



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

RECONSTRUCCIÓN CONJUNTA DE IMÁGENES DE RESONANCIA MAGNÉTICA Y DE TOMOGRAFÍA POR EMISIÓN DE POSITRONES USANDO DATOS SINTÉTICOS

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE MATEMÁTICO**

PEDRO XAVIER GALINDO VERA

pedro.galindo@epn.edu.ec

DIRECTOR: EVELYN GABRIELA CUEVA JARAMILLO, PHD

evelyn.cuevaj@epn.edu.ec

DMQ, AGOSTO 2023

CERTIFICACIONES

Yo, PEDRO XAVIER GALINDO VERA, declaro que el trabajo de integración curricular aquí descrito 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.

Pedro Xavier Galindo Vera

Certifico que el presente trabajo de integración curricular fue desarrollado por Pedro Xavier Galindo Vera, bajo mi supervisión.

Evelyn Gabriela Cueva Jaramillo, PhD
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Pedro Xavier Galindo Vera

Evelyn Gabriela Cueva Jaramillo, PhD

a

Guadalupe y Manuel

Índice general

Índice de figuras	XII
Resumen	XV
Abstract	XVII
1. Introducción a los problemas inversos	1
1.1. Dualidad de problemas en matemática	2
1.2. Problemas duales	3
1.3. Problemas inversos	3
1.4. Definiciones fundamentales	5
1.5. Ejemplos	6
1.5.1. Problemas inversos en matemática	6
1.5.2. Aplicación de problemas inversos	7
1.5.3. Inversión de imágenes	8
1.6. Definición matemática de imagen	9
2. Problemas inversos lineales	13
2.1. Medida lineal indirecta	13
2.2. Problemas mal puestos	14
2.3. Par Convolución–Deconvolución	15

2.3.1. Formulación lineal inversa	17
2.3.2. Modelo discreto	18
2.4. Tomografía Computarizada de Rayos X (CT)	19
2.4.1. Modelo determinista	20
2.4.2. Formulación lineal inversa	22
2.4.3. Modelo discreto	23
2.5. Transformadas de Fourier y de Radon	26
2.6. Tomografía por Emisión de Positrones	31
2.6.1. Modelo probabilístico	34
2.6.2. Formulación lineal inversa	35
2.6.3. Datos sintéticos	38
2.7. Imagen por Resonancia Magnética (MRI)	39
2.7.1. Modelo dinámico	39
2.7.2. Formulación lineal inversa	41
2.7.3. Datos sintéticos	42
3. Inversión Regularizada	45
3.1. Descomposición en valores singulares (SVD)	47
3.2. Descomposición en valores singulares truncada (TSVD)	50
3.3. Regularización de Tikhonov	52
3.4. Regularización de Variación Total	54
3.5. Variación Total Conjunta	59
3.6. Conjuntos de Nivel Paralelos	60
4. Reconstrucción conjunta de imágenes	69
4.1. Modelo probabilístico para la reconstrucción conjunta	70
4.2. Optimización numérica	73
4.2.1. L-BFGS	74
4.2.2. L-BFGS-B	76

4.3. Experimentación numérica	77
4.4. Evaluación visual	85
5. Conclusiones Generales	87
A. Códigos MATLAB	95
A.1. tic	95
A.2. JR	102
Bibliografía	121

Índice de figuras

1.1. Inversión de color para una imagen en escala de grises.	8
1.2. Interpretación para una imagen digital.	10
2.1. Desenfoque o convolución de un pulso o señal.	16
2.2. Efecto de la transformada de Radon.	22
2.3. Escaneo CT por haces paralelos.	23
2.4. Geometría de la trayectoria recta de un haz de Rayos X.	24
2.5. Discretización CT empleando geometría de haces paralelos.	26
2.6. Tipos de eventos en PET.	33
2.7. Esquema de escaner de anillo PET.	36
2.8. Proceso de generación de datos sintéticos PET.	38
2.9. Datos sintéticos MRI.	43
3.1. Discretización de la convolución o desenfoque.	45
3.2. Convolución con y sin ruido, e inversión ingenua.	47
3.3. Comportamiento de los valores singulares de una aproximación discretizada de la convolución.	50
3.4. Reconstrucción TSVD de la convolución.	51
3.5. Reconstrucción Tikhonov de la convolución.	53
3.6. Reconstrucciones de la convolución empleando factorización matricial a partir de SVD.	54

3.7. Error relativo variando el parámetro de regularización y reconstrucciones TSVD, Tikhonov y TV de la convolución. . . .	56
3.8. Convolución con y sin ruido, y deconvolución de la fotografía de un camarógrafo empleando reconstrucciones por: inversión ingenua, TSVD, Tikhonov y TV.	57
3.9. Representación gráfica de la variación total para imágenes. . .	58
3.10. Representación gráfica de la “similitud” entre imágenes. . . .	68
4.1. Diagrama del modelo de reconstrucción conjunta PET/MRI.	70
4.2. Reconstrucción conjunta PET/MRI muestra: full	79
4.3. Reconstrucción conjunta PET/MRI muestra: lines2	80
4.4. Reconstrucción conjunta PET/MRI muestra: radial15	81
4.5. Reconstrucción conjunta PET/MRI muestra: radial20	82
4.6. Reconstrucción conjunta PET/MRI muestra: spiralHigh . . .	83
4.7. Reconstrucción conjunta PET/MRI muestra: spiralUni . . .	84

RESUMEN

Muchas veces la única forma de medir un fenómeno físico, es hacerlo indirectamente a través de una variable distinta a la de interés, si además las medidas resultan de una muestra finita y pretendemos determinar una variable continua, entonces estamos ante un *problema de medida indirecta*. Su formulación matemática corresponde a lo que se denomina *problemas inversos lineales* y su solución requiere de herramientas como el cálculo variacional y la optimización numérica. Este es justamente el caso de la reconstrucción de imágenes médicas, cuyo origen se encuentra en la tomografía computarizada de Rayos X y que ha evolucionado durante el último medio siglo aprovechando diferentes fenómenos físicos. Al día de hoy, por ejemplo, los escáneres son capaces de realizar simultáneamente tanto *tomografía por emisión de positrones (PET)* e *imagen por resonancia magnética (MRI)*. El comportamiento aleatorio en PET, determina que sus imágenes sean poco definidas, mientras que las imágenes MRI gozan de muy buenas reconstrucciones. Se aprovecha la similitud estructural de ambas modalidades de imagen médica para que las reconstrucciones se beneficien una de la otra. Para ello se aplica un modelo estadístico de estimación paramétrica de *máxima verosimilitud conjunta (JMAP)* que emplea *regularizadores* conocidos a priori, y cuya solución se aproxima con el *Algoritmo L-BFGS-B*. El presente trabajo expone el marco teórico, las herramientas matemáticas y la implementación computacional necesarias para la *reconstrucción conjunta PET/MRI* de datos sintéticos simulados a partir de imágenes de prueba generadas por computadora; llamadas *fantomas*.

Palabras clave: reconstrucción conjunta, imágenes, PET, MRI, problemas inversos, inversión regularizada, TSVD, Tikhonov, TV, JTV, QPLS, LPLS, estimador de máxima verosimilitud conjunta, JMAP, L-BFGS-B.

ABSTRACT

Often, the only way to measure a physical phenomenon is indirectly through a variable other than the one of interest. If, in addition, the measurements result from a finite sampling and we intend to determine a continuous variable, then we are faced with an *indirect measurement problem*. Its mathematical formulation corresponds to what is called *linear inverse problems*, and its solution requires tools such as variational calculus and numerical optimization. An example is medical image reconstruction, which originated in X-ray computed tomography and has evolved by taking advantage of different physical phenomena over the last half-century. For instance, today, scanners are capable of simultaneously performing both *positron emission tomography (PET)* and *magnetic resonance imaging (MRI)*. The random behavior in PET determines that its images are poorly defined, while MRI images enjoy excellent reconstructions. The structural similarity of both medical imaging modalities helps the reconstructions to benefit from each other. To do this, we will use a statistical model of *joint maximum a posteriori (JMAP)* estimation that takes advantage of known a priori *regularizers* and whose solution is approximated with the *L-BFGS-B Algorithm*. This work presents the theoretical framework, mathematical tools, and computational implementation necessary for the *joint PET/MRI reconstruction* of synthetic data simulated from computer-generated test images called *phantoms*.

Keywords: joint reconstruction, images, PET, MRI, inverse problems, regularized inversion, TSVD, Tikhonov, TV, JTV, QPLS, LPLS, joint maximum a posteriori estimator, JMAP, L-BFGS-B.

Capítulo 1

Introducción a los problemas inversos

Conforme se avanza en la carrera de Matemática aprendemos a usar el método deductivo demostrando teoremas a partir de otros teoremas previos, de axiomas, de términos no definidos, y de la lógica, es decir, reconstruimos una y otra vez las teorías matemáticas. Por otra parte, vamos desarrollando poco a poco esa intuición matemática que nos permite identificar patrones [12] y esperamos poder plantear nuestros propios resultados que nos permitan reflejar, generalizar o preservar características que nos interesan en nuestras aplicaciones. En este quehacer científico subyace una determinación ambigua de problemas [7]. Demostrar resultados en el marco de una teoría matemática pareciera ser un problema de una naturaleza diferente, al de establecer los axiomas que fundamentan la misma teoría partiendo de propiedades que deseamos conservar en un dominio de interpretación [37]. De forma similar, aplicar un teorema luego de verificar que se cumplen sus hipótesis es un problema de una clase distinta al motivado por la intuición que nos lleva a plantear conjeturas a partir de la identificación de patrones que pueden o no llevarnos a un teorema.

Lejos ya de la disputa fundamentalista y epistemológica entre lógicistas al mando de Bertrand Russell, intuicionistas liderados por Henri Poincaré y formalistas encabezados por David Hilbert, hoy conjugamos lo mejor de las escuelas matemáticas esperando confiados en que cada avance de la matemática constituye en mejoras pequeñas o sustanciales

para la calidad de vida de las personas. Luego de estas breves ideas motivadas por las obras de Bertrand Russell [48], Morris Kline [25], Keith Devlin [12], Douglas Hofstadter [22], Benoît Mandelbrot [32][33], David Richeson [47], Jesús Mosterín [38], y la novela gráfica “Logicomix” del escritor y matemático griego Apostolos Doxiadis [13], se invita literal y metafóricamente hablando a adentrarnos en las entrañas matemáticas de un caso de la reconstrucción de imágenes con aplicaciones médicas.

1.1. Dualidad de problemas en matemática

Los problemas de demostrar teoremas como el de aplicarlos, señalan una “dirección” del razonamiento. En el primer caso partimos de una teoría en donde se prueba la validez de un teorema, en el segundo caso verificamos las hipótesis del teorema para luego aplicarlo, y es esta dirección del razonamiento la que motiva llamarlos, “directos”. Por otro lado, en los problemas en los que se busca fijar los axiomas para una teoría, o respectivamente se busca la conjetura que antecede a un teorema, empleamos el adjetivo “inverso”. En este punto, es inevitable pensar que los problemas inversos pueden conllevar una dificultad mayor que los directos. Axiomatizar una teoría es mucho más complejo que demostrar un resultado en virtud de la axiomatización dada, pero además pueden haber diferentes axiomatizaciones para una misma teoría. Un ejemplo clásico es el de la geometría euclidiana y su quinto postulado [25]. Otro ejemplo son las diferentes axiomatizaciones para la teoría de conjuntos (ZF, NBG) [44]; solo por citar dos ocurrencias bien conocidas de este fenómeno.

Un ejemplo propuesto en [34] refleja de manera elegante la idea que intentamos transmitir: “Un problema directo extremadamente sencillo, de Aritmética Elemental, consiste en sumar dos números primos impares. El resultado obviamente es un número par, pero, ¿qué sucede con el problema inverso? ¿Es posible descomponer un número par como suma de dos números primos? Esta pregunta dio origen a la famosa ‘conjetura de Goldbach’ (postulada en 1742), cuya solución más de 250 años después sigue siendo desconocida.”

Lo primero que podemos observar de estos ejemplos, es que se asocian en parejas y, aún cuando un problema directo tiene una única solución,

no hay garantía de que su problema inverso tenga siquiera una. También puede suceder que exista unicidad o multiplicidad de la solución, incluso que se dé o no un fenómeno deseado llamado “estabilidad” en el que pequeñas perturbaciones de los datos de entrada, apenas afectan la solución. Estas características determinarán si un problema es *bien puesto* o no, como lo formalizaremos más adelante.

1.2. Problemas duales

Determinar una ley que rige un fenómeno y por ende su modelo y parámetros, constituye un problema inverso, mientras, determinar los efectos de la ley conocidas sus variables es, a lo que llamamos, un problema directo. Descubrir, crear, identificar patrones y luego demostrar teoremas son parte del trabajo matemático, justamente la parte creativa de esa labor constituye un problema inverso y la deductiva, uno directo.

Implícitamente como matemáticos transitamos entre plantear y resolver problemas inversos y directos sin detenernos en diferenciarlos. Bertero et al. [7] nos dicen que: “En general, estos problemas están relacionados por una especie de dualidad en el sentido de que un problema puede obtenerse del otro intercambiando el papel de los datos y el de las incógnitas: los datos de un problema son las incógnitas del otro y viceversa. Como consecuencia de esta dualidad, puede parecer arbitrario decidir cuál es el problema directo y cuál es el inverso.” El presente trabajo será una experiencia más en ese sentido, un alarde del método deductivo que tanto apreciamos de nuestra venerada matemática formalista pero intentaremos que sea una experiencia más consciente, puesto resolveremos un problema inverso con fines médicos en el que podremos admirar la creatividad humana empeñada en ver lo que, literalmente, está fuera del alcance de nuestros ojos.

1.3. Problemas inversos

Bertero et al. [7] afirman que el concepto de problema inverso es ambiguo y depende del contexto histórico y matemático; citando a Keller

[23], quien define un *problema inverso* como aquel que usa la solución de otro problema como dato. Por otro lado, Engl et al. [16] y Bertero et al. [7] coinciden en que los problemas directos e inversos se definen en función de las leyes físicas que relacionan las causas y los efectos de un fenómeno, siendo el *problema directo* el que calcula los efectos a partir de las causas y el *problema inverso* el que infiere las causas a partir de los efectos. Para Müller y Siltanen [39]: “Los problemas inversos surgen de la necesidad de interpretar mediciones indirectas e incompletas.” En general estos conceptos dependerán del dominio físico en el que se trabaja, donde a la pareja de dualidad se lo conoce como *par directo-inverso*.

El intento mismo de formalizar el concepto de “problema inverso” a partir de diferentes interpretaciones y conceptualizaciones para “problemas inversos”, constituye en sí mismo, un *problema inverso*. Extendiendo el concepto de adjetivo autológico¹; podría decirse que la definición de problemas inversos es un “problema autológico”, pues se describe a sí mismo. Para el caso mencionado, determinar qué problemas son inversos a partir de su definición, constituye el “problema directo”.

Como aplicación práctica se deduce, que hay dos razones principales para estudiar problemas inversos. Por un lado, se busca conocer los estados o parámetros anteriores de un sistema físico. Por otro, se quiere descubrir cómo influir en un sistema a través de su estado actual o mediante la manipulación de sus parámetros para guiarlo hacia un estado deseado en el futuro. En palabras de Engl et al. [16]: “los problemas inversos se ocupan de determinar las causas de un efecto deseado u observado.”

Siguiendo el planteamiento de Müller y Siltanen [39], si representamos con \mathcal{A} el “vínculo” o “conexión” entre los datos u y las medidas (o solución) f de un problema, o si \mathcal{A} simboliza la ley que explica un fenómeno físico entre la causa y su efecto, podemos representar matemáticamente a los problemas directo e inverso como se muestra en el siguiente cuadro.

Problema Directo	Problema Inverso
$\mathcal{A}(u) = f$	$u = \mathcal{A}^{-1}(f)$
$\mathcal{A}(\text{causa}) = \text{efecto}$	$\text{causa} = \mathcal{A}^{-1}(\text{efecto})$

¹Un adjetivo se denomina *autológico* si la propiedad denotada por el adjetivo se aplica al propio adjetivo [37].

1.4. Definiciones fundamentales

Para fundamentar el empleo de la anterior notación, recurrimos al concepto de *grafo* de la teoría de conjuntos, con la cual podemos concretar las siguientes definiciones.

Definición 1 (Grafo). Sean A y B dos conjuntos cualesquiera, un *grafo* G es un subconjunto arbitrario del producto cartesiano $X \times Y$, es decir

$$G \subseteq X \times Y = \{(x, y) \mid x \in X, y \in Y\}.$$

Definición 2 (Grafo Inverso). Dado un grafo G , su *grafo inverso* G^{-1} se define como el grafo

$$G^{-1} = \{(x, y) \mid (y, x) \in G\} \subseteq Y \times X.$$

Definición 3 (Problema Directo). Sea $P(\cdot, \cdot)$ una función proposicional binaria, es decir, un predicado o propiedad sobre dos variables, x e y , pertenecientes respectivamente a los conjuntos X e Y . Un *problema directo* se define como el grafo

$$\mathcal{A} = \{(x, y) \mid x \in X, y \in Y, P(x, y)\}.$$

Definición 4 (Problema Inverso). Dado un problema directo \mathcal{A} , se define como *problema inverso* a un grafo cualquiera $\text{inv}(\mathcal{A})$ que contenga al grafo inverso \mathcal{A}^{-1} , en notación conjuntista tendremos

$$\mathcal{A}^{-1} \subseteq \text{inv}(\mathcal{A}).$$

Definición 5 (Par Directo–Inverso). También conocido como *par de dualidad*; un *par directo–inverso*, es la pareja formada por un problema directo \mathcal{A} y su inverso $\text{inv}(\mathcal{A})$, es decir la dupla $(\mathcal{A}, \text{inv}(\mathcal{A}))$.

Abusando de la notación se puede recurrir a la de función; en vez de escribir $(x, y) \in \mathcal{A}$ notamos $y = \mathcal{A}(x)$, y leemos: *dado x , determinar $y = \mathcal{A}(x)$* . Si no cabe confusión escribimos $x = \mathcal{A}^{-1}(y)$ en vez de $(y, x) \in \text{inv}(\mathcal{A})$, y leemos: *dado $y = \mathcal{A}(x)$, encontrar x* .

Proposición 1. *Todo problema directo es un problema inverso y viceversa.*

Demostración. Sea \mathcal{A} un problema directo. Como $\mathcal{A}^{-1} \subseteq \mathcal{A}^{-1}$, tomemos como problema inverso $\text{inv}(\mathcal{A}) = \mathcal{A}^{-1}$. Aplicando el grafo inverso en cada lado $\text{inv}(\mathcal{A})^{-1} = (\mathcal{A}^{-1})^{-1}$, luego podemos repetir el argumento y tomar $\text{inv}(\text{inv}(\mathcal{A})) = \text{inv}(\mathcal{A}^{-1}) = (\mathcal{A}^{-1})^{-1} = \mathcal{A}$, así por definición \mathcal{A} es problema inverso de \mathcal{A}^{-1} , o dicho de otra forma, \mathcal{A} es el problema inverso, del problema inverso de \mathcal{A} . \square

Definición 6 (Solución de un Problema Directo o Inverso). Sea \mathcal{A} un problema directo ó un problema inverso, si $(x, y) \in \mathcal{A}$, entonces a x denominamos *dato* y a y *solución* del problema.

La definición planteada es más general que la de función, pues no se exige que \mathcal{A} esté *bien definida* en el sentido de función, sino que admite cualquier propiedad que relacione a la abscisa y a la ordenada. Si interpretamos el conjunto X como el de las causas o datos, y al conjunto Y como el de los efecto, medidas o soluciones, entonces la definición propuesta para “problema directo” no exige que cada abscisa tenga una correspondiente ordena, mientras si admite la multiasignación para las abscisas.

1.5. Ejemplos

1.5.1. Problemas inversos en matemática

A continuación presentamos algunos ejemplos de pares directo–inversos, tomados o inspirados en la bibliografía revisada [24][39].

Ejemplo 1 (Par polinomio–ceros). Determinar un polinomio p de grado n cononidos sus ceros x_1, \dots, x_n , es el problema inverso al problema directo de encontrar los ceros de un polinomio dado, p . Notemos que el problema inverso es simple de resolver, puesto la solución viene dada directamente por $p(x) = c(x - x_1) \dots (x - x_n)$, donde c es una constante arbitraria.

Ejemplo 2 (Par puntos–polinomio de interpolación). Encontrar un polinomio p de grado n que toma los valores dados $y_1, \dots, y_n \in \mathbb{R}$ en los puntos conocidos $x_1, \dots, x_n \in \mathbb{R}$. Este problema es inverso al problema directo de calcular el polinomio dado en los puntos dados x_1, \dots, x_n . A este problema inverso, lo conocemos como el problema de interpolación de Lagrange.

Ejemplo 3 (Par diagonal–valores propios). Sean, A una matriz simétrica real $n \times n$, y $\lambda_1, \dots, \lambda_n \in \mathbb{R}$, determinar una matriz diagonal D tal que $A + D$ tenga los valores propios $\lambda_1, \dots, \lambda_n$. Este problema es inverso al problema directo de calcular los valores propios de la matriz dada $A + D$.

Ejemplo 4 (Par sucesión–términos). Sean a_1, a_2, \dots, a_k los primeros k términos de una sucesión, determinar la ley de formación de la sucesión; es decir, encontrar $(a_n)_{\mathbb{N}}$. El problema directo está dado por evaluar (a_n) de acuerdo a su ley de formación. Para este problema inverso existen múltiples soluciones.

1.5.2. Aplicación de problemas inversos

Hemos mencionado ya algunos ejemplos de problemas inversos en matemática, ahora describimos brevemente algunas aplicaciones en diferentes ámbitos, citados en la literatura revisada [4][7][16][39][40].

Trayectoria de partículas. En la mecánica clásica calculamos la trayectoria de partículas conociendo la ley que las rige, o lo que es lo mismo, conociendo la fuerza que produce el movimiento, y de forma inversa podemos calcular la fuerza necesaria para que el móvil siga una determinada trayectoria.

Diagnóstico de enfermedades. En medicina un ejemplo de problema directo podría ser la identificación de los síntomas asociados con la gripe, para este caso, enumerar los más comunes, como fiebre, tos y dolor de cabeza. Por otro lado, si un paciente presenta fiebre, tos y dolor de cabeza, el problema inverso sería determinar si estos síntomas son indicativos de la gripe u otra enfermedad. En otras palabras, el problema directo implica conocer la enfermedad y determinar sus síntomas, mientras que el problema inverso implica conocer los síntomas e identificar la enfermedad.

Diferenciación. La diferenciación y la integración son operaciones inversas entre sí, lo que las convierte en problemas matemáticos inversos. No está claro cuál de estas operaciones debería considerarse el problema directo y cuál el problema inverso [16]. Por lo tanto, se podría argumentar que la diferenciación es el “problema inverso”, aunque en la mayoría de los cursos de cálculo se enseña primero (con excepción por ejemplo del

“Calculus” de Apostol [2]).

Distribución y densidad de probabilidad. Análogamente, encontrar las funciones de distribución y de densidad de probabilidad una a partir de la otra, son problemas inversos en Teoría de Probabilidades.

Estimadores y estadísticos, conocida la probabilidad. La estimación de estadísticos empleando estimadores a partir de una muestra de una población es un problema inverso que incluye pérdida de información [51], respecto de calcular de forma directa un estadístico sobre la población completa.

Ondas y obstáculos, conociendo la fuente. En la teoría de la dispersión y difracción, el problema directo implica calcular las ondas dispersas o difractadas a partir de las fuentes y obstáculos conocidos. Por otro lado, el problema inverso implica determinar los obstáculos a partir de las fuentes y ondas dispersas o difractadas. Estos problemas inversos son fundamentales para métodos de evaluación no invasiva, como imágenes médicas, que implican, por ejemplo, hacer resonar un objeto con la frecuencia adecuada [7]. Este último ejemplo refleja las intenciones del presente trabajo.

1.5.3. Inversión de imágenes

Un ejemplo muy sencillo e intuitivo de problemas inversos, es el de obtener una fotografía a partir de su negativo, un proceso indispensable en la fotografía analógica llamado “revelado”, que tiene su equivalente en

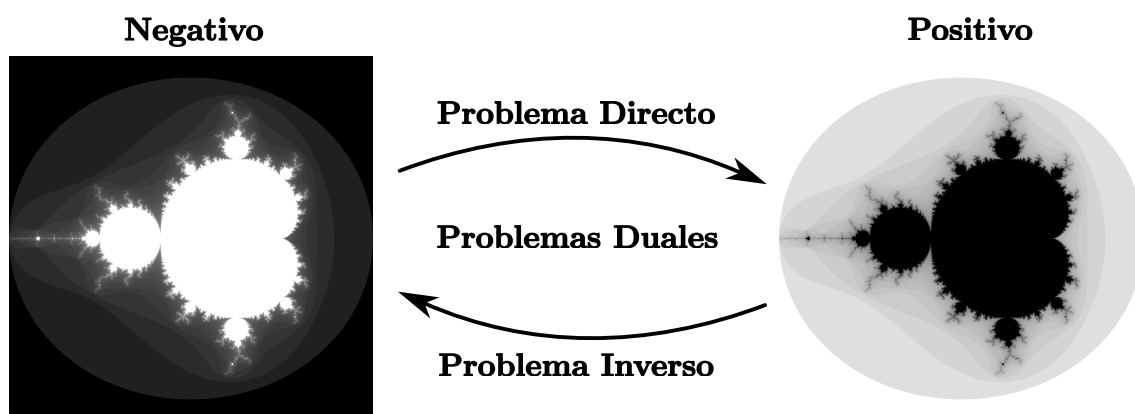


Figura 1.1: Conjunto de Mandelbrot generado en escala de grises y en su negativo, con 100 iteraciones y tonos.

el ámbito digital denominado “inversión” y que consiste en restar de un valor máximo² admisible, el valor numérico de la intensidad de color de cada uno de los píxeles que conforman una imagen. Para este caso el “problema directo” viene dado por el origen físico del problema, en el que se busca obtener a partir de una película en negativo, es decir la “causa”, una fotografía (el positivo) que es el “efecto”. El “problema inverso” resulta entonces, de realizar el proceso contrario, es decir, obtener una imagen negativa de su positivo, lo cual puede ser más complejo en el proceso analógico y de hacerse seguramente conllevará a pérdida en la calidad de la fotografía, en contraposición, el revelado puede repetirse muchas veces a partir del negativo. Por otro lado, invertir el color en una imagen digital es una operación reversible cuantas veces se quiera sin ninguna pérdida de información. Se presenta en la Figura 1.1 las imágenes generadas a partir del conjunto de Mandelbrot³, en el que primero se produjo el negativo a la izquierda y posteriormente se calculó su versión en positivo. Consideremos el conjunto RGB de fotografías digitales en tres capas RGB, es decir, el conjunto de tripletas de matrices $m \times n$ entero no negativas menores o iguales a 255, en este caso tanto el negativo $x = (x_R, x_G, x_B)$, como el positivo $y = (y_R, y_G, y_B)$ de una fotografía digital pertenecen al mismo conjunto, es decir, $x, y \in RGB$. A su vez, $x_R, x_G, x_B, y_R, y_G, y_B \in M^{m \times n}([0, 255] \cap \mathbb{Z})$ con lo que $y = \text{invertir}_{RGB}(x) = 255(I_{m \times n}, I_{m \times n}, I_{m \times n}) - x$, y también $x = \text{invertir}_{RGB}(y) = 255(I_{m \times n}, I_{m \times n}, I_{m \times n}) - y$. Lo que implica que el problema inverso está dado por el mismo operador que el problema directo.

1.6. Definición matemática de imagen

En la sección anterior se presentó la “inversión de imágenes” como un ejemplo muy sencillo de problema inverso sin dar una definición formal de imagen. Puesto que el presente es un trabajo sobre imágenes, cabe motivar y definir el concepto matemático para ello. Una imagen digital contiene la información de las tonalidades en cada píxel que la conforman, la cual se almacena a través de una matriz de valores numéricos. Los valores admitidos por esta matriz pueden variar de acuerdo al pro-

²En el modelo RGB, la intensidad de color es un número entero entre 0 y 255.

³En su obra “La geometría fractal de la naturaleza” [33], Benoît Mandelbrot denomina a esta “curva fractal autocuadrática”, como conjunto o mapa μ .

toloco empleado, pudiendo ser a valores enteros entre 0 y 255, o reales en un intervalo determinado como $[0, 1]$, incluso dependiendo de la información que contiene la imagen; pueden emplearse intervalos mayores, así como varias “capas” de información para lo cual se emplean arreglos matriciales en 3 dimensiones.

De forma general una *imagen digital* es una discretización de un campo escalar $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ al que denominamos *imagen*. Para el dominio Ω consideremos un mallado dado por $x_1 < x_2 < \dots < x_m$ y $y_1 < y_2 < \dots < y_n$, con lo cual para el subdominio discreto

$$\{(x_i, y_j) \in \mathbb{R}^2 \mid 1 \leq i \leq m, 1 \leq j \leq n, i, j, m, n \in \mathbb{N}\} \subset \Omega,$$

la imagen u se puede reinterpretar como la matriz

$$u = (u_{ij})_{m \times n}, \quad \text{donde} \quad u_{ij} = u(x_i, y_j).$$

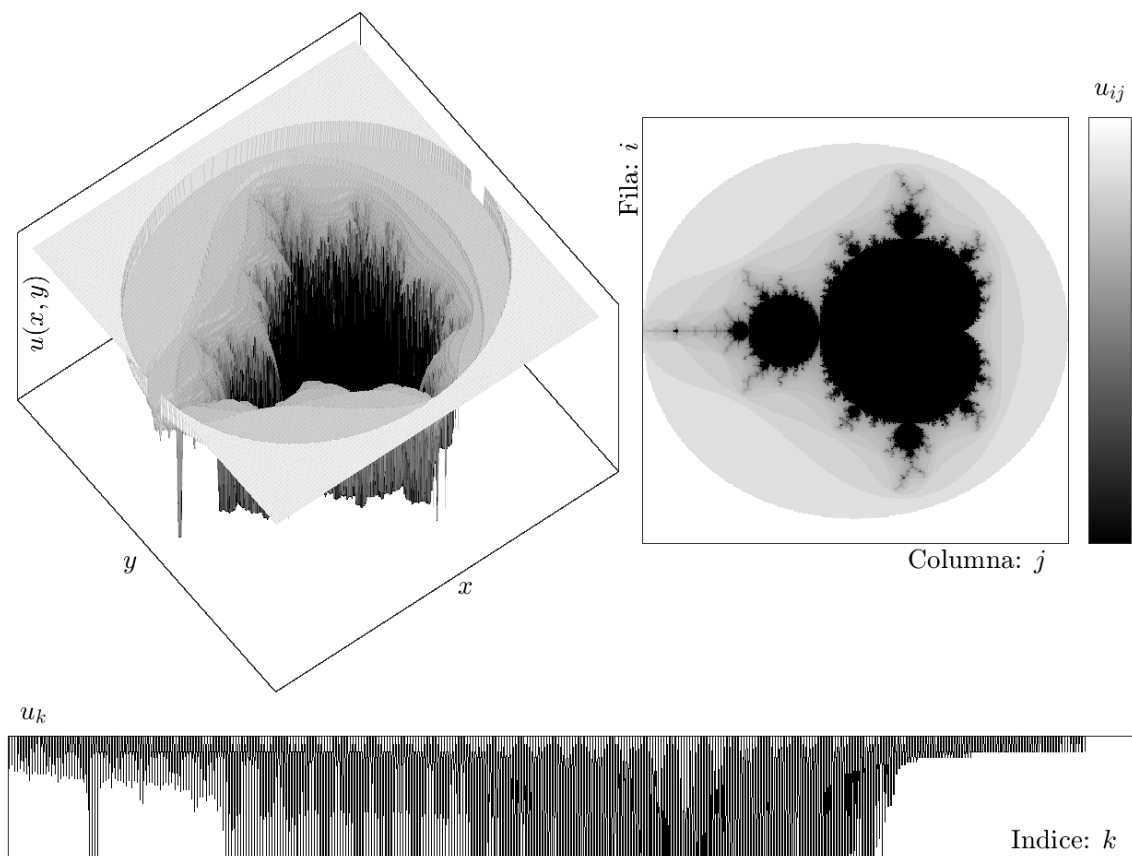


Figura 1.2: Representación gráfica de la interpretación de una imagen digital como: el campo escalar u , la matriz (u_{ij}) y el vector (u_k) .

Por isomorfismo entre el espacio de matrices reales de orden $m \times n$ y el espacio euclídeo \mathbb{R}^{mn} [41], y definiendo $w_k = (x_i, y_j)$, para reescribir el subdominio discreto como

$$\{w_k \in \mathbb{R}^2 \mid 1 \leq k \leq mn, k, m, n \in \mathbb{N}\} \subset \Omega.$$

Tenemos que, otra interpretación equivalente de la imagen u , es como el vector

$$u = (u_1, u_2, \dots, u_{mn}), \quad \text{donde} \quad u_k = u(w_k).$$

La notación de u se emplea indistintamente para imagen, e imagen digital, sea en cualquiera de sus interpretaciones discretizadas equivalentes, tanto como matriz o como vector. En la Figura 1.2 se puede apreciar las representaciones gráficas, para las diferentes interpretaciones matemáticas que damos a una imagen y su versión digital.

Capítulo 2

Problemas inversos lineales

2.1. Medida lineal indirecta

Müller y Siltanen [39] dan un planteamiento significativo a un problema directo práctico que surge en ciencia y tecnología, el de “medida lineal indirecta” dado por

$$m = \mathcal{A}f + \varepsilon \tag{2.1}$$

donde f es una función continua por partes sobre un subconjunto de \mathbb{R}^d , $m \in \mathbb{R}^k$ es un vector que representa las medidas de un dispositivo instrumental, $\mathcal{A}: \mathbb{R}^d \rightarrow \mathbb{R}^k$ es un operador lineal, generalmente una EDP o una ecuación integral, y donde $\varepsilon \in \mathbb{R}^k$ representa el error de medición.

El interés específico sobre este tipo de problemas, es primordial, cuando se busca determinar f en un ámbito físico en el que no se lo puede medir directamente, pero que está asociado con alguna otra cantidad física medible (con “resolución finita” [16]), a la que hemos representado por m . Todo esto siempre y cuando la interacción entre las dos cantidades esté modelada por un operador lineal; \mathcal{A} .

Definición 7 (Par problema lineal directo–inverso). Sean X, Y dos espacios de Hilbert, $\mathcal{A}: X \rightarrow Y$ un operador lineal, al problema de encontrar $y = \mathcal{A}x$ dado x , lo denominamos *problema lineal directo*. Y al problema de encontrar x dados $y = \mathcal{A}x + \varepsilon$ y $\delta > 0$ tal que $\|\varepsilon\| \leq \delta$, lo denominamos *problema lineal inverso*.

El operador lineal \mathcal{A} al ser función, define un grafo, por lo que el problema lineal directo es un problema directo, y puesto que para $\delta \rightarrow 0$, el problema lineal inverso es un problema inverso, entonces el grafo que contiene al grafo inverso \mathcal{A}^{-1} , queda determinado por la condición $\|\varepsilon\| \leq \delta$.

2.2. Problemas mal puestos

La concepción determinista de la física del siglo XIX, limitó la comprensión de los fenómenos físicos a aquellos problemas que permitieran una determinación *única, completa y estable*. En términos matemáticos, dado el planteamiento diferencial (*e.d.* una ecuación en derivadas parciales) asociado a un problema, esta concepción implica la búsqueda de una solución que satisfaga unicidad, existencia y continuidad respecto de los datos (en este caso, condiciones iniciales o de frontera) tal como lo propuso Jacques Hadamard¹. Es así que fueron durante mucho tiempo considerados como problemas legítimos de la física, únicamente aquellos que cumplieran estas características, a los que se denomina *problemas bien puesto*, hasta que la utilidad de los problemas inversos mal puestos fue descubierta ya adentrado el siglo XX [7][16][39][40].

En matemática, la *existencia* de una solución una y otra vez se fuerza al ampliar el espacio de soluciones. Creamos nuevas estructuras como la extensión de cuerpos en álgebra, o los espacios de Sobolev en Análisis. Para la dificultad de obtener más de una *solución única*, especificamos características, como agregar propiedades adicional tales como las condiciones iniciales o de frontera para el caso de una ecuación diferencial o más simples como el signo de la solución, o el de incorporar restricciones como en polítopos, en general se hace el proceso contrario que el de ampliar el espacio de soluciones, es decir, se restringe el dominio. Mientras Kirsch [24] (cuya definición de problema mal puesto, reproducimos) nos comenta que: “El requisito de *estabilidad* es el más importante. Si un problema carece de la propiedad de estabilidad, entonces su solución es prácticamente imposible de calcular porque cualquier medición o cálcu-

¹Jacques Hadamard (1865-1963) fue un matemático francés conocido por su trabajo en teoría de números, análisis complejo y EDP. Demostró el teorema de la distribución asintótica de los números primos y estableció la noción de problema bien puesto.

lo numérico está contaminado por errores inevitables: ¡así los datos de un problema siempre están perturbados por ruido! Si la solución de un problema no depende continuamente de los datos, entonces en general la solución calculada no tiene nada que ver con la solución verdadera.”

Definición 8 (Problema Bien Puesto). Sean X y Y espacios de Hilbert, y $\mathcal{A} : X \rightarrow Y$ un operador lineal. La ecuación $\mathcal{A}x = y$ se llama *problema bien puesto* si se cumple lo siguiente:

1. *Existencia*: Para cada $y \in Y$, existe un $x \in X$ tal que $\mathcal{A}x = y$.
2. *Unicidad*: Para cada $y \in Y$, existe un único $x \in X$ tal que $\mathcal{A}x = y$.
3. *Estabilidad*: La solución x depende continuamente de y ; es decir, para cada sucesión (x_n) en X tal que $\mathcal{A}x_n \rightarrow \mathcal{A}x$ cuando $n \rightarrow \infty$, implica que $x_n \rightarrow x$ para $n \rightarrow \infty$.

Las ecuaciones para las cuales (al menos) una de estas propiedades no se cumple se denominan *problemas mal puestos*.

2.3. Par Convolución–Deconvolución

Dos fenómenos bien estudiados en la teoría y aplicación de problemas inversos son el de imágenes “desenfocadas” (“borrosas” o blurring, en inglés) y ruidosas (noising), que de paso son un ejemplo clásico para retratar a los problemas mal puestos y que también constituye un problema de medida lineal indirecta [7][16][39].

En el ámbito de procesamiento de imágenes, se denomina *blurring* o *desenfoque* a la degradación producida en el proceso de formación óptica de una imagen y se estudia como un problema determinista. Durante el proceso de formación de imágenes, ciertas propiedades espaciales del objeto pueden perderse o atenuarse, lo que resulta en una imagen que parece borrosa o desenfocada en comparación con el objeto original. En cambio, llamamos *ruido* a la degradación generada por el proceso de registro, grabación o captación de una imagen al que se plantea como un problema estadístico. Durante el proceso de grabación de una imagen,

pueden ocurrir errores en la medición de la luz, entonces el ruido puede manifestarse como variaciones aleatorias en el brillo o el color de los píxeles de la imagen para el caso digital [7].

Matemáticamente el modelo continuo para el *desenfoque* de imágenes aprovecha el efecto de suavizado que produce la convolución sobre funciones no diferenciables en un número finito de puntos.

Definición 9 (Desenfoque). Sean $f \in L^2(\mathbb{R})$ una función periódica y $\psi : \mathbb{R} \rightarrow \mathbb{R}$ una función también periódica, además, par, no negativa y normalizada con soporte centrado en el intervalo $[-a, a]$ a la que denominamos *función de dispersión de punto* (PSF)², es decir, tal que

$$\psi(x) \geq 0, \quad \psi(x) = \psi(-x), \quad \text{para } x \in \mathbb{R} \quad \text{y} \quad \int_{-a}^a \psi(x) dx = 1, \quad (2.2)$$

entonces el *modelo de convolución continua, desenfoque o blurring*, está dado por la siguiente integral definida para todo $t \in \mathbb{R}$ como:

$$(\psi * f)(t) = \int_{-a}^a \psi(s) f(t - s) ds. \quad (2.3)$$

Dada una función de dispersión de punto ψ , definimos el operador $\mathcal{A}_\psi : L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})$, como $\mathcal{A}_\psi f(t) = (\psi * f)(t)$ para todo $t \in \mathbb{R}$. Notemos además por m , una evaluación discretizada de la convolución en los puntos t_1, \dots, t_k del dominio de f , es decir,

$$m = [(\psi * f)(t_1), (\psi * f)(t_2), \dots, (\psi * f)(t_k)]^T \in \mathbb{R}^k.$$

En la Figura 2.1, se representa el problema directo del desenfoque, apli-

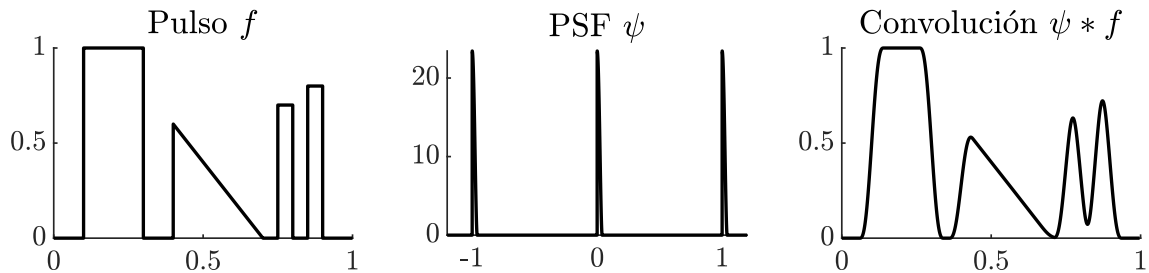


Figura 2.1: Efecto de suavizado $m = \mathcal{A}_\psi f = \psi * f$ (imagen a la derecha) producido por el núcleo ψ (imagen central) sobre el pulso f definido en la imagen a la izquierda.

²En inglés, *point spread function*.

cado sobre el pulso f y dado por el efecto de suavizado que produce la convolución en una dimensión con PSF o núcleo ψ ; definido periódicamente en $[n - a, n + a]$, $n \in \mathbb{Z}$ a partir de $\psi_0(x) = C_a(x + a)^2(x - a)^2$, con $a = 0,04$, $C_a = \int_{-a}^a (x + a)^2(x - a)^2 dx$ y tal que $\int_{-a}^a \psi_0(x) dx = 1$.

2.3.1. Formulación lineal inversa

El problema de recuperar la “imagen” borrosa f , se puede expresar para un “desenfoque” específico ψ , y las medidas m , como el problema de *encontrar f dado*

$$m = \mathcal{A}_\psi f. \quad (2.4)$$

El operador \mathcal{A}_ψ es lineal [7], lo cual se sigue de la linealidad, de la convolución y a su vez, de la, de la integral. Aún cuando $\mathcal{A}_\psi f \in L^2(\mathbb{R})$ [8][24][29], y $m \in \mathbb{R}^k$, la notación $m = \mathcal{A}_\psi f$ se justifica, puesto representa que $(f, m) \in \mathcal{A}_\psi \subseteq L^2(\mathbb{R}) \times \mathbb{R}^k$ donde \mathcal{A}_ψ representa tanto el grafo de la discretización, como el operador integral. Si además consideramos el efecto de ruido al captar una imagen desenfocada, y modelamos ese ruido como una variable aleatoria $\varepsilon \in \mathbb{R}^k$, entonces el modelo en (2.4) se expresa como

$$m = \mathcal{A}_\psi f + \varepsilon, \quad (2.5)$$

donde buscamos determinar f ; esto define un problema inverso de medida lineal indirecta que retomaremos más adelante en el Capítulo 3; para introducir los conceptos de regularización e inversión regularizada.

Una presentación más concreta del *modelo matemático para el desenfoque* [16], se formula para un par de imágenes integrables $u, v : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$; donde u representa la imagen original, v la borrosa, y siendo $k : \mathbb{R}^2 \rightarrow \mathbb{R}$ la función de desenfoque (blurring, en inglés). La *inversión del desenfoque* (deblurring) equivale a resolver la siguiente ecuación integral

$$\mathcal{A}_k u(x) = \int_{\Omega} k(y)u(y) dy = v(x), \quad \forall x \in \Omega \subset \mathbb{R}^2. \quad (2.6)$$

2.3.2. Modelo discreto

Por la naturaleza analítica de las funciones de dispersión empleadas en (2.3), una solución de (2.4) o (2.6) solo puede ser aproximada numéricamente; por lo que el modelo debe ser discretizado. Entonces es de nuestro interés determinar una discretización para la transformación integral \mathcal{A}_ψ . Por el álgebra lineal, toda aplicación lineal definida sobre espacios vectoriales de dimensión finita, tiene asociada o se puede representar con una matriz [11]. Este no es nuestro caso, puesto los espacios sobre los que opera \mathcal{A}_ψ son de dimensión infinita, en cuya situación podemos extender la idea de la representación matricial, a la de una representación por “sucesión” de matrices de orden cuadrado y creciente $(A_{n \times n})_{n \in \mathbb{N}}$, es decir

$$A_{n \times n} [f(x_1) \cdots f(x_n)]^T \longrightarrow [\mathcal{A}_\psi f(x_1) \cdots \mathcal{A}_\psi f(x_n)]^T,$$

cuando $n \longrightarrow +\infty$. Nuestras aproximaciones son mucho menos ambiciosas y se limitan a un muestreo sobre la variable espacial reducida a un número finito de elementos, es decir, buscamos una matriz $A_{n \times n} \approx \mathcal{A}_\psi$.

En concreto, supongamos que la función f está definida en el intervalo $[b, b + 1]$, y considerando la discretización de la variable espacial dada por los n puntos $x_j = b + \frac{j-1}{n}$, con $j = 1, 2, \dots, n \in \mathbb{N}$ y $\Delta x = 1/n$. Luego, f queda representada por el vector $\mathbf{f} = [f(x_1) f(x_2) \cdots f(x_n)]^T \in \mathbb{R}^n$ y diremos que \mathbf{f} aproxima a f , o en notación $\mathbf{f} \approx f$. En virtud de las expresiones en (2.2), se define $\tilde{\mathbf{p}}_j = \psi_0(j\Delta x)$ para $j = -v, \dots, v$, con $v > 0$ el entero más pequeño que satisface la desigualdad $(v + 1)\Delta x > a$ y que luego normalizamos para obtener la versión discreta de PSF, notada $\mathbf{p} = [\mathbf{p}_{-v} \mathbf{p}_{-v+1} \cdots \mathbf{p}_{-1} \mathbf{p}_0 \mathbf{p}_1 \cdots \mathbf{p}_{v-1} \mathbf{p}_v]^T$, donde $\mathbf{p}_j = \left(\Delta x \sum_{j=-v}^v \tilde{\mathbf{p}}_j \right)^{-1} \tilde{\mathbf{p}}_j$ de tal forma que

$$\Delta x \sum_{j=-v}^v \mathbf{p}_j = 1.$$

Con esto, la aproximación para (2.3) viene dada por la siguiente cuadratura para la integral dada para los punto x_j , donde para cada $j = 1, \dots, n$,

$$\mathcal{A}_\psi f(x_j) = \int_{-a}^a \psi(y) f(x_j - y) dy \approx \Delta x \sum_{\ell=-v}^v \psi(x_\ell) f(x_j - x_\ell) \approx \Delta x \sum_{\ell=-v}^v \mathbf{p}_\ell \mathbf{f}_{j-\ell}.$$

Notemos que $f(x_j - x_\ell)$ a sido aproximado por $\mathbf{f}_{j-\ell} = f(x_{j-\ell})$, para lo cual

se debe considerar las condiciones de frontera para el periodo de f en los casos $j - \ell < 1$ y $j - \ell > n$.

Por ejemplo, si $\mathbf{p} = [\mathbf{p}_{-1} \ \mathbf{p}_0 \ \dots \ \mathbf{p}_1]^T$, la expresión anterior produce las n ecuaciones siguientes

$$\begin{aligned}\mathcal{A}_\psi f(x_1) &= \Delta x (\mathbf{p}_0 \mathbf{f}_1 + \mathbf{p}_{-1} \mathbf{f}_2 + \mathbf{p}_1 \mathbf{f}_n), \\ \mathcal{A}_\psi f(x_2) &= \Delta x (\mathbf{p}_1 \mathbf{f}_1 + \mathbf{p}_0 \mathbf{f}_2 + \mathbf{p}_{-1} \mathbf{f}_3), \\ \mathcal{A}_\psi f(x_3) &= \Delta x (\mathbf{p}_1 \mathbf{f}_2 + \mathbf{p}_0 \mathbf{f}_3 + \mathbf{p}_{-1} \mathbf{f}_4), \\ &\vdots \\ \mathcal{A}_\psi f(x_{n-1}) &= \Delta x (\mathbf{p}_1 \mathbf{f}_{n-2} + \mathbf{p}_0 \mathbf{f}_{n-1} + \mathbf{p}_{-1} \mathbf{f}_n), \\ \mathcal{A}_\psi f(x_n) &= \Delta x (\mathbf{p}_{-1} \mathbf{f}_1 + \mathbf{p}_1 \mathbf{f}_{n-1} + \mathbf{p}_0 \mathbf{f}_n),\end{aligned}$$

de donde se colige que la matriz A , tal que $\mathcal{A}_\psi f \approx Af$ tiene la forma

$$A = \Delta x \begin{pmatrix} \mathbf{p}_0 & \mathbf{p}_{-1} & 0 & 0 & 0 & \cdots & 0 & \mathbf{p}_1 \\ \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_{-1} \\ \mathbf{p}_{-1} & 0 & \cdots & 0 & 0 & 0 & \mathbf{p}_1 & \mathbf{p}_0 \end{pmatrix}.$$

Se ha implementado computacionalmente en MATLAB, esta discretización a través de los métodos: `entrada`, `psf_discreta`, `matriz_convolucion`, `calculo_eta` y `convolucion_discreta`. Lo que ha permitido realizar el gráfico en la Figura 2.1 y varias de las figuras en el Capítulo 3. Específicamente en la Figura 3.1 se representan cuatro diferentes discretizaciones para 32, 64, 128 y 256 elementos.

2.4. Tomografía Computarizada de Rayos X (CT)

La radiología nació el 8 de noviembre de 1895 con el descubrimiento de los Rayos X por Wilhelm Conrad Röntgen³ mientras estudiaba el poder de penetración de los rayos catódicos y observó que una placa de cartón

³Ingeniero mecánico y físico alemán, galardonado con el premio Premio Nobel de Física en 1901.

cubierta de cristales de platino-cianuro de bario emitía una fluorescencia. Después de semanas de investigación sobre su hallazgo, Röntgen se decide a realizar su primera prueba en humanos y descubrió que podía ver los huesos de la mano, imprimió la imagen en una placa fotográfica y así nació la primera radiografía. Algunos años después, en 1917, el matemático austriaco Johann Radon estableció los fundamentos matemáticos de la tomografía computarizada al demostrar que es posible reconstruir un objeto bidimensional a tridimensional a partir de un conjunto infinito de proyecciones, para lo cual estableció la transformada de Radon bidimensional, que en concreto se describe como una transformación integral que consiste valga la redundancia, en la integral de una función sobre un conjunto de rectas. Este avance teórico fue aplicado por Allan McLeod Cormack⁴, quien propuso el uso de los resultados de Radon para fines médicos. En 1963 Cormack mostró que se podían calcular los coeficientes de absorción de una estructura plana y medir desde un cierto número de direcciones las variaciones de intensidad de los rayos transmitidos. Así y gracias al desarrollo de la computación durante la segunda mitad del siglo XX, surgió la técnica conocida como tomografía computarizada [42][20][9][19].

2.4.1. Modelo determinista

En la Tomografía⁵ Computarizada de Rayos X (CT) un haz fino de Rayos X atraviesa una sección del cuerpo humano y su atenuación de intensidad es captada por un detector y procesada por una computadora para generar una imagen bidimensional que se muestra en una pantalla, así surge el problema que describiremos a continuación [40].

Con el fin de mantener la sencillez, se plantea una versión simplificada del modelo físico de la CT a partir de las propuestas de Natterer [40] y Engl et al. [16]. Para ello, consideremos un tejido como una estructura plana e interpretada como un dominio compacto Ω en \mathbb{R}^2 , con una “densidad espacial” variable $f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$. En el modelo, cada

⁴Físico y cristalógrafo sudafricano-estadounidense quien junto al ingeniero eléctrico británico Godfrey Hounsfield recibieron el premio Nobel de Fisiología o Medicina en 1979 [45].

⁵‘Tomografía’ viene del griego $\tau\acute{o}\mu\omicron\varsigma$ que significa rebanada, corte, o sección [40].

$f(x)$ constituye el coeficiente de atenuación de Rayos X o *densidad* del tejido en cada punto $x \in \Omega$. Considerando además, que la interacción física entre la radiación y la materia solo puede producir disminución en la variable I para la *intensidad* de los Rayos X [39] medida en valores positivos, implica que $I \in \mathbb{R}^+$. El *decaimiento* de la intensidad relativa $\Delta I/I : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}$ que sufren los Rayos X, es directamente proporcional a la distancia recorrida $\Delta x : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow [0, +\infty)$. Matemáticamente, esto se expresa en la siguiente ecuación

$$\frac{\Delta I}{I} = f(x)\Delta x. \quad (2.7)$$

Aplicando diferenciación, y para ello tomando los límites $\Delta I \rightarrow 0$ y $\Delta x \rightarrow 0$ obtenemos una ecuación diferencial. Luego, integrando en la trayectoria de la línea recta L , que sigue un haz de Rayos X, y considerando que existe una pérdida de intensidad producida en el trayecto de atravesar un tejido, se tiene que $I_0 > I_1$ y por tanto para cada $x \in \Omega$,

$$\begin{aligned} \frac{dI}{I} &= f(x) dx \\ \int_{I_0}^{I_1} \frac{dI}{I} &= \int_L f(x) dx \\ \ln(I) \Big|_{I_0}^{I_1} &= \int_L f(x) dx \\ \ln(I_1) - \ln(I_0) &= \int_L f(x) dx \\ \ln\left(\frac{I_1}{I_0}\right) &= \int_L f(x) dx, \end{aligned} \quad (2.8)$$

donde I_0 es la medida de intensidad inicial, e I_1 su medida después de haber atravesado el cuerpo.

Respecto de la aplicación en medicina, Ω simboliza una sección transversal del cuerpo humano y la ecuación (2.8) implica que el proceso de escaneo nos proporciona la integral de línea de la función f a lo largo de cada una de las líneas L recorridas. Por lo tanto, el objetivo será recuperar la densidad f de las mediciones de Rayos X en el plano que contiene a Ω y que interpretamos como una “imagen” CT. Matemáticamente, esto implica encontrar la f conociendo las integrales de un número finito de líneas. Evidentemente, estamos ante un problema inverso.

2.4.2. Formulaci3n lineal inversa

Si representamos por $R : L^2(\mathbb{R}^2) \longrightarrow L^2(\mathbb{R}^2)$ al operador integral (en consecuencia, lineal) definido por $Rf = \int_L f(x) dx$ y dado por el miembro a la derecha en la ecuaci3n (2.8), entonces podemos plantear el *problema directo* de la tomograf3a de Rayos X, como *determinar el decaimiento de la intensidad* $m = \ln(I_1/I_0)$, tal que

$$m = Rf, \tag{2.9}$$

donde f es la densidad conocida de un objeto a lo largo de la linea que sigue un haz de Rayos X en el plano de un corte. En la Figura 2.2, esbozamos a la izquierda, el proceso de medici3n del decaimiento para una imagen de prueba formada por dos cuadrados y simulada en MATLAB con densidad conocida $f(x) \in \{0, 1\}$ (colores negro y blanco respectivamente), que representa un ‘‘corte’’ de tomograf3a. A la derecha representamos gr3ficamente el efecto del operador R a lo largo de cuatro direcciones diferentes; dadas por el 3ngulo de incidencia especificado sobre cada recuadro, emple3ndose la funci3n `radon` disponible en MATLAB.

Tiene mayor relevancia el problema inverso, es decir, la tomograf3a computarizada, y al igual que en deblurring, se considera un ruido ε sobre las mediciones $m \in \mathbb{R}^k$ de la intensidad de los Rayos X, lo que

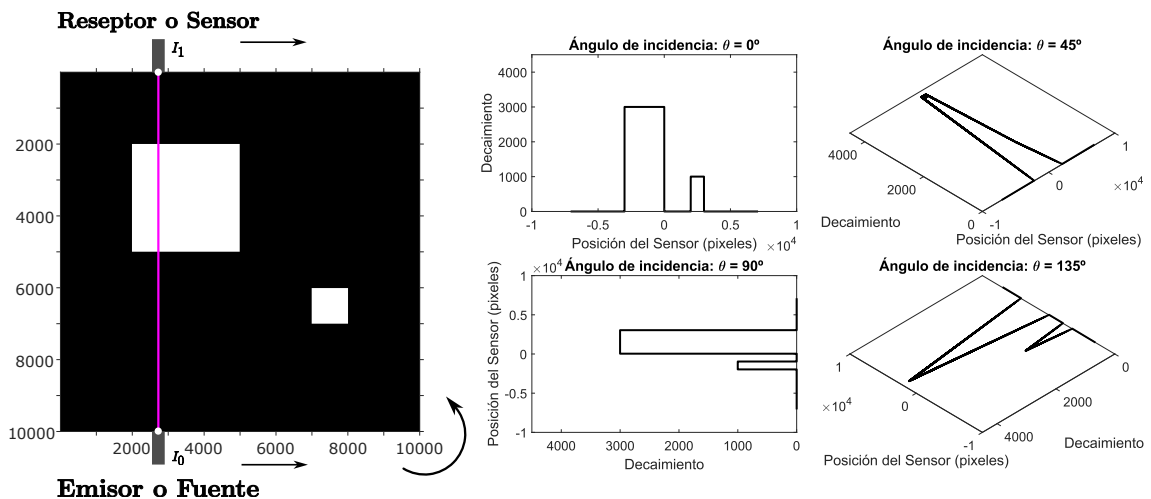


Figura 2.2: Medici3n de un haz de Rayos X y transformaci3n integral de la densidad de una imagen simulada, para 4 3ngulos de incidencia.

configura el esquema del siguiente problema de medida lineal indirecta:

$$\begin{aligned} \text{Dado } m &= \mathbf{R}f + \varepsilon, \\ \text{encontrar la densidad } f &\text{ para un dominio compacto } \Omega \subset \mathbb{R}^2. \end{aligned} \tag{2.10}$$

2.4.3. Modelo discreto

Para modelar los datos prácticos de tomografía de Rayos X, consideremos el conjunto acotado $\Omega \subset \mathbb{R}^2$ que representa el dominio de nuestro objeto de interés. Consideremos además un coeficiente de atenuación no negativo f , soportado en $\bar{\Omega}$ que representa la capacidad del material para absorber los Rayos X. Este coeficiente de atenuación definido en cada punto de $\bar{\Omega}$ se interpreta como la imagen que se busca “reconstruir”. Además, es indispensable definir una colección finita $\{L_j\}_{j=1}^k$ de líneas $L_j \subset \mathbb{R}^2$ que intersecan Ω [15, Sec. 2.3.4]. Estas líneas representan los caminos que siguen los Rayos X a través del material y modelan tanto las posibilidades concretas de realizar las mediciones como la disposición de los emisores y sensores con este fin. Al medir la atenuación de los Rayos X a lo largo de estas líneas, podemos obtener información sobre el coeficiente de atenuación f y por lo tanto, sobre la estructura interna del objeto “escaneado”. Justamente a este proceso de medición por barrido, empleando líneas rectas, es al que denominamos *escaneo*. La discretización nos permitirá representar y resolver computacionalmente el problema (2.10).

Geometría de haces paralelos

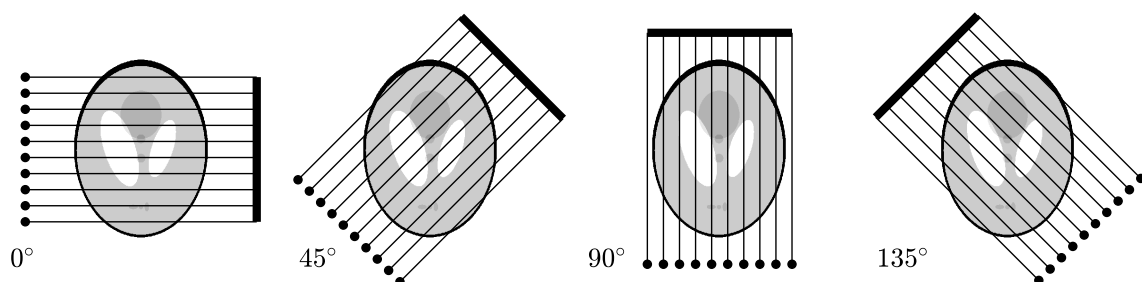


Figura 2.3: Escaneo empleando 10 haces paralelos (líneas negras delgadas), para 4 ángulos diferentes. Los puntos negros representan la posición de los sensores que miden el decaimiento de la intensidad de los Rayos X emitidos desde la fuente (barra negra de mayor grosor).

Hay diferentes disposiciones para las rectas $\{L(\theta_j, s_\nu)\}_{j=1, \dots, J}^{\nu=1, \dots, N}$, entre ellas las de abanico y las de cono [40], en este trabajo se aborda únicamente la de haces paralelos, representada en la Figura 2.3. Para lo cual el parámetro angular θ que determina la dirección de las rectas paralelas se muestrea de forma equidistante en el semicírculo superior, por el conjunto finito $\{\theta_j\}_{j=1}^J$, es decir, que a partir de un ángulo de referencia $\theta_1 \in \mathbb{R}$ y para una cantidad fija $J \in \mathbb{N}$ convenientes, definimos

$$\theta_j = \theta_1 + \left(\frac{j-1}{J}\right) \pi, \quad 1 \leq j \leq J.$$

La geometría de una de estas líneas paralelas, representadas en la recta $L(\theta, s)$ de la Figura 2.4, se muestra para el ángulo θ y la distancia s ambos fijos y arbitrarios. Por la Geometría Euclidiana: del triángulo rectángulo Ox_1C se sigue que, la hipotenusa OC tiene una longitud $s \sec(\theta)$. Luego por la ecuación explícita de la recta, y dado que $L(\theta, s)$ es perpendicular al segmento Ox_1 con pendiente $\tan(\theta)$ y punto de corte en C , se sigue que,

$$x_2 = -\frac{1}{\tan(\theta)}x_1 + s \sec(\theta) \quad \text{o equivalentemente} \quad (x_1, x_2) \cdot (\sin(\theta), \cos(\theta)) = s.$$

Luego podemos definir cada recta de la discretización como

$$L(\theta_j, s_\nu) = \{x \in \mathbb{R}^2 \mid \alpha(\theta_j) \cdot x = s_\nu\} \cap \Omega, \quad (2.11)$$

con la función $\alpha : \mathbb{R} \rightarrow \mathbb{R}^2$, tal que $\alpha(\theta) = (\sin(\theta), \cos(\theta))$, y el parámetro

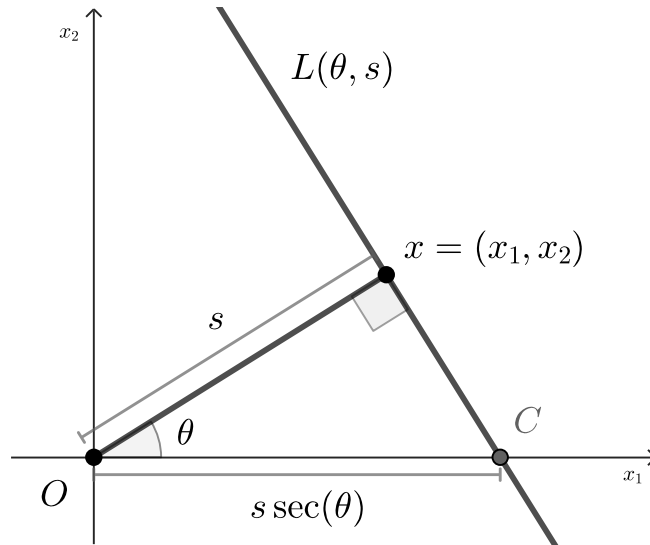


Figura 2.4: Geometría de la trayectoria recta de un haz de Rayos X.

radial s muestreado uniformemente con el conjunto finito $\{s_\nu\}_{\nu=1}^N$ donde

$$s_\nu = -S + 2 \left(\frac{\nu - 1}{n} \right) S, \quad 1 \leq \nu \leq N,$$

con $S > 0$ y $N \in \mathbb{N}$.

El dominio Ω para la tomografía debe ser discretizado, dividiéndolo en n píxeles, es decir, n rectángulos congruentes con intersecciones vacías dos a dos que recubren, es decir teselan a Ω . Además, asumimos que el valor de la atenuación es homogéneo en cada pixel, es decir, constante al interior de cada rectángulo.

Enumeraremos cada píxel de 1 a n , y notaremos por $f_l > 0$ a su correspondiente valor para el coeficiente de atenuación con $l = 1, \dots, n$. Recurriendo a una enumeración similar, las rectas $L(\theta_j, s_\nu)$ se notan simplemente como L_i , con $i = 1, \dots, k$, y $k = J \cdot N$. Luego, la versión vectorial aproximada de (2.9), se escribe como

$$m = \mathbf{R}f \approx \left(\int_{L_1} f(x) dx, \dots, \int_{L_k} f(x) dx \right). \quad (2.12)$$

Si notamos $m = (\mathbf{m}_1, \dots, \mathbf{m}_k) \in \mathbb{R}^k$ al vector de mediciones, \mathbf{m}_i representa la integral en cada línea L_i ; la cual se aproxima por cuadratura, es decir

$$\mathbf{m}_i = \int_{L_i} f(x) dx \approx \sum_{l=1}^n a_{il} f_l, \quad (2.13)$$

donde a_{il} es la porción o segmento de L_i que corta (interseca) al píxel l . Notemos que solo los píxeles que intersecan el haz L_i son incluidos en esta suma. Así (2.12) y (2.13) producen la ecuación matricial $m = Af$, donde la matriz de orden $k \times n$, queda definida por $A = (a_{il})$.

El proceso se ilustra en la Figura 2.5, generada a partir del *fantoma* de Shepp–Logan (*phantom* en inglés), una imagen en escala de grises para representar un corte craneal, que consiste de una gran elipse (craneo y cerebro) que contiene varias elipses más pequeñas (diferentes tejidos en el cerebro). Esta imagen de prueba estándar fue presentada por Larry Shepp y Benjamin Logan para su artículo de 1974 “The Fourier Reconstruction of a Head Section” y se emplea como modelo en el desarrollo y prueba de algoritmos de reconstrucción de imágenes [52] [35].

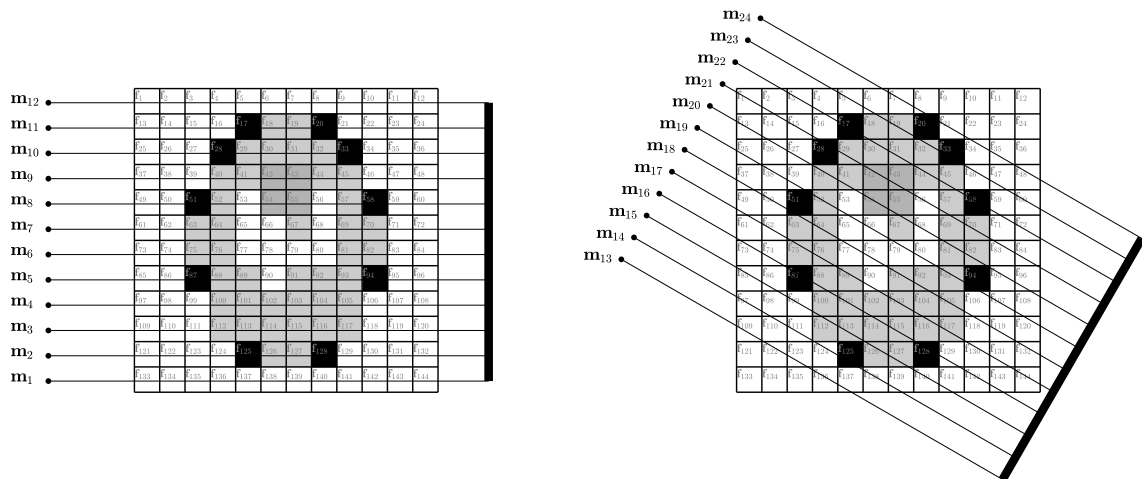


Figura 2.5: Se muestra una subdivisión en 144 píxeles del dominio de la imagen coloreada en escala de grises, que representa al coeficiente de atenuación $\{f_l\}_{l=1}^{144}$ constante en cada uno de los píxeles. A la izquierda se muestran 12 líneas paralelas con un ángulo $\theta_1 = 0^\circ$ y a la derecha otras 12 paralelas, con $\theta_2 = 30^\circ$. Las mediciones de atenuación son el conjunto $\{m_k\}_{k=1}^{24}$ resultantes de cada una de las líneas muestreadas.

2.5. Transformadas de Fourier y de Radon

La transformada que asigna una función en \mathbb{R}^2 , al conjunto de sus integrales de línea, se denomina *transformada de Radon Bidimensional*. Por la Teoría de Operadores Lineales y de la ecuación (2.9), se sigue que, el problema de “reconstrucción” de una imagen en CT involucra la inversión de la transformada de Radon. La forma de conseguirlo es a través de la transformada de Fourier, y bastará estudiarla en los espacios de Schartz. Con este propósito, a lo largo de esta sección, se desarrolla la teoría suficiente y generalizada a \mathbb{R}^n [17][40][24][30][55]. Para representar el vector director $\alpha(\theta)$ de los hiperplanos $x \cdot \alpha(\theta) = s$ que generalizan a las rectas en la expresión (2.11), utilizaremos la notación θ en lugar de $\alpha(\theta)$, y además notaremos θ^\perp para representar tanto al vector unitario perpendicular, como al espacio ortogonal $\{x \in \mathbb{R}^n \mid x \cdot \theta = 0\}$, a el vector θ según corresponda.

Definición 10 (Espacio de Schwartz). Se denomina *espacio de Schwartz*, al espacio vectorial $\mathcal{S}(\mathbb{R}^n)$, cuyos elementos son las funciones $f \in \mathcal{C}^\infty(\mathbb{R}^n)$ tales que

$$\|f\|_{\alpha,\beta} = \|x^\alpha D^\beta f\|_\infty = \sup_{x \in \mathbb{R}^n} \left| x_{i_1}^{\alpha_1} \cdots x_{i_k}^{\alpha_k} \frac{\partial^{|\beta|}}{\partial_{j_1}^{\beta_1} \cdots \partial_{j_l}^{\beta_l}} \right|$$

es finito para todo *multi-índice* $\alpha, \beta \in \mathbb{N}^n$, con $|\beta| = \sum_{j=1}^l \beta_j$, y $k, l \leq n$.

Definición 11 (Transformada de Fourier). Para $f \in L^1(\mathbb{R}^n)$, la *transformada de Fourier* \hat{f} y la *transformada inversa de Fourier* \check{f} se definen como

$$\hat{f}(\xi) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} f(x) e^{-ix \cdot \xi} dx \quad \text{y} \quad \check{f}(\xi) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} f(x) e^{ix \cdot \xi} dx.$$

Este par de transformadas son lineales e inversas una de la otra, lo que produce la siguiente *formula de inversión de Fourier*

$$\check{\hat{f}} = \hat{\check{f}} = f, \quad \forall f \in \mathcal{S}(\mathbb{R}^n), \quad (2.14)$$

de donde se sigue que, la transformada de Fourier es un isomorfismo de $\mathcal{S}(\mathbb{R}^n)$ con sí mismo. Además, si f tiene soporte compacto, entonces \hat{f} extiende a una función analítica en \mathbb{C}^n .

Siempre que se requiera tomar la transformada de Fourier sobre funciones de (θ, s) , ésta se aplicará a la segunda variable. Por ejemplo,

$$\hat{h}(\theta, \sigma) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} h(\theta, s) e^{-is\sigma} ds.$$

Sea S^{n-1} , la $(n-1)$ -esfera que pertenece a \mathbb{R}^n , el *fibrado tangente* está dado por

$$T = \{(\theta, x) \mid \theta \in S^{n-1}, x \in \theta^\perp\},$$

en general se tiene que para $h \in \mathcal{S}(T)$,

$$\hat{h}(\theta, \xi) = (2\pi)^{\frac{1-n}{2}} \int_{\theta^\perp} h(\theta, x) e^{-ix \cdot \xi} dx, \quad \xi \in \theta^\perp.$$

Definición 12 (Transformada de Radon). La *transformada de Radon n -dimensional*, notada \mathbf{R} , asigna una función en \mathbb{R}^n al conjunto de sus integrales sobre los hiperplanos de \mathbb{R}^n , es decir, sobre subespacios afines de dimensión $n-1$. Más específicamente, si $\theta \in S^{n-1}$, y $s \in \mathbb{R}$, entonces

$$\mathbf{R}f(\theta, s) = \int_{x \cdot \theta = s} f(x) dx = \int_{\theta^\perp} f(s\theta + y) dy$$

es la integral de $f \in \mathcal{S}(\mathbb{R}^n)$ sobre el hiperplano perpendicular a θ con una “distancia con signo” s medida desde el origen. $\mathbf{R}f$ es una función par en el cilindro unitario $Z = S^{n-1} \times \mathbb{R} \subset \mathbb{R}^{n+1}$, es decir, $\mathbf{R}f(-\theta, -s) = \mathbf{R}f(\theta, s)$. Se

suele usar, alternativamente, la notación

$$\mathbf{R}_\theta f(s) = \mathbf{R}f(\theta, s).$$

Definición 13 (Transformada de Rayos X). La *transformación de Rayos X* n -dimensional, notada \mathbf{P} , envía una función de \mathbb{R}^n en el conjunto de sus integrales de línea. Más específicamente, si $\theta \in S^{n-1}$ y $x \in \mathbb{R}^n$, entonces

$$\mathbf{P}f(\theta, x) = \int_{-\infty}^{+\infty} f(x + t\theta) dt$$

es la integral de $f \in \mathcal{S}(\mathbb{R}^n)$ sobre la línea recta que pasa por x con dirección θ . Notemos que, $\mathbf{P}f(\theta, x)$ no cambia si x se mueve en la dirección θ . Por lo tanto, normalmente restringimos x a θ^\perp , lo que hace que $\mathbf{P}f$ sea una función en el fibrado tangente T a S^{n-1} . También notaremos

$$\mathbf{P}_\theta f(x) = \mathbf{P}f(\theta, x).$$

Para $n = 2$, \mathbf{P} y \mathbf{R} coinciden excepto por la notación de los argumentos. Además, si $\omega \in S^{n-1}$, es posible expresar $\mathbf{R}f(\omega, s)$ como una integral sobre $\mathbf{P}f$, para cualquier $\theta \in S^{n-1}$ con $\theta \perp \omega$. En otras palabras

$$\mathbf{R}f(\omega, s) = \int_{x \in \theta^\perp, x \cdot \omega = s} \mathbf{P}f(\theta, x) dx. \quad (2.15)$$

Muchas propiedades importantes de las transformadas integrales presentadas anteriormente se derivan de fórmulas que involucran convoluciones y transformadas de Fourier. A continuación, se presentan algunos resultados que relacionan a la transformada de Radon y la transformada de Fourier. Esto permitirá definir una fórmula de inversión para la transformada de Radon [40] [4].

Teorema 1 (Teorema de corte de Fourier [40]). Para $f \in \mathcal{S}(\mathbb{R}^n)$ tenemos

$$\widehat{\mathbf{R}_\theta f}(\sigma) = (2\pi)^{\frac{n-1}{2}} \widehat{f}(\sigma\theta), \quad \sigma \in \mathbb{R},$$

$$\widehat{\mathbf{P}_\theta f}(\eta) = (2\pi)^{1/2} \widehat{f}(\eta), \quad \eta \in \theta^\perp.$$

Demostración. Tenemos que

$$\widehat{\mathbf{R}_\theta f}(\sigma) = (2\pi)^{-1/2} \int_{\mathbb{R}} \mathbf{R}_\theta f(s) e^{-i\sigma s} ds = (2\pi)^{-1/2} \int_{\mathbb{R}} e^{-i\sigma s} \int_{\theta^\perp} f(s\theta + y) dy ds,$$

consideremos el cambio de variable $x = s\theta + y$ que también se escribe como $s = \theta \cdot x$, y donde $dx = dy ds$. Haciendo los reemplazos correspondientes en la expresión anterior y recordando que $\theta^\perp \subset \mathbb{R}^{n-1}$, tenemos

$$\begin{aligned} \widehat{\mathbf{R}_\theta f}(\sigma) &= (2\pi)^{-1/2} \int_{\mathbb{R}} e^{-i\sigma(\theta \cdot x)} \int_{\theta^\perp} f(x) dx \\ &= (2\pi)^{-1/2} \frac{(2\pi)^{n/2}}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(x) e^{-i\sigma\theta \cdot x} dx \\ &= (2\pi)^{(n-1)/2} \widehat{f}(\sigma\theta). \end{aligned}$$

De forma similar,

$$\begin{aligned} \widehat{\mathbf{P}_\theta f}(\eta) &= (2\pi)^{-(n-1)/2} \int_{\theta^\perp} \mathbf{P}_\theta f(y) e^{-i\eta \cdot y} dy \\ &= (2\pi)^{-(n-1)/2} \int_{\theta^\perp} e^{-i\eta \cdot y} \int_{-\infty}^{\infty} f(y + t\theta) dt dy \\ &= (2\pi)^{-(n-1)/2} \int_{\mathbb{R}^n} e^{-i\eta \cdot x} f(x) dx \\ &= (2\pi)^{1/2} \widehat{f}(\eta). \end{aligned}$$

□

Corolario 1 (Teorema del corte central). *Sea f una función absolutamente integrable definida sobre \mathbb{R} . Para cualquier número real r y vector unitario θ , tenemos la identidad*

$$\widehat{\mathbf{R}_\theta f}(r) = \widehat{f}(r\theta).$$

Demostración. El resultado se sigue del Teorema 1, tomando $n = 1$. □

Empleando alternativamente, la notación $\tilde{h}(s, \theta)$ para la *transformada de Fourier unidimensional* en el parámetro escalar, el teorema del corte central se escribe como

$$\widetilde{\mathbf{R}_\theta f}(\theta, r) = \widehat{f}(r\theta). \quad (2.16)$$

La fórmula de inversión de Radon proporciona una forma de obtener f a partir de su transformada de Radon.

Teorema 2 (Retroproyección filtrada [39]). *Si f es una función absolutamente integrable definida en \mathbb{R} y además \hat{f} es absolutamente integrable, entonces*

$$f(x) = \frac{1}{2\pi} \int_0^\pi \int_{-\infty}^{\infty} \widetilde{\mathbf{R}}f(\theta, s) e^{isx \cdot \theta} |s| ds d\theta. \quad (2.17)$$

Demostración. Se mencionó que la transformada de Radon es par, es decir, satisface $\mathbf{R}f(-\theta, -s) = \mathbf{R}f(\theta, s)$, de donde sigue que

$$\begin{aligned} \widetilde{\mathbf{R}}f(-\theta, -s) &= \int_{-\infty}^{\infty} \mathbf{R}f(-\theta, t) e^{-it(-s)} dt = \int_{-\infty}^{\infty} \mathbf{R}f(-\theta, t) e^{-i(-t)s} dt \\ &= \int_{-\infty}^{\infty} \mathbf{R}f(-\theta, -t) e^{-its} dt = \int_{-\infty}^{\infty} \mathbf{R}f(\theta, t) e^{-its} dt \\ &= \widetilde{\mathbf{R}}f(\theta, s). \end{aligned}$$

Ahora por la fórmula de inversión de Fourier (2.14), para $\xi = (r \cos \theta, r \sin \theta)$ y considerando la integración en coordenadas polares [17, Teorema 2.49],

$$\begin{aligned} f(x) &= \check{f}(x) \\ &= \frac{1}{2\pi} \int_{\mathbb{R}^2} \hat{f}(\xi) e^{ix \cdot \xi} d\xi \\ &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} \hat{f}(r\theta) e^{irx \cdot \theta} r dr d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} \widetilde{\mathbf{R}}f(\theta, r) e^{irx \cdot \theta} r dr d\theta \\ &= \frac{1}{2\pi} \int_0^\pi \int_{-\infty}^{\infty} \widetilde{\mathbf{R}}f(\theta, r) e^{irx \cdot \theta} |r| dr d\theta \end{aligned}$$

la última identidad se obtiene de propiedades del valor absoluto y de la paridad $\widetilde{\mathbf{R}}f(-\theta, -s) = \widetilde{\mathbf{R}}f(\theta, s)$. \square

En la siguiente sección presentaremos una aplicación relacionada a Tomografía Computarizada, específicamente para la Tomografía por Emisión de Positrones (PET). El modelo físico intrínseco está relacionado, también, con la Transformada de Radon. Posteriormente, se introducirá la técnica de Resonancia Magnética (MR) que involucra la transformada de Fourier de la imagen a reconstruir.

2.6. Tomografía por Emisión de Positrones

Existen cuatro fuerzas fundamentales de la naturaleza: la gravitacional, electromagnética, fuerte y débil, que actúan sobre las propiedades de las partículas fundamentales. La bien conocida carga electrostática o simplemente *carga* es una de estas propiedades, otra menos familiar, es el *spin*, el momento angular intrínseco de una partícula. El concepto de spin llevó a Dirac desde el paradigma de simetría a sugerir la existencia del *positrón*, el equivalente de *antimateria* del electrón. El *iso-spin* es otra propiedad mecánica cuántica utilizada para describir la simetría entre diferentes partículas que se comportan casi idénticamente bajo la influencia de la fuerza fuerte. Bajo ciertas condiciones, un fotón de alta energía puede materializarse espontáneamente en un par electrón-positrón, conservando tanto la carga como el iso-spin. Dado que el electrón tiene carga -1 y spin $+\frac{1}{2}$, y el positrón tiene carga $+1$ y spin $-\frac{1}{2}$. El par electrón-positrón tiene un corto *periodo* de tiempo de duración, en el que se combinan o “eliminan” mutuamente, a este proceso se denomina aniquilación, y a la energía liberada en él, *radiación de aniquilación*. La energía de la radiación es equivalente a la masa en reposo del electrón y el positrón. Hay dos fotones producidos por cada desintegración de positrones y aniquilación. Cada fotón tiene una alta energía (511 keV) y se emiten en direcciones opuestas cercanas a 180° para conservar el momento (cercano a cero antes de la aniquilación). Esta propiedad de colinealidad se explota en PET, permitiendo definir la “línea de visión” del evento sin necesidad de colimación⁶. Sin embargo, muchos pares de fotones no se emiten estrictamente a 180° ; debido al momento no nulo cuando el positrón y el electrón se aniquilan. Se ha estimado que esta fracción puede ser tan alta como el 65% en agua. Cuando la radiación de alta energía interactúa con la materia, la energía puede transferirse al material. Un resultado común es la excitación de los átomos en el material absorbente. La producción de pares electrón-positrón ofreció la primera evidencia experimental de la *antimateria* postulada por Dirac, es decir, que para cada partícula en el universo existe una versión *espejo* de ella [4].

⁶La colimación, es un proceso utilizado en imagen médica para limitar la exposición a la radiación y mejorar la calidad de la imagen; empleando dispositivos que restringen el haz de radiación a una región específica del cuerpo del paciente.

La Tomografía por Emisión de Positrones (PET) es una técnica de imagen médica en la que se inyecta en el paciente un “trazador” radiactivo marcado por positrones, es decir, utiliza radioisótopos que se desintegran emitiendo positrones. Estos positrones se aniquilan con los electrones del tejido, generando dos fotones electromagnéticos que se emiten en direcciones opuestas. La detección de estos fotones en coincidencia temporal permite reconstruir la “distribución” del radioisótopo en el cuerpo del paciente. Visto en forma general, esto corresponde a la transformada de Rayos X, que coincide con la transformada de Radon en bidimensional [4][15][31].

Para entender el proceso PET, es importante mencionar brevemente una descripción general para la detección de sus eventos basada en la colimación electrónica, en Bailey [4] se describe este proceso con claridad. Un evento se considera *válido* si:

1. Se detectan dos fotones dentro de una ventana de tiempo electrónico predefinida conocida como la ventana de coincidencia,
2. la línea de respuesta formada entre ellos está dentro de un ángulo de aceptación válido del tomógrafo y,
3. la energía depositada en el cristal (detector) por ambos fotones está dentro de la ventana de energía seleccionada.

Estos eventos coincidentes a menudo se denominan *eventos rápidos* (o “prompts” en inglés). Algunos eventos registrados como válidos son en realidad *eventos no deseados*. Hay varios tipos de eventos en la detección de PET:

1. *Único*: un solo fotón es contado por un detector.
2. *Real (o verdadera)*: evento que deriva de una sola aniquilación positrón-electrón.
3. *Aleatorio (o accidental)*: ocurre cuando dos núcleos se desintegran aproximadamente al mismo tiempo.
4. *Múltiple*: similar al evento aleatorio, excepto que se detectan tres eventos de dos aniquilaciones dentro de la ventana de tiempo de coincidencia.

5. *Disperso*: uno o ambos fotones de una sola aniquilación, detectados dentro de la ventana de tiempo de coincidencia, han sufrido un efecto Compton (Compton scattering) ⁷.

Esto causa inconsistencias en los datos de proyección y conduce a una disminución del contraste y una cuantificación inexacta en la imagen final. Podemos apreciar una representación gráfica de estos eventos en la Figura 2.6, donde se ha empleado una nube de círculos de centro y radio aleatorios inspirado en el fantoma de Derenzo [1] y simulada en MATLAB.

Siguiendo la descripción y notación empleadas en [4], una vez descritos los posibles eventos que un escáner es capaz de detectar, se deduce que PET funciona contando eventos aleatorios pero coincidentes entre pares de detectores en posiciones similares a las representadas en los anillos

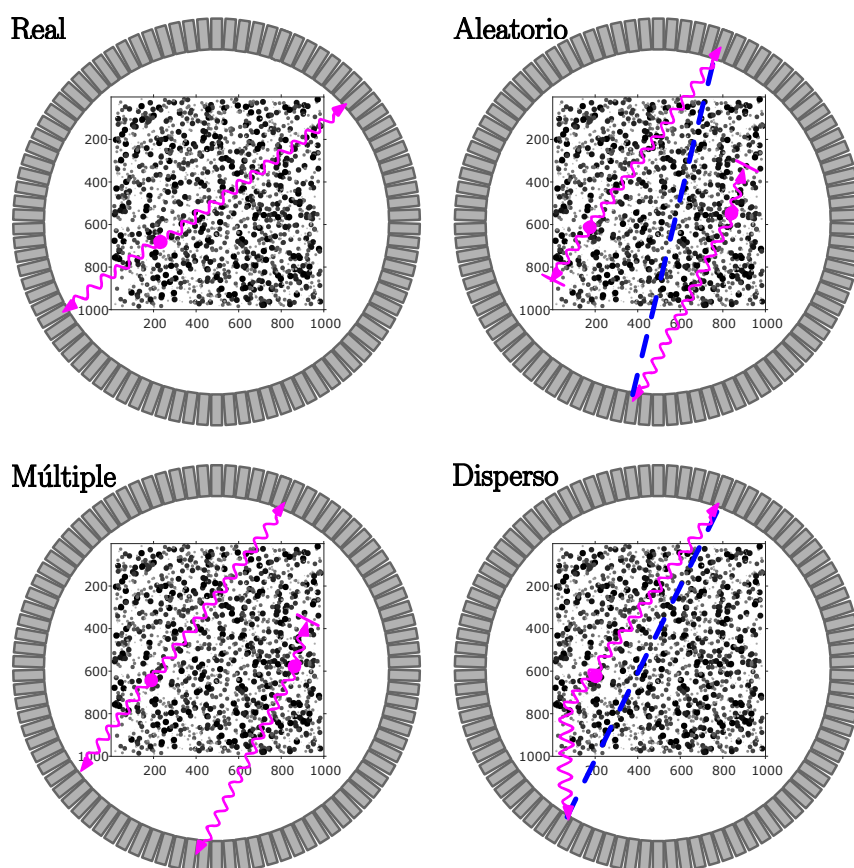


Figura 2.6: Tipos de eventos en PET. Después del fenómeno de aniquilación electrón-positrón, un evento puede ser registrado por coincidencia: real, aleatoria, múltiple o dispersa.

⁷El *efecto Compton* es la interacción entre un fotón y un electrón orbital ligeramente unido. El electrón está tan ligeramente conectado al átomo que puede considerarse esencialmente libre. Este efecto es predominante en el tejido humano [4].

de la Figura 2.6 en un “campo de visión”. La línea recta que conecta los centros de dos detectores se llama *línea de respuesta* (LOR, por sus siglas en inglés). Los pares de fotones no dispersos registrados para una LOR específica, provienen de eventos aleatorios de aniquilación, ubicados en una vecindad “adecuada”; generalmente el volumen de un paralelepípedo alargado, conocido como “tubo de respuesta”.

2.6.1. Modelo probabilístico

A diferencia de CT, para la cual se emplea un modelo determinista. Para PET, por la naturaleza cuántica del fenómeno que se aprovecha, hace falta un modelo indeterminista. A la región del espacio que un escáner PET puede abarcar dentro de su anillo de sensores denominamos *campo de visión* (FOV, por sus siglas en inglés) y la notemos $\Omega \subset \mathbb{R}^n$, con $n \in \{2, 3\}$. Cada par de detectores $d_a, d_b \in \mathbb{R}^n$, vistos como puntos definen una LOR, notada $L_{a,b}$ y además, tienen asociada una *función de sensibilidad* $\psi_{a,b} : \Omega \rightarrow \{0, 1\}$, definida para cada punto en el espacio $x \in \mathbb{R}^n$ que ocupa el dominio. Cada par de detectores que definen a $L_{a,b}$, permiten realizar un conteo de los eventos de aniquilación especificados en la subsección anterior, por tanto, este fenómeno de conteo se modela como una variable aleatoria $f_{a,b}$ con distribución de Poisson, cuya esperanza viene dada por

$$\mathbb{E}(f_{a,b}) = \tau \int_{\Omega} u(x) \psi_{a,b}(x) dx, \quad (2.18)$$

donde τ es el tiempo de adquisición y $u : \Omega \rightarrow \mathbb{R}$ denota la *concentración* (distribución o densidad) del trazador. Se asume que esta concentración es estacionaria en su soporte, mientras que para un radio adecuado R_F al interior del campos de visión, la concentración desaparece, es decir, se tiene que $f(x) = 0$ cuando $\|x - O\| > R_F$, donde O es el origen del sistema de referencia ubicado en el centro del anillo del escáner. El número de parejas de detectores (d_a, d_b) está en el orden de $N_{\text{LOR}} \geq 10^9$ para los escáneres modernos [4].

2.6.2. Formulación lineal inversa

El problema de reconstrucción PET, consiste entonces en el problema inverso de recuperar $u(x)$ para cada $x \in \Omega$ a partir de los datos adquiridos $f_{a,b}$ en cada línea de respuesta. Pero para llegar a un modelo práctico caben aún algunas aproximaciones más.

Supongamos que los datos en la ecuación (2.18) han sido previamente corregidos para varios fenómenos aleatorios, dispersión y atenuación. Además, que cada “tubo” (volumen) de respuesta se modela como una línea recta que une los centros de la cara frontal de ambos cristales en los detectores, como se ilustra en la Figura 2.7. Entonces se sigue que, la función de sensibilidad tiene el siguiente comportamiento

$$\psi_{a,b}(x) = \begin{cases} 1, & x \in L_{a,b} \\ 0, & x \notin L_{a,b}. \end{cases}$$

Dada esta aproximación, la esperanza para los datos PET (2.18) se modelan como integrales de línea de la distribución del trazador, es decir,

$$\mathbb{E}(f_{a,b}) = \int_{L_{a,b}} u(x) dx. \quad (2.19)$$

Esta aproximación por la integral de línea para los datos PET, es una estimación [4] de la transformada de Rayos X tridimensional para la LOR que conecta un par de detectores (ver Definición 13), es decir,

$$f_{a,b} \approx \mathbf{P}u(\theta, s) = \int_{-\infty}^{\infty} u(s \cos(\theta) - t \sin(\theta), s \sin(\theta) + t \cos(\theta), z_0) dt,$$

considerando únicamente la sección transaxial dada por $x_3 = z_0$ donde se realiza la medición por el anillo de detectores, tal como se ilustra en la Figura 2.7. Donde los parámetros θ y s corresponden a la posición radial y angular de la recta $L_{a,b}$. Por último, dado que en dos dimensiones, la transformada de Rayos X coincide con las transformada de Radon (2.15), se sigue que $f_{a,b} \approx \mathbf{R}u(\theta, s)_{a,b}$ [4], y por tanto

$$\mathbb{E}(f_{a,b}) = \mathbb{E}(\mathbf{R}u(\theta, s)_{a,b}) = \mathbf{R}u(\theta, s)_{a,b}. \quad (2.20)$$

La notación $\mathbf{R}u(\theta, s)_{a,b}$ se emplea para recalcar que los parámetros θ y s dependen de la posición del par de detectores d_a y d_b . Si n es el número de posibles combinaciones en el FOV de los detectores y por tanto de sus rectas asociadas, entonces podemos reescribir (2.20) como $\mathbb{E}(f_k) = \mathbf{R}u(\theta, s)_k$, y por tanto $f_k \sim \text{Pois}(\mathbf{R}u(\theta, s)_k)$ para $k \in \{1, \dots, n\}$. Luego empleando notación vectorial para el vector aleatorio de los datos $f = (f_1, \dots, f_n)$ y para la discretización $\mathbf{R}u(\theta, s) = (\mathbf{R}u(\theta, s)_1, \dots, \mathbf{R}u(\theta, s)_n)$, donde $\mathbf{R}u(\theta, s)_k$ es la transformada de Radon de u en la k -ésima recta $L(\theta_k, s_k)$, tenemos

$$f \sim \text{Pois}(\mathbf{R}u). \quad (2.21)$$

Luego de esta breve descripción de PET, nos adherimos a la notación propuesta en [15], para lo cual, sea $u : \Omega \subset \mathbb{R}^l \rightarrow [0, \infty)$ la *actividad del trazador*, donde $l = 2, 3$. Buscamos reconstruir u en un dominio discretizado, para ello simplemente consideramos $u \in \mathbb{R}^n$. Consideramos además el operador directo discreto de PET $\mathcal{A} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ que describe la probabilidad de que se detecte un par de fotones en uno de los pares de detectores, es decir, si $A = (a_{ij}) \in \mathbb{R}^{k \times n}$ es la matriz asociada al operador \mathcal{A} , la entrada a_{ij} es la probabilidad de que un evento en el voxel j sea detectado por el i -ésimo par de detectores.

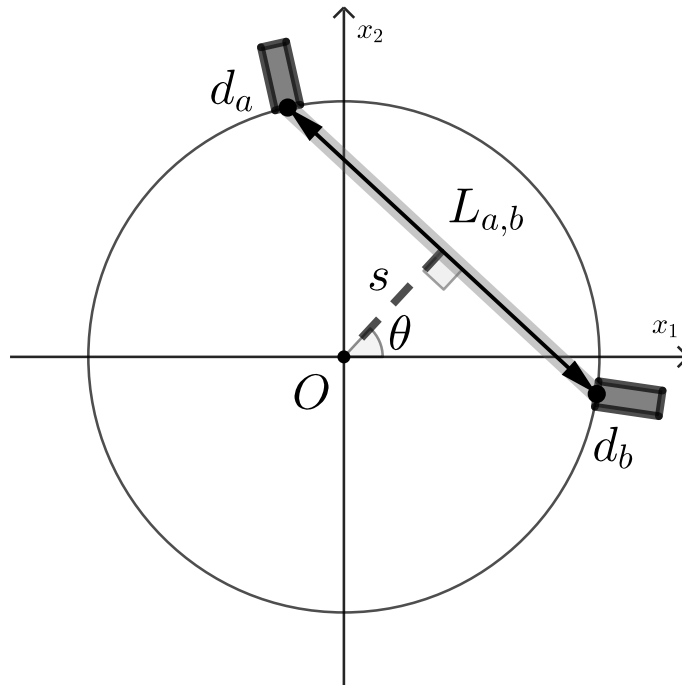


Figura 2.7: Esquema de escaner de anillo PET.

En PET, el objetivo es reconstruir la actividad del radioisótopo, denotada como $u(x)$, en un dominio tridimensional discretizado. Algebraicamente esto se expresa como encontrar el vector u tal que:

$$Au = f \quad (2.22)$$

donde f es el vector de datos PET adquiridos y A es la matriz del sistema que relaciona la actividad con los datos medidos, y por tanto (2.22) determina un problema de medida indirecta. Se asume que estos datos siguen una distribución de Poisson, con una esperanza determinada por el producto Au . Esta distribución de Poisson proporciona una representación probabilística de los datos capturados en un escáner PET [15], es decir,

$$f \sim \text{Pois}(Au). \quad (2.23)$$

La diferencia entre (2.21) y (2.23), es que la primera es la versión continua y la segunda la discreta. En el contexto continuo, la transformada de Radon se utiliza para describir la formación de la imagen en PET. Esta transformada se puede expresar como:

$$\mathbf{R}_{\mu,c}u(\theta, s) = \int_{L(\theta,s)} u(x) \exp\{-[\mathbf{D}\mu(\theta^\perp, x) + \mathbf{D}\mu(-\theta^\perp, x)]\} dx, \quad (2.24)$$

donde θ es el ángulo que determina la dirección de los fotones, s es un parámetro que representa la posición con el ángulo θ , $u(x)$ es la distribución de actividad del radioisótopo y $\mu(x)$ es el *coeficiente de atenuación* del tejido. La matriz \mathbf{D} es el operador de atenuación, que relaciona el coeficiente de atenuación con la posición en el espacio. $L(\theta, s)$ es la línea $x \cdot \alpha(\theta) = s$, donde $\alpha(\theta) = (\sin(\theta), \cos(\theta))$. Y el término $\mathbf{R}_\mu(\theta, s)$ representa la transformada de Radon de la atenuación.

En conclusión, la reconstrucción de la imagen PET se reduce a resolver el sistema asociado al operador \mathcal{A} dado por

$$\mathcal{A}u = \mathbf{R}_{\mu,c}u(\theta, s) = \exp\{-\mathbf{R}_\mu(\theta, s)\}\mathbf{R}u(\theta, s),$$

donde \mathbf{R} es la transformada de Radon clásica [31].

2.6.3. Datos sintéticos

A partir de la discretización de la transformada de Radon presentada en la Subsección 2.4.3 como el modelo discreto para la tomografía computarizada, notemos que, la función de distribución acumulada de probabilidad en (2.19) tiene la misma forma que (2.13) para la cuadratura de la integral. Para simular datos PET, debemos empezar por aplicar una transformada de Radon a una imagen de prueba. Para la imagen de prueba PET, se emplea un fantoma de “Shepp–Logan” modificado con elementos circulares y rectangulares, que permitan probar posteriormente, en el Capítulo 4, la recuperación de vértices en la reconstrucción conjunta de imágenes. A esta imagen digital, ilustrada en la Figura 2.8 y representada por u , llamaremos *fantoma PET*, cuya implementación se encuentra en el código `fantoma`.

Aprovechando los recursos disponibles en MATLAB, sobre la imagen u se aplica el método `radon` con 300 ángulos tomados de [14], a cuyo resultado se denomina *sinograma* y se ilustra en la parte superior derecha de la Figura 2.8. El desenfoque por convolución se agrega a través de `imfilter` y el PSF provisto en [14]. De forma similar, el escalamiento al

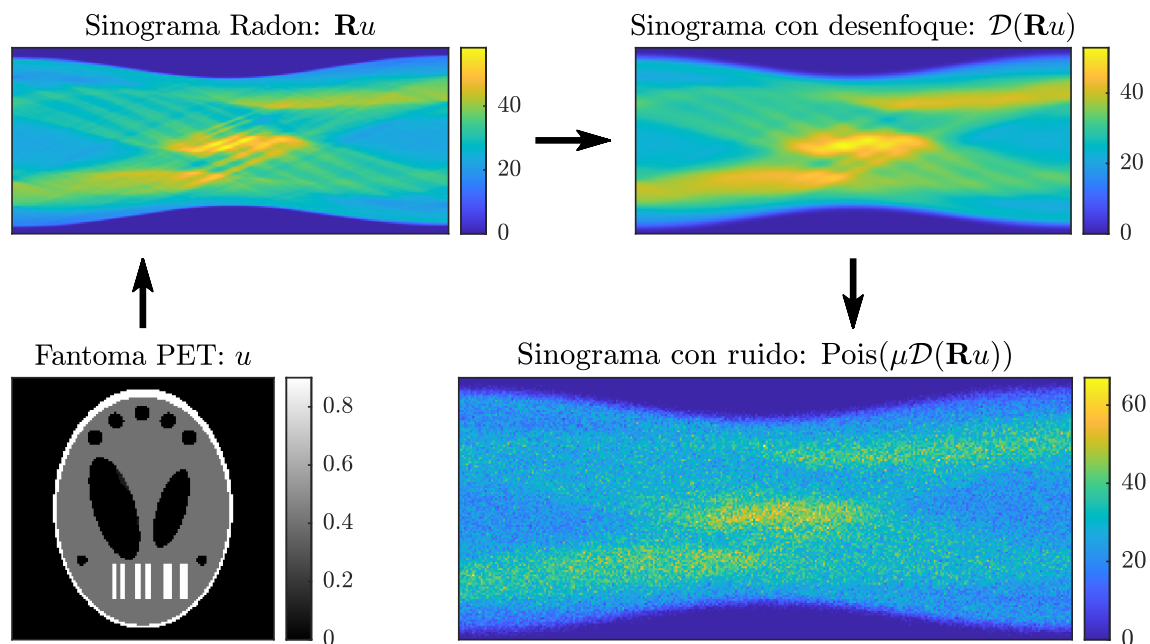


Figura 2.8: Proceso de generación de datos sintéticos PET representados a través de un sinograma (desenfoque y ruido). El sinograma indica el número de coincidencias medidas en cada línea $L_{a,b}$. El ancho del sinograma depende del número de ángulos considerados y su altura, del número de líneas observadas en cada ángulo.

sinograma desenfocado viene dado por el factor también tomado de [14], con ello se simula la sensibilidad presente en (2.18). Finalmente para convertir este sinograma desenfocado y escalado en una variable aleatoria Poisson se emplea `poissrnd` como simulación de los datos PET, ilustrado en la parte inferior de la Figura 2.8.

La generación de los datos sintéticos para las pruebas posteriores de reconstrucción se implementan con los parámetros tomados de [14] en `generar_data.m`.

2.7. Imagen por Resonancia Magnética (MRI)

La Resonancia Magnética (MRI) es una técnica utilizada para obtener imágenes detalladas del interior del cuerpo humano u otros materiales que aprovecha las propiedades del espín de los núcleos de hidrógeno (^1H) en el cuerpo humano cuando son sometidos a un campo magnético. Es además, una de las modalidades de imágenes médicas más versátiles debido a su capacidad para mostrar diferentes características anatómicas y funcionales modificando propiedades magnéticas y de radiofrecuencia en el resonador. Motivo por el que se emplea comúnmente en una variedad de aplicaciones médicas, como la detección de tumores, el diagnóstico de esclerosis múltiple y epilepsia, la evaluación de accidentes cerebrovasculares isquémicos y la identificación de estenosis o aneurismas mediante angiografía por resonancia magnética (MRA). También se utiliza en el campo de la cardiología, el sistema musculoesquelético y el estudio del funcionamiento del cerebro a través de la “RM funcional”. Además, la cirugía guiada por RM, como “MRI-Linac” entre otras [50].

2.7.1. Modelo dinámico

La descripción del modelo matemático para MRI no es tan intuitiva como las de CT o PET, por lo que cabe previamente una descripción de los principios físicos que rigen el fenómeno magnético que se aprovecha en RM. Siguiendo la exposición de Van Leeuwen [50], durante una exploración por resonancia magnética, el vector de magnetización es rotado por el campo magnético aplicado, cambiando ambos en el tiempo. Los

diferentes tipos de tejido tienen diferentes comportamientos de *desintegración* (o relajación) debido a sus propiedades y es la interacción entre estas dos dinámicas, la que permite generar una imagen. Así la cantidad física “central”, es la *magnetización*, una función que varía en el espacio-tiempo, notada como $m : \mathbb{R}^3 \times [0, +\infty) \rightarrow \mathbb{R}$ que describe el efecto neto de los momentos magnéticos nucleares, principalmente de los átomos de hidrógeno presentes en el cuerpo. De forma similar, se representa por $b : \mathbb{R}^3 \times [0, +\infty) \rightarrow \mathbb{R}$ al *campo magnético* externo $b(t) = (b_x(t), b_y(t), b_z(t))$ producido por el resonador, luego el efecto de la magnetización m producida por la presencia del campo magnético b incidente en un cuerpo sometida al mismo, está dado por la ecuación (diferencial parabólica con condiciones iniciales) de Bloch:

$$\frac{dm}{dt} = \begin{pmatrix} -1/T_2 & \gamma b_z(t) & -\gamma b_y(t) \\ -\gamma b_z(t) & -1/T_2 & \gamma b_x(t) \\ \gamma b_y(t) & -\gamma b_x(t) & -1/T_1 \end{pmatrix} m(t) + \begin{pmatrix} 0 \\ 0 \\ \rho/T_1 \end{pmatrix}, \quad m(0) = \begin{pmatrix} 0 \\ 0 \\ \rho \end{pmatrix}, \quad (2.25)$$

donde los parámetros T_1, T_2 y ρ corresponden a *propiedades del tejido* y varían dependiendo del tipo y estructura del mismo, γ es una constante física llamada *relación giromagnética*. Se suele representar a la ecuación (2.25) en su notación lineal, bajo la correspondientes asignaciones, como

$$\frac{dm}{dt} = A(t)m(t) + d, \quad m(0) = m_0. \quad (2.26)$$

Una descomposición adecuada de la matriz $A(t)$, separando las variables de los parámetros, permite identificar las interacciones involucradas, si

$$A(t) = \begin{pmatrix} 0 & \gamma b_z(t) & -\gamma b_y(t) \\ -\gamma b_z(t) & 0 & \gamma b_x(t) \\ \gamma b_y(t) & -\gamma b_x(t) & 0 \end{pmatrix} + \begin{pmatrix} -1/T_2 & 0 & 0 \\ -\gamma b_z(t) & 0 & 0 \\ 0 & 0 & -1/T_1 \end{pmatrix}$$

entonces, la matriz de la izquierda (sesgada simétrica), produce una rotación sobre $m(t)$ alrededor del campo magnético b , mientras la matriz a la derecha describe dos “desintegraciones” exponenciales independientes, una para $m_{xy} = m_x + im_y$ con tasa de desintegración $-1/T_2$ y otra para m_z con tasa $-1/T_1$. A m_z se denomina *componente longitudinal*, correspondiente a la componente de la magnetización generada por el imán del

resonador en la dirección de los pies a la cabeza pero que no podemos medir al no tener recopilación de su señal, mientras la *componente transversal* m_{xy} tiene una distribución aleatoria en un volumen “pequeño”.

Durante un escaneo de resonancia magnética, se alcanza una condición de estado estacionario en la que el valor de m_{xy} no cambia, si esto se mantiene durante un tiempo adecuado para captar los datos y si consideramos un pequeño volumen *voxel* en la posición $r = (x, y, z)^T$, entonces se logra “codificar” la información espacial relativa a todas las posiciones de la *imagen* $v(r) = m_{xy}(r)$. Hay dos tipos de campos magnéticos superpuestos: los *campos de radiofrecuencia (RF)* (componentes transversales b_x y b_y) y los *campos de gradiente* (componente longitudinal b_z con forma de gradiente $b_z(t, r) = g(t) \cdot r$ donde $g = (g_x, g_y, g_z) \in \mathbb{R}^3$). En el proceso, estos dos tipos de campos no se reproducen continuamente sino que se alternan. Los campos RF se utilizan para llevar la magnetización a un estado deseado, mientras que los campos de gradiente se utilizan para codificar información y recopilar una señal que pueda reconstruirse en una imagen.

Si suponemos que al inicio de la medición, es decir, cuando $t = t_0$, se tiene que $d = 0$ y se produce un estado estacionario $m_{xy} = v$; bajo el efecto de los campos de gradiente [50], la ecuación (2.26) tiene por solución

$$m_{xy}(t, r) = v(r) e^{-i\gamma \int_{t_0}^t g(t') \cdot r dt'}.$$

2.7.2. Formulaci3n lineal inversa

En los dispositivos MRI, se emplean bobinas receptoras de radiofrecuencia para capturar la cantidad total de magnetizaci3n transversal presente en el objeto escaneado. Por la ley de inducci3n de Faraday, se sigue que la se1al en el tiempo $s : [0, +\infty) \rightarrow \mathbb{R}$ registrada por estos sensores (bobinas) es directamente proporcional a la integral de volumen de todas las aportaciones locales (bajo el efecto de los campos de gradiente) de la magnetizaci3n transversal $m_{xy}(t, r) = v(r) e^{-i\gamma \int_{t_0}^t g(t') \cdot r dt'}$, es decir

$$s(t) \propto \int_{\mathbb{R}^n} m_{xy}(t, r) dr = \int_{\mathbb{R}^n} v(r) e^{-i\gamma \int_{t_0}^t g(\tau) \cdot r d\tau} dr. \quad (2.27)$$

La expresión integral en (2.27) para la señal $s(t)$, mide indirectamente la distribución espacial de v , lo que nos recuerda a (2.6) en el problema de deblurring, (2.8) en CT, y también a (2.18) en PET. Si definimos la *frecuencia espacial* $k : \mathbb{R} \rightarrow \mathbb{R}$ y al operador integral B respectivamente como sigue

$$k(t) = \frac{\gamma}{2\pi} \int_{t_0}^t g(\tau) d\tau \quad \text{y} \quad B_k v(x) = \int_{\mathbb{R}^n} v(x) e^{-2\pi i k(t) \cdot x} dx,$$

entonces (2.27) se escribe como

$$s(t) = \int_{\mathbb{R}^n} v(x) e^{-2\pi i k(t) \cdot x} dx = B_k v(x) \quad (2.28)$$

donde, buscamos determinar v conocido $s(t) = B_k v(x)$ lo que configura plenamente un problema de medida indirecta. Cabe señalar además que (2.28) es justamente una transformada de Fourier (Definición 11) entre la señal y la imagen. Al dominio de esta señal se denomina *espacio- k* .

Siguiendo la notación en [15], representaremos la imagen MRI como $v : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{C}$. Alébricamente buscamos encontrar el vector v que satisface la ecuación:

$$Bv = g, \quad (2.29)$$

donde g es el vector de datos MRI medidos y el operador directo discreto de MRI queda representado por $B : \mathbb{C}^M \rightarrow \mathbb{C}^L \simeq \mathbb{R}^{2L}$ que realiza la transformada de Fourier seguida de un operador de medición. Mediciones que en MRI están sujetas a ruido Gaussiano aditivo, modelado con una desviación estándar $\sigma > 0$ que depende de varios factores, tales como la temperatura, el ancho de banda del receptor y la intensidad del campo [15], con lo que

$$g \sim \mathcal{N}(Bv, \sigma I). \quad (2.30)$$

2.7.3. Datos sintéticos

Como imagen de prueba MRI se ha construido una segunda variante del fantoma “Shepp–Logan”, al que denominamos *fantoma MRI* y notamos v , también implementado en conjunto al *fantoma PET* en `fantoma.m` y representado a la izquierda en la Figura 2.9. Puesto que el modelado

para MRI (2.28), se reduce al empleo de la transformada de Fourier; para una imagen digital, corresponde la transformada discreta de Fourier, pero además se dispone de la transformada rápida de Fourier ambas bien representadas en la literatura [55] [30] y además esta última se encuentra implementada, por ejemplo en MATLAB a través de la función `fft2` para arreglos bidimensionales. Lo que produce un espacio- k con un orden diferente al mostrado en la imagen central de la Figura 2.9 que debe ser reorganizada a través de la función `fftshift` empleada para visualizar una transformada de Fourier con el componente de frecuencia cero en el medio del espectro tal como se observa en el punto más luminoso de las dos recuadros a la derecha. Finalmente se muestra el efecto apenas perceptible de agregar un ruido blanco aditivo y gaussiano del 5%.

De forma similar se han generado los datos para las señales de frecuencia MRI a emplearse en los experimentos de reconstrucción conjunta del Capítulo 4. Implementados en `generar_data.m`, en base a [14].

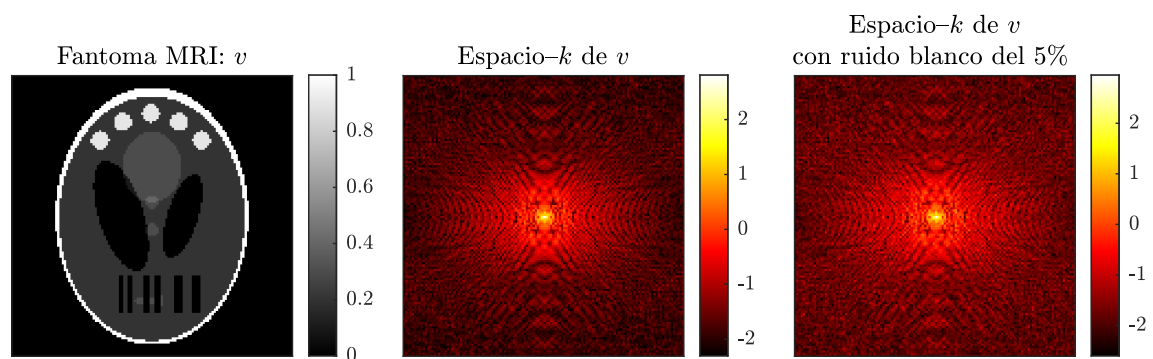


Figura 2.9: Datos sintéticos MRI.

Capítulo 3

Inversión Regularizada

La regularización es una necesidad en problemas inversos que ilustraremos con el ejemplo de la discretización, por una matriz A con dimensión $k \times n$ [39], del operador integral \mathcal{A}_ψ asociado al par convolución–deconvolución (2.4). Aproximando los datos $\mathbf{f} = [f(x_1) \cdots f(x_n)]^T \in \mathbb{R}^n$ para el caso ideal sin ruido, considerando $A\mathbf{f} \approx \mathcal{A}_\psi f = m \in \mathbb{R}^k$ y $\mathbf{f} \approx A^{-1}m$. En la Figura 3.1, se pueden apreciar 4 aproximaciones por discretización, para $k = 32, 64, 128, 256$, de la convolución en una dimensión del pulso f ilustrada en la Figura 2.1. Las gráficas a la izquierda en la Figura 3.2

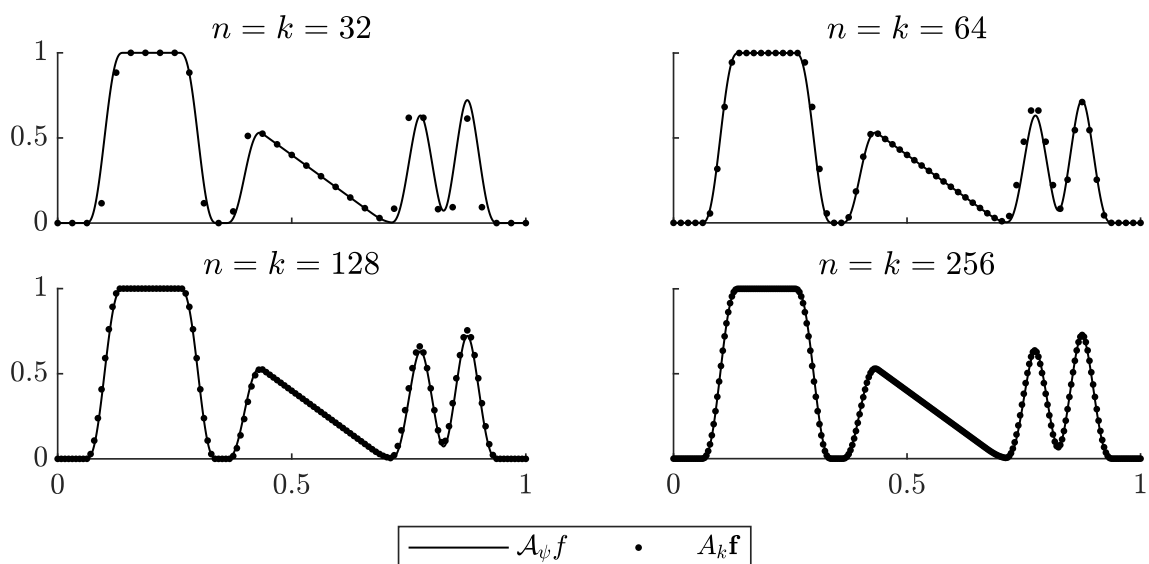


Figura 3.1: Representación gráfica de $\mathcal{A}_\psi f$ en línea delgada, y en puntos, sus aproximaciones $A_k \mathbf{f} = A_k[f(x_1) \cdots f(x_k)]^T$ para valores diferentes de k .

muestran respectivamente las discretizaciones de la convolución Af y de la deconvolución $A^{-1}m$, teniendo a la izquierda casos ideales libres de ruido y a la derecha considerándolo. En ausencia de ruido $m = Af$, entonces la inversa permite una reconstrucción discreta exacta del pulso, es decir, $A^{-1}m = A^{-1}(Af) = f$. En el caso del desenfoque con ruido (2.5), consideremos la aproximación dada por $m = \mathcal{A}_\psi f + \varepsilon \approx Af + \varepsilon$. Al igual que antes, empleando la matriz inversa A^{-1} para determinar los datos f a partir de las mediciones m , se llega al siguiente esquema que Müller y Siltanen [39], denominan *reconstrucción ingenua*:

$$f \approx A^{-1}m \approx A^{-1}(Af + \varepsilon) = f + A^{-1}\varepsilon, \quad (3.1)$$

cuyo resultado se aleja de los datos originales; tal como se observa en el gráfico inferior derecho de la Figura 3.2. La lejanía entre la aproximación dada por el uso ingenuo de la inversa A^{-1} sobre las medidas con ruido $m \approx Af + \varepsilon$ para estimar f , motiva la necesidad de emplear *inversiones regularizadas* que eviten amplificar el ruido y que permitan aproximar la solución de los problemas mal puestos. Esta noción, es más evidente al considerar el error que produce esta aproximación, partiendo de (3.1) consideremos la acotación $\|f - A^{-1}m\| = \|A^{-1}\varepsilon\| \leq \|A^{-1}\|\|\varepsilon\|$. Para el caso ideal sin ruido, cuando $\varepsilon = 0$, tenemos que $A^{-1}m = f$ e independientemente del valor de $\|A^{-1}\|$ no hay error. En la práctica, toda medición siempre incluye errores. Así para valores pequeños de $\|\varepsilon\|$ y no muy grandes de $\|A^{-1}\|$, el error $A^{-1}\varepsilon$ es pequeño, pero para valores grandes de $\|A^{-1}\|$, el error $A^{-1}\varepsilon$ puede ser enorme aún cuando ε sea una pequeña perturbación. Justamente este fenómeno es al que se denomina *amplificación del ruido* y al término $A^{-1}\varepsilon$ *ruido amplificado*. Este es el motivo que nos previene a emplear A^{-1} para reconstruir problemas mal puestos.

Computacionalmente, el modelo de medida continua $m = \mathcal{A}f + \varepsilon$ empleado como generalización para los casos de CT (2.9), PET (2.22) y MRI (2.29) será aproximado a una versión discretizada de la forma $m = Af + \varepsilon$, donde la matriz A es de dimensión $k \times n$, $f \in \mathbb{R}^n$ y $m \in \mathbb{R}^k$ [39].

A continuación presentaremos algunas herramientas que permiten aproximar la solución a estos problemas, donde la presencia de ruido influye de forma determinante asociado a la ausencia de estabilidad. La idea de la regularización será motivada a lo largo del capítulo, presen-

tándola como un problema de minimización. Previo a ello hará falta la definición de *pseudoinversa* y por tanto de factorización matricial *SVD*. A breves rasgos, esta última, consiste de dos matrices ortogonales y una diagonal, que permiten escalar y realizar un doble cambio de bases al ser aplicadas a un vector. Se aprovecha este efecto para disminuir la ampli-ficación de ruidos cuando se trunca la matriz diagonal por un *parámetro de regularización*, este recurso se denomina *TSVD*, por las siglas en inglés de *Truncated singular value decomposition*. Luego, abordamos de lleno la *solución regularizada de Tikhonov* que mejora a TSVD con una ligera modificación.

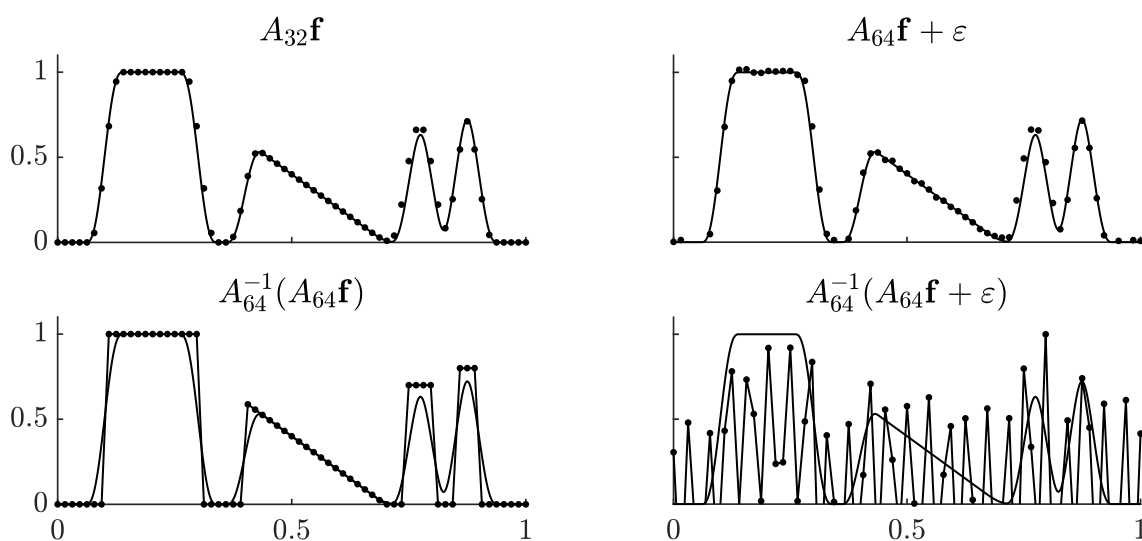


Figura 3.2: En todos los gráficos se representa con línea delgada el pulso convolucionado $\mathcal{A}_\psi f$. La línea punteada, representa las convoluciones discretizadas: a la izquierda sin ruido Af y a la derecha incluyéndolo, $Af + \epsilon$. En la parte inferior en línea con puntos las reconstrucciones ingenuas: a la izquierda en ausencia de ruido $A^{-1}(Af)$ y a la derecha el resultado producido por el ruido amplificado $A^{-1}(Af + \epsilon)$. El ruido simulado de tipo normal con desviación estándar proporcional al 1% de las observaciones.

3.1. Descomposición en valores singulares (SVD)

Definición 14. Sea la matriz $A \in \mathbb{R}^{m \times n}$. Una *descomposición en valores singulares (SVD)* de la matriz A , es la factorización:

$$A = U\Sigma V^T, \quad (3.2)$$

con $U \in \mathbb{R}^{m \times m}$ y $V \in \mathbb{R}^{n \times n}$ matrices ortogonales, es decir, tales que

$$U^T U = U U^T = I, \quad V^T V = V V^T = I,$$

y

$$\Sigma = \begin{pmatrix} D & O_{r \times (n-r)} \\ O_{(m-r) \times r} & O_{(m-r) \times (n-r)} \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad \text{donde } D = \text{diag}(\sigma_1, \dots, \sigma_r).$$

Los elementos σ_j de la diagonal son no negativos y decrecientes, es decir, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ con $r = \max\{j \mid 1 \leq j \leq \min(m, n), \sigma_j > 0\}$, y corresponden a cada j -ésimo *valor singular*, definidos como: $\sigma_j = \sqrt{\lambda_j}$, donde λ_j es un valor propio o autovalor de la matriz $A^T A$, es decir, son las soluciones del polinomio característico $\det(A^T A - \lambda I) = 0$. Por otra parte, $O_{l \times r}$ representa la matriz nula de orden $l \times r$. Los vectores columna de las matrices V y U se denominan *vectores singulares* a la derecha y a la izquierda respectivamente. La inversa de la matriz diagonal D está dada por $D^{-1} = \text{diag}(\sigma_1, \dots, \sigma_r)^{-1} = \text{diag}(1/\sigma_1, \dots, 1/\sigma_r)$, lo que junto a la ortogonalidad de las matrices U y V , motiva la denominación de *pseudoinversa* o *inversa de Moore–Penrose* para la matriz

$$A^+ = V \Sigma^+ U^T, \quad \text{donde } \Sigma^+ = \begin{pmatrix} D^{-1} & O_{r \times (n-r)} \\ O_{(m-r) \times r} & O_{(m-r) \times (n-r)} \end{pmatrix}.$$

Teorema 3. *Toda matriz admite una descomposición SVD.*

Demostración. Sea la matriz $A \in \mathbb{R}^{m \times n}$, y sean $\lambda_1, \dots, \lambda_n$ los autovalores de la matriz $A^T A$. Supongamos que están ordenados en forma decreciente y exactamente r de ellos no son nulos ($r \leq n$). Sea $\{v_1, \dots, v_n\}$ una base ortogonal de \mathbb{R}^n tal que $A^T A v_j = \lambda_j v_j$ para todo $j = 1, \dots, n$. Definimos $V = (v_1 \cdots v_n)$ que es por tanto, una matriz ortogonal ($V^{-1} = V^T$), Σ y D como antes. Para U consideremos $u_j = A v_j / \sigma_j$, ortogonales dos a dos.

Sin pérdida de generalidad, si $r < m$ y por tanto, $n = \min\{m, n\}$, buscamos los vectores u_{r+1}, \dots, u_m de tal modo que $\{u_1, \dots, u_m\}$ es una base ortonormal en \mathbb{R}^m . Luego definimos $U = (u_1 \cdots u_m)$ que es una matriz ortogonal. Por un lado, $AV = (A v_1 \cdots A v_n) = (\sigma_1 u_1 \cdots \sigma_r u_r \ 0 \cdots 0)$, ya que $A v_j = \sigma_j u_j$ para $j = 1, \dots, r$, por la definición de los vectores u_j y $A v_j = 0$,

para $j = r + 1, \dots, n$.

Desarrollando el producto $U\Sigma$, se tiene que

$$U\Sigma = (u_1 \ u_2 \ \dots \ u_m) \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} = (\sigma_1 u_1 \ \sigma_2 u_2 \ \dots \ \sigma_r u_r \ 0 \ \dots \ 0)$$

por tanto, $AV = U\Sigma$ y considerando que $V^{-1} = V^T$, tenemos $A = U\Sigma V^T$. □

Teorema 4. Sea A una matriz $k \times n$, notemos su descomposición SVD por $A = U\Sigma V^T$ y su pseudoinversa por $A^+ = V\Sigma^+ U^T$. Entonces, para $m \in \mathbb{R}^k$

$$\|AA^+m - m\| \leq \|Az - m\|, \quad \forall z \in \mathbb{R}^n.$$

Demostración. [39, Teorema 4.1]. □

Esta interesante propiedad, establece que A^+m minimiza el error en norma, es decir, que $\|AA^+m - m\| \leq \|Az - m\|$ para todo $z \in \mathbb{R}^n$ y en particular $\|AA^+m - m\| \leq \|Af - m\|$. Lo que motiva la siguiente definición relacionada a problemas inversos discretos sin ruido ($Af = m$), por lo que es previa a la idea de regularización ante la presencia de ruido.

Definición 15. El vector $f^* \in \mathbb{R}^n$ se llama *solución de mínimos cuadrados* de la ecuación matricial $Af = m$ si

$$f^* = \arg \min_{z \in \mathbb{R}^n} \|Az - m\|. \quad (3.3)$$

Definición 16. Sea A una matriz de orden $m \times n$, y sean $\sigma_1, \dots, \sigma_r$ sus valores singulares. El *número de condición* de A se denota y define como

$$\text{Cond}(A) = \frac{\sigma_1}{\sigma_r}.$$

Aprovechando que toda matriz admite una descomposición SVD, se

tiene que un problema es mal puesto, cuando los valores singulares de las matrices $(A_k)_{k \in \mathbb{N}}$ que aproximan al operador \mathcal{A} asociado a un problema inverso, decrecen rápidamente a cero. Formalmente, esto implica que un problema inverso no es estable cuando la sucesión de *números de condición* divergen, es decir, $\text{Cond}(A_k) \rightarrow \infty$ [39, sec. 2.3]. En la Figura 3.3 tenemos una representación gráfica en escala logarítmica de los valores singulares de la descomposición SVD para las matrices de orden $2^k \times 2^k$ ($k = 7, \dots, 13$ impares) que aproximan la convolución unidimensional. Para el ejemplo que venimos desarrollando desde el inicio del capítulo, se aprecia que los valores singulares decrecen rápidamente sin llegar a cero conforme el orden de la matriz aumenta.

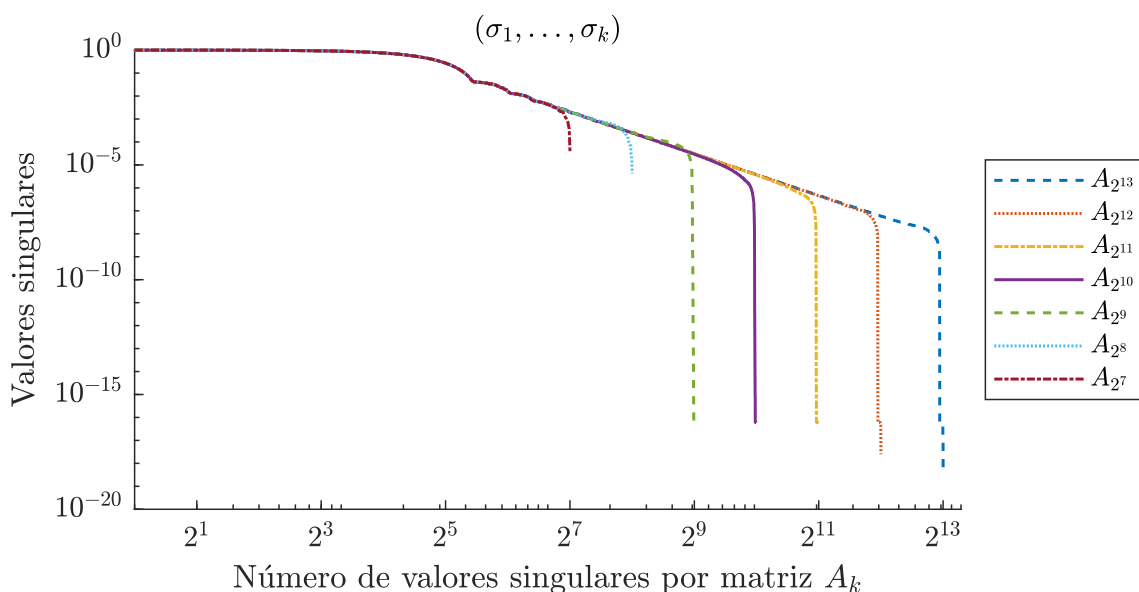


Figura 3.3: Para las matrices $A_{2^7}, \dots, A_{2^{13}}$ que aproximan por discretización al operador de convolución \mathcal{A}_ψ , tenemos, en líneas continuas, la representación de los valores singulares correspondientes a cada matriz.

3.2. Descomposición en valores singulares truncada (TSVD)

Definición 17. Dada una descomposición SVD de una matriz de orden $n \times k$: $A = U\Sigma V^T$. Para todo $\alpha > 0$, definimos la *descomposición en valores*

singulares truncada (TSVD) como la matriz $A_\alpha^+ = V\Sigma_\alpha^+U^T$, donde

$$\Sigma_\alpha^+ = \begin{pmatrix} D_\alpha & O_{r_\alpha \times (k-r_\alpha)} \\ O_{(n-r_\alpha) \times r_\alpha} & O_{(n-r_\alpha) \times (k-r_\alpha)} \end{pmatrix} \in \mathbb{R}^{n \times k}, \quad \text{con } D_\alpha = \text{diag} \left(\frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_{r_\alpha}} \right),$$

y

$$r_\alpha = \min \left\{ r, \max \{ j \mid 1 \leq j \leq \min(k, n), \sigma_j > \alpha \} \right\}.$$

Se denomina *función de reconstrucción*, a la transformación lineal $\mathcal{L}_\alpha : \mathbb{R}^k \rightarrow \mathbb{R}^n$ dada por

$$\mathcal{L}_\alpha(m) = V\Sigma_\alpha U^T m.$$

Las condición de Hadamard se satisfacen para \mathcal{L}_α , por lo que el problema mal puesto $m = Af + \varepsilon$, se transforma en el problema bien puesto,

$$\mathcal{L}_\alpha(m) = V\Sigma_\alpha \Sigma V^T \mathbf{f} + V\Sigma_\alpha U^T \varepsilon.$$

Para la aproximación $V\Sigma_\alpha \Sigma V^T \mathbf{f}$ de \mathbf{f} , tenemos la estimación del error dada por $\|V\Sigma_\alpha U^T \varepsilon\| \leq \sigma_{r_\alpha}^{-1} \|\varepsilon\|$, y puesto que los valores singulares invertidos son crecientes, es decir, $\sigma_1^{-1} \geq \sigma_2^{-1} \geq \dots \geq \sigma_r^{-1}$. Se sigue, que el ruido se amplifica cada vez menos al mantener la menor cantidad de valores singulares o, equivalentemente, al aumentar α .

En la Figura 3.4 se representa a la izquierda el *error relativo* $\frac{\|T(\mathbf{f}) - \mathbf{f}\|_2}{\|\mathbf{f}\|_2} \cdot 100\%$ para $T(\mathbf{f}) = \mathcal{L}_{r_\alpha}(Af + \varepsilon)$, producido por diferentes truncamientos de la función de reconstrucción \mathcal{L}_{r_α} para los valores fijos $r_\alpha = 1, \dots, r = 64$. Se obtiene un error relativo de 16,28% como el mínimo, cuando $r_\alpha = 41$. A la derecha, tenemos justamente la representación gráfica tanto del pulso

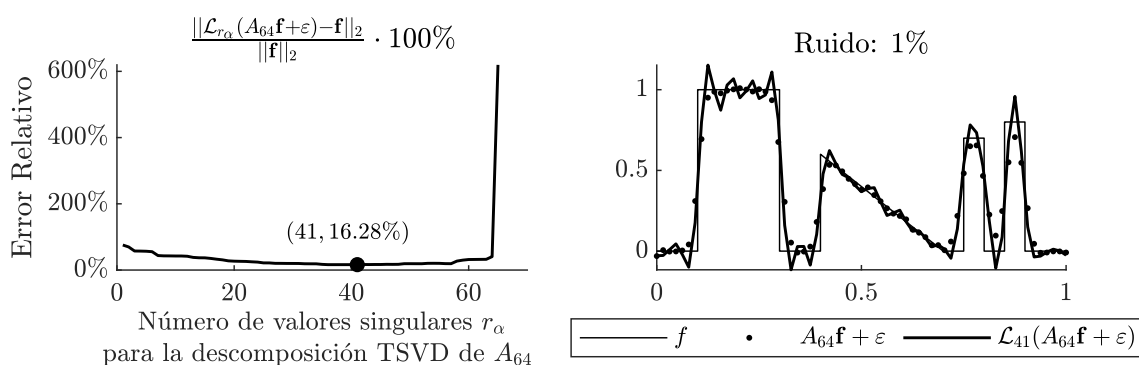


Figura 3.4: Izquierda: error relativo (e.r.) de cada truncamiento posible de la descomposición TSVD. Derecha: reconstrucción TSVD de mínimo e.r. El ruido de 1% empleado para la simulación, es blanco y gaussiano.

f , de la convolución discretizada con ruido $A_{64}f + \varepsilon$ y de la función de reconstrucción $\mathcal{L}_{43}(A_{64}f + \varepsilon)$. Una característica común al resolver problemas inversos es balancear entre la precisión y la robustez al ruido, es decir, elegir adecuadamente el mayor índice de los valores singulares a ser truncado, tal que permita tener el menor error pero también que evite amplificar la mayor cantidad de ruido posible [39, cap. 4].

3.3. Regularización de Tikhonov

Teorema 5. Sea A una matriz de orden $k \times n$, cuya descomposición SVD está dada por $A = U\Sigma V^T$. Para $a > 0$ y $m \in \mathbb{R}^k$ fijos arbitrarios, notemos

$$T_\alpha(m) = V\mathcal{D}_\alpha U^T m, \quad (3.4)$$

al vector en \mathbb{R}^n , con

$$\mathcal{D}_\alpha = \text{diag} \left(\frac{\sigma_1}{\sigma_1^2 + \alpha}, \dots, \frac{\sigma_{\min(k,n)}}{\sigma_{\min(k,n)}^2 + \alpha} \right) \in \mathbb{R}^{n \times k}. \quad (3.5)$$

Entonces, para todo $\mathbf{z} \in \mathbb{R}^n$ se tiene que

$$\|AT_\alpha(m) - m\|^2 + \alpha\|T_\alpha(m)\|^2 \leq \|A\mathbf{z} - m\|^2 + \alpha\|\mathbf{z}\|^2. \quad (3.6)$$

Demostración. [39, Teorema 5.1]. □

La utilidad de este resultado se puede apreciar en la Figura 3.6, donde se representan los diferentes valores de los elementos de las variantes de las matrices diagonales hasta ahora presentadas. Se observa que los valores de la diagonal Σ^+ crecen rápidamente saliendo de los límites del gráfico tan pronto como los valores de Σ se aproximan a cero. El truncamiento de Σ_α^+ se identifica claramente pues siguen el curso de Σ^+ hasta que se hacen todos cero antes de que $\sigma_j < \alpha$, o lo que es lo mismo, antes de que $1/\sigma_j$ alcance a $1/\alpha = 10$. Hemos mencionado que la reconstrucción TSVD disminuye la amplificación del ruido, ahora diremos que la reconstrucción de Tikhonov regulariza la solución.

Definición 18. Para un problema de medida indirecta $m = Af + \varepsilon$, y dado un parámetro de regularización $\alpha > 0$, se define la solución regularizada

de Tikhonov, como el vector $T_\alpha(m) \in \mathbb{R}^n$ tal que

$$T_\alpha(m) = \arg \min_{\mathbf{z} \in \mathbb{R}^n} \{ \|A\mathbf{z} - m\|^2 + \alpha \|\mathbf{z}\|^2 \}. \quad (3.7)$$

Corolario 2. Si $T_\alpha(m)$ es la solución regularizada de Tikhonov para el problema de medida indirecta $m = Af + \varepsilon$, entonces

$$T_\alpha(m) = (A^T A + \alpha I)^{-1} A^T m. \quad (3.8)$$

Demostración. [39, sec. 5.2]. □

Para problemas inversos existen infinitos vectores $T_\alpha(m)$ que permiten obtener en norma L^2 , el menor residuo de $AT_\alpha(m) - m$ y de $T_\alpha(m)$. No

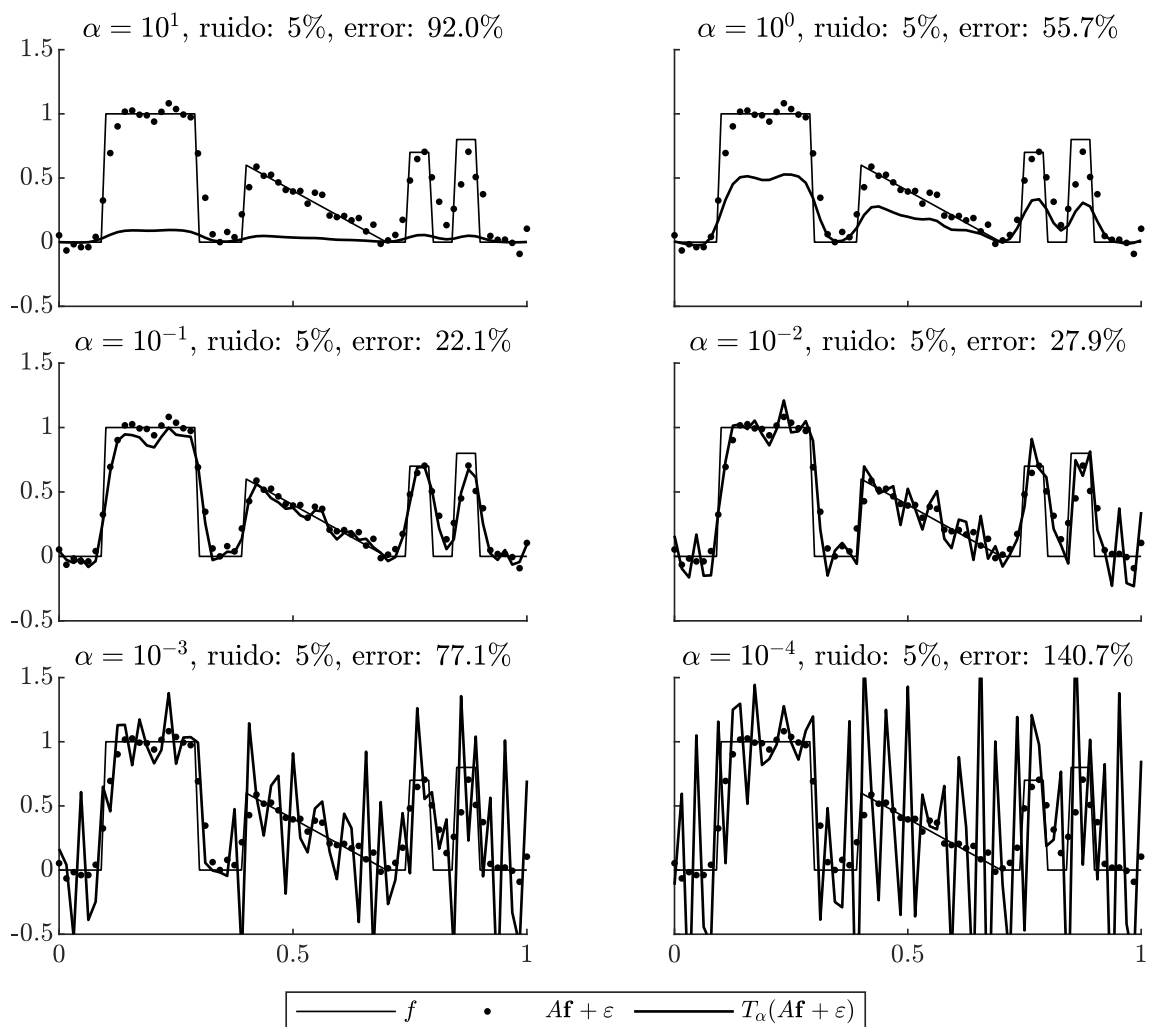


Figura 3.5: Empleos de la solución regularizada de Tikhonov para diferentes valores de α sobre datos con un ruido blanco gaussiano del 5%.

se conoce un método que permita elegir un parámetro de regularización α que produzca una solución óptima [39, p. 63]. En la Figura 3.5 tenemos algunos empleos de la solución regularizada de Tikhonov, sobre la convolución en una dimensión discretiza con ruido; $m = Af + \varepsilon$, es decir, representamos $T_\alpha(m)$ para diferentes valores de α .

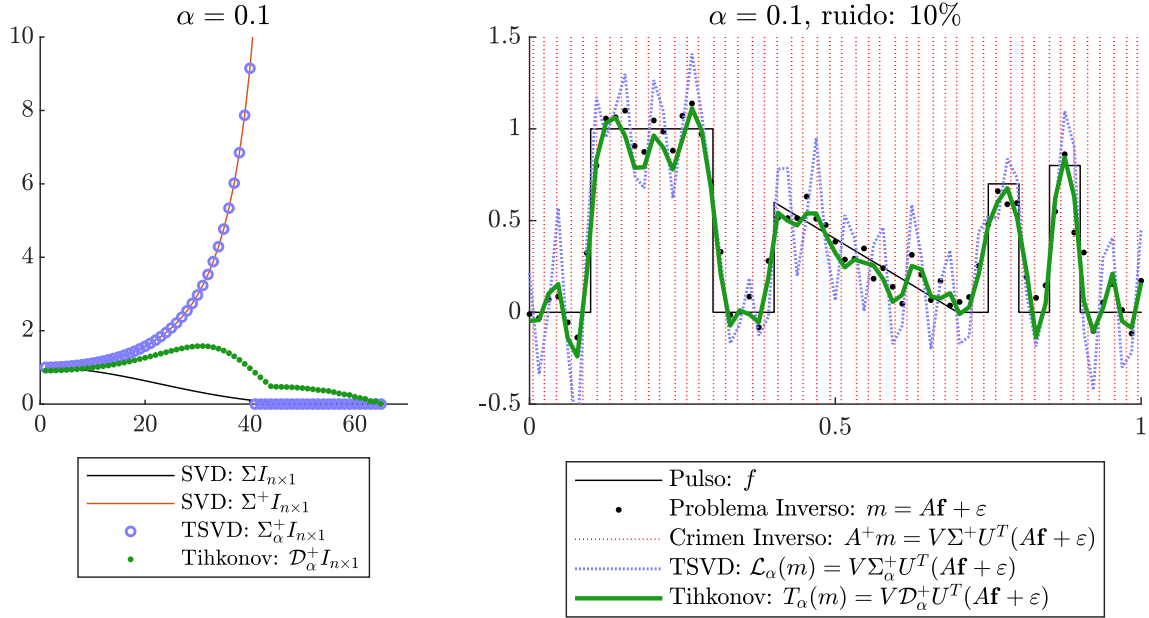


Figura 3.6: Izquierda: variantes de la matriz diagonal. Derecha: comparación entre las reconstrucciones ingenua, TSVD y Tikhonov.

3.4. Regularización de Variación Total

Definición 19. Dada una función a valores reales $f : [a, b] \rightarrow \mathbb{R}$. La *variación total* de f , notada $TV(f)$, se define en su forma discreta como

$$TV(f) = \sup_{\substack{x_i \in [a, b] \\ k \in \mathbb{N}}} \left\{ \sum_{i=1}^k |f(x_i) - f(x_{i-1})| \right\}, \quad (3.9)$$

y en general para un campo escalar continuo definido en un dominio compacto $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$, como

$$TV(f) = \int_{\Omega} \|\nabla f(x)\|_1 dx. \quad (3.10)$$

En atletismo y ciclismo de montaña, el desnivel es una característica a tener muy en cuenta durante las competiciones o entrenamientos,

pues representa la dificultad de la ruta, y justamente es una de las características que se aprecia en estos deportes. La definición de variación total vista como la menor de las cotas superiores para el conjunto de sumas parciales de variaciones de una función, es una magnífica manera de cuantificar y formalizar el concepto de desnivel. En el caso de una función contante por partes con un número finito de discontinuidades, la variación total es la suma del tamaño de los saltos de la función. A diferencia de las regularizaciones TSVD y Tikhonov, que pueden producir reconstrucciones suavizadas, la importancia de la regularización empleando variación total radica en su capacidad de preservar bordes, una propiedad deseada en reconstrucción de imágenes en general; debida a la presencia de fronteras o “desniveles” entre las diferentes secciones que componen una imagen. Para lograr este objetivo, la regularización de la variación total consiste justamente en minimizar TV , lo que en relación a la solución regularizada de Tikhonov (3.7) implica: considerar como *término de regularización* $\alpha\|Lz\|_1$, en vez de $\alpha\|z\|^2$, para un L adecuado. Donde $\|\cdot\|_1$ representa la norma 1, definida como $\|x\|_1 = |x_1| + \dots + |x_n|$, mientras se emplea usualmente $\|\cdot\|$, como la norma euclideana $\|x\| = x_1^2 + \dots + x_n^2$.

Definición 20. Para un problema de medida indirecta $m = Af + \varepsilon$, y dado un *parámetro de regularización* $\alpha > 0$, se define la *solución regularizada de variación total*, como el vector $G_\alpha(m) \in \mathbb{R}^n$ tal que

$$G_\alpha(m) = \arg \min_{z \in \mathbb{R}^n} \{ \|Az - m\|^2 + \alpha \|Lz\|_1 \}, \quad (3.11)$$

donde, para $\Delta x = x_{j+1} - x_j$ y $x_j \in [a, b]$ se define la *matriz de diferencias finitas* por

$$L = \frac{1}{\Delta x} \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & -1 \end{pmatrix}. \quad (3.12)$$

La matriz L se obtiene de encontrar la matriz asociada al sistema lineal $Lf = f'$, dado por las ecuaciones de aproximación de primer orden para la

derivada de la función f ,

$$\mathbf{f}'_j = \frac{df}{ds}(x_j) \approx \frac{f(x_{j+1}) - f(x_j)}{\Delta x} = \frac{\mathbf{f}_{j+1} - \mathbf{f}_j}{\Delta x} = \frac{1}{\Delta x} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{f}_j \\ \mathbf{f}_{j+1} \end{bmatrix}$$

de donde

$$L = \frac{1}{\Delta x} (l_{i,j})_{n \times n}, \quad \mathbf{y} \quad l_{i,j} = \begin{cases} -1, & j = i \\ 1, & j = i + 1 \text{ mód } n \\ 0, & \text{caso contrario.} \end{cases}$$

La elección de la norma $\|\cdot\|_1$ admite reconstrucciones mínimas con funciones constante por partes, mientras que la norma $\|\cdot\|_2$ privilegia acentuadamente las reconstrucciones suavizadas, por sobre las discontinuas [39, p. 85]. Haciendo un símil, con el efecto amplificador de ruido que produce el empleo de la inversa para reconstrucción ingenua de imágenes (ver Figuras 3.2, 3.6, 3.8), también $\|Lf\|_2^2$ amplifica las variaciones de la función f , lo que redundaría en amonestar a soluciones con discontinuidades finitas al momento de buscar una solución óptima $\arg \min_{z \in \mathbb{R}^n} \{\|Az - m\|^2 + \alpha \|Lz\|^2\}$. A ésta última, se denomina *solución regularizada de Tikhonov generalizada*, donde simplemente se exige que L sea una matriz de orden adecuado. Intuitivamente, para los casos de interés,

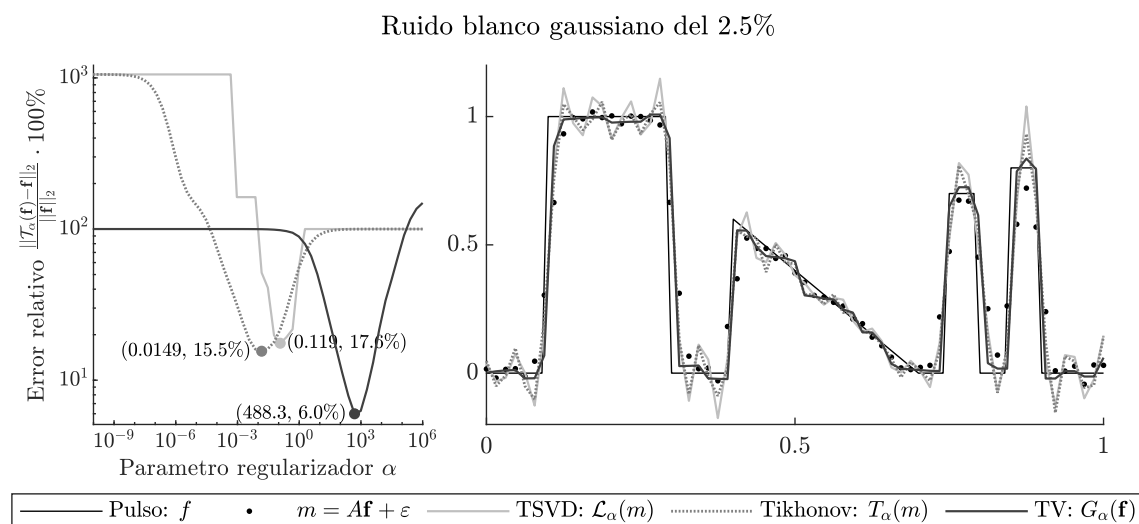


Figura 3.7: Izquierda: Comportamiento del error relativo respecto del parámetro de regularización. Derecha: Reconstrucciones TSVD, Tikhonov y TV de α óptimo para el pulso f .

podemos considerar que $\|Lf\|_1$ tiene menor tamaño que $\|Lf\|_2^2$. En la parte derecha de la Figura 3.7 podemos apreciar como la reconstrucción por regularización de variación total (TV), “preserva” los saltos de discontinuidad de mejor forma que las reconstrucciones TSVD y Tikhonov. Para el caso ilustrado, previamente se ha aproximado el valor óptimo para los parámetros de regularización de cada reconstrucción, partiendo de un valor elevado y arbitrario, para luego realizar cada reconstrucción, calcu-

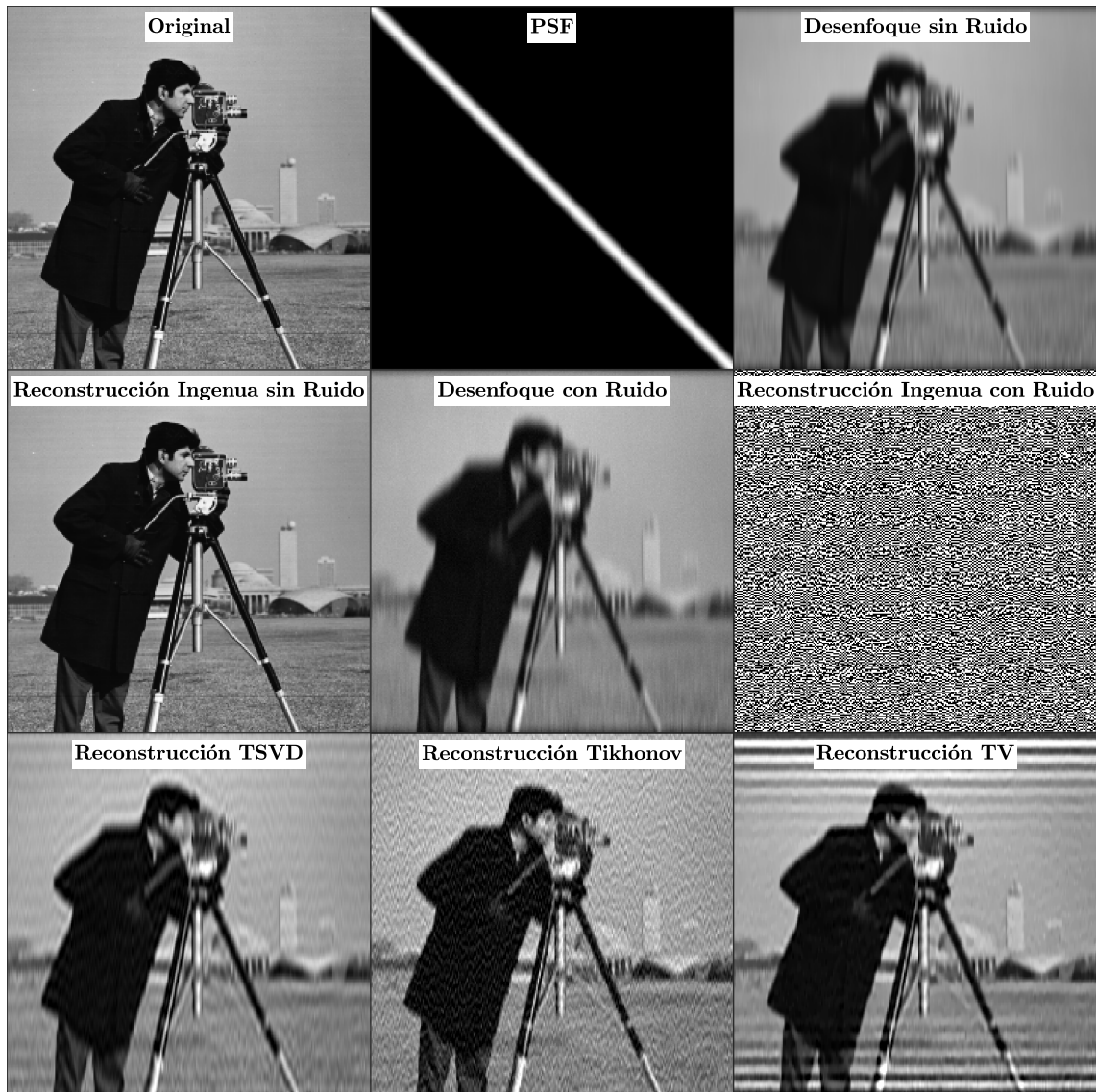


Figura 3.8: Empleo de la imagen `cameraman` de MATLAB etiquetada como “Original”, para mostrar el efecto de desenfoque empleando la convolución con la PSF mostrada y etiqueta como tal, con y sin ruido. Diferentes reconstrucciones, incluyendo las ingenuas con y sin ruido, así como TSVD, Tikhonov y TV. El nivel de ruido blanco gaussiano para el experimento es del 1%. Los parámetros de regularización empleados son: $\alpha_{TSVD} = 0,1$, $\alpha_{Tikhonov} = 0,001$ y $\alpha_{TV} = 20$. Para la regularización TV, se empleo el paquete `tvreg` [18], disponible en [MATLAB Central File Exchange](#).

lar el error relativo y disminuir a la mitad el parámetro. Así es como se ha producido el gráfico a la izquierda y se ha encontrado el mínimo para cada caso, señalado con un punto con sus respectivas coordenadas indicando en la abscisa el valor óptimo de cada parámetro de regularización, y en la ordenada, el error relativo correspondiente a cada reconstrucción de α óptimo. Siendo para el caso ejemplificado y el ruido especificado, la reconstrucción TV la de menor error relativo, cercano al 6%.

En la Figura 3.8 se pueden apreciar algunos experimentos de convolución y deconvolución para una imagen de prueba, empleando los métodos de reconstrucción hasta ahora presentados. En TV aparecen “franjas” horizontales blancas muy marcadas, a estos errores se denomina *artefactos*. Mientras en la Figura 3.9 se puede apreciar el efecto que tiene la norma del gradiente sobre cada punto de las imágenes (fantomas “Shepp–Logan” modificados, el uno para PET y el otro para MRI). Las zonas homogéneas se hacen cero, en contraposición a las regiones de transición que son las

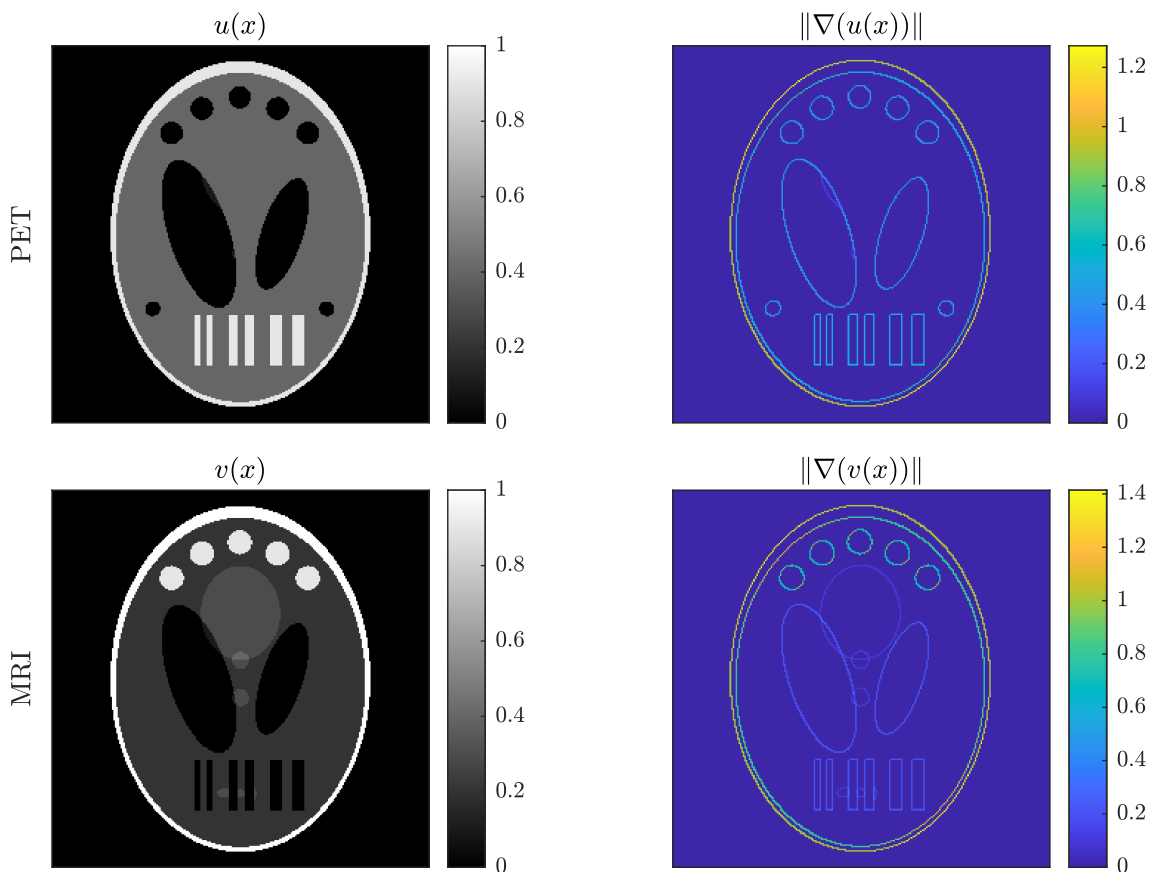


Figura 3.9: Representación gráfica de la norma del gradiente en cada punto del dominio tanto para el fantoma PET y el fantoma MRI definidos en las Subsecciones 2.6.3 y 2.7.3 respectivamente.

únicas que quedan marcadas. A través de la variación total, se busca aprovechar globalmente para toda la imagen, esta propiedad puntual.

3.5. Variación Total Conjunta

Definición 21. Dados dos campos escalares $u, v : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciables. La *variación total conjunta* de (u, v) , notada $JTV(u, v)$, se define como

$$JTV(u, v) = \int_{\Omega} \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right)^{1/2} dx. \quad (3.13)$$

Además se define la *variación total conjunta suavizada* para el parámetro de regularización $\beta > 0$, como

$$JTV_{\beta}(u, v) = \int_{\Omega} \left(\|\nabla u\|^2 + \|\nabla v\|^2 + \beta^2 \right)^{1/2} dx. \quad (3.14)$$

Con fines ilustrativos, para apreciar la “similitud” entre un par de imágenes diferenciables $u, v : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ en un mismo punto x del dominio común Ω , notaremos

$$d_{\nabla}(\nabla(u(x)), \nabla(v(x))) = \|\nabla(u(x))\| + \|\nabla(v(x))\|,$$

y también

$$d_{\nabla, \beta}(\nabla(u(x)), \nabla(v(x))) = \left(\|\nabla(u(x))\|^2 + \|\nabla(v(x))\|^2 + \beta^2 \right)^{1/2}.$$

La variación total favorece la esparcidad en el dominio del gradiente, reforzando la esparcidad en la magnitud del gradiente. De forma similar, la variación total conjunta favorece la esparcidad en la magnitud del “gradiente conjunto” $(\nabla u, \nabla v)$. Con lo cual se incrementa la posibilidad de que los gradientes ocurran en la misma posición [15].

En las expresiones para las soluciones regularizadas de Tikhonov (3.7) y TV (3.11), aparecen los *términos de regularización* $\alpha\|z\|^2$ y $\alpha\|Lz\|_1$ respectivamente. La presencia del término de regularización distingue entre *problemas de minimización regularizados* como los que se acaba de mencionar y los *problemas de minimización no regularizados* como el de mínimos cuadrados (3.3). De forma general, para un dominio compacto $\Omega \subset \mathbb{R}^n$

denotamos por $\mathcal{R} : L^2(\Omega) \longrightarrow \mathbb{R}$ al *operador de regularización* involucrado en el término de regularización. En los ejemplos señalados anteriormente y de forma respectiva, tenemos que $\mathcal{R}(z) = \|z\|^2$ y $\mathcal{R}(z) = \|Lz\|_1$.

Los operadores de regularización basados en la información del gradiente a menudo conducen a una derivada de tipo difusión [15], lo que produce el siguiente resultado.

Proposición 2. *Dados un campo escalar u , y un operador de regularización con la forma $\mathcal{R}(u) = \int_{\Omega} K(\nabla u) \nabla u dx$, donde $K(\nabla u)$ es la difusividad. Entonces la derivada de Gâteaux de \mathcal{R} esta dada por*

$$D \mathcal{R}[u] = -\operatorname{div}(K(\nabla u) \nabla u).$$

Demostración. [15, Sec. 4]. □

Corolario 3. *La derivada de Gâteaux para la variación total conjunta, está dada por*

$$D JTV_{\beta}[u, v] = -\operatorname{div} \begin{pmatrix} \eta(u, v) \nabla u \\ \eta(u, v) \nabla v \end{pmatrix},$$

donde la difusividad conjunta η , se define como

$$\eta(u, v) := \left(\|\nabla u\|^2 + \|\nabla v\|^2 + \beta^2 \right)^{-1/2}.$$

Demostración. [15, Subsec. 4.2]. □

3.6. Conjuntos de Nivel Paralelos

En la sección 1.6 se definió de forma general a una imagen como un campo escalar $u : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$. Desde este punto de vista, el dominio de una imagen es un subconjunto del espacio tridimensional, es decir, que formalmente $\Omega \subset \mathbb{R}^2 \times \mathbb{R} = \mathbb{R}^3$ lo cual corresponde a una interpretación tridimensional para una imagen, en la que además de la posición en el plano o dominio, la tonalidad, luminosidad, intensidad u otras características ópticas o, más generalmente, físicas son representadas a lo largo de una tercera dirección ortogonal a las otras dos.

Exigir además, propiedades como la continuidad o diferenciabilidad sobre este campo escalar al que denominamos imagen, nos permite em-

plear las herramientas que hemos venido usando, entre ellas la variación total y los métodos variacionales. Una nueva herramienta a explotar serán las curvas o conjuntos de nivel. Los conjuntos de nivel al igual que la variación total se pueden asociar o identificar con las tonalidades o en general con la estructura de una imagen, por lo que son significativamente considerables; por ejemplo para buscar o preservar bordes.

Definición 22. Sea $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ un campo escalar y sea $s \in \mathbb{R}$ fijo arbitrario, se denomina *curva* o *conjunto de nivel* a

$$\Omega_s = \{x \in \Omega \mid u(x) = s\}.$$

Para campos escalares diferenciables, se tiene que el gradiente es ortogonal a los conjuntos de nivel, esto hace que las estructuras similares pueden medirse mediante la similitud de la orientación de los gradientes. Así, para dos imágenes, el paralelismo de los gradientes en cada punto es una medida de su similitud estructural [15, Subsec. 3.1.1]. Medir el paralelismo entre dos vectores supone poder medir el ángulo entre ellos, para lo cual se aprovecha el siguiente resultado bien conocido para el producto interno $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ en \mathbb{R}^n [21, 28, 3, 17].

Definición 23. En un espacio euclídeo \mathbb{R}^n dos vectores no nulos son *ortogonales* si el ángulo que forman en el plano que generan es $\pi/2$, y se denominan *paralelos* si el ángulo que forman es 0 ó π .

Teorema 6. Sean x, y dos vectores no nulos del espacio euclídeo \mathbb{R}^n . Si θ es el ángulo entre ellos (medido en el plano generado por x y y), entonces

$$\cos(\theta) = \frac{\langle x, y \rangle}{\|x\| \|y\|}.$$

Demostración. [21, Teorema. 4.2.2]. □

Proposición 3. Sea $(\mathcal{X}, \mathbb{R}, +, \cdot)$ un espacio vectorial con producto interno $\langle \cdot, \cdot \rangle$ y norma $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$. Para $x, y \in \mathcal{X}$ se tiene que

1. Si θ es el ángulo formado por x y y en el plano que generan, entonces

$$|\langle x, y \rangle| = |\cos(\theta)| \|x\| \|y\|. \tag{3.15}$$

2. x, y son ortogonales si y solo si $\langle x, y \rangle = 0$.
3. x, y son paralelos si y solo si $|\langle x, y \rangle| = \|x\| \|y\|$, $x, y \neq 0$.
4. $|\langle x, y \rangle| = \|x\| \|y\|$ si y solo si $\{x, y\}$ es linealmente dependiente.
5. x, y son paralelos si y solo si $\frac{x}{\|x\|} = \frac{y}{\|y\|}$ ó $\frac{x}{\|x\|} = -\frac{y}{\|y\|}$.

Demostración.

1. Se sigue de las propiedades del valor absoluto.
2. En el teorema 6 basta considerar el ángulo recto $\theta = \pi/2$.
3. En (3.15) tomar ángulos planos $\theta = k\pi$, con $k \in \mathbb{Z}$.
4. La demostración consta de dos partes.

Supongamos que $|\langle x, y \rangle| = \|x\| \|y\|$, entonces

$$\begin{aligned} |\langle x, y \rangle| \|y\| &= \|x\| \|y\| \|y\| \\ \|y\| |\langle x, y \rangle| &= \|x\| \|y\|^2 \\ |\langle \|y\|x, y \rangle| &= |\langle \|x\|y, y \rangle|. \end{aligned}$$

Resolver esta ecuación para los valores absolutos, implica abordar los 4 casos posibles. Por la similitud de los mismos, y sin pérdida de generalidad se presenta el caso cuando ambos valores al interior de los absolutos son positivos, es decir

$$\begin{aligned} \langle \|y\|x, y \rangle &= \langle \|x\|y, y \rangle \\ \langle \|y\|x, y \rangle - \langle \|x\|y, y \rangle &= 0 \\ \langle \|y\|x - \|x\|y, y \rangle &= 0 \end{aligned}$$

puesto que x, y son fijos arbitrarios, podemos introducir el cuantificador universal

$$\begin{aligned} \langle \|y\|x - \|x\|y, y \rangle &= 0 \quad \forall y \in \mathcal{X} \\ \|y\|x - \|x\|y &= 0. \end{aligned} \tag{3.16}$$

Por definición del neutro multiplicativo escalar en espacios vectoriales, se sigue que el vector nulo es combinación lineal de todo vector.

El caso de interés se da cuando uno de los dos vectores es no nulo. Sin pérdida de generalidad consideremos que $y \neq 0$ y por tanto $\|y\| > 0$, entonces de (3.16) se sigue que $\{x, y\}$ es un conjunto linealmente dependiente.

Para el recíproco, supongamos que $\{x, y\}$ es un conjunto linealmente dependiente, así por definición existen $\alpha, \beta \in \mathbb{R}$ ambos no nulos tales que

$$\alpha x + \beta y = 0,$$

de donde, sin pérdida de generalidad, tomando $\beta \neq 0$ se tiene que

$$y = -\alpha\beta^{-1}x \quad \text{y además} \quad \|y\| = |-\alpha\beta^{-1}| \|x\|. \quad (3.17)$$

De la segunda identidad, multiplicando en cada lado por $\|x\|$ se tiene

$$\|x\| \|y\| = |-\alpha\beta^{-1}| \|x\|^2 = |-\alpha\beta^{-1}| |\langle x, x \rangle| = |\langle x, -\alpha\beta^{-1}x \rangle| = |\langle x, y \rangle|$$

para la última identidad, en la cadena anterior, se empleo la expresión a la izquierda en (3.17).

5. Los 4 casos posibles, mencionados en la demostración anterior se reducen a 2 casos que resultan de forma similar a (3.16) en $\|y\|x - \|x\|y = 0$ y en $\|y\|x + \|x\|y = 0$, de donde se sigue el resultado.

□

Proposición 4. *En el espacio euclídeo \mathbb{R}^n , la función a la que denominaremos medida de paralelismo definida a continuación*

$$d : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$$

$$d(x, y) = \|x\|\|y\| - |\langle x, y \rangle| \quad (3.18)$$

tiene las siguientes propiedades:

1. $d(x, y) \geq 0$, para todo $x, y \in \mathbb{R}^n$.
2. $d(x, y) = 0$ si y solo si el conjunto $\{x, y\}$ es linealmente dependiente.
3. $d(x, y) = d(y, x)$, para todo $x, y \in \mathbb{R}^n$.

4. No se satisface la desigualdad triangular.

Demostración. Por (3.15) y (3.18), para x, y no nulos podemos reescribir

$$d(x, y) = (1 - |\cos(\theta)|) \|x\| \|y\|. \quad (3.19)$$

1. Si $x = 0$ o $y = 0$, entonces $d(x, y) = \|x\| \|y\| - |\langle x, y \rangle| = 0$. Para el caso $x \neq 0$ y $y \neq 0$ se sigue que $0 \leq 1 - |\cos(\theta)| \leq 1$ y por tanto junto con (3.19), se tiene que $d(x, y) \geq 0$.
2. Notemos que $d(x, y) = 0$, es equivalente a que $\|x\| \|y\| = |\langle x, y \rangle|$, luego por 4 en la proposición 3, se sigue la equivalencia dada.
3. La simetría para d se hereda de la simetría del producto interno y de la conmutatividad del producto en \mathbb{R} .
4. Supongamos que se satisface la desigualdad triangular, es decir

$$d(x, y) \leq d(x, z) + d(z, y).$$

Sean $x, y \in \mathbb{R}^n$ distintos y tomemos $z \in \mathbb{R}^n$ como combinación lineal de x , es decir, que $z = \alpha x$ con $\alpha \in \mathbb{R}$. De la propiedad 2 y de la definición, se sigue respectivamente que:

$$d(x, z) = d(x, \alpha x) = 0 \quad \text{y} \quad d(z, y) = d(\alpha x, y) = |\alpha| d(x, y). \quad (3.20)$$

Finalmente tomando $\alpha \in (-1, 1)$, se tiene que $|\alpha| < 1$. Multiplicando por $d(x, y) \geq 0$ a cada lado y junto con (3.20) se sigue que

$$d(x, y) > |\alpha| d(x, y) = d(x, z) + d(z, y).$$

Esta última relación, contradice la desigualdad triangular.

□

Aunque d no sea una métrica, las propiedades que posee permiten que esta *pseudo-distancia* sea capaz de “medir” el paralelismo entre vectores de un espacio euclídeo [15, Subsec. 3.1.2]. Es más, garantiza hacerse cero, para cada par de vectores paralelos.

Definición 24. Dadas dos funciones no negativas, estrictamente crecientes $\varphi, \psi : [0, \infty) \rightarrow [0, \infty)$. Denominamos *medida de paralelismo generalizada* a la función $d_{\varphi, \psi} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ definida por

$$d_{\varphi, \psi}(x, y) = \varphi\left(\psi(\|x\|\|y\|) - \psi(|\langle x, y \rangle|)\right).$$

Proposición 5. En el espacio euclídeo \mathbb{R}^n , la medida de paralelismo generalizada $d_{\varphi, \psi}$, tiene las siguientes propiedades:

1. $d_{\varphi, \psi}(x, y) > 0$, para todo $x, y \in \mathbb{R}^n$.
2. $d_{\varphi, \psi}(x, y) = d_{\varphi, \psi}(y, x)$, para todo $x, y \in \mathbb{R}^n$.
3. $d_{\varphi, \psi}(x, y) = \min_{t \in [0, +\infty)} \varphi(t)$ si y solo si x y y son paralelos.

Demostración. Sean $\varphi, \psi : [0, \infty) \rightarrow [0, \infty)$ dos funciones no negativas y estrictamente crecientes y sean además, $x, y \in (\mathcal{X}, \mathbb{R}, +, \cdot)$, con producto interno $\langle \cdot, \cdot \rangle$ y norma $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$.

1. De la desigualdad de Cauchy–Schwartz y de la monotonía estricta tanto de φ y ψ , se sigue la no negatividad de $d_{\varphi, \psi}$. En efecto,

$$\begin{aligned} |\langle x, y \rangle| &\leq \|x\|\|y\| \\ \psi(|\langle x, y \rangle|) &< \psi(\|x\|\|y\|) \\ \psi(\|x\|\|y\|) - \psi(|\langle x, y \rangle|) &> 0 \\ d_{\varphi, \psi}(x, y) &= \varphi\left(\psi(\|x\|\|y\|) - \psi(|\langle x, y \rangle|)\right) > \varphi(0) \geq 0. \end{aligned}$$

2. De forma similar que con la medida de paralelismo, en este caso la simetría también se hereda del producto interno y de la conmutabilidad del producto en \mathbb{R} .
3. Consideremos la siguiente cadena de equivalencias:

x, y son paralelos	hipótesis
$\Leftrightarrow \langle x, y \rangle = \ x\ \ y\ $	4 en Proposición 3
$\Leftrightarrow \psi(\langle x, y \rangle) = \psi(\ x\ \ y\)$	definición de función
$\Leftrightarrow \psi(\ x\ \ y\) - \psi(\langle x, y \rangle) = 0$	propiedades de la suma
$\Leftrightarrow \varphi\left(\psi(\ x\ \ y\) - \psi(\langle x, y \rangle)\right) = \varphi(0)$	definición de función

Como φ es estrictamente creciente y $[0, +\infty)$ es un conjunto cerrado y acotado inferiormente, entonces φ alcanza su mínimo en 0. Por tanto junto a la cadena de equivalencias anteriores, se concluye que

$$x \text{ y } y \text{ son paralelos si y solo si } d_{\varphi, \psi}(x, y) = \min_{t \in [0, +\infty)} \varphi(t).$$

□

En particular, se puede emplear la medida de paralelismo generalizada para determinar el “paralelismo” entre dos imágenes diferenciables $u, v : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ en un mismo punto x del dominio común Ω , calculando $d_{\varphi, \psi}$ sobre sus gradientes, es decir que

$$d_{\varphi, \psi}(\nabla u(x), \nabla v(x)) = \varphi\left(\psi(\|\nabla u(x)\| \|\nabla v(x)\|) - \psi(|\langle \nabla u(x), \nabla v(x) \rangle|)\right)$$

es una medida de paralelismo puntual entre las imágenes u y v . Luego es necesario definir una “medida” global de la “similitud” entre las dos imágenes [15, Subsec. 3.1.3].

Definición 25. Dadas dos imágenes diferenciables $u, v : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$. El conjunto de nivel paralelo de (u, v) , notado $PLS(u, v)$, se define como

$$PLS(u, v) = \int_{\Omega} d_{\varphi, \psi}(\nabla u(x), \nabla v(x)) \, dx. \quad (3.21)$$

Además se define el conjunto de nivel paralelo suavizado para el parámetro de regularización $\beta > 0$, como

$$PLS_{\beta}(u, v) = \int_{\Omega} d_{\varphi, \psi, \beta}(\nabla u(x), \nabla v(x)) \, dx, \quad (3.22)$$

donde

$$d_{\varphi,\psi,\beta}(\nabla u(x), \nabla v(x)) = \varphi\left(\psi(\|\nabla u(x)\|_{\beta}\|\nabla v(x)\|_{\beta}) - \psi(|\langle \nabla u(x), \nabla v(x) \rangle|)\right), \quad (3.23)$$

y la “norma suavizada” está dada por

$$\|\cdot\|_{\beta} = (\|\cdot\|^2 + \beta^2)^{1/2}. \quad (3.24)$$

Se denomina *conjunto de nivel paralelo lineal* y se nota $LPSL$, cuando

$$\varphi(t) = \psi(t) = t,$$

y se llama *conjunto de nivel paralelo cuadrático*, notado $QPLS$ si

$$\varphi(t) = (1+t)^{1/2} \quad \text{y} \quad \psi(t) = t^2.$$

En las regiones homogéneas del campo escalar v , el gradiente ∇v se anula, en cuyo caso las funciones PLS y PLS_{β} no aportan ninguna información para u . Junto a TV y JTV , se emplean $QPLS$ y $LPLS$ como regularizadores para la reconstrucción conjunta de imágenes [15, Subsec. 3.1.3]. En la Figura 3.10 se representa para las imágenes u y v de los fantasmas “Shepp–Logan” modificados para PET y MRI (ver Figura 3.9) a las “medidas de variación” puntuales empleadas para definir TV y JTV , así como las “medidas de paralelismo” puntuales que definen $QPLS$ y $LPLS$. Se emplea el mismo parámetro β en todos los casos.

Proposición 6. *Notemos por $C_*^1(\Omega)$ al espacio de funciones continuamente diferenciables con condiciones de frontera Neumann cero. Sean φ, ψ dos funciones continuamente diferenciables. La derivada de Gâteaux de $PLS_{\beta} : C_*^1(\Omega) \times C_*^1(\Omega) \rightarrow \mathbb{R}$ definida por (3.22) en (u, v) está dada por*

$$D PLS_{\beta}(u, v) = -\operatorname{div} \begin{pmatrix} K(\nabla u, \nabla v) \nabla u \\ K(\nabla v, \nabla u) \nabla v \end{pmatrix},$$

donde las difusividades están dadas por

$$\begin{aligned}
K(x, y) &= \kappa(x, y)I + \mu(x, y)yy^T \in \mathbb{R}^{N \times N}, \\
\kappa(x, y) &= \rho(x, y)\|y\|_\beta\|x\|_\beta^{-1} \psi'(\|x\|_\beta\|y\|_\beta) \geq 0, \\
\mu(x, y) &= -\rho(x, y)|\langle x, y \rangle|_\beta^{-1} \psi'(|\langle x, y \rangle|_\beta) \leq 0, \\
\rho(x, y) &= \varphi'(\psi(\|x\|_\beta\|y\|_\beta) - \psi(|\langle x, y \rangle|_\beta)) \geq 0.
\end{aligned}$$

Demostración. [15, Lema 4.1, Proposición 4.3]. □

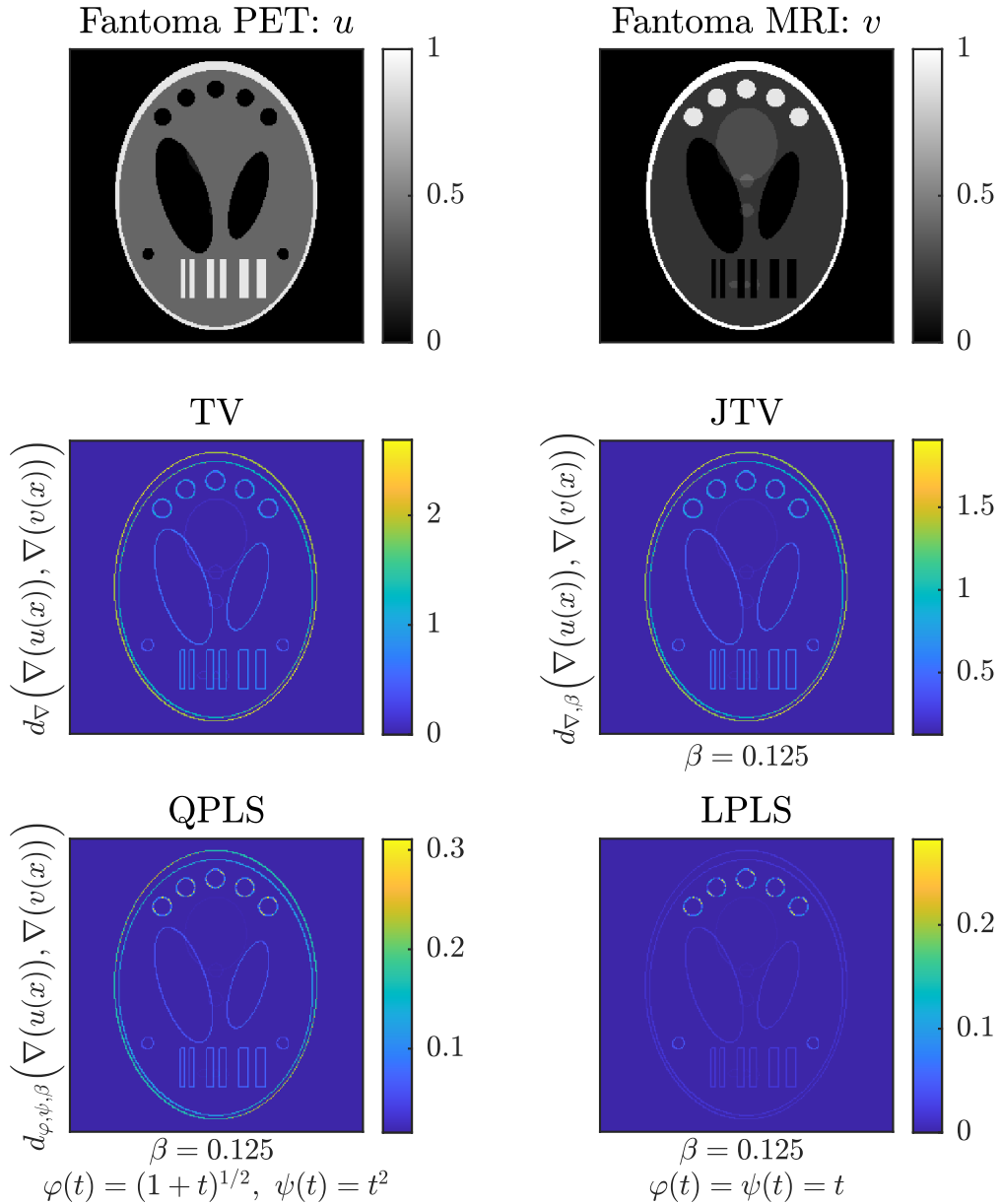


Figura 3.10: Representación gráfica de la “similitud” entre imágenes, dada por las “medidas de variación” asociadas a *TV* y *JTV*, y a las “medidas de paralelismo” asociadas a *QPLS* y *LPLS*.

Capítulo 4

Reconstrucción conjunta de imágenes

Luego de haber abordado en los capítulos anteriores los tópicos en problemas inversos necesarios para modelar tanto PET como MRI, y habiendo presentado la regularización multimodal o conjunta para dos imágenes que comparten una misma estructura. En el presente capítulo abordaremos el problema de la reconstrucción conjunta PET/MRI.

En las secciones 2.6 y 2.7 mencionamos, respectivamente, que PET provee información funcional y MRI información anatómica. Esta observación es fundamental, pues cada una de estas modalidades de imagen médica mide diferentes fenómenos físicos, generando datos muy diferentes los unos de los otros, pero que provienen de una misma fuente. Partiendo del supuesto de que la función y la anatomía del tejido de los organismos vivos están estrechamente relacionados, se pretende reconstruir imágenes PET e imágenes MRI aprovechando la similitud estructural del origen de los datos para ambas modalidades. Esto es posible gracias al desarrollo tecnológico de escáneres capaces de realizar simultáneamente la adquisición de datos PET y MRI. El objetivo final de realizar reconstrucciones conjuntas, es mejorar los resultados obtenidos respecto de las reconstrucciones individuales, es decir, que aprovechando las estructuras comunes, la reconstrucción PET se beneficie de la MRI y viceversa [15].

El presente trabajo busca reproducir y emplear tanto los resultados como los algoritmos desarrollados por Matthias Ehrhardt *et al.* en su tra-

bajo denominado “Joint reconstruction of pet-mri by exploiting structural similarity” [15]. Para realizar los experimentos numéricos se han generado datos sintéticos a partir de imágenes de prueba. Se ha modificado el fantoma “Shepp–Logan” tanto para simular una imagen PET como para simular MRI tal como se especificó en las Subsecciones 2.6.3 y 2.7.3, respectivamente.

Para plantear el problema de la reconstrucción conjunta se recurre a un modelo probabilístico y gráfico como forma de vincular PET y MRI [15], que se expone a continuación.

4.1. Modelo probabilístico para la reconstrucción conjunta

Supongamos que las imágenes PET y MRI, u y v respectivamente son variables aleatorias. Además que ambas dependen de una fuente en común y por tanto no son independientes una de otra. Esta relación se ilustra en la Figura 4.1. Los datos son modelados como variables aleatorias con esperanza linealmente dependiente de las imágenes. En consecuencia, los datos PET f y los datos MRI g no son independientes entre sí. Supongamos también que ningún conocimiento sobre la pareja (u, v) aporta información adicional respecto de f , ni viceversa. Es decir, dados u y v , supongamos que f y g son condicionalmente independientes. Por

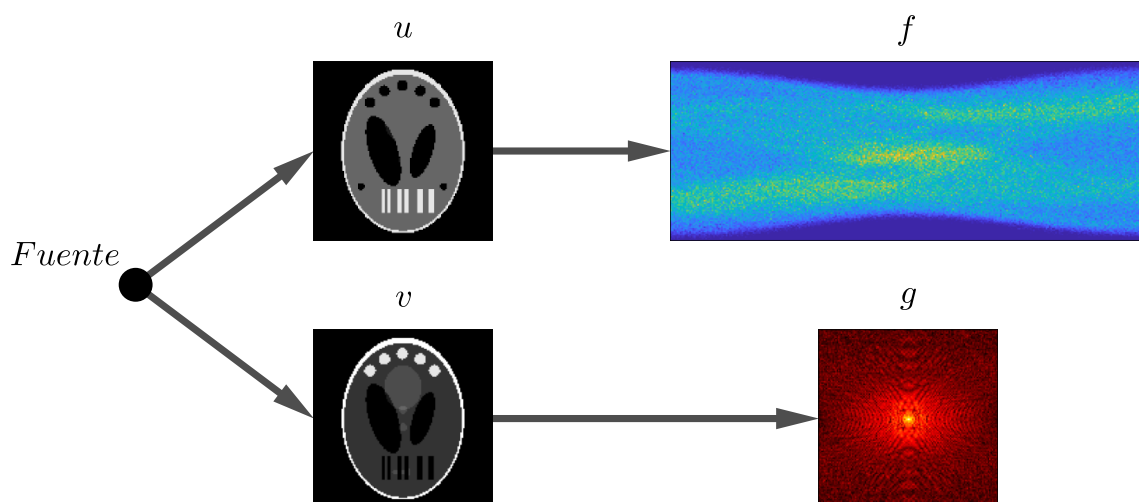


Figura 4.1: Diagrama del modelo de reconstrucción conjunta PET/MRI.

tanto, la verosimilitud multi modal se puede separar [15], es decir

$$\mathbb{P}(f, g | u, v) = \mathbb{P}(f | u, v)\mathbb{P}(g | u, v) = \mathbb{P}(f | u)\mathbb{P}(g | v),$$

luego, por el teorema de Bayes, se tiene que la probabilidad conjunta de la imagen PET/MRI (u, v) dados los datos observados PET/MRI (f, g) está dada por

$$\mathbb{P}(u, v | f, g) = \frac{\mathbb{P}(f, g | u, v)\mathbb{P}(u, v)}{\mathbb{P}(f, g)} = \frac{\mathbb{P}(f | u)\mathbb{P}(g | v)\mathbb{P}(u, v)}{\mathbb{P}(f, g)}.$$

Notemos que el denominador no depende de (u, v) , por tanto, $\mathbb{P}(u, v | f, g)$ es suficiente [26][49][51] para $\mathbb{P}(f | u)\mathbb{P}(g | v)\mathbb{P}(u, v)$, en otras palabras, podemos omitir el denominador [4], con lo cual

$$\mathbb{P}(u, v | f, g) \propto \mathbb{P}(f | u)\mathbb{P}(g | v)\mathbb{P}(u, v). \quad (4.1)$$

A partir de (4.1), la estadística matemática [26][49] nos permite calcular el estimador de máxima verosimilitud conjunta para (u, v) . Como el logaritmo es estrictamente creciente, entonces, el maximizador de $\mathbb{P}(u, v | f, g)$, es el mismo para función de costo

$$\begin{aligned} Q(u, v, f, g) &= \log (\mathbb{P}(f | u)\mathbb{P}(g | v)\mathbb{P}(u, v)) \\ &= \log (\mathbb{P}(f | u)) + \log (\mathbb{P}(g | v)) + \log (\mathbb{P}(u, v)). \end{aligned} \quad (4.2)$$

Supongamos que los datos PET, son realizaciones de una variable aleatoria Poisson con esperanza idéntica al operador asociado al problema inverso PET (2.23), es decir, $f \sim \text{Pois}(Au)$, supongamos además, un ruido blanco gaussiano aditivo en MRI (2.30), es decir, $g \sim N(Bv, \sigma I)$, con $\sigma > 0$ y asumiendo a priori que $\mathbb{P}(u, v) = \exp(-\alpha\mathcal{R}(u, v))$, podemos calcular cada uno de los términos en (4.2).

Del supuesto de que $f \sim \text{Pois}(Au)$, se sigue que, $f_k \sim \text{Pois}((Au)_k)$, luego

$$\mathbb{P}(f_k | u_k) = \frac{1}{f_k} \exp\{-(Au)_k\}(Au)_k^{f_k}.$$

Si los datos f_k son independientes entre si, tenemos que

$$\begin{aligned}\log(\mathbb{P}(f | u)) &= \log\left(\prod_{k=1}^n \mathbb{P}(f_k | u_k)\right) = \sum_{k=1}^n \log\left(\frac{1}{f_k} \exp\{-(Au)_k\}((Au)_k)^{f_k}\right) \\ &= \sum_{k=1}^n f_k \log((Au)_k) - (Au)_k - \log(f_k).\end{aligned}$$

En cambio, de que $g \sim N(Bv, \sigma I)$, se sigue que $g_k \sim N(Bv_k, \sigma)$, de donde

$$\mathbb{P}(g_k | v_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(g_k - Bv_k)^2}{2\sigma^2}\right\}$$

y si asumimos independencia entre cada medición, entonces

$$\begin{aligned}\log(\mathbb{P}(g | v)) &= \log\left(\prod_{k=1}^n \mathbb{P}(g_k | v_k)\right) = \sum_{k=1}^n \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(g_k - Bv_k)^2}{2\sigma^2}\right\}\right) \\ &= \sum_{k=1}^n \log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2} \sum_{k=1}^n (g_k - Bv_k)^2.\end{aligned}$$

Haciendo los reemplazos correspondientes en (4.2), se tiene

$$\begin{aligned}Q(u, v, f, g) &= \sum_{k=1}^n f_k \log(Au)_k - (Au)_k - \frac{1}{2\sigma^2} \sum_{k=1}^n (g_k - Bv_k)^2 \\ &\quad - \log\left(\exp\{\alpha\mathcal{R}(u, v)\}\right) - n \log(\sqrt{2\pi}\sigma) - \sum_{k=1}^n \log(f_k) \\ &= \sum_{k=1}^n f_k \log(Au)_k - (Au)_k - \frac{\|Bv - g\|^2}{2\sigma^2} - \alpha\mathcal{R}(u, v) - \sum_{k=1}^n \log\left(\frac{f_k}{\sqrt{2\pi}}\right),\end{aligned}$$

puesto que $\sum_{k=1}^n \log(f_k/\sqrt{2\pi})$ no depende de (u, v) , podemos despreciar este término al buscar un $(u, v)^*$ que maximice a Q . Bastará considerar como función de verosimilitud equivalente a (4.1), a la definida por

$$\mathcal{L}(u, v, f, g) = \sum_{k=1}^n f_k \log(Au)_k - (Au)_k - \frac{1}{2\sigma^2} \|Bv - g\|^2 - \alpha\mathcal{R}(u, v). \quad (4.3)$$

Por último, como maximizar la verosimilitud en (4.3) es equivalente a minimizar su negativo, buscaremos un $(u, v)^*$ que minimice al siguiente

funcional objetivo

$$\mathcal{J}(u, v) = \sum_i (Au)_i - f_i \log(Au)_i + \frac{1}{2\sigma^2} \|Bv - g\|^2 + \alpha \mathcal{R}(u, v). \quad (4.4)$$

El operador PET A y el operador MRI B son los descritos en las Subsecciones 2.6.2 y 2.7.2, respectivamente. En el caso particular de PET/MRI, ambas imágenes comparten bordes al tratarse de una misma fuente (paciente), por lo que se considera reguladores que impongan a las reconstrucciones a preservar esta propiedad, por tanto, el operador de regularización \mathcal{R} será cualquiera de los presentados en las Secciones 3.5 y 3.6, específicamente variación total conjunta (3.13), (3.14) y conjuntos de nivel paralelo (3.22) en sus versiones lineal y cuadrática. Al estimador buscado, también se conoce como *estimador de máxima verosimilitud conjunta a posteriori* (JMAP, por sus siglas en ingles) [4].

Definición 26. Para un problema de medida indirecta bimodal $h = \mathcal{A}_{A,B}(w)$, donde $h = (f, g)$, $w = (u, v)$ y $\mathcal{A}_{A,B} = (A, B)$. Dado un parámetro de regularización $\alpha > 0$, se define la solución regularizada conjunta, como el vector $h^* = H_\alpha(f, g) \in \mathbb{R}^n$ tal que

$$H_\alpha(f, g) = \arg \min_{(u,v) \in \mathbb{R}^n \times \mathbb{R}^n} \left\{ \mathcal{D}_1(f, A(u)) + \mathcal{D}_2(g, B(v)) + \alpha \mathcal{R}(u, v) \right\}, \quad (4.5)$$

donde \mathcal{D}_1 y \mathcal{D}_2 se denominan *términos de ajuste* y a \mathcal{R} *regularizador conjunto*.

4.2. Optimización numérica

El software desarrollado en [14] para la reconstrucción conjunta de imágenes PET/MRI, emplea el algoritmo L-BFGS-B como método numérico para obtener una solución aproximada de la función de costo (4.4). Este software requiere una interfaz a MATLAB con C [6] desarrollada para adaptar el algoritmo [10] implementado en FORTRAN [54]. L-BFGS-B es un algoritmo cuasi-Newton que extiende al algoritmo L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno)[43] empleado en aplicaciones de memoria computacional limitada, para resolver “grandes” problemas de optimización no lineal con restricciones simples en las variables.

4.2.1. L-BFGS

Los métodos cuasi-Newton de memoria limitada son útiles para resolver *problemas grandes*, por ejemplo, aquellos donde la matriz Hessiana es densa de orden $n \times n$, para un n grande. Se recurre a aproximaciones simples y compactas de esta matriz, guardando únicamente algunos vectores de longitud n que representan las aproximaciones implícitas. Esta estrategia produce tasas de convergencia lineales. El algoritmo L-BFGS, utiliza información de curvatura solo de las iteraciones más recientes para construir la aproximación hessiana. La información de curvatura de iteraciones anteriores se descarta para ahorrar almacenamiento [43].

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ un campo escalar diferenciable, se aproxima numéricamente el mín $f(x)$, alcanzándose el mínimo local bajo condiciones de convexidad y en su ausencia únicamente mínimos relativos.

Cada paso en el método L-BFGS tiene la forma

$$x_{k+1} = x_k - \alpha_k H_k \nabla f_k,$$

donde α_k es la longitud de paso y haciendo que:

$$\begin{aligned} \rho_k &= \frac{1}{y_k^T s_k}, \\ V_k &= I - \rho_k y_k s_k^T, \\ s_k &= x_{k+1} - x_k, \\ y_k &= \nabla f_{k+1} - \nabla f_k, \end{aligned}$$

aquí, $\nabla f_k = \nabla f(x_k)$ y, x_k es la aproximación del mínimo deseado en la k -ésima iteración del algoritmo.

Se almacenan únicamente un subconjunto $\{s_i, y_i\}_{i=k-m}^{k+m}$ del total $\{s_i, y_i\}_{i=1}^n$, para un $m < n$ significativamente menor (entre 3 y 20). En la iteración k , se elige una aproximación inicial de la Hessiana H_k^0 . En la práctica, una elección efectiva [43], es tomar

$$H_k^0 = \gamma_k I, \quad \text{donde} \quad \gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \quad (4.6)$$

para luego realizar la actualización

$$\begin{aligned}
H_k &= (V_{k-1}^T \cdots V_{k-m}^T) H_k^0 (V_{k-m} \cdots V_{k-1}) \\
&\quad + \rho_{k-m} (V_{k-1}^T \cdots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_{k-1}) \\
&\quad + \rho_{k-m+1} (V_{k-1}^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_{k-1}) \\
&\quad + \cdots \\
&\quad + \rho_{k-1} s_{k-1} s_{k-1}^T.
\end{aligned}$$

El cálculo recursivo del producto $H_k \nabla f_k$ se muestra en el Algoritmo 1.

Algorithm 1 Recursión de dos lazos L-BFGS

```

 $q \leftarrow \nabla f_k$ 
for  $i = k - 1, k - 2, \dots, k - m$  do
     $\alpha_i \leftarrow \rho_i s_i^T q$ 
     $q \leftarrow q - \alpha_i y_i$ 
end for
 $r \leftarrow H_k^0 q$ 
for  $i = k - m, k - m + 1, \dots, k - 1$  do
     $\beta \leftarrow \rho_i y_i^T r$ 
     $r \leftarrow r + s_i (\alpha_i - \beta)$ 
end for
detener con resultado  $H_k \nabla f_k = r$ .

```

En general, el paso α_k debe satisfacer las condiciones de Wolfe¹.

Algorithm 2 L-BFGS

```

Elija el punto de partida  $x_0$ , el entero  $m > 0$  (entre 3 y 20);
 $k \leftarrow 0$ ;
repeat
    Elija  $H_k^0$  (por ejemplo, usando (4.6));
    Calcule  $p_k \leftarrow -H_k \nabla f_k$  a partir del Algoritmo 1;
    Calcule  $x_{k+1} \leftarrow x_k + \alpha_k p_k$ , donde  $\alpha_k$  se elige para satisfacer las
    condiciones de Wolfe;
    if  $k > m$  then
        Descarte el par de vectores  $\{s_{k-m}, y_{k-m}\}$  del almacenamiento;
    end if
    Calcule y guarde  $s_k \leftarrow x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$ ;
     $k \leftarrow k + 1$ ;
until convergencia

```

¹Se denomina *condiciones de Wolfe*, a las condiciones suficientes de decrecimiento y curvatura, dadas para $0 < c_1 < c_2 < 1$, tales que: $f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k$ y $\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k$.

La optimización numérica de memoria limitada BFGS se enuncia formalmente en el Algoritmo 2. La elección realizada para el *factor de escalamiento* γ_k en (4.6) garantiza que la dirección de búsqueda p_k esté bien escala y por tanto, que el tamaño del paso $\alpha_k = 1$ sea aceptable para la mayoría de las iteraciones [43].

4.2.2. L-BFGS-B

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ un campo escalar no lineal diferenciable, se busca aproximar una solución al problema de:

$$\begin{aligned} & \text{mín } f(x), \\ & \text{sujeto a: } l < x < u. \end{aligned}$$

Algorithm 3 L-BFGS-B

1. Elija un punto de partida x_0 , y un entero m que determina el número de correcciones de memoria limitada almacenadas. Defina la matriz de memoria limitada inicial como la identidad;
 2. $k \leftarrow 0$;
 3. Calcule la proyección del gradiente $\mathbb{P}(x_k - g_k, l, u)$;
if $\|P(x_k - g_k, l, u) - x_k\|_\infty < 10^{-5}$ **then**
 Detener;
end if
 1. Calcule el punto de Cauchy mediante el Algoritmo CP;
 2. Calcule una dirección de búsqueda d_k mediante el método primal directo, el método del gradiente conjugado o el método dual;
 3. Realice una búsqueda de línea a lo largo de d_k , sujeta a los límites del problema, para calcular una longitud de paso λ_k . La búsqueda de línea comienza con la longitud de paso unitaria, satisface $f(x_{k+1}) \leq f(x_k) + \alpha \lambda_k g_k^T d_k$, con $\alpha = 10^{-4}$, e intenta satisfacer $|g_{k+1}^T d_k| \leq \beta |g_k^T d_k|$ con $\beta = 0,9$;
 4. Calcule $x_{k+1} \leftarrow x_k + \lambda_k d_k$;
 5. Calcule $\nabla f(x_{k+1})$;
 - if** $s_k^T y_k > \text{eps} \|y\|^2$, con $\text{eps} = 2,2 \times 10^{-16}$ **then**
 Agregue s_k y y_k a S_k y Y_k ;
 if se almacenan más de m actualizaciones **then**
 Elimine la columna más antigua de S_k y Y_k ;
 end if
 1. Actualice $S_k^T S_k, Y_k^T Y_k, L_k$ y R_k ;
 2. Calcule $\theta \leftarrow \frac{y_k^T s_k}{y_k^T y_k}$;
 3. Calcule $k \leftarrow k + 1$ y volver al inicio;
- end if**
-

Donde para un n grande, los vectores l y u representan límites inferiores y superiores para cada componente. Supongamos además que el gradiente es conocido y notemos $g = \nabla f$.

El algoritmo no requiere segundas derivadas ni conocimiento de la estructura de la función objetivo y, por tanto, se puede aplicar aún cuando la matriz Hessiana no es práctica de calcular. Se utiliza una actualización cuasi-Newton de memoria limitada para aproximar la matriz Hessiana de tal manera que el almacenamiento requerido sea lineal en n [10].

Por su extensión y detalle, reproducimos únicamente un esbozo general del Algoritmo L-BFGS-B. Se requiere complementariamente del Algoritmo CP, de los métodos primal directo, gradiente conjugado, y dual que se encuentran en la presentación original, donde explícitamente se señala no haber realizado el análisis para el orden de su convergencia [10].

4.3. Experimentación numérica

En esta sección se presentan los resultados obtenidos de calcular la solución regularizada conjunta para la reconstrucción PET/MRI. Es decir, para los términos de ajuste en la Definición 26 que determinan al funcional objetivo en (4.4), por tanto, tomando

$$\mathcal{D}_1(f, A(u)) = \sum_i (Au)_i - f_i \log(Au)_i \quad \text{y} \quad \mathcal{D}_2(g, B(v)) = \frac{1}{2\sigma^2} |Bv - g|^2.$$

Se emplean datos sintéticos PET/MRI (f, g) como se detallaron en la sección anterior, simulados como se ha especificado en las Secciones 2.6.3 y 2.7.3 para PET y MRI respectivamente a partir de sus fantomas y las discretizaciones realizadas.

Con el fin de replicar los experimentos en [15], y comparar los resultados, se han considerado reconstrucciones individuales y conjuntas, con y sin el regularizador. Se ha instalado y empleado el software disponible en [14], lo cual ha supuesto un reto, principalmente por las minimizaciones numéricas empleando el algoritmo L-BFGS-B, cuya implementación a su vez depende de un “encapsulamiento” a MATLAB de la versión en C

del algoritmo original programado en FORTRAN [54]. Finalmente se ha encontrado solución a esta dificultad, en la actualización realizada en [6] e implementando las adaptaciones pertinentes al código en [14].

Se reproducen los 6 tipos de submuestreo ensayados en [15] sobre los datos MRI: el primer tipo es un muestreo completo (`full`), el segundo es un muestreo “Cartesiano” realizado únicamente sobre líneas horizontales pares (`lines2`), el tercero y cuarto submuestreos se realizan sobre líneas concéntricas, definidas para los ángulos resultantes de dividir respectivamente el ángulo plano en 15 (`radial15`) y en 20 (`radial20`) ángulos iguales. Los dos últimos casos son muestreos espirales, uno con densidad uniforme (`spiralUni`) y el otro con densidad mayor en las frecuencias altas (`spiralHigh`). Para poder reproducir las geometrías de estos submuestreos, se ha recurrido a una ingeniería inversa de software para identificar los índices de los elementos coincidentes entre el espacio- k completo y sus versiones submuestreales para los datos originalmente empleados en [14], y de esta forma reproducir la misma geometría sobre cualquier otro dato sintético. Esto ha sido posible gracias a que los valores son complejos y se ha podido verificar la unicidad de cada uno de los datos que componen la discretización del espacio- k con la que computacionalmente se trabaja. La implementación de estos submuestreos se encuentra en `enmascarar.m`.

Los parámetros α y β para los regularizadores JTV, QPLS y LPLS se han mantenido intactos en el software [14]. Para la representación como imagen de los errores en cada reconstrucción, se emplea el error relativo $\|u - u^*\| / \|u^*\|$, donde u^* corresponde al fantoma. Las reconstrucciones JTV, se emplean para inicializar las reconstrucciones QPLS, y a su vez éstas para inicializar LPLS.

Los resultados de la experimentación numérica para la reconstrucción individual y conjunta de imágenes de resonancia magnética e imágenes de tomografía por emisión de positrones a partir del modelo estadístico para estimación de parámetros de máxima verosimilitud, y empleando el algoritmo L-BFGS-B para optimización numérica, se representan gráficamente para los submuestreos anteriormente señalados, en la Figuras de 4.2 a 4.7. Considerando como regularizadores al nulo, a TV, JTV, QPLS y a LPLS.

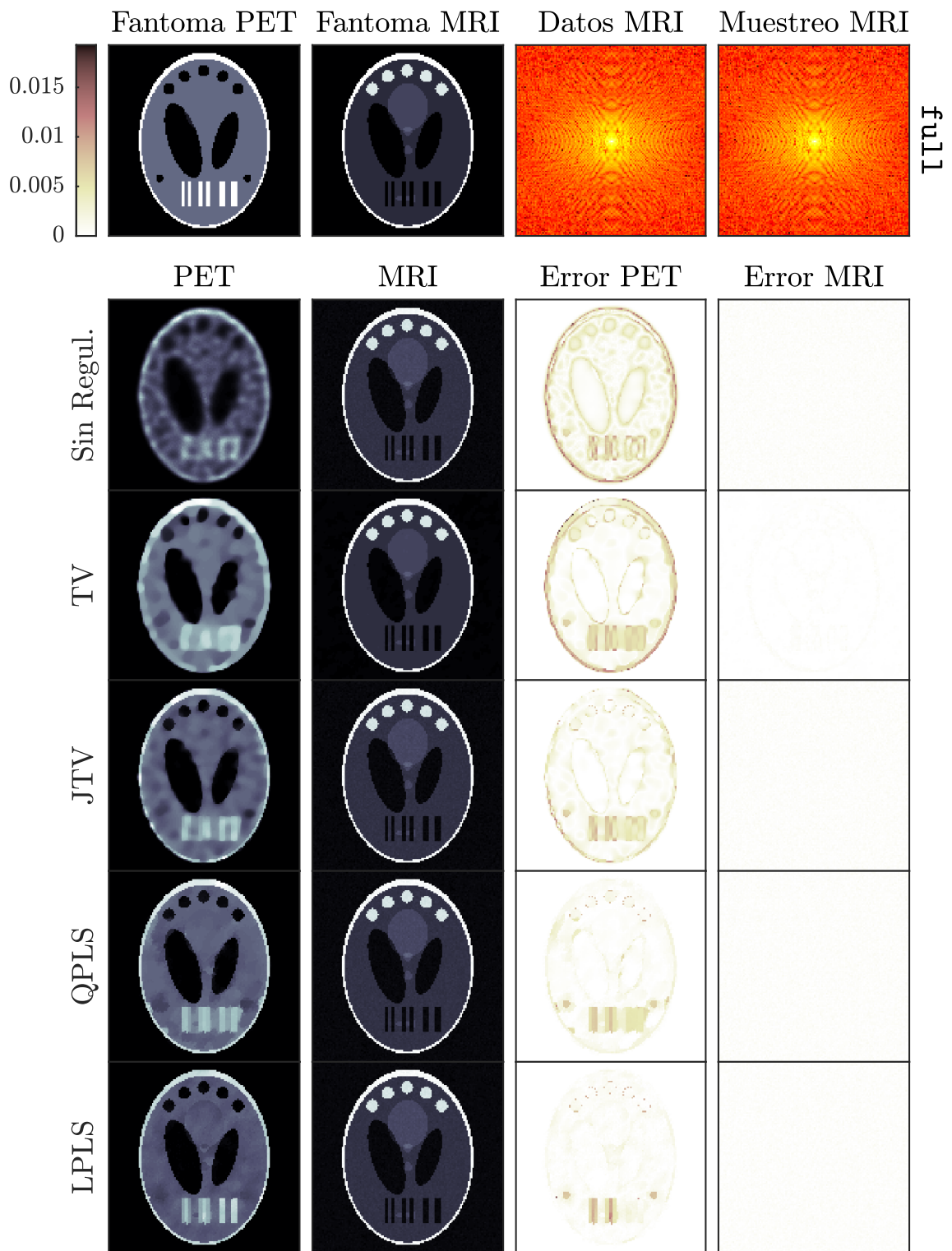


Figura 4.2: Reconstrucción conjunta PET/MRI para el muestreo completo: full.

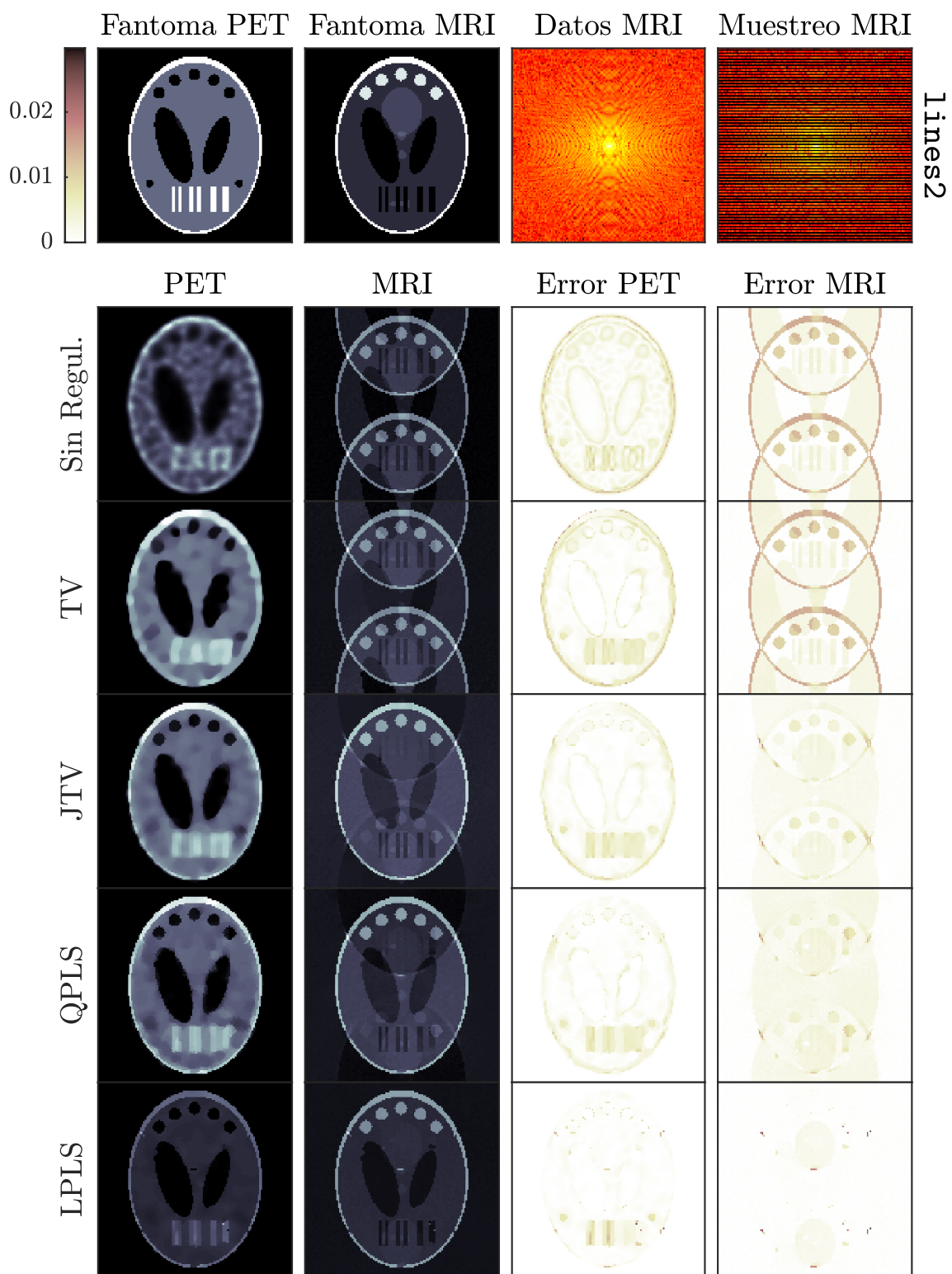


Figura 4.3: Reconstrucción conjunta PET/MRI para el muestreo "Cartesiano": lines2.

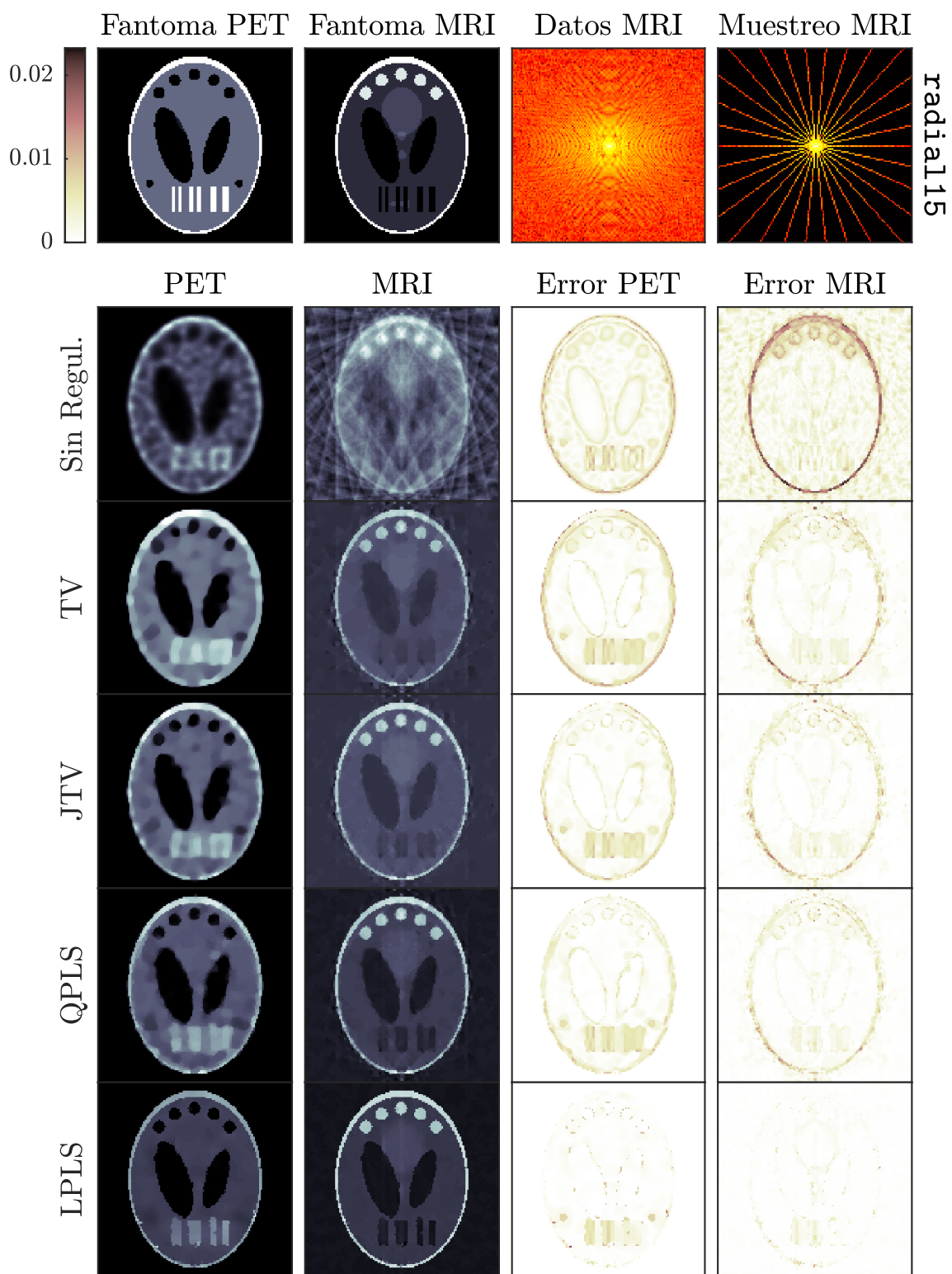


Figura 4.4: Reconstrucción conjunta PET/MRI para el muestreo radial: radial15.

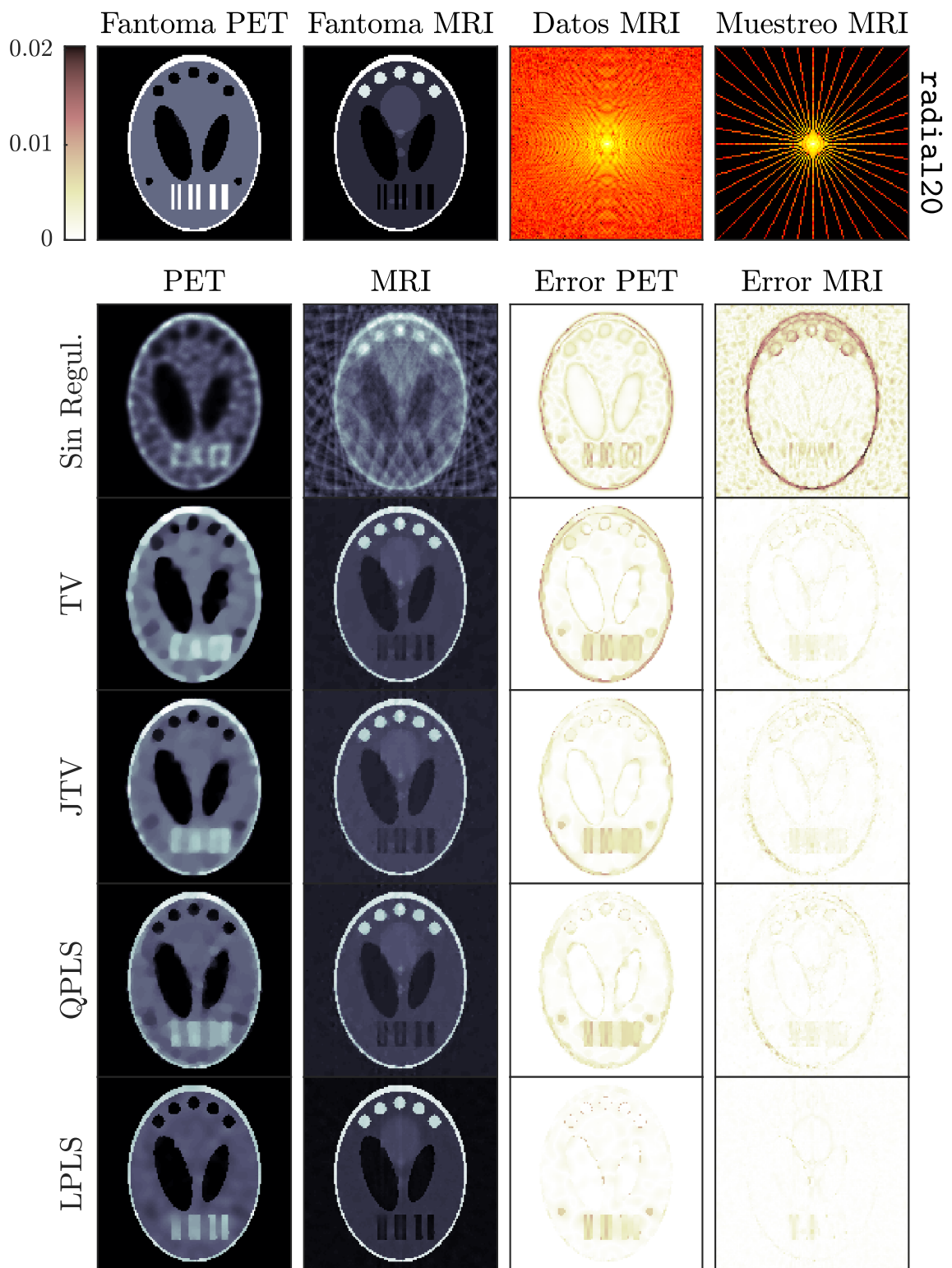


Figura 4.5: Reconstrucción conjunta PET/MRI para el muestreo radial: radial120.

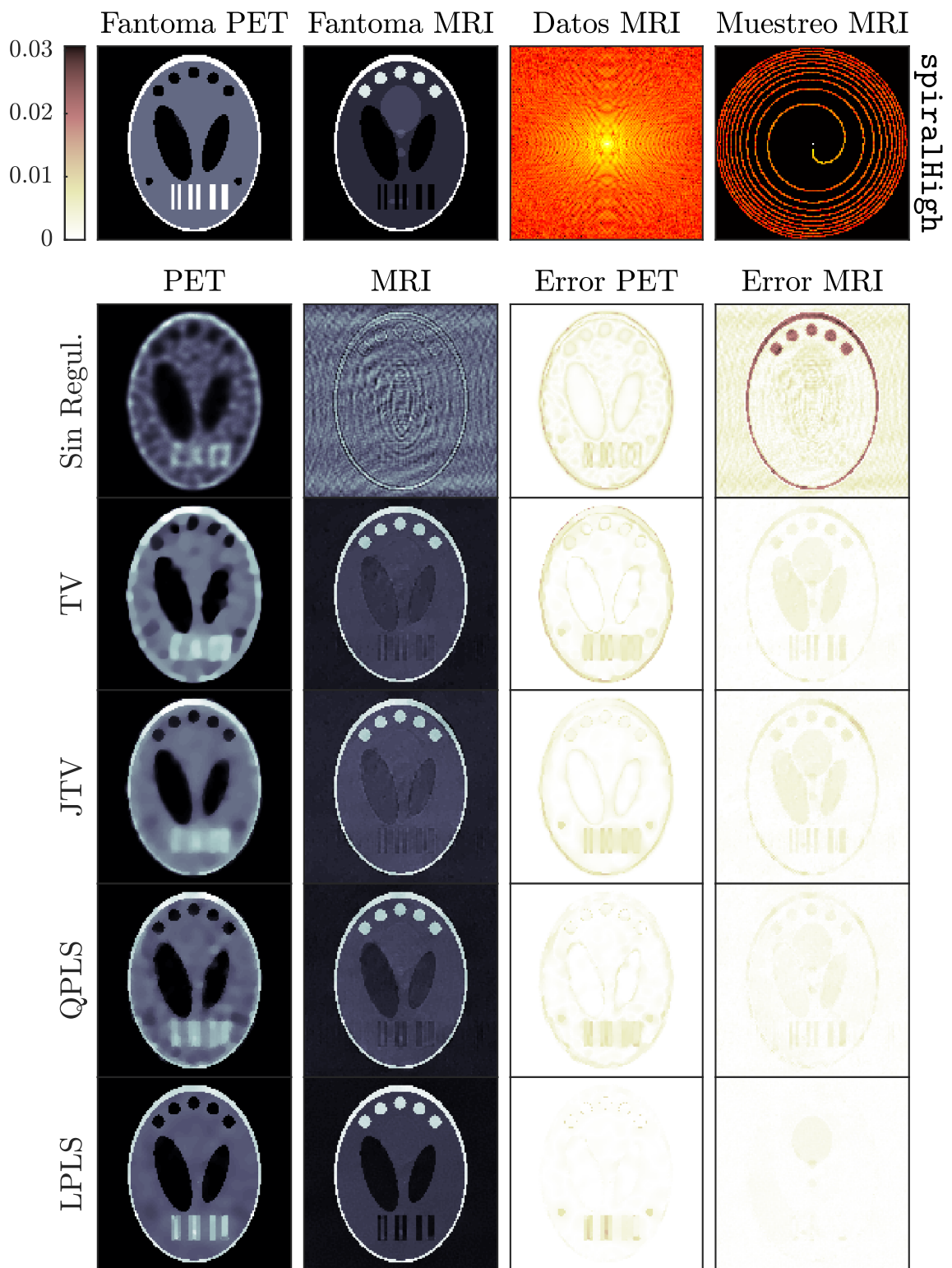


Figura 4.6: Reconstrucción conjunta PET/MRI para el muestreo espiral: spiralHigh.

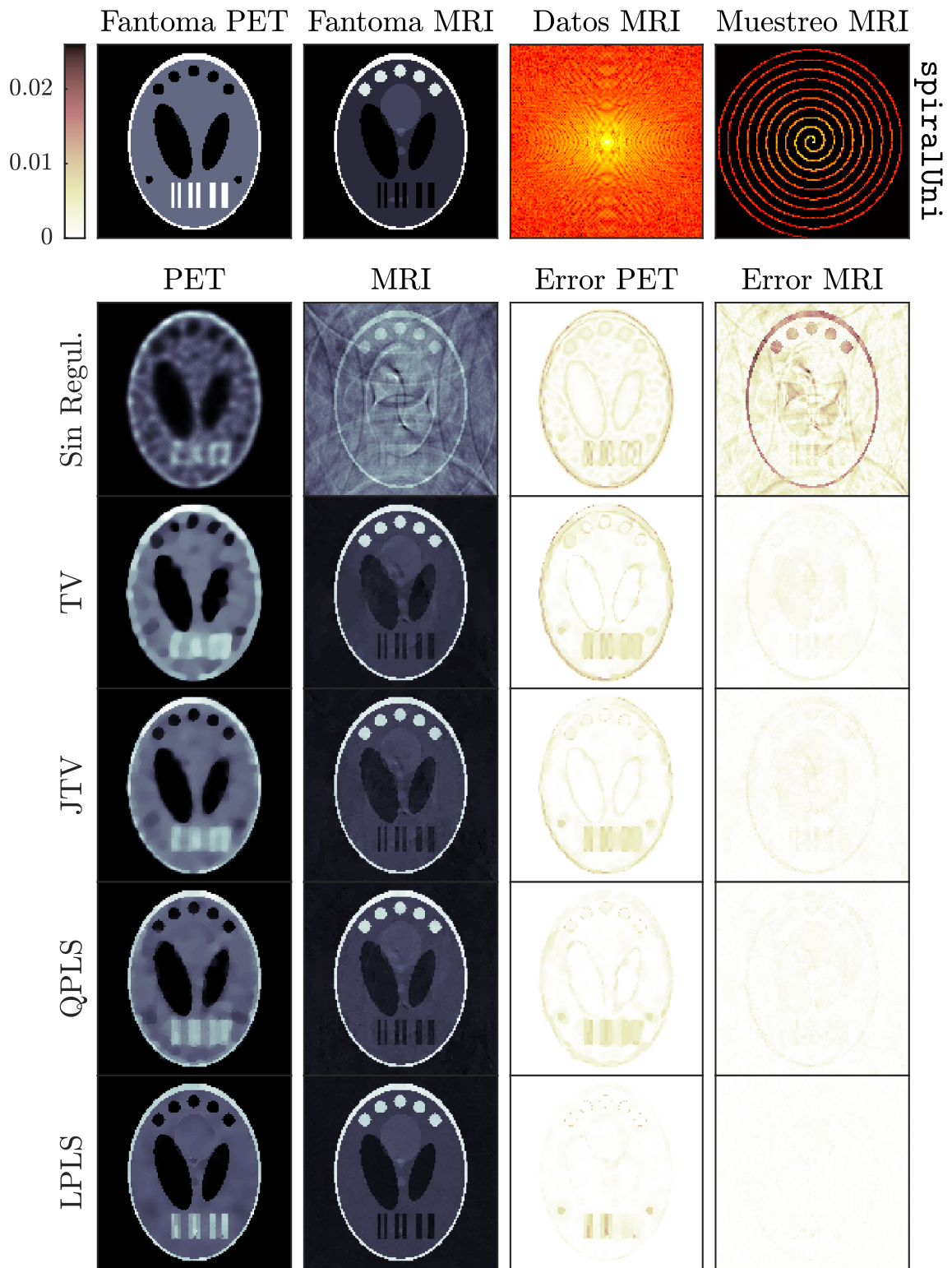


Figura 4.7: Reconstrucción conjunta PET/MRI para el muestreo espiral: spiralUni.

4.4. Evaluación visual

La primera fila en cada una de las Figuras de 4.2 a 4.7, es referencial tanto para las imágenes originales de los fantasmas PET y MRI, como para el espacio- k y la geometría empleada en cada tipo de submuestreo. En la primera columna de cada una de las Figuras se presentan las reconstrucciones PET, mientras en la segunda columna, las reconstrucciones MRI. La tercera y cuarta columna representan gráficamente a los errores relativos con respecto a su correspondiente fantoma. Cada una de las filas a partir de la segunda, corresponde al tipo de regularizador etiquetado: Sin Regularizador, TV, JTV, QPLS y LPLS, respectivamente. Por último cabe señalar que, las filas 2 y 3 son reconstrucciones individuales, mientras las filas 4, 5 y 6 las reconstrucciones conjuntas.

Submuestreo full

En la Figura 4.2 para el submuestreo MRI `full`, los resultados de reconstrucción no presentan la aparición de artefactos. El error para las reconstrucciones individuales y conjuntas MRI son prácticamente inapreciables. Para PET, se observa que el resultado va mejorando progresivamente, según el orden presentado para los regularizadores. Teniendo la mejor preservación de bordes compartidos para el caso LPLS, en esta última también se aprecia como MRI aporta a PET con estructuras únicamente presentes en MRI. El error disminuye en el mismo orden presentado, es decir, de arriba hacia abajo, marcándose especialmente en las estructuras que difieren de forma y tamaño (los círculos superiores blancos en el fantoma MRI son de mayor diámetro que los círculos superiores negros en el fantoma PET), también se aprecia que el error aumenta a medida de que dos estructuras diferentes se juntan, como en las barra paralelas izquierdas.

Submuestreo lines2

Los resultados para el submuestreo `lines2`, se muestran en la Figura 4.3. En todas las reconstrucciones MRI se observa de mayor a menor la superposición de imágenes, lo que se evidencia en el error, el mejor re-

sultado se consigue para el regularizador LPLS con pocas incrustaciones de artefactos. Las estructuras que componen las imágenes reconstruidas PET se van consolidando de abajo hacia arriba. Los artefactos presentes en MRI son los mismos que en PET para el regularizador LPLS.

Submuestreos radial15 y radial20

Los resultados para el submuestreo `radial15` (Figura 4.4) y el submuestreo `radial20` (Figura 4.5) son similares, hay una abundante presencia de artefactos en el caso de la reconstrucciones individual MRI sin regularizador explícito, y se logran seguir apreciando para TV y JTV. Poco o nada apreciables con los regularizadores PLS, de nuevo el menor error se observa en el caso lineal. PET tiene un comportamiento similar que en otros tipos de submuestreo para MRI, las estructuras se definen mejor conforme el orden presentado para los priors considerados, se observa además que estructuras únicamente presentes en PET se van perdiendo en el mismo orden. Por otra parte no se observa mayor incidencia de estructuras pasadas de MRI, como si sucede en `full`.

Submuestreos spiralHigh y spiralUni

En los submuestreo `spiralHigh` (4.6) y `spiralUni` (4.7) hay diferencias importantes. MRI tiene una importante incidencia de artefactos más evidente para `spiralHigh` donde la imágenes lucen opacas, al igual que los demás casos los resultados mejoran de arriba hacia abajo. PET se comporta de forma similar a los demás muestreo probados, con la particularidad de que empleando el prior LPLS en el caso `spiralUni` se aprecia un traspaso de estructuras provenientes de MRI como en `full`.

Capítulo 5

Conclusiones Generales

El presente trabajo se enmarcó, principalmente, en estudiar la Teoría relacionada a los problemas inversos en reconstrucción de imágenes con aplicaciones en Medicina y probada en datos sintéticos. Bajo este contexto, se detallan a continuación los principales ejes desarrollados.

1. La literatura especializa, tanto clásica como reciente no presenta un tratamiento fundamental de la Teoría de Problemas Inversos. Lo que constituye una oportunidad para fundamentar y divulgar esta rama de las matemáticas. En este trabajo se ha propuesto una definición formal para “problema inverso” en el marco de la Teoría de Conjuntos; que debería ser explorada en la perspectiva de la matemática pura. El concepto de “dualidad” que subyace en el de *par directo-inverso* pareciera ser una idea fundamental en matemáticas, puesto se encuentra presente transversalmente en diferentes ramas de esta ciencia. Un tratamiento formal de esta Teoría y desde la perspectiva general del Análisis y del Cálculo Variacional, se encuentra muy bien desarrollada; por ejemplo en la obra de Heinz Werner Engl et al [16], así como en la de Andreas Kirsch [24]. Se destaca el aporte de Jennifer Müller y Samuli Siltanen [39] por la claridad y sencillez de su presentación, se recomienda la lectura de su libro “Linear and nonlinear inverse problems with practical applications” como introducción amigable a los Problemas Inversos, pudiendo considerarse como texto guía para una asignatura en este tópico.

2. Una diversidad de problemas en física pueden ser interpretados y luego resueltos matemáticamente como problemas inversos, por lo que la Teoría de Problemas Inversos es relevante en el ámbito de la matemática aplicada. Destacan especialmente los *problemas de medida indirecta*, que aluden a la posibilidad de medir una cantidad física en virtud de otra distinta y de la que, de otra manera, no se podría obtener información. Estos son interpretados como *problemas lineales inversos*, cuyo operador asociado goza de una teoría muy rica y desarrollada. En la matemática básica superior, el primer contacto explícito con uno de estos operadores, sucede en Cálculo, cuando se estudia la *antiderivada* como un operador que devuelve la expresión de una función para la que se conoce su derivada, y cuyo inverso es la *diferenciación*. En Álgebra Lineal, específicamente en sistemas de ecuaciones lineales, el operador lo constituye la *matriz asociada al sistema*; que tiene solución única justamente cuando existe la *matriz inversa*, en general la existencia, unicidad o multiplicidad de la solución dependerá del rango la matriz. Más adelante en Ecuaciones Diferenciales y Análisis de Fourier, aparecen operadores integrales como la *Convolución*, la *Transforma de Laplace* y la *Transformada de Fourier*. Otro ejemplo interesante, y está vez en Matemática Numérica, es la *Descomposición en Valores Singulares* (SVD), que permite calcular *pseudo-inversas* para cualquier matriz, incluso si son singulares o si ni siquiera son cuadradas. En Estadística, ese operador es la *Esperanza Matemática* de una variable aleatoria, mientras la inversión requiere de estimación paramétrica. Cabe recalcar la importancia que cobran los denominados *problemas mal puestos* o sus versiones discretas llamadas *problemas mal condicionados*, que esencialmente permiten clasificar en dos a los problemas inversos: aquellos que dependen continuamente de los *datos*, cuya solución se calcula directamente empleando una “simple inversión” del operador y aquellos para los que una pequeña perturbación en los datos vuelve inútil el uso de una “inversión ingenua”. Para estos segundos, es que el desarrollo de la Teoría ha permitido obtener soluciones aproximadas en las que intervienen *inversiones regularizadas* sobre el operador. Uno de estos casos, es el de la reconstrucción de imágenes médicas.

3. El estudio de la convolución y *deconvolución* como un ejemplo sencillo de par directo-inverso en el marco de la Teoría de Problemas Inversos, permite comprender primeramente la necesidad de una discretización del operador integral que define a la convolución y en segunda instancia la necesidad de una inversión regularizada que permita “recuperar” una señal a partir de una muestra perturbada de datos. Se ensayó la solución sin regularizador, y las soluciones regularizadas: para la *Descomposición en Valores Singulares Truncada* (TSVD), de *Tikhonov* y de *Variación Total*. Se implementó computacionalmente estos resultados con el objetivo de aplicarlos en la “corrección” del *desenfoque* a una imagen con ruido blanco.
4. El modelo determinista para la atenuación de intensidad de un haz de Rayos X, es el punto de partida para la reconstrucción de imágenes médicas. Su interpretación como un problema inverso lineal y la definición de *Transformada de Radon* han motivado un importante desarrollo de la Teoría, que ha hecho posible gracias a la fórmula de inversión de *Transformada de Fourier* y al *Teorema del corte central*, encontrar la solución conocida como *Retroproyección filtrada*. La discretización por cuadratura de esta transformación integral y su implementación computacional correspondiente, permiten probar que se trata de un *problema mal condicionado*, y sobre todo hace posible su aplicación médica en *Tomografía Computarizada* de Rayos X (CT). En MATLAB, como en otros lenguajes, se dispone de las funciones `radon` e `iradon` implementando computacionalmente al operador CT y su inversa.
5. La *Tomografía por Emisión de Positrones* (PET) se interpreta indeterminísticamente como un modelo probabilístico de datos correspondientes a la realización de un vector aleatorio con una distribución Poisson y cuya media se estima como una *Transformada de Rayos X* sobre el vector aleatorio correspondiente a la “imagen” PET; que cuenta únicamente en las *líneas de respuesta* (LOR) definidas entre cada par de detectores que conforman el anillo de sensores PET, cada par de fotones producidos por los fenómenos de aniquilación subatómica de electrones y positrones como reacción a un “contraste” radiactivo inyectado en el organismo. Una *reconstrucción* de

imágenes PET, es posible en su planteamiento como problema inverso lineal, empleando una discretización por cuadratura similar a la de CT; que permite verificar el mal condicionamiento de la matriz resultante. La aproximación buscada, viene dada por una estimación paramétrica de *Máxima Verosimilitud a Posteriori* (MAP) a partir de los datos medidos por el escáner correspondiente, y el empleo de *regularizadores* a priori que “preserven bordes”; en este trabajo se ensaya el empleo de la *Variación Total* (TV) y como referencia, se incluye el caso sin regularizador.

6. *Imagen por Resonancia Magnética* (MRI), es una técnica de imagen médica que aprovecha el fenómeno de magnetización, en específico, del spin de los núcleos de los átomos de hidrógeno (^1H) presentes en los tejidos orgánicos. La señal captada o medida por las bobinas que componen los sensores de los escáneres MRI, se corresponde matemáticamente con la transformada de Fourier en 3 dimensiones de la “imagen”; deducida de un modelamiento dinámico en ecuaciones diferenciales ordinarias resultante de las Leyes de Maxwell y de la Ley de inducción de Faraday. En la literatura correspondiente a matemática superior básica se puede encontrar referencias a la *Transformada Discreta de Fourier* (DFT) y a la *Transformada Rápida de Fourier* (FFT). Para MRI se considera un ruido blanco Gaussiano de origen instrumental, propio de los fenómenos electromagnéticos y que configuran un problema de medida indirecta de “imágenes”. Así una reconstrucción es posible gracias la inversión regularizada de la DFT o de la FFT; en este trabajo únicamente se ensaya a TV como regularizador para la modalidad de reconstrucción individual.
7. Empleando imágenes de prueba, generadas numéricamente con formas geométricas compuestas llamadas *fantomas*, una para PET y otra para MRI; que comparten estructura pero también incluyen diferencias entre ellas. Se sintetizaron y almacenaron los datos correspondientes a cada una de estas modalidades de imagen médica. Para ello, se implementó computacionalmente la matriz asociada al operador lineal PET, la cual resulta de calcular para una base canónica de las matrices $m \times n$, la versión discreta de la transformada de Rayos X que, en dos dimensiones, equivale a la transformada de

Radon, aprovechando su implementación en `MATLAB` como la función homónima `radon`. Además se aplicó un desenfoque con un filtro convolutivo aprovechando la función `imfilter` con una *función de dispersión de punto* (PSF) discreta y fija. Finalmente se aplicó un factor de escalamiento y por último, cada uno de estos valores se tomaron como el parámetro para simular el vector aleatorio Poisson, empleando `poissrnd`, y produciendo así el *sinograma* PET buscado. El caso MRI, resultó bastante más simple, pues se dispone en `MATLAB` de las funciones `fft2`, `fftshift` y `randn` que permitieron calcular con facilidad, por ejemplo, partiendo del fantoma MRI, sus datos correspondientes al *espacio- k* y se agregó un ruido Gaussiano.

8. Suponiendo que los datos PET y MRI provenientes de una misma fuente son condicionalmente independientes, se empleó el modelo probabilístico de reconstrucción conjunta PET/MRI, directamente del Teorema de Bayes y siguiendo la propuesta en [15]. Se estimaron las imágenes PET de sus datos Poisson y, las imágenes MRI de sus datos Gaussianos. Además, se consideró un regularizador conjunto conocido a priori; que permitió aprovechar la similitud estructural entre ambas modalidades de imagen médica. Se construyó un estimador de *máxima verosimilitud conjunta a posteriori* (JMAP), y se lo convirtió en un funcional objetivo a minimizar numéricamente, para lo cual se empleó el Algoritmo L-BFGS-B, con el cual se calcularon las reconstrucciones PET y MRI. Se ensayó el empleo de *Variación Total Conjunta* (JTV), *Conjuntos de Nivel Paralelo*, tanto *Cuadráticos* (QPLS) y *Lineales* (LPLS), como regularizadores conjuntos, y se ensayó también con submuestreos MRI considerando la totalidad de los datos y diferentes geometrías como: líneas horizontales paralelas, lineales radiales para diferente número de ángulos, y espirales con diferentes densidades de acumulación.
9. Se instaló y empleó el software existente para reconstrucción conjunta PET/MRI desarrollado en `MATLAB` [14]. Se superó la dificultad generada por una dependencia desactualizada del interfaz `FORTRAN-C-MATLAB` para la implementación original del Algoritmo L-BFGS-B [54][5], distribuida con el nombre `L-BFGS-B-C` [6]. Se realizaron actualizaciones sobre todos los métodos programados que hacen lla-

mado a la función `lbfgsb` con el fin de adaptar el software a la actualización mencionada. Se implementó el método `enmascarar` con el fin de recuperar la geometría para los submuestreo MRI de los datos originales en [14] y presentados en [15]. Se verificó que cada uno de los datos de variable compleja que conforman el espacio- k son únicos para la muestra total y por comparación con los demás submuestreos se determinó el índice correspondiente a cada pixel de las geometrías empleadas. Las ‘‘imágenes’’ iniciales tanto PET como MRI, necesarias para inicializar la reconstrucción conjunta con regularizador QPLS, se obtuvieron de las *soluciones regularizadas conjuntas JTV*, que a su vez se emplearon para inicializar la reconstrucciones conjuntas para la regularizaciones LPLS de cada tipo de submuestreo.

10. La evaluación visual de los resultados gráficos obtenidos muestran en general, superioridad en los resultados para las reconstrucciones individuales regularizadas sobre las no regularizadas explícitamente, tanto para PET como para MRI. A su vez, la reconstrucción conjunta regularizada evidencia mejoras en la totalidad de los resultados respecto de las reconstrucciones individuales. Es claro que ambas modalidades de imagen médica se benefician mutuamente de la reconstrucción conjunta regularizada, el caso de PET muestra sus mejores resultados en definición de bordes para LPLS, seguida de QPLS y finalmente JTV en el mismo orden para todos los submuestreos de los datos MRI. Por otro lado MRI, también saca provecho de PET, en específico al corregir la pérdida de datos producida por los submuestreos, una vez más se observa el menor error para el regularizador conjunto LPLS, seguido de QPLS y finalmente de JTV, en estos últimos dos se puede llegar apreciar la presencia de artefactos. Con el uso de LPLS en PET, se observa la presencia de estructuras ajenas que son traspasadas desde MRI. Cabe mencionar que la propuesta desarrollada en [15] y, que ha servido de guía para el presente trabajo, fue uno de los primeros acercamientos al estudio de la reconstrucción conjunta para la técnica PET/MRI. Otros trabajos en esta dirección pueden encontrarse en [27], [36], [46] y [53].

11. El presente trabajo, de Matemática aplicada a la reconstrucción conjunta de imágenes médicas, requirió de una importante interdisciplinariedad entre varias ramas de las matemáticas, así como de conocimientos, habilidades y experiencia en programación. Por ejemplo, se empleó *Geometría Plana y Analítica* en la deducción de la ecuación de la recta que modela la trayectoria de un haz de Rayos X, de *Fundamentos de la Matemática* se empleó el concepto de *grafo* para fundamentar la definición de problema inverso y en general la notación conjuntista, del *Cálculo en una variable* se obtiene el modelo básico para CT, el *Álgebra Lineal* es protagonista tanto como notación, teoría y discretización para los problemas inversos lineales, para los vectores de datos, o las matrices para los métodos directos. Del *Cálculo Vectorial* se sigue las integrales de línea para las transformadas de Radon y Rayos X, el gradiente para la variación total y también la integración en coordenadas polares empleada en el teorema principal de la *retroproyección filtrada* para la inversión de la transformada de Radon, la *Probabilidad y Estadística Básica* permiten entender el modelo para PET como un fenómeno Poisson pero también en el ruido Gaussiano de los datos MRI, un importante conocimiento en *Física* facilita la comprensión tanto en la dificultad de realizar mediciones, como en los modelos para CT que es una simple proporción directa entre la atenuación de la intensidad de Rayos X y la distancia que recorren, PET involucra fenómenos subatómicos cuantizables y MRI electromagnéticos. *Variable Compleja* se emplea tanto para la transformada de Fourier en las inversiones, como en el modelo MRI, que se describe como un sistema en *Ecuaciones en Derivadas Ordinarias* de donde resulta una solución para el problema de valor inicial. El *Análisis* en general también es protagonista en este trabajo, pues se han empleado: su técnica, definiciones como la de métrica o distancia, los espacios funcionales, la teoría de operadores lineales, las transformaciones integrales, los conceptos de ortogonalidad y paralelismo empleadas en PLS. La *Matemática Numérica* aporta la matriz de diferencias finitas empleada en la solución regularizada de variación total, pero también en la discretización de la convolución, y en los métodos directos como en SVD, TSVD, Tikhonov empleados en la corrección del desenfoque.

Análisis de Fourier aporta en nociones de ortogonalidad para espacios funcionales que luego permite comprender la transformada de Fourier. Pero también un conocimiento en matemática pura como en *Estructuras Algebraicas*, *Geometría Diferencial* y *Topología* permitieron identificar resultados y conceptos que surgieron tales como: el isomorfismos entre las representaciones de imagen digital, el automorfismo de los espacios de Schwartz, la aparición del fibrado tangente y la teselación de una imagen por sus píxeles. El tercer rol es coprotagonizado por la *Estadística Avanzada* que provee el método de estimación de parámetros por máxima verosimilitud conjunta que hace posible en el marco de la *Teoría de Probabilidades* el modelo de la reconstrucción conjunta de imágenes, el *Cálculo Variacional* proveyó la teoría y la técnica de acotación para probar la existencia de soluciones óptimas como se mostró en el caso de la corrección del desenfoque, y finalmente la *Optimización Numérica* que proveyó los métodos y algoritmos numéricos para aproximar las soluciones por inversión regularizada que permiten reconstruir las imágenes médicas.

Capítulo A

Códigos MATLAB

A.1. tic

tic/funciones

calculo_eta.m

```
1 function eta = calculo_eta(delta_x,a)
2     eta = 1;
3     while ~( (eta+1)*delta_x > a )
4         eta = eta + 1;
5     end
6 end
```

convolucion_discreta.m

```
1 function estructura = convolucion_discreta(f,phi_0,n,varargin)
2     a = 0.04;
3     for i = 1:2:length(varargin)
4         name = varargin{i};
5         value = varargin{i+1};
6         switch name
7             case 'a'
8                 a = value;
9             otherwise
10                error(['Argumento no válido: ', name])
11        end
```

```

12     end
13
14     delta_x = 1/n;
15     eta = calculo_eta(delta_x, a);
16     if eta == 1
17         warning('Para el parámetro a = %g, preferiblemente ingresar un n > %d.\n', a, 1/a)
18     end
19     l = -eta:1:eta;
20     p_tilde = phi_0(1*delta_x)';
21     p = (delta_x*sum(p_tilde))^( -1)*p_tilde;
22     P_extendida = zeros(n+2*eta,n);
23     for k = 1 : n
24         P_extendida(k:k+length(l)-1,k) = p;
25     end
26     % prueba_identidad_convmtx = all(all(P_extendida == convmtx(p,n)));
27     P = P_extendida(1+eta:end-eta, :);
28     % P(1:eta,:) = P(1:eta,:) + P_extendida(end-(eta-1):end,:);
29     % P(end-(eta-1):end,:) = P(end-(eta-1):end,:) + P_extendida(1:eta,:);
30     A = delta_x*P;
31
32     xn = x_discreto(0,n)';
33     fn = f(xn);
34     An = A;
35     mn = An*fn;
36     maximo = max(mn);
37
38     estructura = struct('p',p,'x',xn,'f',fn,'A',An,'m',mn,'max',maximo);
39 end

```

diag_truncada.m

```

1 function D_alpha = diag_truncada(D_n,varargin)
2     alpha = 0.8;
3     r_alpha = ceil(0.4*max(size(D_n)));
4     d_n = diag(D_n);
5
6     for i = 1:2:length(varargin)
7         name = varargin{i};
8         value = varargin{i+1};
9         switch name
10             case 'alpha'
11                 alpha = value;
12                 %r_alpha = length(d_n(d_n > alpha));
13                 [~,r_alpha] = min(d_n(d_n > alpha));
14             case 'r_alpha'
15                 r_alpha = value;
16             otherwise
17                 error(['Argumento no válido: ', name])
18         end
19     end

```

```

20
21     n = size(D_n,1);
22     D_alpha = zeros(n,n);
23     D_alpha(1:r_alpha,1:r_alpha) = diag(1./d_n(1:r_alpha));
24 end

```

entrada.m

```

1 function [f,a,phi_0,phi,convolucion] = entrada()
2     %% Pulso
3     f = @(x) (x>=0.00 & x<0.1).*0 + ...
4             (x>=0.10 & x<0.30).*1 + ...
5             (x>=0.30 & x<0.4).*0 + ...
6             (x>=0.40 & x<0.7).* (1.4-2*x) + ...
7             (x>=0.70 & x<0.75).*0 + ...
8             (x>=0.75 & x<0.8).*0.7 + ...
9             (x>=0.80 & x<0.85).*0 + ...
10            (x>=0.85 & x<0.9).*0.8 + ...
11            (x>=0.90 & x<1.0).*0;
12     %% PSF
13     a = 0.04;
14     syms x
15     C_a = (int((x+a)^2*(x-a)^2, x, -a, a))^(-1);
16     C_a = double(C_a);
17     phi_0 = @(x) (x>=-a & x<=a).*C_a.*(x+a).^2.*(x-a).^2;
18     phi = @(x) phi_0(mod(x,1));
19     %% Convolución
20     convolucion = @(phi,f,x,a) arrayfun(@(x) integral(@(y) phi(y).*f(x-y),-a,a), x);
21 end

```

error_relativo.m

```

1 function error = error_relativo(x,y)
2     error = 100*norm(x - y)/norm(x);
3 end

```

forma_apilada.m

```

1 function pseudoinversa = forma_apilada(A,m,alpha)
2     N = size(A,2);
3     A_tilde = [A; sqrt(alpha) * eye(N,N)];
4     m_tilde = [m; zeros(N,1)];
5     pseudoinversa = A_tilde \ m_tilde;
6 end

```

incrementar_dimensiones.m

```
1 function incrementar_dimensiones(porcentaje_incremento)
2     posicion = get(gca, 'Position');
3     posicion(3:4) = ((100 + porcentaje_incremento) / 100) * posicion(3:4);
4     set(gca, 'Position', posicion)
5 end
```

matriz_convolucion.m

```
1 function A = matriz_convolucion(psf,n)
2     p = psf;
3     eta = floor(numel(p)/2);
4     l = -eta:1:eta;
5     delta_x = 1/n;
6     P_extendida = zeros(n+2*eta,n);
7     for k = 1 : n
8         P_extendida(k:k+length(l)-1,k) = p;
9     end
10    % prueba_identidad_convmtx = all(all(P_extendida == convmtx(p,n)));
11    P = P_extendida(1+eta:end-eta, :);
12    P(1:eta,:) = P(1:eta,:) + P_extendida(end-(eta-1):end,:);
13    P(end-(eta-1):end,:) = P(end-(eta-1):end,:) + P_extendida(1:eta,:);
14    A = delta_x*P;
15 end
```

pseudoinversa_svd.m

```
1 function pseudoinversa = pseudoinversa_svd(A)
2     [U,D,V] = svd(A);
3     pseudoinversa = V * D^(-1) * U';
4 end
```

pseudoinversa_tikhonov.m

```
1 function pseudoinversa = pseudoinversa_tikhonov(A,alpha)
2     [V,D_alpha,U] = tikhonov(A,alpha);
3     pseudoinversa = V * D_alpha * U';
4 end
```

pseudoinversa_tsvd.m

```
1 function pseudoinversa = pseudoinversa_tsvd(A,alpha)
2     [V,D_alpha,U] = tsvd(A,alpha);
3     pseudoinversa = V * D_alpha * U';
4 end
```

psf_discreta.m

```
1 function p = psf_discreta(phi_0,n,varargin)
2     a = 0.04;
3     for i = 1:2:length(varargin)
4         name = varargin{i};
5         value = varargin{i+1};
6         switch name
7             case 'a'
8                 a = value;
9             otherwise
10                error(['Argumento no válido: ', name])
11        end
12    end
13
14    delta_x = 1/n;
15    eta = determinar_eta(delta_x, a);
16    l = -eta:1:eta;
17    p_tilde = phi_0(1*delta_x)';
18    p = (delta_x*sum(p_tilde))^( -1)*p_tilde;
19 end
20
21
22 function eta = determinar_eta(delta_x,a)
23     eta = 1;
24     while ~( (eta+1)*delta_x > a )
25         eta = eta + 1;
26     end
27 end
```

quitar_marcas_etiquetas.m

```
1 function quitar_marcas_etiquetas()
2     set(gca, 'XTick', [], 'YTick', []) % Quita las marcas de los ejes
3     xlabel('') % Quita la etiqueta del eje x
4     ylabel('') % Quita la etiqueta del eje y
5 end
```

ruido.m

```
1 function x_ruido = ruido(x,porcentaje_ruido)
2     dimension = size(x);
3     maximo = max(max(x));
4     epsilon = (porcentaje_ruido / 100) * maximo * randn(dimension);
5     x_ruido = x + epsilon;
6 end
```

tikhonov.m

```
1 function [V,D_alpha,U] = tikhonov(A,alpha)
2     [U,D,V] = svd(A);
3     D_alpha = D./(D.^2 + alpha);
4 end
```

tsvd.m

```
1 function [V,D_alpha,U,r_alpha] = tsvd(A,alpha)
2     [U,D,V] = svd(A);
3     r_alpha = length(D(D > alpha));
4     D_alpha = 1./D;
5     D_alpha(D <= alpha) = 0;
6 end
```

x_discreto.m

```
1 function xn = x_discreto(b,n)
2     j = 1 : n;
3     xn = b + (j-1)/(n-1);
4 end
```

tic/figuras

grafico_deblurring_tsvd_tikhonov_tv.m

```

1  %% Deblurring TSVD, Tikhonov, TV
2
3  [f,a,phi_0,phi,convolucion] = entrada();
4
5  imagen = imread('cameraman.tif');
6  %[porcentaje_ruido,alpha_tsvd,alpha_tikhonov,alpha_tv] = deal(1,0.05,0.001,32);
7  [porcentaje_ruido,alpha_tsvd,alpha_tikhonov,alpha_tv] = deal(5,0.2,0.05,6);
8
9  n = size(imagen,1);
10 desenfoque = convolucion_discreta(f,phi_0,n);
11 A_n = desenfoque.A;
12 p_n = desenfoque.p;
13 K_phi = p_n/sum(p_n);
14 f_n = double(imagen);
15 m_n = A_n * f_n;
16 psf = uint8(255*A_n/max(max(A_n)));
17 imagen_desenfocada = uint8(m_n);
18 reconstruccion_ingenua = uint8(A_n \ m_n);
19 m_n_ruido = ruido(A_n * f_n, porcentaje_ruido);
20 m_n_ruido = 255 * m_n_ruido / max(max(m_n_ruido));
21 imagen_desenfocada_ruido = uint8(m_n_ruido);
22 reconstruccion_ingenua_ruido = uint8(A_n \ m_n_ruido);
23 reconstruccion_tsvd = uint8(pseudoinversa_tsvd(A_n,alpha_tsvd)*m_n_ruido);
24 reconstruccion_tikhonov = uint8(pseudoinversa_tikhonov(A_n,alpha_tikhonov)*m_n_ruido);
25 reconstruccion_tv = uint8(tvdeconv(m_n_ruido,alpha_tv,K_phi,[],[],[],[]));
26 error = struct;
27 error.tsvd = error_relativo(f_n,double(reconstruccion_tsvd));
28 error.tikhonov = error_relativo(f_n,double(reconstruccion_tikhonov));
29 error.tv = error_relativo(f_n,double(reconstruccion_tv));
30
31 %figure
32 fig = figure;
33 set(fig, 'Position', [000 000 600 600])
34
35 tamaño_eje = [1/3 1/3];
36
37 subplot('Position', [0 2*tamaño_eje(2) tamaño_eje])
38 imshow(imagen)
39 titulo('Original')
40
41 subplot('Position', [tamaño_eje(1) 2*tamaño_eje(2) tamaño_eje])
42 imshow(psf)
43 titulo('PSF')
44
45 subplot('Position', [2*tamaño_eje(1) 2*tamaño_eje(2) tamaño_eje])
46 imshow(imagen_desenfocada)
47 titulo('Desenfoque sin Ruido')
48
49 subplot('Position', [0 tamaño_eje(2) tamaño_eje])
50 imshow(reconstruccion_ingenua)
51 titulo('Reconstrucci\''on Ingenua sin Ruido')
52
53 subplot('Position', [tamaño_eje(1) tamaño_eje(2) tamaño_eje])
54 imshow(imagen_desenfocada_ruido)

```

```

55 titulo('Desenfoque con Ruido')
56
57 subplot('Position', [2*tamano_eje(1) tamaño_eje(2) tamaño_eje])
58 imshow(reconstruccion_ingenua_ruido)
59 titulo('Reconstrucci\''on Ingenua con Ruido')
60
61 subplot('Position', [0 0 tamaño_eje])
62 imshow(reconstruccion_tsvd)
63 titulo('Reconstrucci\''on TSVD')
64
65 subplot('Position', [tamaño_eje(1) 0 tamaño_eje])
66 imshow(reconstruccion_tikhonov)
67 titulo('Reconstrucci\''on Tikhonov')
68
69 subplot('Position', [2*tamaño_eje(1) 0 tamaño_eje])
70 imshow(reconstruccion_tv)
71 titulo('Reconstrucci\''on TV')
72
73 print('-dsvg', 'tic\imagenes\grafico_deblurring_tsvd_tikhonov_tv.svg')
74
75 function textitulo = titulo(texto)
76     textitulo = text(0.5, 0.94, ['\textbf{' texto '}'], ...
77         'Units', 'normalized', ...
78         'Interpreter', 'latex', ...
79         'BackgroundColor', [1 1 1], ...
80         'Margin', 1, ...
81         'HorizontalAlignment', 'center', ... 'Color', 'white', ...
82         'FontSize', 8.4);
83     quitar_marcas_etiquetas()
84 end

```

A.2. JR

JR/matlab/otros

JR_PETMRI_JTV_script_semilla.m

```

1  %% JR_PETMRI_JTV_wrapper('full', 1.39, 0.00072)
2
3  clear
4
5  [sampling, alpha, beta] = deal('full', 1.39, 0.00072);
6
7  run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
8
9  param = [alpha, beta];
10 solution = JR_PETMRI_JTV(options{index.misc}.size, ...

```



```

11     data{index.pet},operator{index.pet},adjoint{index.pet},1e-3,...
12     data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
13     options{index.mri}.sigma_abs, param);
14
15     initial_images = solution';
16
17     save(fullfile(['../data/JR_initial_QPLS_' sampling '.mat']), ...
18         'initial_images');
19
20     %% JR_PETMRI_JTV_wrapper('lines2', 2.78, 0.000725)
21
22     clear
23
24     [sampling, alpha, beta] = deal('lines2', 2.78, 0.000725);
25
26     run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
27
28     param = [alpha, beta];
29     solution = JR_PETMRI_JTV(options{index.misc}.size, ...
30         data{index.pet},operator{index.pet},adjoint{index.pet},1e-3,...
31         data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
32         options{index.mri}.sigma_abs, param);
33
34     initial_images = solution';
35
36     save(fullfile(['../data/JR_initial_QPLS_' sampling '.mat']), ...
37         'initial_images');
38
39     %% JR_PETMRI_JTV_wrapper('radial20', 2.44, 0.000215)
40
41     clear
42
43     [sampling, alpha, beta] = deal('radial20', 2.44, 0.000215);
44
45     run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
46
47     param = [alpha, beta];
48     solution = JR_PETMRI_JTV(options{index.misc}.size, ...
49         data{index.pet},operator{index.pet},adjoint{index.pet},1e-3,...
50         data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
51         options{index.mri}.sigma_abs, param);
52
53     initial_images = solution';
54
55     save(fullfile(['../data/JR_initial_QPLS_' sampling '.mat']), ...
56         'initial_images');
57
58     %% JR_PETMRI_JTV_wrapper('radial15', 1.34, 0.0001)
59
60     clear
61
62     [sampling, alpha, beta] = deal('radial15', 1.34, 0.0001);
63
64     run(fullfile('..', 'data', ['JR_data_' sampling '.m']));

```

```

65
66 param = [alpha, beta];
67 solution = JR_PETMRI_JTV(options{index.misc}.size, ...
68     data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
69     data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
70     options{index.mri}.sigma_abs, param);
71
72 initial_images = solution';
73
74 save(fullfile(['../data/JR_initial_QPLS_' sampling '.mat']), ...
75     'initial_images');
76
77 %% JR_PETMRI_JTV_wrapper('spiralHigh', 5.18, 0.00193)
78
79 clear
80
81 [sampling, alpha, beta] = deal('spiralHigh', 5.18, 0.00193);
82
83 run(fullfile(['..', 'data', ['JR_data_' sampling '.m']]));
84
85 param = [alpha, beta];
86 solution = JR_PETMRI_JTV(options{index.misc}.size, ...
87     data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
88     data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
89     options{index.mri}.sigma_abs, param);
90
91 initial_images = solution';
92
93 save(fullfile(['../data/JR_initial_QPLS_' sampling '.mat']), ...
94     'initial_images');
95
96 %% JR_PETMRI_JTV_wrapper('spiralUni', 2.68, 0.00193)
97
98 clear
99
100 [sampling, alpha, beta] = deal('spiralUni', 2.68, 0.00193);
101
102 run(fullfile(['..', 'data', ['JR_data_' sampling '.m']]));
103
104 param = [alpha, beta];
105 solution = JR_PETMRI_JTV(options{index.misc}.size, ...
106     data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
107     data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
108     options{index.mri}.sigma_abs, param);
109
110 initial_images = solution';
111
112 save(fullfile(['../data/JR_initial_QPLS_' sampling '.mat']), ...
113     'initial_images');
114

```

JR_PETMRI_QPLS_script_semilla.m

```
1  %% JR_PETMRI_QPLS_wrapper('full', 538, 0.00228)
2
3  clear
4
5  [sampling, alpha, beta] = deal('full', 538, 0.00228);
6
7  run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
8
9  param = [alpha, beta];
10 load(fullfile('..', 'data', ['JR_initial_QPLS_' sampling '.mat']))
11
12 solution = JR_PETMRI_QPLS(initial_images, ...
13 data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
14 data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
15 options{index.mri}.sigma_abs, param);
16
17 initial_images = solution';
18
19 save(fullfile(['../data/JR_initial_LPLS_' sampling '.mat']), ...
20 'initial_images');
21
22 %% JR_PETMRI_QPLS_wrapper('lines2', 386, 0.00386)
23
24 clear
25
26 [sampling, alpha, beta] = deal('lines2', 386, 0.00386);
27
28 run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
29
30 param = [alpha, beta];
31 load(fullfile('..', 'data', ['JR_initial_QPLS_' sampling '.mat']))
32
33 solution = JR_PETMRI_QPLS(initial_images, ...
34 data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
35 data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
36 options{index.mri}.sigma_abs, param);
37
38 initial_images = solution';
39
40 save(fullfile(['../data/JR_initial_LPLS_' sampling '.mat']), ...
41 'initial_images');
42
43 %% JR_PETMRI_QPLS_wrapper('radial20', 722, 0.00216)
44
45 clear
46
47 [sampling, alpha, beta] = deal('radial20', 722, 0.00216);
48
49 run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
50
51 param = [alpha, beta];
```

```

52 load(fullfile('..', 'data', ['JR_initial_QPLS_' sampling '.mat']))
53
54 solution = JR_PETMRI_QPLS(initial_images, ...
55 data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
56 data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
57 options{index.mri}.sigma_abs, param);
58
59 initial_images = solution';
60
61 save(fullfile(['../data/JR_initial_LPLS_' sampling '.mat']), ...
62 'initial_images');
63
64 %% JR_PETMRI_QPLS_wrapper('radial15', 399, 0.00328)
65
66 clear
67
68 [sampling, alpha, beta] = deal('radial15', 399, 0.00328);
69
70 run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
71
72 param = [alpha, beta];
73 load(fullfile('..', 'data', ['JR_initial_QPLS_' sampling '.mat']))
74
75 solution = JR_PETMRI_QPLS(initial_images, ...
76 data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
77 data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
78 options{index.mri}.sigma_abs, param);
79
80 initial_images = solution';
81
82 save(fullfile(['../data/JR_initial_LPLS_' sampling '.mat']), ...
83 'initial_images');
84
85 %% JR_PETMRI_QPLS_wrapper('spiralHigh', 129, 0.0104)
86
87 clear
88
89 [sampling, alpha, beta] = deal('spiralHigh', 129, 0.0104);
90
91 run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
92
93 param = [alpha, beta];
94 load(fullfile('..', 'data', ['JR_initial_QPLS_' sampling '.mat']))
95
96 solution = JR_PETMRI_QPLS(initial_images, ...
97 data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
98 data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
99 options{index.mri}.sigma_abs, param);
100
101 initial_images = solution';
102
103 save(fullfile(['../data/JR_initial_LPLS_' sampling '.mat']), ...
104 'initial_images');
105

```

```

106 %% JR_PETMRI_QPLS_wrapper('spiralUni', 455, 0.00328)
107
108 clear
109
110 [sampling, alpha, beta] = deal('spiralUni', 455, 0.00328);
111
112 run(fullfile('..', 'data', ['JR_data_' sampling '.m']));
113
114 param = [alpha, beta];
115 load(fullfile('..', 'data', ['JR_initial_QPLS_' sampling '.mat']))
116
117 solution = JR_PETMRI_QPLS(initial_images, ...
118 data{index.pet}, operator{index.pet}, adjoint{index.pet}, 1e-3, ...
119 data{index.mri}, operator{index.mri}, adjoint{index.mri}, ...
120 options{index.mri}.sigma_abs, param);
121
122 initial_images = solution';
123
124 save(fullfile(['../data/JR_initial_LPLS_' sampling '.mat']), ...
125 'initial_images');
126

```

enmascarar.m

```

1 function [s_kspace_masked, mask, s_kspace_sampled] = enmascarar(k_espacio, sampling)
2
3 load(fullfile('otros/data', ['JR_data' '.mat']), 'data');
4 s_kspace = data{2};
5 clear data
6
7 load(fullfile('otros/data', ['JR_data_' sampling '.mat']), 'data');
8 s_kspace_sampling = data{2};
9 clear data
10
11 [index, mask] = determinacion_mascara(s_kspace, s_kspace_sampling);
12
13 s_kspace_masked = zeros(size(mask));
14 s_kspace_masked(index) = k_espacio(index);
15
16 s_kspace_sampled = k_espacio(index);
17
18 end
19
20 function [index, mask] = determinacion_mascara(s_kspace, s_kspace_sampling)
21 % Si cada dato del s_kspace es único, se puede determinar la
22 % asignación única para la geometría de cada tipo de muestreo.
23 unicidad_datos = numel(unique(s_kspace)) == numel(s_kspace);
24 if unicidad_datos
25 sampling_size = numel(s_kspace_sampling);
26 mask = zeros(size(s_kspace));

```

```

27     index = zeros(sampling_size, 1);
28     for k = 1 : sampling_size
29         mask(s_kspace == s_kspace_sampling(k)) = 1;
30         index(k) = find(s_kspace == s_kspace_sampling(k));
31     end
32 end
33 mask = logical(reshape(mask, sqrt(numel(s_kspace))*ones(1,2)));
34 end

```

fantoma.m

```

1  function [fantoma_pet, fantoma_mri] = fantoma(n)
2
3  P = phantom(n);
4
5  fantoma_mri = P;
6  %fantoma_mri(P>0.2 & P<0.3) = 0.3;
7  fantoma_mri = agregar_circulos(fantoma_mri,0.032,0.9);
8  fantoma_mri = agregar_rectangulos(fantoma_mri,0);
9
10 fantoma_pet = P;
11 % Aclarado de fondo
12 fantoma_pet(P>=0.1 & P<=0.4) = 0.4;
13 % Oscurecido de contorno
14 fantoma_pet(P>=1.0) = 0.9;
15 fantoma_pet = agregar_rectangulos(fantoma_pet,0.9);
16 fantoma_pet = agregar_circulos(fantoma_pet,0.030,0);
17 fantoma_pet = agregar_circulo(fantoma_pet,[0.270 0.700],0.020,0);
18 fantoma_pet = agregar_circulo(fantoma_pet,[0.730 0.700],0.020,0);
19
20 end
21
22 function fantoma = agregar_rectangulos(fantoma,tono)
23     [r,c] = size(fantoma);
24     fantoma(unique(ceil(0.78*r)+ceil(-0.065*r:0.065*r)), ...
25             unique(ceil(0.385*c)+ceil(-0.006*c:0.006*c)) ) = tono;
26     fantoma(unique(ceil(0.78*r)+ceil(-0.065*r:0.065*r)), ...
27             unique(ceil(0.415*c)+ceil(-0.006*c:0.006*c)) ) = tono;
28     fantoma(unique(ceil(0.78*r)+ceil(-0.065*r:0.065*r)), ...
29             unique(ceil(0.48*c)+ceil(-0.01*c:0.01*c)) ) = tono;
30     fantoma(unique(ceil(0.78*r)+ceil(-0.065*r:0.065*r)), ...
31             unique(ceil(0.52*c)+ceil(-0.01*c:0.01*c)) ) = tono;
32     fantoma(unique(ceil(0.78*r)+ceil(-0.065*r:0.065*r)), ...
33             unique(ceil(0.59*c)+ceil(-0.015*c:0.015*c)) ) = tono;
34     fantoma(unique(ceil(0.78*r)+ceil(-0.065*r:0.065*r)), ...
35             unique(ceil(0.65*c)+ceil(-0.015*c:0.015*c)) ) = tono;
36 end
37
38 function fantoma = agregar_circulo(fantoma,centro,radio,tono)
39     [r,c] = size(fantoma);

```

```

40     centro = size(fantoma).*centro;
41     radio = max(size(fantoma))*radio;
42     [X, Y] = meshgrid(1:r, 1:c);
43     circulo = sqrt((X - centro(1)).^2 + (Y - centro(2)).^2) <= radio;
44     fantoma(circulo) = tono;
45 end
46
47 function fantoma = agregar_circulos(fantoma,radio,tono)
48     fantoma = agregar_circulo(fantoma,[0.32 0.235],radio,tono);
49     fantoma = agregar_circulo(fantoma,[0.4 0.170],radio,tono);
50     fantoma = agregar_circulo(fantoma,[0.5 0.140],radio,tono);
51     fantoma = agregar_circulo(fantoma,[0.6 0.170],radio,tono);
52     fantoma = agregar_circulo(fantoma,[0.68 0.235],radio,tono);
53 end

```

gaussian_noice.m

```

1 function y = gaussian_noice(x,level_noise)
2     y = x + level_noise * abs(max(x)) * randn(size(x));
3 end

```

generar_data.m

```

1 %% Parámetros
2 clear
3
4 run otros/data/JR_data
5
6 level_noise = cell(1,2);
7 level_noise{index.pet} = 1;
8 level_noise{index.mri} = 0.001;
9
10 %% JR_data.mat
11
12 [ground_truth{index.pet}, ground_truth{index.mri}] = fantoma(unique(options{3}.size));
13
14 data{index.pet} = operator{index.pet}(ground_truth{index.pet});
15 data{index.pet} = poissonian_noise(data{index.pet},level_noise{index.pet});
16
17 data{index.mri} = operator{index.mri}(ground_truth{index.mri});
18 data{index.mri} = gaussian_noise(data{index.mri},level_noise{index.mri});
19
20 save(fullfile('../data/JR_data.mat'), 'data', 'ground_truth', 'index', 'options');
21
22 %% JR_data_sampling.mat
23
24 sampling = {'full', 'lines2', 'radial15', 'radial20', 'spiralHigh', 'spiralUni'};

```

```

25 cellfun(@data_sampling_generator, sampling)
26
27 %% JR_initial_QPLS_sampling.mat
28
29 generar_JR_PETMRI_JTV_script_semilla
30 JR_PETMRI_JTV_script_semilla
31
32 %% JR_initial_LPLS_sampling.mat
33
34 generar_JR_PETMRI_QPLS_script_semilla
35 JR_PETMRI_QPLS_script_semilla
36
37 %% Herramientas
38
39 function data_sampling_generator(sampling)
40     name = fullfile('data', ['JR_data_' sampling '.mat']);
41
42     run otros/data/JR_data
43
44     load('../data/JR_data.mat','data');
45     data_sampling = data;
46
47     [~,~,data_sampled] = enmascarar(data_sampling{index.mri},sampling);
48     data_sampling{index.mri} = JR_column(data_sampled);
49
50     load(fullfile('otros', name),'options');
51     options_sampling = options;
52
53     clear data options
54     data = data_sampling;
55     options = options_sampling;
56
57     save(fullfile('..', name), 'data', 'options');
58 end

```

pet.m

```

1 function [sinograma, angles, image] = pet(x,angles)
2     if nargin < 2
3         n_angles = 300;
4         angles = -90:(90+90)/(n_angles-1):90;
5     end
6     sinograma_extendido = radon(x,angles);
7     filas_eliminar = ceil((size(sinograma_extendido,1) - size(x,1))/2);
8     sinograma = sinograma_extendido(filas_eliminar:end-filas_eliminar,:);
9     image = sinograma;
10    sinograma = sinograma(:);
11 end

```


poissonian_noise.m

```
1 function y = poissonian_noise(x, level_noise)
2     y = level_noise * poissrnd(x/level_noise);
3 end
```

JR/matlab/otros

grafico_discretizacion_CT.m

```
1 clear
2
3 resolucion = 12;
4 fantoma = phantom(resolucion);
5
6 figure
7
8 subplot('Position',[0.05 0 0.45 1])
9 angulo_grados = 0;
10 iniciar_contador = 1;
11 escaneo_fantoma_CT_angulos(fantoma, angulo_grados, iniciar_contador)
12
13 subplot('Position',[0.55 0 0.45 1])
14 angulo_grados = 30;
15 iniciar_contador = resolucion + 1;
16 escaneo_fantoma_CT_angulos(fantoma, angulo_grados, iniciar_contador)
17 print('-dsvg', fullfile('IP_31_1_015001_suppdata_1\matlab\otros', ...
18     'imagenes\grafico_discretizacion_CT.svg'))
19
20 function escaneo_fantoma_CT_angulos(fantoma, angulo_grados, iniciar_contador)
21     escalado_inferior = -0.3;
22     escalado_superior = 1.4;
23
24     resolucion = sqrt(numel(fantoma));
25
26     P = ones(resolucion + 1);
27     P(1:end-1, 1:end-1) = 1 - rot90(fantoma, 3)';
28
29     hold on
30     pcolor(P)
31     colormap('gray')
32     color = 0.5*ones(1,3);
33     font_size = 4;
34     agregar_etiquetas_densidad_discretizada_f(resolucion, font_size, color)
35     agregar_haces_paralelos(resolucion, angulo_grados, iniciar_contador)
36     configuracion_general(resolucion, escalado_inferior, escalado_superior)
37 end
38
```

```

39 function agregar_haces_paralelos(resolucion, angulo_grados, iniciar_contador)
40     angulo = angulo_grados * pi/180;
41     centro = [resolucion/2, resolucion/2];
42     rotacional = [cos(angulo), -sin(angulo); sin(angulo), cos(angulo)];
43
44     P_inicial_0 = [-0.2 0.12].*resolucion;
45     P_final_0 = [1.25 0.12].*resolucion;
46
47     separacion = resolucion / (resolucion + 0);
48     for i = 0 : resolucion - 1
49         P_inicial = [P_inicial_0(1), P_inicial_0(2) + i*separacion];
50         P_final = [P_final_0(1), P_final_0(2) + i*separacion];
51
52         P_inicial_rotado = centro + (P_inicial - centro) * rotacional;
53         P_final_rotado = centro + (P_final - centro) * rotacional;
54
55         if i == 0
56             P_final_rotado_0 = P_final_rotado;
57         end
58
59         haz_paralelo = [P_inicial_rotado(1), P_final_rotado(1);
60             P_inicial_rotado(2), P_final_rotado(2)];
61         color = 0*ones(1,3);
62         plot(haz_paralelo(1,:), haz_paralelo(2,:), ...
63             'Color', color, 'LineWidth', 0.1)
64         plot(P_inicial_rotado(1), P_inicial_rotado(2), ...
65             'Marker', '.', 'MarkerSize', 6, 'Color', color)
66         latex = sprintf('\mathbf{m}_{%d}', i+iniciar_contador);
67         text(P_inicial_rotado(1) - 1.5, P_inicial_rotado(2) + 0.15, latex, ...
68             'Interpreter', 'latex', 'FontSize', 6, 'Color', 'k')
69     end
70     barra_emisor = [P_final_rotado_0(1), P_final_rotado(1);
71         P_final_rotado_0(2), P_final_rotado(2)];
72     plot(barra_emisor(1,:), barra_emisor(2,:), 'Color', color, 'LineWidth', 3)
73 end
74
75 function agregar_etiquetas_densidad_discretizada_f(resolucion, font_size, color)
76     k = resolucion^2;
77     for y = 1 : resolucion
78         for x = 1 : resolucion
79             latex = sprintf('\mathbf{f}_{%d}', k);
80             text(resolucion - x + 1.0, y+0.7, latex, ...
81                 'Interpreter', 'latex', 'FontSize', font_size, 'Color', color)
82             k = k - 1;
83         end
84     end
85 end
86
87 function configuracion_general(resolucion, escalado_inferior, escalado_superior)
88     xlim([escalado_inferior*resolucion escalado_superior*resolucion+1])
89     ylim([escalado_inferior*resolucion escalado_superior*resolucion+1])
90     set(gca, 'Visible', 'off', 'XTick', [], 'YTick', [])
91     xticklabels([])
92     yticklabels([])

```

```
93     pbaspect([1 1 1])
94 end
```

grafico_espaciok_MRI.m

```
1  clear
2
3  resolucion = 128;
4  level_noise = 0.05;
5
6  load otros/data/JR_data.mat index options
7
8  [~, fantoma] = fantoma(resolucion);
9
10 a = 0.1;
11 espacio_k = fftshift(fft2(fantoma)) / sqrt(numel(fantoma));
12 espacio_k_modulo = abs(espacio_k);
13 espacio_k_log = log(a + abs(espacio_k));
14 espacio_k_ruido = gaussian_noise(espacio_k_log, level_noise);
15
16 figure
17
18 subplot('Position',[000 1/3 1/3 1/3])
19 imshow(fantoma)
20 title('Fantoma MRI: $v$', 'Interpreter', 'latex')
21 configuracion_general
22
23 subplot('Position',[1/3 1/3 1/3 1/3]);
24 imagesc(espacio_k_log)
25 title('Espacio\textendash$k$ de $v$', 'Interpreter', 'latex')
26 colormap(gca, hot)
27 configuracion_general
28
29 subplot('Position',[2/3 1/3 1/3 1/3])
30 imagesc(espacio_k_ruido)
31 title(['Espacio\textendash$k$ de $v$'; 'con ruido blanco del $5\%$ '], ...
32       'Interpreter', 'latex');
33 colormap(gca, hot)
34 configuracion_general
35
36 print('-dsvg', fullfile('otros', '\imagenes\grafico_espaciok_MRI.svg'))
37
38 function configuracion_general()
39     set(gca, 'XTick', [], 'YTick', [])
40     colorbar('TickLabelInterpreter', 'latex')
41     xticklabels([])
42     yticklabels([])
43     pbaspect([1 1 1])
44 end
```

grafico_reconstruccion_full.m

```
1 clear
2
3 run ../data/JR_data.m
4 load ../data/JR_data.mat
5
6 PET = struct;
7 PET.NR = ...
8     load('../results/JR_PET/JR_PET_20230801_2156_results.mat');
9 PET.TV = ...
10     load('../results/JR_PET_TV/JR_PET_TV_20230801_2156_results.mat');
11 PET.JTV = ...
12     load(['../results/JR_PETMRI_full_JTV/JR_PETMRI_full_JTV_' ...
13         '20230801_2159_PET_results.mat']);
14 PET.QPLS = ...
15     load(['../results/JR_PETMRI_full_QPLS/JR_PETMRI_full_QPLS_' ...
16         '20230802_0147_PET_results.mat']);
17 PET.LPLS = ...
18     load(['../results/JR_PETMRI_full_LPLS/JR_PETMRI_full_LPLS_' ...
19         '20230802_0155_PET_results.mat']);
20
21 MRI = struct;
22 MRI.NR = ...
23     load('../results/JR_MRI_full/JR_MRI_full_20230801_2154_results.mat');
24 MRI.TV = ...
25     load('../results/JR_MRI_full_TV/JR_MRI_full_TV_20230801_2155_results.mat');
26 MRI.JTV = ...
27     load(['../results/JR_PETMRI_full_JTV/JR_PETMRI_full_JTV_' ...
28         '20230801_2159_MRI_results.mat']);
29 MRI.QPLS = ...
30     load(['../results/JR_PETMRI_full_QPLS/JR_PETMRI_full_QPLS_' ...
31         '20230802_0147_MRI_results.mat']);
32 MRI.LPLS = ...
33     load(['../results/JR_PETMRI_full_LPLS/JR_PETMRI_full_LPLS_' ...
34         '20230802_0155_MRI_results.mat']);
35
36 sampling = 'full';
37
38 u = ground_truth(index.pet);
39 v = ground_truth(index.mri);
40 espacio_k = Im(data(index.mri));
41 espacio_k_graficar = log(abs(espacio_k));
42 [espacio_k_muestreo, patron_muestreo] = enmascarar(espacio_k, sampling);
43 espacio_k_muestreo = log(abs(espacio_k_muestreo));
44
45 PET.NR.error = error_relativo(u, PET.NR.sol);
46 MRI.NR.error = error_relativo(v, MRI.NR.sol);
47 PET.TV.error = error_relativo(u, PET.TV.sol);
48 MRI.TV.error = error_relativo(v, MRI.TV.sol);
49 PET.JTV.error = error_relativo(u, PET.JTV.sol);
50 MRI.JTV.error = error_relativo(v, MRI.JTV.sol);
51 PET.QPLS.error = error_relativo(u, PET.QPLS.sol);
```

```

52 MRI.QPLS.error = error_relativo(v,MRI.QPLS.sol);
53 PET.LPLS.error = error_relativo(u,PET.LPLS.sol);
54 MRI.LPLS.error = error_relativo(v,MRI.LPLS.sol);
55
56 max_error = 0;
57 tipo = ["PET" "MRI"];
58 muestreo = ["NR" "TV" "JTV" "QPLS" "LPLS"];
59 for i = 1 : numel(tipo)
60     for j = 1 : numel(muestreo)
61         error = eval([char(tipo(i)) '.' char(muestreo(j)) '.error']);
62         max_error = max(max_error,max(max(error)));
63     end
64 end
65
66 n_filas = 6;
67 n_columnas = 4;
68
69 MS = 0.06;
70 MB = 0.01;
71 MI = 0.1;
72 sep = 0.01;
73
74 dimension = [1 1]*0.15;
75
76 load spine
77
78 figure('Position',[0 0 500 500])
79
80 subplot('Position', [0*dimension(1)+MI+0*sep 5*dimension(2)+MS dimension])
81 imagesc(u)
82 colormap(gca, map)
83 title('Fantoma PET','Interpreter','latex')
84 configuracion_general
85
86 subplot('Position', [1*dimension(1)+MI+1*sep 5*dimension(2)+MS dimension])
87 imagesc(v)
88 colormap(gca, map)
89 title('Fantoma MRI','Interpreter','latex')
90 configuracion_general
91
92 subplot('Position', [2*dimension(1)+MI+2*sep 5*dimension(2)+MS dimension])
93 imagesc(espacio_k_graficar)
94 title('Datos MRI','Interpreter','latex')
95 colormap(gca, 'hot')
96 configuracion_general
97
98 subplot('Position', [3*dimension(1)+MI+3*sep 5*dimension(2)+MS dimension])
99 imagesc(espacio_k_muestreo)
100 title('Muestreo MRI','Interpreter','latex')
101 colormap(gca, 'hot')
102 configuracion_general
103 annotation('textbox',...
104     [4*dimension(1)+MI+3*sep, 6*dimension(2)+MS, dimension(2), 0.04],...
105     'VerticalAlignment','cap',...

```

```

106     'String', {sprintf('\texttt{%s}', sampling)}, ...
107     'Rotation', 270, ...
108     'Interpreter', 'latex', ...
109     'HorizontalAlignment', 'center', ...
110     'FontSize', 10, ...
111     'FitBoxToText', 'off', ...
112     'EdgeColor', 'none')
113
114 % Reconstrucciones
115
116 for k = 1 : 5
117
118     subplot('Position', ...
119           [0*dimension(1)+MI+0*sep (5-k)*dimension(2)+MB dimension])
120     imagesc(eval(['PET.' char(muestreo(k)) '.sol']))
121     if k == 1; title('PET', 'Interpreter', 'latex'); end
122     if k == 1
123         ylabel('Sin Regul.', 'Interpreter', 'latex')
124     else
125         ylabel(char(muestreo(k)), 'Interpreter', 'latex')
126     end
127     colormap(gca, map)
128     configuracion_general
129
130     subplot('Position', ...
131           [1*dimension(1)+MI+1*sep (5-k)*dimension(2)+MB dimension])
132     imagesc(eval(['MRI.' char(muestreo(k)) '.sol']))
133     if k == 1; title('MRI', 'Interpreter', 'latex'); end
134     colormap(gca, map)
135     configuracion_general
136
137     subplot('Position', ...
138           [2*dimension(1)+MI+2*sep (5-k)*dimension(2)+MB dimension])
139     imagesc(eval(['PET.' char(muestreo(k)) '.error']))
140     if k == 1; title('Error PET', 'Interpreter', 'latex'); end
141     colormap(gca, flipud(pink))
142     clim([0 max_error])
143     configuracion_general
144
145     subplot('Position', ...
146           [3*dimension(1)+MI+3*sep (5-k)*dimension(2)+MB dimension])
147     imagesc(eval(['MRI.' char(muestreo(k)) '.error']))
148     if k == 1; title('Error MRI', 'Interpreter', 'latex'); end
149     colormap(gca, flipud(pink))
150     clim([0 max_error])
151     if k==1
152         colorbar('TickLabelInterpreter', 'latex', 'Position', ...
153               [0*dimension(1)+MI-2.5*sep, 5*dimension(2)+MS, 0.015, dimension(2)])
154     end
155     configuracion_general
156
157 end
158
159 print('-dsvg', ['otros\imagenes\grafico_reconstruccion_', sampling, '.svg'])

```

```

160
161 function configuracion_general()
162     set(gca, 'XTick', [], 'YTick', [])
163     xticklabels([])
164     yticklabels([])
165 end
166
167 function er = error_relativo(x,y)
168     er = abs(x - y)/norm(reshape(x,[1 numel(x)]));
169 end

```

grafico_sinograma_PET.m

```

1  clear
2
3  resolucion = 128;
4  level_noise = 1;
5
6  load otros/data/JR_data.mat index options
7  psf = options{index.pet}.psf;
8  scaling = options{index.pet}.scaling;
9
10 fantoma = fantoma(resolucion);
11
12 [sinograma, ~, sinograma_original] = pet(fantoma);
13 dim_128 = size(sinograma_original);
14 sinograma_blur = reshape(imfilter(sinograma,psf,'conv'), dim_128);
15 sinograma_scaling = reshape(scaling*sinograma_blur, dim_128);
16 sinograma_noise = ...
17     reshape(poissonian_noice(sinograma_scaling,level_noise), dim_128);
18
19 dim_128 = 0.21;
20 dim_300 = 0.351;
21
22 figure('Position',[0 0 500 500])
23
24 % SUPERIOR
25
26 subplot('Position',[0.0, 1.6*dim_128, dim_300, (128/300)*dim_300])
27 imagesc(sinograma_original)
28 title('Sinograma Radon: $\mathbf{R}u$', 'Interpreter', 'latex')
29 configuracion_general
30 colorbar('TickLabelInterpreter', 'latex', ...
31     'Position',[dim_300 + 0.01, 1.6*dim_128, 0.02, (128/300)*dim_300])
32
33 annotation('textarrow', ...
34     [dim_300+0.01+0.02, dim_300+(1-dim_300-dim_300)/2] + 0.02*[2, -1],...
35     ( 1.6*dim_128 + (128/300)*dim_300/2 ) * [1, 1], ...
36     'LineWidth',2, 'HeadStyle', 'cback2', 'HeadWidth',6, 'HeadLength',10)
37
38 subplot('Position',[ ...

```

```

39     dim_300 + (1 - dim_300 - dim_300)/2 ...
40     1.6*dim_128 ...
41     dim_300 ...
42     (128/300)*dim_300])
43 imagesc(sinograma_blur)
44 title('Sinograma con desenfoque:  $\mathcal{D}(\mathbf{R}u)$ ', ...
45       'Interpreter','latex')
46 % imagesc(sinograma_scaling)
47 % title('Sinograma con escalamiento:  $\mu\mathcal{D}(\mathbf{R}u)$ ', ...
48 %       'Interpreter','latex')
49 configuracion_general
50 colorbar('TickLabelInterpreter', 'latex', ...
51         'Position',[ ...
52         0.5 + dim_300 + 0.01 ...
53         1.6*dim_128 ...
54         0.02 ...
55         (128/300)*dim_300])
56
57 annotation('textarrow', ...
58           ( 0.5 + dim_300 / 2 ) * [1, 1],...
59           [ 1.6*dim_128, 0.01 + dim_128 ] + 0.02 * [-1, 2], ...
60           'LineWidth',2,'HeadStyle','cback2','HeadWidth',6,'HeadLength',10)
61
62 % INFERIOR
63
64 subplot('Position',[0.0, 0.01, dim_128, dim_128])
65 imagesc(fantoma)
66 title('Fantoma PET:  $\$u\$$ ','Interpreter','latex')
67 colormap(gca, 'gray')
68 configuracion_general
69 colorbar('TickLabelInterpreter', 'latex', ...
70         'Position',[dim_128 + 0.01, 0.01, 0.02, dim_128])
71
72 annotation('textarrow', ...
73           ( dim_128 / 2 ) * [1, 1],...
74           [ 0.01 + dim_128, 1.6*dim_128 ] + 0.02 * [2, -1], ...
75           'LineWidth',2,'HeadStyle','cback2','HeadWidth',6,'HeadLength',10)
76
77 subplot('Position',[ ...
78         dim_128 + (1 - dim_128 - (300/128)*dim_128)/2 ...
79         0.01 ...
80         (300/128)*dim_128 ...
81         dim_128])
82 imagesc(sinograma_noise)
83 title('Sinograma con ruido:  $\mathrm{Pois}(\mu\mathcal{D}(\mathbf{R}u))$ ', ...
84       'Interpreter','latex')
85 configuracion_general
86 colorbar('TickLabelInterpreter', 'latex', ...
87         'Position',[ ...
88         dim_128 + (300/128)*dim_128 + (1 - dim_128 - (300/128)*dim_128)/2 + 0.01 ...
89         0.01 ...
90         0.02 ...
91         dim_128])
92

```



```
93 print('-dsvg', fullfile('IP_31_1_015001_suppdata_1\matlab\otros', ...
94     '\imagenes\grafico_sinograma_PET.svg'))
95
96 function configuracion_general()
97     set(gca, 'XTick', [], 'YTick', [])
98     xticklabels([])
99     yticklabels([])
100 end
```

Bibliografía

- [1] Anna. Derenzo phantom. [Enlace](#), 2023. [Accedido: 20 de junio de 2023].
- [2] Tom M. Apostol. *Calculus, Vol. I*. John Wiley & Sons, 2 edition, 1991.
- [3] George Bachman and Lawrence Narici. *Functional Analysis*. Academic Press, New York and London, 1966.
- [4] Dale L. Bailey, Michael N. Maisey, David W. Townsend, and Peter E. Valk. *Positron emission tomography*, volume 2. Springer, 2005.
- [5] Stephen Becker. L-BFGS-B: Mex wrapper. [Enlace](#), 2014.
- [6] Stephen Becker. L-BFGS-B-C: L-BFGS-B, converted from fortran to c, with matlab wrapper. [Enlace](#), 2023.
- [7] Mario Bertero, Patrizia Boccacci, and Christine De Mol. *Introduction to inverse problems in imaging*. CRC press, 2 edition, 2022.
- [8] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.
- [9] U. Busch. Wilhelm conrad roentgen. el descubrimiento de los rayos x y la creación de una nueva profesión médica. *Revista Argentina de Radiología*, 80(4):298–307, 2016.
- [10] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

- [11] Renzo Cairoli. *Algèbre linéaire*. Presses polytechniques et universitaires romandes, Lausanne, Switzerland, 2 edition, 2004.
- [12] Keith Devlin. *The Language of Mathematics: Making the Invisible Visible*. W.H. Freeman & Company, 2000.
- [13] Apostolos Doxiadis, Christos H. Papadimitriou, and Alecos Papadatos. *Logicomix: una búsqueda épica de la verdad*. Salamandra Graphic, 2020.
- [14] Matthias J. Ehrhardt. Supplementary material for joint reconstruction of PET-MRI by exploiting structural similarity. [Enlace](#), 2015.
- [15] Matthias J. Ehrhardt, Kris Thielemans, Luis Pizarro, David Atkinson, Sébastien Ourselin, Brian F Hutton, and Simon R Arridge. Joint reconstruction of PET-MRI by exploiting structural similarity. *Inverse Problems*, 31(015001):23, 2015.
- [16] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.
- [17] Gerald B. Folland. *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, Inc., New York, second edition, 1999.
- [18] Pascal Getreuer. tvreg: Variational imaging methods for denoising, deconvolution, inpainting, and segmentation. [Enlace](#), 2010.
- [19] Deiby Giraldo and Luisa Marín. Historia de la tomografía. [Enlace](#), 2022. [Accedido: 22 de mayo de 2023].
- [20] Georgina González and Manuel González Pieri. Historia de la radiología. *Rev. Cient. Esc. Univ. Cienc. Salud*, pages 45–48, 2017.
- [21] Stanley I. Grossman and José J. Flores. *Álgebra Lineal*. McGraw-Hill, México, D.F., 7 edition, 2012.
- [22] Douglas R. Hofstadter. *Gödel, Escher, Bach: Una Eterna Trenza Dorada*. Consejo nacional de ciencia, 1979.
- [23] Joseph B. Keller. Inverse problems. *The American Mathematical Monthly*, 83:107–118, 1976.

- [24] Andreas Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*, volume 120 of *Applied Mathematical Sciences*. Springer, 3 edition, 2020.
- [25] Morris Kline. *Mathematics: The Loss of Certainty*. Oxford University Press, 1980.
- [26] Keith Knight. *Mathematical Statistics*. Texts in Statistical Science. Chapman & Hall, 1999.
- [27] Florian Knoll, Martin Holler, Thomas Koesters, Ricardo Otazo, Kristian Bredies, and Daniel K Sodickson. Joint mr-pet reconstruction using a multi-channel image regularizer. *IEEE transactions on medical imaging*, 36(1):1–16, 2016.
- [28] Bernard Kolman and David R. Hill. *Álgebra Lineal*. Pearson Educación, México, D.F., 8 edition, 2006.
- [29] Vilmos Komornik. *Précis d'Analyse Réelle, Volume 2. Analyse fonctionnelle, Intégrale de Lebesgue, Espaces fonctionnels*. Ellipse, 2002.
- [30] Erwin Kreyszig. *Advanced engineering mathematics*. John Wiley & Sons, 10 edition, 2011.
- [31] Alfred K. Louis. Medical imaging: state of the art and future development. *Inverse Problems*, 8(5):709, 1992.
- [32] Benoît Mandelbrot. *Los objetos fractales: forma, azar y dimensión*. Tusquets Editores, 1996.
- [33] Benoît Mandelbrot. *La geometría fractal de la naturaleza*. Tusquets Editores, 1997.
- [34] Víctor Martínez. Problemas inversos: los casi olvidados de la matemática educativa. In Patricia Lestón, editor, *Acta Latinoamericana de Matemática Educativa*, pages 439–447, México, DF, 2011. Comité Latinoamericano de Matemática Educativa.
- [35] MathWorks. phantom. [Enlace](#), 2023. [Accedido: 14 de agosto de 2023].

- [36] Abolfazl Mehranian, Martin A Belzunce, Claudia Prieto, Alexander Hammers, and Andrew J Reader. Synergistic pet and sense mr image reconstruction using joint sparsity regularization. *IEEE transactions on medical imaging*, 37(1):20–34, 2017.
- [37] Elliott Mendelson. *Introduction to Mathematical Logic*. Chapman and Hall/CRC, 6 edition, 2015.
- [38] Jesús Mosterín. *Los lógicos*. Espasa Libros, 2000.
- [39] Jennifer L. Müller and Samuli Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, 2012.
- [40] Frank Natterer. *The mathematics of computerized tomography*. SIAM, 2001.
- [41] nbro. How the vector space isomorphism between $\mathbb{R}^{m \times n}$ and \mathbb{R}^{mn} is established? [Enlace](#), Diciembre 2021. [Accedido: 5 de agosto de 2023].
- [42] Marcos Nebreda López. Historia de la radiografía: el origen de la imagen diagnóstica — campustraining.es. [Enlace](#), 2019. [Accedido: 22 de mayo de 2023].
- [43] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2006.
- [44] Charles C. Pinter. *Set Theory*. Addison-Wesley Publishing Company, 1971.
- [45] The Nobel Prize. Godfrey N. Hounsfield - Biographical. [Enlace](#), 2023. [Accedido: 22 de mayo de 2023].
- [46] Julian Rasch, Eva-Maria Brinkmann, and Martin Burger. Joint reconstruction via coupled bregman iterations with applications to pet-mr imaging. *Inverse Problems*, 34(1):014001, 2017.
- [47] David S. Richeson. *Tales of Impossibility: The 2000-Year Quest to Solve the Mathematical Problems of Antiquity*. Princeton University Press, 2019.

- [48] Bertrand Russell. *Introduction to Mathematical Philosophy*. Gutenberg, 2013.
- [49] Jun Shao. *Mathematical Statistics*. Springer, 2nd edition, 2003.
- [50] Tristan van Leeuwen. Lecture notes on inverse problems and imaging. [Enlace](#), 2023. [Accedido: 20 de junio de 2023. Licencia: CC BY-NC 4.0. Última actualización: 8-5-2023].
- [51] Dennis D. Wackerly, William Mendenhall III, and Richard L. Scheaffer. *Mathematical Statistics with Applications*. Thomson, 7 edition, 2008.
- [52] Wikipedia. Shepp–logan phantom. [Enlace](#), 2023. [Accedido: 14 de agosto de 2023].
- [53] Ying Zhang and Xiaoqun Zhang. Pet-mri joint reconstruction with common edge weighted total variation regularization. *Inverse Problems*, 34(6):065006, 2018.
- [54] Ciyou Zhu, Richard Byrd, Jorge Nocedal, and Jose Luis Morales. L-BFGS-B software for large-scale bound-constrained optimization. [Enlace](#), 2011.
- [55] Dennis G. Zill, Warren S. Wright, and Michael R. Cullen. *Matemáticas Avanzadas Para Ingeniería*. McGraw-Hill Interamerican, 2012.