

# Tomographic reconstruction

## Lectures 2 and 3: Iterative reconstruction methods

Camille Pouchol<sup>1</sup>  
pouchol@kth.se

Department of Mathematics  
KTH, Stockholm

HL2027: 3D Image Reconstruction and Analysis in Medicine  
January/February 2019

---

<sup>1</sup>based on a previous lecture by Sebastian Banert, Ozan Öktem

# Review from last lecture

The fully discretised reconstruction problem

## Image reconstruction

Recover the digital image  $\alpha_{\text{true}} \in \mathbb{R}^n$  from measured data  $g \in \mathbb{R}^m$  assuming

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}. \quad (1)$$

Here,  $\mathbf{A}$  is the  $(m \times n)$ -measurement matrix and  $g_{\text{noise}} \in \mathbb{R}^m$  is the noise component of data,  $\alpha_{\text{true}} = \langle f, \phi_j \rangle_{\mathcal{X}}$ .

Measurement matrix:

$$\mathbf{A} := \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}.$$

Transmission tomography and using the voxel basis:

$a_{i,j}$  = length of  $j$ :th line through  $i$ :th voxel.

# Reconstruction methods

## Least-squares solutions

One can show that  $\alpha^\dagger \in \mathbb{R}^n$  solves

$$\min_{\alpha} \|\mathbf{A} \cdot \alpha - g\|_2^2 \quad (2)$$

if it solves the normal equations:  $\mathbf{A}^t \cdot (\mathbf{A} \cdot \alpha^\dagger - g) = \mathbf{0}$ .

- If  $\mathbf{A}$  is the matrix representing the discretised ray transform, then  $\mathbf{A}^t$  (the transpose of  $\mathbf{A}$ ) is the matrix representing the discretised backprojection.
- $\mathbf{A}^t \cdot \mathbf{A}$  is not necessarily sparse even though  $\mathbf{A}$  is.
- If there are infinitely many solutions to (2), then it is common to choose the one with least 2-norm.
- If columns of  $\mathbf{A}$  are linearly independent, then  $\mathbf{A}^t \cdot \mathbf{A}$  is invertible and (2) has a unique solution.
- A solution to (2), there may be infinitely many, is often undesirable (overfitting).

# Reconstruction methods

## Kaczmarz method

Problem:  $A$  is huge and cannot be stored.

Stefan Kaczmarz (1895–1939), Polish mathematician active at University of Lviv (now in Ukraine).

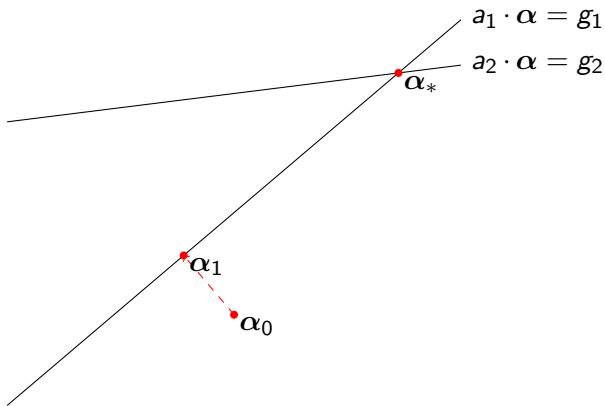


**Kaczmarz method:** Solving linear systems of equations without needing to store the matrix.

- Introduced by Stefan Kaczmarz in 1937.
- Rediscovered in 1970 by Richard Gordon, Robert Bender, and Gabor Herman, then under the name algebraic reconstruction technique (ART).

# Reconstruction methods

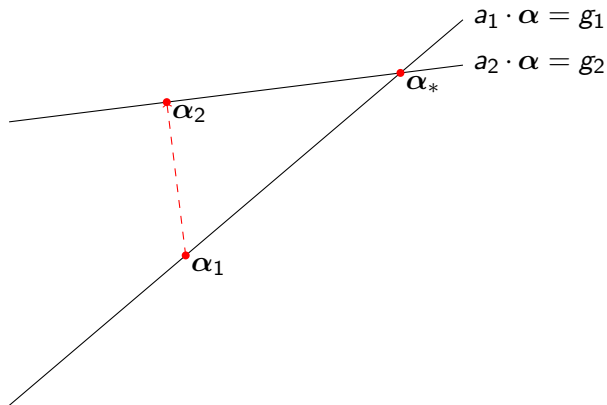
Kaczmarz method: The case  $n = m = 2$



$\alpha_1 :=$  projection of  $\alpha_0$  into hyperplane  $a_1 \cdot \alpha = g_1$ .

# Reconstruction methods

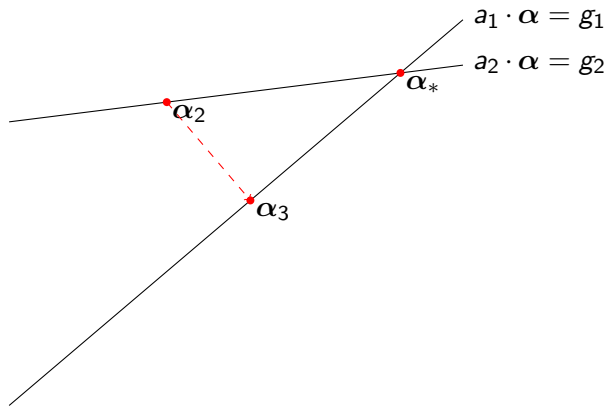
Kaczmarz method: The case  $n = m = 2$



$\alpha_2 :=$  projection of  $\alpha_1$  into hyperplane  $a_2 \cdot \alpha = g_2$ .

# Reconstruction methods

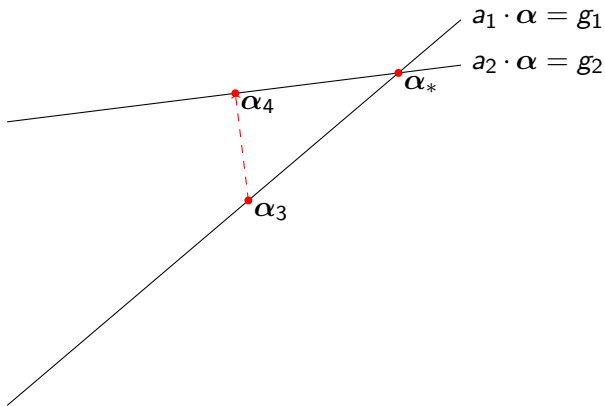
Kaczmarz method: The case  $n = m = 2$



$\alpha_3 :=$  projection of  $\alpha_2$  into hyperplane  $a_1 \cdot \alpha = g_1$ .

# Reconstruction methods

Kaczmarz method: The case  $n = m = 2$

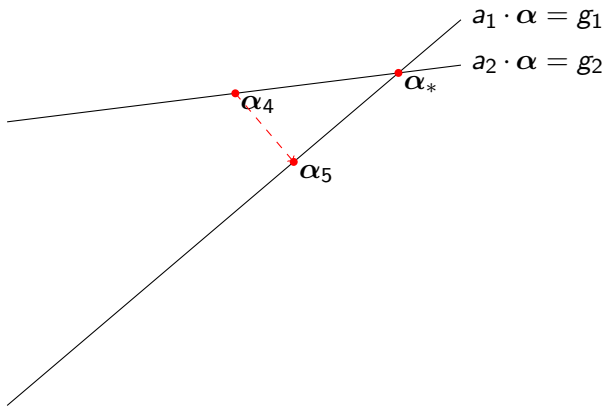


$\alpha_4 :=$  projection of  $\alpha_3$  into hyperplane  $a_2 \cdot \alpha = g_2$ .



# Reconstruction methods

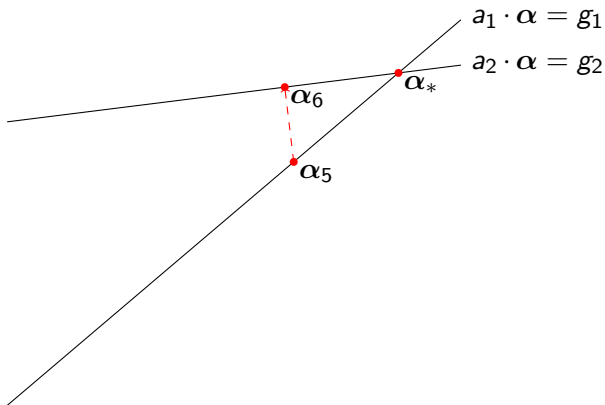
Kaczmarz method: The case  $n = m = 2$



$\alpha_5 :=$  projection of  $\alpha_4$  into hyperplane  $a_1 \cdot \alpha = g_1$ .

# Reconstruction methods

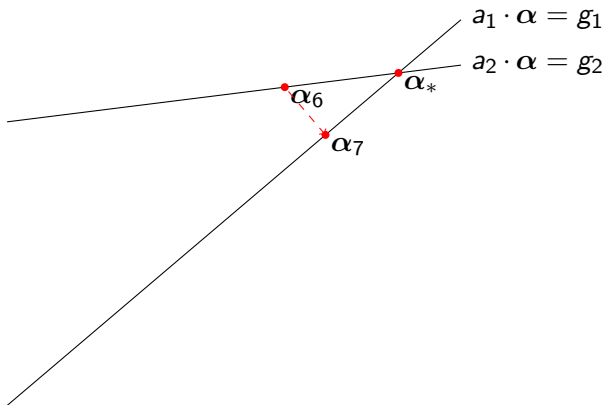
Kaczmarz method: The case  $n = m = 2$



$\alpha_6 :=$  projection of  $\alpha_5$  into hyperplane  $a_2 \cdot \alpha = g_2$ .

# Reconstruction methods

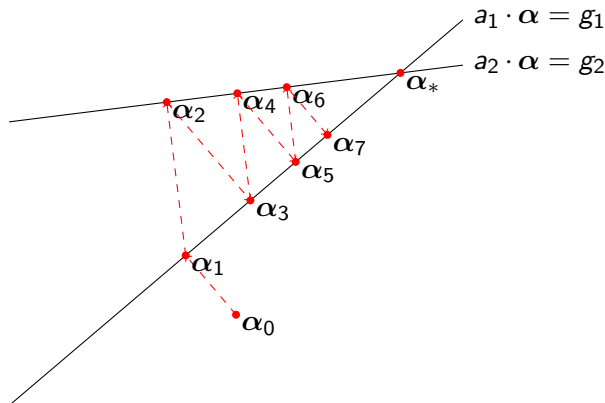
Kaczmarz method: The case  $n = m = 2$



$\alpha_7 :=$  projection of  $\alpha_6$  into hyperplane  $a_1 \cdot \alpha = g_1$ .

# Reconstruction methods

Kaczmarz method: The case  $n = m = 2$

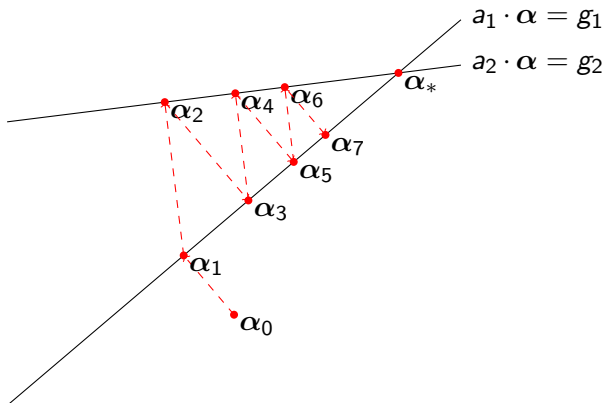


**Iterative scheme:** If  $\pi_i(\alpha)$  denotes the projection of  $\alpha$  into the  $i$ :th hyperplane  $a_i \cdot \alpha = g_i$  ( $i = 1, \dots, m$ ), then

$$\alpha_{k+1} := \pi_i(\alpha_k) \quad \text{where } k \text{ sweeps through } 1, \dots, m.$$

# Reconstruction methods

Kaczmarz method: The case  $n = m = 2$



Iterative scheme:

$$\alpha_{k+1} := \alpha_k + \frac{g_i - a_i \cdot \alpha_k}{\|a_i\|_2^2} a_i, \quad \text{where } k \text{ sweeps through } 1, \dots, m.$$

# Reconstruction methods

## Kaczmarz method: Proof of iterative scheme

Orthogonal projection of a vector into a hyperplane.

- The vector  $a_i$  is orthogonal to the hyperplane  $a_i \cdot \alpha = g_i$ .

*The equation  $a_i \cdot \alpha = g_i$  is the scalar equation of a hyperplane with  $a_i$  as the normal, so  $a_i$  is orthogonal to all vectors in the hyperplane  $a_i \cdot \alpha = g_i$ .*

- The orthogonal projection  $\pi(x)$  of a vector  $x \in \mathbb{R}^n$  onto  $a_i \cdot \alpha = g_i$  is found by subtracting a multiple of  $a_i$  from  $x$ :

$$\pi(x) = x - \gamma a_i \quad \text{for some } \gamma \in \mathbb{R}.$$

- $\gamma$  must satisfy

$$g_i = \pi(x) \cdot a_i = (x - \gamma a_i) \cdot a_i = x \cdot a_i - \gamma a_i \cdot a_i.$$

- Solving this equation for  $\gamma$  gives

$$\gamma = \frac{a_i \cdot \alpha - g_i}{a_i \cdot a_i} \implies \pi(x) = x + \frac{g_i - a_i \cdot \alpha}{\|a_i\|_2^2} a_i$$

# Reconstruction methods

## Kaczmarz method: properties

- **Convergence:** let  $\alpha_k$  denotes the iterates generated by the Kaczmarz method for solving  $\mathbf{A} \cdot \alpha = g$ .
  - If the linear system has a unique solution, then Kaczmarz iterates converge towards this solution.
  - For overdetermined systems, Kaczmarz method with initial vector  $\alpha_0 = \mathbf{0}$  converges to the least-squares solution.
  - For underdetermined systems,  $\alpha_k$  converges to the least-squares solution closest to the initial vector  $\alpha_0$  (Tanabe, 1971).
- **Convergence rate:** faster convergence when the angle between the hyperplanes is large. Extreme case is when hyperplanes are orthogonal, convergence in  $m + 1$  steps.
- **Speed-up:** often not worth doing Gram-Schmidt orthogonalization. Instead, most efficient approach is to project on random hyperplanes.
- Need a balance between accurately computing  $\mathbf{A}$  and the inconsistencies that result from crude approximations.

# Examples of reconstructions

- Simulated parallel beam data of 2D phantom (true image).
- Relative error (in %)  $:= 100 \cdot \frac{\|\alpha - \alpha_{\text{true}}\|_2}{\|\alpha_{\text{true}}\|_2}$



# Reconstruction methods

Kaczmarz method: example of 2D reconstruction

## Simulation protocol

- **Phantom:**  $256 \times 256$  pixel 2D Shepp-Logan  
 $n = 256 \cdot 256 = 65\,536$
- **Data:** Full angular range  $[0^\circ, 180^\circ]$  with  $1^\circ$  step (180 directions) and 400 detector elements.  
 $m = 180 \cdot 400 = 72\,000$ .
- **Noise component in data:** Additive Gaussian noise with relative noise level 5%.

Overdetermined problem since  $m > n$ .

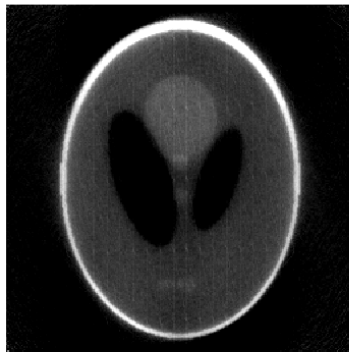
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 1 iteration(s), error = 40.8%



# of iterations = 1  
Relative error = 40.8%

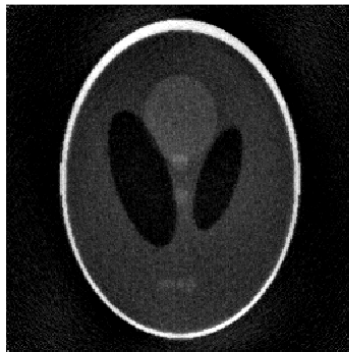
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 2 iteration(s), error = 31.7%



# of iterations = 2  
Relative error = 31.7%

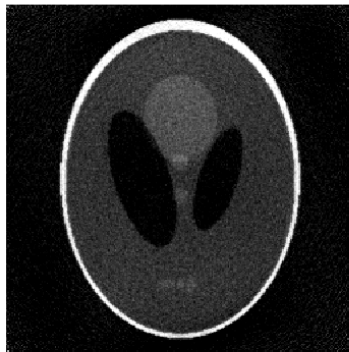
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 3 iteration(s), error = 29.7%



# of iterations = 3  
Relative error = 29.7%

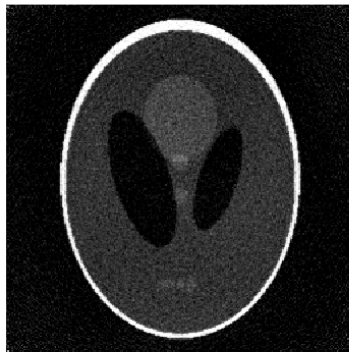
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 4 iteration(s), error = 30.3%



# of iterations = 4  
Relative error = 30.3%

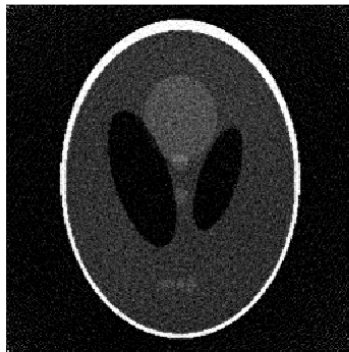
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 5 iteration(s), error = 31.7%



# of iterations = 5  
Relative error = 31.7%

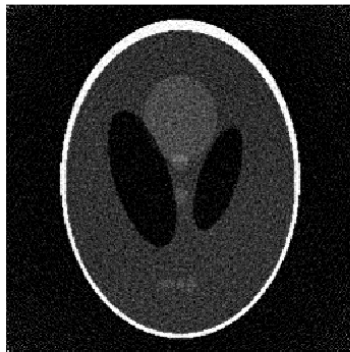
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 6 iteration(s), error = 33.2%



# of iterations = 6  
Relative error = 33.2%

# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 7 iteration(s), error = 34.6%



# of iterations = 7  
Relative error = 34.6%



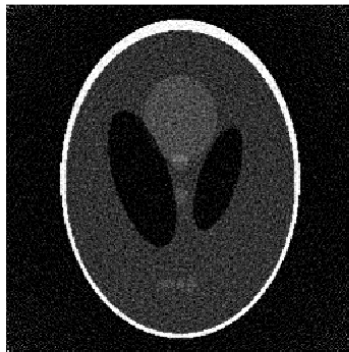
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 8 iteration(s), error = 35.8%



# of iterations = 8  
Relative error = 35.8%

# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 9 iteration(s), error = 36.9%



# of iterations = 9  
Relative error = 36.9%

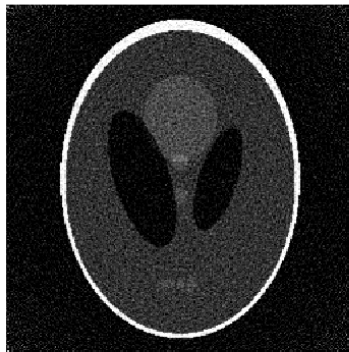
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 10 iteration(s), error = 37.9%



# of iterations = 10  
Relative error = 37.9%

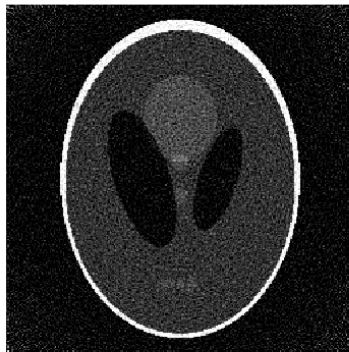
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 20 iteration(s), error = 44.8%



# of iterations = 20  
Relative error = 44.8%

# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 30 iteration(s), error = 49.5%



# of iterations = 30  
Relative error = 49.5%

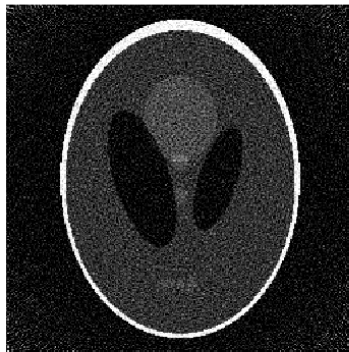
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 40 iteration(s), error = 53.4%



# of iterations = 40  
Relative error = 53.4%

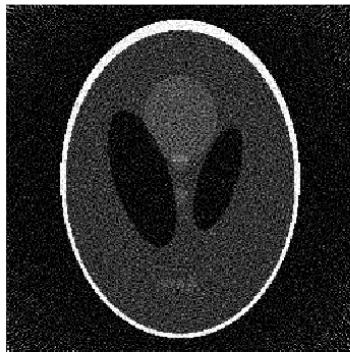
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 50 iteration(s), error = 57.1%



# of iterations = 50  
Relative error = 57.1%

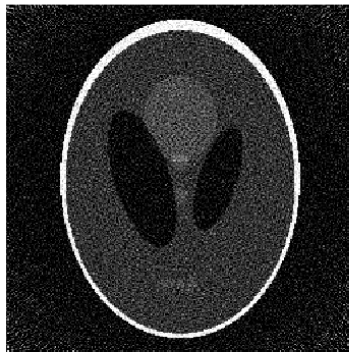
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 60 iteration(s), error = 60.4%



# of iterations = 60  
Relative error = 60.4%



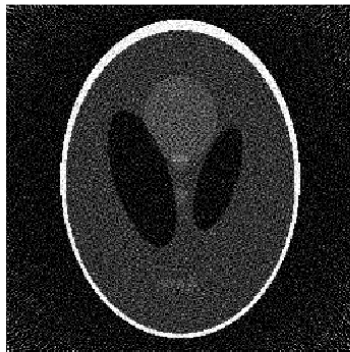
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 70 iteration(s), error = 63.6%



# of iterations = 70  
Relative error = 63.6%

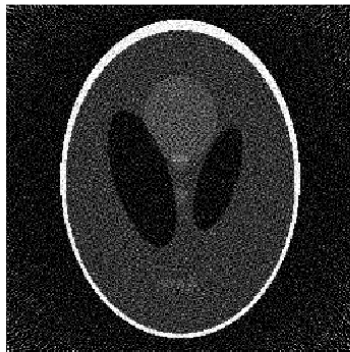
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 80 iteration(s), error = 66.7%



# of iterations = 80  
Relative error = 66.7%

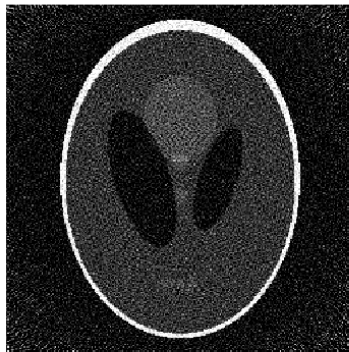
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 90 iteration(s), error = 69.7%



# of iterations = 90  
Relative error = 69.7%

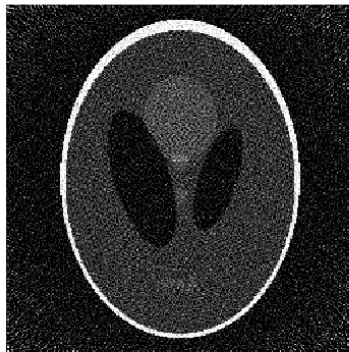
# Reconstruction methods

Kaczmarz method: impact of number of iterations

Exact phantom



Kaczmarz reconstruction: 100 iteration(s), error = 72.5%



# of iterations = 100  
Relative error = 72.5%

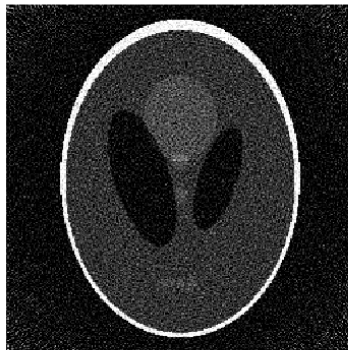
# Reconstruction methods

Kaczmarz method: impact of enforcing non-negativity

Exact phantom



Kaczmarz: 50 iterations, error = 57.1%



# of iterations = 50  
Non-negativity not enforced  
Relative error = 57.1%

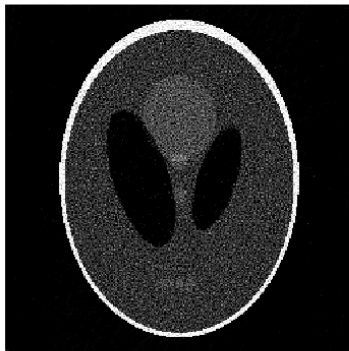
# Reconstruction methods

Kaczmarz method: impact of enforcing non-negativity

Exact phantom



Kaczmarz with non-negativity: 50 iterations, error = 33.9%



# of iterations = 50  
Non-negativity enforced  
Relative error = 33.9%

# Reconstruction methods

Kaczmarz method: impact of enforcing non-negativity

## Simulation protocol

- **Phantom:**  $256 \times 256$  pixel 2D Shepp-Logan  
 $n = 256 \cdot 256 = 65\,536$
- **Data:** Full angular range  $[0^\circ, 180^\circ]$  with  $5^\circ$  step (36 directions) and 256 detector elements.  
 $m = 36 \cdot 256 = 9216$ .
- **Noise component in data:** Additive Gaussian noise with relative noise level 5%.

Underdetermined problem since  $m < n$ .

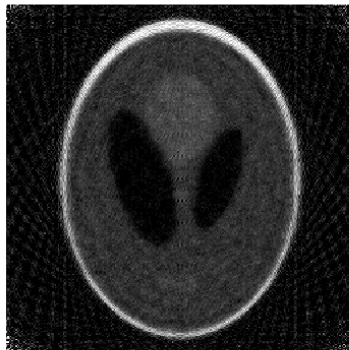
# Reconstruction methods

Kaczmarz method: impact of enforcing non-negativity

Exact phantom



Kaczmarz: 50 iterations, error = 56.7%



# of iterations = 50  
Non-negativity not enforced  
Relative error = 56.7%



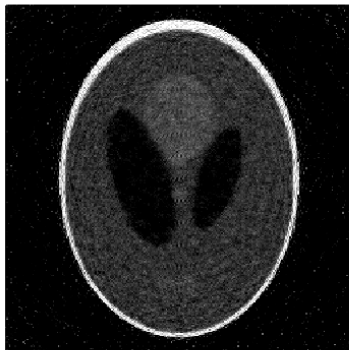
# Reconstruction methods

Kaczmarz method: impact of enforcing non-negativity

Exact phantom



Kaczmarz with non-negativity: 50 iterations, error = 35%



# of iterations = 50  
Non-negativity enforced  
Relative error = 35.0%

# Reconstruction methods

Kaczmarz method: regularised variants

- **Constraints:** Enforce bound constraints, such as positivity, by projection techniques.
- **Semi-convergence:** Reconstruction improves during the first few iterates, then it begins to deteriorate (semi-convergence). In medical imaging only a few iterations are used.  
For ART: Empirical observation, no theoretical backing:

- T. Elfving, P. C. Hansen, and T. Nikazad. *Semi-convergence properties of Kaczmarz's method*, Inverse Problems, vol. 30, no. 5 (055007), 2014.

Stopping rule: Rule for choosing the number of iterations.

- **Regularisation by relaxation:** Modification of iterates

$$\alpha_{k+1} := \alpha_k + \lambda_k \frac{g_i - a_i \cdot \alpha_k}{\|a_i\|_2^2} a_i \quad \text{where } i = k \bmod m + 1.$$

$0 < \lambda_k < 2$  is a relaxation parameter.

**Regularisation parameters:** Number of iterates and relaxation parameter.

# Reconstruction methods

Kaczmarz method: regularised variants

- **Constraints:** Enforce bound constraints, such as positivity, by projection techniques.
- **Semi-convergence:** Reconstruction improves during the first few iterates, then it begins to deteriorate (semi-convergence). In medical imaging only a few iterations are used.  
For ART: Empirical observation, no theoretical backing:

- T. Elfving, P. C. Hansen, and T. Nikazad. *Semi-convergence properties of Kaczmarz's method*, Inverse Problems, vol. 30, no. 5 (055007), 2014.

Stopping rule: Rule for choosing the number of iterations.

- **Regularisation by relaxation:** Modification of iterates

$$\alpha_{k+1} := \alpha_k + \lambda_k \frac{g_i - a_i \cdot \alpha_k}{\|a_i\|_2^2} a_i \quad \text{where } i = k \bmod m + 1.$$

$0 < \lambda_k < 2$  is a relaxation parameter.

**Regularisation parameters:** Number of iterates and relaxation parameter.

# Reconstruction methods

Kaczmarz method: regularised variants

- **Constraints:** Enforce bound constraints, such as positivity, by projection techniques.
- **Semi-convergence:** Reconstruction improves during the first few iterates, then it begins to deteriorate (semi-convergence). In medical imaging only a few iterations are used.  
For ART: Empirical observation, no theoretical backing:

- T. Elfving, P. C. Hansen, and T. Nikazad. *Semi-convergence properties of Kaczmarz's method*, Inverse Problems, vol. 30, no. 5 (055007), 2014.

Stopping rule: Rule for choosing the number of iterations.

- **Regularisation by relaxation:** Modification of iterates

$$\alpha_{k+1} := \alpha_k + \lambda_k \frac{g_i - a_i \cdot \alpha_k}{\|a_i\|_2^2} a_i \quad \text{where } i = k \bmod m + 1.$$

$0 < \lambda_k < 2$  is a relaxation parameter.

Regularisation parameters: Number of iterates and relaxation parameter.

# Reconstruction methods

Kaczmarz method: regularised variants

- **Constraints:** Enforce bound constraints, such as positivity, by projection techniques.
- **Semi-convergence:** Reconstruction improves during the first few iterates, then it begins to deteriorate (semi-convergence). In medical imaging only a few iterations are used.  
For ART: Empirical observation, no theoretical backing:

- T. Elfving, P. C. Hansen, and T. Nikazad. *Semi-convergence properties of Kaczmarz's method*, Inverse Problems, vol. 30, no. 5 (055007), 2014.

Stopping rule: Rule for choosing the number of iterations.

- **Regularisation by relaxation:** Modification of iterates

$$\alpha_{k+1} := \alpha_k + \lambda_k \frac{g_i - a_i \cdot \alpha_k}{\|a_i\|_2^2} a_i \quad \text{where } i = k \bmod m + 1.$$

$0 < \lambda_k < 2$  is a relaxation parameter.

**Regularisation parameters:** Number of iterates and relaxation parameter.

# Iterative reconstruction methods

## Row-action methods

- Algebraic reconstruction technique (ART)

- Based on vector multiplications: A single iterative update makes use of a single row of  $\mathbf{A}$ .
- Good semi-convergence observed (no theory explaining this).

Algorithms: Kaczmarz's method and variants of it.

- Simultaneous iterative reconstruction technique (SIRT)

- Based on matrix multiplications: A single iterative update uses all the rows of  $\mathbf{A}$  simultaneously.
- Slower semi-convergence, but otherwise good understanding of convergence theory.

Algorithms: Landweber, Cimmino, CAV, DROP, and SART.

- Krylov subspace methods: Class of iterative methods based on matrix multiplications where iterates are given as

$$\alpha_k = \alpha_{k-1} + \mathbf{K}^{-1} \cdot (g - \mathbf{A} \cdot \alpha_{k-1}).$$

$\mathbf{K}$  is here a simple invertible matrix that is “close” to  $\mathbf{A}$ .

Algorithms: CGLS, LSQR, GMRES, ...

# Iterative reconstruction methods

## Row-action methods

- Algebraic reconstruction technique (ART)

- Based on vector multiplications: A single iterative update makes use of a single row of  $\mathbf{A}$ .
- Good semi-convergence observed (no theory explaining this).

Algorithms: Kaczmarz's method and variants of it.

- Simultaneous iterative reconstruction technique (SIRT)

- Based on matrix multiplications: A single iterative update uses all the rows of  $\mathbf{A}$  simultaneously.
- Slower semi-convergence, but otherwise good understanding of convergence theory.

Algorithms: Landweber, Cimmino, CAV, DROP, and SART.

- Krylov subspace methods: Class of iterative methods based on matrix multiplications where iterates are given as

$$\alpha_k = \alpha_{k-1} + \mathbf{K}^{-1} \cdot (g - \mathbf{A} \cdot \alpha_{k-1}).$$

$\mathbf{K}$  is here a simple invertible matrix that is “close” to  $\mathbf{A}$ .

Algorithms: CGLS, LSQR, GMRES, ...

# Iterative reconstruction methods

## Row-action methods

- Algebraic reconstruction technique (ART)

- Based on vector multiplications: A single iterative update makes use of a single row of  $\mathbf{A}$ .
- Good semi-convergence observed (no theory explaining this).

Algorithms: Kaczmarz's method and variants of it.

- Simultaneous iterative reconstruction technique (SIRT)

- Based on matrix multiplications: A single iterative update uses all the rows of  $\mathbf{A}$  simultaneously.
- Slower semi-convergence, but otherwise good understanding of convergence theory.

Algorithms: Landweber, Cimmino, CAV, DROP, and SART.

- Krylov subspace methods: Class of iterative methods based on matrix multiplications where iterates are given as

$$\alpha_k = \alpha_{k-1} + \mathbf{K}^{-1} \cdot (g - \mathbf{A} \cdot \alpha_{k-1}).$$

$\mathbf{K}$  is here a simple invertible matrix that is “close” to  $\mathbf{A}$ .

Algorithms: CGLS, LSQR, GMRES, ...



# Iterative reconstruction methods

## ART

**Typical step:** Update of iterate involves a single row of  $\mathbf{A}$ :

$$\alpha_{k+1} := \alpha_k + \lambda_k \frac{g_{i_k} - a_{i_k} \cdot \alpha_k}{\|a_{i_k}\|_2^2} a_{i_k} \quad \text{where } k = 1, 2, \dots$$

and  $i_k \in \{1, \dots, m\}$  is given by the row ordering.

- **Relaxation parameter:**  $0 < \lambda_k < 2$ , setting  $\lambda_k = 1$  implies projecting  $\alpha_k$  onto hyperplane  $g_i = a_i \cdot \alpha$  (original un-regularised Kaczmarz's method).
- **Row ordering:** How  $i_k \in \{1, \dots, m\}$  depends on  $k$ , i.e., how the iterates sweep through the  $m$  rows  $a_1, \dots, a_m$ :
  - **Classical Kaczmarz:**  $i_k = 1, 2, \dots, m, 1, 2, \dots, m, \dots$
  - **Symmetric Kaczmarz:**  
 $i_k = 1, 2, \dots, m-1, m, m-1, \dots, 2, 1, \dots$
  - **Randomized Kaczmarz:** At each  $k$ , let  $i_k$  be the  $i$ :th row  $a_i$  randomly with probability proportional to the row norm  $\|a_i\|_2$ .

# Iterative reconstruction methods

## ART

- **Semi-convergence:** Not formally proved, only empirically observed
  - Fast initial convergence  $\implies$  method of choice when only a few iterations can be afforded.
  - After some initial iterations the convergence can be very slow.
- **Convergence rate:**
  - Estimates of convergence rates are based on quantities of **A** that are hard to compute and difficult to compare with convergence estimates of other iterative methods.
  - Need to exploit the analytic structure associated with the forward operator.
  - Rate of convergence depends on the ordering of the equations.

# Iterative reconstruction methods

## ART

- Choosing relaxation parameter

For  $\lambda_k = 1$  the high-frequency components (such as noise) show up early in the iteration, while overall features are determined later.

For  $\lambda_k \ll 1$ , say 0.1, the iterations first determine the smooth parts of the image and small details in later iterates

$\implies$  surprisingly small values of  $\lambda_k$  (e.g.,  $\lambda_k = 0.05$ ) are quite common in ART.

- Stopping rules

- The discrepancy principle
- The L-curve
- Generalised cross-validation (GCV)
- Normalised cumulative periodogram (NCP)
- ...

# Iterative reconstruction methods

## Influence of relaxation

Phantom

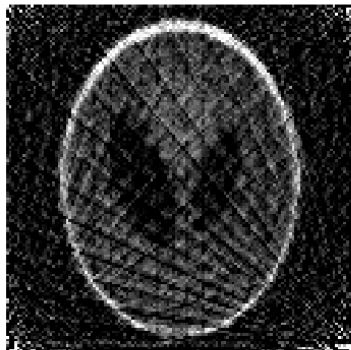


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 1$

Figure of merits  
Rel. error = 145%  
MSE = 0.126  
PSNR = 9

Reconstructions using classical Kaczmarz



$$\lambda_k = 1.0$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom

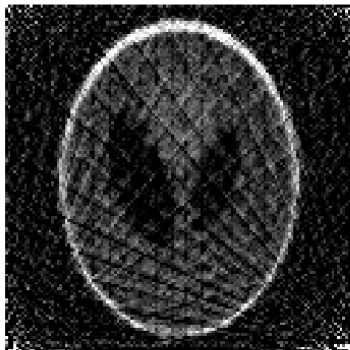


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 0.9$

Figure of merits  
Rel. error = 129%  
MSE = 0.0996  
PSNR = 10

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.9$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda_k = 0.8$

Figure of merits  
Rel. error = 115%  
MSE = 0.08  
PSNR = 11

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.8$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda_k = 0.7$

Figure of merits  
Rel. error = 104%  
MSE = 0.0648  
PSNR = 11.9

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.7$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom



Reconstructions using classical Kaczmarz



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 0.6$

Figure of merits  
Rel. error = 93.8%  
MSE = 0.0528  
PSNR = 12.8

$$\lambda_k = 0.6$$



# Iterative reconstruction methods

## Influence of relaxation

Phantom



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 0.5$

Figure of merits  
Rel. error = 84.5%  
MSE = 0.0428  
PSNR = 13.7

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.5$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 0.4$

Figure of merits  
Rel. error = 75.8%  
MSE = 0.0345  
PSNR = 14.6

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.4$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda_k = 0.3$

Figure of merits  
Rel. error = 68%  
MSE = 0.0277  
PSNR = 15.6

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.3$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom

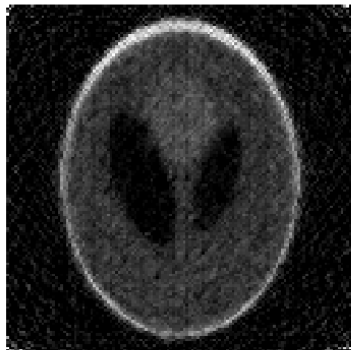


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda_k = 0.2$

Figure of merits  
Rel. error = 61.9%  
MSE = 0.023  
PSNR = 16.4

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.2$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom

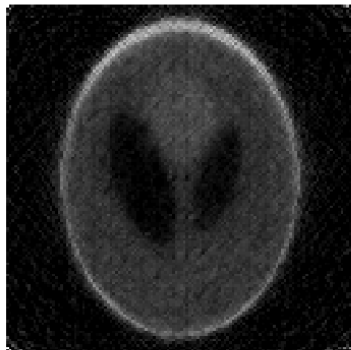


Problem size & noise  
n = 16384  
m = 5400  
noise level = 10%

Iterates: 5  
Non negativity: No  
Lambda = 0.1

Figure of merits  
Rel. error = 60.4%  
MSE = 0.0219  
PSNR = 16.6

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.1$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom

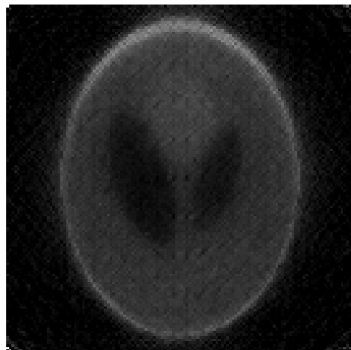


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 0.05$

Figure of merits  
Rel. error = 64.4%  
MSE = 0.0249  
PSNR = 16

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.05$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom

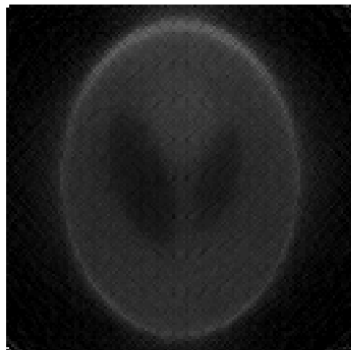


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 0.025$

Figure of merits  
Rel. error = 70%  
MSE = 0.0294  
PSNR = 15.3

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.025$$

# Iterative reconstruction methods

## Influence of relaxation

Phantom

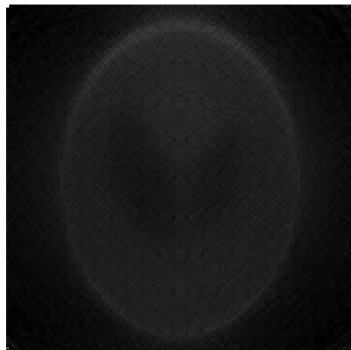


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\lambda = 0.01$

Figure of merits  
Rel. error = 77.7%  
MSE = 0.0363  
PSNR = 14.4

Reconstructions using classical Kaczmarz



$$\lambda_k = 0.01$$



# Iterative reconstruction methods

## Influence of row ordering

Phantom

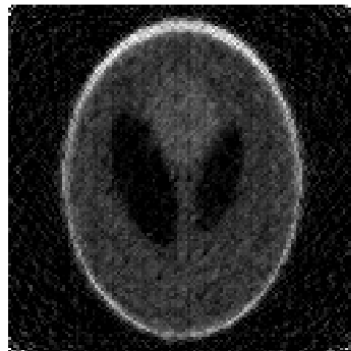


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\Lambda = 0.2$

Figure of merits  
Rel. error = 61.9%  
MSE = 0.023  
PSNR = 16.4

Reconstructions using classical Kaczmarz



Classical

# Iterative reconstruction methods

## Influence of row ordering

Phantom



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Iterates: 5  
Non negativity: No  
 $\text{Lambda} = 0.2$

Figure of merits  
Rel. error = 64.3%  
MSE = 0.0248  
PSNR = 16.1

Reconstructions using symmetric Kaczmarz



Symmetric

# Iterative reconstruction methods

## Influence of row ordering

Phantom

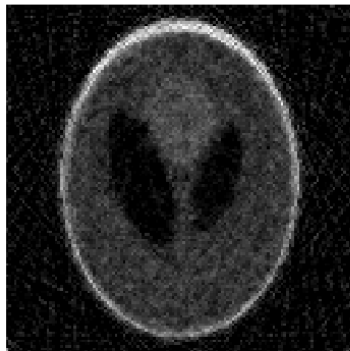


Problem size & noise  
 $n=16384$   
 $m=5400$   
noise level= 10%

Iterates: 5  
Non negativity: No  
 $\Lambda=0.2$

Figure of merits  
Rel. error= 51.5%  
MSE= 0.0159  
PSNR= 18

Reconstructions using randomized Kaczmarz



Random

# Iterative reconstruction methods

## SIRT

Least-squares problem for the reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) \quad \text{where} \quad Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2. \quad (3)$$

Gradient-descent method: Solve (3) by the iterative scheme

$$\alpha_{k+1} := \alpha_k - \lambda_{k+1} \nabla Q(\alpha_k) \quad \text{for } k = 0, 1, 2, \dots$$

- The gradient of the quadratic form  $Q: \mathbb{R}^n \rightarrow \mathbb{R}_+$  is

$$\nabla Q(\alpha) = \mathbf{A}^t \cdot (\mathbf{A} \cdot \alpha - g).$$

- Value of step size  $\lambda_k$  may change at every iteration.
- Convergence to a local minima of (3) can be guaranteed for certain methods for choosing  $\lambda_k$ .
- Serves as a basis for simultaneous iterative reconstruction technique (SIRT) methods.

# Iterative reconstruction methods

## SIRT

Least-squares problem for the reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) \quad \text{where} \quad Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2. \quad (3)$$

**Gradient-descent method:** Solve (3) by the iterative scheme

$$\alpha_{k+1} := \alpha_k - \lambda_{k+1} \nabla Q(\alpha_k) \quad \text{for } k = 0, 1, 2, \dots$$

- The gradient of the quadratic form  $Q: \mathbb{R}^n \rightarrow \mathbb{R}_+$  is

$$\nabla Q(\alpha) = \mathbf{A}^t \cdot (\mathbf{A} \cdot \alpha - g).$$

- Value of step size  $\lambda_k$  may change at every iteration.
- Convergence to a local minima of (3) can be guaranteed for certain methods for choosing  $\lambda_k$ .
- Serves as a basis for simultaneous iterative reconstruction technique (SIRT) methods.

# Iterative reconstruction methods

## SIRT

General form of iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha_k - g) \quad \text{for } k = 0, 1, 2, \dots$$

Iterates are stopped early and  $\lambda_k > 0$  is a relaxation parameter.

Methods vary depending on choices of the diagonal  $(n \times n)$ -matrix  $\mathbf{T}$  and the diagonal  $(m \times m)$ -matrix  $\mathbf{M}$ .

- **Classical Landweber:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{I}_m$ .
- **Cimmino:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}$ .
- **CAV:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}_S$ .
- **DROP:**  $\mathbf{T} = \text{diag}(1/s_j)$  and  $\mathbf{M} = m\mathbf{D}$ .
- **SART:**  $\mathbf{T} = \text{diag}\left(1/\sum_{i=1}^m a_{i,j}\right)$  and  $\mathbf{M} = \text{diag}\left(1/\sum_{j=1}^n a_{i,j}\right)$ .

# Iterative reconstruction methods

## SIRT

General form of iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha_k - g) \quad \text{for } k = 0, 1, 2, \dots$$

Iterates are stopped early and  $\lambda_k > 0$  is a relaxation parameter.

Methods vary depending on choices of the diagonal  $(n \times n)$ -matrix  $\mathbf{T}$  and the diagonal  $(m \times m)$ -matrix  $\mathbf{M}$ .

- Classical Landweber:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{I}_m$ .
- Cimmino:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}$ .
- CAV:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}_S$ .
- DROP:  $\mathbf{T} = \text{diag}(1/s_j)$  and  $\mathbf{M} = m\mathbf{D}$ .
- SART:  $\mathbf{T} = \text{diag}\left(1/\sum_{i=1}^m a_{i,j}\right)$  and  $\mathbf{M} = \text{diag}\left(1/\sum_{j=1}^n a_{i,j}\right)$ .

Here

$$\mathbf{D} = \frac{1}{m} \text{diag}\left(\frac{1}{\|a_i\|_2^2}\right)$$

# Iterative reconstruction methods

## SIRT

General form of iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha_k - g) \quad \text{for } k = 0, 1, 2, \dots$$

Iterates are stopped early and  $\lambda_k > 0$  is a relaxation parameter.

Methods vary depending on choices of the diagonal  $(n \times n)$ -matrix  $\mathbf{T}$  and the diagonal  $(m \times m)$ -matrix  $\mathbf{M}$ .

- Classical Landweber:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{I}_m$ .
- Cimmino:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}$ .
- CAV:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}_S$ .
- DROP:  $\mathbf{T} = \text{diag}(1/s_j)$  and  $\mathbf{M} = m\mathbf{D}$ .
- SART:  $\mathbf{T} = \text{diag}\left(1/\sum_{i=1}^m a_{i,j}\right)$  and  $\mathbf{M} = \text{diag}\left(1/\sum_{j=1}^n a_{i,j}\right)$ .

Here

$$\mathbf{D} = \frac{1}{m} \text{diag}\left(\frac{1}{\|a_i\|_2^2}\right) \quad \text{and} \quad \mathbf{D}_S := \text{diag}\left(\frac{1}{\sum_{j=1}^n s_j a_{i,j}^2}\right)$$

$s_j$  = number of non-zero elements in  $j$ :th column  $\mathbf{A}$



# Iterative reconstruction methods

## SIRT

General form of iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha_k - g) \quad \text{for } k = 0, 1, 2, \dots$$

Iterates are stopped early and  $\lambda_k > 0$  is a relaxation parameter.

Methods vary depending on choices of the diagonal  $(n \times n)$ -matrix  $\mathbf{T}$  and the diagonal  $(m \times m)$ -matrix  $\mathbf{M}$ .

- Classical Landweber:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{I}_m$ .
- Cimmino:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}$ .
- CAV:  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}_S$ .
- DROP:  $\mathbf{T} = \text{diag}(1/s_j)$  and  $\mathbf{M} = m\mathbf{D}$ .
- SART:  $\mathbf{T} = \text{diag}\left(1/\sum_{i=1}^m a_{i,j}\right)$  and  $\mathbf{M} = \text{diag}\left(1/\sum_{j=1}^n a_{i,j}\right)$ .

Here

$$\mathbf{D} = \frac{1}{m} \text{diag}\left(\frac{1}{\|a_i\|_2^2}\right)$$

$s_j =$  number of non-zero elements in  $j$ :th column  $\mathbf{A}$

# Iterative reconstruction methods

## SIRT

General form of iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha_k - g) \quad \text{for } k = 0, 1, 2, \dots$$

Iterates are stopped early and  $\lambda_k > 0$  is a relaxation parameter.

Methods vary depending on choices of the diagonal  $(n \times n)$ -matrix  $\mathbf{T}$  and the diagonal  $(m \times m)$ -matrix  $\mathbf{M}$ .

- **Classical Landweber:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{I}_m$ .
- **Cimmino:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}$ .
- **CAV:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}_S$ .
- **DROP:**  $\mathbf{T} = \text{diag}(1/s_j)$  and  $\mathbf{M} = m\mathbf{D}$ .
- **SART:**  $\mathbf{T} = \text{diag}\left(1/\sum_{i=1}^m a_{i,j}\right)$  and  $\mathbf{M} = \text{diag}\left(1/\sum_{j=1}^n a_{i,j}\right)$ .

# Iterative reconstruction methods

## SIRT

General form of iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha_k - g) \quad \text{for } k = 0, 1, 2, \dots$$

Iterates are stopped early and  $\lambda_k > 0$  is a relaxation parameter.

Methods vary depending on choices of the diagonal  $(n \times n)$ -matrix  $\mathbf{T}$  and the diagonal  $(m \times m)$ -matrix  $\mathbf{M}$ .

- **Classical Landweber:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{I}_m$ .
- **Cimmino:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}$ .
- **CAV:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}_S$ .
- **DROP:**  $\mathbf{T} = \text{diag}(1/s_j)$  and  $\mathbf{M} = m\mathbf{D}$ .
- **SART:**  $\mathbf{T} = \text{diag}\left(1/\sum_{i=1}^m a_{i,j}\right)$  and  $\mathbf{M} = \text{diag}\left(1/\sum_{j=1}^n a_{i,j}\right)$ .

**Note:**  $\mathbf{T}$  and  $\mathbf{M}$  are diagonal matrices  $\Rightarrow$  can be stored.

# Iterative reconstruction methods

## SIRT

General form of iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha_k - g) \quad \text{for } k = 0, 1, 2, \dots$$

Iterates are stopped early and  $\lambda_k > 0$  is a relaxation parameter.

Methods vary depending on choices of the diagonal  $(n \times n)$ -matrix  $\mathbf{T}$  and the diagonal  $(m \times m)$ -matrix  $\mathbf{M}$ .

- **Classical Landweber:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{I}_m$ .
- **Cimmino:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}$ .
- **CAV:**  $\mathbf{T} = \mathbf{I}_n$  and  $\mathbf{M} = \mathbf{D}_S$ .
- **DROP:**  $\mathbf{T} = \text{diag}(1/s_j)$  and  $\mathbf{M} = m\mathbf{D}$ .
- **SART:**  $\mathbf{T} = \text{diag}\left(1/\sum_{i=1}^m a_{i,j}\right)$  and  $\mathbf{M} = \text{diag}\left(1/\sum_{j=1}^n a_{i,j}\right)$ .

Several stopping rules and approaches to choose the relaxation parameter  $\lambda_k$ .

# Iterative reconstruction methods

SIRT-: Classical Landweber

Iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \cdot \mathbf{A}^t \cdot (\mathbf{A} \cdot \alpha - g) = \alpha_k - \lambda_k \nabla Q(\alpha_k)$$

where  $Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2$ .

- Each step in Landweber's method is a step in the direction of steepest descent.
- Semi-convergence property well-established.
- $0 < \lambda_k < 2/\sigma^2$  where  $\sigma$  is an estimate of the largest singular value of  $\mathbf{A}$ .

# Iterative reconstruction methods

SIRT: Cimmino

Iterates:

$$\alpha_{k+1} := \alpha_k - \lambda_k \cdot \mathbf{A}^t \cdot \mathbf{D} \cdot (\mathbf{A} \cdot \alpha - g) \quad \text{for } k = 0, 1, 2, \dots$$

where  $\mathbf{D} = \frac{1}{m} \text{diag}\left(\frac{1}{\|a_i\|_2^2}\right)$ , so

$$\alpha_{k+1} := \alpha_k + \frac{\lambda_k}{m} \sum_{i=1}^m \frac{g_i - a_i \cdot \alpha_k}{\|a_i\|_2^2} a_i.$$

- Each step in Cimmino's method is the average of the projections onto the  $m$  hyperplanes  $a_i \cdot \alpha = g_i$  with  $i = 1, \dots, m$  (compare with ART).

# Iterative reconstruction methods

## Non-negativity and box constraints

Assume one knows beforehand that  $\alpha_{\text{true}} \in C \subset \mathbb{R}^n$  where  $C$  is a known convex set. One can incorporate such a constraint into ART and SIRT iterations by projecting iterates onto  $C$ :

- Projected ART

$$\alpha_{k+1} := \mathcal{P}_C \left( \alpha_k + \lambda_k \frac{g_{i_k} - a_{i_k} \cdot \alpha_k}{\|a_{i_k}\|_2^2} a_{i_k} \right)$$

with  $i_k \in \{1, \dots, m\}$  given by the row-ordering scheme.

- Projected SIRT

$$\alpha_{k+1} := \mathcal{P}_C \left( \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha - g) \right)$$

# Iterative reconstruction methods

## Non-negativity and box constraints

Assume one knows beforehand that  $\alpha_{\text{true}} \in C \subset \mathbb{R}^n$  where  $C$  is a known convex set. One can incorporate such a constraint into ART and SIRT iterations by projecting iterates onto  $C$ :

- Projected ART

$$\alpha_{k+1} := \mathcal{P}_C \left( \alpha_k + \lambda_k \frac{g_{i_k} - a_{i_k} \cdot \alpha_k}{\|a_{i_k}\|_2^2} a_{i_k} \right)$$

with  $i_k \in \{1, \dots, m\}$  given by the row-ordering scheme.

- Projected SIRT

$$\alpha_{k+1} := \mathcal{P}_C \left( \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha - g) \right)$$

$C$  can represent box-constraints  $a \leq \alpha_i \leq b$  in which case

$$\mathcal{P}_C(\alpha)_i = \begin{cases} \alpha_i & \text{if } a < \alpha_i < b \\ a & \text{if } \alpha_i \leq a \\ b & \text{if } \alpha_i \geq b \end{cases} \quad \text{for } i = 1, \dots, n.$$



# Iterative reconstruction methods

## Non-negativity and box constraints

Assume one knows beforehand that  $\alpha_{\text{true}} \in C \subset \mathbb{R}^n$  where  $C$  is a known convex set. One can incorporate such a constraint into ART and SIRT iterations by projecting iterates onto  $C$ :

- Projected ART

$$\alpha_{k+1} := \mathcal{P}_C \left( \alpha_k + \lambda_k \frac{g_{i_k} - a_{i_k} \cdot \alpha_k}{\|a_{i_k}\|_2^2} a_{i_k} \right)$$

with  $i_k \in \{1, \dots, m\}$  given by the row-ordering scheme.

- Projected SIRT

$$\alpha_{k+1} := \mathcal{P}_C \left( \alpha_k - \lambda_k \mathbf{T} \cdot \mathbf{A}^t \cdot \mathbf{M} \cdot (\mathbf{A} \cdot \alpha - g) \right)$$

Projected iterates should converge to a least-squares solution in  $C$ , i.e., to a solution of

$$\min_{\alpha \in C} \|\mathbf{A} \cdot \alpha - g\|_2^2.$$

Projected SIRT converges to a solution of the above problem.

# Iterative reconstruction methods

SIRT: examples without enforcing non-negativity

Phantom

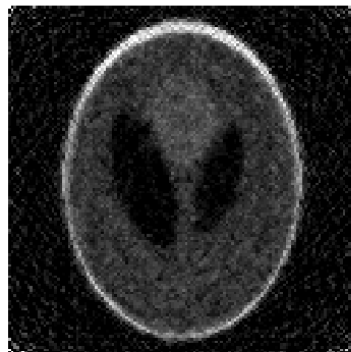


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: No  
 $\lambda = 0.2$

Figure of merits  
Rel. error = 58%  
MSE = 0.0202  
PSNR = 16.9

Randomized Kaczmarz with NCP stop rule



Randomized ART

All iterative schemes make use of the same stopping rule.

# Iterative reconstruction methods

SIRT: examples without enforcing non-negativity

Phantom

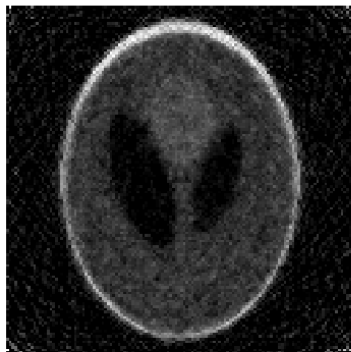


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: No

Figure of merits  
Rel. error = 52.3%  
MSE = 0.0164  
PSNR = 17.8

SART with NCP stop rule & psi 1mod relaxation



SART

All iterative schemes make use of the same stopping rule.

# Iterative reconstruction methods

SIRT: examples without enforcing non-negativity

Phantom



Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: No

Figure of merits  
Rel. error = 59.4%  
MSE = 0.0212  
PSNR = 16.7

CAV with NCP stop rule & psi1 mod relaxation



CAV

All iterative schemes make use of the same stopping rule.

# Iterative reconstruction methods

SIRT: examples without enforcing non-negativity

Phantom

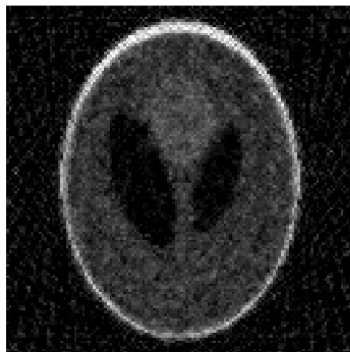


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: No  
 $\text{Lambda} = 0.00027$

Figure of merits  
Rel. error = 49.9%  
MSE = 0.0149  
PSNR = 18.3

Landweber with NCP stop rule



Landweber

All iterative schemes make use of the same stopping rule.

# Iterative reconstruction methods

SIRT: examples enforcing non-negativity

Phantom

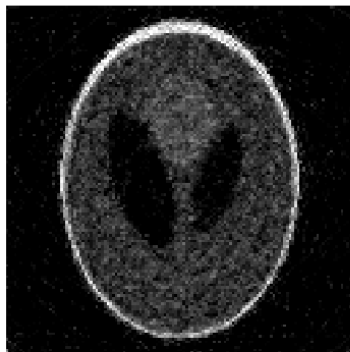


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: Yes  
 $\Lambda = 0.2$

Figure of merits  
Rel. error = 40.6%  
MSE = 0.00987  
PSNR = 20.1

Randomized Kaczmarz with NCP stop rule



Randomized ART

All iterative schemes make use of the same stopping rule.

# Iterative reconstruction methods

SIRT: examples enforcing non-negativity

Phantom

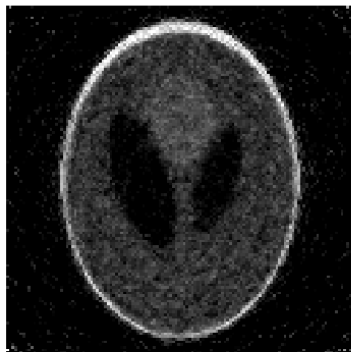


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: Yes

Figure of merits  
Rel. error = 40.9%  
MSE = 0.01  
PSNR = 20

SART with NCP stop rule & psi 1mod relaxation



SART

All iterative schemes make use of the same stopping rule.

# Iterative reconstruction methods

SIRT: examples enforcing non-negativity

Phantom

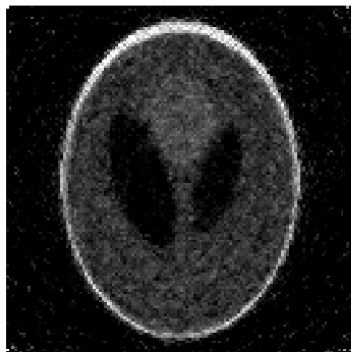


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: Yes

Figure of merits  
Rel. error = 42.9%  
MSE = 0.0111  
PSNR = 19.6

CAV with NCP stop rule & psi1 mod relaxation



CAV

All iterative schemes make use of the same stopping rule.



# Iterative reconstruction methods

SIRT: examples enforcing non-negativity

Phantom

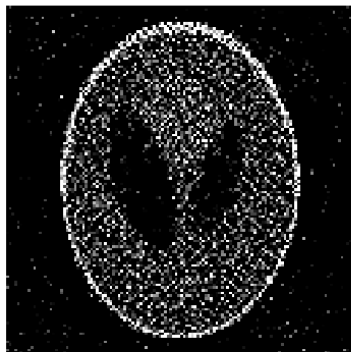


Problem size & noise  
 $n = 16384$   
 $m = 5400$   
noise level = 10%

Non negativity: Yes  
 $\text{Lambda} = 0.00027$

Figure of merits  
Rel. error = 106%  
MSE = 0.0679  
PSNR = 11.7

Landweber with NCP stop rule



Landweber

All iterative schemes make use of the same stopping rule.

# The conjugate gradient (CG) method

## The basic algorithm

A Krylov subspace method – iterative scheme for finding a least-squares solution to the reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) \quad \text{where} \quad Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2. \quad (4)$$

Basic CG algorithm for minimising a non-linear function  $Q$

- 1:  $\alpha_0$  arbitrary,  $r_0 := \nabla Q(\alpha_0)$ ,  $d_0 := -r_0$ .
- 2: **for**  $k := 1, 2, \dots$  **do**
- 3:      $\alpha_k :=$  minima of  $Q$  on half-line  $t \mapsto \alpha_{k-1} + td_{k-1}$
- 4:      $r_k := \nabla Q(\alpha_k)$
- 5:      $\beta_k := \|r_k\|_2^2 / \|r_{k-1}\|_2^2$
- 6:      $d_k := -r_k + \beta_k d_{k-1}$
- 7: **end for**

# The conjugate gradient (CG) method

## The basic algorithm

A Krylov subspace method – iterative scheme for finding a least-squares solution to the reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) \quad \text{where} \quad Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2. \quad (4)$$

Basic CG algorithm for minimising a non-linear function  $Q$

- 1:  $\alpha_0$  arbitrary,  $r_0 := \nabla Q(\alpha_0)$ ,  $d_0 := -r_0$ .
- 2: **for**  $k := 1, 2, \dots$  **do**
- 3:      $\alpha_k :=$  minima of  $Q$  on half-line  $t \mapsto \alpha_{k-1} + td_{k-1}$
- 4:      $r_k := \nabla Q(\alpha_k)$
- 5:      $\beta_k := \|r_k\|_2^2 / \|r_{k-1}\|_2^2$
- 6:      $d_k := -r_k + \beta_k d_{k-1}$
- 7: **end for**

Gradient of  $Q$ :  $\nabla Q(\alpha) = \mathbf{A}^t \cdot (\mathbf{A} \cdot \alpha - g)$ .

# The conjugate gradient (CG) method

## The basic algorithm

A Krylov subspace method – iterative scheme for finding a least-squares solution to the reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) \quad \text{where} \quad Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2. \quad (4)$$

## CGLS algorithm for solving (4)

- 1:  $\alpha_0$  arbitrary,  $r_0 := g - \mathbf{A} \cdot \alpha_0$ ,  $d_0 := \mathbf{A}^t \cdot r_0$ .
- 2: **for**  $k := 1, 2, \dots$  **do**
- 3:      $\tau_k := \|\mathbf{A}^t \cdot r_{k-1}\|_2^2 / \|\mathbf{A} \cdot d_{k-1}\|_2^2$
- 4:      $\alpha_k := \alpha_{k-1} + \tau_k d_{k-1}$
- 5:      $r_k := r_{k-1} - \tau_k \mathbf{A} \cdot d_{k-1}$
- 6:      $\beta_k := \|\mathbf{A}^t \cdot r_k\|_2^2 / \|\mathbf{A}^t \cdot r_{k-1}\|_2^2$
- 7:      $d_k := \mathbf{A}^t \cdot r_k + \beta_k d_{k-1}$
- 8: **end for**

# The conjugate gradient (CG) method

## The basic algorithm

A Krylov subspace method – iterative scheme for finding a least-squares solution to the reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) \quad \text{where} \quad Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2. \quad (4)$$

### CGLS algorithm for solving (4)

- 1:  $\alpha_0$  arbitrary,  $r_0 := g - \mathbf{A} \cdot \alpha_0$ ,  $d_0 := \mathbf{A}^t \cdot r_0$ .
- 2: **for**  $k := 1, 2, \dots$  **do**
- 3:      $\tau_k := \|\mathbf{A}^t \cdot r_{k-1}\|_2^2 / \|\mathbf{A} \cdot d_{k-1}\|_2^2$
- 4:      $\alpha_k := \alpha_{k-1} + \tau_k d_{k-1}$
- 5:      $r_k := r_{k-1} - \tau_k \mathbf{A} \cdot d_{k-1}$
- 6:      $\beta_k := \|\mathbf{A}^t \cdot r_k\|_2^2 / \|\mathbf{A}^t \cdot r_{k-1}\|_2^2$
- 7:      $d_k := \mathbf{A}^t \cdot r_k + \beta_k d_{k-1}$
- 8: **end for**

All operations involve multiplication with  $\mathbf{A}$  or  $\mathbf{A}^t$ .

# The conjugate gradient (CG) method

## The basic algorithm

A Krylov subspace method – iterative scheme for finding a least-squares solution to the reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^n} Q(\alpha) \quad \text{where} \quad Q(\alpha) := \frac{1}{2} \|\mathbf{A} \cdot \alpha - g\|_2^2. \quad (4)$$

## CGLS algorithm for solving (4)

- 1:  $\alpha_0$  arbitrary,  $r_0 := g - \mathbf{A} \cdot \alpha_0$ ,  $d_0 := \mathbf{A}^t \cdot r_0$ .
- 2: **for**  $k := 1, 2, \dots$  **do**
- 3:      $\tau_k := \|\mathbf{A}^t \cdot r_{k-1}\|_2^2 / \|\mathbf{A} \cdot d_{k-1}\|_2^2$
- 4:      $\alpha_k := \alpha_{k-1} + \tau_k d_{k-1}$
- 5:      $r_k := r_{k-1} - \tau_k \mathbf{A} \cdot d_{k-1}$
- 6:      $\beta_k := \|\mathbf{A}^t \cdot r_k\|_2^2 / \|\mathbf{A}^t \cdot r_{k-1}\|_2^2$
- 7:      $d_k := \mathbf{A}^t \cdot r_k + \beta_k d_{k-1}$
- 8: **end for**

Need to regularise by early stopping.

# Choosing the regularisation parameter(s)

Recover a least-squares solution to the ill-posed problem

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}.$$

- ART and SIRT methods involve two regularisation parameters, the relaxation parameter and the number of iterates.
- CG methods have one regularisation parameter, the number of iterates.

# Choosing the regularisation parameter(s)

Recover a least-squares solution to the ill-posed problem

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}.$$

- ART and SIRT methods involve two regularisation parameters, the relaxation parameter and the number of iterates.
- CG methods have one regularisation parameter, the number of iterates.

Two types of errors in regularisation of ill-posed problems:

**Noise (perturbation) error:** error in iterates due to “inverting” the noise  $g_{\text{noise}}$  in data.

**Iteration (regularisation) error:** errors in iterates that are due to semi-convergence (errors get amplified one iterates progress too far).

Both noise and iteration errors are always present in a regularised solution.

$\implies$  their size depends on the regularisation parameter(s).



# Choosing the regularisation parameter(s)

Recover a least-squares solution to the ill-posed problem

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}.$$

- ART and SIRT methods involve two regularisation parameters, the relaxation parameter and the number of iterates.
- CG methods have one regularisation parameter, the number of iterates.

For iterative methods:

- The choice of the relaxation parameter mainly seeks to limit the noise error.
- The choice of number of iterates (stopping rule) mainly seeks to limit the iteration error.

# Choosing the regularisation parameter(s)

The relaxation parameter

- **ART and SIRT methods:** Possible to estimate the noise and iteration errors for a fixed relaxation parameter. Albeit pessimistic, these estimates correctly describe the evolution of these errors as iterations progress.
- Choice of relaxation parameter  $\lambda_k \implies$  limit the noise error

One possible strategy: Choose  $\lambda_0 = \lambda_1 = \sqrt{2}/\sigma^2$  and

$$\lambda_k = \frac{2}{\sigma^2}(1 - \zeta_k) \quad \text{or} \quad \lambda_k = \frac{2}{\sigma^2} \frac{1 - \zeta_k}{(1 - \zeta_k^k)^2} \quad \text{for } k = 2, 3, \dots$$

Here,  $\sigma$  is an estimate of the largest singular value of  $\mathbf{A}$  and  $0 < \zeta_k < 1$  is the unique root of the polynomial

$$p_{k-1}(\zeta) := (2k-1)\zeta^{k-1} - (\zeta^{k-2} + \dots + \zeta + 1).$$

Leads to diminishing step-size.

# Choosing the regularisation parameter(s)

The relaxation parameter

- **ART and SIRT methods:** Possible to estimate the noise and iteration errors for a fixed relaxation parameter. Albeit pessimistic, these estimates correctly describe the evolution of these errors as iterations progress.
- Choice of relaxation parameter  $\lambda_k \implies$  limit the noise error

**One possible strategy:** Choose  $\lambda_0 = \lambda_1 = \sqrt{2}/\sigma^2$  and

$$\lambda_k = \frac{2}{\sigma^2}(1 - \zeta_k) \quad \text{or} \quad \lambda_k = \frac{2}{\sigma^2} \frac{1 - \zeta_k}{(1 - \zeta_k^k)^2} \quad \text{for } k = 2, 3, \dots$$

Here,  $\sigma$  is an estimate of the largest singular value of  $\mathbf{A}$  and  $0 < \zeta_k < 1$  is the unique root of the polynomial

$$p_{k-1}(\zeta) := (2k - 1)\zeta^{k-1} - (\zeta^{k-2} + \dots + \zeta + 1).$$

Leads to diminishing step-size.

# Choosing the Regularisation Parameter

## Stopping rules

- Common to stop the iterations in iterative methods when the residual norm  $\|\mathbf{A} \cdot \alpha_k - g\|_2$  is “sufficiently small” since this may imply that  $\alpha_k$  is close to a least-squares solution.
- Not good for ill-posed problems since a least-squares solution is probably useless as it has overfitting artefacts.
- Stopping rules regulate the iteration error.

# Choosing the Regularisation Parameter

Stopping rules: The discrepancy criterion

**Principle:** Assume the residual norm decreases monotonically with the iterates. Then, stop iterates when the difference to data is smaller than size of the data noise.

- An estimate  $\delta > 0$  of the size of the noise component in data, i.e.,  $\|g_{\text{noise}}\|_2 < \delta$ .
- $\tau > 1$ , a safety factor.

Find  $k$  such that

$$\|\mathbf{A} \cdot \alpha_{k+1} - g\|_2 \leq \tau \delta \leq \|\mathbf{A} \cdot \alpha_k - g\|_2.$$

Properties:

- Unique solution for regularisation parameter  $k$  since residual norm varies monotonically with the iterates.
- Relies on a good estimate  $\delta$  of the size of the noise in data, which may be difficult to obtain in practice.
- Computed regularisation parameter  $k$  is very sensitive to the accuracy of the estimate  $\delta$ . A too small estimate can lead to dramatic under-smoothing (because  $k$  is chosen too large).

# Choosing the Regularisation Parameter

Stopping rules: The discrepancy criterion

**Principle:** Assume the residual norm decreases monotonically with the iterates. Then, stop iterates when the difference to data is smaller than size of the data noise.

- An estimate  $\delta > 0$  of the size of the noise component in data, i.e.,  $\|g_{\text{noise}}\|_2 < \delta$ .
- $\tau > 1$ , a safety factor.

Find  $k$  such that

$$\|\mathbf{A} \cdot \alpha_{k+1} - g\|_2 \leq \tau \delta \leq \|\mathbf{A} \cdot \alpha_k - g\|_2.$$

Properties:

- Unique solution for regularisation parameter  $k$  since residual norm varies monotonically with the iterates.
- Relies on a good estimate  $\delta$  of the size of the noise in data, which may be difficult to obtain in practice.
- Computed regularisation parameter  $k$  is very sensitive to the accuracy of the estimate  $\delta$ . A too small estimate can lead to dramatic under-smoothing (because  $k$  is chosen too large).

# Choosing the Regularisation Parameter

Stopping rules: The discrepancy criterion

**Principle:** Assume the residual norm decreases monotonically with the iterates. Then, stop iterates when the difference to data is smaller than size of the data noise.

- An estimate  $\delta > 0$  of the size of the noise component in data, i.e.,  $\|g_{\text{noise}}\|_2 < \delta$ .
- $\tau > 1$ , a safety factor.

Find  $k$  such that

$$\|\mathbf{A} \cdot \alpha_{k+1} - g\|_2 \leq \tau \delta \leq \|\mathbf{A} \cdot \alpha_k - g\|_2.$$

**Properties:**

- Unique solution for regularisation parameter  $k$  since residual norm varies monotonically with the iterates.
- Relies on a good estimate  $\delta$  of the size of the noise in data, which may be difficult to obtain in practice.
- Computed regularisation parameter  $k$  is very sensitive to the accuracy of the estimate  $\delta$ . A too small estimate can lead to dramatic under-smoothing (because  $k$  is chosen too large).

# Choosing the Regularisation Parameter

Stopping rules: The L-curve criterion

**Principle:** For small  $k$  smaller than some threshold, the iteration error dominates in  $\alpha_k$  so the 2-norm  $\|\alpha_k\|_2$  is expected to be small while the residual norm  $\|\mathbf{A} \cdot \alpha_k - g\|_2$  is large.

- $\|\alpha_k\|_2$  is almost a constant given by  $\|\alpha_{\text{true}}\|_2$  except for very small  $k$  where  $\|\alpha_k\|_2$  gets smaller as  $k \rightarrow 0$ .
- The residual norm  $\|\mathbf{A} \cdot \alpha_k - g\|_2$  increases as  $k \rightarrow 0$ , until it reaches its maximum value at  $k = 0$ .

For  $k$  larger than some threshold the noise error dominates  $\alpha_k$  leading to the following  $k$  dependency:

- $\|\alpha_k\|_2$  increases as  $k$  increases (overfitting).
- The residual norm  $\|\mathbf{A} \cdot \alpha_k - g\|_2$  stays almost constant at the noise level in data.



# Choosing the Regularisation Parameter

Stopping rules: The L-curve criterion

The curve

$$k \mapsto \left( \|\alpha_k\|_2, \|\mathbf{A} \cdot \alpha_k - g\|_2 \right)$$

is “L”-formed with two distinctly different parts:

- Part where it is quite flat (when the noise error dominates)
- Part that is more vertical (when the iteration error dominates)

A log-log scale emphasizes the different characteristics of these two parts leading to the definition of the L-curve:

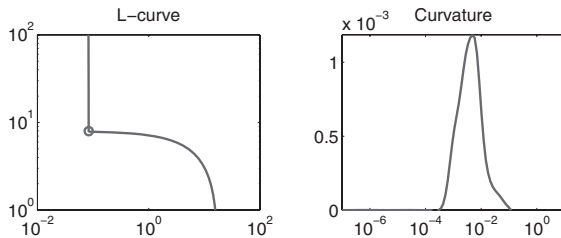
The L-curve:

$$k \mapsto \left( \log(\|\alpha_k\|_2), \log(\|\mathbf{A} \cdot \alpha_k - g\|_2) \right)$$

# Choosing the Regularisation Parameter

Stopping rules: The L-curve criterion

**L-curve criterion:** Choose  $k$  that corresponds to the corner (point with highest curvature) of the L-curve.



**Properties:**

- Heuristic criteria with no guarantee that it will always produce a good regularisation parameter.
- Typically fails when the change in the residual and solution norms is small for two consecutive values of  $k$ .
- Computing the corner can be challenging, there may be many small local corners.

# Choosing the Regularisation Parameter

Stopping rules: the generalised cross-validation criterion

**Principle:** Remove data and select the value for the regularisation parameter that minimises the error in predicting the removed data. Consider the reduced problem:

$$\mathbf{A}^i \cdot \boldsymbol{\alpha}_k^i = g^i \quad \text{for fixed } i = 1, \dots, m.$$

- $g^i \in \mathbb{R}^{m-1}$  is the data after we leave out the  $i$ :th data point.
- $\mathbf{A}^i$  is the  $\mathbf{A}$  with the  $i$ :th row left out
- $\boldsymbol{\alpha}_k^i \in \mathbb{R}^n$  is the reconstruction obtained after  $k$  iterates when solving the reduced problem above.

Use  $\boldsymbol{\alpha}_k^i \in \mathbb{R}^n$  and the  $i$ :th row  $a_i$  of  $\mathbf{A}$  to predict  $i$ :th data element:

$$g_i^{\text{predict}} := a_i \cdot \boldsymbol{\alpha}_k^i.$$

$$\text{Prediction error} = |g_i^{\text{predict}} - g_i| = |a_i \cdot \boldsymbol{\alpha}_k^i - g_i|.$$

# Choosing the Regularisation Parameter

Stopping rules: the generalised cross-validation criterion

**Principle:** Remove data and select the value for the regularisation parameter that minimises the error in predicting the removed data. Consider the reduced problem:

$$\mathbf{A}^i \cdot \boldsymbol{\alpha}_k^i = \mathbf{g}^i \quad \text{for fixed } i = 1, \dots, m.$$

- $\mathbf{g}^i \in \mathbb{R}^{m-1}$  is the data after we leave out the  $i$ :th data point.
- $\mathbf{A}^i$  is the  $\mathbf{A}$  with the  $i$ :th row left out
- $\boldsymbol{\alpha}_k^i \in \mathbb{R}^n$  is the reconstruction obtained after  $k$  iterates when solving the reduced problem above.

Choose the regularisation parameter  $k$  (number of iterations) such that the total prediction error is minimised, i.e., solve

$$\min_k \sum_{i=1}^m \left( g_i^{\text{predict}} - g_i \right)^2 = \min_k \sum_{i=1}^m \left( a_i \cdot \boldsymbol{\alpha}_k^i - g_i \right)^2.$$

Computationally unfeasible since  $m$  different reconstruction problems are involved  $\implies$  need to simplify above minimisation.

# Choosing the Regularisation Parameter

Stopping rules: the generalised cross-validation criterion

**Principle:** Remove data and select the value for the regularisation parameter that minimises the error in predicting the removed data. Consider the reduced problem:

$$\mathbf{A}^i \cdot \alpha_k^i = g^i \quad \text{for fixed } i = 1, \dots, m.$$

- $g^i \in \mathbb{R}^{m-1}$  is the data after we leave out the  $i$ :th data point.
- $\mathbf{A}^i$  is the  $\mathbf{A}$  with the  $i$ :th row left out
- $\alpha_k^i \in \mathbb{R}^n$  is the reconstruction obtained after  $k$  iterates when solving the reduced problem above.

**Properties:**

- Quite robust and accurate, as long as the noise is white.
- Occasional failure of GCV is well understood, and it often reveals itself by the ridiculous under-smoothing it leads to.
- Statistical and asymptotic properties is very well understood.
- Computationally demanding.

# Methods for choosing the regularisation parameter

## Summary

- The discrepancy principle is a simple method that seeks to reveal when the residual vector is noise-only. It relies on a good estimate of the size of the noise in data which may be difficult to obtain in practice.
- The L-curve criterion is based on an intuitive heuristic and seeks to balance the two error components via inspection (manually or automated) of the L-curve. This method fails when the solution is very smooth.
- The GCV criterion seeks to minimise the prediction error, and it is often a very robust method – with occasional failure, often leading to ridiculous under-smoothing that reveals itself.

# Methods for choosing the regularisation parameter

## Summary

- The discrepancy principle is a simple method that seeks to reveal when the residual vector is noise-only. It relies on a good estimate of the size of the noise in data which may be difficult to obtain in practice.
- The L-curve criterion is based on an intuitive heuristic and seeks to balance the two error components via inspection (manually or automated) of the L-curve. This method fails when the solution is very smooth.
- The GCV criterion seeks to minimise the prediction error, and it is often a very robust method – with occasional failure, often leading to ridiculous under-smoothing that reveals itself.

# Methods for choosing the regularisation parameter

## Summary

- The discrepancy principle is a simple method that seeks to reveal when the residual vector is noise-only. It relies on a good estimate of the size of the noise in data which may be difficult to obtain in practice.
- The L-curve criterion is based on an intuitive heuristic and seeks to balance the two error components via inspection (manually or automated) of the L-curve. This method fails when the solution is very smooth.
- The GCV criterion seeks to minimise the prediction error, and it is often a very robust method – with occasional failure, often leading to ridiculous under-smoothing that reveals itself.



# Reconstruction methods

## General properties

- Imaging, in particular 3D tomography, often leads to solving large-scale linear inverse problems.
- A useful reconstruction method must avoid factorisation of the measurement matrix:
  - The main “building blocks” must be matrix-vector multiplications, avoiding any factorization of the measurement matrix.
  - Allow the user to select regularisation parameter(s) via a parameter-choice method that does not require solving the reconstruction problem from scratch for each new parameter.

# Reconstruction methods

## General properties

- Imaging, in particular 3D tomography, often leads to solving large-scale linear inverse problems.
- A useful reconstruction method must avoid factorisation of the measurement matrix:
  - The main “building blocks” must be matrix-vector multiplications, avoiding any factorization of the measurement matrix.
  - Allow the user to select regularisation parameter(s) via a parameter-choice method that does not require solving the reconstruction problem from scratch for each new parameter.

# Reconstruction methods

## General remarks

- Focus on a single application, or a specific and narrow class of applications; no reconstruction method is guaranteed to work for a broad class of problems.
- When implementing the reconstruction method, focus on modularity and clarity of the computer code; it is guaranteed that you need to go back and modify/expand the software at some point in time.
- Make sure you understand the performance of the implementation, including computing times, storage requirements, etc.

# Reconstruction methods

## General remarks

- When testing the reconstruction method, make sure to generate test problems that reflect as many aspects as possible of real, measured data.
- When testing, also make sure to model the noise as realistically as possible, and use realistic noise levels.
- Be aware of the concept of "inverse crime" (same ingredients are used to create synthetic data and to recover image):
  - ① As a "proof-of-concept" first use tests that commit inverse crime; if the reconstruction method does not work under such circumstances, it can never work.
  - ② Next, in order to check the robustness to model errors, test the reconstruction method without committing inverse crime.

# Reconstruction methods

## General remarks

- Carefully evaluate the regularised solutions; consider which characteristics are important, and use the appropriate measure of the error (the 2-norm between the exact and regularised solutions is not always the optimal measure).
- Using the same exact data, create many realizations of the noise and perform a systematic study the robustness of the reconstruction method. Use histograms or other tools to investigate if the distribution of the errors has an undesired tail.

# Iterative reconstruction methods

## Summary

Iterative methods produce a sequence of digital images in  $\mathbb{R}^n$

$$\alpha_0 \rightarrow \alpha_1 \rightarrow \alpha_2 \rightarrow \dots$$

### Important properties

- Iterates designed to converge to a least-squares solution of  $\mathbf{A} \cdot \alpha = g$ .
- Semi-convergence, so initial convergence towards  $\alpha_{\text{true}}$  followed by (slow) convergence to least-squares solution.  
 $\implies$  Iteration number is a regularisation parameter.

# Iterative reconstruction methods

## Summary

Iterative methods produce a sequence of digital images in  $\mathbb{R}^n$

$$\alpha_0 \rightarrow \alpha_1 \rightarrow \alpha_2 \rightarrow \dots$$

## Advantages

- Works with any linear forward problem.
- Only uses matrix-vector multiplications, so the matrix  $\mathbf{A}$  is only accessed via matrix-vector multiplications and not explicitly required and never altered.  
 $\implies$  Enough to have a “black box” software component for computing the action of  $\mathbf{A}$  and  $\mathbf{A}^t$ .
- Atomic operations in iterative methods (mat-vec product, norm) suited for high-performance computing.
- Often produce a natural sequence of regularised solutions; stop when the solution is “satisfactory” (parameter choice).