

Capítulo 4

Funções de Walsh

4.1 Considerações iniciais

A utilização do modelo híbrido-misto apresentado no capítulo anterior permite uma enorme liberdade na escolha das funções de aproximação. Uma vez removida a restrição usualmente imposta pelo conceito de interpolação nodal, os campos de tensões e de deslocamentos podem ser modelados através da combinação linear de conjuntos completos de funções linearmente independentes.

A simplicidade e eficiência do modelo numérico resultante depende fortemente das funções de aproximação escolhidas. A utilização de funções com definição matemática simples (tais como as funções polinomiais) facilita, regra geral, o cálculo dos elementos das matrizes estruturais e as operações de pós-processamento. Pelo contrário, se se pretender verificar localmente algumas das condições do problema através da utilização de soluções clássicas da Teoria da Elasticidade, não só a definição das funções se complica, como também se torna mais moroso e delicado o processo de cálculo das matrizes estruturais. Em contrapartida, é natural que a utilização destas funções permita obter uma boa solução com um número de graus de liberdade significativamente inferior àquele que teria que ser considerado se se utilizassem apenas funções polinomiais.

A eficiência numérica do modelo híbrido-misto pode ser melhorada através da consideração de séries completas de funções ortogonais, como são exemplo as funções trigonométricas [66, 72, 147] e os polinómios de Legendre [147, 138]. A utilização de funções ortogonais simplifica consideravelmente as operações envolvidas no cálculo dos elementos das matrizes estruturais. Permite, regra geral, a obtenção de soluções analíticas para os integrais envolvidos no processo de cálculo, tornando desnecessário o recurso a quaisquer esquemas de integração numérica. Este facto pode ser relevante, não só pela economia que proporciona em termos de tempo de cálculo, mas também pela garantia que assegura no que diz respeito à minimização da ocorrência de erros numéricos associados ao cálculo dos integrais.

A utilização de séries hierárquicas ortogonais é também importante quando se pretende efectuar uma análise baseada em processos p -adaptativos, quando o refinamento da solução é conseguido à custa do aumento do número de termos na série utilizada na aproximação, sem no entanto se alterar o número de elementos finitos considerado à partida. A ortogonalidade existente entre as funções permite que as matrizes estruturais possam ser recalculadas de uma forma bastante eficaz, sendo possível maximizar o número de coeficientes que permanecem inalterados em relação à análise anterior.

É no campo da Teoria dos Sinais que as funções trigonométricas têm vindo a desempenhar um papel de grande relevo, sendo a análise de Fourier uma das ferramentas matemáticas com maior aplicação naquele domínio. No entanto, a crescente capacidade e potência de cálculo da computação digital tem favorecido o aparecimento de outros sistemas de funções ortogonais que têm vindo a ser utilizados com assinalável sucesso, substituindo mesmo em muitas aplicações as tradicionais séries trigonométricas. São precisamente dois destes sistemas alternativos que são utilizados neste trabalho para modelar os campos de tensões e de deslocamentos quando se aplica a formulação híbrida-mista apresentada na análise elástica e elastoplástica de problemas planos: as *séries digitais de Walsh* [192] e as famílias de *séries de wavelets* propostas por Daubechies [52].

As séries digitais de Walsh são constituídas por funções que podem tomar apenas dois valores possíveis, +1 e -1. Uma das grandes vantagens existentes na sua utilização resulta da facilidade com que tais funções podem ser geradas e manipuladas. Além disso, todas as multiplicações envolvendo funções de Walsh podem ser reduzidas a uma sequência apropriada de trocas de sinal. Refira-se que para além das séries de Walsh, existem outros sistemas de funções digitais que merecem destaque pela sua importância no domínio do processamento de sinais. É o caso das séries de Haar [94] e de Rademacher [171]. Embora não sejam aqui aplicadas, importa referir que as funções de Rademacher podem desempenhar um papel importante na geração de funções de Walsh. Por outro lado, as funções de Haar constituem uma série que, tal como se verá no seguimento, corresponde à série de wavelets de grau menos elevado.

As potencialidades das séries de wavelets propostas por Daubechies [52] há bem poucos anos, foram rapidamente reconhecidas. Como consequência, estas funções têm vindo a ser cada vez mais utilizadas no domínio do processamento de sinais e de imagens. Em relação às funções trigonométricas, têm a vantagem de apresentar um comportamento local tanto no domínio da frequência como no domínio do tempo, o que se traduz numa assinalável vantagem em muitas situações típicas da análise de sinais. Embora a geração e a manipulação das séries de Walsh seja mais simples, as propriedades de que as wavelets gozam tornam bastante atractiva a sua utilização como funções de aproximação em modelos híbrido-mistos de elementos finitos.

Efectua-se, neste capítulo, a apresentação das funções e séries de Walsh. Discutem-se alguns métodos alternativos de geração e introduzem-se algumas das suas propriedades mais importantes. Indica-se ainda a forma através da qual se pode exprimir

uma dada função como combinação linear de um conjunto completo de funções de Walsh. A apresentação dos sistemas de wavelets é efectuada no capítulo seguinte.

4.2 Funções e séries de Walsh

As séries digitais de Walsh [192] foram apresentadas em 1923. Foi no entanto apenas há pouco tempo que a sua importância foi reconhecida e a sua utilização generalizada. Hoje em dia é bastante vasto o seu campo de aplicações [18], cobrindo áreas tão distintas como a espectroscopia, a sismologia, o processamento e reconhecimento de voz, a medicina e a detecção remota (radares e sonares).



Figura 4.1: *Processamento de imagens utilizando séries de Walsh (extraído de Beauchamp).*

Na figura 4.1 ilustra-se outra das importantes aplicações das séries de Walsh, o processamento e tratamento de imagens. Interessante é verificar a forma como a imagem é sucessivamente refinada à custa da consideração de um número sempre crescente de termos na série utilizada para o processamento. Refira-se que foi precisamente a utilização das funções de Walsh em processamento de imagens que motivou a sua recente aplicação em modelos híbrido-mistos de elementos finitos [67, 74, 30, 32].

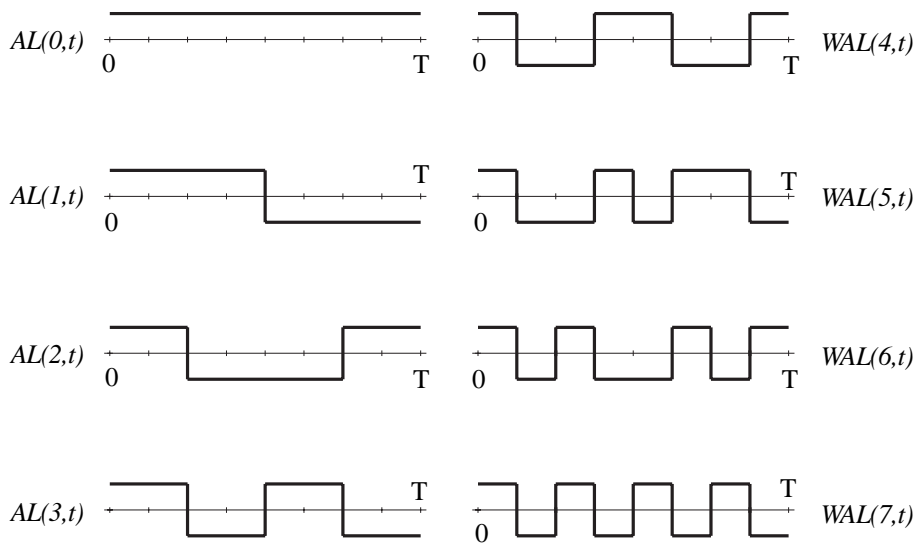


Figura 4.2: *Funções de Walsh ordenadas sequencialmente.*

As funções de Walsh formam um conjunto ordenado de ondas retangulares apresentando apenas duas amplitudes possíveis, $+1$ e -1 . São definidas sobre um intervalo limitado e, tal como no caso das funções trigonométricas, são necessários dois parâmetros para a sua completa identificação. As funções de Walsh serão denotadas no seguimento por $WAL(n, t)$, onde t indica a posição no intervalo de definição e n é um número de ordem que se encontra relacionado com a frequência. No seu artigo, Walsh [192] propõe que para uma dada função, n seja igual ao número total de mudanças de sinal existentes.

As funções de Walsh são ortogonais entre si e formam uma série completa a que se dá o nome de série de Walsh. As propriedades destas funções e séries são discutidas adiante mais detalhadamente. Na figura 4.2 apresentam-se as primeiras oito funções da série de Walsh.

A generalização das funções de Walsh a domínios planos é imediata. Na figura 4.3 apresentam-se as 64 funções bidimensionais obtidas a partir do produto cartesiano das oito funções unidimensionais representadas na figura 4.2. As zonas a negro indicam amplitudes com valor $+1$, enquanto que as zonas a branco representam amplitudes valendo -1 .

Uma notação alternativa para as funções de Walsh, permitindo a sua classificação em termos de simetria par e ímpar, foi introduzida por Harmuth [95]. São definidas duas séries, a série CAL e a série SAL , as quais apresentam bastantes analogias com as séries trigonométricas $COSENO$ e $SENO$, respectivamente. As funções $CAL(k, t)$ são simétricas em relação ao ponto médio do intervalo de definição, enquanto que as funções $SAL(k, t)$ são anti-simétricas em relação a esse mesmo ponto. Define-se

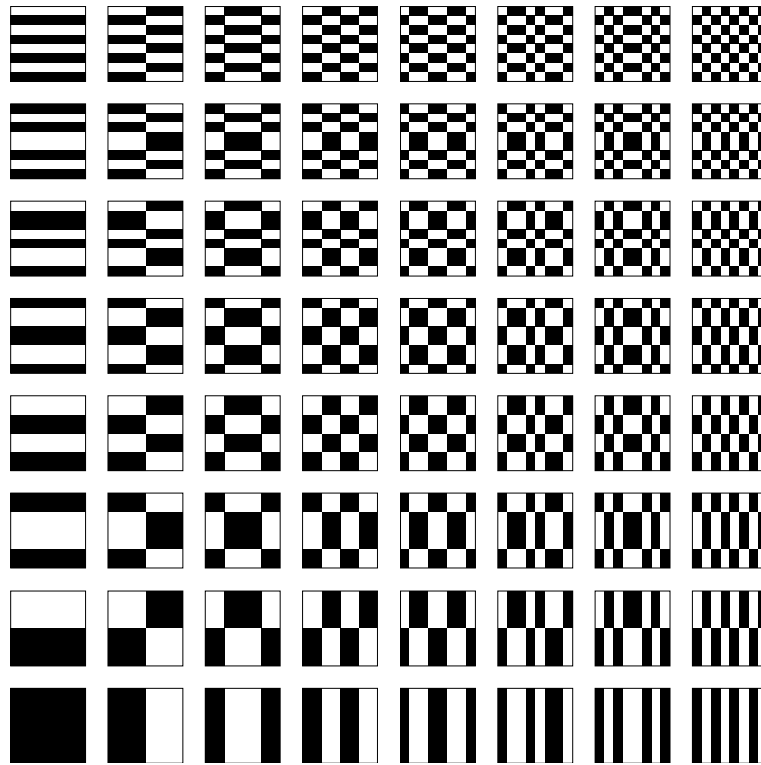


Figura 4.3: *Funções de Walsh bidimensionais.*

então:

$$CAL(k, t) = WAL(2k, t) \quad (4.1)$$

$$SAL(k, t) = WAL(2k - 1, t) \quad (4.2)$$

Nas igualdades anteriores, o parâmetro k é definido como sendo o maior inteiro menor ou igual a $(n + 1)/2$, onde n denota o número total de mudanças de sinal da função em causa. Na figura 4.4 apresentam-se as quatro primeiras funções das séries CAL e SAL .

4.3 Ordenação das funções de Walsh

Tal como Yuen [200] salienta, a natureza matemática das funções de Walsh é tal que não permite a existência de uma forma única para efectuar a sua ordenação. Este facto contrasta com o que sucede no caso das funções exponenciais complexas, que são sempre ordenadas de acordo com a sua frequência. Isto sucede porque as exponenciais complexas são *funções características* do grupo dos inteiros com adição *módulo* N , que tem uma ordenação natural: por valor aritmético. Geometricamente, podem considerar-se os inteiros como sendo pontos pertencentes à recta dos reais. Tendo esta última apenas uma dimensão, existe apenas um modo possível para

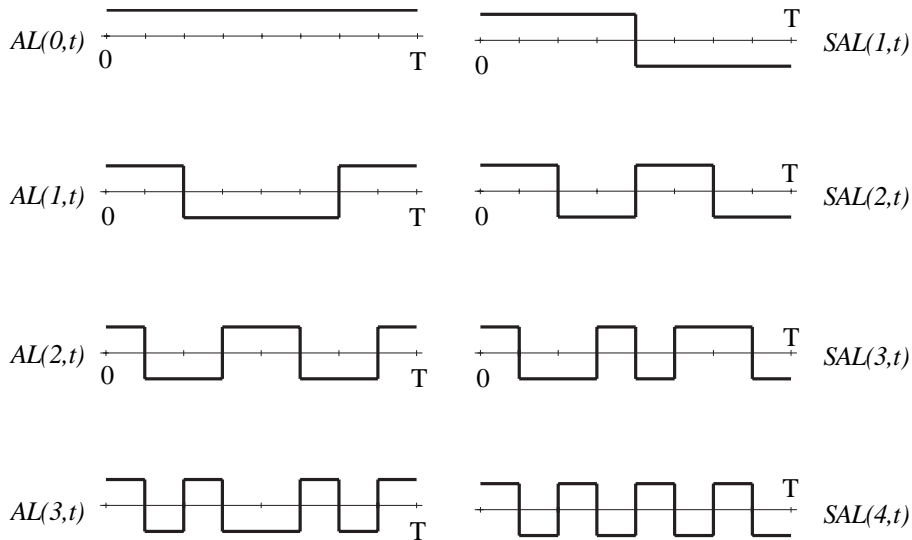


Figura 4.4: Quatro primeiras funções das séries *CAL* e *SAL*.

ordenar os inteiros. Por outro lado, as funções de Walsh são *funções características* do grupo diádico [178], que é o grupo dos vectores binários com adição *módulo 2* (a definir no seguimento). O espaço formado por esses vectores, usualmente conhecido na literatura como espaço diádico, tem várias dimensões mas apenas dois pontos ao longo de cada eixo. Uma vez que não existe uma forma única de ordenação no espaço diádico, o mesmo acontecerá para as funções de Walsh.

São três os principais tipos de ordenação geralmente considerados [18, 200]. O primeiro é conhecido como *Ordenação Sequencial*, o segundo é denominado *Ordenação Diádica* e o terceiro é chamado *Ordenação Natural*. A cada uma destas formas de ordenar as funções de Walsh estão associadas algumas vantagens e inconvenientes. A escolha do tipo de ordenação a utilizar depende muito do tipo de aplicação que se pretenda efectuar.

A ordem sequencial, já referida anteriormente, foi proposta por Walsh [192]. As funções são ordenadas segundo o número de mudanças de sinal verificado em cada uma delas. Embora seja a ordenação que maiores semelhanças apresenta em relação à ordenação seguida noutros sistemas de funções ortogonais (nomeadamente no caso das funções trigonométricas), não é a mais adequada para tratamento matemático e computacional. As funções de Walsh representadas na figura 4.2 encontram-se ordenadas sequencialmente.

A ordem diádica é obtida quando se geram as funções de Walsh a partir de sucessivos produtos de funções de Rademacher [144, 171]. Foi introduzida por Paley [144] e é utilizada fundamentalmente para tratamento matemático. Apresentam-se na figura 4.5 as primeiras oito funções de Walsh dispostas segundo a ordem diádica. Tais funções são denotadas no seguimento por $PAL(n, t)$.

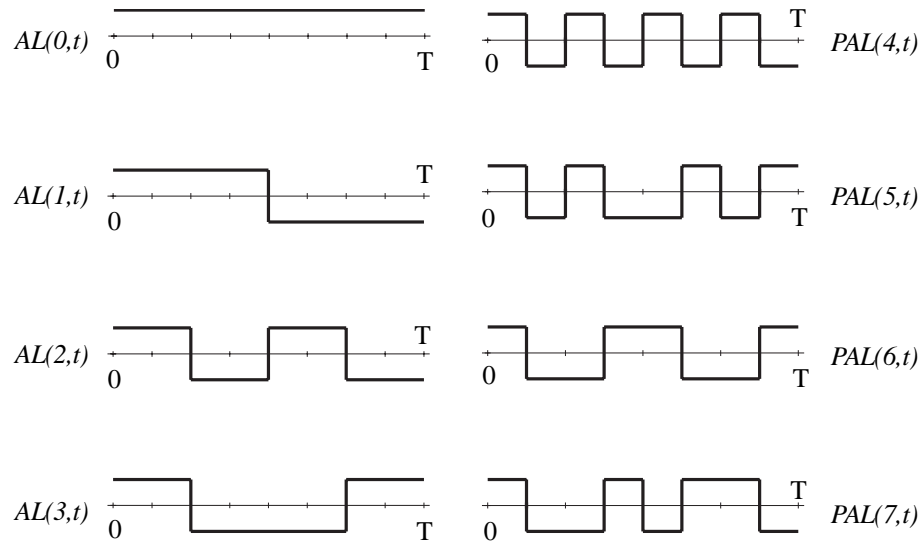


Figura 4.5: Funções de Walsh dispostas segundo a ordem diádica.

A ordem natural, proposta por Henderson [97], baseia-se na matriz de Hadamard, a qual será definida mais adiante. É a ordenação que oferece mais vantagens de um ponto de vista computacional. Na figura 4.6 apresentam-se as oito primeiras funções de Walsh dispostas segundo a ordem natural. Estas funções são denotadas no seguimento por $HAD(n, t)$.

É importante, em determinadas situações, conseguir relacionar as funções componentes das séries que se obtêm aplicando os três sistemas de ordenação atrás descritos. É o que sucede quando se pretendem calcular os coeficientes da expansão em série de Walsh de uma dada função através da aplicação de transformadas do tipo FWT (*Fast Walsh Transform*).

As funções WAL , PAL e HAD são relacionadas através da aplicação de uma operação de *inversão de bits* (*bit-reversal*) à posição de cada uma das funções constituintes das séries, através de uma conversão aplicando o código de Gray [39], ou ainda através da conjugação das duas operações. Convém, antes de mais, indicar o modo pelo qual se podem efectuar as operações de *inversão de bits* e a aplicação do código de Gray. Seja

$$(n)_b = b_k b_{k-1} \dots b_2 b_1 b_0 \tag{4.3}$$

a representação binária (*bit-string*) do inteiro n , de tal forma que,

$$n = \sum_{i=0}^k b_i 2^i, \tag{4.4}$$

com $b_i = 0$ ou 1 . A aplicação de uma operação de inversão de bits conduz a um inteiro u , cuja representação binária é dada por:

$$(u)_b = (n)_{br} = r_k r_{k-1} \dots r_2 r_1 r_0, \tag{4.5}$$

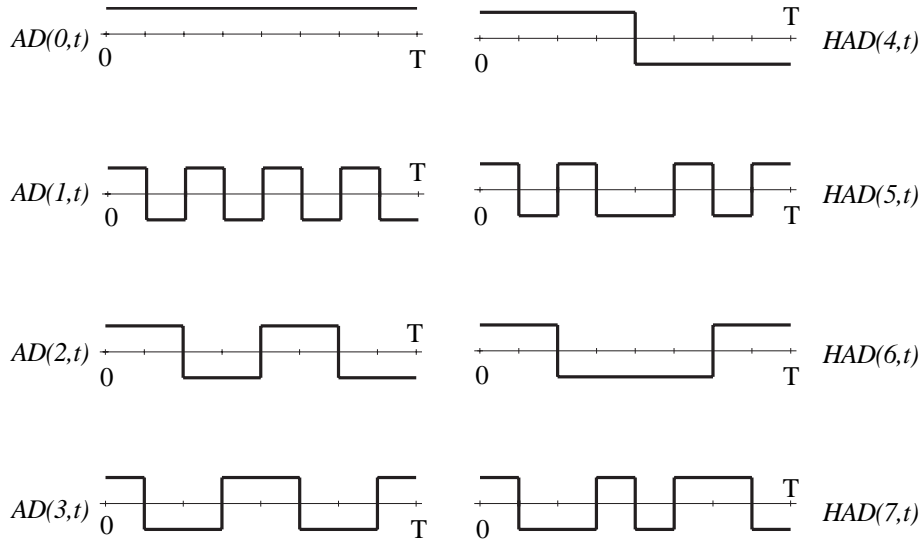


Figura 4.6: *Funções de Walsh dispostas segundo a ordem natural.*

e onde se tem que

$$r_k = b_0, r_{k-1} = b_1, \dots, r_i = b_{k-i}, \dots, r_0 = b_k. \quad (4.6)$$

Quando se aplica ao mesmo inteiro n a conversão binário-código de Gray obtém-se

$$(m)_b = (n)_g = g_k g_{k-1} \dots g_2 g_1 g_0,$$

em que os valores de g_i são obtidos a partir de,

$$(n)_g = b_k (b_k \oplus b_{k-1}) \dots (b_3 \oplus b_2) (b_2 \oplus b_1) (b_1 \oplus b_0), \quad (4.7)$$

e onde é utilizada a adição *módulo 2*. Esta última pode ser definida do seguinte modo:

$$0 \oplus 1 = 1 \oplus 0 = 1, 0 \oplus 0 = 1 \oplus 1 = 0. \quad (4.8)$$

A conversão código de Gray-binário pode ser vista como a operação inversa da transformação definida em (4.7). Seja então m um inteiro com representação binária:

$$(m)_b = m_k m_{k-1} \dots m_2 m_1 m_0.$$

A conversão código de Gray-binário é definida por,

$$(n)_b = (m)_{g^{-1}} = b_k b_{k-1} \dots b_2 b_1 b_0, \quad (4.9)$$

com

$$b_i = m_i \oplus m_{i+1} \oplus \dots \oplus m_k \quad i = 0, \dots, k, \quad (4.10)$$

ou, em alternativa,

$$b_{i-1} = b_i \oplus m_{i-1}. \quad (4.11)$$

É de salientar que a aplicação dos códigos de Gray pode ser programada em Fortran de uma forma bastante simples e eficiente [169].

Seja $PAL(n, t)$ uma função de Walsh com ordenação diádica. A função correspondente nas ordenações sequencial e natural, respectivamente, pode ser obtida através de:

$$PAL(n, t) = WAL(b(n), t) , \quad (4.12)$$

$$PAL(n, t) = HAD(u, t) . \quad (4.13)$$

Nas igualdades anteriores, $b(n)$ denota a aplicação da conversão código de Gray-binário ao inteiro n , enquanto que o inteiro u é obtido efectuando a inversão de bits a n . Na tabela 4.1, exemplifica-se a aplicação destas transformações e relacionam-se os oito primeiros termos da série PAL com os oito primeiros termos das séries WAL e HAD .

Ordem diádica	n	$(n)_b$	<i>Bit reversal</i>	Ordem natural	Código de gray-binário	Ordem sequencial
$PAL(0, t)$	0	000	000	$HAD(0, t)$	000	$WAL(0, t)$
$PAL(1, t)$	1	001	100	$HAD(4, t)$	001	$WAL(1, t)$
$PAL(2, t)$	2	010	010	$HAD(2, t)$	011	$WAL(3, t)$
$PAL(3, t)$	3	011	110	$HAD(6, t)$	010	$WAL(2, t)$
$PAL(4, t)$	4	100	001	$HAD(1, t)$	111	$WAL(7, t)$
$PAL(5, t)$	5	101	101	$HAD(5, t)$	110	$WAL(6, t)$
$PAL(6, t)$	6	110	011	$HAD(3, t)$	100	$WAL(4, t)$
$PAL(7, t)$	7	111	111	$HAD(7, t)$	101	$WAL(5, t)$

Tabela 4.1: *Relação entre a série PAL e as séries WAL e HAD.*

Seja agora $HAD(n, t)$ uma função de Walsh com ordenação natural. A função correspondente nas ordenações diádica e sequencial, respectivamente, pode ser obtida através de:

$$HAD(n, t) = PAL(u, t) , \quad (4.14)$$

$$HAD(n, t) = WAL(b_r(n), t) . \quad (4.15)$$

Na igualdade (4.14), u continua a resultar de uma operação de inversão de bits sobre o inteiro n . Na expressão (4.15), $b_r(n)$ denota a aplicação de uma conversão código de Gray-binário sobre o inteiro u . A tabela 4.2 exemplifica a aplicação das transformações envolvidas na aplicação das igualdades (4.14) e (4.15) e relaciona os oito primeiros termos da série HAD com os oito primeiros termos das séries WAL e PAL .

Por fim, considere-se uma função de Walsh com ordenação sequencial. Tendo em atenção as igualdade (4.12) e (4.15), as funções correspondentes nas séries com ordenação diádica e natural podem ser obtidas através de:

$$WAL(n, t) = PAL(g(n), t) , \quad (4.16)$$

$$WAL(n, t) = HAD(g_r(n), t) . \quad (4.17)$$

Ordem natural	n	$(n)_b$	Bit reversal	Código gray-binário	Ordem sequencial	Bit reversal	Ordem diádica
$HAD(0, t)$	0	000	000	000	$WAL(0, t)$	000	$PAL(0, t)$
$HAD(1, t)$	1	001	100	111	$WAL(7, t)$	100	$PAL(4, t)$
$HAD(2, t)$	2	010	010	011	$WAL(3, t)$	010	$PAL(2, t)$
$HAD(3, t)$	3	011	110	100	$WAL(4, t)$	110	$PAL(6, t)$
$HAD(4, t)$	4	100	001	001	$WAL(1, t)$	001	$PAL(1, t)$
$HAD(5, t)$	5	101	101	110	$WAL(6, t)$	101	$PAL(5, t)$
$HAD(6, t)$	6	110	011	010	$WAL(2, t)$	011	$PAL(3, t)$
$HAD(7, t)$	7	111	111	101	$WAL(5, t)$	111	$PAL(7, t)$

Tabela 4.2: Relação entre a série HAD e as séries WAL e PAL .

Na igualdade (4.16), $g(n)$ denota a aplicação da conversão binário-código de Gray sobre o inteiro n . Em (4.17), $g_r(n)$ resulta da aplicação de uma inversão de bits sobre $g(n)$. Na tabela 4.3 exemplifica-se a aplicação das igualdades (4.16) e (4.17) e relacionam-se os oito primeiros termos da série WAL com os oito primeiros termos das séries PAL e HAD .

Ordem sequencial	n	$(n)_b$	Binário cód. gray	Ordem diádica	Binário cód. gray	Bit reversal	Ordem natural
$WAL(0, t)$	0	000	000	$PAL(0, t)$	000	000	$HAD(0, t)$
$WAL(1, t)$	1	001	001	$PAL(1, t)$	001	100	$HAD(4, t)$
$WAL(2, t)$	2	010	011	$PAL(3, t)$	011	110	$HAD(6, t)$
$WAL(3, t)$	3	011	010	$PAL(2, t)$	010	010	$HAD(2, t)$
$WAL(4, t)$	4	100	110	$PAL(6, t)$	110	011	$HAD(3, t)$
$WAL(5, t)$	5	101	111	$PAL(7, t)$	111	111	$HAD(7, t)$
$WAL(6, t)$	6	110	101	$PAL(5, t)$	101	101	$HAD(5, t)$
$WAL(7, t)$	7	111	100	$PAL(4, t)$	100	001	$HAD(1, t)$

Tabela 4.3: Relação entre a série WAL e as séries PAL e HAD .

4.4 Geração das funções de Walsh

São vários os métodos que permitem efectuar a geração das funções de Walsh. Os mais importantes são os seguintes:

- Aplicação da fórmula de recorrência de Harmuth [95];
- Aplicação da fórmula de Chien [35];
- Produtos sucessivos de funções de Rademacher [114, 51];

- Montagem da matriz de Hadamard [97].

Embora não sejam apresentados neste trabalho, existem outros processos de geração que têm vindo a ser propostos para situações particulares. É o caso da geração de funções através da aplicação de uma síntese booleana [18, 35]. Este método assume especial importância quando se pretendem obter séries digitais de Walsh a partir do desenvolvimento de hardware específico.

4.4.1 Aplicação da fórmula de Harmuth

As funções de Walsh obtidas a partir da utilização da fórmula de recorrência de Harmuth [95, 178] seguem a ordenação sequencial. Para se efectuar a geração, considera-se o intervalo de definição normalizado, com $0 \leq t \leq 1$.

Define-se à partida a função $WAL(0, t)$, a qual toma um valor constante e igual a +1 em todo o intervalo $[0, 1]$ e que se anula para pontos fora deste. As funções de ordem superior são obtidas a partir da definição das anteriores através da fórmula de recorrência introduzida por Harmuth [95]. Esta pode ser escrita na forma

$$WAL(2j + q, t) = (WAL(j, 2t) + (-1)^{(j+q)}WAL(j, 2t - 1)) , \quad (4.18)$$

onde $j = 0, 1, \dots$ e $q = 0$ ou 1 . Sem formalismos matemáticos, a equação (4.18) pode ser interpretada da seguinte forma:

1. Quando se pretende gerar a função de grau $2n$, toma-se a função de grau n obtida anteriormente e *comprime-se* esta de forma a ficar definida no intervalo $[0, 0.5]$. O andamento da função $WAL(2n, t)$ no intervalo $[0.5, 1]$ deve ser tal que se verifique simetria par em relação ao ponto médio do intervalo de definição, $t = 0.5$.
2. A função de grau $2n + 1$ é gerada a partir da função de grau $2n$, invertendo o sinal desta última no intervalo $[0.5, 1]$ de modo a obter-se simetria ímpar em relação a $t = 0.5$.

Considerem-se agora N pontos (com $N = 2^p$, sendo p um inteiro qualquer) igualmente espaçados. A determinação do valor das funções de Walsh em cada um deles pode ser efectuada através da seguinte fórmula, obtida a partir de (4.18):

$$WAL(2j + q, n) = (WAL(j, 2n) + (-1)^{(j+q)}WAL(j, 2n - N)) . \quad (4.19)$$

Na igualdade anterior, $n = 0, 1, \dots, N - 1$; $j = 0, 1, \dots, N - 1$; e $q = 0$ ou 1 . A desvantagem do método de geração de Harmuth tem a ver com o facto de se tratar de um processo de recorrência. Como tal, quando se pretende gerar uma função de um determinado grau, torna-se necessário determinar primeiro as funções de ordem inferior. No entanto, tendo em conta a sua simplicidade, a fórmula de Harmuth continua ainda a ser bastante utilizada [18], especialmente quando se pretendem gerar séries completas de funções de Walsh com ordem sequencial.

4.4.2 Aplicação da fórmula de Chien

Para contornar as dificuldades associadas à utilização de uma fórmula de recorrência, Chien [35] propôs uma equação que permite determinar explicitamente cada uma das funções de Walsh. Deste modo, a geração da função $WAL(n, t)$ é efectuada directamente, sem haver a necessidade de previamente se calcularem as funções de grau inferior. A equação que permite gerar independentemente cada uma das funções de Walsh é dada por [35]

$$WAL(n, t) = (-1)^{\{g_k 2^{k+1} t\} + \{g_{k-1} 2^k t\} + \dots + \{g_1 2^2 t\} + \{g_0 2 t\}}, \quad (4.20)$$

onde $g_i = 0$ ou 1 , com $i = 0, 1, \dots, k$. Os valores de g_i são obtidos aplicando a conversão binário-código de Gray sobre o inteiro n . Na fórmula anterior, o termo $\{val\}$ representa o maior inteiro menor ou igual a val . Esta notação será seguida até ao final do capítulo. Pode demonstrar-se [35] que utilizando a fórmula de Chien as funções de Walsh são geradas segundo a ordenação sequencial.

Para ilustrar a aplicação da definição (4.20), calcula-se de seguida o valor que se obtém para $WAL(n, t)$ quando se considera, como exemplo, $n = 14$ e $t = 0.2$. Sendo a representação binária do inteiro 14 dada por,

$$(n)_b = (1110), \quad (4.21)$$

a conversão binário-código de Gray (4.7) conduz a:

$$(n)_g = (1001). \quad (4.22)$$

Substituindo esta informação na igualdade (4.20), obtém-se:

$$WAL(n, t) = (-1)^{\{2^4 \times 0.2\} + \{2^1 \times 0.2\}} = (-1)^{\{3.2\} + \{0.4\}} = (-1)^{3+0} = -1.$$

A definição de Chien é bastante importante, pois a sua utilização permite demonstrar, de uma forma bastante simples, algumas das mais importantes propriedades das funções de Walsh [35]. Todavia, a equação (4.20) não é de fácil utilização prática. Para efectuar a geração directa das funções de Walsh, Chien propôs uma fórmula alternativa, segundo a qual se tem que:

$$WAL(n, t) = (-1)^{\{2^k t\} + \{2^j t\} + \{2^i t\} + \{2^h t\} + \dots}. \quad (4.23)$$

O valor dos parâmetros k, j, i, h, \dots , é obtido aplicando o procedimento descrito em seguida.

1. o valor de k é determinado de forma a que:

$$2^k > n > 2^{k-1},$$

2. o valor de j é calculado de modo a que:

$$2^k - 2^{j-1} > n \geq 2^k - 2^j ,$$

3. o valor de i é determinado de forma a verificar-se:

$$2^k - 2^j + 2^i \geq n > 2^k - 2^j + 2^{i-1} ,$$

4. o valor de h é calculado de modo a que:

$$2^k - 2^j + 2^i - 2^{h-1} > n \geq 2^k - 2^j + 2^i - 2^h ,$$

e assim sucessivamente. Se n é par, o processo termina quando o somatório alternado converge para n . Quando n é ímpar, é necessário que o somatório convirja para $n+1$.

4.4.3 Produtos de funções de Rademacher

As funções de Walsh podem ser geradas executando produtos sucessivos de funções de Rademacher [144, 114, 51]. Estas funções formam uma série incompleta de funções ortogonais que apresentam simetria ímpar em relação ao ponto médio do seu intervalo de definição. Tal como para as funções de Walsh, são necessários dois parâmetros para a sua completa definição. A função de ordem n apresenta, no intervalo $0 \leq t \leq 1$, 2^n ondas quadradas com apenas duas amplitudes possíveis, $+1$ e -1 .

As funções de Rademacher [171], aqui denotadas por $RAD(n, t)$, podem ser obtidas a partir das funções trigonométricas SENO através da expressão:

$$RAD(n, t) = \text{sign}(\sin(2^n \pi t)) . \quad (4.24)$$

Embora formando uma série incompleta de funções anti-simétricas é possível, a partir das funções de Rademacher, construir séries completas. Em particular, um conjunto completo de funções de Walsh com ordenação diádica pode ser obtido através do produto sucessivo de funções de Rademacher, seleccionadas da forma que de seguida se indica. Considere-se de novo a representação binária de n (4.3). A função $PAL(n, t)$ pode ser gerada a partir de :

$$PAL(n, t) = \prod_{i=0}^k b_i RAD(i+1, t) . \quad (4.25)$$

Por exemplo, tendo em atenção (4.21) pode escrever-se:

$$PAL(14, t) = RAD(4, t) RAD(3, t) RAD(2, t) .$$

Tal como Lackey e Meltzer [114] e mais tarde Davies [51] verificaram, as funções de Rademacher podem também ser utilizadas na geração de uma série completa de

funções de Walsh ordenadas sequencialmente. Considerando a conversão binário-código de Gray (4.7) aplicada sobre o inteiro n , a função $WAL(n, t)$ pode ser obtida através do produto:

$$WAL(n, t) = \prod_{i=0}^k g_i RAD(i+1, t) . \quad (4.26)$$

Como exemplo, verifique-se que tendo em atenção a igualdade (4.22) se pode escrever:

$$WAL(14, t) = RAD(4, t) RAD(1, t) .$$

Pode demonstrar-se [35] que as definições (4.26) e (4.20) são perfeitamente equivalentes. Para tal, basta ter em atenção que o factor $(-1)^i$, com $i = \{2^j t\}$, não é mais do que a representação explícita da função de Rademacher de ordem j .

4.4.4 Montagem da matriz de Hadamard

A geração de uma série de funções de Walsh com ordenação natural pode ser efectuada a partir da montagem da matriz de Hadamard [18, 97], uma matriz quadrada, na qual cada coeficiente pode tomar apenas dois valores possíveis, +1 e -1, e onde as linhas (colunas) são ortogonais entre si. Tendo-se uma matriz de Hadamard, é sempre possível permutar linhas (colunas) e/ou trocar o sinal de todos os elementos pertencentes a uma dada linha (coluna) sem alterar a ortogonalidade existente. Estas características tornam possível a obtenção de uma matriz simétrica, conhecida normalmente por *Forma Normal da Matriz de Hadamard*, na qual a primeira linha e a primeira coluna contêm apenas coeficientes positivos (valendo +1). A matriz de Hadamard de ordem mais baixa é a de ordem 2, para a qual se tem:

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} . \quad (4.27)$$

Matrizes de ordem superior podem ser obtidas através de,

$$\mathbf{H}_N = \mathbf{H}_{N/2} \otimes \mathbf{H}_2 , \quad (4.28)$$

onde \otimes denota o produto de Kronecker e N é uma potência de 2. O produto de Kronecker implica a substituição da cada elemento positivo (negativo) da matriz $\mathbf{H}_{N/2}$ por \mathbf{H}_2 ($-\mathbf{H}_2$). A matriz de quarta ordem é obtida através do produto:

$$\mathbf{H}_4 = \mathbf{H}_2 \otimes \mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} .$$

Considere-se o intervalo de definição das funções de Walsh dividido em N sub-intervalos de igual dimensão. A linha n da matriz de Hadamard \mathbf{H}_N estabelece qual o valor que a função $HAD(n, t)$ toma em cada um daqueles subintervalos. Como

exemplo, verifica-se que as oito primeiras funções de Walsh com ordenação natural podem ser obtidas através da montagem da matriz de Hadamard com $N = 8$.

$$\mathbf{H}_8 = \mathbf{H}_4 \otimes \mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} HAD(0, t) \\ HAD(1, t) \\ HAD(2, t) \\ HAD(3, t) \\ HAD(4, t) \\ HAD(5, t) \\ HAD(6, t) \\ HAD(7, t) \end{bmatrix} .$$

4.5 Propriedades das funções de Walsh

São enunciadas de seguida algumas das mais significativas propriedades das funções de Walsh. Apresentam-se no apêndice B as respectivas demonstrações, realizadas com base nos trabalhos de Paley [144] e Chien [35].

P1 As funções de Walsh, quando definidas no intervalo $0 \leq t \leq 1$, são ortonormais entre si:

$$\int_0^1 WAL(i, t) WAL(j, t) dt = \delta_{ij} . \quad (4.29)$$

Quando as funções de Walsh são definidas sobre o intervalo $[0, T]$, com $T \neq 1$, perde-se a condição de normalidade. Tem-se, nesse caso:

$$\int_0^T WAL(i, t) WAL(j, t) dt = T \delta_{ij} . \quad (4.30)$$

Não é contudo difícil obter, a partir das funções de Walsh, um outro conjunto de funções que sejam ortonormais naquele intervalo de definição. Podem estas ser obtidas através da igualdade:

$$WAL^*(m, t) = \frac{1}{\sqrt{T}} WAL(m, t) . \quad (4.31)$$

P2 As funções de Walsh formam um sistema completo de funções.

Diz-se que uma série de funções $S_n(x)$, com $a \leq x \leq b$, é completa quando não existe nenhuma função $f(x) \in L^2$ que verifique a igualdade

$$\int_a^b f(x) S_n(x) dx = 0 ,$$

para todos os valores de n . Por outras palavras, pode dizer-se que uma série de funções ortogonais é considerada completa se não existir nenhuma função (cujo

quadrado seja integrável no intervalo de definição) que seja ortogonal a todas as funções da série, a menos que o integral do quadrado da própria função seja igual a zero.

P3 O conjunto das funções de Walsh com grau inferior a 2^p , com $p = 0, 1, 2, \dots$, define, quando associado à operação multiplicação, um grupo abeliano.

Resulta da propriedade **P3** que o produto de duas funções de Walsh de grau inferior a 2^p permite obter uma outra função de Walsh ainda com grau inferior a 2^p . Nomeadamente, verifica-se que

$$WAL(i, t) \times WAL(j, t) = WAL(l, t) , \quad (4.32)$$

com:

$$(l)_b = (i)_b \oplus (j)_b . \quad (4.33)$$

P4 Considerando N (com $N = 2^p$) pontos igualmente espaçados no intervalo de definição das funções de Walsh, verifica-se que $HAD(n, m) = HAD(m, n)$, com $m, n = 0, 1, \dots, N - 1$.

Esta propriedade resulta directamente da simetria da matriz de Hadamard utilizada na geração das funções de Walsh com ordenação natural. Tem uma importante consequência prática, pois permite estabelecer que as transformadas discretas de Walsh e as suas inversas são realizadas utilizando o mesmo operador matemático.

4.6 Transformadas de Walsh

Seja $f(t)$ uma função contínua definida no intervalo $0 \leq x \leq T$. Tendo em conta as propriedades **P2** e **P3**, Walsh [192] demonstra que esta função pode ser expressa como combinação linear de um conjunto completo de funções digitais [192, 18, 201]:

$$f'(t) = \sum_{n=0}^{N-1} X_n WAL(n, t) , \quad (4.34)$$

onde os coeficientes X_n são obtidos a partir de:

$$X_n = \frac{1}{T} \int_0^T f(t) WAL(n, t) dt . \quad (4.35)$$

Tendo em conta as igualdades (4.1) e (4.2), a expansão de $f(t)$ em série de Walsh pode assumir a forma:

$$f'(t) = a_0 WAL(0, t) + \sum_{i=1}^{N/2-1} \sum_{j=1}^{N/2} (a_i CAL(i, t) + b_j SAL(j, t)) , \quad (4.36)$$

$$a_i = \frac{1}{T} \int_0^T f(t) CAL(i, t) dt , \quad (4.37)$$

$$b_j = \frac{1}{T} \int_0^T f(t) \text{SAL}(j, t) dt . \quad (4.38)$$

As expressões (4.36-4.38) permitem salientar a semelhança de processos existente entre as expansões em séries de Walsh e as expansões em séries de Fourier. Os somatórios (4.34) e (4.36) convergem para a função $f(t)$ à medida que se aumenta o número de termos considerado na série utilizada no desenvolvimento [192, 201].

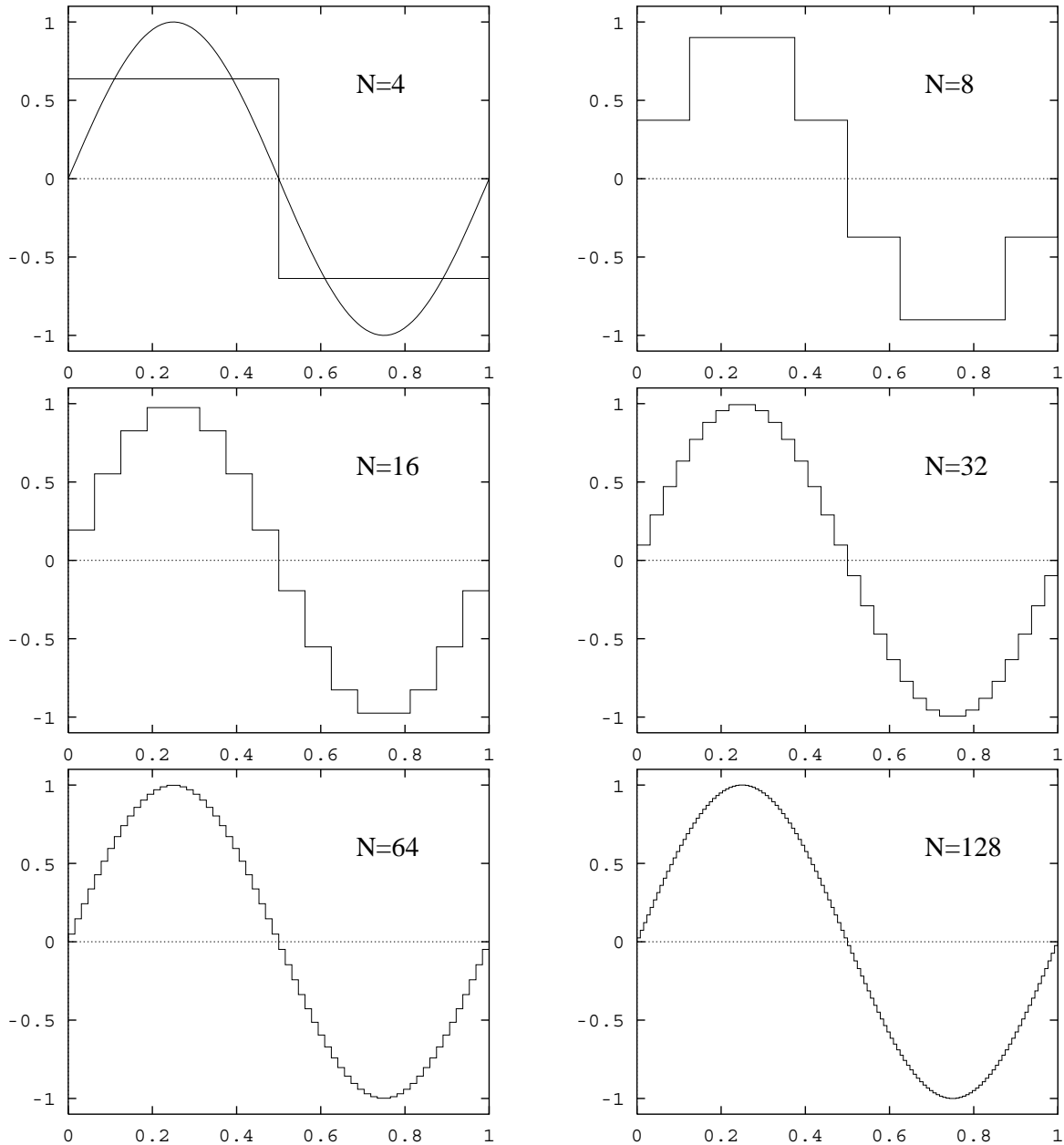


Figura 4.7: *Expansão de uma função trigonométrica em série de Walsh.*

Ilustra-se, na figura 4.7, a expansão em série de Walsh da função $f(t) = \sin(2\pi t)$, definida no intervalo $0 \leq t \leq 1$. Verifica-se, tal como seria de esperar, que é necessário considerar um número de termos elevado para que a aproximação obtida

tenha algum significado. No entanto, importa salientar que embora o número de coeficientes X_n a determinar seja elevado, o seu cálculo é extremamente simples e rápido, pois as multiplicações envolvidas no cálculo da função integranda em (4.35) podem ser reduzidas a uma sequência apropriada de trocas de sinal. Na tabela 4.4 apresenta-se o tempo de CPU gasto na determinação dos N coeficientes do desenvolvimento em série de Walsh da função trigonométrica acima referida. Tal como para todos os testes numéricos apresentados ao longo deste texto, os cálculos foram efectuados numa estação de trabalho RISC 6000/550 da IBM.

N	4	8	16	32	64	128
T	0.3	0.7	2.6	10.5	41.9	166.0

Tabela 4.4: *Tempos de CPU (em milésimos de segundo) gastos no cálculo dos N coeficientes da expansão.*

A análise da tabela 4.4 permite confirmar a rapidez com que os coeficientes X_n podem ser obtidos. Este é um dos aspectos que torna tão atractivo o uso de séries digitais de Walsh no campo do processamento de sinais.

Quando se consideram $N = 2^p$ termos na expansão em série, com p inteiro, a função de aproximação definida em (4.34) ou (4.36) é constante em cada um dos N troços definidos por,

$$(\alpha - 1) \frac{t}{N} \leq i_n^\alpha \leq \alpha \frac{t}{N}, \text{ com } \alpha = 1, 2, \dots, N.$$

Verifica-se que o valor constante que $f'(t)$ toma em cada um daqueles sub-intervalos é igual ao valor médio da função $f(t)$ naqueles mesmos intervalos.

O cálculo dos coeficientes da expansão em série, tal como é definido em (4.35), pressupõe o conhecimento da expressão analítica da função $f(t)$. Mesmo quando esta é conhecida (o que em muitos casos não acontece), torna-se computacionalmente mais vantajoso trabalhar apenas com os valores que a função em causa toma em N pontos igualmente espaçados. Para que haja correspondência entre as propriedades dos sistemas contínuos e discretos, é conveniente considerar $N = 2^p$, com p inteiro. Nestas condições, e utilizando a regra dos trapézios, a integração presente em (4.35) pode ser transformada num somatório. Obtém-se desta forma a chamada *Transformada Discreta de Walsh* (DWT), que pode ser apresentada na forma,

$$x_i = \sum_{n=0}^{N-1} X_n \text{WAL}(n, i) \text{ com } i = 0, 1, \dots, N - 1, \quad (4.39)$$

e com os coeficientes da expansão dados por:

$$X_n = \frac{1}{N} \sum_{i=0}^{N-1} x_i \text{WAL}(n, i) \text{ com } n = 0, 1, \dots, N - 1. \quad (4.40)$$

A transformada (4.40) pode escrever-se matricialmente na forma,

$$\mathbf{X}_N = \frac{1}{N} \mathbf{W}_N \mathbf{x}_i, \quad (4.41)$$

onde \mathbf{x}_i (\mathbf{X}_N) representa um vector onde se agrupam os valores que a função toma nos N pontos igualmente espaçados (os valores dos coeficientes da expansão, X_n) e \mathbf{W}_N uma matriz onde se armazenam os valores que as funções de Walsh tomam em cada um dos N sub-intervalos considerados. Como o produto matricial presente em (4.41) envolve apenas multiplicações por $+1$ ou -1 , pode concluir-se que a transformada discreta de Walsh implica a realização de $N(N-1)$ adições e subtracções. Quando o valor de N é elevado, o volume de operações a realizar tende a tornar o cálculo pesado.

Para contornar este problema, Good [90] desenvolveu um método de factorização de matrizes que deu origem a um algoritmo comumente designado por FWT, ou seja, *Fast Walsh Transform*. A técnica de Good permite factorizar matrizes que tenham sido obtidas a partir de produtos de Kronecker. Para se aplicar directamente o algoritmo proposto por Good, convém então escrever a transformada (4.41) na forma,

$$\mathbf{X}_N = \frac{1}{N} \mathbf{H}_N \mathbf{x}_i, \quad (4.42)$$

onde \mathbf{H}_N denota a forma normal da matriz de Hadamard de ordem N . Convém salientar que agora os coeficientes da expansão armazenados no vector \mathbf{X}_N se encontram dispostos segundo a ordenação natural. Tendo em conta a simetria da matriz de Hadamard, a transformada inversa pode ser escrita matricialmente na forma:

$$\mathbf{x}_i = \mathbf{H}_N \mathbf{X}_N. \quad (4.43)$$

Aplicando a técnica de Good [29, 90], a matriz \mathbf{H}_N é factorizada em p factores de ordem N , onde N denota a ordem da matriz original e p representa o número de matrizes envolvidas no produto de Kronecker. Resultando a matriz original do produto de p matrizes iguais, os p factores são idênticos. Como as matrizes resultantes da factorização das matrizes de Hadamard têm bastantes elementos nulos, o número de operações necessárias para efectuar a transformada baixa para $N \log_2 N$ [29].

Como exemplo da factorização de Good, refira-se que a matriz de Hadamard de quarta ordem pode ser expressa na forma:

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}^2. \quad (4.44)$$

Seja $f(t)$ uma função contínua definida no intervalo $0 \leq t \leq T$. Considere-se que no vector \mathbf{v} se encontram armazenados os valores que aquela função toma em $N = 2^p$ pontos equidistantes. O algoritmo FWT baseado na factorização de Good pode ser

escrito do seguinte modo [29, 18]:

$$\begin{aligned}
 & y_0(n) = v(n) \text{ com } n = 1, 2, \dots, N \\
 & \text{Variar } i \text{ desde } 1 \text{ até } p \\
 & \quad y_i(n) = y_{i-1}(2n-1) + y_{i-1}(2n) \text{ se } 0 < n \leq N/2 \\
 & \quad y_i(n) = y_{i-1}(2n-1) - y_{i-1}(2n-N) \text{ se } N/2 < n \leq N \\
 & \text{Fim do ciclo em } i \\
 & N \mathbf{X}_n = \mathbf{y}_p .
 \end{aligned} \tag{4.45}$$

Como foi salientado anteriormente, a aplicação deste algoritmo conduz à obtenção de resultados com ordenação natural. A sua implementação requer a utilização de memória adicional para armazenar resultados intermédios. Embora não sejam aqui apresentados, existem outros algoritmos capazes de efectuar a transformada FWT de uma forma muito eficaz [18]. A aplicação destes algoritmos permite obter os resultados com diferentes tipos de ordenação. Alguns deles apresentam a vantagem de efectuarem os cálculos *in place*, não sendo necessário reservar espaço em memória para armazenar resultados intermédios, uma vez que os valores obtidos em cada etapa de cálculo podem ser guardados nas posições ocupadas previamente pelos valores iniciais [18]. Dentro deste tipo de algoritmos destacam-se os de Shanks [179], Manz [127] e Larsen [117].

Na tabela 4.5 apresentam-se os tempos de CPU gastos na determinação dos coeficientes da expansão em série de Walsh ilustrada na figura 4.7, quando se utiliza o algoritmo de FWT apresentado em (4.45). É notória a redução de tempo registada em relação aos valores apresentados na tabela 4.4.

N	4	8	16	32	64	128
T	0.01	0.02	0.03	0.08	0.14	0.28

Tabela 4.5: *Tempos de CPU (em milésimos de segundo) gastos no cálculo dos N coeficientes da expansão em série utilizando o algoritmo FWT.*

A generalização do conceito de transformada de Walsh a problemas bidimensionais é imediata. Pode então escrever-se:

$$X_{m,n} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{i,j} \text{WAL}(m,i) \text{WAL}(n,j) \text{ com } m, n = 0, 1, \dots, N-1 . \tag{4.46}$$

A matriz $\mathbf{X}_{m,n}$ ($\mathbf{x}_{i,j}$) armazena os coeficientes da expansão em série (os valores que a função a aproximar toma em cada um dos $N \times N$ pontos em que se discretiza o domínio). A transformada inversa pode ser escrita na forma:

$$x_{i,j} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} X_{m,n} \text{WAL}(m,i) \text{WAL}(n,j) \text{ com } i, j = 0, 1, \dots, N-1 . \tag{4.47}$$

O cálculo definido em (4.46) pode ser realizado em duas etapas. Em primeiro lugar, efectua-se a transformada referente à variável i . Obtém-se então:

$$X_{i,n} = \frac{1}{N} \sum_{j=0}^{N-1} x_{i,j} WAL(n,j) \text{ com } i, n = 0, 1, \dots, N-1. \quad (4.48)$$

Esta etapa equivale a aplicar um transformada de Walsh unidimensional a cada uma das linhas da matriz $\mathbf{x}_{i,j}$. O segundo passo consiste em calcular,

$$X_{m,n} = \frac{1}{N} \sum_{i=0}^{N-1} X_{i,n} WAL(m,i) \text{ com } m, n = 0, 1, \dots, N-1, \quad (4.49)$$

o que equivale a realizar de novo transformadas de Walsh unidimensionais aplicadas agora a cada uma das colunas da matriz $\mathbf{X}_{i,n}$. Consegue-se desta forma executar a transformada bidimensional à custa da aplicação sucessiva de transformadas unidimensionais.

Para maximizar a eficiência do processo de cálculo das transformadas bidimensionais, torna-se imprescindível a aplicação de algoritmos FWT na realização das transformadas unidimensionais descritas em (4.48) e (4.49). É então possível efectuar o cálculo com $2N^2 \log_2 N$ adições e subtracções em vez das $2N^3$ operações envolvidas na aplicação directa da definição (4.46).

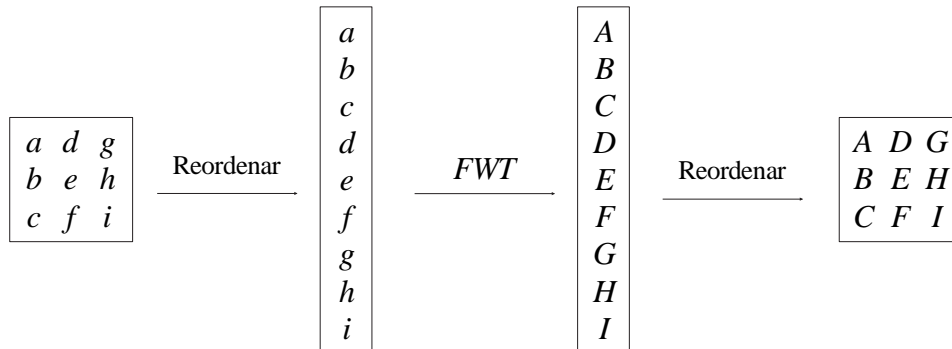


Figura 4.8: *Aplicação da transformada de Walsh bidimensional.*

Beauchamp [18] apresenta um processo alternativo para a execução das transformadas de Walsh bidimensionais. A aplicação deste método, representada esquematicamente na figura 4.8, consiste na realização de três operações. Primeiro, os coeficientes da matriz $\mathbf{x}_{i,j}$ são escritos num vector de dimensão N^2 , havendo o cuidado de arrumar as colunas sequencialmente. Depois, há que efectuar uma transformada unidimensional sobre este vector através da utilização de um algoritmo FWT. Finalmente, reordena-se o vector resultante da aplicação desta segunda operação, de modo a recuperar-se a matriz dos coeficientes da expansão em série.

