Numerical Methods for Computational Science and Engineering

Fall Semester 2017 (HS17)

Prof. Rima Alaifari, SAM, ETH Zurich

Filtering of m-periodic signal

$$x_j = x_{j+m} \quad \forall j \in \mathbb{Z}$$

$$y_k = \sum_{j\in\mathbb{Z}} x_j h_{k-j}$$

$$= \sum_{j\in\mathbb{Z}} x_{k-j} h_j \qquad \text{(i.e. } y \text{ is also m-periodic)}$$

$$= \sum_{j=0}^{m-1} \underbrace{\Big(\sum_{l\in\mathbb{Z}} h_{j+lm}\Big)}_{\text{periodic summation}} x_{k-j}$$

$$p_j := \sum_{l\in\mathbb{Z}} h_{j+lm}$$

with impulse response $h = (\dots, 0, h_0, \dots, h_{n-1}, 0, \dots)$

By definition $p_{j+m} = p_j \quad \forall j \in \mathbb{Z}$

$\Rightarrow p$ is m-periodic

$$y_k = \sum_{j=0}^{m-1} p_j x_{k-j} = \sum_{j=0}^{m-1} p_{k-j} x_j \qquad k \in \mathbb{Z}$$

$$(y_k)_{k\in\mathbb{Z}} = (p_k)_{k\in\mathbb{Z}} *_m (x_k)_{k\in\mathbb{Z}}$$

discrete periodic convolution

Definition 4.1.33. Discrete periodic convolution

The discrete periodic convolution of two n-periodic sequences $(p_k)_{k\in\mathbb{Z}}$, $(x_k)_{k\in\mathbb{Z}}$ yields the n-periodic sequence

$$(y_k) := (p_k) *_n (x_k) \quad , \quad y_k := \sum_{j=0}^{n-1} p_{k-j}x_j = \sum_{j=0}^{n-1} x_{k-j}p_j \, , \quad k \in \mathbb{Z} \, .$$

$$\begin{bmatrix} y_0 \\ \vdots \\ \vdots \\ y_{m-1} \end{bmatrix} = \begin{bmatrix} p_0 & p_{m-1} & \cdots & p_1 \\ p_1 & p_0 & \cdots & p_2 \\ \vdots & p_1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & p_{m-1} \\ p_{m-1} & p_{m-2} & \cdots \; p_1 & p_0 \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ \vdots \\ x_{m-1} \end{bmatrix}$$

$$\begin{aligned} y_0 &= p_0 x_0 + p_{-1} x_1 + \dots + p_{-(m-1)} x_{m-1} \\ &= p_0 x_0 + p_{m-1} x_1 + \dots + p_1 x_{m-1} \\ y_1 &= p_1 x_0 + p_0 x_1 + \dots + p_{-(m-2)} x_{m-1} \\ & p_1 x_0 + p_0 x_1 + \dots + p_2 x_{m-1} \end{aligned}$$

Definition 4.1.38. Circulant matrix $\rightarrow$ [?, Sect. 54]

A matrix $\mathbf{C} = [c_{ij}]_{i,j=1}^{n} \in \mathbb{K}^{n,n}$ is circulant

$:\Leftrightarrow$ $\exists (p_k)_{k\in\mathbb{Z}}$ n-periodic sequence: $\underline{c_{ij} = p_{j-i}}$, $1 \le i,j \le n$.

$$\text{circul}(\mathbf{p}) = \begin{bmatrix} p_0 & p_1 & p_2 & \cdots & \cdots & p_{n-1} \\ p_{n-1} & p_0 & & & & p_{n-2} \\ p_{n-2} & & & & & \vdots \\ \vdots & & & & & \\ \vdots & & & & & \vdots \\ p_2 & & & & & p_1 \\ p_1 & p_2 & \cdots & \cdots & p_{n-1} & p_0 \end{bmatrix} \in \mathbb{K}^{n,n}$$

(Diagonal & all sub-diagonals are constant)

Circulant matrix is determined by a vector

$$\mathbf{p} = [p_0, \dots, p_{n-1}]^T$$

Discrete periodic convolution $\hat{=}$ Multiplication by a circulant matrix

Discrete convolution: reducable to discr. <u>periodic</u> convolution

How? Discrete convolution of $\underline{x} = [x_0, \dots, x_{m-1}]^T$
and $\underline{h} = [h_0, \dots, h_{n-1}]^T$

$$y_k = (\underline{x} * \underline{h})_k = \sum_{j=0}^{m-1} x_j h_{k-j} \qquad k = 0, \dots, m+n-2$$

y has length $m+n-1$

$\Rightarrow$ to "fit" output y in some periodic convolution: Need period of at least

$L := m+n-1$!

zero-pad $\underline{x}$ to length L:

$$\underline{x}^L = [x_0, \dots, x_{m-1}, 0, \dots, 0]^T \in \mathbb{R}^L$$

and take periodic extension $x^L \in \ell^\infty(\mathbb{Z})$

$$h = (\dots, 0, h_0, \dots, h_{n-1}, 0, \dots) \in \ell^\infty(\mathbb{Z})$$

Then, $p_j = \sum_{\ell \in \mathbb{Z}} h_{j+\ell L} = h_{j+mL}$ s.t. $j+mL \in \{0, \dots, n-1\}$

$$\underline{h}^L = [h_0, \dots, h_{n-1}, 0, \dots, 0]^T \in \mathbb{R}^L$$

$p \in \ell^\infty(\mathbb{Z})$: L-periodic extension of $\underline{h}^L$

$$y^L = x^L *_L p$$

$$y_k^L = \sum_{j=0}^{L-1} p_j x_{k-j}^L = \sum_{j=0}^{L-1} h_j x_{k-j}^L$$
$$= \sum_{j=0}^{L-1} h_{k-j} x_j^L = \sum_{j=0}^{m-1} h_{k-j} x_j = y_k$$

$\underline{a} = [a_0, \dots, a_{n-1}]^T$ $\qquad a^L \in \ell^\infty(\mathbb{Z})$

$L = 2n-1$

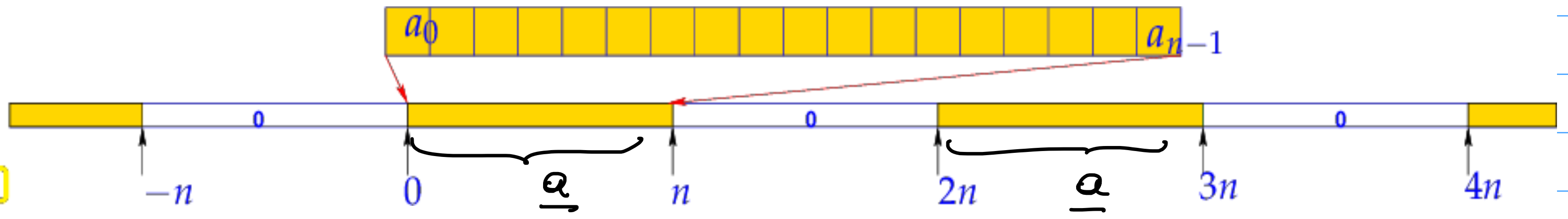


Fig. 139

a: signal of length n
zero-padded to length $2n-1$
and periodized

in matrix format:

$$\begin{bmatrix} y_0 \\ \vdots \\ \vdots \\ y_{L-1} \end{bmatrix} = \begin{bmatrix} p_0 & p_{L-1} & \cdots & p_1 \\ p_1 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \vdots \\ p_{L-1} & p_{L-2} & \cdots\ p_1 & p_0 \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$L = m+n-1$$

$\in \mathbb{R}^{L,m}$

$$\begin{bmatrix} y_0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ y_{L-1} \end{bmatrix} = \underbrace{\begin{bmatrix} h_0 & 0 & 0 & \cdots & 0 & h_{n-1} & h_{n-2} & \cdots & h_1 \\ h_1 & h_0 & 0 & & \vdots & 0 & 0 & & \vdots \\ \vdots & h_1 & h_0 & & \vdots & \vdots & \vdots & & h_{n-1} \\ h_{n-1} & \vdots & h_1 & & 0 & & & & 0 \\ 0 & h_{n-1} & \vdots & & h_0 & 0 & & & \vdots \\ \vdots & 0 & h_{n-1} & & \vdots & h_0 & 0 & & \vdots \\ \vdots & \vdots & 0 & & \vdots & & h_0 & & 0 \\ 0 & 0 & 0 & \cdots & h_{n-1} & h_{n-2} & h_{n-3} & \cdots & h_0 \end{bmatrix}}_{\in \mathbb{R}^{L,L}} \begin{bmatrix} x_0 \\ \vdots \\ x_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

circulant matrix

4.2. Discrete Fourier Transform (DFT)

<u>Eigenvalues & eigenvectors of a circulant matrix</u>

$$C v_k = \lambda_k v_k \qquad C \in \mathbb{C}^{n,n} \text{ circulant}$$

$$\omega_n := e^{-2\pi i/n}$$

$$\omega_n^n = e^{-2\pi i} = 1$$

$$\omega_n^{-\ell} = \overline{\omega_n}^{\ell} \qquad \text{(complex conjugation)}$$

$$\omega_n^{jk} = (\omega_n^j)^k = (\omega_n^k)^j$$

Define $v_k := \left[\omega_n^{jk}\right]_{j=0}^{n-1} \in \mathbb{C}^n \qquad k \in \{0, \dots, n-1\}$

and represent C as $C_{i,j} = c_{ij} = u_{i-j}$

for n-periodic $u \in \ell^\infty(\mathbb{Z})$, $u_i \in \mathbb{C}$

$$(Cv_k)_j = \sum_{\ell=0}^{n-1} c_{j\ell}\,(v_k)_\ell = \sum_{\ell=0}^{n-1} u_{j-\ell}\,\omega_n^{\ell k}$$

$$= \sum_{\ell=0}^{n-1} u_\ell\,\omega_n^{(j-\ell)k} = \omega_n^{jk} \boxed{\underbrace{\sum_{\ell=0}^{n-1} u_\ell\,\omega_n^{-\ell k}}_{=:\lambda_k}}$$

(ind. of j)

$$= (v_k)_j \cdot \lambda_k$$

$$\Rightarrow \quad Cv_k = \lambda_k v_k$$

$\Rightarrow$ v_k is an eigenvector to C with eigenvalue λ_k!

$$\lambda_k = \sum_{\ell=0}^{n-1} u_\ell\,\omega_n^{-\ell k}$$

<u>Note</u>: λ_k depends on C

<u>But</u>: v_k is independent of C!

$\Rightarrow$ all circulant matrices of the same dimensions have the same set of eigenvectors!

$\{v_0, \dots, v_{n-1}\} \subset \mathbb{C}^n$: trigonometric basis of $\mathbb{C}^n$

$$\{\mathbf{v}_0,\dots,\mathbf{v}_{n-1}\} = \left\{ \begin{bmatrix} \omega_n^0 \\ \vdots \\ \\ \vdots \\ \omega_n^0 \end{bmatrix} \begin{bmatrix} \omega_n^0 \\ \omega_n^1 \\ \vdots \\ \vdots \\ \omega_n^{n-1} \end{bmatrix} \cdots \begin{bmatrix} \omega_n^0 \\ \omega_n^{n-2} \\ \omega_n^{2(n-2)} \\ \vdots \\ \vdots \\ \omega_n^{(n-1)(n-2)} \end{bmatrix} \begin{bmatrix} \omega_n^0 \\ \omega_n^{n-1} \\ \omega_n^{2(n-1)} \\ \vdots \\ \vdots \\ \omega_n^{(n-1)^2} \end{bmatrix} \right\}. \tag{4.2.11}$$

Orthogonality:

$k \neq m$: $\quad v_k^H v_m = \sum_{j=0}^{n-1} \omega_n^{-jk} \omega_n^{jm} = \sum_{j=0}^{n-1} \omega_n^{j(m-k)}$

$$\underset{k\neq m}{=} \frac{1-\omega_n^{(m-k)n}}{1-\omega_n^{m-k}} = \frac{1-e^{-2\pi i (m-k)n/n}}{1-e^{-2\pi i (m-k)/n}}$$

$$= 0$$

$$v_m^H v_m = \sum_{j=0}^{n-1} \omega_n^{-jm} \omega_n^{jm} = n$$

Fourier matrix:

$$\mathbf{F}_n = \begin{bmatrix} \omega_n^0 & \omega_n^0 & \cdots & \omega_n^0 \\ \omega_n^0 & \omega_n^1 & \cdots & \omega_n^{n-1} \\ \omega_n^0 & \omega_n^2 & \cdots & \omega_n^{2n-2} \\ \vdots & \vdots & & \vdots \\ \omega_n^0 & \omega_n^{n-1} & \cdots & \omega_n^{(n-1)^2} \end{bmatrix} = \left[\omega_n^{lj}\right]_{l,j=0}^{n-1} \in \mathbb{C}^{n,n}. \tag{4.2.13}$$

Lemma 4.2.14. Properties of Fourier matrix

The scaled Fourier-matrix $\frac{1}{\sqrt{n}}\mathbf{F}_n$ *is unitary (→ Def. 6.2.2) :* $\underline{\underline{\mathbf{F}_n^{-1}}} = \frac{1}{n}\mathbf{F}_n^H = \frac{1}{n}\overline{\mathbf{F}}_n$.

[Columns of F_n are $v_0, \dots, v_{n-1}$]

$\Rightarrow \quad F_n^H F_n = n \cdot I_n \qquad (\overline{F_n} F_n = n I_n)$

$C v_k = \lambda_k v_k$

$C F_n = F_n \operatorname{diag}(\lambda_0, \dots, \lambda_{n-1}) \qquad$ (compact notation)

$C = F_n \operatorname{diag}(\lambda_0, \dots, \lambda_{n-1}) F_n^{-1}$

$$C = F_n \operatorname{diag}(\overline{F}_n u) F_n^{-1}$$

$\lambda_k = v_k^H u$

or equivalently: $C = F_n^{-1} \operatorname{diag}(F_n u) F_n$

(to see this take

$$\overline{C F_n} = \overline{F_n \operatorname{diag}(\overline{F}_n u)}$$

$$\overline{C}\,\overline{F}_n = \overline{F}_n \operatorname{diag}(F_n \overline{u})$$ this is true for any

circulant matrix, also for $\overline{C}$ defined by vector $\overline{u}$

$\Rightarrow \quad C\,\overline{F}_n = \overline{F}_n \operatorname{diag}(F_n u)$.)

Lemma 4.2.16. Diagonalization of circulant matrices ($\to$ Def. 4.1.38)

For any circulant matrix $\mathbf{C} \in \mathbb{K}^{n,n}$, $c_{ij} = u_{i-j}$, $(u_k)_{k\in\mathbb{Z}}$ n-periodic sequence, holds true

$$\mathbf{C}\overline{\mathbf{F}}_n = \overline{\mathbf{F}}_n \operatorname{diag}(d_1,\dots,d_n) \quad , \quad \mathbf{d} = \mathbf{F}_n[u_0,\dots,u_{n-1}]^\top .$$

The mapping $\mathcal{F}_n : y \mapsto F_n y$ is called DFT

Definition 4.2.18. Discrete Fourier transform (DFT)

The linear map $\mathcal{F}_n : \mathbb{C}^n \mapsto \mathbb{C}^n$, $\mathcal{F}_n(\mathbf{y}) := \mathbf{F}_n\mathbf{y}$, $\mathbf{y} \in \mathbb{C}^n$, is called discrete Fourier transform (DFT), i.e. for $\mathbf{c} := \mathcal{F}_n(\mathbf{y})$

$$c_k = \sum_{j=0}^{n-1} y_j\,\omega_n^{kj} \quad , \quad k = 0,\dots,n-1 . \tag{4.2.19}$$

Inverse DFT:

$$\underline{c} = F_n \underline{y} \quad \Rightarrow \quad \underline{y} = F_n^{-1}\underline{c} = \frac{1}{n}\overline{F}_n \underline{c}$$

$$\underbrace{c_k = \sum_{j=0}^{n-1} y_j\,\omega_n^{kj}}_{\text{DFT of } \underline{y}} \;\Leftrightarrow\; y_j = \underbrace{\frac{1}{n}\sum_{k=0}^{n-1} c_k\,\omega_n^{-kj}}_{\text{inverse DFT of } \underline{c}} \tag{4.2.20}$$

Note on total least-squares

$Ax = b \qquad A \in \mathbb{K}^{m,n} \qquad m > n \qquad \text{rank}(A) = n$

if the system is perturbed, then most likely

$b \notin \mathcal{R}(A) \Rightarrow \text{rank}([A \;\; b]) = n+1$

Find $[\hat{A} \;\; \hat{b}]$ s.t. $\text{rank}([\hat{A} \;\; \hat{b}]) = n$

$$\hat{A}\, V_{1:n,n+1} + \underbrace{V_{n+1,n+1}}_{\text{if } =0}\, \hat{b} = 0$$

$V_{1:n,n+1} \neq 0 \qquad \hat{A}\, V_{1:n,n+1} = 0$

$\Rightarrow V_{1:n,n+1} \in \mathcal{N}(\hat{A}) \neq \{0\}$

$\Rightarrow \text{rank}(\hat{A}) = n-1$

Need $V_{n+1,n+1} \neq 0$!

Example:

$$[A \;\; b] = \begin{bmatrix} 1 & & & & 0 \\ & 1 & & & \vdots \\ & & \ddots & & \vdots \\ & & & 1 & 0 \\ & & & 0.1 & 1 \\ & 0 & & & \\ & & & & 0 \end{bmatrix}$$

$$\hat{A} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \hat{b} \cdot 0 = 0 \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \in \mathcal{N}(\hat{A})$$

Note: $b \perp \mathcal{R}(A)$

No small perturbation can result in $\hat{b} \in \mathcal{R}(\hat{A})$

C++-code 3.5.6: Total least squares via SVD

```
// computes only solution x of fitted consistent LSE
VectorXd lsqtotal(const MatrixXd& A, const VectorXd& b) {
  const unsigned m = A.rows(), n = A.cols();
  MatrixXd C(m, n + 1);  C << A, b; // C = [A,b]
  // We need only the SVD-factor V, see (3.5.3)
  MatrixXd V = C.jacobiSvd(Eigen::ComputeThinU |
     Eigen::ComputeThinV).matrixV();

  // Compute solution according to (3.5.5);
  double s = V(n, n);
  if (std::abs(s) < 1.0E-15) { cerr << "No solution!\n"; exit(1); }
  return (-V.col(n).head(n) / s);
}
```

$$V_{n+1,n+1} \neq 0 \iff \sigma_n(A) > \sigma_{n+1}([A \;\; b])$$

An example on discrete convolutions

Signal: $\underline{x} = [\overset{x_0}{2}, \overset{x_1}{1}, \overset{x_2}{3}, \overset{x_3}{2}]^T$

$\underline{h} = [\underset{h_0}{1}, \underset{h_1}{1}, \underset{h_2}{2}]^T$

1 (Linear) discrete convolution

$$\underline{y} = \underline{x} * \underline{h} = \sum_{j=0}^{3} x_j h_{k-j} \qquad (h_j := 0 \text{ for } j<0 \text{ and } j \geq 3)$$

length 6

2. 4-periodic discrete convolution

$y^4 = x^4 *_4 p^4 \qquad y^4$ 4-per. $\quad \underline{y}^4$

3. 6-periodic discrete convolution

$y^6 = x^6 *_6 p^6 \qquad y^6$ 6-per. $\quad \underline{y}^6$

$\underline{x} = [2, 1, 3, 2]^T$

$\underline{h} = [1, 1, 2]^T$

1. $y_0 = x_0 h_0 = 2$ $\qquad y_3 = x_0 h_3 + x_1 h_2 + x_2 h_1 + x_3 h_0 = 7$

$y_1 = x_0 h_1 + x_1 h_0 = 3$ $\qquad y_4 = x_2 h_2 + x_3 h_1 = 8$

$y_2 = x_0 h_2 + x_1 h_1 + x_2 h_0 = 8$ $\qquad y_5 = x_3 h_2 = 4$

$\underline{y} = [2, 3, 8, 7, 8, 4]^T$

2. 4-periodic (circular) discrete convolution

x^4 4-periodic signal in $\ell^\infty(\mathbb{Z})$

$x^4 = (\ldots, 2, 1, 3, 2, 2, 1, 3, 2, \ldots)$

$$y_k = \sum_{j \in \mathbb{Z}} h_{k-j} x_j$$

$$= \sum_{j=0}^{3} \Big(\overbrace{\sum_{\ell \in \mathbb{Z}} h_{k-j-4\ell}}^{p_{k-j}}\Big) x_j$$

$$p_j = \sum_{\ell \in \mathbb{Z}} h_{j+4\ell}$$

$p_{-2} = h_2, \; p_{-6} = h_2$

$$= \sum_{j=0}^{3} p^4_{k-j} \, x_j$$

$p^4 = (\ldots, 1, 1, 2, \underline{0}, 1, 1, 2, \underline{0}, 1, 1, 2, \underline{0}, \ldots)$

$\underline{y}^4 = [y^4_0, y^4_1, y^4_2, y^4_3]^T$

$$y^4_0 = x_0 \underline{p^4_0} + x_1 \underline{p^4_{-1}} + x_2 \underline{p^4_{-2}} + x_3 \underline{p^4_{-3}}$$

$$= x_0 \underline{h_0} + x_1 \underline{h_3} + x_2 \underline{h_2} + x_3 \underline{h_1} = 10$$

$$y^4_1 = x_0 h_1 + x_1 h_0 + x_2 h_3 + x_3 h_2 = 7$$

$$y^4_2 = x_0 h_2 + x_1 h_1 + x_2 h_0 + x_3 h_3 = 8$$

$$y^4_3 = x_0 h_3 + x_1 h_2 + x_2 h_1 + x_3 h_0 = 7$$

$\underline{y}^4 = [10, 7, 8, 7]^T$

$y^4 = (\ldots, 10, 7, 8, 7, 10, 7, 8, 7, \ldots)$

3. 6-periodic discrete convolution:

$$p^6 = (\dots, 1, 1, 2, 0, 0, 0, 1, 1, 2, 0, 0, 0, 1 \dots)$$

$$x^6 = (\dots 2, 1, 3, 2, 0, 0, 2, 1, 3, 2, 0, 0, \dots)$$

$$\underline{y}^6 = [y_0^6, \dots, y_5^6]^T$$

$$y_0^6 = x_0 h_0 + \cancel{x_1 h_5} + \cancel{x_2 h_4} + \cancel{x_3 h_3} + \cancel{x_4 h_2} + \cancel{x_5 h_1} = 2 = y_0$$

$$y_1^6 = \underline{x_0 h_1 + x_1 h_0} + \cancel{x_2 h_5} + \cancel{x_3 h_4} + \cancel{x_4 h_3} + \cancel{x_5 h_2} = 3 = y_1$$

$$y_2^6 = \underline{x_0 h_2 + x_1 h_1 + x_2 h_0} + \cancel{x_3 h_5} + \cancel{x_4 h_4} + \cancel{x_5 h_3} = 8 = y_2$$

$$y_3^6 = 7 = y_3$$

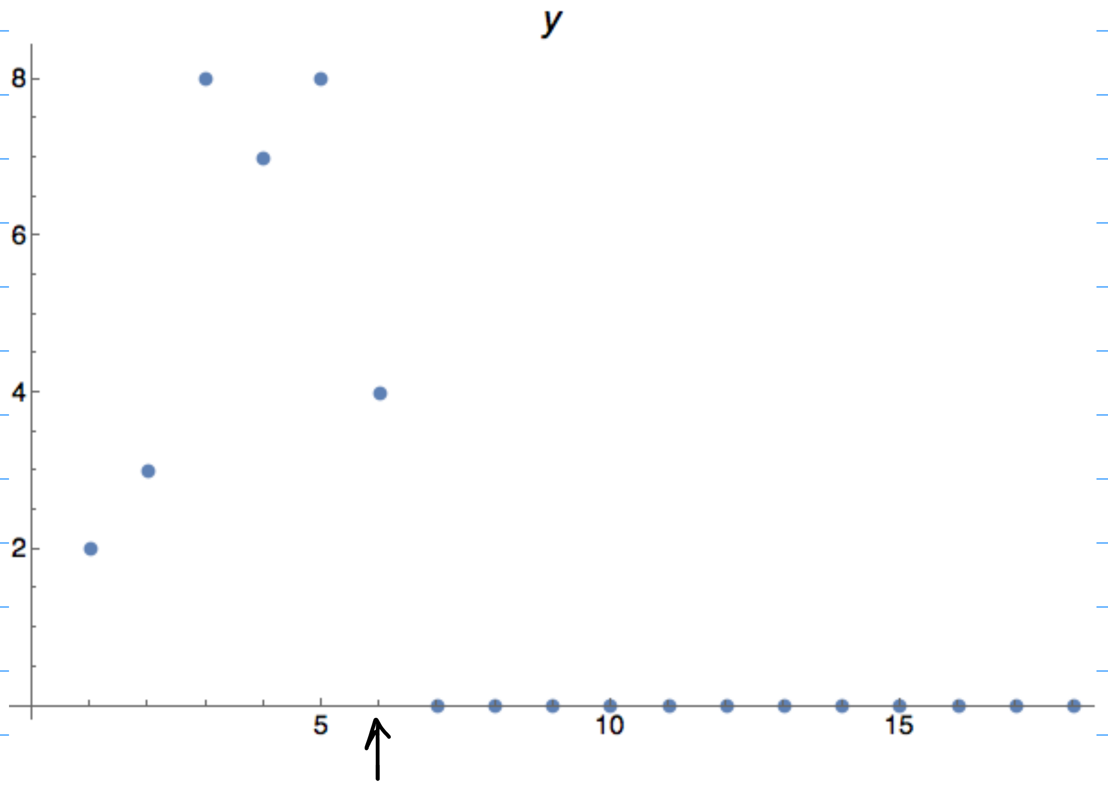
$$y_4^6 = 8 = y_4$$

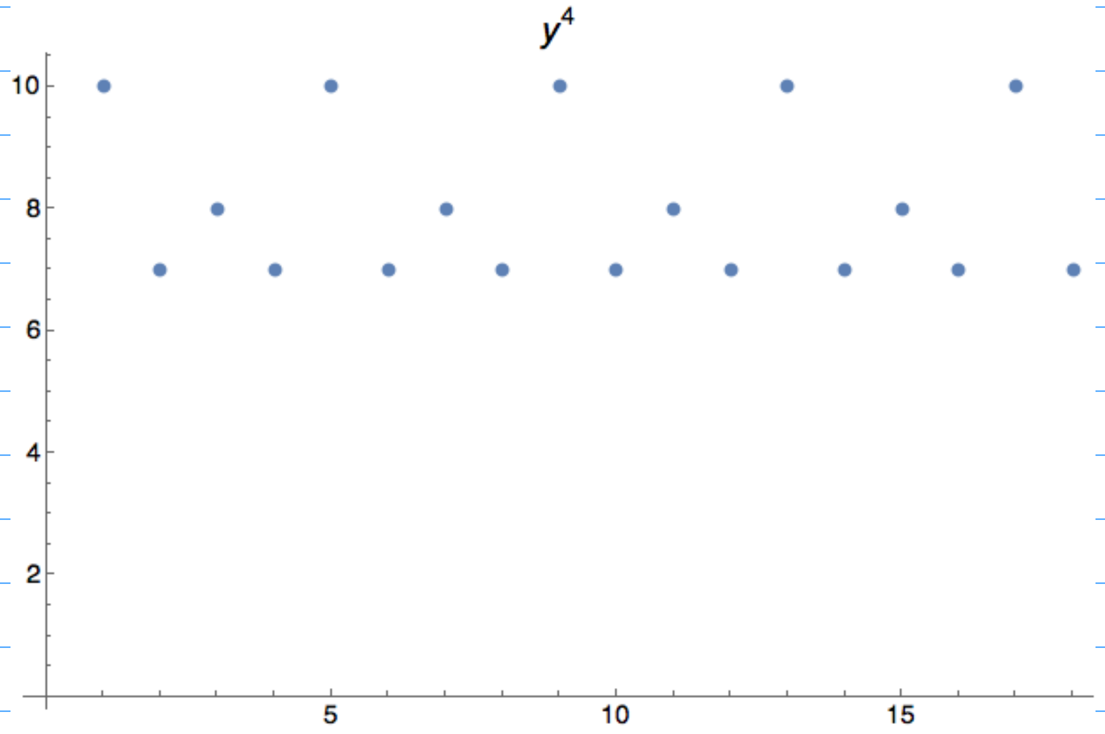
$$y_5^6 = 4 = y_5$$

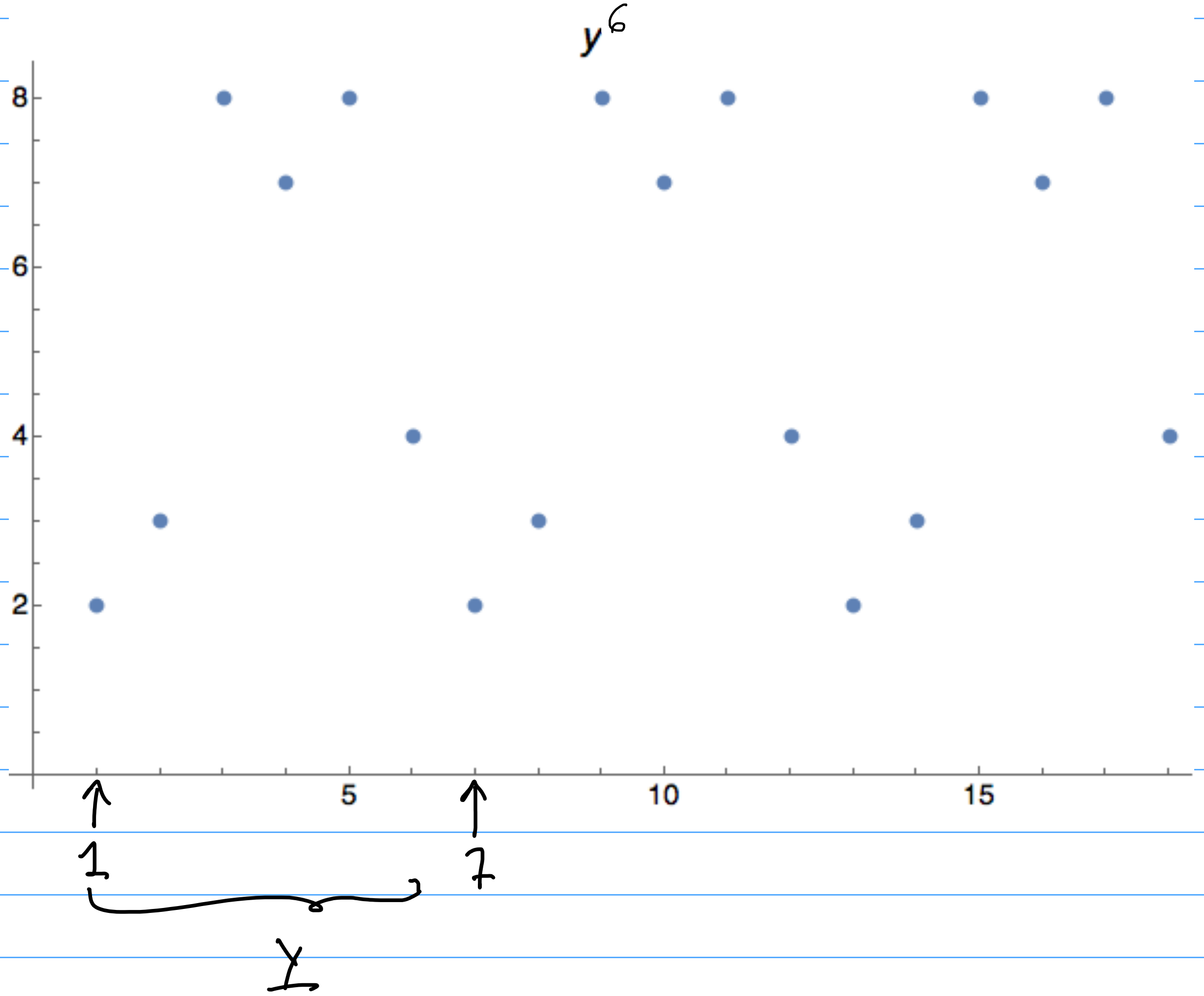
$$y^6 = (\dots, 2, 3, 8, 7, 8, 4, 2, 3, 8, 7, 8, 4, 2, \dots)$$

$\Rightarrow$ y^6 is the 6-periodized signal obtained from $\underline{y} = [2, 3, 8, 7, 8, 4]^T$

In pictures:







$\underline{x} = [2, 1]^T$

$x = (\ldots, 2, 1, 2, 1, \ldots)$

$\underline{h} = [1, 1, 2]^T$

$y^2 = x *_2 p^2$

$$y_k^2 = \sum_{j \in \mathbb{Z}} h_{k-j} x_j \qquad k = 0, 1$$

$$= \sum_{j=0}^{1} \left(\sum_{\ell \in \mathbb{Z}} h_{k-j-2\ell} \right) x_j$$

$$= x_0 \cdot p_k + x_1 p_{k-1}$$

$\underline{p_0 = h_0 + h_2}$

$\underline{p_{-1} = h_1}$

<u>Conclusion</u> : $\underline{x}$ length 4, $\underline{h}$ length 3

$\underline{y} = \underline{x} * \underline{h}$ length 6

$y^6 = x^6 *_6 h^6$ y^6 is periodized version of y

<u>BUT</u>: a shorter period (e.g. length 4) results in a different output!

4.2.1. Discrete Convolution via DFT

$\underline{x} = [x_0, \ldots, x_{m-1}]^T$

$\underline{h} = [h_0, \ldots, h_{n-1}]^T$

$\underline{y} = \underline{x} * \underline{h}$ as periodic discrete convolution

$y^L = x^L *_L p^L$ (L-periodic signals)

$*_L$ can be described by circulant matrix

$C \in \mathbb{C}^{L,L}$ $\underline{y}^L = C \underline{x}^L$

$\left(p_j^L = \sum_{\ell \in \mathbb{Z}} h_{j+\ell L} \right)$

$$\underline{y}^L = \underbrace{F_L^{-1} \operatorname{diag}(F_L \underline{p}^L) F_L}_{=C} \overbrace{F_L \underline{x}^L}^{\in \mathbb{C}^L}$$

carries the information on C

$$\underline{y}^L = F_L^{-1} \left[(F_L \underline{p}^L)_0 \cdot (F_L \underline{x}^L)_0, \dots, (F_L \underline{p}^L)_{L-1} \cdot (F_L \underline{x}^L)_{L-1} \right]^T$$

$\Rightarrow$ discrete periodic convolution of x and p: equivalent to computing DFT(x)·DFT(p) and taking its inverse DFT

$\boxed{\text{Convolution Theorem}}$

Theorem 4.2.24. Convolution theorem

The discrete periodic convolution $*_n$ between n-dimensional vectors $\mathbf{u}$ and $\mathbf{x}$ is equal to the inverse DFT of the component-wise product between the DFTs of $\mathbf{u}$ and $\mathbf{x}$; i.e.:

$$(\mathbf{u}) *_n (\mathbf{x}) := \sum_{j=0}^{n-1} u_{k-j} x_j = \underbrace{\mathbf{F}_n^{-1} \left[(\mathbf{F}_n \mathbf{u})_j (\mathbf{F}_n \mathbf{x})_j \right]_{j=1}^{n}}.$$

Discrete periodic convolution in EIGEN:

C++11 code 4.2.25: Discrete periodic convolution: DFT implementation ➺ GITLAB

```
VectorXcd pconvfft(const VectorXcd& u, const VectorXcd& x) {
  Eigen::FFT<double> fft;
  VectorXcd tmp = ( fft.fwd(u) ).cwiseProduct( fft.fwd(x) );
  return fft.inv(tmp);
}
```

inverse DFT

forward DFT

C++11 code 4.2.26: Implementation of discrete convolution (→ Def. 4.1.22) based on periodic discrete convolution ➜ GITLAB

```
VectorXcd myconv(const VectorXcd& h, const VectorXcd& x) {
  const long n = h.size();
  // Zero padding, cf. (4.1.41)
  VectorXcd hp(2*n - 1), xp(2*n - 1);
  hp << h, VectorXcd::Zero(n - 1);
  xp << x, VectorXcd::Zero(n - 1);
  // Periodic discrete convolution of length 2n-1, ??
  return pconvfft(hp, xp);
}
```

zero-padding to length $2n-1$

The Fast Fourier Transform (cf. Ch. 4.3)

So far: DFT as $F_n x$ is $\Theta(n^2)$

Not much gain in transition from discrete convolution to DFT

BUT: Fast implementation of DFT possible

FFT: any algorithm that performs DFT in $\Theta(n \cdot \log_2 n)$

Idea: Divide-and-conquer algorithm