

UNIVERSIDAD SAN FRANCISCO DE QUITO

**CONSTRUCCIÓN DE COEFICIENTES WAVELETS
ORTOGONALES CON \mathbb{R} .**

Julio César Ibarra Fiallo.

Tesis de grado presentada como requisito para la obtención del título de
MASTER EN MATEMÁTICAS APLICADAS.

Mayo de 2005

Universidad San Francisco de Quito
Colegio de Graduados
HOJA DE APROBACIÓN DE TESIS

CONSTRUCCIÓN DE COEFICIENTES WAVELETS
ORTOGONALES CON R.

Julio César Ibarra Fiallo

Carlos Jiménez, Ph.D.
Director de Tesis (firma)

Gonzalo Mendieta, Ph.D.
Miembro del Comité de Tesis

Carlos Jiménez, Ph.D.
Director de la Maestría en M.A. (firma)

Victor Viteri, Ph.D.
Decano del Colegio de Graduados (firma)

Quito, mayo de 2005

©Derechos de autor

Julio César Ibarra Fiallo

2005

Resumen

La construcción de wavelets es un problema interesante, por el hecho de que no existen reglas cerradas que permitan su cálculo (exceptuando algunas como la de Haar), desde este punto de vista, explorar la construcción de las wavelets es necesaria, pues ellas brindan el fundamento de la construcción de muchos filtros digitales que son usados en la práctica. En el trabajo se explora la construcción de 4 tipos fundamentales de wavelets como filtros, las cuales fueron construidas por Ingrid Doubechies y no tienen implementación en R, esas son: Doublets y su caso particular de la Wavelet de Haar, Symlets, Coiflets y Coiflets generalizadas.

Palabras Clave. Filtro, señal pura, función escala, wavelet, coeficientes escala, coeficientes wavelet, factorización espectral, análisis multiresolución.

Abstract

Building wavelet is an interesting problem, because there are not closed rules that allow us their computation (except some of them as the Haar's wavelet), from this point of view, to explore the construction of wavelets is necessary, because they give us the basis to construct a lot of digital filters which are used in practice. This work explores the construction of 4 types of wavelets as filters, these wavelets were constructed by Ingrid Douthies and they are not implemented in R program, those are: Doublet and its particular case Haar's Wavelet, Symlets, Coiflets and generalized Coiflets.

Key Words. Filter, pure sign, scale function, wavelet, scale coefficients, wavelet coefficients, spectral factorization, multiresolution analysis.

Índice general

1. Preliminares	1
1.1. Objetivos del trabajo.	1
1.2. Introducción.	1
2. Filtros, Convoluciones y Wavelets	4
2.1. Filtros.	4
2.2. Convoluciones.	8
2.3. Wavelets.	10
2.3.1. Análisis Multiresolución.	10
2.3.2. Relación de las Wavelets con los filtros.	14
3. Construcción de 4 tipos de Wavelets famosas.	17
3.1. Doublets.	17
3.1.1. Wavelet Haar.	20
3.1.2. Generación de doublets.	20
3.1.3. Generación de Symlets.	22
3.1.4. Generación de Coiflets.	25
4. Resultados.	29
4.1. Coeficientes y Gráficos de Doublets.	29
4.2. Coeficientes y gráficos Symlets.	31
4.3. Coiflets Generalizadas y gráficas.	38
5. Conclusiones.	43

Índice de figuras

2.1. Amplitud de la respuesta de Frecuencia.	6
2.2. Amplitud de la respuesta de Frecuencia filtro <i>pasaalto</i>	7
3.1. Función de escala Haar	21
3.2. Función de escala Doubechies N=2	23
3.3. Función de escala Doubechies Symlet N=4	26
3.4. Coiflet N=6	28
4.1. Escala de la Doublets. Momentos N=2,3,4,5,6	32
4.2. Doublets. Momentos N=2,3,4,5,6	33
4.3. Escala de la Doublets. Momentos N=2,3,4,5,6	37
4.4. Doublets. Momentos N=2,3,4,5,6	38
4.5. Coiflet.	40
4.6. Escalas y coiflets generalizadas con un momento nulo.	42

Capítulo 1

Preliminares

1.1. Objetivos del trabajo.

El presente trabajo nació con la idea de construir un paquete de R que emule los resultados producidos por el paquete S-Wave para S-PLUS. Con esta idea en mente, se inicia la primera fase de este fabuloso trabajo con los siguientes objetivos en mente:

1. Estudiar la teoría que fundamenta la construcción de las wavelets.
2. Estudiar la relación de la teoría de los filtros con las wavelets.
3. Programar los algoritmos que permiten reproducir los coeficientes de 4 tipos de wavelets ortogonales básicas.
4. Programar una extensión que permita encontrar los coeficientes de las coiflets generalizadas.

1.2. Introducción.

El análisis de señales es un campo de amplia utilidad, pues a través de él se puede descomponer una señal s cuyo dominio es el tiempo, en un conjunto de señales que la componen, estudiando estas componentes es posible *mejorar* la señal original escogiendo con criterio aquellas componentes importantes (en algún sentido) de tal manera que la información de interés se mantenga (generalmente lo que se busca es eliminar la basura o

ruido de la señal original, aunque a veces es importante identificar el comportamiento del ruido).

Las señales en estudio son finitas en el tiempo, por lo tanto es razonable pensar que su *energía* es finita, en tal caso tiene sentido pensar que $s \in L^2(\mathbb{R})$, es decir

$$\int_{-\infty}^{\infty} s^2(t) dt < \infty \quad (1.1)$$

Para cumplir con la descomposición de s , se debe usar un conjunto de funciones $\{f_n\}_{n \in \mathbb{Z}}$ apropiadas que formen una base de $L^2(\mathbb{R})$ de tal manera que s se pueda expresar [QUIAN02] como

$$s(t) = \sum_{n \in \mathbb{Z}} f_n(t) \quad (1.2)$$

Puesto que L^2 es un espacio vectorial de dimensión infinita [KREY78], se espera escoger el conjunto $\{f_n \in L^2/n \in \mathbb{Z}\}$ de manera apropiada. Esta manera se halla bien estudiada bajo el criterio del *análisis multiresolución* (**MRA**) ([DAUB92], Cap. 5), el cual permite encontrar de manera ordenada un criterio de elección. La idea básica es que se elige una función *escala* a partir de la cual se añaden un número finito de funciones (*wavelets*) capaces de captar el detalle de la función original de tal manera que la señal original s puede ser

$$s(t) = c_J \varphi_J(t) + \sum_{i=1}^{J-1} c_i \psi_i(t) \quad (1.3)$$

En esta relación J se le conoce como el nivel de descomposición de la señal original s de la cual poseemos una muestra de impulsos (observaciones) cuya longitud por facilidad se espera que sea 2^J , aunque se intenta resolver este problema dándole a la muestra un tratamiento *cíclico*.

Como el interés del presente trabajo no es principalmente teórico, sino más bien el de mostrar de la manera más sencilla posible el proceso de implementación de rutinas de construcción de wavelets y su uso en el análisis y síntesis de una señal dada, los resultados usados serán remitidos a la bibliografía de la cual fueron obtenidos, la cual proporcionará con

más detalle el fundamento teórico para aquel lector interesado en el tema.

Capítulo 2

Filtros, Convoluciones y Wavelets

2.1. Filtros.

Filtro es un aparato a través del cual se pretende hacer pasar un líquido que se pretende clarificar. en otras palabras, es un aparato que permite retener las impurezas.

La clase de filtros sobre las cuales se pretende discutir no se aleja de esta noción, pues lo que se entiende por filtro va a realizar exactamente la tarea de *clarificar* una señal¹. Esta señal se supondrá muestreada en intervalos T de tiempos iguales (normalizados a 1) es decir:

$$s = \{s(t_k)/t_k = k \in \mathbb{Z}\} \quad (2.1)$$

Donde solamente para efectos teóricos se se toma $k \in \mathbb{Z}$, pues en realidad se analiza s solamente en un intervalo de tiempo finito, en tal caso $k \in A \subset \mathbb{Z}$ donde $|A| = n$.

Para definir un filtro se utilizará la transformada de Fourier discreta de la señal definida por:

$$\hat{s}(\omega) = \sum_{k \in \mathbb{Z}} s(k) e^{-ik\omega} \quad (2.2)$$

Donde ω es la frecuencia y \hat{s} la transformada es periódica con periodo 2π . Entenderemos por bajas frecuencias a valores de ω cercanos (en valor absoluto) a 0, y por frecuencias

¹Dentro del contexto se va entender señal como una muestra de impulsos o simplemente un conjunto de datos.

altas cuando $|\omega| \approx \pi$. Una señal sinusoidal pura viene definida como:

$$s_\omega(k) = e^{-i\omega k} \quad \text{para cada } k \quad (2.3)$$

Notar que si $\omega = 0$ la señal es constante y siempre igual a 1, es decir no posee ninguna clase de vibración, mientras que se $|\omega| = \pi$ la señal que se produce tiene la máxima frecuencia posible y alterna sus valores entre 1 y -1 . Ahora estamos listos para decir qué es un filtro.

Si un sistema recibe una señal pura y al responder, su repuesta es también pura, es decir, su único cambio está dado en la fase y/o en la amplitud, el sistema se llama *lineal invariante en el tiempo*. Un sistema como éstos recibe una señal $s_\omega(k)$ y devuelve una señal del tipo $H(\omega)s_\omega(k)$, donde $H(\omega)$ es un factor que altera solamente altera la fase y/o la amplitud. El factor $H(\omega)$ se lo conoce como *respuesta de frecuencia* y también se lo escribe como $H(e^{i\omega})$. A un sistema lineal invariante en el tiempo (LTI)² se le da el nombre de *filtro*. Para una discusión más cuidadosa del significado de alteración de fase consultar [STRANG00].

Cuando se aplica un filtro H a una señal s , algunas de las componentes de frecuencia son reducidos. Cuando un filtro es de *respuesta de impulso finita* (FIR³) cada respuesta $z(k)$ no es más que una combinación lineal de un número finito de entradas o valores de la señal s . Un ejemplo sencillo tomado de [STRANG00] al respecto es el de la media móvil:

$$z(k) = \frac{1}{2}s(k) + \frac{1}{2}s(k-1)$$

Un filtro como éste es invariante en el tiempo, pues basta ver que los coeficientes de la combinación lineal no cambian con el tiempo, también es *causal* porque no intervienen valores futuros, es decir la *muestra filtrada* $z(k)$ solamente depende de las muestras $s(k)$ y $s(k-1)$, no intervienen valores $s(l)$ con $l \geq k+1$. Prácticamente podríamos decir que un filtro causal es una especie de “*media móvil ponderada*”. Un filtro el cual de una manera estricta promedia valores (combinación lineal con coeficientes cuya suma es positiva e igual a 1) de la señal s es un filtro *pasabajo*, es decir las frecuencias mayormente bajas son

²Del inglés Linear Time Invariant.

³Del inglés Finite Impulse Response

conservadas, por ejemplo si tengo la señal pura $s = (\dots, 1, 1, 1, 1, 1, \dots)$ el filtrado promedio producirá $z = (\dots, 1, 1, 1, 1, 1, \dots)$, note que en este caso la frecuencia de s es 0 porque no existe vibración y es dejada pasar por nuestro filtro. Todo lo contrario sucede con la señal de frecuencia alta dada por $s = (\dots, -1, 1, -1, 1, -1, \dots)$, al ser filtrada por el promedio produce $z = (\dots, 0, 0, 0, 0, \dots)$, la cual tiene frecuencia nula, en conclusión la frecuencia alta $\omega = \pi$ fue parada. Si s es una señal pura es claro que para la frecuencia $\omega \in]0, \pi[$ se tiene que $H(\omega) = \frac{1}{2} + \frac{1}{2}e^{-i\omega}$ de donde la amplitud específicamente es $|H| = \cos(\frac{\omega}{2})$, observe la figura 2.1. Para este caso es claro que las frecuencias cercanas a 0 mantienen

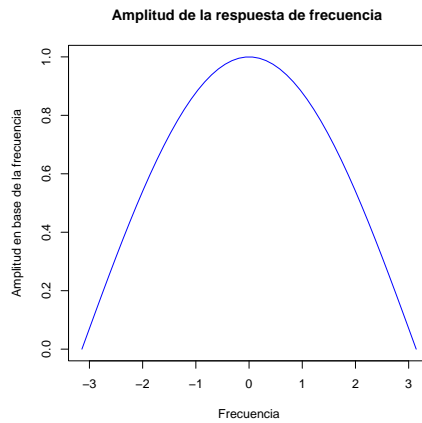


Figura 2.1: Amplitud de la respuesta de Frecuencia.

la amplitud de la respuesta de frecuencia, el filtro es *pasabanda*, es decir se conservan las frecuencias bajas. Por otro lado la amplitud de frecuencia de las altas es mermada e incluso disminuida, lo que le da un comportamiento de *parabanda* (las frecuencias altas no pasan). La fase para la respuesta de frecuencia H de la que se está hablando es $-\frac{\omega}{2}$ y depende linealmente de ω . Que la fase sea lineal depende exclusivamente de la simetría del filtro [DAUB92], para el caso en particular, al revertir $h(0) = \frac{1}{2}$ y $h(1) = \frac{1}{2}$ alrededor de $\frac{1}{2}$ no sufre ninguna alteración.

Para dejar pasar frecuencias altas necesitamos un filtro que se basa en diferencias, pongamos otro ejemplo, digamos que esta vez lo que se desea es que la respuesta mida la diferencia entre la muestra actual y la anterior, en cierta manera lo que se desea es tener

una medida de la variación o de la vibración de la señal, entonces tomemos:

$$z(k) = \frac{1}{2}s(k) - \frac{1}{2}s(k-1)$$

Note que en este caso la señal $s = (\dots, 1, 1, 1, \dots)$ después de ser filtrada produce la señal $z = (\dots, 0, 0, 0, \dots)$, mientras que la señal $s = (\dots, -1, 1, -1, 1, \dots)$ después del filtrado se reproduce exactamente. Observe que la suma de los coeficientes es 0, las frecuencias altas se dejan pasar mientras que las bajas se detienen. Para $s(k) = e^{-i\omega k}$ la respuesta de frecuencia H es ahora $H(\omega) = \frac{1}{2} - \frac{1}{2}e^{-i\omega}$ con amplitud $|H| = |\sin(\frac{\omega}{2})|$ la cual es 0 para $\omega = 0$ y es 1 para $\omega = \pi$, además su factor de fase es $e^{-\frac{i\omega}{2}}$, el factor imaginario aparece por la *antisimetría* del filtro en uso (es decir al ser revertido y cambiadas de signos las posiciones pares el filtro permanece intacto), lo que implica que en esta caso tenemos una función de fase lineal pero con salto. En conclusión los filtros antisimétricos también producen fases lineales [DAUB92]. En corto, podemos decir que los filtro pasabajo

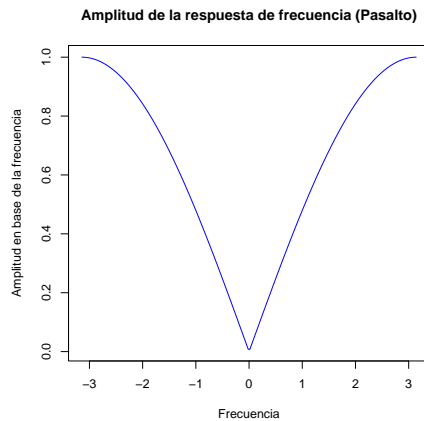


Figura 2.2: Amplitud de la respuesta de Frecuencia filtro *pasaalto*.

preservan la suavidad de las señales, mientras que los pasaalto capturan la rugosidad o vibración de ellas.

En este punto se debería estar consiente de que un *filtro causal* de longitud $K + 1$ está determinado por un conjunto de coeficientes $\{h(0), h(1), \dots, h(K)\}$ el cual filtra una

señal discreta s en base de la combinación lineal dada en la fórmula:

$$z(k) = h(0)s(k) + h(1)s(k-1) + \dots + h(K)s(k-K) \quad (2.4)$$

La cual puede ser expresada de manera reducida como:

$$z(k) = \sum_{l=0}^K h(l)s(k-l) \quad (2.5)$$

La relación dada por (2.5) es una de las más importantes operaciones que se realiza en el proceso de filtrado y descomposición de señales y se la conoce con el nombre de *convolución discreta*.

2.2. Convoluciones.

Para iniciar éste vistazo de lo que son las convoluciones, empezaremos mirando el ejercicio siguiente. Supongamos que deseamos multiplicar el polinomio de segundo grado $p(x) = a + bx + cx^2$ con el polinomio de primer grado $q(x) = v + wx$, dos preguntas que nos vienen casi enseguida a la mente son:

- De qué grado es el polinomio resultante?
- Cuáles son los coeficientes de este polinomio?

Efectuando el cálculo de la manera estándar obtendremos como resultado el polinomio de grado 3:

$$av + awx + bvx + bwx^2 + cvx^2 + cwx^3$$

Agrupando los términos semejantes de manera adecuada nos queda el polinomio:

$$av + (aw + bv)x + (bw + cv)x^2 + cwx^3$$

lo cual no aporta nada a lo ya conocido, pero con un poco de cuidado podríamos observar que el polinomio se puede escribir como:

$$(0w + av) + (aw + bv)x + (bw + cv)x^2 + (0v + cw)x^3$$

lo cual en términos del producto interior bidimensional nos deja ver que:

$$\begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} 0 \\ a \end{bmatrix} + \begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} a \\ b \end{bmatrix} x + \begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} b \\ c \end{bmatrix} x^2 + \begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} c \\ 0 \end{bmatrix} x^3$$

Lo cual, aunque no lo parezca nos recuerda el comportamiento dado para un filtro en la fórmula (2.5). Para que sea aplicable tome $h = (w, v)$ y $s = (\dots, 0, c, b, a, 0, \dots)$ al aplicar la idea de filtro causal, la señal filtrada por los coeficientes de h sería $z = (\dots, 0, vc, wc + vb, wb + va, wa, 0, \dots)$.

El ejercicio que se acaba de realizar es una de las motivaciones para definir lo que es una convolución discreta entre 2 vectores, de distintas longitudes. Digamos por ejemplo, dados los vectores $\alpha = (w, v)$ y $\beta(c, b, a)$ se define la convolución de α y β como:

$$\alpha * \beta = \left(\begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} c \\ 0 \end{bmatrix}, \begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} b \\ c \end{bmatrix}, \begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} w \\ v \end{bmatrix} \bullet \begin{bmatrix} 0 \\ a \end{bmatrix} \right) \quad (2.6)$$

donde $\alpha * \beta$ se lee α convolucionado con β . Al paso podemos hacer algunas observaciones.

- Los coeficientes del polinomio $p(x)q(x)$ es la convolución de los coeficientes de $p(x)$ con los de $q(x)$
- La longitud del vector $\alpha * \beta$ es la longitud de α más la longitud de β menos 1.
- La idea se puede generalizar sin problemas para vectores de cualquier longitud, digamos m y n , en tal caso la longitud de la convolución sería $m + n - 1$.

El tipo de convolución que se acaba de explicar es lo que se conoce como *convolución no cíclica* y es ampliamente utilizada en la transformada de Fourier, para mayor detalle mire el capítulo III en [GOSCHAN99].

Otro tipo de convolución que no aparece de manera natural en lo que va de este trabajo, es la que se llama *convolución cíclica*, en tal caso la idea es que oír decir algo definimos un valor de ciclo, digamos por ejemplo que $x^3 = 1$, lo cual haría que el grado máximo de los polinomios sea 2, en tal caso el resultado de la convolución cíclica tendría exactamente longitud 3 y los vectores a convolucionarse, todos tendrían longitud 3. En [STRANG00C] se puede encontrar un extenso tratado sobre este tópico.

2.3. Wavelets.

Existe numerosos tratados sobre wavelets, uno muy interesante y comprensible es el presentado en el capítulo V de [QUIAN02]. Para el caso se presentarán solamente los principales resultados antes de pasar a describir los algoritmos de análisis de señales.

2.3.1. Análisis Multiresolución.

Como se había manifestado, buscamos encontrar una descomposición de una señal s de energía finita en el espacio L^2 , de tal manera que s pueda ser expresada como en (1.3). Para ello se tomará como base el *análisis multiresolución* (MRA)⁴.

Para entender el análisis multiresolución, tomaremos una familia de subespacios vectoriales $\{V_m\}_{m \in \mathbb{Z}}$ en L^2 , la cual mantiene una estructura anidada, es decir se cumple que $\dots \subset V_{m-1} \subset V_m \subset V_{m+1} \dots$, tal que:

1.

$$\lim_{m \rightarrow \infty} V_m = L^2(\mathbb{R}) \quad (2.7)$$

2.

$$\lim_{m \rightarrow -\infty} V_m = \{0\} \quad (2.8)$$

3. La función $\phi(t) \in V_0$ es ortogonal a todas sus traslaciones $\phi_{0,n} = \phi(t - n)$, $n \in \mathbb{Z}$, es decir

$$\int_{-\infty}^{\infty} \phi(t) \phi_{0,n}^*(t) dt = \delta(n) \quad (2.9)$$

⁴Del inglés Multiresolution Analysis

La función ϕ es llamada la *función de escala*. En este caso el asterisco denota la conjugada de ϕ , porque en general ϕ puede ser compleja, aunque es de nuestro interés trabajar con ϕ real. La función δ es la delta usual.

4. Si $s(t) \in V_0$ entonces

$$s(t) = \sum_{n \in \mathbb{Z}} c_n \phi_{0,n}(t) \quad (2.10)$$

Es decir el conjunto $\{\phi_{0,n}\}_{n \in \mathbb{Z}}$ es una base ortogonal de V_0 .

5. $s(t) \in V_m$ si y sólo si $s(2t) \in V_{m+1}$, para todo $m \in \mathbb{Z}$. Esta es una condición muy importante, y se la puede entender de varias maneras.

- a) Toda función en V_m puede ser expresada con funciones con el doble de frecuencia que se encuentran en V_{m+1} .
- b) La resolución de una función dada en V_m puede ser conseguida en base de resoluciones más altas de funciones en V_{m+1}

Como se manifiesta en [PFAF02] para que una señal $s \in L^2(\mathbb{R})$ pueda ser expresada de manera correcta, no podemos solamente encontrar un nivel de resolución m asociado a V_m , sino que también deseamos encontrar una manera de medir sus variaciones, es decir ser capaces de mostrar información respecto de las frecuencias que la componen. Para lograr este objetivo se procede de la siguiente manera:

Puesto que $V_m \subset V_{m+1}$, se puede encontrar un espacio vectorial W_m que sea el complemento ortogonal de V_m con respecto de V_{m+1} , es decir que

$$V_{m+1} = V_m \oplus W_m \quad (2.11)$$

Donde usando de manera recurrente (2.11) se puede ver que

$$V_{m+1} = V_0 \oplus W_0 \oplus W_1 \oplus \dots \oplus W_m \quad (2.12)$$

La interpretación de la ecuación (2.12) es que una función al nivel de resolución $m + 1$ puede expresarse en base a la suma de una función con resolución base, digamos 0, con

funciones todas ortogonales que le permiten alcanzar el nivel de resolución $m + 1$. Note que la ecuación (1.2) recoge claramente la idea.

Una función $\phi(t) \in L^2(\mathbb{R})$ se considera una buena función de escala si es tal que a partir de ella, todas las propiedades del análisis multiresolución se cumplen. Para generar el subespacio vectorial a partir de ella V_m tomamos las funciones:

$$\phi_{m,k}(t) = 2^{\frac{m}{2}} \phi(2^m t - k) \quad m, k \in \mathbb{Z} \quad (2.13)$$

De donde podemos decir [DAUB92] que:

$$V_m = \overline{\text{span}_{k \in \mathbb{Z}} \{ \phi_{m,k}(t) \}} \quad (2.14)$$

Observe que $\phi(t) = \phi_{0,0}(t) \in V_0$. De W_0 , que es el complemento ortogonal de V_0 podemos obtener una función ψ cuyo comportamiento es idéntico al de ϕ , es decir a que tomando compresiones y traslaciones de ella, se puede [DAUB92] encontrar bases para los espacios W_m con $m \in \mathbb{Z}$. Así podemos definir:

$$\psi_{m,k}(t) = 2^{\frac{m}{2}} \psi(2^m t - k) \quad (2.15)$$

y de la misma manera se obtiene que:

$$W_m = \overline{\text{span}_{k \in \mathbb{Z}} \{ \psi_{m,k}(t) \}} \quad (2.16)$$

La ecuación (1.2) se podría reescribir de la siguiente manera:

$$s(t) = \sum_n c_{m_0,n} \phi_{m_0,n}(t) + \sum_{k=m_0}^{m-1} \sum_n d_{k,n} \psi_{k,n} \quad m > m_0 \quad (2.17)$$

m es el nivel de resolución más alto que puede alcanzar la función s y m_0 es el nivel de resolución base.

En adelante nos referiremos a las funciones $\psi_{k,n}$ como *wavelets*.

En base del producto interior de $L^2(\mathbb{R})$ y de acuerdo a las propiedades del **MRA**

podemos enseguida concluir que, los coeficientes de las funciones escala y wavelets se pueden calcular como sigue:

$$c_{m_0,n} = \langle s, \phi_{m_0,n} \rangle \quad \text{y} \quad d_{k,n} = \langle s, \psi_{k,n} \rangle \quad (2.18)$$

En este punto las preguntas respecto de qué forman tienen las funciones escala y cómo calcularlas deben estar ya saltando a nuestra mente. Solamente se adelantará que en este punto la teoría de los filtros entra al juego y que se lo analizará en la siguiente sección.

La manera de descomponer la señal s tal como se la ha visto se conoce como *descomposición ortogonal*. En general [GOSCHAN99] $\{\phi(t - k), k \in \mathbb{Z}\}$ y $\{\psi(t - k), k \in \mathbb{Z}\}$ no necesitan ser ortogonales a ellas mismas, es decir:

$$\int_{-\infty}^{\infty} \phi(t)\phi(t-l)dt \neq 0 \quad (2.19)$$

$$\int_{-\infty}^{\infty} \psi(t)\psi(t-l)dt \neq 0 \quad (2.20)$$

Relajando la condición de que W_m sea ortogonal a V_m , se asume que la wavelet $\psi_{m,n} \in W_m$ posee una función dual $\tilde{\psi}_{m,n} \in \tilde{W}_m$, la dualidad implica que se cumple la condición de biortogonalidad, digase:

$$\langle \psi_{k,j}, \tilde{\psi}_{m,l} \rangle = \delta_{k,j} \cdot \delta_{m,l} \quad (2.21)$$

Aunque no se requiere que $W_m \perp V_m$, se necesita obligatoriamente que $\tilde{W}_m \perp V_m$, caso contrario la reconstrucción no será factible y el efecto de redundancia será inevitable [GOSCHAN99]. De la misma manera se considera una función escala dual $\tilde{\phi}_{m,k}$, la cual genera otro **MRA** en $L^2(\mathbb{R})$. El proceso de descomposición que resulta de usar el **MRA** “dual” es llamado *descomposición biortogonal*, en tal caso la ecuación (2.17) produce los coeficientes:

$$c_{m_0,n} = \langle s, \tilde{\phi}_{m_0,n} \rangle \quad \text{y} \quad d_{k,n} = \langle s, \tilde{\psi}_{k,n} \rangle \quad (2.22)$$

Para el caso del presente trabajo, solamente tomaremos en cuenta la descomposición ortog-

onal, dedicándonos específicamente a 4 tipos de wavelets básicos⁵, todas ellas ortogonales.

2.3.2. Relación de las Wavelets con los filtros.

A partir de este punto se explicará la relación que existe entre el **MRA** que da origen a las wavelets con los filtros estudiados en la primera sección. Para ello haremos algunas observaciones, puesto que V_m y W_m estén en L^2 podemos decir que tanto $\|\phi\|_{L^2}$ y $\|\psi\|_{L^2}$ son finitas y por conveniencia las podemos normalizar a cualquier valor positivo, digamos 1. Lo que da como resultado que:

$$\int_{-\infty}^{\infty} \phi(t)\phi^*(t)dt = 1 \quad (2.23)$$

$$\int_{-\infty}^{\infty} \psi(t)\psi^*(t)dt = 0 \quad (2.24)$$

En base de (2.23) y (2.24) se puede establecer que las funciones $\phi_{m,k}$ y $\psi_{m,k}$ con $m, k \in \mathbb{Z}$ definidas en (2.13) y (2.15) también tienen norma 1. Además se requiere [STRANG89] que la función de escala posea energía finita digamos 1. De donde se puede concluir a partir de:

$$\phi(t) = \sum_{k \in \mathbb{Z}} c_k \phi(2t - k) \quad (2.25)$$

que

$$\sum_{k \in \mathbb{Z}} c_k = 2 \quad (2.26)$$

Lo cual asegura la unicidad de los coeficientes c_k .

La ecuación (2.25) se puede obtener de manera sencilla, comprendiendo que $\phi(t) \in V_0 \subset V_1$. De la misma manera sucede con la wavelet $\psi(t) \in W_0 \subset V_1$, lo cual se resume en las relaciones:

$$\phi(t) = \sum_{k \in \mathbb{Z}} h_0[k] \phi(2t - k) \quad (2.27)$$

$$\psi(t) = \sum_{k \in \mathbb{Z}} h_1[k] \phi(2t - k) \quad (2.28)$$

⁵Haar, Doublets, Symlets y Coiflets

Hemos notado con h los coeficientes para empezar a recordar las notaciones introducidas para los filtros. Se puede demostrar [DAUB92], que

$$\sum_{k \in \mathbb{Z}} h_1[k] = 0 \quad (2.29)$$

como requerimiento, a partir de que

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$

Tomando la transformada de Fourier a ambos lados de las *relaciones de escala* (2.27) y (2.28) se obtiene

$$\hat{\phi}(\omega) = H_0(z) \hat{\phi}\left(\frac{\omega}{2}\right) \quad (2.30)$$

$$\hat{\psi}(\omega) = H_1(z) \hat{\phi}\left(\frac{\omega}{2}\right) \quad (2.31)$$

donde

$$H_0(z) = \frac{1}{2} \sum_{k \in \mathbb{Z}} h_0[k] z^k \quad (2.32)$$

$$H_1(z) = \frac{1}{2} \sum_{k \in \mathbb{Z}} h_1[k] z^k \quad (2.33)$$

con $z = e^{-\frac{i\omega}{2}}$, en consecuencia $z^k = e^{-\frac{ik\omega}{2}}$ de acuerdo con la ecuación (2.3) es una señal pura.

Mientras que los $d_{m,n}$ son los coeficientes de las wavelets, en cambio los coeficientes $c_{m,n}$ son los coeficientes de la aproximación de la señal s al nivel de escala m ([QUIAN02]). A continuación se dará base a la afirmación anterior.

Utilizando la igualdad de Parseval⁶ y la primera igualdad de (2.18) para m , tenemos que:

$$c_{m,n} = 2^{\frac{m}{2}} \int_{-\infty}^{\infty} s(t) \phi^*(2^m t - n) dt = \frac{1}{2\pi} 2^{-\frac{m}{2}} \int_{-\infty}^{\infty} \hat{s}(\omega) \hat{\phi}^*(2^{-m}\omega) e^{-i2^{-m}\omega n} d\omega \quad (2.34)$$

⁶Mire [QUIAN02] el capítulo II, página 41, fórmula 2.61

Donde \widehat{s} y $\widehat{\phi}$ son las transformadas de Fourier de s y ϕ respectivamente. Por otro lado, debido a que $\widehat{\phi}(0) = \int_{-\infty}^{\infty} \phi(t)dt = 1$ la ecuación (2.34) para un m lo suficientemente grande nos dice que:

$$c_{m,n} \approx \frac{1}{2\pi} 2^{-\frac{m}{2}} \int_{-\infty}^{\infty} \widehat{s}(\omega) e^{-i2^{-m}\omega n} d\omega = 2^{-\frac{m}{2}} s(2^{-m}n) \quad (2.35)$$

Este resultado, permite concluir sin pérdida de generalidad [QUIAN02] que:

$$c_{m,n} \equiv s[n] \equiv s(t) |_{t=2^{-m}n} \quad (2.36)$$

El error relacionado se puede encontrar en [X-GX92] y [X-GX94].

Aplicando la relación (2.36) en el cálculo de los coeficientes $c_{m-1,n}$ y $d_{m-1,n}$, se obtiene que:

$$c_{m-1,n} = \sqrt{2} \sum_{k \in \mathbb{Z}} h_0[2n - k] c_{m,k} \quad (2.37)$$

$$d_{m-1,n} = \sqrt{2} \sum_{k \in \mathbb{Z}} h_1[2n - k] c_{m,k} \quad (2.38)$$

Debido a las características de h_0 y h_1 y de acuerdo a lo expuesto en la primera sección en cuanto a filtros, podemos afirmar que el modo de encontrar los coeficientes $c_{m-1,n}$ y $d_{m-1,n}$ corresponden a un sistema que se lo puede calificar como filtro, donde está por demás decir que la convolución está presente. Es claro que h_0 se comporta como un filtro pasabajo y h_1 como un filtro pasaalto.

En esta sección se ha discutido tres temas básicos para la descomposición de señales: filtros, convoluciones y wavelets. Los resultados presentados permiten encontrar los coeficientes de los filtros basados en características deseadas. En la siguiente sección se discutirá sobre los algoritmos de construcción de las wavelets y se mostrará algunos de los resultados obtenidos en \mathbb{R} . Las construcciones de los filtros están basados en la teoría proporcionada por Ingrid Daubechies en [DAUB92], y una extensión hecha para las coiflets en [WEIBOVEV98]. La mayor parte de los resultados se presentan en la sección correspondiente.

Capítulo 3

Construcción de 4 tipos de Wavelets famosas.

Entrar en la finura de la construcción de las wavelets no es un camino que se diga sencillo, pero es excitante entender los criterios que permiten la construcción de estas poderosas funciones.

Los 4 tipos de wavelets básicos que discutiremos son:

1. Haar (Caso particular de las Doublets).
2. Doublets.
3. Symlets.
4. Coiflets y Coiflets generalizadas.

3.1. Doublets.

Para construir una función ψ que sea considerada una wavelet, podemos requerir que el conjunto de sus traslaciones, dado por $\{\psi(x-l)/l \in \mathbb{Z}$ forme una base de Riesz de W_0 , y en ese caso (2.20) debe ser 0 para $l \neq 0$. Además con el fin de conseguir regularidad de

ψ , se puede exigir que:

$$\int_{-\infty}^{\infty} x^l \psi(x) dx = 0 \quad l = 0, 1, \dots, L-1 \quad (3.1)$$

Lo que nos da una idea del grado de los polinomios que pueden ser dominados por la función ψ , lo cual a su vez mide la rapidez con la que decae ψ hacia cero. La expresión dada por (3.1) se conoce como momento de orden l de la wavelet ψ . Además observe que para $l = 0$ se garantiza que la energía total de ψ sea 0, con lo cual se asegura la obtención de un filtro pasaalto.

En el capítulo 5 de [DAUB92] está demostrado que a partir de la ecuación (2.32), bajo una exigencia de ortonormalidad de la función $\phi_{0,n}$ se tiene en términos de ω que:

$$|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1 \quad (3.2)$$

con H_0 necesariamente continua. Observe además que (3.2) es como una especie de relación fundamental como la de las funciones sin y cos, la cual expresada en términos de la función cos es:

$$|\cos(\omega)|^2 + |\cos(\omega - \frac{\pi}{2})|^2 = 1 \quad (3.3)$$

Se nota claramente que el cambio que posee el polinomio trigonométrico H_0 está en la traslación con respecto de su ortogonal, aunque el hecho de poseer soporte compacto los diferencia del sin y cos los cuales no poseen soporte compacto en \mathbb{R} . Otra ventaja de H_0 es que se vuelven trasladables.

Debido al interés de construir las funciones ϕ y ψ lo suficientemente regulares o al menos razonablemente regulares, en [DAUB92] está demostrado que H_0 debe ser de la siguiente forma:

$$H_0(\xi) = \left(\frac{1 + e^{i\xi}}{2}\right)^N \ell(\xi) \quad (3.4)$$

Con $\xi = -\omega$, para no olvidar la idea de señal pura.

De la identidades $\cos^2\left(\frac{\xi}{2}\right) = \left(\frac{e^{i\xi/2} + e^{-i\xi/2}}{2}\right)^2$ y $\cos^2\left(\frac{\xi}{2}\right) = \frac{1 + \cos(\xi)}{2}$ el siguiente resultado:

$$|H_0(\xi)|^2 = \left(\cos^2\frac{\xi}{2}\right)^N L(\xi) \quad (3.5)$$

donde $L(\xi) = |\ell(\xi)|^2$. es también un polinomio en $\cos(\xi)$ por lo cual puede expresarse en términos de un polinomio $P(\sin^2\frac{\xi}{2}) = L(\xi)$, de donde haciendo $y = \cos^2\frac{\xi}{2}$, la expresión (3.2) se convierte en:

$$(1 - y)^N P(y) + y^N P(1 - y) = 1 \quad (3.6)$$

La cual tiene un único polinomio P que es su solución ([DAUB92], torema 6.1.1). De estos resultados es factible comprobar que la construcción de H_0 es factible, ya que la “raíz” de P puede ser calculada en base de una construcción hecha por Riesz¹ la cual se conoce como *factorización espectral*. Tal factorización puede ser hecha tomando en cuenta que $P(\xi)$ es invariante bajo el cambio $\xi \rightarrow -\xi$ y por lo tanto, si P es de grado M entonces su “raíz cuadrada” se puede calcular y es un polinomio de grado M de la forma:

$$p(x) = \sum_{m=0}^M a_m e^{im\xi} \quad (3.7)$$

La clave de la construcción está en establecer el grado del polinomio p . Para que la relación (3.6) se cumpla, el grado debe ser $M = N - 1$.

Para que (3.5) se cumpla, se debe cumplir ([GOSCHAN99]) que:

$$P(y) = P_N(y) + y^N R\left(\frac{1}{2} - y\right) \quad (3.8)$$

donde

$$P_N(y) = \sum_{k=0}^{N-1} \binom{N-1+k}{k} y^k \quad (3.9)$$

y R es un polinomio, escogido de tal manera que $P(y) \geq 0$ para $y \in [0, 1]$. La elección mas sencilla, se produce cuando $R \equiv 0$. Además, para que (3.1) se cumpla, basta tomar $N = L$.

¹Véase [DAUB92], pág. 172

3.1.1. Wavelet Haar.

Es fácil de observar que para $N = 1$, los coeficientes de (2.32) cuya suma está estandarizada a 1 son $h_0[0] = \frac{1}{2}$ y $h_0[1] = \frac{1}{2}$. La cual utilizando la secuencia de comandos:

```
z=swave(dbcoef(1),N=8) #Produce los valores de la Escala
z1=z[(round(z,8)/round(z[1],8))==1]/z[1]#Elige los valores distintos de 0
z2=z[(round(z,8)/round(z[1],8))!=1]/z[1]#Elige los valores 0
soporte1=seq(0,1,length=length(z1))
paso=soporte1[2]-soporte1[1]
soporte2=seq(from=1+paso,by=paso,length=length(z2))
plot(c(soporte1,soporte2),c(z1,z2),main='Escala Haar',xlab='Soporte Haar.',
ylab='Intensidad',col=4,type='l')
plot(z,main='Escala Haar',xlab='Soporte producido por el algoritmo',
ylab='Intensidad',col=4,type='l')
```

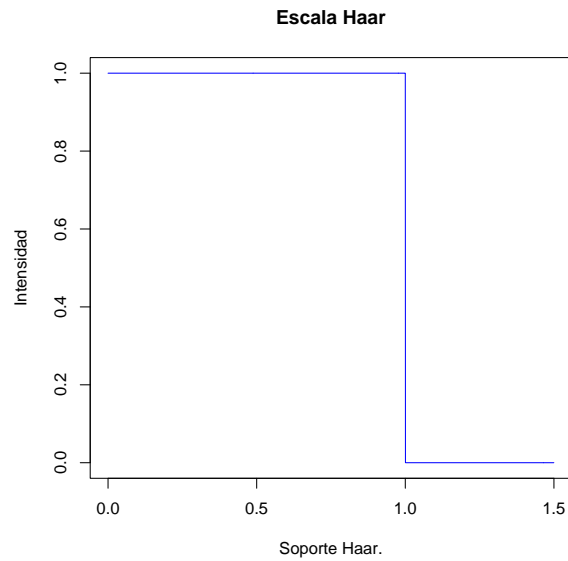
produce los gráficos mostrados en las figura 3.1.

El código de las funciones *swave* y *dbcoef* así como una explicación de su funcionamiento está adjuntada a los resultados.

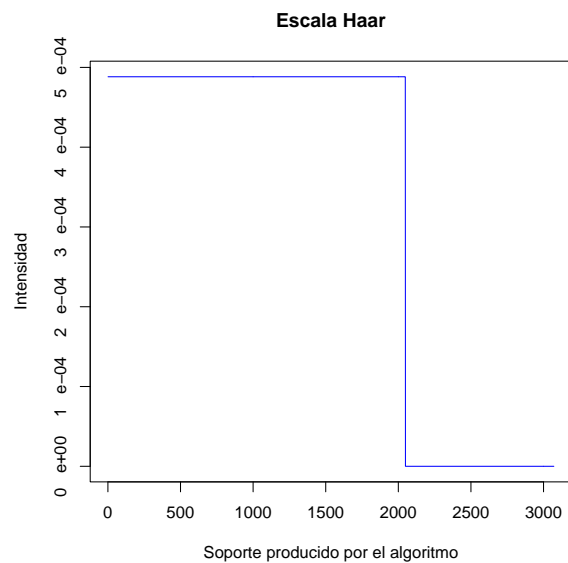
3.1.2. Generación de doublets.

Para el caso de generar Doublets, la situación es más interesante pues el polinomio ℓ en (3.4) debe ser construido, para el caso de acuerdo a las reglas de construcción del polinomio p en (3.7) que es raíz cuadrada de P . Para empezar observe algunas particularidades:

- El grado del polinomio H_0 es $2N - 1$, por lo que siempre existe un número de coeficientes par de (2.32).
- El polinomio será construido bajo el criterio de tomar la fase, lo cual implica tomar las raíces de L dentro del círculo unitario ([DAUB92]). Esto no implica simetría de la escala, pues la distorsión resulta ser la *menos lineal de las fases*, un tratamiento más detallado del tema se encuentra en [MONLUC99] y [STRANG00].



(a) Con el soporte.



(b) Sin el soporte.

Figura 3.1: Función de escala Haar

- El polinomio P produce $2N - 2$ raíces complejas la mitad dentro del círculo unitario, pues se puede comprobar de manera sencilla que si $e^{i\xi}$ es raíz de P , también lo son su conjugada $\overline{e^{i\xi}}$ y su inverso $e^{-i\xi}$.
- La construcción de los polinomios asegura que todos los coeficientes de (2.32) son reales.

En base de los puntos anotados anteriormente, se puede construir los coeficientes de ℓ primero construyendo los coeficientes de L en base de la fórmula dada en la ecuación (3.9), y utilizando el criterio dado en [GOSCHAN99], pág. 127. Luego de ello se eligen sus raíces dentro del círculo unitario y mediante el algoritmo de construcción de un polinomio en base de sus raíces (aplicado en la función *cpoly*) obtenemos los coeficientes del polinomio ℓ que convolucionados con los coeficientes del primer factor del polinomio dado en (3.4), nos permiten obtener los coeficientes de H_0 . La secuencia de comandos que permiten construir estos coeficientes se muestran en los resultados.

Las siguientes líneas calculan los coeficientes de una lafunción escala de Doubechies para $N = 2$.

```
> dbcoef(2)
[1] 0.34150635 0.59150635 0.15849365 -0.09150635
```

La siguiente secuencia de comandos permite obtener función escala de Doubechies para $N=2$ cuyos gráficos son los obtenidos en 3.2

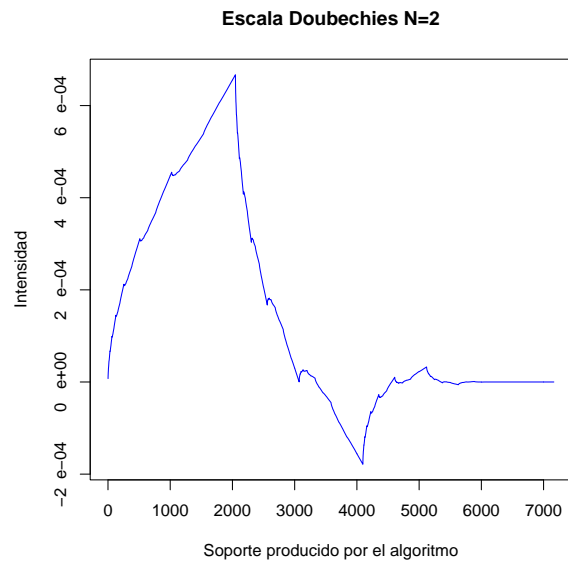
3.1.3. Generación de Symlets.

Las generación de los coeficientes de las Symlets se calculan de la misma manera que para las doublets, pero con el criterio de obtener coeficientes que proporcionen funciones más simétricas, pero como eso depende de la linealidad de la fase, se debe escoger las raíces del polinomio L como una configuración que proporcione la fase, digamos “más lineal”. No se puede conseguir filtros que proporcionen wavelets completamente simétricas, sin embargo, se puede tratar de conseguir la mayor simetría posible. Para ello, la fase de L puede ser calculada en base del conocimiento de que L puede ser expresado como el producto de factores del tipo $(z - z_j)(z - \bar{z}_j)$ cuando la raíz es compleja, y factores del tipo $(z - r_l)$ cuando la raíz es real. Para cada uno de estos factores se puede calcular la fase, de la siguiente manera de acuerdo a [DAUB92].

Tomamos $z_j = R_i e^{i\alpha_j}$, la forma polar del complejo z_j . Luego los factores de L pueden ser expresados como:



(a) Con el soporte.



(b) Sin el soporte.

Figura 3.2: Función de escala Doubechies N=2

1. En el caso complejo:

$$\begin{aligned}
 (e^{i\xi} - R_j e^{i\alpha_j})(e^{i\xi} - R_j \overline{e^{i\alpha_j}}) &= e^{i\xi}(1 - e^{-i\xi} R_j e^{i\alpha_j})(e^{i\xi} - R_j \overline{e^{i\alpha_j}}) \\
 &= e^{i\xi}(e^{i\xi} - 2R_j \cos(\alpha_j) + R_j^2 e^{-i\xi})
 \end{aligned}
 \tag{3.10}$$

2. En el caso real:

$$(e^{i\xi} - r_l) = e^{i\xi/2}(e^{i\xi/2} - e^{-i\xi/2}r_l) \quad (3.11)$$

De lo anterior se puede concluir que las fases correspondientes son:

$$\text{arc tg} \left(\frac{(1 - R_j^2) \sin(\xi)}{(1 + R_j^2) \cos(\xi) - 2R_j \cos(\alpha_j)} \right) \quad (3.12)$$

$$\text{arc tg} \left(\frac{r_l + 1}{r_l - 1} \tan \frac{\xi}{2} \right) \quad (3.13)$$

Solo se necesita estudiar la fase cuando $\xi \in [0, \phi]$, pues la fase es antisimétrica respecto de ϕ . La función *symcoef* toma en cuenta este detalle para el cálculo de la fase. Además, si $|z_j| < 1$ se tiene que $|z_j^{-1}| > 1$, ambas raíces de L , sus fases tienen signo cambiado, por lo que solamente hace falta estudiar las raíces cuyo módulo está menor que 1.

Por último, para elegir la configuración de fases que se necesita, basta analizar todas las posibles combinaciones de sumas y restas de las respectivas fases, y elegir la menor en valor absoluto. Debido a el cambio de signo que se produce de acuerdo al módulo de las raíces, los cálculos se reducen ampliamente, exactamente existen $2^{\frac{N}{2}-1}$ posibles elecciones, de donde tienen sentido estos cálculos para valores de $N \geq 4$.

Los coeficientes resultantes de aplicar la función *symcoef* para $N = 4$ son:

```
> symcoef(4)
```

```
[1] 0.02278517 -0.00891235 -0.07015881 0.21061727 0.56832912
0.35186953 [7] -0.02095548 -0.05357445
```

Con su suma estandarizada a 1.

La siguiente es una secuencia de comandos que reproducen los gráficos de la figura 3.3.

```
z=swave(symcoef(4)) #Produce los valores de la Escala
z2=z[(length(z)-2^10+1):length(z)]
z1=z[1:(length(z)-2^10)]
soporte1=seq(0,7,length=length(z1))
paso=soporte1[2]-soporte1[1]
soporte2=seq(from=7+paso,by=paso,length=length(z2))
```

```

plot(c(soporte1,soporte2),c(z1,z2)/paso,main='Escala Doubechies Symlet N=4',
xlab='Soporte Escala Doubechies Symlet N=4.',
ylab='Intensidad',col=4,type='l')
plot(z,main='Escala Doubechies Symlet N=2',xlab='Soporte producido por el algoritmo',
ylab='Intensidad',col=4,type='l')

```

3.1.4. Generación de Coiflets.

En el caso de la generación de coeficientes para las escalas de las Doublets y las Symlets, la relación (3.1) impone momentos nulos solamente para la ondeleta ψ , pero que tal si se exigen momentos un cierto número de momentos nulos también a la función escala, es decir:

$$\int_{-\infty}^{\infty} x^l \phi(x) dx = 0 \quad l = 1, 2, \dots, N-1 \quad (3.14)$$

Esta condición permite aumentar la simetría de las ondeletas obtenidas.

Debido a (3.14) (mire [DAUB92]) H_0 debe tomar la forma:

$$H_0(\xi) = 1 + \left(\sin^2 \frac{\xi}{e} \right)^N \left[- \sum_{k=0}^{N-1} \binom{N-1+k}{k} \left(\cos^2 \frac{\xi}{2} \right)^k + \left(\cos^2 \frac{\xi}{2} \right)^N f(\xi) \right] \quad (3.15)$$

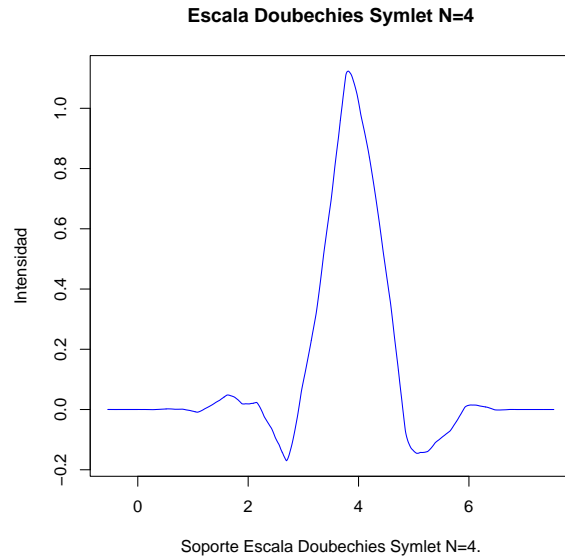
Donde f es un filtro causal de grado impar.

Hasta este momento la idea se ha manejado de la misma manera que en la construcción anterior, pero para el caso el filtro f tiene cierta dificultad para ser construido, así que tomaremos un camino alterno presentado en [WEIBOVEV98].

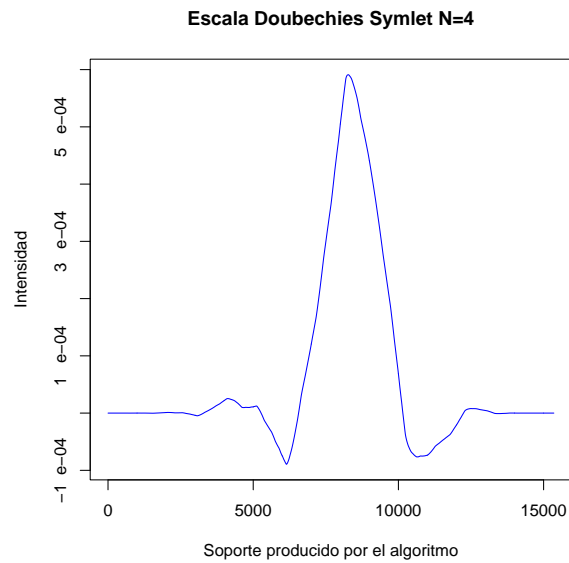
Definición: Una wavelet se llama *coiflet generalizada* de orden l , si para algún $\bar{t} \in \mathbb{R}$, la wavelet $\psi_{\bar{t}}$ y su función de escala $\phi_{\bar{t}}$ satisfacen:

$$\int_{\mathbb{R}} (t - \bar{t})^p \phi_{\bar{t}}(t) dt = \delta_p \quad (3.16)$$

$$\int_{\mathbb{R}} (t - \bar{t})^p \psi_{\bar{t}}(t) dt = 0 \quad (3.17)$$



(a) Con el soporte.



(b) Sin el soporte.

Figura 3.3: Función de escala Doubechies Symlet N=4

para $p = 0, 1, \dots, l - 1$.

\bar{t} se llama el centro de masa de la función de escala $\phi_{\bar{t}}$. Cuando $\bar{t} = 0$, las coiflets son las coiflets tradicionales construidas por Ingrid Doubechies. De acuerdo con (3.16) y (3.17)

se tiene que los coeficientes de H_0 deben satisfacer las condiciones siguientes:

$$\sum_{k \in \mathbb{Z}} k^p h_{\bar{t}}(k) = \sqrt{2\bar{t}}^p \quad (3.18)$$

$$\sum_{k \in \mathbb{Z}} (-1)^k k^p h_{\bar{t}}(k) = 0 \quad (3.19)$$

En base de las relaciones precedentes, las coiflets y las coiflets generalizadas pueden ser obtenidas resolviendo un sistema de ecuaciones no-lineales multivariado. Para ello se define el vector:

$$\mathbf{h}_{\bar{t}} \triangleq [h_{\bar{t}}(-l), \dots, h_{\bar{t}}(N_l - l - 1)]^T \quad (3.20)$$

donde $N_l = 3l$ y l es divisible para 2 (en otras palabras N_l es múltiplo de 6).

Para encontrar los coeficientes de H_0 , definimos $\mathbf{f}_l : \mathbb{R}^{N_l} \rightarrow \mathbb{R}^{N_l}$ una función real valuada, en la forma que se satisfacen los requerimientos [WEIBOVEV98]:

$$\mathbf{f}_l(\mathbf{h}_{\bar{t}}) \triangleq \begin{bmatrix} \sum_k h_{\bar{t}}^2(k) - 1 \\ \sum_k h_{\bar{t}}(k)h_{\bar{t}}(k+2) \\ \vdots \\ \sum_k h_{\bar{t}}(k)h_{\bar{t}}(k+N_l-2) \\ \sum_k kh_{\bar{t}}(k) - \sqrt{2\bar{t}} \\ \sum_k k^3 h_{\bar{t}}(k) - \sqrt{2\bar{t}}^3 \\ \vdots \\ \sum_k k^{N_l/3-1} h_{\bar{t}}(k) - \sqrt{2\bar{t}}^{N_l/3-1} \\ \sum_k (-1)^k h_{\bar{t}}(k) \\ \sum_k (-1)^k k h_{\bar{t}}(k) \\ \vdots \\ \sum_k (-1)^k k^{l-1} h_{\bar{t}}(k) \end{bmatrix} \quad (3.21)$$

donde el índice k de las sumatorias va desde $-l$ hasta $N_l - l - 1$, debido a que el soporte es el intervalo $[-l, N_l - l - 1]$.

Haciendo:

$$\mathbf{f}_l(\mathbf{h}_{\bar{t}}) = \mathbf{0}_{N_l} \quad (3.22)$$

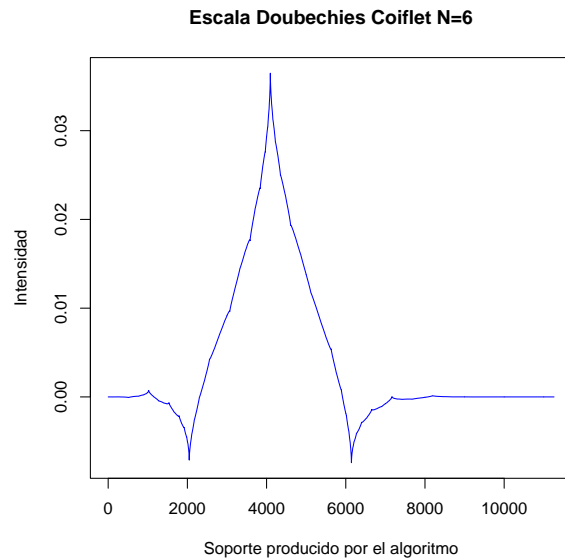


Figura 3.4: Coiflet N=6

obtenemos un sistema de ecuaciones no lineales cuadrado, el cual se puede resolver utilizando el método de Newton generalizado, o el método de la secante generalizada, en el caso del lenguaje R, se utiliza la función *nlm*, para minimizar la norma al cuadrado de \mathbf{f}_l . La siguiente secuencia de comandos proporciona el cálculo para $N_l = 6$ y $\bar{t} = 0$:

```
coif1j=nlm(optcoif,c(0,0,1,0,0,0))$estimate
z=swave(coif1j)
plot(z,main='Escala Doubechies Coiflet N=6',
xlab='Soporte producido por el algoritmo',
ylab='Intensidad',col=4,type='l')
coif1j
[1] -0.07273261  0.33789764  0.85257202  0.38486480 -0.07273261 -0.01565572
```

La escala obtenida a partir de los coeficientes encontrados es la mostrada en la figura 3.4. En la sección de resultados se mostrará el código de las funciones *optcoif* y *concoif* que sirven para encontrar los coeficientes. Para puntos de partida para las iteraciones se sugiere la lectura de [WEIBOVEV98].

Capítulo 4

Resultados.

4.1. Coeficientes y Gráficos de Doublets.

El comando *dbcoef* cuyo código su muestra a continuación recibe el número de momentos nulos de la doublet que se desea generar y crea en base a la factorización espectral los coeficientes de la función de escala (si *wave* es *FALSE*).

```
#dbcoef crea el filtro que genera las doublets de cualquier tamaño par (hasta
#la capacidad del R).
#el algoritmo encuentra las raíces del polinomio que genera un banco de filtros
#de dos bandas para reconstrucción perfecta.
dbcoef=function(N=2,wave=FALSE){ #el valor de N puede ser cualquier
#valor entero positivo
N=round(N)
if (N<1){
#Se comprueba que el valor sea entero positivo
print('Imposible encontrar coeficientes.')
print('Solo se aceptan valores enteros positivos mayores a 1.')
}
#Se inicia el proceso para encontrar los filtros
else{
K=N
```

```

a=1
p=1
qu=1
h_0=c(1,1)
#Generamos los coeficientes del polinomio
if(K!=1){
for(j in 1:(K-1)){
    a=-a*0.25*(j+K-1)/j
    h_0=c(0,h_0)+c(h_0,0)
    p=c(0,-p)+c(p,0)
    p=c(0,-p)+c(p,0)
    qu=c(0,qu,0)+a*p
}
#ordenamos las raices del polinomio en orden ascendente de acuerdo a
#a la parte real primero y luego ponemos las partes imaginarias negativas
#primero.
qu=Conj(sort(polyroot(qu[length(qu):1])))
qut=qut[1:(K-1)]#Toma solo las raices dentro del circulo unitario
#Se calcula los coeficientes por convolucion
h_0=convolve(h_0,rev(Re(cpoly(qut))),type='open')
#Se normaliza a la raiz de 2
h_0=h_0/sum(h_0)
#Procedimiento si la fase es máxima
if(wave)
    return(list(h0=h_0,h1=h_1))
else
    return(h_0)
}
else
    if(wave)

```

```

        return(list(h0=h_0/2,h1=c(h_0[1],-h_0[2])/2))
    else
        return(h_0/2)
}

}

```

Los coeficientes que se producen se muestran a continuación para los valores de momentos nulos de la doublet $N = 2, 3, 4, 5, 6$.

```

> dbcoef(2)
[1] 0.34150635 0.59150635 0.15849365 -0.09150635
> dbcoef(3)
[1] 0.23523360 0.57055846 0.32518250 -0.09546721 -0.06041610 0.02490875
> dbcoef(4)
[1] 0.162901714 0.505472858 0.446100069 -0.019787513 -0.132253584
[6] 0.021808150 0.023251801 -0.007493495
> dbcoef(5)
[1] 0.113209491 0.426971771 0.512163472 0.097883481 -0.171328358
[6] -0.022800566 0.054851329 -0.004413400 -0.008895935 0.002358714
> dbcoef(6)
[1] 0.0788712160 0.3497519070 0.5311318799 0.2229156615 -0.1599932994
[6] -0.0917590320 0.0689440465 0.0194616049 -0.0223318742 0.0003916256
[11] 0.0033780312 -0.0007617669

```

Las funciones escala y doublets respectivas para estos valores se muestran en la figuras 4.1 y 4.2. [!h]

4.2. Coeficientes y gráficos Symlets.

El programa mostrado a continuación calcula los coeficientes de la escala de las Symlets, como ejemplo se muestra los cálculos para momentos nulos $N = 4, 5, 6, 7$. Note que no hace

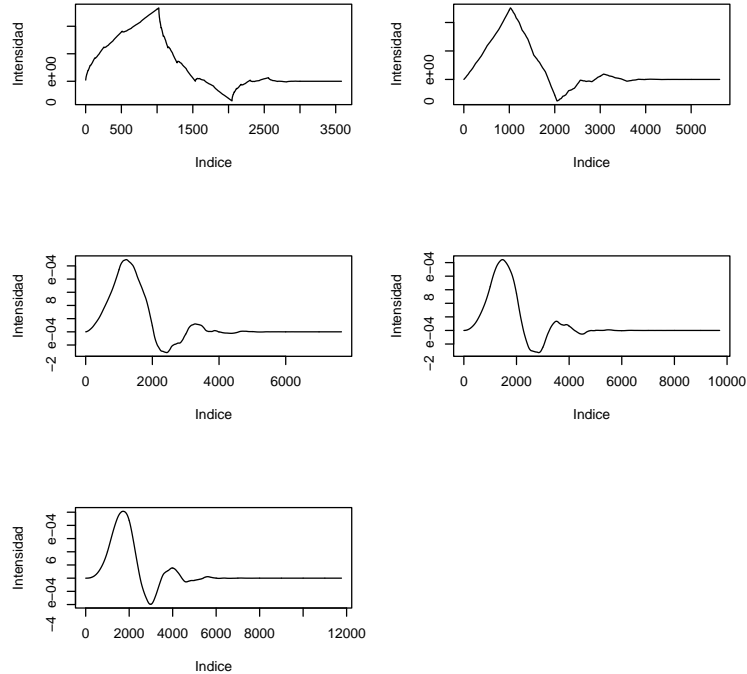


Figura 4.1: Escala de la Doublets. Momentos $N=2,3,4,5,6$

sentido los cálculos par $N, 4$.

```
#####
#####
#El siguiente programa calcula los coeficientes de las Symlets,
#basados en la elección de las raíces del polinomio m0 que proporciona
#la fase más próxima a la fase lineal.
#el algoritmo está basado en el cálculo del Cap. 8 del libro
#Ten lectures on wavelets de I. Daubechies (1992) y la ayuda abierta
#de MATHEMATICA para la construcción de wavelets.
symcoef=function(N=4){ #el valor de N puede ser cualquier valor entero positivo mayor que 3
N=round(N)
if (N<=3){
#Se comprueba que el valor sea entero positivo
print('Imposible encontrar coeficientes.')
print('Solo se aceptan valores enteros positivos mayores a 3.')
```

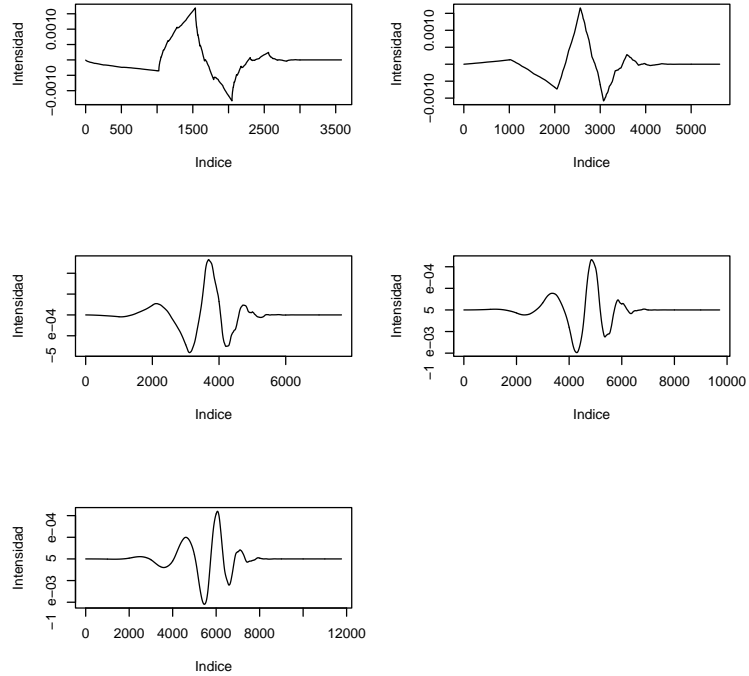


Figura 4.2: Doublets. Momentos $N=2,3,4,5,6$

```

}
#Se inicia el proceso para encontrar los filtros
else{
K=N
a=1
p=1
qu=1
h_0=c(1,1)
#Generamos los coeficientes del polinomio que va a dar origen
#origen a los coeficientes, igual que en las Doublets
for(j in 1:(K-1)){
  a=-a*0.25*(j+K-1)/j
  h_0=c(0,h_0)+c(h_0,0)
  p=c(0,-p)+c(p,0)
  p=c(0,-p)+c(p,0)
}

```



```

    qu=c(0,qu,0)+a*p
}
#ordenamos las raices del polinomio en orden ascendente de acuerdo a
#a la parte real primero y luego ponemos las partes imaginarias negativas
#primero.
qu=Conj(sort(polyroot(qu[length(qu):1])))
#A partir de este momento la construcción de la wavelet se base en la elección de las
#raíces que den la mejor aproximación a una fase lineal que es lo que tiene que ver con
#la simetría y asimetría.
qut=qu[1:(K-1)]#Toma solamente las raices dentre del circulo unitario

#####
#####
#####
#Vamos a calcular de manera cuidadosa las ra'ices reales y las complejas.
irrc=numeric()
for(i in 1:(K-1)){
    cer=sum(round(Conj(qut[i]),11)==round(qut[-i],11))
    if(cer!=0)
        irrc[i]=1
    else
        irrc[i]=0
}
qur=Re(qut[irrc==0])#escoge solamente las reales.
qui=qut[irrc==1]#escoge las imaginarias.
quc=qui[seq(2,length(qui),2)]#de las imaginarias toma la de imaginaria positiva
quc[Im(quc)<0]=Conj(quc[Im(quc)<0])
#Las siguientes sentencias crean la fases para cada raiz.
faser=function(rr,omega){#función para calcular la fase de las raíces reales

```

```

    if(length(rr)==1)
        return(atan(((1+rr)/(1-rr)*tan(omega/2)))
        #No retorna nada en el caso contrario
}
fasec=function(cc,omega){
    mcc=Mod(cc)#Módulo del complejo cc
    acc=Arg(cc)#Argumento del complejo cc
    return(atan(((1-mcc^2)*sin(omega))/((1+mcc^2)*cos(omega)-2*mcc*cos(acc))))
}
#ahora calculamos la desviación de la fase
dfaser=function(rr,omega){
    if(length(rr)==1)
        return(faser(rr,omega)-omega/2)
        #No retorna nada en el caso contrario
}
dfasec=function(cc,omega){
    dfase=numeric()
    confase=cos(omega)>=(2*Mod(cc)*cos(Arg(cc))/(1+Mod(cc)^2))
    dfase[confase==TRUE]=fasec(cc,omega[confase==TRUE])-omega[confase==TRUE]
    dfase[confase==FALSE]=fasec(cc,omega[confase==FALSE])-omega[confase==FALSE]+pi
    return(dfase)
}
#Calculamos la desviación de fase de todas las raíces elegidas.
omega=seq(0,pi,length=200)
fases=matrix(ncol=200,nrow=length(qur)+length(quc))
for(i in 1:(length(qur)+length(quc))){
    if(i<=length(qur))
        fases[i,]=dfaser(qur[i],omega)
    else
        fases[i,]=dfasec(quc[i-length(qur)],omega)
}

```

```

}
#Este rato se genera las combinaciones para saber que fases para elegir
#Matriz de unos y menos unos que permiten calcular todas las posibilidades
tam=length(qur)+length(quc)-1
combina=matrix(1,ncol=tam+1,nrow=2^tam)
for(i in 2:ncol(combina)){
  for(j in seq(2,nrow(combina)/2^(tam-i+1),2))
    combina[seq((j-1)*2^(tam-i+1)+1,j*2^(tam-i+1),1),i]=-1
}
#####
#Identificamos la configuracion que produce la mayor fase lineal
totales=combina%*%fases
tam=numeric()
tam=rowSums(abs(totales))
maxconf=sort(tam,index.return=TRUE)$ix[1] #Identifica la posicion de la mejor configuracion
qut=c(qur,quc)
qut[combina[maxconf,]==-1]=1/qut[combina[maxconf,]==-1]
qui=Conj(qut[Im(qut)!=0])
qut=c(qut,qui)#Concatenamos todas las ra'ices.
h_0=convolve(h_0,rev(Re(cpoly(qut))),type='open')
return(h_0/sum(h_0))
#return(symlet=list(combina=combina,fases=fases,totales=combina%*%fases,qut=qut,qur=qur,quc=quc))
}
}

```

Los coeficientes obtenidos son:

```

> symcoef(4)
[1] 0.02278517 -0.00891235 -0.07015881 0.21061727 0.56832912 0.35186953
[7] -0.02095548 -0.05357445

```

```

> symcoef(5)
[1] 0.01381608 -0.01492125 -0.12397568 0.01173946 0.44829082 0.51152648
[7] 0.14099535 -0.02767209 0.02087343 0.01932740
> symcoef(6)
[1] 0.010892350 0.002468306 -0.083431608 -0.034161561 0.347228986
[6] 0.556946392 0.238952186 -0.051362485 -0.014891876 0.031625281
[11] 0.001249961 -0.005515934
> symcoef(7)
[1] 0.007260697 0.002835671 -0.076231936 -0.099028353 0.204091970
[6] 0.542891355 0.379081301 0.012332830 -0.035039146 0.048007384
[11] 0.021577726 -0.008935216 -0.000740613 0.001896329

```

Los gráficos de las escalas y las correspondientes symlets son los mostrados en las figuras 4.3 y 4.4 respectivamente. [!h]

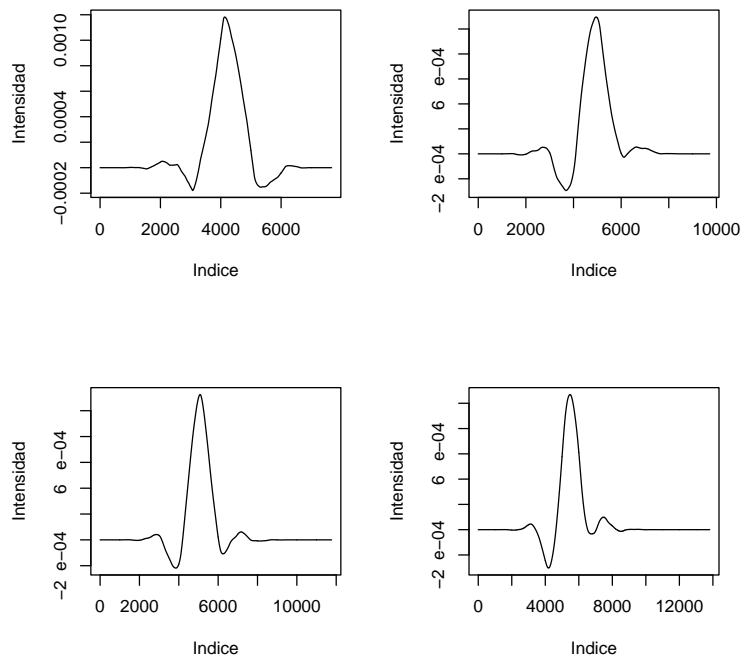


Figura 4.3: Escala de la Doublets. Momentos $N=2,3,4,5,6$

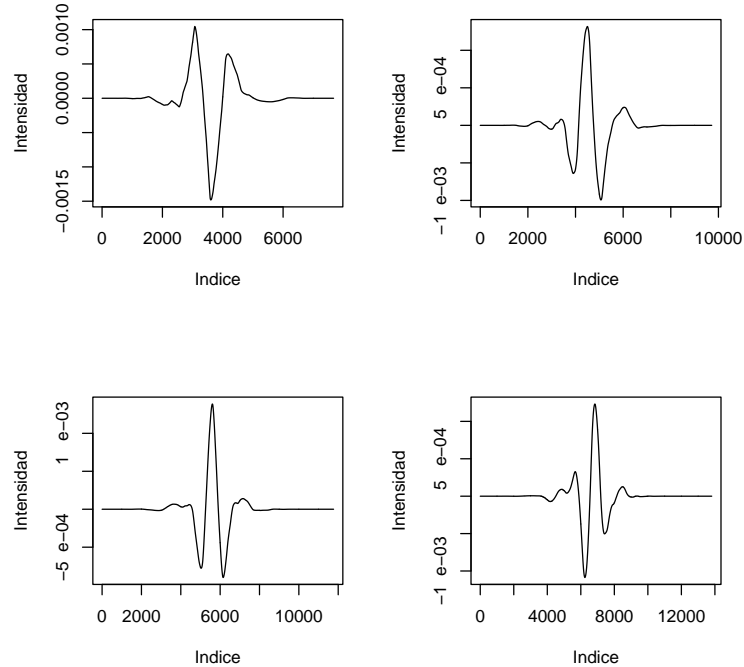


Figura 4.4: Doublets. Momentos $N=2,3,4,5,6$

4.3. Coiflets Generalizadas y gráficas.

El siguiente programa permite el cálculo de los coeficientes de la coiflets basados en las características de del filtro que se desea obtener. Primero se crean las condiciones y luego se utiliza la función *nlm* para resolver el sistema de ecuaciones, el punto de arranque está basa do en [WEIBOVEV98].

```
#Este programa genera el conjunto de coeficientes de las coiflets
generalizadas #basados en la minimización de una función escalar
multivariada #utilizando la función 'nlm' del R.
concoifg=function(x,tm=0){#Crea las condiciones de las coiflets
if(length(x)%6!=0){
  print('La longitud del argumento debe ser multiplo de 6.')
```

```
  return(0)
} else{
  N=length(x)
```

```

L=N/6
n=0:(N-1)
m=(-2*L):(N-1-2*L)
y=numeric()
z=numeric()

#Condicion de la norma del filtro
y[1]=sum(x*x)-1
#Condicion de ortogonalidad con la doble traslaci'on.
z=convolve(x,x,type='open')[seq(2,length(x)-1,2)]
y=c(y,z)
alterna=rep(1,length(x))
alterna[seq(2,length(x),2)]=-1
z=numeric()

#Condiciones de Desvanecimiento.
#2L momentos nulos para la madre
for(k in 1:L)#El soporte del padre está en n
  z[k]=sum(m^(2*k-1)*x)-sqrt(2)*tm^(2*k-1)
y=c(y,z)
z=numeric()

#2l momentos nulos para el padre
for(k in 0:(2*L-1))#El soporte de la madre está en m
  z[k+1]=sum(m^k*alterna*x)
y=c(y,z)
return(round(y,12))
}

}

```

La función a optimizar es:

```
#Esta es la función que se va a minimizar para encontrar los
valores de las coiflets #Está basado en la función cocoif.
optcoifg=function(x){
  return(sum(concoifg(x,0)^2))
}
```

Para obtener los coeficientes de la coiflet con el primer momento nulo junto con su función de escala, se hace:

```
> nlm(optcoifg,c(0,0,1,0,0,0))$estimate
[1] -0.07273261  0.33789764  0.85257202  0.38486480 -0.07273261 -0.01565572
```

La figura 4.5 nos muestra la coiflet que se obtiene

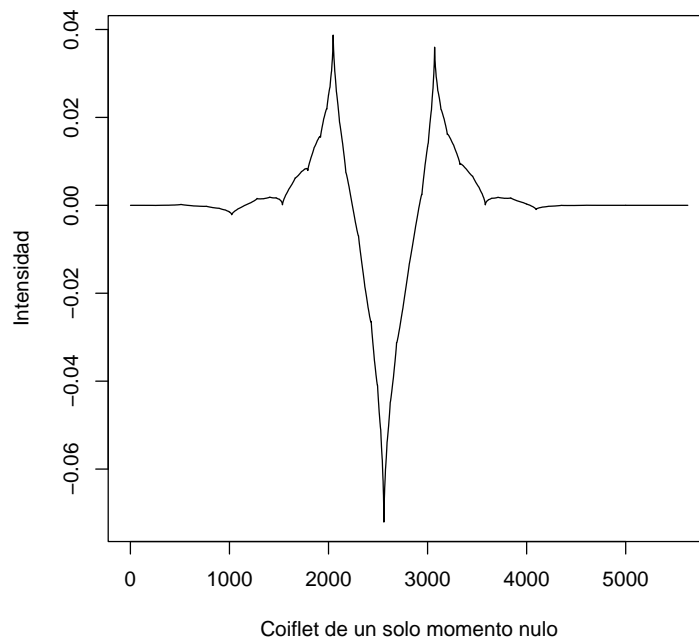


Figura 4.5: Coiflet.

Cambiando el valor de \bar{t} para obtener las coiflets generalizadas a partir de la obtenida, obtenemos los siguientes coeficientes de la escala.

$$\bar{t} = 0,2$$

[1] -0.10388672 0.26997504 0.84414686 0.44996683 -0.03317617
-0.01285686

$$\bar{t} = 0,4$$

[1] -0.134864275 0.216101708 0.835391684 0.487003498 0.006556881
[6] 0.003980176

$$\bar{t} = 0,6$$

[1] -0.1702192 0.1807454 0.8353906 0.4870054 0.0419125 0.0393345

$$\bar{t} = 0,8$$

[1] -0.20996894 0.16400694 0.84420188 0.44980336 0.07287379 0.09329634

Los gráficos de las escalas y coiflets generalizadas correspondientes a los coeficientes anteriores se muestran en la figura 4.6.

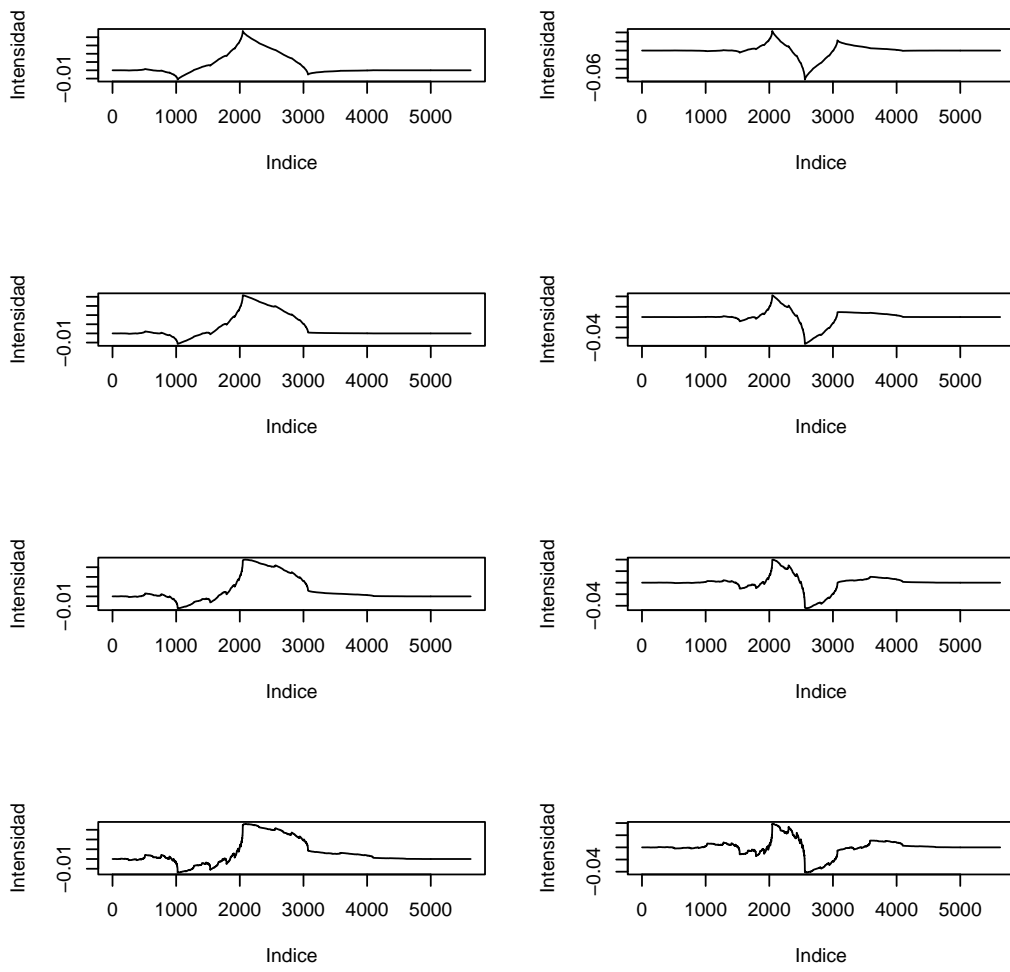


Figura 4.6: Escalas y coiflets generalizadas con un momento nulo.

Capítulo 5

Conclusiones.

1. La construcción de wavelets, puede ser vista como la construcción de filtros para los cuales decidimos sus características, y en base de ellas construimos la wavelet deseada.
2. El lenguaje de programación de R es una potente herramienta numérica que ha permitido explorar diferentes tipos de resultados en la construcción de los coeficientes presentados en este trabajo.
3. La búsqueda de wavelets más simétricas incrementa la complejidad del cálculo numérico, aunque la actual capacidad computacional de R permite hacerlos rápidamente.
4. Existen algoritmos numéricos, muy eficientes que permiten acelerar el cálculo de la convolución, como el de los polinomios de Galois, que no están implementados en R, la convolución implementada en R está basado en la transformada rápida de Fourier, que aunque es buena, es superada por la convolución hecha en base a la de los polinomios de Galois, que es la que usa Matlab, éste es un buen trabajo todavía por implementar.
5. Durante el trabajo se encontró que existen algoritmos que usan técnicas algébricas para resolver polinomios en varias variables, lo cual sería también útil implementar en R.
6. El paquete de R que se está construyendo se llama *cwavej*, y el mismo incluye todas

las funciones desarrolladas con sus respectivas ayudas, el cd adjunto incluye el archivo zip del paquete que se instala de manera estándar.

7. El trabajo continuará implementando algoritmos que permitan utilizar wavelet bio-ortogonales, se mejorará la función de descomposición para que presente automáticamente los análisis en el dominio de partida, también se mejorará la misma para que trabaje sin problemas con longitudes que no sean potencias de 2. Además, se espera mejorar el paquete para que muestre los espectros de las descomposiciones y extender todos los resultados a 2 dimensiones para realizar análisis de imágenes.

Bibliografía

- [QUIAN02] Quian Shie, Time-Frequency and Wavelet Transforms, Prentice-Hall, 2002.
- [PFAF02] Faundez Pablo y Fuentes Alvaro, Procesamiento Digital de Señales Acústicas utilizando Wavelets, Instituto de Matemáticas UACH.
- [STRANG89] Strang Gilbert, Artículo Wavelet and Dilation Equations: A Brief Introduction, Siam Review 1989.
- [VALENS04] Valens Clemens, Artículo A Really Friendly Guide to Wavelets, en <http://perso.wanadoo.fr/polyvalens> 2004.
- [GOSCHAN99] Goswami Jaideva y Chan Andrew, Fundamental of Wavelets Theory, Algorithms and Applications, Wiley Series 1999.
- [KREY78] Kreyszig Erwin, Introductory Functional Analysis with Application, Wiley Edc.1989.
- [DAUB92] Daubchies Ingrid, Ten Lectures on Wavelets, SIAM, Philadelphia, Pennsylvania, 1992.
- [STRANG00] Strang Gilbert, Artículo Signal Proccesing for Evryone, Siam Review 2000.
- [MORETTIN97] Morettin Pedro, Wavelets in Statistics, Institute of Mathematics and Statistics of Sao Paulo, 1997.
- [STRANG00C] Strang Gilbert, Artículo Convolution, Siam Review 2000.
- [X-GX92] X-G. Xia, “Topics in Wavelet transforms”, Ph.D dissertation, University of Southern California, 1992.

- [X-GX94] X-G. Xia, C-C. Jay Kuo, and Z. Zhang, “Wavelet coefficient computation with optimal prefiltering”, *IEEE Trans. Signal Processing*, vol 42, 1992.
- [MONLUC99] Monzón Lucas, A linear System and explicit solutions for approximate linear phase systems, Department of Applied Mathematics, University of Colorado, 1999.
- [WEIBOVEV98] Wei Dong, Vobik Alan y Evans Brian, Generalized Coiflets: A New Family of Orthonormal Wavelets, 1998.