7 DIE DISKRETE FOURIER TRANSFORM

7.1 Die ortogonaliteit van diskrete Fourier modusse

Die rekenaar kan nie kontinue funksies hanteer nie en ook nie oneindige reekse nie. Om 'n funksie rekenaarmatig te kan hanteer, moet dit of voorgestel word met 'n *eindige* reeks of deur slegs die koördinate van 'n eindige aantal punte op die funksie te gee, d.w.s. deur die funksie te *diskretiseer*. In hierdie hoofstuk gaan ons net van gelykverspreide punte gebruik maak.

Beskou die kontinue veranderlike $x \in [0, 2L]$ en kontinue funksie $f(x)$ gedefinieer op hierdie interval. Deel die interval $[0, 2L]$ op in N gelyke intervalle, elk met wydte $h = 2L/N$, en bereken die funksie-waardes net by hierdie punte:

$$x_j := hj = \frac{2Lj}{N} \quad f_j := f(x_j) \quad j = 0, 1, ..., N-1.$$

Sulke diskrete funksies kan geskryf word as 'n lineêre kombinasie van 'n stel diskrete basisfunksies, $\{\phi_{0j}, \phi_{1j}, ...\}$:

$$f_j = \sum_{k=0}^{N-1} c_k \phi_{jk}, \qquad j = 0, 1, ..., N_1 \tag{.1}$$

waar $\{c_j, j = 0, 1, ..., N-1\}$ die koëffisiënte van die kombinasie is.

Let op dat ons vir die basisfunksie ϕ_{kj}, die eerste indeks (d.w.s. k) beskou as die nommer van die funksie en die tweede indeks (d.w.s. j) as die argument van die funksie.

Ons kan (.1) ook interpreteer as 'n matriks vermenigvuldig met 'n vektor:

$$\mathbf{f} = \Phi \mathbf{c}$$

waar die basisfunksies die kolomme van Φ uitmaak.

Om die koëffisiënte-vektor $\mathbf{c}$, te vind, moet ons slegs met die inverse matriks Φ^{-1} van links vermenigvuldig:

$$\mathbf{c} = \Phi^{-1}\mathbf{f}$$

As Φ agter 'n ortogonale matriks is, is die oplossing aansienlik makliker:

$$\mathbf{c} = \Phi^T \mathbf{f}$$

en as Φ 'n komplekse ortogonale matriks (d.w.s. 'n unitêre matriks) is, dan is

$$\mathbf{c} = \Phi^H \mathbf{f}$$

waar die H-boskrif die hermitiese toegevoegde aandui.

Ons gaan nou aantoon dat die diskrete Fourier modusse

$$\phi_{kj} = e^{-2\pi i kj/N}, \qquad k = 0, 1, ..., N-1, j = 0, 1, ..., N-1$$

'n spesiale ortogonaliteitseienskap bevredig.

$$\begin{aligned}
\sum_{j=0}^{N-1} \phi_{kj} \overline{\phi_{mj}} &= \sum_{j=0}^{N-1} e^{-2\pi i kj/N} \overline{e^{-2\pi i mj/N}} \\
&= \sum_{j=0}^{N-1} e^{2\pi i(m-k)j/N} \\
&= \sum_{j=0}^{N-1} r^j \\
& \quad \text{waar } r = e^{2\pi i(m-k)/N} \\
&= \begin{cases} \dfrac{1-r^N}{1-r} & \text{as } r \neq 1 \\ N & \text{as } r = 1 \end{cases}
\end{aligned}$$

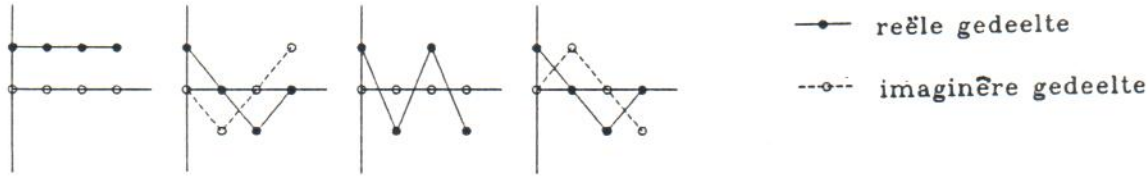
Maar $r^N = e^{2\pi i(m-k)} = 1$ en dus is

$$\sum_{j=0}^{N-1} \phi_{kj} \overline{\phi_{mj}} = \begin{cases} 0 & \text{as } r \neq 1, \text{ dus as } k \neq m \\ N & \text{as } r = 1, \text{ dus as } k = m \end{cases} \tag{.2}$$

<u>Voorbeeld:</u> Met $N = 4$ lyk die matriks so:

$$\Phi = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

Die kolomme van die matriks is 'n stel diskrete ortogonale funksies. Dit is eintlik steeds die sinus en cosinus basisfunksies van gewone Fourier-reekse, maar elk is slegs by vier punte bereken. Die figuur hieronder toon die vier diskrete funksies.



FIGUUR 7.1

7.2 Die Diskrete Fourier Transform

Die Diskrete Fourier Transform (DFT) kan beskou word as 'n analoog van die kontinue Fourier Transform waar die funksies diskreet is en die integrasies vervang is met sommasies. Andersinds kan die DFT ook beskou word as 'n Fourier-reeks met slegs N terme en waar die Fourier koëffisiënte bereken word met die reghoekreël vir integrasie.

Veronderstel ons wil die diskrete funksie $f_j, j = 0, 1, ..., N-1$ uitbrei as 'n lineêre kombinasie van N diskrete Fourier modusse

$$f_j = \sum_{k=0}^{N-1} c_k e^{2\pi i kj/N}.$$

Om die koëffisiënte c_k, te vind, volg ons die volgende stappe:

(1) Vermenigvuldig met $e^{-2\pi i jm/N}$.

(2) Sommeer oor j.

(3) Ruil die volgorde van sommasie aan die regterkant om en neem die koëffisiënte c_k uit onder die sommasie oor j.

(4) Maak gebruik van die ortogonaliteitseienskap van die diskrete Fourier modusse en vereenvoudig.

Die resultaat is dan

$$\begin{aligned} \sum_{j=0}^{N-1} f_j e^{-2\pi i jm/N} &= \sum_{k=0}^{N-1} c_k \sum_{j=0}^{N-1} e^{2\pi i (k-m)j/N} \\ &= N c_m \end{aligned}$$

Ons herdefinieer nou die koëffisiënte $\tilde{f}_m := N c_m$ en noem dit die Diskrete Fourier Transform (DFT) van f_j. Die reeks vir f_j, noem ons die Diskrete Inverse Fourier Transform (DIFT). Dus

$$\tilde{f}_m = \sum_{j=0}^{N-1} f_j e^{-2\pi i jm/N} \qquad ....DFT$$

$$f_j = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{f}_m e^{+2\pi i mj/N} \qquad ....DIFT$$

In matriksnotasie kan dit as volg geskryf word:

$$\tilde{\mathbf{f}} = \Phi \mathbf{f} \qquad ....DFT$$

$$\mathbf{f} = \Phi^{-1} \tilde{\mathbf{f}} \qquad ....DIFT$$

waar

$$\Phi = \begin{bmatrix} w^0 & w^0 & w^0 & w^0 & \cdots & w^0 \\ w^0 & w^1 & w^2 & w^3 & \cdots & w^{N-1} \\ w^0 & w^2 & w^4 & w^6 & \cdots & w^{2(N-1)} \\ w^0 & w^3 & w^6 & w^9 & \cdots & w^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ w^0 & w^{N-1} & w^{2(N-1)} & w^{3(N-1)} & \cdots & w^{(N-1)(N-1)} \end{bmatrix}$$

waar $w = e^{-2\pi i/N}$ en $\Phi^{-1} = \overline{\Phi}/N$.

Voorbeeld: Verkry die DFT van $f_j = \cos(\pi j/2)$, met $N = 4$.

$$\mathbf{f} = \begin{bmatrix} \cos(0) \\ \cos(0.5\pi) \\ \cos(\pi) \\ \cos(1.5\pi) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$$

en dan is

$$\tilde{\mathbf{f}} = \Phi\mathbf{f} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 2 \end{bmatrix}$$

7.3 Fourier-interpolasie en vergladding

Fourier-interpolasie:

'n Interpolaat $p(x)$, deur N punte $\{(x_j, y_j),\ j = 0, 1, ..., N-1\}$ is 'n gladde kromme wat deur hierdie koördinate gaan. Dit bevredig dus die volgende voorwaarde

$$y_j = p(x_j), \qquad j = 0, 1, ..., N-1 \tag{.3}$$

Beskou 'n stel gelykverspreide punte $\{x_j = hj,\ j = 0, 1, ..., N-1\}$ in die interval $[0, 2L]$ met $h = 2L/N$ die staplengte. Dit is maklik om met behulp van die DFT 'n Fourier-reeks te vind wat deur die gegewe stel koördinate gaan. Dit werk as volg:

Bereken die DFT van y_j:

$$\tilde{y}_m = \sum_{j=0}^{N-1} y_j e^{-2\pi i jm/N}$$

Verkry dan weer die DIFT van $\tilde{y}_m$, en noem dit $p(x_j)$:

$$p(x_j) = \frac{1}{N}\sum_{m=0}^{N-1} \tilde{y}_m e^{2\pi i j m/N} \tag{.4}$$

Let op dat

$$\frac{1}{N} = \frac{h}{2L}$$

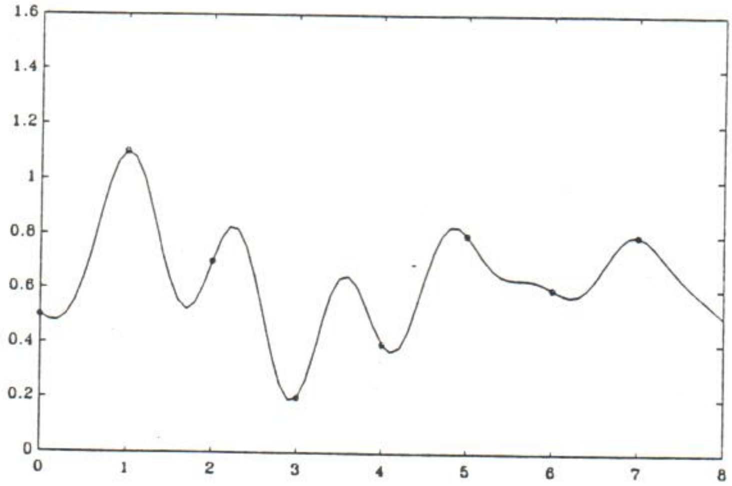
Vervang nou $1/N$ in (.4):

$$p(hj) = \frac{1}{N}\sum_{m=0}^{N-1} \tilde{y}_m e^{2\pi i m j h/(2L)}$$

Beskou nou $x = hj$ as 'n kontinue veranderlike, en $p(x) = p(hj)$ as 'n kontinue funksie:

$$p(x) = \frac{1}{N}\sum_{m=0}^{N-1} \tilde{y}_m e^{\pi i m x/L}$$

Die funksie $p(x)$ is eindige Fourier-reeks wat die voorwaarde (.3) bevredig en stel dus 'n gladde kromme voor wat deur die gegewe stel punte gaan. Figuur 7.2 toon 'n voorbeeld van 'n stel punte met die Fourier-interpolaat daardeur.



FIGUUR 7.2

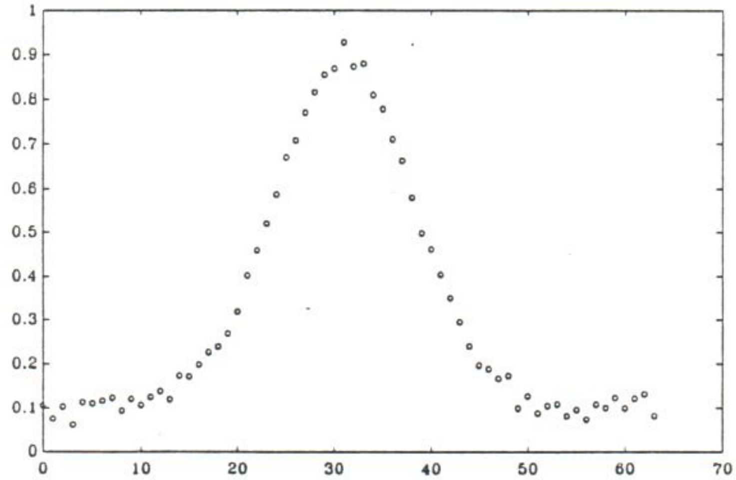
Vergladding:

Indien die interpolaat deur 'n aantal punte baie ossilasies toon is dit soms wenslik om 'n gladde kromme te verkry wat die punte slegs benaderd volg. Hierdie tipe krommes vind veral toepassing waar die punte eksperimentele lesings is en onderworpe aan foute is.

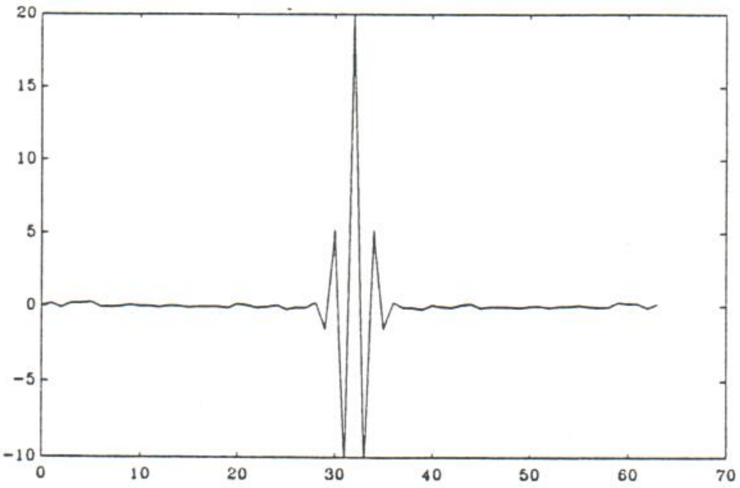
Beskou 'n stel punte $\{(x_j, y_j),\ j = 0, 1, ..., N-1\}$ met die x_j gelykversprei. Die DIFT van die DFT van die diskrete funksie y_j is weer die funksie self. Indien die DFT egter met 'n filtreerfunksie vermeningvuldig word wat die hoër modusse verwyder, voordat die DIFT geneem word, word 'n funksie verkry wat gladder as die oorspronklike is maar wat steeds die strekking van die oorspronklike funksie volg.

Ons beskou slegs een voorbeeld: Figuur 7.3(a) toon 'n stel punte, en Figuur (b) die reële gedeelte van die DFT van die punte. 'n Filtreerfunksie wat lae modusse onveranderd laat maar hoë modusse verwyder word in Figuur (c) getoon. Figuur (d) toon die reële gedeelte van die gefiltreerde DFT en Figuur (e) toon die DIFT saam met die oorspronklike punte.

(a)

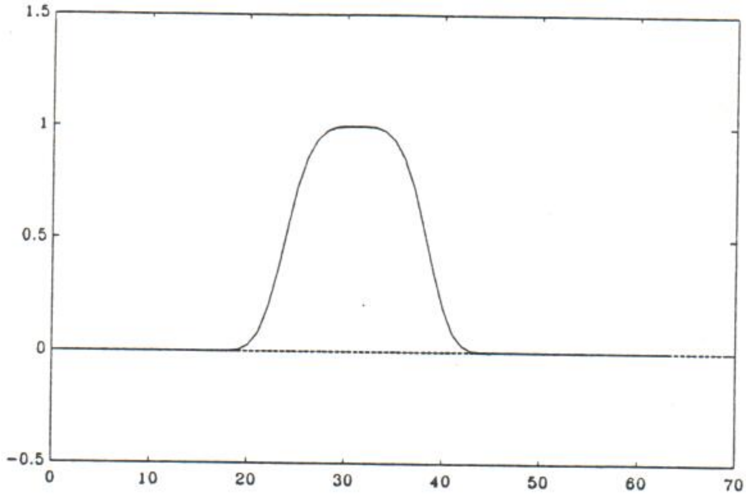


(b)

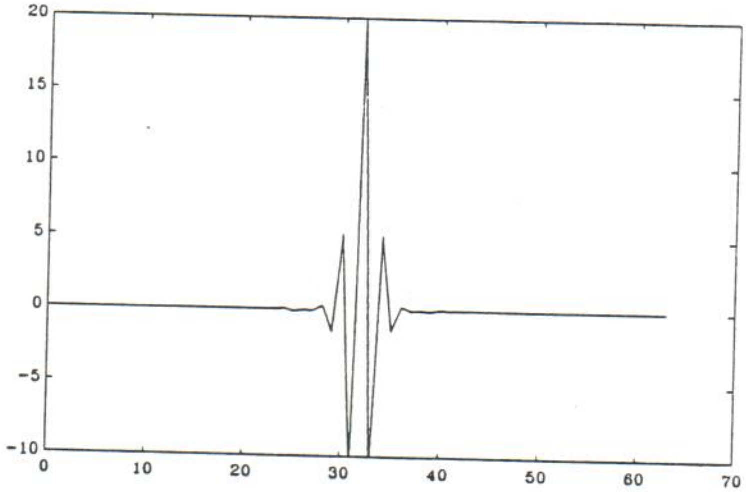


FIGUUR 7.3 (a), (b)

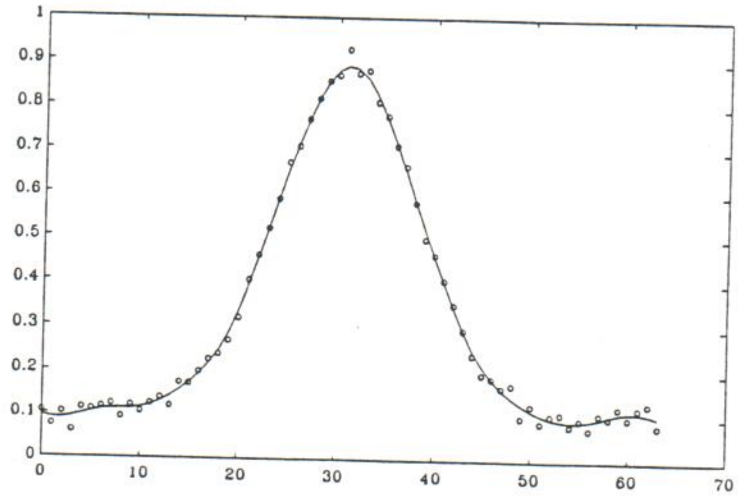
(c)



(d)



(e)



FIGUUR 7.3 (c), (d) en (e)

7.4 Die FFT

Inleiding:

Tot voor 1965 was die numeriese berekening van Diskrete Fourier Transforms glad nie baie gebruik in sein-prosessering en numeriese analise nie. In dié toepassings waar die DFT van nut kon wees, moes die Fourier Transform of die inverse transform van baie groot vektore herhaaldelik verkry word en dit het beteken dat daar elke keer met 'n $N \times N$ matriks veremenigvuldig moes word. Elke matriksvermenigvuldiging het N^2 komplekse vermenigvuldigings vereis, en soveel berekeninge het die proses onuithoubaar stadig laat verloop.

In 1965 egter het ene J.W. Cooley en J.W. Tuckey 'n algoritme gepubliseer wat die DFT bereken in slegs $\frac{N}{2}\log_2 N$ komplekse vermenigvuldigings. Vir groot vektore is die tydsverskil vir die berekening van 'n DFT dus aansienlik. Hierdie algoritme het vinnig baie bekend geword en het groot omwenteling in seinanalise en sommige toepassings van numerise analise teweeg gebring. Dit staan vandag algemeen bekend as die *"Fast Fourier Transform"* of kortweg die FFT.

Alhoewel die FFT ontwerp kan word om die DFT van vektore met enige lengte te bereken, is die bekendste en maklikste geval, díe geval waar die lengte van die vektor 'n mag van twee is. Ons gaan slegs daardie geval in hierdie kursus bespreek.

Laat dus $N = 2^\ell$ waar ℓ 'n heelgetal is.

Die beginsel waarop die FFT werk is dat 'n N-punt transform geskryf kan word as twee $N/2$-punt transforms. Waar 'n N-punt transform N^2 vermenigvuldigings vereis, vereis twee $N/2$-punt transforms $N^2/2$ vermenigvuldigings, dus word die aantal vermenigvuldigings gehalveer. Die aanmekaarlas van die twee $N/2$-punt transforms om die verlangde N-punt transform te vind, neem slegs $N/2$ vermenigvuldigings. Elk van die $N/2$-punt transforms kan egter op dieselfde manier weer gedoen word deur elk op te deel in twee $N/4$-punt transforms, wat weer die aantal vermenigvuldigings wat nodig is halveer.

Een DFT as twee halwes:

Ons bespreek eers een so 'n halveringstap van die proses. Ons gaan die algoritme algemeen herlei vir 'n N-punt transform, maar tussendeur gaan ons sommige stappe illustreer met 'n 8-punt transform.

Laat $w_N = e^{-2\pi i/N}$. Die volgende is dan waar:

$$w_{N/2} = (w_N)^2 \tag{.5}$$

$$(w_{N/2})^{kN/2} = 1, \quad \text{vir } k \text{ heeltallig} \tag{.6}$$

$$(w_N)^{N/2} = -1 \tag{.7}$$

Beskou die N-punt DFT van f_j:

$$\tilde{f}_m = \sum_{j=0}^{N-1} f_j w_N^{jm}, \qquad m = 0, 1, ..., N-1.$$

Herrangskik f_j in twee $N/2$-punt vektore g_k en h_k sodat elke alternatiewe element in 'n ander vektor kom, byvoorbeeld:

$$\begin{array}{cccccccccc}
[& f_0 & f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 &] \\
 & \downarrow & | & \downarrow & | & \downarrow & | & \downarrow & | & \\
[& g_0 & | & g_1 & | & g_2 & | & g_3 & | &] \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
 & [& h_0 & & h_1 & & h_2 & & h_3 &]
\end{array}$$

Ons het dus

$$\begin{aligned} g_k &= f_{2k} \\ h_k &= f_{2k+1} \end{aligned}$$

Verkry nou die DFT van g_k en h_k:

$$\tilde{g}_n = \sum_{k=0}^{N/2-1} g_k (w_{N/2})^{kn}, \qquad n = 0, 1, ..., N/2-1$$

$$\tilde{h}_n = \sum_{k=0}^{N/2-1} h_k (w_{N/2})^{kn}, \qquad n = 0, 1, ..., N/2-1$$

Die DFT van f_j kan as volg opgedeel word:

$$\begin{aligned}
\tilde{f}_m &= \sum_{j=0}^{N-1} f_j w_N^{jm}, \qquad m = 0, 1, ..., N-1 \\
&= \sum_{j=0, j \text{ ewe}}^{N-2} f_j w_N^{jm} \quad + \sum_{j=1, j \text{ onewe}}^{N-1} f_j w_N^{jm} \\
&= \sum_{k=0}^{N/2-1} f_{2k} w_N^{2km} \quad + \sum_{k=0}^{N/2-1} w_N^{2km+m} f_{2k+1} \\
&= \sum_{k=0}^{N/2-1} g_k (w_{N/2})^{mk} \quad + \quad w_N^m \sum_{k=0}^{N/2-1} h_k (w_{N/2})^{mk} ....(*)
\end{aligned}$$

Die N-punt vektor $\tilde{f}_m$ word nou opgedeel in die twee $N/2$-punt vektore. Die eerste een bevat die eerste $N/2$ elemente van $\tilde{f}_m$ en die laaste een die ander elemente, byvoorbeeld:

$$\begin{array}{cccccccccc}
[& \tilde{f}_0 & \tilde{f}_1 & \tilde{f}_2 & \tilde{f}_3 & \tilde{f}_4 & \tilde{f}_5 & \tilde{f}_6 & \tilde{f}_7 &] \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \searrow & \searrow & \searrow & \searrow & \\
[& \tilde{f}_0 & \tilde{f}_1 & \tilde{f}_2 & \tilde{f}_3] & [\tilde{f}_4 & \tilde{f}_5 & \tilde{f}_6 & \tilde{f}_7 &]
\end{array}$$

Ons behandel nou elke vektor apart. Vir $m = 0, 1, ..., N/2-1$ volg uit $(*)$,

$$\tilde{f}_m = \tilde{g}_m + w_N^m \tilde{h}_m, \qquad m = 0, 1, ..., N/2-1$$

Vir die tweede helfte van $\tilde{f}_m$, neem ons $m = n + N/2$ in (*):

$$\tilde{f}_{n+N/2} = \sum_{k=0}^{N/2-1} g_k(w_{N/2})^{k(n+N/2)} + w_N^{n+N/2} \sum_{k=0}^{N/2-1} h_k(w_{N/2})^{k(n+N/2)}$$

Deur gebruik te maak van die eienskappe (.6) en (.7) word die volgende verkry:

$$\begin{aligned} \tilde{f}_{n+N/2} &= \sum_{k=0}^{N/2-1} g_k(w_{N/2})^{kn} - w_N^n \sum_{k=0}^{N/2-1} h_k(w_{N/2})^{kn} \\ &= \tilde{g}_n - w_N^n \tilde{h}_n, \quad n = 0, 1, ..., N/2 - 1 \end{aligned}$$

Ons kan dit as volg opsom:

(**1**) Deel die elemente van f_j op: $g_k = f_{2k}, \quad h_k = f_{2k+1}$.

(**2**) Bereken $\tilde{g}_n$ en $\tilde{h}_n$.

(**3**) Stel $\tilde{f}_m$ as volg saam:

$$\left.\begin{aligned} \tilde{f}_n &= \tilde{g}_n + w_N^n \tilde{h}_n \\ \tilde{f}_{n+N/2} &= \tilde{g}_n - w_N^n \tilde{h}_n \end{aligned}\right\} \quad \text{vir } n = 0, 1, ..., N/2 - 1.$$

Dieselfde stappe word uitgevoer vir die twee $N/2$-punt transforms, en net so vir die daaropvolgende vier $N/4$-punt transforms, ensovoorts.

Operasietelling:

Laat $N = 2^\ell$. In 'n volledige FFT sal daar dus ℓ halveringsfases wees.

As die twee $N/2$-punt transforms bekend is, neem dit slegs $N/2$ vermenigvuldigings om die N-punt transform te bereken, naamlik die $N/2$ keer wat w_N^n met $\tilde{h}_n$ vermenigvuldig word. Die rangskikking van $\tilde{f}_m$ in die twee vektore neem geen ekstra vermenigvuldigings nie.

Byvoorbeeld, die berekening van 'n 2-punt transform neem slegs 1 vermenigvuldiging:

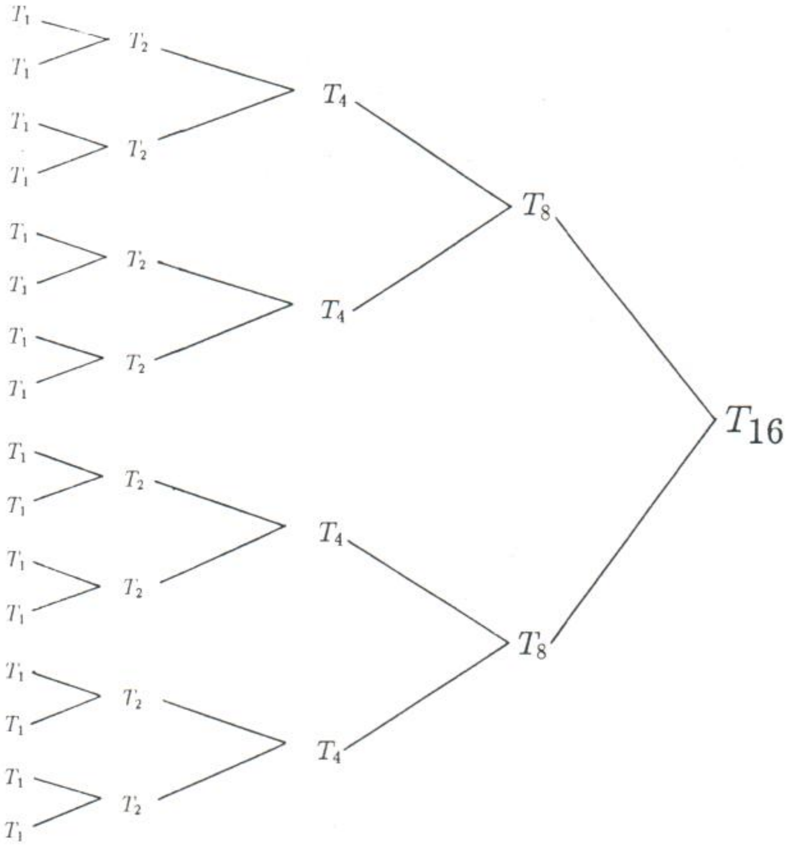
$$\begin{aligned} \tilde{f}_0 &= \tilde{g}_0 + w_N^0 \tilde{h}_0 \\ \tilde{f}_1 &= \tilde{g}_0 - w_N^0 \tilde{h}_0 \end{aligned}$$

waar $\tilde{g}_0 = g_0$ en $\tilde{h}_0 = h_0$. Die enigste vermenigvuldiging is $w_N^0 * \tilde{h}_0$.

Dit is dan duidelik dat die heel eerste fase waar N eenpunt transforms bereken word, geen vermenigvuldigings neem nie, aangesien die eenpunt transform van 'n getal daardie getal self is. Ons noem die die nulste fase. Op die eerste fase word $N/2$ 2-punt transforms bereken, dus $N/2 \times 1 = N/2$ vermenigvuldigings. Op die tweede fase word $N/4$ 4-punt transforms bereken, dus $N/4 \times 2 = N/2$ vermenigvuldigings. Elke fase neem dus $N/2$ vermenigvuldigings en daar is ℓ fases, dus is die totale aantal vermenigvuldigings

$$\ell \times \frac{N}{2} = \frac{N \log_2 N}{2}$$

Byvoorbeeld vir $\ell = 4$ is $N = 16$ en die fases verloop as volg:



waar T_N een N-punt transform voorstel, en die aantal vermenigvuldigings is $8(1) + 4(2) + 2(4) + 1(8) = 4 \times 8 = \ell \times N/2$.

Voorbeeld: TURBO PASCAL 4.0 uitgevoer op die Sperry IT rekenaar, gebruik 0.00138 sekondes vir een komplekse vermenigvuldiging. Veronderstel ons wil die DFT van 'n vektor met $N = 1024 = 2^{10}$ bereken.

— Met die gewone DFT neem dit $N^2 = 1\,048\,576$ vermenigvuldigings en die tyd is 24 minute.

— Met die FFT neem dit $\frac{1}{2}N \log_2 N = 5120$ vermenigvuldigings en die tyd is 7 sekondes, dus meer as 200 keer vinniger !

Die FFT as matriksontbinding:

Beskou 'n 4-punt DFT:

$$\begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

In die FFT opset word f_j eers herrangskik:

$$\begin{bmatrix} g_0 \\ g_1 \\ h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Dan word twee 2-punt transforms verkry

$$\begin{bmatrix} \tilde{g}_0 \\ \tilde{g}_1 \\ \tilde{h}_0 \\ \tilde{h}_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ h_0 \\ h_1 \end{bmatrix}$$

Met $w = e^{-2\pi i/4} = -i$ lyk die volgende stap so:

$$\begin{aligned} \tilde{f}_0 &= \tilde{g}_0 + (1)\tilde{h}_0 \\ \tilde{f}_1 &= \tilde{g}_1 + (-i)\tilde{h}_1 \\ \tilde{f}_2 &= \tilde{g}_0 - (1)\tilde{h}_0 \\ \tilde{f}_3 &= \tilde{g}_1 - (-i)\tilde{h}_1 \end{aligned}$$

en in matriksopset lyk dit so:

$$\begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -i \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & i \end{bmatrix} \begin{bmatrix} \tilde{g}_0 \\ \tilde{g}_1 \\ \tilde{h}_0 \\ \tilde{h}_1 \end{bmatrix}$$

Ons verkry dus die volgende spesiale ontbinding van die Fourier matriks:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -i \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & i \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

116