Gyors Fourier transzformáció vizsgálata

ZSELLÉR PÉTER

A digitális szeizmikus adatfeldolgozás során gyakran szükséges a szeizmikus csatornák spektrumának ismerete. A Fourier transzformáció elvégzése a szokásos úton gyakorlatilag lehetetlen az adatok nagy száma miatt. A cikk röviden áttekinti a Cooley-Tukey algoritmus gyakorlati megvalósítását, mellyel a transzformáció számítási idejét a reális követelmények közé lehet szorítani.

При цифровой обработке сейсмических данных часто приходится знать спектры сейсмических каналов. В связи с большим объемом данных невозможно проводить трансформацию фурье одычным способом. В работе дается короткий обзор практического осуществления алгоритма Cooley – Tukey, позволяющего сократить время вычислений по трансформации в соответствии с реальными требованиями.

Im Laufe der digitalen seismichen Datanverarbeitung benötigen wir vielfach die Kenntnis des Spektrums der seismischen Kanäle. Die Ausfühung der Fourier-Transformation auf dem üblichen Weg ist praktisch unmöglich wegen der grossen Zahl der Daten. Im Aufsatze wird die praktische Verwirklichung des Cooley – Tukeyschen Algoritmus kurz behandelt, wobei die zur Berechnung der Transformation benötigte Zeitspanne zwischen realen Grenzen gehalten werden kann.

Digitális adatrendszerek szűréséhez jelentős segítséget nyújt az adatrendszer Fourier transzformáltjának az ismerete. Ilyenkor a szűrés szorzással végezhető el, az adatrendszer és a szűrő súlyfüggvényének spektrumát kell összeszorozni. Időtartományban ez konvolúcióképzésnek felel meg, de a szűrő súlyfüggvényének meghatározásához itt is fel kell használni a Fourier transzformációt.

Egy $g(t)$ időfüggvény Fourier transzformáltja a következő:

$$G(f) = \int_{-\infty}^{\infty} g(t)\, e^{-j2\pi ft}\, dt$$

A $G(f)$ frekvenciafüggvényből az időfüggvény inverz Fourier transzformációval kapható vissza:

$$g(t) = \int_{-\infty}^{\infty} G(f)\, e^{j\,2\pi\, ft}\, df$$

Mintavételezett adatrendszerek esetében a Fourier transzformációt numerikusan kell végrehajtani.

$$G(n\,\Delta f) = \sum_{m=-\infty}^{\infty} g(m\,\Delta t)\, e^{-j\,2\pi\,\Delta f\,\Delta t\,nm}\,\Delta t$$

$$g(m\,\Delta t) = \sum_{n=-\infty}^{\infty} G(n\,\Delta f)\, e^{j\,2\pi\,\Delta f\,\Delta t\,nm}\,\Delta f$$

ahol Δt az időbeli, Δf a frekvenciatartománybeli mintavételi távolság.

Vegyük figyelembe, hogy digitális adatrendszerek Fourier transzformáltja periodikus. Egy periódus hossza időtartományban: $M\Delta t = \frac{1}{\Delta f}$, frekvenciatartományban: $N\,\Delta f = \frac{1}{\Delta t}$

Tekintsük a frekvenciabeli mintavételt egységnyinek. Ekkor $N = M$ és az előbbi összefüggések a következők lesznek:

$$G(n\,\Delta f) = \frac{1}{M} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} g(m\,\Delta t)\, e^{-j\frac{2\pi}{M}nm} \tag{1}$$

$$g(m\,\Delta t) = \sum_{n=-\frac{M}{2}}^{\frac{M}{2}} G(n\,\Delta f)\, e^{j\frac{2\pi}{M}nm} \tag{2}$$

A gyors Fourier transzformáció

A többezer pontból álló adatrendszerek transzformálása szükségessé tette a Fourier transzformáció műveleti sebességének a megnövelését. Az algoritmus kidolgozása Cooley és Tukey nevéhez fűződik.

A következőkben vázlatosan áttekintjük az algoritmus alkalmazását, az összefüggések bizonyítása nélkül.

A bizonyítások az irodalomban megtalálhatók.

Legyen az (1)-ben szereplő $e^{-j\frac{2\pi}{M}} = w$.

Ekkor az (1) mátrix alakban így írható:

$$[G_n] = [w^{nm}]\,[g_m] \tag{3}$$

ahol $[G_n]$ és $[g_m]$ M elemből álló oszlopvektorok $[w^{nm}]$, pedig egy $M \times M$ alakú mátrix.

A Cooley – Tukey algoritmus alapja, hogy ha $M = 2^k$, akkor a $[w^{nm}]$ mátrixot $k+1$ mátrix szorzatára bontják fel:.

$$\mathbf{W} = \mathbf{W}_k \cdot \mathbf{W}_{k-1}\, \mathbf{W}_{k-2} \quad \ldots\, \mathbf{W}_1 \cdot \mathbf{W}_0$$

Nézzük meg ezt a felbontást egy egyszerű példán. Legyen $M = 2^2 = 4$. Ekkor:

$$w = e^{-j\frac{2\pi}{4}}$$

Vegyük észre, hogy

$$w^{4q} = e^{-j\frac{2\pi 4q}{4}} = e^{-j\,2\pi\,q} = 1 \qquad (q = 0, 1, 2, \ldots M)$$

Ezért

$$w^{4q+p} = w^p$$

A transzformáció mátrixa a következő lesz:

$$\mathbf{W} = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 \\ w^0 & w^2 & w^4 & w^6 \\ w^0 & w^3 & w^6 & w^9 \end{pmatrix} = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 \\ w^0 & w^2 & w^0 & w^2 \\ w^0 & w^3 & w^2 & w^1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 \\ 1 & w^2 & 1 & w^2 \\ 1 & w^3 & w^2 & w \end{pmatrix}$$

Ez a mátrix a következő három mátrix szorzatára bontható:

$$\mathbf{W} = \mathbf{W}_2 \cdot \mathbf{W}_1 \cdot \mathbf{W}_0$$

ahol:

$$\mathbf{W}_0 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & w^2 & 0 \\ 0 & 1 & 0 & w^2 \end{pmatrix}$$

$$\mathbf{W}_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & w^2 & 0 & 0 \\ 0 & 0 & 1 & w \\ 0 & 0 & 1 & w^3 \end{pmatrix}$$

$$\mathbf{W}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A felbontás helyességéről a mátrixok összeszorzásával meggyőződhetünk.

A (3)-ban jelölt Fourier transzformációt tehát úgy kell elvégeznünk, hogy először a $[g_m]$ oszlopvektorral (jobbról) megszorozzuk a $\mathbf{W}_0$ mátrixot, az eredménnyel megszorozzuk a $\mathbf{W}_1$ mátrixot, ennek eredményével pedig a $\mathbf{W}_2$ mátrixot:

$$G = \mathbf{W}_2[\mathbf{W}_1(\mathbf{W}_0 g)] \tag{4}$$

A (4) kifejezés végrehajtása azért könnyebb, mint a (3), mert a $\mathbf{W}_0$ és $\mathbf{W}_1$ mátrixoknak egy-egy sorában csak két nullától különböző elem van, az egyik mindig egyes, a másik egy komplex szám, w valamilyen hatványa. Az utolsó, $\mathbf{W}_2$ mátrix alkalmazása pedig csak az adatok átrendezését jelenti.

Legyenek az egyes részeredmény oszlopvektorok G^0 és G^1. Írjuk fel ezeket:

$$G_0^0 = g_0 + w^0 g_2$$
$$G_1^0 = g_1 + w^0 g_3$$
$$G_2^0 = g_0 + w^2 g_2$$
$$G_3^0 = g_1 + w^2 g_3$$

Ezekből:

$$G_0^1 = G_0^0 + w^0 G_1^0$$
$$G_1^1 = G_0^0 + w^2 G_1^0$$

$$G_2^1 = G_2^0 + w\ G_3^0$$

$$G_3^1 = G_2^0 + w^3 G_3^0$$

Ábrázoljuk G^1 oszlopvektor elemeinek az indexét binárisan, k darab biten (k most *2* volt, mert $4 = 2^2$). A $\mathbf{W}_2$ mátrix alkalmazása nem más, mint G^1 vektor elemeinek az inverz bináris átrendezése.

Ezt úgy kapjuk meg, hogy a vektor elemeinek az indexét binárisan ábrázoljuk, és a középpontjukra nézve tükrözzük. (Esetünkben a középpont a két bit között van.)

Ezután az egyes elemeket az ily módon invertált index által kijelölt helyre tesszük.

Esetünkben az indexek:

(0) 00		*00 (0)*
(1) 01	ezek invertáltja:	*10 (2)*
(2) 10		*01 (1)*
(3) 11		*11 (3)*

Tehát a Fourier transzformáltakat a G^1 vektorból így kapjuk:

$$G_0 = G_0^1$$

$$G_1 = G_2^1$$

$$G_2 = G_1^1$$

$$G_3 = G_3^1$$

A szabályok az általános esetben is hasonlóak, mint az előbbi egyszerű példánál. Tehát, ha $M = 2^k$, akkor a transzformáció k darab számítási lépésből és egy átrendezésből áll. Miden egyes lépés eredményét úgy kapjuk, hogy az előző lépés eredményének egyik elemét megszorozzuk egy w hatvánnyal, egy másikat pedig hozzáadunk. Az adatok indexei a teljes algoritmus alatt fontos szerepet játszanak. Ha az adatrendszer hossza $M = 2^k$, akkor az indexeket, melyek *0*-tól $M-1$-ig mennek, k darab biten ábrázolhatjuk. Az általános szabályok a következőket mondják ki:

Ha $M = 2^k$, akkor a transzformáció k darab számítási lépésből és egy inverz bináris átrendezésből áll. Az m-edik számítási lépésben ($m = 1, 2, \ldots k$) az n idexű elemet a következőképpen kapjuk az $m-1$-edik lépés eredményéből.

Mint már az előzőkben láttuk, az $m-1$-edik lépés egy elemét meg kell szorozni valamilyen w hatvánnyal, és hozzá kell adni az $m-1$-edik lépés egy másik elemét.

1. A w kitevőjét úgy kapjuk, hogy a binárisan ábrázolt n indexet $k-m$ hellyel jobbra toljuk úgy, hogy a megüresedő helyekre nullákat teszünk, a lecsorduló bitek pedig elvesznek. Ezután az így kapott számot a már ismertetett módszer szerint invertáljuk.

Ez a szám lesz az alkalmazandó w kitevője.

2. Az így kapott w hatvánnyal meg kell szorozni az $m-1$-edik lépés eredményének azt az elemét, melynek indexe n, ha n bináris ábrázolásának m-edik bitjén egyes áll (a bitek számozását balról, egytől kell kezdeni). Ha

ezen a helyen nulla áll, a szorzandó indexét úgy kapjuk meg, hogy ezt a bitet egyesre változtatjuk.

3. Az előzőkben előállított szorzathoz hozzá kell adni az $m-1$-edik lépés eredményének azt az elemét, melynek indexe n, ha n bináris ábrázolásának m-edik bitjén nulla áll. Ha ezen a helyen egyes áll, a hozzáadandó elem indexét úgy kapjuk meg, hogy ezt a bitet nullára változtatjuk.

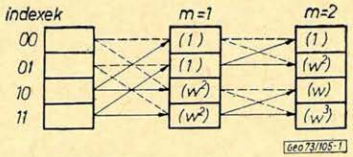
Alkalmazzuk ezeket a szabályokat $M = 4 = 2^2$ esetre! Az indexek: *00, 01, 10, 11.* Az első lépésben alkalmazandó w hatványok: *00, 00, 10, 10.* A szorzandó elemek indexei: *10, 11, 10, 11.*

A hozzáadandó elemek indexei: *00, 01, 00, 01.*

A második lépésben a w hatványok ($k - m = 0$, tehát nem kell az indexeket tolni, csak invertálni): *00, 10, 01, 11.*

A szorzandók indexei: *01, 01, 11, 11.* A hozzáadandóké: *00, 00, 10, 10.*

Ezt mutatja az *1. ábra.* A szorzást nyilak, az összeadást vonalak jelzik. A zárójelekben az alkalmazott szorzók láthatók.



1. ábra. Az FFT algoritmus alkalmazása négy-pontos adatrendszer transzformálására

Рис. 1. Применение алгоритма FFT для трансформации четырех-точечной системы данных

Abb. 1. Anwendung des FFT-Algoritmus zur Transformation eines Vierpunkt-Datensystemy

Az algoritmus alkalmazása jelentős időnyereséget jelent a hagyományos módszerrel szemben. Ha a Fourier transzformációt a hagyományos módszer szerint akarjuk végrehajtani, $M = 2^k$ adat esetén M^2 komplex szorzást kell végeznünk. Az *FFT* algoritmussal $2Mk$ komplex szorzással kapjuk meg az eredményt. A kettő aránya:

$$\frac{M^2}{2Mk} = \frac{2^{k-1}}{k}$$

Ez egy $2K-s$ adatrendszerre kb. *100*-szoros időnyereséget jelent.

Igazi időnyereséget az FFT egységek alkalmazása jelent. Ezek speciális számítógépek, melyek a szorzást és az összeadást gyorsítva végzik el.

Különböző gyártmányú FFT egységek az algoritmust más-más úton hajtják végre a memóriakapacitás kihasználása, és a minél nagyobb időnyereség érdekében.

Tájékoztatásul közöljük az OKGT GKÜ-ben működő FFT egység működési idejét *msec*-ban.

1. táblázat – таблица – Tabelle

Működési idők msec-ban.

Az adatrendszer hossza	Működési mód		
	Fourier transzformálás	Inverz transzformálás	Szűrés
2 K	35	40	80
4 K	65	90	175
8 K	130	200	365

Az FFT-vel való számolás során több gyakorlati probléma vetődik fel. A problémák egy része abból adódik, hogy nem folytonos függvényekkel, hanem digitális adatrendszerekkel dolgozunk. A digitális adatrendszerek spektruma periodikus. Mivel a spektrumot is digitális formában állítjuk elő, az ezekből inverz Fourier transzformációval visszaállított adatrendszer is periodikus lesz. Egy periódus hossza időtartományban:

$$T = \frac{1}{\Delta f}$$

frekvenciatartományban:

$$F = \frac{1}{\Delta t}$$

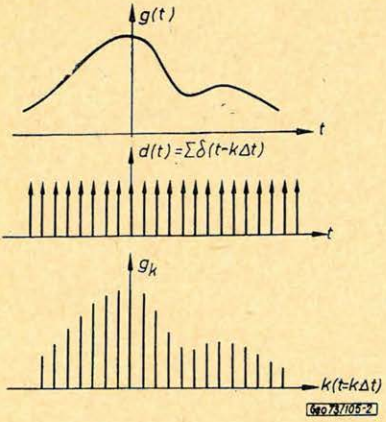
ahol Δf a frekvenciatartománybeli, Δt az időtartománybeli mintavételi távolság.

Ha az adatrendszerünk helyesen volt mintavételezve, ez a periodikusság nem okoz problémát. A helyes mintavételezés ugyanis azt jelenti, hogy az adatrendszerrel ekvivalens folytonos függvény Fourier transzformáltja nem *„lóg túl"* a periódus határán. Pontosabban fogalmazva, akkor helyes a mintavételezés, ha a függvény transzformáltja egy f_H határfrekvencia felett zérus, és

$$f_H < \frac{F}{2} = \frac{1}{2\Delta t}$$

ugyanez érvényes az inverz esetre is:

$$g(t) = 0, \quad \text{ha} \quad t > t_H; \quad \text{és} \quad t_H < \frac{1}{2\Delta f}$$



Ha a mintavételi törvény nem teljesül, a transzformáltakban torzulások lépnek fel.

Legyen az adatrendszerünk a $g(t)$ folytonos függvény mintavételezett formája. Ekkor az adatrendszer a $g(t)$ függvény, és egy olyan Dirac-delta sorozat szorzata, mely a mintavételi helyek kivételével mindenütt nulla *(2. ábra)*.

Mivel az időtartománybeli szorzás frekvenciatartománybeli konvolúciónak felel meg, felel meg a g_k adatrendszer spektrumát úgy kapjuk, hogy konvolváljuk $g(t)$ Fourier transzformáltját $d(t)$ transzformáltjával.

2. ábra. Folytonos függvény mintavételezése Dirac-delta sorozat segítségével
Рис. 2. Взятие выборок из непрерывной функции при помощи серии Дирах-дельта
Abb. 2. Probeentnahme einer kontinuirlichen Funktion mit Hilfe einer Dirac-Delta-Serie

A $d(t)$ függvény spektruma szintén egy Dirac-delta sorozat:

$$D(f) = \frac{1}{\Delta t} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{\Delta t}\right)$$

Tehát, ha $g(t)$ spektruma $G(f)$, akkor g_k spektruma:

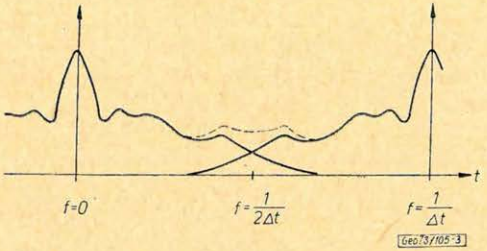
$$G^*(f) = \frac{1}{\Delta t} \sum_{k=-\infty}^{\infty} G\left(f - \frac{k}{\Delta t}\right) \tag{5}$$

Ez látszik, hogy Δt szerint periodikus.

Másrészt viszont a $G^*(f)$ függvényt úgy kapjuk, hogy $G(f)$-et egy-egy periódussal eltoljuk és összeadjuk.

A periódus hossza a mintavételi távolságtól függ. Ha ezt rosszul választottuk, és az összeadandó $G(f) \neq 0$, ha $f > \frac{1}{2\Delta t}$, azaz $G(f)$ „*kilóg*" a periódusból, a kilógó részek összeadódnak és a spektrum eltorzul *(3. ábra)*.

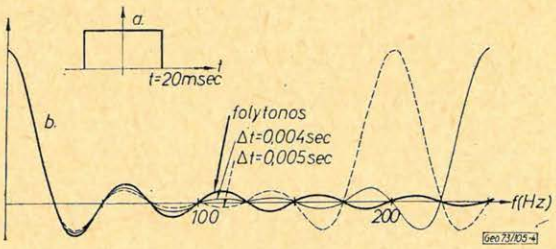
A következő példa ezt a torzulást illusztrálja *(4. ábra)*. Legyen a transzformálandó függvény egy négyszögfüggvény. Ennek a Fourier transzformáltja egy sinus cardinális függvény. Az *a ábrán* négyszögfüggvény, *a*, *b ábrán* a függvény transzformáltjai láthatók, különböző mintavétel mellett.



3. ábra. A helytelen mintavételezés torzító hatása

Рис. 3. Искажающий эффект неправильного взятия выборок

Abb. 3. Verzerreffekt einer unrichtigen Probeentnahme



4. ábra. Négyszögfüggvény Fourier transzformáltja, különböző mintavételezés esetén

Рис. 4. Трансформация Фурье прямоугольной функции при различных шагах выборок

Abb. 4. Fourier-Transformierte einer Viereck-Funktion im Falle verschiedener Probeentnahmen

Az előbbi négyszögfüggvény vizsgálata azért érdekes, mert tulajdonképpen soha nem tudunk úgy mintavételezni, hogy a mintavételi törvény teljesüljön, ugyanis nem tudunk megadni egy olyan nagy f_H felső határfrekvenciát, hogy

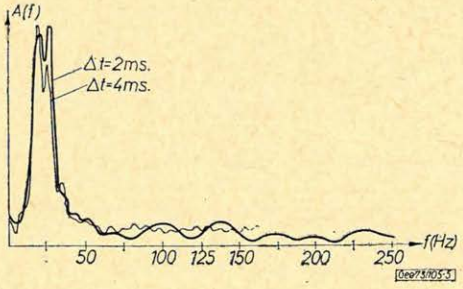
$$\sin c\, 2fT \equiv 0 \quad \text{legyen, ha} \quad |f| > f_H$$

A valóságban is mindig ehhez hasonló esettel állunk szemben. Ilyenkor nem tudunk mást mondani, mint, hogy a $g(t)$ függvényt akkor mintavételeztük helyesen, ha $\Delta t < \frac{1}{2f_H}$, és $g(t)$ spektrumának abszolút értéke $|G(f)| < \varepsilon$, ha $|f| > f_H$, ahol ε egy elhanyagolhatóan kicsi érték.

Szeizmikus gyakorlatban a tapasztalatok szerint az így választott felső határferkvencia nem nagyobb, mint *250 Hz*, ezért a *0,002 sec*-os mintavételi távolság megfelelő.

Az információt hordozó hasznos jelek frekvenciája a *20 – 60 Hz*-es tartományba esik. Ez a nagyobb, *4*, esetleg *8 msec*-os mintavételt is lehetővé tenné. Az áttérés a 2 msec-os mintavételről nagyobbra nem lehet az adatok kihagyása, mert ez a mintavételi törvény megsértését jelentené, ami az (5) szerinti összeadódási jelenséget hozná létre.

Ilyenre mutat példát az *5. ábra.*



5. ábra. A helytelen adatritkítás hatása egy szeizmikus csatorna amplitudósprektrumára

Рис. 5. Влияние неправильного разрежения данных на амплитудный спектр сейсмического канала

Abb. 5. Einfluss einer unrichtigen Datenverdünnung an das Amplitudenspektrum eines seismischen Kanals

Ha például egy *2 msec*-os mintavétellel digitalizált adatrendszerből minden második adatot kihagyunk, az eredetiben *200 Hz*-es zaj az újban úgy jelentkezik, mintha *50 Hz*-es lenne. Ezért a ritkítást úgy kell végrehajtani, hogy először egy felülvágó szűrővel a kritikus frekvenciákat eltávolítjuk. Ez az eltávolítás természetesen történhet analóg szűrőkkel is, még a felvétel készítésekor. Felhasználhatjuk az FFT egységet is az előbbiekben leírt ritkításra, de sűrítésre is. Az OGKT GKÜ-ben üzemelő FFT egység *2K*-s, *4K*-s és *8K*-s adatrendszerrel tud dolgozni. A frekvenciaskálát mindig úgy választja meg, hogy a teljes $f = 0 - f_{\text{Nyquist}}$ intervallumot kitöltse. Ezért egy ritkítási műveletet a következőképpen hajthatunk végre: a ritkítandó csatornát először (a csatornahossztól függően *4K* vagy *8K* üzemmódban) Fourier transzformáljuk. Ezután fele akkora adathosszban (*2K* vagy *4K* üzemmódban) inverz Fourier transzformájuk. Tegyük fel, hogy *2 msec*-os mintavételű volt az eredeti csatornánk.

Ekkor a spektrumot *0 – 250 Hz*-ig állítja elő az FFT. Amikor fele akkora adathosszt transzformálunk vissza, akkor tulajdonképpen kiszűrjük a *125 Hz* feletti spektrum tartományt. Mivel a *125 Hz*-es *Nyquist* frekvenciához *4 msec*-os időtartománybeli mintavételi távolság tartozik, az eredményünk már a fele olyan hosszú kiritkított adatrendszer lesz.

Hasonló eljárást követhetünk az adatrendszerek sűrítésekor. Ilyenkor az adatrendszer Fourier transzformálása után a spektrumot kiegészítjük nullákkal, mégegyszer olyan hosszúra. Ezután ezt a kétszeres hosszúságúra kiegészített adatrendszert transzformáljuk vissza.

Ha a bemenetünk *2 msec*-os mintavételezésű volt, a spektrumot *0 – 250 Hz*-ig kapjuk meg. Ezt kiegészítjük *500 Hz*-ig nullákkal, ezért az inverz transzformáció után *1 msec*-os lesz az időbeli mintavételi távolság. Mivel frekvenciatartományban egy négyszögablakot alkalmaztunk, ez az eljárás a sinus cardinálisos interpolációnak felel meg.

Kétváltozós Fourier transzformáció

Az FFT egység segítségével kétváltozós Fourier transzformáció is végrehajtható. Egy kétváltozós $g(x, y)$ függvény Fourier transzformáltja:

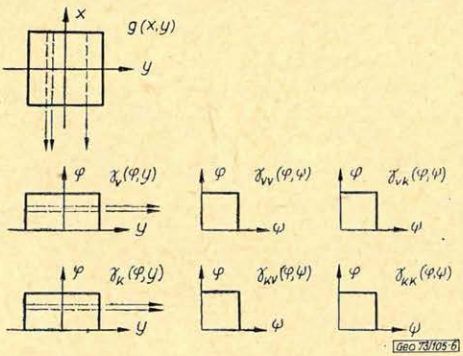
$$G(\varphi, \psi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x,y)\, e^{-j\,2\pi(\varphi x + \psi y)}\, dx\, dy$$

Jelöljük az x szerinti Fourier transzformáltakat $F_x\{\ldots\}$-szel, az y szerintit $F_y\{\ldots\}$-nal.

Ekkor:

$$G(\varphi, \psi) = F_y\{F_x[g(x, y)]\}$$

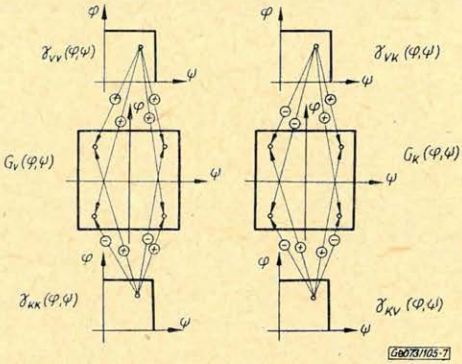
Így felbontva a művelet az FFT egységgel végrehajtható. Maga a transzformálásra fordított idő viszonylag rövid (az OKGT GKÜ FFT egységén egy $2K$-szor $2K$-s adatrendszer transzformációs ideje kb. *2,5* perc), de ehhez hozzá kell adni az adatmozgatások idejét, ami a használt perifériáktól függően az előbbinek sokszorosa lehet. A *6.* és *7. ábra* egy $g(x, y)$ függvény kétváltozós Fourier transzformációjának megvalósítását mutatja.



6. ábra. Kétváltozós Fourier transzformáció megvalósítása két lépésben

Рис. 6. Осуществление трансформации Фурье с двумя переменными в два приема

Abb. 6. Verwirklichung einer zweiveränderlichen Fourier-Transformation in zwei Schritten



7. ábra. Kétváltozós Fourier transzformált valós és képzetesrészének előállítása

Рис. 7. Формирование действительной и мнимой частей трансформанта Фурье

Abb. 7. Realisation des reellen und imaginären Teiles einer zweiveränderlichen-Fourier-Transformierten

Először konstans y-ok mellett az x változó szerint transzformáltunk, ebből egy $\gamma_v(\varphi, y)$ valós és egy $\gamma_k(\varphi, y)$ képzetes adatrendszert kapunk.

Ezeket tovább transzformálva az y változó szerint, kapjuk a γ_{vv}, γ_{vk}, γ_{kv}, γ_{kk} adatrendszereket. (A v index a valós, a k a képzetes részre utal.)

Ebből a négy adatrendszerből a *7. ábra* szerint állíthatjuk elő a $g(x, y)$ függvény Fourier transzformáltjának $G_v(\varphi, \psi)$ valós és $G_k(\varphi, \psi)$ képzetes részét, úgy, hogy az egyes elemeket a bekarikázott előjelekkel összeadjuk.

IRODALOM

Meskó A.: Digitális Szeizmika Alapjai, Egyetemi jegyzet, 1971.

Ron Bracewell: The Fourier Transform and Its Applications, Mc Graw – Hill Electrical and Electronic Engineering Series, 1965.