FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Practical Reconstruction Implementation

Zhengguo Tan
Department Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander-Universität Erlangen-Nürnberg
May 08, 2022

Declaration of Financial Interests or Relationships

Speaker Name: Zhengguo Tan

I have no financial interests or relationships to disclose with regard to the subject matter of this presentation.

Outline

Introduction to Open Source Software: SigPy
- Actual Practice: Locally Low Rank (LLR)
- Actual Practice: LLR regularized Linear Subspace Reconstruction

Examples
- Echo Planar Time Resolve Imaging (EPTI)
- Radial Echo Planar Imaging (REPI)

Summary

1 Introduction to Open Source Software: SigPy

Why is linear operators abstraction convenient?

Iterative image reconstruction (e.g. SENSE [1]) solves:

$$\arg\min_x \|y - \mathcal{F}_u S x\|_2^2 + \lambda R(x) \tag{1}$$

1. The unknown x can go beyond 2D, and the forward operator can be extended.
2. S is the multiplication with coil sensitivity maps, and $\mathcal{F}_u$ is the masked FFT or NUFFT.
3. Its minimization often requires following operations:
   3.1. Gradient of the data consistency term: $S^* \mathcal{F}_u^{-1}(y - \mathcal{F}_u S x)$
   3.2. Adjoint of the regularization term: $R^H R$

[1] Pruessmann KP, et al. SENSE: Sensitivity encoding for fast MRI. *Magn Reson Med* (1999).
[2] Ong F. `https://github.com/mikgroup/sigpy`.

Why is linear operators abstraction convenient?

Iterative image reconstruction (e.g. SENSE [1]) solves:

$$\arg\min_x \|y - \mathcal{F}_u S x\|_2^2 + \lambda R(x) \quad (1)$$

1. The unknown x can go beyond 2D, and the forward operator can be extended.
2. S is the multiplication with coil sensitivity maps, and $\mathcal{F}_u$ is the masked FFT or NUFFT.
3. Its minimization often requires following operations:
   3.1. Gradient of the data consistency term: $S^* \mathcal{F}_u^{-1}(y - \mathcal{F}_u S x)$
   3.2. Adjoint of the regularization term: $R^H R$

SigPy [2] provides linear operators abstraction (e.g. total variation)

```
>>> G = sp.linop.FiniteDifference([256, 256], axes=(-2, -1))
>>> y = G.H * G * x
```

[1] Pruessmann KP, et al. SENSE: Sensitivity encoding for fast MRI. *Magn Reson Med* (1999).
[2] Ong F. https://github.com/mikgroup/sigpy.

Linear Operator Implementation

Linop is the parent class, and all basic child linear operators require:

```
def __init__(self, oshape, ishape): # initialization
def _apply(self, input): # apply the linop, e.g. G(x) or G * x
def _adjoint_linop(self): # G.H
```

e.g. Multiply, which is used for the coil sensitivity operator S

- Its _apply function uses the multiply function in Python,
  e.g. I of shape $[256, 256]$ * C of shape $[4, 256, 256]$ outputs R of C's shape
- Its _adjoint_linop is then
  $\sum_{j=1}^{N_c} C_j^* \cdot I$,
  and is implemented as a **chain** of operators Reshape * Sum * Multiply

Example: Total Variation (TV)

Given matrix A: $\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$

down roll by 1: $\begin{bmatrix} 6 & 7 & 8 \\ 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}$

```
>>> np.roll(A, 1, axis=0)
```

right roll by 1: $\begin{bmatrix} 2 & 0 & 1 \\ 5 & 3 & 4 \\ 8 & 6 & 7 \end{bmatrix}$

```
>>> np.roll(A, 1, axis=1)
```

[3] Rudin LI, et al. Nonlinear total variation based noise removal algorithms. *Physica D* (1992).

Example: Total Variation (TV)

Given matrix A: $\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$

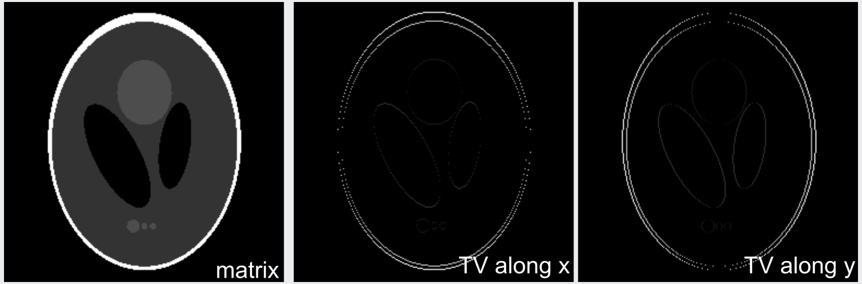
down roll by 1: $\begin{bmatrix} 6 & 7 & 8 \\ 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}$

```
>>> np.roll(A, 1, axis=0)
```

right roll by 1: $\begin{bmatrix} 2 & 0 & 1 \\ 5 & 3 & 4 \\ 8 & 6 & 7 \end{bmatrix}$

```
>>> np.roll(A, 1, axis=1)
```

Total variation [3] is the subtraction between the rolled and the input matrix.




```
>>> I = shepp_logan((256, 256))
>>> G = linop.FiniteDifference(
            I.shape,
            axes=[-2, -1])
>>> y = G * x
```

[3] Rudin LI, et al. Nonlinear total variation based noise removal algorithms. *Physica D* (1992).

How to assert if the linop operator is correct?

Linear operator properties

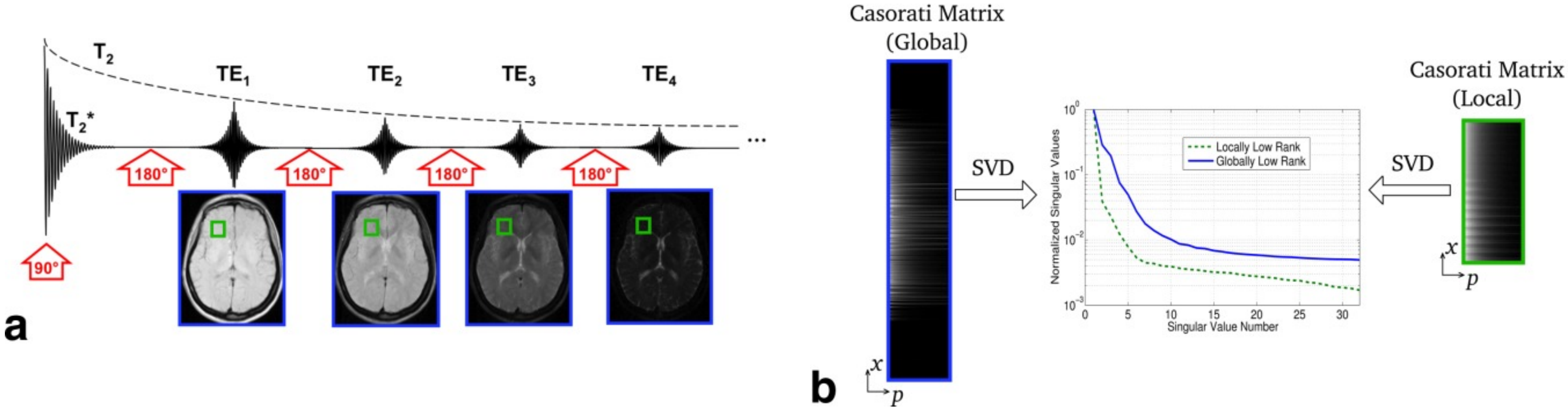
- Unitary: $A^H * A * x = x$ and $< A * x, y >=< x, A^H * y >$
- Linearity: $A(a * x + y) = a * A(x) + A(y)$

Routine Linop Test Functions

```
shape = [256, 256]
A = linop.FiniteDifference(shape)
self.check_linop_adjoint(A)
self.check_linop_normal(A)
self.check_linop_linear(A)
```

Outline

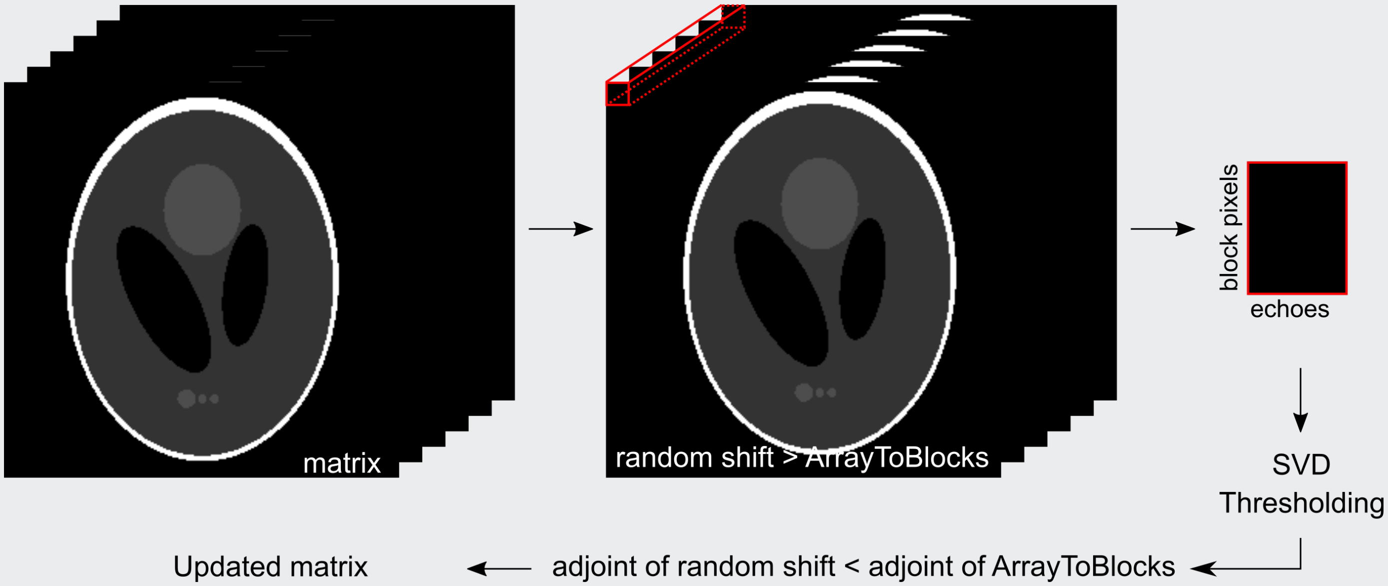
Exploring Sparsity in Multi-Contrast Images [4]



[4]Zhang T, et al. Accelerating parameter mapping with a locally low rank constraint . *Magn Reson Med* (2014).

Locally Low Rank (LLR) Soft Thresholding Process

LLR soft thresholding can be implemented as a proximal operator [5].



[5] Beck A. First-order methods in optimization. *SIAM* (2017).

Locally Low Rank (LLR) Soft Thresholding Implementation

Define all linops

```
def _linop_randshift(self):
    axes = [-2, -1]
    shift = [random.randint(0, self.blk_shape[s]) for s in axes]
    return linop.Circshift(self.shape, shift, axes)

RandShift = self._linop_randshift()
ATB = linop.ArrayToBlocks(shape, blk_shape, blk_strides)

def _linop_reshape(self):
    ...
    return R

Reshape = self._linop_reshape()
```

Locally Low Rank (LLR) Soft Thresholding Implementation

prox.LLRL1Reg

```
# forward
y1 = RandShift(input)
y2 = ATB(y1)
y3 = Reshape(y2)

# SVD soft thresholding
u, s, vh = np.linalg.svd(y3, full_matrices=False)
s_thresh = thresh.soft_thresh(self.lamda, s)
y4 = (u * s_thresh[..., None, :]) @ vh

# adjoint
y5 = Reshape.H(y4)
y6 = ATB.H(y5)
output = RandShift.H(y6)
```

Outline

Introduction to Open Source Software: SigPy

- Actual Practice: Locally Low Rank (LLR)
- Actual Practice: LLR regularized Linear Subspace Reconstruction

Examples

- Echo Planar Time Resolve Imaging (EPTI)
- Radial Echo Planar Imaging (REPI)

Summary

Linear Subspace Modeling Reduces the Number of Unknowns

$$\arg\min_{x} \|y - \mathcal{F}_u S x\|_2^2 + \lambda \|\mathsf{LLR}(x)\|_1 \tag{2}$$

x is multi-contrast images. However, the more contrast in the unknown requires longer reconstruction time.

[6] Huang C, et al. T2 mapping from highly undersampled data by reconstruction of principal component coefficient maps using compressed sensing. *Magn Reson Med* (2012).
[7] Tamir JI, et al. T2 shuffling: Sharp, multicontrast, volumetric fast spin-echo imaging. *Magn Reson Med* (2017).

Linear Subspace Modeling Reduces the Number of Unknowns

$$\arg\min_{x} \|y - \mathcal{F}_u S x\|_2^2 + \lambda \|\text{LLR}(x)\|_1 \tag{2}$$

x is multi-contrast images. However, the more contrast in the unknown requires longer reconstruction time.

Linear subspace modeling [6, 7]

$$\arg\min_{\alpha} \left\|y - \mathcal{F}_u S \hat{U} \alpha\right\|_2^2 + \lambda \|\text{LLR}(\alpha)\|_1 \tag{3}$$

- $\hat{U}$ is the truncated SVD of the simulated dictionary corresponding to the sequence protocol.
- α is the linear subspace coefficient maps

[6] Huang C, et al. T2 mapping from highly undersampled data by reconstruction of principal component coefficient maps using compressed sensing. *Magn Reson Med* (2012).
[7] Tamir JI, et al. T2 shuffling: Sharp, multicontrast, volumetric fast spin-echo imaging. *Magn Reson Med* (2017).

Solving LLR Regularized Linear Subspace Reconstruction

Alternating Direction Method of Multipliers (ADMM) [8]

$$\begin{aligned} &\text{minimize } l(x) + g(z) \\ &\text{subject to } x - z = 0 \end{aligned} \tag{4}$$

$l(x)$ is the data consistency term, and $g(z)$ is the regularization.

$$\begin{aligned} x^{k+1} &:= \arg\min_x l(x) + (\rho/2) \left\| x - x^k + u^k \right\|_2^2 \\ z^{k+1} &:= S_{\lambda/\rho}(x^{k+1} + u^k) \\ u^{k+1} &:= u^k + x^{k+1} - z^{k+1} \end{aligned} \tag{5}$$

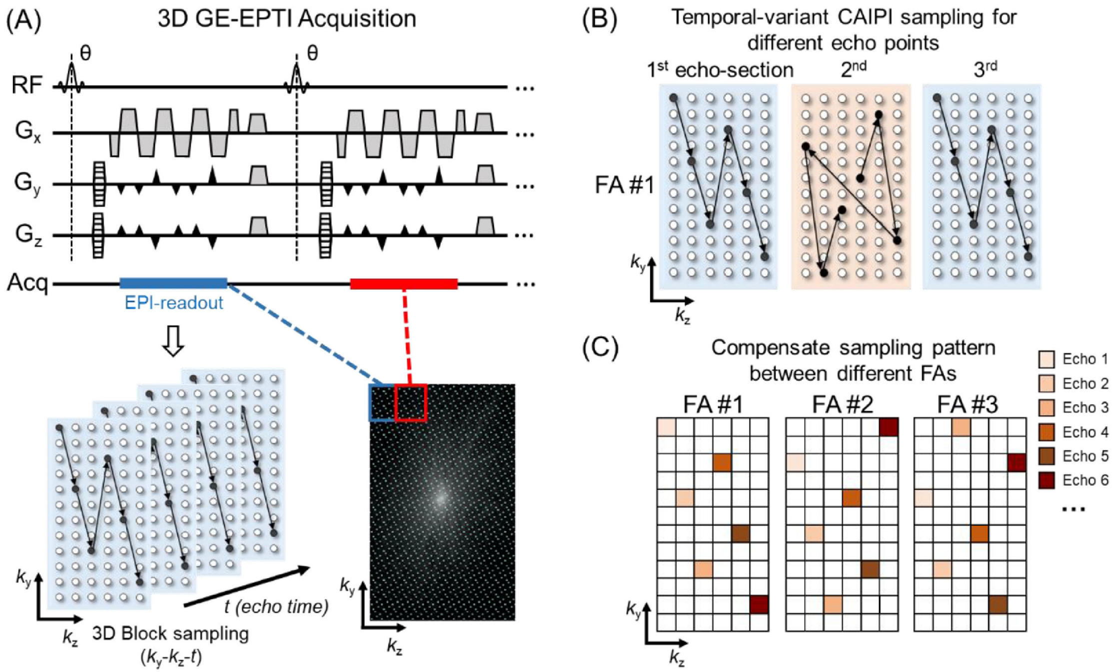
[8] Boyd S, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* (2010).

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

2 Examples

Outline

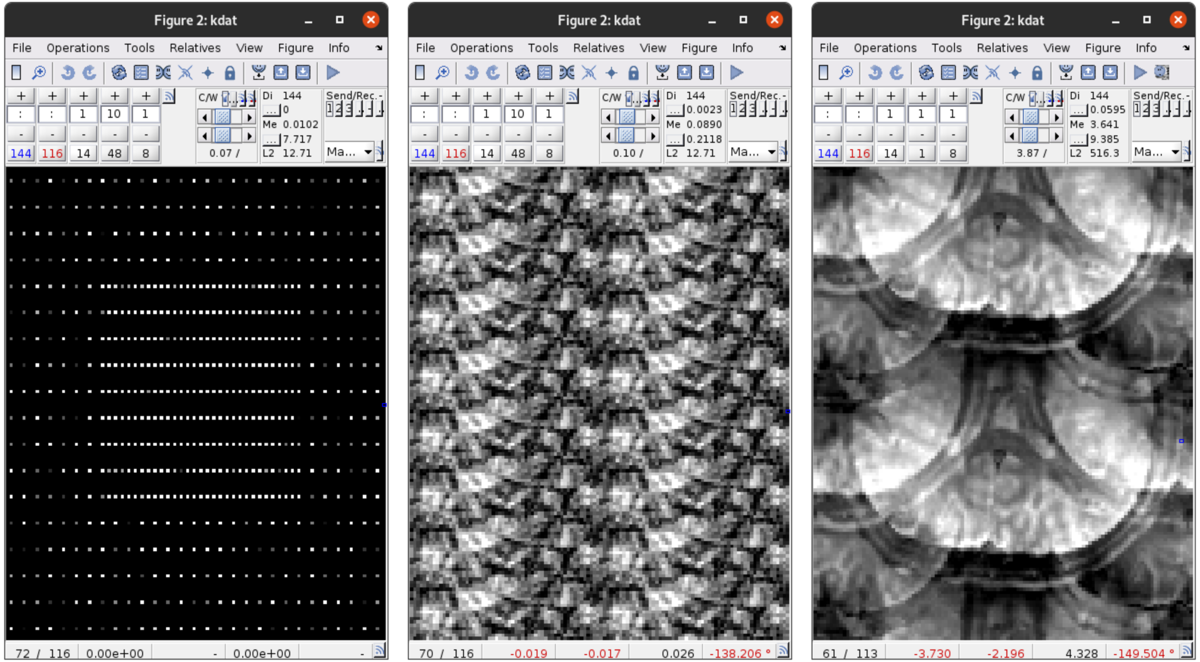
EPTI [9] Sequence Design



[9] Dong Z, et al. Variable flip angle echo planar time-resolved imaging (vFA-EPTI) for fast high-resolution gradient echo myelin water imaging. *NeuroImage* (2021).

EPTI Raw k-Space Data

ArrayShow [10]:



[10] Sumpf T. `https://github.com/tsumpf/arrShow`.

Reproducing EPTI Subspace Reconstruction

$$\arg\min_{\alpha} \left\| y - \mathcal{F}_u S \Phi \hat{U} \alpha \right\| + \lambda \left\| \mathrm{LLR}(\alpha) \right\|_1 \tag{6}$$

- $\hat{U}\alpha$ presents multi-echo multi-flip-angle images.
- $\Phi = e^{i2\pi f_{B_0} \mathrm{TE}_m}$ with f_{B_0} being the 2D B_0 field inhomogeneity map, which is estimated from reference scans.
- Both $\hat{U}$ and Φ can be implemented via the linop.Multiply(...) function in SigPy.

Reproducing EPTI Subspace Reconstruction

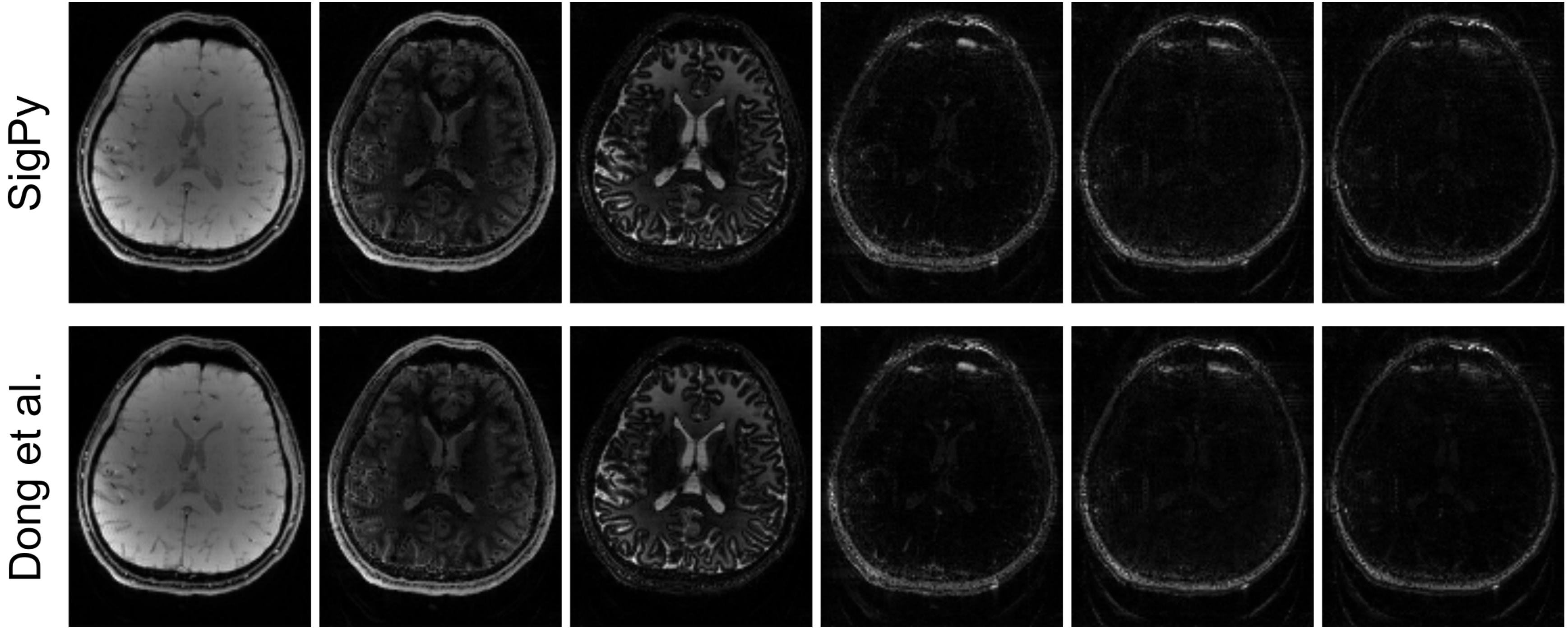
$$\arg\min_{\alpha} \left\| y - \mathcal{F}_u S \varPhi \hat{U} \alpha \right\| + \lambda \left\| \mathrm{LLR}(\alpha) \right\|_1 \tag{6}$$

- $\hat{U}\alpha$ presents multi-echo multi-flip-angle images.
- $\varPhi = e^{i2\pi f_{B_0} \mathrm{TE}_m}$ with f_{B_0} being the 2D B_0 field inhomogeneity map, which is estimated from reference scans.
- Both $\hat{U}$ and $\varPhi$ can be implemented via the linop.Multiply(...) function in SigPy.

```
img = app.SubspaceSenseRecon(kdat, coil, coil_dim=1,
                             lamda=0.005, regu='LLR',
                             basis=U, phase=phase,
                             blk_shape=(4, 8), blk_strides=(4, 8),
                             max_iter=20, solver='ADMM', rho=0.1,
                             device=backend.Device(0),
                             show_pbar=False).run()
```

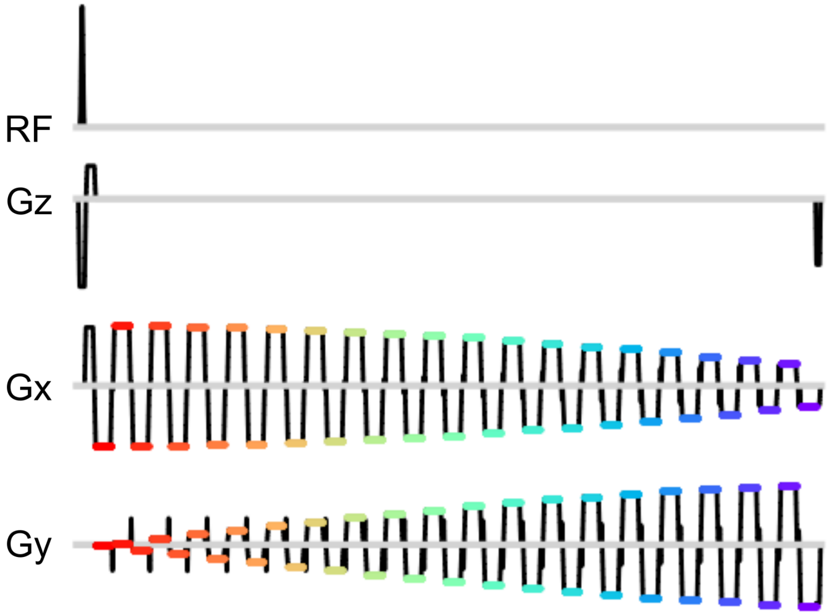
Reproducing EPTI Subspace Reconstruction

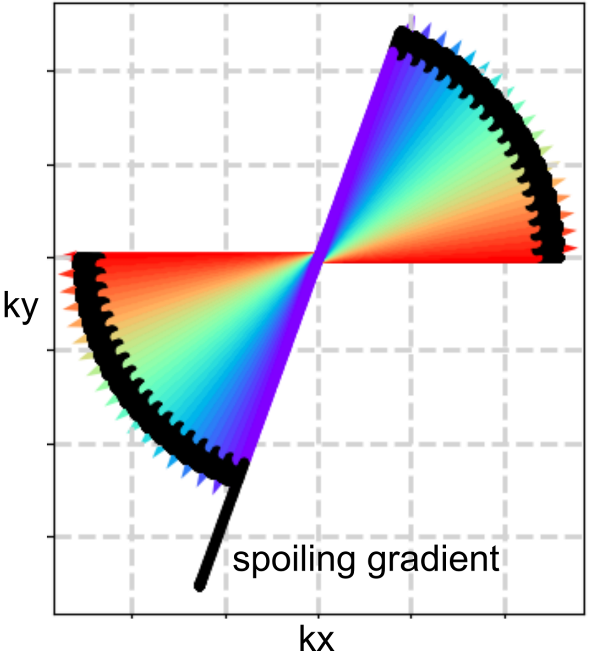
The first six subspace coefficient maps:



Outline

REPI [11] Sequence Design





[11] Tan Z, et al. Dynamic water/fat separation and B_0 inhomogeneity mapping - joint estimation using undersampled triple-echo multi-spoke radial FLASH. *Magn Reson Med* (2019).

REPI Sequence Design

REPI Acquisition Parameters:

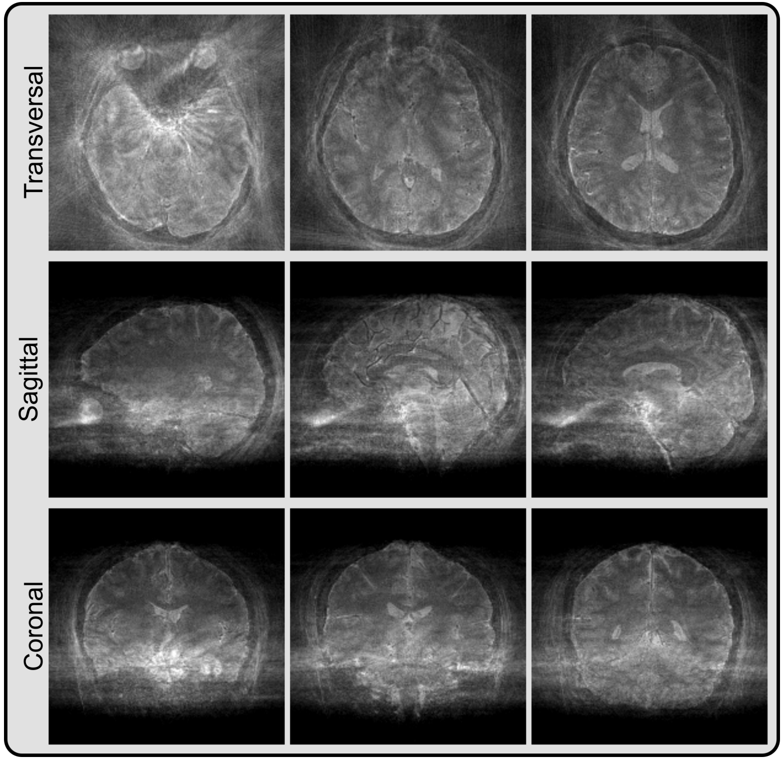
- flip angle 4 degree
- 1 mm^3 isotropic resolution
- image matrix $220 \times 220 \times 192$
- 192 slices with stack-of-stars 3D sampling [12]
- 7 shots per partition (slice)
- 35 echoes per shot with TE from 1.70 to 55.7 ms and TR 57.4 ms
- total scan time 1.3 minutes
- No reference scans.

$\rightarrow$ Acceleration factor: $R = 0.5\pi \times 220/7 \approx 49$.

[12] Block KT, et al. Towards routine clinical use of radial stack-of-stars 3D gradient-echo sequences for reducing motion sensitivity. *J Korean Soc Magn Reson Med* (2014).

REPI with Density-Compensated Adjoint NUFFT Reconstruction

- For non-Cartesian MRI, the $\mathcal{F}_u$ operator is implemented as linop.HDNUFFT, which creates a linop.NUFFT operator corresponding to every echo's sampling trajectory.
- Displayed images are echo- and coil-combined via root sum of square.



REPI with Linear Subspace Modeling

Build Dictionary

$$s_m = \rho \cdot e^{-\mathsf{TE}_m / T_2^*} \cdot e^{i2\pi f_{B_0} \mathsf{TE}_m} \tag{7}$$

- ρ: set as scalar 1.
- T_2^*: linearly distributed between 0.001 and 0.2 s with 100 atoms.
- f_{B_0}: linearly distributed between -50 and 50 Hz with 101 atoms.

REPI with Linear Subspace Modeling

Build Dictionary

$$s_m = \rho \cdot e^{-\mathsf{TE}_m/T_2^*} \cdot e^{i2\pi f_{B_0}\mathsf{TE}_m} \tag{7}$$

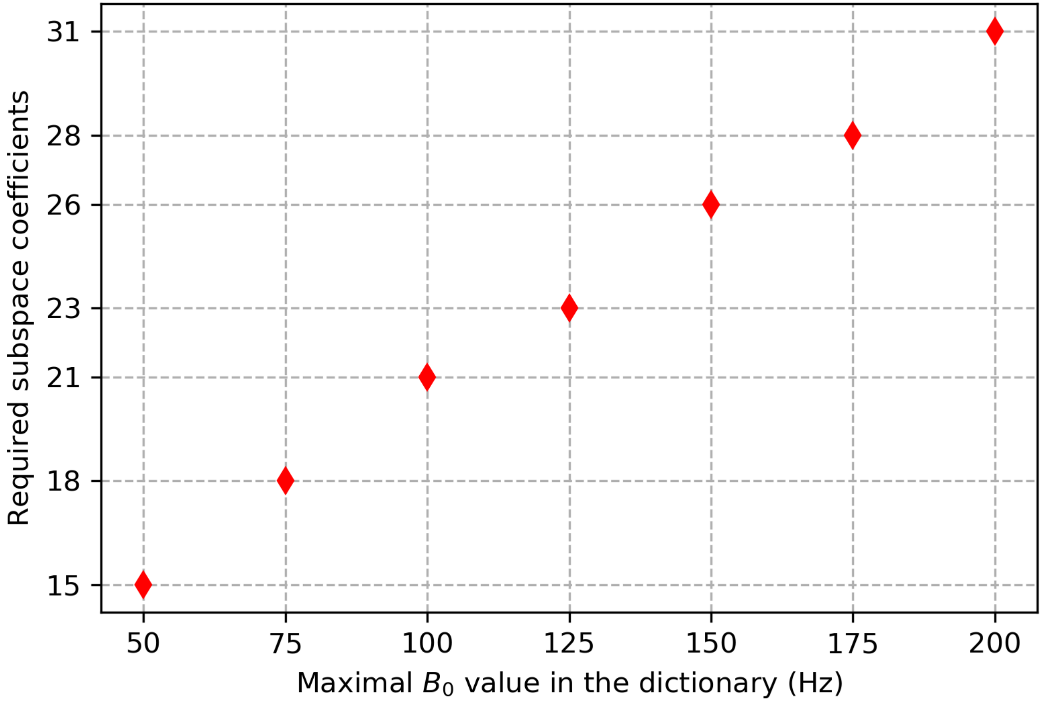
- ρ: set as scalar 1.
- T_2^*: linearly distributed between 0.001 and 0.2 s with 100 atoms.
- f_{B_0}: linearly distributed between -50 and 50 Hz with 101 atoms.

Extract Subspace Matrix $\hat{U}$

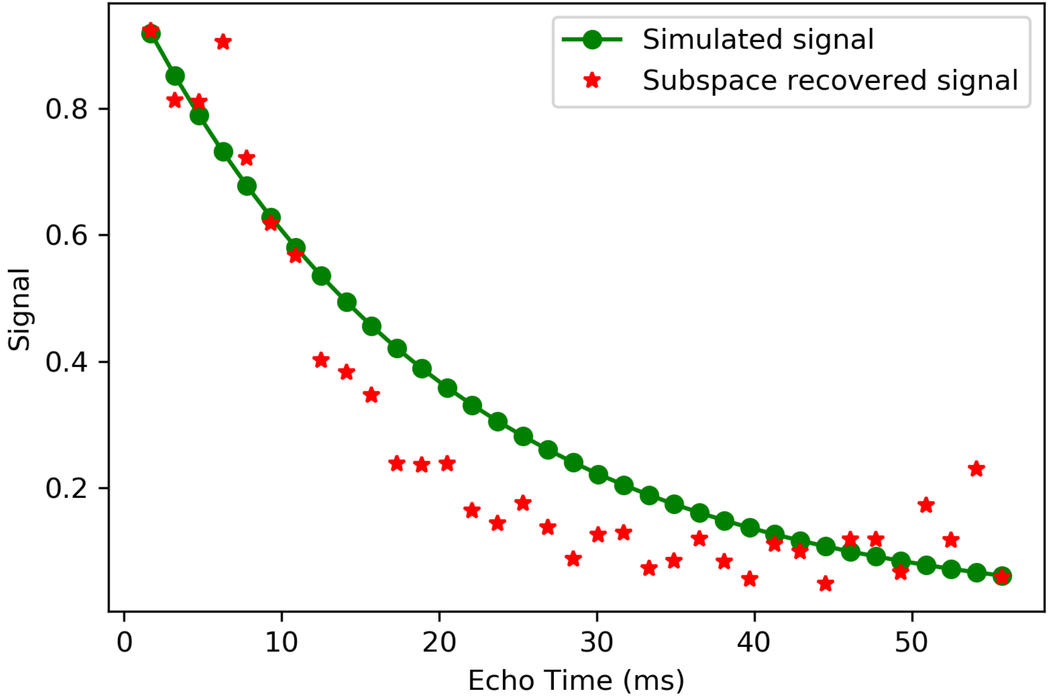
```
sig2 = np.reshape(sig, (sig.shape[-7], -1))
U, S, VH = np.linalg.svd(sig2, full_matrices=False)
U_sub = U[:, :num_coeffs]
recov_sig = U_sub @ U_sub.T @ sig2

err = get_rel_error(recov_sig, sig2)
```

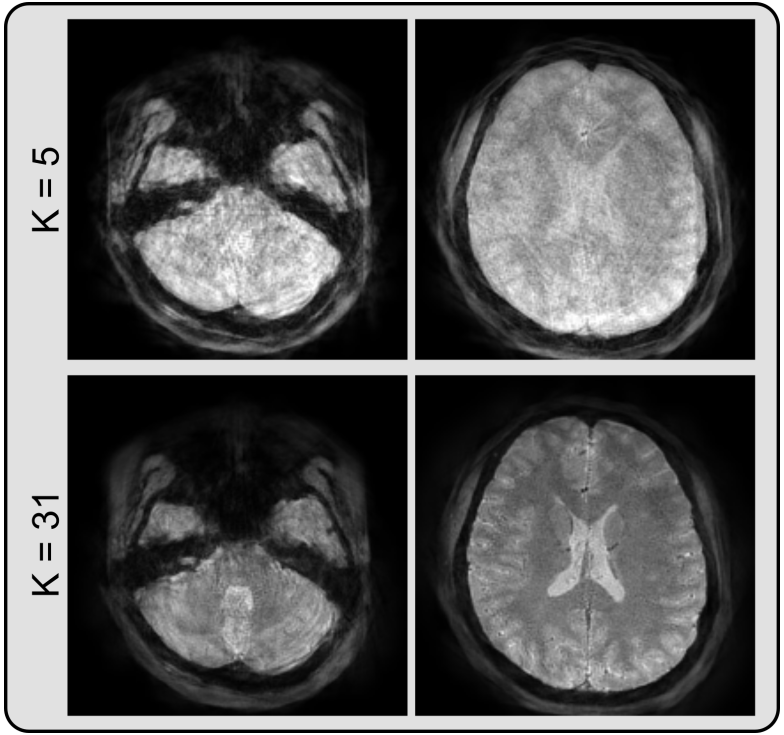
Larger B_0 Range Requires More Subspace Coefficients



Effects of Insufficient Subspace Coefficients



REPI with LLR Regularized Linear Subspace Reconstruction

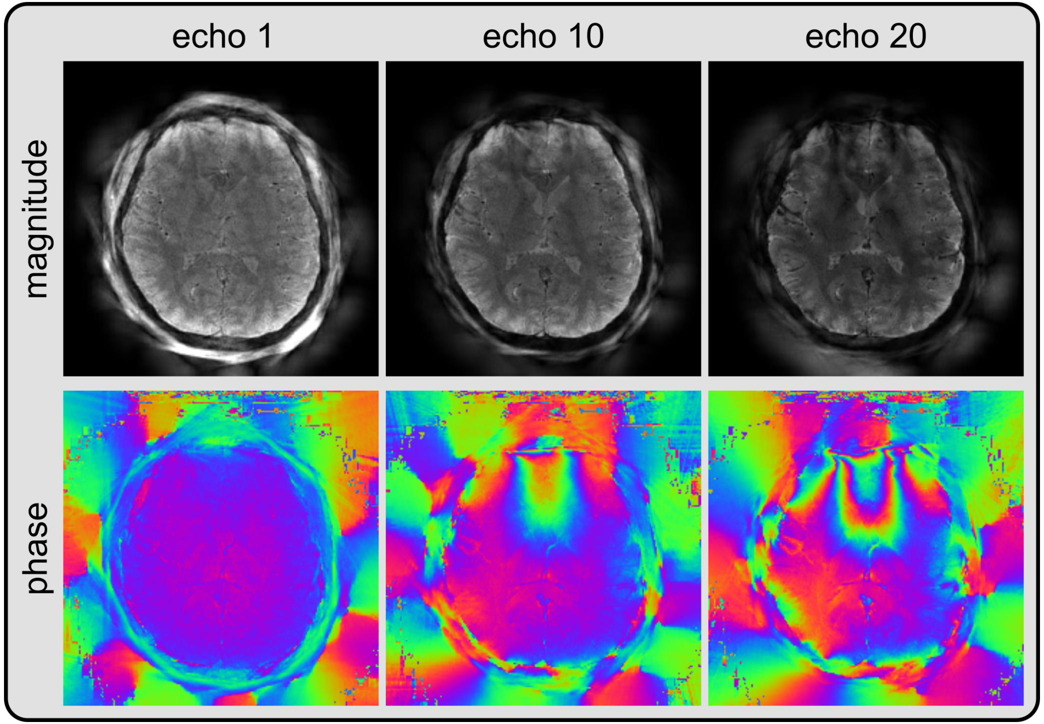


- ◇ Displayed are echo-combined images after reconstruction.
- ◇ Larger K (number of subspace coefficients) is necessary in the case of wide-range phase modulation in the dictionary of MGRE signal.
- ◇ The reconstruction time per slice for $K = 5$ and $K = 31$ was about 24 and 140 seconds, respectively.

REPI with LLR Regularized Linear Subspace Reconstruction [13]

[13] Tan Z, et al. ISMRM 2022;1860.

REPI with LLR Regularized Linear Subspace Reconstruction



- The magnitude and phase images of the 1st, 10th, and 20th echoes.
- Linear phase evolution along echoes.
- Residual streaking artifacts.

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

3 Summary

Summary

- ◇ This talk reviews the convenient linear operator abstraction in SigPy.
- ◇ Based on the generalized linear operator abstraction, I demonstrate an efficient implementation of locally low rank ℓ^1 soft thresholding in SigPy.
- ◇ I further implement a linear subspace reconstruction app, reproduce a recent EPTI reconstruction method, and apply this method to REPI data.

Thank You!

zhengguo.tan@gmail.com