UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

COMPUTACIÓN CIENTÍFICA

"Aproximación de datos dispersos con Series de Fourier y RBFs"

TESIS

QUE PARA OPTAR POR EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

URIEL OCTAVIO MORELES VÁZQUEZ

Director de Tesis:

DRA. SUSANA GÓMEZ GÓMEZ

INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

Ciudad Universitaria, Cd. Mx. Noviembre 2017

Para mi abuela

Agradecimientos

Agradezco a el CONACYT por las becas otorgadas para cubrir los costos de manutención de mi estancia en la ciudad de México y en Alemania por un periodo total de dos años. Agradezco también a mi asesor en la Universidad de Giessen, Alemania, el Prof. Dr. Oleg Davydov, por su excelente asesoría durante mi estancia en ese país. El Prof. Davydov me proporcionó referencias relevantes para completar éste trabajo y me ayudó con la implementación de los métodos descritos en la tesis.

A mi asesora en la UNAM, la Dra. Susana Gómez, quiero agradecerle por su curso de Cómputo Científico en el cual aprendí a paralelizar algorithmos usando la librería MPI. La Dra. Gómez también me dio acceso al cluster IIMAS para compilar el código Fortran del método propuesto en la tesis.

También me gustaría agradecer al Dr. Pedro González Casanova por su curso de Funciones de Base Radial(RBFs). Un compañero de ese curso, Rafael Miranda Cordero, ayudó a demostrar que el método propuesto está bien planteado en el sentido de la existencia y unicidad de la solución.

El Dr. Marcos Aurelio Capistrán Ocampo fue mi asesor de tesis de licenciatura. Él fue quien propuso usar Series de Fourier para un problema particular de aproximación. Trabajando con él me di cuenta que la metodología usada en ese problema se podía usar en problemas más generales. Una diferencia fundamental

entre ese trabajo y el presente es la incorporación del término de regularización también llamado funcional de energía. Dicho término cuantifica la curvatura o rugosidad de la superficie aproximante.

A mis compañeros computólogos me gustaría agradecerles por su disposición en responder preguntas de alguien con una formación académica diferente.

Finalmente, me gustaría mencionar que los profesores que han influido en mi formación me han colocado en una buena posición para entender los temas avanzados necesarios para completar éste trabajo.

Resumen

Un nuevo método de aproximación de datos dispersos es propuesto. El método se basa en la técnica de mínimos cuadrados con un término de regularización y usa series de Fourier. Se compara éste método con dos métodos recientes basados en series de Fourier, splines y RBFs. Se implementan experimentos numéricos con datos de referencia y datos reales.

Índice general

Índice de figuras

Índice de tablas

Siglas

BB Bernstein-Bézier. 16, 17, 18, 23

CONACYT Consejo Nacional de Ciencia y Tecnología. IV

FFT Transformada Rápida de Fourier. 6, 7

IIMAS Instituto de Investigaciones en Matemáticas y en Sistemas. IV

LHA Aproximación Local Híbrida. 30

MDS Conjunto Mínimo Determinante. 17, 21

MPI Message Passage Interface. IV, 46

NDFT Transformada Discreta de Fourier No Equiespaciada. 5, 7, 9

RBF Función de Base Radial. VI, 3, 4, 15, 26, 27, 32, 33, 34

SVD Descomposición en Valores Singulares. 24

UNAM Universidad Nacional Autónoma de México. IV

Introducción

El problema de aproximación de datos dispersos en dos dimensiones consiste en aproximar o reconstruir una función de la cual se conoce sólamente un número finito de valores en puntos dispersos de un dominio acotado Ω. Más precisamente, los datos de entrada del problema es un conjunto finito de puntos $\{(X_i, Y_i, Z_i)\}_{i=1}^{N}$ donde $(X_i, Y_i) \in \Omega \subset \mathbb{R}^2$ y Z_i son observaciones de una función desconocida $g : \Omega \to \mathbb{R}$, $Z_i \approx g(X_i, Y_i)$. El objetivo es encontrar o construir una función f que aproxime a g en todo el dominio Ω, $f(x, y) \approx g(x, y)$ para $(x, y) \in \Omega$. En el caso general, cuando los datos pueden estar contaminados por ruido, el problema es conocido como el problema de ajuste o aproximación de datos dispersos. Por otro lado, si los valores exactos de la función g son conocidos en los nodos (X_i, Y_i), $Z_i = g(X_i, Y_i)$, entonces a el problema se le conoce como el problema de interpolación de datos dispersos. Éste es un caso particular del problema general.

Los métodos de aproximación de datos dispersos se pueden clasificar como globales o locales. Los métodos locales dependen de triangulaciones del dominio en las cuales a cada elemento le corresponde una función componente que aproxima los datos de manera local. En contraste, los métodos globales no dependen de triangulación alguna y producen una función que aproxima los datos en todo el dominio. Entre los distintos métodos de aproximación de datos dispersos los que se basan en funciones de base radial(RBFs) han recibido gran atención gracias a su presición. Sin embargo, métodos globales basados en RBFs sufren de inestabilidad numérica. Ésta es la razón por la cual algunos autores usan RBFs localmente en conjuntos pequeños para desarrollar sus métodos como uno de los tres descritos en éste trabajo.

Otro tipo de funciones como los polinomios se usan para producir aproximaciones globales en una dimensión. Algunos científicos trataron de usar polinomios para atacar el problema en más de una dimensión hasta que en 1952 el teorema de Maihuber-Curtis fue demostrado estableciendo que el problema multivariado no está bien planteado[1]. Éste teorema motivó a los científicos a ir en otras direcciones.

Richard Franke [Franke, 1979] hizo una comparación crítica de 29 métodos de interpolación de datos dispersos de su época. Los criterios que él usó fueron: tiempo de ejecución, memoria utilizada, precisión y facilidad de implementación. Franke concluyó que los métodos basdos en RBFs eran los mejores.

Éste problema es de interés por sus aplicaciones entre las cuales listamos algunas. En restauración de imágenes los métodos de interpolación de datos dispersos se pueden usar para rellenar huecos en los datos. En este caso, datos faltantes en una imagen necesitan ser reconstruidos a partir de los datos disponibles. Una segunda aplicación es el modelado del suelo marino. El suelo marino es escaneado con sensores que se colocan en la base de los barcos. Dichos sensores emiten ondas que rebotan en el suelo marino y regresan al barco en distintos tiempos dependiendo de la profundidad. Los datos que se colectan se conocen como datos batimétricos que son convertidos en puntos espaciales y pueden

[1]el teorema de Mairhuber-Curtis es más general e involucra otro tipo de funciones además de polinomios

ajustarse con superficies que resultan de aplicar métodos de aproximación.

En el presente trabajo describimos e implementamos dos métodos recientes y los comparamos con un tercer método que es propuesto aquí. En el segundo capítulo describimos brevemente el método de Potts [Kunis y Potts, 2007]. Éste es global e interpola los datos de manera exacta, o dicho en otras palabras, considera que los datos no tienen ruido. El método de Potts incorpora la técnica de gradiente conjugado para resolver un sistema lineal. En el tercer capítulo presentamos el método de Davydov [Davydov y Zeilfelder, 2004][Davydov, Sestini y Morandi, 2005][Davydov, Morandi y Sestini, 2006]. Éste método es local y se basa en splines y RBFs. En el capítulo 4, proponemos un nuevo método basado en Series de Fourier y la técnica de mínimos cuadrados con un término de regularización, también conocida como técnica de mínimos cuadrados penalizados. Una característica en común entre el método propuesto y el método de Potts es el uso de series de Fourier. Sin embargo, en el método propuesto, a diferencia del método de Potts, minimizamos la rugosidad de la superficie aproximante y consideramos que los datos pueden tener ruido. El método de Davydov y el método planteado coinciden en el uso de la técnica de mínimos cuadrados.

Capítulo 1

Transformada rápida de Fourier para datos no equiespaciados

En éste capítulo resumimos el trabajo de Steidl[Steidl, 1998] y Potts D. et al.[Potts, Steidl y Tasche, 2001] relacionado al desarrollo de un algoritmo que calcula de manera eficiente la Transformada Discreta de Fourier No Equiespaciada (NDFT). Éste es una herrameinta que se utiliza en el algoritmo de interpolación descrito en la siguiente sección. La NDFT en su forma más general en dos dimensiones está dada por la siguiente fórmula

$$f(v_j) = \sum_{k \in I_N} f_k e^{-2\pi i x_k v_j}, \quad (j \in I_M) \tag{1.1}$$

donde $f_k \in \mathbb{C}$, $I_N := \{k \in \mathbb{Z}^2 : -\frac{N}{2} \leq k < \frac{N}{2}\}$ y $x_k \in \Pi^2$, $v_j \in N\Pi^2$ donde $\Pi^2 := [-\frac{1}{2}, \frac{1}{2})^2$.

En ésta tesis consideraremos un caso particular en el cual los nodos en tiempo o en frecuencia están equiespaciados:

$$
\begin{aligned}
f(v_j) &= \sum_{k\in I_N} f_k e^{-2\pi i k v_j/N} \quad (j \in I_M) \\
h(k) &= \sum_{j\in I_N} f_j e^{-2\pi i k v_j/N} \quad (k \in I_M)
\end{aligned} \tag{1.2}
$$

El algoritmo consiste en approximar $f(v_j)$ mediante una combinación lineal de traslaciones de una función base. Los números que determinan las traslaciones son parte de una malla equiespaciada. La idea esencial del algoritmo es usar la Transformada Rápida de Fourier (FFT) para calcular unos términos que surgen al calcular la serie de Fourier de la función aproximante. Éstos términos se obtienen mediante un cambio de variabe y dependen de la uniformidad de los traslados como veremos a continuación.

Sea φ una función periódica con periodo 1 y cuya serie de Fourier es convergente. Potts et. al. [Potts, Steidl y Tasche, 2001] aproximan f en (1.2) mediante

$$
s_1(v) := \sum_{l\in I_n} g_l \varphi(v - \frac{l}{n}) \tag{1.3}
$$

donde $n := \alpha N$, y $\alpha > 1$ es un factor de sobremuestreo.

Calculando la serie de Fourier de s_1

$$
\begin{aligned}
s_1(v) &= \sum_{l\in I_n} g_l \sum_{k\in\mathbb{Z}^2} \int_{\Pi^2} \varphi(v - \frac{l}{n}) e^{2\pi i k v} dv e^{-2\pi i k v} \\
&= \sum_{l\in I_n} g_l e^{2\pi i k l/n} \sum_{k\in\mathbb{Z}^2} \int_{\Pi^2} \varphi(v) e^{2\pi i k v} dv e^{-2\pi i k v}
\end{aligned}
$$

donde se usó el cambio de variable $v = v - l/n$. Entonces podemos escribir

$$
s_1(v) = \sum_{k\in\mathbb{Z}^2} \hat{g}_k c_k(\varphi) e^{-2\pi i k v} \tag{1.4}
$$

donde

$$\hat{g}_k := \sum_{l\in I_n} g_l e^{2\pi ikl/n}$$
$$c_k(\varphi) := \int_{\Pi^2} \varphi(v) e^{2\pi ikv} dv \quad (k \in \mathbb{Z}^2) \tag{1.5}$$

Luego, la suma en la serie de Fourier truncada es separada en dos partes

$$s_1(v) = \sum_{k\in I_n} \hat{g}_k c_k(\varphi) e^{-2\pi ikv} + \sum_{r\in\mathbb{Z}^2\backslash\{0\}} \sum_{k\in I_n} \hat{g}_k c_{k+nr}(\varphi) e^{-2\pi i(k+nr)v} \tag{1.6}$$

Si los coeficientes de Fourier $c_k(\varphi)$ se vuelven suficientemente pequeños para $k \in \mathbb{Z}^2\backslash I_n$ y si $c_k(\varphi) \neq 0$ para $k \in I_N$, entonces Potts, comparando (1.4) con f en (1.2), sugiere

$$\hat{g}_k := \begin{cases} f_k/c_k(\varphi) & k \in I_N \\ 0 & k \in I_n\backslash I_N \end{cases} \tag{1.7}$$

Ahora, los valores g_l en (1.5) se pueden obtener mediante la FFT inversa. Si φ está bien localizada en el dominio de tiempo de tal manera ques se pueda aproximar por una función periódica ψ con periodo 1 con $\psi \cap \Pi^2 \subset \frac{2m}{n}\Pi^2$ $(2m << n)$, entonces

$$f(w_j) \approx s_1(w_j) \approx s(w_j) = \sum_{l\in I_{n,m}(w_j)} g_l \psi(w_j - \frac{l}{n}) \tag{1.8}$$

con $I_{n,m}(w_j) := \{l \in I_n : nw_j - m \leq l \leq nw_j + m\}$. Para $w_j \in \Pi^2$ fijo, ésta suma sontiene a lo más $(2m+2)^2$ sumandos distintos de cero.

Éste razonamiento se resume en el siguiente algoritmo

Algoritmo 1 Cómputo rápido de NDFT (1.2) $f(w_j)$

Datos de entrada: $N \in \mathbb{N}$, $\alpha > 1$, $n := \alpha N$, $w_j \in \Pi^2$, $f_k \in \mathbb{C}$ $(j, k \in I_N)$

1: Calcular $c_k(\varphi)$ $(k \in I_N)$, $\psi(w_j - \frac{l}{n})$ $(j \in I_N, l \in I_{n,m}(w_j))$

2: Calcular $\hat{g}_k := f_k / c_k(\varphi)$ $(k \in I_N)$.

3: Calcular (mediante la FFT en dos variables)

$$g_l := n^{-2} \sum_{k \in I_N} \hat{g}_k e^{-2\pi i k l/n} \quad (l \in I_n)$$

4: Calcular

$$s(w_j) := \sum_{l \in I_{n,m}(w_j)} g_l \psi(w_j - \frac{l}{n}) \quad (j \in I_N)$$

Datos de salida: $s(w_j)$ valor aproximado de $f(w_j)$ $(j \in I_N)$

Ahora, para aproximar $h(k)$ en (1.2) Potts usa una función de la forma

$$g(x) := \sum_{j \in I_N} f_j \varphi(x + w_j)$$

Como antes, los coeficientes de Fourier se calculan obteniendo

$$c_k(g) = \sum_{j \in I_N} f_j e^{-2\pi i k w_j} c_k(\varphi) = h(k) c_k(\varphi) \quad (k \in \mathbb{Z}^2)$$

Entonces $h(k)$ en (1.2) se puede calcular si $c_k(g)$ es conocido. Por la definición de los coeficientes de Fourier

$$c_k(g) = \int_{\Pi^2} \sum_{j \in I_N} f_j \varphi(x + w_j) e^{2\pi i k x} dx,$$

Potts et al. aproximan ésta ecuación con la regla trapezoidal

$$c_k(g) \approx \frac{1}{n^2} \sum_{l \in I_n} \sum_{j \in I_N} f_j \varphi(w_j - \frac{l}{n}) e^{-2\pi i k l/n}$$

lo cual introduce un *error de aliasing*. Más aún, se reemplaza a φ con su versión truncada ψ, lo cual introduce un *error de truncamiento*. Ésto se resume en el siguiente algoritmo

---

Algoritmo 2 Cómputo rápido de NDFT (1.2) $h(k)$

Datos de entrada: $N \in \mathbb{N}$, $\alpha > 1$, $n := \alpha N$, $w_j \in \Pi^2$, $f_k \in \mathbb{C}$ $(j, k \in I_N)$

1: Calcular $c_k(\varphi)$ $(k \in I_N)$, $\psi(w_j - \frac{l}{n})$ $(l \in I_n, j \in J_{n,m}(l))$

2: Calcular

$$\tilde{g}_l := \sum_{j \in J_{n,m(l)}} f_j \psi(w_j - \frac{l}{n}) \ \ (l \in I_n)$$

3: Mediante la FFT calcular

$$\tilde{c}_k(g) := n^{-2} \sum_{l \in I_n} \tilde{g}_l e^{-2\pi i k l / n} \ \ (k \in I_N)$$

4: Calcular $\tilde{h}(k) := \tilde{c}_k(g)/c_k(\varphi)$ $(k \in I_N)$

Datos de salida: $\tilde{h}(k)$ que aproxima a $h(k)$ $(k \in I_N)$

Ahora incluimos las cotas de error derivadas por Potts et al. en los siguientes teoremas para casos particulares de φ y ψ.

Teorema 1.0.1 (Potts, Steidl y Tasche, 2001)**.** *Sea f en (1.2) calculada por el Algoritmo 1 con la campana Gaussiana dilatada,*

$$\phi(v) = (\pi b)^{-1/2} \sum_{r \in \mathbb{Z}} e^{-(n(v+r))^2/b} \tag{1.9}$$

y con la version truncada de ϕ

$$\psi(v) := (\pi b)^{-1/2} \sum_{r \in \mathbb{Z}} e^{-(n(v+r))^2/b} \chi_{[-m,m]}(n(v+r)) \tag{1.10}$$

donde $\chi_{[-m,m]}$ *es la función característica de* $[-m, m]$*. Sea* $\alpha > 1$ *y* $1 \leq b \leq \frac{2\alpha m}{(2\alpha-1)\pi}$*. Entonces*

$$c_k(\phi) = \frac{1}{n} e^{-\left(\frac{\pi k}{n}\right)^2 b}$$

y

$$\begin{aligned} E_a(w_j) &\leq \|f\|_1 e^{-b\pi^2\left(1-\frac{1}{\alpha}\right)} \Big(1 + \frac{\alpha}{(2\alpha-1)b\pi^2} + e^{-2b\pi^2/\alpha}\Big(1 + \frac{\alpha}{(2\alpha+1)b\pi^2}\Big)\Big), \\ E_t(w_j) &\leq \|f\|_1 \frac{2}{\sqrt{\pi b}}\big(1 + \frac{b}{2m}\big) e^{b\pi^2\left(\left(\frac{m}{b\pi}\right)^2 - \left(\frac{1}{2\alpha}\right)^2\right)}, \\ E_\infty &\leq 4\|f\|_1 e^{-b\pi^2\left(1-\frac{1}{\alpha}\right)} \end{aligned}$$

El error de aproximación decrece al incrementar b. Por lo tanto

$$b = \frac{2\alpha m}{(2\alpha - 1)\pi} \tag{1.11}$$

es una buena opción para el valor de b como una función de α y m.

Teorema 1.0.2 (Potts, Steidl y Tasche, 2001)**.** *Sea* f *en (1.2) calculada por el Algoritmo 1 con el B-spline cardinal centrado dilatado y periodizado*

$$\phi(v) := \sum_{r\in\mathbb{Z}} M_{2m}(n(v + r)) \;\; (m \geq 1) \tag{1.12}$$

de orden $2m$ *y sea* $\psi = \phi$*. Entonces*

$$c_k(\phi) = \begin{cases} \frac{1}{n} & k = 0, \\ \frac{1}{n}\left(\frac{\sin(k\pi/n)}{k\pi/n}\right)^{2m} & \textit{en otro caso} \end{cases}$$

y para $\alpha > 1$ *y* $0 \leq \eta \leq 4m/3$*,*

$$E_\infty \leq \frac{4m}{2m-1}\left(\frac{N}{2}\right)^{-\eta} (2\alpha - 1)^{-2m} |f|_{\eta,1}.$$

donde $|f|_{\eta,1}$ *denota la seminorma de Sobolev* $|f|_{\eta,1} := \sum_{k=-N/2}^{N/2-1} |f_k||k|^\eta$*.*

1.1. Interpolación de datos dispersos con polinomios trigonométricos

En ésta sección describimos el método desarrollado por Potts et. al. que fue introducido a la literatura en el artículo [Kunis y Potts, 2007].

El problema de interpolación de datos dispersos en dos dimensiones consiste en aproximar una función cuyos valores se conocen solamente en un conjunto finito $\Xi := \{\mathbf{X}_j \in \mathbb{T}^2 : j = 0, ..., M-1\}$ donde $\mathbb{T} := [-1/2, 1/2)$ y $\mathbf{X}_j = (X_j, Y_j)$. En otras palabras, dado un conjunto de muestras $(\mathbf{X}_j, Z_j) \in \mathbb{T}^2 \times \mathbb{C}$ que son observaciones de una función por lo demás desconocida, el objetivo es aproximar dicha función en puntos diferentes a $\mathbf{X}_j$. Existen muchas maneras de hacer esto. Potts et. al. usan polinomios trigonométricos en dos variables

$$f(\mathbf{x}) := \sum_{\mathbf{k} \in I_N} \hat{f}_{\mathbf{k}} e^{2\pi i \mathbf{k}\mathbf{x}}$$

donde $N \in 2\mathbb{N}$ es el grado del polinomio, $\mathbf{x} = (x, y)$ y $I_N := \{-N/2, ..., N/2+1\}^2$ es el índice de las frecuencias.

La meta es que la función $f(\mathbf{X})$ aproxime los datos, es decir $f(\mathbf{X}_j) \approx Z_j$. Ésta condición se puede expresar en forma matricial como

$$\mathbf{A}\hat{\mathbf{f}} \approx \vec{z}$$

donde $\hat{\mathbf{f}}$ es el vector de coeficientes de Fourier $\hat{\mathbf{f}} := (\hat{f}_{\mathbf{k}})_{\mathbf{k} \in I_N} \in \mathbb{C}^{N^2}$ y $\vec{z} := (Z_j)_{j=0,...,M-1} \in \mathbb{C}^M$. La matriz de Fourier no equiespaciada se define como

$$\mathbf{A} = \mathbf{A}_\Xi := \left(e^{2\pi i \mathbf{k}\mathbf{X}_j}\right)_{j=0,...,M-1; \mathbf{k} \in I_N} \in \mathbb{C}^{M \times N^2}.$$

Potts et al. [Kunis y Potts, 2007] se concentran en el caso indeterminado ($N^d > M$, esto significa que hay más variables que ecuaciones) que consiste en interpolar los datos exactamente $\mathbf{A}\hat{\mathbf{f}} = \vec{z}$. Ellos mostraron que $\mathbf{A}$ tiene rango completo si $N > 2dq^{-1}$. En su análisis [Kunis y Potts, 2007] *factores de amortiguación* $\hat{w}_{\mathbf{k}>0}$, $\mathbf{k} \in I_N$ son incorporados para compensar agrupaciones densas de datos. El planteamiento de problema toma la forma

$$\|\hat{\mathbf{f}}\|^2_{\hat{\mathbf{W}}^{-1}} = \sum_{\mathbf{k} \in I_N} \frac{|\hat{f}_{\mathbf{k}}|^2}{\hat{w}_{\mathbf{k}}} \xrightarrow{\hat{\mathbf{f}}} \text{mín} \quad \text{sujeto a} \quad \mathbf{A}\hat{\mathbf{f}} = \vec{z} \tag{1.13}$$

donde $\hat{\mathbf{W}} := \operatorname{diag}(\hat{w}_{\mathbf{k}})_{\mathbf{k} \in I_N}$.

Para reinterpretar el problema de tal manera que permita el uso del método de gradiente conjugado necesitamos la siguiente definición

Definición 1.1.1 (Kunis y Potts, 2007). *Sean $N \in 2\mathbb{N}$, $I_N = \{-N/2, ..., N/2-1\}^2$ dados. Para pesos positivos $\hat{w}_{\mathbf{k}} > 0$, $\mathbf{k} \in I_N$ con $\sum_{\mathbf{k} \in I_N} \hat{w}_{\mathbf{k}} = 1$ y para $\mathbf{x} = (x, y) \in \mathbb{T}^2$ el kernel trigonométrico se define como*

$$K_N(\mathbf{x}) := \sum_{\mathbf{k} \in I_N} \hat{w}_{\mathbf{k}} e^{2\pi i \mathbf{k}\mathbf{x}}$$

Dado $\mathcal{X} \subset \mathbb{T}^2$ la matriz del kernel $\mathbf{K}_N$ se define como

$$\mathbf{K}_N := \big(K_N(\mathbf{x}_j - \mathbf{x}_l)\big)_{j,l=0,\ldots,M-1} \in \mathbb{C}^{M \times M} \tag{1.14}$$

$\lambda_{\text{máx}}$ y $\lambda_{\text{mín}}$ denotan el eingenvalor más grande y el eigenvalor más pequeño de $\mathbf{K}_N$

Para resolver el problema (1.13) con la técnica de Gradiente Conjugado Potts et. al. prueban el siguiente lema

Lema 1.1.1 (Kunis y Potts, 2007). *Considerar que el número de nodos $M \in \mathbb{N}$, el conjunto de muestras $\Xi \subset \mathbb{T}^2$, el grado del polinomio $N \in 2\mathbb{N}$, y los factores de amortiguación $\hat{w}_k > 0$, $\mathbf{k} \in I_N$ están dados. El problema de interpolación óptimo (1.13) es equivalente a las ecuaciones normales amortiguadas de segundo tipo*

$$\mathbf{K}_N \tilde{\mathbf{f}} = \vec{z}, \quad \hat{\mathbf{f}} = \hat{\mathbf{W}} \mathbf{A}^H \tilde{\mathbf{f}} \tag{1.15}$$

donde la matriz del kernel $\mathbf{K}_N \in \mathbb{C}^{M \times M}$ obedece la siguiente factorización

$$\mathbf{K}_N = \mathbf{A} \hat{\mathbf{W}} \mathbf{A}^H$$

y por lo tanto es positivo definida.

El método de Potts consiste en resolver las ecuaciones normales (1.15) con la técnica del Gradiente Conjugado. Más precisamente el método calcula los coeficientes de Fourier $\hat{\mathbf{f}}$ implementando el siguiente algoritmo

Algoritmo 3 CGNE

Datos de entrada: número de puntos $M \in \mathbb{N}$, grado del polinomio $N \in 2\mathbb{N}$; conjunto de puntos $\mathcal{X} \subset \mathbb{T}^d$, valores $\mathbf{y} \in \mathbb{C}^M$, y el vector inicial $\hat{\mathbf{f}}_0 \in \mathbb{C}^{N^2}$

$\mathbf{r}_0 = \mathbf{y} - \mathbf{A}\hat{\mathbf{f}}_0$

$\hat{\mathbf{p}}_0 = \mathbf{A}^H \mathbf{r}_0$

for l=0,... **do**

$\quad \alpha_l = \mathbf{r}_l^H \mathbf{r}_l / \hat{\mathbf{p}}_l^H \hat{\mathbf{W}} \mathbf{p}_l$

$\quad \hat{\mathbf{f}}_{l+1} = \hat{\mathbf{f}}_l + \alpha_l \hat{\mathbf{W}} \hat{\mathbf{p}}_l$

$\quad \mathbf{r}_{l+1} = \mathbf{r}_l - \alpha_l \mathbf{A} \hat{\mathbf{W}} \hat{\mathbf{p}}_l$

$\quad \beta_l = \mathbf{r}_{l+1}^H \mathbf{r}_{l+1} / \mathbf{r}_l^H \mathbf{r}_l$

$\quad \hat{\mathbf{p}}_{l+1} = \beta_l \hat{\mathbf{p}}_l + \mathbf{A}^H \mathbf{r}_{l+1}$

end for

Datos de salida: la l-ésima iteración $\hat{\mathbf{f}}_l$

Aplicando una estimación estándar para la convergencia del método de gradiente conjugado, Potts et. al. demuestran el siguiente resultado concerniente a la estabilidad del método.

Lema 1.1.2 (Kunis y Potts, 2007)**.** *Sea* $\mathbf{K}_N$ *la matriz del kernel dada en (1.14) y sea* $\hat{\mathbf{e}}_l := \hat{\mathbf{f}}_l - \hat{\mathbf{W}} \mathbf{A}^H \mathbf{K}_N^{-1} \vec{z}$ *el error de la l-ésima iteración del algoritmo CGNE. Si* $\mathbf{K}_N$ *es regular entonces*

$$\|\hat{\mathbf{e}}_l\|_{\hat{\mathbf{W}}^{-1}} \leq 2 \left(\frac{\sqrt{\lambda_{\text{máx}}} - \sqrt{\lambda_{\text{mín}}}}{\sqrt{\lambda_{\text{máx}}} + \sqrt{\lambda_{\text{mín}}}} \right) \|\hat{\mathbf{e}}_0\|_{\hat{\mathbf{W}}^{-1}}$$

Capítulo 2

Método local basado en splines y RBFs

En éste capítulo resumimos el método presentado en los artículos [Davydov y Zeilfelder, 2004], [Davydov, Sestini y Morandi, 2005] y [Davydov, Morandi y Sestini, 2006]. Básicamente, éste método consiste en aproximar localmente los datos mediante polinomios y después ensamblar una solución global usando condiciones de suavidad tipo spline (otro método basado en splines se puede encontrar en [Zhou y Han, 2008]). Se usa la representación Bernstein-Bézier de los polinomios pues ésta permite expresar las condiciones de suavidad mediante un sistema lineal de ecuaciones. Además, se emplea la técnica de mínimos cuadrados para producir la aproximación local. En la aproximación local, si la matriz es mal condicionada (según una tolerancia especificada por el usuario) se recalcula la matriz usando un polinomio de menor grado. En el segundo y tercer artículos Davydov et. al. producen una solución híbrida al agregar una RBF a los polinomios. Referimos al lector a esos artículos si el(ella) está interesado(a) en las pruebas de los teoremas y lemas que a continuación se presentan.

Como se explicó en la introducción los datos consisten de un conjunto finito de puntos arbitrariamente distribuidos $\Xi = \{\mathbf{X}_i\}_{i=1}^N \subset \Omega$ donde $\mathbf{X}_i = (X_i, Y_i)$ en un dominio acotado $\Omega \subset \mathbb{R}^2$, con un número real Z_i asignado a cada $\mathbf{X}_i$, $i = 1, ..., N$. El objetivo de el *problema de aproximación de datos dispersos* es encontrar una función suave s definida en Ω que aproxime los datos, es decir $s(\mathbf{X}_i) \approx Z_i$, $i = 1, ..., N$. En la primera versión del método de Davydov, los datos son aproximados con un spline de clase C^1 de grado 3 determinado en una triangulación uniforme Δ de Ω definida abajo. Cada pieza del spline es un polinomio dado en la forma Bernstein-Bézier. En [Davydov y Zeilfelder, 2004] Davydov et. al. producen una aproximación en tres espacios diferentes. Sin embargo, dado que el razonamiento es muy similar en cada espacio detallamos el método solamente en el espacio $\mathcal{S}_3^1(\Delta)$ definido abajo.

Primero damos algunas definiciones. Más información puede encontrarse en los artículos [Davydov y Schumaker, 2002],[Farin, 1993].

Sea $T := \langle u, v, w \rangle \subset \mathbb{R}^2$ un triángulo y sean B_{ijk}^3, $i+j+k = 3$, los polinomios de Bernstein de grado 3 correspondientes a T. Cada polinomio $p \in \mathcal{P}_3$ tiene una única representación Bernstein-Bézier (BB)

$$p = \sum_{i+j+k=3} c_{ijk} B_{ijk}^3. \tag{2.1}$$

Los coeficientes c_{ijk} se llaman los BB-coeficientes. Cada BB-coefficiente c_{ijk} está asociado con un punto en el dominio

$$\eta_{ijk}^T := (iu + jv + kw)/3$$

El conjunto de estos puntos en el triángulo T se denota por

$$\mathcal{D}_{3,T} := \{\eta_{ijk}^T : i + j + k = 3\} \tag{2.2}$$

Dada una triangulación Δ que cubre el dominio $\Omega \subset \mathbb{R}^2$, $\mathcal{S}_3^1(\Delta)$ denota el espacio de todas las funciones C^1 polinomiales a pedazos con respecto a Δ,

$$\mathcal{S}_3^1(\Delta) := \{s \in C^1(\Omega) : \; s|_T \in \mathcal{P}_3 \text{ para todos los triángulos } T \in \Delta, T \cap \Omega \neq \emptyset\}$$

donde $\mathcal{P}_3$ es el espacio de polinomios de dos variables de grado 3. Es bien conocido que existe una correspondencia uno a uno entre los elementos del espacio $\mathcal{S}_3^0$ de funciones continuas y polinomiales a pedazos de grado 3 y la sucesión de coeficientes $c_\eta = c_\eta(s)$, $\eta \in \mathcal{D}_{3,\Delta}$ con

$$\mathcal{D}_{3,\Delta} := \bigcup_{T \in \Delta} \mathcal{D}_{3,T} \tag{2.3}$$

Para cada $T \in \Delta$ y $\eta = \eta_{ijk}^T \in T \cap \mathcal{D}_{3,\Delta}$, $c_\eta(s)$ es el coeficiente c_{ijk} en la representación (2.1) para $p = s|_T$.

Si $\mathcal{S}$ es un subespacio lineal $\mathcal{S}_3^0(\Delta)$, entonces un conjunto $\mathcal{M} \subset \mathcal{D}_{3,\Delta}$ se llama un *conjunto determinante* para $\mathcal{S}$ si haciendo los coeficientes de $s \in \mathcal{S}$ asociados con los puntos $\mathcal{M}$ iguales a cero implica que todos los coeficientes de s correspondientes a los puntos en $\mathcal{D}_{3,\Delta}$ son cero. $\mathcal{M}$ es llamado un *Conjunto Mínimo Determinante (MDS)* para $\mathcal{S}$ si ningun subconjunto propio $\mathcal{M}' \subset \mathcal{M}$ es conjunto determinante.

Para construir el spline que aproxima los datos dispersos Davydov et. al. usan (BB-) condiciones de suavidad. Las BB-condiciones de suavidad se expresan como un sistema de ecuaciones lineales que involucran los BB-coeficientes y que determinan la suavidad del spline. Si $s \in \mathcal{S}_3^0(\Delta)$, y $T := \langle v_1, v_2, v_3 \rangle$ y

$\widetilde{T} := \langle v_4, v_3, v_2 \rangle$ son un par de triángulos adjuntos en Δ que coinciden en el lado $e = \langle v_2, v_3 \rangle$, entonces s es C^1 continuo a lo largo de e si y solo si [Farin, 1993]

$$\tilde{c}_{1,m-1,3-m} = \sum_{i+j+k=1} c_{i,j+3-m,k+m-1} B^1_{ijk}(v_4) \tag{2.4}$$

para $m = 1, ..., 3$, donde B^1_{ijk} son los polinomios Berstein de grado 1 con respecto a T y c_{ijk}, $\tilde{c}_{ijk}$ son los BB-coeficientes de $s|_T$, $s|_{\widetilde{T}}$, respectivamente.

Davydov et. al.[Davydov y Zeilfelder, 2004] consideran una triangulación uniforme Δ que cubre a Ω, llamada la *malla cuatro dimensional.* Por simplicidad, definimos Δ para $\Omega = [0,1]^2$. Usando $n+1$ líneas horizontales y verticales, Ω es cubierto con n^2 cuadrados

$$Q_{ij} = \left[\frac{i-1}{n}, \frac{i}{n}\right] \times \left[\frac{n-j}{n}, \frac{n-j+1}{n}\right], \quad i,j = 1, ..., n$$

Cada Q_{ij} se subdivide en cuatro subtriángulosis $T^{[k]}_{ij}$, $k = 1, ..., 4$, insertando las dos diagonales (los triangulos se numeran en sentido contrario a las manecillas del reloj, empezando por el subtriángulo de la izquierda de Q_{ij}). Suponemos que $n \geq 2$ es par.

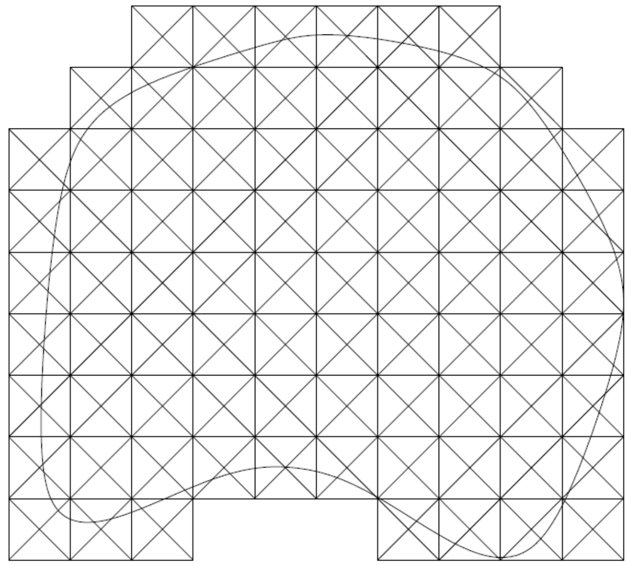


Figura 2.1: La *malla cuatro direccional* Δ que cubre el dominio Ω

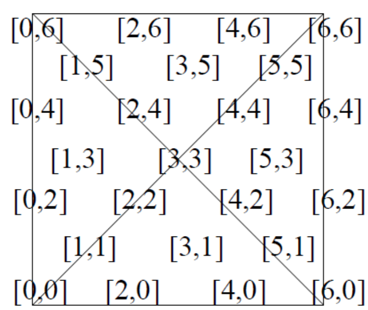


Figura 2.2: Los índices $[m, l]$ de los puntos $\eta_{i,j}^{[m,l]}$ dentro del cuadrado $Q_{i,j}$

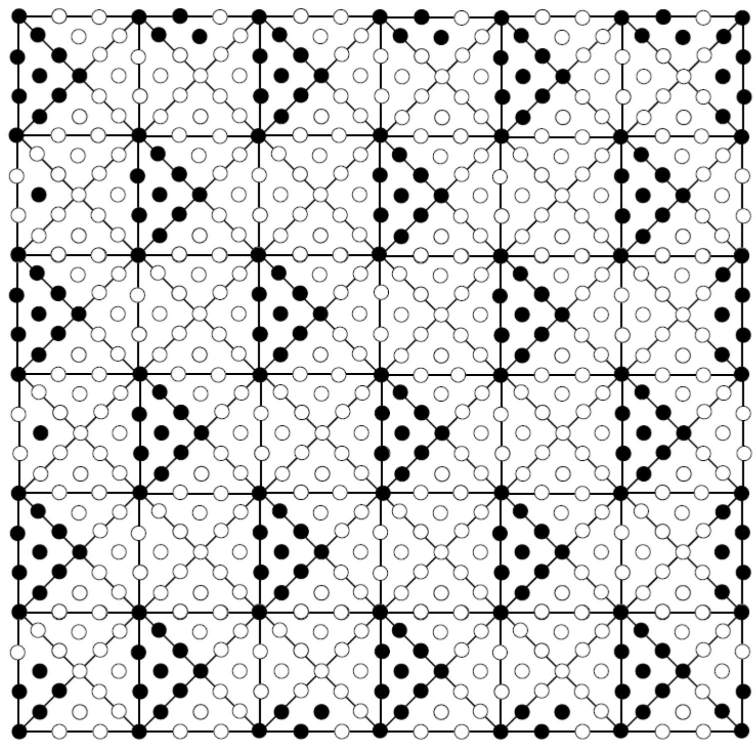


Figura 2.3: Un ejemplo de un MDS $\mathcal{M}$

Los puntos η_{ijk}^{T} exiben una estructura uniforme ya que la triangulación es regular. En la figura Fig. 2.3, los puntos dominio en $\mathcal{D} = \mathcal{D}_{3,\Delta}$ se muestran con círculos. Éstos están dados precisamente por

$$\mathcal{D} \cap Q_{i,j} = \left\{ \eta_{i,j}^{[m,l]} := \left(\frac{i-1}{n} + \frac{m}{6n}, \frac{n-j}{n} + \frac{l}{6n} \right) :\ 0 \leq m, l \leq 6,\ m + l \text{ even} \right\}$$

y $c_{i,j}^{[m,l]} := c_\eta$, si $\eta = \eta_{i,j}^{[m,l]}$ (ver Fig. 2.3).

Sea $\mathcal{M}$ el conjunto de puntos dominio marcados con círculos negros. La mayor parte de $\mathcal{M}$ consiste de $5n^2$ puntos en

$$\widetilde{\mathcal{M}} := \mathcal{D} \cap \bigcup_{i+j \text{ par}} T_{i,j}^{[1]} \tag{2.5}$$

Adicionalmente, $\mathcal{O}(n)$ puntos se localizan cerca de la frontera. Más precisamente, $\mathcal{M}$ es la unión de $\widetilde{\mathcal{M}}$ con los conjuntos

$$\begin{aligned}
&\{\eta_{i,1}^{[0,6]}, \eta_{i,1}^{[1,5]}, \eta_{i,1}^{[2,6]}, \eta_{i,1}^{[3,5]}\}, \quad i \text{ par},\\
&\{\eta_{i,n}^{[0,0]}, \eta_{i,n}^{[1,1]}, \eta_{i,n}^{[2,0]}, \eta_{i,n}^{[3,1]}\}, \quad i \geq 3, \quad i \text{ impar},\\
&\{\eta_{i,j}^{[1,3]}\}, \quad j \text{ par},\\
&\{\eta_{1,n}^{[0,0]}, \eta_{1,n}^{[0,2]}, \eta_{1,n}^{[1,1]}, \eta_{1,n}^{[2,2]}\},\\
&\{\eta_{n,j}^{[5,1]}, \eta_{n,j}^{[5,3]}, \eta_{n,j}^{[5,5]}, \eta_{n,j}^{[6,0]}, \eta_{n,j}^{[6,2]}, \eta_{n,j}^{[6,4]}, \eta_{n,j}^{[6,6]}\}, \quad j \text{ impar}, \text{ y}\\
&\{\eta_{n,n}^{[5,1]}, \eta_{n,n}^{[6,0]}, \eta_{n,n}^{[6,2]}\}.
\end{aligned}$$

El conjunto expresado en el primer renglón son los cuatro puntos de $\mathcal{M}$ en los subcuadrados pares de la parte superior del dominio cuadrado. El segundo renglón contiene los puntos de M en los subcuadrados impares de la parte inferior del dominio excepto el subcuadrado en la esquina inferior izquierda. El tercer renglón expresa los puntos de $\mathcal{M}$ en los subcuadrados pares del lado izquierdo del dominio cuadrado. El cuarto renglón contiene los puntos de $\mathcal{M}$ del subcuadrado de la esquina inferior izquierda. El quinto renglón contiene los puntos de $\mathcal{M}$ de los subcuadrados impares de la lado derecho del dominio cuadrado. El último renglón contiene los puntos de $\mathcal{M}$ en la esquina inferior derecha.

Teorema 2.0.1 (Davydov y Zeilfelder, 2004). *El conjunto $\mathcal{M}$ es un MDS para $\mathcal{S}_3^1(\Delta)$.*

Cada elemento η de un MDS $\mathcal{M}_{\mathcal{S}}$ para un espacio de splines en dos variables $\mathcal{S} \subset \mathcal{S}_3^0(\Delta)$ determina una función base B_η para $\mathcal{S}$, ["Davydov, 2002]. El spline B_η es definido haciendo todos lod BB-coeficientes en $\mathcal{M}_{\mathcal{S}}$ iguales a cero, excepto para $c_\eta(B_\eta) = 1$, y calculando los coeficientes restantes $c_\xi(B_\eta)$, $\xi \in \mathcal{D}_{3,\Delta} \backslash \mathcal{M}_{\mathcal{S}}$, usando las condiciones de suavidad que definen $\mathcal{S}$. Cada spline $s \in \mathcal{S}$ se puede escribir como

$$s = \sum_{\eta \in \mathcal{M}_{\mathcal{S}}} c_\eta(s) B_\eta.$$

Los splines B_η, $\eta \in \mathcal{M}_{\mathcal{S}}$, usados por Davydov et. al. tienen soporte compacto. Dado un vértice v de Δ, $\mathrm{star}(v) = \mathrm{star}^1(v)$ es la union de los triángulos que coinciden en v, y $\mathrm{star}^l(v)$, $l \geq 2$, se define recursivamente como la union de las stars de los vértices en $\mathrm{star}^{l-1}(v)$. Un spline $s \in \mathcal{S}$ se llama *l-localmente soportado* si existe un vértice v de Δ tal que

$$\mathrm{supp}\ s := \overline{\{z \in \Omega : s(z) \neq 0\}} \subset \mathrm{star}^l(v).$$

Sea

$$\widetilde{T} := \{T_{i,j}^{[1]} :\ i + j \text{ par}\},$$

Si dividimos $\mathcal{M} \backslash \widetilde{\mathcal{M}}$ en subconjuntos disjuntos de puntos η que se localizan en el mismo triángulo $T \in \Delta \backslash \widetilde{\mathcal{T}}$. Entonces, añadiendo éstos triángulos frontera a $\widetilde{\mathcal{T}}$, obtenemos un conjunto de triángulos en Δ denotado por $\mathcal{T}$. Sea

$$\mathcal{M}_T := \mathcal{M} \cap T, \ \ T \in \mathcal{T}$$

Para cada $T \in \mathcal{T}$, sea p_T una aproximación apropiada de una función f dada en un subdominio Ω_T que cubre a T. Davydov et. al. consideran el operador de aproximación Q definido por

$$Qf = \sum_{T\in\mathcal{T}} \sum_{\eta\in\mathcal{M}_T} c_\eta(p_T)B_\eta. \tag{2.6}$$

Dado que f se conoce solamente en un conjunto discreto de puntos dispersos en

$$\Xi = \{\mathbf{X}_i : i = 1, ..., N\} \subset \Omega,$$

Davydov et. al. [Davydov y Zeilfelder, 2004] determinan p_T, $T \in \mathcal{T}$, como polinomios cuyos valores

$$p_T(\mathbf{X}), \quad \mathbf{X} \in \Xi_T := \Xi \cap \Omega_T,$$

aproximan los valores correspondientes de f en Ξ_T. Para calcular la aproximación ellos aplican la técnica de mínimos cuadrados adaptativos.

Por conveniencia, se ausme que los datos Z_i son los valores de una función $f : \Omega \to \mathbb{R}$, i.e. $Z_i = f(\mathbf{X}_i)$, $i = 1, ..., N$. La aproximación polinomial local $p_T \in \mathcal{P}_3$ es determinada para cada triángulo $T \in \mathcal{T}$, usando los valores $f(\mathbf{X})$, $\mathbf{X} \in \Xi_T = \Xi \cap \Omega_T$, donde $\Omega_T \subset \Omega$ es el círculo centrado en el baricentro de T y radio igual al diámetro de T. Si éstos datos locales son muy pocos, el radio es incrementado y, en el caso opuesto, un algoritmo de disminución o adelgazamiento es usado para reducir su número N_T. Éste proceso de selección de datos locales es controlado por dos parámetros, $M_{\text{mín}}$ y $M_{\text{máx}}$, $M_{\text{mín}} \leq N_T \leq M_{\text{máx}}$. Con ésta consideración en mente, p_T se determina calculando los (BB-) coeficientes $c_{ijk}^{3,T}$ en su representación Bernstein-Bézier

$$p_T = \sum_{i+j+k=3} c_{ijk}^{3,T} B_{ijk}^3, \tag{2.7}$$

tal que

$$\sum_{\mathbf{X}\in\Xi_T} \Big(\sum_{i+j+k=3} c_{ijk} B_{ijk}^3(\mathbf{X}) - f(\mathbf{X}) \Big)^2$$

es minimizado. Aquí, B_{ijk}^3 son los polinomios Bernstein de grado 3 asociados con T. La manera más confiable de resolver éste problema discreto de mínimos cuadrados es calcular la Descomposición en Valores Singulares (SVD) de la matriz

$$M_{3,T} := \left[B_{ijk}^3(\mathbf{X})\right]_{i+j+k=3,\mathbf{X}\in\Xi_T} \tag{2.8}$$

de tamaño $(\#\Xi_T) \times \binom{3+2}{2}$. Entonces, el vector de coeficientes $\left(c_{ijk}^{3,T}\right)_{i+j+k=3}$ se calcula como el producto de la pseudoinversa $M_{3,T}^{+}$ de $M_{3,T}$ con el vector $(f(\mathbf{X}))_{\mathbf{X}\in\Xi_T}$, [Björck, 1996]. Sin embargo, el polinomio resultante p_T es aceptado como una aproximación confiable de los datos locales sólo si la matriz $M_{3,T}$ tiene rango completo y el recíproco de su valor singular más pequeño $\sigma_{3,T}$ no es más grande que una tolerancia κ_P,

$$\sigma_{3,T}^{-1} \leq \kappa_P. \tag{2.9}$$

Si (2.9) no se satisface, se concluye que la distribución local de los datos no permite una aproximación estable con polinomios de grado 3. En ésta situación el grado del polinomio es reducido por una unidad, y de la misma manera se calcula un polinomio p_T de grado $3-1$ mediante mínimos cuadrados de la forma

$$p_T = \sum_{i+j+k=2} c_{ijk}^{2,T} B_{ijk}^2,$$

donde B_{ijk}^2 son los polinomios Bernstein de grado 2 asociados a T. De ser necesario, el proceso es repetido y el grado de p_T se reduce a $q = 1, 0$. Éste algoritmo de reducción de grado termina en un grado $q > 0$ siempre y cuando el valor singular mínimo $\sigma_{q,T}$ de la matriz $M_{q,T} = \left[B_{ijk}^q(\mathbf{X})\right]_{i+j+k=q,\mathbf{X}\in\Xi_T}$ satisface $\sigma_{q,T}^{-1} \leq \kappa_P$. En este caso la aproximación local polinomial p_T será de grado q. Si $\sigma_{q,T}^{-1} > \kappa_P$ para $q = 3, 2, 1$, p_T se calcula como la mejor aproximación constante en mínimos cuadrados de $f(\mathbf{X})$, $\mathbf{X} \in \Xi_T$, la cual existe siempre que $\#\Xi_T \geq 1$.

Ya que la representación Bernstein-Bézier de grado 3 de p_T (2.7) es necesaria, fórmulas de incremento de grado, ver [Farin, 1993], se pueden aplicar en la situación anterior cuando $q < 3$.

Sea B_η, $\eta \in \mathcal{M}$, una base de $\mathcal{S}_3^1(\Delta)$ asociada con $\mathcal{M}$. Como se mencionó anteriormente B_η tiene soporte compacto (figure 3.4). Además cada B_η está uniformemente acotado.

Lema 2.0.1 (Davydov y Zeilfelder, 2004)**.** *Los splines base B_η, $\eta \in \mathcal{M}$, para $\mathcal{S}_3^1(\Delta)$ son 3-localmente suportados*

Lema 2.0.2 (Davydov y Zeilfelder, 2004)**.** *Los splines base B_η, $\eta \in \mathcal{M}$, para $\mathcal{S}_3^1(\Delta)$ están uniformemente acotados, i.e., existe una constante absoluta K tal que*

$$\|B_\eta\|_\infty \leq K, \quad \textit{para toda } \eta \in \mathcal{M} \tag{2.10}$$

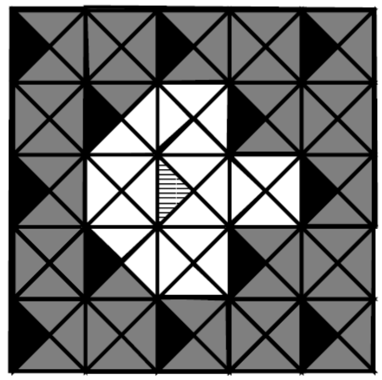


Figura 2.4: Soporte de los splines base (triángulos blancos) de clase C^1 asociados con los puntos en $\mathcal{M}$

Cuando Davydov et. al. introdujeron por primera vez su método también probaron la siguiente cota del error

Teorema 2.0.2 (Davydov y Zeilfelder, 2004)**.** *Sea Q el operador definido en (2.6). Si $f \in W_p^4(\Omega)$, para algún $1 \leq p \leq \infty$ entonces*

$$\|f - Qf\|_{L_p(\Omega)} \leq K_4 h^4 |f|_{W_p^4(\Omega)} + K_5 \Big(\sum_{T \in \mathcal{T}} \|f - p_T\|^p_{L_p(T)} \Big)^{1/p} \; si\; p < \infty$$

$$\|f - Qf\|_{L_\infty(\Omega)} \leq K_6 h^4(\Omega) |f|_{W_\infty^4(\Omega)} + K_7 \max_{T \in \mathcal{T}} \|f - p_T\|_{L_\infty(T)},$$

donde K_4, K_5, K_6, K_7 son constantes absolutas.

2.1. Aproximación local híbrida

En su segundo artículo, [Davydov, Morandi y Sestini, 2006], para mejorar el error de aproximación, Davydov et. al. modificaron la aproximación local polinomial incorporando un término RBF dando como resultado un método híbrido.

La idea detrás del método híbrido es mejorar la precisión de la aproximación local usando una combinación lineal de polinomios y RBFs.

El esquema de aproximación híbrido usa el espacio

$$\Pi_3^2 = \text{span}\{p_1^T, ..., p_m^T\}, \quad m = \binom{3+2}{2}$$

de polinomios bivariados de grado $3 \geq 0$ y una función $\phi_T : \mathbb{R}_{\geq 0} \to \mathbb{R}$. ϕ_T puede ser cualquier función *positiva definida* apropiada o una función *condicionalmente positiva definida* de orden a los más 4 en $\mathbb{R}^2$, ver [Buhmann, 2003] y [Wendland, 2004].

Sea $\Xi_T = \{\mathbf{X}_1, ..., \mathbf{X}_{N_T}\}$ el conjunto de datos dispersos relacionados al triángulo T con vértices $\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T$, y suponer que $N_T \geq m$. La *aproximación local híbrida* g_T de la forma

$$g_T(\cdot) = \sum_{j=1}^{m} a_j^T p_j^T(\cdot) + \sum_{j=1}^{n_T} b_j^T \phi_T(\| \cdot - \mathbf{Y}_j^T \|_2) \tag{2.11}$$

es construida mediante la minimización de la norma l_2 del vector residual en Ξ_T,

$$\left(\sum_{i=1}^{N_T} (f_i - g_T(\mathbf{X}_i))^2 \right)^{1/2} \tag{2.12}$$

donde $0 \leq n_T \leq N_T - m$, y el conjunto de puntos $Y_T = \{\mathbf{Y}_j, j = 1, ..., n_T\}$ es un subconjunto de Ξ_T.

La solubilidad única del problema de interpolación con RBFs condicionalmente positivas definidas se garantiza aplicando las *restricciones de ortogonalidad*

$$\sum_{j=1}^{n_T} b_j^T p(\mathbf{Y}_j^T) = 0, \quad \text{all } p \in \Pi_3^2 \tag{2.13}$$

sobre los coeficientes b_j^T en (2.11) y suponiendo que Y_T contiene un subconjunto unisolvente en Π_3^2, ver [Buhmann, 2003], [Wendland, 2004] y [Fasshauer, 2007].

Sin embargo Davydov et al. tomaron una ruta diferente. Para mantener los $n_T + \binom{3+2}{2}$ grados de libertad ellos ignoran las restricciones (2.13) y consideran el espacio híbrido

$$\mathcal{H}_T := \text{span}\{p_1^T, ..., p_m^T, \phi_T(\| \cdot - \mathbf{Y}_1^T \|_2), ..., \phi_T(\| \cdot - \mathbf{Y}_{n_T}^T \|_2)\}$$

en lugar del subespacio

$$\mathcal{R}_T := \left\{ \sum_{j=1}^{m} a_j p_j^T(\cdot) + \sum_{j=1}^{n_T} b_j \phi_T(\| \cdot - \mathbf{Y}_j^T \|_2) : (b_1, ..., b_{n_T}) \text{ satisface } (2.13) \right\}$$

Debido a que $\mathcal{R}_T \subset \mathcal{H}_T$, la capacidad de aproximación de $\mathcal{H}_T$ es al menos tan buena como la de $\mathcal{R}_T$

$$E(f, \mathcal{H}_T)_{C(T)} \leq E(f, \mathcal{R}_T)_{C(T)},$$

donde $E(f, \mathcal{S})_{C(T)}$ es el error de la mejor aproximación de f desde un espacio lineal $\mathcal{S}$,

$$E(f, \mathcal{S})_{C(T)} := \inf_{g \in \mathcal{S}} \|f - g\|_{C(T)}.$$

Para producir una aproximación local en mínimos cuadrados cuya calidad sea comparable con la calidad de la mejor aproximación local de $\mathcal{H}_T$, Davydov et. al. calculan el valor singular mínimo $\sigma_{\text{mín}}(C_T)$ de la matriz de colocación C_T definida por

$$\begin{bmatrix} p_1^T(\mathbf{X}_1) & ... & p_m^T(\mathbf{X}_1) & \phi_T(\|\mathbf{X}_1-\mathbf{Y}_1^T\|_2) & ... & \phi_T(\|\mathbf{X}_1-\mathbf{Y}_{n_T}^T\|_2) \\ \vdots & & \vdots & \vdots & & \vdots \\ p_1^T(\mathbf{X}_{N_T}) & ... & p_m^T(\mathbf{X}_{N_T}) & \phi_T(\|\mathbf{X}_{N_T}-\mathbf{Y}_1^T\|_2) & ... & \phi_T(\|\mathbf{X}_{N_T}-\mathbf{Y}_{n_T}^T\|_2) \end{bmatrix}$$

y proceden a calcular g_T solo si

$$\sigma_{\text{mín}}^{-1}(C_T) \leq \kappa_H, \tag{2.14}$$

donde κ_H es una tolerancia especificada por el usuario. Esto también garantiza que el problema de mínimos cuadrados considerado se puede resolver de manera única [Davydov, Morandi y Sestini, 2006].

De acuerdo a la estimación dada en ["Davydov, 2002], bajo el supuesto que $f_i = f(\mathbf{X}_i),\ i = 1, ..., N_T$, para una función continua f, tenemos

$$\|f - g_T\|_{C(T)} \leq \left(1 + \frac{K_T\sqrt{N_T}}{\sigma_{\text{mín}}(C_T)}\right) E(f, \mathcal{H}_T)_{C(T)} \tag{2.15}$$

donde

$$K_T := \max_{\{a_j\},\{b_j\}} \frac{\| \sum_{j=1}^m a_j p_j^T(\cdot) + \sum_{j=1}^{n_T} b_j \phi_T(\| \cdot - \mathbf{Y}_j^T\|_2)\|_C(T)}{\left(\sum_{j=1}^m |a_j|^2 + \sum_{j=1}^{n_T} |b_j|^2\right)^{1/2}}$$

Ahora, para una base polinomial escalada apropiadamente $\{p_1^T, ..., p_m^T\}$ (por ejemplo, la base Bernstein-Bézier con respecto a T usada en la implementación de Davydov), y una ϕ_T apropiada, K_T es acotada independientemente de T. Ya que además $N_T \leq M_{\text{máx}}$ donde $M_{\text{máx}}$ es el parámetro introducido en la sección anterior, la estimación (2.15) muestra que el tamaño de $\|f - g_T\|_{C(T)}$ es comparable con el tamaño de la aproximación $E(f, \mathcal{H}_T)_{C(T)}$ si (2.14) se satisface.

Entonces, (2.14) provee un criterio para aceptar o rechazar un conjunto Y_T dado (el cual determina de manera única a la matriz C_T). Para encontrar un conjunto Y_T apropiado, Davydov et. al. usan un procedimiento codicioso[1] ascendente. Ellos empiezan con unos pocos puntos y añaden más puntos en los lugares de máximo error mientras (2.14) se satisface. Si el conjunto inicial Y_T no satisface (2.14), entonces ellos optan por producir una aproximación polinomial local aplicando el procedimiento descrito en [Davydov y Zeilfelder, 2004] empezando con el grado 3.

El método de arriba se resume en el siguiente algoritmo. Además del conjunto de datos locales Ξ_T y los correspondientes valores $f_1, ..., f_{N_T}$, sus *parámetros de entrada* son los tres vértices $\{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\}$ del triángulo T, la cota superior $n_{\text{máx}} \geq 3$ de $n_T = \#Y_T$, y la tolerancia κ_P para la parte polinomial y la tolerancia κ_H en (2.14). Sus *datos de salida* son el conjunto de puntos Y_T con $n_T = \#Y_T \leq \text{mín}\{n_{\text{máx}}, N_T - m\}$ y la aproximación local correspondiente g_T, i.e. los coeficientes $a_1^T, ..., a_m^T$ y $b_1^T, ..., b_{n_T}^T$ que definen a g_T en (2.11).

Como se mencionó anteriormente, el procedimiento es *ascendente*, Davydov et. al. empienzan con solo tres nodos si N_T es lo suficientemente grande. A saber, si $N_T \geq m+3$, ellos seleccionan, por razones de simetría, los primeros tres nodos como los tres puntos distintos de Ξ_T más cercanos a los vértices del triángulo (de lo contrario si $N_T < m + 3$, entonces se calcula la aproximación polinomial local p_T de grado a lo más 3 usando el algoritmo en [(Davydov y Zeilfelder, 2004)]). Más nodos se siguen añadiendo mientras que $n_T < \text{mín}\{n_{\text{máx}}, N_T - m\}$ y (2.14) se satisfaga. (Si la condición (2.14) en un paso en éste ciclo while, entonces el último nodo insertado es removido y se regresa a la aproximación híbrida previa). En cada paso el nuevo nodo candidato se determina eligiendo una ubicación $\mathbf{X}_i \in \Xi_T \backslash Y_T$ donde el error de aproximación actual es el peor.

[1]traducción literal de la palabra en inglés *greedy*

Algoritmo 4 **Procedimiento LHA**

Datos de entrada: N_T, $\Xi_T = \{\mathbf{X}_1, ..., \mathbf{X}_{N_T}\}$, $\{f_1, ..., f_{N_T}\}$, $\{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\}$, $n_{\text{máx}}$, κ_P, κ_H

Datos de salida: Y_T, g_T

Inicializar $n_T = 0$, $Y_T = \emptyset$
if $N_T < \binom{3+2}{2} + 3$ **then**
  Calcular $g_T = p_T$ (polynomial approximation)
else
  Hacer $n_T = 3$
  for $i = 1, 2, 3$ **do**
    Definir $Y_T = Y_T \cup \{\mathbf{X}_{k_i}\}$, donde $\|\mathbf{X}_{k_i} - \mathbf{v}_i^T\|_2 = \text{mín}_{\mathbf{X} \in \Xi_T \setminus Y_T} \|\mathbf{X} - \mathbf{v}_i^T\|$
  end for
  Inicializar la matriz de colocación C_T
  if $\sigma_{\text{mín}}^{-1}(C_T) > \kappa_H$ **then**
    Hacer $n_T = 0$, $Y_T = \emptyset$
    Calcular $g_T = p_T$
  else
    Calcular $g_T =$ aproximación híbrida actual
    while $n_T < \text{mín}\{n_{\text{máx}}, N_T - \binom{3+2}{2}\}$ **do**
      Calcular $err_j = |f_j - g_T(\mathbf{X}_j)|,\ j = 1, ..., N_T$
      Hacer $i = \text{argmax}_{j \in \Xi_T \setminus Y_T} err_j$
      Asignar $n_T = n_T + 1,\ Y_T = Y_T \cup \{\mathbf{X}_i\}$
      Actualizar C_T
      if $\sigma_{\text{mín}}^{-1}(C_T) > \kappa_H$ **then**
        Hacer $n_T = n_T - 1$, $Y_T = Y_T \setminus \{\mathbf{X}_i\}$
        Break
      end if
      Calcular g_T
    end while
  end if
end if

La función híbrida de arriba se incorpora al algoritmo de dos etapas [Davydov y Zeilfelder, 2004], reemplazando la aproximación polinomial originalmente empleada en ese artículo. El resultado de la primer etapa en cada triángulo $T \in \mathcal{T}$ debe ser un polinomio p_T de grado 3 en la forma Bernstein-Bézier. Sin embargo, la aproximación híbrida local g_T está dada de manera distinta, (2.11). Para remediar ésta situación Davydov et. al. reemplazan g_T en (2.11) con un polinomio aplicando otra vez el método de mínimos cuadrados.

Más precisamente, la técnica discreta de mínimos cuadrados se usa con respecto a las evaluaciones de g_T en $\binom{D+2}{2}$ puntos dominio en T relacionados a los polinomios de cierto grado $D \geq 3$. En los experimentos numéricos usamos $D = 2 \times 3$. Enfatizamos que 3 es el grado del polinomio. El inverso del valor singular más pequeño de la matriz de colocación correspondiente es pequeño lo que garantiza que la aproximación en mínimos cuadrados es buena. Adicionalmente, la matriz de colocación es la misma para todos los triángulos $T \in \mathcal{T}$. Por lo tanto, el costo computacional añadido es insignificante.

Davydov et. al. usan la base de polinomios Bernstein con respecto al triángulo T como la base polinomial local $\{p_1^T, ..., p_m^T\}$ usada en (2.11). Respecto al término ϕ_T en (2.11) escogen la RBF *multicuádrica* de Hardy

$$\phi_{MQ}(r) = -\sqrt{1 + r^2}, \tag{2.16}$$

la cual es C^∞ y condicionalmente positiva definida de orden 1. Ésta función es la más comun y la más usada de las RBFs.

Davydov et. al. añaden otro grado de adaptabilidad escalando ϕ_T para cada T mediante un parámetro de escalamiento δ. Entonces la RBF usada en (2.11) toma la forma

$$\phi_T(r) = c_T \phi\Big(\frac{r}{\delta d_T}\Big) \tag{2.17}$$

donde

$$d_T := \max_{1\leq i,j\leq N_T} \|\mathbf{X}_i - \mathbf{X}_j\|_2$$

es el diámetro del conjunto local Ξ_T, y c_T es una constante. El valor de d_T variará con la densidad local de los datos. Ellos usan $c_T = -\delta d_T$ de tal manera que ϕ_T se convierte en

$$\phi_T(r) = \phi_{T,MQ}(r) := -\delta d_T \phi_{MQ}\Big(\frac{r}{\delta d_T}\Big) = \sqrt{(\delta d_T)^2 + r^2}. \tag{2.18}$$

En conjunto, además de la RBF ϕ, el método de Davydov depende de varios parámetros: el ancho de la malla h_x y el alto de la malla h_y, el grado q de la parte polinonmial usada en (2.11), las tolerancias κ_H and κ_P para el inverso del valor singular mínimo, los parámetros para la selección de puntos locales $M_{\text{mín}}$, $M_{\text{máx}}$, el coeficiente de escalamiento δ usado en (2.17), la cota superior $n_{\text{máx}}$ en el número de nodos n_T en (2.11), y el parámetro D de la malla uniforme usada para la evaluación de la aproximación híbrida local g_T.

En lugar de h_x y h_y usamos el tamaño de la malla $n_x \times n_y$. Para producir la malla se calcula el rectángulo mínimo $[a, b] \times [c, d]$ que contiene los puntos $\mathbf{X}_i$, $i = 1, ..., N$. Entonces, éste rectángulo se divide en $n_x n_y$ subrectángulos iguales insertando $n_x - 1$ líneas vecticales equiespaciadas y $n_y - 1$ líneas horizontales equiespaciadas. Entonces

$$h_x = (b - a)/n_x, \quad h_y = (d - c)/n_y$$

La malla cuatro direccional se obtiene insertando las diagonales de cada subrectángulo. Entonces el número de parámetros necesarios para guardar el spline es $5n_xn_y + 4(n_x + n_y) + 3$.

Davydov et. al. [Davydov, Morandi y Sestini, 2006] reportaron que el tamaño de la malla $n_x \times n_y$ así como los parámetros $M_{\text{mín}}$, $M_{\text{máx}}$, δ y κ_H influyen de manera significativa el desempeño del método. Después de ejecutar muchos experimentos ellos proclaman que no hay valores universales de éstos parámetros que sean mejores, pero que deben ser ajustados para el tipo de datos en cuestión. Sugieren que el valor de los parámetros se puede encontrar en la práctica mediante un procedimiento que involucre experimentar con pequeños subconjuntos de los datos y mediante la técnica de validación cruzada. Esto es porque en sus experimentos los mismos valores de los parámetros funcionaron con éxito en todas las aproximaciones locales. Ellos no emplearon validación cruzada. En lugar de eso encontraron buenos valores de los parámetros mediante prueba y error.

Para terminar éste capítulo describimos todos los parámetros usados en el método de Davydov.

parámetro	significado
n_x	Número de divisiones en lo ancho de $[a,b]\times[c,d]$.
n_y	Número de divisiones en lo alto de $[a,b]\times[c,d]$.
q	Grado de la parte polinomial de la aproximación híbrida (Ec. 2.11).
κ_H	Tolerancia del inverso del valor singular mínimo de la matriz de colocación del método híbrido.
κ_P	Tolerancia del inverso del valor singular mínimo de la matriz de colocación cuando un polinomio es usado como aproximación local.
$M_{\text{mín}}$	Cota inferior del número de nodos N_T usados en Ω_T, $M_{\text{mín}} \leq N_T$.
$M_{\text{máx}}$	Cota superior del número de nodos N_T usados en Ω_T, $N_T \leq M_{\text{máx}}$.
δ	Parámetro de escalamiento en la RBF ϕ_T.
$n_{\text{máx}}$	Cota superior $\#Y_T$, del número de nodos usados para la aproximación híbrida en los términos RBF.
D	Grado del polinomio usado para aproximar g_T en la segunda etápa del método.

Tabla 2.1: Significado de los parámetros usados en el método de Davydov [Davydov, Morandi y Sestini, 2006]

Capítulo 3

Método propuesto

En éste capítulo presentamos nuestro método el cual se basa en minimizar la rugosidad de la superficie aproximante así como su distancia a los datos. Originalmente ésta idea fue introducida por Reinsch [Reincsh, 1967] para atacar el problema de aproximación de datos en una dimensión. Reinsch usó splines para atenuar la oscilación de polinomios en la interpolación de datos dispersos. Un razonamiento similar puede encontrarse en el artículo [Kimeldorf y Wahba, 1971]. Aquí el dominio es un rectángulo simétrico $\Omega = [-l', l'] \times [-h', h']$ donde l' y h' son números positivos.

Como se mencionó en capítulos anteriores, los datos del problema es un conjunto de puntos $\{(X_i, Y_i, Z_i)\}_{i=1}^N$ donde $\{(X_i, Y_i)\}_{i=1}^N \subset [-l', l'] \times [-h', h']$. $Z_i = g(X_i, Y_i) + \epsilon_i$ son observaciones de una función desconocida $g(x, y)$ contaminadas con ruido ϵ_i (En capitúlos anteriores usamos $\mathbf{X}_i = (X_i, Y_i)$). El objetivo es aproximar $g(x, y)$.

Motivados por el teorema de convergencia de las series de Fourier, la función aproximante tiene la expresión[1]

[1]Los coeficientes $b_{0,n}$, $c_{m,0}$, $d_{0,n}$ y $d_{m,0}$ son redundantes ya que los términos trigonométricos

$$
\begin{aligned}
f(x,y) := \sum_{m=0}^{K_1}\sum_{n=0}^{K_2}\Big\{ a_{m,n}\cos\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right) \\
+b_{m,n}\sin\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right) + c_{m,n}\cos\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right) \\
+d_{m,n}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\Big\}
\end{aligned}
\tag{3.1}
$$

donde K_1 y K_2 son enteros no negativos que se escogen por el usuario y l, h dependen de l', h'. Si $g(x,y)$ no es periódica en $[-l',l'] \times [-h',h']$, entonces l y h se deben elegir de tal manera que $l > l'$ y $h > h'$ para evitar el fenómeno de Gibbs (Grishin et. al. [Grishin y Strohmer, 2003] aborda éste problema imponiendo condiciones de frontera tipo Neumann). Elegir $l = l'$ y $h = h'$ hace a $f(x,y)$ periódica con periodo l' y h' en x y y respectivamente. Esto produciría discontinuidades a lo largo de la frontera de $[-l',l'] \times [-h',h']$.

En la mayoría de los casos, cuando $g(x,y)$ no es periódica en $[-l',l'] \times [-h',h']$, la extensión periódica de $g(x,y)$ sobre $\mathbb{R}^2$ tendría discontinuidades a lo largo de las líneas $x = (2i+1)l'$ y $y = (2i+1)h'$ para $i \in \mathbb{Z}$. Sin embargo, si tomamos $l > l'$ y $h > h'$, le damos espacio a la función $f(x,y)$ a ser periódica en $[-l,l] \times [-h,h]$. No obstante, nos concentramos en el rectángulo $[-l',l'] \times [-h',h']$. No es recomendable optimizar sobre los valores de l y h porque eso sería costoso computacionalmente. Tenemos buenos resultados tomando $l = 3l'$ y $h = 3h'$.

Considerando el problema de optimización

$$
\min_f \left\{ \frac{1}{N}\sum_{s=1}^{N}(f(X_s,Y_s) - Z_s)^2 + \lambda \int_{-h}^{h}\int_{-l}^{l}(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)dxdy \right\} \tag{3.2}
$$

y definiendo $\mathcal{L}$ como

que les acompañan son iguales a cero

$$\mathcal{L}(\text{coeficientes}) = \frac{1}{N}\sum_{s=1}^{N}(f(X_s,Y_s) - Z_s)^2 + \lambda \int_{-h}^{h}\int_{-l}^{l}(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)dxdy$$

El primer término de la función $\mathcal{L}$ cuantifica la proximidad o distancia de la superficie aproximante a los datos mientras que el segundo término cuantifica su rugosidad. El parámetro λ determina la importancia o el peso que se le da a éstos dos criterios. Si λ es pequeño entonces la superficie aproximante ajustará bien los datos y podría aparecer rugosidad que no es propia de la función a aproximar. Dicha rugosidad se debe a las oscilaciones naturales de las funciones trigonométricas que forman a la función aproximante En cambio, si λ es muy grande la superficie aproximante será suave, e incluso podría llegar a ser plana y no ajustará bien los datos. Un método para determinar λ, que no discutiremos en éste trabajo, es el método de la curva L. El parámetro λ se determina aquí a prueba y error.

Los coeficientes de Fourier $a_{m,n}$, $b_{m,n}$, $c_{m,n}$, $d_{m,n}$ se determinan resolviendo el sistema

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial a_{i,j}} &= 0 \\ \frac{\partial \mathcal{L}}{\partial b_{i,j}} &= 0 \\ \frac{\partial \mathcal{L}}{\partial c_{i,j}} &= 0 \\ \frac{\partial \mathcal{L}}{\partial d_{i,j}} &= 0 \end{aligned} \tag{3.3}$$

o $\nabla \mathcal{L} = 0$.

Ésta técnica es conocida como mínimos cuadrados penalizados. Esto es porque cuando $\lambda = 0$ el problema 3.2 se puede expresar en la forma más familiar

$$\min_{\vec{x}} \|\tilde{A}\vec{x} - \vec{d}\|_2^2 \tag{3.4}$$

donde $\tilde{A}$ es una matriz cuyas filas son

$$\begin{aligned}\tilde{A}_i = \Big(&..., \cos\left(\frac{\pi m X_i}{l}\right)\cos\left(\frac{\pi n Y_i}{h}\right), \sin\left(\frac{\pi m X_i}{l}\right)\cos\left(\frac{\pi n Y_i}{h}\right),\\ &\cos\left(\frac{\pi m X_i}{l}\right)\sin\left(\frac{\pi n Y_i}{h}\right), \sin\left(\frac{\pi m X_i}{l}\right)\sin\left(\frac{\pi n Y_i}{h}\right), ...\Big)\end{aligned} \tag{3.5}$$

para $i = 1, 2, ..., N$ y

$$\vec{d} = (Z_1, ..., Z_N)^T \tag{3.6}$$

El sistema (3.3) se puede expresar como

$$\Delta^2 \mathcal{L}\vec{x} = \vec{b} \tag{3.7}$$

donde $\Delta^2\mathcal{L}$ es el Hessiano de $\mathcal{L}$ y $\vec{x}$ es el vector de coeficientes de Fourier

$$\vec{x} = (..., a_{ij}, b_{ij}, c_{ij}, d_{ij}, ...)^T. \tag{3.8}$$

en orden lexicográfico.

Teorema 3.0.1. *Existe una única solución al sistema lineal (3.7)*

Demostración. $\mathcal{L}(\vec{x})$ es una función cuadrática acotada por abajo por cero y cuyos coeficientes principales son todos estrictamente positivos[2]. Por lo tanto, la gráfica de $\mathcal{L}(\vec{x})$ es un paraboloide estrictamente convexo que tiene un único mínimo global. Por el criterio de la segunda derivada, el punto mínimo es la única solución al sistema (3.7). □

[2]asumimos que los coeficientes redundantes de Fourier son eliminados

Corolario 3.0.1. *La matriz $\Delta^2\mathcal{L}$ es simétrica y positivo definida.*

Demostración. $\Delta^2\mathcal{L}$ es claramente simétrica. Integrando (3.7) tenemos

$$\mathcal{L}(\vec{x}) = \frac{1}{2}\vec{x}^T\Delta^2\mathcal{L}\vec{x} - \vec{b}^T\vec{x} + \vec{c}$$

donde $\vec{c}$ es el vector constante que depende únicamente de Z_i. Sea $\tilde{x}$ la solución de $\Delta^2\mathcal{L}\vec{x} = \vec{b}$ y $p \neq \tilde{x}$ entonces [Sewchuk, 1994]

$$\mathcal{L}(p) = \mathcal{L}(\tilde{x}) + \frac{1}{2}(p - \tilde{x})^T\Delta^2\mathcal{L}(p - \tilde{x}).$$

Ya que $\mathcal{L}(\tilde{x}) < \mathcal{L}(p)$, se sigue que $(p - \tilde{x})^T\Delta^2\mathcal{L}(p - \tilde{x}) > 0$. Entonces, $\Delta^2\mathcal{L}$ es positivo definida. □

3.1. Calculando el Hessiano

Ahora procederemos a calcular $\Delta^2\mathcal{L}$. Dividimos el Hessiano en 16 bloques

$$\Delta^2\mathcal{L} = \begin{pmatrix} \frac{\partial\mathcal{L}}{\partial a_{i,j}\partial a_{m,n}} & \frac{\partial\mathcal{L}}{\partial a_{i,j}\partial b_{m,n}} & \frac{\partial\mathcal{L}}{\partial a_{i,j}\partial c_{m,n}} & \frac{\partial\mathcal{L}}{\partial a_{i,j}\partial d_{m,n}} \\ \frac{\partial\mathcal{L}}{\partial b_{i,j}\partial a_{m,n}} & \frac{\partial\mathcal{L}}{\partial b_{i,j}\partial b_{m,n}} & \frac{\partial\mathcal{L}}{\partial b_{i,j}\partial c_{m,n}} & \frac{\partial\mathcal{L}}{\partial b_{i,j}\partial d_{m,n}} \\ \frac{\partial\mathcal{L}}{\partial c_{i,j}\partial a_{m,n}} & \frac{\partial\mathcal{L}}{\partial c_{i,j}\partial b_{m,n}} & \frac{\partial\mathcal{L}}{\partial c_{i,j}\partial c_{m,n}} & \frac{\partial\mathcal{L}}{\partial c_{i,j}\partial d_{m,n}} \\ \frac{\partial\mathcal{L}}{\partial d_{i,j}\partial a_{m,n}} & \frac{\partial\mathcal{L}}{\partial d_{i,j}\partial b_{m,n}} & \frac{\partial\mathcal{L}}{\partial d_{i,j}\partial c_{m,n}} & \frac{\partial\mathcal{L}}{\partial d_{i,j}\partial d_{m,n}} \end{pmatrix} \tag{3.9}$$

Las entradas del Hessiano están dadas por (ver el Apéndice A)

$$\begin{aligned} \frac{\partial^2\mathcal{L}}{\partial a_{i,j}a_{m,n}} &= 4\lambda\frac{\pi^4}{l^2h^2}mnij\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi mx}{l})\sin(\frac{\pi ny}{h})\sin(\frac{\pi ix}{l})\sin(\frac{\pi jy}{h})dxdy \\ &+2\lambda\Big(\Big(\frac{\pi}{l}\Big)^4 m^2i^2 + \Big(\frac{\pi}{h}^4 n^2j^2\Big)\Big)\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi mx}{l})\cos(\frac{\pi ny}{h})\cos(\frac{\pi ix}{l})\cos(\frac{\pi jy}{h})dxdy \\ &+\frac{2}{N}\sum_{s=1}^{N}\cos(\frac{\pi mX_s}{l})\cos(\frac{\pi nY_s}{h})\cos(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h}) \end{aligned}$$

$$\frac{\partial^2\mathcal{L}}{\partial a_{i,j}b_{m,n}} = \frac{2}{N}\sum_{s=1}^{N}\sin(\frac{\pi mX_s}{l})\cos(\frac{\pi nY_s}{h})\cos(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

$$\frac{\partial^2 \mathcal{L}}{\partial a_{i,j} c_{m,n}} = \frac{2}{N} \sum_{s=1}^{N} \cos(\frac{\pi m X_s}{l}) \sin(\frac{\pi n Y_s}{h}) \cos(\frac{\pi i X_s}{l}) \cos(\frac{\pi j Y_s}{h})$$

$$\frac{\partial^2 \mathcal{L}}{\partial a_{i,j} d_{m,n}} = \frac{2}{N} \sum_{s=1}^{N} \sin(\frac{\pi m X_s}{l}) \sin(\frac{\pi n Y_s}{h}) \cos(\frac{\pi i X_s}{l}) \cos(\frac{\pi j Y_s}{h})$$

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial b_{i,j} b_{m,n}} &= 4\lambda \frac{\pi^4}{l^2 h^2} mnij \int_{-h}^{h} \int_{-l}^{l} \cos(\frac{\pi m x}{l}) \sin(\frac{\pi n y}{h}) \cos(\frac{\pi i x}{l}) \sin(\frac{\pi j y}{h}) dx dy \\ +2\lambda \Big(\Big(\frac{\pi}{l}\Big)^4 m^2 i^2 + \Big(\frac{\pi}{h}\Big)^4 n^2 j^2 \Big) &\int_{-h}^{h} \int_{-l}^{l} \sin(\frac{\pi m x}{l}) \cos(\frac{\pi n y}{h}) \sin(\frac{\pi i x}{l}) \cos(\frac{\pi j y}{h}) dx dy \\ &+ \frac{2}{N} \sum_{s=1}^{N} \sin(\frac{\pi m X_s}{l}) \cos(\frac{\pi n Y_s}{h}) \sin(\frac{\pi i X_s}{l}) \cos(\frac{\pi j Y_s}{h}) \end{aligned}$$

$$\frac{\partial^2 \mathcal{L}}{\partial b_{i,j} c_{m,n}} = \frac{2}{N} \sum_{s=1}^{N} \cos(\frac{\pi m X_s}{l}) \sin(\frac{\pi n Y_s}{h}) \sin(\frac{\pi i X_s}{l}) \cos(\frac{\pi j Y_s}{h})$$

$$\frac{\partial^2 \mathcal{L}}{\partial b_{i,j} d_{m,n}} = \frac{2}{N} \sum_{s=1}^{N} \sin(\frac{\pi m X_s}{l}) \sin(\frac{\pi n Y_s}{h}) \sin(\frac{\pi i X_s}{l}) \cos(\frac{\pi j Y_s}{h})$$

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial c_{i,j} c_{m,n}} &= 4\lambda \frac{\pi^4}{l^2 h^2} mnij \int_{-h}^{h} \int_{-l}^{l} \sin(\frac{\pi m x}{l}) \cos(\frac{\pi n y}{h}) \sin(\frac{\pi i x}{l}) \cos(\frac{\pi j y}{h}) dx dy \\ +2\lambda \Big(\Big(\frac{\pi}{l}\Big)^4 m^2 i^2 + \Big(\frac{\pi}{h}\Big)^4 n^2 j^2 \Big) &\int_{-h}^{h} \int_{-l}^{l} \cos(\frac{\pi m x}{l}) \sin(\frac{\pi n y}{h}) \cos(\frac{\pi i x}{l}) \sin(\frac{\pi j y}{h}) dx dy \\ &+ \frac{2}{N} \sum_{s=1}^{N} \cos(\frac{\pi m X_s}{l}) \sin(\frac{\pi n Y_s}{h}) \cos(\frac{\pi i X_s}{l}) \sin(\frac{\pi j Y_s}{h}) \end{aligned}$$

$$\frac{\partial^2 \mathcal{L}}{\partial c_{i,j} d_{m,n}} = \frac{2}{N} \sum_{s=1}^{N} \sin(\frac{\pi m X_s}{l}) \sin(\frac{\pi n Y_s}{h}) \cos(\frac{\pi i X_s}{l}) \sin(\frac{\pi j Y_s}{h})$$

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial d_{i,j} d_{m,n}} &= 4\lambda \frac{\pi^4}{l^2 h^2} mnij \int_{-h}^{h} \int_{-l}^{l} \cos(\frac{\pi m x}{l}) \cos(\frac{\pi n y}{h}) \cos(\frac{\pi i x}{l}) \cos(\frac{\pi j y}{h}) dx dy \\ +2\lambda \Big(\Big(\frac{\pi}{l}\Big)^4 m^2 i^2 + \Big(\frac{\pi}{h}^4 n^2 j^2\Big) \Big) &\int_{-h}^{h} \int_{-l}^{l} \sin(\frac{\pi m x}{l}) \sin(\frac{\pi n y}{h}) \sin(\frac{\pi i x}{l}) \sin(\frac{\pi j y}{h}) dx dy \\ &+ \frac{2}{N} \sum_{s=1}^{N} \sin(\frac{\pi m X_s}{l}) \sin(\frac{\pi n Y_s}{h}) \sin(\frac{\pi i X_s}{l}) \sin(\frac{\pi j Y_s}{h}) \end{aligned}$$

Las integrales son calculadas exactamente en las siguientes tablas

$\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi m x}{l})\cos(\frac{\pi n y}{h})\cos(\frac{\pi i x}{l})\cos(\frac{\pi j y}{h})dxdy =$	si
$4hl$	$m=n=i=j=0$
$4hl\frac{\sin(j\pi)}{j\pi}$	$m=n=i=0,\ j\neq 0$
$4hl\frac{\sin(i\pi)}{i\pi}$	$m=n=0,\ i\neq 0, j=0$
$4hl\frac{\sin(i\pi)\sin(j\pi)}{ij\pi^2}$	$m=n=0,\ i\neq 0, j\neq 0$
$4hl\frac{\sin(n\pi)}{n\pi}$	$m=0, n\neq 0, i=0, j=0$
$2hl+hl\frac{\sin(2j\pi)}{j\pi}$	$m=0, n\neq 0, i=0, j=n$
$4hjl\frac{\cos(n\pi)\sin(j\pi)}{(j^2-n^2)\pi} - 4hnl\frac{\sin(n\pi)\cos(j\pi)}{(j^2-n^2)\pi}$	$m=0; n\neq 0; i=0; j\neq n,0$
$4hl\frac{\sin(i\pi)\sin(n\pi)}{in\pi^2}$	$m=0, n\neq 0, i\neq 0, j=0$
$2hl\frac{\sin(i\pi)}{i\pi}+hl\frac{\sin(i\pi)\sin(2j\pi)}{ij\pi^2}$	$m=0, n\neq 0, i\neq 0, j=n$
$4hjl\frac{\cos(n\pi)\sin(i\pi)\sin(j\pi)}{i(j^2-n^2)\pi^2}$ $-4hln\frac{\cos(j\pi)\sin(i\pi)\sin(n\pi)}{i(j^2-n^2)\pi^2}$	$m=0; n\neq 0; i\neq 0; j\neq n,0$
$4hl\frac{\sin(m\pi)}{m\pi}$	$m\neq 0, n=0, i=0, j=0$
$4hl\frac{\sin(j\pi)\sin(m\pi)}{jm\pi^2}$	$m\neq 0, n=0, i=0, j\neq 0$
$2hl+hl\frac{\sin(2i\pi)}{i\pi}$	$m\neq 0, n=0, i=m, j=0$
$2hl\frac{\sin((i-m)\pi)}{((i-m)\pi)}+2hl\frac{\sin((i+m)\pi)}{(i+m)\pi}$	$m\neq 0,; n=0; i\neq m,0; j=0$
$2hl\frac{\sin(j\pi)}{j\pi}+hl\frac{\sin(2i\pi)\sin(j\pi)}{ij\pi^2}$	$m\neq 0, n=0, i=m, j\neq 0$
$2hl\frac{\sin(j\pi)\sin((i-m)\pi)}{j(i-m)\pi^2}+2hl\frac{\sin(j\pi)\sin((i+m)\pi)}{j(i+m)\pi^2}$	$m\neq 0; n=0; i\neq m,0; j\neq 0$
$4hl\frac{\sin(m\pi)\sin(n\pi)}{mn\pi^2}$	$m\neq 0, n\neq 0, i=0, j=0$
$2hl\frac{\sin(m\pi)}{m\pi}+hl\frac{\sin(2j\pi)\sin(m\pi)}{jm\pi^2}$	$m\neq 0, n\neq 0, i=0, j=n$
$4hjl\frac{\cos(n\pi)\sin(j\pi)\sin(m\pi)}{m(j^2-n^2)\pi^2} - 4hln\frac{\cos(j\pi)\sin(m\pi)\sin(n\pi)}{m(j^2-n^2)\pi^2}$	$m\neq 0; n\neq 0; i=0; j\neq n,0$
$2hl\frac{\sin(n\pi)}{n\pi}+hl\frac{\sin(2i\pi)\sin(n\pi)}{in\pi^2}$	$m\neq 0, n\neq 0, i=m, j=0$
$2hl\frac{\sin((i-m)\pi)\sin(n\pi)}{(i-m)n\pi^2}+2hl\frac{\sin((i+m)\pi)\sin(n\pi)}{(i+m)n\pi^2}$	$m\neq 0; n\neq 0; i\neq m,0; j=0$
$hl+hl\frac{\sin(2i\pi)}{2i\pi}+hl\frac{\sin(2j\pi)}{2j\pi}+hl\frac{\sin(2i\pi)\sin(2j\pi)}{4ij\pi^2}$	$m\neq 0, n\neq 0, i=m, j=n$
$2hjl\frac{\cos(n\pi)\sin(j\pi)}{(j^2-n^2)\pi}+hjl\frac{\cos(n\pi)\sin(2i\pi)\sin(j\pi)}{i(j^2-n^2)\pi}$ $-2hln\frac{\cos(j\pi)\sin(n\pi)}{(j^2-n^2)\pi}-hln\frac{\cos(j\pi)\sin(2i\pi)\sin(n\pi)}{i(j^2-n^2)\pi}$	$m\neq 0; n\neq 0; i=m; j\neq n,0$
$2hil\frac{\cos(m\pi)\sin(i\pi)}{(i-m)(i+m)\pi}+hil\frac{\cos(m\pi)\sin(i\pi)\sin(2j\pi)}{j(i-m)(i+m)\pi^2}$ $-2hlm\frac{\cos(i\pi)\sin(m\pi)}{(i-m)(i+m)\pi}-hlm\frac{\cos(i\pi)\sin(2j\pi)\sin(m\pi)}{j(i-m)(i+m)\pi^2}$	$m\neq 0; n\neq 0; i\neq m,0; j=n$
$4hijl\frac{\cos(m\pi)\cos(n\pi)\sin(i\pi)\sin(j\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hjlm\frac{\cos(i\pi)\cos(n\pi)\sin(j\pi)\sin(m\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hiln\frac{\cos(j\pi)\cos(m\pi)\sin(i\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $+4hlmn\frac{\cos(i\pi)\cos(j\pi)\sin(m\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$	$m\neq 0, n\neq 0, i\neq m, j\neq n$

Tabla 3.1: Valor de $\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi m x}{l})\cos(\frac{\pi n y}{h})\cos(\frac{\pi i x}{l})\cos(\frac{\pi j y}{h})dxdy$ para distintos valores de m, n, i, j

$\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi m x}{l})\cos(\frac{\pi n y}{h})\sin(\frac{\pi i x}{l})\cos(\frac{\pi j y}{h})dxdy =$	si
0	$m = 0$ o $i = 0$
$2hl - hl\frac{\sin(2i\pi)}{i\pi}$	$m \neq 0, n = 0, i = m, j = 0$
$2hl\frac{\sin((i-m)\pi)}{(i-m)\pi} - 2hl\frac{\sin((i+m)\pi)}{(i+m)\pi}$	$m \neq 0; n = 0; i \neq 0, m; j = 0$
$2hl\frac{\sin(j\pi)}{j\pi} - hl\frac{\sin(2i\pi)\sin(j\pi)}{ij\pi^2}$	$m \neq 0, n = 0, i = m, j \neq 0$
$2hl\frac{\sin(j\pi)\sin((i-m)\pi)}{j(i-m)\pi^2} - 2hl\frac{\sin(j\pi)\sin((i+m)\pi)}{j(i+m)\pi^2}$	$m \neq 0; n = 0; i \neq m, 0; j \neq 0$
$2hl\frac{\sin(n\pi)}{n\pi} - hl\frac{\sin(2i\pi)\sin(n\pi)}{in\pi^2}$	$m \neq 0, n \neq 0, i = m, j = 0$
$2hl\frac{\sin((i-m)\pi)\sin(n\pi)}{(i-m)n\pi^2} + -2hl\frac{\sin((i+m)\pi)\sin(n\pi)}{(i+m)n\pi^2}$	$m \neq 0; n \neq 0; i \neq m, 0; j = 0$
$hl - hl\frac{\sin(2i\pi)}{2i\pi} + hl\frac{\sin(2j\pi)}{2j\pi} - hl\frac{\sin(2i\pi)\sin(2j\pi)}{4ij\pi^2}$	$m \neq 0, n \neq 0, i = m, j = n$
$2hjl\frac{\cos(n\pi)\sin(j\pi)}{(j^2-n^2)\pi} - hjl\frac{\cos(n\pi)\sin(2i\pi)\sin(j\pi)}{i(j^2-n^2)\pi}$ $-2hln\frac{\cos(j\pi)\sin(n\pi)}{(j^2-n^2)\pi} + hln\frac{\cos(j\pi)\sin(2i\pi)\sin(n\pi)}{i(j^2-n^2)\pi}$	$m \neq 0; n \neq 0; i = m; j \neq n, 0$
$2hlm\frac{\cos(m\pi)\sin(i\pi)}{(i-m)(i+m)\pi} + hlm\frac{\cos(m\pi)\sin(i\pi)\sin(2j\pi)}{j(i-m)(i+m)\pi^2}$ $-2hil\frac{\cos(i\pi)\sin(m\pi)}{(i-m)(i+m)\pi} - hil\frac{\cos(i\pi)\sin(2j\pi)\sin(m\pi)}{j(i-m)(i+m)\pi^2}$	$m \neq 0; n \neq 0; i \neq m, 0; j = n$
$4hjlm\frac{\cos(m\pi)\cos(n\pi)\sin(i\pi)\sin(j\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hijl\frac{\cos(i\pi)\cos(n\pi)\sin(j\pi)\sin(m\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hlmn\frac{\cos(j\pi)\cos(m\pi)\sin(i\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $+4hiln\frac{\cos(i\pi)\cos(j\pi)\sin(m\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$	$m \neq 0; n \neq 0; i \neq m, 0; j \neq n, 0$

Tabla 3.2: Valor de $\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi m x}{l})\cos(\frac{\pi n y}{h})\sin(\frac{\pi i x}{l})\cos(\frac{\pi j y}{h})dxdy$ para distintos valores de m, n, i, j

$\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi m x}{l})\sin(\frac{\pi n y}{h})\cos(\frac{\pi i x}{l})\sin(\frac{\pi j y}{h})dxdy =$	si
0	$n=0$ o $j=0$
$2hl - hl\frac{\sin(2j\pi)}{j\pi}$	$m=0, n=j, i=0, j\neq 0$
$4hln\frac{\cos(n\pi)\sin(j\pi)}{(j^2-n^2)\pi} - 4hjl\frac{\cos(j\pi)\sin(n\pi)}{(j^2-n^2)\pi}$	$m=0; n\neq 0, j; i=0, m; j\neq 0$
$2hil\frac{\cos(m\pi)\sin(i\pi)}{(i-m)(i+m)\pi} - hil\frac{\cos(m\pi)\sin(i\pi)\sin(2j\pi)}{j(i-m)(i+m)\pi^2}$ $-2hlm\frac{\cos(i\pi)\sin(m\pi)}{(i-m)(i+m)\pi} + hlm\frac{\cos(i\pi)\sin(2j\pi)\sin(m\pi)}{j(i-m)(i+m)\pi^2}$	$m=0, n=j, i\neq 0, j\neq 0$
$4hln\frac{\cos(n\pi)\sin(i\pi)\sin(j\pi)}{i(j^2-n^2)\pi^2}-$ $4hjl\frac{\cos(j\pi)\sin(i\pi)\sin(n\pi)}{i(j^2-n^2)\pi^2}$	$m=0; n\neq 0; i\neq 0; j\neq 0, n$
$2hl\frac{\sin(m\pi)}{m\pi} - hl\frac{\sin(2j\pi)\sin(m\pi)}{jm\pi^2}$	$m\neq 0, n=j, i=0, j\neq 0$
$4hln\frac{\cos(n\pi)\sin(j\pi)\sin(m\pi)}{m(j^2-n^2)\pi^2}$ $4hjl\frac{\cos(j\pi)\sin(m\pi)\sin(n\pi)}{m(j^2-n^2)\pi^2}$	$m\neq 0; n\neq 0, j; i=0, j\neq 0$
$hl - hl\frac{\sin(2i\pi)}{2i\pi} - hl\frac{\sin(2j\pi)}{2j\pi} - hl\frac{\sin(2i\pi)\sin(2j\pi)}{4ij\pi^2}$	$m=i, n=j, i\neq 0, j\neq 0$
$2hln\frac{\cos(n\pi)\sin(j\pi)}{(j^2-n^2)\pi} + hln\frac{\cos(n\pi)\sin(2i\pi)\sin(j\pi)}{i(j^2-n^2)\pi}$ $-2hjl\frac{\cos(j\pi)\sin(n\pi)}{(j^2-n^2)\pi} - hjl\frac{\cos(j\pi)\sin(2i\pi)\sin(n\pi)}{i(j^2-n^2)\pi^2}$	$m=i; n\neq j, 0; i\neq 0; j\neq 0$
$4hiln\frac{\cos(m\pi)\cos(n\pi)\sin(i\pi)\sin(j\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hlmn\frac{\cos(i\pi)\cos(n\pi)\sin(j\pi)\sin(m\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hijl\frac{\cos(j\pi)\cos(m\pi)\sin(i\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $+4hjlm\frac{\cos(i\pi)\cos(j\pi)\sin(m\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$	$m\neq i, 0; n\neq j, 0; i\neq 0; j\neq 0$

Tabla 3.3: Valor de $\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi m x}{l})\sin(\frac{\pi n y}{h})\cos(\frac{\pi i x}{l})\sin(\frac{\pi j y}{h})dxdy$ para distintos valores de m, n, i, j

$\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi mx}{l})\sin(\frac{\pi ny}{h})\sin(\frac{\pi ix}{l})\sin(\frac{\pi jy}{h})dxdy =$	si
0	$m=0, n=0, i=0, j=0$
$hl - hl\frac{\sin(2i\pi)}{2i\pi} - hl\frac{\sin(2j\pi)}{2j\pi} + hl\frac{\sin(2i\pi)\sin(2j\pi)}{4ij\pi^2}$	$m \neq 0, n \neq 0, i = m, j = n$
$2hln\frac{\cos(n\pi)\sin(j\pi)}{(j^2-n^2)\pi} - hln\frac{\cos(n\pi)\sin(2i\pi)\sin(j\pi)}{i(j^2-n^2)\pi^2}$ $-2hjl\frac{\cos(j\pi)\sin(n\pi)}{(j^2-n^2)\pi} + hjl\frac{\cos(j\pi)\sin(2i\pi)\sin(n\pi)}{i(j^2-n^2)\pi^2}$	$m \neq 0; n \neq 0; i = m; j \neq n, 0$
$2hlm\frac{\cos(m\pi)\sin(i\pi)}{(i-m)(i+m)\pi} - hlm\frac{\cos(m\pi)\sin(i\pi)\sin(2j\pi)}{j(i-m)(i+m)\pi^2)}$ $-2hil\frac{\cos(i\pi)\sin(m\pi)}{(i-m)(i+m)\pi} + hil\frac{\cos(i\pi)\sin(2j\pi)\sin(m\pi)}{j(i-m)(i+m)\pi^2)}$	$m \neq 0; n \neq 0; i \neq m, 0; j = n$
$4hlmn\frac{\cos(m\pi)\cos(n\pi)\sin(i\pi)\sin(j\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hiln\frac{\cos(i\pi)\cos(n\pi)\sin(j\pi)\sin(m\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $-4hjln\frac{\cos(j\pi)\cos(m\pi)\sin(i\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$ $+4hijl\frac{\cos(i\pi)\cos(j\pi)\sin(m\pi)\sin(n\pi)}{(i-m)(i+m)(j^2-n^2)\pi^2}$	$m \neq 0, n \neq 0; i \neq m, 0; j \neq n, 0$

Tabla 3.4: Valor de $\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi mx}{l})\sin(\frac{\pi ny}{h})\sin(\frac{\pi ix}{l})\sin(\frac{\pi jy}{h})dxdy$ para distintos valores de m, n, i, j

Para ilustrar el orden lexicogáfico que le damos a los coeficientes de Fourier dividimos $\Delta^2\mathcal{L}$ en 16 boques

$$\Delta^2\mathcal{L} = \begin{pmatrix} He(1,1) & He(1,2) & He(1,3) & He(1,4) \\ He(2,1) & He(2,2) & He(2,3) & He(2,4) \\ He(3,1) & He(3,2) & He(3,3) & He(3,4) \\ He(4,1) & He(4,2) & He(4,3) & He(4,4) \end{pmatrix} \tag{3.10}$$

.

El bloque $He(1,2)$ contiene las derivadas

$$He(1,1)=\begin{pmatrix} AA(0,0) & AA(0,1) & ... & AA(0,K_2)\\ AA(1,0) & AA(1,1) & ... & AA(1,K_2)\\ \vdots & \vdots & \ddots & \vdots\\ AA(K_1,0) & AA(K_1,1) & ... & AA(K_1,K_2)\end{pmatrix} \tag{3.11}$$

donde los subbloques $AA(i,j)$ están definidos mediante

$$AA(0,0):=\begin{pmatrix} \frac{\partial^2\mathcal{L}}{\partial a_{0,0}^2} & \frac{\partial^2\mathcal{L}}{\partial a_{0,0}a_{0,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{0,0}a_{0,K_2}}\\ \frac{\partial^2\mathcal{L}}{\partial a_{0,1}a_{0,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{0,1}^2} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{0,1}a_{0,K_2}}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2\mathcal{L}}{\partial a_{0,K_2}a_{0,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{0,K2}a_{0,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{0,K_2}^2}\end{pmatrix}, \tag{3.12}$$

$$AA(1,0):=\begin{pmatrix} \frac{\partial^2\mathcal{L}}{\partial a_{1,0}a_{0,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{1,0}a_{0,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{1,0}a_{0,K_2}}\\ \frac{\partial^2\mathcal{L}}{\partial a_{1,1}a_{0,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{1,1}a_{0,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{1,1}a_{0,K_2}}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2\mathcal{L}}{\partial a_{1,K_2}a_{0,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{1,K2}a_{0,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{1,K_2}a_{0,K_2}}\end{pmatrix}, \tag{3.13}$$

$$AA(0,1);=\begin{pmatrix} \frac{\partial^2\mathcal{L}}{\partial a_{0,0}a_{1,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{0,0}a_{1,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{0,0}a_{1,K_2}}\\ \frac{\partial^2\mathcal{L}}{\partial a_{0,1}a_{1,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{1,1}^2} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{0,1}a_{1,K_2}}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2\mathcal{L}}{\partial a_{0,K_2}a_{1,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{0,K2}a_{1,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{0,K_2}a_{1,K_2}}\end{pmatrix}. \tag{3.14}$$

En general

$$AA(i,j)=\begin{pmatrix} \frac{\partial^2\mathcal{L}}{\partial a_{i,0}a_{j,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{i,0}a_{j,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{i,0}a_{j,K_2}}\\ \frac{\partial^2\mathcal{L}}{\partial a_{i,1}a_{j,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{i,1}a_{j,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{i,1}a_{j,K_2}}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2\mathcal{L}}{\partial a_{i,K_2}a_{j,0}} & \frac{\partial^2\mathcal{L}}{\partial a_{i,K2}a_{j,1}} & \cdots & \frac{\partial^2\mathcal{L}}{\partial a_{i,K_2}a_{j,K_2}}\end{pmatrix} \tag{3.15}$$

Los demás subbloques se definen de manera análoga.

El cálculo del Hessiano es muy costoso computacionalmente. Esto es porque cada entrada de la matriz depende de todos los datos. En el Apéndice B mostramos cómo paralelizar el cálculo del Hessiano usando la librería MPI y el lenguaje Fortran.

Capítulo 4

Resultados numéricos

4.1. Datos de Franke

El primer experimento consiste en aplicar los métodos en cuestión a 100 puntos (fig. 4.1) provenientes de la función de prueba de Franke (Fig. 4.2 a)

$$
\begin{aligned}
f(x,y) = {} & \frac{3}{4}\exp\Big[-\frac{(9x-2)^2+(9y-2)^2}{4}\Big] + \frac{3}{4}\exp\Big[-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}\Big] \\
& + \frac{1}{2}\exp\Big[-\frac{(9x-7)^2+(9y-3)^2}{4}\Big] - \frac{1}{5}\exp\Big[-(9x-4)^2-(9y-7)^2\Big].
\end{aligned}
$$

El conjunto de datos está disponible en la página [*Oleg Davydov's Home Page*]. Éstos datos se usaron por R. Franke para comparar varios métodos de aproximación de datos dispersos de su tiempo. Franke concluyó que los métodos basados en RBFs eran los mejores [Franke, 1979].

En la Fig. 4.2 mostramos las superficies producidas al aplicar los métodos expuestos.

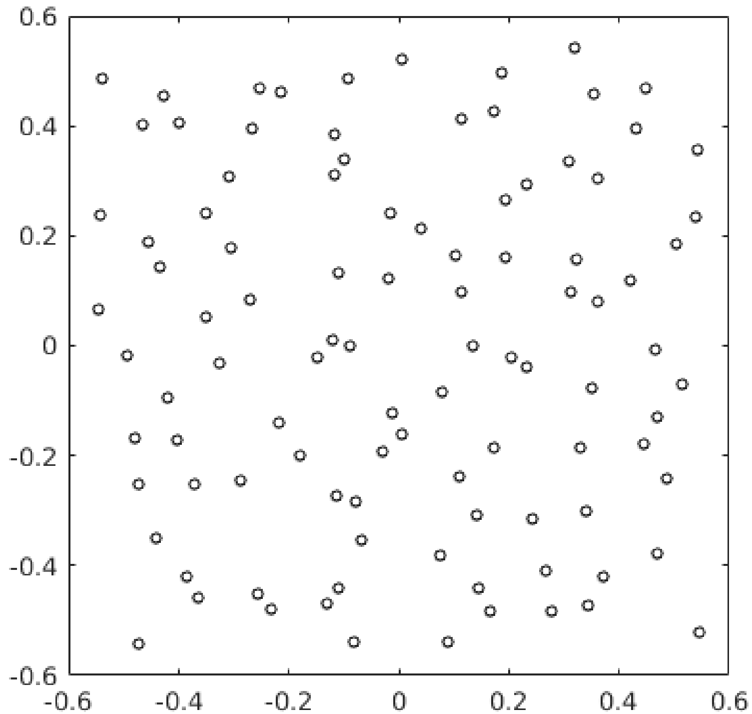


Figura 4.1: Datos de Franke

n_x	n_y	q	$M_{\text{mín}}$	$M_{\text{máx}}$	δ	D
5	5	3	20	100	0.5	3

Tabla 4.1: Parámetros usados en el método de Davydov para los datos de Franke

l	h	K_1	K_2	λ
1.8	1.8	20	20	1×10^{-7}

Tabla 4.2: Parámetros usados en el método propuesto en los datos de Franke

N
32

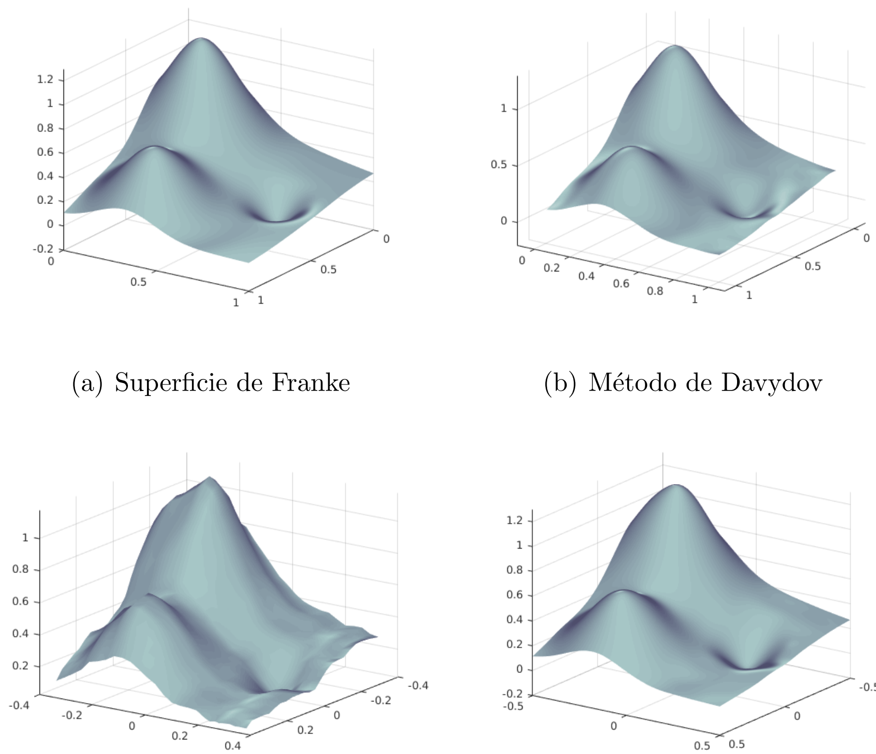
Tabla 4.3: Parámetro usado en el método de Potts para los datos de Franke

En la siguiente tabla listamos el máximo error, el promedio de los errores y la media cuadrática de los métodos en una malla de 41 por 41.

método	máx	promedio	media cuadrática
Potts	0.2651	0.0032	0.0192
Davydov	0.0188	0.0022	0.0035
propuesto	0.0376	0.0108	0.0135

Tabla 4.4: Error de aproximación de los métodos en una malla de tamaño 241 por 241

En la figura 4.2 mostramos las gráficas producidas al implementar los métodos con los parámetros antes mencionados. Visualmente el método de Davydov y el método propuesto reproducen muy bien a la gráfica de Franke. Una de las razones por las cuales el método propuesto no es tan preciso como el de Davydov es por que la función de Franke se compone de términos exponenciales que requieren muchos términos en la serie de Fourier y por la poca cantidad de datos.



(a) Superficie de Franke (b) Método de Davydov

(c) Método de Potts (d) Método propuesto en la tesis

Figura 4.2: Métodos en los datos de Franke

4.2. Datos de Valles curvos

El segundo experimento numérico contrasta con el primero en el sentido que se tiene mucha información disponible o, en otras palabras, los datos son densos. Consideramos la función de referencia "valles curvos"

$$f(x, y) = 0.5 \cos^4(4(x^2 + y - 1))$$

cuya gráfica se muestra en la figura (Fig. 4.3 b). Usamos 10,201 puntos de ésta función en una malla regular

l	h	K_1	K_2	λ
1.5	1.5	20	20	1×10^{-6}

Tabla 4.5: Parámetros usados por el método propuesto en los datos de valles curvos

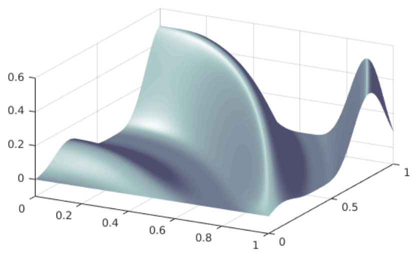
N
256

Tabla 4.6: Parámetro usado en el método de Potts [Kunis y Potts, 2007] para los valles curvos

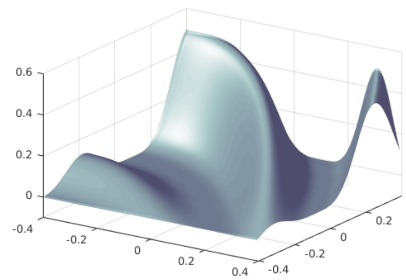
método	máx	promedio	media cuadrática
Potts	0.0490	0.0092	0.0053
propuesto	0.0096	3.4014×10^{-4}	6.9306×10^{-4}

Tabla 4.7: Error de aproximación de los métodos para valles curvos

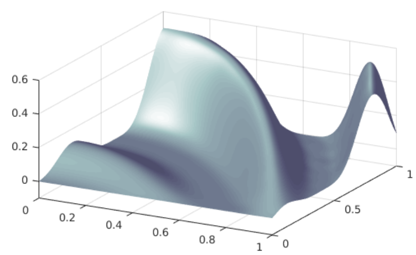
En la figura 4.3 comparamos el método de Potts y el método propuesto. Ambos funcionan muy bien cuando hay una cantidad abundante y bien distribuida de datos.



(a) Superficie de valles curvos



(b) método de Potts



(c) método propuesto en la tesis

Figura 4.3: Métodos en los datos de valles curvos

4.3. Datos del glaciar

En ésta sección usamos los 8345 puntos de referencia del "glaciar" disponibles en [*Oleg Davydov's Home Page*] bajo el nombre vol87. Éstos puntos se muestran en la figura 4.4 a. La figura 4.4 muestra el resultado de aplicar los métodos expuestos a dichos datos. La Fig. 4.5 muestra las curvas de nivel.

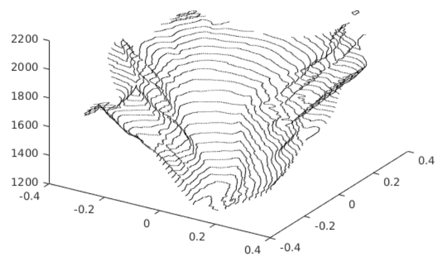
n_x	n_y	q	κ_H	$M_{\text{mín}}$	$M_{\text{máx}}$	δ	D
20	24	3	10^{-5}	60	160	0.4	3

Tabla 4.8: Parámetros usados por el método de Davydov en los datos del glaciar

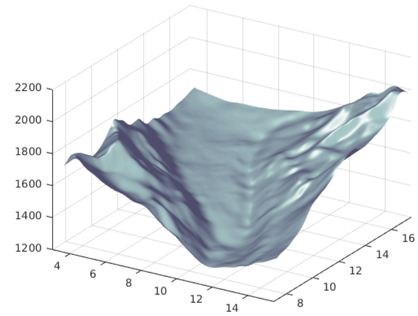
h	l	K_1	K_2	λ	Figura
1.2	1.2	30	30	1×10^{-8}	4.4 (d), 4.5 (d)
1.2	1.2	35	35	1×10^{-8}	4.4 (e), 4.5 (e)

Tabla 4.9: Parámetros usados por el método propuesto en los datos del glaciar

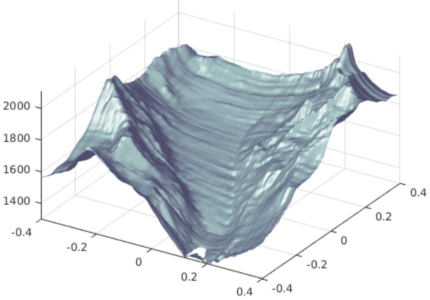
En las figuras 4.4 y 4.5 mostramos las superficies y curvas de nivel resultantes de aplicar los métodos expuestos. Las figuras 4.4, 4.5 (d) y (e) corresponden al método propuesto con distinto número de términos de Fourier. En las figuras 4.4 (d) y 4.5 (d) se usaron 3844 términos mientras que en las figuras 4.5 (e) y 4.5 (e) se usaron 5184 términos. Podemos observar que entre más términos de Fourier el método propuesto aproxima mejor los datos (figuras 4.5 (a), (d) y (e)) pero no es tan bueno como el método de Davydov (figura 4.5 (a) y (b)). Aunque el método de Potts interpola los datos la superficie entre éstos presenta oscilaciones (figura 4.4 (c))



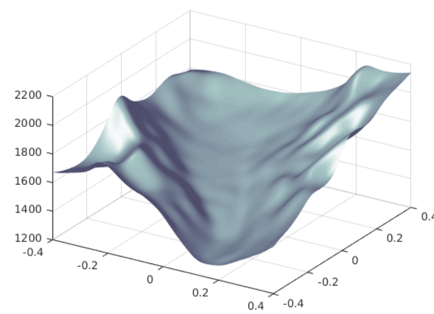
(a) Datos del glaciar



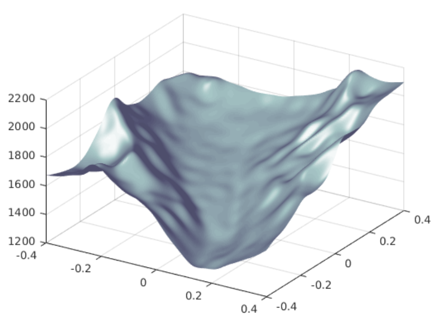
(b) método de Davydov



(c) método de Potts

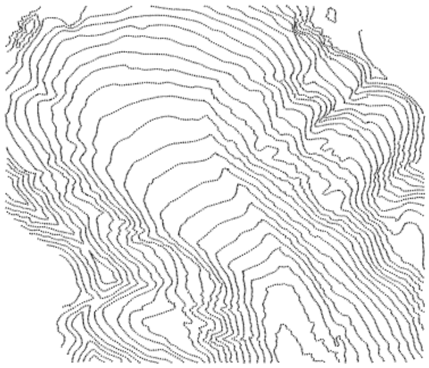


(d) método propuesto en la tesis

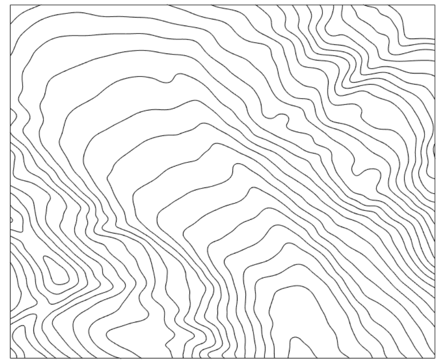


(e) método propuesto en la tesis

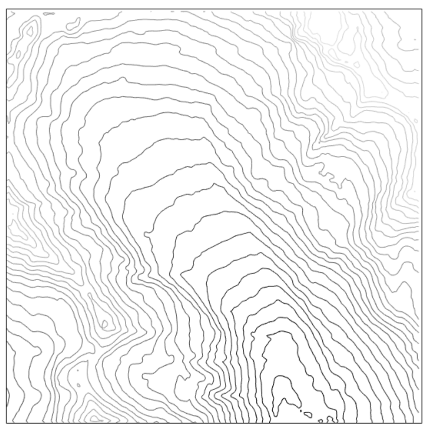
Figura 4.4: Métodos en los datos del glaciar



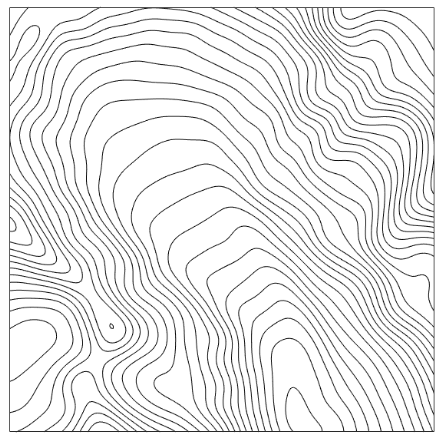
(a) datos del glaciar



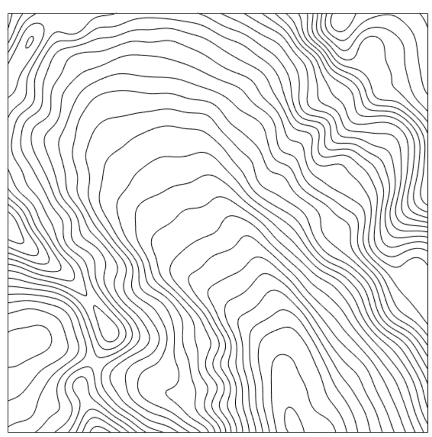
(b) método de Davydov



(c) método de Potts



(d) método propuesto en la tesis



(e) método propuesto en la tesis

Figura 4.5: Curvas de nivel

Capítulo 5

Conclusiones y trabajo futuro

Hemos presentado un método global para el problema de aproximación de datos dispersos basado en series de Fourier y que minimiza la rugosidad de la superficie aproximante y considera que los datos pueden contener ruido. El método propuesto es comparado con dos métodos recientes que abordan el mismo problema. El primero es global y también se basa en series de Fourier pero no minimiza la rugosidad de la superficie y considera que los datos no tienen ruido. El segundo método es local y se basa en splines y RBFs. Otras diferencias entre el primer método y el método propuesto en la tesis es en la representación de las series de Fourier y los algoritmos usados.

Podemos concluir en primer lugar acerca de la eficiencia de los métodos. El método propuesto es más costoso computacionalmente que los otros dos. El costo computacional aumenta con el número de términos de la serie pero da lugar a mejores aproximaciones. Sin embargo, éste se puede paralelizar muy fácilmente y no tiene tiempo muerto. El cómputo de una entrada del Hessiano no depende del valor de ninguna otra entrada. Cada valor se puede calcular de manera independiente siempre y cuando se tenga a disposición un número suficiente de procesadores.

En segundo lugar concluimos que el método propuesto es mucho más simple que los otros dos. Además el número de parámetros depende de la complejidad de la función cuyos datos observables se van a aproximar. Entonces una función simple requiere pocos parámetros sin importar el número de datos disponibles. Un número mayor de datos sólo incrementaría el costo computacional del método. En contraste, los otros dos métodos dependen de muchos parámetros cuyo número es proporcional al número de datos. Ésta característica nos permite, en nuestro método, comprimir los datos usando solamente los coeficientes de Fourier calculados. Cualquier valor de la función aproximante se puede calcular con sus coeficientes de Fourier.

Los experimentos numéricos muestran que el método propuesto es más preciso que el método de Potts y comparable con el método de Davydov.

5.1. Trabajo futuro

En el trabajo futuro creemos que lo que se tiene que hacer para mejorar el método es:

Primero, generalizarlo a dimensiones mayores usando la notación exponencial de las series de Fourier.

Segundo, después de haber eliminado las filas y columnas iguales a cero del Hessiano podemos incorporar métodos iterativos como el método de gradiente conjugado para resolver el sistema lineal.

Tercero, se pueden incorporar algoritmos para calcular el parámetro de suavidad como validación cruzada o la curva L. Éste parámetro tiene un impacto considerable en la solución pues controla la rugosidad y la distancia de la superficie aproximante a los puntos.

Finalmente, podemos derivar cotas del error del método propuesto como aquellas disponibles en el método de Davydov y los métodos globales basados en RBFs.

Apéndice A

Simplificando el sistema lineal

Recordemos que la función aproximante se expresa como

$$f(x,y) := \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} \left\{ a_{m,n} \cos\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) + b_{m,n} \sin\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \right.$$
$$\left. + c_{m,n} \cos\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) + d_{m,n} \sin\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \right\}$$

y el problema consiste en minimizar

$$\min_f \left\{ \frac{1}{N} \sum_{i=1}^{N} (f(X_i, Y_i) - Z_i)^2 + \lambda \int_{-h}^{h} \int_{-l}^{l} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy) \right\} \tag{A.1}$$

Si definimos

$$\mathcal{L}(\text{coefficients}) = \frac{1}{N} \sum_{i=1}^{N} (f(X_i, Y_i) - Z_i)^2 + \lambda \int_{-h}^{h} \int_{-l}^{l} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy$$

entonces el punto mínimo es la solución del sistema lineal

$$\frac{\partial \mathcal{L}}{\partial a_{m,n}} = 0$$
$$\frac{\partial \mathcal{L}}{\partial b_{m,n}} = 0$$
$$\frac{\partial \mathcal{L}}{\partial c_{m,n}} = 0$$
$$\frac{\partial \mathcal{L}}{\partial d_{m,n}} = 0$$

Separamos la sumatoria en (A) de la siguiente manera

$$\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} = \{m=0, n=0\} + \{m=0\}\sum_{n=1}^{K_2} + \sum_{m=1}^{K_1}\{n=0\} + \sum_{m=1}^{K_1}\sum_{n=1}^{K_2}$$

$$\begin{aligned}
f(x,y) &:= a_{0,0} \\
&+\sum_{n=1}^{K_2}\Big\{a_{0,n}\cos\left(\frac{\pi ny}{h}\right) + c_{0,n}\sin\left(\frac{\pi ny}{h}\right)\Big\} + \sum_{m=1}^{K_1}\Big\{a_{m,0}\cos\left(\frac{\pi mx}{l}\right) + b_{m,0}\sin\left(\frac{\pi mx}{l}\right)\Big\} \\
&+\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\Big\{a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) + b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) \\
&+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right) + d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big\}
\end{aligned}$$

La derivada respecto a la variable x es

$$\begin{aligned}
f_x(x,y) &= \sum_{m=1}^{K_1}\frac{\pi m}{l}\Big\{-a_{m,0}\sin\left(\frac{\pi mx}{l}\right) + b_{m,0}\cos\left(\frac{\pi mx}{l}\right)\Big\} \\
&+\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi m}{l}\Big\{-a_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) + b_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) \\
&-c_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right) + d_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big\}
\end{aligned}$$

o

$$\begin{aligned}
f_x(x,y) &:= \\
\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\frac{\pi m}{l}\Big\{&-a_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) + b_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) \\
&-c_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right) + d_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big\}
\end{aligned}$$

La segunda derivada de $f(x,y)$ respecto a x es

$$f_{xx}=\sum_{m=1}^{K_1}\left(\frac{\pi m}{l}\right)^2\Big\{-a_{m,0}\cos\left(\frac{\pi mx}{l}\right)-b_{m,0}\sin\left(\frac{\pi mx}{l}\right)\Big\}$$
$$+\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}-\left(\frac{\pi m}{l}\right)^2\Big[a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)+b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)+d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big]$$

o

$$f_{xx}=\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}-\left(\frac{\pi m}{l}\right)^2\Big[a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)+b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)+d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big]$$

Luego, la derivada de $f(x,y)$ respecto a y es

$$f_{y}=\sum_{n=1}^{K_2}\frac{\pi n}{h}\Big\{-a_{0,n}\sin\left(\frac{\pi ny}{h}\right)+c_{0,n}\cos\left(\frac{\pi ny}{h}\right)\Big\}$$
$$+\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi n}{h}\Big[-a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)-b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)+d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\Big]$$

o

$$f_{y}=\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\frac{\pi n}{h}\Big[-a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)-b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)+d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\Big]$$

La segunda derivada de $f(x,y)$ respecto a y es

$$f_{yy}=\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\Big\{-a_{0,n}\cos\left(\frac{\pi ny}{h}\right)-c_{0,n}\sin\left(\frac{\pi ny}{h}\right)\Big\}$$
$$\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}-\left(\frac{\pi n}{h}\right)^2\Big[a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)+b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)+d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big]$$

o

$$f_{yy}=\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}-\left(\frac{\pi n}{h}\right)^2\Big[a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)+b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)+d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big]$$

La derivada mixta de $f(x,y)$ es

$$f_{xy} = \sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{h}\right)\Big[a_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right) - b_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)$$
$$-c_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) + d_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\Big]$$

Para calcular $f_{xx}(x,y)^2$ recordemos que

$$f_{xx} = \sum_{m=1}^{K_1}\sum_{n=0}^{K_2}-\left(\frac{\pi m}{l}\right)^2\Big[a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) + b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right) + d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big]$$

entonces

$$
\begin{aligned}
f_{xx}(x,y)^2 = \sum_{m,r=1}^{K_1}\sum_{n,s=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi r}{l}\right)^2\Big[\\
& a_{m,n}a_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + a_{m,n}b_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + a_{m,n}c_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + a_{m,n}d_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + b_{m,n}a_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + b_{m,n}b_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + b_{m,n}c_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + b_{m,n}d_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + c_{m,n}a_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + c_{m,n}b_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + c_{m,n}c_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + c_{m,n}d_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + d_{m,n}a_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + d_{m,n}b_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + d_{m,n}c_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + d_{m,n}d_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& \Big]
\end{aligned}
$$

Integrando sobre el rectángulo $[-l,l]\times[-h,h]$ y luego derivando respecto a $a_{i,j}$ obtenemos

$$\frac{\partial}{\partial a_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xx}(x,y)^2 dxdy =$$

$$\sum_{r=1}^{K_1}\sum_{s=0}^{K_2}\left(\frac{\pi i}{l}\right)^2\left(\frac{\pi r}{l}\right)^2 a_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right)dxdy$$
$$+\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$
$$+\sum_{r=1}^{K_1}\sum_{s=0}^{K_2}\left(\frac{\pi i}{l}\right)^2\left(\frac{\pi r}{l}\right)^2 b_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right)dxdy$$
$$+\sum_{r=1}^{K_1}\sum_{s=0}^{K_2}\left(\frac{\pi i}{l}\right)^2\left(\frac{\pi r}{l}\right)^2 c_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right)dxdy$$
$$+\sum_{r=1}^{K_1}\sum_{s=0}^{K_2}\left(\frac{\pi i}{l}\right)^2\left(\frac{\pi r}{l}\right)^2 d_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right)dxdy$$
$$+\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$
$$+\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$
$$+\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$

Combinando los términos

$$\frac{\partial}{\partial a_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xx}(x,y)^2 dxdy =$$

$$2\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$
$$+2\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$
$$+2\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$
$$+2\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi m}{l}\right)^2\left(\frac{\pi i}{l}\right)^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$

Muchas de las integrales en la ecuación anterior son cero porque su integrando es una función impar en x o en y y el intervalo de integración correspondiente es simétrico. entonces la ecuación anterior se reduce a

$$\frac{\partial}{\partial a_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{xx}(x,y)^2 dxdy =$$
$$2 \sum_{m=1}^{K_1} \sum_{n=0}^{K_2} \left(\frac{\pi m}{l}\right)^2 \left(\frac{\pi i}{l}\right)^2 a_{m,n} \int_{-h}^{h} \int_{-l}^{l} \cos\left(\frac{\pi mx}{l}\right) \cos\left(\frac{\pi ny}{h}\right) \cos\left(\frac{\pi ix}{l}\right) \cos\left(\frac{\pi jy}{h}\right) dxdy \tag{A.2}$$

Análogamente

$$\frac{\partial}{\partial b_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{xx}(x,y)^2 dxdy =$$
$$2 \sum_{m=1}^{K_1} \sum_{n=0}^{K_2} \left(\frac{\pi m}{l}\right)^2 \left(\frac{\pi i}{l}\right)^2 b_{m,n} \int_{-h}^{h} \int_{-l}^{l} \sin\left(\frac{\pi mx}{l}\right) \cos\left(\frac{\pi ny}{h}\right) \sin\left(\frac{\pi ix}{l}\right) \cos\left(\frac{\pi jy}{h}\right) dxdy, \tag{A.3}$$

$$\frac{\partial}{\partial c_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{xx}(x,y)^2 dxdy =$$
$$2 \sum_{m=1}^{K_1} \sum_{n=0}^{K_2} \left(\frac{\pi m}{l}\right)^2 \left(\frac{\pi i}{l}\right)^2 c_{m,n} \int_{-h}^{h} \int_{-l}^{l} \cos\left(\frac{\pi mx}{l}\right) \sin\left(\frac{\pi ny}{h}\right) \cos\left(\frac{\pi ix}{l}\right) \sin\left(\frac{\pi jy}{h}\right) dxdy, \tag{A.4}$$

y

$$\frac{\partial}{\partial d_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{xx}(x,y)^2 dxdy =$$
$$2 \sum_{m=1}^{K_1} \sum_{n=0}^{K_2} \left(\frac{\pi m}{l}\right)^2 \left(\frac{\pi i}{l}\right)^2 d_{m,n} \int_{-h}^{h} \int_{-l}^{l} \sin\left(\frac{\pi mx}{l}\right) \sin\left(\frac{\pi ny}{h}\right) \sin\left(\frac{\pi ix}{l}\right) \sin\left(\frac{\pi jy}{h}\right) dxdy \tag{A.5}$$

Recordemos que

$$f_{xy} = \sum_{m=1}^{K_1} \sum_{n=1}^{K_2} \left(\frac{\pi m}{l}\right) \left(\frac{\pi n}{h}\right) \Big[a_{m,n} \sin\left(\frac{\pi mx}{l}\right) \sin\left(\frac{\pi ny}{h}\right) - b_{m,n} \cos\left(\frac{\pi mx}{l}\right) \sin\left(\frac{\pi ny}{h}\right)$$
$$- c_{m,n} \sin\left(\frac{\pi mx}{l}\right) \cos\left(\frac{\pi ny}{h}\right) + d_{m,n} \cos\left(\frac{\pi mx}{l}\right) \cos\left(\frac{\pi ny}{h}\right) \Big]$$

Entonces $f_{xy}(x,y)^2$ está dada por

$$
\begin{aligned}
f_{xy}(x,y)^2 = \sum_{m,r=1}^{K_1}\sum_{n,s=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi r}{l}\right)\left(\frac{\pi s}{l}\right)\Bigg[\\
& a_{m,n}a_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& - a_{m,n}b_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& - a_{m,n}c_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + a_{m,n}d_{r,s}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& - b_{m,n}a_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + b_{m,n}b_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + b_{m,n}c_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& - b_{m,n}d_{r,s}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& - c_{m,n}a_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + c_{m,n}b_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& + c_{m,n}c_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& - c_{m,n}d_{r,s}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + d_{m,n}a_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& - d_{m,n}b_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right) \\
& - d_{m,n}c_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& + d_{m,n}d_{r,s}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right) \\
& \Bigg]
\end{aligned}
$$

Integrando en el rectángulo $[-l, l] \times [-h, h]$ y después derivando respecto a $a_{i,j}$ obtenemos

$$
\begin{aligned}
&\frac{\partial}{\partial a_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xy}(x,y)^2 dxdy = \\
&\sum_{r=1}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)\left(\frac{\pi r}{l}\right)\left(\frac{\pi s}{l}\right)a_{r,s}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right)dxdy \\
&+\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
&-\sum_{r=1}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)\left(\frac{\pi r}{l}\right)\left(\frac{\pi s}{l}\right)b_{r,s}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right)dxdy \\
&-\sum_{r=1}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)\left(\frac{\pi r}{l}\right)\left(\frac{\pi s}{l}\right)c_{r,s}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right)dxdy \\
&+\sum_{r=1}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)\left(\frac{\pi r}{l}\right)\left(\frac{\pi s}{l}\right)d_{r,s}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right)dxdy \\
&-\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
&-\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
&+\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy
\end{aligned}
$$

combinando los términos

$$
\begin{aligned}
&\frac{\partial}{\partial a_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xy}(x,y)^2 dxdy = \\
&2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
&-2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
&-2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
&+2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right)d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy
\end{aligned}
$$

Muchos de los términos en la ecuación anterior son iguales a cero pues su integrando es una función impar respecto a x o respecto a y y el dominio de integración es simétrico. Eliminando dichos términos obtenemos

$$\begin{aligned}&\frac{\partial}{\partial a_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xy}(x,y)^2 dxdy = \\ 2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right) a_{m,n}&\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right) dxdy\end{aligned} \tag{A.6}$$

Análogamente

$$\begin{aligned}&\frac{\partial}{\partial b_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xy}(x,y)^2 dxdy = \\ 2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right) b_{m,n}&\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right) dxdy,\end{aligned} \tag{A.7}$$

$$\begin{aligned}&\frac{\partial}{\partial c_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xy}(x,y)^2 dxdy = \\ 2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right) c_{m,n}&\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right) dxdy,\end{aligned} \tag{A.8}$$

y

$$\begin{aligned}&\frac{\partial}{\partial d_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{xy}(x,y)^2 dxdy = \\ 2\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi m}{l}\right)\left(\frac{\pi n}{l}\right)\left(\frac{\pi i}{l}\right)\left(\frac{\pi j}{l}\right) d_{m,n}&\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right) dxdy\end{aligned} \tag{A.9}$$

Ahora calculemos los términos correspondientes a f_{yy}. Recordemos

$$\begin{aligned}f_{yy} = \sum_{m=0}^{K_1}\sum_{n=1}^{K_2} -\left(\frac{\pi n}{h}\right)^2\Big[&a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) + b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\\ &+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right) + d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big]\end{aligned}$$

$f_{yy}(x,y)^2$ está dado por

$$
\begin{aligned}
f_{yy}(x,y)^2 = \sum_{m,r=0}^{K_1} \sum_{n,s=1}^{K_2} \left(\frac{\pi n}{h}\right)^2 \left(\frac{\pi s}{h}\right)^2 \Big[\\
& a_{m,n} a_{r,s} \cos\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + a_{m,n} b_{r,s} \cos\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + a_{m,n} c_{r,s} \cos\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& + a_{m,n} d_{r,s} \cos\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& + b_{m,n} a_{r,s} \sin\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + b_{m,n} b_{r,s} \sin\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + b_{m,n} c_{r,s} \sin\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& + b_{m,n} d_{r,s} \sin\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& + c_{m,n} a_{r,s} \cos\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + c_{m,n} b_{r,s} \cos\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + c_{m,n} c_{r,s} \cos\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& + c_{m,n} d_{r,s} \cos\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& + d_{m,n} a_{r,s} \sin\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + d_{m,n} b_{r,s} \sin\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \cos\left(\frac{\pi s y}{h}\right) \\
& + d_{m,n} c_{r,s} \sin\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& + d_{m,n} d_{r,s} \sin\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi r x}{l}\right) \sin\left(\frac{\pi s y}{h}\right) \\
& \Big]
\end{aligned}
$$

Integrando sobre $[-l,l]\times[-h,h]$ y luego derivando respecto a $a_{i,j}$ obtenemos

$$
\begin{aligned}
&\frac{\partial}{\partial a_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{yy}(x,y)^2 dxdy = \\
&\sum_{r=0}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi j}{h}\right)^2\left(\frac{\pi s}{h}\right)^2 a_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right)dxdy \\
&+\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
&+\sum_{r=0}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi j}{h}\right)^2\left(\frac{\pi s}{h}\right)^2 b_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\cos\left(\frac{\pi sy}{h}\right)dxdy \\
&+\sum_{r=0}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi j}{h}\right)^2\left(\frac{\pi s}{h}\right)^2 c_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\cos\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right)dxdy \\
&+\sum_{r=0}^{K_1}\sum_{s=1}^{K_2}\left(\frac{\pi j}{h}\right)^2\left(\frac{\pi s}{h}\right)^2 d_{r,s}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)\sin\left(\frac{\pi rx}{l}\right)\sin\left(\frac{\pi sy}{h}\right)dxdy \\
&+\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
&+\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
&+\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy
\end{aligned}
$$

Combinando los términos de la última ecuación

$$
\begin{aligned}
&\frac{\partial}{\partial a_{i,j}}\int_{-h}^{h}\int_{-l}^{l} f_{yy}(x,y)^2 dxdy = \\
&2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
&+2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
&+2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
&+2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy
\end{aligned}
$$

Como antes, muchos términos en la ecuación anterior son iguales a cero pues sus integrandos son funciónes simétricas respecto a x o a y y el dominio de integración es simétrico. Eliminando estos términos obtenemos

$$\frac{\partial}{\partial a_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{yy}(x,y)^2 dxdy =$$
$$2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \tag{A.10}$$

Análogamente

$$\frac{\partial}{\partial b_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{yy}(x,y)^2 dxdy =$$
$$2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy, \tag{A.11}$$

$$\frac{\partial}{\partial c_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{yy}(x,y)^2 dxdy =$$
$$2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right) \tag{A.12}$$

y

$$\frac{\partial}{\partial d_{i,j}} \int_{-h}^{h} \int_{-l}^{l} f_{yy}(x,y)^2 dxdy =$$
$$2\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi n}{h}\right)^2\left(\frac{\pi j}{h}\right)^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \tag{A.13}$$

Usando las ecuaciones anteriores procedemos a calcular el gradiente de $\mathcal{L}$. Combinando las ecuaciones (A.2), (A.6) y (A.10) obtenemos

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial a_{i,j}} &= \frac{2}{N}\sum_{s=1}^{N}(f(X_s,Y_s)-Z_s)\cos\left(\frac{\pi i X_s}{l}\right)\cos\left(\frac{\pi j Y_s}{h}\right) \\
&+2\lambda\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi}{l}\right)^4 m^2 i^2 a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy \\
&+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnij a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy \\
&+2\lambda\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi}{h}\right)^4 n^2 j^2 a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy = \\
&\frac{2}{N}\sum_{s=1}^{N}(f(X_s,Y_s)-Z_s)\cos\left(\frac{\pi i X_s}{l}\right)\cos\left(\frac{\pi j Y_s}{h}\right)+ \\
2\lambda\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}&\left(\left(\frac{\pi}{l}\right)^4 m^2 i^2+\left(\frac{\pi}{h}\right)^4 n^2 j^2\right)a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy \\
&+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnij a_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy
\end{aligned}
$$

Sustituyendo la expresión que define a $f(x,y)$

$$
\begin{aligned}
f(x,y) := \sum_{m=0}^{K_1}\sum_{n=0}^{K_2}\Big\{ & a_{m,n}\cos\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)+b_{m,n}\sin\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right) \\
& +c_{m,n}\cos\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)+d_{m,n}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\Big\}
\end{aligned}
$$

obtenemos

$$\frac{\partial \mathcal{L}}{\partial a_{i,j}} =$$

$$4\lambda \sum_{m=1}^{K_1}\sum_{n=1}^{K_2} \frac{\pi^4}{l^2h^2} mnija_{m,n} \int_{-h}^{h}\int_{-l}^{l} \sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right) dxdy +$$

$$2\lambda \sum_{m=0}^{K_1}\sum_{n=0}^{K_2} \left(\left(\frac{\pi}{l}\right)^4 m^2i^2 + \left(\frac{\pi}{h}\right)^4 n^2j^2\right) a_{m,n} \int_{-h}^{h}\int_{-l}^{l} \cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right) dxdy$$

$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} a_{m,n}\cos\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} b_{m,n}\sin\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} c_{m,n}\cos\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} d_{m,n}\sin\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$-\frac{2}{N}\sum_{s=1}^{N} Z_s\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

Factorizando algunos términos

$$\frac{\partial \mathcal{L}}{\partial a_{i,j}} =$$

$$4\lambda \sum_{m=1}^{K_1}\sum_{n=1}^{K_2} \frac{\pi^4}{l^2h^2} mnija_{m,n} \int_{-h}^{h}\int_{-l}^{l} \sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right) dxdy +$$

$$2\lambda \sum_{m=0}^{K_1}\sum_{n=0}^{K_2} \left(\left(\frac{\pi}{l}\right)^4 m^2i^2 + \left(\frac{\pi}{h}\right)^4 n^2j^2\right) a_{m,n} \int_{-h}^{h}\int_{-l}^{l} \cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right) dxdy$$

$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} a_{m,n}\sum_{s=1}^{N}\cos\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} b_{m,n}\sum_{s=1}^{N}\sin\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} c_{m,n}\sum_{s=1}^{N}\cos\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} d_{m,n}\sum_{s=1}^{N}\sin\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

$$-\frac{2}{N}\sum_{s=1}^{N} Z_s\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

Factorizando los términos que dependen de $a_{m,n}$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial a_{i,j}} = \sum_{m=0}^{K_1}\sum_{n=0}^{K_2} a_{m,n}\Big[4\lambda\frac{\pi^4}{l^2h^2}mnij\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
+2\lambda\Big(\left(\frac{\pi}{l}\right)^4 m^2i^2 + \left(\frac{\pi}{h}\right)^4 n^2j^2\Big)\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
+\frac{2}{N}\sum_{s=1}^{N}\cos\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)\Big] \\
+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} b_{m,n}\sum_{s=1}^{N}\sin\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right) \\
+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} c_{m,n}\sum_{s=1}^{N}\cos\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right) \\
+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} d_{m,n}\sum_{s=1}^{N}\sin\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right) \\
-\frac{2}{N}\sum_{s=1}^{N} Z_s\cos\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)
\end{aligned} \tag{A.14}
$$

Ahora nos concentramos en la derivada de $\mathcal{L}$ respecto a $b_{i,j}$. combinando las ecuaciones (A.3), (A.7) y (A.11) obtenemos

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial b_{i,j}} = \frac{2}{N}\sum_{s=1}^{N}(f(X_s,Y_s)-Z_s)\sin\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right) \\
+2\lambda\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi}{l}\right)^4 m^2i^2 b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnijb_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy \\
+2\lambda\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi}{h}\right)^4 n^2j^2 b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy = \\
\frac{2}{N}\sum_{s=1}^{N}(f(X_s,Y_s)-Z_s)\sin\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right) \\
+2\lambda\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}\Big(\left(\frac{\pi}{l}\right)^4 m^2i^2 + \left(\frac{\pi}{h}\right)^4 n^2j^2\Big)b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy \\
+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnijb_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy
\end{aligned}
$$

Sustityendo

$$f(x,y) := \sum_{m=0}^{K_1}\sum_{n=0}^{K_2}\Big\{a_{m,n}\cos\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right) + b_{m,n}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)$$
$$+c_{m,n}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right) + d_{m,n}\sin\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\Big\}$$

en la expresión anterior obtenemos

$$\frac{\partial \mathcal{L}}{\partial b_{i,j}} =$$
$$4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnijb_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi mx}{l}\right)\sin\left(\frac{\pi ny}{h}\right)\cos\left(\frac{\pi ix}{l}\right)\sin\left(\frac{\pi jy}{h}\right)dxdy+$$
$$2\lambda\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}\left(\left(\frac{\pi}{l}\right)^4 m^2i^2 + \left(\frac{\pi}{h}\right)^4 n^2j^2\right)b_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi mx}{l}\right)\cos\left(\frac{\pi ny}{h}\right)\sin\left(\frac{\pi ix}{l}\right)\cos\left(\frac{\pi jy}{h}\right)dxdy$$
$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}a_{m,n}\sum_{s=1}^{N}\cos\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\sin\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$
$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}b_{m,n}\sum_{s=1}^{N}\sin\left(\frac{\pi mX_s}{l}\right)\cos\left(\frac{\pi nY_s}{h}\right)\sin\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$
$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}c_{m,n}\sum_{s=1}^{N}\cos\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\sin\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$
$$+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}d_{m,n}\sum_{s=1}^{N}\sin\left(\frac{\pi mX_s}{l}\right)\sin\left(\frac{\pi nY_s}{h}\right)\sin\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$
$$-\frac{2}{N}\sum_{s=1}^{N}Z_s\sin\left(\frac{\pi iX_s}{l}\right)\cos\left(\frac{\pi jY_s}{h}\right)$$

Factorizando los términos que dependen de $b_{m,n}$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial b_{i,j}} &= \frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} a_{m,n}\cos\left(\frac{\pi m X_s}{l}\right)\cos\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\cos\left(\frac{\pi j Y_s}{h}\right)\\
\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} b_{m,n}\Big[4\lambda\frac{\pi^4}{l^2h^2}mnij &\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy\\
+2\lambda\Big(\left(\frac{\pi}{l}\right)^4 m^2i^2+\left(\frac{\pi}{h}\right)^4 n^2j^2\Big) &\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy\\
&+\frac{2}{N}\sum_{s=1}^{N}\sin\left(\frac{\pi m X_s}{l}\right)\cos\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\cos\left(\frac{\pi j Y_s}{h}\right)\Big]\\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} c_{m,n}\sum_{s=1}^{N}\cos\left(\frac{\pi m X_s}{l}\right)\sin\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\cos\left(\frac{\pi j Y_s}{h}\right)\\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} d_{m,n}\sum_{s=1}^{N}\sin\left(\frac{\pi m X_s}{l}\right)\sin\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\cos\left(\frac{\pi j Y_s}{h}\right)\\
&-\frac{2}{N}\sum_{s=1}^{N} Z_s\sin\left(\frac{\pi i X_s}{l}\right)\cos\left(\frac{\pi j Y_s}{h}\right)
\end{aligned}
\tag{A.15}
$$

Ahora procedemos a calcular la derivada de $\mathcal{L}$ respecto a $c_{i,j}$. Combinando las ecuaciones (A.4), (A.8) y (A.12) obtenemos

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_{i,j}} &= \frac{2}{N}\sum_{s=1}^{N}(f(X_s,Y_s)-Z_s)\cos\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)\\
&+2\lambda\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi}{l}\right)^4 m^2i^2c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy\\
&+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnijc_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy\\
&+2\lambda\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi}{h}\right)^4 n^2j^2c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy =\\
&\frac{2}{N}\sum_{s=1}^{N}(f(X_s,Y_s)-Z_s)\cos\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)\\
&+2\lambda\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}\Big(\left(\frac{\pi}{l}\right)^4 m^2i^2+\left(\frac{\pi}{h}\right)^4 n^2j^2\Big)c_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy\\
&+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnijc_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy
\end{aligned}
$$

Sustituyendo la expresión de $f(x,y)$

$$\frac{\partial \mathcal{L}}{\partial c_{i,j}} =$$

$$4\lambda \sum_{m=1}^{K_1} \sum_{n=1}^{K_2} \frac{\pi^4}{l^2 h^2} mnij c_{m,n} \int_{-h}^{h} \int_{-l}^{l} \sin\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi i x}{l}\right) \cos\left(\frac{\pi j y}{h}\right) dxdy$$

$$2\lambda \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} \left(\left(\frac{\pi}{l}\right)^4 m^2 i^2 + \left(\frac{\pi}{h}\right)^4 n^2 j^2\right) c_{m,n} \int_{-h}^{h} \int_{-l}^{l} \cos\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi i x}{l}\right) \sin\left(\frac{\pi j y}{h}\right) dxdy$$

$$+\frac{2}{N} \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} a_{m,n} \sum_{s=1}^{N} \cos\left(\frac{\pi m X_s}{l}\right) \cos\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right)$$

$$+\frac{2}{N} \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} b_{m,n} \sum_{s=1}^{N} \sin\left(\frac{\pi m X_s}{l}\right) \cos\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right)$$

$$+\frac{2}{N} \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} c_{m,n} \sum_{s=1}^{N} \cos\left(\frac{\pi m X_s}{l}\right) \sin\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right)$$

$$+\frac{2}{N} \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} d_{m,n} \sum_{s=1}^{N} \sin\left(\frac{\pi m X_s}{l}\right) \sin\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right)$$

$$-\frac{2}{N} \sum_{s=1}^{N} Z_s \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right)$$

y factorizando los términos que dependen de $c_{m,n}$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_{i,j}} = & \frac{2}{N} \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} a_{m,n} \sum_{s=1}^{N} \cos\left(\frac{\pi m X_s}{l}\right) \cos\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right) \\
& + \frac{2}{N} \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} b_{m,n} \sum_{s=1}^{N} \sin\left(\frac{\pi m X_s}{l}\right) \cos\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right) \\
& + \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} c_{m,n} \Big[4\lambda \frac{\pi^4}{l^2 h^2} mnij \int_{-h}^{h} \int_{-l}^{l} \sin\left(\frac{\pi m x}{l}\right) \cos\left(\frac{\pi n y}{h}\right) \sin\left(\frac{\pi i x}{l}\right) \cos\left(\frac{\pi j y}{h}\right) dxdy \\
& + 2\lambda \left(\left(\frac{\pi}{l}\right)^4 m^2 i^2 + \left(\frac{\pi}{h}\right)^4 n^2 j^2\right) \int_{-h}^{h} \int_{-l}^{l} \cos\left(\frac{\pi m x}{l}\right) \sin\left(\frac{\pi n y}{h}\right) \cos\left(\frac{\pi i x}{l}\right) \sin\left(\frac{\pi j y}{h}\right) dxdy \\
& + \frac{2}{N} \sum_{s=1}^{N} \cos\left(\frac{\pi m X_s}{l}\right) \sin\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right) \Big] \\
& + \frac{2}{N} \sum_{m=0}^{K_1} \sum_{n=0}^{K_2} d_{m,n} \sum_{s=1}^{N} \sin\left(\frac{\pi m X_s}{l}\right) \sin\left(\frac{\pi n Y_s}{h}\right) \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right) \\
& - \frac{2}{N} \sum_{s=1}^{N} Z_s \cos\left(\frac{\pi i X_s}{l}\right) \sin\left(\frac{\pi j Y_s}{h}\right)
\end{aligned} \tag{A.16}$$

Ahora calculamos la derivada de $\mathcal{L}$ respecto a $d_{i,j}$. Combinando las ecuaciones (A.5), (A.9) y (A.13) obtenemos

$$\frac{\partial \mathcal{L}}{\partial d_{i,j}} = \frac{2}{N}\sum_{s=1}^{N}(f(X_s,Y_s)-Z_s)\sin\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)$$
$$+2\lambda\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi}{l}\right)^4 m^2 i^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy$$
$$+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnij d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy$$
$$+2\lambda\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi}{h}\right)^4 n^2 j^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy$$

Sustituyendo la expresión que define a $f(x,y)$

$$\frac{\partial \mathcal{L}}{\partial d_{i,j}} = 2\lambda\sum_{m=1}^{K_1}\sum_{n=0}^{K_2}\left(\frac{\pi}{l}\right)^4 m^2 i^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy$$
$$+4\lambda\sum_{m=1}^{K_1}\sum_{n=1}^{K_2}\frac{\pi^4}{l^2h^2}mnij d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\cos\left(\frac{\pi m x}{l}\right)\cos\left(\frac{\pi n y}{h}\right)\cos\left(\frac{\pi i x}{l}\right)\cos\left(\frac{\pi j y}{h}\right)dxdy$$
$$+2\lambda\sum_{m=0}^{K_1}\sum_{n=1}^{K_2}\left(\frac{\pi}{h}\right)^4 n^2 j^2 d_{m,n}\int_{-h}^{h}\int_{-l}^{l}\sin\left(\frac{\pi m x}{l}\right)\sin\left(\frac{\pi n y}{h}\right)\sin\left(\frac{\pi i x}{l}\right)\sin\left(\frac{\pi j y}{h}\right)dxdy$$
$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}a_{m,n}\cos\left(\frac{\pi m X_s}{l}\right)\cos\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)$$
$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}b_{m,n}\sin\left(\frac{\pi m X_s}{l}\right)\cos\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)$$
$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}c_{m,n}\cos\left(\frac{\pi m X_s}{l}\right)\sin\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)$$
$$+\frac{2}{N}\sum_{s=1}^{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2}d_{m,n}\sin\left(\frac{\pi m X_s}{l}\right)\sin\left(\frac{\pi n Y_s}{h}\right)\sin\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)$$
$$-\frac{2}{N}\sum_{s=1}^{N}Z_s\sin\left(\frac{\pi i X_s}{l}\right)\sin\left(\frac{\pi j Y_s}{h}\right)$$

o

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial d_{i,j}} = \\
&4\lambda \sum_{m=1}^{K_1}\sum_{n=1}^{K_2} \frac{\pi^4}{l^2h^2} mnij d_{m,n} \int_{-h}^{h}\int_{-l}^{l} \cos\Big(\frac{\pi m x}{l}\Big)\cos\Big(\frac{\pi n y}{h}\Big)\cos\Big(\frac{\pi i x}{l}\Big)\cos\Big(\frac{\pi j y}{h}\Big)dxdy \\
&+2\lambda \sum_{m=0}^{K_1}\sum_{n=0}^{K_2} \Big(\Big(\frac{\pi}{l}\Big)^4 m^2 i^2 + \Big(\frac{\pi}{h}\Big)^4 n^2 j^2\Big) d_{m,n} \int_{-h}^{h}\int_{-l}^{l} \sin\Big(\frac{\pi m x}{l}\Big)\sin\Big(\frac{\pi n y}{h}\Big)\sin\Big(\frac{\pi i x}{l}\Big)\sin\Big(\frac{\pi j y}{h}\Big)dxdy \\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} a_{m,n}\sum_{s=1}^{N}\cos\Big(\frac{\pi m X_s}{l}\Big)\cos\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big) \\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} b_{m,n}\sum_{s=1}^{N}\sin\Big(\frac{\pi m X_s}{l}\Big)\cos\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big) \\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} c_{m,n}\sum_{s=1}^{N}\cos\Big(\frac{\pi m X_s}{l}\Big)\sin\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big) \\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} d_{m,n}\sum_{s=1}^{N}\sin\Big(\frac{\pi m X_s}{l}\Big)\sin\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big) \\
&-\frac{2}{N}\sum_{s=1}^{N} Z_s \sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big)
\end{aligned}
$$

y factorizando los términos que dependen de $d_{m,n}$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial d_{i,j}} = &\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} a_{m,n}\sum_{s=1}^{N}\cos\Big(\frac{\pi m X_s}{l}\Big)\cos\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big) \\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} b_{m,n}\sum_{s=1}^{N}\sin\Big(\frac{\pi m X_s}{l}\Big)\cos\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big) \\
&+\frac{2}{N}\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} c_{m,n}\sum_{s=1}^{N}\cos\Big(\frac{\pi m X_s}{l}\Big)\sin\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big) \\
&+\sum_{m=0}^{K_1}\sum_{n=0}^{K_2} d_{m,n}\Big[4\lambda\frac{\pi^4}{l^2h^2} mnij \int_{-h}^{h}\int_{-l}^{l} \cos\Big(\frac{\pi m x}{l}\Big)\cos\Big(\frac{\pi n y}{h}\Big)\cos\Big(\frac{\pi i x}{l}\Big)\cos\Big(\frac{\pi j y}{h}\Big)dxdy \\
&+2\lambda\Big(\Big(\frac{\pi}{l}\Big)^4 m^2 i^2 + \Big(\frac{\pi}{h}\Big)^4 n^2 j^2\Big) \int_{-h}^{h}\int_{-l}^{l} \sin\Big(\frac{\pi m x}{l}\Big)\sin\Big(\frac{\pi n y}{h}\Big)\sin\Big(\frac{\pi i x}{l}\Big)\sin\Big(\frac{\pi j y}{h}\Big)dxdy \\
&+\frac{2}{N}\sum_{s=1}^{N}\sin\Big(\frac{\pi m X_s}{l}\Big)\sin\Big(\frac{\pi n Y_s}{h}\Big)\sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big)\Big] \\
&-\frac{2}{N}\sum_{s=1}^{N} Z_s \sin\Big(\frac{\pi i X_s}{l}\Big)\sin\Big(\frac{\pi j Y_s}{h}\Big)
\end{aligned} \tag{A.17}
$$

Derivamos (A.14) respecto a $a_{m,n}$, $b_{m,n}$, $c_{m,n}$ y $d_{m,n}$ para obtener

$$\frac{\partial^2\mathcal{L}}{\partial a_{i,j}a_{m,n}} = 4\lambda\frac{\pi^4}{l^2h^2}mnij\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi mx}{l})\sin(\frac{\pi ny}{h})\sin(\frac{\pi ix}{l})\sin(\frac{\pi jy}{h})dxdy$$
$$+2\lambda\Big(\Big(\frac{\pi}{l}\Big)^4 m^2i^2+\Big(\frac{\pi}{h}^4 n^2j^2\Big)\Big)\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi mx}{l})\cos(\frac{\pi ny}{h})\cos(\frac{\pi ix}{l})\cos(\frac{\pi jy}{h})dxdy$$
$$+\frac{2}{N}\sum_{s=1}^{N}\cos(\frac{\pi mX_s}{l})\cos(\frac{\pi nY_s}{h})\cos(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

$$\frac{\partial^2\mathcal{L}}{\partial a_{i,j}b_{m,n}} = \frac{2}{N}\sum_{s=1}^{N}\sin(\frac{\pi mX_s}{l})\cos(\frac{\pi nY_s}{h})\cos(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

$$\frac{\partial^2\mathcal{L}}{\partial a_{i,j}c_{m,n}} = \frac{2}{N}\sum_{s=1}^{N}\cos(\frac{\pi mX_s}{l})\sin(\frac{\pi nY_s}{h})\cos(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

$$\frac{\partial^2\mathcal{L}}{\partial a_{i,j}d_{m,n}} = \frac{2}{N}\sum_{s=1}^{N}\sin(\frac{\pi mX_s}{l})\sin(\frac{\pi nY_s}{h})\cos(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

Luego derivamos (A.15) respecto a $b_{m,n}$, $c_{m,n}$ y $d_{m,n}$ obteniendo

$$\frac{\partial^2\mathcal{L}}{\partial b_{i,j}b_{m,n}} = 4\lambda\frac{\pi^4}{l^2h^2}mnij\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi mx}{l})\sin(\frac{\pi ny}{h})\cos(\frac{\pi ix}{l})\sin(\frac{\pi jy}{h})dxdy$$
$$+2\lambda\Big(\Big(\frac{\pi}{l}\Big)^4 m^2i^2+\Big(\frac{\pi}{h}\Big)^4 n^2j^2\Big)\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi mx}{l})\cos(\frac{\pi ny}{h})\sin(\frac{\pi ix}{l})\cos(\frac{\pi jy}{h})dxdy$$
$$+\frac{2}{N}\sum_{s=1}^{N}\sin(\frac{\pi mX_s}{l})\cos(\frac{\pi nY_s}{h})\sin(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

$$\frac{\partial^2\mathcal{L}}{\partial b_{i,j}c_{m,n}} = \frac{2}{N}\sum_{s=1}^{N}\cos(\frac{\pi mX_s}{l})\sin(\frac{\pi nY_s}{h})\sin(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

$$\frac{\partial^2\mathcal{L}}{\partial b_{i,j}d_{m,n}} = \frac{2}{N}\sum_{s=1}^{N}\sin(\frac{\pi mX_s}{l})\sin(\frac{\pi nY_s}{h})\sin(\frac{\pi iX_s}{l})\cos(\frac{\pi jY_s}{h})$$

Después derivamos (A.16) respecto a $c_{m,n}$ y $d_{m,n}$ para obtener

$$\frac{\partial^2\mathcal{L}}{\partial c_{i,j}c_{m,n}} = 4\lambda\frac{\pi^4}{l^2h^2}mnij\int_{-h}^{h}\int_{-l}^{l}\sin(\frac{\pi mx}{l})\cos(\frac{\pi ny}{h})\sin(\frac{\pi ix}{l})\cos(\frac{\pi jy}{h})dxdy$$
$$+2\lambda\Big(\Big(\frac{\pi}{l}\Big)^4 m^2i^2+\Big(\frac{\pi}{h}\Big)^4 n^2j^2\Big)\int_{-h}^{h}\int_{-l}^{l}\cos(\frac{\pi mx}{l})\sin(\frac{\pi ny}{h})\cos(\frac{\pi ix}{l})\sin(\frac{\pi jy}{h})dxdy$$
$$+\frac{2}{N}\sum_{s=1}^{N}\cos(\frac{\pi mX_s}{l})\sin(\frac{\pi nY_s}{h})\cos(\frac{\pi iX_s}{l})\sin(\frac{\pi jY_s}{h})$$

$$\frac{\partial^2 \mathcal{L}}{\partial c_{i,j} d_{m,n}} = \frac{2}{N}\sum_{s=1}^{N} \sin(\frac{\pi m X_s}{l})\sin(\frac{\pi n Y_s}{h})\cos(\frac{\pi i X_s}{l})\sin(\frac{\pi j Y_s}{h})$$

Finalmente derivamos (A.17) respecto a $d_{m,n}$

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial d_{i,j} d_{m,n}} &= 4\lambda \frac{\pi^4}{l^2 h^2} mnij \int_{-h}^{h}\int_{-l}^{l} \cos(\frac{\pi m x}{l})\cos(\frac{\pi n y}{h})\cos(\frac{\pi i x}{l})\cos(\frac{\pi j y}{h}) dxdy \\
+2\lambda\Big(\Big(\frac{\pi}{l}\Big)^4 m^2 i^2 + \Big(\frac{\pi}{h}^4 n^2 j^2\Big)\Big) &\int_{-h}^{h}\int_{-l}^{l} \sin(\frac{\pi m x}{l})\sin(\frac{\pi n y}{h})\sin(\frac{\pi i x}{l})\sin(\frac{\pi j y}{h}) dxdy \\
&+\frac{2}{N}\sum_{s=1}^{N} \sin(\frac{\pi m X_s}{l})\sin(\frac{\pi n Y_s}{h})\sin(\frac{\pi i X_s}{l})\sin(\frac{\pi j Y_s}{h})
\end{aligned}$$

Notar que no es necesario calcular las otras entradas del Hessiano $\Delta^2 \mathcal{L}$ pues éste es simétrico.

Apéndice B

Código Fortran

El siguiente código calcula $\Delta^2\mathcal{L}$ y $\vec{b}$ en paralelo empleando MPI y Fortran usando las fórmulas de la sección anterior. Como la matriz es simétrica sólo se calculan los 10 bloques encima de la diagonal principal. El Hessiano es dividido en dos matrices

$$\Delta^2\mathcal{L} = He1 + \lambda He2 \tag{B.1}$$

donde $He2$ es la matriz que contiene las dobles integrales y $He1$ los demás términos. Esto es con la finalidad de calcular varias soluciones para distintos valores del parámetro de suavidad λ sin tener que calcular $\Delta^2\mathcal{L}$ otra vez.

El código usa 10 procesadores, indizados de 0 a 9, para calcular los 10 bloques encima de la diagonal principal de $\Delta^2\mathcal{L}$ (o más bien los 20 bloques, 10 de $He1$ y 10 de $He2$). Primero, en el procesador 0, leemos los datos del archivo "glacier.txt" los guardamos en la matriz D. El vector $\vec{b}$ se calcula en la variable b_vec y también se calcula el primer bloque de las matrices $He1$ y $He2$.

Luego, para los procesadores 1,...,9 se calculan los bloques correspondientes en la variable auxiliar $Hesub$ y se envían al procesador 0 para ser guardados en $He1$ y $He2$. Para saber que entradas de $He1$ y $He2$ le corresponde a cada procesador hacemos uso de la función

$$\phi(i, j, K_2) = i \cdot (K_2 + 1) + j + 1 \text{ for } 0 \leq i \leq K_1 \text{ and} \tag{B.2}$$

Ésta función calcula el índice de la fila (dados i y j) o el índice de la columna (dados m, n) de las matrices $He1$ y $He2$.

```fortran
program SDFFS
  use funcs_SDFFS
  use mpi
  implicit none

  integer, dimension (mpi_status_size) :: statusMPI
  integer :: mynode, tn, error_number

  integer, parameter :: K1 = 35
  integer, parameter :: K2 = 35
  integer, parameter :: NN = 8345 !This is the number of rows (points) in Data file
  integer :: i,j,m,n, aux
  real :: l,h, t1, t2 !Wonder if I can define this variables as parameters
  double precision :: pi !Wonder if I can define this variables as parameters
  double precision, dimension(:,:), allocatable :: D
  double precision, dimension(:,:), allocatable :: He1, He2
  double precision, dimension(:,:), allocatable :: Hesub1, Hesub2, Heaux
  double precision, dimension(:), allocatable :: b_vec

  pi = 4*ATAN(1.0)
  l=1.2 !I need to adjust this value
  h=1.2 !I need to adjust this value
  !lambda=0.00000000001 !I need to adjust this value
  aux = (K1+1)*(K2+1)

  call mpi_init(error_number)
  call mpi_comm_size(mpi_comm_world, tn, error_number)
  call mpi_comm_rank(mpi_comm_world, mynode, error_number)

  t1 = mpi_wtime()

  allocate(Hesub1(aux,aux))
  allocate(Hesub2(aux,aux))
  allocate(D(NN,3))

  !Read data form DataFranke.txt file. This has to be in the same directory as the source file
  open(unit=1,file="glacier.txt",status='old')
  do i=1,NN
     read(1,*) D(i,:)
  end do
  close(unit=1)

  if (mynode == 0) then
  allocate(b_vec(4*aux))
  allocate(Heaux(aux,aux))
  allocate(He1(4*aux,4*aux))
  allocate(He2(4*aux,4*aux))
  He1 = 0.0
  He2 = 0.0
    do i=0,K1
      do j=0,K2
      b_vec(phi(i,j,K2))=(2.0/NN)*SUM(D(:,3)*COS(pi*dble(i)*D(:,1)/l)*COS(pi*dble(j)*D(:,2)/h))
      b_vec(phi(i,j,K2)+aux)=(2.0/NN)*SUM(D(:,3)*SIN(pi*dble(i)*D(:,1)/l)*COS(pi*dble(j)*D(:,2)/h))
      b_vec(phi(i,j,K2)+2*aux)=(2.0/NN)*SUM(D(:,3)*COS(pi*dble(i)*D(:,1)/l)*SIN(pi*dble(j)*D(:,2)/h))
      b_vec(phi(i,j,K2)+3*aux)=(2.0/NN)*SUM(D(:,3)*SIN(pi*dble(i)*D(:,1)/l)*SIN(pi*dble(j)*D(:,2)/h))
      end do
    end do

    open(unit=0,file="b_vec.txt",ACTION="write")
    do j=1,4*(K1+1)*(K2+1)
```

```
      write(0,'(1000F14.7)' ) dble(b_vec(j))
    end do
    close(unit=0)
  deallocate(b_vec)
  print *,'rhs vector saved'
  end if

  if (mynode == 0) then
    do i = 0,K1
          do j = 0,K2
              do m = 0, K1
                  do n = 0,K2
                      !Delta^2L_{1,1} This matrix block contains partial^2L/(partial a_{i,j} partial a_{m,n})
                      He1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(COS(pi*m*D(:,1)*(l**(-1)))*COS(pi*n*D(:,2)*(h**(-1)))*&
                                              COS(pi*i*D(:,1)*(l**(-1)))*COS(pi*j*D(:,2)*(h**(-1))))
                He2(phi(i,j,K2),phi(m,n,K2))=4*(pi**4)/((l**2)*(h**2))*m*n*i*j*int_sin4(m,n,i,j,l,h)+&
                                              2*((pi/l)**4*m**2*i**2 + (pi/h)**4*(n**2)*(j**2) )*int_cos4(m,n,i,j,l,h)
                      !end Delta^2L_{1,1} This matrix block contains partial^2L/(partial a_{i,j} partial a_{m,n})
              end do
          end do
        end do
    end do
  end if
  if (mynode == 1) then
      do i = 0,K1
          do j = 0,K2
              do m = 0, K1
                  do n = 0,K2
                      !Delta^2L_{1,2} This matrix block contains partial^2L/(partial a_{i,j} partial b_{m,n})
                      Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(SIN(pi*m*D(:,1)*(l**(-1)))*COS(pi*n*D(:,2)*(h**(-1)))*&
                                              COS(pi*i*D(:,1)*(l**(-1)))*COS(pi*j*D(:,2)*(h**(-1))))
                      !end Delta^2L_{1,2} This matrix block contains partial^2L/(partial a_{i,j} partial b_{m,n})
                  end do
          end do
        end do
    end do
      !Send to processor 0 with label 1
    call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,1,mpi_comm_world,error_number)
  end if
  if (mynode == 0) then
      !Receive from processor 1 with label 1
     call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,1,1,mpi_comm_world,statusMPI,error_number)
     He1(1:aux,aux+1:2*aux)=Heaux
  end if
  if (mynode == 2) then
      do i = 0,K1
          do j = 0,K2
              do m = 0, K1
                  do n = 0,K2
                      !Delta^2L_{1,3} This matrix block contains partial^2L/(partial a_{i,j} partial c_{m,n})
                      Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(COS(pi*m*D(:,1)*(l**(-1)))*SIN(pi*n*D(:,2)*(h**(-1)))*&
                                              COS(pi*i*D(:,1)*(l**(-1)))*COS(pi*j*D(:,2)*(h**(-1))))
                      !end Delta^2L_{1,3} This matrix block contains partial^2L/(partial a_{i,j} partial c_{m,n})
                  end do
            end do
        end do
    end do
    !Send to processor 0 with label 2
    call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,2,mpi_comm_world,error_number)
  end if
  if (mynode==0) then
      !Receive from processor 2 with label 2
    call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,2,2,mpi_comm_world,statusMPI,error_number)
    He1(1:aux,2*aux+1:3*aux)=Heaux
  end if
  if (mynode == 3) then
      do i = 0,K1
          do j = 0,K2
              do m = 0, K1
                  do n = 0,K2
                      !Delta^2L_{1,4} This matrix block contains partial^2L/(partial a_{i,j} partial d_{m,n})
                      Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(SIN(pi*m*D(:,1)*(l**(-1)))*SIN(pi*n*D(:,2)*(h**(-1)))*&
                                              COS(pi*i*D(:,1)*(l**(-1)))*COS(pi*j*D(:,2)*(h**(-1))))
                      !end Delta^2L_{1,4} This matrix block contains partial^2L/(partial a_{i,j} partial d_{m,n})
                  end do
            end do
        end do
    end do
      !Send to processor 0 with label 3
    call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,3,mpi_comm_world,error_number)
  end if
  if (mynode==0) then
    !Receive from processor 3 with label 3
```

```
      call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,3,3,mpi_comm_world,statusMPI,error_number)
      He1(1:aux,3*aux+1:4*aux)=Heaux
   end if
   if (mynode == 4) then
     do i = 0,K1
           do j = 0,K2
               do m = 0, K1
                   do n = 0,K2
                       !Delta^2L_{2,2} This matrix block contains partial^2L/(partial b_{i,j} partial b_{m,n})
                       Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(SIN(pi*m*D(:,1)*(l**(-1)))*COS(pi*n*D(:,2)*(h**(-1)))*&
                                        SIN(pi*i*D(:,1)*(l**(-1)))*COS(pi*j*D(:,2)*(h**(-1))))
                 Hesub2(phi(i,j,K2),phi(m,n,K2))=4*(pi**4)/((l**2)*(h**2))*m*n*i*j*int_cossincossin(m,n,i,j,l,h)+&
                                        2*((pi/l)**4*m**2*i**2 + (pi/h)**4*(n**2)*(j**2) )*int_sincossincos(m,n,i,j,l,h)
                       !end Delta^2L_{2,2} This matrix block contains partial^2L/(partial b_{i,j} partial b_{m,n})
              end do
            end do
        end do
    end do
    !Send to processor 0 with label 4
    call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,4,mpi_comm_world,error_number)
      !Send to processor 0 with label 5
    call mpi_send(Hesub2(1,1),aux*aux,mpi_double_precision,0,5,mpi_comm_world,error_number)
   end if
   if (mynode==0) then
     !Receive from processor 4 with label 4
     call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,4,4,mpi_comm_world,statusMPI,error_number)
     He1(aux+1:2*aux,aux+1:2*aux)=Heaux
       !Receive from processor 4 with label 5
     call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,4,5,mpi_comm_world,statusMPI,error_number)
     He2(aux+1:2*aux,aux+1:2*aux)=Heaux
   end if
   if (mynode == 5) then
       do i = 0,K1
           do j = 0,K2
               do m = 0, K1
                   do n = 0,K2
                   !Delta^2L_{2,3} This matrix block contains partial^2L/(partial b_{i,j} partial c_{m,n})
                   Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(COS(pi*m*D(:,1)*(l**(-1)))*SIN(pi*n*D(:,2)*(h**(-1)))*&
                                        SIN(pi*i*D(:,1)*(l**(-1)))*COS(pi*j*D(:,2)*(h**(-1))))
                   !end Delta^2L_{2,3} This matrix block contains partial^2L/(partial b_{i,j} partial c_{m,n}
                   end do
              end do
          end do
     end do
       !Send to processor 0 with label 6
     call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,6,mpi_comm_world,error_number)
   end if
   if (mynode==0) then
       !Receive from processor 5 with label 6
     call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,5,6,mpi_comm_world,statusMPI,error_number)
     He1(aux+1:2*aux,2*aux+1:3*aux)=Heaux
   end if
   if (mynode == 6) then
       do i = 0,K1
           do j = 0,K2
               do m = 0, K1
                   do n = 0,K2
                   !Delta^2L_{2,4} This matrix block contains partial^2L/(partial b_{i,j} partial d_{m,n})
                   Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(SIN(pi*m*D(:,1)*(l**(-1)))*SIN(pi*n*D(:,2)*(h**(-1)))*&
                                        SIN(pi*i*D(:,1)*(l**(-1)))*COS(pi*j*D(:,2)*(h**(-1))))
                   !end Delta^2L_{2,4} This matrix block contains partial^2L/(partial b_{i,j} partial d_{m,n})
                   end do
              end do
          end do
     end do
     !Send to processor 0 with label 7
     call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,7,mpi_comm_world,error_number)
   end if
   if (mynode==0) then
       !Receive from processor 6 with label 7
       call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,6,7,mpi_comm_world,statusMPI,error_number)
     He1(aux+1:2*aux,3*aux+1:4*aux)=Heaux
   end if
   if (mynode == 7) then
       do i = 0,K1
           do j = 0,K2
               do m = 0, K1
                   do n = 0,K2
                       !Delta^2L_{3,3} This matrix block contains partial^2L/(partial c_{i,j} partial c_{m,n})
                       Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(COS(pi*m*D(:,1)*(l**(-1)))*SIN(pi*n*D(:,2)*(h**(-1)))*&
                                        COS(pi*i*D(:,1)*(l**(-1)))*SIN(pi*j*D(:,2)*(h**(-1))))
                 Hesub2(phi(i,j,K2),phi(m,n,K2))=4*(pi**4)/((l**2)*(h**2))*m*n*i*j*int_sincossincos(m,n,i,j,l,h)+&
                                        2*((pi/l)**4*m**2*i**2 + (pi/h)**4*(n**2)*(j**2) )*int_cossincossin(m,n,i,j,l,h)
```

```
                !end Delta^2L_{3,3} This matrix block contains partial^2L/(partial c_{i,j} partial c_{m,n})
                end do
            end do
        end do
    end do
    !Send to processor 0 with label 8
    call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,8,mpi_comm_world,error_number)
      !Send to processor 0 with label 9
    call mpi_send(Hesub2(1,1),aux*aux,mpi_double_precision,0,9,mpi_comm_world,error_number)
  end if
  if (mynode==0) then
      !Receive from processor 7 with label 8
    call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,7,8,mpi_comm_world,statusMPI,error_number)
    He1(2*aux+1:3*aux,2*aux+1:3*aux)=Heaux
      !Receive from processor 7 with label 9
    call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,7,9,mpi_comm_world,statusMPI,error_number)
    He2(2*aux+1:3*aux,2*aux+1:3*aux)=Heaux
  end if
  if (mynode == 8) then
      do i = 0,K1
          do j = 0,K2
              do m = 0, K1
                  do n = 0,K2
                      !Delta^2L_{3,4} This matrix block contains partial^2L/(partial c_{i,j} partial d_{m,n})
                  Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(SIN(pi*m*D(:,1)*(l**(-1)))*SIN(pi*n*D(:,2)*(h**(-1)))*&
                                                  COS(pi*i*D(:,1)*(l**(-1)))*SIN(pi*j*D(:,2)*(h**(-1))))
                  !end Delta^2L_{3,4} This matrix block contains partial^2L/(partial c_{i,j} partial d_{m,n})
                  end do
            end do
        end do
    end do
    !Send to processor 0 with label 10
    call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,10,mpi_comm_world,error_number)
  end if
  if (mynode==0) then
      !Receive from processor 8 with label 10
    call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,8,10,mpi_comm_world,statusMPI,error_number)
    He1(2*aux+1:3*aux,3*aux+1:4*aux)=Heaux
  end if
  if (mynode == 9) then
      do i = 0,K1
          do j = 0,K2
              do m = 0, K1
                  do n = 0,K2
                  !Delta^2L_{4,4} This matrix block contains partial^2L/(partial d_{i,j} partial d_{m,n})
                  Hesub1(phi(i,j,K2),phi(m,n,K2))=(2.0/NN)*SUM(SIN(pi*m*D(:,1)*(l**(-1)))*SIN(pi*n*D(:,2)*(h**(-1)))*&
                                            SIN(pi*i*D(:,1)*(l**(-1)))*SIN(pi*j*D(:,2)*(h**(-1))))
                Hesub2(phi(i,j,K2),phi(m,n,K2))=4*(pi**4)/((l**2)*(h**2))*m*n*i*j*int_cos4(m,n,i,j,l,h)+&
                                           2*((pi/l)**4*m**2*i**2 + (pi/h)**4*(n**2)*(j**2) )*int_sin4(m,n,i,j,l,h)
                  !end Delta^2L_{4,4} This matrix block contains partial^2L/(partial d_{i,j} partial d_{m,n})
                  end do
              end do
          end do
    end do
    !Send to processor 0 with label 11
    call mpi_send(Hesub1(1,1),aux*aux,mpi_double_precision,0,11,mpi_comm_world,error_number)
      !Send to processor 0 with label 12
    call mpi_send(Hesub2(1,1),aux*aux,mpi_double_precision,0,12,mpi_comm_world,error_number)
  end if
  if (mynode==0) then
      !Receive from processor 9 with label 11
    call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,9,11,mpi_comm_world,statusMPI,error_number)
    He1(3*aux+1:4*aux,3*aux+1:4*aux)=Heaux
      !Receive from processor 9 with label 12
    call mpi_recv(Heaux(1,1),aux*aux,mpi_double_precision,9,12,mpi_comm_world,statusMPI,error_number)
    He2(3*aux+1:4*aux,3*aux+1:4*aux)=Heaux
  end if

  if (mynode==0) then
      open(unit=1,file="He1.txt",ACTION="write")
      do i=1,4*(K1+1)*(K2+1)
        write(1, *) ( dble(He1(i,j)),j=1,4*(K1+1)*(K2+1))
      end do
      close(unit=1)
      open(unit=2,file="He2.txt",ACTION="write")
      do i=1,4*(K1+1)*(K2+1)
        write(2, *) ( dble(He2(i,j)),j=1,4*(K1+1)*(K2+1))
      end do
      close(unit=2)

    deallocate(He1)
      deallocate(He2)
    deallocate(Heaux)
```

```
     print *,'The Hessian was calculated'

     t2 = mpi_wtime()

     open(unit=3,file="elapsedTime.txt",ACTION="write")
     write(3, *) (t2-t1)
       close(unit=3)

     print *, 'total time ', t2-t1
   end if

   deallocate(Hesub1)
   deallocate(Hesub2)
   deallocate(D)

   call mpi_finalize(error_number)

contains
!The following function computes the row or column index of part of the matrix given i,j or m,n
!It is just one of the 16 blocks of the matrix
integer function phi(i,j,K2) result(s)
   integer, intent(in) :: i
   integer, intent(in) :: j
   integer, intent(in) :: K2
   s = i*(K2+1) + (j+1)
end function phi

end program SDFFS
```

El código anterior hace uso del módulo funcs_SDFFS el cual contiene las funciones que calculan exactamente las integrales dadas en las tablas de la sección anterior.

```
module funcs_SDFFS

   !double precision, parameter :: pi=4*atan(1.0)
   !pi = 4*atan(1.0)
contains

   double precision function int_cos4(m,n,i,j,l,h) result(y)

   implicit none

   integer, intent(in) :: m,n,i,j
   real, intent(in) :: l,h
   real :: pi

   pi=4.0*atan(1.0)

   if (m==0) then
       if (n==0) then
           if (i==0) then
               if (j==0) then
                   y=4*h*l
               else
                   y=(4*h*l*sin(j*pi))/(j*pi)
               end if
           else
               if (j==0) then
                   y=(4*h*l*sin(i*pi))/(i*pi)
               else
                   y=(4*h*l*sin(i*pi)*sin(j*pi))/(i*j*pi**2)
               end if
           end if
       else
           if (i==0) then
               if (j==0) then
                   y=(4*h*l*sin(n*pi))/(n*pi)
               else
                   if (n==j) then
                       y=2*h*l+(h*l*sin(2*j*pi))/(j*pi)
                   else
                       y=(4*h*j*l*cos(n*pi)*sin(j*pi))/((j**2-n**2)*pi)-(4*h*l*n*cos(j*pi)*sin(n*pi))/((j**2-n**2)*pi)
                   end if
               end if
           else
```

```
            if (j==0) then
                y=(4*h*l*sin(i*pi)*sin(n*pi))/(i*n*pi**2)
            else
                if (n==j) then
                    y=(2*h*l*sin(i*pi))/(i*pi)+(h*l*sin(i*pi)*sin(2*j*pi))/(i*j*pi**2)
                else
                    y=(4*h*j*l*cos(n*pi)*sin(i*pi)*sin(j*pi))/(i*(j**2-n**2)*pi**2)&
                      -(4*h*l*n*cos(j*pi)*sin(i*pi)*sin(n*pi))/(i*(j**2-n**2)*pi**2)
                end if
            end if
        end if
    end if
else
    if (n==0) then
        if (i==0) then
            if (j==0) then
                y=(4*h*l*sin(m*pi))/(m*pi)
            else
                y=(4*h*l*sin(j*pi)*sin(m*pi))/(j*m*pi**2)
            end if
        else
            if (j==0) then
                if (m==i) then
                    y=2*h*l+(h*l*sin(2*i*pi))/(i*pi)
                else
                    y=(2*h*l*sin((i-m)*pi))/((i-m)*pi)+(2*h*l*sin((i+m)*pi))/((i+m)*pi)
                end if
            else
                if (m==i) then
                    y=(2*h*l*sin(j*pi))/(j*pi)+(h*l*sin(2*i*pi)*sin(j*pi))/(i*j*pi**2)
                else
                    y=(2*h*l*sin(j*pi)*sin((i-m)*pi))/(j*(i-m)*pi**2)+(2*h*l*sin(j*pi)*sin((i+m)*pi))/(j*(i+m)*pi**2)
                end if
            end if
        end if
    else
        if (i==0) then
            if (j==0) then
                y=(4*h*l*sin(m*pi)*sin(n*pi))/(m*n*pi**2)
            else
                if (n==j) then
                    y=(2*h*l*sin(m*pi))/(m*pi) + (h*l*sin(2*j*pi)*sin(m*pi))/(j*m*pi**2)
                else
                    y=(4*h*j*l*cos(n*pi)*sin(j*pi)*sin(m*pi))/(m*(j**2-n**2)*pi**2)&
                      -(4*h*l*n*cos(j*pi)*sin(m*pi)*sin(n*pi))/(m*(j**2-n**2)*pi**2)
                end if
            end if
        else
            if (j==0) then
                if (m==i) then
                    y=(2*h*l*sin(n*pi))/(n*pi)+(h*l*sin(2*i*pi)*sin(n*pi))/(i*n*pi**2)
                else
                    y=(2*h*l*sin((i-m)*pi)*sin(n*pi))/((i-m)*n*pi**2)+(2*h*l*sin((i+m)*pi)*sin(n*pi))/((i+m)*n*pi**2)
                end if
            else
                if (m==i) then
                    if (n==j) then
                        y=h*l+(h*l*sin(2*i*pi))/(2*i*pi)+(h*l*sin(2*j*pi))/(2*j*pi)&
                          +(h*l*sin(2*i*pi)*sin(2*j*pi))/(4*i*j*pi**2)
                    else
                        y=(2*h*j*l*cos(n*pi)*sin(j*pi))/((j**2-n**2)*pi)&
                          +(h*j*l*cos(n*pi)*sin(2*i*pi)*sin(j*pi))/(i*(j**2-n**2)*pi)&
                          -(2*h*l*n*cos(j*pi)*sin(n*pi))/((j**2-n**2)*pi)&
                          -(h*l*n*cos(j*pi)*sin(2*i*pi)*sin(n*pi))/(i*(j**2-n**2)*pi)
                    end if
                else
                    if (n==j) then
                        y=(2*h*i*l*cos(m*pi)*sin(i*pi))/((i-m)*(i+m)*pi)&
                          +(h*i*l*cos(m*pi)*sin(i*pi)*sin(2*j*pi))/(j*(i-m)*(i+m)*pi**2)&
                          -(2*h*l*m*cos(i*pi)*sin(m*pi))/((i-m)*(i+m)*pi)&
                          -(h*l*m*cos(i*pi)*sin(2*j*pi)*sin(m*pi))/(j*(i-m)*(i+m)*pi**2)
                    else
                        y=(4*h*i*j*l*cos(m*pi)*cos(n*pi)*sin(i*pi)*sin(j*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)&
                        -(4*h*j*l*m*cos(i*pi)*cos(n*pi)*sin(j*pi)*sin(m*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)&
                        -(4*h*i*l*n*cos(j*pi)*cos(m*pi)*sin(i*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)&
                        +(4*h*l*m*n*cos(i*pi)*cos(j*pi)*sin(m*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)
                    end if
                end if
            end if
        end if
    end if
end if
```

```
end function int_cos4

!the prevoous function seems to be right. I tried once

double precision function int_sin4(m,n,i,j,l,h) result(y)

implicit none

integer, intent(in) :: m,n,i,j
real, intent(in) :: l,h
real :: pi

pi=4.0*atan(1.0)

if (m==0 .OR. n==0 .OR. i==0 .OR. j==0) then
    y=0;
else
    if (m==i) then
        if (n==j) then
            y=h*l-(h*l*sin(2*i*pi))/(2*i*pi)-(h*l*sin(2*j*pi))/(2*j*pi)+&
              (h*l*sin(2*i*pi)*sin(2*j*pi))/(4*i*j*pi**2)
        else
            y=(2*h*l*n*cos(n*pi)*sin(j*pi))/((j**2-n**2)*pi)-&
              (h*l*n*cos(n*pi)*sin(2*i*pi)*sin(j*pi))/(i*(j**2-n**2)*pi**2)-&
              (2*h*j*l*cos(j*pi)*sin(n*pi))/((j**2-n**2)*pi)+&
              (h*j*l*cos(j*pi)*sin(2*i*pi)*sin(n*pi))/(i*(j**2-n**2)*pi**2)
        end if
    else
        if (n==j) then
            y=(2*h*l*m*cos(m*pi)*sin(i*pi))/((i-m)*(i+m)*pi)-&
              (h*l*m*cos(m*pi)*sin(i*pi)*sin(2*j*pi))/(j*(i-m)*(i+m)*pi**2)-&
              (2*h*i*l*cos(i*pi)*sin(m*pi))/((i-m)*(i+m)*pi)+&
              (h*i*l*cos(i*pi)*sin(2*j*pi)*sin(m*pi))/(j*(i-m)*(i+m)*pi**2)
        else
            y=(4*h*l*m*n*cos(m*pi)*cos(n*pi)*sin(i*pi)*sin(j*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)-&
              (4*h*i*l*n*cos(i*pi)*cos(n*pi)*sin(j*pi)*sin(m*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)-&
              (4*h*j*l*n*cos(j*pi)*cos(m*pi)*sin(i*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)+&
              (4*h*i*j*l*cos(i*pi)*cos(j*pi)*sin(m*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)
        end if
    end if
end if

end function int_sin4

!the prevoous function seems to be right. I tried once

double precision function int_sincossincos(m,n,i,j,l,h) result(y)

implicit none

integer, intent(in) :: m,n,i,j
real, intent(in) :: l,h
real :: pi

pi=4.0*atan(1.0)

if (m==0 .OR. i==0) then
    y=0
else
    if (n==0) then
        if (j==0) then
            if (m==i) then
                y=2*h*l-(h*l*sin(2*i*pi))/(i*pi)
            else
                y=(2*h*l*sin((i-m)*pi))/((i-m)*pi)-(2*h*l*sin((i+m)*pi))/((i+m)*pi)
            end if
        else
            if (m==i) then
                y=(2*h*l*sin(j*pi))/(j*pi)-(h*l*sin(2*i*pi)*sin(j*pi))/(i*j*pi**2)
            else
                y=(2*h*l*sin(j*pi)*sin((i-m)*pi))/(j*(i-m)*pi**2)-&
                  (2*h*l*sin(j*pi)*sin((i+m)*pi))/(j*(i+m)*pi**2)
            end if
        end if
    else
        if (j==0) then
            if (m==i) then
                y=(2*h*l*sin(n*pi))/(n*pi)-(h*l*sin(2*i*pi)*sin(n*pi))/(i*n*pi**2)
            else
                y=(2*h*l*sin((i-m)*pi)*sin(n*pi))/((i-m)*n*pi**2)-&
                  (2*h*l*sin((i+m)*pi)*sin(n*pi))/((i+m)*n*pi**2)
            end if
        else
```

```fortran
            if (m == i) then
                if (n==j) then
                    y=h*l-(h*l*sin(2*i*pi))/(2*i*pi)+(h*l*sin(2*j*pi))/(2*j*pi)-&
                      (h*l*sin(2*i*pi)*sin(2*j*pi))/(4*i*j*pi**2)
                else
                    y=(2*h*j*l*cos(n*pi)*sin(j*pi))/((j**2-n**2)*pi)-&
                      (h*j*l*cos(n*pi)*sin(2*i*pi)*sin(j*pi))/(i*(j**2-n**2)*pi)-&
                      (2*h*l*n*cos(j*pi)*sin(n*pi))/((j**2-n**2)*pi)+&
                      (h*l*n*cos(j*pi)*sin(2*i*pi)*sin(n*pi))/(i*(j**2-n**2)*pi)
                end if
            else
                if (n==j) then
                    y=(2*h*l*m*cos(m*pi)*sin(i*pi))/((i-m)*(i+m)*pi)+&
                      (h*l*m*cos(m*pi)*sin(i*pi)*sin(2*j*pi))/(j*(i-m)*(i+m)*pi**2)-&
                      (2*h*i*l*cos(i*pi)*sin(m*pi))/((i-m)*(i+m)*pi)-&
                      (h*i*l*cos(i*pi)*sin(2*j*pi)*sin(m*pi))/(j*(i-m)*(i+m)*pi**2)
                else
                    y=(4*h*j*l*m*cos(m*pi)*cos(n*pi)*sin(i*pi)*sin(j*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)-&
                      (4*h*i*j*l*cos(i*pi)*cos(n*pi)*sin(j*pi)*sin(m*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)-&
                      (4*h*l*m*n*cos(j*pi)*cos(m*pi)*sin(i*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)+&
                      (4*h*i*l*n*cos(i*pi)*cos(j*pi)*sin(m*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)
                end if
            end if
        end if
    end if
end if

end function int_sincossincos

!the prevoous function seems to be correct. I tried it once

double precision function int_cossincossin(m,n,i,j,l,h) result(y)

implicit none

integer, intent(in) :: m,n,i,j
real, intent(in) :: l,h
real :: pi

pi=4.0*atan(1.0)

if (n==0 .OR. j==0) then
    y=0
else
    if (m==0) then
        if (i==0) then
            if (n==j) then
                y=2*h*l-(h*l*sin(2*j*pi))/(j*pi)
            else
                y=4*h*l*n*(cos(n*pi)*sin(j*pi))/((j**2-n**2)*pi)-&
      4*h*j*l*(cos(j*pi)*sin(n*pi))/((j**2-n**2)*pi)
            end if
        else
            if (n==j) then
                y=(2*h*i*l*cos(m*pi)*sin(i*pi))/((i-m)*(i+m)*pi)-&
                      (h*i*l*cos(m*pi)*sin(i*pi)*sin(2*j*pi))/(j*(i-m)*(i+m)*pi**2)-&
                      (2*h*l*m*cos(i*pi)*sin(m*pi))/((i-m)*(i+m)*pi)+&
                      (h*l*m*cos(i*pi)*sin(2*j*pi)*sin(m*pi))/(j*(i-m)*(i+m)*pi**2)
            else
                y=4*h*l*n*(cos(n*pi)*sin(i*pi)*sin(j*pi))/(i*(j**2-n**2)*pi**2)-&
      4*h*j*l*(cos(j*pi)*sin(i*pi)*sin(n*pi))/(i*(j**2-n**2)*pi**2)
            end if
        end if
    else
        if (i==0) then
            if (n==j) then
                y=(2*h*l*sin(m*pi))/(m*pi)-(h*l*sin(2*j*pi)*sin(m*pi))/(j*m*pi**2)
            else
                y=4*h*l*n*(cos(n*pi)*sin(j*pi)*sin(m*pi))/(m*(j**2-n**2)*pi**2)-&
      4*h*j*l*(cos(j*pi)*sin(m*pi)*sin(n*pi))/(m*(j**2-n**2)*pi**2)
            end if
        else
            if (n == j) then
                if (m==i) then
                    y=h*l+(h*l*sin(2*i*pi))/(2*i*pi)-(h*l*sin(2*j*pi))/(2*j*pi)-&
                      (h*l*sin(2*i*pi)*sin(2*j*pi))/(4*i*j*pi**2)
                else
                    y=(2*h*i*l*cos(m*pi)*sin(i*pi))/((i-m)*(i+m)*pi)-&
                      (h*i*l*cos(m*pi)*sin(i*pi)*sin(2*j*pi))/(j*(i-m)*(i+m)*pi**2)-&
                      (2*h*l*m*cos(i*pi)*sin(m*pi))/((i-m)*(i+m)*pi)+&
                      (h*l*m*cos(i*pi)*sin(2*j*pi)*sin(m*pi))/(j*(i-m)*(i+m)*pi**2)
                end if
            else
```

```
                if (m==i) then
                    y=(2*h*l*n*cos(n*pi)*sin(j*pi))/((j**2-n**2)*pi)+&
                      (h*l*n*cos(n*pi)*sin(2*i*pi)*sin(j*pi))/(i*(j**2-n**2)*pi**2)-&
                      (2*h*j*l*cos(j*pi)*sin(n*pi))/((j**2-n**2)*pi)-&
                      (h*j*l*cos(j*pi)*sin(2*i*pi)*sin(n*pi))/(i*(j**2-n**2)*pi**2)
                else
                    y=(4*h*i*l*n*cos(m*pi)*cos(n*pi)*sin(i*pi)*sin(j*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)-&
                      (4*h*l*m*n*cos(i*pi)*cos(n*pi)*sin(j*pi)*sin(m*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)-&
                      (4*h*i*j*l*cos(j*pi)*cos(m*pi)*sin(i*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)+&
                      (4*h*j*l*m*cos(i*pi)*cos(j*pi)*sin(m*pi)*sin(n*pi))/((i-m)*(i+m)*(j**2-n**2)*pi**2)
                end if
            end if
        end if
    end if
end if

end function int_cossincossin

!The previous function seems to work correctly. I tried it once

end module funcs_SDFFS
```

Finalmente el vector de coeficientes de Fourier se calcula (en la variable coef) usando el siguiente código MATLAB

```
%import vol87 skiping the first row (it contains metadata)
%and change its name to input_data
border_eps=0.1;
x_range=max(input_data(:,1))-min(input_data(:,1));
input_data(:,1)=(input_data(:,1)-min(input_data(:,1))) /x_range*(1-2*border_eps)-0.5+border_eps;

y_range=max(input_data(:,2))-min(input_data(:,2));
input_data(:,2)=(input_data(:,2)-min(input_data(:,2))) /y_range*(1-2*border_eps)-0.5+border_eps;

%plot3(input_data(:,1),input_data(:,2),input_data(:,3),'.b')

dlmwrite('glacier.txt',input_data)

%run the fortran code and import He1.txt, He2.txt, and b_vec.txt
l=1.2;
h=1.2;
K1=35;
K2=35;
for i=1:4*(K1+1)*(K2+1)
    for j=1:i
        He1(i,j)=He1(j,i);
    end
end

lambda=0.00000001;
He=He1+lambda*He2;
coef=pinv(He,0.0000001)*b_vec;
[u1,v1]=meshgrid(-l/3:0.005:l/3,-h/3:0.005:h/3);
s3=size(u1);
f_vals=zeros(s3(1),s3(2));
for i=1:s3(1)
    for j=1:s3(2)
      f_vals(i,j)=f_new(coef,u1(i,j),v1(i,j),K1,K2,l,h);
    end
end
figure(1)
surfl(u1,v1,f_vals)
colormap(bone);
shading interp
campos([3 -4.8 6741.3])
axis([-0.4,0.4,-0.4,0.4,1200,2200])
%%%%%%%%%%%%%This was done on Sep 2
c=min(input_data(:,3)):25:max(input_data(:,3));
figure(2)
contour(u1,v1,f_vals,c,'k')
%axis off
axis([min(min(u1)),max(max(u1)),min(min(v1)),max(max(v1))])
daspect([5 5 1])
figure(3)
plot(input_data(:,1),input_data(:,2),'k.','MarkerSize',3);
%axis off
axis([min(input_data(:,1)),max(input_data(:,1)),min(input_data(:,2)),max(input_data(:,2))])
colormap(gray(256));
daspect([5 5 1])
```

Para calcular el valor de la serie de Fourier en un punto el código anterior usa la función

```
function z=f_new(coef,x,y,K1,K2,l,h)

[U,V]=meshgrid(x.*[0:K1],y.*[0:K2]);
 %U=U'; V=V';
A=cos(pi*U/l).*cos(pi*V/h);
B=sin(pi*U/l).*cos(pi*V/h);
C=cos(pi*U/l).*sin(pi*V/h);
D=sin(pi*U/l).*sin(pi*V/h);
s=size(A);
value=s(1)*s(2);
z=horzcat(reshape(A,1,value),reshape(B,1,value),reshape(C,1,value),reshape(D,1,value))*reshape(coef,length(coef),1);
end
```

Bibliografía

Bjórck, A. (1996). *Numerical Methods for Least Squares Problems*. SIAM e-books. Society for Industrial y Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104). ISBN: 9781611971484. URL: https://books.google.com.mx/books?id=6YCC9U9_1UkC.

Buhmann, M.D. (2003). *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press. ISBN: 9781139435246. URL: https://books.google.com.mx/books?id=TRMf53opzlsC.

Davydov, O. *Oleg Davydov's Home Page*. URL: http://www.staff.uni-giessen.de/~gc1266/.

"Davydov, O." (2002). "On the approximation power of local least squares polynomials". En: *Proceeding Algorithms for Approximation IV*, págs. 346-353.

Davydov, O., R. Morandi y A. Sestini (2006). "Local hybrid approximation for scattered data fitting with bivariate splines". En: *Computer Aided Geometric Design* 23.9, págs. 703 -721. ISSN: 0167-8396. DOI: https://doi.org/10.1016/j.cagd.2006.04.001. URL: http://www.sciencedirect.com/science/article/pii/S0167839606000422.

Davydov, O. y L. L. Schumaker (2002). "On stable local bases for bivariate polynomial spline spaces". En: *Constructive Approximation* 18, págs. 87-116.

Davydov, O., A. Sestini y R. Morandi (2005). "Local RBF Approximation for Scattered Data Fitting with Bivariate splines". En: *Trends and Applications in Constructive Approximations* 151, págs. 91-102.

Davydov, O. y F. Zeilfelder (2004). "Scattered Data Fitting by Direct Extension of Local Polynomials to Bivariate Splines". En: *Advances in Computational Mathematics* 21.3, págs. 223-271. ISSN: 1572-9044. DOI: 10.1023/B:ACOM.0000032041.68678.fa. URL: https://doi.org/10.1023/B:ACOM.0000032041.68678.fa.

Farin, G.E. (1993). *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Computer science and scientific computing. Academic Press. ISBN: 9780122490521. URL: https://books.google.com.mx/books?id=U_4YAQAAIAAJ.

Fasshauer, G.E. (2007). *Meshfree Approximation Methods with MATLAB*. Interdisciplinary mathematical sciences. World Scientific. ISBN: 9789812706331. URL: https://books.google.com.mx/books?id=gtqBdMEqryEC.

Franke, R. (1979). "A critical comparison of some methods for interpolation of scattered data". En: *Report NPS-53-79-003*.

Grishin, D. y T. Strohmer (2003). "Fast multi-dimensional scattered data approximation with Neumann boundary conditions". En: *ArXiv Mathematics e-prints*. eprint: math/0301152.

Kimeldorf, G. y G. Wahba (1971). "Some Results on Tchebycheffian Spline Functions". En: *Journal of Mathematical Analysis and Applications* 33, págs. 82-95.

Kunis, S. y D. Potts (2007). "Stability Results for Scattered Data Interpolation by Trigonometric Polynomials". En: *ArXiv Mathematics e-prints*. eprint: math/0702019.

Potts, D., G. Steidl y M. Tasche (2001). "Fast Fourier transforms for nonequispaced data: A tutorial". En: *Modern Sampling Theory: Mathematics and Applications*, págs. 247-270.

Reincsh, C. H. (1967). "Smoothing by spline functions". En: *Numerische Mathematik* 10.3, págs. 177-183. ISSN: 0945-3245. DOI: 10.1007/BF02162161. URL: https://doi.org/10.1007/BF02162161.

Sewchuk, J. R. (1994). "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain". En: *Report*.

Steidl, G. (1998). "A note on fast Fourier transforms for nonequispaced grids". En: *Advances in Computational Mathematics* 9.3, págs. 337-352. ISSN: 1572-9044. DOI: 10.1023/A:1018901926283. URL: https://doi.org/10.1023/A:1018901926283.

Wendland, H. (2004). *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press. ISBN: 9781139456654. URL: https://books.google.com.mx/books?id=qy4cbWUmSyYC.

Zhou, T. y D. Han (2008). "A weighted least squares method for scattered data fitting". En: *Journal of Computational and Applied Mathematics* 217.1, págs. 56 -63. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2007.06.015. URL: http://www.sciencedirect.com/science/article/pii/S0377042707003159.