Large-Scale Optimization Algorithms
for Inverse Problems
in Atmospheric Imaging

Curt Vogel
Department of Mathematical Sciences
Montana State University
Bozeman, MT 59717-2400 USA
vogel@math.montana.edu
http://www.math.montana.edu/∼vogel/

Collaborators

Luc Gilles and John Bardsley (MSU)
Bob Plemmons and Todd Torgersen (WFU)
Dave Tyler, et al (MHPCC)

Outline

- Mathematical Model of Atmospheric Optical Image Formation
- Regularization and the Associated Optimization Problem
- Large Scale Optimization Techniques
  - Limited Memory BFGS (L-BFGS)
  - Newton/CG/Trust Region algorithm due to Steihaug
- Implementation Issues
  - Initial Hessian for L-BFGS
  - Preconditioner for CG
- Comparison of Methods

Physics of Atmospheric Image Formation

As light from a distant space object enters the atmosphere, the light rays are bent because of variations in the index of refraction associated with temperature changes. This causes blurring of images. Refractive index variations are time-dependent due to atmospheric turbulence.

Model Assumptions

- The light source is very far from the observer. Light rays are nearly parallel as they enter the atmosphere.
- Image degradation is caused by variations in the index of refraction within the atmosphere.
- Variations in the index of refraction are relatively small.
- Variations in index of refraction occur in a thin layer which is close to the observer.
- Light from the source is incoherent.

Forward Model

$$d_{ij} = (s \star f)(x_i, y_j) + \eta_{ij}, \quad 1 \le i \le n_x, 1 \le j \le n_y.$$

- d_{ij} denotes measured intensity at the ij^{th} pixel. The $n_x \times n_y$ array $\mathbf{d}$ constitutes the discrete image.
- s denotes the point spread function, or PSF.
- f denotes the object. This is the desired "true image".
- $\star$ denotes 2-D convolution product,

$$(s \star f)(x, y) = \int\int s(x - x', y - y')\, f(x', y')\, dx'\, dy'.$$

- η_{ij} represents noise in the data.

The PSF

$$s[\phi] = |\mathcal{F}^{-1}\{pe^{\imath\phi}\}|^2,$$

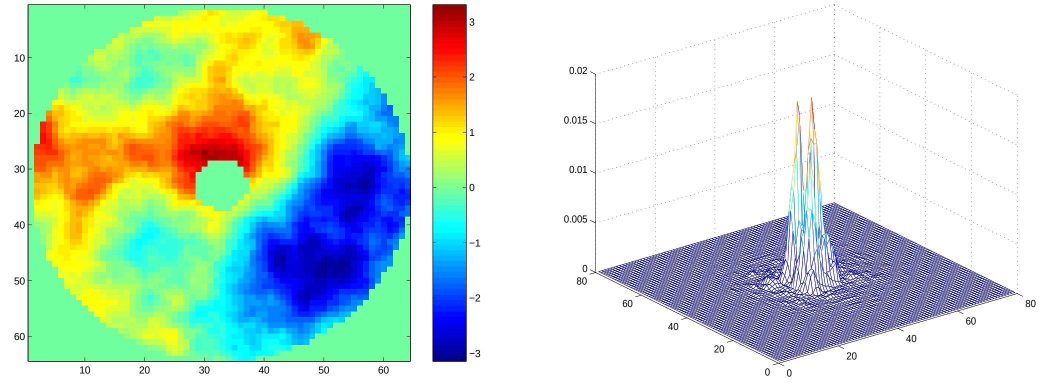
where

- $\mathcal{F}$ denotes 2-D Fourier transform
- $\imath = \sqrt{-1}$
- p denotes the pupil function, or aperture function,

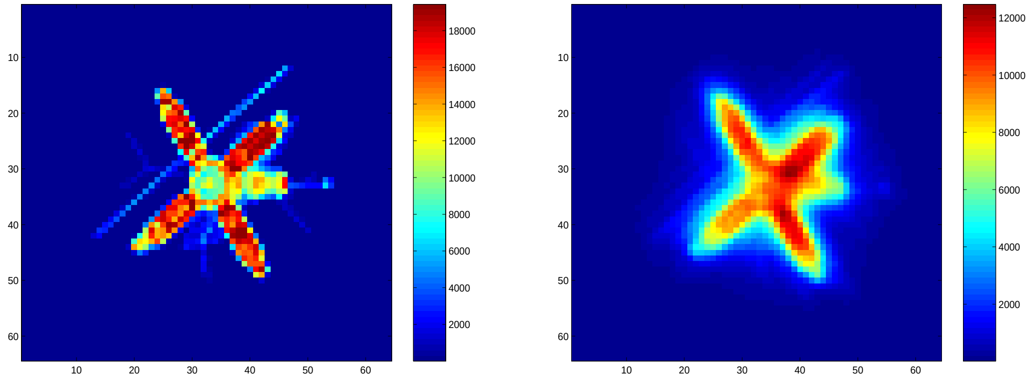
$$p(x, y) = \begin{cases} 1, & (x, y) \in \Omega_p, \\ 0, & \text{otherwise.} \end{cases}$$

- ϕ denotes the phase, or wavefront aberration. This characterizes the medium through which light travels.

Simulated Data



Phase ϕ and PSF $s[\phi] = |\mathcal{F}^{-1}\{pe^{\imath\phi}\}|^2$.



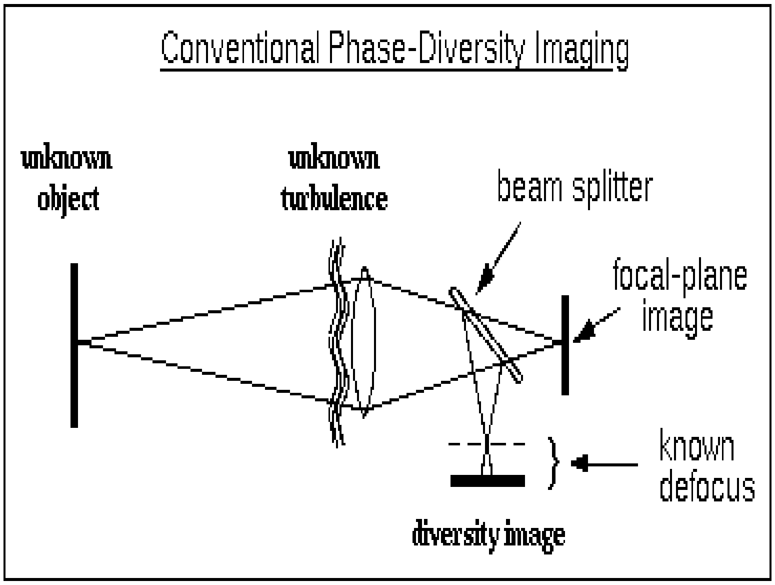
Object f and image $d = s[\phi] \star f + \eta$.

Inverse Problem

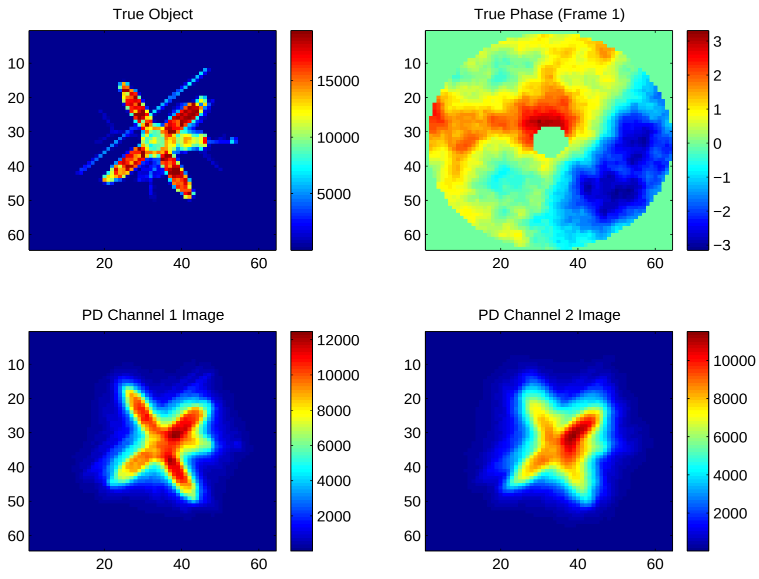
Estimate object f from data $\mathbf{d} = s \star f + \eta$. Difficulty: PSF s is also unknown. Need additional information.

Phase Diversity. Recall $s = s[\phi] = |\mathcal{F}^{-1}\{p \exp(\imath\phi)\}|^2$. Add known phase perturbation to get a second image,

$$\begin{aligned}
\mathbf{d} &= s[\phi] \star f + \eta \\
\mathbf{d}' &= s[\phi + \theta] \star f + \eta'
\end{aligned}$$



Conventional Phase-Diversity Imaging



True Object

True Phase (Frame 1)

PD Channel 1 Image

PD Channel 2 Image

Regularization

Solution to object, phase estimation problem is unstable with respect to noise in the data.

Regularization. To restore stability, apply Tikhonov regularization, or penalized output least squares. Mimimize

$$\begin{aligned} J[\phi, f] &= ||s[\phi] \star f - \mathbf{d}||^2 + ||s[\phi + \theta] \star f - \mathbf{d}'||^2 \\ &\quad + \gamma J_{object}^{reg}[f] + \alpha J_{phase}^{reg}[\phi], \end{aligned}$$

where γ, α are small positive parameters, and J_{object}^{reg}, J_{phase}^{reg} are regularization functions.

Extensions. Much better reconstructions can be obtained by taking more (time-dependent) frames. One can also take more phase diversity channels.

$$\begin{aligned} J[\phi_1, \ldots, \phi_T, f] &= \sum_{k=1}^{K} \sum_{t=1}^{T} ||s[\phi_t + \theta_k] \star f - \mathbf{d}_{k,t}||^2 \\ &\quad + \gamma J_{object}^{reg}[f] + \alpha \sum_{t=1}^{T} J_{phase}^{reg}[\phi_t]. \end{aligned}$$

For object regularization use the "minimum information prior",

$$J_{object}^{reg}[f] = ||f||^2.$$

Phase Modeling and Regularization

Model phase $\phi(\mathbf{x})$ as a realization of a wide-sense stationary stochastic process with zero mean and translation-invariant covariance,

$$\begin{aligned}\mathcal{E}(\phi(\mathbf{x})\phi(\mathbf{y})) &\overset{\text{def}}{=} c_\phi(\mathbf{x},\mathbf{y}) \\ &= c_\phi(\mathbf{x}-\mathbf{y}).\end{aligned}$$

The von Karman model for phase spectrum is

$$\mathcal{F}\{c_\phi\}(\omega) = \frac{a}{(b+|\omega|^2)^{11/6}}.$$

Construct phase regularization operator,

$$J_{phase}^{reg}[\phi] = \left\langle \frac{\mathcal{F}\{\phi\}}{\mathcal{F}\{c_\phi\}}, \mathcal{F}\{\phi\} \right\rangle.$$

This penalizes high spatial-frequency components in reconstructed phase.

Gradient Computations

The least squares fit-to-data portion of the cost function is

$$J[\vec{\phi}, f] = \frac{1}{2} \sum_{t=1}^{T} \sum_{k=1}^{K} ||s_k[\phi_t] \star f - d_{kt}||^2,$$

where $\vec{\phi} = (\phi_1, \ldots, \phi_T)$ and $s_k[\phi_t] = s[\phi_t + \theta_k]$. Let upper case letters denote Fourier transforms, * denotes complex conjugate, and

$$R_{kt} = S_k[\phi_t]F - D_{kt}.$$

Then the gradient of J with respect to f, denoted by g_f, has the following characterization:

$$\begin{aligned}
\langle g_f, w \rangle &= \frac{d}{d\tau} J[\vec{\phi}, f + \tau w]|_{\tau=0} \\
&= \frac{1}{2} \sum_{t,k} \frac{d}{d\tau} \langle S_k[\phi_t](F + \tau W) - D_{kt}, S_k[\phi_t](F + \tau W) - D_{kt} \rangle |_{\tau=0} \\
&= \frac{1}{2} \sum_{t,k} \left(\langle S_k[\phi_t]W, R_{kt} \rangle + \langle R_{kt}, S_k[\phi_t]W \rangle \right) \\
&= \frac{1}{2} \sum_{t,k} \left(\langle W, S_k[\phi_t]^* R_{kt} \rangle + \langle S_k[\phi_t]^* R_{kt}, W \rangle \right) \\
&= \frac{1}{2} \sum_{t,k} \Big(\langle w, \mathcal{F}^{-1}\{S_k[\phi_t]^* R_{kt}\} \rangle + \langle \mathcal{F}^{-1}\{S_k[\phi_t]^* R_{kt}\}, w \rangle \Big).
\end{aligned}$$

From this we obtain

$$\begin{aligned}
g_f &= \frac{1}{2} \sum_{t,k} \Big(\mathcal{F}^{-1}\{S_k[\phi_t]^* R_{kt}\} + \mathcal{F}^{-*}\{S_k[\phi_t]^* R_{kt}\} \Big) \\
&= \text{Real} \left[\mathcal{F}^{-1} \left\{ \sum_{t,k} S_k[\phi_t]^* R_{kt} \right\} \right].
\end{aligned}$$

Gradient Computations, Continued

To obtain the gradient with respect to ϕ_t, which we denote by g_t,

$$\begin{aligned}
\langle g_t, \xi \rangle &= \frac{d}{d\tau} J[\phi_1, \ldots, \phi_t + \tau\xi, \ldots, \phi_T, f]|_{\tau=0} \\
&= \frac{1}{2} \sum_{t,k} \frac{d}{d\tau} \langle S_k[\phi_t + \tau\xi]F - D_{kt}, S_k[\phi_t + \tau\xi]F - D_{kt} \rangle|_{\tau=0} \\
&= \frac{1}{2} \sum_k \left(\langle (S'_k[\phi_t]\xi)F, R_{kt} \rangle + \langle R_{kt}, (S'_k[\phi_t]\xi)F \rangle \right), \\
&= \frac{1}{2} \sum_k \left(\langle S'_k[\phi_t]\xi, Z_{kt} \rangle + \langle Z_{kt}, S'_k[\phi_t]\xi \rangle \right),
\end{aligned}$$

where $Z_{kt} = F^* R_{kt}$, and

$$\begin{aligned}
S'_k[\phi_t]\xi &= \frac{d}{d\tau} S_k[\phi_t + \tau\xi]|_{\tau=0} \\
&= \mathcal{F}\{ i h^*_{kt} \mathcal{F}^{-1}\{\xi H_{kt}\} + i^* h_{kt} \mathcal{F}^{-*}\{\xi H_{kt}\}\},
\end{aligned}$$

with $H_{kt} = p e^{\imath(\phi_t + \theta_k)}$ and $h_{kt} = \mathcal{F}^{-1}\{H_{kt}\}$. Note that $z_{kt} = \mathcal{F}^{-1}\{Z_{kt}\}$ is real-valued. Consequently,

$$\begin{aligned}
\langle g_t, \xi \rangle &= \sum_k \langle z_{kt}, i h^*_{kt} \mathcal{F}^{-1}\{\xi H_{kt}\} + i^* h_{kt} \mathcal{F}^{-*}\{\xi H_{kt}\} \rangle \\
&= \sum_k \langle i^* H^*_{kt} \mathcal{F}\{h_{kt} z_{kt}\}, \xi \rangle + \langle \xi, i^* H^*_{kt} \mathcal{F}\{h_{kt} z_{kt}\} \rangle \\
&= \sum_k \langle i^* H^*_{kt} \mathcal{F}\{h_{kt} z_{kt}\} + i H_{kt} \mathcal{F}^*\{h_{kt} z_{kt}\}, \xi \rangle,
\end{aligned}$$

and hence,

$$g_t = 2 \sum_k \mathrm{Imag}[H^*_{kt} \mathcal{F}\{h_{kt} \mathcal{F}^{-1}\{F^* R_{kt}\}\}].$$

Basic Algorithms for Unconstrained Optimization

Goal: Compute $x_* = \arg\min J(x)$.

Quasi-Newton / Line Search Algorithm

```
$k := 0$;
$x_0 :=$ initial guess for $x_*$;
begin quasi-Newton iterations
    $g_k := \nabla J(x_k)$;    % *compute gradient*
    Compute SPD approx $B_k$ to Hess $J(x_k)$;
    $d_k := -B_k^{-1} g_k$;    % *compute quasi-Newton step*
    $\tau_{k+1} := \arg\min_{\tau > 0} J(x_k + \tau d_k)$;    % *line search*
    $x_{k+1} := x_k + \tau_{k+1} d_k$;    % *update approx solution*
    $k := k + 1$;
end quasi-Newton iterations
```

Newton / Trust Region Algorithm

```
$k := 0$;
$x_0 :=$ initial guess for $x_*$;
$\Delta_0 :=$ initial trust region radius;
begin quasi-Newton iterations
    $g_k := \nabla J(x_k)$;    % *compute gradient*
    Compute solution $s_k$ to trust region subproblem
        $\min_s J(x_k) + g_k^T s + \frac{1}{2} s^T \text{Hess } J(x_k)\ s$
            subject to
        $||s|| \leq \Delta_k$
    $x_{k+1} := x_k + s_k$;    % *update approx solution*
    $k := k + 1$;
end quasi-Newton iterations
```

Limited Memory BFGS / Line Search Algorithm

Let $\mathbf{x}_k = (\phi_1^k, \ldots, \phi_T^k, f^k)$ denote approximate minimizer at iteration k, and suppose $B_k \approx \text{Hess } J(\mathbf{x}_k)$. The usual BFGS recursion for the B_k's give rise to the following recursion for the B_k^{-1}'s:

$$B_{k+1}^{-1} = \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) B_k^{-1} \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}, \tag{1}$$

where

$$s_k = x_{k+1} - x_k, \qquad y_k = \nabla J(x_{k+1}) - \nabla J(x_k).$$

B_{k+1} is guaranteed to be SPD provided that B_k is SPD and

$$y_k^T s_k > 0. \tag{2}$$

Remarks

- "Curvature condition" (2) can be guaranteed with a properly implemented line search.
- Using recursion (1), system $B_{k+1} s = -g$ can be solved using a sequence of vector dot products and one computation of the form $B_0^{-1} v$.
- Initial Hessian approx B_0 can be modified at each iteration k.
- "Limited memory" means that at most m vector pairs $\{(s_k, y_k), \ldots, (s_{k-m+1}, y_{k-m+1})\}$ are stored.
- Asymptotic convergence rate for full BFGS is superlinear. Rate for the limited memory variant is linear.

Newton/CG/Trust Region

Key Idea: To approximately solve Trust Region subproblem

$$\min_s m_k(s) \quad \text{subject to} \quad ||s|| \leq \Delta_k,$$

where

$$m_k(s) = J(x_k) + g_k^T s + \frac{1}{2} s^T H_k s,$$

apply CG iteration to minimize $m_k(s)$, or equivalently, use CG to solve

$$H_k s = -g_k. \tag{3}$$

Key Fact: Let s_ν denote CG iterates for (3). If $s_0 = 0$ and H_k is SPD, then for $\nu = 0, 1, \ldots,$

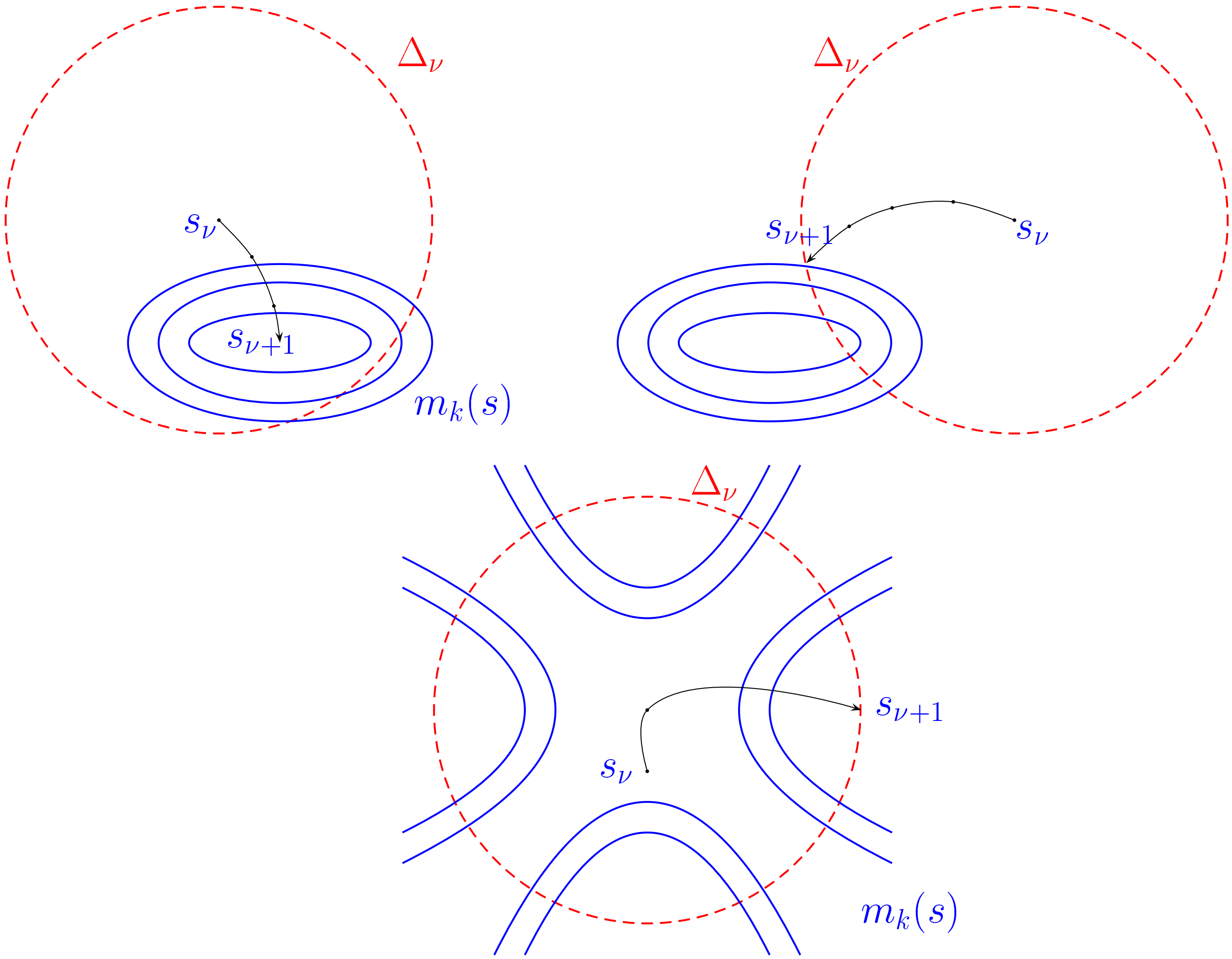
$$||s_{\nu+1}|| > ||s_\nu||,$$

$$m_k(s_{\nu+1}) < m_k(s_\nu)$$

Stop CG iteration when any of the following occur:

- $||s_{\nu+1}|| \geq \Delta_k$.
- CG residual $||H_k s_{\nu+1} + g_k|| \leq \epsilon_k$, where ϵ_k denotes stopping tolerance.
- Nonnegative curvature detected, i.e., $d_\nu^T H_k d_\nu < 0$, where d_ν denotes CG descent direction.

Trust Region Geometry



Preconditioning / Hessian Initialization

Infinite Dimensional Results.

- CG iteration for $(I+K)x=y$, K compact, converges superlinearly [Daniel, SINUM, 1967].
- Broyden iteration (quasi-Newton method for nonlinear systems) converges superlinearly provided initial Jacobian is compact perturbation of Jacobian at solution [Kelley and Sachs].

Structure of Hessian of regularized least squares function

$$\text{Hess } J[\phi_1,\dots,\phi_T,f] = H_{ls} + H_{reg}$$

Hessian of the regularization function has block diagonal form

$$H_{reg} = \begin{bmatrix} \alpha L & 0 & 0 & \cdots & 0 \\ 0 & \alpha L & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & \alpha L & 0 \\ 0 & \dots & 0 & 0 & \gamma I \end{bmatrix}$$

Hessian of the least squares fit-to-data term has "block arrow" form

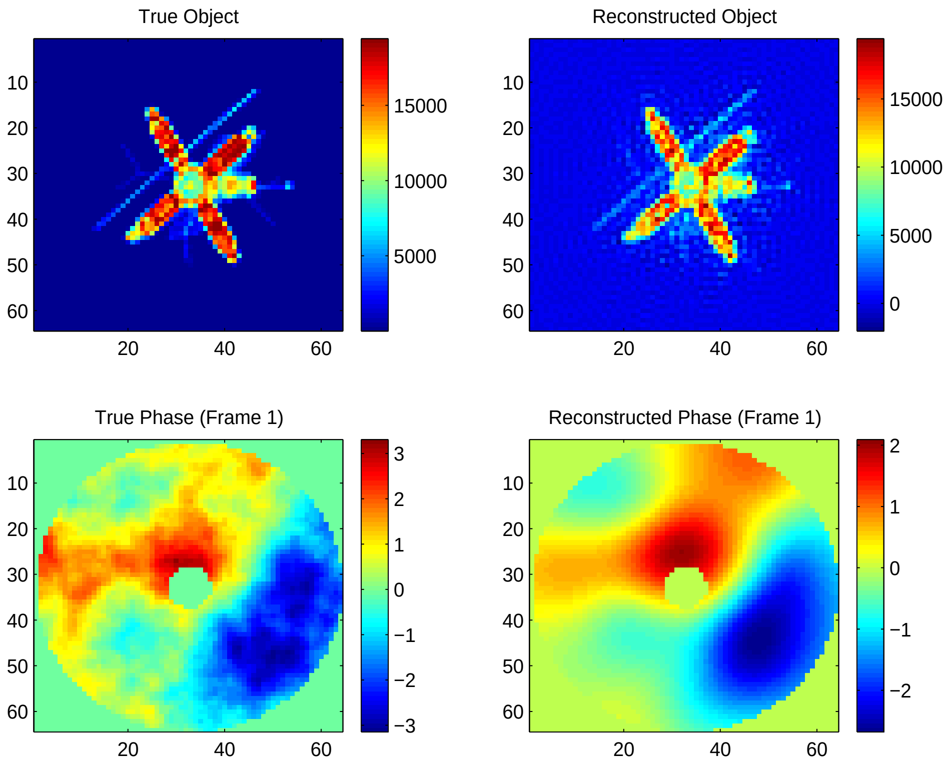
$$H_{ls} = \begin{bmatrix} H_{\phi_1\phi_1} & 0 & 0 & \cdots & H_{\phi_1 f} \\ 0 & H_{\phi_2\phi_2} & 0 & \ddots & H_{\phi_2 f} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & H_{\phi_T\phi_T} & H_{\phi_T f} \\ H_{f\phi_1} & H_{f\phi_2} & \cdots & H_{f\phi_T} & H_{ff} \end{bmatrix}$$

and

$$H_{ff} = \sum_{t=1}^{T}\sum_{k=1}^{K} S^*[\phi_t+\theta_k]S[\phi_t+\theta_k].$$

Numerical Results

Reconstructions obtained with 2 phase diversity channels and one time frame using the Newton / CG / trust region (NCGTR) method. Reconstructions obtained with L-BFGS are nearly identical.



True Object

Reconstructed Object

True Phase (Frame 1)

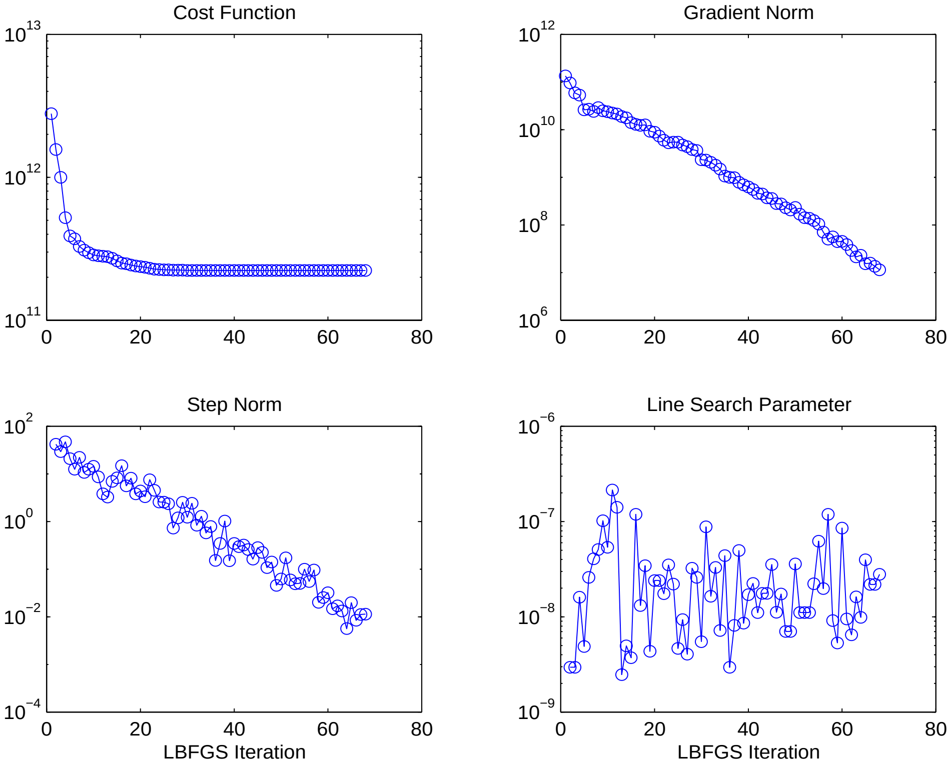
Reconstructed Phase (Frame 1)

Algorithmic Details

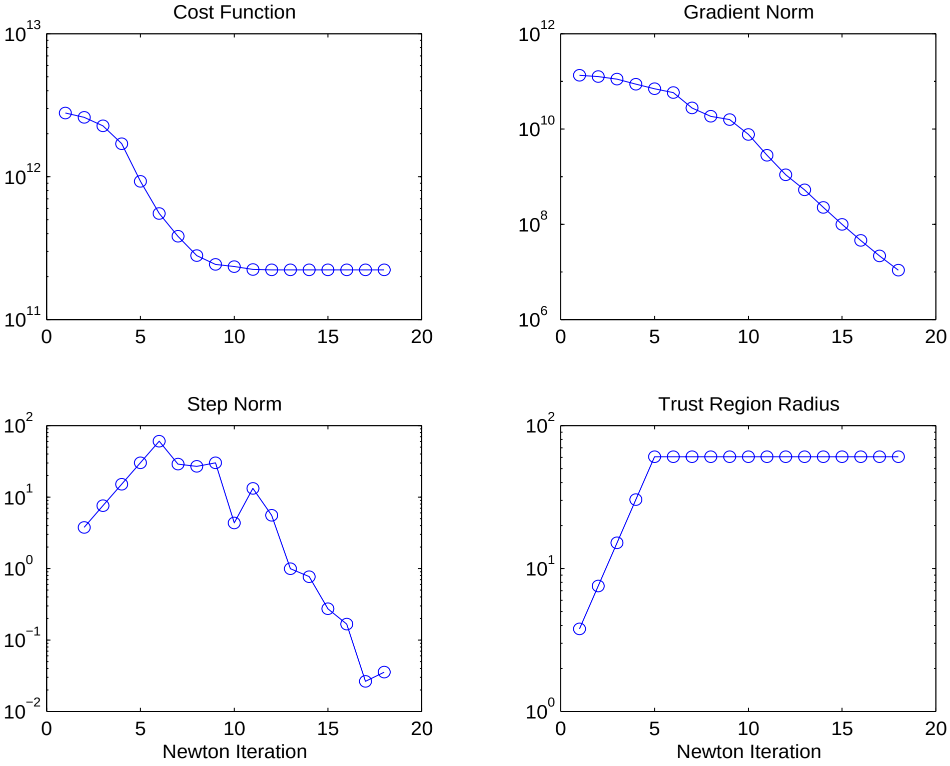
- H_{reg} is used as the initial Hessian for L-BFGS and as the preconditioner for NCGTR.
- Previous $m = 10$ vectors s_k, y_k were saved with L-BFGS.
- Previous step length was used to initialize the L-BFGS line search.
- Image/Phase screen size was $n = 128^2 = 16384$ pixels.
- CG residual stopping tolerance proportional to $||\nabla J(x)||^{3/2}$.

Numerical Performance

L-BFGS / Line Search Results

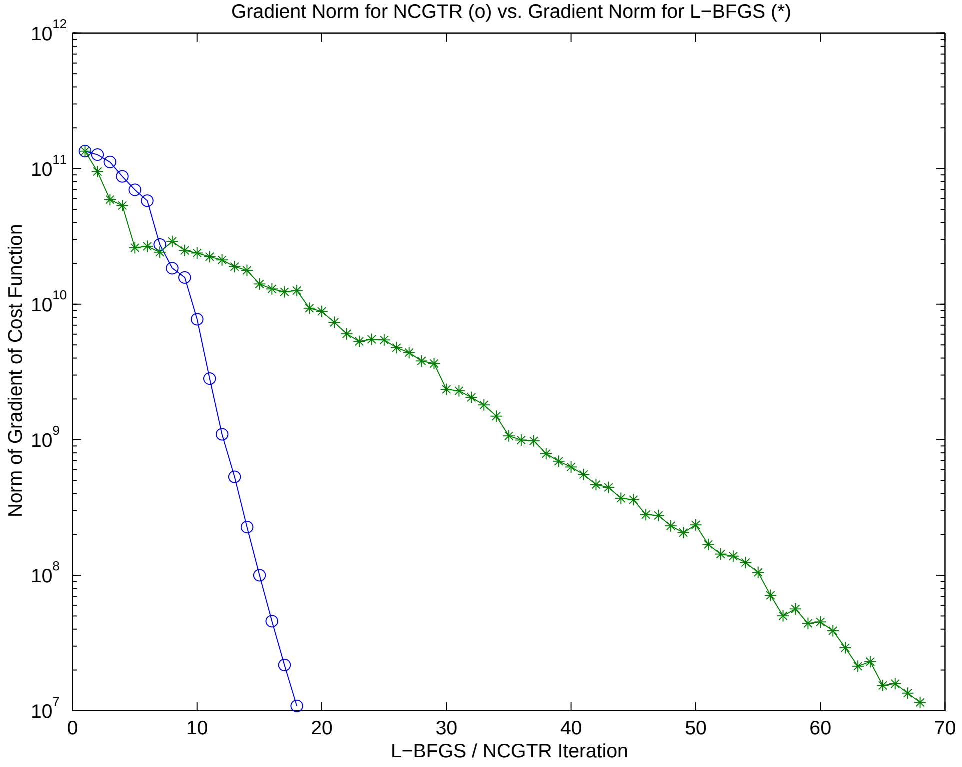


Newton / CG / Trust Region Results

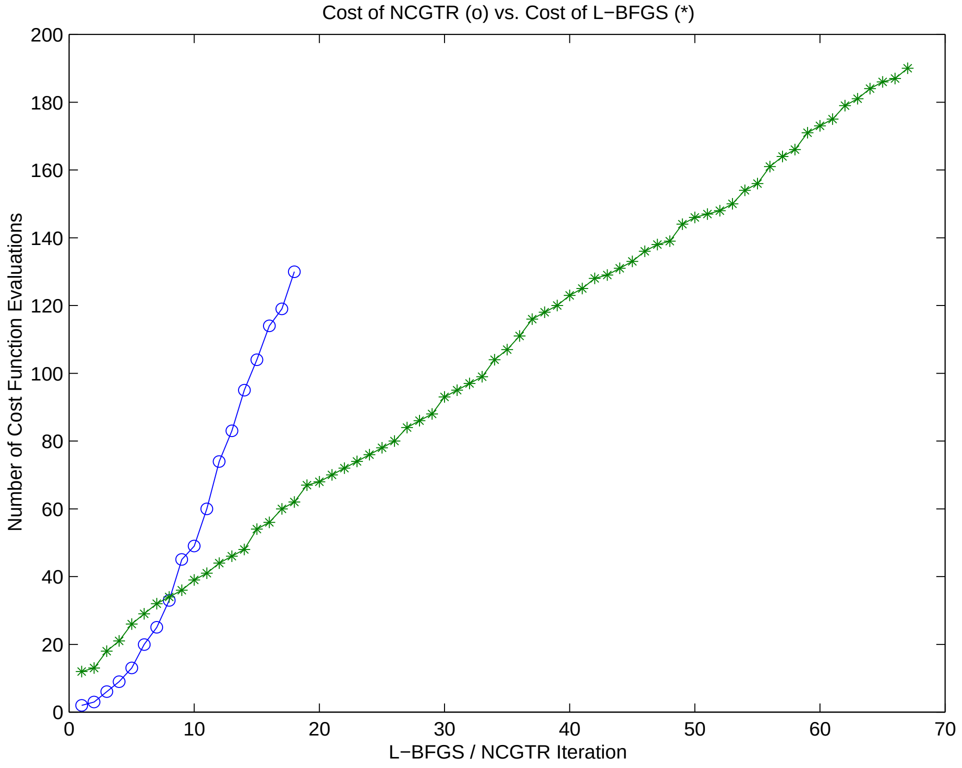


Cost Comparison

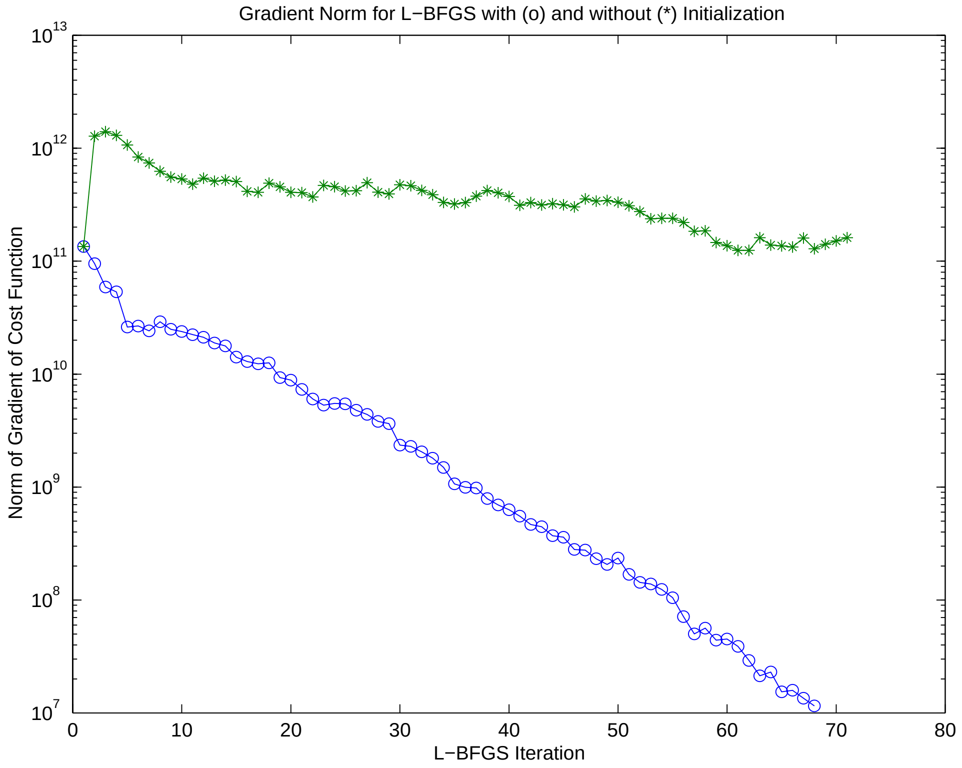
Gradient Norm vs. Iteration



Cumulative Function Evaluations vs. Iteration



Importance of Hessian Initialization



Stars $(*)$ indicate initial L-BFGS Hessian $B_0 = I$.

Circles (o) indicate $B_0 = H_{reg}$.

Conclusions

- Hessian initialization can make a dramatic difference in performance of L-BFGS.
- Steihaug's Newton / CG / Trust Region algorithm is competitive with limited memory BFGS for this application.