}^0, \overline{\psi}^1)$ se obtienen en la forma

$$(\psi^0, \psi^1) = (-\phi(T), \phi_t(T)) \tag{2.6}$$

realizando la extensión, de nuevo, como se vio en el paso 1. Así, la función $\psi = \overline{\psi}|_\Omega$ es solución de

$$\begin{cases} \psi_{tt} - \mu\Delta\psi - (\lambda+\mu)\nabla(\nabla\cdot\psi) = 0 & \text{en} \quad \Omega\times(0,T) \\ \psi(0) = \psi^0,\ \psi_t(0) = \psi^1 & \text{en} \quad \Omega \\ \psi = h & \text{en} \quad \Gamma \end{cases} \tag{2.7}$$

donde $h = \overline{\psi}|_\Gamma$.

Finalmente, consideremos $z(x,t) = \phi(x,t) + \psi(x, T-t)$. Puesto que el operador $-\mu\Delta - (\lambda+\mu)\nabla(\nabla\cdot)$ es lineal, z es solución del problema de controlabilidad exacta

$$\begin{cases} z_{tt} - \mu\Delta z - (\lambda+\mu)\nabla(\nabla\cdot z) = 0 & \text{en} \quad \Omega\times(0,T) \\ z(0) = z^0,\ z_t(0) = z^1 & \text{en} \quad \Omega \\ z = f & \text{en} \quad \Gamma \\ z(T) = z_t(T) = 0 & \text{en} \quad \Omega \end{cases} \tag{2.8}$$

con

$$(z^0, z^1) = (\phi^0 + \psi(T), \phi^1 - \psi_t(T)) \tag{2.9}$$

y

$$f(x,t) = g(x,t) + h(x, T-t).$$

La función $z(x,t)$ calculada mediante este procedimiento verifica la ecuación en derivadas parciales, como se ha dicho, por la linealidad del operador de la elasticidad lineal; y la condición de controlabilidad exacta (2), puesto que, como $(\psi^0, \psi^1) = (-\phi(T), \phi_t(T))$, se cumple que

$$(z(T), z_t(T)) = (\phi(T) + \psi(0), \phi_t(T) - \psi_t(0)) = (\phi(T) - \phi(T), \phi_t(T) - \phi_t(T)) = (0,0)$$

Sólo resta, por tanto, comprobar si esta solución verifica las condiciones iniciales. Puesto que z tiene por condiciones iniciales las dadas por (2.9), la pregunta es si cualquier condición inicial (u^0, u^1) puede ser representada en la forma (2.9). Esto equivale a decir que la aplicación

$$\begin{aligned} L_T \ : \ X \ &\to \ X \\ (\phi^0, \phi^1) \ &\mapsto \ (\phi^0 + \psi(T), \phi^1 - \psi_t(T)) \end{aligned}$$

es sobreyectiva. Si descomponemos L_T en la forma $L_T = I - K_T$, con

$$K_T(\phi^0, \phi^1) = (-\psi(T), \psi_t(T)),$$

basta con probar que $\|K_T\| < 1$, ya que en este caso L_T es invertible y su inverso viene dado por

$$L_T^{-1} = \frac{1}{1 - K_T} = I + K_T + K_T^2 + K_T^3 + \dots$$

Para ello, es suficiente con una doble aplicación de (2.2):

$$\begin{aligned} \|K_T(\phi^0, \phi^1)\|_X &= \|(-\psi(T), \psi_t(T))\|_X \\ &\leq C(T)\|(-\phi(T), \phi_t(T))\|_X \\ &\leq (C(T))^2\|(\phi^0, \phi^1)\|_X \end{aligned}$$

Debemos encontrar (ϕ^0, ϕ^1) tales que $L_T(\phi^0, \phi^1) = (u^0, u^1)$. Efectivamente, si esto es así,

$$\begin{aligned} (u^0, u^1) = L_T(\phi^0, \phi^1) = I - K_T = (\phi^0, \phi^1) - (-\psi(T), \psi_t(T)) \\ = (\phi^0 + \psi(T), \phi^1 - \psi_t(T)) = (z^0, z^1) \end{aligned}$$

y por tanto la solución z verifica las condiciones iniciales. Estos (ϕ^0, ϕ^1) se calcularán haciendo uso de la inversa de L_T:

$$\begin{aligned} L_T^{-1} \ : \ X \ &\to \ X \\ (u^0, u^1) \ &\mapsto \ (\phi^0, \phi^1) \end{aligned}$$

En la práctica sólo podremos obtener una aproximación de L_T^{-1}. Recordemos que

$$L_T^{-1} = I + K_T + K_T^2 + K_T^3 + \dots$$

por tanto, podremos obtener diferentes aproximaciones de L_T^{-1} dependiendo del orden en K_T que consideremos. Para el caso de una aproximación de primer orden en K_T, $L_T^{-1} \approx I + K_T$, y por tanto

$$L_T^{-1}(u^0, u^1) \approx (u^0, u^1) + K_T(u^0, u^1). \tag{2.10}$$

Algoritmo numérico

Con esta aproximación, el algoritmo numérico propuesto es el siguiente:

(a) Tomar (u^0, u^1), extender estos datos a todo $\mathbb{R}^2$ en la forma descrita en el Paso 1, y resolver el sistema (2.3) para tiempo $t = T$.

(b) Con la solución obtenida en (a), construir unas nuevas condiciones iniciales como se vio en (2.6), extender estos datos como en el paso anterior y resolver el sistema (2.5) para $t = T$. Llamemos a la solución de este último sistema $\varphi(x, T)$.

(c) Definir los nuevos datos

$$(\phi^0, \phi^1) = (u^0 - \varphi(T), u^1 + \varphi_t(T)),$$

aproximación de primer orden de $L_T^{-1}(u^0, u^1)$ como en (2.10).

(d) Con estos nuevos datos iniciales, repetir los pasos (a) y (b), calculando las soluciones de estos problemas, $\phi(x, t)$ y $\psi(x, t)$, y los controles $g(x, t)$ y $h(x, t)$ según se definen en (2.4) y (2.7). Las aproximaciones tanto para el estado como para el control vendrán dadas por

$$u^*(x, t) = \phi(x, t) + \psi(x, T - t) \tag{2.11}$$

y

$$v^*(x, t) = g(x, t) + h(x, T - t) \qquad \forall\ (x, t) \in \Gamma$$

respectivamente. La aproximación del estado, (2.11), satisface las condiciones iniciales

$$\begin{cases} u^*(x, 0) \equiv u_0^* = \phi^0 + \psi(T) \\ u_t^*(x, 0) \equiv u_1^* = \phi^1 - \psi_t(T) \end{cases}$$

que son aproximaciones de las originales, (u^0, u^1).

2.2. Resolución del problema de Cauchy

Como hemos visto, el algoritmo para el cálculo del control implica la resolución de varios problemas de Cauchy de la forma

$$\begin{cases} \overline{\phi}_{tt} - \mu\Delta\overline{\phi} - (\lambda + \mu)\nabla(\nabla \cdot \overline{\phi}) = 0 & \text{en} \quad \mathbb{R}^2 \times (0, T) \\ \overline{\phi}(0) = \overline{\phi}^0,\ \overline{\phi}_t(0) = \overline{\phi}^1 & \text{en} \quad \mathbb{R}^2 \end{cases} \tag{2.12}$$

Un sistema de dos ecuaciones en derivadas parciales acopladas para el que no existe solución explícita. La estrategia que seguiremos para la resolución de este sistema será:

(a) Aplicar la transformada de Fourier al sistema (2.12) respecto a las variables espaciales, (x_1, x_2), de forma que convirtamos el sistema de dos ecuaciones en derivadas parciales en un sistema de dos ecuaciones diferenciales ordinarias.

(b) Resolver el problema transformado mediante los medios usuales, en concreto haciendo uso, como veremos, del concepto de exponencial de una matriz.

(c) Aplicar la transformada inversa de Fourier a la solución del problema transformado, obteniendo la solución del problema original.

2.2.1. La transformada rápida de Fourier

La transformada de Fourier es una herramienta ampliamente utilizada en ingeniería. En la presente sección nos limitaremos a recordar algunas de sus propiedades básicas para a continuación ocuparnos de la transformada discreta de Fourier, que con algunas modificaciones se convierte en la transformada rápida de Fourier; herramienta que usaremos, como se ha dicho, para la resolución de (2.12).

La transformada de Fourier

Dada una función $f : \mathbb{R} \to \mathbb{C}$, se define su transformada de Fourier como

$$\widehat{f}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{e}^{-i\xi x} f(x)\, dx.$$

La convergencia de dicha integral está garantizada en caso de ser f absolutamente integrable, es decir, la transformada de Fourier de f está bien definida siempre que $f \in L^1(\mathbb{R})$, siendo

$$L^1(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{C} : \int_{-\infty}^{+\infty} |f(x)|\, dx < \infty \right\}.$$

Algunas propiedades de esta transformada que usaremos al aplicarla al sistema de la elasticidad lineal son:

(a) **Linealidad.** Sean f, $g \in L^1(\mathbb{R})$ y α, $\beta \in \mathbb{C}$. Entonces

$$[\alpha f + \beta g]^\wedge = \alpha \widehat{f} + \beta \widehat{g}.$$

(b) **Traslación.** Sean $f \in L^1(\mathbb{R})$ y $a \in \mathbb{R}$. Entonces

$$f(x-a)^\wedge = \mathrm{e}^{-ia\xi} \widehat{f}(\xi).$$

(c) **Derivación.** Sea $f : \mathbb{R} \to \mathbb{C}$ una función de clase C^1 a trozos y supongamos que tanto f como f' están en $L^1(\mathbb{R})$. Entonces

$$[f']^\wedge(\xi) = i\xi\widehat{f}(\xi).$$

En general se tiene que si $f, f', \ldots, f^{(n)} \in L^1(\mathbb{R})$:

$$\left[f^{(n)}\right]^\wedge(\xi) = (i\xi)^n\widehat{f}(\xi).$$

Las demostraciones de todas estas propiedades pueden encontrarse en [Fol 92].

Consideremos, por último, el problema de recuperar la función f a partir de su transformada de Fourier $\widehat{f}$. Para ello, dada $f \in L^1(\mathbb{R})$ se define su **transformada inversa de Fourier** como

$$(f)^\vee(x) = \int_{-\infty}^{\infty} \mathrm{e}^{i\xi x} f(\xi)\, d\xi.$$

Si tanto f como $\widehat{f}$ pertenecen a $L^1(\mathbb{R})$, entonces $f = \left(\widehat{f}\,\right)^\vee$, es decir,

$$f(x) = \int_{-\infty}^{\infty} \mathrm{e}^{i\xi x}\widehat{f}(\xi)\, d\xi \qquad \text{para todo } x \in \mathbb{R}.$$

Antes de tratar la transformada discreta de Fourier recordemos un resultado que nos será de utilidad.

Teorema 2.1 (Del muestreo de Shannon)**.** *Supongamos que f es continua a trozos, $f \in L^2(\mathbb{R}) \cap L^1(\mathbb{R})$ y que $\widehat{f}(\xi) = 0$ para $\xi \notin [-L, L]$. Entonces*

$$f(t) = \sum_{n=-\infty}^{\infty} f\left(\frac{n\pi}{L}\right) \frac{\sin(Lt - n\pi)}{Lt - n\pi},$$

es decir, f está completamente determinada por los valores de f en los puntos $\frac{n\pi}{L}$.

La demostración puede así mismo encontrarse en [Fol 92].

La transformada discreta de Fourier

Consideremos el problema de la aproximación numérica de la transformada de Fourier. Para ser capaces de utilizar la transformada de Fourier en cálculos informáticos debemos aproximarla mediante algo que involucre únicamente un número finito de operaciones algebráicas efectuadas sobre un conjunto finito de datos.

En primer lugar sustituiremos el intervalo infinito $]-\infty, \infty[$ por un intervalo finito $[a, a+L]$. Podemos suponer que $a = 0$, algo que siempre podremos conseguir tomando $f(x-a)$ en lugar de $f(x)$.

En segundo lugar, intentaremos calcular $\widehat{f}(\xi)$, no en todo $\xi \in \mathbb{R}$, sino en una sucesión de puntos contenida en algún intervalo $[-C, C]$. La elección de C puede ser determinada aproximadamente si conocemos la velocidad a la que $\widehat{f}$ se anula cuando $\xi \to \infty$, pero, ¿qué sucesión elegimos? El teorema del muestreo indica que $\widehat{f}$ puede ser completamente reconstruida a partir de sus valores en puntos $\xi_m = 2\pi m/L,\ m \in \mathbb{N}$. Por tanto, hemos de calcular

$$\widehat{f}\left(\frac{2\pi m}{L}\right) = \int_0^L \mathrm{e}^{-2\pi i m x/L} f(x)\ dx, \qquad |m| \leq \frac{CL}{2\pi}.$$

Finalmente, reemplazamos la integral que aparece en la expresión anterior por una suma parcial de Riemann que obtenemos dividiendo el intervalo $[0, L]$ en N subintervalos obtenidos a partir de los puntos nL/N $(n = 0, \ldots, N)$. De esta forma tenemos:

$$\widehat{f}\left(\frac{2\pi m}{L}\right) \approx \sum_{n=0}^{N-1} \mathrm{e}^{-2\pi i m n/N} f\left(\frac{nL}{N}\right) \frac{L}{N}$$

Para que esta aproximación sea aceptablemente buena es preciso tomar $N \gg CL/(2\pi)$.

En resumen: si denotamos por $a_n = f(\frac{nL}{N})$, entonces

$$\widehat{f}\left(\frac{2\pi m}{L}\right) \approx \frac{L}{N}\, \widehat{a}_m,$$

donde

$$\widehat{a}_m = \sum_{n=0}^{N-1} \mathrm{e}^{-2\pi i m n/N} a_n. \tag{2.13}$$

Puesto que $\mathrm{e}^{-2\pi i m} = 1$, $\{\widehat{a}_m\}$ es periódica con periodo $N : \widehat{a}_{m+N} = \widehat{a}_n$. Por tanto, toda la información está contenida en la sucesión finita $\widehat{a}_0,\ \widehat{a}_1, \ldots,\ \widehat{a}_{N-1}$. De esta forma hemos generado una aplicación

$$\begin{array}{rccl} \mathcal{F}_N\ :\ \mathbb{C}^N & \to & \mathbb{C}^N \\ \mathrm{a} & \mapsto & \mathcal{F}_N \mathrm{a} = \widehat{\mathrm{a}} = (\widehat{a}_0, \ldots, \widehat{a}_m) \end{array}$$

donde

$$\mathrm{a} = (a_0, \ldots, a_{N-1}), \qquad \widehat{a}_m = \sum_{n=0}^{N-1} \mathrm{e}^{-2\pi i m n/N} a_n \quad (0 \leq m < N).$$

Esta aplicación se denomina *transformada discreta de Fourier de N−puntos.*

En cuanto a la fórmula de inversión para la transformada discreta de Fourier, las componentes del vector a, que denotamos por a_n, están dadas por

$$a_n = \frac{1}{N} \sum_{m=0}^{N-1} \mathrm{e}^{2\pi i m n/N} \widehat{a}_m.$$

La fórmula anterior es la fórmula de inversión para la transformada discreta de Fourier.

Como hemos dicho, la transformada discreta de Fourier suele emplearse como aproximación numérica de la transformada de Fourier ordinaria. Desde un punto de vista computacional, sin embargo, la transformada discreta de Fourier presenta un serio inconveniente. Llamemos "operación elemental" a la multiplicación de dos números complejos seguida de la suma de dos números complejos. Observando (2.13), podemos ver que el cálculo de cada $\widehat{a}_m$ requiere N operaciones elementales y hay N elementos $\widehat{a}_m$; por tanto, el cálculo de $\widehat{\mathrm{a}}$ requiere un total de N^2 operaciones elementales. Cuando N es grande, como a menudo es necesario para obtener buenos resultados numéricos, N^2 puede resultar demasiado grande para que el cálculo de la transformada discreta de Fourier sea abordable computacionalmente.

Cuando N es primo poco puede hacerse al respecto. Sin embargo, si N puede expresarse en la forma $N = N_1N_2$ los cálculos pueden reducirse sustancialmente agrupándolos de forma eficiente. Si esto ocurre, el mismo cálculo se repite para N_2 valores de m, de forma que puede ahorrarse tiempo llevándolo a cabo una vez y luego usándolo repetidas veces. Este cálculo requiere N_1 operaciones elementales y debe efectuarse $N_1N_2 = N$ veces. Una vez que se ha hecho esto, se requieren N_2 operaciones elementales para el cálculo de cada $\widehat{\mathrm{a}}$, y hay N de éstos. El total, por tanto, es $N(N_1 + N_2)$ operaciones elementales, en lugar de las $N^2 = N(N_1N_2)$ del método original, lo que supone una apreciable mejora.

Cuando N_1 o N_2 pueden seguir siendo factorizados, el proceso puede repetirse para aumentar aún más la eficiencia. El resultado final es que si $N = N_1N_2 \dots N_k$, el número de operaciones elementales puede ser reducido de N^2 a $N(N_1 + \cdots + N_k)$. En particular, la mayor reducción posible se obtiene cuando N es potencia de 2, el caso más empleado en la práctica. El algoritmo resultante para el cálculo de la transformada discreta de Fourier se conoce como **transformada rápida de Fourier**.

2.2.2. Transformada de Fourier del sistema de la elasticidad lineal

Como vimos en la sección 1.7, nuestro sistema, una vez realizada la extensión a todo $\mathbb{R}^2$ tomará la forma:

$$\begin{cases} u_{1,tt} - \mu\left(\dfrac{\partial^2 u_1}{\partial x_1^2} + \dfrac{\partial^2 u_1}{\partial x_2^2}\right) - (\lambda+\mu)\dfrac{\partial}{\partial x_1}\left(\dfrac{\partial u_1}{\partial x_1} + \dfrac{\partial u_2}{\partial x_2}\right) = 0 & \text{en} \quad \mathbb{R}^2 \times (0,T) \\ u_{2,tt} - \mu\left(\dfrac{\partial^2 u_2}{\partial x_1^2} + \dfrac{\partial^2 u_2}{\partial x_2^2}\right) - (\lambda+\mu)\dfrac{\partial}{\partial x_2}\left(\dfrac{\partial u_1}{\partial x_1} + \dfrac{\partial u_2}{\partial x_2}\right) = 0 & \text{en} \quad \mathbb{R}^2 \times (0,T) \\ u_1(x,0) = u_1^0(x),\ u_2(x,0) = u_2^0(x) & \text{en} \quad \mathbb{R}^2 \\ u_{1,t}(x,0) = u_1^1(x),\ u_{2,t}(x,0) = u_2^1(x) & \text{en} \quad \mathbb{R}^2 \end{cases} \tag{2.14}$$

Hemos de aplicar a este sistema la transformada de Fourier 2D respecto de (x_1, x_2). Si comenzamos por aplicar la transformada a cada uno de los términos de la primera ecuación,

obtenemos:

$$\begin{aligned}(u_{1,tt}(x,t))^{\wedge} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{-2\pi i\omega x}\frac{\partial^2 u_1}{\partial t^2}(x,t)\ dx_1dx_2\\ &= \frac{\partial^2}{\partial t^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{-2\pi i\omega x}u_1(x,t)\ dx_1dx_2\\ &= \frac{\partial^2}{\partial t^2}\widehat{u}_1(\omega,t) = \widehat{u}_1''(\omega,t)\end{aligned}$$

$$\begin{aligned}\left(-\mu\left(\frac{\partial^2 u_1}{\partial x_1^2}+\frac{\partial^2 u_1}{\partial x_2^2}\right)\right)^{\wedge} &= -\mu\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{-2\pi i\omega x}\left(\frac{\partial^2 u_1}{\partial x_1^2}(x,t)+\frac{\partial^2 u_1}{\partial x_2^2}(x,t)\right)\ dx_1dx_2\\ &= -\mu[(\omega_1 i)^2+(\omega_2 i)^2]\widehat{u}_1\\ &= \mu(\omega_1^2+\omega_2^2)\widehat{u}_1\end{aligned}$$

$$\begin{aligned}\left(-(\lambda+\mu)\frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_1}+\frac{\partial u_2}{\partial x_2}\right)\right)^{\wedge} &= -(\lambda+\mu)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{-2\pi i\omega x}\frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_1}+\frac{\partial u_2}{\partial x_2}\right)\ dx_1dx_2\\ &= -(\lambda+\mu)[(\omega_1 i)^2\widehat{u}_1+(\omega_1 i\omega_2 i)\widehat{u}_2]\\ &= (\lambda+\mu)[\omega_1^2\widehat{u}_1+\omega_1\omega_2\widehat{u}_2]\end{aligned}$$

Con lo que la ecuación transformada queda:

$$\widehat{u}_1''+\mu(\omega_1^2+\omega_2^2)\widehat{u}_1+(\lambda+\mu)[\omega_1^2\widehat{u}_1+\omega_1\omega_2\widehat{u}_2]=0$$

Procediendo de forma análoga con la segunda ecuación, el problema transformado toma la forma:

$$\begin{cases}\widehat{u}_1''+\mu(\omega_1^2+\omega_2^2)\widehat{u}_1+(\lambda+\mu)[\omega_1^2\widehat{u}_1+\omega_1\omega_2\widehat{u}_2]=0\\ \widehat{u}_2''+\mu(\omega_1^2+\omega_2^2)\widehat{u}_2+(\lambda+\mu)[\omega_1\omega_2\widehat{u}_1+\omega_2^2\widehat{u}_2]=0\\ \widehat{u}_1(\omega,0)=\widehat{u}_1^0(\omega); \qquad \widehat{u}_2(\omega,0)=\widehat{u}_2^0(\omega)\\ \widehat{u}_1'(\omega,0)=\widehat{u}_1^1(\omega); \qquad \widehat{u}_2'(\omega,0)=\widehat{u}_2^1(\omega)\end{cases} \tag{2.15}$$

siendo $\widehat{u}_1$, $\widehat{u}_2$ las transformadas de las funciones incógnita y $\widehat{u}_1^0$, $\widehat{u}_2^0$, $\widehat{u}_1^1$, $\widehat{u}_2^1$ resultado de aplicar la transformada rápida de Fourier a los datos iniciales del problema.

2.2.3. Resolución del problema transformado

Una vez efectuada la transformación, el sistema (2.15) es fácilmente resoluble; bastará con convertirlo en un sistema de primer orden. Denotemos, por simplicidad:

$$\begin{aligned}P_1(\mu,\omega_1,\omega_2) &= \mu(\omega_1^2+\omega_2^2)\\ P_2(\lambda,\mu,\omega_1,\omega_2) &= (\lambda+\mu)(\omega_1\omega_2)\\ P_3(\lambda,\mu,\omega_1,\omega_2) &= (\lambda+\mu)\omega_1^2\\ P_4(\lambda,\mu,\omega_1,\omega_2) &= (\lambda+\mu)\omega_2^2\end{aligned}$$

y cosideremos

$$y_1 = \widehat{u}_1$$
$$y_2 = \widehat{u}_2$$
$$y_3 = \widehat{u}_1^{'}$$
$$y_4 = \widehat{u}_2^{'}$$

Entonces, el sistema (2.15) toma la forma:

$$\left\{\begin{array}{l} y_1' = y_3 \\ y_2' = y_4 \\ y_3' = -(P_1+P_3)y_1 - P_2 y_2 \\ y_4' = -(P_1+P_4)y_2 - P_2 y_1 \end{array}\right.$$

o, expresado en forma matricial:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -(P_1+P_3) & -P_2 & 0 & 0 \\ -P_2 & -(P_1+P_4) & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

con las condiciones iniciales

$$\begin{pmatrix} y_1(0) \\ y_2(0) \\ y_3(0) \\ y_4(0) \end{pmatrix} = \begin{pmatrix} \widehat{u}_1^0 \\ \widehat{u}_2^0 \\ \widehat{u}_1^1 \\ \widehat{u}_2^1 \end{pmatrix}$$

donde $\widehat{u}_1^0$, $\widehat{u}_2^0$, $\widehat{u}_1^1$, $\widehat{u}_2^1$ son resultado de aplicar la transformada rápida de Fourier a los datos iniciales del problema (2.14).

Con todo esto, el sistema que debemos resolver puede escribirse en la forma

$$\left\{\begin{array}{l} y' = Ay \\ y(0) = y_0 \end{array}\right\}$$

La solución de este sistema vendrá dada por:

$$y(t) = \mathrm{e}^{At} y_0.$$

Una vez calculada esta solución, sólo resta aplicar la transformada inversa, en nuestro caso la transformada rápida inversa, para obtener la solución del sistema original.

CAPÍTULO 3

Simulaciones numéricas

El propósito de este capítulo es aplicar la metodología propuesta a dos casos concretos. La principal ventaja del método es el ser aplicable a problemas con diferentes geometrías (en lo referente al conjunto Ω que define la geometría del cuerpo elástico) y a diferentes tipos de condiciones de frontera (Neumann, Robin, etc.). Para ilustrar este hecho, consideraremos dos casos con dos diferentes geometrías y condiciones de contorno. En primer lugar el cuadrado unidad con controles actuando en dos aristas adyacentes en forma de desplazamientos (controles tipo Dirichlet), y en segundo lugar el círculo unidad con controles actuando sobre toda la frontera en forma de esfuerzo sobre su superficie (controles tipo Neumann).

3.1. Cuadrado unidad con controles en forma de desplazamientos sobre dos aristas adyacentes

En esta primera simulación numérica consideraremos el problema

$$\begin{cases} u_{tt} - \mu\Delta u - (\lambda+\mu)\nabla(\nabla\cdot u) = 0 & \text{en} \quad \Omega\times(0,T) \\ u(x,0) = u^0(x),\ u_t(x,0) = u^1(x) & \text{en} \quad \Omega \\ u(x,t) = 0 & \text{en} \quad \Gamma_0\times(0,T) \\ u(x,t) = v(x,t) & \text{en} \quad \Gamma_1\times(0,T) \end{cases} \tag{3.1}$$

$$u(\,\cdot\,,T) = u'(\,\cdot\,,T) = 0 \quad \text{en } \Omega \tag{3.2}$$

sobre el cuadrado unidad. Es decir, en este caso el conjunto Ω será $\Omega \equiv R_1 = (0,1)^2$. Una porción de la frontera, Γ_0, permanecerá fija, mientras que sobre el resto actuarán los controles

en forma de desplazamientos. En nuestro caso Γ_0 estará formada por los ejes $x_1 = 0$ y $x_2 = 0$, mientras que Γ_1 será el resto de la frontera:

$$\Gamma_1 = \left\{(1, s) \in \mathbb{R}^2 : \quad 0 < s \leq 1, \ \right\} \cup \left\{(s, 1) \in \mathbb{R}^2 : \quad 0 < s \leq 1\right\}.$$

Como vimos en el capítulo anterior, para resolver el problema hemos de comenzar por extender los datos iniciales $\left(u^0, u^1\right)$ del sistema (3.1) a todo $\mathbb{R}^2$, y esta extensión de los datos debe hacerse de forma que la solución del problema de Cauchy asociado cumpla las condiciones de contorno sobre Γ_0. Para lograr esto, consideremos los rectángulos

$$R_2 = (-1, 0) \times (0, 1), \quad R_3 = (-1, 0) \times (-1, 0) \quad \text{and} \quad R_4 = (0, 1) \times (-1, 0)$$

y llamemos

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

Debemos extender $\left(u^0, u^1\right)$ en R de forma impar en R_2 y R_4 y de forma par en R_3. Sobre el resto del plano, extendemos los datos con valor nulo.

Es sabido, véase [Lions 88, p. 474], que para este caso el tiempo mínimo de controlabilidad viene dado por la expresión

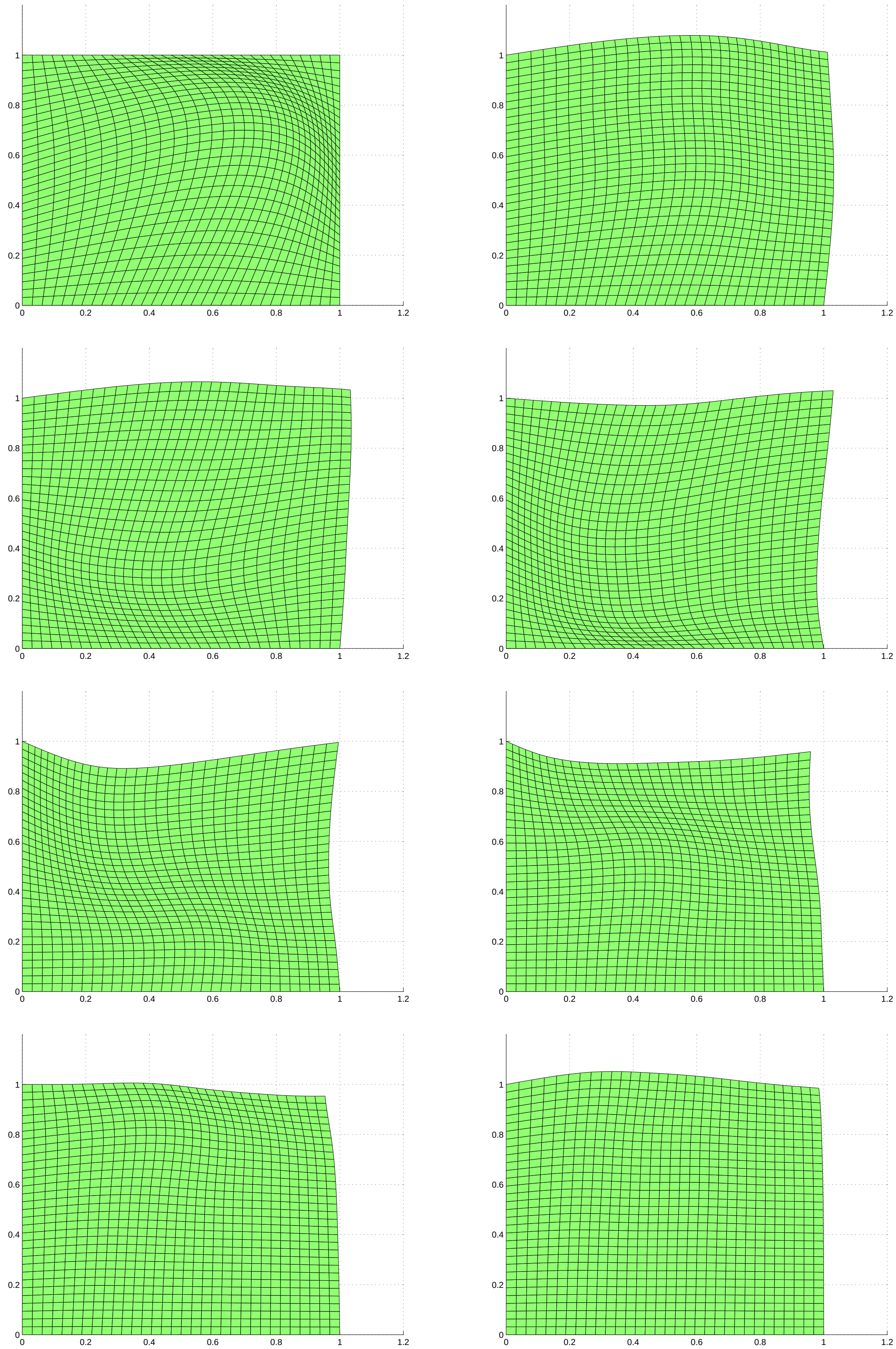
$$T^* = \frac{2\sqrt{2}}{\mu}.$$

Para la simulación tomaremos $\lambda = 0{,}5$, $\mu = 1$, $T = 3$ y consideraremos las condiciones iniciales

$$u^0(x_1, x_2) = (0{,}2 \sin(\pi x_1) \sin(\pi x_2), \ 0{,}2 \sin(\pi x_1) \sin(\pi x_2)), \quad u^1(x_1, x_2) = (0, \ 0).$$

Como se dijo, se ha usado la transformada rápida de Fourier (FFT) para resolver los problemas de Cauchy asociados. Siguiendo la notación de la sección 2.2.1, se ha tomado $N = 1024$ y $L = 32$, lo que proporciona un mallado de tamaño $h = L/N = 0{,}0313$ y una resolución en el dominio de la frecuencia $fr = 2\pi/L = 0{,}1963$. Como es usual, para evitar errores, se ha tomado $K = N/8$. Tanto el algoritmo para el cálculo de la transformada rápida de Fourier como el empleado para el cálculo de su inversa fueron previamente testados en funciones para las que la transformada de Fourier se conoce explícitamente, obteniéndose errores del orden de $10e - 14$. Ambos códigos pueden encontrarse en el Anexo.

Con todo esto, se obtuvo un error para el estado en tiempo $t = T$ de $0,119e-8$ y de $0,064$ para los controles en tiempo $t = 0$. La figura 3.1 muestra el estado para diferentes tiempos en forma de malla deformada.



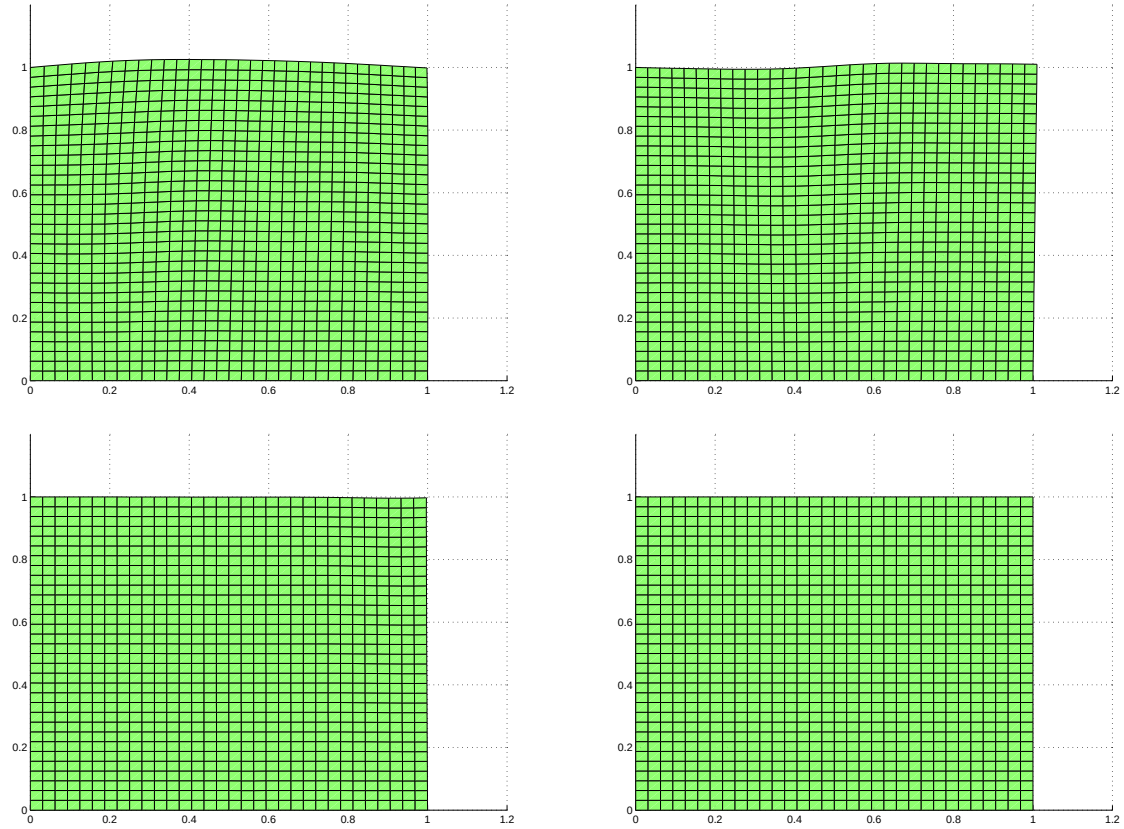
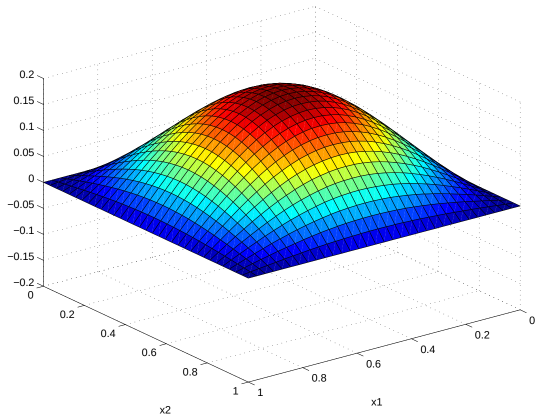
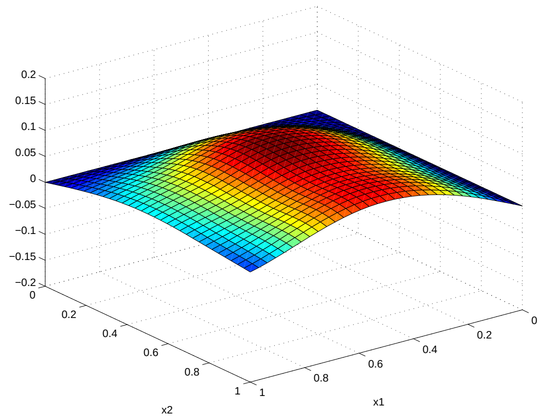
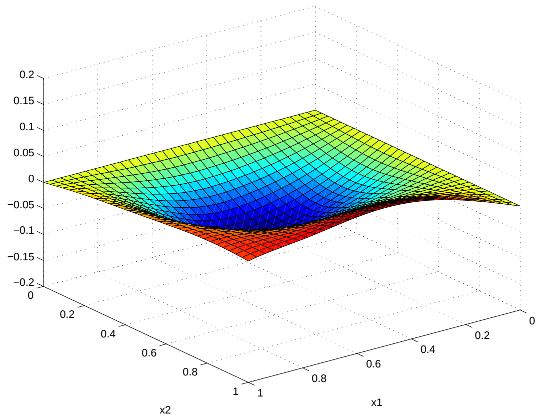


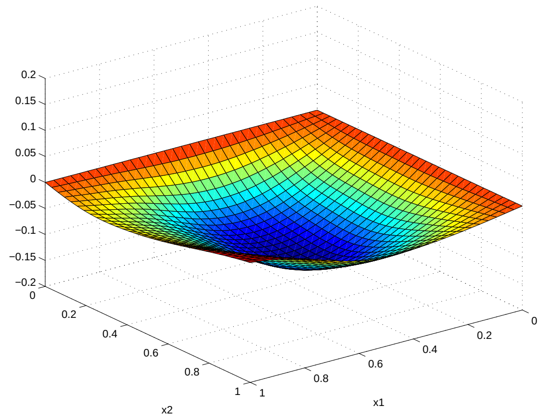
Figura 3.1: Animación del estado $u(x, t_k)$ de izquierda a derecha y de arriba a abajo para t_k =0, 0.15, 0.30, 0.45, 0.60, 0.75, 0.90, 1.05, 1.20, 1.50, 2.10, 3.

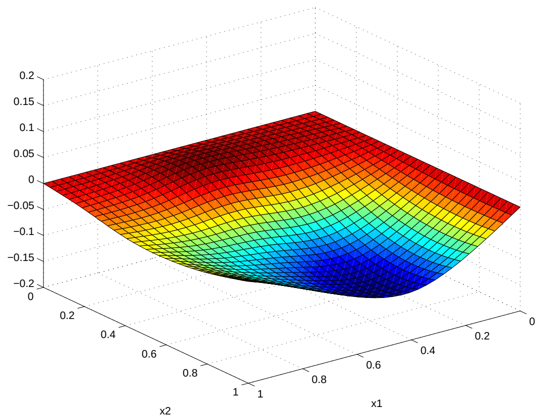
La figura 3.2 muestra en el eje z el valor de la componente en x_1 del desplazamiento para distintos tiempos. Debido a la simetría de las condiciones iniciales los valores en x_2 resultan idénticos.

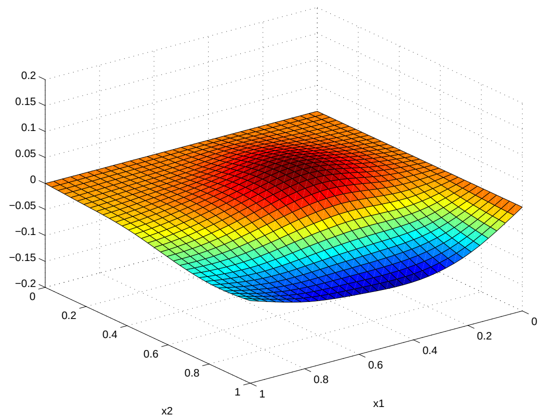

x1
x2

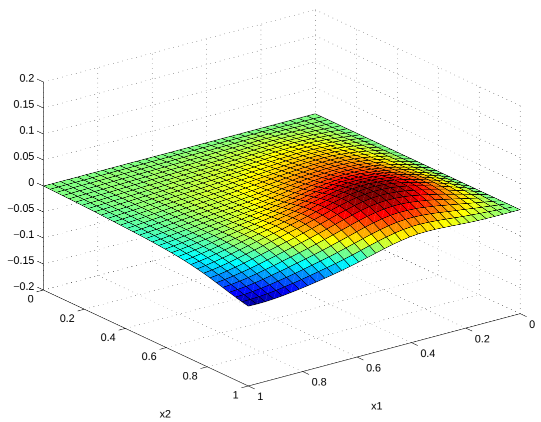

x1
x2

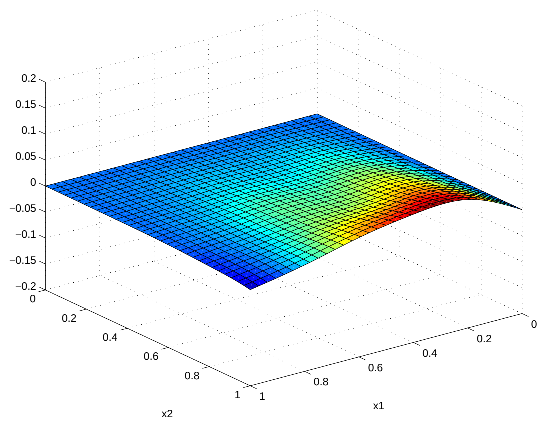

x1
x2


x1
x2


x1
x2


x1
x2


x1
x2


x1
x2

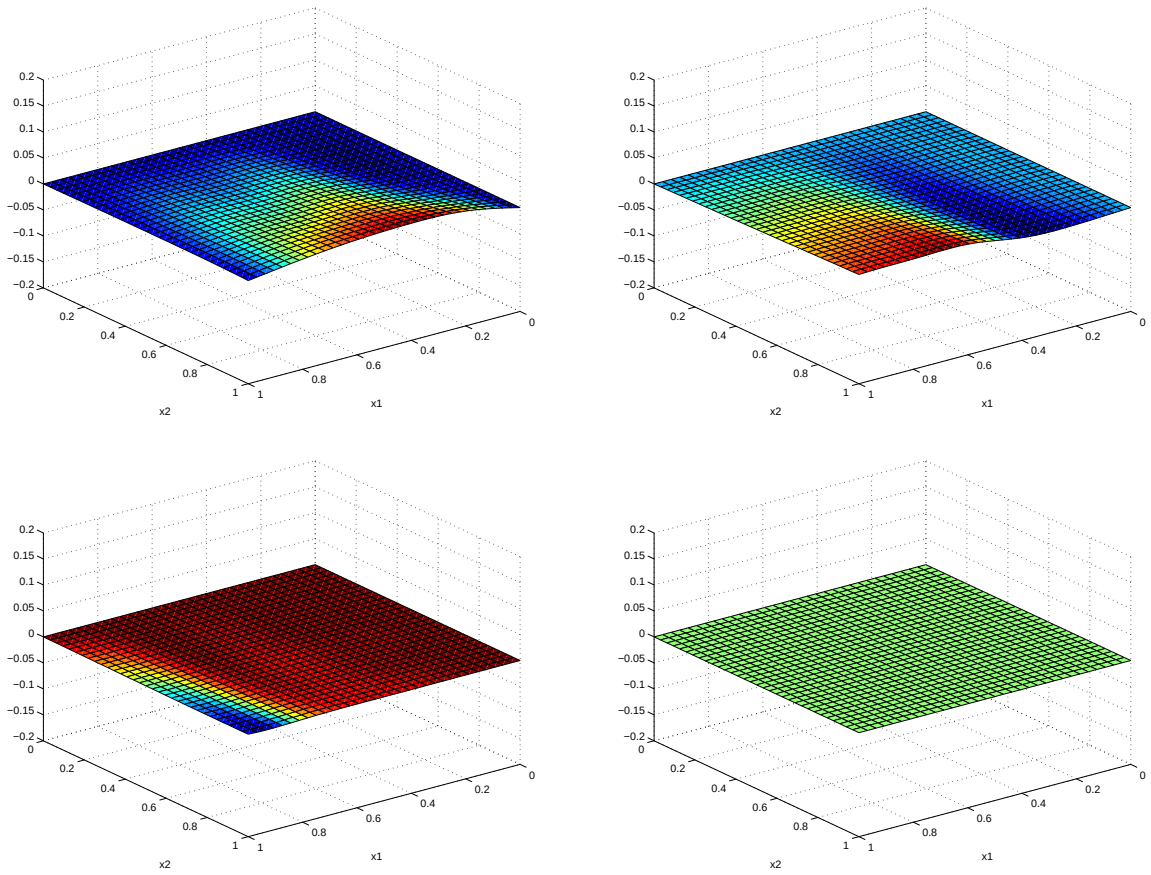


Figura 3.2: Animación del estado $u_1(x, t_k)$ de izquierda a derecha y de arriba a abajo para t_k =0, 0.15, 0.30, 0.45, 0.60, 0.75, 0.90, 1.05, 1.20, 1.50, 2.10, 3.

La figura 3.3, por último, muestra los gráficos de los controles. Llamando $v_1 = \left(v_1^1, v_1^2\right)$ al control en el eje $x_1 = 1$ y $v_2 = \left(v_2^1, v_2^2\right)$ al control sobre $x_2 = 1$, se muestra v_1^1 y v_2^1 ya que, al igual que ocurría con el estado, debido a la simetría de los datos iniciales $v_1^1 = v_1^2$ y $v_2^1 = v_2^2$. El tamaño de paso en la variable temporal es $0,15$.

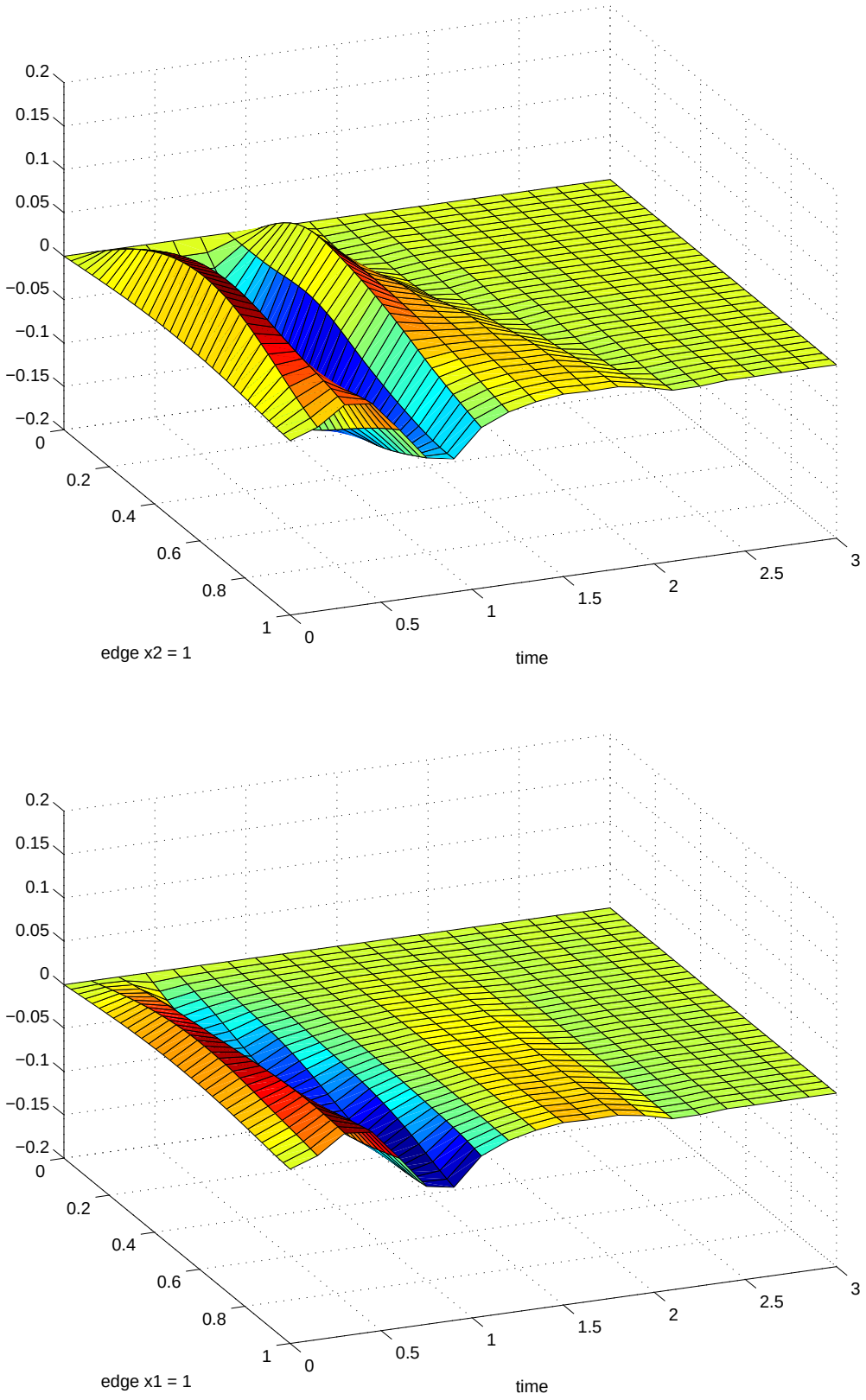


Figura 3.3: Gráficos de los controles $v_1^1(t)$ (arriba) y $v_2^1(t)$ (abajo) para $0 \leq t \leq 3$.

3.2. Círculo unidad con controles en forma de tensiones sobre la frontera

Para esta segunda simulación consideraremos el círculo unidad con controles actuando sobre toda la frontera en forma de tensiones. En este caso nuestro sistema toma la forma:

$$\begin{cases} u_{tt} - \mu\Delta u - (\lambda+\mu)\nabla(\nabla\cdot u) = 0 & \text{en} \quad \Omega\times(0,T) \\ u(x,0) = u^0(x),\ u_t(x,0) = u^1(x) & \text{en} \quad \Omega \\ \sigma\cdot n = v(x,t) & \text{en} \quad \Gamma\times(0,T) \end{cases} \tag{3.3}$$

mientras que el conjunto Ω será:

$$\Omega = \left\{(x_1,x_2)\in\mathbb{R}^2 \ :\ x_1^2+x_2^2<1\right\}.$$

Puesto que los controles actúan sobre toda la frontera, es suficiente con extender las condiciones iniciales con valor nulo fuera de Ω.

Recordando lo visto en el Capítulo 1:

$$\sigma_{ij} = \lambda\varepsilon_{kk}\delta_{ij} + 2\mu\varepsilon_{ij}$$

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right)$$

Así, resulta:

$$\sigma = \begin{pmatrix} \lambda\left(\dfrac{\partial u_1}{\partial x_1}+\dfrac{\partial u_2}{\partial x_2}\right) + 2\mu\dfrac{\partial u_1}{\partial x_1} & \mu\left(\dfrac{\partial u_1}{\partial x_2}+\dfrac{\partial u_2}{\partial x_1}\right) \\ \mu\left(\dfrac{\partial u_1}{\partial x_2}+\dfrac{\partial u_2}{\partial x_1}\right) & \lambda\left(\dfrac{\partial u_1}{\partial x_1}+\dfrac{\partial u_2}{\partial x_2}\right) + 2\mu\dfrac{\partial u_2}{\partial x_2} \end{pmatrix}$$

Puesto que estamos considerando el círculo unidad, el vector unitario normal en la frontera será:

$$\overrightarrow{n}(x_1,x_2) = (x_1,x_2)$$

con lo que el control que debemos calcular valdrá

$$\sigma\cdot n = \begin{pmatrix} \left(\lambda\left(\dfrac{\partial u_1}{\partial x_1}+\dfrac{\partial u_2}{\partial x_2}\right) + 2\mu\dfrac{\partial u_1}{\partial x_1}\right)x_1 + \mu\left(\dfrac{\partial u_1}{\partial x_2}+\dfrac{\partial u_2}{\partial x_1}\right)x_2 \ , \\ \mu\left(\dfrac{\partial u_1}{\partial x_2}+\dfrac{\partial u_2}{\partial x_1}\right)x_1 + \left(\lambda\left(\dfrac{\partial u_1}{\partial x_1}+\dfrac{\partial u_2}{\partial x_2}\right) + 2\mu\dfrac{\partial u_1}{\partial x_1}\right)x_2 \end{pmatrix}$$

Puesto que los desplazamientos se obtienen en el dominio de la transformada de Fourier, podremos calcular de forma sencilla estas derivadas parciales también en el dominio de la transformada de Fourier haciendo

$$\left(\frac{\partial u_i}{\partial x_j}\right)^{\wedge} = \omega_j\widehat{u_i}i$$

para luego aplicar la transformada inversa del mismo modo que hacemos con los desplazamientos.

Para el cálculo de la transformada rápida de Fourier se han usado los mismos parámetros que en el caso anterior. Los tamaños de las mallas son $0,078$ para el ángulo, $0,05$ para el radio y $0,0125$ para el tiempo. Como condiciones iniciales se han tomado

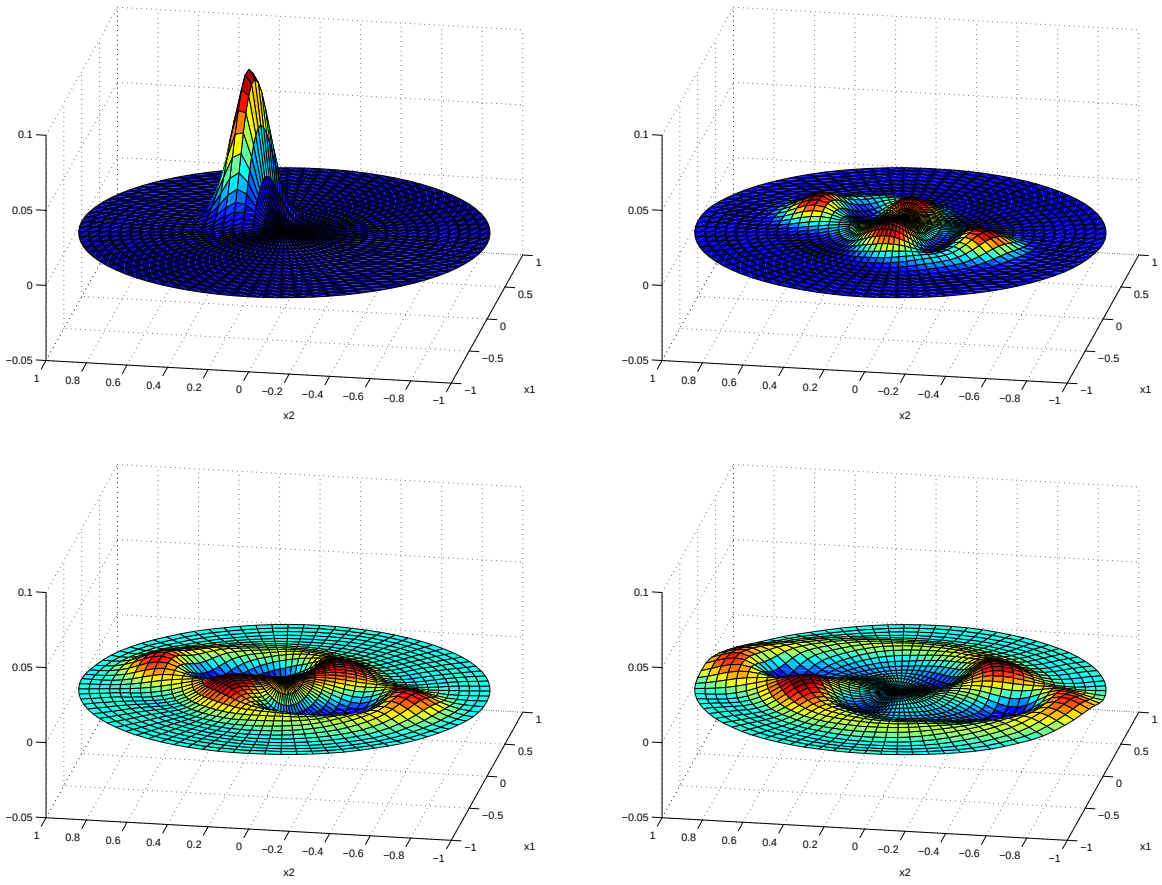
$$u^0(x_1,x_2) = (0{,}1\cdot\exp[-64\cdot((x_1-0{,}2)^2+(x_2-0{,}2)^2)], 0{,}1\cdot\exp[-64\cdot((x_1-0{,}2)^2+(x_2-0{,}2)^2)])$$

$$u^1(x_1,x_2) = (0,0),$$

y como tiempo de controlabilidad $T = 2$.

Con esto, el error obtenido para el estado en $t = T$ es $0,093e-3$ y $0,0265$ para el control en tiempo $t = 0$.

La figura 3.4 muestra, para distintos tiempos, el valor de la componente en x_1 del desplazamiento, representado en el eje z. Debido a la simetría de las condiciones iniciales los resultados obtenidos para la componente x_2 son idénticos.



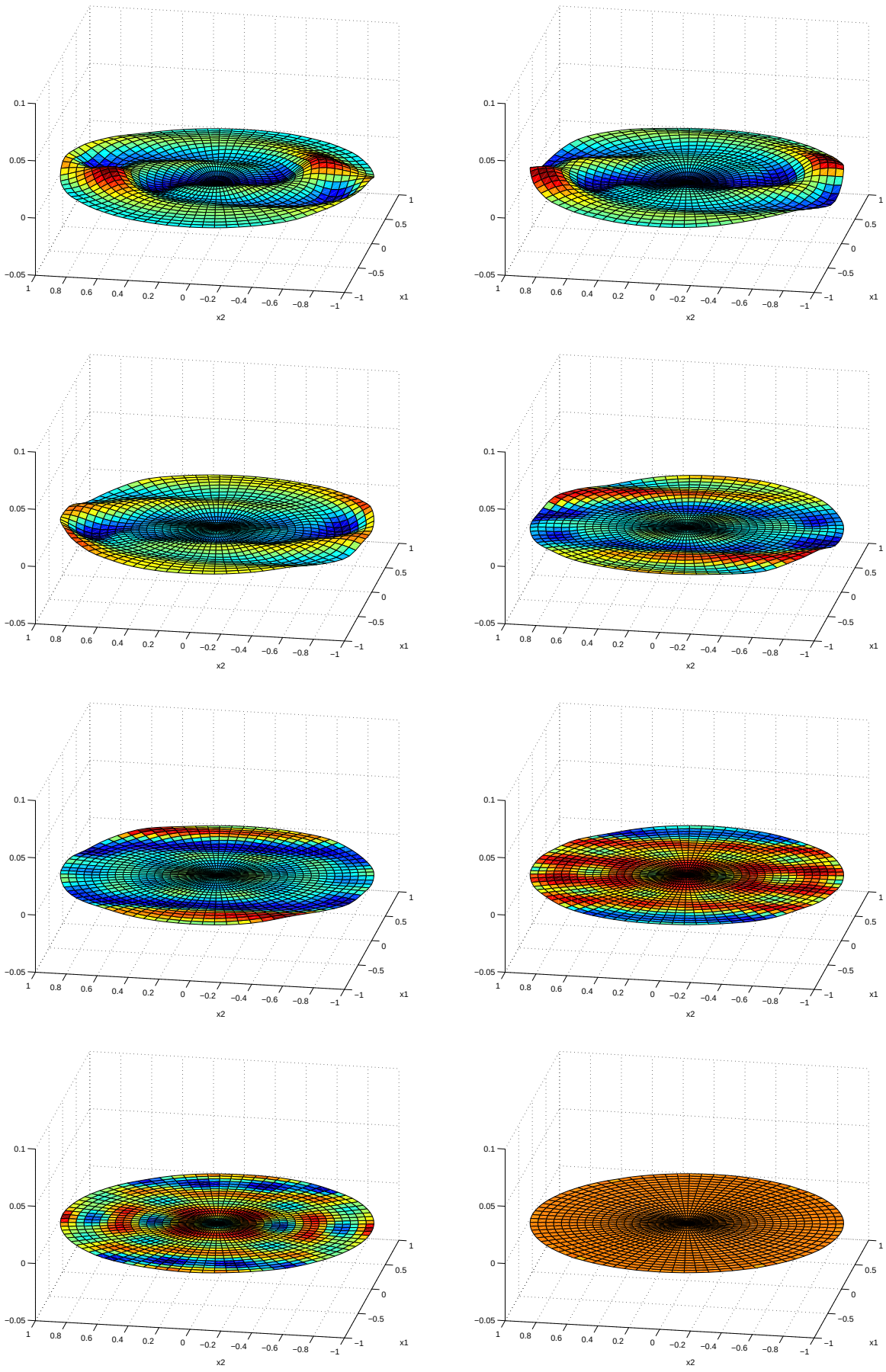
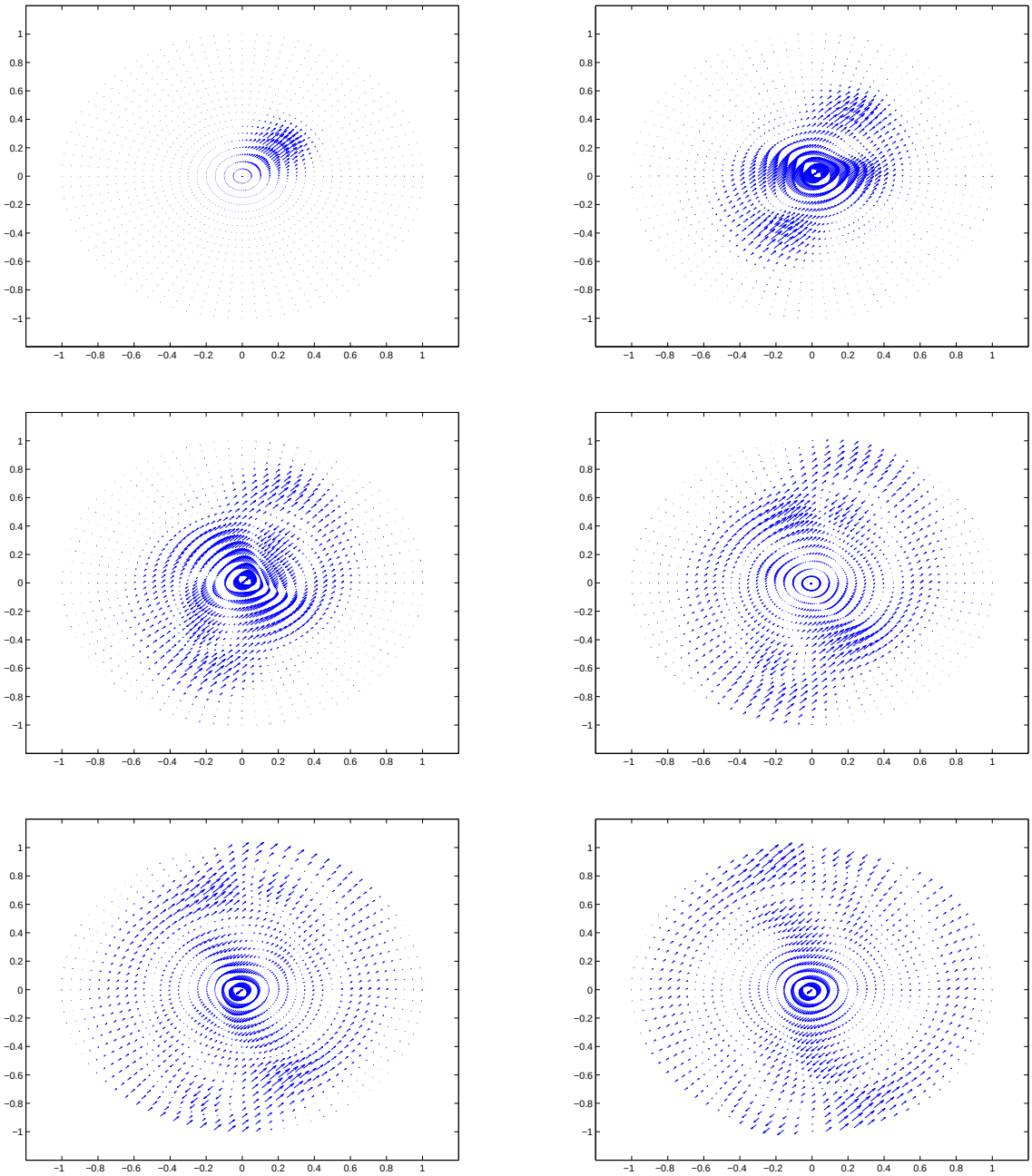


Figura 3.4: Animación del estado $u_1(x, t_k)$ de izquierda a derecha y de arriba a abajo para t_k =0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1.0, 1.3, 1.6, 2.0

En la figura 3.5 se muestra el campo de desplazamientos, para distintos tiempos, en forma de vectores de longitud proporcional al valor del desplazamiento en el punto.

La figura 3.6, por su parte, muestra los gráficos de los controles, $v_1(x_1, x_2)$, componente en x_1, y $v_2(x_1, x_2)$, componente en x_2. El tamaño de paso en la variable temporal es $0,0125$.



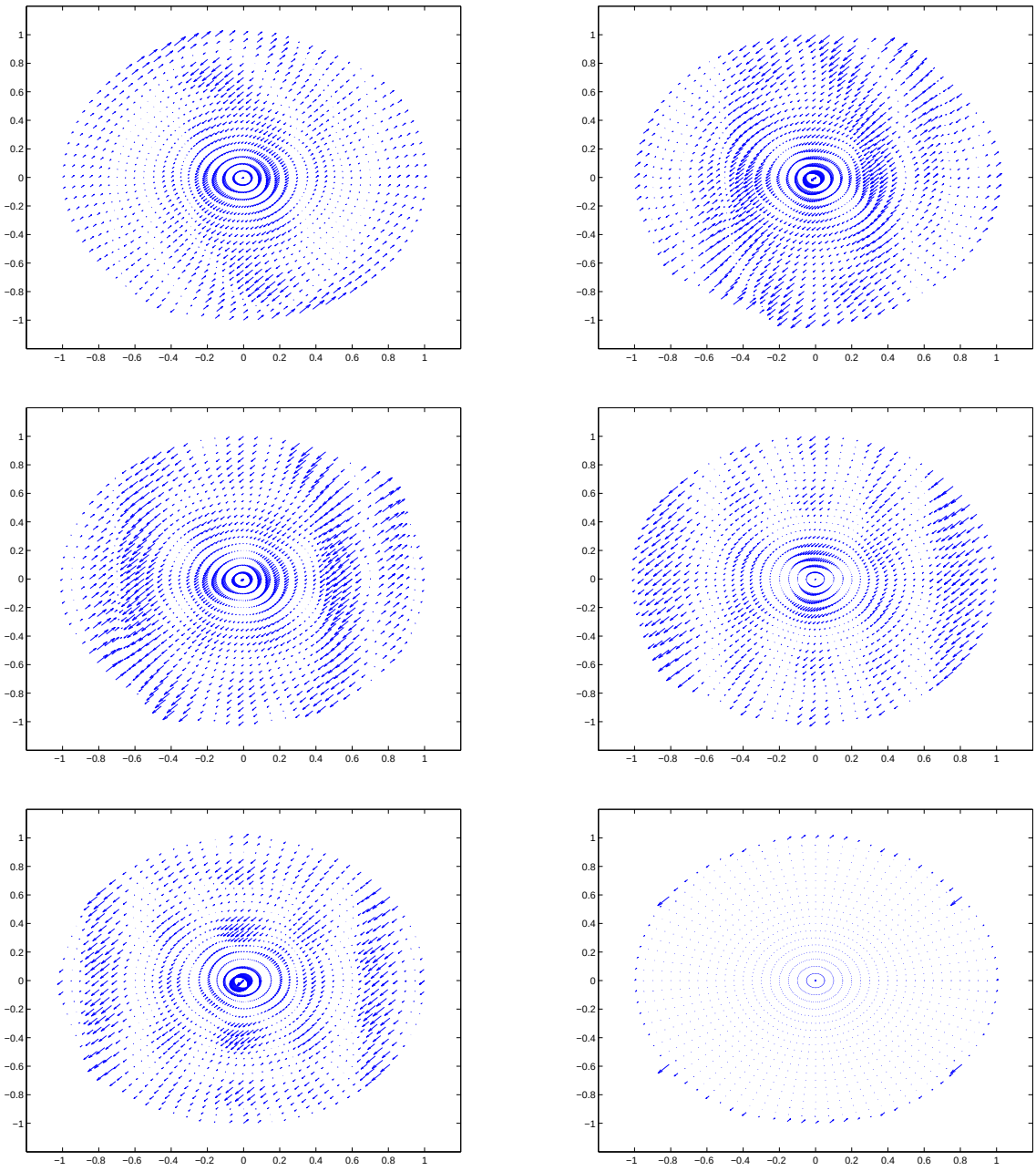


Figura 3.5: Animación del campo de desplazamientos de izquierda a derecha y de arriba a abajo para t_k =0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1.0, 1.3, 1.6, 2.0

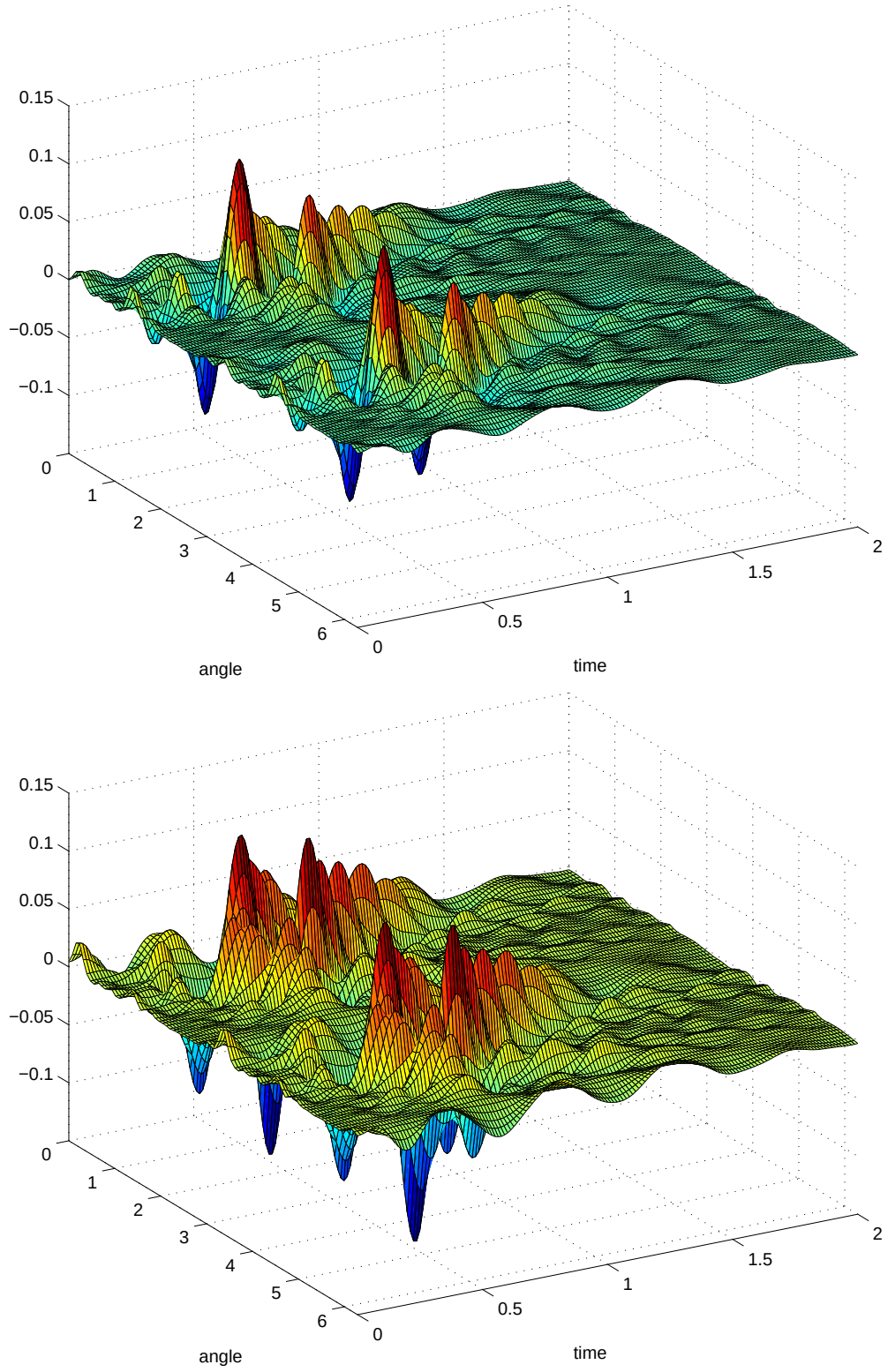


Figura 3.6: Gráficos de la componente en x_1 del control (arriba) y en x_2 (abajo) para $0 \leq t \leq 2$.

CAPÍTULO 4

Conclusiones y problemas abiertos

Se ha empleado un método recientemente propuesto [PPV 08] para resolver numéricamente el problema de control exacto en la frontera para el sitema de la elasticidad lineal, problema cuya resolución mediante otras metodologías presenta serias dificultades. Se ha mostrado, además, la versatilidad del método resolviéndolo para dos diferentes geometrías y controles en la frontera. Como resultado de este trabajo, además del presente Proyecto Fin de Carrera, se ha elaborado un artículo, [FP 09], en la actualidad pendiente de publicación.

La realización del proyecto siguió el mismo esquema lógico descrito en los capítulos 2 y 3. Una vez formulado el problema, el primer paso fue estudiar y comprender el método, el algoritmo que de él se deriva, y las herramientas necesarias para la resolución de los diferentes problemas de Cauchy; en particular la transformada rápida de Fourier y su uso para transformar los sistemas de ecuaciones en derivadas parciales en sistemas de ecuaciones diferenciales ordinarias fácilmente resolubles. Completada esta primera etapa *teórica*, el siguiente paso, y núcleo central del presente trabajo, consistió en aplicar la metodología propuesta a los dos casos considerados, implementando el algoritmo con ayuda del paquete informático MatLab. Los códigos, que pueden consultarse en el Anexo junto con los empleados para el cálculo de la transformada rápida de Fourier y su inversa, debieron ser sometidos, debido a la complejidad del problema, a un exhaustivo proceso de optimización y depuración, con el fin de conseguir soluciones con la calidad requerida. El resultado, aun con toda probabilidad mejorable, permite, en ambos casos, obtener soluciones numéricas de notable precisión en un tiempo de alrededor de una hora.

Entre las principales ventajas del método empleado figuran, como hemos dicho, el ser fácilmente aplicable a problemas con diferentes geometrías y diferentes tipos de control en

la frontera, así como el permitir obtener, de forma simultánea, aproximaciones numéricas tanto del control como del estado. En concreto, el método podría aplicarse a geometrías más complejas que las consideradas, como arcos o láminas de cualquier forma.

Entre las principales limitaciones del método cabe citar que el sistema de ecuaciones en derivadas parciales debe ser lineal y reversible en el tiempo, es decir, no deben aparecer derivadas temporales de primer orden. Físicamente, estas limitaciones implican que para que la metodología sea aplicable debemos despreciar las fuerzas másicas, en particular el peso propio del cuerpo elástico, y cualquier efecto de rozamiento. Aunque estas hipótesis pueden parecer muy restrictivas, aún existen numerosos sistemas reales a los que podría aplicarse nuestra metodología. Citemos, por ejemplo, el caso de antenas o paneles de satélites espaciales; sistemas que, de hecho, suelen requerir en su diseño algún tipo de control de vibraciones.

Anexo. Códigos en MatLab

A.1 Cálculo de la transformada rápida de Fourier

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        fourier2d calcula la Fast Fourier Transform de la funcion Y


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [W1,W2,RDF]=fourier2d(N,L,Y)

K = N/8; %para evitar errores de aliasing

YY = (L^2)*fft2(Y)./N^2; %calculo de la transformada rapida de Fourier
YY = fftshift(YY); %traslacion que acompaña a la FFT

fs = 2*pi/L; %paso en dominio de frecuencias (sampling)
w1 = -K*fs:fs:K*fs; w2 = w1; %sampling points en ambos ejes
[W1,W2] = meshgrid(w1,w2);

trasl = 0.5*L.*(W1+W2); %traslacion

for j=1:(2*K+1)
    for k=1:(2*K+1)
        DF(j,k) = exp(i*trasl(j,k))*YY(N/2-K+j,N/2-K+k);
    end
end

RDF = real(DF); % parte real de la transformada
```

A.2 Cálculo de la transformada rápida inversa de Fourier

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        ifourier2d calcula la inversa FFT de Y

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [X1,X2,RIDF]=ifourier2d(N,L,Y)

K = N/8; %para evitar rerrores de aliasing

YY=((1/(2*pi))^2)*(L^2)*ifft2(Y);
YY=fftshift(YY); %centrando los puntos

fs = 2*pi/L; %paso en dominio de frecuencias (sampling)
x1 = -K*fs:fs:K*fs; x2 = x1; %sampling points en ambos ejes
[X1,X2] = meshgrid(x1,x2);
trasl = 0.5*L.*(X1+X2);
for j=1:(2*K+1)
    for k=1:(2*K+1)
        IDF(j,k) = exp(-i*trasl(j,k))*YY(N/2-K+j,N/2-K+k); %formula para la...
                                                    transformada de la trasladada
    end
end
RIDF = real(IDF); % parte real de la transformada inversa
```

A.3 Cuadrado unidad con controles en desplazamientos sobre dos aristas adyacentes

```
%*****************************************************************
% elast_rect.m calcula la solucion del problema de control frontera
%           para el sistema de la elasticidad en el cuadrado unidad
%           y con controles actuando sobre las aristas x=1 e y=1
%           en la forma de deflection
%
%    PARAMETROS
%
%   T=tiempo de control
%   f=numero de frames por animacion
%   L=longitud del intervalo para la FFT (tomar potencia de 2)
%   N=nº de puntos para la malla temporal de la FFT (tomar potencia de 2)
%
% *****************************************************************


clear all;
close all;

N = input('Introduzca numero de puntos (potencia de 2): '); %2^10
L = input('Introduzca longitud del intervalo: ');%2^5
T = input('Introduzca tiempo de control: '); %3
f = input('Introduzca numero de dibujos para animacion: ');

if (L >= sqrt(pi*N/2))
    display('parametros erroneos');
    return;
end

% parametros de la ecuacion

rho = 1; % densidad
h = 1; % espesor
lambda = 0.5;   % coeficiente primero de Lamé
nu = 1; % coeficiente segundo de Lamé
```

```
% parametros de la malla

dx = L/N; % tamaño de paso para x1, x2
x1 = 0:dx:(L-dx); %puntos de muestreo eje x1 para la funcion a transformar
x2 = x1; %puntos de muestreo eje x2 para la funcion a transformar

K = N/8;
fs = 2*pi/L; %paso en dominio de frecuencias (sampling)
w1 = -K*fs:fs:K*fs; w2 = w1; %sampling points en ambos ejes

% parametros para la animacion en tiempo y el calculo de las  derivadas temporales

dt = T/f; % tamaño de paso del tiempo para la animacion de la solucion
t = 0:dt:T; % malla temporal

% datos iniciales extendidos

for j=1:N
    for k=1:N
        if (x1(j)-L/2 < -1 | x1(j)-L/2 > 1 | x2(k)-L/2 < -1 | x2(k)-L/2 > 1)
            w0_1(j,k) = 0;
        else
            w0_1(j,k) = 0.2*sin(pi*(x1(j)-L/2))*sin(pi*(x2(k)-L/2));
            % posicion inicial
        end
        % velocidad inicial nula
    end
end
w0_2=w0_1;
% figure(1);
% mesh(x1-L/2,x2-L/2,w0_1);
% figure(2);
% mesh(x1-L/2,x2-L/2,w0_2);


%%%%%%%%%%%%%%% Resolucion del primer sistema (P1) con FFT %%%%%%%%%%%%%%%%%%%%

% transformada de Fourier de los datos iniciales
[W1,W2,W0_1]=fourier2d(N,L,w0_1);
```

```
W0_2=W0_1;

% solucion de la ecuacion transformada en tiempo T
for j=1:length(w1)
    for k=1:length(w2)
        p1(j,k) = nu*(w1(j)^2 + w2(k)^2);
        p2(j,k) = (lambda + nu)*(w1(j)*w2(j));
        p3(j,k) = (lambda + nu)*w1(j)^2;
        p4(j,k) = (lambda + nu)*w2(j)^2;

        % datos iniciales del problema transformado
        y0 = [W0_1(j,k); W0_2(j,k); 0; 0];


        A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
             -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];

        y = expm(A*T)*y0;

        Y_1(j,k) = y(1);   % solución transformada en tiempo T
        Y_2(j,k) = y(2);
        Y_1_t(j,k) = y(3); % derivada de la sol. trasnf. en T
        Y_2_t(j,k) = y(4);
    end
end

% solucion de la ecuacion original (P1)

[X1,X2] = meshgrid(x1-L/2,x2-L/2);
% calculamos PHI en los puntos de la malla original interpolando
PHII_1 = interp2(W1,W2,Y_1,X1,X2,'*spline');
PHII_2 = interp2(W1,W2,Y_2,X1,X2,'*spline');
% calculamos PHI_t en los puntos de la malla original interpolando
PHII_1_t = interp2(W1,W2,Y_1_t,X1,X2,'*spline');
PHII_2_t = interp2(W1,W2,Y_2_t,X1,X2,'*spline');

%calculamos la transformada inversa en tiempo T (solucion de (P1))
[W1,W2,phi_1] = ifourier2d(N,L,PHII_1);
[W1,W2,phi_2] = ifourier2d(N,L,PHII_2);
%calculamos la transformada inversa en tiempo T (derivada de solucion de (P1))
```

```
[W1,W2,phi_1_t] = ifourier2d(N,L,PHII_1_t);
[W1,W2,phi_2_t] = ifourier2d(N,L,PHII_2_t);

% figure(3);
% mesh(W1,W2,phi_1);
% figure(4);
% mesh(W1,W2,phi_2);
% figure(5);
% mesh(W1,W2,phi_1_t);
% figure(6);
% mesh(W1,W2,phi_2_t);

%borramos variables para liberar espacio en memoria
clear Y_1 Y_2 Y_1_t Y_2_t PHII_1 PHII_2 PHII_1_t PHII_2_t W0_1 W0_2;


%%%%%%%%%%%%%%% Resolucion del segundo sistema (P2) con FFT %%%%%%%%%%%%%%%%%%%

% Datos iniciales para (P2) : extension de los datos

chi0_1 = -interp2(W1,W2,phi_1,X1,X2,'*spline');
chi0_2 = -interp2(W1,W2,phi_2,X1,X2,'*spline');
chi1_1 = interp2(W1,W2,phi_1_t,X1,X2,'*spline');
chi1_2 = interp2(W1,W2,phi_2_t,X1,X2,'*spline');

for j=1:N
    for k=1:N
        if (x1(j)-L/2 < -1 | x1(j)-L/2 > 1 | x2(k)-L/2 < -1 | x2(k)-L/2 > 1)
            chi0_1(j,k) = 0;
            chi0_2(j,k) = 0;
            chi1_1(j,k) = 0;
            chi1_2(j,k) = 0;
        %2º cuadrante
        elseif (x1(j)-L/2 >= -1 & x1(j)-L/2 < 0 & x2(k)-L/2 >= 0 & x2(k)-L/2 <= 1)
            chi0_1(j,k) = -chi0_1(N+2-j,k);   % posicion inicial
            chi0_2(j,k) = -chi0_2(N+2-j,k);
            chi1_1(j,k) = -chi1_1(N+2-j,k);  % velocidad inicial
            chi1_2(j,k) = -chi1_2(N+2-j,k);
        %3º cuadrante
        elseif (x1(j)-L/2 >= -1 & x1(j)-L/2 < 0 & x2(k)-L/2 >= -1 & x2(k)-L/2 < 0)
```

```
            chi0_1(j,k) = chi0_1(N+2-j,N+2-k);   % posicion inicial
            chi0_2(j,k) = chi0_2(N+2-j,N+2-k);
            chi1_1(j,k) = chi1_1(N+2-j,N+2-k);  % velocidad inicial
            chi1_2(j,k) = chi1_2(N+2-j,N+2-k);
        %4° cuadrante
        elseif (x1(j)-L/2 >= 0 & x1(j)-L/2 <= 1 & x2(k)-L/2 >= -1 & x2(k)-L/2 < 0)
            chi0_1(j,k) = -chi0_1(j,N+2-k);   % posicion inicial
            chi0_2(j,k) = -chi0_2(j,N+2-k);
            chi1_1(j,k) = -chi1_1(j,N+2-k);  % velocidad inicial
            chi1_2(j,k) = -chi1_2(j,N+2-k);
        end
    end
end

chi0_1(N/2+1,:) = 0; chi0_1(:,N/2+1) = 0;
chi0_2(N/2+1,:) = 0; chi0_2(:,N/2+1) = 0;
chi1_1(N/2+1,:) = 0; chi1_1(:,N/2+1) = 0;
chi1_2(N/2+1,:) = 0; chi1_2(:,N/2+1) = 0;


% figure(7);
% mesh(x1-L/2,x2-L/2,chi0_1);
% figure(8);
% mesh(x1-L/2,x2-L/2,chi0_2);
% figure(9);
% mesh(x1-L/2,x2-L/2,chi1_1);
% figure(10);
% mesh(x1-L/2,x2-L/2,chi1_2);

% transformada de Fourier de los datos iniciales

[W1,W2,CHI0_1] = fourier2d(N,L,chi0_1);
[W1,W2,CHI0_2] = fourier2d(N,L,chi0_2);
[W1,W2,CHI1_1] = fourier2d(N,L,chi1_1);
[W1,W2,CHI1_2] = fourier2d(N,L,chi1_2);


% solucion de la ecuacion transformada en tiempo T
for j=1:length(w1)
    for k=1:length(w2)
```

```
        % datos iniciales del problema transformado
        y0 = [CHI0_1(j,k); CHI0_2(j,k); CHI1_1(j,k); CHI1_2(j,k)];

        A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
             -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];

        y = expm(A*T)*y0;

        Y_1(j,k) = y(1);   % solución transformada en tiempo T
        Y_2(j,k) = y(2);
        Y_1_t(j,k) = y(3); % derivada de la sol. trasnf. en T
        Y_2_t(j,k) = y(4);
    end
end


% solucion de la ecuacion original (P2)

% calculamos CHI en los puntos de la malla original interpolando
CHII_1 = interp2(W1,W2,Y_1,X1,X2,'*spline');
CHII_2 = interp2(W1,W2,Y_2,X1,X2,'*spline');
% calculamos CHI_t en los puntos de la malla original interpolando
CHII_1_t = interp2(W1,W2,Y_1_t,X1,X2,'*spline');
CHII_2_t = interp2(W1,W2,Y_2_t,X1,X2,'*spline');

%calculamos la transformada inversa en tiempo T (solucion de (P1))
[W1,W2,chi_1] = ifourier2d(N,L,CHII_1);
[W1,W2,chi_2] = ifourier2d(N,L,CHII_2);
%calculamos la transformada inversa en tiempo T (derivada de solucion de (P1))
[W1,W2,chi_1_t] = ifourier2d(N,L,CHII_1_t);
[W1,W2,chi_2_t] = ifourier2d(N,L,CHII_2_t);

% figure(11);
% mesh(W1,W2,chi_1);
% figure(12);
% mesh(W1,W2,chi_2);
% figure(13);
% mesh(W1,W2,chi_1_t);
% figure(14);
% mesh(W1,W2,chi_2_t);
```

```
%borramos variables para liberar espacio en memoria
clear Y_1 Y_2 Y_1_t Y_2_t CHII_1 CHII_2 CHII_1_t CHII_2_t CHI0_1 CHI0_2...
      CHI1_1 CHI1_2 chi0_1 chi0_2 chi1_1 chi1_2 phi_1 phi_2 phi_1_t phi_2_t;


%%%%%%%% Definimos los datos iniciales para resolver los ultimos sistemas %%%%%%

 %extendemos primero los datos finales de (P2)

chiT_1 = interp2(W1,W2,chi_1,X1,X2,'*spline');
chiT_2 = interp2(W1,W2,chi_2,X1,X2,'*spline');
chiT_1_t = interp2(W1,W2,chi_1_t,X1,X2,'*spline');
chiT_2_t = interp2(W1,W2,chi_2_t,X1,X2,'*spline');

for j=1:N
    for k=1:N
        if (x1(j)-L/2 < -1 | x1(j)-L/2 > 1 | x2(k)-L/2 < -1 | x2(k)-L/2 > 1)
            chiT_1(j,k) = 0;
            chiT_2(j,k) = 0;
            chiT_1_t(j,k) = 0;
            chiT_2_t(j,k) = 0;
        %2° cuadrante
        elseif (x1(j)-L/2 >= -1 & x1(j)-L/2 < 0 & x2(k)-L/2 >= 0 & x2(k)-L/2 <= 1)
            chiT_1(j,k) = -chiT_1(N+2-j,k);   % posicion inicial
            chiT_2(j,k) = -chiT_2(N+2-j,k);
            chiT_1_t(j,k) = -chiT_1_t(N+2-j,k);  % velocidad inicial
            chiT_2_t(j,k) = -chiT_2_t(N+2-j,k);
        %3° cuadrante
        elseif (x1(j)-L/2 >= -1 & x1(j)-L/2 < 0 & x2(k)-L/2 >= -1 & x2(k)-L/2 < 0)
            chiT_1(j,k) = chiT_1(N+2-j,N+2-k);   % posicion inicial
            chiT_2(j,k) = chiT_2(N+2-j,N+2-k);
            chiT_1_t(j,k) = chiT_1_t(N+2-j,N+2-k);  % velocidad inicial
            chiT_2_t(j,k) = chiT_2_t(N+2-j,N+2-k);
        %4° cuadrante
        elseif (x1(j)-L/2 >= 0 & x1(j)-L/2 <= 1 & x2(k)-L/2 >= -1 & x2(k)-L/2 < 0)
            chiT_1(j,k) = -chiT_1(j,N+2-k);   % posicion inicial
            chiT_2(j,k) = -chiT_2(j,N+2-k);
            chiT_1_t(j,k) = -chiT_1_t(j,N+2-k);  % velocidad inicial
            chiT_2_t(j,k) = -chiT_2_t(j,N+2-k);
```

```
        end
    end
end

chiT_1(N/2+1,:) = 0; chiT_1(:,N/2+1) = 0;
chiT_2(N/2+1,:) = 0; chiT_2(:,N/2+1) = 0;
chiT_1_t(N/2+1,:) = 0; chiT_1_t(:,N/2+1) = 0;
chiT_2_t(N/2+1,:) = 0; chiT_2_t(:,N/2+1) = 0;


% ahora corregimos los datos iniciales del problema como en STEP 3

w0star_1 = w0_1-chiT_1;
w0star_2 = w0_2-chiT_2;
w1star_1 = chiT_1_t;
w1star_2 = chiT_2_t;

% figure(15);
% mesh(x1-L/2,x2-L/2,w0star_1);
% figure(16);
% mesh(x1-L/2,x2-L/2,w0star_2);
% figure(17);
% mesh(x1-L/2,x2-L/2,w1star_1);
% figure(18);
% mesh(x1-L/2,x2-L/2,w1star_2);

%borramos variables para liberar espacio en memoria
clear chiT_1 chiT_2 chiT_1_t chiT_2_t chi_1 chi_2 chi_1_t chi_2_t;


%%% Resolvemos de nuevo el sistema (P1) en diferentes tiempos para estos datos %%%

% transformada de Fourier de los datos iniciales

[W1,W2,W0STAR_1] = fourier2d(N,L,w0star_1);
[W1,W2,W0STAR_2] = fourier2d(N,L,w0star_2);
[W1,W2,W1STAR_1] = fourier2d(N,L,w1star_1);
[W1,W2,W1STAR_2] = fourier2d(N,L,w1star_2);

% solucion de la ecuacion transformada en diferentes tiempos
```

```
%
for r=1:length(t)
    for j=1:length(w1)
        for k=1:length(w2)
           % datos iniciales del problema transformado
           y0 = [W0STAR_1(j,k); W0STAR_2(j,k); W1STAR_1(j,k); W1STAR_2(j,k)];

           A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
                -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];

           y = expm(A*t(r))*y0;

           PHISTAR_1(j,k,r) = y(1); % solución transformada en cada tiempo
           PHISTAR_2(j,k,r) = y(2);
           % derivada de la solución (nos quedamos sólo con el valor en t = T)
           TPHISTAR_1(j,k) = y(3);
           TPHISTAR_2(j,k) = y(4);
        end
    end
end

% calculamos la derivada de la solucion de (P1) en tiempo T

% calculamos TPHISTAR en los puntos de la malla original interpolando
TPHIISTAR_1 = interp2(W1,W2,TPHISTAR_1,X1,X2,'*spline');
TPHIISTAR_2 = interp2(W1,W2,TPHISTAR_2,X1,X2,'*spline');
%calculamos la transformada inversa en tiempo T (derivada de solucion de (P1))
[W1,W2,phistar_1_t] = ifourier2d(N,L,TPHIISTAR_1);
[W1,W2,phistar_2_t] = ifourier2d(N,L,TPHIISTAR_2);

% solucion de la ecuacion original (P1) en diferentes tiempos

y1 = 0:dx:1; y2 = y1; %mallas para el calculo de los controles
unos = ones(size(y1));

for r=1:length(t)
    temp_PHISTAR_1(:,:) = PHISTAR_1(:,:,r);
    temp_PHISTAR_2(:,:) = PHISTAR_2(:,:,r);
    % calculamos PHISTAR en los puntos de la malla original interpolando
    PHIISTAR_1 = interp2(W1,W2,temp_PHISTAR_1,X1,X2,'*spline');
```

```
    PHIISTAR_2 = interp2(W1,W2,temp_PHISTAR_2,X1,X2,'*spline');
    %calculamos la transformada inversa (solucion de (P1))
    [W1,W2,temp_phistar_1] = ifourier2d(N,L,PHIISTAR_1);
    [W1,W2,temp_phistar_2] = ifourier2d(N,L,PHIISTAR_2);
    %calculamos phistar en los puntos de la malla original
    phiistar_1 = interp2(W1,W2,temp_phistar_1,X1,X2,'*spline');
    phiistar_2 = interp2(W1,W2,temp_phistar_2,X1,X2,'*spline');
    phistar_1(:,:,r) = phiistar_1(:,:);
    phistar_2(:,:,r) = phiistar_2(:,:);


    %calculo de los controles

    %arista x2 = 1
    temp_f1_1 = interp2(X1,X2,phiistar_1,y1,unos,'*spline');
    temp_f1_2 = interp2(X1,X2,phiistar_2,y1,unos,'*spline');

    %arista x1 = 1
    temp_f2_1 = interp2(X1,X2,phiistar_1,unos,y2,'*spline');
    temp_f2_2 = interp2(X1,X2,phiistar_2,unos,y2,'*spline');

    f1_1(:,r) = temp_f1_1(:);
    f1_2(:,r) = temp_f1_2(:);
    f2_1(:,r) = temp_f2_1(:);
    f2_2(:,r) = temp_f2_2(:);

end

%machacamos phistar para corregir los errores de uso de FFT.
phistar_1(:,:,1) = w0star_1(:,:);
phistar_2(:,:,1) = w0star_2(:,:);

%forzamos 0 en las aristas x1=0 y x2=0
phistar_1(N/2+1,:,length(t)) = 0; phistar_1(:,N/2+1,length(t)) = 0;
phistar_2(N/2+1,:,length(t)) = 0; phistar_2(:,N/2+1,length(t)) = 0;

%save w0star;
%save w1star;

%borramos variables para liberar espacio de memoria
```

```
clear temp_PHISTAR_1 temp_PHISTAR_2 PHIISTAR_1 PHIISTAR_2 w0star_1 w0star_2...
      w1star_1 w1star_2 W0STAR_1 W0STAR_2 W1STAR_1 W1STAR_2 phiistar_1...
      phiistar_2 PHISTAR_1 PHISTAR_2 TPhistar_1 TPhistar_2 TPHISTAR_1...
      TPHISTAR_2 TPHIISTAR_1 TPHIISTAR_2 temp_f1_1 temp_f1_2 temp_f2_1 temp_f2_2;

%%%%%%%%%% Resolvemos finalmente la animacion en tiempo de (P2) %%%%%%%%%%%%%


% extendemos los datos iniciales

phiT_1 = -interp2(W1,W2,temp_phistar_1,X1,X2,'*spline');
phiT_2 = -interp2(W1,W2,temp_phistar_2,X1,X2,'*spline');
phiT_1_t = interp2(W1,W2,phistar_1_t,X1,X2,'*spline');
phiT_2_t = interp2(W1,W2,phistar_2_t,X1,X2,'*spline');

for j=1:N
    for k=1:N
        if (x1(j)-L/2 < -1 | x1(j)-L/2 > 1 | x2(k)-L/2 < -1 | x2(k)-L/2 > 1)
            phiT_1(j,k) = 0;
            phiT_2(j,k) = 0;
            phiT_1_t(j,k) = 0;
            phiT_2_t(j,k) = 0;
        %2º cuadrante
        elseif (x1(j)-L/2 >= -1 & x1(j)-L/2 < 0 & x2(k)-L/2 >= 0 & x2(k)-L/2 <= 1)
            phiT_1(j,k) = -phiT_1(N+2-j,k);   % posicion inicial
            phiT_2(j,k) = -phiT_2(N+2-j,k);
            phiT_1_t(j,k) = -phiT_1_t(N+2-j,k);  % velocidad inicial
            phiT_2_t(j,k) = -phiT_2_t(N+2-j,k);
        %3º cuadrante
        elseif (x1(j)-L/2 >= -1 & x1(j)-L/2 < 0 & x2(k)-L/2 >= -1 & x2(k)-L/2 < 0)
            phiT_1(j,k) = phiT_1(N+2-j,N+2-k);   % posicion inicial
            phiT_2(j,k) = phiT_2(N+2-j,N+2-k);
            phiT_1_t(j,k) = phiT_1_t(N+2-j,N+2-k);  % velocidad inicial
            phiT_2_t(j,k) = phiT_2_t(N+2-j,N+2-k);
        %4º cuadrante
        elseif (x1(j)-L/2 >= 0 & x1(j)-L/2 <= 1 & x2(k)-L/2 >= -1 & x2(k)-L/2 < 0)
            phiT_1(j,k) = -phiT_1(j,N+2-k);   % posicion inicial
            phiT_2(j,k) = -phiT_2(j,N+2-k);
            phiT_1_t(j,k) = -phiT_1_t(j,N+2-k);  % velocidad inicial
            phiT_2_t(j,k) = -phiT_2_t(j,N+2-k);
```

```
        end
    end
end

phiT_1(N/2+1,:) = 0; phiT_1(:,N/2+1) = 0;
phiT_2(N/2+1,:) = 0; phiT_2(:,N/2+1) = 0;
phiT_1_t(N/2+1,:) = 0; phiT_1_t(:,N/2+1) = 0;
phiT_2_t(N/2+1,:) = 0; phiT_2_t(:,N/2+1) = 0;


% transformada de Fourier de los datos iniciales

[W1,W2,CHI0STAR_1] = fourier2d(N,L,phiT_1);
[W1,W2,CHI0STAR_2] = fourier2d(N,L,phiT_2);
[W1,W2,CHI1STAR_1] = fourier2d(N,L,phiT_1_t);
[W1,W2,CHI1STAR_2] = fourier2d(N,L,phiT_2_t);


% solucion de la ecuacion transformada en diferentes tiempos

for r=1:length(t)
    for j=1:length(W1)
        for k=1:length(W2)
            % datos iniciales del problema transformado
            y0 = [CHI0STAR_1(j,k); CHI0STAR_2(j,k); CHI1STAR_1(j,k); CHI1STAR_2(j,k)];

            A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
                 -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];

            y = expm(A*t(r))*y0;

            CHISTAR_1(j,k,r) = y(1); % solución transformada en cada tiempo
            CHISTAR_2(j,k,r) = y(2);
        end
    end
end


% solucion de la ecuacion original (P2) en diferentes tiempos
```

```
for r=1:length(t)
    temp_CHISTAR_1(:,:) = CHISTAR_1(:,:,r);
    temp_CHISTAR_2(:,:) = CHISTAR_2(:,:,r);
    % calculamos CHISTAR en los puntos de la malla original interpolando
    CHIISTAR_1 = interp2(W1,W2,temp_CHISTAR_1,X1,X2,'*spline');
    CHIISTAR_2 = interp2(W1,W2,temp_CHISTAR_2,X1,X2,'*spline');
    %calculamos la transformada inversa (solucion de (P1))
    [W1,W2,temp_chistar_1] = ifourier2d(N,L,CHIISTAR_1);
    [W1,W2,temp_chistar_2] = ifourier2d(N,L,CHIISTAR_2);
    %calculamos chistar en los puntos de la malla original
    chiistar_1 = interp2(W1,W2,temp_chistar_1,X1,X2,'*spline');
    chiistar_2 = interp2(W1,W2,temp_chistar_2,X1,X2,'*spline');
    chistar_1(:,:,r) = chiistar_1(:,:);
    chistar_2(:,:,r) = chiistar_2(:,:);

    %calculo de los controles

    %arista x2 = 1
    temp_g1_1 = interp2(X1,X2,chiistar_1,y1,unos,'*spline');
    temp_g1_2 = interp2(X1,X2,chiistar_2,y1,unos,'*spline');

    %arista x1 = 1
    temp_g2_1 = interp2(X1,X2,chiistar_1,unos,y2,'*spline');
    temp_g2_2 = interp2(X1,X2,chiistar_2,unos,y2,'*spline');

    g1_1(:,r) = temp_g1_1(:);
    g1_2(:,r) = temp_g1_2(:);
    g2_1(:,r) = temp_g2_1(:);
    g2_2(:,r) = temp_g2_2(:);

end

chistar_1(:,:,1) = phiT_1(:,:); %machacamos chistar
chistar_2(:,:,1) = phiT_2(:,:);

clear temp_g1_1 temp_g1_2 temp_g2_1 temp_g2_2 p1 p2 p3 p4 A y0 y CHISTAR_1...
      CHISTAR_2 temp_CHISTAR_1 temp_CHISTAR_2 CHIISTAR_1 CHIISTAR_2...
      temp_chistar_1 temp_chistar_2 temp_phistar_1 temp_phistar_2...
      chiistar_1 chiistar_2 CHI0STAR_1 CHI0STAR_2 CHI1STAR_1...
      CHI1STAR_2 phiT_1 phiT_2 phiT_1_t phiT_2_t
```

```
%%%%%%%%%%%% sumamos ahora los estados y los controles

%malla para representar la solucion

x = 0:dx:1; y = x;
[X,Y] = meshgrid(x,y);

for r=1:length(t)
    temp_sol_1(:,:) = phistar_1(:,:,r) + chistar_1(:,:,length(t)+1-r);
    temp_sol_2(:,:) = phistar_2(:,:,r) + chistar_2(:,:,length(t)+1-r);

    inter_sol_1 = interp2(X1,X2,temp_sol_1,X,Y,'*spline');
    inter_sol_2 = interp2(X1,X2,temp_sol_2,X,Y,'*spline');

    sol_1(:,:,r) = inter_sol_1(:,:);
    sol_2(:,:,r) = inter_sol_2(:,:);
end

for j=1:length(y1)
    for r=1:length(t)
        c1_1(j,r) = f1_1(j,r) + g1_1(j,length(t)+1-r);  %arista x2 = 1
        c1_2(j,r) = f1_2(j,r) + g1_2(j,length(t)+1-r);

        c2_1(j,r) = f2_1(j,r) + g2_1(j,length(t)+1-r);  %arista x1 = 1
        c2_2(j,r) = f2_2(j,r) + g2_2(j,length(t)+1-r);
    end
end

clear temp_sol_1 temp_sol_2 inter_sol_1 inter_sol_2...
      f1_1 f1_2 f2_1 f2_2 g1_1 g1_2 g2_1 g2_2

%save c1_1;
%save c2_1;
%save c1_2;
%save c2_2;

fnames = {'state1_t000', 'state1_t015', 'state1_t030', 'state1_t045',...
              'state1_t060', 'state1_t075', 'state1_t090',...
              'state1_t105', 'state1_t120', 'state1_t135', 'state1_t150',...
```

```
              'state1_t165', 'state1_t180', 'state1_t195',...
              'state1_t210', 'state1_t225', 'state1_t240', 'state1_t255',...
              'state1_t270', 'state1_t285', 'state1_t300',...
              'state2_t000', 'state2_t015', 'state2_t030', 'state2_t045',...
              'state2_t060', 'state2_t075', 'state2_t090',...
              'state2_t105', 'state2_t120', 'state2_t135', 'state2_t150',...
              'state2_t165', 'state2_t180', 'state2_t195',...
              'state2_t210', 'state2_t225', 'state2_t240', 'state2_t255',...
              'state2_t270', 'state2_t285', 'state2_t300','quiver_t000',...
              'quiver_t015', 'quiver_t030', 'quiver_t045',...
              'quiver_t060', 'quiver_t075', 'quiver_t090',...
              'quiver_t105', 'quiver_t120', 'quiver_t135', 'quiver_t150',...
              'quiver_t165', 'quiver_t180', 'quiver_t195',...
              'quiver_t210', 'quiver_t225', 'quiver_t240', 'quiver_t255',...
              'quiver_t270', 'quiver_t285', 'quiver_t300','deformada_t000',...
              'deformada_t015', 'deformada_t030', 'deformada_t045',...
              'deformada_t060', 'deformada_t075', 'deformada_t090',...
              'deformada_t105', 'deformada_t120', 'deformada_t135',...
              'deformada_t150', 'deformada_t165', 'deformada_t180',...
              'deformada_t195', 'deformada_t210', 'deformada_t225',...
              'deformada_t240', 'deformada_t255',...
              'deformada_t270', 'deformada_t285', 'deformada_t300'};


unos = ones(size(X));

for r=1:length(t)
    sol_1_(:,:) = sol_1(:,:,r);
    sol_2_(:,:) = sol_2(:,:,r);
    figure(19);
    surf(x,y,sol_1_);
    axis([0,1,0,1,-0.2,0.2]);
    view(-217,30);
    xlabel('x1');ylabel('x2')
    print('-djpeg',fnames{r});
    print('-depsc',fnames{r});
    M1(r) = getframe;
    figure(20);
    surf(x,y,sol_2_);
    axis([0,1,0,1,-0.2,0.2]);
```

```
    view(-217,30);
    xlabel('x1');ylabel('x2')
    print('-djpeg',fnames{r+21});
    print('-depsc',fnames{r+21});
%     M2(r) = getframe;
    figure(21);
    quiver(x,y,sol_1_,sol_2_);
    print('-djpeg',fnames{r+42});
    print('-depsc',fnames{r+42});
    for j=1:length(x)
        for k=1:length(y)
            Xa(j,k) = X(j,k) + sol_1_(j,k); %mallas para representar la deformada
            Ya(j,k) = Y(j,k) + sol_2_(j,k);
        end
    end
    figure(22);
    surf(Xa,Ya,unos);
    axis([0,1.2,0,1.2,-0.2,0.2]);
    view(0,90);
    print('-djpeg',fnames{r+63});
    print('-depsc',fnames{r+63});
end

% fps = f/T;
% movie2avi(M1,'plate1', 'compression', 'Cinepak', 'fps', fps);
% movie2avi(M2,'plate2', 'compression', 'Cinepak', 'fps', fps);

[Y1,T1] = meshgrid(y1,t);

figure(23); %defection control en x1 en la arista x2 = 1
surf(Y1,T1,c1_1');
axis([0,1,0,T,-0.2,0.2]);
xlabel('edge x2 = 1'); ylabel('time');
view(67.5,30);
print -djpeg Pc1_1;
print -depsc Pc1_1;


figure(24); %deflection control en x1 en la arista x1 =1
surf(Y1,T1,c2_1');
```

```
axis([0,1,0,T,-0.2,0.2]);
xlabel('edge x1 = 1'); ylabel('time');
view(67.5,30);
print -djpeg Pc2_1;
print -depsc Pc2_1;

figure(25); %defection control en x2 en la arista x2 = 1
surf(Y1,T1,c1_2');
axis([0,1,0,T,-0.2,0.2]);
xlabel('edge x2 = 1'); ylabel('time');
view(67.5,30);
print -djpeg Pc1_2;
print -depsc Pc1_2;

figure(26); %defection control en x2 en la arista x1 = 1
surf(Y1,T1,c2_2');
axis([0,1,0,T,-0.2,0.2]);
xlabel('edge x1 = 1'); ylabel('time');
view(67.5,30);
print -djpeg Pc2_2;
print -depsc Pc2_2;
```

A.4 Círculo unidad con controles en forma de tensiones sobre toda la frontera

```
%*****************************************************************
% elast_circ.m calcula la solucion del problema de control frontera
%           para el sistema de la elasticidad en el círculo unidad
%           y con controles actuando sobre la frontera en forma de
%           esfuerzo normal
%
%    PARAMETROS
%
%   T=tiempo de control
%   f=numero de frames por animacion
%   L=longitud del intervalo para la FFT (tomar potencia de 2)
%   N=nº de puntos para la malla temporal de la FFT (tomar potencia de 2)
%
% *****************************************************************


clear all;
close all;

N = input('Introduzca numero de puntos (potencia de 2): '); %2^10
L = input('Introduzca longitud del intervalo: ');%2^5
T = input('Introduzca tiempo de control: '); %2
f = input('Introduzca numero de dibujos para animacion: ');

if (L >= sqrt(pi*N/2))
    display('parametros erroneos');
    return;
end

% parametros de la ecuacion

rho = 1; % densidad
h = 1; % espesor
lambda = 0.5;   % coeficiente primero de Lamé
nu = 1; % coeficiente segundo de Lamé
```

```
% parametros de la malla

dx = L/N; % tamaño de paso para x1, x2
x1 = 0:dx:(L-dx); %puntos de muestreo eje x1 para la funcion a transformar
x2 = x1; %puntos de muestreo eje x2 para la funcion a transformar

K = N/8;
fs = 2*pi/L; %paso en dominio de frecuencias (sampling)
w1 = -K*fs:fs:K*fs; w2 = w1; %sampling points en ambos ejes

% parametros para la animacion en tiempo y el calculo de las  derivadas temporales

dt = T/f; % tamaño de paso del tiempo para la animacion de la solucion
t = 0:dt:T; % malla temporal

% datos iniciales extendidos

for j=1:N
    for k=1:N
        if (((x1(j)-L/2)^2 + (x2(k)-L/2)^2) >= 1)
            w0_1(j,k) = 0;
        else
            % posicion inicial
            w0_1(j,k) = 0.1*exp(-64*((x1(j)-L/2-0.2)^2 + (x2(k)-L/2-0.2)^2));
        end
        % velocidad inicial nula
    end
end
w0_2=w0_1;

% figure(1);
% mesh(x1-L/2,x2-L/2,w0_1);
% axis([-1 1 -1 1 0 1]);
% figure(2);
% mesh(x1-L/2,x2-L/2,w0_2);


%%%%%%%%%%%%%% Resolucion del primer sistema (P1) con FFT %%%%%%%%%%%%%%%%%%
```

```
% transformada de Fourier de los datos iniciales
[W1,W2,W0_1]=fourier2d(N,L,w0_1);
W0_2=W0_1;

% solucion de la ecuacion transformada en tiempo T
for j=1:length(w1)
    for k=1:length(w2)
        p1(j,k) = nu*(w1(j)^2 + w2(k)^2);
        p2(j,k) = (lambda + nu)*(w1(j)*w2(j));
        p3(j,k) = (lambda + nu)*w1(j)^2;
        p4(j,k) = (lambda + nu)*w2(j)^2;

        % datos iniciales del problema transformado
        y0 = [W0_1(j,k); W0_2(j,k); 0; 0];

        A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
             -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];

        Y = expm(A*T)*y0;

        Y_1(j,k) = Y(1);   % solución transformada en tiempo T
        Y_2(j,k) = Y(2);
        Y_1_t(j,k) = Y(3); % derivada de la sol. trasnf. en T
        Y_2_t(j,k) = Y(4);
    end
end

% solucion de la ecuacion original (P1)

[X1,X2] = meshgrid(x1-L/2,x2-L/2);
% calculamos PHI en los puntos de la malla original interpolando
PHII_1 = interp2(W1,W2,Y_1,X1,X2,'*spline');
PHII_2 = interp2(W1,W2,Y_2,X1,X2,'*spline');
% calculamos PHI_t en los puntos de la malla original interpolando
PHII_1_t = interp2(W1,W2,Y_1_t,X1,X2,'*spline');
PHII_2_t = interp2(W1,W2,Y_2_t,X1,X2,'*spline');

%calculamos la transformada inversa en tiempo T (solucion de (P1))
[W1,W2,phi_1] = ifourier2d(N,L,PHII_1);
[W1,W2,phi_2] = ifourier2d(N,L,PHII_2);
```

```
%calculamos la transformada inversa en tiempo T (derivada de solucion de (P1))
[W1,W2,phi_1_t] = ifourier2d(N,L,PHII_1_t);
[W1,W2,phi_2_t] = ifourier2d(N,L,PHII_2_t);

% figure(3);
% mesh(W1,W2,phi_1);
% figure(4);
% mesh(W1,W2,phi_2);
% figure(5);
% mesh(W1,W2,phi_1_t);
% figure(6);
% mesh(W1,W2,phi_2_t);

%borramos variables para liberar espacio en memoria
clear Y_1 Y_2 Y_1_t Y_2_t PHII_1 PHII_2 PHII_1_t PHII_2_t W0_1 W0_2;


%%%%%%%%%%%%%% Resolucion del segundo sistema (P2) con FFT %%%%%%%%%%%%%%%%

% Datos iniciales para (P2) : extension de los datos

chi0_1 = -interp2(W1,W2,phi_1,X1,X2,'*spline');
chi0_2 = -interp2(W1,W2,phi_2,X1,X2,'*spline');
chi1_1 = interp2(W1,W2,phi_1_t,X1,X2,'*spline');
chi1_2 = interp2(W1,W2,phi_2_t,X1,X2,'*spline');

for j=1:N
    for k=1:N
        if (((x1(j)-L/2)^2 + (x2(k)-L/2)^2) >= 1)
            chi0_1(j,k) = 0;
            chi0_2(j,k) = 0;
            chi1_1(j,k) = 0;
            chi1_2(j,k) = 0;
        end
    end
end

% figure(7);
% mesh(x1-L/2,x2-L/2,chi0_1);
% figure(8);
```

```
% mesh(x1-L/2,x2-L/2,chi0_2);
% figure(9);
% mesh(x1-L/2,x2-L/2,chi1_1);
% figure(10);
% mesh(x1-L/2,x2-L/2,chi1_2);

% transformada de Fourier de los datos iniciales

[W1,W2,CHI0_1] = fourier2d(N,L,chi0_1);
[W1,W2,CHI0_2] = fourier2d(N,L,chi0_2);
[W1,W2,CHI1_1] = fourier2d(N,L,chi1_1);
[W1,W2,CHI1_2] = fourier2d(N,L,chi1_2);


% solucion de la ecuacion transformada en tiempo T
for j=1:length(w1)
    for k=1:length(w2)
        % datos iniciales del problema transformado
        y0 = [CHI0_1(j,k); CHI0_2(j,k); CHI1_1(j,k); CHI1_2(j,k)];

        A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
             -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];

        Y = expm(A*T)*y0;

        Y_1(j,k) = Y(1);   % solución transformada en tiempo T
        Y_2(j,k) = Y(2);
        Y_1_t(j,k) = Y(3); % derivada de la sol. trasnf. en T
        Y_2_t(j,k) = Y(4);
    end
end


% solucion de la ecuacion original (P2)

% calculamos CHI en los puntos de la malla original interpolando
CHII_1 = interp2(W1,W2,Y_1,X1,X2,'*spline');
CHII_2 = interp2(W1,W2,Y_2,X1,X2,'*spline');
% calculamos CHI_t en los puntos de la malla original interpolando
CHII_1_t = interp2(W1,W2,Y_1_t,X1,X2,'*spline');
```

```
CHII_2_t = interp2(W1,W2,Y_2_t,X1,X2,'*spline');

%calculamos la transformada inversa en tiempo T (solucion de (P2))
[W1,W2,chi_1] = ifourier2d(N,L,CHII_1);
[W1,W2,chi_2] = ifourier2d(N,L,CHII_2);
%calculamos la transformada inversa en tiempo T (derivada de solucion de (P2))
[W1,W2,chi_1_t] = ifourier2d(N,L,CHII_1_t);
[W1,W2,chi_2_t] = ifourier2d(N,L,CHII_2_t);

% figure(11);
% mesh(W1,W2,chi_1);
% figure(12);
% mesh(W1,W2,chi_2);
% figure(13);
% mesh(W1,W2,chi_1_t);
% figure(14);
% mesh(W1,W2,chi_2_t);

%borramos variables para liberar espacio en memoria
clear Y_1 Y_2 Y_1_t Y_2_t CHII_1 CHII_2 CHII_1_t CHII_2_t CHI0_1 CHI0_2...
CHI1_1 CHI1_2 chi0_1 chi0_2 chi1_1 chi1_2 phi_1 phi_2 phi_1_t phi_2_t;

%%%%%% Definimos los datos iniciales para resolver los ultimos sistemas %%%%%

 %extendemos primero los datos finales de (P2)

chiT_1 = interp2(W1,W2,chi_1,X1,X2,'*spline');
chiT_2 = interp2(W1,W2,chi_2,X1,X2,'*spline');
chiT_1_t = interp2(W1,W2,chi_1_t,X1,X2,'*spline');
chiT_2_t = interp2(W1,W2,chi_2_t,X1,X2,'*spline');

for j=1:N
    for k=1:N
        if (((x1(j)-L/2)^2 + (x2(k)-L/2)^2) >= 1)
            chiT_1(j,k) = 0;
            chiT_2(j,k) = 0;
            chiT_1_t(j,k) = 0;
            chiT_2_t(j,k) = 0;
        end
    end
```

```
end

% ahora corregimos los datos iniciales del problema como en STEP 3

w0star_1 = w0_1-chiT_1;
w0star_2 = w0_2-chiT_2;
w1star_1 = chiT_1_t;
w1star_2 = chiT_2_t;

% figure(15);
% mesh(x1-L/2,x2-L/2,w0star_1);
% figure(16);
% mesh(x1-L/2,x2-L/2,w0star_2);
% figure(17);
% mesh(x1-L/2,x2-L/2,w1star_1);
% figure(18);
% mesh(x1-L/2,x2-L/2,w1star_2);

%borramos variables para liberar espacio en memoria
clear chiT_1 chiT_2 chiT_1_t chiT_2_t chi_1 chi_2 chi_1_t chi_2_t;

%%%%% Resolvemos de nuevo el sistema (P1) en diferentes tiempos para estos datos %%%%

% transformada de Fourier de los datos iniciales

[W1,W2,W0STAR_1] = fourier2d(N,L,w0star_1);
[W1,W2,W0STAR_2] = fourier2d(N,L,w0star_2);
[W1,W2,W1STAR_1] = fourier2d(N,L,w1star_1);
[W1,W2,W1STAR_2] = fourier2d(N,L,w1star_2);

% solucion de la ecuacion transformada en diferentes tiempos

for r=1:length(t)
    for j=1:length(w1)
        for k=1:length(w2)
           % datos iniciales del problema transformado
           y0 = [W0STAR_1(j,k); W0STAR_2(j,k); W1STAR_1(j,k); W1STAR_2(j,k)];

           A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
                -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];
```

```
            Y = expm(A*t(r))*y0;

            PHISTAR_1(j,k,r) = Y(1); % solución transformada en cada tiempo
            PHISTAR_2(j,k,r) = Y(2);
            % derivada de la solución (nos quedamos sólo con el valor en t = T)
            TPHISTAR_1(j,k) = Y(3);
            TPHISTAR_2(j,k) = Y(4);
        end
    end
end

% calculamos la derivada de la solucion de (P1) en tiempo T
% calculamos TPHISTAR en los puntos de la malla original interpolando
TPHIISTAR_1 = interp2(W1,W2,TPHISTAR_1,X1,X2,'*spline');
TPHIISTAR_2 = interp2(W1,W2,TPHISTAR_2,X1,X2,'*spline');
%calculamos la transformada inversa en tiempo T (derivada de solucion de (P1))
[W1,W2,phistar_1_t] = ifourier2d(N,L,TPHIISTAR_1);
[W1,W2,phistar_2_t] = ifourier2d(N,L,TPHIISTAR_2);

% solucion de la ecuacion original (P1) en diferentes tiempos

incr = (2*pi)/80;
ang = 0:incr:(2*pi); rad = 1; %mallas para el calculo de los controles en polares
[theta,rho] = meshgrid(ang,rad);
[x,y] = pol2cart(theta,rho);

for r=1:length(t)
    temp_PHISTAR_1(:,:) = PHISTAR_1(:,:,r);
    temp_PHISTAR_2(:,:) = PHISTAR_2(:,:,r);
    % calculamos PHISTAR en los puntos de la malla original interpolando
    PHIISTAR_1 = interp2(W1,W2,temp_PHISTAR_1,X1,X2,'*spline');
    PHIISTAR_2 = interp2(W1,W2,temp_PHISTAR_2,X1,X2,'*spline');
    %calculamos la transformada inversa (solucion de (P1))
    [W1,W2,temp_phistar_1] = ifourier2d(N,L,PHIISTAR_1);
    [W1,W2,temp_phistar_2] = ifourier2d(N,L,PHIISTAR_2);
    %calculamos phistar en los puntos de la malla original
    phiistar_1 = interp2(W1,W2,temp_phistar_1,X1,X2,'*spline');
    phiistar_2 = interp2(W1,W2,temp_phistar_2,X1,X2,'*spline');
    phistar_1(:,:,r) = phiistar_1(:,:);
```

```
phistar_2(:,:,r) = phiistar_2(:,:);


%calculo de los controles

%Calculamos las derivadas parciales en el dominio de la frecuencia
for j=1:length(w1)
    for k=1:length(w2)
        PARC_u1_x1(j,k) = w1(j)*temp_PHISTAR_1(j,k)*i;
        PARC_u1_x2(j,k) = w2(k)*temp_PHISTAR_1(j,k)*i;
        PARC_u2_x1(j,k) = w1(j)*temp_PHISTAR_2(j,k)*i;
        PARC_u2_x2(j,k) = w2(k)*temp_PHISTAR_2(j,k)*i;
    end
end

%Calculamos las derivadas en la malla original interpolando
IPARC_u1_x1 = interp2(W1,W2,PARC_u1_x1,X1,X2);
IPARC_u1_x2 = interp2(W1,W2,PARC_u1_x2,X1,X2);
IPARC_u2_x1 = interp2(W1,W2,PARC_u2_x1,X1,X2);
IPARC_u2_x2 = interp2(W1,W2,PARC_u2_x2,X1,X2);

%Calculamos la transformada inversa
[W1,W2,iparc_u1_x1] = ifourier2d(N,L,IPARC_u1_x1);
[W1,W2,iparc_u1_x2] = ifourier2d(N,L,IPARC_u1_x2);
[W1,W2,iparc_u2_x1] = ifourier2d(N,L,IPARC_u2_x1);
[W1,W2,iparc_u2_x2] = ifourier2d(N,L,IPARC_u2_x2);

%Calculamos las derivadas parciales interpolando
parc_u1_x1 = interp2(W1,W2,iparc_u1_x1,x,y);
parc_u1_x2 = interp2(W1,W2,iparc_u1_x2,x,y);
parc_u2_x1 = interp2(W1,W2,iparc_u2_x1,x,y);
parc_u2_x2 = interp2(W1,W2,iparc_u2_x2,x,y);

for j=1:length(x)
      f_1(j) = (lambda*(parc_u1_x1(j)+parc_u2_x2(j))+2*nu*parc_u1_x1(j))*x(j)+...
               (parc_u1_x2(j)+parc_u2_x1(j))*nu*y(j);
      f_2(j) = (lambda*(parc_u1_x1(j)+parc_u2_x2(j))+2*nu*parc_u2_x2(j))*y(j)+...
               (parc_u1_x2(j)+parc_u2_x1(j))*nu*x(j);
end
```

```
    f1(:,r) = f_1(:);
    f2(:,r) = f_2(:);

end

%machacamos phistar para corregir los errores de uso de FFT.
phistar_1(:,:,1) = w0star_1(:,:);
phistar_2(:,:,1) = w0star_2(:,:);

%save w0star;
%save w1star;

%borramos variables para liberar espacio de memoria
clear f_1 f_2 PARC_u1_x1 PARC_u1_x2 PARC_u2_x1 PARC_u2_x2 IPARC_u1_x1...
      IPARC_u1_x2 IPARC_u2_x1 IPARC_u2_x2 iparc_u1_x1 iparc_u1_x2 iparc_u2_x1...
      iparc_u2_x2 parc_u1_x1 parc_u1_x2 parc_u2_x1 parc_u2_x2 temp_PHISTAR_1...
      temp_PHISTAR_2 PHIISTAR_1 PHIISTAR_2 w0star_1 w0star_2 w1star_1 w1star_2...
      W0STAR_1 W0STAR_2 W1STAR_1 W1STAR_2 phiistar_1 phiistar_2 PHISTAR_1...
      PHISTAR_2 TPhistar_1 TPhistar_2 TPHISTAR_1 TPHISTAR_2 TPHIISTAR_1 TPHIISTAR_2;

%%%%%%%%% Resolvemos finalmente la animacion en tiempo de (P2) %%%%%%%%%%%%

% extendemos los datos iniciales

phiT_1 = -interp2(W1,W2,temp_phistar_1,X1,X2,'*spline');
phiT_2 = -interp2(W1,W2,temp_phistar_2,X1,X2,'*spline');
phiT_1_t = interp2(W1,W2,phistar_1_t,X1,X2,'*spline');
phiT_2_t = interp2(W1,W2,phistar_2_t,X1,X2,'*spline');

for j=1:N
    for k=1:N
        if (((x1(j)-L/2)^2 + (x2(k)-L/2)^2) >= 1)
            phiT_1(j,k) = 0;
            phiT_2(j,k) = 0;
            phiT_1_t(j,k) = 0;
            phiT_2_t(j,k) = 0;
        end
    end
end
```

```
% transformada de Fourier de los datos iniciales

[W1,W2,CHI0STAR_1] = fourier2d(N,L,phiT_1);
[W1,W2,CHI0STAR_2] = fourier2d(N,L,phiT_2);
[W1,W2,CHI1STAR_1] = fourier2d(N,L,phiT_1_t);
[W1,W2,CHI1STAR_2] = fourier2d(N,L,phiT_2_t);

% solucion de la ecuacion transformada en diferentes tiempos

for r=1:length(t)
    for j=1:length(W1)
        for k=1:length(W2)
            % datos iniciales del problema transformado
            y0 = [CHI0STAR_1(j,k); CHI0STAR_2(j,k); CHI1STAR_1(j,k); CHI1STAR_2(j,k)];

            A = [0 0 1 0; 0 0 0 1; -(p1(j,k)+p3(j,k)) -p2(j,k) 0 0;...
                 -p2(j,k) -(p1(j,k)+p4(j,k)) 0 0];

            Y = expm(A*t(r))*y0;

            CHISTAR_1(j,k,r) = Y(1); % solución transformada en cada tiempo
            CHISTAR_2(j,k,r) = Y(2);
        end
    end
end

clear phiT_1_t phiT_2_t CHI0STAR_1 CHI0STAR_2 CHI1STAR_1 CHI1STAR_2 p1 p2 p3 p4 A y0 Y

% solucion de la ecuacion original (P2) en diferentes tiempos

for r=1:length(t)
    temp_CHISTAR_1(:,:) = CHISTAR_1(:,:,r);
    temp_CHISTAR_2(:,:) = CHISTAR_2(:,:,r);
    % calculamos CHISTAR en los puntos de la malla original interpolando
    CHIISTAR_1 = interp2(W1,W2,temp_CHISTAR_1,X1,X2,'*spline');
    CHIISTAR_2 = interp2(W1,W2,temp_CHISTAR_2,X1,X2,'*spline');
    %calculamos la transformada inversa (solucion de (P1))
    [W1,W2,temp_chistar_1] = ifourier2d(N,L,CHIISTAR_1);
    [W1,W2,temp_chistar_2] = ifourier2d(N,L,CHIISTAR_2);
    %calculamos chistar en los puntos de la malla original
```

```
chiistar_1 = interp2(W1,W2,temp_chistar_1,X1,X2,'*spline');
chiistar_2 = interp2(W1,W2,temp_chistar_2,X1,X2,'*spline');
chistar_1(:,:,r) = chiistar_1(:,:);
chistar_2(:,:,r) = chiistar_2(:,:);

%calculo de los controles

%Calculamos las derivadas parciales en el dominio de la frecuencia
for j=1:length(w1)
    for k=1:length(w2)
        PARC_u1_x1(j,k) = w1(j)*temp_CHISTAR_1(j,k)*i;
        PARC_u1_x2(j,k) = w2(k)*temp_CHISTAR_1(j,k)*i;
        PARC_u2_x1(j,k) = w1(j)*temp_CHISTAR_2(j,k)*i;
        PARC_u2_x2(j,k) = w2(k)*temp_CHISTAR_2(j,k)*i;
    end
end

%Calculamos las derivadas en la malla original interpolando
IPARC_u1_x1 = interp2(W1,W2,PARC_u1_x1,X1,X2);
IPARC_u1_x2 = interp2(W1,W2,PARC_u1_x2,X1,X2);
IPARC_u2_x1 = interp2(W1,W2,PARC_u2_x1,X1,X2);
IPARC_u2_x2 = interp2(W1,W2,PARC_u2_x2,X1,X2);

%Calculamos la transformada inversa
[W1,W2,iparc_u1_x1] = ifourier2d(N,L,IPARC_u1_x1);
[W1,W2,iparc_u1_x2] = ifourier2d(N,L,IPARC_u1_x2);
[W1,W2,iparc_u2_x1] = ifourier2d(N,L,IPARC_u2_x1);
[W1,W2,iparc_u2_x2] = ifourier2d(N,L,IPARC_u2_x2);

%Calculamos las derivadas parciales interpolando
parc_u1_x1 = interp2(W1,W2,iparc_u1_x1,x,y);
parc_u1_x2 = interp2(W1,W2,iparc_u1_x2,x,y);
parc_u2_x1 = interp2(W1,W2,iparc_u2_x1,x,y);
parc_u2_x2 = interp2(W1,W2,iparc_u2_x2,x,y);

for j=1:length(x)
      g_1(j) = (lambda*(parc_u1_x1(j)+parc_u2_x2(j))+2*nu*parc_u1_x1(j))*x(j)+...
               (parc_u1_x2(j)+parc_u2_x1(j))*nu*y(j);
      g_2(j) = (lambda*(parc_u1_x1(j)+parc_u2_x2(j))+2*nu*parc_u2_x2(j))*y(j)+...
               (parc_u1_x2(j)+parc_u2_x1(j))*nu*x(j);
```

```
    end

    g1(:,r) = g_1(:);
    g2(:,r) = g_2(:);
end

chistar_1(:,:,1) = phiT_1(:,:); %machacamos chistar
chistar_2(:,:,1) = phiT_2(:,:);

clear g_1 g_2 phiT_1 phiT_2 CHISTAR_1 CHISTAR_2 temp_CHISTAR_1 temp_CHISTAR_2...
      CHIISTAR_1 CHIISTAR_2 temp_chistar_1 temp_chistar_2 temp_phistar_1...
      temp_phistar_2 chiistar_1 chiistar_2

%%%%%%%%%%%% sumamos ahora los estados y los controles

%malla para representar la solucion

theta = ang; rho = 0:0.05:1; %mallas para el calculo de los controles en polares
[THETA,RHO] = meshgrid(theta,rho);
[X,Y] = pol2cart(THETA,RHO);

for r=1:length(t)
    temp_sol_1(:,:) = phistar_1(:,:,r) + chistar_1(:,:,length(t)+1-r);
    temp_sol_2(:,:) = phistar_2(:,:,r) + chistar_2(:,:,length(t)+1-r);

    inter_sol_1 = interp2(X1,X2,temp_sol_1,X,Y,'*spline');
    inter_sol_2 = interp2(X1,X2,temp_sol_2,X,Y,'*spline');

    sol_1(:,:,r) = inter_sol_1(:,:);
    sol_2(:,:,r) = inter_sol_2(:,:);
end

for j=1:length(x)
    for r=1:length(t)
        c1(j,r) = f1(j,r) + g1(j,length(t)+1-r);   %componente en x1 del control
        c2(j,r) = f2(j,r) + g2(j,length(t)+1-r);   %componente en x2 del control
    end
end

clear temp_sol_1 temp_sol_2 inter_sol_1 inter_sol_2
```

```
%save c1_1;
%save c2_1;
%save c1_2;
%save c2_2;

fnames = {'state1_t00', 'state1_t01', 'state1_t02', 'state1_t03',...
              'state1_t04', 'state1_t05', 'state1_t06',...
              'state1_t07', 'state1_t08', 'state1_t09', 'state1_t10',...
              'state1_t11', 'state1_t12', 'state1_t13',...
              'state1_t14', 'state1_t15', 'state1_t16', 'state1_t17',...
              'state1_t18', 'state1_t19', 'state1_t20',...
              'state2_t00', 'state2_t01', 'state2_t02', 'state2_t03',...
              'state2_t04', 'state2_t05', 'state2_t06',...
              'state2_t07', 'state2_t08', 'state2_t09', 'state2_t10',...
              'state2_t11', 'state2_t12', 'state2_t13',...
              'state2_t14', 'state2_t15', 'state2_t16', 'state2_t17',...
              'state2_t18', 'state2_t19', 'state2_t20','quiver_t00',...
              'quiver_t01', 'quiver_t02', 'quiver_t03',...
              'quiver_t04', 'quiver_t05', 'quiver_t06',...
              'quiver_t07', 'quiver_t08', 'quiver_t09', 'quiver_t10',...
              'quiver_t11', 'quiver_t12', 'quiver_t13',...
              'quiver_t14', 'quiver_t15', 'quiver_t16', 'quiver_t17',...
              'quiver_t18', 'quiver_t19', 'quiver_t20',...
              'deformada_t00', 'deformada_t01', 'deformada_t02',...
              'deformada_t03', 'deformada_t04', 'deformada_t05',...
              'deformada_t06', 'deformada_t07', 'deformada_t08',...
              'deformada_t09', 'deformada_t10', 'deformada_t11',...
              'deformada_t12', 'deformada_t13', 'deformada_t14',...
              'deformada_t15', 'deformada_t16', 'deformada_t17',...
              'deformada_t18', 'deformada_t19', 'deformada_t20'};


unos = ones(size(X));

for r=1:length(t)
    sol_1_(:,:) = sol_1(:,:,r);
    sol_2_(:,:) = sol_2(:,:,r);
    figure(19);
    surf(X,Y,sol_1_);
```

```
    axis([-1,1,-1,1,-0.05,0.1]);
    view(280,30);
    xlabel('x1');ylabel('x2')
    print('-djpeg',fnames{r});
    print('-depsc',fnames{r});
    M1(r) = getframe;
    figure(20);
    surf(X,Y,sol_2_);
    axis([-1,1,-1,1,-0.05,0.1]);
    view(280,30);
    xlabel('x1');ylabel('x2')
    print('-djpeg',fnames{r+21});
    print('-depsc',fnames{r+21});
    figure(21);
    quiver(X,Y,sol_1_,sol_2_);
    axis([-1.2,1.2,-1.2,1.2]);
    print('-djpeg',fnames{r+42});
    print('-depsc',fnames{r+42});
    for j=1:length(rho)
        for k=1:length(theta)
            Xa(j,k) = X(j,k) + sol_1_(j,k); %mallas para representar la deformada
            Ya(j,k) = Y(j,k) + sol_2_(j,k);
        end
    end
    figure(22);
    surf(Xa,Ya,unos);
    axis([-1.2,1.2,-1.2,1.2,-0.15,0.15]);
    view(0,90);
    %print('-djpeg',fnames{r+63});
    print('-depsc',fnames{r+63});
%     M2(r) = getframe;
end

[angulo,Tiempo] = meshgrid(ang,t);

figure(23);%control en x1
surf(angulo,Tiempo,c1');
axis([0,2*pi,0,T,-0.15,0.15]);
view(60,30);
xlabel('angle'); ylabel('time');
```

```
print -depsc pc1;
print -djpeg pc1;

figure(24);%control en x2
surf(angulo,Tiempo,c2');
axis([0,2*pi,0,T,-0.15,0.15]);
view(60,30);
xlabel('angle'); ylabel('time');
print -depsc pc2;
print -djpeg pc2;
```

Bibliografía

[Fol 92] Folland, G. B. *Fourier analysis and its applications*. Pacific Grove, CA: Wadsworth & Brooks/Cole Publishing Company, 1992.

[FP 09] Font, R. J. y Periago, F. Numerical simulation of the boundary exact control for the system of linear elasticity. *Preprint*, 2009.

[Lions 88] Lions, J. L. *Contrôllability exacte, perturbations et stabilization de systèmes distribués*. Tome I, París: Masson, 1988.

[Malv 69] Malvern, L. E. *Introduction to the mechanics of a continuous medium*. Englewood Cliffs, NJ: Prentice-Hall, 1969.

[Par 96] París, F. *Teoría de la elasticidad*. Sevilla: Grupo de Elasticidad y Resistencia de Materiales, Universidad de Sevilla, 1996.

[PPV 08] 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** (2008), 27−47.

[Russ 73] Russell, D. L. A unified boundary controllability theory for hyperbolic and parabolic partial differential equations. *Studies in Applied Mathematics*, **52** : 3 (1973), 189−211.

[Vill 08] Villena Lapaz, J. E. *Simulación numérica del control exacto en la frontera para la ecuación de ondas*. Proyecto fin de carrera, Universidad Politécnica de Cartagena, 2008.