www.math-stockholm.se/cirkel

11 mars 2021

Välkomna!

Idag:

- Repetition av den diskreta fouriertransformen (DFT)
- Sats och bevis (teoretiskt)
- Gå igenom en snabbare algorithm (praktiskt och teoretiskt)
- Gå igenom hur man programmera den snabbare algoritm (praktiskt)

Kom ihåg: föreläsning 4

Nytt mål

Beräkna y_j, $j = 0, \ldots, n-1$, där $\omega = e^{-i2\pi/n}$, x_i kända för $i = 0, \ldots, n-1$ och

$$y_0 + y_1\omega^{-j} + y_2\omega^{-2j} + \ldots + y_{n-1}\omega^{-j(n-1)} = x_j.$$

Skriva om!

$x \in \mathbb{R}^n$ och $X = \begin{pmatrix} e^0 & e^{i2\pi/n} & e^{i4\pi/n} & \ldots & e^{i2(n-1)\pi/n} \end{pmatrix}^T$

$$\frac{1}{\sqrt{n}} \begin{pmatrix} | & | & | & & | \\ X^0 & X^1 & X^2 & \ldots & X^{n-1} \\ | & | & | & & | \end{pmatrix} \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

Kom ihåg: föreläsning 4

Vi måste lösa $B_n y = x$ för y

$$\underbrace{\frac{1}{\sqrt{n}}\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(n-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \dots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \dots & \omega^{-(n-1)^2} \end{pmatrix}}_{=:B_n} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_{n-1} \end{pmatrix}$$

där $\omega = e^{-i2\pi/n}$

- Gausselimination
- Men vi kan lösa det billigare . . .

Kom ihåg: föreläsning 4

Vi har då: $y = F_n x$, där $F_n B_n = B_n F_n = I$ och $F_n = \bar{B}_n$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_{n-1} \end{pmatrix} = \frac{1}{\sqrt{n}} \underbrace{\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \dots & \omega^{(n-1)} \\ \omega^0 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{(n-1)} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{pmatrix}}_{=:F_n} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_{n-1} \end{pmatrix}$$

där $\omega = e^{-i2\pi/n}$

- Multiplicera en matris med en vektor: $\mathcal{O}(n^2)$ (billigare)
- Bättre än Gausselimination: $\mathcal{O}(n^3)$

Då har vi:

Definition: (Diskret fouriertransform)

Den diskreta fouriertransformen (DFT) av en vektor $x = [x_0, x_1, \ldots, x_{n-1}]^T \in \mathbb{R}^n$ är vektorn $y = [y_0, y_1, \ldots, y_{n-1}]^T \in \mathbb{C}^n$ där $\omega = e^{-i2\pi/n}$ och

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

Notera:

- Det här betyder att vi multiplicera en matris med en vektor
- Det är bara ett annat sätt att skriva

Diskreta fouriertransformen

Exempel

- Beräkna den diskreta fouriertransformen av $x = [0, 1, 0, -1]^T$
- $x \in \mathbb{R}^4$, $n = 4$
- Då har vi att: $\omega = e^{-i2\pi/4} = e^{-i\pi/2} = -i$ och

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \underbrace{\frac{1}{\sqrt{4}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix}}_{=:F_4} \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix},$$

- dvs $y = F_4 x$ där y är DFT:n av x

Diskreta fouriertransformen

Mål: beräkna $y \in \mathbb{C}^4$ som är den diskreta fouriertransformen

$\omega = e^{-i2\pi/4} = e^{-i\pi/2} = -i$ och $\omega^k = (e^{-i2\pi/4})^k = (e^{-i2k\pi/4})$

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \frac{1}{\sqrt{4}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}$$

eftersom $(-i)^2 = i^2 = -1$ och $(-i)^3 = (-i^2)i = -i$.

Diskreta fouriertransformen (3)

Mål: beräkna $y \in \mathbb{C}^4$ som är den diskreta fouriertransformen

Då får vi svaret $y \in \mathbb{C}^4$:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ -i \\ 0 \\ i \end{pmatrix}.$$

Inversa diskret fouriertransformen (1)

Definition: (Invers diskret fouriertransform)

Den *inversa diskret fouriertransformen* av en vektor y är vektorn x så att $x = F_n^{-1} y = B_n y$.

Exempel

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\frac{1}{\sqrt{4}} \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \omega^{-3} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \omega^{-6} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \omega^{-9} \end{pmatrix}}_{=:B_4} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

där $\omega = e^{-i2\pi/4} = e^{-i\pi/2} = -i$

Inversa diskret fouriertransformen (2)

$\omega = e^{-i2\pi/4} = e^{-i\pi/2} = -i$ och $\omega^{-k} = (e^{-i2\pi/4})^{-k} = (e^{i2k\pi/4})$

Exempel

Vi kan verifiera:

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \frac{1}{\sqrt{4}} \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \omega^{-3} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \omega^{-6} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \omega^{-9} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 0 \\ -i \\ 0 \\ i \end{pmatrix}$$

Så får vi att $x = \begin{pmatrix} 0 & 1 & 0 & -1 \end{pmatrix}^T$

Teori

Mål

Bevisa att vi kan skriva diskreta fouriertransformen av en vektor (reella tal) med dimension n med exakt n reella tal.

Som vi såg innan: DFT:n av $x = \begin{pmatrix} 0 & 1 & 0 & -1 \end{pmatrix}^T \in \mathbb{R}^4$ är

$$y = \begin{pmatrix} a_0 + b_0 i \\ a_1 + b_1 i \\ a_2 + b_2 i \\ a_3 + b_3 i \end{pmatrix} = \begin{pmatrix} 0 \\ -i \\ 0 \\ i \end{pmatrix} = \begin{pmatrix} 0 + (0)i \\ 0 - 1(i) \\ 0 + (0)i \\ 0 + (1)i \end{pmatrix} \in \mathbb{C}^4$$

- 8 reella tal ($a_i, b_i \in \mathbb{R}, i = 0, 1, 2, 3$)
- Ska **bevisa** att vi behöver bara 4 (dvs: a_0, a_1, a_2, b_1)

Teori

Mål

Bevisa att vi kan skriva diskreta fouriertransformen av en vektor (reella tal) med dimension n med exakt n reella tal.

Plan:

Vi behöver en sats

- Först: En hjälpsats (med bevis)
- Sedan: En sats (med bevis)
- Därefter: använda resultatet från satsen

Teori

Hjälpsats

Det komplexa konjugatet av produkten är produkten av de komplexa konjugaten, dvs $\overline{(a+ib)(c+id)} = (\overline{a+ib})(\overline{c+id})$.

Bevis

Låt $a, b, c, d \in \mathbb{R}$. Då har vi följande ekvation.

$$\begin{aligned}\overline{(a+ib)(c+id)} &= \overline{ac - bd + i(ad+bc)} \\ &= ac - bd - i(ad+bc) \\ &= (a-ib)(c-id) \\ &= (\overline{a+ib})(\overline{c+id}).\end{aligned}$$

Teori

Sats

Anta att $\{y_k\}$ är den diskreta fouriertransformen av $\{x_k\}$, där x_j är reella tal. Då är y_0 ett reellt tal och $y_{n-k} = \overline{y}_k$, för $k = 1, \ldots, n-1$.

Mer konkret: $x \in \mathbb{R}^4$ och DFT:n av x är $y \in \mathbb{C}^4$ där

$$y = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} a_0 + b_0 i \\ a_1 + b_1 i \\ a_2 + b_2 i \\ a_3 + b_3 i \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 + b_1 i \\ a_2 \\ a_1 - b_1 i \end{pmatrix} = \begin{pmatrix} 0 + (0)i \\ 0 - 1(i) \\ 0 + (0)i \\ 0 + (1)i \end{pmatrix} = \begin{pmatrix} 0 \\ -i \\ 0 \\ i \end{pmatrix}.$$

- Behöver beräkna a_0, a_1, a_2 och b_1 men inte a_3, b_0, b_2 och b_3
- Vi måste **bevisa** det här

Teori

Sats

Anta att $\{y_k\}$ *är den diskreta fouriertransformen av* $\{x_k\}$*, där* x_j *är reella tal. Då är* y_0 *ett reellt tal och* $y_{n-k} = \overline{y}_k$*, för* $k = 1, \ldots, n-1$.

Notera:

- Vi ska börja bevisa satsen
- Kom ihåg definitionen av DFT:n

$$y = F_n x$$

$$\iff y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}$$

- Vi ska använda summan i beviset som följer

Bevis (1)

y_0 är reellt eftersom $y_0 = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j$ och

$$\begin{aligned}\omega^{n-k} &= e^{-i2\pi(n-k)/n} \\ &= e^{-i2\pi} e^{i2\pi k/n} \\ &= 1 \cdot e^{i2\pi k/n} \\ &= \cos(2\pi k/n) + i \sin(2\pi k/n)\end{aligned}$$

och

$$\begin{aligned}\omega^{k} &= e^{-i2\pi k/n} \\ &= \cos(2\pi k/n) - i \sin(2\pi k/n).\end{aligned}$$

Alltså gäller att $\omega^{n-k} = \overline{\omega^k}$.

Bevis (2)

Vi har också:

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

som är vad vi ville bevisa.

Teori

Exempel

Anta att $x = [x_0, x_1, \ldots, x_7]^T \in \mathbb{R}^8$. Då har DFT:n y av x följande form.

$$y = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} a_0 + ib_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ a_3 + ib_3 \\ a_4 + ib_4 \\ a_5 + ib_5 \\ a_6 + ib_6 \\ a_7 + ib_7 \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ a_3 + ib_3 \\ a_4 \\ a_3 - ib_3 \\ a_2 - ib_2 \\ a_1 - ib_1 \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_{\frac{n}{2}-1} \\ y_{\frac{n}{2}} \\ \overline{y}_{\frac{n}{2}-1} \\ \vdots \\ \overline{y}_1 \end{pmatrix},$$

där $a_i, b_i \in \mathbb{R}$ for $i = 0, \ldots, 7$. **Så kan vi skriva DFT av en vektor med dimension n med exakt n reella tal.**

FFT: Snabb fouriertransform

- En algoritm som kan beräkna y som är DFT:n av x ännu snabbare
- Algoritmen heter *Snabb fouriertransform* (FFT)
- Vi går igenom Cooley-Tukey FFT sätt att beräkna
- Kom ihåg: $\omega = e^{-i2\pi/n}$ och

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_{n-1} \end{pmatrix} = \frac{1}{\sqrt{n}} \underbrace{\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \dots & \omega^{(n-1)} \\ \omega^0 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{(n-1)} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{pmatrix}}_{=:M_n} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_{n-1} \end{pmatrix}$$

FFT: Snabb fouriertransform

Steg 1: Kom ihåg F_n

- F_4, dvs $y = F_4 x$
- Tar för tillfället bort faktorn $1/2$

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \underbrace{\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix}}_{=:M_4} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

- Vi kan skriva $z = M_4 x$ där $z \in \mathbb{C}^4$
- Mer generellt har vi $z = M_n x$

FFT: Snabb fouriertransform

Steg 2: Skriv om produkten

Skriv om varje rad i produkten så att:

- vi först skriver termerna med **jämna** index och
- därefter de med **udda** index, dvs

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$= \begin{pmatrix} \omega^0 x_0 + \omega^0 x_2 + \omega^0 x_1 + \omega^0 x_3 \\ \omega^0 x_0 + \omega^2 x_2 + \omega^1 x_1 + \omega^3 x_3 \\ \omega^0 x_0 + \omega^4 x_2 + \omega^2 x_1 + \omega^6 x_3 \\ \omega^0 x_0 + \omega^6 x_2 + \omega^3 x_1 + \omega^9 x_3 \end{pmatrix}$$

FFT: Snabb fouriertransform

Steg 3: Faktorisera

På rad i faktorisera ut ω^i från termer x_i med **udda** index, dvs

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \omega^0 x_0 + \omega^0 x_2 + \omega^0 x_1 + \omega^0 x_3 \\ \omega^0 x_0 + \omega^2 x_2 + \omega^1 x_1 + \omega^3 x_3 \\ \omega^0 x_0 + \omega^4 x_2 + \omega^2 x_1 + \omega^6 x_3 \\ \omega^0 x_0 + \omega^6 x_2 + \omega^3 x_1 + \omega^9 x_3 \end{pmatrix}$$

$$= \begin{pmatrix} \omega^0 x_0 + \omega^0 x_2 + \omega^0(\omega^0 x_1 + \omega^0 x_3) \\ \omega^0 x_0 + \omega^2 x_2 + \omega^1(\omega^0 x_1 + \omega^2 x_3) \\ \omega^0 x_0 + \omega^4 x_2 + \omega^2(\omega^0 x_1 + \omega^4 x_3) \\ \omega^0 x_0 + \omega^6 x_2 + \omega^3(\omega^0 x_1 + \omega^6 x_3) \end{pmatrix}$$

Notera: $(\omega^j)(\omega^k) = \omega^{j+k}$

FFT: Snabb fouriertransform

Steg 4: Använd att $\omega^4 = (e^{-i2\pi/4})^4 = e^{-i2\pi} = 1$

Skriv om som följande:

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

$$= \begin{pmatrix} \omega^0 x_0 + \omega^0 x_2 + \omega^0(\omega^0 x_1 + \omega^0 x_3) \\ \omega^0 x_0 + \omega^2 x_2 + \omega^1(\omega^0 x_1 + \omega^2 x_3) \\ \omega^0 x_0 + \omega^0 x_2 + \omega^2(\omega^0 x_1 + \omega^0 x_3) \\ \omega^0 x_0 + \omega^2 x_2 + \omega^3(\omega^0 x_1 + \omega^2 x_3) \end{pmatrix}$$

och notera att vi ser $M_2 := \begin{pmatrix} \omega^0 & \omega^0 \\ \omega^0 & \omega^2 \end{pmatrix}$

FFT: Snabb fouriertransform

Steg 5: Skapa två vektorer och skriva om

Skapa $u = [u_0, u_1]^T$ och $v = [v_0, v_1]^T$ sådant att:

$$\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} := \begin{pmatrix} \omega^0 & \omega^0 \\ \omega^0 & \omega^2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_2 \end{pmatrix} = M_2 \begin{pmatrix} x_0 \\ x_2 \end{pmatrix}$$
$$\begin{pmatrix} v_0 \\ v_1 \end{pmatrix} := \begin{pmatrix} \omega^0 & \omega^0 \\ \omega^0 & \omega^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = M_2 \begin{pmatrix} x_1 \\ x_3 \end{pmatrix}$$

Skriv om ekvationerna för att beräkna z_i med de nya vektorerna:

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \omega^0 x_0 + \omega^0 x_2 + \omega^0(\omega^0 x_1 + \omega^0 x_3) \\ \omega^0 x_0 + \omega^2 x_2 + \omega^1(\omega^0 x_1 + \omega^2 x_3) \\ \omega^0 x_0 + \omega^0 x_2 + \omega^2(\omega^0 x_1 + \omega^0 x_3) \\ \omega^0 x_0 + \omega^2 x_2 + \omega^3(\omega^0 x_1 + \omega^2 x_3) \end{pmatrix} = \begin{pmatrix} u_0 + \omega^0 v_0 \\ u_1 + \omega^1 v_1 \\ u_0 + \omega^2 v_0 \\ u_1 + \omega^3 v_1 \end{pmatrix}.$$

FFT: Snabb fouriertransform

Med nya variabler

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \omega^0 x_0 + \omega^0 x_2 + \omega^0(\omega^0 x_1 + \omega^0 x_3) \\ \omega^0 x_0 + \omega^2 x_2 + \omega^1(\omega^0 x_1 + \omega^2 x_3) \\ \omega^0 x_0 + \omega^0 x_2 + \omega^2(\omega^0 x_1 + \omega^0 x_3) \\ \omega^0 x_0 + \omega^2 x_2 + \omega^3(\omega^0 x_1 + \omega^2 x_3) \end{pmatrix} = \begin{pmatrix} u_0 + \omega^0 v_0 \\ u_1 + \omega^1 v_1 \\ u_0 + \omega^2 v_0 \\ u_1 + \omega^3 v_1 \end{pmatrix}$$

Steg 6:

Notera: y som är DFT:n av x är lika med $\frac{1}{2}z$, där z är vektorn som vi har precis beräknat

FFT: Snabb fouriertransform

Sammanfattning:

- Vi kan beräkna FFT:n av en vektor med dimension $n = 4$ genom att beräkna två DFT på två olika vektorer av dimension 2 ($\tilde{n} = 2$)
- Om dimensionen är större än 4:
  - Så länge $n = 2^j$, $j = 1, 2, ...$
  - Lös mindre delproblem om och om igen
  - När den nya dimensionen $\tilde{n} = 2$, beräkna DFT:n
- Att dela upp ett problem i ett antal mindre delproblem på detta sätt kallas för *rekursion*
- (Ofta lättare att förstå när man ser koden)
- FFT är mycket snabbare än att använda den vanliga DFT

FFT: Snabb fouriertransform

- Först: $\mathcal{O}(n^3)$ med Gausselimination
- Sedan: $\mathcal{O}(n^2)$ med att multiplicera en matris med en vektor
- Nu: FFT tar $\mathcal{O}(n \log_2 n)$ beräkningar
- Notera: Samma svar med alla 3 metoder
- Om n är litet, till exempel 3 eller 4, är det svårt att se hur stor betydelse detta har
- Ta istället $n = 2^{16} = 65536$ så får vi

$$\textbf{Gausselimination}: \; n^3 = 2^{48} \approx 2.8 \cdot 10^{14}$$

$$\textbf{Multiplicera}: \; n^2 = 2^{32} \approx 4 \cdot 10^9$$

$$\textbf{FFT}: \; n \log_2 n = 2^{16} \cdot 16 = 2^{20} \approx 10^6$$

Python kod

Koden

- Python kod finns i kompendium
- Helt okej om man inte kan programmera redan
  - Poängen är att ni kan använda koden här
  - (inte att skriva den själva)
- Innan nästa tillfälle: ladda ner Anaconda så att ni kan programmera i Python på era datorer (länk på hemsidan)
  - På övningar: vi ska gå igenom några enkla Python program
  - Bra om ni experimentar själva också
- Jag använder Spyder när jag programmera i Python
  - Integrated development environment (IDE)
  - Den gör det lätt att skriva program
- Leka lite med FFT kod i kompendium