

# **Registro de imágenes médicas**

**Trabajo Fin de Máster**

**Ana Pilar Mateo Sanz**



**Universidad de Zaragoza  
Curso 2014-2015**

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Cálculo de variaciones</b>	<b>4</b>
2.1. Descripción del problema . . . . .	4
2.2. Condiciones de contorno . . . . .	7
<b>3. El registro de imágenes como problema variacional</b>	<b>9</b>
3.1. Medida de similitud . . . . .	10
3.2. Regularización . . . . .	11
3.2.1. Difusión . . . . .	11
3.2.2. Elástico . . . . .	12
<b>4. Resolución numérica</b>	<b>16</b>
4.1. Regularizador de Difusión . . . . .	16
4.2. Regularizador Elástico . . . . .	19
<b>5. Resultados numéricos</b>	<b>33</b>
<b>A. Anexos</b>	<b>35</b>
A.1. Matriz Laplaciano . . . . .	35
A.2. Interpolación . . . . .	36
A.3. Programa Regularizador Difusión . . . . .	37
A.4. Matriz Elasticidad . . . . .	40
A.5. Programa Regularizador Elástico . . . . .	43
<b>Bibliografía</b>	<b>48</b>

## 1. Introducción

El problema de registro en imágenes consiste en encontrar la transformación geométrica que ponga dos imágenes dadas a la mejor correspondencia posible.

El registro de imágenes surge en una gran variedad de aplicaciones como solución a la comparación de una serie de imágenes. Una de estas aplicaciones es la medicina, donde existe la necesidad de detectar cambios en imágenes de un mismo paciente en momentos diferentes, o imágenes del mismo órgano de diferentes pacientes.

El registro de imágenes basado en la intensidad es impulsado por una medida de similitud que tiene dos, o más, imágenes de entrada y calcula un valor numérico que proporciona información acerca de la bondad de ajuste. En el **registro de imágenes rígidas** sólo se aplican rotaciones y traslaciones para realizar la transformación del espacio. Por lo general es necesario para alinear las imágenes de un mismo sujeto sin ningún tipo de deformación. Las principales contribuciones del registro de este tipo de imágenes se compone de métodos automáticos de estimación de parámetros para las medidas de la imagen de similitud, estadísticos y mejoras del rendimiento mediante técnicas de optimización.

Una de las aplicaciones de este tipo de registro de imágenes consiste en reajustar los parámetros óptimos para una imagen realizada mediante una tomografía computarizada por emisión de positrones (PET/TC).

Por otro lado, el **registro de imágenes no rígidas** es una extensión natural del registro de imágenes rígidas, lo que permite también deformaciones, con el fin de lograr una buena concordancia. Esto es necesario en todos los casos en los que se produce un movimiento en el paciente, tales como los latidos del corazón o la respiración. También es necesario cuando hay que registrar los conjuntos de datos de diferentes pacientes, como por ejemplo, en las solicitudes del registro de altas.

Los algoritmos desarrollados se basan en una formulación paramétrica no rígida, y se utilizan en las solicitudes relativas a la compensación de movimiento en general, la tomografía por emisión de positrón (PET) y la tomografía por emisión de fotón único (SPECT) y corrección de la atenuación en las resonancias magnéticas (PET-MR).

El registro de imágenes rígidas y la fusión de datos 3D con imágenes en 2D de Rayos-X son una tarea importante para muchas de las aplicaciones clínicas, como por ejemplo para el tratamiento del cáncer no invasivo mediante radioterapia. Un tipo de algoritmos para el registro de imágenes 2D/3D se basa únicamente en la intensidad de la imagen. Estos métodos también abordan el problema del registro de computación de radiografías reconstruidas digitalmente (DRR) que se comparan con imágenes reales de la radiografía por medio de medidas de similitud. Las principales contribuciones son el desarrollo de técnicas de optimización robustos y rápidos y la utilización de unidades de procesamiento de gráficos modernos para acelerar el cálculo RRD.

A lo largo de este trabajo estudiaremos dos regularizadores no rígidos distintos para hacer frente a la deformación del paciente y compararemos los resultados obtenidos con cada uno de ellos sobre nuestros datos. Estos regularizadores son:

- Difusión
- Elástico

Consideraremos el registro de imágenes como un problema variacional y realizaremos la discretización de cada uno de estos operadores de regularización.

## 2. Cálculo de variaciones

Los procedimientos de minimización forman uno de los medios más amplios en la formulación de modelos matemáticos que regulan el equilibrio en la configuración de sistemas físicos.

Desarrollaremos el análisis matemático básico de los modelos de minimización no lineal en espacios de funciones de dimensión infinita, esto es conocido como el “cálculo de variaciones”.

Las técnicas matemáticas que se han desarrollado para manejar este tipo de problemas de optimización son fundamentales en muchas áreas de las matemáticas, física, ingeniería y otras aplicaciones.

Las soluciones a los problemas clásicos de minimización en el cálculo de variaciones son obtenidas a través de problemas de contorno que implican ciertos tipos de ecuaciones diferenciales conocidas, como las ecuaciones de Euler-Lagrange. La minimización de un funcional cuadrático requiere la solución de un problema de contorno lineal asociado.

### 2.1. Descripción del problema

Dado un dominio  $\Omega$  y un conjunto de funciones diferenciables  $\mathcal{U} : \bar{\Omega} \rightarrow \mathbb{R}^d$  dado por

$$\mathcal{U} := \{u \mid u \in C^2(\bar{\Omega}, \mathbb{R}^d), u \text{ satisface las condiciones de contorno}\}, \quad (1)$$

consideraremos el problema de minimización de un funcional  $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$

$$\mathcal{J}[u] \xrightarrow{u} \min \quad \text{en } \bar{\Omega}. \quad (2)$$

Llamaremos a  $\mathcal{U}$  el espacio de soluciones admisibles. En la Sección 2.2 se darán ejemplos de condiciones de contorno.

Por  $\mathcal{V}$  denotamos el espacio de funciones test, el cual consta de todas las funciones que pueden ser escritas como la diferencia de cualquier par de funciones admisibles,

$$\mathcal{V} := \{v \mid v = u - \hat{u}, u, \hat{u} \in \mathcal{U}\}.$$

Sus condiciones de contorno vendrán dadas según las condiciones de contorno dadas en  $\mathcal{U}$ . Notemos que  $\mathcal{V}$ , por construcción, es un espacio

vectorial.

**Definición 2.1** (Mínimo local). Sea  $\mathcal{U}$  el espacio de soluciones y  $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$  un funcional, se dice que  $\hat{u} \in \mathcal{U}$  es un mínimo local de  $\mathcal{J}$  si existe  $\epsilon > 0$  tal que

$$\mathcal{J}[\hat{u}] \leq \mathcal{J}[u] \quad \forall u \in \mathcal{B}_\epsilon(\hat{u}),$$

donde  $\mathcal{B}_\epsilon(\hat{u}) := \{u \mid u \in \mathcal{U}, \|u - \hat{u}\|_2 < \epsilon\}$  es un entorno de  $\hat{u}$ .

**Definición 2.2** (Derivada Gâteaux). Dado el espacio de soluciones  $\mathcal{U}$ , el espacio de funciones test  $\mathcal{V}$  y el funcional  $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ , se dice que  $\mathcal{J}$  es diferenciable en el sentido de Gâteaux en  $u \in \mathcal{U}$  en la dirección de  $v \in \mathcal{V}$  si:

1.  $\exists \hat{\tau} > 0$  tal que  $u_\tau := u + \tau v \in \mathcal{U}$  para todo  $\tau \in \mathbb{R}$  tal que  $|\tau| \leq \hat{\tau}$ , y
2. la función  $J(\tau) := \mathcal{J}[u_\tau]$  es diferenciable en  $\tau = 0$ .

Se define la derivada de Gâteaux de primer orden de  $\mathcal{J}$  en  $u$  en la dirección de  $v \in \mathcal{V}$  como:

$$\delta \mathcal{J}[u; v] := J'(0) = \left. \frac{d \mathcal{J}[u + \tau v]}{d\tau} \right|_{\tau=0}.$$

Ahora podemos definir el concepto de punto estacionario.

**Definición 2.3** (Punto estacionario). Sean el espacio de soluciones  $\mathcal{U}$ , el espacio de funciones test  $\mathcal{V}$  y el funcional  $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$  anteriormente descritos. Supongamos que en algún  $\hat{u} \in \mathcal{U}$ ,  $\mathcal{J}$  es diferenciable-Gâteaux para todas las funciones test  $v \in \mathcal{V}$ . Entonces decimos que  $\hat{u}$  es un punto estacionario de  $\mathcal{J}$  si

$$\delta \mathcal{J}[\hat{u}; v] = 0 \quad \forall v \in \mathcal{V}.$$

Podemos formular una condición necesaria para ser mínimo local vinculando este concepto con el de punto estacionario.

**Teorema 2.1** (Condición necesaria para ser mínimo local). *Sean el espacio de soluciones  $\mathcal{U}$  y el espacio de funciones test  $\mathcal{V}$  anteriormente dados. Sea  $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$  un funcional que es diferenciable-Gâteaux en  $\hat{u} \in \mathcal{U}$  en todas las direcciones  $v \in \mathcal{V}$ .*

*Si  $\hat{u}$  es un mínimo local de  $\mathcal{J}$ , entonces  $\hat{u}$  es un punto estacionario de  $\mathcal{J}$ .*

Para ilustrar los conceptos introducidos anteriormente, vamos a considerar un funcional de la forma

$$\mathcal{J}[u] := \int_{\Omega} F[x, u(x), \nabla u(x)] dx, \quad (3)$$

donde  $F : \bar{\Omega} \times \mathbb{R}^d \times \mathbb{R}^{d \times d} \rightarrow \mathbb{R}$  es un funcional que depende de  $x$ ,  $u(x)$  y  $\nabla u(x)$ . De aquí en adelante consideraremos que  $\mathcal{J}$  es un funcional diferenciable-Gâteaux en todas las direcciones del espacio test correspondiente. En consecuencia, asumimos que  $F$  tiene derivadas parciales continuas con respecto a todos sus argumentos.

Para distinguir el gradiente usual  $\nabla F = (\partial F / \partial x_1, \dots, \partial F / \partial x_d)^\top$  del gradiente de  $F$  respecto de  $u$ , es decir, respecto del segundo argumento:

$$\nabla_u F = \begin{pmatrix} \partial F / \partial u_1 \\ \vdots \\ \partial F / \partial u_d \end{pmatrix} \in \mathbb{R}^d,$$

hacemos hincapié en la dependencia del gradiente añadiendo un subíndice ' $u$ ' en el operador. De forma similar, el gradiente de  $F$  respecto  $\nabla u$ , es decir, el tercer argumento, viene dado por

$$\nabla_{\nabla u} F = \begin{pmatrix} \partial F / \partial u_{1,1} & \dots & \partial F / \partial u_{1,d} \\ \vdots & & \vdots \\ \partial F / \partial u_{d,1} & \dots & \partial F / \partial u_{d,d} \end{pmatrix} \in \mathbb{R}^{d \times d},$$

y es de nuevo indexado por la variable dependiente. Aquí,  $u_{i,j}$  es la abreviatura de  $\partial u_i / \partial u_j$ .

Podemos ahora caracterizar la condición de ser punto estacionario de  $\mathcal{J}$  para el funcional definido en (3). Consideramos unas condiciones de contorno específicas con las que los espacios  $\mathcal{U}$  y  $\mathcal{V}$  vendrán dadas como

$$\begin{aligned} \mathcal{U} &:= \{u \mid u \in C^2(\bar{\Omega}, \mathbb{R}^d), u = c \text{ en } \partial\Omega\}, \\ \mathcal{V} &:= \{v \mid v \in C^2(\bar{\Omega}, \mathbb{R}^d), v = 0 \text{ en } \partial\Omega\}. \end{aligned}$$

Denotando el producto escalar y el operador de divergencia como  $(\cdot, \cdot)$  y  $\nabla \cdot$  respectivamente, se puede obtener la siguiente condición para ser punto estacionario de  $\mathcal{J}$ .

**Lema 2.2** (Punto estacionario de  $\mathcal{J}$ ). *Una función  $u \in \mathcal{U}$  es un punto estacionario para el funcional  $\mathcal{J}$  definido en (3) si*

$$\int_{\Omega} (\nabla_u F - \nabla \cdot \nabla_{\nabla u} F, v) dx = 0 \quad \forall v \in \mathcal{V}.$$

**Teorema 2.3** (Lema fundamental del cálculo de variaciones). *Sea la función continua  $\theta : \Omega \rightarrow \mathbb{R}^d$ . Entonces,*

$$\int_{\Omega} (\theta, v) dx = 0$$

*para todas las funciones test  $v \in \mathcal{V}$  si y solo si  $\theta \equiv 0$  en  $\Omega$ .*

De los teoremas anteriores, deducimos que  $u \in \mathcal{U}$  es un punto estacionario para el funcional diferenciable-Gâteaux  $\mathcal{J}$  definido en (3) si

$$\nabla_u F - \nabla \cdot \nabla_{\nabla u} F = 0 \quad \text{en } \Omega. \quad (4)$$

Aplicando el Teorema 2.1 podemos concluir que (4) es una condición necesaria para ser mínimo local del funcional (3).

En la práctica se suele considerar  $d > 1$  por lo que (4) constituye un sistema de ecuaciones en derivadas parciales denominadas *ecuaciones de Euler-Lagrange*. Junto a unas condiciones de contorno adecuadas nos encontraremos ante un problema de contorno relacionado con el problema de minimización (2) del funcional definido en (3), denominado formulación variacional.

## 2.2. Condiciones de contorno

En el apartado anterior, al definir el espacio de soluciones en (1), hemos introducido el concepto de condiciones de contorno que deben ser satisfechas para cada función  $u \in \mathcal{U}$ . Lo que implica que cada punto estacionario del funcional  $\mathcal{J}$  satisface también estas condiciones.

Las condiciones de contorno que consideraremos en este trabajo son:

- Condiciones de contorno de tipo Dirichlet en las que se especifican los valores de la función sobre la frontera del dominio.

- Condiciones de contorno de tipo Neumann en las que se especifican los valores de las derivadas de la función en la frontera del dominio.

Si las condiciones de contorno sobre el espacio de soluciones  $\mathcal{U}$  se especifican de forma explícita se denominan *condiciones esenciales*. Al contrario, si las condiciones de contorno no vienen dadas de forma explícita son denominadas como *condiciones naturales*.

Ahora podemos formular la condición necesaria para ser un mínimo de  $\mathcal{J}$  en el caso general.

**Lema 2.4** (Condición Necesaria). *Toda solución  $u^* \in C^2(\bar{\Omega}, \mathbb{R}^d)$  del problema de minimización (2), con  $\mathcal{J}$  el funcional diferenciable-Gâteaux dado en (3), es solución del problema de contorno formado por el sistema de ecuaciones de Euler-Lagrange*

$$\nabla_u F - \nabla \cdot \nabla_{\nabla u} F = 0, \quad \text{en } \Omega,$$

*con condiciones de contorno que pueden ser de tipo esencial (incorporadas en la definición del espacio de soluciones  $\mathcal{U}$ ) o de tipo natural,*

$$(\nabla_{\nabla u_k} F, \vec{n}) = 0 \quad \text{en } \partial\Omega, \quad k = 1, \dots, d, \tag{5}$$

*donde  $\vec{n}$  es el vector normal a la frontera de  $\Omega$ .*

La condición (5) es denominada condición de contorno natural para el problema de minimización dado en (2) con  $\mathcal{J}$  de la forma (3) y que debe de ser satisfecha por la solución del problema de contorno resultante.

### 3. El registro de imágenes como problema variacional

Una imagen puede ser interpretada como una exposición de una escena del mundo real. Normalmente, suponemos que la escena es en tres dimensiones mientras que la exposición muestra una extracción de la escena. Esta extracción puede variar de dimensión, la cual denotaremos por  $d \in \mathbb{N}$ .

Consideramos  $\Omega$  un abierto, acotado y conexo en  $\mathbb{R}^d$ .

Una exposición de una escena puede ser vista como una aplicación desde el dominio de la exposición a un conjunto de valores que representan la intensidad. Supondremos que estos valores de intensidad, en nuestro caso escala de grises, de una exposición pueden ser descritos por una función suficientemente diferenciable.

**Definición 3.1** (Imagen). Dado un dominio  $\Omega$ , una imagen  $I$  viene definida como una función diferenciable  $I : \bar{\Omega} \rightarrow [0, 1]$ .

Además, definimos como  $\text{Img}(\Omega)$  el conjunto de imágenes del dominio  $\Omega$ .

Las imágenes que no vienen dadas por funciones diferenciables son las denominadas *imágenes digitales*.

A continuación vamos a desarrollar un modelo matemático para formular el problema del registro de imágenes. En general, este problema consiste en

*encontrar una transformación geométrica entre dos imágenes.*

Normalmente una de las imágenes es considerada como una imagen de referencia  $R \in \text{Img}(\Omega_R)$  definida en un dominio  $\Omega_R$ , mientras que la otra imagen es considerada como una imagen patrón  $T \in \text{Img}(\Omega_T)$  definida en un dominio  $\Omega_T$ . El dominio de la imagen de referencia está relacionado con el dominio de la imagen patrón a través de una transformación geométrica  $\varphi : \Omega_R \rightarrow \Omega_T$ .

En general una transformación  $\varphi$  tal que  $T \circ \varphi = R$  puede no existir. Sin embargo, buscamos  $\varphi$  de forma que la imagen transformada  $T \circ \varphi$  sea *similar* a la imagen de referencia  $R$ . Para un trato matemático, la *similitud* debe ser medida de alguna forma. Por tanto se requiere una medida de similitud (ó distancia) que asigne un valor determinado al par de imágenes dado. Así

podemos separar  $\varphi$  en la aplicación identidad como la parte trivial y una parte de desplazamiento  $u$ , es decir,  $\varphi(x) := x + u(x)$ .

**Problema 3.1** (Problema de Registro). *Dados dos dominios  $\Omega_T$  y  $\Omega_R$ , las imágenes  $T \in \text{Img}(\Omega_T)$ ,  $R \in \text{Img}(\Omega_R)$ , y  $\mathcal{D} : \text{Img}(\Omega_T) \times \text{Img}(\Omega_R) \times C^p(\bar{\Omega}_R, \mathbb{R}^d) \rightarrow \mathbb{R}_0^+$  como una medida de similitud, el problema de registro de imágenes consiste en encontrar una aplicación*

$$\varphi \in C^p(\bar{\Omega}_R, \bar{\Omega}_T), \quad \varphi(x) := x + u(x),$$

*tal que para algún funcional  $F$*

$$\mathcal{D}_{T,R}[u] := \int_{\Omega_R} F[T \circ \varphi, R] dx \xrightarrow{u} \min.$$

Así, el problema del registro de imágenes puede ser formulado como la minimización del funcional  $\mathcal{D}$  cuya solución es el desplazamiento  $u$  tal que la similitud de  $T \circ \varphi$  y  $R$  se maximiza.

Sin embargo, la minimización del problema 3.1 no nos garantiza una solución única, por lo que debemos de añadir una restricción en la solución  $u$  a través de un factor de regularización  $\mathcal{S}(u)$  para penalizar a las soluciones irregulares e indeseadas.

Así pues, podemos plantear el problema del registro de imágenes como un problema de minimización del funcional dado por

$$\mathcal{J}[\mathbf{u}] = \mathcal{D}_{T,R}[\mathbf{u}] + \alpha \mathcal{S}[\mathbf{u}] \tag{6}$$

con  $\alpha \in \mathbb{R}^+$  un factor de ponderación positivo.

A continuación vamos a describir una medida de similitud  $\mathcal{D}$  y dos factores de regularización llamados de difusión y elástico.

### 3.1. Medida de similitud

Por simplicidad omitiremos la diferencia entre  $\Omega_T$  y  $\Omega_R$  definiendo un único dominio  $\Omega := \Omega_R \cup \Omega_T$ .

**Definición 3.2** (Medida de la suma de las diferencias al cuadrado, SSD). Dadas dos imágenes  $T, R \in \text{Img}(\Omega)$  y un campo de desplazamientos  $\mathbf{u} \in \mathcal{U}$ , definimos la suma de diferencias al cuadrado como

$$\mathcal{D}_{T,R}^{SSD}[\mathbf{u}] := \frac{1}{2} \int_{\Omega} [R(\mathbf{x}) - T_{\mathbf{u}}(\mathbf{x})]^2 d\mathbf{x},$$

donde  $T_u$  es la abreviación de la imagen transformada  $T(\mathbf{x} + \mathbf{u}(\mathbf{x}))$ .

**Lema 3.1** (Ecuaciones de Euler-Lagrange para la medida SSD). *El sistema de ecuaciones de Euler-Lagrange para la medida SSD viene dado por*

$$-\left[R(\mathbf{x}) - T_u(\mathbf{x})\right] \nabla T_u(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega. \quad (7)$$

*Demostración.* Aplicamos el Lema (2.4) para el funcional  $F[\mathbf{x}, \mathbf{u}(\mathbf{x})] = [R(\mathbf{x}) - T_u(\mathbf{x})]^2$ . Basta ver que  $\nabla \cdot \nabla_{\nabla_u} F = 0$  y  $\nabla_{\nabla_u} F = -2 [R(\mathbf{x}) - T_u(\mathbf{x})] \nabla T_u(\mathbf{x})$ .  $\square$

Notemos que el signo no se suprime ya que estamos interesados en el sistema de ecuaciones de Euler-Lagrange para todo el funcional  $\mathcal{J}$ , el cual incluye el término de regularización.

## 3.2. Regularización

Primero estudiaremos el caso más simple, el denominado regularizador de difusión, y continuaremos con el regularizador elástico el cual permite tratar de forma adecuada otros tipos de problemas.

### 3.2.1. Difusión

El regularizador de difusión permite obtener un campo de desplazamientos que es más suave. Fue introducido por FISCHER & MODERSITZKI [6].

**Definición 3.3** (Regularizador de Difusión). Dado un campo de desplazamientos  $\mathbf{u} \in \mathcal{U}$ , el regularizador de difusión se define de la forma

$$\mathcal{S}^{diff}[\mathbf{u}] := \frac{1}{2} \int_{\Omega} \sum_{n=1}^d \| \nabla \mathbf{u}_n(\mathbf{x}) \|_2^2 \, d\mathbf{x}. \quad (8)$$

**Lema 3.2** (Ecuaciones de Euler-Lagrange para el Regularizador de Difusión). *El sistema de ecuaciones de Euler-Lagrange para el regularizador de difusión viene dado por*

$$-\Delta \mathbf{u}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega. \quad (9)$$

*Demostración.* Aplicación del Lema (2.4).

Para  $F := \frac{1}{2} \sum_{n=1}^d \| \nabla \mathbf{u}_n(\mathbf{x}) \|_2^2$ , se tiene que  $\nabla_{\nabla_u} F = 0$  y  $\nabla_{\nabla_u} F$  es el Jacobiano de  $\mathbf{u}$ .  $\square$

Las ecuaciones de Euler-Lagrange (9) pueden ser vistas como un suavizado isotrópico del campo de desplazamiento.

Dado que  $\nabla_{\nabla \mathbf{u}} F = \nabla \mathbf{u}_n$ , las condiciones de contorno naturales para el regularizador de difusión vienen dadas por la condición de Neumann

$$(\nabla \mathbf{u}(\mathbf{x}), \vec{n}(\mathbf{x})) = 0, \quad \mathbf{x} \in \partial\Omega.$$

También podrían utilizarse condiciones de contorno esenciales.

El sistema de ecuaciones de Euler-Lagrange del funcional

$$\mathcal{J}[\mathbf{u}] = \frac{1}{2} \int_{\Omega} [R(\mathbf{x}) - T_{\mathbf{u}}(\mathbf{x})]^2 d\mathbf{x} + \alpha \frac{1}{2} \int_{\Omega} \sum_{n=1}^2 \| \nabla \mathbf{u}_n(\mathbf{x}) \|_2^2 d\mathbf{x},$$

viene dado por el sistema de ecuaciones en derivadas parciales

$$-\alpha \Delta \mathbf{u}(\mathbf{x}) = [T_{\mathbf{u}}(\mathbf{x}) - R(\mathbf{x})] \nabla T_{\mathbf{u}}(\mathbf{x}).$$

### 3.2.2. Elástico

El regularizador elástico se remonta a principios de los años 80 cuando BROIT [?] sugiere la siguiente definición

**Definición 3.4** (Regularizador Elástico). Dado un campo de desplazamiento  $\mathbf{u} \in \mathcal{U}$  y dos escalares  $\lambda \in \mathbb{R}_0^+$  y  $\mu \in \mathbb{R}$ , definimos el regularizador elástico como

$$\mathcal{S}_{\lambda, \mu}^{elas}[\mathbf{u}] := \int_{\Omega} \frac{\mu}{4} \sum_{i,j=1}^d (\partial_{x_j} u_i(x) + \partial_{x_i} u_j(x))^2 + \frac{\lambda}{2} (\nabla \cdot \mathbf{u}(\mathbf{x}))^2 d\mathbf{x}. \quad (10)$$

Esta formulación se basa en el potencial elástico lineal. Se asume un comportamiento isotrópico y tener en cuenta las propiedades del material a través de los parámetros de Lamé  $\lambda$  y  $\mu$ . Estos parámetros pueden relacionarse con el módulo de Young,  $E \in \mathbb{R}_0^+$ , y el coeficiente de Poisson,  $\nu \in [0, 1/2]$ , por las relaciones (cf., e.g. MALVERN, 1969)[8],

$$\begin{aligned} E &= \frac{\mu(2\mu + 3\lambda)}{\mu + \lambda}, & \nu &= \frac{\lambda}{2(\lambda + \mu)}, \\ \lambda &= \frac{Ev}{(1 + \nu)(1 - 2\nu)}, & \mu &= \frac{E}{2(1 + \nu)}. \end{aligned}$$

La cantidad  $1/E$  describe la elasticidad relativa de un objeto en dirección longitudinal dependiendo de las fuerzas aplicadas. Es equivalente a  $1/\mu$  con  $\lambda = 0$ . La cantidad  $\nu$  relaciona la contracción transversal de un objeto con su dilatación longitudinal relativa. Un valor pequeño de  $\nu$  corresponde a tomar  $\lambda \ll \mu$  e indica compresibilidad. Por otro lado, un valor cercano a 0,5 corresponde a la conservación del volumen. En este caso, el objeto es incompresible y se vuelve más delgado conforme se estira. Notar que estos argumentos están basados en la Ley de Hooke, y por tanto son sólo válidos para desplazamientos pequeños.

**Lema 3.3** (Ecuaciones de Euler-Lagrange para el Regularizador Elástico). *El sistema de ecuaciones de Euler-Lagrange para el regularizador elástico viene dado por*

$$-\mu \Delta \mathbf{u}(\mathbf{x}) - (\lambda + \mu) \nabla \cdot \nabla \mathbf{u}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega.$$

*Demostración.* Aplicación del Lema (2.4) tomando  $F = F_1 + F_2$  donde

$$\begin{aligned} F_1 &= \frac{\mu}{4} \sum_{i,j=1}^d (\partial_{x_j} u_i(\mathbf{x}) + \partial_{x_i} u_j(\mathbf{x}))^2 \\ F_2 &= \frac{\lambda}{2} (\nabla \cdot \mathbf{u}(\mathbf{x}))^2 \end{aligned}$$

Se tiene que  $\nabla_{\nabla_{\mathbf{u}}} F_1 / \mu$  es la suma del Jacobiano de  $\mathbf{u}$  y su traspuesta. Por tanto

$$\nabla \cdot \nabla_{\nabla_{\mathbf{u}}} F_1 = \mu \Delta \mathbf{u} + \mu \nabla \cdot \nabla \mathbf{u}.$$

Además,

$$\nabla \cdot \nabla_{\nabla_{\mathbf{u}}} F_2 = \lambda \nabla \cdot \nabla \mathbf{u}.$$

□

Notemos que tomando los valores  $\lambda = -1$  y  $\mu = 1$  este regularizador se simplifica en el regularizador de difusión ya estudiado.

Recordando la definición del funcional (6) se tiene que el funcional a minimizar viene dado por

$$\mathcal{J}[\mathbf{u}] = \frac{1}{2} \int_{\Omega} [R(\mathbf{x}) - T_{\mathbf{u}}(\mathbf{x})]^2 d\mathbf{x} + \alpha \int_{\Omega} \frac{\mu}{4} \sum_{i,j=1}^d (\partial_{x_j} u_i(\mathbf{x}) + \partial_{x_i} u_j(\mathbf{x}))^2 + \frac{\lambda}{2} (\nabla \cdot \mathbf{u}(\mathbf{x}))^2 d\mathbf{x},$$

cuyo sistema de ecuaciones de Euler-Lagrange queda determinado por

$$\alpha \left( -\mu \Delta \mathbf{u}(\mathbf{x}) - (\lambda + \mu) \nabla \cdot \nabla \mathbf{u}(\mathbf{x}) \right) = [T_{\mathbf{u}}(\mathbf{x}) - R(\mathbf{x})] \nabla T_{\mathbf{u}}(\mathbf{x})$$

A continuación introducimos dos conceptos que serán fundamentales en el siguiente capítulo para discretizar el problema de la elasticidad: el **tensor de deformaciones**  $\epsilon$  y el **tensor de tensiones**  $\sigma$ .

**Definición 3.5** (Tensor de deformaciones  $\epsilon$ ). El tensor de deformaciones es un tensor simétrico usado en mecánica de medios continuos y mecánica de sólidos deformables para caracterizar el cambio de forma y volumen de un cuerpo. Su forma general, en tres dimensiones, es

$$\epsilon = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{pmatrix},$$

donde cada uno de los elementos que forman esta matriz son una función cuyo dominio es el conjunto de puntos del cuerpo cuya deformación pretende caracterizarse.

En mecánica de medios continuos se distinguen varios tipos de tensores para representar la deformación, cuando las deformaciones son pequeñas, como ocurre en nuestro caso, es adecuado utilizar el tensor infinitesimal de deformaciones.

Utilizaremos el **tensor infinitesimal de Green-Cauchy**, que constituye una aproximación para caracterizar las deformaciones en el caso de que estas sean pequeñas. En el caso bidimensional dicho tensor se expresa de la forma

$$\epsilon = \frac{1}{2}(\nabla u + (\nabla u)^\top) = \frac{1}{2} \begin{pmatrix} 2u_x & u_y + v_x \\ u_y + v_x & 2v_y \end{pmatrix},$$

donde  $\mathbf{u} = (u, v)$  representa el campo vectorial de desplazamiento del cuerpo y  $\mathbf{x} = (x, y)$  son las coordenadas de cada punto material del cuerpo.

**Definición 3.6** (Tensor de tensiones  $\sigma$ ). En mecánica de medios continuos, el tensor de tensiones es el tensor que da cuenta de la distribución de tensiones y esfuerzos internos en el medio continuo.

Para nuestro problema utilizaremos el **tensor de tensiones de Cauchy**. El teorema de Cauchy sobre las tensiones de un cuerpo establece que, dada una distribución de tensiones internas sobre la geometría de un medio continuo deformado que satisfaga las condiciones del **principio de Cauchy**, existe un campo tensorial  $\sigma$  simétrico definido sobre la geometría deformada con dos de sus propiedades:

- $\nabla \cdot \sigma(\mathbf{u}(\mathbf{x})) + f(\mathbf{x}, \mathbf{u}(\mathbf{x})) = 0,$
- $\sigma(\mathbf{u}(\mathbf{x})) = \sigma(\mathbf{u}(\mathbf{x}))^\top.$

Por la segunda propiedad, deducimos que este tensor vendrá dado sobre las coordenadas específicas por una matriz simétrica. Cuando las deformaciones son pequeñas empleamos este tensor definidos sobre las coordenadas del cuerpo sin deformar. Así, fijado un sistema de referencia ortogonal, el tensor de tensiones de Cauchy viene dado por

$$\sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}.$$

En nuestro problema, definimos el tensor de tensiones de la forma

$$\sigma = 2\mu\epsilon + \lambda \text{tr}(\epsilon)I.$$

Por tanto, en el caso bidimensional se tiene que:

$$\sigma = \begin{pmatrix} (2\mu + \lambda)u_x + \lambda v_y & \mu(u_y + v_x) \\ \mu(u_y + v_x) & (2\mu + \lambda)v_y + \lambda u_x \end{pmatrix}.$$

Y así las ecuaciones de Euler-Lagrange se pueden escribir también como

$$\alpha \nabla \cdot \sigma(\mathbf{u}(\mathbf{x})) = [T_u(\mathbf{x}) - R(\mathbf{x})] \nabla T_u(\mathbf{x}).$$

## 4. Resolución numérica

### 4.1. Regularizador de Difusión

Recordemos que las ecuaciones de Euler-Lagrange que se obtienen para este regularizador son

$$-\alpha \Delta \mathbf{u}(\mathbf{x}) = [T_{\mathbf{u}}(\mathbf{x}) - R(\mathbf{x})] \nabla T_{\mathbf{u}}(\mathbf{x}).$$

Una forma de resolver las ecuaciones en derivadas parciales no lineales es añadir un término temporal  $\partial_t \mathbf{u}$  y calcular la correspondiente solución estacionaria. De esta forma, resulta el siguiente sistema de ecuaciones en derivadas parciales no lineales

$$\partial_t \mathbf{u}(\mathbf{x}, t) = [T_{\mathbf{u}}(\mathbf{x}, t) - R(\mathbf{x}, t)] \nabla T_{\mathbf{u}}(\mathbf{x}, t) + \alpha \Delta \mathbf{u}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, t \geq 0.$$

Para discretizar este problema, vamos a usar un esquema semi-implícito que Consiste en tratar el término difusivo de forma implícita y el de reacción de forma explícita. Dada una malla uniforme temporal con paso  $\tau$ , la discretización correspondiente es

$$\frac{\mathbf{u}(\mathbf{x}, t_{k+1}) - \mathbf{u}(\mathbf{x}, t_k)}{\tau}, \quad y - \alpha \Delta \mathbf{u}(\mathbf{x}, t_{k+1}) = f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)), \quad k = 0, 1, \dots$$

donde

$$f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)) = [T_{\mathbf{u}}(\mathbf{x}, t) - R(\mathbf{x}, t)] \nabla T_{\mathbf{u}}(\mathbf{x}, t)$$

$\mathbf{u}(\mathbf{x}, t_0)$  se toma por simplicidad  $\mathbf{u}(\mathbf{x}, t_0) = 0$ .

Una propiedad importante del sistema de ecuaciones en derivadas parciales a resolver es que es desacoplado, esto es, podemos obtener una aproximación numérica  $u_i, i = 1, 2$ , de cada una de las ecuaciones de forma independiente. En lo siguiente, por conveniencia denotaremos a las componentes de  $\mathbf{u}$  como  $(u, v)$  en lugar de  $(u_1, u_2)$ . Obtenemos el problema para la incógnita  $u$ :

$$(Dif_u) = \begin{cases} \frac{u(\mathbf{x}, t_{k+1}) - u(\mathbf{x}, t_k)}{\tau} = \alpha \Delta u(\mathbf{x}, t_{k+1}) + f_1(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)), & \mathbf{x} \in \Omega, k = 0, 1, \dots, \\ u(\mathbf{x}, t_0) = 0, & \mathbf{x} \in \Omega, \\ \frac{\partial u(\mathbf{x}, t_{k+1})}{\partial \vec{n}} = 0 & \mathbf{x} \in \partial \Omega, k = 0. \end{cases} \quad (11)$$

Para la segunda componente  $v$  obtendremos un problema similar que denotaremos aquí por  $(Dif_v)$ .

A continuación realizamos la discretización espacial del problema  $(Dif_u)$  descrito en (11) mediante el método de diferencias finitas. Este método nos permite la resolución aproximada de ecuaciones diferenciales en derivadas parciales definidas en un recinto finito, en nuestro caso  $\Omega = (0, 1)^2$ . Es de una gran sencillez conceptual y constituye un procedimiento muy adecuado para la resolución de una ecuación bidimensional como la que se nos ha planteado.

Comenzamos construyendo una malla uniforme  $\Omega_h$  de paso  $h = 1/(n - 1)$  en ambas direcciones:

$$\Omega_h = \{x_{i,j} = ((i - 1)h, (j - 1)h) \mid i, j = 1, \dots, n\}.$$

Aproximando el Laplaciano de la forma usual,

$$\Delta_h u_{i,j} = \frac{u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1}}{h^2} \quad i, j = 2, \dots, n - 1.$$

Obtenemos la siguiente ecuación para cada uno de los nodos interiores

$$(I - \tau \alpha \Delta_h) u_{i,j}^{k+1} = u_{i,j}^k + \tau f_{i,j}^k, \quad i, j = 2, \dots, n - 1, k = 0, 1, \dots,$$

donde  $u_{i,j}^k$  denota la solución aproximada en el nodo  $x_{i,j}$  y en el tiempo  $t_k$ .

Eso es válido para los nodos interiores de la malla. En la frontera hay que tener en cuenta las condiciones de contorno que se nos especifican en la definición del problema (11). Aquí emplearemos la técnica del nodo fantasma, la cual consiste en considerar nodos fuera del dominio para aplicar la condición de contorno.

Ilustramos esta técnica considerando un nodo arbitrario en el lado derecho del cuadrado, es decir, cuando  $i = n$  y  $j = 2, \dots, n - 1$ .

Por la condición de contorno sabemos que

$$\frac{\partial u}{\partial \vec{n}} = 0 \quad \Rightarrow \quad \left( \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) \cdot \vec{n} = 0 \quad \stackrel{\vec{n}=(1,0)}{\Rightarrow} \quad \frac{\partial u}{\partial x} = 0.$$

Consideramos la aproximación de diferencias centrales de segundo orden para la primera derivada

$$(u_x)_{i,j} \approx \frac{u_{n+1,j} - u_{n-1,j}}{2h} \quad \stackrel{u_x=0}{\Rightarrow} \quad u_{n+1,j} = u_{n-1,j}.$$

Finalmente sustituyendo el valor de  $u_{n+1,j}$  en la discretización del Laplaciano para el nodo  $n, j$  resulta

$$\Delta_h u_{n,j} = \frac{2u_{n-1,j} + u_{n,j+1} - 4u_{n,j} + u_{n,j-1}}{h^2} \quad j = 2, \dots, n-1.$$

De la misma forma procedemos con el resto de nodos en los lados de la frontera obteniendo:

$$\begin{aligned}\Delta_h u_{1,j} &= \frac{2u_{2,j} + u_{1,j+1} - 4u_{1,j} + u_{1,j-1}}{h^2} & j &= 2, \dots, n-1. \\ \Delta_h u_{i,n} &= \frac{u_{i+1,n} + 2u_{i,n-1} - 4u_{i,n} + u_{i-1,n}}{h^2} & i &= 2, \dots, n-1. \\ \Delta_h u_{i,1} &= \frac{u_{i+1,1} + 2u_{i,2} - 4u_{i,1} + u_{i-1,1}}{h^2} & i &= 2, \dots, n-1.\end{aligned}$$

En cuanto a los vértices se procede de forma análoga. Por ejemplo para el vértice superior derecho tendríamos las condiciones de contorno

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial y} = 0.$$

Realizando el proceso análogo al caso de los nodos en los lados de la frontera, se obtienen las discretizaciones del Laplaciano siguientes:

$$\begin{aligned}\Delta_h u_{n,n} &= \frac{2u_{n-1,n} + 2u_{n,n-1} - 4u_{n,n}}{h^2}, \\ \Delta_h u_{1,n} &= \frac{2u_{2,n} + 2u_{1,n-1} - 4u_{1,n}}{h^2}, \\ \Delta_h u_{1,n} &= \frac{2u_{2,n} + 2u_{1,n-1} - 4u_{1,n}}{h^2}, \\ \Delta_h u_{n,1} &= \frac{2u_{n-1,1} + 2u_{n,2} - 4u_{n,1}}{h^2}.\end{aligned}$$

A partir de la discretización del Laplaciano formamos una matriz  $L_h$  cuyos coeficientes en cada elemento  $(i, j)$  son los coeficientes que acompañan a cada  $u_{i,j}$ . En el Anexo A.1 se presenta el programa de Matlab que construye dicha matriz.

Finalmente los sistemas a resolver en cada paso de tiempo son:

$$A U^{k+1} = b_1, \text{ y } A V^{k+1} = b_2. \quad (12)$$

donde

$$A = I - \tau \alpha L_h,$$

y  $U^{k+1}, V^{k+1}$  son los vectores de incógnitas compuestos por las aproximaciones de las soluciones en los nodos de la malla.

Para finalizar veamos cómo se calcula el término  $b_1$  para cada iteración,  $b_2$  se calcula de forma análoga. Para ello recordamos

$$f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)) = [T_{\mathbf{u}}(\mathbf{x}, t_k) - R(\mathbf{x})] \nabla T_{\mathbf{u}}(\mathbf{x}, t_k)$$

Para calcular  $T_{\mathbf{u}}(\mathbf{x}, t_k)$  en cada iteración realizamos una interpolación bilineal de la imagen  $T$  ya que para cada píxel  $(i, j)$  de la imagen este término queda determinado por

$$(T_u^k)_{i,j} = T(x_{i,j} + u_{i,j}^k, y_{i,j} + v_{i,j}^k).$$

Notemos que va variando en cada paso de tiempo, ya que depende de las soluciones  $u$  y  $v$  obtenidas en el paso de tiempo anterior. La interpolación utilizada puede verse con más detalle en el Anexo A.2.

Para el término  $\nabla T_{\mathbf{u}}(\mathbf{x}, t_k)$  realizamos una aproximación por diferencias finitas centrales.

En la Sección 5 se presentarán resultados obtenidos este método para distintas imágenes y comentaremos la elección de los parámetros  $\tau$  y  $\alpha$ .

## 4.2. Regularizador Elástico

Recordemos que las ecuaciones de Euler-Lagrange que se obtienen para este regularizador son

$$\alpha \nabla \cdot \sigma(\mathbf{u}(\mathbf{x})) = [T_{\mathbf{u}}(\mathbf{x}) - R(\mathbf{x})] \nabla T_{\mathbf{u}}(\mathbf{x}).$$

Resolvemos la ecuación en derivadas parciales añadiendo un término artificial de la misma forma que hicimos para el regularizador anterior, así nos queda

$$\partial_t \mathbf{u}(\mathbf{x}, t) = f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t)) + \alpha \nabla \cdot \sigma(\mathbf{u}(\mathbf{x}, t)), \quad \mathbf{x} \in \Omega, t \geq 0$$

donde

$$f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t)) = [T_{\mathbf{u}}(\mathbf{x}, t) - R(\mathbf{x}, t)] \nabla T_{\mathbf{u}}(\mathbf{x}, t)$$

Utilizando el esquema semi-implícito como en el caso anterior se tiene que la ecuación en derivadas parciales a resolver es

$$\frac{\mathbf{u}(\mathbf{x}, t_{k+1}) - \mathbf{u}(\mathbf{x}, t_k)}{\tau} - \alpha \nabla \cdot \sigma(\mathbf{u}(\mathbf{x}, t_{k+1})) = f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)), \quad \mathbf{x} \in \Omega, k = 0, 1, \dots$$

Añadiendo las condiciones de contorno y el desplazamiento inicial, obtenemos el siguiente problema a resolver

$$(Elas) = \begin{cases} \frac{\mathbf{u}(\mathbf{x}, t_{k+1}) - \mathbf{u}(\mathbf{x}, t_k)}{\tau} = \alpha \nabla \cdot \sigma(\mathbf{u}(\mathbf{x}, t_{k+1})) + f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)), & \mathbf{x} \in \Omega, k = 0, 1, \dots \\ \mathbf{u}(\mathbf{x}, t_0) = 0, & \mathbf{x} \in \Omega, \\ \dot{\sigma} \mathbf{n} = 0 & \mathbf{x} \in \partial\Omega, k = 0, 1, \dots \end{cases} \quad (13)$$

Observamos que en este caso el sistema de ecuaciones en derivadas parciales a resolver está acoplado.

El método empleado para la discretización del problema también será distinto, puesto que el método en diferencias finitas se vuelve demasiado complejo, sobre todo al tratar las condiciones de contorno. Por ello utilizaremos el método de volúmenes finitos, el cual es una alternativa a los métodos de diferencias finitas y elementos finitos.

Consideramos una malla espacial uniforme de paso  $h = 1/(n-1)$  con  $n$  el tamaño de la imagen como en el caso anterior. Para cada punto de esta malla construiremos un volumen de control de manera que el dominio sea la unión disjunta de todos los volúmenes de control. Construída la malla, se integrará sobre cada uno de estos volúmenes de control la ecuación diferencial a resolver.

De la misma forma que en el problema anterior, comenzamos con una discretización temporal del problema utilizando un esquema semi-implícito. De esta forma aproximando

$$\frac{\partial \mathbf{u}(\mathbf{x}, t_{k+1})}{\partial t} \approx \frac{\mathbf{u}(\mathbf{x}, t_{k+1}) - \mathbf{u}(\mathbf{x}, t_k)}{\tau},$$

se obtiene

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t_{k+1}) &= \mathbf{u}(\mathbf{x}, t_k) + \tau \alpha \nabla \sigma(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_{k+1})) + \tau f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)) \\ &\Downarrow \\ (I - \tau \alpha \nabla \sigma) \mathbf{u}(\mathbf{x}, t_{k+1}) &= \mathbf{u}(\mathbf{x}, t_k) + \tau f(\mathbf{x}, \mathbf{u}(\mathbf{x}, t_k)) \end{aligned}$$

Para discretizar  $\nabla \cdot \sigma$  tendremos en cuenta que

$$\nabla \cdot \sigma \Rightarrow \begin{pmatrix} \frac{\partial \sigma_{11}}{\partial x} + \frac{\partial \sigma_{12}}{\partial y} \\ \frac{\partial \sigma_{21}}{\partial x} + \frac{\partial \sigma_{22}}{\partial y} \end{pmatrix}.$$

Procederemos de la misma forma que para el método de difusión, discretizando de forma separada los nodos interiores, los lados y las esquinas, ya que para cada uno de estos subconjuntos del dominio varían las ecuaciones a aproximar.

Comenzamos con los nodos interiores, con  $i, j = 2, \dots, n-1$ . El esquema sobre el que vamos a trabajar es el que se observa en la Figura 1, donde el recinto en color es el volúmen de control sobre el cual vamos a integrar.

El esquema general de integración del método de discretización a través de volúmenes finitos consiste en

$$\int_{\Omega} \nabla \cdot \sigma = \int_{\partial\Omega} \sigma \cdot \vec{n} = \int_{\Gamma_1} \sigma \cdot \vec{n} + \int_{\Gamma_2} \sigma \cdot \vec{n} + \int_{\Gamma_3} \sigma \cdot \vec{n} + \int_{\Gamma_4} \sigma \cdot \vec{n}$$

Para la componente  $u$  del desplazamiento tenemos que aproximar el término

$$\frac{\partial \sigma_{11}}{\partial x} + \frac{\partial \sigma_{12}}{\partial y}.$$

Siguiendo el esquema de la Figura 1 las integrales anteriores se reducen a

$$\int_{\Gamma_1} \sigma_{11} \int_{\Gamma_2} \sigma_{12} - \int_{\Gamma_3} \sigma_{11} - \int_{\Gamma_4} \sigma_{12}.$$

Teniendo en cuenta que

$$\begin{aligned} \sigma_{11} &= (\lambda + 2\mu) u_x + \lambda v_y, \\ \sigma_{12} &= \mu (u_y + v_x), \end{aligned}$$

Se obtiene que

$$\begin{aligned} &h \left[ (\lambda + 2\mu) u_x + \lambda v_y \right]_{i+1/2,j} + h \left[ \mu (u_y + v_x) \right]_{i,j+1/2} \\ &- h \left[ (\lambda + 2\mu) u_x + \lambda v_y \right]_{i-1/2,j} - h \left[ \mu (u_y + v_x) \right]_{i,j-1/2} \end{aligned}$$

Aproximando cada función mediante diferencias finitas centrales en

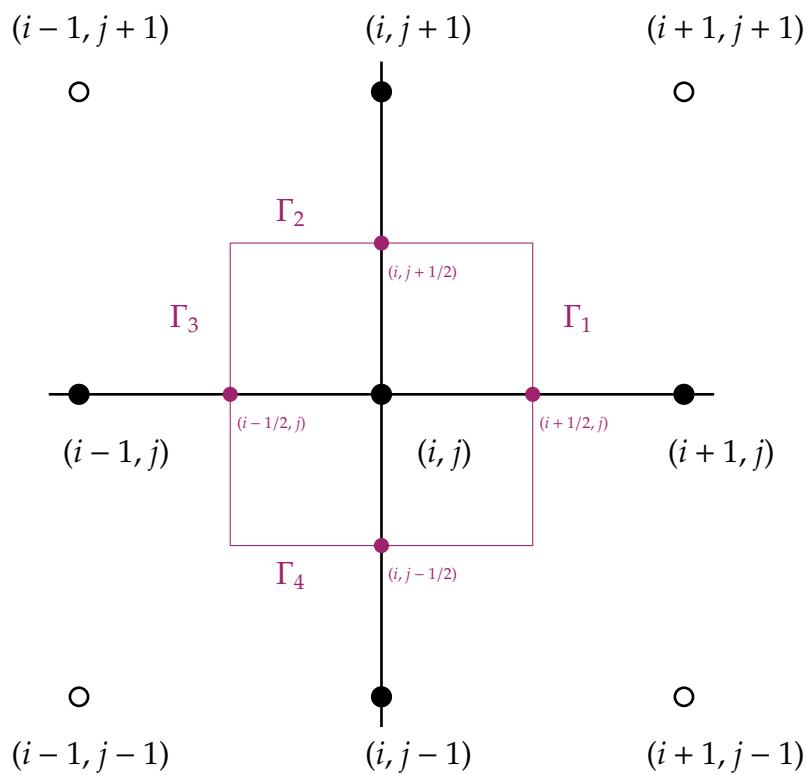


Figura 1: Esquema correspondiente a los nodos interiores para la discretización por volúmenes finitos.

sus nodos correspondientes obtenemos

$$\begin{aligned}
 & +h \left[ (\lambda + 2\mu) \frac{u_{i+1,j} - u_{i,j}}{h} + \frac{\lambda}{2} \left( \frac{v_{i+1,j+1} - v_{i+1,j-1}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h} \right) \right] + \\
 & +h \left[ \mu \left( \frac{u_{i,j+1} - u_{i,j-1}}{h} + \frac{1}{2} \left( \frac{v_{i+1,j+1} - v_{i-1,j+1}}{2h} + \frac{v_{i+1,j} - v_{i-1,j}}{2h} \right) \right) \right] + \\
 & -h \left[ (\lambda + 2\mu) \frac{u_{i,j} - u_{i-1,j}}{h} + \frac{\lambda}{2} \left( \frac{v_{i,j+1} - v_{i,j-1}}{2h} + \frac{v_{i-1,j+1} - v_{i-1,j-1}}{2h} \right) \right] + \\
 & -h \left[ \mu \left( \frac{u_{i,j} - u_{i,j-1}}{h} + \frac{1}{2} \left( \frac{v_{i+1,j} - v_{i-1,j}}{2h} + \frac{v_{i+1,j-1} - v_{i-1,j-1}}{2h} \right) \right) \right]
 \end{aligned}$$

Se obtiene que los coeficientes que acompañan a cada nodo son:

$$\begin{array}{ll}
 u_{i+1,j} : \lambda + 2\mu & v_{i+1,j+1} : \frac{\lambda + \mu}{4} \\
 u_{i,j} : -2(\lambda + 3\mu) & v_{i-1,j+1} : -\frac{\lambda + \mu}{4} \\
 u_{i-1,j} : \lambda + 2\mu & v_{i+1,j-1} : -\frac{\lambda + \mu}{4} \\
 u_{i,j+1} : \mu & v_{i-1,j-1} : \frac{\lambda + \mu}{4} \\
 u_{i,j-1} : \mu
 \end{array}$$

De forma análoga se obtiene una discretización para la componente  $v$  del desplazamiento, teniendo en cuenta el término

$$\frac{\partial \sigma_{21}}{\partial x} + \frac{\partial \sigma_{22}}{\partial y},$$

con  $\sigma_{22} = (\lambda + 2\mu)v_y + \lambda u_x$  y  $\sigma_{21} = \sigma_{12}$ .

Ahora la integración sobre  $\Gamma_i$ ,  $i = 1, \dots, 4$  queda

$$\int_{\Gamma_1} \sigma_{12} + \int_{\Gamma_2} \sigma_{22} - \int_{\Gamma_3} \sigma_{12} - \int_{\Gamma_4} \sigma_{22}$$

Sustituyendo

$$\begin{aligned}
 & +h \left[ \mu(u_y + v_x) \right]_{i+1/2,j} + h \left[ (\lambda + 2\mu)v_y + \lambda u_x \right]_{i,j+1/2} + \\
 & -h \left[ \mu(u_y + v_x) \right]_{i-1/2,j} - h \left[ (\lambda + 2\mu)v_y + \lambda u_x \right]_{i,j-1/2}
 \end{aligned}$$

Procediendo de forma análoga

$$\begin{aligned}
 v_{i,j+1} : \lambda + 2\mu & \quad u_{i+1,j+1} : \frac{\lambda + \mu}{4} \\
 v_{i,j} : -2(\lambda + 3\mu) & \quad u_{i-1,j+1} : -\frac{\lambda + \mu}{4} \\
 v_{i,j-1} : \lambda + 2\mu & \quad u_{i+1,j-1} : -\frac{\lambda + \mu}{4} \\
 v_{i+1,j} : \mu & \quad u_{i-1,j-1} : \frac{\lambda + \mu}{4} \\
 v_{i-1,j} : \mu
 \end{aligned}$$

En cuanto a los nodos de la frontera del dominio, realizamos el proceso para el lateral derecho con  $i = n$  y  $j = 2, \dots, n - 1$ . El resto se calculan de forma análoga. El esquema a seguir ahora se encuentra en la Figura 2.

La condición de contorno es  $\sigma \cdot \vec{n} = 0$ . En este lateral se tiene  $\vec{n} = (1, 0)$  lo que implica  $\sigma_{11} = \sigma_{12} = 0$ .

Para calcular los coeficientes que calculan el desplazamiento  $u$ , se tiene que se anulan las integrales sobre todos los caminos excepto para  $\Gamma_3$ . Nos queda entonces

$$-\int_{\Gamma_3} \sigma_{11}$$

Sustituyendo y aproximando cada función en sus nodos correspondientes

$$\begin{aligned}
 -h[(\lambda + 2\mu)u_x + \lambda v_y]_{n-1/2,j} + h^2 f_{i,j} &= 0 \\
 -h\left[(\lambda + 2\mu)\frac{u_{n,j} - u_{n-1,j}}{h} + \frac{\lambda}{2}\left(\frac{v_{n,j+1} - v_{n,j-1}}{2h} + \frac{v_{n-1,j+1} - v_{n-1,j-1}}{2h}\right)\right]
 \end{aligned}$$

Así los coeficientes que acompañan a cada nodo son

$$\begin{aligned}
 u_{n,j} : -(\lambda + 2\mu) & \quad v_{n,j+1} : -\frac{\lambda}{4} \\
 u_{n-1,j} : \lambda + 2\mu & \quad v_{n,j-1} : \frac{\lambda}{4} \\
 v_{n-1,j+1} : -\frac{\lambda}{4} & \\
 v_{n-1,j-1} : \frac{\lambda}{4}
 \end{aligned}$$

Para calcular el desplazamiento  $v$  se anula la integral sobre  $\Gamma_1$

$$+\int_{\Gamma_2} \sigma_{22} - \int_{\Gamma_3} \sigma_{12} - \int_{\Gamma_4} \sigma_{22}$$

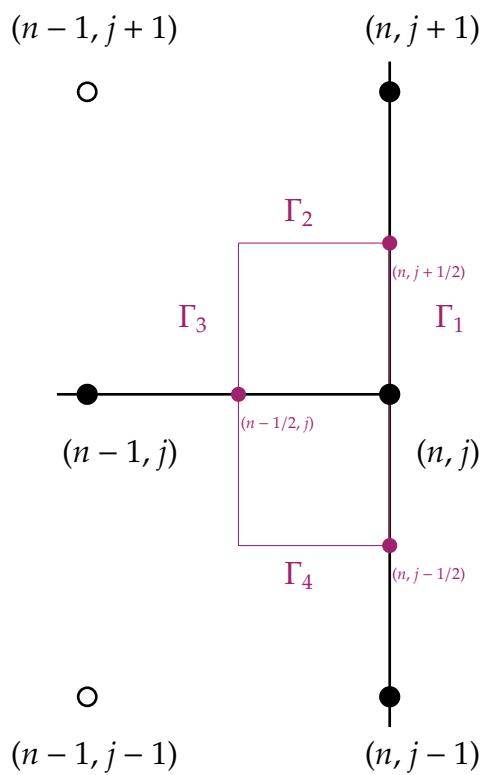


Figura 2: Esquema correspondiente a los nodos del lateral derecho para la discretización por volúmenes finitos.

Sustituyendo y aproximando cada función en sus nodos correspondientes nos queda que los coeficientes que acompañan a cada nodo son

$$\begin{array}{ll} v_{n,j+1} : \frac{\lambda + 2\mu}{2} & u_{n,j+1} : \frac{\lambda - \mu}{4} \\ v_{n,j} : -(\lambda + 3\mu) & u_{n,j-1} : -\frac{\lambda + \mu}{4} \\ v_{n,j-1} : \frac{\lambda + 2\mu}{2} & u_{n-1,j+1} : -\frac{\lambda - \mu}{4} \\ v_{n-1,j} : \mu & u_{n-1,j-1} : \frac{\lambda + \mu}{4} \end{array}$$

Los coeficientes para el resto de laterales se calculan de forma análoga. La declaración de los caminos  $\Gamma_i$  con  $i = 1, \dots, 4$  seguirá el mismo orden para todos los lados de los recintos rectangulares que tomemos.

Para el lateral superior con  $i = 2, \dots, n-1$  y  $j = n$ , se tiene  $\vec{n} = (0, 1)$  y por tanto  $\sigma_{12} = \sigma_{22} = 0$ . Para calcular el desplazamiento  $u$  se anula la integral sobre  $\Gamma_2$  y quedan los coeficientes

$$\begin{array}{ll} u_{i+1,n} : \frac{\lambda + 2\mu}{2} & v_{i+1,n} : \frac{\lambda + \mu}{4} \\ u_{i,n} : -(\lambda + 3\mu) & v_{i-1,n} : \frac{\lambda - \mu}{4} \\ u_{i-1,n} : \frac{\lambda + 2\mu}{2} & v_{i+1,n-1} : \frac{-\lambda + \mu}{4} \\ u_{i,n-1} : \mu & v_{i-1,n-1} : -\frac{\lambda + \mu}{4} \end{array}$$

Para el desplazamiento  $v$  se anulan todas las integrales excepto la que se realiza sobre el camino  $\Gamma_4$ . Quedan los coeficientes

$$\begin{array}{ll} v_{i,n} : -(\lambda + 2\mu) & u_{i+1,n} : -\frac{\lambda}{4} \\ v_{i,n-1} : \lambda + 2\mu & u_{i-1,n} : \frac{\lambda}{4} \\ & u_{i+1,n-1} : -\frac{\lambda}{4} \\ & u_{i-1,n-1} : \frac{\lambda}{4} \end{array}$$

Para el **lateral inferior**, donde  $i = 1, \dots, n-1$  y  $j = 1$  se tiene que  $\vec{n} = (0, -1)$  lo cual implica que  $\sigma_{12} = \sigma_{22} = 0$ . Al calcular los coeficientes

para el desplazamiento  $u$  se anula la integral sobre  $\Gamma_4$ . Los coeficientes que acompañan a cada nodo son

$$\begin{array}{ll} u_{i+1,1} : \frac{\lambda + 2\mu}{2} & v_{i+1,2} : \frac{\lambda + \mu}{4} \\ u_{i,1} : -(\lambda + 3\mu) & v_{i+1,1} : \frac{\lambda - \mu}{4} \\ u_{i-1,1} : \frac{\lambda + 2\mu}{2} & v_{i-1,2} : \frac{-\lambda + \mu}{4} \\ u_{i,2} : \mu & v_{i-1,1} : -\frac{\lambda + \mu}{4} \end{array}$$

Para el desplazamiento  $v$  se anulan las integrales sobre todos los caminos excepto la aplicada sobre  $\Gamma_2$  y sus coeficientes quedan

$$\begin{array}{ll} v_{i,1} : -(\lambda + 2\mu) & u_{i-1,2} : -\frac{\lambda}{4} \\ v_{i,2} : \lambda + 2\mu & u_{i+1,2} : \frac{\lambda}{4} \\ u_{i-1,1} : -\frac{\lambda}{4} & \\ u_{i+1,1} : \frac{\lambda}{4} & \end{array}$$

Por último, en el lateral izquierdo con  $i = 1$  y  $j = 2, \dots, n - 1$  se tiene  $\vec{n} = (-1, 0)$  por lo que  $\sigma_{11} = \sigma_{12} = 0$ . Entonces para calcular los coeficientes del desplazamiento  $u$  se anulan todas las integrales excepto la que opera sobre  $\Gamma_1$ , y se tiene

$$\begin{array}{ll} u_{1,j} : -(\lambda + 2\mu) & v_{2,j-1} : -\frac{\lambda}{4} \\ u_{2,j} : \lambda + 2\mu & v_{2,j+1} : \frac{\lambda}{4} \\ v_{1,j-1} : -\frac{\lambda}{4} & \\ v_{1,j+1} : \frac{\lambda}{4} & \end{array}$$

Y para los del desplazamiento  $v$  se anula la integral sobre  $\Gamma_3$  quedando

$$\begin{array}{ll} v_{1,j+1} : \frac{\lambda + 2\mu}{2} & u_{2,j-1} : \frac{-\lambda + \mu}{4} \\ v_{1,j} : -(\lambda + 3\mu) & u_{2,j+1} : \frac{\lambda + \mu}{4} \\ v_{1,j-1} : \frac{\lambda + 2\mu}{2} & u_{1,j-1} : \frac{\lambda - \mu}{4} \\ v_{2,j} : \mu & u_{1,j+1} : -\frac{\lambda + \mu}{4} \end{array}$$

Para finalizar la discretización nos quedan las cuatro esquinas. Sobre cada una de ellas actúan dos vectores  $\vec{n}$ .

Veamos en la Figura 3 el esquema de la esquina superior derecha, donde  $i = j = n$ .

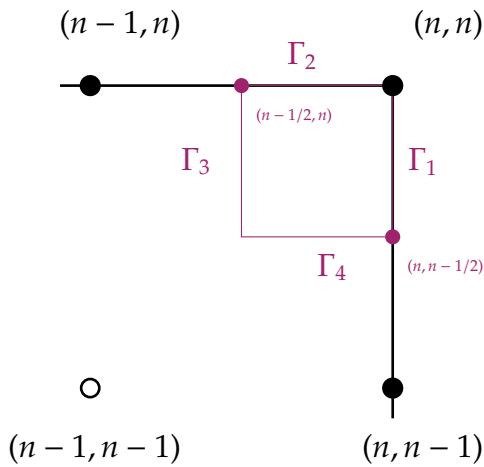


Figura 3: Esquema correspondiente a la esquina superior derecha para la discretización por volúmenes finitos.

Sobre  $\Gamma_1$  se tiene que  $\vec{n}_1 = (1, 0)$  y por tanto  $\sigma_{11} = \sigma_{12} = 0$ , y sobre  $\Gamma_2$  se tiene  $\vec{n}_2 = (0, 1)$  y entonces  $\sigma_{12} = \sigma_{22} = 0$ .

Realizando los cálculos de la misma forma que anteriormente se tiene que los coeficientes para el desplazamiento  $u$  son

$$\begin{aligned} u_{n,n} &: -\frac{\lambda + 2\mu}{2} & v_{n,n} &: -\frac{\lambda}{4} \\ u_{n-1,n} &: \frac{\lambda + 2\mu}{2} & v_{n,n-1} &: \frac{\lambda}{4} \\ && v_{n-1,n} &: -\frac{\lambda}{4} \\ && v_{n-1,n-1} &: \frac{\lambda}{4} \end{aligned}$$

Y para el desplazamiento  $v$

$$\begin{aligned} v_{n,n} &: -\frac{\lambda + 2\mu}{2} & u_{n,n} &: -\frac{\lambda}{4} \\ v_{n,n-1} &: \frac{\lambda + 2\mu}{2} & u_{n,n-1} &: -\frac{\lambda}{4} \\ && u_{n-1,n} &: \frac{\lambda}{4} \\ && u_{n-1,n-1} &: \frac{\lambda}{4} \end{aligned}$$

Análogamente se obtienen los coeficientes para calcular  $u$  y  $v$  en el resto de esquinas.

En la **esquina superior izquierda**, donde  $i = 1$  y  $j = n$  se tiene  $\vec{n}_1 = (-1, 0)$  y por tanto  $\sigma_{11} = \sigma_{12} = 0$  sobre  $\Gamma_3$ , y  $\vec{n}_2 = (0, 1)$  y entonces  $\sigma_{12} = \sigma_{22} = 0$  sobre  $\Gamma_2$ .

Los coeficientes para el desplazamiento  $u$  son

$$\begin{aligned} u_{2,n} &: -\frac{\lambda + 2\mu}{2} & v_{2,n} &: \frac{\lambda}{4} \\ u_{1,n} &: \frac{\lambda + 2\mu}{2} & v_{2,n-1} &: -\frac{\lambda}{4} \\ && v_{1,n-1} &: -\frac{\lambda}{4} \\ && v_{1,n} &: \frac{\lambda}{4} \end{aligned}$$

Y para el desplazamiento  $v$

$$\begin{array}{ll} v_{1,n} : -\frac{\lambda + 2\mu}{2} & u_{2,n} : -\frac{\lambda}{4} \\ v_{1,n-1} : \frac{\lambda + 2\mu}{2} & u_{1,n} : -\frac{\lambda}{4} \\ & u_{2,n-1} : \frac{\lambda}{4} \\ & u_{1,n-1} : \frac{\lambda}{4} \end{array}$$

En la esquina inferior derecha, donde  $i = n$  y  $j = 1$  se tiene  $\vec{n}_1 = (1, 0)$  y por tanto  $\sigma_{11} = \sigma_{12} = 0$  sobre  $\Gamma_1$ ,  $\vec{n}_2 = (0, 1)$  y entonces  $\sigma_{12} = \sigma_{22} = 0$  sobre  $\Gamma_4$ .

Los coeficientes para el desplazamiento  $u$  son

$$\begin{array}{ll} u_{n-1,1} : -\frac{\lambda + 2\mu}{2} & v_{n,1} : \frac{\lambda}{4} \\ u_{n,1} : \frac{\lambda + 2\mu}{2} & v_{n,2} : -\frac{\lambda}{4} \\ & v_{n-1,2} : -\frac{\lambda}{4} \\ & v_{n-1,1} : \frac{\lambda}{4} \end{array}$$

Y para el desplazamiento  $v$

$$\begin{array}{ll} v_{n,1} : -\frac{\lambda + 2\mu}{2} & u_{n-1,2} : -\frac{\lambda}{4} \\ v_{n,2} : \frac{\lambda + 2\mu}{2} & u_{n-1,1} : -\frac{\lambda}{4} \\ & u_{n,2} : \frac{\lambda}{4} \\ & u_{n,1} : \frac{\lambda}{4} \end{array}$$

Por último la **esquina inferior izquierda**, donde  $i = 1$  y  $j = 1$  se tiene  $\vec{n}_1 = (-1, 0)$  y por tanto  $\sigma_{11} = \sigma_{12} = 0$  sobre  $\Gamma_3$ ,  $\vec{n}_2 = (0, -1)$  y entonces  $\sigma_{12} = \sigma_{22} = 0$  sobre  $\Gamma_4$ .

Los coeficientes para el desplazamiento  $u$  son

$$\begin{aligned} u_{2,1} &: \frac{\lambda + 2\mu}{2} & v_{2,2} &: \frac{\lambda}{4} \\ u_{1,1} &: \frac{\lambda + 2\mu}{2} & v_{2,1} &: -\frac{\lambda}{4} \\ && v_{1,1} &: -\frac{\lambda}{4} \\ && v_{1,2} &: \frac{\lambda}{4} \end{aligned}$$

Y para el desplazamiento  $v$

$$\begin{aligned} v_{1,1} &: -\frac{\lambda + 2\mu}{2} & u_{1,2} &: -\frac{\lambda}{4} \\ v_{1,2} &: \frac{\lambda + 2\mu}{2} & u_{1,1} &: -\frac{\lambda}{4} \\ && u_{2,2} &: \frac{\lambda}{4} \\ && u_{2,1} &: \frac{\lambda}{4} \end{aligned}$$

De la misma forma que en la técnica anterior, en cada paso de tiempo, resulta un sistema de ecuaciones a resolver

$$A U^{k+1} = \mathbf{b}$$

donde

$$A = I - \tau \alpha L_h$$

con  $L_h$  corresponde a la discretización del operador elasticidad.

El término  $\mathbf{b}$  se calcula de la misma forma que los términos  $b_1$  y  $b_2$  para el problema de Difusión.

En el Anexo A.4 se observa la construcción de la matriz  $A$ .

Para finalizar esta sección, se presentan los algoritmos correspondientes para los dos métodos de regularización:

---

**Algorithm 1** Esquema Regularizador de Difusión.

---

- 1: Definir  $\mathbf{u}^0$ ;
  - 2: Determinar  $\alpha, \tau$ ;
  - 3: **for**  $k = 0, \dots$  **do**
  - 4:     Calcular  $\mathbf{f}^k = \nabla_h T(\mathbf{u}^k) (T(\mathbf{u}^k) - R)$ ;
  - 5:     Calcular  $\mathbf{b}^k = \tau \mathbf{f}^k + \mathbf{u}^k$ ;
  - 6:     Resolver  $(I - \tau \alpha \Delta_h) \mathbf{u}^{k+1} = \mathbf{b}^k$ ;
- 

---

**Algorithm 2** Esquema Regularizador de Elasticidad.

---

- 1: Definir  $\mathbf{u}^0$ ;
  - 2: Determinar  $\alpha, \tau, \lambda, \mu$ ;
  - 3: **for**  $k = 0, \dots$  **do**
  - 4:     Calcular  $\mathbf{f}^k = \nabla_h T(\mathbf{u}^k) (T(\mathbf{u}^k) - R)$ ;
  - 5:     Calcular  $\mathbf{b}^k = \tau \mathbf{f}^k + \mathbf{u}^k$ ;
  - 6:     Resolver  $(I - \tau \alpha \operatorname{Div}(\sigma)) \mathbf{u}^{k+1} = \mathbf{b}^k$ ;
-

## 5. Resultados numéricos

Para poner en práctica todo lo desarrollado en las secciones anteriores vamos a aplicar los algoritmos anteriormente descritos sobre dos imágenes que están desplazadas.

Utilizaremos la ya conocida imagen de Lena.



Figura 4: Imagen de Lena.

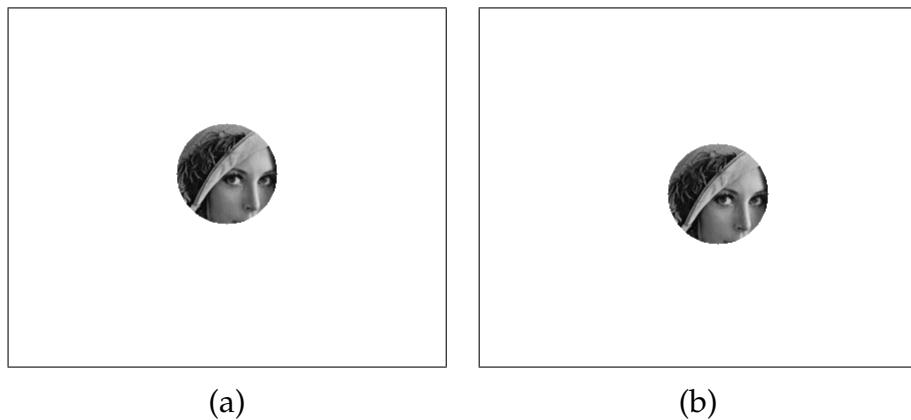


Figura 5: (a) Imagen patrón de Lena, (b) Imagen referencia de Lena.

Las imágenes que resultan de realizar a la Figura 4 una sección circular y posteriormente realizar una traslación lineal vienen dadas en la Figura 5:

Aplicamos sobre ellas el algoritmo de difusión con parámetros  $\tau = 1$  y  $\alpha = 1/8$  y el algoritmo elástico con los mismos parámetros  $\tau$  y  $\alpha$  y con

$\lambda = 100$  y  $\mu = -100$ .

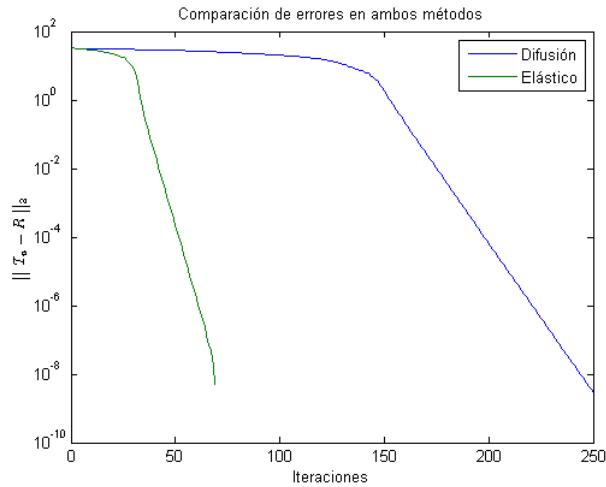


Figura 6: Comparación del error que cometen los métodos de Difusión y Elástico para la resolución del problema de minimización.

Con ambos métodos, tras 250 iteraciones, llegamos a la solución. En la Figura 6 se ilustran los errores cometidos por ambos métodos.

Como se puede observar, con el regularizador elástico se alcanza la convergencia con un número de iteraciones mucho menor.

En la descripción de ambos problemas ya comentamos que el método de difusión es un caso particular del de elasticidad, y que se consigue tomando  $\lambda = 1$  y  $\mu = -1$ . Por ello podíamos deducir, antes de realizar la prueba que este último método sería más acertado que el de difusión por la posibilidad de elegir unos parámetros que consigan la convergencia mucho más rápido.

## A. Anexos

### A.1. Matriz Laplaciano

Programa de la matriz del Laplaciano.

```

function A = laplaciano (n,h)

A = sparse(n*n, n*n);

% Esquina inferior izquierda
j = 1;
i = 1;
ind = 1;
A(ind,ind+1)    = -2/(h*h);
A(ind,ind+n)    = -2/(h*h);
A(ind,ind)       =  4/(h*h);

% Recta inferior
for i=2:n-1
    ind = ind+1;
    A(ind,ind+1)    = -1/(h*h);
    A(ind,ind-1)    = -1/(h*h);
    A(ind,ind+n)    = -2/(h*h);
    A(ind,ind)       =  4/(h*h);
end

% Esquina inferior derecha
i=n-1;
ind=ind+1;
A(ind,ind-1)    = -2/(h*h);
A(ind,ind+n)    = -2/(h*h);
A(ind,ind)       =  4/(h*h);

% Recta izquierda
for j=2:n-1
    i=1;
    ind = ind+1;
    A(ind,ind+1)    = -2/(h*h);
    A(ind,ind+n)    = -1/(h*h);
    A(ind,ind-n)    = -1/(h*h);
    A(ind,ind)       =  4/(h*h);

    % Nodos interiores
    for i=2:n-1
        ind = ind+1;
        A(ind,ind+1)    = -1/(h*h);
        A(ind,ind-1)    = -1/(h*h);
    end
end

```

```

A(ind,ind+n)      = -1/(h*h);
A(ind,ind-n)      = -1/(h*h);
A(ind,ind)         =  4/(h*h);
end

% Recta derecha
i=n;
ind=ind+1;
A(ind,ind-1)      = -2/(h*h);
A(ind,ind+n)      = -1/(h*h);
A(ind,ind-n)      = -1/(h*h);
A(ind,ind)         =  4/(h*h);
end

% Esquina superior izquierda
j=n;
i = 1;
ind = ind+1;
A(ind,ind+1)      = -2/(h*h);
A(ind,ind-n)      = -2/(h*h);
A(ind,ind)         =  4/(h*h);

% Recta superior
for i=2:n-1
    ind = ind+1;
    A(ind,ind+1)      = -1/(h*h);
    A(ind,ind-1)      = -1/(h*h);
    A(ind,ind-n)      = -2/(h*h);
    A(ind,ind)         =  4/(h*h);
end

% Esquina superior derecha
i=n;
ind=ind+1;
A(ind,ind-1)      = -2/(h*h);
A(ind,ind-n)      = -2/(h*h);
A(ind,ind)         =  4/(h*h);

end

```

## A.2. Interpolación

Interpolación bilineal para calcular  $T_u$ .

```
function sol = interpola2(n, TT, x, y, i, j)
```

```

k = i + n/2 - 1;
l = j + n/2 - 1;

omx=(k-x)-floor(k-x);
omy=(l-y)-floor(l-y);
A=TT(floor(k-x), floor(l-y));
B=TT(ceil(k-x), floor(l-y));
C=TT(floor(k-x), ceil(l-y));
D=TT(ceil(k-x), ceil(l-y));
sol = (1-omx)*(1-omy)*A + (1-omy)*omx*B + (1-omx)*omy*C + omx*omy*D;
end

```

### A.3. Programa Regularizador Difusión

Programa del Regularizador de Difusión.

```

function [Edifu] = Difusion

T = double(imread('lena.jpg'))/255;
T = T(:,:,1);

[n,n] = size(T);

h = 1/(n-1);

% Definimos la imagen T como un sector de esa imagen y el resto blanco

cx = 256/2;
cy = 256/2;

for i = 1:n
    for j = 1:n
        if (((i-cx)^2 + (j-cy)^2) > 50^2)
            T(i,j) = 1;
        end
    end
end

% Definimos la imagen R como la imagen T trasladada 10 pixeles hacia abajo

pix = 20;

R = ones(n,n);

for i = pix+1:n
    for j = pix+1:n

```

```

R(i,j) = T(i-pix,j-pix);
end
end

figure(1)
imshow(T); % title La imagen T

figure(2)
imshow(R); % title La imagen R

pause

% Inicializamos las matrices ux, uy

ux = zeros(n,n);

uy = zeros(n,n);

D = zeros(n,n);
for i = 1:n
    for j = 1:n
        D(i,j) = T(i - ux(i,j), j - uy(i,j)) - R(i,j);
    end
end

Tx = zeros(n,n);
Ty = zeros(n,n);

% Definimos los laplacianos llamando a la función
Lx = laplaciano(n,h);
Ly = Lx;

% Comenzamos el método.

iter = 250;

tau = 1;
alpha = 1/8;

% Definimos las matrices Ax, Ay del sistema que luego resolveremos

Ax = speye(n*n) + alpha * tau * Lx;
Ay = speye(n*n) + alpha * tau * Ly;

nuevoux = sparse(n*n,1);
nuevouy = sparse(n*n,1);

aux = 2*h;

```

```
%%% PONEMOS LA IMAGEN T EN UNA MÁS GRANDE CON 0 EN EL RESTO
% Para luego interpolar T con i-ux, j-uy
```

```
TT = ones(2*n, 2*n);

for k = n/2:(3*n/2 - 1)
    for l = n/2:(3*n/2 - 1)
        TT(k,l) = T(k - n/2 + 1, l - n/2 + 1);
    end
end

Edifu = [];

for k = 1:iter

    % Definimos la diferencia
    for i = 1:n
        for j = 1:n
            D(i,j) = interpola2(n, TT, ux(i,j), uy(i,j), i, j) - R(i,j);
        end
    end

    Edifu = [Edifu norm(D)];

    % Definimos el gradiente de T, Tx y Ty
    for i = 2:n-1
        for j = 1:n
            Tx(i,j) = (interpola2(n, TT, ux(i+1,j), uy(i+1,j), i+1, j) -
                        interpola2(n, TT, ux(i-1,j), uy(i-1,j), i-1, j))/aux;
        end
    end

    for i = 1:n
        for j = 2:n-1
            Ty(i,j) = (interpola2(n, TT, ux(i,j+1), uy(i,j+1), i, j+1) -
                        interpola2(n, TT, ux(i,j-1), uy(i,j-1), i, j-1))/aux;
        end
    end

    % Hay que resolver dos sistemas
    % Ax ux = bx // Ay uy = by

    bx = ux + tau * D .* Tx;
    by = uy + tau * D .* Ty;

    % Tenemos que pasar bx, by a vector (n*n,1)
```

```

bx = reshape(bx', n*n, 1);
by = reshape(by', n*n, 1);

% Ya podemos resolver el sistema:

nuevoux = Ax\bx;
nuevouy = Ay\by;

% Rellamamos ux, uy para la siguiente iteración

ux = reshape(nuevoux,n,n)';
uy = reshape(nuevouy,n,n)';

end

sol = zeros(n,n);
for i=1:n
    for j=1:n
        sol(i,j)= interpola2(n, TT, ux(i,j), uy(i,j), i, j);
    end
end

figure(3);
imshow(sol);

end

```

## A.4. Matriz Elasticidad

Matriz del problema de Elasticidad.

```

function A = matriz_Elasticidad(n,h, lambda, mu)

% DEFINICIÓN DE LOS BLOQUES:
A11 = sparse(n*n, n*n);
A12 = sparse(n*n, n*n);
A21 = sparse(n*n, n*n);
A22 = sparse(n*n, n*n);

% DEFINICIÓN DE CONSTANTES:
aux = h*h;
% (lambda + 2*mu) / (h*h)
l2m = (lambda + 2*mu) / aux;
% 2*(lambda + 3*mu) / (h*h)
l3m = 2*(lambda + 3*mu) / aux;
% mu / (h*h);
muh = mu / aux;

```

```
% (Lambda + mu) / (4*h*h)
lm = (lambda + mu) / (4*aux);
% lambda/(4*h*h)
l4h = lambda/ (4*aux);
% lambda/(2*h*h)
l2h = lambda/ (2*aux);
% (- lambda + mu) / (4*h*h)
menoslm = (-lambda + mu) / (4*aux);

% Esquina inferior izquierda
j = 1;
i = 1;
ind = 1;
A11(ind,ind+1) = 2/(aux);
A11(ind,ind+n) = 2/(aux);
A11(ind,ind) = -4/(aux);
A22(ind,ind+1) = 2/(aux);
A22(ind,ind+n) = 2/(aux);
A22(ind,ind) = -4/(aux);

% Recta inferior
for i=2:n-1
    ind = ind+1;
    A11(ind,ind+1) = 1/(aux);
    A11(ind,ind-1) = 1/(aux);
    A11(ind,ind+n) = 2/(aux);
    A11(ind,ind) = -4/(aux);
    A22(ind,ind+1) = 1/(aux);
    A22(ind,ind-1) = 1/(aux);
    A22(ind,ind+n) = 2/(aux);
    A22(ind,ind) = -4/(aux);
end

% Esquina inferior derecha
i=n-1;
ind=ind+1;
A11(ind,ind-1) = 2/(aux);
A11(ind,ind+n) = 2/(aux);
A11(ind,ind) = -4/(aux);
A22(ind,ind-1) = 2/(aux);
A22(ind,ind+n) = 2/(aux);
A22(ind,ind) = -4/(aux);

% Recta izquierda
for j=2:n-1
    i=1;
    ind = ind+1;
    A11(ind,ind+1) = 2/(aux);
    A11(ind,ind+n) = 1/(aux);
```

```

A11(ind,ind-n) = 1/(aux);
A11(ind,ind) = -4/(aux);
A22(ind,ind+1) = 2/(aux);
A22(ind,ind+n) = 1/(aux);
A22(ind,ind-n) = 1/(aux);
A22(ind,ind) = -4/(aux);

% Nodos interiores
for i=2:n-1
    ind = ind+1;
    % A11:
    A11(ind,ind+1) = 12m;
    A11(ind,ind-1) = 12m;
    A11(ind,ind+n) = muh;
    A11(ind,ind-n) = muh;
    A11(ind,ind) = -13m;
    % A12:
    A12(ind, ind+n-1) = -lm;
    A12(ind, ind-n-1) = lm;
    A12(ind, ind+n+1) = lm;
    A12(ind, ind-n+1) = -lm;
    % A21:
    A21(ind, ind+n-1) = -lm;
    A21(ind, ind-n-1) = lm;
    A21(ind, ind+n+1) = lm;
    A21(ind, ind-n+1) = -lm;
    % A22:
    A22(ind,ind+1) = muh;
    A22(ind,ind-1) = muh;
    A22(ind,ind+n) = 12m;
    A22(ind,ind-n) = 12m;
    A22(ind,ind) = -13m;
end

% Recta derecha
i=n;
ind=ind+1;
A11(ind,ind-1) = 2/(aux);
A11(ind,ind+n) = 1/(aux);
A11(ind,ind-n) = 1/(aux);
A11(ind,ind) = -4/(aux);
A22(ind,ind-1) = 2/(aux);
A22(ind,ind+n) = 1/(aux);
A22(ind,ind-n) = 1/(aux);
A22(ind,ind) = -4/(aux);
end

% Esquina superior izquierda
j=n;

```

```

i = 1;
ind = ind+1;
A11(ind,ind+1) = 2/(aux);
A11(ind,ind-n) = 2/(aux);
A11(ind,ind) = -4/(aux);
A22(ind,ind+1) = 2/(aux);
A22(ind,ind-n) = 2/(aux);
A22(ind,ind) = -4/(aux);

% Recta superior
for i=2:n-1
    ind = ind+1;
    A11(ind,ind+1) = 1/(aux);
    A11(ind,ind-1) = 1/(aux);
    A11(ind,ind-n) = 2/(aux);
    A11(ind,ind) = -4/(aux);
    A22(ind,ind+1) = 1/(aux);
    A22(ind,ind-1) = 1/(aux);
    A22(ind,ind-n) = 2/(aux);
    A22(ind,ind) = -4/(aux);
end

% Esquina superior derecha
i=n;
ind=ind+1;
A11(ind,ind-1) = 2/(aux);
A11(ind,ind-n) = 2/(aux);
A11(ind,ind) = -4/(aux);
A22(ind,ind-1) = 2/(aux);
A22(ind,ind-n) = 2/(aux);
A22(ind,ind) = -4/(aux);

A = sparse(2*n*n, 2*n*n);
A = [A11 A12; A12 A11];
end

```

## A.5. Programa Regularizador Elástico

Programa del Regularizador Elástico.

```

function [Eelas] = Elastico

T = double(imread('lena.jpg'))/255;
T = T(:,:,1);

[n,n] = size(T);

```

```
h = 1/(n-1);

% Definimos la imagen T como un sector de esa imagen y el resto blanco

cx = 256/2;
cy = 256/2;

for i = 1:n
    for j = 1:n
        if (((i-cx)^2 + (j-cy)^2) > 50^2)
            T(i,j) = 1;
        end
    end
end

% Definimos la imagen R como la imagen T trasladada 20 pixeles hacia abajo

pix = 20;

R = ones(n,n);

for i = pix+1:n
    for j = pix+1:n
        R(i,j) = T(i-pix, j-pix);
    end
end

for i = 1:pix
    for j = 1:pix
        R(i,j) = 1;
    end
end

figure(1)
imshow(T); % title La imagen T

figure(2)
imshow(R); % title La imagen R

% Inicializamos las matrices u_x, u_y

ux = zeros(n,n);

uy = zeros(n,n);

D = zeros(n,n);
for i = 1:n
    for j = 1:n
```

```

D(i,j) = T(i - ux(i,j), j - uy(i,j)) - R(i,j);
end
end

Tx = zeros(n,n);
Ty = zeros(n,n);

% Definimos la matriz y los parámetros lambda y mu.
lambda = 100;
mu = -100;

Lh = - matriz_Elasticidad(n,h,lambda,mu);

% Comenzamos el método.

iter = 75;

tau = 1;
alpha = 1/8;

% Definimos la matriz A del sistema que luego resolveremos
A = speye(2*n*n) + tau * alpha * Lh;

nuevou = sparse(2*n*n,1);
nuevoux = sparse(n*n,1);
nuevouy = sparse(n*n,1);

aux = 2*h;

%%%%%PONEMOS LA IMAGEN T EN UNA MÁS GRANDE CON 0 EN EL RESTO
% Para luego interpolar T con i-ux, j-uy

TT = ones(2*n, 2*n);

for k = n/2:(3*n/2 - 1)
    for l = n/2:(3*n/2 - 1)
        TT(k,l) = T(k - n/2 + 1, l - n/2 + 1);
    end
end

Eelas = [];

for k = 1:iter

    k

    for i = 1:n
        for j=1:n
            IT(i,j) = interpola(n, TT, ux(i,j), uy(i,j), i, j);
        end
    end
end

```

```

        end
    end
    % Definimos la diferencia
    for i = 1:n
        for j = 1:n
            D(i,j) = IT(i,j) - R(i,j);
        end
    end

Eelas = [Eelas norm(D)];

    % Definimos el gradiente de T, Tx y Ty
    for i = 2:n-1
        for j = 1:n
            Tx(i,j) = (IT(i+1,j) - IT(i-1,j))/aux;
        end
    end

    for i = 1:n
        for j = 2:n-1
            Ty(i,j) = (IT(i,j+1) - IT(i,j-1))/aux;
        end
    end

    % Hay que resolver dos sistemas
    % Ax ux = bx // Ay uy = by

bx = ux + tau * D .* Tx;
by = uy + tau * D .* Ty;

    % Tenemos que pasar bx, by a vector (2*n*n,1) para resolver el sistema
    % A u = b

bx = reshape(bx', n*n, 1);
by = reshape(by', n*n, 1);

b = sparse(2*n*n,1);
b(1:n*n) = bx;
b(n*n+1:2*n*n) = by;

    % Ya podemos resolver el sistema:

nuevou = A\b;

    % Rellamamos ux, uy para la siguiente iteración

nuevoux = nuevou(1:n*n);
nuevouy = nuevou(n*n+1:2*n*n);

```

```
nux = reshape(nuevoux,n,n)';
nuy = reshape(nuevouy,n,n)';

ux = nux;
uy = nuy;

end

sol = zeros(n,n);
for i=1:n
    for j=1:n
        sol(i,j)= interpola(n, TT, ux(i,j), uy(i,j), i, j);
    end
end

figure(3);
imshow(sol);

end
```

## Referencias

- [1] REGISTRO DE IMÁGENES: *Definición y aplicaciones*. Documento accesible en la dirección electrónica [http://es.wikipedia.org/wiki/Registro\\_de\\_la\\_imagen](http://es.wikipedia.org/wiki/Registro_de_la_imagen).
- [2] REGISTRO DE LA IMAGEN MÉDICA: Documento accesible en la dirección electrónica <https://www5.cs.fau.de/research/areas/medical-image-registration/>.
- [3] I. M. Gelfand, S. V. Fomin: CALCULUS OF VARIATIONS.
- [4] Sven Kabus: *Multiple-Material Variational Image Registration*. Octubre 2006.
- [5] Noppadol Chumchob, Ke Chen: *A robust multigrid approach for variational image registration models*. Journal of Computational and Applied mathematics 236 (2011) 653-674.
- [6] B. Fischer, J. Modersitzki: *Fast diffusion registration*. Contemporary Mathematics 313 (2002) 117-129.
- [7] C. Broit: *Optimal registration of deformed images*. Doctoral Dissertation, University of Pennsylvania, August 1981.
- [8] L.E. Malvern, 1969. *Introduction to the mechanics of a continuous medium*. Prentice-Hall.
- [9] MÉTODO EN DIFERENCIAS FINITAS: Documento accesible en la dirección electrónica [https://www.uam.es/personal\\_pdi/ciencias/carlosp/html/pid/DiferenciasFinitas.html](https://www.uam.es/personal_pdi/ciencias/carlosp/html/pid/DiferenciasFinitas.html).
- [10] Profesor Y. Niño: *Método de los volúmenes finitos. Modelación numérica en ingeniería hidráulica y ambiental*. Departamento de Ingeniería Civil, Universidad de Chile.
- [11] TENSOR DE DEFORMACIONES: *Definición*. Documento accesible en la dirección electrónica [https://es.wikipedia.org/wiki/Tensor\\_deformaci%C3%B3n](https://es.wikipedia.org/wiki/Tensor_deformaci%C3%B3n).
- [12] TENSOR DE TENSIONES: *Definición*. Documento accesible en la dirección electrónica [https://es.wikipedia.org/wiki/Tensor\\_tensi%C3%B3n](https://es.wikipedia.org/wiki/Tensor_tensi%C3%B3n).