

(curs 11 - 56)

8. Interpolarea trigonometrică și TFR

8.1. Transformata Fourier.

8.1.1. Numerele complexe

- calculele pentru funcțiile trigonometrice pot fi mult simplificate prin adoptarea limbajului numerelor complexe
- orice număr complex este de forma $z = a + bi$, unde $i = \sqrt{-1}$
- fiecare z este reprezentat geometric ca un vector bidimensional de lungime a de-a lungul axei reale (orizontale), și de lungime b de-a lungul axei imaginare (verticale), după cum se arată în Figura 1

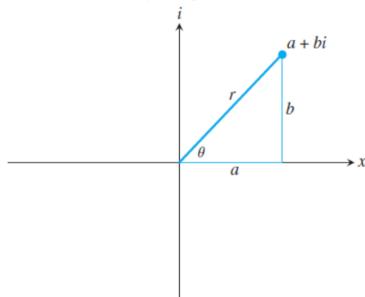


Figura 1: Reprezentarea unui număr complex. Părțile reală și imaginară sunt a , și, respectiv, bi . Reprezentarea polară este $a + bi = re^{i\theta}$.

- norma complexă** a numărului $z = a + bi$ este definită ca fiind $|z| = \sqrt{a^2 + b^2}$ și este exact distanța numărului complex de la originea planului complex
- conjugatul complex** al numărului complex $z = a + bi$ este $\bar{z} = a - bi$
- formula lui Euler** pentru numere complexe ne spune că $e^{i\theta} = \cos \theta + i \sin \theta$
- norma complexă a lui $z = e^{i\theta}$ este 1, astfel că numerele complexe de această formă se află pe cercul unitate din planul complex, după cum se arată în Figura 2
- orice număr complex $a + bi$ poate fi scris în reprezentarea polară, astfel:

$$z = a + bi = re^{i\theta}, \quad (1)$$

unde r este norma complexă $|z| = \sqrt{a^2 + b^2}$ și $\theta = \arctan b/a$

- cercul unitate din planul complex corespunde numerelor complexe de normă $r = 1$

- pentru a înmulți două numere $e^{i\theta}$ și $e^{i\gamma}$ de pe cercul unitate, putem transforma în funcții trigonometrice, și apoi efectua înmulțirea:

$$\begin{aligned} e^{i\theta} e^{i\gamma} &= (\cos \theta + i \sin \theta)(\cos \gamma + i \sin \gamma) \\ &= \cos \theta \cos \gamma - \sin \theta \sin \gamma + i(\sin \theta \cos \gamma + \sin \gamma \cos \theta). \end{aligned}$$

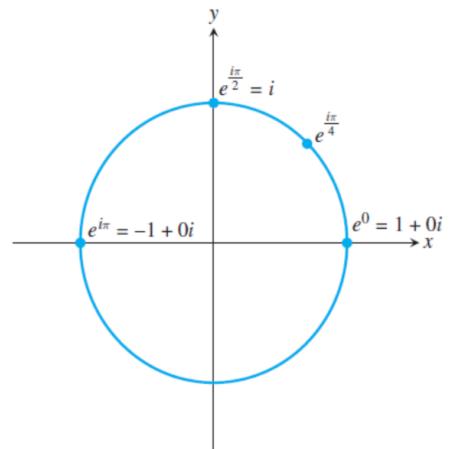
- recunoscând formulele pentru adunarea unghiurilor la cos și la sin, putem scrie

$$\cos(\theta + \gamma) + i \sin(\theta + \gamma) = e^{i(\theta+\gamma)}.$$

- echivalent, putem doar aduna exponentii:

$$e^{i\theta} e^{i\gamma} = e^{i(\theta+\gamma)}. \quad (2)$$

- ecuația (2) ne arată că produsul a două numere de pe cercul unitate ne dă un nou punct aflat pe cercul unitate, al cărui unghi este suma celor două unghiuri
- formula lui Euler ascunde detalii de trigonometrie, cum ar fi formulele pentru adunarea unghiurilor, și face calculele mult mai ușoare
- acesta este motivul pentru care introducem numerele complexe în studiul interpolării trigonometrice



- un număr complex z este o **rădăcină de ordinul n a unității** dacă $z^n = 1$
- pe axa numerelor reale, există doar două rădăcini ale unității, -1 și 1
- în planul complex, însă, există mai multe
- de exemplu, i este o rădăcină de ordinul 4 a unității, deoarece $i^4 = (-1)^2 = 1$
- o rădăcină de ordinul n a unității se numește **primitivă** dacă nu este o rădăcină de ordinul k a unității, pentru $k < n$
- din această definiție, -1 este o rădăcină primitivă de ordinul doi a unității și o rădăcină neprimitivă de ordinul patru a unității
- este ușor de verificat că, pentru orice n , numărul complex $\omega_n = e^{-i2\pi/n}$ este o rădăcină primitivă de ordinul n a unității
- numărul $e^{i2\pi/n}$ este tot o rădăcină primitivă de ordinul n a unității, dar vom urma convenția obișnuită de a folosi pe prima dintre ele ca bază pentru transformata Fourier

• vom da acum o identitate cheie, de care vom avea nevoie mai târziu, pentru a face mai simplă calcularea Transformantei Fourier Discrete

- fie ω rădăcina de ordin n a unității $\omega = e^{-i2\pi/n}$, unde $n > 1$

• atunci

$$1 + \omega + \omega^2 + \omega^3 + \cdots + \omega^{n-1} = 0. \quad (3)$$

- demonstrația acestei identități rezultă din suma telescopică

$$(1 - \omega)(1 + \omega + \omega^2 + \omega^3 + \cdots + \omega^{n-1}) = 1 - \omega^n = 0. \quad (4)$$

- deoarece primul termen din stânga nu este zero, al doilea trebuie să fie zero

- o metodă asemănătoare poate fi folosită pentru a demonstra că

$$\begin{aligned} 1 + \omega^2 + \omega^4 + \omega^6 + \cdots + \omega^{2(n-1)} &= 0 \\ 1 + \omega^3 + \omega^6 + \omega^9 + \cdots + \omega^{3(n-1)} &= 0 \\ &\vdots \\ 1 + \omega^{n-1} + \omega^{(n-1)2} + \omega^{(n-1)3} + \cdots + \omega^{(n-1)(n-1)} &= 0. \end{aligned} \quad (5)$$

- următoarea este diferită:

$$\begin{aligned} 1 + \omega^n + \omega^{2n} + \omega^{3n} + \cdots + \omega^{n(n-1)} &= 1 + 1 + 1 + 1 + \cdots + 1 \\ &= n. \end{aligned} \quad (6)$$

- aceste informații sunt colectate în următoarea lemă

Lema 1 (Rădăcinile primitive ale unității)

Fie ω o rădăcină primitivă de ordinul n a unității și k un întreg. Atunci

$$\sum_{j=0}^{n-1} \omega^{jk} = \begin{cases} n & \text{dacă } k/n \text{ este un întreg} \\ 0 & \text{altfel.} \end{cases}$$

8.1.2. Transformata Fourier disreta

- fie $x = [x_0, \dots, x_{n-1}]^T$ un vector n -dimensional (cu valori reale), și notăm $\omega = e^{-i2\pi/n}$
- definiția fundamentală din acest capitol este dată mai jos

Definiția 1

Transformata Fourier Discretă (TFD) a lui $x = [x_0, \dots, x_{n-1}]^T$ este vectorul n -dimensional $y = [y_0, \dots, y_{n-1}]^T$, unde $\omega = e^{-i2\pi/n}$ și

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}. \quad (7)$$

- de exemplu, Lema 1 ne arată că TFD a lui $x = [1, 1, \dots, 1]^T$ este $y = [\sqrt{n}, 0, \dots, 0]^T$

- Figura 3 prezintă o rădăcină primitivă de ordinul opt a unității $\omega_8 = e^{-i2\pi/8}$, și celelalte șapte rădăcini ale unității, care sunt puteri ale lui ω_8

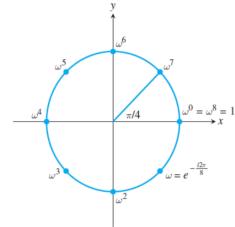


Figura 3: **Rădăcinile unității.** Rădăcinile de ordinul 8 ale unității sunt prezentate. Ele sunt generate de către $\omega = e^{-i2\pi/8}$, ceea ce înseamnă că fiecare rădăcină este de forma ω^k , pentru un anumit întreg k . Deși ω și ω^3 sunt rădăcini primitive de ordinul 8 ale unității, ω^2 nu este, pentru că este, de asemenea, o rădăcină de ordinul 4 a unității.

- în termeni de matrici, această definiție devine

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 + ib_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ \vdots \\ a_{n-1} + ib_{n-1} \end{bmatrix} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}. \quad (8)$$

- fiecare $y_k = a_k + ib_k$ este un număr complex
- matricea $n \times n$ din (8) se numește **matrice Fourier**:

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}. \quad (9)$$

- cu excepția rândului de sus, fiecare rând din matricea Fourier are suma egală cu zero

- același lucru este valabil și pentru coloane, deoarece F_n este o matrice simetrică
- matricea Fourier are o inversă explicită

$$F_n^{-1} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \cdots & \omega^{-3(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)^2} \end{bmatrix}, \quad (10)$$

și **Transformata Fourier Discretă inversă** a vectorului y este $x = F_n^{-1}y$

- verificarea faptului că (10) este inversa matricii F_n necesită Lema 1 despre rădăcinile de ordinul n ale unității
- fie $z = e^{i\theta} = \cos \theta + i \sin \theta$ un punct de pe cercul unitate
- atunci inversul său $e^{-i\theta} = \cos \theta - i \sin \theta$ este conjugatul său complex
- prin urmare, TFD inversă este matricea conjugatelor complexe ale intrărilor matricii F_n :

$$F_n^{-1} = \overline{F_n}.$$

Definiția 2

Norma unui vector complex v este numărul real $\|v\| = \sqrt{v^T v}$. O matrice complexă F este **unitară** dacă $\overline{F}^T F = I$.

- o matrice unitară, cum ar fi matricea Fourier, este versiunea complexă a matricilor reale ortogonale
- dacă F este unitară, atunci $\|Fv\|^2 = \overline{v}^T \overline{F}^T F v = \overline{v}^T v = \|v\|^2$
- prin urmare, norma unui vector este neschimbată la înmulțirea la stânga cu F —sau cu F^{-1}
- aplicarea Transformantei Fourier Discrete se reduce la înmulțirea cu matricea $n \times n$ F_n , și prin urmare necesită $O(n^2)$ operații (mai exact, n^2 înmulțiri și $n(n-1)$ adunări)
- Transformata Fourier Discretă inversă, care este aplicată prin înmulțirea cu F_n^{-1} , necesită tot $O(n^2)$ operații
- în Subsecțiunea 8.1.3, vom dezvolta o versiune a TFD care necesită semnificativ mai puține operații, numită Transformata Fourier Rapidă

Exemplul 1

- găsiți TFD a vectorului $x = [1, 0, -1, 0]^T$
- fie ω rădăcina de ordinul 4 a unității, sau $\omega = e^{-i\pi/2} = \cos(\pi/2) - i \sin(\pi/2) = -i$
- aplicând TFD, obținem

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \quad (11)$$

- chiar dacă vectorul x are componente care sunt numere reale, nu există niciun motiv pentru ca elementele lui y să fie numere reale
- dar dacă x_j sunt reale, numerele complexe y_k au o proprietate specială:

Lema 2

Fie $\{y_k\}$ TFD a lui $\{x_j\}$, unde x_j sunt numere reale. Atunci (a) y_0 este real, și (b) $y_{n-k} = \overline{y_k}$ pentru $k = 1, \dots, n-1$.

- motivul pentru care (a) este adevărat este clar din (7), deoarece y_0 este suma valorilor x_j împărțită la \sqrt{n}
- partea (b) rezultă din faptul că

$$\omega^{n-k} = e^{-i2\pi(n-k)/n} = e^{-i2\pi} e^{i2\pi k/n} = \cos(2\pi k/n) + i \sin(2\pi k/n),$$

în vreme ce

$$\omega^k = e^{-i2\pi k/n} = \cos(2\pi k/n) - i \sin(2\pi k/n),$$

ceea ce implică faptul că $\omega^{n-k} = \overline{\omega^k}$

- din definiția transformatei Fourier,

$$\begin{aligned} y_{n-k} &= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j (\omega^{n-k})^j \\ &= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j (\overline{\omega^k})^j = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \overline{x_j (\omega^k)^j} = \overline{y_k}. \end{aligned}$$

- aici am folosit faptul că produsul conjugatelor complexe este conjugatul complex al produsului
- Lema 2 are o consecință interesantă
- fie n par și x_0, \dots, x_{n-1} numere reale
- atunci TFD le înlocuiește cu exact n alte numere reale $a_0, a_1, b_1, a_2, b_2, \dots, a_{n/2},$ părțile reale și imaginare ale transformatei Fourier y_0, \dots, y_{n-1}
- de exemplu, TFD pentru $n = 8$ are forma

$$F_8 \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ a_3 + ib_3 \\ a_4 \\ a_3 - ib_3 \\ a_2 - ib_2 \\ a_1 - ib_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix}. \quad (12)$$

8.1.3. Transformata Fourier Rapida

- după cum am menționat în subsecțiunea anterioară, Transformata Fourier Discretă aplicată unui n -vector în modul tradițional necesită $O(n^2)$ operații
- Cooley și Tukey au găsit o modalitate de a calcula TFD în $O(n \log n)$ operații într-un algoritm numit **Transformata Fourier Rapidă** (TFR)
- popularitatea TFR pentru analiza datelor a urmat aproape imediat
- domeniul procesării de semnale s-a transformat din analogic în digital în mare parte datorită acestui algoritm
- vom explica metoda lor și îi vom arăta superioritatea față de TFD naivă (8) prinț-o numărare a operațiilor
- putem să scriem TFD $F_n x$ ca

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = \frac{1}{\sqrt{n}} M_n \begin{bmatrix} x_0 \\ \vdots \\ x_{n-1} \end{bmatrix},$$

- unde

$$M_n = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \dots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{bmatrix}.$$

- vom arăta cum se poate calcula $z = M_n x$ recursiv
- la încheierea calculului TFD necesită împărțirea cu \sqrt{n} , sau $y = F_n x = z / \sqrt{n}$
- începem prin să arătăm cum funcționează cazul $n = 4$, pentru a înțelege ideea algoritmului
- cazul general va fi atunci clar
- fie $\omega = e^{-i2\pi/4} = -i$
- transformata Fourier Discretă va fi

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}. \quad (13)$$

- detaliem produsul matricial, dar rearanjăm ordinea termenilor astfel încât termenii cu puteri pare apar primii:

$$\begin{aligned} z_0 &= \omega^0 x_0 + \omega^0 x_2 + \omega^0 (\omega^0 x_1 + \omega^0 x_3) \\ z_1 &= \omega^0 x_0 + \omega^2 x_2 + \omega^1 (\omega^0 x_1 + \omega^2 x_3) \\ z_2 &= \omega^0 x_0 + \omega^4 x_2 + \omega^2 (\omega^0 x_1 + \omega^4 x_3) \\ z_3 &= \omega^0 x_0 + \omega^6 x_2 + \omega^3 (\omega^0 x_1 + \omega^6 x_3). \end{aligned}$$

- folosind faptul că $\omega^4 = 1$, putem rescrie aceste ecuații sub forma

$$\begin{aligned} z_0 &= (\omega^0 x_0 + \omega^0 x_2) + \omega^0 (\omega^0 x_1 + \omega^0 x_3) \\ z_1 &= (\omega^0 x_0 + \omega^2 x_2) + \omega^1 (\omega^0 x_1 + \omega^2 x_3) \\ z_2 &= (\omega^0 x_0 + \omega^0 x_2) + \omega^2 (\omega^0 x_1 + \omega^0 x_3) \\ z_3 &= (\omega^0 x_0 + \omega^2 x_2) + \omega^3 (\omega^0 x_1 + \omega^2 x_3). \end{aligned}$$

- observăm că fiecare termen din paranteze din primele două rânduri este repetat la fel în ultimele două rânduri
- definim

$$\begin{aligned} u_0 &= \mu^0 x_0 + \mu^0 x_2 \\ u_1 &= \mu^0 x_0 + \mu^1 x_2 \end{aligned}$$

și

$$\begin{aligned} v_0 &= \mu^0 x_1 + \mu^0 x_3 \\ v_1 &= \mu^0 x_1 + \mu^1 x_3, \end{aligned}$$

unde $\mu = \omega^2$ este rădăcina de ordinul 2 unității

- atât $u = [u_0, u_1]^T$ cât și $v = [v_0, v_1]^T$ sunt practic TFD-uri cu $n = 2$; mai precis,

$$\begin{aligned} u &= M_2 \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} \\ v &= M_2 \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}. \end{aligned}$$

- putem să scriem produsul inițial $M_4 x$ sub forma

$$\begin{aligned} z_0 &= u_0 + \omega^0 v_0 \\ z_1 &= u_1 + \omega^1 v_1 \\ z_2 &= u_0 + \omega^2 v_0 \\ z_3 &= u_1 + \omega^3 v_1. \end{aligned}$$

- în concluzie, calculul lui TFD(4) a fost redus la o pereche de TFD(2)-uri plus câteva înmulțiri și adunări suplimentare
- ignorând factorul $1/\sqrt{n}$ pentru moment, TFD(n) poate fi redusă la calculul a două TFD($n/2$)-uri plus încă $2n - 1$ operații ($n - 1$ înmulțiri și n adunări)
- o numărare atentă a adunărilor și înmulțirilor necesare ne dă Teorema 1

Teorema 1 (Numărarea operațiilor pentru TFR)

Fie n o putere a lui 2. Atunci Transformata Fourier Rapidă de dimensiune n poate fi calculată în $n(2 \log_2 n - 1) + 1$ adunări și înmulțiri, plus o împărțire cu \sqrt{n} .

- ignorăm rădăcina pătrată pentru moment, care este aplicată la sfârșit
- rezultatul este echivalent cu a zice că TFD(2^m) poate fi calculată în $2^m(2m - 1) + 1$ adunări și înmulțiri
- de fapt, am văzut mai sus că TFD(n), unde n este par, poate fi redusă la o pereche de TFD($n/2$)-uri
- dacă n este o putere a lui doi—de exemplu, $n = 2^m$ —atunci putem sparge problema recursiv până când ajungem la TFD(1), care este înmulțirea cu matricea identitate 1×1 , ceea ce necesită zero operații
- începând de jos în sus, TFD(1) nu necesită nicio operație, și TFD(2) necesită două adunări și o înmulțire: $y_0 = u_0 + 1 v_0$, $y_1 = u_0 + \omega v_0$, unde u_0 și v_0 sunt TFD(1)-uri (și anume, $u_0 = y_0$ și $v_0 = y_1$)
- TFD(4) necesită două TFD(2)-uri plus $2 * 4 - 1 = 7$ operații în plus, totalizând $2(3) + 7 = 2^m(2m - 1) + 1$ operații, unde $m = 2$
- continuăm prin inducție: presupunem că formula este corectă pentru un anumit m

- atunci TFD(2^{m+1}) necesită două TFD(2^m)-uri, care necesită $2(2^m(2m - 1) + 1)$ operații, plus încă $2 \cdot 2^{m+1} - 1$, totalizând

$$\begin{aligned} 2(2^m(2m - 1) + 1) + 2^{m+2} - 1 &= 2^{m+1}(2m - 1 + 2) + 2 - 1 \\ &= 2^{m+1}(2(m + 1) - 1) + 1. \end{aligned}$$

- prin urmare, formula cu $2^m(2m - 1) + 1$ operații este demonstrată pentru versiunea rapidă a TFD(2^m), de unde rezultă enunțul teoremei
- algoritmul rapid pentru TFD poate fi exploatat pentru a da naștere unui algoritm rapid pentru TFD inversă fără efort suplimentar
- TFD inversă este matricea complexă conjugată \bar{F}_n
- pentru a efectua TFD inversă a unui vector complex y , conjugăm, aplicăm TFR, și apoi conjugăm rezultatul, pentru că

$$F_n^{-1}y = \bar{F}_n y = \bar{F}_n \bar{y}. \quad (14)$$

8.2.1. Teorema de interpolare a TFD

- fie $[c, d]$ un interval și n un număr întreg pozitiv
- definim $\Delta t = (d - c)/n$ și $t_j = c + j\Delta t$ pentru $j = 0, \dots, n - 1$ ca fiind puncte egal depărtate din acest interval
- pentru un vector de intrare x dat transformatei Fourier, vom interpreta componenta x_j ca cea de-a j -a componentă a unui semnal măsurat
- de exemplu, putem să ne gândim la componentele lui x ca la o serie de măsurători, efectuate la momentele de timp discrete și egal depărtate t_j , după cum se arată în Figura 4
- fie $y = F_n x$ TFD a lui x
- deoarece x este TFD inversă a lui y , putem scrie o formulă explicită pentru componentele lui x din (10), amintindu-ne că $\omega = e^{-i2\pi/n}$:

$$x_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k (\omega^{-k})^j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k e^{i2\pi k j / n} = \sum_{k=0}^{n-1} y_k \frac{e^{i2\pi k(t_j - c)}}{\sqrt{n}}. \quad (15)$$

- putem vedea aceasta ca interpolarea punctelor (t_j, x_j) prin funcțiile de bază trigonometrice, unde coeficienții sunt y_k

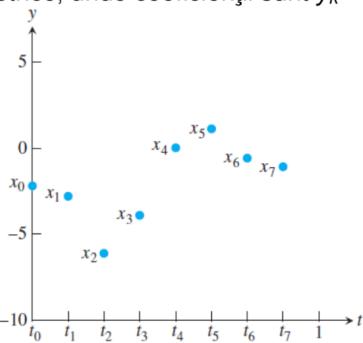


Figura 4: Componentele lui x privite ca o serie de timp. Transformata Fourier este o modalitate de a calcula polinomul trigonometric care interpolează aceste date.

Teorema 2 este o simplă reformulare a lui (15), care ne spune că punctele (t_j, x_j) sunt interpolate de către funcțiile de bază $e^{i2\pi k(t - c)/(d - c)} / \sqrt{n}$ pentru $k = 0, \dots, n - 1$, unde coeficienții de interpolare sunt date de către $F_n x$

Teorema 2 (Teorema de interpolare a TFD)

Fiind dat un interval $[c, d]$ și un întreg pozitiv n , fie $t_j = c + j(d - c)/n$ pentru $j = 0, \dots, n - 1$, și fie $x = [x_0, \dots, x_{n-1}]^T$ un vector de n numere. Definim $[a_0, \dots, a_{n-1}]^T + [b_0, \dots, b_{n-1}]^T i = F_n x$, unde F_n este matricea Transformantei Fourier Discrete. Atunci funcția complexă

$$Q(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k + i b_k) e^{i2\pi k(t - c)/(d - c)}$$

satisfacă $Q(t_j) = x_j$ pentru $j = 0, \dots, n - 1$. Mai mult, dacă valorile x_j sunt reale, funcția reală

$$P(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(a_k \cos \frac{2\pi k(t - c)}{d - c} - b_k \sin \frac{2\pi k(t - c)}{d - c} \right)$$

satisfacă $P(t_j) = x_j$ pentru $j = 0, \dots, n - 1$.

- cu alte cuvinte, transformata Fourier F_n transformă datele $\{x_j\}$ în coeficienți de interpolare
- explicația pentru ultima parte a teoremei este că, folosind formula lui Euler, putem rescrie funcția de interpolare din (15) ca

$$Q(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k + ib_k) \left(\cos \frac{2\pi k(t-c)}{d-c} + i \sin \frac{2\pi k(t-c)}{d-c} \right).$$

- separăm funcția de interpolare $Q(t) = P(t) + iI(t)$ în părțile ei reală și imaginară
- deoarece x_j sunt numere reale, doar partea reală a lui $Q(t)$ este necesară pentru a interpola x_j
- partea reală este

$$P(t) = P_n(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(a_k \cos \frac{2\pi k(t-c)}{d-c} - b_k \sin \frac{2\pi k(t-c)}{d-c} \right). \quad (16)$$

- un indice n identifică numărul de termeni dintr-un model trigonometric
- vom numi câteodată P_n ca fiind o **funcție trigonometrică de ordinul n**
- Lema 2 și următoarea Lemă 3 pot fi folosite pentru a simplifica funcția de interpolare $P_n(t)$ în continuare:

Lema 3

Fie $t = j/n$, unde j și n sunt numere întregi. Fie k un număr întreg.
Atunci

$$\cos 2(n-k)\pi t = \cos 2k\pi t \text{ și } \sin 2(n-k)\pi t = -\sin 2k\pi t. \quad (17)$$

- de fapt, formula de adunare pentru cosinus ne dă $\cos 2(n-k)\pi j/n = \cos(2\pi j - 2jk\pi/n) = \cos(-2jk\pi/n)$ și la fel pentru sinus
- Lema 3, împreună cu Lema 2, implică faptul că a doua jumătate a dezvoltării trigonometrice (16) este redundantă
- putem interpola în punctele t_j folosind doar prima jumătate a termenilor (cu excepția unei schimbări a semnului pentru termenii cu sinus)
- din Lema 2, coeficienții din a doua jumătate a dezvoltării sunt aceiași ca cei din prima jumătate (cu excepția unei schimbări a semnului nentru termenii cu sinus)
- prin urmare, schimbările de semn se anulează reciproc, și am arătat că versiunea simplificată a lui P_n este

$$P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi(t-c)}{d-c} - b_k \sin \frac{2k\pi(t-c)}{d-c} \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \frac{n\pi(t-c)}{d-c}.$$

- pentru a scrie această expresie, am presupus că n este par
- formula este ușor diferită pentru n impar

Corolarul 1

Pentru un întreg par n , fie $t_j = c + j(d-c)/n$ pentru $j = 0, \dots, n-1$, și fie $x = [x_0, \dots, x_{n-1}]^T$ un vector de n numere reale. Definim $[a_0, \dots, a_{n-1}]^T + [b_0, \dots, b_{n-1}]^T i = F_n x$, unde F_n este Transformata Fourier Discretă. Atunci funcția

$$P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi(t-c)}{d-c} - b_k \sin \frac{2k\pi(t-c)}{d-c} \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \frac{n\pi(t-c)}{d-c} \quad (18)$$

satisfacă $P_n(t_j) = x_j$ pentru $j = 0, \dots, n-1$.

Exemplul 2

- găsiți interpolantul trigonometric pentru Exemplul 1
- intervalul este $[c, d] = [0, 1]$
- fie $x = [1, 0, -1, 0]^T$ și îl calculăm TFD ca fiind $y = [0, 1, 0, 1]^T$
- coeficienții de interpolare sunt $a_k + ib_k = y_k$
- prin urmare, $a_0 = a_2 = 0$, $a_1 = a_3 = 1$, și $b_0 = b_1 = b_2 = b_3 = 0$
- potrivit cu (18), avem nevoie doar de a_0, a_1, a_2 , și b_1
- o funcție de interpolare trigonometrică pentru x este dată prin

$$P_4(t) = \frac{a_0}{2} + (a_1 \cos 2\pi t - b_1 \sin 2\pi t) + \frac{a_2}{2} \cos 4\pi t = \cos 2\pi t.$$

- interpolarea punctelor (t, x) , unde $t = [0, 1/4, 1/2, 3/4]^T$ și $x = [1, 0, -1, 0]^T$, este prezentată în Figura 5

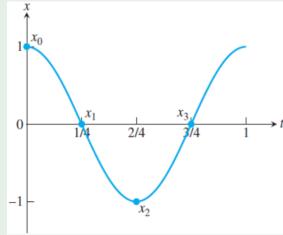


Figura 5: Interpolarea trigonometrică. Vectorul de intrare x este $[1, 0, -1, 0]^T$. Formula (18) ne dă funcția de interpolare ca fiind $P_4(t) = \cos 2\pi t$.

Exemplul 3

- găsiți interpolantul trigonometric pentru datele de temperatură: $x = [-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, -1.1]^T$ pe intervalul $[0, 1]$
- ieșirea transformatei Fourier, cu o precizie de patru zecimale exacte, este

$$y = \begin{bmatrix} -5.5154 \\ -1.0528 + 3.6195i \\ 1.5910 - 1.1667i \\ -0.5028 - 0.2695i \\ -0.7778 \\ -0.5028 + 0.2695i \\ 1.5910 + 1.1667i \\ -1.0528 - 3.6195i \end{bmatrix} \rightarrow \text{TFB ca la urm}$$

potrivit formulei (18), funcția de interpolare este

$$\begin{aligned} P_8(t) &= \frac{-5.5154}{\sqrt{8}} - \frac{1.0528}{\sqrt{2}} \cos 2\pi t - \frac{3.6195}{\sqrt{2}} \sin 2\pi t \\ &\quad + \frac{1.5910}{\sqrt{2}} \cos 4\pi t + \frac{1.1667}{\sqrt{2}} \sin 4\pi t \\ &\quad - \frac{0.5028}{\sqrt{2}} \cos 6\pi t + \frac{0.2695}{\sqrt{2}} \sin 6\pi t \\ &\quad - \frac{0.7778}{\sqrt{8}} \cos 8\pi t \\ &= -1.95 - 0.7445 \cos 2\pi t - 2.5594 \sin 2\pi t \\ &\quad + 1.125 \cos 4\pi t + 0.825 \sin 4\pi t \\ &\quad - 0.3555 \cos 6\pi t + 0.1906 \sin 6\pi t \\ &\quad - 0.2750 \cos 8\pi t. \end{aligned} \quad (19)$$

Figura 6 de mai jos prezintă punctele și funcția trigonometrică de interpolare

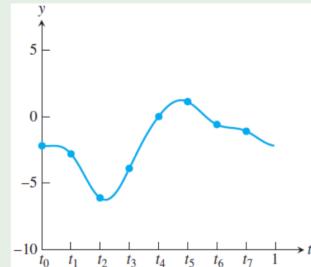


Figura 6: Interpolarea trigonometrică a datelor din Exemplul 3. Datele $t = [0, 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8]^T$, $x = [-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, 1.1]^T$ sunt interpolate folosind transformata Fourier cu $n = 8$. Graficul este făcut cu $p = 100$.

8.2.2. Evaluarea eficiență a funcțiilor trigonometrice

- pentru a privi coeficienții lui $P_n(t)$ ca fiind coeficienții unui polinom trigonometric de ordinul p , observăm că putem scrie (18) sub forma

$$P_p(t) = \frac{\sqrt{\frac{p}{n}} a_0}{\sqrt{p}} + \frac{2}{\sqrt{p}} \sum_{k=1}^{p/2-1} \left(\sqrt{\frac{p}{n}} a_k \cos \frac{2k\pi(t-c)}{d-c} - \sqrt{\frac{p}{n}} b_k \sin \frac{2k\pi(t-c)}{d-c} \right) + \frac{\sqrt{\frac{p}{n}} a_{p/2}}{\sqrt{p}} \cos n\pi t, \quad (20)$$

unde luăm $a_k = b_k = 0$ pentru $k = \frac{n}{2} + 1, \dots, \frac{p}{2}$

- concluzionăm din (20) că modul de a produce p puncte aflate pe curba (18) în $t_j = c + j(d-c)/n$ pentru $j = 0, \dots, n-1$ este de a înmulți coeficienții Fourier cu $\sqrt{p/n}$ și apoi de a inversa TFD
- un caz suficient de simplu și de folositor este cel pentru care $c = 0$, $d = n$
- punctele x_j sunt colectate în nodurile întregi de interpolare $s_j = j$ pentru $j = 0, \dots, n-1$
- punctele (j, x_j) sunt interpolate de către funcția trigonometrică

$$P_n(s) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi}{n} s - b_k \sin \frac{2k\pi}{n} s \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \frac{n\pi}{n} s. \quad (21)$$

- în Capitolul 9, vom folosi doar noduri de interpolare întregi, pentru compatibilitatea cu convențiile obișnuite pentru algoritmii de compresie a datelor audio și video

8.3. TFR si procesarea semnalelor

8.3.1. Ortogonalitatea si interpolarea

- rezultatul de interpolare foarte simplu din Teorema 2 a fost făcut posibil de faptul că $F_n^{-1} = \overline{F_n}^T$, ceea ce face ca F_n să fie o matrice unitară
- am întâlnit versiunea reală a acestei definiții în Capitolul 5, unde am numit o matrice U ortogonală dacă $U^{-1} = U^T$
- vom studia acum o formă specială de matrice ortogonală care se va traduce imediat într-un bun interpolant

Teorema 3 (Teorema interpolării prin funcții ortogonale)

Fie $f_0(t), \dots, f_{n-1}(t)$ funcții de t și t_0, \dots, t_{n-1} numere reale.

Presupunem că matricea $n \times n$

$$A = \begin{bmatrix} f_0(t_0) & f_0(t_1) & \cdots & f_0(t_{n-1}) \\ f_1(t_0) & f_1(t_1) & \cdots & f_1(t_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{n-1}(t_0) & f_{n-1}(t_1) & \cdots & f_{n-1}(t_{n-1}) \end{bmatrix} \quad (22)$$

este o matrice reală $n \times n$ ortogonală. Dacă $y = Ax$, funcția

$$F(t) = \sum_{k=0}^{n-1} y_k f_k(t)$$

interpolează $(t_0, x_0), \dots, (t_{n-1}, x_{n-1})$, și anume $F(t_j) = x_j$ pentru $j = 0, \dots, n - 1$.

- faptul că $y = Ax$ implică

$$x = A^{-1}y = A^T y,$$

și rezultă că

$$x_j = \sum_{k=0}^{n-1} a_{kj} y_k = \sum_{k=0}^{n-1} y_k f_k(t_j),$$

pentru $j = 0, \dots, n - 1$, ceea ce încheie demonstrația

Exemplul 4

- fie $[c, d]$ un interval și fie n un întreg pozitiv par
- arătați că ipotezele din Teorema 3 sunt satisfăcute pentru $t_j = c + j(d - c)/n$, $j = 0, \dots, n - 1$, și

matricea este

$$A = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ 1 & \cos \frac{2\pi}{n} & \cdots & \cos \frac{2\pi(n-1)}{n} \\ 0 & \sin \frac{2\pi}{n} & \cdots & \sin \frac{2\pi(n-1)}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \cos \pi & \cdots & \frac{1}{\sqrt{2}} \cos \pi(n-1) \end{bmatrix}. \quad (23)$$

Lema 4 arată că rândurile lui A sunt ortogonale două câte două

Lema 4

Fie $n \geq 1$ și k, l numere întregi. Atunci

$$\sum_{j=0}^{n-1} \cos \frac{2\pi j k}{n} \cos \frac{2\pi j l}{n} = \begin{cases} n & \text{dacă atât } (k-l)/n \text{ cât și } (k+l)/n \text{ sunt întregi} \\ \frac{n}{2} & \text{dacă exact unul dintre } (k-l)/n \text{ și } (k+l)/n \text{ este întreg} \\ 0 & \text{dacă niciunul nu este întreg} \end{cases}$$

$$\sum_{j=0}^{n-1} \cos \frac{2\pi j k}{n} \sin \frac{2\pi j l}{n} = 0$$

$$\sum_{j=0}^{n-1} \sin \frac{2\pi j k}{n} \sin \frac{2\pi j l}{n} = \begin{cases} 0 & \text{dacă atât } (k-l)/n \text{ cât și } (k+l)/n \text{ sunt întregi} \\ \frac{n}{2} & \text{dacă } (k-l)/n \text{ este întreg și } (k+l)/n \text{ nu este} \\ -\frac{n}{2} & \text{dacă } (k+l)/n \text{ este întreg și } (k-l)/n \text{ nu este} \\ 0 & \text{dacă niciunul nu este întreg.} \end{cases}$$

$$\begin{aligned} f_0(t) &= \sqrt{\frac{1}{n}} \\ f_1(t) &= \sqrt{\frac{2}{n}} \cos \frac{2\pi(t-c)}{d-c} \\ f_2(t) &= \sqrt{\frac{2}{n}} \sin \frac{2\pi(t-c)}{d-c} \\ f_3(t) &= \sqrt{\frac{2}{n}} \cos \frac{4\pi(t-c)}{d-c} \\ f_4(t) &= \sqrt{\frac{2}{n}} \sin \frac{4\pi(t-c)}{d-c} \\ &\vdots \\ f_{n-1}(t) &= \sqrt{\frac{1}{n}} \cos \frac{n\pi(t-c)}{d-c}. \end{aligned}$$

- Teorema 3 ne dă imediat funcția de interpolare

$$\begin{aligned}
 F(t) = & \sqrt{\frac{1}{n}} y_0 \\
 & + \sqrt{\frac{2}{n}} y_1 \cos \frac{2\pi(t-c)}{d-c} + \sqrt{\frac{2}{n}} y_2 \sin \frac{2\pi(t-c)}{d-c} \\
 & + \sqrt{\frac{2}{n}} y_3 \cos \frac{4\pi(t-c)}{d-c} + \sqrt{\frac{2}{n}} y_4 \sin \frac{4\pi(t-c)}{d-c} \\
 & \vdots \\
 & + \sqrt{\frac{1}{n}} y_{n-1} \cos \frac{n\pi(t-c)}{d-c}, \tag{24}
 \end{aligned}$$

pentru punctele (t_j, x_j) , în acord cu (18)

Exemplul 5

- folosiți funcțiile de bază din Exemplul 4 pentru a interpola punctele $x = [-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, -1.1]^T$ din Exemplul 3
- calculând produsul matricii $8 \times 8 A$ cu x , obținem

$$Ax = \sqrt{\frac{2}{8}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \\ 1 & \cos 2\pi \frac{1}{8} & \cos 2\pi \frac{2}{8} & \dots & \cos 2\pi \frac{7}{8} \\ 0 & \sin 2\pi \frac{1}{8} & \sin 2\pi \frac{2}{8} & \dots & \sin 2\pi \frac{7}{8} \\ 1 & \cos 4\pi \frac{1}{8} & \cos 4\pi \frac{2}{8} & \dots & \cos 4\pi \frac{7}{8} \\ 0 & \sin 4\pi \frac{1}{8} & \sin 4\pi \frac{2}{8} & \dots & \sin 4\pi \frac{7}{8} \\ 1 & \cos 6\pi \frac{1}{8} & \cos 6\pi \frac{2}{8} & \dots & \cos 6\pi \frac{7}{8} \\ 0 & \sin 6\pi \frac{1}{8} & \sin 6\pi \frac{2}{8} & \dots & \sin 6\pi \frac{7}{8} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \cos \pi & \frac{1}{\sqrt{2}} \cos 2\pi & \dots & \frac{1}{\sqrt{2}} \cos 7\pi \end{bmatrix} \begin{bmatrix} -2.2 \\ -2.8 \\ -6.1 \\ -3.9 \\ 0.0 \\ 1.1 \\ -0.6 \\ -1.1 \end{bmatrix} = \begin{bmatrix} -5.5154 \\ -1.4889 \\ -5.1188 \\ 2.2500 \\ 1.6500 \\ -0.7111 \\ 0.3812 \\ -0.7778 \end{bmatrix}$$

- formula (24) ne dă funcția de interpolare,

$$\begin{aligned}
 P(t) = & -1.95 - 0.7445 \cos 2\pi t - 2.5594 \sin 2\pi t \\
 & + 1.125 \cos 4\pi t + 0.825 \sin 4\pi t \\
 & - 0.3555 \cos 6\pi t + 0.1906 \sin 6\pi t \\
 & - 0.2750 \cos 8\pi t,
 \end{aligned}$$

în acord cu Exemplul 3

8.3.2. Cele mai mici patrate în funcții trigonometrice

- Corolarul 1 ne-a arătat cum TFD face ușoară interpolarea a n puncte egal depărtate din $[0, 1]$ cu o funcție trigonometrică de forma

$$P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} (a_k \cos 2k\pi t - b_k \sin 2k\pi t) + \frac{a_{n/2}}{\sqrt{n}} \cos n\pi t. \tag{25}$$

- observăm că numărul de termeni este n , egal cu numărul de puncte
- ca de obicei în acest capitol, presupunem că n este par
- cu cât există mai multe puncte, cu atât mai mulți cosinuși și sinuși sunt adăugați pentru a ajuta la interpolare
- după cum am descoperit în Capitolul 4, când numărul de puncte n este mare, este mai puțin comun să interpolăm un model exact
- de fapt, o aplicație comună a unui model este de a ignora anumite detalii (compresie cu pierdere de informații) cu scopul de a simplifica lucrurile
- un al doilea motiv pentru a nu folosi interpolarea exactă, discutat în Capitolul 5, este cazul în care punctele însăși sunt presupuse a fi inexacte, astfel încât constrângerea riguroasă a unei funcții de interpolare este nepotrivită

- în oricare dintre aceste situații, suntem motivați să facem o interpolare de tip cele mai mici pătrate folosind o funcție de tipul (25)
- deoarece coeficienții a_k și b_k apar liniar în model, putem continua cu același program descris în Capitolul 5, folosind rezolvarea ecuațiilor normale pentru a găsi cei mai buni coeficienți
- când încercăm acest lucru, găsim un rezultat surprinzător, care ne va duce înapoi la TFD
- ne întoarcem la Teorema 3
- fie n numărul de puncte x_j , pe care le gândim ca apărând la momente de timp egal depărtate $t_j = j/n$ în $[0, 1]$, pentru simplitate
- vom introduce numărul întreg pozitiv par m ca fiind numărul de funcții de bază pe care le vom folosi pentru interpolarea de tip cele mai mici pătrate
- și anume, vom interpola folosind primele m funcții de bază, $f_0(t), \dots, f_{m-1}(t)$
- funcția folosită pentru a interpola cele n puncte va fi

$$P_m(t) = \sum_{k=0}^{m-1} c_k f_k(t), \quad (26)$$

unde coeficienții c_k vor trebui determinați

- când $m = n$, problema este tot o interpolare
- când $m < n$, problema este o problemă de compresie
- în acest caz, ne așteptăm să interpolăm punctele folosind P_m cu eroarea pătratică minimă
- problema de tip cele mai mici pătrate este de a găsi coeficienții c_0, \dots, c_{m-1} astfel încât egalitatea

$$\sum_{k=0}^{m-1} c_k f_k(t_j) = x_j$$

este satisfăcută cu o eroare cât mai mică posibil

- în termeni de matrici,

$$A_m^T c = x, \quad (27)$$

unde A_m este matricea primelor m rânduri ale lui A

- în ipotezele Teoremei 3, A_m^T are coloanele ortonormale două câte două
- când formăm ecuațiile normale

$$A_m A_m^T c = A_m x$$

pentru a găsi c , $A_m A_m^T$ este matricea identitate

- prin urmare, soluția în sensul celor mai mici pătrate,

$$c = A_m x, \quad (28)$$

este ușor de calculat

- am demonstrat următorul rezultat folositor, care extinde Teorema 3:

Teorema 4 (Teorema aproximării de tip cele mai mici pătrate prin funcții ortogonale)

Fie $m \leq n$ numere întregi, și presupunem că punctele $(t_0, x_0), \dots, (t_{n-1}, x_{n-1})$ sunt date. Luăm $y = Ax$, unde A este o matrice ortogonală de forma (22). Atunci polinomul de interpolare pentru funcțiile de bază $f_0(t), \dots, f_{n-1}(t)$ este

$$F_n(t) = \sum_{k=0}^{n-1} y_k f_k(t), \quad (29)$$

și cea mai bună aproximare de tip cele mai mici pătrate, folosind doar funcțiile f_0, \dots, f_{m-1} , este

$$F_m(t) = \sum_{k=0}^{m-1} y_k f_k(t). \quad (30)$$

- acesta este un rezultat frumos și folositor
- ne spune că, fiind date n puncte, pentru a găsi cea mai bună funcție trigonometrică în sensul celor mai mici pătrate cu $m < n$ termeni care interpolează datele, este suficient să calculăm interpolantul adevărat cu n termeni și să păstrăm doar primii m termeni doriti
- cu alte cuvinte, coeficienții de interpolare Ax pentru x se degradează cât de puțin posibil când se renunță la termenii cu frecvențele cele mai înalte
- păstrând primii m termeni din dezvoltarea cu n termeni garantează cea mai bună interpolare posibilă cu cei m termeni de frecvențe joase
- această proprietate reflectă „ortogonalitatea” funcțiilor de bază
- raționamentul care precedă Teorema 4 poate fi ușor adaptat pentru a demonstra un rezultat mai general
- am arătat cum să găsim soluția în sensul celor mai mici pătrate pentru primele m funcții de bază, dar, în realitate, ordinul nu este relevant; am fi putut specifica orice submulțime de funcții de bază
- soluția în sensul celor mai mici pătrate este găsită pur și simplu prin renunțarea la toți acei termeni din (29) care nu sunt inclusi în submulțime
- versiunea (30) este un filtru „trece-jos”, presupunând că funcțiile cu indicele mai mic au „frecvențele” cele mai joase; dar prin schimbarea submulțimii de funcții de bază care sunt păstrate, putem trece orice frecvențe de interes prin simpla renunțare la coeficienții nedoriți
- ne întoarcem acum la polinomul trigonometric (25) și demonstrăm cum să interpolăm o versiune de ordinul m pentru cele n puncte, unde $m < n$
- funcțiile de bază folosite sunt funcțiile din Exemplul 4, care satisfac ipotezele Teoremei 3
- Teorema 4 arată că, oricare ar fi coeficienții de interpolare, coeficienții celei mai bune aproximări în sensul celor mai mici pătrate de ordinul m sunt găsiți prin renunțarea la termenii de ordin mai mare decât m
- am dedus următoarea aplicație:

Corolarul 2

Fie $[c, d]$ un interval, fie $m < n$ întregi pozitivi pari, $x = [x_0, \dots, x_{n-1}]^T$ un vector de n numere reale, și fie $t_j = c + j(d - c)/n$ pentru $j = 0, \dots, n - 1$. Fie $\{a_0, a_1, b_1, a_2, b_2, \dots, a_{n/2-1}, b_{n/2-1}, a_{n/2}\} = F_n x$ coeficienții de interpolare pentru x astfel încât

$$x_j = P_n(t_j) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi(t_j - c)}{d - c} - b_k \sin \frac{2k\pi(t_j - c)}{d - c} \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \frac{n\pi(t_j - c)}{d - c}$$

pentru $j = 0, \dots, n - 1$. Atunci

$$P_m(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{m/2-1} \left(a_k \cos \frac{2k\pi(t - c)}{d - c} - b_k \sin \frac{2k\pi(t - c)}{d - c} \right) + \frac{2a_{m/2}}{\sqrt{n}} \cos \frac{m\pi(t - c)}{d - c}$$

este cea mai bună aproximare de tip cele mai mici pătrate de ordinul m pentru punctele (t_j, x_j) , pentru $j = 0, \dots, n - 1$.

- o altă modalitate de a aprecia puterea Teoremei 4 este de a o compara cu funcțiile de bază monomiale pe care le-am folosit anterior pentru modele de tip cele mai mici pătrate
- cea mai bună parabolă care interpolează punctele $(0, 3), (1, 3), (2, 5)$ este $y = x^2 - x + 3$
- cu alte cuvinte, cei mai buni coeficienți pentru modelul $y = a + bx + cx^2$ pentru aceste date sunt $a = 3, b = -1$, și $c = 1$ (în acest caz, deoarece eroarea pătratică este zero—aceasta este parabola de interpolare)
- acum să folosim interpolarea pentru o submulțime a funcțiilor de bază—de exemplu, să schimbăm modelul în $y = a + bx$
- calculăm cea mai bună dreaptă care interpolează punctele ca fiind $a = 8/3, b = 1$
- observăm că, pentru interpolarea de gradul 1, coeficienții nu au nicio relație aparentă cu coeficienții corespunzători pentru interpolarea de gradul 2
- aceasta este exact ceea ce nu se întâmplă pentru funcțiile de bază trigonometrice
- o interpolare exactă, sau orice interpolare de tip cele mai mici pătrate de forma (26), conține explicit toate informațiile despre interpolările de tip cele mai mici pătrate de ordin inferior
- datorită răspunsului extrem de simplu pe care TFD îl are pentru cele mai mici pătrate, este foarte simplu să scriem un program care efectuează acești pași
- fie $m < n < p$ numere întregi, unde n este numărul de puncte, m este ordinul modelului trigonometric de tip cele mai mici pătrate, și p guvernează rezoluția reprezentării grafice a celui mai bun model
- putem să ne gândim că cele mai mici pătrate „filtrează” contribuțiile cu cele mai înalte frecvențe ale interpolantului de ordinul n și păstrează doar contribuțiile cu cele mai joase m frecvențe

Exemplul 6

- interpolați datele de temperatură din Exemplul 3 folosind funcții trigonometrice de tip cele mai mici pătrate de ordinul 4 și de ordinul 6
- ideea Corolarului 2 este că putem pur și simplu să interpolăm punctele aplicând F_n și apoi să renunțăm la termeni după cum dorim pentru a obține interpolările de tip cele mai mici pătrate de ordinele inferioare
- rezultatul din Exemplul 3 a fost că

$$\begin{aligned} P_8(t) = & 1.95 - 0.7445 \cos 2\pi t - 2.5594 \sin 2\pi t \\ & + 1.125 \cos 4\pi t + 0.825 \sin 4\pi t \\ & - 0.3555 \cos 6\pi t + 0.1906 \sin 6\pi t \\ & - 0.2750 \cos 8\pi t. \end{aligned} \quad (31)$$

- prin urmare, modelele de tip cele mai mici pătrate de ordinele 4 și 6 sunt

$$\begin{aligned} P_4(t) = & -1.95 - 0.7445 \cos 2\pi t - 2.5594 \sin 2\pi t + 1.125 \cos 4\pi t \\ P_6(t) = & -1.95 - 0.7445 \cos 2\pi t - 2.5594 \sin 2\pi t \\ & + 1.125 \cos 4\pi t + 0.825 \sin 4\pi t - 0.3555 \cos 6\pi t \end{aligned}$$

- interpolările de tip cele mai mici pătrate sunt prezentate în Figura 7

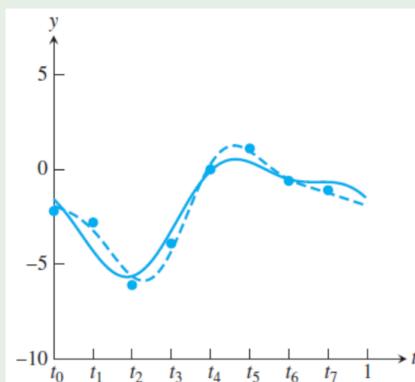


Figura 7: Interpolări trigonometrice de tip cele mai mici pătrate pentru Exemplul 6. Interpolările pentru $m = 4$ (curba continuă) și 6 (curba întreruptă) sunt prezentate. Vectorul de intrare x este $[-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, 1.1]^T$. Interpolarea pentru $m = 8$ este interpolarea trigonometrică, prezentată în Figura 6.