Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales

Modelos y soluciones para la restauración de imágenes fuera de foco

Tesis de Licenciatura en Ciencias de Datos

Delfina Belén Comerso Salzer

Directora: Gabriela Jeronimo
Codirectora: Malena Español
Buenos Aires, 2025

MODELOS Y SOLUCIONES PARA LA RESTAURACIÓN DE IMÁGENES FUERA DE FOCO

Los problemas de deconvolución ciega y semiciega que surgen en el contexto de restauración de imágenes fuera de foco pueden ser modelados como problemas de mínimos cuadrados no lineales separables. El método de proyección de variables (VarPro) es un método eficiente para la resolución de este tipo de problemas, transformando el problema original en un problema de mínimos cuadrados no lineales reducido, que puede ser resuelto mediante el método de Gauss-Newton.

En este trabajo proponemos extender el método VarPro para resolver problemas de mínimos cuadrados no lineales separables con regularización Tikhonov en forma general, para abordar el problema de deconvolución semiciega. Introducimos términos regularizantes sobre los parámetros del operador de desenfoque e investigamos distintas variantes del método cuando se utilizan diferentes aproximaciones de la matriz Jacobiana. Estudiamos los casos especiales en los que existe una descomposición espectral conjunta de operadores directos y de regularización, proporcionando métodos eficientes para calcular los Jacobianos y la solución de los subproblemas lineales. Presentamos un análisis de convergencia local del método bajo hipótesis de diferenciabilidad y continuidad Lipschitz de los operadores involucrados, demostrando que bajo condiciones adecuadas el método puede alcanzar tasas de convergencia superlineales e incluso cuadráticas. Para los casos donde la demanda computacional para calcular Jacobianos y residuos en el método quasi-Newton se vuelve muy compleja se pueden utilizar métodos iterativos, específicamente LSQR, para calcular Jacobianos y residuos aproximados.

Esta tesis analiza el impacto de añadir estos nuevos términos regularizantes, así como las aproximaciones en el método de proyección de variables. Presentamos experimentos numéricos donde aplicamos los métodos propuestos para resolver problemas de deconvolución semiciega, con el fin de ilustrar y confirmar nuestros resultados teóricos y la eficacia del método.

Palabras claves: Método de proyección de variables, Regularización de Tikhonov, Deconvolución semiciega, Descomposiciones espectrales, Problemas inversos, LSQR.

MODELS AND SOLUTIONS FOR RESTORING OUT-OF-FOCUS IMAGES

Blind and semi-blind deconvolution problems arising in the context of out-of-focus image restoration can be modeled as separable nonlinear least squares problems. The variable projection method (VarPro) is an efficient method for solving this type of problems, transforming the original problem into a reduced nonlinear least squares problem, which can be solved using the Gauss-Newton method.

In this work, we propose to extend the VarPro method to solve separable nonlinear least squares problems with general-form Tikhonov regularization, to address the semi-blind deconvolution problem. We introduce regularization terms on the parameters of the blurring operator and investigate different variants of the method when different approximations of the Jacobian matrix are used. We study the special cases in which a joint spectral decomposition of direct and regularization operators exists, providing efficient methods for calculating the Jacobians and solving the linear subproblems. We present a local convergence analysis of the method under the assumptions of Lipschitz differentiability and continuity of the operators involved, demonstrating that under appropriate conditions, the method can achieve superlinear and even quadratic convergence rates. For cases where the computational demand for calculating Jacobians and residuals in the quasi-Newton method becomes very challenging, iterative methods, specifically LSQR, can be used as internal solvers to calculate Jacobians and approximate residuals.

This thesis analyzes the impact of adding these new regularization terms, as well as the approximations in the variable projection method. Numerical experiments are presented in which we apply the proposed methods to solve semi-blind deconvolution problems, to illustrate and confirm our theoretical results and the effectiveness of the method.

Keywords: Variable projection method, Tikhonov regularization, Semi-blind deconvolution, Spectral decompositions, Inverse problems, LSQR.

AGRADECIMIENTOS

A mi mamá, por enseñarme lo que es el esfuerzo y la dedicación, por empujarme siempre a superarme.

A mi familia por su apoyo incondicional, en especial cuando empecé la carrera en pandemia y todo era nuevo, confuso y frustrante. A mi perra Umma, por ser mi compañera más fiel.

A mis tutoras Gabriela y Malena por su dedicación, compromiso y paciencia. Todo este proceso fue muy desafiante, pero altamente gratificante, y esto se lo debo completamente a ustedes.

A *las mariposas al día*, mis amigos de datos y compu, por las cursadas, los momentos compartidos, las juntadas, que a veces eran de estudio... pero la mayoría de las veces no. La carrera no hubiera sido lo mismo sin ustedes.

A mis amigos de mate: Nico, Manu, Vicky, Lucho y Franco, los que conocí en mi primer año presencial y estuvieron siempre a lo largo de estos años. Gracias por el aguante, las charlas, las risas y por haber formado parte de este camino. A mis otros amigos de aplicada, que fui conociendo este último tiempo, gracias por el apoyo y la buena onda.

A la Facultad de Ciencias Exactas y Naturales por su excelencia académica y a toda la comunidad que la integra.

"I myself know nothing, except just a little—
enough to extract an argument from another man who is wise and to receive it fairly."

— Plato, Theaetetus 161b

Índice general

1. INTRODUCCIÓN

1.1. Motivación

El problema de la restauración de imágenes aparece en diversas aplicaciones, tales como reconocimiento de patrones, visión por computadora e inteligencia artificial. En muchos casos, como puede ser en el de imágenes médicas (tomografías computadas, resonancias magnéticas, etc.), se tiene solo una cantidad limitada de oportunidades para capturar una imagen. En estos casos, es necesario poder extraer información significativa a partir de las imágenes ruidosas obtenidas durante el proceso de adquisición de datos [1]. Si bien el proceso de desenfoque puede modelarse eficazmente mediante un proceso lineal, el reenfoque (enfocar imágenes desenfocadas) no es tan simple como invertir dicho proceso lineal y muchas veces produce resultados erróneos. El proceso de reenfocar pertenece a una clase de problemas importantes conocidos como *problemas discretos mal planteados* (ill-posed) [12].

Las imágenes digitales pueden ser representadas por matrices de píxeles, que denotamos con $\mathbf{X}$, donde cada píxel tiene asociada una posición en la matriz con un valor que indica la intensidad de la luz. Para facilitar su manipulación matemática y computacional, estas matrices pueden reformularse como vectores $\mathbf{x}$, apilando las columnas de $\mathbf{X}$ una encima de la otra. Suponiendo que el desenfoque es espacialmente invariante, es decir, que es igual en todas las regiones de la imagen, éste se puede modelar mediante una operación de convolución entre la imagen original y una función de dispersión del punto (PSF, por sus siglas en inglés), que describe cómo un punto de luz se distribuye en la imagen [1, 14]. En términos matriciales, este proceso puede escribirse como un sistema lineal de la forma

$$\mathbf{A}\mathbf{x} \approx \mathbf{b}, \tag{1.1}$$

donde $\mathbf{A}$ representa la transformación inducida por la PSF, también referida como operador de desenfoque, $\mathbf{x}$ es la imagen original vectorizada y $\mathbf{b}$ es la imagen desenfocada observada. El reenfoque se refiere al proceso de intentar obtener la imagen $\mathbf{x}$ original a partir de $\mathbf{b}$. En la práctica, este problema es típicamente mal condicionado y afectado por ruido, lo que dificulta la recuperación de $\mathbf{x}$. La deconvolución busca invertir este proceso para recuperar $\mathbf{x}$ a partir de $\mathbf{b}$. Cuando $\mathbf{A}$ es desconocida o solo parcialmente conocida, se habla de deconvolución ciega o semiciega, respectivamente.

El problema de deconvolución semiciega es especialmente complejo porque implica estimar simultáneamente la imagen original y ciertos parámetros del operador de desenfoque. Este problema se puede abordar mediante técnicas avanzadas de optimización no lineal, que buscan las mejores estimaciones para minimizar el error en la reconstrucción de la imagen. Uno de los métodos más avanzados para tratar la deconvolución semiciega es el método de proyección de variables (Variable Projection Method) [8]. Este método descompone el problema en dos subproblemas más manejables: uno lineal y otro no lineal. Alternando entre la resolución de estos subproblemas, se pueden obtener soluciones más precisas y robustas.

Como mencionamos al comienzo, invertir la ecuación (1.1) para obtener la solución $\mathbf{x}$ no da los resultados adecuados. En esta tesis vamos a considerar un método estándar para resolver el problema de deconvolución semiciega, el método de mínimos cuadrados con regularización de Tikhonov en forma general [11, 12, 13]. El método de Tikhonov consiste en resolver el problema de minimización

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{L}\mathbf{x}\|_2^2, \tag{1.2}$$

donde $\lambda > 0$ se llama parámetro de regularización y $\mathbf{L}$ es una matriz de regularización en $\mathbf{x}$.

1.2. Formulación general y trabajos previos

Consideramos el problema discreto mal planteado de la forma

$$\mathbf{A}(\mathbf{y})\,\mathbf{x} \approx \mathbf{b} = \mathbf{b}_{\text{true}} + \varepsilon \quad \text{con } \mathbf{A}(\mathbf{y}_{\text{true}})\,\mathbf{x}_{\text{true}} = \mathbf{b}_{\text{true}},$$

donde $\mathbf{x}_{\text{true}} \in \mathbb{R}^n$ es la imagen original que queremos recuperar, $\mathbf{b}_{\text{true}} \in \mathbb{R}^m$ denota el vector desconocido sin errores asociado con los datos disponibles y $\varepsilon \in \mathbb{R}^m$ es un vector desconocido que representa el ruido o errores en $\mathbf{b}$. La matriz $\mathbf{A}(\mathbf{y}) \in \mathbb{R}^{m\times n}$ con $m \geq n$ modela un operador directo y está típicamente muy mal condicionada. Esta situación puede deberse, por ejemplo, a que sus valores singulares se concentran en el origen, o bien a que la matriz presenta un rango numérico deficiente. Este último caso ocurre cuando, a pesar de tener rango completo en sentido teórico, la matriz posee valores singulares cuya magnitud es menor que la precisión de máquina ε, haciendo que en la práctica se los considere como si fuesen cero. En este trabajo asumimos que $\mathbf{A}$ es desconocida, pero puede parametrizarse mediante un vector $\mathbf{y} \in \mathbb{R}^r$ con $r \ll n$ de tal manera que la función $\mathbf{y} \mapsto \mathbf{A}(\mathbf{y})$ sea diferenciable. Estos problemas, donde las observaciones dependen no linealmente de los parámetros desconocidos $\mathbf{y}$, y linealmente de la solución buscada $\mathbf{x}$, se denominan problemas inversos no lineales separables. Nuestro objetivo es calcular buenas aproximaciones de $\mathbf{x}_{\text{true}}$ e $\mathbf{y}_{\text{true}}$, dado un vector de datos $\mathbf{b}$ y una matriz $\mathbf{A}$ de $m \times n$. Se obtiene entonces la siguiente reformulación del problema (1.2)

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{L}\mathbf{x}\|_2^2. \tag{1.3}$$

Asumimos que el problema (1.3) tiene solución $(\mathbf{x}^*, \mathbf{y}^*)$ en un conjunto abierto de $\mathbb{R}^{n+r}$. Además, asumimos que $\mathbf{L} \in \mathbb{R}^{q\times n}$ verifica que

$$\mathcal{N}(\mathbf{A}(\mathbf{y})) \cap \mathcal{N}(\mathbf{L}) = \{\mathbf{0}\}$$

para todos los valores de $\mathbf{y}$ en el dominio considerado, donde $\mathcal{N}(\mathbf{M})$ denota el núcleo de la matriz $\mathbf{M}$, de modo que el problema de minimización (1.3) tiene una solución única para $\mathbf{y}$ fijo [1].

Esta tesis se enfoca en utilizar y extender el método VarPro [8], el cual fue introducido para resolver problemas de mínimos cuadrados no lineales separables no regularizados, es decir, problemas de la forma

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2. \tag{1.4}$$

VarPro es un método eficiente cuya idea principal es eliminar la variable lineal $\mathbf{x}$, resolviendo un problema de mínimos cuadrados lineales para cada variable no lineal $\mathbf{y}$. Asumiendo que $\mathbf{A}(\mathbf{y})$ es de rango completo, entonces podemos escribir $\mathbf{x} = \mathbf{x}(\mathbf{y}) = \mathbf{A}(\mathbf{y})^\dagger\mathbf{b}$, donde $\mathbf{A}(\mathbf{y})^\dagger = \left(\mathbf{A}(\mathbf{y})^\top\mathbf{A}(\mathbf{y})\right)^{-1}\mathbf{A}(\mathbf{y})^\top$ es la pseudo-inversa de Moore-Penrose de $\mathbf{A}(\mathbf{y})$. El funcional a minimizar se reduce a un funcional que solamente depende de la variable $\mathbf{y}$, obteniendo así el problema de minimización

$$\min_{\mathbf{y}} \frac{1}{2}\|\mathbf{A}(\mathbf{y})\mathbf{A}(\mathbf{y})^\dagger\mathbf{b} - \mathbf{b}\|_2^2. \tag{1.5}$$

En [8] se demostró que este problema tiene la misma solución que (1.4). El problema de minimización reducido, el cual es un problema de mínimos cuadrados no lineales, se puede resolver mediante el método de Gauss-Newton (GN). Si para cada $\mathbf{y}$, escribimos $\mathcal{P}^\perp_{\mathbf{A}(\mathbf{y})} = \mathbf{I} - \mathbf{A}(\mathbf{y})\mathbf{A}(\mathbf{y})^\dagger$, el proyector ortogonal sobre el complemento ortogonal del espacio columna de $\mathbf{A}(\mathbf{y})$, entonces el problema reducido (1.5) se puede reescribir como $\min_{\mathbf{y}} \frac{1}{2}\| - \mathcal{P}^\perp_{\mathbf{A}(\mathbf{y})}\mathbf{b}\|_2^2$. Para resolverlo mediante el método de GN, en [8] se proporciona una expresión analítica de la derivada de Fréchet de $\mathcal{P}^\perp_{\mathbf{A}(\mathbf{y})}$ con respecto a la variable $\mathbf{y}$

$$\mathcal{D}\mathcal{P}^\perp_{\mathbf{A}(\mathbf{y})} = -\mathcal{P}^\perp_{\mathbf{A}(\mathbf{y})}\,\mathcal{D}(\mathbf{A}(\mathbf{y}))\,\mathbf{A}(\mathbf{y})^\dagger - \left(\mathcal{P}^\perp_{\mathbf{A}(\mathbf{y})}\,\mathcal{D}(\mathbf{A}(\mathbf{y}))\,\mathbf{A}(\mathbf{y})^\dagger\right)^\top. \tag{1.6}$$

En [15], Kaufman sugirió que el segundo término de (1.6) podría descartarse para acelerar el cálculo de la matriz Jacobiana si el residuo $\mathcal{P}^{\perp}_{\mathbf{A}(\mathbf{y})}\mathbf{b}$ es pequeño. Por lo tanto, la matriz Jacobiana podría aproximarse utilizando solo el primer término de la expresión (1.6). En [23], Ruano et al. propusieron otra expresión para la matriz Jacobiana, utilizando $\mathcal{D}(\mathbf{A}(\mathbf{y}))\mathbf{A}(\mathbf{y})^{\dagger}$, para la optimización de problemas no lineales separables que surgen en problemas de aprendizaje mediante redes neuronales. Un resumen del método VarPro, sus variantes y aplicaciones, se puede encontrar en [10], y aplicaciones más recientes de VarPro se pueden encontrar en [4, 18, 22]. En [10], Golub y Pereyra revisaron los desarrollos y aplicaciones de VarPro para resolver problemas de mínimos cuadrados no lineales separables con una combinación lineal de funciones no lineales como modelo subyacente. Para esta misma familia de problemas, O'Leary y Rust presentaron una implementación más robusta de VarPro y calcularon la matriz Jacobiana con mayor eficiencia, demostrando que se requieren menos iteraciones para converger si se considera la matriz Jacobiana completa [20]. Varios trabajos han demostrado computacionalmente y teóricamente que separar la variable lineal $\mathbf{x}$ de la variable no lineal $\mathbf{y}$, como propone VarPro, acelera la convergencia de los métodos iterativos para resolver (1.4) (véase [9, 15, 24] para un análisis más detallado).

Para problemas de la forma (1.3), elecciones típicas de $\mathbf{L}$ incluyen la matriz identidad, operadores discretos de primera o segunda derivada, o alguna otra transformada discreta, por ejemplo, transformada wavelet discreta o transformada discreta de Fourier (DFT, por sus siglas en inglés). Si $\mathbf{L} = \mathbf{I}$, decimos que (1.3) está en forma estándar. De lo contrario, decimos que está en forma general. En este último caso, el problema puede transformarse a la forma estándar [12].

El uso de VarPro para resolver problemas de mínimos cuadrados no lineales separables regularizados de la forma (1.3) fue introducido por primera vez en [3], para el caso donde $\mathbf{L}$ es la matriz identidad. Los autores presentaron un método eficiente que utiliza un enfoque híbrido de subespacio de Krylov para superar el alto costo computacional de resolver (1.3) para problemas inversos de gran escala y lo aplicaron a problemas de deconvolución semiciega. En [7], los autores modificaron el enfoque de [3] incorporando un método de Krylov inexacto para resolver el subproblema lineal.

En [6], VarPro se extendió para resolver (1.3) para matrices de regularización generales $\mathbf{L}$. Esta nueva versión se denominó GenVarPro. Ese trabajo también incluyó expresiones para calcular el Jacobiano y las aproximaciones dadas por Kaufman y por Ruano, Jones y Fleming, utilizando la descomposición en valores singulares generalizada y la descomposición espectral conjunta de operadores directos y de regularización cuando están disponibles o son factibles. Para problemas inversos a gran escala, se emplearon métodos iterativos basados en proyecciones y métodos generalizados del subespacio de Krylov para resolver los subproblemas lineales necesarios para aproximar Jacobianos y residuos. Se incluyeron ejemplos numéricos, en el contexto de problemas de imágenes a gran escala, como la corrección de desenfoque semiciega, que demostraron la eficacia de GenVarPro.

En [5] se desarrolló un método eficiente para resolver problemas inversos no lineales separables a gran escala con regularización de Tikhonov, donde la matriz del sistema es demasiado grande para almacenarse explícitamente y solo se pueden realizar multiplicaciones de matrices-vectores con $\mathbf{A}$ y posiblemente $\mathbf{A}^{\top}$. Aprovechando métodos iterativos como LSQR (ver Sección 2.1.2), que sólo requieren multiplicaciones matriz-vector, se propuso la versión Inexact-GenVarPro. Este método incorpora aproximaciones del Jacobiano y del residuo mediante soluciones iterativas de $\mathbf{x}$, manteniendo fijo el parámetro de regularización. Se asumió además que el tamaño r de $\mathbf{y} \in \mathbb{R}^r$ es pequeño. Se realizó un análisis riguroso de la convergencia local que valida la eficacia de esta aproximación, demostrando su aplicabilidad, por ejemplo, en problemas de deconvolución semiciega. En el trabajo se probó que partiendo de un punto inicial $\mathbf{y}_0$ cercano al mínimo $\mathbf{y}^*$ del problema original, se puede elegir una sucesión de tolerancias $\{\epsilon^{(k)}\}_{k\geq 1}$ como criterio de parada de LSQR, de manera que la sucesión $\{\mathbf{y}^{(k)}\}_{k\geq 1}$ generada por el algoritmo Inexact-GenVarPro converja a $\mathbf{y}^*$. Este avance mejora la viabilidad computacional para problemas de gran dimensión, diferenciándose de otros enfoques que requieren factorizaciones o ajustes iterativos del parámetro de regularización. El análisis realizado en [5] no se aplica a los métodos propuestos en [15] o [24], donde se utilizan diferentes Jacobianos aproximados; ni tampoco se aplica a la convergencia de los métodos de [3, 6, 7], ya que en estos métodos el parámetro de regularización se elige en cada iteración.

1.3. Objetivos

En esta tesis nos proponemos extender los método de proyección de variables GenVarPro e Inexact-GenVarPro para resolver problemas de deconvolución semiciega de imágenes desenfocadas, incorporando términos de regularización sobre los parámetros del operador de desenfoque. El modelo considerado es un problema de mínimos cuadrados no lineales separables con regularización de Tikhonov en forma general:

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2} \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda^2}{2} \|\mathbf{L}\mathbf{x}\|_2^2 + \mathcal{R}(\mathbf{y}), \tag{1.7}$$

donde $\mathcal{R}(\mathbf{y})$ representa un término regularizante sobre los parámetros del desenfoque. En particular, vamos a estudiar los casos $\mathcal{R}(\mathbf{y}) = \mu^2\|\mathbf{y}-\mathbf{y}_0\|_2^2$, con $\mathbf{y}_0$ un valor cercano a la solución $\mathbf{y}_{\text{true}}$ buscada, y $\mathcal{R}(\mathbf{y}) = -\sum \mu_j^2 \log(y_j)$, con $\mathbf{y} = (y_1, \ldots, y_r)$. Los valores μ y $(\mu_1, \ldots, \mu_r)$ son parámetros de regularización que se eligen en función del problema, y cuya finalidad es balancear los distintos términos en la expresión (1.7), evitando sesgos y asegurando que ninguna norma tenga un peso desproporcionado respecto de las demás. Esta extensión permitirá mejorar la precisión y estabilidad de las estimaciones del operador de desenfoque, especialmente en presencia de ruido y otros factores perturbadores. Al incorporar estos términos regularizantes, buscamos desarrollar un enfoque más robusto y efectivo para la restauración de imágenes desenfocadas, contribuyendo así al avance en las técnicas de procesamiento de imágenes y resolución de problemas inversos.

Vamos a desarrollar distintas variantes del método VarPro extendido utilizando la expresión del Jacobiano introducida por Golub-Pereyra [8] y las aproximaciones del Jacobiano introducidas por Kaufman [15] y Ruano et al. [23], y estudiar casos especiales en los que los operadores de desenfoque y regularización admiten una descomposición espectral conjunta [14], lo que permite implementar algoritmos altamente eficientes mediante transformadas rápidas de Fourier. Al identificar y utilizar estas características, podemos optimizar el procesamiento y mejorar significativamente la eficiencia computacional de los algoritmos de deconvolución. Para escenarios donde el cómputo exacto del Jacobiano resulta costoso, se propone una extensión de Inexact-GenVarPro [5].

Realizamos un análisis de convergencia local del método bajo hipótesis sobre los operadores involucrados, mostrando que, bajo ciertas condiciones, se puede alcanzar convergencia superlineal e incluso cuadrática (para una definición precisa de estos tipos de convergencia, véase la Sección 2.2). Finalmente, se presentan experimentos numéricos que validan los resultados teóricos, ilustrando la eficacia de los métodos propuestos en problemas reales de deconvolución semiciega. Estos experimentos demuestran que la incorporación de regularización en los parámetros del desenfoque mejora notablemente la calidad de las soluciones reconstruidas, el cálculo de los parámetros del operador de desenfoque, al tiempo que mantiene una eficiencia computacional adecuada.

1.4. Estructura de la tesis

Esta tesis se organiza de la siguiente manera. En el Capítulo 2 se presenta el desarrollo teórico del trabajo: se introduce una extensión del método de proyección de variables existente, se establece un resultado de convergencia local y se definen los términos de regularización que se utilizarán en los experimentos numéricos. El Capítulo 3 está dedicado a la representación y la implementación eficiente de matrices que surgen en problemas de desenfoque de imágenes, destacando estructuras que permiten acelerar los cálculos. En el Capítulo 4 se llevan a cabo experimentos numéricos que permiten validar los resultados teóricos y evaluar la efectividad de las distintas variantes del método propuestas en el Capítulo 2. Finalmente, el Capítulo 5 presenta una síntesis de los resultados obtenidos y propone posibles líneas de investigación futura orientadas a continuar el desarrollo y la aplicación de los métodos estudiados.

2. DESARROLLO TEÓRICO

2.1. Método de proyección de variables para problemas regularizados

Queremos resolver problemas de mínimos cuadrados no lineales separables regularizados de la forma

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2} \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda^2}{2} \|\mathbf{L}\mathbf{x}\|_2^2 + \mathcal{R}(\mathbf{y}), \tag{2.1}$$

donde $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A}(\mathbf{y}) \in \mathbb{R}^{m\times n}$ con $m \geq n$, $\mathbf{y} \in \mathbb{R}^r$ con $r \ll n$ y $\mathbf{L} \in \mathbb{R}^{q\times n}$. Asumimos que el problema tiene solución $(\mathbf{x}^*, \mathbf{y}^*)$ en un conjunto abierto de $\mathbb{R}^{n+r}$.

La idea principal detrás de VarPro [8] consiste en eliminar la variable $\mathbf{x}$ de la formulación del problema original y proporcionar un funcional reducido para minimizar solo con respecto a $\mathbf{y}$. Es decir, para resolver el problema (2.1), para cada $\mathbf{y}$ en un conjunto abierto $\Omega \subseteq \mathbb{R}^r$ que contiene la solución $\mathbf{y}^*$, obtenemos la solución $\mathbf{x}(\mathbf{y})$ del problema de minimización

$$\min_{\mathbf{x}} \mathcal{F}(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 + \mathcal{R}(\mathbf{y}), \tag{2.2}$$

y luego minimizamos el funcional $\varphi(\mathbf{y}) = \mathcal{F}(\mathbf{x}(\mathbf{y}), \mathbf{y})$.

Bajo la hipótesis $\mathcal{N}(\mathbf{A}(\mathbf{y})) \cap \mathcal{N}(\mathbf{L}) = \{\mathbf{0}\}$, la matriz $\begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix}$ tiene rango máximo, y entonces para un $\mathbf{y}$ fijo

$$\mathbf{x}(\mathbf{y}) = \arg\min_{\mathbf{x}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 + \mathcal{R}(\mathbf{y}).$$

Como el término $\mathcal{R}(\mathbf{y})$ no depende de $\mathbf{x}$, no influye en la minimización respecto a $\mathbf{x}$, y, por lo tanto, puede omitirse en la expresión para calcular el argumento mínimo. Luego

$$\mathbf{x}(\mathbf{y}) = \arg\min_{\mathbf{x}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \tag{2.3}$$

y obtenemos una solución cerrada explícita de la forma

$$\mathbf{x}(\mathbf{y}) = \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix},$$

que puede ser utilizada para reescribir el problema no lineal con respecto a la variable $\mathbf{y}$, obteniendo el problema de minimización reducido

$$\min_{\mathbf{y}} \varphi(\mathbf{y}) = \min_{\mathbf{y}} \frac{1}{2} \|\mathbf{f}(\mathbf{y})\|_2^2 + \mathcal{R}(\mathbf{y}), \tag{2.4}$$

donde $\mathbf{f} : \mathbb{R}^r \to \mathbb{R}^{m+q}$ se define como

$$\begin{aligned} \mathbf{f}(\mathbf{y}) &= \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix} \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \\ &= \left(\begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix} \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix}^\dagger - \mathbf{I} \right) \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}. \end{aligned} \tag{2.5}$$

Para simplificar notación definimos

$$\mathbf{K}(\mathbf{y}) = \begin{bmatrix} \mathbf{A}(\mathbf{y}) \\ \lambda\mathbf{L} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \tag{2.6}$$

y

$$\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})} = \mathbf{I} - \mathbf{K}(\mathbf{y})\mathbf{K}(\mathbf{y})^{\dagger}.$$

Luego tenemos que $\mathbf{f}(\mathbf{y}) = -\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d}$.

El siguiente resultado muestra que la formulación que obtenemos a partir de usar el método de proyección de variables nos permite resolver el problema de minimización original. La demostración se basa en la presentada en [8] para el caso no regularizado, adaptada aquí al contexto con regularización en $\mathbf{y}$.

Teorema 1. *Supongamos que $\mathcal{N}(\mathbf{A}(\mathbf{y})) \cap \mathcal{N}(\mathbf{L}) = \{\mathbf{0}\}$ para todo $\mathbf{y}$ en un dominio abierto $\Omega \subset \mathbb{R}^r$ y que el operador $\mathbf{y} \mapsto \mathbf{A}(\mathbf{y})$ es diferenciable en Ω. Sean $\mathcal{F}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\ \|\mathbf{K}(\mathbf{y})\mathbf{x} - \mathbf{d}\|_2^2 + \mathcal{R}(\mathbf{y})$ y $\varphi(\mathbf{y}) = \frac{1}{2}\ \|\mathbf{f}(\mathbf{y})\|_2^2 + \mathcal{R}(\mathbf{y})$. Entonces valen los siguientes resultados:*

(1) Si $(\mathbf{x}^*, \mathbf{y}^*)$ es un mínimo global de $\mathcal{F}(\mathbf{x}, \mathbf{y})$ en $\mathbb{R}^n \times \Omega$, entonces $\mathbf{x}^* = \mathbf{K}(\mathbf{y}^*)^{\dagger}\mathbf{d}$ e $\mathbf{y}^*$ es un mínimo global de $\varphi(\mathbf{y})$ en Ω.

(2) Si $\mathbf{y}^*$ es un punto crítico (resp. un mínimo global) de $\varphi(\mathbf{y})$ en Ω y $\mathbf{x}^* = \mathbf{K}(\mathbf{y}^*)^{\dagger}\mathbf{d}$, entonces $(\mathbf{x}^*, \mathbf{y}^*)$ es un punto crítico (resp. un mínimo global) de $\mathcal{F}(\mathbf{x}, \mathbf{y})$ en $\mathbb{R}^n \times \Omega$.

Demostración. Para todo $\mathbf{y} \in \Omega$, como vale $\mathcal{N}(\mathbf{A}(\mathbf{y})) \cap \mathcal{N}(\mathbf{L}) = \{\mathbf{0}\}$, tenemos que

$$\mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d} = \arg\min_{\mathbf{x}} \|\mathbf{K}(\mathbf{y})\mathbf{x} - \mathbf{d}\|_2^2.$$

Esto implica que, para $\mathbf{y} \in \Omega$, vale la siguiente desigualdad para todo $\mathbf{x} \in \mathbb{R}^n$:

$$\varphi(\mathbf{y}) = \frac{1}{2}\|\mathbf{K}(\mathbf{y})\mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d} - \mathbf{d}\|_2^2 + \mathcal{R}(\mathbf{y}) \leq \frac{1}{2}\|\mathbf{K}(\mathbf{y})\mathbf{x} - \mathbf{d}\|_2^2 + \mathcal{R}(\mathbf{y}) = \mathcal{F}(\mathbf{x}, \mathbf{y}), \tag{2.7}$$

y la desigualdad es estricta para $\mathbf{x} \neq \mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d}$.

(1) Supongamos que $(\mathbf{x}^*, \mathbf{y}^*)$ es un mínimo global de $\mathcal{F}(\mathbf{x}, \mathbf{y})$ en $\mathbb{R}^n \times \Omega$. Por la desigualdad de recién, $\varphi(\mathbf{y}^*) \leq \mathcal{F}(\mathbf{x}^*, \mathbf{y}^*)$. Ahora, si $\mathbf{x}^{\diamond} = \mathbf{K}(\mathbf{y}^*)^{\dagger}\mathbf{d}$, de la definición de φ y como $(\mathbf{x}^*, \mathbf{y}^*)$ es un mínimo global de $\mathcal{F}(\mathbf{x}, \mathbf{y})$, tenemos que $\varphi(\mathbf{y}^*) = \mathcal{F}(\mathbf{x}^{\diamond}, \mathbf{y}^*) \geq \mathcal{F}(\mathbf{x}^*, \mathbf{y}^*)$. Entonces, vale la igualdad $\varphi(\mathbf{y}^*) = \mathcal{F}(\mathbf{x}^*, \mathbf{y}^*)$, y en consecuencia, $\mathbf{x}^* = \mathbf{K}(\mathbf{y}^*)^{\dagger}\mathbf{d}$.

Luego, para cada $\mathbf{y}_0 \in \Omega$, si $\mathbf{x}_0 = \mathbf{K}(\mathbf{y}_0)^{\dagger}\mathbf{d}$, tenemos que $\varphi(\mathbf{y}_0) = \mathcal{F}(\mathbf{x}_0, \mathbf{y}_0) \geq \mathcal{F}(\mathbf{x}^*, \mathbf{y}^*) = \varphi(\mathbf{y}^*)$, lo que implica que $\mathbf{y}^*$ es un mínimo global de $\varphi(\mathbf{y})$ en Ω.

(2) Sea $\mathbf{y}^* \in \Omega$ un punto crítico de $\varphi(\mathbf{y})$ y $\mathbf{x}^* = \mathbf{K}(\mathbf{y}^*)^{\dagger}\mathbf{d}$. Para poder mostrar que $(\mathbf{x}^*, \mathbf{y}^*)$ es un punto crítico de $\mathcal{F}$, debemos calcular el gradiente $\nabla\mathcal{F}(\mathbf{x}^*, \mathbf{y}^*)$. Primero notar que,

$$\nabla_{\mathbf{x}}\mathcal{F}(\mathbf{x}^*, \mathbf{y}^*) = \mathbf{K}(\mathbf{y}^*)^{\top}(\mathbf{K}(\mathbf{y}^*)\mathbf{x}^* - \mathbf{d}) = -\mathbf{K}(\mathbf{y}^*)^{\top}\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y}^*)}\mathbf{d} = 0.$$

Ahora, para $j = 1, \ldots, r$,

$$\frac{\partial\mathcal{F}}{\partial\mathbf{y}_j}(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial\mathbf{K}(\mathbf{y})}{\partial\mathbf{y}_j}\mathbf{x}\right)^{\top}(\mathbf{K}(\mathbf{y})\mathbf{x} - \mathbf{d}) + \frac{\partial\mathcal{R}}{\partial\mathbf{y}_j}(\mathbf{y}).$$

Usando que $\mathbf{x}^* = \mathbf{K}(\mathbf{y}^*)^{\dagger}\mathbf{d}$, obtenemos

$$\frac{\partial\mathcal{F}}{\partial\mathbf{y}_j}(\mathbf{x}^*, \mathbf{y}^*) = (\mathbf{K}(\mathbf{y}^*)^{\dagger}\mathbf{d})^{\top}\left(\frac{\partial\mathbf{K}}{\partial\mathbf{y}_j}(\mathbf{y}^*)\right)^{\top}(-\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y}*)}\mathbf{d}) + \frac{\partial\mathcal{R}}{\partial\mathbf{y}_j}(\mathbf{y}^*). \tag{2.8}$$

Por otro lado,

$$\frac{\partial\varphi(\mathbf{y})}{\partial\mathbf{y}_j} = \left(\frac{\partial\mathbf{f}(\mathbf{y})}{\partial\mathbf{y}_j}\right)^{\top}\mathbf{f}(\mathbf{y}) + \frac{\partial\mathcal{R}(\mathbf{y})}{\partial\mathbf{y}_j}.$$

La j-ésima columna de $\mathbf{J_f} : \mathbb{R}^r \to \mathbb{R}^{(m+q)\times r}$, la matriz Jacobiana de $\mathbf{f}$, se puede calcular de la siguiente manera [6]:

$$\begin{aligned}\frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{y}_j} &= \frac{\partial}{\partial \mathbf{y}_j}(-\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d}) = \frac{\partial}{\partial \mathbf{y}_j}\left(\mathbf{K}(\mathbf{y})\mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d}\right) = \frac{\partial}{\partial \mathbf{y}_j}\left(\mathbf{K}(\mathbf{y})\mathbf{K}(\mathbf{y})^{\dagger}\right)\mathbf{d} \\ &= \left(\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger} + \mathbf{K}(\mathbf{y})\frac{\partial \mathbf{K}(\mathbf{y})^{\dagger}}{\partial \mathbf{y}_j}\right)\mathbf{d}.\end{aligned}$$

Escribiendo $\mathbf{K}(\mathbf{y})^{\dagger} = (\mathbf{K}(\mathbf{y})^{\top}\mathbf{K}(\mathbf{y}))^{-1}\mathbf{K}(\mathbf{y})^{\top}$, aplicando la regla del producto, y usando que para una matriz inversible $\mathbf{M}(\mathbf{y})$ vale $\dfrac{\partial \mathbf{M}(\mathbf{y})^{-1}}{\partial \mathbf{y}_j} = -\mathbf{M}(\mathbf{y})^{-1}\dfrac{\partial \mathbf{M}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{M}(\mathbf{y})^{-1}$, obtenemos que

$$\frac{\partial \mathbf{K}(\mathbf{y})^{\dagger}}{\partial \mathbf{y}_j} = (\mathbf{K}(\mathbf{y})^{\top}\mathbf{K}(\mathbf{y}))^{-1}\frac{\partial \mathbf{K}(\mathbf{y})^{\top}}{\partial \mathbf{y}_j}\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})} - \mathbf{K}(\mathbf{y})^{\dagger}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}.$$

Luego, siguiendo [6] obtenemos que la j-ésima columna de $\mathbf{J_f}$ está dada por

$$\begin{aligned}\frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{y}_j} &= \left(\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger} + \mathbf{K}(\mathbf{y})\left((\mathbf{K}(\mathbf{y})^{\top}\mathbf{K}(\mathbf{y}))^{-1}\frac{\partial \mathbf{K}(\mathbf{y})^{\top}}{\partial \mathbf{y}_j}\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})} - \mathbf{K}(\mathbf{y})^{\dagger}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}\right)\right)\mathbf{d} \\ &= \left(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger} + \left(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}\right)^{\top}\right)\mathbf{d} \\ &= \mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d} + \left(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}\right)^{\top}\mathbf{d}. \qquad (2.9)\end{aligned}$$

Entonces,

$$\begin{aligned}\frac{\partial \varphi(\mathbf{y})}{\partial \mathbf{y}_j} &= \mathbf{d}^{\top}\left(\left(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}\right)^{\top} + \mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}\right)(-\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})})\mathbf{d} + \frac{\partial \mathcal{R}(\mathbf{y})}{\partial \mathbf{y}_j} \\ &= -\mathbf{d}^{\top}\left(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\mathbf{K}(\mathbf{y})^{\dagger}\right)^{\top}\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d} + \frac{\partial \mathcal{R}(\mathbf{y})}{\partial \mathbf{y}_j}\end{aligned}$$

ya que $\mathbf{K}(\mathbf{y})^{\dagger}\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})} = 0$. Distribuyendo la transpuesta en el primer término, y usando que $(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})})^{\top} = \mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}$ y $(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})})^2 = \mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}$ obtenemos que,

$$\begin{aligned}\frac{\partial \varphi(\mathbf{y})}{\partial \mathbf{y}_j} &= -\mathbf{d}^{\top}(\mathbf{K}(\mathbf{y})^{\dagger})^{\top}\left(\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\right)^{\top}(\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})})^{\top}\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d} + \frac{\partial \mathcal{R}(\mathbf{y})}{\partial \mathbf{y}_j} \\ &= -(\mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d})^{\top}\left(\frac{\partial \mathbf{K}(\mathbf{y})}{\partial \mathbf{y}_j}\right)^{\top}\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d} + \frac{\partial \mathcal{R}(\mathbf{y})}{\partial \mathbf{y}_j}. \qquad (2.10)\end{aligned}$$

De (2.8) y (2.10), como $\mathbf{y}^*$ es un punto crítico de $\varphi(\mathbf{y})$, se sigue que

$$\frac{\partial \mathcal{F}}{\partial \mathbf{y}_j}(\mathbf{x}^*, \mathbf{y}^*) = \frac{\partial \varphi}{\partial \mathbf{y}_j}(\mathbf{y}^*) = 0.$$

Concluimos que $\nabla \mathcal{F}(\mathbf{x}^*, \mathbf{y}^*) = 0$, esto es, $(\mathbf{x}^*, \mathbf{y}^*)$ es un punto crítico de $\mathcal{F}$.

Si $\mathbf{y}^*$ es un mínimo global de $\varphi(\mathbf{y})$ en Ω, para $(\mathbf{x}_0, \mathbf{y}_0) \in \mathbb{R}^n \times \Omega$, usando la desigualdad (2.7) deducimos que $\mathcal{F}(\mathbf{x}_0, \mathbf{y}_0) \geq \varphi(\mathbf{y}_0) \geq \varphi(\mathbf{y}^*) = \mathcal{F}(\mathbf{x}^*, \mathbf{y}^*)$. Entonces $(\mathbf{x}^*, \mathbf{y}^*)$ es un mínimo global de $\mathcal{F}(\mathbf{x}, \mathbf{y})$ en $\mathbb{R}^n \times \Omega$. ■

2.1.1. GenVarPro Regularizado

Para resolver (2.4) vamos a desarrollar un método *quasi-Newton* [19]. Estos métodos iterativos ofrecen una alternativa eficiente al método de Newton al evitar el cálculo explícito del Hessiano. Para minimizar la función $\varphi(\mathbf{y}) = \frac{1}{2}\|\mathbf{f}(\mathbf{y})\|_2^2 + \mathcal{R}(\mathbf{y})$ proponemos, en lugar del cálculo explícito del Hessiano $\nabla^2\varphi(\mathbf{y}^{(k)})$ en cada iteración k, construir una matriz $\mathbf{H}(\mathbf{y}^{(k)})$ que lo aproxima. Luego, tenemos que las iteraciones del método *quasi-Newton* están determinadas por

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \mathbf{s}^{(k)},\ k = 0, 1, 2, ...,$$

donde $\mathbf{s}^{(k)}$ es solución de

$$\Big(\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)}) + \nabla^2\mathcal{R}(\mathbf{y}^{(k)})\Big)\mathbf{s}^{(k)} = -\Big(\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{f}(\mathbf{y}^{(k)}) + \nabla\mathcal{R}(\mathbf{y}^{(k)})\Big), \tag{2.11}$$

con

$$\nabla\varphi(\mathbf{y}) = \mathbf{J}_\mathbf{f}^\top(\mathbf{y})\mathbf{f}(\mathbf{y}) + \nabla\mathcal{R}(\mathbf{y}),$$

y en el que usamos la siguiente aproximación del Hessiano $\nabla^2\varphi(\mathbf{y})$:

$$\nabla^2\varphi(\mathbf{y}) \simeq \mathbf{J}_\mathbf{f}^\top(\mathbf{y})\mathbf{J}_\mathbf{f}(\mathbf{y}) + \nabla^2\mathcal{R}(\mathbf{y}) := \mathbf{H}(\mathbf{y}). \tag{2.12}$$

En la demostración del Teorema 1 calculamos $\mathbf{J}_\mathbf{f} : \mathbb{R}^r \to \mathbb{R}^{(m+q)\times r}$, la matriz Jacobiana de $\mathbf{f}$ (2.9). La j-ésima columna de $\mathbf{J}_\mathbf{f}$ está dada entonces por

$$\begin{aligned}\frac{\partial\mathbf{f}(\mathbf{y})}{\partial\mathbf{y}_j} &= \mathcal{P}^\perp_{\mathbf{K}(\mathbf{y})}\frac{\partial\mathbf{K}(\mathbf{y})}{\partial\mathbf{y}_j}\mathbf{K}(\mathbf{y})^\dagger\mathbf{d} + \left(\mathcal{P}^\perp_{\mathbf{K}(\mathbf{y})}\frac{\partial\mathbf{K}(\mathbf{y})}{\partial\mathbf{y}_j}\mathbf{K}(\mathbf{y})^\dagger\right)^\top\mathbf{d} \\ &=: \mathbf{A}_j + \mathbf{B}_j.\end{aligned} \tag{2.13}$$

Usando que $(\mathcal{P}^\perp_{\mathbf{K}(\mathbf{y})})^\top = \mathcal{P}^\perp_{\mathbf{K}(\mathbf{y})}$ y $\mathcal{P}^\perp_{\mathbf{K}(\mathbf{y})}\mathbf{d} = -\mathbf{f}(\mathbf{y})$ obtenemos que

$$\mathbf{B}_j = \left(\mathcal{P}^\perp_{\mathbf{K}(\mathbf{y})}\frac{\partial\mathbf{K}(\mathbf{y})}{\partial\mathbf{y}_j}\mathbf{K}(\mathbf{y})^\dagger\right)^\top\mathbf{d} = -\left(\mathbf{K}(\mathbf{y})^\dagger\right)^\top\frac{\partial\mathbf{K}(\mathbf{y})^\top}{\partial\mathbf{y}_j}\mathbf{f}(\mathbf{y}). \tag{2.14}$$

En [15], se sugiere ignorar el segundo término $\mathbf{B}_j$ de (2.13) cuando el residuo $\mathbf{f}(\mathbf{y})$ es pequeño. Es decir, podemos aproximar el Jacobiano utilizando

$$\frac{\partial\mathbf{f}(\mathbf{y})}{\partial\mathbf{y}_j} \approx \mathbf{A}_j.$$

Esto se justifica porque el término $\mathbf{B}_j$ es proporcional al residuo, como se ve en (2.14), por lo que su contribución disminuye al mejorar el ajuste. Además, omitirlo reduce significativamente el costo computacional sin afectar de manera importante la precisión o la convergencia del método.

Denotamos por GP (por Golub-Pereyra) el Jacobiano que contiene ambos términos y por KAUF (de Kaufman), el Jacobiano que incluye solo el primer término en (2.13). Trabajos previos han realizado comparaciones del rendimiento de los Jacobianos GP y KAUF para diferentes aplicaciones, por ejemplo en [17, 20]. En los ejemplos numéricos del Capítulo 4, investigamos la convergencia de los métodos propuestos al utilizar ambos Jacobianos en el contexto de la deconvolución semiciega. Además, también probamos la aproximación de la matriz Jacobiana propuesta por Ruano et al. [23], que se define mediante:

$$\frac{\partial\mathbf{f}(\mathbf{y})}{\partial\mathbf{y}_j} \approx \frac{\partial\mathbf{K}(\mathbf{y})}{\partial\mathbf{y}_j}\mathbf{K}(\mathbf{y})^\dagger\mathbf{d}. \tag{2.15}$$

Nos referimos a esta matriz Jacobiana aproximada como el Jacobianao RJF (de Ruano-Jones-Fleming).

En [6] se ve que los tres Jacobianos dan el mismo gradiente de la función $\psi(\mathbf{y}) = \frac{1}{2}\|\mathbf{f}(\mathbf{y})\|_2^2$, $\nabla\psi(\mathbf{y}) = \mathbf{J}_\mathbf{f}^\top(\mathbf{y})\mathbf{f}(\mathbf{y})$. Es trivial notar que la misma prueba vale para $\varphi(\mathbf{y})$ de nuestro problema de mínimos cuadrados no lineales separables extendido, donde $\nabla\varphi(\mathbf{y}) = \mathbf{J}_\mathbf{f}^\top(\mathbf{y})\mathbf{f}(\mathbf{y}) + \nabla\mathcal{R}(\mathbf{y})$. Para simplificar la notación, denotamos $\mathbf{K}(\mathbf{y}) = \mathbf{K}$ y $\mathcal{P}_\mathbf{K}^\perp(\mathbf{y}) = \mathcal{P}_\mathbf{K}^\perp$. Sean

$$\begin{aligned}\mathbf{J}_{\mathrm{GP}} &= \mathcal{P}_\mathbf{K}^\perp\mathcal{D}(\mathbf{K})\mathbf{K}^\dagger\mathbf{d}^\top + \left(\mathcal{P}_\mathbf{K}^\perp\mathcal{D}(\mathbf{K})\mathbf{K}^\dagger\right)^\top\mathbf{d}^\top,\\ \mathbf{J}_{\mathrm{KAUF}} &= \mathcal{P}_\mathbf{K}^\perp\mathcal{D}(\mathbf{K})\mathbf{K}^\dagger\mathbf{d}^\top,\\ \mathbf{J}_{\mathrm{RJF}} &= \mathcal{D}(\mathbf{K})\mathbf{K}^\dagger\mathbf{d}^\top,\end{aligned}$$

donde $\mathcal{D}(\mathbf{K})$ es la derivada Fréchet de $\mathbf{K}$ y $\mathbf{d}^\top = \begin{bmatrix}\mathbf{b}^\top, & \mathbf{0}\end{bmatrix}^\top$. Luego, si vale que

$$\mathbf{J}_{\mathrm{GP}}^\top\mathbf{f}(\mathbf{y}) = \mathbf{J}_{\mathrm{KAUF}}^\top\mathbf{f}(\mathbf{y}) = \mathbf{J}_{\mathrm{RJF}}^\top\mathbf{f}(\mathbf{y}),$$

entonces

$$\mathbf{J}_{\mathrm{GP}}^\top\mathbf{f}(\mathbf{y}) + \nabla\mathcal{R}(\mathbf{y}) = \mathbf{J}_{\mathrm{KAUF}}^\top\mathbf{f}(\mathbf{y}) + \nabla\mathcal{R}(\mathbf{y}) = \mathbf{J}_{\mathrm{RJF}}^\top\mathbf{f}(\mathbf{y}) + \nabla\mathcal{R}(\mathbf{y}).$$

Aunque los tres Jacobianos dan el mismo gradiente, dan diferentes aproximaciones del Hessiano. Por lo tanto, la solución $\mathbf{s}^{(k)}$ de (2.11) va a tener el mismo lado derecho, pero una aproximación del Hessiano distinta dependiendo de la elección del Jacobiano.

Definimos entonces un nuevo algoritmo llamado GenVarPro Regularizado que es una extensión del algoritmo del método de VarPro [6].

Algoritmo 1: Algoritmo GenVarPro Regularizado

1: Input: Operador $\mathbf{A} : \mathbf{y} \mapsto \mathbf{A}(\mathbf{y})$, vector $\mathbf{b}$, punto inicial $\mathbf{y}^{(0)}$, matriz de regularizacion $\mathbf{L}$, función regularizante $\mathcal{R}(\mathbf{y})$
2: **for** $k = 0, 1, \ldots$ *hasta que se cumpla el criterio de parada* **do**
3: $\quad\mathbf{x}^{(k)} = \mathbf{K}(\mathbf{y}^{(k)})^\dagger\mathbf{d} = \left(\mathbf{A}(\mathbf{y}^{(k)})^\top\mathbf{A}(\mathbf{y}^{(k)}) + \lambda^2\mathbf{L}^\top\mathbf{L}\right)^{-1}\mathbf{A}(\mathbf{y}^{(k)})^\top\mathbf{b}$
4: $\quad\mathbf{f}^{(k)} = \begin{bmatrix}\mathbf{A}(\mathbf{y}^{(\mathbf{k})})\\ \lambda\mathbf{L}\end{bmatrix}\mathbf{x}^{(k)} - \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix}$
5: $\quad$Calcular la matriz Jacobiana $\mathbf{J}_\mathbf{f}^{(k)} = \mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})$
6: $\quad$Calcular $\mathbf{s}^{(k)}$ como solución de
$\quad\left((\mathbf{J}_\mathbf{f}^{(k)})^\top\mathbf{J}_\mathbf{f}^{(k)} + \nabla^2\mathcal{R}(\mathbf{y}^{(k)})\right)\mathbf{s}^{(k)} = -\left((\mathbf{J}_\mathbf{f}^{(k)})^\top\mathbf{f}^{(k)} + \nabla\mathcal{R}(\mathbf{y}^{(k)})\right)$
7: $\quad\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \mathbf{s}^{(k)}$
8: **end**

2.1.2. GenVarPro Regularizado Inexacto

En esta sección nos basamos en [5] para extender el algoritmo Inexact-GenVarPro al caso de problemas de mínimos cuadrados no lineales separables de la forma (2.1). Recordando que podemos calcular a $\mathbf{x} = \mathbf{x}(\mathbf{y})$ (ver (2.3)) de forma exacta como $\mathbf{x}(\mathbf{y}) = \mathbf{K}(\mathbf{y})^\dagger\mathbf{d}$, podemos entonces expresar las columnas del Jacobiano dado por (2.13) como

$$\frac{\partial\mathbf{f}(\mathbf{y})}{\partial\mathbf{y}_j} = \mathcal{P}_{\mathbf{K}(\mathbf{y})}^\perp\frac{\partial\mathbf{K}(\mathbf{y})}{\partial\mathbf{y}_j}\mathbf{x}(\mathbf{y}) + (\mathbf{K}(\mathbf{y})^\dagger)^\top\left(\frac{\partial\mathbf{K}(\mathbf{y})}{\partial\mathbf{y}_j}\right)^\top(\mathbf{b} - \mathbf{A}\mathbf{x}(\mathbf{y})). \tag{2.16}$$

Aunque hemos reformulado el problema de minimización original (2.1) solo en términos de $\mathbf{y}$, el valor de $\mathbf{x} = \mathbf{x}(\mathbf{y})$ todavía aparece en el Jacobiano (2.16) que usamos para resolver el problema de minimización reducido (2.4). Además, el residuo

$$\mathbf{f}(\mathbf{y}) = \mathbf{K}(\mathbf{y})\mathbf{x}(\mathbf{y}) - \mathbf{d}$$

también depende de $\mathbf{x}(\mathbf{y})$. Para problemas de gran tamaño, calcular estos valores es computacionalmente costoso. Una posible forma de reducir el costo computacional del algoritmo consiste en aproximar la solución exacta $\mathbf{x}(\mathbf{y})$ de (2.2) aplicando un método iterativo. En este caso, adoptamos este enfoque incorporando el método iterativo LSQR [21]. Más precisamente, proponemos aproximar el Jacobiano $\mathbf{J_f}(\mathbf{y}^{(k)})$ en la k-ésima iteración del algoritmo mediante la matriz $\overline{\mathbf{J}}^{(k)}$, cuyas columnas están definidas por

$$[\overline{\mathbf{J}}^{(k)}]_j = \mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y}^{(k)})} \frac{\partial \mathbf{K}(\mathbf{y}^{(k)})}{\partial \mathbf{y}_j} \overline{\mathbf{x}}^{(k)} + \left(\mathbf{K}(\mathbf{y}^{(k)})^{\dagger}\right)^{\top} \left(\frac{\partial \mathbf{K}(\mathbf{y}^{(k)})}{\partial \mathbf{y}_j}\right)^{\top} (\mathbf{b} - \mathbf{A}(\mathbf{y}^{(k)})\overline{\mathbf{x}}^{(k)}), \tag{2.17}$$

para $j = 1, ..., r$, y el residuo por

$$\mathbf{g}^{(k)} = \mathbf{K}(\mathbf{y}^{(k)})\overline{\mathbf{x}}^{(k)} - \mathbf{d},$$

donde $\overline{\mathbf{x}}^{(k)}$ es una solución aproximada del subproblema lineal

$$\min_{\mathbf{x}} \mathcal{F}(\mathbf{x}, \mathbf{y}^{(\mathbf{k})}) = \min_{\mathbf{x}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}^{(\mathbf{k})}) \\ \lambda \mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 + \mathcal{R}(\mathbf{y}^{(\mathbf{k})}).$$

Como la minimización no depende de $\mathbf{y}^{(k)}$, entonces descartamos el término $\mathcal{R}(\mathbf{y}^{(\mathbf{k})})$ y obtenemos que $\overline{\mathbf{x}}^{(k)}$ es una aproximación de

$$\arg\min_{\mathbf{x}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}^{(\mathbf{k})}) \\ \lambda \mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2.$$

El algoritmo LSQR que utilizamos para calcular $\overline{\mathbf{x}}^{(k)}$ funciona con criterios de parada que dependen de tolerancias `atol`, `btol` y `conlim` [21], que especificamos para cada experimento en el Capítulo 4. Estas tolerancias cumplen,

1: Parar si $\|\mathbf{A}\overline{\mathbf{x}}^{(k)} - \mathbf{b}\| \leq \texttt{atol}\|\mathbf{A}\|\|\overline{\mathbf{x}}^{(k)}\| + \texttt{btol}\|\mathbf{b}\|$.

2: Parar si $\frac{\|\mathbf{A}^{\top}(\mathbf{A}\overline{\mathbf{x}}^{(k)} - \mathbf{b})\|}{\|\mathbf{A}\|\|\mathbf{A}\overline{\mathbf{x}}^{(k)} - \mathbf{b}\|} \leq \texttt{atol}$.

3: Parar si $\text{cond}(\mathbf{A}) \geq \texttt{conlim}$.

Llamamos al nuevo algoritmo GenVarPro Regularizado Inexacto. El siguiente algoritmo resume el método. En el paso 3 del Algoritmo 2, el método LSQR podría reemplazarse por cualquier método iterativo siempre que se utilice un criterio de parada análogo.

Algoritmo 2: Algoritmo GenVarPro Regularizado Inexacto

1: Input: Operador $\mathbf{A} : \mathbf{y} \mapsto \mathbf{A}(\mathbf{y})$, vector $\mathbf{b}$, punto inicial $\mathbf{y}^{(0)}$, matriz de regularizacion $\mathbf{L}$, función regularizante $\mathcal{R}(\mathbf{y})$

2: **for** $k = 0, 1, \ldots$ *hasta que se cumpla el criterio de parada* **do**

3: Calcular $\overline{\mathbf{x}}^{(k)}$ aplicando el algoritmo LSQR al problema
$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}(\mathbf{y}^{(k)})\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda^2}{2} \|\mathbf{L}\mathbf{x}\|_2^2$

4: $\mathbf{g}^{(k)} = \begin{bmatrix} \mathbf{A}(\mathbf{y}^{(k)}) \\ \lambda \mathbf{L} \end{bmatrix} \overline{\mathbf{x}}^{(k)} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$

5: Calcular la matriz Jacobiana $\overline{\mathbf{J}}^{(k)}$ de acuerdo a (2.17)

6: Calcular $\mathbf{s}^{(k)}$ como solución de
$\left((\overline{\mathbf{J}}^{(k)})^{\top}\overline{\mathbf{J}}^{(k)} + \nabla^2\mathcal{R}(\mathbf{y}^{(k)})\right)\mathbf{s}^{(k)} = -\left((\overline{\mathbf{J}}^{(k)})^{\top}\mathbf{g}^{(k)} + \nabla\mathcal{R}(\mathbf{y}^{(k)})\right)$

7: $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \mathbf{s}^{(k)}$

8: **end**

2.2. Convergencia Local

Extendemos los resultados de convergencia local presentados en [2] al caso particular de problemas de mínimos cuadrados no lineales separables de la forma (2.1), bajo el supuesto de que el término de regularización $\mathcal{R}(\mathbf{y})$ es convexo y dos veces diferenciable y que la solución $\mathbf{x}$ del subproblema lineal se obtiene de forma exacta, como se describe en la Sección 2.1.1.

Antes de iniciar el análisis, introducimos la notación que utilizaremos a lo largo de esta sección. Denotamos por $\mathbf{y}^*$ una solución del problema de minimización (2.4), y por $\mathbf{y}$ un punto factible cualquiera. Sea $\{\mathbf{y}^{(k)}\}_{k\geq 0}$ la sucesión generada por el Algoritmo 1, y $\mathcal{B}(\mathbf{y}^*, \rho)$ la bola abierta de radio ρ centrada en $\mathbf{y}^*$. Definimos el vector de error como $\mathbf{e} = \mathbf{y} - \mathbf{y}^*$, y el error en la k-ésima iteración como $\mathbf{e}_k = \mathbf{y}^{(k)} - \mathbf{y}^*$. Además, recordamos la notación para la aproximación del Hessiano dada por (2.12):

$$\mathbf{H}(\mathbf{y}) := \mathbf{J}_\mathbf{f}^\top(\mathbf{y})\mathbf{J}_\mathbf{f}(\mathbf{y}) + \nabla^2\mathcal{R}(\mathbf{y}).$$

Los siguientes lemas son fundamentales para el análisis de la convergencia local de las distintas variantes del Algoritmo 1, dependiendo de la elección del Jacobiano.

Lema 1. *Supongamos que la matriz Jacobiana $\mathbf{J}_\mathbf{f}(\mathbf{y})$ es Lipschitz continua en el conjunto acotado $\mathcal{B}(\mathbf{y}^*, \rho)$, es decir, existe una constante $L_\mathbf{f} > 0$ tal que*

$$\|\mathbf{J}_\mathbf{f}(\mathbf{y}_1) - \mathbf{J}_\mathbf{f}(\mathbf{y}_2)\|_2 \leq L_\mathbf{f}\|\mathbf{y}_1 - \mathbf{y}_2\|_2, \quad \forall \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{B}(\mathbf{y}^*, \rho).$$

Además, supongamos que $\mathcal{R}(\mathbf{y})$ es de clase $\mathcal{C}^2$ y $\nabla^2\mathcal{R}(\mathbf{y})$ es también Lipschitz continua con constante $L_\mathcal{R} > 0$. Entonces la aproximación del Hessiano $\mathbf{H}(\mathbf{y})$ es Lipschitz continua en $\mathcal{B}(\mathbf{y}^*, \rho)$, con constante Lipschitz $2KL_\mathbf{f} + L_\mathcal{R}$, donde

$$K := \sup\{\|\mathbf{J}_\mathbf{f}(\mathbf{y})\|_2, \mathbf{y} \in \overline{\mathcal{B}(\mathbf{y}^*, \rho)}\}.$$

Demostración. Sean $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{B}(\mathbf{y}^*, \rho)$. Entonces:

$$\begin{aligned}\left\|\mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1)\mathbf{J}_\mathbf{f}(\mathbf{y}_1) + \nabla^2\mathcal{R}(\mathbf{y}_1) - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}_2)\mathbf{J}_\mathbf{f}(\mathbf{y}_2) - \nabla^2\mathcal{R}(\mathbf{y}_2)\right\|_2 &\leq \left\|\mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1)\mathbf{J}_\mathbf{f}(\mathbf{y}_1) - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}_2)\mathbf{J}_\mathbf{f}(\mathbf{y}_2)\right\|_2 \\ &\quad + \left\|\nabla^2\mathcal{R}(\mathbf{y}_1) - \nabla^2\mathcal{R}(\mathbf{y}_2)\right\|_2. \qquad (2.18)\end{aligned}$$

El primer término de (2.18) se descompone:

$$\begin{aligned}\left\|\mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1)\mathbf{J}_\mathbf{f}(\mathbf{y}_1) - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}_2)\mathbf{J}_\mathbf{f}(\mathbf{y}_2)\right\|_2 &= \left\|\mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1)\mathbf{J}_\mathbf{f}(\mathbf{y}_1) - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1)\mathbf{J}_\mathbf{f}(\mathbf{y}_2) + \mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1)\mathbf{J}_\mathbf{f}(\mathbf{y}_2) - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}_2)\mathbf{J}_\mathbf{f}(\mathbf{y}_2)\right\|_2 \\ &\leq \left\|\mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1)\right\|_2\left\|\mathbf{J}_\mathbf{f}(\mathbf{y}_1) - \mathbf{J}_\mathbf{f}(\mathbf{y}_2)\right\|_2 + \left\|\mathbf{J}_\mathbf{f}(\mathbf{y}_2)\right\|_2\left\|\mathbf{J}_\mathbf{f}^\top(\mathbf{y}_1) - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}_2)\right\|_2. \qquad (2.19)\end{aligned}$$

Como $\mathcal{B}(\mathbf{y}^*, \rho)$ es un conjunto acotado y $\mathbf{J}_\mathbf{f}(\mathbf{y})$ es Lipschitz continua por hipótesis, podemos definir $K := \sup\{\|\mathbf{J}_\mathbf{f}(\mathbf{y})\|_2, \mathbf{y} \in \overline{\mathcal{B}(\mathbf{y}^*, \rho)}\} < \infty$ de manera que (2.19) $< 2KL_\mathbf{f}\|\mathbf{y}_1 - \mathbf{y}_2\|_2$.

El segundo término está acotado por hipótesis:

$$\|\nabla^2\mathcal{R}(\mathbf{y}_1) - \nabla^2\mathcal{R}(\mathbf{y}_2)\|_2 \leq L_\mathcal{R}\|\mathbf{y}_1 - \mathbf{y}_2\|_2.$$

Por lo tanto, se concluye que (2.18) $\leq (2KL_\mathbf{f} + L_\mathcal{R})\|\mathbf{y}_1 - \mathbf{y}_2\|_2$, es decir, la aproximación del Hessiano es Lipschitz continua en $\mathcal{B}(\mathbf{y}^*, \rho)$ con constante $2KL_\mathbf{f} + L_\mathcal{R}$. ■

Lema 2. *Supongamos que la aproximación del Hessiano $\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{J}_\mathbf{f}(\mathbf{y}^*) + \nabla^2\mathcal{R}(\mathbf{y}^*)$ es invertible y la matriz Jacobiana $\mathbf{J}_\mathbf{f}(\mathbf{y})$ es Lipschitz continua en el conjunto acotado $\mathcal{B}(\mathbf{y}^*, \rho)$. Además, supongamos que $\mathcal{R}(\mathbf{y})$ es de clase $\mathcal{C}^2$ y $\nabla^2\mathcal{R}(\mathbf{y})$ es también Lipschitz continua con constante $L_\mathcal{R} > 0$. Entonces existe $0 < \sigma \leq \rho$ tal que para todo $\mathbf{y} \in \mathcal{B}(\mathbf{y}^*, \sigma)$ valen las siguientes desigualdades:*

$$\|\mathbf{H}(\mathbf{y})\|_2 \leq 2\|\mathbf{H}(\mathbf{y}^*)\|_2 \qquad (2.20)$$

$$\|\mathbf{H}(\mathbf{y})^{-1}\|_2 \leq 2\|\mathbf{H}(\mathbf{y}^*)^{-1}\|_2. \qquad (2.21)$$

Demostración. Por el Lema 1 tenemos que $\mathbf{H}(\mathbf{y})$ es Lipschitz continua, esto es, existe $\gamma > 0$ tal que

$$\|\mathbf{H}(\mathbf{y})\|_2 - \|\mathbf{H}(\mathbf{y}^*)\|_2 \leq \|\mathbf{H}(\mathbf{y}) - \mathbf{H}(\mathbf{y}^*)\|_2 \leq \gamma\|\mathbf{y} - \mathbf{y}^*\|_2 = \gamma\|\mathbf{e}\|_2. \tag{2.22}$$

De (2.22), eligiendo un σ apropiado tal que $\gamma\|\mathbf{e}\|_2 \leq \|\mathbf{H}(\mathbf{y}^*)\|_2$, obtenemos la desigualdad (2.20) para cualquier $\mathbf{y} \in \mathcal{B}(\mathbf{y}^*, \sigma)$.

Para verificar la ecuación (2.21) consideramos

$$\|\mathbf{I} - \mathbf{H}(\mathbf{y}^*)^{-1}\mathbf{H}(\mathbf{y})\|_2 = \|\mathbf{H}(\mathbf{y}^*)^{-1}(\mathbf{H}(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}))\|_2 \leq \gamma\|\mathbf{H}(\mathbf{y}^*)^{-1}\|_2\|\mathbf{e}\|_2 \leq \gamma\sigma\|\mathbf{H}(\mathbf{y}^*)^{-1}\|_2.$$

Si elegimos $\sigma = \text{mín}\left(\frac{\|\mathbf{H}(\mathbf{y}^*)^{-1}\|_2^{-1}}{2\gamma}, \rho\right)$ deducimos que

$$\|\mathbf{I} - \mathbf{H}(\mathbf{y}^*)^{-1}\mathbf{H}(\mathbf{y})\|_2 < \frac{1}{2}.$$

Aplicamos el Lema de Banach [16, Theorem 1.2.1], que dice que dadas $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ matrices tales que $\mathbf{B}$ es una inversa aproximada de $\mathbf{A}$, es decir, $\|\mathbf{I} - \mathbf{BA}\|_2 < 1$, entonces, $\mathbf{A}$ y $\mathbf{B}$ son no singulares y se satisface:

$$\|\mathbf{A}^{-1}\|_2 \leq \frac{\|\mathbf{B}\|_2}{1 - \|\mathbf{I} - \mathbf{BA}\|_2}.$$

Como $\mathbf{H}(\mathbf{y}^*)^{-1}$ es una inversa aproximada de $\mathbf{H}(\mathbf{y})$, entonces se satisface

$$\|\mathbf{H}(\mathbf{y})^{-1}\|_2 \leq \frac{\|\mathbf{H}(\mathbf{y}^*)^{-1}\|_2}{1 - \|\mathbf{I} - \mathbf{H}(\mathbf{y}^*)^{-1}\mathbf{H}(\mathbf{y})\|_2}.$$

Usamos la cota $\|\mathbf{I} - \mathbf{H}(\mathbf{y}^*)^{-1}\mathbf{H}(\mathbf{y})\|_2 < \frac{1}{2}$, de donde

$$1 - \|\mathbf{I} - \mathbf{H}(\mathbf{y}^*)^{-1}\mathbf{H}(\mathbf{y})\|_2 > 1 - \tfrac{1}{2} = \tfrac{1}{2}$$

$$\Rightarrow \|\mathbf{H}(\mathbf{y})^{-1}\|_2 \ \leq \ \frac{\|\mathbf{H}(\mathbf{y}^*)^{-1}\|_2}{\frac{1}{2}} = 2\,\|\mathbf{H}(\mathbf{y}^*)^{-1}\|_2,$$

y así obtenemos la desigualdad (2.21). ■

Teorema 2. *Sea* $\varphi(\mathbf{y}) = \frac{1}{2}\|\mathbf{f}(\mathbf{y})\|_2^2 + \mathcal{R}(\mathbf{y})$ *y sea* $\mathbf{y}^*$ *una solución óptima de* $\text{mín}_{\mathbf{y}}\,\varphi(\mathbf{y})$. *Supongamos que la matriz Jacobiana* $\mathbf{J}_\mathbf{f}(\mathbf{y})$ *de* $\mathbf{f}$ *es Lipschitz en* $\mathcal{B}(\mathbf{y}^*, \rho)$ *y tiene rango columna completo en* $\mathbf{y}^*$. *Además, supongamos que* $\mathcal{R}(\mathbf{y})$ *es convexa, de clase* $\mathcal{C}^2$ *y* $\nabla^2\mathcal{R}(\mathbf{y})$ *también es Lipschitz continua con constante* $L_\mathcal{R} > 0$. *Entonces existe* $K > 0$ *y* $0 < \sigma \leq \rho$ *tal que para todo* $\mathbf{y} \in \mathcal{B}(\mathbf{y}^*, \sigma)$ *la iteración del Algoritmo 1 con* $\mathbf{y}^{(0)} = \mathbf{y}$ *y el Jacobiano de Golub Pereyra (2.13) satisface*

$$\|\mathbf{e}_{k+1}\|_2 \leq K\Big(\|\mathbf{e}_k\|_2^2 + \|\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\|_2\|\mathbf{e}_k\|_2\Big)$$

Demostración. Consideremos el problema reducido

$$\min_{\mathbf{y}} \varphi(\mathbf{y}) = \min_{\mathbf{y}} \frac{1}{2}\|\mathbf{f}(\mathbf{y})\|_2^2 + \mathcal{R}(\mathbf{y}) = \min_{\mathbf{y}} \frac{1}{2}\| - \mathcal{P}_\mathbf{K}^\perp(\mathbf{y})\|_2^2 + \mathcal{R}(\mathbf{y}).$$

Nos basamos en la expresión para el Jacobiano de $\mathbf{f}$, definido por las columnas dadas en (2.13). Por hipótesis $\mathbf{J}_\mathbf{f}(\mathbf{y})$ tiene rango columna completo en $\mathbf{y}^*$ y $\mathcal{R}(\mathbf{y})$ es convexa, en particular, semidefinida positiva, entonces $\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{J}_\mathbf{f}(\mathbf{y}^*) + \nabla^2\mathcal{R}(\mathbf{y}^*)$ es estrictamente definida positiva y no singular. Por el Lema 2 existe $0 < \sigma \leq \rho$ tal que la aproximación del Hessiano dada por $\mathbf{J}_\mathbf{f}^\top(\mathbf{y})\mathbf{J}_\mathbf{f}(\mathbf{y}) + \nabla^2\mathcal{R}(\mathbf{y})$ es no singular para todo $\mathbf{y} \in \mathcal{B}(\mathbf{y}^*, \sigma)$. La iteración de $\mathbf{y}$ está definida entonces por

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \mathbf{H}(\mathbf{y}^{(k)})^{-1}\nabla\varphi(\mathbf{y}^{(k)}) \tag{2.23}$$

donde $\mathbf{H}(\mathbf{y}^{(k)}) = \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)}) + \nabla^2\mathcal{R}(\mathbf{y}^{(k)})$ y $\nabla\varphi(\mathbf{y}^{(k)}) = \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{f}(\mathbf{y}^{(k)}) + \nabla\mathcal{R}(\mathbf{y}^{(k)})$.

Restando $\mathbf{y}^*$ a ambos lados de la ecuación (2.23) y reorganizando los términos obtenemos

$$\begin{aligned}\mathbf{e}_{k+1} &= \mathbf{e}_k - \mathbf{H}(\mathbf{y}^{(k)})^{-1}\Big(\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{f}(\mathbf{y}^{(k)}) + \nabla\mathcal{R}(\mathbf{y}^{(k)})\Big)\\ &= \mathbf{H}(\mathbf{y}^{(k)})^{-1}\Big(\mathbf{H}(\mathbf{y}^{(k)})\mathbf{e}_k - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{f}(\mathbf{y}^{(k)}) - \nabla\mathcal{R}(\mathbf{y}^{(k)})\Big)\\ &= \mathbf{H}(\mathbf{y}^{(k)})^{-1}\Big(\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k + \nabla^2\mathcal{R}(\mathbf{y}^{(k)})\mathbf{e}_k - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{f}(\mathbf{y}^{(k)}) - \nabla\mathcal{R}(\mathbf{y}^{(k)})\Big)\\ &= \mathbf{H}(\mathbf{y}^{(k)})^{-1}\Big[\underbrace{\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\Big(\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - \mathbf{f}(\mathbf{y}^{(k)})\Big)}_{A} + \underbrace{\nabla^2\mathcal{R}(\mathbf{y}^{(k)})\mathbf{e}_k - \nabla\mathcal{R}(\mathbf{y}^{(k)})}_{B}\Big]\end{aligned} \tag{2.24}$$

Primero manipulamos A. Notar que

$$\begin{aligned}\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - \mathbf{f}(\mathbf{y}^{(k)}) &= \mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - \mathbf{f}(\mathbf{y}^*) + \mathbf{f}(\mathbf{y}^*) - \mathbf{f}(\mathbf{y}^{(k)})\\ &= -\mathbf{f}(\mathbf{y}^*) + \Big(\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - (\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{f}(\mathbf{y}^*))\Big)\end{aligned} \tag{2.25}$$

Haciendo una expansión de Taylor de orden dos centrada en $\mathbf{y}^*$ del residuo $\mathbf{f}(\mathbf{y}^{(k)})$ obtenemos

$$\mathbf{f}(\mathbf{y}^{(k)}) = \mathbf{f}(\mathbf{y}^*) + \mathbf{J}_\mathbf{f}(\mathbf{y}^*)(\mathbf{y}^{(k)} - \mathbf{y}^*) + \mathcal{O}(\|\mathbf{y}^{(k)} - \mathbf{y}^*\|_2^2) = \mathbf{f}(\mathbf{y}^*) + \mathbf{J}_\mathbf{f}(\mathbf{y}^*)\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)$$

Usando la relación $\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{f}(\mathbf{y}^*) = \mathbf{J}_\mathbf{f}(\mathbf{y}^*)\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)$ llegamos a que

$$\begin{aligned}\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - (\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{f}(\mathbf{y}^*)) &= \mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - \mathbf{J}_\mathbf{f}(\mathbf{y}^*)\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\\ &= (\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{J}_\mathbf{f}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\end{aligned} \tag{2.26}$$

y entonces

$$\begin{aligned}\|\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - (\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{f}(\mathbf{y}^*))\|_2 &\le \|\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{J}_\mathbf{f}(\mathbf{y}^*)\|_2\|\mathbf{e}_k\|_2 + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\\ &\le L_\mathbf{f}\|\mathbf{e}_k\|_2\|\mathbf{e}_k\|_2 + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\\ &\le C_1\|\mathbf{e}_k\|_2^2,\end{aligned} \tag{2.27}$$

donde $L_\mathbf{f}$ es la constante Lipschitz de la matriz Jacobiana y $C_1 = L_\mathbf{f} + L$, con L una constante tal que $\mathcal{O}(\|\mathbf{e}_k\|_2^2) \le L\|\mathbf{e}_k\|_2^2$.

Haciendo una expansión de Taylor de $\mathbf{J}_\mathbf{f}$ centrada en $\mathbf{y}^*$ obtenemos

$$\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)}) = \mathbf{J}_\mathbf{f}(\mathbf{y}^*) + \mathbf{J}'_\mathbf{f}(\mathbf{y}^*)\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2).$$

Luego

$$\begin{aligned}\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\mathbf{f}(\mathbf{y}^*) &= \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) + \mathbf{e}_k^\top\mathbf{J}'_\mathbf{f}(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\\ &= \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) + \mathbf{e}_k^\top\Big(\nabla^2\varphi(\mathbf{y}^*) - \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{J}_\mathbf{f}(\mathbf{y}^*) - \nabla^2\mathcal{R}(\mathbf{y}^*)\Big) + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\\ &= \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) + \mathbf{e}_k^\top\Big(\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\Big) + \mathcal{O}(\|\mathbf{e}_k\|_2^2),\end{aligned} \tag{2.28}$$

ya que $\nabla^2\varphi(\mathbf{y}^*) = \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{J}_\mathbf{f}(\mathbf{y}^*) + \mathbf{J}'_\mathbf{f}(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) + \nabla^2\mathcal{R}(\mathbf{y}^*)$.

Sustituyendo (2.25), (2.26) y (2.28) en la expresión para A entonces

$$\begin{aligned}A &= \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\Big(\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)})\mathbf{e}_k - \mathbf{f}(\mathbf{y}^{(k)})\Big)\\ &= \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\Big(-\mathbf{f}(\mathbf{y}^*) + (\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{J}_\mathbf{f}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\Big)\\ &= -\mathbf{J}_\mathbf{f}^\top(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) - \mathbf{e}_k^\top\Big(\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\Big) + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\\ &\quad + \mathbf{J}_\mathbf{f}^\top(\mathbf{y}^{(k)})\Big((\mathbf{J}_\mathbf{f}(\mathbf{y}^{(k)}) - \mathbf{J}_\mathbf{f}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\Big).\end{aligned} \tag{2.29}$$

Por otro lado, haciendo la expansión de Taylor centrada en $\mathbf{y}^*$ de $\mathcal{R}(\mathbf{y}^{(k)})$ obtenemos

$$\nabla\mathcal{R}(\mathbf{y}^{(k)}) = \nabla\mathcal{R}(\mathbf{y}^*) + \nabla^2\mathcal{R}(\mathbf{y}^*)\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2). \tag{2.30}$$

Además,

$$\|\nabla^2\mathcal{R}(\mathbf{y}^{(k)}) - \nabla^2\mathcal{R}(\mathbf{y}^*)\|_2\|\mathbf{e}_k\|_2 + \mathcal{O}(\|\mathbf{e}_k\|_2^2) \leq L_{\mathcal{R}}\|\mathbf{e}_k\|_2^2 + \mathcal{O}(\|\mathbf{e}_k\|_2^2) \leq C_2\|\mathbf{e}_k\|_2^2, \tag{2.31}$$

donde $L_{\mathcal{R}}$ es la constante Lipschitz de $\nabla^2\mathcal{R}$ y $C_2 = L_{\mathcal{R}}+L$, con L una constante tal que $\mathcal{O}(\|\mathbf{e}_k\|_2^2) \leq L\|\mathbf{e}_k\|_2^2$.

Sustituyendo (2.30) en B, la expresión toma la forma

$$\begin{aligned} B &= \nabla^2\mathcal{R}(\mathbf{y}^{(k)})\mathbf{e}_k - \nabla\mathcal{R}(\mathbf{y}^{(k)}) \\ &= \nabla^2\mathcal{R}(\mathbf{y}^{(k)})\mathbf{e}_k - \nabla\mathcal{R}(\mathbf{y}^*) - \nabla^2\mathcal{R}(\mathbf{y}^*)\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2) \\ &= -\nabla\mathcal{R}(\mathbf{y}^*) + (\nabla^2\mathcal{R}(\mathbf{y}^{(k)}) - \nabla^2\mathcal{R}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2). \end{aligned} \tag{2.32}$$

Sumando $A + B$ llegamos a que

$$\begin{aligned} A + B = &-\mathbf{J}_{\mathbf{f}}^\top(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) - \nabla\mathcal{R}(\mathbf{y}^*) \\ &- \mathbf{e}_k^\top\Big(\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\Big) + \mathcal{O}(\|\mathbf{e}_k\|_2^2) \\ &+ \mathbf{J}_{\mathbf{f}}^\top(\mathbf{y}^{(k)})\Big((\mathbf{J}_{\mathbf{f}}(\mathbf{y}^{(k)}) - \mathbf{J}_{\mathbf{f}}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\Big) \\ &+ (\nabla^2\mathcal{R}(\mathbf{y}^{(k)}) - \nabla^2\mathcal{R}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2). \end{aligned} \tag{2.33}$$

Como $\mathbf{y}^*$ es un punto estacionario, entonces $\mathbf{J}_{\mathbf{f}}^\top(\mathbf{y}^*)\mathbf{f}(\mathbf{y}^*) + \nabla\mathcal{R}(\mathbf{y}^*) = 0$ y finalmente la expresión (2.33) queda

$$\begin{aligned} A + B = &-\mathbf{e}_k^\top\Big(\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\Big) \\ &+ \mathbf{J}_{\mathbf{f}}^\top(\mathbf{y}^{(k)})\Big((\mathbf{J}_{\mathbf{f}}(\mathbf{y}^{(k)}) - \mathbf{J}_{\mathbf{f}}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2)\Big) \\ &+ (\nabla^2\mathcal{R}(\mathbf{y}^{(k)}) - \nabla^2\mathcal{R}(\mathbf{y}^*))\mathbf{e}_k + \mathcal{O}(\|\mathbf{e}_k\|_2^2). \end{aligned} \tag{2.34}$$

Usando las cotas (2.27) y (2.31) llegamos a que

$$\|A + B\|_2 \leq \|\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\|_2\|\mathbf{e}_k\|_2 + C_1\|\mathbf{J}_{\mathbf{f}}^\top(\mathbf{y}^{(k)})\|_2\|\mathbf{e}_k\|_2^2 + C_2\|\mathbf{e}_k\|_2^2. \tag{2.35}$$

Reemplazando (2.35) en (2.24) obtenemos

$$\|\mathbf{e}_{k+1}\|_2 \leq \|\mathbf{H}(\mathbf{y}^{(k)})^{-1}\|_2\Big((C_1\|\mathbf{J}_{\mathbf{f}}^\top(\mathbf{y}^{(k)})\|_2 + C_2)\|\mathbf{e}_k\|_2^2 + \|\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\|_2\|\mathbf{e}_k\|_2\Big).$$

Eligiendo una constante apropiada

$$K = \sup_{\mathbf{y}^{(k)}\in\overline{\mathcal{B}(\mathbf{y}^*,\sigma)}} \left\|\mathbf{H}(\mathbf{y}^{(k)})^{-1}\right\|_2 \left(1 + C_1\|\mathbf{J}_{\mathbf{f}}^\top(\mathbf{y}^{(k)})\|_2 + C_2\right),$$

concluimos que

$$\|\mathbf{e}_{k+1}\|_2 \leq K\Big(\|\mathbf{e}_k\|_2^2 + \|\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\|_2\|\mathbf{e}_k\|_2\Big).$$

■

Observación: Para problemas en los que existe baja nolinealidad, es decir, problemas cuyo modelo es técnicamente no lineal pero su comportamiento es casi lineal en la región de interés, como los problemas de descomposición matricial de bajo rango, la diferencia entre

$$\nabla^2\varphi(\mathbf{y}^*) = \nabla^2\left(\tfrac{1}{2}\|\mathbf{f}(\mathbf{y}^*)\|_2^2 + \mathcal{R}(\mathbf{y}^*)\right) \quad \text{y} \quad \mathbf{H}(\mathbf{y}^*) = \mathbf{J}_{\mathbf{f}}(\mathbf{y}^*)^\top\mathbf{J}_{\mathbf{f}}(\mathbf{y}^*) + \nabla^2\mathcal{R}(\mathbf{y}^*)$$

suele ser insignificante. En estos casos, el Algoritmo 1 con Jacobiano GP puede alcanzar tasas de convergencia superlineales, e incluso tasas de convergencia cuadrática en ciertas condiciones.

Recordemos que, dado un método iterativo con iteraciones $\{\mathbf{y}^{(k)}\}$ y solución $\mathbf{y}^*$, se definen las siguientes tasas de convergencia:

- **Convergencia lineal:** existe una constante $0 < C < 1$ tal que

$$\lim_{k\to\infty} \frac{\|\mathbf{y}^{(k+1)} - \mathbf{y}^*\|}{\|\mathbf{y}^{(k)} - \mathbf{y}^*\|} = C.$$

- **Convergencia superlineal:**

$$\lim_{k\to\infty} \frac{\|\mathbf{y}^{(k+1)} - \mathbf{y}^*\|}{\|\mathbf{y}^{(k)} - \mathbf{y}^*\|} = 0.$$

- **Convergencia cuadrática:** existe una constante $C > 0$ tal que

$$\lim_{k\to\infty} \frac{\|\mathbf{y}^{(k+1)} - \mathbf{y}^*\|}{\|\mathbf{y}^{(k)} - \mathbf{y}^*\|^2} = C.$$

- **Convergencia de orden** $p > 1$**:** existe una constante $C > 0$ tal que

$$\lim_{k\to\infty} \frac{\|\mathbf{y}^{(k+1)} - \mathbf{y}^*\|}{\|\mathbf{y}^{(k)} - \mathbf{y}^*\|^p} = C.$$

Equivalente, se tiene que la secuencia converge con orden p si

$$\|\mathbf{y}^{(k+1)} - \mathbf{y}^*\| \leq C\|\mathbf{y}^{(k)} - \mathbf{y}^*\|^p \quad \text{para } k \text{ suficientemente grande,}$$

con $C > 0$ una constante finita .

Corolario 1 (Convergencia local)**.** *Sea* $\mathbf{y}^*$ *una solución del problema de mínimos cuadrados no lineales* $\min_{\mathbf{y}} \varphi(\mathbf{y}) = \min_{\mathbf{y}} \frac{1}{2}\|\mathbf{f}(\mathbf{y})\|_2^2 + \mathcal{R}(\mathbf{y})$*. Definamos*

$$\delta \;:=\; \left\|\nabla^2\varphi(\mathbf{y}^*) - \mathbf{H}(\mathbf{y}^*)\right\|_2.$$

Sea $\mathbf{e}_k = \mathbf{y}^{(k)} - \mathbf{y}^*$ *el error en la k-ésima iteración. Supongamos que en un entorno de* $\mathbf{y}^*$ *el Jacobiano* $\mathbf{J}_{\mathbf{f}}(\mathbf{y})$ *y* $\nabla^2\mathcal{R}(\mathbf{y})$ *son Lipschitz tal que la constante* K *en la cota dada en el Teorema 2:*

$$\|\mathbf{e}_{k+1}\|_2 \leq K\Big(\|\mathbf{e}_k\|_2^2 + \delta\|\mathbf{e}_k\|_2\Big)$$

existe y es finita. Sea $\eta := K\delta$*. Si* $0 \leq \eta < 1$ *y*

$$\|\mathbf{e}_0\|_2 < \frac{1-\eta}{2K},$$

Entonces

$$\|\mathbf{e}_k\|_2 \leq \Big(\frac{1+\eta}{2}\Big)^k \|\mathbf{e}_0\|_2,$$

garantizando convergencia lineal*. En particular, si* $\nabla^2\varphi(\mathbf{y}^*) = \mathbf{H}(\mathbf{y}^*)$ *entonces* $\delta = 0$*, de manera que*

$$\|\mathbf{e}_{k+1}\|_2 \leq K\|\mathbf{e}_k\|_2^2,$$

garantizando convergencia cuadrática*.*

Demostración. Por el Teorema 2 existe $K > 0$ y un radio $\sigma > 0$ tales que, para todo $\mathbf{y} \in B(\mathbf{y}^*, \sigma)$,

$$\|\mathbf{e}_{k+1}\|_2 \leq K\big(\|\mathbf{e}_k\|_2^2 + \delta\|\mathbf{e}_k\|_2\big) = \big(K\|\mathbf{e}_k\|_2 + \eta\big)\|\mathbf{e}_k\|.$$

Para $k = 0$ se tiene

$$K\|\mathbf{e}_0\|_2 \;<\; \frac{1-\eta}{2} \quad \Rightarrow \quad K\|\mathbf{e}_0\|_2 + \eta \;\leq\; \frac{1-\eta}{2} + \eta = \frac{1+\eta}{2} < 1,$$

de donde

$$\|\mathbf{e}_1\|_2 \leq \big(K\|\mathbf{e}_0\|_2 + \eta\big)\,\|\mathbf{e}_k\|_2 \leq \frac{1+\eta}{2}\,\|\mathbf{e}_0\|_2.$$

Suponiendo inductivamente $\|\mathbf{e}_k\|_2 \leq (\frac{1+\eta}{2})^k \|\mathbf{e}_0\|_2$, se tiene que

$$\begin{aligned}\|\mathbf{e}_{k+1}\|_2 &\leq \big(K\|\mathbf{e}_k\|_2 + \eta\big)\|\mathbf{e}_k\|_2 \leq \Big(\Big(\frac{1+\eta}{2}\Big)^k K\|\mathbf{e}_0\|_2 + \eta\Big)\Big(\frac{1+\eta}{2}\Big)^k \|\mathbf{e}_0\|_2 \\ &< \Big(K\|\mathbf{e}_0\|_2 + \eta\Big)\Big(\frac{1+\eta}{2}\Big)^k \|\mathbf{e}_0\|_2 \leq \Big(\frac{1+\eta}{2}\Big)\Big(\frac{1+\eta}{2}\Big)^k \|\mathbf{e}_0\|_2 = \Big(\frac{1+\eta}{2}\Big)^{k+1} \|\mathbf{e}_0\|_2,\end{aligned}$$

ya que $\frac{1+\eta}{2} < 1$ implica $(\frac{1+\eta}{2})^k < 1$. Por inducción concluimos $\|\mathbf{e}_k\|_2 \leq (\frac{1+\eta}{2})^k \|\mathbf{e}_0\|_2$ para todo k, implicando convergencia lineal.

Si $\delta = 0$, la misma cota se reduce a $\|\mathbf{e}_{k+1}\|_2 \leq K\,\|\mathbf{e}_k\|_2^2$, lo cual da la convergencia cuadrática. ■

2.3. Funciones regularizantes

En esta sección vamos a considerar el método definido en la Sección 2.1.1 con dos elecciones particulares para $\mathcal{R}(\mathbf{y})$. Los resultados también valen para el caso inexacto, recordando que en cada iteración se calcula una aproximación de $\mathbf{x}^{(k)}$.

La primera opción consiste en añadir un término regularizante cuadrático de la forma $\mathcal{R}(\mathbf{y}) = \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|^2$, donde $\mu > 0$ e $\mathbf{y}_0$ representa un valor cercano del valor original $\mathbf{y}_{\text{true}}$. Esta regularización es convexa, diferenciable y computacionalmente eficiente (en particular, tiene una expresión sencilla de calcular). Además, permite incorporar conocimiento a priori sobre $\mathbf{y}$, favoreciendo soluciones cercanas a un valor considerado razonable.

Una segunda alternativa es plantear $\mathcal{R}(\mathbf{y}) = -\sum_{j=1}^{r} \mu_j^2 \log(y_j)$, donde $\mu_j > 0$ e $y_j > 0$ para todo j. Esta función regularizante es útil cuando se desea garantizar la positividad de los parámetros, como ocurre en muchos modelos físicos, incluyendo aquellos relacionados con el desenfoque. Al penalizar fuertemente valores cercanos a cero, esta regularización evita soluciones triviales o inestables, y ayuda a controlar la escala de los parámetros sin necesidad de imponer restricciones explícitas.

2.3.1. Regularización vía norma 2

Consideraremos primero el caso en que la función de regularización es de tipo cuadrático, es decir, $\mathcal{R}(\mathbf{y}) = \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|_2^2$. En este caso, el problema de minimización (2.1) adopta la siguiente forma:

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{L}\mathbf{x}\|_2^2 + \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|_2^2. \tag{2.36}$$

Utilizando la notación (2.6), tenemos que para un $\mathbf{y}$ fijo vale

$$\mathbf{x}(\mathbf{y}) := \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{K}(\mathbf{y})\mathbf{x} - \mathbf{d}\|_2^2 + \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|_2^2 = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{K}(\mathbf{y})\mathbf{x} - \mathbf{d}\|_2^2,$$

de donde obtenemos la solución cerrada para $\mathbf{x}$

$$\mathbf{x}(\mathbf{y}) = \mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d}.$$

Substituimos $\mathbf{x} = \mathbf{x}(\mathbf{y})$ en el problema (2.36) y obtenemos

$$\min_{\mathbf{y}} \frac{1}{2}\|\mathbf{K}(\mathbf{y})\mathbf{K}(\mathbf{y})^{\dagger}\mathbf{d} - \mathbf{d}\|_2^2 + \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|_2^2 = \min_{\mathbf{y}} \frac{1}{2}\left\| \begin{bmatrix} -\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d} \\ \mu(\mathbf{y} - \mathbf{y}_0) \end{bmatrix} \right\|_2^2.$$

A partir de la igualdad anterior se define

$$\mathbf{F}(\mathbf{y}) = \begin{bmatrix} -\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d} \\ \mu(\mathbf{y} - \mathbf{y}_0) \end{bmatrix}.$$

Entonces debemos resolver el siguiente problema de minimización no lineal reducido

$$\min_{\mathbf{y}} \frac{1}{2}\|\mathbf{F}(\mathbf{y})\|_2^2.$$

Para resolverlo, aplicamos el método de Gauss-Newton, cuyas iteraciones quedan definidas por

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \mathbf{s}^{(k)},\ k = 0, 1, 2, ...,$$

donde $\mathbf{s}^{(k)}$ se obtiene como solución de

$$\mathbf{s}^{(k)} = \arg\min_{\mathbf{s}} \left\| \mathbf{J}_{\mathbf{F}}(\mathbf{y}^{(k)})\mathbf{s} + \mathbf{F}(\mathbf{y}^{(k)}) \right\|_2^2, \tag{2.37}$$

con $\mathbf{J}_{\mathbf{F}}\colon \mathbb{R}^r \to \mathbb{R}^{(m+q+r)\times r}$ la matriz Jacobiana de $\mathbf{F}$. Usando que

$$\nabla\mathcal{R}(\mathbf{y}) = \mu^2(\mathbf{y} - \mathbf{y}_0) \text{ y } \nabla^2\mathcal{R}(\mathbf{y}) = \mu^2\mathbf{I}_r,$$

obtenemos que $\mathbf{J}_{\mathbf{F}}(\mathbf{y})$ tiene forma de bloque del tipo

$$\mathbf{J}_{\mathbf{F}}(\mathbf{y}) = \begin{bmatrix} \mathbf{J}_{\mathbf{f}}(\mathbf{y}) \\ \mu\mathbf{I}_r \end{bmatrix}$$

donde $\mathbf{J}_{\mathbf{f}}(\mathbf{y})$ es la matriz Jacobiana de la función $\mathbf{f}(\mathbf{y}) = -\mathcal{P}^{\perp}_{\mathbf{K}(\mathbf{y})}\mathbf{d}$. La columna $j$ésima de $\mathbf{J}_{\mathbf{f}}(\mathbf{y})$ está dada por (2.13), de donde concluimos que

$$[\mathbf{J}_{\mathbf{F}}(\mathbf{y})]_j = \begin{bmatrix} \frac{\partial\mathbf{f}(\mathbf{y})}{\partial\mathbf{y}_j} \\ \mu e_j \end{bmatrix}$$

Notar que el funcional regularizante $\mathcal{R}(\mathbf{y}) = \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|_2^2$ cumple con las hipótesis del Teorema 2 y que entonces podemos asegurar convergencia local para este método.

2.3.2. Regularización vía logaritmos

En este caso consideramos la regularización del parámetro $\mathbf{y} = (y_1, \ldots, y_r)$, con $y_j > 0\ \forall j$, de la forma

$$\mathcal{R}(\mathbf{y}) = -\sum_{1\le j\le r} \mu_j^2 \log(y_j),$$

con constantes positivas μ_j, $j = 1, \ldots, r$. Luego nuestro problema de minimización (2.1) toma la forma

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{L}\mathbf{x}\|_2^2 - \sum_{1\le j\le r} \mu_j^2 \log(y_j).$$

Al igual que el caso anterior, aplicando el modelo de VarPro, llegamos al problema de minimización $\min_{\mathbf{y}} \varphi(\mathbf{y})$, con

$$\varphi(\mathbf{y}) = \frac{1}{2}\| - \mathcal{P}^{\perp}_{\mathbf{K}}(\mathbf{y})\mathbf{d}\|_2^2 - \sum_{1\le j\le r} \mu_j^2 \log(y_j).$$

Para poder utilizar el método de Newton para resolver este problema de minimización necesitamos calcular el gradiente y el Hessiano de φ:

$$\nabla\varphi(\mathbf{y}) = \mathbf{J}_{\mathbf{f}}^{\top}(\mathbf{y})\mathbf{f}(\mathbf{y}) - \left[\begin{array}{ccc} \frac{\mu_1^2}{y_1} & \cdots & \frac{\mu_r^2}{y_r} \end{array}\right]^{\top}.$$

Utilizamos la siguiente aproximación del Hessiano

$$\nabla^2\varphi(\mathbf{y}) \simeq \mathbf{J}_{\mathbf{f}}^{\top}(\mathbf{y})\mathbf{J}_{\mathbf{f}}(\mathbf{y}) + \mathbf{D}^{\top}(\mathbf{y})\mathbf{D}(\mathbf{y}),$$

donde $\mathbf{D}(\mathbf{y})$ es la matriz diagonal con $[\mathbf{D}(\mathbf{y})]_{jj} = -\frac{\mu_j}{y_j}$, $j = 1, \ldots, r$.

Aplicamos entonces un método iterativo cuyas iteraciones están definidas por

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \mathbf{s}^{(k)},\ k = 0, 1, 2, ...,$$

donde $\mathbf{s}^{(k)}$ se define como

$$\mathbf{s}^{(k)} = \arg\min_{\mathbf{s}} \left\| \left[\begin{array}{c} \mathbf{J}_{\mathbf{f}}(\mathbf{y}^{(k)}) \\ \mathbf{D}(\mathbf{y}^{(k)}) \end{array}\right] \mathbf{s} + \left[\begin{array}{c} \mathbf{f}(\mathbf{y}^{(k)}) \\ \mathbf{u} \end{array}\right] \right\|_2^2 \tag{2.38}$$

y $\mathbf{u} = [\mu_1, \ldots, \mu_r]^{\top}$.

3. REPRESENTACIÓN

Nuestro objeto de estudio son los problemas en los que el desenfoque de una imagen se modela como una convolución entre la imagen original y una función de dispersión del punto (PSF, por sus siglas en inglés). Este modelo es ampliamente utilizado en el procesamiento de imágenes digitales debido a que refleja con precisión los efectos físicos de muchos sistemas de adquisición, como cámaras ópticas, telescopios o microscopios. En particular, cuando el desenfoque es espacialmente invariante, la convolución se puede expresar de forma matricial como en (1.1), donde $\mathbf{A}$ representa la transformación lineal inducida por la PSF. Esta matriz, conocida también como operador de desenfoque, encapsula el efecto del sistema óptico sobre cada punto de la imagen y juega un papel central en la formulación y resolución del problema inverso de restauración.

Comprender cómo se construye la matriz $\mathbf{A}$ es fundamental, no solo desde un punto de vista modelístico, sino también computacional. Su estructura, determinada por la forma de la PSF, las condiciones de borde y la dimensión de la imagen, tiene implicaciones directas en la complejidad y estabilidad de los algoritmos numéricos utilizados. En particular, ciertas estructuras, como las matrices Toeplitz o circulantes, permiten una representación matricial compacta y operaciones eficientes mediante técnicas espectrales, como la transformada rápida de Fourier (FFT, por sus siglas en inglés). Esta posibilidad de explotar la estructura interna de $\mathbf{A}$ resulta clave para el diseño de métodos numéricos rápidos y escalables.

Además, en contextos donde se introduce regularización, como en la formulación de Tikhonov o variantes con penalización adaptativa, tanto $\mathbf{A}$ como la matriz de regularización $\mathbf{L}$ deben ser manipuladas de forma conjunta. En estos casos, resulta particularmente ventajoso que ambas matrices compartan propiedades estructurales, como la diagonalización conjunta (ver la Sección 3.3 para la definición de diagonalización conjunta), lo que permite reducir el costo computacional del cálculo de soluciones aproximadas y derivadas.

El análisis detallado que se presenta a continuación se basa en los Capítulos 3 y 4 de [14], y tiene por objetivo exponer tanto la construcción práctica del operador de desenfoque $\mathbf{A}$ como las distintas formas de aprovechar su estructura algebraica en la implementación de algoritmos eficientes para la restauración de imágenes desenfocadas.

3.1. Función de dispersión del punto

En muchos casos la intensidad de la luz de la PSF se limita a una pequeña región alrededor de su centro (la ubicación del píxel de la fuente puntual), y fuera de un cierto radio, dicha intensidad es prácticamente nula: el desenfoque es un fenómeno local. Además, si asumimos que el proceso de captura de imágenes conserva toda la luz, los valores de los píxeles de la PSF deben sumar 1. Cuando la PSF es invariante respecto a la posición de la fuente puntual decimos que el desenfoque es espacialmente invariante. Como consecuencia de esta naturaleza lineal y local del desenfoque, es común representar la PSF mediante una matriz $\mathbf{P}$ de dimensiones mucho menores que la imagen desenfocada, lo que permite ahorrar espacio de almacenamiento. Nos referimos a $\mathbf{P}$ como la matriz PSF.

En algunos casos, la PSF puede describirse mediante una expresión analítica y, por lo tanto, $\mathbf{P}$ puede construirse a partir de una función. Si el desenfoque tiene un origen físico, es posible obtener una formulación explícita de la PSF. Por ejemplo, la PSF para el desenfoque causado por

la turbulencia atmosférica se puede describir como una función Gaussiana bidimensional, cuyos elementos se definen como

$$p_{ij} = \exp\left(-\frac{1}{2}\begin{bmatrix} i-k \\ j-\ell \end{bmatrix}^T \begin{bmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-\ell \end{bmatrix}\right),$$

donde los parámetros s_1, s_2 y ρ determinan el ancho y la orientación de la PSF, centrada en el elemento (k, ℓ) de la matriz $\mathbf{P}$. Esta matriz siempre debe de escalarse de modo que sus elementos sumen 1. Si $\rho = 0$ en la fórmula para el desenfoque Gaussiano, entonces la PSF es simétrica a lo largo de los ejes vertical y horizontal, y la fórmula toma la forma más simple

$$p_{ij} = \exp\left(-\frac{1}{2}\left(\frac{i-k}{s_1}\right)^2 - \frac{1}{2}\left(\frac{j-\ell}{s_2}\right)^2\right).$$

Si además $s_1 = s_2$, entonces la PSF resulta rotacionalmente simétrica.

Una vez especificada la PSF, se puede construir la matriz de desenfoque $\mathbf{A}$, columna por columna, simplemente insertando los elementos de $\mathbf{P}$ en las posiciones correspondientes y completando con ceros en el resto de la columna. Cada columna de $\mathbf{A}$ es el resultado de difuminar un solo píxel, lo que significa que cada columna de $\mathbf{A}$ representa la función de dispersión de puntos centrada en su píxel correspondiente (i, j).

3.2. Condiciones de borde

Una de las principales dificultades al modelar el desenfoque de imágenes radica en el tratamiento de los bordes. En estas zonas, la información de la imagen original se extiende más allá del dominio observado de la imagen, lo que lleva a una pérdida de información irrecuperable. En situaciones reales, solo se puede tener conocimiento sobre una región finita, el llamado campo de visión (FOV, por sus siglas in inglés), que delimita el área visible del objeto observado. La estrategia más común para mitigar esta deficiencia consiste en asumir ciertas suposiciones sobre el comportamiento específico de la imagen nítida fuera del FOV. Al incorporar estas suposiciones en el modelo, se dice que se imponen condiciones de borde.

La condición de borde más sencilla consiste en asumir la pérdida de fotones fuera del campo de visión, en cuyo caso decimos que la matriz tiene condiciones de borde cero. La imagen exacta es negra (es decir, está compuesta de ceros) fuera del límite. Matricialmente esto se representa como

$$\mathbf{X}_{\text{ext}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{X} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Si al construir la matriz de desenfoque $\mathbf{A}$ se ignoran las condiciones de borde, se está asumiendo implícitamente una condición de borde cero. Esta hipótesis resulta adecuada cuando la imagen real tiene valores cercanos a cero fuera del dominio visible, como ocurre en imágenes astronómicas con fondo negro. No obstante, puede generar artefactos en la reconstrucción, como la aparición de un borde negro artificial. Por lo tanto, a menudo debemos usar otras condiciones de contorno que imponen un modelo más realista del comportamiento de la imagen en el límite, pero que solo utilizan la información disponible, es decir, la imagen dentro de los límites.

La condición de contorno periódica se usa con frecuencia en el procesamiento de imágenes. Esto implica que la imagen se repite (infinitamente) en todas las direcciones. Matricialmente se representa como

$$\mathbf{X}_{\text{ext}} = \begin{bmatrix} \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \end{bmatrix}.$$

También pueden imponerse condiciones de borde reflexivo, que modelan una reflexión especular de la imagen en sus bordes, lo cual puede resultar más realista en ciertos contextos.

La elección de la condición de borde adecuada depende del problema en cuestión. En la práctica, es habitual extender periódicamente únicamente una franja de píxeles cercana al límite.

3.3. Cómputo con matrices estructuradas

En esta sección, analizamos el caso en el que se puede calcular una diagonalización conjunta de $\mathbf{A}$ y $\mathbf{L}$, y cómo dicha propiedad puede utilizarse para calcular Jacobianos de manera eficiente [6]. En particular, supondremos que ambas matrices son diagonalizables simultáneamente mediante una matriz unitaria $\mathbf{Q}$, de modo que

$$\mathbf{A} = \mathbf{Q}^*\mathbf{C}\mathbf{Q} \quad \text{y} \quad \mathbf{L} = \mathbf{Q}^*\mathbf{S}\mathbf{Q} \tag{3.1}$$

donde $\mathbf{C}$ y $\mathbf{S}$ son matrices diagonales que contienen los autovalores de $\mathbf{A}$ y $\mathbf{L}$, respectivamente. Esta es la descomposición espectral unitaria de ambas matrices.

En general, la descomposición espectral de una matriz $\mathbf{A}$ se puede calcular si y solo si $\mathbf{A}$ es una matriz normal, es decir $\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^*$. Si $\mathbf{A}$ tiene entradas reales, los elementos en la descomposición espectral pueden ser complejos, asimismo, sus autovalores son reales o aparecen en pares conjugados complejos. Aunque calcular estas factorizaciones para matrices genéricas grandes puede resultar costoso, existen enfoques eficientes para ciertas matrices estructuradas. Por eficiente se entiende que la descomposición puede calcularse rápidamente y que los requisitos de almacenamiento son manejables.

Para problemas de desenfoque de imágenes espacialmente invariantes, la estructura específica de la matriz $\mathbf{A}$ está determinada por las condiciones de borde impuestas y puede involucrar matrices de Toeplitz, circulantes o de Hankel. Introducimos primero algunas definiciones básicas:

- Una **matriz de Toeplitz** es una matriz constante a lo largo de sus diagonales, es decir, $T_{i,j} = T_{i+1,j+1} = t_{i-j}$. Tiene la forma:

$$\mathbf{T} = \begin{bmatrix} t_0 & t_{-1} & t_{-2} & \cdots \\ t_1 & t_0 & t_{-1} & \cdots \\ t_2 & t_1 & t_0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

- Una **matriz circulante** es un caso particular de matriz de Toeplitz en la que cada fila es una rotación cíclica de la anterior. Tiene la forma:

$$\mathbf{C} = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_1 \\ c_1 & c_0 & \cdots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & c_{n-2} & \cdots & c_0 \end{bmatrix}.$$

- Una **matriz de Hankel** es una matriz constante a lo largo de las antidiagonales, es decir, $H_{i,j} = H_{i+k,j-k} = h_{i+j}$ para $k = 0, \ldots, j-i$. Tiene la forma:

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots \\ h_1 & h_2 & h_3 & \cdots \\ h_2 & h_3 & h_4 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

A partir de estas estructuras unidimensionales, se definen bloques bidimensionales con patrones similares. Usamos la siguiente notación:

BTTB: Bloque Toeplitz con bloques Toeplitz.

BCCB: Bloque circulante con bloques circulantes.

BHHB: Bloque Hankel con bloques Hankel.

BTHB: Bloque Toeplitz con bloques Hankel.

BHTB: Bloque Hankel con bloques Toeplitz.

Con esta terminología podemos describir con precisión la estructura de la matriz de coeficientes $\mathbf{A}$ para las diversas condiciones de borde:

- **Condiciones de contorno cero.** En este caso $\mathbf{A}$ es una matriz BTTB.
- **Condiciones de contorno periódicas.** En este caso $\mathbf{A}$ es una matriz BCCB.
- **Condiciones de contorno reflexivas.** En este caso $\mathbf{A}$ es la suma de las matrices BTTB, BTHB, BHTB y BHHB.

Nos enfocamos en el caso BCCB, que surge cuando se emplean condiciones de borde periódicas. Cualquier matriz BCCB es normal, es decir, $\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^*$. Por lo tanto, $\mathbf{A}$ tiene una descomposición espectral unitaria. Además, cualquier matriz BCCB tiene la descomposición espectral particular:

$$\mathbf{A} = \mathbf{F}^*\mathbf{C}\mathbf{F},$$

donde $\mathbf{F}$ es la matriz bidimensional unitaria de transformada de Fourier discreta (DFT, por sus siglas en inglés). $\mathbf{F}$ es una matriz que aplica la DFT bidimensional a una señal bidimensional (en este caso, una imagen) y es *unitaria*, es decir, $\mathbf{F}^*\mathbf{F} = \mathbf{I}$.

La matriz DFT unitaria de tamaño $n \times n$ en una dimensión está dada por

$$\mathbf{F}_n = \frac{1}{\sqrt{n}} \left[e^{-2\pi i \frac{jk}{n}} \right]_{j,k=0}^{n-1}.$$

Para una señal bidimensional de tamaño $n \times n$, la matriz DFT unitaria bidimensional se construye como el producto de Kronecker de la DFT unitaria unidimensional

$$\mathbf{F} = \mathbf{F}_n \otimes \mathbf{F}_n.$$

Esta matriz actúa sobre la versión vectorizada de la imagen, $\mathrm{vec}(\mathbf{X})$, donde $\mathbf{X} \in \mathbb{C}^{n\times n}$. Así, se cumple que

$$\mathrm{vec}(\widehat{\mathbf{X}}) = (\mathbf{F}_n \otimes \mathbf{F}_n) \cdot \mathrm{vec}(\mathbf{X}).$$

Esta matriz tiene una propiedad muy conveniente: se puede utilizar un método de *divide & conquer* para realizar multiplicaciones matriz-vector con $\mathbf{F}$ y $\mathbf{F}^*$, sin construir $\mathbf{F}$ explícitamente, utilizando transformadas rápidas de Fourier (FFT, por sus siglas en inglés). Adicionalmente, se puede asumir que $\mathbf{L}$ también tiene estructura de BCCB, para obtener (3.1).

Usando la diagonalización conjunta de $\mathbf{A}$ y $\mathbf{L}$ dada por (3.1), obtenemos las siguientes expresiones de $\mathbf{K}^\dagger$ y $\mathcal{P}_{\mathbf{K}}^{\perp}$ [6]:

$$\mathbf{K}^\dagger = \mathbf{Q}^* \left(\mathbf{C}^*\mathbf{C} + \lambda^2\mathbf{S}^*\mathbf{S}\right)^{-1} \left[\mathbf{C}^*\mathbf{Q}, \quad \lambda\mathbf{S}^*\mathbf{Q}\right]$$

$$\mathcal{P}_{\mathbf{K}}^{\perp} = \mathbf{I} - \begin{bmatrix} \mathbf{Q}^*\mathbf{C} \\ \lambda\mathbf{Q}^*\mathbf{S} \end{bmatrix} \left(\mathbf{C}^*\mathbf{C} + \lambda^2\mathbf{S}^*\mathbf{S}\right)^{-1} \left[\mathbf{C}^*\mathbf{Q}, \quad \lambda\mathbf{S}^*\mathbf{Q}\right],$$

y por lo tanto, cada término en la j-ésima columna del Jacobiano (2.13) puede calcularse muy eficientemente como sigue:

$$\mathbf{A}_j = \left(\mathbf{I} - \begin{bmatrix}\mathbf{Q}^*\mathbf{D_C}\\ \lambda\mathbf{Q}^*\mathbf{D_S}\end{bmatrix}\left[\mathbf{C}^*\mathbf{Q}, \quad \lambda\mathbf{S}^*\mathbf{Q}\right]\right)\begin{bmatrix}\frac{\partial\mathbf{A}}{\partial\mathbf{y}_j}\\ \mathbf{0}\end{bmatrix}\mathbf{Q}^*\mathbf{D_C}\mathbf{Q}\mathbf{d}$$

$$\mathbf{B}_j = \begin{bmatrix}\mathbf{Q}^*\mathbf{D_C}\\ \lambda\mathbf{Q}^*\mathbf{D_S}\end{bmatrix}\mathbf{Q}\frac{\partial\mathbf{A}^\top}{\partial\mathbf{y}_j}\left(\mathbf{d} - \mathbf{Q}^*\mathbf{D_C}\mathbf{C}^*\mathbf{Q}\mathbf{d}\right),$$

donde $\mathbf{D_C}$ y $\mathbf{D_S}$ son las matrices diagonales definidas por $\mathbf{D_C} = \mathbf{C}\left(\mathbf{C}^*\mathbf{C} + \lambda^2\mathbf{S}^*\mathbf{S}\right)^{-1}$ y $\mathbf{D_S} = \mathbf{S}\left(\mathbf{C}^*\mathbf{C} + \lambda^2\mathbf{S}^*\mathbf{S}\right)^{-1}$.

En este caso, la expresión para el Jacobiano RJF (2.15) toma la forma:

$$\frac{\partial\mathbf{f}(\mathbf{y})}{\partial\mathbf{y}_j} \approx -\begin{bmatrix}\frac{\partial\mathbf{A}}{\partial\mathbf{y}_j}\\ \mathbf{0}\end{bmatrix}\mathbf{Q}^*\mathbf{D_C}\mathbf{Q}\mathbf{d}.$$

4. EJEMPLOS NUMÉRICOS

En este capítulo presentamos diversos ejemplos numéricos con el objetivo de evaluar el rendimiento de los métodos desarrollados en los capítulos anteriores. Nos enfocaremos en problemas de deconvolución semiciegos bidimensionales, es decir, de restauración de imágenes desenfocadas, donde tanto la imagen original como el parámetro de desenfoque deben ser estimados simultáneamente. En todos los casos considerados, asumimos que las matrices $\mathbf{A}$ (operador de desenfoque) y $\mathbf{L}$ (matriz de regularización) admiten una diagonalización conjunta, lo que permite explotar eficientemente su estructura en la implementación de los algoritmos.

Primero vamos a analizar el desempeño del Algoritmo 1, evaluando el efecto de utilizar diferentes expresiones para la matriz Jacobiana del modelo: GP (Golub–Pereyra), KAUF (Kaufman) y RJF (forma reducida). Luego, consideramos el escenario inexacto, en el que la variable $\mathbf{x}$ es aproximada en cada iteración k mediante el método iterativo LSQR, de acuerdo al Algoritmo 2. En este caso también comparamos las tres formas de la matriz Jacobiana previamente mencionadas.

Todas los experimentos fueron realizados utilizando Google Colaboratory (Colab) como entorno de ejecución. Este entorno se basa en el sistema operativo **Ubuntu 22.04.4 LTS (Jammy Jellyfish)** y proporciona una máquina virtual configurada con los siguientes recursos de hardware:

- **Procesador**: Intel(R) Xeon(R) CPU de doble núcleo a 2.20 GHz
- **Arquitectura**: x86_64 (64 bits)
- **Memoria RAM disponible**: aproximadamente 12 GB
- **Almacenamiento**: sistema de archivos con capacidad aproximada de 108 GB, de los cuales unos 71 GB estaban disponibles durante la ejecución
- **Virtualización**: basada en hipervisor KVM

Los experimentos se ejecutaron sin aceleración por GPU, utilizando únicamente los recursos de CPU proporcionados por el entorno virtual. El código fue implementado en **Python 3.10**, empleando bibliotecas del ecosistema científico estándar, tales como **NumPy**, **SciPy** y **Matplotlib**.

Para evaluar la calidad de las soluciones reconstruidas en cada iteración k, calculamos el error de reconstrucción relativo (RRE, por sus siglas en inglés) definido por

$$\mathrm{RRE}(\mathbf{x}^{(k)}) = \frac{\|\mathbf{x}^{(k)} - \mathbf{x}_{\mathrm{true}}\|_2}{\|\mathbf{x}_{\mathrm{true}}\|_2}.$$

Análogamente, para cuantificar la precisión en la estimación del parámetro de desenfoque $\mathbf{y}$, se empleó el RRE definido por

$$\mathrm{RRE}(\mathbf{y}^{(k)}) = \frac{\|\mathbf{y}^{(k)} - \mathbf{y}_{\mathrm{true}}\|_2}{\|\mathbf{y}_{\mathrm{true}}\|_2}.$$

Además del cálculo de los errores relativos, se incluye una inspección visual de las imágenes reconstruidas, así como el análisis de las métricas de calidad, los tiempos de ejecución y la precisión en la recuperación de los parámetros verdaderos.

Finalmente, cabe destacar que en todos los experimentos los parámetros de regularización λ y μ fueron mantenidos fijos a lo largo de las iteraciones.

4.1. Problema de deconvolución semiciego

Consideramos el problema de restauración de imágenes contaminadas por desenfoque y ruido. El sistema

$$\mathbf{A}(\mathbf{y})\mathbf{x} \approx \mathbf{b}$$

representa el modelo matemático que describe el proceso de desenfoque, donde $\mathbf{x}$ denota el vector que representa la imagen verdadera (desconocida) que pretendemos reconstruir, $\mathbf{A}$ es una matriz mal condicionada que describe el operador de desenfoque asociado a la función de dispersión del punto (PSF, por sus siglas en inglés), que se define por el parámetro $\mathbf{y}$, y $\mathbf{b}$ es el vector que representa la imagen desenfocada y ruidosa.

La PSF describe cómo se distribuye la intensidad luminosa de un punto en los píxeles vecinos al píxel (k, ℓ), y se representa por una matriz $\mathbf{P}(\mathbf{y}) \in \mathbb{R}^{n\times n}$ dependiente del parámetro $\mathbf{y}$. Por tanto, la matriz de desenfoque se puede expresar como $\mathbf{A}(\mathbf{y}) = \mathbf{A}(\mathbf{P}(\mathbf{y}))$. En este ejemplo, consideramos una PSF modelada por una función Gaussiana bidimensional

$$[\mathbf{P}(\mathbf{y})]_{i,j} = c\exp\left(-\frac{1}{2}\begin{bmatrix} i-k \\ j-\ell \end{bmatrix}^T \begin{bmatrix} \sigma_1^2 & \rho^2 \\ \rho^2 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-\ell \end{bmatrix}\right),$$

donde $\mathbf{y} = (\sigma_1, \sigma_2, \rho)$ es el vector de parámetros que controla la forma y orientación de la dispersión, sujeto a la condición $\sigma_1^2\sigma_2^2 - \rho^4 > 0$ para garantizar la positividad definida de la matriz de covarianza. La constante c se elige de modo que $\sum_{i,j}[\mathbf{P}(\mathbf{y})]_{i,j} = 1$.

Cabe destacar que no es necesario construir explícitamente la matriz $\mathbf{A}$, ya que el producto $\mathbf{Ax}$ puede evaluarse directamente a través de convoluciones implementadas mediante $\mathbf{P}$. Para aplicar los algoritmos de optimización, se requiere el cálculo de la matriz Jacobiana de $\mathbf{A}(\mathbf{y})\mathbf{x}$ respecto a $\mathbf{y}$. Aplicando la regla de la cadena, se tiene que

$$\frac{\partial}{\partial y_j}\left[\mathbf{A}(\mathbf{P}(\mathbf{y}))\mathbf{x}\right] = \mathbf{A}\left(\frac{\partial}{\partial y_j}\left[\mathbf{P}(\mathbf{y})\right]\right)\mathbf{x}.$$

Suponemos además que la imagen $\mathbf{x}$ tiene condiciones de borde periódicas, lo que resulta en que la matriz $\mathbf{A}(\mathbf{P}(\mathbf{y}))$ sea diagonalizable mediante la transformada discreta de Fourier (DFT, por sus siglas en inglés) $\mathbf{Q}$, es decir, $\mathbf{A}(\mathbf{P}(\mathbf{y})) = \mathbf{Q}^*\mathbf{C}\mathbf{Q}$, donde $\mathbf{C}$ es una matriz diagonal que contiene los autovalores de $\mathbf{A}(\mathbf{P}(\mathbf{y}))$ (ver sección 3.3). Por lo tanto, podemos utlizar la transformada rápida de Fourier (FFT, por sus siglas en inglés) y su inversa para calcular soluciones aproximadas de manera eficiente. Mediante cálculos sencillos, se puede demostrar que

$$\frac{\partial}{\partial y_j}\left[\mathbf{A}(\mathbf{P}(\mathbf{y}))\mathbf{x}\right] = \mathbf{Q}^*\mathbf{C}_p\mathbf{Q}\mathbf{x},$$

donde $\mathbf{C}_p$ es una matriz diagonal cuyas entradas se obtienen aplicando la FFT a la derivada parcial de $\mathbf{P}(\mathbf{y})$ respecto a y_j [14].

En los experimentos considerados, se utilizó una imagen original de tamaño 512×512, la cual se presenta en la Figura 4.1a. Esta imagen fue desenfocada aplicando una PSF Gaussiana con parámetros $\sigma_1 = \sigma_2 = 3$ y $\rho = 0$. Posteriormente, añadimos un 5 % de ruido Gaussiano. La imagen desenfocada resultante se puede observar en (4.1b). En este caso particular fijamos entonces $\rho = 0$ y consideramos una única variable $\mathbf{y} = \sigma_1 = \sigma_2$, es decir, $\mathbf{y}_{\text{true}} = 3$. La matriz $\mathbf{P}$ de la PSF toma la forma

$$[\mathbf{P}(\mathbf{y})]_{i,j} = c\exp\left(-\frac{1}{2}\left(\frac{i-k}{y}\right)^2 - \frac{1}{2}\left(\frac{j-l}{y}\right)^2\right).$$

(a) Imagen original

(b) Imagen desenfocafa

Fig. 4.1: Comparación de imágenes: (a) La solución verdadera $\mathbf{x}_{\text{true}}$ y (b) los datos borrosos y con ruido $\mathbf{b}$.

Consideramos la plantilla

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

que se utiliza para generar la matriz de regularización Laplaciana discreta $\mathbf{L}$. Dado que asumimos condiciones de borde periódicas en $\mathbf{x}$, la matriz $\mathbf{L}$ correspondiente también es diagonalizable por $\mathbf{Q}$. Por lo tanto, podemos utilizar directamente las formulaciones presentadas en la sección 3.3 para problemas con matrices diagonalizables conjuntamente. En este problema particular, el único parámetro de desenfoque que debemos estimar es $\mathbf{y} = \sigma_1$, por lo que se establece una condición inicial $\mathbf{y}^{(0)} = 2$.

4.2. Implementación de GenVarPro Regularizado

En esta sección analizamos los resultados obtenidos por el Algoritmo 1 para la tres variantes de Jacobiano, usando los dos funcionales de regularización que definimos en la sección 2.3.

4.2.1. Regularización vía norma 2

Comenzamos analizando el caso particular en el que el funcional de regularización $\mathcal{R}(\mathbf{y})$ se define como $\mathcal{R}(\mathbf{y}) = \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|_2^2$, con $\mathbf{y}_0 = 3{,}8$, un valor cercano al valor verdadero $\mathbf{y}_{\text{true}} = 3$. Como mencionamos previamente, la estimación inicial utilizada es $\mathbf{y}^{(0)} = 2$. En este caso, los parámetros de regularización se fijan en $\lambda = 1{,}5$ y $\mu = 3{,}8$ para todas las iteraciones, en los tres casos de Jacobiano evaluados.

La Figura 4.2 muestra las imágenes reconstruidas obtenidas al aplicar el algoritmo GenVarPro Regularizado. Se observa que las soluciones obtenidas con los tres tipos de Jacobiano proveen soluciones similares.

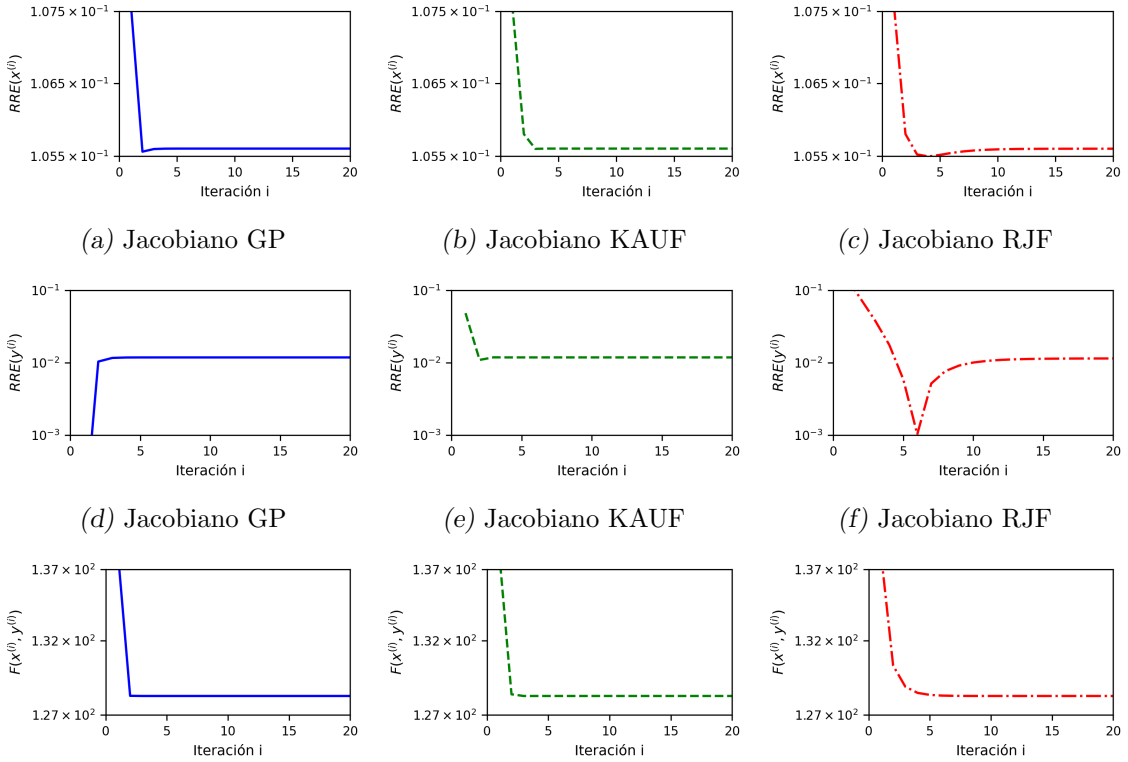
(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

Fig. 4.2: Soluciones de Tikhonov reconstruidas $\mathbf{x}(\mathbf{y})$ cuando se conoce $\mathbf{y}$ para los tres Jacobianos: (a) Golub-Pereyra, (b) Kaufman, (c) Ruano-Jones-Fleming, obtenidas por GenVarPro Regularizado para el caso de regularización vía norma 2.

La Figura 4.3 presenta las curvas de convergencia correspondientes a la aplicación del algoritmo GenVarPro Regularizado con los distintos Jacobianos. En todos los casos, los errores relativos de reconstrucción (RRE($\mathbf{x}^{(k)}$) y RRE($\mathbf{y}^{(k)}$)) tienden a estabilizarse en niveles similares. Para los Jacobianos GP y KAUF la cotas de RRE($\mathbf{y}^{(k)}$) se alcanzan aproximadamente en la quinta iteración, mientras que en el caso del Jacobiano RJF existe una semi-convergencia y se requieren alrededor de diez iteraciones para que la cota se alcance. Esta diferencia puede atribuirse a que RJF es una aproximación más simple y menos precisa del Jacobiano completo, lo cual se refleja en una menor eficiencia del algoritmo, requiriendo más iteraciones para alcanzar el mismo nivel de precisión.

En la Tabla 4.1 se presentan los valores estimados del parámetro $\mathbf{y}$, los cuales son muy cercanos al valor original, junto con los tiempos de cómputo (en segundos). Además, se incluye el valor de medida del índice de similitud estructural (SSIM, por sus siglas en inglés), una métrica que evalúa la calidad de la imagen reconstruida con respecto a la imagen original de referencia. Un valor más cercano a 1 indica una mejor calidad de imagen. En este caso, los tres Jacobianos generan valores del índice similares, lo cual justifica que las imágenes reconstruidas mostradas en la Figura 4.2 sean similares entre sí.

Finalmente, para completar el análisis de este caso, en la Figura 4.4 se ilustra la importancia de la regularización en $\mathbf{y}$. Comparamos los resultados obtenidos mediante nuestro enfoque con los obtenidos al omitir el término de regularización en $\mathbf{y}$. Sin este término, obtenemos la solución trivial $\mathbf{y} = 0$, la función objetivo presenta un mínimo local en $\mathbf{y} = 0$. En contraste, el enfoque regularizado logra identificar un mínimo local cercano al valor real, lo cual valida la efectividad del método propuesto.



(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

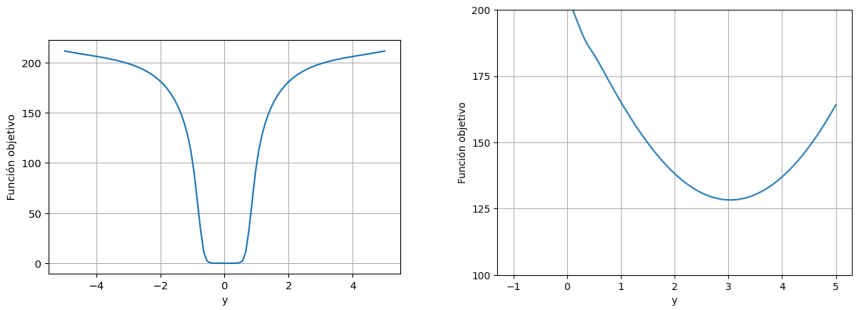
(d) Jacobiano GP *(e)* Jacobiano KAUF *(f)* Jacobiano RJF

(g) Jacobiano GP *(h)* Jacobiano KAUF *(i)* Jacobiano RJF

Fig. 4.3: Curvas de convergencia del método GenVarPro Regularizado para el caso de regularización vía norma 2, utilizando el mismo valor de λ y μ para todas las iteraciones y diferentes Jacobianos: GP (línea azul), KAUF (línea con guiones verde) y RJF (línea con guiones y puntos roja). La fila superior contiene los RRE de las soluciones reconstruidas, RRE($\mathbf{x}^{(k)}$). La fila central contiene RRE del parámetro reconstruido $\mathbf{y}$, RRE($\mathbf{y}^{(k)}$). La fila inferior contiene el valor del funcional en cada iteración, $\mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$. Cada columna corresponde a un Jacobiano diferente.

Tab. 4.1: Comparación de los valores obtenidos del índice SSIM, el $\mathbf{y}$ estimado y el tiempo de ejecución para los tres Jacobianos con GenVarPro Regularizado para el caso de regularización vía norma 2.

	SSIM	$\mathbf{y}_{\text{est}}$	**CPU time (s)**
GP	0.668	3.036	4.19
KAUF	0.668	3.036	4.84
RJF	0.668	3.035	4.42



(a) Sin $\mathcal{R}(\mathbf{y})$ *(b)* Con $\mathcal{R}(\mathbf{y})$

Fig. 4.4: Función objetivo en función de $\mathbf{y}$: (a) resultados preexistentes, (b) nuevos resultados obtenidos por GenVarPro Regularizado para el caso de regularización vía norma 2.

4.2.2. Regularización vía logaritmos

Consideramos ahora el caso en que la función de regularización es $\mathcal{R}(\mathbf{y}) = -\sum_{1\leq j\leq r} \mu_j^2 \log(y_j)$. La estimación inicial sigue siendo $\mathbf{y}^{(0)} = 2$, pero en este caso los valores fijos utilizados para los parámetros λ y μ son 0,425 y 3,8, respectivamente, en los tres Jacobianos. Notar entonces que los valores de los parámetros van a depender del tipo de regularización que se utilice. La Figura 4.5 muestra las imágenes reconstruidas obtenidas, donde se observa que los tres Jacobianos generan resultados visualmente muy similares.

(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

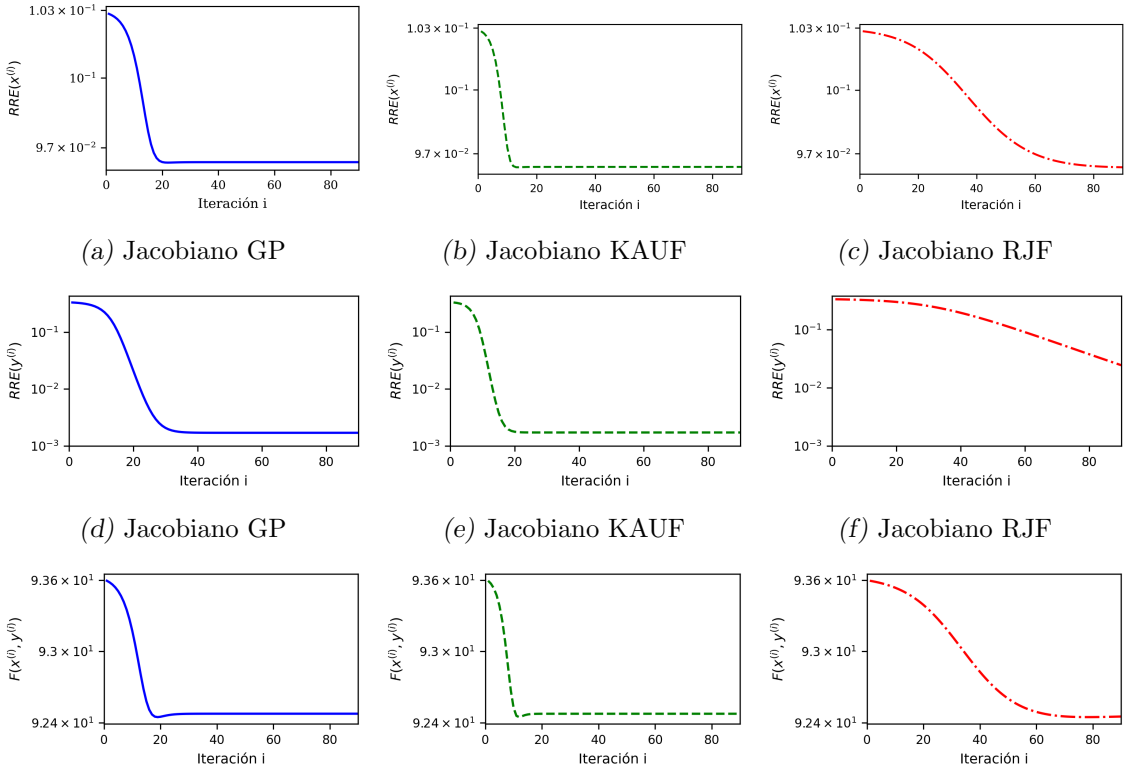
Fig. 4.5: Soluciones de Tikhonov reconstruidas $\mathbf{x}(\mathbf{y})$ cuando se conoce $\mathbf{y}$ para los tres Jacobianos: (a) Golub-Pereyra, (b) Kaufman, (c) Ruano-Jones-Fleming, obtenidas por GenVarPro Regularizado para el caso de regularización vía logaritmos.

La Figura 4.6 presenta las curvas de convergencia correspondientes a aplicar el método GenVarPro Regularizado con los distintos Jacobianos. En los tres casos se alcanzan cotas de error similares. A diferencia del caso anterior, se realizan 90 iteraciones, debido a que el cálculo asociado al Jacobiano RFJ es más lento. Aunque GP y KAUF alcanzan convergencia alrededor de la iteración 20, usamos el mismo número total de iteraciones para facilitar la comparación. Sus cotas de error se estabilizan rápidamente, mientras que en RFJ el error en $\mathbf{y}$ sigue disminuyendo más allá de la iteración 90, lo que sugiere que podría beneficiarse de más iteraciones.

La Tabla 4.2 muestra los valores del índice SSIM, las estimaciones de $\mathbf{y}$, muy cercanos al valor de $\mathbf{y}$ real, y los tiempos de cómputo en segundos. Nuevamente, los tres Jacobianos arrojan valores

similares del índice SSIM, lo cual respalda la similitud visual observada en las imágenes de la Figura 4.5.

Siguiendo el mismo análisis realizado en la sección anterior, la Figura 4.7 muestra el gráfico de la función objetivo del problema en función de $\mathbf{y}$. Se observa que, sin el término de regularización $\mathcal{R}(\mathbf{y})$, el modelo converge a la solución trivial $\mathbf{y} = 0$. En cambio, la inclusión del término logarítmico permite alcanzar un mínimo local cercano al valor verdadero $\mathbf{y}_{\text{true}} = 3$.



(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

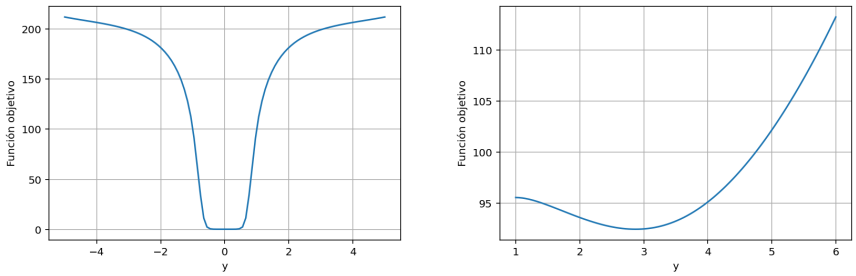
(d) Jacobiano GP *(e)* Jacobiano KAUF *(f)* Jacobiano RJF

(g) Jacobiano GP *(h)* Jacobiano KAUF *(i)* Jacobiano RJF

Fig. 4.6: Curvas de convergencia del método GenVarPro Regularizado para el caso de regularización vía logaritmos, utilizando el mismo valor de λ y μ para todas las iteraciones y diferentes Jacobianos: GP (línea azul), KAUF (línea con guiones verde) y RJF (línea con guiones y puntos roja). La fila superior contiene los RRE de las soluciones reconstruidas, RRE($\mathbf{x}^{(k)}$). La fila central contiene RRE del parámetro reconstruido $\mathbf{y}$, RRE($\mathbf{y}^{(k)}$). La fila inferior contiene el valor del funcional en cada iteración, $\mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$. Cada columna corresponde a un Jacobiano diferente.

Tab. 4.2: Comparación de los valores obtenidos del índice SSIM, el $\mathbf{y}$ estimado y el tiempo de ejecución para los tres Jacobianos con GenVarPro Regularizado para el caso de regularización vía logaritmos.

	SSIM	$\mathbf{y}_{\text{est}}$	**CPU time (s)**
GP	0.634	2.995	20.29
KAUF	0.634	2.995	20.86
RJF	0.635	2.927	17.83



(a) Sin $\mathcal{R}(\mathbf{y})$ *(b)* Con $\mathcal{R}(\mathbf{y})$

Fig. 4.7: Función objetivo en función de $\mathbf{y}$: (a) resultados preexistentes, (b) nuevos resultados obtenidos por GenVarPro Regularizado para el caso de regularización vía logaritmos.

4.3. Implementación de GenVarPro Regularizado Inexacto

En esta sección analizamos los resultados obtenidos mediante el Algoritmo 2 para la tres variantes de Jacobiano, utilizando los dos parámetros de regularización definidos en (2.3).

4.3.1. Regularización vía norma 2

Al igual que el caso exacto, comenzamos con el escenario donde $\mathcal{R}(\mathbf{y}) = \frac{\mu^2}{2}\|\mathbf{y} - \mathbf{y}_0\|_2^2$, con $\mathbf{y}_0 = 3{,}8$. La estimación inicial es $\mathbf{y}^{(0)} = 2$, y se fijan los valores de $\lambda = 1{,}5$ y $\mu = 3{,}8$ para los tres Jacobianos. Las tolerancias utilizadas en el método iterativo LSQR son `atol` $= 10^{-2}$, `btol` $= 0$ y `conlim` $= 10^8$ (valor por default). La Figura 4.8 muestra las imágenes reconstruidas, donde se observa que los tres enfoques generan resultados similares.

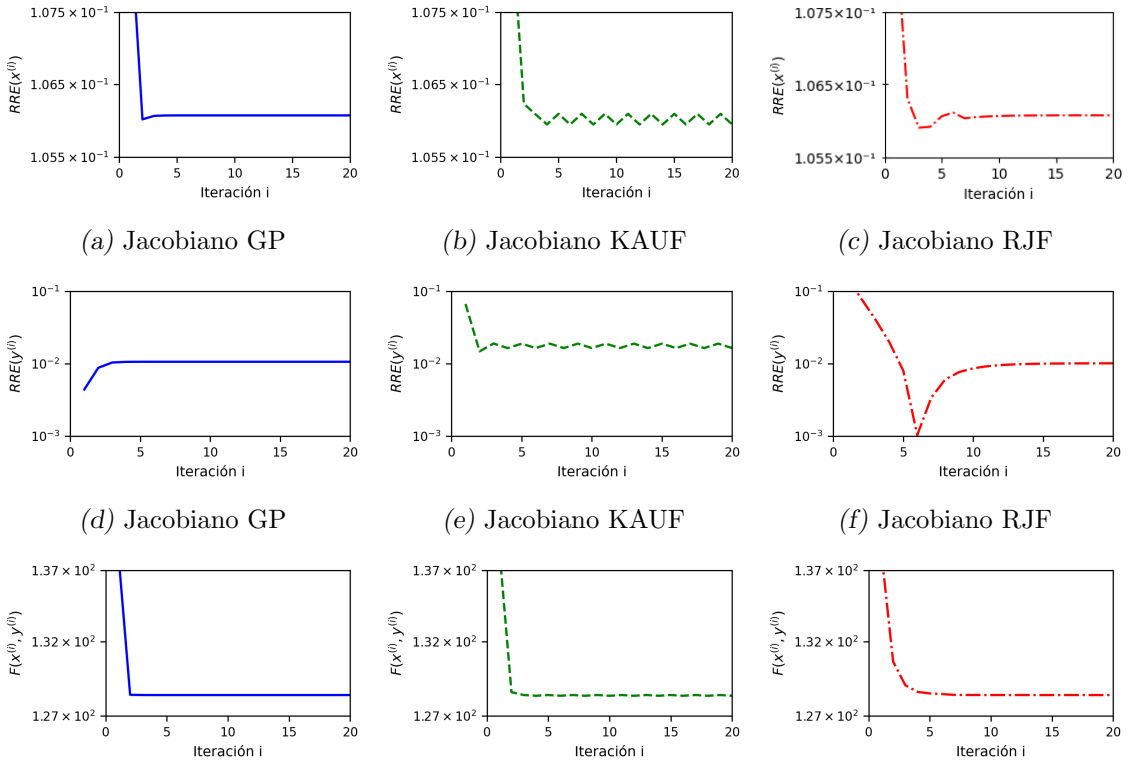
(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

Fig. 4.8: Soluciones de Tikhonov reconstruidas $\mathbf{x}(\mathbf{y})$ cuando se conoce $\mathbf{y}$ para los tres Jacobianos: (a) Golub-Pereyra, (b) Kaufman, (c) Ruano-Jones-Fleming, obtenidas por GenVarPro Regularizado Inexacto para el caso de regularización vía norma 2.

La Figura 4.9 presenta las curvas de convergencia obtenidas al aplicar el método GenVarPro Regularizado Inexacto con los distintos Jacobianos. En los tres casos se alcanzan cotas de errores similares. Sin embargo, con el Jacobiano KAUF se evidencia un comportamiento oscilatorio en las curvas de RRE($\mathbf{x}^{(k)}$) y RRE($\mathbf{y}^{(k)}$). Este fenómeno se debe a que el algoritmo no siempre

avanza consistentemente en la dirección del mínimo de la función objetivo. Para mitigar este comportamiento, trabajos futuros podrían incorporar un hiperparámetro de control α que ajuste el paso en cada iteración y garantice que el algoritmo descienda en la dirección del mínimo. También se observa nuevamente un fenómeno de semiconvergencia en RRE($\mathbf{y}^{(k)}$).

La Tabla 4.3 resume los valores del índice SSIM, los valores estimados de $\mathbf{y}$, muy cercanos al valor original, y los tiempos de cómputo. Estos tiempos son mayores en comparación con el método GenVarPro Regularizado exacto, debido a que en cada iteración se calcula una aproximación de $\mathbf{x}^{(k)}$ mediante LSQR, incrementando el costo computacional. Los tres Jacobianos proporcionan valores similares del índice SSIM, lo cual justifica la similitud entre las imágenes reconstruidas presentadas en la Figura 4.8.



(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

(d) Jacobiano GP *(e)* Jacobiano KAUF *(f)* Jacobiano RJF

(g) Jacobiano GP *(h)* Jacobiano KAUF *(i)* Jacobiano RJF

Fig. 4.9: Curvas de convergencia del método GenVarPro Regularizado Inexacto para el caso de regularización vía norma 2, utilizando el mismo valor de λ y μ para todas las iteraciones y diferentes Jacobianos: GP (línea azul), KAUF (línea con guiones verde) y RJF (línea con guiones y puntos roja). La fila superior contiene los RRE de las soluciones reconstruidas, RRE($\mathbf{x}^{(k)}$). La fila central contiene RRE del parámetro reconstruido $\mathbf{y}$, RRE($\mathbf{y}^{(k)}$). La fila inferior contiene el valor del funcional en cada iteración, $\mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$. Cada columna corresponde a un Jacobiano diferente.

Tab. 4.3: Comparación de los valores obtenidos del índice SSIM, el $\mathbf{y}$ estimado y el tiempo de ejecución para los tres Jacobianos con GenVarPro Regularizado Inexacto para el caso de regularización vía norma 2.

	SSIM	$\mathbf{y}_{\text{est}}$	**CPU time (s)**
GP	0.667	3.032	16.62
KAUF	0.667	3.049	17.71
RJF	0.667	3.030	16.71

4.3.2. Regularización vía logaritmos

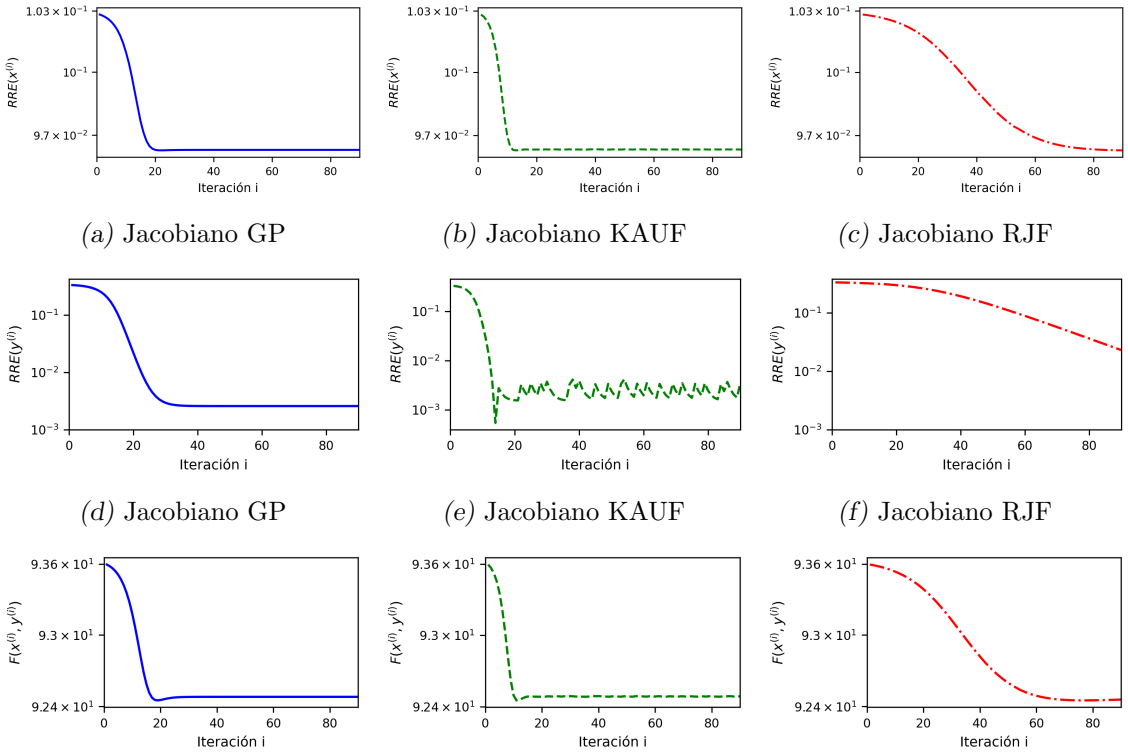
Concluimos este análisis numérico considerando el caso donde $\mathcal{R}(\mathbf{y}) = -\sum_{1\leq j\leq r} \mu_j^2 \log(y_j)$. Se parte de la estimación inicial $\mathbf{y}^{(0)} = 2$, y se utilizan los valores $\lambda = 0{,}425$ y $\mu = 3{,}8$ para todos los Jacobianos. Las tolerancias utilizadas en el método iterativo LSQR son `atol` $= 10^{-3}$, `btol` $= 0$ y `conlim` $= 10^8$ (valor por default), lo que permite mayor precisión en la solución del sistema inexacto. La Figura 4.10 muestra las imágenes reconstruidas obtenidas. Vemos que los tres Jacobianos proveen soluciones similares.

(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

Fig. 4.10: Soluciones de Tikhonov reconstruidas $\mathbf{x}(\mathbf{y})$ cuando se conoce $\mathbf{y}$ para los tres Jacobianos: (a) Golub-Pereyra, (b) Kaufman, (c) Ruano-Jones-Fleming, obtenidas por GenVarPro Regularizado Inexacto para el caso de regularización vía logaritmos.

La Figura 4.11 muestra las curvas de convergencia correspondientes a aplicar GenVarPro Regularizado Inexacto con los diferentes Jacobianos. En los tres casos, los errores se estabilizan en niveles comparables. Al igual que en el caso exacto, se realizan 90 iteraciones para observar el comportamiento a largo plazo. Para los Jacobianos GP y KAUF, los errores se estabilizan hacia la iteración 20, mientras que en el caso RJF, la curva de error en $\mathbf{y}$ indica que aún podría mejorar si se permitiera un mayor número de iteraciones. También se observa que el Jacobiano KAUF repite el patrón oscilatorio en $\text{RRE}(\mathbf{y}^{(k)})$, el cual podría corregirse con un parámetro α como se discutió previamente.

En la Tabla 4.4 se presentan los valores del índice SSIM, los $\mathbf{y}$ obtenidos, nuevamente cercanos al valor de $\mathbf{y}$ original, y los tiempos de cómputo en segundos. Como es de esperarse, los tiempos son más elevados que en la versión exacta del algoritmo debido al uso de LSQR en cada iteración. En este caso, los tres Jacobianos generan valores del índice similares, esto justifica por qué las imágenes reconstruidas en la Figura 4.10 son parecidas.



(a) Jacobiano GP *(b)* Jacobiano KAUF *(c)* Jacobiano RJF

(d) Jacobiano GP *(e)* Jacobiano KAUF *(f)* Jacobiano RJF

(g) Jacobiano GP *(h)* Jacobiano KAUF *(i)* Jacobiano RJF

Fig. 4.11: Curvas de convergencia del método GenVarPro Regularizado Inexacto para el caso de regularización vía logaritmos, utilizando el mismo valor de λ y μ para todas las iteraciones y diferentes Jacobianos: GP (línea azul), KAUF (línea con guiones verde) y RJF (línea con guiones y puntos roja). La fila superior contiene los RRE de las soluciones reconstruidas, RRE($\mathbf{x}^{(k)}$). La fila central contiene RRE del parámetro reconstruido $\mathbf{y}$, RRE($\mathbf{y}^{(k)}$). La fila inferior contiene el valor del funcional en cada iteración, $\mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$. Cada columna corresponde a un Jacobiano diferente.

Tab. 4.4: Comparación de los valores obtenidos del índice SSIM, el $\mathbf{y}$ estimado y el tiempo de ejecución para los tres Jacobianos con GenVarPro Regularizado Inexacto para el caso de regularización vía logaritmos.

	SSIM	$\mathbf{y}_{\text{est}}$	**CPU time (s)**
GP	0.636	2.99	129.20
KAUF	0.636	3.01	123.06
RJF	0.637	2.93	105.61

4.4. Comparación de resultados

En esta sección resumimos y comparamos los resultados obtenidos al aplicar los Algoritmos 1 y 2. El objetivo es comparar el desempeño tanto en términos de calidad de reconstrucción como de robustez frente a errores y eficiencia computacional.

4.4.1. Resultados GenVarPro Regularizado

Regularización cuadrática: Esta elección impone una penalización simétrica y estrictamente convexa alrededor del valor inicial $\mathbf{y}_0$, favoreciendo soluciones cercanas a este punto. En las pruebas realizadas, esta regularización condujo a una estimación del parámetro $\mathbf{y}$ relativamente estable. Las imágenes reconstruidas fueron suaves, con una ligera pérdida de detalles finos.

Regularización logarítmica: Esta forma de penalización introdujo un sesgo más adaptativo, penalizando con mayor intensidad desviaciones en direcciones específicas. Como resultado, la estimación de $\mathbf{y}$ fue más precisa en los experimentos realizados, logrando converger más cerca del valor real incluso en presencia de ruido. Las reconstrucciones presentaron mejor preservación de bordes y estructuras, aunque el método fue más sensible a la elección de λ, además de generar un costo computacional mucho mayor.

4.4.2. Resultados GenVarPro Regularizado Inexacto

Regularización cuadrática: Al igual que en el caso exacto, esta regularización proporcionó un comportamiento estable y controlado para $\mathbf{y}$. La calidad de la reconstrucción fue similar al caso exacto.

Regularización logarítmica: El uso de esta regularización en el esquema inexacto resultó ser más costosa computacionalmente. Las reconstrucciones obtenidas conservaron mejor los detalles, aunque el método fue más sensible a perturbaciones numéricas y a la calidad de la aproximación de $\mathbf{x}$ en cada iteración.

En resumen, la regularización logarítmica sobre $\mathbf{y}$ ofreció mejores resultados en términos de fidelidad de la reconstrucción y precisión en la estimación del desenfoque, a costa de una mayor sensibilidad y tiempo computacional. La regularización cuadrática, en cambio, fue más robusta y sencilla de implementar, pero con un sesgo más fuerte hacia el valor inicial $\mathbf{y}_0$.

5. CONCLUSIONES

En esta tesis abordamos el problema de la deconvolución semiciega de imágenes desenfocadas mediante la extensión del método de proyección de variables (VarPro). Propusimos una generalización de este enfoque para tratar problemas inversos no lineales separables con regularización de Tikhonov en forma general, incorporando términos regularizantes sobre los parámetros del operador de desenfoque. En particular, desarrollamos variantes del método que permitieron introducir dos tipos de regularización sobre dichos parámetros: una penalización cuadrática y una penalización logarítmica.

Asimismo, analizamos el impacto de distintas elecciones para el cálculo del Jacobiano, y evaluamos dos estrategias para resolver el subproblema interno: mediante soluciones exactas y utilizando el método iterativo LSQR para obtener soluciones aproximadas. Para mejorar la eficiencia computacional, implementamos técnicas basadas en descomposiciones espectrales que permitieron calcular de forma eficiente tanto el Jacobiano como el residuo en problemas con operadores estructurados.

Presentamos un análisis de convergencia local del algoritmo VarPro regularizado, en el caso en que se utiliza un Jacobiano de tipo Golub-Pereyra y se resuelve de forma exacta el subproblema interno. Bajo hipótesis de diferenciabilidad y continuidad Lipschitz de los operadores involucrados, demostramos que bajo condiciones adecuadas, el algoritmo puede alcanzar tasas de convergencia superlineales e incluso cuadráticas. Esta contribución amplía el marco teórico existente y proporciona fundamentos sólidos para el análisis y la aplicación de métodos de proyección en problemas inversos regularizados.

Finalmente, llevamos a cabo experimentos numéricos aplicando las variantes propuestas al problema de deconvolución semiciega, con el objetivo de evaluar la calidad de reconstrucción y la velocidad de convergencia de los métodos. Implementamos los algoritmos en entornos sintéticos utilizando Google Colaboratory. Los resultados obtenidos muestran que el enfoque propuesto permite estimar de forma conjunta tanto la imagen original como los parámetros del desenfoque, y que la regularización adicional sobre estos últimos contribuye significativamente a mejorar la estabilidad frente al ruido y al mal condicionamiento del operador. En particular, observamos que una elección adecuada del término de regularización es crucial para garantizar la precisión de la reconstrucción. Además, la similitud en los errores relativos entre las distintas variantes sugiere que el algoritmo es robusto y confiable. Cabe señalar que en este trabajo asumimos que los parámetros de regularización son conocidos y se mantienen fijos a lo largo de las iteraciones. Según nuestra experiencia, esta elección conduce a buenas tasas de convergencia; no obstante, determinar con antelación valores óptimos puede resultar difícil en la práctica.

En conjunto, este trabajo aporta herramientas teóricas y computacionales para el tratamiento robusto de problemas de restauración de imágenes desenfocadas, integrando técnicas de optimización no lineal, regularización y álgebra lineal numérica.

5.1. Trabajos a futuro

Como líneas futuras de trabajo, proponemos investigar esquemas adaptativos para la selección automática de los parámetros de regularización, con el fin de optimizar la calidad de la reconstrucción sin necesidad de ajuste manual. Asimismo, resulta relevante extender los métodos desarrollados

a problemas tridimensionales y a operadores más complejos, propios de aplicaciones avanzadas en microscopía, astrofísica u otros campos.

En el caso particular de los métodos inexactos, es importante experimentar con nuevas estrategias para la elección de tolerancias en el cálculo de soluciones aproximadas, buscando un equilibrio entre eficiencia computacional y precisión.

Otra línea de estudio es la incorporación de un hiperparámetro en la iteración de los algoritmos, que garantice un descenso controlado del funcional objetivo, asegurando la convergencia y estabilidad del algoritmo durante la iteración.

Además, resulta de gran interés explorar estrategias avanzadas de precondicionamiento y paralelización, que permitan escalar los métodos propuestos para abordar problemas de gran dimensión y con alta complejidad computacional.

Finalmente, sigue siendo un desafío y una oportunidad profundizar en el análisis teórico de las variantes aproximadas del algoritmo, incluyendo aspectos de convergencia y comportamiento en presencia de ruido y regularización no estándar.

Bibliografía

[1] David Austin, Malena I. Español, and Mirjeta Pasha. The image deblurring problem: Matrices, wavelets, and multilevel methods. *Notices of the American Mathematical Society*, 69(8):1284–1293, 2022.

[2] Guangyong Chen, Peng Xue, Min Gan, Wenzhong Guo, Jing Chen, and C. L. Philip. Chen. Variable projection algorithms: Theoretical insights and a novel approach for problems with large residual. *arXiv preprint arXiv:2402.13865*, 2025.

[3] Julianne Chung and James G. Nagy. An efficient iterative approach for large-scale separable nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 31(6):4654–4674, 2010.

[4] Suchuan Dong and Jielin Yang. Numerical approximation of partial differential equations by a variable projection method with artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 398:115284, 2022.

[5] Malena I. Español and Gabriela Jeronimo. Local convergence analysis of a variable projection method for regularized separable nonlinear inverse problems. *SIAM Journal on Matrix Analysis and Applications*, 46(2), 2025.

[6] Malena I. Español and Mirjeta Pasha. Variable projection methods for separable nonlinear inverse problems with general-form tikhonov regularization. *Inverse Problems*, 39, 2023.

[7] Silvia Gazzola and Malena Sabaté Landman. Regularization by inexact Krylov methods with applications to blind deblurring. *SIAM Journal on Matrix Analysis and Applications*, 42(4):1528–1552, 2021.

[8] Gene H. Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973.

[9] Gene H. Golub and Victor Pereyra. Differentiation of pseudoinverses, separable nonlinear least square problems and other tales. *Generalized Inverse and Applications*, 1976.

[10] Gene H. Golub and Victor Pereyra. Separable nonlinear least squares: The variable projection method and its applications. *Inverse Problems*, 19(2):R1, 2003.

[11] Charles W. Groetsch. *The theory of Tikhonov regularization for Fredholm equations*, volume 104 of *Pitman Research Notes in Mathematics Series*. Pitman, Boston, 1984.

[12] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion.* SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[13] Per Christian Hansen. *Discrete inverse problems: insight and algorithms*, volume 7 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2010.

[14] Per Christian Hansen, James G. Nagy, and Dianne P. O'leary. *Deblurring images: Matrices, Spectra, and Filtering*, volume 3 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[15] Linda Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT Numererical Mathematics*, 15:49–57, 1975.

[16] Carl T. Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.

[17] Katherine M. Mullen, Mikas Vengris, and Ivo H. M. van Stokkum. Algorithms for separable nonlinear least squares with application to modelling time-resolved spectra. *Journal of Global Optimization*, 38(2):201–2013, 2007.

[18] Elizabeth Newman, Lars Ruthotto, Joseph Hart, and Bart van Bloemen Waanders. Train like a (var)pro: Efficient training of neural networks with variable projection. *SIAM Journal on Mathematics of Data Science*, 3(4):1041–1066, 2021.

[19] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York,NY, 2nd edition, 2006.

[20] Dianne P. O'Leary and Bert W. Rust. Variable projection for nonlinear least squares problems. *Computational Optimization and Applications*, 54(3):579–593, 2013.

[21] Christopher C. Paige and Michael A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):43–71, 1982.

[22] Victor Pereyra and Godela Scherer. Imaging applications with variable projections. *American Journal of Computational Mathematics*, 9(4):261–281, 2019.

[23] Antonio E. B. Ruano, Dewi I. Jones, and Peter J. Fleming. A new formulation of the learning problem of a neural network controller. In *Proceedings of the 30th IEEE Conference on Decision and Control*, number 865-866. IEEE, 1991.

[24] Axel Ruhe and Per Åke Wedin. Algorithms for separable nonlinear least squares problems. *Technical Report*, 1974.