

DEPARTAMENT DE MATEMÀTICA APLICADA

TÉCNICAS DE INTERPOLACIÓN WENO Y SU
APLICACIÓN AL PROCESAMIENTO DE IMÁGENES.

ANA MARÍA BELDA GARCÍA

UNIVERSITAT DE VALÈNCIA
Servei de Publicacions
2010

Aquesta Tesi Doctoral va ser presentada a València el dia 1 de març de 2010 davant un tribunal format per:

- Dr. Francesc J. Ferri Rabasa
- Dr. Saturnino Maldonado Bascón
- Dr. Vicente Martínez García
- Dr. Guillaume Chiavassa
- Dr. Pep Mulet Mestre

Va ser dirigida per:

Dr. Francesc Aràndiga Llaudes

©Copyright: Servei de Publicacions
Ana María Belda García

Dipòsit legal: V-2101-2011

I.S.B.N.: 978-84-370-7864-9

Edita: Universitat de València

Servei de Publicacions

C/ Arts Gràfiques, 13 baix

46010 València

Spain

Telèfon:(0034)963864115

TESIS DOCTORAL

Técnicas de interpolación
WENO y su aplicación al
procesamiento de imágenes

Ana María Belda García

Director: Francesc Aràndiga Llaudes

Universitat de València
Valencia, 2010.

TÉCNICAS DE INTERPOLACIÓN WENO Y SU APLICACIÓN AL PROCESAMIENTO DE IMÁGENES

Memoria presentada por Ana María Belda García, Licenciada en Ciencias Matemáticas; realizada en el Departamento de Matemática Aplicada de la Universitat de València bajo la Dirección de Francesc Aràndiga Llaudes, Profesor del mencionado Departamento, con el objetivo de aspirar al Grado de Doctora en Matemáticas.

Valencia, 8 de enero de 2010

Francesc Aràndiga Llaudes
Director de la Memoria

Ana María Belda García
Aspirante al Grado de Doctora

DEPARTAMENTO DE MATEMÁTICA APLICADA
FACULTAD DE MATEMÁTICAS
UNIVERSITAT DE VALÈNCIA

Agradecimientos

En primer lugar, quiero expresar mi agradecimiento a Paco Aràndiga, sin su direcci3n, sus ànimos constantes y su ayuda decidida no hubiese podido conseguir llegar al final del largo camino que ha representado este trabajo de investigaci3n.

Tambi3n quiero expresar mi profundo agradecimiento a mis compa˜eros de doctorado y a todos los miembros del departamento de Matemàtica Aplicada por el apoyo recibido a lo largo de todos estos a˜os.

A Josep Vicent quiero agradecerle su apoyo en la implementaci3n de los algoritmos que he analizado a lo largo de la tesis.

Y por ulti3mo, un recuerdo para Tron que me ayuda a encontrarme mejor todos los dıas y siempre es un refuerzo moral importante para mı.

Valencia, 2010

Ana

Índice general

Introducción	IX
1. Multirresolución de Harten	1
1.1. Introducción	1
1.2. Multirresolución por valores puntuales	8
1.3. Multirresolución por medias en celdas	10
1.4. Interpolación lineal	14
1.5. Interpolación ENO	17
1.6. Interpolación ENO-SR	18
1.7. Compresión de datos y control del error	20
1.8. Algoritmos de multirresolución en 2D	26
1.8.1. Valores puntuales	29
1.8.2. Medias en celda	30
2. Interpolación WENO para medias en celda	35
2.1. Introducción	35
2.2. Propiedades del polinomio interpolador $p(x)$	38
2.3. Definición de los pesos	39
2.4. Pesos óptimos	41
2.5. Indicadores de suavidad	45
2.5.1. Indicador de suavidad de Liu et al.	47
2.5.2. Indicador de suavidad de Jiang y Shu	51
2.5.3. Indicador de suavidad modificado	55
3. Experimentos numéricos para medias en celda	61
3.1. Introducción	61

3.2. Señales	63
3.3. Imágenes	77
3.3.1. Comparación entre reconstrucciones	77
3.3.2. Estabilidad	100
4. Interpolación WENO para valores puntuales	109
4.1. Introducción	109
4.2. Interpolación WENO	110
4.3. Pesos óptimos	112
4.4. Indicadores de suavidad	116
4.4.1. Indicador de suavidad de Liu et al.	118
4.4.2. Indicador de suavidad de Jiang y Shu	121
5. Experimentos numéricos 1D para valores puntuales	129
5.1. Introducción	129
5.2. Experimentos numéricos	131
6. Resultados sobre WENO para valores puntuales	143
6.1. Introducción	143
6.2. Interpolación WENO	144
6.3. Diseño de pesos óptimos no lineales	148
6.4. Indicadores de suavidad	152
6.5. Evaluación del indicador de suavidad	158
7. Experimentos numéricos 2D para valores puntuales	169
7.1. Esquemas adaptativos para PDE	170
7.2. Compresión	174
8. Conclusiones y perspectivas	189
A. Algoritmos 2D error-control para valores puntuales	199
B. Algoritmos 2D error-control para medias en celda	205

Introducción

Un problema común en la teoría de aproximación es la reconstrucción de una función a partir de un conjunto de valores discretos de datos que dan información sobre la función misma. Esta información por lo general viene dada como valores puntuales o medias en celda de la función sobre un conjunto finito de puntos o celdas, respectivamente. La función entonces es aproximada por un interpolante, es decir, por otra función cuyos valores puntuales o medias en celda coinciden con los de la función original.

Este interpolante puede ser construido por interpolación lineal. La interpolación es un procedimiento lineal sobre los valores de la función en un conjunto de puntos. En este caso la exactitud de la aproximación cerca de una singularidad está limitada y depende del orden de la singularidad, de modo que si construimos el polinomio interpolador basándonos en un conjunto de puntos (*stencil*) que cruzan la singularidad obtendremos una aproximación insatisfactoria. Esto significa que el aumento del grado del polinomio producirá regiones más grandes de mala aproximación alrededor de las singularidades.

Así pues, para aumentar la exactitud la solución no es aumentar el número de puntos que se utiliza en la interpolación, sino escoger los puntos de forma que el *stencil* quede dentro de la parte suave de la función, siempre que esto sea posible. Esta es la idea que hay detrás de la técnica de interpolación ENO (Esencialmente No Oscilatoria), introducida por Harten et al. en [24] en el contexto de esquemas de alta resolución que capturan choques (*High resolution shock capturing*) para leyes de conservación.

La técnica ENO es un procedimiento no lineal, dependiente de

los datos, en el que el polinomio interpolador se construye utilizando sólo información de las regiones de suavidad de la función interpolada, siempre que sea posible. Se consideran dos algoritmos para la selección de esta información, que se estudian en profundidad en [9, 17]. Estos algoritmos proporcionan el stencil más suave para cada celda, es decir, el stencil que más se aleja de la singularidad. Así, con la técnica de interpolación ENO la región de poca exactitud queda reducida al intervalo que contiene la singularidad, siempre y cuando las singularidades estén suficientemente bien separadas.

En [18] y en [41] se presentó una forma alternativa de selección de stencil, consistente en una estrategia de sesgo, que toma como base un stencil “preferido”, que será el stencil centrado en la celda donde se realiza la interpolación, y lo utiliza para modificar el criterio de selección de stencil con un parámetro de sesgo. La idea es no alejarse de este stencil, excepto en el caso de que en el stencil alternativo la función sea “mucho” más suave, y este “mucho” viene dado por el sesgo.

La aproximación ENO se puede mejorar si se conoce la posición de la singularidad en la celda (o una aproximación suficientemente buena de ella). En este caso, la pérdida de exactitud se puede evitar en una región que contiene casi todo el intervalo donde está la singularidad. Esta es la idea básica que lleva a la técnica de Resolución Subcelda (*Subcell Resolution*, ENO-SR) de Harten [20].

Otra mejora sobre la técnica de interpolación ENO fue introducida en [33] por Liu et al. Se conoce como ENO ponderado (*weighted ENO*, WENO), y parte de un enfoque de combinaciones convexas. Esta técnica se basa en la versión para medias en celda de la interpolación ENO, y consiste en reconstruir un polinomio que interpola los valores puntuales de la solución de una ley de conservación hiperbólica a partir de las medias en celda de la solución débil.

En la interpolación WENO se asignan a cada celda todos los stencils que la contienen, y el polinomio interpolador se calcula como la combinación lineal convexa de todos los polinomios correspondientes a los stencils asignados a la celda. De este modo se utiliza la información proporcionada por todas las celdas contenidas en los stencils que se consideran en el proceso de selección

ENO, y el resultado es un mayor orden de exactitud en las regiones suaves de la función interpolada.

La clave es asignar los pesos más convenientes a la combinación convexa. Estos pesos deben ser elegidos de forma que la combinación convexa sea esencialmente una combinación convexa de los polinomios interpoladores en los stencils suaves, y que los polinomios interpoladores en los stencils que cruzan una singularidad no tengan prácticamente ninguna contribución en la combinación.

Como el peso que se le asigna a cada stencil debe variar de acuerdo con la suavidad de dicho stencil con respecto a los otros stencils, el modo en que se evalúa la suavidad de una función sobre un stencil es crucial en la definición del peso. En [33], Liu et al. definen un indicador de suavidad de la función interpolada basado en las diferencias de las medias en celda de la función en cada stencil. Jiang y Shu [27] introdujeron una nueva forma de medir la suavidad de una función, basada en la minimización de la norma L2 de las derivadas de los polinomios interpoladores, emulando la idea de minimizar la variación total de la aproximación. Esta es una definición más eficiente que la de Liu et al. Mientras que Liu et al. consiguen aumentar en uno el orden de exactitud con respecto a la reconstrucción ENO estándar, con la medida de la suavidad presentada en [27] la reconstrucción WENO alcanza un orden de exactitud óptimo en las zonas suaves de la función interpolada. Se puede encontrar una revisión de las técnicas ENO y WENO en [42].

Un enfoque diferente a WENO se presentó en [35]. En este trabajo se presenta un esquema de reconstrucción dependiente de los datos basado en mínimos cuadrados ponderados con stencil fijo. Los pesos se eligen de forma que enfatizan los datos suaves en la reconstrucción, y satisfacen las condiciones WENO establecidas por Liu et al. en [33]. Este enfoque es similar al de la técnica WENO, pero se diferencia en que los pesos se asignan a los puntos individualmente, y no a los stencils como ocurre con WENO. Eso permite una flexibilidad adicional para seleccionar los datos suaves que se considerarán para la reconstrucción.

En los últimos años se han publicado muchos trabajos relacionados con la técnica WENO. Por ejemplo, en [19] y [26] se construyen versiones de WENO de volúmenes finitos en mallas estructu-

radas y no estructuradas. Se han desarrollado esquemas basados en una nueva versión centrada de la reconstrucción WENO de valores puntuales a partir de medias en celda (ver [29], [30], [31], [32]), y también una técnica WENO basada en polinomios de Hermite (*Hermite WENO*, HWENO) para resolver sistemas de leyes de conservación hiperbólicas no lineales unidimensionales (ver [37], [38] y [39]).

En [25] los autores demuestran que las condiciones inicialmente planteadas para WENO no son suficientes, ya que el orden de la aproximación se reduce ante puntos críticos en los que la derivada de la función se anula, y proponen una modificación en la definición de los pesos, que permite obtener orden óptimo incluso en presencia de puntos críticos.

Como se ha mencionado antes, la técnica WENO se introdujo originalmente con el objetivo de interpolar valores puntuales a partir de medias en celda. En este trabajo se ha integrado la técnica WENO en el entorno de multirresolución de Harten, y se ha adaptado a los contextos de medias en celda y valores puntuales. En el primer caso, se utilizan medias en celda para reconstruir un polinomio que interpole el conjunto de medias en celda de origen, mientras que en el segundo caso se utilizan valores puntuales para recuperar el conjunto original de valores puntuales de una función.

Por otra parte, tanto en el contexto de medias en celda como en el de valores puntuales, se proponen nuevas medidas para la suavidad de una función, partiendo de los indicadores de suavidad definidos por Jiang y Shu [27]. Además, en el contexto de valores puntuales se propone una modificación en la definición de los pesos WENO, diferente de la propuesta por Henrick et al. en [25], que mejora el orden de la aproximación en presencia de singularidades. En la definición de los pesos WENO se introduce un ϵ positivo para evitar que el denominador se anule. En la literatura se suele tomar una constante $\epsilon = 10^{-5}$ o $\epsilon = 10^{-6}$. En esta tesis se propone tomar $\epsilon = h^2$, lo que permite demostrar que si la función es suave en al menos $r + 1$ puntos y tiene una discontinuidad dentro del stencil de $2r$ puntos, entonces obtenemos al menos una aproximación de orden $r + 1$, es decir, como mínimo obtenemos el mismo orden que el interpolante ENO, y en las zonas suaves de

la función el orden de la aproximación es óptimo incluso en presencia de puntos críticos en los que se anulen las dos primeras derivadas. Parte de este trabajo ha sido recogido en los artículos [4] (ya publicado) y [5] (en revisión).

Las técnicas de interpolación WENO se comparan, mediante diferentes experimentos numéricos, con las técnicas de interpolación lineal, ENO y ENO-SR, para poder concluir qué método proporciona la mayor exactitud en cada caso.

Esquema de la Tesis

Esta tesis está organizada de la siguiente forma:

Capítulo 1: Introducimos los hechos básicos de la multirresolución de Harten, en el contexto de valores puntuales y medias en celda, así como las diferentes técnicas de interpolación con las que se comparará posteriormente la técnica WENO.

Capítulo 2: Presentamos la técnica de interpolación WENO desarrollada por Liu et al., adaptada para recuperar medias en celda a partir de medias en celda, y proponemos una nueva medida de la suavidad de una función.

Capítulo 3: Mostramos los experimentos numéricos realizados en una y dos dimensiones con las distintas técnicas de interpolación consideradas en el contexto de medias en celda.

Capítulo 4: Adaptamos las ideas básicas de la técnica WENO al contexto de valores puntuales y probamos que obtenemos un orden de aproximación $2r$ cuando utilizamos stencils de $2r$ puntos y la función es suave.

Capítulo 5: Mostramos los resultados de varios experimentos numéricos con señales unidimensionales en el contexto de valores puntuales.

Capítulo 6: Proponemos tres interpoladores WENO por valores puntuales, basados en el indicador de suavidad propuesto por Jiang y Shu. Además proponemos tomar $\epsilon = h^2$, en lugar de $\epsilon = 10^{-5}$ o $\epsilon = 10^{-6}$, como se ha hecho tradicionalmente en la literatura sobre WENO. Esto permite demostrar que el interpolante WENO alcanza un orden de aproximación óptimo en regiones suaves de la función aunque haya algún punto crítico en el que $f' = f'' = 0$. También demostramos que si la función es suave en al menos $r + 1$ puntos y tiene una discontinuidad dentro del stencil de $2r$ puntos, entonces obtenemos al menos una aproximación de orden $r + 1$, es decir, como mínimo obtenemos el mismo orden que el interpolante ENO.

Capítulo 7: Incluimos resultados numéricos obtenidos con distintos métodos de interpolación, con aplicación en dos entornos diferentes en el contexto de valores puntuales: por una parte, esquemas adaptativos para EDP, y por otra, compresión de imágenes digitales en blanco y negro.

Apéndice A: Detallamos los algoritmos de codificación y decodificación en dos dimensiones en el contexto de valores puntuales con el mecanismo de control del error.

Apéndice B: Detallamos los algoritmos de codificación y decodificación en dos dimensiones en el contexto de medias en celda con el mecanismo de control del error.

1

Multirresolución de Harten

1.1

Introducción

En este capítulo resumimos el análisis de multirresolución de Harten haciendo hincapié en las partes que van a ser utilizadas en esta tesis. En [1, 9, 6, 7, 21, 8] se pueden encontrar más detalles.

El objetivo de la multirresolución es obtener una reordenación multiescala de la información contenida en un conjunto de datos discretos a una cierta resolución. Esta información es un elemento perteneciente a un espacio V^k , donde k indica el nivel de resolución. Un mayor valor de k indica una mayor resolución.

Para realizar la transición entre distintos niveles de resolución

se utilizan dos operadores llamados *decimación* y *predicción*. El operador decimación proporciona información discreta a un nivel de resolución $k - 1$ a partir de la información contenida en el nivel k :

$$D_k^{k-1} : V^k \longrightarrow V^{k-1}$$

y debe ser lineal y biyectivo. El operador predicción actúa en sentido opuesto, dando una aproximación a la información discreta en el nivel k a partir de la información contenida en el nivel $k - 1$:

$$P_{k-1}^k : V^{k-1} \longrightarrow V^k$$

Además, al operador predicción no se le exige que sea lineal.

Los datos discretos se obtienen a partir de la *discretización* de una función f , para lo cual existen distintos operadores. Dependiendo del operador discretización utilizado, la secuencia de datos f^k que resulta es diferente. El objetivo del enfoque propuesto por Harten es la construcción de esquemas de multirresolución adaptados a cada proceso de discretización. Esto se consigue definiendo un operador *reconstrucción* apropiado. Estos operadores, discretización y reconstrucción, son los elementos a partir de los cuales se construyen los operadores decimación y predicción del esquema de multirresolución.

Formalmente, sea \mathcal{F} un espacio de funciones:

$$\mathcal{F} \subset \{f \mid f : \Omega \subset \mathbb{R}^m \longrightarrow \mathbb{R}\}.$$

El operador discretización asigna a cada elemento de este espacio, $f \in \mathcal{F}$, una secuencia f^k de datos discretos perteneciente al espacio V^k . De este modo se define el operador discretización:

$$\mathcal{D}_k : \mathcal{F} \longrightarrow V^k = \mathcal{D}_k(\mathcal{F})$$

que ha de ser lineal y biyectivo.

La reconstrucción opera en sentido inverso, tomando una secuencia de datos discretos para reconstruir, a partir de la información proporcionada por dichos datos, la función de la que provienen:

$$\mathcal{R}_k : V^k \longrightarrow \mathcal{F}.$$

La principal novedad introducida por Harten consiste en que a este operador reconstrucción no se le exige que sea lineal.

Por motivos de consistencia, se requiere que los operadores discretización y reconstrucción satisfagan la siguiente condición:

$$\mathcal{D}_k \mathcal{R}_k f^k = f^k, \quad \forall f^k \in V^k \quad (1.1)$$

o expresado de otro modo:

$$\mathcal{D}_k \mathcal{R}_k = I_{V^k}, \quad (1.2)$$

es decir, si tomamos la información reconstruida a partir de unos datos discretos con una cierta resolución y la discretizamos a ese mismo nivel de resolución, la información discreta obtenida coincide con la original.

En la Figura 1.1 se muestran las relaciones existentes entre los operadores discretización y reconstrucción, y los operadores decimación y predicción.

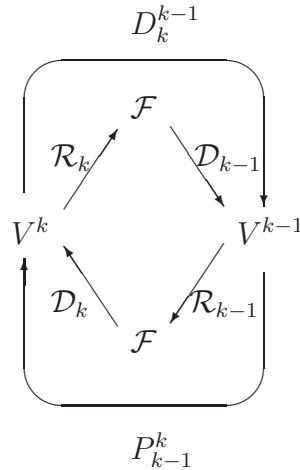


Figura 1.1: Definición de operadores.

Según estas relaciones, el operador decimación se define del siguiente modo:

$$D_k^{k-1} := \mathcal{D}_{k-1} \mathcal{R}_k. \quad (1.3)$$

Aunque aparentemente el operador decimación depende de la elección del operador reconstrucción en realidad no es así si (y sólo

si) la sucesión de operadores discretización $\{\mathcal{D}_k\}$ es *ennidada*, es decir, si se tiene:

$$\mathcal{D}_k f = 0 \implies \mathcal{D}_{k-1} f = 0, \quad \forall f \in \mathcal{F}. \quad (1.4)$$

La propiedad de ennidamiento significa que la información contenida en los datos a un cierto nivel de resolución k no será nunca mayor que la información contenida en un nivel de resolución superior.

De forma similar, el operador predicción se construye según la expresión:

$$P_{k-1}^k := \mathcal{D}_k \mathcal{R}_{k-1}. \quad (1.5)$$

A partir de estas definiciones se obtiene una relación de consistencia equivalente a (1.1) para los operadores decimación y predicción:

$$D_k^{k-1} P_{k-1}^k = \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k \mathcal{R}_{k-1} = \mathcal{D}_{k-1} \mathcal{R}_{k-1} = I_{V^{k-1}}. \quad (1.6)$$

Para demostrar la igualdad (1.6) basta con demostrar que:

$$\mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k = \mathcal{D}_{k-1}. \quad (1.7)$$

Esta relación es una consecuencia de la propiedad de ennidamiento (1.4) de la secuencia de operadores discretización, ya que, tomando una función f cualquiera:

$$\mathcal{D}_k (\mathcal{R}_k \mathcal{D}_k f) = (\mathcal{D}_k \mathcal{R}_k) \mathcal{D}_k f = \mathcal{D}_k f,$$

donde en la última igualdad utilizamos la condición de consistencia (1.2). Como el operador discretización es lineal, tenemos que:

$$\mathcal{D}_k (\mathcal{R}_k \mathcal{D}_k f - f) = 0,$$

y por la propiedad de ennidamiento, esto implica que:

$$\mathcal{D}_{k-1} (\mathcal{R}_k \mathcal{D}_k f - f) = 0$$

es decir:

$$\mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k f = \mathcal{D}_{k-1} f \implies \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k = \mathcal{D}_{k-1},$$

con lo que queda demostrado (1.7).

La relación (1.6) significa que cuando utilizamos estos operadores no inventamos información, es decir, si decimos la información obtenida a partir de la predicción realizada sobre una información con resolución dada por V^{k-1} , obtenemos exactamente la misma información de partida, sin introducir ningún elemento nuevo.

Consideremos ahora f^k , la información discreta en el nivel de resolución k . Si aplicamos el operador decimación sobre f^k obtenemos f^{k-1} , es decir, la información contenida en el nivel de resolución $k-1$:

$$f^{k-1} = D_k^{k-1} f^k.$$

En este caso, podemos interpretar que $P_{k-1}^k f^{k-1}$ es una aproximación a f^k , con un error:

$$e^k := (I_{V^k} - P_{k-1}^k D_k^{k-1}) f^k =: Q_k f^k \in V^k. \quad (1.8)$$

De esta forma podemos representar la información contenida en f^k equivalentemente mediante $f^{k-1} \in V^{k-1}$ y $e^k \in V^k$:

$$f^k \equiv \{f^{k-1}, e^k\}$$

ya que a partir de f^k se pueden obtener f^{k-1} y e^k , en la forma ya descrita, y recíprocamente, conociendo f^{k-1} y e^k se puede calcular f^k mediante la expresión $P_{k-1}^k f^{k-1} + e^k = f^k$.

El problema es que haciendo esto incluimos información redundante, ya que, si suponemos que V^k es un espacio de dimensión finita (como lo es habitualmente en la práctica), $\dim V^k = N_k$, tenemos por un lado f^k , que contiene la información codificada en N_k elementos, mientras que $\{f^{k-1}, e^k\}$ contiene la misma información codificada en $N_{k-1} + N_k$ elementos. Esta información redundante puede ser eliminada, como consecuencia del siguiente resultado:

$$\begin{aligned} D_k^{k-1} e^k &= D_k^{k-1} (I_{V^k} - P_{k-1}^k D_k^{k-1}) v^k \\ &= D_k^{k-1} v^k - D_k^{k-1} P_{k-1}^k D_k^{k-1} v^k \\ &= D_k^{k-1} v^k - D_k^{k-1} v^k \\ &= 0 \end{aligned}$$

es decir, $e^k \in \mathcal{N}(D_k^{k-1}) = \{f^k \in V^k : D_k^{k-1} f^k = 0\}$, cuya dimensión es $\dim \mathcal{N}(D_k^{k-1}) = \dim V^k - \dim V^{k-1} = N_k - N_{k-1}$.

Sea $\{\mu_i^k\}$ el conjunto de elementos que generan el espacio $\mathcal{N}(D_k^{k-1})$. Podemos expresar el error e^k como:

$$e^k = \sum d_i^k \mu_i^k$$

Si definimos un operador G_k que asocie a cada elemento $e^k \in \mathcal{N}(D_k^{k-1})$ su correspondiente conjunto de coeficientes $\{d_i^k\}$, y un operador E_k , inverso de G_k , que a un conjunto de coeficientes $\{d_i^k\}$ asocie la correspondiente combinación $\sum d_i^k \mu_i^k$ podemos establecer la siguiente equivalencia:

$$f^k \equiv \{f^{k-1}, d^k\}$$

mediante las relaciones:

$$\begin{aligned} f^{k-1} &= D_k^{k-1} f^k \\ d^k &= G_k (I - P_{k-1}^k D_k^{k-1}) f^k, \end{aligned}$$

para obtener $\{f^{k-1}, d^k\}$ a partir de f^k , y:

$$P_{k-1}^k f^{k-1} + E_k d^k = f^k,$$

en sentido inverso.

En este caso la equivalencia de información lo es también en cuanto a número de elementos utilizados para codificar dicha información, ya que en $\{f^{k-1}, d^k\}$ tenemos $N_{k-1} + (N_k - N_{k-1}) = N_k$ elementos, los mismos que hay en f^k . Decimos entonces que los coeficientes $\{d_i^k\}$ contienen la información no redundante del error de predicción.

Iterando este procedimiento en cada nivel de resolución, se consigue la descomposición multiescala que se muestra en la Figura 1.2, y que permite establecer la siguiente equivalencia:

$$f^L \equiv \{f^0, d^L, \dots, d^1\}$$

Por tanto, y para resumir, los algoritmos para las transformaciones directa e inversa de la multirresolución son los siguientes:

$$\begin{aligned} \text{(Directa)} \quad f^L &\rightarrow M f^L = \{f^0, d^1, \dots, d^L\} \begin{cases} \text{Do } k = L, \dots, 1 \\ f^{k-1} = D_k^{k-1} f^k \\ d^k = G_k (f^k - P_{k-1}^k f^{k-1}) \end{cases} \\ \text{(Inversa)} \quad M f^L &\rightarrow M^{-1} M f^L \begin{cases} \text{Do } k = 1, \dots, L \\ f^k = P_{k-1}^k f^{k-1} + E_k d^k \end{cases} \end{aligned}$$

$$\begin{array}{ccccccc}
 f^L & \longrightarrow & f^{L-1} & \longrightarrow & f^{L-2} & \longrightarrow & \dots \\
 & \searrow & & \searrow & & \searrow & \\
 & & d^L & & d^{L-1} & & \dots
 \end{array}$$

Figura 1.2: Descomposición multiescala.

Por otra parte, las expresiones (1.3) y (1.5) permiten expresar el error de predicción (1.8) en función de los operadores discretización y reconstrucción. Considerando $f^k = \mathcal{D}_k f$ (donde $\{\mathcal{D}_k\}$ es una secuencia ennidada):

$$e^k = (I - P_{k-1}^k D_k^{k-1}) f^k = \mathcal{D}_k (I - \mathcal{R}_{k-1} \mathcal{D}_{k-1}) f.$$

Comprobemos esta afirmación:

$$\begin{aligned}
 e^k &= (I - P_{k-1}^k D_k^{k-1}) f^k \\
 &= (I - \mathcal{D}_k \mathcal{R}_{k-1} \mathcal{D}_{k-1} \mathcal{R}_k) \mathcal{D}_k f \\
 &= \mathcal{D}_k f - \mathcal{D}_k \mathcal{R}_{k-1} \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k f \\
 &= \mathcal{D}_k (I - \mathcal{R}_{k-1} \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k) f \\
 &= \mathcal{D}_k (I - \mathcal{R}_{k-1} \mathcal{D}_{k-1}) f
 \end{aligned}$$

donde en la última igualdad hemos utilizado el resultado (1.7).

De este modo, el problema de encontrar una buena predicción para la multirresolución se puede expresar también como un problema de teoría de aproximación:

Conociendo $\mathcal{D}_{k-1} f$, $f \in \mathcal{F}$, encontrar una “buena aproximación” a $\mathcal{D}_k f$.

El paso fundamental en la construcción de un esquema de multirresolución es la definición de un operador reconstrucción apropiado para la discretización que se está considerando. Habitualmente se utilizan dos tipos de discretización: la discretización por valores puntuales y la discretización por medias en celda.

1.2

Multirresolución por valores puntuales

Consideramos el espacio de las funciones acotadas definidas en el intervalo unidad cerrado $\mathcal{F} = \mathcal{B}[0, 1]$, y sea X^L una partición uniforme de dicho intervalo:

$$X^L = \{x_i^L\}_{i=0}^{J_L}, \quad x_0^L = 0, \quad x_i^L = i \cdot h_L, \quad h_L = \frac{1}{J_L}, \quad J_L = 2^L J_0, \quad (1.9)$$

donde J_0 es un número entero dado.

Definimos las particiones correspondientes a niveles de resolución más bajos del siguiente modo:

$$X^k = \{x_i^k\}_{i=0}^{J_k}, \quad k = L - 1, \dots, 0 \quad (1.10)$$

donde

$$x_i^{k-1} = x_{2i}^k, \quad i = 0, \dots, J_{k-1} := \frac{J_k}{2}.$$

Consideramos el operador discretización $f^k = \mathcal{D}_k f$ definido como:

$$f_i^k = (\mathcal{D}_k f)_i = f(x_i^k), \quad f^k = \{f_i^k\}_{i=0}^{J_k}.$$

Esto es lo que se conoce como *discretización por valores puntuales*. En este caso, el espacio V^k es de dimensión finita, igual a $\dim V^k = N_k = J_k + 1$.

Una condición necesaria y suficiente para que la sucesión de operadores discretización sea ennidada es que $X^{k-1} \subset X^k$, para todo k . Esta condición se cumple para la secuencia de particiones definida previamente.

Esta discretización nos lleva a la siguiente definición del operador decimación:

$$f_i^{k-1} = (D_k^{k-1} f^k)_i = f_{2i}^k, \quad i = 0, \dots, J_{k-1}$$

y en esta definición no interviene en ningún momento el operador reconstrucción.

El operador reconstrucción será un operador \mathcal{R}_k que cumple:

$$\mathcal{R}_k : V^k \longrightarrow \mathcal{B}([0, 1]), \quad \mathcal{D}_k \mathcal{R}_k f^k = f^k$$

es decir, $(\mathcal{R}_k f^k)(x_i^k) = f_i^k = f(x_i^k)$. Por tanto, $\mathcal{R}_k f^k(x)$ tiene que ser una función acotada que interpole a $\{f_i^k\}$ en los nodos $\{x_i^k\}$. Cualquier técnica de interpolación es válida como operador reconstrucción. Denotamos:

$$\mathcal{I}(x; f^k) := (\mathcal{R}_k f^k)(x). \quad (1.11)$$

De aquí deducimos que la predicción es:

$$(P_{k-1}^k f^{k-1})_i = (\mathcal{D}_k \mathcal{I}(\cdot; f^{k-1}))_i = \mathcal{I}(x_i^k; f^{k-1}). \quad (1.12)$$

El error de predicción será por tanto $e_i^k = f_i^k - \mathcal{I}(x_i^k; f^{k-1})$, que evidentemente es igual a 0 para los nodos x_{2i}^k . Entonces los coeficientes $d^k = \{d_j^k\}$ se definen como el error de interpolación en los puntos x_{2j-1}^k , es decir, $d_j^k = e_{2j-1}^k$.

De este modo, los algoritmos para las transformaciones directa e inversa para valores puntuales vienen dados por las expresiones:

$$f^L \rightarrow M f^L = \{f^0, d^1, \dots, d^L\} \text{ Directa}$$

$$\begin{cases} \text{Do } k = L, \dots, 1 \\ f_i^{k-1} = f_{2i}^k, & 0 \leq i \leq J_{k-1}, \\ d_i^k = f_{2i-1}^k - \mathcal{I}(x_{2i-1}^k; f^{k-1}), & 1 \leq i \leq J_{k-1}, \end{cases} \quad (1.13)$$

$$M^{-1} M f^L \rightarrow f^L \text{ Inversa}$$

$$\begin{cases} \text{Do } k = 1, \dots, L \\ f_{2i}^k = f_i^{k-1}, & 0 \leq i \leq J_{k-1}, \\ f_{2i-1}^k = \mathcal{I}(x_{2i-1}^k; f^{k-1}) + d_i^k, & 1 \leq i \leq J_{k-1}. \end{cases} \quad (1.14)$$

En estos algoritmos se puede utilizar cualquier técnica de interpolación. Si utilizamos interpolantes que sean polinomios a trozos de grado r , como los coeficientes d_j^k corresponden al error de interpolación en los nodos con índice impar en el nivel de resolución k , tendremos que:

$$d_j^k = O(h_{k-1}^{r+1})$$

en las regiones donde la función sea suave. Ante una singularidad aislada, a la que denotamos $x_d \in (x_{j-1}^{k-1}, x_j^{k-1})$, en la derivada p -ésima de la función, es decir, tal que $[f^{(p)}]_{x_d} = O(1)$ ($p \leq r$):

$$d_j^k = O(h_{k-1}^p).$$

El valor de los coeficientes d_j^k varía en función del orden de la singularidad, por lo que podemos pensar en el análisis de multirresolución como un análisis de la regularidad de una función. Los mayores coeficientes van asociados a las singularidades de la función, lo que significa que no podemos predecir la información contenida en esas regiones.

Más importante aún es el hecho de que si utilizamos una técnica de interpolación independiente de los datos, el conjunto de los intervalos afectados por una singularidad no se reduce únicamente a aquel intervalo en el que se localiza dicha singularidad, sino a todos los intervalos cuyo stencil contenga el intervalo donde se encuentra la singularidad, por lo que habrá una mayor cantidad de coeficientes con valores significativos, y la capacidad de compresión del esquema de multirresolución se verá reducida. Esto es lo que ocurre cuando se utiliza interpolación lineal centrada (ver Sección 1.4).

Por otra parte, al utilizar técnicas de interpolación que dependen de los datos, es decir, no lineales, se minimiza la zona afectada por cada singularidad, y la capacidad de compresión del esquema de multirresolución mejora. Esto ocurre cuando se utilizan las técnicas no lineales ENO (ver Sección 1.5) y ENO-SR (ver Sección 1.6).

1.3

Multirresolución por medias en celdas

Sea $\mathcal{F} = L^1[0, 1]$ el espacio de funciones integrables en el intervalo unidad. Consideramos de nuevo la partición del intervalo unidad (1.9) y sus refinamientos (1.10), y sea $c_i^k = (x_{i-1}^k, x_i^k)$, $i = 1, \dots, J_k$. La *discretización por medias en celda* a un nivel de resolución k viene definida por:

$$\mathcal{D}_k : \mathcal{F} \longrightarrow V^k, \quad \bar{f}_i^k = (\mathcal{D}_k f)_i = \frac{1}{|c_i^k|} \int_{c_i^k} f(x) dx = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} f(x) dx. \quad (1.15)$$

Es importante observar que cuando $\mathcal{F} \subset L^1[0, 1]$, la discretización por valores puntuales podría no tener sentido, pero la discretización por medias en celda (1.15) siempre lo tiene.

La sucesión de operadores discretización es ennidada, como consecuencia de la aditividad de la integral:

$$(\mathcal{D}_{k-1}f)_i = \frac{1}{h_{k-1}} [h_k(\mathcal{D}_k f)_{2i-1} + h_k(\mathcal{D}_k f)_{2i}]$$

y esto además sirve como definición del operador decimación:

$$D_k^{k-1} \mathcal{D}_k f := \mathcal{D}_{k-1} f$$

es decir, el valor asignado a una celda es igual a la media de los valores correspondientes a las celdas del nivel de resolución superior incluidas en ella. Por tanto, el operador decimación viene dado por la expresión:

$$(D_k^{k-1} \bar{f}^k)_i = \frac{1}{2} (\bar{f}_{2i-1}^k + \bar{f}_{2i}^k).$$

El error de predicción verifica que $(D_k^{k-1} e^k)_i = 0$, o lo que es lo mismo:

$$\frac{1}{2} (e_{2i-1}^k + e_{2i}^k) = 0$$

lo cual nos lleva a definir los coeficientes d_j^k de la siguiente manera:

$$d_j^k = e_{2j-1}^k$$

y podemos recuperar el error a partir de los coeficientes d_j^k :

$$\begin{cases} e_{2j-1}^k &= d_j^k, \\ e_{2j}^k &= -d_j^k, \end{cases} \quad 1 \leq j \leq J_{k-1} = J_k/2.$$

El operador reconstrucción \mathcal{R}_k para esta discretización deberá cumplir la condición de consistencia (1.1), es decir:

$$(\mathcal{D}_k \mathcal{R}_k f^k)_i = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} \mathcal{R}_k(x; f^k) dx = f_i^k, \quad \forall f^k \in V^k \quad (1.16)$$

El procedimiento de reconstrucción vía función primitiva nos permite enlazar con la teoría de interpolación, lo cual resulta muy útil. Brevemente, este procedimiento se describe a continuación.

Se define la sucesión $\{F_i^k\}$ a partir de los nodos correspondientes al nivel de resolución k :

$$F_i^k = h_k \sum_{j=1}^i \bar{f}_j^k = \int_0^{x_i^k} f(y) dy.$$

Si consideramos F , una primitiva de f , es decir $F(x) = \int_0^x f(y) dy$, que será una función continua en $[0, 1]$, entonces la sucesión $\{F_i^k\}$ no es más que una discretización por valores puntuales de la función F en el nivel de resolución k . Estos valores están relacionados unívocamente con las medias en celda de f según la relación:

$$\bar{f}_i^k = \frac{F_i^k - F_{i-1}^k}{h_k} \quad (1.17)$$

teniendo en cuenta que $F_0^k = 0, \forall k$.

Sea $\mathcal{I}(x; F^k)$ un operador reconstrucción de la primitiva, utilizando interpolación. Se define el operador reconstrucción \mathcal{R}_k para la discretización por medias en celda de la siguiente forma:

$$\mathcal{R}_k(x; \bar{f}^k) := \frac{d}{dx} \mathcal{I}(x; F^k).$$

Este operador verifica la condición de consistencia (1.1):

$$(\mathcal{D}_k(\mathcal{R}_k(x; \bar{f}^k)))_j = \frac{1}{h_k} \int_{x_{j-1}^k}^{x_j^k} \frac{d}{dx} \mathcal{I}(x; F^k) = \frac{F_j^k - F_{j-1}^k}{h_k} = \bar{f}_j^k$$

ya que $(\mathcal{R}_k \bar{f}^k)(x) \in L^1[0, 1]$.

De este modo, los algoritmos para la multirresolución directa e inversa para medias en celdas quedan así:

$$\begin{aligned} \bar{f}^L &\rightarrow M \bar{f}^L = \{f^0, d^1, \dots, d^L\} \text{ Directa} \\ &\begin{cases} \text{Do } k = L, \dots, 1 \\ \bar{f}_i^{k-1} = \frac{1}{2}(\bar{f}_{2i-1}^k + \bar{f}_{2i}^k), & 1 \leq i \leq J_{k-1} \\ d_i^k = \bar{f}_{2i-1}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i-1}, & 1 \leq i \leq J_{k-1}, \end{cases} \quad (1.18) \end{aligned}$$

$$\begin{aligned} M \bar{f}^L &\rightarrow M^{-1} M \bar{f}^L \text{ Inversa} \\ &\begin{cases} \text{Do } k = 1, \dots, L \\ \bar{f}_{2i-1}^k = (P_{k-1}^k \bar{f}^{k-1})_{2i-1} + d_i^k, & 1 \leq i \leq J_{k-1}, \\ \bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k, & 1 \leq i \leq J_{k-1}. \end{cases} \quad (1.19) \end{aligned}$$

donde

$$\begin{aligned}
(P_{k-1}^k \bar{f}^{k-1})_{2i-1} &= (\mathcal{D}_k \mathcal{R}_{k-1} \bar{f}^{k-1})_{2i-1} = \frac{1}{h_k} \int_{x_{2i-2}^k}^{x_{2i-1}^k} \frac{d}{dx} \mathcal{I}(x; F^{k-1}) dx \\
&= \frac{1}{h_k} (\mathcal{I}(x_{2i-1}^k; F^{k-1}) - \mathcal{I}(x_{2i-2}^k; F^{k-1})) \\
&= \frac{1}{h_k} (\mathcal{I}(x_{2i-1}^k; F^{k-1}) - F_{i-1}^{k-1}). \tag{1.20}
\end{aligned}$$

Los coeficientes $d^k(f)$ también están relacionados con los coeficientes $d^k(F)$, ya que:

$$\begin{aligned}
d_j^k(f) &= e_{2j-1}^k = \bar{f}_{2j-1}^k - (P_{k-1}^k D_k^{k-1} \bar{f}^k)_{2j-1} = \bar{f}_{2j-1}^k - (\mathcal{D}_k \mathcal{R}_{k-1} \bar{f}^{k-1})_{2j-1} \\
&= \frac{1}{h_k} (F_{2j-1}^k - F_{2j-2}^k) - \frac{1}{h_k} \int_{x_{2j-2}^k}^{x_{2j-1}^k} \frac{d}{dx} \mathcal{I}(x; F^{k-1}) \\
&= \frac{1}{h_k} [(F_{2j-1}^k - F_{2j-2}^k) - (\mathcal{I}(x_{2j-1}^k; F^{k-1}) - \mathcal{I}(x_{2j-2}^k; F^{k-1}))] \\
&= \frac{1}{h_k} (F_{2j-1}^k - \mathcal{I}(x_{2j-1}^k; F^{k-1})) = \frac{d_j^k(F)}{h_k}. \tag{1.21}
\end{aligned}$$

Notar que si $\mathcal{I}(x; F^k)$ es un polinomio interpolador a trozos de grado $r + 1$, es decir, si:

$$\mathcal{I}(x; F^k) = q_i(x; F^k), \quad x \in [x_{i-1}^k, x_i^k], \quad \text{grad } q_i = r + 1$$

entonces la reconstrucción $\mathcal{R}_k(x; \bar{f}^k)$ es un polinomio a trozos de grado r .

Por otra parte, como el stencil está formado por $s(r) = r + 1$ celdas, el correspondiente stencil para valores puntuales de la función primitiva, formado por los extremos de las celdas, contendrá $s(r) + 1 = r + 2$ puntos.

El proceso de reconstrucción por función primitiva implica que la reconstrucción \mathcal{R}_k es exacta para polinomios de grado menor o igual a r . Además, como la función primitiva es un grado más suave que f , y los polinomios que forman $\mathcal{I}(x; F^k)$ son de grado $r + 1$, por la relación (1.21) se tiene que:

$$d^k(f) = O(h_{k-1}^{r+1}),$$

en las regiones suaves, mientras que si el stencil cruza una discontinuidad de salto en la derivada $f^{(p)}(p \leq r)$, entonces:

$$d^k(f) = O([f^{(p)}]) h_{k-1}^p.$$

Por tanto, vemos que el orden de la singularidad determina el valor de todos los coeficientes en cuyo cálculo se vea involucrado un stencil que cruce dicha singularidad. Como pasaba en el caso de valores puntuales, utilizar técnicas no lineales reduce el error de aproximación en los coeficientes, con lo cual la capacidad de compresión mejora.

Una vez descritos los operadores discretización que se utilizan más a menudo, el siguiente paso consiste en analizar las distintas posibilidades existentes para construir un operador reconstrucción adecuado. Para ello, en las siguientes secciones se describen brevemente algunas técnicas de interpolación de uso común.

El problema a resolver consiste en, a partir de una información discreta dada sobre una función, recuperar la función original. El objetivo es encontrar una función sencilla (generalmente polinomios o funciones trigonométricas) que aproxime a la función original, utilizando la información disponible. Para ello se pueden utilizar técnicas lineales, independientes de los datos, que resultan muy sencillas de aplicar, o también técnicas no lineales, que al ser de tipo adaptativo, proporcionan una mejor aproximación.

1.4

Interpolación lineal

Un procedimiento particularmente sencillo es la interpolación de Lagrange. Dado un stencil \mathcal{S} formado por $r+1$ puntos, x_0, \dots, x_r , existe un único polinomio $q(x)$ de grado r tal que $q(x_i) = f(x_i)$, $\forall i = 0, \dots, r$. Este polinomio aproxima a la función $f(x)$ con un error que viene dado por la expresión:

$$f(x) = q(x) + f[x_0, \dots, x_r, x] \prod_{i=0}^r (x - x_i) \quad (1.22)$$

donde, si $f(x)$ es suficientemente suave (es decir, es al menos $r + 1$ veces diferenciable), se tiene que:

$$f[x_0, \dots, x_r, x] = \frac{f^{(r+1)}(\xi_x)}{(r+1)!}, \quad (1.23)$$

donde ξ_x es un punto que pertenece a la envoltura convexa de los nodos $\{x_0, \dots, x_r\}$.

Consideremos, por simplicidad, el intervalo $[0, 1]$, y una partición formada por puntos igualmente espaciados $X = \{x_i\}_{i=0}^J$, $x_i = ih$, tal que $Jh = 1$. Sea $\{f_i\}_{i=0}^J$ el conjunto formado por los valores puntuales de $f(x)$ en los puntos de la partición X , es decir, $f_i = f(x_i)$. Sobre cada intervalo $I_i = [x_{i-1}, x_i]$, $i = 1, \dots, J$ construimos un polinomio $q_i(x)$, a partir de un stencil de puntos \mathcal{S}_i centrado en el intervalo I_i . La unión de los distintos polinomios $q_i(x)$ proporciona el interpolante global $\mathcal{I}(x)$:

$$\mathcal{I}(x) = q_i(x), \quad x \in [x_{i-1}, x_i], \quad i = 1, \dots, J.$$

En general, si $\mathcal{I}(x)$ es un polinomio interpolador a trozos formado por polinomios de grado r y $f(x)$ es una función suficientemente suave, se obtiene la siguiente relación a partir de (1.22) y (1.23):

$$\mathcal{I}(x) = f(x) + O(h^{r+1}), \quad \forall x \in [0, 1], \quad (1.24)$$

en la que el término $O(h^{r+1})$ está relacionado con la derivada $(r+1)$ -ésima de $f(x)$, lo que significa que la interpolación permite recuperar de forma exacta los polinomios de grado menor o igual a r . Esto se expresa de otro modo diciendo que el *conjunto de exactitud* de la regla de interpolación contiene todos los polinomios de grado menor o igual a r , o dicho de otra forma, que la interpolación es de orden $r + 1$.

Cuando la función que se interpola presenta alguna singularidad, la estimación del error (1.24) ya no es válida, ya que ante una discontinuidad de salto en $f(x)$, localizada en el subintervalo I_j , se tiene que:

$$f[x_{j-1}, x_j] = \frac{f(x_j) - f(x_{j-1})}{x_j - x_{j-1}} = \frac{O([f])}{h},$$

donde $[f]$ representa el salto de $f(x)$ en I_j . Cualquier diferencia dividida basada en un conjunto de $s+1$ ($s > 1$) puntos que contenga

x_{j-1}, x_j cumplirá:

$$f[x_l, \dots, x_{l+s}] = \frac{O([f])}{h^s}.$$

Esto significa que cualquier polinomio $q_l(x)$ que cruce la discontinuidad, es decir, cuyo stencil S_l contenga los extremos del subintervalo I_l , producirá un error:

$$f(x) = q_l(x) + O([f]), \quad x \in I_l. \quad (1.25)$$

Si la discontinuidad no está en $f(x)$, sino en alguna de sus derivadas, $f^{(p)}(x)$, con $p \geq 1$, también se produce una pérdida de exactitud, ya que en ese caso:

$$f[x_l, \dots, x_{l+s}] = \begin{cases} \frac{O([f^{(p)}])}{h^{s-p}} & \text{si } s > p \\ O(\|f^{(s)}\|) & \text{en otro caso} \end{cases}$$

donde $\|f^{(s)}\|$ representa la norma supremo de $f^{(s)}$ en la envoltura convexa del stencil utilizado para calcular la diferencia dividida, y $[f^{(p)}] = f^{(p)}(x_d^+) - f^{(p)}(x_d^-)$.

De este modo, la fórmula del error para polinomios $q_l(x)$ que crucen una singularidad de orden p será la siguiente:

$$f(x) = q_l(x) + O([f^{(p)}])h^p, \quad \text{si } p \leq r. \quad (1.26)$$

Cuando se utiliza interpolación lineal centrada, cada stencil se elige de forma independiente de los datos, es decir, no se tiene en cuenta la suavidad de la función que se interpola. Esto significa que una singularidad aislada en un intervalo I_j producirá una pérdida de exactitud en varios subintervalos de la partición, ya que los stencils correspondientes a los subintervalos adyacentes a I_j contendrán también la singularidad, con lo que el error de interpolación vendrá dado por la expresión (1.25) o por (1.26).

Más concretamente, si la función f presenta una singularidad aislada en el intervalo $I_j = [x_{j-1}, x_j]$, y se utilizan para la interpolación polinomios a trozos de grado r , habrá r stencils que contengan los extremos de I_j (r se supone impar para que la interpolación sea centrada):

$$S_{j+k} = \{x_{j+k-r+1}, \dots, x_{j+k+1}\}, \quad k = 0, \dots, r-1$$

Como consecuencia de esto, aumentar el orden de la interpolación para conseguir una mejor aproximación a la función original hace que el número de intervalos en los que el error es grande sea mayor. Para evitar esto, Harten et al. [24] desarrollaron una técnica no lineal, en la que la selección del stencil es un proceso que depende de los datos. De este modo, el polinomio interpolador se construye utilizando sólo información de las regiones suaves de la función, siempre que esto sea posible. Esta técnica se conoce con el nombre de ENO (esencialmente no oscilatoria), y permite reducir el conjunto de intervalos afectados por una singularidad a únicamente aquel que la contiene.

1.5

Interpolación ENO

Evidentemente, el punto clave en la interpolación ENO es el proceso de selección del stencil. El objetivo es elegir, para cada subintervalo $I_i = [x_{i-1}, x_i]$ en el que la función f sea suave, un stencil $\tilde{S}_i = \{x_{s_i-1}, \dots, x_{s_i+r-1}\}$ que no cruce ninguna singularidad, siendo $r + 1$ el orden de la interpolación. Existen dos estrategias para realizar esta selección: la estrategia jerárquica y la no jerárquica.

El algoritmo jerárquico de selección del stencil consiste en, partiendo de los extremos del intervalo I_i , ir añadiendo puntos a derecha o izquierda, comparando las diferencias divididas correspondientes a los conjuntos formados por los extremos del intervalo más los puntos añadidos, y escogiendo la de menor valor absoluto. El algoritmo para este procedimiento es el siguiente:

```

for  $i = 1 : J$ 
   $s_0 = i$ 
  for  $l = 0 : r - 2$ 
    if  $|f[x_{s_l-2}, \dots, x_{s_l+l}]| < |f[x_{s_l-1}, \dots, x_{s_l+l+1}]|$  then  $s_{l+1} = s_l - 1$ 
  end
   $s_i = s_r - 1$ 
end

```

La estrategia no jerárquica considera las diferencias divididas correspondientes a todos los stencils posibles, y calcula el mínimo de entre los valores absolutos de dichas diferencias. El algoritmo es el siguiente:

```

for  $i = 1 : J$ 
   $|f[x_{s_i}, \dots, x_{s_i+r-1}]| = \min\{|f[x_{l-1}, \dots, x_{l+r-1}]|, i - r + 1 \leq l \leq i\}$ 
end

```

No hay diferencia entre ambos algoritmos cuando $r = 2$, pero los stencils obtenidos pueden variar cuando $r > 2$. En cualquier caso, se observa que el stencil siempre contiene los extremos del subintervalo en el que se realiza la interpolación, evitando siempre los intervalos que contienen singularidades. Un análisis exhaustivo de estos procedimientos se puede encontrar en [9].

Mediante esta técnica de interpolación, el efecto de una singularidad aislada de f , $x_d \in I_j = [x_{j-1}, x_j]$, se manifiesta sólo en el intervalo I_j . De este modo, el error de interpolación será:

$$f(x) = q_i(x) + O(h^{r+1}), \quad x \in I_i, \forall i \neq j$$

mientras que debido a la singularidad:

$$f(x) = q_j(x) + O([f^{(d)}])h^d, \quad x \in I_j$$

1.6

Interpolación ENO-SR

La aproximación obtenida mediante la técnica ENO se puede mejorar si conocemos la localización de la singularidad. Esta es la idea en la que se basa la técnica de interpolación ENO-SR (Subcell Resolution), introducida por Harten en [20]. Utilizando la información discreta proporcionada por las medias en celda de la función f , se puede localizar aproximadamente una discontinuidad. Una discontinuidad en f corresponde a una esquina en su primitiva, y de este modo, se utilizan los interpolantes de la función primitiva para calcular la aproximación a la función f .

Consideremos una función f que presenta una esquina, $x_d \in I_j = [x_{j-1}, x_j]$. Tomamos los polinomios interpoladores de grado r correspondientes a los intervalos I_{j-1} e I_{j+1} , construidos utilizando la técnica ENO, a los que denotamos $q_{j-1}(x)$ y $q_{j+1}(x)$, y que recuperan la función con una exactitud $O(h^{r+1})$.

$$\begin{aligned} f(x) &= q_{j-1}(x) + O(h^{r+1}) & x \in [x_{j-2}, x_{j-1}] \\ f(x) &= q_{j+1}(x) + O(h^{r+1}) & x \in [x_j, x_{j+1}] \end{aligned}$$

Se define la función:

$$G_j(x) := q_{j+1}(x) - q_{j-1}(x).$$

Si h es suficientemente pequeño (de forma que en el intervalo I_j no haya ninguna otra esquina), se puede demostrar que la función $G_j(x)$ tiene una raíz, $G_j(\xi) = 0$, que cumple:

$$|\xi - x_d| = O(h^{r+1}).$$

Una vez localizado ξ , el interpolante se construye utilizando los polinomios laterales en el subintervalo que contiene la singularidad:

$$\mathcal{I}^{SR}(x) = \begin{cases} q_l(x), & x \in [x_{l-1}, x_l], l \neq i, \\ q_{j-1}(x), & x \in [x_{j-1}, \xi], \\ q_{j+1}(x), & x \in [\xi, x_j]. \end{cases}$$

En la Figura 1.3 se muestran los polinomios que intervienen en esta definición.

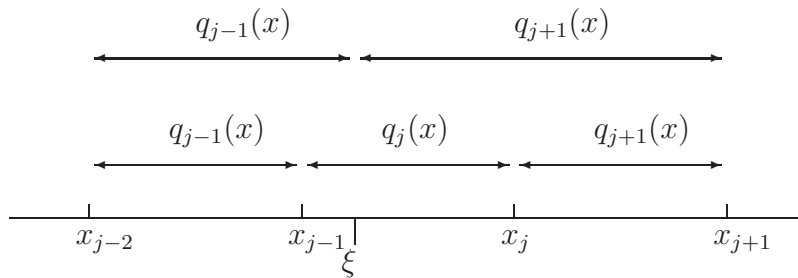


Figura 1.3: Polinomios interpoladores y sus modificaciones en SR.

De este modo se consigue que la zona afectada por la singularidad x_d sea sólo una banda de anchura $O(h^{r+1})$ alrededor de x_d , y

el error de interpolación es:

$$f(x) = \mathcal{I}(x) + O(h^{r+1}),$$

excepto en dicha banda (ver más detalles en [9, 8]).

Partiendo de distintos puntos de vista, se han propuesto otras muchas técnicas de interpolación que intentan mejorar los resultados proporcionados por la técnica ENO. Una de estas técnicas es la llamada WENO (*Weighted ENO*, ENO ponderado), que se analiza en esta tesis.

1.7

Compresión de datos y control del error

Las representaciones multirresolutivas conducen naturalmente a algoritmos de compresión. La forma más sencilla de obtener un mecanismo de compresión es, dada f^L , calcular la transformada $Mf^L = \{f^0, d^1, \dots, d^L\}$. Una vez tenemos esta descomposición igualamos a cero todos los coeficientes de escala (detalles) que son menores que un cierto umbral. Denotemos

$$(\hat{d}^k)_j = \mathbf{tr}(d_j^k; \epsilon_k) = \begin{cases} 0, & \text{si } |d_j^k| \leq \epsilon_k \\ d_j^k, & \text{en otro caso} \end{cases} \quad (1.27)$$

y nos referiremos a esta operación como *truncamiento*. Este tipo de compresión de datos se utiliza para reducir la *dimensionalidad* de los datos.

Una operación diferente, que se utiliza normalmente para reducir la representación digital de los datos, es la *cuantización*, que se puede escribir como

$$(\hat{d}^k)_j = \mathbf{qu}(d_j^k; \epsilon_k) = 2\epsilon_k \cdot \text{round} \left[\frac{d_j^k}{2\epsilon_k} \right], \quad (1.28)$$

donde $\text{round}[\cdot]$ denota el número entero obtenido mediante redondeo. Por ejemplo, si $|d_j^k| \leq 256$ y $\epsilon_k = 4$ entonces podemos representar d_j^k mediante un entero que no es mayor que 32 y no

cometemos un error mayor que 4. Observemos que si $|d_j^k| < \epsilon_k \Rightarrow \mathbf{qu}(d_j^k; \epsilon_k) = 0$ y que en cualquier caso se tiene $|d_j^k - \hat{d}_j^k| \leq \epsilon_k$.

Esquemáticamente este proceso sería:

$$f^L \longrightarrow f_d = M f^L = (f^0, d^1, \dots, d^L) \longrightarrow \hat{f}_d = Q_{\epsilon} M f^L = (\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L).$$

Si aplicamos la transformación de multirresolución inversa a la representación comprimida, obtenemos $\hat{f}^L = M^{-1}\{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$, una aproximación a la señal original f^L que esperamos que sea parecida a esta, y para que esto suceda, la estabilidad del esquema de multirresolución con respecto a perturbaciones es esencial. El estudio del efecto de utilizar \hat{d}_j^k en vez de d_j^k en los datos a los que se aplica M^{-1} es equivalente a estudiar el resultado que obtenemos al aplicar la transformación de multirresolución inversa cuando perturbamos los coeficientes de escala.

Dada una sucesión discreta f^L y un nivel de tolerancia ϵ , nuestro objetivo es obtener una transformación comprimida:

$$\{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\} \quad (1.29)$$

tal que si $\hat{f}^L = M^{-1}\{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$, entonces tenemos

$$\|f^L - \hat{f}^L\| \leq C\epsilon \quad (1.30)$$

en una cierta norma.

Como observó Harten [21], una forma de conseguir esto es modificar la transformación directa de forma que se tenga controlado el error acumulado y truncar o cuantizar mientras se calcula la descomposición multirresolución. Es decir, el proceso debería ser de la forma:

$$f^L \longrightarrow \hat{f}_d = M_{\epsilon}^M f^L = (\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L). \quad (1.31)$$

Un procedimiento modificado para calcular la transformada directa en el contexto interpolatorio sería (ver [9]):

Algoritmo 1.1. *Transformada directa modificada para valores puntuales.* $f^L \longrightarrow M_{\epsilon}^M f^L = \{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$

for $k = L, \dots, 1$

```

for  $j = 1, \dots, J_{k-1}$ 
     $f_j^{k-1} = f_{2j}^k$ 
end
end
Set  $\hat{f}^0 = \mathbf{proc}(f^0; \epsilon_0)$ 
for  $k = 1, \dots, L$ 
     $\hat{f}_0^k = f_0^L$ 
    for  $j = 1, \dots, J_{k-1}$ 
         $f^P = (P_{k-1}^k \hat{f}^{k-1})_{2j-1}$ 
         $\tilde{d}_j^k = (f_{2j-1}^k - f^P)$ 
         $\hat{d}_j^k = \mathbf{proc}(\tilde{d}_j^k; \epsilon_k)$ 
         $\hat{f}_{2j-1}^k = f^P + \hat{d}_j^k$ 
         $\hat{f}_{2j}^k = \hat{f}_j^{k-1}$ 
    end
end
end
 $M_{\bar{\epsilon}}^M f^L = \{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$ 

```

donde **proc** puede ser el operador truncamiento (1.27) o el de cuantización (1.28).

Si aplicamos este algoritmo a unos valores f^L se puede probar el siguiente resultado:

Proposición 1.1. *Dada una sucesión discreta f^L y una tolerancia ϵ , si los parámetros de truncamiento ϵ_k en el algoritmo modificado directo (Algoritmo 1.1) son elegidos de manera que $\epsilon_k = \epsilon$, entonces la sucesión $\hat{f}^L = M^{-1}\{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$ satisface*

$$\|f^L - \hat{f}^L\|_p \leq \epsilon \quad (1.32)$$

para $p = \infty, 1$ y 2 .

Demostración. Observemos que

$$f_{2j-1}^k - \hat{f}_{2j-1}^k = \tilde{d}_j^k - \hat{d}_j^k, \quad (1.33)$$

$$f_{2j}^k - \hat{f}_{2j}^k = f_j^{k-1} - \hat{f}_j^{k-1}. \quad (1.34)$$

Entonces

$$\|f^k - \hat{f}^k\|_\infty \leq \max\{\|\tilde{d}^k - \hat{d}^k\|_\infty, \|f^{k-1} - \hat{f}^{k-1}\|_\infty\}$$

y

$$\|f^L - \hat{f}^L\|_\infty \leq \max\{\epsilon_L, \epsilon_{L-1}, \dots, \epsilon_0\} = \epsilon.$$

También

$$\begin{aligned} \|f^k - \hat{f}^k\|_1 &= \frac{1}{J_k} \sum_{j=1}^{J_k} |f_j^k - \hat{f}_j^k| = \frac{1}{J_k} \sum_{j=1}^{J_{k-1}} \left(|f_{2j-1}^k - \hat{f}_{2j-1}^k| + |f_{2j}^k - \hat{f}_{2j}^k| \right) \\ &= \frac{1}{J_k} \sum_{j=1}^{J_{k-1}} \left(|f_j^{k-1} - \hat{f}_j^{k-1}| + |\tilde{d}_j^k - \hat{d}_j^k| \right) \\ &= \frac{1}{2} \left(\|f^{k-1} - \hat{f}^{k-1}\|_1 + \|\tilde{d}^k - \hat{d}^k\|_1 \right) \end{aligned}$$

y,

$$\|f^L - \hat{f}^L\|_1 = \frac{1}{2^L} \|f^0 - \hat{f}^0\|_1 + \sum_{k=1}^L \frac{1}{2^{L-k+1}} \|\tilde{d}^k - \hat{d}^k\|_1.$$

De aquí, si $\epsilon_k = \epsilon$,

$$\|f^L - \hat{f}^L\|_1 \leq \epsilon.$$

Además,

$$\begin{aligned} \|f^k - \hat{f}^k\|_2^2 &= \frac{1}{J_k} \sum_{j=1}^{J_k} |f_j^k - \hat{f}_j^k|^2 = \frac{1}{J_k} \sum_{j=1}^{J_{k-1}} \left(|f_j^{k-1} - \hat{f}_j^{k-1}|^2 + |\tilde{d}_j^k - \hat{d}_j^k|^2 \right) \\ &= \frac{1}{2} \left(\|f^{k-1} - \hat{f}^{k-1}\|_2^2 + \|\tilde{d}^k - \hat{d}^k\|_2^2 \right) \end{aligned}$$

entonces

$$\|f^L - \hat{f}^L\|_2^2 = \frac{1}{2^L} \|f^0 - \hat{f}^0\|_2^2 + \sum_{k=1}^L \frac{1}{2^{L-k+1}} \|\tilde{d}^k - \hat{d}^k\|_2^2$$

y, si $\epsilon_k = \epsilon$

$$\|f^L - \hat{f}^L\|_2^2 \leq \epsilon^2.$$

□

Un procedimiento modificado en el contexto de medias en celda sería:

Algoritmo 1.2. Transformada directa modificada para medias en celda. $\bar{f}^L \rightarrow M_\epsilon^M \bar{f}^L = \{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$

```

for  $k = L, \dots, 1$ 
  for  $j = 1, \dots, J_{k-1}$ 
     $\bar{f}_j^{k-1} = \frac{1}{2}(\bar{f}_{2j-1}^k + \bar{f}_{2j}^k)$ 
  end
end
Set  $\hat{f}^0 = \mathbf{proc}(\bar{f}^0; \epsilon_0)$ 
for  $k = 1, \dots, L$ 
  for  $j = 1, \dots, J_{k-1}$ 
     $f^P = (P_{k-1}^k \hat{f}^{k-1})_{2j-1}$ 
     $\tilde{d}_j^k = [\bar{f}_{2j-1}^k - f^P] - [\bar{f}_j^{k-1} - \hat{f}_j^{k-1}]$ 
     $\hat{d}_j^k = \mathbf{proc}(\tilde{d}_j^k; \epsilon_k)$ 
     $\hat{f}_{2j-1}^k = f^P + \hat{d}_j^k$ 
     $\hat{f}_{2j}^k = 2\hat{f}_j^{k-1} - \hat{f}_{2j-1}^k$ 
  end
end
 $M_\epsilon^M \bar{f}^L = \{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$ 

```

donde **proc** puede ser el operador truncamiento (1.27) o el de cuantización (1.28).

Proposición 1.2. Dada una sucesión discreta \bar{f}^L y una tolerancia ϵ , si los parámetros de truncamiento ϵ_k del algoritmo de codificación transformado (Algoritmo 1.2) son elegidos de forma que $\epsilon_k = \epsilon/2^{L-k}$, entonces la sucesión $\hat{f}^L = M^{-1}\{\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L\}$ cumple

$$\|\bar{f}^L - \hat{f}^L\|_p \leq C\epsilon \quad (1.35)$$

siendo $C = 2$ para $p = \infty, 1$ y $C = (\frac{4}{3})^{1/2}$ para $p = 2$.

Demostración. Observemos que

$$\bar{f}_{2j-1}^k - \hat{f}_{2j-1}^k = (\bar{f}_j^{k-1} - \hat{f}_j^{k-1}) + (\tilde{d}_j^k - \hat{d}_j^k) \quad (1.36)$$

$$\bar{f}_{2j}^k - \hat{f}_{2j}^k = (\bar{f}_j^{k-1} - \hat{f}_j^{k-1}) - (\tilde{d}_j^k - \hat{d}_j^k). \quad (1.37)$$

Entonces

$$\|\bar{f}^k - \hat{f}^k\|_\infty \leq \|\tilde{d}^k - \hat{d}^k\|_\infty + \|\bar{f}^{k-1} - \hat{f}^{k-1}\|_\infty$$

y

$$\|\bar{f}^L - \hat{f}^L\|_\infty \leq \|\bar{f}^0 - \hat{f}^0\|_\infty + \sum_{k=1}^L \|\tilde{d}^k - \hat{d}^k\|_\infty \leq \sum_{k=0}^L \epsilon_k$$

También

$$\frac{1}{J_k} \sum_{j=1}^{J_k} |\bar{f}_j^k - \hat{f}_j^k| \leq \frac{1}{J_{k-1}} \sum_{j=1}^{J_{k-1}} |\bar{f}_j^{k-1} - \hat{f}_j^{k-1}| + \frac{1}{J_{k-1}} \sum_{j=1}^{J_{k-1}} |\tilde{d}_j^k - \hat{d}_j^k|$$

entonces

$$\|\bar{f}^k - \hat{f}^k\|_1 \leq \|\bar{f}^{k-1} - \hat{f}^{k-1}\|_1 + \|\tilde{d}^k - \hat{d}^k\|_1$$

y

$$\|\bar{f}^L - \hat{f}^L\|_1 \leq \|\bar{f}^0 - \hat{f}^0\|_1 + \sum_{k=1}^L \|\tilde{d}^k - \hat{d}^k\|_1 \leq \sum_{k=0}^L \epsilon_k.$$

Entonces, si $\epsilon_k = \epsilon/2^{L-k}$, tendremos

$$\|\bar{f}^L - \hat{f}^L\|_{\infty,1} \leq \epsilon \sum_{k=0}^L \frac{1}{2^{L-k}} = \epsilon \frac{1 - \frac{1}{2}^{L+1}}{1 - \frac{1}{2}} < 2\epsilon$$

Finalmente,

$$\frac{1}{J_k} \sum_{j=1}^{J_k} |\bar{f}_j^k - \hat{f}_j^k|^2 = \frac{1}{J_{k-1}} \sum_{j=1}^{J_{k-1}} |\bar{f}_j^{k-1} - \hat{f}_j^{k-1}|^2 + \frac{1}{J_{k-1}} \sum_{j=1}^{J_{k-1}} |\tilde{d}_j^k - \hat{d}_j^k|^2$$

entonces

$$\|\bar{f}^k - \hat{f}^k\|_2^2 = \|\bar{f}^{k-1} - \hat{f}^{k-1}\|_2^2 + \|\tilde{d}^k - \hat{d}^k\|_2^2$$

y

$$\|\bar{f}^L - \hat{f}^L\|_2^2 = \|\bar{f}^0 - \hat{f}^0\|_2^2 + \sum_{k=1}^L \|\tilde{d}^k - \hat{d}^k\|_2^2 \leq \sum_{k=0}^L \epsilon_k^2.$$

De aquí, si $\epsilon_k = \epsilon/2^{L-k}$ tenemos

$$\|\bar{f}^L - \hat{f}^L\|_2^2 \leq \epsilon^2 \sum_{k=0}^L \frac{1}{4^{L-k}} = \epsilon^2 \frac{1 - \frac{1}{4}^{L+1}}{1 - \frac{1}{4}} < \frac{4}{3}\epsilon^2$$

□

Estos algoritmos pueden ser utilizados para cualquier predicción P_{k-1}^k . Si el operador de reconstrucción \mathcal{R}_k es no lineal (dependiente de datos) estos algoritmos garantizan (1.30) y de ahí la estabilidad del procedimiento de compresión de datos.

Si el operador de reconstrucción es un funcional lineal (P_{k-1}^k es un proceso de predicción lineal), la técnica de control de error que hemos descrito nos permite especificar el nivel deseado de exactitud en la señal sometida a descompresión.

La capacidad de controlar la calidad de los datos descifrados en vez del nivel de compresión es conveniente para usos donde el control de calidad tiene importancia extrema y donde nos gustaría ser tan económicos como fuera posible en lo que concierne al almacenaje y la velocidad de cálculo (nótese que *no podemos* especificar la tasa de compresión al mismo tiempo).

1.8

Algoritmos de multirresolución en 2D

La forma más sencilla de obtener algoritmos en 2D es vía producto tensorial. Esto es, si tenemos una señal 2D, f^L , le aplicamos la transformada directa unidimensional ((1.13) cuando estamos en el contexto de valores puntuales y (1.18) si estamos en el contexto de medias en celda) a las filas y después a las columnas obteniendo $\{f^{L-1}, d^L(1), d^L(2), d^L(3)\}$. Seguidamente se vuelven a aplicar a f^{L-1} los algoritmos unidimensionales a las filas y a las columnas obteniendo $\{f^{L-2}, d^{L-1}(1), d^{L-1}(2), d^{L-1}(3)\}$. Esquemáticamente este

proceso se puede describir de la siguiente forma:

$$\begin{aligned}
 f^L &\longrightarrow \{f^{L-1}, d^L(1), d^L(2), d^L(3)\} \\
 &\longrightarrow \{f^{L-2}, d^{L-1}(1), d^{L-1}(2), d^{L-1}(3), d^L(1), d^L(2), d^L(3)\} \\
 &\quad \vdots \\
 &\longrightarrow \{f^0, d^1(1), d^1(2), d^1(3), \dots, d^L(1), d^L(2), d^L(3)\} = Mf^L
 \end{aligned}$$

En la Figura 1.4 vemos cómo a partir de unos datos originales vamos obteniendo la matriz transformada.

Al igual que en 1D, una vez tenemos la matriz transformada truncamos o cuantizamos los valores ((1.27) o (1.28)) obteniendo

$$\{\hat{f}^0, \hat{d}^1(1), \hat{d}^1(2), \hat{d}^1(3), \dots, \hat{d}^L(1), \hat{d}^L(2), \hat{d}^L(3)\} = Q_{\varepsilon} M f^L$$

que es lo que habría que almacenar.

Cuando aplicamos la transformada inversa obtenemos $\hat{f}^L = M^{-1}Q_{\varepsilon}Mf^L$ que nos gustaría que satisficiera (1.30).

Si la transformación que utilizamos es lineal se puede probar que (1.30) es cierto, aunque puede ser difícil encontrar la cota C . Si las transformaciones son no lineales no está garantizado (1.30).

En lo que sigue presentamos una modificación de la extensión bidimensional vía producto tensorial de los algoritmos unidimensionales que aparecen en la Sección 1.7 (ver también [21, 9]). Dada una tolerancia ϵ , el resultado que obtenemos con la transformación directa modificada que aquí presentamos es una representación comprimida (1.29) que satisface (1.30). Esto nos permite, a priori, fijar la exactitud de la señal descomprimida. La transformación directa modificada se diseña teniendo en cuenta la transformada inversa que ha de ser utilizada.

Para lo que sigue necesitamos las definiciones siguientes:

$$\|f^k\|_{\infty} = \sup_{i,j} |f_{i,j}^k|; \quad \|f^k\|_1 = \frac{1}{N_k^2} \sum_{i,j} |f_{i,j}^k|; \quad \|f^k\|_2^2 = \frac{1}{N_k^2} \sum_{i,j} |f_{i,j}^k|^2$$

donde $f^k = \{f_{i,j}^k\}$.

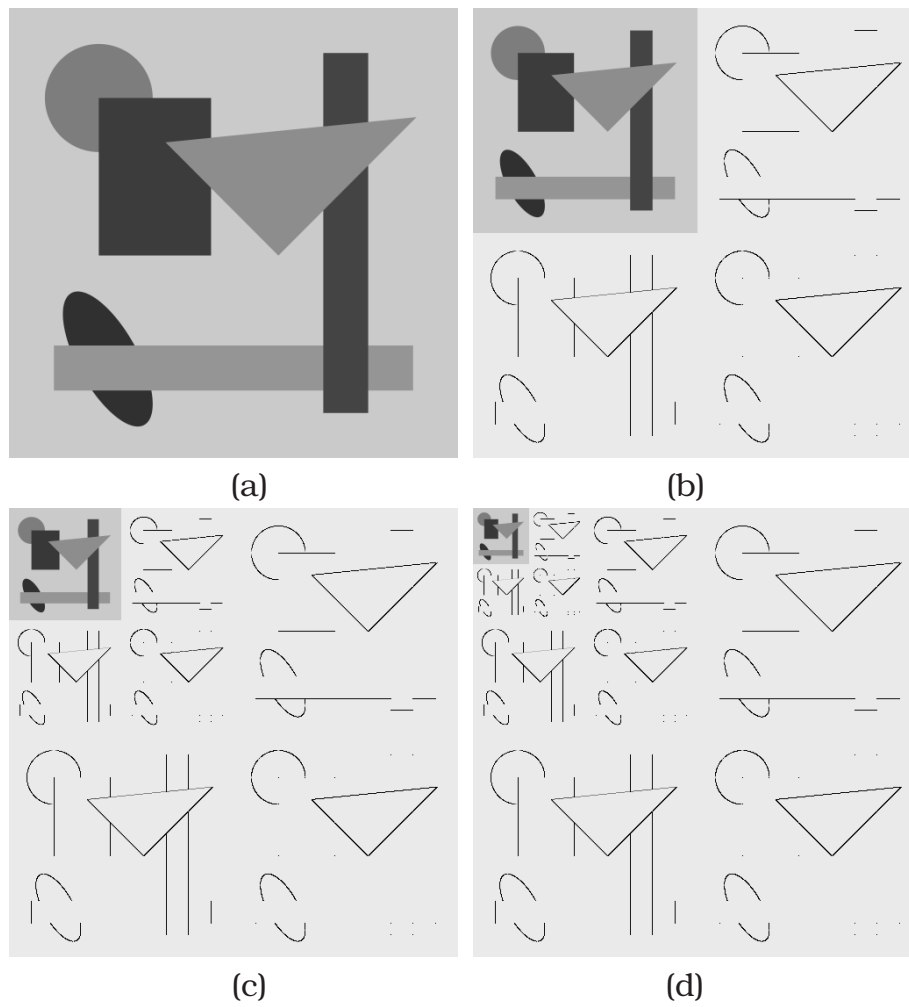


Figura 1.4: (a) Imagen original. (b) Primer paso de la transformada, (c) segundo y (d) tercero

1.8.1

Valores puntuales

Proposición 1.3. *Dados unos valores discretos f^L , con la transformación directa modificada en 2D para valores puntuales (Algoritmo A.1) obtenemos una representación multirresolutiva*

$$Mf^L = \{\hat{f}^0, \hat{d}^1(1), \hat{d}^1(2), \hat{d}^1(3), \dots, \hat{d}^L(1), \hat{d}^L(2), \hat{d}^L(3)\}$$

tal que si aplicamos la transformada inversa (Algoritmo A.2) obtenemos \hat{f}^L que cumple:

$$\|f^L - \hat{f}^L\|_\infty \leq \max_k (|\tilde{d}^k - \hat{d}^k|_\infty, \|f^0 - \hat{f}^0\|_\infty) \quad (1.38)$$

$$\|f^L - \hat{f}^L\|_1 = \frac{1}{4^L} \|f^0 - \hat{f}^0\|_1 + \sum_{k=1}^L \frac{1}{4^{L-k+1}} |\tilde{d}^k - \hat{d}^k|_1 \quad (1.39)$$

$$\|f^L - \hat{f}^L\|_2^2 = \frac{1}{4^L} \|f^0 - \hat{f}^0\|_2^2 + \sum_{k=1}^L \frac{1}{4^{L-k+1}} |\tilde{d}^k - \hat{d}^k|_2^2 \quad (1.40)$$

donde

$$|\tilde{d}^k - \hat{d}^k|_\infty = \max(|\tilde{d}^k(1) - \hat{d}^k(1)|_\infty, |\tilde{d}^k(2) - \hat{d}^k(2)|_\infty, |\tilde{d}^k(3) - \hat{d}^k(3)|_\infty)$$

$$|\tilde{d}^k - \hat{d}^k|_1 = |\tilde{d}^k(1) - \hat{d}^k(1)|_1 + |\tilde{d}^k(2) - \hat{d}^k(2)|_1 + |\tilde{d}^k(3) - \hat{d}^k(3)|_1$$

$$|\tilde{d}^k - \hat{d}^k|_2^2 = |\tilde{d}^k(1) - \hat{d}^k(1)|_2^2 + |\tilde{d}^k(2) - \hat{d}^k(2)|_2^2 + |\tilde{d}^k(3) - \hat{d}^k(3)|_2^2$$

Demostración. Ver la Proposición A.1. □

Es trivial probar el siguiente corolario

Corolario 1.1. Consideremos el esquema multirresolutivo de control del error descrito en la Proposición 1.3, y una estrategia de procesamiento de los coeficientes de escala tal que

$$\|\tilde{d}^k(l) - \hat{d}^k(l)\|_p \leq \epsilon_k \quad \forall l = 1, 2, 3, \quad p = \infty, 1, \text{ ó } 2 \quad (1.41)$$

$$\|f^0 - \hat{f}^0\|_p \leq \epsilon_0 \quad (1.42)$$

entonces

$$\|f^L - \hat{f}^L\|_\infty \leq \max_k(\epsilon_k, \epsilon_0)$$

$$\|f^L - \hat{f}^L\|_1 \leq \frac{\epsilon_0}{4^L} + 3 \sum_{k=1}^L \frac{\epsilon_k}{4^{L-k+1}}$$

$$\|f^L - \hat{f}^L\|_2^2 \leq \frac{\epsilon_0}{4^L} + 3 \sum_{k=1}^L \frac{\epsilon_k^2}{4^{L-k+1}}$$

En particular, si asumimos que $\|f^0 - \hat{f}^0\|_p = 0$ y consideramos

$$\epsilon_k = \epsilon \quad k = 0, \dots, L. \quad (1.43)$$

tenemos

$$\|f^L - \hat{f}^L\|_p \leq \epsilon, \quad p = 1, 2, \infty \quad (1.44)$$

Nota 1.1. Las estrategias de truncamiento dadas por (1.27) y (1.28) satisfacen (1.41) y (1.42).

1.8.2

Medias en celda

En el contexto de medias en celda, tenemos:

Proposición 1.4. *Dados un conjunto de valores discretos \bar{f}^L , con la transformada inversa modificada en 2D en el contexto de medias en celda (Algoritmo B.1) obtenemos una representación multirresolutiva*

$$M\bar{f}^L = \{\hat{f}^0, \hat{d}^1(1), \hat{d}^1(2), \hat{d}^1(3), \dots, \hat{d}^L(1), \hat{d}^L(2), \hat{d}^L(3)\}$$

tal que si aplicamos la transformada inversa (Algoritmo B.2) obtenemos \hat{f}^L que cumple:

$$\|\bar{f}^L - \hat{f}^L\|_\infty \leq \|\bar{f}^0 - \hat{f}^0\|_\infty + \sum_{k=1}^L \|\tilde{d}^k(2) - \hat{d}^k(2)\|_\infty \quad (1.45)$$

$$\begin{aligned} &+ \sum_{k=1}^L \max(\|\tilde{d}^k(1) - \hat{d}^k(1)\|_\infty, \|\tilde{d}^k(3) - \hat{d}^k(3)\|_\infty) \\ \|\bar{f}^L - \hat{f}^L\|_1 &\leq \|\bar{f}^0 - \hat{f}^0\|_1 \quad (1.46) \end{aligned}$$

$$\begin{aligned} &+ \frac{1}{2} \sum_{k=1}^L \max(\|\tilde{d}^k(2) - \hat{d}^k(2)\|_1, \|\tilde{d}^k(1) - \hat{d}^k(1)\|_1) \\ &+ \frac{1}{2} \sum_{k=1}^L \max(\|\tilde{d}^k(2) - \hat{d}^k(2)\|_1, \|\tilde{d}^k(3) - \hat{d}^k(3)\|_1) \\ \|\bar{f}^L - \hat{f}^L\|_2^2 &= \|\bar{f}^0 - \hat{f}^0\|_2^2 + \sum_{k=1}^L \|\tilde{d}^k(2) - \hat{d}^k(2)\|_2^2 \quad (1.47) \\ &+ \frac{1}{2} \sum_{k=1}^L \|\tilde{d}^k(1) - \hat{d}^k(1)\|_2^2 + \frac{1}{2} \sum_{k=1}^L \|\tilde{d}^k(3) - \hat{d}^k(3)\|_2^2 \end{aligned}$$

Demostración. Ver la Proposición B.1. □

Corolario 1.2. Consideremos el algoritmo de control del error presentado en la Proposición 1.4 y una estrategia de procesamiento de los coeficientes tal que

$$\|\tilde{d}^k(l) - \hat{d}^k(l)\|_p \leq \epsilon_k \quad \forall l = 1, 2, 3, \quad p = \infty, 1, \text{ ó } 2 \quad (1.48)$$

$$\|\bar{f}^0 - \hat{f}^0\|_p \leq \epsilon_0 \quad (1.49)$$

Entonces tenemos

$$\|\bar{f}^L - \hat{f}^L\|_\infty \leq \epsilon_0 + 2 \sum_{k=1}^L \epsilon_k,$$

$$\|\bar{f}^L - \hat{f}^L\|_1 \leq \epsilon_0 + \sum_{k=1}^L \epsilon_k,$$

$$\|\bar{f}^L - \hat{f}^L\|_2^2 \leq \epsilon_0^2 + 2 \sum_{k=1}^L \epsilon_k^2.$$

En particular, si asumimos que $\|\bar{f}^0 - \hat{f}^0\|_p = 0$ y consideramos

$$\epsilon_k = \frac{\epsilon}{2^{L-k}} \quad k = 0, \dots, L \quad (1.50)$$

obtenemos

$$\|\bar{f}^L - \hat{f}^L\|_\infty \leq 2\epsilon; \quad \|\bar{f}^L - \hat{f}^L\|_1 \leq \epsilon; \quad \|\bar{f}^L - \hat{f}^L\|_2 \leq \sqrt{\frac{2}{3}}\epsilon. \quad (1.51)$$

Nota 1.2. Tanto en valores puntuales como para medias en celda, cuando apliquemos la transformada 2D directa vía producto tensorial (sin control del error) lo que haremos será aplicar el Algoritmo A.1 (para valores puntuales) o el Algoritmo B.1 (para medias en celda) truncando con $\epsilon_k = 0$ obteniendo:

$$Mf^L = \{f^0, d^1(1), d^1(2), d^1(3), \dots, d^L(1), d^L(2), d^L(3)\}.$$

Una vez tenemos Mf^L truncamos o cuantizamos con los parámetros adecuados y obtendremos los datos que después hay que

almacenar:

$$Q_{\varepsilon} M f^L = \{\hat{f}^0, \hat{d}^1(1), \hat{d}^1(2), \hat{d}^1(3), \dots, \hat{d}^L(1), \hat{d}^L(2), \hat{d}^L(3)\}$$

2

Interpolación WENO para medias en celda

2.1

Introducción

Mediante la interpolación ENO se consigue una aproximación con un buen orden de exactitud en todos los intervalos excepto en aquellos que contienen singularidades. Pero existen algunas características de esta técnica de interpolación que se podrían mejorar:

- En primer lugar, el proceso de selección del stencil es muy sensible a las perturbaciones: si los valores de las diferencias divididas que se comparan en el criterio de selección son muy similares, una pequeña perturbación, como un error de re-

donde, por ejemplo, podría hacer que el stencil seleccionado cambiase, porque podría cambiar la dirección de la desigualdad (ver Sección 1.5).

- Por otra parte, en las regiones suaves de la función no es necesario realizar esta selección de stencil, ya que el stencil centrado que utiliza la interpolación lineal producirá la misma aproximación.
- Por último, se podría aumentar el orden de exactitud de la aproximación. El método ENO consiste en seleccionar uno de entre todos los stencils posibles, para hacer la interpolación en cada celda. Tomando stencils formados por r celdas, esto significa que la interpolación ENO se hará seleccionando un stencil entre r posibles, dando una aproximación con un orden de exactitud igual a r . Hay $2r - 1$ celdas contenidas en los r stencils, por lo que se pierde la información proporcionada por $r - 1$ de estas celdas. Si la función es suave en estas regiones, esta información podría servir para obtener una mejor aproximación, de manera que utilizando la información dada por las $2r - 1$ celdas contenidas en los distintos stencils, se podría obtener un orden de exactitud máximo igual a $2r - 1$.

Para solucionar los dos primeros problemas, se presentó en [18] y en [41] una estrategia de sesgo, que consiste en tomar como base un stencil “preferido”, que será el stencil centrado en la celda donde se realiza la interpolación, y utilizarlo para modificar el criterio de selección de stencil con un parámetro de sesgo. La idea es no alejarse de este stencil, excepto en el caso de que en el stencil alternativo la función sea “mucho” más suave, y este “mucho” viene dado por el sesgo.

Posteriormente, Liu et al. [33] introdujeron, en el contexto de las leyes de conservación, la técnica WENO, como mejora de la interpolación ENO en todos los puntos enumerados al principio de esta sección. Una versión más eficiente de esta técnica fue propuesta por Jiang y Shu en [27].

La técnica desarrollada por Liu et al. se basa en la versión para medias en celda de la interpolación ENO, y consiste en reconstruir un polinomio que interpola los valores puntuales de la solución

de una ley de conservación hiperbólica a partir de las medias en celda de la solución débil. En el contexto de interpolación para la compresión y reconstrucción de imágenes, donde una imagen se interpreta como la discretización por medias en celda de una función, la técnica WENO consistirá en utilizar medias en celda para reconstruir un polinomio que interpole el conjunto de medias en celda de origen.

La diferencia entre ENO y WENO radica en la forma en que se construye el interpolante. El interpolante ENO se construye seleccionando un stencil para cada celda, mientras que en el método WENO, a cada celda se le asignan todos los stencils posibles, y el polinomio interpolador se calcula como una combinación lineal convexa de los polinomios correspondientes a dichos stencils. En esta combinación lineal, se da más importancia a los polinomios construidos a partir de stencils en los que la función es suave, de forma que los polinomios que cruzan alguna singularidad tienen un efecto prácticamente nulo. De este modo, se conserva el efecto ENO, es decir, la interpolación en celdas próximas a singularidades se hace tomando información sólo de regiones donde la función es suave, y además, como se utiliza una combinación convexa de polinomios interpoladores, los errores cometidos por unos se pueden cancelar con los de otros, resultando en un orden de aproximación mayor.

Dicho de modo más formal, si denotamos los r stencils posibles para realizar la interpolación en la celda $I_j = [x_{j-1}, x_j]$ como:

$$S_{j+k} = \{I_{j+k-r+1}, \dots, I_{j+k}\}, \quad k = 0, \dots, r-1,$$

y llamamos $p_{j+k}(x)$ al polinomio construido a partir del stencil S_{j+k} , entonces el polinomio que se utiliza para interpolar a la función f en I_j es:

$$p_j(x) = \sum_{k=0}^{r-1} \omega_k^j p_{j+k}(x) \quad (2.1)$$

donde los pesos ω_k^j se definen de forma que el polinomio $p_j(x)$ resultante satisfaga una serie de requisitos que se explican más adelante.

El polinomio interpolador $p(x)$ vendrá dado por la unión de todos estos polinomios, es decir:

$$p(x) = p_j(x), \quad x \in I_j.$$

2.2

Propiedades del polinomio interpolador $p(x)$

El polinomio obtenido con la técnica de interpolación WENO debe cumplir las siguientes condiciones:

- Es un polinomio interpolador a trozos, formado por polinomios de grado $r - 1$ que proporcionan un orden de exactitud r :

$$p_j(x) = f(x) + O(h^r), \quad x \in I_j.$$

- Cada polinomio $p_j(x)$ satisface la “propiedad ENO”:
 1. Si la función f es suave en el stencil S_{j+k} , entonces el peso correspondiente a este stencil verifica:

$$\omega_k^j = O(1). \tag{2.2}$$

2. Si la función f presenta una singularidad en el stencil S_{j+k} , entonces su correspondiente ω_k^j verifica:

$$\omega_k^j \leq O(h^r). \tag{2.3}$$

Si los pesos ω_k^j de la combinación convexa satisfacen la propiedad ENO (2.2)-(2.3), entonces el polinomio $p_j(x)$ dado por (2.1) es una combinación lineal de los polinomios construidos a partir de stencils en los que la función es suave (por (2.2)), en la que los polinomios correspondientes a stencils que cruzan alguna singularidad tienen una contribución prácticamente nula (por (2.3)).

2.3

Definición de los pesos

Una correcta elección de los pesos de la combinación lineal (2.1) es fundamental para el éxito de la interpolación WENO. Una condición básica que deben satisfacer, por motivos de estabilidad y consistencia, es la siguiente:

$$\omega_k^j \geq 0, \forall k, \quad \sum_{k=0}^{r-1} \omega_k^j = 1. \quad (2.4)$$

Por otra parte, el polinomio interpolador $p_j(x)$ debe cumplir la propiedad ENO (2.2)-(2.3), por lo que en la definición de los pesos tendrá una gran importancia la elección de una medida de la suavidad de la función en cada stencil S_{j+k} . Esta medida viene dada por un *indicador de suavidad*, IS_{j+k} .

Estas consideraciones llevan a la siguiente expresión propuesta por Liu et al. [33] para los pesos:

$$\omega_k^j = \frac{\alpha_k^j}{\sum_{l=0}^{r-1} \alpha_l^j}, \quad (2.5)$$

donde α_k^j viene dado por la expresión:

$$\alpha_k^j = \frac{C_k^j}{(\varepsilon + IS_{j+k})^r}, \quad k = 0, \dots, r-1. \quad (2.6)$$

Aquí, ε es una constante positiva que se introduce para evitar que el denominador se anule. En la literatura se han venido utilizando los valores $\varepsilon = 10^{-5}$ o $\varepsilon = 10^{-6}$. Las C_k^j son constantes, también positivas, llamadas *pesos óptimos*, cuya procedencia se explica a continuación.

Nota 2.1. En [27], Jiang y Shu sugieren utilizar en (2.6) como exponente 2 en lugar de r , tal como hacen Liu et al.

Proposición 2.1. Para que los pesos de la combinación lineal satisfagan la propiedad ENO (2.2)-(2.3), es suficiente que el indicador de suavidad cumpla la siguiente condición:

- Si la función f es suave en el stencil S_{j+k} , entonces:

$$IS_{j+k} = O(h^2), \quad (2.7)$$

- Si la función f presenta una singularidad en el stencil S_{j+k} , entonces:

$$IS_{j+k} = O(1). \quad (2.8)$$

Demostración. Comprobemos esta afirmación:

Cuando la función f es suave en el stencil S_{j+k} ,

$$IS_{j+k} = O(h^2) \Rightarrow \alpha_k^j = \frac{O(1)}{(\varepsilon + O(h^2))^r} \Rightarrow \sum_{l=0}^{r-1} \alpha_l^j = \frac{O(1)}{(\varepsilon + O(h^2))^r}$$

y así:

$$\omega_k^j = O(1).$$

Si S_{j+k} contiene alguna singularidad de la función f ,

$$IS_{j+k} = O(1) \Rightarrow \alpha_k^j = \frac{O(1)}{(\varepsilon + O(1))^r} = O(1).$$

Si asumimos que la función es suave en al menos uno de los stencils posibles, entonces:

$$\sum_{l=0}^{r-1} \alpha_l^j = \frac{O(1)}{(\varepsilon + O(h^2))^r}$$

por lo que se tiene:

$$\omega_k^j = \frac{O(1)}{\frac{O(1)}{(\varepsilon + O(h^2))^r}} \leq \max\{O(\varepsilon^r), O(h^{2r})\}.$$

Si $\varepsilon = 10^{-6}$, como $r \geq 2$ se tiene que $O(\varepsilon^r) \leq O(10^{-12})$, con lo cual la propiedad ENO se cumple. \square

2.4

Pesos óptimos

Consideramos los r polinomios $p_{j+k}(x)$, $k = 0, \dots, r-1$, que interpolan a la función f en los stencils S_{j+k} . En el contexto de multirresolución en el que nos encontramos la interpolación se utiliza como un proceso de reconstrucción para recuperar la información de un nivel de resolución a partir de la información contenida en el nivel inmediatamente inferior.

En el caso de interpolación por medias en celda (ver Sección 1.3), esto se hace dividiendo cada celda I_j^{l-1} del nivel de resolución $l-1$ en dos celdas iguales, I_{2j-1}^l e I_{2j}^l del nivel de resolución l , y calculando la media del polinomio interpolador en la celda I_{2j-1}^l . De este modo, en cada celda I_{2j-1}^l obtenemos r aproximaciones \hat{p}_{j+k}^r a \bar{f}_{2j-1}^l . Como podemos considerar el nivel de resolución l como fijo, en el resto del capítulo eliminaremos el superíndice que indica el nivel de resolución para facilitar la notación, y simplemente denotaremos \bar{f}_{2j-1} en lugar de \bar{f}_{2j-1}^l .

$$\hat{p}_{j+k}^r = \bar{f}_{2j-1} + O(h^r). \quad (2.9)$$

Por otra parte, el stencil que resulta de la unión de los r stencils S_{j+k} contiene $2r-1$ celdas, y con él podemos calcular una aproximación \hat{p}_j^{2r-1} con un mayor orden de exactitud, igual a $2r-1$. Si la función f es suave en todos los stencils, entonces existen constantes C_k^j tales que:

$$\hat{p}_j^{2r-1} = \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r.$$

Estas constantes C_k^j (pesos óptimos) son siempre positivas y, por consistencia, cumplen:

$$\sum_{k=0}^{r-1} C_k^j = 1. \quad (2.10)$$

Sería deseable, para conseguir el mayor orden de aproximación posible con el interpolante WENO, que los pesos de la combinación

lineal convexa (2.1) aproximen a estos pesos óptimos. De hecho tenemos la siguiente proposición:

Proposición 2.2. Si los pesos ω_k^j cumplen la siguiente relación:

$$\omega_k^j = C_k^j + O(h^m), \quad k = 0, \dots, r-1, \quad (2.11)$$

con $1 \leq m \leq r-1$, entonces

$$\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r = \bar{f}_{2j-1} + O(h^{\min(r+m, 2r-1)}) \quad (2.12)$$

Demostración. Comprobemos esta afirmación:

$$\begin{aligned} & \sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \hat{p}_j^{2r-1} = \\ &= \sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r \\ &= \sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r + \bar{f}_{2j-1} \sum_{k=0}^{r-1} C_k^j - \bar{f}_{2j-1} \sum_{k=0}^{r-1} \omega_k^j \\ &= \sum_{k=0}^{r-1} \omega_k^j (\hat{p}_{j+k}^r - \bar{f}_{2j-1}) - \sum_{k=0}^{r-1} C_k^j (\hat{p}_{j+k}^r - \bar{f}_{2j-1}) \\ &= \sum_{k=0}^{r-1} (\omega_k^j - C_k^j) (\hat{p}_{j+k}^r - \bar{f}_{2j-1}) \\ &= \sum_{k=0}^{r-1} O(h^m) O(h^r) = O(h^{r+m}) \end{aligned} \quad (2.13)$$

Aquí utilizamos el hecho de que $\sum_{k=0}^{r-1} \omega_k^j = \sum_{k=0}^{r-1} C_k^j = 1$, por (2.4) y (2.10), y también, en la penúltima igualdad, las relaciones dadas por (2.9) y (2.11).

A partir de esta relación concluimos que el interpolante WENO aproxima a la función con un orden de exactitud igual a $\min(r +$

$m, 2r - 1)$, ya que:

$$\begin{aligned} \sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \bar{f}_{2j-1} &= \left(\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \hat{p}_j^{2r-1} \right) + (\hat{p}_j^{2r-1} - \bar{f}_{2j-1}) \\ &= O(h^{r+m}) + O(h^{2r-1}) = O(h^{\min(r+m, 2r-1)}) \end{aligned}$$

□

Nota 2.2. Si $m = 1$ tendremos

$$\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r = \bar{f}_{2j-1} + O(h^{r+1}),$$

es decir, mejoramos el orden de aproximación de \hat{p}_{j+k}^r en 1, consiguiendo $r + 1$. En cambio, si $m = r - 1$

$$\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r = \bar{f}_{2j-1} + O(h^{2r-1})$$

con lo que conseguimos un orden de exactitud igual al de \hat{p}_j^{2r-1} que es $2r - 1$. Es decir, conseguimos el orden óptimo.

Los pesos óptimos C_k^j se pueden calcular del siguiente modo:

1. Dado r , tomamos las aproximaciones \hat{p}_{j+k}^r obtenidas a partir de los stencils $S_{j+k} = \{I_{j+k-r+1}, \dots, I_{j+k}\}$. Por otra parte, consideramos \hat{p}_j^{2r-1} calculada a partir del stencil que resulta de la unión de los stencils anteriores, es decir, $S = \{I_{j-r+1}, \dots, I_{j+r-1}\}$.
2. Se igualan coeficientes en la combinación:

$$\hat{p}_j^{2r-1} = \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r$$

3. Por último, se resuelve el sistema de ecuaciones resultante, en el que las incógnitas son C_k^j , $k = 0, \dots, r - 1$.

Cálculo de los pesos óptimos para $r = 2$

A partir de los polinomios interpoladores, calculamos las aproximaciones:

$$\begin{aligned}\hat{p}_j^2 &= \frac{1}{4}\bar{f}_{j-1} + \frac{3}{4}\bar{f}_j \\ \hat{p}_{j+1}^2 &= \frac{5}{4}\bar{f}_j - \frac{1}{4}\bar{f}_{j+1}\end{aligned}$$

y también:

$$\hat{p}_j^3 = \frac{1}{8}\bar{f}_{j-1} + \bar{f}_j - \frac{1}{8}\bar{f}_{j+1}.$$

Consideramos la combinación:

$$\hat{p}_j^3 = C_0^j \hat{p}_j^2 + C_1^j \hat{p}_{j+1}^2$$

e igualamos coeficientes, obteniendo el siguiente sistema de ecuaciones:

$$\begin{aligned}\frac{1}{8} &= \frac{1}{4}C_0^j \\ 1 &= \frac{3}{4}C_0^j + \frac{5}{4}C_1^j \\ -\frac{1}{8} &= -\frac{1}{4}C_1^j\end{aligned}$$

cuya solución es:

$$C_0^j = C_1^j = \frac{1}{2}.$$

Cálculo de los pesos óptimos para $r = 3$

Si aumentamos el orden de la interpolación, y consideramos $r = 3$, las aproximaciones en este caso son las siguientes:

$$\begin{aligned}\hat{p}_j^3 &= -\frac{1}{8}\bar{f}_{j-2} + \frac{1}{2}\bar{f}_{j-1} + \frac{5}{8}\bar{f}_j \\ \hat{p}_{j+1}^3 &= \frac{1}{8}\bar{f}_{j-1} + \bar{f}_j - \frac{1}{8}\bar{f}_{j+1} \\ \hat{p}_{j+2}^3 &= \frac{11}{8}\bar{f}_j - \frac{1}{2}\bar{f}_{j+1} + \frac{1}{8}\bar{f}_{j+2}\end{aligned}$$

y la aproximación construida a partir de las 5 celdas es:

$$\hat{p}_j^5 = -\frac{3}{128}\bar{f}_{j-2} + \frac{22}{128}\bar{f}_{j-1} + \bar{f}_j - \frac{22}{128}\bar{f}_{j+1} + \frac{3}{128}\bar{f}_{j+2}.$$

Considerando la combinación de las distintas aproximaciones e igualando coeficientes, tenemos el sistema:

$$\begin{aligned} -\frac{3}{128} &= -\frac{1}{8}C_0^j \\ \frac{22}{128} &= \frac{1}{2}C_0^j + \frac{1}{8}C_1^j \\ 1 &= \frac{5}{8}C_0^j + C_1^j + \frac{11}{8}C_2^j \\ -\frac{22}{128} &= -\frac{1}{8}C_1^j - \frac{1}{2}C_2^j \\ \frac{3}{128} &= \frac{1}{8}C_2^j. \end{aligned}$$

La solución de este sistema viene dada por:

$$C_0^j = C_2^j = \frac{3}{16} \text{ y } C_1^j = \frac{10}{16}$$

Resumiendo, hemos obtenido los siguientes pesos óptimos:

	$k = 0$	$k = 1$	$k = 2$
$r = 2$	1/2	1/2	-
$r = 3$	3/16	10/16	3/16

Tabla 2.1: Pesos óptimos para $r = 2, 3$.

2.5

Indicadores de suavidad

Como ha quedado explicado anteriormente, es importante en la construcción del interpolante WENO encontrar una forma conveniente de medir la suavidad de la función que se interpola. Para ello se utiliza una función llamada *indicador de suavidad*.

Recordando lo expuesto en la Sección 2.3, si el indicador de suavidad cumple $IS_{j+k} = O(h^2)$ en los stencils S_{j+k} en los que la función es suave, e $IS_{j+k} = O(1)$ en los stencils S_{j+k} en los que la función cruza alguna singularidad, entonces los pesos (2.5) satisfacen la propiedad ENO (2.2)-(2.3).

Por otro lado, tenemos el siguiente resultado:

Proposición 2.3. *Si el indicador de suavidad satisface la relación*

$$IS_{j+k} = D(1 + O(h^m)), \quad k = 0, \dots, r-1, \quad (2.14)$$

con D una constante distinta de cero e independiente de k , aunque puede depender de h , entonces los pesos ω_k^j satisfacen la relación (2.11).

Demostración. Partiendo de la expresión (2.5) para los pesos ω_k^j :

$$\omega_k^j = \frac{\alpha_k^j}{\sum_{l=0}^{r-1} \alpha_l^j} = \frac{C_k^j}{(\varepsilon + IS_{j+k})^r} \cdot \frac{1}{\sum_{l=0}^{r-1} \frac{C_l^j}{(\varepsilon + IS_{j+l})^r}}. \quad (2.15)$$

Haciendo uso de la relación (2.14), y teniendo en cuenta que ε es una constante que se puede despreciar, D es una constante independiente del stencil considerado, y $\sum_{l=0}^{r-1} C_l^j = 1$ (por (2.10)), tenemos:

$$\begin{aligned} \omega_k^j &= \frac{C_k^j}{[D(1 + O(h^m))]^r} \cdot \frac{1}{\sum_{l=0}^{r-1} \frac{C_l^j}{[D(1 + O(h^m))]^r}} \\ &= \frac{C_k^j}{[D(1 + O(h^m))]^r} \cdot \frac{[D(1 + O(h^m))]^r}{\sum_{l=0}^{r-1} C_l^j} \\ &= \frac{C_k^j}{[D(1 + O(h^m))]^r} [D(1 + O(h^m))]^r \\ &= \frac{C_k^j}{(1 + O(h^m))^r} (1 + O(h^m))^r \end{aligned} \quad (2.16)$$

Por otra parte, desarrollando la potencia $(1 + O(h^m))^r$, obtenemos:

$$(1 + O(h^m))^r = 1 + O(h^m) + O(h^{2m}) + \dots + O(h^{rm}) = 1 + O(h^m)$$

con lo que llegamos al resultado buscado:

$$\omega_k^j = \frac{C_k^j}{(1 + O(h^m))} (1 + O(h^m)) = C_k^j + O(h^m) \quad (2.17)$$

□

2.5.1

Indicador de suavidad de Liu et al.

La interpolación ENO consiste en seleccionar el stencil en el que la función f es más suave, y para ello se comparan las diferencias divididas (o no divididas) de la función (ver Sección 1.5). Esto es así porque estas diferencias se pueden utilizar para medir la suavidad de una función en un stencil. Partiendo de esta idea, Liu et al. introdujeron en [33] un indicador de suavidad basado en diferencias no divididas de las medias en celda de una función.

Para definir el indicador de suavidad IS_j de $f(x)$ en el stencil $S_j = \{I_{j-r+1}, \dots, I_j\}$ (donde $I_j = [x_{j-1}, x_j]$), en primer lugar se construye una tabla de diferencias de las medias en celda $\{\bar{f}_k\}$ cuya información está contenida en el stencil:

$$\begin{array}{ccccccc} \Delta[\bar{f}_{j-r+1}] & & & & & & \\ \Delta[\bar{f}_{j-r+2}] & \Delta^2[\bar{f}_{j-r+1}] & & & & & \\ \vdots & \vdots & \ddots & & & & \\ \Delta[\bar{f}_{j-2}] & \Delta^2[\bar{f}_{j-3}] & & \ddots & & & \\ \Delta[\bar{f}_{j-1}] & \Delta^2[\bar{f}_{j-2}] & \cdots & \cdots & \Delta^{r-1}[\bar{f}_{j-r+1}] & & \end{array}$$

en la que:

$$\begin{aligned} \Delta[\bar{f}_i] &= \bar{f}_{i+1} - \bar{f}_i \\ \Delta^k[\bar{f}_i] &= \Delta^{k-1}[\bar{f}_{i+1}] - \Delta^{k-1}[\bar{f}_i]. \end{aligned}$$

Entonces se define el indicador de suavidad IS_j como la suma de todas las medias aritméticas de los cuadrados de las diferencias del mismo orden:

$$IS_j = \sum_{l=1}^{r-1} \left(\sum_{k=1}^l (\Delta^{r-l}[\bar{f}_{j-k+r}])^2 \right) / l. \quad (2.18)$$

En particular, obtenemos la siguiente expresión para el indicador de suavidad cuando $r = 2$:

$$IS_j = (\Delta[\bar{f}_{j-1}])^2$$

o equivalentemente:

$$IS_{j+k} = (\bar{f}_{j+k} - \bar{f}_{j+k-1})^2, \quad k = 0, 1. \quad (2.19)$$

En el caso $r = 3$, el indicador de suavidad viene dado por la expresión:

$$IS_j = \frac{1}{2} \left[(\Delta[\bar{f}_{j-2}])^2 + (\Delta[\bar{f}_{j-1}])^2 \right] + (\Delta^2[\bar{f}_{j-2}])^2$$

es decir:

$$IS_{j+k} = \frac{1}{2} \left[(\bar{f}_{j+k-1} - \bar{f}_{j+k-2})^2 + (\bar{f}_{j+k} - \bar{f}_{j+k-1})^2 \right] + (\bar{f}_{j+k} - 2\bar{f}_{j+k-1} + \bar{f}_{j+k-2})^2, \quad k = 0, 1, 2. \quad (2.20)$$

Sea $F(x) = \int_0^x f(y) dy$ una primitiva de la función f . Si tomamos una diferencia cualquiera de primer orden $\Delta[\bar{f}_l]$ y la expresamos en función de los valores puntuales de la primitiva utilizando la relación (1.17), tenemos:

$$\Delta[\bar{f}_l] = \bar{f}_{l+1} - \bar{f}_l = \frac{F_{l+1} - F_l}{h} - \frac{F_l - F_{l-1}}{h} = \frac{1}{h}(F_{l+1} - 2F_l + F_{l-1}).$$

Desarrollamos esta expresión por Taylor entorno a x_j , suponiendo que la función es suave:

$$\begin{aligned} & \frac{1}{h}(F_{l+1} - 2F_l + F_{l-1}) = \\ & = \frac{1}{h} \left[F_j + (l+1-j)hF'_j + \frac{(l+1-j)^2}{2}h^2F''_j + O(h^3) \right. \\ & \quad - 2(F_j + (l-j)hF'_j + \frac{(l-j)^2}{2}h^2F''_j + O(h^3)) \\ & \quad \left. + F_j + (l-1-j)hF'_j + \frac{(l-1-j)^2}{2}h^2F''_j + O(h^3) \right] \\ & = \frac{1}{h}(h^2F''_j + O(h^3)) \\ & = hF''_j + O(h^2) = hf'_j + O(h^2) = O(h) \end{aligned} \quad (2.21)$$

de donde:

$$\Delta^2[\bar{f}_i] = \Delta[\bar{f}_{i+1}] - \Delta[\bar{f}_i] = O(h) - O(h) = O(h)$$

y por inducción se tiene que:

$$\Delta^k[\bar{f}_i] = O(h), \quad \forall k.$$

Como consecuencia de esto concluimos que en regiones de suavidad de la función f , el indicador de suavidad (2.18) verifica:

$$IS_j = O(h^2).$$

Por otra parte, la magnitud de una discontinuidad es siempre $O(1)$, independientemente del tamaño de las celdas, y por ello se tiene que $IS_j = O(1)$ cuando el stencil S_j cruza una singularidad.

De este modo vemos que el indicador de suavidad definido por Liu et al. satisface el primero de los requisitos expuestos al principio de la Sección 2.5, con lo cual, como se vio en la Sección 2.3, los pesos construidos a partir de dicho indicador verifican la propiedad ENO (2.2)-(2.3).

Queda por ver si este indicador de suavidad cumple también el requisito (2.14). Para ello, tomamos las expresiones del indicador para los casos $r = 2, 3$, y hacemos el desarrollo de Taylor en las regiones suaves de la función f , recurriendo como antes a la función primitiva. En primer lugar, consideramos el caso $r = 2$:

$$\begin{aligned} IS_{j+k} &= (\bar{f}_{j+k} - \bar{f}_{j+k-1})^2 \\ &= (hF_j'' + O(h^2))^2 \\ &= (hF_j'')^2 + O(h^4) + 2hF_j''O(h^2) \\ &= (hF_j'')^2 + O(h^3) \\ &= (hf_j')^2(1 + O(h)), \quad k = 0, 1, \end{aligned}$$

donde en la primera igualdad hemos utilizado el desarrollo (2.21).

Análogamente, y aprovechando este resultado, desarrollamos

la expresión del indicador de suavidad para $r = 3$:

$$\begin{aligned}
IS_{j+k} &= \frac{1}{2} [(\bar{f}_{j+k-1} - \bar{f}_{j+k-2})^2 + (\bar{f}_{j+k} - \bar{f}_{j+k-1})^2] \\
&+ (\bar{f}_{j+k} - 2\bar{f}_{j+k-1} + \bar{f}_{j+k-2})^2 \\
&= \frac{1}{2} [(\bar{f}_{j+k-1} - \bar{f}_{j+k-2})^2 + (\bar{f}_{j+k} - \bar{f}_{j+k-1})^2] \\
&+ [(\bar{f}_{j+k} - \bar{f}_{j+k-1}) - (\bar{f}_{j+k-1} - \bar{f}_{j+k-2})]^2 \\
&= \frac{1}{2} [(hF_j'')^2(1 + O(h)) + (hF_j'')^2(1 + O(h))] \\
&+ (hF_j'' + O(h^2) - hF_j'' - O(h^2))^2 \\
&= (hF_j'')^2(1 + O(h)) + O(h^4) \\
&= (hf_j')^2(1 + O(h)), \quad k = 0, 1, 2.
\end{aligned}$$

De este modo vemos que el indicador de suavidad verifica en ambos casos la siguiente relación:

$$IS_{j+k} = D(h)(1 + O(h)), \quad k = 0, \dots, r-1 \quad (r = 2, 3) \quad (2.22)$$

donde $D(h) = (hf_j')^2$ es una constante que depende de h , pero no de k , y si además la derivada de f no se anula en x_j , tendremos $D(h) \neq 0$.

En el Capítulo 6 se analiza el caso en que algunas derivadas se anulen en un cierto punto. Los resultados allí expuestos se pueden extender a este capítulo.

La relación (2.22) se corresponde con la relación (2.14) tomando $m = 1$, y por lo tanto el orden de exactitud que alcanza el interpolante WENO construido utilizando el indicador de suavidad (2.18) no será el óptimo $2r - 1$. De hecho, de las Proposiciones 2.2 y 2.3 llegamos a la conclusión de que con este indicador de suavidad el orden de exactitud alcanzado por el interpolante WENO es $r + 1$:

$$\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \bar{f}_{2j-1} = O(h^{r+1}) + O(h^{2r-1}) = O(h^{r+1})$$

Para mejorar este resultado Jiang y Shu presentaron en [27] un nuevo indicador de suavidad, que permite obtener un orden de exactitud mayor.

2.5.2

Indicador de suavidad de Jiang y Shu

Jiang y Shu [27] parten de un enfoque diferente al utilizado por Liu et al. para definir su indicador de suavidad. Como ya hemos visto, a partir de cada uno de los r stencils S_{j+k} se construye un polinomio interpolador que proporciona una aproximación al valor \bar{f}_{2j-1} . La variación total es una buena forma de medir la suavidad de una función, por lo que sería deseable minimizar la variación total de dicha aproximación. Utilizando esta idea como base, Jiang y Shu definieron el siguiente indicador:

$$IS_{j+k} = \sum_{l=1}^{r-1} \int_{x_{j-1}}^{x_j} h^{2l+1} (p_{j+k}^{(l)}(x))^2 dx \quad (2.23)$$

donde $p_{j+k}^{(l)}(x)$ es la derivada l -ésima del polinomio interpolador $p_{j+k}(x)$ construido a partir del stencil S_{j+k} . El término h^{2l+1} se introduce para eliminar factores dependientes de h en las derivadas de los polinomios.

La expresión (2.23) no es más que la suma de la norma L^2 de las derivadas de los polinomios interpoladores $p_{j+k}(x)$ sobre el intervalo $I_j = [x_{j-1}, x_j]$, de forma similar a la variación total, que está basada en la norma L^1 .

El procedimiento para calcular los indicadores de suavidad correspondientes a cada valor de r es el siguiente: en primer lugar se construyen los polinomios interpoladores $p_{j+k}(x)$ cuyas medias en celda coinciden con las de la función f , a continuación se calculan sus derivadas y se evalúa la suma de las integrales de los cuadrados de estas derivadas normalizados con el término h^{2l+1} .

Cálculo del indicador de suavidad para $r = 2$

Consideramos los polinomios interpoladores, centrados en $x_{j-\frac{1}{2}}$, construidos sobre los stencils $S_j = \{I_{j-1}, I_j\}$ y $S_{j+1} = \{I_j, I_{j+1}\}$:

$$\begin{aligned} p_j(x) &= \frac{\bar{f}_j}{h} + \frac{\bar{f}_j - \bar{f}_{j-1}}{h^2} (x - x_{j-\frac{1}{2}}) \\ p_{j+1}(x) &= \frac{\bar{f}_j}{h} + \frac{\bar{f}_{j+1} - \bar{f}_j}{h^2} (x - x_{j-\frac{1}{2}}) \end{aligned}$$

cuyas derivadas son:

$$\begin{aligned} p'_j(x) &= \frac{\bar{f}_j - \bar{f}_{j-1}}{h^2} \\ p'_{j+1}(x) &= \frac{\bar{f}_{j+1} - \bar{f}_j}{h^2} \end{aligned}$$

Sustituyendo en (2.23) obtenemos la expresión para el indicador de suavidad cuando $r = 2$, que coincide con la que venía dada por la definición de Liu et al. (2.18):

$$\begin{aligned} IS_j &= (\bar{f}_j - \bar{f}_{j-1})^2 \\ IS_{j+1} &= (\bar{f}_{j+1} - \bar{f}_j)^2. \end{aligned}$$

Este indicador de suavidad ya ha sido analizado en la Sección 2.5.1, por lo que nos centraremos en el análisis del indicador que se obtiene en el caso $r = 3$.

Cálculo del indicador de suavidad para $r = 3$

En este caso, las expresiones dadas por el indicador de Jiang y Shu difieren del de Liu et al. Los polinomios interpoladores son:

$$\begin{aligned} p_j(x) &= \frac{23\bar{f}_j + 2\bar{f}_{j-1} - \bar{f}_{j-2}}{24h} + \frac{3\bar{f}_j - 4\bar{f}_{j-1} + \bar{f}_{j-2}}{2h^2}(x - x_{j-\frac{1}{2}}) \\ &+ \frac{\bar{f}_j - 2\bar{f}_{j-1} + \bar{f}_{j-2}}{2h^3}(x - x_{j-\frac{1}{2}})^2; \\ p_{j+1}(x) &= \frac{-\bar{f}_{j+1} + 26\bar{f}_j - \bar{f}_{j-1}}{24h} + \frac{\bar{f}_{j+1} - \bar{f}_{j-1}}{2h^2}(x - x_{j-\frac{1}{2}}) \\ &+ \frac{\bar{f}_{j+1} - 2\bar{f}_j + \bar{f}_{j-1}}{2h^3}(x - x_{j-\frac{1}{2}})^2; \\ p_{j+2}(x) &= \frac{-\bar{f}_{j+2} + 2\bar{f}_{j+1} + 23\bar{f}_j}{24h} + \frac{-\bar{f}_{j+2} + 4\bar{f}_{j+1} - 3\bar{f}_j}{2h^2}(x - x_{j-\frac{1}{2}}) \\ &+ \frac{\bar{f}_{j+2} - 2\bar{f}_{j+1} + \bar{f}_j}{2h^3}(x - x_{j-\frac{1}{2}})^2. \end{aligned}$$

Derivando estos polinomios obtenemos las expresiones siguien-

tes:

$$\begin{aligned}
 p'_j(x) &= \frac{3\bar{f}_j - 4\bar{f}_{j-1} + \bar{f}_{j-2}}{2h^2} + \frac{\bar{f}_j - 2\bar{f}_{j-1} + \bar{f}_{j-2}}{h^3}(x - x_{j-\frac{1}{2}}) \\
 p''_j(x) &= \frac{\bar{f}_j - 2\bar{f}_{j-1} + \bar{f}_{j-2}}{h^3}; \\
 p'_{j+1}(x) &= \frac{\bar{f}_{j+1} - \bar{f}_{j-1}}{2h^2} + \frac{\bar{f}_{j+1} - 2\bar{f}_j + \bar{f}_{j-1}}{h^3}(x - x_{j-\frac{1}{2}}) \\
 p''_{j+1}(x) &= \frac{\bar{f}_{j+1} - 2\bar{f}_j + \bar{f}_{j-1}}{h^3}; \\
 p'_{j+2}(x) &= \frac{-\bar{f}_{j+2} + 4\bar{f}_{j+1} - 3\bar{f}_j}{2h^2} + \frac{\bar{f}_{j+2} - 2\bar{f}_{j+1} + \bar{f}_j}{h^3}(x - x_{j-\frac{1}{2}}) \\
 p''_{j+2}(x) &= \frac{\bar{f}_{j+2} - 2\bar{f}_{j+1} + \bar{f}_j}{h^3}.
 \end{aligned}$$

Sustituyendo estas derivadas en la expresión (2.23) obtenemos los indicadores de Jiang y Shu para $r = 3$:

$$\begin{aligned}
 IS_j &= \frac{13}{12}(\bar{f}_j - 2\bar{f}_{j-1} + \bar{f}_{j-2})^2 + \frac{1}{4}(3\bar{f}_j - 4\bar{f}_{j-1} + \bar{f}_{j-2})^2 \\
 IS_{j+1} &= \frac{13}{12}(\bar{f}_{j+1} - 2\bar{f}_j + \bar{f}_{j-1})^2 + \frac{1}{4}(\bar{f}_{j+1} - \bar{f}_{j-1})^2 \\
 IS_{j+2} &= \frac{13}{12}(\bar{f}_{j+2} - 2\bar{f}_{j+1} + \bar{f}_j)^2 + \frac{1}{4}(-\bar{f}_{j+2} + 4\bar{f}_{j+1} - 3\bar{f}_j)^2
 \end{aligned} \tag{2.24}$$

Veamos qué orden de exactitud se consigue utilizando el interpolante WENO construido a partir de estos indicadores. El siguiente resultado será de gran utilidad para calcular los desarrollos de Taylor de los indicadores de suavidad. Si consideramos el desarrollo (2.21) entorno a $x_{j-\frac{1}{2}}$, obtenemos:

$$\begin{aligned}
 \bar{f}_{l+1} - \bar{f}_l &= \frac{1}{h}(F_{l+1} - 2F_l + F_{l-1}) \\
 &= \frac{1}{h} \left(h^2 F''_{j-\frac{1}{2}} + \left(l - j + \frac{1}{2} \right) h^3 F'''_{j-\frac{1}{2}} + O(h^4) \right) \\
 &= h F''_{j-\frac{1}{2}} + \left(l - j + \frac{1}{2} \right) h^2 F'''_{j-\frac{1}{2}} + O(h^3) \\
 &= h f'_{j-\frac{1}{2}} + \left(l - j + \frac{1}{2} \right) h^2 f''_{j-\frac{1}{2}} + O(h^3).
 \end{aligned} \tag{2.25}$$

Reescribiendo las expresiones de los indicadores podemos expresarlos en función de unidades del tipo de la que aparece en la expresión anterior, con lo cual el cálculo del desarrollo se simplifica:

$$\begin{aligned}
IS_j &= \frac{13}{12} [(\bar{f}_j - \bar{f}_{j-1}) - (\bar{f}_{j-1} - \bar{f}_{j-2})]^2 + \frac{1}{4} [3(\bar{f}_j - \bar{f}_{j-1}) - (\bar{f}_{j-1} - \bar{f}_{j-2})]^2 \\
&= \frac{13}{12} \left(hF''_{j-\frac{1}{2}} - \frac{h^2}{2}F'''_{j-\frac{1}{2}} - hF''_{j-\frac{1}{2}} + \frac{3}{2}h^2F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&+ \frac{1}{4} \left(3hF''_{j-\frac{1}{2}} - \frac{3}{2}h^2F'''_{j-\frac{1}{2}} - hF''_{j-\frac{1}{2}} + \frac{3}{2}h^2F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&= \frac{13}{12} \left(h^2F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 + \frac{1}{4} \left(2hF''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&= \left(hF''_{j-\frac{1}{2}} \right)^2 + O(h^4) \\
&= \left(hf'_{j-\frac{1}{2}} \right)^2 (1 + O(h^2))
\end{aligned}$$

$$\begin{aligned}
IS_{j+1} &= \frac{13}{12} [(\bar{f}_{j+1} - \bar{f}_j) - (\bar{f}_j - \bar{f}_{j-1})]^2 + \frac{1}{4} [(\bar{f}_{j+1} - \bar{f}_j) + (\bar{f}_j - \bar{f}_{j-1})]^2 \\
&= \frac{13}{12} \left(hF''_{j-\frac{1}{2}} + \frac{1}{2}h^2F'''_{j-\frac{1}{2}} - hF''_{j-\frac{1}{2}} + \frac{1}{2}h^2F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&+ \frac{1}{4} \left(hF''_{j-\frac{1}{2}} + \frac{1}{2}h^2F'''_{j-\frac{1}{2}} + hF''_{j-\frac{1}{2}} - \frac{1}{2}h^2F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&= \frac{13}{12} \left(h^2F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 + \frac{1}{4} \left(2hF''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&= \left(hf'_{j-\frac{1}{2}} \right)^2 (1 + O(h^2))
\end{aligned}$$

$$\begin{aligned}
IS_{j+2} &= \frac{13}{12} [(\bar{f}_{j+2} - \bar{f}_{j+1}) - (\bar{f}_{j+1} - \bar{f}_j)]^2 + \frac{1}{4} [-(\bar{f}_{j+2} - \bar{f}_{j+1}) + 3(\bar{f}_{j+1} - \bar{f}_j)]^2 \\
&= \frac{13}{12} \left(hF''_{j-\frac{1}{2}} + \frac{3h^2}{2}F'''_{j-\frac{1}{2}} - hF''_{j-\frac{1}{2}} - \frac{h^2}{2}F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&+ \frac{1}{4} \left(-hF''_{j-\frac{1}{2}} - \frac{3h^2}{2}F'''_{j-\frac{1}{2}} + 3hF''_{j-\frac{1}{2}} + \frac{3h^2}{2}F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 \\
&= \frac{13}{12} \left(h^2F'''_{j-\frac{1}{2}} + O(h^3) \right)^2 + \frac{1}{4} (2hF''_{j-\frac{1}{2}} + O(h^3))^2 \\
&= \left(hf'_{j-\frac{1}{2}} \right)^2 (1 + O(h^2))
\end{aligned}$$

Por tanto llegamos a la conclusión de que este indicador de suavidad proporciona el orden óptimo de exactitud para la interpolación por medias en celda, ya que tenemos:

$$IS_{j+k} = (hf'_{j-\frac{1}{2}})^2(1 + O(h^2)), \quad k = 0, 1, 2, \quad (2.26)$$

y esto implica, por la Proposición 2.2:

$$\omega_k^j = C_k^j + O(h^2), \quad k = 0, 1, 2.$$

De la Proposición 2.3 deducimos que el orden de exactitud que se consigue es el óptimo, $O(h^5)$.

Por otra parte, el resultado (2.26) demuestra también que en zonas de suavidad de la función el indicador de suavidad cumple:

$$IS_{j+k} = O(h^2).$$

Además, siguiendo el mismo razonamiento usado en el caso del indicador definido por Liu et al., si el stencil S_{j+k} cruza alguna singularidad se tiene $IS_{j+k} = O(1)$. Por tanto, el indicador de suavidad definido por Jiang y Shu cumple la propiedad ENO (2.2)-(2.3).

2.5.3

Indicador de suavidad modificado

En los experimentos numéricos realizados (ver Capítulo 3) se observa que los resultados obtenidos mediante los indicadores anteriores mejoran, en general, los obtenidos con la técnica ENO. Sin embargo, en el caso de imágenes geométricas es la técnica ENO-SR la que proporciona los mejores resultados cuando utilizamos el mecanismo de error-control (en caso contrario la mejor elección resulta ser la interpolación WENO).

Esta situación cambia cuando trabajamos con imágenes reales. En este caso los mejores resultados se obtienen utilizando interpolación lineal, y el interpolante WENO sólo mejora ligeramente dichos resultados en algunos casos (con un coste computacional bastante superior). Es por ello que se plantea la necesidad de buscar un nuevo indicador de suavidad, con el que se obtengan mejores resultados. Para ello, partimos del indicador definido

por Jiang y Shu, visto en la sección anterior, y proponemos una modificación.

Nuestro objetivo es definir un indicador de suavidad para la función f , basado en sus medias en celda. Del mismo modo que para construir el polinomio interpolador de dichas medias utilizamos la técnica de reconstrucción por primitiva, podríamos pensar en utilizar los valores puntuales de la función primitiva de f para construir el indicador de suavidad.

La idea es construir un indicador basado en los valores puntuales de F (función primitiva de f), utilizando la definición dada por Jiang y Shu (2.23):

$$IS_{j+k}(F) = \sum_{l=1}^r \int_{x_{j-1}}^{x_j} h^{2l-1} \left(q_{j+k}^{(l)}(x) \right)^2 dx$$

donde $q_{j+k}(x)$ es el polinomio que interpola a F en el stencil de nodos $S_{j+k} = \{x_{j+k-r}, \dots, x_{j+k}\}$.

Una vez calculado este indicador, utilizamos la relación (1.17) existente entre las medias en celda de una función y los valores puntuales de su primitiva, para convertir el indicador anterior en un indicador basado en las medias en celda de f .

Dado que mediante la técnica de reconstrucción por primitiva calculamos el polinomio interpolador de $\{\bar{f}_i\}$ como la derivada del polinomio interpolador de $\{F_i\}$:

$$p_j(x) = \frac{d}{dx} q_j(x),$$

el proceso descrito arriba es equivalente a calcular el indicador para las medias en celda de f del siguiente modo:

$$IS_{j+k}(f) = \sum_{l=0}^{r-1} \int_{x_{j-1}}^{x_j} h^{2l+1} \left(p_{j+k}^{(l)}(x) \right)^2 dx, \quad (2.27)$$

donde $p_{j+k}(x)$ es el polinomio que interpola a f en el stencil de celdas $S_{j+k} = \{I_{j+k-r+1}, \dots, I_{j+k}\}$, y $p_{j+k}^{(0)}(x) = p_{j+k}(x)$.

De este modo, lo que hacemos en realidad no es más que añadir en el indicador definido por Jiang y Shu un nuevo sumando, que

corresponde al polinomio interpolador sin derivar. Las expresiones que resultan para $r = 2$ son las siguientes:

$$\begin{aligned} IS_j &= (\bar{f}_j)^2 + \frac{13}{12}(\bar{f}_j - \bar{f}_{j-1})^2 \\ IS_{j+1} &= (\bar{f}_j)^2 + \frac{13}{12}(\bar{f}_{j+1} - \bar{f}_j)^2 \end{aligned} \quad (2.28)$$

En el caso $r = 3$ se obtiene:

$$\begin{aligned} IS_j &= \frac{1}{576}(23\bar{f}_j + 2\bar{f}_{j-1} - \bar{f}_{j-2})^2 + \frac{13}{48}(3\bar{f}_j - 4\bar{f}_{j-1} + \bar{f}_{j-2})^2 \\ &+ \frac{1043}{960}(\bar{f}_j - 2\bar{f}_{j-1} + \bar{f}_{j-2})^2 \\ &+ \frac{1}{288}(23\bar{f}_j + 2\bar{f}_{j-1} - \bar{f}_{j-2})(\bar{f}_j - 2\bar{f}_{j-1} + \bar{f}_{j-2}) \\ IS_{j+1} &= \frac{1}{576}(-\bar{f}_{j+1} + 26\bar{f}_j - \bar{f}_{j-1})^2 + \frac{13}{48}(\bar{f}_{j+1} - \bar{f}_{j-1})^2 \\ &+ \frac{1043}{960}(\bar{f}_{j+1} - 2\bar{f}_j + \bar{f}_{j-1})^2 \\ &+ \frac{1}{288}(-\bar{f}_{j+1} + 26\bar{f}_j - \bar{f}_{j-1})(\bar{f}_{j+1} - 2\bar{f}_j + \bar{f}_{j-1}) \\ IS_{j+2} &= \frac{1}{576}(-\bar{f}_{j+2} + 2\bar{f}_{j+1} + 23\bar{f}_j)^2 + \frac{13}{48}(-\bar{f}_{j+2} + 4\bar{f}_{j+1} - 3\bar{f}_j)^2 \\ &+ \frac{1043}{960}(\bar{f}_{j+2} - 2\bar{f}_{j+1} + \bar{f}_j)^2 \\ &+ \frac{1}{288}(-\bar{f}_{j+2} + 2\bar{f}_{j+1} + 23\bar{f}_j)(\bar{f}_{j+2} - 2\bar{f}_{j+1} + \bar{f}_j) \end{aligned} \quad (2.29)$$

Utilizando el resultado (2.25) calculamos el desarrollo de Taylor de estas expresiones, para analizar el orden de exactitud obtenido mediante estos indicadores. En primer lugar, estudiamos el caso $r = 2$:

$$\begin{aligned} IS_{j+k} &= (\bar{f}_j)^2 + \frac{13}{12}(\bar{f}_j - \bar{f}_{j-1})^2 \\ &= \frac{1}{h^2}(F_j - F_{j-1})^2 + \frac{13}{12}(hF''_{j-\frac{1}{2}} + O(h^2))^2 \\ &= (F'_{j-\frac{1}{2}} + O(h^2))^2 + O(h^2) \\ &= (F'_{j-\frac{1}{2}})^2 + O(h^2) \\ &= (f_{j-\frac{1}{2}})^2(1 + O(h^2)). \end{aligned}$$

Para analizar el caso $r = 3$, desarrollamos por separado los sumandos que componen el indicador. Para IS_j tenemos:

$$\begin{aligned}
23\bar{f}_j + 2\bar{f}_{j-1} - \bar{f}_{j-2} &= 24\bar{f}_j - (\bar{f}_j - \bar{f}_{j-1}) + (\bar{f}_{j-1} - \bar{f}_{j-2}) \\
&= 24F'_{j-\frac{1}{2}} + O(h^2) - hF''_{j-\frac{1}{2}} + O(h^2) + hF''_{j-\frac{1}{2}} + O(h^2) \\
&= 24F'_{j-\frac{1}{2}} + O(h^2) \\
3\bar{f}_j - 4\bar{f}_{j-1} + \bar{f}_{j-2} &= 3(\bar{f}_j - \bar{f}_{j-1}) - (\bar{f}_{j-1} - \bar{f}_{j-2}) \\
&= 3(hF''_{j-\frac{1}{2}} + O(h^2)) - (hF''_{j-\frac{1}{2}} + O(h^2)) \\
&= 2hF''_{j-\frac{1}{2}} + O(h^2) \\
\bar{f}_j - 2\bar{f}_{j-1} + \bar{f}_{j-2} &= \bar{f}_j - \bar{f}_{j-1} - (\bar{f}_{j-1} - \bar{f}_{j-2}) \\
&= hF''_{j-\frac{1}{2}} + O(h^2) - hF''_{j-\frac{1}{2}} + O(h^2) \\
&= O(h^2).
\end{aligned}$$

Los sumandos de los indicadores IS_{j+1} e IS_{j+2} producen los mismos desarrollos:

$$\begin{aligned}
-\bar{f}_{j+1} + 26\bar{f}_j - \bar{f}_{j-1} &= -(\bar{f}_{j+1} - \bar{f}_j) + 24\bar{f}_j + (\bar{f}_j - \bar{f}_{j-1}) \\
&= 24F'_{j-\frac{1}{2}} + O(h^2) \\
\bar{f}_{j+1} - \bar{f}_{j-1} &= (\bar{f}_{j+1} - \bar{f}_j) + (\bar{f}_j - \bar{f}_{j-1}) \\
&= 2hF''_{j-\frac{1}{2}} + O(h^2) \\
\bar{f}_{j+1} - 2\bar{f}_j + \bar{f}_{j-1} &= O(h^2).
\end{aligned}$$

$$\begin{aligned}
-\bar{f}_{j+2} + 2\bar{f}_{j+1} + 23\bar{f}_j &= -(\bar{f}_{j+2} - \bar{f}_{j+1}) + (\bar{f}_{j+1} - \bar{f}_j) + 24\bar{f}_j \\
&= 24F'_{j-\frac{1}{2}} + O(h^2) \\
-\bar{f}_{j+2} + 4\bar{f}_{j+1} - 3\bar{f}_j &= -(\bar{f}_{j+2} - \bar{f}_{j+1}) + 3(\bar{f}_{j+1} - \bar{f}_j) \\
&= 2hF''_{j-\frac{1}{2}} + O(h^2) \\
\bar{f}_{j+2} - 2\bar{f}_{j+1} + \bar{f}_j &= O(h^2)
\end{aligned}$$

Por tanto el desarrollo de Taylor será el mismo para los tres

indicadores:

$$\begin{aligned}
 IS_{j+k} &= \frac{1}{576}(24F'_{j-\frac{1}{2}} + O(h^2))^2 + \frac{13}{48}(2hF''_{j-\frac{1}{2}} + O(h^2))^2 + \frac{1043}{960}O(h^4) \\
 &+ \frac{1}{288}(24F'_{j-\frac{1}{2}} + O(h^2))O(h^2) \\
 &= (F'_{j-\frac{1}{2}})^2 + O(h^2) \\
 &= (f_{j-\frac{1}{2}})^2(1 + O(h^2))
 \end{aligned}$$

Por tanto llegamos a la conclusión de que este indicador de suavidad también proporciona el orden óptimo de exactitud para la interpolación por medias en celda, ya que tenemos:

$$IS_{j+k} = (f'_{j-\frac{1}{2}})^2(1 + O(h^2)), \quad k = 0, 1, 2. \quad (2.30)$$

Por la Proposición 2.2:

$$\omega_k^j = C_k^j + O(h^2), \quad k = 0, 1, 2,$$

y de la Proposición 2.3 deducimos que el orden de exactitud que se consigue es el óptimo, $O(h^5)$.

Sin embargo, el resultado (2.30) muestra que este indicador de suavidad conduce a pesos que no satisfacen la propiedad ENO (2.2)-(2.3), ya que se tiene $IS_{j+k} = O(1)$, independientemente de que el stencil S_{j+k} contenga o no alguna singularidad.

3

Experimentos numéricos para medias en celda

3.1

Introducción

El objetivo de este capítulo es mostrar los resultados obtenidos mediante distintos experimentos numéricos, realizados tanto con funciones unidimensionales como con imágenes, en el contexto de interpolación de medias en celda.

En primer lugar, se analiza la compresión de señales unidimensionales. Se consideran dos funciones: una de ellas presenta una discontinuidad de salto, mientras que la otra tiene una dis-

continuidad en su primera derivada. Los datos discretos que se comprimirán son las medias en celda de ambas funciones en una partición del intervalo $[0, 1]$ formada por 1024 celdas.

Realizamos la descomposición multiescala del conjunto de datos original según el Algoritmo 1.2 de multirresolución directa con control del error visto en la Sección 1.7. La localización de los coeficientes de escala en la representación multiescala obtenida nos proporciona información sobre la presencia de singularidades en la función, ya que las singularidades se corresponden con zonas de apilamiento de estos coeficientes. Un mayor número de coeficientes de escala no nulos implica un mayor número de celdas afectado por la singularidad.

La compresión que realizamos es una compresión con pérdida de información, ya que los coeficientes de escala son procesados mediante una estrategia de truncamiento durante la descomposición. Posteriormente se realiza la reconstrucción del conjunto de datos original utilizando el algoritmo (1.19) visto en la Sección 1.3, y se compara el error obtenido en la reconstrucción con las distintas técnicas de interpolación utilizadas.

El segundo bloque de experimentos se ha realizado con tres imágenes diferentes: una imagen geométrica, una imagen real con poca textura, y una imagen real. Estos experimentos son de dos tipos. Por una parte, hemos comparado las reconstrucciones obtenidas con distintas técnicas de interpolación a partir de la descomposición multiescala obtenida aplicando el Algoritmo B.1 con distintas técnicas de interpolación como estrategia para la reconstrucción. Para analizar los resultados de las técnicas utilizadas, hemos considerado las reconstrucciones obtenidas procesando los coeficientes de escala de la descomposición mediante una estrategia de truncamiento, y las reconstrucciones obtenidas a partir de los datos discretos en un nivel de resolución más bajo, sin utilizar coeficientes de escala.

Por otra parte, hemos comparado numéricamente la estabilidad de los algoritmos de multirresolución en 2D cuando utilizamos distintas técnicas de interpolación en la reconstrucción. En estos experimentos se realiza la descomposición multiescala con y sin control del error, procesando los coeficientes de escala mediante cuantización, y se comparan los errores producidos en las

reconstrucciones con las cotas teóricas del error (1.51).

3.2

Señales

En estos experimentos analizamos la capacidad de compresión de distintas técnicas de interpolación utilizadas como operador predicción en los algoritmos de multirresolución. Identificamos cada técnica mediante los siguientes acrónimos:

- LIN - interpolación lineal centrada, en los intervalos donde el stencil centrado queda dentro del dominio computacional (Sección 1.4).

En los intervalos cercanos a la frontera, utilizamos stencils hacia la izquierda o hacia la derecha, según el caso, de forma que se utilice sólo la información disponible (se puede encontrar más información sobre esta técnica en [10] y [23]).

- ENO - interpolación ENO, con selección jerárquica del stencil (Sección 1.5).
- ENOII - interpolación ENO, con selección no jerárquica del stencil (Sección 1.5).
- SR - interpolación ENO con resolución subcelda y selección jerárquica del stencil (Sección 1.6).
- LIU - interpolación WENO con el indicador definido por Liu et al. (Sección 2.5.1).
- SHU - interpolación WENO con el indicador de suavidad definido por Jiang y Shu (Sección 2.5.2).
- MOD - interpolación WENO con el indicador de Jiang y Shu modificado (Sección 2.5.3).

A continuación se indica el número de celdas contenido en cada stencil considerado en la interpolación. En los casos de interpolación lineal, ENO y ENO-SR, utilizaremos los valores 3 y 5 para

indicar interpolación con 3 y 5 celdas. En el caso de interpolación WENO, utilizaremos los valores 2 y 3 para indicar que el interpolante se construye como combinación de polinomios construidos sobre stencils de 2 y 3 celdas, respectivamente.

Consideramos dos funciones unidimensionales: la primera presenta una discontinuidad de salto, mientras que la segunda tiene una discontinuidad en su primera derivada. Los datos discretos que se comprimirán son las medias en celda de ambas funciones en una partición del intervalo $[0, 1]$.

Con el objetivo de asegurar la estabilidad de la interpolación ENO y ENO-SR realizamos la descomposición multiescala del conjunto de datos original según el Algoritmo 1.2 de multirresolución directa con control del error (ver [1] y [9]). Los coeficientes de escala son procesados mediante el operador de truncamiento (1.27) utilizando los siguientes umbrales, que dependen de cada nivel de resolución:

$$\epsilon_k = \frac{\epsilon}{2^{L-k+1}}.$$

donde L es el número de niveles considerado. En nuestros experimentos tomaremos $J_L = 1024$ y $J_0 = 8$ (ya que el nivel de resolución más bajo debe contener suficientes datos para poder construir al menos un polinomio interpolador), es decir, $L = 7$. Además, realizaremos nuestros experimentos con dos valores distintos de ϵ , $\epsilon = 10^{-2}$ y $\epsilon = 5 \cdot 10^{-6}$.

Para analizar la capacidad de compresión de cada una de las técnicas de reconstrucción, lo que haremos será representar la localización de los coeficientes de escala que se encuentran por encima de la tolerancia dada para cada nivel de resolución. Estos coeficientes contienen la información no redundante del error de predicción en cada escala k , y están relacionados con los errores de interpolación. Cuando una función presenta una singularidad, se obtienen reconstrucciones con zonas de poca exactitud, y como consecuencia se obtienen valores grandes en los correspondientes coeficientes de escala. De este modo, podemos ver dónde se encuentran las singularidades observando los apilamientos de coeficientes de escala.

Pasamos ahora al primer grupo de experimentos, consistente en la compresión de los datos discretos asociados a una función con una discontinuidad de salto.

Función con discontinuidad de salto

La función que consideramos es la siguiente (ver Figura 3.1):

$$f(x) = \begin{cases} 0,5 \sin(\pi x), & 0 \leq x < 2/3, \\ -\sin(\pi x) + 0,5, & 2/3 \leq x \leq 1. \end{cases} \quad (3.1)$$

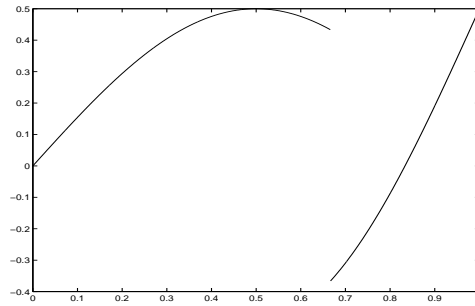


Figura 3.1: Función con discontinuidad de salto.

Consideramos como datos discretos las medias en celda de la función (3.1). La interpolación lineal basada en stencils formados por $2r - 1$ celdas produce regiones de poca exactitud formadas también por $2r - 1$ celdas, como se muestra en las Figuras 3.2(a)-(b) y 3.4(a)-(b). Además para $\epsilon = 10^{-2}$ no se aprecian grandes diferencias entre las interpolaciones de distintos órdenes. Esta diferencia es más significativa cuando reducimos la tolerancia.

En las Figuras 3.2(c)-(d) y 3.4(c)-(d) se muestran los resultados obtenidos mediante la técnica ENO. Observamos que en este caso la región afectada por la discontinuidad se reduce a un solo intervalo, y también que la interpolación de orden 5 mejora los resultados obtenidos con interpolación de orden 3.

Por otra parte, si consideramos los resultados que se consiguen utilizando la técnica ENO-SR, vemos que la exactitud es mayor que la que se obtiene con ENO, ya que una discontinuidad de salto en la función equivale a una esquina en su primitiva, y esto hace que el efecto producido por la discontinuidad resulte prácticamente eliminado, como se puede ver en las Figuras 3.2(e)-(f) y algo menos en las Figuras 3.4(e)-(f).

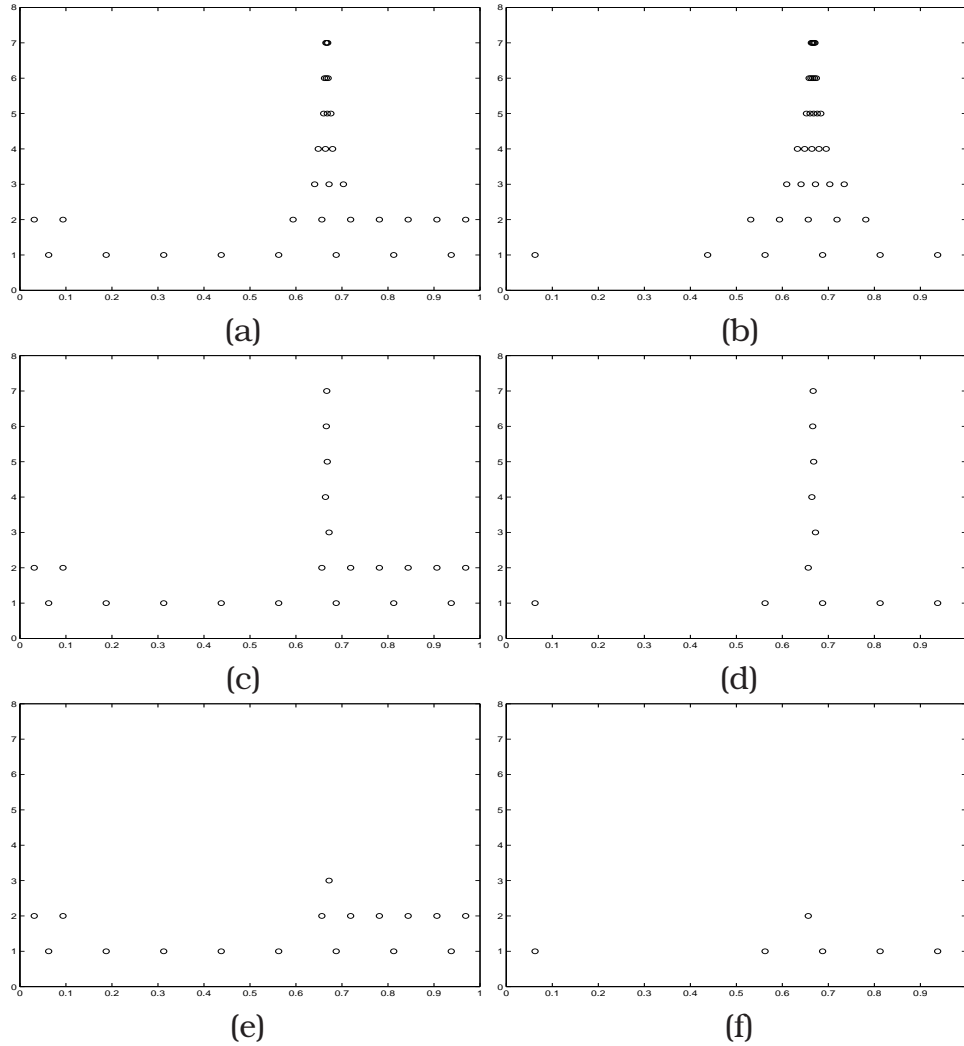


Figura 3.2: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 10^{-2}$, al aplicar la transformada directa a la función con discontinuidad de salto con distintas reconstrucciones: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) SR3, (f) SR5.

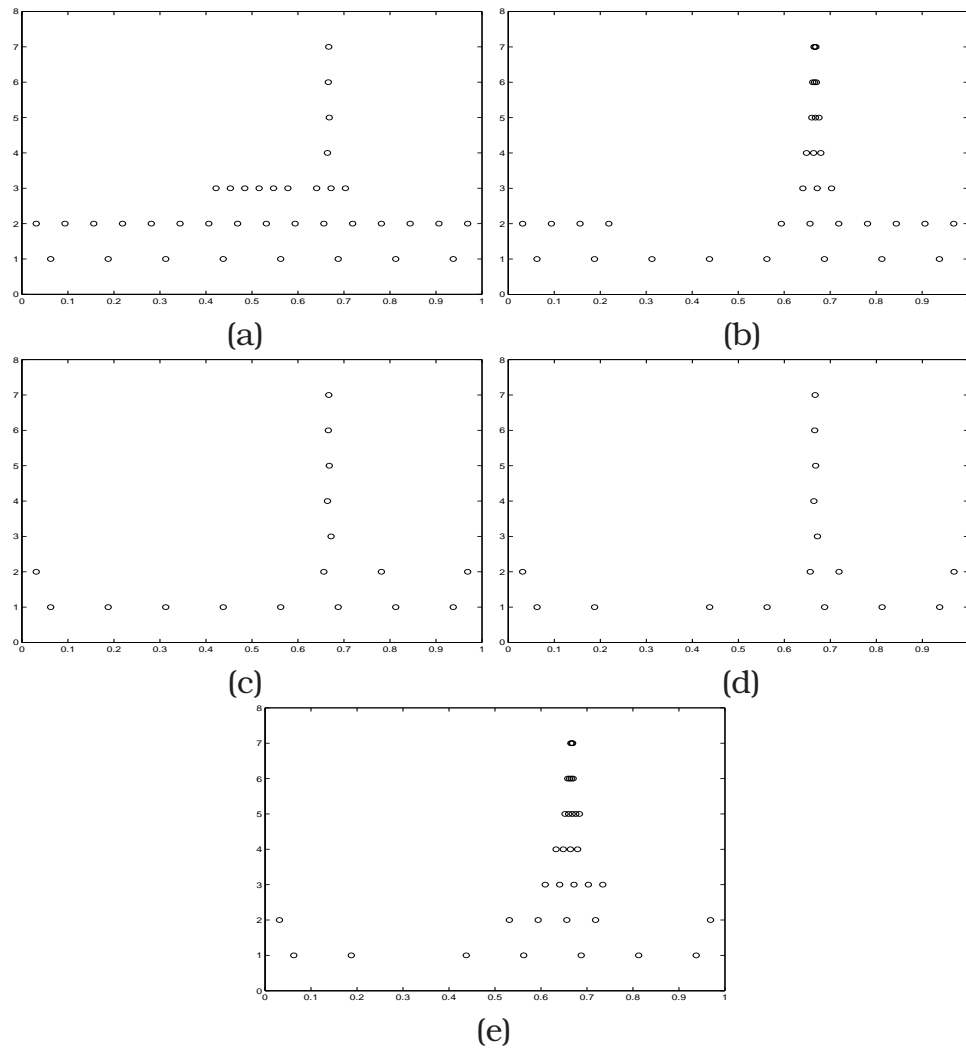


Figura 3.3: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 10^{-2}$, al aplicar la transformada directa a la función con discontinuidad de salto con distintas reconstrucciones: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3.

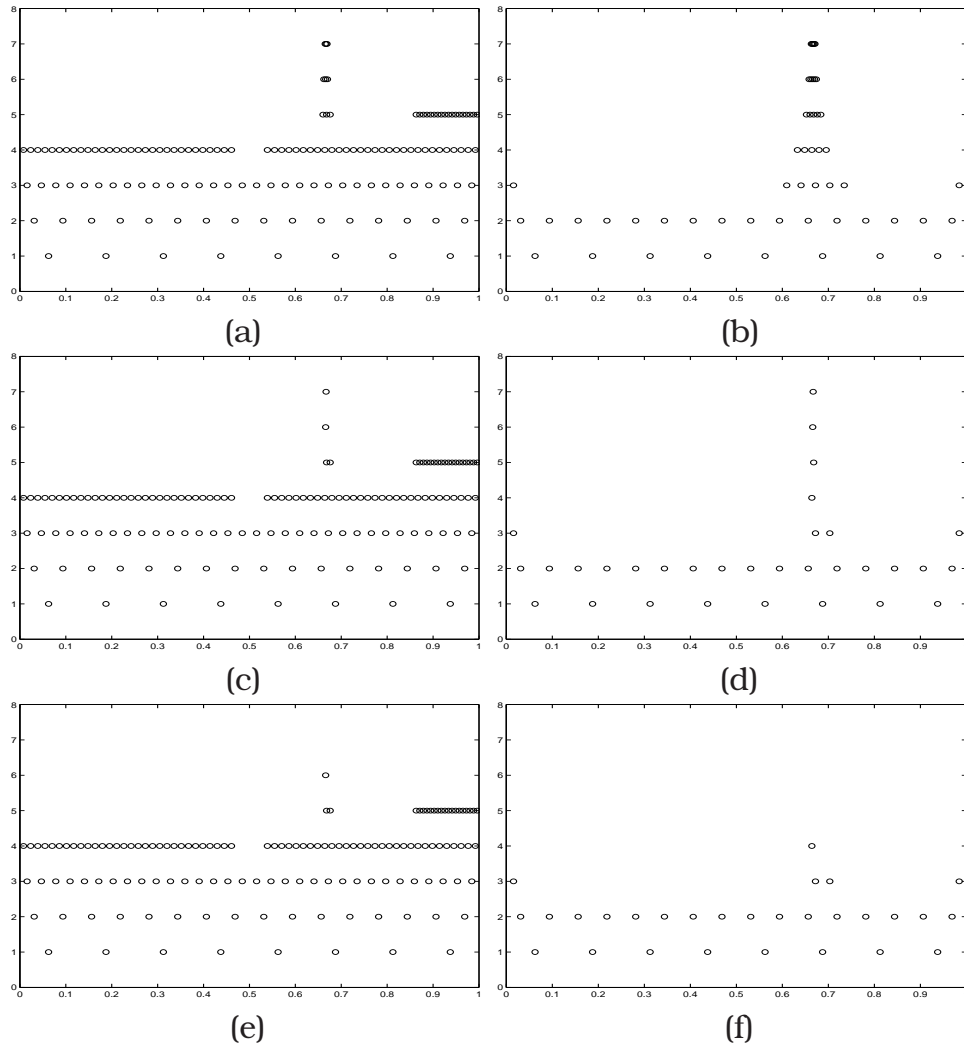


Figura 3.4: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 5 \cdot 10^{-6}$, al aplicar la transformada directa a la función con discontinuidad de salto con distintas reconstrucciones: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) SR3, (f) SR5.

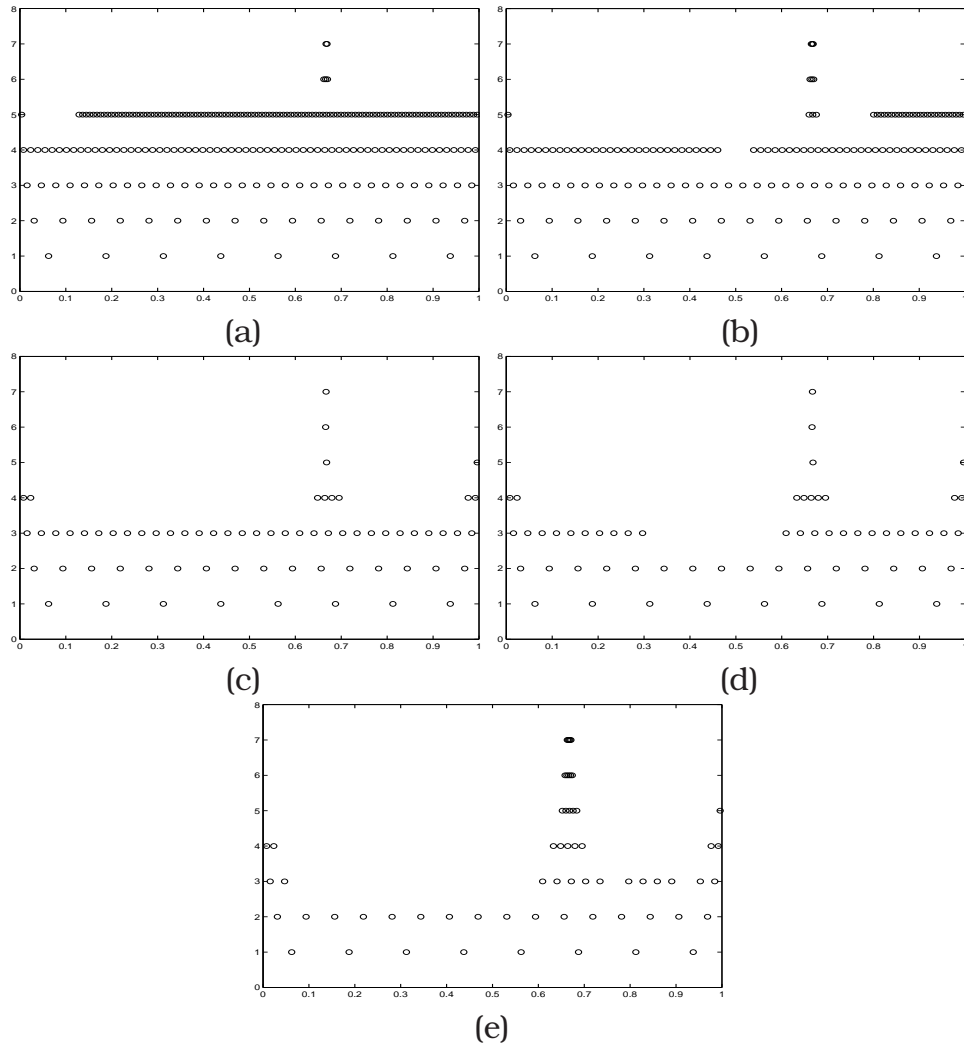


Figura 3.5: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 5 \cdot 10^{-6}$, al aplicar la transformada directa a la función con discontinuidad de salto con distintas reconstrucciones: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3.

Por último, consideramos los resultados obtenidos a partir de la técnica WENO, que se muestran en las Figuras 3.3 y 3.5. Los indicadores de suavidad definidos por Liu et al. y por Jiang y Shu coinciden para $r = 2$ en el contexto de medias en celda, por lo que sólo se incluyen los resultados obtenidos con este último. Para $r = 3$ observamos que ambos indicadores producen resultados similares, aunque ligeramente mejores cuando utilizamos el indicador de Jiang y Shu. Vemos también que en este caso el interpolante WENO mejora los resultados obtenidos con la reconstrucción ENO de orden 3, pero no llega a igualar los resultados obtenidos con la de orden 5. Por otra parte, la técnica ENO-SR proporciona mejores resultados que WENO, debido a que la región afectada por la discontinuidad es casi inexistente.

En cuanto a los resultados obtenidos utilizando el indicador modificado MOD (2.27), se observa un comportamiento similar al producido con la interpolación lineal. Recordemos que este indicador no satisface la propiedad ENO (2.2)-(2.3), y por tanto la discontinuidad produce zonas de mala exactitud que afectan a $2r - 1$ celdas, cuando se consideran r stencils formados por r celdas.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz
LIN3	2.2e-4	5.1e-5	7.4e-5	32	1.5e-6	2.4e-7	3.0e-7	143
ENO3	4.0e-4	5.8e-5	8.6e-5	21	1.5e-6	2.5e-7	3.5e-7	138
SR3	5.3e-4	5.9e-5	8.7e-5	17	1.5e-6	2.5e-7	3.5e-7	137
SHU2	5.0e-4	1.1e-4	1.4e-4	37	2.0e-6	3.4e-7	4.1e-7	238
MOD2	2.3e-4	4.7e-5	6.5e-5	34	8.3e-7	2.1e-7	2.6e-7	153
LIN5	9.5e-5	1.2e-5	2.1e-5	36	3.0e-7	3.7e-8	5.3e-8	51
ENO5	1.4e-4	2.2e-5	2.9e-5	11	3.0e-7	4.5e-8	6.2e-8	32
SR5	3.9e-4	2.5e-5	3.9e-5	6	4.8e-7	4.7e-8	6.8e-8	29
LIU3	2.1e-4	3.1e-5	3.9e-5	19	1.5e-6	9.3e-8	1.5e-7	68
SHU3	2.2e-4	4.9e-5	6.9e-5	16	1.5e-6	3.8e-8	9.2e-8	60
MOD3	2.2e-3	4.8e-5	1.6e-4	34	1.5e-6	3.6e-8	8.6e-8	62

Tabla 3.1: Errores y coeficientes distintos de cero (nnz) para la función con discontinuidad de salto, $\epsilon = 10^{-2}$ (izquierda) y $\epsilon = 5 \cdot 10^{-6}$ (derecha).

Para acabar con el análisis de la discontinuidad de salto, en la Tabla 3.1 se muestran de forma resumida los resultados obtenidos

con las distintas técnicas de reconstrucción, para los dos valores de la tolerancia considerados, $\epsilon = 10^{-2}$ y $\epsilon = 5 \cdot 10^{-6}$. Para cada uno de estos valores se indica el valor del error de la reconstrucción evaluado en las normas $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$, así como el número de coeficientes de escala distintos de cero (*nnz*) que quedan después de aplicar la estrategia de truncamiento.

Función con esquina

Consideramos ahora para nuestros experimentos una función con una discontinuidad de salto en su primera derivada. Esta función se muestra en la Figura 3.6 y su expresión viene dada por:

$$f(x) = \left| \sin \left(\frac{3}{2}\pi \left(x - \frac{1}{3} \right) \right) \right| \quad (3.2)$$

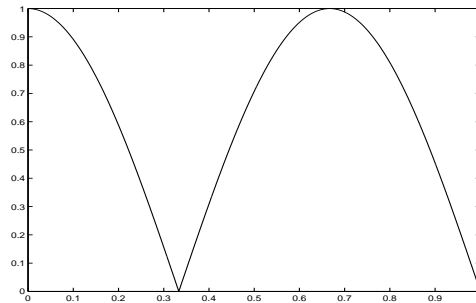


Figura 3.6: *Función con esquina.*

La función (3.2) presenta una esquina en $x = 1/3$.

Observamos en las Figuras 3.7(a)-(b) y 3.9(a)-(b) que la interpolación lineal centrada produce regiones de poca exactitud, aunque para $\epsilon = 10^{-2}$ el efecto de la singularidad es más suave que el del salto. Esto es debido a que en presencia de una discontinuidad de salto $d_j^k = \mathcal{O}(1)$ y cuando la primera derivada de la función presenta una discontinuidad se tiene $d_j^k = \mathcal{O}(h)$ (Sección 1.4).

Cuando utilizamos la técnica ENO, el proceso de selección jerárquica del stencil no consigue reconocer la singularidad, y hace que más de un intervalo resulte afectado por ella en cada nivel de

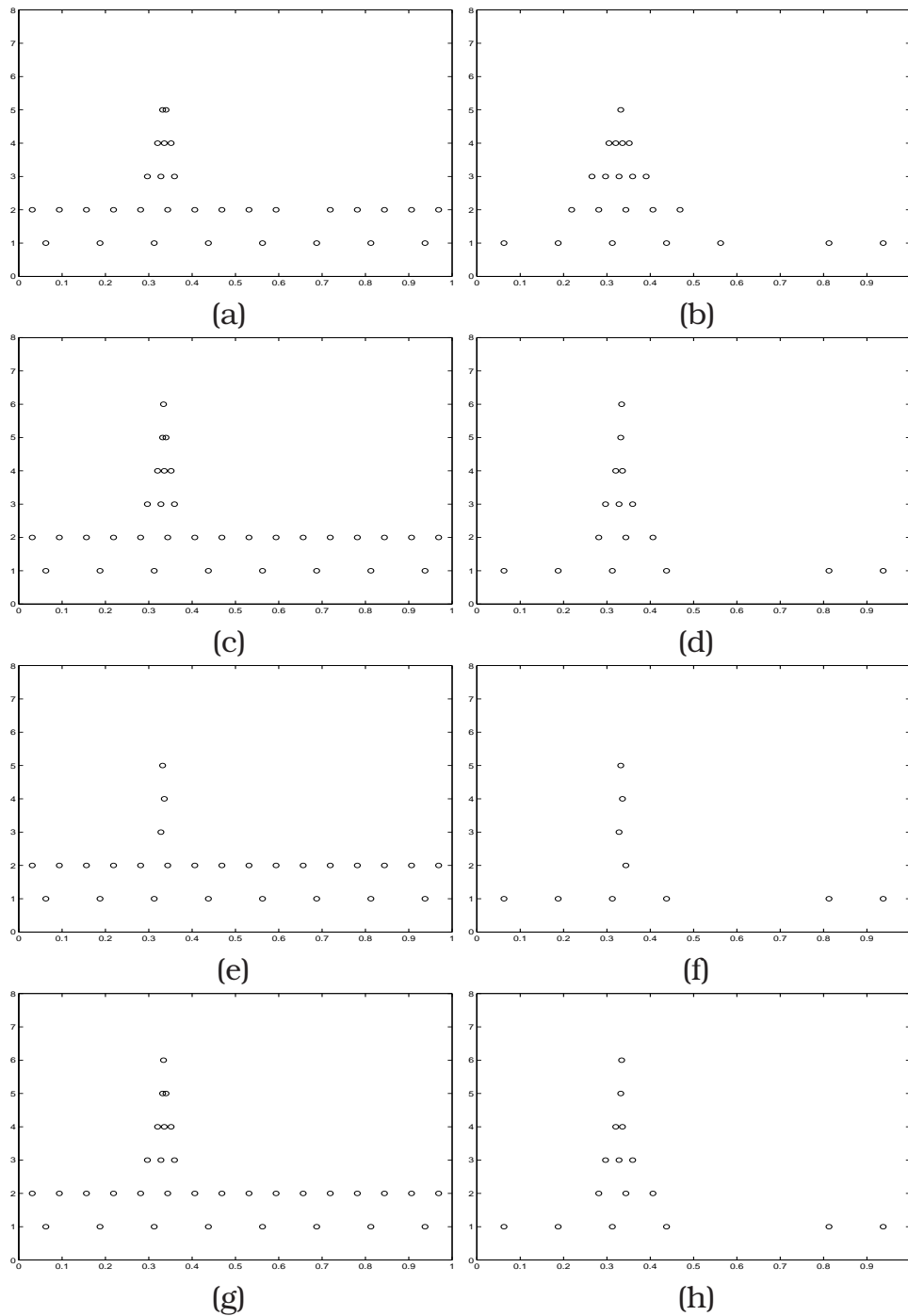


Figura 3.7: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 10^{-2}$, al aplicar la transformada directa a la función con esquina con distintas reconstrucciones: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) ENOII3, (f) ENOII5, (g) SR3, (h) SR5.

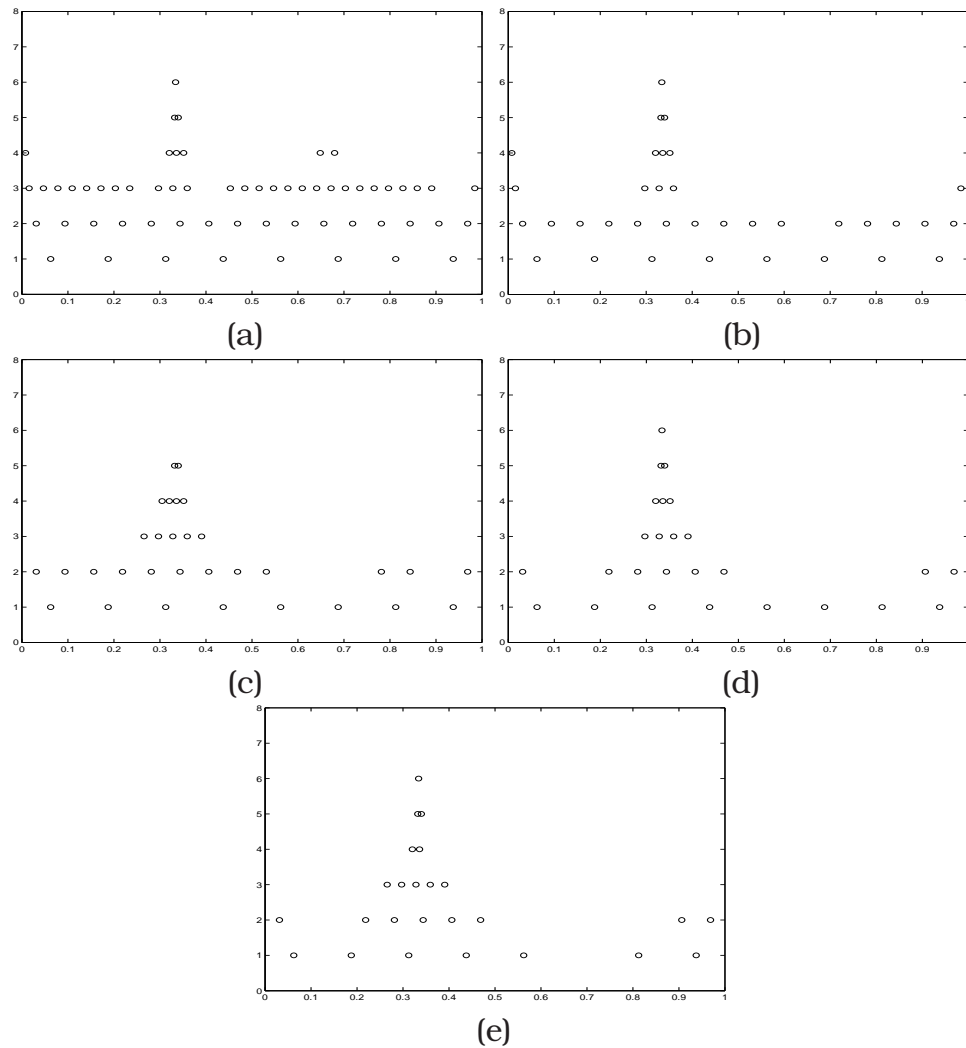


Figura 3.8: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 10^{-2}$, al aplicar la transformada directa a la función con esquina con distintas reconstrucciones: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3.

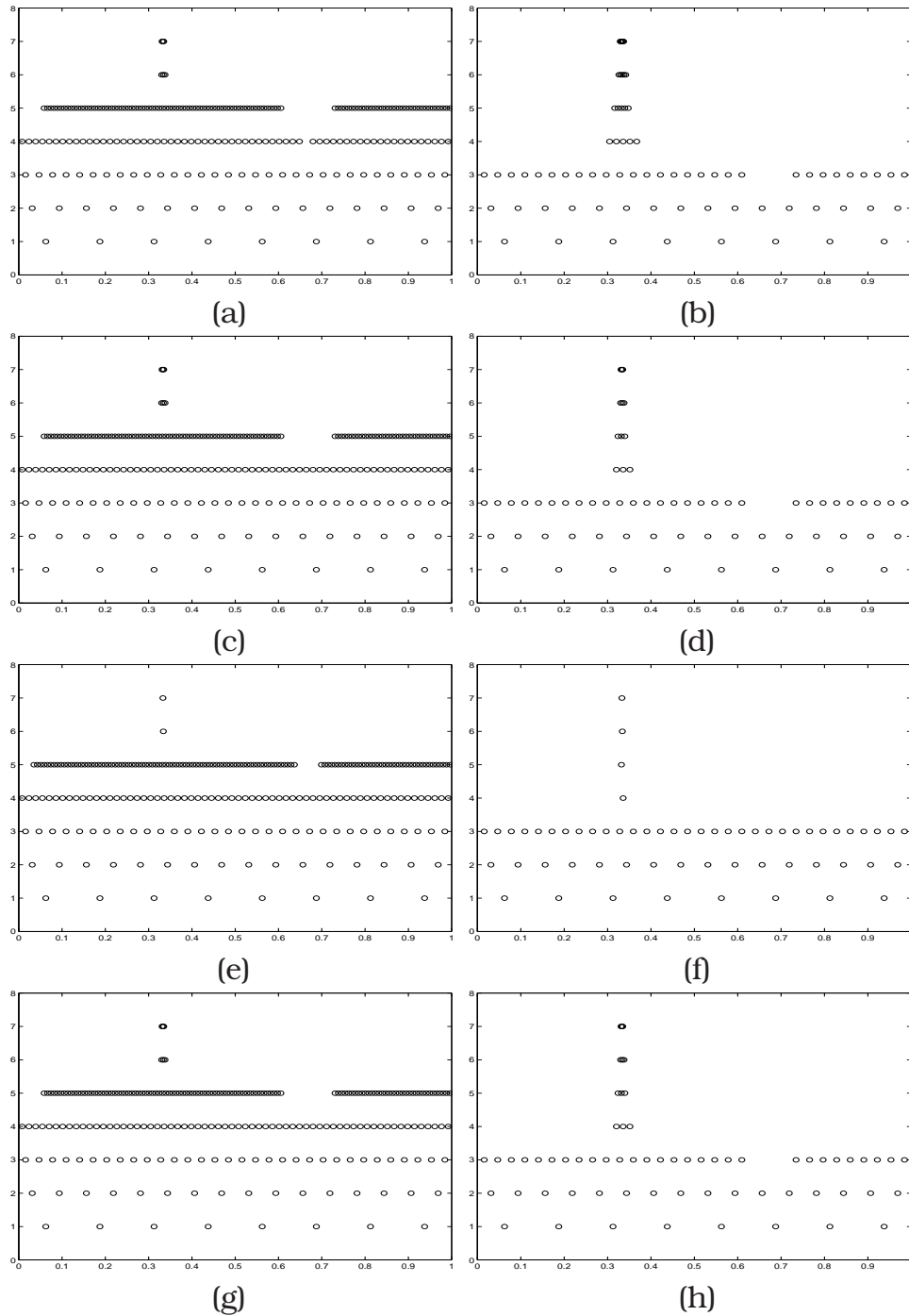


Figura 3.9: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 5 \cdot 10^{-6}$, al aplicar la transformada directa a la función con esquina con distintas reconstrucciones: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) ENOII3, (f) ENOII5, (g) SR3, (h) SR5.

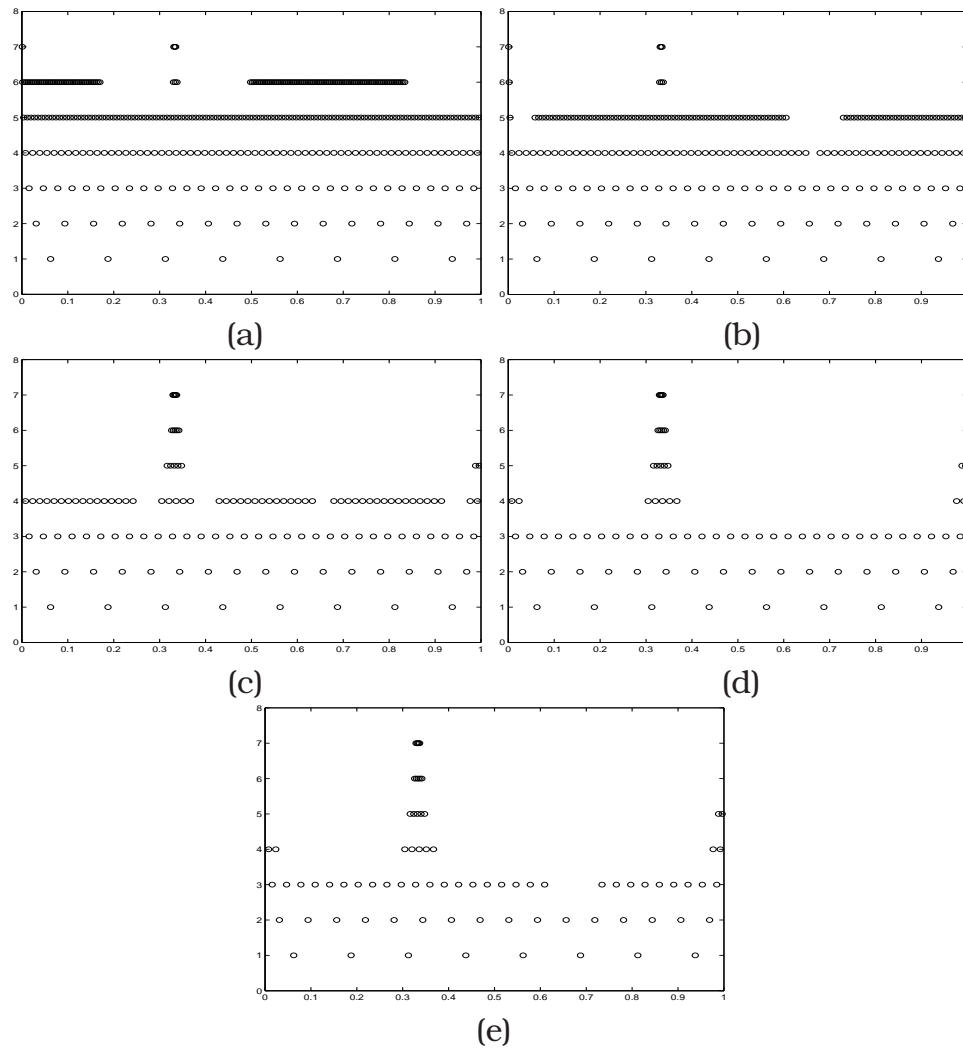


Figura 3.10: Localización de los detalles d_j^k que se encuentran por encima de la tolerancia ϵ_k para cada nivel de resolución, con $\epsilon = 5 \cdot 10^{-6}$, al aplicar la transformada directa a la función con esquina con distintas reconstrucciones: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3.

resolución. Esto no ocurre si utilizamos la selección no jerárquica del stencil, como se puede ver en las Figuras 3.7(e)-(f) y 3.9(e)-(f).

La técnica ENO-SR basada en la selección jerárquica del stencil produce un efecto idéntico al de la técnica ENO, con lo que se obtienen los mismos resultados (ver Figuras 3.7(g)-(h) y 3.9(g)-(h)).

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz
LIN3	2.5e-3	1.1e-4	1.8e-4	31	9.4e-7	2.3e-7	2.6e-7	231
ENO3	2.0e-3	1.0e-4	1.8e-4	33	1.3e-6	2.3e-7	3.0e-7	232
ENOII3	4.8e-3	1.9e-4	3.0e-4	27	1.8e-6	3.6e-7	4.7e-7	239
SR3	2.0e-3	1.0e-4	1.8e-4	33	1.3e-6	2.3e-7	3.0e-7	232
SHU2	2.7e-3	1.9e-4	3.1e-4	60	1.5e-6	5.1e-7	5.8e-7	386
MOD2	2.3e-3	1.1e-4	1.9e-4	35	1.1e-6	2.3e-7	2.6e-7	236
LIN5	2.0e-3	5.3e-5	1.6e-4	22	3.1e-7	2.1e-8	3.5e-8	73
ENO5	1.9e-3	5.1e-5	1.5e-4	16	3.1e-7	2.5e-8	4.2e-8	65
ENOII5	4.4e-3	5.9e-5	1.7e-4	10	6.0e-7	9.0e-8	1.3e-7	60
SR5	1.9e-3	5.1e-5	1.5e-4	16	3.1e-7	2.5e-8	4.2e-8	65
LIU3	1.7e-3	7.0e-5	1.5e-4	31	9.6e-7	8.5e-8	1.3e-7	126
SHU3	1.7e-3	7.1e-5	1.4e-4	26	9.6e-7	8.0e-8	1.1e-7	82
MOD3	1.7e-3	5.7e-5	1.5e-4	25	9.6e-7	2.5e-8	5.4e-8	79

Tabla 3.2: Errores y coeficientes distintos de cero (nnz) para la función con esquina, $\epsilon = 10^{-2}$ (izquierda) y $\epsilon = 5 \cdot 10^{-6}$ (derecha).

Finalmente, con la técnica WENO (Figuras 3.8 y 3.10) con los indicadores de Liu et al. y Jiang y Shu la singularidad provoca el mismo efecto que se observaba con la técnica ENO seleccionando el stencil de forma jerárquica. Los resultados obtenidos con ambos indicadores utilizando 3 celdas para la interpolación mejoran claramente los resultados obtenidos con la reconstrucción ENO de orden 3, y en el caso del indicador definido por Jiang y Shu (Figura 3.10(c)) son bastante parecidos, cuando la tolerancia es pequeña, a los obtenidos mediante la reconstrucción ENO de quinto orden, aunque ligeramente peores.

Con el indicador modificado MOD (2.27) los resultados son prácticamente idénticos a los obtenidos con la interpolación lineal. Como ocurría en el caso de la función con discontinuidad de salto se producen zonas de mala exactitud que afectan a $2r - 1$ celdas,

cuando se consideran r stencils formados por r celdas, ya que este indicador no emula el comportamiento de la técnica ENO.

Por último, en la Tabla 3.2 se resumen los resultados anteriores, junto con los errores en las reconstrucciones obtenidas con las distintas técnicas de interpolación.

3.3

Imágenes

3.3.1

Comparación entre reconstrucciones

En estos experimentos consideramos tres imágenes de distintos tipos:

- Una imagen geométrica sintética, formada por paralelogramos, triángulos y círculos (Figura 3.11).
- Una imagen real con poca textura, de forma que resulta en cierto modo geométrica: la imagen pimientos (Figura 3.17).
- Una imagen real: la imagen Lena (Figura 3.23).

En estos experimentos hemos comparado las reconstrucciones obtenidas con distintas técnicas de interpolación a partir de la descomposición multiescala que resulta aplicando el Algoritmo B.1 de multirresolución directa con control del error para medias en celda en 2D, con distintas técnicas de interpolación como estrategia para la reconstrucción. Este algoritmo garantiza a priori una cierta calidad de la imagen reconstruida (Proposición 1.4 y Corolario 1.2).

Para identificar las distintas técnicas de reconstrucción utilizaremos acrónimos similares a los utilizados en la Sección 3.2:

- LIN - interpolación lineal centrada, con la modificación en la frontera ya explicada (Sección 1.4).
- ENO - interpolación ENO, con selección jerárquica del stencil (Sección 1.5).

- SR - interpolación ENO con resolución subcelda y selección jerárquica del stencil (Sección 1.6).
- LIU - interpolación WENO con el indicador de suavidad definido por Liu et al. (Sección 2.5.1).
- SHU - interpolación WENO con el indicador de suavidad definido por Jiang y Shu (Sección 2.5.2).
- MOD - interpolación WENO con el indicador de Jiang y Shu modificado (Sección 2.5.3).

Junto con el acrónimo de la técnica utilizada indicamos con un número el total de celdas contenido en el stencil de interpolación. Para la interpolación lineal, ENO y ENO-SR utilizaremos los valores 3 y 5 para indicar interpolación con 3 y 5 celdas. En el caso de interpolación WENO, utilizaremos los valores 2 y 3 para indicar que el interpolante se construye como combinación de polinomios construidos sobre stencils de 2 y 3 celdas, respectivamente.

Consideraremos dos tipos de pruebas. En el primer grupo de experimentos, obtendremos la descomposición multiescala, y los coeficientes de escala correspondientes a cada nivel de resolución serán procesados mediante una estrategia de truncamiento (1.27) con los siguientes umbrales, que dependen del nivel de resolución:

$$\epsilon_k = \frac{\epsilon}{2^{L-k}}.$$

En estos experimentos tomaremos como valor inicial de la tolerancia $\epsilon = 45$. Además, las imágenes que consideraremos tienen tamaño $N_L \times N_L$ con $N_L = 512$ y $L = 4$, por lo que la resolución en el nivel de resolución más bajo será $N_0 = 32$.

La reconstrucción de la imagen original se obtendrá mediante el Algoritmo B.2 a partir de su representación en el nivel más bajo de resolución, \bar{f}^0 , que se utilizará sin aplicarle ninguna estrategia de truncamiento, es decir, $\hat{f}^0 = \bar{f}^0$. Para comparar la calidad de las reconstrucciones obtenidas con las distintas técnicas de interpolación utilizaremos el número de detalles no nulos almacenados en la descomposición multirresolutiva y los errores en la reconstrucción, medidos en $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$.

En el segundo grupo de experimentos, consideraremos la descomposición multiescala hasta una resolución $N_0 = 128$, por lo que tomaremos $L = 2$, con $N_L = 512$. En este caso la reconstrucción se hará sin utilizar los coeficientes de escala, y compararemos las reconstrucciones obtenidas utilizando los errores de reconstrucción, medidos en $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$.

Geométrica

Comenzamos nuestros experimentos con una imagen geométrica sintética (Figura 3.11):

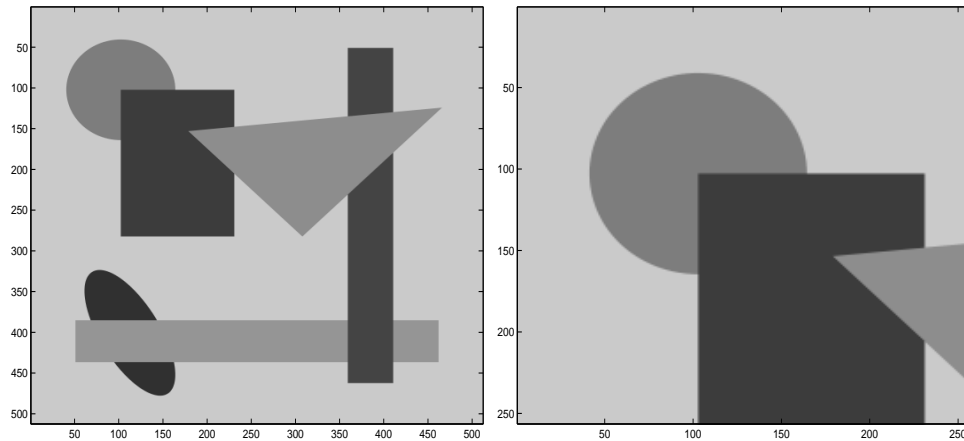


Figura 3.11: Imagen geométrica y detalle de la esquina superior izquierda.

Las reconstrucciones obtenidas utilizando los coeficientes de escala truncados se muestran en las Figuras 3.12-3.13. Para poder apreciar mejor las diferencias en las reconstrucciones correspondientes a cada técnica de interpolación, se muestra sólo la esquina superior izquierda de la imagen reconstruida. Recordemos que los indicadores de suavidad definidos por Liu et al. y por Jiang y Shu coinciden para $r = 2$ en el contexto de medias en celda, por lo que sólo se incluyen los resultados obtenidos con este último.

En la reconstrucción obtenida mediante interpolación lineal se observa un emborronamiento en los contornos de la imagen, que

desaparece cuando utilizamos técnicas no lineales de interpolación.

La Tabla 3.3 contiene los errores en las distintas reconstrucciones, medidos en las normas $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$, así como el número de coeficientes de escala distintos de cero (nnz) en la descomposición multirresolución después de aplicar la estrategia de truncamiento con $\epsilon = 45$.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz
LIN3	42.3263	1.5000	3.5746	3041
ENO3	53.2406	0.5485	2.6499	2261
SR3	56.4709	0.2875	1.8367	722
SHU2	54.3151	0.4974	2.7875	2244
MOD2	45.5844	1.4446	3.4348	2572
LIN5	48.9393	1.8534	3.6795	3070
ENO5	60.0824	0.8629	2.7751	2368
SR5	54.9925	0.7324	2.5018	1298
LIU3	53.2386	0.5296	2.6318	2261
SHU3	53.2555	0.5225	2.5502	2244
MOD3	45.8815	1.6336	3.5064	2567

Tabla 3.3: Imagen geométrica. Errores y coeficientes distintos de cero (nnz) (reconstrucción con detalles utilizando Algoritmo B.2); $\epsilon = 45$.

Por último, en la Figura 3.14 se representa el error de la reconstrucción medido en $\|\cdot\|_2$ frente al número de detalles no nulos utilizados para obtener dicha reconstrucción. Consideramos todas las técnicas de interpolación basadas en stencils formados por 3 celdas.

Se observa un comportamiento similar en las reconstrucciones obtenidas mediante interpolación lineal y WENO con indicador modificado MOD. Con MOD se consiguen errores menores que con interpolación lineal cuando se reconstruye la imagen original con menos detalles, pero cuando se reduce el error de la reconstrucción el número de detalles requerido para la reconstrucción es mucho mayor con MOD que con interpolación lineal. Hay una diferencia clara entre estas dos técnicas y el resto de técnicas no lineales.

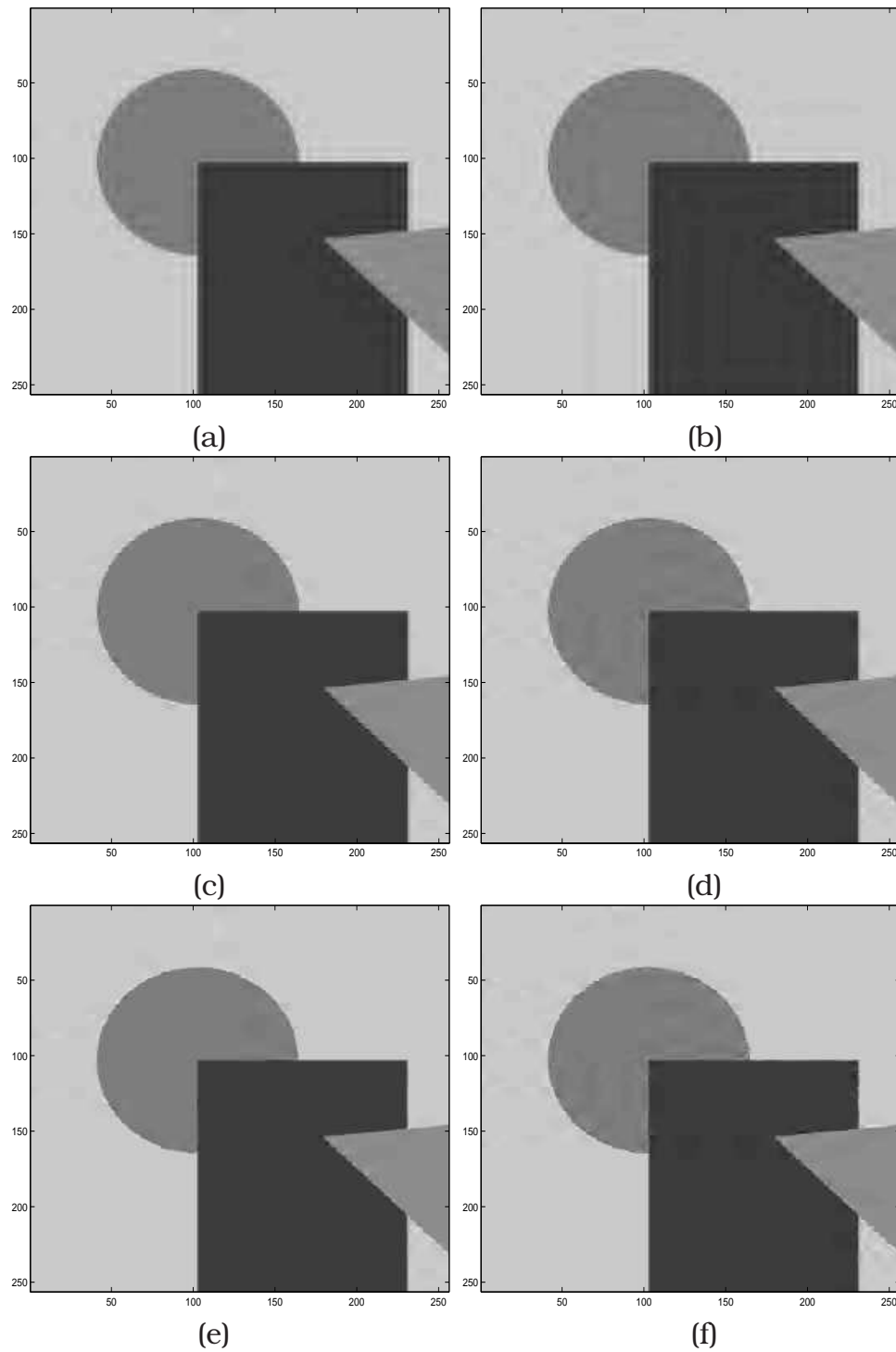


Figura 3.12: Imagen geométrica. Reconstrucción utilizando detalles con Algoritmo B.2: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) SR3, (f) SR5; $\epsilon = 45$.

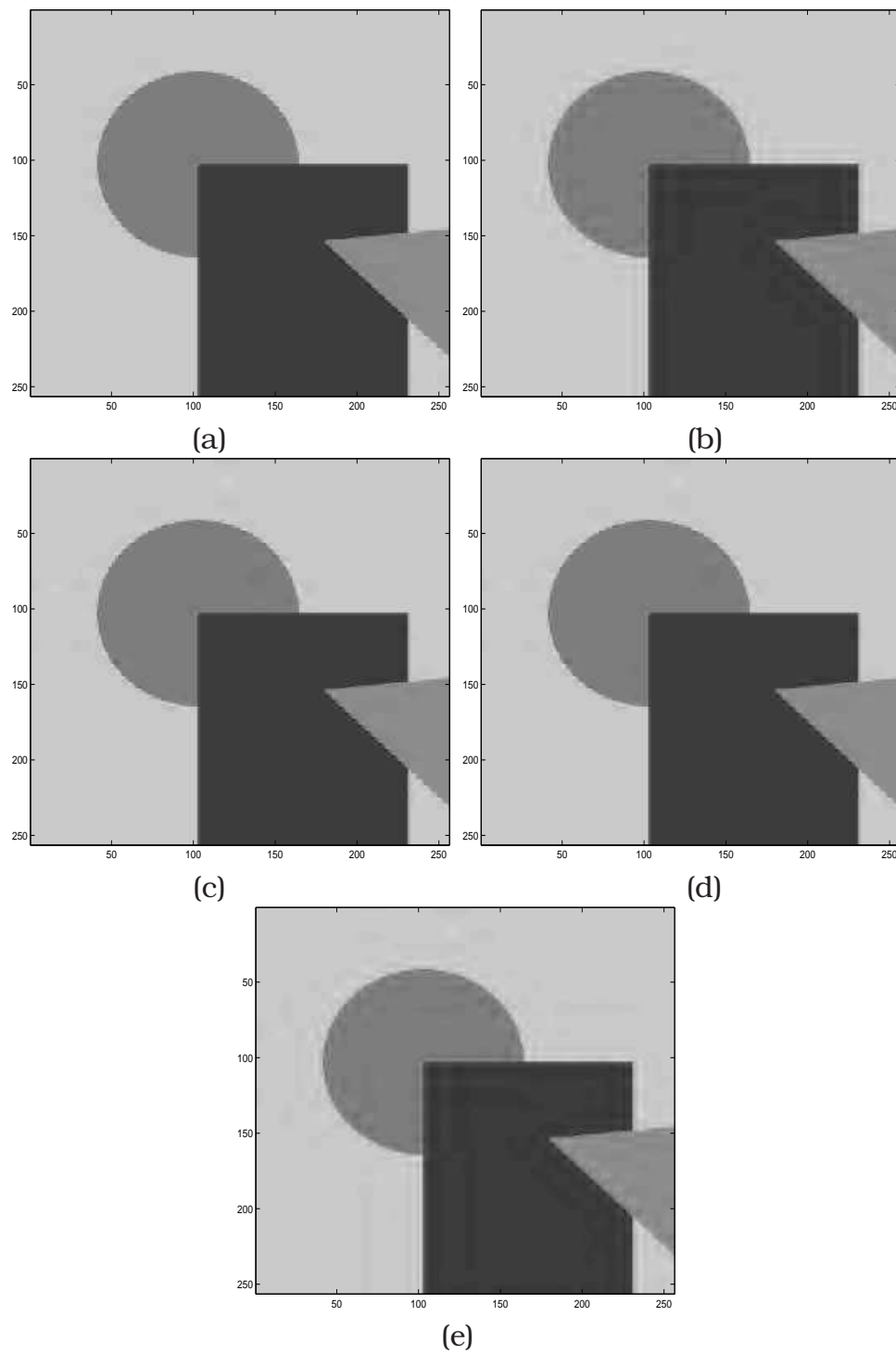


Figura 3.13: Imagen geométrica. Reconstrucción utilizando detalles con Algoritmo B.2: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3; $\epsilon = 45$.

Dentro del grupo de técnicas de interpolación con comportamiento ENO, los errores obtenidos con las técnicas ENO, LIU y SHU para un mismo número de detalles son similares, aunque ligeramente inferiores en el caso de SHU. La técnica ENO-SR resulta claramente superior a todas las demás.

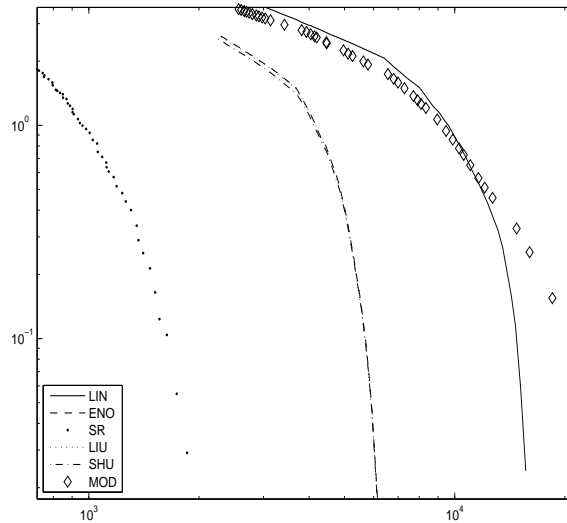


Figura 3.14: Imagen geométrica. Gráfico log-log de norma L^2 del error de reconstrucción vs. número de detalles no nulos (nnz).

Las reconstrucciones obtenidas a partir de la representación de la imagen geométrica con resolución $N_0 = 128$ sin utilizar los coeficientes de escala se muestran en las Figuras 3.15-3.16.

La interpolación lineal produce el mismo emborronamiento de los contornos que se observaba en las reconstrucciones con detalles. Este emborronamiento queda atenuado en las reconstrucciones obtenidas mediante la técnica ENO, y en los contornos rectos desaparece por completo cuando se utiliza la técnica ENO-SR. Con las técnicas LIU y SHU los resultados que se consiguen son mejores que los obtenidos con ENO, pero son superados por la técnica ENO-SR. La reconstrucción obtenida mediante la técnica WENO con indicador modificado muestra un emborronamiento en los contornos similar al de la técnica lineal.

En la Tabla 3.4 se muestra el error en las reconstrucciones

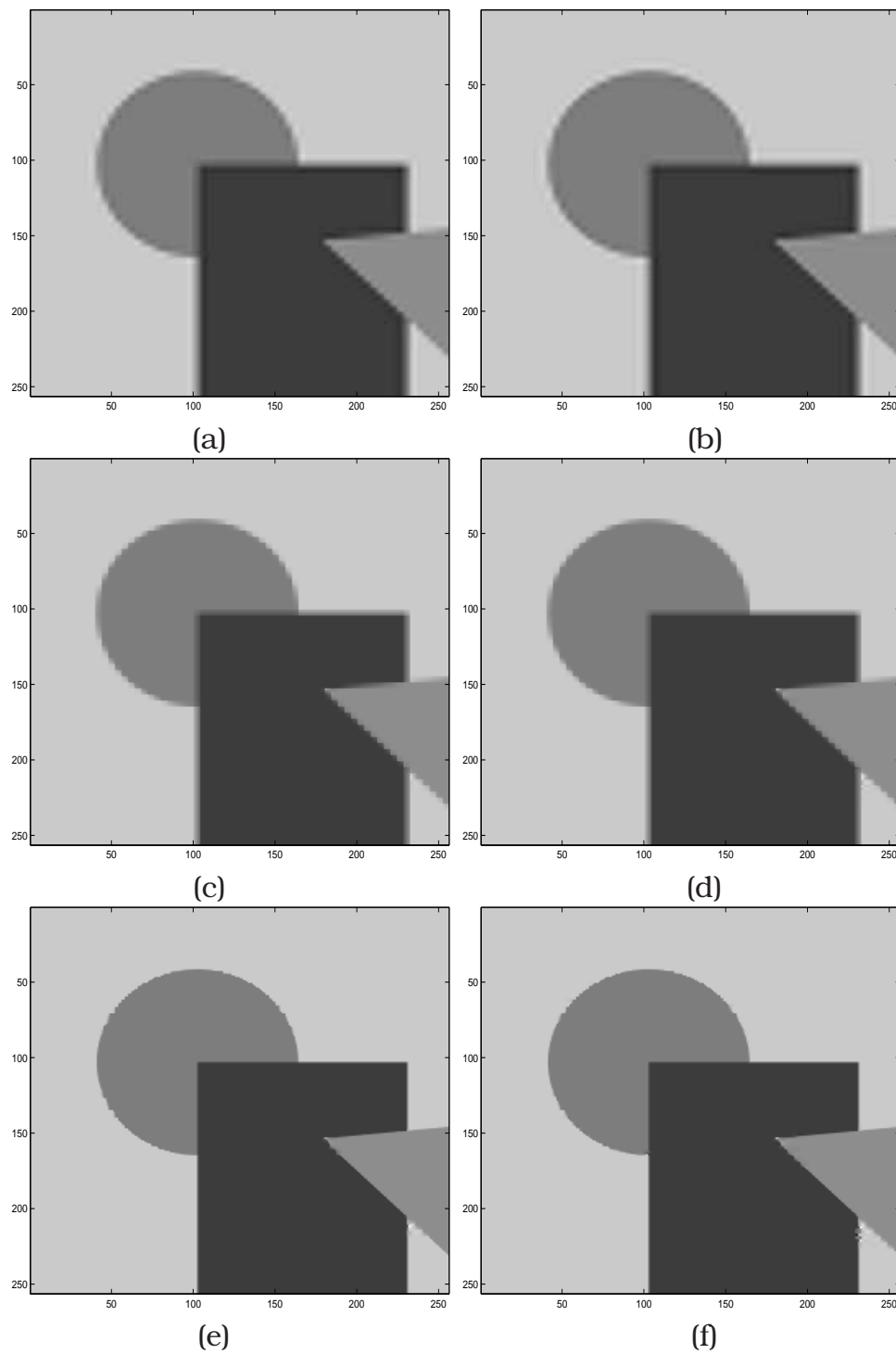


Figura 3.15: Imagen geométrica. Reconstrucción sin utilizar detalles: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) SR3, (f) SR5.

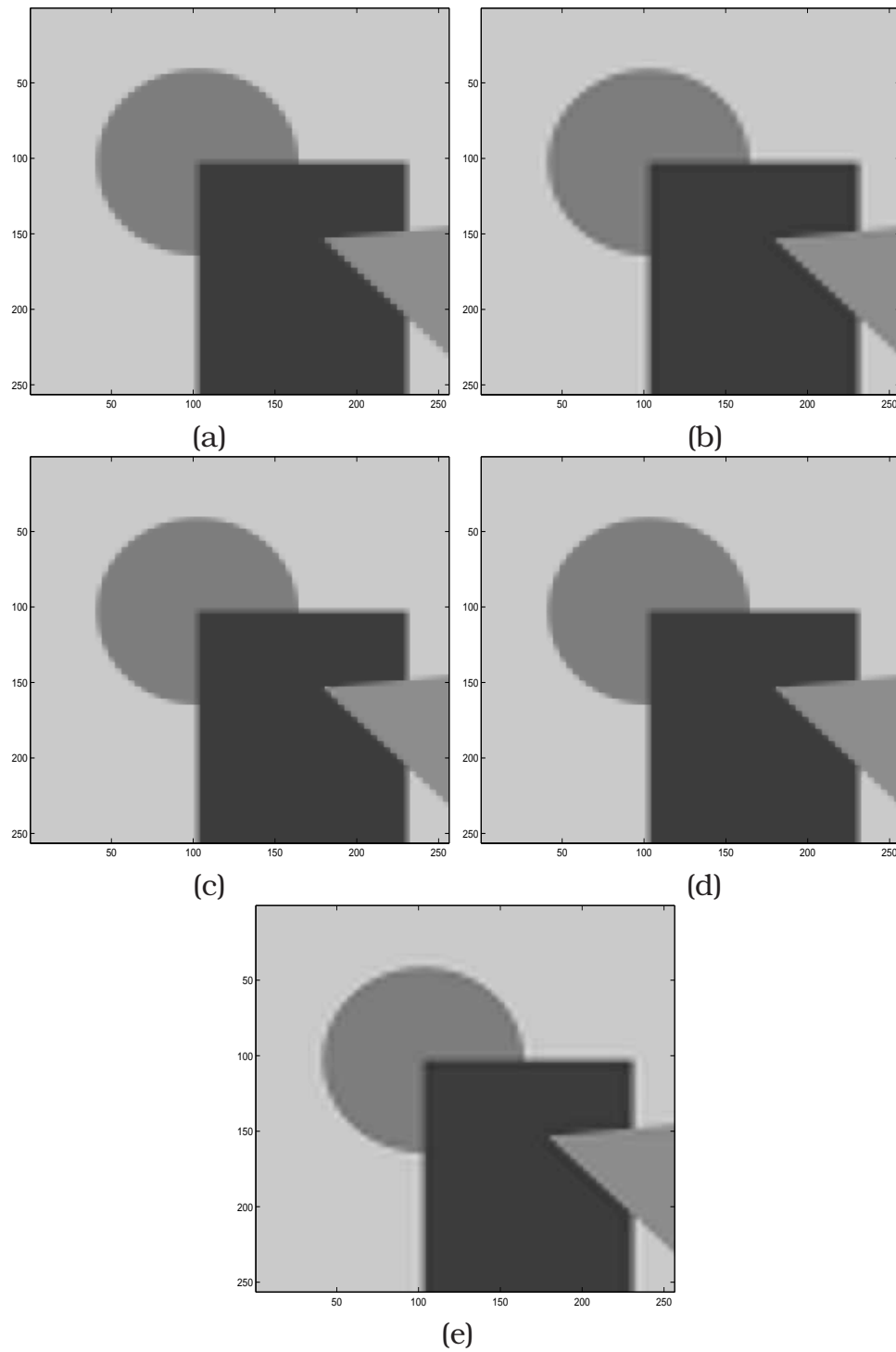


Figura 3.16: Imagen geométrica. Reconstrucción sin utilizar detalles: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3.

calculadas sin utilizar los detalles, medido en las normas $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$
LIN3	88.8244	1.6727	6.5617
ENO3	131.0604	1.1171	6.3849
SR3	172.9115	0.1107	2.1480
SHU2	125.3936	1.1959	6.7732
MOD2	108.1380	1.4928	6.3879
LIN5	83.0968	1.7075	6.2501
ENO5	163.0635	1.0322	5.8862
SR5	199.0896	0.1468	2.3291
LIU3	112.3902	1.0841	6.1569
SHU3	110.8214	1.0945	6.2212
MOD3	93.6822	1.5253	6.0963

Tabla 3.4: Imagen geométrica. Errores de la reconstrucción sin detalles.

Pimientos

Consideramos ahora la imagen pimientos (Figura 3.17).

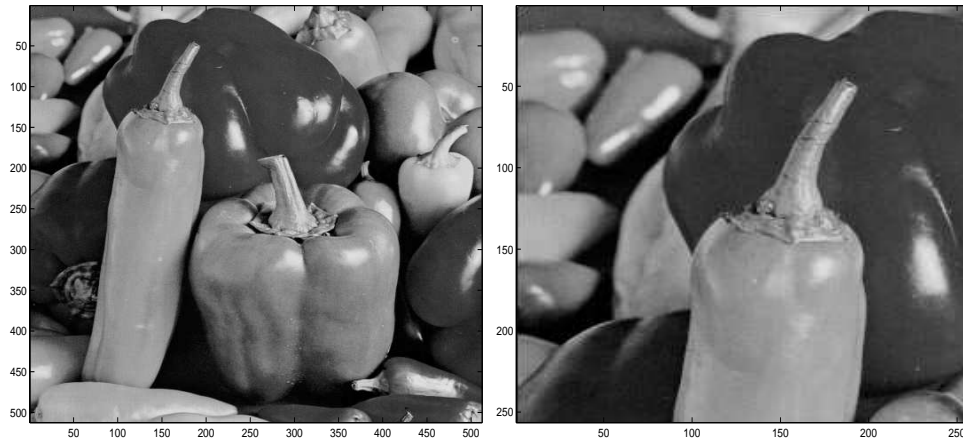


Figura 3.17: Pimientos y detalle de la esquina superior izquierda.

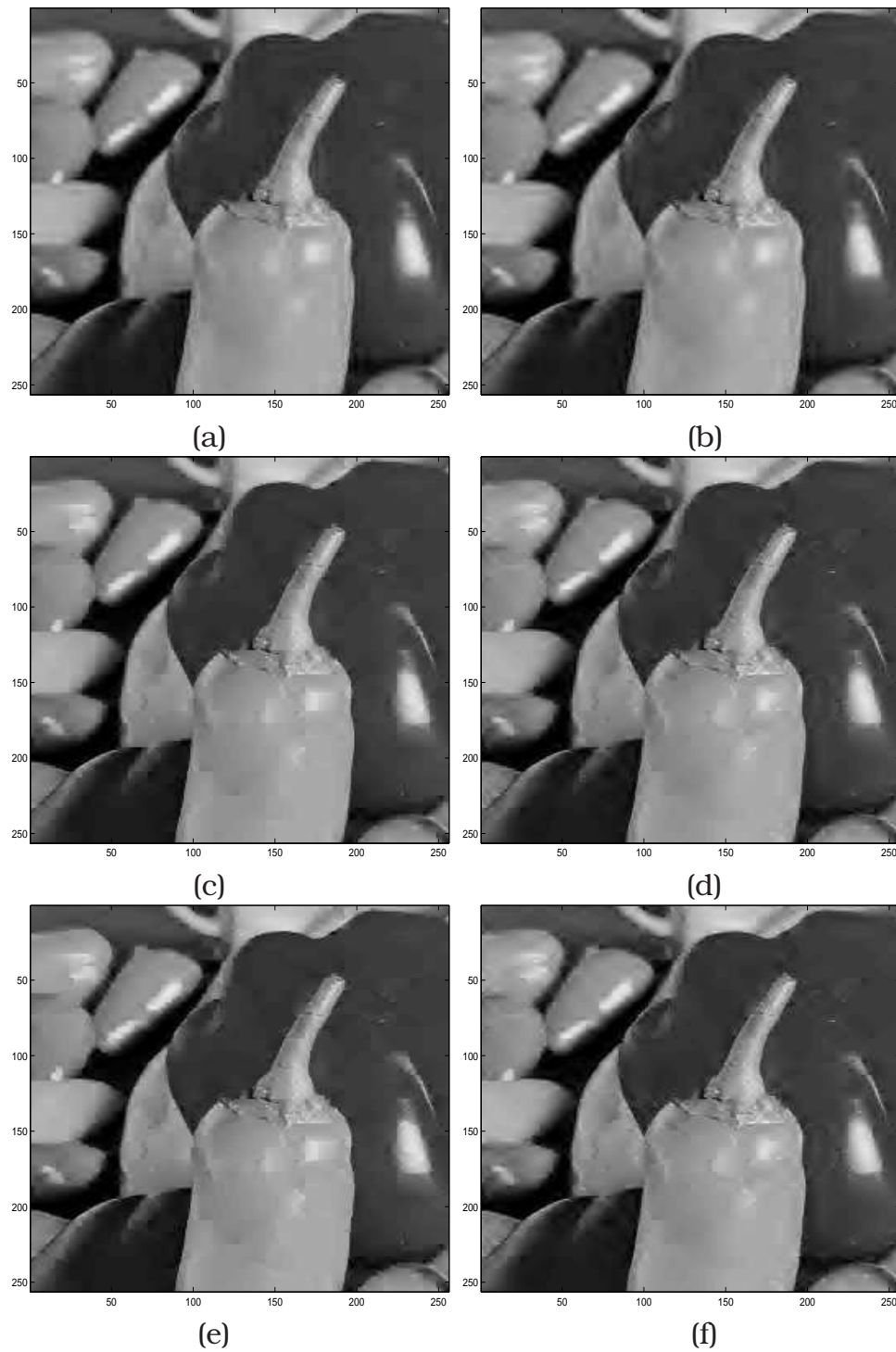


Figura 3.18: Pimientos. Reconstrucción utilizando detalles con el Algoritmo B.2: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) SR3, (f) SR5; $\epsilon = 45$.

Las Figuras 3.18-3.19 muestran las reconstrucciones obtenidas mediante las distintas técnicas de interpolación, a partir de la representación de la imagen pimientos con resolución $N_0 = 32$, utilizando los detalles de la descomposición multiescala.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz
LIN3	60.6383	4.8494	6.6527	6396
ENO3	58.7068	4.8860	6.8662	6846
SR3	65.1362	4.9590	7.0083	7197
SHU2	63.3685	4.9094	6.8885	7203
MOD2	60.7441	4.7478	6.5812	6176
LIN5	58.8383	4.8914	6.6777	6437
ENO5	62.6460	5.0857	7.0851	7177
SR5	62.6463	5.1025	7.1140	7651
LIU3	62.5883	4.8274	6.7639	6529
SHU3	61.9316	4.8045	6.7142	6412
MOD3	59.6006	4.7646	6.5642	5923

Tabla 3.5: Pimientos. Errores y elementos distintos de cero (nnz) (reconstrucción con detalles Algoritmo B.2), $\epsilon = 45$.

La Tabla 3.5 contiene los errores de las distintas reconstrucciones, medidos en las normas $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$, así como la cantidad de elementos distintos de cero (nnz) de la transformada de la multirresolución después de truncar con $\epsilon = 45$.

Visualmente no se aprecian grandes diferencias entre las distintas reconstrucciones, aunque se observa un ligero emborronamiento en los contornos, producido cuando se utiliza interpolación lineal e interpolación MOD. Desde el punto de vista numérico, es esta última técnica de interpolación la que produce un menor error en norma L^2 , con menor número de detalles.

Para finalizar estos experimentos, se muestra en la Figura 3.20 la comparación entre las distintas técnicas interpolatorias basadas en stencils formados por 3 celdas, representando el error de la reconstrucción, medido en $\|\cdot\|_2$, frente al número de detalles utilizados en la reconstrucción. A diferencia de lo que ocurría con la imagen geométrica, con la imagen pimientos el comportamiento de todas las reconstrucciones es similar, aunque la interpolación

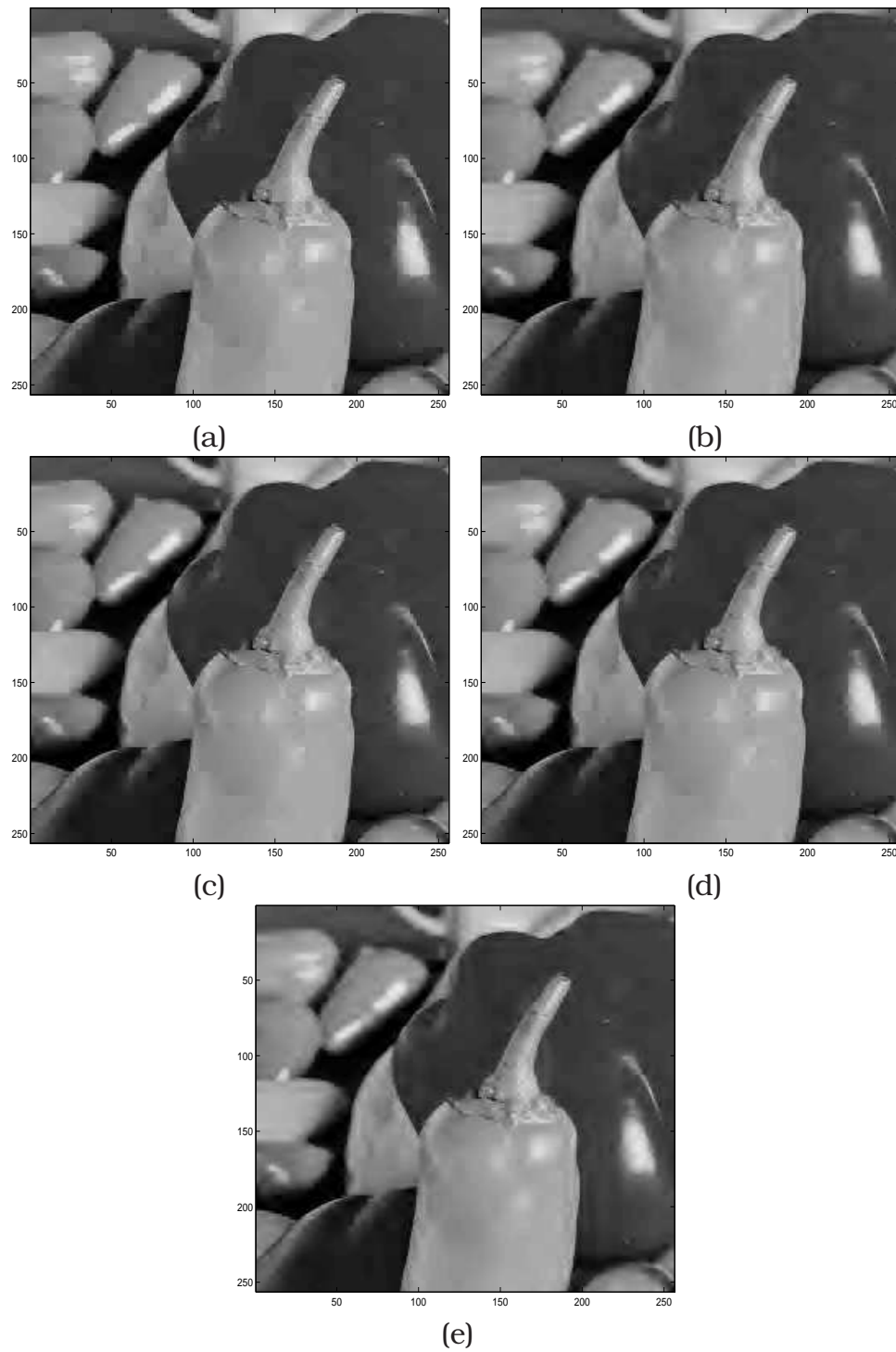


Figura 3.19: Pimientos. Reconstrucción utilizando detalles con el Algoritmo B.2: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3; $\epsilon = 45$.

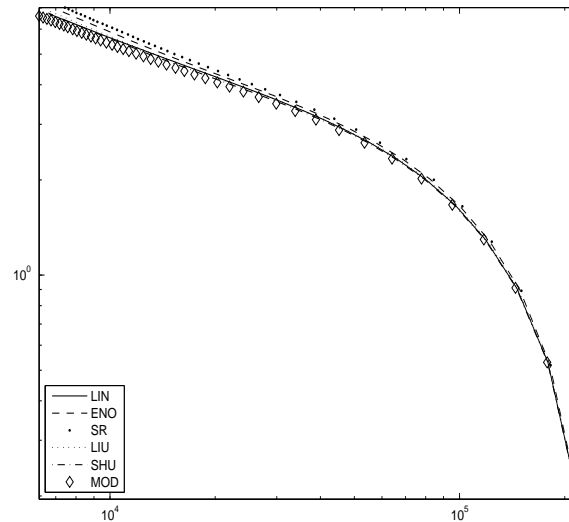


Figura 3.20: Imagen pimientos. Gráfico log-log de norma L^2 del error de reconstrucción vs. número de detalles no nulos (nnz).

WENO con indicador modificado proporciona resultados un poco mejores en general. Sin embargo, para valores pequeños del error todas las técnicas dan resultados prácticamente idénticos.

Las Figuras 3.21-3.22 muestran las distintas reconstrucciones obtenidas a partir de la representación de la imagen pimientos con resolución $N_0 = 128$, sin utilizar los detalles de la descomposición multiescala.

En la Tabla 3.6 se muestran los errores correspondientes a todas las reconstrucciones mostradas en las Figuras 3.21-3.22.

La reconstrucción obtenida mediante la técnica ENO-SR produce unos contornos más definidos, pero con un efecto de “escalera” que hace que el error aumente, como se ve en la Tabla 3.6. Como ocurría con la reconstrucción con detalles el menor error viene dado por la reconstrucción WENO con el indicador modificado, aunque la diferencia con la técnica de interpolación lineal es menor que en el caso de la reconstrucción con detalles.

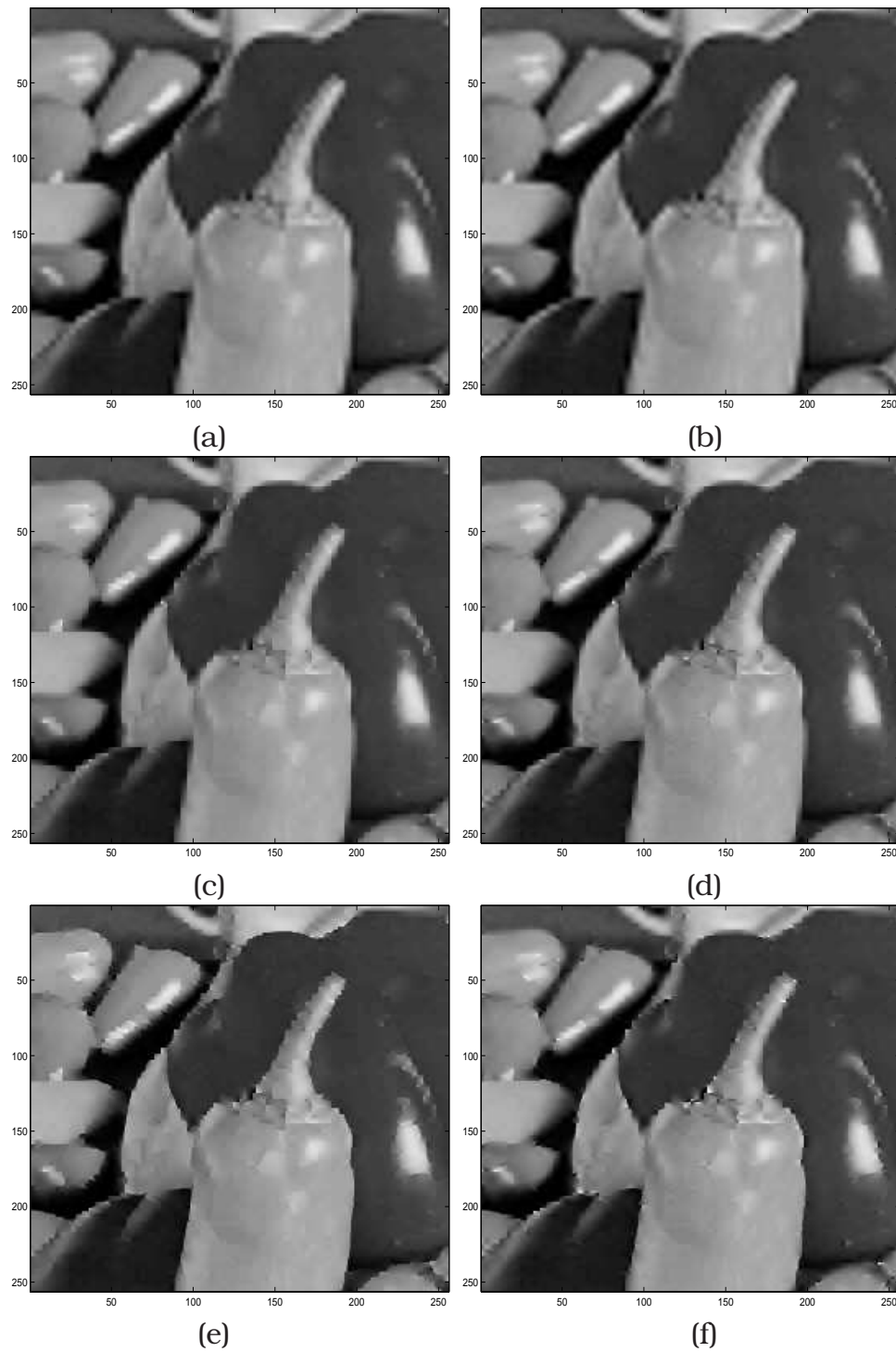


Figura 3.21: Imagen pimientos. Reconstrucción sin utilizar detalles: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) SR3, (f) SR5.

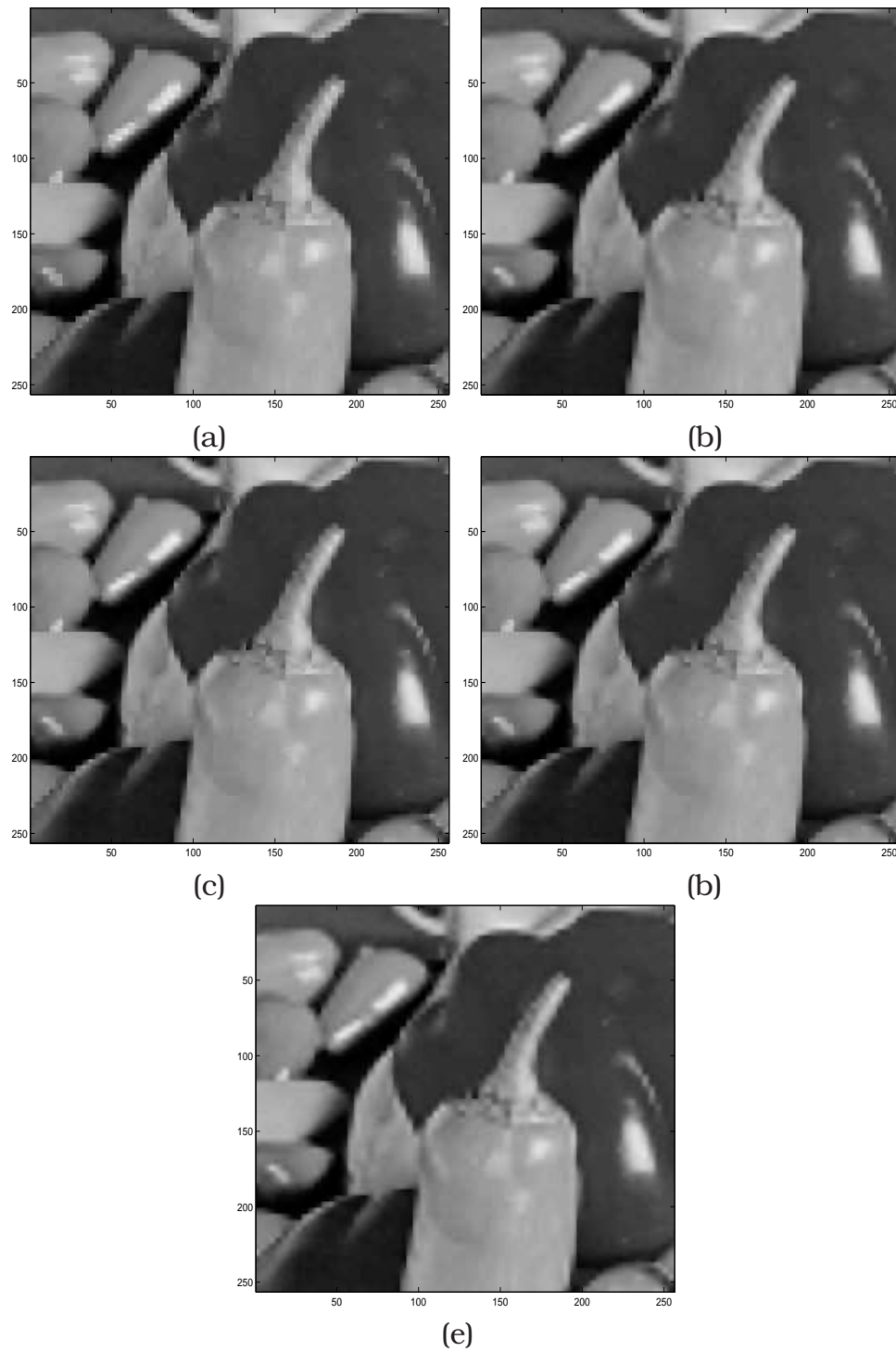


Figura 3.22: Imagen pimientos. Reconstrucción sin utilizar detalles: (a) SHU2, (a) MOD2, (c) LIU3, (d) SHU3, (e) MOD3.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$
LIN3	110.9541	5.0495	8.6230
ENO3	165.2893	5.1565	9.3013
SR3	169.0302	5.2391	9.5577
SHU2	144.0673	5.3290	9.6427
MOD2	139.1723	5.0037	8.8659
LIN5	140.2137	5.0747	8.6050
ENO5	208.4160	5.2459	9.4672
SR5	381.3199	5.3901	10.2841
LIU3	147.3120	5.0395	9.0154
SHU3	147.5007	4.9941	8.9060
MOD3	139.9533	4.8692	8.4497

Tabla 3.6: Imagen pimientos. Errores de la reconstrucción sin detalles.

Lena

Para finalizar con esta sección, consideramos ahora la imagen de Lena, que se muestra en la Figura 3.23.



Figura 3.23: Lena y detalle de la esquina inferior derecha.

Las Figuras 3.24-3.25 muestran las reconstrucciones de la imagen de Lena correspondientes a las diferentes técnicas de interpo-

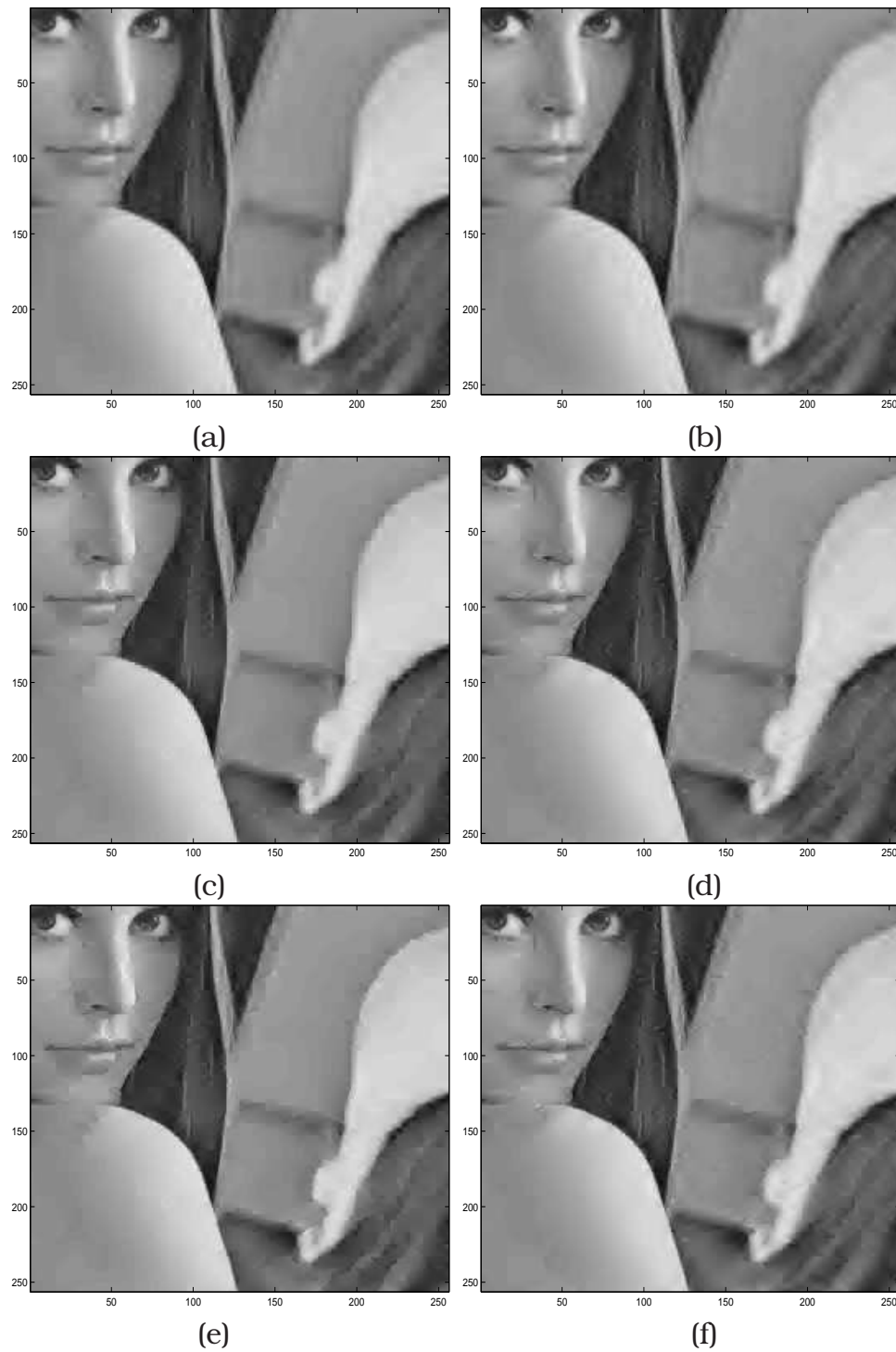


Figura 3.24: Imagen Lena. Reconstrucción utilizando detalles (Algoritmo B.2): (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5 (e) SR3, (f) SR5; $\epsilon = 45$.

lación que consideramos, partiendo de la representación con resolución $N_0 = 32$ y utilizando los coeficientes de escala.

Como ocurre con la imagen pimientos, las diferencias entre las reconstrucciones no se pueden apreciar visualmente, por lo que necesitamos recurrir a un análisis numérico de los errores para determinar qué reconstrucción es mejor.

En la Tabla 3.7 se muestran los errores de las distintas reconstrucciones, medidos en las normas $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$, junto con el número de detalles utilizado en la reconstrucción.

Se observa que de nuevo los menores errores corresponden a la técnica de interpolación WENO con indicador modificado, que además es la técnica que conserva un menor número de coeficientes de escala para la reconstrucción, por lo que desde el punto de vista de capacidad de compresión, resulta la mejor elección, aunque con resultados muy similares a los obtenidos con la interpolación lineal.

De entre las técnicas con comportamiento ENO, vemos que los menores errores corresponden a la técnica SHU, que realiza la reconstrucción con un número de detalles menor que las técnicas ENO, ENO-SR y LIU.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz
LIN3	48.8837	4.6951	6.6613	7018
ENO3	58.7125	4.9392	7.1549	8111
SR3	57.1683	4.9902	7.2731	9106
SHU2	56.9493	4.9613	7.1592	8248
MOD2	52.7003	4.6778	6.6663	7082
LIN5	49.8505	4.7104	6.6295	6926
ENO5	60.1738	4.9881	7.1932	8541
SR5	60.1738	4.9874	7.2053	9201
LIU3	58.6238	4.8485	7.0246	7639
SHU3	53.7337	4.7897	6.9176	7356
MOD3	58.3731	4.6356	6.5897	6734

Tabla 3.7: Imagen Lena. Errores y elementos distintos de cero (nnz) (reconstrucción con detalles, Algoritmo B.2), $\epsilon = 45$.

Por último, en la Figura 3.26 se comparan los errores, en nor-



Figura 3.25: Imagen Lena. Reconstrucción utilizando detalles (Algoritmo B.2): (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3 (e) MOD3; $\epsilon = 45$.

ma L^2 , correspondientes a las técnicas de interpolación basadas en stencils formados por 3 celdas, representándolos respecto al número de coeficientes de escala utilizados para la reconstrucción. La situación es similar a la que teníamos con la imagen pimientos, pero en este caso la técnica de interpolación lineal es en todo momento mejor que las técnicas no lineales.

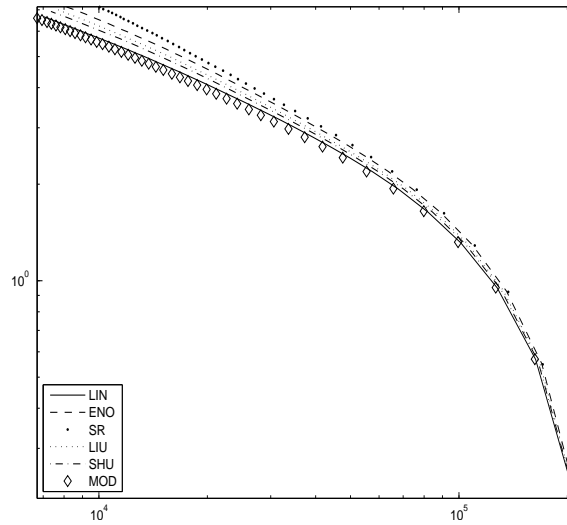


Figura 3.26: Imagen Lena. Gráfico log-log de norma L^2 del error de reconstrucción vs. número de detalles no nulos.

Los últimos experimentos que realizamos consisten en la reconstrucción de la imagen Lena, a partir de su representación con resolución $N_0 = 128$, sin utilizar los detalles de la descomposición multiescala. Estas reconstrucciones se muestran en las Figuras 3.27-3.28.

Como las diferencias no son visualmente apreciables, recurrimos a la Tabla 3.8 para analizar las distintas reconstrucciones.

Se observa que a diferencia de lo que ocurría en las reconstrucciones que utilizaban los detalles, la interpolación lineal proporciona los mejores resultados, aunque no muestra grandes diferencias respecto a la interpolación MOD. Sin embargo, si comparamos las distintas técnicas ENO entre sí, la técnica SHU resulta ser la mejor opción, con un error significativamente menor que el obtenido

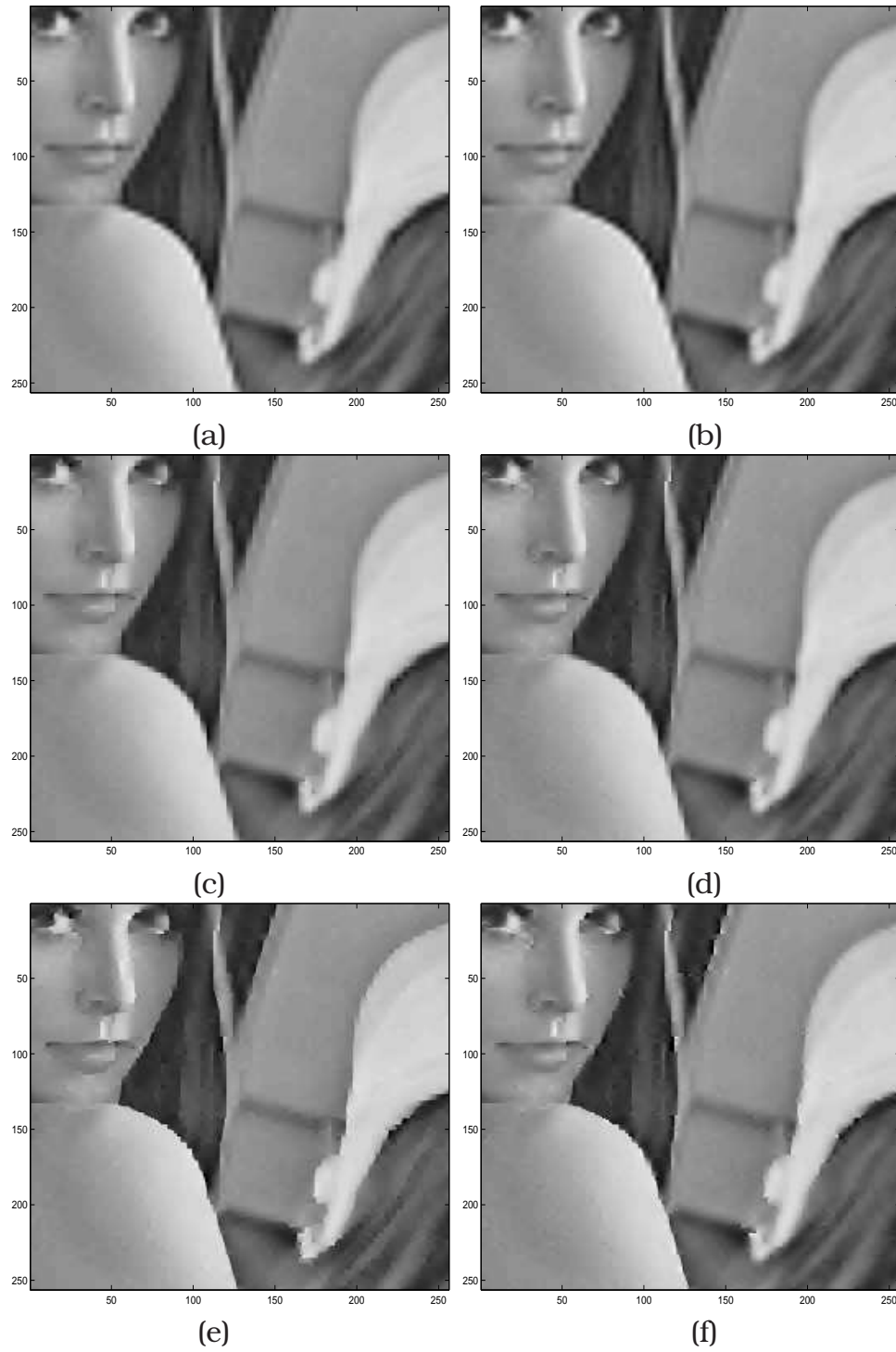


Figura 3.27: Imagen Lena. Reconstrucción sin utilizar detalles: (a) LIN3, (b) LIN5, (c) ENO3, (d) ENO5, (e) SR3, (f) SR5.



Figura 3.28: Imagen Lena. Reconstrucción sin utilizar detalles: (a) SHU2, (b) MOD2, (c) LIU3, (d) SHU3, (e) MOD3.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$
LIN3	97.5111	4.9921	8.8796
ENO3	172.2599	5.4191	10.1188
SR3	191.9553	5.7251	10.9625
SHU2	133.6459	5.5105	10.1123
MOD2	113.9781	5.0071	9.0471
LIN5	99.4631	4.9124	8.6640
ENO5	277.5211	5.4857	10.5451
SR5	393.9281	5.7049	11.4575
LIU3	159.1352	5.2387	9.7032
SHU3	132.2640	5.1426	9.4662
MOD3	107.5727	4.8724	8.7518

Tabla 3.8: Imagen Lena. Errores de la reconstrucción sin detalles.

con las técnicas ENO, ENO-SR y LIU.

3.3.2

Estabilidad

En esta sección vamos a comparar numéricamente la estabilidad de los algoritmos en 2D cuando utilizamos distintas estrategias de reconstrucción (interpolación lineal, ENO, ENO-SR, WENO). El análisis de la estabilidad en 1D ya ha sido estudiado por distintos autores, y se ha demostrado que la técnica WENO es estable (ver por ejemplo [15, 16, 34]).

Dada una imagen original \bar{f}^L se le aplica una transformada 2D para medias en celda y los coeficientes de escala de la descomposición son cuantizados tomando en cada nivel de resolución un parámetro de cuantización $\epsilon_k = \epsilon/2^{L-k}$, con lo que obtenemos:

$$Q_\epsilon M \bar{f}^L = \{\hat{f}^0, \hat{d}^1(1), \hat{d}^1(2), \hat{d}^1(3), \dots, \hat{d}^L(1), \hat{d}^L(2), \hat{d}^L(3)\}.$$

Al aplicar la transformada inversa a $Q_\epsilon M \bar{f}^L$ obtenemos \hat{f}^L que idealmente tendría que satisfacer:

$$\|\bar{f}^L - \hat{f}^L\| \leq C\epsilon \quad (3.3)$$

Si la transformación 2D que utilizamos está basada en una reconstrucción lineal se puede probar que (3.3) es cierto, aunque puede ser difícil encontrar la cota C . Si la reconstrucción es no lineal, en principio no está garantizado que (3.3) sea cierto.

Teniendo en cuenta que se ha probado que la reconstrucción WENO es estable en 1D, es esperable que también lo sea en 2D. En el caso de que utilicemos la reconstrucción ENO no hay ninguna garantía de estabilidad, y si se utiliza la reconstrucción ENO-SR tenemos todavía menos esperanzas de tener estabilidad debido a que esta reconstrucción es más *no lineal*.

Compararemos los resultados obtenidos con la transformada 2D para medias en celda sin control del error (ver Nota 1.2), que denotaremos **Algoritmo sin EC**, al utilizar distintas reconstrucciones con diferentes órdenes de aproximación y distintos parámetros de cuantización ϵ , con el algoritmo de control del error explicado en la Sección 1.8 (Algoritmo B.1), que denotaremos **Algoritmo con EC**, también con las mismas reconstrucciones de distintos órdenes. Recordemos que con este algoritmo tenemos garantizada la estabilidad independientemente de la reconstrucción que se utilice (Corolario 1.2). Concretamente tenemos:

$$\|\bar{f}^L - \hat{f}^L\|_\infty \leq 2\epsilon; \quad \|\bar{f}^L - \hat{f}^L\|_1 \leq \epsilon; \quad \text{y} \quad \|\bar{f}^L - \hat{f}^L\|_2 \leq \sqrt{\frac{2}{3}}\epsilon. \quad (3.4)$$

Las imágenes que utilizamos son de tamaño 512×512 y tomamos en todos los casos $L = 4$, con lo que el coarse tiene tamaño 32×32 .

En las Tablas 3.9-3.14 se muestran los errores en la reconstrucción, medidos en norma $\|\cdot\|_1$, $\|\cdot\|_2$ y $\|\cdot\|_\infty$, y los bytes que ocupa la matriz transformada cuantizada (que incluye la imagen en el nivel de resolución más bajo), una vez esta ha sido comprimida con el programa bzip2. Las imágenes originales ocupan 262598 bytes.

Podemos observar que siempre que utilizamos el **Algoritmo con EC**, independientemente del tipo de reconstrucción y del orden de esta, tenemos que el error es menor que las cotas de (3.4).

Con el **Algoritmo sin EC** sí que vemos que en algunas reconstrucciones los errores sobrepasan las cotas de (3.4). Si utilizamos reconstrucción lineal obtenemos cotas de error parecidas a las obtenidas con el **Algoritmo con EC**. Con las reconstrucciones WENO

también obtenemos cotas de error comparables a las obtenidas con el **Algoritmo con EC**. Vemos que si la reconstrucción es ENO los errores son mayores al utilizar el **Algoritmo sin EC** que cuando utilizamos el **Algoritmo con EC**. La diferencia aún es mayor cuando la reconstrucción es ENO-SR.

Algoritmo	Técnica	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	bytes
Con EC	LIN3	20.8936	0.8326	1.8606	4554
	ENO3	23.9508	0.3758	1.2030	3372
	SR3	23.5598	0.2901	1.0924	2320
	SHU2	21.8330	0.3369	1.1741	3440
	MOD2	20.3334	0.8421	1.8158	4212
	LIN5	22.6471	1.1073	2.0291	4578
	ENO5	23.9204	0.6591	1.5511	3708
	SR5	23.8511	0.6331	1.5237	3365
	LIU3	23.5823	0.3942	1.2145	3436
	SHU3	23.3604	0.3725	1.1844	3357
MOD3	21.5767	1.0081	1.9247	4301	
Sin EC	LIN3	21.8181	0.8184	1.8497	4498
	ENO3	35.7314	0.3528	1.2874	3380
	SR3	164.8990	0.3255	2.5422	1545
	SHU2	32.8713	0.2994	1.1964	3415
	MOD2	30.9264	0.8256	1.8129	4072
	LIN5	21.8575	1.1049	2.0539	4389
	ENO5	69.4122	0.6507	1.7105	3389
	SR5	256.5238	1.0300	7.8957	1826
	LIU3	37.6450	0.3681	1.2865	3345
	SHU3	40.5905	0.3446	1.2493	3284
MOD3	36.5960	0.9932	1.9248	4144	

Tabla 3.9: Imagen geométrica. Errores y compresión al cuantizar con parámetro $\epsilon = 16$.

Algoritmo	Técnica	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	bytes
Con EC	LIN3	38.8990	1.7497	3.6136	2885
	ENO3	46.2864	0.9556	2.7814	2526
	SR3	45.5914	0.8103	2.4369	2050
	SHU2	43.8198	0.9697	2.7774	2743
	MOD2	41.6227	1.6448	3.4877	2873
	LIN5	46.3447	2.0376	3.6831	2869
	ENO5	49.0711	1.3030	3.1252	2736
	SR5	47.8608	1.2009	3.1339	2760
	LIU3	47.0022	0.9645	2.7517	2577
	SHU3	46.0822	0.9546	2.6659	2431
MOD3	44.2753	1.8180	3.5674	2624	
Sin EC	LIN3	43.3918	1.6365	3.5217	2743
	ENO3	112.7888	0.7694	2.9862	2489
	SR3	244.9701	0.5964	3.7913	1201
	SHU2	58.3830	0.6681	2.7882	2613
	MOD2	45.4650	1.5141	3.3937	2730
	LIN5	48.4312	1.9254	3.5959	2776
	ENO5	205.9682	1.2567	3.5535	2238
	SR5	368.5212	1.5826	6.4661	1339
	LIU3	64.1621	0.7657	2.9117	2486
	SHU3	64.9415	0.7484	2.8646	2284
MOD3	67.3870	1.7115	3.5131	2482	

Tabla 3.10: Imagen geométrica. Errores y compresión al cuantizar con parámetro $\epsilon = 32$.

Algoritmo	Técnica	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	bytes
Con EC	LIN3	25.7423	3.6709	4.7559	13905
	ENO3	25.0052	3.7443	4.9145	14777
	SR3	25.3342	3.7892	4.9708	15472
	SHU2	24.6040	3.7164	4.8631	14859
	MOD2	24.1688	3.6224	4.7097	13679
	LIN5	24.4506	3.7242	4.8088	13908
	ENO5	25.0146	3.8644	5.0657	15927
	SR5	26.8820	3.8763	5.0793	16448
	LIU3	24.7006	3.7015	4.8433	14385
	SHU3	25.1608	3.6609	4.7832	13920
MOD3	24.7526	3.6435	4.7272	13245	
Sin EC	LIN3	31.1104	3.6682	4.7678	13820
	ENO3	166.8953	4.1047	5.9321	14177
	SR3	315.0907	5.5269	9.8596	14651
	SHU2	36.0790	3.7548	4.9702	14584
	MOD2	32.9079	3.6158	4.7297	13526
	LIN5	72.8602	3.7704	4.9257	13682
	ENO5	553.5395	4.8934	8.1667	14559
	SR5	582.3165	6.2854	12.6566	14955
	LIU3	51.9830	3.8830	5.2117	13737
	SHU3	41.9003	3.7733	5.0173	13487
MOD3	34.9254	3.6628	4.7917	13002	

Tabla 3.11: Imagen pimientos. Errores y compresión al cuantizar con parámetro $\epsilon = 16$.

Algoritmo	Técnica	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	bytes
Con EC	LIN3	48.0995	4.8931	6.5789	7140
	ENO3	49.5317	5.0191	6.9469	7943
	SR3	50.3176	5.1251	7.1166	8427
	SHU2	48.2072	4.9524	6.8260	7945
	MOD2	46.8461	4.7959	6.5074	7159
	LIN5	46.8735	4.9995	6.7070	7192
	ENO5	51.5539	5.2745	7.3151	8719
	SR5	51.5186	5.3112	7.3638	9333
	LIU3	49.8200	4.9521	6.8148	7629
	SHU3	49.2560	4.8723	6.6764	7421
MOD3	49.5505	4.8352	6.5332	6938	
Sin EC	LIN3	54.0835	4.8153	6.5362	6996
	ENO3	340.5430	5.3512	8.3075	7441
	SR3	390.7369	6.9078	12.5697	7428
	SHU2	63.7675	4.9262	6.9211	7714
	MOD2	59.3957	4.7096	6.4753	7082
	LIN5	172.0886	4.9884	6.7996	6936
	ENO5	798.3322	6.1694	10.5992	7392
	SR5	733.6987	7.6011	15.3016	7840
	LIU3	91.2859	5.0687	7.2449	7146
	SHU3	69.7785	4.9078	6.9352	7058
MOD3	73.3445	4.7788	6.5677	6699	

Tabla 3.12: Imagen pimientos. Errores y compresión al cuantizar con parámetro $\epsilon = 32$.

Algoritmo	Técnica	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	bytes
Con EC	LIN3	24.8942	3.3528	4.4487	14226
	ENO3	27.0554	3.5259	4.7413	16538
	SR3	27.0351	3.5790	4.8015	17581
	SHU2	24.4476	3.5052	4.6861	16376
	MOD2	24.8742	3.3389	4.4358	14690
	LIN5	25.2184	3.3750	4.4576	14008
	ENO5	25.3544	3.5821	4.8031	17220
	SR5	25.3544	3.5972	4.8209	17724
	LIU3	26.4442	3.4723	4.6494	15645
	SHU3	24.6749	3.4197	4.5701	15213
MOD3	24.4941	3.3360	4.4167	14091	
Sin EC	LIN3	27.1093	3.3403	4.4522	14127
	ENO3	313.1435	3.9447	6.0103	15866
	SR3	311.3847	5.1877	9.4346	16800
	SHU2	35.2387	3.5511	4.8204	16148
	MOD2	33.3230	3.3215	4.4465	14481
	LIN5	59.7006	3.4020	4.5346	13590
	ENO5	493.0142	4.8965	8.8727	15980
	SR5	542.2287	6.0255	12.6338	16442
	LIU3	48.9898	3.6881	5.1208	15300
	SHU3	41.7458	3.5207	4.8121	14865
MOD3	32.2507	3.3411	4.4662	13785	

Tabla 3.13: Imagen Lena. Errores y compresión al cuantizar con parámetro $\epsilon = 16$.

Algoritmo	Técnica	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	bytes
Con EC	LIN3	49.5737	4.6940	6.4972	7512
	ENO3	48.9870	4.9998	7.1398	8853
	SR3	51.0962	5.0954	7.2860	9712
	SHU2	46.8296	4.9558	7.0171	8799
	MOD2	47.8891	4.6743	6.5029	7799
	LIN5	47.3279	4.7713	6.5425	7438
	ENO5	50.9343	5.1615	7.3327	9674
	SR5	50.8891	5.1989	7.3863	10220
	LIU3	51.5471	4.9075	6.9667	8537
	SHU3	47.8834	4.8087	6.7961	8218
MOD3	45.3051	4.6697	6.4568	7407	
Sin EC	LIN3	56.2794	4.5926	6.4362	7460
	ENO3	286.7613	5.3577	8.4999	8416
	SR3	322.5753	6.7422	12.4446	8873
	SHU2	68.3184	4.9205	7.1437	8489
	MOD2	59.9845	4.5694	6.4602	7655
	LIN5	107.4039	4.7144	6.5716	7200
	ENO5	317.4725	6.3762	10.9718	8568
	SR5	642.7080	7.8054	16.0333	8924
	LIU3	83.1361	5.0287	7.4800	8080
	SHU3	80.4102	4.8431	7.0638	7862
MOD3	75.3917	4.5807	6.4515	7299	

Tabla 3.14: Imagen Lena. Errores y compresión al cuantizar con parámetro $\epsilon = 32$.

4

Interpolación WENO para valores puntuales

4.1

Introducción

Las ideas que conducen de reconstrucciones ENO a ENO Ponderado (WENO), desarrollado y de uso extendido para el diseño de esquemas de alto orden para leyes de conservación, son aplicadas en este capítulo en el contexto de interpolación de valores puntuales.

Probaremos que generalmente se puede, utilizando r stencils de $r + 1$ puntos, alcanzar un orden de exactitud de $2r$ en regiones donde la función interpolada es suave. Esta técnica interpolatoria puede ser utilizada en el marco de multirresolución de Harten para

aplicaciones de compresión de imagen.

Más concretamente, calculamos los pesos no lineales sobre los cuales se basa la interpolación que presentamos siguiendo las ideas de [33] y estos dependen de los indicadores de suavidad de los *sub-stencils*, definidos a partir de los indicadores de suavidad propuestos en [27], pero calculados utilizando los interpoladores de Lagrange correspondientes, en lugar de los interpoladores de las medias en celda.

4.2

Interpolación WENO

En el contexto de valores puntuales, si denotamos por S_{j+k} , $k = 0, \dots, r-1$, a los r stencils posibles que contienen los nodos x_{j-1}, x_j :

$$S_{j+k} = \{x_{j+k-r}, \dots, x_{j+k}\}, \quad k = 0, \dots, r-1,$$

y llamamos $p_{j+k}(x)$ al polinomio interpolador construido a partir del stencil S_{j+k} , entonces el interpolante WENO viene dado por la combinación:

$$p_j(x) = \sum_{k=0}^{r-1} \omega_k^j p_{j+k}(x) \quad (4.1)$$

donde los pesos ω_k^j se definen de forma que el polinomio $p_j(x)$ resultante satisfaga una serie de requisitos que se explican más adelante.

El operador predicción correspondiente será:

$$\mathcal{I}(x_{j-\frac{1}{2}}^{l-1}; f^{l-1}) = \sum_{k=0}^{r-1} \omega_k^j p_{j+k}(x_{j-\frac{1}{2}}^{l-1}). \quad (4.2)$$

Dado que podemos considerar fijado el nivel de resolución l , en adelante se eliminará el superíndice $l-1$ para facilitar la notación, y denotaremos por h, x_j, f_j , a $h^{l-1}, x_j^{l-1}, f_j^{l-1}$.

Al igual que en el Capítulo 2, el polinomio obtenido mediante la técnica de interpolación WENO debe cumplir las siguientes condiciones:

En primer lugar, es un polinomio interpolador a trozos formado por polinomios de grado r que proporcionan un orden de exactitud $r + 1$:

$$p_j(x) = f(x) + O(h^{r+1}), \quad x \in I_j.$$

En segundo, cada polinomio $p_j(x)$ debe satisfacer la “propiedad ENO”:

1. Si la función f es suave en el stencil S_{j+k} , entonces el peso correspondiente a este stencil verifica:

$$\omega_k^j = O(1). \quad (4.3)$$

2. Si la función f presenta una singularidad en el stencil S_{j+k} , entonces su correspondiente ω_k^j verifica:

$$\omega_k^j \leq O(h^r). \quad (4.4)$$

Si los pesos ω_k^j de la combinación convexa satisfacen la propiedad ENO (4.3)-(4.4), entonces el polinomio $p_j(x)$ dado por (4.1) es una combinación lineal de los polinomios construidos a partir de stencils en los que la función es suave (por (4.3)), en la que los polinomios correspondientes a stencils que cruzan alguna singularidad tienen una contribución prácticamente nula (por (4.4)).

Al igual que en el WENO para medias en celda, una correcta elección de los pesos de la combinación lineal (4.1) es fundamental para el éxito de la interpolación WENO. Una condición básica que deben satisfacer, por motivos de estabilidad y consistencia, es la siguiente:

$$\omega_k^j \geq 0, \quad \forall k, \quad \sum_{k=0}^{r-1} \omega_k^j = 1. \quad (4.5)$$

Por otra parte, el polinomio interpolador $p_j(x)$ debe cumplir la propiedad ENO (4.3)-(4.4), por lo que en la definición de los pesos tendrá una gran importancia la elección de una medida de la suavidad de la función en cada stencil S_{j+k} . Esta medida viene dada por un *indicador de suavidad*, IS_{j+k} .

Estas consideraciones llevan a la siguiente expresión propuesta por Liu et al. [33] para los pesos:

$$\omega_k^j = \frac{\alpha_k^j}{\sum_{l=0}^{r-1} \alpha_l^j}, \quad (4.6)$$

donde α_k^j viene dado por la expresión:

$$\alpha_k^j = \frac{C_k^j}{(\varepsilon + IS_{j+k})^r}, \quad k = 0, \dots, r-1. \quad (4.7)$$

Al igual que en el Capítulo 2, ε es una constante positiva que se introduce para evitar que el denominador se anule, y se suele tomar el valor $\varepsilon = 10^{-5}$ o $\varepsilon = 10^{-6}$. Las C_k^j son constantes también positivas, llamadas *pesos óptimos*, cuya cálculo se explica a continuación.

Respecto a los indicadores de suavidad se tiene el siguiente resultado, ya visto en el contexto de interpolación de medias en celda:

Proposición 4.1. *Para que los pesos de la combinación lineal satisfagan la propiedad ENO (4.3)-(4.4), es suficiente que el indicador de suavidad cumpla la siguiente condición:*

- Si la función f es suave en el stencil S_{j+k} , entonces:

$$IS_{j+k} = O(h^2), \quad (4.8)$$

- Si la función f presenta una singularidad en el stencil S_{j+k} , entonces:

$$IS_{j+k} = O(1), \quad (4.9)$$

Demostración. Ver Proposición 2.1. □

4.3

Pesos óptimos

Consideramos los r polinomios $p_{j+k}(x)$, $k = 0, \dots, r-1$, que interpolan a la función f en los stencils S_{j+k} . En el contexto de multirresolución en el que nos encontramos la interpolación se utiliza como un proceso de reconstrucción para recuperar la información de un nivel de resolución a partir de la información contenida en

el nivel inmediatamente inferior. En el caso de interpolación por valores puntuales esto se consigue evaluando el polinomio interpolador en el punto medio de cada subintervalo I_j , x_{2j-1} . De este modo, para cada par de puntos x_{j-1}, x_j obtenemos r aproximaciones \hat{p}_{j+k}^r a f_{2j-1} :

$$\hat{p}_{j+k}^r = f_{2j-1} + O(h^{r+1}) \quad (4.10)$$

Por otra parte, si tomamos el stencil que resulta de la unión de los r stencils S_{j+k} , podemos calcular una aproximación \hat{p}_j^{2r-1} con un mayor orden de exactitud, igual a $2r$. Si la función f es suave en todos los stencils, entonces existen constantes C_k^j tales que:

$$\hat{p}_j^{2r-1} = \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r.$$

Estas constantes C_k^j (pesos óptimos) son siempre positivas, y por consistencia cumplen:

$$\sum_{k=0}^{r-1} C_k^j = 1. \quad (4.11)$$

Sería deseable, para conseguir el mayor orden de aproximación posible con el interpolante WENO, que los pesos de la combinación lineal convexa (4.1) aproximen a estos pesos óptimos. De hecho tenemos la siguiente proposición:

Proposición 4.2. Si los pesos ω_k^j cumplen la siguiente relación:

$$\omega_k^j = C_k^j + O(h^m), \quad k = 0, \dots, r-1, \quad (4.12)$$

entonces

$$\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r = f_{2j-1} + O(h^{\min(r+m+1, 2r)}) \quad (4.13)$$

Demostración. Comprobemos esta afirmación:

$$\begin{aligned}
\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \hat{p}_j^{2r-1} &= \sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r \\
&= \sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r + f_{2j-1} \sum_{k=0}^{r-1} C_k^j - f_{2j-1} \sum_{k=0}^{r-1} \omega_k^j \\
&= \sum_{k=0}^{r-1} \omega_k^j (\hat{p}_{j+k}^r - f_{2j-1}) - \sum_{k=0}^{r-1} C_k^j (\hat{p}_{j+k}^r - f_{2j-1}) \\
&= \sum_{k=0}^{r-1} (\omega_k^j - C_k^j) (\hat{p}_{j+k}^r - f_{2j-1}) \\
&= \sum_{k=0}^m O(h^m) O(h^{r+1}) = O(h^{r+m+1}). \tag{4.14}
\end{aligned}$$

Aquí utilizamos el hecho de que $\sum_{k=0}^{r-1} \omega_k^j = \sum_{k=0}^{r-1} C_k^j = 1$, por (4.5) y (4.11), y también, en la penúltima igualdad, las relaciones dadas por (4.10) y (4.12).

A partir de esta relación concluimos que el interpolante WENO aproxima a la función con un orden de exactitud $\min(r+m+1, 2r)$, ya que:

$$\begin{aligned}
\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - f_{2j-1} &= \left(\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r - \hat{p}_j^{2r-1} \right) + (\hat{p}_j^{2r-1} - f_{2j-1}) \\
&= O(h^{r+m+1}) + O(h^{2r}) = O(h^{\min(r+m+1, 2r)}).
\end{aligned}$$

□

Nota 4.1. Si $m = 1$ tendremos

$$\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r = f_{2j-1} + O(h^{r+2}),$$

es decir, mejoramos el orden de aproximación de \hat{p}_{j+k}^r en 1 consiguiendo $r+2$. En cambio, si $m = r-1$

$$\sum_{k=0}^{r-1} \omega_k^j \hat{p}_{j+k}^r = f_{2j-1} + O(h^{2r})$$

con lo que conseguimos un orden de exactitud igual al de \hat{p}_j^{2r-1} que es $2r$. Es decir, conseguimos el orden óptimo.

La forma más sencilla de calcular los pesos óptimos C_k^j es la siguiente:

1. Dado r , tomamos las aproximaciones \hat{p}_{j+k}^r obtenidas a partir de los stencils $S_{j+k} = \{x_{j+k-r}, \dots, x_{j+k}\}$. Por otra parte, consideramos \hat{p}_j^{2r-1} calculada a partir del stencil que resulta de la unión de los stencils anteriores, es decir, $S = \{x_{j-r}, \dots, x_{j+r-1}\}$.
2. Se igualan coeficientes en la combinación:

$$\hat{p}_j^{2r-1} = \sum_{k=0}^{r-1} C_k^j \hat{p}_{j+k}^r$$

3. Por último, se resuelve el sistema de ecuaciones resultante, en el que las incógnitas son justamente los pesos óptimos C_k^j , $k = 0, \dots, r-1$.

Veamos algunos ejemplos.

Cálculo de los pesos óptimos para $r = 2$

A partir de los polinomios interpoladores, calculamos las aproximaciones:

$$\begin{aligned}\hat{p}_j^2 &= -\frac{1}{8}f_{j-2} + \frac{3}{4}f_{j-1} + \frac{3}{8}f_j \\ \hat{p}_{j+1}^2 &= \frac{3}{8}f_{j-1} + \frac{3}{4}f_j - \frac{1}{8}f_{j+1}\end{aligned}$$

y también:

$$\hat{p}_j^3 = -\frac{1}{16}f_{j-2} + \frac{9}{16}f_{j-1} + \frac{9}{16}f_j - \frac{1}{16}f_{j+1}.$$

Si consideramos la combinación:

$$\hat{p}_j^3 = C_0^j \hat{p}_j^2 + C_1^j \hat{p}_{j+1}^2$$

e igualamos coeficientes obtenemos un sistema cuya solución es:

$$C_0^j = C_1^j = \frac{1}{2}$$

Cálculo de los pesos óptimos para $r = 3$

Si aumentamos el orden de la interpolación, y consideramos $r = 3$, las aproximaciones en este caso son las siguientes:

$$\begin{aligned}\hat{p}_j^3 &= \frac{1}{16}f_{j-3} - \frac{5}{16}f_{j-2} + \frac{15}{16}f_{j-1} + \frac{5}{16}f_j \\ \hat{p}_{j+1}^3 &= -\frac{1}{16}f_{j-2} + \frac{9}{16}f_{j-1} + \frac{9}{16}f_j - \frac{1}{16}f_{j+1} \\ \hat{p}_{j+2}^3 &= \frac{5}{16}f_{j-1} + \frac{15}{16}f_j - \frac{5}{16}f_{j+1} + \frac{1}{16}f_{j+2}\end{aligned}$$

y la aproximación construida a partir de los 6 nodos es:

$$\hat{p}_j^5 = \frac{3}{256}f_{j-3} - \frac{25}{256}f_{j-2} + \frac{150}{256}f_{j-1} + \frac{150}{256}f_j - \frac{25}{256}f_{j+1} + \frac{3}{256}f_{j+2}.$$

Considerando la combinación de las distintas aproximaciones e igualando coeficientes, obtenemos un sistema de ecuaciones, cuya solución viene dada por:

$$C_0^j = C_2^j = \frac{3}{16}, C_1^j = \frac{10}{16}$$

Resumiendo, hemos obtenido los siguientes pesos óptimos:

	$k = 0$	$k = 1$	$k = 2$
$r = 2$	1/2	1/2	-
$r = 3$	3/16	10/16	3/16

Tabla 4.1: Pesos óptimos para $r = 2, 3$.

4.4

Indicadores de suavidad

Como ha quedado explicado anteriormente, un elemento importante en la construcción del interpolante WENO es encontrar una forma conveniente de medir la suavidad de la función que

se interpola. Para ello se utiliza una función llamada *indicador de suavidad*.

Recordando lo expuesto en la Sección 4.2, el indicador de suavidad debe cumplir que $IS_{j+k} = O(h^2)$ en los stencils S_{j+k} en los que la función es suave, e $IS_{j+k} = O(1)$ en los stencils S_{j+k} en los que la función cruza alguna singularidad.

Además, tenemos el siguiente resultado:

Proposición 4.3. *Si tenemos*

$$\frac{IS_{j+k} - IS_{j+l}}{IS_{j+k}} = O(h^m), \quad 0 \leq k, l \leq r-1, \quad (4.15)$$

o equivalentemente

$$IS_{j+k} = D(1 + O(h^m)), \quad k = 0, \dots, r-1, \quad (4.16)$$

con D una constante distinta de cero e independiente de k , aunque puede depender de h , entonces

$$\omega_k^j = C_k^j + O(h^m) \quad (4.17)$$

Demostración. Supongamos que tenemos (4.16), que es equivalente a (4.15). Partiendo de la expresión (4.6) para los pesos ω_k^j :

$$\omega_k^j = \frac{\alpha_k^j}{\sum_{l=0}^{r-1} \alpha_l^j} = \frac{C_k^j}{(\varepsilon + IS_{j+k})^r} \cdot \frac{1}{\sum_{l=0}^{r-1} \frac{C_l^j}{(\varepsilon + IS_{j+l})^r}}. \quad (4.18)$$

Haciendo uso de la relación (4.16), y teniendo en cuenta que ε es una constante que se puede despreciar y $\sum_{l=0}^{r-1} C_l^j = 1$ (por (4.11)), tenemos:

$$\begin{aligned} \omega_k^j &= \frac{C_k^j}{[D(1 + O(h^m))]^r} \cdot \frac{1}{\sum_{l=0}^{r-1} \frac{C_l^j}{[D(1 + O(h^m))]^r}} \\ &= \frac{C_k^j}{[D(1 + O(h^m))]^r} \cdot \frac{[D(1 + O(h^m))]^r}{\sum_{l=0}^{r-1} C_l^j} \\ &= \frac{C_k^j}{[D(1 + O(h^m))]^r} [D(1 + O(h^m))]^r \end{aligned} \quad (4.19)$$

y como D es una constante independiente de k :

$$\omega_k^j = \frac{C_k^j}{(1 + O(h^m))^r} (1 + O(h^m))^r. \quad (4.20)$$

Por otra parte, desarrollando la potencia $(1 + O(h^m))^r$, obtenemos:

$$(1 + O(h^m))^r = 1 + O(h^m) + O(h^{2r-2}) + \dots + O(h^{r(m)}) = 1 + O(h^m)$$

con lo que llegamos al resultado buscado:

$$\omega_k^j = \frac{C_k^j}{(1 + O(h^m))} (1 + O(h^m)) = C_k^j + O(h^m). \quad (4.21)$$

□

4.4.1

Indicador de suavidad de Liu et al.

El indicador de suavidad (2.18) ha sido definido en términos de las medias en celda de la función interpolada. Para poder construir el interpolante WENO utilizando los valores puntuales de dicha función necesitamos modificar la expresión del indicador para que se adapte a este nuevo contexto.

Para realizar esta adaptación, partimos de la relación (1.17) vista en la Sección 1.3, que relaciona las medias en celda de una función con los valores puntuales de su primitiva. Según esta relación, la media de una función f en una celda $I_j = [x_{j-1}, x_j]$ es igual a la diferencia dividida de su primitiva F en los nodos x_{j-1}, x_j . Por ello, para definir el indicador de suavidad en el caso de interpolación por valores puntuales, simplemente tomamos la expresión ya definida para medias en celda, y sustituimos las medias en celda por diferencias no divididas de los valores puntuales de la función en los extremos de la celda correspondiente. Utilizamos las diferencias no divididas en vez de las divididas para evitar la dependencia del indicador de suavidad de h .

En otras palabras, si denotamos a estas diferencias no divididas de la siguiente manera:

$$f[x_{j-1}, x_j] = f(x_j) - f(x_{j-1})$$

el indicador de suavidad definido por Liu et al. vendrá dado en el caso $r = 2$ por:

$$\begin{aligned} IS_{j+k} &= (f[x_{j+k-1}, x_{j+k}] - f[x_{j+k-2}, x_{j+k-1}])^2, \\ &= (f_{j+k} - 2f_{j+k-1} + f_{j+k-2})^2, \quad k = 0, 1, \end{aligned} \quad (4.22)$$

y cuando $r = 3$:

$$\begin{aligned} IS_{j+k} &= \frac{1}{2}[(f[x_{j+k-2}, x_{j+k-1}] - f[x_{j+k-3}, x_{j+k-2}])^2 \\ &\quad + (f[x_{j+k-1}, x_{j+k}] - f[x_{j+k-2}, x_{j+k-1}])^2] \\ &\quad + (f[x_{j+k-1}, x_{j+k}] - 2f[x_{j+k-2}, x_{j+k-1}] \\ &\quad + f[x_{j+k-3}, x_{j+k-2}])^2, \\ &= \frac{1}{2}[(f_{j+k} - 2f_{j+k-1} + f_{j+k-2})^2 + (f_{j+k+1} - 2f_{j+k} + f_{j+k-1})^2] \\ &\quad + (f_{j+k+1} - 3f_{j+k} + 3f_{j+k-1} - f_{j+k-2})^2 \quad k = 0, 1, 2. \end{aligned} \quad (4.23)$$

Para saber con qué orden de exactitud aproxima el interpolante WENO construido a partir de estos indicadores de suavidad a la función f , desarrollamos por Taylor las expresiones anteriores. En primer lugar, si $r = 2$:

$$\begin{aligned} IS_{j+k} &= \frac{1}{2}(f_{j+k} - 2f_{j+k-1} + f_{j+k-2})^2 \\ &= [f_j + khf'_j \frac{(hk)^2}{2} + f''_j + O(h^3) \\ &\quad - 2(f_j + (k-1)hf'_j + \frac{((k-1)h)^2}{2}f''_j + O(h^3)) \\ &\quad + f_j + (k-2)hf'_j + \frac{((k-2)h)^2}{2}f''_j + O(h^3)] \\ &= (h^2 f''_j + O(h^3))^2 \\ &= (h^2 f''_j)^2 (1 + O(h)), \quad k = 0, 1, \end{aligned}$$

y si $r = 3$,

$$\begin{aligned}
IS_{j+k} &= \frac{1}{2}[(f_{j+k} - 2f_{j+k-1} + f_{j+k-2})^2 \\
&\quad + (f_{j+k+1} - 2f_{j+k} + f_{j+k-1})^2] \\
&\quad + (f_{j+k+1} - 3f_{j+k} + 3f_{j+k-1} - f_{j+k-2})^2 \\
&= [(h^2 f_j'' + O(h^3))^2 + (h^2 f_j'' + O(h^3))^2] \\
&\quad + [f_j + (k+1)hf_j' + \frac{((k+1)h)^2}{2} f_j'' + O(h^3) \\
&\quad - 3(f_j + khf_j' + \frac{(kh)^2}{2} f_j'' + O(h^3)) \\
&\quad + 3(f_j + (k-1)hf_j' + \frac{((k-1)h)^2}{2} f_j'' + O(h^3)) \\
&\quad - (f_j + (k-2)hf_j' + \frac{((k-2)h)^2}{2} f_j'' + O(h^3))]^2 \\
&= (h^2 f_j'')^2 (1 + O(h)) + (O(h^3))^2 \\
&= (h^2 f_j'')^2 (1 + O(h)), \quad k = 0, 1, 2.
\end{aligned}$$

En ambos casos, utilizando la Proposición 4.3 se tiene que:

$$\omega_k^j = C_k^j + O(h).$$

Ahora, por la Proposición 4.2 tenemos que si $r = 2$

$$\sum_{k=0}^1 \omega_k^j \hat{p}_{j+k}^2 = f_{2j-1} + O(h^4)$$

que es el peso óptimo que se puede esperar cuando se utiliza en la interpolación con 4 nodos. Observemos que si $r = 2$, $r + 2 = 2r$.

En el caso en que $r = 3$ tenemos, también por la Proposición 4.2,

$$\sum_{k=0}^2 \omega_k^j \hat{p}_{j+k}^3 = f_{2j-1} + O(h^5).$$

Aquí, al igual que pasaba en medias en celda, utilizando el indicador de suavidad definido por Liu et al. no conseguimos una aproximación con el orden óptimo de exactitud, $O(h^6)$, que se tiene cuando se considera interpolación con 6 nodos. Para mejorar este resultado Jiang y Shu presentaron en [27] un nuevo indicador de suavidad, que permite obtener un orden de exactitud mayor.

4.4.2

Indicador de suavidad de Jiang y Shu

Jiang y Shu [27] parten de un enfoque diferente al utilizado por Liu et al. para definir su indicador de suavidad. Como ya hemos visto, a partir de cada uno de los r stencils S_{j+k} se construye un polinomio interpolador que proporciona una aproximación al valor f_{2j-1} . El procedimiento para calcular el indicador de suavidad definido por Jiang y Shu en el caso de interpolación por valores puntuales es el mismo que el utilizado en el caso de medias en celda. En primer lugar se construyen los polinomios interpoladores $p_{j+k}(x)$ para los distintos stencils en consideración, se calculan todas sus derivadas y se evalúa la expresión (4.24), que recordamos aquí, adaptada al caso de valores puntuales:

$$IS_{j+k} = \sum_{l=1}^{r-1} \int_{x_{j-1/2}}^{x_{j+1/2}} h^{2l-1} (p_{j+k}^{(l)}(x))^2 dx \quad (4.24)$$

Cálculo del indicador de suavidad para $r = 2$

Los polinomios que interpolan en los stencils $S_j = \{x_{j-2}, x_{j-1}, x_j\}$ y $S_{j+1} = \{x_{j-1}, x_j, x_{j+1}\}$ a la función f son los siguientes:

$$\begin{aligned} p_j(x) &= f_j + \frac{3f_j - 4f_{j-1} + f_{j-2}}{2h}(x - x_j) + \frac{f_j - 2f_{j-1} + f_{j-2}}{2h^2}(x - x_j)^2 \\ p_{j+1}(x) &= f_j + \frac{f_{j+1} - f_{j-1}}{2h}(x - x_j) + \frac{f_{j+1} - 2f_j + f_{j-1}}{2h^2}(x - x_j)^2 \end{aligned}$$

y derivando estos polinomios obtenemos:

$$\begin{aligned} p'_j(x) &= \frac{3f_j - 4f_{j-1} + f_{j-2}}{2h} + \frac{f_j - 2f_{j-1} + f_{j-2}}{h^2}(x - x_j) \\ p''_j(x) &= \frac{f_j - 2f_{j-1} + f_{j-2}}{h^2} \\ p'_{j+1}(x) &= \frac{f_{j+1} - f_{j-1}}{2h} + \frac{f_{j+1} - 2f_j + f_{j-1}}{h^2}(x - x_j) \\ p''_{j+1}(x) &= \frac{f_{j+1} - 2f_j + f_{j-1}}{h^2} \end{aligned}$$

De este modo, sustituyendo en (4.24):

$$\begin{aligned} IS_j &= \frac{13}{12}(f_j - 2f_{j-1} + f_{j-2})^2 + \frac{1}{4}(3f_j - 4f_{j-1} + f_{j-2})^2 \\ IS_{j+1} &= \frac{13}{12}(f_{j+1} - 2f_j + f_{j-1})^2 + \frac{1}{4}(f_{j+1} - f_{j-1})^2 \end{aligned} \quad (4.25)$$

Cálculo del indicador de suavidad para $r = 3$

En este caso los stencils considerados son $S_j = \{x_{j-3}, x_{j-2}, x_{j-1}, x_j\}$, $S_{j+1} = \{x_{j-2}, x_{j-1}, x_j, x_{j+1}\}$ y $S_{j+2} = \{x_{j-1}, x_j, x_{j+1}, x_{j+2}\}$, y los polinomios que interpolan a f en estos stencils son los siguientes:

$$\begin{aligned} p_j(x) &= f_j + \frac{11f_j - 18f_{j-1} + 9f_{j-2} - 2f_{j-3}}{6h}(x - x_j) \\ &+ \frac{6f_j - 15f_{j-1} + 12f_{j-2} - 3f_{j-3}}{6h^2}(x - x_j)^2 \\ &+ \frac{f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3}}{6h^3}(x - x_j)^3 \\ p_{j+1}(x) &= f_j + \frac{2f_{j+1} + 3f_j - 6f_{j-1} + f_{j-2}}{6h}(x - x_j) \\ &+ \frac{3(f_{j+1} - 2f_j + f_{j-1})}{6h^2}(x - x_j)^2 \\ &+ \frac{f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2}}{6h^3}(x - x_j)^3 \\ p_{j+2}(x) &= f_j + \frac{-f_{j+2} + 6f_{j+1} - 3f_j - 2f_{j-1}}{6h}(x - x_j) \\ &+ \frac{3(f_{j+1} - 2f_j + f_{j-1})}{2h^2}(x - x_j)^2 \\ &+ \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{6h^3}(x - x_j)^3 \end{aligned}$$

Derivando estos tres polinomios obtenemos:

$$\begin{aligned}
p'_j(x) &= \frac{11f_j - 18f_{j-1} + 9f_{j-2} - 2f_{j-3}}{6h} \\
&+ \frac{6f_j - 15f_{j-1} + 12f_{j-2} - 3f_{j-3}}{3h^2}(x - x_j) \\
&+ \frac{f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3}}{2h^3}(x - x_j)^2 \\
p''_j(x) &= \frac{6f_j - 15f_{j-1} + 12f_{j-2} - 3f_{j-3}}{3h^2} \\
&+ \frac{f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3}}{h^3}(x - x_j) \\
p'''_j(x) &= \frac{f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3}}{h^3} \\
p'_{j+1}(x) &= \frac{2f_{j+1} + 3f_j - 6f_{j-1} + f_{j-2}}{6h} \\
&+ \frac{f_{j+1} - 2f_j + f_{j-1}}{3h^2}(x - x_j) \\
&+ \frac{f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2}}{2h^3}(x - x_j)^2 \\
p''_{j+1}(x) &= \frac{f_{j+1} - 2f_j + f_{j-1}}{3h^2} \\
&+ \frac{f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2}}{h^3}(x - x_j) \\
p'''_{j+1}(x) &= \frac{f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2}}{h^3} \\
p'_{j+2}(x) &= \frac{-f_{j+2} + 6f_{j+1} - 3f_j - 2f_{j-1}}{6h} \\
&+ \frac{f_{j+1} - 2f_j + f_{j-1}}{h^2}(x - x_j) \\
&+ \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{2h^3}(x - x_j)^2 \\
p''_{j+2}(x) &= \frac{f_{j+1} - 2f_j + f_{j-1}}{h^2} \\
&+ \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{h^3}(x - x_j) \\
p'''_{j+2}(x) &= \frac{f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}}{h^3}
\end{aligned}$$

Con lo cual llegamos a las siguientes expresiones para el indicador de suavidad:

$$\begin{aligned}
IS_j &= \frac{1}{13}(11f_j - 18f_{j-1} + 9f_{j-2} - 2f_{j-3})^2 & (4.26) \\
&+ \frac{13}{12}(2f_j - 5f_{j-1} + 4f_{j-2} - f_{j-3})^2 \\
&+ \frac{1043}{960}(f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3})^2 \\
&+ \frac{1}{72}(11f_j - 18f_{j-1} + 9f_{j-2} - 2f_{j-3})(f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3}) \\
IS_{j+1} &= \frac{1}{36}(2f_{j+1} + 3f_j - 6f_{j-1} + f_{j-2})^2 \\
&+ \frac{13}{12}(f_{j+1} - 2f_j + f_{j-1})^2 \\
&+ \frac{1043}{960}(f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2})^2 \\
&+ \frac{1}{72}(2f_{j+1} + 3f_j - 6f_{j-1} + f_{j-2})(f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2}) \\
IS_{j+2} &= \frac{1}{36}(-f_{j+2} + 6f_{j+1} - 3f_j - 2f_{j-1})^2 \\
&+ \frac{13}{12}(f_{j+2} - 2f_{j+1} + f_j)^2 \\
&+ \frac{1043}{960}(f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1})^2 \\
&+ \frac{1}{72}(-f_{j+2} + 6f_{j+1} - 3f_j - 2f_{j-1})(f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1})
\end{aligned}$$

Si analizamos ahora estos indicadores por medio de desarrollos de Taylor, vemos que el orden de exactitud alcanzado por el interpolante WENO correspondiente es en general el orden óptimo $2r$.

En primer lugar, estudiamos el caso $r = 2$:

$$\begin{aligned}
IS_j &= \frac{13}{12} \left[(f_j - 2f_{j-1} + f_{j-2}) \right]^2 + \frac{1}{4} \left[3f_j - 4f_{j-1} + f_{j-2} \right]^2 \\
&= \frac{13}{12} \left(f_j - 2(f_j - hf'_j + \frac{h^2}{2}f''_j - \frac{h^3}{6}f'''_j + \frac{h^4}{24}f^{(4)}_j + O(h^5)) \right. \\
&\quad \left. + (f_j - 2hf'_j + \frac{4h^2}{2}f''_j - \frac{8h^3}{6}f'''_j + O(h^4)) \right)^2 \\
&\quad + \frac{1}{4} \left(3f_j - 4(f_j - hf'_j + \frac{h^2}{2}f''_j - \frac{h^3}{6}f'''_j + O(h^4)) \right. \\
&\quad \left. + (f_j - 2hf'_j + \frac{4h^2}{2}f''_j - \frac{8h^3}{6}f'''_j + O(h^4)) \right)^2 \\
&= \frac{13}{12} (h^2f''_j - h^3f'''_j + O(h^4))^2 + \frac{1}{4} \left(2hf'_j - \frac{4}{6}h^3f'''_j + O(h^4) \right)^2 \\
&= \frac{13}{12} (h^2f''_j)^2 (1 + O(h)) + (hf'_j)^2 (1 + O(h^2)) \\
&= (hf'_j)^2 (1 + O(h^2))
\end{aligned}$$

$$\begin{aligned}
IS_{j+1} &= \frac{13}{12} \left[(f_{j+1} - 2f_j + f_{j-1}) \right]^2 + \frac{1}{4} \left[f_{j+1} - f_{j-1} \right]^2 \\
&= \frac{13}{12} \left((f_j + hf'_j + \frac{h^2}{2}f''_j + \frac{h^3}{6}f'''_j + O(h^4)) - 2f_j \right. \\
&\quad \left. + (f_j - hf'_j + \frac{h^2}{2}f''_j - \frac{h^3}{6}f'''_j + O(h^4)) \right)^2 \\
&\quad + \frac{1}{4} \left((f_j + hf'_j + \frac{h^2}{2}f''_j + \frac{h^3}{6}f'''_j + O(h^4)) \right. \\
&\quad \left. - (f_j - hf'_j + \frac{h^2}{2}f''_j - \frac{h^3}{6}f'''_j + O(h^4)) \right)^2 \\
&= \frac{13}{12} (h^2f''_j + O(h^4))^2 + \frac{1}{4} \left(2hf'_j - \frac{2}{6}h^3f'''_j + O(h^4) \right)^2 \\
&= \frac{13}{12} (h^2f''_j)^2 (1 + O(h^2)) + (hf'_j)^2 (1 + O(h^2)) \\
&= (hf'_j)^2 (1 + O(h^2))
\end{aligned}$$

En este caso obtenemos orden $r + 2 = 2r = 4$ que es el mejor al que podemos aspirar.

Para $r = 3$, después de un detallado análisis de los desarrollos de Taylor obtenemos

$$\begin{aligned}
IS_j &= \frac{1}{36} \left(11f_j - 18f_{j-1} + 9f_{j-2} - 2f_{j-3} \right)^2 \\
&\quad + \frac{13}{108} \left(6f_j - 15f_{j-1} + 12f_{j-2} - 3f_{j-3} \right)^2 \\
&\quad + \frac{1043}{960} \left(f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3} \right)^2 \\
&\quad + \frac{1}{70} \left(11f_j - 18f_{j-1} + 9f_{j-2} - 2f_{j-3} \right) \left(f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3} \right) \\
&= \frac{1}{36} \left(11f_j - 18(f_j - hf'_j + \frac{h^2}{2}f''_j - \frac{h^3}{6}f'''_j + \frac{h^4}{24}f^{iv}_j) + O(h^5) \right) \\
&\quad + 9 \left(f_j - 2hf'_j + \frac{4h^2}{2}f''_j - \frac{8h^3}{6}f'''_j + \frac{16}{24}h^4f^{iv}_j + O(h^5) \right) \\
&\quad - 2 \left(f_j - 3hf'_j + \frac{9h^2}{2}f''_j - \frac{27h^3}{6}f'''_j + \frac{81}{24}h^4f^{iv}_j + O(h^5) \right)^2 \\
&\quad + \frac{13}{108} \left(6f_j - 15(f_j - hf'_j + \frac{h^2}{2}f''_j - \frac{h^3}{6}f'''_j + \frac{h^4}{24}f^{iv}_j) + O(h^5) \right) \\
&\quad + 12 \left(f_j - 2hf'_j + \frac{4h^2}{2}f''_j - \frac{8h^3}{6}f'''_j + \frac{16}{24}h^4f^{iv}_j + O(h^5) \right) \\
&\quad - 3 \left(f_j - 3hf'_j + \frac{9h^2}{2}f''_j - \frac{27h^3}{6}f'''_j + \frac{16}{24}h^4f^{iv}_j + O(h^4) \right)^2 \\
&\quad + \frac{1043}{960} \left(f_j - 3(f_j - hf'_j + \frac{h^2}{2}f''_j - \frac{h^3}{6}f'''_j + \frac{h^4}{24}f^{iv}_j) + O(h^5) \right) \\
&\quad + 3 \left(f_j - 2hf'_j + \frac{4h^2}{2}f''_j - \frac{8h^3}{6}f'''_j + \frac{16}{24}h^4f^{iv}_j + O(h^5) \right) \\
&\quad - \left(f_j - 3hf'_j + \frac{9h^2}{2}f''_j - \frac{27h^3}{6}f'''_j + \frac{81}{24}h^4f^{iv}_j + O(h^5) \right)^2 \\
&\quad + \frac{1}{70} \left(11f_j - 18f_{j-1} + 9f_{j-2} - 2f_{j-3} \right) \left(f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3} \right) \\
&= \frac{1}{36} \left(6hf'_j - \frac{3}{2}h^4f^{iv}_j + O(h^5) \right)^2 + \frac{13}{108} \left(3h^2f''_j - \frac{11}{6}h^4f^{iv}_j + O(h^5) \right)^2 \\
&\quad + \frac{1043}{960} \left(h^3f'''_j - \frac{3}{2}h^4f^{iv}_j + O(h^5) \right)^2 \\
&\quad + \frac{1}{70} \left(6hf'_j - \frac{3}{2}h^4f^{iv}_j + O(h^5) \right) \left(h^3f'''_j - \frac{3}{2}h^4f^{iv}_j + O(h^5) \right).
\end{aligned}$$

De igual forma, obtenemos

$$\begin{aligned}
IS_{j+1} &= \frac{1}{36}(2f_{j+1} + 3f_j - 6f_{j-1} + f_{j-2})^2 \\
&\quad + \frac{13}{12}(f_{j+1} - 2f_j + f_{j-1})^2 \\
&\quad + \frac{1043}{960}(f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2})^2 \\
&\quad + \frac{1}{72}(2f_{j+1} + 3f_j - 6f_{j-1} + f_{j-2})(f_{j+1} - 3f_j + 3f_{j-1} - f_{j-2}) \\
&= \frac{1}{36} \left(6hf'_j + \frac{1}{2}h^4f_j^{iv} + O(h^5) \right)^2 + \frac{13}{12} \left(h^2f_j'' + \frac{1}{12}h^4f_j^{iv} + O(h^5) \right)^2 \\
&\quad + \frac{1043}{960} \left(h^3f_j''' - \frac{1}{2}h^4f_j^{iv} + O(h^5) \right)^2 \\
&\quad + \frac{1}{70} \left(6hf'_j + \frac{1}{2}h^4f_j^{iv} + O(h^5) \right) \left(h^3f_j''' - \frac{1}{2}h^4f_j^{iv} + O(h^5) \right)
\end{aligned}$$

y

$$\begin{aligned}
IS_{j+2} &= \frac{1}{36}(-f_{j+2} + 6f_{j+1} - 3f_j - 2f_{j-1})^2 \\
&\quad + \frac{13}{12}(f_{j+2} - 2f_{j+1} + f_j)^2 \\
&\quad + \frac{1043}{960}(f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1})^2 \\
&\quad + \frac{1}{72}(-f_{j+2} + 6f_{j+1} - 3f_j - 2f_{j-1})(f_{j+2} - 3f_{j+1} + 3f_j - f_{j-1}) \\
&= \frac{1}{36} \left(6hf'_j - \frac{1}{2}h^4f_j^{iv} + O(h^5) \right)^2 + \frac{13}{12} \left(h^2f_j'' + h^3f_j''' + \frac{7}{12}h^4f_j^{iv} + O(h^5) \right)^2 \\
&\quad + \frac{1043}{960} \left(h^3f_j''' + \frac{1}{2}h^4f_j^{iv} + O(h^5) \right)^2 \\
&\quad + \frac{1}{70} \left(6hf'_j - \frac{1}{2}h^4f_j^{iv} + O(h^5) \right) \left(h^3f_j''' + \frac{1}{2}h^4f_j^{iv} + O(h^5) \right)
\end{aligned}$$

En los tres casos, si $f'_j \neq 0$ obtenemos $IS_{j+k} = D(1 + O(h^2))$ y, por tanto, obtendríamos el orden óptimo. En cambio, si $f'_j = f''_j = 0$ tan sólo tenemos $IS_{j+k} = D(1 + O(h))$ y según las Proposiciones 4.2 y 4.3 sólo podremos alcanzar $O(h^5)$. Esto motiva a buscar alguna alternativa para conseguir orden óptimo independientemente de si

$f'_j = f''_j = 0$ en algún punto. Este problema también aparece para $r = 3$ en medias en celda. En el Capítulo 6 planteamos modificar las hipótesis de las Proposiciones 4.2 y 4.3, de forma que si tomamos $\varepsilon = h^2$ podemos probar que también en el caso en que $f'_j = f''_j = 0$ obtenemos el orden óptimo.

5

Experimentos numéricos 1D para valores puntuales

5.1

Introducción

En este capítulo se muestran los resultados obtenidos mediante los distintos experimentos numéricos realizados con funciones unidimensionales.

En estos experimentos se utilizan los algoritmos de multirresolución en el contexto de valores puntuales con operadores reconstrucción calculados a partir de polinomios interpoladores de Lagrange a trozos.

Identificamos cada técnica de reconstrucción mediante los siguientes acrónimos.

- LIN - interpolación lineal centrada, en los intervalos donde el stencil centrado queda dentro del dominio computacional (Sección 1.4). En los intervalos cercanos a la frontera, utilizamos stencils hacia la izquierda o hacia la derecha, según el caso, de forma que se utilice sólo la información disponible (se puede encontrar más información sobre esta técnica en [10] y [23]).
- ENO - interpolación ENO, con selección jerárquica del stencil (Sección 1.5).
- SR - interpolación ENO con resolución subcelda y selección jerárquica del stencil (Sección 1.6).
- LIU - interpolación WENO con el indicador de Liu et al (Sección 4.4.1).
- SHU - interpolación WENO con el indicador de suavidad definido por Jiang y Shu (Sección 4.4.2).

A continuación se indica el número de nodos utilizados en la interpolación. En el caso de interpolación lineal, ENO y ENO-SR, indicaremos con 4 y 6 la interpolación hecha con stencils formados por 4 y 6 nodos, respectivamente. Para la interpolación WENO utilizaremos los valores 2 y 3 para indicar el número de stencils que intervienen en la combinación convexa que da el interpolante.

En esta serie de experimentos, los coeficientes de escala son truncados según la siguiente estrategia: dada una tolerancia ϵ , consideramos los parámetros de truncamiento $\epsilon_k = \epsilon$ en cada nivel de resolución. Además, con el objetivo de asegurar la estabilidad de la interpolación ENO y ENO-SR, utilizamos la técnica de control del error (Algoritmo 1.1).

Al igual que en el Capítulo 3, para analizar la capacidad de compresión de cada una de las técnicas de reconstrucción, lo que haremos será representar la localización de los coeficientes de escala que se encuentran por encima de la tolerancia dada para cada nivel de resolución.

Tomaremos $J_L = 1024$ y $J_0 = 8$.

5.2

Experimentos numéricos

Pasamos ahora al primer grupo de experimentos, consistente en la compresión de los datos discretos asociados a una función con una discontinuidad de salto.

Función con discontinuidad de salto

La función que consideramos, que se muestra en la Figura 5.1, es la siguiente:

$$f(x) = \begin{cases} 0,5 \sin(\pi x), & 0 \leq x \leq 2/3 \\ -\sin(\pi x) + 0,5, & 2/3 < x \leq 1 \end{cases} \quad (5.1)$$

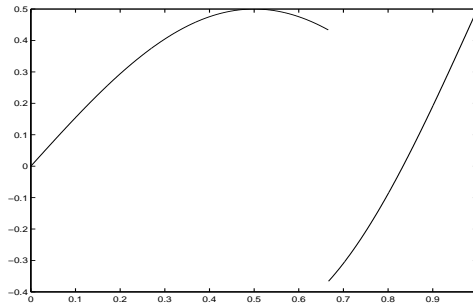


Figura 5.1: Función con discontinuidad de salto.

Consideramos como datos discretos los valores puntuales de la función (5.1). Como se vio en la Sección 1.4, la interpolación lineal basada en stencils centrados formados por $2r$ nodos resulta en regiones de poca exactitud formadas por $2r - 1$ intervalos. Esto se puede apreciar en las Figuras 5.2(a)-(b) y 5.4(a)-(b), donde se muestra la localización de los coeficientes de escala obtenidos con la reconstrucción lineal para 4 y 6 nodos. En el primer caso se observan tres intervalos de poca exactitud en cada nivel de resolución, mientras que para 6 nodos hay cinco intervalos afectados por la discontinuidad en cada escala.

Por otra parte, vemos que aunque para $\epsilon = 10^{-2}$ no existe mucha diferencia entre los resultados obtenidos con las interpolaciones de cuarto y sexto orden, si reducimos la tolerancia la diferencia es más significativa, y la interpolación de sexto orden proporciona una mayor exactitud con un número similar de detalles en la descomposición, por lo que su capacidad de compresión será mayor que la de la interpolación de cuarto orden.

Si comparamos estos resultados con los obtenidos para la técnica ENO (Figuras 5.2(c)-(d) y 5.4(c)-(d)) podemos ver, como se explicó en la Sección 1.5, que la región afectada por la discontinuidad se reduce a un intervalo, que es el que contiene dicha discontinuidad, independientemente del orden de interpolación. Como en el caso lineal, las diferencias entre distintos órdenes de interpolación no son visibles a menos que tomemos tolerancias pequeñas.

Por otra parte, si consideramos la técnica ENO-SR, vemos en las Figuras 5.2(e)-(f) y 5.4(e)-(f) que los resultados obtenidos coinciden con los de la técnica ENO. Esto es debido a que los valores puntuales de una función con discontinuidad de salto no proporcionan ninguna información sobre la localización de la discontinuidad. En el contexto de medias en celda, este salto corresponderá a una esquina en la función primitiva de f , por lo que la técnica SR sí que introducirá mejoras en la reconstrucción.

Por último, consideramos los resultados obtenidos a partir de la técnica WENO, que se muestran en las Figuras 5.3 y 5.5.

Observamos que cuando utilizamos la definición de Jiang y Shu los resultados mejoran respecto a los obtenidos con la definición de Liu et al. Además, la región afectada por la discontinuidad está formada por un intervalo en cada nivel de resolución, con lo que comprobamos que el interpolante WENO emula el comportamiento ENO frente a las singularidades. Comparando estos resultados con los correspondientes a las técnicas anteriores vemos que cuando se utilizan 4 nodos en la construcción del polinomio interpolador, el interpolante WENO (que alcanza un orden de exactitud teórico igual a 6) proporciona una mayor exactitud que la reconstrucción ENO de cuarto orden, pero no llega a igualar los resultados obtenidos con la interpolación ENO de sexto orden, cuando la tolerancia es pequeña.

Para acabar con el análisis de la discontinuidad de salto, en la

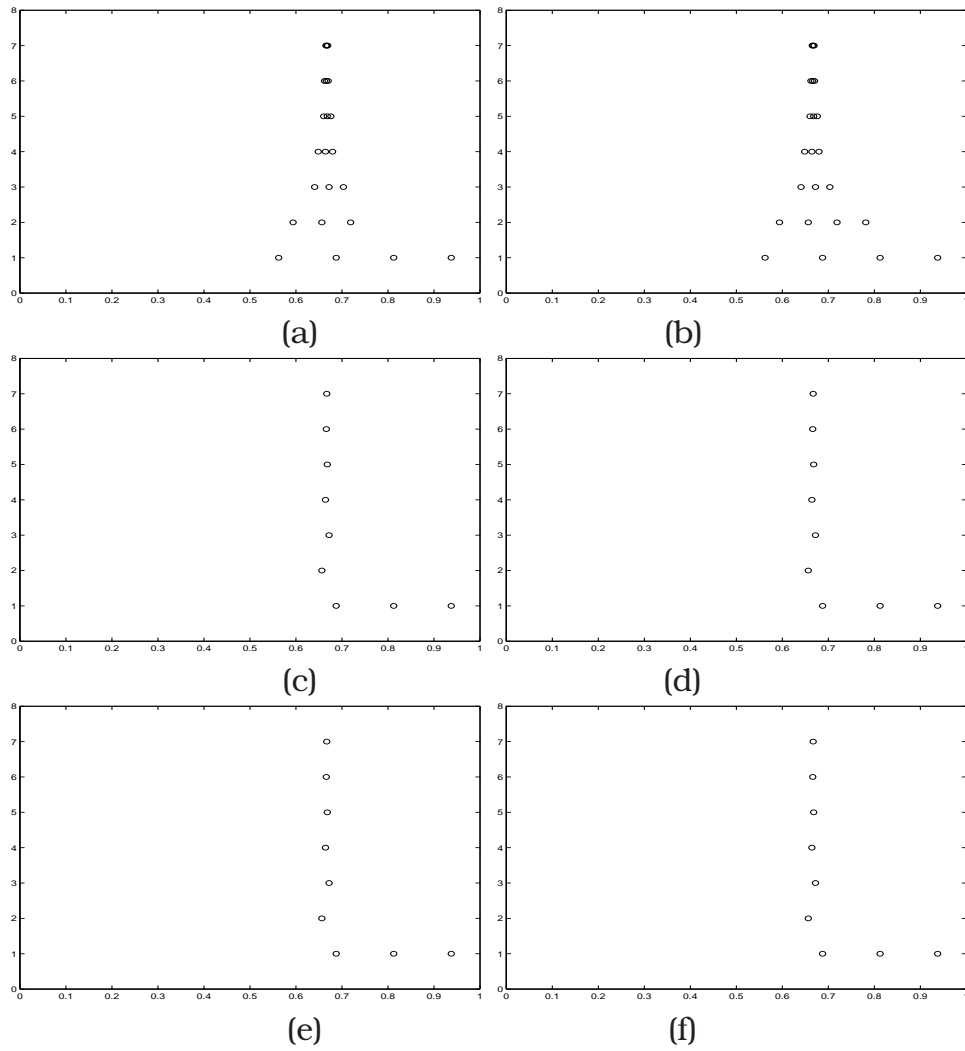


Figura 5.2: Función con discontinuidad de salto: (a) LIN4, (b) LIN6, (c) ENO4, (d) ENO6, (e) SR4, (f) SR6; $\epsilon = 10^{-2}$.

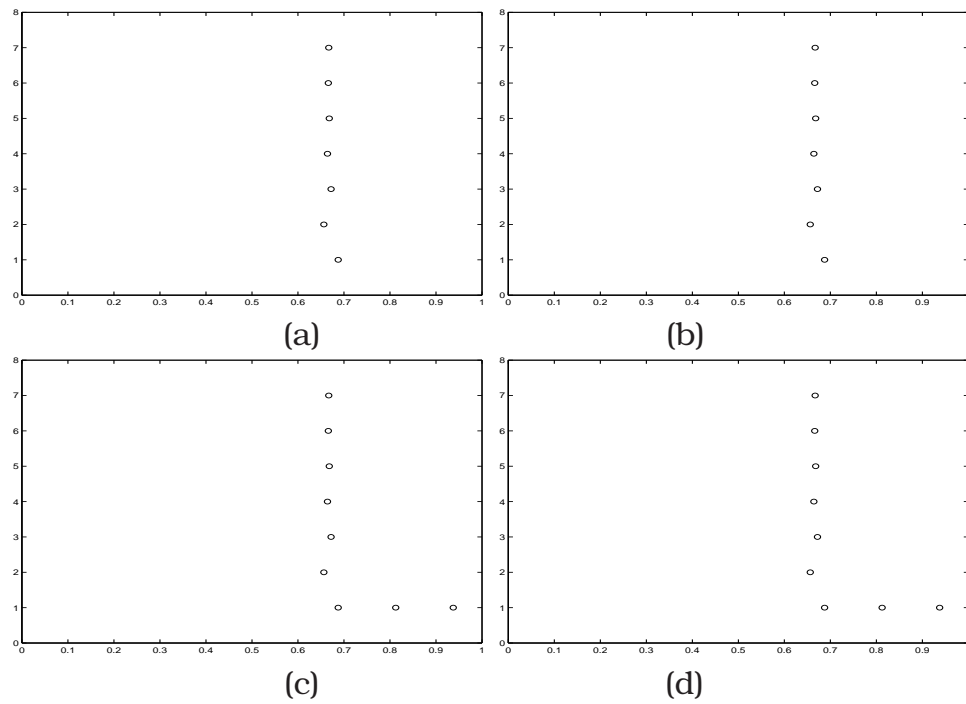


Figura 5.3: Función con discontinuidad de salto: (a) LIU2, (b) SHU2, (c) LIU3, (d) SHU3; $\epsilon = 10^{-2}$.

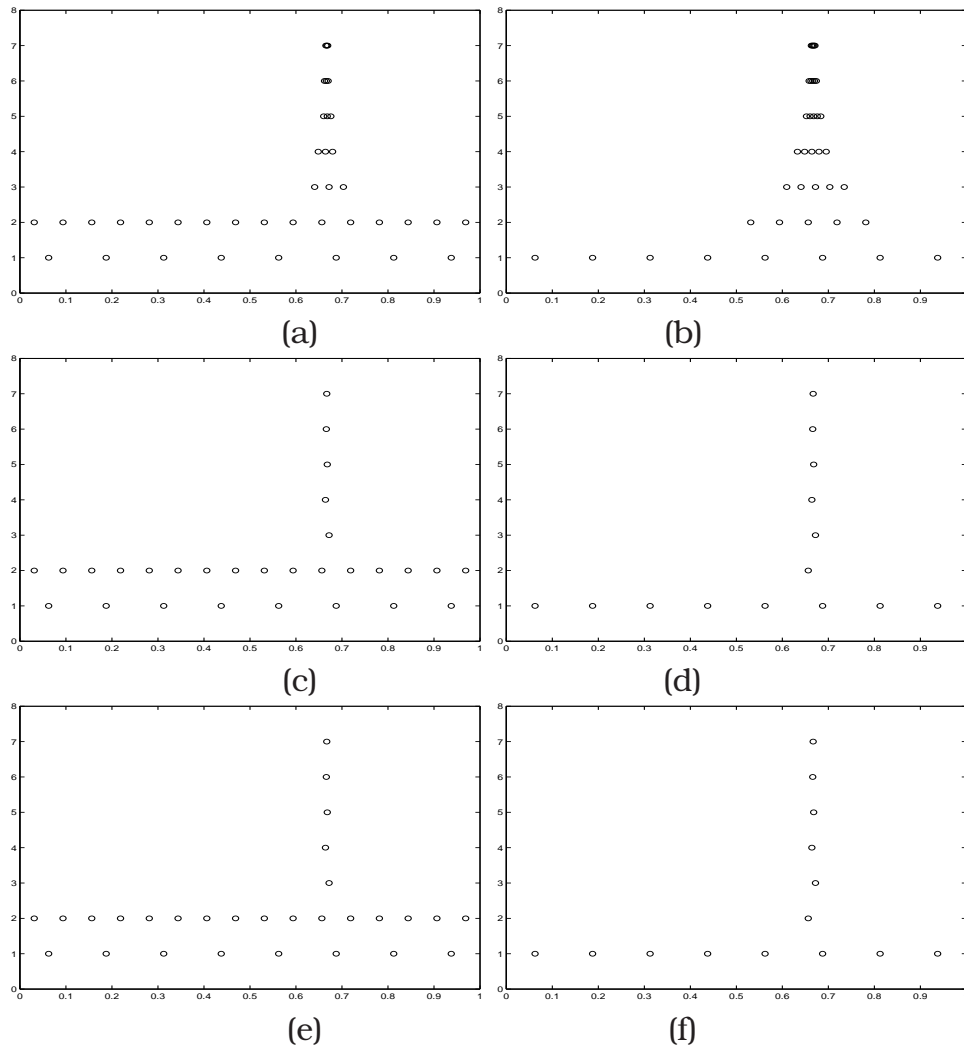


Figura 5.4: Función con discontinuidad de salto: (a) LIN4, (b) LIN6, (c) ENO4, (d) ENO6, (e) SR4, (f) SR6; $\epsilon = 5 \cdot 10^{-6}$.

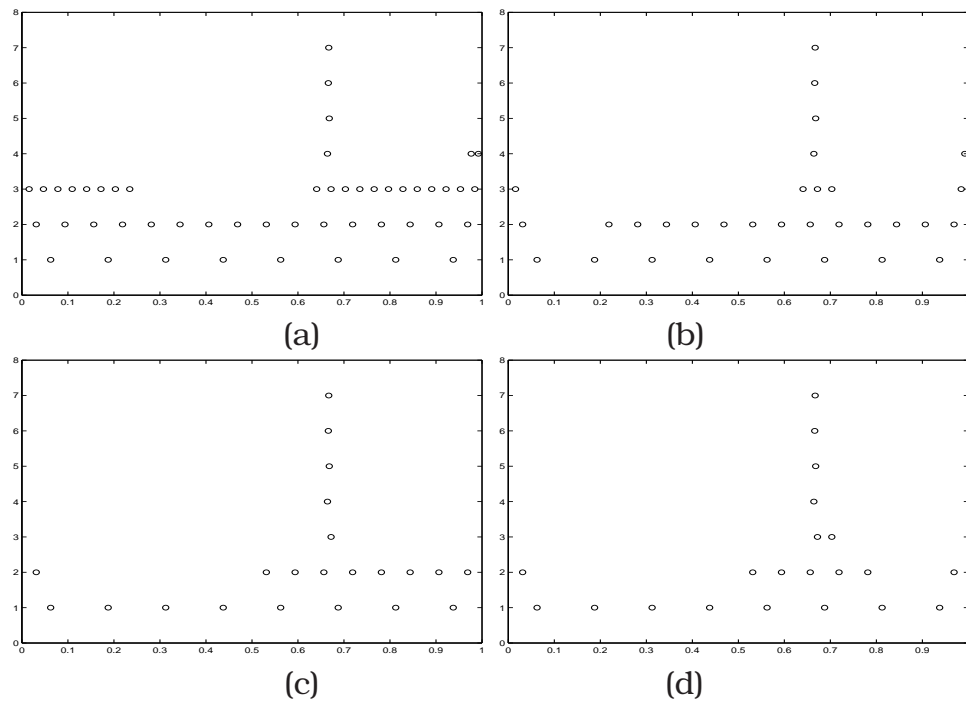


Figura 5.5: Función con discontinuidad de salto: (a) LIU2, (b) SHU2, (c) LIU3, (d) SHU3; $\epsilon = 5 \cdot 10^{-6}$.

Tabla 5.1 se muestran de forma resumida los resultados obtenidos con las distintas técnicas de reconstrucción, para los dos valores de la tolerancia considerados, $\epsilon = 10^{-2}$ y $\epsilon = 5 \cdot 10^{-6}$. Para cada uno de estos valores se indica el valor del error de la reconstrucción evaluado en las normas $\|\cdot\|_\infty$, $\|\cdot\|_1$ y $\|\cdot\|_2$, así como el número de coeficientes de escala que quedan después de aplicar la estrategia de truncamiento.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	<i>nnz</i>	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	<i>nnz</i>
LIN4	2.7e-4	6.6e-5	1.1e-4	22	1.6e-6	4.1e-7	5.4e-7	39
ENO4	4.8e-4	1.2e-4	1.7e-4	9	2.7e-6	5.8e-7	7.4e-7	29
SR4	4.8e-4	1.2e-4	1.7e-4	9	2.7e-6	5.8e-7	7.4e-7	29
LIU2	3.6e-3	9.3e-4	1.4e-3	7	5.0e-6	8.9e-7	1.3e-6	50
SHU2	3.6e-3	7.1e-4	1.2e-3	7	4.1e-6	6.9e-7	9.7e-7	32
LIN6	9.8e-3	1.4e-3	2.9e-3	23	5.6e-7	6.9e-8	1.2e-7	38
ENO6	3.9e-5	7.9e-6	1.3e-5	9	7.2e-6	1.5e-7	2.2e-7	14
SR6	3.9e-5	7.9e-6	1.3e-5	9	7.2e-6	1.5e-7	2.2e-7	14
LIU3	5.2e-4	8.1e-5	1.4e-4	9	4.6e-6	9.0e-7	1.5e-6	22
SHU3	5.4e-4	8.6e-5	1.5e-4	9	2.0e-6	2.5e-7	4.6e-7	21

Tabla 5.1: Errores y y elementos distintos de cero (*nnz*) para la función con discontinuidad de salto, $\epsilon = 10^{-2}$ (izquierda) y $\epsilon = 5 \cdot 10^{-6}$ (derecha).

Función con esquina

Consideramos ahora para nuestros experimentos una función con una discontinuidad de salto en su primera derivada. Esta función se muestra en la Figura 5.6 y su expresión viene dada por:

$$f(x) = \left| \sin \left(\frac{3}{2}\pi \left(x - \frac{1}{3} \right) \right) \right| \tag{5.2}$$

La función (5.2) presenta una esquina en $x = 1/3$. El efecto de esta singularidad cuando trabajamos con valores puntuales es más débil que el efecto de un salto, como se puede ver en las Figuras 5.7, 5.8, 5.9 y 5.10.

En el caso de interpolación lineal la región afectada por la singularidad es mayor que para las técnicas no lineales, aunque en

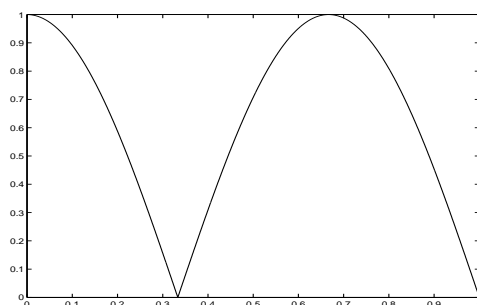


Figura 5.6: Función con esquina.

este caso no se produce un aumento del número de intervalos que forman esta región cuando aumentamos el orden de la interpolación, si tomamos un parámetro de truncamiento $\epsilon = 10^{-2}$.

Para la técnica ENO observamos un comportamiento similar al presentado ante la discontinuidad de salto, excepto cuando reducimos la tolerancia. En este caso, el proceso de selección del stencil resulta en cierto modo “engañado” por la singularidad, y observamos un efecto similar al que se produce cuando utilizamos interpolación lineal.

Con la técnica SR vemos que el efecto de la singularidad queda anulado, debido a que en este caso se trata de una esquina, que proporciona información para localizar la singularidad.

Por último, por lo que respecta a los resultados obtenidos utilizando la técnica WENO, vemos que el indicador de suavidad definido por Liu et al. reconoce la singularidad, mientras que con el indicador definido por Jiang y Shu se produce la misma situación que con la reconstrucción ENO. Comparando con las otras técnicas no lineales, observamos el mismo comportamiento que con la discontinuidad de salto, es decir, que con interpolación basada en 4 nodos, el interpolante WENO mejora los resultados obtenidos con ENO, sin llegar a equipararse a la reconstrucción ENO de orden 6, pero la técnica ENO-SR continúa proporcionando mejores resultados que la técnica WENO.

En la Tabla 5.2 se resumen los resultados anteriores, junto con los errores en las reconstrucciones obtenidas con las distintas técnicas de interpolación.

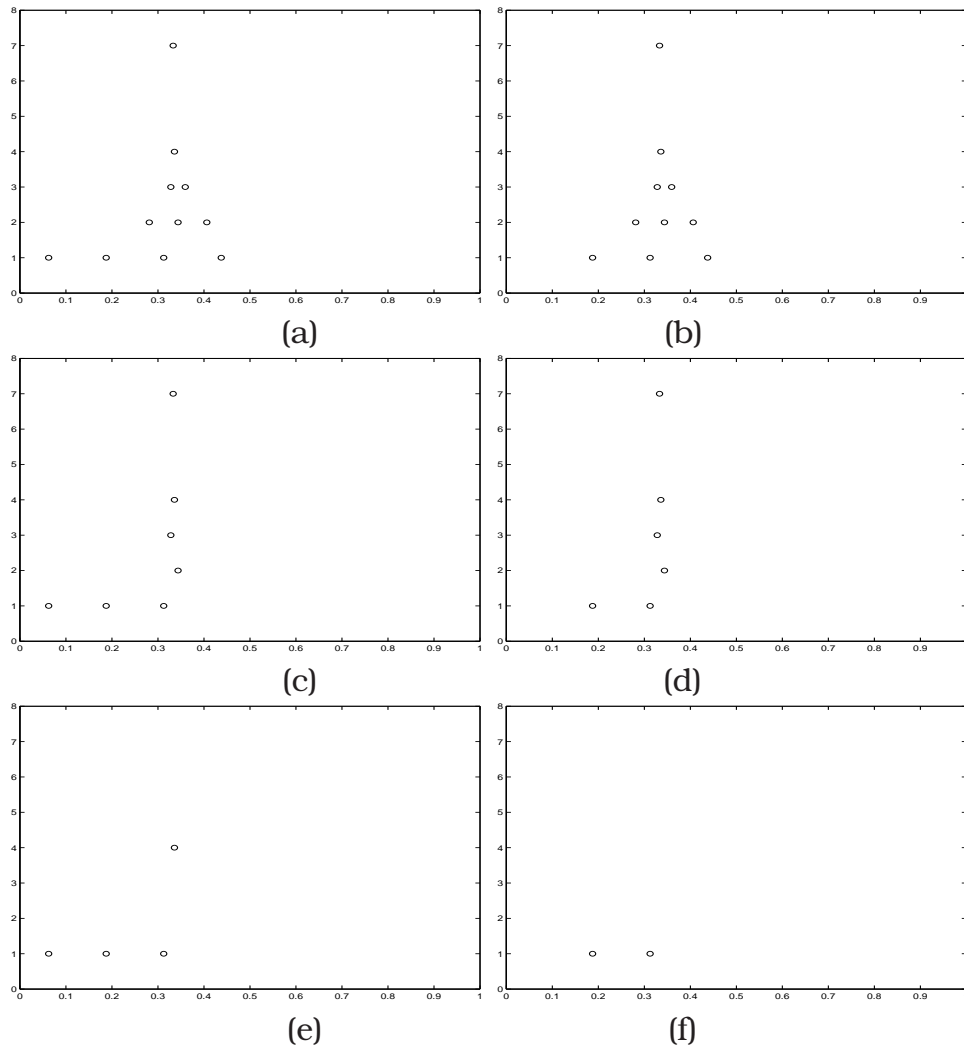


Figura 5.7: Función con esquina: (a) LIN4, (b) LIN6, (c) ENO4, (d) ENO6, (e) SR4, (f) SR6; $\epsilon = 10^{-2}$.

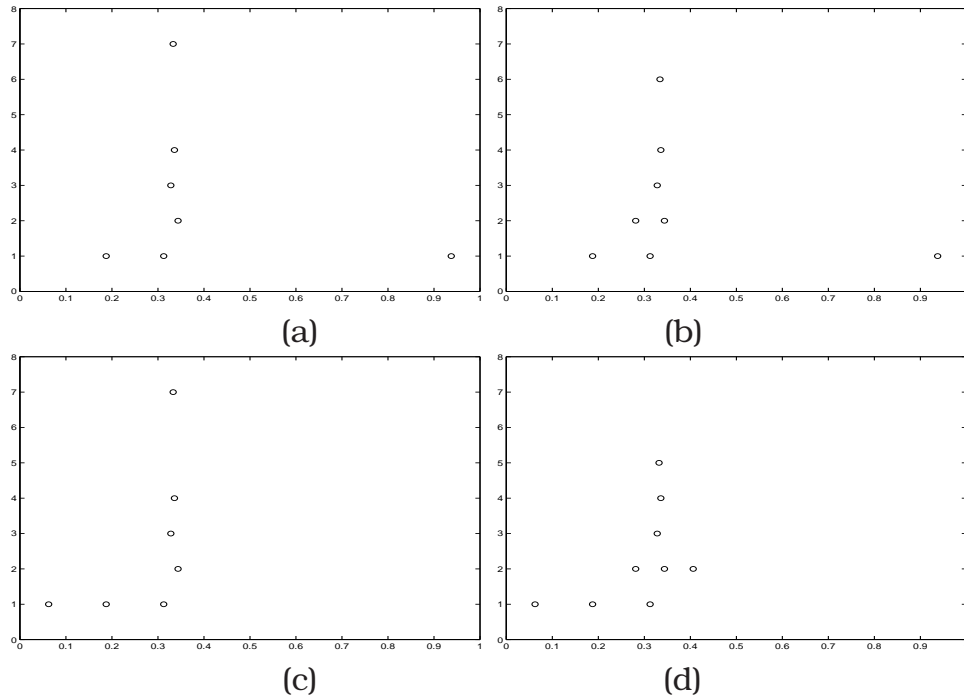


Figura 5.8: Función con discontinuidad de esquina: (a) LIU2, (b) SHU2, (c) LIU3, (d) SHU3; $\epsilon = 10^{-2}$.

	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _2$	nnz
LIN4	8.5e-3	1.0e-3	1.6e-3	11	4.9e-6	6.7e-7	1.2e-6	61
ENO4	8.7e-3	1.2e-3	1.7e-3	7	4.9e-6	8.4e-7	1.4e-6	55
SR4	8.4e-3	1.4e-3	1.9e-3	4	4.9e-6	8.5e-7	1.4e-6	52
LIU2	9.8e-3	2.7e-3	3.7e-3	7	4.7e-6	8.8e-7	1.3e-6	83
SHU2	9.4e-3	2.1e-3	2.9e-3	8	4.8e-6	4.3e-7	7.8e-7	65
LIN6	8.8e-3	1.5e-3	2.5e-3	10	3.4e-6	9.5e-7	1.4e-6	40
ENO6	9.0e-3	5.6e-4	1.1e-3	6	3.4e-6	1.3e-6	1.7e-6	18
SR6	4.1e-3	6.1e-4	1.1e-3	2	3.4e-6	1.4e-6	1.7e-6	14
LIU3	8.7e-3	8.1e-4	1.3e-3	7	4.9e-6	8.9e-7	1.3e-6	32
SHU3	8.5e-3	9.2e-4	1.7e-3	9	4.5e-6	8.2e-7	1.5e-6	44

Tabla 5.2: Errores y elementos distintos de cero (nnz) para la función con discontinuidad con esquina, $\epsilon = 10^{-2}$ (izquierda) y $\epsilon = 5 \cdot 10^{-6}$ (derecha).

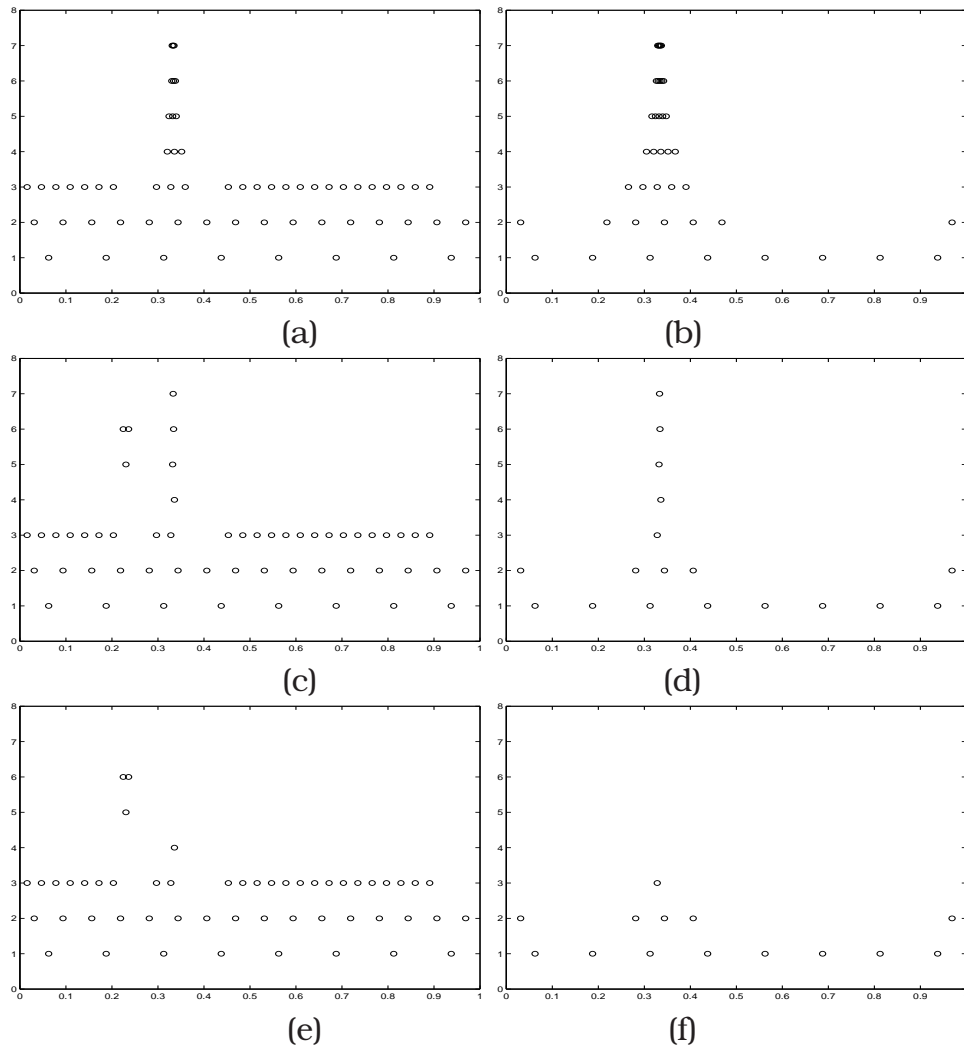


Figura 5.9: Función con esquina: (a) LIN4, (b) LIN6, (c) ENO4, (d) ENO6, (e) SR4, (f) SR6; $\epsilon = 5 \cdot 10^{-6}$.

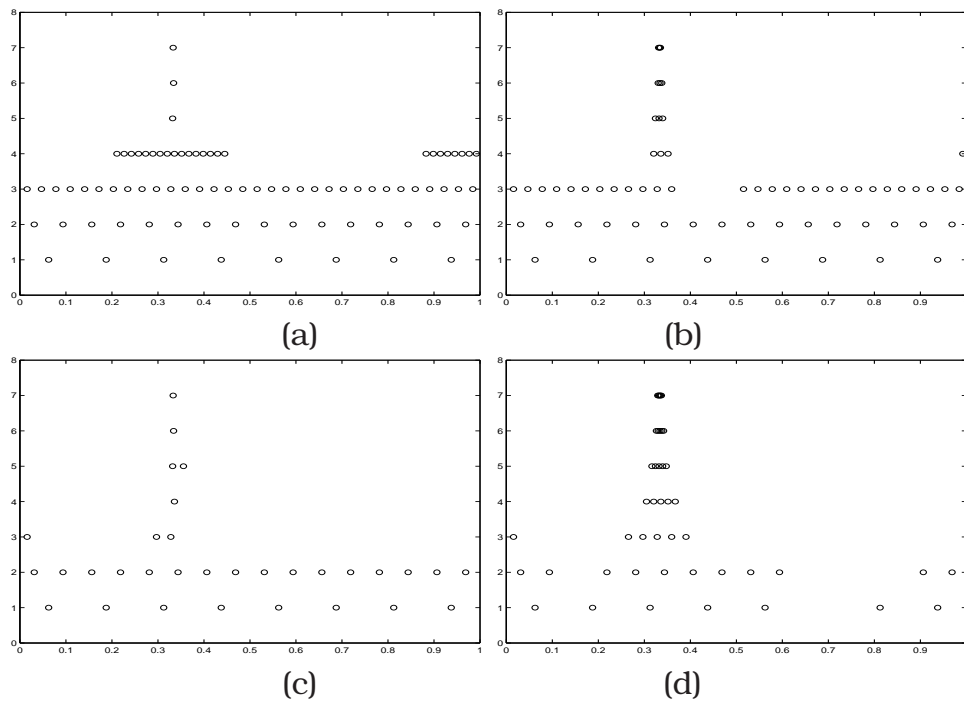


Figura 5.10: Función con discontinuidad de esquina: (a) LIU2, (b) SHU2, (c) LIU3, (d) SHU3; $\epsilon = 5 \cdot 10^{-6}$.

6

Resultados sobre WENO para valores puntuales

6.1

Introducción

Este Capítulo puede entenderse como una continuación del Capítulo 4 sobre WENO para valores puntuales. Aquí, por una parte, introducimos tres nuevos indicadores de suavidad. También presentamos un marco unificado que permite, esencialmente usando las propiedades de los interpoladores de Lagrange, probar los siguientes resultados para cualquier r :

1. El orden de la interpolación es $2r$ en regiones suaves. Esto es cierto incluso si en algún punto $f'_j = f''_j = 0$. Recordemos que según hemos visto al final del Capítulo 4 en este caso no se

deducía el orden óptimo.

2. El orden de la interpolación es $r + 1$, el mismo que se obtiene con los interpoladores ENO, cuando la función tiene una discontinuidad en el stencil de $2r$ puntos pero es suave en al menos uno de los substencils de $r + 1$ puntos.
3. Los pesos ideales se obtienen de forma cerrada.

Todos estos resultados han sido obtenidos utilizando un estudio cuidadoso que destaca la importancia de la elección del parámetro ε , que aparece en la definición de los pesos para evitar denominadores nulos, como $\varepsilon = h^2$.

6.2

Interpolación WENO

El objetivo de esta sección es definir un operador *predicción* (1.13) con un orden alto de exactitud en las regiones suaves de la función f .

La técnica de interpolación WENO apareció como una mejora sobre las reconstrucciones ENO de valores de puntuales a partir de medias en celda, utilizados para esquemas HRSC para leyes de conservación. Usando los stencils de $r + 1$ nodos, las reconstrucciones ENO proporcionan un orden de exactitud de $r + 1$ excepto en aquellos subintervalos que contienen singularidades. Esto se consigue después de un proceso de selección del stencil en el cual se escoge el stencil en el cual la función interpolada es “más suave” en algún sentido. Este proceso de selección es sensible al error de redondeo y puede conducir a la elección de un stencil que no sea el más conveniente.

Por otra parte, en vez de considerar sólo un stencil conteniendo $r + 1$ nodos, podríamos tomar la información proporcionada por los $2r$ nodos que participan en el proceso de selección y obtener un orden de exactitud $2r$ en las regiones donde la función es suave. Si denotamos $S_{j+k}, k = 0, \dots, r - 1$, los r stencils candidatos que

contienen x_{j-1}, x_j :

$$S_{j+k} = \{x_{j+k-r}, \dots, x_{j+k}\}, \quad k = 0, \dots, r-1 \quad (6.1)$$

y $p_{j+k}^r(x)$ el polinomio interpolador definido en el stencil S_{j+k} , entonces el interpolante WENO viene dado por una combinación convexa:

$$q_j(x) = \sum_{k=0}^{r-1} \omega_k^j p_{j+k}^r(x) \text{ donde } \omega_k^j \geq 0, k = 0, \dots, r-1, \quad \sum_{k=0}^{r-1} \omega_k^j = 1, \quad (6.2)$$

y el correspondiente operador predicción viene dado por:

$$\mathcal{I}(x_{j-\frac{1}{2}}^{l-1}; f^{l-1}) = \sum_{k=0}^{r-1} \omega_k^j p_{j+k}^r(x_{j-\frac{1}{2}}^{l-1}). \quad (6.3)$$

Como podemos considerar el nivel de resolución l como fijo, en el resto de la sección eliminaremos el superíndice $l-1$ para facilitar la notación y simplemente denotaremos h, x_j, f_j , en vez de $h^{l-1}, x_j^{l-1}, f_j^{l-1}$.

Los pesos deberían ser seleccionados con el objetivo de alcanzar el orden de exactitud $2r$ en regiones donde la función es suave. Siguiendo las ideas originales de [33], brevemente recordamos que esta exactitud puede ser alcanzada basada en una aproximación

$$p_{j+r-1}^{2r-1}(x_{j-\frac{1}{2}}) = f(x_{j-\frac{1}{2}}) + \mathcal{O}(h^{2r}), \quad (6.4)$$

por el interpolador p_{j+r-1}^{2r-1} definido por el stencil $\{x_{j-r}, \dots, x_{j+r-1}\}$, siempre que f sea suave en la región determinada por dicho stencil. Por otra parte, tenemos r aproximaciones de orden $r+1$

$$p_{j+k}^r(x_{j-\frac{1}{2}}) = f(x_{j-\frac{1}{2}}) + \mathcal{O}(h^{r+1}), \quad (6.5)$$

basadas en los stencils S_{j+k} , $k = 0, \dots, r-1$. Como en WENO para medias en celda, lo primero que haremos será determinar los pesos óptimos, $C_i^r \geq 0, \forall i$ y $\sum_{i=0}^{r-1} C_i^r = 1$ tales que

$$p_{j+r-1}^{2r-1}(x_{j-\frac{1}{2}}) = \sum_{i=0}^{r-1} C_i^r p_{j+k}^r(x_{j-\frac{1}{2}}). \quad (6.6)$$

El siguiente resultado nos da fórmulas explícitas para dichos pesos óptimos.

Proposición 6.1. *Los pesos óptimos vienen dados por*

$$C_i^r = \frac{1}{2^{2r-1}} \binom{2r}{2i+1}, \quad i = 0, \dots, r-1.$$

Demostración. Los pesos óptimos deben verificar

$$p_{j+r-1}^{2r-1}(x_{j-\frac{1}{2}}) = \sum_{i=0}^{r-1} C_i^r p_{j+i}^r(x_{j-\frac{1}{2}}), \quad (6.7)$$

para cualquier función f y cualquier r . Usando un cambio de variables adecuado, podemos asumir $j = 0$, $x_{-\frac{1}{2}} = 0$ y $x_k = 2k + 1$, $k = -r, \dots, r-1$. Utilizando la base de Lagrange para cada polinomio interpolador, obtenemos

$$p_{r-1}^{2r-1} = \sum_{k=-r}^{r-1} f_k L_k, \quad L_k = \prod_{l=-r, l \neq k}^{r-1} \frac{x - (2l + 1)}{2k + 1 - (2l + 1)}; \quad (6.8)$$

$$p_i^r = \sum_{k=r-i}^i f_k L_k^i, \quad L_k^i = \prod_{l=i-r, l \neq k}^i \frac{x - (2l + 1)}{2k + 1 - (2l + 1)}. \quad (6.9)$$

De (6.7), (6.8) y (6.9) obtenemos

$$\begin{aligned} \sum_{k=-r}^r f_k L_k(0) &= \sum_{i=0}^{r-1} C_i^r \sum_{k=r-i}^i f_k L_k^i(0) \\ &= \sum_{k=-r}^k f_k \sum_{i=\max(k,0)}^{\min(k+r, r-1)} C_i^r L_k^i(0). \end{aligned} \quad (6.10)$$

Igualando los coeficientes de f_k en (6.10) deducimos las ecuaciones equivalentes:

$$\sum_{i=\max(k,0)}^{\min(k+r, r-1)} C_i^r L_k^i(0) = L_k(0), \quad k = -r, \dots, r-1 \quad (6.11)$$

De (6.8) y (6.9) podemos calcular:

$$L_k(0) = \left(-\frac{1}{2}\right)^{2r-1} \prod_{l=-r, l \neq k}^{r-1} \frac{2l+1}{k-l} = \frac{2r-1}{2k+1} \binom{2r-2}{r-1} \binom{2r-1}{k+r} \frac{(-1)^k}{2^{4r-3}}$$

$$L_k^i(0) = \left(-\frac{1}{2}\right)^r \prod_{l=i-r, l \neq k}^i \frac{2l+1}{k-l} = \frac{2r-1}{2k+1} \binom{2r-2}{r-1} \frac{\binom{r-1}{i} \binom{r}{r-i+k}}{\binom{2r}{2i+1}} \frac{(-1)^k}{2^{2r-2}},$$

lo que nos da las ecuaciones siguientes, después de sustituir en (6.11):

$$\sum_{i=\max(k,0)}^{\min(k+r,r-1)} \binom{r-1}{i} \binom{r}{r-i+k} = \binom{2r-1}{k+r}, \quad k = -r, \dots, r-1$$

$$\sum_{i=\max(m-r,0)}^{\min(m,r-1)} \binom{r-1}{i} \binom{r}{m-i} = \binom{2r-1}{m}, \quad m = 0, \dots, 2r-1.$$

La última ecuación se satisface debido a propiedades elementales de combinatoria.

Por otra parte, como $\binom{2r}{2i+1} = \binom{2r-1}{2i} + \binom{2r-1}{2i+1}$, deducimos:

$$\sum_{i=0}^{r-1} C_i^r = \frac{1}{2^{2r-1}} \sum_{i=0}^{r-1} \binom{2r}{2i+1} = \frac{1}{2^{2r-1}} \sum_{i=0}^{2r-1} \binom{2r-1}{i} = 1$$

□

Los pesos óptimos para $r = 2, 3, 4, 5$, obtenidos de la Proposición 6.1, aparecen en la Tabla 6.1.

	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$r = 2$	1/2	1/2	-	-	-
$r = 3$	3/16	10/16	3/16	-	-
$r = 4$	1/16	7/16	7/16	1/16	-
$r = 5$	5/256	60/256	126/256	60/256	5/256

Tabla 6.1: Pesos óptimos para $r = 2, 3, 4, 5$.

6.3

Diseño de pesos óptimos no lineales

Como en el artículo sobre WENO [33], la idea clave consiste en fijarse en que si se definen pesos no lineales ω_k^j que satisfacen la condición

$$\omega_k^j = C_k^r + O(h^m), \quad k = 0, \dots, r-1 \quad (6.12)$$

con $m \leq r-1$, entonces podemos probar (ver [33] o la proposición siguiente), que

$$f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) = O(h^{r+m+1}). \quad (6.13)$$

Notar que si $m = r-1$ en (6.12), entonces el orden de la aproximación de (6.13) es el mismo que el obtenido por el interpolador p_{j+r-1}^{2r-1} , basado en todos los nodos implicados en el cálculo.

Otro objetivo del diseño de los pesos es la emulación de la interpolación ENO, en que los polinomios correspondientes a subintervalos donde la función tiene una singularidad deberían tener una influencia insignificante en la combinación lineal, es decir, sus pesos en la combinación deberían ser muy pequeños. Además, el orden de aproximación alcanzado con la interpolación WENO no debe ser peor que el de la interpolación ENO. Por otra parte, los pesos deberían ser funciones suaves de los valores puntuales de la función interpolada y poder ser calculados de una forma eficiente.

Con el objetivo de satisfacer dichas condiciones, los pesos se definen, de igual forma que en [33], como sigue:

$$\omega_k^j = \frac{\alpha_k^j}{\sum_{i=0}^{r-1} \alpha_i^j}, \quad k = 0, \dots, r-1 \quad \text{donde} \quad \alpha_k^j = \frac{C_k^r}{(\varepsilon_h + I_k^j)^t}, \quad (6.14)$$

donde los valores positivos ε_h (pequeños y posiblemente dependientes de h) son, en principio, introducidos para evitar denominadores nulos e I_k^j es un indicador de la suavidad de la función f en el stencil S_{j+k} . Los pesos definidos así verifican $\sum_k \omega_k = 1$ independientemente de los indicadores de suavidad elegidos.

Omitimos el superíndice j , que indica el centro de los *stencils* considerados para el cálculo de los pesos, para clarificar la exposición de la siguiente proposición, en la que se tratan algunas condiciones que deben satisfacer los indicadores de suavidad para obtener las propiedades de la interpolación WENO.

Proposición 6.2. *Sea I_k el indicador de suavidad de f en S_{j+k} que verifica:*

1. $I_k = \mathcal{O}(h^2)$ si f es suave en S_{j+k} ;
2. $I_k \not\rightarrow 0, h \rightarrow 0$, si f no es suave en S_{j+k} y
3. para algún $m \geq r - 1$,

$$I_l - I_k = \mathcal{O}(h^{m+2}), \quad \forall 0 \leq k, l < r, \quad (6.15)$$

siempre que f sea suave en todos los *stencils* S_{j+k} , $k = 0, \dots, r - 1$.

Entonces, si tomamos $\varepsilon_h = h^2$,

$$\begin{aligned} \omega_l &= C_l^r (1 + \mathcal{O}(h^m)) \\ f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) &= \mathcal{O}(h^{2r}), \end{aligned} \quad (6.16)$$

en regiones donde f es suave en todos los *stencils* y

$$f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) = \mathcal{O}(h^{\min(2t, r+1)}), \quad (6.17)$$

si f es suave en al menos uno de los *stencils* S_{j+k} , $k = 0, \dots, r - 1$.

Demostración. Tenemos

$$\begin{aligned}
\frac{I_l - I_k}{\varepsilon_h + I_k} \sum_{i=0}^{t-1} \left(\frac{\varepsilon_h + I_l}{\varepsilon_h + I_k} \right)^i &= \frac{I_l - I_k}{\varepsilon_h + I_k} \frac{\left(\frac{\varepsilon_h + I_l}{\varepsilon_h + I_k} \right)^t - 1}{\frac{\varepsilon_h + I_l}{\varepsilon_h + I_k} - 1} \\
&= \left(\frac{\varepsilon_h + I_l}{\varepsilon_h + I_k} \right)^t - \left(\frac{\varepsilon_h + I_l}{\varepsilon_h + I_l} \right)^t \\
&= \frac{1}{(\varepsilon_h + I_k)^t} - \frac{1}{(\varepsilon_h + I_l)^t} \\
&= \frac{1}{(\varepsilon_h + I_l)^t} \quad (6.18)
\end{aligned}$$

Si f es suave en todos los *stencils* y asumimos las hipótesis de la Proposición, tenemos $I_l - I_k = \mathcal{O}(h^{m+2})$, $\varepsilon_h + I_l = \mathcal{O}(h^2)$ y $\varepsilon_h + I_k = \mathcal{O}(h^2)$, de donde:

$$\frac{I_l - I_k}{\varepsilon_h + I_k} = \mathcal{O}(h^m) \text{ y } \frac{\varepsilon_h + I_l}{\varepsilon_h + I_k} = \mathcal{O}(1),$$

lo cual, junto con (6.18), nos da

$$\frac{1}{(\varepsilon_h + I_k)^t} = \frac{1 + \mathcal{O}(h^m)}{(\varepsilon_h + I_l)^t}. \quad (6.19)$$

Entonces, para un valor fijo l y cualquier $0 \leq k < r$,

$$\alpha_k = \frac{C_k^r}{(\varepsilon_h + I_k)^t} = \frac{C_k^r(1 + \mathcal{O}(h^m))}{(\varepsilon_h + I_l)^t},$$

de donde,

$$\begin{aligned}
\sum_{k=0}^{r-1} \alpha_k &= \frac{1}{(\varepsilon_h + I_l)^t} \sum_{k=0}^{r-1} C_k^r(1 + \mathcal{O}(h^m)) = \frac{1 + \mathcal{O}(h^m)}{(\varepsilon_h + I_l)^t}, \\
\omega_l &= \frac{\alpha_l}{\sum_{k=0}^{r-1} \alpha_k} = \frac{C_l^r / (\varepsilon_h + I_l)^t}{(1 + \mathcal{O}(h^m)) / (\varepsilon_h + I_l)^t} = C_l^r(1 + \mathcal{O}(h^m)). \quad (6.20)
\end{aligned}$$

Ahora, como en [33], por (6.4), (6.5), (6.2), (6.20), $r + m + 1 \geq 2r$

y $\sum_{k=0}^{r-1} (C_k^r - \omega_k) = 0$, obtenemos

$$\begin{aligned}
f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) &= f(x_{j-\frac{1}{2}}) - p_{j+r-1}^{2r-1}(x_{j-\frac{1}{2}}) + p_{j+r-1}^{2r-1}(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) \\
&= \mathcal{O}(h^{2r}) + \sum_{k=0}^{r-1} (C_k^r - \omega_k) p_{j+k}^r(x_{j-\frac{1}{2}}) \\
&= \mathcal{O}(h^{2r}) + \sum_{k=0}^{r-1} (C_k^r - \omega_k) (p_{j+k}^r(x_{j-\frac{1}{2}}) - f(x_{j-\frac{1}{2}})) \\
f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) &= \mathcal{O}(h^{2r}) + \sum_{k=0}^{r-1} \mathcal{O}(h^m) \mathcal{O}(h^{r+1}) = \mathcal{O}(h^{2r}). \tag{6.21}
\end{aligned}$$

Supongamos ahora que f tenga una singularidad en alguno pero no todos los *stencils* S_{j+k} , $k = 0, \dots, r-1$, y además asumimos $I_k \neq 0$ para los *stencils* donde f tiene una singularidad, siendo $I_k = \mathcal{O}(h^2)$ en cualquier otro caso, entonces

$$\alpha_k = \begin{cases} \mathcal{O}(1) & f \text{ no suave en } S_{j+k} \\ \mathcal{O}(h^{-2t}) & f \text{ suave en } S_{j+k} \end{cases}, \tag{6.22}$$

con lo que $\sum_{k=0}^{r-1} \alpha_k = \mathcal{O}(h^{-2t})$ y $\omega_k = \mathcal{O}(h^{2t})$ si S_{j+k} no es suave. Si denotamos $\mathcal{K} = \{k/f \text{ no suave en } S_{j+k}\}$, entonces

$$\begin{aligned}
f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) &= \sum_{k=0}^{r-1} \omega_k (f(x_{j-\frac{1}{2}}) - p_{j+k}^r(x_{j-\frac{1}{2}})) \\
&= \sum_{k \in \mathcal{K}} \mathcal{O}(h^{2t}) \mathcal{O}(1) + \sum_{k \notin \mathcal{K}} \mathcal{O}(1) \mathcal{O}(h^{r+1}) \\
&= \mathcal{O}(h^{\min(r+1, 2t)}). \tag{6.23}
\end{aligned}$$

□

Nota 6.1. Si $\varepsilon_h = h^2$ y $2t \geq r+1$, entonces (6.17) implica que el orden de exactitud del interpolante WENO es $r+1$, al menos tan bueno como el interpolante ENO. Desde el punto de vista teórico, la utilización en (6.14) de un ε_h fijo no evita que (6.16) sea cierto, pero sería peor que ENO en las regiones no suaves. En la práctica, se suele utilizar un ε_h fijo, $\varepsilon_h \approx 10^{-5}, 10^{-6}$. Esto verifica $\varepsilon_h \approx h^2$ para $h \approx 10^{-3}$, pero, como muestra el Ejemplo 6.1, la aproximación WENO con un ε_h fijo puede ser peor que la correspondiente aproximación ENO cerca de singularidades cuando $h \rightarrow 0$.

6.4

Indicadores de suavidad

En [27] Jiang y Shu, dentro del contexto de medias en celdas, definieron un indicador de suavidad, de la siguiente forma:

$$I_k^j = \sum_{l=1}^r \int_{x_{j-1}}^{x_j} h^{2l-1} (p_{j+k}^{(l)}(x))^2 dx. \quad (6.24)$$

El término h^{2l-1} fue introducido para eliminar factores dependientes de la h en las derivadas de los polinomios $p_{j+k}(x)$ obtenidos a partir de las medias en celdas.

Nuestro objetivo es utilizar (6.24) como el indicador de suavidad I^a , solamente sustituyendo los polinomios p_{j+k} obtenidos a partir de medias en celda por los obtenidos a partir de valores puntuales. También propondremos dos indicadores más, obtenidos discretizando (6.24) según la regla del punto medio:

$$(I^b)_k^j = \sum_{l=1}^r h^{2l} (p_{j+k}^{(l)}(x_{j-1/2}))^2, \quad (6.25)$$

o la regla del trapecio:

$$(I^c)_k^j = \sum_{l=1}^r h^{2l} ((p_{j+k}^{(l)}(x_{j-1}))^2 + (p_{j+k}^{(l)}(x_j))^2)/2. \quad (6.26)$$

Nota 6.2. *Obtendremos las propiedades deseadas de orden de aproximación óptimo utilizando un contexto común, en el que previamente trasladamos j a 0.*

Para $f \in \mathcal{C}^r(-d, d)$, con d proporcional a h , definimos la forma cuadrática semidefinida positiva:

$$I_h(f) = \sum_{l=1}^r h^{2l-1} \phi_h((f^{(l)})^2)$$

donde ϕ_h es un funcional lineal que verifica

$$\phi_h(f) \geq 0, \quad \text{si } f(x) \geq 0, \forall x \in [-h, 0], \quad (6.27)$$

y

$$\phi_h(x^m) = \alpha_m h^{m+1}, \quad \alpha_0 = 1. \quad (6.28)$$

Hay que notar que las hipótesis nos están dando que una serie de potencias de Taylor de f conduce a series de potencias de $\phi_h(f)$ en h :

$$\phi_h(f) = h \sum_{m=0}^{\infty} \frac{\alpha_m f^{(m)}(0)}{m!} h^m. \quad (6.29)$$

Los indicadores de suavidad anteriores pueden ser respectivamente obtenidos con los siguientes funcionales:

$$\phi_h(f) = \int_{-h}^0 f(x) dx, \quad \alpha_m = \frac{(-1)^m}{m+1} \quad (6.30)$$

$$\phi_h(f) = hf(-h/2), \quad \alpha_m = \frac{(-1)^m}{2^m} \quad (6.31)$$

$$\phi_h(f) = \frac{h}{2}(f(0) + f(-h)), \quad \alpha_0 = 1, \alpha_m = \frac{(-1)^m}{2}, m \geq 1 \quad (6.32)$$

Los resultados siguientes fijan el punto principal del diseño de pesos de orden óptimo, asegurando que los indicadores de suavidad obtenidos como $I_h(p)$, con p un polinomio interpolador de f , verifican las hipótesis de la Proposición 6.2 con $m = r \geq r - 1$.

Proposición 6.3. Si $f \in C^r(-d, d)$, $d > 0$, y denotamos $s = \max\{j/f^{(i)}(0) = 0, \forall i \leq j\}$ entonces:

1. $I_h(f) = K(f^{(s+1)}(0))^2 h^{2s+2} + o(h^{2s+2})$, $K \neq 0$.
2. Si p es un polinomio interpolador de f en los nodos $(-d, d)$, entonces $I_h(p) - I_h(f) = \mathcal{O}(h^{r+s+2})$.
3. Si p, q son polinomios interpoladores de f en los nodos $(-d, d)$, entonces

$$I_h(q) - I_h(p) = \mathcal{O}(h^{r+s+2}), \quad (6.33)$$

$$\frac{I_h(q) - I_h(p)}{I_h(p)} = \mathcal{O}(h^{r-s}). \quad (6.34)$$

Demostración. Para $l \leq s + 1$, tenemos las expansiones de Taylor de $f^{(l)}$:

$$f^{(l)}(x) = \frac{f^{(s+1)}(0)}{(s+1-l)!} x^{s+1-l} + o(x^{s+1-l}), \quad (6.35)$$

la cual, elevando al cuadrado, nos da:

$$(f^{(l)})^2(x) = \frac{(f^{(s+1)}(0))^2}{((s+1-l)!)^2} x^{2(s+1-l)} + o(x^{2(s+1-l)}).$$

Aplicando ϕ_h obtenemos

$$\phi_h((f^{(l)})^2) = (f^{(s+1)}(0))^2 \frac{\alpha_{2(s+1-l)}}{((s+1-l)!)^2} h^{2(s+1)-2l+1} + o(h^{2(s+1)-2l+1}),$$

lo que nos da:

$$\begin{aligned} I_h(f) &= \sum_{l=1}^r h^{2l-1} \phi_h((f^{(l)})^2) \\ &= (f^{(s+1)}(0))^2 \sum_{l=1}^{s+1} \left(\frac{\alpha_{2(s+1-l)}}{((s+1-l)!)^2} h^{2(s+1)} + o(h^{2(s+1)}) \right) + \sum_{l=s+2}^r h^{2l} \mathcal{O}(1) \\ &= K (f^{(s+1)}(0))^2 h^{2s+2} + o(h^{2s+2}), \end{aligned}$$

donde $K = \sum_{l=0}^s \frac{\alpha_{2l}}{(2l)!^2} > 0$, por lo que (6.27) y (6.28) implica:

$$\alpha_{2l} \geq \frac{\min_{x \in [-h,0]} x^{2l}}{h^{2l+1}} \geq 0, \alpha_0 = 1.$$

De (6.27) y (6.28) deducimos

$$h \min_{x \in [-h,0]} f(x) \leq \phi_h(f) \leq h \max_{x \in [-h,0]} f(x), \quad (6.36)$$

y de esto:

$$|\phi_h(f)| \leq \|f\| h, \quad \|f\| = \max_{x \in [-h,0]} |f(x)|,$$

de donde

$$|I_h(p) - I_h(f)| \leq \sum_{l=1}^r h^{2l} \|(p^{(l)})^2 - (f^{(l)})^2\| \quad (6.37)$$

Ahora, sea p un polinomio interpolador en nodos de $(-d, d)$. Si tenemos en cuenta que $p^{(l)} - f^{(l)} = \mathcal{O}(h^{r+1-l})$ y que (6.35) no da $f^{(l)} = \mathcal{O}(h^{s+1-l})$, $l \leq s+1$, entonces deducimos:

$$\begin{aligned} (p^{(l)})^2 - (f^{(l)})^2 &= (p^{(l)} - f^{(l)})^2 + 2(p^{(l)} - f^{(l)})f^{(l)} \\ &= \mathcal{O}(h^{r+1-l})^2 + \mathcal{O}(h^{r+1-l})\mathcal{O}(h^{s+1-l}) = \mathcal{O}(h^{r+s+2-2l}), \quad \forall l \leq s+1. \end{aligned}$$

Si sustituimos esto en (6.37):

$$|I_h(p) - I_h(f)| \leq \sum_{l=1}^{s+1} h^{2l} \mathcal{O}(h^{r+s+2-2l}) + \sum_{l=s+2}^r h^{2l} \mathcal{O}(1) = \mathcal{O}(h^{r+s+2}).$$

Ahora, para obtener (6.33) y (6.34) utilizamos los resultados previos:

$$\begin{aligned} I_h(q) - I_h(p) &= I_h(q) - I_h(f) + I_h(f) - I_h(p) = \mathcal{O}(h^{r+s+2}) \\ \frac{I_h(q) - I_h(p)}{I_h(p)} &= \frac{I_h(q) - I_h(p)}{I_h(f) - I_h(f) + I_h(p)} \\ &= \frac{\mathcal{O}(h^{r+s+2})}{Ah^{2s+2} + \mathcal{O}(h^{r+s+2})} \\ &= \frac{\mathcal{O}(h^{r-s})}{A + \mathcal{O}(h^{r-s})} = \mathcal{O}(h^{r-s}), \end{aligned}$$

donde $A = K(f^{(s+1)}(0))^2 \neq 0$.

□

Los corolarios siguientes resumen los resultados más importantes obtenidos con el interpolante WENO:

Corolario 6.1. Sea $f \in \mathcal{C}^{2r}([x_{j-r}, x_{j+r-1}])$ y $s = \max\{m/f^{(i)}(x_j) = 0, \forall i \leq m\}$. Si definimos

$$\mathcal{I}(x_{j-\frac{1}{2}}; f) = \sum_{k=0}^{r-1} \omega_k p_{j+k}^r(x_{j-\frac{1}{2}}). \quad (6.38)$$

donde

$$\omega_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i}, \quad k = 0, \dots, r-1 \text{ y } \alpha_k = \frac{C_k^r}{(h^2 + I_k^{a,b,c})^t}, \quad (6.39)$$

entonces

$$f(x_{j-\frac{1}{2}}) - \mathcal{I}(x_{j-\frac{1}{2}}; f) = \mathcal{O}(h^{2r}), \quad (6.40)$$

Demostración. El resultado se sigue de (6.16), ya que (6.33) nos permite tomar $r + s \geq r - 1$ en (6.15). \square

El resultado siguiente indica que la interpolación WENO es al menos tan exacta como la interpolación ENO en discontinuidades de salto si al menos hay un *stencil* en el cual la función es suave.

Corolario 6.2. Sea f tal que $f \in \mathcal{C}^{r+1}((-d, d) \setminus \{0\})$, $d > 0$ y $f(0) = f(0^-) \neq f(0^+)$ y elegimos t en (6.14) tal que $2t \geq r + 1$. Entonces

$$f(-h/2) - \mathcal{I}(-h/2; f) = \mathcal{O}(h^{r+1}). \quad (6.41)$$

Demostración. Sea p el interpolador de f en los nodos $\{x_{k-r}, \dots, x_k\}$. Por la Proposición 6.2 y la hipótesis de suavidad, tenemos $I_0(f) = \mathcal{O}(h^2)$.

Ahora, consideremos $k > 0$. El primer término en la definición del indicador de suavidad $I_k^a(f)$ (la prueba para los indicadores $I^{b,c}$ es similar) en el *stencil* $\{x_{k-r}, \dots, x_k\}$, es

$$h \int_{-h}^0 (p'(x))^2 dx.$$

Con el cambio de variables $x = hy$, esto puede ser escrito como

$$\int_{-1}^0 (q'_h(y))^2 dy,$$

donde q_h es el polinomio de interpolación de $f_h(y) = f(hy)$ en el stencil $\{k-r, \dots, k\}$. La expresión para este polinomio es:

$$q_h(y) = \sum_{i=0}^r f_h[k-r, \dots, k-r+i] q_i(y), \quad q_i(y) = (y-(k-r)) \dots (y-(k-r+i)),$$

donde $f_h[\dots]$ denota las diferencias no divididas de f_h . La derivada de q_h viene dada por

$$q'_h(y) = \sum_{i=1}^r f_h[k-r, \dots, k-r+i] q'_i(y).$$

Entonces

$$\begin{aligned} \lim_{h \rightarrow 0} h \int_{-h}^0 (p'(x))^2 dx &= \int_{-1}^0 (\lim_{h \rightarrow 0} q'_h(y))^2 dy \\ &= \int_{-1}^0 \left(\sum_{i=1}^r \lim_{h \rightarrow 0} f_h[k-r, \dots, k-r+i] q'_i(y) \right)^2 dy. \end{aligned} \tag{6.42}$$

Pero, para $k-r+i > 0$, esto es, para $i > r-k$, tenemos:

$$\begin{aligned} \lim_{h \rightarrow 0} f_h[k-r, \dots, k-r+i] &= \lim_{h \rightarrow 0} \sum_{j=0}^i (-1)^j \binom{i}{j} f_h(k-r+j) \\ &= \sum_{j=0}^i (-1)^j \binom{i}{j} \lim_{h \rightarrow 0} f((k-r+j)h) \\ &= \sum_{j=0}^{r-k} (-1)^j \binom{i}{j} f(0^-) + \sum_{j=r-k+1}^i (-1)^j \binom{i}{j} f(0^+) \\ &= (f(0^+) - f(0^-)) \sum_{j=r-k+1}^i (-1)^j \binom{i}{j} \neq 0, \end{aligned}$$

y por tanto

$$\sum_{i=1}^r \lim_{h \rightarrow 0} f_h[k-r, \dots, k-r+i] q'_i(y) = \sum_{i=m-k+r+1}^r \lim_{h \rightarrow 0} f_h[k-r, \dots, k-r+i] q'_i(y) \neq 0,$$

lo cual, junto con (6.42), implica que $I_k(f) \not\rightarrow 0$ cuando $h \rightarrow 0$. Por la Proposición 6.2, obtenemos (6.41). \square

Nota 6.3. Si f es continua no podemos garantizar una aproximación de orden $r+1$ en los puntos donde f' tiene una discontinuidad de salto, como veremos en el ejemplo 6.2.

6.5

Evaluación del indicador de suavidad

Sea p un polinomio de grado r , dado por:

$$p = \sum_{k=0}^r \delta_k (x/h)^k.$$

La derivada l -ésima de p viene dada por:

$$p^{(l)} = l! \sum_{k=0}^{r-l} \frac{\delta_{k+l}}{h^{k+l}} \binom{k+l}{l} x^k, \quad l \leq r.$$

Entonces deducimos:

$$(p^{(l)})^2 = (l!)^2 \sum_{k=0}^{r-l} \sum_{i=0}^{r-l} \delta_{k+l} \delta_{i+l} \binom{k+l}{l} \binom{i+l}{l} x^{i+k} \frac{1}{h^{i+k+2l}}$$

Si aplicamos ϕ_h a ambos lados de dicha ecuación, teniendo en cuenta (6.28) y utilizando la igualdad de la definición de $I_h(p)$, ob-

tenemos:

$$\begin{aligned}
I_h(p) &= \sum_{l=1}^r h^{2l-1} \phi_h((p^{(l)})^2) \\
&= \sum_{l=1}^r h^{2l-1} (l!)^2 \sum_{k=0}^{r-l} \sum_{i=0}^{r-l} \delta_{k+l} \delta_{i+l} \binom{k+l}{l} \binom{i+l}{l} \phi(x^{i+k}) \frac{1}{h^{i+k+2l}} \\
&= \sum_{l=1}^r h^{2l-1} (l!)^2 \sum_{k=0}^{r-l} \sum_{i=0}^{r-l} \delta_{k+l} \delta_{i+l} \binom{k+l}{l} \binom{i+l}{l} \alpha_{i+k} h^{i+k+1} \frac{1}{h^{i+k+2l}} \\
&= \sum_{l=1}^r (l!)^2 \sum_{k=0}^{r-l} \sum_{i=0}^{r-l} \delta_{k+l} \delta_{i+l} \binom{k+l}{l} \binom{i+l}{l} \alpha_{i+k} \\
I_h(p) &= \sum_{i=1}^r \delta_i^2 \sum_{l=1}^i (l!)^2 \binom{i}{l}^2 \alpha_{2i-2l} + 2 \sum_{i=1}^r \sum_{j=i+1}^r \delta_i \delta_j \sum_{l=1}^i (l!)^2 \binom{i}{l} \binom{j}{l} \alpha_{i+j-2l}.
\end{aligned} \tag{6.43}$$

De (6.30)–(6.32) calculamos los coeficientes de los monomios cuadráticos que aparecen en (6.43) para los indicadores $I^{a,b,c}$ y $r = 2, 3$ y obtenemos expresiones que son más eficientes computacionalmente tomando los cuadrados de cada expresión.

Para $r = 2$, el indicador de suavidad es:

$$I^a = \delta_1^2 - 2\delta_1\delta_2 + \frac{16}{3}\delta_2^2 = (\delta_1 - \delta_2)^2 + \frac{13}{3}\delta_2^2 \tag{6.44}$$

$$I^b = \delta_1^2 - 2\delta_1\delta_2 + 5\delta_2^2 = (\delta_1 - \delta_2)^2 + 4\delta_2^2 \tag{6.45}$$

$$I^c = \delta_1^2 - 2\delta_1\delta_2 + 6\delta_2^2 = (\delta_1 - \delta_2)^2 + 5\delta_2^2 \tag{6.46}$$

Los coeficientes δ_1, δ_2 de p_0^2 vienen dados por:

$$\delta_1 = \frac{3}{2}f_0 - 2f_{-1} + \frac{1}{2}f_{-2}$$

$$\delta_2 = \frac{1}{2}f_0 - f_{-1} + \frac{1}{2}f_{-2}$$

y para p_1^2 por:

$$\delta_1 = -\frac{1}{2}f_{-1} + \frac{1}{2}f_1$$

$$\delta_2 = \frac{1}{2}f_{-1} - f_0 + \frac{1}{2}f_1$$

Ahora, para $r = 3$, el indicador de suavidad es:

$$\begin{aligned} I^a &= \delta_1^2 - 2\delta_1\delta_2 + 2\delta_1\delta_3 + \frac{16}{3}\delta_2^2 - 15\delta_2\delta_3 + \frac{249}{5}\delta_3^2 \\ &= (\delta_1 - \delta_2 + \delta_3)^2 + \frac{13}{3}(\delta_2 - \frac{3}{2}\delta_3)^2 + \frac{781}{20}\delta_3^2. \end{aligned} \quad (6.47)$$

$$\begin{aligned} I^b &= \delta_1^2 - 2\delta_1\delta_2 + \frac{3}{2}\delta_1\delta_3 + 5\delta_2^2 - \frac{27}{2}\delta_2\delta_3 + \frac{729}{16}\delta_3^2 \\ &= (\delta_1 - \delta_2 + \frac{3}{4}\delta_3)^2 + 4(\delta_2 - \frac{3}{2}\delta_3)^2 + 36\delta_3^2. \end{aligned} \quad (6.48)$$

$$\begin{aligned} I^c &= \delta_1^2 - 2\delta_1\delta_2 + 3\delta_1\delta_3 + 6\delta_2^2 - 18\delta_2\delta_3 + \frac{117}{2}\delta_3^2 \\ &= (\delta_1 - \delta_2 + \frac{3}{2}\delta_3)^2 + 5(\delta_2 - \frac{3}{2}\delta_3)^2 + 45\delta_3^2. \end{aligned} \quad (6.49)$$

donde los coeficientes vienen dados por:

$$\begin{aligned} \delta_1 &= -\frac{1}{3}f_{-3} + \frac{3}{2}f_{-2} - 3f_{-1} + \frac{11}{6}f_0 \\ \delta_2 &= -\frac{1}{2}f_{-3} + 2f_{-2} - \frac{5}{2}f_{-1} + f_0 \\ \delta_3 &= -\frac{1}{6}f_{-3} + \frac{1}{2}f_{-2} - \frac{1}{2}f_{-1} + \frac{1}{6}f_0 \end{aligned}$$

para p_0^3 ,

$$\begin{aligned} \delta_1 &= \frac{1}{6}f_{-2} - f_{-1} + \frac{1}{2}f_0 + \frac{1}{3}f_1 \\ \delta_2 &= \frac{1}{2}f_{-1} - f_0 + \frac{1}{2}f_1 \\ \delta_3 &= -\frac{1}{6}f_{-2} + \frac{1}{2}f_{-1} - \frac{1}{2}f_0 + \frac{1}{6}f_1 \end{aligned}$$

para p_1^3 , y

$$\begin{aligned} \delta_1 &= -\frac{1}{3}f_{-1} - \frac{1}{2}f_0 + f_1 - \frac{1}{6}f_2 \\ \delta_2 &= \frac{1}{2}f_{-1} - f_0 + \frac{1}{2}f_1 \\ \delta_3 &= -\frac{1}{6}f_{-1} + \frac{1}{2}f_0 - \frac{1}{2}f_1 + \frac{1}{6}f_2 \end{aligned}$$

para p_2^3 .

Nota 6.4. A pesar de que en principio $\varepsilon_h > 0$ es introducido para evitar denominadores nulos en (6.14), resulta que su presencia en el cálculo de α_k es crucial para la estimación de $|I_l - I_k|/(\varepsilon_h + I_k) = \mathcal{O}(h^m)$ que aparece en la prueba de la Proposición 6.2. Por ejemplo, si $\varepsilon_h = 0$, una estimación para esto puede ser obtenida de $|I_l - I_k|/I_k$, en vez de $|I_l - I_k|$ como en el último resultado. Esto puede producir una bajada del orden $2r$ en la interpolación de los puntos donde $s > 1$ de la Proposición 6.3 ($f' = f'' = 0$, ver (6.34)), como ocurre para $f(x) = x^3$ y $r = 2$, por ejemplo. En este caso, el error $q(-h/2) - (-h/2)^3$ para el interpolador WENO basado en el indicador de suavidad I^a y $\varepsilon_h = h^2$ viene dado por:

$$\frac{(-117)(41h^4 + 2)}{8(1601h^8 + 82h^4 + 2)}h^7 = \mathcal{O}(h^7),$$

mientras que la misma técnica, con $\varepsilon_h = 0$, nos da un orden menor que $2r = 4$:

$$-4797/12808 h^3 = \mathcal{O}(h^3).$$

Ejemplo 6.1. Consideremos la función discontinua f dada por $\exp(x)$ para $x \leq 0$ y $1 + \exp(x)$ para $x > 0$. Tomamos $r = 2$ y calculamos la interpolación WENO con los pesos calculados con los indicadores de suavidad I^a . Concretamente, calculamos los errores de interpolación $e_i = q(-h/2) - f(-h/2)$ con una precisión de 32 dígitos, con el interpolador WENO q calculado con $\varepsilon_h = 0, h^2, 10^{-6}$ y $h = 0, 1/2^i$, $i = 1, \dots, 20$. En la Tabla 6.2 mostramos e_i , $i = 1, \dots, 20$ y los cocientes e_i/e_{i+1} para $i = 1, \dots, 19$ para las distintas alternativas de ε_h . Los cocientes correspondientes a $\varepsilon_h = 0$ y h^2 muestran un convergencia hacia $8 = 2^{r+1}$. Por otra parte, la columna de e_i para $\varepsilon_h = 10^{-6}$ se estabiliza en 10^{-13} , lo cual es, en cualquier caso, bastante aproximado en doble precisión.

Ejemplo 6.2. Consideremos

$$f(x) = \begin{cases} 0 & \text{si } x \leq x_j = a \\ x - a & \text{si } x \geq x_j. \end{cases}$$

$\varepsilon_h = 0$		$\varepsilon_h = h^2$		$\varepsilon_h = 10^{-6}$	
e_i	e_i/e_{i+1}	e_i	e_i/e_{i+1}	e_i	e_i/e_{i+1}
6.8820e-06	7.5133e+00	4.9852e-06	6.2802e+00	6.8815e-06	7.5138e+00
9.1598e-07	7.7488e+00	7.9379e-07	7.1848e+00	9.1585e-07	7.7499e+00
1.1821e-07	7.8726e+00	1.1048e-07	7.6038e+00	1.1818e-07	7.8748e+00
1.5015e-08	7.9359e+00	1.4530e-08	7.8048e+00	1.5007e-08	7.9406e+00
1.8920e-09	7.9679e+00	1.8616e-09	7.9031e+00	1.8899e-09	7.9797e+00
2.3746e-10	7.9839e+00	2.3556e-10	7.9517e+00	2.3683e-10	8.0267e+00
2.9742e-11	7.9919e+00	2.9623e-11	7.9759e+00	2.9506e-11	8.2361e+00
3.7215e-12	7.9960e+00	3.7141e-12	7.9880e+00	3.5825e-12	1.0212e+01
4.6543e-13	7.9980e+00	4.6496e-13	7.9940e+00	3.5081e-13	6.9680e+00
5.8193e-14	7.9990e+00	5.8164e-14	7.9970e+00	5.0346e-14	5.1095e-01
7.2750e-15	7.9995e+00	7.2732e-15	7.9985e+00	9.8534e-14	9.2934e-01
9.0944e-16	7.9997e+00	9.0932e-16	7.9992e+00	1.0603e-13	9.9343e-01
1.1368e-16	7.9999e+00	1.1368e-16	7.9996e+00	1.0673e-13	1.0024e+00
1.4211e-17	7.9999e+00	1.4210e-17	7.9998e+00	1.0647e-13	9.9951e-01
1.7763e-18	8.0000e+00	1.7763e-18	7.9999e+00	1.0652e-13	1.0003e+00
2.2204e-19	8.0000e+00	2.2204e-19	8.0000e+00	1.0650e-13	9.9948e-01
2.7756e-20	8.0000e+00	2.7755e-20	8.0000e+00	1.0655e-13	1.0012e+00
3.4694e-21	8.0000e+00	3.4694e-21	8.0000e+00	1.0643e-13	9.9857e-01
4.3368e-22	8.0000e+00	4.3368e-22	8.0000e+00	1.0658e-13	1.0005e+00
5.4210e-23		5.4210e-23		1.0652e-13	

Tabla 6.2: Errores de interpolación WENO para $r = 2$, $h = 0,1/2^i$, $i = 1, \dots, 20$ y $\varepsilon_h = 0, h^2, 10^{-6}$.

Queremos evaluar (6.38), para $r = 2$. Si tomamos el indicador de suavidad I^b (obtenemos los mismos resultados con los otros) obtenemos, (6.45), $I_1^b = (f_j - f_{j-1})^2 + 4(\frac{1}{2}f_{j+1} - f_j + \frac{1}{2}f_{j-1})^2 = h^2$ y $I_1^b = (f_{j-1} - f_{j-2})^2 + 4(\frac{1}{2}f_j - f_{j-1} + \frac{1}{2}f_{j-2})^2 = 0$.

Como $\alpha_1^j = \frac{1/2}{(h^2+I_1^b)^2} = \frac{1}{8h^4}$ y $\alpha_0^j = \frac{1/2}{(h^2+I_0^b)^2} = \frac{1}{h^4}$ tenemos $\omega_1^j = \frac{\alpha_1^j}{\alpha_0^j + \alpha_1^j} = \frac{1}{5}$ y $\omega_0^j = \frac{\alpha_0^j}{\alpha_0^j + \alpha_1^j} = \frac{4}{5}$.

Ahora, de $p_{j+1}^2(x_{j-\frac{1}{2}}) = \frac{3}{8}f_{j-1} + \frac{3}{4}f_j - \frac{1}{8}f_{j+1} = -\frac{1}{8}h$ y $p_j^2(x_{j-\frac{1}{2}}) = -\frac{1}{8}f_{j-2} + \frac{3}{4}f_{j-1} + \frac{3}{8}f_j = 0$ obtenemos

$$\mathcal{I}(x_{j-\frac{1}{2}}; f) = \sum_{k=0}^1 \omega_k^j p_{j+k}^2(x_{j-\frac{1}{2}}) = \frac{1}{5}0 + \frac{4}{5}(-\frac{1}{8}h) = -\frac{1}{40}h,$$

lo cual no es una aproximación $\mathcal{O}(h^2)$ a $f(x_{j-\frac{1}{2}}) = 0$. Notemos que dicha aproximación, $-\frac{1}{40}h$, está más cerca del valor exacto que $p_j^3(x_{j-\frac{1}{2}}) = -\frac{1}{16}f_{j-2} + \frac{9}{16}f_{j-1} + \frac{9}{16}f_j - \frac{1}{16}f_{j+1} = -\frac{1}{16}h$.

Ejemplo 6.3. Consideramos ahora la siguiente función (Figura 6.1(a)):

$$f(x) = \begin{cases} 5(x - 0,25)^3 e^{x^2}, & 0 \leq x < 2/3, \\ 1,5 - (x - 0,25)^3 e^{x^2} & 2/3 \leq x \leq 0. \end{cases} \quad (6.50)$$

que tiene dos singularidades:

1. En $x = 0,25$ tenemos $f(0,25) = f'(0,25) = f''(0,25) = 0$, es decir, $s = 2$, con s definida en la Proposición 6.3.
2. En $x = 2/3$ tenemos una discontinuidad.

Partiendo de una discretización de f utilizando 33 nodos, es decir de $\{f_i\}_{i=0}^{32}$ siendo $f_i = f(ih)$ y $h = 1/32$, intentamos obtener una aproximación a $\{f_i^0\}_{i=0}^{64}$ con $f_i^0 = f(ih/2)$ utilizando la interpolación WENO- I^a con $r = 3$. En la Figura 6.1 están los distintos pesos que hemos obtenido para distintos coeficientes ε_h y en la Figura 6.2 podemos ver distintas aproximaciones a $\{f_i^0\}_{i=0}^{64}$ utilizando la reconstrucción WENO- I^a con distintos coeficientes ε_h .

En primer lugar analizamos la singularidad localizada en $x = 0,25$. En la Figura 6.1 dibujamos los distintos pesos obtenidos al variar ε_h . Comparamos los pesos obtenidos con los pesos ideales

$C_0^3 = C_2^3 = 3/16$ y $C_1^3 = 10/16$. Observamos que obtenemos mejores aproximaciones cuando $\varepsilon_h = 1$ y cuando es h^2 . Para $\varepsilon_h = 10^{-6}$ tenemos una mala aproximación a los pesos ideales. El peor resultado lo obtenemos cuando tomamos $\varepsilon_h = 0$ y $\varepsilon_h = h^6$.

Si observamos las reconstrucciones obtenidas con WENO y tomando distintos valores ε_h (Figura 6.2) no vemos diferencias alrededor de $x = 0,25$.

Para analizar los resultados obtenidos, recordemos que en la demostración de la Proposición 6.2 tenemos que de $\frac{I_l - I_k}{\varepsilon_h + I_k} = \mathcal{O}(h^m)$ deducíamos $\omega_l = C_l^r + \mathcal{O}(h^m)$ y de aquí $f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) = \mathcal{O}(h^{\min(2r, m+r+1)})$.

Por otra parte, por la Proposición 6.3, tenemos $I_h(p) = \mathcal{O}(h^{2s+2})$ y $I_h(q) - I_h(p) = \mathcal{O}(h^{r+s+2})$.

	$\varepsilon_h = 1$	$\varepsilon_h = 0$	$\varepsilon_h = h^2$	$\varepsilon_h = 10^{-6} \approx h^4$	$\varepsilon_h = h^6$
$\frac{I_l - I_k}{\varepsilon_h + I_k}$	$\mathcal{O}(h^7)$	$\mathcal{O}(h)$	$\mathcal{O}(h^5)$	$\mathcal{O}(h^3)$	$\mathcal{O}(h)$
$\omega_l - C_l^r$	$\mathcal{O}(h^7)$	$\mathcal{O}(h)$	$\mathcal{O}(h^5)$	$\mathcal{O}(h^3)$	$\mathcal{O}(h)$
$f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}})$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^5)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^5)$

Tabla 6.3: Cómo afecta a los distintos órdenes de aproximación cerca de $x = 0,25$ el tomar distintos ε_h

Como en $x = 0,25$ $s = 2$ y hemos tomado $r = 3$ tendremos $I_h(p) = \mathcal{O}(h^6)$ y $I_h(q) - I_h(p) = \mathcal{O}(h^7)$. En la Tabla 6.3 reflejamos las distintas aproximaciones teóricas que obtenemos al variar ε_h . Hay que señalar que el coeficiente que ha sido utilizado habitualmente en la literatura es $\varepsilon_h = 10^{-6}$. En este ejemplo, como $h = 1/32$, ε_h vale aproximadamente h^4 . Vemos que los resultados teóricos y los numéricos concuerdan.

Pasemos a analizar la discontinuidad en $x = 2/3$. En la Proposición 6.2 vemos que si tenemos $I_k = \mathcal{O}(h^2)$ cuando f es suave en S_{j+k} , que si $I_k \not\rightarrow 0$, $h \rightarrow 0$ cuando f no es suave en S_{j+k} y si f es suave en al menos uno de los stencils S_{j+k} , entonces $f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) = \mathcal{O}(h^{\min(2t, r+1)}) = \mathcal{O}(h^4)$ si $t = 2$ y $r = 3$ (el mismo orden que con la aproximación ENO).

Si miramos detenidamente la demostración vemos que esto se

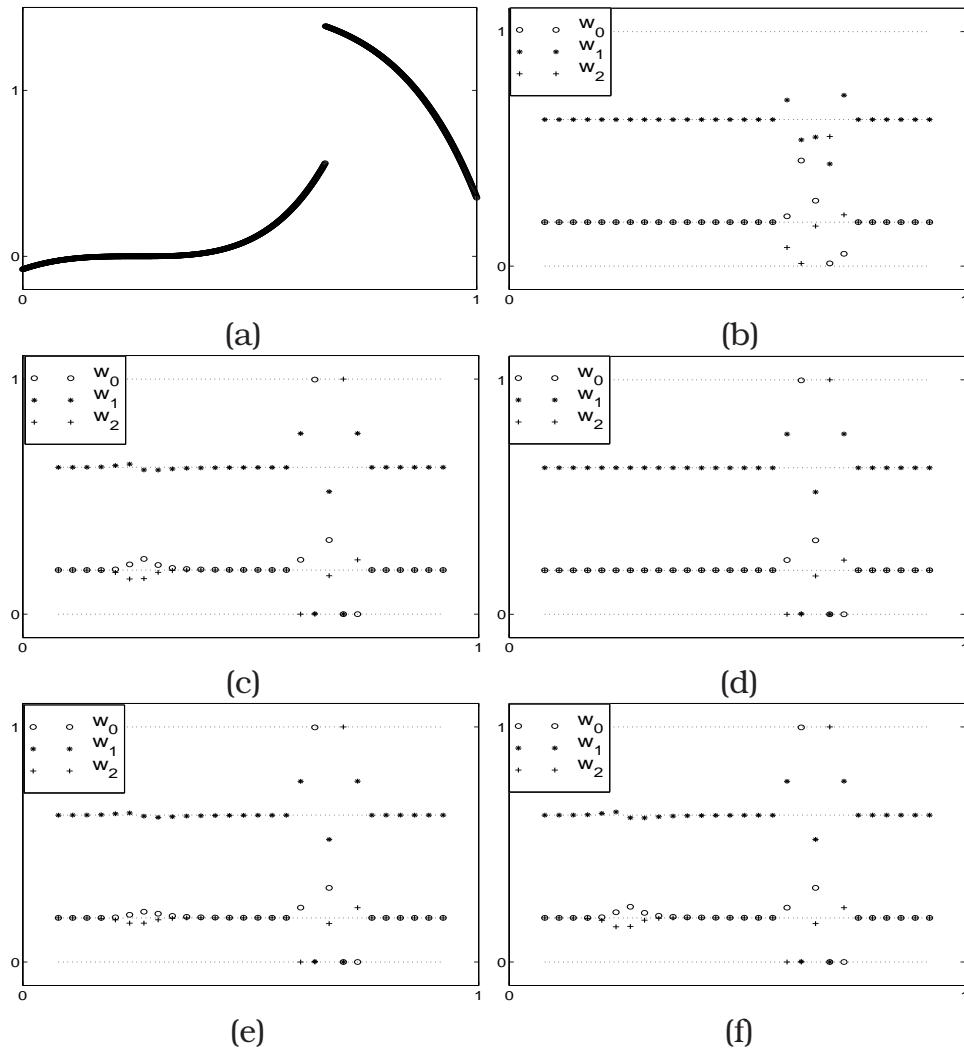


Figura 6.1: (a) Función original (6.50). Pesos w_0 , w_1 y w_2 obtenidos con distintos ϵ_h (b) 1 (c) 0 (d) h^2 (e) 10^{-6} (f) h^6

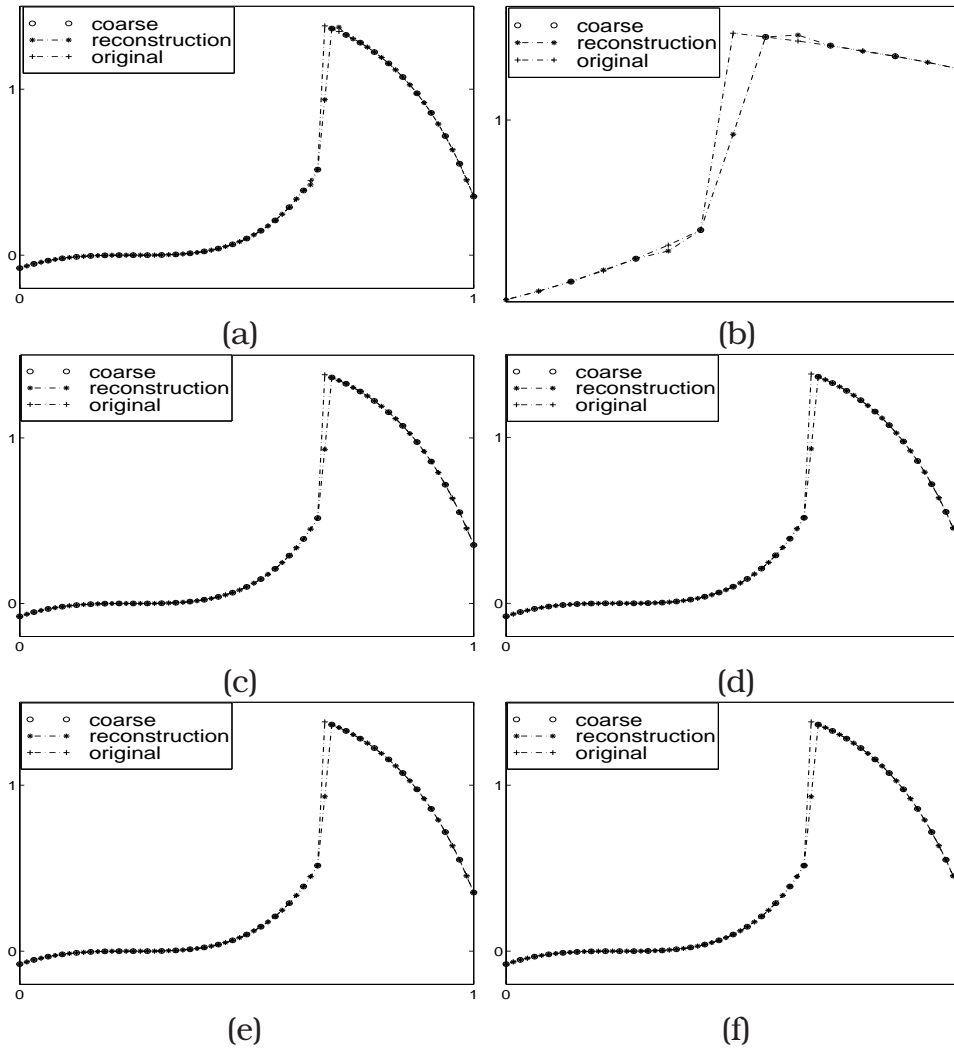


Figura 6.2: A partir de $\{f_i\}_{i=0}^{32}$ (coarse) se intenta recuperar $\{f_i\}_{i=0}^{64}$ (original) utilizando reconstrucción WENO- I^a utilizando unos pesos (6.14) calculados utilizando los valores ε_h : (b) 1 (c) 0 (d) h^2 (e) 10^{-6} (f) h^6 . (b) es un zoom de (a) alrededor de $x = 2/3$.

puede probar porque si tomamos $\varepsilon_h = h^2$ y como $I_i^t = \mathcal{O}(h^2)$ tenemos

$$\alpha_k = \frac{C_k^r}{(\varepsilon_h + I_k)^t} = \frac{C_k^r(1 + \mathcal{O}(h^m))}{(\varepsilon_h + I_l)^t} = \mathcal{O}(h^{-2t}).$$

Esto también sería cierto cuando $\varepsilon_h = 0, h^2, h^4, h^6$. Sin embargo, si $\varepsilon_h = 1$ tenemos $\alpha_k = \mathcal{O}(1)$. En este caso tendremos siempre $\alpha_k = \mathcal{O}(1)$, en lugar de (6.22) y de aquí se deducirá que $f(x_{j-\frac{1}{2}}) - q(x_{j-\frac{1}{2}}) = \mathcal{O}(1)$ en lugar de (6.23). En la Figura 6.1 vemos que alrededor de $x = 2/3$ todos los pesos han sido calculados correctamente con los distintos ε_h exceptuando $\varepsilon_h = 1$. En la Figura 6.2 también observamos que sólo cuando $\varepsilon_h = 1$ no tenemos una buena aproximación (Figura 6.2 (a), (b)).

7

Experimentos numéricos 2D para valores puntuales

En este capítulo vamos a comparar el comportamiento de la multirresolución utilizando esquemas interpolatorios lineales y esquemas WENO. Consideraremos dos técnicas interpolatorias del mismo grado ($r = 3$):

- Una técnica interpolatoria centrada que utiliza cuatro puntos (LINEAL).
- La técnica interpolatoria WENO que utiliza 3 stencils de cuatro puntos cada uno (en total 6 puntos) tal y como se presenta en el Capítulo anterior.

Aplicaremos la transformada directa modificada para valores puntuales descrita en el Algoritmo A.1, que garantiza a priori una

cierta calidad de la imagen reconstruida (Proposición 1.3 y Corolario 1.1). Hemos procesado las imágenes del *algoritmo de control del error 2D* cuantizando con $\varepsilon_k = \varepsilon$.

En todos nuestros tests tomamos $L = 4$, $N_L = 512 \times 512$. Mediremos la calidad evaluando la raíz de la media error *Root Mean Square Error (RMSE)* y el error en la norma máximo *Peak Absolute Error (PAE)* donde

$$PAE = \|\bar{f}^k\|_\infty = \sup_{i,j} |\bar{f}_{i,j}^k| \quad \text{y} \quad RMSE = \|\bar{f}^k\|_2 = \frac{1}{N_k} \sqrt{\sum_{i,j} |\bar{f}_{i,j}^k|^2}$$

7.1

Esquemas adaptativos para PDE

Nuestros primeros experimentos se pueden ver en la Figura 7.1, están basados en diferentes etapas de una simulación obtenida con un esquema conservativo para la evolución de un problema de dinámica de gases en 2D gobernada por las ecuaciones de Euler (ver [40, 28, 12] y las referencias que hay allí).

La motivación de utilizar estos ejemplos se basa en la posibilidad de utilizar técnicas multiescala en algoritmos de refinamiento adaptativos *adaptive mesh refinement codes (AMR)* para decidir en qué partes del dominio computacional se necesita alta resolución (ver [12]).

En las Tablas 7.1, 7.2 y 7.3 mostramos los valores distintos de cero de la transformada cuantizada así como los errores en las diferentes normas para cada uno de los esquemas de compresión que consideramos.

En la Figura 7.2 podemos ver los valores distintos de cero cuando cuantizamos con $\varepsilon = 8$ para las diferentes imágenes y cuando utilizamos las interpolaciones LINEAL y WENO- I^{3b} .

En todos los casos el número de valores distintos de cero conseguidos con el algoritmo WENO es significativamente menor que el de los conseguidos con el esquema LINEAL correspondiente. Podemos ver también que con el esquema WENO las singularidades están mejor localizadas.

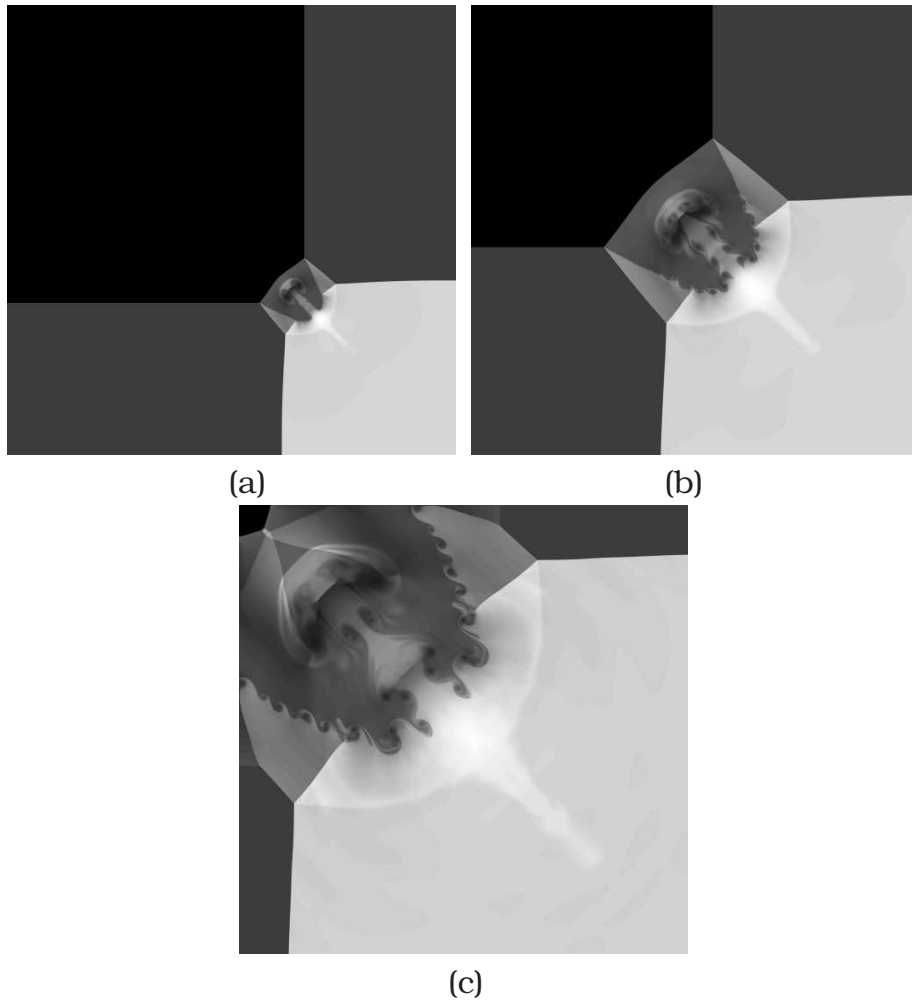


Figura 7.1: Simulación de la evolución de un problema de dinámica de gases para distintas etapas: (a) $t=20$, (b) $t=50$ y (c) $t=102$

ε	step	PAE	RMSE	Nnz	ε	step	PAE	RMSE	Nnz
2	LINEAL	2	0.480	6929	8	LINEAL	8	1.775	4203
	WENO- I^{3a}	2	0.330	5129		WENO- I^{3a}	8	1.085	3791
	WENO- I^{3b}	2	0.330	5119		WENO- I^{3b}	8	1.088	3788
	WENO- I^{3c}	2	0.339	5161		WENO- I^{3c}	8	1.083	3793
4	LINEAL	4	0.889	5419	10	LINEAL	10	2.140	4180
	WENO- I^{3a}	4	0.595	4596		WENO- I^{3a}	10	1.388	3920
	WENO- I^{3b}	4	0.598	4602		WENO- I^{3b}	10	1.403	3906
	WENO- I^{3c}	4	0.600	4611		WENO- I^{3c}	10	1.399	3917
6	LINEAL	6	1.309	4765	12	LINEAL	12	2.105	3875
	WENO- I^{3a}	6	0.799	4268		WENO- I^{3a}	12	1.357	3472
	WENO- I^{3b}	6	0.871	4532		WENO- I^{3b}	12	1.340	3475
	WENO- I^{3c}	6	0.868	4557		WENO- I^{3c}	12	1.346	3496

Tabla 7.1: Valores distintos de cero de la transformada cuantizada así como los errores en las diferentes normas para cada uno de los esquemas de compresión para la imagen t_{20}

ε	step	PAE	RMSE	Nnz	ε	step	PAE	RMSE	Nnz
2	LINEAL	2	0.594	11189	8	LINEAL	8	2.077	6140
	WENO- I^{3a}	2	0.518	8251		WENO- I^{3a}	8	1.489	5334
	WENO- I^{3b}	2	0.517	8268		WENO- I^{3b}	8	1.481	5345
	WENO- I^{3c}	2	0.516	8308		WENO- I^{3c}	8	1.479	5341
4	LINEAL	4	1.068	8659	10	LINEAL	10	2.474	5374
	WENO- I^{3a}	4	0.859	6822		WENO- I^{3a}	10	1.842	4593
	WENO- I^{3b}	4	0.857	6820		WENO- I^{3b}	10	1.850	4565
	WENO- I^{3c}	4	0.855	6821		WENO- I^{3c}	10	1.857	4575
6	LINEAL	6	1.558	7195	12	LINEAL	12	3.683	0.118
	WENO- I^{3a}	6	1.209	5948		WENO- I^{3a}	12	2.738	4419
	WENO- I^{3b}	6	1.196	6053		WENO- I^{3b}	12	2.017	4295
	WENO- I^{3c}	6	1.196	6044		WENO- I^{3c}	12	2.011	3856

Tabla 7.2: Valores distintos de cero de la transformada cuantizada así como los errores en las diferentes normas para cada uno de los esquemas de compresión para la imagen t_{50}

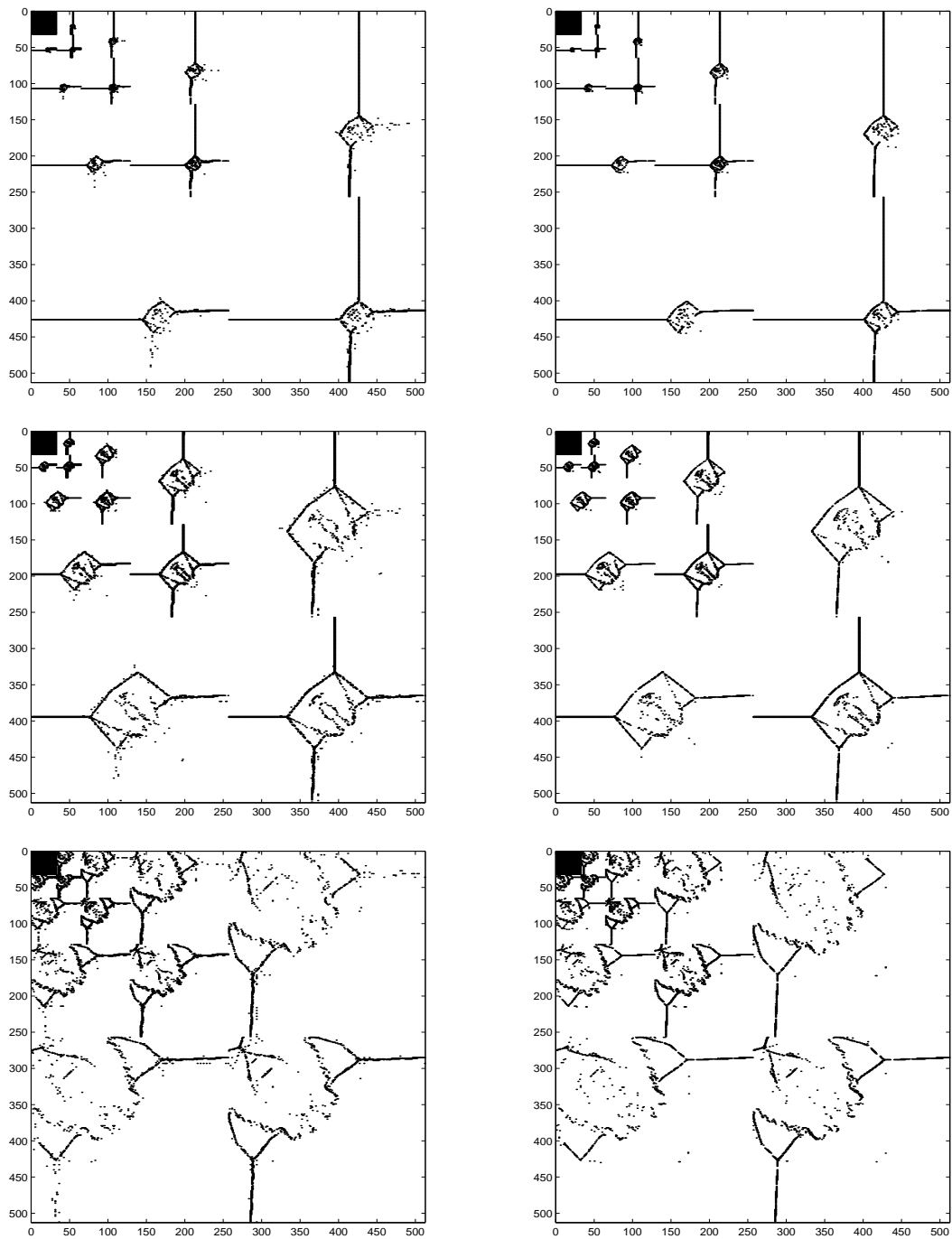


Figura 7.2: Valores distintos de cero obtenidos por la transformada después de cuantizar los detalles con $\varepsilon = 8$. De arriba abajo: imagen $t=20$, imagen $t=50$ e imagen $t=102$. Izquierda LINEAL, Derecha WENO-I^{3b}

ε	step	PAE	RMSE	Nnz	ε	step	PAE	RMSE	Nnz
2	LINEAL	2	0.863	15349	8	LINEAL	8	2.469	6776
	WENO- I^{3a}	2	0.824	12601		WENO- I^{3a}	8	2.064	6045
	WENO- I^{3b}	2	0.826	12570		WENO- I^{3b}	8	2.061	6083
	WENO- I^{3c}	2	0.826	12657		WENO- I^{3c}	8	2.066	6081
4	LINEAL	4	1.360	10764	10	LINEAL	10	2.931	5936
	WENO- I^{3a}	4	1.272	8837		WENO- I^{3a}	10	2.429	5239
	WENO- I^{3b}	4	1.281	9021		WENO- I^{3b}	10	2.469	5379
	WENO- I^{3c}	4	1.277	9030		WENO- I^{3c}	10	2.464	5326
6	LINEAL	6	1.958	8695	12	LINEAL	12	3.344	5254
	WENO- I^{3a}	6	1.725	7128		WENO- I^{3a}	12	2.713	4629
	WENO- I^{3b}	6	1.723	7150		WENO- I^{3b}	12	2.717	4623
	WENO- I^{3c}	6	1.728	7159		WENO- I^{3c}	12	2.714	4628

Tabla 7.3: Valores distintos de cero de la transformada cuantizada así como los errores en las diferentes normas para cada uno de los esquemas de compresión para la imagen $t102$

7.2

Compresión

Ahora vamos a comparar los resultados de compresión obtenidos con las técnicas lineal y WENO.

En este caso, la transformada 2D modificada $M_{\varepsilon}^M f^L$ obtenida con el *algoritmo de control del error para valores puntuales* (Algoritmo A.1) es comprimida utilizando el algoritmo de compresión sin pérdidas (*lossless*) PPMZ [13].

A pesar de que el objetivo de este capítulo es presentar resultados numéricos de la interpolación WENO y compararla con la interpolación lineal centrada, también hemos utilizado el estándar de compresión sin pérdidas *JPEG Lossless (JLS)* (ver [47]), el cual tiene un modo casi-sin pérdidas (*near lossless*), denominado **JPEG-LS**, donde el error máximo permitido puede ser especificado antes de comprimir, igual que el algoritmo que nosotros utilizamos.

Nuestros estudios numéricos están basados en resultados obtenidos con cuatro tipos de imágenes:

- Una imagen geométrica (Fig. 7.3(a)).
- Una imagen médica (Fig. 7.3(b)).
- Una imagen real (Fig. 7.3(c)).
- Una imagen geométrica con ruido (Fig. 7.3(d)).

En todos nuestros tests numéricos tomamos $L = 4$, $N_L = 512 \times 512$.

Imagen geométrica

Empezamos considerando una imagen geométrica simple. Consideramos la función bidimensional de Harten, definida por:

$$f(x, y) = \begin{cases} u_{\sqrt{\pi/2}}(x, y), & x \leq \frac{1}{2}\cos(\pi y) \\ u_{-\sqrt{\pi/2}}(x, y) + \cos(2\pi y), & x > \frac{1}{2}\cos(\pi y) \end{cases} \quad (7.1)$$

donde u_ϕ es

$$u_\phi(x, y) = \begin{cases} -r \sin(\frac{\pi}{2}r^2), & r \leq -\frac{1}{3} \\ |\sin(\pi 2r)|, & r < \frac{1}{3} \\ 2r - 1 + \frac{1}{6} \sin(3\pi r), & r \geq \frac{1}{3} \end{cases}$$

y $r = x + \tan(\phi)y$ (Fig. 7.3(a)).

Esta función está compuesta por partes suaves separadas por saltos y esquinas.

En la Figura 7.4, mostramos el error L^2 (RMSE) versus el *bit-per-pixel* rate y el bit rate versus el error L^∞ (peak error).

En la Tabla 7.4, mostramos el bit per pixel de la transformada codificada así como los errores en las diferentes normas para cada uno de los esquemas de compresión basados en multirresolución que nosotros consideramos.

En este escenario la compresión conseguida con el algoritmo WENO es significativamente mayor que el esquema LINEAL correspondiente. En el caso del esquema JPEG-LS, la mejora es incluso más impresionante.

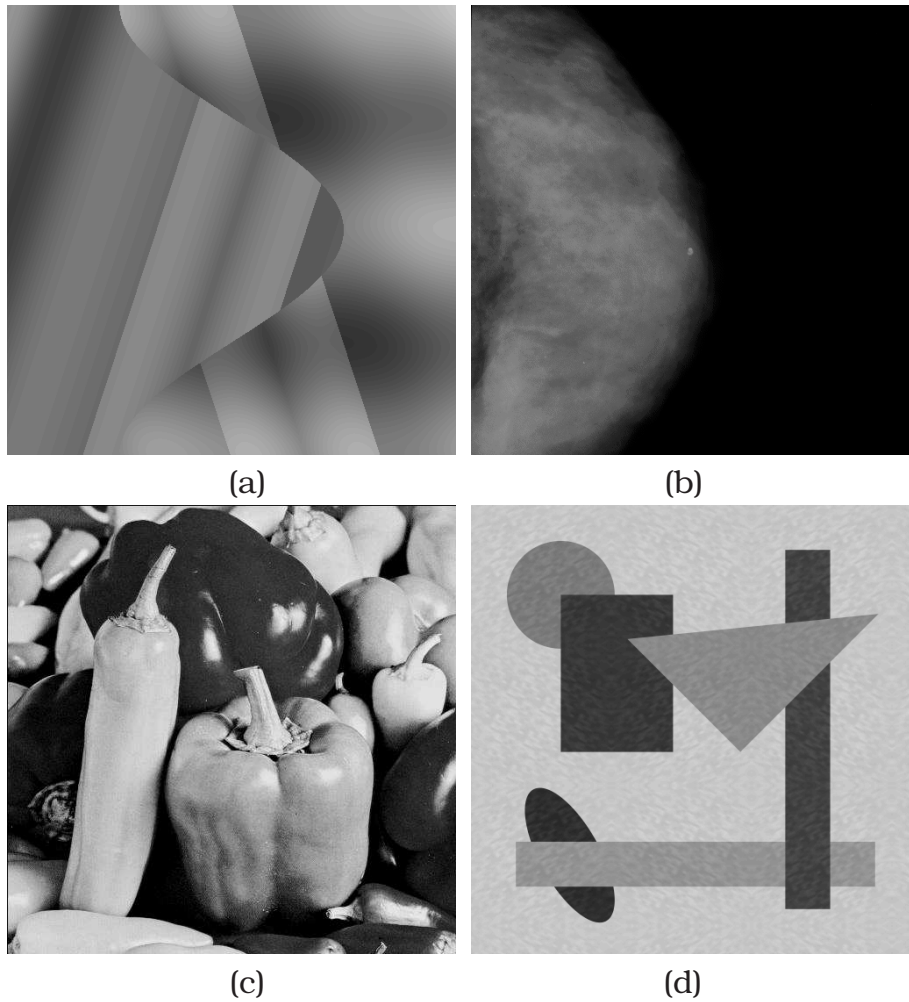


Figura 7.3: (a) Función 2D de Harten. (b) Imagen médica (c) Pimientos e (d) Imagen geométrica con ruido.

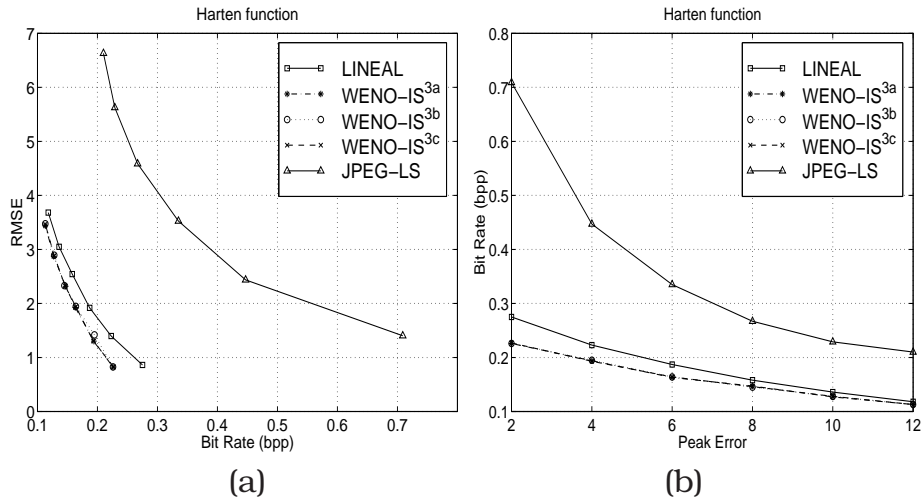


Figura 7.4: Comparación de los resultados obtenidos con la función de Harten. Reconstrucción lineal (LINEAL), Reconstrucción WENO con diferentes indicadores de suavidad. JPEG-LS (near-lossless): (a) RMSE vs bit rate; (b) bit-rate versus peak-error.

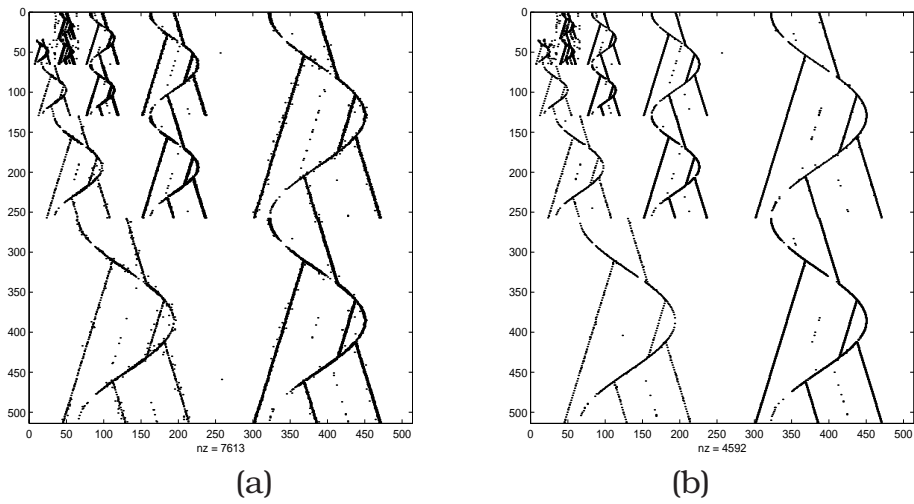


Figura 7.5: Localización (y número total) de los coeficientes de escala distintos de cero en la representación multiresolutiva después de cuantizar con $\varepsilon = 10$. (a) LINEAL. (b) WENO- I^{3b}

ε	step	PAE	RMSE	bpp	ε	step	PAE	RMSE	bpp
2	LINEAL	2	0.862	0.275	8	LINEAL	8	2.543	0.158
	WENO- I^{3a}	2	0.821	0.226		WENO- I^{3a}	8	2.319	0.147
	WENO- I^{3b}	2	0.826	0.226		WENO- I^{3b}	8	2.330	0.145
	WENO- I^{3c}	2	0.824	0.227		WENO- I^{3c}	8	2.327	0.146
	JPEG-LS	2	1.402	0.709		JPEG-LS	8	4.585	0.267
4	LINEAL	4	1.397	0.223	10	LINEAL	10	3.050	0.136
	WENO- I^{3a}	4	1.303	0.194		WENO- I^{3a}	10	2.873	0.127
	WENO- I^{3b}	4	1.311	0.193		WENO- I^{3b}	10	2.893	0.128
	WENO- I^{3c}	4	1.306	0.193		WENO- I^{3c}	10	2.883	0.128
	JPEG-LS	4	2.436	0.447		JPEG-LS	10	5.625	0.229
6	LINEAL	6	1.919	0.187	8	LINEAL	12	3.683	0.118
	WENO- I^{3a}	6	1.927	0.163		WENO- I^{3a}	12	3.437	0.113
	WENO- I^{3b}	6	1.946	0.164		WENO- I^{3b}	12	3.475	0.113
	WENO- I^{3c}	6	1.941	0.164		WENO- I^{3c}	12	3.467	0.113
	JPEG-LS	6	3.525	0.335		JPEG-LS	12	6.631	0.210

Tabla 7.4: Imagen de Harten. Errores y ratios de compresión (bpp) (re-construcción con detalles utilizando Algoritmo A.2)

En la Figura 7.5 mostramos la localización de los coeficientes de escala distintos de cero en la representación multirresolutiva cuantizada. Como la figura está compuesta por regiones suaves separadas por discontinuidades, los únicos coeficientes distintos de cero están concentrados alrededor de las discontinuidades. Notemos que el número de coeficientes de escala distintos de cero para el esquema LINEAL es mayor que los obtenidos con el esquema WENO.

Imagen Médica

También hemos procesado la imagen de la mamografía mostrada en la Figura 7.3(b).

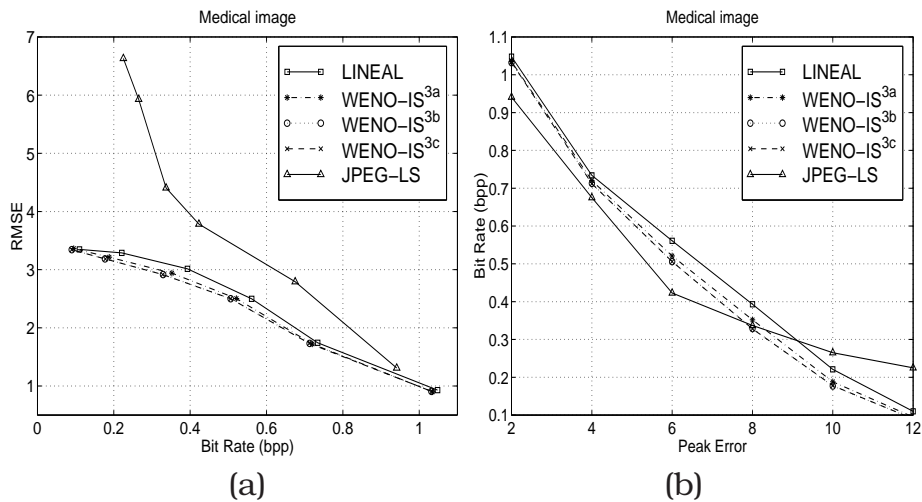


Figura 7.6: Comparación de los resultados obtenidos con la imagen médica. Reconstrucción lineal (LINEAL), Reconstrucción WENO con diferentes indicadores de suavidad. JPEG-LS (near-lossless). (a) RMSE vs bit rate; (b) bit-rate versus peak-error.

En la Figura 7.6, mostramos el error L^2 (RMSE) versus the *bit-per-pixel* rate y el bit rate versus the error L^∞ (peak error).

En la Tabla 7.5 mostramos el bit per pixel de la transformada codificada así como el error en diferentes normas para cada uno de los esquemas de compresión basados en la multirresolución que hemos considerado.

La compresión obtenida con el algoritmo WENO es significativamente mejor que la correspondiente al esquema LINEAL. En el caso del esquema JPEG-LS, la ganancia es más impresionante.

ε	step	PAE	RMSE	bpp	ε	step	PAE	RMSE	bpp
2	LINEAL	2	0.930	1.048	8	LINEAL	8	3.015	0.393
	WENO- I^{3a}	2	0.900	1.035		WENO- I^{3a}	8	2.940	0.352
	WENO- I^{3b}	2	0.908	1.032		WENO- I^{3b}	8	2.916	0.329
	WENO- I^{3c}	2	0.907	1.033		WENO- I^{3c}	8	2.916	0.328
	JPEG-LS	2	1.310	0.941		JPEG-LS	8	4.406	0.337
4	LINEAL	4	1.744	0.734	10	LINEAL	10	3.289	0.221
	WENO- I^{3a}	4	1.724	0.720		WENO- I^{3a}	10	3.215	0.188
	WENO- I^{3b}	4	1.732	0.713		WENO- I^{3b}	10	3.186	0.177
	WENO- I^{3c}	4	1.732	0.712		WENO- I^{3c}	10	3.184	0.178
	JPEG-LS	4	2.793	0.675		JPEG-LS	10	5.930	0.265
6	LINEAL	6	2.499	0.561	12	LINEAL	12	3.348	0.110
	WENO- I^{3a}	6	2.504	0.521		WENO- I^{3a}	12	3.341	0.092
	WENO- I^{3b}	6	2.501	0.506		WENO- I^{3b}	12	3.343	0.090
	WENO- I^{3c}	6	2.501	0.506		WENO- I^{3c}	12	3.342	0.090
	JPEG-LS	6	3.784	0.423		JPEG-LS	12	6.631	0.225

Tabla 7.5: Imagen Médica. Errores y ratios de compresión (bpp) (reconstrucción con detalles utilizando Algoritmo A.2)

Las figuras obtenidas para cada una de las representaciones multirresolutivas *reconstruidas* (Figura 7.7) son “visualmente similares”, y esto es debido al hecho de que el error en la norma 2 es aproximadamente del mismo orden en los tres casos. La norma 2 obtenida con el esquema LINEAL es un poco mayor debido al emborronamiento en las discontinuidades que se obtiene cuando utilizamos técnicas de interpolación centradas en zonas que tienen discontinuidades (fenómeno de Gibbs).

A pesar de que nuestra técnica de compresión no es muy sofisticada, los ratios de compresión obtenidos con ambos algoritmos son bastante parecidos. Hay, sin embargo, una diferencia significativa en el RMSE, y en la calidad de las imágenes descomprimidas (ver Figura 7.7). En la imagen original podemos observar claramente un punto blanco. Si esto significa un tumor o una anomalía

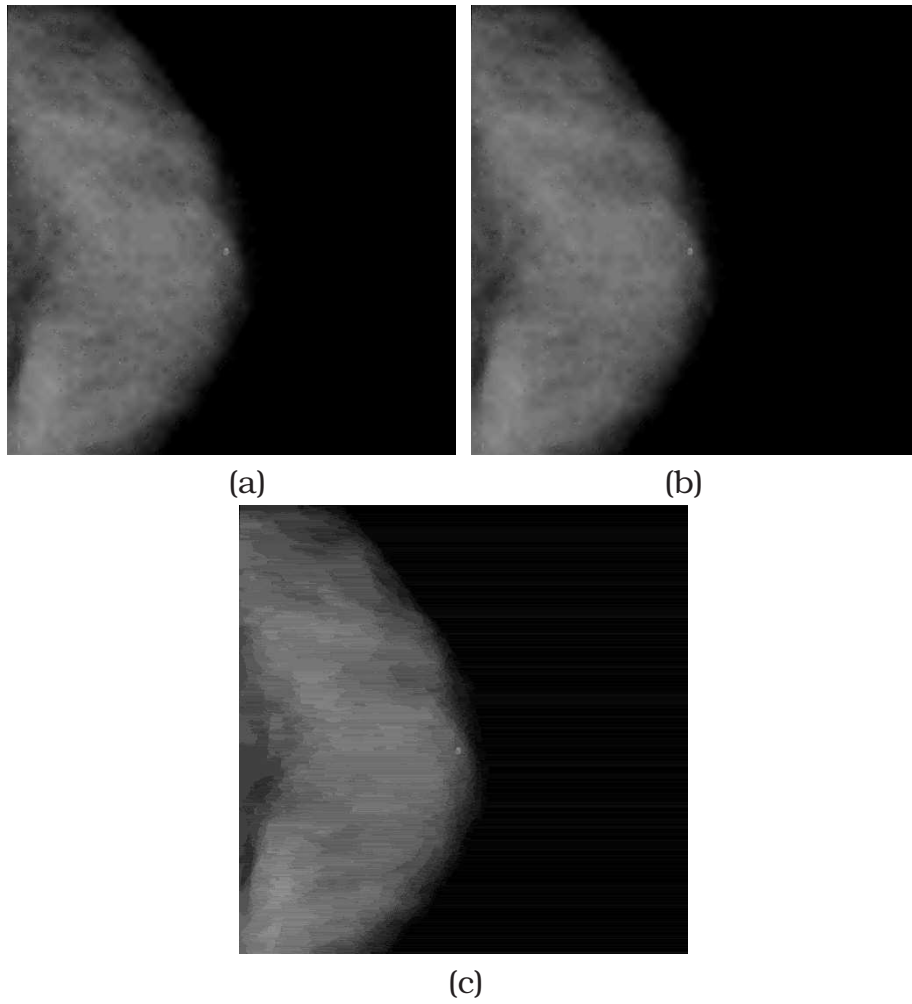


Figura 7.7: Imágenes reconstruidas a partir de las representaciones comprimidas con $\varepsilon = 12$. (a) LINEAL. (b) WENO-I^{3b}. (c) JPEG-LS

esto deberá ser claramente observado después de que la imagen sea descomprimida, como ocurre en todos los casos. La mayor diferencia está en la *textura* de la mamografía, la cual tiene mejor apariencia visual con los algoritmos que aquí presentamos.

Pimientos

Consideramos ahora una imagen real (Pimientos) (ver Tabla 7.6, Figura 7.8 y Figura 7.9). Observamos que con la interpolación WENO- I^3 obtenemos resultados mejores que con la interpolación LINEAL.

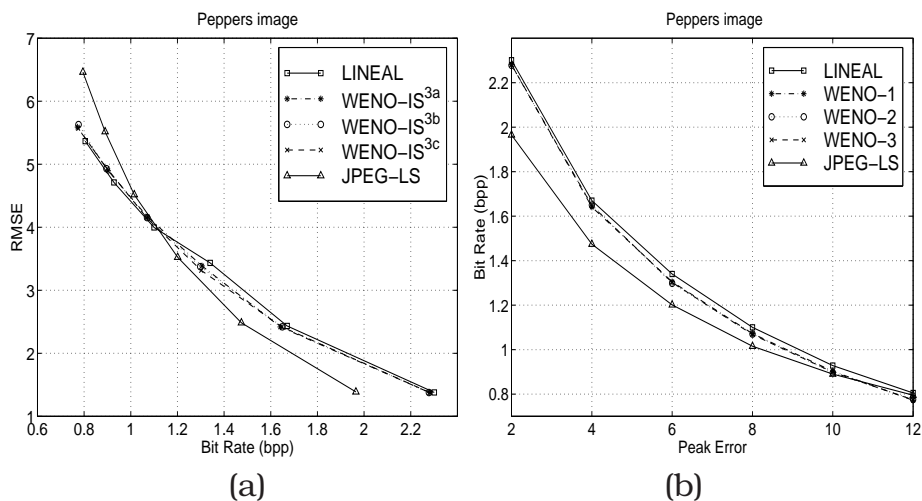


Figura 7.8: Comparación de la compresión en el test de la imagen Pimientos. Reconstrucción LINEAL, Reconstrucción WENO con diferentes indicadores de suavidad. JPEG-LS (near-lossless): (a) RMSE vs bit rate; (b) bit-rate versus peak-error.

En este caso, las diferencias entre los resultados obtenidos con dichas técnicas no es tan grande como las observadas en los casos anteriores. La razón es que las imágenes reales usualmente no tienen grandes regiones de suavidad y la compresión WENO puede sólo obtener ventajas de su alto orden en regiones suaves cuando estas son grandes.

ε	step	PAE	RMSE	bpp	ε	step	PAE	RMSE	bpp
2	LINEAL	2	1.379	2.301	8	LINEAL	8	3.997	1.100
	WENO- I^{3a}	2	1.377	2.281		WENO- I^{3a}	8	4.165	1.072
	WENO- I^{3b}	2	1.378	2.278		WENO- I^{3b}	8	4.154	1.070
	WENO- I^{3c}	2	1.378	2.281		WENO- I^{3c}	8	4.152	1.068
	JPEG-LS	2	1.388	1.965		JPEG-LS	8	4.517	1.015
4	LINEAL	4	2.435	1.670	10	LINEAL	10	4.711	0.928
	WENO- I^{3a}	4	2.419	1.641		WENO- I^{3a}	10	4.897	0.903
	WENO- I^{3b}	4	2.420	1.650		WENO- I^{3b}	10	4.929	0.896
	WENO- I^{3c}	4	2.420	1.647		WENO- I^{3c}	10	4.931	0.899
	JPEG-LS	4	2.486	1.474		JPEG-LS	10	5.518	0.890
6	LINEAL	6	3.435	1.340	12	LINEAL	12	5.367	0.805
	WENO- I^{3a}	6	3.386	1.305		WENO- I^{3a}	12	5.590	0.772
	WENO- I^{3b}	6	3.375	1.298		WENO- I^{3b}	12	5.629	0.776
	WENO- I^{3c}	6	3.316	1.302		WENO- I^{3c}	12	5.570	0.774
	JPEG-LS	6	3.524	1.201		JPEG-LS	12	6.461	0.795

Tabla 7.6: Imagen pimientos. Errores y ratios de compresión (bpp) (reconstrucción con detalles utilizando Algoritmo A.2)

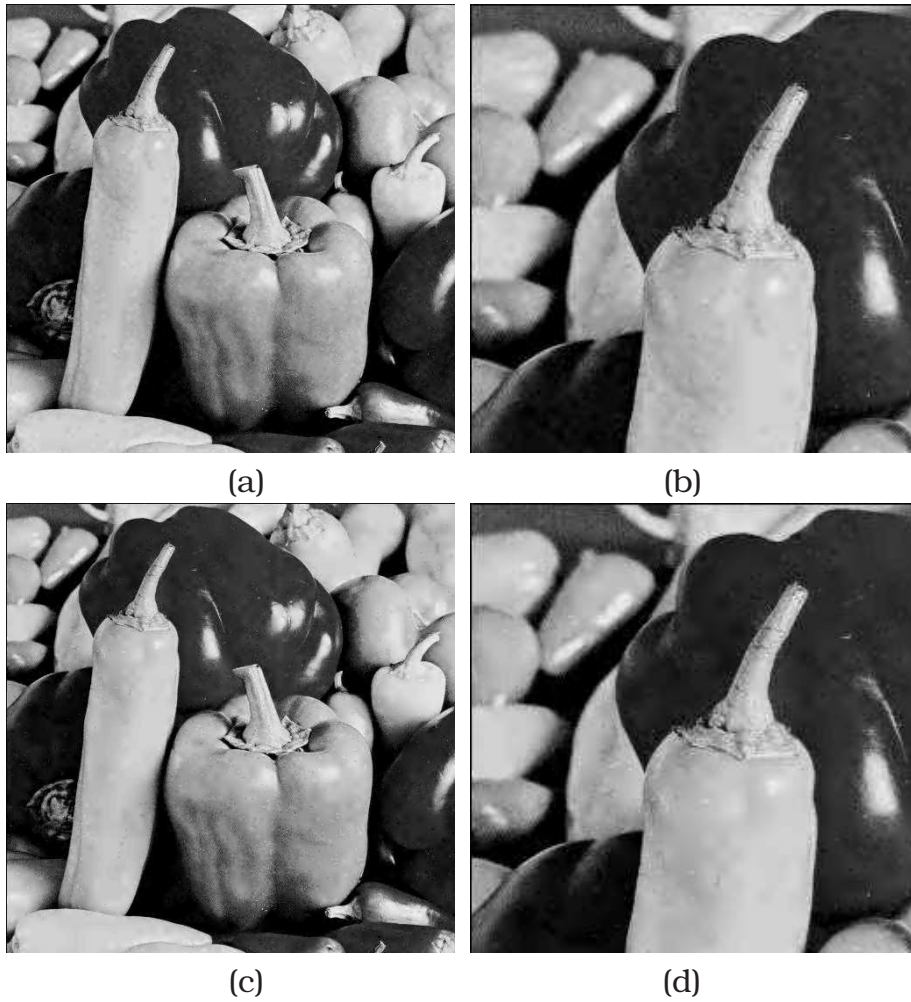


Figura 7.9: Imágenes reconstruidas a partir de las representaciones comprimidas con $\varepsilon = 12$. (a) Reconstrucción LINEAL (c) Reconstrucción WENO- I^{3b} . (b) y (d) Esquinas superior izquierda de (a) y (c) respectivamente.

Imagen geométrica con ruido

Finalizamos con la imagen geométrica con ruido (Fig. 7.3(d)), compuesta por partes con textura separadas por saltos.

En la Figura 7.10, mostramos el error L^2 (RMSE) versus el *bit-per-pixel* rate y el bit rate versus el error L^∞ (peak error).

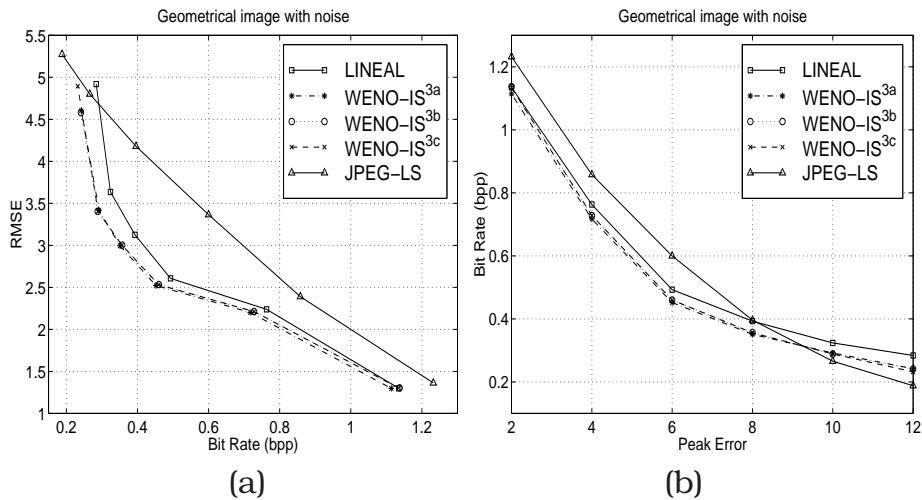


Figura 7.10: Comparación de la compresión en el test de la imagen geométrica con ruido. Reconstrucción LINEAL, Reconstrucción WENO con diferentes indicadores de suavidad. JPEG-LS (near-lossless): (a) RMSE vs bit rate; (b) bit-rate versus peak-error.

En la Tabla 7.7, mostramos el bit per pixel de la transformada codificada así como los errores en las diferentes normas para cada uno de los esquemas de compresión basados en multirresolución que nosotros consideramos.

Podemos observar que la compresión conseguida con el algoritmo WENO es significativamente mejor que con el esquema LINEAL correspondiente y con el esquema JPEG-LS.

La Figura 7.11 muestra las imágenes reconstruidas (utilizando el Algoritmo A.2) a partir de las representaciones comprimidas obtenidas con el Algoritmo A.1 cuantizando con $\varepsilon = 10$ y utilizando las reconstrucciones LINEAL Figura 7.11(a) (b) y Figura 7.11(c) (d) WENO- I^{3b} . En las figuras se observa que la calidad de la imagen

reconstruida cuando se utiliza la reconstrucción WENO- I^{3b} es mucho mejor. En la tabla vemos que en efecto el error es más grande en el caso LINEAL ($RMSE = 3,636$) que en el caso WENO- I^{3b} , donde $RMSE = 3,406$. Si a esto añadimos que la tasa de compresión obtenida con WENO- I^{3b} es mejor ($bpp = 0,288$) que la obtenida con LINEAL ($bpp = 0,324$) llegamos a la conclusión de que en este caso WENO- I^{3b} es la reconstrucción con la que obtenemos mejores resultados.

ε	step	PAE	RMSE	bpp	ε	step	PAE	RMSE	bpp
2	LINEAL	2	1.299	1.135	8	LINEAL	8	3.126	0.393
	WENO- I^{3a}	2	1.298	1.114		WENO- I^{3a}	8	2.995	0.351
	WENO- I^{3b}	2	1.305	1.137		WENO- I^{3b}	8	3.003	0.357
	WENO- I^{3c}	2	1.304	1.134		WENO- I^{3c}	8	3.004	0.356
	JPEG-LS	2	1.363	1.232		JPEG-LS	8	4.179	0.396
4	LINEAL	4	2.239	0.763	10	LINEAL	10	3.636	0.324
	WENO- I^{3a}	4	2.202	0.707		WENO- I^{3a}	10	3.416	0.292
	WENO- I^{3b}	4	2.212	0.728		WENO- I^{3b}	10	3.406	0.288
	WENO- I^{3c}	4	2.212	0.728		WENO- I^{3c}	10	3.407	0.289
	JPEG-LS	4	2.390	0.858		JPEG-LS	10	4.800	0.266
6	LINEAL	6	2.608	0.493	12	LINEAL	12	4.919	0.284
	WENO- I^{3a}	6	2.522	0.452		WENO- I^{3a}	12	4.602	0.242
	WENO- I^{3b}	6	2.534	0.460		WENO- I^{3b}	12	4.581	0.241
	WENO- I^{3c}	6	2.532	0.460		WENO- I^{3c}	12	4.578	0.242
	JPEG-LS	6	3.365	0.600		JPEG-LS	12	5.272	0.188

Tabla 7.7: Imagen Geométrica con ruido. Errores y ratios de compresión (bpp) (reconstrucción con detalles utilizando Algoritmo A.2)

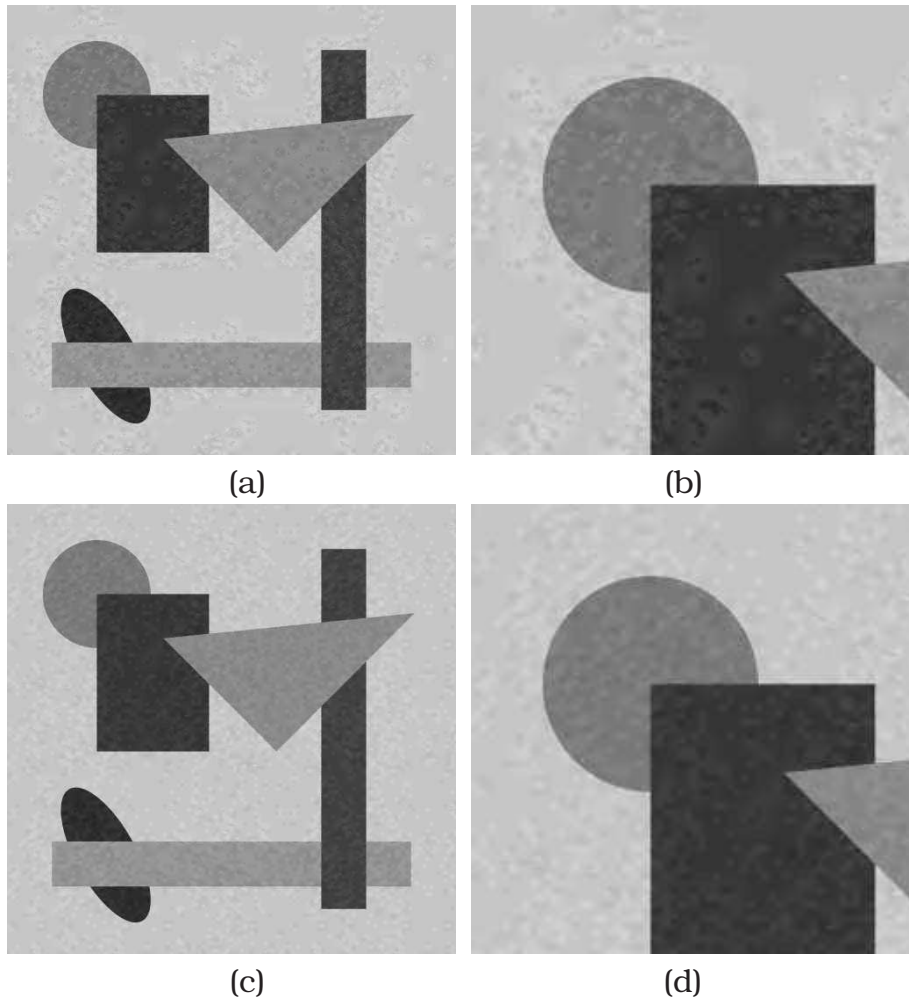


Figura 7.11: Imágenes reconstruidas a partir de las representaciones comprimidas con $\varepsilon = 12$. (a) Reconstrucción LINEAL (c) Reconstrucción WENO- I^{3b} . (b) y (d) Esquinas superior izquierda de (a) y (c) respectivamente.

8

Conclusiones y perspectivas

En el presente trabajo de tesis doctoral se ha analizado la técnica de interpolación WENO. Esta técnica, inicialmente desarrollada para la interpolación de valores puntuales a partir de medias en celda, ha sido adaptada, por una parte, al contexto de interpolación de medias en celda, de forma que se utilizan datos discretos de medias en celda para interpolar medias en celda, y por otra parte se ha adaptado también al contexto de interpolación de valores puntuales, para recuperar un conjunto de valores puntuales de una función a partir de valores puntuales en un nivel de resolución inferior.

La técnica de interpolación WENO se ha utilizado como operador predicción dentro del entorno de multirresolución de Harten, y se ha comparado su capacidad de compresión con otras técnicas de interpolación.

En el contexto de medias en celda, se ha propuesto una modificación del indicador de suavidad definido por Jiang y Shu [27] que mejora los resultados numéricos obtenidos con los indicadores de suavidad presentes en la literatura, aunque pierde la emulación del comportamiento ENO. En los experimentos numéricos realizados, el interpolante construido a partir de este indicador modificado proporciona resultados similares, aunque en general mejores, que la interpolación lineal.

También hemos comprobado numéricamente que la interpolación WENO aplicada por producto tensorial a imágenes bidimensionales resulta ser un método mucho más estable que las técnicas de interpolación no lineales consideradas (ENO, ENO-SR).

Hemos modificado los indicadores de suavidad para medias en celda de la literatura sobre WENO para aplicarlos en el contexto de valores puntuales. La adaptación del indicador de Jiang y Shu proporciona un orden óptimo de exactitud en regiones suaves de la función interpolada, pero hemos visto que si en algún punto se tiene $f' = f'' = 0$ el orden de la aproximación se reduce. Esto ha motivado la investigación desarrollada en el Capítulo 6.

Después de un estudio sobre el efecto que sobre el orden de la aproximación tiene la elección de ε en la definición de los pesos WENO hemos llegado a la conclusión de que tomar un ε variable $\varepsilon(h) = h^2$ en lugar de un $\varepsilon = 10^{-6}$ fijo, como se ha venido haciendo en la literatura, asegura un orden óptimo de aproximación en las zonas suaves de la función interpolada, independientemente de que en algún punto se anulen las dos primeras derivadas. También hemos probado que incluso en presencia de una discontinuidad en la función, con el interpolante WENO se obtiene al menos un orden de interpolación $r + 1$, es decir, una aproximación como mínimo tan buena como la obtenida con interpolación ENO.

Entre las líneas de investigación que se podrían considerar como continuación de este trabajo se encuentra estudiar la extensión a dos dimensiones de la interpolación WENO en su aplicación al procesamiento de imágenes, en lugar de la extensión vía producto tensorial de la interpolación unidimensional que se utiliza en este trabajo.

Por otra parte, no disponemos de resultados teóricos que confirmen la estabilidad de la interpolación WENO en 2D, aunque

los experimentos numéricos analizados en el Capítulo 3 parecen indicar que es un método bastante estable.

También sería interesante extender las aplicaciones vistas en el tratamiento de imágenes a imágenes en color, donde la multirresolución se tendría que aplicar separadamente a los tres canales de color de la imagen, y a la codificación de video, donde las discontinuidades aparecen tanto en el espacio como en el tiempo, y se tendría que aplicar la representación multiescala para estos dos aspectos.

Bibliografía

- [1] S. Amat, F. Aràndiga, A. Cohen, R. Donat, *Tensor product multiresolution analysis with error control*, Signal Processing **82**, 587–608 (2002).
- [2] S. Amat, F. Aràndiga, A. Cohen, R. Donat, G. García, M. von Oehsen, *Data compression with ENO schemes: a case study*, Applied and Computational Harmonic Analysis **11 (2)**, 273–288 (2001).
- [3] F. Aràndiga, J. Baccou, M. Doblus, J. Liandrat, *Image compression based on a multi-directional map-dependent algorithm*, Applied and Computational Harmonic Analysis **23**, 181–197 (2007).
- [4] F. Aràndiga, A. M. Belda, *Weighted ENO interpolation and applications*, Communications in Nonlinear Science and Numerical Simulation **9 (2)** 187–195 (2003).
- [5] F. Aràndiga, A. M. Belda, P. Mulet, *Point-value ENO interpolation and multiresolution application*, Submitted (2009).
- [6] F. Aràndiga, V. Candela, *Multiresolution standard form of a matrix*, SIAM Journal of Numerical Analysis **33**, 417–434 (1996).
- [7] F. Aràndiga, V. Candela, R. Donat, *Fast multiresolution algorithms for solving linear equations: A comparative study*, SIAM Journal on Scientific Computing **16**, 581–600 (1995).

- [8] F. Aràndiga, A. Cohen, R. Donat, N. Dyn, *Interpolation and approximation of piecewise smooth functions*, SIAM Journal of Numerical Analysis **43 (1)**, 41–57 (2005).
- [9] F. Aràndiga, R. Donat, *Nonlinear multiscale decompositions: The approach of A. Harten*, Numerical Algorithms **23**, 175–216 (2000).
- [10] F. Aràndiga, R. Donat, A. Harten, *Multiresolution based on weighted averages of the hat function I: Linear reconstruction operators*, SIAM Journal of Numerical Analysis, **36 (1)**, 160–203 (1999).
- [11] F. Aràndiga, R. Donat, A. Harten, *Multiresolution based on weighted averages of the hat function II: Nonlinear reconstruction operators*, SIAM Journal on Scientific Computing **20 (3)**, 1053–1093 (1999).
- [12] A. Baeza, P. Mulet, *Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations*, International Journal for Numerical Methods in Fluids **52 (4)**, 455–471 (2006).
- [13] C. Bloom, *Solving the problems of context modeling*, <http://www.cbloom.com/papers/> (1998).
- [14] R. L. Claypoole, G. Davis, W. Sweldens, R. Baraniuk, *Nonlinear wavelet transforms for image coding via lifting*, IEEE Transactions on Image Processing **12**, 1449–1459 (2003).
- [15] A. Cohen, N. Dyn, B. Matei, *Quasilinear subdivision schemes with applications to ENO interpolation*, Applied and Computational Harmonic Analysis **15 (2)**, 89–116 (2003).
- [16] K. Dadourian, *Schémas de subdivision, analyses multirésolutions non-linéaires. Applications*, PhD thesis, Université de Provence. U.F.R. M.I.M. Marseilles (2008).
- [17] R. Donat, *Studies on error propagation for certain nonlinear approximations to hyperbolic equations: discontinuities in derivative*, SIAM Journal of Numerical Analysis **31**, 655–679 (1994).

- [18] E. Fatemi, J. Jerome, S. Osher, *Solution of the hydrodynamic device model using high order non-oscillatory shock capturing algorithms*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **10**, 232–244 (1991).
- [19] O. Friedrich, *Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids*, Journal of Computational Physics **144**, 194–212 (1998).
- [20] A. Harten, *ENO Schemes with subcell resolution*, Journal of Computational Physics **83**, 148–184 (1989).
- [21] A. Harten. *Discrete multiresolution analysis and generalized wavelets*, Journal of Applied Numerical Mathematics **12**, 153–193 (1993).
- [22] A. Harten, *Multiresolution representation of cell-averaged data*, Technical Report, UCLA CAM Report 94–21 (1994).
- [23] A. Harten, *Multiresolution representation of data II: General framework*, SIAM Journal of Numerical Analysis **33**, 1205–1256 (1996).
- [24] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes III*, Journal of Computational Physics **71**, 231–303 (1987).
- [25] A. K. Henrick, T. D. Aslam, J. M. Powers, *Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points*, Journal of Computational Physics **207 (2)**, 542–567 (2005).
- [26] C. Hu, C.-W. Shu, *Weighted essentially non-oscillatory schemes on triangular meshes*, Journal of Computational Physics **150**, 97–127 (1999).
- [27] G.-S. Jiang, C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics **126**, 202–228 (1996).

- [28] P. D. Lax, X.-D. Liu, *Solution of two-dimensional Riemann problems of gas dynamics by positive schemes*, SIAM Journal on Scientific Computing **19** (2), 319–340 (1998).
- [29] D. Levy, G. Puppo, G. Russo, *Central WENO schemes for hyperbolic systems of conservation laws*, Mathematical Modelling and Numerical Analysis **33** (3), 547–571 (1999).
- [30] D. Levy, G. Puppo, G. Russo, *Compact central WENO schemes for multidimensional conservation laws*, SIAM Journal on Scientific Computing **22** (2), 656–672 (2000).
- [31] D. Levy, G. Puppo, G. Russo, *A third order central WENO scheme for 2D conservation laws*, Applied Numerical Mathematics **33**, 415–421 (2000).
- [32] D. Levy, G. Puppo, G. Russo, *A fourth order central WENO scheme for multidimensional hyperbolic systems of conservation laws*, SIAM Journal on Scientific Computing **24** (2), 480–506 (2002).
- [33] X.-D. Liu, S. Osher, T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics **115**, 200–212 (1994).
- [34] B. Matei, *Méthodes multirésolutions non-linéaires: Applications au traitement d'image*, PhD thesis, Université Pierre et Marie Curie, Paris (2003).
- [35] C. F. Ollivier-Gooch, *Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction*, Journal of Computational Physics **133**, 6–17 (1997).
- [36] G. Piella, H. J. A. M. Heijmans, B. Pesquet-Popescu, *Adaptive update lifting with a decision rule based on derivative filters*, IEEE Signal Processing Letters **9**, 329–332 (2002).
- [37] J. Qiu, C.-W. Shu, *Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case*, Journal of Computational Physics **193** (1), 115–135 (2004).

- [38] J. Qiu, C.-W. Shu, *Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two dimensional case*, Computational Fluids **34**, 642–663 (2005).
- [39] J. Qiu, C.-W. Shu, *Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method III: unstructured meshes*, Journal of Scientific Computing **39 (2)**, 293–321 (2009).
- [40] C. W. Schulz-Rinne, J. P. Collins, H. M. Glaz, *Numerical solution of the Riemann problem for two-dimensional gas dynamics*, SIAM Journal on Scientific Computing **14 (6)**, 1394–1414 (1993).
- [41] C.-W. Shu, *Numerical experiments on the accuracy of ENO and modified ENO schemes*, Journal on Scientific Computing **5**, 127–149 (1990).
- [42] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, A. Quarteroni, ed., Lecture Notes in Mathematics, Vol. 1697, Springer, Berlin, 325–432 (1998).
- [43] C.-W. Shu, S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics **77**, 439–471 (1988).
- [44] C.-W. Shu, S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes II*, Journal of Computational Physics **83**, 32–78 (1989).
- [45] W. Sweldens, *The lifting scheme: a customer-design construction of biorthogonal wavelets*, Applied and Computational Harmonic Analysis **3 (2)**, 186–200 (1996).
- [46] W. Sweldens, *The lifting scheme: a construction of second generation wavelets*, SIAM Journal on Mathematical Analysis **29 (2)**, 511–546 (1998).

- [47] M. Weinberger, G. Seroussi, G. Sapiro, *From LOCO-I to the JPEG-LS standard*, Proceedings of the International Conference on Image Processing, Vol. 4, 68–72 (1999).

A

Algoritmos 2D de control del error para valores puntuales

Los algoritmos que utilizaremos en 2D para la transformada directa e inversa para valores puntuales son:

Algoritmo A.1. *Codificación para valores puntuales*

```
for  $k = L, \dots, 1$ 
  for  $i = 0, \dots, N_k$ 
    for  $j = 0, \dots, N_{k-1}$ 
       $f_{i,j}^{k-\frac{1}{2}} = f_{i,2j}^k$ 
    end
  end
end
for  $i, j = 0, \dots, N_{k-1}$ 
```

```

       $f_{i,j}^{k-1} = f_{2i,j}^{k-\frac{1}{2}}$ 
    end
  end
  Set  $\hat{f}^0 = \mathbf{proc}(f^0, \epsilon_0)$ 
  for  $k = 1, \dots, L$ 
    for  $j = 0, \dots, N_{k-1}$ 
      for  $i = 0, \dots, N_{k-1}$ 
         $u(i) = \hat{f}_{i,j}^{k-1}$ 
      end
      for  $i = 1, \dots, N_{k-1}$ 
         $\tilde{d}_{i,j}^k(2) = f_{2i-1,j}^{k-\frac{1}{2}} - (P_{k-1}^k u)_{2i-1,j}$ 
      end
       $\hat{d}_{i,j}^k(2) = \mathbf{proc}(\tilde{d}_{i,j}^k(2), \epsilon_k)$ 
      for  $i = 1, \dots, N_{k-1}$ 
         $\hat{f}_{2i-1,j}^{k-\frac{1}{2}} = (P_{k-1}^k u)_{2i-1,j} + \hat{d}_{i,j}^k(2)$ 
      end
      for  $i = 0, \dots, N_{k-1}$ 
         $\hat{f}_{2i,j}^{k-\frac{1}{2}} = \hat{f}_{i,j}^{k-1}$ 
      end
    end
    for  $i = 1, \dots, N_{k-1}$ 
      for  $j = 0, \dots, N_{k-1}$ 
         $u(j) = \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$ 
      end
      for  $j = 1, \dots, N_{k-1}$ 
         $\tilde{d}_{i,j}^k(1) = f_{2i-1,2j-1}^k - (P_{k-1}^k u)_{2i-1,2j-1}$ 
      end
       $\hat{d}_{i,j}^k(1) = \mathbf{proc}(\tilde{d}_{i,j}^k(1), \epsilon_k)$ 
      for  $j = 1, \dots, N_{k-1}$ 
         $\hat{f}_{2i-1,2j-1}^k = (P_{k-1}^k u)_{2i-1,2j-1} + \hat{d}_{i,j}^k(1)$ 
      end
      for  $j = 0, \dots, N_{k-1}$ 
         $\hat{f}_{2i-1,2j}^k = \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$ 
      end
    end
  end
  for  $i = 0, \dots, N_{k-1}$ 

```



```

for  $j = 0, \dots, N_{k-1}$ 
     $u(j) = \hat{f}_{2i,j}^{k-\frac{1}{2}}$ 
end
for  $j = 1, \dots, N_{k-1}$ 
     $\tilde{d}_{i,j}^k(3) = f_{2i,2j-1}^k - (P_{k-1}^k u)_{2i,2j-1}$ 
end
 $\hat{d}_{i,j}^k(3) = \mathbf{proc}(\tilde{d}_{i,j}^k(3), \epsilon_k)$ 
for  $j = 1, \dots, N_{k-1}$ 
     $\hat{f}_{2i,2j-1}^k = (P_{k-1}^k u)_{2i,2j-1} + \hat{d}_{i,j}^k(3)$ 
end
for  $j = 0, \dots, N_{k-1}$ 
     $\hat{f}_{2i,2j}^k = \hat{f}_{2i,j}^{k-\frac{1}{2}}$ 
end
end
end
end

```

Algoritmo A.2. *Descodificación para valores puntuales*

```

for  $k = 1, \dots, L$ 
    for  $j = 0, \dots, N_{k-1}$ 
        for  $i = 0, \dots, N_{k-1}$ 
             $u(i) = \hat{f}_{i,j}^{k-1}$ 
        end
        for  $i = 1, \dots, N_{k-1}$ 
             $\hat{f}_{2i-1,j}^{k-\frac{1}{2}} = (P_{k-1}^k u)_{2i-1,j} + \hat{d}_{i,j}^k(2)$ 
        end
        for  $i = 0, \dots, N_{k-1}$ 
             $\hat{f}_{2i,j}^{k-\frac{1}{2}} = \hat{f}_{i,j}^{k-1}$ 
        end
    end
    for  $i = 1, \dots, N_{k-1}$ 
        for  $j = 0, \dots, N_{k-1}$ 
             $u(j) = \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$ 
        end
        for  $j = 1, \dots, N_{k-1}$ 

```

```

       $\hat{f}_{2i-1,2j-1}^k = (P_{k-1}^k u)_{2i-1,2j-1} + \hat{d}_{i,j}^k (1)$ 
    end
    for  $j = 0, \dots, N_{k-1}$ 
       $\hat{f}_{2i-1,2j}^k = \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$ 
    end
  end
end
for  $i = 0, \dots, N_{k-1}$ 
  for  $j = 0, \dots, N_{k-1}$ 
     $u(j) = \hat{f}_{2i,j}^{k-\frac{1}{2}}$ 
  end
  for  $j = 1, \dots, N_{k-1}$ 
     $\hat{f}_{2i,2j-1}^k = (P_{k-1}^k u)_{2i,2j-1} + \hat{d}_{i,j}^k (3)$ 
  end
  for  $j = 0, \dots, N_{k-1}$ 
     $\hat{f}_{2i,2j}^k = \hat{f}_{2i,j}^{k-\frac{1}{2}}$ 
  end
end
end
end

```

Proposición A.1. *Dados unos valores discretos f^L , con la transformación directa modificada en 2D para valores puntuales (Algoritmo A.1) obtenemos una representación multirresolutiva*

$$Mf^L = \{\hat{f}^0, \hat{d}^1(1), \hat{d}^1(2), \hat{d}^1(3), \dots, \hat{d}^L(1), \hat{d}^L(2), \hat{d}^L(3)\}$$

tal que si aplicamos la transformada inversa (Algoritmo A.2) obtenemos \hat{f}^L que cumple:

$$\|f^L - \hat{f}^L\|_\infty \leq \max_k (|\tilde{d}^k - \hat{d}^k|_\infty, \|f^0 - \hat{f}^0\|_\infty) \quad (\text{A.1})$$

$$\|f^L - \hat{f}^L\|_1 = \frac{1}{4^L} \|f^0 - \hat{f}^0\|_1 + \sum_{k=1}^L \frac{1}{4^{L-k+1}} |\tilde{d}^k - \hat{d}^k|_1 \quad (\text{A.2})$$

$$\|f^L - \hat{f}^L\|_2^2 = \frac{1}{4^L} \|f^0 - \hat{f}^0\|_2^2 + \sum_{k=1}^L \frac{1}{4^{L-k+1}} |\tilde{d}^k - \hat{d}^k|_2^2 \quad (\text{A.3})$$

donde

$$\begin{aligned} |\tilde{d}^k - \hat{d}^k|_\infty &= \max(|\tilde{d}^k(1) - \hat{d}^k(1)|_\infty, |\tilde{d}^k(2) - \hat{d}^k(2)|_\infty, \\ &\quad |\tilde{d}^k(3) - \hat{d}^k(3)|_\infty) \\ |\tilde{d}^k - \hat{d}^k|_1 &= |\tilde{d}^k(1) - \hat{d}^k(1)|_1 + |\tilde{d}^k(2) - \hat{d}^k(2)|_1 + |\tilde{d}^k(3) - \hat{d}^k(3)|_1 \\ |\tilde{d}^k - \hat{d}^k|_2^2 &= |\tilde{d}^k(1) - \hat{d}^k(1)|_2^2 + |\tilde{d}^k(2) - \hat{d}^k(2)|_2^2 + |\tilde{d}^k(3) - \hat{d}^k(3)|_2^2 \end{aligned}$$

Demostración. Del algoritmo de codificación (Algoritmo A.1) obtenemos:

$$\begin{aligned} f_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k &= \tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2) \\ f_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k &= \tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3) \\ f_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k &= \tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1) \\ f_{2i,2j}^k - \hat{f}_{2i,2j}^k &= f_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}. \end{aligned}$$

Entonces

$$\begin{aligned}
\|f^k - \hat{f}^k\|_\infty &= \sup_{i,j} |f_{i,j}^k - \hat{f}_{i,j}^k| \\
&= \sup_{i,j} (|f_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}|, |\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)|, |\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)|, |\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)|) \\
&\leq \max(\|\tilde{d}^k - \hat{d}^k\|_\infty, \|f^{k-1} - \hat{f}^{k-1}\|_\infty)
\end{aligned}$$

y de aquí se deduce (A.1).

Como $N_k = 2N_{k-1}$, tomando $p = 1$ (o 2),

$$\begin{aligned}
\|f^k - \hat{f}^k\|_p^p &= \frac{1}{N_k^2} \sum_{i,j}^{N_k} |f_{i,j}^k - \hat{f}_{i,j}^k|^p \\
&= \frac{1}{N_k^2} \sum_{i,j}^{N_{k-1}} \left(|f_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}|^p + |\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)|^p + \right. \\
&\quad \left. |\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)|^p + |\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)|^p \right) \\
&= \frac{1}{4} \|f^{k-1} - \hat{f}^{k-1}\|_p^p + \frac{1}{4} \|\tilde{d}^k - \hat{d}^k\|_p^p
\end{aligned}$$

lo cual prueba (A.2) y (A.3).

□

B

Algoritmos 2D de control del error para medias en celda

Algoritmo B.1. *Codificación para medias en celda*

```
for  $k = L, \dots, 1$ 
  for  $i = 1, \dots, N_k$ 
    for  $j = 1, \dots, N_{k-1}$ 
       $\bar{f}_{i,j}^{k-\frac{1}{2}} = \frac{1}{2}(\bar{f}_{i,2j-1}^k + \bar{f}_{i,2j}^k)$ 
    end
  end
  for  $i, j = 1, \dots, N_{k-1}$ 
     $\bar{f}_{i,j}^{k-1} = \frac{1}{2}(\bar{f}_{2i-1,j}^{k-\frac{1}{2}} + \bar{f}_{2i,j}^{k-\frac{1}{2}})$ 
  end
```

```

end
Set  $\hat{f}^0 = \mathbf{proc}(\bar{f}^0, \epsilon_0)$ 
for  $k = 1, \dots, L$ 
  for  $j = 1, \dots, N_{k-1}$ 
    for  $i = 1, \dots, N_{k-1}$ 
       $u(i) = \hat{f}_{i,j}^{k-1}$ 
    end
    for  $i = 1, \dots, N_{k-1}$ 
       $u_{2i-1,j}^{p-\frac{1}{2}} = (P_{k-1}^k u)_{2i-1,j}$ 
       $\tilde{d}_{i,j}^k(2) = \frac{1}{2}(\bar{f}_{2i-1,2j-1}^k + \bar{f}_{2i-1,2j}^k) - u_{2i-1,j}^{p-\frac{1}{2}} - [\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}]$ 
    end
     $\hat{d}_{i,j}^k(2) = \mathbf{proc}(\tilde{d}_{i,j}^k(2), \epsilon_k)$ 
    for  $i = 1, \dots, N_{k-1}$ 
       $\hat{f}_{2i-1,j}^{k-\frac{1}{2}} = u_{2i-1,j}^{p-\frac{1}{2}} + \hat{d}_{i,j}^k(2)$ 
       $\hat{f}_{2i,j}^{k-\frac{1}{2}} = 2\hat{f}_{i,j}^{k-1} - \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$ 
    end
  end
end
for  $i = 1, \dots, N_{k-1}$ 
  for  $j = 1, \dots, N_{k-1}$ 
     $u(j) = \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$ 
  end
  for  $j = 1, \dots, N_{k-1}$ 
     $u_{2i-1,2j-1}^p = (P_{k-1}^k u)_{2i-1,2j-1}$ 
     $\tilde{d}_{i,j}^k(1) = \bar{f}_{2i-1,2j-1}^k - u_{2i-1,2j-1}^p - [\frac{1}{2}(\bar{f}_{2i-1,2j-1}^k + \bar{f}_{2i-1,2j}^k) - \hat{f}_{2i-1,j}^{k-\frac{1}{2}}]$ 
  end
   $\hat{d}_{i,j}^k(1) = \mathbf{proc}(\tilde{d}_{i,j}^k(1), \epsilon_k)$ 
  for  $j = 1, \dots, N_{k-1}$ 
     $\hat{f}_{2i-1,2j-1}^k = u_{2i-1,2j-1}^p + \hat{d}_{i,j}^k(1)$ 
     $\hat{f}_{2i-1,2j}^k = 2\hat{f}_{2i-1,j}^{k-\frac{1}{2}} - \hat{f}_{2i-1,2j-1}^k$ 
  end
end
for  $j = 1, \dots, N_{k-1}$ 
   $u(j) = \hat{f}_{2i,j}^{k-\frac{1}{2}}$ 
end
for  $j = 1, \dots, N_{k-1}$ 
   $u_{2i,2j-1}^p = (P_{k-1}^k u)_{2i,2j-1}$ 

```

```


$$\tilde{d}_{i,j}^k(3) = \bar{f}_{2i,2j-1}^k - u_{2i,2j-1}^p - [\frac{1}{2}(\bar{f}_{2i,2j-1}^k + \bar{f}_{2i,2j}^k) - \hat{f}_{2i,j}^{k-\frac{1}{2}}]$$

end

$$\hat{d}_{i,j}^k(3) = \mathbf{proc}(\tilde{d}_{i,j}^k(3), \epsilon_k)$$

for  $j = 1, \dots, N_{k-1}$ 

$$\hat{f}_{2i,2j-1}^k = u_{2i,2j-1}^p + \hat{d}_{i,j}^k(3)$$


$$\hat{f}_{2i,2j}^k = 2\hat{f}_{2i,j}^{k-\frac{1}{2}} - \hat{f}_{2i,2j-1}^k$$

end
end
end

```

Algoritmo B.2. *Decodificación para medias en celda*

```

for  $k = 1, \dots, L$ 
  for  $j = 1, \dots, N_{k-1}$ 
    for  $i = 1, \dots, N_{k-1}$ 
      
$$u(i) = \hat{f}_{i,j}^{k-1}$$

    end
    for  $i = 1, \dots, N_{k-1}$ 
      
$$u_{2i-1,j}^{p-\frac{1}{2}} = (P_{k-1}^k u)_{2i-1,j}$$

      
$$\hat{f}_{2i-1,j}^{k-\frac{1}{2}} = u_{2i-1,j}^{p-\frac{1}{2}} + \hat{d}_{i,j}^k(2)$$

      
$$\hat{f}_{2i,j}^{k-\frac{1}{2}} = 2\hat{f}_{i,j}^{k-1} - \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$$

    end
  end
  for  $i = 1, \dots, N_{k-1}$ 
    for  $j = 1, \dots, N_{k-1}$ 
      
$$u(j) = \hat{f}_{2i-1,j}^{k-\frac{1}{2}}$$

    end
    for  $j = 1, \dots, N_{k-1}$ 
      
$$u_{2i-1,2j-1}^p = (P_{k-1}^k u)_{2i-1,2j-1}$$

      
$$\hat{f}_{2i-1,2j-1}^k = u_{2i-1,2j-1}^p + \hat{d}_{i,j}^k(1)$$

      
$$\hat{f}_{2i-1,2j}^k = 2\hat{f}_{2i-1,j}^{k-\frac{1}{2}} - \hat{f}_{2i-1,2j-1}^k$$

    end
  end
  for  $j = 1, \dots, N_{k-1}$ 
    
$$u(j) = \hat{f}_{2i,j}^{k-\frac{1}{2}}$$

  end
end

```

```

for  $j = 1, \dots, N_{k-1}$ 
   $u_{2i,2j-1}^p = (P_{k-1}^k u)_{2i,2j-1}$ 
   $\hat{f}_{2i,2j-1}^k = u_{2i,2j-1}^p + \hat{d}_{i,j}^k$  (3)
   $\hat{f}_{2i,2j}^k = 2\hat{f}_{2i,j}^{k-\frac{1}{2}} - \hat{f}_{2i,2j-1}^k$ 
end
end
end

```

Proposición B.1. *Dados un conjunto de valores discretos \bar{f}^L , con la transformada inversa modificada en 2D en el contexto de medias en celda (Algoritmo B.1) obtenemos una representación multirresolutiva*

$$M\bar{f}^L = \{\hat{f}^0, \hat{d}^1(1), \hat{d}^1(2), \hat{d}^2(3), \dots, \hat{d}^L(1), \hat{d}^L(2), \hat{d}^L(3)\}$$

tal que si aplicamos la transformada inversa (Algoritmo B.2) obtenemos \hat{f}^L que cumple:

$$\|\bar{f}^L - \hat{f}^L\|_\infty \leq \|\bar{f}^0 - \hat{f}^0\|_\infty + \sum_{k=1}^L \|\tilde{d}^k(2) - \hat{d}^k(2)\|_\infty \quad (\text{B.1})$$

$$+ \sum_{k=1}^L \max(\|\tilde{d}^k(1) - \hat{d}^k(1)\|_\infty, \|\tilde{d}^k(3) - \hat{d}^k(3)\|_\infty)$$

$$\|\bar{f}^L - \hat{f}^L\|_1 \leq \|\bar{f}^0 - \hat{f}^0\|_1 \quad (\text{B.2})$$

$$+ \frac{1}{2} \sum_{k=1}^L \max(\|\tilde{d}^k(2) - \hat{d}^k(2)\|_1, \|\tilde{d}^k(1) - \hat{d}^k(1)\|_1)$$

$$+ \frac{1}{2} \sum_{k=1}^L \max(\|\tilde{d}^k(2) - \hat{d}^k(2)\|_1, \|\tilde{d}^k(3) - \hat{d}^k(3)\|_1)$$

$$\|\bar{f}^L - \hat{f}^L\|_2^2 = \|\bar{f}^0 - \hat{f}^0\|_2^2 + \sum_{k=1}^L \|\tilde{d}^k(2) - \hat{d}^k(2)\|_2^2 \quad (\text{B.3})$$

$$+ \frac{1}{2} \sum_{k=1}^L \|\tilde{d}^k(1) - \hat{d}^k(1)\|_2^2 + \frac{1}{2} \sum_{k=1}^L \|\tilde{d}^k(3) - \hat{d}^k(3)\|_2^2$$

Demostración. Como consecuencia del algoritmo de codificación (Algoritmo B.1) tenemos:

$$\left\{ \begin{array}{l} \bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k = \bar{f}_{2i-1,j}^{k-\frac{1}{2}} - \hat{f}_{2i-1,j}^{k-\frac{1}{2}} - (\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)) \\ \bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k = \bar{f}_{2i-1,j}^{k-\frac{1}{2}} - \hat{f}_{2i-1,j}^{k-\frac{1}{2}} + (\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)) \\ \bar{f}_{2i,2j}^k - \hat{f}_{2i,2j}^k = \bar{f}_{2i,j}^{k-\frac{1}{2}} - \hat{f}_{2i,j}^{k-\frac{1}{2}} - (\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)) \\ \bar{f}_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k = \bar{f}_{2i,j}^{k-\frac{1}{2}} - \hat{f}_{2i,j}^{k-\frac{1}{2}} + (\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)) \\ \bar{f}_{2i-1,j}^{k-\frac{1}{2}} - \hat{f}_{2i-1,j}^{k-\frac{1}{2}} = \bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1} + (\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)) \\ \bar{f}_{2i,j}^{k-\frac{1}{2}} - \hat{f}_{2i,j}^{k-\frac{1}{2}} = \bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1} - (\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)) \end{array} \right. \quad (\text{B.4})$$

y

$$\left\{ \begin{array}{l} \bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k = \bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1} + (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2)) - (\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)) \\ \bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k = \bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1} + (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2)) + (\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)) \\ \bar{f}_{2i,2j}^k - \hat{f}_{2i,2j}^k = \bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1} - (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2)) - (\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)) \\ \bar{f}_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k = \bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1} - (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2)) + (\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)) \end{array} \right. \quad (\text{B.5})$$

Como consecuencia de la fórmula

$$\text{máx}(|a|, |b|) = \frac{1}{2}(|a+b| + |a-b|) \quad (\text{B.6})$$

tenemos

$$\begin{aligned} \text{máx}(|\bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k|, |\bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k|) &= |\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)| \\ &\quad + |(\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}) + (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2))| \end{aligned}$$

y

$$\begin{aligned} \text{máx}(|\bar{f}_{2i,2j}^k - \hat{f}_{2i,2j}^k|, |\bar{f}_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k|) &= |\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)| \\ &\quad + |(\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}) - (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2))|. \end{aligned}$$

De aquí

$$\begin{aligned}
& \text{máx} \left(|\bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k|, |\bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k|, |\bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k|, \right. \\
& \quad \left. |\bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k| \right) \\
& \leq \text{máx} \left(|\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)|, |\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)| \right) \\
& \quad + \text{máx} \left(|(\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}) + (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2))|, |(\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}) - (\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2))| \right) \\
& = \text{máx} \left(|(\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1))|, |(\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3))| \right) + |\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}| + |\tilde{d}_{i,j}^k(2) - \bar{d}_{i,j}^k(2)|
\end{aligned}$$

con lo que

$$\begin{aligned}
\|\hat{f}^k - \bar{f}^k\|_\infty & \leq \|\hat{f}^{k-1} - \bar{f}^{k-1}\|_\infty + \|\tilde{d}^k(2) - \hat{d}^k(2)\|_\infty \\
& \quad + \text{máx}(\|\tilde{d}^k(1) - \hat{d}^k(1)\|_\infty, \|\tilde{d}^k(3) - \hat{d}^k(3)\|_\infty)
\end{aligned}$$

y deducimos (B.1).

Además, de (B.5) y (B.6) obtenemos

$$\begin{aligned}
|\bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k| + |\bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k| & = 2|\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}| \\
& \quad + 2 \text{máx}(|\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)|, |\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)|) \\
|\bar{f}_{2i,2j}^k - \hat{f}_{2i,2j}^k| + |\bar{f}_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k| & = 2|\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}| \\
& \quad + 2 \text{máx}(|\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)|, |\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)|)
\end{aligned}$$

y

$$\begin{aligned}
& |\bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k| + |\bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k| + |\bar{f}_{2i,2j}^k - \hat{f}_{2i,2j}^k| + |\bar{f}_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k| \\
& \leq 4|\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1}| + 2 \text{máx}(|\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)|, |\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1)|) \\
& \quad + 2 \text{máx}(|\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2)|, |\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3)|).
\end{aligned}$$

De la fórmula anterior y de la definición de la norma $\|\cdot\|_1$ obtenemos

$$\begin{aligned}
\|\bar{f}^k - \hat{f}^k\|_1 & \leq \|\bar{f}^{k-1} - \hat{f}^{k-1}\|_1 + \frac{1}{2} \text{máx}(\|\tilde{d}^k(2) - \hat{d}^k(2)\|_1, \|\tilde{d}^k(1) - \hat{d}^k(1)\|_1) \\
& \quad + \frac{1}{2} \text{máx}(\|\tilde{d}^k(2) - \hat{d}^k(2)\|_1, \|\tilde{d}^k(3) - \hat{d}^k(3)\|_1)
\end{aligned}$$

con lo que hemos probado (B.2).

Para probar (B.3), obtenemos de (B.4)

$$\begin{aligned}
 (\bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k)^2 + (\bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k)^2 &= 2(\bar{f}_{2i-1,j}^{k-\frac{1}{2}} - \hat{f}_{2i-1,j}^{k-\frac{1}{2}})^2 + 2(\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1))^2 \\
 (\bar{f}_{2i,2j}^k - \hat{f}_{2i,2j}^k)^2 + (\bar{f}_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k)^2 &= 2(\bar{f}_{2i,j}^{k-\frac{1}{2}} - \hat{f}_{2i,j}^{k-\frac{1}{2}})^2 + 2(\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3))^2 \\
 (\bar{f}_{2i-1,j}^{k-\frac{1}{2}} - \hat{f}_{2i-1,j}^{k-\frac{1}{2}})^2 + (\bar{f}_{2i,j}^{k-\frac{1}{2}} - \hat{f}_{2i,j}^{k-\frac{1}{2}})^2 &= 2(\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1})^2 + 2(\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2))^2
 \end{aligned}$$

y

$$\begin{aligned}
 &(\bar{f}_{2i-1,2j-1}^k - \hat{f}_{2i-1,2j-1}^k)^2 + (\bar{f}_{2i-1,2j}^k - \hat{f}_{2i-1,2j}^k)^2 + (\bar{f}_{2i,2j}^k - \hat{f}_{2i,2j}^k)^2 + (\bar{f}_{2i,2j-1}^k - \hat{f}_{2i,2j-1}^k)^2 \\
 &= 4(\bar{f}_{i,j}^{k-1} - \hat{f}_{i,j}^{k-1})^2 + 4(\tilde{d}_{i,j}^k(2) - \hat{d}_{i,j}^k(2))^2 + 2(\tilde{d}_{i,j}^k(1) - \hat{d}_{i,j}^k(1))^2 + 2(\tilde{d}_{i,j}^k(3) - \hat{d}_{i,j}^k(3))^2
 \end{aligned}$$

entonces

$$\begin{aligned}
 \|\bar{f}^k - \hat{f}^k\|_2^2 &= \|\bar{f}^{k-1} - \hat{f}^{k-1}\|_2^2 + \|\tilde{d}^k(2) - \hat{d}^k(2)\|_2^2 \\
 &\quad + \frac{1}{2}\|\tilde{d}^k(1) - \hat{d}^k(1)\|_2^2 + \frac{1}{2}\|\tilde{d}^k(3) - \hat{d}^k(3)\|_2^2
 \end{aligned}$$

con lo que la proposición ha sido probada. \square