Tradeoffs and complexities in new reconstruction methods

Jeffrey A. Fessler
EECS Department
The University of Michigan

ISMRM Imaging Strategies Course
May 4, 2008

Acknowledgements: Doug Noll, Brad Sutton, Valur Olafsson, Amanda Funai
Legalese

The presenter has no conflicts of interest with regards to this talk.
Outline

- MR image reconstruction problem description
- Overview of image reconstruction methods
- MR image reconstruction introduction
- Conventional reconstruction
- Model-based image reconstruction
- Iterations and computation (NUFFT etc.)
- Regularization
- Field inhomogeneity correction

Image reconstruction toolbox:
http://www.eecs.umich.edu/~fessler
Example: Iterative Reconstruction under $\Delta B_0$
Introduction to Reconstruction
Cartesian sampling in k-space. An inverse FFT. End of story.

Commercial MR system quotes 400 FFTs ($256^2$) per second.
Non-Cartesian MR Image Reconstruction

“k-space” data
\[ y = (y_1, \ldots, y_M) \]

image
\[ f(\vec{r}) \]

k-space trajectory:
\[ \vec{k}(t) = (k_x(t), k_y(t)) \]

spatial coordinates:
\[ \vec{r} \in \mathbb{R}^d \]
Textbook MRI Measurement Model

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

\[ y_i = s(t_i) + \text{noise}_i, \quad i = 1, \ldots, M \]

\[ s(t) = \int f(\vec{r}) e^{-t^2 \pi \vec{k}(t) \cdot \vec{r}} d\vec{r} = F(\vec{k}(t)). \]

\( \vec{r} \): spatial coordinates
\( \vec{k}(t) \): k-space trajectory of the MR pulse sequence
\( f(\vec{r}) \): object’s unknown transverse magnetization
\( F(\vec{k}) \): Fourier transform of \( f(\vec{r}) \). We get noisy samples of this!
\( e^{-t^2 \pi \vec{k}(t) \cdot \vec{r}} \) provides spatial information \( \implies \) Nobel Prize

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 \( y = (y_1, \ldots, y_M) \) are finite.

Under-determined (ill posed) problem \( \implies \) 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:
  
  \[ F(\vec{k}) = \int f(\vec{r}) e^{-i2\pi \vec{k} \cdot \vec{r}} d\vec{r} . \]
  
  The “solution” is an inverse Fourier transform:
  
  \[ f(\vec{r}) = \int F(\vec{k}) e^{i2\pi \vec{k} \cdot \vec{r}} d\vec{k} . \]
  
  Now discretize the integral solution:
  
  \[ \hat{f}(\vec{r}) = \sum_{i=1}^{M} F(\vec{k}_i) e^{i2\pi \vec{k}_i \cdot \vec{r}} w_i \approx \sum_{i=1}^{M} y_i w_i e^{i2\pi \vec{k}_i \cdot \vec{r}} , \]
  
  where \( w_i \) values are “sampling density compensation factors.” Numerous methods for choosing \( w_i \) values 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:
  
  \[
  \mathbf{y} = \mathcal{A} \mathbf{f} + \mathbf{\epsilon}, \quad \text{where} \quad \mathcal{A} : L_2(\mathbb{R}^d) \rightarrow \mathbb{C}^M.
  \]

  Minimum norm solution (*cf.* “natural pixels”):
  
  \[
  \begin{aligned}
  &\min_{\mathbf{\hat{f}}} \|\mathbf{\hat{f}}\|_2 \quad \text{subject to} \quad \mathbf{y} = \mathcal{A} \mathbf{\hat{f}} \\
  \mathbf{\hat{f}} = \mathcal{A}^* (\mathcal{A} \mathcal{A}^*)^{-1} \mathbf{y} = \sum_{i=1}^{M} c_i e^{-i2\pi \vec{\kappa}_i \cdot \vec{r}}, \quad \text{where} \quad \mathcal{A} \mathcal{A}^* \mathbf{c} = \mathbf{y}.
  \end{aligned}
  \]

  
  • **Discrete-discrete formulation**
  
  Assume parametric model for object:
  
  \[
  f(\vec{r}) = \sum_{j=1}^{N} f_j p_j(\vec{r}).
  \]
  
  Estimate parameter vector \( \mathbf{f} = (f_1, \ldots, f_N) \) from data vector \( \mathbf{y} \).
Model-Based Image Reconstruction: Overview
Model-Based Image Reconstruction

MR signal equation with more complete physics:

\[ s(t) = \int f(\vec{r}) s^{\text{coil}}(\vec{r}) e^{-i \omega(\vec{r}) t} e^{-R^*_{2}(\vec{r}) t} e^{-i 2\pi \vec{\kappa}(t) \cdot \vec{r}} d\vec{r} \]

\[ y_i = s(t_i) + \text{noise}_i, \quad i = 1, \ldots, M \]

- \( s^{\text{coil}}(\vec{r}) \): Receive-coil sensitivity pattern(s) (for SENSE)
- \( \omega(\vec{r}) \): Off-resonance frequency map (due to field inhomogeneity / magnetic susceptibility)
- \( R^*_{2}(\vec{r}) \): Relaxation map

Other physical factors (?)
- Eddy current effects; in \( \vec{\kappa}(t) \)
- Concomitant gradient terms
- Chemical shift
- Motion

Goal? (it depends)
Field Inhomogeneity-Corrected Reconstruction

\[ s(t) = \int f(\vec{r}) \, s^\text{coil}(\vec{r}) \, e^{-i\omega(\vec{r})t} \, e^{-R_2^*(\vec{r})t} \, e^{-i2\pi\vec{\kappa}(t) \cdot \vec{r}} \, d\vec{r} \]

Goal: reconstruct \( f(\vec{r}) \) given field map \( \omega(\vec{r}) \).
(Assume all other terms are known or unimportant.)

**Motivation**

Essential for functional MRI of brain regions near sinus cavities!

(Sutton et al., ISMRM 2001; T-MI 2003)
Sensitivity-Encoded (SENSE) Reconstruction

\[
s(t) = \int f(\vec{r}) s^{\text{coil}}(\vec{r}) e^{-i\omega(\vec{r}) t} e^{-R_z(\vec{r}) t} e^{-i2\pi\vec{K}(t) \cdot \vec{r}} d\vec{r}
\]

Goal: reconstruct \( f(\vec{r}) \) given sensitivity maps \( s^{\text{coil}}(\vec{r}) \).
(Assume all other terms are known or unimportant.)

Can combine SENSE with field inhomogeneity correction “easily.”

(Sutton et al., ISMRM 2001, Olafsson et al., ISBI 2006)
Joint Estimation of Image and Field-Map

\[ s(t) = \int f(\vec{r}) s^{\text{coil}}(\vec{r}) e^{-i\omega(\vec{r})t} e^{-R_t(\vec{r})t} e^{-i2\pi \vec{K}(t) \cdot \vec{r}} d\vec{r} \]

Goal: estimate both the image \( f(\vec{r}) \) and the field map \( \omega(\vec{r}) \) (Assume all other terms are known or unimportant.)

Analogy:
joint estimation of emission image and attenuation map in PET.

(Sutton et al., ISMRM Workshop, 2001; ISBI 2002; ISMRM 2002; ISMRM 2003; MRM 2004)
The Kitchen Sink

\[ s(t) = \int f(\vec{r}) s^\text{coil}(\vec{r}) e^{-i\omega(\vec{r})t} e^{-R_2^*(\vec{r})} e^{-i2\pi \vec{k}(t) \cdot \vec{r}} \, d\vec{r} \]

Goal: estimate image \( f(\vec{r}) \), field map \( \omega(\vec{r}) \), and relaxation map \( R_2^*(\vec{r}) \)

Requires “suitable” k-space trajectory.

(Sutton et al., ISMRM 2002; Twieg, MRM, 2003)
Estimation of Dynamic Rate Maps

\[ s(t) = \int f(\vec{r}) s^{\text{coil}}(\vec{r}) e^{-i\omega(\vec{r})t} e^{-\frac{R^*_2(\vec{r})}{2} t} e^{-i 2\pi \vec{k}(t) \cdot \vec{r}} d\vec{r} \]

Goal: estimate dynamic field map \( \omega(\vec{r}) \) and “BOLD effect” \( R^*_2(\vec{r}) \) given baseline image \( f(\vec{r}) \) in fMRI.

Motion...

(Olafsson et al., IEEE T-MI 2008)
Model-Based Image Reconstruction: Details
Basic Signal Model

\[ y_i = s(t_i) + \varepsilon_i, \quad \mathbb{E}[y_i] = s(t_i) = \int f(\vec{r}) e^{-i2\pi \vec{k}_i \cdot \vec{r}} \, d\vec{r} \]

Goal: reconstruct \( f(\vec{r}) \) from \( y = (y_1, \ldots, y_M) \).

Series expansion of unknown object:

\[ f(\vec{r}) \approx \sum_{j=1}^{N} f_j p(\vec{r} - \vec{r}_j) \quad \text{--- usually 2D rect functions.} \]

Substituting into signal model yields

\[
\mathbb{E}[y_i] = \int \left[ \sum_{j=1}^{N} f_j p(\vec{r} - \vec{r}_j) \right] e^{-i2\pi \vec{k}_i \cdot \vec{r}} \, d\vec{r} = \sum_{j=1}^{N} \left[ \int p(\vec{r} - \vec{r}_j) e^{-i2\pi \vec{k}_i \cdot \vec{r}} \, d\vec{r} \right] f_j
\]

\[ = \sum_{j=1}^{N} a_{ij} f_j, \quad a_{ij} = P(\vec{k}_i) e^{-i2\pi \vec{k}_i \cdot \vec{r}_j}, \quad p(\vec{r}) \iff P(\vec{k}). \]

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

\[ y = Af + \varepsilon. \]

Goal: estimate coefficients (pixel values) \( \hat{f} = (f_1, \ldots, f_N) \) from \( y \).
Small Pixel Size Need Not Matter

x true

N=32

N=64

N=128

N=256

N=512
Least-Squares Estimation

Estimate object by minimizing a simple cost function:

$$\hat{f} = \arg\min_{f \in \mathbb{C}^N} \Psi(f), \quad \Psi(f) = \|y - Af\|^2$$

- **Data fit term** $\|y - Af\|^2$
  - Corresponds to negative log-likelihood of Gaussian distribution
- Equivalent to maximum-likelihood (ML) estimation

Issues:
- Computing minimizer rapidly
- Stopping iteration (?)
- Image quality
Iterative Minimization by Conjugate Gradients

Choose initial guess \( f^{(0)} \) (e.g., fast conjugate phase / gridding).

Iteration (unregularized):

\[
\begin{align*}
\mathbf{g}^{(n)} &= \nabla \Psi(f^{(n)}) = \mathbf{A}'(\mathbf{Af}^{(n)} - \mathbf{y}) \quad \text{gradient} \\
\mathbf{p}^{(n)} &= \mathbf{P} \mathbf{g}^{(n)} \quad \text{precondition} \\
\gamma_n &= \begin{cases} 
0, & n = 0 \\
\frac{\langle \mathbf{g}^{(n)}, \mathbf{p}^{(n)} \rangle}{\langle \mathbf{g}^{(n-1)}, \mathbf{p}^{(n-1)} \rangle}, & n > 0
\end{cases} \\
\mathbf{d}^{(n)} &= -\mathbf{p}^{(n)} + \gamma_n \mathbf{d}^{(n-1)} \quad \text{search direction} \\
\mathbf{v}^{(n)} &= \mathbf{Ad}^{(n)} \\
\alpha_n &= \frac{\langle \mathbf{d}^{(n)}, -\mathbf{g}^{(n)} \rangle}{\| \mathbf{v}^{(n)} \|^2} \quad \text{step size} \\
f^{(n+1)} &= f^{(n)} + \alpha_n \mathbf{d}^{(n)} \quad \text{update}
\end{align*}
\]

Bottlenecks: computing \( \mathbf{Af}^{(n)} \) and \( \mathbf{A}' \mathbf{r} \).

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

$$[Af]_i = \sum_{j=1}^{N} a_{ij} f_j = P(\vec{\kappa}_i) \sum_{j=1}^{N} e^{-i2\pi \vec{\kappa}_i \cdot \vec{r}_j} f_j, \quad i = 1, \ldots, M$$

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

$\implies$ 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

**NUFFT toolbox:** [http://www.eecs.umich.edu/~fessler/code](http://www.eecs.umich.edu/~fessler/code)
Worst-Case NUFFT Interpolation Error

Maximum error for $K/N=2$

- Min–Max (uniform)
- Gaussian (best)
- Min–Max (best $L=2$)
- Kaiser–Bessel (best)
- Min–Max ($L=13$, $\beta=1$ fit)
Further Acceleration using Toeplitz Matrices

Cost-function gradient:

\[
g^{(n)} = A'(Af^{(n)} - y) = Tf^{(n)} - b,
\]

where

\[
T \triangleq A'A, \quad b \triangleq A'y.
\]

In the absence of field inhomogeneity, the Gram matrix \( T \) is Toeplitz:

\[
[A'A]_{jk} = \sum_{i=1}^{M} |P(\vec{k}_i)|^2 e^{-i2\pi \vec{k}_i \cdot (\vec{r}_j - \vec{r}_k)}.
\]

Computing \( Tf^{(n)} \) 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 \( T \) and \( b \) requires a couple NUFFTs.


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

<table>
<thead>
<tr>
<th>Method</th>
<th>$A'Dy$</th>
<th>$b = A'y$</th>
<th>$T$</th>
<th>20 iter</th>
<th>Total Time</th>
<th>NRMS (50dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conj. Phase</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
<td>7.8%</td>
</tr>
<tr>
<td>CG-NUFFT</td>
<td></td>
<td></td>
<td>12.5</td>
<td>12.5</td>
<td>4.1%</td>
<td></td>
</tr>
<tr>
<td>CG-Toeplitz</td>
<td>0.3</td>
<td>0.8</td>
<td>3.5</td>
<td>4.6</td>
<td>4.1%</td>
<td></td>
</tr>
</tbody>
</table>

- Toeplitz approach reduces CPU time by more than $2 \times$ on conventional workstation (Xeon 3.4GHz)
- Eliminates k-space interpolations $\implies$ 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.)
Unregularized Example: Simulated Data

4 × under-sampled radial k-space data
Analytical k-space data generation
Unregularized Example: Images

Iterations 1:4:60 of unregularized CG reconstruction
Unregularized Example: Movie
Unregularized Example: RMS Error

Unregularized CG

Zero image

Noisy LS image

"Best" image?

Iteration

NRMS Error (%)

Unregularized Eigenspectrum

Eigenvalues of $A'A$ for 4x under-sampled radial, 32x32
Regularized Example: Movie
Regularized Example: Image Comparison

True | Unregularized | Edge preserving regularization
Regularized Example: RMS Error

Graph showing the comparison of NRMS Error (%) between Unregularized and Regularized methods over CG Iteration.
Regularized Least-Squares Estimation

Estimate object by minimizing a regularized cost function:

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

- **data fit** term $\|y - Af\|^2$
  corresponds to negative log-likelihood of Gaussian distribution
- **regularizing** term $R(f)$ controls noise by penalizing roughness,
  e.g.:
  $$R(f) \approx \int \|\nabla f\|^2 \, d\vec{r}$$
- **regularization parameter** $\alpha > 0$
  controls tradeoff between spatial resolution and noise
- Equivalent to Bayesian MAP estimation with prior $\propto e^{-\alpha R(f)}$

Complexities:
- choosing $R(f)$
- choosing $\alpha$
- computing minimizer rapidly.
Quadratic regularization

1D example: squared differences between neighboring pixel values:
\[
R(f) = \sum_{j=2}^{N} \frac{1}{2} \left| f_j - f_{j-1} \right|^2.
\]

In matrix-vector notation, \( R(f) = \frac{1}{2} \| Cf \|_2^2 \) where
\[
C = \begin{bmatrix}
-1 & 1 & 0 & 0 & \ldots & 0 \\
0 & -1 & 1 & 0 & \ldots & 0 \\
& \ddots & \ddots & \ddots \\
0 & \ldots & 0 & 0 & -1 & 1
\end{bmatrix}, \text{ so } Cf = \begin{bmatrix}
f_2 - f_1 \\
\vdots \\
f_N - f_{N-1}
\end{bmatrix}.
\]

For 2D and higher-order differences, modify differencing matrix \( C \).

Leads to closed-form solution:
\[
\hat{f} = \arg \min_f \| y - Af \|^2 + \alpha \| Cf \|^2
\]
\[
= \left[ A' A + \alpha C'C \right]^{-1} A'y.
\]

(a formula of limited practical use for computing \( \hat{f} \))
Choosing the Regularization Parameter

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

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

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

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

\( A'A \) and \( C'C \) are Toeplitz \( \Rightarrow \) blur is approximately shift-invariant.

Frequency response of blur:

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

- \( H(\omega_k) = \text{FFT}(A'A e_j) \) (lowpass)
- \( R(\omega_k) = \text{FFT}(C'C e_j) \) (highpass)

Adjust \( \alpha \) to achieve desired spatial resolution.
Spatial Resolution Example

Radial k-space trajectory, FWHM of PSF is 1.2 pixels
Spatial Resolution Example: Profiles

H(ω)

R(ω)

L(ω)

ω

−π

0

π
Tabulating Spatial Resolution vs Regularization

Trajectory specific, but easily computed using a few FFTs Works only for quadratic regularization
Resolution/noise tradeoffs

Noise analysis:

\[
\text{Cov}\{\hat{f}\} = \left(A'A + \alpha C'C\right)^{-1} A' \text{Cov}\{y\} A \left[A'A + \alpha C'C\right]^{-1}
\]

Using circulant approximations to \(A'A\) and \(C'C\) yields:

\[
\text{Var}\{\hat{f}_j\} \approx \sigma^2 \varepsilon \sum_k \frac{H(\omega_k)}{(H(\omega_k) + \alpha R(\omega_k))^2}
\]

- \(H(\omega_k) = \text{FFT}(A'A e_j)\) (lowpass)
- \(R(\omega_k) = \text{FFT}(C'C e_j)\) (highpass)

⇒ Predicting reconstructed image noise requires just 2 FFTs. (cf. gridding approach?)

Adjust \(\alpha\) to achieve desired spatial resolution / noise tradeoff.
In short: one can choose $\alpha$ rapidly and predictably for quadratic regularization.
NUFFT with Field Inhomogeneity?


Recall signal model including field inhomogeneity:

\[ s(t) = \int f(\vec{r}) e^{-i\omega(\vec{r}) t} e^{-i2\pi \vec{\kappa}(t) \cdot \vec{r}} d\vec{r}. \]

Temporal interpolation approximation (aka “time segmentation”):

\[ e^{-i\omega(\vec{r}) t} \approx \sum_{l=1}^{L} a_l(t) e^{-i\omega(\vec{r}) \tau_l} \]

for min-max optimized temporal interpolation functions \( \{a_l(\cdot)\}_{l=1}^{L} \).

\[ s(t) \approx \sum_{l=1}^{L} a_l(t) \int f(\vec{r}) e^{-i\omega(\vec{r}) \tau_l} e^{-i2\pi \vec{\kappa}(t) \cdot \vec{r}} d\vec{r} \]

Linear combination of \( L \) NUFFT calls.
Field Corrected Reconstruction Example

Simulation using known field map $\omega(\vec{r})$. 
Simulation Quantitative Comparison

- Computation time?
- NRMSE between $\hat{f}$ and $f^{\text{true}}$?

<table>
<thead>
<tr>
<th>Reconstruction Method</th>
<th>Time (s)</th>
<th>NRMSE complex</th>
<th>NRMSE magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Correction</td>
<td>0.06</td>
<td>1.35</td>
<td>0.22</td>
</tr>
<tr>
<td>Full Conjugate Phase</td>
<td>4.07</td>
<td>0.31</td>
<td>0.19</td>
</tr>
<tr>
<td>Fast Conjugate Phase</td>
<td>0.33</td>
<td>0.32</td>
<td>0.19</td>
</tr>
<tr>
<td>Fast Iterative (10 iters)</td>
<td>2.20</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>Exact Iterative (10 iters)</td>
<td>128.16</td>
<td>0.04</td>
<td>0.04</td>
</tr>
</tbody>
</table>
Human Data: Field Correction

Uncorrected  Conjugate Phase  Fast Iterative  Field Map (Hz)

Uncorrected  Conjugate Phase  Fast Iterative  Field Map (Hz)
Acceleration using Toeplitz Approximations

In the presence of field inhomogeneity, the system matrix is:

\[ a_{ij} = P(\vec{\kappa}_i) e^{-i\omega(\vec{r}_j)t_i} e^{-i2\pi \vec{\kappa}_i \cdot \vec{r}_j} \]

The Gram matrix \( T = A'A \) is not Toeplitz:

\[ [A'A]_{jk} = \sum_{i=1}^{M} |P(\vec{\kappa}_i)|^2 e^{-i2\pi \vec{\kappa}_i \cdot (\vec{r}_j - \vec{r}_k)} e^{-i(\omega(\vec{r}_j) - \omega(\vec{r}_k))t_i}. \]

Approximation ("time segmentation"):

\[ e^{-i(\omega(\vec{r}_j) - \omega(\vec{r}_k))t_i} \approx \sum_{l=1}^{L} b_{il} e^{-i(\omega(\vec{r}_j) - \omega(\vec{r}_k))\tau_l} \]

\[ T = A'A \approx \sum_{l=1}^{L} D_l' T_l D_l, \quad D_l \triangleq \text{diag}\{e^{-i\omega(\vec{r}_j)\tau_l}\} \]

\[ [T_l]_{jk} \triangleq \sum_{i=1}^{M} |P(\vec{\kappa}_i)|^2 b_{il} e^{-i2\pi \vec{\kappa}_i \cdot (\vec{r}_j - \vec{r}_k)} . \]

Each \( T_l \) is Toeplitz \( \implies T f \) using \( L \) pairs of FFTs.

(Fessler et al., IEEE T-SP, Sep. 2005, brain imaging special issue)
Toeplitz Results

Uncorrected Conj. Phase, L=6

CG–NUFFT L=6

CG–Toeplitz L=8
## Toeplitz Acceleration

<table>
<thead>
<tr>
<th>Method</th>
<th>$L$</th>
<th>$B,C$</th>
<th>$A'Dy$</th>
<th>$b = A'y$</th>
<th>$T_l$</th>
<th>15 iter</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conj. Phase</td>
<td>6</td>
<td>0.4</td>
<td>0.2</td>
<td></td>
<td></td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>CG-NUFFT</td>
<td>6</td>
<td>0.4</td>
<td></td>
<td>5.0</td>
<td>5.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CG-Toeplitz</td>
<td>8</td>
<td>0.4</td>
<td>0.2</td>
<td>0.6</td>
<td>1.3</td>
<td>2.5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NRMS % vs SNR</th>
<th>$\infty$</th>
<th>50 dB</th>
<th>40 dB</th>
<th>30 dB</th>
<th>20 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conj. Phase</td>
<td>30.7</td>
<td>37.3</td>
<td>46.5</td>
<td>65.3</td>
<td>99.9</td>
</tr>
<tr>
<td>CG-NUFFT</td>
<td>5.6</td>
<td>16.7</td>
<td>26.5</td>
<td>43.0</td>
<td>70.4</td>
</tr>
<tr>
<td>CG-Toeplitz</td>
<td>5.5</td>
<td>16.7</td>
<td>26.4</td>
<td>42.9</td>
<td>70.4</td>
</tr>
</tbody>
</table>

- Reduces CPU time by $2 \times$ on conventional workstation (Mac G5)
- No SNR compromise
- Eliminates k-space interpolations $\implies$ ideal for FFT hardware
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

- Nonquadratic regularization: analysis / design
- 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