Variational Models and Numerical Algorithms for Compressive Sensing in Computerized Tomography [1]

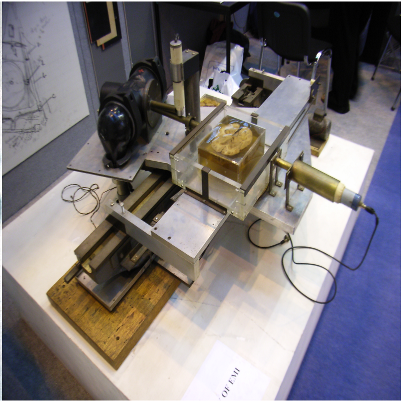
Ming Yan
Department of Mathematics

Supervisor: Professor Luminita Vese

Oral Qualifying Examination
September 9, 2010

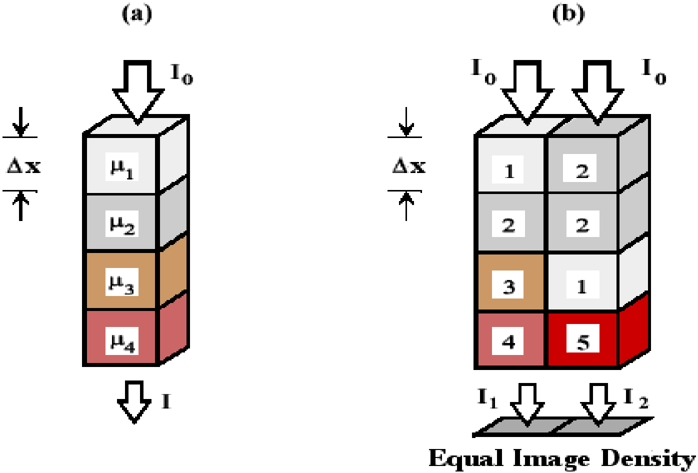
[1] Supported by the Center for Domain Specific Computing (CDSC), UCLA

History (Nobel Prize)



- Wilhelm Conrad Röntgen (1901), Discovery of X-rays.
- Sir Godfrey Newbold Hounsfield and Allan MacLeod Cormack (1979), Computed Tomography.

Attenuation of X-Rays



$I = I_0 \exp\text{-}(\mu_1+\mu_2+\mu_3+\mu_4)\,\Delta x$

$I_1 = I_2 = I_0 \exp(-10\Delta x)$

Attenuation of X-Rays

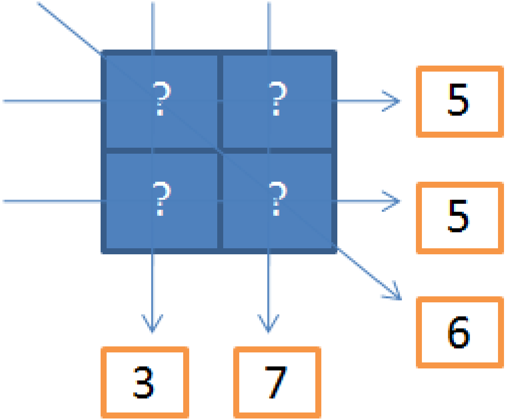
If I_0 is the intensity of the source, $f(x)$ is the linear attenuation coefficient of the object at the point x. L is the ray along which the radiation propagates, and I is the intensity of the radiation at the detector. The mathematical formula is

$$I = I_0 e^{-\int_L f(x)dx}. \quad (1)$$

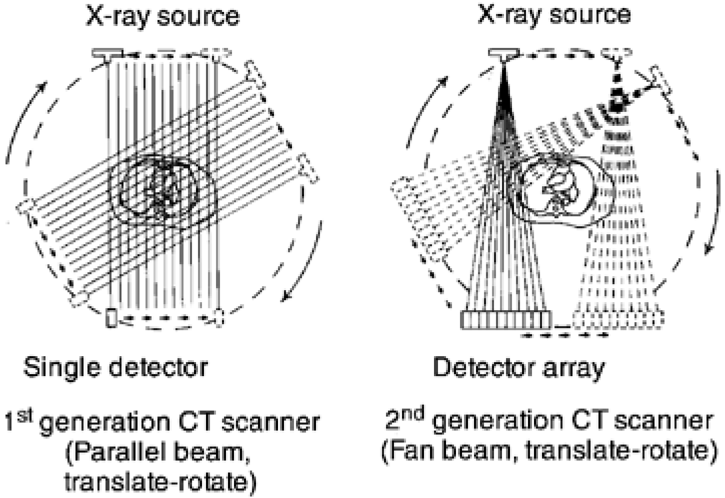
Equivalently

$$\int_L f(x)dx = \log \frac{I_0}{I}. \quad (2)$$

A Toy Example of CT

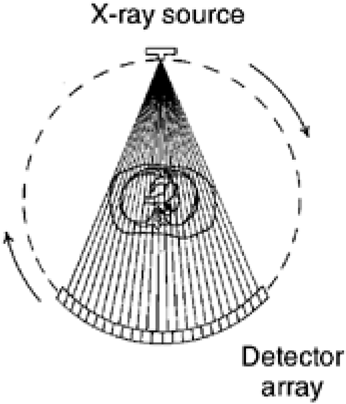


1st & 2nd generations

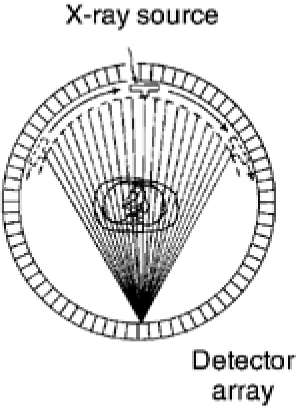


The 1st generation of CT has only one measurement beam, alternately moved in a translational and rotational manner.

3rd & 4th generations



3rd generation CT scanner
(Fan beam, rotate only)



4th generation CT scanner
(Fan beam, stationary circular detector)

The 3rd generation of CT rotates the source and beams. The 4th generation only rotates the source.

other generations

- Electron beam CT (5th generation): eliminate all mechanical motion by employing electron beams controlled electromagnetically.
- Helical CT (6th generation): The source/detector pair rotates continuously through 360^o while the patient is moved at a constant speed.
- Multi-slice CT (7th generation): Using parallel banks of detectors to collect volumetric CT data simultaneously.

Radon Transform (2D)

A straight line can be represented as

$$x \cos\theta + y \sin\theta = t.$$

For each pair (θ, t), we have

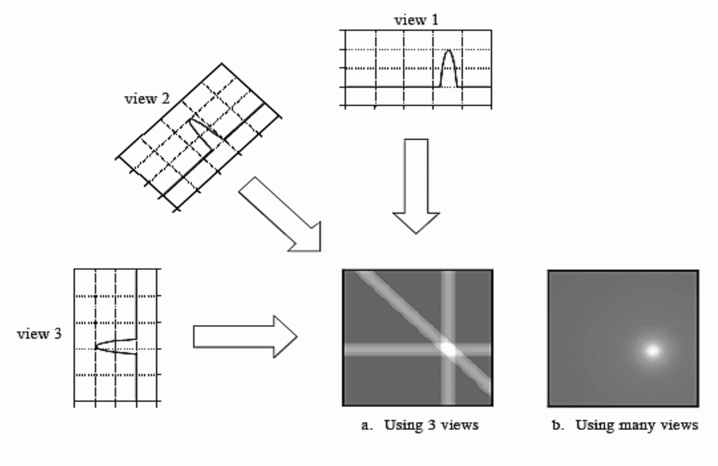
$$g(\theta, t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\delta(x \cos\theta + y \sin\theta - t)dxdy,$$

where δ is the delta function, which is ∞ on the line $x \cos\theta + y \sin\theta - t = 0$, 0 elsewhere.
All the measurements are stored as sinogram data.

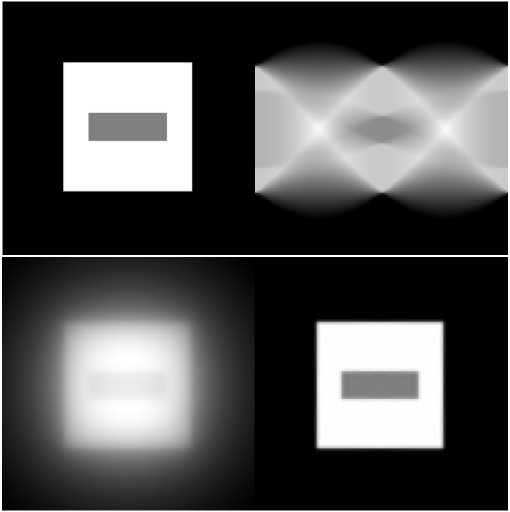
Back Projection

$$f_\theta = g(\theta, x\cos\theta + y\sin\theta)$$

$$f^{BP}(x,y) = \int_0^\pi f_\theta(x,y)\,d\theta$$



Back Projection



Top Left, the original image, Top Right, the sinogram data of the image. Bottom Left, the result of back projection, Bottom Right, the result of filtered back projection.

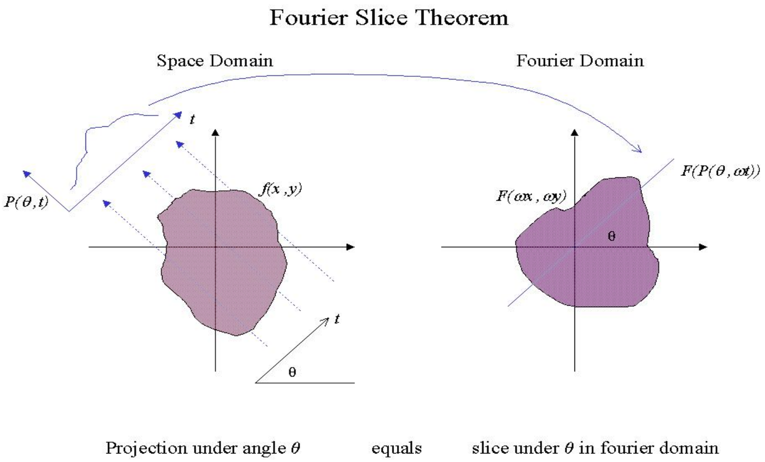
Reconstruction methods

- Filtered Back Projection: The most commonly used algorithm in practice by manufacturer. Fast, need more projections from good geometry.
- Iterative Reconstruction: Insensitive to noise, works when the data is incomplete.
- Compressive Sensing, Total Variation minimization.

Fourier-Slice Theorem

$$
\begin{aligned}
&G(w,\theta)\\
&=\int_{-\infty}^{+\infty} g(\theta,t)e^{-i2\pi wt}dt\\
&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\delta(x\cos\theta+y\sin\theta-t)e^{-i2\pi wt}dxdydt\\
&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)(\int_{-\infty}^{+\infty}\delta(x\cos\theta+y\sin\theta-t)e^{-i2\pi wt}dt)dxdy\\
&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-i2\pi w(x\cos\theta+y\sin\theta)}dxdy\\
&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-i2\pi(ux+vy)}dxdy|_{u=w\cos\theta,v=w\sin\theta}.
\end{aligned}
$$

Fourier Slice Theorem



$$G(w,\theta) = F(u,v)|_{u=w\cos\theta, v=w\sin\theta} = F(w\cos\theta, w\sin\theta). \qquad (3)$$

Consider the inverse Fourier transform of F, we have

$$\begin{aligned}
& f(x,y) \\
&= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{i2\pi(ux+vy)}dudv \\
&= \int_{0}^{2\pi}\int_{0}^{\infty} F(w\cos\theta, w\sin\theta)e^{i2\pi w(x\cos\theta+y\sin\theta)}wdwd\theta \\
&= \int_{0}^{2\pi}\int_{0}^{\infty} G(w,\theta)e^{i2\pi w(x\cos\theta+y\sin\theta)}wdwd\theta \\
&= \int_{0}^{\pi}\int_{-\infty}^{+\infty} |w|G(w,\theta)e^{i2\pi w(x\cos\theta+y\sin\theta)}dwd\theta \\
&= \int_{0}^{\pi}(\int_{-\infty}^{+\infty} |w|G(w,\theta)e^{i2\pi w\rho}dw)|_{\rho=x\cos\theta+y\sin\theta}d\theta.
\end{aligned}$$

Filtered Backprojection

- Take projections and obtain sinogram data (done by machine)
- Transform sinogram data to the frequency domain (FFT)
- Filter data in frequency domain
- Inverse transform, obtain smoothed sinogram
- Backprojection

Algebraic Reconstruction Techniques

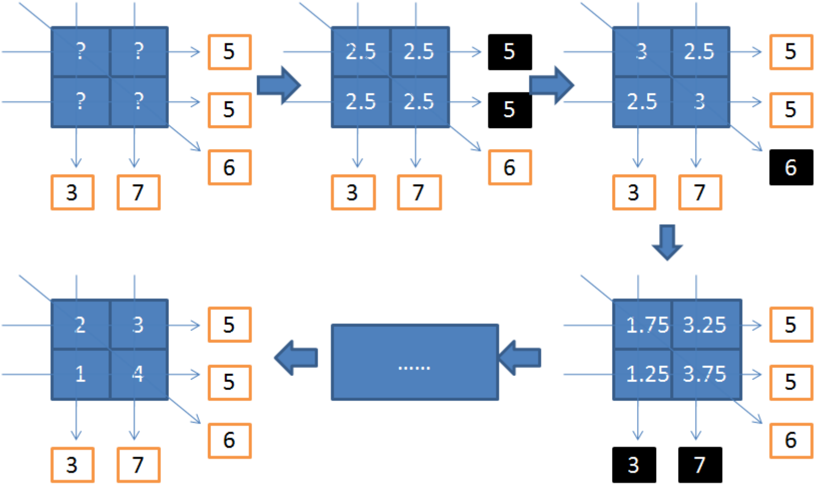
The reconstruction problem is to estimate an image vector x from the following system of equations

$$b = Ax + e,$$

where b is the measurement vector, A is discrete Radon transform, and e is the error in the measurements.

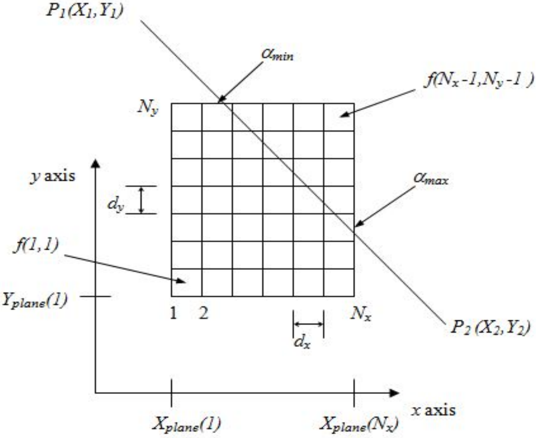
ART method is an iterative procedure, producing a sequence of vectors $x^0, x^1, \cdot$ converging to x^*, approximate solution to the inverse problem.

Toy Example



? ?
? ?
5
5
6
3
7
2.5 2.5
2.5 2.5
5
5
6
3
7
3 2.5
2.5 3
5
5
6
3
7
1.75 3.25
1.25 3.75
5
5
6
3
7
......
2 3
1 4
5
5
6
3
7

Construction of Matrix A: Siddon's algorithm[2]



$$X(\alpha) = X_1 + \alpha(X_2 - X_1),$$
$$Y(\alpha) = Y_1 + \alpha(Y_2 - Y_1).$$

[2]Siddon, 1985

1. calculate range of parametric values $\{\alpha_{min}, \alpha_{max}\}$
2. calculate range of indices $\{i_{min}, i_{max}\}, \{j_{min}, j_{max}\}$
3. calculate parametric sets $\{\alpha_x, \alpha_y\}$
4. merge sets to form set $\{\alpha\}$
5. calculate pixel lengths
6. calculate pixel indices

Expectation Maximization[3]

Expectation Maximization is the method based on maximizing the probability to observe the given results in the coincidence detectors. The probability based on Poisson PDF is as follows:

$$P(b|Ax) = \prod_{i=1}^{M} e^{-a_i x} \frac{(a_i x)^{b_i}}{b_i!}, \tag{4}$$

where a_i is the i^{th} row of A.
Instead of considering the probability directly, we minimize

$$-\log P(b|Ax) = \sum_{i=1}^{M} (a_i x - b_i \log(a_i x)) + const, \tag{5}$$

with a constraint that $x \geq 0$.

[3] Shepp and Vardi, 1982

Expectation Maximization (cont'd)

For this constraint problem, the Karush-Kuhn-Tucker (KKT) condition is

$$\sum_{i=1}^{M}(a_{ji}(1-\frac{b_i}{a_i x}))-y_j=0, \qquad j=1,\cdots,N,$$
$$y\geq 0, \quad x\geq 0, \quad y^T x=0.$$

Thus, $y_j * x_j = 0$ and

$$x_j\sum_{i=1}^{M}(a_{ji}(1-\frac{b_i}{a_i x}))=0, \qquad j=1,\cdots,N.$$

Therefore, we have the following iterative procedure (EM)

$$x_j^{n+1}=\frac{\sum_{i=1}^{M}(a_{ji}(\frac{b_i}{a_i x^n}))}{\sum_{i=1}^{M}a_{ji}}x_j^n. \tag{6}$$

SART[4]

EM is based on Poisson noise, while for weighted Gaussian noise, we have Simultaneous Algebraic Reconstruction Technique (SART). Define

$$w_i = A_{i,+} = \sum_{j=1}^{M} a_{i,j},$$

$$v_j = A_{+,j} = \sum_{i=1}^{N} a_{i,j}.$$

In addition V and W are diagonal matrices with diagonal elements v_j and w_i.

[4] Andersen and Kak, 1984

SART (cont'd)

Then the probability based on Gaussian PDF is

$$P(b|Ax) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi w_i}} e^{-\frac{(b_i - a_i x)^2}{2w_i}},$$

where a_i is the i^{th} row of A.
Similarly, instead of considering the probability, we minimize

$$-\log P(b|Ax) = \sum_{i=1}^{M} w_i^{-1}(a_i x - b_i)^2 + const.$$

Using steepest descent method, we have the following updating

$$x^{n+1} = x^n - wV^{-1}A^T W^{-1}(Ax^n - b). \tag{7}$$

Other Approaches for SART

The sequence of SART will converge to the solution of $Ax = b$ with the least $\sum_{j=1}^{N} v_j x_j^2$, if b is in the range of A. The optimization problem is

$$\begin{cases} \underset{x}{\text{minimize}} & \sum_{j=1}^{N} v_j x_j^2, \\ \text{subject to} & Ax = b. \end{cases} \tag{8}$$

SART can be derived from linearized Bregman iteration and gradient descent of the dual problem[5].

[5] M. Yan, CAM report 10-27

Some Notations

The ellipsoidal norm of $x \in \mathbf{R}^n$ is defined as follows:

$$\|x\|_G^2 = \langle x, x \rangle_G = \langle x, Gx \rangle = x^T Gx,$$

where G is an $n \times n$ symmetric positive definite matrix.

We will use $\|x\|_V$ and $\|y\|_{W^{-1}}$ in the following analysis.

SART as Linearized Bregman Iteration

The goal is to solve this constraint problem

$$\begin{cases} \underset{x}{\text{minimize}} & \|x\|_V^2, \\ \text{subject to} & Ax = b. \end{cases} \tag{9}$$

Define $J(x) = \mu\|x\|_V^2$, and we have the Bregman distance

$$D_J^p(x^1, x^2) = J(x^1) - J(x^2) - \langle p, x^1 - x^2 \rangle, \tag{10}$$

with $p \in \partial J(x^2)$, p is some subgradient of J at x^2. In general, the Bregman distance is not a distance in the common sense. However, for this special J, the Bregman distance is a distance, and we have

$$D_J^p(x^1, x^2) = J(x^1 - x^2) = \mu\|x^1 - x^2\|_V^2.$$

SART as Linearized Bregman Iteration (cont'd)

The Bregman iteration is as follows

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \ \mu \|x - x^k\|_V^2 + \frac{1}{2}\|Ax - b\|_{W^{-1}}^2,$$

for $k = 0, 1, \cdots$, starting with initial guess $x^0 = 0$.
We have the iteration

$$(2\mu V + A^T W^{-1} A)x^{k+1} = A^T W^{-1} b + 2\mu V x^k.$$

The linearized Bregman iteration is as follows,

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \ \mu \|x - x^k\|_V^2 + \langle Ax^k - b, Ax \rangle_{W^{-1}} + \frac{1}{2\alpha}\|x - x^k\|_V^2,$$

The iteration is

$$x^{k+1} = x^k + \frac{1}{2\mu + \frac{1}{\alpha}} V^{-1} A^T W^{-1}(b - Ax^k) \qquad (SART).$$

Convergence Analysis

Lemma 1: $V - A^T W^{-1} A$ and $W - AV^{-1}A^T$ are positive semidefnite matrices.
Theorem 2: The W^{-1}-norm of the residual $b - Ax^k$ is decreasing if $0 < w < 2$. Furthermore, for $k = 0, 1, \cdots$, we have

$$\|Ax^{k+1} - b\|_{W^{-1}}^2 + (\frac{2}{w} - 1)\|x^{k+1} - x^k\|_V^2 \leq \|Ax^k - b\|_{W^{-1}}^2.$$

The W^{-1}-norm of the residual will decrease until x^k remains unchanged. In addition, we can show that x^k will converge to $\bar{x}$.
Theorem 3: Assume that x^* is the solution of the constraint problem, we have

$$\mu\|\bar{x}\|_V^2 \leq \mu\|x^*\|_V^2 + \frac{1}{\alpha}\langle \bar{x}, x^* - \bar{x}\rangle_V.$$

Thus $\bar{x} = x^*$.

SART as Dual Gradient Descent

The constraint problem can be rewritten as

$$\begin{cases} \underset{x}{\text{minimize}} & \frac{1}{2w}\|x\|_V^2, \\ \text{subject to} & W^{-\frac{1}{2}}Ax = W^{-\frac{1}{2}}b. \end{cases} \tag{11}$$

Lagrangian function is

$$L(x,y) = \frac{1}{w}\|x\|_V^2 + y^T W^{-\frac{1}{2}}(Ax - b).$$

For fixed y, we can find the minimizer with respect to x only,

$$x = -wV^{-1}A^T W^{-\frac{1}{2}}y.$$

Plugging this into $L(x,y)$ leaves a function of y

$$L(y) = -y^T W^{-\frac{1}{2}}b - \frac{w}{2}\|V^{-1}A^T W^{-\frac{1}{2}}y\|_V^2.$$

SART as Dual Gradient Descent (cont'd)

Dual problem is

$$\underset{y}{\text{minimize}}\, F(y) = y^T W^{-\frac{1}{2}} b + \frac{w}{2}\|V^{-1}A^T W^{-\frac{1}{2}} y\|_V^2.$$

Using gradient descent method to solve it,

$$\nabla F(y) = W^{-\frac{1}{2}} b + wW^{-\frac{1}{2}} AV^{-1}A^T W^{-\frac{1}{2}} y.$$

and

$$y^{k+1} - y^k = -\nabla F(y^k) = -W^{-\frac{1}{2}} b - wW^{-\frac{1}{2}} AV^{-1}A^T W^{-\frac{1}{2}} y^k.$$

Multiply by $-wV^{-1}A^T W^{-\frac{1}{2}}$, and we have

$$x^{k+1} - x^k = wV^{-1}A^T W^{-1} b - wV^{-1}A^T W^{-1} Ax^k \qquad (SART).$$

Other Algorithms with Different V and W

Cimmino's Algorithm[6]**:** $V = I$ and W is the diagonal matrix with diagonal elements $M\|A_{i,\cdot}\|^2$.
Component Averaging[7]**:** $V = I$ and W is the diagonal matrix with diagonal elements $\sum_{j=1}^{N} s_j A_{i,j}^2$, with s_j being the number of nonzero elements in the j^{th} column of matrix A.

[6] Cimmino 1938
[7] Censor, et al. 2001

Comparing of these three methods

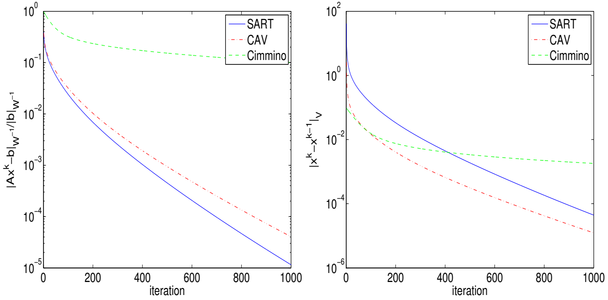


Figure: Decay of residual and difference between 2 iterations in corresponding ellipsoidal norms.

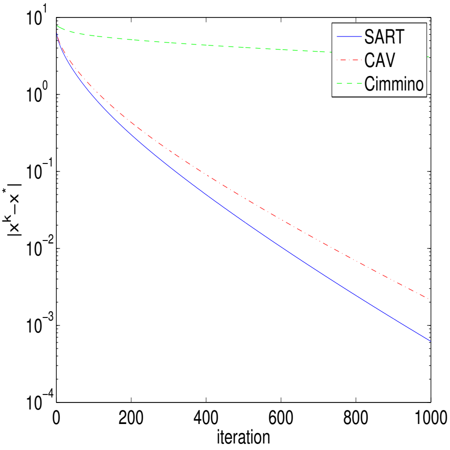


Figure: Error in $l2$-norm for three methods

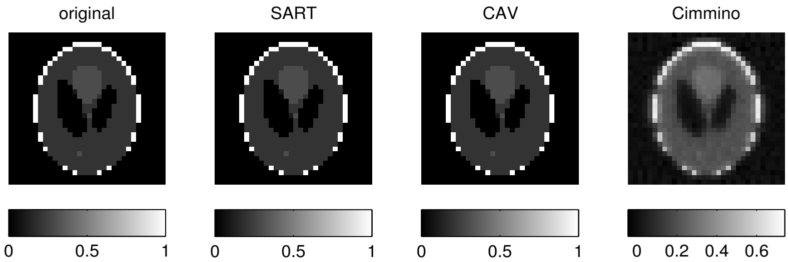


Figure: Image reconstruction results by three methods

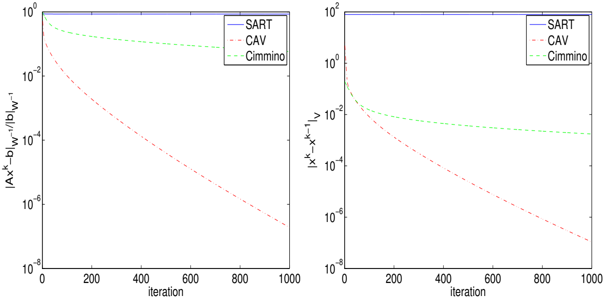


Figure: Decay of residual and difference between 2 iterations in ellipsoidal norms

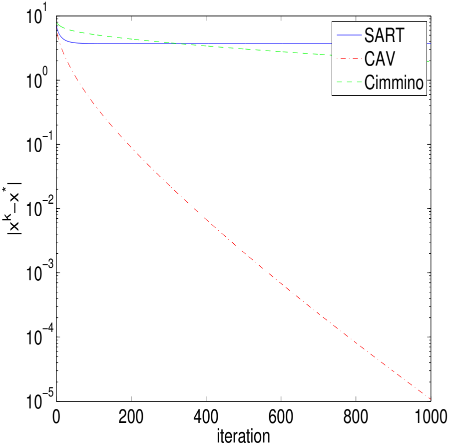


Figure: Error in $l2$-norm for three methods

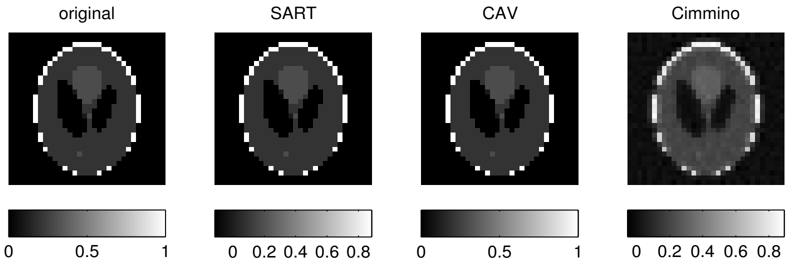


Figure: Reconstruction results of a 32x32 image with $w = 2$.

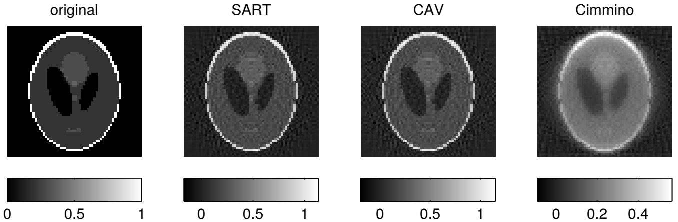


Figure: Reconstruction of a 64x64 image with less measurements by these three methods.

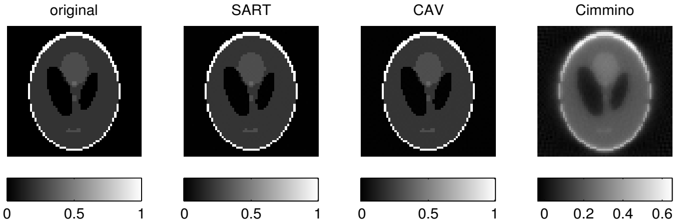


Figure: Reconstruction of a 64x64 image with more measurements by these three methods.

SART and CG

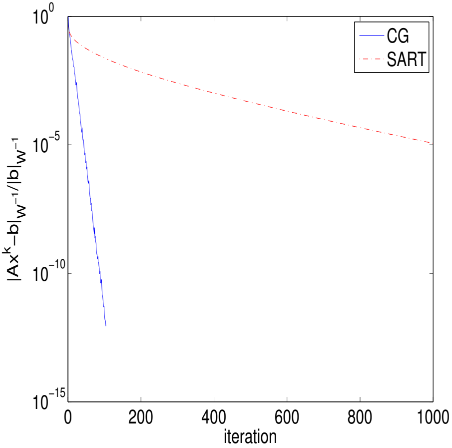


Figure: Decay of residual of SART and CG in W^{-1}-norm

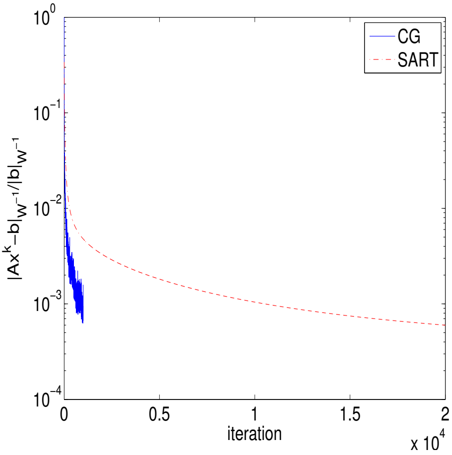


Figure: Decay of residual of SART and CG in W^{-1}-norm

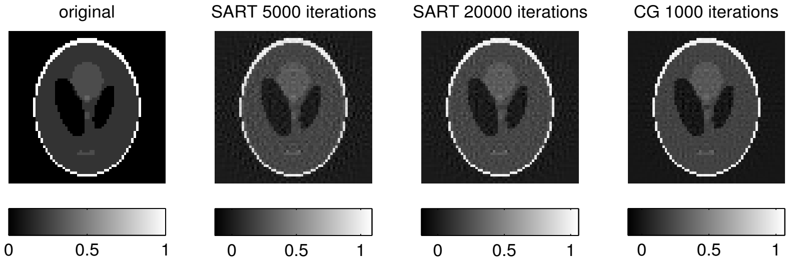


Figure: Reconstruction of a 64x64 image with less measurements by SART and CG.

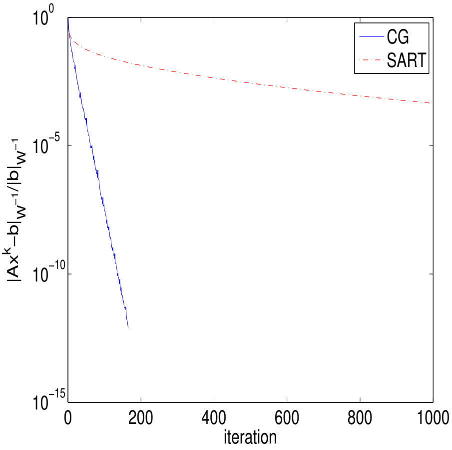


Figure: Decay of residual of SART and CG in W^{-1}-norm

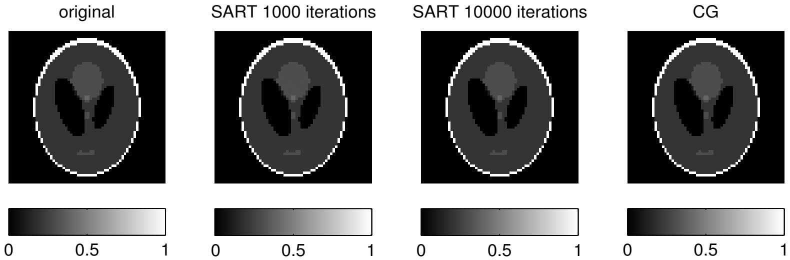


Figure: Reconstruction of a 64x64 image with more measurements by SART and CG

EM+TV

Combining EM and total variation (TV), we have the optimization problem,

$$\underset{x}{\text{minimize}} \int |\nabla x| - \alpha \log P$$
$$\text{subject to } x \geq 0,$$

with $P = \prod_{i=1}^{M} e^{-a_i x} \frac{(a_i x)^{b_i}}{b_i!}$. Or

$$\underset{x}{\text{minimize}} \int |\nabla x| + \alpha \sum_{i=1}^{M} (a_i x - b_i \log(a_i x))$$
$$\text{subject to } x \geq 0. \tag{12}$$

This is a convex problem, and we can use many algorithms for convex problems to solve this.

EM+TV (cont'd)

The KKT condition of this is

$$-\mathrm{div}(\frac{\nabla x}{|\nabla x|})_j + \alpha \sum_{i=1}^{M}(a_{ji}(1-\frac{b_i}{a_i x})) - y_j = 0 \qquad j = 1, \cdots, N$$

$$y \geq 0, \quad x \geq 0, \quad y^T x = 0.$$

It becomes

$$-\frac{x_j}{\sum_{i=1}^{M} a_{ji}}\mathrm{div}(\frac{\nabla x}{|\nabla x|})_j + \alpha \frac{\sum_{i=1}^{M}(a_{ji}(1-\frac{b_i}{a_i x}))}{\sum_{i=1}^{M} a_{ji}} x_j = 0 \qquad j = 1, \cdots, N.$$

or

$$-\frac{x_j}{\sum_{i=1}^{M} a_{ji}}\mathrm{div}(\frac{\nabla x}{|\nabla x|})_j + \alpha x_j - \alpha \frac{\sum_{i=1}^{M}(a_{ji}(\frac{b_i}{a_i x}))}{\sum_{i=1}^{M} a_{ji}} x_j = 0 \qquad j = 1, \cdots, N.$$

EM+TV (cont'd)

Denote

$$x_j^{EM} = \frac{\sum_{i=1}^{M}(a_{ji}(\frac{b_i}{a_i x}))}{a_j^T \overrightarrow{1}} x_j \tag{13}$$

and this is the EM update.
The KKT condition becomes

$$-\frac{x_j}{a_j^T \overrightarrow{1}} \mathrm{div}(\frac{\nabla x}{|\nabla x|})_j + \alpha x_j - \alpha x_j^{EM} = 0 \qquad j = 1, \cdots, N, \tag{14}$$

and this is the optimality for the following TV minimization problem

$$\underset{x}{\text{minimize}} \int |\nabla x| + \alpha \sum_{j=1}^{N} (a_j^T \overrightarrow{1})(x_j - x_j^{EM} \log x_j).$$

We can solve it by iterations. For this step, we consider this to be 2D or 3D image. For 2D, we have the new notations $u_{i,j}$.

EM+TV (cont'd)

This is a nonlinear equation, so we consider a linearized version

$$
\begin{aligned}
&-\frac{u_{i,j}^n}{V_{i,j}}\frac{u_{i+1,j}^n-u_{i,j}^{n+1}}{\sqrt{\epsilon+(u_{i+1,j}^n-u_{i,j}^n)^2+(u_{i,j+1}^n-u_{i,j}^n)^2}}\\
&+\frac{u_{i,j}^n}{V_{i,j}}\frac{u_{i,j}^{n+1}-u_{i-1,j}^n}{\sqrt{\epsilon+(u_{i,j}^n-u_{i-1,j}^n)^2+(u_{i-1,j+1}^n-u_{i-1,j}^n)^2}}\\
&-\frac{u_{i,j}^n}{V_{i,j}}\frac{u_{i,j+1}^n-u_{i,j}^{n+1}}{\sqrt{\epsilon+(u_{i+1,j}^n-u_{i,j}^n)^2+(u_{i,j+1}^n-u_{i,j}^n)^2}}\\
&+\frac{u_{i,j}^n}{V_{i,j}}\frac{u_{i,j}^{n+1}-u_{i,j-1}^n}{\sqrt{\epsilon+(u_{i+1,j-1}^n-u_{i,j-1}^n)^2+(u_{i,j}^n-u_{i,j-1}^n)^2}}+\alpha u_{i,j}^{n+1}-\alpha u_{i,j}^{EM}=0
\end{aligned}
$$

Figure: test image

To construct the sinogram data, we use the fan beam geometry, and for each view, we choose 301 measurements. We will compare the result of EMTV and the result of FBP.

Sinogram Data

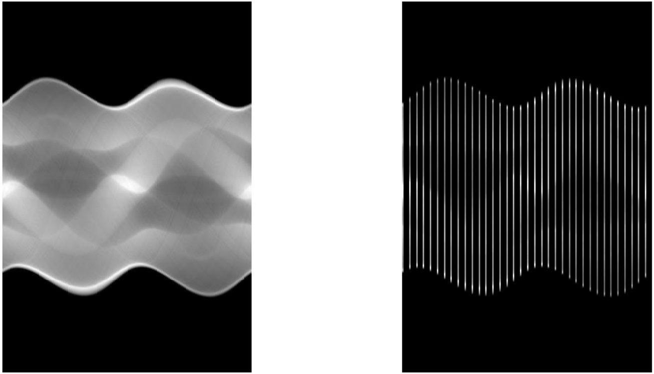
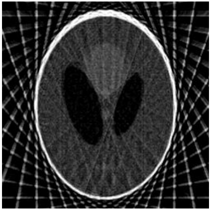


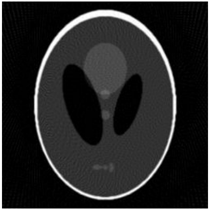
Figure: The sinogram data for 180 views and 36 views, to compare these, for unknown columns, we use 0 instead.

2D Reconstruction Results(Noise Free)

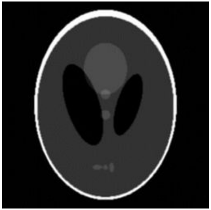
FBP 36 views(RMSE = 50.8394)



FBP 180 views(RMSE = 14.1995)



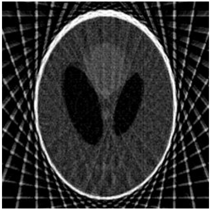
FBP 360 views(RMSE = 12.6068)



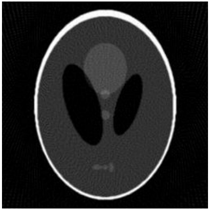
EMTV 36 views(RMSE = 2.3789)

2D Reconstruction Results(Noise)

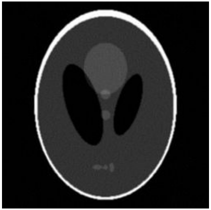
FBP 36 views(RMSE = 51.1003)



FBP 180 views(RMSE = 14.3698)



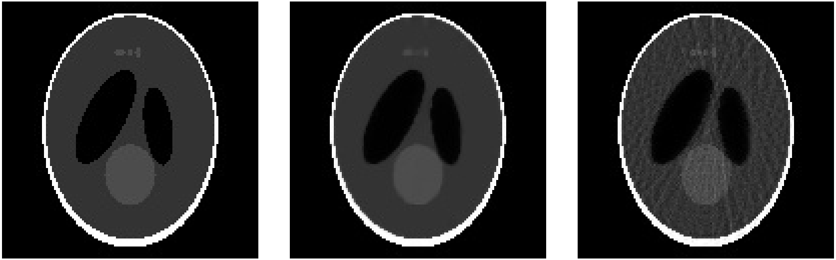
FBP 360 views(RMSE = 12.7039)



EMTV 36 views(RMSE = 3.0868)

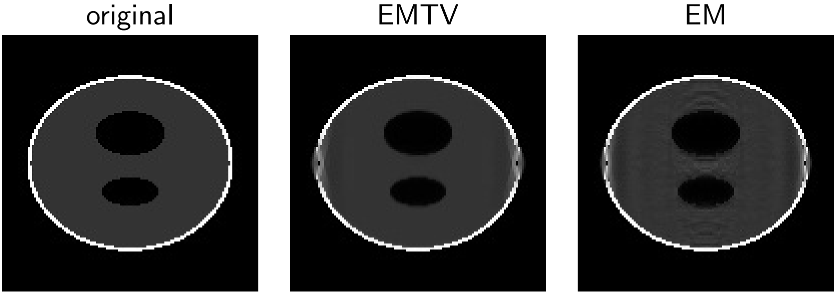
3D Reconstruction Results

original EMTV EM



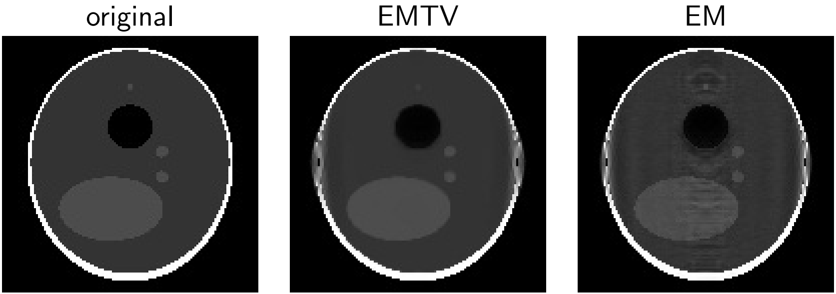
The middle slice in z-direction.

3D Reconstruction Results



The middle slice in y-direction.

3D Reconstruction Results



The middle slice in x-direction.

Future Work

- parallel computing
- modify the fidelity term, assumption of noise on I instead of b
- other regularization methods (nonlocal TV, Cartoon + Texture, Frames)
- image reconstruction for MRI (compressive sensing + Rician denoising)
- apply this algorithm to real helical cone-beam data

Thank You!

Comparing of Different Methods for Poisson Noise Removal

We have to find a minimizer of [8]

$$E(u) := \int_\Omega |\nabla u| + \alpha \sum_{i,j} \left(u_{i,j} - f_{i,j} \log u_{i,j} \right).$$

The Euler-Lagrange equation for minimizing it is

$$0 = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|} \right) + \alpha \frac{1}{u}(f - u).$$

[8] Le et al. 2007

Gradient and Sobolev Gradient Descent

Gradient Descent

$$u_t = -\nabla_{L^2} E(u) = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \alpha \frac{1}{u}(f - u).$$

Sobolev Gradient Descent

$$u_t = -(1 - \triangle)^{-1} \nabla_{L^2} E(u) = (1 - \triangle)^{-1} (\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \alpha \frac{1}{u}(f - u)).$$

Semi-Implicit

Similarly, from the Euler-Lagrange equations

$$0 = u_{i,j} \mathrm{div} \left(\frac{\nabla u}{|\nabla u|} \right)_{i,j} + \alpha (f_{i,j} - u_{i,j}),$$

we have

$$\begin{aligned} &a^n_{i,j}(u^n_{i+1,j} - u^{n+1}_{i,j}) + b^n_{i,j}(u^n_{i,j+1} - u^{n+1}_{i,j}) \\ +&c^n_{i,j}(u^n_{i-1,j} - u^{n+1}_{i,j}) + d^n_{i,j}(u^n_{i,j-1} - u^{n+1}_{i,j}) + \alpha f_{i,j} - \alpha u^{n+1}_{i,j} = 0 \end{aligned}$$

Chambolle's Projection

The original problem is

$$\min_u \int_\Omega |\nabla u| + \alpha \sum_{i,j} \left(u_{i,j} - f_{i,j} \log u_{i,j} \right).$$

It is equivalent to

$$\min_u \max_{p \in Y} \alpha \sum_{i,j} (u_{i,j} - f_{i,j} \log u_{i,j}) + \langle p, \nabla u \rangle,$$

where, $Y = \{p : |p_{i,j}| \leq 1\}$. Change the order of min and max,

$$\max_{p \in Y} \min_u \alpha \sum_{i,j} (u_{i,j} - f_{i,j} \log u_{i,j}) + \langle p, \nabla u \rangle,$$

Chambolle's Projection (cont'd)

For fixed p, we can easily find the optimal u with

$$u^*_{i,j} = \frac{f_{i,j}}{1 - \frac{1}{\alpha}(\text{div } p)_{i,j}}.$$

Plug this back into the problem and we have

$$\min_{p \in Y} \sum_{i,j} -f_{i,j} \log\left(1 - \frac{1}{\alpha}(\text{div } p)_{i,j}\right),$$

The Karush-Kuhn-Tucker condition of this problem is

$$-\nabla\left(\frac{f_{i,j}}{1 - \frac{1}{\alpha}(\text{div } p)_{i,j}}\right) + \beta_{i,j} p_{i,j} = 0$$

with $\beta_{i,j} \geq 0$ being the Lagrange multiplier associated with $|p_{i,j}| \leq 1$.

Chambolle's Projection

If $\beta_{i,j} > 0$, then $|p_{i,j}| = 1$, and $\beta_{i,j} = \left|\nabla\left(\frac{f_{i,j}}{1-\frac{1}{\alpha}(\text{div } p)_{i,j}}\right)\right|$.

Otherwise, $\beta = 0$, and we also have $\beta_{i,j} = \left|\nabla\left(\frac{f_{i,j}}{1-\frac{1}{\alpha}(\text{div } p)_{i,j}}\right)\right|$.

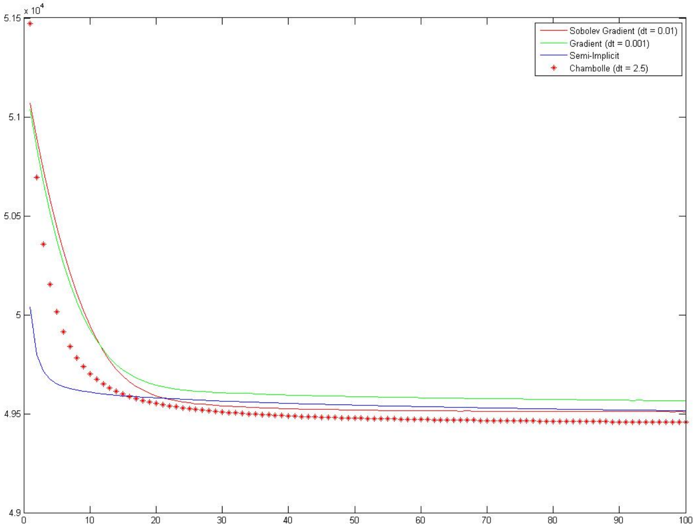
Therefore,

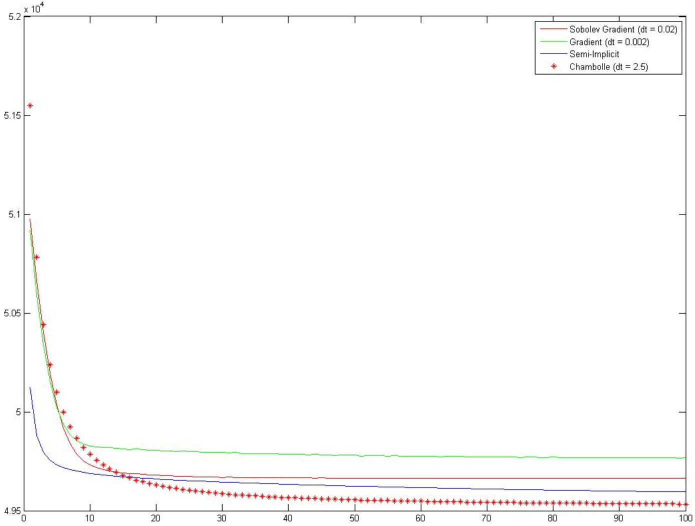
$$\beta_{i,j} = \left|\nabla\left(\frac{f_{i,j}}{1-\frac{1}{\alpha}(\text{div } p)_{i,j}}\right)\right|$$

is satisfied for all case.

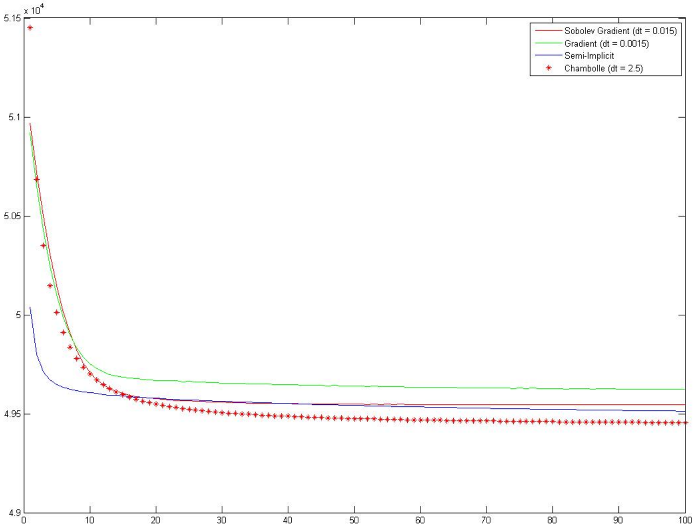
Then, we have the following semi-implicit algorithm

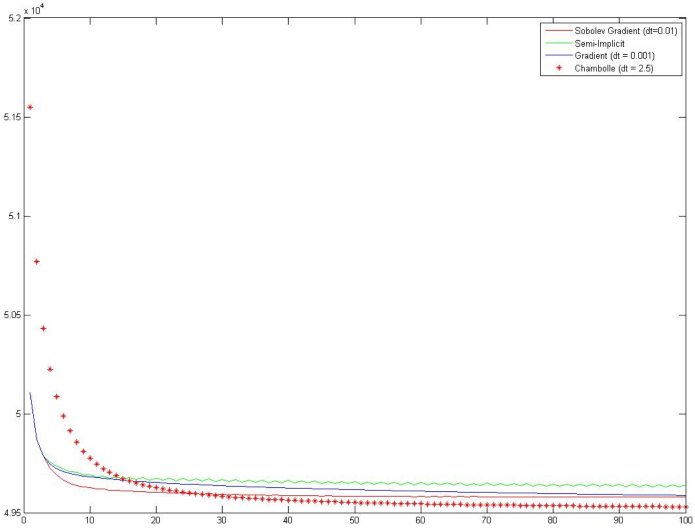
$$p_{i,j}^{n+1} = p_{i,j}^{n} + \tau_{i,j}\left(\nabla\left(\frac{f_{i,j}}{1-\frac{1}{\alpha}(\text{div } p^n)_{i,j}}\right) - \left|\nabla\left(\frac{f_{i,j}}{1-\frac{1}{\alpha}(\text{div } p^n)_{i,j}}\right)\right| p_{i,j}^{n+1}\right.$$

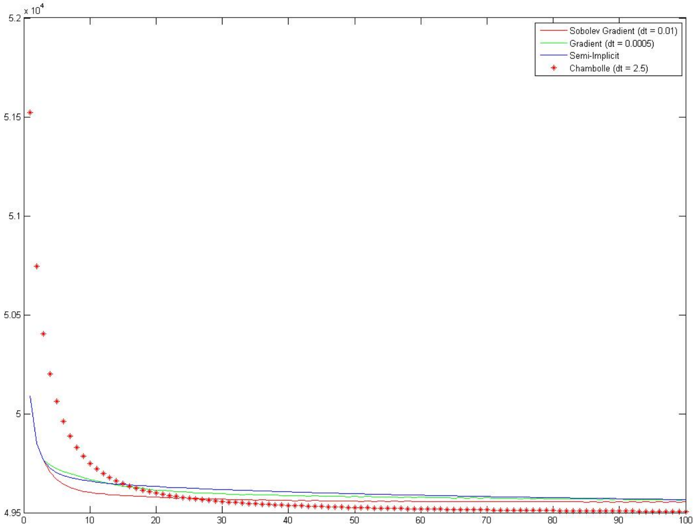

x 10^4
5.15
5.1
5.05
5
4.95
4.9
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt = 0.01)
Gradient (dt = 0.001)
Semi-Implicit
Chambolle (dt = 2.5)

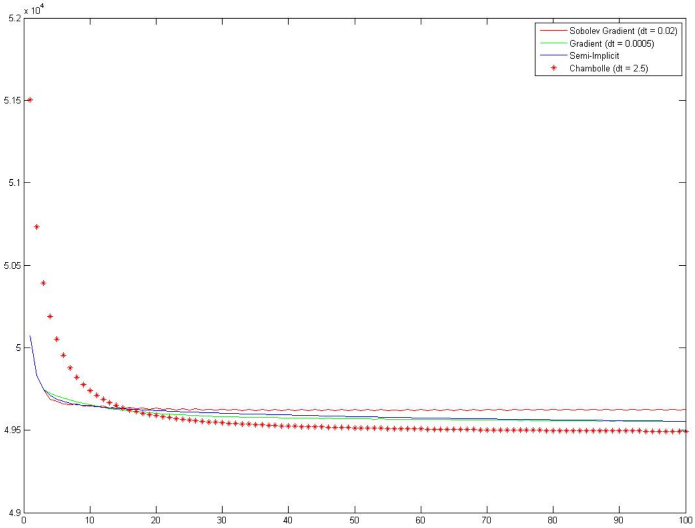


x 10^4
5.2
5.15
5.1
5.05
5
4.95
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt = 0.02)
Gradient (dt = 0.002)
Semi-Implicit
Chambolle (dt = 2.5)

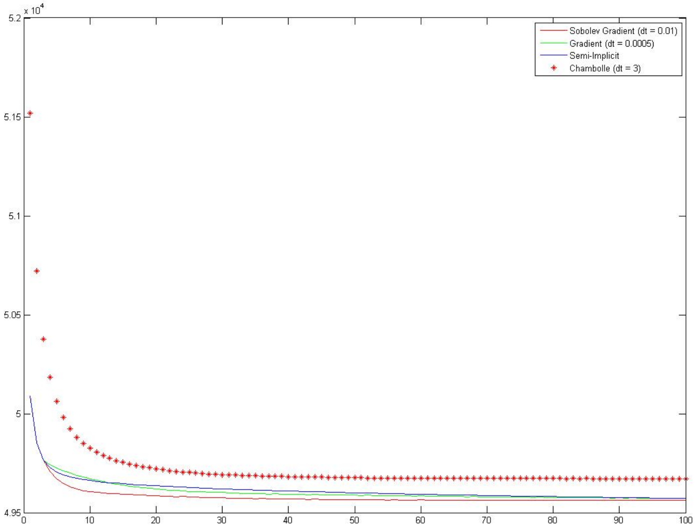

x 10^4
5.15
5.1
5.05
5
4.95
4.9
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt = 0.015)
Gradient (dt = 0.0015)
Semi-Implicit
Chambolle (dt = 2.5)

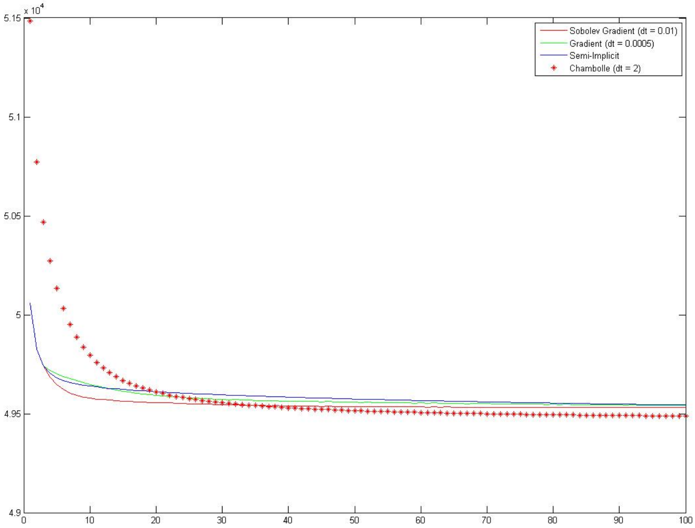

x 10^4
5.2
5.15
5.1
5.05
5
4.95
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt=0.01)
Semi-Implicit
Gradient (dt = 0.001)
Chambolle (dt = 2.5)


x 10^4
5.2
5.15
5.1
5.05
5
4.95
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt = 0.01)
Gradient (dt = 0.0005)
Semi-Implicit
Chambolle (dt = 2.5)



x 10^4
5.2
5.15
5.1
5.05
5
4.95
4.9
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt = 0.02)
Gradient (dt = 0.0005)
Semi-Implicit
Chambolle (dt = 2.5)


x 10^4
5.2
5.15
5.1
5.05
5
4.95
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt = 0.01)
Gradient (dt = 0.0005)
Semi-Implicit
Chambolle (dt = 3)


x 10^4
5.15
5.1
5.05
5
4.95
4.9
0
10
20
30
40
50
60
70
80
90
100
Sobolev Gradient (dt = 0.01)
Gradient (dt = 0.0005)
Semi-Implicit
Chambolle (dt = 2)