

# Structured Matrices, Multigrid, and Image Processing

MARCO DONATELLI

Dept. of Science and High Technology – U. Insubria (Italy)

TUM 2016





# Outline

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution

Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

- 1 Convolution and Structured Matrices
- 2 Ill-posed problems and regularization
- 3 Multigrid Methods for Structured Matrices



# Outline

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution  
Discrete  
Fourier  
Transform  
Applications  
and generalization  
2D Case

## 1 Convolution and Structured Matrices

- Convolution
- Discrete Fourier Transform
- Applications and generalization
- 2D Case

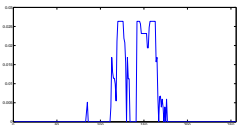
## Definition

The **convolution** of two functions (signals)  $f$  and  $k$  is

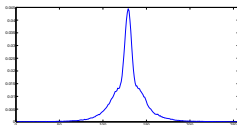
$$g(x) = \int_{\mathbb{R}} k(x-s)f(s)ds$$

In the applications usually  $f$  is the original signal,  $k$  is the convolution kernel and  $g$  is the observed signal.

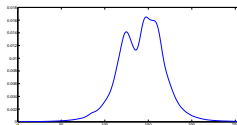
## Example



$f$



$k$



$g$

**Shift-invariant:** every point is subject to the same phenomenon.



## Assumption

$f(x) = 0$  for  $x \notin [a, b]$ .

- The convolution becomes

$$g(x) = \int_a^b k(x-s)f(s)ds$$

- Discretize the integral using  $n$  rectangles defining the grid points

$$x_j = a + jh, \quad h = \frac{b-a}{n}, \quad j = 0, \dots, n-1.$$

- Approximate  $g$  at the grid points  $x_i$ , for  $i = 0, \dots, n-1$ , by

$$\begin{aligned} g(x_i) &= \int_a^b k(x_i-s)f(s)ds \\ &\approx h \sum_{j=0}^{n-1} k(x_i-x_j)f(x_j) \end{aligned}$$



- Defining

$$K_{i,j} = hk(x_i - x_j) = hk((i - j)h)$$

we have that

$$g(x_i) \approx \sum_{j=0}^{n-1} K_{i,j} f(x_j), \quad i = 0, \dots, n - 1,$$

which is the linear system

$$\mathbf{g} = \mathbf{Kf}, \tag{1}$$

where  $g_i = g(x_i)$  and  $f_i = f(x_i)$ , for  $i = 0, \dots, n - 1$ .

- Note that

$$k_{i-j} := hk((i - j)h) = K_{i,j} \tag{2}$$

shift-invariant property.

- Thanks to (2), the matrix

$$K = \begin{bmatrix} k_0 & k_{-1} & \dots & k_{-(n-1)} \\ k_1 & k_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{-1} \\ k_{n-1} & \dots & k_1 & k_0 \end{bmatrix} \quad (3)$$

has constant elements along the diagonals and it is called **Toeplitz matrix**.

- The matrix  $K$  depends on only  $2n - 1$  parameters

$$\mathbf{k} = [k_{-n+1}, \dots, k_{-1}, k_0, k_1, \dots, k_{n-1}]^T.$$

## How to work with $K$

- Memorize only  $\mathbf{k} \in \mathbb{R}^{2n-1}$ .
- Is it possible to save CPU time for the computations (matrix-vector product, inversion, etc.)?



# Discrete Convolution

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution  
Discrete  
Fourier  
Transform  
Applications  
and generalization  
2D Case

- Let  $\mathbf{f} = \mathbf{e}_i$  the  $i$ -th vector of the canonical base, then

$$\mathbf{g} = \mathbf{K}\mathbf{e}_i = [\dots, k_{-1}, k_0, k_1, \dots]^T,$$

hence, if  $k_i = 0$  for  $|i| > n/2$  then  $\mathbf{k}$  can be obtained observing a point in the middle of the interval ... next lesson on inverse problems.

- The linear system (1) is the **discrete convolution with zero-Dirichlets boundary conditions**

$$g_i = \sum_{j=0}^{n-1} K_{i,j} f_j = \sum_{j=0}^{n-1} k_{i-j} f_j, \quad i = 0, \dots, n-1$$

- Rotate the vector  $\mathbf{k}$ , shift, multiply component wise with  $\mathbf{f}$  and then sum:

$$\begin{aligned} g_j &= \sum \begin{matrix} k_{n-1} & \cdots & k_1 & k_0 & k_{-1} & \cdots & k_{-n+1} \\ * & \cdots & * & * & * & \cdots & * \\ \tilde{f}_{j-(n-1)} & \cdots & \tilde{f}_{j-1} & \tilde{f}_j & \tilde{f}_{j+1} & \cdots & \tilde{f}_{j+n-1} \end{matrix} \\ &= k_{n-1} \cdot \tilde{f}_{j-(n+1)} + \cdots + k_1 \cdot \tilde{f}_{j-1} + k_0 \cdot \tilde{f}_j + k_{-1} \cdot \tilde{f}_{j+1} + \cdots + k_{-n+1} \cdot \tilde{f}_{j+n-1} \end{aligned}$$

where

$$\tilde{f}_i = \begin{cases} f_i & \text{if } i = 0, 1, \dots, n-1, \\ 0 & \text{otherwise.} \end{cases}$$





# Full Convolution

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution

Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

- Let  $k_i = 0$  for  $|i| > m$ ,  $m < n - 1$ , then removing Assumption 1 we have the full discrete convolution

$$\mathbf{g} = K_{\text{full}} \tilde{\mathbf{f}}$$

$$= \left[ \begin{array}{cccc|cccc} k_m & \dots & k_1 & k_0 & k_{-1} & \dots & k_{-m} & 0 \\ & k_m & \dots & k_1 & k_0 & k_{-1} & \dots & \ddots \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & \ddots & \dots & k_1 & k_0 & k_{-1} & \dots & k_{-m} \\ 0 & k_m & \dots & k_1 & k_0 & & & & k_{-1} & \dots & k_{-m} \end{array} \right] \tilde{\mathbf{f}}$$

where  $\tilde{\mathbf{f}} = [f_{-m} \dots f_{-1} \mid \mathbf{f}^T \mid f_n, \dots, f_{n+m-1}]^T \in \mathbb{R}^{n+2m}$ .

- No assumptions on the boundary conditions

$$g_i = \sum_{j \in \mathbb{Z}} k_{i-j} f_j, \quad i = 0, \dots, n-1$$



# Circulant matrix

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution

Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

## Assumption

Assume that the *function  $f$  is periodic with period  $b - a$* . Then

$$f_{-i} = f_{n-i}, \quad f_{n+i-1} = f_{i-1}, \quad i = 1, 2, \dots, m.$$

- Let  $m < n/2$ , i.e.,  $\text{supp}(k) \subset [a, b]$  as in the example, then

$$\mathbf{g} = K_{\text{full}} \tilde{\mathbf{f}} = K_{\text{circ}} \mathbf{f}$$

where

$$K_{\text{circ}} = \begin{bmatrix} k_0 & \dots & k_{-m} & 0 & k_m & \dots & k_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ k_m & \ddots & \ddots & \ddots & \ddots & \ddots & k_m \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ k_{-m} & \ddots & \ddots & \ddots & \ddots & \ddots & k_{-m} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ k_{-1} & \dots & k_{-m} & 0 & k_m & \dots & k_0 \end{bmatrix}$$



## Circulant matrix 2

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution

Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

- The matrix  $K_{\text{circ}}$  depends on only  $n$  parameters in the first column

$$\mathbf{k} = [k_0, \dots, k_m, \mathbf{0}, k_{-m}, \dots, k_{-1}]^T \in \mathbb{R}^n$$

- Given the observation of  $\mathbf{e}_{\frac{n+1}{2}}$  ( $n$  odd for simplicity)

$$\check{\mathbf{k}} = [\mathbf{0}, k_{-m}, \dots, k_{-1}, k_0, k_1, \dots, k_m, \mathbf{0}]^T \in \mathbb{R}^n$$

we have that

$$\mathbf{k} = \text{circshift}(\check{\mathbf{k}}, n - i_0),$$

where  $\check{k}_{i_0} = k_0$  (indices start from zero).

- Using the congruence relation index  $\mathbf{k}$  in the standard way

$$\mathbf{k} = \begin{bmatrix} k_0 \\ \vdots \\ k_m \\ \mathbf{0} \\ k_{-m} \\ \vdots \\ k_{-1} \end{bmatrix} \xrightarrow{\text{mod } n} \begin{bmatrix} k_0 \\ k_1 \\ \vdots \\ k_{n-2} \\ k_{n-1} \end{bmatrix}$$

- Circular discrete convolution

$$\mathbf{g} = K_{\text{circ}} \mathbf{f} = \mathbf{f} * \mathbf{k} \quad (4)$$

where

$$g_i = \sum_{j=0}^{n-1} k_{(i-j) \bmod n} f_j, \quad i = 0, \dots, n-1$$



# Discrete Fourier Transform

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution  
Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

## Definition

Let  $\mathbf{f} \in \mathbb{C}^n$  the **Discrete Fourier Transform (DFT)** of  $\mathbf{f}$  is

$$\hat{f}_k := \sum_{j=0}^{n-1} f_j e^{-\frac{i2\pi jk}{n}}, \quad k = 0, \dots, n-1.$$

To simplify the notation define

$$\omega_n := e^{-\frac{i2\pi}{n}}$$

(note that  $\omega_n^k$  is the  $n$ -th root of the unity, for  $k = 0, \dots, n-1$ ), thus

$$\hat{f}_k := \sum_{j=0}^{n-1} \omega_n^{jk} f_j, \quad k = 0, \dots, n-1.$$

In matrix form

$$\hat{\mathbf{f}} = F_n \mathbf{f},$$

where  $[F_n]_{k,j} = \omega_n^{jk}$ , for  $k, j = 0, \dots, n-1$ .

## Proposition

$$\sum_{j=0}^{n-1} \omega_n^{jk} = \begin{cases} n & \text{if } k = sn, s \in \mathbb{Z}, \\ 0 & \text{otherwise.} \end{cases}$$

## Properties of $F_n$

- 1  $F_n = F_n^T$ .
- 2  $F_n^{-1} = \frac{1}{n} F_n^H$ .
- 3  $F_n^H = \mathcal{J} F_n = F_n \mathcal{J}$  where  $\mathcal{J}$  is the permutation matrix

$$\mathcal{J} = \left[ \begin{array}{c|cccc} 1 & & & & \\ \hline & & & & 1 \\ & & & & \\ & & & \ddots & \\ & & 1 & & \end{array} \right]$$

## Corollary

$$F_n^{-1} = \frac{1}{n} \mathcal{J} F_n \quad (5)$$



# Fast Fourier Transform (FFT)

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution

Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

## Remark

*Thanks to (5),  $F_n^{-1}\mathbf{x}$  can be computed using the same algorithm implemented for the direct product  $F_n\mathbf{x}$ .*

## Fast Fourier Transform (FFT)

- The matrix-vector requires  $O(n^2)$  arithmetic operations but when the matrix is  $F_n$  it can be computed in  $O(n \log(n))$  by FFT for  $n = 2^\alpha$ .
- FFT was included in the Top 10 Algorithms of 20<sup>th</sup> Century.
- Different algorithms (decimation in time or decimation in space) can be used and several implementation details can be found in C. Van Loan, ‘‘Computational Frameworks for the Fast Fourier Transform’’, Frontiers in Applied Mathematics, SIAM, 1992.



# Convolution Theorem

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution  
Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

## Theorem

Let  $\mathbf{f}, \mathbf{k} \in \mathbb{C}^n$ , then

$$F_n(\mathbf{f} * \mathbf{k}) = (F_n \mathbf{f}) \circ (F_n \mathbf{k}),$$

where  $*$  is defined in (4) and  $\circ$  is the Hadamard (entrywise) product.

## Spectral decomposition of $K_{\text{circ}}$

From the previous theorem and Property 2, it holds

$$K_{\text{circ}} \mathbf{f} = \mathbf{f} * \mathbf{k} = \frac{1}{n} F_n^H (F_n \mathbf{f} \circ F_n \mathbf{k}) = \frac{1}{n} F_n^H \text{diag}(F_n \mathbf{k}) F_n \mathbf{f} \quad (6)$$

Since (6) has to hold for all  $\mathbf{f} \in \mathbb{C}^n$ , it must be

$$K_{\text{circ}} = \frac{1}{n} F_n^H \text{diag}(F_n \mathbf{k}) F_n \quad (7)$$





# Toeplitz matrices vs circulants

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution

Discrete  
Fourier  
Transform

Applications  
and generalization

2D Case

- Goal: compute the **product  $Kx$  with  $K$  Toeplitz** matrix in (3).
- Construct the circulant matrix

$$C = \begin{bmatrix} K & M_1 \\ M_2 & M_3 \end{bmatrix} \in \mathbb{C}^{m \times m}$$

with  $m \geq 2n - 1$  and  $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix}$ .

- Compute  $\mathbf{z} = C\mathbf{y}$ , thus

$$K\mathbf{x} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$$

- Choosing  $m$  as the smallest power of 2 such that  $m \geq 2n - 1$ , pad with zeros if necessary, we can use FFT with a cost of  $O(n \log n)$ .



# FFT of arbitrary size

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution  
Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

- The matrix-vector product of Toeplitz matrices of arbitrary size  $n$  can be computed by immersion into a circulant of size  $m = 2^\alpha$  and then applying the FFT.
- **How compute FFT of arbitrary size? By Toeplitz matrices!**
- Use the relation  $-jk = ((k-j)^2 - k^2 - j^2)/2$ .
- It holds

$$F_n = \left[ e^{-\frac{i2\pi jk}{n}} \right]_{j,k=0}^{n-1} = \left[ e^{\frac{i\pi(((k-j)^2 - k^2 - j^2))}{n}} \right]_{j,k=0}^{n-1} = DTD$$

where

$$D = \text{diag}_{k=0,\dots,n-1} \left( e^{-\frac{i\pi k^2}{n}} \right), \quad T = \left[ e^{\frac{i\pi(k-j)^2}{n}} \right]_{k,j=0}^{n-1}.$$



# Boundary conditions

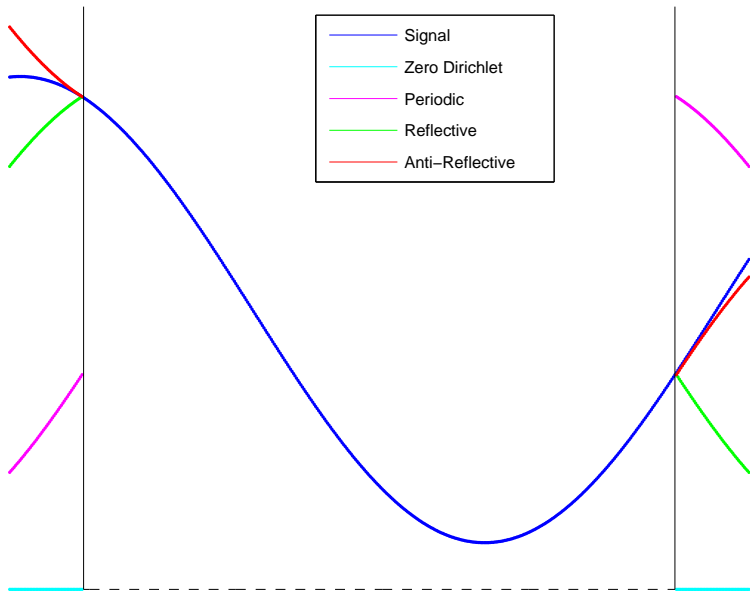
Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution  
Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

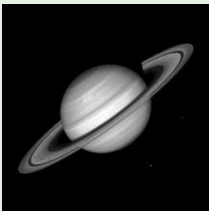


## Definition

The *convolution* of two functions (signals)  $f$  and  $k$  is

$$g(x_1, x_2) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} k(x_1 - s_1, x_2 - s_2) f(s_1, s_2) ds_1 ds_2$$

## Example



$f$



$k$



$g$



# Discretization

Structured  
Matrices,  
Multigrid,  
and Image  
Processing

M.  
Donatelli

Convolution  
and  
Structured  
Matrices

Convolution

Discrete  
Fourier  
Transform

Applications  
and generalization  
2D Case

- Discretize on a uniform grid on  $[a_1, b_1] \times [a_2, b_2]$ .
- Resizing  $n \times m$  images in vectors of length  $nm$  concatenating the columns, we obtain the linear system

$$\mathbf{g} = K\mathbf{f}$$

where  $K$  is the **block-Toeplitz-Toeplitz-block (BTTB)** matrix

$$K = \begin{bmatrix} K_0 & K_{-1} & \dots & K_{-(n-1)} \\ K_1 & K_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & K_{-1} \\ K_{n-1} & \dots & K_1 & K_0 \end{bmatrix},$$

$$K_j = \begin{bmatrix} k_{j,0} & k_{j,-1} & \dots & k_{j,-m+1} \\ k_{j,1} & k_{j,0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{j,-1} \\ k_{j,m-1} & \dots & k_{j,1} & k_{j,0} \end{bmatrix}, \quad k_{j,s} = h_x h_y k(jh_x, sh_j).$$



- Circulant matrices have a similar block circulant circulant block (BCCB) structure.
- **2D DFT by tensor product**

$$F_n^{2D} = F_n \otimes F_n.$$

- Since  $(A \otimes B)\text{vec}(X) = \text{vec}(BXA^T)$  it holds

$$F_n^{2D} \text{vec}(X) = \text{vec}(F_n X F_n),$$

which is the application of the 1D DFT to each row and column of  $X$ .

# Ill-posed problems and regularization

MARCO DONATELLI

Dept. of Science and High Technology – U. Insubria (Italy)

TUM 2016





# Outline

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- 1 Inverse and ill-posed problems
- 2 Least squares
- 3 Regularization
- 4 Iterative regularization methods
- 5 Sparsity constraint



## Inverse problems

From the observation of a phenomenon we would obtain its birth.

## Example

- A classical example is the **Fredholm integral equation of the first kind**

$$g(x) = \int_{\mathbb{R}} k(x, s)f(s)ds \quad (1)$$

- **Discrete example**

$$Ax = y$$

- The matrix-vector product is the direct problem.
- The solution of the linear system, i.e.,  $x = A^{-1}y$  is the inverse problem.

## Remark

*Continuous inverse problems are often ill-posed.*



# Ill-posed Problems

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

A good book: H.W. Engl, M. Hanke, A. Neubauer, ‘‘Regularization of Inverse Problems’’, Kluwer Academic Publishers, 1996.

## Definition

We say that a mathematical problem is well-posed if

- 1 a solution exists;
- 2 the solution is unique;
- 3 the solution depends continuously on the data.

We say that a mathematical problem is **ill-posed** if one of the conditions above **does not hold**.

- The Riemann-Lebesgue lemma states that the integral will approach zero as the number of oscillations increases  $\Rightarrow$  (1) is ill-posed.
- The discretization of an ill-posed problem is severely ill-conditioned  $\Rightarrow$  discrete ill-posed problems, see  
P. C. Hansen, ‘‘Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion’’,  
Mathematical Modeling and Computation, SIAM, 1998.



# Outline

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- 1 Inverse and ill-posed problems
- 2 Least squares**
- 3 Regularization
- 4 Iterative regularization methods
- 5 Sparsity constraint

Given  $A \in \mathbb{C}^{m \times n}$  and  $\mathbf{b} \in \mathbb{C}^m$ , instead of to solve  $A\mathbf{x} = \mathbf{b}$  compute

$$\operatorname{argmin}_{\mathbf{x} \in \mathbb{C}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2. \quad (2)$$

## Definition

Let  $A \in \mathbb{C}^{m \times n}$ , then exist  $U$  and  $V$  unitary matrices such that the **singular values decomposition (SVD)** of  $A$  is

$$A = U\Sigma V^H,$$

with  $\Sigma = \operatorname{diag}_{i=1, \dots, t}(\sigma_i) \in \mathbb{R}^{m \times n}$ ,  $t = \min(m, n)$  and  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_t \geq 0$ .

- Let  $r = \operatorname{rank}(A)$ , then

$$A = U\Sigma V^H = U_r \Sigma_r V_r^H = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^H.$$

- The minimum solution of (2) is

$$\mathbf{x}^\dagger = A^\dagger \mathbf{b}.$$

## Definition

The **condition number** of a matrix is

$$\nu_2(A) = \|A\|_2 \|A^\dagger\|_2 = \frac{\sigma_1}{\sigma_r}.$$

If  $\sigma_r \approx 0$  then  $\nu_2(A) \gg 1$  and it may be that in exact arithmetic  $\sigma_r = 0$ .

## Definition

Given a matrix  $A$ , its **truncated singular values decomposition** of order  $s \leq r$  is

$$A_s = \sum_{i=1}^s \sigma_i \mathbf{u}_i \mathbf{v}_i^H.$$

## Lemma

$$\|A - A_s\|_2 = \min_{\substack{B \in \mathbb{C}^{m \times n} \\ \text{rank}(B) = s}} \|A - B\|_2 = \sigma_{s+1}.$$



# Sensitivity analysis

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- The observed object is usually affected by noise:

$$\mathbf{b}^\delta = \mathbf{b} + \boldsymbol{\xi}, \quad \mathbf{b} = \mathbf{A}\mathbf{x}^\dagger,$$

where  $\delta = \|\boldsymbol{\xi}\|_2$  is the noise level.

- The computed solution becomes

$$\tilde{\mathbf{x}} = \mathbf{A}^\dagger \mathbf{b}^\delta = \mathbf{A}^\dagger (\mathbf{b} + \boldsymbol{\xi}) = \mathbf{x}^\dagger + \mathbf{e},$$

where

$$\mathbf{e} = \mathbf{A}^\dagger \boldsymbol{\xi} = \sum_{i=1}^r \frac{\mathbf{u}_i^H \boldsymbol{\xi}}{\sigma_i} \mathbf{v}_i.$$

If  $\sigma_i \ll 1$  and  $\mathbf{u}_i^H \boldsymbol{\xi} \neq 0$ , then  $\mathbf{e}$  can be large even if  $\delta$  is small.



# Outline

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- 1 Inverse and ill-posed problems
- 2 Least squares
- 3 Regularization**
- 4 Iterative regularization methods
- 5 Sparsity constraint

## Discrete ill-posed problems

- The singular values decays exponentially at zero without a significant gap.
- The singular vectors  $\mathbf{v}_j$  and  $\mathbf{u}_j$  are the  $j$ -th frequency.
- The noise  $\xi$  has nonzero components also in the high frequencies.

$$\Rightarrow \|\mathbf{e}\|_2 \gg 1 \Rightarrow$$

## Regularization

Change a little bit the problem obtaining a new nearby problem well-posed.  
There is always a **parameter** which balances:

- 1 how the new problem is far from the original one (approximation error)
- 2 how much the new problem is sensible to noise (stability).





# Truncated SVD (TSVD)

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- Instead of  $A^\dagger \mathbf{b}^\delta$  take  $A_s^\dagger \mathbf{b}^\delta$ ,  $0 < s < r$ , as approximation of  $\mathbf{x}^\dagger$ :

$$\begin{aligned}\mathbf{x}_s^\delta &= A_s^\dagger \mathbf{b}^\delta = \sum_{i=1}^s \frac{\mathbf{u}_i^H \mathbf{b}}{\sigma_i} \mathbf{v}_i + \sum_{i=1}^s \frac{\mathbf{u}_i^H \boldsymbol{\xi}}{\sigma_i} \mathbf{v}_i \\ &= \mathbf{x}^\dagger - \sum_{i=s+1}^r \frac{\mathbf{u}_i^H \mathbf{b}}{\sigma_i} \mathbf{v}_i + \sum_{i=1}^s \frac{\mathbf{u}_i^H \boldsymbol{\xi}}{\sigma_i} \mathbf{v}_i\end{aligned}$$

where the first term is the truncation (approximation) error and the second term is the noise amplification (stability).

- $s$  is the regularization parameter.
- The computed solution can be written as

$$\mathbf{x}_s^\delta = V \Phi_s \Sigma^\dagger U^H \mathbf{b}^\delta, \quad \Phi_s = \begin{bmatrix} I_s & \\ & 0 \end{bmatrix}_{n \times n}$$

- A different choice of  $\Phi_s$  can be applied, but must be a low-pass filter.

- Tikhonov method is

$$\operatorname{argmin}_{\mathbf{x} \in \mathbb{C}^n} \left\{ \|\mathbf{A}\mathbf{x} - \mathbf{b}^\delta\|_2^2 + \alpha \|\mathbf{x}\|_2^2 \right\}, \quad (3)$$

where  $\alpha$  balances the data fitting and the noise explosion.

- The solution of (3) is equivalent to the linear system

$$(\mathbf{A}^H \mathbf{A} + \alpha \mathbf{I}) \mathbf{x} = \mathbf{A}^H \mathbf{b}^\delta,$$

thus

$$\begin{aligned} \mathbf{x} &= \mathbf{V}(\boldsymbol{\Sigma}^T \boldsymbol{\Sigma} + \alpha \mathbf{I})^{-1} \boldsymbol{\Sigma}^T \mathbf{U}^H \mathbf{b}^\delta \\ &= \mathbf{V} \boldsymbol{\Phi}_{Tik} \boldsymbol{\Sigma}^\dagger \mathbf{U}^H \mathbf{b}^\delta \end{aligned}$$

where  $\boldsymbol{\Phi}_{Tik} = (\boldsymbol{\Sigma}^T \boldsymbol{\Sigma} + \alpha \mathbf{I})^{-1} \boldsymbol{\Sigma}^T \boldsymbol{\Sigma} = \operatorname{diag}_{i=1, \dots, t}(\phi_i)$ , such that

$$\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha} \approx \begin{cases} 1 & i \text{ small,} \\ 0 & i \text{ large,} \end{cases} \quad i = 1, \dots, t.$$



# Outline

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- 1 Inverse and ill-posed problems
- 2 Least squares
- 3 Regularization
- 4 Iterative regularization methods**
- 5 Sparsity constraint



# Iterative regularization methods

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- The relative error begins to decrease until a certain “optimal” iteration is reached and then begins to increase because of the presence of noise, which starts to dominate the restoration process (**semi-convergence**).
- By stopping the iterations when the error is low, we obtain a regularized approximation of the solution.
- **Landweber method** (gradient descent method for (2))

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau A^H (\mathbf{b}^\delta - A\mathbf{x}_k), \quad (4)$$

which is convergent if  $0 < \tau < \frac{2}{\rho(A^H A)}$ .

Convergence in the noise free case

## Lemma

If  $\mathbf{x}_0 = \mathbf{0}$  then Landweber method in (4) converges to  $\mathbf{x}^\dagger = A^\dagger \mathbf{b}^\delta$ .



# Filter factor of Landweber method

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- From the proof of the previous Lemma we obtain the **filter factors**  $\theta_{k,i}$  s.t.

$$\mathbf{x}_k = V\Phi_{L,k}\Sigma^\dagger U^H \mathbf{b}^\delta, \quad \Phi_{L,k} = \text{diag}_{i=1,\dots,t}(\theta_{k,i}), \quad \theta_{k,i} = 1 - (1 - \tau\sigma_i^2)^{k+1}$$

- Fix  $k$ , it holds

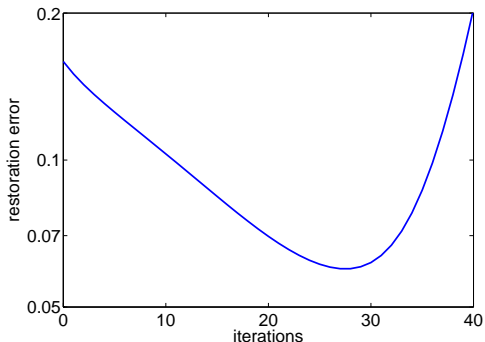
$$\theta_{k,i} = \begin{cases} 1 & i \text{ small,} \\ 0 & i \text{ large,} \end{cases} \quad i = 1, \dots, t.$$

- Fix  $i$  and let  $s > k$ , then it holds

$$\theta_{s,i} > \theta_{k,i}$$

**Landweber methods starts reconstructing the low frequencies** and then passes at the medium and high frequencies.

- Classical **iterative methods** firstly reduce the algebraic error into the low frequencies (well-conditioned subspace), when they arrive to reduce the algebraic error into the high frequencies then the restoration error increases because of the noise.
- **The regularization parameter is the stopping iteration.**





# Iterated Tikhonov method

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- **Iterated Tikhonov method** is obtained refining a given approximation  $\mathbf{x}_k$  by solving the error equation using Tikhonov method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (A^H A + \alpha I)^{-1} A^H (\mathbf{b}^\delta - A \mathbf{x}_k). \quad (5)$$

- It is convergent for  $\alpha > 0$  and  $\mathbf{x} \rightarrow \mathbf{x}^\dagger$  whenever  $\mathbf{x}_0 = \mathbf{0}$ .
- The iteration (5) can be interpreted as a **preconditioned Landweber** method, where  $\tau = 1$  and the preconditioner is  $(A^H A + \alpha I)^{-1}$ .
- Further **regularization parameter**  $\alpha$ , which balances the convergence speed and how much is steep the semiconvergence:
  - small  $\alpha \implies$  fast convergence but unstable convergence,
  - large  $\alpha$  slow convergence like Landweber.
- $(A^H A + \alpha I)^{-1}$  could be computationally expensive. Hence it should be approximated, but what happens at the convergence?



# Estimation of the regularization parameter

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

Let  $\beta$  be the regularization parameter.

- **Discrepancy principle**, largely used with iterative methods, requires to know  $\delta$ . Compute the approximation corresponding to the smallest  $\beta$  that satisfies the condition

$$\|\mathbf{Ax}_\beta - \mathbf{b}^\delta\|_2 < \nu\delta, \quad \nu > 1.$$

- **L-curve**: Compute  $\mathbf{x}_\beta$  for several values of  $\beta$  and plot in log-scale  $\|\mathbf{x}_\beta\|_2$  and  $\|\mathbf{Ax}_\beta - \mathbf{b}^\delta\|_2$ , then the best value of  $\beta$  is in the corner of the L-shape curve balancing data fitting and explosion of noise.
- Generalized cross-validation (GCV), etc.





# Outline

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- 1 Inverse and ill-posed problems
- 2 Least squares
- 3 Regularization
- 4 Iterative regularization methods
- 5 Sparsity constraint



- The  $\ell_2$ -norm leads to over-smoothed restorations.
- In some applications other regularization terms could be useful, like the  $\ell_1$ -norm if the solution is sparse (e.g. images in the wavelets domain).
- Let  $W^H$  be a wavelet or tight-frame synthesis operator ( $W^H W = I$ ) and  $\mathbf{y}$  the frame coefficients such that

$$\mathbf{x} = W^H \mathbf{y}.$$

- Let  $\mathbf{x}$  be an image, then  $\mathbf{y}$  is sparse (wavelet coefficients).
- The  $\ell_2 - \ell_1$  minimum problem is

$$\operatorname{argmin}_{\mathbf{y} \in \mathbb{C}^n} \left\{ \frac{1}{2} \|B\mathbf{y} - \mathbf{b}^\delta\|_2^2 + \alpha \|\mathbf{y}\|_1 \right\}, \quad (6)$$

where  $B = AW^H$ .



# Iterative soft-thresholding algorithm (ISTA)

Ill-posed  
problems  
and regula-  
rization

M.  
Donatelli

Inverse and  
ill-posed  
problems

Least  
squares

Regularization

Iterative re-  
gularization  
methods

Sparsity  
constraint

- Let the nonlinear **soft-thresholding** operator  $\mathbf{S}_\mu$  be defined component-wise as

$$[\mathbf{S}_\mu(\mathbf{y})]_i = S_\mu(y_i),$$

with  $S_\mu$  the soft-thresholding function

$$S_\mu(y_i) = \text{sgn}(y_i) \max\{|y_i| - \mu, 0\}.$$

- Combining Landweber and soft-thresholding we obtain the ISTA

$$\mathbf{x}_{k+1} = \mathbf{S}_\mu(\mathbf{x}_k + \tau A^H(\mathbf{b}^\delta - A\mathbf{x}_k)), \quad (7)$$

which converges to the solution of (6)

I. Daubechies, M. Defrise, and C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Comm. Pure Appl. Math.*, 57--11 (2004), pp. 1413--1457.