Computational MRI

Parallel Imaging III

Non-Cartesian Imaging and Iterative Reconstruction

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

NYU

Overview

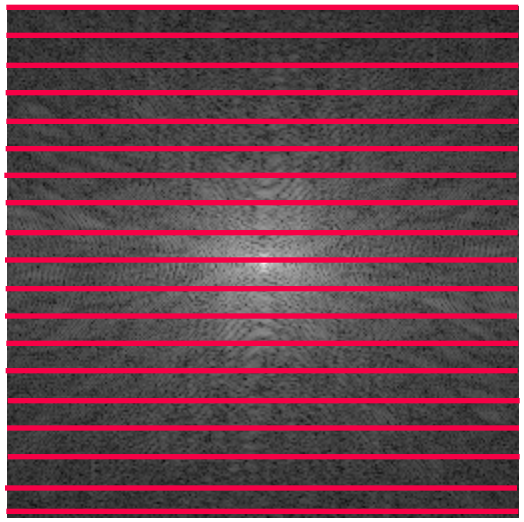
- Non-Cartesian parallel imaging
- Iterative image reconstruction
- The conjugate gradient algorithm and CG-SENSE

Magnetic Resonance in Medicine 46:638–651 (2001)

Advances in Sensitivity Encoding With Arbitrary *k*-Space Trajectories

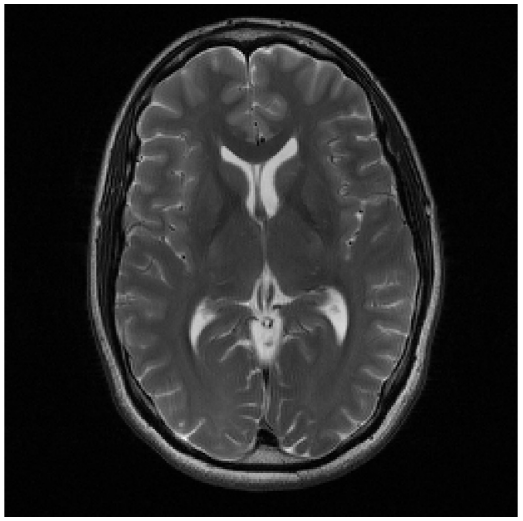
Klaas P. Pruessmann,[1] Markus Weiger,[1] Peter Börnert,[2] and Peter Boesiger[1*]

Cartesian subsampling



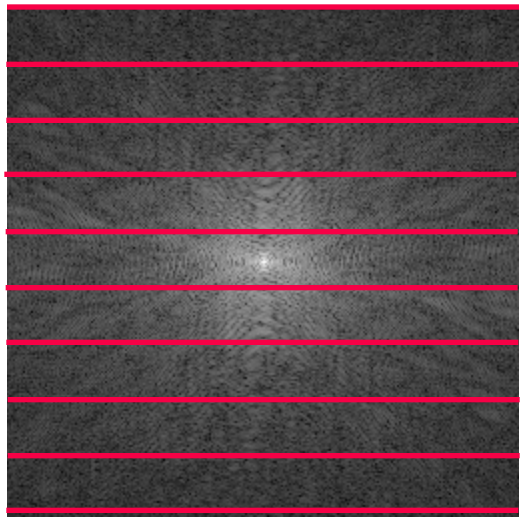
Δk_y

IFT



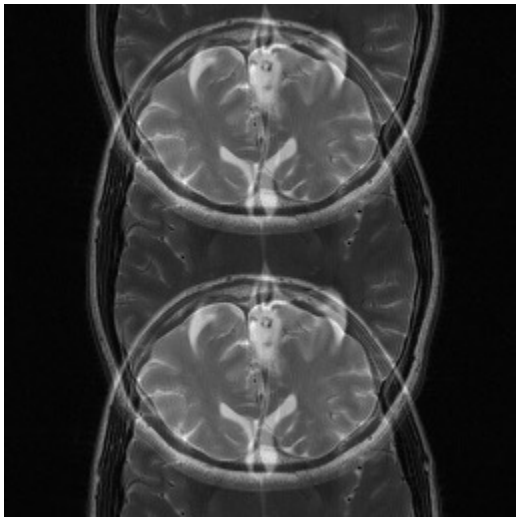
FOV

Cartesian subsampling



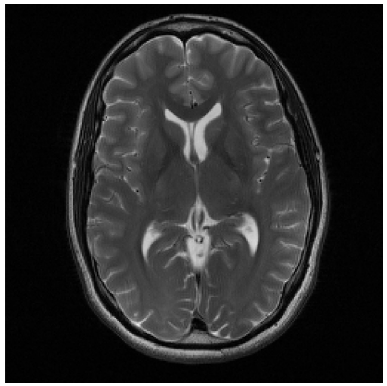
$2 \cdot \Delta k_y$

IFT

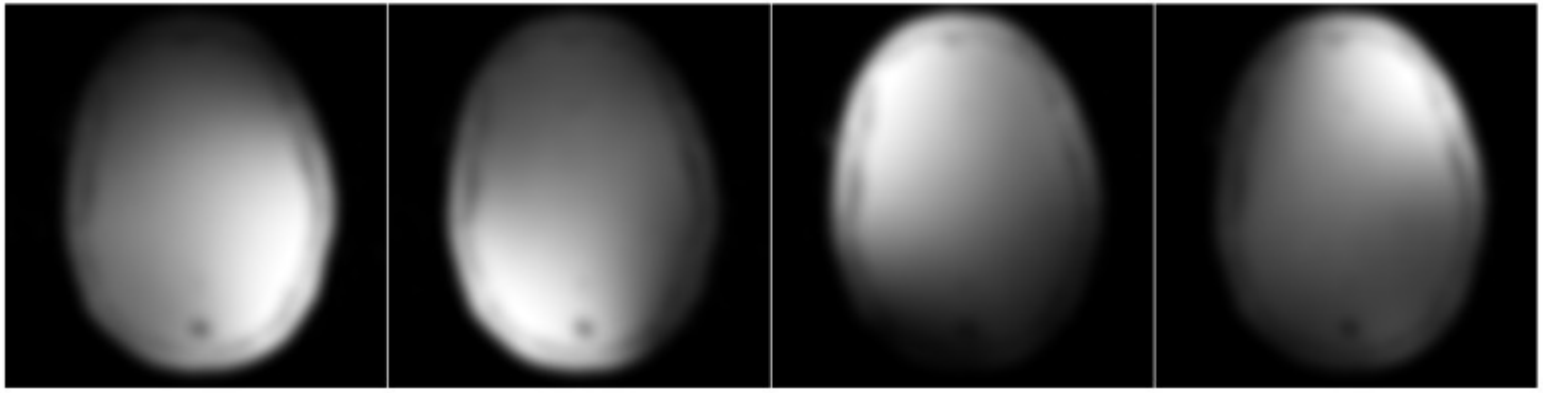


$\frac{FOV}{2}$

Cartesian SENSE

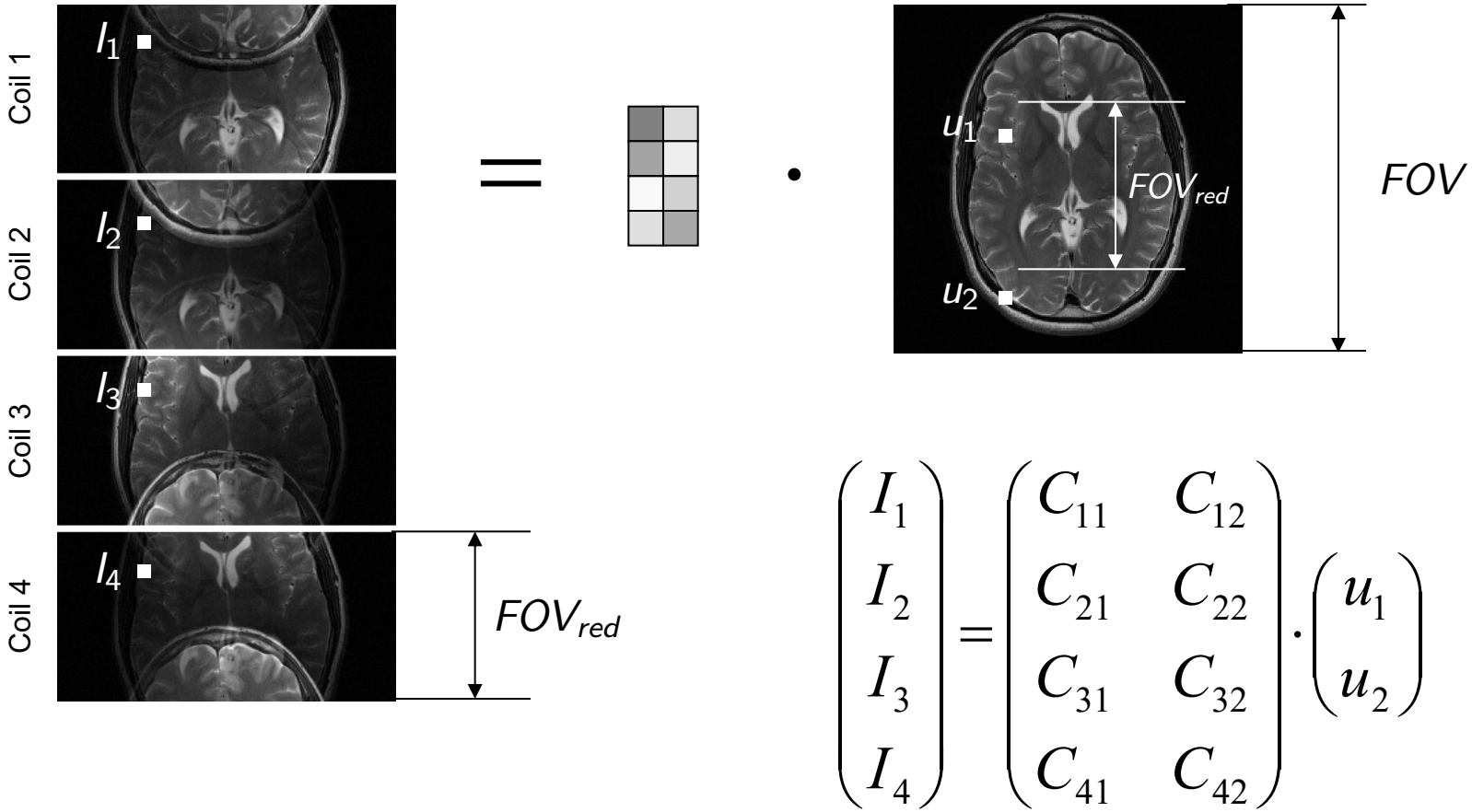


- T2 weighted brain scan
- 4-Channel receive coil
- Sensitivities are known



Pruessmann et al., MRM 42: 952-962 (1999)

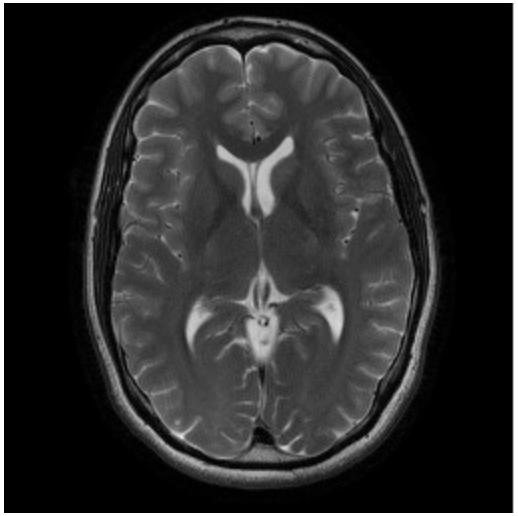
Cartesian SENSE



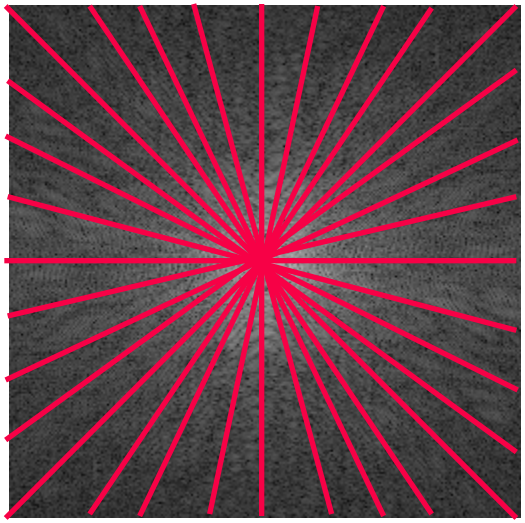
$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \\ C_{31} & C_{32} \\ C_{41} & C_{42} \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

Parallel Imaging: SENSE

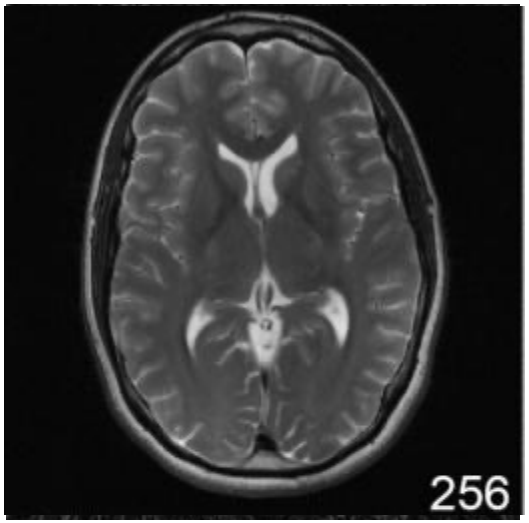
SENSE, R=2



Radial Subsampling



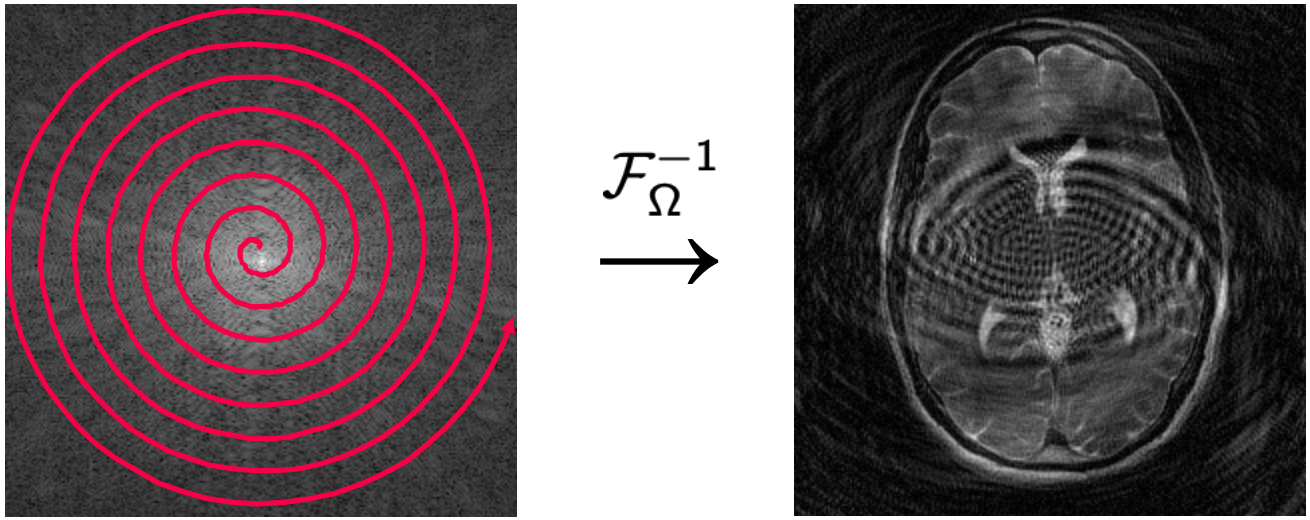
$\mathcal{F}_{\Omega}^{-1}$



R=8

Different structure of aliasing

Spiral Subsampling



Different structure of aliasing

Building the encoding operator

$$g_k(k_x, k_y) = \int\int c_k(x, y) e^{-i(k_x x + k_y y)} u(x, y) dx dy$$

Discretization

$$g_k(k_x, k_y) = \sum\sum c_k(x, y) e^{-i(k_x x + k_y y)} u(x, y)$$

Forward operator, encoding matrix

$$\vec{g} = \mathcal{F}_\Omega C \vec{u} = K \vec{u}$$

Image reconstruction as inverse problem

Encoding: $\vec{g} = K\vec{u}$ linear system of equations

Reconstruction:

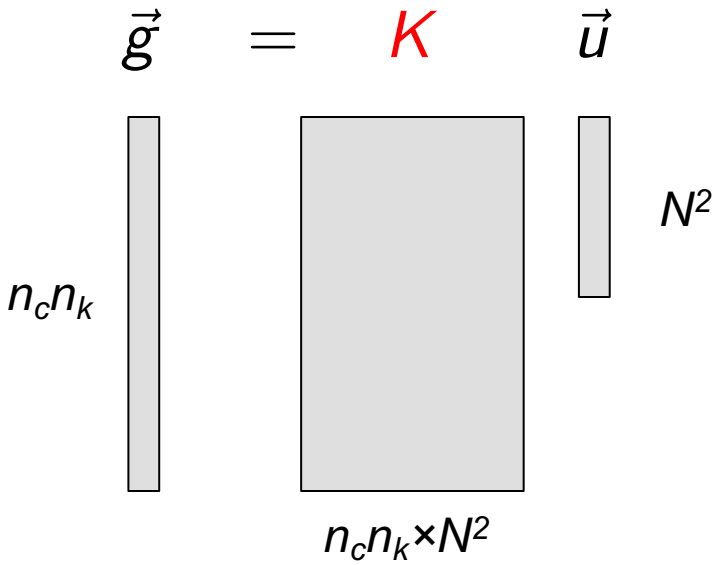
- Solution of the system of equations
- Inversion of the encoding matrix

$$\vec{u} = K^{-1}\vec{g}$$

Inversion of the encoding matrix

Direct methods: Pseudoinverse: $\vec{u} = (K^H K)^{-1} K^H \vec{g}$

Matrix K can become very large!



- Use iterative methods
- Gradient descent
- Conjugate Gradient algorithm

nk: number of sampling points in kspace

Magnus R and Stiefel E: Journal of Research of the National Bureau of Standards 49 (6) (1952)
Pruessmann et al., MRM 46: 638-651 (2001)

Iterative reconstruction

Given:

- Forward Operator: K
- Measured k-space data: $\vec{g}$

Forward problem

$$K\vec{u} = \vec{g}$$

Iterative reconstruction

$$\hat{\vec{u}} = \arg\min_{\vec{u}} \|K\vec{u} - \vec{g}\|_2^2$$

Iterative optimization: Gradient descent

$$K\vec{u} = \vec{g}$$

$$\hat{\vec{u}} = \arg\min_{\vec{u}} \|K\vec{u} - \vec{g}\|_2^2$$

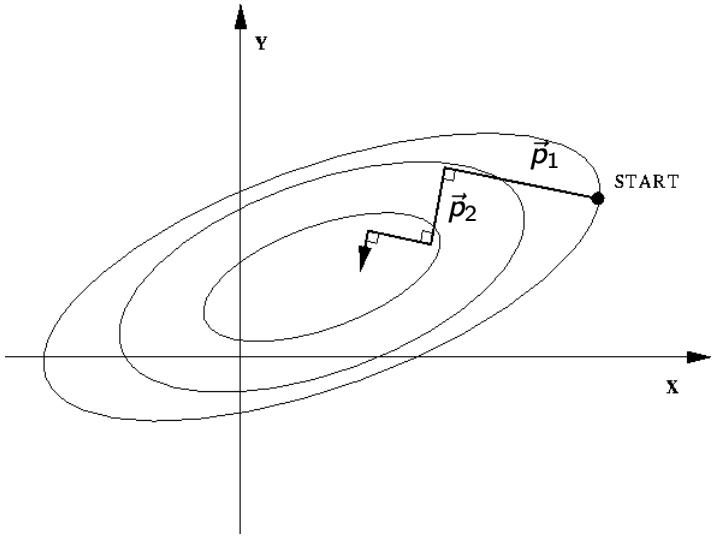
$$\vec{u}_{n+1} = \vec{u}_{n+1} - t\frac{\partial}{\partial \vec{u}}\|K\vec{u} - \vec{g}\|_2^2$$

$$\frac{\partial}{\partial \vec{u}}\|K\vec{u} - \vec{g}\|_2^2$$

$$\frac{\partial}{\partial \vec{u}}\|K\vec{u} - \vec{g}\|_2^2 = 2K^T(K\vec{u} - \vec{g})$$

Exercise :-)

Steepest descent



$$\vec{p}_k^T\,\vec{p}_{k+1} = 0$$

gI: radial traj kspace.
(num_points,num_spokes,num_coils)

GD implementation (exercise)

radial traj kspace
3维度

coil sen
3维

img:u
2维度

$$K : g_l = \mathcal{F}(c_l \odot u)$$

FourierTransform

ajoint conjungate

$$K^T : u = \sum_l c_l^T \odot (\mathcal{F}^T g_l)$$

from radial traj kspace to
cartesian image

FTgI: (256,256,8), num_cartesian_points, num_cartesian_points, num_coils

$$u_0 = 0$$

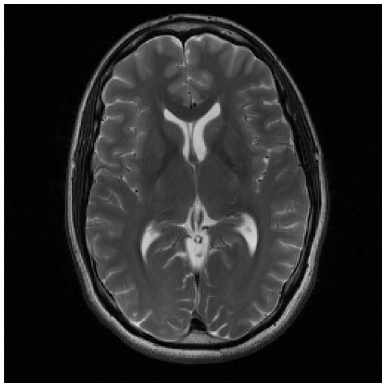
$$t > 0$$

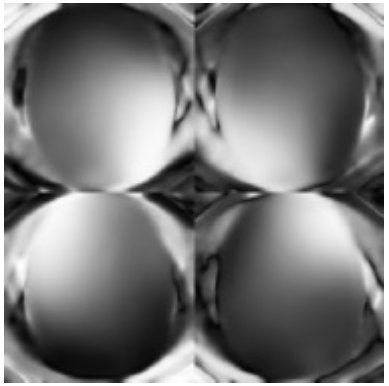
$$for : n = 1 : maxit$$

$$u_{n+1} = u_{n+1} - t(2K^T(Ku - g))$$

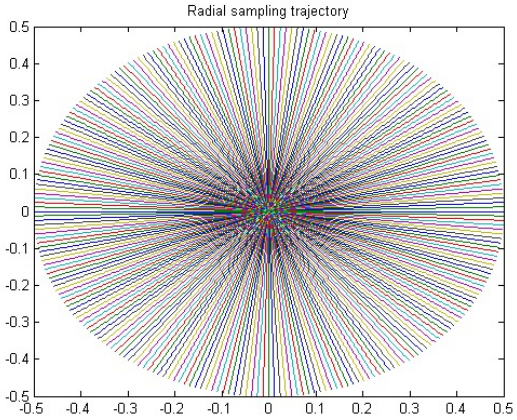
Exercise example

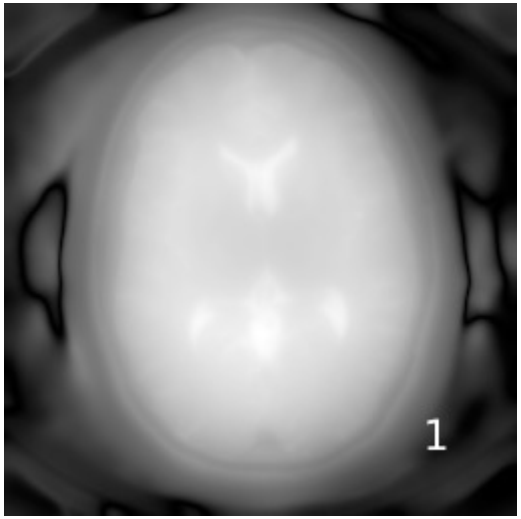
- T2 weighted brain scan
- 3T System
- 4 channel head coil
- TSE: Sequence parameters:
  - TR 5000ms
  - TE 99ms
  - Turbo Factor 10
  - Matrix Size 256x256
  - In Plane Resolution 0.9mmx0.9mm
  - Slice Thickness 4mm

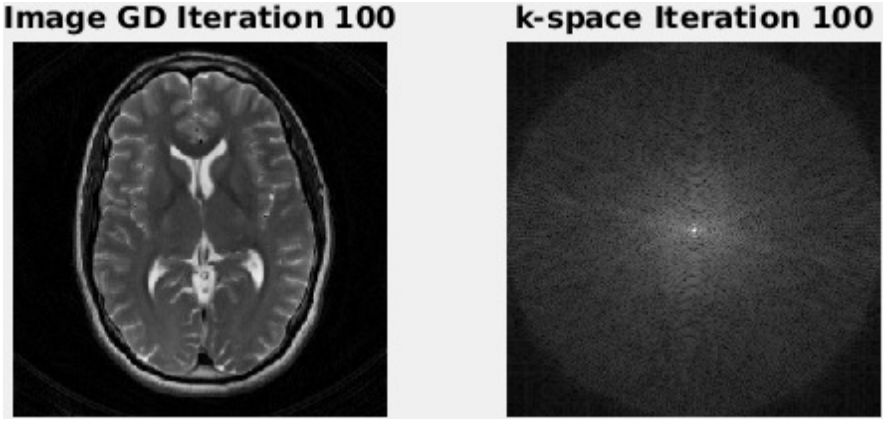




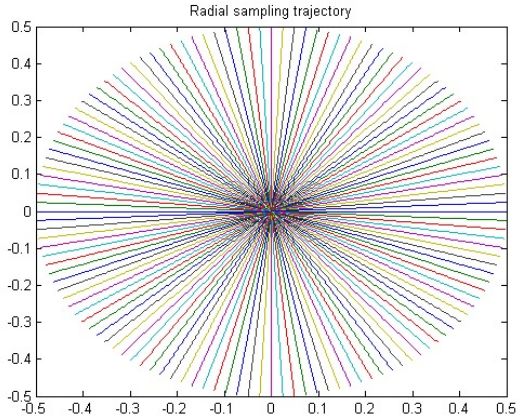
GD radial trajectory, 128 projections, R=2

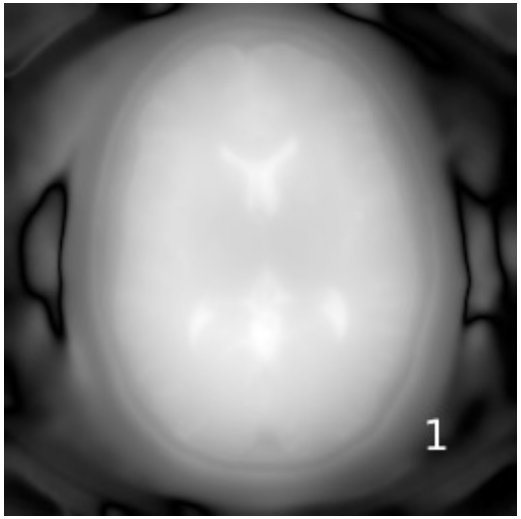


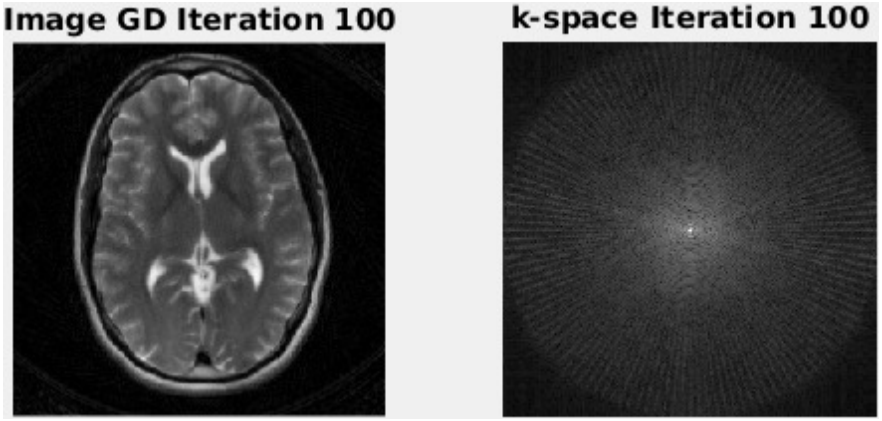




GD radial trajectory, 64 projections, R=4







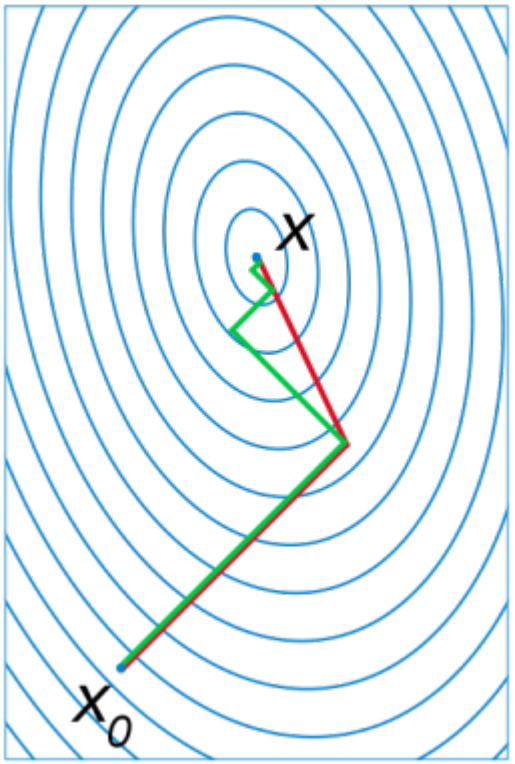
The conjugate gradient method

Minimize $f(\vec{u})$ in subspace spanned by search directions that are *K* orthogonal (conjugate):

$$\vec{p}_k^T K \vec{p}_{k+1} = 0$$

Converges to exact solution in *n* steps (n is matrix size)

Source: Wikipedia



n=2 quadratic form
Gradient descent (red)
Conjugate gradient minimization (green)

The conjugate gradient method

$$\vec{r}_k = \vec{g} - K\vec{u}_k$$ Residual

$$\vec{p}_k = \vec{r}_k - \sum_{i<k} \frac{\vec{p}_i^T K \vec{r}_k}{\vec{p}_i^T K \vec{p}_i} \vec{p}_i$$ CG search direction

$$\vec{u}_{k+1} = \vec{u}_k + \alpha \vec{p}_k$$ Update step

$$\alpha_k = \frac{\vec{p}_k^T \vec{r}_{k-1}}{\vec{p}_k^T K \vec{p}_k}$$ Step length

The conjugate gradient method

- Exact solution: Interpretation as direct method
- In practice: Used as iterative method
  - Unstable to perturbations, e.g. noise
  - Required tolerance usually achieved after $it << n$
- Very easy to implement (≈10 lines in Matlab, ≈10 minutes of work)
- Implementations for Matlab, Numerical Recipes etc.

```
%% CG iterations
maxit=size(A,1);
x = 0; r = b; p = r; rr = r'*r;
for it = 1:maxit
    alpha = rr/(p'*A*p);
    x = x + alpha*p;
    rnew = r - alpha*A*p;
    beta = (rnew'*rnew)/rr;
    r=rnew;
    rr = r'*r;
    p = r + beta*p;
end
```

```
% Use Matlab CG
u = pcg(K,g(:),tol,maxit);
```

CG on the normal equations and preconditioning

$K\vec{u} = \vec{g}$ Requirement: K symmetric and positive definite

$K^T K\vec{u} = K^T\vec{g}$ CG on the normal equations

$E^{-1}K(E^{-1})^T\tilde{\vec{u}} = E^{-1}\vec{g}$

with $\tilde{\vec{u}} = E^T\vec{u}$ and $EE^T = M$

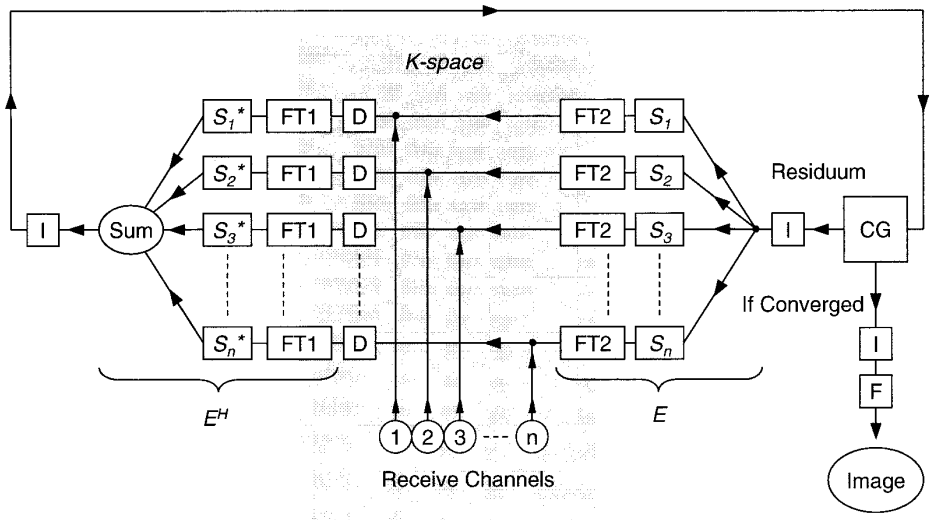
Preconditioning: „Modify equation such that right hand side is approximate solution“

Improve condition number

Magnetic Resonance in Medicine 46:638–651 (2001)

Advances in Sensitivity Encoding With Arbitrary *k*-Space Trajectories

Klaas P. Pruessmann,[1] Markus Weiger,[1] Peter Börnert,[2] and Peter Boesiger[1*]



CG — Conjugate gradient method

I — Intensity correction

D — Density correction

S_γ — Multiply by sensitivity of γ-th coil

S_γ^* — Multiply by complex conjugate of sensitivity of γ-th coil

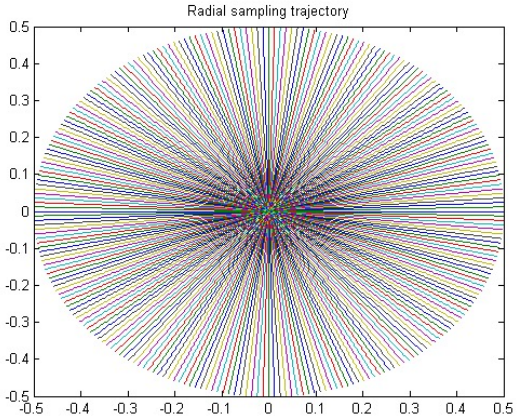
Sum — Complex sum of input images

F — K-space filter

FT1 — Fourier Transform to image domain
1. Convolve with gridding kernel
2. Oversample 2x along Cartesian grid
3. FFT to image domain
4. Trim image matrix
5. Divide by Fourier transform of gridding kernel

FT2 — Fourier Transform to k-space
1. Divide by Fourier transform of gridding kernel
2. Double image matrix and zero-pad
3. Inverse FFT to k-space
4. Convolve with gridding kernel
5. Resample along non-standard trajectory

CG radial trajectory, 128 projections, R=2



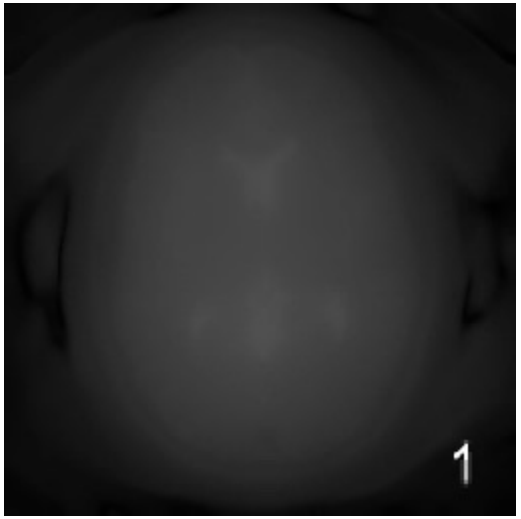
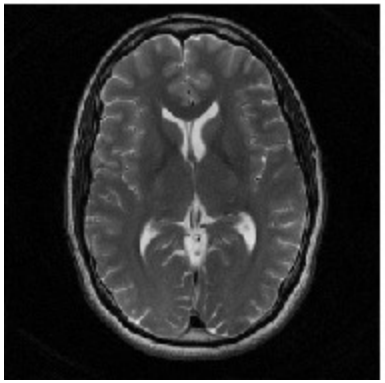
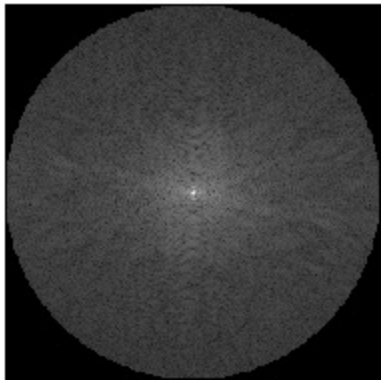


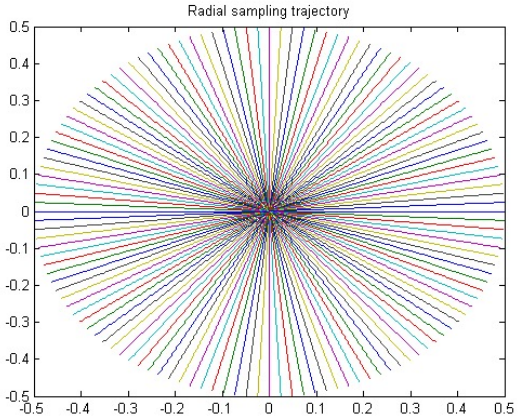
Image CG iteration 40



k-space iteration 40



CG radial trajectory, 64 projections, R=4



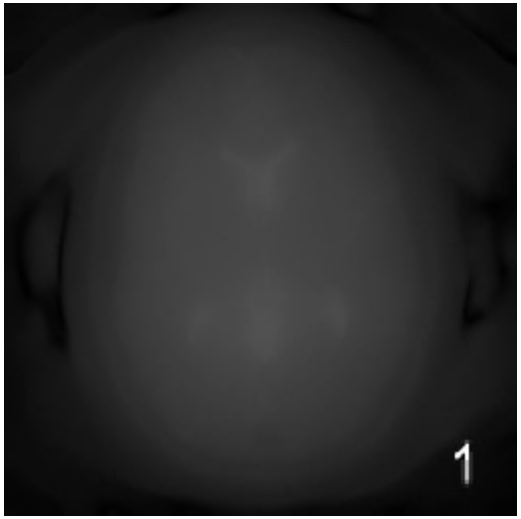
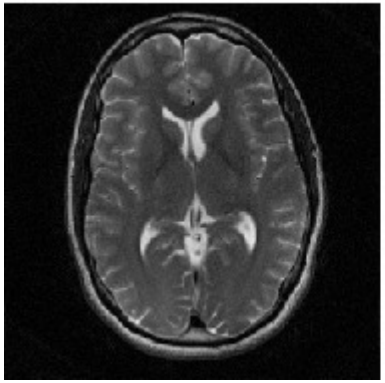
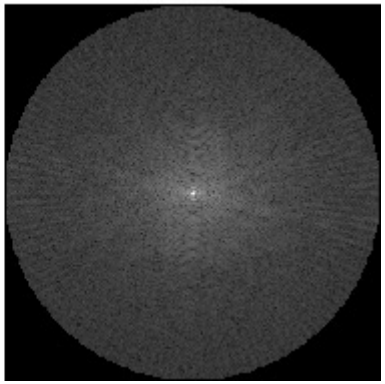
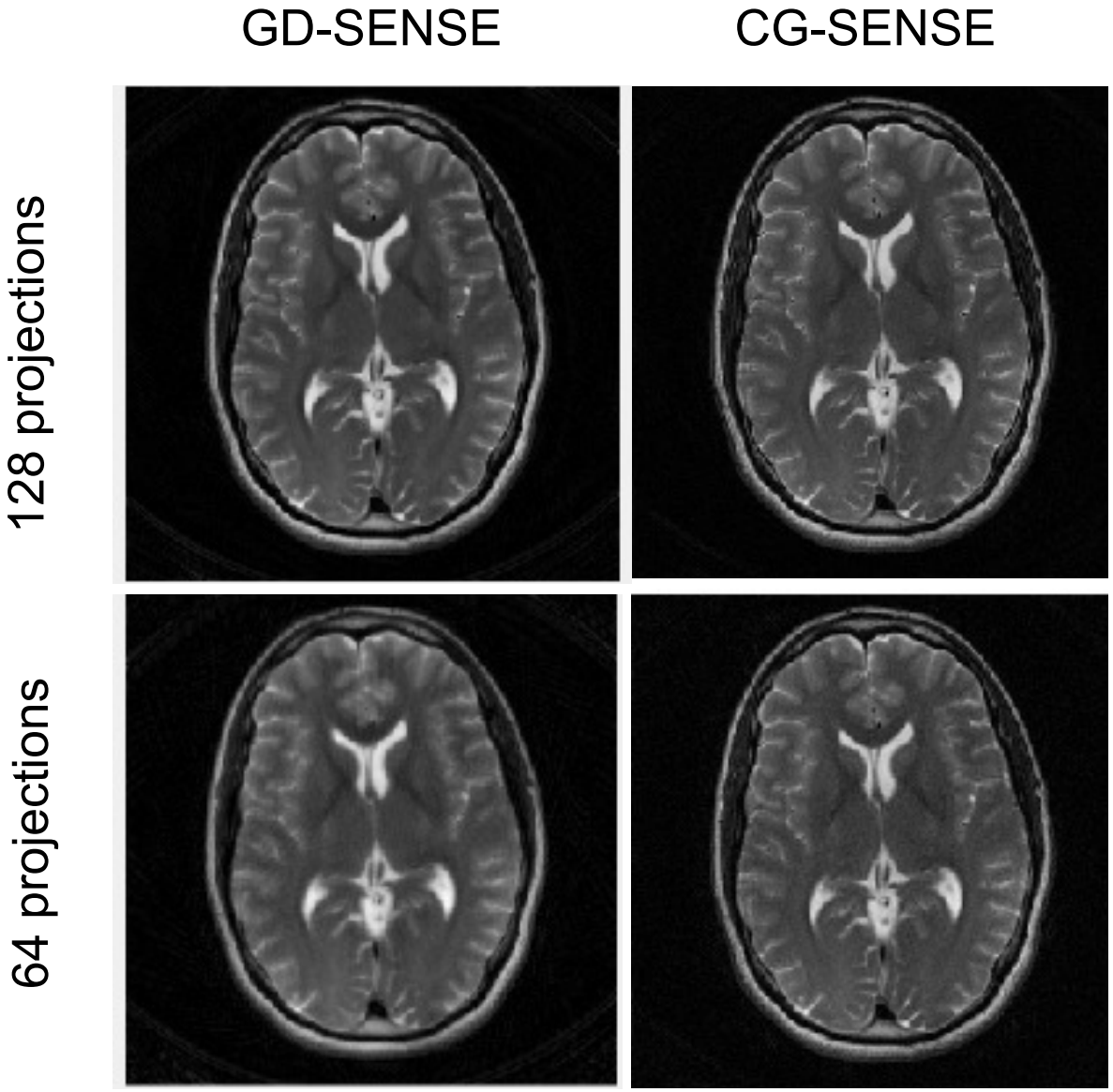


Image CG iteration 40

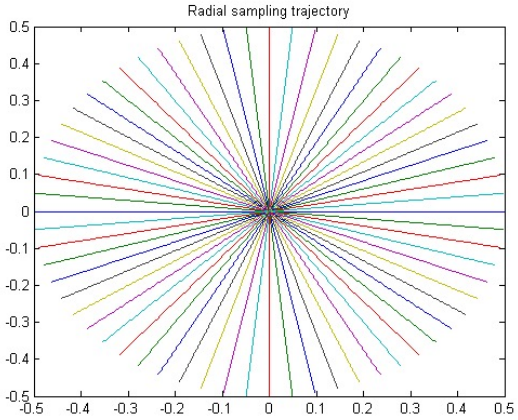


k-space iteration 40




GD-SENSE
CG-SENSE
128 projections
64 projections

CG radial trajectory, 32 projections, R=8



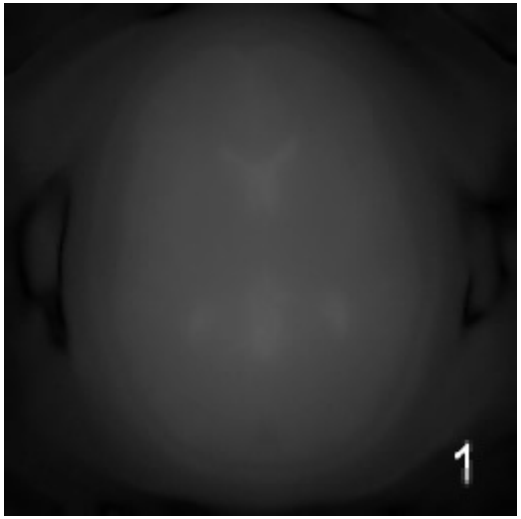
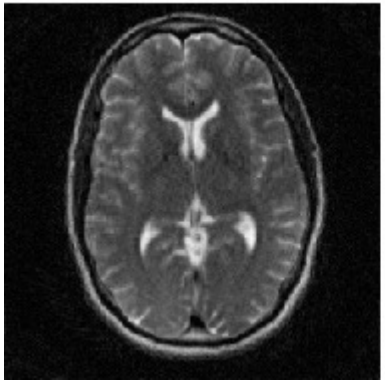
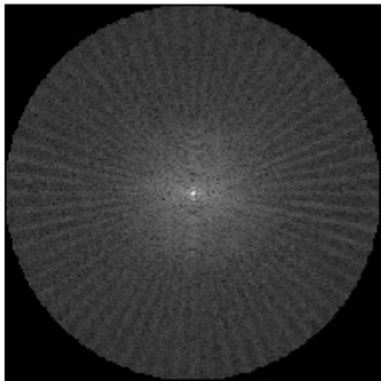


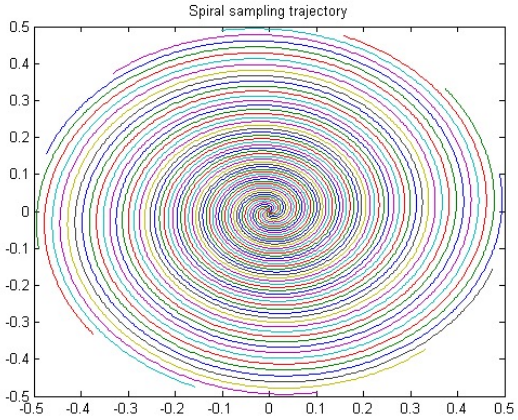
Image CG iteration 40



k-space iteration 40



CG spiral trajectory, 12 interleaves, R≈4



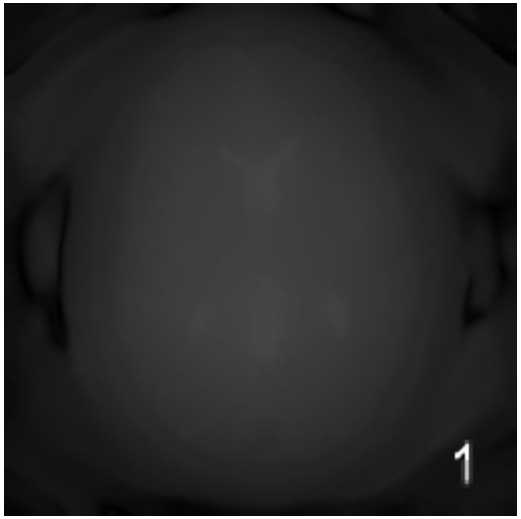
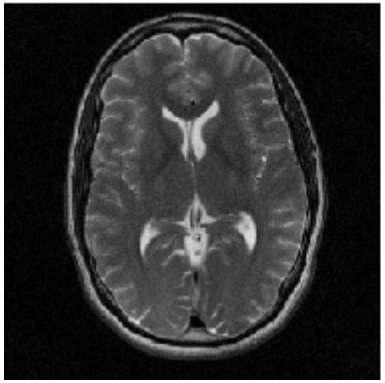
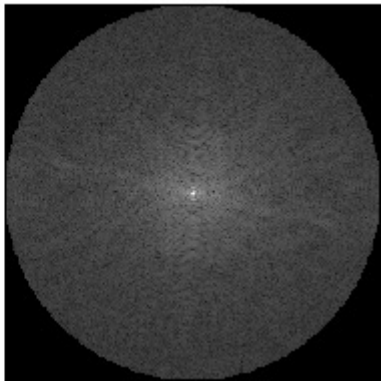


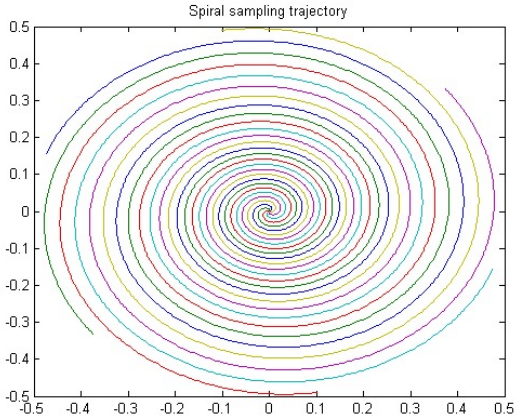
Image CG iteration 40



k-space iteration 40



CG spiral trajectory, 6 interleaves, R≈8



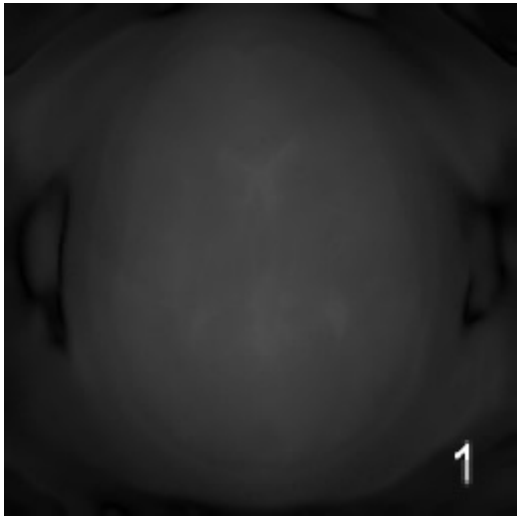
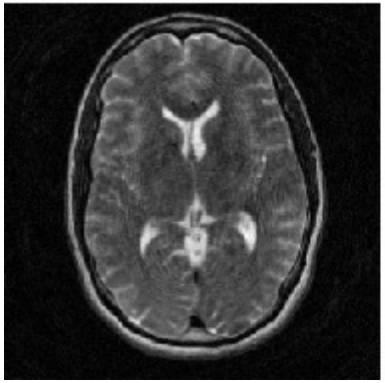
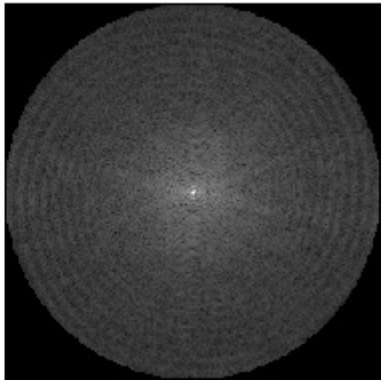


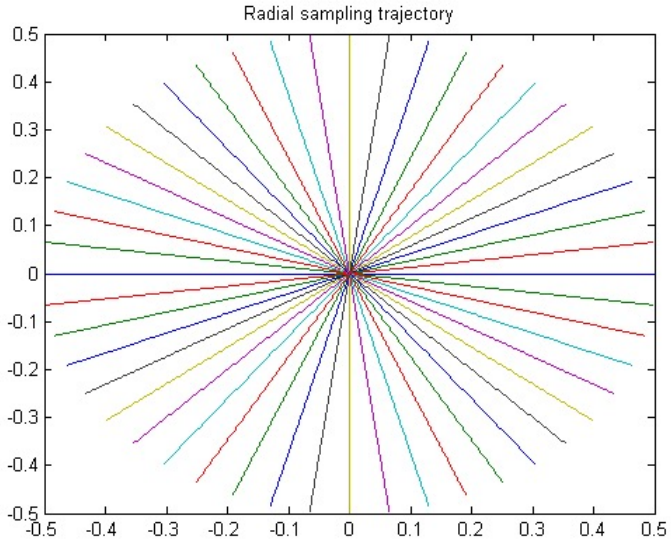
Image CG iteration 40

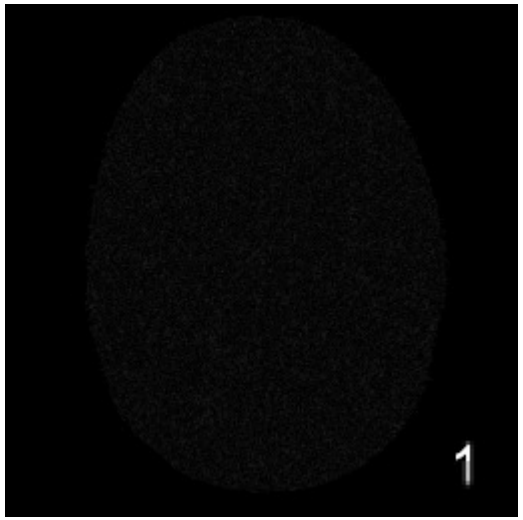


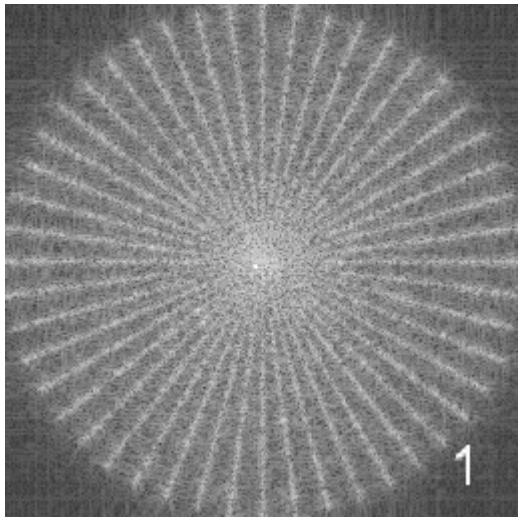
k-space iteration 40



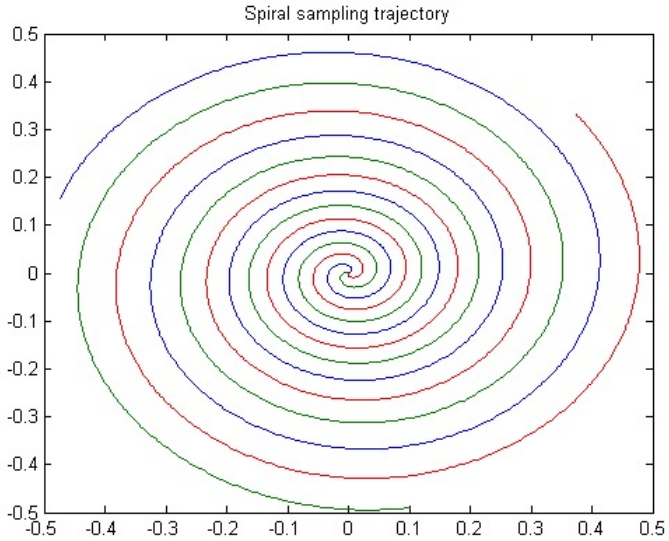
Noise propagation: Radial trajectory, 24 projections

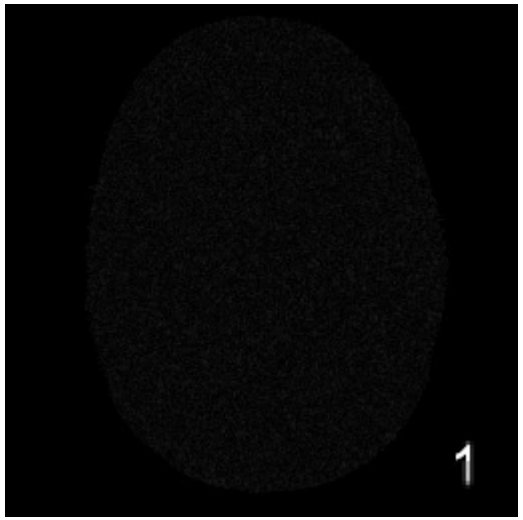


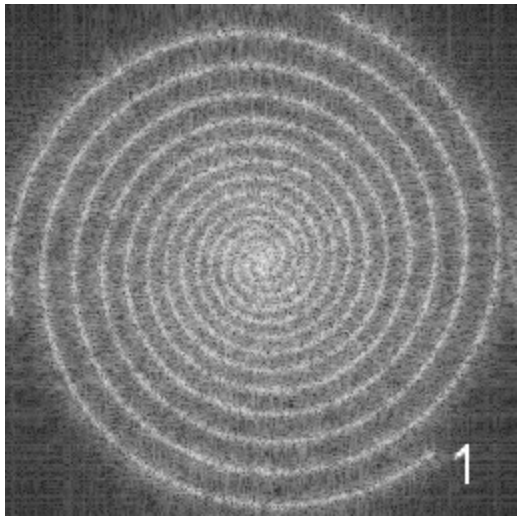




Noise propagation: Spiral trajectory, 3 interleaves



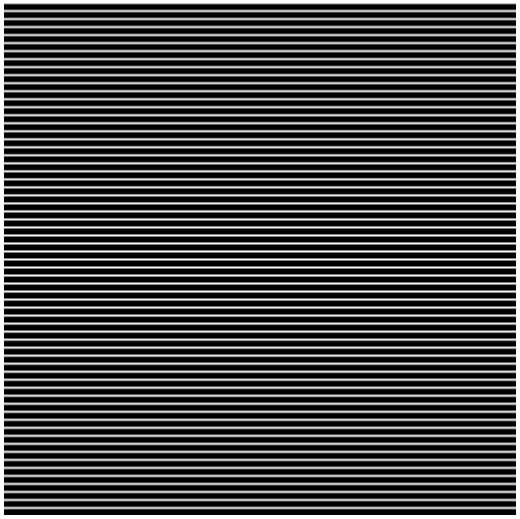


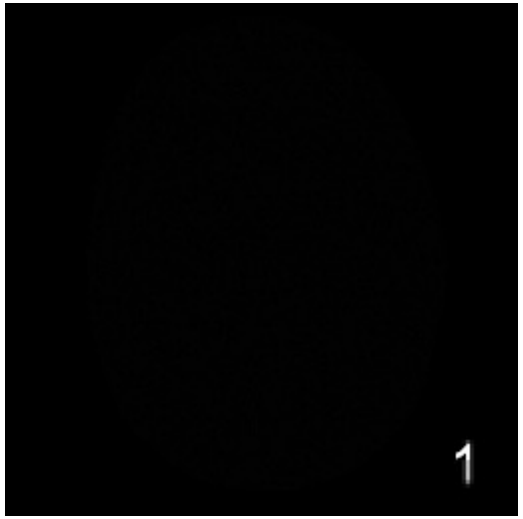


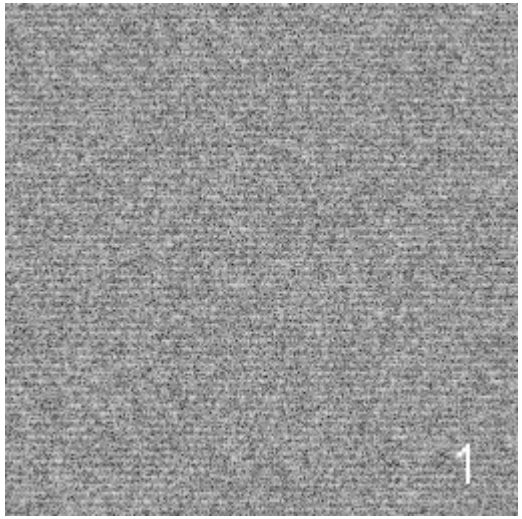
Noise propagation: Cartesian, R=4

noise in image

FFT ,kspace







Acceleration is limited

- Acceleration factor approaches number of coils
- Severe noise amplification
- Residual artifacts
- Ill conditioning of the encoding matrix
- Regularize the inversion:
  - CG SENSE: Stopping after a certain amount of iterations
  - Regularization constraints tailored to imaging problem

Received: 10 August 2020 | Revised: 2 October 2020 | Accepted: 2 October 2020

DOI: 10.1002/mrm.28569

REVIEW

Magnetic Resonance in Medicine

CG-SENSE revisited: Results from the first ISMRM reproducibility challenge

Oliver Maier[1] | **Steven Hubert Baete**[2] | **Alexander Fyrdahl**[3] | **Kerstin Hammernik**[4,5] | **Seb Harrevelt**[6] | **Lars Kasper**[7,8,9] | **Agah Karakuzu**[10] | **Michael Loecher**[11] | **Franz Patzig**[7] | **Ye Tian**[12,13] | **Ke Wang**[14] | **Daniel Gallichan**[15] | **Martin Uecker**[16,17,18,19] | **Florian Knoll**[2]

[1]Institute of Medical Engineering, Graz University of Technology, Graz, Austria

[2]Center for Biomedical Imaging, New York University School of Medicine, New York, NY, USA

[3]Department of Clinical Physiology, Karolinska University Hospital, and Karolinska Institutet, Stockholm, Sweden

[4]Department of Computing, Imperial College London, London, UK

[5]Institute of Computer Graphics and Vision, Graz University of Technology, Graz, Austria

[6]Department of Biomedical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

[7]Institute for Biomedical Engineering, ETH Zurich and University of Zurich, Zurich, Switzerland

[8]Translational Neuromodeling Unit, Institute for Biomedical Engineering, University of Zurich and ETH Zurich, Zurich, Switzerland

[9]Techna Institute, University Health Network, Toronto, ON, Canada

[10]NeuroPoly Lab, Institute of Biomedical Engineering, Polytechnique Montréal, Montréal, QC, Canada

[11]Department of Radiology, Stanford University, Stanford, CA, USA

[12]Utah Center for Advanced Imaging Research (UCAIR), Department of Radiology and Imaging Sciences, University of Utah, Salt Lake City, UT, USA

[13]Ming Hsieh Department of Electrical and Computer Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

[14]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA

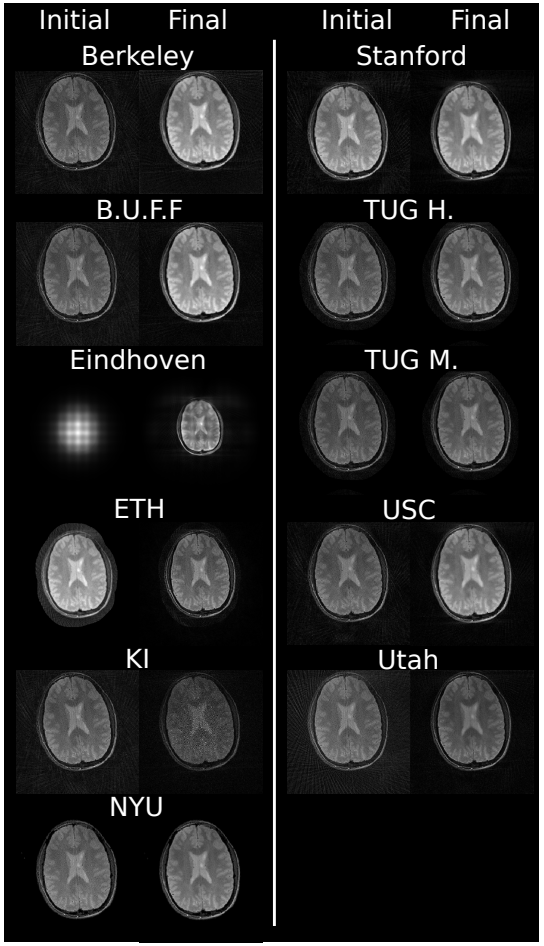
[15]Cardiff University Brain Research Imaging Centre, Cardiff, UK

[16]Institute for Diagnostic and Interventional Radiology, University Medical Center Göttingen, Göttingen, Germany

[17]German Centre for Cardiovascular Research (DZHK), Berlin, Germany

[18]Cluster of Excellence "Multiscale Bioimaging: from Molecular Machines to Networks of Excitable Cells" (MBExC), University of Göttingen, Göttingen, Germany

[19]Campus Institute Data Science (CIDAS), University of Göttingen, Göttingen, Germany



Can you reproduce this seminal MRM paper? Participate in the reproducible research study group challenge!

April 2, 2019

If you believe spending time on reproducibility makes SENSE, we encourage you to join the ISMRM reproducible research study group's 2019 challenge! As the challenge unfolds, so will sub-Nyquist aliasing artifacts. Have we dropped enough hints about the MRM paper selected to be reproduced?

Yes, you probably guessed right. The paper selected for the 2019 reproducibility challenge is:

Klaas P. Pruessmann, Markus Weiger, Peter Börnert, Peter Boesiger. *Advances in sensitivity encoding with arbitrary k-space trajectories.* Magn Reson Med. 2001 Oct;46(4):638-51.

The data

We provided two example datasets, brain (12 receive channels, 96 radial projections) and cardiac (34 receive channels, 55 radial projections), from a radial trajectory acquired with multi-channel coils. The data is provided in the h5 format, and we are following the conventions of the BART toolbox regarding array dimensions of the raw data [1, Readout, Spokes, Channels] and the trajectory [3, Readout, Spokes] where the first dimension encodes the k-space coordinate (for 2D acquisitions the third coordinate is always zero) and the unit of measurement is 1 / FOV.

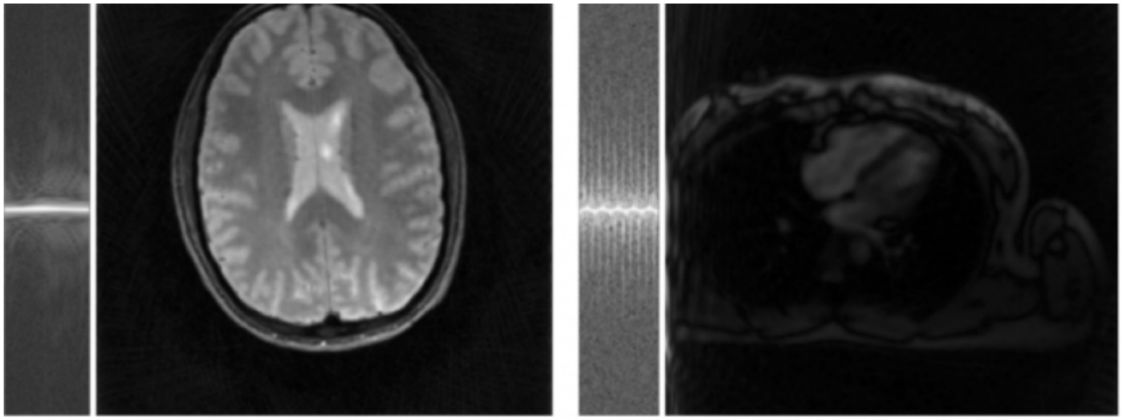


Figure 1: Raw k-space data from one coil and a gridding sum of squares example reconstruction of the provided brain (left) and cardiac (right) data

Brain data (5.3 MB): rawdata_brain_radial_6proj_12ch.h5: rawdata: [1, 512, 96, 12], trajectory: [1, 512, 96]

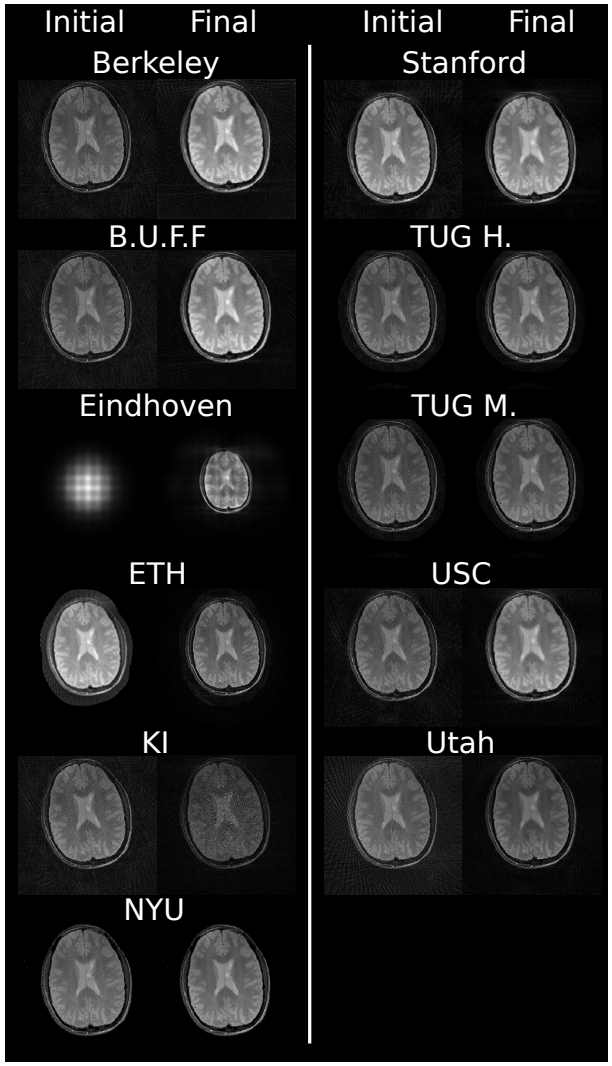
Cardiac data (5 MB): rawdata_heart_radial_55proj_34ch.h5: rawdata: [1, 320, 55, 34], trajectory: [1, 320, 55]

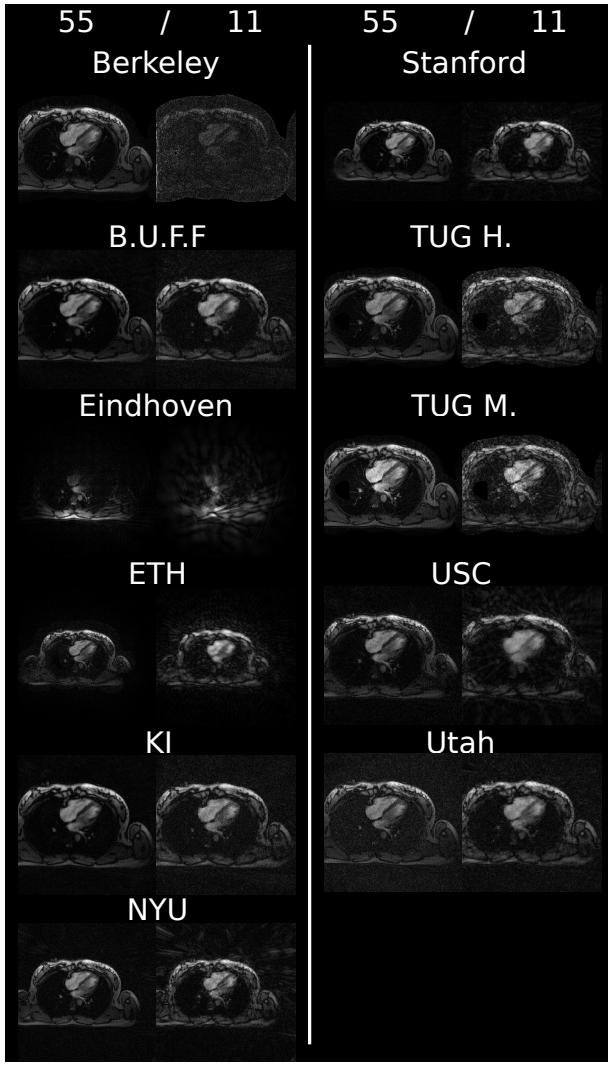
We also provided starter-scripts for MATLAB and Python which are able to read in and display the data. These scripts are also available from the Github repository.

Submissions

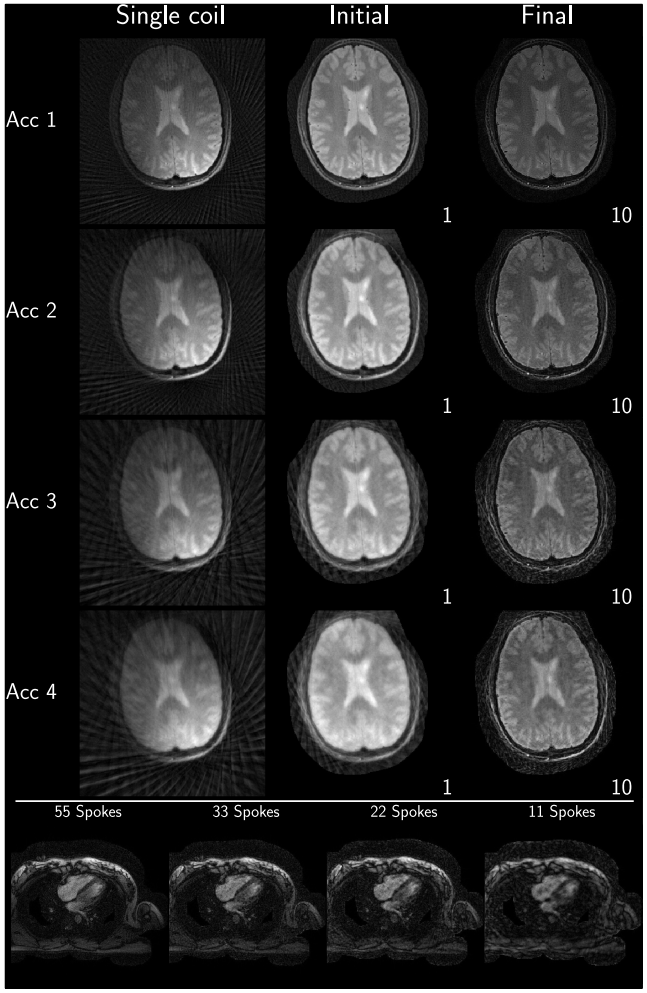
Collated list of submissions - May 2019:

Authors (or principal author)	Link	Info
Steven Baete (NYU)	https://bitbucket.org/sbaete/ismrm2019reprodcgsense	MATLAB, NUFFT (Fessler), gpuNUFFT (Schwarzl, Knoll)
Alexander Fyrdahl (Karolinska Institutet and Karolinska University Hospital, Sweden)	https://github.com/fyrdahl/rrsg_challenge	MATLAB, NUFFT (Fessler)
Kerstin Hammernik (Graz University of Technology)	https://github.com/khammernik/ISMRM2019_RRSG	Python, BART, primal-dual-toolbox, medutils
Seb Harrevelt (Eindhoven University of Technology)	https://github.com/zwep/ismrm19_challenge	Python, PyNUFFT (Lin),
Namgyun Lee (University of Southern California)	https://drive.google.com/file/d/10qD6K-sCEkNjpynRZTpLm8VBUPJFhJCt/view	MATLAB, custom MEX for gridding
Gilad Liberman (MGH)	https://github.com/giladddd/LinopScript	MATLAB, *Demo of linear-operator scripting for BART on challenge datasets*
Michael Loecher (Stanford)	https://github.com/mloecher/rrsg_challenge	Python, custom Cython for gridding
Oliver Maier (Graz University of Technology)	https://github.com/MaierOli2010/ISMRM_RRSG	Python, BART, requires GPU for use of GPyFFT
Franz Patzig, Lars Kasper, Thomas Ulrich, Maria Engel, Johanna Vannesjo, Markus Weiger, David Brunner, Bertram Wilm, Klaas Prüssmann (ETHZ)	https://github.com/mrtm-zurich/rrsg-arbitrary-sense	MATLAB, custom gridding in MATLAB
Ludger Starke (MDC-Berlin)	.../reproducibleResearch19_LudgerStarke.zip	MATLAB, BART
Ye Tian (UCAIR - University of Utah)	https://github.com/YeTianMRI/ISMRM-2019-reproducible	MATLAB, NUFFT (Fessler)
Ke Wang, Miki Lustig, Ekin Karasan, Suma Anand, Volert Roeloffs (Berkeley)	https://github.com/KeWang0622/rrsg_challenge_sigpy	Python, SigPy (Ong)

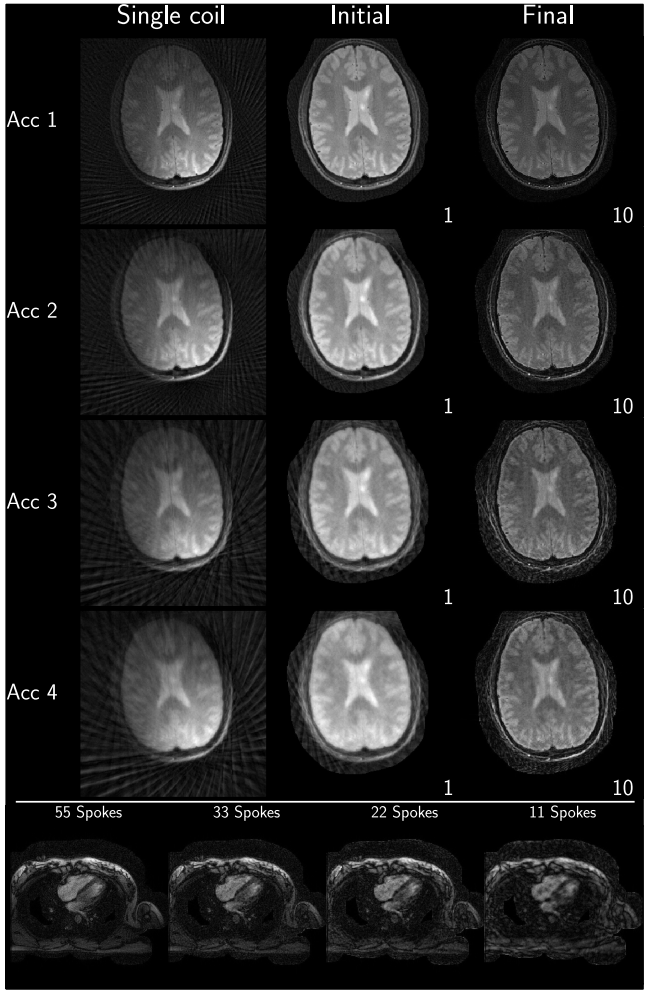

Initial
Final
Initial
Final
Berkeley
Stanford
B.U.F.F
TUG H.
Eindhoven
TUG M.
ETH
USC
KI
Utah
NYU


55
/
11
55
/
11
Berkeley
Stanford
B.U.F.F
TUG H.
Eindhoven
TUG M.
ETH
USC
KI
Utah
NYU

Consolidated Python implementation



Consolidated Matlab implementation



Summary

- Aliasing for non-Cartesian subsampling
- Iterative reconstruction: GD, and CG
- Convergence behavior
- Noise propagation

Received: 10 August 2020 | Revised: 2 October 2020 | Accepted: 2 October 2020

DOI: 10.1002/mrm.28569

REVIEW

Magnetic Resonance in Medicine

CG-SENSE revisited: Results from the first ISMRM reproducibility challenge

Oliver Maier[1] | Steven Hubert Baete[2] | Alexander Fyrdahl[3] | Kerstin Hammernik[4,5] | Seb Harrevelt[6] | Lars Kasper[7,8,9] | Agah Karakuzu[10] | Michael Loecher[11] | Franz Patzig[7] | Ye Tian[12,13] | Ke Wang[14] | Daniel Gallichan[15] | Martin Uecker[16,17,18,19] | Florian Knoll[2]

Challenge submissions and data: https://ismrm.github.io/rrsg/challenge_one/

Reference implementations, evaluation scripts: https://github.com/ISMRM/rrsg_challenge_01

Supplementary spiral brain and radial cardiac data: https://doi.org/10.5281/zenodo.3975887

Exercise

Computational MR imaging
Laboratory 7: Parallel Imaging III: Non-Cartesian Imaging and Iterative Reconstruction

This lab session will be held on January 13th.

Report is due on Wednesday the week after the lab session at 23:59 (January 19th). Send your report by email to Bruno Riemenschneider (bruno.riemenschneider@fau.de) and Florian Knoll (florian.knoll@fau.de).

Learning objectives
The first part of this exercise deals with iterative parallel imaging reconstruction for non-Cartesian sampling patterns, using gradient descent. In the second part, you will explore the conjugate gradient SENSE method.

1. Derivation of gradient descent (analytical):

In gradient descent methods, we need to calculate the gradient of our objective function. In the lecture (slide 21), we said that the gradient is:

$$\frac{\partial}{\partial x}\|Ax-b\|_2^2 = 2A^T(Ax-b)$$

Show that this is indeed true. For this exercise, assume that A, x and b are real valued (the derivation for complex numbers is more involved). To simplify the notation, you can do the derivation with the assumption that A is a 2x2 matrix without loss of generality. You should only need standard linear algebra and vector analysis for this proof. Hints:

i. Remember these expressions: $\|Ax-b\|_2^2 = (Ax-b)^T(Ax-b)$
$$x^TA^Tb = (b^TAx)^T$$

ii. It will be useful during the derivation to use the following substitutions to simplify the notation: $\underbrace{2b^TA}_{c^T}x$ and $x^T\underbrace{A^TA}_{B}x$

2. Iterative image reconstruction with gradient descent

You will find the following items in the data file data_radial_brain_4ch.mat:

kdata (512,64,4): radial k-space data, 64 spokes, 512 readout points, 4 channels
c (256,256,4): receive coil sensitivity maps, 4 channel coil
k (512,64): radial trajectory
w *(512,64):* density compensation
img_senscomb (256,256): Sensitivity combined fully sampled ground truth

2.1. Plot the data and at the sampling trajectory.

2.2. Build a NUFFT operator in the same way as in lab 4 and do a simple gridding reconstruction using density compensation.

2.3. Implement a gradient descent reconstruction as described in the lecture:
  2.3.1. Build the forward and adjoint operators
  2.3.2. Choose a stepsize t=10^{-2}
  2.3.3. Initialize u with a zeros matrix
  2.3.4. Implement the gradient descent update equation
  2.3.5. Run for e.g. 250 iterations

2.4. Plot the fully sampled ground truth, the regridding reconstruction, the gradient descent reconstructed image and the difference image between the reconstructed image and the ground truth.

2.5. Plot the MSE to the ground truth (*img_senscomb)* and the l2 norm of the gradient over the iterations.

3. CG-SENSE

You will find a CG-SENSE implementation in:
cg_sense.m

3.1. Build a NUFFT operator in the same way as in lab 4 and do a simple gridding reconstruction using density compensation.

3.2. Perform a conjugate gradient SENSE reconstruction using the provided code. Compare the convergence behavior of CG to that of gradient descent in exercise You should see convergence in 20-30 iterations.

3.3. Plot the fully sampled ground truth, the regridding reconstruction, the CG reconstructed image and the difference image between the CG image and the ground truth.

3.4. Repeat CG reconstruction with noise instead of the k-space data and plot the result in k-space. Perform at least 500 iterations. What did you just obtain?