Iterative Image Reconstruction Methods for Non-Cartesian MRI

Jeff Fessler
EECS Department

Doug Noll
BME Department

The University of Michigan

ISMRM Workshop on Non-Cartesian MRI
Feb. 27, 2007

Acknowledgements: Brad Sutton

Outline

- MR image reconstruction problem description
- Overview of image reconstruction methods
- Model-based image reconstruction
- Regularization
- Iterations and computation (NUFFT etc.)
- Myths about iterative reconstruction
- Example for partial non-Cartesian k-space

Image reconstruction toolbox:
`http://www.eecs.umich.edu/~fessler`

Why Iterative Image Reconstruction?

- Statistical modeling may reduce noise
- Incorporate prior information, *e.g.*:
  - support constraints
  - (piecewise) smoothness
  - phase constraints
- No density compensation needed
- "Non-Fourier" physical effects such as field inhomogeneity
- Incorporation of coil sensitivity maps
- Improved results for under-sampled trajectories (?)
- ...

("Avoiding k-space interpolation" is not a compelling reason!)

Primary drawbacks of Iterative Methods

- Choosing regularization parameter(s)
- Algorithm speed

Introduction

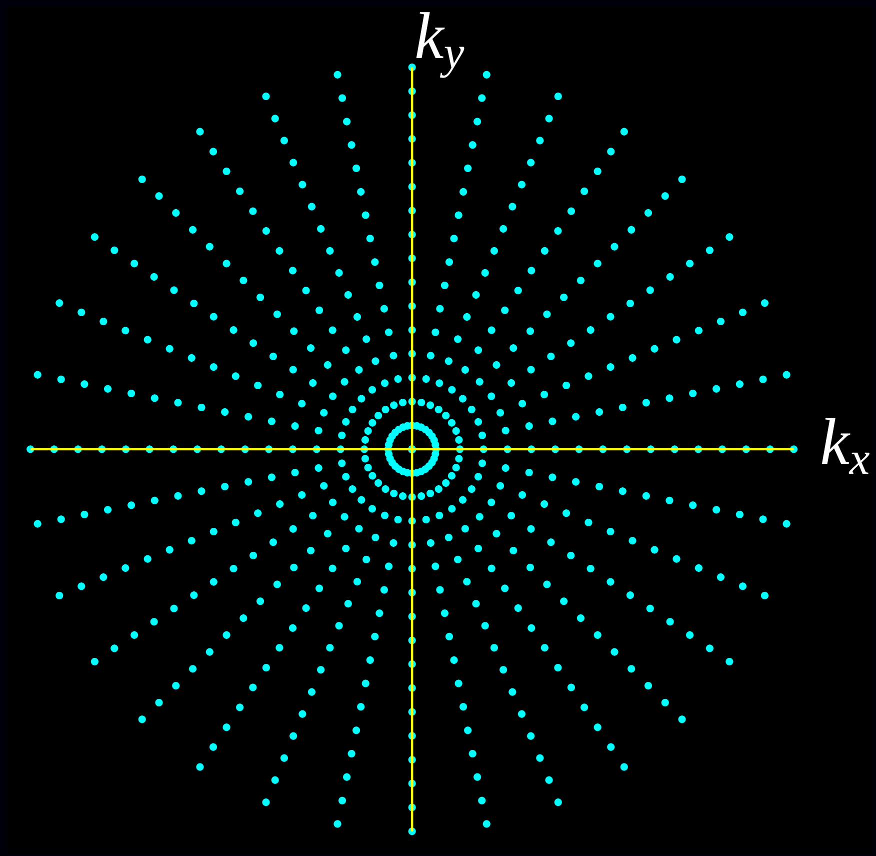
Non-Cartesian MR Image Reconstruction

"k-space" data

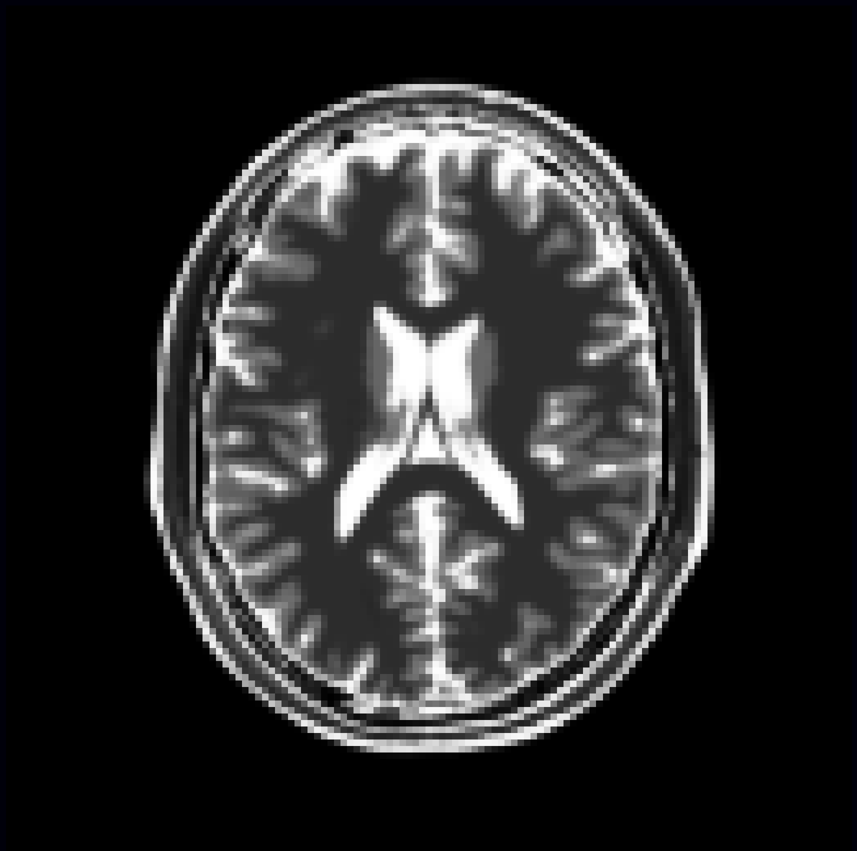
$\boldsymbol{y} = (y_1, \ldots, y_M)$

image

$f(\vec{r})$



$\Longrightarrow$



k-space trajectory:

$\vec{\kappa}(t) = (k_x(t), k_y(t))$

spatial coordinates:

$\vec{r} \in \mathbb{R}^2$

Textbook MRI Measurement Model

Ignoring *lots* of things, the standard measurement model is:

$$y_i = s(t_i) + \text{noise}_i, \qquad i = 1, \ldots, M$$

$$s(t) = \int f(\vec{r})\, e^{-\imath 2\pi \vec{\kappa}(t) \cdot \vec{r}}\, d\vec{r} = F(\vec{\kappa}(t))\,.$$

$\vec{r}$: spatial coordinates
$\vec{\kappa}(t)$: k-space trajectory
$f(\vec{r})$: object's unknown transverse magnetization
$F(\vec{\kappa})$: Fourier transform of $f(\vec{r})$

Goal of image reconstruction: find $f(\vec{r})$ from measurements $\{y_i\}_{i=1}^M$.

The unknown object $f(\vec{r})$ is a continuous-space function,
but the recorded measurements $\boldsymbol{y} = (y_1, \ldots, y_M)$ are finite.

Inherently under-determined (ill posed) problem
$\Longrightarrow$ no canonical solution.

All MR scans provide only "partial" k-space data.

Image Reconstruction Strategies

- Continuous-continuous formulation

  Pretend that a continuum of measurements are available:

$$F(\vec{\kappa}) = \int f(\vec{r})\, e^{-\imath 2\pi \vec{\kappa}\cdot\vec{r}}\, d\vec{r}.$$

  The “solution” is an inverse Fourier transform:

$$f(\vec{r}) = \int F(\vec{\kappa})\, e^{\imath 2\pi \vec{\kappa}\cdot\vec{r}}\, d\vec{\kappa}.$$

  Now discretize the integral solution:

$$\hat{f}(\vec{r}) = \sum_{i=1}^{M} F(\vec{\kappa}_i)\, e^{\imath 2\pi \vec{\kappa}_i\cdot\vec{r}}\, w_i \approx \sum_{i=1}^{M} y_i w_i\, e^{\imath 2\pi \vec{\kappa}_i\cdot\vec{r}},$$

  where w_i values are “sampling density compensation factors.”
  Numerous methods for choosing w_i value in the literature.

  For Cartesian sampling, using $w_i = 1/N$ suffices,
  and the summation is an inverse FFT.
  For non-Cartesian sampling, replace summation with gridding.

- Continuous-discrete formulation

  Use many-to-one linear model:

$$\boldsymbol{y} = \mathcal{A} f + \boldsymbol{\varepsilon}, \text{ where } \mathcal{A} : \mathcal{L}_2(\mathbb{R}^2) \to \mathbb{C}^M.$$

  Minimum norm solution (*cf.* “natural pixels”):

$$\min_{\hat{f}} \|\hat{f}\| \text{ subject to } \boldsymbol{y} = \mathcal{A}\hat{f}$$

$$\hat{f} = \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1}\boldsymbol{y} = \textstyle\sum_{i=1}^M c_i \, \mathrm{e}^{-\imath 2\pi \vec{\kappa}_i \cdot \vec{r}}, \text{ where } \mathcal{A}\mathcal{A}^*\boldsymbol{c} = \boldsymbol{y}.$$

- Discrete-discrete formulation

  Assume parametric model for object:

$$f(\vec{r}) = \sum_{j=1}^{N} f_j \, p_j(\vec{r}).$$

  Estimate parameter vector $\boldsymbol{f} = (f_1, \ldots, f_N)$ from data vector $\boldsymbol{y}$.

Model-Based Image Reconstruction: Details

Substitute series expansion of unknown object:

$$f(\vec{r}) = \sum_{j=1}^{N} f_j \, p(\vec{r} - \vec{r}_j) \longleftarrow \text{usually 2D rect functions}$$

into signal model $y_i = s(t_i) + \varepsilon_i$, where

$$\mathsf{E}[y_i] = s(t_i) = \int f(\vec{r}) \, \mathrm{e}^{-\imath 2\pi \vec{\kappa}_i \cdot \vec{r}} \, \mathrm{d}\vec{r},$$

yields:

$$\mathsf{E}[y_i] = \int \left[\sum_{j=1}^{N} f_j \, p(\vec{r} - \vec{r}_j) \right] \mathrm{e}^{-\imath 2\pi \vec{\kappa}_i \cdot \vec{r}} \, \mathrm{d}\vec{r} = \sum_{j=1}^{N} \left[\int p(\vec{r} - \vec{r}_j) \, \mathrm{e}^{-\imath 2\pi \vec{\kappa}_i \cdot \vec{r}} \, \mathrm{d}\vec{r} \right] f_j$$

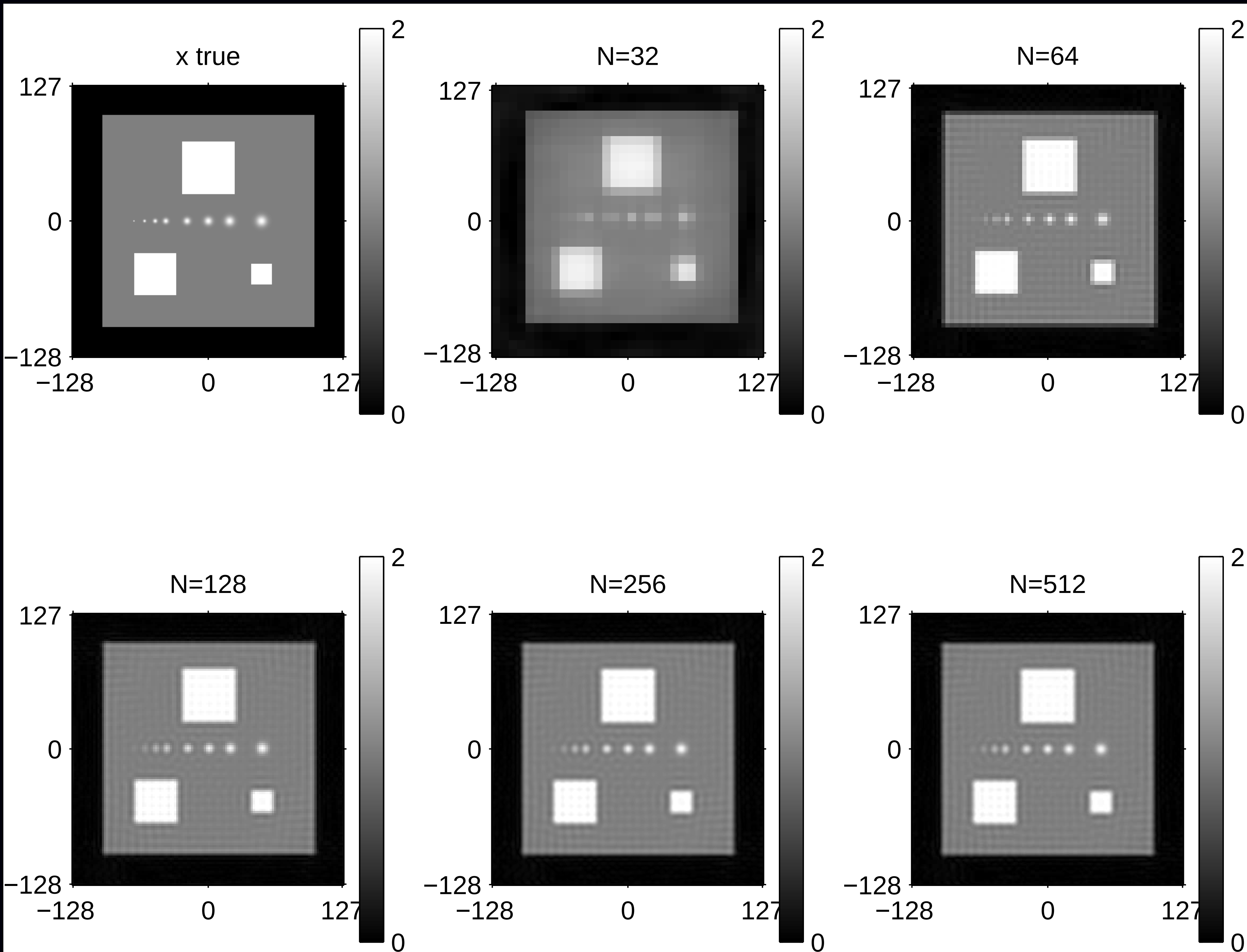
$$= \sum_{j=1}^{N} a_{ij} f_j, \qquad a_{ij} = P(\vec{\kappa}_i) \, \mathrm{e}^{-\imath 2\pi \vec{\kappa}_i \cdot \vec{r}_j}, \qquad p(\vec{r}) \overset{\text{FT}}{\Longleftrightarrow} P(\vec{\kappa}).$$

Discrete-discrete measurement model with system matrix $\boldsymbol{A} = \{a_{ij}\}$:

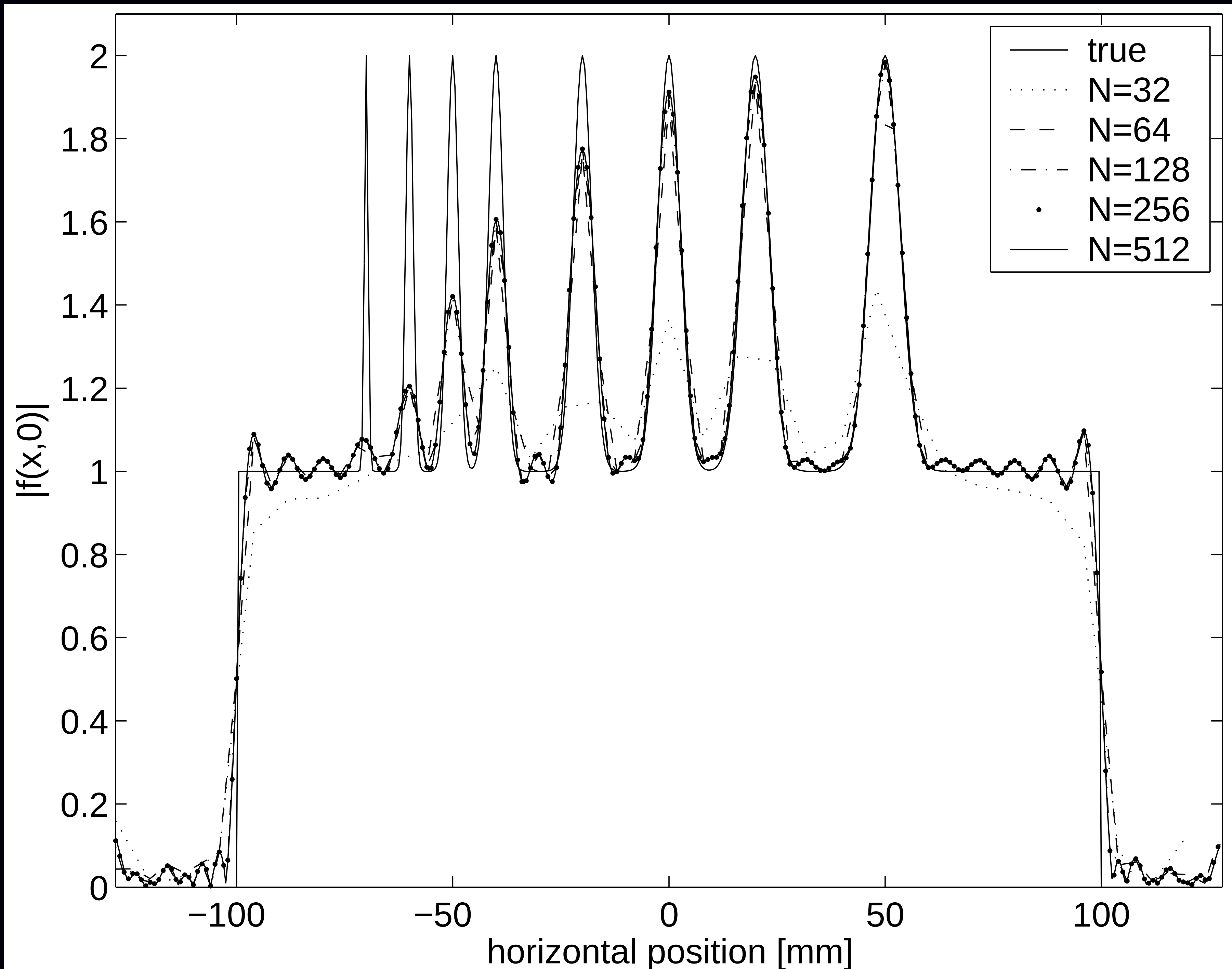
$$\boxed{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{f} + \boldsymbol{\varepsilon}.}$$

Goal: estimate coefficients (pixel values) $\boldsymbol{f} = (f_1, \ldots, f_N)$ from $\boldsymbol{y}$.

Small Pixel Size Need Not Matter



Profiles



true
N=32
N=64
N=128
N=256
N=512
|f(x,0)|
horizontal position [mm]
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−100
−50
0
50
100

Regularized Least-Squares Estimation

Estimate object by minimizing a cost function:

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f} \in \mathbb{C}^N} \Psi(\boldsymbol{f}), \qquad \Psi(\boldsymbol{f}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{f}\|^2 + \alpha \mathsf{R}(\boldsymbol{f})$$

- data fit term $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{f}\|^2$ corresponds to negative log-likelihood of Gaussian distribution
- regularizing term $\mathsf{R}(\boldsymbol{f})$ controls noise by penalizing roughness,

$$\textit{e.g.}: \quad \mathsf{R}(\boldsymbol{f}) \approx \int \|\nabla f\|^2 \, d\vec{r}$$

- regularization parameter $\alpha > 0$ controls tradeoff between spatial resolution and noise
- Equivalent to Bayesian MAP estimation with prior $\propto e^{-\alpha \mathsf{R}(\boldsymbol{f})}$

Issues:

- choosing $\mathsf{R}(f)$
- choosing α
- computing minimizer rapidly.

Quadratic regularization

1D example: squared differences between neighboring pixel values:

$$\mathsf{R}(f) = \sum_{j=2}^{N} \frac{1}{2} |f_j - f_{j-1}|^2.$$

In matrix-vector notation, $\mathsf{R}(\boldsymbol{f}) = \frac{1}{2} \|\boldsymbol{C} \boldsymbol{f}\|^2$ where

$$\boldsymbol{C} = \begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ & & & \ddots & \ddots & \\ 0 & \dots & 0 & 0 & -1 & 1 \end{bmatrix}, \text{ so } \boldsymbol{C} \boldsymbol{f} = \begin{bmatrix} f_2 - f_1 \\ \vdots \\ f_N - f_{N-1} \end{bmatrix}.$$

For 2D and higher-order differences, modify differencing matrix $\boldsymbol{C}$.

Leads to closed-form solution:

$$\begin{aligned} \hat{\boldsymbol{f}} &= \arg\min_{\boldsymbol{f}} \|\boldsymbol{y} - \boldsymbol{A} \boldsymbol{f}\|^2 + \alpha \|\boldsymbol{C} \boldsymbol{f}\|^2 \\ &= \left[\boldsymbol{A}' \boldsymbol{A} + \alpha \boldsymbol{C}' \boldsymbol{C}\right]^{-1} \boldsymbol{A}' \boldsymbol{y}. \end{aligned}$$

(a formula of limited practical use for computing $\hat{\boldsymbol{f}}$)

Choosing the Regularization Parameter

Spatial resolution analysis (Fessler & Rogers, IEEE T-IP, 1996):

$$\hat{\boldsymbol{f}} = \left[\boldsymbol{A}'\boldsymbol{A} + \alpha \boldsymbol{C}'\boldsymbol{C}\right]^{-1} \boldsymbol{A}'\boldsymbol{y}$$

$$\mathsf{E}\left[\hat{\boldsymbol{f}}\right] = \left[\boldsymbol{A}'\boldsymbol{A} + \alpha \boldsymbol{C}'\boldsymbol{C}\right]^{-1} \boldsymbol{A}' \,\mathsf{E}[\boldsymbol{y}]$$

$$\mathsf{E}\left[\hat{\boldsymbol{f}}\right] = \underbrace{\left[\boldsymbol{A}'\boldsymbol{A} + \alpha \boldsymbol{C}'\boldsymbol{C}\right]^{-1} \boldsymbol{A}'\boldsymbol{A}}_{\text{blur}} \, \boldsymbol{f}$$

$\boldsymbol{A}'\boldsymbol{A}$ and $\boldsymbol{C}'\boldsymbol{C}$ are Toeplitz $\Longrightarrow$ blur is approximately shift-invariant.

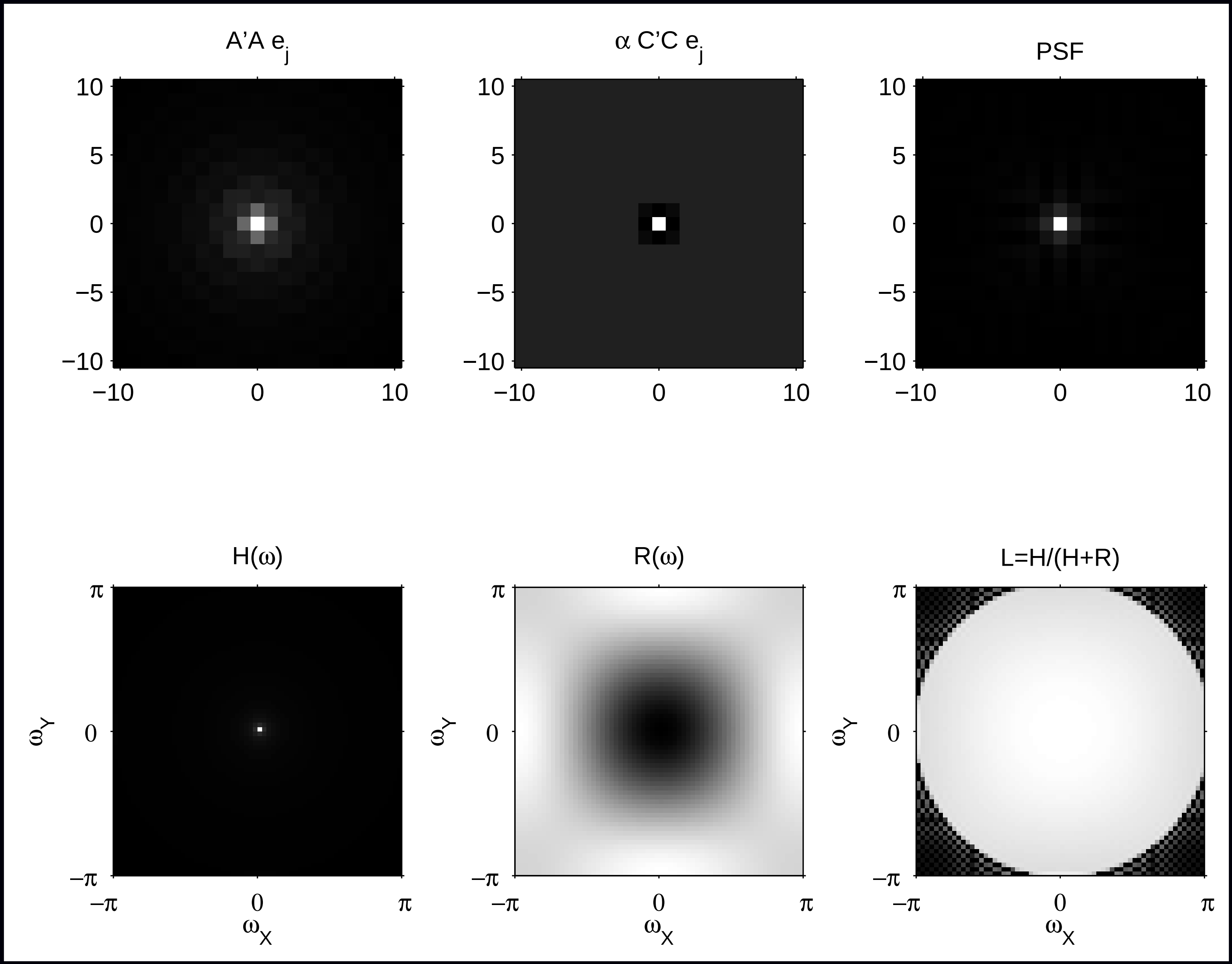
Frequency response of blur:

$$L(\omega) = \frac{H(\omega)}{H(\omega) + \alpha R(\omega)}$$

- $H(\omega_k) = \mathrm{FFT}(\boldsymbol{A}'\boldsymbol{A}\, e_j)$ (lowpass)
- $R(\omega_k) = \mathrm{FFT}(\boldsymbol{C}'\boldsymbol{C}\, e_j)$ (highpass)

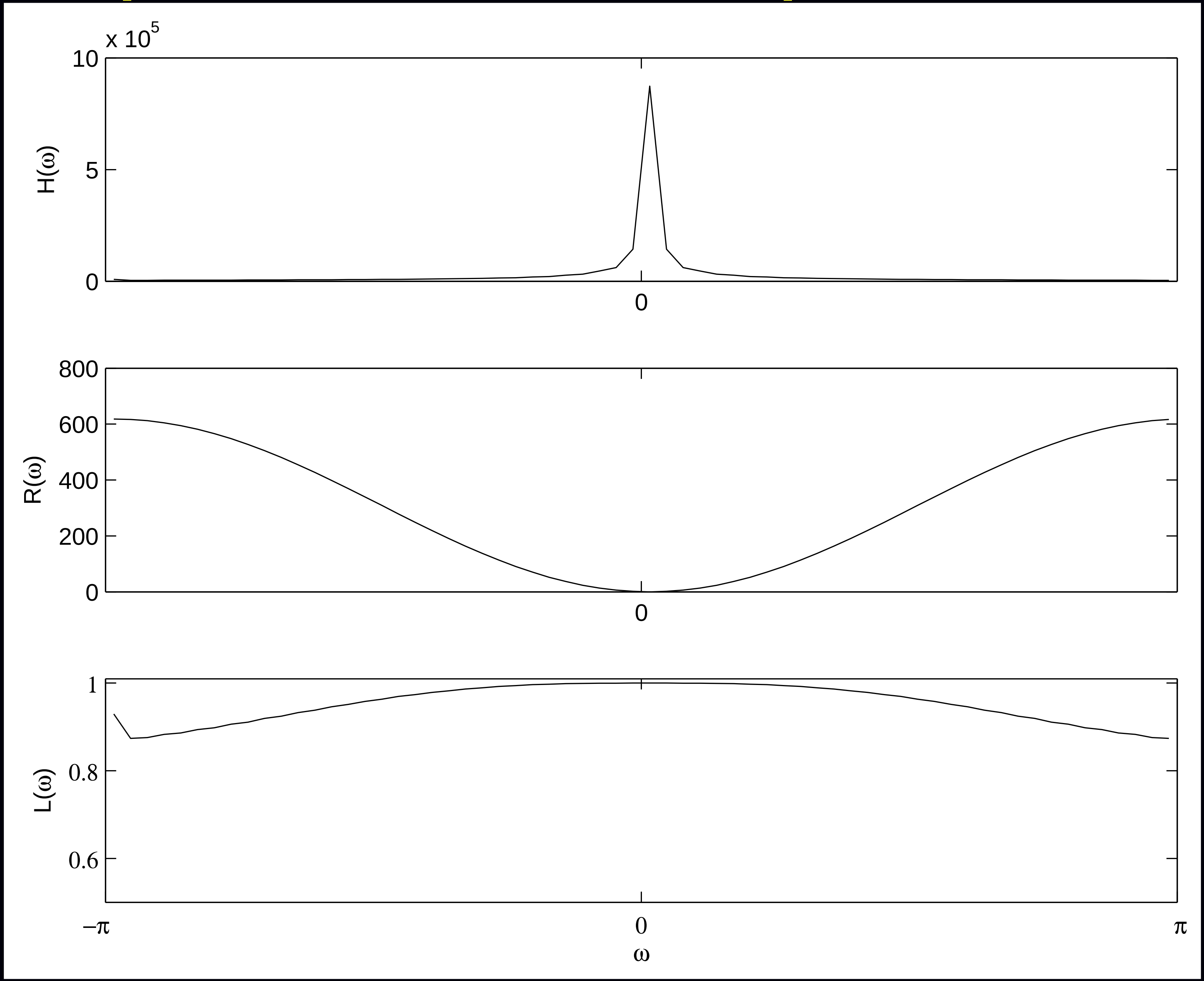
Adjust α to achieve desired spatial resolution.

Spatial Resolution Example



Radial k-space trajectory, FWHM of PSF is 1.2 pixels

Spatial Resolution Example: Profiles



Resolution/noise tradeoffs

Noise analysis:

$$\mathsf{Cov}\left\{\hat{\boldsymbol{f}}\right\} = \left[\boldsymbol{A}'\boldsymbol{A} + \alpha \boldsymbol{C}'\boldsymbol{C}\right]^{-1} \boldsymbol{A}' \,\mathsf{Cov}\{\boldsymbol{y}\}\, \boldsymbol{A} \left[\boldsymbol{A}'\boldsymbol{A} + \alpha \boldsymbol{C}'\boldsymbol{C}\right]^{-1}$$

Using circulant approximations to $\boldsymbol{A}'\boldsymbol{A}$ and $\boldsymbol{C}'\boldsymbol{C}$ yields:

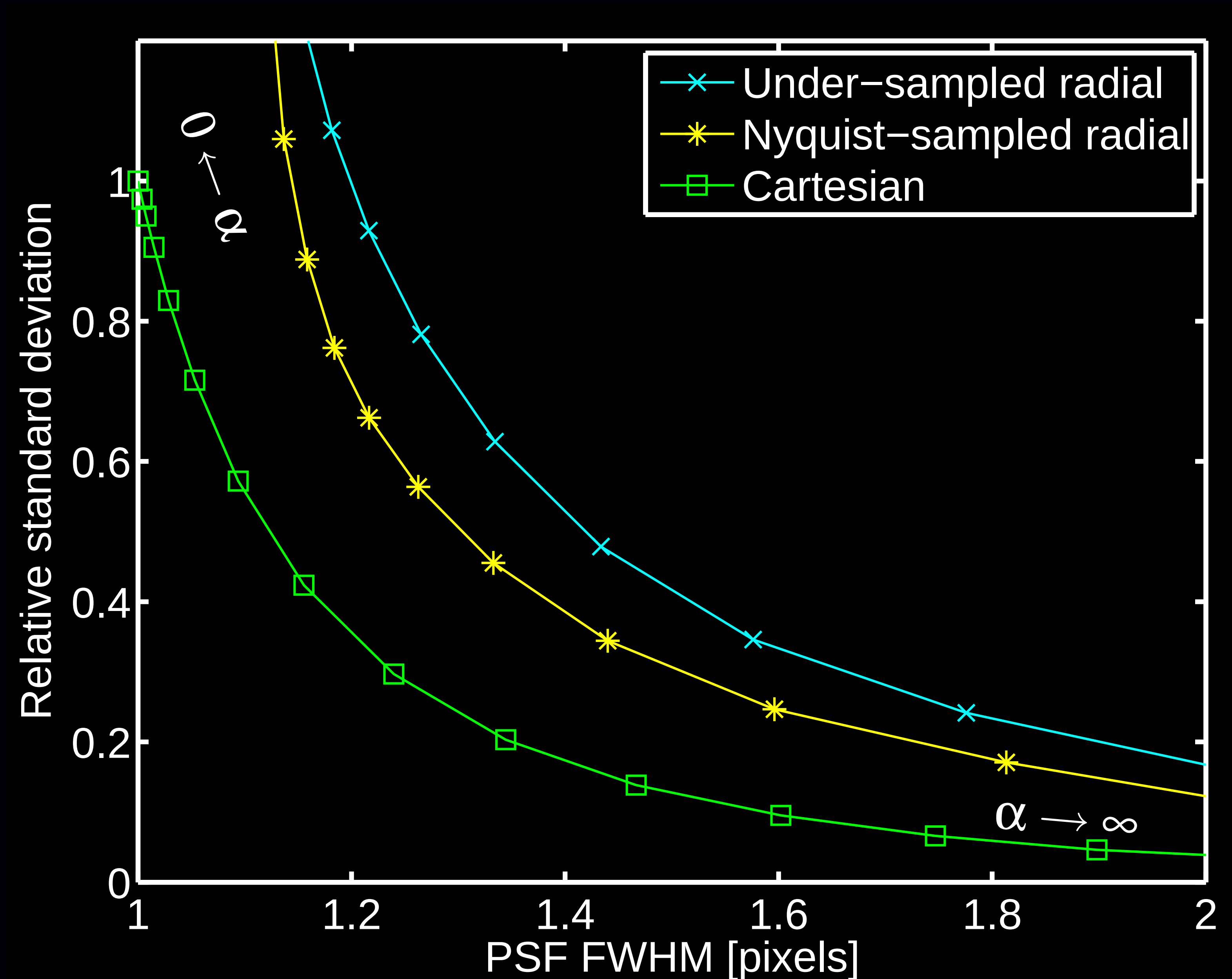
$$\mathsf{Var}\left\{\hat{f}_j\right\} \approx \sigma_\varepsilon^2 \sum_k \frac{H(\omega_k)}{(H(\omega_k) + \alpha R(\omega_k))^2}$$

- $H(\omega_k) = \mathrm{FFT}(\boldsymbol{A}'\boldsymbol{A}\, e_j)$ (lowpass)
- $R(\omega_k) = \mathrm{FFT}(\boldsymbol{C}'\boldsymbol{C}\, e_j)$ (highpass)

$\Longrightarrow$ Predicting reconstructed image noise requires just 2 FFTs.
(*cf.* gridding approach?)

Adjust α to achieve desired spatial resolution / noise tradeoff.

Resolution/Noise Tradeoff Example



In short: one can choose α rapidly and predictably for quadratic regularization.

Iterative Minimization by Conjugate Gradients

Choose initial guess $\boldsymbol{f}^{(0)}$ (*e.g.*, fast conjugate phase / gridding).
Iteration (unregularized):

$$
\begin{aligned}
\boldsymbol{g}^{(n)} &= \nabla\Psi\left(\boldsymbol{f}^{(n)}\right) = \boldsymbol{A}'(\boldsymbol{A}\boldsymbol{f}^{(n)} - \boldsymbol{y}) && \text{gradient} \\
\boldsymbol{p}^{(n)} &= \boldsymbol{P}\boldsymbol{g}^{(n)} && \text{precondition} \\
\gamma_n &= \begin{cases} 0, & n = 0 \\ \dfrac{\langle \boldsymbol{g}^{(n)}, \boldsymbol{p}^{(n)} \rangle}{\langle \boldsymbol{g}^{(n-1)}, \boldsymbol{p}^{(n-1)} \rangle}, & n > 0 \end{cases} \\
\boldsymbol{d}^{(n)} &= -\boldsymbol{p}^{(n)} + \gamma_n \boldsymbol{d}^{(n-1)} && \text{search direction} \\
\alpha_n &= \langle \boldsymbol{d}^{(n)}, -\boldsymbol{g}^{(n)} \rangle / \langle \boldsymbol{A}\boldsymbol{f}^{(n)}, \boldsymbol{A}\boldsymbol{f}^{(n)} \rangle && \text{step size} \\
\boldsymbol{f}^{(n+1)} &= \boldsymbol{f}^{(n)} + \alpha_n \boldsymbol{d}^{(n)} && \text{update}
\end{aligned}
$$

Bottlenecks: computing $\boldsymbol{A}\boldsymbol{f}^{(n)}$ and $\boldsymbol{A}'\boldsymbol{r}$.

- $\boldsymbol{A}$ is too large to store explicitly (not sparse)
- Even if $\boldsymbol{A}$ were stored, directly computing $\boldsymbol{A}\boldsymbol{f}$ is $O(MN)$ per iteration, whereas FFT is only $O(M \log M)$.

Computing Af Rapidly

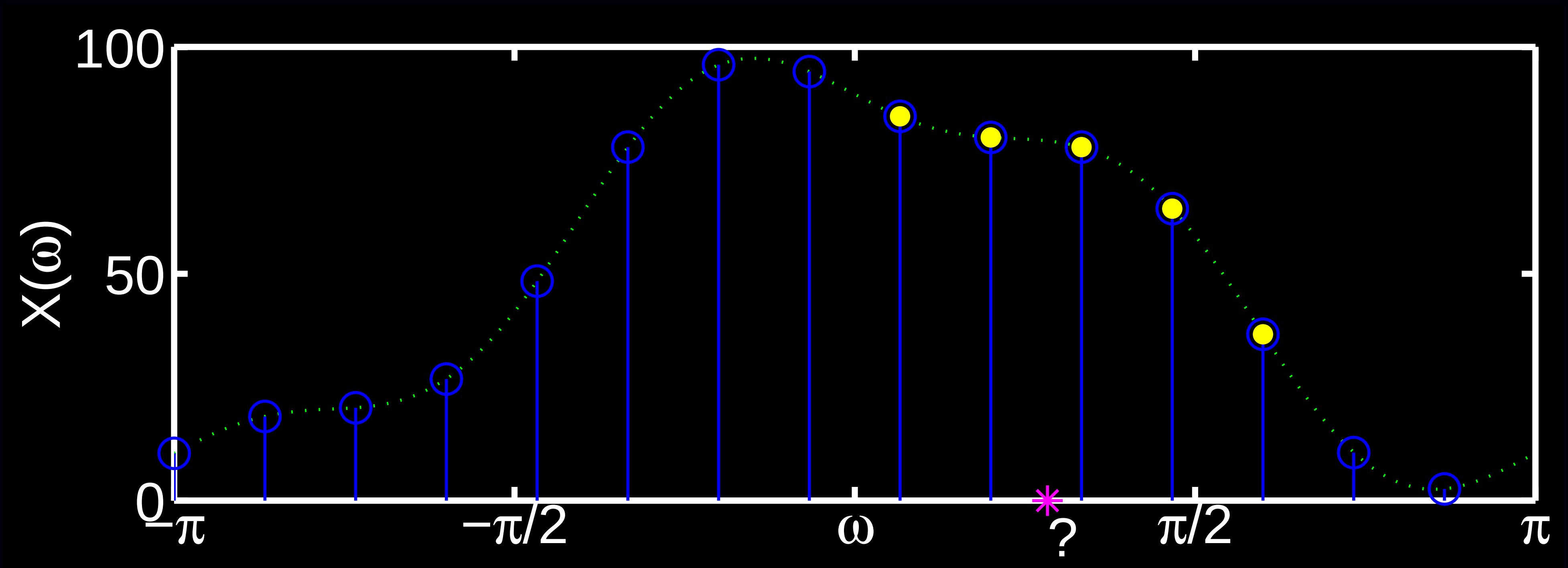
$$[\boldsymbol{A f}]_i = \sum_{j=1}^{N} a_{ij} f_j = P(\vec{\kappa}_i) \sum_{j=1}^{N} \mathrm{e}^{-\imath 2\pi \vec{\kappa}_i \cdot \vec{r}_j} f_j, \qquad i = 1, \ldots, M$$

- Pixel locations $\{\vec{r}_j\}$ are uniformly spaced
- k-space locations $\{\vec{\kappa}_i\}$ are unequally spaced

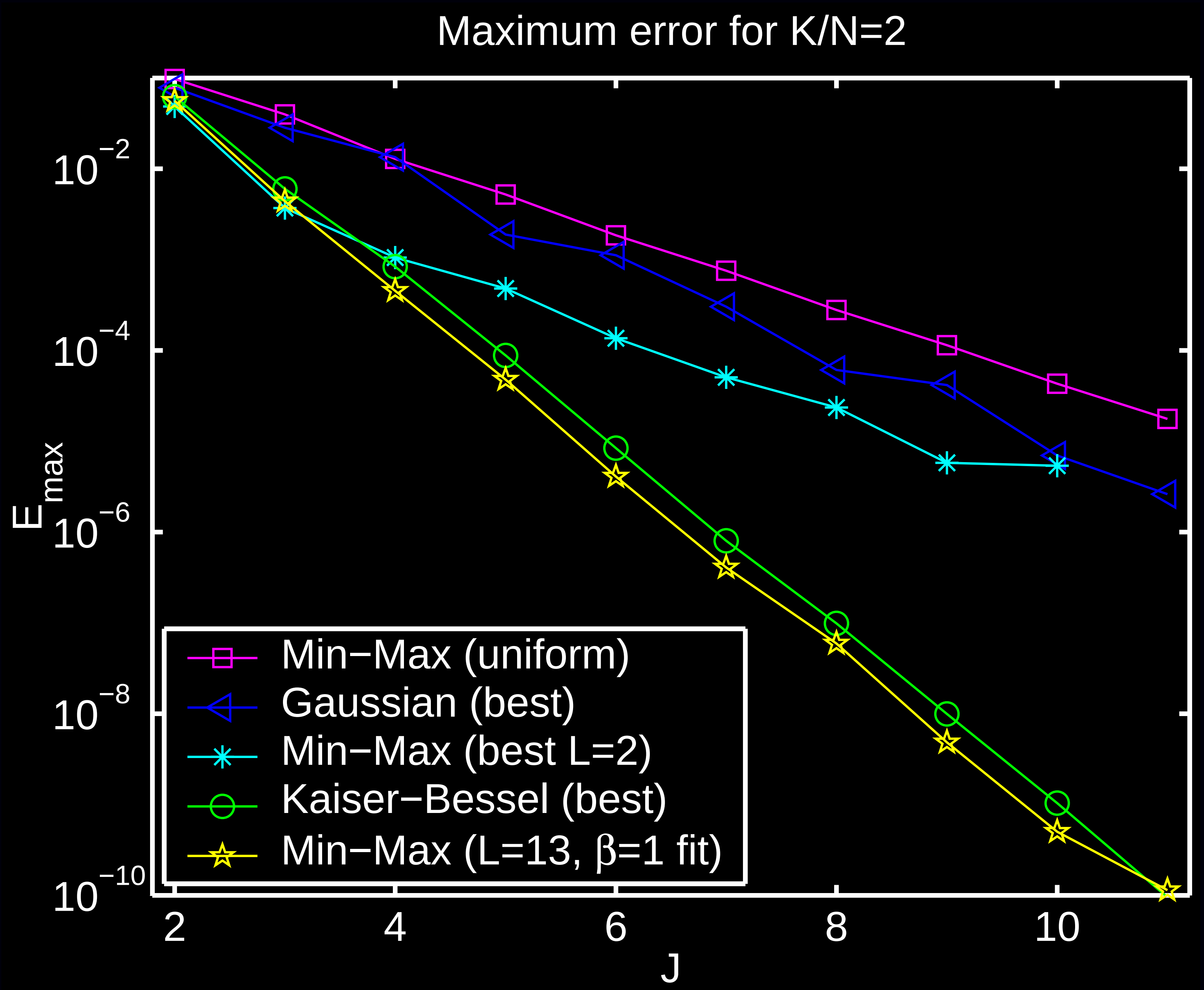
$\Longrightarrow$ needs nonuniform fast Fourier transform (NUFFT)

NUFFT (Type 2)

- Compute over-sampled FFT of equally-spaced signal samples
- Interpolate onto desired unequally-spaced frequency locations
- Dutt & Rokhlin, SIAM JSC, 1993, Gaussian bell interpolator
- Fessler & Sutton, IEEE T-SP, 2003, min-max interpolator
  and min-max optimized Kaiser-Bessel interpolator.
  NUFFT toolbox: `http://www.eecs.umich.edu/~fessler/code`



Worst-Case NUFFT Interpolation Error



Further Acceleration using Toeplitz Matrices

Cost-function gradient:

$$\begin{aligned} \boldsymbol{g}^{(n)} &= \boldsymbol{A}'(\boldsymbol{A}\boldsymbol{f}^{(n)} - \boldsymbol{y}) \\ &= \boldsymbol{T}\boldsymbol{f}^{(n)} - \boldsymbol{b}, \end{aligned}$$

where

$$\boldsymbol{T} \triangleq \boldsymbol{A}'\boldsymbol{A}, \qquad \boldsymbol{b} \triangleq \boldsymbol{A}'\boldsymbol{y}.$$

In the absence of field inhomogeneity, the Gram matrix $\boldsymbol{T}$ is Toeplitz:

$$[\boldsymbol{A}'\boldsymbol{A}]_{jk} = \sum_{i=1}^{M} |P(\vec{\kappa}_i)|^2 \, \mathrm{e}^{-\imath 2\pi \vec{\kappa}_i \cdot (\vec{r}_j - \vec{r}_k)}.$$

Computing $\boldsymbol{T}\boldsymbol{f}^{(n)}$ requires an ordinary ($2\times$ over-sampled) FFT.

(Chan & Ng, SIAM Review, 1996)

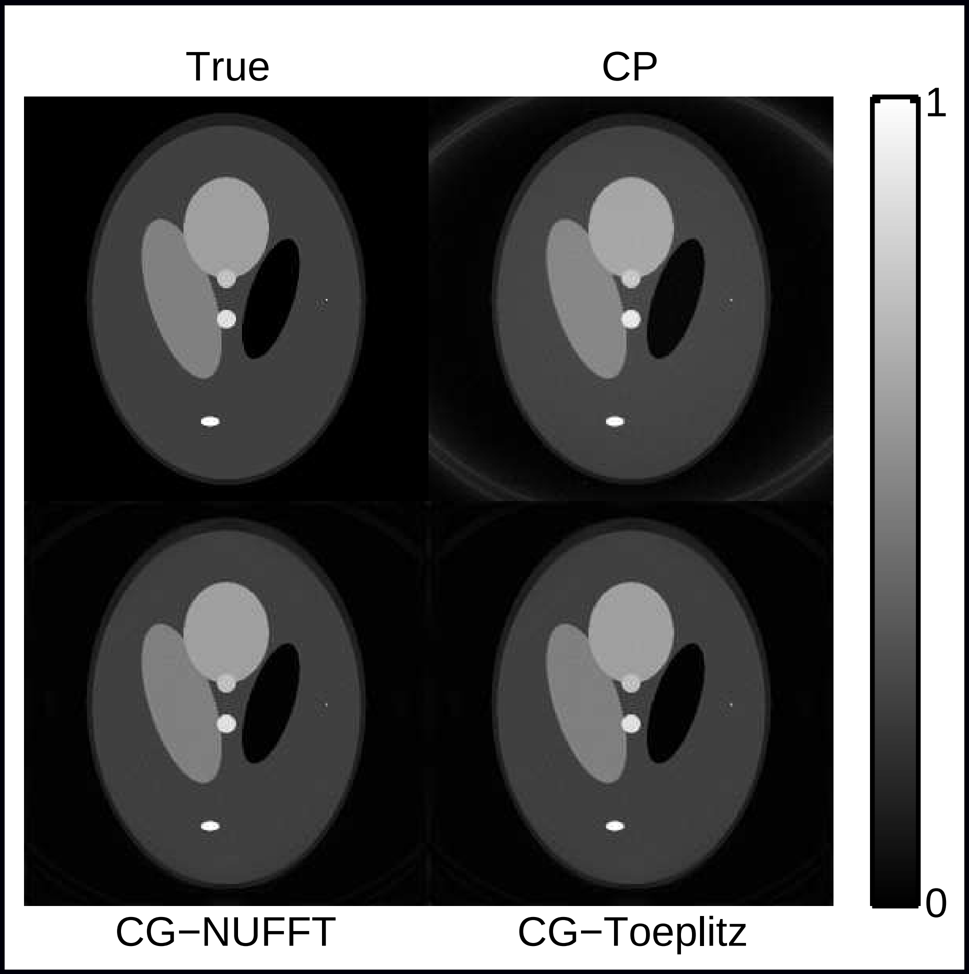
In 2D: block Toeplitz with Toeplitz blocks (BTTB).

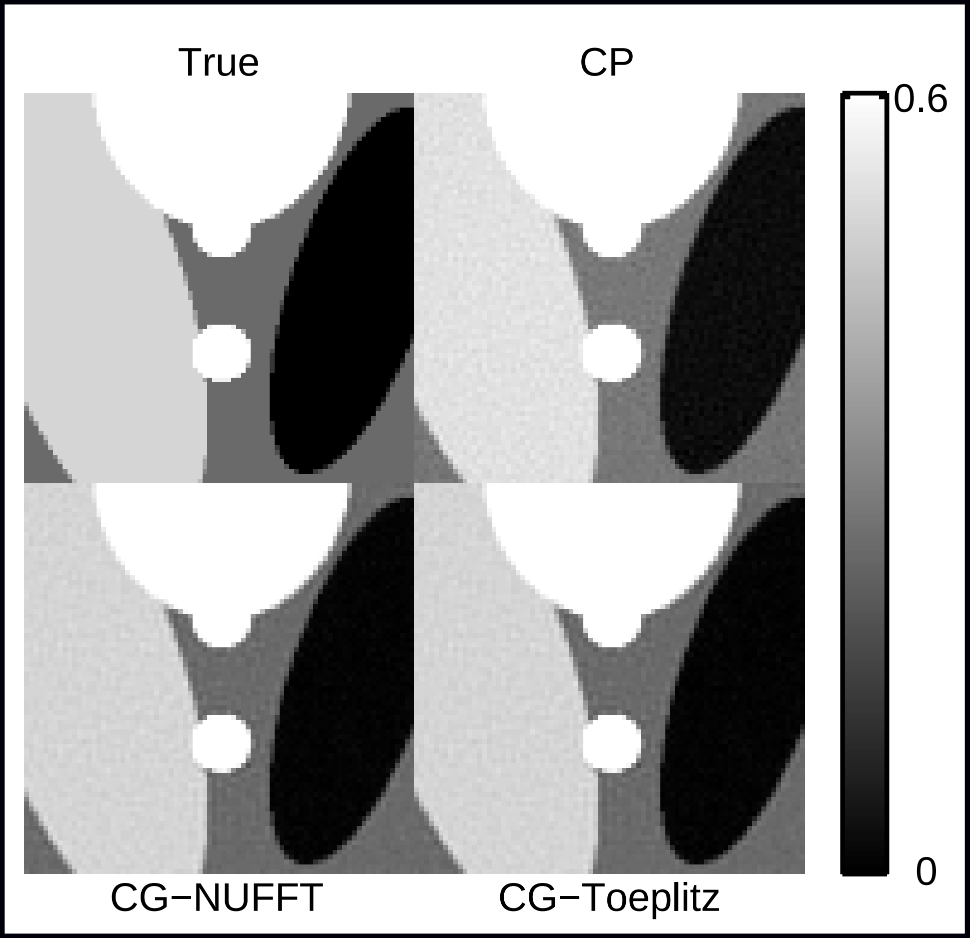
Precomputing the first column of $\boldsymbol{T}$ and $\boldsymbol{b}$ requires a couple NUFFTs.
(Wajer, ISMRM 2001, Eggers ISMRM 2002, Liu ISMRM 2005)

This formulation seems ideal for “hardware” FFT systems.

Toeplitz Acceleration

Example: 256^2 image. radial trajectory, $2\times$ angular under-sampling.





(Iterative provides reduced aliasing energy.)

Toeplitz Acceleration

Method	$A'Dy$	$b = A'y$	T	20 iter	Total Time	NRMS (50dB)
Conj. Phase	0.3				0.3	7.8%
CG-NUFFT				12.5	12.5	4.1%
CG-Toeplitz		0.3	0.8	3.5	4.6	4.1%

- Toeplitz aproach reduces CPU time by more than $2\times$ on conventional workstation (Xeon 3.4GHz)
- Eliminates k-space interpolations $\Longrightarrow$ ideal for FFT hardware
- No SNR compromise
- CG reduces NRMS error relative to CP, but $15\times$ slower... (More dramatic improvements seen in fMRI when correcting for field inhomogeneity.)

Myths

- Choosing α is difficult
- Sample density weighting is desirable

Sampling density weighted LS

Some researchers recommend using a weighted LS cost function:

$$\Psi(\boldsymbol{f}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{f}\|_{\boldsymbol{W}}^2$$

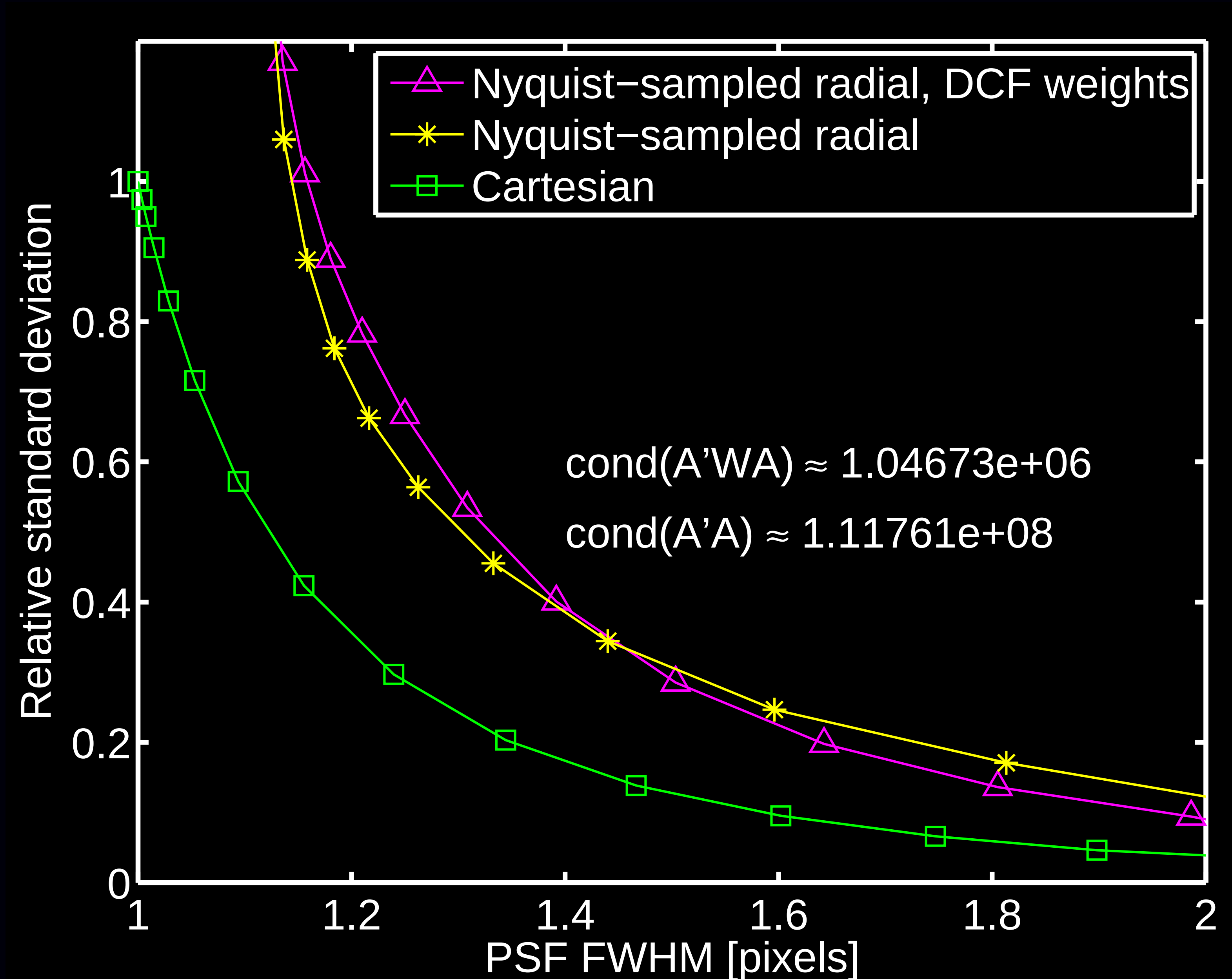
where the weighting matrix $\boldsymbol{W}$ is related to the k-space sample density compensation factors (DCF).

Purported benefits:

- Faster convergence
- Better conditioning

But, Gauss-Markov theorem from statistical estimation theory states that lowest estimator variance is realized by choosing $\boldsymbol{W} = \sigma_\varepsilon^{-2}\boldsymbol{I}$, the inverse of the data noise covariance.

Resolution/Noise Tradeoff: Example with Weighting



Don’t just take it from me...

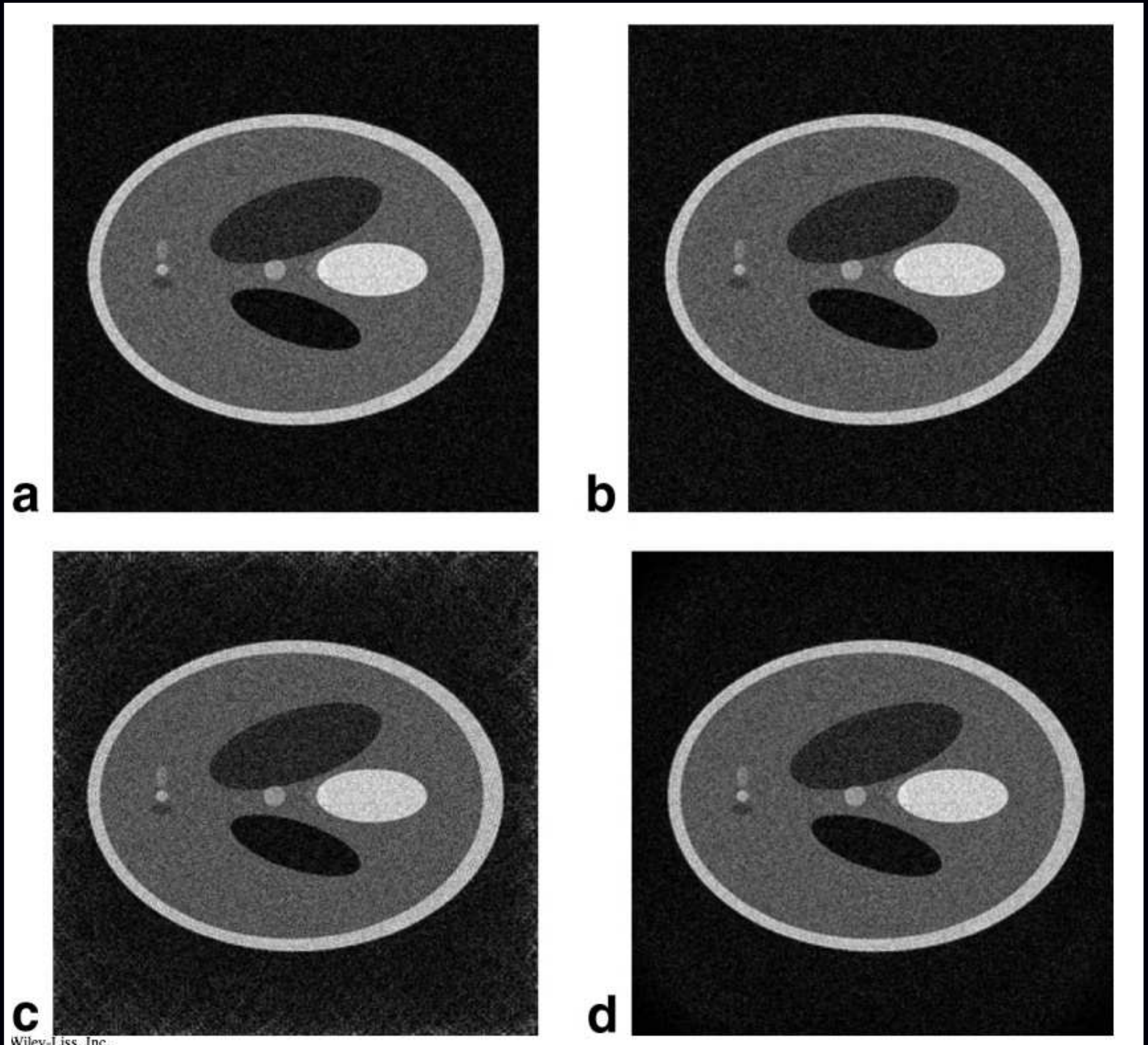


Figure 5. Reconstructed images for simulated spiral sampling of the noisy *k*-space of the SL phantom (60% noise) using conventional gridding (**a**), F-CG without (**b**) and with spiral DCF (**c**), and DING (**d**).

Fig. 5 of Gabr *et al.*, MRM, Dec. 2006

Acceleration via Weighting?

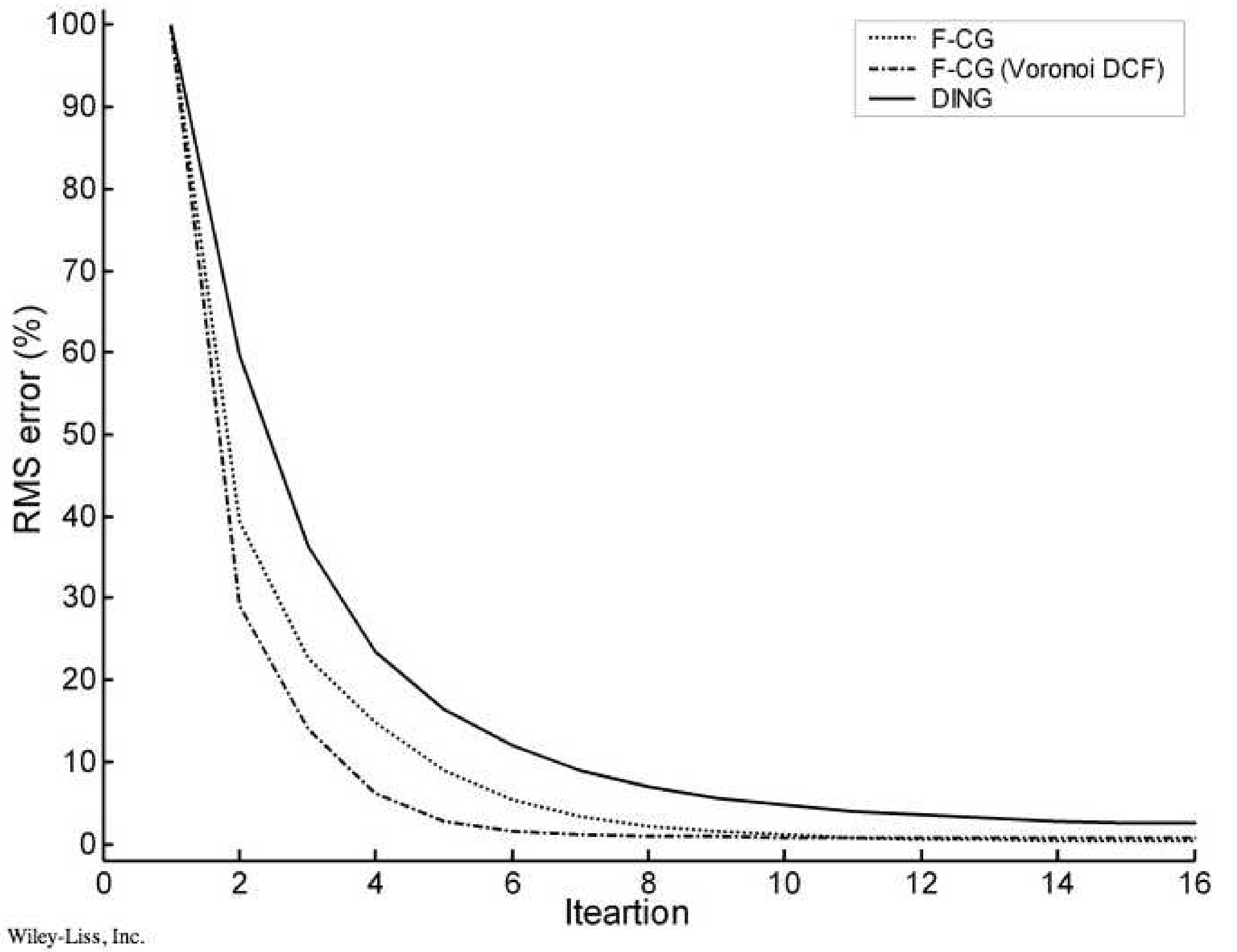
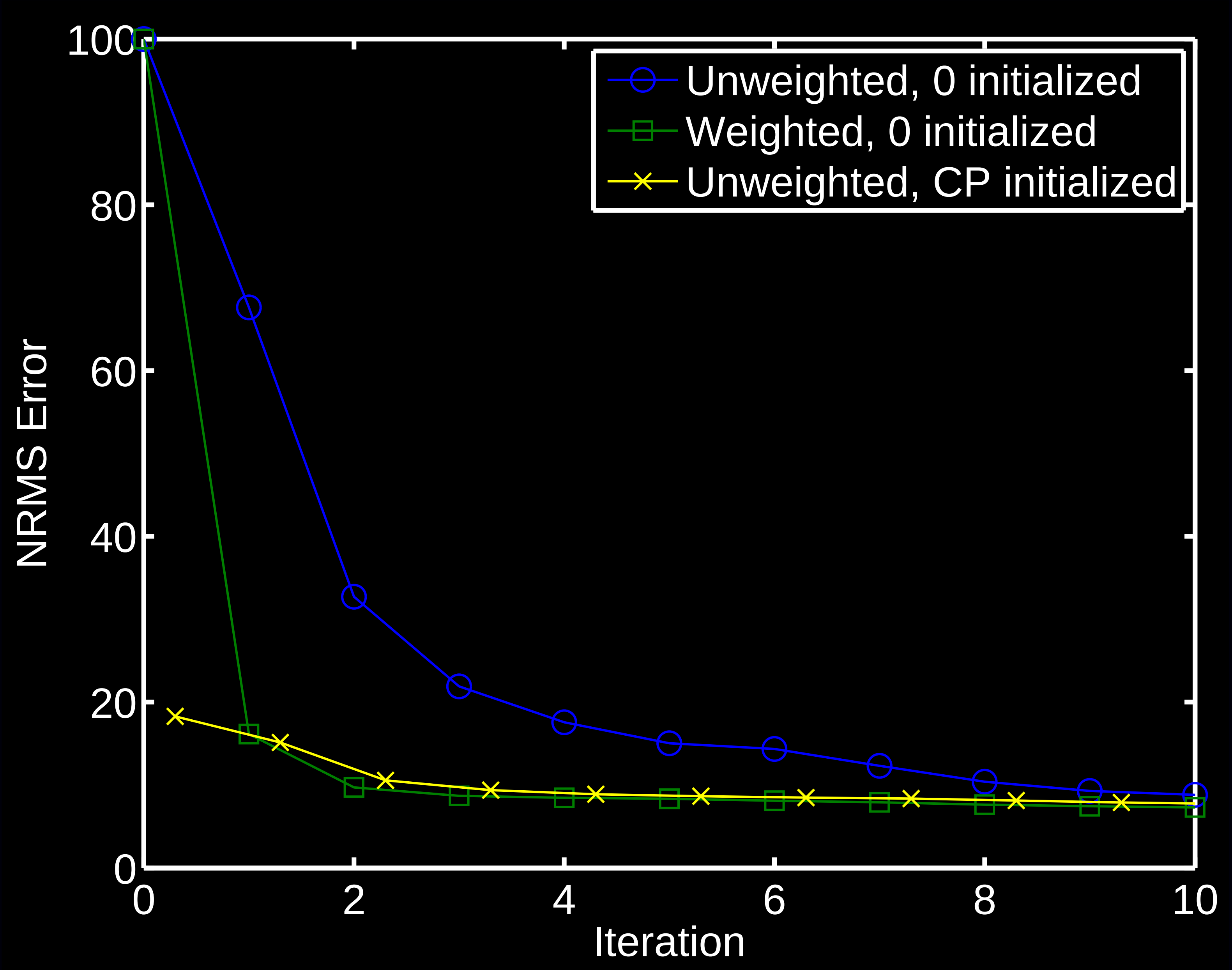


Figure 8. The RMS error of F-CG and DING vs. the number of CG iterations for the simulated noiseless SL the random trajectory. Errors stabilize after approximately 15 iterations.

Fig. 8 of Gabr *et al.*, MRM, Dec. 2006. Zero initialization!

Acceleration via Initialization



Summary

- Iterative reconstruction: much potential in MRI
- Quadratic regularization parameter selection is tractable
- Computation: reduced by tools like NUFFT / Toeplitz
- But optimization algorithm design remains important
  (*cf.* Shepp and Vardi, 1982, PET)

Some current challenges

- Sensitivity pattern mapping for SENSE
- Through-voxel field inhomogeneity gradients
- Motion / dynamics / partial k-space data
- Establishing diagnostic efficacy with clinical data...

Image reconstruction toolbox:
`http://www.eecs.umich.edu/~fessler`