

**KONINKLIJK NEDERLANDS
METEOROLOGISCH INSTITUUT**

VERSLAGEN

V - 364

B. Heijna

Walsh-funkties en enkele
van hun toepassingen.

De Bilt 1980

Publikatienummer: K.N.M.I. V-364(MBW)

Koninklijk Nederlands Meteorologisch Instituut,
Machinale Bewerking van Waarnemingen,
Postbus 201,
3730 AE De Bilt,
Nederland.

U.D.C.: 518.5

I N H O U D

1. Inleiding
2. Enkele opmerkingen over digitaliseren
3. Walsh funkties, keuze van definitie en notatie
4. Afbeelding op vectorruimte
5. Fast Walsh Fourier Transform
6. Integreren en differentieren
7. Toepassing op differentieel vergelijkingen
8. Toepassing op de variatierekening
9. Sequentietransformatie en filteren
10. Fase verschuiving en balfunkties.

1. Inleiding

Walsh funkties zijn funkties die alleen de waarden +1 of -1 aannemen, terwijl de overgang van de ene waarden naar de andere alleen plaatsvindt als het argument van de funktie gelijk is aan $K/2^m$, waarin K en m integers zijn.

Walsh funkties vertonen dus een sterk lineair karakter en sluiten dus goed aan bij de digitale technieken, zoals deze onder meer in computers en datatransmissie gebruikt worden.

Helaas zijn de definities, die men bij verschillende auteurs aantreft niet alle aan elkaar gelijk en bovendien soms zeer ingewikkeld. In paragraaf 3 is daarom een definitie ontwikkeld die de voordelen van de verschillende definities in zich verenigt en redelijk eenvoudig is.

Om Walsh funkties op analoge grootheden te kunnen toepassen zal men eerst moeten digitaliseren. In paragraaf 1 worden daarom enkele opmerkingen over digitaliseren gemaakt.

Voor het verdere betoog blijkt het voordelen te hebben om gebruik te kunnen maken van de theorie der abstracte funktieruimten. Daarom wordt in paragraaf 4 een afbeelding van de Walsh-funkties op een vectorruimte (Unitair of Hilbert) gegeven.

In de volgende paragrafen worden toepassingen van het voorgaande gegeven.

2. Enkele opmerkingen over digitaliseren

Vele grootheden uit de praktijk (b.v. luchtdruk, temperatuur etc.) vertonen een analoog karakter, d.w.z. ze kunnen in principe iedere waarde aannemen, maar we kennen deze waarde niet tengevolge van waarnemingsfouten e.d.

Digitale grootheden kunnen dat slechts waarden aannemen uit een eindige verzameling, maar dat doen ze d.e.t. ook "exact": Als we b.v. de bitconfiguratie van een digitale grootheid kennen dan korrepondeert hiermee precies een bepaalde waarde.

Wanneer men over een rekenmachine spreekt dan bedoelt men daarmee als regel een digitale. Weliswaar bestaan er ook analoge rekenmachines, maar de huidige techniek voor het herwaardigen van zgn. geïntegreerde circuits heeft zo'n grote vlucht genomen dat digitaal verwerken in de meeste gevallen de voorkeur verdient uit hoofde van efficiëntie, snelheid e.d.

Wil men de digitale verwerkingsmethoden ook op analoge grootheden toepassen, dan zullen deze eerst in digitale vorm overgezet moeten worden. Hiervoor bestaan zgn. analoog-digitaal omzetters, die de omzetting automatisch verrichten. Lang voor de komst van deze apparatuur echter werd analoog-digitaal omzetting in het kort: digitaliseren, al toegepast. Als de waarnemer de luchttemperatuur op gehele graden afrondt, om deze mee te geven in het synoptische weerbericht, dan verricht hij in feite een analoog-digitaal omzetting. Het uittrekken van registreerstroken is een vorm van digitalisering, waaraan niet iedereen even prettige herinneringen zal bewaren.

Er blijft nog te beantwoorden de vraag met hoeveel decimalen c.q. bits men een te digitaliseren grootheid wil weergeven. Voor afzonderlijke waarnemingen is de gewenste (of bereikbare) nauwkeurigheid een richtlijn. Meer bits dan waarmee men deze nauwkeurigheid kan weergeven leveren geen enkele nuttige informatie. Soms kan men echter met minder volstaan, b.v. een automobilist die wil weten of hij overdag zijn koplampen moet ontsteken, hoeft daarvoor alleen maar te weten: zicht <200 m of niet. Voor hem geeft 1 bit dus voldoende informatie over het zicht. Op een vliegveld det zal men alle significante bits van het zicht willen weten.

Iets ingewikkelder is het wanneer een grootheid continue geregistreerd wordt. Men heeft dan een functie $x(t)$, waarin t als regel de tijd zal zijn, en waarin nu x en t beide gedigitaliseerd moeten worden. Voor x kan weer de gewenste nauwkeurigheid als maatstaaf dienen, voor t geldt het zogenaamde Nyquist criterium. (8)

Als men bemonstert met een tijdstap ΔT en het te bemonsteren signaal bevat geen hogere frequenties dan $f_u = \frac{1}{2 \Delta T}$, dan kan men het oorspronkelijke signaal weer uit de bemonsterde waarden terugwinnen.

Bevat het signaal wel hogere frequenties, dan kan men twee dingen doen: Of de tijdstap verkleinen, (als de hogere frquenties belangrijk zijn), of de hogere frequenties eerst uit filteren, wanneer ze niet belangrijk zijn.

3. Walshfuncties, keuze van definitie en notatie

3.1. Overzicht

In 1923 introduceerde J.L. Walsh [14] een stelsel van orthogonale functies, die in vele opzichten overeenkomst¹⁾ vertoonden met andere orthogonale functies, zoals sinus/cos, Legendre, Storm-Leouville etc., maar in één opzicht daarvan duidelijk verschillen: ze waren niet continue. De interesse hiervoor lag voornamelijk in het zuiver wiskundige vlak. Pas veel later, nadat de digitale rekenapparatuur algemeen ingang gevonden had, werden de Walshfuncties in de praktijk toegepast.

Helaas gebruikten latere auteurs niet allemaal dezelfde definitie van Walshfuncties (vgl. paragraaf 3.3 en 3.4.), een der doelstellingen van dit rapport is daarin eenheid te brengen (zie paragraaf 3.5).

3.2. De definitie van Walsh

Walsh (14) gaf een recursieve formule voor een aftelbaar oneindige verzameling van functies $\phi_n^{(k)}(x)$ die er a.v. uitzag:

$$\phi_0(x) \equiv 1, \quad 0 \leq x \leq 1, \quad \phi_1(x) \equiv \begin{cases} 1, & 0 \leq x < \frac{1}{2}, \\ -1, & \frac{1}{2} < x \leq 1, \end{cases}$$

$$\phi_2^{(1)}(x) \equiv \begin{cases} 1, & 0 \leq x < \frac{1}{4}, \quad \frac{3}{4} < x \leq 1, \\ -1, & \frac{1}{4} < x < \frac{3}{4}, \end{cases}$$

$$\phi_2^{(2)}(x) \equiv \begin{cases} 1, & 0 \leq x < \frac{1}{4}, \quad \frac{1}{2} < x < \frac{3}{4}, \\ -1, & \frac{1}{4} < x < \frac{1}{2}, \quad \frac{3}{4} < x \leq 1, \end{cases}$$

- 1) Zoals: de n functie heeft n nulpunten (beter: tekenwisselingen) in het beschouwde interval, elke functie is even resp. oneven t.a.v. het midden van het interval. Geen functie is $= 0$ over een subinterval van het oorspronkelijk interval, de gehele rij is uniform begrensd, etc.

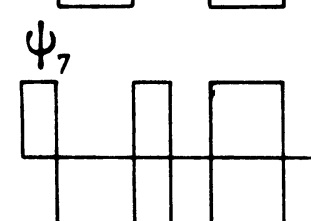
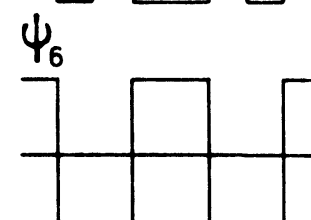
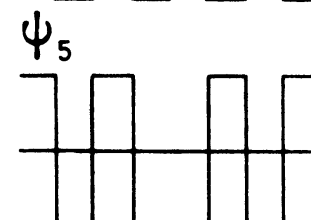
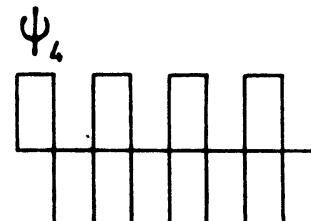
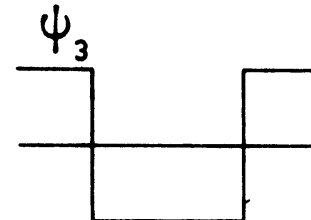
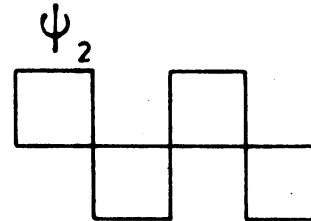
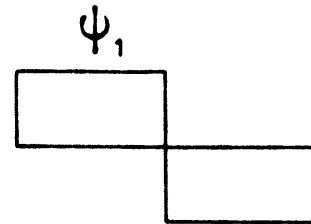
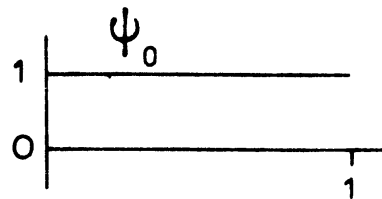
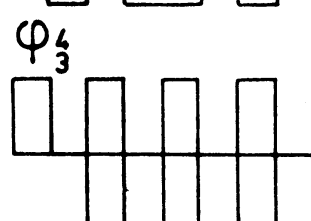
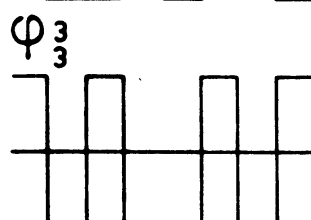
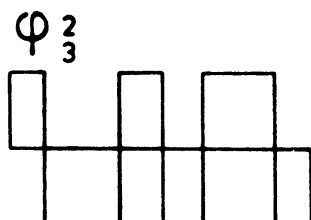
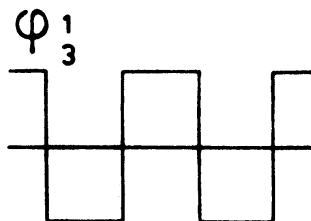
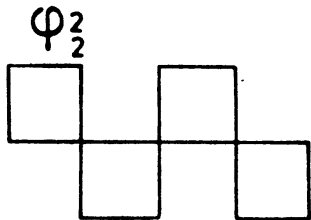
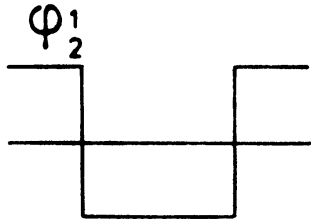
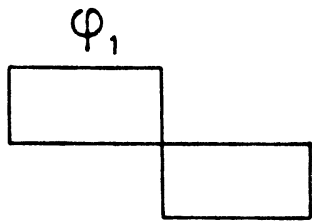
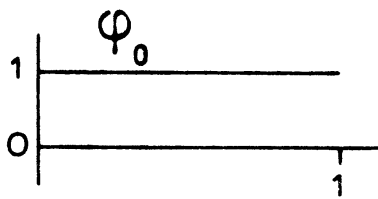


Fig. 1 De Funkties

φ_n^k van Walsh.

Fig. 2 De Funkties

ψ_n van Chen.

$$\phi_{n+1}^{(2k-1)}(x) \equiv \begin{cases} \phi_n^{(k)}(2x), & 0 \leq x < \frac{1}{2}, \\ (-1)^{k+1} \phi_n^{(k)}(2x-1), & \frac{1}{2} < x \leq 1, \end{cases} \quad (1)$$

$$\phi_{n+1}^{(2k)}(x) \equiv \begin{cases} \phi_n^{(k)}(2x), & 0 \leq x < \frac{1}{2}, \\ (-1)^k \phi_n^{(k)}(2x-1), & \frac{1}{2} < x \leq 1, \end{cases}$$

$$k = 1, 2, 3, \dots, 2^{n-1}, \quad n = 1, 2, 3, \dots, \infty$$

We kunnen de funkties $\phi_n^{(k)}$ indelen in groepen, elke groep bevat de funkties met dezelfde index n . Per groep zijn er dan 2^{n-1} verschillende funkties. Heeft men voor een bepaalde waarde van n alle funkties uit de groep bepaald, dan kan men hieruit de funkties voor de volgende groep (met index $n+1$) bepalen. De formules (1) zijn tamelijk ingewikkeld. Om een indruk te geven hoe de funkties er uitzien zijn de eerste 8 getekend in figuur 1.

In de discontinuïteitspunten neemt Walsh $\phi_n^{(k)} = 0$, omdat hij zich bezighield met puntsgewijze convergentie van een reeks $\sum C_{n,k} \phi_n^{(k)}(x)$. Tegenwoordig hecht men meer belang aan convergentie in norm (zie paragraaf 4) en dan geeft men de functie in de discontinuïteitspunten de waarde ± 1 (zie Harmuth). Zodat de funkties alleen de waarden $+1$ of -1 aannemen. We vervangen daarvoor de voorwaarde $\frac{1}{2} < x \leq 1$ in (1) door $\frac{1}{2} \leq x < 1$.

Hiermee is het mogelijk om de oplossing, die Walsh geeft voor het recursieve stelsel (1) te vereenvoudigen, omdat nu geen bijzondere maatregelen voor de sprongpunten genomen behoeven te worden. Daartoe schrijven we

$$x = \frac{b_1}{2} + \frac{b_2}{2^2} + \frac{b_3}{2^3} + \dots, \quad b_i = 0 \text{ of } 1 \quad (2)$$

dus $0 \leq b_1 \leq b_2 \leq b_3 \dots$ is de lineaire voorstelling van x , welke eindig of oneindig veel bits kan bevatten. Dan krijgen we voor de eerste 16 ϕ funkties de volgende uitkomst:

$$\begin{array}{ll}
 \phi_0 & = 1 \\
 \phi_2^{(1)} & = (-1)^{b_1+b_2} \\
 \phi_3^{(1)} & = (-1)^{b_2+b_3} \\
 \phi_3^{(3)} & = (-1)^{b_1+b_3} \\
 \phi_4^{(1)} & = (-1)^{b_3+b_4} \\
 \phi_4^{(3)} & = (-1)^{b_1+b_2+b_3+b_4} \\
 \phi_4^{(5)} & = (-1)^{b_2+b_4} \\
 \phi_4^{(7)} & = (-1)^{b_1+b_4} \\
 \phi_1 & = (-1)^{b_1} \\
 \phi_2^{(2)} & = (-1)^{b_2} \\
 \phi_3^{(2)} & = (-1)^{b_1+b_2+b_3} \\
 \phi_3^{(4)} & = (-1)^{b_3} \\
 \phi_4^{(2)} & = (-1)^{b_1+b_3+b_4} \\
 \phi_4^{(4)} & = (-1)^{b_2+b_3+b_4} \\
 \phi_4^{(6)} & = (-1)^{b_1+b_2+b_4} \\
 \phi_4^{(8)} & = (-1)^{b_4}
 \end{array} \tag{3a}$$

We kunnen de notatie met 2 indices vervangen door een notatie met een enkele index volgens

$$\phi_n^{(k)}(x) = \phi_{2 \uparrow (n-1) + k-1}(x) \tag{3b}$$

en kunnen daarmee de functies ϕ rangschikken in een rij $\phi_0 \phi_1 \phi_2 \phi_3 \phi_4 \dots$. Deze rij vormt een onthogonale basis in de Hilbertruimte van kwadratisch integreerbare functies (zie paragraaf 4).

Als algemene oplossing voor (3a) geeft Walsh

$$\phi_n^{(1)} = (-1)^{b_{n-1}+b_n} \tag{3c}$$

$$\phi_n^{(k)} = \phi_{k-1} \phi_n^{(1)}$$

Ook dit is nog een recursieve formule, we zullen in paragraaf 3.5 aantonen dat hieruit een niet recursieve definitie afgeleid kan worden.

3.3. Definitie van Chen en medewerkers

Chen en medewerkers (4) en (5) geven een definitie, die gebruik maakt van Rademacher functies. Deze definitie is een stuk eenvoudiger dan die van Walsh (zie paragraaf 3.2.), maar geeft resultaten die afwijken van die van Walsh.

Rademacher-funkties zijn blok-funkties:

$$r_n(t) = \begin{cases} +1 & \text{voor } 2^j \leq t < 2^{j+1} \\ -1 & \text{voor } 2^{j-1} \leq t < 2^j \end{cases} \quad j \text{ integer,} \quad (4)$$

Hiermee definieert Chen een rij funkties $\psi_n(t)$ a.v.

$$\Psi_0(t) = r_0(t)$$

$$\Psi_1(t) = r_1(t)$$

$$\Psi_2(t) = \{r_2(t)\}^1 \{r_1(t)\}^0$$

$$\Psi_3(t) = \{r_2(t)\}^1 \{r_1(t)\}^1$$

en algemeen

$$\Psi_n(t) = \prod_{k=0}^m \{r_k(t)\}^{b_k} \quad (5)$$

waarin $b_0 b_1 b_2$ de bits zijn van de binaire notatie van n , dus

$$n = b_m 2^m + b_{m-1} 2^{m-1} + \dots + b_1 2 + b_0$$

en m het grootste gehele getal zodanig dat

$$2^m \leq n$$

De funkties Ψ zijn dus het product van een (eindig) aantal Rademacher-funkties nl. van die waarbij de bits van n gelijk aan 1 zijn.

De funkties $\Psi_0(t)$ t/m $\Psi_7(t)$ zijn weergegeven in figuur 2.

Bij nadere inspectie blijkt dat de funkties $\Psi_2(t)$ als groep wel dezelfde zijn als $\phi_n(t)$, maar de volgorde is anders,

$$\Psi_0(t) = \phi_0(t)$$

$$\Psi_1(t) = \phi_1(t)$$

$$\Psi_2(t) = \phi_3(t)$$

$$\Psi_3(t) = \phi_2(t)$$

$$\Psi_4(t) = \phi_7(t)$$

$$\Psi_5(t) = \phi_6(t)$$

$$\Psi_6(t) = \phi_4(t)$$

$$\Psi_7(t) = \phi_5(t)$$

etc.

(6)

Om verwarring te voorkomen zullen we de functies $\Psi_n(t)$ Chen functies noemen. In paragraaf 3.5. komen we hier nog op terug.

3.4. De definitie van Harmuth

Harmuth [7] voert de notatie $wal(j, t)$ in voor de Walsh functies en geeft een recursieve betrekking als definitie:

$$wal(2j+p, t) = (-1)^{\lfloor j/2 \rfloor + p} \{ wal(j, 2t + \frac{1}{2}) + (-1)^{j+p} wal(j, 2t - \frac{1}{2}) \} \quad (7)$$

voor $-\frac{1}{2} \leq t \leq \frac{1}{2}$; $p = 0$ of 1 ; $j = 0, 1, 2, \dots$; $wal(0, t) = 1$

en $\lfloor j/2 \rfloor =$ entier $(j/2)$. Verder is $wal(j, t) = 0$ voor $t < -\frac{1}{2}$ en $t > \frac{1}{2}$.

Harmuth definieert de wal-functies dus op het vak $-\frac{1}{2} \leq t \leq \frac{1}{2}$, en toont aan dat wal-functies even resp. oneven functies zijn, wat uiteraard ook het geval is voor $\cos t$ en $\sin t$, terwijl bovendien het aantal tekenwisselingen van $wal(2j-p)$ gelijk is aan dat van $\cos(2j\pi t)$ en $\sin(2j\pi t)$ op het vak $(-\frac{1}{2}, \frac{1}{2})$. Dit is aanleiding voor hem om de wal-functies te splitsen in cal en sal-functies:

$$\begin{aligned} wal(2j, t) &= cal(j, t) \\ wal(2j-1, t) &= sal(j, t) \end{aligned} \quad (8)$$

waarmee in toepassingen bepaalde overeenkomsten worden verkregen tussen

$$\begin{aligned} wal(2j, t) &\text{ en } \exp(2\pi i j t); \quad i = \sqrt{-1} \\ cal(2j, t) &\text{ en } \cos(2\pi j t) \\ sal(2j, t) &\text{ en } \sin(2\pi j t). \end{aligned}$$

Deze overeenkomst was de reden voor de keuze van de letters c en s in (8), de letters "al" refereren aan Walsh.

In (7) is het eerste argument van $wal(j, t)$ steeds een integer. Pichler [1] breidt de definitie uit tot gevallen waarbij ook j een reëel getal is, aan deze uitbreiding hebben wij echter geen behoefte. Wij be-

perken ons dus tot $j = \text{integer}$, en kunnen dan j ook als index gebruiken:

$$\text{wal}(j, t) = \text{wal}_j(t) , j = 0, 1, 2,$$

en verkrijgen daarmee weer een rij

$$\text{wal}_0(t), \text{wal}_1(t), \text{wal}_2(t), \text{wal}_3(t)$$

Deze functies kunnen recursief uit (7) berekend worden, we vinden dan

$$\text{wal}_0(t) = \phi_0(t)$$

$$\text{wal}_1(t) = \text{sal}_1(t) = \phi_1(t)$$

$$\text{wal}_2(t) = \text{cal}_1(t) = \phi_2(t)$$

$$\text{wal}_3(t) = \text{sal}_2(t) = \phi_3(t)$$

$$\text{wal}_4(t) = \text{cal}_2(t) = \phi_4(t)$$

$$\text{wal}_5(t) = \text{sal}_3(t) = \phi_5(t)$$

$$\text{wal}_6(t) = \text{cal}_3(t) = \phi_6(t)$$

$$\text{wal}_7(t) = \text{sal}_4(t) = \phi_7(t)$$

Uit figuur 1 zien we, dat we de cal en sal-functies kunnen opvatten als zeer sterk gedigitaliseerde cos resp. sin functies.

In $\cos(2\pi ft)$ heeft f de naam frequentie gekregen, f is het halve aantal tekenwisselingen van $\cos(2\pi ft)$ in een vak ter lengte 1. Evenzo is j in (8) het halve aantal tekenwisselingen van $\text{cal}(j, t)$ c.q. $\text{sal}(j, t)$ in het vak $-\frac{1}{2} \leq t < \frac{1}{2}$. Dit was voor Harmuth reden om j de sequentie te noemen.

In een Fourier reeks

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos 2\pi ft + b_n \sin 2\pi ft)$$

$$\text{heet } P(f) = a_f^2 + b_f^2$$

het (frequentie) energie spectrum.

Dit is voor Harmuth aanleiding om uitgaande van de reeks

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \text{cal}_n(t) + b_n \text{sal}_n(t))$$

als sequentie energiespectrum te definiëren

$$\phi(n) = a_n^2 + b_n^2, n = 1, 2, \dots \quad (9)$$

waarin n de sequentie is.

(9) geeft de verdeling van de energie over de verschillende componenten cal_n en sal_n .

Vanwege de analogie tussen cal en cos functies mag men ook overeenkomst verwachten tussen het frequentiespectrum en het sequentiespectrum. Of men voor het ene of voor het andere kiest, zal per geval bekeken moeten worden. In paragraaf 5 zal aangetoond worden dat het berekenen van een sequentiespectrum in ieder geval minder rekenwerk kost dan het berekenen van een frequentiespectrum.

Harmuth (7) geeft ook een formule voor het produkt van twee Walsh functies:

$$wal(h, t) \times wal(k, t) = wal(h \diamond k, t) \quad (10)$$

Hierin is $h \diamond k$ de logische som van h en k, die men verkrijgt door h en k linair te schrijven en de bits van h en k op te tellen volgens onderstaand schema

h_i	k_i	$h_i \diamond k_i$
0	0	0
0	1	1
1	0	1
1	1	0

Hierin zijn h_i en k_i , $i = 0, 1, 2, \dots$ de i bit van h resp k.

Bijvoorbeeld

$$\begin{array}{r}
 5 = 0101 \text{ (linair)} \\
 12 = 1100 \\
 \hline
 5 \diamond 12 = 1001 = 9
 \end{array}$$

Het produkt van twee Walsh-functies is dus weer een Walsh-functie, de Walsh-functies vormen dus een groep met betrekking tot de vermenigvuldiging.

Het eenheidselement in deze groep is $wal(0, t)$, want

$$wal(k, t) \times w(0, t) = w(k, t) \text{ omdat}$$

$$k \circ 0 = k.$$

Het inverse-element van $wal(k, t)$ is $wal(k, t)$ zelf, omdat

$$wal(k, t) \times wal(k, t) = wal(0, t) = 1 \text{ wegens}$$

$$k \circ k = 0.$$

3.5. De aanbevolen definitie van Walsh-funkties

We hebben reeds gezien (paragraaf 3.3.) dat met de definitie van Chen [3] de volgorde van de Walsh-funkties verstoord wordt. Daardoor verliest het begrip sequentie (paragraaf 3.4.) zijn betekenis. Aangezien we hiervan in het vervolg nog gebruik willen maken, vervalt de definitie van Chen als kandidaat voor een mogelijke keuze.

De definities van Walsh [14] en Harmuth [7] (paragraaf 3.2. resp. 3.4.) berusten beide op een ingewikkelde, recursieve formule ((1) resp. (7)), waarmee moeilijk te werken valt. We zullen aantonen, dat uit (1) via (3c) een expliciete formule afgeleid kan worden, en dat de daarmee gevonden funkties overeenstemmen met (7), wanneer men het interval $(-\frac{1}{2}, \frac{1}{2})$ transformeert tot $(0, 1)$ volgens $wal(k, t) = wal(k, t-1)$ voor $-\frac{1}{2} \leq t < 0$.

Daarvoor hebben we nodig, de logische vermenigvuldiging, de "slang operator" en de funktie $par(n)$.

De logische vermenigvuldiging, evenals de reeds genoemde logische optelling, is in rekenmachine-kringen wel bekend en berust op het bit van bit vermenigvuldigen van twee getallen:

$$\text{als } h = \sum_{i=0}^n h_i \cdot 2^i \text{ en } k = \sum_{i=0}^n k_i \cdot 2^i, h_i, k_i = 0 \text{ of } 1$$

dan is $h \otimes k = \sum_{i=0}^n h_i k_i 2^i$

Bijvoorbeeld $21 = 10101$ (linair)

$$\begin{array}{r} 12 = 01100 \\ \hline \otimes \end{array}$$

$21 \otimes 12 = 00100 = 4$

De "slang operator", opereert op gehele getallen en op echte breuken.

Zij m geheel en $m_n m_{n-1} \dots m_1 m_0$ de bits van m , waarbij we $m_n = 0$ veronderstellen (anders voegen we links nog een bit 0 toe). Dan is

$$\tilde{m} = \sum_{i=0}^{n-1} (m_i \oplus m_{i+1}) 2^i, \tag{11a}$$

dus de bits van m zijn resp. $m_{n+1} \oplus m_n, m_n \oplus m_{n-1}, \dots, m_1 \oplus m_0$.

Heeft men eenmaal de lineaire notatie van m , dan is \tilde{m} gemakkelijk op te schrijven, b.v.

$m = 010110111$

$\tilde{m} = 11101100$

Voor een echte breuk x , dus $0 \leq x < 1$ definiëren we

$$\tilde{x} = \sum_{i=0}^{\infty} (b_i \oplus b_{i+1}) 2^i \tag{11b}$$

waarin $b_0 = 0$ en b_i gegeven door (2) van $i \geq 1$

\tilde{x} is dus een geheel getal, waarvan de bits zijn $\dots b_3 \oplus b_2, b_2 \oplus b_1, b_1 \oplus b_0$

Aangezien x oneindig veel bits kan hebben, kan \tilde{x} oneindig worden. We zullen \tilde{x} echter alleen gebruiken in uitdrukkingen als $m \otimes x$, waarin m geheel is en dus eindig veel bits heeft, het logisch produkt dus ook en dus eindig is.

De functie $\text{par}(k)$ (van pariteit) is alleen gedefinieerd als k integer is. Als de bits van k zijn $k_n k_{n-1} \dots k_1 k_0$, dan is $\text{par}(k) =$

$$k_n \oplus k_{n-1} \oplus k_{n-2} \oplus \dots \oplus k_2 \oplus k_1 \oplus k_0 \tag{11c}$$

$\text{par}(k)$ heeft dus de waarde 1 of 0.

Hiermee kunnen we een expliciete oplossing geven voor de vergelijkingen (1). Walsh zelf heeft al een eerste stap gezet met (3b), maar deze vergelijkingen zijn nog recursief. We gaan daarom nog een stap verder. We definiëren

$$\text{wal}(k, x) = \phi_k(x), \quad \text{wal}(0, x) = 1 \quad (12)$$

dan wordt (3c), m.b.v. (2) en (3b)

$$\begin{aligned} \text{wal}(2^{n-1}, x) &= (-1)^{b_n \odot b_{n-1}} \\ \text{wal}(2^{n-1} + k-1, x) &= \text{wal}(k-1, x) \text{wal}(2^{n-1}, x) \end{aligned} \quad (13)$$

Wegens $1 \leq k \leq 2^{n-1}$ kunnen we schrijven

$$k-1 = k_{n-2} 2^{n-2} + k_{n-3} 2^{n-3} + \dots + k_2 2^2 + k_1 2 + k_0$$

De tweede vergelijking van (13) kunnen we dan schrijven:

$$\begin{aligned} \text{wal}(k_{n-1} 2^{n-1} + k_{n-2} 2^{n-2} + \dots + k_0, x) &= \\ \text{wal}(k_{n-2} 2^{n-2} + \dots + k_0, x) \left[\text{wal}(2^{n-1}, x) \right]^{k_{n-1}} \end{aligned} \quad (14)$$

voor $k_{n-1} = 1$. (14) geldt echter ook voor $k_{n-1} = 0$, zoals gemakkelijk blijkt door substitutie.

(14) heeft de vorm van een differentie vergelijking.

Voeren we namelijk de verkorte notatie

$$\begin{aligned} Y_n &= \text{wal}(k_{n-1} 2^{n-1} + k_{n-2} 2^{n-2} + \dots + k_1 2 + k_0, x) \\ C_n &= \{\text{wal}(2^{n-1}, x)\}^{k_{n-1}} = (-1)^{(b_n \odot b_{n-1})} \otimes k_{n-1} \end{aligned}$$

in, dan gaat (14) over in de differentie vergelijking

$$Y_n = Y_{n-1} C_n$$

waarvan de oplossing volgens Milne [9] is.

$$Y_n = Y_1 \prod_{i=2}^n C_i = (-1)^{\sum_{i=1}^n (b_i \oplus b_{i-1}) \otimes k_{i-1}}$$

$$= (-1)^{\text{par}(\tilde{x} \otimes k)}$$

zodat

$$\text{wal}(k, x) = (-1)^{\text{par}(\tilde{x} \otimes k)} \quad (15)$$

Deze formule gebruiken we nu als definitie van $\text{wal}(k, x)$.

Om te laten zien hoe deze formule werkt, geven we eerst een paar voorbeelden,

gevraagd $\text{wal}(5, \frac{3}{16})$

$x = \frac{3}{16} = 0,00110$ (lineair) dus $b_0 = b_1 = b_2 = b_3 = 0, b_4 = b_5 = 1$

$\tilde{x} = \quad 1 \ 0 \ 1 \ 0 \ 0$

$k = \quad 0 \ 0 \ 1 \ 0 \ 1$
 $\hline \quad 0 \ 0 \ 1 \ 0 \ 0 \quad \otimes$

$\text{par}(0 \ 0 \ 1 \ 0 \ 0) = 1$, dus $\text{wal}(5, \frac{3}{16}) = -1$ (vgl. figuur 1)

gevraagd $\text{wal}(6, 0.7)$

$x = \quad 0.10110011001100\dots$

$\tilde{x} = \quad \dots \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1$

$k = \quad 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0$
 $\hline \quad \quad \quad \quad \quad \quad \quad \otimes$

$k \otimes \tilde{x} = \quad 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0$

$\text{par}(k \otimes x) = 0$ dus $\text{wal}(6, 0.7) = 1$.

Sommige auteurs schrijven $\text{wal}_k(x)$ in plaats van $\text{wal}(k,x)$, speciaal als het over rijen van Walsh-funkties gaat. Voor de eerste 8 wal-funkties geeft formule (14) dan

$$w_0(x) = 1$$

$$w_1(x) = (-1)^{b_1}$$

$$w_2(x) = (-1)^{b_2 + b_1}$$

$$w_3(x) = (-1)^{b_2}$$

$$w_4(x) = (-1)^{b_3 + b_2}$$

$$w_5(x) = (-1)^{b_3 + b_2 + b_1}$$

$$w_6(x) = (-1)^{b_3 + b_1}$$

$$w_7(x) = (-1)^{b_3}$$

Ook komt het voor dat men het kleinste vak waarover $wal(k,x)$ konstant is, als eenheid neemt. Dit komt neer op een strekken van de x -as: $h = x \cdot 2^n$. Om verwarring te voorkomen schrijven we hiervoor $WAL(k,h)$ dus $WAL(k,h) = wal(k, x \cdot 2^n)$.

Door uitschrijven kan men aantonen dat

$$k \otimes \left(\frac{\tilde{h}}{2^n}\right) = h \otimes \left(\frac{\tilde{k}}{2^n}\right)$$

en dus

$$WAL(k,h) = WAL(h,k) \tag{16}$$

dus in $WAL(k,h)$ mogen de argumenten verwisseld worden.

Eveneens door uitschrijven kan men aantonen dat

$$h \oplus k = \tilde{h} \oplus \tilde{k}$$

waaruit volgt dat

$$wal_h(x) \cdot wal_k(x) = wal_{h \oplus k}(x) \tag{17}$$

Verder blijkt dat ook de Rademacher-funkties (zie (4)) als macht van (-1) geschreven kunnen worden, nl.

$$R_n(x) = (-1)^{b_n}$$

Hiermee kunnen we aantonen dat onze definitie (15) overeenstemt met de definities van Walsh en van Harmuth. Voor de eerste volgt dit uit (12), waaruit (15) afgeleid was. Uit de definitie van Harmuth volgt nl. de eigenschap

$$wal(2^n - 1, x) = r_n(x), \quad \text{dus}$$

$$wal(2^n - 1, x) = (-1)^{b_n}$$

en omdat $2^n = (2^{n+1} - 1) \oplus (2^n - 1)$ volgt daaruit met (10) dat

$$wal(2^n, x) = (-1)^{b_n \oplus b_{n+1}}$$

wat gelijk is aan de eerste gelijkheid in (13).

4. Afbeelding op een vectorruimte

Bekend is de ontwikkeling van een periodieke functie (met periode = 1) in een Fourierreeks:

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos 2 \pi n t + b_n \sin 2 \pi n t) \quad (1)$$

waarin

$$a_n = 2 \int_0^1 f(t) \cos 2 \pi n t \, dt \quad (2)$$

$$b_n = 2 \int_0^1 f(t) \sin 2 \pi n t \, dt$$

In het verleden is veel onderzoek verricht naar de voorwaarden waaronder (1) geldig is. De resultaten waren weinig doorzichtig. Zo is continuïteit van $f(t)$ alleen niet voldoende, anderzijds zijn er discontinue-functies, zoals de blokfuncties, waarvoor (1) wel geldt. Later heeft men ingezien, dat punt voor punt convergentie van de reeks in (1) niet het geschikte kenmerk was, maar dat convergentie in norm een meer natuurlijk kenmerk levert.

We vatten de in deze paragraaf voorkomende functies $f(t)$, $g(t)$, $h(t)$... op als elementen f , g , h , ... van een abstracte funktieruimte L , en beperken ons tot kwadratisch Lebesque integreerbare functies (hieronder vallen onder meer $\cos 2 \pi n t$ en $\sin 2 \pi n t$). Dan is L een Hilbertruimte [12] met als inwendig produkt

$$(f, g) = \int_0^1 f(t) g(t) \, dt \quad (3)$$

en norm

$$\| f \| = (f, f)^{\frac{1}{2}}$$

Een funktierij $f_0, f_1, f_2, f_3 \dots$ heet convergent in norm, als er bij elke $\varepsilon > 0$ een N bestaat, zodanig dat

$$\| f_n - f \| < \varepsilon \text{ voor } n > N \quad (4)$$

Twee elementen uit L heten onderling loodrecht,

$$f \perp g \text{ als } (f, g) = 0. \quad (5)$$

Een funktierij g_1, g_2, g_3, \dots heet orthonogonaal als

$$(g_m, g_n) = 0 \text{ voor iedere } m \text{ en } n \neq m. \quad (6)$$

Wanneer bovendien $\|g_n\| = 1$ voor elke n , dan heet de rij g_1, g_2, g_3, \dots een orthonormaal stelsel (ONS). Een ONS heet maximaal als het niet uitgebreid kan worden, met andere woorden er bestaat, behalve de nulfunctie, geen element in L dat \perp op alle g_n .

Zij nu $\{g_n\} = g_0, g_1, g_2, g_3, \dots$ een ONS en f een element van L . Dan vormen we de reeks

$$C_0 g_0 + C_1 g_1 + C_2 g_2 + \dots = \sum_0^{\infty} C_n g_n \quad (7)$$

waarin

$$C_n = (f, g_n) \quad (8)$$

Twee vragen doen zich voor:

- a. convergeert de reeks ?
- b. zo ja, is de limiet gelijk aan f ?

ad. a. Uit de orthonormaliteit volgt de ongelijkheid van Bessel:

$$\sum_1^{\infty} |C_n|^2 \leq (f, f) \quad (9)$$

dus de reeks C_0, C_1, C_2, \dots convergeert. Daaruit volgt dan de convergentie van $\sum C_k g_k$. Vraag a kan dus bevestigd beantwoord worden.

ad. b. Ook deze vraag kan bevestigd beantwoord worden, mits $\{g_n\}$ een maximaal ONS is.

Voor een maximaal ONS geldt dus:

$$f = \sum_0^{\infty} C_n g_n, \text{ met } C_n = (f, g_n) \quad (10)$$

Men noemt (10) de Fourier ontwikkeling van f , en C_n de Fouriercoëfficiënten. Voor een andere functie krijgt men andere Fouriercoëfficiënten, wegens (9) convergeren deze reeksen en vormen zij een Hilbertruimte H .

Door (10) is er een afbeelding (of transformatie) gedefinieerd van de ruimte L van de kwadratisch Lebesques integreerbare funkties op de ruimte H van de oneindige reeksen C_n met $|C_n|^2$ convergent. Deze transformatie heet de gegeneraliseerde Fouriertransformatie of kortweg Fouriertransformatie ([6], [7]).

Nu vormen de funkties $V_2 \cos 2 \pi n t$, $V_2 \sin 2 \pi n t$ een ONS en we kunnen in (10) dus substitueren

$$\begin{aligned} g_{2n}(t) &= V_2 \cos 2 \pi n t \\ g_{2n+1}(t) &= V_2 \sin 2 \pi n t \\ C_{2n} &= 1/V_2 a_n \\ C_{2n+1} &= 1/V_2 b_n \end{aligned} \tag{11}$$

en krijgen dan, op de normeringsfactor V_2 na, de formules (1) en (2).

We noemen dit Fouriertransformatie in engere zin.

De sin/cos funkties zijn echter niet het enige ONS systeem. Ook de Walsh-funkties

$$wal_0, wal_1, wal_2, wal_3, \dots \tag{12}$$

vormen zo'n maximaal ONS.

Dat ze een ONS zijn, is door Walsh [14] bewezen.

Om te bewijzen dat dit ONS maximaal is, laten we zien dat een funktie f , die loodrecht staat op alle funkties uit (12), de nulfunctie is. We stellen dus

$$(f, wal_n) = 0 \text{ voor } n = 0, 1, 2, 3, \dots \tag{13}$$

waarbij uiteraard $f \in L$, dus f Lebesque integreerbaar over het segment $(0,1)$. Dan is

$$F(t) = \int_0^t f(x) dx \dots \tag{14}$$

een kontinu funktie. We kiezen $n = 2^m$ (m geheel > 0) (15)

en stellen

$$y_k = \frac{F\left(\frac{k+1}{n}\right) - F\left(\frac{k}{n}\right)}{n}, \quad k = 0, 1, 2, \dots, n-1 \quad (16)$$

dan is

$$\begin{aligned} (f, \text{wal}_j) &= \int_0^1 f(x) \text{wal}_j(x) dx \\ &= \sum_{k=0}^{n-1} \int_{k/n}^{(k+1)/n} \text{wal}_j(x) f(x) dx \\ &= \sum \text{wal}_j\left(\frac{k}{n}\right) \int_{k/n}^{(k+1)/n} f(x) dx \\ &= \sum_{k=0}^{n-1} \text{wal}_j\left(\frac{k}{n}\right) \cdot y_k = 0, \quad k = 0, 1, \dots, n-1 \end{aligned}$$

en dit geldt voor alle j .

We kunnen dit kort schrijven als

$$W \vec{y} = 0 \quad (17)$$

waar W de Walshmatrix is:

$$W_{i,j} = \text{wal}_i\left(\frac{j}{n}\right)$$

en \vec{y} de vector

$$\vec{y} = (y_0 \ y_1 \ \dots \ y_{n-1})^T$$

W is niet singulier, want

$$W^2 = n \cdot I$$

I is de eenheidsmatrix, en dus heeft (17) als enige oplossing de nul-oplossing

$$y_0 = y_1 = y_2 = \dots = y_{n-1} = 0$$

Dan volgt uit (16), dat

$$F(0) = F\left(\frac{1}{n}\right) = F\left(\frac{2}{n}\right) = \dots = F\left(\frac{n-1}{n}\right) = F(1)$$

waar $n = 2^m$ en dit geldt voor alle m .

$F(t)$ is dus konstant op een verzameling die overal dicht is in $(0,1)$, wegens de continuïteit is dus $F(t)$ konstant op $(0,1)$ en dus bijna overal $f = 0$.

Dus is f de nulfunctie q.e.d..

Elke functie $f \in L$ kunnen we dus schrijven als

$$f = \sum_{k=0}^{\infty} a_k \text{ wal}_k \tag{18a}$$

waarin $\sum_{k=0}^{\infty} a_k^2$ convergent is en

$$a_k = (f, \text{wal}_k) \tag{18b}$$

Hiermee is dus een correspondentie gevonden tussen de kwadratisch Lebesques integreerbare functies en de rijen

$$a_0 \ a_1 \ a_2 \quad \text{met} \quad \sum_{k=0}^{\infty} a_k^2 \text{ eindig.}$$

Dit geeft een aanzienlijke vereenvoudiging, want onder de Lebesque integreerbare functies komen zeer ingewikkelde voor, maar deze vereenvoudiging is nog niet voldoende voor ons doel. De reeks bevat nog oneindig veel termen en een rekenmachine kan er slechts een eindig aantal in zijn geheugen opbergen. We kunnen dit oplossen door na een zeker rangnummer n alle termen van de reeks weg te laten. Uiteraard maken we hiermee een fout, maar wegens de convergentie van (18) kunnen we die fout zo klein maken als we zelf willen:

$$\left\| f - \sum_{k=0}^{n-1} a_k \text{ wal}_k \right\| < \epsilon \text{ voor } n > N \tag{19}$$

$$a_k = (f, \text{wal}_k)$$

$\sum_{k=0}^{n-1} a_k \text{wal}_k$ is dus een benadering van f . In [6] is bewezen dat men

met hetzelfde aantal termen geen betere benadering kan vinden, of anders gezegd; als

$$g = b_0 \text{wal}_0 + b_1 \text{wal}_1 + \dots + b_{n-1} \text{wal}_{n-1}$$

dan is $\|f - g\|$ minimaal voor

$$b_k = (f, \text{wal}_k) = a_k \tag{20}$$

We kunnen dit ook nog a.v. formuleren: Bij gegeven aantal termen levert (19) de beste benadering volgens de methode van de kleinste kwadraten. Uiteindelijk hebben we dus de ruimte L afgebeeld op de vectorruimte V_n , de ruimte van de n dimensionele vectoren. Deze afbeelding is niet 1 op 1, want alle functies waarvan de eerste n coëfficiënten van (18) gelijk zijn worden op dezelfde vector $(a_0 \ a_1 \ \dots \ a_{n-1})$ afgebeeld. Uit (19) volgt echter dat die functies (in norm) slechts weinig van elkaar verschillen.

Bewerkingen op functies $f \in L$ hebben hun pendant in V_n ; in vele gevallen blijkt dat een matrix bewerking te zijn. Iets concreter gezegd: Laat $f \in L$ en f wordt afgebeeld op $\vec{a} \in V_n$, waarin

$$\vec{a} = (a_0, a_1, \dots, a_{n-1})$$

dan schrijven we

$$f \rightarrow \vec{a}$$

Zij verder T een operator en $T f = g$, waarbij

$$g \rightarrow \vec{b} = b_0, b_1, \dots, b_{n-1}$$

dan korrespondeert met T een operator M , zodanig dat

$$\vec{a} M = \vec{b} T.$$

In de toepassingen die in de hierna volgende paragrafen gegeven worden, zal blijken dat M een matrix is en matrix bewerkingen lenen zich uitstekend om door computers uitgevoerd te worden.

Opmerking: Als f een stapfunctie is, dan kan het inwendig product (f, wal_k) in 19 vervangen worden door

$$a_k = \frac{1}{n} \sum_{m=0}^{n-1} f_m wal_k(m/n) \quad (21)$$

waarin $f_0 f_1 \dots f_{n-1}$ de waarden van f in de respectievelijke intervallen voorstelt. De W F T kan dan ook in matrix vorm geschreven:

$$\vec{a} = \frac{1}{n} W \vec{f} \quad (22)$$

waarin W de Walsh matrix is. De inverse transformatie luidt

$$\vec{f} = W \vec{a} \quad (23).$$

5. De Walsh Fourier Transformatie (W F T)

5.1. Algemeen

In de vorige paragraaf (par. 4, (18)) hebben we gezien dat we een functie $f(t)$ kunnen ontwikkelen in een reeks

$$f(t) = \sum_{k=0}^{\infty} a_k \text{wal}_k(t); a_k = (f, \text{wal}_k); k = 0, 1, 2, \dots$$

Harmuth (7) noemt dit de Walsh Fourier Transformatie. De "gewone" Fourier-transformatie, die we krijgen als we $\text{wal}_k(t)$ vervangen door $\exp(2\pi ikt)$, noemen we Fourier transformatie in engere zin. Beide transformaties bezittten een aantal overeenkomstigheden, zoals ook vermoed kan worden uit de overeenkomst van de functies $\text{wal}(2k,t)$ en $\exp(2\pi ikt)$ (par. 3.4.). Voor beide bestaat er een snelle versie, voor de Fourier-transformatie in engere zin noemen we dit de F F T (Fast Fourier Transform) en voor de Walsh Fourier Transformatie noemen we dit de W F T, waarbij de F naar keuze voor "Fast" of "Fourier" (of beide) staat.

Zowel de F F T als de W F T berusten op wat men wel de "vlinderoperatie" noemt, zie fig. 3 en 4 en [13].

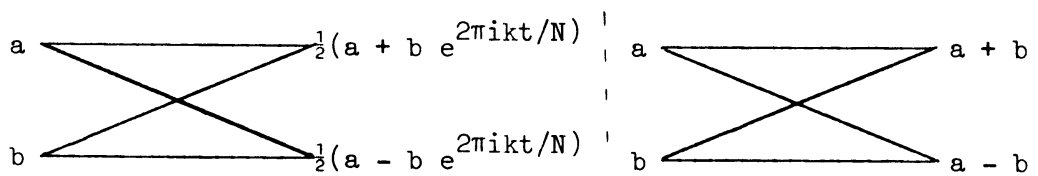


Fig. 3 F F T "vlinder"

Fig. 4 W F T "vlinder"

De betekenis hiervan is dat de getallen rechts worden berekend uit de getallen links. De Fouriercoëfficiënten worden dan verkregen door $\frac{1}{2} N \cdot \log_2 N$ vlinderbewerkingen uit te voeren [13]. Men ziet duidelijk dat de W F T minder rekenwerk vergt dan de F F T: Het verschil is een complexe vermenigvuldiging per vlinderoperatie, wat gelijk

staat met 4 vermenigvuldigingen en twee optellingen van reële getallen. Maar als men digitaliseert volgens de voorgaande paragrafen, dan kan men de optelling en aftrekkingen in de W F T in vaste komma uitvoeren wat ook weer minder rekentijd kost dan de drijvende komma bewerkingen die voor de F F T nodig zijn. De winst in tijd is zelfs zeer aanzienlijk voor rekenmachines, die geen hardware drijvende komma apparatuur bezitten.

Voor een nadere uitwerking van de W F T veronderstellen we dat de te transformeren functie gegeven is als de elementen van een dimensionaal array, zeg $A[0:n-1]$, waarin $n = 2^m$.

Bij de F F T komt men in de literatuur verschillende methoden tegen, die verschillen in de manier waarop de vlinder-bewerkingen gerangschikt zijn. Ditzelfde geldt voor de W F T. De verschillende methoden kunnen we indelen in twee groepen:

In de eerste groep worden de bewerkingen "in situ" uitgevoerd, dat wil zeggen de gevonden getallen worden weer in hetzelfde array A ingevuld. In Algol:

$$\delta f \ a: = A[k]; \ b: = A[l]; \ A[k]: = a + b; \ A[l]: = a - b; \quad (1)$$

$$\delta f \ a: = A[k]; \ b: = A[l]; \ A[k]: = a - b; \ A[l]: = a + b; \quad (2)$$

In de tweede groep is een tweede array $B[0:n-1]$ nodig, de vlinder-bewerking wordt dan

$$a: = A[k]; \ b: = A[l]; \ B[p]: = a + b; \ B[g]: = a - b; \quad (3)$$

In de eerste groep komen de Fourier-coëfficiënten er in "invers-binaire volgorde" uit, in de tweede groep kan men bereiken, dat de coëfficiënten er in de normale volgorde uitkomen.

De prijs die men daarvoor betaalt is meer geheugen en vooral meer administratieve bewerkingen. Daarom is voor de eerste methode gekozen.

5.2. W F T in één array

We voeren in twee operatoren P_m en Q_m , die elk werken op een array van $n = 2^m$ elementen:

$$(x_0, x_1 \dots x_{n-1}) P_m = (y_0, y_1 \dots, y_{n/2-1}, z_{n/2-1} \dots, z_1, z_0) \quad (4)$$

waarin

$$\begin{aligned} y_i &= x_i + x_{n-1-i} \\ z_i &= x_i - x_{n-1-i} \end{aligned} \quad (5)$$

$$(x_{n-1}, x_{n-2} \dots x_1, x_0) Q_m = (y_0, y_1 \dots y_{n/2-1}, z_{n/2-1} \dots z_1, z_0) \quad (6)$$

De uitkomsten in (4) en (6) zijn dus gelijk, maar Q_m werkt op de elementen in omgekeerde volgorde als die in P_m .

We berekenen nu de W F T in stappen:

1e stap:

x_0		y_0
		.
x_1		.
.	$P_m =$	
.		$y_{n/2-1}$
.		.
.		$z_{n/2-1}$
.		.
x_{n-1}		z_0

2e stap: op bovenste helft de operator P_{n-1} , op onderste helft

	Q_{m-1}	
y_0		$y_0 + y_{n/2-1}$
		.
y_1		.
.		.
.	$P_{m-1} =$	$y_{n/4-1} + y_{n/4}$
.		.
.		$y_{n/4-1} - y_{n/4}$
.		.
$y_{n/2-1}$		$y_0 - y_{n/2-1}$

(8)

$$\begin{array}{r}
 z_{n/2-1} \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 z_1 \\
 z_0
 \end{array}
 =
 \begin{array}{r}
 z_0 + z_{n/2-1} \\
 \cdot \\
 z_{n/4-1} + z_{n/4} \\
 z_{n/4-1} - z_{n/4} \\
 \cdot \\
 z_0 + z_{n/2-1}
 \end{array}$$

Het resultaat kunnen we nu in 4 delen verdelen, waarop achtereenvolgens P_{m-2} , Q_{m-2} , P_{m-2} , Q_{m-2} worden toegepast enz.

Als we de vectoren eenvoudigheidshalve door een verticale streep weergeven, dan krijgen we onderstaand schema

1e stap	2e stap	3e stap	
$\left \begin{array}{c} P_m \end{array} \right.$	$\left \begin{array}{c} P_{m-1} \\ Q_{m-1} \end{array} \right.$	$\left \begin{array}{c} P_{m-2} \\ Q_{m-2} \\ P_{m-2} \\ Q_{m-2} \end{array} \right.$	$\left \begin{array}{c} P_{m-3} \\ Q_{m-3} \\ P_{m-3} \\ Q_{m-3} \\ P_{m-3} \\ Q_{m-1} \\ P_{m-3} \\ Q_{n-1} \end{array} \right.$ etc.

(9)

Het proces eindigt na m stappen, want dan korrespondeert elk verticale streep met 1 element en is verdere splitsing niet meer mogelijk.

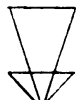




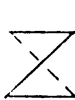

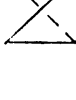
Laat nu $\vec{a} = (a_0, a_1, \dots, a_{n-1})$ de W F T (Walsh Fourier Transform) van $\vec{x} = (x_0, x_1, \dots, x_{n-1})$ zijn. We veronderstellen nu dat P_{m-1} en volgende transformaties de W F T coëfficiënten in invers binaire volgorde geven, waarbij er 2^{m-1} coëfficiënten zijn, dus 2^{m-1} bits.

Als we nu in $y_{P_{m-1}}$ invullen $y_0 = x_0 + x_n$, $y_1 = x_1 + x_{n-1}$, etc. volgens (2), dan zien we dat aan elk der elementen $x_0 \dots x_{n/2-1}$ juist de gespiegelde tov $t = \frac{1}{2}$ wordt toegevoegd, dat betekent dat aan de index (in binaire vorm van a_i) een bit = 0 toegevoegd wordt.

Bij Q_{m-1} echter moeten we substitueren $Z_0 = x_0 - x_2$ etc., ook nu weer wordt de gespiegelde tov $t = \frac{1}{2}$ toegevoegd, maar nu met verandering van teken, d.w.z. er moet nu een bit = 1 toegevoegd worden.

Tezamen geeft dit juist de invers binaire volgorde van m bits, waarbij we evenwel verondersteld hadden dat het voor $n = 2^{m-1}$ al bewezen was. We zullen nu laten zien dat het voor $m=2$ geldt, dan is het bewijs door volledige inductie geleverd.

De W F T voor $m=2$, dus voor een vector met 4 componenten, ziet er als volgt uit:

x_0		$x_0 + x_3$		$x_0 + x_1 + x_2 + x_3 = (W_0, x)$	BC = binaire coëff. v. W_i 00
x_1		$x_1 + x_2$		$x_0 - x_1 - x_2 + x_3 = (W_2, x)$	10
x_2		$x_1 - x_2$		$x_0 + x_1 - x_2 - x_3 = (W_1, x)$	01
x_3		$x_0 - x_3$		$x_0 - x_1 + x_2 - x_3 = (W_3, x)$	11

(10)

Met behulp van de operatoren P en Q kunnen we dit schrijven

x_0	y_0	P_1	$y_0 + y_1 = (W_0, y)$	0	(11)
x_1	y_1		$y_0 - y_1 = (W_1, y)$	1	
x_2	z_1	Q_1	$z_0 + z_1 = (W_0, z)$	0	
x_3	z_0		$z_0 - z_1 = (W_1, z)$	1	

waarin $y_i = x_i + x_{n-i}$ $i = 0, 1$ (12)
 $z_i = x_i - x_{n-i}$

dus i loopt van 0 tot $n-1$, omdat $n=2$.

Wanneer we nu (12) substitueren in (11), dan komt er

$$\begin{array}{rcl} & & \text{BC} \\ y_0 + y_1 = x_0 + x_1 + x_2 + x_3 = (W_0, x) & 00 \\ y_0 - y_1 = x_0 - x_1 - x_2 + x_3 = (W_2, x) & 10 \\ z_0 + z_1 = x_0 + x_1 - x_2 - x_3 = (W_1, x) & 01 \\ z_0 - z_1 = x_0 - x_1 + x_2 - x_3 = (W_3, x) & 11 \end{array} \quad (13)$$

We zien: achter de bovenste helft van (11) wordt er in (13) een bit 0 toegevoegd, achter de onderste helft een bit 1 en zo krijgen we de coëfficiënten in de invers lineaire volgorde (voor de normale volgorde zouden de enen en nullen juist ervoor in plaats van er achter geplaatst moeten worden).

5.3. Een W F T programma voor microcomputers

Om misverstand te voorkomen merken we alvast op, dat men niet een uitgeschreven programma voor een of andere microcomputer moet verwachten. Dat zou te specifiek voor één bepaald type/merk zijn. Daarom is het programma in Algol geschreven, maar in een zo eenvoudige vorm, dat vertaling naar machinecode van een microcomputer weinig problemen zal opleveren.

Wanneer we de methode van paragraaf 5.2. als uitgangspunt kiezen, dan krijgen we een programma met 3 lussen (zie (9)):

een buitenlus voor de iteratie stappen

een middenlus voor de operatoren P en Q

een binnenlus, korreponderend met de vlinder van fig. 4.

De binnenlus, die dus het meest vaak doorlopen wordt, bevat aan rekenkundige bewerkingen slechts één optelling en één aftrekking, een vermenigvuldiging is niet nodig. Zoals reeds is opgemerkt (paragraaf 5.1.) geeft dit een aanmerkelijke besparing in rekentijd tov de F F T.

Maar het geeft ook een aanzienlijke besparing in apparatuur: een drijvende komma module is niet vereist, en zelfs een gewone vermenigvuldiger (vaste komma) is niet nodig. De (weinig) voorkomende delingen door 2 kunnen door een "rechtse uitschuif" geëffectueerd worden.

Om de rekenkundige bewerkingen in de juiste volgorde te laten verlopen zijn, zoals bij elk programma, uiteraard een aantal administratieve handelingen nodig, zoals testen, transporten van het geheugen naar de accumulator en omgekeerd etc. Het is zaak deze bijkomende handelingen zoveel mogelijk te beperken, want de tijd van één zo'n bewerking is vergelijkbaar met die van een optelling met vaste komma. In het bijzonder moeten het ophalen en opbergen van array elementen zoveel mogelijk beperkt worden, want deze bewerkingen vergen relatief veel tijd. Omdat een vlinder 4 getallen bevat, die elk met een array-element korresponderen, is 4 het minimale aantal array-elementen dat nodig is. Dit minimum wordt bereikt in (1), (2) en (3). In (3) zijn echter in het algemeen 4 adres berekeningen nodig, in (1) of (2) slechts 2 en daarom is voor dit laatste gekozen.

We zouden (1) en (2) in één formule kunnen samenvatten, zoals gebeurd is in [2] m.b.v. een hulpvariable Z en een extra loop maar dit vergt uiteraard extra tijd, zodat hiervan is afgezien. Bovendien blijkt deze tekenwisseling automatisch ingebouwd te worden wanneer we de hieronder volgorde wijziging in de volgorde van de vlinders invoeren.

Wanneer we operator P en Q (zie (9)) stuk voor stuk afwerken dan hebben we niet alleen tellertjes voor de vlinders nodig, maar ook de grenzen moeten voor elke P en Q apart berekend worden en voor elke stap zijn de grenzen anders.

Wanneer we echter van elke P en Q eerst alle 1e vlinders afwerken, daarna alle 2e enz., dan is de grens vast, voor elke P en Q en elke stap n_l gelijk aan n. We moeten afwisselend een vlinder uit P en Q behandelen (uitgezonderd de 1e stap die apart geprogrammeerd is) en dit correspondeert met het afwisselend toepassen van (1) en (2), die daarna na elkaar in het programma opgenomen zijn en aldus afwisselend uitgevoerd worden.

Een voorbeeld van een programma dat volgens deze richtlijnen is opgezet, volgt hieronder:

```
begin comment programma W F T;  
integer a, b, j, k, l, m, n, nk, nl, kj;  
comment hier wordt aan m een waarde toegekend, b.v. door  
een assignment statement of een leesopdracht;  $n = 2^m$ ;  
begin integer array X [ 0 : n-1 ];  
comment hier moeten de getallen waarop meer de W F T  
wil toepassen, ingevuld worden in het array X;  
nl: = n; nk: = n/2; k: = 0;  
HH: l: = nl-1-k; a: = X[k]; b: = X[l];  
X[k]: = a + b; X[l]: = a - b; k: = k + 1;  
if k < nk then goto HH;  
j: = 1;  
JJ: nl: = nk; nk: = nk/2; kj: = 0;  
KK: k: = kj; l: = -1-k;  
LL: l: = l+nl; a: = X[k]; b: = X[l]; X[k]: = a+b; X[l]: = a-b;  
k: = k+nl; l: = l+nl;  
a: = X[k]; b: = X[l]; X[k]: = a+b; X[l]: = b-a;  
k: = k+nl; if k < n then goto LL;
```

$kj: = kj+1$; if $kj < nk$ then goto KK;

$j: = j+1$; if $j < m$ then goto JJ;

comment. Hier komt een uitover procedure voor de elementen $X[k]$. Voor de direkte W F T moeten ze eerst nog gedeeld worden door n , voor de inverse W F T vervalt die deling;

end

end

Zoals reeds opgemerkt is, berust het programma op de methode van paragraaf 5.2. en komen de W F T coëfficiënten er in invers binaire volgorde uit. Men zou dus aan het eind van het programma een voorziening kunnen verwachten om de coëfficiënten te herordenen in de normale volgorde. Dit is niet gedaan om de volgende redenen:

- a. Er bestaan reeds een aantal programma's voor deze herordening [13]. Weliswaar zijn deze voor de F F T geschreven, maar voor de herordening maakt het geen verschil of de coëfficiënten door de F F T dan wel de W F T verkregen zijn.
- b. In een aantal gevallen is herordening niet nodig, b.v. als van de W F T weer een inverse W F T volgt. Aan deze laatste transformatie worden dan de gegevens in invers binaire volgorde aangeboden en dan komen ze er in de normale volgorde uit. Dit kan voorkomen o.a. bij het digitaal overbrengen van beelden. Het aantal over te seinen bits kan hierbij vaak aanzienlijk gereduceerd worden door voor verzending een W F T toe te passen. Aan de ontvangtzijde past men dan de inverse W F T toe.

5.4. Nauwkeurigheid en benodigde apparatuur

In het voorgaande is al gesuggereerd, dat men de W F T op een microcomputer kan verwerken. We zullen dit nader toelichten aan de hand van een voorbeeld. Bijzonder gunstig komt het uit, als we de W F T toepassen op een array van 256 getallen van 8 bits elk en we de beschikking hebben over een microcomputer met een woordlengte van 16 bits en vaste komma opstelling en aftrekking. Dan is $m=8$ en moeten we dus 8 stappen uitvoeren (paragraaf 5.3.).

Wanneer we twee getallen van 8 bits optellen, dan heeft het resultaat maximaal 9 bits, na 8 stappen krijgen we dus getallen van maximaal 16 bits en alle bewerkingen kunnen dus exact, zonder afrondingen e.d. uitgevoerd worden. Ook in dit opzicht is de W F T dus gunstiger dan de F F T, waar in bijna elke vlinderoperatie afgerond moet worden, waardoor extra fouten in het eindresultaat ontstaan [13].

Met 8 bits kan men de integers van 0 t/m 255 exact weergeven, de reeks van 0 tot 255 met een fout van hoogstens 0,5 (na afronden op de dichtst bij gelegen integer), dat is dus circa 0,2% van de maximale waarde. Heeft men aan deze nauwkeurigheid niet genoeg, of wil men langere reeksen dan 256 getallen bewerken, dan kan men verschillende dingen doen:

- overgaan op dubbele woordlengte;
- afronden wanneer een tussenresultaat meer dan 16 bits dreigt te krijgen;
- splitsen van de reeks in deelreeksen [8];
- een andere computer kiezen.

De totale bewerkingsduur kan hierdoor wat groter worden, het voordeel dat geen drijvende komma-bewerkingen nodig zijn blijft.

5.5. W F T versus F F T

De aanzienlijke besparing aan rekentijd en apparatuur van de W F T t.o.v. de F F T doet de vraag rijzen, of we de F F T door de W F T kunnen vervangen. Het antwoord hangt van de omstandigheden af.

Wanneer men geïnteresseerd is in de sinusvormige componenten, dan zal de F F T de voorkeur verdienen. Wanneer men van de sin/cos functies echter alleen de orthogonaliteit gebruikt, dan kan men ook een ander orthogonaal stelsel gebruiken, en dan verdienen Walsh-functies de voorkeur wegens de genoemde besparing aan rekentijd. Dit is b.v. het geval bij het overbrengen van gesproken tekst. Mathematisch gezien kan men de functie die de luchtdrukveranderingen aangeeft, ontwikkelen in een Fourier reeks (engere sin) en het geluid dus opvatten als een som van sinusvormige componenten. Fysisch gezien is er echter geen overweldigende reden om nu juist sinusvormige componenten te kiezen. Weliswaar kan men een lang aangehouden klinker redelijk benaderen met de som van enkele sinusvormige trillingen, maar dit geldt niet voor medeklinkers, en zeker niet voor de "plofklanken" zoals p, t en k. Voor het overbrengen van spraak gaat men dan ook steeds meer, en met succes, over tot PCM (Pulse Code Modulatie) en daarvoor bieden Walsh-functies zich aan als de natuurlijke componenten.

Een veelvuldig gebruik van de F F T is het bepalen van het energie frequentie spectrum. Op dezelfde manier kan men met de W F T het segmentie energie spectrum bepalen. Deze twee zijn niet aan elkaar gelijk, maar vertonen wel veel overeenkomst [2]. We zullen dit aan een paar eenvoudige voorbeelden laten zien. Als eerste ontwikkelen

we $w_{a1}(t)$ in een Fourier reeks. Omdat $w_{a1}(t)$ een oneven funktie is, komen er alleen sinuscomponenten:

$$w_{a1}(t) = \sum b_n \sin 2\pi nt \quad (14)$$

Met behulp van [3] of door rechtstreeks uitrekenen vindt men

$$b_{2m} = 0, \quad b_{2m+1} = \frac{4}{\pi} \cdot \frac{1}{2m+1}$$

en dus het frequentie energiespectrum (fig. 5b) is

$$(b_{2m+1})^2 = \frac{16}{\pi^2} \cdot \frac{1}{(2m+1)^2}$$

De totale energie is

$$\sum_{m=0}^{\infty} b_{2m+1}^2 = \frac{16}{\pi^2} \cdot \sum \frac{1}{(2m+1)^2} = 2$$

terwijl men 1 zou verwachten. Dit komt doordat de sinus funkties niet genormaliseerd zijn. Men kan ze normaliseren door toevoegen van een factor V_2 (zie paragraaf 4, (11)), dan wordt (14)

$$w_1(t) = \sum c_n (V_2 \cdot \sin 2\pi nt)$$

met

$$c_n = \frac{b_n}{V_2}$$

en dan is $\sum c_n^2 = 1$.

Als tweede voorbeeld nemen de funkties $V_2 \cdot \cos 2\pi t$, waarvan het frequentiespectrum uit één enkele piek bestaat, en bepalen hiervan het sequentiespectrum. We moeten dan weer digitaliseren, en stellen $x(t) = V_2 \cdot \cos((2k+1) \pi/16)$, waarbij het val(0,1) dus is verdeeld in 16 subvakken. Met behulp van het programma uit paragraaf 5.3. bepalen we Wals-Fouriercoëfficiënten, de kwadraten hiervan vormen het sequentiespectrum (fig. 6).

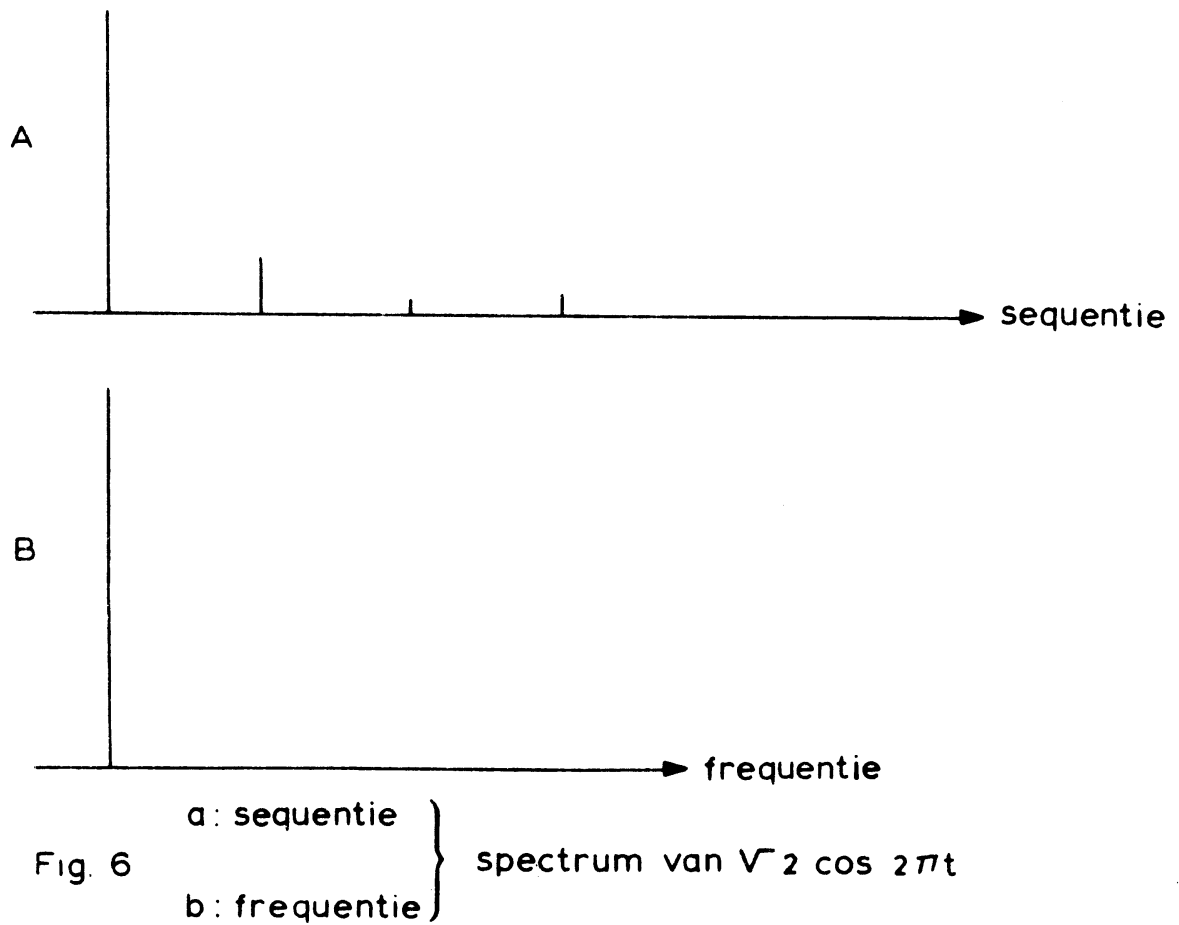
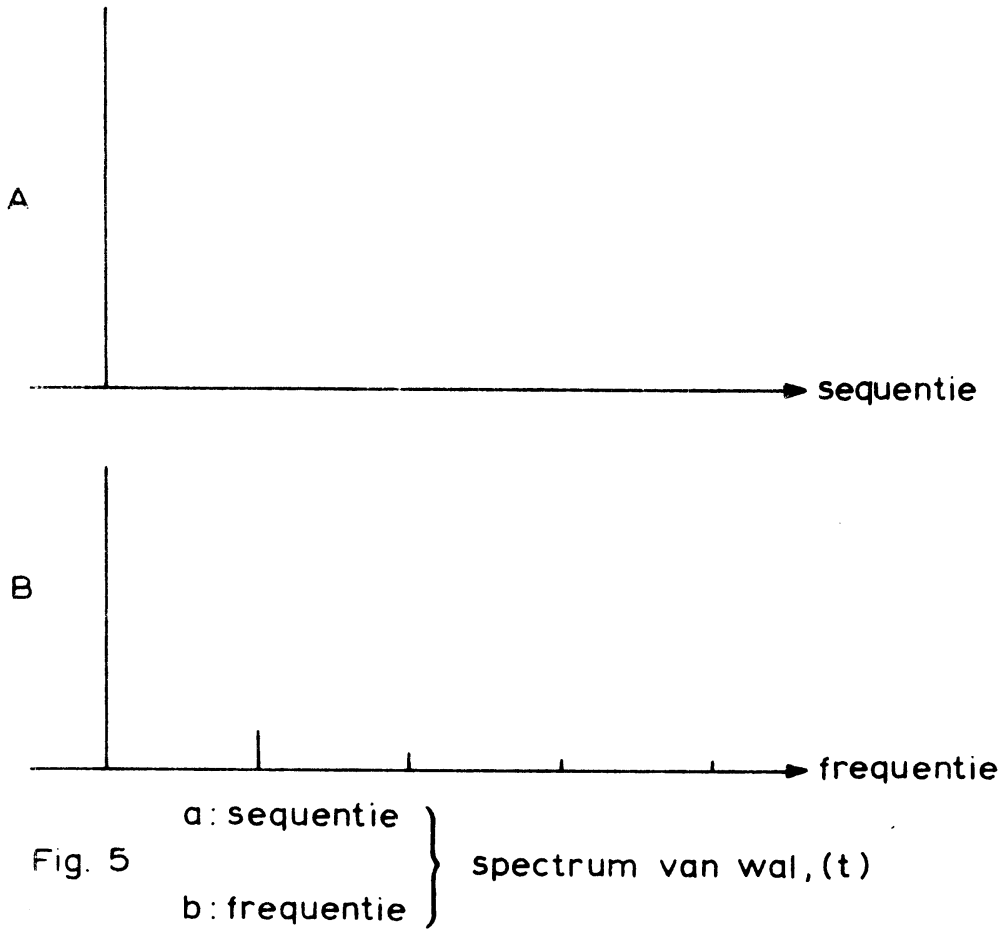


Fig. 5 en fig. 6 zijn in zekere zin elkaars tegenpolen: in fig. 5 correspondeert één enkele sequentie met een aantal frequenties, in fig. 6 corresponderen een aantal sequenties met één enkele frequentie. Het algemene geval zal tussen deze beide uitersten liggen. Overigens zijn de afwijkingen klein. Zelfs als de F F T de juiste is en niet de W F T (wat nog bewezen moet worden) dan kan de W F T voor bewaking dienen (b.v. tijdens meetcampagnes op de oceaan) om ter plaatse maatregelen te nemen die achteraf niet meer mogelijk zijn.

6. Differentieren en integreren van Walsh-funkties

6.1. Differentieren

6.1.1. Klassiek differentieren van een Walsh-functie geeft als afgeleide een functie, die bijna overal nul is, die dus identiek aan de nulfunctie is in de zin van paragraaf 4. Hier kunnen we dus weinig mee beginnen.

6.1.2. Via differentie quotient kunnen we definiëren

$$\frac{d \text{wal}(t)}{d t} \cong \frac{\text{wal}(t+h) - \text{wal}(t-h)}{2h}$$

waarin $h = 2^{-n}$ het basisinterval voor $\text{wal}(t)$ is. Ook dit geeft geen bevredigend resultaat. B.v. voor die Walsh-funkties, die gelijk zijn aan een Rademacher-functie (zie paragraaf 3), dat zijn $\text{wal}_1(t)$, $\text{wal}_3(t)$, $\text{wal}_7(t)$ enz. is het differentie-quotient identiek = 0, zodat deze componenten verdwijnen bij het differentieren, wat niet de bedoeling is.

6.1.3. Mirza en Bansal [10] definieren differentiatie door middel van een logische operator

$$\begin{aligned} D \text{ sal}(k, t) &= k \text{ cal}(k, t) \\ D \text{ cal}(k, t) &= k \text{ sal}(k, t) \end{aligned} \tag{1}$$

en leiden daaruit af een formule voor het differentieren van een gegeven functie $f(t)$:

$$D f(t) = M.F \tag{2}$$

waarin M een matrix is die het produkt is van 3 stuk voor stuk

ingewikkelde matrices en F een kolomvector die verkregen wordt door de functie $f(t)$ in $n=2^m$ punten te bemonsteren.

Uit (2) leiden zij een schema af, dat veel lijkt op eerst toepassen van de W F T, daarna vermenigvuldigen met de coëfficiënten k en dan toepassen van de inverse W F T. We kunnen dit ook rechtstreeks afleiden, zonder de ingewikkelde matrix M , waarbij dan tevens een paar schoonheidsfoutjes aan het licht komen. We schrijven daartoe

$$f(t) = \sum_{k=1}^{n/2} (a_{k-1} \text{cal}_{k-1}(t) + b_k \text{sal}_k(t)) \quad (3)$$

met $n=2^m$, dan volgt daaruit, (als we de operator D lineair veronderstellen)

$$D f(t) = \sum_{k=1}^{n/2} (k-1) a_{k-1} \text{sal}_{k-1}(t) + k a_k \text{cal}_k(t) \quad (4)$$

Hierin komen voor de functies sal_0 en $\text{cal}_{n/2}$. De overige sal en cal functies kunnen, evenals $f(t)$, volgens paragraaf 4 afgebeeld worden op een n dimensionele vectorruimte V^n ; $\text{cal}_{n/2}$ valt hier buiten. Willen we $\text{cal}_{n/2}$ handhaven, dan moeten we in een ruimte V^{2n} werken, waardoor de dimensie van alle matrices en vectoren tweemaal zo groot wordt, met alle gevolgende van dien (meer rekentijd, meer geheugen etc.).

Mirza en Bansel laten daarom $\text{cal}_{n/2}$ weg, evenals sal_0 (die overigens niet gedefinieerd is), maar dat betekent een projectie van V^n op een V^{n-2} , dus verlies aan informatie.

Zij gebruiken deze methode voor het konstrueren van sequentiefilters. Door de overeenkomst tussen (1) en de korresponderende formules voor \sin en \cos , verkrijgen ze goede resultaten voor deze toepassing. Voor het bepalen van de afgeleide van een functie is hun methode weinig geschikt, zoals uit een eenvoudig voor-

beeld blijkt.

$$\begin{aligned} \text{Nemen we } f(t) = \frac{1}{2} \text{cal}_0(t) - \frac{1}{4} \text{sal}_1(t) - \frac{1}{8} \text{sal}_2(t) - \\ \frac{1}{16} \text{sal}_4(t) - \frac{1}{32} \text{sal}_8(t), \end{aligned} \quad (5)$$

dan is $f(t)$ een stapfunctie (fig. 7).

Uit (4) volgt dan

$$D f(t) = -\frac{1}{4} (\text{cal}_1(t) + \text{cal}_2(t) + \text{cal}_4(t)) \quad (6)$$

(zie fig. 8).

Aangezien we (5) op kunnen vatten als een benadering van de functie $f(t) = t$, verwachten we als afgeleide iets dat lijkt op $f(t) = 1$. $D f(t)$ lijkt hier echter niet op, zodat we deze methode niet kiezen voor het bepalen van de afgeleide functie.

6.1.4. Als omgekeerde van integreren.

In paragraaf 6.2. zullen we aantonen, dat we integreren van Walsh functies door een matrix kunnen weergeven. De matrix is niet singulier en heeft dus een inverse. Deze inverse matrix gebruiken we voor differentieren.

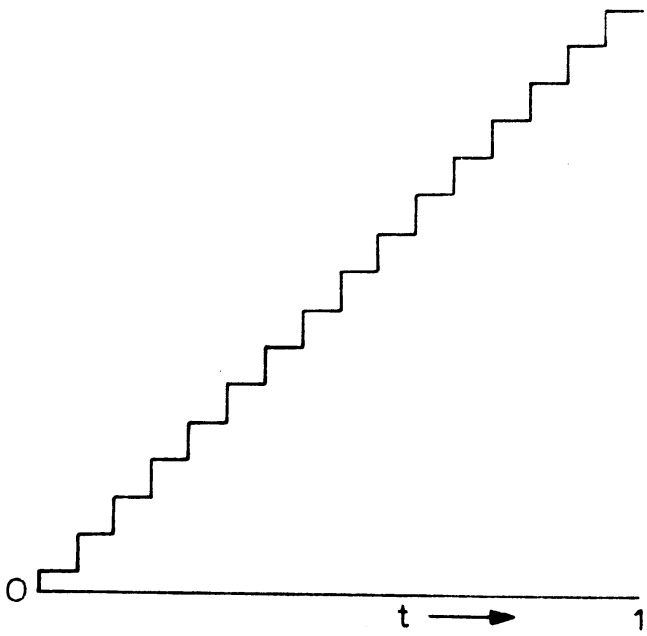


Fig. 7. $x = f(t)$

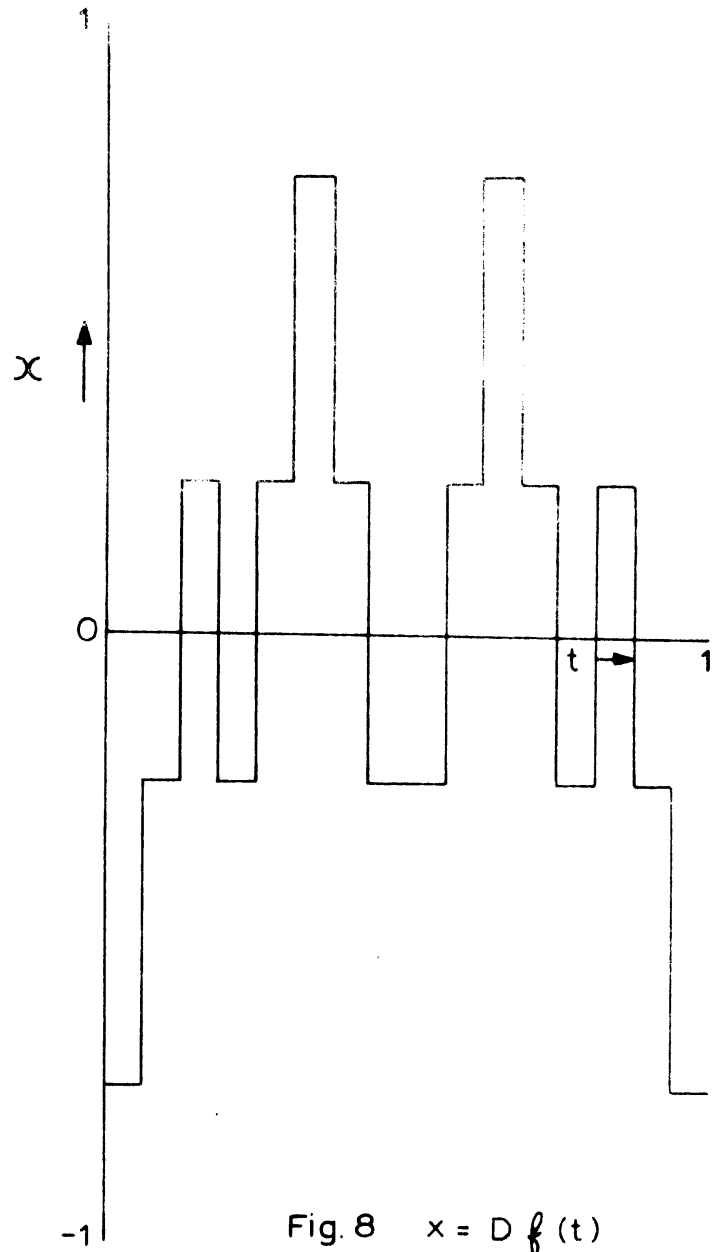


Fig. 8 $x = D f(t)$

6.2. Integreren

Walsh-funkties zijn continue, behalve in een eindig aantal diskontinuiteitspunten, en dus integreerbaar. De analytische uitdrukking voor $\int_0^t \text{wal}_k(x) dx$ wordt echter nogal gekompliceerd voor grote k vanwege het grote aantal sprongpunten. Daarom gaan we digitaliseren naar t en geven

$$f_k(t) = \int_0^t \text{wal}_k(x) dx \quad (7)$$

weer als een vector

$$\vec{f}_k = (f_{1,k} \ f_{2,k} \ \dots \ f_{n,k}) \quad , \ k = 0, 1, 2, \dots, n-1 \quad (8)$$

waarin

$$f_{j,k} = n \int_{(j-1)/n}^{j/n} f_k(t) dt \quad \text{met } n=2^m, \ j = 1, 2, \dots, n \quad (9)$$

het gemiddelde is van $f_k(t)$ over het j^e subinterval.

Voor $m = 3$ krijgen we dan voor de eerste 8 Wals-funkties de volgende tabel, waarbij de integraal met 16 is vermenigvuldigd om een overmaat van breuken te vermijden:

Tabel 1

k	$16 \times \int_0^t \text{wal}_k(x) dx, \text{ gedigitaliseerd}$							
0 0 0	1	3	5	7	9	11	13	15
0 0 1	1	3	5	7	7	5	3	1
0 1 0	1	3	3	1	-1	-3	-3	-1
0 1 1	1	3	3	1	1	3	3	1
1 0 0	1	1	-1	-1	1	1	-1	-1
1 0 1	1	1	-1	-1	-1	-1	1	1
1 1 0	1	1	1	1	-1	-1	-1	-1
1 1 1	1	1	1	1	1	1	1	1

Met behulp van (7), (8), (9) en paragraaf 3.5. kunnen we deze tabel regel voor regel uitrekenen, maar het kan ook a.v. De eerste

en de derde regel van (1), paragraaf 3.2. laten zich aldus samenvatten:

$$\text{wal}_{2^{j+1}}(t) = \text{wal}_{2^j}(t) = \text{wal}_j(2t) \text{ voor } 0 < t < \frac{1}{2} \quad (10)$$

Uit $\text{wal}_0(t) = 1$ volgt met (7)

$$f_0(t) = t$$

en daaruit met (8) en (9)

$$f_{j,0} = \frac{2^j - 1}{16}, \quad j = 1, 2, \dots, n$$

waaruit de eerste regel van tabel 1 volgt. Met (10) kunnen we dan de getallen links van de stippellijn invullen. Uit [7] volgt dat

$$\text{wal}_{2^j}(t) \text{ is een even funktiet.o.v. } t = \frac{1}{2}$$

$$\text{wal}_{2^{j+1}}(t) \text{ is een oneven funktiet.o.v. } t = \frac{1}{2}$$

Door (10) nogmaals toe te passen volgt

	oneven	t.o.v. $t = \frac{1}{2}$	voor $q = 1$
	even	t.o.v. $t = \frac{1}{2}$	voor $q = 0$
$\text{wal}_{4^{j+2p+q}}$	is	oneven	t.o.v. $t = \frac{1}{4}$ voor $p = 1$
		even	t.o.v. $t = \frac{1}{4}$ voor $p = 0$

p en q zijn de laatste 2 bits van $k = 4j+2p+q$.

Zo doorgaande vindt men dat

$$\text{wal}_k(t) \text{ is } \begin{cases} \text{even t.o.v. } t = 2^{-i-1} & \text{als } k_i = 0 \\ \text{oneven t.o.v. } t = 2^{-i-1} & \text{als } k_i = 1 \end{cases}$$

waarin k_i de i te bit van k is.

Bijvoorbeeld, met $5 = 101$ binair,

$$\text{wal}_5 = \begin{array}{ccc} \begin{array}{c} a \quad s \quad a \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \end{array} & \begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} & \begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \\ = +1 \cdot -1 \cdot -1 & +1 \cdot -1 & +1 \quad +1 \quad -1 \end{array}$$

waar s staat voor een spiegeling zonder tekenwisseling (even functie) a voor een spiegeling met tekenwisseling (oneven functie). Uit de lineaire representatie van k kan men dus zo wal_k opschrijven. Dit gebruiken we om de getallen rechts van de stippellijn in tabel 1 in te vullen, waarbij we moeten bedenken dat door het integreren even en oneven verwisselen. Dus b.v. de 2e regel van tabel 1 is even t.o.v. $t = \frac{1}{2}$, omdat $wal_1(t)$ oneven t.o.v. $t = \frac{1}{2}$

We kunnen de integraal in (7) als een operator opvatten:

$$f_k = J.wal_k \quad (11)$$

en zoeken de daarmee korresponderende operator in de Walsh-Fourier getransformeerde ruimte. De W F T van f_0 bepalen we met behulp van de zaagtand functie $Z_n(t)$, die periodiek is met periode $1/n$:

$$Z_n(t) = t - \frac{1}{2n}, \quad n=2^m, \quad 0 < t < \frac{1}{n} \quad (12)$$

$$Z_n(t + \frac{1}{n}) = Z_n(t)$$

dan is $Z_1(t) = t - \frac{1}{2} wal_0(t)$ voor $0 < t < 1$

$$Z_2 = Z_1 + \frac{1}{4} wal_1$$

$$Z_4 = Z_2 + \frac{1}{8} wal_3$$

⋮

⋮

$$Z_n = Z_{n/2} + \frac{1}{2n} wal_{n-1}$$

waaruit volgt:

$$f_0(t) = t = \frac{1}{2} - \sum_{j=1}^{\infty} 2^{-j-1} wal_{2^j-1}$$

Wanneer we ons beperken tot V^n , dan is de beste benadering in deze ruimte (zie paragraaf 4)

$$f_0(t) \sim \frac{1}{2} - \sum_{j=1}^m 2^{-j-1} \text{wal}_{2^j-1}, \quad 2^m = n \quad (13)$$

Hiermee is de W F T van $f_0(t)$ in V^n gevonden.

Om de formules wat eenvoudiger te houden, kiezen we $m = 3$, de bewijsvoering voor een andere m kan hieruit eenvoudig afgeleid worden. (13) wordt dan:

$$f_0(t) \sim \frac{1}{2} - \frac{1}{4} \text{wal}_1 - \frac{1}{8} \text{wal}_3 - \frac{1}{16} \text{wal}_7 \quad (13a)$$

Uit (10) volgt door integreren (zie ook (7))

$$f_{2^j+p} = \frac{1}{2} f_j, \quad 0 < t < \frac{1}{2}, \quad p = 0 \text{ of } 1 \quad (14)$$

Met (13) en (14) kunnen we de matrix P_n konstrueren, waarvoor geldt, als $x(t) = \sum a_k \text{wal}_k$

$$\int x(t) dt = P_n \vec{a}$$

P_n is dus de gezochte operator in de getransformeerde ruimte, die korrespondeert met integreren in de oorspronkelijke ruimte.

Voor $n=8$ ziet P_n er als volgt uit:

$$P_8 = \begin{pmatrix} \frac{1}{2} & -\frac{1}{4} & 0 & -\frac{1}{8} & 0 & 0 & 0 & -\frac{1}{16} \\ \frac{1}{4} & 0 & -\frac{1}{8} & 0 & 0 & 0 & -\frac{1}{16} & 0 \\ 0 & \frac{1}{8} & 0 & 0 & 0 & -\frac{1}{16} & 0 & 0 \\ \frac{1}{8} & 0 & 0 & 0 & -\frac{1}{16} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (15)$$

Elke rij van P_8 is gelijk aan de W F T van de overeenkomstige rij in tabel 1, met (13a) kunnen we de eerste rij van P_8 meteen invullen, f_1 , (de tweede rij uit tabel 1) is even t.o.v. $t = \frac{1}{2}$, de W F T van f , bevat dus alleen coëfficiënten met even index, wegens (14) zijn deze gelijk aan de 1e vier coëfficiënten van de eerste rij, vermenigvuldigd met $\frac{1}{2}$.

f_2 is even t.o.v. $t = \frac{1}{4}$ en oneven t.o.v. $t = \frac{1}{2}$, de W F T van f_2 bevat dus alleen coëfficiënten die (lineair) op 01 eindigen, die we vinden door tweemaal (14) toe te passen enz.

Chen c.s. [4] leiden langs andere weg een matrix af, die op P_n lijkt maar er niet aan gelijk is. Het verschil wordt veroorzaakt doordat zij uitgaan van een andere definitie van de Walsh-funkties (zie paragraaf 3.3.).

In paragraaf 7 en paragraaf 8 zullen toepassingen van de operator P_n gegeven worden.

7. Toepassing op differentiaalvergelijkingen

Sommige differentiaalvergelijkingen (b.v. de lineaire met konstante coëfficiënten) kan men oplossen door eerst de Fourier-transform van de vergelijking te bepalen, de getransformeerde vergelijking op te lossen en op de oplossing de inverse Fourier-transformatie toe te passen. Hetzelfde kan men doen met de W F T. We zullen dit aan enkele voorbeelden demonstreren, die zo gekozen zijn dat de exacte oplossing bekend is en dus met de verkregen resultaten vergeleken kan worden. Om de verschillen goed uit te laten komen verdelen we $\text{vak}(0,1)$ in slechts 4 subintervallen; nauwkeuriger resultaten kan men verkrijgen bij een fijnere indeling.

Als eerste voorbeeld nemen we

$$\frac{dx}{dt} + x = 0 ; x(0) = 1 \quad (1)$$

Van de afgeleide kunnen we de inverse van de integraaloperator gebruiken (zie paragraaf 6.2.), maar de matrix inversie wordt vermeden door eerst te integreren:

$$x(t) + \int_0^t x(z)dz = C = C \text{ wal}_0(t) \quad (2)$$

waarin $C = 1$ wegens $x(0) = 1$

De W F T van 2 geeft dan, voor $n=4$:

$$\vec{a} + \vec{a} P_4 = (1, 0, 0, 0) \quad (3)$$

waarin \vec{a} de W F T van $x(t)$ is, dus

$$x(t) = \sum_{j=0}^3 a_j \text{ wal}_j(t) \text{ en } \vec{a} = (a_0, a_1, a_2, a_3)$$

(3) wordt uitgeschreven:

$$a_0 + \left(\frac{1}{2} a_0 + \frac{1}{4} a_1 + \frac{1}{8} a_3\right) = 1$$

$$a_1 + \left(-\frac{1}{4} a_0 + \frac{1}{8} a_2\right) = 0$$

$$a_2 - \frac{1}{8} a_1 = 0$$

$$a_3 - \frac{1}{8} a_0 = 0$$

Dit zijn 4 lineaire vergelijkingen in a_0 , a_1 , a_2 en a_3 .

De oplossing is:

$$a_0 = 0,63405$$

$$a_1 = 0,15607$$

$$a_2 = 0,01951$$

$$a_3 = 0,07926$$

en hieruit volgt $x(t)$ door toepassen van de inverse W F T:

t	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{5}{8}$	$\frac{7}{8}$	
$x(t)$	0.8889	0.6914	0.5378	0.4182	(benaderde oplossing)
e^{-t}	0.8825	0.6873	0.5353	0.4169	(exacte oplossing)

(5)

Gezien de grove benadering (slechts 4 subintervallen) mag de overeenstemming tussen exacte en benaderde oplossing bevredigend genoemd worden.

Als tweede voorbeeld nemen we

$$x''(t) + x(t) = 0 ; x(0) = 1, x'(0) = 0 \tag{6}$$

tweemaal integreren geeft:

$$x(t) + \int_0^t \int_0^u x(v) \, dv \, du = c_0 + c_1 t = 1 \tag{7}$$

$c_0 = 1$ en $c_1 = 0$ wegens beginvoorwaarden in (6).

De W F T van (7) luidt:

$$\vec{a} + \vec{a} P_4^2 = (1, 0, 0, 0) \quad (8)$$

dus

$$64 \vec{a} = 64 \vec{a} P_4^2 = (64, 0, 0, 0) \quad (9)$$

Uitgeschreven

$$75a_0 + 8a_1 + 2a_2 + 4a_3 = 64$$

$$-8a_0 + 59a_1 - 2a_3 = 0$$

$$2a_0 + 63a_2 = 0$$

$$-4a_0 - 2a_1 + 63a_3 = 0$$

waaruit:

$$a_0 = + 0.8386$$

$$a_1 = + 0.1156$$

$$a_2 = - 0.0266$$

$$a_3 = + 0.0569$$

waaruit

t	1/8	3/8	5/8	7/8
x(t)	0.9845	0.9339	0.8065	0.6393
exact: cos(t)	0.9922	0.9305	0.8110	0.6410

Ook hier is de overeenkomst bevredigend. We merken nog op dat in beide voorbeelden de fout niet toeneemt met toenemende t, wat bij zogenaamde "stap" methoden vaak wel het geval is. Uiteraard is hiermee nog niet aangetoond dat deze methode voor het oplossen van differentiaalvergelijkingen beter zou zijn dan de vele daarvoor bestaande methoden. Zoals in het begin van deze paragraaf is vermeld, was het de bedoeling de bruikbaarheid van de operator P aan te tonen.

8. Toepassing in de variatie-rekening

In de variatie-rekening kent men problemen van de volgende vorm: gevraagd een functie $x(t)$ die de integraal

$$J = \int_0^1 F \{t, x(t)\}, \dot{x}(t), dt \quad (1)$$

extremaal maakt. Hierin is F een gegeven functie, een stip betekent differentieren naar t . In leerboeken over variatie-rekening wordt aangetoond dat $x(t)$ een oplossing is van de vergelijking van Euler

$$\frac{\partial F}{\partial x} - \frac{d}{dt} \frac{\partial F}{\partial \dot{x}} = 0 \quad (2)$$

maar deze vergelijking blijkt in de praktijk zelden langs analytische weg opgelost te kunnen worden, zodat men zijn toevlucht tot andere methoden (numeriek, Fourier etc.) moet nemen.

Chen en Hsiao [4] laten zien, dat men het probleem, uitgaande van (1), rechtstreeks kan oplossen met behulp van Walsh-funkties. Daartoe benaderen zij $\dot{x}(t)$ met een Walsh-Fourier reeks (paragraaf 4)

$$\dot{x}(t) = c_0 \text{wal}_1(t) + c_1 \text{wal}_1(t) + \dots + c_{n-1} \text{wal}_{n-1}(t) \quad (3)$$

Wanneer het lukt om de vooralsnog onbekende coëfficiënten $c_0 \dots c_{n-1}$ te berekenen, dan kan men met behulp van de operator P_n (paragraaf 6.2.) $x(t)$ berekenen en is de oplossing gevonden. Wij zullen dit demonstreren aan een eenvoudig voorbeeld, ontleend aan [4] en aangepast aan de definitie van paragraaf 3.5. (in [4] wordt een afwijkende definitie, zie paragraaf 3.3. gebruikt). Om schrijfwerk te beperken nemen we $n=4$, nauwkeuriger resultaten worden uiteraard bereikt met grotere n .

Voorbeeld: Bepaal de extrema van de integraal

$$J = \int_0^1 (\dot{x}^2 + 4tx) dt \quad (4)$$

met randvoorwaarden

$$x(0) = 0, x(1) = 1 \quad (5)$$

We stellen dus (zie (3))

$$\dot{x}(t) = c_0 \text{wal}_0 + c_1 \text{wal}_1 + c_2 \text{wal}_2 + c_3 \text{wal}_3 \quad (6)$$

Uit paragraaf 6.2. volgt

$$t = \frac{1}{2} \text{wal}_0 - \frac{1}{4} \text{wal}_1 - \frac{1}{8} \text{wal}_3 \quad (7)$$

en daarmee

$$\begin{aligned} J &= \int_0^1 \dot{x}^2 dt + 4 \int_0^1 tx \dot{x} dt \\ &= \int_0^1 (c_0 \text{wal}_0 + c_1 \text{wal}_1 + c_2 \text{wal}_2 + c_3 \text{wal}_3)^2 dt \\ &\quad + \int_0^1 (2 \text{wal}_0 - \text{wal}_1 - \frac{1}{2} \text{wal}_3)(c_0 \text{wal}_0 + c_1 \text{wal}_1 + c_2 \text{wal}_2 + c_3 \text{wal}_3) dt \end{aligned} \quad (8)$$

Wegens de orthonormaliteit van de Walsh-funkties kunnen we (8) vereenvoudigen tot

$$J = c_0^2 + c_1^2 + c_2^2 + c_3^2 + 2c_0 - c_1 - \frac{1}{2}c_3 \quad (9)$$

Tot zover is nog geen gebruik gemaakt van de randvoorwaarden. Uit (5) en (6) volgt

$$x(t) = c_0 \int_0^t \text{wal}_0(z) dz + c_1 \int_0^1 \text{wal}_1(z) dz + c_2 \int_0^1 \text{wal}_2(z) dz + \int_0^1 \text{wal}_3(z) dz \quad (10)$$

$x(1) = 1$ levert

$$1 = c_0 + 0 + 0 + 0, \text{ dus } c_0 = 1 \quad (11)$$

Ingevuld in (9)

$$J = 3 + c_1^2 + c_2^2 + c_3^2 - c_1 - \frac{1}{2} c_3 \quad (12)$$

en deze uitdrukking is extremaal als

$$\frac{\partial J}{\partial c_i} = 0 \quad \text{voor } i = 1, 2, 3$$

wat levert

$$\begin{aligned} 2c_1 - 1 &= 0 & , c_1 &= \frac{1}{2} \\ 2c_2 &= 0 & , c_2 &= 0 \\ 2c_3 - \frac{1}{2} &= 0 & , c_3 &= \frac{1}{4} \end{aligned} \quad (13)$$

en dus

$$\dot{x}(t) = \text{wal}_0 + \frac{1}{2} \text{wal}_1 + \frac{1}{4} \text{wal}_3 \quad (14)$$

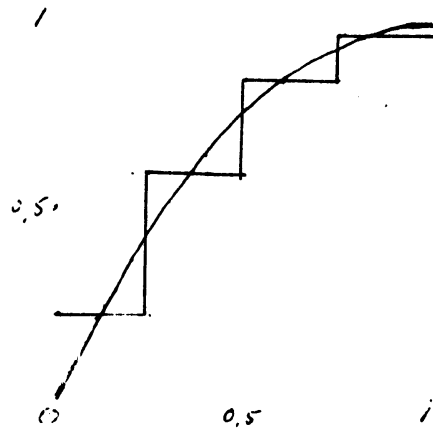
Met behulp van de operator P_4 vinden we de W F T coëfficiënten van $x(t)$

$$(1, \frac{1}{2}, 0, \frac{1}{4}) \begin{vmatrix} \frac{1}{2} & -\frac{1}{4} & 0 & -\frac{1}{8} \\ \frac{1}{4} & 0 & -\frac{1}{8} & 0 \\ 0 & \frac{1}{8} & 0 & 0 \\ \frac{1}{8} & 0 & 0 & 0 \end{vmatrix} = (\frac{21}{32}, -\frac{1}{4}, -\frac{1}{16}, -\frac{1}{8}) \quad (15)$$

en dus

$$x(t) = \frac{1}{32} (21 \text{ wal}_0 - 8 \text{ wal}_1 - 2 \text{ wal}_2 - 4 \text{ wal}_3) \quad (16)$$

Een grafische voorstelling hiervan, tezamen met de exacte oplossing, is gegeven in figuur 9



Figuur 9: Benaderde oplossing $x(t)$ (traplijn) en exacte oplossing $2t-t^2$.

9. Sequentietransformatie en filteren

9.1. Sequentietransformatie

In de praktijk komt het nogal eens voor dat men een signaal wil overbrengen, maar dat de frequenties die dat signaal bevat niet of slecht passen bij het kanaal waarover men het wil verzenden. Een oplossing is het signaal te moduleren op een draaggolf, die wel in het gewenste frequentiegebied ligt. Dit berust voor amplitude modulatie op de formule

$$2 \cos(2\pi ft) \cdot \cos(2\pi gt) = \cos 2\pi(g+f)t + \cos 2\pi(g-f)t \quad (1)$$

waarin $g > f$ de frequentie van de draaggolf is, en f de frequentie van een component van het signaal. Wanneer in het signaal de frequenties $0 < f < f_m$ voorkomen, dan bevat het gemoduleerde signaal de frequenties tussen $g - f_m$ en $g + f_m$, waarin g - binnen zekere praktische grenzen-vrijgekozen kan worden. Men noemt dit ook wel frequentietransformatie.

Hetzelfde kan men doen voor sequenties met behulp van de formule (10), paragraaf 3.4. Een verschil valt hierbij meteen op: Een sequentie geeft slechts één getransformeerde component, frequentie transformatie geeft er twee (zie (11)).

Sequentietransformatie is daardoor eenvoudiger dan frequentietransformatie. Zo kan men aan de ontvangstzijde het oorspronkelijke signaal eenvoudig terug krijgen door nogmaals met de draaggolf te moduleren volgens het schema:

signaal: wal_h

$$\text{gemoduleerd: } wal_h \times wal_j = wal_{h \oplus j} \quad (2)$$

$$\text{gedemoduleerd: } wal_{h \oplus j} \times wal_j = wal_{h \oplus j \oplus j} = wal_h \quad (3)$$

Bij frequentietransformatie gaat dat niet zo eenvoudig. Het onderscheid wordt uiteraard nog groter als een aantal transformaties uitgevoerd moet worden, b.v. bij een transport via een aantal achtereenvolgende kanalen.

We merken nog op dat sequenties zowel omhoog (2) als omlaag (3) getransformeerd kunnen worden.

9.2. Filteren

Oorspronkelijk was een filter een apparaat, dat sommige frequenties van een signaal onderdrukte, zoals tot uitdrukking komt in namen als laag-doorlaat filter, bandfilter e.d. Deze begrippen kent men ook voor sequentiefilters. Een laag-doorlaat sequentiefilter laat zich eenvoudig konstrueren. Men middelt het signaal over intervallen ter lengte van 2^{-m} , dan zijn alle sequenties $\geq 2^m$ geëlimineerd, de lagere blijven onveranderd. Voor een elektrisch signaal kan dit met een eenvoudige schakeling bereikt worden: integreer het signaal over het subinterval en deel door de intervallengte. We merken op dat zo'n filter een scherpe afsnij sequentie heeft: alle sequenties $> 2^m$ gaan er onveranderd door, sequenties $\geq 2^m$ worden onderdrukt.

Met behulp van sequentie transformatie (paragraaf 9.1.) laten zich sequentie-bandfilters konstrueren. Stel, dat we van een signaal alleen de sequenties van $3n/4$ tot $n=2^m$ willen doorlaten. Met een laag-doorlaat-filter verwijderen we eerst de sequenties $\geq 2^m$. Daarna passen we sequentietransformatie toe (2) en wel zodanig dat de sequenties van

$3n/4$ tot n worden getransformeerd tot sequenties van 0 tot $n/4$. De overige sequenties zijn dan groter, en kunnen door een tweede laag-doorlaatfilter onderdrukt worden. Door terugtransformatie (3) krijgen de overblijvende sequenties weer hun oorspronkelijke waarde, terwijl de sequenties $< 3n/4$ of $\geq n$ onderdrukt zijn.

Voorgaande filters zijn beperkt, omdat hun bovengrenzen bij machten van 2 gelegen zijn. Door combinatie van zulke filters kan men meer mogelijkheden realiseren.

Meer algemene filters kan men maken met behulp van een digitale rekenmachine. Men past dan eerst de W F T toe, maakt de coëfficiënten nul door de sequenties die men uit wil filteren en past dan de inverse W F T toe. Bovenstaande methode komt er op neer, dat men de Walsh-Fouriercoëfficiënten vermenigvuldigd met een factor, die 0, of 1 is. Het gebruik van een rekenmachine laat echter ook andere factoren toe, men kan dan b.v. bepaalde sequenties versterken en andere verzwakken. Hiervan wordt o.a. gebruik gemaakt voor het bewerken van digitale beelden, b.v. onderdrukken van een stoor-sig-naal, wazige grenzen weer scherp maken etc.

10. Faseverschuiving en balfuncties

In paragraaf 9.2. is vermeld, dat de lage sequenties niet vervormd worden als ze een laag-doorlaat sequentiefilter passeren. Wel treedt er dan een verschuiving in de tijd op: een waarde aan de uitgang is pas bekend als de invoer over een basisinterval geïntegreerd is, dus één tijdstapje later. In formule: als de lengte van de basisintervallen Δt is, met

$$\Delta t = 1/n, \quad n = 2^m \quad (1)$$

en de invoer is $wal_k(t)$ met $k < n$, dan is de uitvoer $wal_k(t - \Delta t)$. Men noemt dit faseverschuiving, naar aanleiding van een soortgelijk verschijnsel bij lineaire analoge filters, waar een sinusvormig signaal, $\sin(2\pi ft)$, als invoer een uitvoer $C \sin(2\pi ft - \phi)$ geeft. (2)

Laat men ϕ toenemen van 0 tot $\frac{\pi}{2}$, dan verschuift het maximum van de functie in (2) van $t = f/4$ naar $t = 0$.

Om iets soortgelijks te bereiken voor Walsh-functies voeren Banzal en Mirza [1], naar aanleiding van de formule

$$C \sin(2\pi ft - \phi) = A \sin(2\pi ft) + B \cos(2\pi ft) \quad (3)$$

met $A = C \cos \phi$ en $B = C \sin \phi$

balfuncties in, gedefinieerd als

$$\left[M_k, N_k \right] bal_k(t) = S_k sal_k(t) + C_k cal_k(t) \quad (4)$$

waarin

$$M_k = S_k + C_k, \quad N_k = S_k - C_k \quad (5)$$

Laat men in (4) N_k variëren van $-M_k$ tot $+M_k$, dan kan men hierin een soort verschuiving van het maximum herkennen ¹⁾(zie figuur 10).

1) Door de konstante stukken in $bal_k(t)$ kan men niet zomaar spreken over de plaats van het maximum. Daarom is in de fig. met een driehoekje het zwaartepunt van het positieve deel van de (periodieke) balfunctie aangegeven. Men ziet dit verschuiven van $t = \frac{1}{4}$ naar $t = 0$ als N van $+1$ naar -1 gaat.

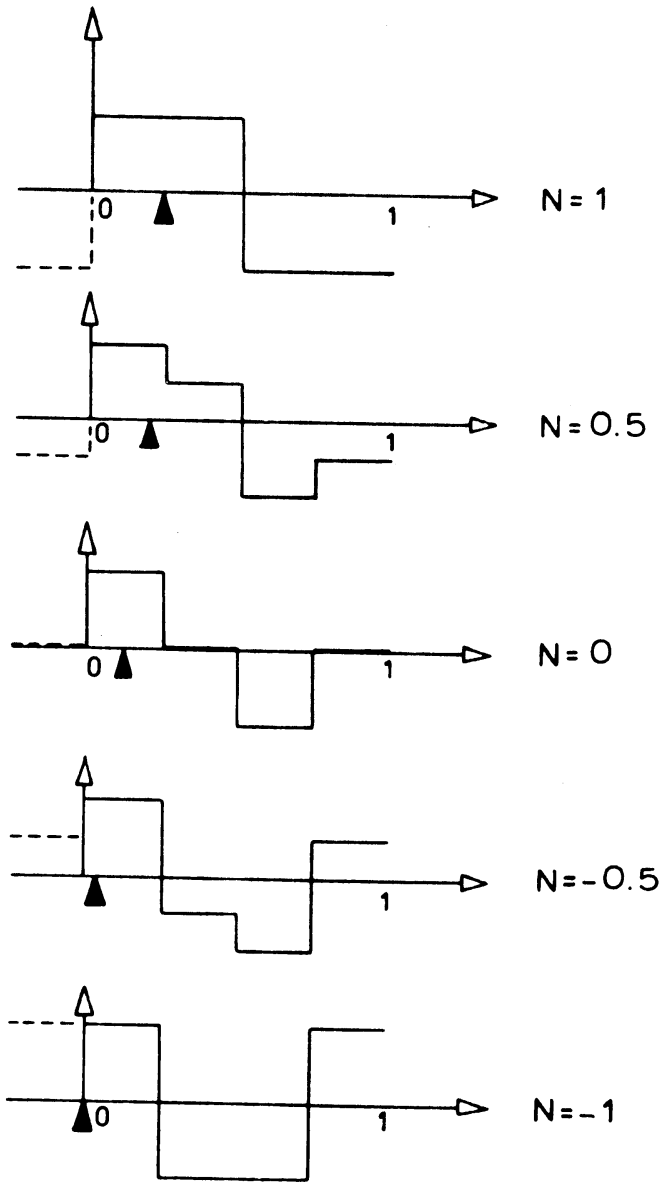


Fig.10 Balfuncties voor $M=k=1$

Banzal en Mirza passen dit met succes toe op niet lineaire digitale filters. (cos en sin funkties worden door zo'n filter sterk vervormd en zijn daarvoor dus minder geschikt). Toch heb ik nog wel enkele bezwaren:

a. De notatie in (4) is nodeloos ingewikkeld en afwijkend van wat in de wiskunde gebruikelijk is. Beter is m.i. $\text{bal}(M_k, N_k, k, t)$ of $\text{bal}_k(M, N, t)$.

b. De bijdrage tot het sequentiespectrum van (4) is

$$S^2 + C^2 = (M^2 + N^2)/2 \quad (6)$$

en dit verandert als we M konstant houden en N laten variëren.

(zoals in figuur 10). Wat we willen is echter, dat (6) konstant blijft bij faseverschuiving, (zoals in (3) $A^2 + B^2$ konstant blijft als ϕ varieert). Dit kunnen we bereiken door de definitie (4) wat te wijzigen:

$$\text{bal}_k(a, t) = a \text{cal}_k(t) + \sqrt{1-a^2} \text{sal}_k(t) \quad (7)$$

waarin a de faseverschuiving bepaalt.

c. De grootheid a in (7) kan elke waarde aannemen en is dus in wezen een analoge grootheid, die daarom wat uit de toon valt te midden van allemaal gedigitaliseerde grootheden.

Harmuth [7] behandelt de faseverschuiving geheel digitaal:

$$\Delta t = j \cdot 2^{-m}, \quad j \text{ geheel} \quad (8)$$

waarin 2^{-m} de lengte van het basisinterval voorstelt (we beperken ons dus tot sequenties $\langle n=2^m \rangle$).

Voor sommige waarde van j geldt dan

$$\text{cal}_k(t-j \cdot 2^{-m}) = \text{sal}_k(t) \quad (9)$$

$$\text{sal}_k(t-j \cdot 2^{-m}) = -\text{cal}_k(t)$$

wat doet denken aan de formule $\cos(t - \frac{\pi}{2}) = \sin(t)$.

De waarden van j , waarvoor (9) geldt hangen op een wat ingewikkelde manier van k af. zie tabel 2.

Tabel 2.

k	k	$\Delta t =$	k	k	$\Delta t =$
dec.	binair	$j \cdot 2^{-m}$	dec.	linair	$j \cdot 2^{-m}$
1	0001	1/4	8	1000	1/32
2	0010	1/8	9	1001	1/4
3	0011	-1/4	10	1010	1/8
4	0100	1/16	11	1011	-1/4
5	0101	1/4	12	1100	-1/16
6	0110	-1/8	13	1101	1/4
7	0111	-1/4	14	1110	-1/8
			15	1111	-1/4

We kunnen Δt afleiden uit de lineaire representatie van k [11].

Als $k_{m-1} k_{m-2} \dots k_2 k_1 k_0$ de bits van k zijn dan is

$$|\Delta t| = 2^{-i-2} \quad \text{als } k_i = 1 \text{ en } k_{i-1} = k_{i-2} = \dots = k_2 = k_1 = k_0 = 0 \quad (10)$$

Door (10) is de absolute waarde van Δt bepaald. Het teken van Δt volgt uit (11)

$$\left. \begin{aligned} \Delta t &= -2^{-i-2} \quad \text{als } k_{i+1} = 1 \\ \Delta t &= 2^{-i-2} \quad \text{als } k_{i+1} = 0 \end{aligned} \right\} \text{ onder voorwaarde (10)} \quad (11)$$

Uit 9 volgt nog

$$\left. \begin{aligned} \text{cal}_k(t - j \cdot 2^{-m+1}) &= -\text{cal}_k(t) \\ \text{sal}_k(t - j \cdot 2^{-m+1}) &= -\text{sal}_k(t) \end{aligned} \right\} \quad (12)$$

en

$$\begin{aligned} \text{cal}_k(t-j \cdot 2^{-m+2}) &= \text{cal}_k(t) \\ \text{sal}_k(t-j \cdot 2^{-m+2}) &= \text{sal}_k(t) \end{aligned} \quad (13)$$

Harmuth past het voorgaande toe bij het overbrengen van een signaal, dat voor het verzenden gemoduleerd wordt op een draaggolf $\text{wal}_k(t)$, met een hogere sequentie dan alle in het signaal voorkomende sequenties. Aan de ontvangstzijde kan het signaal dan gedemoduleerd worden met behulp van een aantal bekende elementen zoals sequentie laag-doorlaatfilters, generatoren van en vermenigvuldigen voor Walsh-funkties etc. Ook correlatiefunkties

$\int \text{wal}_k(t) \text{wal}_k(t-j \cdot \Delta t)$ kunnen hiermee behandeld worden.

Uit (8) en (10) volgt $|j| = 2^{m-i-2}$, waarin i een functie is van k (zie (10)). Verschillende waarden van k leveren niet dezelfde waarden van j (zie tabel 2), wat bezwaarlijk kan zijn als men een aantal Walsh-funkties wil combineren. In zo'n geval kan men de W F T toepassen. Voor elke k is namelijk $\text{wal}_k(t-2^{-m})$ kwadratisch integreerbaar, dus behoort tot V^n (met $n=2^m$) en heeft dus een Walsh Fourier getransformeerde

$$\text{wal}_k(t-2^{-m}) = \sum_{i=0}^{n-1} a_{i,k} \text{wal}_i(t)$$

De coëfficiënten $a_{i,k}$ kan men berekenen met de W F T voor elke k . Men krijgt dan een matrix A met elementen $a_{i,k}$. Voeren we nog een vector $\vec{\phi}(t)$ in, waarvan de componenten Walsh-funkties zijn:

$$\vec{\phi}(t) = (\text{wal}_0(t), \text{wal}_1(t) \dots \text{wal}_{n-1}(t)),$$

dan kunnen we de faseverschuiving over één elementaire stap kort weergeven

Geraadpleegde litteratuur

- (1) V.S. Bansal and K.B. Mirza. On a class of describing functions for dyadic systems. Journal of Franklin Institute Vol. 306, p 275-282, 1978.
- (2) K.G. Beauchamp. Walsh-funktions and their applications. Academic Press, London 1975.
- (3) G.A. Campbel and R.M. Foster. Fourier integrals for practical applications, New York, 1948.
- (4) C.F. Chen and C.H. Hsiao. A Walsh series direct method for solving variational problems. Journal of Franklin Institute Vol 300, p 265-280, 1975.
- (5) C.F. Chen and Y.T. Tsay. Walsh operational matrices for fractional calculus and their application to distributed systems. Journal of Franklin Institute. Vol 303, p 267-284, 1977.
- (6) L. Collatz. Funktional analysis and Numerisch Mathematik. Springer Verlag 1964.
- (7) H.F. Harmuth. Transmission of information by orthogonal functions. Springer Verlag 1970.
- (8) W. Kuik. Inleiding in enkele methoden voor het schatten van spectra. Verslag V-335, 1980.
- (9) W.E. Milne. Numerical Calculus. Princeton University Press, 1949.
- (10) K.B. Mirza and V.S. Bansal. Logical derivative and integral operators for sequency filtering. Journal of Franklin Institute. Vol. 306, p 267-274, 1978.

- (11) F. Pichler. Das System der sal- und cal-Funktionen als Erweiterung des Systems der Walsh-Funktionen und die Theorie der sal- und cal-Fouriertransformation. Thesis, Dep. of Mathematics, Innsbruck University, 1967.
- (12) F. Riesz und B.Sc. Nagy. Vorlesungen über Funktional analysis, Berlin 1956.
- (13) E.H.J. Vermaas. Fast Fourier transformatie. Wetenschappelijk Rapport W.R. 74-1.
- (14) J.L. Walsh. A closed set of normal orthogonal funkties. Amer. J. of Mathematics. Vol. 45, p 5-24, 1923.