

TESIS DOCTORAL

Transformaciones multiescala
no lineales

José Jaime Noguera Noguera

Director: Francesc Aràndiga Llaudes

Universitat de València
Valencia, 2013.

TRANSFORMACIONES MULTIESCALA NO LINEALES

Memoria presentada por José Jaime Noguera Noguera,
Licenciado 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 Doctor en Matemáticas.

Valencia, 14 de abril de 2013

Francesc Aràndiga Llaudes
Director de la Memoria

José Jaime Noguera Noguera
Aspirante al Grado de Doctor

Programa de doctorado
Código: 655 255F. Título: Matemática Aplicada

DEPARTAMENTO DE MATEMÁTICA APLICADA
FACULTAD DE MATEMÁTICAS
UNIVERSITAT DE VALÈNCIA

Agradecimientos

En primer lugar, deseo expresar mi agradecimiento a todos los profesores de cualquier nivel educativo que en mayor o menor medida han contribuido a mi formación, en especial a los del departamento de Matemática Aplicada de la Universidad de Valencia.

A Rosa Donat, catedrática de dicho departamento, por animarme en los últimos años de la licenciatura a continuar con los estudios de doctorado. Además fue directora de mi trabajo de investigación y algunas de las ideas que aparecen en esta tesis fueron propuestas por ella.

A mi director de tesis, Francesc Aràndiga, por brindarme la oportunidad de trabajar con él. Sin sus propuestas, revisiones e indicaciones a lo largo de estos años, esta tesis no hubiese sido posible. En ningún momento me ha faltado su ayuda y comprensión, con un trato excepcional. Por todo ello, mi profundo respeto y admiración.

También quiero agradecer a todas aquellas personas que de forma prácticamente anónima y desinteresada publican contenido gratuito en Internet, ya que sus materiales me han resultado provechosos en ciertos aspectos como el uso de MATLAB® y L^AT_EX.

Finalmente, y cómo no, a mis padres, mi hermana y a Zoraida.

Valencia, 2013

José Jaime

A mis padres,

Índice general

| | |
|--|----------|
| 1. Preliminares | 5 |
| 1.1. Análisis multirresolutivo <i>à la Harten</i> | 5 |
| 1.1.1. Descomposiciones Multiescala mediante Decimación y Predicción | 6 |
| 1.2. Esquemas de multirresolución para valores puntuales | 13 |
| 1.2.1. Técnicas de interpolación lineales y no lineales | 15 |
| 1.2.2. Técnica <i>ENO</i> | 17 |
| 1.2.3. Filtros para esquemas en valores puntuales . . | 20 |
| 1.2.4. Técnica <i>WENO</i> | 23 |
| 1.2.5. Técnica <i>PPH</i> | 29 |
| 1.2.6. Técnica de resolución subcelda (<i>SR</i>) | 30 |
| 1.2.7. Ejemplos | 38 |
| 1.3. Esquemas de multirresolución para medias en celda . | 39 |
| 1.3.1. Técnicas de interpolación lineales y no lineales para medias en celda | 42 |
| 1.3.2. Filtros para esquemas en medias en celda . . . | 46 |
| 1.3.3. <i>WENO</i> para medias en celda | 49 |
| 1.3.4. <i>PPH</i> para medias en celda | 52 |
| 1.3.5. <i>ENO-SR</i> para medias en celda | 53 |
| 1.3.6. Ejemplos | 59 |
| 1.4. Transformada <i>wavelet</i> | 60 |
| 1.4.1. Transformada <i>wavelet</i> continua | 62 |
| 1.4.2. Transformada <i>wavelet</i> discreta | 65 |
| 1.5. Conclusiones | 71 |

| | |
|--|------------|
| 2. Nuevas reconstrucciones | 75 |
| 2.1. Introducción | 75 |
| 2.2. Reconstrucción <i>interpolación aproximación</i> | 76 |
| 2.2.1. <i>Interpolación aproximación</i> para valores puntuales (\overline{IAP}) | 76 |
| 2.2.2. <i>Interpolación aproximación</i> para medias en celda (\overline{IAC}) | 92 |
| 2.2.3. Funciones límite | 107 |
| 2.2.4. Diseño de curvas | 107 |
| 2.3. Reconstrucción utilizando <i>aproximaciones</i> | 110 |
| 2.3.1. <i>Aproximaciones</i> para valores puntuales (\overline{AP}) | 111 |
| 2.3.2. <i>Aproximaciones</i> para medias en celda (\overline{AC}) | 119 |
| 2.3.3. Funciones límite | 129 |
| 2.3.4. Diseño de curvas | 132 |
| 2.4. Conclusiones | 132 |
| 3. Detección de discontinuidades | 137 |
| 3.1. Introducción | 137 |
| 3.2. Métodos de detección de discontinuidades para funciones sin ruido | 138 |
| 3.2.1. Interpolación adaptativa: técnica <i>ENO</i> | 138 |
| 3.2.2. Magnitud de coeficientes <i>wavelet</i> | 147 |
| 3.3. Métodos de detección de discontinuidades para funciones con ruido | 152 |
| 3.3.1. Combinación <i>ENO</i> y mínimos cuadrados | 153 |
| 3.3.2. Magnitud de coeficientes <i>wavelet</i> para funciones con ruido | 156 |
| 3.3.3. Reconstrucción por <i>aproximaciones de stencil</i> creciente | 159 |
| 3.3.4. Comparativa entre los métodos de detección para funciones con ruido. | 162 |
| 3.4. Conclusiones | 163 |
| 4. ENO-wavelets | 165 |
| 4.1. Función de extrapolación directa | 166 |
| 4.2. Extrapolación en el nivel grosero | 167 |
| 4.3. Error de aproximación para <i>ENO-wavelets</i> | 169 |
| 4.4. <i>ENO-DB4</i> | 171 |

| | |
|---|------------|
| 4.4.1. Reconstrucción | 174 |
| 4.5. <i>ENO-DB6</i> | 175 |
| 4.5.1. Reconstrucción | 180 |
| 4.6. Algoritmos utilizando la transformada inversa (<i>TI</i>) . . | 182 |
| 4.6.1. <i>ENO-DB4-TI</i> | 182 |
| 4.6.2. <i>ENO-DB6-TI</i> | 184 |
| 4.6.3. Algoritmo general | 185 |
| 4.7. Equivalencia de los métodos <i>ENO-wavelet</i> y <i>ENO-wavelet-TI</i> | 189 |
| 4.8. Invertibilidad de los métodos <i>TI</i> | 194 |
| 4.9. Ejemplos | 196 |
| 4.10. Conclusiones | 199 |
| 5. Aplicaciones a la eliminación de ruido | 201 |
| 5.1. Introducción | 201 |
| 5.2. Trabajando en una escala | 203 |
| 5.2.1. Aproximación por Mínimos Cuadrados (<i>AMC</i>) entre discontinuidades | 203 |
| 5.2.2. Reconstrucción \overline{AP} para cada punto | 207 |
| 5.2.3. Iteración por sustitución de valores medios . . | 212 |
| 5.3. Trabajando con varias escalas | 213 |
| 5.3.1. Eliminación de ruido mediante descomposicio- nes multiescala <i>à la Harten</i> | 215 |
| 5.3.2. <i>Multiscale Shrinkage</i> | 220 |
| 5.3.3. Multirresolución <i>à la Harten</i> con <i>Shrinkage</i> . . | 227 |
| 5.3.4. <i>ENO-wavelets</i> | 232 |
| 5.4. Comparativa | 234 |
| 5.5. Conclusiones | 240 |
| 6. Conclusiones generales y perspectivas | 243 |

Introducción general

El tratamiento de señales digitales ha experimentado un creciente interés dentro del área de Matemática Aplicada debido a sus múltiples aplicaciones como, entre otras, la compresión de imágenes, eliminación de ruido, reconocimiento de patrones, recuperación de fotografías, en campos tan diversos como la Informática, Física, Medicina o Ingeniería.

En los años 80 Mallat, [55], formula el análisis de multirresolución, donde la transformada *wavelet* discreta adquiere su máxima potencialidad. Autores como Meyer, [58], y Daubechies, [31], entre otros, popularizaron estas ideas, aunque cabe destacar que la primera referencia al concepto de *wavelet* se remonta a 1909 en la tesis de Haar, [42]. La transformada *wavelet* resulta especialmente eficiente para extraer información de señales no periódicas, ya que se caracteriza por estar localizada en tiempo y frecuencia, a diferencia de la clásica transformada de Fourier, que está localizada únicamente en frecuencia y es adecuada para funciones 2π periódicas. Otra de las ventajas frente a otros métodos es el de disponer de una amplia familia de *wavelets*, permitiendo tratar señales de diversa índole. Básicamente la transformada *wavelet* discreta permite dividir la señal en dos tipos de subseñales, las tendencias o coeficientes de bajas frecuencias y las fluctuaciones o coeficientes de altas frecuencias. Las tendencias pueden entenderse como una copia de la señal a menor resolución, mientras que las fluctuaciones representan los detalles necesarios para recuperar la función original. En un siguiente nivel se puede aplicar la misma descomposición a las tendencias obtenidas y así sucesivamente, dando lugar a un algoritmo piramidal con excelentes

capacidades de compresión, ya que normalmente (por ejemplo si la función es suave) los coeficientes de altas frecuencia son de pequeña magnitud y podemos eliminarlos (aproximación lineal) sin perder excesiva información.

En 1993 A. Harten, [44], generaliza un cierto tipo de *wavelets* biortogonales tomando ideas de tres campos diferentes: la teoría de funciones *wavelet*, la solución numérica de ecuaciones en derivadas parciales y los esquemas de subdivisión. En dicha formulación surgen nuevos operadores que permiten introducir no linealidad en el proceso de recuperación de datos de la escala más fina (reconstrucciones). Surgen así una serie de técnicas que mejoran las aproximaciones en presencia de discontinuidades como *ENO* (*Essentially Non-Oscillatory*, [47]), *SR* (*Subcelde Resolution*, [43]) y más recientemente *WENO* (*Weighted ENO*, [52]) y *PPH* (*Piecewise Polynomial Harmonic*, [6]). Generalmente, las funciones de reconstrucción en esquemas de multirresolución *à la Harten* han sido las interpolatorias. En esta tesis nos planteamos el uso de otro tipo de reconstrucciones que combinen interpolación y aproximación. Surgen entonces varias preguntas: ¿cumplen los requisitos de la multirresolución *à la Harten*?, ¿podemos aplicar las técnicas no lineales *ENO*, *SR*, *WENO* y *PPH*?. En este trabajo responderemos a dichas cuestiones.

En el año 2000 Chan y Zhou presentan en [26] la transformada *ENO-wavelet* donde se modifica el cálculo de los coeficientes tomando como base la idea *ENO*, consistente en seleccionar para cada intervalo un *stencil* que no cruce discontinuidades. Con esto se evita la creación de valores elevados en los coeficientes de altas frecuencias, manteniéndose el orden de aproximación para funciones discontinuas, bajo ciertos requisitos como una correcta localización de las discontinuidades y una suficiente separación de las mismas. Además se consigue preservar el tamaño de las secuencias de entrada y salida dando lugar al mismo tipo de algoritmo piramidal, con un coste computacional extra despreciable. En este trabajo hemos introducido cambios en la manera de calcular dichos coeficientes lo cual proporcionará diversas ventajas, como veremos en el desarrollo posterior.

La localización de discontinuidades en señales digitales es otro campo de estudio con múltiples aplicaciones, siendo especialmen-

te importante en esta tesis como paso previo a la aplicación de diversos algoritmos. Es por ello que dedicaremos un capítulo al estudio y diseño de métodos de detección de discontinuidades tanto para funciones con ruido como sin él.

La introducción de reconstrucciones no exclusivamente interpolatorias nos hace plantearnos la hipótesis de su posible aplicación a la eliminación de ruido. Este campo ha sido tratado con éxito haciendo uso de la transformada *wavelet* mediante la utilización de umbrales que conservan ciertos coeficientes de altas frecuencias, conocido como *Wavelet Shrinkage Denoising* (ver p.ej. [63, 35, 23, 41]). Con estas premisas responderemos a preguntas como: ¿se elimina ruido eficientemente mediante la multirresolución *à la Harten* con reconstrucciones no interpolatorias?, ¿se obtienen errores comparables a los métodos *Wavelet Shrinkage Denoising*? Además diseñaremos nuevos esquemas que trabajen en un sólo nivel de resolución, combinaremos ideas de diferentes métodos, los compararemos e intentaremos determinar con cuál se obtienen mejores resultados.

Esquema de la Tesis

La organización de este trabajo es como sigue:

Capítulo 1: Se introducen las herramientas matemáticas que se utilizarán en capítulos posteriores. En primer lugar revisamos la multirresolución *à la Harten* para valores puntuales y medias en celda, y la aplicación de diferentes técnicas no lineales (*ENO*, *SR*, *WENO* y *PPH*), haciendo especial hincapié en el cálculo de filtros. En segundo lugar se describe la transformada *wavelet*, tanto continua como discreta y la detallamos para *DB4*, ya que será ésta la principal base utilizada.

Capítulo 2: Desarrollamos la nueva reconstrucción por *interpolación aproximación* para valores puntuales y medias en celda en el contexto de multirresolución *à la Harten*. Estudiamos sus órdenes, la introducción de técnicas no lineales (*ENO*, *SR*, *WENO* y *PPH*), funciones límite, generación de curvas, calculamos sus filtros y la aplicamos a ejemplos concretos.

Posteriormente se realiza un estudio similar para la reconstrucción por *aproximaciones* (basada en mínimos cuadrados).

Capítulo 3: Tratamos el tema de la localización de discontinuidades para funciones discontinuas compuestas por trozos suaves, tanto para señales con ruido añadido como sin él. Estudiamos métodos existentes en la literatura relacionados con nuestras herramientas matemáticas, proponemos nuevos esquemas de localización y realizamos comparativas entre ellos.

Capítulo 4: Describimos la transformada *ENO-wavelet* introducida por Chan y Zhou en [26]. Presentamos una nueva manera de calcular los coeficientes de altas y bajas frecuencias en caso de discontinuidad, dando lugar a la transformada *ENO-wavelet-TI*. Analizamos su posible equivalencia con la transformada *ENO-wavelet*, su invertibilidad, sus órdenes, proponemos un algoritmo para *ENO-DB2 p -TI* con p natural y estudiamos ejemplos concretos.

Capítulo 5: Aplicamos los esquemas vistos anteriormente a la eliminación de ruido de funciones discontinuas compuestas por trozos suaves con ruido añadido. En primer lugar trabajamos en una única escala, donde diseñamos un esquema basado en la reconstrucción por aproximaciones y un método iterativo con bajo coste computacional. En segundo lugar nos basamos en métodos multiescala, adaptando las distintas reconstrucciones al problema propuesto. Se realiza una comparativa entre métodos y se ofrecen indicaciones para elegir un método en función de los datos de entrada y de nuestros objetivos.



<http://www.uv.es/animis/noguera/> Permite la descarga de esta tesis en formato PDF y se ofrece material adicional como programas y GUI's en MATLAB® para el cálculo de filtros de las diferentes reconstrucciones, detección de discontinuidades y eliminación de ruido.

1

Preliminares

En este capítulo expondremos las herramientas matemáticas que utilizaremos a lo largo de este trabajo. En primer lugar desarrollaremos el análisis multirresolutivo *à la Harten* y en segundo lugar describiremos la teoría asociada a la transformada *wavelet*.

1.1

Análisis multirresolutivo *à la Harten*

1.1.1

Descomposiciones Multiescala mediante Decimación y Predicción

El concepto de análisis de multirresolución (MRA) fue desarrollado en los años 80 por Meyer [58] y Mallat [55], y como su nombre sugiere, permite estudiar una función complicada en diferentes resoluciones, dando lugar a funciones más simples. En este trabajo nos basaremos en la noción de multirresolución introducida por Ami Harten en [44] y posteriormente generalizada en [45] y [46].

A continuación describiremos los aspectos básicos de los esquemas de multirresolución à la Harten. El lector interesado encontrará mayor nivel de detalle y ejemplos de aplicación en los artículos citados, así como en [16], [15] y [1].

DEFINICIÓN 1.1. *Un conjunto de multirresolución es una secuencia de espacios lineales, $\{V^k\}$, con bases numerables, que denotamos $\{\eta_i^k\}$, junto a una secuencia de operadores lineales $\{D_k^{k-1}\}$ de V^k a V^{k-1} denominados operadores decimación, es decir,*

$$D_k^{k-1} : V^k \rightarrow V^{k-1}, \quad V^{k-1} = D_k^{k-1}(V^k).$$

Los espacios V^k representan los diferentes niveles de resolución que podemos estudiar de un conjunto de datos discretos. En este trabajo se utiliza que a mayor k mayor resolución, lo cual implica mayor detalle.

DEFINICIÓN 1.2. *Decimos que $\{P_{k-1}^k\}_k$ es una secuencia de operadores predicción para el conjunto de multirresolución $\{\{V^k\}, \{D_k^{k-1}\}\}$, si cada operador P_{k-1}^k es inverso por la derecha de D_k^{k-1} en V^{k-1} , es decir,*

$$P_{k-1}^k : V^{k-1} \rightarrow V^k, \quad D_k^{k-1} P_{k-1}^k = I_{V^{k-1}}.$$

NOTA 1.1. *La definición 1.2 es válida incluso para P_{k-1}^k no lineal.*

Según la definición 1.2 los operadores predicción deben satisfacer la siguiente relación de consistencia:

$$D_k^{k-1}P_{k-1}^k = I_{V_{k-1}}, \quad (1.1)$$

que implica que los valores obtenidos en la predicción en el nivel de resolución k contienen la misma información discreta que los valores originales restringidos al nivel $k-1$.

Por otro lado el operador $P_{k-1}^k D_k^{k-1} : V^k \rightarrow V^k$ produce aproximaciones a cada vector v^k en V^k a partir de la información del nivel $k-1$. Luego tiene sentido estudiar el error que se produce en dicha aproximación.

DEFINICIÓN 1.3. *Denominamos error de predicción, e^k , asociado al conjunto de multiresolución $\{\{V^k\}, \{D_k^{k-1}\}\}$ a la diferencia entre v^k y el valor que proporciona el esquema de predicción al tratar de reconstruir v^k a partir de la información contenida en el nivel de resolución inmediato inferior, es decir,*

$$e^k := (I_{V^k} - P_{k-1}^k D_k^{k-1}) v^k =: Q_k v^k. \quad (1.2)$$

PROPOSICIÓN 1.1. *El error de predicción e^k es un vector en V^k que pertenece al núcleo del operador decimación, $\mathcal{N}(D_k^{k-1})$.*

Demostración. Dado que

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

se sigue que $e^k \in \mathcal{N}(D_k^{k-1}) = \{v/v \in V^k, D_k^{k-1}v = 0\}$.

□

Si expresamos el error de predicción en una base de V^k , existe información redundante que podemos eliminar eligiendo una determinada base de funciones de $\mathcal{N}(D_k^{k-1})$, $\mathcal{N}(D_k^{k-1}) = \langle \mu_j^k \rangle$ y expresando e^k en dicha base.

Sea $G_k : \mathcal{N}(D_k^{k-1}) \rightarrow \mathcal{G}^k$ el operador que asigna a cada $e^k \in \mathcal{N}(D_k^{k-1})$ la secuencia d^k de sus coordenadas en la base $\{\mu_j^k\}$, y sea E_k la inyección canónica $\mathcal{N}(D_k^{k-1}) \hookrightarrow V^k$, si

$$e^k = \sum d_j^k \mu_j^k \quad \Rightarrow \quad d^k = G_k e^k \quad \text{y} \quad E_k d^k = \sum d_j^k \mu_j^k \in \mathcal{N}(D_k^{k-1}).$$

Obviamente $G_k E_k = I_{\mathcal{G}^k}$, $E_k G_k = I_{\mathcal{N}(D_k^{k-1})}$ y hay una correspondencia biunívoca entre v^k y $\{d^k, v^{k-1}\}$:

$$\text{Paso de } v^k \text{ a } \{d^k, v^{k-1}\} \rightarrow \begin{cases} v^{k-1} = D_k^{k-1} v^k, \\ d^k = G_k Q_k v^k. \end{cases} \quad (1.3)$$

$$\text{Paso de } \{d^k, v^{k-1}\} \text{ a } v^k \rightarrow P_{k-1}^k v^{k-1} + E_k d^k = P_{k-1}^k D_k^{k-1} v^k + E_k G_k Q_k v^k = v^k. \quad (1.4)$$

NOTA 1.2. Se puede interpretar d^k como la información no redundante presente en v^k y no recuperable de v^{k-1} mediante el esquema de predicción definido por P_{k-1}^k . Por esto, las componentes de d^k suelen denominarse coeficientes de la escala k de la representación por multirresolución.

Si los espacios V^k son de dimensión finita, podemos decir que $\dim V^k = N_k$, con lo cual el vector v^k viene representado por sus N_k coordenadas en la base $\{\mu_i^k\}$ y se cumplen las siguientes relaciones:

$$\text{card}(v^{k-1}) + \text{card}(d^k) = \dim(v^k),$$

$$\dim V^k - \dim V^{k-1} = \dim(\mathcal{N}(D_k^{k-1})).$$

Repetiendo el proceso descrito en (1.3) y (1.4) para cada nivel de resolución, obtenemos un esquema de multirresolución formado por $\{\{V^k\}_{k=0}^L, \{D_k^{k-1}\}_{k=1}^L\}$ y una sucesión de operadores predicción, $\{P_{k-1}^k\}_{k=1}^L$, que pueden ser o no lineales.

La descripción algorítmica de esta transformación uno a uno y su inversa es como sigue:

Codificación: $v^L \longrightarrow Mv^L = (v^0, d^1, \dots, d^L)$

$$\left\{ \begin{array}{l} \text{Para } k = L, \dots, 1 \\ v^{k-1} = D_k^{k-1} v^k, \\ d^k = G_k(v^k - P_{k-1}^k v^{k-1}). \end{array} \right. \quad (1.5)$$

Decodificación: $Mv^L \longrightarrow M^{-1}Mv^L$

$$\left\{ \begin{array}{l} \text{Para } k = 1, \dots, L \\ v^k = P_{k-1}^k v^{k-1} + E_k d^k. \end{array} \right. \quad (1.6)$$

Nos referimos a Mv^L como la representación multirresolutiva de v^L , y a los algoritmos (1.5) y (1.6) como las transformaciones de multirresolución directa e inversa, respectivamente.

Los operadores D_k^{k-1} y P_{k-1}^k pueden ser expresados en una estructura piramidal, del modo utilizado en proceso de señales:

$$\begin{array}{ccccccc} v^k & \longrightarrow & v^{k-1} & \longrightarrow & v^{k-2} & \longrightarrow & \dots \\ & \searrow & & \searrow & & \searrow & \\ & & e^k & & e^{k-1} & & \dots \end{array}$$

O bien, si expresamos los errores en una base de $\mathcal{N}(D_k^{k-1})$:

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

Con esto, podemos considerar que las descomposiciones se realizan aplicando dos filtros, el filtro de paso bajo, decimación, que es siempre lineal, y el filtro de paso alto, predicción, que no necesariamente debe ser lineal.

Hasta este momento hemos trabajado sobre aplicaciones definidas sobre espacios lineales que poseen bases numerables. Los datos discretos sobre los que necesitamos trabajar van a provenir de ciertas funciones. Para generar datos discretos a partir de éstas, aplicamos un determinado *proceso de discretización* que consiste en un operador que aplicado a cada función proporciona unos

datos discretos en V^k . Dicho proceso de discretización define de manera natural los espacios V^k y da significado al concepto de nivel de resolución asociado a dichos espacios. En lo que sigue definiremos estos conceptos y veremos cómo una sucesión de operadores decimación puede construirse a partir de una *sucesión anidada de operadores discretización*.

DEFINICIÓN 1.4. Sea \mathcal{D} un operador lineal en un espacio lineal \mathcal{F} , con rango V . Si V posee una base numerable, $\{\mu_i\}$, decimos que \mathcal{D} es un operador discretización de \mathcal{F} y, para cada $f \in \mathcal{F}$, nos referimos a $v = \mathcal{D}f$ como la discretización de f al nivel de resolución especificado por V .

$$\begin{aligned} \mathcal{D} : \mathcal{F} &\longrightarrow V = \mathcal{D}(\mathcal{F}) = \langle \mu_i \rangle \\ f &\longrightarrow \mathcal{D}f = v \end{aligned}$$

DEFINICIÓN 1.5. Sea $\{\mathcal{D}_k\}$ una sucesión de operadores discretización en \mathcal{F} ,

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

Decimos que la sucesión $\{\mathcal{D}_k\}$ es anidada si para todo k y para toda $f \in \mathcal{F}$,

$$\mathcal{D}_k f = 0 \Rightarrow \mathcal{D}_{k-1} f = 0. \quad (1.7)$$

La propiedad de sucesión anidada implica que la información discreta en un nivel de resolución está incluida en la información discreta de todos los niveles de resolución más finos, o dicho de otro modo, que cada nivel de resolución contiene no más información que los niveles de resolución mayores.

Una sucesión anidada de discretizaciones define una sucesión de operadores decimación, y así, un esquema de multirresolución. Este resultado se sigue del siguiente lema:

LEMA 1.1. Si \mathcal{D}_k es una sucesión anidada de operadores discretización, entonces la aplicación de V^k a V^{k-1} definida como sigue:

Para cada $v \in V^k$ tomar una $f \in \mathcal{F}$ tal que $v = \mathcal{D}_k f$ y asignar

$$u := \mathcal{D}_{k-1} f,$$

está bien definida.

Demostración. Dado que $V^k = \mathcal{D}_k(\mathcal{F})$, para cualquier $v \in V^k$ existe al menos una $f \in \mathcal{F}$ tal que $v = \mathcal{D}_k f$. Queremos probar que la asignación $u = \mathcal{D}_{k-1} f$ es independiente de la elección de f . Para ello debemos ver que

$$\mathcal{D}_k f = \mathcal{D}_k g \Rightarrow \mathcal{D}_{k-1} f = \mathcal{D}_{k-1} g.$$

Esto es evidente debido a que los operadores discretización son anidados y lineales:

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

□

Cada operador decimación queda definido como sigue:

$$\begin{aligned} D_k^{k-1} : V^k &\longrightarrow V^{k-1} \\ v^k (:= \mathcal{D}_k f) &\longrightarrow D_k^{k-1} v^k (:= \mathcal{D}_{k-1} f) \end{aligned}$$

Gracias al lema 1.1, la definición es independiente de f , con lo cual:

$$D_k^{k-1} \mathcal{D}_k = \mathcal{D}_{k-1}, \tag{1.8}$$

siendo entonces un operador lineal. Además $D_k^{k-1} V^k = V^{k-1}$, ya que dado $u \in V^{k-1}$, tomando $f \in \mathcal{F}$ tal que $u = \mathcal{D}_{k-1} f$, entonces si $v = \mathcal{D}_k f \in V^k$, por (1.8),

$$D_k^{k-1} v = D_k^{k-1}(\mathcal{D}_k f) = \mathcal{D}_{k-1} f = u.$$

DEFINICIÓN 1.6. Dado $v \in V^k$, decimos que $f \in \mathcal{F}$ es una reconstrucción de v en \mathcal{F} si satisface que $v = \mathcal{D}_k f$.

Para concluir el diseño del esquema de multirresolución debemos definir adecuadamente los operadores predicción, P_{k-1}^k . Para ello necesitamos definir el operador reconstrucción.

DEFINICIÓN 1.7. Decimos que $\mathcal{R} : V \rightarrow \mathcal{F}$, con $V = \mathcal{D}(\mathcal{F})$, es un operador reconstrucción para \mathcal{D} en V si es un operador inverso por la derecha de \mathcal{D} , es decir $\mathcal{D}\mathcal{R} = I_V$.

NOTA 1.3. En la definición de \mathcal{R} no se exige linealidad.

Estamos ya en condiciones de definir un operador inverso por la derecha de D_k^{k-1} . Dada una sucesión de operadores de discretización \mathcal{D}_k y una sucesión de sus correspondientes operadores de reconstrucción \mathcal{R}_k , se define el operador predicción:

$$P_{k-1}^k := \mathcal{D}_k \mathcal{R}_{k-1} : V^{k-1} \rightarrow V^k. \quad (1.9)$$

Claramente el operador definido es inverso por la derecha de D_k^{k-1} :

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

Por tanto $\{P_{k-1}^k\}$ es una secuencia de operadores predicción para el conjunto de multirresolución $\{\{V^k\}, \{D_k^{k-1}\}\}$. La Figura 1.1 clarifica la definición de los operadores descritos en este apartado.

El encontrar un esquema de multirresolución adecuado para una función determinada puede ser formulado como un problema típico de teoría de aproximación:

Dado $\mathcal{D}_{k-1}f$, $f \in \mathcal{F}$, hallar una "buena aproximación" a $\mathcal{D}_k f$.

El proceso de reconstrucción juega un papel crucial en la multirresolución à la Harten. El elegir técnicas de reconstrucción adecuadas para las aplicaciones dadas, facilita el análisis e interpretación de las descomposiciones de multirresolución obtenidas mediante los correspondientes esquemas de predicción. Además

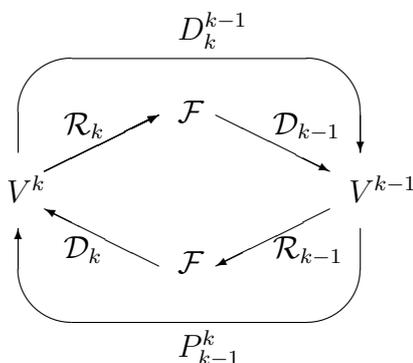


Figura 1.1: Definición de operadores en multirresolución à la Harten.

como hemos advertido, en las definiciones de operadores reconstrucción no se exige linealidad, con lo cual podemos utilizar reconstrucciones adaptativas no-lineales que mejorarán la calidad del esquema. Esto se desarrollará en apartados posteriores.

Como hemos visto, el proceso de discretización determina el esquema de multirresolución. En lo que sigue, veremos dos ejemplos de discretización que dan lugar a dos esquemas de multirresolución típicos¹: por *valores puntuales* y por *medias en celda*. Mediante estos esquemas aplicaremos la teoría vista a casos concretos y además serán de especial importancia en capítulos posteriores.

1.2

Esquemas de multirresolución para valores puntuales

Consideramos el espacio de funciones continuas² en el intervalo $[0, 1]$, $\mathcal{F} = \mathcal{C}[0, 1]$. Tomamos las secuencias:

$$X^L = \{x_i^k\}_{i=0}^{J_L}, \quad x_0^L = 0, \quad x_i^L = i \cdot h_L, \quad h_L = 1/J_L, \quad J_L = 2^L J_0. \quad (1.10)$$

¹Otra posible discretización son los *hat averages* [18, 19, 15], aunque en este trabajo no los utilizaremos.

²Podríamos partir simplemente de un espacio de funciones acotadas en un dominio acotado, ver [16].

14 1.2. Esquemas de multirresolución para valores puntuales

donde J_0 es cualquier entero. Definimos el operador discretización, $f^k = \mathcal{D}_k f$, de la siguiente manera:

$$\mathcal{D}_k : \mathcal{F} \rightarrow V^k, \quad f_i^k = (\mathcal{D}_k f)_i = f(x_i^k), \quad f^k = \{f_i^k\}_{i=0}^{J_k}, \quad (1.11)$$

donde $\mathcal{F} = \mathcal{C}[0, 1]$ y V^k es el espacio de sucesiones de dimensión $J_k + 1$. Nos referimos a (1.11) como la *discretización por valores puntuales*. Dado que los espacios V^k son de dimensión $J_k + 1$ finita, podemos tomar como base $\mu_i^k = \delta_i^k$, la base canónica de funciones en espacios de sucesiones.

Definimos las secuencias de menor resolución de la siguiente manera:

$$x_i^{k-1} = x_{2i}^k, \quad i = 0, \dots, J_{k-1} := J_k/2, \quad k = L-1, \dots, 0. \quad (1.12)$$

Con esto, se cumple que $X^{k-1} \subset X^k$ ya que en X^{k-1} hemos eliminado los puntos de X^k de índice impar. Así pues la secuencia de operadores discretización $\{\mathcal{D}_k\}$ es anidada y por tanto definirá un conjunto de multirresolución, según el Lema 1.1. La decimación vendrá dada por

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

NOTA 1.4. *El operador decimación se define directamente en términos de la sucesión $\{\mathcal{D}_k\}$ sin ninguna referencia explícita a un operador reconstrucción.*

La reconstrucción $\mathcal{R}_k : V^k \rightarrow \mathcal{C}[0, 1]$, debe ser tal que $\mathcal{D}_k \mathcal{R}_k f^k = f^k$. Por tanto $(\mathcal{R}_k f^k)(x)$ debe ser una función acotada que interpo-
le al conjunto $\{f_i^k\}$ en los nodos $\{x_i^k\}$, es decir, podemos utilizar cualquier técnica de interpolación, que denotamos: $\overline{\mathcal{I}\mathcal{P}}_k(x; f^k) := (\mathcal{R}_k f^k)(x)$. Con esto:

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

El error de predicción es $e_i^k = f_i^k - \overline{\mathcal{I}\mathcal{P}}_{k-1}(x_i^k; f^{k-1})$, $i = 0, \dots, J_k$, y $e_i^k = 0$, $\forall i$ tal que $x_i^k \in X^{k-1}$. El núcleo, $\mathcal{N}\{D_k^{k-1}\}$, está compuesto por las sucesiones tales que $f_i^k = 0$ si $x_i^k \in X^{k-1}$. Los coeficientes $d^k = \{d_j^k\}$ se definen como $d_j^k = e_{2j-1}^k$ para $j = 1, \dots, J_{k-1}$.

En definitiva, los esquemas (1.5) y (1.6) para este caso son:

Codificación, $f^L \rightarrow Mf^L$:

$$\left\{ \begin{array}{ll} \text{Para } k = L, \dots, 1 & \\ f_i^{k-1} = f_{2i}^k & i = 0, \dots, J_{k-1}, \\ d_i^k = f_{2i-1}^k - \overline{\mathcal{I}\mathcal{P}}_{k-1}(x_{2i-1}^k; f^{k-1}) & i = 1, \dots, J_{k-1}. \end{array} \right. \quad (1.13)$$

Decodificación, $M^{-1}Mf^L \rightarrow f^L$:

$$\left\{ \begin{array}{ll} \text{Para } k = 1, \dots, L & \\ f_{2i}^k = f_i^{k-1} & i = 0, \dots, J_{k-1}, \\ f_{2i-1}^k = \overline{\mathcal{I}\mathcal{P}}_{k-1}(x_{2i-1}^k; f^{k-1}) + d_i^k & i = 1, \dots, J_{k-1}. \end{array} \right. \quad (1.14)$$

1.2.1

Técnicas de interpolación lineales y no lineales

En la sección anterior hemos expuesto la teoría relativa a la multirresolución *à la Harten*, particularizándola para la discretización por valores puntuales. En los esquemas (1.13) y (1.14) aparecen los operadores de interpolación que aún no han sido desarrollados. Iniciaremos esta sección describiendo de forma breve el tipo de interpolación utilizada. Posteriormente expondremos las técnicas de interpolación *ENO* y *SR*, y las aplicaremos a los esquemas multirresolutivos a fin de calcular sus filtros.

Como es bien conocido, mediante la interpolación hallamos, a partir de los datos discretos de una función, un polinomio de un cierto grado que coincide con la función en los puntos de la discretización.

Dada $f(x)$ y un conjunto de $r + 1$ puntos, $\{x_0, x_1, \dots, x_r\}$, con $x_i \neq x_j$ si $i \neq j$, existe un único polinomio $q(x)$ de grado r tal que $q(x_i) = f(x_i)$ para $i = 0, 1, \dots, r$. Según la forma de Newton, este polinomio viene dado por la fórmula [5], [29]:

$$q(x) = f(x_0) + \sum_{j=1}^r \left(f[x_0, \dots, x_j] \prod_{i=0}^{j-1} (x - x_i) \right). \quad (1.15)$$

16 1.2. Esquemas de multirresolución para valores puntuales

El conjunto de puntos utilizado para definir $q(x)$ lo denominamos *stencil* de $q(x)$ y lo denotamos por \mathcal{S} . En cuanto al error que se comete al interpolar sabemos que:

$$f(x) = q(x) + f[x_0, \dots, x_r, x] \prod_{i=0}^r (x - x_i) = q(x) + f[\mathcal{S}, x] \prod_{x_i \in \mathcal{S}} (x - x_i), \quad (1.16)$$

donde $\mathcal{S} = \{x_0, \dots, x_r\}$ y $f[\mathcal{S}, x]$ es la $(r + 1)$ diferencia dividida de $f(x)$ en los puntos del *stencil* añadiendo x . Si $f(x)$ es diferenciable al menos $r + 1$ veces, entonces:

$$f[x_0, \dots, x_r, x] = \frac{f^{(r+1)}(\xi_x)}{(r + 1)!}, \quad (1.17)$$

donde ξ_x es un punto interior al conjunto convexo $\{x_0, \dots, x_r, x\}$.

Una buena estrategia para minimizar el error de interpolación, $|f(x) - q(x)|$, en todo el dominio de la función es utilizar interpolación segmentaria [5]. Para simplificar la exposición, tomamos el intervalo $[0, 1]$ y una malla de puntos equiespaciados $X = \{x_i\}_{i=0}^J$, $x_i = ih$, $Jh = 1$. Consideremos un conjunto discreto de datos $\{f_i\}_{i=1}^J$, con $f_i = f(x_i)$. La interpolación polinómica a trozos consiste en construir para cada subintervalo $I_i = [x_{i-1}, x_i]$ un polinomio $q_i(x)$, para $i = 1, \dots, J$. El interpolante global viene definido de la siguiente manera:

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

donde $q_i(x_{i-1}) = f_{i-1}$ y $q_i(x_i) = f_i$.

Si tomamos *stencils* compuestos por $r + 1$ puntos consecutivos (polinomios de grado r) y $f(x)$ es suficientemente suave, la fórmula del error (1.16), junto a (1.17) nos proporciona

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

es decir, de orden $r + 1$. Además tomando derivadas en (1.16) y suponiendo que f es suficientemente suave es fácil ver que si $\overline{\mathcal{I}\mathcal{P}}(x)$ es una técnica de interpolación polinómica a trozos de orden $r + 1$:

$$\frac{d^m}{dx^m} \overline{\mathcal{I}\mathcal{P}}(x \pm 0; Df) = \frac{d^m}{dx^m} f(x) + O(h^{r+1-m}), \quad 0 \leq m \leq r.$$

El problema fundamental de esta técnica proviene de la falta de suavidad de la función inicial. Si $f(x)$ presenta un salto en I_i ,

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

donde $[f]$ representa el salto de $f(x)$ en I_j . Esto provoca que todas las diferencias divididas basadas en *stencils* de $s + 1$ puntos ($s > 0$) que contengan x_{i-1} y x_i satisfagan $f[x_l, \dots, x_{l+s}] = O([f])/h^s$. Por tanto, para cualquier polinomio interpolante $q_l(x)$ que utilice un *stencil* que crucen la discontinuidad: $f(x) = q_l(x) + O([f])$, con x perteneciente a dicho *stencil*.

Para singularidades débiles, es decir, si $f(x)$ posee una discontinuidad en $f^{(p)}(x_d)$ para $p \geq 1$ y $x_d \in I_i$ se cumple que cualquier diferencia dividida basada en $s + 1$ puntos que contengan x_{i-1} y x_i :

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

donde $\|f^{(s)}\|$ es la norma supremo de $f^{(s)}$ en el conjunto convexo del *stencil* usado para calcular la diferencia dividida y $[f^{(p)}] = f^{(p)}(x_d^+) - f^{(p)}(x_d^-)$.

Por tanto, si $\overline{\mathcal{IP}}(x)$ es un interpolante formado por trozos polinómicos de grado r , cualquier trozo $q_l(x)$ formado a partir de un *stencil* que cruce una singularidad cumplirá:

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

Si utilizamos una técnica de interpolación centrada, con *stencils* de $2s$ puntos ($r = 2s - 1$), cada singularidad en el intervalo I_i afectará a un intervalo de $2s$ puntos, $[x_{i-s-1}, \dots, x_{i+s}]$ donde la aproximación empeorará. Para evitarlo se introduce la técnica *ENO*, que veremos en el siguiente apartado.

1.2.2

Técnica *ENO*

Harten et al. en [47] introducen la técnica de interpolación *ENO* (*Essentially Non-Oscillatory*) para evitar la pérdida de aproximación alrededor de los saltos (*fenómeno de Gibbs*) al utilizar técnicas de interpolación centradas. La idea se basa en elegir *stencils*

18 1.2. Esquemas de multirresolución para valores puntuales

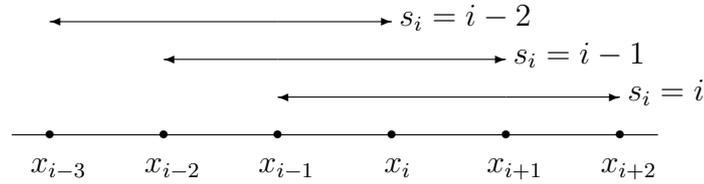


Figura 1.2: Posibles elecciones de stencils de 4 nodos para el intervalo $I_i = [x_{i-1}, x_i]$.

que no crucen discontinuidades, construyendo los trozos polinomiales a partir de regiones suaves. Esto es posible hacerlo siempre que las discontinuidades estén lo suficientemente separadas, tanto entre ellas como de las fronteras (si los datos no son periódicos) y siempre que no trabajemos con el propio intervalo que contiene la discontinuidad. Por ejemplo, para el intervalo $I_i = [x_{i-1}, x_i]$ y stencils de 4 nodos, podemos elegir entre $\mathcal{S}_i = \{x_{s_i-1}, x_{s_i}, x_{s_i+1}, x_{s_i+2}\}$ para s_i que puede tomar los valores $i - 2, i - 1, i$ como vemos en la Figura 1.2.

Para decidir si un stencil es o no suave nos basamos en la magnitud de las diferencias divididas. El stencil ENO de $r + 1$ puntos elegido para el intervalo I_i lo denotamos $\mathcal{S}_i^{ENO} = \{x_{s_i-1}, \dots, x_{s_i+r-1}\}$.

Los algoritmos más utilizados para evitar discontinuidades, [16, 4, 3, 28, 47], son el jerárquico, ENO-VP-I (Algoritmo 1.1) y el no jerárquico, ENO-VP-II (Algoritmo 1.2).

NOTA 1.5. Para $r = 1$, $\mathcal{S}_i = \{x_{i-1}, x_i\}$ y no hay ninguna elección a realizar. Para $r = 2$, ambos algoritmos producen el mismo resultado. Si $r > 2$ los stencils elegidos mediante cada algoritmo sí pueden cambiar. El algoritmo ENO-VP-I proporciona en regiones suaves stencils más centrados respecto al intervalo de trabajo, dando mejores aproximaciones que el algoritmo ENO-VP-II, pero puede fallar para discontinuidades de $f^{(p)}$ con $p > 1$, [16]. ENO-VP-II siempre produce stencils que evitan las discontinuidades. Por tanto, para singularidades débiles debemos utilizar el algoritmo ENO-VP-II, mientras que para saltos y esquinas podemos emplear cualquiera de los dos.

Algoritmo 1.1 *ENO-VP-I*. Elección jerárquica del *stencil*.

Entrada: $\{f_i\}_{i=1}^J$, los datos iniciales; $r + 1$, tamaño del *stencil*

Salida: $\{s_i\}_{i=1}^{J-r+1}$

```

1: para  $i = 1, \dots, J - r + 1$  hacer
2:    $s_0 = i$ 
3:   para  $l = 0, \dots, r - 2$  hacer
4:     si  $|f[x_{s_l-2}, \dots, x_{s_l+l}]| < |f[x_{s_l-1}, \dots, x_{s_l+l+1}]|$  entonces
5:        $s_{l+1} = s_l - 1$ 
6:     si no
7:        $s_{l+1} = s_l$ 
8:     fin si
9:   fin para
10:   $s_i = s_{r-1}$ 
11: fin para

```

Algoritmo 1.2 *ENO-VP-II*. Elección no jerárquica del *stencil*.

Entrada: $\{f_i\}_{i=1}^J$, los datos iniciales; $r + 1$, tamaño del *stencil*

Salida: $\{s_i\}_{i=1}^{J-r+1}$

```

para  $i = 1, \dots, J - r + 1$  hacer
   $|f[x_{s_i-1}, \dots, x_{s_i+r-1}]| = \min \{|f[x_{l-1}, \dots, x_{l+r-1}]|, i - r + 1 \leq l \leq i\}$ 
fin para

```

Si las discontinuidades están suficientemente separadas, el algoritmo *ENO-VP-II* (y el algoritmo *ENO-VP-I* para saltos y esquinas) proporciona un interpolante polinomial a trozos que satisface:

$$\frac{d^m}{dx^m} \overline{\mathcal{I}\mathcal{P}}(x \pm 0; \mathcal{D}f) = \frac{d^m}{dx^m} f(x) + O(h^{r+1-m}), \quad 0 \leq m \leq r,$$

excepto cuando x pertenece al intervalo que contiene la discontinuidad.

Tratemos la cuestión de la separación de discontinuidades. Para cada intervalo que no posea discontinuidad deben existir tantos intervalos libres de discontinuidades como para que pueda tomarse al menos un *stencil* en una región suave. Esto significa que si estamos utilizando *stencils* de longitud $r + 1$ debe haber al menos r intervalos no discontinuos. Ver la Figura 1.3.

20 1.2. Esquemas de multirresolución para valores puntuales

Stencil ENO asignado a I_j , para $j = i - 1, \dots, i + 2$

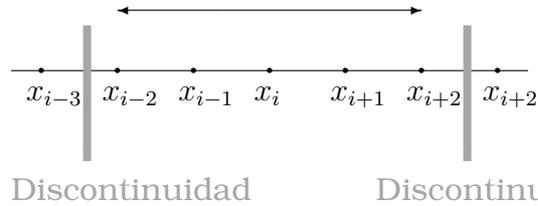


Figura 1.3: Separación mínima entre dos discontinuidades para que la elección ENO proporcione stencils que no crucen saltos para intervalos no discontinuos. En este caso utilizamos stencils de 5 ($r + 1$) puntos, debiendo existir al menos 4 (r) intervalos entre discontinuidades.

NOTA 1.6. Existe un caso especial en el que a pesar de haber discontinuidad no se produce pérdida de aproximación en ningún intervalo. Esto ocurre cuando la discontinuidad es una esquina que cae justo en un punto del grid, x_d , ya que en este caso, los algoritmos ENO-VP-I y ENO-VP-II evitan cruzar dicha discontinuidad, tanto para el intervalo $[x_{d-1}, x_d]$ como para $[x_d, x_{d+1}]$.

1.2.3

Filtros para esquemas en valores puntuales

En este apartado expondremos los filtros para los grados más usuales.

Caso lineal

Como veremos más adelante, al sustituir $\overline{\mathcal{I}\mathcal{P}}_{k-1}(x_i^k; f^{k-1})$ por los polinomios interpoladores centrados de grado $r = 2s - 1$, los esquemas (1.13) y (1.14) quedan:

Codificación:

$$\left\{ \begin{array}{l} \text{Para } k = L, \dots, 1 \\ f_i^{k-1} = f_{2i}^k, \\ d_i^k = f_{2i-1}^k - \sum_{l=1}^s (\beta_{s,s}^{2s-1})_l (f_{i+l-1}^{k-1} + f_{i-l}^{k-1}), \end{array} \right. \quad \begin{array}{l} i = 0, \dots, J_{k-1}, \\ i = 1, \dots, J_{k-1}. \end{array}$$

Decodificación:

$$\left\{ \begin{array}{l} \text{Para } k = 1, \dots, L \\ f_{2i-1}^k = d_i^k + \sum_{l=1}^s (\beta_{s,s}^{2s-1})_l (f_{i+l-1}^{k-1} + f_{i-l}^{k-1}), \quad i = 1, \dots, J_{k-1}, \\ f_{2i}^k = f_i^{k-1}, \quad i = 0, \dots, J_{k-1}. \end{array} \right.$$

donde:

$$\left\{ \begin{array}{l} r = 1 (s = 1) \implies (\beta_{1,1}^1)_1 = \frac{1}{2}. \\ r = 3 (s = 2) \implies (\beta_{2,2}^3)_1 = \frac{9}{16}, (\beta_{2,2}^3)_2 = -\frac{1}{16}. \\ r = 5 (s = 3) \implies (\beta_{3,3}^5)_1 = \frac{150}{256}, (\beta_{3,3}^5)_2 = -\frac{25}{256}, (\beta_{3,3}^5)_3 = \frac{3}{256}. \end{array} \right.$$

Observar que si $f(x) = P(x)$, siendo $P(x)$ un polinomio de grado menor o igual a r , entonces $q_i^{\overline{\mathcal{I}\mathcal{P}}}(x; f^{k-1}, r) = f(x)$ para $x \in [x_{i-1}^{k-1}, x_i^{k-1}]$, es decir $\overline{\mathcal{I}\mathcal{P}}_{k-1}(x, f^{k-1}) = f(x)$. Con esto podemos enunciar:

PROPOSICIÓN 1.2. Si $q_i^{\overline{\mathcal{I}\mathcal{P}}}(x; f^{k-1}, r)$ es el polinomio $\overline{\mathcal{I}\mathcal{P}}$ de grado r , la reconstrucción $\overline{\mathcal{I}\mathcal{P}}$ recupera de forma exacta polinomios de grado menor o igual a r .

COROLARIO 1.1. El orden de la reconstrucción $\overline{\mathcal{I}\mathcal{P}}$ es $r + 1$.

Caso ENO

El calcular los filtros para el caso *ENO*, es equivalente a calcular los filtros para hallar los valores de interpolación en las fronteras en funciones no periódicas. En ambos casos se trata de realizar una interpolación no centrada.

Veamos como calcular los filtros³. Fijado un k , buscamos un polinomio de grado r , $q(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$, que interpole a $\{f_i^{k-1}\}$ en $r + 1$ nodos.

³En la literatura sobre el tema, es usual calcular los filtros desarrollando las formas de Newton de los polinomios interpoladores. Sin embargo, en este trabajo los calcularemos mediante sistemas, para dar otra visión y por consonancia con el cálculo de filtros de capítulos posteriores.

22 1.2. Esquemas de multirresolución para valores puntuales

NOTA 1.7. Con el fin de simplificar la exposición y sin pérdida de generalidad, calcularemos $q(x)$ sobre los nodos $0, 1, 2, \dots$ y no sobre $\{x_i^k\}_i$, ya que éstos los suponemos equiespaciados.

Tomamos nl nodos a la izquierda de x_{2i-1}^k y nr a la derecha ($r = nl + nr - 1$). Queremos que el polinomio satisfaga:

$$q(j) = f_{i+j-1}^{k-1}, \quad j = -nl + 1, -nl + 2, \dots, nr. \quad (1.18)$$

En forma matricial, tenemos la matriz $(r + 1) \times (r + 1)$

$$A_{nl,nr}^{\overline{IP}} = \begin{bmatrix} 1 & (-nl + 1) & \cdots & (-nl + 1)^{nr+nr} \\ & \vdots & \ddots & \vdots \\ 1 & -1 & \cdots & (-1)^{nl+nr} \\ 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1^{nl+nr} \\ 1 & 2 & \cdots & 2^{nl+nr} \\ & \vdots & \ddots & \vdots \\ 1 & (nr) & \cdots & (nr)^{nl+nr} \end{bmatrix}, \quad (1.19)$$

y los vectores $\hat{a} = [a_1, \dots, a_{r+1}]^T$ y $\hat{f} = [f_{i-nl}^{k-1}, \dots, f_{i+nr-1}^{k-1}]^T$.

Para obtener los coeficientes $\{a_1, \dots, a_{nl+nr}\}$ del polinomio $q(x)$ debemos calcular la solución de $A_{nl,nr}^{\overline{IP}} \hat{a} = \hat{f}$. Dado que $A_{nl,nr}^{\overline{IP}}$ es una matriz de *Vandermonde*, $\det(A_{nl,nr}^{\overline{IP}}) \neq 0$, y la solución del sistema es:

$$\hat{a} = \left(A_{nl,nr}^{\overline{IP}} \right)^{-1} \hat{f}. \quad (1.20)$$

Al polinomio que hemos calculado lo denotaremos $p_{s_i}^{\overline{IP}}(x; f^{k-1}, r)$.

NOTA 1.8. La notación $p_{s_i}^{\overline{IP}}(x; f^{k-1}, r)$ se empleará al utilizar los nodos como enteros. Cuando trabajemos sobre los nodos $\{x_i^{k-1}\}_i$ se expresará $q_{s_i}^{\overline{IP}}(x; f^{k-1}, r)$.

NOTA 1.9. Utilizamos el subíndice s_i para indicar que el stencil es no centrado según la elección ENO. En este caso el stencil utilizado es $\{x_{i-nl}^{k-1}, \dots, x_{i+nr-1}^{k-1}\} \equiv \{x_{s_i-1}^{k-1}, \dots, x_{s_i+r-1}^{k-1}\}$, es decir, $s_i = i - nl + 1$. Dicha notación se utilizará a lo largo del presente trabajo.

El valor que aproxima a f_{2i-1}^k será $p_{s_i}^{\overline{\mathcal{IP}}}(\frac{1}{2}; f^{k-1}, r)$. Así:

$$\begin{aligned} \tilde{f}_{2i-1}^k &= \sum_{l=1}^{nr+nl} a_l \frac{1}{2^{l-1}} = \left(1, \frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^r}\right) \left(A_{nl, nr}^{\overline{\mathcal{IP}}}\right)^{-1} \hat{f} \\ &= \sum_{l=-nl}^{nr-1} (\beta_{nl, nr}^r)_l f_{i+l} \equiv \overline{\mathcal{IP}}_{nl, nr}^r(x_{2i-1}^k, f^{k-1}), \end{aligned}$$

siendo⁴:

$$(\beta_{nl, nr}^r)_l = \left(1, \frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^r}\right) \left(A_{nl, nr}^{\overline{\mathcal{IP}}}\right)^{-1} e_l, \quad (1.21)$$

con $l = -nl, -nl + 1, \dots, nr - 1$ y $e_j = (\delta_{j,k})_{k=-nl}^{nr-1}$.

Debido a la simetría de la interpolación tenemos

$$(\beta_{nl, nr}^r)_l = (\beta_{nr, nl}^r)_{-l-1}, \quad l = -nl, \dots, nr - 1.$$

En la Tabla 1.1 vemos⁵ los resultados obtenidos para diferentes r , nl y nr con $r = nl + nr - 1$.

1.2.4

Técnica WENO

Mediante la interpolación ENO se consigue una aproximación con un buen orden de exactitud en todos los intervalos excepto en

⁴Dado que el nivel k , en la notación $\overline{\mathcal{IP}}_k(x; f^k, r)$ está presente en f^k , utilizaremos también la notación $\overline{\mathcal{IP}}_{nl, nr}^r(x; f^k, r)$ cuando necesitemos indicar el grado y el stencil del polinomio de reconstrucción

⁵En <http://www.uv.es/anims/noguera/> podemos descargar una GUI y programas en MATLAB® que calculan dichos filtros para cualquier combinación introducida.

24 1.2. Esquemas de multirresolución para valores puntuales

| l | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
|---------------------|-----------------|-------------------|------------------|--------------------|-------------------|-------------------|-------------------|------------------|------------------|
| $(\beta_{2,1}^2)_l$ | | | | $-\frac{1}{8}$ | $\frac{3}{4}$ | $\frac{3}{8}$ | | | |
| $(\beta_{1,2}^2)_l$ | | | | | $\frac{3}{8}$ | $\frac{3}{4}$ | $-\frac{1}{8}$ | | |
| $(\beta_{3,1}^3)_l$ | | | $\frac{1}{16}$ | $-\frac{5}{16}$ | $\frac{15}{16}$ | $\frac{5}{16}$ | | | |
| $(\beta_{2,2}^3)_l$ | | | | $-\frac{1}{16}$ | $\frac{9}{16}$ | $\frac{9}{16}$ | $-\frac{1}{16}$ | | |
| $(\beta_{1,3}^3)_l$ | | | | | $\frac{5}{16}$ | $\frac{15}{16}$ | $-\frac{5}{16}$ | $\frac{1}{16}$ | |
| $(\beta_{5,1}^5)_l$ | $\frac{7}{256}$ | $-\frac{45}{256}$ | $\frac{63}{128}$ | $-\frac{105}{128}$ | $\frac{315}{256}$ | $\frac{63}{256}$ | | | |
| $(\beta_{4,2}^5)_l$ | | $-\frac{3}{256}$ | $\frac{21}{256}$ | $-\frac{35}{128}$ | $\frac{105}{128}$ | $\frac{105}{256}$ | $-\frac{7}{256}$ | | |
| $(\beta_{3,3}^5)_l$ | | | $\frac{3}{256}$ | $-\frac{25}{256}$ | $\frac{75}{128}$ | $\frac{75}{128}$ | $-\frac{25}{256}$ | $\frac{3}{256}$ | |
| $(\beta_{2,4}^5)_l$ | | | | $-\frac{7}{256}$ | $\frac{105}{256}$ | $\frac{105}{128}$ | $-\frac{35}{128}$ | $\frac{21}{256}$ | $-\frac{3}{256}$ |
| f_{i+l}^{k-1} | f_{i-5}^{k-1} | f_{i-4}^{k-1} | f_{i-3}^{k-1} | f_{i-2}^{k-1} | f_{i-1}^{k-1} | f_i^{k-1} | f_{i+1}^{k-1} | f_{i+2}^{k-1} | f_{i+3}^{k-1} |

Tabla 1.1: Aproximaciones a f_{2i-1}^{k-1} , de orden $r + 1 = nl + nr$, $\overline{\mathcal{IP}}_{nl,nr}^r(x_{2i-1}^k, f^{k-1}) = \sum_{l=-nl}^{nr-1} (\beta_{nl,nr}^r)_l f_{i+l}^{k-1}$, utilizando interpolación con nl puntos a la izquierda y nr puntos a la derecha de x_{2i-1}^k .

aquellos que contienen singularidades. Pero existen aspectos 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 redondeo, por ejemplo, podría hacer que el *stencil* seleccionado cambiase, porque podría cambiar la dirección de la desigualdad.

Por otra parte, en las regiones suaves de la función no es necesario realizar esta selección, ya que el *stencil* centrado conseguirá una buena 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. Si tomamos *stencils* formados por r celdas, 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, se podría alcanzar una mejor aproximación gracias a la información dada por las $2r - 1$ celdas contenidas en los distintos *stencils*, pudiendo obtener un orden de exactitud máximo igual a $2r - 1$.

Para solucionar los dos primeros problemas, se presentó en [50] 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 se utilizará 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. [52] 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 [50].

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 contruidos 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

26 1.2. Esquemas de multirresolución para valores puntuales

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.

Considerando $\mathcal{S}_{i+k} = \{x_{i+k-r}, \dots, x_{i+k}\}$, $k = 0, \dots, r-1$, los r *stencils* que contienen al intervalo I_i , y denotando $p_{i-s+1+k}^{\overline{\mathcal{IP}}^r_{s,s}}(x)$ el polinomio de grado $r = 2s-1$ construido a partir del *stencil* \mathcal{S}_{i+k} , definimos el polinomio interpolador *WENO* en el intervalo I_i como

$$p_i^{\text{WENO}-\overline{\mathcal{IP}}}(x) = \sum_{k=0}^{r-1} \omega_k^i p_{i-s+1+k}^{\overline{\mathcal{IP}}^r_{s,s}}(x), \quad (1.22)$$

donde

$$\omega_k^i \geq 0, \quad k = 0, \dots, r-1, \quad \text{y} \quad \sum_{k=0}^{r-1} \omega_k^i = 1. \quad (1.23)$$

NOTA 1.10. En este trabajo asumiremos que en (1.22) los pesos deben cumplir $\omega_k^i \geq 0$, aunque existen artículos, [66], donde se estudia el tratamiento de pesos negativos.

Notar que para cada intervalo, el interpolante *WENO* toma información de $2r$ nodos. De este modo se pretende alcanzar una aproximación de orden $2r$ en aquellos intervalos donde la función sea suave. Veamos como podemos conseguir dicho orden de aproximación. Sea $p_i^{\overline{\mathcal{IP}}^{2r-1}_{r,r}}$ la aproximación obtenida empleando el *stencil* formado por $2r$ nodos $\{x_{i-r}, \dots, x_{i+r-1}\}$, y sean $p_{i-s+1+k}^{\overline{\mathcal{IP}}^r_{s,s}}$ las aproximaciones obtenidas con los *stencils* \mathcal{S}_{i+k} . Podemos expresar la aproximación de orden $2r$ como una combinación lineal de las r aproximaciones de orden $r+1$ (Figura 1.4), i.e.:

$$p_i^{\overline{\mathcal{IP}}^{2r-1}_{r,r}}(x) = \sum_{k=0}^{r-1} C_k^r p_{i-s+1+k}^{\overline{\mathcal{IP}}^r_{s,s}}(x), \quad (1.24)$$

con $C_k^r \geq 0$, $\forall k$ y $\sum_{k=0}^{r-1} C_k^r = 1$. Las constantes C_k^r se denominan *pesos óptimos*. La Tabla 1.2 nos muestra los pesos óptimos para

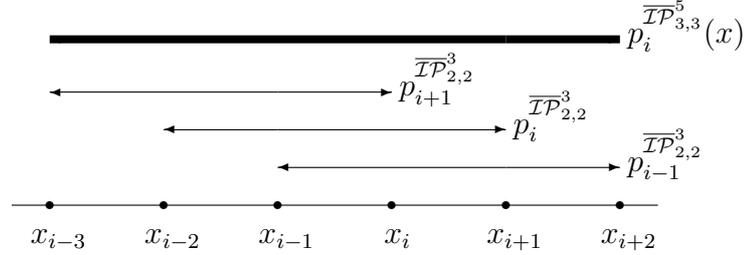


Figura 1.4: Esquema de la aproximación de orden $2r$ como C.L. de las r aproximaciones de orden $r+1$ ($r = 2s - 1$): $p_i^{\overline{\mathcal{IP}}_{r,r}^{2r-1}}(x) = \sum_{k=0}^{r-1} C_k^r p_{i-s+1+k}^{\overline{\mathcal{IP}}_{s,s}^r}(x)$, para $r = 3$.

$r = 2, 3$. En [14, 21] podemos encontrar los detalles sobre cómo obtener dichos pesos.

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

Tabla 1.2: Pesos óptimos para $r = 2, 3$.

Si definimos los pesos ω_k^i de modo que verifiquen la siguiente relación:

$$\omega_k^i = C_k^r + O(h^{r-1}), \quad k = 0, \dots, r-1, \quad (1.25)$$

se puede demostrar, ver [50, 21], que la aproximación obtenida es de orden $2r$. Además, se desea que la contribución de los polinomios cuyo *stencil* cruce alguna singularidad sea lo menor posible, y por tanto sus pesos en (1.22) deben ser muy próximos a 0.

Con el objetivo de satisfacer las condiciones anteriores, se definen los pesos como sigue:

$$\omega_k^i = \frac{\alpha_k^i}{\sum_{s=0}^{r-1} \alpha_s^i}, \quad k = 0, \dots, r-1, \quad \text{con } \alpha_k^i = \frac{C_k^r}{(\epsilon + \mathcal{IS}_{i+k})^2}. \quad (1.26)$$

Notar que el valor ϵ introducido en (1.26) es una constante positiva para evitar que el denominador se anule, y \mathcal{IS}_{i+k} es un indicador de suavidad de $f(x)$ en el *stencil* \mathcal{S}_{i+k} .

28 1.2. Esquemas de multirresolución para valores puntuales

Resulta evidente que así definidos, los pesos verifican (1.23) independientemente del indicador de suavidad utilizado. Para que la contribución de los polinomios que cruzan una singularidad sea prácticamente nula, es suficiente con que $\mathcal{IS}_{i+k} = O(h^2)$ para aquellos *stencils* en los que $f(x)$ sea suave, y $\mathcal{IS}_{i+k} = O(1)$ en aquellos *stencils* que la función presente alguna singularidad. Se puede demostrar que si el indicador de suavidad verifica

$$\mathcal{IS}_{i+k} = D(1 + O(h^{r-1})), \quad k = 0, \dots, r-1, \quad (1.27)$$

se satisface la relación (1.25), ver [14, 21], donde D es una constante no nula que puede depender de h pero no de k . En particular se tiene el siguiente resultado:

PROPOSICIÓN 1.3. *Si tenemos*

$$\mathcal{IS}_{j+l} = D(1 + O(h^m)), \quad l = 0, \dots, r-1,$$

con D una constante distinta de cero e independiente de l , aunque puede depender de h , entonces, si $r = 2s - 1$:

$$\omega_l^j = C_l^j + O(h^m) \quad \text{y} \quad \sum_{l=0}^{r-1} \omega_l^j \overline{\mathcal{IP}}_{j-s+1+l}^r = f_{2j-1} + O(h^{\min(r+m+1, 2r)}).$$

Liu et al.([52]) proponen el siguiente indicador de suavidad basado en las diferencias no divididas de $f(x)$ en los nodos de interpolación,

$$\mathcal{IS}_{i+k} = \sum_{l=1}^r \left(\sum_{m=1}^l (\Delta^{r-l+1}[f_{i+k+m-r-1}])^2 \right) / l, \quad (1.28)$$

donde $\Delta^s[f_l]$ no es más que la diferencia dividida de f en los nodos $\{x_l, \dots, x_{l+s}\}$. Se puede ver que con este indicador de suavidad se alcanza el orden óptimo 4, si $r = 2$, pero sin embargo cuando $r = 3$ se obtiene orden 5, siendo 6 el orden óptimo en este caso.

Jiang y Shu, definen en [52] un nuevo indicador de suavidad,

$$\mathcal{IS}_{i+k} = \sum_{l=1}^r \int_{x_{i-1}}^{x_i} h^{2l-1} (p_{i+k}^{(l)}(x))^2 dx, \quad (1.29)$$

donde $p_{i+k}^{(l)}(x)$ representa la l -ésima derivada del polinomio interpolador basado en el *stencil* S_{i+k} . Este nuevo indicador hace posible alcanzar órdenes óptimos tanto para el caso $r = 2$ como $r = 3$. Las demostraciones de los órdenes de exactitud alcanzados con ambos métodos, para $r = 2, 3$, se pueden consultar en [14, 21].

1.2.5

Técnica PPH

En el contexto de compresión de imágenes, se ha presentado en [6, 30] un nuevo método basado en una multirresolución unidimensional no lineal, denominada *PPH (Piecewise Polynomial Harmonic)*. En [7] se describe *PPH* como una modificación de la multirresolución lineal, lo cual permite demostrar la convergencia y estabilidad del método. En [2] se aplica al procesamiento de señales digitales tales como compresión y eliminación de ruido de imágenes, obteniendo resultados satisfactorios.

Pasemos a describir este método de cuarto orden. Como:

$$\overline{\mathcal{I}\mathcal{P}}_{2,2}^3(x_{2i-1}^k, f^{k-1}) = -\frac{1}{16}f_{i-2}^{k-1} + \frac{9}{16}f_{i-1}^{k-1} + \frac{9}{16}f_i^{k-1} - \frac{1}{16}f_{i+1}^{k-1},$$

tras una simple reordenación:

$$\overline{\mathcal{I}\mathcal{P}}_{2,2}^3(x_{2i-1}^k, f^{k-1}) = \frac{1}{2}(f_{i-1}^{k-1} + f_i^{k-1}) - \frac{1}{8} \left(\frac{d^2 f_{i-1}^{k-1} + d^2 f_i^{k-1}}{2} \right),$$

con $d^2 f_n = f_{n-1} - 2f_n + f_{n+1}$. La multirresolución *PPH* se basa en sustituir la media aritmética del segundo sumando por la media armónica, obteniendo:

$$\left(\overline{\mathcal{I}\mathcal{P}}_{2,2}^3 \right)^{PPH} (x_{2i-1}^k, f^{k-1}) = \frac{1}{2}(f_{i-1}^{k-1} + f_i^{k-1}) - \frac{1}{8}H(d^2 f_{i-1}^{k-1}, d^2 f_i^{k-1}),$$

siendo $H(x, y)$ la media armónica, es decir:

$$H(x, y) := \frac{(\text{sign}(x) + \text{sign}(y))}{2} \frac{2|xy|}{|x| + |y|}, \quad (1.30)$$

con $\text{sign}(x) = 1$ si $x \geq 0$ y $\text{sign}(x) = -1$ si $x < 0$.

1.2.6

Técnica de resolución subcelda (SR)

Con el fin de mejorar la aproximación en el intervalo donde se halla la discontinuidad, surge la técnica *SR*. Fue desarrollada por Harten, [43], en el contexto de esquemas *ENO* para *Leyes de Conservación Hiperbólicas*. Básicamente se trata de extender de manera adecuada los interpolantes adyacentes a un intervalo hasta el punto de discontinuidad. Esta técnica desarrolla su máxima potencialidad para discretizaciones por medias en celda, aunque la describiremos para valores puntuales. En [11] podemos ver unos resultados teóricos, en [13] unas aplicaciones 2D para valores puntuales y en [12] una generalización para medias en celda en 2D.

Denotamos⁶ $q_{s_i}(x)$ al polinomio interpolador de $f(x)$ formado a partir del *stencil ENO*, S_i^{ENO} . Supongamos que $f(x)$ es una función continua con una esquina en $x_d \in (x_{i-1}, x_i)$ y que los polinomios $q_{s_{i\pm 1}}(x)$ satisfacen:

$$q_{s_{i-1}}(x) = f(x) + O(h^{r+1}) \quad x \in [x_{i-2}, x_{i-1}], \quad (1.31)$$

$$q_{s_{i+1}}(x) = f(x) + O(h^{r+1}) \quad x \in [x_i, x_{i+1}]. \quad (1.32)$$

La localización del salto puede hallarse (con orden de aproximación $O(h^{r+1})$) definiendo la siguiente función:

$$G_i(x) := q_{s_{i+1}}(x) - q_{s_{i-1}}(x). \quad (1.33)$$

Desarrollando por Taylor, en las regiones suaves obtenemos:

$$G_i(x) = (x - x_d)[f'] + O(x - x_d)^2.$$

Luego para h suficientemente pequeño, debe existir una y sólo una raíz de G_i en (x_{i-1}, x_i) . Llamemos a dicha raíz $\xi \in (x_{i-1}, x_i)$, es decir $G_i(\xi) = 0$.

⁶Este polinomio ya lo denotamos como $q_{s_i}^{LP}(x; f^{k-1}, r)$, pero en este apartado utilizaremos esta notación menos engorrosa.

En lo que sigue veremos qué error cometemos al aproximar x_d por ξ . Supongamos que $f(x)$ es una función que cumple:

$$f(x) = \begin{cases} P_L(x), & x \leq x_d, \\ P_R(x), & x \geq x_d, \end{cases}$$

siendo $P_L(x)$ y $P_R(x)$ dos funciones suficientemente suaves tales que $P_L(x_d) = P_R(x_d)$ y $P'_L(x_d) \neq P'_R(x_d)$. Notar que en el caso particular en el que P_L y P_R sean polinomios de grado menor o igual a r , las piezas polinómicas son de aproximación exacta:

$$q_{s_{i-1}}(x) = P_L(x), \quad q_{s_{i+1}}(x) = P_R(x), \quad (1.34)$$

con lo cual $\xi = x_d$.

En el caso general, podemos reescribir (1.31) y (1.32) como:

$$q_{s_{i-1}}(x) = P_L(x) + O(h^{r+1}), \quad q_{s_{i+1}}(x) = P_R(x) + O(h^{r+1}).$$

Dado que $G_i(x) = P_R(x) - P_L(x) + O(h^{r+1})$ si definimos $\hat{G} := P_R(x) - P_L(x)$, entonces

$$G_i(x) = \hat{G}(x) + O(h^{r+1}) \Rightarrow G_i(\xi) = \hat{G}(\xi) + O(h^{r+1}).$$

Desarrollando por Taylor $\hat{G}(x)$ alrededor de x_d , obtenemos:

$$0 = G_i(\xi) = \hat{G}(x_d) + (x_d - \xi)\hat{G}'(x_d) + \cdots + O(h^{r+1}).$$

Dado que $\hat{G}(x_d) = 0$, $\hat{G}'(x_d) \neq 0$ y $|\xi - x_d| \leq h$:

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

Una vez localizada la discontinuidad para realizar la *técnica de resolución subcelda*, SR (*Subcelda Resolution*), se extienden los interpolantes adyacentes hasta ξ de la siguiente manera (ver las Figuras 1.5 y 1.6):

$$\overline{\mathcal{IP}}^{SR}(x) = \begin{cases} q_{s_l}(x), & x \in [x_{l-1}, x_l], \quad l \neq i, \\ q_{s_{i-1}}(x), & x \in [x_{i-1}, \xi], \\ q_{s_{i+1}}(x), & x \in [\xi, x_i]. \end{cases}$$

Es claro que $\overline{\mathcal{IP}}^{SR}(x_l) = f(x_l)$, $\forall x_l \in X$ y

$$\frac{d^m}{dx^m} \overline{\mathcal{IP}}(x \pm 0; \mathcal{D}f) = \frac{d^m}{dx^m} f(x) + O(h^{r+1-m}), \quad 0 \leq m \leq r,$$

para todos los puntos excepto para un entorno de orden $O(h^{r+1})$ alrededor de x_d .

32 1.2. Esquemas de multirresolución para valores puntuales

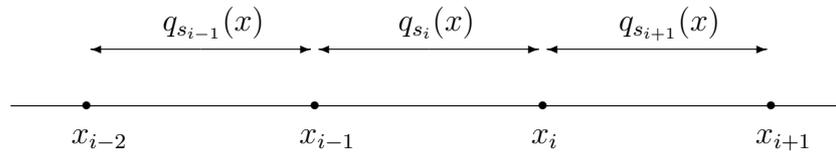


Figura 1.5: Alcance de las piezas polinómicas en la técnica ENO.

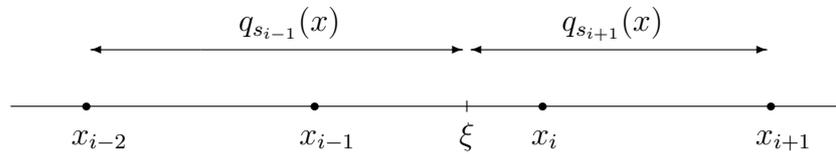


Figura 1.6: Alcance de las piezas polinómicas en la técnica SR.

NOTA 1.11. Al aplicar la técnica SR para singularidades en las cuales $[f^{(p)}] = O(1)$ con $p > 1$ hay que tener en cuenta:

- Se deben elegir los stencils ENO de forma no-jerárquica, es decir mediante ENO-VP-II, Algoritmo 1.2.
- Un salto en $f^{(p)}(x)$ es una esquina en $f^{(p-1)}(x)$. Por tanto la función $G(x)$ debe ser sustituida por $G^{(p-1)}(x)$ en (1.33).

Filtros SR para valores puntuales

Apliquemos las técnicas ENO-SR para discretizaciones por valores puntuales. Las funciones interpolatorias que utilizamos vienen dadas por:

$$\overline{\mathcal{IP}}_{k-1}(x, f^{k-1}) = \overline{q}_{s_i}^{\mathcal{IP}}(x; f^{k-1}, r), \quad x \in [x_{i-1}^{k-1}, x_i^{k-1}] \quad , 1 \leq i \leq J_k, \quad (1.35)$$

donde $\overline{q}_{s_i}^{\mathcal{IP}}(x; f^{k-1}, r)$ representa el único polinomio de grado r que interpola a $f(x)$ en el stencil S_i^{ENO} .

Como vimos, si $f(x) = P(x)$, siendo $P(x)$ un polinomio de grado menor o igual a r ,

$$\overline{\mathcal{I}\mathcal{P}}_{k-1}(x, f^{k-1}) = q_{s_i}^{\overline{\mathcal{I}\mathcal{P}}}(x; f^{k-1}, r) = f(x), \quad x \in [x_{i-1}^{k-1}, x_i^{k-1}].$$

Entonces, el espacio de polinomios de grado menor o igual a $r = 2s - 1$, satisface que $\overline{\mathcal{I}\mathcal{P}}_{k-1}\mathcal{D}_{k-1}f = f$. Además para funciones suaves,

$$\overline{\mathcal{I}\mathcal{P}}_{k-1}(x, f^{k-1}) = f(x) + O(h_{k-1})^{2s}.$$

Por tanto, el orden del esquema de reconstrucción es $p = r + 1 = 2s$.

Al utilizar interpolación polinómica a trozos de orden $r + 1$ se obtiene que los coeficientes d_i^k son los errores de interpolación en los puntos impares de la escala k . Según los resultados de la sección 1.2.1 si las piezas polinómicas son de grado r , en las regiones de suavidad se obtiene:

$$d_l^k = O(h_{k-1}^{r+1}). \quad (1.36)$$

Sin embargo si hay una única discontinuidad, $x_d \in (x_{i-1}^{k-1}, x_i^{k-1})$, tal que $[f^{(p)}]_{x_d} = O(1)$ con $p \leq r$, $d_i^k = O(h_{k-1}^p)$. En las cercanías de la discontinuidad, se crearán una serie de *coeficientes de escala* de magnitud elevada. En concreto, si utilizamos una técnica de interpolación centrada con $r = 2s - 1$, se verán afectados los siguientes coeficientes:

$$d_l^k = O([f^{(p)}]) h_{k-1}^p, \quad l = i - s + 1, \dots, i + s - 1. \quad (1.37)$$

NOTA 1.12. *Es deseable evitar estos intervalos de poca aproximación ya que, por ejemplo, en el contexto de compresión, al eliminar los coeficientes de escala dicha información no podrá ser recuperada por el esquema de predicción. Otra opción es almacenarlos a costa de perder capacidad de compresión.*

Al aplicar la técnica *ENO*, si las discontinuidades están lo suficientemente separadas, sólo se producirá un coeficiente de magnitud elevada por cada discontinuidad.

La región de buena aproximación puede maximizarse aplicando la técnica *SR*, en cuyo caso se obtiene (1.36) para todo $l =$

34 1.2. Esquemas de multirresolución para valores puntuales

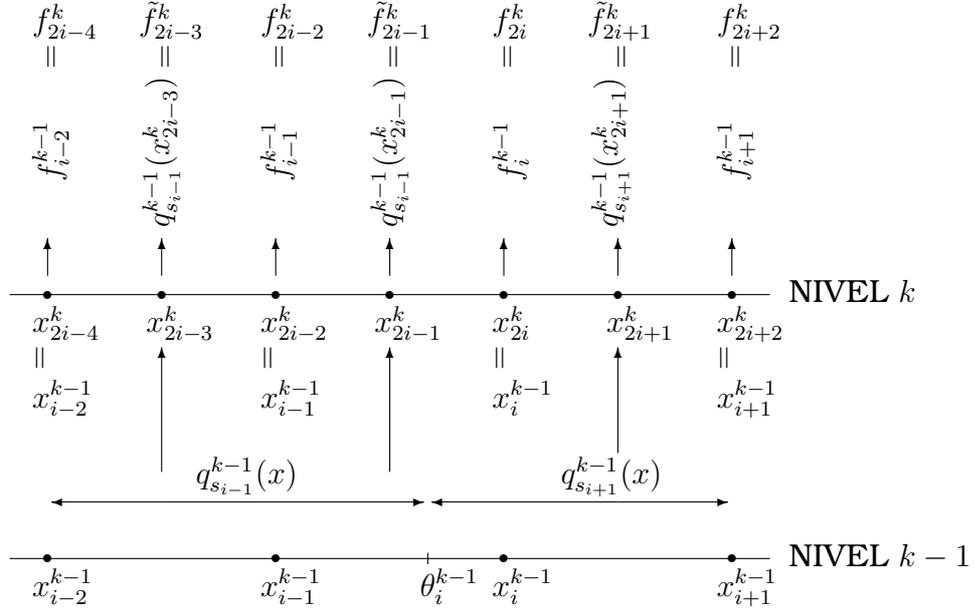


Figura 1.7: Paso del nivel $k-1$ al k mediante multirresolución a partir de discretización por valores puntuales aplicando la técnica ENO-SR al suponer que en el intervalo $(x_{i-1}^{k-1}, x_i^{k-1})$ se encuentra la discontinuidad θ_i^{k-1} . En la Figura $q_{S_i}^k(x) := q_{S_i}^{\overline{IP}}(x; f^k, r)$, es decir, el polinomio construido a partir del stencil S_i^{ENO} que interpola a los valores $\{f^k\}$.

$1, \dots, J_{k-1}$ excepto si el punto x_{2l-1}^k se halla en un entorno de $O(h^{r+1})$ de una singularidad.

A continuación describiremos los pasos necesarios para aplicar ENO-SR en los algoritmos (1.13) y (1.14):

1.- En el nivel $k-1$ calcular los *stencils ENO* utilizando ENO-VP-I (Algoritmo 1.1) o ENO-VP-II (Algoritmo 1.2).

2.- Detectar las posibles *celdas singulares*: si I_i^{k-1} es singular, el *stencil ENO* de la celda I_{i+1}^{k-1} no contendrá el punto x_{i-1}^{k-1} , mientras que el de la celda I_{i-1}^{k-1} no contendrá el punto x_i^{k-1} . Por tanto, si

$$S_{i-1}^{k-1} \cap S_{i+1}^{k-1} = \emptyset, \quad (1.38)$$

etiquetamos la celda como *sospechosa* de contener una singularidad.

3.- Para cada celda sospechosa definir la función

$$G_i^{\overline{TP}}(x) := q_{s_{i+1}}^{\overline{TP}}(x; f^{k-1}, r) - q_{s_{i-1}}^{\overline{TP}}(x; f^{k-1}, r), \quad (1.39)$$

donde $q_{s_i}^{\overline{TP}}(x; h, r)$ representa el polinomio de grado r construido a partir del *stencil* \mathcal{S}_i^{ENO} que interpola a los valores de h . A continuación estudiamos el signo de $G_i^{\overline{TP}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{TP}}(x_i^{k-1})$ para decidir si hay una esquina en la celda sospechosa:

Si $G_i^{\overline{TP}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{TP}}(x_i^{k-1}) < 0 \Rightarrow$ existe una raíz θ_i^{k-1} de $G_i^{\overline{TP}}(x)$, y etiquetamos la celda como *singular*.

4.- Dada una celda singular $(x_{i-1}^{k-1}, x_i^{k-1})$, para decidir si el punto x_{2i-1}^k está a la derecha o a la izquierda de la singularidad procedemos como sigue:

- Si $G_i^{\overline{TP}}(x_{2i-1}^k) \cdot G_i^{\overline{TP}}(x_i^{k-1}) < 0$ tenemos $\theta_i^{k-1} > x_{2i-1}^k$ y

$$\begin{aligned} \tilde{f}_{2i-1}^k &= q_{s_{i-1}}^{\overline{TP}}(x_{2i-1}^k; f^{k-1}, r) = p_{s_{i-1}}^{\overline{TP}}(1/2; f^{k-1}, r) \\ &= \frac{1}{2} \mathbf{1}_{2s}^T \left(A_{2s,0}^{\overline{TP}} \right)^{-1} [f_{i-2s}^{k-1}, \dots, f_{i-1}^{k-1}]^T \\ &= \sum_{l=-2s}^{-1} (\beta_{2s,0}^{2s-1})_l f_{i+l}^{k-1}. \end{aligned} \quad (1.40)$$

donde $(\beta_{2s,0}^{2s-1})_l = \frac{1}{2} \mathbf{1}_{2s}^T \left(A_{2s,0}^{\overline{TP}} \right)^{-1} e_l$, $l = -2s, \dots, -1$, siendo $A_{2s,0}^{\overline{TP}}$ la definida en (1.19), $(\delta_{j,k})_{k=-2s}^{-1}$ y $\mathbf{1}_n = \underbrace{(1, \dots, 1)}_n^T$.

NOTA 1.13. La notación $\mathbf{1}_n = \underbrace{(1, \dots, 1)}_n^T$ se utilizará a lo largo del presente trabajo.

- En otro caso, $\theta_i^{k-1} < x_{2i-1}^k \Rightarrow$

$$\begin{aligned} \tilde{f}_{2i-1}^k &= q_{s_{i+1}}^{\overline{TP}}(x_{2i-1}^k; f^{k-1}, r) = p_{s_{i+1}}^{\overline{TP}}(1/2; f^{k-1}, r) \\ &= \frac{1}{2} \mathbf{1}_{2s} \left(A_{0,2s}^{\overline{TP}} \right)^{-1} [f_i^{k-1}, \dots, f_{i+2s-1}^{k-1}]^T \end{aligned}$$

36 1.2. Esquemas de multirresolución para valores puntuales

$$= \sum_{l=0}^{2s-1} (\beta_{0,2s}^{2s-1})_l f_{i+l}^{k-1}. \quad (1.41)$$

donde $(\beta_{0,2s}^{2s-1})_l = \frac{1}{2} \mathbf{1}_{2s}^T \left(A_{0,2s}^{\overline{\mathcal{IP}}} \right)^{-1} e_l$, $l = 0, \dots, 2s-1$, siendo $e_j = (\delta_{j,k})_{k=0}^{2s-1}$.

En la Tabla 1.3 podemos ver los filtros $\beta_{2s,0}^{2s-1}$ para $s = 2, 3$. Simétricamente obtenemos $\beta_{0,2s}^{2s-1}$ ya que

$$(\beta_{2s,0}^{2s-1})_l = (\beta_{0,2s}^{2s-1})_{-1-l}, \quad l = -2s, \dots, -1.$$

| l | -6 | -5 | -4 | -3 | -2 | -1 |
|---------------------|-------------------|-------------------|--------------------|-------------------|---------------------|-------------------|
| $(\beta_{4,0}^3)_l$ | | | $-\frac{5}{16}$ | $\frac{21}{16}$ | $-\frac{35}{16}$ | $\frac{35}{16}$ |
| $(\beta_{6,0}^5)_l$ | $-\frac{63}{256}$ | $\frac{385}{256}$ | $-\frac{495}{128}$ | $\frac{693}{128}$ | $-\frac{1155}{256}$ | $\frac{693}{256}$ |
| f_{i-l-1}^{k-1} | f_{i-6}^{k-1} | f_{i-5}^{k-1} | f_{i-4}^{k-1} | f_{i-3}^{k-1} | f_{i-2}^{k-1} | f_{i-1}^{k-1} |

Tabla 1.3: Aproximaciones a f_{2i-1}^{k-1} , $\overline{\mathcal{IP}}_{2s,0}^r = \sum_{l=-2s}^{-1} (\beta_{2s,0}^r)_l f_{i+l}^{k-1}$, de orden $r+1 = 2s$, utilizando extrapolación con $2s$ puntos a la izquierda de x_{2i-1}^k .

Los valores que necesitamos para decidir si una celda es singular y para conocer la localización de x_{2i-1}^k respecto a la discontinuidad son $G_i^{\overline{\mathcal{IP}}}(x_{i-1}^{k-1})$, $G_i^{\overline{\mathcal{IP}}}(x_{2i-1}^k)$ y $G_i^{\overline{\mathcal{IP}}}(x_i^{k-1})$:

$$\begin{aligned} G_i^{\overline{\mathcal{IP}}}(x_i^{k-1}) &= q_{s_{i+1}}^{\overline{\mathcal{IP}}}(x_i^{k-1}; f^{k-1}, r) - q_{s_{i-1}}^{\overline{\mathcal{IP}}}(x_i^{k-1}; f^{k-1}, r) \\ &= p_{s_{i+1}}^{\overline{\mathcal{IP}}}(1; f^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{IP}}}(1; f^{k-1}, r) \\ &= f_i^{k-1} - \mathbf{1}_{2s}^T \left(A_{2s,0}^{\overline{\mathcal{IP}}} \right)^{-1} [f_{i-2s}^{k-1}, \dots, f_{i-1}^{k-1}]^T = \sum_{l=-2s}^1 (\sigma_{2s,0}^{2s-1})_l f_{i+l}. \end{aligned}$$

donde $(\sigma_{2s,0}^{2s-1})_1 = 1$ y $(\sigma_{2s,0}^{2s-1})_l = -\mathbf{1}_{2s}^T \left(A_{2s,0}^{\overline{\mathcal{IP}}} \right)^{-1} e_l$, $l = -2s, \dots, 0$, siendo $e_j = (\delta_{j,k})_{k=-2s}^{-1}$.

$$G_i^{\overline{\mathcal{IP}}}(x_{2i-1}^k) = q_{s_{i+1}}^{\overline{\mathcal{IP}}}(x_{2i-1}^k; f^{k-1}, r) - q_{s_{i-1}}^{\overline{\mathcal{IP}}}(x_{2i-1}^k; f^{k-1}, r)$$

$$= \sum_{l=0}^{2s-1} (\beta_{2s,0}^{2s-1})_l f_{i+l} - \sum_{l=0}^{2s-1} (\beta_{0,2s}^{2s-1})_l f_{i+l}, \quad (1.42)$$

con $(\beta_{nl,nr}^r)_l$ los definidos en (1.40) y (1.41). La Tabla 1.3 muestra $\beta_{2s,0}$ para $s = 2, 3$.

$$\begin{aligned} G_i^{\overline{IP}}(x_{i-1}^{k-1}) &= q_{s_{i+1}}^{\overline{IP}}(x_{i-1}^{k-1}; f^{k-1}, r) - q_{s_{i-1}}^{\overline{IP}}(x_{i-1}^{k-1}; f^{k-1}, r) \\ &= p_{s_{i+1}}^{\overline{IP}}(0; f^{k-1}, r) - p_{s_{i-1}}^{\overline{IP}}(0; f^{k-1}, r) \\ &= (1, 0, \dots, 0) \left(A_{0,2s}^{\overline{IP}} \right)^{-1} [f_i^{k-1}, \dots, f_{i+2s-1}^{k-1}]^T - f_{i-1}^{k-1} \\ &= \sum_{l=-1}^{2s-1} (\sigma_{0,2s}^{2s-1})_l f_{i+l}, \end{aligned} \quad (1.43)$$

donde $(\sigma_{0,2s}^{2s-1})_{-1} = -1$, $(\sigma_{0,2s}^{2s-1})_l = (1, 0, \dots, 0) \left(A_{0,2s}^{\overline{IP}} \right)^{-1} e_l$, $l = 0, \dots, 2s - 1$, siendo $e_j = (\delta_{j,k})_{k=0}^{2s-1}$.

En términos de filtros obtenemos:

■ $r = 3$

$$\left\{ \begin{array}{l} G_i^{\overline{IP}}(x_{i-1}^{k-1}) = f_{i-1}^{k-1} + 4f_i^{k-1} - 6f_{i+1}^{k-1} + 4f_{i+2}^{k-1} - f_{i+3}^{k-1}, \\ G_i^{\overline{IP}}(x_{2i-1}^k) = \frac{5}{16}f_{i-4}^{k-1} - \frac{21}{16}f_{i-3}^{k-1} + \frac{35}{16}f_{i-2}^{k-1} - \frac{35}{16}f_{i-1}^{k-1} \\ \quad + \frac{35}{16}f_i^{k-1} - \frac{35}{16}f_{i+1}^{k-1} + \frac{21}{16}f_{i+2}^{k-1} - \frac{5}{16}f_{i+3}^{k-1}, \\ G_i^{\overline{IP}}(x_i^{k-1}) = f_{i-4}^{k-1} - 4f_{i-3}^{k-1} + 6f_{i-2}^{k-1} - 4f_{i-1}^{k-1} + f_i^{k-1}. \end{array} \right.$$

■ $r = 5$

$$\left\{ \begin{array}{l} G_i^{\overline{IP}}(x_{i-1}^{k-1}) = +f_{i-1}^{k-1} + 6f_i^{k-1} - 15f_{i+1}^{k-1} + 20f_{i+2}^{k-1} \\ \quad - 15f_{i+3}^{k-1} + 6f_{i+4}^{k-1} - f_{i+5}^{k-1}, \\ G_i^{\overline{IP}}(x_{2i-1}^k) = -\frac{63}{256}f_{i-6}^{k-1} + \frac{385}{256}f_{i-5}^{k-1} - \frac{495}{128}f_{i-4}^{k-1} + \frac{693}{128}f_{i-3}^{k-1} \\ \quad - \frac{1155}{256}f_{i-2}^{k-1} + \frac{693}{256}f_{i-1}^{k-1} + \frac{693}{256}f_i^{k-1} - \frac{1155}{256}f_{i+1}^{k-1} \\ \quad + \frac{495}{128}f_{i+2}^{k-1} - \frac{495}{128}f_{i+3}^{k-1} + \frac{385}{256}f_{i+4}^{k-1} - \frac{63}{256}f_{i+6}^{k-1}, \\ G_i^{\overline{IP}}(x_i^{k-1}) = -f_{i-6}^{k-1} + 6f_{i-5}^{k-1} - 15f_{i-4}^{k-1} + 20f_{i-3}^{k-1} \\ \quad - 15f_{i-2}^{k-1} + 6f_{i-1}^{k-1} + f_i^{k-1}. \end{array} \right.$$

38 1.2. Esquemas de multirresolución para valores puntuales

NOTA 1.14. *El anterior algoritmo únicamente se aplica para discontinuidades esquina, pero no en saltos, ya que en ese caso en el paso 3 nunca se etiqueta la celda como singular. Por tanto, en valores puntuales la técnica SR no mejora la aproximación para saltos.*

1.2.7

Ejemplos

Veamos sobre una función concreta los diversos métodos. La función utilizada presenta dos discontinuidades. La primera es un salto y la segunda una esquina. Su definición analítica es la siguiente:

$$f(x) = \begin{cases} -\frac{4x-3}{5} \sin\left(\frac{3}{2}\pi\left(\frac{4x-3}{5}\right)^2\right) + \frac{1}{10} & \text{si } 0 \leq x < \frac{\pi}{10}, \\ \left| \sin 2\pi\left(\frac{4x-3}{5}\right) + \frac{\pi}{100} \right| & \text{si } \frac{\pi}{10} \leq x \leq 1. \end{cases} \quad (1.44)$$

Utilizamos los algoritmos (1.13) y (1.14). Partiendo de una discretización de 257 nodos, bajamos tres niveles aplicando el operador decimación por valores puntuales. Posteriormente subimos tres niveles aplicando operadores predicción $\overline{\mathcal{I}\mathcal{P}}$ de grado 3 por valores puntuales, eliminando los coeficientes de escala. Obtenemos la Figura 1.8, donde representamos la función original, la aproximación, la diferencia entre ambas y los valores que obtenemos en el nivel más grosero. En (a) empleamos interpolación centrada excepto en las fronteras donde sí se han adaptado los *stencils* a fin de centrar la atención en las discontinuidades interiores.

En (b) mostramos la localización de los coeficientes de escala mayores que 10^{-4} . El eje de ordenadas representa el nivel k . Para la discontinuidad de salto, según (1.37), dado que $r = 3$ y $s = 2$ se verán afectados los coeficientes $d_l^1 = O([f])$, para $l = i - 1, i, i + 1$, donde suponemos que la discontinuidad se localiza en $[x_{i-1}^0, x_i^0]$. De ahí los tres coeficientes de escala mayores que 10^{-4} que aparecen en (b). Al igual ocurre en la discontinuidad esquina, aunque en este caso la magnitud de la diferencia es menor debido al producto por h_0 : $d_l^1 = O([f'])h_0$, para $l = i - 1, i, i + 1$.

Al introducir la técnica *ENO*, Figura 1.8 (c), desaparece el *fenómeno de Gibbs* y los intervalos de escasa aproximación se reducen a uno, el que contiene la discontinuidad. Lo mismo ocurre con *WENO* y *PPH*, Figuras (e) y (f) respectivamente .

Con la técnica *SR*, Figura 1.8 (d), se mejora la aproximación en la esquina, aunque no en el salto, ya que al trabajar con valores puntuales la celda que contiene el salto, aunque sea sospechosa, no se ha etiquetado como singular ($G_i^{\overline{LP}}(x)$ no cambia de signo en los extremos del intervalo).

1.3

Esquemas de multirresolución para medias en celda

Consideramos⁷ $\mathcal{F} = L^1([0, 1])$ y una sucesión de intervalos anidados en $[0, 1]$ dados por (1.10), $c_i^k = (x_{i-1}^k, x_i^k)$, para $i = 1, 2, \dots, J_k$. La discretización por medias en celda se define como

$$\mathcal{D}_k : L^1[0, 1] \rightarrow V^k, \quad \bar{f}_i^k = (\mathcal{D}_k f)_i = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} f(x) dx, \quad 1 \leq i \leq J_k,$$

donde $L^1[0, 1]$ es el espacio de funciones absolutamente integrables en $[0, 1]$ y V^k es el espacio de sucesiones con J_k componentes. Dado que:

$$\begin{aligned} (\mathcal{D}_{k-1} f)_i = \bar{f}_i^{k-1} &= \frac{1}{h_{k-1}} \int_{x_{i-1}^{k-1}}^{x_i^{k-1}} f(x) dx = \frac{1}{2h_k} \int_{x_{2i-2}^k}^{x_{2i}^k} f(x) dx \\ &= \frac{1}{2} \left(\frac{1}{h_k} \int_{x_{2i-2}^k}^{x_{2i-1}^k} f(x) dx + \frac{1}{h_k} \int_{x_{2i-1}^k}^{x_{2i}^k} f(x) dx \right) \\ &= \frac{1}{2} (\bar{f}_{2i-1}^k + \bar{f}_{2i}^k) = \frac{1}{2} ((\mathcal{D}_k f)_{2i-1} + (\mathcal{D}_k f)_{2i}), \end{aligned}$$

⁷Se puede definir de manera similar para $\mathcal{F} = L^1(\Omega)$, tomando un cubrimiento finito de Ω por celdas disjuntas, ver [16, 9].

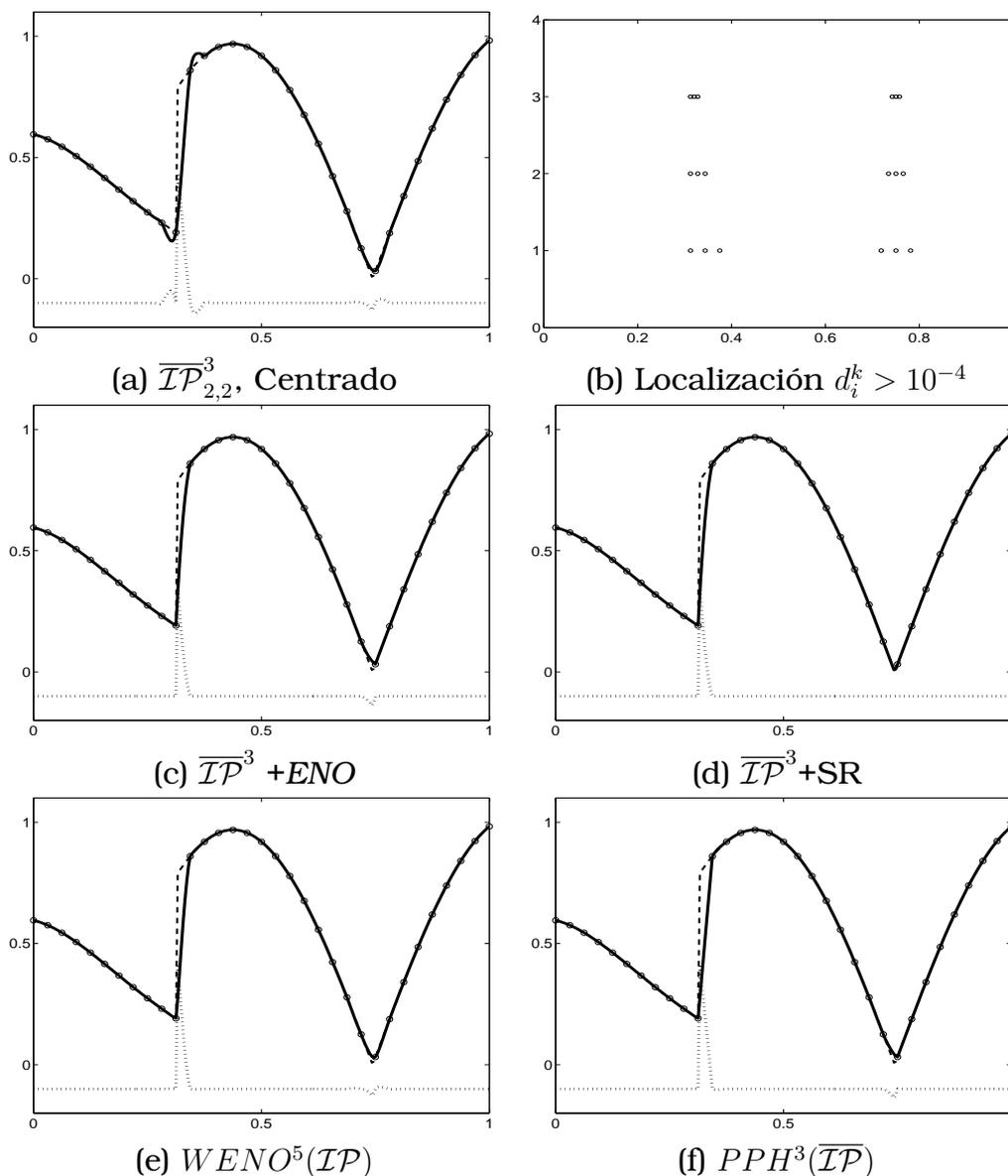


Figura 1.8: Aplicamos 3 niveles de decimación por valores puntuales y 3 niveles de predicción a la función (1.44) discretizada con 257 nodos. En (b) 'o' representa, para cada nivel, la localización de los coeficientes $d_i^k > 10^{-4}$. En el resto, '- -' representa la función de partida, '-.' la reconstrucción, 'o' la función en el nivel más grosero y '..' la diferencia entre la discretización inicial y la reconstrucción (desplazada $-0,1$ en el eje OY).

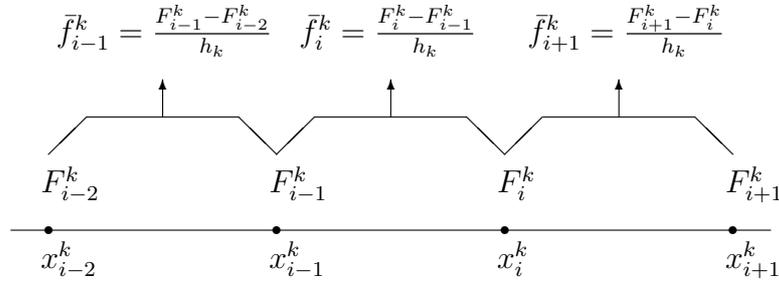


Figura 1.9: Correspondencia $\{\bar{f}_i^k\} \leftrightarrow \{F_i^k\}$

la discretización es anidada (Definición 1.7) y además el operador decimación toma la forma:

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

El error de predicción e^k satisface $0 = (D_k^{k-1} e^k)_i = (e_{2i-1}^k + e_{2i}^k)/2$. Entonces tenemos

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

proporcionando los operadores: $(G_k)_{i,j} = \delta_{2i-1,j}$, $(E_k)_{i,j} = \delta_{i,2j-1} - \delta_{i,2j}$.

A continuación describimos cómo construir la reconstrucción mediante la función primitiva, [16]. Definimos la secuencia $F_i^k = h_k \sum_{j=1}^i \bar{f}_j^k = \int_0^{x_i^k} f(y) dy$, con $F_i^k = F(x_i^k)$ y $F(x) = \int_0^x f(y) dy \in \mathcal{C}[0, 1]$ la primitiva de $f(x)$. $\{F_i^k\}_{i=0}^{J_k}$ corresponde a la discretización por valores puntuales de $F(x)$. Por tanto $\bar{f}_i^k = \frac{F_i^k - F_{i-1}^k}{h_k}$, estableciendo así una relación biunívoca entre los valores puntuales de $F(x)$ y las medias en celda $\bar{f}(x)$ (notar que $F_0^k = 0 \forall k$). Dicha relación puede apreciarse en la Figura 1.9.

Sea $\overline{\mathcal{I}\mathcal{P}}_k(x, F^k)$ una reconstrucción interpolatoria de $F(x)$ (en [16] se justifica dicha elección). Se define el operador reconstrucción de la siguiente manera: $(\mathcal{R}_k \bar{f}^k)(x) := \frac{d}{dx} \overline{\mathcal{I}\mathcal{P}}_k(x; F^k)$. Es fácil probar que $(\mathcal{R}_k \bar{f}^k)(x) \in L^1[0, 1]$ y que además es un operador reconstrucción (Definición 1.7) ya que:

$$((D_k \mathcal{R}_k \bar{f}^k)(x))_i = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} \frac{d}{dx} \overline{\mathcal{I}\mathcal{P}}_k(x; F^k) dx$$

42 1.3. Esquemas de multirresolución para medias en celda

$$= \frac{1}{h_k} (\overline{\mathcal{I}\mathcal{P}}_k(x_i^k; F^k) - \overline{\mathcal{I}\mathcal{P}}_k(x_{i-1}^k; F^k)) = \frac{F_i^k - F_{i-1}^k}{h_k} = \bar{f}_i^k.$$

El operador predicción se calcula como sigue:

$$\begin{aligned} (P_{k-1}^k f^{k-1})_{2i-1} &= (\mathcal{D}_k \mathcal{R}_{k-1} f^{k-1})_{2i-1} = \frac{1}{h_k} \int_{x_{2i-2}^k}^{x_{2i-1}^k} \frac{d}{dx} \overline{\mathcal{I}\mathcal{P}}_{k-1}(x; F^{k-1}) dx \\ &= \frac{1}{h_k} (\overline{\mathcal{I}\mathcal{P}}_{k-1}(x_{2i-1}^k; F^{k-1}) - \overline{\mathcal{I}\mathcal{P}}_{k-1}(x_{2i-2}^k; F^{k-1})) \\ &= \frac{1}{h_k} (\overline{\mathcal{I}\mathcal{P}}_{k-1}(x_{2i-1}^k; F^{k-1}) - F_{i-1}^{k-1}) \\ &\equiv \overline{\mathcal{I}\mathcal{C}}_{k-1}(x_{2i-1}^k; \bar{f}^{k-1}). \end{aligned} \quad (1.45)$$

Los esquemas (1.5) y (1.6) en este caso son:

Codificación:

$$\begin{cases} \text{Para } k = L, \dots, 1 \\ \bar{f}_i^{k-1} = \frac{1}{2}(\bar{f}_{2i-1}^k + \bar{f}_{2i}^k), & i = 1, \dots, J_{k-1}, \\ d_i^k = \bar{f}_{2i-1}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i-1}, & i = 1, \dots, J_{k-1}. \end{cases} \quad (1.46)$$

Decodificación:

$$\begin{cases} \text{Para } k = 1, \dots, L \\ \bar{f}_{2i-1}^k = (P_{k-1}^k \bar{f}^{k-1})_{2i-1} + d_i^k, & i = 1, \dots, J_{k-1}, \\ \bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k, & i = 1, \dots, J_{k-1}. \end{cases} \quad (1.47)$$

Se puede demostrar, [16], que hay una relación directa entre los coeficientes de escala $d^k(f)$ y los coeficientes de escala de la función primitiva $d^k(F)$:

$$d_i^k(f) = d_i^k(F)/h_k. \quad (1.48)$$

1.3.1

Técnicas de interpolación lineales y no lineales para medias en celda

Trabajaremos con la función F como describimos en la sección 1.3, aunque en la práctica sólo se emplea la función \bar{f} .

La función interpolatoria que utilizamos es la siguiente:

$$\overline{\mathcal{I}\mathcal{P}}_{k-1}^r(x, F^{k-1}) = q_i^{\overline{\mathcal{I}\mathcal{P}}}(x; F^{k-1}, r), \quad x \in [x_{i-1}^{k-1}, x_i^{k-1}], \quad 1 \leq i \leq J_{k-1}, \quad (1.49)$$

donde $q_i^{\overline{\mathcal{I}\mathcal{P}}}(x; F^{k-1}, r)$ representa el único polinomio de grado r que interpola a $F(x)$ en los puntos $\{x_i^{k-1}\}_i$. Alternativamente podemos expresar la función interpolatoria en función de las medias en celda:

$$\overline{\mathcal{I}\mathcal{C}}_{k-1}^{r-1}(x, \bar{f}^{k-1}) = q_i^{\overline{\mathcal{I}\mathcal{C}}}(x; \bar{f}^{k-1}, r-1) \equiv q_i^{\overline{\mathcal{I}\mathcal{P}}}(x; F^{k-1}, r) \equiv \overline{\mathcal{I}\mathcal{P}}_{k-1}^r(x, F^{k-1}).$$

Gracias a (1.45), por la Definición (1.49):

$$(P_{k-1}^k \bar{f}^{k-1})_{2i-1} = \frac{1}{h_k} \left[q_i^{\overline{\mathcal{I}\mathcal{P}}}(x_{2i-1}^k; F^{k-1}, r) - q_i^{\overline{\mathcal{I}\mathcal{P}}}(x_{2i-2}^k; F^{k-1}, r) \right].$$

NOTA 1.15. *Por inducción es fácil probar que [15]:*

$$\bar{f}[x_i^k, \dots, x_{i+n}^k] = (n+1)F[x_{i-1}^k, \dots, x_{i+n}^k]. \quad (1.50)$$

Utilizando la forma de Newton de cada pieza polinómica y la Nota 1.15 se puede escribir $(P_{k-1}^k \bar{f}^{k-1})_{2i-1}$ en términos de $\{\bar{f}_i^{k-1}\}$ y sus diferencias divididas por lo que F es únicamente una herramienta de diseño y en la práctica se trabaja con los valores de \bar{f} .

En el caso en que $\bar{f}(x) = P(x)$ donde $P(x)$ es un polinomio de grado menor o igual que $r-1 = 2s-2$, $F(x)$ es de grado menor o igual que $r = 2s-1$ y

$$\overline{\mathcal{I}\mathcal{P}}_{k-1}^r(x, F^{k-1}) = q_i^{\overline{\mathcal{I}\mathcal{P}}}(x; F^{k-1}, 2s-1) = F(x), \quad x \in [x_{i-1}^{k-1}, x_i^{k-1}].$$

Luego el espacio de polinomios de grado menor o igual a $2s-2$ satisface que $\mathcal{R}_{k-1} \mathcal{D}_{k-1} \bar{f} = \bar{f}$. Además para funciones suaves

$$\begin{aligned} \overline{\mathcal{I}\mathcal{P}}_{k-1}^r(x, F^{k-1}) &= F(x) + O(h_{k-1})^{2s}, \\ (\mathcal{R}_{k-1} \bar{f}^{k-1})(x) &= \frac{d}{dx} \overline{\mathcal{I}\mathcal{P}}_{k-1}^r(x, F^{k-1}) = \bar{f}(x) + O(h_{k-1})^{2s-1}. \end{aligned}$$

Por tanto, el orden de la reconstrucción es $2s-1 = r$ y podemos enunciar los siguientes resultados:

44 1.3. Esquemas de multirresolución para medias en celda

PROPOSICIÓN 1.4. Si $q_i^{\overline{IC}}(x; \bar{f}^k, r-1)$ es el polinomio \overline{IC} de grado $r-1$ (de grado r para $q_i^{\overline{IP}}(x; F^k, r)$), la reconstrucción \overline{IC} recupera de forma exacta polinomios de grado menor o igual a $r-1$.

COROLARIO 1.2. El orden de la reconstrucción \overline{IC} es r .

En regiones suaves, si las piezas polinómicas de $\overline{IP}_{k-1}(x, F^{k-1})$ son de grado r (grado de las piezas polinómicas de $\mathcal{R}_{k-1}(x, \bar{f}^{k-1})$ son de grado $r-1$) se cumplirá, en virtud de la propiedad (1.48):

$$d^k(\bar{f}) = \frac{d^k(F)}{h_k} = \frac{O(h_{k-1}^{r+1})}{h_k} = \frac{O(h_{k-1}^{r+1})}{2h_{k-1}} = O(h_{k-1}^r).$$

Si el *stencil* a partir del cual se construye la pieza polinómica cruza una discontinuidad de $\bar{f}^{(p)}$ ($p \leq r-1$): $d^k(\bar{f}) = O([\bar{f}^{(p)}]) h_{k-1}^p$.

En las proximidades de cada discontinuidad se producen una serie de intervalos de mala aproximación. Si para $\overline{IP}_{k-1}^r(x, F^{k-1})$ utilizamos interpolación centrada de grado r , con $r = 2s-1$, los coeficientes de escala afectados por una discontinuidad localizada en (x_{2i-1}^k, x_{2i}^k) serán

$$d_l^k(\bar{f}) = O([\bar{f}^{(p)}]) h_{k-1}^p, \quad l = i-s+1, \dots, i+s-1, \quad (1.51)$$

mientras que si la discontinuidad se halla entre (x_{2i-2}^k, x_{2i-1}^k) :

$$d_l^k(\bar{f}) = O([\bar{f}^{(p)}]) h_{k-1}^p, \quad l = i-s+1, \dots, i+s-2. \quad (1.52)$$

El uso de las técnicas ENO-SR maximiza el intervalo de buena aproximación. Para valores puntuales, vimos que SR mejoraba la aproximación para discontinuidades esquina pero no para saltos. Harten en [43] expone la siguiente observación:

**Una discontinuidad de salto en $\bar{f}(x)$
se convierte en una esquina en $F(x)$.**

$$\begin{array}{c}
\bar{f}[x_{i-2}, x_{i-1}, x_i] = 3F[x_{i-3}, x_{i-2}, x_{i-1}, x_i] \\
\leftarrow \text{-----} \text{-----} \text{-----} \rightarrow s_i = i - 2 \\
\bar{f}[x_{i-1}, x_i, x_{i+1}] = 3F[x_{i-2}, x_{i-1}, x_i, x_{i+1}] \\
\leftarrow \text{-----} \text{-----} \text{-----} \rightarrow s_i = i - 1 \\
\bar{f}[x_i, x_{i+1}, x_{i+2}] = 3F[x_{i-1}, x_i, x_{i+1}, x_{i+2}] \\
\leftarrow \text{-----} \text{-----} \text{-----} \rightarrow s_i = i \\
\text{-----} \\
x_{i-3} \quad x_{i-2} \quad x_{i-1} \quad x_i \quad x_{i+1} \quad x_{i+2}
\end{array}$$

Figura 1.10: Se representa en línea fina las posibles elecciones de stencils de 4 nodos ($r = 3$) para la función $F(x)$ el intervalo $I_i = [x_{i-1}, x_i]$ y en línea gris y más gruesa los nodos sobre los que la notación indica que se calculan las diferencias divididas para $\bar{f}(x)$ según (1.50).

Gracias a lo cual, si trabajamos con medias en celda de $\bar{f}(x)$ la técnica SR sí mejorará el caso ENO en los saltos, ya que visto desde la función $F(x)$, donde se trabaja con valores puntuales, cada salto será una esquina que sí será localizada por SR y por tanto también lo será para el salto de $\bar{f}(x)$.

Para hallar los stencils ENO de los polinomios $q_{s_i}^{\overline{TP}}(x; F^k, r)$ no es necesario calcular explícitamente $\{F^k\}$ ni sus diferencias divididas, sino que la elección del stencil se puede llevar a cabo directamente en términos de \bar{f}^k (ver Nota 1.15).

Los algoritmos ENO para el caso de medias en celda se deducen directamente a partir de los de valores puntuales para $F(x)$. Aunque la notación⁸ parezca indicar que $\bar{f}[x_i^k, \dots, x_{i+n}^k]$ se calcule sobre el stencil $\{x_i^k, \dots, x_{i+n}^k\}$ realmente se calcula sobre el stencil $\{x_{i-1}^k, \dots, x_{i+n}^k\}$. En dichos algoritmos el valor $(n + 1)$ de la relación (1.50) puede eliminarse, ya que en cada comparación la longitud de los stencils es la misma. Ver la Figura 1.10.

Los algoritmos ENO para el caso de medias en celda son el Algoritmo 1.3 y el Algoritmo 1.4.

⁸En la literatura sobre el tema suelen utilizarse conjuntos de nodos distintos para F y \bar{f} , como $\bar{x}_i = (1/2)(x_{i-1} + x_i)$. Dado que adoptar una u otra notación no afecta al diseño del método hemos optado por mantener el mismo conjunto de nodos.

46 1.3. Esquemas de multirresolución para medias en celda

Algoritmo 1.3 ENO-MC-I. Elección jerárquica del *stencil* para medias en celda

Entrada: $\{\bar{f}_i\}_{i=1}^J$, los datos iniciales; $r + 1$, tamaño del *stencil*

Salida: $\{s_i\}_{i=1}^{J-r+1}$

```
1: para  $i = 1, \dots, J - r + 1$  hacer
2:    $s_0 = i$ 
3:   para  $l = 0, \dots, r - 2$  hacer
4:     si  $|\bar{f}[x_{s_l-1}, \dots, x_{s_l+l}]| < |\bar{f}[x_{s_l}, \dots, x_{s_l+l+1}]|$  entonces
5:        $s_{l+1} = s_l - 1$ 
6:     si no
7:        $s_{l+1} = s_l$ 
8:     fin si
9:   fin para
10:   $s_i = s_{r-1}$ 
11: fin para
```

Algoritmo 1.4 ENO-MC-II. Elección no jerárquica del *stencil* para medias en celda

Entrada: $\{\bar{f}_i\}_{i=1}^J$, los datos iniciales; $r + 1$, tamaño del *stencil*

Salida: $\{s_i\}_{i=1}^{J-r+1}$

```
para  $i = 1, \dots, J - r + 1$  hacer
   $|\bar{f}[x_{s_i}, \dots, x_{s_i+r-1}]| = \min \{|\bar{f}[x_l, \dots, x_{l+r-1}]|, i - r + 1 \leq l \leq i\}$ 
fin para
```

1.3.2

Filtros para esquemas en medias en celda

Al indicar orden r nos referimos a que utilizamos $r + 1$ puntos para los *stencils* de F (polinomios de grado r) y por tanto r puntos para los *stencils* de \bar{f} (grado $r - 1$).

Caso lineal

Los esquemas (1.46) y (1.47) al sustituir la interpolación son (posteriormente lo comprobaremos):

Codificación:

$$\left\{ \begin{array}{l} \text{Para } k = L, \dots, 1 \\ \bar{f}_i^{k-1} = \frac{1}{2}(\bar{f}_{2i-1}^k + \bar{f}_{2i}^k), \quad i = 1, \dots, J_{k-1}, \\ d_i^k = \bar{f}_{2i-1}^k - \left(\bar{f}_i^{k-1} + \sum_{l=1}^{s-1} (\gamma_{s-1, s-1}^r)_l (\bar{f}_{i+l}^{k-1} - \bar{f}_{i-l}^{k-1}) \right), \quad i = 1, \dots, J_{k-1}. \end{array} \right.$$

Decodificación:

$$\left\{ \begin{array}{l} \text{Para } k = 1, \dots, L \\ \bar{f}_{2i-1}^k = d_i^k + \left(\bar{f}_i^{k-1} + \sum_{l=1}^{s-1} (\gamma_{s-1, s-1}^r)_l (\bar{f}_{i+l}^{k-1} - \bar{f}_{i-l}^{k-1}) \right), \quad i = 1, \dots, J_{k-1}, \\ \bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k, \quad i = 1, \dots, J_{k-1}. \end{array} \right.$$

donde $r = 2s - 1$, y

$$\left\{ \begin{array}{l} r = 3 \implies (\gamma_{1,1}^3)_1 = -\frac{1}{8}. \\ r = 5 \implies (\gamma_{2,2}^5)_1 = -\frac{22}{128}, (\beta_{2,2}^5)_2 = \frac{3}{128}. \end{array} \right.$$

Caso ENO

Fijado k , buscamos un polinomio de grado r , $q(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$, cuyas medias aproximen a las medias en celda $\{\bar{f}_i^{k-1}\}_i$. O dicho de otro modo, $q(x)$ debe interpolar a la función $F^{k-1}(x)$.

Al considerar las medias, a_1 será indeterminado y quedarán por calcular r coeficientes. Sin pérdida de generalidad y para facilitar la exposición, trabajaremos sobre nodos enteros:

$$\frac{q(x_{i+j}^{k-1}) - q(x_{i+j-1}^{k-1})}{h_{k-1}} = \frac{p(j+1) - p(j)}{1}, \text{ para } j = -nl, -nl+1, \dots, nr.$$

Imponiendo nl condiciones a la izquierda de \bar{f}_i^{k-1} y nr a la derecha ($r = nl + nr + 1$), las ecuaciones serán:

$$p(j+1) - p(j) = \sum_{l=2}^{r+1} a_l ((j+1)^{l-1} - j^{l-1}) = \bar{f}_{i+j}^{k-1}, \quad (1.53)$$

para $j = -nl, -nl+1, \dots, -1, 0, 1, 2, \dots, nr$.

48 1.3. Esquemas de multirresolución para medias en celda

En forma matricial el sistema será $A_{nl, nr}^{\overline{IC}} \hat{a} = \hat{f}$, donde:

$$A_{nl, nr}^{\overline{IC}} = \begin{bmatrix} (-nl+1)^1 - (-nl)^1 & \cdots & (-nl+1)^r - (-nl)^r \\ (-nl+2)^1 - (nl+1)^1 & \cdots & (-nl+2)^r - (-nl+1)^r \\ \vdots & \ddots & \vdots \\ 1 & \cdots & -(-1)^r \\ 1 & \cdots & 1^r \\ 1 & \cdots & 2^r - 1^r \\ \vdots & \ddots & \vdots \\ (nr+1)^1 - (nr)^1 & \cdots & (nr+1)^r - (nr)^r \end{bmatrix}, \quad (1.54)$$

$$\hat{a} = [a_2, \dots, a_{r+2}]^T \text{ y } \hat{f} = [\bar{f}_{i-nl}^{k-1}, \dots, \bar{f}_{i+nr}^{k-1}]^T.$$

Para obtener los coeficientes $\{a_2, \dots, a_{r+2}\}$ del polinomio $p(x)$ resolvemos el sistema:

$$A_{nl, nr}^{\overline{IC}} \hat{a} = \hat{f} \Rightarrow \hat{a} = \left(A_{nl, nr}^{\overline{IC}} \right)^{-1} \hat{f}. \quad (1.55)$$

NOTA 1.16. El coeficiente a_1 queda sin ser calculado, aunque esto no supone ningún problema, ya que como finalmente buscamos las medias del polinomio $q(x)$ el coeficiente a_1 se elimina.

Al polinomio $p(x)$ que hemos calculado lo denotamos por

$$p_{s_i}^{\overline{IP}}(x; F^{k-1}, r) \equiv p_{s_i}^{\overline{IC}}(x; \bar{f}^{k-1}, r-1).$$

Si los nodos no son sobre enteros sino sobre $\{x_i^k\}$ lo indicaremos con la notación: $q_{s_i}^{\overline{IP}}(x; F^{k-1}, r)$

Una vez hallados $\{a_l\}_{l=2}^{r+1}$ el valor que aproxima a \bar{f}_{2i-1}^k será:

$$\begin{aligned} \bar{f}_{2i-1}^k &= \frac{p_{s_i}^{\overline{IP}}(\frac{1}{2}; F^{k-1}, r) - p_{s_i}^{\overline{IP}}(0; F^{k-1}, r)}{\frac{1}{2}} \\ &= 2 \left(\sum_{l=1}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1} - a_1 \right) = \sum_{l=2}^{r+1} a_l \left(\frac{1}{2}\right)^{l-2} \\ &= \left(1, \frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T \hat{a} = \left(1, \frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T \left(A_{nl, nr}^{\overline{IC}} \right)^{-1} \hat{f} \end{aligned}$$

$$= \sum_{l=-nl}^{nr} (\gamma_{nl,nr}^r)_l \bar{f}_{i+l} \equiv \overline{\mathcal{IC}}_{nl,nr}^{r-1}(x_{2i-1}^k, \bar{f}^{k-1}). \quad (1.56)$$

siendo⁹ (ver Tabla 1.4):

$$(\gamma_{nl,nr}^r)_l = \left(1, \frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T \left(A_{nl,nr}^{\overline{\mathcal{IC}}}\right)^{-1} e_l, \quad l = -nl, \dots, nr,$$

con $e_j = (\delta_{j,k})_{k=-nl}^{nr}$ y $A_{nl,nr}^{\overline{\mathcal{IC}}}$ la definida en (1.54).

| l | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| $(\gamma_{2,0}^3)_l$ | | | $-\frac{1}{8}$ | $\frac{4}{8}$ | $\frac{5}{8}$ | | | | |
| $(\gamma_{1,1}^3)_l$ | | | | $\frac{1}{8}$ | $\frac{8}{8}$ | $-\frac{1}{8}$ | | | |
| $(\gamma_{0,2}^3)_l$ | | | | | $\frac{11}{8}$ | $-\frac{4}{8}$ | $\frac{1}{8}$ | | |
| $(\gamma_{4,0}^5)_l$ | $-\frac{7}{128}$ | $\frac{38}{128}$ | $-\frac{88}{128}$ | $\frac{122}{128}$ | $\frac{63}{128}$ | | | | |
| $(\gamma_{3,1}^5)_l$ | | $\frac{3}{128}$ | $-\frac{18}{128}$ | $\frac{52}{128}$ | $\frac{98}{128}$ | $-\frac{7}{128}$ | | | |
| $(\gamma_{2,2}^5)_l$ | | | $-\frac{3}{128}$ | $\frac{22}{128}$ | $\frac{128}{128}$ | $-\frac{22}{128}$ | $\frac{3}{128}$ | | |
| $(\gamma_{1,3}^5)_l$ | | | | $\frac{7}{128}$ | $\frac{158}{128}$ | $-\frac{52}{128}$ | $\frac{18}{128}$ | $-\frac{3}{128}$ | |
| $(\gamma_{0,4}^5)_l$ | | | | | $\frac{193}{128}$ | $-\frac{122}{128}$ | $\frac{88}{128}$ | $-\frac{38}{128}$ | $\frac{7}{128}$ |
| \bar{f}_{i+l}^{k-1} | \bar{f}_{i-4}^{k-1} | \bar{f}_{i-3}^{k-1} | \bar{f}_{i-2}^{k-1} | \bar{f}_{i-1}^{k-1} | \bar{f}_i^{k-1} | \bar{f}_{i+1}^{k-1} | \bar{f}_{i+2}^{k-1} | \bar{f}_{i+3}^{k-1} | \bar{f}_{i+4}^{k-1} |

Tabla 1.4: Aproximaciones a \bar{f}_{2i-1}^k , de orden r , $\overline{\mathcal{IC}}_{nl,nr}^{r-1}(x, \bar{f}^{k-1}) = \sum_{l=-nl}^{nr-1} (\gamma_{nl,nr}^r)_l \bar{f}_{i+l}^{k-1}$, utilizando interpolación por medias en celda con $nl + nr + 1 = r$ celdas, nl a la izquierda y nr a la derecha de \bar{f}_i^{k-1} .

1.3.3

WENO para medias en celda

El esquema WENO para medias en celda es similar al que expusimos para valores puntuales en la sección 1.2.4. En esencia, se basa en la utilización de una combinación convexa de todos los

⁹En <http://www.uv.es/anims/noguera/> se pueden descargar una GUI y programas que calculan dichos filtros según los datos introducidos por el usuario.

50 1.3. Esquemas de multirresolución para medias en celda

posibles *stencils* que contienen cada intervalo, asignándoles a cada uno de ellos un peso no lineal dependiente de la suavidad de la función. Con esto se consigue aumentar el orden de aproximación debido a la utilización de la información disponible en todos los *stencils* que contienen cada intervalo.

Veamos cómo se materializan estas ideas. Denotemos por $\bar{\mathcal{S}}_{i+k}$, $k = 0, \dots, r-1$, los r *stencils* conteniendo al intervalo $I_i = [x_{j-1}, x_j]$,

$$\bar{\mathcal{S}}_{i+k} = \{I_{i+k-r+1}, \dots, I_{i+k}\}, \quad k = 0, \dots, r-1, \quad (1.57)$$

y sea $\bar{\mathcal{I}}_{i+k-s+1}^{r-1}(x)$ el polinomio interpolador por medias en celda de orden $r = 2s - 1$ construido a partir del *stencil* $\bar{\mathcal{S}}_{i+k}$. Definimos el polinomio interpolador *WENO* en el intervalo I_i como

$$\bar{p}_i^{WENO-\bar{\mathcal{I}}}(x) = \sum_{k=0}^{r-1} \omega_k^i \bar{\mathcal{I}}_{i+k-s+1}^{r-1}(x), \quad (1.58)$$

donde $\omega_k^i \geq 0$, $k = 0, \dots, r-1$, y $\sum_{k=0}^{r-1} \omega_k^i = 1$.

Para cada intervalo, el interpolante *WENO* toma información de $2r - 1$ celdas, pudiéndose alcanzar un orden de exactitud de hasta $2r - 1$. Si denotamos por $\bar{\mathcal{I}}_{i+k-s+1}^{r-1}$ las aproximaciones obtenidas por interpolación para medias en celda con los *stencils* $\bar{\mathcal{S}}_{i+k}$ y denotamos por $\bar{\mathcal{I}}_{i+k-s+1}^{2r-2}$ la aproximación obtenida empleando el *stencil* $\mathcal{T} = \cup_{r-1}^{k=0} \bar{\mathcal{S}}_{i+k-r+1}$, es fácil demostrar que podemos expresar la aproximación de orden $2r - 1$ como una combinación lineal de las r aproximaciones de orden r :

$$\bar{\mathcal{I}}_{i+k-s+1}^{2r-2}(x) = \sum_{k=0}^{r-1} C_k^r \bar{\mathcal{I}}_{i+k-s+1}^{r-1}(x), \quad (1.59)$$

con $C_k^r \geq 0$, $\forall k$ y $\sum_{k=0}^{r-1} C_k^r = 1$. Las constantes C_k^r se denominan *pesos óptimos*. La Tabla 1.5 nos muestra los pesos óptimos para $r = 2, 3$. En [8, 14, 21] podemos encontrar los detalles sobre cómo obtener dichos pesos.

Las siguientes proposiciones, cuyas demostraciones podemos encontrar en [14, 21], nos ayudan a definir los pesos de la reconstrucción *WENO*:

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

Tabla 1.5: Pesos óptimos para $r = 2, 3$ en WENO para medias en celda.

PROPOSICIÓN 1.5. Si los pesos ω_k^i cumplen la relación:

$$\omega_k^i = C_k^i + O(h^m), \quad k = 0, \dots, r-1, \quad (1.60)$$

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

$$\sum_{l=0}^{r-1} \omega_l^i \overline{\mathcal{I}C}_{i+l-s+1}^{r-1} = \bar{f}_{2i-1} + O(h^{\min(r+m, 2r-1)}).$$

PROPOSICIÓN 1.6. Si el indicador de suavidad satisface la relación

$$\overline{\mathcal{I}S}_{i+l} = D(1 + O(h^m)), \quad l = 0, \dots, r-1, \quad (1.61)$$

con D una constante distinta de cero e independiente de l , aunque puede depender de h , entonces los pesos ω_j^i satisfacen la relación (1.60).

Con el objetivo de satisfacer las condiciones anteriores, se definen los pesos como sigue:

$$\omega_k^i = \frac{\alpha_k^i}{\sum_{s=0}^{r-1} \alpha_s^i}, \quad k = 0, \dots, r-1, \quad \text{con } \alpha_k^i = \frac{C_k^r}{(\epsilon + \overline{\mathcal{I}S}_{i+k})^2}. \quad (1.62)$$

Al igual que en valores puntuales, el valor ϵ introducido en (1.62) es una constante positiva para evitar que el denominador se anule, y $\overline{\mathcal{I}S}_{i+k}$ es un indicador de suavidad de $\bar{f}(x)$ en el stencil \bar{S}_{i+k} .

En [14, 21] se demuestra que los pesos así definidos verifican la relación (1.60). Para que se cumpla la condición ENO, es suficiente que los indicadores de suavidad verifiquen que $\overline{\mathcal{I}S}_{i+k} = O(h^2)$ en los

52 1.3. Esquemas de multirresolución para medias en celda

stencils en los que la función es suave y $\overline{\mathcal{I}\mathcal{S}}_{i+k} = O(1)$ en los *stencils* en los que la función presenta discontinuidades.

Liu et al. ([52]) proponen el siguiente *indicador de suavidad* basado en las diferencias no divididas de $f(x)$ en los nodos de interpolación,

$$\overline{\mathcal{I}\mathcal{S}}_{i+k} = \sum_{l=1}^{r-1} \left(\sum_{m=1}^l (\Delta^{r-l} [\bar{f}_{i+k+m-r}])^2 \right) / l, \quad (1.63)$$

donde $\Delta^s[f_l]$ no es más que la diferencia dividida de \bar{f} en los intervalos $\{I_l, \dots, I_{l+s}\}$. Se puede demostrar que este indicador de suavidad no siempre es óptimo, pudiéndose asegurar únicamente un orden de exactitud $r + 1$.

Jiang y Shu, definen en [52] un nuevo indicador de suavidad,

$$\overline{\mathcal{I}\mathcal{S}}_{i+k} = \sum_{l=1}^{r-1} \int_{x_{i-1}}^{x_i} h^{2l+1} (\bar{p}_{i+k}^{(l)}(x))^2 dx, \quad (1.64)$$

donde $\bar{p}_{i+k}^{(l)}$ es la derivada l -ésima del polinomio interpolador para medias en celda $\bar{p}_{j+k}(x)$ construido a partir del *stencil* \bar{S}_{i+k} .

En [14, 21] se demuestra que el indicador de Jiang y Shu cumplen la propiedad *ENO* y hace posible alcanzar órdenes óptimos tanto para el caso $r = 2$ como $r = 3$.

1.3.4

PPH para medias en celda

En este apartado adaptaremos la técnica *PPH* vista para valores puntuales al caso de medias en celda. Tenemos que

$$\overline{\mathcal{I}\mathcal{C}}_{1,1}^2(x_{2i-1}^k, f^{k-1}) = \bar{f}_i^{k-1} - \frac{1}{8}(\bar{f}_{i+1}^{k-1} - \bar{f}_{i-1}^{k-1}) = \bar{f}_i^{k-1} - \frac{1}{4} \frac{d^1 \bar{f}_{i-1}^{k-1} + d^1 \bar{f}_i^{k-1}}{2},$$

con $d^1 \bar{f}_i^{k-1} = \bar{f}_{i+1}^{k-1} - \bar{f}_i^{k-1}$. Sólo resta sustituir la media aritmética:

$$\left(\overline{\mathcal{I}\mathcal{C}}_{1,1}^2 \right)^{PPH} (x_{2i-1}^k, f^{k-1}) = \bar{f}_i^{k-1} - \frac{1}{4} H(d^1 \bar{f}_i^{k-1}, d^1 \bar{f}_{i-1}^{k-1}), \quad (1.65)$$

donde $H(x, y)$ es la media armónica definida en (1.30).

1.3.5

ENO-SR para medias en celda

Los pasos para aplicar *ENO-SR* para medias en celda son (ver Figura 1.11):

1.- En el nivel $k - 1$ calcular los *stencils ENO* mediante el Algoritmo 1.3 o 1.4, (*ENO-MC-I* o *ENO-MC-II*).

2.- Detectar las posibles *celdas singulares*: si $\bar{S}_{i-1}^{k-1} \cap \bar{S}_{i+1}^{k-1} = \emptyset$, etiquetar la celda como *sospechosa*.

3.- Para cada celda no-singular calculamos $(P_{k-1}^k f^{k-1})_{2i-1}$ mediante la relación (1.45).

4.- Para cada celda sospechosa definir la función

$$G_i^{\overline{IC}}(x) := q_{s_{i+1}}^{\overline{IP}}(x; F^{k-1}, r) - q_{s_{i-1}}^{\overline{IP}}(x; F^{k-1}, r), \quad (1.66)$$

donde $q_{s_{i+1}}^{\overline{IP}}(x; F^{k-1}, r)$ es el polinomio de grado r formado a partir del *stencil* \bar{S}_{i+1}^{ENO} (en este caso $\{x_i^{k-1}, \dots, x_{i+r}^{k-1}\}$) que interpola a F^{k-1} y $q_{s_{i-1}}^{\overline{IP}}(x; F^{k-1}, r)$ es el construido a partir del *stencil* \bar{S}_{i-1}^{ENO} (en este caso $\{x_{i-r-1}^{k-1}, \dots, x_{i-1}^{k-1}\}$) que interpola a F^{k-1} .

Si $G_i^{\overline{IC}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{IC}}(x_i^{k-1}) < 0 \Rightarrow$ existe una raíz θ_i^{k-1} de $G_i^{\overline{IC}}(x)$, y etiquetamos la celda como *singular*.

5.- Dada una celda singular $[x_{i-1}^{k-1}, x_i^{k-1}]$,

■ Si $G_i^{\overline{IC}}(x_{2i-1}^k) \cdot G_i^{\overline{IC}}(x_i^{k-1}) < 0 \rightarrow \theta_i^{k-1} > x_{2i-1}^k$ y calculamos

$$\bar{f}_{2i-1}^k = \frac{1}{h_k} (q_{s_{i-1}}^{\overline{IP}}(x_{2i-1}^k; F^{k-1}, r) - \overbrace{q_{s_{i-1}}^{\overline{IP}}(x_{2i-2}^k; F^{k-1}, r)}^{F_{i-1}^{k-1}}). \quad (1.67)$$

■ En otro caso, $\theta_i^{k-1} < x_{2i-1}^k$ y por tanto:

$$\bar{f}_{2i-1}^k = \frac{1}{h_k} (q_{s_{i+1}}^{\overline{IP}}(x_{2i-1}^k; F^{k-1}, r) - \overbrace{q_{s_{i-1}}^{\overline{IP}}(x_{2i-2}^k; F^{k-1}, r)}^{F_{i-1}^{k-1}}).$$

54 1.3. Esquemas de multirresolución para medias en celda

Para hallar los valores que faltan tenemos dos opciones: o bien extendemos adecuadamente tomando datos de la izquierda (o derecha) de la discontinuidad, o bien hacemos uso de la relación $\bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k$. Esta última opción es válida incluso para los datos obtenidos mediante SR. Por ejemplo, si x_{2i-1}^k se halla a la izquierda del salto, obtenemos:

$$(P_{k-1}^k \bar{f}^{k-1})_{2i-1} = \frac{1}{h_k} (q_{s_{i-1}}^{\overline{IP}}(x_{2i-1}^k; F^{k-1}, r) - \overbrace{q_{s_{i-1}}^{\overline{IP}}(x_{2i-2}^k; F^{k-1}, r)}^{F_{i-1}^{k-1}}),$$

$$(P_{k-1}^k \bar{f}^{k-1})_{2i} = \frac{1}{h_k} (\overbrace{q_{s_{i+1}}^{\overline{IP}}(x_{2i}^k; F^{k-1}, r)}^{F_i^{k-1}} - q_{s_{i-1}}^{\overline{IP}}(x_{2i-1}^k; F^{k-1}, r)).$$

Se cumple que

$$\frac{\bar{f}_{2i-1}^k + \bar{f}_{2i}^k}{2} = \frac{(P_{k-1}^k \bar{f}^{k-1})_{2i-1} + (P_{k-1}^k \bar{f}^{k-1})_{2i}}{2h_k} = \frac{F_i^{k-1} - F_{i-1}^{k-1}}{h_{k-1}} = \bar{f}_i^{k-1}.$$

Este razonamiento puede aplicarse de manera análoga si x_{2i-1}^k se halla a la derecha de la discontinuidad, con lo cual en los todas las medias, obtenidas incluso por SR, se cumple la relación $\bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k$.

En ambos casos $(P_{k-1}^k \bar{f}^{k-1})_{2i-1}$ se calcula utilizando información que no cruza la discontinuidad.

Filtros SR para medias en celda

Para calcular los filtros desarrollaremos en primer lugar las expresiones que se han expuesto al describir los pasos de ENO-SR. Para $G_i^{\overline{IC}}(x)$ utilizaremos el siguiente resultado:

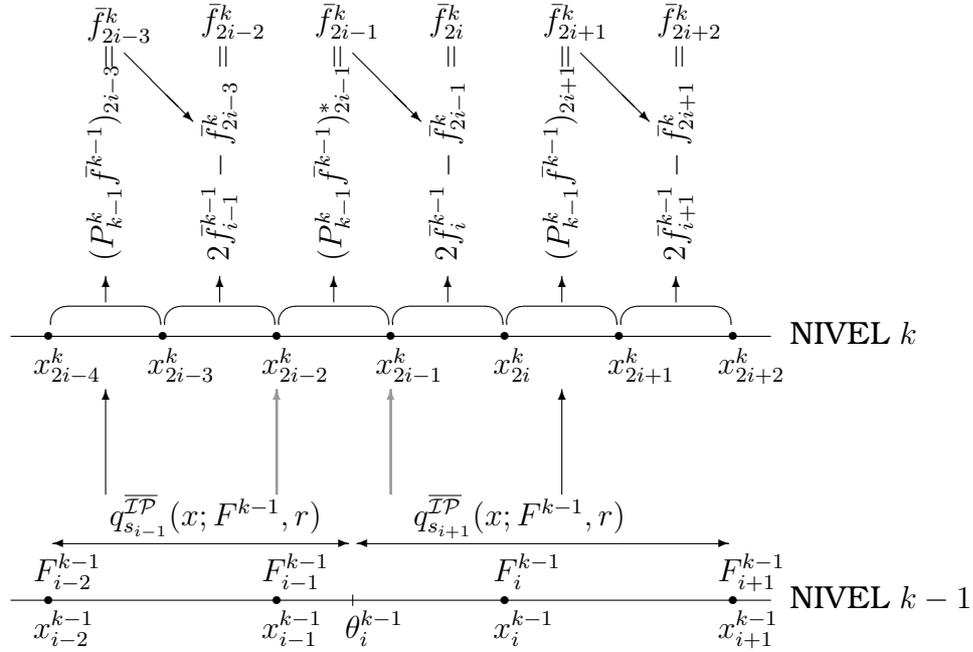


Figura 1.11: Paso del nivel $k - 1$ al k mediante multiresolución a partir de discretización por medias en celda utilizando ENO-SR. $(P_{k-1}^{f^{k-1}})_{2j-1} = \frac{1}{h_k} \left(q_{s_j}^{\overline{IP}}(x_{2j-1}^k; F^{k-1}, r) - F_{j-1}^{k-1} \right)$ según (1.45), excepto $(P_{k-1}^{f^{k-1}})_{2i-1}^*$ que en este caso es $\frac{1}{h_k} \left(q_{s_{i+1}}^{\overline{IP}}(x_{2i-1}^k; F^{k-1}, r) - F_{i-1}^{k-1} \right)$ según (1.67).

PROPOSICIÓN 1.7. Sea $A_{nl,nr}^{\overline{IC}}$ la matriz asociada al sistema que resuelve el polinomio interpolador para medias en celda, con los nodos expresados sobre enteros (definida en 1.54) y sea $(F_{i-1}^{k-1}, F_i^{k-1})$ la localización de la discontinuidad. La función $G_i^{\overline{IC}}$, para grado r , puede expresarse como:

$$G_i^{\overline{IC}}(x) = \bar{f}_i^{k-1} + [x - 1, \dots, x^r - 1] \left(A_{-1,r}^{\overline{IC}} \right)^{-1} [\bar{f}_{i+1}^{k-1}, \dots, \bar{f}_{i+r}^{k-1}]^T \\ - [x, \dots, x^r] \left(A_{r,-1}^{\overline{IC}} \right)^{-1} [\bar{f}_{i-r}^{k-1}, \dots, \bar{f}_{i-1}^{k-1}]^T.$$

56 1.3. Esquemas de multirresolución para medias en celda

Demostración: Trabajamos con las notaciones expuestas en 1.3.2 para el caso *ENO*. Sean

$$p_{s_{i+1}}^{\overline{IP}}(x; F^{k-1}, r) = \sum_{l=1}^{r+1} a_l x^{l-1} \quad \text{y} \quad p_{s_{i-1}}^{\overline{IP}}(x; F^{k-1}, r) = \sum_{l=1}^{r+1} b_l x^{l-1}.$$

Por definición del polinomio interpolador para medias en celda se cumple que $p_{s_{i+1}}^{\overline{IC}}(1; F^{k-1}, r) - p_{s_{i-1}}^{\overline{IC}}(0; F^{k-1}, r) = \bar{f}_i^{k-1}$. Por tanto:

$$\sum_{l=1}^{r+1} a_l - b_1 = \bar{f}_i^{k-1} \implies a_1 - b_1 = \bar{f}_i^{k-1} - \sum_{l=2}^{r+1} a_l. \quad (1.68)$$

Con esto:

$$\begin{aligned} G_i^{\overline{IC}}(x) &= q_{s_{i+1}}^{\overline{IP}}(x; F^{k-1}, r) - q_{s_{i-1}}^{\overline{IC}}(x; F^{k-1}, r) = \sum_{l=1}^{r+1} a_l x^{l-1} - \sum_{l=1}^{r+1} b_l x^{l-1} \\ &= a_1 - b_1 + \sum_{l=2}^{r+2} a_l x^{l-1} - \sum_{l=2}^{r+2} b_l x^{l-1} \\ &= \bar{f}_i^{k-1} - \sum_{l=2}^{r+1} a_l + \sum_{l=2}^{r+1} a_l x^{l-1} - \sum_{l=2}^{r+1} b_l x^{l-1} \\ &= \bar{f}_i^{k-1} + \sum_{l=2}^{r+1} a_l (x^{l-1} - 1) - \sum_{l=2}^{r+1} b_l x^{l-1} \\ &= \bar{f}_i^{k-1} + [x - 1, x^2 - 1, \dots, x^r - 1] \left(A_{-1,r}^{\overline{IC}} \right)^{-1} [\bar{f}_{i+1}^{k-1}, \dots, \bar{f}_{i+r}^{k-1}]^T \\ &\quad - [x, x^2, \dots, x^r] \left(A_{r,-1}^{\overline{IC}} \right)^{-1} [\bar{f}_{i-r}^{k-1}, \dots, \bar{f}_{i-1}^{k-1}]^T. \end{aligned}$$

□

Los valores que necesitamos para decidir si la celda es singular y para conocer la localización de x_{2i-1}^k respecto a la discontinuidad son $G_i^{\overline{IC}}(x_{i-1}^{k-1})$, $G_i^{\overline{IC}}(x_{2i-1}^k)$, $G_i^{\overline{IC}}(x_i^{k-1})$. Como trabajamos sobre nodos enteros debemos calcular:

$$\begin{aligned} G_i^{\overline{IC}}(x_{i-1}^{k-1}) &\rightarrow h_{k-1} G_i^{\overline{IC}}(0), \\ G_i^{\overline{IC}}(x_{2i-1}^k) &\rightarrow h_{k-1} G_i^{\overline{IC}}(1/2), \end{aligned}$$

$$G_i^{\overline{IC}}(x_i^{k-1}) \rightarrow h_{k-1} G_i^{\overline{IC}}(1).$$

Utilizando la Proposición 1.7:

■ $r = 3$

$$\left\{ \begin{array}{l} G_i^{\overline{IC}}(x_{i-1}^k) = h_{k-1} (\bar{f}_i^{k-1} - 3\bar{f}_{i+1}^{k-1} + 3\bar{f}_{i+2}^{k-1} - \bar{f}_{i+3}^{k-1}), \\ G_i^{\overline{IC}}(x_{2i-1}^k) = h_{k-1} \left(-\frac{5}{16}\bar{f}_{i-3}^{k-1} + \bar{f}_{i-2}^{k-1} - \frac{19}{16}\bar{f}_{i-1}^{k-1} \right. \\ \quad \left. + \bar{f}_i^{k-1} - \frac{19}{16}\bar{f}_{i+1}^{k-1} + \bar{f}_{i+2}^{k-1} - \frac{5}{16}\bar{f}_{i+3}^{k-1} \right), \\ G_i^{\overline{IC}}(x_i^k) = h_{k-1} (-\bar{f}_{i-3}^{k-1} + 3\bar{f}_{i-2}^{k-1} - 3\bar{f}_{i-1}^{k-1} + \bar{f}_i^{k-1}). \end{array} \right.$$

■ $r = 5$

$$\left\{ \begin{array}{l} G_i^{\overline{IC}}(x_{i-1}^k) = h_{k-1} (\bar{f}_i^{k-1} - 5\bar{f}_{i+1}^{k-1} + 10\bar{f}_{i+2}^{k-1} - 10\bar{f}_{i+3}^{k-1} \\ \quad + 5\bar{f}_{i+4}^{k-1} - \bar{f}_{i+5}^{k-1}), \\ G_i^{\overline{IC}}(x_{2i-1}^k) = h_{k-1} \left(-\frac{63}{256}\bar{f}_{i-5}^{k-1} + \frac{161}{128}\bar{f}_{i-4}^{k-1} - \frac{167}{64}\bar{f}_{i-3}^{k-1} \right. \\ \quad + \frac{359}{128}\bar{f}_{i-2}^{k-1} - \frac{437}{256}\bar{f}_{i-1}^{k-1} + \bar{f}_i^{k-1} - \frac{437}{256}\bar{f}_{i+1}^{k-1} \\ \quad \left. + \frac{359}{128}\bar{f}_{i+2}^{k-1} - \frac{167}{64}\bar{f}_{i+3}^{k-1} + \frac{161}{128}\bar{f}_{i+4}^{k-1} - \frac{63}{256}\bar{f}_{i+5}^{k-1} \right), \\ G_i^{\overline{IC}}(x_i^k) = h_{k-1} (-\bar{f}_{i-5}^{k-1} + 5\bar{f}_{i-4}^{k-1} - 10\bar{f}_{i-3}^{k-1} \\ \quad + 10\bar{f}_{i-2}^{k-1} - 5\bar{f}_{i-1}^{k-1} + \bar{f}_i^{k-1}). \end{array} \right.$$

NOTA 1.17. En las expresiones sobre la función G para medias en celda, el término h_k puede ser eliminado en la práctica, ya que de éstas sólo nos interesa su signo. Evitamos así trabajar con los nodos iniciales.

Dada una celda singular $[x_{i-1}^{k-1}, x_i^{k-1}]$, si $G_i^{\overline{IC}}(x_{2i-1}^k) \cdot G_i^{\overline{IC}}(x_i^{k-1}) < 0 \rightarrow \theta_i^{k-1} > x_{2i-1}^k$

$$(P_{k-1}^k \bar{f}^{k-1})_{2i-1} = \frac{p_{s_{i-1}}^{\overline{IP}}(\frac{1}{2}; F^{k-1}, r) - p_{s_{i-1}}^{\overline{IP}}(0; F^{k-1}, r)}{1/2} = 2 \sum_{l=2}^{r+1} b_l \left(\frac{1}{2}\right)^{l-1}$$

58 1.3. Esquemas de multirresolución para medias en celda

$$\begin{aligned}
 &= 2 \left(\frac{1}{2}, \dots, \frac{1}{2^r} \right) \left(A_{r,-1}^{\overline{IC}} \right)^{-1} \left[\bar{f}_{i-r}^{k-1}, \dots, \bar{f}_{i-1}^{k-1} \right]^T \\
 &= \sum_{l=-r}^{-1} (\gamma_{r,-1}^r)_l \bar{f}_{i+l}^{k-1}. \tag{1.69}
 \end{aligned}$$

donde $\gamma_{r,-1}^r$ pueden verse en la Tabla 1.6 para $r = 3, 5$.

| l | -5 | -4 | -3 | -2 | -1 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| $(\gamma_{3,-1}^3)$ | | | $\frac{5}{8}$ | -2 | $\frac{19}{8}$ |
| $(\gamma_{5,-1}^5)$ | $\frac{63}{128}$ | $-\frac{322}{128}$ | $\frac{668}{128}$ | $-\frac{718}{128}$ | $\frac{437}{128}$ |
| \bar{f}_{i+l}^{k-1} | \bar{f}_{i-5}^{k-1} | \bar{f}_{i-4}^{k-1} | \bar{f}_{i-3}^{k-1} | \bar{f}_{i-2}^{k-1} | \bar{f}_{i-1}^{k-1} |

Tabla 1.6: Aproximaciones a \bar{f}_{2i-1}^{k-1} de orden $r = nr + nl + 1$, $\overline{IC}_{r,-1}^{r-1}(x, \bar{f}^{k-1}) = \sum_{l=-r}^{-1} (\gamma_{r,-1}^r)_l \bar{f}_{i+l}^{k-1}$, utilizando extrapolación con r puntos a la izquierda de \bar{f}_i^{k-1} .

En otro caso, $\theta_i^{k-1} < x_{2i-1}^k$, y de manera análoga a la demostración de la Proposición 1.7:

$$\begin{aligned}
 \bar{f}_{2i-1}^k &= \frac{p_{s_{i+1}}^{\overline{IP}}(1/2; F^{k-1}, r) - p_{s_{i-1}}^{\overline{IP}}(0; F^{k-1}, r)}{1/2} \\
 &= \frac{\sum_{l=1}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1} - b_1}{1/2} = 2(a_1 - b_1 + \sum_{l=2}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1}) \\
 &= 2(\bar{f}_i^{k-1} + \sum_{l=2}^{r+1} a_l \left(\left(\frac{1}{2}\right)^{l-1} - 1\right)) \\
 &= 2\bar{f}_i^{k-1} + 2 \left[\frac{1}{2} - 1, \dots, \frac{1}{2^r} - 1 \right] \left(A_{-1,r}^{\overline{IC}} \right)^{-1} \left[\bar{f}_{i+1}^{k-1}, \dots, \bar{f}_{i+r}^{k-1} \right]^T.
 \end{aligned}$$

Dado que sabemos que se cumple la relación $\bar{f}_{2i-1}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i}^k$, también podemos calcular primero \bar{f}_{2i}^k y posteriormente despejar. De este modo podemos aprovechar el cálculo de filtros de la Tabla 1.6 :

$$\left(P_{k-1}^k \bar{f}^{k-1} \right)_{2i-1} = 2\bar{f}_i^{k-1} - \bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \sum_{l=1}^r (\gamma_{-1,r}^r)_l \bar{f}_{i+l}^{k-1},$$

donde $\gamma_{-1,r}$ son los simétricos de $\gamma_{r,-1}$:

$$(\gamma_{-1,r})_{-l} = (\gamma_{r,-1})_l, \quad l = -r, \dots, -1.$$

Para $r = 3, 5$, los filtros resultantes son:

- $r = 3$

$$(P_{k-1}^k \bar{f}^{k-1})_{2i-1} = 2\bar{f}_i^{k-1} - \frac{19}{8}\bar{f}_{i+1}^{k-1} + 2\bar{f}_{i+2}^{k-1} - \frac{5}{8}\bar{f}_{i+3}^{k-1}.$$

- $r = 5$

$$(P_{k-1}^k \bar{f}^{k-1})_{2i-1} = 2\bar{f}_i^{k-1} - \frac{437}{128}\bar{f}_{i+1}^{k-1} + \frac{359}{64}\bar{f}_{i+2}^{k-1} - \frac{167}{32}\bar{f}_{i+3}^{k-1} + \frac{161}{64}\bar{f}_{i+4}^{k-1} - \frac{63}{128}\bar{f}_{i+5}^{k-1}.$$

1.3.6

Ejemplos

Realizamos el mismo experimento que en el apartado 1.2.7 pero en este caso los datos iniciales corresponden a medias en celda. En concreto, discretizamos la siguiente función en los nodos centrales de n celdas equiespaciadas :

$$\bar{f}(x) = \begin{cases} -\frac{4x-3}{5} \sin\left(\frac{3}{2}\pi\left(\frac{4x-3}{5}\right)^2\right) & \text{si } 0 \leq x < \frac{3\pi}{29}, \\ \left| \sin 2\pi\left(\frac{4x-3}{5}\right) + \frac{\pi}{1000} \right| & \text{si } \frac{3\pi}{29} \leq x \leq 1. \end{cases} \quad (1.70)$$

Utilizamos los algoritmos (1.46) y (1.47) con los coeficientes de escala anulados. Bajamos tres niveles por medias en celda ($\bar{f}_i^{k-1} = (\bar{f}_{2i}^k + \bar{f}_{2i-1}^k)/2$) y posteriormente intentamos recuperar los datos del nivel de partida mediante distintas reconstrucciones. En la Figura 1.12 (a) empleamos interpolación centrada con polinomios de grado 3 en $F(x)$ (grado 2 para $\bar{f}(x)$) excepto en las fronteras donde sí adaptamos los *stencils*. Representamos la función original, la aproximación, la diferencia entre ambas y los puntos del nivel grosero. En la Figura (c) representamos los coeficientes de escala mayores que 10^{-3} para cada nivel. Observamos que hay tres intervalos afectados para la discontinuidad de salto. Por intervalos afectados entendemos tres d_i^k afectados. Así cada d_i^k de magnitud

elevada, se traduce en 2 puntos, \bar{f}_{2i-1}^k y \bar{f}_{2i}^k donde la aproximación empeora. Esto concuerda con (1.51), ya que en este caso la interpolación para F es de grado $r = 3$ con lo que $s = 2$ y por tanto si la discontinuidad se halla en (x_{2i-1}^k, x_{2i}^k) obtenemos tres coeficientes de escala afectados: $d_l^1(\bar{f}) = O([\bar{f}^{(p)}])h_0^p$, para $i - 1 \leq l \leq i + 1$. La magnitud de los coeficientes afectados para el salto es mayor que para la esquina, ya que en este caso $p = 0$, mientras que para la esquina $p = 1$. De hecho en la Figura 1.12 (b) sólo aparecen dos coeficientes de escala afectados en la discontinuidad de salto del primer nivel, ya que el tercero no es mayor que 10^{-3} .

Al introducir las técnicas *ENO* y *WENO*, Figura 1.12 (c) y (e) respectivamente, los intervalos de escasa aproximación se reducen a uno, el que contiene la discontinuidad. Finalmente con la técnica *SR*, Figura 1.12 (d), se mejora la aproximación tanto en la esquina como en el salto.

1.4

Transformada *wavelet*

En los últimos años, la transformada *wavelet* ha sido estudiada ampliamente como una herramienta para la compresión¹⁰ y eliminación de ruido en imágenes digitales. La primera mención de lo que hoy conocemos como *wavelet*¹¹ fue en 1909 en la tesis de Alfred Haar [42] en la que propuso que cualquier función continua podía ser aproximada por un desarrollo de funciones ortogonales. En los ochenta se retomaron estas ideas por un equipo dirigido por A. Grossman que contaba con la colaboración de los investigadores J. Morlet, Y. Meyer, S. Mallat, G. Battle y P. Lemarié entre otros. Ellos establecieron el concepto de *wavelet* en su forma teórica actual, siendo Y. Meyer [59, 58] y S. Mallat [54, 55] los principales desarrolladores de los métodos de análisis basados en *wavelets*. En 1988 los trabajos realizados por I. Daubechies [31, 32] popularizaron estas ideas como herramientas para

¹⁰Por ejemplo *JPEG-2000* se basa en *wavelets*.

¹¹En algunos textos en español se utiliza el término *ondícula* como traducción de *wavelet*.

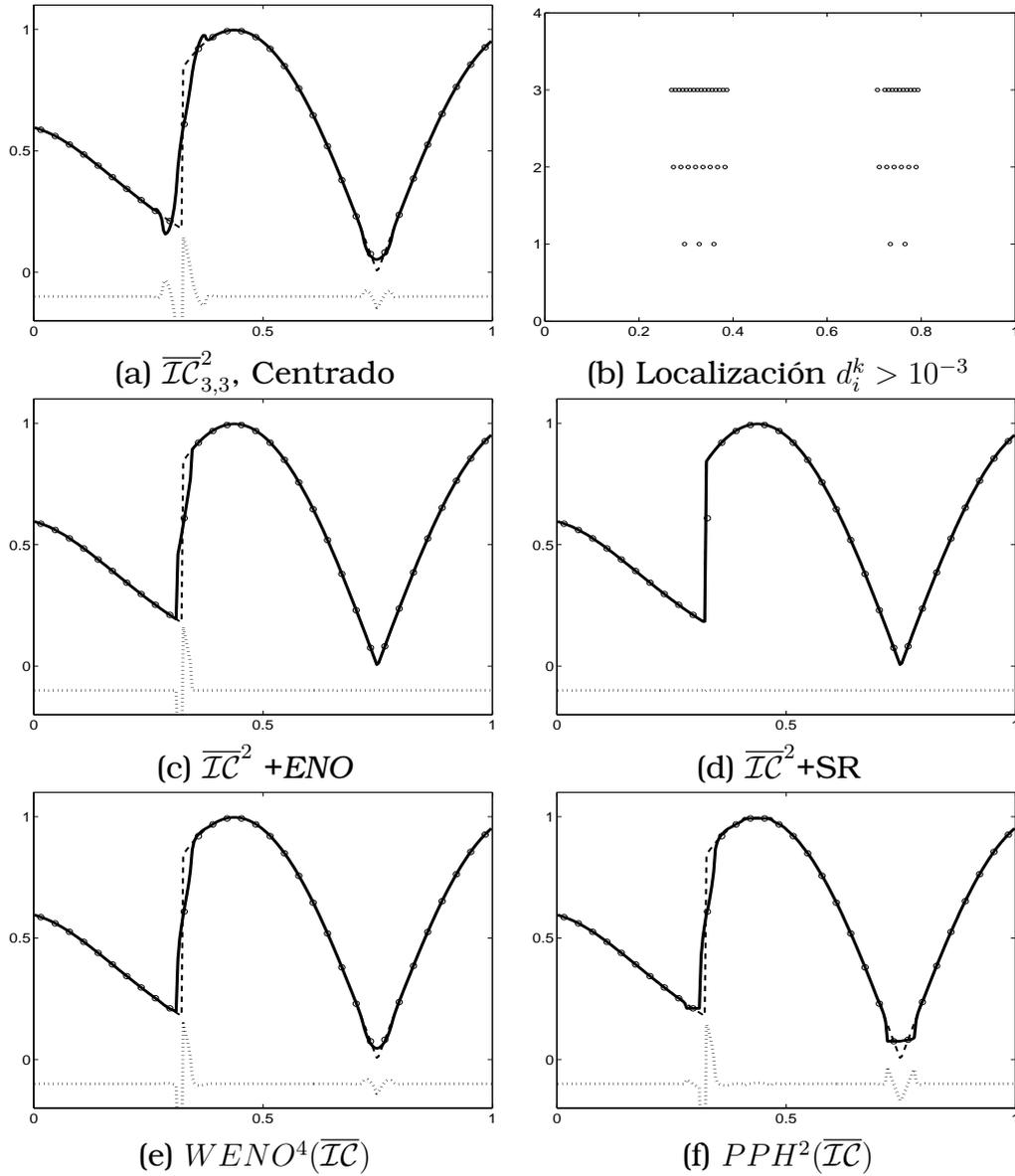


Figura 1.12: Aplicamos 3 niveles de decimación por medias en celda y 3 niveles de predicción a la función (1.70) discretizada con 256 nodos. En (b) 'o' representa, para cada nivel, la localización de los coeficientes $d_i^k > 10^{-3}$. En el resto, '-' representa la función de partida, '-.' la reconstrucción, 'o' la función en el nivel más grosero y '..' la diferencia entre la discretización inicial y la reconstrucción (desplazada $-0,1$ en el eje OY).

el procesado de señales, compresión de datos y el análisis numérico. En [49] también podemos encontrar una buena introducción a dicho concepto.

En esta sección veremos un breve resumen de la transformada *wavelet*, con el objetivo principal de fijar la notación. Introduciremos la transformada tanto en forma continua como en forma discreta, ya que, aunque trabajaremos en forma discreta, la forma continua es preferible para demostraciones teóricas.

Aunque aquí no lo trataremos, se puede demostrar la conexión existente entre cierto tipo de reconstrucciones multiescala y la transformada *wavelet*. En concreto, las descomposiciones multiescala dadas por valores puntuales y por medias en celda pueden considerarse un caso particular de transformadas *wavelet*, [16, 44, 22, 48, 10, 17].

1.4.1

Transformada *wavelet* continua

La transformada *wavelet* equivale a una descomposición en subespacios del espacio funcional donde nos encontremos. Una función cualquiera se descompondrá de forma única como suma de funciones que serán la proyección de la función inicial en dichos subespacios. La idea es descomponer la función original en una señal sin atributos, quedándonos sólo con lo esencial, y otras que corresponderán a los detalles.

Para realizar esta descomposición se utilizan las funciones *wavelet*, que llevan aparejadas otras funciones llamadas *scaling*. Estos pares *wavelet-scaling* se pueden interpretar como una pareja de filtros, uno de paso alto (altas frecuencias) y el otro de paso bajo (bajas frecuencias). Mientras que los *wavelets* proporcionan los detalles de una señal en una determinada escala, el *scaling* muestra versiones suavizadas de la señal en la misma escala.

Llamamos $\phi(x)$ a la función *scaling* y $\psi(x)$ al correspondiente *wavelet*, con soporte finito $[0, l]$ y l un entero positivo.

La función $\phi(x)$ cumple la llamada ecuación de dilatación:

$$\phi(x) = \sqrt{2} \sum_0^l c_s \phi(2x - s), \quad (1.71)$$

y $\psi(x)$ satisface la ecuación *wavelet*:

$$\psi(x) = \sqrt{2} \sum_0^l h_s \phi(2x - s), \quad (1.72)$$

donde c_s y h_s son constantes llamadas coeficientes del filtro de paso bajo y paso alto respectivamente.

Supondremos que $\psi(x)$ tiene p momentos nulos, es decir:

$$\int \psi(x) x^j dx = 0, \quad \text{con } j = 0, 1, \dots, p-1. \quad (1.73)$$

Consideramos las siguientes notaciones:

$$\phi_{j,i} = 2^{\frac{j}{2}} \phi(2^j x - i) \quad \text{y} \quad \psi_{j,i} = 2^{\frac{j}{2}} \psi(2^j x - i). \quad (1.74)$$

Tomamos $V_j, W_j \in L^2$ definidos como:

$$V_j = \langle \phi_{j,i}(x), i \in Z \rangle \quad \text{y} \quad W_j = \langle \psi_{j,i}(x), i \in Z \rangle.$$

Se cumple que $V_j = V_{j+1} \oplus W_{j+1}$, es decir, W_{j+1} es el complemento de V_{j+1} en V_j . Los subespacios $\{V_j : j \in Z\}$ forman una multirresolución de L^2 , que permite realizar una descomposición ortonormal como sigue:

$$L^2 = V_J \oplus \sum_{j=J}^{\infty} W_j, \quad (1.75)$$

pudiéndose entonces descomponer una función proyectándola en cada subespacio V_j :

$$f_j(x) = \sum_i \alpha_{j,i} \phi_{j,i}(x), \quad \text{donde } \alpha_{j,i} = \int f(x) \phi_{j,i}(x) dx, \quad i \in Z,$$

que corresponden a los coeficientes *wavelet* de bajas frecuencias (también llamados coeficientes *scaling* o *tendencias*). Análogamente, proyectando $f(x)$ en W_j obtenemos:

$$w_j(x) = \sum_i \beta_{j,i} \psi_{j,i}(x), \quad \text{donde } \beta_{j,i} = \int f(x) \psi_{j,i}(x) dx, \quad i \in Z,$$

que son los coeficientes *wavelet* de altas frecuencias (llamados coeficientes *wavelet* o *fluctuaciones*). Así, podemos descomponer la función $f(x)$ de manera única como:

$$f(x) = f_j(x) + \sum_{k=j}^{\infty} w_k(x). \quad (1.76)$$

La proyección $f_j(x)$ es la aproximación lineal de la función $f(x)$ en el subespacio V_j y se puede interpretar como la señal original sin detalles. Las funciones w_k contendrán los detalles de la función original.

NOTA 1.18. *En la práctica se trabaja con unos límites en las escalas: una más grosera, donde 2^{-j} coincide o es mayor que la longitud del intervalo donde estamos analizando la función y una más fina, a partir de la cual sobrepasamos la precisión de los datos analizados. Así:*

$$V_j = W_{j+1} \oplus W_{j+2} \oplus \cdots \oplus W_{j+l} \oplus V_{j+l},$$

$$f(t) = w_{j+1}(t) \oplus w_{j+2}(t) \oplus \cdots \oplus w_{j+l}(t) \oplus f_{j+l}(t),$$

donde $w_i(t) \in W_i$, $f_i \in V_i$, j es la escala más fina y $j+l$ la más grosera.

Para el cálculo de los coeficientes de altas y bajas frecuencias en la transformada directa, se utilizan las siguientes fórmulas, que pueden ser deducidas de (1.74), conocidas como la transformada *wavelet* rápida:

$$\alpha_{j,i} = \sum_{s=0}^l c_s \alpha_{j+1,2i+s} \quad \text{y} \quad \beta_{j,i} = \sum_{s=0}^l h_s \alpha_{j+1,2i+s}. \quad (1.77)$$

Error de aproximación

Enunciamos un resultado, cuya demostración puede consultarse en [68], acerca del error cometido cuando realizamos una aproximación lineal, es decir, cuando eliminamos los coeficientes de altas frecuencias.

TEOREMA 1.1. *Sea el wavelet $\psi(x)$ generado por la función de escala $\phi(x)$, con p momentos nulos. Sea $f_j(x)$ la aproximación lineal de $f(x)$ proyectada en V_j , con base $\phi_{j,k}(x)$, la cual posee derivadas de orden p acotadas, entonces:*

$$\|f(x) - f_j(x)\| \leq C(\Delta x)^p \|f^{(p)}(x)\|$$

donde $\Delta x = 2^{-j}$ y C es una constante independiente de j .

Además, si la función de escala y su derivada tienen soporte finito, el teorema es válido para la norma de L^∞ . El teorema nos dice que el error de aproximación depende de dos factores, por una parte del paso espacial y por otra de la norma de la derivada p de la función. Por tanto al realizar una aproximación lineal en funciones con saltos, $\|f^{(p)}(x)\|$ será grande y la aproximación pobre, por mucho que Δx sea pequeño.

1.4.2

Transformada *wavelet* discreta

Cuando trabajamos con datos discretos utilizamos la versión discreta de la transformada *wavelet*. En este caso, la transformada es un operador lineal que actúa sobre un vector (de longitud potencia entera de 2), transformándolo en otro vector de la misma longitud. Dicha transformación es invertible y ortogonal. Partiremos de un conjunto discreto de números que son los coeficientes de las bajas frecuencias de la función $f(x)$ en el subespacio V_{j+1} . En muchas aplicaciones este conjunto de números es simplemente una discretización de la función.

En este contexto discreto, definimos los coeficientes de los filtros *wavelet* mediante matrices:

$$L = \begin{pmatrix} c_0 & c_1 & \cdots & c_l & & & \\ & & c_0 & c_1 & \cdots & c_l & \\ & & & & \cdots & \cdots & \cdots \\ & & & & & c_0 & c_1 & \cdots & c_l \end{pmatrix},$$

$$H = \begin{pmatrix} h_0 & h_1 & \cdots & h_l & & & & & \\ & & h_0 & h_1 & \cdots & h_l & & & \\ & & & & \cdots & \cdots & \cdots & & \\ & & & & & h_0 & h_1 & \cdots & h_l \end{pmatrix}.$$

Dichas matrices deben cumplir la condición de ortogonalidad, es decir:

$$L^T L + H^T H = I, \quad (1.78)$$

aunque cada matriz por separado no sea ortogonal.

Llamando:

$$\vec{\alpha}_j = (\cdots, \alpha_{j,i}, \alpha_{j,i+1}, \cdots)^T \text{ y } \vec{\beta}_j = (\cdots, \beta_{j,i}, \beta_{j,i+1}, \cdots)^T,$$

las ecuaciones (1.77) pueden ser escritas en forma matricial:

$$\vec{\alpha}_j = L\vec{\alpha}_{j+1}, \quad \vec{\beta}_j = H\vec{\alpha}_{j+1}.$$

Estas ecuaciones resumen la transformada *wavelet* directa.

Gracias a la ortogonalidad, (1.78), la transformada *wavelet* inversa es:

$$\vec{\alpha}_{j+i} = L^T \vec{\alpha}_j + H^T \vec{\beta}_j, \quad (1.79)$$

es decir:

$$\alpha_{j+1,2i-1} = \sum_{k=0}^{l-1} (r_{2k} \alpha_{j,i-k} + d_{2k} \beta_{j,i-k}), \quad (1.80)$$

$$\alpha_{j+1,2i} = \sum_{k=0}^{l-1} (r_{2k+1} \alpha_{j,i-k} + d_{2k+1} \beta_{j,i-k}), \quad (1.81)$$

donde $r_{2k} = c_{2k+1}$, $d_{2k} = h_{2k+1}$, $r_{2k+1} = c_{2k}$, $d_{2k+1} = h_{2k}$, con $k = 0, 1, 2, \dots$

La aproximación lineal de la función consiste en reconstruir $\vec{\alpha}_{j+1}$ eliminando el término $\vec{\beta}_j$, es decir: $\vec{\alpha}_{j+i} = L^T \vec{\alpha}_j$.

Si la función es suave la eliminación de los términos $\vec{\beta}_j$ no entraña graves consecuencias y es ahí donde reside la capacidad de compresión de los métodos *wavelet*. Sin embargo, en las discontinuidades se generan valores de $\beta_{j,i}$ de mayor magnitud, que si se eliminan sí provocan pérdidas de información sustanciales y por tanto malas aproximaciones cerca de los saltos (*fenómeno de Gibbs*).

En lo que sigue se describirá la transformada *wavelet DB4*, presentada por Daubechies, ya que es la que utilizaremos en nuestros experimentos.

Daubechies 4

En la transformada *DB4*, se utilizan 4 coeficientes:

$$c_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, c_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, c_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, c_3 = \frac{1-\sqrt{3}}{4\sqrt{2}},$$

$$h_0 = c_3, h_1 = -c_2, h_2 = c_1, h_3 = -c_0.$$

Dichos coeficientes se hallan imponiendo la condición de ortogonalidad y resolviendo las ecuaciones que obtenemos, [56]. Las matrices de la transformada tienen la forma:

$$L = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & \\ & & & & \dots & \dots & \dots & \\ & & & & & & c_0 & c_1 & c_2 & c_3 \end{pmatrix},$$

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & \\ & & & & \dots & \dots & \dots & \\ & & & & & & h_0 & h_1 & h_2 & h_3 \end{pmatrix}.$$

Si realizamos la operación $L^T L + H^T H$ con matrices 3×8 , obtenemos:

$$H = \begin{pmatrix} 0,250 & -0,433 & & & & & & & & \\ -0,433 & 0,750 & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 0,750 & 0,433 & & \\ & & & & & & 0,433 & 0,250 & & \end{pmatrix}.$$

El resultado no es exactamente la identidad debido a las condiciones frontera. Si deseamos obtener la identidad, debemos imponer

condiciones frontera periódicas, con lo cual las matrices serían de la forma:

$$L = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & \\ & & & & \dots & \dots & \dots & \\ & & & & & c_0 & c_1 & c_2 & c_3 \\ c_2 & c_3 & & & & & & c_0 & c_1 \end{pmatrix},$$

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & \\ & & & & \dots & \dots & \dots & \\ & & & & & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & & & & & & h_0 & h_1 \end{pmatrix}.$$

Con esto sí se obtiene $L^T L + H^T H = I$.

La transformada directa será $\vec{\alpha}_j = L\vec{\alpha}_{j+1}$, $\vec{\beta}_j = H\vec{\alpha}_{j+1}$.

Para ver qué forma tiene esta transformación, la desarrollaremos suponiendo que tenemos 8 datos iniciales.

$$\begin{pmatrix} c_0 & c_1 & c_2 & c_3 & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & c_0 & c_1 & c_2 & c_3 \end{pmatrix} \begin{pmatrix} \alpha_{j+1,0} \\ \alpha_{j+1,1} \\ \alpha_{j+1,2} \\ \alpha_{j+1,3} \\ \alpha_{j+1,4} \\ \alpha_{j+1,5} \\ \alpha_{j+1,6} \\ \alpha_{j+1,7} \end{pmatrix} = \begin{pmatrix} \alpha_{j,0} \\ \alpha_{j,1} \\ \alpha_{j,2} \end{pmatrix}.$$

Haciendo el producto obtenemos:

$$\begin{aligned} \alpha_{j,0} &= c_0\alpha_{j+1,0} + c_1\alpha_{j+1,1} + c_2\alpha_{j+1,2} + c_3\alpha_{j+1,3}, \\ \alpha_{j,1} &= c_0\alpha_{j+1,2} + c_1\alpha_{j+1,3} + c_2\alpha_{j+1,4} + c_3\alpha_{j+1,5}, \\ \alpha_{j,2} &= c_0\alpha_{j+1,4} + c_1\alpha_{j+1,5} + c_2\alpha_{j+1,6} + c_3\alpha_{j+1,7}. \end{aligned}$$

Para la transformada inversa utilizamos las traspuestas,

$$\vec{\alpha}_{j+1} = L^T \vec{\alpha}_j + H^T \vec{\beta}_j,$$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 & c_0 \\ c_3 & c_1 \\ & c_2 & c_0 \\ & & c_3 & c_1 \\ & & & c_2 \\ & & & & c_3 \end{pmatrix} \begin{pmatrix} \alpha_{j,0} \\ \alpha_{j,1} \\ \alpha_{j,2} \end{pmatrix} + \begin{pmatrix} h_0 \\ h_1 \\ h_2 & h_0 \\ h_3 & h_1 \\ & h_2 & h_0 \\ & & h_3 & h_1 \\ & & & h_2 \\ & & & & h_3 \end{pmatrix} \begin{pmatrix} \beta_{j,0} \\ \beta_{j,1} \\ \beta_{j,2} \end{pmatrix} = \begin{pmatrix} \alpha_{j+1,0} \\ \alpha_{j+1,1} \\ \alpha_{j+1,2} \\ \alpha_{j+1,3} \\ \alpha_{j+1,4} \\ \alpha_{j+1,5} \\ \alpha_{j+1,6} \\ \alpha_{j+1,7} \end{pmatrix}.$$

Desarrollando obtenemos:

$$\begin{aligned} \alpha_{j+1,2} &= c_2\alpha_{j,0} + c_0\alpha_{j,1} + h_2\beta_{j,0} + h_0\beta_{j,1}, \\ \alpha_{j+1,3} &= c_3\alpha_{j,0} + c_1\alpha_{j,1} + h_3\beta_{j,0} + h_1\beta_{j,1}, \\ \alpha_{j+1,4} &= c_2\alpha_{j,1} + c_0\alpha_{j,2} + h_2\beta_{j,1} + h_0\beta_{j,2}, \\ \alpha_{j+1,5} &= c_3\alpha_{j,1} + c_1\alpha_{j,2} + h_3\beta_{j,1} + h_1\beta_{j,2}. \end{aligned}$$

Resumiendo lo visto, en general, la transformación directa será (ver la Figura 1.13):

$$\alpha_{j,i} = c_0\alpha_{j+1,2i} + c_1\alpha_{j+1,2i+1} + c_2\alpha_{j+1,2i+2} + c_3\alpha_{j+1,2i+3}, \quad (1.82)$$

$$\beta_{j,i} = h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + h_2\alpha_{j+1,2i+2} + h_3\alpha_{j+1,2i+3}, \quad (1.83)$$

y la inversa:

$$\alpha_{j+1,2i} = r_1\alpha_{j,i} + d_1\beta_{j,i} + r_3\alpha_{j,i-1} + d_3\beta_{j,i-1}, \quad (1.84)$$

$$\alpha_{j+1,2i+1} = r_0\alpha_{j,i} + d_0\beta_{j,i} + r_2\alpha_{j,i-1} + d_2\beta_{j,i-1}, \quad (1.85)$$

donde: $r_{2i+1} = c_{2i}$, $r_{2i} = c_{2i+1}$, $d_{2i+1} = h_{2i}$, $d_{2i} = h_{2i+1}$, para $i = 0, 1$.

En el caso de fronteras no periódicas debemos añadir las nuevas condiciones. Si n es el número de datos de partida, en la transformada directa debemos incluir:

$$\alpha_{j,n-1} = c_0\alpha_{j+1,2n-2} + c_1\alpha_{j+1,2n-1} + c_2\alpha_{j+1,0} + c_3\alpha_{j+1,1}.$$

En la transformada inversa las nuevas condiciones en la frontera izquierda son:

$$\alpha_{j+1,0} = c_0\alpha_{j,0} + c_2\alpha_{j,3} + h_0\beta_{j,0} + h_2\beta_{j,3},$$

$$\alpha_{j+1,1} = c_1\alpha_{j,0} + c_3\alpha_{j,3} + h_1\beta_{j,0} + h_3\beta_{j,3},$$

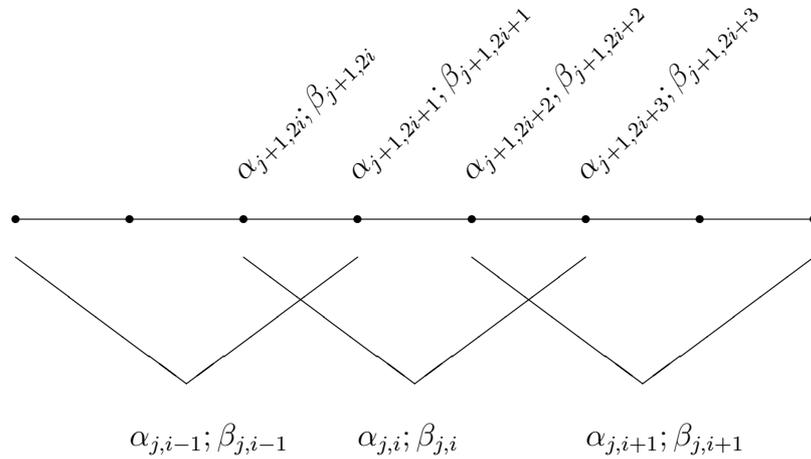


Figura 1.13: Transformada directa DB4

y en la frontera derecha:

$$\begin{aligned}\alpha_{j+1,2n-2} &= c_2\alpha_{j,2} + c_0\alpha_{j,3} + h_2\beta_{j,2} + h_0\beta_{j,3}, \\ \alpha_{j+1,2n-1} &= c_3\alpha_{j,2} + c_1\alpha_{j,3} + h_3\beta_{j,2} + h_1\beta_{j,3}.\end{aligned}$$

Siguiendo con el ejemplo de función propuesto (1.44), en las Figuras 1.14 y 1.15 mostramos los coeficientes $\alpha_{j,i}$ y $\beta_{j,i}$ obtenidos al utilizar 4 niveles. Observamos que los coeficientes $\beta_{j,i}$ contienen los detalles de la función. En las zonas suaves los coeficientes $\beta_{j,i}$ son de pequeña magnitud y por tanto podremos despreciarlos. De ahí las excelentes cualidades de esta transformada para la compresión. Sin embargo si los coeficientes $\beta_{j,i}$ son calculados a partir de *stencils* que cruzan discontinuidades su magnitud es elevada y si los eliminamos se produce el *fenómeno de Gibbs* en las proximidades de cada discontinuidad (ver la Figura 1.16). Para evitar estos problemas existen estrategias como los *ENO-wavelets* [26] y *wavelet-shrinkage* [63, 34, 51, 70] que veremos en secciones posteriores.

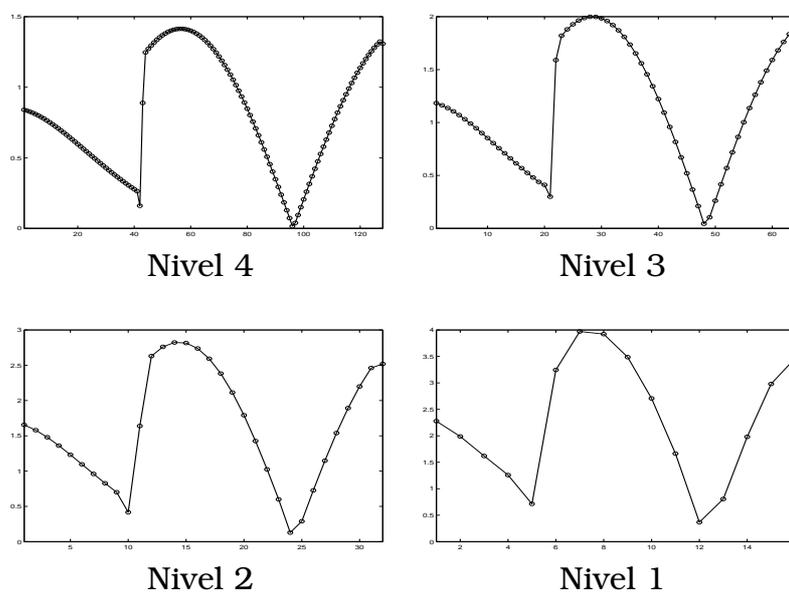


Figura 1.14: Coeficientes de bajas frecuencias al aplicar 4 niveles de la transformada wavelet DB4. Consideramos las fronteras periódicas y par-timos de 256 puntos iniciales.

1.5

Conclusiones

Hemos presentado las teorías matemáticas que utilizaremos en los capítulos posteriores. En primer lugar se han desarrollado las ideas de las descomposiciones multiescala *à la Harten*. Se han descrito con detalle dos realizaciones de estas descomposiciones: la multirresolución dada por la discretización por valores puntuales y la dada por medias en celda. A continuación se han descrito las técnicas de interpolación necesarias para las reconstrucciones multirresolutivas. Se ha visto cómo las técnicas de interpolación no lineales (*ENO* y *SR*) mejoran las aproximaciones en presencia de discontinuidades, aplicando cada caso a ejemplos concretos. Se ha prestado especial atención al cálculo de filtros, tanto de las técnicas lineales como de las no lineales. En la mayor parte de

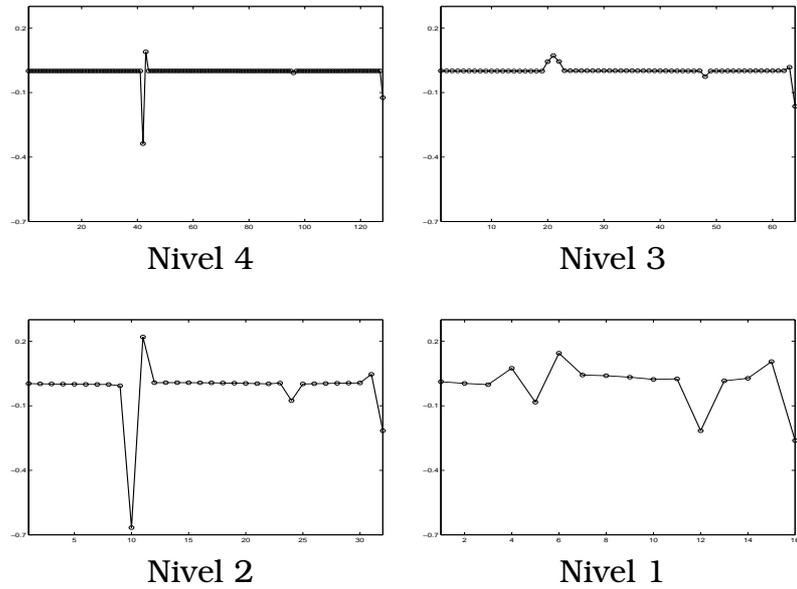


Figura 1.15: Coeficientes de altas frecuencias al aplicar 4 niveles de la transformada wavelet DB4. Consideramos las fronteras periódicas y partimos de 256 puntos iniciales.

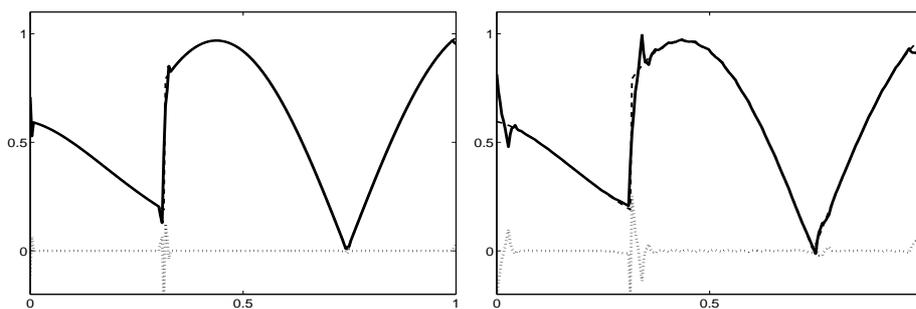


Figura 1.16: Aproximación obtenida mediante la transformada wavelet DB4. En la izquierda se utiliza un único nivel y en la derecha 3. Se eliminan todos los $\beta_{j,i}$ y las fronteras se tratan como periódicas. Partimos de 256 nodos iniciales. La función de partida se representa mediante la línea '-', la reconstrucción mediante línea continua y la diferencia entre ambas, a la que le restamos 0.1, se representa mediante puntos discontinuos.

las publicaciones, este cálculo se realiza a partir de desarrollos de la forma de Newton del polinomio interpolador, sin embargo en este trabajo hemos optado por calcularlos directamente a partir de la definición, haciendo uso de ecuaciones matriciales. Con esto se consiguen obtener de manera sencilla los filtros para diferentes grados y además, mediante cálculos similares, nos permitirá diseñar nuevas reconstrucciones.

En un segundo bloque nos hemos centrado en la transformada *wavelet*. Se ha descrito, tanto en su versión continua como en su versión discreta y se ha desarrollado con detalle la transformada *DB4*, aplicándola a ejemplos concretos.

En conclusión, se han presentado las teorías de multirresolución *à la Harten* y la transformada *wavelet*. Hemos seleccionado aquello que utilizaremos en desarrollos posteriores pero se ha intentado que la exposición sea clara y autocontenida, en el sentido que el lector pueda seguir los desarrollos sin acudir necesariamente a otros trabajos.

2

Nuevas reconstrucciones

2.1

Introducción

En el capítulo 1 expusimos la teoría asociada a la multirresolución *à la Harten*. La concretamos para dos discretizaciones (valores puntuales y medias en celda) y en ambos casos utilizamos polinomios interpoladores para definir las reconstrucciones. En este capítulo nos planteamos introducir nuevas reconstrucciones. Veremos si es posible incluirlas en esquemas de multirresolución *à la Harten*, estudiaremos su comportamiento y en capítulos posteriores las aplicaremos a la eliminación de ruido en señales digitales.

2.2

Reconstrucción interpolación aproximación

La nueva reconstrucción vendrá dada por un polinomio tal que interpole los extremos del intervalo de trabajo y aproxime por mínimos cuadrados en una serie de nodos adyacentes a dicho intervalo. Esta elección no es arbitraria, ya que la interpolación en los extremos será necesaria para que se cumpla que $\mathcal{D}_k \mathcal{R}_k f^k = f^k$, y la aproximación se introduce con el fin de trabajar con funciones a las que se les ha introducido ruido.

Nos referiremos a esta reconstrucción como *interpolación aproximación para valores puntuales* ($\overline{\mathcal{IAP}}$).

2.2.1

Interpolación aproximación para valores puntuales ($\overline{\mathcal{IAP}}$)

Tomando la discretización por valores puntuales y teniendo en cuenta las notaciones expuestas en la sección 1.2, definimos la reconstrucción **interpolación aproximación para valores puntuales** como sigue:

Dado un k y el intervalo $I_i^k = [x_{i-1}^k, x_i^k]$ construimos el polinomio que interpola a x_{i-1}^k, x_i^k y aproxima por mínimos cuadrados en m puntos de X^k adyacentes a I_i^k , para $i = 1, 2, \dots, J_k$. Llamamos $q_i^{\overline{\mathcal{IAP}}}(x; f^k, r)$ a dicho polinomio. El interpolante global viene definido por:

$$\overline{\mathcal{IAP}}_k(x; f^k) = q_i^{\overline{\mathcal{IAP}}}(x; f^k, r), \quad x \in [x_{i-1}^k, x_i^k], \quad i = 1, \dots, J_k. \quad (2.1)$$

Gracias a la interpolación en los extremos de cada intervalo, podemos asegurar que se cumple $\mathcal{D}_k \mathcal{R}_k f^k = f^k$, siendo $\overline{\mathcal{IAP}}_k(x; f^k) := \mathcal{R}_k f^k(x)$ un operador reconstrucción según la Definición 1.7. Así pues:

$$(P_{k-1}^k f^{k-1})_i = (\mathcal{D}_k \overline{\mathcal{IAP}}_{k-1}(\cdot f^{k-1}))_i = \overline{\mathcal{IAP}}_{k-1}(x_i^k; f^{k-1}).$$

El error de predicción se define como $e_i^k = f_i^k - \overline{\mathcal{IAP}}_{k-1}(x_i^k; f^{k-1})$, $i = 0, \dots, J_k$, con $e_i^k = 0$, $\forall i$ tal que $x_i^k \in X^{k-1}$. Los coeficientes $d^k = \{d_j^k\}$ son $d_j^k = e_{2j-1}^k$ para $j = 1, \dots, J_{k-1}$.

Con esto queda definida la reconstrucción *interpolación aproximación* cumpliendo las condiciones necesarias para ser una reconstrucción en el contexto de multirresolución *à la Harten*.

Cálculo de la reconstrucción $\overline{\mathcal{IAP}}$

Para cada intervalo $I_i^{k-1} = [x_{i-1}^{k-1}, x_i^{k-1}]$ buscamos un polinomio de grado r que interpole a la función en los extremos de dicho intervalo y que la aproxime por mínimos cuadrados en m puntos adyacentes al intervalo.

Sea $q(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$ dicho polinomio y $\{f_i^k\}_i$ la discretización de la función a reconstruir.

NOTA 2.1. Sin pérdida de generalidad y con el fin de simplificar la exposición, calcularemos $q(x)$ sobre los nodos $0, 1, 2, \dots$ y no sobre $\{x_i^{k-1}\}_i$ ya que éstos los suponemos equiespaciados.

Imponiendo que interpole en los extremos:

$$q(0) = a_1 = f_{i-1}^{k-1}, \quad (2.2)$$

$$q(1) = \sum_{l=1}^{r+1} a_l = f_i^{k-1} \Rightarrow a_2 = f_i^{k-1} - f_{i-1}^{k-1} - \sum_{l=3}^{r+1} a_l. \quad (2.3)$$

Por tanto,

$$\begin{aligned} q(x) &= f_{i-1}^{k-1} + \left(f_i^{k-1} - f_{i-1}^{k-1} - \sum_{l=3}^{r+1} a_l \right) x + \sum_{l=3}^{r+1} a_l x^{l-1} \\ &= (1-x)f_{i-1}^{k-1} + x f_i^{k-1} + \sum_{l=3}^{r+1} a_l (x^{l-1} - x), \end{aligned} \quad (2.4)$$

donde desconocemos $r - 1$ incógnitas.

Para determinar dichas incógnitas podemos imponer $nl - 1$ condiciones a la izquierda de x_{i-1}^{k-1} y $nr - 1$ a la derecha de x_i^{k-1} , con $nl + nr > r + 1$ (si $nl + nr = r + 1$ estaríamos únicamente interpolando):

$$q(-j) = f_{i-j-1}^{k-1}, \quad j = 1, 2, \dots, nl - 1, \quad (2.5)$$

$$q(j+1) = f_{i+j}^{k-1}, \quad j = 1, 2, \dots, nr - 1. \quad (2.6)$$

De (2.4), (2.5) y (2.6), obtenemos las $m = nl + nr - 2$ ecuaciones:

$$\sum_{l=3}^{r+1} a_l ((-j)^{l-1} + j) = f_{i-j-1}^{k-1} + (-j-1)f_{i-1}^{k-1} + jf_i^{k-1}, \quad j = 1, \dots, nl - 1,$$

$$\sum_{l=3}^{r+1} a_l ((j+1)^{l-1} - (j+1)) = f_{i+j}^{k-1} + jf_{i-1}^{k-1} - (j+1)f_i^{k-1}, \quad j = 1, \dots, nr - 1.$$

En forma matricial sería $A_{nl, nr}^{\overline{\mathcal{IAP}}} \hat{a} = \hat{f} - \hat{b}_0 f_{i-1}^{k-1} - \hat{b}_1 f_i^{k-1}$, donde tenemos la matriz $(nl + nr - 2) \times (r - 1)$:

$$A_{nl, nr}^{\overline{\mathcal{IAP}}} = \begin{bmatrix} (-(nl-1))^2 + nl - 1 & \cdots & (-(nl-1))^r + nl - 1 \\ \vdots & \ddots & \vdots \\ 6 & \cdots & (-2)^r + 2 \\ 2 & \cdots & (-1)^r + 1 \\ 2 & \cdots & 2^r - 2 \\ 6 & \cdots & 3^r - 3 \\ \vdots & \ddots & \vdots \\ (nr)^2 - nr & \cdots & (nr)^r - nr \end{bmatrix}, \quad (2.7)$$

y los vectores

$$\hat{a} = \begin{bmatrix} a_3 \\ a_4 \\ \vdots \\ a_{r+1} \end{bmatrix}, \quad \hat{f} = \begin{bmatrix} f_{i-nl}^{k-1} \\ \vdots \\ f_{i-3}^{k-1} \\ f_{i-2}^{k-1} \\ f_{i+1}^{k-1} \\ f_{i+2}^{k-1} \\ \vdots \\ f_{i+nr-1}^{k-1} \end{bmatrix}, \quad \hat{b}_0 = \begin{bmatrix} nl \\ \vdots \\ 3 \\ 2 \\ -1 \\ -2 \\ \vdots \\ -nr + 1 \end{bmatrix}, \quad \hat{b}_1 = \begin{bmatrix} -nl + 1 \\ \vdots \\ -2 \\ -1 \\ 2 \\ 3 \\ \vdots \\ nr \end{bmatrix}.$$

Para obtener los coeficientes $\{a_3, \dots, a_{r+1}\}$ del polinomio $q(x)$ debemos resolver el sistema $A_{nl,nr}^{\overline{\mathcal{IAP}}}^T A_{nl,nr}^{\overline{\mathcal{IAP}}} \hat{a} = A_{nl,nr}^{\overline{\mathcal{IAP}}}^T (\hat{f} - \hat{b}_0 f_{i-1}^{k-1} - \hat{b}_1 f_i^{k-1})$:

$$\hat{a} = \left(A_{nl,nr}^{\overline{\mathcal{IAP}}}^T A_{nl,nr}^{\overline{\mathcal{IAP}}} \right)^{-1} A_{nl,nr}^{\overline{\mathcal{IAP}}}^T (\hat{f} - \hat{b}_0 f_{i-1}^{k-1} - \hat{b}_1 f_i^{k-1}). \quad (2.8)$$

Finalmente mediante (2.2) y (2.3) hallamos a_1 y a_2 . En efecto, si definimos la matriz $(r-1) \times m : C = \left(A_{nl,nr}^{\overline{\mathcal{IAP}}}^T A_{nl,nr}^{\overline{\mathcal{IAP}}} \right)^{-1} A_{nl,nr}^{\overline{\mathcal{IAP}}}^T$, tenemos:

$$\begin{aligned} a_1 &= f_{i-1}^{k-1}, \\ a_2 &= f_i^{k-1} - f_{i-1}^{k-1} - \mathbf{1}_{r-1}^T \hat{a} \\ &= f_i^{k-1} - f_{i-1}^{k-1} - \mathbf{1}_{r-1}^T C \hat{f} + \mathbf{1}_{r-1}^T C \hat{b}_0 f_{i-1}^{k-1} + \mathbf{1}_{r-1}^T C \hat{b}_1 f_i^{k-1}. \end{aligned}$$

Al polinomio $q(x)$ que hemos calculado lo denotaremos

$$\overline{\mathcal{IAP}}_{nl,nr}^r(x, f^{k-1}) = p_i^{\overline{\mathcal{IAP}}}(x; f^{k-1}, r).$$

Sobre los nodos $\{x_i^{k-1}\}$, la notación será $q_i^{\overline{\mathcal{IAP}}}(x; f^{k-1}, r)$.

El valor que aproxima a f_{2i-1}^k será $p_i^{\overline{\mathcal{IAP}}}(\frac{1}{2}; f^{k-1}, r)$. Así:

$$\begin{aligned} \tilde{f}_{2i-1}^k &= \sum_{l=1}^{r+1} a_l \frac{1}{2^{l-1}} = f_{i-1}^{k-1} + \frac{1}{2} f_i^{k-1} - \frac{1}{2} f_{i-1}^{k-1} - \frac{1}{2} \mathbf{1}_{r-1}^T C \hat{f} \\ &\quad + \frac{1}{2} \mathbf{1}_{r-1}^T C \hat{b}_0 f_{i-1}^{k-1} + \frac{1}{2} \mathbf{1}_{r-1}^T C \hat{b}_1 f_i^{k-1} + \left(\frac{1}{2^2}, \dots, \frac{1}{2^r} \right) C \hat{f} \\ &\quad - \left(\frac{1}{2^2}, \dots, \frac{1}{2^r} \right) C \hat{b}_0 f_{i-1}^{k-1} - \left(\frac{1}{2^2}, \dots, \frac{1}{2^r} \right) C \hat{b}_1 f_i^{k-1} \\ &= \left(1 - \frac{1}{2} + \left(\frac{1}{2} - \frac{1}{2^2}, \dots, \frac{1}{2} - \frac{1}{2^r} \right) C \hat{b}_0 \right) f_{i-1}^{k-1} \\ &\quad + \left(\frac{1}{2} + \left(\frac{1}{2} - \frac{1}{2^2}, \dots, \frac{1}{2} - \frac{1}{2^r} \right) C \hat{b}_1 \right) f_i^{k-1} + \left(\frac{1}{2^2} - \frac{1}{2}, \dots, \frac{1}{2^r} - \frac{1}{2} \right) C \hat{f} \\ &= \sum_{l=-nl}^{nr-1} (\beta_{nl,nr}^r)_l f_{i+l}^{k-1} \equiv \overline{\mathcal{IAP}}_{nl,nr}^r(x_{2i-1}^k, f^{k-1}), \end{aligned}$$

donde:

$$\begin{aligned} (\beta_{nl,nr}^r)_l &= \left(\frac{1}{2^2} - \frac{1}{2}, \dots, \frac{1}{2^r} - \frac{1}{2} \right) C e_{l+nl+1}, \quad l = -nl, \dots, -2, \\ (\beta_{nl,nr}^r)_{-1} &= \left(1 - \frac{1}{2} + \left(\frac{1}{2} - \frac{1}{2^2}, \dots, \frac{1}{2} - \frac{1}{2^r} \right) C \hat{b}_0 \right), \\ (\beta_{nl,nr}^r)_0 &= \left(\frac{1}{2} + \left(\frac{1}{2} - \frac{1}{2^2}, \dots, \frac{1}{2} - \frac{1}{2^r} \right) C \hat{b}_1 \right), \end{aligned}$$

$$(\beta_{nl,nr}^r)_l = \left(\frac{1}{2^2} - \frac{1}{2}, \dots, \frac{1}{2^r} - \frac{1}{2}\right) C e_{nl-1+l}, \quad l = 1, \dots, nr - 1,$$

siendo $e_j = (\delta_{j,k})_{k=1}^{nl+nr-2}$. En la Tabla 2.1, podemos ver¹ los filtros para ciertos grados y tamaños del stencil. Se cumple:

$$(\beta_{nl,nr}^r)_{-nl+l} = (\beta_{nr,nl}^r)_{nl-l-1}, \quad l = 0, \dots, nl + nr - 1. \quad (2.9)$$

NOTA 2.2. En (2.9) algunos coeficientes pueden no ser exactamente la misma fracción, aunque en su forma decimal serán muy aproximados. Se debe a que las últimas cifras decimales difieren por precisión en los cálculos y al pasarlas a fracción no son exactamente iguales.

| l | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
|---------------------|------------------|------------------|-------------------|-------------------|--------------------|--------------------|------------------|------------------|------------------|
| $(\beta_{3,1}^2)_l$ | | | $\frac{-3}{80}$ | $\frac{-1}{80}$ | $\frac{51}{80}$ | $\frac{33}{80}$ | | | |
| $(\beta_{2,2}^2)_l$ | | | | $\frac{-1}{16}$ | $\frac{9}{16}$ | $\frac{9}{16}$ | $\frac{-1}{16}$ | | |
| $(\beta_{4,1}^2)_l$ | | $\frac{-3}{184}$ | $\frac{-3}{368}$ | $\frac{-1}{368}$ | $\frac{219}{368}$ | $\frac{159}{368}$ | | | |
| $(\beta_{3,2}^2)_l$ | | | $\frac{-3}{88}$ | $\frac{-1}{88}$ | $\frac{27}{44}$ | $\frac{39}{88}$ | $\frac{-1}{88}$ | | |
| $(\beta_{5,1}^2)_l$ | $\frac{-5}{584}$ | $\frac{-3}{584}$ | $\frac{-3}{1168}$ | $\frac{-1}{1168}$ | $\frac{669}{1168}$ | $\frac{519}{1168}$ | | | |
| $(\beta_{4,2}^2)_l$ | | $\frac{-3}{188}$ | $\frac{-3}{376}$ | $\frac{-1}{376}$ | $\frac{111}{188}$ | $\frac{165}{376}$ | $\frac{-1}{376}$ | | |
| $(\beta_{3,3}^3)_l$ | | | $\frac{-3}{160}$ | $\frac{-1}{160}$ | $\frac{21}{40}$ | $\frac{21}{40}$ | $\frac{-1}{160}$ | $\frac{-3}{160}$ | |
| $(\beta_{4,4}^3)_l$ | | $\frac{-3}{368}$ | $\frac{-3}{736}$ | $\frac{-1}{736}$ | $\frac{189}{368}$ | $\frac{189}{368}$ | $\frac{-1}{736}$ | $\frac{-3}{736}$ | $\frac{-3}{368}$ |
| f_{i-l}^{k-1} | f_{i-5}^{k-1} | f_{i-4}^{k-1} | f_{i-3}^{k-1} | f_{i-2}^{k-1} | f_{i-1}^{k-1} | f_i^{k-1} | f_{i+1}^{k-1} | f_{i+2}^{k-1} | f_{i+3}^{k-1} |

Tabla 2.1: Aproximaciones a f_{2i-1}^k , con grado del polinomio r , $\overline{\mathcal{IAP}}_{nl,nr}^r(x_{2i-1}^k, f^{k-1}) = \sum_{l=-nl}^{nr-1} (\beta_{nl,nr}^r)_l f_{i+l}^{k-1}$, utilizando interpolación con 2 puntos y aproximación con $nl - 1$ puntos a la izquierda de x_{i-1}^{k-1} y $nr - 1$ puntos a la derecha de x_i^{k-1} .

Si $n = nl = nr$ obtenemos que el intervalo donde interpolamos está centrado con respecto a los nodos en los que aproximamos.

¹En <http://www.uv.es/anims/noguera/> podemos descargar una GUI y programas en MATLAB® que calculan los filtros según los datos introducidos por el usuario.

En caso contrario hay más nodos a un lado que al otro, lo cual se utilizará para evitar las discontinuidades, además de localizarlas.

Orden de la reconstrucción $\overline{\mathcal{IAP}}$

Para determinar de qué orden es la reconstrucción nos basaremos en estudiar qué polinomios reconstruye de forma exacta.

PROPOSICIÓN 2.1. Si $q_i^{\overline{\mathcal{IAP}}}(x; f^{k-1}, r)$ es el polinomio $\overline{\mathcal{IAP}}$ para valores puntuales de grado r , la reconstrucción $\overline{\mathcal{IAP}}$ recupera de forma exacta polinomios de grado menor o igual a r .

Demostración. Sin pérdida de generalidad, trabajamos sobre nodos enteros.

Sea $p_i^{\overline{\mathcal{IAP}}}(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$ y sea $f(x) = q(x) = \sum_{l=1}^{s+1} b_l x^{l-1}$ la función a reconstruir.

Si $s > r$, la reconstrucción no es exacta en general, ya que un polinomio de grado s no puede ser recuperado por otro de grado inferior.

Si $s = r$, se deben cumplir las ecuaciones de interpolación:

$$p_i^{\overline{\mathcal{IAP}}}(0) = q(0) \quad \text{y} \quad p_i^{\overline{\mathcal{IAP}}}(1) = q(1) \quad (2.10)$$

y las de aproximación;

$$p_i^{\overline{\mathcal{IAP}}}(j) = q(j), \quad j = -(nl - 1), \dots, -1, 2, 3, \dots, nr, \quad (2.11)$$

donde $nl \geq 1$, $nr \geq 1$ y $nl + nr > r + 1$.

Mediante la matriz $r + 1 \times (nl + nr)$:

$$M_{r+1} = \begin{bmatrix} 1 & 1 - nl & (1 - nl)^2 & \cdots & (1 - nl)^r \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & -1 & 1 & \cdots & (-1)^r \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 4 & \cdots & 2^r \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & nr & nr^2 & \cdots & nr^r \end{bmatrix}, \quad (2.12)$$

podemos expresar (2.10) y (2.11) en forma matricial:

$$M_{r+1} \begin{bmatrix} a_1 \\ \vdots \\ a_{r+1} \end{bmatrix} = M_{r+1} \begin{bmatrix} b_1 \\ \vdots \\ b_{r+1} \end{bmatrix}.$$

Haciendo el producto

$$M_{r+1} \begin{bmatrix} b_1 \\ \vdots \\ b_{r+1} \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_{r+1} \end{bmatrix},$$

el sistema

$$M_{r+1} \begin{bmatrix} a_1 \\ \vdots \\ a_{r+1} \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_{r+1} \end{bmatrix},$$

es de solución única si $\text{rang}(M_{r+1}) = r + 1$, en cuyo caso, dado que $[b_1, \dots, b_{r+1}]^T$ es una solución, por unicidad $a_i = b_i$ para $i = 1, \dots, r + 1$ y por tanto el polinomio $q(x)$ se recupera de forma exacta.

Si tomamos $r + 1$ filas de la matriz M_{r+1} tenemos una matriz de *Vandermonde* con determinante distinto de cero, con lo que efectivamente $\text{rang}(M_{r+1}) = r + 1$ y ya estaría probado.

Si $s < r$ se reduce al caso anterior sin más que anular los coeficientes a_l para $s + 1 < l \leq r + 1$.

Por tanto la reconstrucción es exacta para polinomios de grado menor o igual a r .

□

COROLARIO 2.1. El orden de la reconstrucción $\overline{\mathcal{IAP}}$ es $r + 1$.

ENO

Al igual que en las reconstrucciones exclusivamente interpolatorias, para cada discontinuidad se producen una serie de intervalos de escasa aproximación. En concreto, si el intervalo que contiene la discontinuidad es $(x_{i-1}^{k-1}, x_i^{k-1})$ y estamos trabajando con condiciones centradas con $nl = nr = n$, los intervalos afectados son: $[x_{i-n}^{k-1}, \dots, x_{i+n-1}^{k-1}]$. Y en términos de coeficientes de escala:

$$d_l^{k-1}, \quad i - nl + 1 \leq l \leq i + nr. \quad (2.13)$$

Incluiremos adaptatividad en las discontinuidades con el objetivo de evitar utilizar *stencils* que crucen discontinuidades. Si la función con la que trabajamos no contiene ruido, podemos utilizar el esquema ENO aplicando el Algoritmo 1.1 o 1.2 (*ENO-VP-I* o *ENO-VP-II* respectivamente) para *stencils* de longitud $nl + nr$.

Cuando desplazamos un *stencil* según la asignación ENO hemos de tener en cuenta que en nuestra reconstrucción, si trabajamos en el intervalo I_i^{k-1} , debemos interpolar en los nodos x_{i-1}^{k-1} , x_i^{k-1} y podemos variar el número de nodos donde aproximamos a izquierda y derecha para conseguir dicho desplazamiento. Para cada $m = nl + nr$ fijo, esto se traduce en tomar (ver Figura 2.1):

$$nl_i = i - s_i + 1, \quad nr_i = m - nl_i, \quad (2.14)$$

con $i - m + 2 \leq s_i \leq i + m - 2$, siendo s_i el definido en la sección 1.2.2.

Con esto conseguimos que para una función suave los *stencils* no crucen discontinuidades, mejorando los resultados.

WENO

Basándonos en lo expuesto en la sección 1.2.4, veremos si es posible adaptar WENO a la reconstrucción $\overline{\mathcal{IAP}}$. Los pasos son:

1.- Expresar la aproximación de orden $r + 1$ como una combinación lineal de las n aproximaciones de orden $s + 1$, con $r > s$:

$$p_i^{\overline{\mathcal{IAP}}^{r,n}}(x) = \sum_{k=0}^{n-1} C_k^s p_i^{\overline{\mathcal{IAP}}^{s,n-k,k+1}}(x),$$

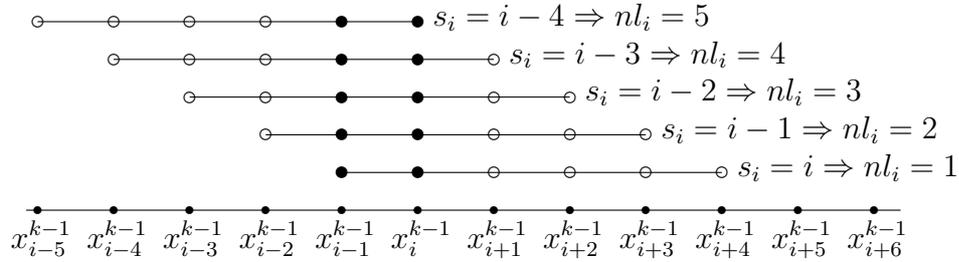


Figura 2.1: Posibles stencils de 6 nodos ($m = 6 = nl + nr$) para el intervalo $I_i^{k-1} = [x_{i-1}^{k-1}, x_i^{k-1}]$ en la reconstrucción $\overline{\mathcal{IAP}}$. Se marca con un círculo los nodos donde se interpola y con una circunferencia donde se aproxima.

con $C_k^s \geq 0 \forall k$, y $\sum_{k=0}^{n-1} C_k^s = 1$, siendo $p_i^{\overline{\mathcal{IAP}}^{nl, nr, t}}$ el polinomio interpolación aproximación de orden $t + 1$ construido tomando nl nodos a la izquierda y nr a la derecha de x_{2i-1}^k .

En WENO para valores puntuales y reconstrucción interpolatoria $r = 2s - 1$ debido a que la longitud del stencil determina el grado del polinomio interpolador. En este caso debe cumplirse simplemente que $r > s$ para que WENO produzca un aumento del orden de aproximación. Además también podemos elegir el número de nodos, $n = nl = nr$, donde aproximamos, teniendo en cuenta que $2n > r + 1$ y $2(n - 1) > s + 1$ para que se aproxime y no se interpole.

En la Tabla 2.2 vemos los coeficientes C_k^r que obtenemos para algunos valores de n , r y s . Notar que los casos viables para WENO son los que cumplen $C_k^s \geq 0$ y $\sum_{k=0}^{n-1} C_k^s = 1$. La mayoría de casos nos salen incompatibles. En otros casos, como para $s = 2$, $r = 4$ y $n = 3$ no se cumple que $C_k^s \geq 0 \forall k$.

Tras efectuar los cálculos con diferentes grados y longitudes de stencil, los casos factibles para aplicar WENO son:

$$p_i^{\overline{\mathcal{IAP}}^{n, n}}(x) = \frac{1}{2} p_i^{\overline{\mathcal{IAP}}^{n, 1}}(x) + \frac{1}{2} p_i^{\overline{\mathcal{IAP}}^{1, n}}(x), \quad n \geq 2,$$

es decir:

$$\sum_{l=-n}^{n-1} (\beta_{n, n}^3)_l f_{i+l} = \frac{1}{2} \sum_{l=-n}^0 (\beta_{n, 1}^2)_l f_{i+l} + \frac{1}{2} \sum_{l=-1}^{n-1} (\beta_{1, n}^2)_l f_{i+l}, \quad n \geq 2. \quad (2.15)$$

2.- Definimos $\alpha_k^s = \frac{C_k^s}{(\varepsilon + \mathcal{I}S_{n-k, k+1})^2}$, $k = 0, \dots, n - 1$, con $\mathcal{I}S_{n-k, k+1}$ un

| | | s = 2 | | | s = 3 | | |
|-------|---------|-----------------|---------------|---------------|-------|-------|-------|
| | | n = 3 | n = 4 | n = 5 | n = 4 | n = 5 | n = 6 |
| r = 3 | C_0^s | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | | | |
| | C_1^s | 0 | 0 | 0 | | | |
| | C_2^s | $\frac{1}{2}$ | 0 | 0 | | | |
| | C_3^s | | $\frac{1}{2}$ | 0 | | | |
| | C_4^s | | | $\frac{1}{2}$ | | | |
| r = 4 | C_0^s | $-\frac{5}{16}$ | S.I. | S.I. | S.I. | S.I. | S.I. |
| | C_1^s | $\frac{13}{8}$ | S.I. | S.I. | S.I. | S.I. | S.I. |
| | C_2^s | $-\frac{5}{16}$ | S.I. | S.I. | S.I. | S.I. | S.I. |
| | C_3^s | | S.I. | S.I. | S.I. | S.I. | S.I. |
| | C_4^s | | | S.I. | | S.I. | S.I. |
| | C_5^s | | | | | | S.I. |

Tabla 2.2: Coeficientes C_k^s tales que $p_i^{\overline{\mathcal{IAP}}_{n,n}^r}(x) = \sum_{k=0}^{n-1} C_k^s p_i^{\overline{\mathcal{IAP}}_{n-k,k+1}^s}(x)$. S.I. significa que el sistema es incompatible.

indicador de suavidad de $f(x)$ en el stencil $\{x_{i-n-k}^{k-1}, \dots, x_{i+k}^{k-1}\}$. Para comprobar el funcionamiento del algoritmo emplearemos en indicador de suavidad de Liu et al. (1.28) o el de Jiang y Shu (1.29). Dichos indicadores están diseñados para reconstrucciones interpolatorias, aunque los utilizaremos en $\overline{\mathcal{IAP}}$ ya que nuestra pretensión es únicamente ver la viabilidad de WENO para la nueva reconstrucción. Para ser totalmente rigurosos deberíamos adaptar los indicadores de suavidad a la reconstrucción $\overline{\mathcal{IAP}}$.

3.- La aproximación WENO será

$$p_i^{WENO-\overline{\mathcal{IAP}}}(x) = \sum_{k=0}^{n-1} \omega_k^s p_i^{\overline{\mathcal{IAP}}_{n-k,k+1}^s}(x),$$

donde $\omega_k^s = \frac{\alpha_k^s}{\sum_{j=0}^{n-1} \alpha_j^s}$, para $k = 0, \dots, n - 1$.

PPH

Este caso es análogo al visto para la reconstrucción interpolatoria, sección 1.2.5, ya que los coeficientes son los mismos:

$$\overline{\mathcal{IAP}}_{2,2}^2(x_{2i-1}^k, f^{k-1}) = -\frac{1}{16}f_{i-2}^{k-1} + \frac{9}{16}f_{i-1}^{k-1} + \frac{9}{16}f_i^{k-1} - \frac{1}{16}f_{i+1}^{k-1}. \quad (2.16)$$

con lo cual, tomando la media armónica $H(x, y)$, (1.30):

$$\left(\overline{\mathcal{IAP}}_{2,2}^2\right)^{PPH}(x_{2i-1}^k, f^{k-1}) = \frac{1}{2}(f_{i-1}^{k-1} + f_i^{k-1}) - \frac{1}{8}H(d^2 f_{i-1}^{k-1}, d^2 f_i^{k-1}),$$

La anterior es la única elección posible para proporcionar una aproximación con *stencils* de 4 nodos, ya que si $r = 1$ no se cumple la condición de aproximación.

ENO-SR

Basándonos en el algoritmo *ENO-SR* para valores puntuales y reconstrucción interpolatoria, lo adaptamos para la reconstrucción $\overline{\mathcal{IAP}}$ de la siguiente manera:

1.- Tomando *stencils* de $m = nl + nr$ nodos, calculamos los *stencils* *ENO* mediante el Algoritmo 1.1 o 1.2 (*ENO-VP-I* o *ENO-VP-II*) y asignamos los nodos donde aproximaremos para cada intervalo según (2.14).

2.- Si $nl_{i-1} = m - 1$ y $nl_{i+1} = 1$ los *stencils* para las celdas I_{i-1}^{k-1} y I_{i+1}^{k-1} son disjuntos. Etiquetamos dicha celda como *sospechosa* de contener una discontinuidad.

3.- Para cada celda sospechosa definimos la función

$$G_i^{\overline{\mathcal{IAP}}}(x) = q_{s_{i+1}}^{\overline{\mathcal{IAP}}}(x; f^{k-1}, r) - q_{s_{i-1}}^{\overline{\mathcal{IAP}}}(x; f^{k-1}, r),$$

donde s_i indica que elegimos el *stencil* según la elección *ENO*. Si $G_i^{\overline{\mathcal{IAP}}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{\mathcal{IAP}}}(x_i^{k-1}) < 0$ etiquetamos la celda I_i^{k-1} como singular.

4.- Si $G_i^{\overline{\mathcal{IAP}}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{\mathcal{IAP}}}(x_{2i-1}^k) < 0$, el punto x_{2i-1}^k se halla a la derecha de la discontinuidad.

Antes de exponer el cálculo detallado de filtros una cuestión de notación. En lo que resta de apartado denotaremos:

$$\begin{aligned}
C^I &= \left(A_{nl+nr-1,1}^{\overline{\mathcal{IAP}}} \quad A_{nl+nr-1,1}^{\overline{\mathcal{IAP}}} \right)^{-1} A_{nl+nr-1,1}^{\overline{\mathcal{IAP}}} \quad T, \\
\hat{b}_0^I &= [nl + nr - 1, nl + nr - 2, \dots, 2]^T, \\
\hat{b}_1^I &= [-nl - nr + 2, -nl - nr + 3, \dots, -1]^T, \\
C^D &= \left(A_{1,nl+nr-1}^{\overline{\mathcal{IAP}}} \quad A_{1,nl+nr-1}^{\overline{\mathcal{IAP}}} \right)^{-1} A_{1,nl+nr-1}^{\overline{\mathcal{IAP}}} \quad T, \\
\hat{b}_0^D &= [-1, -2, \dots, -nl - nr + 2]^T, \\
\hat{b}_1^D &= [2, 3, \dots, nl + nr - 1]^T, \\
\hat{f}_D &= [f_{i+1}^{k-1}, \dots, f_{i+nl+nr-2}^{k-1}]^T, \\
\hat{f}_I &= [f_{i-nl-nr}^{k-1}, \dots, f_{i-2}^{k-1}]^T.
\end{aligned}$$

Los filtros para \tilde{f}_{2i-1}^k cuando nos encontramos a la derecha de la discontinuidad se obtienen de la siguiente manera:

$$\begin{aligned}
\tilde{f}_{2i-1}^k &= q_{s_{i+1}}^{\overline{\mathcal{IAP}}}(x_{2i-1}^k; f^{k-1}, r) = p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(-1/2; f^{k-1}, r) \\
&= \sum_{l=1}^{r+1} a_l \left(-\frac{1}{2}\right)^{l-1} = a_1 - \frac{1}{2}a_2 + \sum_{l=3}^{r+1} a_l \left(-\frac{1}{2}\right)^{l-1} \\
&= f_i^{k-1} - \frac{1}{2} \left[f_{i+1}^{k-1} - f_i^{k-1} - \mathbf{1}_{r-1}^T C^D \left(\hat{f}_D - \hat{b}_0^D f_i^{k-1} - \hat{b}_1^D f_{i+1}^{k-1} \right) \right] \\
&\quad + \left[\left(-\frac{1}{2}\right)^2, \dots, \left(-\frac{1}{2}\right)^r \right] C^D \left(\hat{f}_D - \hat{b}_0^D f_i^{k-1} - \hat{b}_1^D f_{i+1}^{k-1} \right) \\
&= \sum_{l=0}^{nl+nr-1} (\beta_{0,nl+nr}^r)_l f_l^{k-1}, \tag{2.17}
\end{aligned}$$

donde:

$$\begin{aligned}
(\beta_{0,nl+nr}^r)_0 &= 1 + \frac{1}{2} - \frac{1}{2} \mathbf{1}_{r-1}^T C^D \hat{b}_0^D - \left[\left(-\frac{1}{2}\right)^2, \dots, \left(-\frac{1}{2}\right)^r \right] C^D \hat{b}_0^D, \\
(\beta_{0,nl+nr}^r)_1 &= -\frac{1}{2} - \frac{1}{2} \mathbf{1}_{r-1}^T C^D \hat{b}_1^D - \left[\left(-\frac{1}{2}\right)^2, \dots, \left(-\frac{1}{2}\right)^r \right] C^D \hat{b}_1^D, \\
(\beta_{0,nl+nr}^r)_l &= \left(\frac{1}{2} \mathbf{1}_{r-1}^T + \left[\left(-\frac{1}{2}\right)^2, \dots, \left(-\frac{1}{2}\right)^r \right] \right) C^D e_l,
\end{aligned}$$

siendo $e_l = (\delta_{l,k})_{k=2}^{nl+nr-1}$ para $l = 2, \dots, nl + nr - 1$. En la Tabla 2.3 podemos ver dichos coeficientes para ciertos valores de nl , nr y r .

En otro caso x_{2i-1}^k se halla a la izquierda de la discontinuidad:
 $\tilde{f}_{2i-1}^k = q_{s_{i-1}}^{\overline{\mathcal{IAP}}}(x_{2i-1}^k; f^{k-1}, r)$ (o bien $\tilde{f}_{2i-1}^k = p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(3/2; f^{k-1}, r)$).

$$\begin{aligned}
\tilde{f}_{2i-1}^k &= q_{s_{i-1}}^{\overline{\mathcal{IAP}}}(x_{2i-1}^k; f^{k-1}, r) = p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(3/2; f^{k-1}, r) \\
&= \sum_{l=1}^{r+1} a_l \left(\frac{3}{2}\right)^{l-1} = a_1 + \frac{3}{2}a_2 + \sum_{l=3}^{r+1} a_l \left(\frac{3}{2}\right)^{l-1} \\
&= f_{i-2}^{k-1} + \frac{3}{2} \left[f_{i-1}^{k-1} - f_{i-2}^{k-1} - \mathbf{1}_{r-1}^T C^I \left(\hat{f}_I - \hat{b}_0^I f_{i-2}^{k-1} - \hat{b}_1^I f_{i-1}^{k-1} \right) \right] \\
&\quad + \left[\left(\frac{3}{2}\right)^2, \dots, \left(\frac{3}{2}\right)^r \right] C^I \left(\hat{f}_I - \hat{b}_0^I f_{i-2}^{k-1} - \hat{b}_1^I f_{i-1}^{k-1} \right) \\
&= \sum_{l=-nl-nr}^{-1} (\beta_{nl+nr,0}^r)_l f_l^{k-1}, \tag{2.18}
\end{aligned}$$

donde:

$$\begin{aligned}
(\beta_{nl+nr,0}^r)_{-1} &= \frac{3}{2} + \frac{3}{2} \mathbf{1}_{r-1}^T C^I \hat{b}_1^I - \left[\left(\frac{3}{2}\right)^2, \dots, \left(\frac{3}{2}\right)^r \right] C^I \hat{b}_1^D, \\
(\beta_{nl+nr,0}^r)_{-2} &= 1 - \frac{3}{2} + \frac{3}{2} \mathbf{1}_{r-1}^T C^I \hat{b}_0^I - \left[\left(\frac{3}{2}\right)^2, \dots, \left(\frac{3}{2}\right)^r \right] C^I \hat{b}_0^D, \\
(\beta_{nl+nr,0}^r)_l &= - \left(\frac{3}{2} \mathbf{1}_{r-1}^T + \left[\left(\frac{3}{2}\right)^2, \dots, \left(\frac{3}{2}\right)^r \right] \right) C^I e_l,
\end{aligned}$$

siendo $e_l = (\delta_{l,k})_{k=-nl-nr}^{-3}$ para $l = -nl - nr, \dots, -3$. Dichos coeficientes pueden deducirse de los hallados para el caso derecho ya que se cumple (ver Nota 2.2):

$$(\beta_{0,nl+nr}^r)_l = (\beta_{nl+nr,0}^r)_{-l-1}, \quad l = 0, \dots, nl + nr - 1$$

NOTA 2.3. Al aplicar SR de la forma descrita, f_{2i-1}^k no está en el centro de un intervalo donde se interpolan los extremos, sino que el intervalo donde se interpola está a izquierda o a derecha de f_{2i-1}^k . Esta modificación nos permite adaptar SR para interpolación aproximación.

Para finalizar el apartado, detallamos el cálculo de filtros para la función $G^{\overline{\mathcal{IAP}}}$:

$$G_i^{\overline{\mathcal{IAP}}}(x) = \sum_{l=1}^{r+1} a_l (x-1)^{l-1} - \sum_{l=1}^{r+1} b_l (x+1)^{l-1}$$

| l | 0 | 1 | 2 | 3 | 4 | 5 |
|---------------------|---------------------|---------------------|--------------------|------------------|--------------------|---------------------|
| $(\beta_{0,4}^2)_l$ | $\frac{141}{80}$ | $-\frac{73}{80}$ | $\frac{3}{80}$ | $\frac{9}{80}$ | | |
| $(\beta_{0,6}^2)_l$ | $\frac{1947}{1168}$ | $-\frac{839}{1168}$ | $\frac{3}{1168}$ | $\frac{9}{1168}$ | $\frac{9}{584}$ | $\frac{15}{584}$ |
| $(\beta_{0,6}^3)_l$ | $\frac{1044}{545}$ | $-\frac{626}{521}$ | $\frac{388}{4425}$ | $\frac{94}{569}$ | $\frac{314}{2331}$ | $-\frac{326}{3207}$ |
| f_{i+l}^{k-1} | f_i^{k-1} | f_{i+1}^{k-1} | f_{i+2}^{k-1} | f_{i+3}^{k-1} | f_{i+4}^{k-1} | f_{i+5}^{k-1} |

Tabla 2.3: Aproximaciones a f_{2i-1}^{k-1} , de orden $r+1$, $\overline{\mathcal{IAP}}_{0,nl+nr}^r(x_{2i-1}^k, f^{k-1}) = \sum_{l=0}^{nl+nr-1} (\beta_{0,nl+nr}^r)_l f_{i+l}^{k-1}$, utilizando interpolación en f_i^{k-1} , f_{i+1}^{k-1} y aproximación en $nl + nr - 2$ puntos a la derecha de f_{i+1}^{k-1} .

$$\begin{aligned}
&= a_1 + (x-1)a_2 + \sum_{l=3}^{r+1} a_l (x-1)^{l-1} - b_1 - (x+1)b_2 - \sum_{l=3}^{r+1} (x+1)^{l-1} \\
&= f_i^{k-1} + (x-1) \left[f_{i+1}^{k-1} - f_i^{k-1} - \mathbf{1}_{r-1}^T C^D \left(\hat{f}_D - \hat{b}_0^I f_i^{k-1} - \hat{b}_1^I f_{i+1}^{k-1} \right) \right] \\
&\quad + [(x-1)^2, \dots, (x-1)^r] C^D \left(\hat{f}_D - \hat{b}_0^D f_i^{k-1} - \hat{b}_1^D f_{i+1}^{k-1} \right) \\
&= -f_{i-2}^{k-1} - (x-1) \left[f_{i-1}^{k-1} - f_{i-2}^{k-1} - \mathbf{1}_{r-1}^T C^I \left(\hat{f}_I - \hat{b}_0^I f_{i-2}^{k-1} - \hat{b}_1^I f_{i-1}^{k-1} \right) \right] \\
&\quad + [(x+1)^2, \dots, (x+1)^r] C^I \left(\hat{f}_I - \hat{b}_0^I f_{i-2}^{k-1} - \hat{b}_1^I f_{i-1}^{k-1} \right) \\
&= \sum_{l=-nl-nr}^{nl+nr-1} (\sigma_{nl,nr}^r)_l f_l^{k-1},
\end{aligned}$$

siendo:

$$\begin{aligned}
(\sigma_{nl,nr}^r)_0 &= 2 - x + (x-1) \mathbf{1}_{r-1}^T C^D \hat{b}_0^D - [(x-1)^2, \dots, (x-1)^r] C^D \hat{b}_0^D, \\
(\sigma_{nl,nr}^r)_1 &= (x-1) + (x-1) \mathbf{1}_{r-1}^T C^D \hat{b}_1^D - [(x-1)^2, \dots, (x-1)^r] C^D \hat{b}_1^D, \\
(\sigma_{nl,nr}^r)_{-1} &= -(x+1) - (x+1) \mathbf{1}_{r-1}^T C^I \hat{b}_1^I + [(x+1)^2, \dots, (x+1)^r] C^I \hat{b}_1^I, \\
(\sigma_{nl,nr}^r)_{-2} &= x - (x+1) \mathbf{1}_{r-1}^T C^I \hat{b}_0^I + [(x+1)^2, \dots, (x+1)^r] C^I \hat{b}_0^I, \\
(\sigma_{nl,nr}^r)_l &= [(x-1) \mathbf{1}_{r-1}^T C^D + [(x-1)^2, \dots, (x-1)^r] C^D] e_l^D, \\
&\quad \text{para } l = 2, \dots, nl + nr - 1, \\
(\sigma_{nl,nr}^r)_l &= [(x+1) \mathbf{1}_{r-1}^T C^I - [(x+1)^2, \dots, (x+1)^r] C^I] e_l^I, \\
&\quad \text{para } l = -nl - nr, \dots, -1,
\end{aligned}$$

donde $e_j^D = (\delta_{j,k})_{k=2}^{nl+nr-1}$ y $e_j^I = (\delta_{j,k})_{k=-nl-nr}^{-1}$.

Como ejemplo, los filtros para $nl + nr = 6$ y $r = 3$ son:

$$\begin{aligned}
G_i^{\overline{\mathcal{IAP}}}(x_{i-1}^{k-1}) &= -f_{i-1}^{k-1} + \frac{1063}{335} f_i^{k-1} - \frac{2856}{953} f_{i+1}^{k-1} + \frac{244}{953} f_{i+2}^{k-1} \\
&\quad + \frac{459}{953} f_{i+3}^{k-1} + \frac{372}{953} f_{i+4}^{k-1} - \frac{290}{953} f_{i+5}^{k-1}, \\
G_i^{\overline{\mathcal{IAP}}}(x_{2i-1}^k) &= \frac{470}{1207} f_{i-6}^{k-1} - \frac{3395}{6147} f_{i-5}^{k-1} - \frac{1727}{2581} f_{i-4}^{k-1} - \frac{918}{2593} f_{i-3}^{k-1} \\
&\quad + \frac{1245}{358} f_{i-2}^{k-1} - \frac{1919}{583} f_{i-1}^{k-1} + \frac{1044}{545} f_i^{k-1} - \frac{626}{521} f_{i+1}^{k-1} \\
&\quad + \frac{388}{4425} f_{i+2}^{k-1} + \frac{94}{569} f_{i+3}^{k-1} + \frac{314}{2331} f_{i+4}^{k-1} - \frac{326}{3207} f_{i+5}^{k-1}, \\
G_i^{\overline{\mathcal{IAP}}}(x_i^{k-1}) &= \frac{1650}{953} f_{i-6}^{k-1} - \frac{2248}{953} f_{i-5}^{k-1} - \frac{1537}{953} f_{i-4}^{k-1} \\
&\quad - \frac{1454}{256} f_{i-3}^{k-1} + \frac{3447}{256} f_{i-2}^{k-1} - \frac{8990}{953} f_{i-1}^{k-1} + f_i^{k-1}.
\end{aligned}$$

Ejemplos

Aplicaremos las diferentes reconstrucciones a la función (1.44). El experimento consiste en bajar tres niveles por valores puntuales, eliminar los coeficientes de escala y subir tres niveles aplicando los diferentes operadores reconstrucción. En primer lugar utilizamos la reconstrucción $\overline{\mathcal{IAP}}$ con *stencils* centrados (excepto en las fronteras que sí se han adaptado) de 6 puntos, es decir con $nl = nr = 3$. Como vemos en la Figura 2.2 (a) para cada discontinuidad hay 5 coeficientes de escala afectados, en concordancia con (2.13). En la Figura (b) representamos la posición de los $d_i^k \geq 10^{-3}$ para cada nivel (eje Y).

Al introducir la técnica *ENO* adaptada para esta reconstrucción obtenemos que el intervalo de escasa aproximación se reduce al que contiene la discontinuidad, Figura 2.2 (c). Lo mismo podemos decir de la técnica *WENO* y *PPH*, Figuras (e) y (f) respectivamente.

Finalmente, con la técnica *SR* la aproximación mejora en la esquina pero no en el salto, ya que en este caso la función $G_i^{\overline{\mathcal{IAP}}}(x)$ no cambia de signo en los extremos del intervalo que contiene la discontinuidad, Figura 2.2 (c).

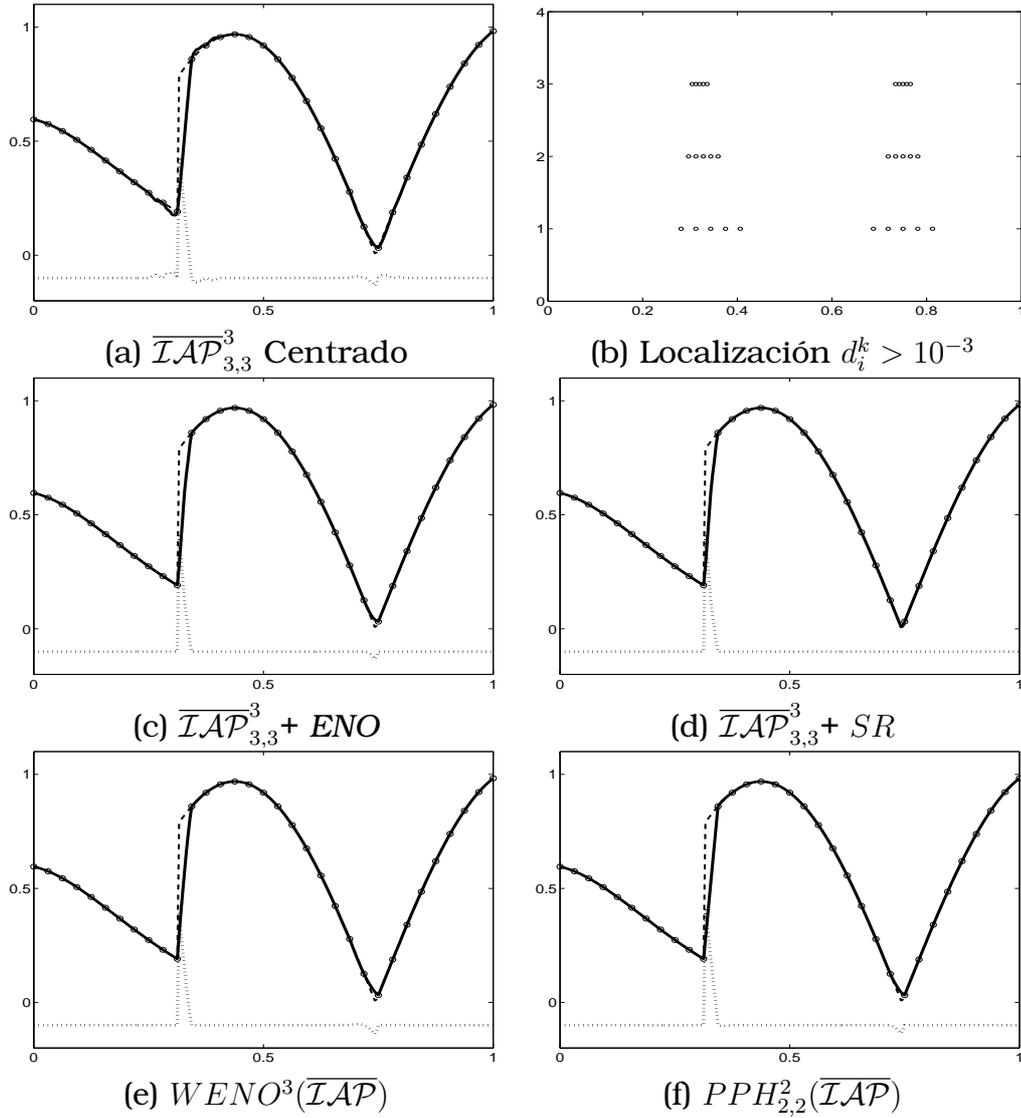


Figura 2.2: Aplicamos 3 niveles de decimación por valores puntuales y 3 niveles de predicción a la función (1.44) discretizada con 257 nodos. En (b) 'o' representa, para cada nivel, la localización de los coeficientes $d_i^k > 10^{-3}$. En el resto, '-' representa la función de partida, '-.' la reconstrucción, 'o' la función en el nivel más grosero y '..' la diferencia entre la discretización inicial y la reconstrucción (desplazada $-0,1$ en el eje OY).

2.2.2

Interpolación aproximación para medias en celda ($\overline{\mathcal{IAC}}$)

Trabajando con la discretización por medias en celda, definimos la reconstrucción **interpolación aproximación para medias en celda** y la denotamos por $\overline{\mathcal{IAC}}$ de la siguiente manera:

$$(\overline{\mathcal{IAC}} f^k)(x) := \frac{d}{dx} \overline{\mathcal{IAP}}(x, F^k),$$

donde $\overline{\mathcal{IAP}}$ es la reconstrucción definida en la sección 2.2.1.

Dado que $\overline{\mathcal{IAP}}$ interpola a los valores de $F(x)$ en los extremos de cada intervalo:

$$\begin{aligned} (\mathcal{D}_k(\overline{\mathcal{IAC}} f^k)(x))_i &= \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} \frac{d}{dx} \overline{\mathcal{IAP}}(x, F^k) \\ &= \frac{\overline{\mathcal{IAP}}(x_i^k, F^k) - \overline{\mathcal{IAP}}(x_{i-1}^k, F^k)}{h_k} = \frac{F_i^k - F_{i-1}^k}{h_k} = \bar{f}_i^k. \end{aligned}$$

y por tanto es un operador reconstrucción (Definición 1.7).

El operador predicción se calcula como sigue:

$$\begin{aligned} (P_{k-1}^k f^{k-1})_{2i-1} &= (\mathcal{D}_k \overline{\mathcal{IAC}} f^{k-1})_{2i-1} = \frac{1}{h_k} \int_{x_{2i-2}^k}^{x_{2i-1}^k} \frac{d}{dx} \overline{\mathcal{IAP}}(x; F^{k-1}) \\ &= \frac{1}{h_k} (\overline{\mathcal{IAP}}(x_{2i-1}^k, F^{k-1}) - \overline{\mathcal{IAP}}(x_{2i-2}^k, F^{k-1})) \\ &= \frac{1}{h_k} (\overline{\mathcal{IAP}}(x_{2i-1}^k, F^{k-1}) - F_{i-1}^{k-1}) \equiv \overline{\mathcal{IAC}}(x_{2i-1}^k, f^{k-1}). \end{aligned}$$

Cálculo de la reconstrucción $\overline{\mathcal{IAC}}$

Fijado k , buscamos un polinomio de grado r , $q(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$, cuya media interpole a la media en celda \bar{f}_i^{k-1} y aproxime en un cierto número de nodos adyacentes. Dicho polinomio puede verse como el de *interpolación aproximación* para valores puntuales a la función $F^{k-1}(x)$. Sin pérdida de generalidad, expresamos el

polinomio sobre números enteros para facilitar la exposición. Las condiciones de interpolación requeridas se traducen en:

$$\frac{q(1) - q(0)}{1} = \sum_{l=2}^{r+1} a_l = \bar{f}_i^{k-1}.$$

Dado que $a_2 = \bar{f}_i^{k-1} - \sum_{l=3}^{r+1} a_l$, entonces:

$$q(x) = a_1 + (\bar{f}_i^{k-1} - \sum_{l=3}^{r+1} a_l)x + \sum_{l=3}^{r+1} a_l x^{l-1} = a_1 + x\bar{f}_i^{k-1} + \sum_{l=3}^{r+1} a_l (x^{l-1} - x).$$

Calculemos las medias en celda de la función en $[j, j + 1]$:

$$\begin{aligned} \frac{q(j+1) - q(j)}{1} &= a_1 + (j+1)\bar{f}_i^{k-1} + \sum_{l=3}^{r+1} a_l ((j+1)^{l-1} - j - 1) \\ &\quad - a_1 - j\bar{f}_i^{k-1} - \sum_{l=3}^{r+1} a_l (j^{l-1} - j) \\ &= \bar{f}_i^{k-1} + \sum_{l=3}^{r+1} a_l ((j+1)^{l-1} - j^{l-1} - 1). \end{aligned} \quad (2.19)$$

Imponemos nl condiciones a la izquierda y nr a la derecha de \bar{f}_i^{k-1} , con $nl + nr \geq r$ para asegurarnos que aproximemos y no interpolamos. Las ecuaciones de mínimos cuadrados son:

$$\sum_{l=3}^{r+1} a_l ((j+1)^{l-1} - j^{l-1} - 1) = \bar{f}_{i+j}^{k-1} - \bar{f}_i^{k-1}, \quad (2.20)$$

para $j = -nl, -nl + 1, \dots, -1, 1, 2, \dots, nr$.

En forma matricial tenemos $A_{nl,nr}^{\overline{\mathcal{IAC}}} \hat{a} = \hat{f} - \mathbf{1}_{nl+nr} \bar{f}_i$, donde:

$$A_{nl,nr}^{\overline{\mathcal{IAC}}} = \begin{bmatrix} (-nl+1)^2 - (-nl)^2 - 1 & \cdots & (-nl+1)^r - (-nl)^r - 1 \\ \vdots & \ddots & \vdots \\ -2 & \cdots & -(-1)^r - 1 \\ 2 & \cdots & 2^r - 2 \\ 4 & \cdots & 3^r - 2^r - 1 \\ \vdots & \ddots & \vdots \\ (nr+1)^2 - (nr)^2 - 1 & \cdots & (nr+1)^r - (nr)^r - 1 \end{bmatrix}, \quad (2.21)$$

con $\hat{a} = [a_3, \dots, a_{r+1}]^T$ y $\hat{f} = [\bar{f}_{i-nl}^{k-1}, \dots, \bar{f}_{i-1}^{k-1}, \bar{f}_{i+1}^{k-1}, \dots, \bar{f}_{i+nr}^{k-1}]^T$.

Para obtener los coeficientes $\{a_3, \dots, a_{r+1}\}$ del polinomio $q(x)$ resolvemos el sistema $A_{nl, nr}^{\overline{\mathcal{IAC}}}^T A_{nl, nr}^{\overline{\mathcal{IAC}}} \hat{a} = A_{nl, nr}^{\overline{\mathcal{IAC}}}^T (\hat{f} - \mathbf{1}_{nl+nr} \bar{f}_i)$:

$$\hat{a} = \left(A_{nl, nr}^{\overline{\mathcal{IAC}}}^T A_{nl, nr}^{\overline{\mathcal{IAC}}} \right)^{-1} A_{nl, nr}^{\overline{\mathcal{IAC}}}^T (\hat{f} - \mathbf{1}_{nl+nr} \bar{f}_i). \quad (2.22)$$

Definiendo $C = \left(A_{nl, nr}^{\overline{\mathcal{IAC}}}^T A_{nl, nr}^{\overline{\mathcal{IAC}}} \right)^{-1} A_{nl, nr}^{\overline{\mathcal{IAC}}}^T$, tenemos que:

$$\begin{aligned} a_2 &= \bar{f}_i - \mathbf{1}_{r-1}^T \hat{a} = \bar{f}_i - \mathbf{1}_{r-1}^T C \hat{f} + \mathbf{1}_{r-1}^T C \mathbf{1}_{nl+nr} \bar{f}_i \\ &= (1 + \mathbf{1}_{r-1}^T C \mathbf{1}_{nl+nr}) \bar{f}_i - \mathbf{1}_{r-1}^T C \hat{f}. \end{aligned}$$

NOTA 2.4. El coeficiente a_1 queda sin ser calculado, aunque esto no supone ningún problema, ya que, como finalmente buscamos las medias del polinomio $q(x)$, el coeficiente a_1 se elimina.

Al polinomio $q(x)$ que hemos calculado lo denotamos por:

$$p_i^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r) \equiv p_i^{\overline{\mathcal{IAC}}}(x; \bar{f}^{k-1}, r-1).$$

Sobre $\{x_i^{k-1}\}_i$ se denotará $q_i^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r) \equiv q_i^{\overline{\mathcal{IAC}}}(x; \bar{f}^{k-1}, r-1)$.

Una vez hallados $\{a_l\}_{l=3}^{r+1}$ el valor que aproxima a \bar{f}_{2i-1}^k será:

$$\begin{aligned} \bar{f}_{2i-1}^k &= \frac{p_i^{\overline{\mathcal{IAP}}}(\frac{1}{2}; F^{k-1}, r) - p_i^{\overline{\mathcal{IAP}}}(0; F^{k-1}, r)}{1/2} = 2 \left(\sum_{l=1}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1} - a_1 \right) \\ &= \sum_{l=2}^{r+1} a_l \left(\frac{1}{2}\right)^{l-2} = a_2 + \left(\frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T \hat{a} \\ &= (1 + \mathbf{1}_{r-1}^T C \mathbf{1}_{nl+nr}) \bar{f}_i - \mathbf{1}_{r-1}^T C \hat{f} \\ &\quad + \left(\frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T C \hat{f} - \left(\frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T C \mathbf{1}_{nl+nr} \bar{f}_i \\ &= (1 + (1 - \frac{1}{2^1}, \dots, 1 - \frac{1}{2^{r-1}})^T C \mathbf{1}_{nl+nr}) \bar{f}_i + \left(\frac{1}{2^1} - 1, \dots, \frac{1}{2^{r-1}} - 1\right)^T C \hat{f} \\ &= \sum_{l=-nl}^{nr} (\gamma_{nl, nr}^r)_l \bar{f}_{i+l} \equiv \overline{\mathcal{IAC}}_{nl, nr}^{r-1}(x_{2i-1}^k, \bar{f}^{k-1}), \end{aligned}$$

siendo² (ver la Tabla 2.4):

$$\begin{aligned}
 (\gamma_{nl,nr}^r)_0 &= 1 + \left(1 - \frac{1}{2^1}, \dots, 1 - \frac{1}{2^{r-1}}\right)^T C \mathbf{1}_{nl+nr}, \\
 (\gamma_{nl,nr}^r)_l &= \left(\frac{1}{2^1} - 1, \dots, \frac{1}{2^{r-1}} - 1\right)^T C e_{l+nl+1}, \quad l = -nl, \dots, -1, \\
 (\gamma_{nl,nr}^r)_l &= \left(\frac{1}{2^1} - 1, \dots, \frac{1}{2^{r-1}} - 1\right)^T C e_{nl+l}, \quad l = 1, \dots, nr,
 \end{aligned}$$

donde $e_j = (\delta_{j,k})_{k=1}^{nl+nr}$.

| l | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| $(\gamma_{2,0}^2)_l$ | | | $\frac{2}{20}$ | $\frac{1}{20}$ | $\frac{17}{20}$ | | | | |
| $(\gamma_{3,0}^2)_l$ | | $\frac{3}{56}$ | $\frac{2}{56}$ | $\frac{1}{56}$ | $\frac{50}{56}$ | | | | |
| $(\gamma_{4,0}^2)_l$ | $\frac{4}{120}$ | $\frac{3}{120}$ | $\frac{2}{120}$ | $\frac{1}{120}$ | $\frac{110}{120}$ | | | | |
| $(\gamma_{0,2}^2)_l$ | | | | | $\frac{23}{20}$ | $-\frac{1}{20}$ | $-\frac{2}{20}$ | | |
| $(\gamma_{0,3}^2)_l$ | | | | | $\frac{62}{56}$ | $-\frac{1}{56}$ | $-\frac{2}{56}$ | $-\frac{3}{56}$ | |
| $(\gamma_{0,4}^2)_l$ | | | | | $\frac{130}{120}$ | $-\frac{1}{120}$ | $-\frac{2}{120}$ | $-\frac{3}{120}$ | $-\frac{4}{120}$ |
| $(\gamma_{2,2}^3)_l$ | | | $\frac{2}{40}$ | $\frac{1}{40}$ | $\frac{40}{40}$ | $-\frac{1}{40}$ | $-\frac{2}{40}$ | | |
| $(\gamma_{3,3}^3)_l$ | | $\frac{3}{112}$ | $\frac{2}{112}$ | $\frac{1}{112}$ | $\frac{112}{112}$ | $-\frac{1}{112}$ | $-\frac{2}{112}$ | $-\frac{3}{112}$ | |
| $(\gamma_{4,4}^3)_l$ | $\frac{4}{240}$ | $\frac{3}{240}$ | $\frac{2}{240}$ | $\frac{1}{240}$ | $\frac{240}{240}$ | $-\frac{1}{240}$ | $-\frac{2}{240}$ | $-\frac{3}{240}$ | $-\frac{4}{240}$ |
| \bar{f}_{i+l}^{k-1} | \bar{f}_{i-4}^{k-1} | \bar{f}_{i-3}^{k-1} | \bar{f}_{i-2}^{k-1} | \bar{f}_{i-1}^{k-1} | \bar{f}_i^{k-1} | \bar{f}_{i+1}^{k-1} | \bar{f}_{i+2}^{k-1} | \bar{f}_{i+3}^{k-1} | \bar{f}_{i+4}^{k-1} |

Tabla 2.4: Aproximaciones a \bar{f}_{2i-1}^k , con grado del polinomio $r - 1$ (r para F), $\overline{\mathcal{IAC}}_{nl,nr}^r(x_{2i-1}^k, \bar{f}^{k-1}) = \sum_{l=-nl}^{nr-1} (\gamma_{nl,nr}^r)_l \bar{f}_{i+l}^{k-1}$, interpolando en \bar{f}_i^{k-1} y aproximando con nl nodos a la izquierda y nr nodos a la derecha de \bar{f}_i^{k-1} .

Si $nl = nr$ el stencil es centrado, pero en caso contrario se tomarían más nodos a un lado que al otro, hecho que utilizaremos para adaptar la técnica ENO al caso que nos ocupa.

Orden de la reconstrucción $\overline{\mathcal{IAC}}$

Siendo $F(x) = p(x) = \sum_{l=1}^{s+1} b_l x^{l-1}$ un polinomio de grado s , veremos si puede ser recuperado de forma exacta mediante la reconstrucción interpolación aproximación por medias en celda. Tomemos

²En <http://www.uv.es/anims/noguera/> se pueden descargar programas y una GUI en MATLAB® que calculan dichos filtros según los datos introducidos por el usuario.

$p^{\overline{\mathcal{IAC}}}(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$ como el polinomio de *interpolación aproximación* de grado r construido a partir de $F(x)$.

En primer lugar estudiamos el caso $s = r$, debiéndose cumplir

$$\frac{p^{\overline{\mathcal{IAC}}}(j+1) - p^{\overline{\mathcal{IAC}}}(j)}{1} = \frac{p(j+1) - p(j)}{1}, \quad j = -nl, \dots, nr,$$

o desarrollando:

$$\sum_{l=1}^{r+1} a_l ((j+1)^{l-1} - j^{l-1}) = \sum_{l=1}^{r+1} b_l ((j+1)^{l-1} - j^{l-1}), \quad j = -nl, \dots, nr,$$

siendo $p^{\overline{\mathcal{IAC}}}(1) - p^{\overline{\mathcal{IAC}}}(0) = \sum_{l=2}^{r+1} a_l = \sum_{l=2}^{r+1} b_l$ la condición interpolatoria.

En forma matricial:

$$M_n \begin{bmatrix} a_1 \\ \vdots \\ a_{r+1} \end{bmatrix} = M_n \begin{bmatrix} b_1 \\ \vdots \\ b_{r+1} \end{bmatrix},$$

con

$$M_n = \begin{bmatrix} 0 & 1 & (-nl+1)^2 - (-nl)^2 & \cdots & (-nl+1)^{r+1} - (-nl)^{r+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & -1 & \cdots & -(-1)^{r+1} \\ 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 3 & \cdots & 2^{r+1} - 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & (nr+1)^2 - nr^2 & \cdots & (nr+1)^{r+1} - nr^{r+1} \end{bmatrix}.$$

Observamos que la primera columna es nula lo cual indica que el coeficiente b_1 no puede ser determinado. Esto no representa ningún problema, ya que como lo que vamos a calcular es $p^{\overline{\mathcal{IAC}}}(j+1) - p^{\overline{\mathcal{IAC}}}(j)$, dicho coeficiente se elimina.

Si trabajamos con la matriz

$$M'_n = \begin{bmatrix} 1 & (-nl+1)^2 - (-nl)^2 & \cdots & (-nl+1)^{r+1} - (-nl)^{r+1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & -1 & \cdots & -(-1)^{r+1} \\ 1 & 1 & \cdots & 1 \\ 1 & 3 & \cdots & 2^{r+1} - 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (nr+1)^2 - nr^2 & \cdots & (nr+1)^{r+1} - nr^{r+1} \end{bmatrix},$$

el sistema

$$M'_n \begin{bmatrix} a_2 \\ \vdots \\ a_{r+1} \end{bmatrix} = M'_n \begin{bmatrix} b_2 \\ \vdots \\ b_{r+1} \end{bmatrix},$$

tiene solución única al igual que vimos en la Proposición 2.1. El resto de casos ($r \neq s$) también fueron estudiados en dicha proposición.

Dado que $\bar{f}(x)$ es un grado menor que $F(x)$, podemos enunciar los siguientes resultados:

PROPOSICIÓN 2.2. Si $q_i^{\overline{\mathcal{IAC}}}(x; \bar{f}^{k-1}, r-1)$ es el polinomio de interpolación aproximación para medias en celda de grado $r-1$ (de grado r para $q_i^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r)$), la reconstrucción $\overline{\mathcal{IAC}}$ recupera de forma exacta polinomios de grado menor o igual a $r-1$.

COROLARIO 2.2. El orden de la reconstrucción $\overline{\mathcal{IAC}}$ es r .

ENO

Utilizando *stencils* centrados obtenemos coeficientes de escala de elevada magnitud. Con $2n+1$ nodos, si $nl = nr = n$ y en (x_{2i-1}^k, x_{2i}^k) hay una discontinuidad, los coeficientes de escala afectados son

$$d_l^k, \quad i-n \leq l \leq i+n, \quad (2.23)$$

mientras que si la discontinuidad se halla en (x_{2i-2}^k, x_{2i-1}^k) son

$$d_l^k, \quad i-n \leq l \leq i+n-1. \quad (2.24)$$

Para minimizar el número de coeficientes de escala afectados por discontinuidades debemos utilizar la técnica *ENO*. La elección del *stencil* se realiza con el Algoritmo 1.3 o 1.4 (*ENO-MC-I* o *ENO-MC-II*) para medias en celda, vistos en la sección 1.3.1, obteniendo para cada i el valor s_i , que en términos de nl será:

$$nl_i = i - s_i. \quad (2.25)$$

WENO

Partiendo de lo expuesto en la sección 1.3.3, veremos si es posible adaptar *WENO* a la reconstrucción $\overline{\mathcal{IAC}}$. Los pasos son:

1.- Expresar la aproximación de orden r como una combinación lineal de las $n + 1$ aproximaciones de orden s :

$$p_i^{\overline{\mathcal{IAC}}^{r-1}} = \sum_{k=0}^n C_k^s p_i^{\overline{\mathcal{IAC}}^{s-1}}, \quad (2.26)$$

con $C_k^s \geq 0 \forall k$ y $\sum_{k=0}^n C_k^s = 1$, siendo $p_i^{\overline{\mathcal{IAC}}^{t-1}}$ el polinomio de orden t de *interpolación aproximación* para medias en celda construido tomando nl nodos a la izquierda y nr nodos a la derecha de \bar{f}_i^{k-1} .

En (2.26) se debe cumplir:

- $r > s$ para que aumente el orden de aproximación.
- $2n \geq r$ para que en $p_i^{\overline{\mathcal{IAC}}^{r-1}}$ se aproxime y no sólo se interpole.
- $n \geq s$ para que en $p_i^{\overline{\mathcal{IAC}}^{s-1}}$ se aproxime y no únicamente se interpole.

Es fácil comprobar que los únicos casos que cumplen las condiciones deseadas son:

$$p_i^{\overline{\mathcal{IAC}}^{2n}} = \frac{1}{2} p_i^{\overline{\mathcal{IAC}}^{2n,0}} + \frac{1}{2} p_i^{\overline{\mathcal{IAC}}^{2n,n}}, \quad n \geq 2,$$

es decir:

$$\sum_{l=-n}^{n-1} (\beta_{n,n}^3)_l \bar{f}_{i+l}^{k-1} = \frac{1}{2} \sum_{l=-n}^0 (\beta_{n,0}^2)_l \bar{f}_{i+l}^{k-1} + \frac{1}{2} \sum_{l=-1}^{n-1} (\beta_{0,n}^2)_l \bar{f}_{i+l}^{k-1}, \quad n \geq 2. \quad (2.27)$$

2.- Definimos $\alpha_k^s = \frac{C_k^s}{(\varepsilon + \overline{\mathcal{I}S}_{n-k,k})^2}$, $k = 0, \dots, n$, con $\overline{\mathcal{I}S}_{n-k,k}$ un indicador de suavidad de $\bar{f}(x)$ en el *stencil* $\{x_{i-n+k}^{k-1}, \dots, x_{i+k}^{k-1}\}$, como el de Liu et al. (1.63) o el de Jiang y Shu (1.64) de las reconstrucciones interpolatorias.

3.- La aproximación *WENO* será

$$p_i^{WENO-\overline{\mathcal{IAC}}}(x) = \sum_{k=0}^n \omega_k^s p_i^{\overline{\mathcal{IAC}}_{n-k,k}^{s-1}}(x),$$

donde $\omega_k^s = \frac{\alpha_k^s}{\sum_{j=0}^n \alpha_j^s}$, para $k = 0, \dots, n$.

PPH

Los coeficientes de la reconstrucción $\overline{\mathcal{IAP}}$ son los mismos que para $\overline{\mathcal{IP}}$, es decir:

$$\overline{\mathcal{IAP}}_{2,2}^2(x_{2i-1}^k, \bar{f}^{k-1}) = -\frac{1}{16}\bar{f}_{i-2}^{k-1} + \frac{9}{16}\bar{f}_{i-1}^{k-1} + \frac{9}{16}\bar{f}_i^{k-1} - \frac{1}{16}\bar{f}_{i+1}^{k-1}. \quad (2.28)$$

Siendo ésta la única elección posible que cumpla la condición de aproximación con *stencils* de 4 nodos. Repitiendo el razonamiento expuesto para $\overline{\mathcal{IC}}$ en la sección 1.3.4, obtenemos

$$\left(\overline{\mathcal{IAC}}_{1,1}^1\right)^{PPH}(x_{2i-1}^k, \bar{f}^{k-1}) = \bar{f}_i^{k-1} - \frac{1}{4}H(d^1 \bar{f}_i^{k-1}, d^1 \bar{f}_i^{k-1}). \quad (2.29)$$

donde $H(x, y)$ es la media armónica definida en (1.30).

SR

Los pasos para aplicar *SR* para la reconstrucción $\overline{\mathcal{IAC}}$ son:

1.- Calculamos los *stencils ENO* mediante el Algoritmo 1.3 o 1.4 (*ENO-MC-I* o *ENO-MC-II*) y asignamos los nodos donde interpolaremos para cada intervalo según (2.25).

2.- Si $nr_{i-1} = 0$ y $nl_{i+1} = 0$ los *stencils* para las celdas I_{i-1}^{k-1} y I_{i+1}^{k-1} son disjuntos. Etiquetamos dicha celda como *sospechosa* de contener una discontinuidad.

3.- Para cada celda sospechosa definimos la función

$$G_i^{\overline{\mathcal{IAC}}}(x) = q_{s_{i+1}}^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r) - q_{s_{i-1}}^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r). \quad (2.30)$$

NOTA 2.5. En (2.30) el polinomio $q_{s_{i+1}}^{\overline{IAP}}(x; F^{k-1}, r)$ lo calculamos interpolando en \bar{f}_{i+1}^{k-1} y aproximando en los nodos a la derecha de dicho punto y para $q_{s_{i-1}}^{\overline{IAC}}(x)$ interpolamos en \bar{f}_{i-1}^{k-1} y aproximamos en nodos a la izquierda. Aunque en este intervalo no realizamos interpolación en \bar{f}_i^{k-1} , el interpolar en \bar{f}_{i-1}^{k-1} y \bar{f}_{i+1}^{k-1} nos permite expresar $G_i^{\overline{IAC}}$ en función de $\{\bar{f}_i^{k-1}\}_i$

Si $G_i^{\overline{IAC}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{IAC}}(x_i^{k-1}) < 0$ etiquetamos I_i^{k-1} como singular.

4.- Si $G_i^{\overline{IAC}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{IAC}}(x_{2i-1}^k) < 0$, el punto x_{2i-1}^k se halla a la derecha de la discontinuidad, por tanto:

$$\bar{f}_{2i}^k = \frac{q_{s_{i+1}}^{\overline{IAP}}(x_{2i}^k; F^{k-1}, r) - q_{s_{i+1}}^{\overline{IAP}}(x_{2i-1}^k; F^{k-1}, r)}{h_k} \equiv \frac{p_{s_{i+1}}^{\overline{IAP}}(0; F^{k-1}, r) - p_{s_{i+1}}^{\overline{IAP}}(-1/2; F^{k-1}, r)}{1/2}.$$

Para asegurar consistencia $\bar{f}_{2i-1}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i}^k$.

En otro caso x_{2i-1}^k se halla a la izquierda de la discontinuidad, con lo cual:

$$\bar{f}_{2i-1}^k = \frac{q_{s_{i-1}}^{\overline{IAP}}(x_{2i-1}^k; F^{k-1}, r) - q_{s_{i-1}}^{\overline{IAP}}(x_{2i}^k; F^{k-1}, r)}{h_k} \equiv \frac{p_{s_{i-1}}^{\overline{IAP}}(3/2; F^{k-1}, r) - p_{s_{i-1}}^{\overline{IAP}}(1; F^{k-1}, r)}{1/2}.$$

Para asegurar consistencia $\bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k$.

NOTA 2.6. Gracias a las condiciones de interpolación en \bar{f}_{i+1}^{k-1} de $q_{s_{i+1}}^{\overline{IAP}}(x; F^{k-1}, r)$ y en \bar{f}_{i-1}^{k-1} de $q_{s_{i-1}}^{\overline{IAP}}(x; F^{k-1}, r)$ se puede aplicar el mismo razonamiento expuesto para la reconstrucción \overline{IC} en la sección 1.3.5, con lo que la consistencia queda garantizada.

A continuación expondremos el cálculo detallado de filtros para la realización de la técnica SR de la reconstrucción *interpolación aproximación*. Debemos tener en cuenta que hay una diferencia importante en la definición de $A_{nl, nr}^{\overline{IC}}$ y de $A_{nl, nr}^{\overline{IAC}}$. En el primer caso, nl y nr admiten valores negativos, mientras que en el segundo caso

no. Por esto, para adaptar la Proposición 1.7 a $\overline{\mathcal{IAC}}$ no podemos simplemente cambiar la matriz $A_{nl,nr}^{\overline{\mathcal{IC}}}$ por $A_{nl,nr}^{\overline{\mathcal{IAC}}}$ sino que deberemos introducir modificaciones adicionales.

Con el objetivo de reducir la exposición, en lo que resta del apartado asumiremos la siguiente notación:

$$\begin{aligned} C^D &= \left(A_{0,nl+nr}^{\overline{\mathcal{IAC}}} \quad A_{0,nl+nr}^{\overline{\mathcal{IAC}}} \right)^{-1} A_{0,nl+nr}^{\overline{\mathcal{IAC}}} \quad T, \\ \hat{f}_D &= [\bar{f}_{i+2}^{k-1}, \dots, \bar{f}_{i+nl+nr+1}^{k-1}]^T, \\ C^I &= \left(A_{nl+nr,0}^{\overline{\mathcal{IAC}}} \quad A_{nl+nr,0}^{\overline{\mathcal{IAC}}} \right)^{-1} A_{nl+nr,0}^{\overline{\mathcal{IAC}}} \quad T, \\ \hat{f}_I &= [\bar{f}_{i-nl-nr-1}^{k-1}, \dots, \bar{f}_{i-2}^{k-1}]^T. \end{aligned}$$

Para el cálculo de filtros de $G_i^{\overline{\mathcal{IAC}}}(x)$ utilizaremos el resultado:

PROPOSICIÓN 2.3. *Sea $A_{nl,nr}^{\overline{\mathcal{IAC}}}$ la matriz asociada al sistema que resuelve el polinomio de interpolación aproximación para medias en celda, con los nodos expresados sobre enteros (definida en 2.21) y sea $(F_{i-1}^{k-1}, F_i^{k-1})$ la localización de la discontinuidad, la función $G_i^{\overline{\mathcal{IAC}}}$ para grado r puede ser expresada como sigue:*

$$\begin{aligned} G_i^{\overline{\mathcal{IAC}}}(x) &= \bar{f}_i^{k-1} + (x-1) \left((1 + \mathbf{1}_{r-1}^T C^D \mathbf{1}_{nl+nr}) \bar{f}_{i+1}^{k-1} - \mathbf{1}_r^T C^D \hat{f}_D \right) \\ &\quad + [(x-1)^2, \dots, (x-1)^{r+1}] C^D \left(\hat{f}_D - \mathbf{1}_{nl+nr} \bar{f}_{i+1}^{k-1} \right) \\ &\quad - x \left((1 + \mathbf{1}_{r-1}^T C^I \mathbf{1}_{nl+nr}) \bar{f}_{i-1}^{k-1} - \mathbf{1}_{r-1}^T C^I \hat{f}_I \right) \\ &\quad + [(x+1)^2 - 1, \dots, (x+1)^r - 1] C^I \left(\hat{f}_I - \mathbf{1}_{nl+nr} \bar{f}_{i-1}^{k-1} \right). \end{aligned}$$

Demostración: Tomamos los polinomios de interpolación aproximación para medias en celda construidos a izquierda y derecha de la celda singular:

$$p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r) = \sum_{l=1}^{r+1} a_l x^{l-1} \quad \text{y} \quad p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r) = \sum_{l=1}^{r+1} b_l x^{l-1}.$$

Gracias a la condición de interpolación, se cumple que:

$$p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(0; F^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(1; F^{k-1}, r) = \frac{F_i^{k-1} - F_{i-1}^{k-1}}{1} = \bar{f}_i^{k-1}. \quad (2.31)$$

Por tanto:

$$a_1 - \sum_{l=1}^{r+1} b_l = \bar{f}_i^{k-1} \implies a_1 - b_1 = \bar{f}_i^{k-1} + \sum_{l=2}^{r+1} b_l. \quad (2.32)$$

Tal como hemos definido la matriz $A_{nl, nr}^{\overline{\mathcal{IAC}}}$, necesitamos utilizar un juego de nodos enteros distintos al trabajar con $p_{s_{i-1}}^{\overline{\mathcal{IAP}}}$ y con $p_{s_{i+1}}^{\overline{\mathcal{IAP}}}$. La Figura 2.3 muestra las elecciones de nodos enteros para el cálculo de cada polinomio de *interpolación aproximación*. Para la función $G_i^{\overline{\mathcal{IAC}}}(x)$ necesitamos una expresión que escoja correctamente los nodos según cada polinomio. Si observamos la Figura 2.3 es sencillo escribir las siguientes expresiones:

$$\begin{aligned} G_i^{\overline{\mathcal{IC}}}(x_{i-1}^{k-1}) &\rightarrow p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(-1; F^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(1; F^{k-1}, r), \\ G_i^{\overline{\mathcal{IC}}}(x_{2i-1}^{k-1}) &\rightarrow p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(-1/2; F^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(3/2; F^{k-1}, r), \\ G_i^{\overline{\mathcal{IC}}}(x_i^{k-1}) &\rightarrow p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(0; F^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(2; F^{k-1}, r). \end{aligned}$$

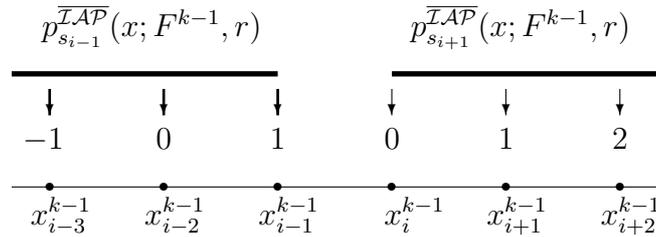


Figura 2.3: Elecciones de nodos enteros necesarios para la correcta construcción de los polinomios $p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r)$ y $p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(x; F^{k-1}, r)$.

Con esto, la función $G_i^{\overline{\mathcal{IAC}}}(x)$ puede expresarse:

$$G_i^{\overline{\mathcal{IAC}}}(x) = p_{s_{i+1}}^{\overline{\mathcal{IAP}}}(x-1; F^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(x+1; F^{k-1}, r)$$

$$\begin{aligned}
&= \sum_{l=1}^{r+1} a_l (x-1)^{l-1} - \sum_{l=1}^{r+1} b_l (x+1)^{l-1} \\
&= a_1 - b_1 + \sum_{l=2}^{r+2} a_l (x-1)^{l-1} - \sum_{l=2}^{r+2} b_l (x+1)^{l-1} \\
&= \bar{f}_i^{k-1} + \sum_{l=2}^{r+1} b_l + \sum_{l=2}^{r+1} a_l (x-1)^{l-1} - \sum_{l=2}^{r+1} b_l (x+1)^{l-1} \\
&= \bar{f}_i^{k-1} + \sum_{l=2}^{r+1} a_l (x-1)^{l-1} - \sum_{l=2}^{r+1} b_l ((x+1)^{l-1} - 1) \\
&= \bar{f}_i^{k-1} + a_2(x-1) + \sum_{l=3}^{r+1} a_l (x-1)^{l-1} - b_2 x - \sum_{l=3}^{r+1} b_l ((x+1)^{l-1} - 1) \\
&= \bar{f}_i^{k-1} + (x-1) \left((1 + \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^D \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}) \bar{f}_{i+1}^{k-1} - \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^D \hat{f}_D \right) \\
&\quad + [(x-1)^2, \dots, (x-1)^r] C^D \left(\hat{f}_D - \mathbf{1}_{\mathbf{nl}+\mathbf{nr}} \bar{f}_{i+1}^{k-1} \right) \\
&\quad - x \left((1 + \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^I \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}) \bar{f}_{i-1}^{k-1} - \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^I \hat{f}_I \right) \\
&\quad + [(x+1)^2 - 1, \dots, (x+1)^r - 1] C^I \left(\hat{f}_I - \mathbf{1}_{\mathbf{nl}+\mathbf{nr}} \bar{f}_{i-1}^{k-1} \right) \\
&= \sum_{l=-\mathbf{nl}-\mathbf{nr}-1}^{\mathbf{nl}+\mathbf{nr}+1} (\sigma_{\mathbf{nl},\mathbf{nr}}^r)_l \bar{f}_{i+l}^{k-1}. \tag{2.33}
\end{aligned}$$

siendo:

$$\begin{aligned}
(\sigma_{\mathbf{nl},\mathbf{nr}}^r)_l &= (x \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} + [(x+1)^2 - 1, \dots, (x+1)^r - 1]) C^I e_l^I, \\
&\quad \text{para } l = -\mathbf{nl} - \mathbf{nr} - 1, \dots, -2, \\
(\sigma_{\mathbf{nl},\mathbf{nr}}^r)_{-1} &= -x(1 + \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^I \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}) - [(x+1)^2 - 1, \dots, (x+1)^r - 1] C^I \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}, \\
(\sigma_{\mathbf{nl},\mathbf{nr}}^r)_0 &= 1, \\
(\sigma_{\mathbf{nl},\mathbf{nr}}^r)_1 &= (x-1)(1 + \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^D \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}) - [(x-1)^2, \dots, (x-1)^r] C^D \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}, \\
(\sigma_{\mathbf{nl},\mathbf{nr}}^r)_l &= (-(x-1) \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} + [(x-1)^2, \dots, (x-1)^r]) C^D e_l^D, \\
&\quad \text{para } l = 2, \dots, \mathbf{nl} + \mathbf{nr} + 1,
\end{aligned}$$

donde $e_j^D = (\delta_{j,k})_{k=2}^{\mathbf{nl}+\mathbf{nr}+1}$ y $e_j^I = (\delta_{j,k})_{k=-\mathbf{nl}-\mathbf{nr}-1}^2$.

□

Los valores que necesitamos para decidir si la celda es singular y para conocer la localización de x_{2i-1}^k respecto a la discontinuidad son $G_i^{\overline{\mathcal{IAC}}}(x_{i-1}^{k-1})$, $G_i^{\overline{\mathcal{IAC}}}(x_{2i-1}^k)$, $G_i^{\overline{\mathcal{IAC}}}(x_i^{k-1})$. Como trabajamos sobre nodos enteros debemos calcular:

$$\begin{aligned} G_i^{\overline{\mathcal{IAC}}}(x_{i-1}^{k-1}) &\rightarrow h_{k-1} G_i^{\overline{\mathcal{IAC}}}(0), \\ G_i^{\overline{\mathcal{IAC}}}(x_{2i-1}^k) &\rightarrow h_{k-1} G_i^{\overline{\mathcal{IAC}}}(1/2), \\ G_i^{\overline{\mathcal{IAC}}}(x_i^{k-1}) &\rightarrow h_{k-1} G_i^{\overline{\mathcal{IAC}}}(1), \end{aligned}$$

aunque como ya advertimos en la Nota 1.17, en la práctica el producto por h_{k-1} puede ser eliminado, ya que en la función G sólo nos interesa el signo, evitando así trabajar con los nodos iniciales.

Utilizando la Proposición 2.3, a modo de ejemplo exponemos los filtros para $nl + nr = 4$ y $r = 3$:

$$\left\{ \begin{array}{l} G_i^{\overline{\mathcal{IAC}}}(x_{i-1}^k) = h_{k-1} \left(\bar{f}_i^{k-1} - \frac{63}{31} \bar{f}_{i+1}^{k-1} + \frac{81}{155} \bar{f}_{i+2}^{k-1} + \frac{97}{155} \bar{f}_{i+3}^{k-1} \right. \\ \quad \left. + \frac{48}{155} \bar{f}_{i+4}^{k-1} - \frac{66}{155} \bar{f}_{i+5}^{k-1} \right), \\ G_i^{\overline{\mathcal{IAC}}}(x_{2i-1}^k) = h_{k-1} \left(-\frac{89}{620} \bar{f}_{i-5}^{k-1} + \frac{273}{2480} \bar{f}_{i-4}^{k-1} + \frac{271}{1240} \bar{f}_{i-3}^{k-1} + \frac{451}{2480} \bar{f}_{i-2}^{k-1} \right. \\ \quad \left. - \frac{215}{248} \bar{f}_{i-1}^{k-1} + \bar{f}_i^{k-1} - \frac{215}{248} \bar{f}_{i+1}^{k-1} + \frac{451}{2480} \bar{f}_{i+2}^{k-1} \right. \\ \quad \left. + \frac{271}{1240} \bar{f}_{i+3}^{k-1} + \frac{273}{2480} \bar{f}_{i+4}^{k-1} - \frac{89}{620} \bar{f}_{i+5}^{k-1} \right), \\ G_i^{\overline{\mathcal{IAC}}}(x_i^k) = h_{k-1} \left(-\frac{66}{155} \bar{f}_{i-5}^{k-1} + \frac{48}{155} \bar{f}_{i-4}^{k-1} + \frac{97}{155} \bar{f}_{i-3}^{k-1} + \frac{81}{155} \bar{f}_{i-2}^{k-1} \right. \\ \quad \left. - \frac{63}{31} \bar{f}_{i-1}^{k-1} + \bar{f}_i^{k-1} \right). \end{array} \right.$$

Pasemos al cálculo de filtros para el punto cuatro de SR. Dada una celda singular $[x_{i-1}^{k-1}, x_i^{k-1}]$, si $G_i^{\overline{\mathcal{IAC}}}(x_{2i-1}^k) \cdot G_i^{\overline{\mathcal{IAC}}}(x_i^{k-1}) < 0 \rightarrow \theta_i^{k-1} > x_{2i-1}^k$:

$$\begin{aligned} (P_{k-1}^k \bar{f}^{k-1})_{2i-1} &= \frac{p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(\frac{3}{2}; F^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{IAP}}}(1; F^{k-1}, r)}{1/2} \\ &= 2 \left(\sum_{l=1}^{r+1} b_l \left(\frac{3}{2}\right)^{l-1} - \sum_{l=1}^{r+1} b_l 1^{l-1} \right) \\ &= 2 \left(b_2 \left(\frac{3}{2} - 1\right) + \sum_{l=3}^{r+1} b_l \left(\left(\frac{3}{2}\right)^{l-1} - 1 \right) \right) \end{aligned}$$

$$\begin{aligned}
 &= (1 + \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^I \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}) \bar{f}_{i-1}^{k-1} - \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^I \hat{f}_I \\
 &\quad + 2 \left[\left(\frac{3}{2}\right)^2 - 1, \dots, \left(\frac{3}{2}\right)^r - 1 \right] C^I (\hat{f}_I - \mathbf{1}_{\mathbf{nl}+\mathbf{nr}} \bar{f}_{i-1}^{k-1}) \\
 &= \sum_{l=-\mathbf{nl}-\mathbf{nr}-1}^{-1} (\gamma_{\mathbf{nl}+\mathbf{nr}+1,-1}^r)_l \bar{f}_l^{k-1},
 \end{aligned}$$

donde:

$$(\gamma_{\mathbf{nl}+\mathbf{nr}+1,-1}^r)_{-1} = (1 + \mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^I \mathbf{1}_{\mathbf{nl}+\mathbf{nr}}) - 2 \left[\left(\frac{3}{2}\right)^2 - 1, \dots, \left(\frac{3}{2}\right)^r - 1 \right] C^I \mathbf{1}_{\mathbf{nl}+\mathbf{nr}},$$

$$(\gamma_{\mathbf{nl}+\mathbf{nr}+1,-1}^r)_l = -\mathbf{1}_{\mathbf{r}-1}^{\mathbf{T}} C^I + 2 \left[\left(\frac{3}{2}\right)^2 - 1, \dots, \left(\frac{3}{2}\right)^r - 1 \right] C^I e_{-l-\mathbf{nl}-\mathbf{nr}}.$$

siendo $e_{-l-\mathbf{nl}-\mathbf{nr}} = (\delta_{-l-\mathbf{nl}-\mathbf{nr},-k})_{k=1}^{\mathbf{nl}+\mathbf{nr}-2}$, para $l = -\mathbf{nl} - \mathbf{nr} - 1, \dots, -2$.

La notación $\gamma_{\mathbf{nl}+\mathbf{nr}+1,-1}^r$ no debe confundirnos. El subíndice -1 indica que estamos interpolando en \bar{f}_{i-1}^{k-1} y aproximando a la izquierda de este punto. Se utiliza esta notación por coherencia con expuesto para reconstrucciones interpolatorias. Los filtros resultantes pueden consultarse en la Tabla 2.5.

| l | -7 | -6 | -5 | -4 | -3 | -2 | -1 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| $(\gamma_{3,-1}^2)$ | | | | | $-\frac{3}{10}$ | $-\frac{3}{20}$ | $\frac{29}{20}$ |
| $(\gamma_{5,-1}^2)$ | | | $-\frac{1}{10}$ | $-\frac{3}{40}$ | $-\frac{1}{20}$ | $-\frac{1}{40}$ | $\frac{5}{4}$ |
| $(\gamma_{5,-1}^3)$ | | | $\frac{89}{310}$ | $-\frac{273}{1240}$ | $-\frac{271}{620}$ | $-\frac{451}{1240}$ | $\frac{215}{124}$ |
| $(\gamma_{7,-1}^3)$ | $\frac{81}{512}$ | $-\frac{65}{3584}$ | $-\frac{241}{1792}$ | $-\frac{171}{896}$ | $-\frac{671}{3584}$ | $-\frac{443}{3584}$ | $\frac{383}{256}$ |
| $(\gamma_{5,-1}^4)$ | | | $-\frac{971}{2208}$ | $\frac{1194}{985}$ | $-\frac{549}{1472}$ | $-\frac{2113}{1123}$ | $\frac{1318}{531}$ |
| $(\gamma_{7,-1}^4)$ | $-\frac{384}{1349}$ | $\frac{511}{1275}$ | $\frac{263}{821}$ | $-\frac{357}{2497}$ | $-\frac{454}{749}$ | $-\frac{1913}{2788}$ | $\frac{1663}{832}$ |
| \bar{f}_{i+l}^{k-1} | \bar{f}_{i-7}^{k-1} | \bar{f}_{i-6}^{k-1} | \bar{f}_{i-5}^{k-1} | \bar{f}_{i-4}^{k-1} | \bar{f}_{i-3}^{k-1} | \bar{f}_{i-2}^{k-1} | \bar{f}_{i-1}^{k-1} |

Tabla 2.5: Aproximaciones para \bar{f}_{2i-1}^{k-1} de grado r para \bar{f} ($r + 1$ para F), $\overline{\mathcal{IAC}}_{\mathbf{nl},\mathbf{nr}}^{r-1}(x_{2i-1}^k, \bar{f}^{k-1}) = \sum_{l=-\mathbf{nl}-\mathbf{nr}-1}^{-1} (\gamma_{\mathbf{nl}+\mathbf{nr}+1,-1}^r)_l \bar{f}_{i+l}^{k-1}$, utilizando interpolación en \bar{f}_i^{k-1} y aproximación en $\mathbf{nl} + \mathbf{nr}$ puntos a la izquierda de dicho punto.

En otro caso, $\theta_i^{k-1} < x_{2i-1}^k$:

$$(P_{k-1}^k \bar{f}^{k-1})_{2i} = \frac{p_{s_{i+1}}^{\overline{\mathcal{IAC}}}(0; F^{k-1}, r) - p_{s_{i+1}}^{\overline{\mathcal{IAC}}}(\frac{-1}{2}; F^{k-1}, r)}{1/2}$$

$$\begin{aligned}
&= -2\left(-\frac{1}{2}a_2 + \sum_{l=3}^{r+1} a_l \left(-\frac{1}{2}\right)^{l-1}\right) \\
&= (1 + \mathbf{1}_{r-1}^T C^D \mathbf{1}_{nl+nr}) \bar{f}_{i+1}^{k-1} - \mathbf{1}_{r-1}^T C^D \hat{f}_D \\
&\quad - 2\left[\left(-\frac{1}{2}\right)^2, \dots, \left(-\frac{1}{2}\right)^r\right] C^D (\hat{f}_D - \mathbf{1}_{nl+nr} \bar{f}_{i+1}^{k-1}) \\
&= \sum_{l=1}^{nl+nr+1} (\gamma_{-1, nl+nr+1}^r)_l \bar{f}_l^{k-1},
\end{aligned}$$

donde

$$\begin{aligned}
(\gamma_{-1, nl+nr+1}^r)_1 &= 1 + \mathbf{1}_{r-1}^T C^D \mathbf{1}_{nl+nr} + 2 \left[\left(-\frac{1}{2}\right)^2, \dots, \left(-\frac{1}{2}\right)^r \right] C^D \mathbf{1}_{nl+nr}, \\
(\gamma_{-1, nl+nr+1}^r)_l &= \left(-\mathbf{1}_{r-1}^T - 2 \left[\left(-\frac{1}{2}\right)^2, \dots, \left(-\frac{1}{2}\right)^r \right] \right) C^D e_l,
\end{aligned}$$

siendo $e_l = (\delta_{l,k})_{k=2}^{nl+nr+1}$, para $l = 2, \dots, nl + nr + 1$.

Se cumple (ver la Nota 2.2):

$$(\gamma_{-1, nl+nr+1}^r)_l = (\gamma_{nl+nr+1, -1}^r)_{-l}, \quad l = 1, \dots, nl + nr + 1.$$

El valor de \bar{f}_{2i-1}^k se puede hallar mediante $\bar{f}_{2i-1}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i}^k$.

Ejemplos

Los datos de partida corresponden a las medias en celda de una función, en concreto discretizamos la función (1.70) en los nodos centrales de 256 celdas equiespaciadas. Bajamos tres niveles por medias en celda, eliminamos los coeficientes de escala y reconstruimos 3 niveles por *interpolación aproximación*. En la Figura 2.4 (a) usamos la reconstrucción $\overline{\mathcal{IAC}}$ centrada con $nl = 3$, $nr = 3$ y $r = 3$. En (b) mostramos los d_i^k mayores que 10^{-3} . En el salto hay 7 afectados en concordancia con (2.23). Dichos coeficientes son los que provocan el *fenómeno de Gibbs* en la Figura (a). En la Figura 2.4 (c) utilizamos la técnica *ENO*. Observamos que se mejora la aproximación tanto en la esquina como en el salto, excepto en el intervalo que contiene la discontinuidad, al igual que con *WENO* y *PPH*, Figuras (e) y (f) respectivamente.

En la Figura 2.4 (d), aplicamos la técnica *SR* adaptada a la reconstrucción *interpolación aproximación*. Con dicha adaptación,

aunque la interpolación se produzca en las celdas contiguas a la discontinua y no en ella propia, observamos que la aproximación mejora, consiguiendo los objetivos de dicha técnica.

2.2.3

Funciones límite

Para aproximarnos de forma experimental a la función límite de cada reconstrucción, aplicamos predicción 6 veces a una función nula excepto en un punto.

En la Figura 2.5 partimos de una función con 16 nodos y se aplica predicción 6 niveles. Las funciones límite que se obtienen son suaves y por tanto adecuadas para los esquemas de multirresolución.

Para medias en celda, las funciones límite son más quebradas y por tanto inadecuadas para un esquema de multirresolución como comprobamos en la Figura 2.6.

2.2.4

Diseño de curvas

En este apartado utilizaremos las reconstrucciones de *interpolación aproximación* para el diseño de curvas. Para ello, partimos de una serie de puntos de control y aplicamos varios niveles de la reconstrucción. Tomamos 8 puntos de control $(-3, -5)$, $(-3, -3)$, $(1, 1)$, $(1, 3)$, $(-1, 3)$, $(-1, 1)$, $(3, -3)$, $(3, -5)$.

En la Figura 2.7 utilizamos \overline{IAP} , con diferentes valores para r , nl , nr y cinco niveles. En la Figura 2.8 aplicamos la reconstrucción a otros puntos de control iniciales para 2 dimensiones (derecha) y tres dimensiones (izquierda). Las curvas que obtenemos son suaves en todos los casos.

Para medias en celda, los resultados obtenidos, Figura 2.9, no son satisfactorios ya que se obtienen curvas de escasa suavidad, en consonancia con sus funciones límite vistas en el apartado anterior.

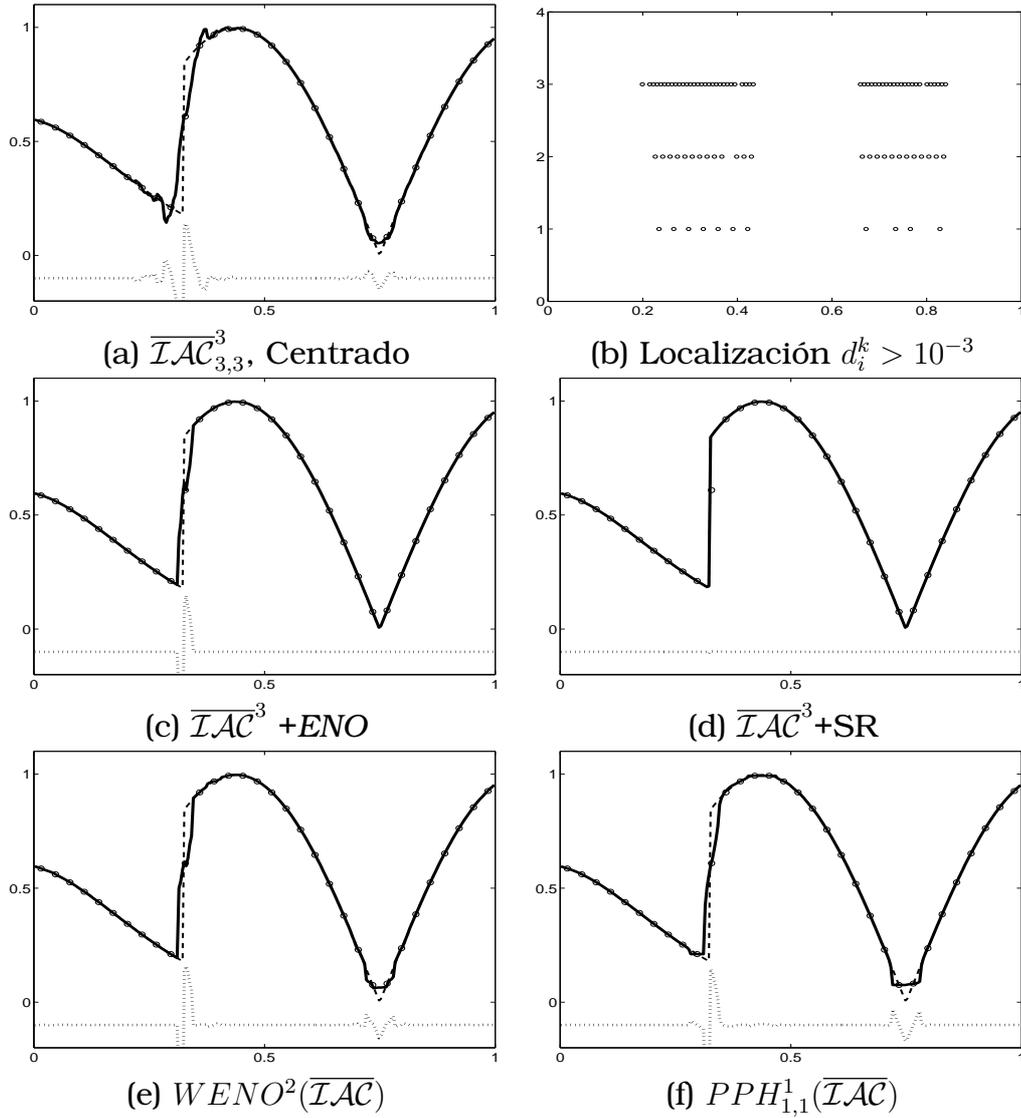


Figura 2.4: Aplicamos 3 niveles de decimación por medias en celda y 3 niveles de predicción a la función (1.70) discretizada con 256 nodos. En (b) 'o' representa, para cada nivel, la localización de los coeficientes $d_i^k > 10^{-3}$. En el resto, '-' representa la función de partida, '-.' la reconstrucción, 'o' la función en el nivel más grosero y '..' la diferencia entre la discretización inicial y la reconstrucción (desplazada $-0,1$ en el eje OY).

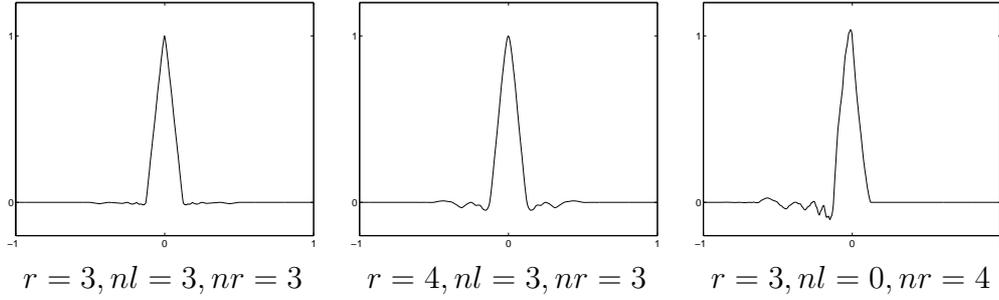


Figura 2.5: Función límite para la reconstrucción $\overline{\mathcal{IAP}}$. Utilizamos diferentes grados, r , y números de nodos donde aproximamos, nl, nr , para el polinomio $p_i^{\overline{\mathcal{IAP}}}(x; f^k, r)$.

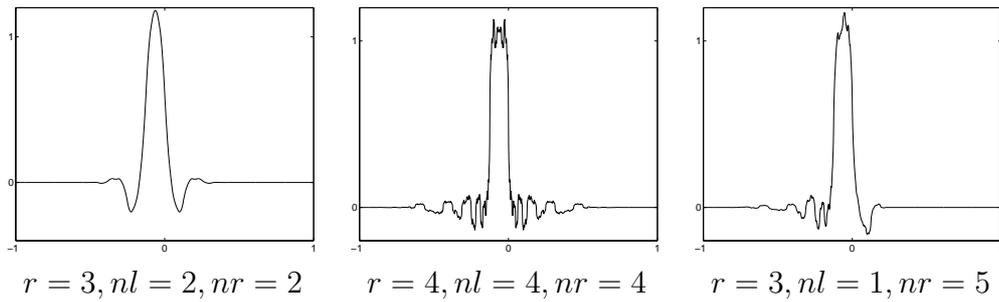


Figura 2.6: Función límite para la reconstrucción $\overline{\mathcal{IAC}}$. Utilizamos diferentes grados, r , y número de nodos, nl, nr , para el polinomio $p_i^{\overline{\mathcal{IAC}}}(x; f^k, r)$.

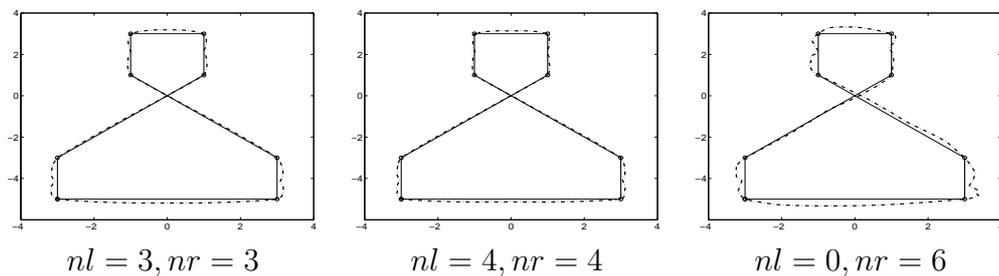


Figura 2.7: Puntos de control iniciales (o-) y curva generada (-) utilizando cinco niveles de la reconstrucción $\overline{\mathcal{IAP}}$ para $r = 3$ y diferentes valores nl (número de nodos a la izquierda de x_{2i-1}^k) y nr (número de nodos a la derecha de x_{2i-1}^k).

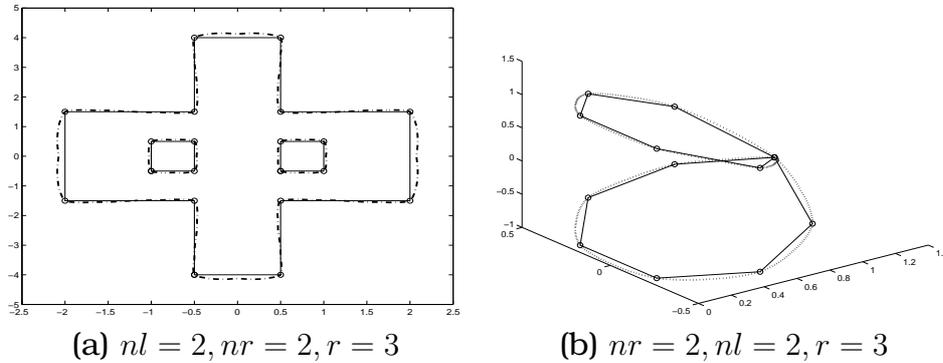


Figura 2.8: Puntos de control iniciales (o-) y curva generada (-) aplicando 5 niveles de la reconstrucción $\overline{\mathcal{IAP}}$ para valores puntuales en un ejemplo en 2 dimensiones (a) y otro para 3 dimensiones (b).

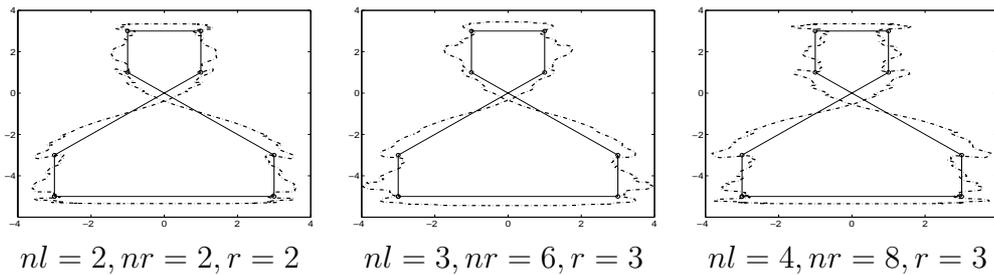


Figura 2.9: Puntos de control iniciales y curva generada utilizando 5 niveles de la reconstrucción $\overline{\mathcal{IAC}}$.

2.3

Reconstrucción utilizando aproximaciones

En este apartado partiremos de los bien conocidos mínimos cuadrados para introducir la reconstrucción por *aproximaciones* en esquemas de multirresolución *à la Harten*, particularizándola para valores puntuales y para medias en celda. Los esquemas tendrán una diferencia fundamental respecto a la reconstrucción *interpolación aproximación*, ya que al eliminar las condiciones interpolatorias, las transformaciones entre $\{F_i^k\}_i$ y sus medias $\{\bar{f}_i^k\}_i$

no serán factibles en todas las situaciones. Aun así, trataremos de aplicar las técnicas vistas en apartados anteriores, *ENO*, *SR*, *WENO*, *PPH*, calculando todos los filtros necesarios para construir las reconstrucciones.

2.3.1

Aproximaciones para valores puntuales

($\overline{\mathcal{AP}}$)

Trabajando con la discretización por valores puntuales (sección 1.2), definimos la **reconstrucción por aproximaciones para valores puntuales** y la denotamos por $\overline{\mathcal{AP}}$ como sigue:

Dados k y el intervalo $[x_{i-1}^{k-1}, x_i^{k-1}]$, construimos el polinomio de mínimos cuadrados que aproxima a los valores de f en m puntos adyacentes a $[x_{i-1}^{k-1}, x_i^{k-1}]$ para $i = 1, 2, \dots, J_k$. Llamando $q_i^{\overline{\mathcal{AP}}}(x; f^k, r)$ a dicho polinomio:

$$\overline{\mathcal{AP}}_{k-1}(x; f^{k-1}) = q_i^{\overline{\mathcal{AP}}}(x; f^{k-1}, r), \quad x \in [x_{i-1}^k, x_i^k], \quad i = 1, \dots, J_k.$$

NOTA 2.7. No podemos asegurar que la reconstrucción $\overline{\mathcal{AP}}$ cumpla $\mathcal{D}_k \mathcal{R}_k \overline{\mathcal{AP}} = \overline{\mathcal{AP}}$ con lo cual no es un operador reconstrucción según la Definición 1.7, aunque por abuso del lenguaje seguiremos denominándola como tal.

La predicción se define de la siguiente manera:

$$(P_{k-1}^k f^{k-1})_i = (\mathcal{D}_k \overline{\mathcal{AP}}_{k-1}(\cdot, f^{k-1}))_i = \overline{\mathcal{AP}}_{k-1}(x_i^k, f^{k-1}).$$

El error de predicción será: $e_i^k = f_i^k - \overline{\mathcal{AP}}_{k-1}(x_i^k; f^{k-1})$, $i = 0, \dots, J_k$. Dado que estamos en valores puntuales $e_i^k = 0$, $\forall i$ tal que $x_i^k \in X^{k-1}$ y $d_j^k = e_{2j-1}^k$ para $j = 1, \dots, J_{k-1}$.

Cálculo de la reconstrucción $\overline{\mathcal{AP}}$

Para cada intervalo $I_i^{k-1} = [x_{i-1}^{k-1}, x_i^{k-1}]$ buscamos un polinomio de grado r que aproxime a la función por mínimos cuadrados en

m puntos adyacentes al intervalo. Sea $q(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$ dicho polinomio y $\{f_i^{k-1}\}_i$ la discretización de la función a reconstruir,

NOTA 2.8. Con el fin de simplificar la exposición, al igual que en la reconstrucción \overline{AP} y sin pérdida de generalidad, calcularemos $q(x)$ sobre los nodos $0, 1, 2, \dots$ y no sobre $\{x_i^k\}_i$ ya que éstos los suponemos equiespaciados.

Para aplicar mínimos cuadrados necesitamos al menos $r + 2$ condiciones. Si $m \geq r + 2$ es el número de condiciones, tomamos nl a la izquierda de f_{2i-1}^k y nr a la derecha:

$$q(j) = f_{i+j-1}^{k-1}, \quad j = -nl + 1, \dots, nr. \quad (2.34)$$

En forma matricial, tenemos la matriz $(nl + nr) \times (r + 1)$

$$A_{nl, nr}^{\overline{AP}} = \begin{bmatrix} 1 & (-nl + 1) & \cdots & (-nl + 1)^r \\ & \vdots & \ddots & \vdots \\ 1 & -1 & \cdots & (-1)^r \\ 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1^r \\ 1 & 2 & \cdots & 2^r \\ & \vdots & \ddots & \vdots \\ 1 & (nr) & \cdots & (nr)^r \end{bmatrix},$$

con $\hat{a} = [a_1, \dots, a_{r+1}]$, y $\hat{f} = [f_{i-nl}^{k-1}, \dots, f_{i-1}^{k-1}, f_i^{k-1}, \dots, f_{i+nr-1}^{k-1}]^T$.

Para obtener los coeficientes $\{a_1, \dots, a_{r+1}\}$ del polinomio $q(x)$ debemos resolver el sistema $A_{nl, nr}^{\overline{AP} T} A_{nl, nr}^{\overline{AP}} \hat{a} = A_{nl, nr}^{\overline{AP} T} \hat{f}$, es decir:

$$\hat{a} = \left(A_{nl, nr}^{\overline{AP} T} A_{nl, nr}^{\overline{AP}} \right)^{-1} A_{nl, nr}^{\overline{AP} T} \hat{f}. \quad (2.35)$$

Al polinomio calculado, $q(x)$, lo denotaremos $p_i^{\overline{AP}}(x; f^{k-1}, r)$, o bien $q_i^{\overline{AP}}(x; f^{k-1}, r)$ si los nodos son $\{x_i^{k-1}\}_i$.

El valor que aproxima a f_{2i-1}^k será $p_i^{\overline{AP}}(\frac{1}{2}; f^{k-1}, r)$. Así:

$$\tilde{f}_{2i-1}^k = \sum_{l=1}^{r+1} a_l \frac{1}{2^{l-1}} = \left(1, \frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^r}\right) C \hat{f}$$

$$= \sum_{l=-nl}^{nr-1} (\beta_{nl,nr}^r)_l f_{i+l}^{k-1} \equiv \overline{\mathcal{AP}}_{nl,nr}^r(x_{2i-1}^{k-1}; f^{k-1}),$$

siendo

$$C = \left(A_{nl,nr}^{\overline{\mathcal{AP}}} \right)^{-1} A_{nl,nr}^{\overline{\mathcal{AP}}} \text{ }^T, \\ (\beta_{nl,nr}^r)_l = \left(1, \frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^r} \right) C e_{nl+l+1}, \quad l = -nl, \dots, nr - 1,$$

donde $e_j = (\delta_{j,k})_{k=1}^{nl+nr}$. En la Tabla 2.6 mostramos³ algunos de estos filtros. Se cumple (ver Nota 2.2):

$$(\beta_{nl,nr}^r)_{-nl+l} = (\beta_{nr,nl}^r)_{nl-l-1}, \quad l = 0, \dots, nl + nr - 1. \quad (2.36)$$

| l | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
|---------------------|--------------------|---------------------|-------------------|-------------------|-------------------|--------------------|-------------------|-----------------|-----------------|
| $(\beta_{3,1}^1)_l$ | | | $\frac{-1}{20}$ | $\frac{3}{20}$ | $\frac{7}{20}$ | $\frac{11}{20}$ | | | |
| $(\beta_{2,2}^1)_l$ | | | | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | $\frac{1}{4}$ | | |
| $(\beta_{3,1}^2)_l$ | | | $\frac{-9}{80}$ | $\frac{17}{80}$ | $\frac{33}{80}$ | $\frac{39}{80}$ | | | |
| $(\beta_{3,2}^2)_l$ | | | $\frac{-3}{20}$ | $\frac{11}{40}$ | $\frac{9}{20}$ | $\frac{3}{8}$ | $\frac{1}{20}$ | | |
| $(\beta_{3,3}^2)_l$ | | | $\frac{-3}{32}$ | $\frac{7}{32}$ | $\frac{3}{8}$ | $\frac{3}{8}$ | $\frac{7}{32}$ | $\frac{-3}{32}$ | |
| $(\beta_{5,1}^3)_l$ | $\frac{151}{2016}$ | $\frac{-323}{2016}$ | $\frac{-23}{504}$ | $\frac{113}{504}$ | $\frac{131}{288}$ | $\frac{911}{2016}$ | | | |
| $(\beta_{4,2}^3)_l$ | | $\frac{9}{224}$ | $\frac{-33}{224}$ | $\frac{1}{8}$ | $\frac{27}{56}$ | $\frac{123}{224}$ | $\frac{-11}{224}$ | | |
| $(\beta_{4,4}^3)_l$ | | $\frac{-3}{32}$ | $\frac{3}{32}$ | $\frac{7}{32}$ | $\frac{9}{32}$ | $\frac{9}{32}$ | $\frac{7}{32}$ | $\frac{3}{32}$ | $\frac{-3}{32}$ |
| f_{i+l}^{k-1} | f_{i-5}^{k-1} | f_{i-4}^{k-1} | f_{i-3}^{k-1} | f_{i-2}^{k-1} | f_{i-1}^{k-1} | f_i^{k-1} | f_{i+1}^{k-1} | f_{i+2}^{k-1} | f_{i+3}^{k-1} |

Tabla 2.6: Aproximaciones a f_{2i-1}^k , de grado r , $\overline{\mathcal{AP}}_{nl,nr}^r(x_{2i-1}^k, f^{k-1}) \equiv \sum_{l=-nl}^{nr-1} (\beta_{nl,nr}^r)_l f_{i+l}^{k-1}$, con nl nodos a izquierda y nr a derecha de f_{2i-1}^k .

Orden de la reconstrucción

Los siguientes resultados son triviales teniendo en cuenta la técnica de mínimos cuadrados polinómicos:

³En <http://www.uv.es/anims/noguera/> podemos descargar programas y una GUI en MATLAB® que calculan dichos filtros según los datos introducidos por el usuario.

PROPOSICIÓN 2.4. Si $q_i^{\overline{\mathcal{AP}}}(x; f^{k-1}, r)$ es el polinomio por aproximaciones para valores puntuales de grado r , la reconstrucción $\overline{\mathcal{AP}}$ recupera de forma exacta polinomios de grado menor o igual a r .

COROLARIO 2.3. El orden de la reconstrucción $\overline{\mathcal{AP}}$ es $r + 1$.

ENO

Al utilizar condiciones centradas obtenemos una serie de intervalos de escasa aproximación alrededor de cada discontinuidad. Si $n = nl = nr$ y la discontinuidad se halla en $[x_{i-1}^{k-1}, x_i^{k-1}]$ los intervalos afectados son $[x_{i-n}^{k-1}, \dots, x_{i+n-1}^{k-1}]$. Y en términos de coeficientes de escala:

$$d_l^{k-1}, \quad i - nl + 1 \leq l \leq i + nr. \quad (2.37)$$

Para evitar utilizar *stencils* que crucen discontinuidades se utiliza el Algoritmo 1.1 o 1.2 (*ENO-VP-I* o *ENO-VP-II*) para *stencils* de longitud m . Para cada intervalo $[x_{i-1}^{k-1}, x_i^{k-1}]$ obtenemos el valor s_i que da lugar al *stencil* $\mathcal{S}_i^{ENO} = \{x_{s_i-1}^{k-1}, \dots, x_{m+s_i-2}^{k-1}\}$. Para construir el polinomio de mínimos cuadrados sobre \mathcal{S}_i^{ENO} utilizamos las ecuaciones (2.34) con (de forma similar a la Figura 2.1 pero aproximado en todos los nodos):

$$nl_i = i - s_i + 1, \quad nr_i = m - nl_i. \quad (2.38)$$

WENO

El primer paso para aplicar *WENO* a la reconstrucción por aproximaciones para valores puntuales es expresar la aproximación de orden $r + 1$ como una combinación lineal de las n aproximaciones de orden $s + 1$, con $r > s$:

$$p_i^{\overline{\mathcal{AP}}_{n,n}^r} = \sum_{k=0}^{n-1} C_k^s p_i^{\overline{\mathcal{AP}}_{n-k,k+1}^s}, \quad (2.39)$$

con $C_k^s \geq 0$ y $\sum_{k=0}^{n-1} C_k^s = 1$, siendo $p_i^{\overline{\mathcal{AP}}^{nl, nr}}$ el polinomio de la reconstrucción por aproximaciones de grado t construido tomando nl nodos a la izquierda y nr a la derecha de x_{2i-1}^k . Para que realmente se aproxime, en (2.39) debemos tomar $2n \geq r+2$ y $2n \geq s+2$.

En (2.39) no se ha podido hallar ninguna combinación que cumpla las condiciones expuestas, con lo que WENO no es viable para esta reconstrucción.

PPH

Si trabajamos con *stencils* de 4 nodos ($nl + nr = 4$), tenemos dos posibles elecciones para aplicar PPH: $\overline{\mathcal{AP}}_{2,2}^1$ y $\overline{\mathcal{AP}}_{2,2}^2$. Para $r > 2$ no se cumple la condición $r + 2 \leq nl + nr$.

Para $r = 1$, tenemos

$$\begin{aligned} \overline{\mathcal{AP}}_{2,2}^1(x_{2i-1}^k; f^{k-1}) &= \frac{1}{4}f_{i-2}^{k-1} + \frac{1}{4}f_{i-1}^{k-1} + \frac{1}{4}f_i^{k-1} + \frac{1}{4}f_{i+1}^{k-1} \\ &= \frac{1}{2}(f_{i-1}^{k-1} + f_i^{k-1}) + \frac{1}{2}\left(\frac{d^2 f_{i-1}^{k-1} + d^2 f_i^{k-1}}{2}\right). \end{aligned}$$

Con esto

$$\left(\overline{\mathcal{AP}}_{2,2}^1\right)^{PPH}(x_{2i-1}^k; f^{k-1}) = \frac{1}{2}(f_{i-1}^{k-1} + f_i^{k-1}) + \frac{1}{2}H(d^2 f_{i-1}^{k-1}, d^2 f_i^{k-1}),$$

donde $H(x, y)$ es la media armónica definida en (1.30).

Para $r = 2$ los coeficientes son los mismos que para $\overline{\mathcal{IP}}_{2,2}^3$, con lo cual:

$$\left(\overline{\mathcal{AP}}_{2,2}^2\right)^{PPH}(x_{2i-1}^k; f^{k-1}) = \frac{1}{2}(f_{i-1}^{k-1} + f_i^{k-1}) - \frac{1}{8}H(d^2 f_{i-1}^{k-1}, d^2 f_i^{k-1}).$$

SR

El algoritmo SR adaptado a la reconstrucción por aproximaciones para valores puntuales se resume en:

1.- Calculamos los *stencils* ENO mediante el Algoritmo 1.1 o 1.2 (ENO-VP-I o ENO-VP-II) y construimos el polinomio $p_{s_i}^{\overline{\mathcal{AP}}}$ según (2.38) y (2.34).

2.- Si $nr_{i-1} = 1$ y $nl_{i+1} = 1$ los *stencils* para las celdas I_{i-1}^{k-1} y I_{i+1}^{k-1} son disjuntos. Etiquetamos dicha celda como *sospechosa* de contener una discontinuidad.

3.- Para cada celda sospechosa definimos la función

$$G_i^{\overline{AP}}(x) = q_{s_{i+1}}^{\overline{AP}}(x; f^{k-1}, r) - q_{s_{i-1}}^{\overline{AP}}(x; f^{k-1}, r).$$

Si $G_i^{\overline{AP}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{AP}}(x_i^{k-1}) < 0$ etiquetamos la celda I_i^{k-1} como singular

4.- Si $G_i^{\overline{AP}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{AP}}(x_{2i-1}^k) < 0$, el punto x_{2i-1}^k se halla a la derecha de la discontinuidad, por tanto $\tilde{f}_{2i-1}^k = q_{s_{i+1}}^{\overline{AP}}(x_{2i-1}^k; f^{k-1}, r)$ o según nuestra forma de calcular:

$$\begin{aligned} \tilde{f}_{2i-1}^k &= p_{s_{i+1}}^{\overline{AP}}(1/2; f^{k-1}, r) = \sum_{l=1}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1} \\ &= \left[1, \frac{1}{2}, \dots, \frac{1}{2^r}\right] C^D [f_i^{k-1}, \dots, f_{i+nl+nr-1}^{k-1}]^T = \sum_{l=0}^{nl+nr-1} (\beta_{0, nl+nr}^r)_l f_l^{k-1}, \end{aligned}$$

siendo

$$\begin{aligned} C^D &= \left(A_{0, nl+nr}^{\overline{AP} T} A_{0, nl+nr}^{\overline{AP}} \right)^{-1} A_{0, nr+nr}^{\overline{AP} T}, \quad (2.40) \\ (\beta_{0, nl+nr}^r)_l &= \left[1, \frac{1}{2}, \dots, \frac{1}{2^r}\right] C^D e_l^D, \quad l = 0, \dots, nl + nr - 1, \end{aligned}$$

donde $e_j^D = (\delta_{j,k})_{k=0}^{nl+nr-1}$. En la Tabla 2.7 mostramos estos filtros para ciertos valores de nl , nr y r .

En otro caso x_{2i-1}^k se halla a la izquierda de la discontinuidad, con lo cual $\tilde{f}_{2i-1}^k = q_{s_{i-1}}^{\overline{AP}}(x_{2i-1}^k; f^{k-1}, r)$ o según nuestra forma calcular:

$$\begin{aligned} \tilde{f}_{2i-1}^k &= p_{s_{i-1}}^{\overline{AP}}\left(\frac{1}{2}; f^{k-1}, r\right) = \sum_{l=1}^{r+1} b_l \left(\frac{1}{2}\right)^{l-1} \\ &= \left[1, \frac{1}{2}, \dots, \frac{1}{2^r}\right] C^I [f_{i-nl-nr}^{k-1}, \dots, f_{i-1}^{k-1}]^T = \sum_{l=-nl-nr}^{-1} (\beta_{nl+nr, 0}^r)_l f_l^{k-1}, \end{aligned}$$

siendo

$$C^I = \left(A_{nl+nr, 0}^{\overline{AP} T} A_{nl+nr, 0}^{\overline{AP}} \right)^{-1} A_{nr+nr, 0}^{\overline{AP} T}, \quad (2.41)$$

$$(\beta_{nl+nr,0}^r)_l = \left[1, \frac{1}{2}, \dots, \frac{1}{2^r}\right] C^I e_l^I, \quad l = -nl - nr, \dots, -1,$$

donde $e_j^I = (\delta_{j,k})_{k=-nl-nr}^{-1}$. Estos filtros son simétricos de los anteriores, es decir, se cumple (ver Nota 2.2):

$$(\beta_{0,nl+nr}^r)_l = (\beta_{nl+nr,0}^r)_{-l-1}, \quad l = 0, \dots, nl + nr - 1.$$

| l | 0 | 1 | 2 | 3 | 4 | 5 |
|---------------------|---------------------|---------------------|--------------------|-------------------|---------------------|-------------------|
| $(\beta_{0,3}^1)_l$ | $\frac{13}{12}$ | $\frac{1}{3}$ | $-\frac{5}{12}$ | | | |
| $(\beta_{0,4}^2)_l$ | $\frac{123}{80}$ | $-\frac{19}{80}$ | $-\frac{51}{80}$ | $\frac{27}{80}$ | | |
| $(\beta_{0,5}^3)_l$ | $\frac{2129}{1120}$ | $-\frac{583}{560}$ | $-\frac{57}{140}$ | $\frac{467}{560}$ | $-\frac{321}{1120}$ | |
| $(\beta_{0,6}^3)_l$ | $\frac{636}{377}$ | $-\frac{543}{1199}$ | $-\frac{313}{504}$ | $\frac{43}{504}$ | $\frac{590}{1037}$ | $-\frac{77}{288}$ |
| f_{i+l}^{k-1} | f_i^{k-1} | f_{i+1}^{k-1} | f_{i+2}^{k-1} | f_{i+3}^{k-1} | f_{i+4}^{k-1} | f_{i+5}^{k-1} |

Tabla 2.7: Aproximaciones a f_{2i-1}^k , de grado r , $\overline{\mathcal{AP}}_{0,nr+nl}^r(x_{2i-1}^k; f^{k-1}) \equiv \sum_{l=-nl}^{nr-1} (\beta_{nl,nr}^r)_l f_{i+l}^{k-1}$, con 0 nodos a izquierda y $nl + nr$ a derecha de f_{2i-1}^k .

Finalmente desarrollemos el cálculo para la función G :

$$\begin{aligned} G_i^{\overline{\mathcal{AP}}}(x) &= p_{s_{i+1}}^{\overline{\mathcal{AP}}}(x; f^{k-1}, r) - p_{s_{i-1}}^{\overline{\mathcal{AP}}}(x; f^{k-1}, r) \\ &= [1, x, \dots, x^r] (C^D [\bar{f}_i^{k-1}, \dots, \bar{f}_{i+nl+nr-1}^{k-1}]^T - C^I [\bar{f}_{i-nl-nr}^{k-1}, \dots, \bar{f}_{i-1}^{k-1}]^T) \\ &= \sum_{l=-nl-nr}^{nl+nr-1} (\sigma_{nl,nr}^r)_{i+l} \bar{f}_{i+l}^{k-1}, \end{aligned}$$

siendo

$$\begin{aligned} (\sigma_{nl,nr}^r)_l &= -[1, x, \dots, x^r] C^I e_l^I, \quad l = -nl - nr, \dots, -1, \\ (\sigma_{nl,nr}^r)_l &= [1, x, \dots, x^r] C^D e_l^D, \quad l = 0, \dots, nl + nr - 1, \end{aligned}$$

con $e_j^I = (\delta_{j,k})_{k=-nl-nr}^{-1}$, $e_j^D = (\delta_{j,k})_{k=0}^{nl+nr-1}$, y C^D , C^I las definidas en (2.40) y (2.41) respectivamente.

Como ejemplo exponemos los filtros para $r = 3$ y $nl + nr = 6$:

$$\begin{aligned} G_i^{\overline{\mathcal{AP}}}(x_{i-1}^{k-1}) &= \frac{2}{63} f_{i-6}^{k-1} - \frac{11}{126} f_{i-5}^{k-1} + \frac{2}{63} f_{i-4}^{k-1} + \frac{1}{9} f_{i-3}^{k-1} - \frac{8}{63} f_{i-2}^{k-1} - \frac{121}{126} f_{i-1}^{k-1} \\ &\quad + \frac{8}{3} f_i^{k-1} - \frac{4}{3} f_{i+1}^{k-1} - \frac{4}{3} f_{i+2}^{k-1} + \frac{1}{3} f_{i+3}^{k-1} + \frac{4}{3} f_{i+4}^{k-1} - \frac{2}{3} f_{i+5}^{k-1}, \end{aligned}$$

$$\begin{aligned}
G_i^{\overline{\mathcal{AP}}}(x_{2i-1}^k) &= \frac{77}{288}f_{i-6}^{k-1} - \frac{590}{1037}f_{i-5}^{k-1} - \frac{43}{504}f_{i-4}^{k-1} + \frac{313}{504}f_{i-3}^{k-1} + \frac{543}{1199}f_{i-2}^{k-1} - \frac{636}{377}f_{i-1}^{k-1} \\
&\quad + \frac{636}{377}f_i^{k-1} - \frac{543}{1199}f_{i+1}^{k-1} - \frac{313}{504}f_{i+2}^{k-1} + \frac{43}{504}f_{i+3}^{k-1} + \frac{590}{1037}f_{i+4}^{k-1} - \frac{77}{288}f_{i+5}^{k-1}, \\
G_i^{\overline{\mathcal{AP}}}(x_i^{k-1}) &= \frac{2}{3}f_{i-6}^{k-1} - \frac{4}{3}f_{i-5}^{k-1} - \frac{1}{3}f_{i-4}^{k-1} + \frac{4}{3}f_{i-3}^{k-1} - \frac{4}{3}f_{i-2}^{k-1} - \frac{8}{3}f_{i-1}^{k-1} \\
&\quad + \frac{121}{126}f_i^{k-1} + \frac{8}{63}f_{i+1}^{k-1} - \frac{1}{9}f_{i+2}^{k-1} - \frac{2}{63}f_{i+3}^{k-1} + \frac{11}{126}f_{i+4}^{k-1} - \frac{2}{63}f_{i+5}^{k-1}.
\end{aligned}$$

Ejemplos

Aplicamos la reconstrucción $\overline{\mathcal{AP}}$ a la función (1.44) compuesta por trozos suaves que contiene un salto y una esquina. Bajamos tres niveles por valores puntuales y subimos tres mediante la reconstrucción $\overline{\mathcal{AP}}$.

En la Figura 2.10 utilizamos *stencils* de 6 nodos centrados y grado 3 para el polinomio de mínimos cuadrados. En (a) se aprecia el *fenómeno de Gibbs* alrededor de los saltos, dando lugar a coeficientes de escala de magnitud elevada, como apreciamos en la Figura (b). En dicho gráfico representamos para cada nivel (eje Y) los coeficientes $d_i^k \geq 10^{-3}$. En el primer nivel aparecen 5 coeficientes de elevada magnitud en consonancia con (2.37).

Al utilizar la técnica *ENO*, Figura 2.10 (c), dichos intervalos se reducen a uno, el que contiene la discontinuidad. Finalmente, mediante *SR*, la aproximación mejora en la esquina y no en el salto ya que $G_i^{\overline{\mathcal{AP}}}(x)$ no cambia de signo en los extremos del intervalo que lo contiene, Figura 2.10 (d).

WENO no aparece, ya que hemos comprobado que no es posible aplicarlo en este caso. Para *PPH*, sin embargo tenemos dos posibilidades con *stencils* de 4 nodos: $r = 1$ y $r = 2$. En ambos casos desaparece el *fenómeno de Gibbs*. Para $r = 2$ la aproximación *PPH* es la misma que para la reconstrucción interpolatoria. Para $r = 1$ la aproximación empeora, debido a que utilizamos rectas para reconstruir.

Observamos que en el caso centrado se produce una reconstrucción de escasa suavidad alrededor de las discontinuidades. Se debe a la imposición de consistencia, $f_{2i}^k = f_i^{k-1}$. Podríamos obtener

aproximaciones más suaves tomando:

$$\tilde{f}_{2i}^k = \sum_{l=1}^{r+1} a_l 1^{l-1} = \mathbf{1}_r^T \left(A_{nl,nr}^{\overline{\mathcal{AP}}} \right)^{-1} A_{nl,nr}^{\overline{\mathcal{AP}}} \left[f_{i-nl}^{k-1}, \dots, f_{i+nl-1}^{k-1} \right]^T. \quad (2.42)$$

Con estos cambios obtenemos la Figura 2.11. En el caso centrado conseguimos una reconstrucción suave, pero en la técnica *SR* no se obtiene mejora en la discontinuidad esquina, ya que esta reconstrucción no sería consistente y por tanto perdemos información necesaria para recuperar con exactitud los datos de niveles superiores, hecho que también se observa en el salto.

2.3.2

Aproximaciones para medias en celda ($\overline{\mathcal{AC}}$)

Utilizando la discretización por medias en celda definida con detalle en la sección 1.3, definimos la reconstrucción por **aproximaciones para medias en celda** y la denotamos por $\overline{\mathcal{AC}}$ de la siguiente manera:

$$(\overline{\mathcal{AC}} \bar{f}^k)(x) := \frac{d}{dx} \overline{\mathcal{AP}}(x; F^k).$$

NOTA 2.9. Para asegurar consistencia, (1.1), en la reconstrucción $\overline{\mathcal{AC}}$ debemos utilizar la relación $\bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k$ para hallar uno de los dos valores que reconstruimos, es decir, \bar{f}_{2i}^k o \bar{f}_{2i-1}^k .

NOTA 2.10. $\overline{\mathcal{AC}}$ no es una reconstrucción según la Definición 1.7 ya que no cumple que $\mathcal{D}_k \overline{\mathcal{AC}} = I_{V^k}$, aunque seguiremos denominándola como tal.

El operador predicción se calcula como sigue:

$$(P_{k-1}^k f^{k-1})_{2i-1} = (\mathcal{D}_k \overline{\mathcal{AC}}_{k-1} f^{k-1})_{2i-1} = \frac{1}{h_k} \int_{x_{2i-2}^k}^{x_{2i-1}^k} \frac{d}{dx} \overline{\mathcal{AP}}(x; F^{k-1})$$

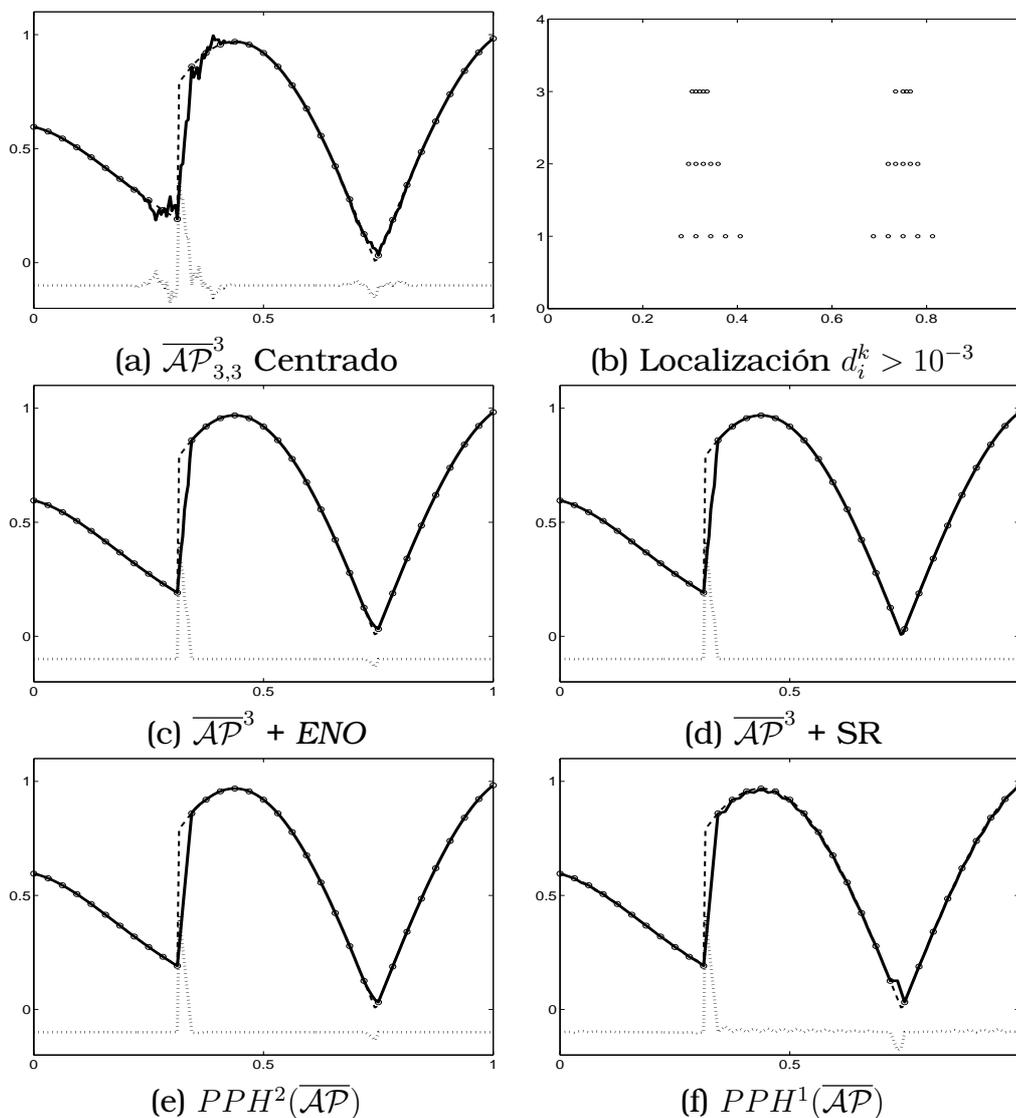


Figura 2.10: Aplicamos 3 niveles de decimación por valores puntuales y 3 niveles de predicción a la función (1.44) discretizada con 257 nodos. En (b) 'o' representa, para cada nivel, la localización de los coeficientes $d_i^k > 10^{-3}$. En el resto, '-' representa la función de partida, '-.' la reconstrucción, 'o' la función en el nivel más grosero y '..' la diferencia entre la discretización inicial y la reconstrucción (desplazada $-0,1$ en el eje OY).

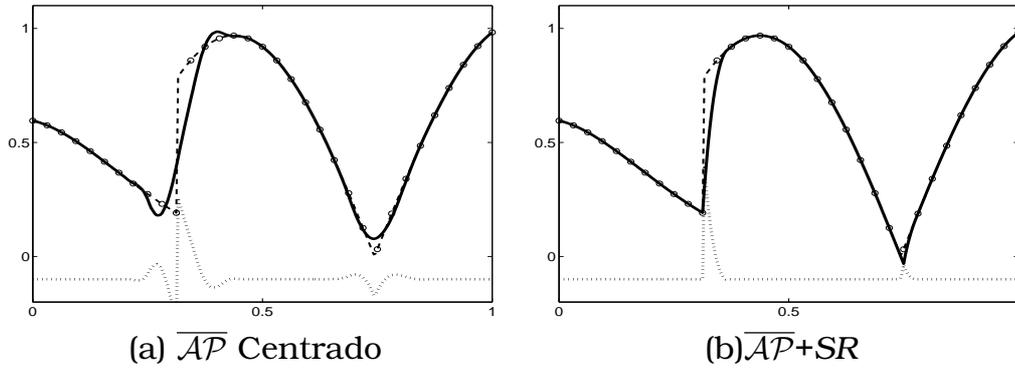


Figura 2.11: Repetimos ciertos casos del experimento de la Figura 2.10 aproximando al reconstruir tanto en f_{2i-1}^k como en f_{2i}^k , según (2.42).

$$= \frac{\overline{\mathcal{AP}}(x_{2i-1}^k; F^{k-1}) - \overline{\mathcal{AP}}(x_{2i-2}^k; F^{k-1})}{h_k} \equiv \overline{\mathcal{AC}}(x_{2i-1}^k, \bar{f}^{k-1}).$$

Cálculo de la reconstrucción $\overline{\mathcal{AC}}$

Fijado k , buscamos un polinomio de grado r , $q(x) = \sum_{l=1}^{r+1} a_l x^{l-1}$, cuyas medias aproximen a las medias en celda $\{\bar{f}_i^{k-1}\}_i$. O dicho de otro modo, $q(x)$ debe interpolar a la función $F^{k-1}(x)$.

Al considerar las medias, a_1 será indeterminado y quedarán por calcular r coeficientes. Para aplicar mínimos cuadrados necesitaremos pues m condiciones, con $m \geq r + 1$, de las cuales nl se tomarán a la izquierda de \bar{f}_i^{k-1} y nr a la derecha ($m = nl + nr + 1 > r$). Sin pérdida de generalidad, expresamos el polinomio sobre números enteros para facilitar la exposición. Imponiendo dichas condiciones, las ecuaciones de mínimos cuadrados serán:

$$\sum_{l=1}^{r+1} a_l ((j+1)^{l-1} - j^{l-1}) = \bar{f}_{i+j}^{k-1}, \quad j = -nl, \dots, nr. \quad (2.43)$$

En forma matricial el sistema será $A_{nl,nr}^{\overline{AC}} \hat{a} = \hat{f}$, donde:

$$A_{nl,nr}^{\overline{AC}} = \begin{bmatrix} (-nl+1)^1 - (-nl)^1 & \cdots & (-nl+1)^r - (-nl)^r \\ (-nl+2)^1 - (nl+1)^1 & \cdots & (-nl+2)^r - (-nl+1)^r \\ \vdots & \ddots & \vdots \\ 1 & \cdots & -(-1)^r \\ 1 & \cdots & 1^r \\ 1 & \cdots & 2^r - 1^r \\ \vdots & \ddots & \vdots \\ (nr+1)^1 - (nr)^1 & \cdots & (nr+1)^r - (nr)^r \end{bmatrix},$$

con $\hat{a} = [a_2, \dots, a_{r+1}]^T$, y $\hat{f} = [\bar{f}_{i-nl}^{k-1}, \dots, \bar{f}_{i-1}^{k-1}, \bar{f}_i^{k-1}, \dots, \bar{f}_{i+nr}^{k-1}]^T$.

Para obtener los coeficientes $\{a_2, \dots, a_{r+1}\}$ del polinomio $q(x)$ resolvemos el sistema $A_{nl,nr}^{\overline{AC}T} A_{nl,nr}^{\overline{AC}} \hat{a} = A_{nl,nr}^{\overline{AC}T} \hat{f}$:

$$\hat{a} = (A_{nl,nr}^{\overline{AC}T} A_{nl,nr}^{\overline{AC}})^{-1} A_{nl,nr}^{\overline{AC}T} \hat{f} = C \hat{f}.$$

NOTA 2.11. El coeficiente a_1 queda sin ser calculado, aunque esto no supone ningún problema, ya que como finalmente buscamos las medias del polinomio $q(x)$, el coeficiente a_1 se elimina.

Al polinomio $q(x)$ que hemos calculado lo denotamos por:

$$q_i^{\overline{AC}}(x; \bar{f}^{k-1}, r-1) \equiv q_i^{\overline{AP}}(x; F^{k-1}, r),$$

o $p_i^{\overline{AC}}(x; \bar{f}^{k-1}, r-1) \equiv p_i^{\overline{AP}}(x; F^{k-1}, r)$ sobre nodos enteros.

Una vez hallados $\{a_l\}_{l=2}^{r+1}$ el valor que aproxima a \bar{f}_{2i-1}^k será:

$$\begin{aligned} \bar{f}_{2i-1}^k &= \frac{p_{s_i}^{\overline{AC}}(\frac{1}{2}; \bar{f}^{k-1}, r-1) - p_{s_i}^{\overline{AC}}(0; \bar{f}^{k-1}, r-1)}{\frac{1}{2}} \\ &= 2 \left(\sum_{l=1}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1} - a_1 \right) = \sum_{l=2}^{r+1} a_l \left(\frac{1}{2}\right)^{l-2} \\ &= \left(1, \frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T \hat{a} = \left(1, \frac{1}{2^1}, \dots, \frac{1}{2^{r-1}}\right)^T C \hat{f} \\ &= \sum_{l=-nl}^{nr} (\gamma_{nl,nr}^r)_l f_{i+l}^{k-1} \equiv \overline{AC}_{nl,nr}^{r-1}(x_{2i-1}^k, \bar{f}^{k-1}), \end{aligned} \quad (2.44)$$

siendo:

$$(\gamma_{nl,nr}^r)_l = (1, \frac{1}{2}, \dots, \frac{1}{2^{r-1}})^T C e_{l+nl+1}, \quad l = -nl, \dots, nr,$$

$$C = (A_{nl,nr}^{\overline{AC}})^T (A_{nl,nr}^{\overline{AC}})^{-1} A_{nl,nr}^{\overline{AC}} \quad ,$$

con $e_j = (\delta_{j,k})_{k=1}^{nl+nr+1}$. En la Tabla 2.8 podemos ver⁴ algunos de estos coeficientes.

| l | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| $(\gamma_{2,0}^2)_l$ | | | $-\frac{1}{14}$ | $\frac{1}{3}$ | $\frac{17}{24}$ | | | | |
| $(\beta_{1,1}^2)_l$ | | | | $\frac{11}{24}$ | $\frac{1}{3}$ | $\frac{5}{24}$ | | | |
| $(\gamma_{0,2}^2)_l$ | | | | | $\frac{23}{24}$ | $\frac{1}{3}$ | $-\frac{7}{24}$ | | |
| $(\gamma_{4,0}^3)_l$ | $-\frac{1}{140}$ | $-\frac{13}{280}$ | $\frac{2}{35}$ | $\frac{17}{56}$ | $\frac{97}{140}$ | | | | |
| $(\gamma_{3,1}^3)_l$ | | $-\frac{23}{140}$ | $\frac{13}{56}$ | $\frac{29}{70}$ | $\frac{107}{280}$ | $\frac{19}{140}$ | | | |
| $(\gamma_{2,2}^3)_l$ | | | $-\frac{1}{28}$ | $\frac{103}{280}$ | $\frac{17}{35}$ | $\frac{89}{280}$ | $-\frac{19}{140}$ | | |
| $(\gamma_{1,3}^3)_l$ | | | | $\frac{53}{140}$ | $\frac{101}{280}$ | $\frac{19}{70}$ | $\frac{31}{280}$ | $-\frac{17}{140}$ | |
| $(\gamma_{0,4}^3)_l$ | | | | | $\frac{151}{140}$ | $\frac{59}{280}$ | $-\frac{8}{35}$ | $-\frac{67}{280}$ | $\frac{5}{28}$ |
| $(\gamma_{4,4}^4)_l$ | $-\frac{315}{2688}$ | $\frac{565}{2688}$ | $\frac{925}{2688}$ | $\frac{1}{3}$ | $\frac{616}{2688}$ | $\frac{29}{384}$ | $-\frac{65}{896}$ | | |
| $(\gamma_{4,4}^4)_l$ | $-\frac{7}{64}$ | $\frac{383}{4224}$ | $\frac{553}{2636}$ | $\frac{1257}{4825}$ | $\frac{59}{231}$ | $\frac{242}{1169}$ | $\frac{784}{6131}$ | $\frac{43}{1408}$ | $-\frac{51}{704}$ |
| \bar{f}_{i+l}^{k-1} | \bar{f}_{i-4}^{k-1} | \bar{f}_{i-3}^{k-1} | \bar{f}_{i-2}^{k-1} | \bar{f}_{i-1}^{k-1} | \bar{f}_i^{k-1} | \bar{f}_{i+1}^{k-1} | \bar{f}_{i+2}^{k-1} | \bar{f}_{i+3}^{k-1} | \bar{f}_{i+4}^{k-1} |

Tabla 2.8: Aproximaciones a \bar{f}_{2i-1}^k con grado $r - 1$ para \bar{f} (r para F), $\overline{AC}_{nl,nr}^{r-1}(x_{2i-1}^k, \bar{f}^{k-1}) = \sum_{l=-nl}^{nr} (\gamma_{nl,nr}^r)_l \bar{f}_{i+l}^{k-1}$, utilizando aproximación con nl puntos a la izquierda y nr puntos a la derecha de \bar{f}_i^{k-1} (este incluido).

Orden de la reconstrucción \overline{AC}

Teniendo en cuenta la definición de la reconstrucción \overline{AC} , se cumplen los siguientes resultados:

⁴Desde <http://www.uv.es/animations/noguera/> podemos descargar programas y una GUI en MATLAB® que calculan dichos filtros según los datos introducidos por el usuario.

PROPOSICIÓN 2.5. Si $q_i^{\overline{\mathcal{AC}}}(x; \bar{f}^{k-1}, r-1)$ es el polinomio por aproximaciones para medias en celda de grado $r-1$ (grado r para $q_i^{\overline{\mathcal{AP}}}(x; F^{k-1}, r)$), la reconstrucción $\overline{\mathcal{AC}}$ recupera de forma exacta polinomios de grado menor o igual a $r-1$.

COROLARIO 2.4. El orden de la reconstrucción $\overline{\mathcal{AC}}$ es r .

ENO

Utilizamos *stencils* centrados de $n+1$ nodos, es decir si $nl = nr = n$ y en (x_{2i-1}^k, x_{2i}^k) hay una discontinuidad, los coeficientes de escala afectados son

$$d_l^k, \quad i-n \leq l \leq i+n, \quad (2.45)$$

mientras que si la discontinuidad se halla en (x_{2i-2}^k, x_{2i-1}^k) son

$$d_l^k, \quad i-n \leq l \leq i+n-1. \quad (2.46)$$

Para evitar utilizar *stencils* que crucen discontinuidades aplicamos la técnica *ENO* al igual que en apartados anteriores. La elección del *stencil* se realiza con el Algoritmo 1.3 o 1.4 (*ENO-MC-I* o *ENO-MC-II*), obteniendo para cada i el valor s_i , que en términos de nl_i y nr_i será:

$$nl_i = i - s_i, \quad nr_i = m - nl_i - 1. \quad (2.47)$$

WENO

La primera condición que se debe satisfacer es la expresión de la aproximación de orden r como una combinación lineal de las $n+1$ aproximaciones de orden s :

$$p_i^{\overline{\mathcal{AC}}^{r-1}} = \sum_{k=0}^n C_k^s p_i^{\overline{\mathcal{AC}}^{s-1}}, \quad (2.48)$$

con $C_k^s \geq 0$, y $\sum_{k=0}^n C_k^s = 1$, siendo $p_i^{\overline{\mathcal{AC}}^{nl, nr}}$ el polinomio de orden t de aproximaciones para medias en celda construido tomando nl nodos a la izquierda y nr nodos a la derecha de \bar{f}_i^{k-1} .

En (2.48) se debe cumplir:

- $r > s$ para que aumente el orden de aproximación.
- $2n \geq r$ para que en $p_i^{\overline{\mathcal{AC}}^{n, n}}$ se aproxime y no sólo se interpole.
- $n \geq s$ para que en $p_i^{\overline{\mathcal{AC}}^{n-k, k}}$ se aproxime y no únicamente se interpole.

En (2.48) no se ha podido hallar ninguna combinación que cumpla las condiciones expuestas, con lo que WENO no es viable para esta reconstrucción.

PPH

Como vimos para el caso de valores puntuales hay dos posibles elecciones partiendo de *stencils* de 4 nodos para F :

$$\begin{aligned} \overline{\mathcal{AP}}_{2,2}^1(x_{2i-1}^k, F^{k-1}) &= \frac{1}{4}F_{i-2}^{k-1} + \frac{1}{4}F_{i-1}^{k-1} + \frac{1}{4}F_i^{k-1} + \frac{1}{4}F_{i+1}^{k-1}, \\ \overline{\mathcal{AP}}_{2,2}^2(x_{2i-1}^k, F^{k-1}) &= -\frac{1}{16}F_{i-2}^{k-1} + \frac{9}{16}F_{i-1}^{k-1} + \frac{9}{16}F_i^{k-1} - \frac{1}{16}F_{i+1}^{k-1}. \end{aligned}$$

Escogiendo $\overline{\mathcal{AP}}_{2,2}^1$:

$$\begin{aligned} \overline{\mathcal{AC}}^1(x_{2i-1}^k, \bar{f}^{k-1}) &= \frac{q_{s_i}^{\overline{\mathcal{AP}}}(x_{2i-1}^{k-1}; F^{k-1}) - q_{s_i}^{\overline{\mathcal{AP}}}(x_{2i-2}^{k-1}; F^{k-1})}{h_k} \\ &= \frac{-\frac{3}{20}F_{i-2}^{k-1} - \frac{1}{20}F_{i-1}^{k-1} + \frac{1}{20}F_i^{k-1} - \frac{3}{20}F_{i+1}^{k-1}}{h_k}. \end{aligned}$$

A partir de la expresión anterior no se ha podido hallar una reordenación que cumpla las condiciones deseadas, es decir algo del estilo

$$\frac{a(F_i^{k-1} - F_{i-1}^{k-1}) + b(d^2 F_{i-1} + d^2 F_i)}{h_k}.$$

Lo mismo ocurre con el caso $\overline{\mathcal{AP}}_{2,2}^2$, con lo que PPH no es aplicable para la reconstrucción $\overline{\mathcal{AC}}$.

Para aplicar SR tenemos el siguiente problema: como no estamos interpolando en ningún punto, la función

$$G_i^{\overline{\mathcal{AC}}}(x) = q_{s_{i+1}}^{\overline{\mathcal{AC}}}(x; \bar{f}^{k-1}, r-1) - q_{s_{i-1}}^{\overline{\mathcal{AC}}}(x; \bar{f}^{k-1}, r-1),$$

no puede expresarse con valores de $\{\bar{f}_i^{k-1}\}_i$. La demostración es análoga a la de la Proposición 2.3, en donde la relación (2.32) no se cumple, y por tanto la diferencia $a_1 - b_1$ no puede ser expresada en términos de $\{\bar{f}_i^{k-1}\}_i$.

Una posible solución es utilizar la función de $\overline{\mathcal{IAC}}$, $G_i^{\overline{\mathcal{IAC}}}(x)$, para decidir si la celda es o no singular. En el resto de pasos ya utilizamos la reconstrucción $\overline{\mathcal{AC}}$. Aunque esto signifique no trabajar únicamente con la reconstrucción por *aproximaciones*, esta licencia nos permite realizar SR para esta reconstrucción, mejorando las soluciones como veremos en los experimentos realizados.

Así pues, la técnica SR puede aplicarse siguiendo estos pasos:

1.- Calculamos los *stencils ENO* mediante el Algoritmo 1.3 o 1.4 (*ENO-MC-I* o *ENO-MC-II* respectivamente) y asignamos los nodos donde interpolaremos para cada intervalo según (2.47).

2.- Si $nr_{i-1} = 0$ y $nl_{i+1} = 0$ los *stencils* para las celdas I_{i-1}^{k-1} y I_{i+1}^{k-1} son disjuntos. Etiquetamos dicha celda como *sospechosa* de contener una discontinuidad.

3.- Para cada celda sospechosa definimos la función

$$G_i^{\overline{\mathcal{IAC}}}(x) = q_{s_{i+1}}^{\overline{\mathcal{IAC}}}(x; \bar{f}^{k-1}, r-1) - q_{s_{i-1}}^{\overline{\mathcal{IAC}}}(x; \bar{f}^{k-1}, r-1).$$

Si $G_i^{\overline{\mathcal{IAC}}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{\mathcal{IAC}}}(x_i^{k-1}) < 0$ etiquetamos la celda I_i^{k-1} como *singular*

4.- Si $G_i^{\overline{\mathcal{IAC}}}(x_{i-1}^{k-1}) \cdot G_i^{\overline{\mathcal{IAC}}}(x_{2i-1}^k) < 0$, el punto x_{2i-1}^k se halla a la derecha de la discontinuidad, por tanto:

$$\bar{f}_{2i}^k = \frac{q_{s_{i+1}}^{\overline{\mathcal{AC}}}(x_{2i}^k; \bar{f}^{k-1}, r-1) - q_{s_{i+1}}^{\overline{\mathcal{AC}}}(x_{2i-1}^k; \bar{f}^{k-1}, r-1)}{h_k} = \frac{p_{s_{i+1}}^{\overline{\mathcal{AP}}}(1; F^{k-1}, r) - p_{s_{i+1}}^{\overline{\mathcal{AP}}}(1/2; F^{k-1}, r)}{1/2}$$

$$\begin{aligned}
 &= 2\left(\sum_{l=1}^{r+1} a_l 1^{l-1} - \sum_{l=1}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1}\right) = 2\left(\sum_{l=1}^{r+1} a_l \left(1 - \left(\frac{1}{2}\right)^{l-1}\right)\right) \\
 &= 2\left[1 - \frac{1}{2}, \dots, 1 - \left(\frac{1}{2}\right)^r\right] C^D [f_{i+1}^{k-1}, \dots, f_{i+nl+nr+2}^{k-1}]' \\
 &= \sum_{l=1}^{nl+nr+1} (\gamma_{-1, nl+nr+1}^r)_l \bar{f}_l^{k-1}, \tag{2.49}
 \end{aligned}$$

siendo:

$$\begin{aligned}
 C^D &= (A_{-1, nl+nr+1}^{\overline{AC}} A_{-1, nl+nr+1}^{\overline{AC}})^{-1} A_{-1, nl+nr+1}^{\overline{AC}}{}^T, \\
 (\beta_{-1, nl+nr+2}^r)_l &= 2\left[1 - \frac{1}{2}, \dots, 1 - \left(\frac{1}{2}\right)^r\right] C^D e_l^D,
 \end{aligned}$$

con $e_l^D = (\delta_{l,k})_{k=1}^{nl+nr+1}$ para $l = 1, \dots, nl + nr + 1$. En la Tabla 2.9 podemos ver dichos coeficientes para ciertos valores de r, nl, nr .

Para asegurar consistencia, el valor f_{2i-1}^k lo hallamos mediante la relación $f_{2i-1}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i}^k$.

En otro caso x_{2i-1}^k se halla a la izquierda de la discontinuidad, con lo cual:

$$\begin{aligned}
 \bar{f}_{2i-1}^k &= \frac{q_{s_{i-1}}^{\overline{AC}}(x_{2i-1}^k; \bar{f}^{k-1, r-1}) - q_{s_{i-1}}^{\overline{AC}}(x_{2i}^k; \bar{f}^{k-1, r-1})}{h_k} = \frac{p_{s_{i-1}}^{\overline{AP}}(1/2; F^{k-1, r}) - p_{s_{i-1}}^{\overline{AP}}(0; F^{k-1, r})}{1/2} \\
 &= 2\left(\sum_{l=2}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1} - \sum_{l=2}^{r+1} a_l (0)^{l-1}\right) = 2\left(\sum_{l=2}^{r+1} a_l \left(\frac{1}{2}\right)^{l-1}\right) \\
 &= 2\left[\frac{1}{2}, \dots, \left(\frac{1}{2}\right)^r\right] C^I [f_{i-nl-nr-1}^{k-1}, \dots, f_{i-1}^{k-1}]' \\
 &= \sum_{l=-nl-nr-1}^{-1} (\gamma_{nl+nr+1, -1}^r)_l \bar{f}_l^{k-1}, \tag{2.50}
 \end{aligned}$$

siendo:

$$\begin{aligned}
 C^I &= (A_{nl+nr+1, -1}^{\overline{AC}} A_{nl+nr+1, -1}^{\overline{AC}})^{-1} A_{nl+nr+1, -1}^{\overline{AC}}{}^T, \\
 (\gamma_{nl+nr+1, -1}^r)_l &= 2\left[\frac{1}{2}, \dots, \left(\frac{1}{2}\right)^r\right] C^I e_{-l-nr-nl}^I,
 \end{aligned}$$

con $e_l^I = (\delta_{l,k})_{k=1}^{nl+nr+1}$ para $l = -nl - nr - 1, \dots, -1$.

Posteriormente, hallamos el valor que falta mediante la relación $\bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k$ con el fin de asegurar consistencia. Dichos filtros podemos deducirlos de la Tabla 2.9, ya que se cumple (ver la Nota 2.2):

$$(\gamma_{-1, nl+nr+1}^r)_l = (\gamma_{-1, nl+nr+1}^r)_{-l}, \quad l = 1, \dots, nl + nr.$$

| l | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $(\gamma_{-1,4}^2)$ | $\frac{37}{40}$ | $\frac{19}{40}$ | $\frac{1}{40}$ | $-\frac{17}{40}$ | | | |
| $(\gamma_{-1,5}^3)$ | $\frac{43}{28}$ | $\frac{23}{240}$ | $-\frac{41}{70}$ | $-\frac{131}{280}$ | $\frac{61}{140}$ | | |
| $(\gamma_{-1,6}^4)$ | $\frac{2394}{1135}$ | $-\frac{6547}{8064}$ | $-\frac{1861}{2016}$ | $\frac{349}{2016}$ | $\frac{529}{605}$ | $-\frac{159}{377}$ | |
| $(\gamma_{-1,7}^4)$ | $\frac{1663}{898}$ | $-\frac{478}{1565}$ | $-\frac{501}{629}$ | $-\frac{1}{3}$ | $-\frac{514}{1397}$ | $\frac{227}{384}$ | $-\frac{1021}{2688}$ |
| $\frac{\bar{f}_{i+l}^{k-1}}{\bar{f}_{i+1}^{k-1}}$ | $\frac{\bar{f}_{i+1}^{k-1}}{\bar{f}_{i+1}^{k-1}}$ | $\frac{\bar{f}_{i+2}^{k-1}}{\bar{f}_{i+2}^{k-1}}$ | $\frac{\bar{f}_{i+3}^{k-1}}{\bar{f}_{i+3}^{k-1}}$ | $\frac{\bar{f}_{i+4}^{k-1}}{\bar{f}_{i+4}^{k-1}}$ | $\frac{\bar{f}_{i+5}^{k-1}}{\bar{f}_{i+5}^{k-1}}$ | $\frac{\bar{f}_{i+6}^{k-1}}{\bar{f}_{i+6}^{k-1}}$ | $\frac{\bar{f}_{i+7}^{k-1}}{\bar{f}_{i+7}^{k-1}}$ |

Tabla 2.9: Aproximaciones para \bar{f}_{2i}^k de grado $r - 1$ para \bar{f} (r para F), $\overline{\mathcal{AC}}^{r-1}_{nl+nr+1,-1}(x_{2i-1}^k, \bar{f}^{k-1}) = \sum_{l=1}^{nl+nr} (\gamma_{-1,nl+nr+1}^r)_l \bar{f}_l^{k-1}$, utilizando aproximación en $nl + nr$ puntos a la derecha de \bar{f}_i^{k-1} .

Ejemplos

Aplicamos la reconstrucción $\overline{\mathcal{AC}}$ a la función (1.70) obteniendo la Figura 2.12. Bajamos tres niveles por medias en celda, eliminamos los coeficientes de escala y subimos tres niveles mediante la reconstrucción por *aproximaciones* para medias en celda. En la Figura (a) utilizamos *stencils* centrados de 7 nodos para \bar{f} (8 nodos para F). Observamos las malas aproximaciones alrededor de las discontinuidades (en las fronteras sí se ha adaptado). En la Figura (b) mostramos para cada nivel los coeficientes de escala mayores o iguales a 10^{-3} . En concreto en el primer nivel observamos que hay 7 coeficientes afectados en concordancia con (2.45). Mediante la técnica *ENO*, Figura (c), los intervalos de escasa aproximación se reducen a uno, al igual que con *WENO*. Al aplicar *SR* se mejora la aproximación tanto en saltos como en esquinas, Figura (d).

Las aproximaciones tan dispares para el caso centrado provienen de imponer consistencia a la reconstrucción $\overline{\mathcal{AC}}$, ya que el valor \bar{f}_{2i-1}^k es fruto de una aproximación, pero $\bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k$. Para mejorarlas podríamos obtener el valor \bar{f}_{2i}^k mediante aproximación pero en este caso perderíamos la propiedad de consistencia, con lo que los operadores predicción no estarían correctamente definidos. Aún así, probamos esta idea para ver los resultados que obtenemos, a fin de aplicar esta reconstrucción en casos prácticos, como por ejemplo la eliminación de ruido. Así pues el valor \bar{f}_{2i}^k se

aproximaría por:

$$\begin{aligned}
 \hat{f}_{2i}^k &= \frac{p_{s_i}^{\overline{AC}}(1; \bar{f}^{k-1}, r-1) - p_{s_i}^{\overline{AC}}(\frac{1}{2}; \bar{f}^{k-1}, r-1)}{1/2} \\
 &= 2 \left(\sum_{l=2}^{r+1} a_l (1 - (\frac{1}{2})^{l-1}) \right), \\
 &= \left[1 - \frac{1}{2}, \dots, 1 - (\frac{1}{2})^r \right] (A_{nl, nr}^{\overline{AC}})^T (A_{nl, nr}^{\overline{AC}})^{-1} (A_{nl, nr}^{\overline{AC}})^T [\bar{f}_{i-nl}^{k-1}, \dots, \bar{f}_{i+nr}^{k-1}].
 \end{aligned} \tag{2.51}$$

Podemos realizar una nueva modificación que asegura la consistencia. Una vez calculados tanto \hat{f}_{2i}^{k-1} como \hat{f}_{2i-1}^{k-1} por aproximación, la reconstrucción definitiva sería:

$$\begin{cases} \tilde{f}_{2i-1}^k = \hat{f}_{2i-1}^k + \hat{f}_i^{k-1} - \frac{1}{2}(\hat{f}_{2i-1}^k + \hat{f}_{2i}^k), \\ \tilde{f}_{2i}^k = \hat{f}_{2i}^k + \hat{f}_i^{k-1} - \frac{1}{2}(\hat{f}_{2i-1}^k + \hat{f}_{2i}^k). \end{cases} \tag{2.52}$$

Con estas modificaciones obtenemos la Figura 2.13. Observamos que en el caso centrado se obtienen reconstrucciones de mayor suavidad. Al aplicar SR mediante la reconstrucción no consistente la discontinuidad de salto queda desplazada en la reconstrucción, ya que dejamos de almacenar información necesaria para la correcta reconstrucción de los datos de partida. Al introducir la segunda modificación, se pierde suavidad en la discontinuidad.

2.3.3

Funciones límite

En este apartado representamos las funciones límite para valores puntuales, Figura 2.14, y medias en celda, Figura 2.15. Para su cálculo debemos utilizar suficientes condiciones para realmente aproximar y no interpolar. Lo cual se traduce en que $nl + nr \geq r + 2$ para valores puntuales y $nl + nr \geq r$ para medias en celda. En general, las funciones que se obtienen no son suaves y por tanto inadecuadas para esquemas de multirresolución.

En valores puntuales, la condición $\hat{f}_{2i}^k = \hat{f}_i^{k-1}$ es la que va a producir las oscilaciones que causarán la pérdida de suavidad. Para

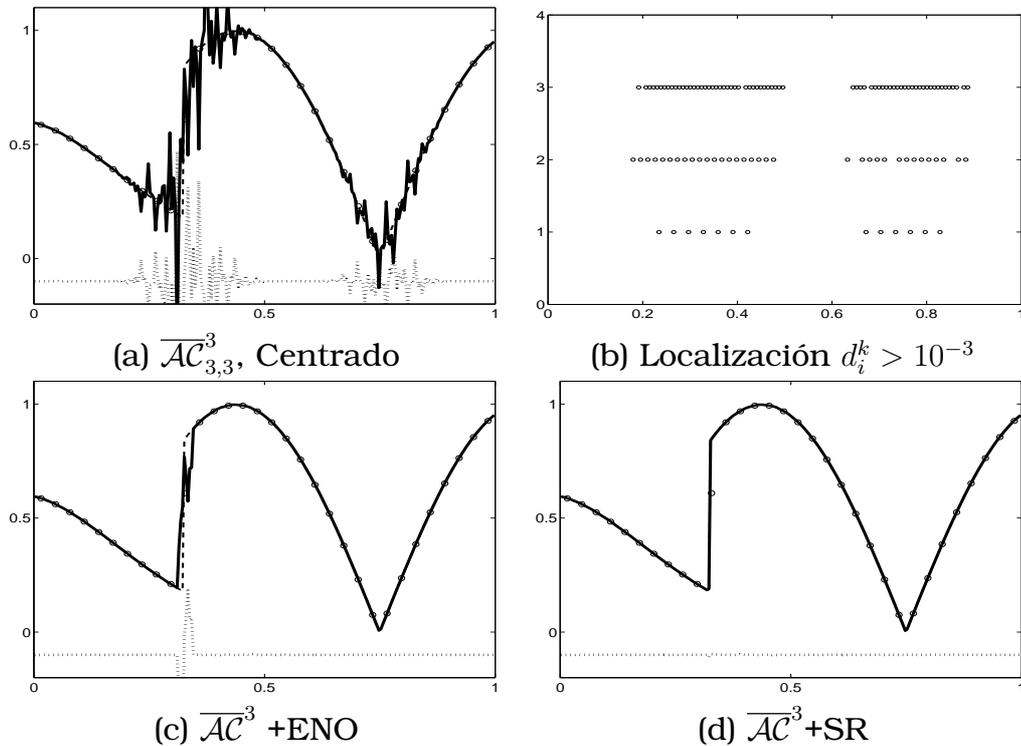


Figura 2.12: Aplicamos 3 niveles de decimación por medias en celda y 3 niveles de predicción a la función (1.70) discretizada con 256 nodos. En (b) 'o' representa, para cada nivel, la localización de los coeficientes $d_i^k > 10^{-3}$. En el resto, '-' representa la función de partida, '-.' la reconstrucción, 'o' la función en el nivel más grosero y '..' la diferencia entre la discretización inicial y la reconstrucción (desplazada $-0,1$ en el eje OY).

evitarlo podríamos modificar el esquema de reconstrucción aproximando también en los nodos pares tal como expusimos en (2.42). Para medias en celda ocurre algo similar, pudiendo introducir las modificaciones (2.51) y (2.52).

En las Figuras 2.16 y 2.17 representamos algunos casos obtenidos por aproximación en ambos nodos para valores puntuales y medias en celda respectivamente. Las funciones límite son más suaves, aunque cabe recordar que no son reconstrucciones consistentes y por tanto las aproximaciones obtenidas pueden desviarse de la original. Forzando consistencia tras aproximar en am-

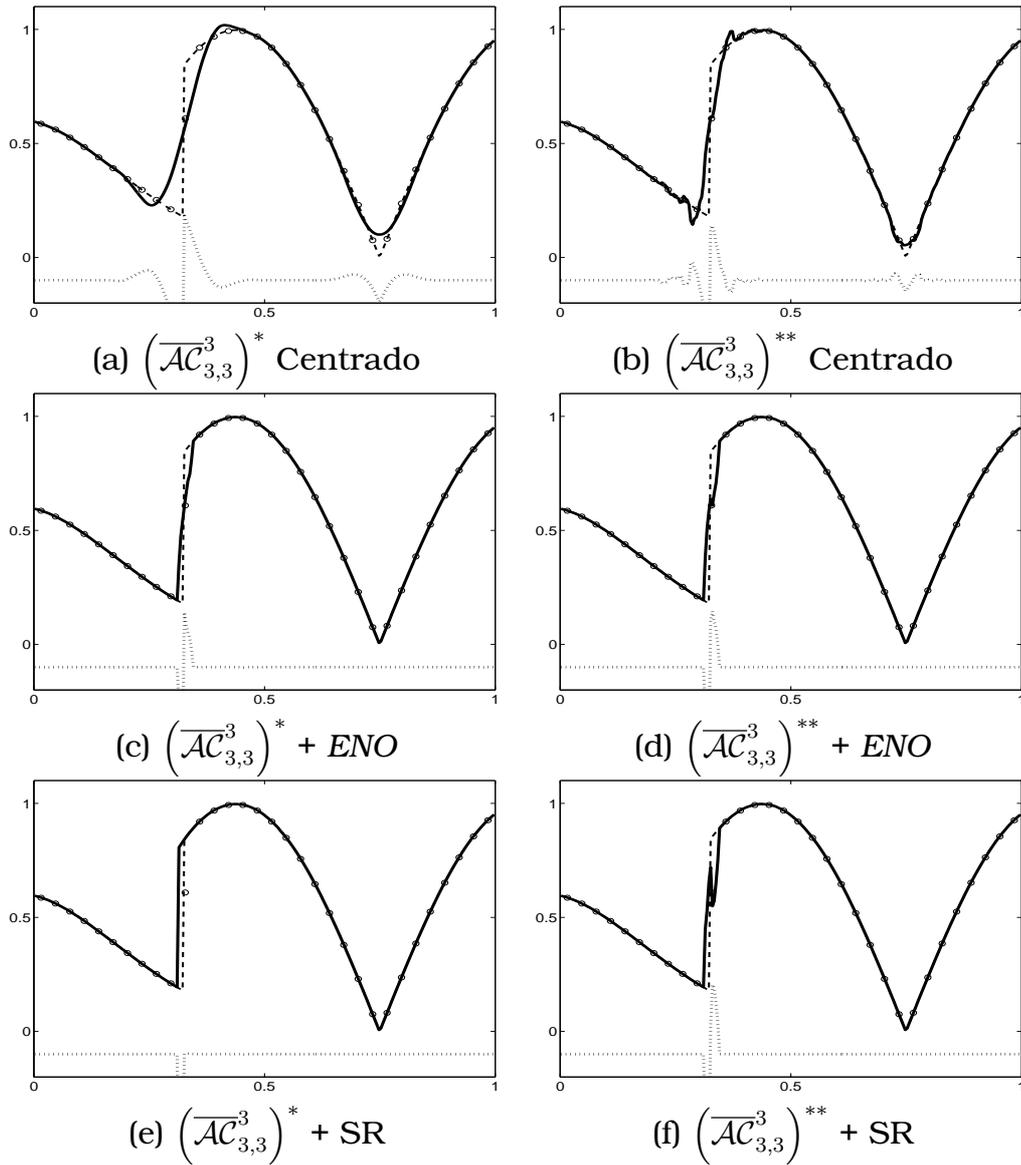


Figura 2.13: Repetimos algunos casos del experimento de la Figura 2.12, pero en (a), (c) y (e), marcado con *, aproximamos tanto en \bar{f}_{2i-1}^k como en \bar{f}_{2i}^k , según (2.51) y en (b), (d) y (f), marcado con **, introducimos la modificación (2.52) para dotar de consistencia a la reconstrucción por aproximaciones en ambos puntos.

bos, (2.52), obtenemos la Figura 2.18 donde observamos que la función límite vuelve a quebrarse, y por tanto no representa una mejora relevante.

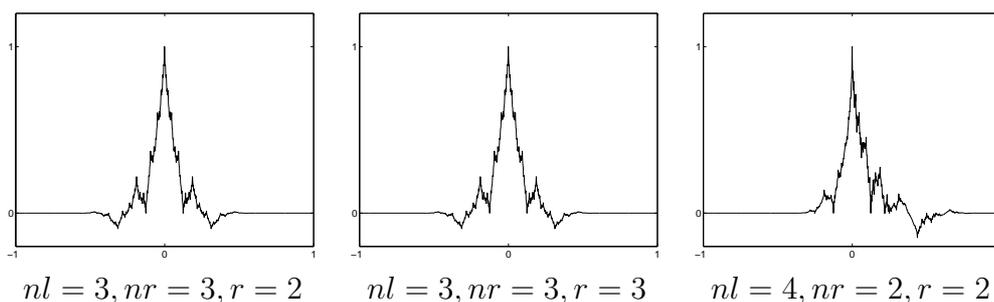


Figura 2.14: Función límite para la reconstrucción \overline{AP} . Utilizamos 6 niveles de reconstrucción.

2.3.4

Diseño de curvas

En consonancia con las funciones límite obtenidas, las reconstrucciones por mínimos cuadrados, en general, no son aplicables para la generación de curvas, como observamos en la Figura 2.19.

Con las modificaciones (2.42) y (2.51) se obtienen curvas suaves aunque alejadas de los puntos de control, Figura 2.20, (a) y (b). En (c) forzamos la consistencia mediante (2.52) y obtenemos de nuevo curvas muy quebradas.

2.4

Conclusiones

En este capítulo hemos desarrollado reconstrucciones no exclusivamente interpolatorias. En primer lugar se ha construido una nueva reconstrucción basada en combinar interpolación y aproximación por mínimos cuadrados, comprobando que cumple los requisitos para ser utilizada en esquemas de multiresolución *à la*

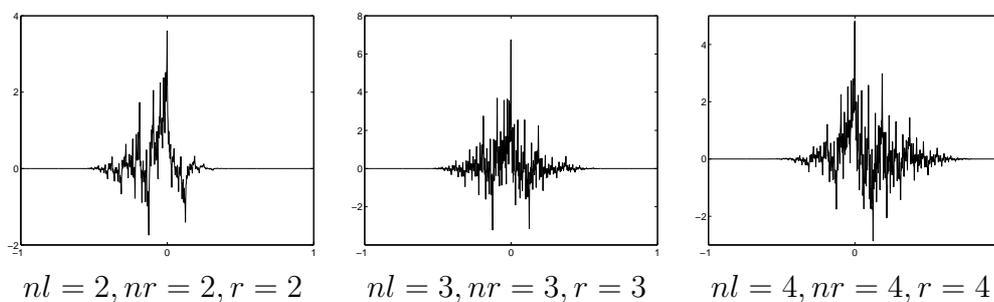


Figura 2.15: Función límite para la reconstrucción \overline{AC} . Utilizamos 6 niveles de reconstrucción.

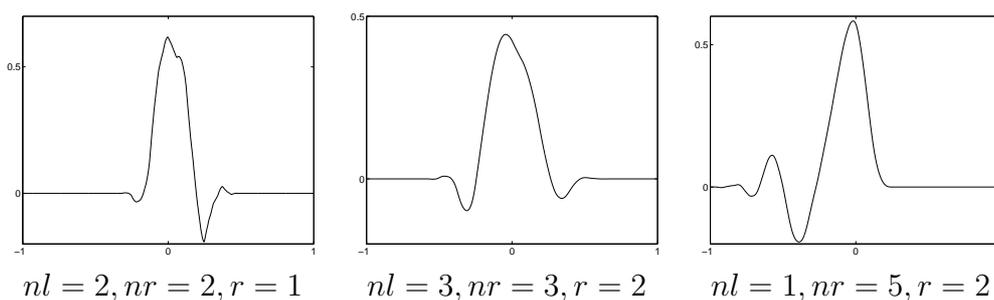


Figura 2.16: Función límite para la reconstrucción \overline{AP} con la modificación (2.42). Utilizamos 6 niveles de reconstrucción.

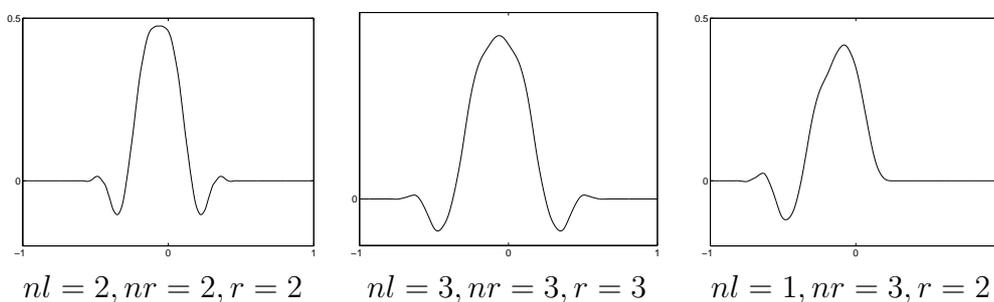


Figura 2.17: Función límite para la reconstrucción \overline{AC} con la modificación (2.51). Utilizamos 6 niveles de reconstrucción.

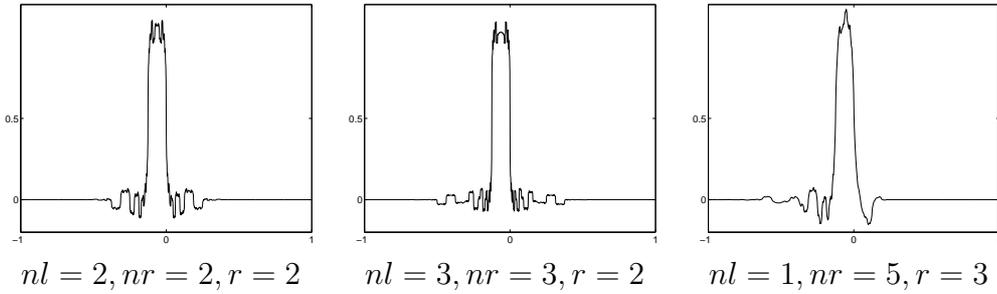


Figura 2.18: Función límite para la reconstrucción $\overline{\mathcal{AC}}$ con la modificación (2.52). Utilizamos 6 niveles de reconstrucción.

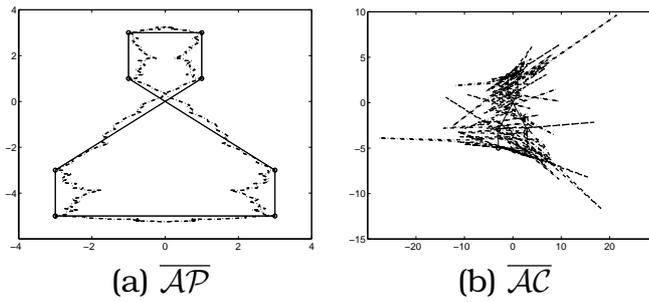


Figura 2.19: Puntos de control iniciales y curva generada aplicando 5 niveles de la reconstrucción $\overline{\mathcal{AP}}$, en (a) y $\overline{\mathcal{AC}}$ en (b), ambas con $nl = 3$, $nr = 3$ y $r = 3$.

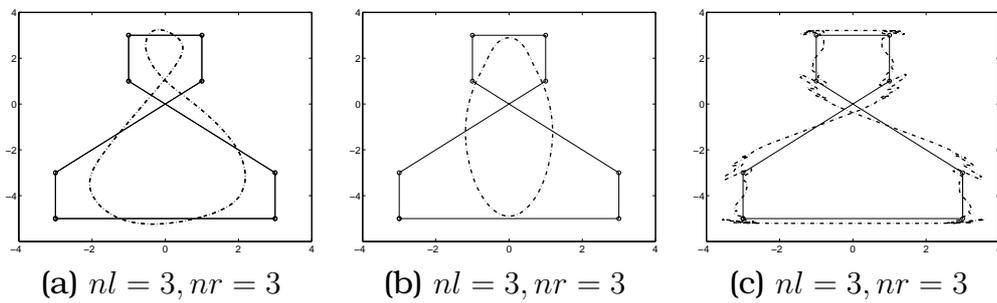


Figura 2.20: Puntos de control iniciales y curva generada aplicando 6 niveles de la reconstrucción $\overline{\mathcal{AP}}$ con grado 3, introduciendo la modificación (2.42) en (a) y para la reconstrucción $\overline{\mathcal{AC}}$ con grado 3, introduciendo (2.51) en (b) y (2.52) en (c).

Harten. Dicha reconstrucción ha sido desarrollada para las discretizaciones por valores puntuales y por medias en celda, calculando los filtros asociados para los casos más utilizados. Tras simples adaptaciones, los esquemas *ENO* y *SR* son aplicables, mejorando la aproximación en las cercanías de las discontinuidades. La técnica *WENO* sólo es aplicable para ciertos casos, en concreto aumentando de orden tres a cuatro para valores puntuales y de dos a tres para medias en celda. *PPH* con *stencils* de 4 nodos puede también ser aplicado, pero únicamente para orden 3 en valores puntuales y 2 en medias en celda. Finalmente se han estudiado algunas características como el orden de la reconstrucción, funciones límite y aplicación al diseño de curvas. Para valores puntuales se obtienen funciones límite de mayor suavidad que para medias en celda y por tanto son más adecuadas para ser utilizadas en esquemas de multirresolución.

En segundo lugar se han estudiado los *mínimos cuadrados* como reconstrucción dentro de esquemas multirresolutivos basados en discretizaciones por valores puntuales y por medias en celda. Aunque en rigor no podemos hablar de operadores reconstrucción, ya que no cumplen la propiedad $\mathcal{D}_k \mathcal{R}_k = I_{V^k}$, los hemos incluido en los esquemas de multirresolución a fin de conocer su comportamiento. La propiedad de consistencia de la multirresolución a *la Harten* no se cumple de manera natural para medias en celda debido a que no se interpola, pero puede conseguirse forzando a que cumpla la ecuación correspondiente, siendo viable utilizarla dentro de esquemas multiescala y pudiéndose adaptar las ideas *ENO* y *SR* para mejorar las aproximaciones en las cercanías de los saltos. Al imponer consistencia las aproximaciones empeoran notablemente. Se han ofrecido algunas alternativas, aunque no reportan resultados satisfactorios, siendo la mejor opción forzar consistencia e incluir adaptatividad mediante la técnica *SR*. Para *WENO* no ha podido determinarse ningún caso que cumpla los requisitos de esta técnica, ni para valores puntuales ni medias en celda. En cuanto a *PPH* para *stencils* de 4 nodos sólo ha sido posible aplicarlo en valores puntuales de orden 2 y 3. Se han estudiado sus funciones límite obteniendo resultados de escasa suavidad, al igual que las curvas generadas a partir de una serie de puntos de control.

En ambas reconstrucciones se ha hecho incapié en el cálculo mediante filtros en todos los pasos de la reconstrucciones, tanto en el cálculo de los elementos a reconstruir como de las funciones intermedias necesarias para aplicar las técnicas *ENO* y *SR*.

En capítulos posteriores se estudiará la viabilidad de estas reconstrucciones para eliminar ruido de funciones discontinuas compuestas por trozos suaves, aprovechando sus capacidades de aproximación.

3

Detección de discontinuidades

3.1

Introducción

La detección de discontinuidades en señales digitales es una materia de estudio que tiene interés por sí misma. Su utilidad abarca campos como el procesamiento digital de imágenes, tratamiento de señales acústicas y dinámica de fluidos, entre otras. En métodos de compresión e interpolación, la localización de las discontinuidades permite evitar el llamado *fenómeno de Gibbs*, adaptando los algoritmos cuando nos encontramos ante una discontinuidad. Nosotros lo aplicaremos como paso previo en problemas de eliminación de ruido.

Para la detección de discontinuidades existen multitud de métodos y enfoques pero nos centraremos en estudiar los relacionados con las teorías expuestas en el presente trabajo. Trataremos en primer lugar los métodos de detección basados en interpolación adaptativa *ENO* y en magnitud de coeficientes *wavelet* para funciones sin ruido. Los analizaremos detalladamente e introduciremos modificaciones para que localicen únicamente discontinuidades reales. En segundo lugar nos centraremos en funciones con ruido, para las cuales los métodos de funciones suaves no sirven. Estudiaremos algunos métodos específicos para dichas funciones, propondremos nuevos esquemas y realizaremos comparativas a fin de decidir cuál proporciona mejores resultados.

Es importante señalar que la localización de discontinuidades es independiente del uso posterior de esta información. Así pues, es interesante disponer de localizadores fiables de discontinuidades que puedan ser aplicados a un amplio abanico de problemas.

3.2

Métodos de detección de discontinuidades para funciones sin ruido

3.2.1

Interpolación adaptativa: técnica *ENO*

La técnica *ENO* permite seleccionar para cada intervalo de trabajo el *stencil* de menor diferencia dividida. Con esto se evita utilizar *stencils* que crucen discontinuidades, mejorando la aproximación alrededor de éstas. Una evolución de lo anterior es la técnica *SR* donde se localiza la posición de esquinas dentro de una celda singular y se extienden los interpolantes adyacentes hasta el punto de discontinuidad.

Las técnicas *ENO* y *SR* están diseñadas como métodos de mejora de la aproximación en discontinuidades y no como localizadores

de éstas. Si se produce una situación donde es posible una mejora al suponer que existe una discontinuidad, se aplica entonces una modificación del algoritmo estándar. Por tanto, el localizar algunas falsas discontinuidades no plantea un error grave. En esta sección utilizaremos el algoritmo *SR* para hallar la posición de las discontinuidades, introduciendo las modificaciones necesarias para que cubran mayor parte de situaciones de detección que los métodos de partida.

Algoritmos

En el capítulo 1 se describió la técnica *ENO* y *SR*. Volvemos aquí a exponer estas ideas pero centrándonos en la detección. Los algoritmos *ENO* se utilizan aplicando varios niveles, aunque aquí sólo nos interesa uno. En lo que sigue, $(x_i, f_i)_{i=1}^N$ representará la función de partida, $f_i = f(x_i)$, y $\mathcal{S}_i^{ENO} = \{x_{s_i-1}, \dots, x_{s_i+r-1}\}$ el *stencil* *ENO* para el intervalo $I_i = [x_{i-1}, x_i]$.

Valores puntuales

Al trabajar con valores puntuales, las únicas discontinuidades que podemos localizar con seguridad serán las esquinas. Los saltos pueden no ser detectados debido a que la función $G_i(x)$ puede no cambiar de signo en los extremos de un intervalo que contenga un salto. El algoritmo que se sigue es:

1.- Para cada i , calcular el *stencil* *ENO* asociado (\mathcal{S}_i^{ENO}) mediante una elección jerárquica, Algoritmo 1.1 (*ENO-VP-I*) o no jerárquica, Algoritmo 1.2 (*ENO-VP-II*).

2.- Detectar las posibles *celdas singulares*: si $\mathcal{S}_{i-1}^{ENO} \cap \mathcal{S}_{i+1}^{ENO} = \emptyset$, etiquetar la celda $I_i = [x_{i-1}, x_i]$ como *sospechosa* de contener una singularidad.

3.- Para cada celda sospechosa definir la función

$$G_i^{\overline{IP}}(x) := \overline{q}_{i+1}^{\overline{IP}}(x; f^{k-1}, r) - \overline{q}_{i-1}^{\overline{IP}}(x; f^{k-1}, r), \quad (3.1)$$

donde $\overline{q}^{\overline{IP}}$ es el polinomio que interpola a $f(x)$ en \mathcal{S}_i^{ENO} .

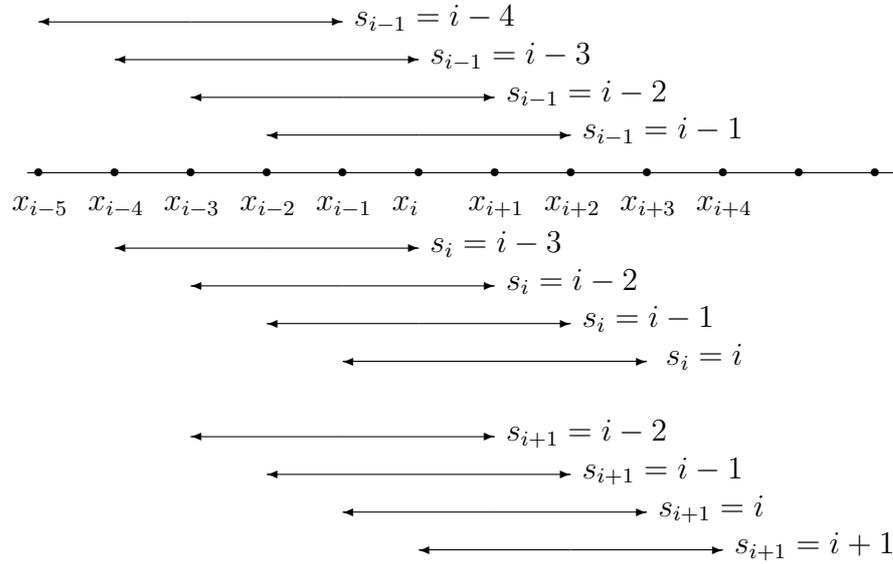


Figura 3.1: Posibles elecciones para la técnica ENO de stencils de 5 nodos para los intervalos I_{i-1} , I_i y I_{i+1} .

Si $G_i^{\overline{TP}}(x_{i-1})G_i^{\overline{TP}}(x_i) < 0 \Rightarrow$ existe una raíz θ_i de $G_i^{\overline{TP}}(x)$, y etiquetamos la celda como *singular*, es decir, contiene una discontinuidad.

Mediante el algoritmo anterior podemos detectar las discontinuidades esquina que no se hallen en puntos del *grid*.

Supongamos, por ejemplo, que estamos trabajando con *stencils* de cinco nodos. Los diferentes *stencils* que podemos elegir para los intervalos I_{i-1} , I_i , I_{i+1} , los podemos ver en la Figura 3.1. Los casos donde la detección puede ser fallida son:

- Si en x_{i-1} hay una esquina y la elección ENO es tal que $s_{i-1} = i - 4$ y $s_{i+1} = i$, se cumple que $\mathcal{S}_{i-1}^{ENO} \cap \mathcal{S}_{i+1}^{ENO} = x_{i-1}$ y por tanto la detección falla.
- Si en x_{i-1} hay una esquina, $G_i^{\overline{TP}}(x_{i-1}) = 0$ (teóricamente) y puede fallar la condición $G_i^{\overline{TP}}(x_{i-1})G_i^{\overline{TP}}(x_i) < 0$.

El primer punto se soluciona sin más que incluir la posibilidad de que la intersección de *stencils* sea un punto. Para ello empleamos la siguiente notación: $der(s_i)$ es el índice del extremo derecho

del intervalo ENO para la celda I_i . Es decir, si r es el grado del polinomio \overline{IP} , $der(s_i) = s_i + r - 1$ y $izq(s_i) = s_i - 1$. Con esta notación, la condición $\mathcal{S}_{i-1}^{ENO} \cap \mathcal{S}_{i+1}^{ENO} = \emptyset$ será sustituida por $der(s_{i-1}) \leq izq(s_{i+1})$.

Para el segundo punto, cuando cabe la posibilidad de esquina en x_{i-1} utilizamos la condición $G_{i-1}^{\overline{IP}}(x_{i-2}) \cdot G_{i-1}^{\overline{IP}}(x_i) < 0$.

Tras dichas modificaciones obtenemos el Algoritmo 3.1. Veamos que cubre todos los casos:

1.- La esquina cae en (x_{i-1}, x_i) . Como $der(s_{i-1}) < izq(s_{i+1})$, entra en el primer **si**. Además $G_i^{\overline{IP}}(x_{i-1}) \cdot G_i^{\overline{IP}}(x_i) < 0$, $|G_i^{\overline{IP}}(x_{i-1})| > 0$ y $|G_i^{\overline{IP}}(x_i)| > 0$ ya que la esquina está en el interior del intervalo de trabajo. Luego se detecta correctamente.

2.- La esquina cae en x_{i-1} . Se cumple que $der(s_{i-1}) \leq izq(s_{i+1})$ con lo que entra en el primer **si**. Dado que $|G_i^{\overline{IP}}(x_{i-1})| = 0$ no cumple el segundo **si**, aunqu sí el tercero. La detección es correcta.

3.- La esquina cae en x_i . En el paso i no se detectará, ya que no cumple las condiciones, pero sí será detectada en el paso $i + 1$.

Algoritmo 3.1 LOCALIZA-ENO-SR-VP: Localización de esquinas mediante ENO -SR y valores puntuales.

Entrada: $\{\bar{f}_i\}_{i=1}^N$, los datos iniciales; $\{s_i\}_{i=1}^N$, asignación ENO

Salida: Localización de las esquinas

- 1: **para** $i = 2, \dots, N - 1$ **hacer**
 - 2: **si** $der(s_{i-1}) \leq izq(s_{i+1})$ **entonces**
 - 3: **si** $G_i^{\overline{IP}}(x_{i-1}) \cdot G_i^{\overline{IP}}(x_i) < 0$ & $der(s_{i-1}) < izq(s_{i+1})$ &
 $|G_i^{\overline{IP}}(x_{i-1})| > 0$ & $|G_i^{\overline{IP}}(x_i)| > 0$ **entonces**
 - 4: Esquina en (x_{i-1}, x_i)
 - 5: **si no si** $G_{i-1}^{\overline{IP}}(x_{i-2}) \cdot G_{i-1}^{\overline{IP}}(x_i) < 0$ & $der(s_{i-1}) = izq(s_i)$ &
 $|G_{i-1}^{\overline{IP}}(x_{i-1})| = 0$ & $|G_{i-1}^{\overline{IP}}(x_{i-2})| > 0$ & $|G_{i-1}^{\overline{IP}}(x_i)| > 0$ **entonces**
 - 6: Esquina en el punto x_{i-1}
 - 7: **fin si**
 - 8: **fin si**
 - 9: **fin para**
-

NOTA 3.1. En la práctica no podemos asegurar las igualdades a 0, ya que trabajamos con precisión finita. Al programarlo debemos sustituir el 0 por ϵ . En nuestro caso tomamos $\epsilon = 10^{-10}$.

Teniendo en cuenta la Nota 3.1, advertimos que las discontinuidades muy próximas a x_{i-1} se pueden detectar como discontinuidades en x_{i-1} . Esto no es un problema, más bien una ventaja, ya que es más útil detectar como discontinuidad en un punto del *grid* una esquina muy próxima a dicho punto, que detectarla en el interior de un intervalo sin conocer su localización. La magnitud de dicho ϵ determinará un pequeño entorno de cada punto del *grid* donde se localizarán las esquinas como discontinuidades que caen justo en puntos del *grid*.

Las condiciones sobre $G_{i-1}^{\overline{IP}}$ no se pueden sustituir por condiciones sobre $G_i^{\overline{IP}}(x)$. En el siguiente ejemplo vemos porqué. Supongamos que trabajando con *stencils* de longitud 4 nos encontramos con la situación mostrada en la Figura 3.2: $s_{i-1} = i - 4$, $s_i = i$ y $s_{i+1} = i$. Con esto, puede detectarse erróneamente una esquina en el punto del *grid* ya que:

- $G_i^{\overline{IP}}(x_i)$ y $G_i^{\overline{IP}}(x_{i-2})$ pueden variar de signo, con lo cual $G_i^{\overline{IP}}(x_i) \cdot G_i(x_{i-2}) < 0$.
- $der(s_{i-1}) = izq(s_{i+1})$.
- $|G_i^{\overline{IP}}(x_{i-1})| < eps$.
- $|G_i^{\overline{IP}}(x_i)| > eps$.
- $|G_i^{\overline{IP}}(x_{i-2})| > eps$.

Al utilizar $G_{i-1}^{\overline{IP}}(x)$ en vez de $G_i^{\overline{IP}}(x)$ se asegura que los polinomios $q_{s_i}^{\overline{IP}}$ y $q_{s_{i-2}}^{\overline{IP}}$ se construyan con *stencils* que no crucen la discontinuidad. Si la discontinuidad es de salto, no se cumplirá que $|G_{i-1}^{\overline{IP}}(x_{i-1})| < eps$ y por tanto no se entrará en el **si** de detección de esquinas en puntos del *grid*. Si la discontinuidad es una esquina en un punto del *grid* puede comprobarse que será detectada correctamente utilizando las condiciones sobre $G_{i-1}^{\overline{IP}}$.

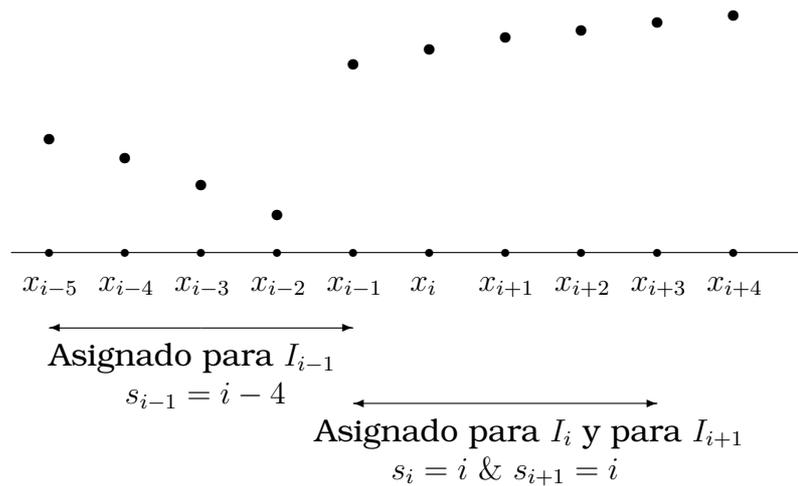


Figura 3.2: Ejemplo de posible asignación de stencils para una función con un salto que da lugar a una detección de esquina en el punto x_{i-1} si utilizásemos condiciones sobre $G_i^{\overline{TP}}(x)$ en lugar de hacerlo sobre $G_{i-1}^{\overline{TP}}(x)$ en el Algoritmo 3.1, LOCALIZA-ENO-SR-VP.

NOTA 3.2. Las comprobaciones sobre la magnitud de G son necesarias, pues se pueden dar casos en que se cumplen el resto de condiciones, aunque realmente no estemos ante una esquina.

Mediante el Algoritmo 3.1 se detectan correctamente discontinuidades esquinas, diferenciando si se trata de una esquina localizada en el interior de un intervalo o en un extremo. Si deseamos incluir la detección de saltos debemos hacer uso del concepto de medias en celda.

Medias en celda

Como vimos en el capítulo de preliminares, disponemos de una herramienta matemática que permite transformar saltos en esquinas. Al suponer que los valores de f (que denotaremos \bar{f}) son las medias en celda de una función F , se obtiene que en I_i hay una esquina para F si y sólo si en I_i hay un salto para \bar{f} .

Por tanto, la estrategia consiste en detectar esquinas en F me-

diante ENO-SR que equivaldrán a saltos en \bar{f} . Además recordemos que F nunca se forma explícitamente, sino que trabajamos sobre los valores de \bar{f} .

Para detectar las esquinas de F tenemos dos opciones:

1. Trabajar con los valores puntuales de F y el Algoritmo 3.1.
2. Trabajar con \bar{f} y utilizar $G^{\overline{IP}'}$.

Nosotros nos hemos decantado por la primera opción por simplicidad, aunque la segunda es igualmente válida.

Utilizando los algoritmos que aparecen en [16], [15], la detección de esquinas en F es como sigue:

- 1.- Calcular los *stencils* ENO según el Algoritmo 1.3 o el 1.4.
- 2.- Detectar las posibles *celdas singulares*: si $\mathcal{S}_{i-1}^{ENO} \cap \mathcal{S}_{i+1}^{ENO} = \emptyset$, etiquetar la celda como *sospechosa* de contener una singularidad.
- 3.- Para cada celda sospechosa definir la función

$$G_i^{\overline{IC}}(x) := q_{i+1}^{\overline{IP}}(x; F^{k-1}, r) - q_{i-1}^{\overline{IP}}(x; F^{k-1}, r). \quad (3.2)$$

Si $G_i^{\overline{IC}}(x_{i-1})G_i^{\overline{IC}}(x_i) < 0 \Rightarrow$ existe una raíz θ_i de $G_i^{\overline{IC}}(x)$, y etiquetamos la celda como *singular*.

El algoritmo presenta el siguiente problema: una discontinuidad de salto en $[x_{i-1}, x_i]$ se convierte en una esquina de F que cae justo en x_{i-1} . Con lo cual se tiene que $G_i^{\overline{IC}}(x_{i-1}) \approx 0$ y la condición $G_i^{\overline{IC}}(x_{i-1})G_i^{\overline{IC}}(x_i) < 0$ puede no cumplirse. Por ello la sustituiremos por $G_i^{\overline{IC}}(x_{i-2})G_i^{\overline{IC}}(x_i) < 0$

Con el fin de localizar saltos hemos introducido una nueva condición al Algoritmo 3.1, obteniendo el Algoritmo 3.2. En caso de no cumplir las condiciones para esquina consideramos el intervalo como sospechoso para contener una esquina para F interpretando en este caso f como las medias en celda de F . Entonces si efectivamente contiene una esquina para F , concluimos que existe un salto para f . De esta manera conseguimos localizar en un mismo algoritmo esquinas interiores a un intervalo, esquinas en

puntos del *grid* y saltos, considerando los datos iniciales como valores puntuales o medias en celda según nos interese.

Algoritmo 3.2 LOCALIZA-ENO-SR-VP-MC: Localización de esquinas y saltos mediante *ENO-SR*, valores puntuales y medias en celda.

Entrada: $\{\bar{f}_i\}_{i=1}^N$, los datos iniciales; $\{s_i\}_{i=1}^N$, asignación *ENO*

Salida: Localización de saltos y esquinas

```

1: para  $i = 2, \dots, N - 1$  hacer
2:   si  $der(s_{i-1}) \leq izq(s_{i+1})$  entonces
3:     si  $G_i^{\overline{TP}}(x_{i-1}) \cdot G_i^{\overline{TP}}(x_i) < 0$  &  $der(s_{i-1}) < izq(s_{i+1})$  &
        $|G_i^{\overline{TP}}(x_{i-1})| > 0$  &  $|G_i^{\overline{TP}}(x_i)| > 0$  entonces
4:       Esquina en  $(x_{i-1}, x_i)$ 
5:     si no si  $G_{i-1}^{\overline{TP}}(x_{i-2}) \cdot G_{i-1}^{\overline{TP}}(x_i) < 0$  &  $der(s_{i-1}) = izq(s_i)$  &
        $|G_{i-1}^{\overline{TP}}(x_{i-1})| = 0$  &  $|G_{i-1}^{\overline{TP}}(x_{i-2})| > 0$  &  $|G_{i-1}^{\overline{TP}}(x_i)| > 0$  entonces
6:       Esquina en el punto  $x_{i-1}$ 
7:     si no si  $G_i^{\overline{TC}}(x_{i-2}) \cdot G_i^{\overline{TC}}(x_i) < 0$  &  $|G_i^{\overline{TC}}(x_{i-2})| > 0$  &  $|G_i^{\overline{TC}}(x_i)| >$ 
        $0$  &  $|G_i^{\overline{TC}}(x_{i-1})| = 0$  entonces
8:       Salto en  $[x_{i-1}, x_i]$ 
9:     si no
10:      Celda  $[x_{i-1}, x_i]$  sospechosa pero no discontinua
11:    fin si
12:  fin si
13: fin para

```

Experimentos numéricos

Aplicamos los distintos métodos a dos tipos de funciones, ambas con 65 nodos iniciales. La primera de ellas posee una discontinuidad esquina localizada exactamente en un punto del *grid* y otra discontinuidad de salto. La segunda contiene un salto y una esquina localizada en un punto interior a un intervalo, definida en (1.44). Obtenemos la Figura 3.3. En (a) y (b) utilizamos el esquema *LOCALIZA-ENO-SR-VP*, Algoritmo 3.1. La localización de esquinas

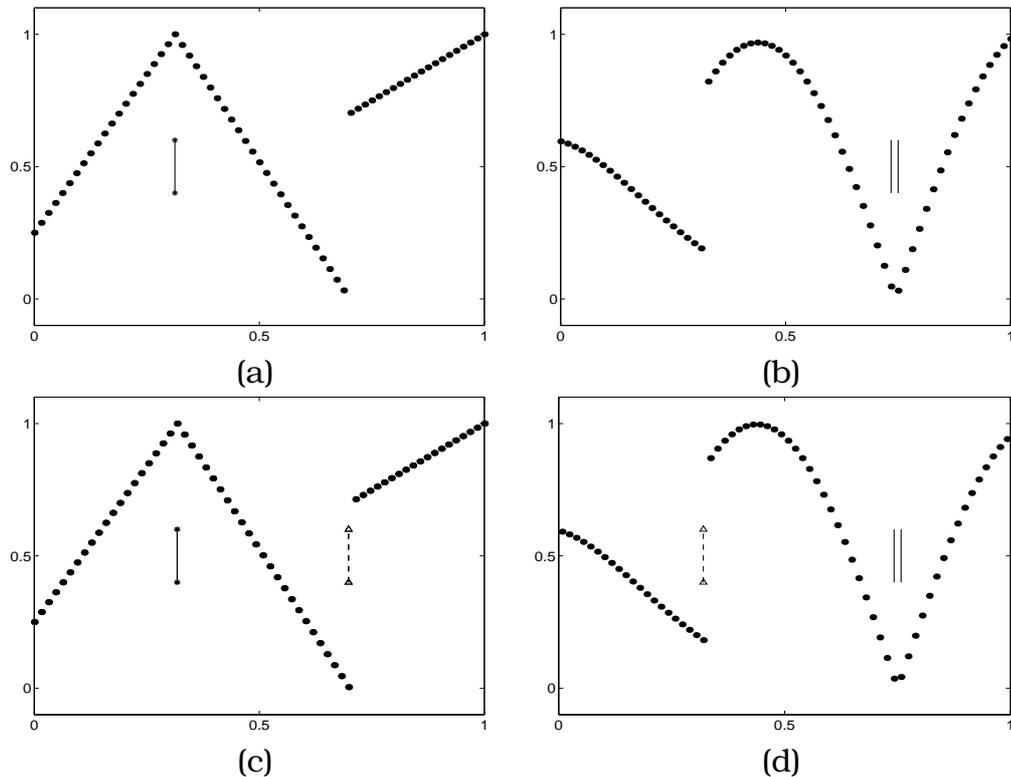


Figura 3.3: Aplicación del Algoritmo 3.1 (*LOCALIZA-ENO-SR-VP*) en (a) y (b), y el Algoritmo 3.2 (*LOCALIZA-ENO-SR-MC*) en (c) y (d), a una función con una esquina en un punto del grid y un salto, y a otra con un salto y una esquina en el interior de un intervalo del grid, ambas con 65 nodos iniciales. Una doble línea continua significa que se ha detectado una esquina en un punto interior a un intervalo); una línea continua con * corresponde a una esquina en un punto del grid) y una línea discontinua con Δ será un salto.

es correcta, discriminando si se trata de esquinas en puntos del *grid* o interiores a un intervalo. En las Figuras (c) y (d) aplicamos el esquema *LOCALIZA-ENO-SR-VP-MC*, Algoritmo 3.2 lo cual permite además localizar los saltos.

3.2.2

Magnitud de coeficientes *wavelet*

Los coeficientes de altas frecuencias que se obtienen al trabajar con la transformada *wavelet* contienen información acerca de la suavidad de la función. Así, aquellos coeficientes calculados a partir de una región suave son de pequeña magnitud, mientras que los calculados a partir de regiones que contienen discontinuidades son de magnitud elevada. Aprovechando este hecho podemos localizar la posición exacta de las discontinuidades comparando la magnitud de dichos coeficientes. Este método de localización ha sido desarrollado por Chan y Zhou en [26] para ser aplicado en la transformada *ENO-wavelet*.

En lo sucesivo supondremos que la función *wavelet* utilizada posee p momentos nulos y que la discretización aplicada es suficientemente fina como para que cada discontinuidad esté localizada al menos a una distancia de un *stencil* y un punto de otra. O dicho de manera más rigurosa:

Sea una función $f(x)$, llamamos D al conjunto de sus discontinuidades, es decir:

$$D = \{x_i : f(x) \text{ es discontinua en } x_i\} \quad (3.3)$$

DEFINICIÓN 3.1. *Dado un filtro wavelet con stencils de longitud $l + 1$, decimos que la proyección de f en el espacio V_j , con paso espacial $h = 2^{-j}$ satisface la Propiedad de Separación de Discontinuidades (DSP), si $(l + 2) \cdot h < t$, con $t = \min\{|x_i - x_j| : x_i, x_j \in D\}$ y D definido en (3.3).*

Luego esto nos dice que en una proyección que satisface la DSP no debe haber dos *stencils* consecutivos que contengan dos discontinuidades.

Pasemos a estudiar cómo influyen las discontinuidades en la magnitud de los coeficientes de altas frecuencias para poder diseñar el método de localización. En [20], [69], se demuestra que en las regiones suaves se cumple $|\beta_{j,i}| = |f^{(p)}(x)| O(\Delta x^p)$, mientras que

en las que hay una discontinuidad en la función o en sus derivadas, $|\beta_{j,i}| = |f^{(p-1)}(x)| O(\Delta x^{p-1})$. Así, comparando las magnitudes de los $\beta_{j,i}$, podemos encontrar la posición exacta de las discontinuidades.

Veamos cómo lo hacemos. En las regiones suaves, desarrollando por Taylor, se cumple que $|\beta_{j,i}| = (1 + O(\Delta x)) |\beta_{j,i-1}|$. Luego tomando un cierto $a > 1$, se cumplirá que $|\beta_{j,i}| \leq a |\beta_{j,i-1}|$. Si en $\beta_{j,i}$ hay una discontinuidad, pero no en $\beta_{j,i-1}$:

$$|\beta_{j,i}| = |f^{(p-1)}(x)| O(\Delta x^{p-1}) > a |f^{(p)}(x)| O(\Delta x^p) = |\beta_{j,i-1}|.$$

Posteriormente veremos qué a debemos elegir. También podemos razonar comparando cada $\beta_{j,i}$ con su posterior. Si $\beta_{j,i}$ contiene una discontinuidad, pero no $\beta_{j,i+1}$, se cumple: $|\beta_{j,i}| > a |\beta_{j,i+1}|$. Finalmente, si tanto $\beta_{j,i}$ como $\beta_{j,i+1}$ no contienen saltos: $|\beta_{j,i}| \leq a |\beta_{j,i+1}|$.

Con esta idea, veamos cómo hallamos la localización exacta del salto. Si el filtro tiene longitud $l + 1$, con $l = 2k - 1$, los $\beta_{j,i}$ están formados a partir de los siguientes *stencils*:

$$\begin{aligned} \beta_{j,i-1} &= \{x_{2i-2}, x_{2i-1}, \dots, x_{2i+l-3}, x_{2i+l-2}\}, \\ \beta_{j,i} &= \{x_{2i}, x_{2i+1}, \dots, x_{2i+l-1}, x_{2i+l}\}, \\ \dots &= \dots \\ \beta_{j,i+k} &= \{x_{2i+l+1}, x_{2i+l+2}, \dots, x_{2i+2l}, x_{2i+2l+1}\}. \end{aligned}$$

Supongamos que $|\beta_{j,i}| > a |\beta_{j,i-1}|$, entonces $\beta_{j,i}$ a sido calculado a partir de un *stencil* que cruza el salto. Como el *stencil* anterior no contiene el salto, sólo hay dos posibles posiciones donde podemos encontrar la discontinuidad: entre $\{x_{2i+l-2}, x_{2i+l-1}\}$ o entre $\{x_{2i+l-1}, x_{2i+l}\}$. Ahora bien, si el salto está entre $\{x_{2i+l-2}, x_{2i+l-1}\}$ habrá $(k-1)$ *stencils* afectados por la discontinuidad, en concreto $\beta_{j,m}$ para $i \leq m \leq (i+k-2)$. Sin embargo si el salto se encuentra entre $\{x_{2i+l-1}, x_{2i+l}\}$ entonces encontraremos k *stencils* afectados, ya que el $\beta_{j,i+k-1}$ también lo estará. Así pues, simplemente debemos comprobar si $\beta_{j,i+k-1}$ está o no afectado por el salto. Para ello debemos comparar $|\beta_{j,i+k-1}|$ y $|\beta_{j,i+k}|$. Con esto, podemos enunciar:

LEMA 3.1. Sean filtros wavelet de longitud $l + 1$, con $l = 2k - 1$. Dado un índice i , si $|\beta_{j,i-1}| \leq a |\beta_{j,i-2}|$, pero $|\beta_{j,i}| > a |\beta_{j,i-1}|$, entonces:

1. Si $|\beta_{j,i+k-1}| > a |\beta_{j,i+k}|$, hay k stencils consecutivos que contienen la discontinuidad y se halla en (x_{2i+l-1}, x_{2i+l}) .
2. Si $|\beta_{j,i+k-1}| \leq a |\beta_{j,i+k}|$, hay $(k - 1)$ stencils consecutivos que contienen la discontinuidad y se halla en (x_{2i+l-2}, x_{2i+l-1}) .

Para finalizar este apartado veremos cómo hemos de tomar el valor de a para localizar correctamente la discontinuidad. Como se ve en [20], [69], si la función es Lipschitz $\gamma \leq p$ en x , es decir $|f(x + \delta) - f(x)| \leq \delta^\gamma$ para cualquier δ pequeño, el correspondiente coeficiente wavelet de alta frecuencia es de orden $O(\Delta x^\gamma)$. Como habíamos señalado antes, $|\beta_{j,i}| = |f^{(p)}(x)| O(\Delta x^p)$. Con lo cual, tomando desarrollos de Taylor, se cumple que en las regiones suaves $|\beta_{j,i}| = (1 + O(\Delta x)) |\beta_{j,i-1}|$.

Sin embargo, si el $\beta_{j,i}$ ha sido calculado a partir de un stencil que cruza una discontinuidad, su orden es al menos uno menor que en las regiones suaves. En concreto, suponiendo que $f(x)$ tiene una discontinuidad en la derivada de orden m en un punto $x_0 \in (i\Delta x, (i+l)\Delta x)$ para algún entero i , utilizando desarrollos de Taylor, obtenemos en un entorno de x_0 :

$$f(x) = g(x) + \begin{cases} f^{(m)}(x_{o-})(x - x_0)^m + O(x - x_0)^{(m+1)} & x \leq x_0, \\ f^{(m)}(x_{o+})(x - x_0)^m + O(x - x_0)^{(m+1)} & x > x_0, \end{cases}$$

con

$$g(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{f^{(m-1)}(x_0)}{(m-1)!}(x - x_0)^{(m-1)}.$$

Utilizando que la función wavelet posee p momentos nulos:

$$\begin{aligned} |\beta_{j,i}| &= \left| \int f(x) \psi_{j,i}(x) dx \right| \\ &= \left| \int_{i\Delta x}^{x_0} (f^{(m)}(x_{o-})(x - x_0)^m + O(x - x_0)^{(m+1)}) \psi_{j,i}(x) dx + \right. \end{aligned}$$

$$\begin{aligned}
 & \left| \int_{x_0}^{(i+1)\Delta x} (f^{(m)}(x_0+)(x-x_0)^m + O(x-x_0)^{(m+1)}) \psi_{j,i}(x) dx \right| \\
 = & \left| [f^{(m)}(x_0)] \right| O(\Delta x^m). \tag{3.4}
 \end{aligned}$$

Con lo cual la elección de a debe ser la siguiente:

$$(1 + O(\Delta x)) \leq a \leq \min_x \{ |[f^{(m)}(x)]| O(\Delta x^{(m-p)}) \}. \tag{3.5}$$

Veamos que realmente cumple lo que queremos. Supongamos que $\beta_{j,i-1}$ no está afectado por una discontinuidad, entonces:

- Si $\beta_{j,i}$ no está afectado, $|\beta_{j,i}| = (1 + O(\Delta x)) |\beta_{j,i-1}|$ con lo cual, como $(1 + O(\Delta x)) \leq a$ se cumple que $|\beta_{j,i}| \leq a |\beta_{j,i-1}|$.
- Si $\beta_{j,i}$ está afectado, como $a \leq \min_x \{ |[f^{(m)}(x)]| O(\Delta x^{(m-p)}) \}$, tenemos que:

$$\begin{aligned}
 |\beta_{j,i}| &= |[f^{(m)}(x_0)]| O(\Delta x^{(m)}) \\
 &> \min_x \{ |[f^{(m)}]| O(\Delta x^{(m-p)}) \} |f^{(p)}(x)| O(\Delta x^p) \geq a |\beta_{j,i-1}|.
 \end{aligned}$$

De la misma manera podemos comprobar que se cumplen los resultados cuando $\beta_{j,i}$ está afectado por la discontinuidad pero $\beta_{j,i+1}$ no lo está.

NOTA 3.3. Para prevenir la inestabilidad numérica causada por β 's pequeños, en el algoritmo introduciremos la condición $|\beta_{j,i}| \geq \epsilon$, con ϵ un valor predefinido y mayor que cero.

NOTA 3.4. Una discontinuidad de la derivada m -ésima de intensidad menor que $O(\Delta x^{(p-m)})$ no será detectada por el método descrito anteriormente. Sin embargo, el error causado al no detectarse es relativamente pequeño.

El Lema 3.1 puede fallar en el siguiente supuesto: $\beta_{j,i}$ ha sido calculado a partir de un *stencil* que cruza un salto, pero $\beta_{j,i-2} = 0$. En este caso no se cumple que $|\beta_{j,i-1}| \leq a |\beta_{j,i-2}|$ y la detección falla. El caso descrito, aunque poco frecuente, puede darse si estamos trabajando con rectas. Para evitarlo proponemos modificar el Lema 3.1 de la siguiente manera:

LEMA 3.2. Sean filtros wavelet de longitud $l = 2k - 1$. Dado un índice i , si $(|\beta_{j,i-1}| \leq a |\beta_{j,i-2}| \vee (|\beta_{j,i-2}| = 0 \wedge |\beta_{j,i-1}| < \epsilon))$, pero $|\beta_{j,i}| > a |\beta_{j,i-1}|$, entonces (tomando ϵ como en la Nota 3.3):

1. Si $|\beta_{j,i+k-1}| > a |\beta_{j,i+k}|$, hay k stencils consecutivos que contienen el salto, con lo cual la discontinuidad está localizada entre $\{x_{2i+l-1}, x_{2i+l}\}$.
2. Si $(|\beta_{j,i+k-1}| \leq a |\beta_{j,i+k}| \vee (|\beta_{j,i+k}| = 0 \wedge |\beta_{j,i+k-1}| < \epsilon))$, hay $(k-1)$ stencils consecutivos que contienen el salto, con lo cual la discontinuidad está localizada entre $\{x_{2i+l-2}, x_{2i+l-1}\}$.

Ejemplos

Nuestro objetivo es localizar saltos y esquinas. Por tanto podemos utilizar una base *wavelet* con $p = 2$ momentos nulos, como por ejemplo *DB4*. En (3.5) hemos establecido unas cotas para la constante a . La cota inferior se conoce directamente al disponer del espaciado de la discretización. Sin embargo, la cota superior no es calculable, ya que $|f^{(m)}(x)|$ es desconocido. En la práctica el valor de a abarca un intervalo amplio de valores y determinará la magnitud de las discontinuidades que deseemos considerar. Una elección aceptable de a es 2 y será la que utilicemos en nuestros experimentos.

Aplicamos el método del Lema 3.2 a dos funciones que poseen saltos y esquinas. La Figura 3.4 muestra el resultado. La localización de saltos y esquinas es correcta, aunque en este caso no se discrimina entre ellas.

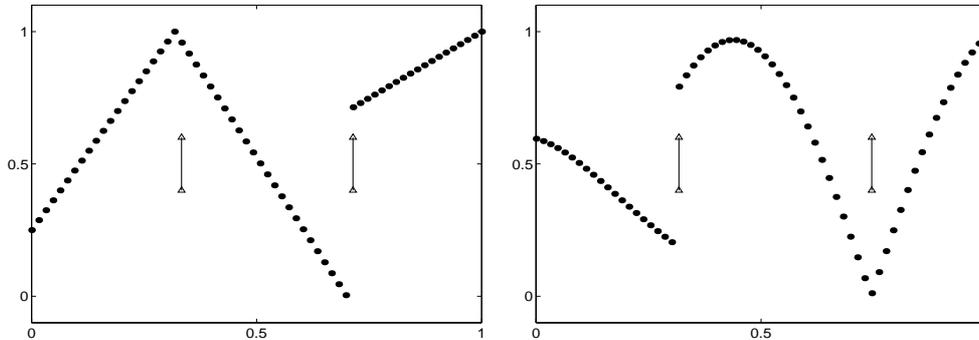


Figura 3.4: Discontinuidades detectadas según el Lema 3.2. Parámetros de detección: $a = 2$ y $\epsilon = 0,001$. La localización de la discontinuidad se representa mediante una línea vertical con triángulos en los extremos.

3.3

Métodos de detección de discontinuidades para funciones con ruido

En esta sección trabajaremos con funciones del siguiente tipo:

$$\tilde{f}(x) = f(x) + n(x),$$

donde $f(x)$ es una función discontinua compuesta por trozos suaves y $n(x)$ representa el ruido *blanco gaussiano*. El ruido *blanco* es una señal aleatoria (proceso estocástico) que se caracteriza porque sus valores de señal en dos instantes de tiempo diferentes no guardan correlación estadística. Por *gaussiano* entendemos que sigue una distribución de Gauss¹.

Para expresar la magnitud del ruido utilizamos la medida conocida como *SNR* (*Signal-to-Noise-Ratio*), [38]:

$$SNR = \frac{\text{potencia media de la seal}}{\text{potencia media del ruido}}. \quad (3.6)$$

¹En capítulos posteriores se justificará esta elección del tipo de ruido introducido.

Dado que SNR es una razón de dos potencias, se define a menudo en decibelios:

$$SNR(f, g) := 10 \log_{10} \frac{\sum_{i=1}^N f_i^2}{\sum_{i=1}^N (f_i - g_i)^2}, \quad (3.7)$$

Así cuando decimos que hemos introducido ruido de magnitud $SNR = c$ entendemos que se ha añadido *ruido blanco gaussiano*² y la relación señal-ruido entre la función de partida y la que contiene ruido es c .

SNR es la razón entre lo que se quiere (señal) y lo que no se quiere (ruido). Una SNR alta indica que la señal está menos corrompida por ruido, mientras que una SNR baja indica que la señal está muy corrompida por el ruido.

3.3.1

Combinación *ENO* y mínimos cuadrados

Las diferencias divididas retienen la información necesaria para huir de las discontinuidades en funciones suaves. Sin embargo al trabajar con funciones con ruido dicha información no sirve y debemos aplicar métodos específicos para funciones con ruido. En los trabajos de Daphna Mizrahi [60] se propone aplicar la idea *ENO*, pero sustituyendo las diferencias divididas por mínimos cuadrados. Basándonos en estas ideas, las adaptaremos a nuestra nomenclatura y estudiaremos su aplicabilidad.

Evitando *stencils* que crucen discontinuidades

Sea $q_i^{\overline{AP}^{nl, nr}}(x; f^{k-1}, r)$ el polinomio de grado r que calcula la reconstrucción por *aproximaciones* en x a partir de $[x_{i-nl}^{k-1}, \dots, x_{i+nr-1}^{k-1}]$. Llamando $n := nl + nr$, los n posibles *stencils* que contienen al punto x_i son $\{x_{i-(n-k)}^{k-1}, \dots, x_{i+k-1}^{k-1}\}$, $k = 1, \dots, n$.

Para decidir cuál es el *stencil* que no cruza la discontinuidad se utilizan dos medidas. Veamos cada una de ellas.

²En MATLAB® podemos introducir *ruido blanco gaussiano* con las instrucciones `randn`, `mvnrnd`, `wgn` y `awgn`.

Calculamos los siguientes términos para cada i :

$$E_1(x_i, k) \equiv \left| \frac{d^r}{dx^r} q_i^{\overline{\mathcal{AP}}_{n-k,k}}(x_i; f^{k-1}, r) \right|. \quad (3.8)$$

Dado que el polinomio $q_i^{\overline{\mathcal{AP}}_{n-k,k}}(x_i; f^{k-1}, r) = \sum_{i=1}^{k+1} a_i x^{i-1}$ es de grado k dicha medida será $E_1(x_i, k) \equiv |r! a_{r+1}|$. Calculamos k_1^* como sigue:

$$E_1(x_i, k_1^*) = \min \{E_1(x_i, 1), E_1(x_i, 2), \dots, E_1(x_i, n)\}.$$

Definimos $i_1(i) := i - (n - k_1^*)$. Si para trabajar con x_i utilizamos el *stencil* $\{x_{i_1(i)}, x_{i_1(i)+2}, \dots, x_{i_1(i)+n-1}\}$, nos evitamos cruzar discontinuidades.

Otra manera de seleccionar el *stencil* es fijándonos en los errores entre la aproximación y la función con ruido:

$$E_2(x_i, k_2) = \sum_{j=1}^n \left(q_i^{\overline{\mathcal{AP}}_{n-k,k}}(x_{i-(n-k)+j}; f^{k-1}, r) - \tilde{f}(x_{i-(n-k)+j}) \right)^2. \quad (3.9)$$

Como antes, calculamos k_2^* :

$$E_2(x_i, k_2^*) = \min \{E_2(x_i, 1), E_2(x_i, 2), \dots, E_2(x_i, n)\}. \quad (3.10)$$

Se define

$$i_2(i) := i - (n - k_2^*), \quad (3.11)$$

y el *stencil* asignado a x_i será $\{x_{i_2(i)}, x_{i_2(i)+1}, \dots, x_{i_2(i)+n-1}\}$.

Detección de discontinuidades

Definimos x_d como un punto de discontinuidad si, dado n (tamaño del *stencil*), se cumple:

$$i_1(d) = d \quad \wedge \quad i_1(d-1) = d - n, \quad (3.12)$$

$$i_2(d) = d \quad \wedge \quad i_2(d-1) = d - n. \quad (3.13)$$

La Figura 3.5 muestra todas las posibles elecciones de *stencils* que contienen a x_{i-1}^{k-1} y a x_i^{k-1} , donde es fácil ver que si la discontinuidad se halla en $d = i$ se cumplen (3.12) y (3.13).

El porqué utilizamos simultáneamente E_1 y E_2 proviene de la observación de casos prácticos. Si nos basamos únicamente en E_1 o en E_2 se localizan numerosas falsas discontinuidades. Al usar ambos criterios se reducen dichos errores.

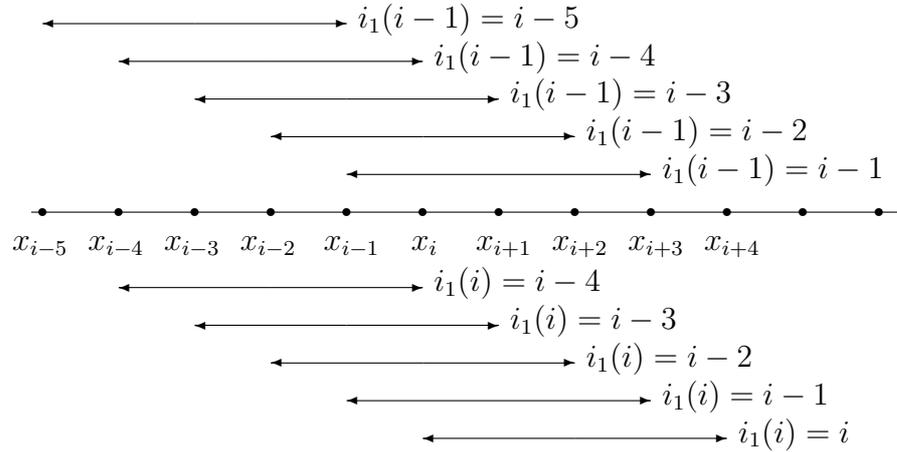


Figura 3.5: Posibles stencils de 5 nodos que incluyen el punto x_i y los que incluyen el punto x_{i-1} .

Ejemplos

En la Figura 3.6 aplicamos los métodos de detección a la función (1.44) con 256 nodos a la que hemos añadido ruido blanco gaussiano con un determinado SNR . En cada gráfico se han representado las discontinuidades detectadas mediante el criterio dado por E_1 , por E_2 y por ambos simultáneamente. Cada línea vertical que aparece en el punto x_i significa que el intervalo $[x_{i-1}, x_i]$ contiene una discontinuidad. En ambas hemos utilizado $r = 1$ como grado del polinomio por aproximación y $n = 10$ como longitud de los *stencils*. Se observa claramente la necesidad de utilizar simultáneamente ambos criterios, ya que por separado se localizan un elevado número de falsas discontinuidades. En la figura izquierda la localización del salto es correcta, mientras que en la derecha a parte de ésta se localizan dos falsas discontinuidades. Otra conclusión de este ejemplo es que la discontinuidad esquina queda sin ser localizada, ya que el ruido enmascara la naturaleza de la singularidad.

La elección del grado del polinomio y del tamaño del *stencil* va a ser fundamental para que el método detecte las discontinuidades de forma satisfactoria. Para estudiar esta cuestión hemos creado

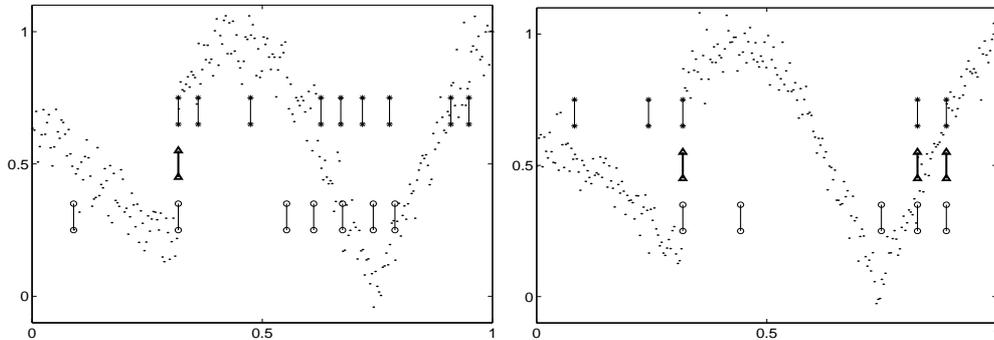


Figura 3.6: Detección de discontinuidades con $E1$ y $E2$. Tomamos 256 nodos de la señal (1.44) con $SNR = 23$ (izquierda) y $SNR = 25$ (derecha). Para los polinomios utilizamos tamaño del stencil $n = 10$ y grado $r = 1$. Una línea vertical en x_i indica que el intervalo $[x_{i-1}, x_i]$ contiene una discontinuidad según el criterio: '*' con $E1$ (3.12); 'o' con $E2$ (3.13); ' Δ ' simultáneamente con ambas medidas.

la Tabla 3.1 donde aplicamos la detección con diferentes parámetros a la función (1.44) a la que añadimos diferentes niveles de ruido. Aunque los resultados corresponden a ejemplos concretos podemos extrapolar algunas conclusiones. Es preferible el uso de grado lineal, ya que ofrece mejores resultados que el resto. En cuanto al tamaño del *stencil* una buena elección es $n = 10$.

3.3.2

Magnitud de coeficientes *wavelet* para funciones con ruido

En este apartado aplicaremos el método descrito en la sección 3.2.2 a una función con ruido. Para una función sin ruido se localizan correctamente tanto discontinuidades de salto como esquinas. Sin embargo al añadir ruido, la detección falla, como puede observarse en la Figura 3.7, izquierda. Es sencillo comprobar que no existe una elección de a y ϵ que dé buenos resultados, con lo cual debemos estudiar otras alternativas.

En la Figura 3.7, derecha, representamos los coeficientes de altas frecuencias de la función (1.44) con 256 nodos a la que he-

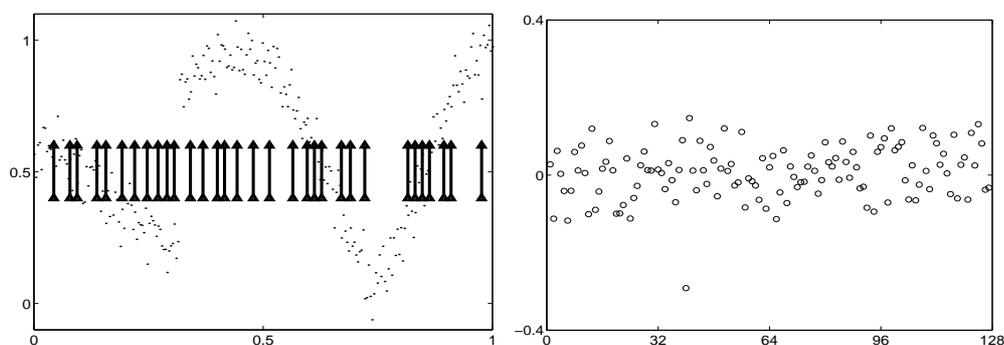


Figura 3.7: Izquierda: detección de discontinuidades (líneas verticales) siguiendo el Lema 3.1 con $a = 2$, $\epsilon = 0,001$ en la función (1.44) con ruido de magnitud $SNR = 23$. Derecha: coeficientes de altas frecuencias asociados.

mos añadido ruido con $SNR = 23$. Observamos que en el salto se produce un aumento detectable de la magnitud de los coeficientes $\beta_{j,i}$, aunque no para esquinas. Aunque el Lema 3.2 no sea viable en el caso de funciones con ruido, el anterior ejemplo pone de

| | | Ruido introducido | | | | | | | |
|-----|-----|-------------------|------|------------|------|------------|------|------------|------|
| | | $SNR = 21$ | | $SNR = 23$ | | $SNR = 25$ | | $SNR = 27$ | |
| k | n | D | FD | D | FD | D | FD | D | FD |
| 1 | 5 | C | 1 | C | 3 | C | 4 | C | 1 |
| 1 | 10 | C | 0 | C | 0 | C | 0 | C | 0 |
| 1 | 15 | C | 0 | C | 0 | C | 1 | C | 1 |
| 2 | 5 | C | 1 | C | 1 | I | 3 | C | 2 |
| 2 | 10 | I | 0 | C | 0 | C | 0 | I | 0 |
| 2 | 15 | I | 0 | I | 0 | I | 0 | I | 0 |
| 3 | 5 | I | 2 | I | 2 | C | 2 | I | 0 |
| 3 | 10 | I | 1 | I | 0 | I | 0 | I | 0 |
| 3 | 15 | I | 0 | I | 0 | I | 0 | I | 0 |

Tabla 3.1: Detección de la discontinuidad de salto de la función (1.44) con 256 nodos y ruido, mediante las medidas E_1 (3.12) y E_2 (3.13), variando el grado del polinomio por aproximación, r , y el tamaño del stencil, n . Las columnas rotuladas con D admiten dos indicadores: C (se ha detectado correctamente el salto), I (no se ha detectado el salto). Las columnas FD indican el número de falsas discontinuidades.

manifiesto que podemos utilizar la información de los coeficientes de altas frecuencias para la detección de discontinuidades tipo salto.

La detección de β 's de magnitud elevada constituye la base en *Wavelet Shrinkage Denoising* (p. ej. [63], [70], [35],) donde se utilizan cortes para los coeficientes de altas frecuencias. Para dichos cortes se define un umbral, λ , que decide si los coeficientes β constituyen información significativa de la señal. Así, los coeficientes mayores que λ se mantienen mientras que el resto se eliminan (*hard threshold*) o se reducen (*soft threshold*). Esto se describirá con mayor detalle en capítulos posteriores.

No existe una manera general y óptima para calcular λ en todos los casos, pero sí buenas aproximaciones como *VisuShrink* y *SureShrink*. *VisuShrink* cubre nuestras necesidades, ya que *SURE* mejora a *VISU* pero únicamente cuando trabajamos con más de un nivel de multirresolución.

VisuShrink, propuesto por Donoho y Johnstone [34] en 1994, ofrece el umbral óptimo en sentido asintótico cuando trabajamos con ruido blanco gaussiano. Minimiza la función $E \|Y_{Thres} - Y_{Orig}\|^2$, donde Y_{Thres} y Y_{Orig} son la aproximación y la función original respectivamente. Este valor de λ se conoce como *Universal Threshold* y viene dado por la expresión $\lambda^U = \sigma \sqrt{2 \log(N)}$, siendo N el tamaño de la señal original y σ^2 la varianza del ruido (suponemos que el ruido que hemos añadido sigue una $N(0, \sigma^2)$).

El dato que desconocemos es σ y es necesario estimarlo. A partir de la fórmula de una $N(0, \sigma^2)$ se puede deducir una aproximación razonablemente precisa mediante $\sigma \approx \frac{MAD}{0,6475}$, donde *MAD* es la mediana de las magnitudes de todos los coeficientes de altas frecuencias en la escala más fina.

A pesar de la optimalidad asintótica del método, en la práctica se pueden hallar mejores candidatos para λ . En la literatura referente a este método se utilizan otros cortes como: $\lambda = \sigma \sqrt{2 \log(N \cdot \log(N))}$ o $\lambda = \sigma \frac{\sqrt{2 \log(N)}}{\sqrt{N}}$.

El umbral *VISU* nos proporciona aquellos coeficientes de altas frecuencias que han sido calculados a partir de *stencils* que cruzan discontinuidades. A partir de esta información, la localización del salto es trivial. Por ejemplo para *DB4*, observando el esquema representado en la Figura 1.13, se deduce que:

- Si $|\beta_{1,i}| \geq \lambda$ y $|\beta_{1,i+1}| \geq \lambda$, la discontinuidad se halla entre f_{2i+2} y f_{2i+3} .
- Si $|\beta_{1,i-1}| \leq \lambda$ y $|\beta_{1,i}| \geq \lambda$ y $|\beta_{1,i+1}| \leq \lambda$, la discontinuidad se halla entre f_{2i+1} y f_{2i+2} .

Ejemplos

La Figura 3.8 muestra la detección obtenida mediante el corte *VISU* y base *DB4* de los coeficientes de altas frecuencias. Hemos tomado la función (1.44) a la que hemos añadido ruido con $SNR = 23$. En la figura izquierda se muestra la localización de las discontinuidades. En este caso el salto queda correctamente localizado pero no la esquina. En la figura derecha hemos representado los coeficientes de altas frecuencias y el corte *VISU*. Observamos que hay un único $\beta_{j,i}$ que sobrepasa el umbral *VISU*. Dicho coeficiente proviene de un *stencil* que cruza el salto, de ahí su magnitud elevada y en consecuencia la correcta detección. Sin embargo la esquina no proporciona variaciones significativas en la magnitud de los coeficientes y por tanto es indetectable.

En base a esta función, en los experimentos numéricos hemos comprobado que el método detecta correctamente los saltos cuando el ruido añadido es de magnitud $SNR \geq 25$. Cuando $SNR = 23$ la detección es buena en la mayoría de casos. Pero si $SNR \leq 21$ la detección falla.

3.3.3

Reconstrucción por aproximaciones de *stencil* creciente

En esta sección presentamos un nuevo detector de discontinuidades para funciones con ruido. La idea se basa en comparar los errores que se obtienen a medida que se van añadiendo nodos en el *stencil* para la reconstrucción por *aproximaciones* mientras mantenemos un grado fijo. Cuando al introducir un nuevo nodo, el error entre la función original y la reconstrucción sea elevado, significará que hemos cruzado un salto.

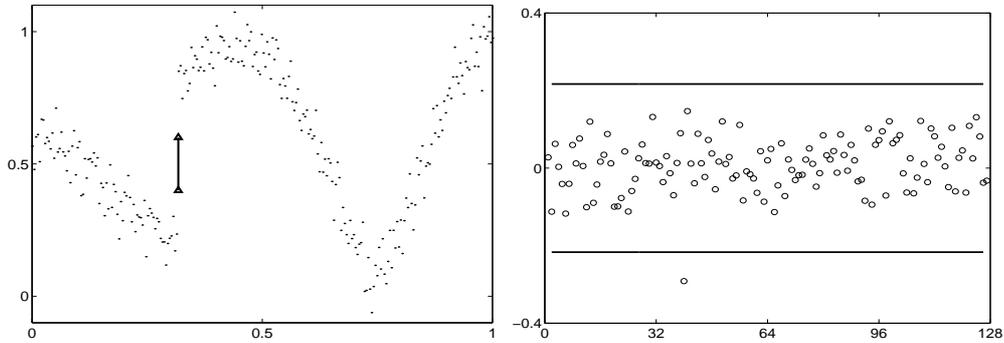


Figura 3.8: Izquierda: detección de discontinuidades (línea vertical) según la magnitud de los coeficientes wavelet que sobrepasan el umbral VISU. Derecha: representamos los coeficientes de altas frecuencias asociados, con 'o', y el umbral VISU, con una línea horizontal. La función de partida es (1.44) con 256 nodos a la que se le ha añadido ruido con $SNR = 23$.

Partiendo de los datos iniciales $\{(x_i, f_i)\}_{i=1}^N$, el esquema es el siguiente:

1. Tomamos un valor inicial, $i = 10$.
2. Tomamos los primeros i puntos y construimos el polinomio por aproximaciones con un grado fijo, por ejemplo $r = 3$.
3. Calculamos el error

$$e(i) = \sum_{j=1}^i \left(q_i^{\overline{\mathcal{AP}}_{i-1,1}}(x_j; f, r) - f_j \right)^2. \quad (3.14)$$

4. Añadimos un punto ($i = i + 1$) y volvemos al paso 2.

Cuando se detecte un aumento relevante del error cometido, ahí habrá una discontinuidad y volveremos a empezar de nuevo a la derecha de la discontinuidad detectada, iterando el proceso.

Veamos algún ejemplo sencillo. En la Figura 3.9 representamos los errores cometidos utilizando una función con un salto. Observamos que el salto produce un aumento detectable en el error. Evidentemente las esquinas no serán detectables, aunque es razonable al trabajar con funciones con ruido.

El siguiente paso es localizar salto. Para ello se propone lo siguiente:

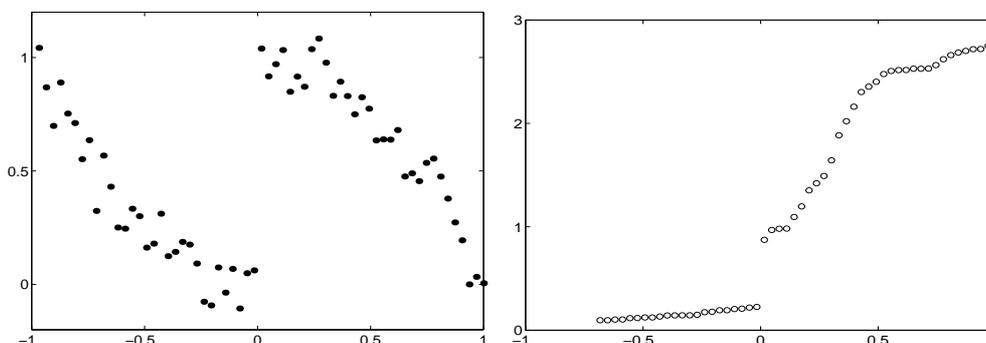


Figura 3.9: Función con un salto y vector de errores e_i según (3.14).

1. El vector d contiene la posición de las discontinuidades, de forma que si $d(k) = i$ entre $[x_{i-1}, x_i]$ hay un salto ($d(1) = 1$).
2. Calculamos las diferencias de los errores, $errdif(k) = |e(k) - e(k-1)|$ para $k = d(j-1), \dots, i-1$ (en el primer paso $j = 2$).
3. Si $errdif(i) > factor \cdot \max\{errdif(d(j-1)), \dots, i-1\} \Rightarrow d(j) = i$, es decir hay una discontinuidad entre $[x_{i-1}, x_i] \Rightarrow j = j + 1$, volver al paso 2.

NOTA 3.5. El valor de $factor$ determinará la magnitud de los saltos que consideremos. Nosotros utilizamos $factor = 3$ y puede interpretarse que una celda será discontinua si el error que se produce al incluir su extremo derecho es el triple que la media de los errores acumulados entre la función original y la reconstrucción por aproximaciones para valores puntuales de grado fijo.

Con este esquema y tomando unos 10 puntos como mínimo entre discontinuidad y discontinuidad, obtenemos los resultados de la Figura 3.10. Utilizamos a la derecha la función (1.44) y a la izquierda la función

$$f(x) = \begin{cases} -x \operatorname{sen}\left(\frac{3}{2}\pi x^2\right) & \text{si } -1 \leq x < -\frac{1}{3}, \\ |\operatorname{sen}(2\pi x)| & \text{si } -\frac{1}{3} \leq x < \frac{1}{3}, \\ 2x - 1 - \frac{1}{6}\operatorname{sen}(3\pi x) & \text{si } \frac{1}{3} \leq x < 1, \end{cases} \quad (3.15)$$

que contiene dos saltos y una esquina. La detección es correcta excepto en el caso de discontinuidades en la primera derivada. Del estudio de los resultados se deduce que el método reporta escasas falsas discontinuidades y funciona correctamente incluso con cantidades elevadas de ruido.

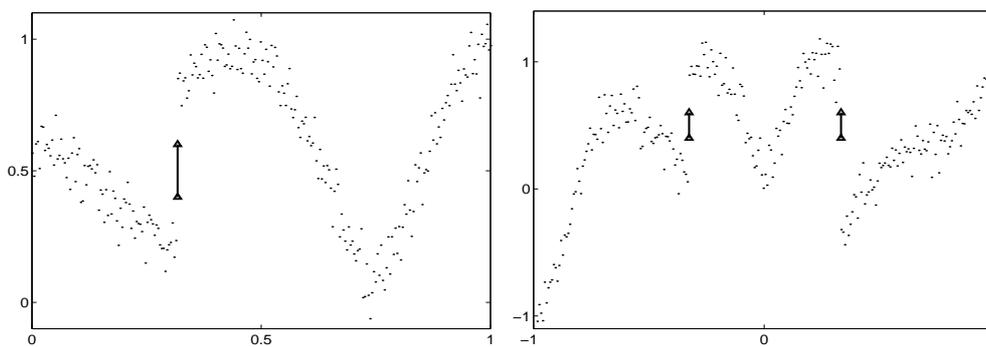


Figura 3.10: Discontinuidades detectadas (lineas verticales) mediante el método reconstrucción por aproximaciones de stencil creciente. A la izquierda utilizamos la función (1.44) y a la derecha la función (3.15). Se utiliza grado $r = 3$ y $SNR = 23$.

3.3.4

Comparativa entre los métodos de detección para funciones con ruido.

Para conocer qué localizadores proporcionan mejores resultados, hemos realizado³ la Tabla 3.2. En ella aplicamos a la función (1.44) los diversos localizadores descritos en esta sección. Para cada método, marcamos con una C si se ha localizado correctamente el salto e incluimos el número de falsas discontinuidades. La cantidad de ruido introducido varía desde $SNR = 21$ hasta $SNR = 27$.

Según lo estudiado en este capítulo, podemos deducir:

E_1 y E_2 . Es preferible utilizar grado 1 para el polinomio de la reconstrucción por aproximaciones. Es un método que localiza

³En <http://www.uv.es/anims/noguera/> podemos descargar una GUI para realizar experimentos de detección tanto para funciones con ruido como sin él.

| MÉTODO | Ruido introducido | | | | | | | |
|----------------------|-------------------|----|----------|----|----------|----|----------|----|
| | SNR = 21 | | SNR = 23 | | SNR = 25 | | SNR = 27 | |
| | D | FD | D | FD | D | FD | D | FD |
| E_1 & E_2 | C | 0 | C | 0 | C | 1 | C | 2 |
| $MW - VISU$ | I | 0 | C | 0 | C | 0 | C | 0 |
| $\overline{AP} - SC$ | C | 0 | C | 0 | C | 0 | C | 0 |

Tabla 3.2: Detección de la discontinuidad de salto de la función (1.44) con 256 nodos y ruido de diferentes magnitudes. Se aplican varios métodos: E_1 & E_2 (con grado 1 para el polinomio y tamaño del stencil 10), $MW - VISU$ magnitud de coeficientes wavelet que sobrepasan el umbral $VISU$, con la base $DB4$) y $\overline{AP} - SC$ (reconstrucción por aproximaciones por valores puntuales de stencil creciente, con grado del polinomio 3). Las columnas rotuladas con D admiten dos indicadores: C (se ha detectado correctamente el salto), I (no se ha detectado el salto). Las columnas FD indican el número de falsas discontinuidades detectadas.

correctamente las discontinuidades pero produce mayor número de falsas discontinuidades que el resto.

Corte $VISU$ para coeficientes wavelet. Es un método fiable que produce escasas falsas discontinuidades, aunque con ruido elevado no localiza algunas discontinuidades verdaderas.

Reconstrucción por aproximaciones de stencil creciente. Ofrece mejores resultados que sus rivales incluso para valores de ruido elevados, aunque su coste computacional es mayor.

3.4

Conclusiones

En este capítulo hemos centrado nuestra atención en la localización de discontinuidades en funciones compuestas por trozos suaves. En primer lugar hemos revisado los algoritmos de localización para funciones sin ruido basados en la técnica $ENO-SR$ y en la magnitud de coeficientes *wavelet*. Para los métodos basa-

dos en *ENO-SR*, [16], pueden darse casos en los que la detección sea fallida, tanto para valores puntuales como para medias en celda. Hemos introducido nuevas condiciones para que la detección sea correcta en cualquier situación en las que aparezcan saltos y esquinas, siempre y cuando estén lo suficientemente alejadas entre sí. El algoritmo definitivo basado en *ENO-SR* se ha construido combinando las ideas de valores puntuales y medias en celda. Se ha revisado la detección de discontinuidades basada en la magnitud de coeficientes *wavelet*, [26]. Hemos incluido condiciones para aumentar las situaciones de detección y hemos comprobado que los métodos funcionan correctamente.

En segundo lugar hemos estudiado la detección de discontinuidades para funciones con ruido. Se ha comprobado que los métodos para funciones sin ruido no sirven en este caso. Hemos estudiado los trabajos de Daphna Mizrachi [60], centrándonos en la detección de discontinuidades para funciones con ruido. Se ha comprobado que el método no produce resultados óptimos. A continuación se ha estudiado la detección de discontinuidades basadas en el umbral *VISU*, [34], de *Wavelet Shrinkage Denoising* según los coeficientes de altas frecuencias. Finalmente hemos diseñado un nuevo método de detección para funciones con ruido basado en la comparación de los errores que se obtienen entre la función original y su reconstrucción por aproximaciones utilizando un grado fijo pero aumentando la longitud del *stencil*. Al comparar experimentalmente los métodos se ha obtenido que este último es el que proporciona mejores resultados, ya que localiza eficientemente discontinuidades verdaderas y es aplicable a funciones con una elevada magnitud de ruido introducido.

En conclusión disponemos de métodos fiables para localizar discontinuidades tipo salto y esquina de funciones discontinuas compuestas por trozos suaves. Sin embargo, al introducir ruido únicamente podemos aspirar a localizar saltos, debiendo para ello utilizar métodos específicos para funciones con ruido.

4

ENO-wavelets

En este capítulo nos centramos en describir el método *ENO-wavelet*, presentado por T. F. Chan y H. M. Zhou en el año 2000, [26], [25]. Veremos una nueva manera de calcular sus coeficientes que proporciona ciertas ventajas respecto del esquema original, como por ejemplo generalizar el método de forma sencilla para orden arbitrario. La transformada *ENO-wavelet* se basa en combinar dos ideas: los algoritmos *wavelet* y la técnica *ENO*. Como vimos en la sección 1.4.2, si trabajamos con funciones discontinuas compuestas por trozos suaves y aplicamos la transformada *wavelet* lineal, aparece el *fenómeno de Gibbs* en las cercanías de los saltos, fruto de eliminar los coeficientes de altas frecuencias de magnitud elevada que surgen al utilizar *stencils* que crucen discontinuidades. Por otra parte, la técnica *ENO* consiste en elegir un *stencil* que no cruce las discontinuidades, según la magnitud de las diferencias divididas. En este caso no se utilizarán diferencias divididas pero sí la idea de evitar discontinuidades y emplear información

de un único lado de la discontinuidad. El método constará de dos pasos:

1. Localización de discontinuidades.
2. Cálculo de los coeficientes de altas y bajas frecuencias utilizando información de un sólo lado de la discontinuidad.

El primer punto fue estudiado con detalle en el capítulo anterior, tanto para funciones con ruido como sin él. Este paso es independiente del segundo y podemos utilizar el método de localización que más nos interese. La base *wavelet* utilizada¹ en todo el capítulo será *DB2p* y supondremos que se cumple a *Propiedad de Separación de Discontinuidades, DSP* (Definición 3.1).

En el segundo punto, el objetivo es utilizar información de un único lado de la discontinuidad. La idea más sencilla se basa en utilizar extrapolaciones en aquellos *stencils* que crucen discontinuidades en el presente nivel y calcular los datos del siguiente nivel con dichos valores. Es lo que se conoce como *función de extrapolación directa*.

4.1

Función de extrapolación directa

Supongamos que estamos en el nivel $j + 1$ y que tenemos localizada una discontinuidad entre $\alpha_{j+1,2i+l-2}$ y $\alpha_{j+1,2i+l-1}$, siendo $l + 1$ la longitud del *stencil*. Teniendo en cuenta las ecuaciones de la transformada *wavelet* rápida (1.77), se verían afectados los coeficientes $\alpha_{j,i+m}$, $\beta_{j,i+m}$ para $i \leq m \leq i + k - 2$ con $l = 2k - 1$. Para evitarlo, si estamos a la izquierda de la discontinuidad, extrapolamos $\hat{\alpha}_{j+1,2i+l-1}, \dots, \hat{\alpha}_{j+1,2i+2l-3}$ a partir de información a la izquierda y hallamos los coeficientes del siguiente nivel $\hat{\alpha}_{j,i+m}$, $\hat{\beta}_{j,i+m}$ para $i \leq m \leq i + k - 2$ con $l = 2k - 1$. Análogamente por la derecha, extrapolamos $\bar{\alpha}_{j+1,2i}, \dots, \bar{\alpha}_{j+1,2i+l-2}$ con información de la derecha y

¹El método es aplicable para otras bases, como por ejemplo la base de *Haar*, ver [26].

hallamos los coeficientes $\bar{\alpha}_{j,i+m}$, $\bar{\beta}_{j,i+m}$ para $i \leq m \leq i+k-2$ con $l = 2k - 1$.

Para mantener el orden de aproximación, si la base *wavelet* posee p momentos nulos, la extrapolación debe ser de grado $(p - 1)$. Si trabajamos con funciones sin ruido, podemos utilizar extrapolación por polinomios de Lagrange o por polinomios de Taylor, mientras que si la función de partida contiene ruido, podemos utilizar extrapolación por mínimos cuadrados o por polinomios de *interpolación aproximación* (introducidos en el capítulo 2).

En la posición (j, i) se obtienen 4 valores $\hat{\alpha}_{j,i}$, $\hat{\beta}_{j,i}$, $\bar{\alpha}_{j,i}$ y $\bar{\beta}_{j,i}$, en vez de 2 del método estándar. Esto es un problema de vital importancia ya que las secuencias de salida son diferentes que las de entrada, imposibilitando la inclusión de este método en los algoritmos piramidales propios de la transformada *wavelet*.

4.2

Extrapolación en el nivel grosero

Chan y Zhou plantean en [26] un método que permite utilizar información de un sólo lado de la discontinuidad y mantiene la misma longitud de las secuencias de salida tanto si hay discontinuidad o no. La idea consiste en invertir el orden de los pasos de la *función de extrapolación directa*. Situándonos a la izquierda de una discontinuidad, extrapolamos los coeficientes de bajas frecuencias que la crucen de los niveles más groseros, obteniendo $\hat{\alpha}_{j,m}$ con información de la izquierda y extendemos los coeficientes $\hat{\beta}_{j,m}$ a cero (o a otro valor predefinido). Posteriormente con estos datos hallamos los coeficientes del nivel $j + 1$. Pero esto plantea problemas en ciertos casos. En *DB2p* no siempre podemos predefinir $\hat{\alpha}_{j,m}$ y $\hat{\beta}_{j,m}$ simultáneamente, ya que pueden ser linealmente dependientes. Veamos un ejemplo. Supongamos que la discontinuidad se halla entre $\alpha_{j+1,2i+l-2}$ y $\alpha_{j+1,2i+l-1}$. Según (1.77) el cálculo de $\hat{\alpha}_{j,i}$ y $\hat{\beta}_{j,i}$ es:

$$\begin{pmatrix} \hat{\alpha}_{j,i} \\ \hat{\beta}_{j,i} \end{pmatrix} = \begin{pmatrix} \sum_{s=0}^{l-2} c_s \alpha_{j+1,2i+s} + c_{l-1} \hat{\alpha}_{j+1,2i+l-1} + c_l \hat{\alpha}_{j+1,2i+l} \\ \sum_{s=0}^{l-2} h_s \alpha_{j+1,2i+s} + h_{l-1} \hat{\alpha}_{j+1,2i+l-1} + h_l \hat{\alpha}_{j+1,2i+l} \end{pmatrix}$$

$$= \begin{pmatrix} \delta_{j,i} \\ \gamma_{j,i} \end{pmatrix} + A \begin{pmatrix} \hat{\alpha}_{j+1,2i+l-1} \\ \hat{\alpha}_{j+1,2i+l} \end{pmatrix}, \quad (4.1)$$

donde $\delta_{j,i} = \sum_{s=0}^{l-2} c_s \alpha_{j+1,2i+s}$ y $\gamma_{j,i} = \sum_{s=0}^{l-2} h_s \alpha_{j+1,2i+s}$ son datos conocidos por estar a la izquierda de la discontinuidad y A viene dada por

$$A = \begin{pmatrix} c_{l-1} & c_l \\ h_{l-1} & h_l \end{pmatrix}.$$

Para la transformada de Daubechies se cumple $\frac{h_{l-1}}{c_{l-1}} = \frac{h_l}{c_l}$, con lo cual A será singular. Para que el sistema tenga solución, la matriz ampliada

$$\begin{bmatrix} c_{l-1} & c_l & \hat{\alpha}_{j,i} - \delta_{j,i} \\ h_{l-1} & h_l & \hat{\beta}_{j,i} - \gamma_{j,i} \end{bmatrix}$$

debe de tener rango 1. En este caso podemos hallar $\hat{\alpha}_{j,i}$ a partir de $\hat{\beta}_{j,i}$ y viceversa:

$$\hat{\beta}_{j,i} = \gamma_{j,i} + \frac{h_l}{c_l} (\hat{\alpha}_{j,i} - \delta_{j,i}), \quad \hat{\alpha}_{j,i} = \delta_{j,i} + \frac{c_l}{h_l} (\hat{\beta}_{j,i} - \gamma_{j,i}).$$

Lo cual nos dice que no podemos predefinir $\hat{\alpha}_{j,i}$ y $\hat{\beta}_{j,i}$ a la vez. Tenemos dos posibles elecciones:

1. Extrapolamos los coeficientes $\hat{\alpha}_{j,i}$ con información de un sólo lado de la discontinuidad y después determinamos los $\hat{\beta}_{j,i}$.
2. Damos a $\hat{\beta}_{j,i}$ un valor predefinido (cero normalmente) y después hallamos los $\hat{\alpha}_{j,i}$.

Notar que en el primer caso sólo es necesario almacenar los $\hat{\beta}_{j,i}$, ya que el $\hat{\alpha}_{j,i}$ lo podemos deducir. En el segundo caso ocurre lo mismo, sólo hemos de almacenar los $\hat{\alpha}_{j,i}$, ya que los $\hat{\beta}_{j,i}$ poseen un valor predefinido. Por tanto, el número de coeficientes se mantiene constante.

Hemos de decidir con qué elección nos quedamos, incluso podemos usar la misma para ambos lados de la discontinuidad. Si por ejemplo elegiésemos la primera en ambos lados, almacenaríamos dos valores de $\hat{\beta}_{j,i}$ para cada *stencil* que cruce una discontinuidad. Esto plantea problemas, ya que para aplicar los algoritmos correctamente necesitamos igual número de α 's y β 's. Para mantener el

mismo número de α 's y β 's debemos tomar elecciones distintas a izquierda y derecha de la discontinuidad. Nosotros hemos optado por la elección 1 si nos encontramos a la izquierda, y por la 2 si estamos a la derecha. Con este método solucionamos los problemas de la *función de extrapolación directa*, ya que mantenemos el mismo tamaño de las secuencias de entrada y salida. En la Figura 4.1 exponemos un esquema del funcionamiento de la *extrapolación en el nivel grosero* para un caso de DB4.

Para poder reconstruir los datos, necesitamos almacenar la posición de las discontinuidades para saber en cada caso si los coeficientes de bajas y altas frecuencias son fruto de la *extrapolación en el nivel grosero*. Para cada *stencil* que cruce una discontinuidad, necesitamos un bit extra para almacenar dicha posición. Este almacenamiento extra (en coma flotante) por *stencil* que se necesita en la extrapolación de los coeficientes de bajas frecuencias es de $O(1)$ y de $O(l)$ para el cálculo de los correspondientes coeficientes de altas y bajas frecuencias, siendo l el tamaño del *stencil*. En general el coste extra sobre la transformada *wavelet* estándar es de orden $O(dl)$ siendo d el número de discontinuidades. Dado que el coste de la transformada estándar es de orden $O(ln)$, siendo n el número de datos iniciales, la ratio del coste extra respecto a la transformada estándar es $O(d/n)$, independiente de l , y despreciable cuando n es elevado.

4.3

Error de aproximación para *ENO-wavelets*

En el Teorema 1.1 enunciamos un resultado sobre el error de aproximación que se comete mediante la transformada *wavelet* al realizar una aproximación lineal. Para *ENO-wavelets* podemos encontrar un resultado similar, cuya demostración se encuentra en [26]:

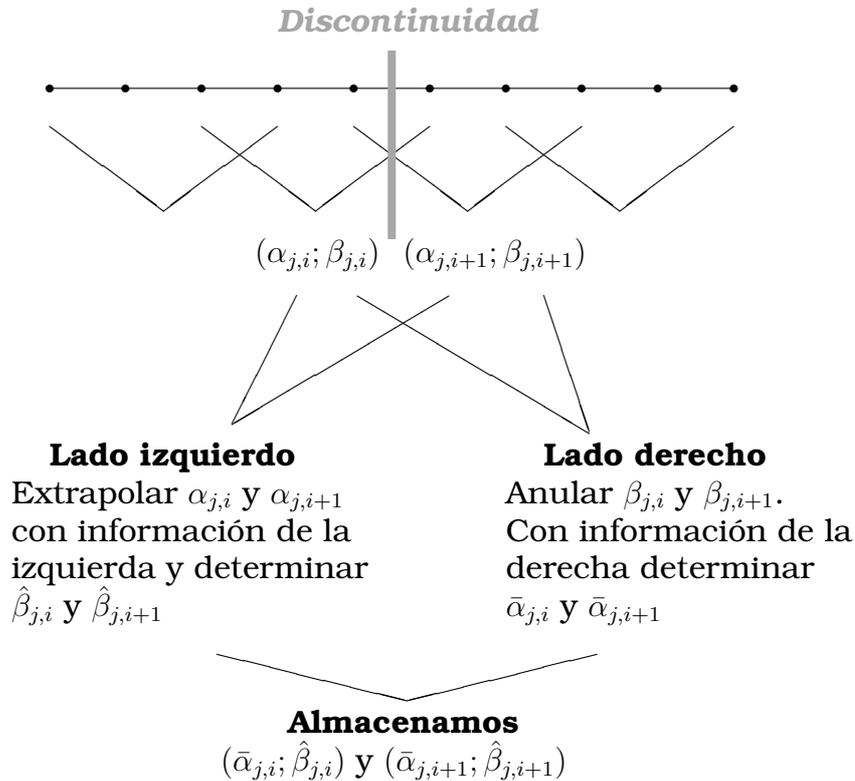


Figura 4.1: Extrapolación en el nivel grosero para DB4.

TEOREMA 4.1. Supongamos que la función de escala $\phi(x)$ y su correspondiente wavelet $\psi(x)$ tienen soporte finito en $[0, l]$ y que $\psi(x)$ posee p momentos nulos. Sea $f(x)$ una función suave a trozos en $[a, b]$, con p derivadas acotadas en cada región suave. Sea $f_j(x)$ la proyección ENO-wavelet en el nivel j obtenida por extrapolación, con una elección que satisface (3.5). Si la proyección $f_{j+1}(x)$ cumple la DSP, entonces:

$$\|f(x) - f_j(x)\| \leq C(\Delta x)^p \|f^p(x)\|_{(a,b) \setminus D},$$

donde $\Delta x = 2^{-j}$ y D es el conjunto de discontinuidades de la función o de sus derivadas mayores de orden p . $\|\cdot\|$ puede ser tanto la norma en L^2 como en L^∞ .

El teorema nos asegura que la aproximación es similar a la que obtenemos con la transformada *wavelet* estándar, excepto en aquellos puntos donde haya discontinuidades. En los experimentos numéricos comprobaremos que el teorema se cumple.

4.4

ENO-DB4

Esta transformada la describimos en la sección 1.4.2. Para bajar un nivel utilizamos las ecuaciones (1.82), (1.82) y para subir (1.84), (1.85). Suponiendo las discontinuidades detectadas, es decir, si el *stencil* $\{\alpha_{j+1,2i}, \dots, \alpha_{j+1,2i+3}\}$ cruza una discontinuidad, guardamos esta posición como $s_{j,i} = 1$ y en caso contrario $s_{j,i} = 0$. Tenemos dos opciones posibles: cuando la discontinuidad se halla entre $\alpha_{j+1,2i+1}$ y $\alpha_{j+1,2i+2}$ hay un único *stencil* afectado, mientras que si se encuentra entre $\alpha_{j+1,2i+2}$ y $\alpha_{j+1,2i+3}$ hay dos *stencils* afectados, como se indica en la Figura 4.2. Veamos cómo modificamos la transformada *DB4* en presencia de discontinuidades:

Caso 1: un *stencil* afectado

Para hallar $\beta_{j,i}$ utilizando información únicamente a la izquierda de la discontinuidad seguimos el siguiente esquema:

- Extrapolamos $\alpha_{j,i}$ y lo denotamos $\hat{\alpha}_{j,i}$. Para mantener el orden del método utilizamos extrapolación de grado uno, es decir $\hat{\alpha}_{j,i} = 2\alpha_{j,i-1} - \alpha_{j,i-2}$.
- Despejamos de (1.82) el bloque

$$c_2\hat{\alpha}_{j+1,2i+2} + c_3\hat{\alpha}_{j+1,2i+3} = \hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1}.$$

- Utilizando que $\frac{h_2}{c_2} = \frac{h_3}{c_3}$, obtenemos

$$\hat{\beta}_{j,i} = h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + \frac{h_2}{c_2} (\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1}).$$

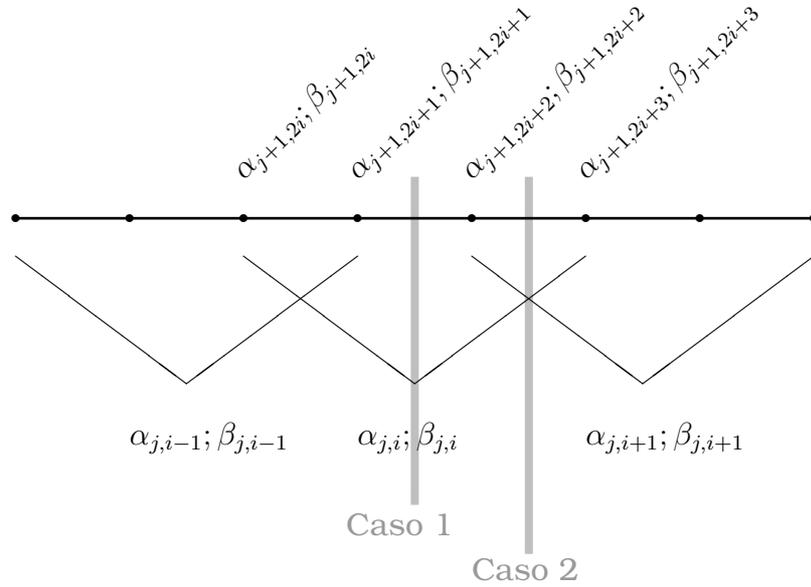


Figura 4.2: Transformación directa DB4, donde indicamos los dos posibles casos en que podemos localizar una discontinuidad, dando lugar a uno o dos stencils afectados.

Con esto hallamos $\hat{\beta}_{j,i}$ utilizando únicamente información a la izquierda de la discontinuidad. Para calcular $\bar{\alpha}_{j,i}$ con información de la derecha:

- Igualamos $\beta_{j,i} = 0$. Luego de (1.83),

$$h_0 \bar{\alpha}_{j+1,2i} + h_1 \bar{\alpha}_{j+1,2i+1} = -h_2 \alpha_{j+1,2i+2} - h_3 \alpha_{j+1,2i+3}.$$

- Utilizando de nuevo la relación $\frac{h_0}{c_0} = \frac{h_1}{c_1}$ podemos hallar $\bar{\alpha}_{j,i}$ con información a la derecha de la discontinuidad:

$$\bar{\alpha}_{j,i} = -\frac{c_0}{h_0} (h_2 \alpha_{j+1,2i+2} + h_3 \alpha_{j+1,2i+3}) + c_2 \alpha_{j+1,2i+2} + c_3 \alpha_{j+1,2i+3}.$$

Hemos obtenido $(\bar{\alpha}_{j,i}; \hat{\beta}_{j,i})$ donde cada valor ha sido calculado utilizando información de un sólo lado de la discontinuidad. Por tanto el valor de $\hat{\beta}_{j,i}$ será pequeño y no habrá una gran pérdida de información si lo eliminamos. Los valores que almacenamos son $(\bar{\alpha}_{j,i}; \hat{\beta}_{j,i})$, en lugar de $(\alpha_{j,i}; \beta_{j,i})$.

Caso 2: dos *stencils* afectados

Debemos calcular $(\alpha_{j,i}; \beta_{j,i})$ y $(\alpha_{j,i+1}; \beta_{j,i+1})$. La discontinuidad se encuentra entre $\alpha_{j+1,2i+2}$ y $\alpha_{j+1,2i+3}$, con lo que el cálculo de $\beta_{j,i}$ es sencillo:

- Extrapolamos $\alpha_{j,i}$, obteniendo $\hat{\alpha}_{j,i}$. Despejando de (1.82):

$$\hat{\alpha}_{j+1,2i+3} = \frac{1}{c_3} (\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1} - c_2\alpha_{j+1,2i+2}).$$

- Para hallar $\bar{\beta}_{j,i}$ basta con sustituir:

$$\bar{\beta}_{j,i} = h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + h_2\alpha_{j+1,2i+2} + h_3\hat{\alpha}_{j+1,2i+3}.$$

Pasemos al cálculo de $\beta_{j,i+1}$. Sólo conocemos $\alpha_{j+1,2i+2}$, pero también podemos utilizar $\hat{\alpha}_{j+1,2i+3}$, que ha sido calculado con información de la derecha. Los pasos a realizar son:

- Extrapolamos $\alpha_{j,i+1}$, obteniendo $\hat{\alpha}_{j,i+1}$. Calculamos el bloque

$$c_2\hat{\alpha}_{j+1,2i+4} + c_3\hat{\alpha}_{j+1,2i+5} = \hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i+2} - c_1\hat{\alpha}_{j+1,2i+3}.$$

- Por tanto:

$$\hat{\beta}_{j,i+1} = h_0\alpha_{j+1,2i+2} + h_1\hat{\alpha}_{j+1,2i+3} + \frac{h_3}{c_3} (\hat{\alpha}_{j,i+1} - c_0\alpha_{j+1,2i+2} - c_1\hat{\alpha}_{j+1,2i+3}).$$

En el lado derecho de la discontinuidad queremos obtener $\alpha_{j,i}$ y $\alpha_{j,i+1}$. Los pasos a seguir son análogos al lado izquierdo, pero ahora en vez de extrapolar igualamos los β 's a cero. Empecemos con $\alpha_{j,i+1}$:

- Igualamos $\beta_{j,i+1} = 0$. Calculamos $\alpha_{j+1,2i+2}$ con información de la derecha:

$$\bar{\alpha}_{j+1,2i+2} = -\frac{1}{h_0} (h_1\alpha_{j+1,2i+3} + h_2\alpha_{j+1,2i+4} + h_3\alpha_{j+1,2i+5}).$$

- Con lo que podemos hallar $\bar{\alpha}_{j,i+1}$:

$$\bar{\alpha}_{j,i+1} = c_0\bar{\alpha}_{j+1,2i+2} + c_1\alpha_{j+1,2i+3} + c_2\alpha_{j+1,2i+4} + c_3\alpha_{j+1,2i+5}.$$

Para calcular $\alpha_{j,i}$ utilizamos el valor de $\bar{\alpha}_{j+1,2i+2}$:

- Igualamos $\beta_{j,i} = 0$. Obtenemos el bloque

$$h_0\bar{\alpha}_{j+1,2i} + h_1\bar{\alpha}_{j+1,2i+1} = -h_2\bar{\alpha}_{j+1,2i+2} - h_3\alpha_{j+1,2i+3}.$$

- Con lo cual:

$$\bar{\alpha}_{j,i} = \frac{c_0}{h_0} (h_2\bar{\alpha}_{j+1,2i+2} + h_3\alpha_{j+1,2i+3}) + c_2\bar{\alpha}_{j+1,2i+2} + c_3\alpha_{j+1,2i+3}.$$

4.4.1

Reconstrucción

Para recuperar los datos iniciales cuando trabajamos con *stencils* que cruzan discontinuidades debemos extrapolar las α 's para recuperar los datos de la izquierda e igualar los β 's a cero para recuperar la información de la derecha. Detalladamente:

1. $s_{j,i} = 0$ y $s_{j,i-1} = 0$. En este caso no intervienen discontinuidades y, por tanto, utilizamos la reconstrucción habitual (1.84) y (1.85).

2. $s_{j,i} = 1$ y $s_{j,i-1} = 0$. Queremos hallar $\alpha_{j+1,2i}$ y $\alpha_{j+1,2i+1}$, que están ambos a la izquierda de la discontinuidad. Debemos extrapolar $\alpha_{j,i}$, ya que este valor está calculado a partir de información a la derecha del salto. El $\beta_{j,i}$ es correcto, pues ha sido calculado con información de la izquierda. El resto, $(\alpha_{j,i-1}; \beta_{j,i-1})$ también es correcto porque han sido calculados con *stencils* a la izquierda que no cruzan la discontinuidad. Luego:

$$\begin{aligned} \alpha_{j+1,2i} &= r_1\hat{\alpha}_{j,i} + d_1\beta_{j,i} + r_3\alpha_{j,i-1} + d_3\beta_{j,i-1}, \\ \alpha_{j+1,2i+1} &= r_0\hat{\alpha}_{j,i} + d_0\beta_{j,i} + r_2\alpha_{j,i-1} + d_2\alpha_{j,i-1}. \end{aligned}$$

3. $s_{j,i} = 0$ y $s_{j,i-1} = 1$. Los datos se hallan a la derecha de la discontinuidad. $(\alpha_{j,i}; \beta_{j,i})$ son correctos, ya que provienen de *stencils* a la derecha del salto y que no lo cruzan. El valor de $\alpha_{j,i-1}$ es válido ya que proviene de información a la derecha. Pero $\beta_{j,i-1}$ proviene de datos de la izquierda y por tanto debemos igualarlo a cero:

$$\alpha_{j+1,2i} = r_1\alpha_{j,i} + d_1\beta_{j,i} + r_3\alpha_{j,i-1},$$

$$\alpha_{j+1,2i+1} = r_0\alpha_{j,i} + d_0\beta_{j,i} + r_2\alpha_{j,i-1}.$$

4. $s_{j,i} = 1$ y $s_{j,i-1} = 1$. $\alpha_{j+1,2i}$ estará a la izquierda de la discontinuidad y $\alpha_{j+1,2i+1}$ a la derecha. Empecemos por $\alpha_{j+1,2i}$. En este caso $\beta_{j,i-1}$ y $\beta_{j,i}$ son correctos, ya que provienen de datos a la izquierda. Pero $\alpha_{j,i-1}$ y $\alpha_{j,i}$ son ambos incorrectos y debemos extrapolarlos:

$$\alpha_{j+1,2i} = r_1\hat{\alpha}_{j,i} + d_1\beta_{j,i} + r_3\hat{\alpha}_{j,i-1} + d_3\beta_{j,i-1}.$$

Nos queda calcular $\alpha_{j+1,2i+1}$. Ahora $\alpha_{j,i-1}$ y $\alpha_{j,i}$ son válidos, pero $\beta_{j,i-1}$ y $\beta_{j,i}$ deben ser igualados a cero. Por tanto:

$$\alpha_{j+1,2i+1} = r_0\alpha_{j,i} + r_2\alpha_{j,i-1}.$$

4.5

ENO-DB6

En este caso la transformada directa es:

$$\alpha_{j,i} = \sum_{k=0}^5 c_k \alpha_{j+1,2i+k}, \quad (4.2)$$

$$\beta_{j,i} = \sum_{k=0}^5 h_k \alpha_{j+1,2i+k}, \quad (4.3)$$

donde los coeficientes² son:

$$\begin{aligned} c_0 &= 0,03522629188210, & h_0 &= -0,33267055295096, \\ c_1 &= -0,08544127388224, & h_1 &= 0,80689150931334, \\ c_2 &= -0,13501102001039, & h_2 &= -0,45987750211933, \\ c_3 &= 0,45987750211933, & h_3 &= -0,13501102001039, \\ c_4 &= 0,80689150931334, & h_4 &= 0,08544127388224, \\ c_5 &= 0,33267055295096, & h_5 &= 0,03522629188210. \end{aligned}$$

La inversa es como sigue:

$$\alpha_{j+1,2i} = r_1\alpha_{j,i} + d_1\beta_{j,i} + r_3\alpha_{j,i-1} + d_3\beta_{j,i-1} + r_5\alpha_{j,i-2} + d_5\beta_{j,i-2} \quad (4.4)$$

²Estos coeficientes pueden obtenerse con MATLAB® mediante *wfilters.m*.

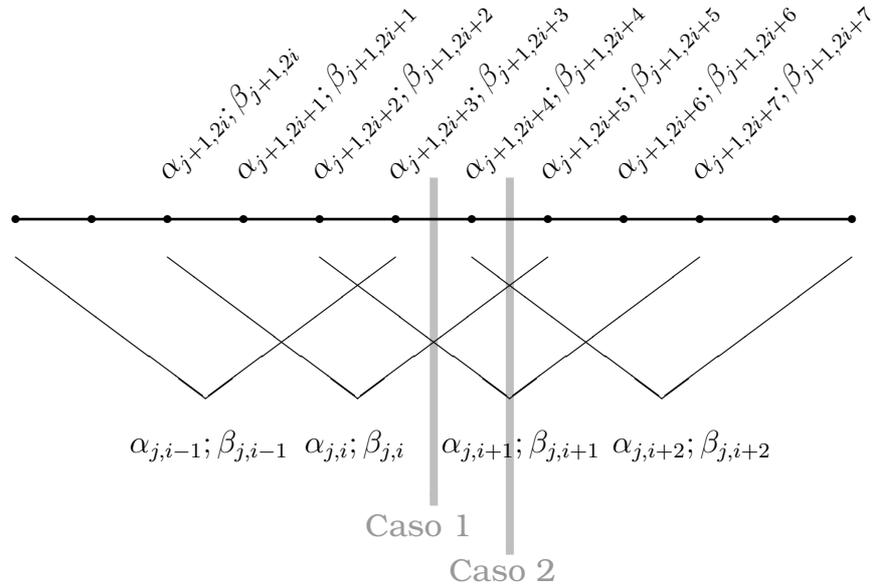


Figura 4.3: Transformación directa DB6, donde indicamos los dos posibles casos en que podemos localizar una discontinuidad, dando lugar a dos o tres stencils afectados

$$\alpha_{j+1,2i+1} = r_0\alpha_{j,i} + d_0\beta_{j,i} + r_2\alpha_{j,i-1} + d_2\beta_{j,i-1} + r_4\alpha_{j,i-2} + d_4\beta_{j,i-2} \quad (4.5)$$

donde: $r_{2i+1} = c_{2i}$, $d_{2i+1} = h_{2i}$, $r_{2i} = c_{2i+1}$, $d_{2i} = h_{2i+1}$, $i = 0, 1, 2$.

Al igual que en DB4 tenemos dos casos. En el primero la discontinuidad se localiza entre $\alpha_{j+1,2i+3}$ y $\alpha_{j+1,2i+4}$, dando lugar a dos stencils afectados. En el segundo caso la discontinuidad se halla entre $\alpha_{j+1,2i+4}$ y $\alpha_{j+1,2i+5}$ y afecta a tres stencils, ver la Figura 4.3.

Caso 1: dos stencils afectados

Lado izquierdo

1. $\beta_{j,i}$. Extrapolamos $\alpha_{j,i}$ y lo denotamos $\hat{\alpha}_{j,i}$. Para mantener el orden del método utilizamos extrapolación de orden 2. Despejamos de (4.2) el bloque

$$c_4\hat{\alpha}_{j+1,2i+4} + c_5\hat{\alpha}_{j+1,2i+5} = \hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1}$$

$$-c_2\alpha_{j+1,2i+2} - c_3\alpha_{j+1,2i+3}.$$

Utilizando que $\frac{h_4}{c_4} = \frac{h_5}{c_5}$, obtenemos

$$\begin{aligned}\hat{\beta}_{j,i} &= h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + h_2\alpha_{j+1,2i+2} + h_3\alpha_{j+1,2i+3} \\ &+ \frac{h_4}{c_4} (\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1} - c_2\alpha_{j+1,2i+2} - c_3\alpha_{j+1,2i+3}).\end{aligned}$$

2. $\beta_{j,i+1}$. Extrapolamos $\alpha_{j,i+1} \rightarrow \hat{\alpha}_{j,i+1}$. Con lo cual tenemos:

$$\begin{aligned}\hat{\alpha}_{j,i+1} &= c_0\alpha_{j+1,2i+2} + c_1\alpha_{j+1,2i+3} + c_2\alpha_{j+1,2i+4} \\ &+ c_3\alpha_{j+1,2i+5} + c_4\alpha_{j+1,2i+6} + c_5\alpha_{j+1,2i+7}. \\ \hat{\beta}_{j,i+1} &= h_0\alpha_{j+1,2i+2} + h_1\alpha_{j+1,2i+3} + h_2\alpha_{j+1,2i+4} \\ &+ h_3\alpha_{j+1,2i+5} + h_4\alpha_{j+1,2i+6} + h_5\alpha_{j+1,2i+7}.\end{aligned}$$

De estas ecuaciones nos falta por conocer $\alpha_{j+1,i+m}$, $m = 4, 5, 6, 7$ y $\hat{\beta}_{j,i+1}$. Con lo cual tenemos un problema, ya que es evidente que con la información disponible no podemos determinar $\hat{\beta}_{j,i+1}$. No podemos utilizar el bloque $c_4\hat{\alpha}_{j+1,2i+4} + c_5\hat{\alpha}_{j+1,2i+5}$ que hemos calculado en el paso previo ya que la relación $\frac{c_2}{c_4} = \frac{c_3}{c_5}$ no se cumple.

Chan y Zhou sugieren en [26] que se extrapole $\hat{\alpha}_{j+1,2i+4}$ y se determine entonces $\hat{\alpha}_{j+1,2i+5}$. Nosotros proponemos una nueva forma de proceder, utilizando la información que proporciona la transformada inversa. Esta idea evita hacer extrapolaciones en el nivel más fino, simplifica los cálculos y permite una generalización más sencilla para $DB2p$, con p un entero arbitrario. En cualquier caso expondremos este método sin aplicar esta idea, que será desarrollada en un apartado posterior.

Así pues, extrapolamos $\hat{\alpha}_{j+1,2i+4}$, con un método de grado 2 y calculamos $\hat{\alpha}_{j+1,2i+5}$ de la siguiente manera:

$$\begin{aligned}\hat{\alpha}_{j+1,2i+5} &= \frac{1}{c_5} (\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1} \\ &- c_2\alpha_{j+1,2i+2} - c_3\alpha_{j+1,2i+3} - c_4\hat{\alpha}_{j+1,2i+4}).\end{aligned}$$

Con esto despejamos el bloque:

$$\begin{aligned}c_4\alpha_{j+1,2i+6} + c_5\alpha_{j+1,2i+7} &= \hat{\alpha}_{j,i+1} - c_0\alpha_{j+1,2i+2} \\ &- c_1\alpha_{j+1,2i+3} - c_2\hat{\alpha}_{j+1,2i+4} - c_3\hat{\alpha}_{j+1,2i+5}.\end{aligned}$$

Haciendo uso de la relación $\frac{h_4}{c_4} = \frac{h_5}{c_5}$:

$$\begin{aligned}\hat{\beta}_{j,i+1} &= h_0\alpha_{j+1,2i+2} + h_1\alpha_{j+1,2i+3} + h_2\hat{\alpha}_{j+1,2i+4} + h_3\hat{\alpha}_{j+1,2i+5} \\ &\quad + \frac{h_4}{c_4} (\hat{\alpha}_{j,i+1} - c_0\alpha_{j+1,2i+2} - c_1\alpha_{j+1,2i+3} - c_2\hat{\alpha}_{j+1,2i+4} - c_3\hat{\alpha}_{j+1,2i+5}).\end{aligned}$$

Lado derecho

1. $\alpha_{j,i+1}$. Imponemos que $\bar{\beta}_{j,i+1} = 0$. Dado que $\frac{c_0}{h_0} = \frac{c_1}{h_1}$:

$$\begin{aligned}\bar{\alpha}_{j,i+1} &= -\frac{c_0}{h_0} (h_2\alpha_{j+1,2i+4} + h_3\alpha_{j+1,2i+5} + h_4\alpha_{j+1,2i+6} + h_5\alpha_{j+1,2i+7}) \\ &\quad + c_2\alpha_{j+1,2i+4} + c_3\alpha_{j+1,2i+5} + c_4\alpha_{j+1,2i+6} + c_5\alpha_{j+1,2i+7}.\end{aligned}$$

2. $\alpha_{j,i}$. Imponemos que $\bar{\beta}_{j,i} = 0$. Partiendo de (4.2) y (4.3) nos encontramos en la misma situación que en el cálculo de $\beta_{j,i+1}$, ya que sólo conocemos $\alpha_{j+1,2i+4}$ y $\alpha_{j+1,2i+5}$. Realizamos la extrapolación con un método de grado 2 en el nivel $j+1$, obteniendo $\bar{\alpha}_{j+1,2i+3}$ con información de la derecha. Así:

$$\begin{aligned}\bar{\alpha}_{j+1,2i+2} &= \frac{-1}{h_0} (h_1\bar{\alpha}_{j+1,2i+3} + h_2\alpha_{j+1,2i+4} + h_3\alpha_{j+1,2i+5} \\ &\quad + h_4\alpha_{j+1,2i+6} + h_5\alpha_{j+1,2i+7}).\end{aligned}$$

Mediante la relación $\frac{c_0}{h_0} = \frac{c_1}{h_1}$:

$$\begin{aligned}\bar{\alpha}_{j,i} &= -\frac{c_0}{h_0} (h_2\bar{\alpha}_{j+1,2i+2} + h_3\bar{\alpha}_{j+1,2i+3} + h_4\alpha_{j+1,2i+4} + h_5\alpha_{j+1,2i+5}) \\ &\quad + c_2\bar{\alpha}_{j+1,2i+2} + c_3\bar{\alpha}_{j+1,2i+3} + c_4\alpha_{j+1,2i+4} + c_5\alpha_{j+1,2i+5}.\end{aligned}$$

Caso 2: tres *stencils* afectados

Lado Izquierdo.

1. $\beta_{j,i}$. Extrapolamos $\alpha_{j,i}$, denotándolo $\hat{\alpha}_{j,i}$. De (4.2) y (4.3) conocemos $\alpha_{j+1,2i+m}$, $m = 0, 1, 2, 3, 4$. Despejando:

$$\begin{aligned}\hat{\alpha}_{j+1,2i+5} &= \frac{1}{c_5} (\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1} - c_2\alpha_{j+1,2i+2} \\ &\quad - c_3\alpha_{j+1,2i+3} - c_4\alpha_{j+1,2i+4}).\end{aligned}\tag{4.6}$$

Con esto:

$$\begin{aligned} \beta_{j,i} = & h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + h_2\alpha_{j+1,2i+2} \\ & + h_3\alpha_{j+1,2i+3} + h_4\alpha_{j+1,2i+4} + h_5\hat{\alpha}_{j+1,2i+5}. \end{aligned} \quad (4.7)$$

2. $\beta_{j,i+1}$. Extrapolamos $\alpha_{j,i+1}$. Conocemos $\alpha_{j+1,i+2}$, $\alpha_{j+1,i+3}$ y $\alpha_{j+1,i+4}$ ya que se encuentran a la izquierda de la discontinuidad. Además podemos utilizar $\hat{\alpha}_{j+1,2i+5}$ que ha sido calculado en el paso anterior con información de la izquierda. Con lo cual sabiendo que $\frac{h_4}{c_4} = \frac{h_5}{c_5}$ podemos determinar $\beta_{j,i+1}$:

$$\begin{aligned} \beta_{j,i+1} = & h_0\alpha_{j+1,2i+2} + h_1\alpha_{j+1,2i+3} + h_2\alpha_{j+1,2i+4} + h_3\hat{\alpha}_{j+1,2i+5} \\ & + \frac{h_4}{c_4} (\hat{\alpha}_{j,i+1} - c_0\alpha_{j+1,2i+2} - c_1\alpha_{j+1,2i+3} - c_2\alpha_{j+1,2i+4} - c_3\hat{\alpha}_{j+1,2i+5}). \end{aligned}$$

3. $\beta_{j,i+2}$. En este caso no podemos determinar el coeficiente de la manera habitual debiendo utilizar extrapolaciones en el nivel $j+1$. Así, $\alpha_{j+1,2i+6}$ lo extrapolamos con información de la izquierda:

$$\begin{aligned} \hat{\alpha}_{j+1,2i+7} = & \frac{1}{c_5} (\hat{\alpha}_{j,i+1} - c_0\alpha_{j+1,2i+2} - c_1\alpha_{j+1,2i+3} \\ & - c_2\alpha_{j+1,2i+4} - c_3\hat{\alpha}_{j+1,2i+5} - c_4\hat{\alpha}_{j+1,2i+6}). \end{aligned}$$

Con estos datos, extrapolando $\alpha_{j,i+2}$ y haciendo uso de la relación $\frac{h_4}{c_4} = \frac{h_5}{c_5}$:

$$\begin{aligned} \hat{\beta}_{j,i+2} = & h_0\alpha_{j+1,2i+4} + h_1\hat{\alpha}_{j+1,2i+5} + h_2\hat{\alpha}_{j+1,2i+6} + h_3\hat{\alpha}_{j+1,2i+7} \\ & + \frac{h_4}{c_4} (\hat{\alpha}_{j,i+2} - c_0\alpha_{j+1,2i+4} - c_1\hat{\alpha}_{j+1,2i+5} - c_2\hat{\alpha}_{j+1,2i+6} - c_3\hat{\alpha}_{j+1,2i+7}). \end{aligned}$$

Lado Derecho.

1. $\alpha_{j,i+2}$. Primero hallamos $\alpha_{j+1,2i+4}$,

$$\begin{aligned} \bar{\alpha}_{j+1,2i+4} = & -\frac{1}{h_0} (h_1\alpha_{j+1,2i+5} + h_2\alpha_{j+1,2i+6} + h_3\alpha_{2i+7} \\ & + h_4\alpha_{j+1,2i+8} + h_5\alpha_{j+1,2i+9}), \end{aligned}$$

donde $\beta_{j,i+2} = 0$. Con lo cual podemos determinar $\alpha_{j,i+2}$,

$$\bar{\alpha}_{j,i+2} = c_0\bar{\alpha}_{j+1,2i+4} + c_1\alpha_{j+1,2i+5} + c_2\alpha_{j+1,2i+6} + c_3\alpha_{2i+7}$$

$$+c_4\alpha_{j+1,2i+8} + c_5\alpha_{j+1,2i+9}.$$

2. $\alpha_{j,i+1}$. Igualamos $\beta_{j,i+1}$ a 0. Utilizando de nuevo que $\frac{c_0}{h_0} = \frac{c_1}{h_1}$:

$$\begin{aligned}\bar{\alpha}_{j,i+1} = & -\frac{c_0}{h_0} (h_2\bar{\alpha}_{j+1,2i+4} + h_3\alpha_{j+1,2i+5} + h_4\alpha_{j+1,2i+6} + h_5\alpha_{j+1,2i+7}) \\ & +c_2\bar{\alpha}_{j+1,2i+4} + c_3\alpha_{j+1,2i+5} + c_4\alpha_{j+1,2i+6} + c_5\alpha_{j+1,2i+7}.\end{aligned}$$

3. $\alpha_{j,i}$. Igualamos $\beta_{j,i} = 0$. Extrapolamos $\alpha_{j+1,2i+3}$ con información de la derecha, obteniendo $\bar{\alpha}_{j+1,2i+3}$. Con esto:

$$\begin{aligned}\bar{\alpha}_{j+1,2i+2} = & \frac{-1}{h_0} (h_1\bar{\alpha}_{j+1,2i+3} + h_2\bar{\alpha}_{j+1,2i+4} \\ & +h_3\alpha_{j+1,2i+5} + h_4\alpha_{j+1,2i+6} + h_5\alpha_{j+1,2i+7}).\end{aligned}$$

Mediante la relación $\frac{c_0}{h_0} = \frac{c_1}{h_1}$ y los datos anteriores obtenemos:

$$\begin{aligned}\bar{\alpha}_{j,i} = & -\frac{c_0}{h_0} (h_2\bar{\alpha}_{j+1,2i+2} + h_3\bar{\alpha}_{j+1,2i+3} + h_4\bar{\alpha}_{j+1,2i+4} + h_5\alpha_{j+1,2i+5}) \\ & +c_2\bar{\alpha}_{j+1,2i+2} + c_3\bar{\alpha}_{j+1,2i+3} + c_4\bar{\alpha}_{j+1,2i+4} + c_5\alpha_{j+1,2i+5}.\end{aligned}$$

4.5.1

Reconstrucción

Para recuperar los datos iniciales, utilizamos (4.4) y (4.5). Al igual que con ENO-DB4 cuando trabajamos con *stencils* que cruzan discontinuidades, debemos extrapolar las α 's e igualar los β 's a cero según si nos hallamos a izquierda o derecha de la discontinuidad respectivamente. Recordemos que hemos guardado un vector s tal que $s_{j,i} = 0$ si el *stencil* $\{\alpha_{j+1,2i}, \alpha_{j+1,2i+1}, \alpha_{j+1,2i+2}, \alpha_{j+1,2i+3}\}$ no cruza una discontinuidad, y $s_{j,i} = 1$ si la cruza. Nos podemos encontrar con los siguientes casos:

1. $s_{j,i} = 0$ y $s_{j,i-1} = 0$ y $s_{j,i-2} = 0$. No hay discontinuidades y por tanto utilizamos las transformadas (4.4) y (4.5).

2. $s_{j,i} = 0$ y $s_{j,i-1} = 0$ y $s_{j,i-2} = 1$. Como $\alpha_{j+1,2i}$ y $\alpha_{j+1,2i+1}$ se encuentran a la derecha de la discontinuidad, $\alpha_{j,i-2}$ es correcto y $\beta_{j,i-2}$ debe ser igualado a cero. Por tanto:

$$\alpha_{j+1,2i} = r_1\alpha_{j,i} + d_1\beta_{j,i} + r_3\alpha_{j,i-1} + d_3\beta_{j,i-1} + r_5\alpha_{j,i-2},$$

$$\alpha_{j+1,2i+1} = r_0\alpha_{j,i} + d_0\beta_{j,i} + r_2\alpha_{j,i-1} + d_2\beta_{j,i-1} + r_4\alpha_{j,i-2}.$$

3. $s_{j,i} = 0$ y $s_{j,i-1} = 1$ y $s_{j,i-2} = 1$. Los datos a recuperar se hallan a la derecha de la discontinuidad. Luego los α 's son correctos mientras que $\beta_{j,i-m}$, $m = 1, 2$, deben ser igualados a cero. Entonces:

$$\begin{aligned}\alpha_{j+1,2i} &= r_1\alpha_{j,i} + d_1\beta_{j,i} + r_3\alpha_{j,i-1} + r_5\alpha_{j,i-2}, \\ \alpha_{j+1,2i+1} &= r_0\alpha_{j,i} + d_0\beta_{j,i} + r_2\alpha_{j,i-1} + r_4\alpha_{j,i-2}.\end{aligned}$$

4. $s_{j,i} = 1$ y $s_{j,i-1} = 1$ y $s_{j,i-2} = 1$. $\alpha_{j+1,2i}$ se halla a la izquierda de la discontinuidad. Los β 's son correctos mientras que los α 's deben ser extrapolados. Por tanto:

$$\alpha_{j+1,2i} = r_1\hat{\alpha}_{j,i} + d_1\beta_{j,i} + r_3\hat{\alpha}_{j,i-1} + d_3\beta_{j,i-1} + r_5\hat{\alpha}_{j,i-2} + d_5\beta_{j,i-2}.$$

$\alpha_{j+1,2i+1}$ está a la derecha del salto. Luego los α 's son correctos y los β 's deben ser igualados a cero. Con lo cual,

$$\alpha_{j+1,2i+1} = r_0\alpha_{j,i} + r_2\alpha_{j,i-1} + r_4\alpha_{j,i-2}.$$

5. $s_{j,i} = 1$ y $s_{j,i-1} = 1$ y $s_{j,i-2} = 0$. Los datos a recuperar se encuentran a la izquierda del salto. Los β son correctos y debemos extrapolar $\alpha_{j,i}$ y $\alpha_{j,i-1}$ con información de la izquierda:

$$\begin{aligned}\alpha_{j+1,2i} &= r_1\hat{\alpha}_{j,i} + d_1\beta_{j,i} + r_3\hat{\alpha}_{j,i-1} + d_3\beta_{j,i-1} + r_5\alpha_{j,i-2} + d_5\beta_{j,i-2}, \\ \alpha_{j+1,2i+1} &= r_0\hat{\alpha}_{j,i} + d_0\beta_{j,i} + r_2\hat{\alpha}_{j,i-1} + d_2\beta_{j,i-1} + r_4\alpha_{j,i-2} + d_4\beta_{j,i-2}.\end{aligned}$$

6. $s_{j,i} = 1$ y $s_{j,i-1} = 0$ y $s_{j,i-2} = 0$. Nos encontramos a la izquierda del salto. Luego $\beta_{j,i}$ es correcto y $\alpha_{j,i}$ debe ser extrapolado:

$$\begin{aligned}\alpha_{j+1,2i} &= r_1\hat{\alpha}_{j,i} + d_1\beta_{j,i} + r_3\alpha_{j,i-1} + d_3\beta_{j,i-1} + r_5\alpha_{j,i-2} + d_5\beta_{j,i-2}, \\ \alpha_{j+1,2i+1} &= r_0\hat{\alpha}_{j,i} + d_0\beta_{j,i} + r_2\alpha_{j,i-1} + d_2\beta_{j,i-1} + r_4\alpha_{j,i-2} + d_4\beta_{j,i-2}.\end{aligned}$$

4.6

Algoritmos utilizando la transformada inversa (TI)

En esta sección presentamos una nueva manera de deducir los coeficientes del nivel más fino una vez que se ha realizado la extrapolación de los coeficientes del nivel grosero, utilizando la información de la transformada inversa. Esta idea evita realizar extrapolaciones en el nivel fino, simplifica los cálculos, garantiza la invertibilidad del método y permite una generalización sencilla para la base DB de cualquier orden.

4.6.1

ENO-DB4-TI

La transformada inversa en varios puntos puede expresarse:

$$\alpha_{j+1,2(i+l)} = r_1\alpha_{j,i+l} + d_1\beta_{j,i+l} + r_3\alpha_{j,i+l-1} + d_3\beta_{j,i+l-1}, \quad (4.8)$$

$$\alpha_{j+1,2(i+l)+1} = r_0\alpha_{j,i+l} + d_0\beta_{j,i+l} + r_2\alpha_{j,i+l-1} + d_2\beta_{j,i+l-1}. \quad (4.9)$$

Caso 1: un *stencil* afectado

Lado izquierdo

1. $\beta_{j,i}$. Extrapolamos $\hat{\alpha}_{j,i}$ y despejamos utilizando (4.8) con $l = 0$:

$$\hat{\beta}_{j,i} = \frac{1}{d_1} (\alpha_{j+1,2i} - r_1\hat{\alpha}_{j,i} - r_3\alpha_{j,i-1} - d_3\beta_{j,i-1}),$$

donde todos los datos a la izquierda del igual son conocidos y se hallan de información a la izquierda de la discontinuidad.

NOTA 4.1. En cada paso tenemos un par de ecuaciones que podemos elegir para despejar, una asociada a $\alpha_{j+1,m}$ y la otra a $\alpha_{j+1,m+1}$. Dado que la reconstrucción es invertible, como se demostrará posteriormente, podemos elegir una u otra indistintamente siempre que se reconstruyan datos de un mismo lado de la discontinuidad. Escogemos la asociada a $\alpha_{j+1,m}$ si calculamos β 's, para asegurarnos que $\alpha_{j+1,m}$ está a la izquierda de la discontinuidad. Siguiendo el mismo razonamiento tomamos la ecuación asociada a $\alpha_{j+1,m+1}$ si estamos calculando α 's. En lo que sigue, la elección se realizará de esta manera.

Lado derecho

1. $\alpha_{j,i}$. Igualamos $\bar{\beta}_{j,i} = 0$. Para poder despejar $\bar{\alpha}_{j,i}$ a partir de datos conocidos, elegimos (4.9) con $l = 1$, obteniendo:

$$\bar{\alpha}_{j,i} = \frac{1}{r_2} (\alpha_{j+1,2i+3} - r_0 \alpha_{j,i+1} - d_0 \beta_{j,i+1}).$$

Caso 2: dos stencils afectados

Lado izquierdo

1. $\beta_{j,i}$. Al estar a la izquierda de la discontinuidad, extrapolamos $\hat{\alpha}_{j,i}$. Hacemos uso de (4.8) con $l = 0$ y despejamos:

$$\hat{\beta}_{j,i} = \frac{1}{d_1} (\alpha_{j+1,2i} - r_1 \hat{\alpha}_{j,i} - r_3 \alpha_{j,i-1} - d_3 \beta_{j,i-1}),$$

2. $\beta_{j,i+1}$. Extrapolamos $\alpha_{j,i+1}$ y utilizando (4.8) con $l = 1$ podemos hallar $\beta_{j,i+1}$ con información a la izquierda del salto:

$$\hat{\beta}_{j,i+1} = \frac{1}{d_1} (\alpha_{j+1,2i+2} - r_1 \hat{\alpha}_{j,i+1} - r_3 \hat{\alpha}_{j,i} - d_3 \hat{\beta}_{j,i}).$$

Lado derecho

1. $\alpha_{j,i+1}$. Igualamos $\bar{\beta}_{j,i+1} = 0$. Mediante (4.9) con $l = 2$:

$$\bar{\alpha}_{j,i+1} = \frac{1}{r_2} (\alpha_{j+1,2i+5} - r_0 \alpha_{j,i+2} - d_0 \beta_{j,i+2}).$$

2. $\alpha_{j,i}$. Igualamos $\bar{\beta}_{j,i} = 0$. Utilizando (4.9) con $l = 1$ nos queda:

$$\bar{\alpha}_{j,i} = \frac{1}{r_2} (\alpha_{j+1,2i+3} - r_0 \bar{\alpha}_{j,i+1}).$$

4.6.2

ENO-DB6-TI

La transformada inversa para DB6 podemos expresarla como:

$$\alpha_{j+1,2i+2l} = \sum_{k=0}^2 r_{2k+1} \alpha_{j,i-k+l} + d_{2k+1} \beta_{j,i-k+l}, \quad (4.10)$$

$$\alpha_{j+1,2i+2l+1} = \sum_{k=0}^2 r_{2k} \alpha_{j,i-k+l} + d_{2k} \beta_{j,i-k+l}. \quad (4.11)$$

Caso 1: dos *stencils* afectados

Lado izquierdo

1. $\beta_{j,i}$. Extrapolamos $\hat{\alpha}_{j,i}$ y utilizando (4.10) para $l = 0$:

$$\hat{\beta}_{j,i} = \frac{1}{d_1} (\alpha_{j+1,2i} - r_1 \hat{\alpha}_{j,i} - r_3 \alpha_{j,i-1} - d_3 \beta_{j,i-1} - r_5 \alpha_{j,i-2} - d_5 \beta_{j,i-2}).$$

2. $\beta_{j,i+1}$. Extrapolamos $\hat{\alpha}_{j,i+1}$ y utilizando (4.10) para $l = 1$:

$$\hat{\beta}_{j,i+1} = \frac{1}{d_1} (\alpha_{j+1,2i+2} - r_1 \hat{\alpha}_{j,i+1} - r_3 \hat{\alpha}_{j,i} - d_3 \hat{\beta}_{j,i} - r_5 \alpha_{j,i-1} - d_5 \beta_{j,i-1}).$$

Lado derecho

1. $\alpha_{j,i+1}$. Igualamos $\bar{\beta}_{j,i+1}$ a cero. Utilizando (4.11) para $l = 3$:

$$\bar{\alpha}_{j,i+1} = \frac{1}{r_4} (\alpha_{j+1,2i+7} - r_0 \alpha_{j,i+3} - d_0 \beta_{j,i+3} - r_2 \alpha_{j,i+2} - d_2 \beta_{j,i+2}).$$

2. $\alpha_{j,i}$. Igualamos $\bar{\beta}_{j,i}$ a cero. Utilizando (4.11) para $l = 2$:

$$\bar{\alpha}_{j,i} = \frac{1}{r_4} (\alpha_{j+1,2i+5} - r_0 \alpha_{j,i+2} - d_0 \beta_{j,i+2} - r_2 \bar{\alpha}_{j,i+1}).$$

Caso 2: tres *stencils* afectados**Lado izquierdo**

1. $\hat{\beta}_{j,i}$ y $\hat{\beta}_{j,i+1}$. Se calculan igual que en el caso de dos *stencils* afectados.

2. $\beta_{j,i+2}$. Extrapolamos $\hat{\alpha}_{j,i+2}$. Utilizando (4.10) con $l = 2$:

$$\hat{\beta}_{j,i+2} = \frac{1}{d_1} \left(\alpha_{j+1,2i+4} - r_1 \hat{\alpha}_{j,i+2} - r_3 \hat{\alpha}_{j,i+1} - d_3 \hat{\beta}_{j,i+1} - r_5 \hat{\alpha}_{j,i} - d_5 \hat{\beta}_{j,i} \right).$$

Lado derecho

1. $\alpha_{j,i+2}$. Igualamos $\beta_{j,i+2}$ a cero. Utilizando (4.11) con $l = 4$:

$$\bar{\alpha}_{j,i+2} = \frac{1}{r_4} \left(\alpha_{j+1,2i+9} - r_0 \alpha_{j,i+4} - d_0 \beta_{j,i+4} - r_2 \alpha_{j,i+3} - d_2 \beta_{j,i+3} \right).$$

2. $\alpha_{j,i+1}$. Igualamos $\beta_{j,i+1}$ a cero. Mediante (4.11) con $l = 3$:

$$\bar{\alpha}_{j,i+1} = \frac{1}{r_4} \left(\alpha_{j+1,2i+7} - r_0 \alpha_{j,i+3} - d_0 \beta_{j,i+3} - r_2 \bar{\alpha}_{j,i+2} \right).$$

3. $\alpha_{j,i}$. Igualamos $\beta_{j,i}$ a cero. Utilizando (4.11) con $l = 2$:

$$\bar{\alpha}_{j,i} = \frac{1}{r_4} \left(\alpha_{j+1,2i+5} - r_0 \bar{\alpha}_{j,i+2} - r_2 \bar{\alpha}_{j,i+1} \right).$$

4.6.3**Algoritmo general**

Mediante la transformada inversa hemos obtenido los algoritmos *ENO-DB4-TI* y *ENO-DB6-TI*. Observamos que la manera de proceder es similar en ambos casos. En esta sección construiremos un algoritmo general que llamaremos *ENO-DB2p-TI*, con p natural.

La transformada directa $DB2p$ en general puede expresarse:

$$\alpha_{j,i} = \sum_{k=2i}^{2i+2p-1} c_{k-2i} \alpha_{j+1,k}, \quad (4.12)$$

$$\beta_{j,i} = \sum_{k=2i}^{2i+2p-1} h_{k-2i} \alpha_{j+1,k}, \quad (4.13)$$

y la transformada inversa como:

$$\alpha_{j+1,2i-1} = \sum_{k=0}^{p-1} (r_{2k} \alpha_{j,i-k} + d_{2k} \beta_{j,i-k}), \quad (4.14)$$

$$\alpha_{j+1,2i} = \sum_{k=0}^{p-1} (r_{2k+1} \alpha_{j,i-k} + d_{2k+1} \beta_{j,i-k}), \quad (4.15)$$

donde $r_{2k} = c_{2k+1}$, $d_{2k} = h_{2k+1}$, $r_{2k+1} = c_{2k}$, $d_{2k+1} = h_{2k}$, con $k = 0, 1, 2, \dots$

Caso de $p - 1$ stencils afectados

Lado izquierdo

Extrapolamos $\alpha_{j,k} \leftrightarrow \hat{\alpha}_{j,k}$, para $k = i, i + 1, \dots, i + p - 2$. La extrapolación debe de ser de grado $p - 1$. El sobrescribir los coeficientes no es ningún problema, ya que para el lado derecho no se utilizan los α 's originales. Los β 's a modificar son:

$$\beta_{j,i+m} = \frac{1}{d_1} (\alpha_{j+1,2(i+m)} - r_1 \alpha_{j,i+m} - \sum_{k=1}^{p-1} [r_{2k+1} \alpha_{j,i-k+m} + d_{2k+1} \beta_{j,i-k+m}]) \quad (4.16)$$

con $m = 0, 1, \dots, p - 2$.

Guardamos la posición de los coeficientes calculados a partir de *stencils* que cruzan discontinuidades en un vector auxiliar: $s_{j,m} = 1$, para $m = i, i + 1, \dots, i + p - 2$.

Lado derecho

Igualamos $\beta_{j,k} = 0$, para $k = p - 2, p - 1, \dots, 0$. En este caso sí necesitamos almacenar los β 's que hemos calculado para el

lado izquierdo en vectores auxiliares para asegurar la invertibilidad, a menos que realicemos una aproximación lineal. Los α 's modificados son:

$$\alpha_{j,i+m} = \frac{1}{r_{2^{p-2}}} (\alpha_{j+1,2(i+p+m)-1} - \sum_{k=0}^{p-2} [r_{2^k} \alpha_{j,i-k+p+m-1} + d_{2^k} \beta_{j,i-k+p+m-1}]) \quad (4.17)$$

con $m = p - 2, p - 3, \dots, 0$.

Caso de p stencils afectados

Lado izquierdo

Extrapolamos $\alpha_{j,k} \leftrightarrow \hat{\alpha}_{j,k}$ para $k = i, i + 1, \dots, i + p - 1$. Utilizamos (4.16) con $m = 0, 1, \dots, p - 1$.

Lado derecho

Igualamos $\beta_{j,k} = 0$, para $k = p - 1, p - 2, \dots, 0$ y utilizamos (4.17) con $m = p - 1, p - 2, \dots, 0$.

Reconstrucción

Definimos la matriz S de dimensiones $2p \times p$, nula excepto las diagonales $-1, -2, \dots, -p$ que estarán formadas por unos. Las filas de dicha matriz contienen las distintas posibilidades que podemos encontrarnos para $\{s_{j,i}\}_i$. Si $\{s_{j,i-p}, s_{j,i-p+1}, \dots, s_{j,i}\}$ corresponde a la fila n de S , tenemos los siguientes casos:

- Si $n = 0$, no tenemos ningún *stencil* afectado. Por tanto la reconstrucción es como (4.14) y (4.15).
- Si $n > 0$ y $n < p$, estaremos en el caso de n *stencils* afectados, que van a ser utilizados para recuperar datos a la derecha de la discontinuidad. Luego debemos anular $\beta_{j,k}$, para $k = i - p + 1, \dots, i - p + n - 2$. Posteriormente utilizando (4.14) y (4.15), tenemos la reconstrucción buscada.

En estos casos también hemos de tener cuidado en no modificar términos que posteriormente necesitaremos. Al igual que en la transformada directa esto se consigue mediante vectores auxiliares.

- Si $n = p$, todos los *stencils* que utilizamos están afectados por el salto. Nos encontramos con dos casos:

$\alpha_{j,2i-1}$ se halla a la izquierda del salto, luego extrapolamos $\alpha_{j,k}$, para $k = i - p - 1, \dots, i$ y usamos (4.14) y (4.15).

$\alpha_{j,2i}$ se localiza a la derecha del salto. Por tanto anulamos $\beta_{j,k}$ para $k = i - p + 1, \dots, i$ y aplicamos (4.14) y (4.15).

- Si $n > p$ reconstruiremos datos a la izquierda del salto. Luego extrapolamos $\alpha_{j,k}$ para $k = i - 2p + l - 1, \dots, i$ y utilizamos (4.14) y (4.15).

Separación de discontinuidades

Para aplicar *ENO-DB2p* correctamente debe cumplirse la *DSP*, Definición 3.1, ¿será válida para *ENO-DB2p-TI*?

El caso más extremo se da cuando hay p *stencils* afectados y la discontinuidad se halla en $(\alpha_{j+1,2(i+p-1)}, \alpha_{j+1,2(i+p-1)+1})$. Por la izquierda, para hallar $\alpha_{j,i}$ necesitamos extrapolaciones de grado $p-1$, es decir utilizamos el *stencil* $\{\alpha_{j,i-p}, \dots, \alpha_{j,i-1}\}$. El valor más a la izquierda del nivel $j + 1$ necesario para calcular $\alpha_{j,i-p}$ es $\alpha_{j+1,2(i-p)}$. Además en *ENO-DB2p-TI* (ver el caso de p *stencils* afectados) no hacemos uso de ningún valor más a la izquierda que éste. Luego entre $\alpha_{j+1,2(i-p)}$ y $\alpha_{j+1,2(i+p-1)}$ no debe haber discontinuidades, es decir, $4p - 2$ intervalos.

Por la derecha el mayor valor que se utiliza es $\alpha_{j,i+2p-2}$ que proviene de $\alpha_{j+1,2(i+2p-2)+p-1}$. Entre este último y $\alpha_{j+1,2(i+p-1)+1}$ no debe haber discontinuidades, es decir $3p - 4$ intervalos, siendo este valor menor que $4p - 2$.

La *DSP* nos dice que para *stencils* de longitud $l + 1$ debe cumplirse que $(l + 2)h < t$, siendo h la longitud de cada intervalo y t la mínima distancia entre dos discontinuidades. En nuestro caso significará que haya $2p + 1$ intervalos libres de discontinuidades.

Dado que $4p - 2 \geq 2p + 1$ para $p > 1$, debemos utilizar una nueva definición para la DSP:

DEFINICIÓN 4.1. Dado un filtro wavelet con stencils de longitud $l + 1$ con $l > 2$ decimos que la proyección de f en el espacio V_j , con paso espacial $h = 2^{-j}$ satisface la Propiedad de Separación de Discontinuidades para TI (DSP-TI), si $2l \cdot h < t$, con $t = \min\{|x_i - x_j| : x_i, x_j \in D\}$.

4.7

Equivalencia de los métodos ENO-wavelet y ENO-wavelet-TI

En esta sección estudiaremos la posible equivalencia entre los métodos estándar y los basados en la transformada inversa. Empezamos estudiando los diferentes casos para DB4:

Caso 1: un stencil afectado

Lado izquierdo

Nos preguntamos si $(\hat{\beta}_{j,i})_{ENO-DB4} = (\hat{\beta}_{j,i})_{ENO-DB4-TI}$. Veamos:

$$\begin{aligned}
 & (\hat{\beta}_{j,i})_{ENO-DB4} - (\hat{\beta}_{j,i})_{ENO-DB4-TI} = \\
 & = [h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + \frac{h_2}{c_2}(\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1})] \\
 & \quad - [\frac{1}{d_1}(\alpha_{j+1,2i} - r_1\hat{\alpha}_{j,i} - r_3\alpha_{j,i-1} - d_3\beta_{j,i-1})] \\
 & = [h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + \frac{h_2}{c_2}(\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1})] \\
 & \quad - [\frac{1}{d_1}(\alpha_{j+1,2i} - r_1\hat{\alpha}_{j,i} \\
 & \quad - r_3(c_0\alpha_{j+1,2i-2} + c_1\alpha_{j+1,2i-1} + c_2\alpha_{j+1,2i} + c_3\alpha_{j+1,2i+1}) \\
 & \quad - d_3(h_0\alpha_{j+1,2i-2} + h_1\alpha_{j+1,2i-1} + h_2\alpha_{j+1,2i} + h_3\alpha_{j+1,2i+1}))].
 \end{aligned}$$

Los coeficientes que multiplican a cada α son:

$$\hat{\alpha}_{j,i} \longrightarrow \frac{h_2}{c_2} + \frac{r_1}{d_1} = 0,$$

$$\begin{aligned}
\alpha_{j+1,2i-2} &\longrightarrow \frac{1}{d_1} (r_3 c_0 + d_3 h_0) = 0, \\
\alpha_{j+1,2i-1} &\longrightarrow \frac{1}{d_1} (r_3 c_1 + d_3 h_1) = 0, \\
\alpha_{j+1,2i} &\longrightarrow -\frac{1}{d_1} + \frac{1}{d_1} (r_3 c_2 + d_3 h_2) + h_0 - \frac{h_2 c_0}{c_2} = 0, \\
\alpha_{j+1,2i+1} &\longrightarrow h_1 + \frac{1}{d_1} (r_3 c_3 + d_3 h_3) - \frac{h_2 c_1}{c_2} = 0.
\end{aligned}$$

Lado derecho

$$\begin{aligned}
&(\bar{\alpha}_{j,i})_{ENO-DB4} - (\bar{\alpha}_{j,i})_{ENO-DB4-TI} = \\
&= -\frac{c_0}{h_0} (h_2 \alpha_{j+1,2i+1} + h_3 \alpha_{j+1,2i+3}) + c_2 \alpha_{j+1,2i+1} + c_3 \alpha_{j+1,2i+3} \\
&\quad - \frac{1}{r_2} (\alpha_{j+1,2i+3} - r_0 \alpha_{j,i+1} - d_0 \beta_{j,i+1}) \\
&= -\frac{c_0}{h_0} (h_2 \alpha_{j+1,2i+1} + h_3 \alpha_{j+1,2i+3}) + c_2 \alpha_{j+1,2i+1} + c_3 \alpha_{j+1,2i+3} \\
&\quad - \frac{1}{r_2} (\alpha_{j+1,2i+3} - r_0 (c_0 \alpha_{j+1,2i+2} + c_1 \alpha_{j+1,2i+3} + c_2 \alpha_{j+1,2i+4} + c_3 \alpha_{j+1,2i+5}) \\
&\quad - d_0 (h_0 \alpha_{j+1,2i+2} + h_1 \alpha_{j+1,2i+3} + h_2 \alpha_{j+1,2i+4} + h_3 \alpha_{j+1,2i+5})).
\end{aligned}$$

Los coeficientes de cada α son:

$$\begin{aligned}
\alpha_{j+1,2i+1} &\longrightarrow -\frac{c_0}{h_0} h_2 + c_2 = 0, \\
\alpha_{j+1,2i+2} &\longrightarrow -\frac{c_0}{h_0} h_2 + c_2 + \frac{r_0}{r_2} c_0 + \frac{d_0}{r_2} h_0 = 0, \\
\alpha_{j+1,2i+3} &\longrightarrow -\frac{c_0}{h_0} h_3 + c_3 - \frac{1}{r_2} + \frac{r_0}{r_2} c_1 + \frac{d_0}{r_2} h_1 = 0, \\
\alpha_{j+1,2i+4} &\longrightarrow \frac{r_0}{r_2} c_2 + \frac{d_0}{r_2} h_2 = 0, \\
\alpha_{j+1,2i+5} &\longrightarrow \frac{r_0}{r_2} c_3 + \frac{d_0}{r_2} h_3 = 0.
\end{aligned}$$

Caso 2: dos stencils afectados

Lado izquierdo

Empecemos por $\beta_{j,i}$:

$$\begin{aligned}
&(\hat{\beta}_{j,i})_{ENO-DB4} - (\hat{\beta}_{j,i})_{ENO-DB4-TI} = \\
&= h_0 \alpha_{j+1,2i} + h_1 \alpha_{j+1,2i+1} + h_2 \alpha_{j+1,2i+2} + h_3 \hat{\alpha}_{j+1,2i+3} \\
&\quad - \frac{1}{d_1} (\alpha_{j+1,2i} - r_1 \hat{\alpha}_{j,i} - r_3 \alpha_{j,i-1} - d_3 \beta_{j,i-1}) \\
&= h_0 \alpha_{j+1,2i} + h_1 \alpha_{j+1,2i+1} + h_2 \alpha_{j+1,2i+2} \\
&\quad + \frac{h_3}{c_3} (\hat{\alpha}_{j,i} - c_0 \alpha_{j+1,2i} - c_1 \alpha_{j+1,2i+1} + c_2 \alpha_{j+1,2i+2}) \\
&\quad - \frac{1}{d_1} (\alpha_{j+1,2i} - r_1 \hat{\alpha}_{j,i}
\end{aligned}$$

$$\begin{aligned}
& -r_3(c_0\alpha_{j+1,2i-2} + c_1\alpha_{j+1,2i-1} + c_2\alpha_{j+1,2i} + c_3\alpha_{j+1,2i+1}) \\
& -d_3(h_0\alpha_{j+1,2i-2} + h_1\alpha_{j+1,2i-1} + h_2\alpha_{j+1,2i} + h_3\alpha_{j+1,2i+1}).
\end{aligned}$$

Obtenemos:

$$\begin{aligned}
\hat{\alpha}_{j,i} & \longrightarrow \frac{h_3}{c_3} + \frac{r_1}{d_1} = 0, \\
\alpha_{j+1,2i-2} & \longrightarrow \frac{1}{d_1}(r_3c_0 + d_3h_0) = 0, \\
\alpha_{j+1,2i-1} & \longrightarrow \frac{1}{d_1}(r_3c_1 + d_3h_1) = 0, \\
\alpha_{j+1,2i} & \longrightarrow h_0 - \frac{h_3}{c_3}c_0 - \frac{1}{d_1} + \frac{1}{d_1}(r_3c_2 + d_3h_2) = 0, \\
\alpha_{j+1,2i+1} & \longrightarrow h_1 - \frac{h_3}{c_3}c_1 + \frac{1}{d_1}(r_3c_3 + d_3h_3) = 0, \\
\alpha_{j+1,2i+2} & \longrightarrow h_2 - \frac{h_3}{c_3}c_2 = 0.
\end{aligned}$$

Veamos para $\beta_{j,i+1}$:

$$\begin{aligned}
& (\hat{\beta}_{j,i+1})_{ENO-DB4} - (\hat{\beta}_{j,i+1})_{ENO-DB4-TI} = \\
& = h_0\alpha_{j+1,2i+2} + h_1\hat{\alpha}_{j+1,2i+3} \\
& + \frac{h_3}{c_3}(\hat{\alpha}_{j,i+1} - c_0\alpha_{j+1,2i+2} + c_1\hat{\alpha}_{j+1,2i+3}) \\
& - \frac{1}{d_1}(\alpha_{j+1,2i+2} - r_1\hat{\alpha}_{j,i+1} - r_3\hat{\alpha}_{j,i} - d_3\hat{\beta}_{j,i}) \\
& = h_0\alpha_{j+1,2i+2} + \frac{h_1}{c_3}(\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1} - c_2\alpha_{j+1,2i+2}) \\
& + \frac{h_2}{c_2}(\hat{\alpha}_{j,i+1} - c_0\alpha_{j+1,2i+2} - \frac{c_1}{c_3}(\hat{\alpha}_{j,i} - c_0\alpha_{j+1,2i} - c_1\alpha_{j+1,2i+1} - c_2\alpha_{j+1,2i+2})) \\
& - \frac{1}{d_1}(\alpha_{j+1,2i+2} - r_1\hat{\alpha}_{j,i+1} - r_3\hat{\alpha}_{j,i} - d_3(\frac{1}{d_1}(\alpha_{j+1,2i} \\
& - r_1\hat{\alpha}_{j,i} - r_3(c_0\alpha_{j+1,2i-2} + c_1\alpha_{j+1,2i-1} + c_2\alpha_{j+1,2i} + c_3\alpha_{j+1,2i+1}) \\
& - d_3(h_0\alpha_{j+1,2i-2} + h_1\alpha_{j+1,2i-1} + h_2\alpha_{j+1,2i} + h_3\alpha_{j+1,2i+1}))).
\end{aligned}$$

Con esto:

$$\begin{aligned}
\hat{\alpha}_{j,i} & \longrightarrow \frac{h_1}{c_3} - \frac{h_2c_1}{c_2c_3} + \frac{r_3}{d_1} - \frac{d_3r_1}{d_1^2} = 0, \\
\hat{\alpha}_{j,i+1} & \longrightarrow \frac{h_2}{c_2} + \frac{r_1}{d_1} = 0, \\
\alpha_{j+1,2i-2} & \longrightarrow -\frac{d_3}{d_1^2}(r_3c_0 + d_3h_0) = 0, \\
\alpha_{j+1,2i-1} & \longrightarrow -\frac{d_3}{d_1^2}(r_3c_1 + d_3h_1) = 0, \\
\alpha_{j+1,2i} & \longrightarrow -\frac{h_1c_0}{c_3} + \frac{h_2c_1c_0}{c_2c_3} - \frac{d_3}{d_1^2}(r_3c_2 + d_3h_2) + \frac{d_3}{d_1^2} = 0, \\
\alpha_{j+1,2i+1} & \longrightarrow -\frac{h_1c_1}{c_3} + \frac{h_2c_1^2}{c_2c_3} - \frac{d_3}{d_1^2}(r_3c_3 + d_3h_3) = 0,
\end{aligned}$$

$$\alpha_{j+1,2i+2} \longrightarrow h_0 - \frac{h_1 c_2}{c_3} + \frac{h_2 c_1 c_2}{c_2 c_3} - \frac{h_2 c_0}{c_2} - \frac{1}{d_1} = 0.$$

Lado derechoPara $\alpha_{j,i+1}$:

$$\begin{aligned} & (\bar{\alpha}_{j,i+1})_{ENO-DB4} - (\bar{\alpha}_{j,i+1})_{ENO-DB4-TI} \\ &= c_0 \bar{\alpha}_{j+1,2i+2} + c_1 \alpha_{j+1,2i+3} + c_2 \alpha_{j+1,2i+4} + c_3 \alpha_{j+1,2i+5} \\ & \quad - \frac{1}{r_2} (\alpha_{j+1,2i+5} - r_0 \alpha_{j,i+2} - d_0 \beta_{j,i+2}) \\ &= -c_0 \frac{1}{h_0} (h_1 \alpha_{j+1,2i+3} + h_2 \alpha_{j+1,2i+4} + h_3 \alpha_{j+1,2i+5}) \\ & \quad + c_1 \alpha_{j+1,2i+3} + c_2 \alpha_{j+1,2i+4} + c_3 \alpha_{j+1,2i+5} \\ & \quad - \frac{1}{r_2} (\alpha_{j+1,2i+5} - r_0 (c_0 \alpha_{j+1,2i+4} + c_1 \alpha_{j+1,2i+5} + c_2 \alpha_{j+1,2i+6} + c_3 \alpha_{j+1,2i+7}) \\ & \quad - d_0 (h_0 \alpha_{j+1,2i+4} + h_1 \alpha_{j+1,2i+5} + h_2 \alpha_{j+1,2i+6} + h_3 \alpha_{j+1,2i+7})). \end{aligned}$$

Los coeficientes que multiplican a los α son:

$$\begin{aligned} \alpha_{j+1,2i+3} &\longrightarrow -\frac{c_0 h_1}{h_0} + c_1 = 0, \\ \alpha_{j+1,2i+4} &\longrightarrow -\frac{c_0 h_2}{h_0} + c_2 + \frac{r_0 c_0}{r_2} + \frac{d_0 h_0}{r_2} = 0, \\ \alpha_{j+1,2i+5} &\longrightarrow -\frac{c_0 h_3}{h_0} + c_3 + \frac{r_0 c_1}{r_2} + \frac{d_0 h_1}{r_2} - \frac{1}{r_2} = 0, \\ \alpha_{j+1,2i+6} &\longrightarrow \frac{c_2 r_0}{r_2} + \frac{d_0 h_2}{r_2} = 0, \\ \alpha_{j+1,2i+7} &\longrightarrow \frac{c_3 r_0}{r_2} + \frac{d_0 h_3}{r_2} = 0. \end{aligned}$$

Continuando con $\alpha_{j,i}$:

$$\begin{aligned} & (\bar{\alpha}_{j,i})_{ENO-DB4} - (\bar{\alpha}_{j,i})_{ENO-DB4-TI} = \\ &= -\frac{c_0}{h_0} (h_2 \bar{\alpha}_{j+1,2i+2} + h_3 \alpha_{j+1,2i+3}) + c_2 \bar{\alpha}_{j+1,2i+2} + c_3 \alpha_{j+1,2i+3} \\ & \quad - \frac{1}{r_2} (\alpha_{j+1,2i+3} - r_0 \bar{\alpha}_{j,i+1}) \\ &= -\frac{c_0}{h_0} (-h_2 \frac{1}{h_0} (h_1 \alpha_{j+1,2i+3} + h_2 \alpha_{j+1,2i+4} + h_3 \alpha_{j+1,2i+5}) + h_3 \alpha_{j+1,2i+3}) \\ & \quad + c_2 \frac{1}{h_0} (h_1 \alpha_{j+1,2i+3} + h_2 \alpha_{j+1,2i+4} + h_3 \alpha_{j+1,2i+5}) + c_3 \alpha_{j+1,2i+3} \\ & \quad - \frac{1}{r_2} (\alpha_{j+1,2i+3} - r_0 \frac{1}{r_2} (\alpha_{j+1,2i+5} \\ & \quad - r_0 (c_0 \alpha_{j+1,2i+4} + c_1 \alpha_{j+1,2i+5} + c_2 \alpha_{j+1,2i+6} + c_3 \alpha_{j+1,2i+7}) \\ & \quad - d_0 (h_0 \alpha_{j+1,2i+4} + h_1 \alpha_{j+1,2i+5} + h_2 \alpha_{j+1,2i+6} + h_3 \alpha_{j+1,2i+7}))). \end{aligned}$$

Así pues:

$$\alpha_{j+1,2i+3} \longrightarrow \frac{c_0 h_2 h_1}{h_0^2} - \frac{c_0 h_3}{h_0} - \frac{c_2 h_1}{h_0} + c_3 - \frac{1}{r_2} = 0,$$

$$\begin{aligned}
\alpha_{j+1,2i+4} &\longrightarrow \frac{c_0 h_2^2}{h_0^2} - \frac{c_2 h_2}{h_0} - \frac{r_0^2 c_0}{r_2^2} - \frac{r_0 d_0 h_0}{r_2^2} = 0, \\
\alpha_{j+1,2i+5} &\longrightarrow \frac{c_0 h_2 h_3}{h_0^2} - \frac{c_2 h_3}{h_0} + \frac{r_0}{r_2} - \frac{r_0^2 c_1}{r_2^2} - \frac{r_0 d_0 h_1}{r_2^2} = 0, \\
\alpha_{j+1,2i+6} &\longrightarrow -\frac{r_0^2 c_2}{r_2^2} - \frac{r_0 d_0 h_2}{r_2^2} = 0, \\
\alpha_{j+1,2i+7} &\longrightarrow -\frac{r_0^2 c_3}{r_2^2} - \frac{r_0 d_0 h_3}{r_2^2} = 0.
\end{aligned}$$

Con esto queda demostrada la equivalencia para *DB4*.

Procediendo de manera análoga es fácil demostrar la equivalencia para *ENO-DB6* y *ENO-DB6-TI*. Para ello debemos partir de los coeficientes $\{c_i\}$ en su forma exacta y es recomendable utilizar un programa de cálculo simbólico (nosotros hemos utilizado MATHEMATICA®).

Podemos pues enunciar la siguiente proposición:

PROPOSICIÓN 4.1. *Los métodos ENO-DB2p y ENO-DB2p-TI son equivalentes para $p = 2, 3$.*

Todo parece indicar que para $p > 3$ los algoritmos *ENO-DB2p* y *ENO-DB2p-TI* son también equivalentes. Para $p = 4, 5$ los autores W.-C. Chan y C.-C. Yen en [65] han hallado una forma exacta para los coeficientes *scaling* $\{c_i\}$, con lo que se podría demostrar la equivalencia en dichos casos. Sin embargo, para $p > 5$ no disponemos de los coeficientes $\{c_i\}$, en forma exacta y por tanto no podemos demostrar la equivalencia exacta entre los dos esquemas tal como lo hemos hecho para *DB4*. Otra opción sería trabajar con las condiciones que deben cumplir los coeficientes $\{c_i\}$ sin tener en cuenta su valor.

A medida que aumenta p el algoritmo *ENO-DB2p* es más complejo que *ENO-DB2p-TI* ya que, en cada paso, debemos decidir si extrapolar o no en el nivel $j+1$ en función de los datos que conocemos y entendemos que el algoritmo que hemos presentado es más sencillo de generalizar. La equivalencia exacta o no con *ENO-DB2p* para $p > 3$ no ofrece mayor interés, ya que es preferible el uso de *ENO-DB2p-TI*.

Por otra parte, según la *DSP-TI*, Definición 4.1, utilizamos nodos diferentes para *ENO-DB2p* y *ENO-DB2p-TI* con lo cual las re-

construcciones no serían equivalentes. Sin embargo hemos demostrado que para $p = 2, 3$, los coeficientes que multiplican a esos nodos distintos son nulos, con lo que realmente la *DSP-TI* no sería necesaria y podemos utilizar la *DSP* (ya que nuestra hipótesis es que ambos algoritmos son equivalentes $\forall p$). Sin embargo, dado que trabajamos con precisión finita (y partimos de coeficientes $\{c_i\}$ no exactos) es preferible utilizar la *DSP-TI* para *ENO-DB2p-TI*, ya que los valores que multiplican a nodos distintos de los de *ENO-DB2p*, aunque de magnitud muy pequeña, no tienen porqué ser exactamente 0.

4.8

Invertibilidad de los métodos *TI*

Tomemos el esquema *ENO-DB4-TI* y veamos su invertibilidad:

1. $s_{j,i-1} = 0$ y $s_{j,i} = 0$. En este caso no hay discontinuidades y por tanto los datos recuperados son correctos, ya que utilizamos la transformada inversa estándar.

2. $s_{j,i-1} = 0$ y $s_{j,i} = 1$. Evidentemente $\alpha_{j+1,2i}$ es correcto, ya que se define $\beta_{j,i}$ para que cumpla dicho requisito. Sin embargo, debemos comprobar que $\alpha_{j+1,2i+1}$ es correcto:

$$\begin{aligned}
 \alpha_{j+1,2i+1} &:= r_0 \hat{\alpha}_{j,i} + d_0 \hat{\beta}_{j,i} + r_1 \alpha_{j,i-1} + d_1 \beta_{j,i-1} \\
 &= r_0 \hat{\alpha}_{j,i} + \frac{d_0}{d_1} (\alpha_{j+1,2i} - r_1 \hat{\alpha}_{j,i} + r_3 \alpha_{j,i-1} + d_3 \beta_{j,i-1}) + r_1 \alpha_{j,i-1} + d_1 \beta_{j,i-1} \\
 &= (r_0 - \frac{d_0}{d_1} r_1) \hat{\alpha}_{j,i} + \frac{d_0}{d_1} \alpha_{j+1,2i} + (r_2 - \frac{d_0}{d_1} r_2) \alpha_{j,i-1} + (d_2 - \frac{d_0}{d_1} d_2) \beta_{j,i-1} \\
 &= (r_0 - \frac{d_0}{d_1} r_1) \hat{\alpha}_{j,i} \\
 &\quad + \frac{d_0}{d_1} \alpha_{j+1,2i} (r_2 - \frac{d_0}{d_1} r_2) (c_0 \alpha_{j+1,2i-2} + c_1 \alpha_{j+1,2i-1} + c_2 \alpha_{j+1,2i} + c_3 \alpha_{j+1,2i+1}) \\
 &\quad (d_2 - \frac{d_0}{d_1} d_2) (h_0 \alpha_{j+1,2i-2} + h_1 \alpha_{j+1,2i-1} + h_2 \alpha_{j+1,2i} + h_3 \alpha_{j+1,2i+1}).
 \end{aligned}$$

Con lo cual el coeficiente de cada α es:

$$\begin{aligned}
 \hat{\alpha}_{j,i} &\longrightarrow r_0 - \frac{d_0}{d_1} r_1 = 0, \\
 \alpha_{j+1,2i-2} &\longrightarrow (r_2 - \frac{d_0}{d_1} r_3) c_0 + (d_2 - \frac{d_0}{d_1} d_3) h_0 = 0,
 \end{aligned}$$

$$\begin{aligned}
\alpha_{j+1,2i-1} &\longrightarrow (r_2 - \frac{d_0}{d_1}r_3)c_1 + (d_2 - \frac{d_0}{d_1}d_3)h_1 = 0, \\
\alpha_{j+1,2i} &\longrightarrow \frac{d_0}{d_1} + (r_2 - \frac{d_0}{d_1}r_3)c_2 + (d_2 - \frac{d_0}{d_1}d_3)h_2 = 0, \\
\alpha_{j+1,2i+1} &\longrightarrow (r_2 - \frac{d_0}{d_1}r_3)c_3 + (d_2 - \frac{d_0}{d_1}d_3)h_3 = 1.
\end{aligned}$$

Así el valor $\alpha_{j+1,2i+1}$ se recupera correctamente.

3. $s_{j,i-1} = 1$ y $s_{j,i} = 0$. En este caso $\alpha_{j+1,2i+1}$ es correcto, ya que $\bar{\alpha}_{j,i-1}$ se define con este propósito. Veamos $\alpha_{j+1,2i}$:

$$\begin{aligned}
\alpha_{j+1,2i} &:= r_1\alpha_{j,i} + d_1\beta_{j,i} + r_3\bar{\alpha}_{j,i-1} \\
&= r_1(c_0\alpha_{j+1,2i} + c_1\alpha_{j+1,2i+1} + c_2\alpha_{j+1,2i+2} + c_3\alpha_{j+1,2i+3}) \\
&\quad + d_1(h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + h_2\alpha_{j+1,2i+2} + h_3\alpha_{j+1,2i+3}) \\
&\quad + \frac{r_3}{r_2}(\alpha_{j+1,2i+1} - r_0(c_0\alpha_{j+1,2i} + c_1\alpha_{j+1,2i+1} + c_2\alpha_{j+1,2i+2} + c_3\alpha_{j+1,2i+3}) \\
&\quad - d_0(h_0\alpha_{j+1,2i} + h_1\alpha_{j+1,2i+1} + h_2\alpha_{j+1,2i+2} + h_3\alpha_{j+1,2i+3})).
\end{aligned}$$

Los coeficientes que multiplican a cada α son:

$$\begin{aligned}
\alpha_{j+1,2i} &\longrightarrow (r_1 - \frac{r_3r_0}{r_2})c_0 + (d_1 - \frac{r_3d_0}{r_2})h_0 = 1, \\
\alpha_{j+1,2i+1} &\longrightarrow (r_1 - \frac{r_3r_0}{r_2})c_1 + \frac{r_3}{r_2} + (d_1 - \frac{r_3d_0}{r_2})h_1 = 0, \\
\alpha_{j+1,2i+2} &\longrightarrow (r_1 - \frac{r_3r_0}{r_2})c_2 + (d_1 - \frac{r_3d_0}{r_2})h_2 = 0, \\
\alpha_{j+1,2i+3} &\longrightarrow (r_1 - \frac{r_3r_0}{r_2})c_3 + (d_1 - \frac{r_3d_0}{r_2})h_3 = 0.
\end{aligned}$$

4. $s_{j,i-1} = 1$ y $s_{j,i} = 1$. No hay nada que demostrar, ya que $\alpha_{j+1,2i}$ se calcula a partir de $\hat{\beta}_{j,i-1}$ y $\hat{\beta}_{j,i}$, que se definen precisamente para asegurar la recuperación de $\alpha_{j+1,2i}$ y lo mismo ocurre con $\alpha_{j+1,2i}$, que hace uso de los valores $\bar{\alpha}_{j,i-1}$ y $\bar{\alpha}_{j,i}$, definidos igualmente para asegurar la reconstrucción de dicho valor.

Con esto queda demostrada la invertibilidad para *ENO-DB4-TI*. Haciendo uso del software MATHEMATICA® hemos demostrado también la invertibilidad de *ENO-DB6-TI*. Procediendo de manera análoga podríamos demostrar la invertibilidad para *ENO-DB2p-TI*, aunque debemos tener en cuenta las consideraciones expuestas al final del apartado 4.7: o bien trabajamos con coeficientes de escala aproximados (y no podremos comprobar la invertibilidad exacta) o bien debemos hacer uso de las condiciones que deben cumplir dichos coeficientes, como por ejemplo la ortogonalidad.

4.9

Ejemplos

En la Figura 4.4 vemos el resultado de aplicar 5 niveles realizando una aproximación lineal $ENO-DB2p$ para $p = 2, 3$ a la función (1.44) con 2048 puntos iniciales. En ellas mostramos la función original, la reconstrucción y la diferencia entre ambas. Los resultados son excelentes, desapareciendo por completo el *fenómeno de Gibbs*. Notar que, sin tener en cuenta el almacenamiento de la posición de los saltos, hemos conseguido una compresión del $(1 - 32/2048)100 = 98,4\%$ con teóricamente un orden de aproximación 2 y 3 respectivamente.

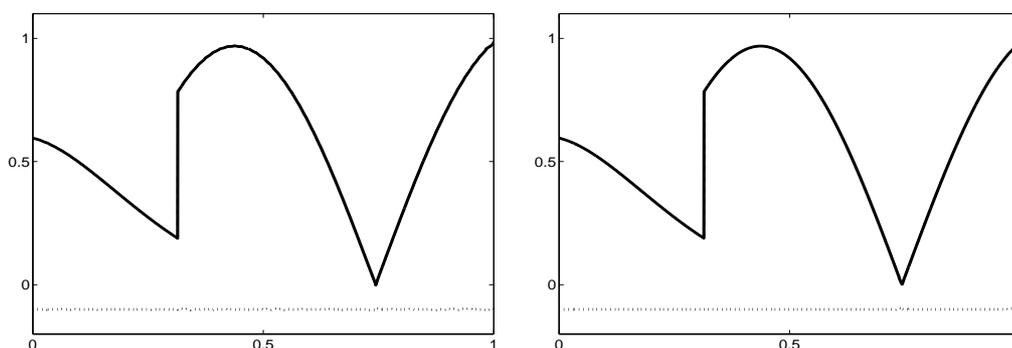


Figura 4.4: Aproximación lineal $ENO-DB2p-TI$ para $p = 2$ (izquierda) y $p = 3$ (derecha). La línea discontinua es la función original (aunque se sobrescribe y no se aprecia), la continua es la aproximación y la punteada es la diferencia entre ambas (desplazada $-0,1$).

En la Figura 4.5 comparamos los coeficientes de las altas frecuencias producidos por $DB4$ y $ENO-DB4$. Podemos comprobar que la magnitud de dichos coeficientes es muy pequeña en todos ellos. Por tanto al eliminarlos la pérdida de información es escasa, lo cual se traduce en una buena aproximación en todos los puntos.

La detección correcta de discontinuidades juega un papel crucial en los esquemas $ENO-wavelet$. En la Figura 4.6 aplicamos

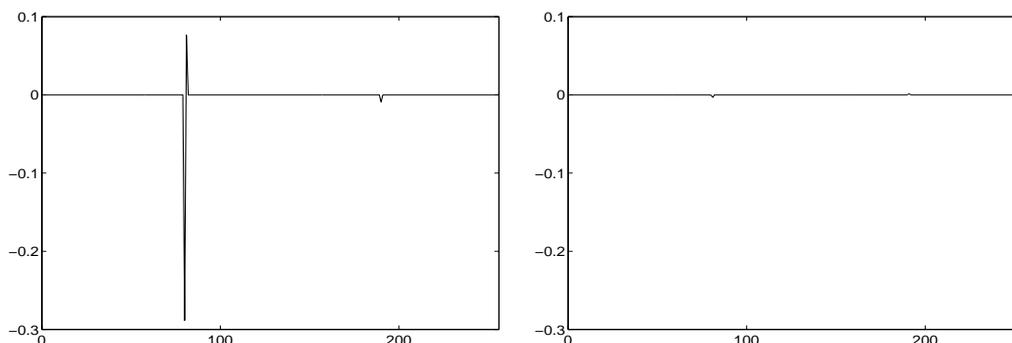


Figura 4.5: Coeficientes de altas frecuencias para DB4 y ENO-DB4 con un nivel, partiendo de la función (1.44) con 512 nodos.

cinco niveles de la transformada ENO-DB6-TI. Visualmente la reconstrucción parece correcta, pero la ampliación de la esquina nos revela que la detección ha fallado en algún paso. Con DB4 este problema no suele ocurrir. La explicación es sencilla, [56]: si partimos de una señal obtenida como muestreo de una función de clase C^p , los coeficientes de altas frecuencias de la transformada DB2p son de orden $O(h^p)$, siendo h el paso espacial de la discretización. De ahí que a mayor p , la compresión y eliminación de ruido mejoran. Sin embargo si se pretenden localizar los puntos donde la pendiente de la señal cambia más bruscamente (esquinas) entonces DB4 es más eficiente que DB2p para $p \geq 3$. Incluso siendo cuidadosos en la elección de las condiciones a y $\beta_{j,i} > \epsilon$, la detección de las esquinas falla en numerosas ocasiones. Podríamos aplicar otro método de detección de discontinuidades como LOCALIZA-ENO-SR-VP-MC (Algoritmo 3.2), aunque también puede fallar en las discontinuidades esquina si utilizamos bastante niveles. En cualquier caso, el error en la detección de discontinuidades esquina no provoca grandes errores en las reconstrucciones.

Para comparar de manera precisa los resultados, utilizaremos los errores L_2 y L_∞ . Para calcular el orden de aproximación, tomamos:

$$\text{Orden}_\infty = \log_2 \frac{E_{\infty,i}}{E_{\infty,i-1}}, \quad \text{Orden}_2 = \log_2 \frac{E_{2,i}}{E_{2,i-1}} \quad (4.18)$$

donde $E_{\infty,i} = \|f(x) - f_j(x)\|_\infty$ y $E_{2,j} = \|f(x) - f_j(x)\|_2$.

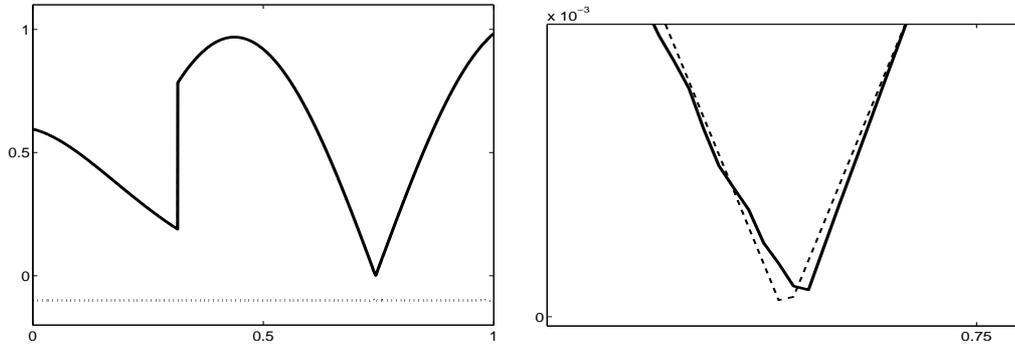


Figura 4.6: Izquierda: 5 niveles de ENO-DB6-TI para la función (1.44) con 2048 nodos. Derecha: ampliación en la discontinuidad esquina.

Definimos la función

$$f(x) = \begin{cases} 0, & \text{si } 0 \leq x < 0,1, \\ -\frac{4x-3}{5} \sin\left(\frac{3}{2}\pi\left(\frac{4x-3}{5}\right)^2\right) - \frac{7}{10}, & \text{si } 0,1 \leq x < \frac{\pi}{10}, \\ \left| \sin 2\pi\left(\frac{4x-3}{5}\right) + \frac{\pi}{100} \right| - \frac{1}{2}, & \text{si } \frac{\pi}{10} \leq x < 0,7, \\ x - \frac{1}{2}, & \text{si } 0,7 \leq x < 0,9, \\ 0, & \text{si } 0,9 \leq x \leq 1, \end{cases} \quad (4.19)$$

sin discontinuidades esquina (para evitar los problemas antes mencionados) y con fronteras nulas. Estudiando los errores vemos que ENO-DB2p-TI mantiene el orden p como nos muestran las Tablas 4.1, 4.2 y 4.3 para $p = 2, 3, 4$, a diferencia del método DB2p donde no se alcanzan dichos órdenes en presencia de discontinuidades.

| nivel | E_∞ | $Orden_\infty$ | E_2 | $Orden_2$ |
|-------|------------|----------------|-----------|-----------|
| 1 | 4.5671e-6 | | 6.1104e-5 | |
| 2 | 3.4454e-5 | 2.9153 | 2.5558e-4 | 2.0645 |
| 3 | 9.3240e-5 | 1.4363 | 1.0271e-3 | 2.0067 |
| 4 | 6.6871e-4 | 2.8423 | 4.3392e-3 | 2.0788 |

Tabla 4.1: E_∞ , E_2 y órdenes de aproximación para ENO-DB4. Hemos utilizado la función (4.19) discretizada con 2048 puntos y las condiciones $a = 2$ y $\beta_{j,i} > 0,001$.

| nivel | E_∞ | $Orden_\infty$ | E_2 | $Orden_2$ |
|-------|------------|----------------|-----------|-----------|
| 1 | 9.4552e-8 | | 1.4113e-7 | |
| 2 | 6.8610e-7 | 2.8592 | 1.1479e-6 | 3.0225 |
| 3 | 1.2568e-5 | 4.1952 | 2.1884e-5 | 4.2527 |
| 4 | 1.3834e-4 | 3.4604 | 3.2290e-4 | 3.9244 |

Tabla 4.2: E_∞ , E_2 y órdenes de aproximación para ENO-DB6-TI. Hemos utilizado la función (4.19) discretizada con 2048 puntos y las condiciones $a = 2$ y $\beta_{j,i} > 0,001$.

| nivel | E_∞ | $Orden_\infty$ | E_2 | $Orden_2$ |
|-------|------------|----------------|------------|-----------|
| 1 | 1.4635e-10 | | 3.7453e-10 | |
| 2 | 6.1745e-9 | 5.3987 | 1.1469e-8 | 4.9335 |
| 3 | 1.5814e-7 | 4.6780 | 3.0279e-7 | 4.7225 |
| 4 | 3.5362e-6 | 4.4829 | 1.0071e-5 | 5.0558 |

Tabla 4.3: E_∞ , E_2 y órdenes de aproximación para ENO-DB8-TI. Hemos utilizado la función (4.19) discretizada con 2048 puntos y las condiciones $a = 2$ y $\beta_{j,i} > 0,001$.

4.10

Conclusiones

En este capítulo hemos revisado el método *ENO-wavelet* propuesto por Chan y Zhou en [26]. A semejanza de la idea *ENO*, se utilizan *stencils* que no cruzan discontinuidades, evitando la creación de coeficientes de escala de magnitud elevada. Así, al realizar la aproximación lineal se elimina el *fenómeno de Gibbs* alrededor de los saltos.

Hemos presentado una nueva manera de calcular los coeficientes *ENO-wavelet*. En el método original para calcular ciertos coeficientes se recurre a extrapolaciones en el nivel más fino, contradiciendo la idea de la *extrapolación en el nivel grosero*. Nosotros hemos utilizado la información de la transformada inversa para dicho cálculo, evitando así el uso de nuevas extrapolaciones. Los nuevos esquemas se han denominado *ENO-wavelet-TI*. Con ellos se consigue generalizar de manera sencilla el método *ENO-DB2p*

TI para p un natural cualquiera. Se ha demostrado la invertibilidad de $ENO-DB2p-TI$ y su equivalencia con $ENO-DB2p$ para $p = 2, 3$ (si $p > 3$ todo parece indicar que también se cumplirá la equivalencia aunque no ha sido demostrado). También se ha adaptado la *Propiedad de Separación de Discontinuidades* (Definición 3.1) al nuevo esquema.

El único problema para aplicar eficientemente el método reside en la localización de las discontinuidades. Se ha trabajado con el localizador propuesto por Chan y Zhou. Funciona correctamente, aunque debemos ser cuidadosos en la elección de los parámetros del método ya que la detección de saltos, y sobre todo esquinas, es muy sensible a dichas condiciones. Dado que el método es en dos pasos, podemos aplicar otro detector de discontinuidades, como los presentados en el capítulo anterior.

En definitiva, para funciones discontinuas compuestas por trozos suaves, el método *ENO-wavelet* consigue eliminar el *fenómeno de Gibbs* al aproximar linealmente, mantiene el orden de aproximación, su coste computacional es comparable al método estándar y pueden ser incluidos en esquemas de multirresolución ya que no varían el tamaño de las secuencias de salida. Además, mediante el uso de la transformada inversa se evitan extrapolaciones en el nivel fino, manteniendo todas las propiedades de la transformada *ENO-wavelet* y permitiendo generalizar el esquema de forma sencilla.

5

Aplicaciones a la eliminación de ruido

5.1

Introducción

En este capítulo estudiaremos la capacidad de las reconstrucciones y transformadas vistas en capítulos anteriores para la eliminación de ruido en señales digitales.

Una de las causas del deterioro de las señales digitales es el ruido. La definición y tipo de ruido depende del contexto en el que trabajemos. Podemos definir el ruido como aquel elemento independiente de la señal que interviene en la comunicación, procesamiento o medida de dicha señal, [27]. En el contexto de transmisión de señales por cable, podemos hablar de ruido térmico (debido al mo-

vimiento aleatorio de los electrones en un cable), inducido (debido a fuentes externas como motores y electrodomésticos), cruces (debido a el efecto de un cable sobre otro al actuar uno como antena emisora y otro como receptora) e impulsos (debido a picos producidos por líneas de potencia, iluminación...). En el contexto del sonido podemos encontrar el ruido aditivo (debido a la adición de otra señal procedente de fuentes que coexisten en el entorno), reverberación (producida por la propagación multitrayecto que se da en los entornos acústicos cerrados o semicerrados) y eco (producido generalmente por el acoplamiento entre micrófonos y altavoces). Aunque estos dos últimos no se tratan exactamente de ruido sino de distorsión (la señal cambia su forma de onda). En el contexto de imágenes el ruido se introduce en la captación de la imagen (los sensores utilizados para digitalizar pueden dañarse, o afectarles las condiciones climáticas, fuentes electromagnéticas, ruido eléctrico...), o puede introducirse en el proceso de transmisión.

Nosotros nos centraremos únicamente en ruido aditivo. Al introducir matemáticamente el ruido también podemos hacerlo de varias maneras, mediante una distribución uniforme, gaussiana, gamma, exponencial,... Además con correlación en el tiempo o no. Cada distribución modelará mejor un tipo de caso real¹ que contamina las señales digitales, aunque el modelo que cubre un mayor número de casos es el ruido *blanco gaussiano*. Será pues éste el que introduciremos en el presente trabajo.

Al igual que en la sección 3.3 consideraremos funciones del tipo $\tilde{f}(x) = f(x) + n(x)$, con $f(x)$ una función discontinua compuesta por trozos suaves y $n(x)$ el ruido *blanco gaussiano*.

Como medida objetiva para medir el ruido en una señal, a parte de SNR, (3.7), utilizaremos también el *Error Cuadrático Medio*:

$$ECM(f, \tilde{f}) = \frac{1}{N^2} \sum_{i=1}^N (f_i - \tilde{f}_i)^2. \quad (5.1)$$

¹Por ejemplo si el CCD de la cámara tiene polvo, el ruido que se introduce es de tipo *sal y pimienta* o de *impulso*, que correspondería a una función de densidad $P(z)$ tal que $P(z) = P_a$ para $z = a$, $P(z) = P_b$ para $z = b$ y $P(z) = 0$ en otro caso.

5.2

Trabajando en una escala

En esta sección veremos varios métodos de reducción de ruido pero trabajando únicamente en una escala.

5.2.1

Aproximación por Mínimos Cuadrados (AMC) entre discontinuidades

Si la función no presenta discontinuidades, los mínimos cuadrados polinómicos son una excelente solución para el problema planteado. Sin embargo, si la función contiene discontinuidades no podemos aplicar mínimos cuadrados directamente, ya que la aproximación obtenida es un polinomio continuo y por tanto difiere en exceso de la función a reconstruir.

Motivados por los trabajos de Daphna Mizrahi [60], una sencilla solución para abordar el problema de las discontinuidades consiste en:

1. Localizar las discontinuidades.
2. Aproximar por mínimos cuadrados cada trozo entre las discontinuidades.

El problema se basa entonces en la localización correcta de las discontinuidades, punto que fue tratado el capítulo 3. Para funciones con ruido disponemos de varios métodos que localizan discontinuidades de salto pero no esquinas.

Localización de discontinuidades

En [60] se utiliza el método de localización basado en la combinación de las medidas E_1 (3.8) y E_2 (3.9). Como se comprobó en el capítulo 3 dicho método era poco fiable y diseñamos mejores

alternativas de localización. Nos hemos decantado por la *reconstrucción por aproximaciones de stencil creciente* ($\overline{AP} - SC$, pag. 159) ya que, según los experimentos numéricos, produce escasas falsas discontinuidades y detecta correctamente saltos incluso con funciones en las que el ruido introducido es elevado.

Grado de la AMC de cada trozo entre discontinuidades

Si suponemos localizadas correctamente las discontinuidades debemos establecer un algoritmo que permita obtener el grado óptimo para la AMC en cada trozo suave.

El parámetro de correlación

$$r(\epsilon) = \frac{\sum_{i=2}^n (\epsilon_i \epsilon_{i-1})}{\sum_{i=2}^n \epsilon_i^2}, \quad (5.2)$$

varía en el intervalo $[-1, 1]$. Los valores cercanos a 0 indican no correlación, mientras que los valores cercanos a 1 y -1 nos revelan una estructura correlada. Como ya indicamos, nosotros trabajamos con ruido no correlado, y por tanto si ϵ representa el ruido, $r(\epsilon) = 0$. Este hecho puede ayudarnos a nuestro propósito, ya que si hemos realizado una buena eliminación de ruido a una señal suave, la diferencia entre la señal con ruido y la que hemos obtenido tras eliminarlo, producirá una serie de datos no correlados, pues constituirá el error (en el caso ideal) que hemos introducido. Dicho en otros términos, si f representa la señal inicialmente suave, \tilde{f} , a la que añadimos ruido, y \hat{f} la obtenida tras su eliminación, debería ocurrir que $r(f - \hat{f}) \approx 0$.

Sea $P(x, \tilde{f})$ la función polinómica a trozos que aproxima \tilde{f} en los puntos $\{x_i\}$. Entre dos puntos en los que \tilde{f} es discontinua, $x_{d_i} < x < x_{d_{i+1}}$, P viene dada mediante la AMC, de grado m :

$$P(x; \tilde{f}) \equiv q^{\overline{AP}}(x; \tilde{f}, m) \text{ para } x_{d_i} < x < x_{d_{i+1}}.$$

Observemos las siguientes cantidades para $d_i < s \leq d_{i+1} - 1$:

- Los elementos de ruido: $\epsilon(x_s) \equiv \tilde{f}(x_s) - f(x_s)$,

- El error de aproximación: $E_m(x_s) \equiv q^{\overline{\mathcal{AP}}}(x_s; \tilde{f}, m) - f(x_s)$,
- Los residuales: $e_m(x_s) = q^{\overline{\mathcal{AP}}}(x; \tilde{f}, m) - \tilde{f}(x_s)$.

Las dos primeras cantidades no son calculables ya que desconocemos $f(x)$. Sin embargo podemos poner la última en función de las anteriores:

$$\begin{aligned} e_m(x_s) &= q^{\overline{\mathcal{AP}}}(x_s; \tilde{f}, m) - \tilde{f}(x_s) = q^{\overline{\mathcal{AP}}}(x; \tilde{f}, m) - f(x_s) + f(x_s) - \tilde{f}(x_s) \\ &= E_m(x_s) - \epsilon(x_s). \end{aligned} \quad (5.3)$$

Así, los residuales dependen del error de aproximación (que a su vez depende de r) y de los elementos de ruido (que siguen una distribución con estructura no correlada). Podemos extraer las siguientes conclusiones:

- Para m pequeño:

$$q^{\overline{\mathcal{AP}}}(x; \tilde{f}, m) \approx q^{\overline{\mathcal{AP}}}(x; f, m) \implies E_m(x) \approx q^{\overline{\mathcal{AP}}}(x; f, m) - f(x).$$

- Al incrementar m (para m suficientemente pequeño) E_m es mucho menor que el ruido y por tanto $r(e_m) \approx r(\epsilon) \approx 0$.
- Para valores altos de m , $q^{\overline{\mathcal{AP}}}(x; \tilde{f}, m)$ se acerca a \tilde{f} .

De lo anterior deducimos que lo que nos interesa buscar es el menor m tal que $r(e_m) \approx 0$. El estadístico más comúnmente utilizado para medir el parámetro de correlación es:

$$d = \frac{\sum_{i=2}^n (\epsilon_i - \epsilon_{i-1})^2}{\sum_{i=2}^n \epsilon_i^2}. \quad (5.4)$$

Existe la siguiente relación entre d y r : $d = 2(1 - r)$. Por tanto, que $r(\epsilon) = 0$ equivale a $d(\epsilon) \approx 2$. Nos interesa minimizar la cantidad $|d(e_m) - 2|$ y utilizar un test de autocorrelación. El estadístico d se utiliza para verificar la hipótesis nula H_0 (residuales no correlados), frente la hipótesis alternativa H_1 (correlados). Un conocido test de autocorrelación es el de Durbin-Watson², que funciona de la siguiente manera: calculamos el estadístico d , si

²En MATLAB® podemos utilizar el programa `dwttest.m` para dicho test.

1. $d < d_L$ rechazar H_0 ,
2. $d > d_U$ no rechazar H_0 ,
3. $d_L < d < d_U$ el test es inconcluyente.

Los valores (d_L, d_U) , según el número de muestras, fueron tabulados por Durbin y Watson (ver [37]), y posteriormente ampliados en [64] por Saving y White.

El Algoritmo 5.1 resume lo expuesto para hallar el grado óptimo, m_0 , según cada trozo entre discontinuidades.

Algoritmo 5.1 Grado óptimo para *AMC* entre discontinuidades

Entrada: $\{f_i\}_{d_i \leq i < d_{i+1}}$, los datos entre discontinuidades, m_{max} el máximo grado admitido

Salida: m_0 , el grado óptimo

```

1:  $m = 1$ 
2:  $entra = 0$ 
3: cálculo de  $e_1$  y  $d(e_1)$ 
4: mientras  $m \leq m_{max}$  hacer
5:   cálculo de  $e_m$  y  $d(e_m)$ 
6:   si  $d(e_m) > d_U$  entonces
7:     si  $|d(e_{m-1}) - 2| < |d(e_m) - 2|$  entonces
8:        $m_0 = m - 1$ ;
9:        $m = m_{max}$ ;
10:       $entra = 1$ ;
11:     si no
12:        $m = m + 1$ ;
13:     fin si
14:   si no
15:      $m = m + 1$ ;
16:   fin si
17: fin mientras
18: si  $entra == 0$  entonces
19:    $m_0 = m_{max}$ ;
20: fin si

```

Experimentos numéricos

Hemos realizado varios experimentos con funciones que contienen saltos y esquinas. Se parte de 256 puntos iniciales y se utiliza grado 3 para realizar el método de detección de discontinuidades $\overline{\mathcal{AP}} - SC$. En cada gráfico aparece la función sin ruido, con ruido y la aproximación, indicándose los grados calculados para cada trozo suave y el ECM y SNR entre la función original con ruido y la reconstrucción³. El grado máximo admitido, m_{max} , es 10.

En la Figura 5.1 (a) partimos de la función (1.44) a la que añadimos ruido blanco gaussiano con $SNR=24$. La función contiene un salto que es detectado correctamente. El primer trozo se aproxima con un polinomio de grado 1, mientras que el segundo trozo se aproxima con un polinomio de grado 10, ya que la esquina no se localiza y por tanto se necesita un grado elevado para el polinomio por *aproximaciones*. En (b) la función de partida contiene dos saltos y una esquina, (3.15), con la misma magnitud de ruido. Los grados hallados son 5, 8 y 3. El grado 8 corresponde a la zona central que incluye una discontinuidad esquina que no ha sido detectada.

En general el método proporciona resultados aceptables siempre que se localicen correctamente los saltos. Si esto no ocurre, se obtienen soluciones que distan mucho de la función original, lo cual motiva a que sigamos analizando otros métodos de eliminación de ruido.

5.2.2

Reconstrucción $\overline{\mathcal{AP}}$ para cada punto

En este apartado estudiaremos la aplicación de mínimos cuadrados 'punto a punto', es decir, se tomará un determinado *stencil* que contenga el punto y el valor de la función se sustituirá por el valor del polinomio $q^{\overline{\mathcal{AP}}}$ en dicho punto.

³También podríamos haber calculado el error entre la función original sin ruido y la reconstrucción. Pero dado que la función original sin ruido nos es, en principio, desconocida, optamos por calcular (mientras no indiquemos lo contrario) los errores entre la original con ruido y la reconstrucción.

De forma más rigurosa, dada $\{(x_i, f_i)\}_{i=1}^N$ la función inicial, para cada x_i se calcula $q_i^{\overline{\mathcal{AP}}^{nl, nr}}(x; f, r)$ como el polinomio de grado r obtenido por mínimos cuadrados calculados a partir de los $m = nl + nr \geq r + 2$ puntos $\{x_{i-nl}, \dots, x_{i+nr-1}\}$. Con esto sustituimos cada f_i por $q_i^{\overline{\mathcal{AP}}^{nl, nr}}(x_i; f, r)$.

Si centramos el intervalo de trabajo, es decir $nl = nr$ obtenemos la Figura 5.2. En ella observamos:

- El aumentar el grado del polinomio no mejora la eliminación de ruido, ya que la aproximación de los datos con ruido aumenta.
- Al aumentar la longitud del *stencil* los resultados mejoran, aunque los saltos quedan más suavizados y aumenta el *fenómeno de Gibbs*.

Para resolver el problema planteado en el segundo punto debemos evitar la utilización de *stencils* que crucen discontinuidades.

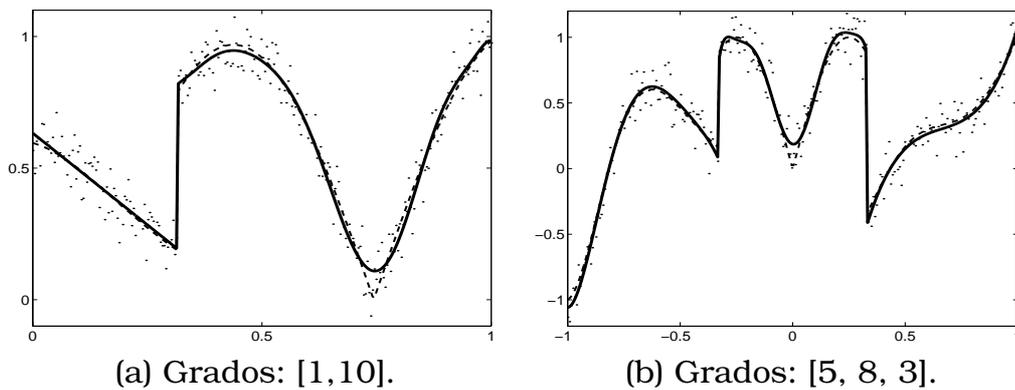


Figura 5.1: Aplicamos AMC entre discontinuidades, determinadas por el método $\overline{\mathcal{AP}} - SC$ a las funciones (1.44) en (a) y (3.15) en (b), ambas con 256 nodos y $SNR = 24$. El grado de cada trozo, determinado por el test de Durbin-Watson, se indica bajo la figura. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM = 1.79e-5$, $SNR = 19.53$; (b) $ECM = 5.20e-5$, $SNR = 14.36$.

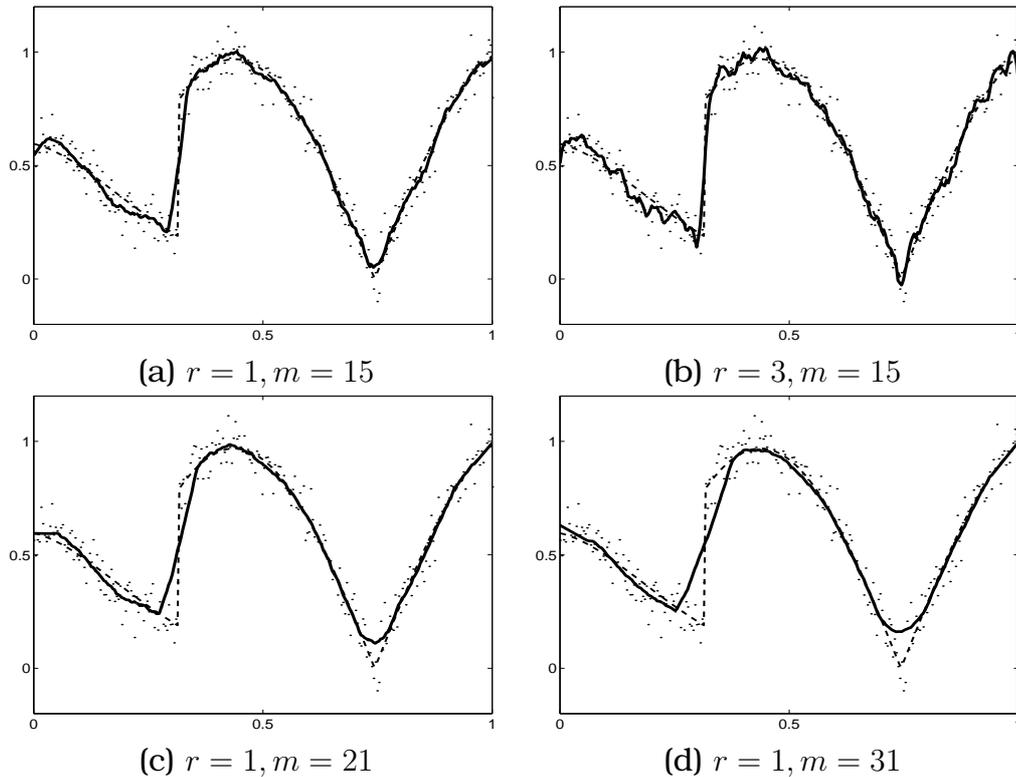


Figura 5.2: \overline{AP} para cada punto con stencil centrado (en las fronteras sí se adapta) aplicado a la función (1.44) con 256 puntos y $SNR = 24$. r es el grado del polinomio de la reconstrucción, m es el tamaño del stencil. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=1.81e-5$, $SNR=19.43$; (b) $ECM=1.42e-5$, $SNR=20.51$; (c) $ECM=2.41e-5$, $SNR=18.13$; (d) $ECM=3.26e-5$, $SNR=16.77$.

Evitando cruzar discontinuidades

Aplicaremos la idea *ENO* eligiendo *stencils* que no crucen discontinuidades. Dado que trabajamos con ruido, el elegir el *stencil* de menor diferencia dividida no asegura que no se cruce un salto. Podemos recurrir a otras medidas, como E_1 y E_2 , introducidas en la sección 3.3.1, que consiguen evitar utilizar *stencils* que cruzan discontinuidades para funciones que contienen ruido.

En la Figura 5.3 representamos las soluciones que obtenemos al elegir el *stencil* que minimiza el valor de E_1 o E_2 . Al comparar con métodos centrados, se observa que la disminución del error es menor, aunque se consigue no suavizar el salto en la mayoría de casos. La disminución de ruido es más escasa ya que para un cierto intervalo de puntos se utiliza el mismo *stencil* (el que minimiza E_1 o E_2) y no uno diferente cada vez como en el caso centrado.

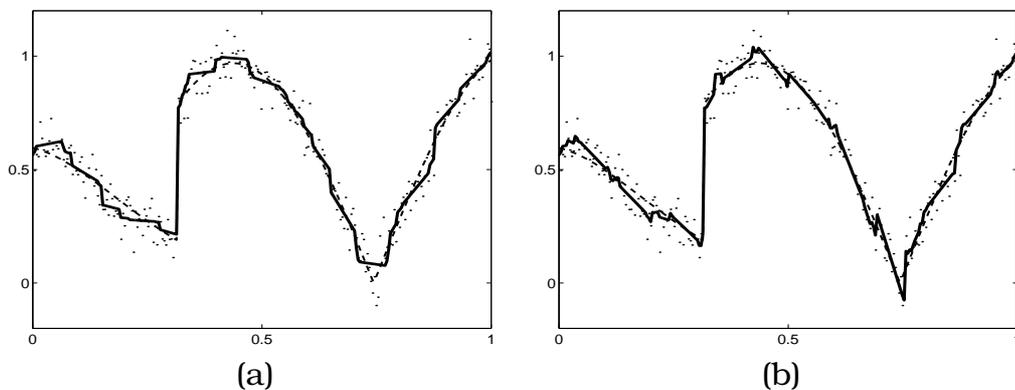


Figura 5.3: Tomando 256 puntos de la función (1.44), con $SNR = 24$, sustituimos cada punto la reconstrucción \overline{AP} de grado 1 y longitud de stencil 31, eligiendo el stencil según la medida E_1 en (a) y según E_2 en (b). La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=1.53e-5$, $SNR=20.26$; (b) $ECM=1.20e-5$, $SNR=21.29$.

Stencil centrado excepto en las discontinuidades

Hemos visto que al tomar *stencils* centrados, la eliminación de ruido es mayor excepto en las zonas de discontinuidad. Si evitamos cruzar los saltos, mejoramos en la discontinuidad pero empeoramos la eliminación de ruido en las zonas suaves. Para aprovechar los beneficios de cada caso planteamos el siguiente esquema: localizamos las discontinuidades y utilizamos un *stencil* centrado, excepto en los puntos donde se crucen saltos. En dichos puntos se toman *stencils* que no crucen el salto. El Algoritmo 5.2 muestra este esquema. Una vez más la localización de discontinuidades juega un papel importante, aunque si hubiese algún error

no sería tan grave como en el método de aproximar por AMC entre discontinuidades. En este caso si no se localiza un salto, simplemente quedará suavizado.

Algoritmo 5.2 Reconstrucción \overline{AP} con *stencil* centrado excepto en las discontinuidades.

Entrada: Datos iniciales $\{f_i\}_{i=1}^N$, localización de discontinuidades $\{d_i\}_{i=1}^p$, m tamaño del *stencil*, r grado del polinomio \overline{AP} .

Salida: Aproximación obtenida $\{\tilde{f}_i\}_{i=1}^N$.

```

1: para  $j = 2, \dots, p$  hacer
2:    $k = 0$ ;
3:   para  $i = d_{j-1}, \dots, d_{j-1} + \frac{m-1}{2} - 1$  hacer
4:      $\tilde{f}_i = q^{\overline{AP}^r}$  con stencil  $\{x_{i-k}, \dots, x_{i+m-1-k}\}$ 
5:      $k = k + 1$ 
6:   fin para
7:   para  $i = d_{j-1} + \frac{m-1}{2}, \dots, d_j - \frac{m-1}{2}$  hacer
8:      $\tilde{f}_i = q^{\overline{AP}^r}$  con stencil  $\{x_{i-\frac{m-1}{2}}, \dots, x_{i+\frac{m-1}{2}}\}$ 
9:   fin para
10:   $k = 0$ 
11:  para  $i = d_j - \frac{m-1}{2}, \dots, d_j - 1$  hacer
12:     $k = k + 1$ 
13:     $\tilde{f}_i = q^{\overline{AP}^r}$  con stencil  $\{x_{i-\frac{m-1}{2}-k}, \dots, x_{i+\frac{m-1}{2}-k}\}$ 
14:  fin para
15: fin para

```

En la Figura 5.4 hemos aplicado este método para la funciones con uno y dos saltos. El método de localización de discontinuidades será $\overline{AP} - SC$ con grado 3. Apreciamos la correcta aproximación en los saltos en todos los casos, ya que han sido localizados correctamente. Las discontinuidades esquina quedan suavizadas ya que el método de detección no las localiza. Si aumentamos el grado del polinomio de la reconstrucción \overline{AP} , la eliminación de ruido empeora, aunque la aproximación en las esquinas mejora.

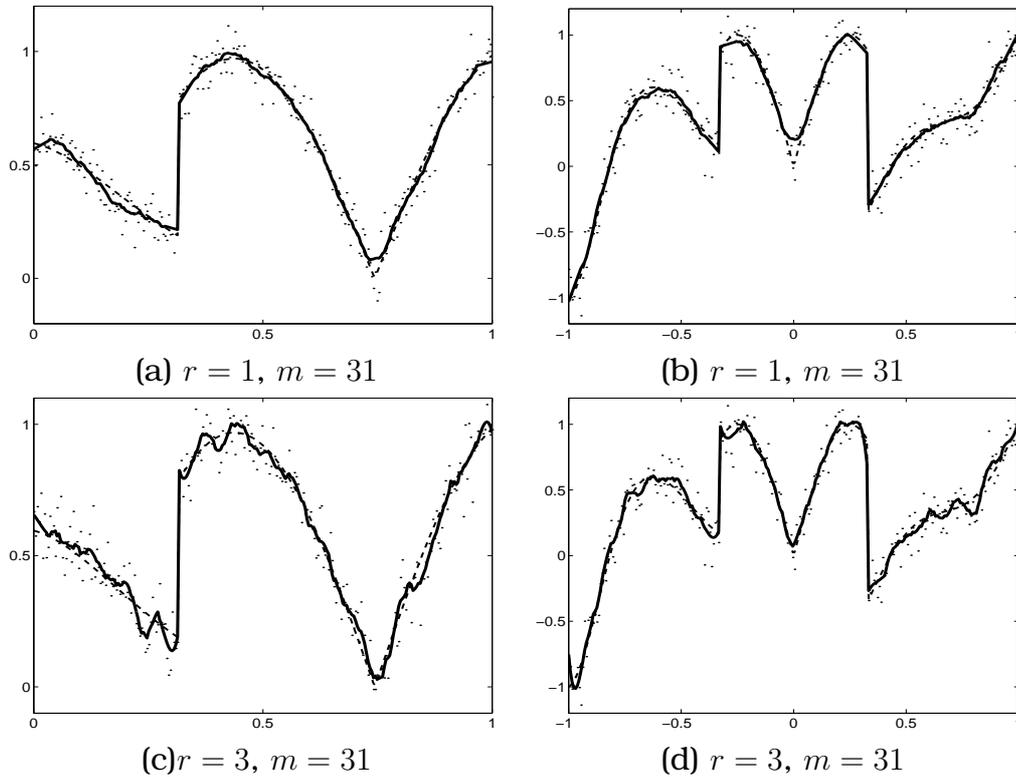


Figura 5.4: Tomando 256 puntos de (1.44), con $SNR = 24$, se sustituye cada punto por el valor de \overline{AP} de grado r y longitud de stencil m , con stencil centrado excepto en las discontinuidades, localizadas mediante $\overline{AP} - SC$ (Algoritmo 5.2). La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=1.55e-5$; $SNR=20.10$; (b) $ECM=5.07e-5$, $SNR=14.22$; (c) $ECM=1.75e-5$, $SNR=19.60$; (d) $ECM=3.92e-5$, $SNR=15.46$.

5.2.3

Iteración por sustitución de valores medios

En este apartado presentamos un sencillo algoritmo de reducción de ruido que no necesita calcular reconstrucciones. El algoritmo se basa en sustituir cada punto por la media de él mismo y

su anterior, excepto si se detecta una posible discontinuidad, en cuyo caso se sustituye por la media entre él mismo y el siguiente. Posteriormente este paso se itera.

La condición de discontinuidad es la siguiente:

$$factor \cdot \frac{1}{i-2} \cdot \sum_{j=2}^{i-1} |\bar{f}_j - f_j| < |\bar{f}_i - f_i|, \quad (5.5)$$

donde \bar{f}_i representa la media $|f_i - f_{i-1}|/2$.

El Algoritmo 5.3 muestra este proceso. Tomando un valor adecuado de *factor* se consigue que la media de los valores se calcule a partir de datos de un solo lado de la discontinuidad. Experimentalmente hemos comprobado que valores de *factor* iguales a 5 o 6 dan buenos resultados. La magnitud de *factor* determinará el tamaño de los saltos que respetaremos. En el algoritmo sobreescribimos f_i en cada iteración con lo que se consigue un aumento de la velocidad de eliminación de ruido. Dado que la mayoría de valores se sustituyen por la media de él mismo con su anterior, se produce un desplazamiento de la función hacia la derecha. Para compensarlo creamos el Algoritmo 5.4, donde la media se realiza con el siguiente valor. Así pues, tras aplicar el algoritmo que realiza la media a la izquierda, se aplica el que la realiza a la derecha, compensando de esta manera el desplazamiento. Posteriormente se itera el proceso, obteniendo el Algoritmo 5.5.

Veamos algunos ejemplos. En la Figura 5.5 aplicamos 8 iteraciones a varias funciones en las que se han introducido diferentes magnitudes de ruido. Con tan sólo 8 iteraciones se consigue una eliminación eficiente de ruido. El aumentarlas no mejora los resultados ya que las esquinas se siguen suavizando con cada nueva iteración.

5.3

Trabajando con varias escalas

Los esquemas de multirresolución son utilizados satisfactoriamente para la compresión de señales. Al bajar cada nivel se elimi-

Algoritmo 5.3 Media de cada valor con su anterior excepto en posibles discontinuidades.

Entrada: $\{f_i\}_{i=1}^n$, la función de partida; *factor*, determina la magnitud de los saltos respetados.

Salida: $\{f_i\}_{i=1}^n$

- 1: $err_2 = |f_2 - f_1|/2$
- 2: **para** $i = 3, 4, \dots, n - 1$ **hacer**
- 3: $aux = f_i$
- 4: $f_i = (f_{i-1} + f_i)/2$
- 5: $err_i = |aux - f_{i-1}|/2$
- 6: **si** $factor * \frac{1}{i-2} \sum_{j=2}^{i-1} err_j < err_i$ **entonces**
- 7: $f_i = (aux + f_{i+1})/2$
- 8: $err = |f_i - f_{i+1}|/2$
- 9: **fin si**
- 10: $f_1 = (f_1 + f_2)/2$
- 11: $f_2 = (f_2 + f_3)/2$
- 12: $f_n = (f_{n-1} + f_n)/2$
- 13: **fin para**

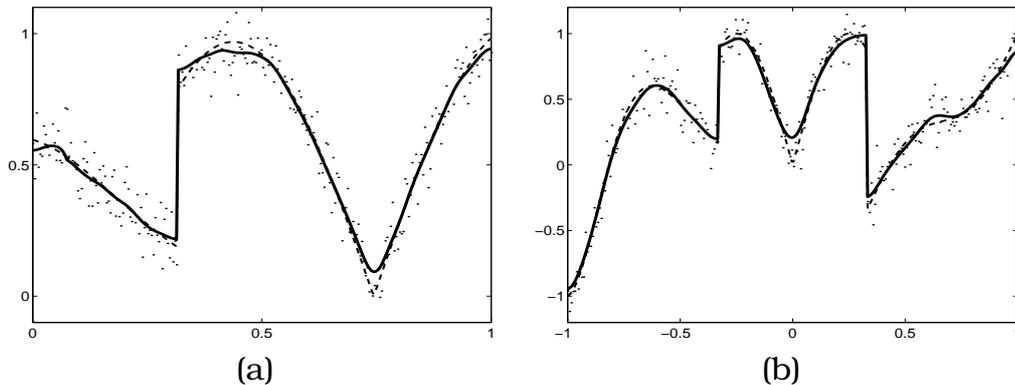


Figura 5.5: 8 iteraciones del Algoritmo 5.5. Partimos con 256 puntos de las funciones (1.44) en (a), y (3.15) en (b), con ruido de magnitud $SNR = 24$. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=1.96e-5$, $SNR=29.53$; (b) $ECM=4.88e-5$, $SNR=22.80$.

Algoritmo 5.4 Media de cada valor con su siguiente excepto en posibles discontinuidades.

Entrada: $\{f_i\}_{i=1}^n$, la función de partida; *factor*, determina la magnitud de los saltos respetados

Salida: $\{f_i\}_{i=1}^n$

```

1:  $err_n = |f_n - f_{n-1}|/2$ ;
2: para  $i = n - 1, n - 2, \dots, 2$  hacer
3:    $aux = f_i$ 
4:    $f_i = (f_i + f_{i+1})/2$ 
5:    $err_i = |f_{i+1} - aux|/2$ 
6:   si  $factor \cdot \frac{1}{n-i} \sum_{j=i+1}^n err_j < err_i$  entonces
7:      $f_i = (aux + f_{i-1})/2$ 
8:      $err_i = |f_i - f_{i-1}|/2$ 
9:   fin si
10:   $f_1 = (f_1 + f_2)/2$ 
11:   $f_n = (f_{n-1} + f_n)/2$ 
12: fin para

```

na información prescindible para recuperar la señal del nivel superior bajo una cierta tolerancia. Esta idea guarda gran relación con el problema aquí tratado, ya que dicha información prescindible puede interpretarse como ruido en este caso. Es por ello que en esta sección trabajaremos con varias escalas.

Comenzaremos estudiando las posibilidades que ofrecen las descomposiciones multiescala vistas en los capítulos 1 y 2. Posteriormente revisaremos la eliminación de ruido utilizando *wavelets* y finalmente veremos si estas ideas se pueden adaptar a las descomposiciones multiescala.

5.3.1

Eliminación de ruido mediante descomposiciones multiescala à la Harten

Debemos plantearnos cuáles son las mejores opciones que conducen a nuestro objetivo: la eliminación de ruido en funciones

Algoritmo 5.5 Eliminación de ruido mediante proceso iterativo por sustitución de medias.

Entrada: $\{f_i\}_{i=1}^n$, la función de partida; $factor$, determina la magnitud de los saltos respetados; $iter$, número de iteraciones

Salida: $\{f_i\}_{i=1}^n$, la función de salida

- 1: **para** $i = 1, 3, \dots, iter$ **hacer**
 - 2: $\{f_i\}_{i=1}^n \leftarrow$ Algoritmo 5.3 con datos iniciales $\{f_i\}_{i=1}^n, factor$
 - 3: $\{f_i\}_{i=1}^n \leftarrow$ Algoritmo 5.4 con datos iniciales $\{f_i\}_{i=1}^n, factor$
 - 4: **fin para**
-

compuestas por trozos suaves. Para la decimación hemos estudiado dos posibilidades: los valores puntuales y las medias en celda. Para la predicción tres: reconstrucciones por *interpolación*, por *interpolación aproximación* y por *aproximaciones*.

En cuanto a la decimación, si utilizamos valores puntuales al bajar de niveles no se produce ninguna eliminación de ruido. Con esto, tras bajar n niveles nos encontramos con una función con menos nodos pero con la misma magnitud de ruido que la inicial. A continuación, si queremos eliminar ruido, podemos utilizar $\overline{\mathcal{AP}}$. Aquí nos encontramos con dos problemas:

1. *ENO* basado en la magnitud de diferencias divididas no es adecuado, ya que los datos contienen ruido.
2. La reconstrucción basada en

$$\begin{cases} \text{Para } k = 1, \dots, L \\ f_{2i}^k = f_i^{k-1} & i = 0, \dots, J_{k-1}, \\ f_{2i-1}^k = \overline{\mathcal{AP}}(x_{2i-1}^k; f^{k-1}) & i = 1, \dots, J_{k-1}. \end{cases} \quad (5.6)$$

no contribuye a la eliminación de ruido, ya que los nodos pares no se modifican.

Aunque estos problemas podrían solucionarse utilizando la medida E_2 y el algoritmo que aproxima también en los pares (2.42), los saltos quedarán también suavizados, ya que *SR* no mejora la aproximación en ellos. Así pues, el uso de medias en celda proporcionará mejores resultados en cuanto a eliminación de ruido que valores puntuales. Es por ello que descartamos trabajar con

valores puntuales y nos centramos en medias en celda, donde la eliminación de ruido es doble: al decimar y al predecir

Como puede comprobarse en la Figura 5.6, si bajamos el suficiente número de niveles no es necesario modificar la asignación *ENO*, ya que el ruido disminuye y las diferencias divididas pueden mantenerse. Aunque este 'suficiente' número de niveles va a depender del ruido inicial introducido, que a priori nos es desconocido. Como alternativa podemos sustituir las diferencias divididas por E_2 que también evita discontinuidades y puede ser utilizado en presencia de ruido. Para reconstruir f_{2i-1}^k debemos utilizar el stencil $\{x_{i_2(i)}^{k-1}, x_{i_2(i)+1}^{k-1}, \dots, x_{i_2(i)+m-1}^{k-1}\}$, donde $m = nl + nr + 1$, y $i_2(i)$ se halla según (3.9), (3.10) y (3.11).

Llegados a este punto existen varias combinaciones de algoritmos que podemos aplicar. Intentaremos razonar qué opciones son las mejores:

Reconstrucción interpolatoria, \overline{IC} : utilizamos diferencias divididas para la asignación *ENO*, ya que para E_2 los *stencils* no son de suficiente longitud como para realizar mínimos cuadrados.

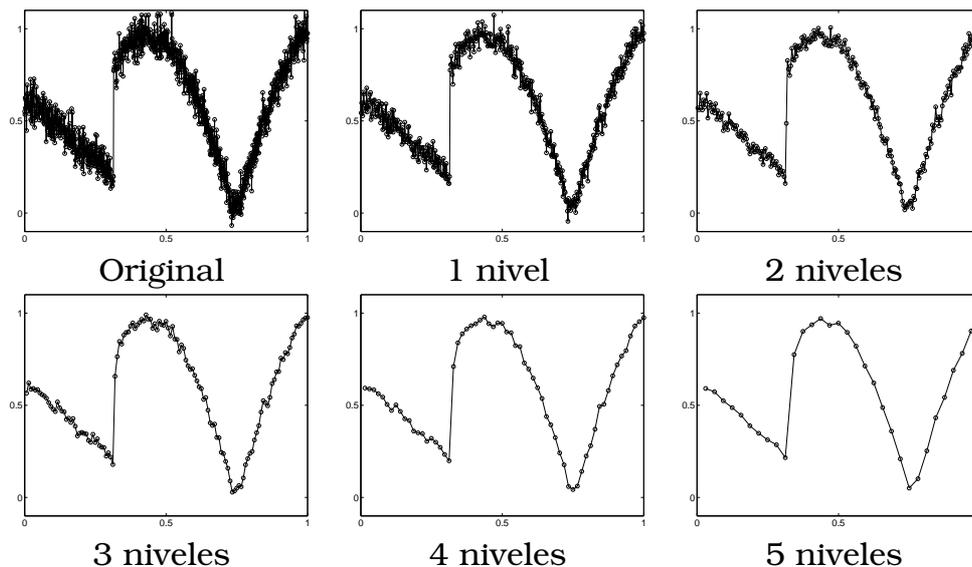


Figura 5.6: Decimación por medias en celda.

Por tanto no introducimos ninguna modificación al algoritmo, pero debemos utilizar un número de niveles suficiente para que la reducción de ruido permita la utilización correcta del esquema (unos 4 niveles es suficiente, aunque depende de la magnitud del ruido).

Reconstrucción interpolación-aproximación, $\overline{\mathcal{IAC}}$: en este caso sí podemos utilizar la medida E_2 . Aunque esto no es una gran ventaja, ya que si bajamos pocos niveles, por un lado la reducción del ruido es escasa y por otro la condición para que se cumpla la consistencia en presencia de ruido no favorece su eliminación. Así pues debemos bajar suficientes niveles y podemos utilizar E_2 para asegurar que se evita cruzar discontinuidades.

Reconstrucción por aproximaciones, $\overline{\mathcal{AC}}$: podemos hacer uso de la medida E_2 , aunque tenemos los mismos problemas que el caso anterior. Sin embargo podemos utilizar la modificación (2.51) que daba lugar a una reconstrucción no consistente. Otra opción es (2.52) que sí es consistente, pero también limita la reducción de ruido. Esto puede utilizarse en casos de ruido elevado en el que no podemos bajar suficientes niveles, aunque al ser no consistente la posición de los saltos al reconstruir no queda garantizada.

En definitiva, si no pretendemos utilizar un número suficiente (al menos 4) de niveles, es mejor decantarnos por los métodos vistos para una escala. Si a pesar de ello queremos bajar algún nivel podemos utilizar $\overline{\mathcal{AC}}$ con la modificación de no consistencia (2.51) y aplicando ENO mediante la medida E_2 .

Ejemplos

Trabajando con la función (1.70), tomamos 1024 puntos iniciales para que nos permita la utilización de suficientes niveles.

En un primer experimento introducimos ruido de magnitud $SNR = 25$ y bajamos 5 niveles por medias en celda, obteniendo la Figura 5.7. Las reconstrucciones utilizadas son $\overline{\mathcal{IC}}$ (a), $\overline{\mathcal{IAC}}$ (b), $\overline{\mathcal{AC}}$ (c) y $\overline{\mathcal{AC}}$ con la modificación (2.51) (d), todas ellas haciendo uso de la técnica SR . En los tres últimos sustituimos las diferencias divididas por la medida E_2 para decidir los *stencils* ENO . Los resul-

tados son satisfactorios, fruto de haber bajado 5 niveles. Al utilizar medias en celda, la función se ha suavizado lo suficiente como para que las diferencias divididas puedan ser utilizadas en la técnica *ENO*, como vemos en (a). Además los requisitos de consistencia no introducen variaciones significativas dado que las reconstrucciones obtenidas son suaves entre discontinuidades. Notar que en (d) el salto ha sido desplazado debido a la utilización de la reconstrucción no consistente.

En un segundo experimento nos planteamos qué ocurre si aumentamos el ruido a magnitud $SNR = 20$ y bajamos únicamente tres niveles. Se obtiene la Figura 5.8. Observamos que en (a) se aprecia el *fenómeno de Gibbs* al utilizarse *stencils* que cruzan el salto debido a que trabajamos con diferencias divididas. En (b) y (c) este problema queda solucionado al hacer uso de E_2 . Sin embargo la eliminación de ruido es escasa, incluso con el uso de la reconstrucción $\overline{\mathcal{AC}}$ debido a que con 3 niveles no se elimina suficiente ruido. Además el problema se agrava debido a la condición de consistencia, lo cual provoca que la reconstrucción se quiebre más. Al utilizar la reconstrucción $\overline{\mathcal{AC}}$ aproximando también en los nodos pares (2.51) obtenemos (d), donde se elimina más ruido, aunque aún se producen algunos picos indeseables fruto de la localización de falsas discontinuidades por parte de E_2 .

En la mayoría de casos las diferencias divididas pueden utilizarse sin problemas siempre que bajemos 4 o 5 niveles. Aún así E_2 da lugar a resultados más fiables que las diferencias divididas. En la Figura 5.9 trabajamos con la reconstrucción $\overline{\mathcal{IAC}}$, con 5 niveles y $SNR = 20$. A la derecha utilizamos diferencias divididas y a la izquierda E_2 , obteniendo una mejor aproximación en el salto en este último caso.

En vista de los resultados podemos concluir que este método elimina eficientemente ruido siempre que podamos bajar suficientes niveles. El número de niveles adecuado dependerá de la magnitud del ruido inicial. Además es preferible el uso de E_2 frente a diferencias divididas. Dicho cambio sólo puede utilizarse en las reconstrucciones $\overline{\mathcal{IAC}}$ y $\overline{\mathcal{AC}}$. Con la modificación de no consistencia para $\overline{\mathcal{AC}}$ no se obtienen ventajas claras, ya que si utilizamos suficientes niveles no se asegura una reconstrucción correcta de los saltos (y con $\overline{\mathcal{AC}}$ ya se obtienen buenos resultados) y además

con pocos niveles no se elimina eficientemente ruido, siendo preferible otros métodos como por ejemplo los vistos para una escala.

5.3.2

Multiscale Shrinkage

En los capítulos previos hemos utilizado las descomposiciones multiescala para eliminar ruido en señales discontinuas. Para me-

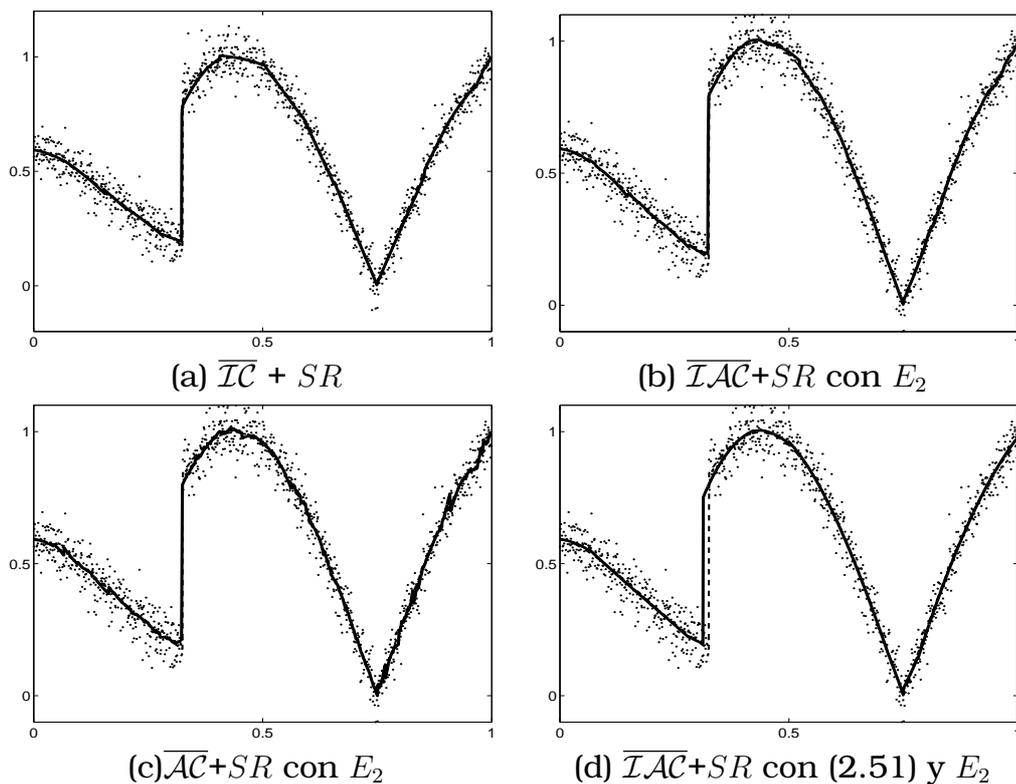


Figura 5.7: Decimamos 5 niveles mediante medias en celda y subimos otros 5 según cada reconstrucción, con $r = 3$ y $nl = nr = 3$ (excepto en (a)), para la función (1.70) con 1024 puntos y $SNR = 25$. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=3.71e-6$, $SNR=20.33$; (b) $ECM=3.57e-6$, $SNR=20.50$; (c) $ECM=3.49e-6$, $SNR=20.59$; (d) $7,24e-6$, $SNR=17.50$.

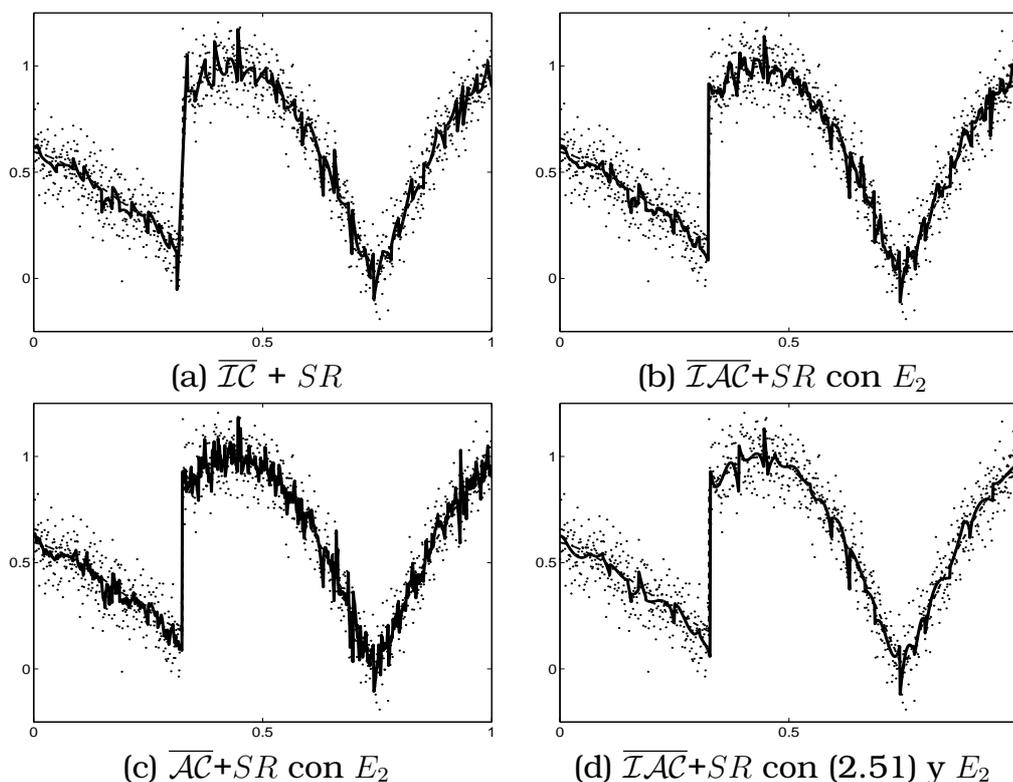


Figura 5.8: Decimamos 3 niveles mediante medias en celda y subimos 3 veces según cada reconstrucción, con $r = 3$ y $nl = nr = 3$ (excepto en (a)), para la función (1.70) con 1024 puntos y $SNR = 20$. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=9.42e-6$, $SNR=16.31$; (b) $ECM=9.04e-6$, $SNR=16.50$; (c) $ECM=1.02e-5$, $SNR=15.99$; (d) $SNR=1.12e-5$, $SNR=15.53$.

jorar la aproximación en las proximidades de las discontinuidades hemos aplicado las técnicas *ENO* y *SR*, evitando crear coeficientes de escala de magnitud elevada y poder así eliminarlos sin graves consecuencias. Trabajaremos un enfoque diferente, empleado en *wavelets*, basado en cortar los coeficientes de altas frecuencias que superen un cierto umbral. Dicho umbral deberá eliminar los coeficientes que provienen de ruido y mantener aquellos que surgen alrededor de las discontinuidades. Esto se conoce como *Wavelet Shrinkage Denoising*. En lo que sigue analizaremos estos

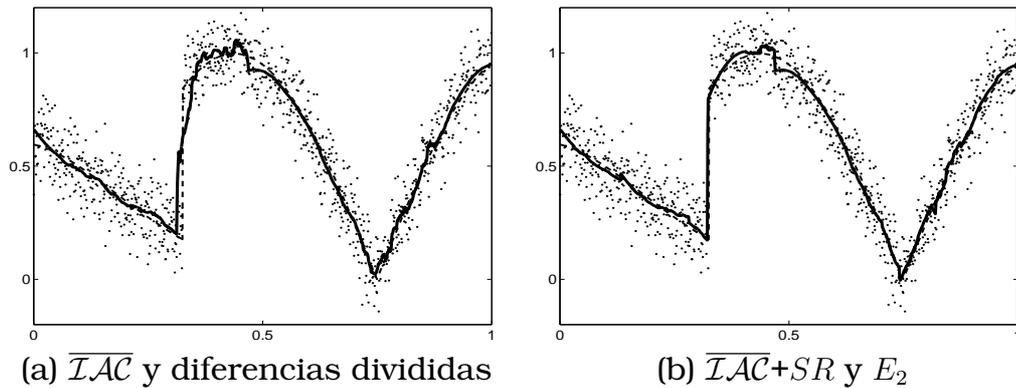


Figura 5.9: Decimamos 5 niveles mediante medias en celda y subimos 5 veces según cada reconstrucción, con $r = 3$ y $nl = nr = 3$, para la función (1.70) con 1024 puntos y $SNR = 20$. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=1.08e-5$, $SNR=15.70$; (b) $ECM= 9.98e-6$, $SNR=16.06$.

métodos y estudiaremos la viabilidad de adaptarlos a la multirresolución à la Harten.

Wavelet Shrinkage Denoising

Una de las técnicas comúnmente utilizadas para la eliminación de ruido en señales se basa en la utilización de las transformadas *wavelet*, [63], [70], [35], [23], [41]. Es por ello que no podemos obviarlas si de eliminar ruido se trata. Ciertos investigadores han afirmado que el método *wavelet shrinkage*⁴ ofrece "todo cuanto podemos desear de una técnica, desde generalidad hasta optimalidad" [36], aunque esta afirmación es sólo aceptable bajo la teoría asintótica, ya que en experimentos extraídos del mundo real debemos ser más cautos.

⁴*Wavelet shrinkage denoising* no debe ser confundido con suavizado (*smoothing*), aunque algunos autores utilicen *smoothing* como sinónimo de *denoising*. Mientras que *smoothing* elimina las altas frecuencias y mantiene las bajas, *denoising* intenta eliminar sólo el ruido que contamina la señal original, preservando sus frecuencias. Por ejemplo, al eliminar ruido en música, queremos mantener las altas (*treble*) y bajas (*bass*) frecuencias.

En 1984, Donoho y Johnstone establecieron el algoritmo *Wavelet Shrinkage Denoising* como una nueva herramienta de eliminación de ruido. Básicamente consiste en tres pasos: aplicación de una transformada *wavelet* directa, una eliminación no lineal de ruido y finalmente la aplicación de la transformada *wavelet* inversa correspondiente. Aunque las transformadas *wavelet* son métodos lineales, el segundo paso indica que el método es no lineal.

En términos de la transformada *wavelet* rápida, según las notaciones (1.77) y (1.77), si $W(\cdot)$ y $W^{-1}(\cdot)$ denotan los operadores *wavelet* directo e inverso y $T(\cdot, \lambda)$ es el operador de umbralizado, el proceso en el nivel j es el siguiente, (Figura 5.10):

1. Aplicamos la Transformada *Wavelet* Directa a los coeficientes de bajas frecuencias $W(\{\alpha_{j,i}\}_i) = (\{\alpha_{j-1,i}\}_i, \{\beta_{j-1,i}\}_i)$.
2. Aplicamos el Operador de Umbralizado (*shrink*) a los coeficientes de altas frecuencias $T(\{\beta_{j-1,i}\}_i, \lambda) = \{\hat{\beta}_{j-1,i}\}_i$.
3. Aplicamos la Transformada *Wavelet* Inversa con los coeficientes de altas frecuencias obtenidos: $W^{-1}(\{\alpha_{j-1,i}\}_i, \{\hat{\beta}_{j-1,i}\}_i) = \{\hat{\alpha}_{j,i}\}_i$.

Los coeficientes $\{\hat{\alpha}_{j,i}\}_i$ son una aproximación a $\{\alpha_{j,i}\}_i$ con un menor contenido en ruido.

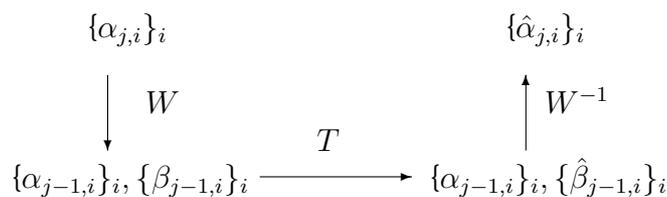


Figura 5.10: Un paso de *Wavelet Shrinkage Denoising*.

El operador de umbralizado $T(\cdot, \lambda)$ se define de tres maneras:

Lineal

$$T(U, \lambda) = 0 \quad \forall \lambda. \tag{5.7}$$

Hard-threshold

$$T(U, \lambda) = \begin{cases} U & \text{si } |U| > \lambda, \\ 0 & \text{en otro caso.} \end{cases} \tag{5.8}$$

Soft-threshold

$$T(U, \lambda) = \text{sgn}(U) \max(0, |U| - \lambda). \quad (5.9)$$

Hard-threshold en apariencia es más intuitivo, sin embargo la continuidad de *soft-threshold* tiene varias ventajas, ya que produce algoritmos mejor tratables matemáticamente. Además, los coeficiente de ruido puro en ocasiones pueden pasar el *hard-threshold* y aparecer como saltos en los resultados. Por estos motivos se utiliza mayoritariamente *soft-threshold* en *wavelet shrinkage*.

La elección de λ juega un papel determinante en el diseño del método. Existen numerosos trabajos para definir un λ apropiado, como *TAWS*, [71], *Firm Shrink*, [40], *Oracle Shrink*, [70], *non-negative Garrote*, [39]. También podemos encontrar diversas transformaciones, como *Ridgelets*, [24], [61], *Contourlets*, [57], [33], *Curvelets*, [67]. El analizar y comparar los diferentes métodos existentes sería una tarea excesivamente extensa y constituiría otro trabajo independiente por sí mismo. Lo que pretendemos es conocer las técnicas básicas de *Wavelets Shrinkage* y estudiar su posible aplicación a la multirresolución *à la Harten*. Es por ello que en este apartado nos centraremos en los tres métodos básicos y más utilizados de determinación de λ : *VisuShrink*, *SureShrink* y *BayeShrink*.

VisuShrink

Este método se describió en la sección 3.3.2. Recordemos que el umbral utilizado es $\lambda_U = \sigma \sqrt{2 \log(N)}$, siendo N el tamaño de la señal original y σ^2 la varianza del ruido, Suponemos que el ruido que hemos añadido es *blanco, gaussiano* y sigue una $N(0, \sigma^2)$. El valor de σ se estima mediante $\sigma \approx \frac{MAD}{0.6475}$ con *MAD* la mediana de las magnitudes de todos los coeficientes de altas frecuencias en la escala más fina.

El λ_U es óptimo en sentido asintótico [56], aunque si lo aplicamos a imágenes reales observamos un efecto de emborronado en los bordes lo cual nos indica que ciertos coeficientes que no provienen de ruido son eliminados (o suavizados). Cuantos más niveles se utilicen, más evidente es este efecto. En la práctica, dado que no trabajamos con tamaños de la señal que tienden a ∞ , podemos buscar mejores candidatos de λ .

SureShrink

SURE, *Stein's Unbiased Risk Estimator*, [51], fue desarrollado por Donoho y Johnstone en 1995 y se basa en calcular un umbral diferente para cada nivel. Si $\beta = \{\beta_i\}_{i=1,\dots,n}$ son los coeficientes de altas frecuencias del nivel de resolución en el que nos hallamos, el umbral *SURE* se calcula a partir del resultado de Stein [51], :

$$SURE(t; \beta) = n - 2 \cdot \#\{i : |\beta_i| < t\} + \sum_{i=1}^n \min(|\beta_i|, t)^2. \quad (5.10)$$

El estimador imparcial de riesgo de Stein, ofrece una estimación del riesgo para un valor particular del umbral λ . El umbral para *SURE*, λ_S , es el valor de t , menor que el corte universal, λ_U , que minimiza la anterior función:

$$\lambda_S = \operatorname{argmin}_{0 < t < \lambda_U} SURE(t; \beta). \quad (5.11)$$

Este problema de optimización es computacionalmente sencillo. Sin pérdida de generalidad, podemos ordenar $\{|\beta_i|\}_{i=1,\dots,n}$ en orden creciente. En los intervalos donde t se halla entre dos valores de $|\beta_i|$, $SURE(t)$ es estrictamente creciente. Con lo cual, el mínimo valor de λ_S es uno de los $|\beta_i|$. Como máximo hay n valores y λ_S se puede obtener mediante $O(n \log(n))$ operaciones.

NOTA 5.1. Algebraicamente, para la deducción de *SURE* se necesita que la función umbralizadora sea Lipschitz. Por tanto *SURE* sólo puede utilizarse bajo *soft-threshold*, [62].

BayeShrink

Mediante este método se realiza una estimación del valor óptimo que minimiza el Riesgo Bayesiano suponiendo una Distribución Gaussiana Generalizada, [70]. En este caso se utiliza un *threshold* diferente para cada nivel y está diseñado para ser utilizado con *soft-threshold*. Denotamos X, Y y V a las transformadas *wavelet* de la señal sin ruido, con ruido y de las componentes de ruido respectivamente. Podemos expresar que $Y = X + V$. Dado de

X y V son mutuamente independientes, las varianzas σ_x , σ_y y σ_v de X , Y , V cumplen: $\sigma_y^2 = \sigma_x^2 + \sigma_v^2$, donde v sigue una $N(0, \sigma_v^2)$.

En primer lugar debemos estimar la varianza del ruido mediante el estimador robusto $\hat{\sigma}_v = \frac{MAD}{0,6745}$. En segundo lugar estimamos la varianza de la señal con ruido: $\hat{\sigma}_y^2 = (\frac{1}{M} \sum_{i=1}^M \beta_{j,i}^2)^2$, donde $\{\beta_{j,i}\}_{i=1,\dots,M}$ son los coeficientes de altas frecuencias del nivel j .

El umbral que se utiliza es el siguiente, [70]: $\frac{\hat{\sigma}_v^2}{\hat{\sigma}_x}$, donde $\hat{\sigma}_x = \sqrt{\max(\hat{\sigma}_y^2 - \hat{\sigma}_v^2, 0)}$.

Notar que si $\hat{\sigma}_y^2 \leq \hat{\sigma}_v^2$ entonces $\lambda_B \rightarrow \infty$ y tendríamos problemas al calcularlo. En la práctica si esto ocurre, elegimos $\lambda^B = \max_i\{|\beta_{j,i}|\}$, con lo que todos los coeficientes se igualan a 0.

Resumiendo, el corte utilizado para *BayeShrink* es como sigue:

$$\lambda_B = \begin{cases} \frac{\hat{\sigma}_v^2}{\hat{\sigma}_x}, & \text{si } \sigma_y^2 > \sigma_v^2, \\ \max_i\{|\beta_{j,i}|\}, & \text{en otro caso.} \end{cases}$$

Ejemplos

Aplicaremos *VisuShrink*, *SureShrink* y *BayeShrink* a ejemplos concretos mediante la base *wavelet DB4*, que es adecuada para la eliminación de ruido, [53]. Como vimos en el capítulo 1 para aplicar la transformada *wavelet* y mantener la ortogonalidad de las matrices de coeficientes, podemos considerar que la función es periódica, estando formada por copias de sí misma. Así pues, las fronteras se considerarían como discontinuidades y al cortar los coeficientes de altas frecuencias podrían producirse malas aproximaciones en ellas. Dado que nos interesa comparar el *ECM* de estas soluciones con las obtenidas mediante multirresolución, es preferible eliminar dicho error para que no afecte al cómputo total del *ECM*. Una posible solución consiste en extender a izquierda y derecha los valores. Por ejemplo, para *DB4*, si los datos iniciales son $\{f_i\}_{i=1,\dots,n}$, pasamos a utilizar $\{f_1, f_1, f_1, f_2, \dots, f_{n-1}, f_n, f_n, f_n\}$. Esto lo realizamos en cada nivel, evitando la mala aproximación en las fronteras. Si la función no contuviese ruido, la solución más elegante sería utilizar en las fronteras la técnica *ENO-wavelet* vista en el capítulo anterior, pero las extrapolaciones de datos con ruido también pueden afectar al error.

En la Figura 5.11 hemos aplicado 5 niveles de *DB4* a una función con 1024 puntos iniciales a la que hemos añadimos ruido de magnitud $SNR = 25$. En (a) eliminamos todos los coeficientes de altas frecuencias. Hay una notable reducción de ruido pero el salto queda suavizado. En (b), (c) y (d) aplicamos *VisuShrink*, *SureShrink* y *BayeShrink* respectivamente, utilizando *soft-threshold*. En este caso la aproximación de los saltos mejora. Aunque *BayeShrink* consigue mejores resultados desde el punto de vista de la *ECM* y *SNR*, visualmente es mucho mejor *VISU* o *SURE*. Se debe a que estamos comparando la señal con ruido con la reconstrucción y no la señal sin ruido y la reconstrucción. En cualquier caso, aunque *ECM* y *SNR* son medidas objetivas, es bien conocido que para comprobar el resultado de la eliminación de ruido la impresión visual es también un dato a considerar. En este caso los mejores resultados se obtienen con *SureShrink*.

En la Figura 5.12 mostramos los coeficientes de altas frecuencias y los umbrales según cada método en un nivel concreto.

5.3.3

Multirresolución à la Harten con Shrinkage

En esta sección estudiaremos la viabilidad de adaptar las técnicas utilizadas en *Wavelet Shrinkage Denoising* para multirresolución à la Harten. Según las notaciones del capítulo 1, deberemos cortar los coeficientes de escala d_i^k de modo similar a los coeficientes de altas frecuencias, $\beta_{j,i}$.

Si los datos de entrada en el nivel k vienen representados por v^j , un paso del proceso se reduce a:

1. Mediante decimación (1.3) bajamos de nivel, es decir, $v^{k-1} = D_k^{k-1}v^k$ y $d^k = G_k Q_k v^k$, siendo los operadores D, G, Q los definidos en la sección 1.1.
2. Aplicamos el operador de umbralizado a los coeficientes de escala: $T(d^k, \lambda) = \hat{d}^k$

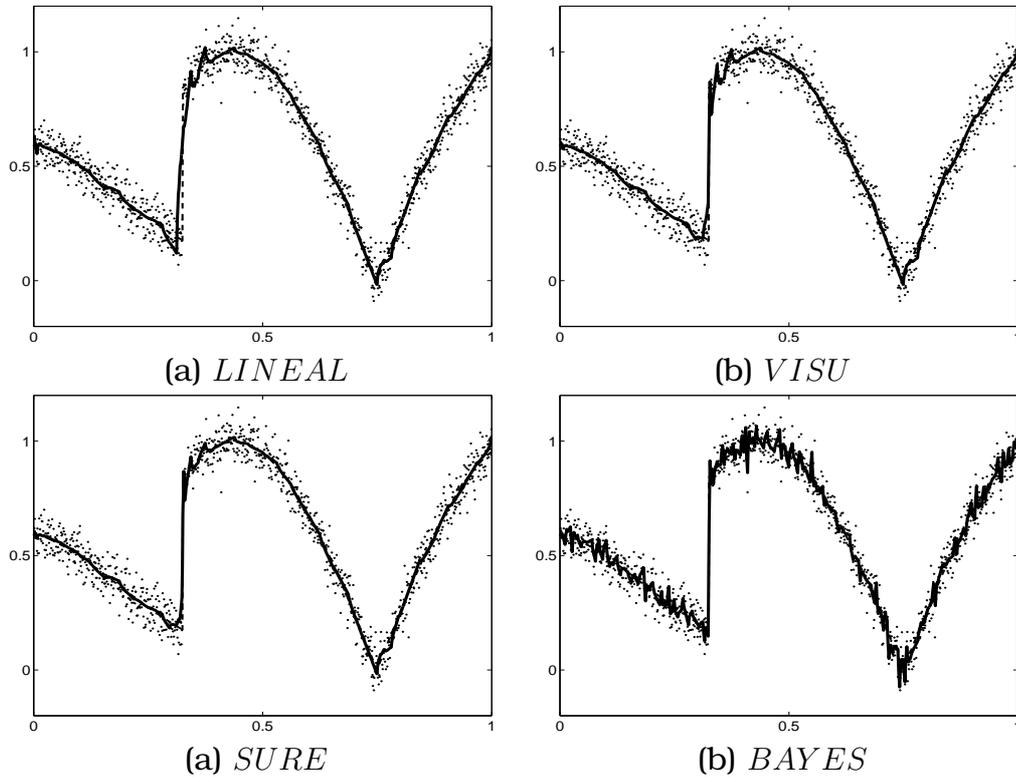


Figura 5.11: Tomando 1024 puntos de la función (1.70) con $SNR = 25$, aplicamos 5 niveles de la transformada DB4, donde los coeficientes de altas frecuencias se modifican mediante soft-threshold según cada umbral. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=4.37e-6$, $SNR=19.62$; (b) $ECM= 3.32e-6$, $SNR=20.81$; (c) $ECM=3.17e-6$, $SNR=21.02$; (d) $ECM= 2.41e-6$, $SNR=22.21$.

3. Finalmente aplicamos predicción con los nuevos coeficientes de escala: $P_{k-1}^k v^{k-1} + E_k \hat{d}^k = \hat{v}^k$.

El problema se reduce a determinar el λ adecuado en el operador $T(\cdot, \lambda)$. En este caso los λ 's utilizados en *wavelets* no son aplicables directamente, ya que los coeficientes *wavelet* de altas frecuencias y los coeficientes de escala son conceptos distintos.

Antes de buscar matemáticamente un umbral, lo hacemos experimentalmente. En base a los resultados de eliminación de ruido que obtengamos decidiremos si es provechoso el cálculo teórico de

dicho umbral. Al igual que en apartados anteriores trabajamos con medias en celda ya que al bajar niveles se elimina ruido. Para buscar el umbral óptimo, partiendo de la función (1.70) con 1024 puntos y $SNR = 25$, tomamos valores de λ de 0 a 0.5 en intervalos de 0.01 y representamos el ECM y el SNR entre la reconstrucción obtenida y la función sin ruido. Obtenemos la Figura 5.13, donde el umbral óptimo para este ejemplo concreto es $\lambda = 0,05$.

En la Figura 5.14 (a), mostramos el resultado de aplicar 5 niveles con reconstrucción interpolatoria y medias en celda utilizando *soft-threshold* con umbral 0,05. Observamos que no se produce fenómeno de Gibbs y por tanto los coeficientes de escala que provienen de *stencils* que cruzan el salto, han sido eliminados. Hay

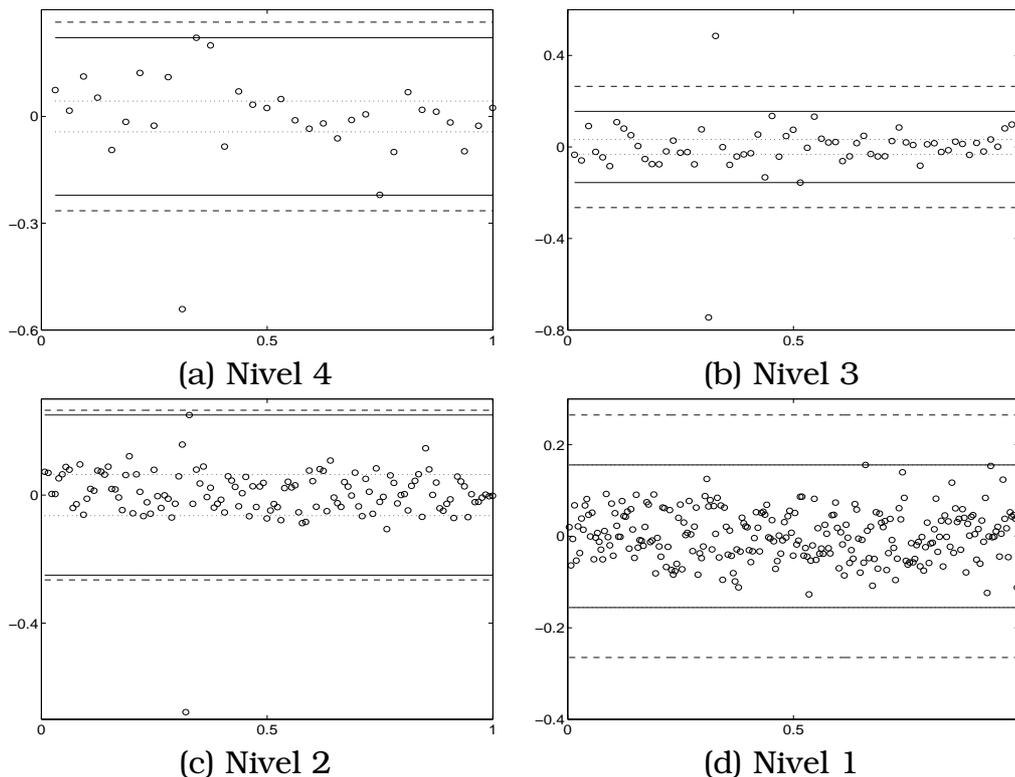


Figura 5.12: Tomamos 1024 valores de la función (1.70) con $SNR = 25$. Representamos los coeficientes de altas frecuencias ('o') y los umbrales λ_U (línea continua), λ_S (línea discontinua) y λ_B (línea punteada), según varios niveles. En el nivel 1 λ_U y λ_B coinciden.

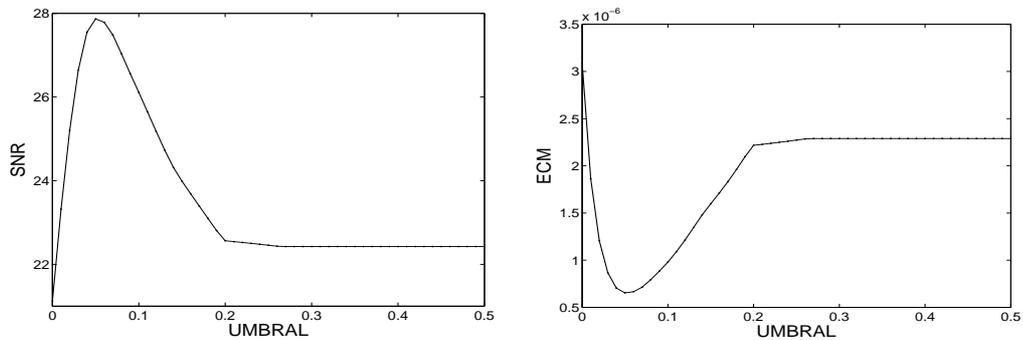


Figura 5.13: Aplicamos 5 niveles de multirresolución por medias en cada con reconstrucción interpolatoria y soft-threshold a la función (1.70) con 1024 puntos y $SNR = 25$. Izquierda: umbral vs SNR entre la reconstrucción obtenida y la función sin ruido. Derecha: umbral vs ECM.

eliminación de ruido, pero menor que con otros métodos vistos en apartados anteriores. Para mejorar los resultados podemos buscar un umbral para cada nivel, como se hace en *SURE*. Nuevamente, procediendo de forma experimental, hemos obtenido los umbrales $[0,01, 0,02, 0,04, 0,06, 0,1]$, siendo 0,1 el umbral para el nivel más grueso. Con dichos umbrales obtenemos la Figura 5.14 (b) donde la eliminación de ruido ha mejorado.

En la Figura 5.14 (c) y (d) hemos utilizado la reconstrucción $\overline{\mathcal{IAC}}$. En (c) el umbral es el mismo para todos los niveles, mientras que en (d) varía para cada nivel. Dichos umbrales se calculan experimentalmente y en este ejemplo concreto coinciden con el caso de reconstrucción interpolatoria.

En la Figura 5.15 hemos utilizado la reconstrucción $\overline{\mathcal{AC}}$ con un umbral para todos los niveles (a) y con umbrales adaptados (b). En este caso la eliminación de ruido es respecto de las otras reconstrucciones.

Los mejores resultados se obtienen para $\overline{\mathcal{IC}}$ y $\overline{\mathcal{IAC}}$, aunque no mejoran sustancialmente a la aplicación directa de las reconstrucciones eliminando todos coeficientes de escala (umbral lineal) y aplicando las técnicas *ENO-SR*. Una posible ventaja podría ser si se observase una mejora al aplicar menos niveles, ya que con los métodos anteriores necesitamos bajar un suficiente número de niveles. Sin embargo, en la Figura 5.15 (c) y (d) hemos repetido algu-

nos experimentos de este apartado con 3 niveles, obteniendo una escasa eliminación de ruido. Por tanto desestimamos la búsqueda teórica de umbrales óptimos para los coeficientes de escala ya que la eliminación de ruido es similar a otros métodos ya estudiados, aunque con los experimentos expuestos hemos comprobado la viabilidad de la adaptación de las ideas de *Wavelets Shrinkage Denoising* a descomposiciones multiescala.

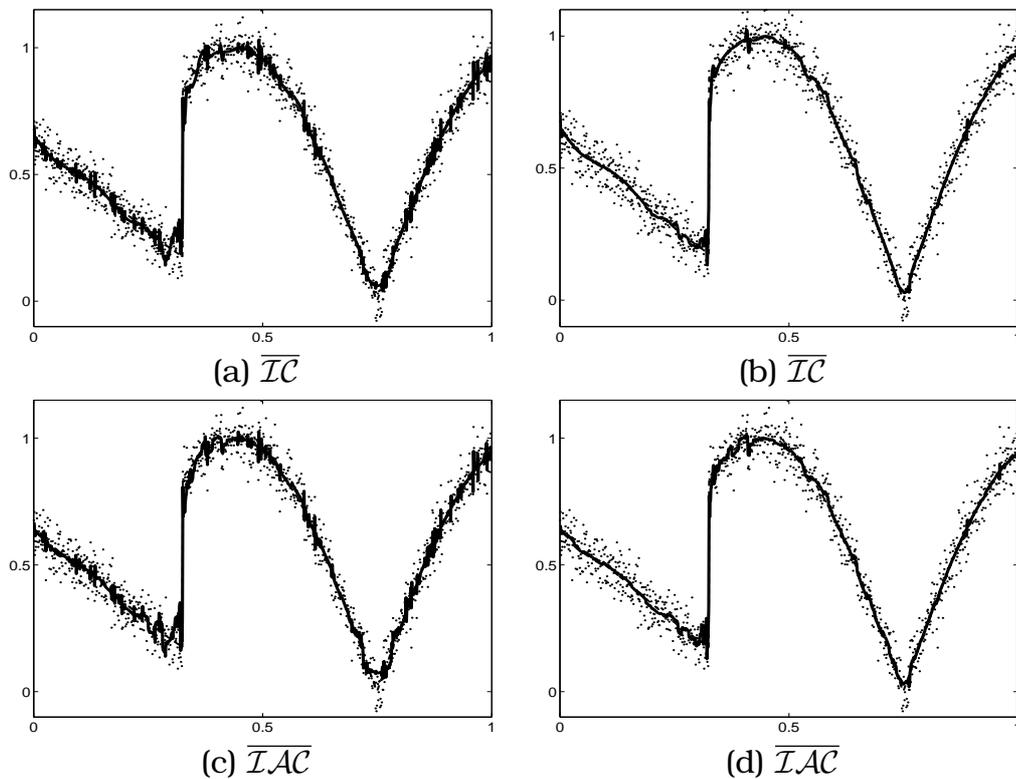


Figura 5.14: Partimos de (1.70) con 1024 puntos y $SNR = 25$. Aplicamos cinco niveles de multiresolución por medias en cada con las reconstrucciones indicadas y *soft-threshold*. En (a) y (c) usamos el umbral $\lambda = 0,05$ para todos los niveles. En (b) y (d) utilizamos un umbral diferente para cada nivel $\lambda_0 = 0,01$, $\lambda_1 = 0,02$, $\lambda_2 = 0,04$, $\lambda_3 = 0,06$, $\lambda_4 = 0,1$. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=2.79e-6$, $SNR=21.55$; (b) $ECM=3.01e-6$, $SNR= 21.22$; (c) $ECM=2.87e-6$, $SNR=21.43$; (d) $ECM=3.02e-6$, $SNR=21.24$.

5.3.4

ENO-wavelets

Para aplicar *ENO-DB2p-TI* a una función con ruido, debemos introducir las siguientes modificaciones:

- Aplicar un detector de discontinuidades válido para funciones

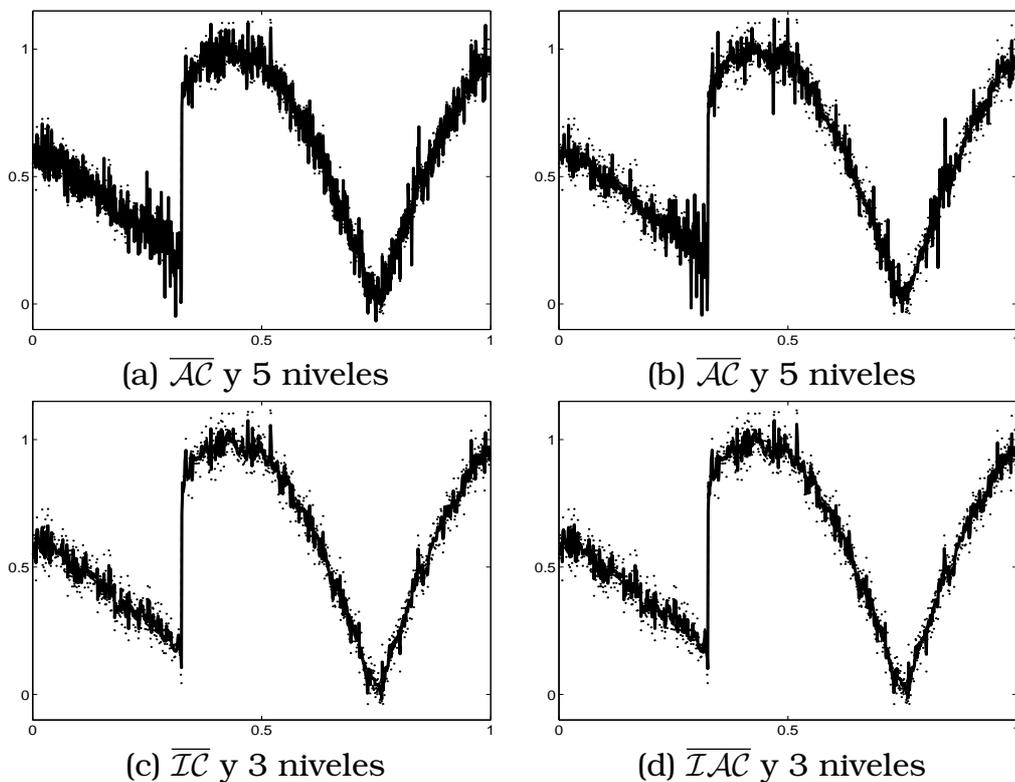


Figura 5.15: Tomando 1024 puntos de (1.70) con $SNR = 25$, decimamos 5 niveles por medias en cada y subimos con las reconstrucciones indicadas con *soft-threshold*. En (a) usamos $\lambda = 0,01$ para todos los niveles. En (b) utilizamos un umbral diferente para cada nivel: $\lambda_0 = 0,01$, $\lambda_1 = 0,01$, $\lambda_2 = 0,03$, $\lambda_3 = 0,04$, $\lambda_4 = 0,07$; al igual que en (c) y (d). La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. Errores: (a) $ECM=1.20e-6$, $SNR=25.28$; (b) $ECM=3.26e-6$, $SNR=20.92$; (c) $ECM=1.09e-6$, $SNR=25.68$; (d) $ECM=1.08e-6$, $SNR=25.70$.

con ruido, como por ejemplo $\overline{AP} - SC$, $E_1 \& E_2$ o $VISU$.

- Realizar las extrapolaciones mediante polinomios de mínimos cuadrados, como $q^{\overline{AP}}$ o de interpolación aproximación, $q^{\overline{IAP}}$.

$ENO-DB2p-TI$ con las anteriores modificaciones lo denotaremos $ENO-DB2p-TI-N$. Siempre que la detección de discontinuidades sea correcta, no se van a crear coeficientes de altas frecuencias de magnitud elevada. Así pues, no es necesario ningún procesado posterior de los coeficientes de altas frecuencias, como *soft* o *hard threshold*, y la eliminación total de los coeficientes de altas frecuencias dará como resultado la eliminación del ruido.

Veamos algún ejemplo. En la Figura 5.16 hemos aplicado el método $ENO-DB2p-TI-N$ con 5 niveles a funciones con uno, (1.44), y dos, (3.15), saltos. Hemos utilizado dos detectores de discontinuidades: $\overline{AP} - SC$ y $VISU$, vistos en la sección 3.3. Para $p = 2$ la extrapolación se ha realizado con 4 nodos y grado 1, y para $p = 3$ con 5 nodos y grado 2. Si observamos las Figuras (a), (b), (c) y (d), los mejores resultados se obtienen para $DB4$ con detector $VISU$. En ese caso no se produce fenómeno de Gibbs y la reconstrucción es muy próxima a lo buscado. Sin embargo, este resultado no es extrapolable en general. Observemos la Figura (e), donde la discontinuidad derecha se ha prolongado en exceso debido a las extrapolaciones de grado 1, perdiendo la curvatura de la señal original. La solución no consiste en aumentar el grado de la extrapolación, como en (f), ya que a medida que aumentamos el grado las extrapolaciones también distan más de lo esperado. Si aplicamos pocos niveles las soluciones son buenas, pero la eliminación de ruido es escasa. Los pequeños errores de extrapolación a partir de datos con ruido se van acumulando al subir de niveles lo cual provoca pérdida de aproximación.

El método necesita localizar con exactitud la posición de los saltos y los detectores que manejamos no son excesivamente fiables. Además, para $DB6$, si analizamos los coeficientes de altas frecuencias podemos encontrarnos que hay un sólo coeficiente de magnitud elevada, aunque teóricamente las únicas opciones serían 2 o 3 coeficientes consecutivos afectados, dificultando de este modo la localización exacta del salto.

Por estos motivos, concluimos que la aplicación de este método, tal como se ha diseñado, es poco fiable para la eliminación de ruido, debiéndose mejorar el modo de realizar las extrapolaciones y las detecciones para un uso práctico.

5.4

Comparativa

En esta sección aplicaremos a una misma función los métodos en los que hemos obtenido mejores resultados en cuanto a eliminación de ruido según cada apartado⁵. Esto no significará que el método sea el mejor y debemos desechar el resto, ya que hay situaciones en que debemos utilizar uno en concreto. Por ejemplo si la función de partida tiene 128 puntos, los métodos por multirresolución son inviables, ya que no podremos bajar suficientes niveles. En cualquier caso para realizar esta comparativa partiremos de una función con 1024 puntos a la que añadimos ruido de magnitud $SNR = 25$. Los métodos que compararemos son:

- M1** AMC entre discontinuidades, con localizador de discontinuidades $\overline{AP} - SC$.
- M2** \overline{AP} punto a punto con *stencil* centrado excepto en las discontinuidades (localizadas con $\overline{AP} - SC$).
- M3** 10 iteraciones por sustitución de valores medios a izquierda y derecha.
- M4** 5 niveles de \overline{IAC} con *SR*, evitando cruzar discontinuidades con E_2 .
- M5** 5 niveles de *SureShrink* con base *DB4*.
- M6** 5 niveles de \overline{IAC} , con *soft-threshold* y umbral adaptado a cada nivel (calculado de forma experimental).

⁵Desde <http://www.uv.es/animations/noguera/> podemos descargar una GUI en MATLAB® que permite realizar experimentos de eliminación de ruido según los datos introducidos por el usuario.

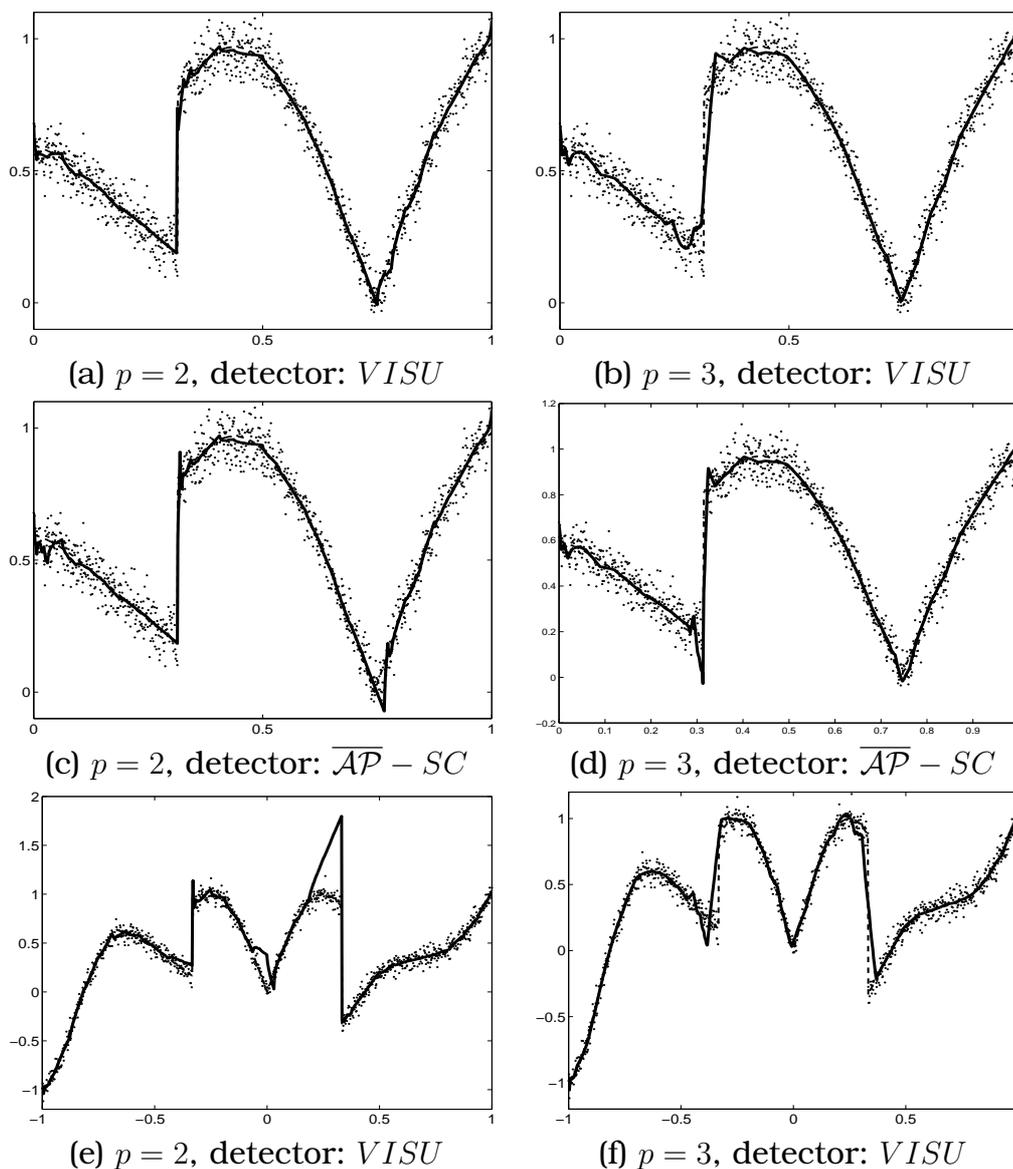


Figura 5.16: Aplicamos 5 niveles de ENO-DBp-TI-N con los detectores y p que se indican, a las funciones (1.44) ((a), (b), (c), (d)) y (3.15) ((e), (f)) con 1024 puntos y $SNR = 25$. Se realiza una aproximación lineal. La función original sin ruido se representa mediante línea discontinua, con ruido mediante puntos y la reconstrucción mediante línea continua. Errores: (a) $ECM=3.45e-6$, $SNR=20.60$; (b) $ECM=3.87e-6$, $SNR=20.10$; (c) $ECM=3.09e-6$, $SNR=21.09$; (d) $ECM=3.62e-6$, $SNR=20.36$; (e) $ECM=2.44e-5$, $SNR=12.49$; (f) $ECM=1.09e-5$, $SNR=15.02$.

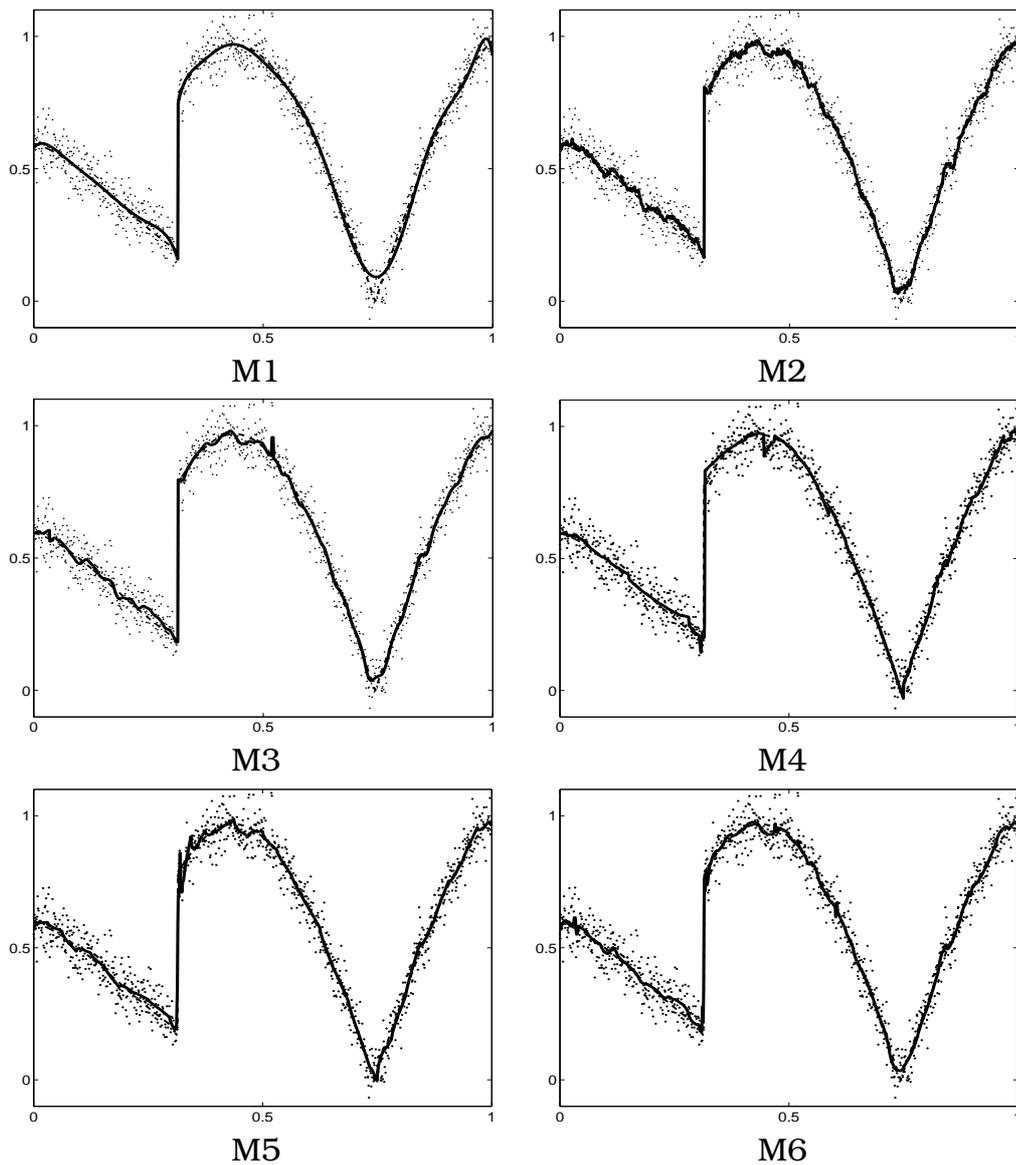


Figura 5.17: Aplicación de diferentes métodos de eliminación de ruido especificados en la sección 5.4 aplicados a la función (1.70) con 1024 puntos y ruido de magnitud $SNR = 25$. La línea discontinua representa la función sin ruido, los puntos son la función con ruido añadido y la línea continua es la reconstrucción obtenida. La Tabla 5.1 muestra el ECM y SNR de cada método.

| | Entre \tilde{f} y \hat{f} | | Entre f y \hat{f} | |
|----|-------------------------------|------------|-----------------------|------------|
| | <i>ECM</i> | <i>SNR</i> | <i>ECM</i> | <i>SNR</i> |
| M1 | 3.2499e-6 | 20.1955 | 3.4076e-7 | 30.6755 |
| M2 | 2.8282e-6 | 21.5024 | 2.7289e-7 | 31.6401 |
| M3 | 2.7100e-6 | 21.6829 | 2.1780e-7 | 32.5345 |
| M4 | 3.6203e-6 | 20.4396 | 8.2422e-7 | 26.8396 |
| M5 | 3.0380e-6 | 21.1859 | 3.3788e-7 | 30.7124 |
| M6 | 2.8344e-6 | 21.4930 | 2.8918e-7 | 31.3884 |

Tabla 5.1: *SNR* y *EMC* entre la función original sin ruido (f) y la reconstrucción (\hat{f}); y entre la función original con ruido (\tilde{f}) y la reconstrucción, para los experimentos de la Figura 5.17.

La Figura 5.17 muestra las reconstrucciones obtenidas. Los errores *ECM* y *SNR* son de orden similar en todos los casos, Tabla 5.1, aunque hay diferencias notables para decidir la aplicación de uno u otro método:

AMC entre discontinuidades. Lo fundamental del método es la localización de los saltos. Si esto no se cumple la solución obtenida es inaceptable. Por tanto evitaremos su aplicación en señales donde el método de detección pueda fallar, por ejemplo si el ruido es elevado, si tenemos gran número de discontinuidades y no están lo suficientemente separadas,...

\overline{AP} **punto a punto**, con *stencil* centrado excepto en las discontinuidades (localizadas con $\overline{AP} - SC$). El método también está sujeto a una localización previa de saltos. Sin embargo un fallo no va a provocar una pérdida total de las características de la función, sino un suavizado del salto. Para evitarlo se podría elegir el *stencil* según E_1 o E_2 , pero como vimos se eliminaba poco ruido. El método es aplicable con escasos nodos iniciales, siendo la única restricción que los saltos estén suficientemente separados, como mínimo que entre dos discontinuidades haya un número de nodos igual al tamaño del *stencil* usado (preferiblemente mayor que 20 ya que para menos nodos no se elimina ruido eficientemente).

Iteración por sustitución de de valores medios a izquierda y derecha. El método también contiene un localizador de posibles discontinuidades, pero si falla sólo provoca un suavizado del salto. La mayor ventaja de este método reside en su bajo coste computacional (es de orden lineal en cada iteración) y que las discontinuidades sólo necesitan estar separadas por 1 punto.

\overline{IAC} con *SR*, evitando cruzar discontinuidades con E_2 . No se necesita el uso de un localizador de discontinuidades. Es un método que proporciona buenos resultados, aunque tiene dos limitaciones. La primera es que las discontinuidades deben estar suficientemente separadas para aplicar *ENO* con E_2 en todos los niveles y la segunda es que partamos de un número suficiente de datos como para poder aplicar un mínimo de 4 niveles de multi-resolución.

SureShrink con base DB4. Es un método fiable, y su única restricción es que dispongamos de un tamaño de datos que permita utilizar suficientes niveles (mínimo 4) y que se cumpla la *DSP*.

\overline{IAC} con *soft-threshold* y umbral adaptado a cada nivel. Es similar a *SURE* con sus mismas ventajas e inconvenientes. No hemos calculado de forma teórica un umbral, sino de forma experimental con lo cual los resultados que mostramos corresponden a la mejor solución posible para el ejemplo expuesto. Para conseguir un método aplicable en la práctica se debería calcular el umbral de forma teórica ya que de forma experimental el coste computacional se dispara.

ENO-wavelets con detector *VISU* para cada nivel. Es un método poco fiable en presencia de ruido y por ello no ha sido incluido en la comparativa. Se consiguen unos errores de aproximación similares a los anteriores pero el *fenómeno de Gibbs* aparece con frecuencia. Si se solucionasen estos problemas sería un buen método para eliminar ruido y comprimir de forma simultánea.

Con estas ideas y de forma simplificada, en la Figura 5.18 mostramos un diagrama que ayuda a decidir entre uno u otro método.

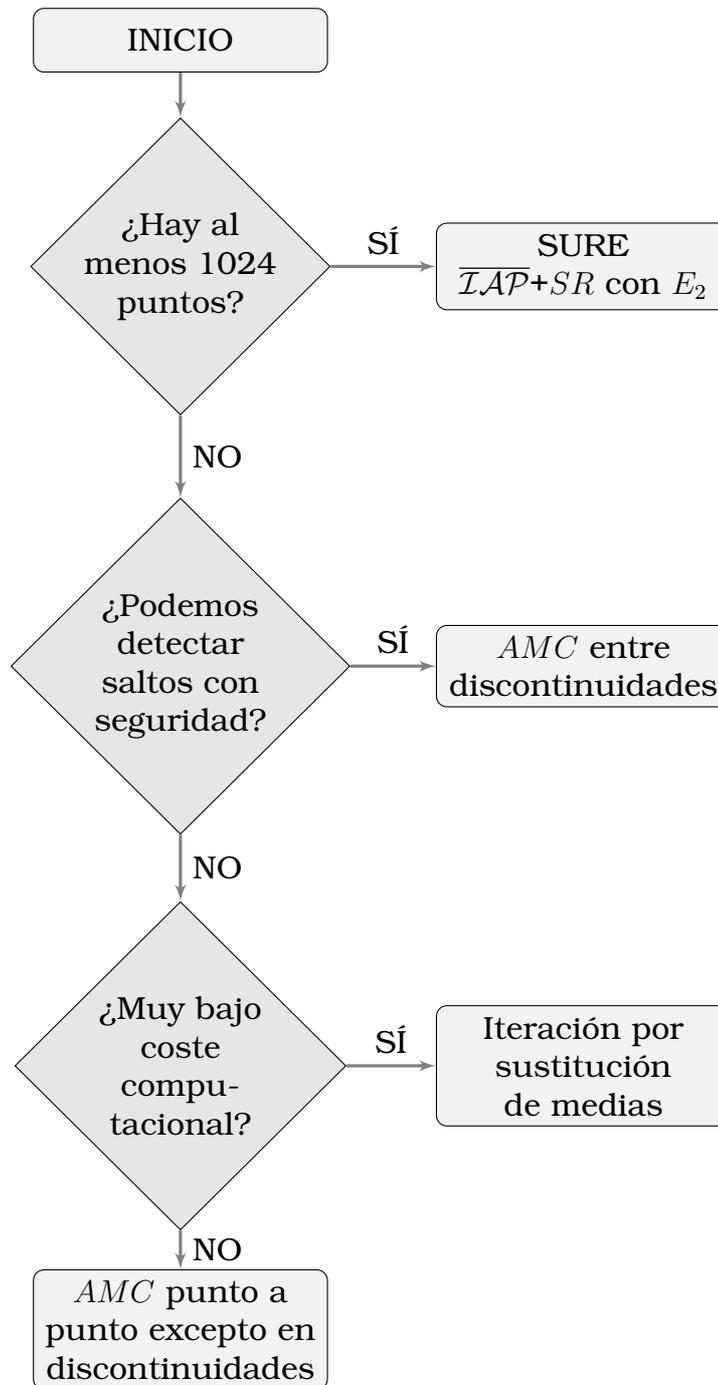


Figura 5.18: Indicaciones para la elección de un método de eliminación de ruido.

5.5

Conclusiones

En este capítulo hemos aplicado las herramientas matemáticas propuestas en capítulos anteriores a la eliminación de ruido. Hemos dividido este estudio en dos grandes bloques. El primero de ellos se ha basado en el estudio en una escala. Cuando la función de partida tiene pocos nodos ésta será nuestra única opción. Se han estudiado tres maneras de proceder. La primera se basa en una idea sencilla expuesta en [60], donde se sustituye cada trozo que proviene de una función suave por su aproximación por mínimos cuadrados. Los resultados son satisfactorios siempre que la localización de saltos no falle, en cuyo caso el método no es aplicable ya que los resultados distan en exceso de la función original. En segundo lugar se ha estudiado la idea de sustituir cada punto por su aproximación por mínimos cuadrados bajo un determinado *stencil*. Tras el estudio realizado se concluye que con estas ideas la mejor opción es utilizar *stencils* centrados excepto en las discontinuidades, que nuevamente deberán ser previamente detectadas. Para finalizar este bloque se propone un algoritmo iterativo extremadamente simple, pero que consigue una eliminación de ruido aceptable respetando las discontinuidades. Este algoritmo será una buena opción cuando nos interese un bajo coste computacional.

En un segundo bloque hacemos uso de las descomposiciones multiescala. La multirresolución *à la Harten* ha sido comúnmente aplicada a la compresión de señales pero en escasas ocasiones a la reducción de ruido. Hemos estudiado este tema adaptando las técnicas *ENO-SR* para mejorar la eficacia de los algoritmos en presencia de ruido. Trabajando con medias en celda y disponiendo de datos como para aplicar suficientes niveles, la reducción de ruido es satisfactoria, siendo la mejor opción la reconstrucción por *interpolación aproximación*, ya que permite una selección del *stencil* mediante la medida E_2 (en presencia de ruido más fiable que diferencias divididas) y proporciona una mayor reducción de ruido que la reconstrucción por *aproximación*.

Dado que la técnica más utilizada para la eliminación de ruido

es *Wavelet Shrinkage Denoising*, se han estudiado sus métodos básicos para compararlos con los anteriores y se ha comprobado la viabilidad de aplicación de estas ideas a las descomposiciones multiescala que manejamos en este trabajo.

Siguiendo la transformada *wavelet* se han introducido cambios en el esquema *ENO-wavelet* para poder adaptarlos a la eliminación de ruido. Su principal ventaja es que no es necesario un postproceso en los coeficientes de altas frecuencias y su punto débil reside en que los pequeños errores en detección de discontinuidades o en las extrapolaciones de datos con ruido se propagan y aumentan, disminuyendo la fiabilidad del método.

En definitiva, los esquemas de multirresolución *à la Harten*, bajo simples adaptaciones, son aplicables a la eliminación de ruido de funciones discontinuas compuestas por trozos suaves contaminadas por ruido *blanco gaussiano*, obteniendo resultados tan buenos como los que se utilizan normalmente en este campo, basados en *wavelets*.

6

Conclusiones generales y perspectivas

En esta tesis se ha trabajado en dos líneas de investigación, la multirresolución *à la Harten* y los esquemas *wavelet*, aunque ambas están íntimamente relacionadas, ya que lo primero puede considerarse una generalización de un caso particular de lo segundo (ver [44, 45, 46]).

En el primer capítulo hemos revisado estas teorías, aplicando diferentes métodos no lineales a las descomposiciones por multirresolución *à la Harten* como *ENO*, *SR*, *WENO* y *PPH*, tanto para esquemas por valores puntuales como por medias en celda. En la literatura sobre este tema es usual el cálculo de filtros a partir de la forma de Newton de los polinomios interpoladores, sin embargo, en este trabajo se ha realizado mediante soluciones de sistemas matriciales, lo cual permite el cálculo de filtros de las diferentes reconstrucciones (y de las técnicas *ENO-SR*) como fruto de simples

productos matriciales.

Generalmente, los esquemas de multirresolución *à la Harten* han sido aplicados mediante reconstrucciones interpolatorias. En el segundo capítulo se ha trabajado con reconstrucciones no exclusivamente interpolatorias. En esta línea se ha diseñado una nueva reconstrucción para valores puntuales consistente en la interpolación en los extremos del intervalo de trabajo y la aproximación en una serie de nodos adyacentes, denominándola reconstrucción *interpolación aproximación*. Se ha estudiado la consistencia, los órdenes, las funciones límite, la generación de curvas y la posibilidad de aplicación de las técnicas *ENO*, *SR*, *WENO* y *PPH*, así como sus adaptaciones para medias en celda. Finalmente se ha introducido la reconstrucción por mínimos cuadrados en los esquemas por multirresolución *à la Harten* tanto para valores puntuales como para medias en celda. Debido a la falta de condiciones interpolatorias se pierde la conexión entre dos niveles consecutivos, haciendo inviable el uso de algunas técnicas no lineales. Se ha estudiado con detalle cada caso y propuesto alternativas que, aunque se desvíen de la técnica estándar, proporcionan mejoras significativas en las reconstrucciones.

En un tercer capítulo, y con vistas a la aplicación de la eliminación de ruido, se ha tratado el tema de la detección de discontinuidades compuestas por trozos suaves para funciones con y sin ruido, haciendo uso de las herramientas expuestas en los capítulos anteriores. Para funciones sin ruido destacamos el diseño de un detector basado en la combinación de las ideas *ENO* para valores puntuales y medias en celda capaz de discriminar entre saltos, esquinas interiores a intervalos y esquinas en puntos del *grid*. En cuanto a funciones con ruido hemos estudiado algunos métodos ya publicados y hemos propuesto un nuevo algoritmo que mejora la detección en presencia de niveles elevados de ruido.

En el cuarto capítulo se ha estudiado el método *ENO-wavelet*, [26]. Este método conecta varias de las ideas trabajadas en capítulos anteriores, ya que aplica las ideas *ENO* de las reconstrucciones interpolatorias a la transformada *wavelet* haciendo uso de la detección de discontinuidades de funciones discontinuas compuestas por trozos suaves. Se ha analizado su funcionamiento y se ha propuesto una nueva forma de calcular los coeficientes mediante

la transformada inversa, dando lugar a los métodos que hemos denominado *ENO-wavelet-TI*. Con dicha modificación se asegura la invertibilidad del método, se reducen el número de datos extrapolados y lo más importante: se generaliza de forma sencilla el método para la base $DB2p$, siendo p cualquier entero.

En el último capítulo se han aplicado todas las reconstrucciones a la eliminación de ruido de funciones discontinuas compuestas por trozos suaves con ruido aditivo. Se ha trabajado en una escala y en varias escalas, estudiando métodos ya publicados y proponiendo nuevos. Los resultados obtenidos son similares a los existentes en la literatura, como por ejemplo *Wavelet Shrinkage Denoising*. Hemos realizado un estudio comparativo entre ellos y se han dado indicaciones de cuál es más adecuado aplicar en función de los datos iniciales y los objetivos buscados.

Aportaciones

Las contribuciones de esta tesis pueden resumirse en:

- Cálculo detallado de filtros mediante productos matriciales para las reconstrucciones interpolatorias estándar y con las técnicas *ENO-SR*, tanto para valores puntuales, \overline{IP} , como para medias en celda, \overline{IC} . Elaboración de una Interfaz Gráfica de Usuario (GUI) para MATLAB® que calcula dichos filtros para diferentes grados¹.
- Diseño de la reconstrucción *interpolación aproximación* tanto para valores puntuales, \overline{IAP} , como para medias en celda, \overline{IAC} , en el contexto de multirresolución *à la Harten*. Se demuestra que la reconstrucción es consistente de orden $r + 1$ para valores puntuales y de orden r para medias en celda, donde r representa el grado del polinomio de *interpolación aproximación*. Se pueden aplicar las técnicas *ENO-SR* de forma satisfactoria. *WENO* y *PPH* sólo son aplicables para ciertos grados y longitudes de *stencils*. En todos los casos se expone un cálculo detallado de filtros y se proporcionan algunas

¹Disponible en <http://www.uv.es/animations/noguera/>

GUI's en MATLAB® para calcularlos según diferentes grados y tamaños de *stencil*. Se estudian las funciones límite y la generación de curvas, obteniendo resultados satisfactorios para $\overline{\mathcal{IAP}}$ pero no para $\overline{\mathcal{IAC}}$ ya que las soluciones son de escasa suavidad.

- Adaptación de la reconstrucción por mínimos cuadrados al contexto de mutirresolución *à la Harten* para valores puntuales, $\overline{\mathcal{AP}}$, y media en celda, $\overline{\mathcal{AC}}$, siendo una reconstrucción de orden $r+1$ para valores puntuales y de orden r para medias en celda. La imposición de consistencia empeora los resultados en cuanto a aproximación y se proponen alternativas, (2.42) y (2.51), a costa de perder consistencia. *ENO* y *SR* se pueden introducir bajo ciertas modificaciones, pero *WENO* y *PPH* no son aplicables para esta reconstrucción (excepto un caso en que los filtros coinciden con los de $\overline{\mathcal{IAP}}$). Se expone un cálculo detallado de filtros y se proporcionan algunas GUI's en MATLAB® para calcularlos según diferentes grados y tamaños de *stencil*. En cuanto a funciones límite y generación de curvas, tanto para $\overline{\mathcal{AP}}$ como para $\overline{\mathcal{AC}}$ se obtienen soluciones muy quebradas, a menos que eliminemos la condición de consistencia.
- Implementación del esquema *LOCALIZA-ENO-SR-VP-MC*, Algoritmo 3.2, para la localización de discontinuidades en funciones discontinuas compuestas por tozos suaves, capaz de discriminar entre saltos, esquinas en puntos del *grid* y esquinas interiores a un intervalo.
- Se propone una mejora del algoritmo presentado en [26] para la localización de discontinuidades basado en magnitud de coeficientes *wavelet*, que evita fallos de detección al trabajar con rectas, dando lugar al Lema 3.2.
- Diseño del método de localización de saltos $\overline{\mathcal{AP}} - SC$ para funciones con ruido, visto en la sección 3.3.3, más eficaz que el resto de métodos analizados (combinación E_1-E_2 , [60], y el basado en umbral *VISU*) en presencia de una magnitud elevada de ruido.

- Diseño del método *ENO-wavelet-TI* donde hacemos uso de la transformada inversa para el cálculo de los coeficientes de altas y bajas frecuencias construidos a partir de *stencils* que crucen discontinuidades. Demostramos que *ENO-DB2p-TI* es invertible y equivale a *ENO-DB2p* para $p = 2, 3$. Se consigue generalizar el método de manera sencilla, presentando el algoritmo *ENO-DB2p-TI*, con p natural.
- Comparativa entre métodos de detección de discontinuidades y realización de una GUI en MATLAB® donde el usuario puede realizar experimentos de detección con los métodos estudiados en esta tesis.
- Diseño de un método de reducción de ruido en una escala utilizando la reconstrucción \overline{AP} con *stencils* centrados donde no se crucen saltos y no centrados donde afecten discontinuidades, Algoritmo 5.2.
- Diseño de un método iterativo de eliminación de ruido en una escala que respeta los saltos de una magnitud preestablecida, Algoritmo 5.5. Su principal virtud es el bajo coste computacional.
- Aplicación de la reconstrucción \overline{TC} con *ENO-SR* en varios niveles, para la eliminación de ruido. Los resultados son satisfactorios siempre que dispongamos de suficientes datos iniciales como para que las discontinuidades estén lo suficientemente separadas para aplicar un número elevado de niveles (alrededor de 5) para que la reducción de ruido sea tal que permita el uso de diferencias divididas en la elección *ENO*.
- Aplicación de la reconstrucción \overline{TAC} con *ENO-SR* y elección *ENO* mediante la medida E_2 . Al igual que en el punto anterior debemos disponer de suficientes nodos iniciales, aunque en este caso los resultados son, en general, mejores.
- Aplicación de la reconstrucción \overline{AC} con *ENO-SR* y elección *ENO* mediante la medida E_2 . Para que los resultados se aproximen a lo deseado necesitamos incluir la modificación (2.51) de no consistencia, lo cual provoca posibles desplazamientos de los saltos. Es por tanto un método peor que los anteriores.

- Aplicación de *ENO-DB2p-TI* a la eliminación de ruido haciendo uso de extrapolaciones por mínimos cuadrados. Su principal virtud es que en un mismo algoritmo se consigue comprimir y eliminar ruido al realizar la aproximación lineal, sin postproceso de coeficientes de altas frecuencias. Fruto de las extrapolaciones, se introducen pequeñas perturbaciones que se propagan a niveles superiores y provocan malas localizaciones, apareciendo con frecuencia el *fenómeno de Gibbs*
- Comparativa entre los métodos de eliminación de ruido estudiados, concluyendo que no existe un claro vencedor, si no que la aplicación de uno u otro va a depender de los datos iniciales y de nuestros objetivos. Hemos realizado un diagrama de flujo con indicaciones para decidir cuál aplicar en cada caso, además de una GUI en MATLAB® donde el usuario puede modificar los parámetros y realizar experimentos de eliminación de ruido de los principales métodos vistos en esta tesis.

Perspectivas

Como futuras líneas de investigación proponemos:

- Diseño de reconstrucciones que interpolen en un conjunto de nodos (adyacentes o no al intervalo de trabajo) y que aproximen en el resto del *stencil*. Estudio de su posible inclusión dentro de esquemas de multirresolución *à la Harten* y aplicación de las técnicas *ENO*, *SR*, *WENO* y *PPH*.
- Diseño de las reconstrucciones de *interpolación aproximación* y por *aproximaciones* para la discretización *hat averages*, así como el estudio de sus propiedades, introducción de técnicas no lineales y aplicación a la eliminación de ruido.
- Diseño de indicadores de suavidad específicos para reconstrucciones no interpolatorias con el fin de introducirlos en los esquemas *WENO*.
- Estudio de la aplicabilidad de la técnica *WENO* para las nuevas reconstrucciones admitiendo pesos negativos.

- Estudio de la técnica *PPH* para *stencils* de longitud mayor que 4 tanto para reconstrucciones interpolatorias como para las nuevas reconstrucciones vistas en esta tesis.
- Diseño de nuevos localizadores de discontinuidades tanto para funciones con ruido como sin él.
- Aplicación de la idea *ENO-wavelet-TI* a bases *wavelet* diferentes de *DB2p*.
- Demostración para cualquier orden de la invertibilidad para *ENO-DB2p-TI* y de su equivalencia con *ENO-DB2p*.
- Introducción de mejoras que permitan aplicar el algoritmo *ENO-wavelet-TI* de forma eficiente como algoritmo simultáneo de compresión y eliminación de ruido para funciones discontinuas compuestas por trozos suaves con ruido aditivo.
- Cálculo teórico para los umbrales en multirresolución *à la Harten* con *Shrinkage*,
- Partiendo de las ideas de esta tesis, diseñar métodos eficientes para la eliminación de ruido de imágenes en 2D.
- Creación de un algoritmo que tome decisiones en cuanto al método óptimo de eliminación de ruido a aplicar en función de los datos iniciales y de nuestros objetivos.

Bibliografía

- [1] S. Amat. *Avances en la multirresolución de Harten y aplicaciones*. PhD thesis, Departament de Matemàtica Aplicada. Universitat de València, 2001.
- [2] S. Amat. A review on the piecewise polynomial harmonic interpolation. *Applied Numerical Mathematics*, 58:1168–1185, 2008.
- [3] S. Amat, F. Aràndiga, A. Cohen, and R. Donat. Tensor product multiresolution analysis with error control for compact image representation. *Signal Processing*, 4:587–608, 2002.
- [4] S. Amat, F. Aràndiga, A. Cohen, R. Donat, G. Garcia, and M. von Oehsen. Data compression with ENO schemes: a case study. *Appl. Comput. Harmon. Anal.*, 11(2):273–288, 2001.
- [5] S. Amat, F. Aràndiga, J. V. Arnau, R. Donat, P. Mulet, and R. Peris. *Aproximació numèrica*. Publicacions de la Universitat de València, 2002.
- [6] S. Amat, R. Donat, J. Liandrat, and J.C. Trillo. Analysis of a new non-linear subdivision scheme. applications in image processing. *Foundations of Computational Mathematics*, 6(2):193–225, 2006.
- [7] S. Amat and J. Liandrat. On the stability of the PPH nonlinear multiresolution. *Appl. Comput. Harmon. Anal.*, 18:198–206, 2005.

-
- [8] F. Aràndiga, A. Baeza, A. M. Belda, and P. Mulet. Analysis of WENO schemes for full and global accuracy. *SIAM J. Numer. Anal.*, 49(2):893–915, 2011.
- [9] F. Aràndiga, A. Baeza, and R. Donat. Vector cell-average multiresolution based on Hermite interpolation. *Adv. Comput. Math.*, 28(1):1–22, 2008.
- [10] F. Aràndiga and V. F. Candela. Multiresolution standard form of a matrix. *SIAM J. Numer. Anal.*, 33(2):417–434, 1996.
- [11] F. Aràndiga, A. Cohen, R. Donat, and N. Dyn. Interpolation and approximation of piecewise smooth functions. *SIAM J. Numer. Anal.*, 43(1):41–57, 2005.
- [12] F. Arandiga, A. Cohen, R. Donat, N. Dyn, and B. Matei. Approximation of piecewise smooth functions and images by edge-adapted (ENO-EA) nonlinear multiresolution techniques. *Appl. Comput. Harmon. Anal.*, 24(2):225–250, 2008.
- [13] F. Aràndiga, R. Donat, and P. Mulet. Adaptive interpolation of images. *Signal Processing*, 83:459–464, 2003.
- [14] F. Aràndiga, A. M. Belda, and P. Mulet. Point-value WENO multiresolution applications to stable image compression. *Journal of Scientific Computing*, 43(2):158–182, 2010.
- [15] F. Aràndiga and R. Donat. Multi-scale descomposicion by decimacion and prediccion. Technical report, Departament de Matemàtica Aplicada. Universitat de València, 00-02.
- [16] F. Aràndiga and R. Donat. Nonlinear mustiscale decompositions: The aproach of A. Harten. *Numerical Algorithms*, 23:175–216, 2000.
- [17] F. Aràndiga and R. Donat. Stability through synchronization in nonlinear multiscale transformations. *SIAM J. Sci. Comput.*, 29(1):265:289, 2007.
- [18] F. Aràndiga, R. Donat, and A. Harten. Multiresolution based on weighthted averages of the hat function I: Linear reconstruction oparators. *SIAM J. Numer. Anal.*, 36(1):160–203, 1999.

- [19] F. Aràndiga, R. Donat, and A. Harten. Multiresolution based on weighted averages of the hat function II: Nonlinear reconstruction operators. *SIAM J. Sci. Comput.*, 20(3):1053–1093, 1999.
- [20] A. Arneodo. *Wavelet Analysis of fractals: From the Mathematical Concepts to Experimental Reality: Theory and Applications*. Ed. G. Erlebacher and M. Hussaini and L. Jameson, Oxford Univ. Press, 1996.
- [21] A. M. Belda. *Técnicas de Interpolación WENO y su aplicación al procesamiento de imágenes*. PhD thesis, Universitat de València, 2010.
- [22] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *SIAM J. Numer. Anal.*, 44:141–183, 1991.
- [23] Taswell C. The what, how, and why of wavelet shrinkage denoising. *Comput Sci Eng*, 2:12–9, 2000.
- [24] P. Carré and D. Helbert. Ridgelet decomposition: Discrete implementation and color denoising. In *Wavelet Applications in Industrial Processing III*. SPIE, October 2005.
- [25] T. F. Chan and H. M. Zhou. Adaptive eno-wavelet transforms and its application in image compression. In *Proceedings to the 12th International Conference on Domain Decomposition Method*, Oct. 25-29 1999.
- [26] T. F. Chan and H. M. Zou. ENO-wavelet transforms for piecewise smooth functions. *SIAM J. Numer. Anal.*, 40(4):1369–1404, 1994.
- [27] J. Chen, J. Benesti, Y. Huang, and E.J. Diethorn. *Fundamentals of Noise Reduction*. Springer Handbook, 2008.
- [28] A. Cohen, N. Dyn, and B. Matei. Quasilinear subdivision schemes with applications to ENO interpolation. *Appl. Comput. Harmon. Anal.*, 15:89–116, 2003.

-
- [29] S. Conte and C. de Boor. *Análisis numérico*. McGraw Hill, 1974.
- [30] K. Dadourian. *Schemes de subdivision, analyses multiresolutions non-lyneaires. Aplications*. PhD thesis, Université de Provence. U.F.R. M.I.M. Marseilles, 2008.
- [31] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure and Applied Mathematics*, 41:909–996, 1988.
- [32] I. Daubechies. *Ten lectures on Wavelets*. CBMS-NSF Regional Conference series in Applied Mathematics. Rutgers University and AT&T Bell Laboratories, 1993.
- [33] M. N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *A paraître dans IEEE Transactions Image on Processing*, 2004.
- [34] D. L. Donoho. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [35] D. L Donoho. Denoising via soft thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.
- [36] D. L Donoho. Wavelet shrinkage: Asymptotia? with discussion. *J. Royal Statistical Soc., Series B*, 57(2):301–369, 1995.
- [37] J. Durbin and G. S. Watson. Testing for serial correlation in least-squares regression. *Biometrika*, 38:159–171, 1951.
- [38] B. A. Forouzan. *Data communication and Networking*. McGraw Hill, 2003.
- [39] H.-Y. Gao. Wavelet shrinkage denoising using the nonnegative garrote. *J. Comput. Graph. Statist.*, 7(4):469–488, 1998.
- [40] H.-Y. Gao and A. G. Bruce. *Waveshrink with firm shrinkage*. Technical Report 39, StatSci Division of MathSoft, Inc., 1996.
- [41] D. Gnanadurai and V.Sadasivam. An efficient adaptive thresholding technique for wavelet based image denoising. *International Journal of Signal Processing*, 2:114–119, 2006.

-
- [42] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69:331–371, 1910.
- [43] A. Harten. ENO schemes with subcell resolution. *J. Comput. Phys.*, 83:148–184, 1989.
- [44] A. Harten. Discrete multiresolution analysis and generalized wavelets. *J. Appl. Numer. Math.*, 12:153–192, 1993.
- [45] A. Harten. Multiresolution representation of data. *UCLA CAM Report*, pages 93–13, 1993.
- [46] A. Harten. Multiresolution representation of data II: General framework. *SIAM J. Numer. Anal.*, 33:1205–1256, 1996.
- [47] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes III. *J. Comput. Phys.*, 71:231–303, 1987.
- [48] A. Harten and I. Yad-Shalom. Fast multiresolution algorithms for matrix-vector multiplication. *SIAM J. Numer. Anal.*, 37:1191–1218, 1994.
- [49] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36(3):377–412, 1994.
- [50] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126(1):202–228, 1996.
- [51] I. M. Johnstone and D. L. Donoho. Adapting to smoothness via wavelet shrinkage. *Journal of the Statistical Association*, 90(432):1200–1224, 1995.
- [52] Xu-Dong Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, 1994.
- [53] F. Luisier, T. Blu, B. Forster, and M. Unser. Which wavelet bases are the best for image denoising?. *Proceedings of the SPIE*, 5914:100–111, 2005.

- [54] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases $L^2(\mathbb{R})$. *Transactions of the American mathematical society*, 315(1):69–87, 1989.
- [55] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):647–693, 1989.
- [56] F. Martínez, A. Peris, and F. Rodenas. *Tratamiento de señales digitales mediante wavelets y su uso con MATLAB*. Editorial Club Universitario, 2004.
- [57] B. Matalon, M. Elad, and M. Zibulevsky. Image denoising with the contourlet transform. In *In Proceedings of SPARSE-05*, 2005.
- [58] Y. Meyer. *Wavelets: Algorithms and applications*. SIAM: Philadelphia, 1993.
- [59] Y. Meyer. L'analyses par ondelettes. *Pour la Science*, September 1987.
- [60] D. Mizrahi. *Remoiving Noise from Discontinous Data*. PhD thesis, School of Mathematical Sciences. Applied Math Department, 1991.
- [61] N. Nezamoddini-Kachouie, P. Fieguth, and E. Jernigan. Bayesshrink ridgelets for image denoising. In *Proc. ICIAR 2004*, September 2004.
- [62] H.C. Noriega, R. Rios, and P. Reumay. Compactación de ondeletes para remoción de ruido en señales de vibraciones transversales de sierras circulares. *Información Tecnológica*, 14:3:89–96, 2003.
- [63] R. Rangarajen and R. Venkataramanan. Image denoising using wavelets. *Wavelets and Time Frecuency*, 2002.
- [64] N. E. Saving and k. J. White. The durbin-watson test for serial correlation with extreme sample sizes of many regressors. *Econometrica*, 45:1989–1996, 1977.

-
- [65] W.-C. Shann and C.-C. Yen. Exact solutions for daubechies orthonormal scaling coefficients. Technical Report 9704, Department of Mathematics, National Central University, Sep 13 1996.
- [66] J. Shi, C. Hu, and C.-W. Shu. A technique of treating negative weights in weno schemes. *Journal of Computational Physics*, 175:108–127, 2002.
- [67] J. L. Starck, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Trans. Image Processing*, 11(6):670–684, 2002.
- [68] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA., 1996.
- [69] P. Tchamitchian. *Wavelets, Functions, and Operators in Wavelets: Theory and Applications*. Ed. D. Erlebacher and M. Hussaino and L. Jameson, Oxford Univ. Press, 1996.
- [70] M. Vetterli, S. G. Chang, and Bin Yu. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.
- [71] J. S. Walker and Y. Chen. Image denoising using tree-based wavelet subband correlations and shrinkage. *Optical Engineering*, 39(11):2900–2908, 2000.