

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

MÉTODOS VARIACIONALES PARA LA RESTAURACIÓN DE DOMINIOS
PERDIDOS DE IMÁGENES Y SU RESOLUCIÓN NUMÉRICA MEDIANTE
MÉTODOS DE SEGUNDO ORDEN

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA MATEMÁTICA

MARIBEL KATERYN HERRERA TERÁN

`kateryn.herrera@epn.edu.ec`

Director: DR. JUAN CARLOS DE LOS REYES BUENO

`juan.delosreyes@epn.edu.ec`

QUITO, MARZO 2015

DECLARACIÓN

Yo MARIBEL KATERYN HERRERA TERÁN, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Maribel Kateryn Herrera Terán

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por MARIBEL KATERYN HERRERA TERÁN, bajo mi supervisión.

Dr. Juan Carlos De Los Reyes Bueno
Director del Proyecto

AGRADECIMIENTOS

A mis padres, por su esfuerzo y por creer en mí. A mis hermanos, por su alegría. Este trabajo es nuestro, fruto de las risas y lágrimas que juntos compartimos.

A mi director de tesis Juan Carlos De los Reyes, quien hizo posible mi participación en este proyecto de investigación, por sus palabras de aliento y por el tiempo dedicado a la supervisión de este trabajo.

A los colegas del Centro de Modelización Matemática por la oportunidad de ser parte de este grupo de trabajo.

Finalmente, al Instituto Espacial Ecuatoriano quien financió este proyecto de titulación, dentro del marco del proyecto: *Incidencia del Cambio Climático y Nutrición en Cultivos de Arroz, Maíz Duro y Papa con Modelos de predicción de cosechas mediante métodos espaciales y espectrales.*

DEDICATORIA

*A mis padres,
mis hermanos.
A la pequeña Amelia.
A Edwin, por su invaluable apoyo.*

Índice general

Resumen	IX
Abstract	X
1. Introducción	1
1.1. El rol de las imágenes en la sociedad	4
1.2. Imagen digital	5
1.3. Importancia del procesamiento digital de imágenes	6
1.4. Aplicaciones potenciales	7
1.5. Preliminares	9
1.6. Métodos en el procesamiento digital de imágenes	12
1.6.1. Métodos variacionales	13
1.6.2. Métodos basados en Ecuaciones Diferenciales Parciales(EDPs)	14
2. Modelos de restauración de imágenes	16
2.1. Filtrado de ruido (denoising)	16
2.2. Modelos de denoising basados en métodos variacionales	17
2.2.1. Regularización de Tikhonov	17
2.2.2. Regularización de variación total (TV)	17
2.3. Modelos de filtrado de ruido basados en EDPs	18
2.3.1. Ecuación del calor	18
2.3.2. Ecuación de difusión no lineal	19
2.4. Restauración de dominios perdidos (Inpainting)	19
2.5. Modelos de inpainting basados en métodos variacionales	20
2.5.1. Modelo TV para inpainting	21
2.5.2. Modelo de bordes incrustados (Embedded edge model)	22

2.5.3.	Modelos basados en la curva elástica de Euler	22
2.6.	Modelos basados en ecuaciones diferenciales parciales	23
2.6.1.	Enfoque de Bertalmio	24
2.6.2.	Modelo de curvatura media	25
2.6.3.	Modelo CDD, enfoque de Chan y Shen	25
2.6.4.	Ecuación modificada de Cahn-Hilliard	25
3.	Tratamiento numérico	27
3.1.	Aspectos metodológicos	27
3.2.	Métodos a un paso	28
3.2.1.	Propiedades de los métodos a un paso	29
3.3.	Regularizaciones	31
3.3.1.	Regularización Berkovier Engelman	32
3.3.2.	Regularización de Huber	33
3.4.	Método de Newton semi-suave	35
3.5.	Aproximación por Diferencias Finitas (FMD)	36
3.5.1.	En una dimensión	37
3.5.2.	En dos dimensiones	43
3.6.	Aplicaciones	46
3.6.1.	Problemas lineales	46
3.6.2.	Problemas no lineales	48
3.7.	Algoritmos propuestos para el procesamiento de imágenes	51
3.7.1.	Filtrado de ruido	51
3.7.2.	Inpainting	54
4.	Implementación y experimentos computacionales	61
4.1.	Comparación de modelos de inpainting	61
4.2.	Análisis experimental de los algoritmos implementados	62
4.3.	Aplicaciones de inpainting	73
4.3.1.	Remover texto	74
5.	Conclusiones	75
A.	Anexo A	77

A.1. Cálculo Variacional	77
A.1.1. Regularización de Tikhonov	78
A.1.2. Regularización de de variación total (TV)	79
B. Anexo B	82
B.1. Códigos de programación en Matlab	82
B.1.1. Algoritmos generales	82
B.1.2. Algoritmos para el procesamiento de imágenes	85
Bibliografía	98

Resumen

El procesamiento digital de imágenes es un conjunto de modelos y algoritmos que se aplican a imágenes deterioradas, con el objetivo de mejorar su calidad y facilitar la extracción de algún tipo de información, ya sea para la interpretación humana o de máquinas autónomas.

Entre los problemas más interesantes a tratar dentro del procesamiento de imágenes se encuentra el problema de inpainting, que consiste en reconstruir o restaurar partes deterioradas o perdidas de la imagen observada, a partir de la información disponible alrededor del área a ser recuperada, además de remover objetos que no sean de interés pues ocultan información.

En la última década, varios modelos de inpainting basados en distintos conceptos matemáticos han sido planteados para resolver este problema. En este trabajo, nos enfocaremos en la formulación variacional de Chan y Shen [34] basado en la regularización de variación total y el modelo basado en la *ecuación del movimiento de la curvatura media* [12].

No sólo el planteamiento y la ejecución de la restauración de imágenes son un proceso complejo, sino también su implementación computacional, por la gran cantidad de datos que se deben procesar. Es así que, en este trabajo se proponen algoritmos numéricos de segundo orden basados en el método Newton semi-suave, pues existen términos no definidos que regularizados presentan términos no diferenciables.

Abstract

Digital image processing is a set of models and algorithms that are applied to damaged images, with the goal of improving its quality and easy removal of disturbing information.

The *inpainting* or *filling in* is one of the most interesting problems in image processing. It consists in reinventing some missing parts using the information around these. The goal of it is to restore a damaged or corrupted image where the information has been lost. May also be an interesting tool for remove artificially some parts of an image such as overlapping text and modify the content.

Several kinds of approaches have been proposed to solve this problem. We will focus on the variational formulation proposed by Chan and Shen [34] based on total variation minimization and the model based on the mean curvature motion [12].

Not only the modeling of image restoration is a complex process, but also the numerical implementation, due to the large amount of data to be processed. Thus, in this paper we propose second-order algorithms based on semi-smooth Newton by the presence of non-differentiable terms product of regularization of terms not defined.

Capítulo 1

Introducción

En la sociedad actual, las imágenes digitales desempeñan un papel cada vez más importante, pues son fuente de información dentro de diversas áreas como: la medicina, sistemas de información geográfica, astronomía, ciencia de los materiales, industria del entretenimiento y la industria de la seguridad, entre otros.

A menudo, se designa a la sociedad como *sociedad de la información* aunque, podría definirse también como *sociedad de la imagen*. Ya que, las imágenes digitales juegan un papel cada vez más importante no sólo como medio de comunicación, sino también porque permiten la representación compacta del mundo físico que nos rodea.

Si lo pensamos con detenimiento, es sorprendente darse cuenta de lo mucho que dependemos de las imágenes, pues están presentes a diario en nuestras vidas. El origen de tal fenómeno está en los avances de los dispositivos de adquisición, además del aumento de la capacidad de los equipos para procesar más y más datos. Esto ha dado lugar a una nueva disciplina conocida como la *visión computacional*. El extenso campo de aplicación de los mecanismos de visión artificial tales como: rayos X, tomografía computarizada, resonancias magnéticas, imágenes satelitales y cadenas de montaje, es una de las causas por las que el procesamiento digital de imágenes ha ganado gran interés en las últimas décadas.

El procesamiento digital de imágenes es un conjunto de modelos y algoritmos que se aplican a imágenes deterioradas con el objetivo de mejorar la calidad y facilitar la extracción de algún tipo de información, ya sea para la interpretación humana o de máquinas autónomas. Los avances computacionales han permitido implementar métodos más robustos que han hecho atractiva la aplicación de procesamientos digitales en el mundo real.

Entre los problemas más interesantes a tratar dentro del procesamiento de imágenes se encuentran:

- **Filtrado de ruido (denoising):** proceso mediante el cual se recupera a la imagen de borrosidades por la intervención de ruido, filtrando el mismo.

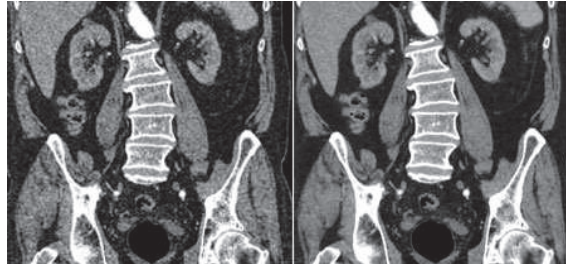


Figura 1.1. Filtrado de ruido de una tomografía.

- **Deconvolución (deblurring):** proceso de corrección de una imagen, cuando los píxeles se ven afectados por un movimiento brusco en el momento en el que se capturó la imagen, creando un efecto de desenfocado (blurring).

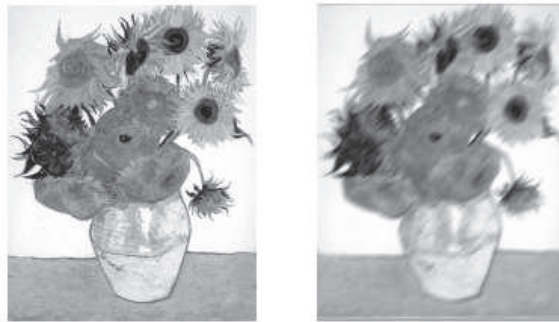


Figura 1.2. Desenfocado en una fotografía.

- **Escalado (zooming):** en ocasiones el envío de imágenes de alta resolución se ve limitado por el ancho de banda de transmisión, siendo necesaria una compresión de datos. Durante la descompresión, las imágenes pierden información y se requiere un algoritmo de escalado capaz de estimar la información y minimizar así la pérdida de información.



Figura 1.3. Escalado de una fotografía.

- **Restauración de dominios perdidos en una imagen (inpainting):** proceso que permite reconstruir partes deterioradas, perdidas o con obstrucción de objetivos que ocultan información en una imagen.



Figura 1.4. Restauración de dominios perdidosFuente: Elaboración propia.

- **Segmentación:** localización de objetos dentro de una imagen.

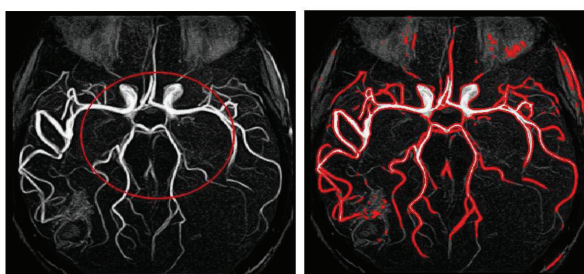


Figura 1.5. Segmentación del sistema neurológico.

El tema central de este trabajo es la restauración de dominios perdidos, pues es sin duda un campo de especial interés tanto desde el punto de vista matemático como comercial. Este es un importante desafío en la sociedad moderna, ya que se pueden reconstruir imágenes: satelitales, moleculares, médicas y de obras de arte antiguo para su restauración. Motivada en estas aplicaciones, se presentan modelos basados en ecuaciones diferenciales parciales y el cálculo de variaciones, que buscan resolver este problema.

La alta complejidad de los métodos de inpainting hace menos atractiva la manipulación de imágenes y su aplicación en el mundo real. No sólo el planteamiento y la ejecución de la restauración de imágenes son un proceso complejo, sino también su implementación computacional, por la gran cantidad de datos que se deben procesar. Es así que, en este trabajo se proponen algoritmos numéricos de segundo orden basados en el método Newton semi-suave, que resuelva el problema eficientemente.

El presente trabajo será de utilidad en futuras investigaciones en el campo del procesamiento matemático de imágenes y visión artificial, permitiendo que dichos campos se desarrollen en nuestro país y puedan adquirir un nivel científico considerable.

Se presenta, en el primer capítulo la importancia y las potenciales aplicaciones del procesamiento matemático de imágenes. Luego, en el segundo capítulo se recopilan los principales modelos de restauración de imágenes basados en métodos variacionales y en ecuaciones en derivadas parciales. En el tercer y cuarto capítulo se detallan el tratamiento numérico y experimentos computacionales respectivamente. Finalmente, se muestran y analizan los resultados obtenidos.

1.1. El rol de las imágenes en la sociedad

A menudo, se designa a la sociedad como *sociedad de la información* aunque, podría definirse también como *sociedad de la imagen*. Las imágenes digitales están presentes en una gran cantidad de situaciones, desde la vida cotidiana hasta su uso profesional en: medicina, biología, ciencias de la tierra, predicción meteorológica, las artes, aplicaciones de seguridad entre otras, como lo muestra la Figura 1.6. El origen de tal fenómeno está en los avances de los dispositivos de adquisición, además del aumento de la capacidad de los equipos para procesar más y más datos. Esto ha dado lugar a una nueva disciplina conocida como la *visión computacional*, donde se organizan y procesan las imágenes digitales.

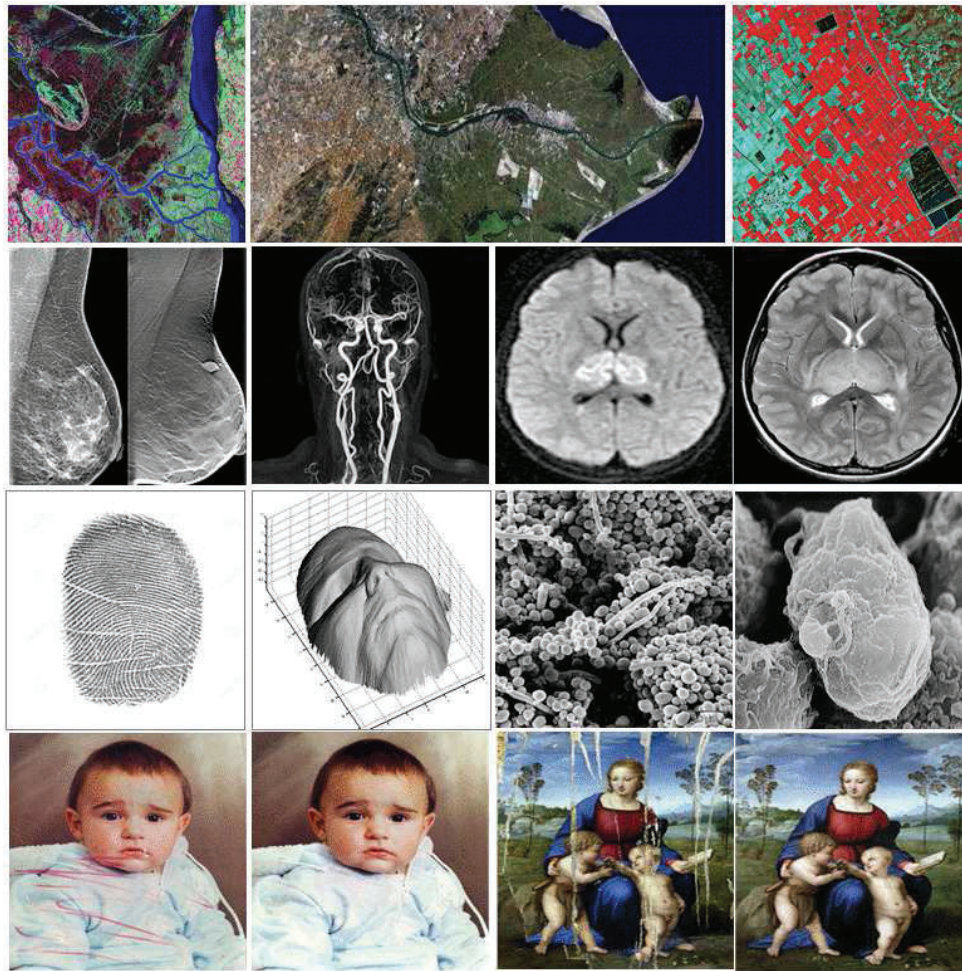


Figura 1.6. Imágenes digitales en la sociedad. **Fuente:** Elaboración propia.

Por ejemplo, en el área médica se ha hecho un uso considerable de las imágenes, pues varios diagnósticos se basan en ultrasonidos, rayos X, tomografías, resonancias, etc. Las mismas que se pueden procesar para mejorar su calidad y extraer así, información con mayor precisión. Otro campo importante es la teledetección, donde se busca realizar un seguimiento y cuantificación de los cambios en los bosques, suministros de agua, urbanización y la

contaminación. Además de aplicaciones como: la vigilancia de vídeo, el análisis del tráfico por carretera y la observación de recursos de la tierra.

1.2. Imagen digital

Con el fin de apreciar la importancia del procesamiento digital de imágenes, es necesario entender el proceso para pasar de una imagen continua a una digital.

En términos generales, una imagen digital o también conocida como *imagen discreta* se obtiene a partir de una imagen analógica mediante *muestreo y cuantización*. El *muestreo* se refiere a la discretización de las coordenadas, y la *cuantización* a digitalizar los valores de la amplitud que definirán los tonos de gris, en el caso de imágenes en escala de grises. En la Figura 1.7 se observa de izquierda a derecha, una imagen continua cortada a lo largo por el segmento AB, a la derecha la función unidimensional extraída, para luego realizar el proceso de muestreo y cuantización, que se muestra en la parte inferior.

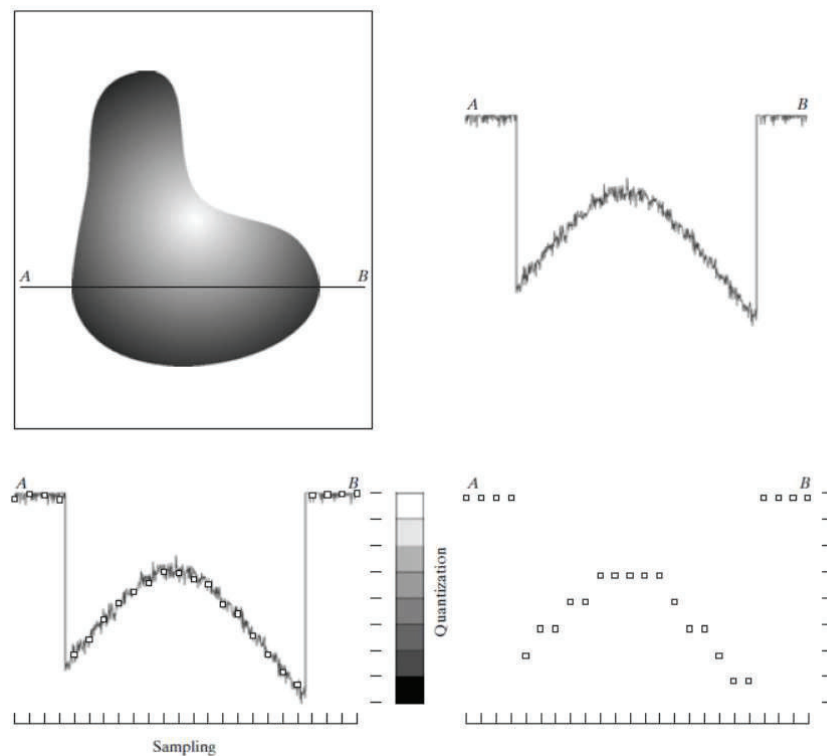


Figura 1.7. Proceso de muestreo y cuantización. Fuente: [19]

En la Figura 1.8 se muestra una imagen digital después de aplicar el proceso de muestreo y cuantización. Como se observa, una imagen digital es la aproximación de una imagen continua. Esta aproximación dependerá de la cantidad de unidades de muestreo y cuantización. La resolución de una imagen dependerá de las unidades de muestreo, mientras que el tamaño está definido por las unidades de cuantización.

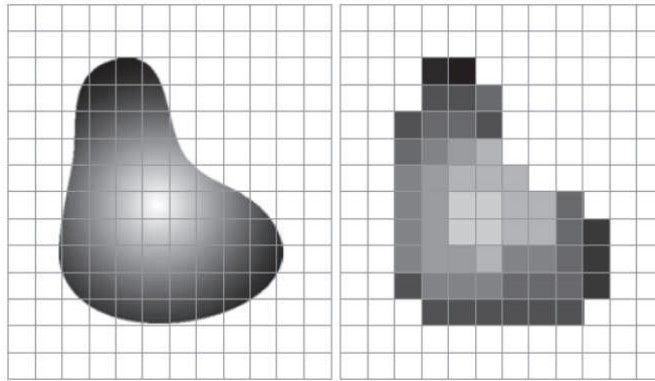


Figura 1.8. A la izquierda, la imagen continua y a la derecha la imagen después de aplicar el proceso de muestreo y cuantización. **Fuente:** [19]

Resumiendo, la idea es superponer una rejilla sobre una imagen analógica y asignar un número a cada cuadrado de la rejilla, que puede ser el brillo o el nivel de gris promedio en esa casilla. Cada casilla se conoce como *píxel* (abreviatura picture element). Esta malla de píxels puede ser tratada como una matriz de n filas y m columnas, que corresponderá a la representación matricial de una imagen. Sólo para dar una idea, de las cámaras digitales se obtienen imágenes de tamaño 320×240 y pueden alcanzar hasta 3060×2036 en cámaras profesionales. El tamaño de imágenes médicas de resonancia magnética está alrededor de 128×128 .

En una imagen a color, por ejemplo en el modelo de colores *RGB* (Red-Green-Blue), se tienen tres matrices que representan la intensidad de cada color por píxel, como se observa en la Figura 1.9.

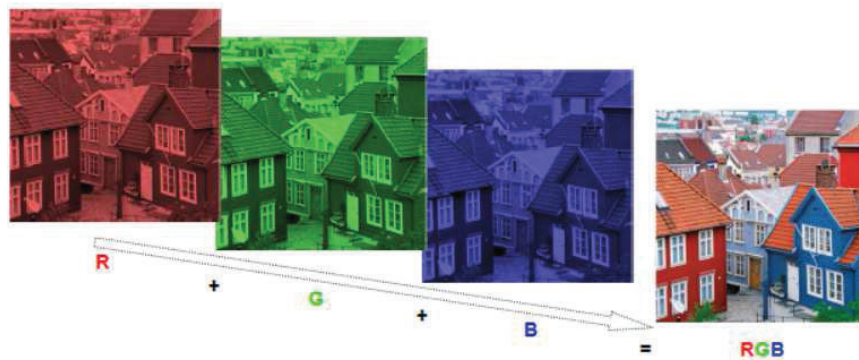


Figura 1.9. Superposición de las componentes de color en el modelo RGB.

1.3. Importancia del procesamiento digital de imágenes

En las dos últimas décadas, el procesamiento digital de imágenes ha ganado gran importancia dentro de distintas áreas de la ciencia, debido a que cada día la información visual

es más importante y abundante.

Sin necesariamente saberlo, somos consumidores del procesamiento de imágenes, pues a menudo tenemos que lidiar con este. Es más, la mayor parte de información que procesamos los seres humanos está en forma de imágenes y de alguna manera, nosotros realizamos procesamiento de imágenes sin notarlo, por ejemplo, cuando miramos algo, la primera imagen que los ojos envían al cerebro está fuera de foco; el cerebro intenta corregir esto ajustando los lentes oculares, así una nueva imagen es enviada de los ojos al cerebro. Otro ejemplo es la *estereovisión*, donde el cerebro convierte las imágenes bidimensionales registradas por nuestros ojos en una representación tridimensional, la que nos permite comprender la profundidad de los objetos. Estos procesos de retroalimentación son tan rápidos que no se pueden percibir ni sentir [19].

El campo del procesamiento digital de imágenes estudia métodos que permitan mejorar la calidad y la extracción de información de imágenes. Algunos ejemplos de imágenes dañadas provienen de:

- herramientas médicas por la falta de definición o el ruido presente en: resonancias magnéticas del cerebro, tomografías de órganos internos como el corazón, rayos X de nuestro esqueleto, entre otros.
- Satélites, pues las imágenes de la tierra a menudo son borrosas o presentan nubosidades que ocultan información relevante.
- Cámaras fijas, instaladas en cadenas de montaje con poca iluminación, producen blurring, pues la cámara debe emplear un tiempo de exposición relativamente alto.
- Robots equipados con cámaras empleados en procesos de fabricación como el control de calidad.

1.4. Aplicaciones potenciales

En esta sección, se presenta una recopilación de las potenciales aplicaciones del procesamiento matemático de imágenes (ver [23]).

- **Restauración digital**, útil en la preservación y restauración de arte como se muestra en la Figura 1.10, donde se toman fotografías de obras antiguas de arte para recuperar o completar la información perdida. Las mismas que se almacenan para luego ser utilizadas como plantillas para restauradores de arte.



Figura 1.10. Restauración de arte. Fuente: Elaboración propia.

- **Reconocimiento y rastreo de objetos**, se podría utilizar en el control de calidad de partes industriales, donde se requiere extraer o seguir dichas partes para constatar la correcta posición y diámetro de los agujeros, como se muestra en la Figura 1.11. Se utilizan aquí la segmentación de imágenes, el reconocimiento de objetos y la teledetección.

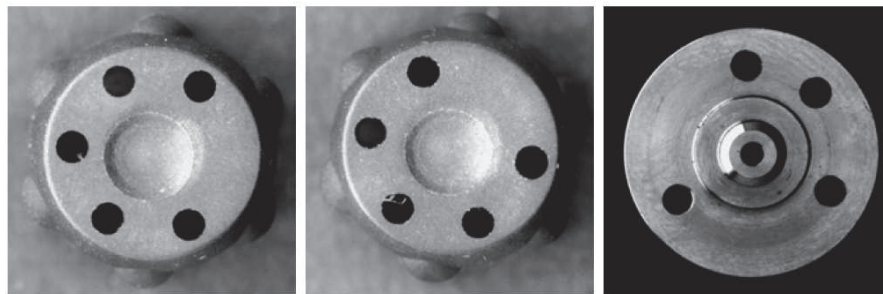


Figura 1.11. Control de calidad de partes industriales. Fuente: [23].

- **Reconstrucción en 3D**, la profundidad de los objetos se pierde en las imágenes y se requieren técnicas para recuperarla. Un ejemplo de esto son las imágenes de *Resonancia Magnética*, donde gracias a la técnica de imágenes volumétricas, se puede observar el interior de objetos en 3D Figura 1.12.

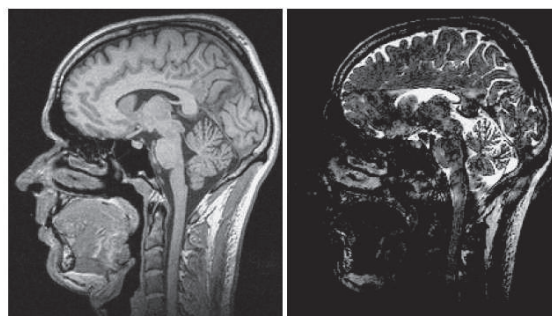


Figura 1.12. Imágenes de resonancia magnética. Fuente: [23].

- **Clasificación de objetos**, en tareas de inspección industrial se busca reconocer obje-

tos con características específicas para su posterior clasificación. En la Figura 1.13 se muestra a la derecha conectores en distintas posiciones y a la izquierda el conector a identificar.

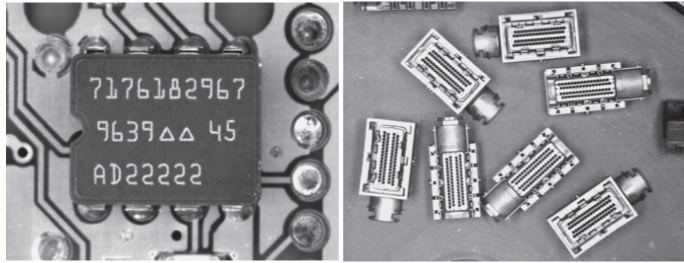


Figura 1.13. Reconocimiento de una etiqueta sobre un circuito integrado. **Fuente:** [23].

No sólo se necesitan procesar imágenes, sino también secuencias de imágenes, por lo que el procesamiento de vídeo se está convirtiendo en un área con mayor interés. Esto es necesario, por ejemplo para **pronosticar el clima**, mediante la estimación y la predicción del movimiento de las nubes. En el reconocimiento facial de personas en movimiento en un aeropuerto, por mencionar algunas aplicaciones.

1.5. Preliminares

La idea de una imagen digital es considerar una malla rectangular de píxeles y asignar a cada píxel un número, que representará el nivel de intensidad de gris o brillo en la imagen. Para hacer uso de los modelos basados en ecuaciones diferenciales parciales y el cálculo de variaciones, es necesaria una representación continua de las imágenes digitales, que corresponderán a una interpolación de la imagen digital. Por lo que es necesario introducir las siguientes definiciones:

DEFINICIÓN 1.1. (Espacio L^p) Sea Ω un subconjunto abierto de \mathbb{R}^n y $1 \leq p < \infty$, se define

$$L^p = \{f : \Omega \rightarrow \mathbb{R}, f \text{ medible y } \int_{\Omega} |f(x)|^p dx < \infty\}.$$

Este espacio está dotado de la norma

$$\|f\|_{L^p} = \left[\int_{\Omega} |f(x)|^p dx \right]^{1/p}.$$

DEFINICIÓN 1.2. (Espacio L^∞)

$$L^\infty = \{f : \Omega \rightarrow \mathbb{R}, f \text{ medible y } |f(x)| < C \text{ en casi todo punto de } \Omega \text{ para alguna constante } C > 0\}.$$

Este espacio está dotado de la norma

$$\|f\|_{L^\infty} = \inf \{C : |u(x)| \leq C \text{ en casi todo punto de } \Omega\}.$$

OBSERVACIÓN. $L^p(\Omega)$ es un espacio cuyos elementos son clases de equivalencia de funciones, donde dos funciones pertenecen a la misma clase, si en casi todo punto (c.t.p) de Ω coinciden, definiendo así la siguiente relación de equivalencia:

$$f \sim g \text{ si y solo si } f(x) = g(x), \text{ c.t.p. en } \Omega.$$

Sin embargo, es común considerar $L^p(\Omega)$ como un espacio de funciones.

Sea $\alpha = (\alpha_1, \dots, \alpha_N)$ un multi-índice y D^α el operador de diferenciación con respecto al multi-índice, es decir

$$D^\alpha = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_N^{\alpha_N}}$$

con $|\alpha| = \sum_{i=1}^N \alpha_i$.

DEFINICIÓN 1.3. (Espacio de Sobolev) Sea $1 \leq p < \infty$ y $k \in \mathbb{N}$. El espacio de las funciones $y \in L^p(\Omega)$, cuya derivada $D^\alpha y$, para $|\alpha| \leq k$, existe y pertenece a $L^p(\Omega)$ se denota por $W^{k,p}(\Omega)$ y se conoce como espacio de Sobolev. Este espacio está dotado de la norma

$$\|f\|_{W^{k,p}} = \left(\sum_{|\alpha| \leq k} \int_{\Omega} |D^\alpha y|^p dx \right)^{1/p}.$$

Sea $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, $|\cdot|$ denotará la norma euclideana,

$$|x| = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}.$$

OBSERVACIÓN. En el caso especial de $p = 2$, el espacio de Sobolev se denota por $H^k(\Omega) = W^{k,2}(\Omega)$. Un espacio utilizado con frecuencia es $H^1(\Omega)$,

$$H^1(\Omega) = \{y \in L^2(\Omega) : D_i y \in L^2(\Omega), \forall i = 1, \dots, N\}$$

dotado de la norma

$$\|y\|_{H^1} = \left(\int_{\Omega} (y^2 + |\nabla y|^2) dx \right)^{1/2},$$

y el producto escalar

$$(y, z)_{H^1} = \int_{\Omega} y \cdot z dx + \int_{\Omega} \nabla y \cdot \nabla z dx.$$

DEFINICIÓN 1.4. (Imagen) Sea $\Omega \subset \mathbb{R}^n$ un abierto acotado, se define una imagen u como una función acotada $u : \Omega \rightarrow \mathbb{R}$, es decir, $u \in L^\infty(\Omega)$.

El dominio Ω más frecuente para una imagen u es $\Omega = [0, 1] \times [0, 1]$.

DEFINICIÓN 1.5. (Función de variación acotada en Ω) El conjunto de las funciones u de va-

riación acotada $BV(\Omega)$ es un subconjunto de $L^1(\Omega)$, tal que

$$|Du|(\Omega) < \infty,$$

donde

$$|Du|(\Omega) = \sup \left\{ \int_{\Omega} u(x) \operatorname{div} \phi(x) dx : \phi \in C_c^1(\Omega, \mathbb{R}^n), \|\phi\|_{L^\infty(\Omega)} \leq 1 \right\}$$

es la variación total de u .

Es decir, una función $u \in BV(\Omega)$ es integrable y existe una medida Du tal que

$$\int_{\Omega} u(x) \operatorname{div} \phi(x) dx = \int_{\Omega} \langle \phi(x), Du(x) \rangle dx \text{ para todo } \phi \in C^1(\Omega, \mathbb{R}^n).$$

OBSERVACIÓN. Si u es suficientemente suave, es decir, $\nabla u(x) \in L^1(\Omega)$, entonces la variación total de u puede expresarse como:

$$Du(x) = \nabla u(x) dx$$

y la seminorma de variación total es equivalente a la magnitud de su gradiente,

$$\|u\|_{TV(\Omega)} = \int_{\Omega} |\nabla u(x)| dx. \quad (1.1)$$

Para tener una idea clara de las funciones de variación acotada, se analiza el caso unidimensional. Sea $f: [a, b] \mathbb{R} \rightarrow \mathbb{R}$, se define la variación total de f en $[a, b]$ como:

$$\|f\|_{TV(\Omega)} = \sup_{P \in \mathfrak{P}_{[a,b]}} \sum_{i=1}^n |f(x_i) - f(x_{i-1})|$$

donde P es una partición del conjunto de particiones $\mathfrak{P}[a, b]$ del intervalo $[a, b]$. En la Figura 1.14 se muestran dos funciones que siendo acotadas en el intervalo $[0, 1]$, no son de variación acotada, mientras que en la Figura 1.15 se muestran dos funciones que pertenecen a $BV([0, 1])$.

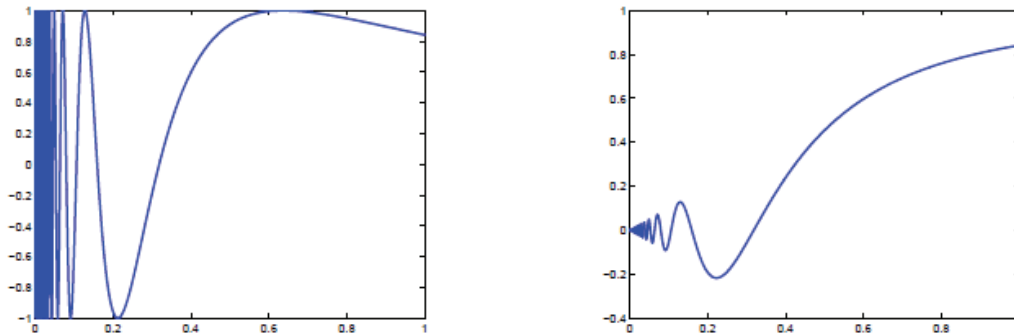


Figura 1.14. Ejemplo de funciones que no son de variación acotada en $[0, 1]$.

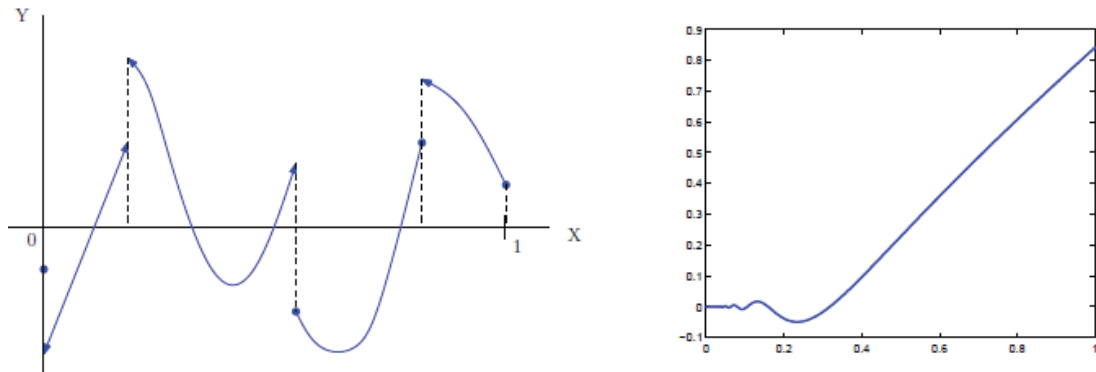


Figura 1.15. Funciones de variación acotada en $[0, 1]$.

En estos ejemplos, se observa que las funciones de variación acotada son funciones sin oscilaciones de alta frecuencia. El concepto de variación total se aplica a imágenes con regiones que presentan discontinuidades, que pueden interpretarse como los bordes de dichas regiones.

1.6. Métodos en el procesamiento digital de imágenes

A lo largo de los últimos años, se han desarrollado varios enfoques para abordar el procesamiento digital de imágenes. Se pueden mencionar los principales métodos basados en: la teoría del filtro, el análisis espectral, o en algunos conceptos básicos de probabilidad y estadística. En el libro de Jain [24] se pueden consultar el detalle de estos modelos.

Otro tipo de herramientas se basan en modelos estocásticos, la transformada de Wavelets, en ecuaciones en derivadas parciales (EDPs), y en métodos variacionales. Los modelos estocásticos hacen uso de la teoría del campo aleatorio de Markov ([26], [17]), mientras que la transformada de Wavelet emplea técnicas de descomposición [15].

En este trabajo nos enfocaremos en los métodos basados en EDPs y en el cálculo de variaciones, conocido como *método de energía*, los mismos que se han desarrollado intensamente en el análisis de imágenes desde la década de los 90. Las ecuaciones en derivadas parciales están estrechamente relacionadas con conceptos físicos, que con pequeñas variantes se han adaptado para su uso en el procesamiento digital de imágenes. Los modelos basados en el método de energía pueden explicarse mejor desde el punto de vista de los problemas inversos, donde dada una imagen dañada f se quiere hallar una imagen u sin daños, el más simple y común de estos defectos, es el ruido. Es así que, para lograr resolver el problema de los dominios perdidos resolveremos primero el problema de *filtrado de ruido*.

1.6.1. Métodos variacionales

Los problemas inversos en el procesamiento digital de imágenes están asociados a un esquema estándar descrito por:

$$f = Tu + \eta$$

donde f es la imagen dañada conocida y T un operador lineal que representa la influencia del sistema óptico sobre la imagen original u . Dependiendo de las características de este operador, aparecen diferentes problemas como: denoising, deblurring, zooming, inpainting y la descomposición geométrica de una imagen. Se supone que η es una variable aleatoria con media cero y varianza σ^2 conocida, que podría representar ruido.

Así, la idea es encontrar una imagen u a partir de la imagen observada f . La solución más obvia sería

$$T^{-1}f = u + T^{-1}\eta.$$

En la práctica esto no es factible; ya que el operador T muchas veces es no invertible o su inversa es imposible de calcular, además que la presencia de la variable aleatoria η da lugar a un problema *mal condicionado*. Hadamard [21] fue el primero en establecer los criterios del condicionamiento de un problema como éste.

DEFINICIÓN 1.6. Se dice que un problema está bien condicionado si se verifica que:

- **Existencia**, para cualquier f dado, existe una solución u .
- **Unicidad**, la solución u es única.
- **Estabilidad**, la solución u depende en forma continua del dato f .

En los problemas inversos, la condición de estabilidad es constantemente quebrantada. Para abordar este inconveniente, se utilizan *métodos de regularización*, los mismos que buscan restaurar la estabilidad de la solución con relación a los datos. En términos generales, se puede decir que la regularización permite aproximar un problema mal planteado por medio de una familia cercana de problemas bien planteados.

Los métodos usados en el procesamiento de imágenes se basan principalmente en la minimización de la energía de un funcional aplicado a una imagen u , por lo que se los conoce como *métodos de energía* [1]. En forma general, están determinados por:

$$E[u] := R[u] + \lambda \|f - u\|^2 \tag{1.2}$$

donde,

- el primer término se conoce como *término regularizador* o de *suavidad*. Este término cuantifica la variación dentro de una propia imagen, tratando que el gradiente de u varíe suavemente. Busca que las esquinas y bordes de la imagen queden lo mejor definidas posibles.

- El segundo, es la distancia entre f y u que cuantifica el nivel de fidelidad entre la imagen solución u (imagen procesada) y la imagen original f (imagen defectuosa observada).
- El parámetro de regularización λ , determina la influencia de los términos. Su valor es crítico para conseguir el equilibrio entre la estabilidad de la solución y la calidad de los datos. De hecho, si λ es demasiado pequeño, se obtiene una solución sobre-regularizada y el suavizado en la imagen restaurada es excesivo. Por el contrario, si el parámetro λ es demasiado grande se llegaría a la misma solución inestable por la presencia de ruido.

Para aplicar este método se parte de la hipótesis de que la imagen defectuosa f contiene suficiente información para ser restaurada, y así obtener un alto porcentaje de mejoras respecto a su estado original. Dependiendo de la elección del término de regularización $R[u]$ se han desarrollado diversos modelos de restauración de dominios perdidos.

1.6.2. Métodos basados en Ecuaciones Diferenciales Parciales(EDPs)

En la sección anterior se consideró un enfoque variacional, que plantea resolver el problema de minimización del funcional (1.2). La condición necesaria de optimalidad para encontrar un mínimo de E , bajo ciertos supuestos, esta dada por la ecuación de *Euler-Lagrange*, donde la primera variación de E tiene que ser cero. En $L^2(\Omega)$, será

$$-\nabla R[u] + 2\lambda(f - u) = 0 \text{ en } \Omega, \quad (1.3)$$

que no es más que una ecuación en derivadas parciales, donde $\nabla R[u]$ denota la derivada según Fréchet de R en $L^2(\Omega)$, que en términos más generales, es un elemento del subdiferencial de $R[u]$.

Para resolver esta ecuación se puede plantear un esquema de evolución en el tiempo o también conocida como *time marching*. Esta técnica consiste en reformular el problema tal que, se busca el estado estacionario de la versión dinámica de la ecuación (1.3):

$$\frac{\partial u}{\partial t} = -\nabla R[u] + 2\lambda(f - u)$$

con $u(x, 0) = f(x)$. Por tanto, pasamos de una ecuación estacionaria a una evolutiva con condición inicial. En esta formulación se debe destacar la introducción artificial de la variable temporal t , que permite describir la evolución de la función u hasta estar lo suficientemente cerca de un mínimo de E . En el contexto de las imágenes, no se crean nuevas estructuras, sino se va transformando la imagen inicial.

Existen EDPs que no provienen de formulaciones variacionales como son los modelos de CDD inpainting [11], Cahn-Hilliard, y TV- H^1 inpainting. Otra posibilidad es trabajar directamente con ecuaciones, sin pensar en la energía. En general, la idea de los métodos

basados en EDPs es hallar la solución de una ecuación diferencial no lineal, de la forma:

$$\begin{cases} \partial_t u + F(x, u(x, t), \nabla u(x, t), \nabla^2 u(x, t)) = 0 & \text{si } (x, t) \in \Omega \times (0, T] \\ u(x, 0) = f(x) & \text{si } x \in \Omega \\ \partial_n u = 0 & \text{si } (x, t) \in \partial\Omega \times (0, T] \end{cases} \quad (1.4)$$

donde $u(x, t)$ es la versión restaurada de la imagen degradada inicial $u(x, 0) = f$, ∇u y $\nabla^2 u$ representan el gradiente y la Hessiana de u respectivamente, que dependen del espacio de la variable x . El proceso evolutivo que genera la ecuación (1.4) da lugar a una familia de imágenes $\{u(x, t)\}_{t>0}$ a partir de la imagen inicial f . A medida que t aumenta, se espera que $u(x, t)$ sea cada vez una imagen más y más simplificada de f , como lo muestra la Figura 1.16. Por esta razón, se denomina a t como variable de escala (*scale-space theory* [25]). La evolución de u depende del orden de la EDP.



Figura 1.16. Evolución en el tiempo.

Como se vera más adelante, la elección de F en (1.3) es determinante en el propósito de preservar características tales como bordes, esquinas y uniones.

Capítulo 2

Modelos de restauración de imágenes

En este trabajo, nos enfocaremos en la formulación variacional de Chan y Shen [34] basado en la regularización de variación total y el modelo basado en la *ecuación del movimiento de la curvatura media* [12] para resolver el problema de dominios perdidos.

La técnica de *regularización de variación total (TV)* fue desarrollada por Rudin, Osher y Fatemi [33], para dar solución al filtrado de ruido gaussiano. Sin embargo, esta técnica ha sido aplicada en varios problemas del procesamiento matemático de imágenes, como el problema de dominios perdidos, que es de interés en este trabajo. Es así que, para lograr resolver el problema de inpainting resolveremos primero el problema de filtrado de ruido.

2.1. Filtrado de ruido (denoising)

El filtrado de ruido, se refiere al proceso de remover el ruido de una imagen, en donde dada una imagen f con ruido se pretende encontrar una imagen limpia de ruido u , que conserve las características de la imagen observada f .

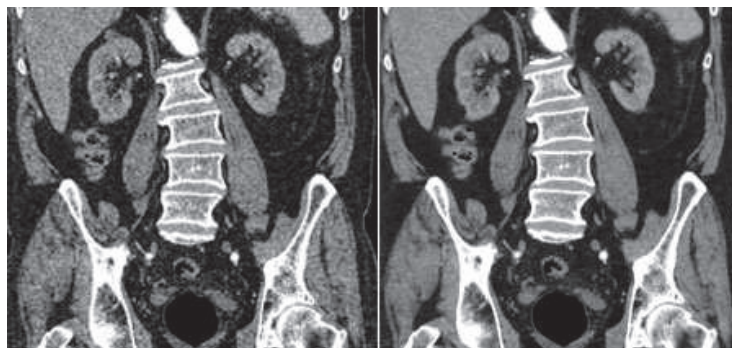


Figura 2.1. Filtrado de ruido de una tomografía.

El caso más estudiado es el *ruido blanco Gaussiano*, donde el operador T es el operador identidad y la imagen con ruido f se puede describir a través de la imagen u sin ruido más

una perturbación de la misma, es decir que el problema inverso es:

$$f = u + \varepsilon$$

donde ε es una variable aleatoria independiente con distribución normal de media cero y varianza σ^2 . La presencia de esta variable aleatoria da lugar a un problema *mal condicionado*. Como se presentó en el capítulo anterior, los métodos variacionales se emplean para resolver este inconveniente.

2.2. Modelos de denoising basados en métodos variacionales

2.2.1. Regularización de Tikhonov

Una forma clásica de superar el problema de condicionamiento es añadir un término de regularización a la energía. Esta idea fue introducida en 1977 por Tikhonov y Arsenin [36], donde los autores proponen considerar el siguiente problema de minimización, utilizando la norma $L^2(\Omega)$:

$$E[u|f] = \varepsilon \int_{\Omega} |\nabla u(x)|^2 dx + \frac{\lambda}{2} \int_{\Omega} (u(x) - f(x))^2 dx. \quad (2.1)$$

Se busca u que mejor se ajuste a los datos, tal que su gradiente sea bajo. El problema de minimizar $E[u]$ con $u \in W^{1,2}(\Omega)$ admite una única solución caracterizada por la ecuación de Euler-Lagrange (ver anexo),

$$-2\varepsilon \Delta u + \lambda(u - f) = 0,$$

con condiciones homogéneas de Neumann en el borde $\frac{\partial u}{\partial n} = 0$, donde n es el vector normal exterior a $\partial\Omega$.

Como se puede observar, se penalizan las discontinuidades, puesto que encontramos sólo soluciones suaves. Nuestro objetivo es mantener las características generales de la imagen inicial f , es decir, se busca determinar soluciones discontinuas, si es el caso. Se puede considerar, reemplazar la norma $L^2(\Omega)$ en la regularización de Tikhonov por la norma $L^1(\Omega)$, esta es la llamada *regularización de variación total (TV)*.

2.2.2. Regularización de variación total (TV)

Rudin, Osher, y Fatemi [33] propusieron el siguiente problema de minimización

$$\min_{u \in BV(\Omega)} E_{\lambda}[u|f] \quad (2.2)$$

con,

$$E_{\lambda}[u|f] = \|u\|_{TV(\Omega)} + \frac{\lambda}{2} \int_{\Omega} (f(x) - u(x))^2 dx$$

donde la solución u es la aproximación de la imagen sin ruido, y λ una constante positiva. Así, el problema de reducir el ruido se resuelve como un problema de minimización $n - dimensional$, donde el espacio de búsqueda es el espacio de todas las imágenes de variación acotada (BV). Cuando u es suave, es decir, $\nabla u(x) \in L^1(\Omega)$ se tiene (1.1), de donde el problema de minimización puede escribirse:

$$\min_{u \in BV(\Omega)} \int_{\Omega} |\nabla u(x)| dx + \frac{\lambda}{2} \int_{\Omega} (f(x) - u(x))^2 dx. \quad (2.3)$$

El primer termino de (2.3) tiene como objetivo que la función u varíe suavemente en Ω , es decir, evita oscilaciones en la solución; mientras que el segundo, garantiza que la solución u preserve la características de la imagen observada f . Si $f \in L^2(\Omega)$, el problema de minimización (2.2), tiene solución única en $L^2(\Omega)$ (ver [20]).

Se han propuesto varios algoritmos numéricos para minimizar E_{λ} con $u \in BV(\Omega)$. La mayoría de estos se basan principalmente en tres enfoques la optimización directa, el uso de la variable dual de forma explícita en el proceso de solución, y la resolución de la ecuación de Euler-Lagrange asociada. Nos centraremos en el último enfoque.

La condición necesaria de optimalidad para encontrar un mínimo de E_{λ} bajo ciertos supuestos, esta dada por la ecuación de *Euler-Lagrange*, donde la primera variación de E_{λ} tiene que ser cero; el cálculo de esta se presenta en el anexo (1.2). Se busca la solución de la ecuación

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda(u - f) = 0. \quad (2.4)$$

2.3. Modelos de filtrado de ruido basados en EDPs

El objetivo de esta sección es presentar algunos modelos basados en EDPs clásicas para la restauración de imágenes, tratando de seguir el orden cronológico en el que aparecieron. Las ecuaciones en derivadas parciales están estrechamente relacionadas con conceptos físicos, que pueden ser adaptadas para su uso en el procesamiento digital de imágenes. Como es el caso de la difusión isotrópica, que es una técnica que apunta a la eliminación del ruido en una imagen sin afectar los bordes.

2.3.1. Ecuación del calor

Suponga que en una habitación se tiene una fuente de calor en el medio. Con el pasar del tiempo, el calor se propaga por la habitación en círculos concéntricos, los cuales van perdiendo intensidad a medida que se propagan. Es decir, el calor avanza y se aleja de su fuente y va disminuyendo, ya que la temperatura de la habitación tiende a ser homogénea, por la transferencia de calor de la fuente. En el contexto del procesamiento de imágenes, si se considera a la habitación como una imagen y la fuente de calor como un punto originado por ruido, se puede imaginar como el ruido se propaga y se atenúa a través de la imagen. Se puede emplear la *ecuación de calor* para resolver el problema del filtrado de ruido.

La ecuación más antigua y más investigada en el procesamiento de imágenes es, probablemente, la ecuación parabólica lineal de calor [35]:

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) - \Delta u(x, t) = 0 & \text{con } x \in \Omega \text{ y } t \geq 0 \\ u(x, 0) = f(x). \end{cases} \quad (2.5)$$

A continuación, se presentan ecuaciones diferenciales parciales no lineales como modelos para la restauración de imágenes.

2.3.2. Ecuación de difusión no lineal

Consideremos la EDP, propuesta inicialmente por Perona y Malik [31] en 1987, que consiste en realizar una difusión selectiva a la imagen, introduciendo en la ecuación de calor un coeficiente de conductividad no lineal.

$$\begin{cases} \frac{\partial u}{\partial t} = \operatorname{div}(g(|\nabla u|^2)\nabla u) & \text{con } (x, t) \in \Omega \times (0, T) \\ \frac{\partial u}{\partial t} = 0 & \text{si } (x, t) \in \Omega \times (0, T) \\ u(x, 0) = f(x) & \text{si } x \in \Omega. \end{cases} \quad (2.6)$$

En este modelo se busca seleccionar una función g de difusividad adecuada, que permita disminuir el ruido y realzar la imagen manteniendo cierta fidelidad con la imagen original f .

2.4. Restauración de dominios perdidos (Inpainting)

Durante el proceso de captura de imágenes, es inevitable que en ocasiones aparezcan cuerpos extraños, errores en la transmisión o defectos de la cámara, entre otros factores, que provoquen la pérdida de información, como se muestra en la Figura 2.3. Esto da origen a regiones $D \subset \Omega$ del dominio de la imagen f , donde no hay información disponible.

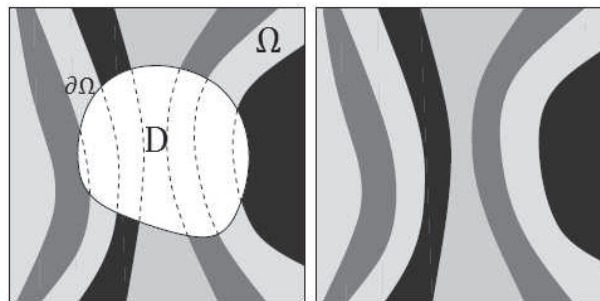


Figura 2.2. Notación general y objetivo del inpainting. **Fuente:** Elaboración propia.

El problema de inpainting consiste en recuperar la información de regiones dañadas o perdidas de la imagen observada, a partir de la información disponible en las regiones alrededor del área a ser recuperada, además de remover objetos que no sean de interés,

como se observa en la Figura 2.3. Varios modelos matemáticos han sido planteados para este problema, entre los cuales se destacan: el modelo de Mumford-Shah para inpainting propuesto por Esedoglu y Shen [16] y el de, Chan y Shen [34] aplicando la *regularización TV*.



Figura 2.3. Ejemplos de aplicaciones de inpainting. **Fuente:** Elaboración propia.

A pesar de su similitud con modelos de procesamiento de imagen habituales, tales como la eliminación de ruido o deblurring, el inpainting es un reto mayor, pues las regiones que faltan son generalmente de gran tamaño comparadas con los dominios con ruido, donde los píxeles contienen tanto el ruido como información de la imagen original; mientras que en problemas de inpainting no existe información significativa dentro del dominio faltante. La información se encuentra alrededor de las áreas a ser tratadas.

Gracias a métodos de procesamiento de imágenes, específicamente el proceso de segmentación, es posible localizar los píxeles defectuosos, permitiendo la recuperación casi total de la información en la imagen. Se debe aclarar que los algoritmos implementados en este documento requieren del conocimiento previo de la localización de la región perdida de la imagen original. No obstante, es posible emplear algoritmos de segmentación para la detección de regiones como paso previo; el resultado dependerá así en gran medida de la precisión de dicho algoritmo.

En esta sección se presentan modelos basados en el enfoque variacional, y en ecuaciones diferenciales parciales que se utilizan para resolver el problema de inpainting.

2.5. Modelos de inpainting basados en métodos variacionales

El método de energía, descrito en el capítulo anterior, se puede emplear para resolver el problema de la reconstrucción de dominios perdidos con una pequeña modificación al segundo término de la ecuación (1.2), pues en el dominio $D \subset \Omega$ no se conoce información acerca de la imagen f .

En términos generales, se formula el problema de minimización del funcional E , determinado por la ecuación:

$$E[u] = R[u] + \|\lambda_D(f - u)\|^2, \quad (2.7)$$

donde λ_D es la función indicatriz de D dada por:

$$\lambda_D(x) = \begin{cases} 0 & \text{si } x \in D \\ \lambda > 0 & \text{si } x \in \Omega \setminus D. \end{cases}$$

Se debe notar que la formulación variacional esta sobre todo el dominio Ω de la imagen, en lugar de plantear sólo el problema de dominio perdido D . La ventaja de este planteamiento es la eliminación simultánea de ruido en toda la imagen y la aproximación será independiente del número y formas de los agujeros de la imagen. En este modelo global, la condición sobre el contorno de D se resume en el término de fidelidad $\|\lambda_D(f - u)\|^2$.

Dependiendo de la elección del término de regularización $R[u]$ se han desarrollado diversos modelos, que pueden dividirse en dos grandes categorías: los geométricos que se concentran en recuperar la parte geométrica de la imagen dentro del dominio perdido y los que se enfocan en identificar la textura para luego completarla. En lo que sigue nos concentraremos en modelos de inpainting geométricos.

2.5.1. Modelo TV para inpainting

Como se explicó previamente, la *regularización de variación total (TV)* fue desarrollada originalmente para el problema de la eliminación de ruido por Rudin, Osher, y Fatemi [33]. Posteriormente, Chan y Shen [34] la aplicaron al problema de inpainting. La regularización TV es una técnica efectiva para resolver el problema restauración de dominios perdidos, capaz de recuperar bordes bajo ciertas condiciones, que luego serán descritas.

El modelo TV inpainting consiste en encontrar $u \in BV(\Omega)$, es decir una función con variación acotada, solución del problema de minimización,

$$\min_{u \in BV(\Omega)} E_{TV}[u|f, D] = \|u\|_{TV(\Omega)} + \frac{\lambda}{2} \int_{\Omega \setminus D} (f(x) - u(x))^2 dx, \quad (2.8)$$

donde λ es un parámetro positivo. Como se puede observar, el problema de minimización (2.8) es idéntico al problema denoising (2.3). Se diferencian en el dominio del segundo término, pues el último considera a todo Ω . Sin embargo, pueden ser resueltos de forma similar considerando la variante $\lambda_D(x)$, tal que

$$\min_{u \in BV(\Omega)} \|u\|_{TV(\Omega)} + \frac{1}{2} \int_{\Omega} \lambda_D(x) (f(x) - u(x))^2 dx, \quad (2.9)$$

donde

$$\lambda_D(x) = \begin{cases} 0 & \text{si } x \in D \\ \lambda > 0 & \text{si } x \in \Omega \setminus D. \end{cases}$$

Notemos que para $x \in D$, el valor de $u(x)$ está influenciado solamente por el término $\|u\|_{TV}$; mientras que, en $\Omega \setminus D$ el problema se reduce a la eliminación de ruido (2.3) utilizando la regularización TV. Al tomar $\lambda \gg 1$ la eliminación de ruido es limitado, por lo que la imagen se mantiene casi sin cambios fuera del dominio perdido, es decir, en $\Omega \setminus D$.

La condición necesaria de optimalidad para encontrar un mínimo de E_{TV} (2.8) bajo ciertos supuestos, esta dada por la ecuación de *Euler-Lagrange*, determinada por:

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda_D(u - f) = 0.$$

2.5.2. Modelo de bordes incrustados (Embedded edge model)

El modelo original de Mumford-Shah [30] emplea la fórmula de la longitud de curva, y está determinado por:

$$E[u|\Gamma] = \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + \alpha \text{length}(\Gamma)$$

donde Γ es el conjunto de los bordes. Este modelo permite considerar los saltos bruscos en las imágenes, lo que le ha permitido alcanzar mucho éxito en la eliminación de ruido y segmentación de imágenes.

Modelo de Mumford-Shah a trozos lisos

La idea de aplicar el modelo de Mumford-Shah a los problemas de inpainting, apareció por primera vez con Tsai, Yezzi, y Willsky [37], y fue estudiada nuevamente por Esedoglu and Shen [16]. El modelo consiste en minimizar el funcional dado por:

$$E_{MS}[u, \Gamma|f, D] = \frac{\gamma}{2} \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + \alpha \text{length}(\Gamma) + \frac{\lambda}{2} \int_{\Omega \setminus D} (u - f)^2 dx \quad (2.10)$$

donde γ, α y λ son parámetros positivos. Es importante notar que la solución de este problema de minimización, no es sólo la imagen u completa y limpia, sino también el conjunto de bordes Γ .

Observe además que con $D = \emptyset$, es decir, si no existen dominios perdidos este modelo corresponde exactamente al modelo clásico de Mumford-Shah para: la eliminación de ruido y la segmentación [30].

2.5.3. Modelos basados en la curva elástica de Euler

La elástica es la curva que representa la deformada por flexión del eje longitudinal de una viga recta, la cual se debe a la aplicación de cargas transversales en el plano xy sobre la viga. En esta sección se presenta la aplicación de la formulación variacional de la elástica al procesamiento de imágenes.

Modelo elástica de inpainting

En [7], Chan, Kang y Shen proponen mejorar el modelo TV inpainting usando la formulación variacional de la elástica, determinada por

$$E[u] = \int_{\Omega} (\alpha + \beta\kappa^2) |\nabla u| dx, \quad \kappa = \nabla \cdot \left[\frac{\nabla u}{|\nabla u|} \right].$$

Así, el modelo de elástica para inpainting consiste en minimizar el funcional dado por:

$$E_e[u|f, D] = \int_{\Omega} \phi(\kappa) |\nabla u| dx + \frac{\lambda}{2} \int_{\Omega \setminus D} (u - f)^2 dx$$

donde,

$$\phi(s) = \alpha + \beta s^2.$$

Modelo de Mumford-Shah-Euler

Al igual que el modelo de TV inpainting, el modelo de Mumford-Shah es insuficiente para problemas de inpainting a gran escala debido a la longitud de curva de energía. Para mejorar esto, Esedoglu y Shen [16], propusieron el esquema inpainting basado en el modelo de Mumford-Shah-Euler. En este modelo, se busca minimizar

$$E_{mse}[u, \Gamma|f, D] = \frac{\gamma}{2} \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + \int_{\Gamma} (\alpha + \beta\kappa^2) ds + \frac{\lambda}{2} \int_{\Omega \setminus D} (u - f)^2 dx,$$

notemos que la longitud de energía de la ecuación (2.10) fue reemplazada por la energía elástica de Euler.

2.6. Modelos basados en ecuaciones diferenciales parciales

Existen un grupo de modelos de inpainting basados en la noción física de difusión, los cuales han sido usados exitosamente para suavizar y recuperar dominios perdidos de imágenes. Sea $D \subset \Omega$ un área pequeña a ser reconstruida y sea ∂D su borde, el proceso de inpainting puede ser visto como un proceso de difusión isotrópica que propaga la información desde ∂D hacia D . Es decir, que esta técnica tiene la intención de difuminar los colores en las áreas faltantes.

En el año 2000, Bertalmio [3] introduce la técnica de inpainting, donde se restaura una imagen automáticamente mediante el uso de ecuaciones diferenciales parciales. El algoritmo es capaz de llenar simultáneamente pequeños agujeros, necesitando solamente la información que se encuentra alrededor de dichos hoyos. El mismo está inspirado en técnicas de restauración de arte. El modelo matemático resultante es una EDP de tercer orden.

El segundo modelo, es el propuesto por Chan y Shen [98], llamado *curvature driven diffusion* (CDD), donde la cantidad de difusión aplicada está basada en la cantidad de curvas

isofotas en ese punto. Las isofotas son líneas dibujadas en regiones de brillo constante, usadas para caracterizar la estructura que poseen objetos extensos en el cielo como galaxias o cometas. Las imágenes en isofotas son similares a los mapas de contorno, los cuales muestran líneas en regiones de elevación constante.

El trabajo de Chan y Shen mejora el algoritmo presentado por Bertalmio, ya que evita que el suavizado en las áreas perdidas se vea borroso.

En esta sección se revisan modelos basados en EDPs que no provienen de formulaciones variacionales.

2.6.1. Enfoque de Bertalmio

Este modelo se basa en la idea intuitiva de transportar la información a lo largo de las líneas de nivel, o isofotas incompletas. Para cada punto x donde $|\nabla u(x)| \neq 0$, se define el vector normal $N(x) = \nabla u(x)/|\nabla u(x)|$ y $T(x)$ el vector ortogonal a $N(x)$ y tangente a la isofota en el punto x como lo muestra la Figura 2.4,

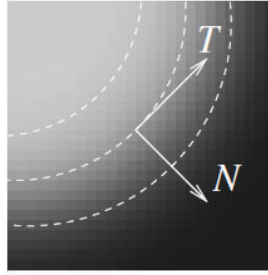


Figura 2.4. Líneas de nivel o isofotas. Fuente: [1]

La idea básica sigue siendo la misma, extender la estructura que rodea el orificio D . Por estructura se entienden a las líneas de nivel que llegan al límite del agujero. En [3], los autores proponen utilizar ecuaciones de transporte, para que la información se propague en la dirección necesaria. La dirección más natural para elegir es la dirección de las isofotas, es decir, la dirección ortogonal al gradiente de la imagen ∇u^\perp , que es igual a $(-\partial_y u, \partial_x u)$. El modelo propuesto modela la propagación del laplaciano en la dirección de las isofotas mediante la resolución de:

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) = \nabla(\Delta u(x, t))\nabla u^\perp(x, t) & \text{con } x \in \Omega \\ u(x, t) = f(x) & \text{con } x \in \partial\Omega \text{ y } t \geq 0 \\ u(x, 0) = f(x). \end{cases} \quad (2.11)$$

Para evitar que de líneas de nivel se crucen, los autores aplican, además, pasos intermedios de difusión anisotrópica, lo que resulta en:

$$\frac{\partial u}{\partial t} = \nabla(\Delta u)\nabla u^\perp + v\nabla \cdot (g(|\nabla u|)\nabla u)$$

donde $g(s)$ define el coeficiente de difusión y v un parámetro mayor que cero. En [2] los autores interpretan la solución de la última EDP como una solución directa de la ecuación de *Navier Stokes* para un fluido incompresible, donde la función de intensidad de la imagen hace el papel de la función de corriente cuyas líneas de nivel son las líneas de corriente del flujo. Se debe notar que, la ventaja de este punto de vista es que los problemas de fluidos estan bien establecidos, tanto analítica como numéricamente.

2.6.2. Modelo de curvatura media

La curvatura de u está determinada por:

$$\text{curv}(u) = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right),$$

que representa la curvatura de las isofotas y es independiente de los valores de gris relativos [12]. El modelo basado en el *movimiento de curvatura media* (mean curvature motion) para inpainting esta definido por:

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \\ u = f \text{ en } \Omega \setminus D, \end{cases} \quad (2.12)$$

2.6.3. Modelo CDD, enfoque de Chan y Shen

Este modelo, fue propuesto por Chan y Shen en [11], después del trabajo realizado sobre el modelo de variación total para inpainting descrito anteriormente (ver(2.9)). Los autores proponen un modelo que involucra a la curvatura κ de las isofotas. El modelo (2.9) consideraba el coeficiente de difusión, $1/|\nabla u|$, que solamente depende del contraste o la fuerza de las isofotas, lo que le impide completar bordes. Es por esto, que los autores llamaron a su modelo de *modelo de curvatura de difusión impulsada* (curvature driven diffusion, CDD). La EDP propuesta es:

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) = \text{div} \left(\frac{g(\kappa) \nabla u}{|\nabla u|} \right) \nabla u^\perp(x, t) \text{ en } D \\ u = f \text{ en } \Omega \setminus D, \end{cases} \quad (2.13)$$

donde g es una función creciente tal que $g(0) = 0$ y $g(+\infty) = +\infty$. Es típico elegir

$$g(s) = s^p \text{ con } p \geq 1.$$

2.6.4. Ecuación modificada de Cahn-Hilliard

La ecuación de Cahn-Hilliard, determinada por

$$\frac{\partial u}{\partial t} = \Delta^2 u - \Delta W'(u) \quad (2.14)$$

con $W(u) = (u^3 - u)$ un potencial no lineal, es muy importante en la ciencia de los materiales[5]. Esta ecuación en derivadas parciales no lineal es una ecuación de difusión de cuarto orden.

Este es un modelo simple para procesos de separación en fases de una aleación binaria a temperatura fija. La función solución u representa la concentración del componente metálico en una aleación de dos metales.

La modificación a la ecuación de Cahn-Hilliard y su aplicación al problema de inpainting, fue propuesta por Bertozzi, Esedoglu y Gillette [34]

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta(\epsilon \Delta u - \frac{1}{\epsilon} W(u)) + \lambda_D(u - f) \\ u = 0 \text{ en } \partial\Omega \end{cases} \quad (2.15)$$

donde λ_D es la función indicatriz del domino perdido D .

El papel de ϵ en la ecuación original de Cahn-Hilliard (2.14) es muy importante, pues sirve como una medida de la región de transición entre dos metales en una aleación, después de alcanzar un estado constante. En el contexto del procesamiento de imágenes, el parámetro ϵ será una medida de la transición entre los dos estados en la escala de grises.

Capítulo 3

Tratamiento numérico

Se han propuesto varios algoritmos numéricos para la minimización de la variación total y esta continúa siendo un área de investigación activa. Como primera propuesta, en 1992 Rudin, Osher y Fatemi [33] proponen resolver el problema de minimización usando el método evolutivo conocido también como *time marching* y un esquema explícito para la aproximación numérica. En lugar de utilizar el método evolutivo, en 2001 Chan, Osher, y Shen [10] proponen resolver el estado estacionario directamente. Los métodos más recientes, emplean la distancia de Bregman [18].

En este trabajo, se plantea resolver la ecuación evolutiva usando un método implícito, donde cada iteración en el tiempo está asociada un sub-problema no lineal y no diferenciable que será resuelto con el método de *Newton semi-suave*.

Este capítulo está estructurado en tres partes; en el primero se presentan aspectos metodológicos, en el segundo bloque el método de discretización y finalmente los algoritmos propuestos para la resolución numérica del problema de inpainting.

3.1. Aspectos metodológicos

En esta sección presentaremos la metodología que se utilizará para encontrar la solución al problema de inpainting utilizando el modelo de variación total (2.9) y el modelo de curvatura media (2.12).

La ecuación de *Euler-Lagrange* asociada al problema de minimizar el funcional E_{TV} en $L^2(\Omega)$ está dada por:

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda_D(u - f) = 0, \quad (3.1)$$

que no es más que una ecuación en derivadas parciales. Para resolver la ecuación (3.1) se puede plantear su versión dinámica, que consiste en reformular el problema tal que, se busca el estado estacionario de la ecuación:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda_D(f - u) \quad (3.2)$$

con $u(x, 0) = f(x)$. Por tanto, pasamos de una ecuación estacionaria a una evolutiva, la cual es una ecuación de **reacción-difusión anisotrópica** [12] en el dominio $\Omega \setminus D$.

Finalmente, se presenta la resolución de

$$\frac{\partial u}{\partial t} = |\nabla u| \left[\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right] \quad (3.3)$$

que corresponde al modelo de curvatura media. Este último también puede ser visto como una técnica de preconditionamiento para (3.2) y cancelar las singularidades debido al coeficiente de difusión degenerada $1/|\nabla u|$ en $\Omega \setminus D$, como lo proponen Marquina y Osher en [29].

Para resolver numéricamente (3.2) y (3.3) se plantea utilizar el esquema de *Euler implícito*, es decir, un *método implícito a un paso*. En este método se obtienen esquemas iterativos de la forma

$$u^{k+1} = u^k + \Delta t \Psi(u^{k+1})$$

donde en cada iteración se debe resolver el sistema de ecuaciones asociado, contrario a lo que sucedía en los métodos explícitos, propuestos para estos problemas hasta ahora, donde u^{k+1} estaba determinado totalmente en la iteración k .

3.2. Métodos a un paso

Los métodos a múltiples pasos tienen asociada la siguiente expresión para k, p enteros positivos

$$\alpha_0 u^k \dots + \alpha_p u^{k+p} = h[\beta_0 \Psi(u^k) \dots \beta_p \Psi(u^{k+p})],$$

donde p determinará el número de pasos del método. Si α_0, α_p y β_0 son distintos de cero el método se conoce como *método a p pasos*. Tal que se busca obtener el valor de u_{k+p} suponiendo conocidos los valores de $u_k, u_{k+1} \dots, u_{k+p-1}$.

Dentro de los métodos a pasos múltiples los métodos de Adams son los más populares y tienen la siguiente forma:

$$u^{k+p} - u^{k+p-1} = h[\beta_0 \Psi(u^k) \dots \beta_p \Psi(u^{k+p})],$$

si $\beta_p = 0$ se conoce como método de *Adams explícito* o método de *Adams-Bashforth*. Y si es distinto de cero, se conoce como método de *Adams implícito* o método de *Adams-Moulton* que en general tienen un error global menor y son más estables, pero computacionalmente más costosos pues se debe resolver una ecuación implícita.

El esquema de Euler explícito está determinado por:

$$u^{k+1} = u^k + h \Psi(u^k),$$

que coincide con el método Adams-Bashforth a un paso. Mientras que el esquema de Euler

implícito,

$$u^{k+1} = u^k + h\Psi(u^{k+1}),$$

coincide con el método Adams-Moulton a un paso.

3.2.1. Propiedades de los métodos a un paso

En esta sección, se definen la *convergencia*, *consistencia* y *estabilidad* de los métodos a un paso, así como las propiedades del esquema de Euler explícito por ser el método numérico que se empleará. Para aproximar $y = y(x)$ solución del problema de valor inicial, con $x \in [a, b]$:

$$\begin{cases} y' = f(x, y) \\ y(a) = \alpha, \end{cases}$$

considere el esquema general de los métodos a un paso:

$$\begin{aligned} y_0 &= \alpha \\ y^{k+1} &= y^k + h\Phi(x^k, y^k, y^{k+1}, f) \quad k = 0, \dots, n-1, \end{aligned}$$

donde $x^k = a + kh$, $h = (b - a)/n$ y Φ identifica al método a un paso, se denomina función de incremento, si depende de la variable y^{k+1} será un esquema implícito y caso contrario será un esquema explícito.

DEFINICIÓN 3.1. (Convergencia) Se dice que un método a un paso es convergente en $[a, b]$, si se verifica que:

$$\lim_{h \rightarrow 0} \max_{0 \leq k \leq n} |y(x^k) - y^k| = 0$$

independiente de la condición inicial $y(a) = \alpha$, donde y es la solución exacta del problema de valor inicial.

DEFINICIÓN 3.2. (Estabilidad) Un método a un paso se dice estable, si existen constantes c_1 y c_2 independientes de h tales que, para cualquier par de sucesiones $\{x_n\}_{n \geq 0}$ y $\{y_n\}_{n \geq 0}$ definidas por:

$$\begin{cases} x_0 = \alpha_1 \\ x^{k+1} = x^k + h\Phi(t^k, x^k, x^{k+1}, f) \end{cases} \quad \begin{cases} y_0 = \alpha_2 \\ y^{k+1} = y^k + h[\Phi(t^k, y^k, y^{k+1}, f) + \epsilon_n] \end{cases}$$

se tiene

$$\max_{0 \leq k \leq n} |x_n - y_n| \leq c_1 |x_0 - y_0| + c_2 \max_{0 \leq k \leq n-1} |\epsilon_n|.$$

Esta noción de estabilidad quiere decir que, una pequeña perturbación en los datos implica una pequeña perturbación en la solución, independientemente del paso h . Note que siempre se introducen perturbaciones debido a los errores de redondeo, lo que hace que esta propiedad sea indispensable para un método numérico.

DEFINICIÓN 3.3. (Error de truncamiento) El error de truncamiento en el paso n es ξ_n deter-

minado por:

$$y(x^{k+1}) = y(x^k) + h\Phi(x^k, y(x^k), y(x^{k+1}), f) + h\zeta_n.$$

DEFINICIÓN 3.4. (Consistencia) El método a un paso se dice consistente si

$$\lim_{h \rightarrow 0} \max_{0 \leq k \leq n-1} |\zeta_n| = 0$$

para toda solución y suficientemente suave de $y' = f(x, y)$.

TEOREMA 3.1. Si el método a un paso es consistente y estable entonces es convergente.

PROPOSICIÓN 3.2. Si $\Phi = \Phi(x, y, f)$ es Lipschitz respecto de la segunda variable, es decir, si existe M tal que $\forall x \in [a, b]$ se verifica que:

$$|\Phi(x, y_1, f) - \Phi(x, y_2, f)| \leq M|y_1 - y_2|.$$

El método a un paso es convergente si, y sólo si, es consistente.

El método de Euler implícito converge con orden 1 respecto a h .

DEFINICIÓN 3.5. (A-estabilidad) Se dice que un método a un paso es A-estable si al aplicarlo al problema de valor inicial test, definido por:

$$\begin{cases} y' = \beta y & \text{Re}(\beta) < 0 \\ y(0) = \mu, \end{cases}$$

la solución numérica $\{y_n\}_{n \geq 0}$ verifica que:

$$\lim_{n \rightarrow \infty} y_n = 0, \quad \forall n > 0.$$

Si se aplica el esquema explícito de Euler, se tiene

$$\begin{aligned} y^{k+1} &= y^k + h\beta y^k \\ &= (1 + \beta h)y^k \\ &\vdots \\ &= (1 + \beta h)^{k+1}y_0, \end{aligned}$$

y^{k+1} tiende a cero si, y solo si,

$$|1 + \beta h| < 1.$$

Los valores βh que verifican esto, son los valores que pertenecen al círculo de centro en $(-1, 0)$ y radio 1, como lo muestra la Figura 3.1. Luego, el esquema explícito de Euler no es A-estable.

Al aplicar el esquema implícito de Euler al problema test, se obtiene

$$\begin{aligned} y^{k+1} &= y^k + h\beta y^{k+1} \\ y^{k+1} &= \frac{1}{1-\beta h} y^k \\ &\vdots \\ &= \left(\frac{1}{1-\beta h} \right)^{k+1} y_0, \end{aligned}$$

luego, la solución numérica tiende a cero si, y solo si, se cumple que:

$$\left| \frac{1}{1-\beta h} \right| < 1,$$

condición que se verifica para cualquier βh con $Re(\beta h) < 0$ y por lo tanto para cualquier valor de $h > 0$. Por lo tanto el esquema implícito de Euler es A-estable.

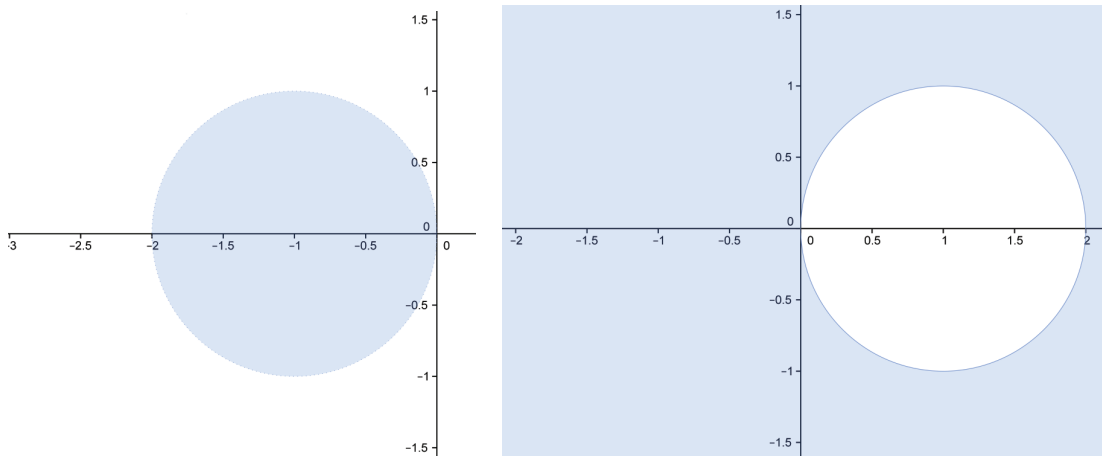


Figura 3.1. Regiones de A-estabilidad para los esquemas explícito e implícito de Euler.

3.3. Regularizaciones

En las ecuaciones (3.1), (3.2) y (3.3) el término,

$$\frac{\nabla u}{|\nabla u|}$$

no está definido si $\nabla u = 0$ y no es diferenciable en cero. Para obtener información de segundo orden de este término, usaremos una regularización.

Las regularizaciones son funciones diferenciables que aproximan términos no diferenciables y dependen en general de un parámetro de regularización. Existen varios tipos, en este trabajo detallaremos la regularización de Huber y la de Berkovier Engelman, que han mostrados resultados satisfactorios en la implementación numérica de diferentes tipos de

problemas como se muestra en [27],[6].

Considere la función $h: \mathbb{R} \rightarrow \mathbb{R}$ tal que:

$$h(w) = \frac{w}{|w|} = \begin{cases} 1, & \text{si } w > 0, \\ -1, & \text{si } w < 0. \end{cases}$$

3.3.1. Regularización Berkovier Engelman

Esta regularización es de tipo global, y consiste en reemplazar

$$|w| \text{ por } |w|_\epsilon$$

donde $|w|_\epsilon = \sqrt{w^2 + \epsilon^2}$ con ϵ un parámetro positivo muy pequeño, que es la regularización de Berkovier Engelman para el valor absoluto. En la Figura 3.2 se observa que, mientras más pequeño sea el parámetro ϵ la regularización en cero del valor absoluto será mejor.

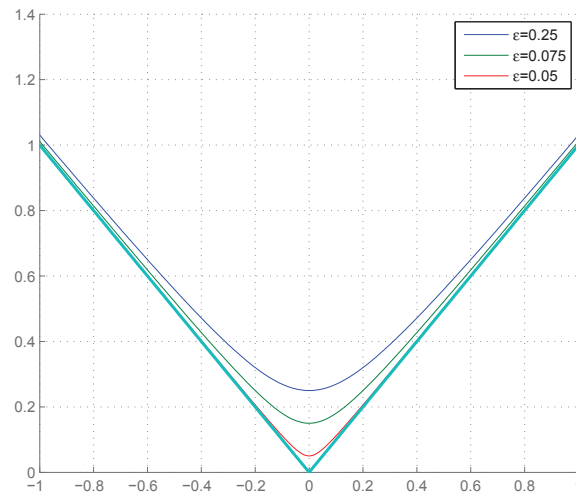


Figura 3.2. Distintos valores para el parámetro de regularización ϵ .

Así se regulariza h con una función $C^\infty(\Omega)$, determinada por:

$$h_\epsilon(w) = \frac{w}{|w|_\epsilon}.$$

Como se observa en la Figura 3.3 mientras más pequeño sea el parámetro ϵ la regularización de h será mejor.

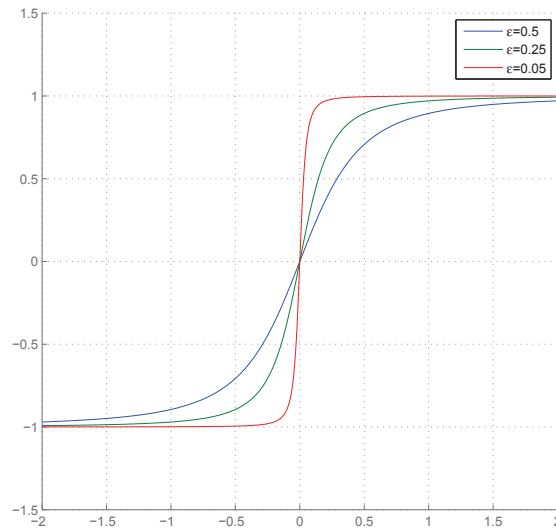


Figura 3.3. Distintos valores para el parámetro de regularización ϵ .

3.3.2. Regularización de Huber

Esta regularización es de tipo local, es decir, que la función h sólo se modificará en una vecindad del punto donde no es diferenciable, en este caso cero.

Sea γ un parámetro positivo suficientemente grande, la regularización de Huber para el valor absoluto se define como:

$$H_{\gamma}(w) = \begin{cases} \gamma \frac{|w|^2}{2} & \text{si } |w| \leq \frac{1}{\gamma}, \\ |w| - \frac{1}{2\gamma} & \text{si } |w| > \frac{1}{\gamma}. \end{cases}$$

Como se observa en la Figura 3.4 mientras más grande sea el parámetro γ la regularización del valor absoluto será mejor.

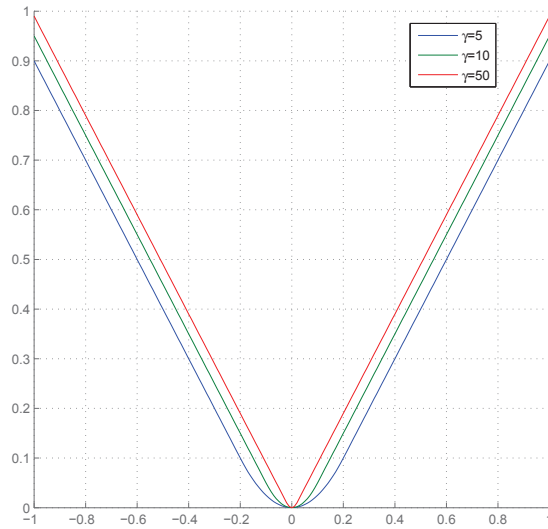


Figura 3.4. Distintos valores para el parámetro de regularización γ .

La primera derivada de la función H_γ con respecto a w está determinada por:

$$H'_\gamma(w) = \frac{\gamma w}{\max(1, \gamma|w|)}$$

y es la regularización que se propone emplear en este trabajo.

Como se observa en la Figura 3.5 mientras más grande sea el parámetro γ la regularización de h será mejor.

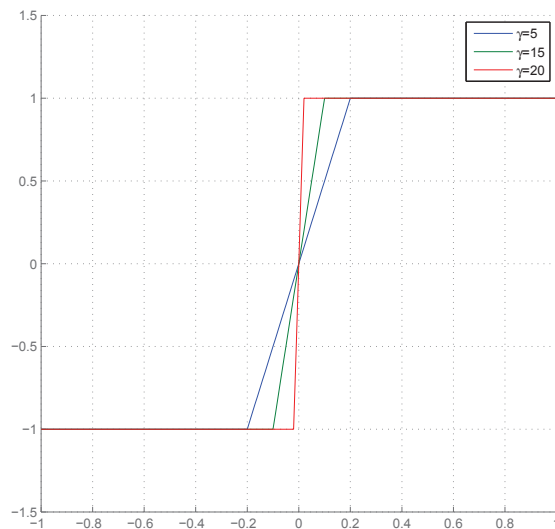


Figura 3.5. Distintos valores para el parámetro de regularización γ .

La solución numérica al problema se aproximará utilizando el método de diferencias finitas, el cual se reduce a resolver un sistema no lineal $L(u) = 0$. Si L es diferenciable se puede utilizar el método iterativo de Newton clásico, pero este no es el caso, pues la presencia de la función $\max(1, \cdot)$ proveniente de la regularización, hace que no se pueda aplicar este método. Para este caso, en [13] se define la *Newton-diferenciabilidad*. A continuación detallaremos estos métodos.

3.4. Método de Newton semi-suave

Considere la ecuación escalar no lineal $f(x) = 0$, el método de Newton es un algoritmo eficiente para encontrar aproximaciones de los ceros o raíces de f . Dado x_0 , la iteración del método de Newton será:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \text{ con } n = 0, 1, \dots$$

Note que este método requiere que f sea una función continuamente diferenciable y $f'(x_n) \neq 0$ para $n = 0, 1, \dots$. El método se puede extender a sistemas de ecuaciones no lineales, del tipo $F(x) = 0$, con:

$$F : \mathbb{R}^n \rightarrow \mathbb{R}^n.$$

La iteración del método de Newton será:

$$\begin{aligned} F'(x_n)\Delta x_n &= -F(x_n) \\ x_{n+1} &= x_n + \Delta x_n \text{ con } n = 0, 1, \dots \end{aligned}$$

No es necesario calcular la inversa de la matriz jacobiana ($[F'(x_n)]^{-1}$) en cada iteración, puesto que se obtiene Δx_n resolviendo el sistema lineal respectivo. De esta manera resolver un sistema de ecuaciones no lineales, se reduce a resolver una sucesión de problemas lineales.

DEFINICIÓN 3.6. (*Newton-diferenciabilidad*) Sean D un subconjunto abierto de un espacio de Banach X y $F: D \rightarrow Z$ un funcional. F es Newton diferenciable en el subconjunto abierto $V \subset D$ si existe $G: V \rightarrow \mathcal{L}(X, Z)$, una derivada generalizada tal, que:

$$\lim_{h \rightarrow 0} \frac{1}{\|h\|_X} \|F(x+h) - F(x) - G(x+h)h\|_Z = 0,$$

para todo $x \in V$.

EJEMPLO 1. Considere la función valor absoluto f :

$$\begin{aligned} f: \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto |x|. \end{aligned}$$

La función f es no diferenciable en 0. Sin embargo, mediante el uso de la derivada generali-

zada

$$g(x) = \begin{cases} 1, & \text{si } x < 0; \\ -1, & \text{si } x \geq 0. \end{cases}$$

para el caso $x = 0$ se obtiene:

- si $h > 0$: $||x + h| - |x| - |h|| = 0$,
- si $h < 0$: $||x + h| - |x| + |h|| = |-x - h - x + h| = 0$.

Así,

$$\lim_{h \rightarrow 0} \frac{1}{|h|} |f(x+h) - f(x) - g(x+h)h| = 0$$

y f es Newton diferenciable.

TEOREMA 3.3. (Iteración de Newton) Sea \tilde{x} una solución de $F(x) = 0$, con F Newton diferenciable en una vecindad abierta de V de \tilde{x} . Si

$$\|G(x)^{-1}\|_{\mathcal{L}(Z,X)} \leq C,$$

para alguna constante $C > 0$ y para todo $x \in V$, entonces la iteración Newton

$$x_{k+1} = -G(x_k)^{-1}F(x_k)$$

converge superlinealmente a \tilde{x} siempre que $\|x_0 - \tilde{x}\|_X$ sea suficientemente pequeño.

PROPOSICIÓN 3.4. La función $\max(1, \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es Newton diferenciable en \mathbb{R}^n , con derivada generalizada:

$$G_{\max} : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$(G_{\max}(x))_i = \begin{cases} 1, & (x)_i > 1, \\ 0, & (x)_i \leq 1. \end{cases}$$

3.5. Aproximación por Diferencias Finitas (FMD)

Para discretizar una ecuación diferencial parcial existen varios enfoques. Entre los más importantes se pueden mencionar la aproximación por diferencias finitas, los elementos finitos, y los métodos espectrales. En este trabajo, se aplicó el método de aproximación por diferencias finitas, que es ampliamente utilizado en el procesamiento matemático de imágenes, debido a la estructura de la imagen digital.

El método de diferencias finitas se utiliza para aproximar la solución de ecuaciones diferenciales. Básicamente consiste en aproximar las derivadas de una función en un punto mediante la combinación de las imágenes de la función en puntos del entorno (puntos de la malla). Dicha combinación se obtiene a partir del desarrollo de Taylor en un punto.

3.5.1. En una dimensión

Por el desarrollo de Taylor sabemos que

$$f(x+h) = f(x) + hf'(x) + \vartheta(h)$$

de donde, la derivada de primer orden de la función f en el punto x se puede aproximar por:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \quad (3.4)$$

que se denomina *diferencia hacia adelante*. De forma similar se obtiene la *diferencia hacia atrás*

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}. \quad (3.5)$$

Sumando (3.4) y (3.5), se obtiene (3.6) *diferencia centrada*:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \quad (3.6)$$

La aproximación de la derivada de segundo orden de f se obtiene tomando $v(x) = f'(x)$,

$$\begin{aligned} f''(x) &= v'(x) \\ &\approx \frac{v(x+h) - v(x)}{h} \\ &\approx \frac{f'(x+h) - f'(x)}{h} \\ &\approx \frac{\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h}}{h} \\ &\approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \end{aligned}$$

Para presentar el método de diferencias finitas, consideremos el problema unidimensional que consiste en resolver una ecuación diferencial de segundo orden con valores en la frontera.

PROBLEMA 1. Hallar $u : [a, b] \rightarrow \mathbb{R}$ tal que verifique:

$$\begin{aligned} u''(x) &= f(x) \quad \text{con } x \in (a, b) \\ u(a) &= u_a, \\ u(b) &= u_b, \end{aligned}$$

donde, tanto f como los valores de frontera, u_a y u_b son conocidos.

Solución: se considera una partición del intervalo $[a, b]$ en $n + 1$ subintervalos, todos ellos de longitud $h = (b - a) / (n + 1)$, que es el tamaño del paso de la discretización:

$$a = x_0 < x_1 < \dots < x_{n+1} = b.$$

Se calcula, en cada uno de los puntos x_i , un valor aproximado de $u(x_i)$, al que se notará como u_i . Se conocen los valores en la frontera, es decir, u_0 y u_{n+1} , así nuestras incógnitas serán: $u_1, u_2 \dots u_n$. Utilizando la aproximación de la derivada de segundo orden en $x_i = a + ih$, con $i = 0, \dots, n + 1$ se obtiene el problema discretizado, determinado por:

$$\begin{aligned} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} &= f_i, \quad \forall i \leq n \\ u_0 &= u_a, \\ u_{n+1} &= u_b. \end{aligned}$$

Desarrollando para $n=3$, se obtiene:

$$\begin{aligned} \frac{u_2 - 2u_1 + u_0}{h^2} &= f_1 \text{ para } i = 1, \\ \frac{u_3 - 2u_2 + u_1}{h^2} &= f_2 \text{ para } i = 2, \\ \frac{u_4 - 2u_3 + u_2}{h^2} &= f_3 \text{ para } i = 3. \end{aligned}$$

En general, despejando y expresando en forma matricial se busca resolver el siguiente sistema lineal

$$Au = f,$$

donde $A \in \mathbb{R}^{n \times n}$ es de la forma:

$$A = -\frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix},$$

y los vectores $u \in \mathbb{R}^n$ y $f \in \mathbb{R}^n$

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} \quad f = \begin{pmatrix} f_1 + u_a/h^2 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n + u_b/h^2 \end{pmatrix}.$$

La matriz A es tridiagonal, simétrica y definida positiva, lo que garantiza que el sistema lineal tenga solución única.

Considere un problema más general, donde los coeficientes de u' y u son funciones de x ,

$$\begin{aligned} -u'' + p(x)u' + q(x)u &= f(x) \text{ con } x \in [a, b] \\ u(a) &= u_0, \\ u(b) &= u_{n+1}. \end{aligned}$$

Solución: el problema discretizado queda determinado por:

$$\begin{aligned} -\frac{1}{h^2}(u_{i+1} - 2u_i + u_{i-1}) + p(x_i)\frac{1}{h}(u_i - u_{i-1}) + q(x_i)u_i &= f_i, \quad \forall i \leq n \\ u_0 &= u_a, \\ u_{n+1} &= u_b. \end{aligned}$$

El desarrollo para $n = 3$ será:

- $i = 1$

$$\frac{1}{h^2}(-u_2 + 2u_1 - u_0) + \frac{p_1}{h}(u_1 - u_0) + q_1u_1 = f_1,$$
- $i = 2$

$$\frac{1}{h^2}(-u_3 + 2u_2 - u_1) + \frac{p_2}{h}(u_2 - u_1) + q_2u_2 = f_2,$$
- $i = 3$

$$\frac{1}{h^2}(-u_4 + 2u_3 - u_2) + \frac{p_3}{h}(u_3 - u_2) + q_3u_3 = f_3.$$

Despejando y expresando en forma matricial se tiene el siguiente sistema lineal

$$\begin{aligned} \left\{ \frac{1}{h^2} \begin{pmatrix} 2 + h^2q_1 & -1 & 0 \\ -1 & 2 + h^2q_2 & -1 \\ 0 & -1 & 2 + h^2q_3 \end{pmatrix} + \frac{1}{h} \begin{pmatrix} p_1 & 0 & 0 \\ -p_2 & p_2 & 0 \\ 0 & -p_3 & p_3 \end{pmatrix} \right\} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \\ = \begin{pmatrix} f_1 + u_0/h^2 + p_1u_0/h \\ f_2 \\ f_3 + u_4/h^2 \end{pmatrix}. \end{aligned}$$

Cuando en una ecuación diferencial se especifican condiciones sobre la frontera del dominio, se pueden destacar dos tipos de condiciones:

- **condiciones de Dirichlet:** se especifican los valores que la solución debe satisfacer en la frontera del dominio, y son de la forma

$$y(x) = f(x), \quad \forall x \in \partial\Omega$$

donde f es una función conocida sobre $\partial\Omega$. Si $f \equiv 0$, se conocen como **condiciones homogéneas de Dirichlet**.

EJEMPLO 2. Consideremos un problema clásico de valor en la frontera del intervalo

$\Omega = [0, 1]$, descrito por la *Ecuación de Poisson* en una dimensión,

$$-\frac{\partial^2 u}{\partial x^2} = f, \text{ en } \Omega$$

$$u(0) = u(1) = 0.$$

Solución: aproximamos la derivada de segundo orden para valores de u en la frontera. Se discretiza el problema al igual que el *problema 1* tomando $u_a = u_b = 0$

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f_i, \quad \forall i \leq n$$

$$u_0 = u_{n+1} = 0.$$

- **Condiciones de Neumann:** se especifican los valores, que la derivada de la solución debe satisfacer en la frontera del dominio y son de la forma:

$$\frac{\partial y}{\partial \vec{n}}(x) = f(x), \quad \forall x \in \partial\Omega$$

donde \vec{n} es el vector normal a la frontera $\partial\Omega$ y f es una función escalar conocida sobre $\partial\Omega$.

EJEMPLO 3. En una dimensión, supongamos que ahora tenemos

$$\frac{\partial^2 u}{\partial x^2} = f, \text{ en } \Omega = [a, b]$$

$$\frac{\partial u}{\partial x}(a) = du_a,$$

$$u(b) = u_b.$$

Solución: la única diferencia con respecto al caso anterior es que, desconocemos el valor de u_0 pero conocemos el valor de su primera derivada. Considerando diferencias centradas para la discretización de la primera derivada, se verifica que:

$$u'(a) = \frac{u_1 - u_{-1}}{2h} = du_a = du_0,$$

despejando,

$$u_{-1} = u_1 - 2hdu_0,$$

como tenemos una incógnita mas por determinar, añadimos una nueva ecuación para $i = 0$; reemplazando u_{-1} en la discretización de la segunda derivada se obtiene:

$$\frac{u_1 - 2u_0 + u_1 - 2hdu_0}{h^2} = \frac{2u_1 - 2u_0 - 2hdu_0}{h^2} = f_0.$$

Por lo tanto, en forma matricial tenemos $A \in \mathbb{R}^{n \times n}$ de la forma

$$A = -\frac{1}{h^2} \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix},$$

$$u = (u_0, u_1, \dots, u_{n-1})^T \text{ y}$$

$$f = \begin{pmatrix} (f_0 + (2du_a/h))/2 \\ f_1 \\ \vdots \\ f_{n-1} - u_b/h^2 \end{pmatrix}.$$

En el caso del otro extremo, es decir, $\frac{\partial u}{\partial x}(b) = du_b$ se procede de forma similar y se obtiene:

$$A = -\frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix},$$

$$u = (u_1, u_2, \dots, u_n)^T \text{ y}$$

$$f = \begin{pmatrix} f_1 - u_0/h^2 \\ \vdots \\ f_{n-1} \\ (f_n + (2du_b/h))/2 \end{pmatrix}.$$

PROBLEMA 2. Hallar una función $u = u(x, t)$ con $x \in [0, 1]$ y $t > 0$, solución de la ecuación:

$$\frac{\partial u}{\partial t} - Lu = f, \quad x \in (0, 1), \quad t > 0 \quad (3.7)$$

con condiciones de borde

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

y condición inicial

$$u(x, 0) = u_0(x), \quad x \in [0, 1].$$

Si el operador diferencial L está dado por:

$$Lu = v \frac{\partial^2 u}{\partial t^2},$$

la ecuación (3.7) se denomina ecuación del calor. Considere una barra metálica de longitud

unitaria, $u(x, t)$ describe la temperatura de la barra en el punto x en el instante t , el coeficiente de conductividad térmica está dado por $v > 0$ y $f(x, t)$ representa la fuente calor.

Solución: para resolver numéricamente la ecuación de calor se discretiza tanto la variable espacial como la temporal. Se utiliza la siguiente notación

$$\dot{u}_i = u(x_i, t), \quad i = 0, \dots, n,$$

y se aproxima el problema con:

$$\begin{aligned} \dot{u}_i(t) + v \frac{1}{h^2} (u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)) &= f_i(t), \quad i = 1, \dots, n-1, \\ u_0(t) = u_n(t) &= 0, \quad \forall t > 0, \\ u_i(0) &= u_0(x_i), \quad i = 0, \dots, n. \end{aligned}$$

Matricialmente este esquema corresponde a un sistema de ecuaciones diferenciales ordinarias (EDO) de la forma:

$$\begin{cases} \dot{u}_i(t) = vAu(t) + f(t), & \forall t > 0 \\ u(0) = u_0 \end{cases} \quad (3.8)$$

donde $u(t) = (u_1(t), \dots, u_{n-1}(t))$, $u_0 = (u_0(x_1), \dots, u_0(x_{n-1}))$ y A es la matriz de discretización.

Los esquemas más utilizados para resolver (3.8) son los de Euler semi-implícitos, donde v^k será el valor de v en el instante $t = k\Delta t$ para $\Delta t > 0$, el esquema semi-implícito con parámetro θ es de la siguiente forma:

$$\begin{cases} \frac{u^{k+1} - u^k}{\Delta t} = vA(\theta u^{k+1} + (1 - \theta)u^k) + \theta f^{k+1} + (1 - \theta)f^k, & k = 0, 1, \dots \\ u^0 = u_0, \end{cases}$$

o, de manera equivalente,

$$(I + v\theta\Delta t A)u^{k+1} = (I - v\Delta t(1 - \theta)A)u^k + g^{k+1}$$

con $g^{k+1} = \Delta t(\theta f^{k+1} + (1 - \theta)f^k)$.

- Si $\theta = 0$, se tiene el método de Euler explícito:

$$u^{k+1} = (I + v\Delta t A)u^k + \Delta t f^k, \quad k = 1, 2, \dots$$

si $\Delta t < \frac{1}{2v}h^2$ el método es estable.

- Si $\theta = 1$, se tiene el método de Euler implícito:

$$u^{k+1} = (I + v\Delta t A)^{-1}u^k + \Delta t f^{k+1}, \quad k = 1, 2, \dots$$

el cual es incondicionalmente estable.

- Si $\theta = 1/2$ se obtiene el método de Crank-Nicolson, también es incondicionalmente estable.

3.5.2. En dos dimensiones

Para el caso bidimensional, el procedimiento es idéntico al empleado al aproximar problemas unidimensionales.

PROBLEMA 3. Sea $\Omega = (a_x, b_x) \times (a_y, b_y) \subset \mathbb{R}^2$, un dominio rectangular. Hallar $u : \Omega \rightarrow \mathbb{R}$ solución de la Ecuación de Poisson,

$$-\Delta u = -\frac{\partial^2 u}{\partial^2 x} - \frac{\partial^2 u}{\partial^2 y} = f \text{ en } \Omega$$

$$u = g \text{ sobre } \partial\Omega$$

donde $\partial\Omega$ es la frontera de Ω y tanto f como g son funciones conocidas.

Solución: Se define el mallado sobre el rectángulo Ω , a través de la partición sobre $[a_x, b_x]$ con un paso de discretización $h_x = (b_x - a_x)/(n_x + 1)$ y de forma similar sobre $[a_y, b_y]$ con un paso de discretización $h_y = (b_y - a_y)/(n_y + 1)$. Estas particiones definen los puntos $(x_i, y_j) \in \mathbb{R}^2$, tal que

$$x_i = a_x + ih_x, \text{ con } i = 0, 1, \dots, n_x + 1$$

$$y_j = a_y + jh_y, \text{ con } j = 0, 1, \dots, n_y + 1.$$

En cada uno de los puntos (x_i, y_j) de la malla, se calcula una aproximación del valor $u(x_i, y_j)$, al que notaremos como u_{ij} . De forma similar que en el caso en una dimensión, podemos aproximar cada una de las derivadas parciales usando el desarrollo de Taylor. La aproximación del laplaciano estará dada por:

$$\frac{\partial^2 u}{\partial^2 x}(x, y) + \frac{\partial^2 u}{\partial^2 y}(x, y) \approx \frac{u(x - h_x, y) - 2u(x, y) + u(x + h_x, y)}{h_x^2} + \frac{u(x, y - h_y) - 2u(x, y) + u(x, y + h_y)}{h_y^2}.$$

Utilizando esta fórmula en los puntos (x_i, y_j) de la malla (ver Figura 3.6), obtenemos el problema discretizado:

$$\begin{cases} -\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2} - \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h_y^2} = f_{i,j} \\ u_{i,j} = g_{i,j} \end{cases} \quad (3.9)$$

donde $f_{i,j} = f(x_i, y_j)$, con $i = 1, \dots, n_x$ y $j = 1, \dots, n_y$.

donde I es la matriz identidad $n \times n$ y la matriz B tiene la forma

$$B = \begin{pmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -1 & 4 & -1 \\ 0 & \dots & 0 & -1 & 4 \end{pmatrix}.$$

Notemos que la matriz A es tridiagonal por bloques, se puede probar que es definida positiva, y por tanto invertible lo que garantiza que el sistema tenga solución única. Los vectores $u \in \mathbb{R}^{n^2}$ y $f \in \mathbb{R}^{n^2}$, son de la siguiente forma:

$$u = (u_{1,1}, \dots, u_{n,1}, u_{1,2}, \dots, u_{n,2}, \dots, u_{n,n})^T$$

$$f = (f_{1,1}, \dots, f_{n,1}, f_{1,2}, \dots, f_{n,2}, \dots, f_{n,n})^T$$

Si se consideran condiciones de Neumann, la matriz A será:

$$A = \frac{1}{h^2} \begin{pmatrix} -B_1 & I & 0 & \dots & 0 \\ I & -B_2 & I & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & I & -B_2 & I \\ 0 & \dots & 0 & I & -B_1 \end{pmatrix},$$

donde I es la matriz identidad $n \times n$, B_1 y B_2 tienen la forma:

$$B_1 = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 3 & -1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -1 & 3 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 3 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -1 & 4 & -1 \\ 0 & \dots & 0 & -1 & 3 \end{pmatrix}.$$

A continuación, se presenta las matrices asociadas a la discretización de las derivadas parciales de u

$$\frac{\partial u}{\partial x} \approx D_x^+ \text{ o } (D_x^-) \quad \frac{\partial u}{\partial y} \approx D_y^+ \text{ o } (D_y^-)$$

donde

- D_x^+ : diferencias en x hacia adelante,
- D_x^- : diferencias en x hacia atrás,
- D_y^+ : diferencias en y hacia adelante,
- D_y^- : diferencias en y hacia atrás

las mismas que son matrices $n^2 \times n^2$, definidas por:

$$D_x^+ = \frac{1}{h} \begin{pmatrix} D_1 & 0 & 0 & \dots & 0 \\ 0 & D_1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & D_1 & 0 \\ 0 & \dots & 0 & 0 & D_1 \end{pmatrix}, \quad D_y^+ = \frac{1}{h} \begin{pmatrix} -I & I & 0 & \dots & 0 \\ 0 & -I & I & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -I & I \\ 0 & \dots & 0 & 0 & -I \end{pmatrix}$$

$$D_x^- = \frac{1}{h} \begin{pmatrix} D_2 & 0 & 0 & \dots & 0 \\ 0 & D_2 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & D_2 & 0 \\ 0 & \dots & 0 & 0 & D_2 \end{pmatrix}, \quad D_y^- = \frac{1}{h} \begin{pmatrix} I & 0 & 0 & \dots & 0 \\ -I & I & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -I & I & 0 \\ 0 & \dots & 0 & -I & I \end{pmatrix}$$

con I la matriz identidad $n \times n$, D_1 y D_2 matrices de dimensión $n \times n$ de la forma:

$$D_1 = \frac{1}{h} \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -1 & 1 \\ 0 & \dots & 0 & 0 & -1 \end{pmatrix}, \quad D_2 = \frac{1}{h} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & -1 & 1 & 0 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix}.$$

3.6. Aplicaciones

Después de establecer el método de discretización que se empleará para resolver el problema de inpainting, se presenta a continuación la resolución en MATLAB de una serie de problemas que nos permitirán alcanzar nuestro objetivo.

3.6.1. Problemas lineales

A continuación se presenta la discretización de ecuaciones en derivadas parciales lineales en una y dos dimensiones.

PROBLEMA 4. Hallar $y : (0, 10) \rightarrow \mathbb{R}$, que satisface la ecuación (3.10),

$$y'' + \frac{1}{10}y' + \frac{101}{100}y = \frac{5}{10}\cos\left(\frac{x}{4}\right) \quad (3.10)$$

sujeto a las condiciones de borde,

$$y(0) = 0, \quad y(10) = -1/2.$$

Solución: El problema discretizado, utilizando el método de aproximación por diferencias finitas hacia adelante con condiciones de Dirichlet, queda determinado por el siguiente sistema lineal:

$$Au + \frac{1}{10}Du + \frac{101}{100}u = f$$

donde: A es la matriz de discretización para la derivada de segundo orden, D es la matriz de discretización para la derivada de primer orden, $f \in \mathbb{R}^n$ determinado por:

$$f(i)_{i \leq n} = \left(\frac{5}{10} \cos \left(\frac{x_i}{4} \right) \right)$$

y $u = (u_1, u_2, \dots, u_n)^T$. Conocemos la solución exacta de la ecuación (3.10), que se obtuvo utilizando el comando *dsolve* del software matemático Matlab,

`y=dsolve(' -D2y-1/10*Dy-101/100*y=-5/10*cos(t/4) ', 'y(0)=0, y(10)=-1/2', 't')`.

La Figura 3.7 muestra la solución aproximada utilizando 100 y 700 puntos de discretización. Como se observa, mientras más pequeño sea el paso de discretización mejor será la aproximación.

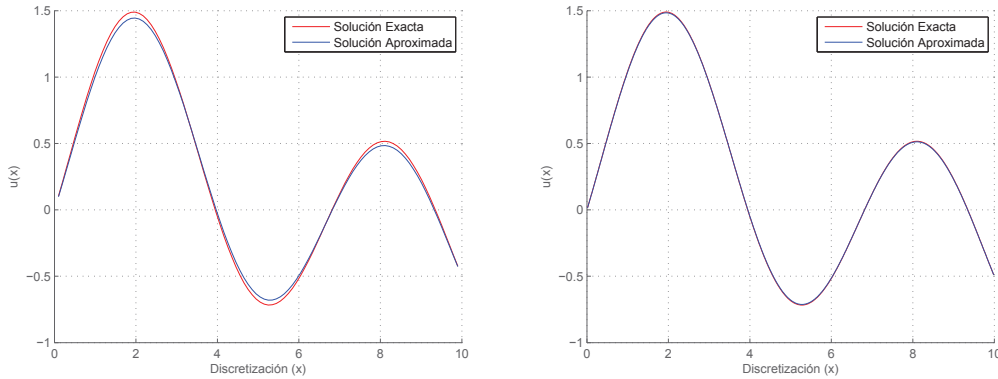


Figura 3.7. Comparación entre la solución exacta y la solución aproximada encontrada por el algoritmo implementado con $n = 100$ (a la izquierda) y $n = 700$ (a la derecha).

PROBLEMA 5. Resolver el problema elíptico en $\Omega = (0, 1) \times (0, 1)$:

$$\begin{cases} \Delta u = -10x^2 & (x, y) \in \Omega \\ u(x, y) = 0 & \forall (x, y) \in \partial\Omega. \end{cases} \quad (3.11)$$

Solución: El problema discretizado, utilizando el método de aproximación por diferencias finitas hacia adelante con condiciones homogéneas de Dirichlet en dos dimensiones, queda determinado por el siguiente sistema lineal:

$$Au = f,$$

donde A es la matriz de discretización del laplaciano y $f \in \mathbb{R}^{n^2}$ tal que:

$$f(i)_{i \leq n} = -10x_i^2.$$

La Figura 3.8 muestra la solución aproximada de la ecuación (3.11), utilizando 50 puntos de discretización.

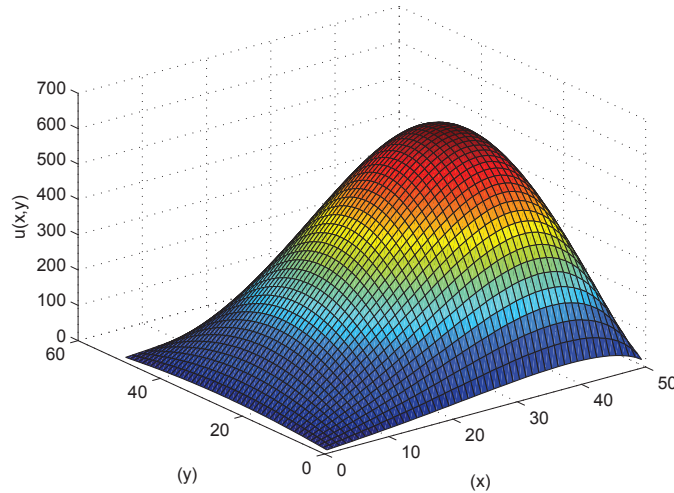


Figura 3.8. Superficie de u aproximación de la solución (3.11).

3.6.2. Problemas no lineales

A continuación se buscará aproximar la solución numérica de ecuaciones en derivadas parciales no lineales utilizando el método de Newton.

PROBLEMA 6. Resolver el siguiente problema elíptico no lineal en $\Omega = (0,1) \times (0,1)$:

$$\begin{cases} \epsilon \Delta u - u^3 = f \\ u(x,y) = 0 \quad \forall (x,y) \in \partial\Omega \end{cases} \quad (3.12)$$

Solución: el problema (3.12) discretizado está determinado por:

$$\epsilon \frac{1}{h^2} (u_{i+1,j} - 4u_{i,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) - u_{i,j}^3 = f_i, \quad i, j = 1, \dots, n.$$

En forma matricial con $u^3 = (u_1^3, \dots, u_n^3)^T$ y A matriz de discretización del laplaciano, se obtiene:

$$\epsilon Au - u^3 = f,$$

nótese que el sistema que se obtiene después de la discretización es un sistema no lineal por

lo que se requiere del método de Newton para resolverlo. Sea

$$F(u) = [\epsilon Au - u^3 - f],$$

$$F'(u) = [\epsilon A - 3\text{diag}(u^2)],$$

de donde el esquema iterativo será:

$$F'(u_n)\Delta u_n = -F(u_n) \quad n = 0, 1, \dots$$

$$u_{n+1} = u_n + \Delta u_n.$$

Se implementó este algoritmo utilizando una tolerancia de 10^{-6} y $\|\Delta u_n\| < 10^{-6}$ como condición de parada del algoritmo. De este modo, se obtiene una aproximación a la solución de (3.12), con $\epsilon = 10^{-2}$, $f = 1$ y $n = 50$ puntos de discretización, como se observa en la Figura 3.9.

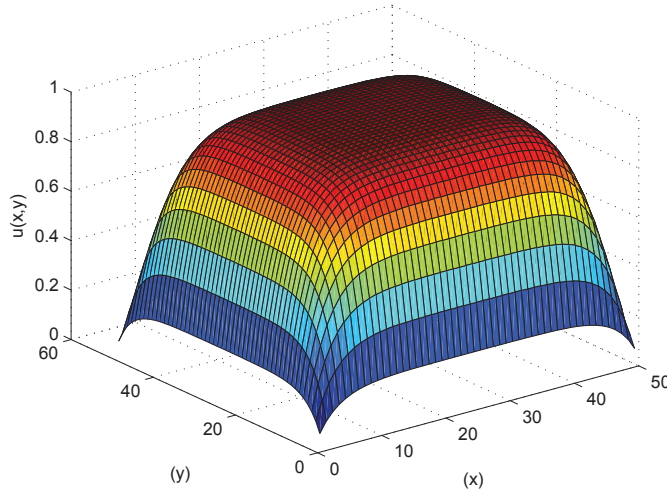


Figura 3.9. Superficie de u aproximación de la solución (3.12), tomando $\epsilon = 10^{-2}$, $f = 1$.

PROBLEMA 7. Dada una función $f: (0,1) \rightarrow \mathbb{R}$ con ruido, se busca una función que mantenga características como saltos, y que no contenga ruido. Ya que este problema puede ser tratado como una imagen en una dimensión se resolverá utilizando el modelo de variación total de filtrado de ruido (2.3).

Solución: A partir de (2.4), el problema se reduce a resolver:

$$-\gamma u'' - \left(\frac{u'}{|u'|} \right)' + \lambda(u - f) = 0, \quad (3.13)$$

sujeto a la condición inicial $u(0) = u_0$ y $u(1) = u_{n+1}$.

Notemos que este problema contiene un término no diferenciable. A continuación, se presenta la solución numérica del problema (3.13) utilizando la regularización de Berkovier

Engelman antes mencionada.

Se define la variable q tal que se obtiene el siguiente sistema

$$\begin{cases} -\gamma u'' - q' + \lambda(u - f) = 0, \\ |u'|q = u', \\ u(0) = u_0, \quad u(1) = u_{n+1}. \end{cases} \quad (3.14)$$

El problema (3.14) discretizado está determinado por:

$$\begin{cases} -\gamma \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{q_i - q_{i-1}}{h} + \lambda(u_i - f_i) = 0, \quad \forall i \leq n \\ \sqrt{\left(\frac{u_{i+1} - u_i}{h}\right)^2 + \epsilon^2} q_i = \frac{u_{i+1} - u_i}{h} \\ u(0) = u_0, \quad u(1) = u_{n+1}. \end{cases}$$

matricialmente se reescribe:

$$F(u, q) = \begin{bmatrix} -\gamma Au - D^- q + \lambda(u - f) \\ \left(\sqrt{(D^+ u)^2 + \epsilon^2}\right) q - (D^+ u) \end{bmatrix} \quad (3.15)$$

donde A es la matriz de discretización de la derivada de segundo orden, D^- la matriz de diferencias hacia atrás y D^+ , la matriz de diferencias hacia adelante. La matriz jacobiana asociada a F , está dada por:

$$F'(u, q) = \begin{bmatrix} -\gamma A + \lambda I & -D^- q \\ B - D^+ & \text{diag}\left(\sqrt{(D^+ u)^2 + \epsilon^2}\right) \end{bmatrix}$$

donde,

$$B = \text{diag}(b) D^+ \text{diag}(q)$$

$$b = \frac{D^+ u}{\sqrt{(D^+ u)^2 + \epsilon^2}}.$$

El esquema iterativo está dado por:

$$F'(u_k, q_k) \Delta(u_k, q_k) = -F(u_k, q_k) \quad k = 0, 1, \dots$$

$$(u_{k+1}, q_{k+1}) = (u_k, q_k) + \Delta(u_k, q_k).$$

Se implementó el algoritmo con $n = 1000$, $\gamma = 10^{-6}$, $\epsilon = 10^{-2}$, $\lambda = 10^3$ y tolerancia de 10^{-6} para el criterio de parada $\|\Delta(u_k, q_k)\| < \text{tolerancia}$. El problema resulta inestable para valores pequeños de ϵ , por tal motivo, se proyecta q al intervalo $(-1, 1)$ en cada iteración del método de Newton; es decir, se utiliza \hat{q} en lugar de q :

$$\hat{q} = \text{mín}(\text{máx}(-1, q), 1).$$

La Figura 3.10 muestra el resultado para $f = \exp(-x) + \eta$ con η un número aleatorio con distribución normal de media cero y varianza 0,1

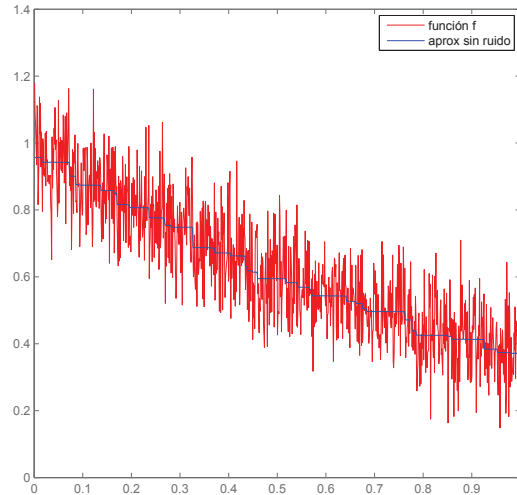


Figura 3.10. Aproximación de la función f

OBSERVACIÓN. En (3.14) se pueden utilizar en lugar de la matriz de diferencias hacia atrás y la matriz de diferencias hacia adelante, D^c la matriz de diferencias centradas obteniendo el mismo resultado. Sin embargo, no se pueden utilizar sólo diferencias hacia adelante o sólo hacia atrás, pues se estarían tomando como conocidos valores distintos a las condiciones de frontera.

3.7. Algoritmos propuestos para el procesamiento de imágenes

3.7.1. Filtrado de ruido

PROBLEMA 8. Dada una imagen f con ruido, se pretende encontrar una imagen limpia de ruido u , que conserve las características de la imagen observada.



Solución: se resolverá este problema de filtrado de ruido utilizando el modelo de variación total,

$$\min_u \int_{\Omega} |\nabla u(x)| dx + \frac{\lambda}{2} \int_{\Omega} (f(x) - u(x))^2 dx,$$

el cual se reduce a resolver numéricamente la condición de optimalidad asociada, es decir, resolver la ecuación de Euler-Lagrange determinada por:

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda(u - f) = 0 \quad (3.16)$$

con condiciones homogéneas de Neumann $\partial_n u = 0$, donde n es el vector normal exterior a $\partial\Omega$. El problema regularizado utilizando la regularización de Berkovier Engelman será:

$$\begin{cases} -\operatorname{div} \vec{\phi} + \lambda(u - f) = 0, \\ \vec{\phi} = \left(\frac{\nabla u}{|\nabla u|_\epsilon} \right), \\ \partial_n u = 0 \text{ en } \partial\Omega. \end{cases} \quad (3.17)$$

Note que este problema es la extensión al caso bidimensional del problema anterior (3.14), así la implementación numérica es similar considerando las siguientes variantes:

$\vec{\phi}: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ un campo vectorial tal que

$$\vec{\phi} = (\phi_1, \phi_2),$$

de donde

$$\nabla \cdot \vec{\phi} = \operatorname{div} \vec{\phi} = \frac{\partial \phi_1}{\partial x} + \frac{\partial \phi_2}{\partial y}.$$

Así,

$$\left(\sqrt{(\partial_x u)^2 + (\partial_y u)^2 + \epsilon^2} \right) \phi_1 = \partial_x u, \quad \left(\sqrt{(\partial_x u)^2 + (\partial_y u)^2 + \epsilon^2} \right) \phi_2 = \partial_y u.$$

El problema discretizado en forma matricial tiene asociado F tal que

$$F(u, \phi_1, \phi_2) = \begin{pmatrix} -D_x^- \phi_1 - D_y^- \phi_2 + \lambda(u - f) \\ \left(\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2} \right) \phi_1 - D_x^+ u \\ \left(\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2} \right) \phi_2 - D_y^+ u \end{pmatrix},$$

y su matriz jacobiana asociada F'

$$F'(u, \phi_1, \phi_2) = \begin{pmatrix} \lambda I & -D_x^- & -D_y^- \\ B_1 - D_x^+ & \operatorname{diag}(R) & 0 \\ B_1 - D_y^+ & 0 & \operatorname{diag}(R) \end{pmatrix}$$

donde $R = \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}$,

$$B_1 = [\operatorname{diag}(b_1)D_x^+ + \operatorname{diag}(b_2)D_y^+] \operatorname{diag}(\phi_1), \quad B_2 = [\operatorname{diag}(b_1)D_x^+ + \operatorname{diag}(b_2)D_y^+] \operatorname{diag}(\phi_2)$$

con

$$b_1 = \frac{D_x^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}}, \quad b_2 = \frac{D_y^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}}.$$

El esquema iterativo asociado está dado por:

$$\begin{aligned} F'(u_k, \phi_k^1, \phi_k^2) \Delta(u_k, \phi_k^1, \phi_k^2) &= -F(u_k, \phi_k^1, \phi_k^2) \quad k = 0, 1, \dots \\ (u_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) &= (u_k, \phi_k^1, \phi_k^2) + \Delta(u_k, \phi_k^1, \phi_k^2). \end{aligned}$$

Se implementó el algoritmo con $n = 256$, $\epsilon = 10^{-4}$, $\lambda = 3000$ y tolerancia de 10^{-6} para el criterio de parada dado por $\|\Delta(u_k, \phi_k^1, \phi_k^2)\| < \text{tolerancia}$. El problema resulta inestable para valores pequeños de ϵ , por tal motivo, se proyectan ϕ_1 y ϕ_2 en cada iteración del método de Newton, es decir, se utiliza el vector $\hat{\phi} = (\hat{\phi}_1, \hat{\phi}_2)$ en lugar de $\vec{\phi}$:

$$\hat{\phi}_i = \frac{\phi_i}{\max(1, \|\vec{\phi}\|)} = \begin{bmatrix} \phi_i^1 / \max(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}) \\ \vdots \\ \phi_i^n / \max(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}) \end{bmatrix}.$$

Así,

$$\hat{B}_i = [\text{diag}(b_1)D_x^+ + \text{diag}(b_2)D_y^+] \text{diag}(\hat{\phi}_i) \quad \text{para } i = 1, 2.$$

Se observa, el resultado obtenido después de 22 iteraciones en la Figura 3.11,



Figura 3.11. Imagen filtrada.

En la Figura 3.12, se observa que el filtrado de ruido depende del valor que tome el parámetro λ . Para valores pequeños de este parámetro se obtiene una solución sobre-regularizada, donde el suavizado en la imagen filtrada es excesivo. Por el contrario, si λ es demasiado grande el algoritmo no filtra el ruido.



Figura 3.12. Imagen filtrada tomando $\lambda = 500, 1000$ y 5000 .

3.7.2. Inpainting

PROBLEMA 9. Dada una imagen f con dominio perdido D , se busca una función u que será la imagen restaurada de f pues aproxima los dominios perdidos D con la información disponible alrededor de estos.

Algoritmo para inpainting utilizando la regularización de Huber.

Solución: se resolverá este problema de inpainting utilizando el modelo de variación total

$$\min_u E_{TV}[u|f, D] = \int_{\Omega} |\nabla u| dx + \frac{1}{2} \int_{\Omega} \lambda_D(x) (f(x) - u(x))^2 dx,$$

el cual se reduce a resolver numéricamente la condición de optimalidad asociada $E'_{TV} = 0$, es decir, resolver la ecuación de Euler-Lagrange determinada por:

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda_D(x)(u - f) = 0. \quad (3.18)$$

Note que (3.18) es similar al problema de filtrado ruido, por lo cual se procede de manera similar. El problema regularizado utilizando la primera derivada de Huber se presenta a

continuación:

$$\begin{cases} -\operatorname{div} \vec{\phi} + \lambda_D(x)(u - f) = 0 \\ \max(1, \gamma |\nabla u|) \vec{\phi} = \gamma \nabla u. \end{cases}$$

La discretización y formulación matricial asociada está determinada por F donde:

$$F(u, \phi_1, \phi_2) = \begin{pmatrix} -(D_x^- \phi_1 + D_y^- \phi_2) + \lambda_D(x)(u - f) \\ \max\left(1, \gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2}\right) \phi_1 - \gamma D_x^+ u \\ \max\left(1, \gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2}\right) \phi_2 - \gamma D_y^+ u \end{pmatrix}.$$

El método de Newton semi-suave permite encontrar las raíces de $F(u, \phi_1, \phi_2)$, es decir, la solución de

$$F(u, \phi_1, \phi_2) = 0.$$

Ya que la función \max no es diferenciable, F' representará la derivada en el sentido de Newton y está dada por:

$$F'(u, \phi_1, \phi_2) = \begin{pmatrix} \lambda_D(x)I & -D_x^- & -D_y^- \\ M_1 - \gamma D_x^+ & \operatorname{diag}(R) & 0 \\ M_2 - \gamma D_y^+ & 0 & \operatorname{diag}(R) \end{pmatrix},$$

notando $R = \max\left(1, \gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2}\right)$, M_1 y M_2 definidas por:

$$\begin{aligned} M_1 &= \operatorname{diag}(H) \gamma \left[\operatorname{diag}(h_1) D_x^+ + \operatorname{diag}(h_2) D_y^+ \right] \operatorname{diag}(\tilde{\phi}_1), \\ M_2 &= \operatorname{diag}(H) \gamma \left[\operatorname{diag}(h_2) D_x^+ + \operatorname{diag}(h_1) D_y^+ \right] \operatorname{diag}(\tilde{\phi}_2) \end{aligned}$$

con h_1, h_2 y H dados por:

$$h_1 = \frac{D_x^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2}}, \quad h_2 = \frac{D_y^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2}},$$

$$(H)_i = \begin{cases} 0, & \text{si } \left(\gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2} \right)_i < 1 \\ 1, & \text{caso contrario.} \end{cases}$$

Además, $\tilde{\phi}_1$ y $\tilde{\phi}_2$ denotan las proyecciones de ϕ_1 and ϕ_2 , respectivamente, es decir:

$$\begin{aligned} \tilde{\phi}_1 &= \frac{\phi_1}{\max\left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}\right)} \\ \tilde{\phi}_2 &= \frac{\phi_2}{\max\left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}\right)}. \end{aligned}$$

El esquema iterativo que se resolverá hasta que $F(u_k, \phi_k^1, \phi_k^2)$ este lo suficientemente cerca de cero, será:

$$\begin{aligned} F'(u_k, \phi_k^1, \phi_k^2) \Delta(u_k, \phi_k^1, \phi_k^2) &= -F(u_k, \phi_k^1, \phi_k^2) \quad k = 0, 1, \dots \\ (u_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) &= (u_k, \phi_k^1, \phi_k^2) + \Delta(u_k, \phi_k^1, \phi_k^2). \end{aligned}$$

Gracias a la proyección realizada, la matriz F' es una matriz definida positiva, lo que desde el punto de vista de la optimización matemática es una ventaja pues se obtiene una dirección de descenso

$$p_k = -(F'_k)^{-1} F_k.$$

Así, el esquema de Newton semi-suave sigue siendo un esquema que converge superlinealmente, pero con la ventaja de ser globalmente convergente (ver [22],[14]).

Algorithm 1 Ecuación de Euler-Lagrange

- 1: **Inicializar** $k = 0 \quad u_0 = f$
 - 2: $\phi_k^1 = \phi_k^2 = 0$
 - 3: $\lambda > 0$
 - 4: $\gamma > 0$
 - 5: **Repetir**
 - 6: $F'(\mu_k, \phi_k^1, \phi_k^2) \Delta(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) = -F(\mu_k, \phi_k^1, \phi_k^2)$
 - 7: **Actualizar**
 - 8: $(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) = (\mu_k, \phi_k^1, \phi_k^2) + \Delta(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2)$
 - 9: $k = k + 1$
 - 10: **Hasta** $\|F(\mu_k, \phi_k^1, \phi_k^2)\| < tol$
 - 11: **Retorna** μ_k
-

Algoritmo ecuación de reacción-difusión anisotrópica

Se presenta a continuación la implementación numérica de la ecuación evolutiva de (3.18). Se busca resolver el sistema regularizado utilizando la primera derivada de Huber,

$$\begin{cases} \frac{\partial u}{\partial t} = \operatorname{div} \vec{\phi} - \lambda_D(x)(u - f) \\ \max(1, \gamma |\nabla u|) \vec{\phi} = \gamma \nabla u. \end{cases}$$

Se utilizó un esquema implícito para la discretización de la variable temporal, determinado por:

$$u^{k+1} = u^k + \Delta t \left[\operatorname{div} \left(\frac{\nabla u^{k+1}}{|\nabla u^{k+1}|} \right) - \lambda_D(x)(u^{k+1} - f) \right].$$

La discretización y formulación matricial asociado está determinado por F :

$$F(u, \phi_1, \phi_2) = \begin{pmatrix} u - d - \Delta t [D_x^- \phi_1 + D_y^- \phi_2 - \lambda_D(x)(u - f)] \\ \text{máx} \left(1, \gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2} \right) \phi_1 - \gamma D_x^+ u \\ \text{máx} \left(1, \gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2} \right) \phi_2 - \gamma D_y^+ u \end{pmatrix}.$$

El método de Newton semi-suave permite encontrar las raíces de $F(u, \phi_1, \phi_2)$, es decir, la solución de

$$F(u, \phi_1, \phi_2) = 0.$$

Ya que la función máx no es diferenciable, F' representará la derivada en el sentido de Newton y está dada por:

$$F'(u, \phi_1, \phi_2) = \begin{pmatrix} I + \Delta t \lambda_D(x) I & -\Delta t D_x^- & -\Delta t D_y^- \\ M_1 - \gamma D_x^+ & \text{diag}(R) & 0 \\ M_2 - \gamma D_y^+ & 0 & \text{diag}(R) \end{pmatrix},$$

notando $R = \text{máx} \left(1, \gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2} \right)$, M_1 y M_2 están definidas por:

$$\begin{aligned} M_1 &= \text{diag}(H) \gamma \left[\text{diag}(h_1) D_x^+ + \text{diag}(h_2) D_y^+ \right] \text{diag}(\tilde{\phi}_1), \\ M_2 &= \text{diag}(H) \gamma \left[\text{diag}(h_2) D_x^+ + \text{diag}(h_1) D_y^+ \right] \text{diag}(\tilde{\phi}_2) \end{aligned}$$

con h_1, h_2 y H dados por:

$$h_1 = \frac{D_x^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2}}, \quad h_2 = \frac{D_y^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2}},$$

$$(H)_i = \begin{cases} 0, & \text{si } \left(\gamma \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2} \right)_i < 1 \\ 1, & \text{caso contrario.} \end{cases}$$

Además, $\tilde{\phi}_1$ y $\tilde{\phi}_2$ denotan las proyecciones de ϕ_1 and ϕ_2 , respectivamente, es decir:

$$\begin{aligned} \tilde{\phi}_1 &= \frac{\phi_1}{\text{máx} \left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2} \right)} \\ \tilde{\phi}_2 &= \frac{\phi_2}{\text{máx} \left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2} \right)}. \end{aligned}$$

Algorithm 2 Ecuación de reacción difusión anisotrópica

- 1: **Inicializar** $k = 0$
 - 2: $u_0 = f$
 - 3: $\phi_k^1 = \phi_k^2 = 0$
 - 4: $\lambda > 0$
 - 5: $\gamma > 0$
 - 6: **Resolver**
 - 7: $F'(\mu_k, \phi_k^1, \phi_k^2) \Delta(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) = -F(\mu_k, \phi_k^1, \phi_k^2)$
 - 8: **Actualizar**
 - 9: $(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) = (\mu_k, \phi_k^1, \phi_k^2) + \Delta(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2)$
 - 10: $k = k + 1$
 - 11: **SI** la solución no es estacionaria
 - 12: $k = k + 1$ y volver a 6
 - 13: **Caso Contrario**
 - 14: **Retorna** μ_k
-

Algoritmo movimiento de la curvatura media

Se presenta a continuación la implementación numérica de la ecuación:

$$\frac{\partial u}{\partial t} = |\nabla u| \left[\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right] - \lambda_{D(x)}(u - f). \quad (3.19)$$

Para su implementación numérica se utilizó un esquema implícito para aproximar la variable temporal y se implementó el método de Newton semi-suave para resolver el sistema asociado en cada iteración.

$$u^{k+1} = u^k + \Delta t \left[|\nabla u^{k+1}| \operatorname{div} \left(\frac{\nabla u^{k+1}}{|\nabla u^{k+1}|} \right) - \lambda_{D(x)}(u^{k+1} - f) \right].$$

Así, se busca resolver el sistema regularizado utilizando la regularización de Berkovier Engelman, dado por:

$$\begin{cases} \frac{\partial u}{\partial t} = |\nabla u|_\epsilon \operatorname{div} \vec{\phi} - \lambda_D(x)(u - f) \\ |\nabla u|_\epsilon \vec{\phi} = \nabla u. \end{cases}$$

El problema discretizado en forma matricial tiene asociado a F tal que

$$F(u, \phi_1, \phi_2) = \begin{pmatrix} u - d - \Delta t \left(R(D_x^- \phi_1 + D_y^- \phi_2) - \lambda_{D(x)}(u - f) \right) \\ R\phi_1 - D_x^+ u \\ R\phi_2 - D_y^+ u \end{pmatrix},$$

donde $R = \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}$. Su matriz jacobiana asociada F'

$$F'(u, \phi_1, \phi_2) = \begin{pmatrix} I - \Delta t(C - \lambda_{D(x)}I) & -\Delta t \operatorname{diag}(R) D_x^- & -\Delta t \operatorname{diag}(R) D_y^- \\ B_1 - D_x^+ & \operatorname{diag}(R) & 0 \\ B_1 - D_y^+ & 0 & \operatorname{diag}(R) \end{pmatrix}$$

donde,

$$B_1 = [\text{diag}(b_1)D_x^+ + \text{diag}(b_2)D_y^+]\text{diag}(\tilde{\phi}_1), \quad B_2 = [\text{diag}(b_1)D_x^+ + \text{diag}(b_2)D_y^+]\text{diag}(\tilde{\phi}_2)$$

$$C = \left(\text{diag}(b_1)D_x^+ + \text{diag}(b_2)D_y^+ \right) \left(D_x^- \phi_1 + D_y^- \phi_2 \right),$$

con

$$b_1 = \frac{D_x^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}}, \quad b_2 = \frac{D_y^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}}$$

Además, $\tilde{\phi}_1$ y $\tilde{\phi}_2$ denotan las proyecciones de ϕ_1 and ϕ_2 , respectivamente, es decir:

$$\tilde{\phi}_1 = \frac{\phi_1}{\max\left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}\right)}$$

$$\tilde{\phi}_2 = \frac{\phi_2}{\max\left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}\right)}.$$

El esquema iterativo asociado está dado por:

$$F'(u_k, \phi_k^1, \phi_k^2) \Delta(u_k, \phi_k^1, \phi_k^2) = -F(u_k, \phi_k^1, \phi_k^2) \quad k = 0, 1, \dots$$

$$(u_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) = (u_k, \phi_k^1, \phi_k^2) + \Delta(u_k, \phi_k^1, \phi_k^2),$$

Se propone además, resolver la versión estacionaria de la ecuación (3.19), es decir:

$$|\nabla u| \left[\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right] - \lambda_{D(x)}(u - f) = 0. \quad (3.20)$$

Así, se busca resolver el sistema regularizado utilizando la regularización de Berkovier Engelman, dado por:

$$\begin{cases} |\nabla u|_\epsilon \text{div } \vec{\phi} - \lambda_{D(x)}(u - f) = 0 \\ |\nabla u|_\epsilon \vec{\phi} = \nabla u. \end{cases}$$

El problema discretizado en forma matricial tiene asociado a F tal que

$$F(u, \phi_1, \phi_2) = \begin{pmatrix} \sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2} \left(D_x^- \phi_1 + D_y^- \phi_2 \right) - \lambda_{D(x)}(u - f) \\ \left(\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2} \right) \phi_1 - D_x^+ u \\ \left(\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2} \right) \phi_2 - D_y^+ u \end{pmatrix},$$

y su matriz jacobiana asociada F'

$$F'(u, \phi_1, \phi_2) = \begin{pmatrix} C - \lambda_{D(x)} I & \text{diag}(R)D_x^- & \text{diag}(R)D_y^- \\ B_1 - D_x^+ & \text{diag}(R) & 0 \\ B_1 - D_y^+ & 0 & \text{diag}(R) \end{pmatrix}$$

donde,

$$R = \left(\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2} \right),$$

$$B_1 = [\text{diag}(b_1)D_x^+ + \text{diag}(b_2)D_y^+] \text{diag}(\tilde{\phi}_1), \quad B_2 = [\text{diag}(b_1)D_x^+ + \text{diag}(b_2)D_y^+] \text{diag}(\tilde{\phi}_2)$$

$$C = \left(\text{diag}(b_1)D_x^+ + \text{diag}(b_2)D_y^+ \right) \left(D_x^- \phi_1 + D_y^- \phi_2 \right).$$

con

$$b_1 = \frac{D_x^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}},$$

$$b_2 = \frac{D_y^+ u}{\sqrt{(D_x^+ u)^2 + (D_y^+ u)^2 + \epsilon^2}}$$

Además, $\tilde{\phi}_1$ y $\tilde{\phi}_2$ denotan las proyecciones de ϕ_1 and ϕ_2 , respectivamente, es decir:

$$\tilde{\phi}_1 = \frac{\phi_1}{\max\left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}\right)}$$

$$\tilde{\phi}_2 = \frac{\phi_2}{\max\left(1, \sqrt{(\phi_1)^2 + (\phi_2)^2}\right)}.$$

Algorithm 3 Ecuación del movimiento de curvatura media

1: **Inicializar**

2: $u_0 = \phi_k^1 = \phi_k^2 = 0$

3: $\lambda > 0$

4: $\epsilon > 0$

5: **Repetir**

6: $F'(\mu_k, \phi_k^1, \phi_k^2) \Delta(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) = -F(\mu_k, \phi_k^1, \phi_k^2)$

7: **Actualizar**

8: $(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2) = (\mu_k, \phi_k^1, \phi_k^2) + \Delta(\mu_{k+1}, \phi_{k+1}^1, \phi_{k+1}^2)$

9: $k = k + 1$

10: **Verificar**

11: **SI** la solución no es estacionaria

12: $k = k + 1$ y volver a 5

13: **Caso Contrario**

14: **Retorna** μ_k

Capítulo 4

Implementación y experimentos computacionales

En este capítulo, se exponen los resultados y conclusiones obtenidas a partir de la implementación de los algoritmos propuestos para resolver el problema de inpainting.

Se aplicó el método de aproximación por diferencias finitas, ampliamente utilizado en el procesamiento matemático de imágenes, debido a la estructura de la imagen digital. Se consideran $\Omega = [0, 1] \times [0, 1]$ el dominio de las imágenes en escala de grises, $n_x \times n_y$ la dimensión de las mismas y $h_x = 1/(n_x + 1)$ el paso de discretización de la variable x y en y , $h_y = 1/(n_y + 1)$.

Para evidenciar que estos algoritmos son eficientes cuando no se buscan recuperar pequeños detalles, se presentan ejemplos sencillos de dominios perdidos así como dominios que serán un reto mayor.

4.1. Comparación de modelos de inpainting

En esta sección se presenta la comparación gráfica de los resultados obtenidos utilizando: el modelo de variación total, la regularización de Tikhonov para inpainting (ver anexo A.1.1), movimiento de curvatura media y la ecuación modificada de Cahn-Hilliard.

Como se observa en la Figura 4.1, para el primer dominio perdido que tiene el 15% de daño, los modelos de TV inpainting y curvatura media obtienen los mejores resultados pues recuperan todo el borde, lo que no ocurre con el resto de modelos.

En los dominios perdidos restantes, ninguno de los modelos presentados es capaz de recuperar los bordes, sin embargo se debe notar que la restauración de estos es un reto mayor pues el dominio a restaurar es de mayor tamaño y dificultad. Se puede concluir así que, para recuperar dominios relativamente pequeños se obtienen excelentes resultados visuales con el modelo de TV inpainting, pues es capaz de recuperar bordes.

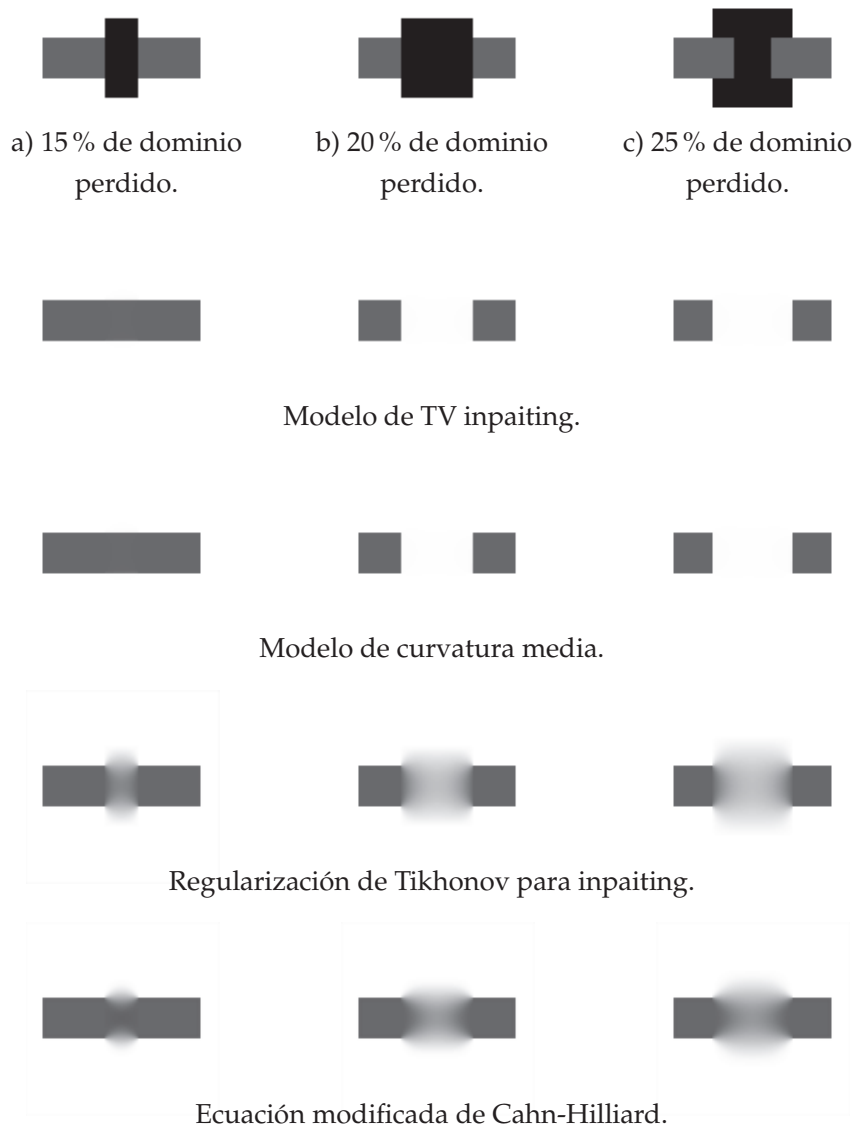


Figura 4.1. Simulación de dominios perdidos. **Fuente:** Elaboración propia.

4.2. Análisis experimental de los algoritmos implementados

En esta sección, se presenta la simulación de dominios perdidos D y la reconstrucción de los mismos. Para esto se utilizó imágenes f con distintas características como: brillo, bordes, contenido y textura. Lo que permitió analizar el comportamiento de los algoritmos propuestos en este trabajo.

La implementación numérica se realizó haciendo uso del software de programación Matlab, en un equipo con procesador de Intel(R) Core(TM) i5 de 1.7 GHz con 8GB de memoria RAM.

Experimento 1: en la primera simulación se utilizó la imagen de Lena muy popular en el procesamiento de imágenes, ya que presenta una mezcla de detalles como sombras, bordes y texturas. En este primer experimento se pierde el 25% de la imagen utilizando puntos como se observa en la Figura 4.2,



Figura 4.2. Simulación de domino perdido utilizando puntos.

En este experimento se pretende analizar el comportamiento del algoritmo 1, donde se emplea la regularización de Huber, al que se denominó *TV Huber* y del algoritmo *TV Engelman* donde se utiliza la regularización Berkovier Engelman (ver apéndice B.1.2). Se busca evidenciar la influencia de los parámetros de regularización sobre estos algoritmos, para lo cual analizaremos el número de iteraciones y el tiempo de ejecución, es decir, el tiempo que tarda cada algoritmo en alcanzar la solución u para distintos parámetros de regularización.

Para la implementación numérica se fija la tolerancia en 10^{-6} , para el criterio de parada dado por $\|F(u_k, \phi_k^1, \phi_k^2)\| < \text{tolerancia}$, que corresponde al test de parada empleado para hallar las raíces utilizando el método de Newton semi-suave del sistema asociado en cada iteración.

El parámetro λ como se explicó previamente, es crucial para conseguir el equilibrio entre la estabilidad de la solución y la calidad de los datos. Si λ es demasiado pequeño, se obtiene una solución sobre-regularizada como lo muestra la Figura 4.4, donde el suavizado en la imagen restaurada es excesivo. Por el contrario, si λ es demasiado grande el algoritmo puede ser inestable. Esto se ve reflejado en la Tabla 4.1, pues mientras más pequeño sea el valor que tome λ son necesarias más iteraciones y el tiempo de ejecución es mayor.

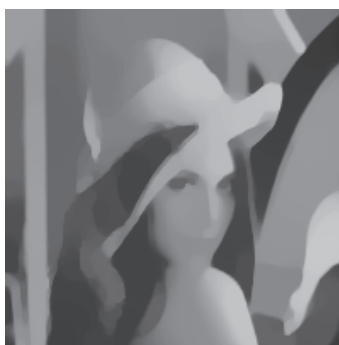


Figura 4.3. Imagen restaurada utilizando $\lambda = 10^2$.

Los parámetros de regularización γ y ϵ determinarán una mejor aproximación de los términos no diferenciables, tema muy importante en la implementación numérica, ya que el valor que tomen estos no influye en la solución u pero sí, en el tiempo y número de iteraciones de la convergencia de los algoritmos planteados.

En el caso del parámetro de regularización γ , teóricamente hablando, debería tomar un valor muy grande lo que ocasiona más iteraciones y más tiempo de ejecución como se observa en la Tabla 4.1. En la práctica basta tomar el valor $\gamma = 10$ para obtener buenos resultados. Por otro lado, el parámetro ϵ debería ser muy pequeño, lo que en la implementación numérica resulta en un problema inestable, por lo cual se utiliza una proyección. Así, tomar 10^{-4} es suficiente en la práctica. Se puede concluir que los algoritmos planteados son estables, independientemente del valor que tomen los parámetros λ , γ y ϵ .

	Parámetro λ	Número de iteraciones	Parámetro de regularización	Tiempo (min.)	
TV Huber	10^6	14	$\gamma = 10$	1.35	
	10^4	12		1.14	
	10^2	16		3.72	
	10^6		20	$\gamma = 100$	3.03
			12	$\gamma = 50$	1.71
			12	$\gamma = 25$	1.69
TV Engelman	10^6	13	$\epsilon = 10^{-4}$	0.55	
	10^4	15		0.88	
	10^2	27		5.77	
	10^6		13	$\epsilon = 10^{-2}$	0.33
			16	$\epsilon = 10^{-5}$	0.79

Tabla 4.1. Número de iteraciones y tiempo de ejecución, para distintos valores de γ y ϵ .

Como se esperaba la solución encontrada por los algoritmos TV Huber y TV Engelman con $\lambda = 10^6$ y los parámetros de regularización $\gamma = 10$ y $\epsilon = 10^{-4}$, es la misma a simple vista como se observa en la Figura 4.4. El emplear distintas regularizaciones en la implementación numérica con los parámetros adecuados llevará a utilizar menos tiempo y menos iteraciones.

La elección del parámetro λ , para el cual se obtiene una solución aceptable visualmente hablando, se realizó experimentalmente lo que se podría plantear como un problema de optimización paralelo al problema de inpainting.



Figura 4.4. Imagen restaurada utilizando TV Huber y TV Engelman respectivamente.

Se evidencia que los algoritmos planteados son potentes, pues son capaces de recuperar la imagen prácticamente en su totalidad utilizando aproximadamente un minuto lo que es eficiente, considerando que las matrices que se emplean son de dimensión $3n^2 \times 3n^2$. Se debe notar que los resultados dependerán de la precisión en la localización de los píxeles defectuosos. En este experimento se posee información exacta de su ubicación como se observa en la Figura 4.5. La precisión de la solución dependerá del método de segmentación que se emplee para localizar los píxeles defectuosos.

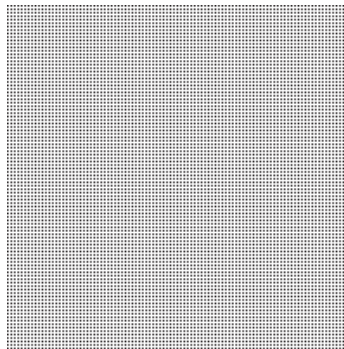


Figura 4.5. Domino perdido utilizando puntos.

Experimento 2: se utilizó una imagen con varios bordes a la cual se incorporaron líneas para perder el 25% del domino como se observa en la Figura 4.4, lo que incrementa la dificultad respecto al experimento anterior.

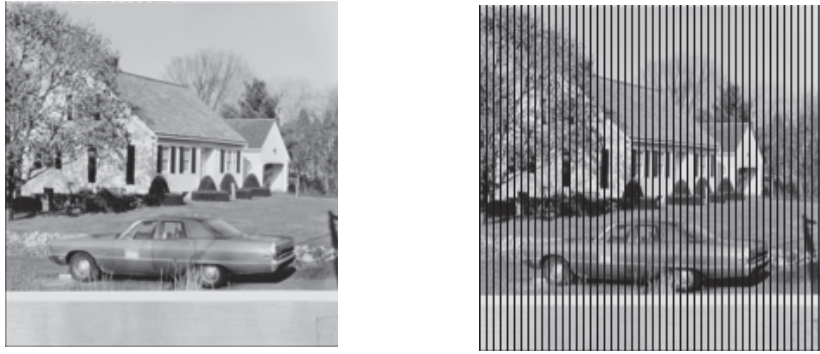


Figura 4.6. Simulación de domino perdido usando líneas.

En este experimento se pretende analizar el comportamiento del algoritmo 2, donde se emplea la regularización de Huber, al que se denominó *EDA Huber* y del algoritmo *EDA Engelman* donde se utiliza la regularización Berkovier Engelman (ver apéndice B.1.2).

El paso de discretización de la variable temporal Δt , se empezó probando con $1/2$ hasta tomar $1/n$, se evidenció que se emplean un número de iteraciones mayor. Se muestran los resultados para $\Delta = 1/2$.

	Parámetro λ	Número de iteraciones	Parámetro de regularización	Tiempo (min.)
EDA Huber	10^6	4/20	$\gamma = 10$	1.35
	10^4	4/18		1.14
	10^2	5/22		3.72
	10^6	3/27	$\gamma = 100$	3.03
		3/23	$\gamma = 50$	1.71
	10^4	3/18	$\gamma = 25$	1.69
10^5		3/19		
EDA Engelman	10^6	3/20	$\epsilon = 10^{-4}$	2.06
	10^4	3/13		0.68
	10^2	3/19		3.36
	10^6	3/18	$\epsilon = 10^{-2}$	2.27
		3/16	$\epsilon = 10^{-5}$	2.20
	10^4	3/18	$\epsilon = 10^{-3}$	0.96
		10^5	3/13	$\epsilon = 10^{-5}$

Tabla 4.2. Número de iteraciones/sistemas no lineales y tiempo de ejecución, para distintos valores de λ y ϵ .

Como se esperaba la solución encontrada por los algoritmos *EDA Huber* y *EDA Engelman* con $\lambda = 10^6$ y los parámetros de regularización $\gamma = 10$ y $\epsilon = 10^{-4}$, es la misma a simple vista, pues el emplear distintas regularizaciones en la implementación numérica considerando parámetros adecuados, llevará a utilizar menos tiempo y menos iteraciones. Se observa en la Figura 4.7 la imagen restaurada:



Figura 4.7. Imagen restaurada utilizando EDA Engelman.

El resultado sigue siendo satisfactorio a la vista, a pesar de que existen zonas donde no se alcanza la precisión anterior (ver Figura 4.7).

Para evidenciar la relación entre el paso de discretización h de las variables espaciales x y y , y el paso de discretización de la variable temporal Δt se presentan en la Figura 4.8 los gráficos del error absoluto

$$\|u_o - u_t\|_{L^2(\Omega)}$$

donde u_o es la imagen original sin daños de color azul y de color rojo se muestra el error absoluto

$$\|u - u_t\|_{L^2(\Omega)}$$

donde u es la imagen restaurada que se obtiene con el algoritmo EDA Huber.

La Figura 4.8 muestra que el algoritmo es estable independientemente del valor que pueda tomar Δt . Ha medida que se aumenta este paso de discretización el error absoluto tiende a cero, lo que resalta la potencia del método pues en los gráficos de la primera columna se está comparando la solución obtenida en cada iteración con la imagen original sin daño. No se espera que este error tienda a cero pues el algoritmo proporciona una aproximación numérica de la imagen restaurada.

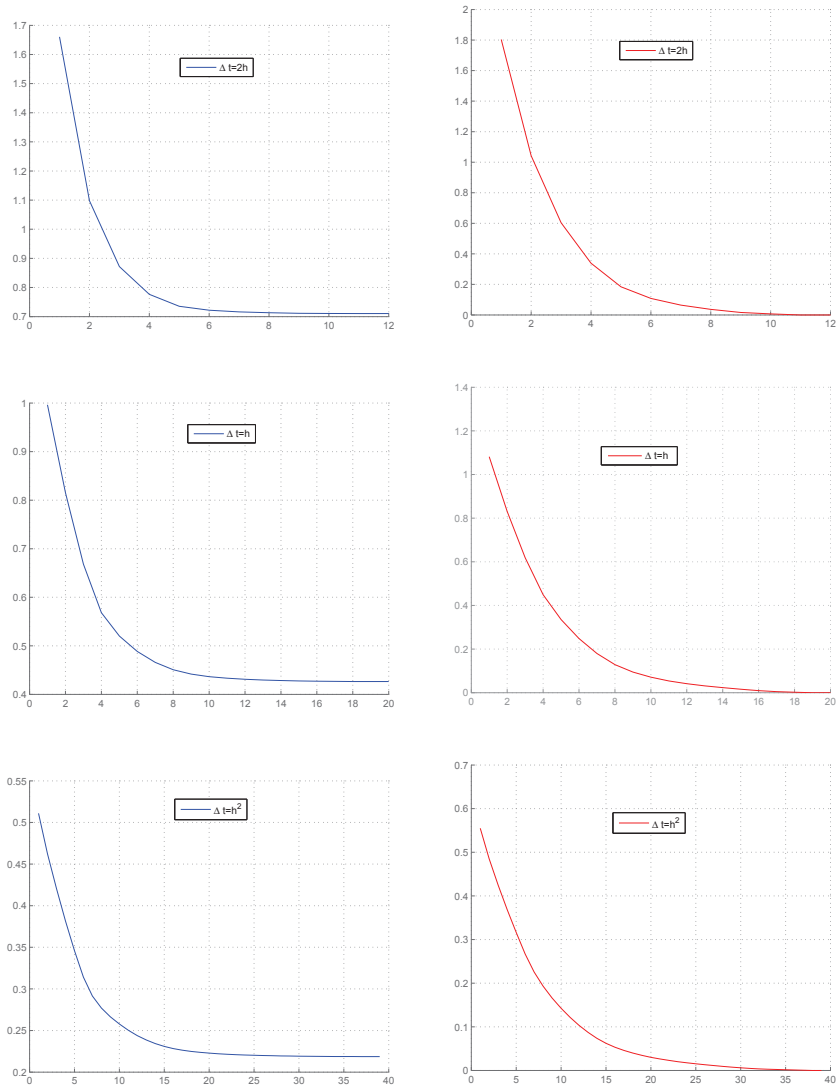


Figura 4.8. Error absoluto versus el número de iteraciones, con h, γ y λ fijos.

Experimento 3: se utilizó la imagen con detalles grandes a la cual se le provoca un daño del 25 % en su dominio de la imagen utilizando puntos como se observa en la Figura 4.9,



Figura 4.9. Simulación de domino perdido utilizando puntos.

En este experimento se pretende analizar el comportamiento del algoritmo 3, donde se resuelve el modelo de curvatura media estacionaria, al que se denominó *Curvaturam* y del algoritmo *CurvaturamT* donde se resuelve la versión dinámica; en estos se empleó la regularización Berkovier Engelman. Se busca evidenciar la influencia de los parámetros sobre estos algoritmos, para lo cual analizaremos el número de iteraciones y el tiempo de ejecución, es decir, el tiempo que tarda cada algoritmo en alcanzar la solución u .

Para la implementación numérica se fijó la tolerancia en 10^{-3} , que corresponde al test de parada empleado para hallar las raíces utilizando el método de Newton clásico del sistema asociado en cada iteración dado por $\|F(u_k, \phi_k^1, \phi_k^2)\| < \text{tolerancia}$. En el algoritmo *CurvaturamT*, el paso de discretización de la variable temporal Δt , se empezó probando con $1/2$ hasta tomar $1/n$, se evidenció que los algoritmos converge en un número de iteraciones mayor pero no se encontraron diferencias marcadas, por lo cual solo se muestran los resultados para $\Delta = 1/2$.

En la Tabla 4.3 se muestran los experimentos realizados para distintos valores de λ y ϵ , como se observa el algoritmo *Curvaturam* no converge cuando se toma $\lambda = 10^2$ o $\lambda = 10^3$ y $\epsilon = 10^{-4}$, mientras que el algoritmo *CurvaturamT* si lo hace en el caso de $\lambda = 10^3$. Lo que no se considera una desventaja, pues nuestro objetivo no es obtener soluciones sobre regularizadas, lo que se obtiene para $\lambda \leq 10^2$. Respecto al tiempo de ejecución el primer algoritmo tarda aproximadamente 0.2 minutos menos que el algoritmo *CurvaturamT*.

	Parámetro λ	Número de iteraciones	Parámetro de regularización	Tiempo (min.)	
Curvaturam	10^6	13	$\epsilon = 10^{-4}$	0.49	
	10^4	12		0.42	
	10^2	-		-	
	10^6	8	$\epsilon = 10^{-2}$	0.21	
		12	$\epsilon = 10^{-5}$	0.41	
	10^5	12	$\epsilon = 10^{-5}$	0.42	
	10^4	11	$\epsilon = 10^{-3}$	0.29	
	10^3	-	$\epsilon = 10^{-4}$	-	
	CurvaturamT	10^6	17/26	$\epsilon = 10^{-4}$	0.75
		10^4	17/26		0.65
10^2		-	-		
10^6		16/26	$\epsilon = 10^{-2}$	0.66	
		16/26	$\epsilon = 10^{-5}$	0.68	
10^5		16/26	$\epsilon = 10^{-5}$	0.68	
10^4		17/26	$\epsilon = 10^{-3}$	0.65	
10^3		17/28	$\epsilon = 10^{-4}$	0.68	

Tabla 4.3. Número de iteraciones/sistemas no lineales y tiempo de ejecución, para distintos valores de λ y ϵ .

Como se esperaba la solución encontrada por los algoritmos Curvaturam y CurvaturamT con $\lambda = 10^6$ y $\epsilon = 10^{-4}$, es la misma a simple vista como se observa en la Figura 4.10. El emplear la ecuación estacionaria y la evolutiva del movimiento de curvatura media para resolver el problema de inpainting con los parámetros adecuados llevará a utilizar menos tiempo y menos iteraciones que en los experimentos anteriores.



Figura 4.10. Imagen restaurada utilizando los algoritmos Curvaturam y CurvaturamT.

Experimento 4: en este experimento se pretende comparar los resultados obtenidos utilizando el modelo de TV inpainting y el modelo de curvatura media. Para lo cual se empleó una imagen con círculos donde se debe completar varios bordes, como se observa en la Figura 4.11



Figura 4.11. Imagen con 10 % de dominio defectuoso.

El resultado obtenido por los modelos de TV inpainting y el de, curvatura media se muestran en la Figura 4.12, visualmente la imagen restaurada u es la misma para estos dos modelos. Para verificar esto, calculamos el error relativo respecto de u_o la imagen original sin daños, es decir,

$$\|u_o - u\|_{L^2(\Omega)},$$

para el modelo de TV inpainting el error es de 0.23 y para el modelo de curvatura media es de 0.22, lo que no representa una diferencia significativa. Respecto del tiempo de ejecución el modelo de curvatura media es más rápido pues utiliza 0.4 minutos menos que el modelo de TV inpainting.



Figura 4.12. Imagen restaurada utilizando los modelos de curvatura media y TV inpainting.

Experimento 5: para este experimento se implemento el método del descenso del gradiente que tiene asociado el siguiente esquema:

$$u^{k+1} = u^k + \alpha(-E'[u^k])$$

si lo miramos con detenimiento este esquema es equivalente al esquema explícito a un paso, donde el paso de descenso α representa el paso de discretización de la variable temporal Δt . La elección de este parámetro afecta directamente en la estabilidad del método. Para la implementación numérica de este método se utilizo la regularización de Berkovier Engelman.

En este experimento se fijo $\alpha = 1/100$, para valores mayores por ejemplo $1/2$ el algoritmo no es estable. Lo mismo sucede con el paso de discretización de las variables espaciales,

donde se tuvieron que fijar en 1, pues también se pierde estabilidad. Además, el parámetro de regularización ϵ se fijó en 0,01 pues para valores más pequeños el algoritmo no converge, por lo que no se alcanza el mínimo del funcional E_{TV} . Ya que no se verifica la condición de parada $\|u_{k+1} - u_k\| < tol$ se detuvo el algoritmo en 200 iteraciones. El objetivo de este experimento es comparar este algoritmo con los algoritmos propuestos y obtener conclusiones del aporte de este trabajo.

Se puede recuperar regiones perdidas relativamente grandes, sólo sí, se tiene suficiente información alrededor de dichas regiones. A continuación, se presentan imágenes con regiones perdidas del 5%, 7% y 10% junto con la imagen reconstruida. Los algoritmos de inpainting reparan la zona dañada, pero a medida que el área a reparar aumenta, más costoso será el método pues, es mayor la cantidad de píxeles que se deben estimar.



Figura 4.13. Dominio perdido del 5%.

Como se observa en la Figura 4.13 se tiene mucha información alrededor de cada pixel defectuoso y se obtiene una buena aproximación, trata de adoptar los colores del sombrero en la zona más clara y en la zona más oscura toma los colores de la parte de la viga de madera; como se esperaba no se recupera el borde del sombrero.



Figura 4.14. Dominio perdido del 7%.

En la Figura 4.14, se evidencia que la restauración dependerá no solo del tamaño del dominio perdido, sino también de la información que se tiene disponible y de las características como brillo, sombra que rodean al área a restaurar. Por ejemplo, los píxeles defectuosos del hombro se recuperan perfectamente, pues al rededor del mismo se tiene suficiente información, lo que no sucede con los píxeles localizados en el espejo, pues como se esperaba

no se pueden reconstruir el borde ya que los píxeles son de tamaño mayor a la información disponible y prevalece el color que los rodea en mayor proporción.



Figura 4.15. Dominio perdido del 10 %.

En la Figura 4.15, se muestran áreas perdidas de mayor tamaño y con texturas. Como es el caso del área de las plumas donde la restauración a la vista no es buena, pues se recupera el tono de gris más no la textura, lo que se esperaba pues el modelo de Chan-Shen recupera bordes de regiones perdidas de tamaño relativamente pequeño, como los píxeles de las vigas donde se recuperan el borde de las mismas totalmente. En la práctica dependerá del detalle que se quiera obtener en la restauración y del tipo de información que se quiera extraer, y en este caso los algoritmos implementados cumplen lo esperado.

El número de iteraciones y el tiempo de ejecución obtenidos en los ejemplos que se acaban de analizar, se muestran en la Tabla 4.4:

% Dominio perdido	Algoritmo								
	Gradiente	TV Engelman		EDA Engelman		Curvaturam		CurvaturamT	
5 %	200	11	0.20	3/25	0.20	9	0.11	17/26	0.65
7 %	200	11	0.22	3/24	0.20	12	0.15	17/26	0.68
10 %	200	14	0.29	3/20	0.18	13	0.18	18/27	0.67

Tabla 4.4. Número de iteraciones/sistemas no lineales y tiempo de ejecución.

Se evidencia así que, el algoritmo basado en la ecuación del movimiento de curvatura media emplea menos iteraciones y tarda menos en hallar la imagen restaurada, lo que no implica que se debe desmerecer al resto de algoritmos pues todos resuelven los ejemplos planteados en menos de 0.7 min y el costo computacional se puede decir que es pequeño comparado con la cantidad de datos que se deben procesar, y con el método del gradiente.

4.3. Aplicaciones de inpainting

Los algoritmos de inpainting presentados no sólo son capaces de recuperar información a través de la reconstrucción de partes perdidas, puesto que se puede modificar el contexto de las imágenes considerando los objetos que ocultan información como dominios perdidos.

Se puede además remover texto y restaurar imágenes deterioradas. La localización de los píxeles defectuosos se determinó con la ayuda de la herramienta *Color Thresholder app* de Matlab, que permite crear una máscara a través de la intensidad de color. A continuación se presentan estas aplicaciones:

4.3.1. Remover texto

La Figura 4.16 muestra la aplicación de la restauración de dominios perdidos para suprimir o remover el texto impreso sobre una imagen.



Figura 4.16. Ejemplo de la remoción de texto.

Para remover el texto, se considera como dominio perdido todos los píxeles donde se encuentra el texto. La localización de los píxeles a ser restaurados se realizó en Matlab como se observa en la Figura 4.17 la localización es exacta, lo que hace que el proceso de inpainting obtenga una excelente aproximación de la imagen sin texto.



Figura 4.17. Domino perdido.

Capítulo 5

Conclusiones

El algoritmo propuesto para inpainting puede ser empleado para reconstruir partes dañadas, remover objetos indeseados o cambiar el contexto de una imagen. Este es un proceso complejo, especialmente cuando la intervención del usuario es mínima, pues existen aplicaciones donde el usuario señala segmentos a seguir en la imagen para la coherencia en la reconstrucción de las áreas defectuosas, otras donde se indica parches, es decir, patrones para la restauración.

En este trabajo se presentaron algoritmos numéricos de segundo orden basados en el método de Newton semi-suave, que resuelven el problema de la reconstrucción de dominios perdidos eficientemente.

Se presentan a continuación las conclusiones obtenidas y las líneas de trabajo futuras:

- Las imágenes empleadas en los experimentos tuvieron que cambiarse a escala de grises, como se vio esta simplificación no afectó los resultados de inpainting. Este estudio se puede extender a imágenes a color.
- La introducción artificial de la variable temporal permite aprovechar el extenso campo desarrollado para las ecuaciones diferenciales, y así analizar propiedades como la convergencia y estabilidad de los métodos a un paso. Además, existen fenómenos físicos asociados a estas ecuaciones, lo que nos permiten entender la lógica que está detrás del proceso de inpainting y proponen variantes para mejorar los modelos.
- Al realizar los experimentos computacionales se pudo evidenciar que los algoritmos planteados son más rápidos que el método del gradiente, ampliamente utilizado para resolver numéricamente el problema de inpainting.
- Experimentalmente se mostró que los algoritmos propuestos son incondicionalmente estables, lo que se podría demostrar formalmente.
- Se obtuvieron resultados similares al emplear el modelo de TV inpainting y el modelo de curvatura media, para resaltar las ventajas de este último, se podría analizar el orden de convergencia del método.

- Como se vio en los experimentos realizados en imágenes con dominios perdidos relativamente pequeños, los bordes se completan satisfactoriamente, pues no es perceptible a simple vista la reconstrucción realizada. Y así, no es necesario aplicar un post-procesamiento. Si se busca completar bordes que presentan dominios perdidos de mayor tamaño, se pueden utilizar modelos de orden mayor, como es el caso del *modelo geodésico* donde se reemplaza $|\nabla u|$ por $g(|\nabla u|)$, donde $g(s) = s^p$ con $p \geq 1$.
- Se puede utilizar la paralelización de los algoritmos planteados si se busca restaurar imágenes con tamaños superiores a los 400×400 píxeles.
- La elección del parámetro λ influye notablemente en el resultado del problema de inpainting. Se puede plantear como trabajo futuro, el problema de la optimización del parámetro de regularización λ que depende de la imagen a restaurar.
- El modelo de Chan-Shen requiere de la localización de los píxeles defectuosos y como se vio la precisión del algoritmo dependerá del método empleado para la segmentación. Se puede plantear resolver un modelo donde se incorpore la segmentación al problema de inpainting.
- Las aplicaciones mostradas en este trabajo contemplan solo imágenes, sin embargo, se puede extender su aplicación a secuencias de imágenes, es decir, vídeos. Por ejemplo, se desea retirar un cuerpo en movimiento de una escena. La reconstrucción de imágenes estáticas se considera de mayor complejidad que reconstruir un vídeo pues se puede extraer información a través de las imágenes en varios instantes, mientras que en las imágenes eso no se tiene.
- Se podría realizar una interfaz sencilla para una aplicación amigable con el usuario.

Apéndice A

Anexo A

A.1. Cálculo Variacional

En esta sección, se introduce una noción del *cálculo variacional*. Para un estudio más detallado se puede consultar el libro de *Reddy y Rasmussen* [32]. Sea $\Omega \subset \mathbb{R}^2$, con $\partial\Omega \in C^{0,1}$. Se considera el funcional

$$\Theta[v] = \int_{\Omega} F(x, y, v, v_x, v_y) dx \quad (\text{A.1})$$

donde F es una función suficientemente suave, conocida como *lagrangiano*. Se considera el conjunto de funciones suaves $v : \overline{\Omega} \rightarrow \mathbb{R}$ que satisfacen la condición de contorno

$$v = h \text{ en } \partial\Omega. \quad (\text{A.2})$$

El cálculo variacional pretende hallar u tal que minimiza el valor del funcional (A.1) y verifica (A.2). En este caso se demuestra que, la función u es solución de una ecuación en derivadas parciales no lineal, conocida como *Ecuación de Euler-Lagrange*.

En efecto, sean u un mínimo de Θ , $w \in C_0^\infty(\Omega)$ la dirección y se define ρ como:

$$\rho(\epsilon) = \Theta[u + \epsilon w] \text{ con } \epsilon \in \mathbb{R}. \quad (\text{A.3})$$

Puesto que u es un mínimo de Θ , por definición ρ alcanza un mínimo en $\epsilon = 0$,

$$\rho'(0) = 0.$$

Por tanto, se requiere calcular explícitamente la derivada de ρ :

$$\begin{aligned} \rho(\epsilon) &= \Theta[u + \epsilon w] \\ &= \int_{\Omega} F(x, y, u + \epsilon w, u_x + \epsilon w_x, u_y + \epsilon w_y) dx, \end{aligned}$$

para calcular la derivada de ρ respecto de ϵ , se consideran la suavidad de F y la *regla de la*

cadena, tal que

$$\begin{aligned}
\frac{d\rho}{d\epsilon}(\epsilon) |_{\epsilon=0} &= \int_{\Omega} \frac{\partial F}{\partial \epsilon}(x, y, u + \epsilon w, u_x + \epsilon w_x, u_y + \epsilon w_y) dx \\
&= \int_{\Omega} \left[\frac{\partial F}{\partial w} w + \frac{\partial F}{\partial u_x} w_x + \frac{\partial F}{\partial u_y} w_y \right] dx \\
&= \int_{\Omega} \left[\frac{\partial F}{\partial u} w + \frac{\partial}{\partial x} \frac{\partial F}{\partial u_x} w + \frac{\partial}{\partial y} \frac{\partial F}{\partial u_y} w \right] dx \\
&= \int_{\Omega} \left[\frac{\partial F}{\partial u} + \frac{\partial}{\partial x} \frac{\partial F}{\partial u_x} + \frac{\partial}{\partial y} \frac{\partial F}{\partial u_y} \right] w dx,
\end{aligned}$$

entonces $\forall w \in C_0^\infty(\Omega)$

$$\frac{d\rho}{d\epsilon}(\epsilon) |_{\epsilon=0} = \int_{\Omega} \left[\frac{\partial F}{\partial u} + \frac{\partial}{\partial x} \frac{\partial F}{\partial u_x} + \frac{\partial}{\partial y} \frac{\partial F}{\partial u_y} \right] w dx = 0$$

debe cumplirse así que,

$$\frac{\partial F}{\partial u} + \frac{\partial}{\partial x} \frac{\partial F}{\partial u_x} + \frac{\partial}{\partial y} \frac{\partial F}{\partial u_y} = 0$$

esta es conocida como la *ecuación de Euler-Lagrange* y

$$\delta\Theta[u] = \frac{\partial F}{\partial u} + \frac{\partial}{\partial x} \frac{\partial F}{\partial u_x} + \frac{\partial}{\partial y} \frac{\partial F}{\partial u_y}$$

es la primera variación del funcional $\Theta[u]$. Con lo cual la ecuación de Euler-Lagrange se puede expresar como:

$$\delta\Theta[u] = 0.$$

A.1.1. Regularización de Tikhonov

La regularización de Tikhonov supone que los datos son suaves y continuos, y está determinada por:

$$R[u] = \int_{\Omega} |\nabla u|^2$$

de donde, el funcional de energía a minimizar será:

$$E[u] = \int_{\Omega} |\nabla u(x)|^2 dx + \frac{1}{2} \int_{\Omega} (u(x) - f(x))^2 dx. \quad (\text{A.4})$$

Para hallar el mínimo, se debe calcular la primera derivada direccional del funcional E con respecto a la dirección $v \in C_0^1(\Omega)$.

Dados u y v , la primera variación de E será $[E'(u)](v) = \rho'(0)$ donde $\rho(\epsilon) = E[u + \epsilon v]$.

Desarrollando (A.4), se obtiene:

$$E[u] = \int_{\Omega} (u_x^2 + u_y^2) dx + \frac{1}{2} \int_{\Omega} (u - f)^2 dx,$$

$\rho(\epsilon)$ será:

$$\begin{aligned}\rho(\epsilon) &= \int_{\Omega} (u + \epsilon v)_x^2 + (u + \epsilon v)_y^2 + \frac{1}{2}(u + \epsilon v - f)^2 dx \\ \rho'(\epsilon) &= \int_{\Omega} 2[(u + \epsilon v)_x v_x + (u + \epsilon v)_y v_y] + (u + \epsilon v - f)v dx\end{aligned}$$

y evaluando ρ' en 0

$$\begin{aligned}\rho'(0) &= 2 \int_{\Omega} u_x v_x + u_y v_y + (u - f)v dx \\ &= 2 \int_{\Omega} \nabla u \cdot \nabla v + (u - f)v dx.\end{aligned}$$

Sabemos que

$$\nabla v \cdot w = \nabla v \cdot w = -v \operatorname{div}(w) + \operatorname{div}(vw)$$

entonces, tomando $w = \nabla u$ el primer termino de $\rho'(0)$, obtenemos:

$$\begin{aligned}\int_{\Omega} \nabla u \cdot \nabla v dx &= \int_{\Omega} \nabla v \cdot w dx \\ &= \int_{\Omega} -v \operatorname{div}(w) + \operatorname{div}(vw) dx\end{aligned}$$

aplicando el teorema de la divergencia al segundo termino y, ya que $\operatorname{div}(\nabla u) = \Delta u$, se tiene que:

$$\int_{\Omega} \nabla u \cdot \nabla v dx = - \int_{\Omega} v \Delta u dx + \int_{\partial \Omega} v \frac{\nabla u}{|\nabla u|} \cdot n dS$$

suponiendo condiciones homogéneas de Neumann en la frontera, $\partial_n u = 0$, donde n es el vector normal exterior, se sigue que:

$$\int_{\Omega} \nabla u \cdot \nabla v dx = - \int_{\Omega} v \Delta u dx$$

por lo tanto,

$$[E'(u)](v) = \rho'(0) = -2 \int_{\Omega} v \Delta u dx + (u - f)v dx$$

y,

$$E'[u] = -2\Delta u + (u - f).$$

Así, la ecuación de Euler-Lagrange será:

$$\begin{cases} -2\Delta u + (u - f) = 0 \\ \partial_n u = 0 \end{cases}$$

A.1.2. Regularización de de variación total (TV)

El inconveniente en usar funcionales cuadráticos, como en el caso de la *regularización de Tikhonov*, es que no permite recuperar las discontinuidades de la imagen. Rudin, Osher y Fatemi [33], propusieron la variación total como funcional de regularización (A.5), para dar

solución al filtrado de ruido gaussiano.

$$R(u) = \|u\|_{TV(\Omega)} = \int_{\Omega} |\nabla u(x)| dx = \int_{\Omega} \sqrt{u_x^2 + u_y^2} dx \quad (\text{A.5})$$

Los mismo que propusieron minimizar el funcional de energía determinado por:

$$E[u] = \int_{\Omega} |\nabla u(x)| dx + \frac{1}{2} \int_{\Omega} (u(x) - f(x))^2 dx. \quad (\text{A.6})$$

Para hallar el mínimo de (A.6), se debe calcular la primera derivada direccional del funcional E , suponiendo que $\nabla u \neq 0$.

Dados u y v la primera variación de E será $[E'(u)](v) = \rho'(0)$ donde $\rho(\epsilon) = E[u + \epsilon v]$

$$\begin{aligned} \rho(\epsilon) &= \int_{\Omega} \sqrt{(u_x + \epsilon v_x)^2 + (u_y + \epsilon v_y)^2} + \frac{1}{2} (u + \epsilon v - f)^2 dx \\ \rho'(\epsilon) &= \int_{\Omega} \frac{v_x(u_x + \epsilon v_x) + v_y(u_y + \epsilon v_y)}{\sqrt{(u_x + \epsilon v_x)^2 + (u_y + \epsilon v_y)^2}} + (u + \epsilon v - f)v dx \\ \rho'(0) &= \int_{\Omega} \frac{v_x u_x + v_y u_y}{\sqrt{u_x^2 + u_y^2}} + (u - f)v dx \end{aligned}$$

de donde,

$$[E'(u)](v) = \int_{\Omega} \frac{\nabla v \nabla u}{|\nabla u|} + (u - f)v dx$$

integrando por partes el término

$$\int_{\Omega} \nabla v \frac{\nabla u}{|\nabla u|} dx$$

y con $w = \frac{\nabla u}{|\nabla u|}$ en $\nabla v \cdot w = -v \operatorname{div}(w) + \operatorname{div}(vw)$, se tiene:

$$\int_{\Omega} \nabla v \frac{\nabla u}{|\nabla u|} = - \int_{\Omega} v \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) dx + \int_{\Omega} \operatorname{div} \left(v \frac{\nabla u}{|\nabla u|} \right)$$

y por el teorema de la divergencia,

$$\int_{\Omega} \nabla v \frac{\nabla u}{|\nabla u|} = - \int_{\Omega} v \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) dx + \int_{\partial\Omega} v \frac{\nabla u}{|\nabla u|} n dS$$

se toman condiciones homogéneas de Neumann $\partial_n u = 0$, donde n es el vector normal exterior, se sigue que:

$$\begin{aligned} [E'(u)](v) &= - \int_{\Omega} v \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) + (u - f)v dx \\ &= - \int_{\Omega} v \left(\operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) + (u - f) \right) dx \end{aligned}$$

por tanto para cualquier dirección v se tiene

$$E'(u) = -\operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) + (u - f).$$

pero esta debe ser cero. Así, la ecuación de Euler-Lagrange será:

$$\begin{cases} -\operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) + (u - f) = 0 \\ \partial_n u = 0. \end{cases}$$

Apéndice B

Anexo B

B.1. Códigos de programación en Matlab

B.1.1. Algoritmos generales

Algoritmo para resolver Problema 4

```
function [w,u]=dfqp(a,b,p,q,f,n,u0,un)
syms x;
h=(b-a)/(n+1);

Au=(1/(h^2))*(-diag(ones(1,n-1),1)+2*diag(ones(1,n))-diag(ones(1,n-1),-1));
t=1;

Aj=(-diag(ones(1,n-1),-1)+diag(ones(1,n)));
Ap(t,:)=subs(p,x,a+(t*h))*h*Aj(t,:);
Ap(n,:)=subs(p,x,a+(n*h))*h*Aj(n,:);
K(t)=subs(f,x,a+(t*h))+u0/(h^2)+(subs(p,x,a+n*h)*u0)/h;
K(n)=subs(f,x,a+(n*h))+un/(h^2);
Q(t)=h^2*subs(q,x,a+(t*h));
Q(n)=h^2*subs(q,x,a+(n*h));

for t=2:n-1;
K(t)=subs(f,x,a+(t*h));
Q(t)=h^2*subs(q,x,a+(t*h));
Ap(t,:)=subs(p,x,a+(t*h))*h*Aj(t,:);
end

B=Au+(1/(h^2))*diag(Q)+(1/(h^2))*Ap;
u=B\K';
%y=dsolve('-D2y-2*Dy=10*t','y(0)=1,Dy(1)=2','t');
y=dsolve('-D2y-1/10*Dy-101/100*y=-5/10*cos(t/4)','y(0)=0,y(10)=-1/2','t');
%y=dsolve('-D2y+2*Dy-y=-t^2+1','y(0)=5,y(1)=10','t');
%y=dsolve('-D2y+y=t','y(0)=1,y(1)=2','t');
```

```

%y=dsolve('-D2y=t','y(0)=1,y(1)=2','t');
%y=dsolve('-D2y-y/4=-t/4','y(0)=4,y(5)=13.5','t');
%y=dsolve('-D2y+3*Dy-2*y=0','y(0)=0,y(1)=10','t');
%t=0:0.01:1;

t=a+h:h:b-h;
z=a+h:h:b-h;
w=eval(vectorize(y));

plot(t,w,'r',z,u)
legend('Solución Exacta','Solución Aproximada')
grid on
xlabel('Discretización (x)')
ylabel('u(x)')

```

Algoritmo para resolver Problema 5

```

function [u]=df2(a,b,f,n)
syms x y;
h=(b-a)/(n+1);

%Matriz de diferencias
K=zeros(n^2,1);
A=mlap(n,h);

for i=1:n^2;
K(i)=subs(f,[x,y],[a+i*h,a+i*h]);
end

u=A\K;
surf(reshape(u,n,n));
xlabel(' (x) ')
ylabel(' (y) ')
zlabel('u(x,y)')
end

```

Algoritmo para resolver Problema 6

```

function [u]=df2nl(a,b,f,n)
syms x y;
tol=1e-6;
ep=10^-2;

h=(b-a)/(n+1);
d=ones(n^2,1);
u=zeros(n^2,1);
K=ones(n^2,1);

```

```

A=mlap(n,h);

for i=1:n^2;
K(i)=subs(f,[x,y],[a+i*h,a+i*h]);
end

while norm(d)>tol
    F=(-ep*A*u)+u.^3-K;
    Fp=(-ep*A)+3*diag(u.^2);

    d=-Fp\F;
    u=u+d;
end

surf(reshape(u,n,n));
xlabel('x')
ylabel('y')
zlabel('u(x,y)')
end

```

Pkatym

Algoritmo para resolver Problema 7

```

function[u,iter]=Pkatym(n,l,ep,f)
%l=10^2;
%ep=1e-2;

ep1=1e-6;
syms x;
tol=1e-6;
h=1/(n+1);
d=ones(n,1);
e=ones(n,1);

A=(1/(h^2))*spdiags([e -2*e e],[-1:1,n,n]);

%Diferencia hacia atras
D=(1/h)*spdiags([-e e],[-1:0,n,n]);

%Diferencia hacia adelante
D2=-D';

Y=zeros(n,1);
u=ones(n,1);
q=ones(n,1);
uq=[u;q];

for i=1:n

```

```

    Y(i)=subs(f,x,i*h);
    z(i)=i*h;
end

iter=1;
while norm(d)>tol
    R=(ep^2.+(D2*u).^2).^ (1/2);

    B=diag(min(max(-1,q),1))*diag((D2*u)./R)*D2;
    F=[(-ep1*A*u)-(D*q)+1*(u-Y) ; R.*q-D2*u];
    Fp=[(-ep1*A)+1*eye(n) -D;B-D2 diag(R)];

    d=-Fp\F;
    uq=uq+d;
    u=uq(1:n);
    q=uq(n+1:2*n);
    iter=iter+1;
end

plot(z,Y,'r',z,u)
legend('función f','aprox sin ruido')
end

```

B.1.2. Algoritmos para el procesamiento de imágenes

Algoritmo para la eliminación de ruido Problema 8.

```

function[u,iter]=TV_denoised(I,l,ep)
A1=imread(I);
if(isa(A1,'uint8'))
    A1=rgb2gray(A1);
    A1=im2double(A1);
else
    A1=im2double(A1);
end
[m,n]=size(A1);
rg=imnoise(A1,'gaussian',0,0.01);
imshow(A1)
dim=n*n;

tol=1e-6;
d=ones(3*dim,1);
u=zeros(dim,1);
q1=zeros(dim,1);
q2=zeros(dim,1);

uq=[u;q1;q2];

```

```

h1=1/(m+1);
h2=1/(n+1);
[Dxt,Dxa,Dyt,Dya]=mdiv(n,m,h1,h2);

K=reshape(rg,[dim 1]);
iter=0;

while norm(d)>tol
    R=(ep^2.+( (Dxa*u).^2+(Dya*u).^2)).^(1/2);
    nq=((q1.^2)+(q2.^2)).^(1/2);
    qmax=max(1,nq);
    W=(spdiags((Dxa*u./R),0,dim,dim)*Dxa)+(spdiags((Dya*u./R),0,dim,dim)*Dya);
    B1=spdiags((q1./qmax),0,dim,dim)*W;
    B2=spdiags((q2./qmax),0,dim,dim)*W;

    F=[-(Dxt*q1)-(Dyt*q2)+1*(u-K);(R.*q1)-(Dxa*u);(R.*q2)-(Dya*u)];
    Fp=[1*speye(dim) -Dxt -Dyt;B1-Dxa sparse(dim,dim);B2-Dya sparse(dim,

    d=-Fp\F;
    norm(d);
    uq=uq+d;
    u=uq(1:dim);
    q1=uq(dim+1:2*(dim));
    q2=uq(2*(dim)+1:3*(dim));
    iter=iter+1;
end

subplot(1,2,1)
    imshow(reshape(Y,n,n))
    subplot(1,2,2)
    imshow(reshape(u,n,n))
end

```

Algoritmo para inpainting utilizando la regularización de Huber.

```

function [uk,iter]=TV_inpainting(n,m,I,P,l,gamma1)
%parametros
%l=10^6;
%gamma1=10;

tol=1e-6;

%Matriz de discretizacion
h1=1/(m+1);
h2=1/(n+1);
d1=n*m;
[D1,G1,D2,G2]=mdiv(n,m,h1,h2);

%Inicializacion

```



```

K=reshape(I,[d1 1]);
D=reshape(P,[d1 1]);
M = [];
M(:,1)=K;

    iter=0;
    uk = zeros(n^2,1);
    pk = sparse(d1,1);
    qk = sparse(d1,1);
    zk = sparse(d1+2*d1,1);
    C = sparse(d1,d1);
    Wp = sparse(d1,1);

F = @(u,q,p) ([-(D1*q)-(D2*p)+1*(D.*(u-K));
    max(1,gamma1*sqrt((G1*u).^2+(G2*u).^2)).*q-(gamma1*speye(d1,d1)*G1*u);
    max(1,gamma1*sqrt((G1*u).^2+(G2*u).^2)).*p-(gamma1*speye(d1,d1)*G2*u)]);

tic;
while norm(F(uk,qk,pk))>tol
%while norm(d)>tol
    W1=gamma1*sqrt((G1*uk).^2+(G2*uk).^2);
    W = max(1,W1);
    T=sqrt((G1*uk).^2+(G2*uk).^2);
    v1 =spdiags(gamma1*(G1*uk)./T,0,d1,d1);
    v2 =spdiags(gamma1*(G2*uk)./T,0,d1,d1);
    for i=1:size(W1);
        if (W1(i)<1)
            Wp(i)=0;
        else
            Wp(i)=1;
        end
    end
    end

    Wp=spdiags(Wp,0,d1,d1);

    norma=max(1,sqrt(qk.^2+pk.^2));
    norma=1./norma;
    norma=spdiags(norma,0,d1,d1);

    B1 = -gamma1*speye(d1,d1)*G1...
    +Wp*(v1*spdiags(norma*qk,0,d1,d1)*G1...
    +v2*spdiags(norma*qk,0,d1,d1)*G2);
    B2 = -gamma1*speye(d1,d1)*G2...
    +Wp*(v1*spdiags(norma*pk,0,d1,d1)*G1...
    +v2*spdiags(norma*pk,0,d1,d1)*G2);

    E = spdiags(W,0,d1,d1);

    Fp = [+1*spdiags(D,0,d1,d1), -D1, -D2;B1,E,C;B2, C, E];
    Fe = F(uk,qk,pk);
    norm(Fe);

```

```

        d=-Fp\Fe;
        zk = zk+d;
        uk = zk(1:n^2);
        qk = zk(n^2+1:n^2+d1);
        pk = zk(n^2+d1+1:end);
        iter=iter+1;
        M(:,iter+1) = uk;
end
toc;
t=toc/60
% %Evolucion en el tiempo
% [~,b]=size(M);
% for i=1:b
%     L = reshape(M(:,i),n,m);
%     imshow(L);
%     pause
% end
subplot(2,2,1)
    imshow(reshape(K,n,m))
    subplot(2,2,2)
imshow(reshape(uk,n,m))
    subplot(2,2,3)
    imshow(reshape(D,n,m))
end

```

Algoritmo para inpainting utilizando la regularización de Berkovier Engelman.

```

function [u, iter]=TV(n,m,I,P,lambda,ep)
%parametros
%lambda=10e6;
%ep=1e-4;

tol=1e-6;

%Matriz de discretizacion
h1=1/(m+1);
h2=1/(n+1);
dim=m*n;
[Dxt,Dxa,Dyt,Dya]=mdiv(n,m,h1,h2);

%Inicializacion
K=reshape(I,[dim 1]);
D=reshape(P,[dim 1]);
u=zeros(dim,1);
q1=zeros(dim,1);
q2=zeros(dim,1);
uq=[u;q1;q2];
%d=ones(3*dim,1);
iter=0;

```

```

U=[];
k=1;
U(:,k)=K;

F = @(u,q,p) ([lambda*(D.*(u-K))-(Dxt*q+Dyt*p); ...
    sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).*q-Dxa*u; ...
    sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).*p-Dya*u]);
tic;

while norm(F(u,q1,q2))> tol
    %while norm(d)>tol
        iter=iter+1;

        R=(ep^2.+((Dxa*u).^2)+(Dya*u).^2)).^(1/2);
        nq=((q1.^2)+(q2.^2)).^(1/2);
        qmax=max(1,nq);
        W=(spdiags((Dxa*u./R),0,dim,dim)*Dxa)...
            +(spdiags((Dya*u./R),0,dim,dim)*Dya);
        B1=spdiags((q1./qmax),0,dim,dim)*W;
        B2=spdiags((q2./qmax),0,dim,dim)*W;

        Fp=[lambda*spdiags(D,0,dim,dim) -Dxt -Dyt;...
            B1-Dxa spdiags(R,0,dim,dim) sparse(dim,dim);...
            B2-Dya sparse(dim,dim) spdiags(R,0,dim,dim)];
        Fc=F(u,q1,q2);
        norm(Fc)
        d=-Fp\Fc;
        uq=uq+d;
        u=uq(1:dim);
        q1=uq(dim+1:2*(dim));
        q2=uq((2*(dim))+1:3*(dim));
    U(:,k+1)=u;
        k=k+1;
end
toc;
t=toc/60

% % %Evolucion en el tiempo
% [~,b]=size(U);
% for i=1:b
%     L = reshape(U(:,i),n,m);
%     imshow(L);
%     pause
% end

subplot(2,2,1)
    imshow(reshape(K,n,m))
    subplot(2,2,2)
imshow(reshape(u,n,m))
    subplot(2,2,3)
    imshow(reshape(D,n,m))

```

```
end
```

Algoritmo de reacción-difusión anisotrópica utilizando la regularización de Huber.

```
function [uk,iter,L]=TV_int(n,m,I,P,Ao,l)

toll=10^-3;
gamma1=100;
r=1/2;

%Matrices de diferencias
h1=1/(m+1);
h2=1/(n+1);
d1=n*m;
[D1,G1,D2,G2]=mdiv(n,m,h1,h2);

%Dominio perdido rectangulo mxp
K=reshape(I,[d1 1]);
D=reshape(P,[d1 1]);
f=reshape(Ao,[d1 1]);
M = [];
M(:,1)=K;
N=100*ones(n^2,1);

iter=0;
uk = zeros(n^2,1);
pk = sparse(d1,1);
qk = sparse(d1,1);
zk = sparse(n^2+2*d1,1);
C = sparse(d1,d1);
Wp = sparse(d1,1);

F = @(u,q,p,d) ([u-d-r*((D1*q)+(D2*p))-l*(D.*(u-K))];
max(1,gamma1*sqrt((G1*u).^2+(G2*u).^2)).*q-(gamma1*speye(d1,d1)*G1*u);
max(1,gamma1*sqrt((G1*u).^2+(G2*u).^2)).*p-(gamma1*speye(d1,d1)*G2*u)];

l=1;

while norm(M(:,1)-N)>toll

    dk=M(:,1);
    tol=1/l;
while norm(F(uk,qk,pk,dk))>tol
    R=sqrt((G1*uk).^2+(G2*uk).^2);
    Wl=gamma1*R;
    W = max(1,Wl);
    nq=((qk.^2)+(pk.^2)).^(1/2);
```

```

qmax=max(1,nq);
B=(spdiags((G1*uk)./R,0,d1,d1)*G1)...
  +(spdiags((G2*uk)./R,0,d1,d1)*G2);

for i=1:d1;
    if (W1(i)<1)
        Wp(i)=0;
    else
        Wp(i)=1;
    end
end
Wp=spdiags(Wp,0,d1,d1);

B1=gamma1*Wp*spdiags((qk./qmax),0,n^2,n^2)*B;
B2=gamma1*Wp*spdiags((pk./qmax),0,n^2,n^2)*B;
E = spdiags(W,0,d1,d1);

Fp = [speye(d1,d1)+r*(1*spdiags(D,0,d1,d1)),
      -r*D1, -r*D2;B1-(gamma1*speye(d1,d1)*G1),E,C;
      B2-(gamma1*speye(d1,d1)*G2), C, E];
Fe = F(uk,qk,pk,dk);
norm(Fe)
zk = zk-Fp\Fe;
uk = zk(1:n^2);
qk = zk(n^2+1:n^2+d1);
pk = zk(n^2+d1+1:end);
iter=iter+1
end
% iter=0;
M(:,l+1) = uk;
N=M(:,l);
l=l+1
end

[~,b]=size(M);
L=zeros(b,1);
for i=1:b
    L(i) =normaL2(f,M(:,i),1/b);
end
plot(L)
legend('\Delta t=h/5')
grid on
figure
%Evolucion en el tiempo
% for i=1:b
%     L = reshape(M(:,i),n,n);
%     imshow(L);
%     pause
% end
subplot(2,2,1)
imshow(reshape(K,n,m))

```

```

subplot(2,2,2)
imshow(reshape(uk,n,m))
subplot(2,2,3)
imshow(reshape(D,n,m))
end

```

Algoritmo de reacción-difusión anisotrópica utilizando la regularización de Berkovier Engelman.

```

function [u,iter]=TV_itc(n,m,F,D,lambda,ep)
%parametros
%lambda=10e6;
%ep=1e-4;

tol=1e-6;

%Matriz de discretizacion
dim=m*n;
h1=1/(m+1);
h2=1/(n+1);
[Dxt,Dxa,Dyt,Dya]=mdiv(n,m,h1,h2);

%Inicializacion
K=reshape(F,[dim 1]);
D=reshape(D,[dim 1]);
u=zeros(dim,1);
q1=zeros(dim,1);
q2=zeros(dim,1);
uq=[u;q1;q2];
iter=0;
U=[];
N=100*ones(dim,1);
%U(:,1)=sparse(dim,1);
U(:,1)=K;
k=1;

%Discretización variable temporal
r=1/2;

F = @(u,q,p,v) ([u-v-r*(-lambda*(D.*(u-K))+(Dxt*q+Dyt*p)); ...
    sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).*q-Dxa*u; ...
    sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).*p-Dya*u]);

tic;
while norm(U(:,k)-N)>tol

    v=U(:,k);
    tol=1/k;

```

```

while norm(F(u,q1,q2,v)) > tol

    iter=iter+1;

    R=(ep^2.+((Dxa*u).^2)+((Dya*u).^2)).^(1/2);
    nq=((q1.^2)+(q2.^2)).^(1/2);
    qmax=max(1,nq);
    W=(spdiags((Dxa*u./R),0,dim,dim)*Dxa)...
        +(spdiags((Dya*u./R),0,dim,dim)*Dya);
    B1=spdiags((q1./qmax),0,dim,dim)*W;
    B2=spdiags((q2./qmax),0,dim,dim)*W;

    Fp=[speye(dim,dim)-r*(-lambda*spdiags(D,0,dim,dim)) -r*Dxt -r*Dyt;...
        B1-Dxa spdiags(R,0,dim,dim) sparse(dim,dim);...
        B2-Dya sparse(dim,dim) spdiags(R,0,dim,dim)];
    Fc=F(u,q1,q2,v);
    norm(Fc)
    d=-Fp\Fc;
    uq=uq+d;
    u=uq(1:dim);
    q1=uq(dim+1:2*(dim));
    q2=uq((2*(dim))+1:3*(dim));

end

U(:,k+1)=u;
N=U(:,k);
k=k+1
end
toc;
t=toc/60
subplot(2,2,1)
imshow(reshape(K,n,m))
subplot(2,2,2)
imshow(reshape(u,n,m))
subplot(2,2,3)
imshow(reshape(D,n,m))

% %Evolucion en el tiempo
% [~,b]=size(U);
% for i=1:b
%     L = reshape(U(:,i),n,m);
%     imshow(L);
%     pause
end

```

Algoritmo de curvatura media.

```
function [u,iter]=curvaturam(n,m,I,P,l,ep)
%parametros
% l=10^5;
% ep=1e-3;

tol=1e-3;

%Matriz de discretizacion
h1=1/(m+1);
h2=1/(n+1);
dim=m*n;
[Dxt,Dxa,Dyt,Dya]=mdiv(n,m,h1,h2);

%Inicializacion
u=zeros(dim,1);
q1=zeros(dim,1);
q2=zeros(dim,1);
uq=[u;q1;q2];
K=reshape(I,[dim 1]);
D=reshape(P,[dim 1]);

iter=1;
M = [];
M(:,iter)=K;

Fe = @(u,q,p) ([ (sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).* (Dxt*q)) ...
    +(sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).* (Dyt*p))-1*(D.*(u-K)) ;
    sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).*q-Dxa*u;
    sqrt((Dxa*u).^2+(Dya*u).^2+ep^2).*p-Dya*u] );
tic;

while norm(Fe(u,q1,q2))>tol
    R=(ep^2.+((Dxa*u).^2+(Dya*u).^2)).^(1/2);
    nq=((q1.^2)+(q2.^2)).^(1/2);
    qmax=max(1,nq);
    W=(spdiags((Dxa*u./R),0,n^2,n^2)*Dxa)...
        +(spdiags((Dya*u./R),0,n^2,n^2)*Dya);
    B1=spdiags((q1./qmax),0,n^2,n^2)*W;
    B2=spdiags((q2./qmax),0,n^2,n^2)*W;
    C1=spdiags(Dxt*(q1),0,n^2,n^2)*W;
    C2=spdiags(Dyt*(q2),0,n^2,n^2)*W;

    Fp=[-1*spdiags(D,0,n^2,n^2)+C1+C2,
        spdiags(R,0,n^2,n^2)*Dxt spdiags(R,0,n^2,n^2)*Dyt;
        B1-Dxa spdiags(R,0,n^2,n^2) sparse(n^2,n^2);
        B2-Dya sparse(n^2,n^2) spdiags(R,0,n^2,n^2)];
    F=Fe(u,q1,q2);
```



```

d=-Fp\F;
norm(F)
uq=uq+d;
u=uq(1:n^2);
q1=uq(n^2+1:2*(n^2));
q2=uq((2*(n^2))+1:3*(n^2));
iter=iter+1
M(:,iter+1) = u;
%tol=1/iter;
end

% %Evolucion en el tiempo
% [~,b]=size(M);
% for i=1:b
%     L = reshape(M(:,i),n,m);
%     imshow(L);
%     pause
% end
toc;
t=toc/60
subplot(2,2,1)
    imshow(reshape(K,n,m))
subplot(2,2,2)
    imshow(reshape(u,n,m))
subplot(2,2,3)
    imshow(reshape(D,n,m))
end

```

Algoritmo movimiento de la curvatura media.

```

function [uk,iter]=curvaturam_t(n,m,I,P,l,ep)
%parametros
%l=10^4;
%ep=1e-4;

%Matriz de discretizacion
h1=1/(m+1);
h2=1/(n+1);
dim=m*n;
[D1,G1,D2,G2]=mdiv(n,m,h1,h2);
r=1/2;

K=reshape(I,[dim l]);
D=reshape(P,[dim l]);
iter=0;
M = [];
N=100*K;
M(:,1)=K;

```

```

uk = zeros(dim,1);
pk = sparse(dim,1);
qk = sparse(dim,1);
zk = sparse(3*dim,1);

F = @(u,q,p,t) ([u-t-r*(sqrt((G1*u).^2+(G2*u).^2)...
.*(D1*q)+sqrt((G1*u).^2+(G2*u).^2).*(D2*p)-1*(D.*(u-K))));
sqrt((G1*u).^2+(G2*u).^2+ep^2).*q-G1*u;
sqrt((G1*u).^2+(G2*u).^2+ep^2).*p-G2*u]);

j=1;
tol=1;
tic;
while norm(M(:,j)-N)>tol

    tk=M(:,j);
    tol=1/j;
while norm(F(uk,qk,pk,tk))>tol
    W = ((G1*uk).^2+(G2*uk).^2+ep^2).^(1/2);
    nor=(qk.^2+pk.^2).^(1/2);
    qmax=max(1,nor);
    C=(spdiags((G1*uk)./W,0,dim,dim)*G1)...
      +(spdiags((G2*uk)./W,0,dim,dim)*G2);
    B1 =spdiags(qk./qmax,0,dim,dim)*C;
    B2 =spdiags(pk./qmax,0,dim,dim)*C;
    C1=spdiags(D1*qk,0,dim,dim)*C;
    C2=spdiags(D2*pk,0,dim,dim)*C;
    E = spdiags(W,0,dim,dim);

    Fp = [speye(dim,dim)-r*C1-r*C2+r*I*spdiags(D,0,dim,dim),
          -r*E*D1, -r*E*D2;
          B1-G1,E,sparse(dim,dim);B2-G2, sparse(dim,dim), E];
    Fe = F(uk,qk,pk,tk);
    d=-Fp\Fe;
    norm(Fe)
    zk = zk+d;
    uk = zk(1:dim);
    qk = zk(dim+1:2*dim);
    pk = zk((2*dim)+1:3*dim);

    iter=iter+1;
end
M(:,j+1) = uk;
N=M(:,j);
j=j+1

end
toc;
t=toc/60
% Evolucion en el tiempo
% [~,b]=size(M);
% for i=1:b
%     L = reshape(M(:,i),n,n);

```

```
%     imshow(L);
%     pause
%     end
subplot(2,2,1)
    imshow(reshape(K,n,m))
    subplot(2,2,2)
imshow(reshape(uk,n,m))
    subplot(2,2,3)
    imshow(reshape(D,n,m))
end
```

Bibliografía

- [1] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer, 2006.
- [2] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–355. IEEE, 2001.
- [3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press / Addison-Wesley Publishing Co., 2000.
- [4] Andrea L Bertozzi, Selim Esedoglu, and Alan Gillette. Inpainting of binary images using the cahn-hilliard equation. *IEEE Transactions on image processing*, 16(1):285–291, 2007.
- [5] John W Cahn and John E Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958.
- [6] Tony F Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM journal on applied mathematics*, 66(5):1632–1648, 2006.
- [7] Tony F Chan, Sung Ha Kang, and Jianhong Shen. Euler’s elastica and curvature-based inpainting. *SIAM Journal on Applied Mathematics*, pages 564–592, 2002.
- [8] Tony F Chan and Pep Mulet. *Iterative methods for total variation image restoration*. Citeseer, 1996.
- [9] Tony F Chan and Pep Mulet. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM journal on numerical analysis*, 36(2):354–367, 1999.
- [10] Tony F Chan, Stanley Osher, and Jianhong Shen. The digital tv filter and nonlinear denoising. *Image Processing, IEEE Transactions on*, 10(2):231–241, 2001.
- [11] Tony F Chan and Jianhong Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.

- [12] Tony F Chan and Jianhong Jackie Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. Siam, 2005.
- [13] Juan Carlos De los Reyes. *Numerical PDE-Constrained Optimization*. Springer, 2015.
- [14] Juan Carlos De Los Reyes and Sergio González-Andrade. Numerical simulation of thermally convective viscoplastic fluids by semismooth second order type methods. *Journal of Non-Newtonian Fluid Mechanics*, 193:43–48, 2013.
- [15] Ronald A DeVore, Björn Jawerth, and Bradley J Lucier. Image compression through wavelet transform coding. *Information Theory, IEEE Transactions on*, 38(2):719–746, 1992.
- [16] Selim Esedoglu and Jianhong Shen. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics*, 13(04):353–370, 2002.
- [17] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.
- [18] Pascal Getreuer. Total variation inpainting using split bregman. *Image Processing 372 Online 2012*, <http://dx.doi.org/10.5201/ipol.2012.g-tvi>, 373, 2012.
- [19] Rafael C Gonzalez and Richard E Woods. *Digital image processing*, 2002.
- [20] A Haddad. Stability in a class of variational methods. *Applied and Computational Harmonic Analysis*, 23(1):57–73, 2007.
- [21] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, volume 4. Siam, 1998.
- [22] Michael Hintermüller and Georg Stadler. An infeasible primal-dual algorithm for total bounded variation–based inf-convolution-type image restoration. *SIAM Journal on Scientific Computing*, 28(1):1–23, 2006.
- [23] Bernd Jähne. *Digital image processing: electronic version of the book, exercises, additional images and runtime version of the heurisko image processing software*. Springer, 2002.
- [24] Anil K Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [25] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.
- [26] Stan Z Li. *Markov random field modeling in computer vision*. Springer-Verlag New York, Inc., 1995.
- [27] De Los Reyes, Juan Carlos, et al. Path following methods for steady laminar bingham flow in cylindrical pipes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(01):81–117, 2009.
- [28] JBA Maintz. *Digital and medical image processing. Lecture Notes, Utrecht University*, 2002.

- [29] Antonio Marquina and Stanley Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM Journal on Scientific Computing*, 22(2):387–405, 2000.
- [30] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.
- [31] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.
- [32] JN Reddy and ML Rasmussen. *Análisis Matemático Avanzado*. Limusa, 1992.
- [33] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.
- [34] Jianhong Shen and Tony F Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.
- [35] Bart M ter Haar Romeny. *Geometry-driven diffusion in computer vision*. Kluwer academic Norwell, MA, 1994.
- [36] Andreï Tikhonov. *Solutions of ill-posed problems*.
- [37] Andy Tsai, Anthony Yezzi Jr, and Alan S Willsky. Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. *Image Processing, IEEE Transactions on*, 10(8):1169–1186, 2001.
- [38] Curtis R Vogel and Mary E Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *Image Processing, IEEE Transactions on*, 7(6):813–824, 1998.