

Computational Methods for a Large-Scale Inverse Problem Arising in Atmospheric Optics

Luc Gilles† C. R. Vogel† § and J. M. Bardsley†

† Department of Mathematical Sciences, Montana State University, Bozeman MT
59717-2400

Abstract. The inverse problem of interest is the reconstruction of an astronomical object and wavefront aberrations due to atmospheric turbulence from a sequence of short time exposure (speckle) phase diversity images obtained from a ground-based telescope. A regularized least squares approach is taken, and two numerical techniques are applied to the resulting unconstrained optimization problem. The first technique is the limited memory BFGS method with line search globalization. The second is a Newton/trust region iteration in which the trust region subproblem is solved using a truncated conjugate gradient method. A numerical comparison based on real data is presented, and the roles of preconditioning and cost functional reduction (elimination of the object) are examined.

1. Introduction

As light propagates through the atmosphere, light rays are bent due to random variations in the index of refraction caused by atmospheric turbulence. This causes blurring of images of an astronomical object captured with a ground-based telescope [7]. The inverse problem of interest is to reconstruct the object from these blurred images. Unfortunately, the point spread function, or PSF, which characterizes the blur, is also unknown. The PSF can be accurately modeled in terms of the phase, or wavefront aberration, which quantifies the deviation from planarity of the wavefront. Phase diversity [3, 6, 9] provides additional observable information in the form of auxiliary images produced by introducing known additional wavefront aberrations. The inverse problem then becomes the estimation of both the object and the unknown wavefront aberrations from phase diversity image data. Mathematical modeling of phase-diverse image formation is the topic of Section 2.

Several features make the inverse problem very difficult to solve. The problem is ill-posed, so regularization is required. In Section 3 we apply a penalized least squares approach, which yields an unconstrained optimization problem whose unknowns are the discretized object and wavefront aberrations. One can eliminate the object [6, 11], thereby obtaining another minimization problem whose unknowns are the discretized

§ To whom correspondence should be addressed (vogel@math.montana.edu)

wavefront aberrations. While technically well-posed, these optimization problems can still be quite ill-conditioned. Moreover, the dependence of the image on the wavefront aberrations is nonlinear, so the cost functionals are not quadratic. Finally, due to the large number of pixels needed to obtain high resolution images, the problems have a large number of unknowns, and efficient numerical techniques are required. In Section 4 we discuss the implementation of a pair of methods for large scale unconstrained optimization to solve our inverse problem. The first is the limited memory BFGS (L-BFGS) method [5] with line search globalization. The second is a Newton iteration with trust region globalization in which the trust region subproblem is solved using a truncated conjugate gradient (CG) algorithm due to Steihaug and Toint [10, 1]. Preconditioning can improve the performance of both methods. In Section 4.3 we introduce preconditioners which make use of the special structure of the regularized least squares cost functionals.

In Section 5 we present results obtained from observed (not simulated) astronomical image data. The L-BFGS method and the Newton/truncated CG trust region method are both applied to minimize the (full) cost functional and its reduced variant. The storage requirements and the numerical performance are compared for the various combinations of method and cost functional. We also present the results of a numerical experiment which clearly demonstrates the effectiveness of our preconditioners. Conclusions are presented in Section 6

2. The Mathematical Model

We model short time exposure phase diversity atmospheric image data by

$$d_{k,t} = s[\phi_t + \theta_k] \star f_{true} + \eta_{k,t}, \quad k = 1, \dots, K, \quad t = 1, \dots, T. \quad (2.1)$$

Here f_{true} denotes the true *object*, or *light source*; $s[\cdot]$ represents the *point spread function* (PSF); $\eta_{k,t}$ represents noise; and the operator \star denotes two dimensional convolution,

$$(s \star f)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(x - x', y - y') f(x', y') dx' dy'.$$

For details see [7, 3, 6, 9, 11]. The PSF takes the form

$$s[\phi] = C |\mathcal{F}^{-1}\{pe^{i\phi}\}|^2, \quad (2.2)$$

where \mathcal{F} denotes the 2-D Fourier transform, ϕ represents the *phase*, or *wavefront aberration*, and p is the *pupil*, or *aperture, function*. We assume

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

where the region A represents the aperture and is an annulus in the case of imaging with large astronomical telescopes. The θ_k 's in (2.1) represent known phase distortions used with phase diversity. In the typical implementation, defocused diversity images are formed, and the θ_k 's are quadratic,

$$\theta_k = c_k \times (x^2 + y^2), \quad k = 1, \dots, K. \quad (2.3)$$

Phase diversity introduces independent information which eliminates some of the ambiguity, or nonuniqueness, in the determination of f from data $d = s \star f$ when s is also unknown. The number of diversity channels is usually $K = 2$.

The index t in (2.1) represents discrete time, and the ϕ_t 's represent time-varying phase aberrations due to temporal variations in the index of refraction of the atmosphere, caused by turbulence. Since f_{true} is assumed to be fixed for each t , taking $T \geq 2$ time frames may further reduce ambiguity and helps to reduce the effects of noise.

3. Regularized Cost Functionals

To estimate the phases ϕ_t , $t = 1, \dots, T$, and the object f_{true} from data (2.1), we consider the least squares fit-to-data functional

$$J_{data}[\vec{\phi}, f] = \frac{1}{2TK} \sum_{t=1}^T \sum_{k=1}^K \|s[\phi_t + \theta_k] \star f - d_{k,t}\|^2. \quad (3.1)$$

Here $\vec{\phi} = (\phi_1, \dots, \phi_T)$ and $\|\cdot\|$ denotes the standard L^2 norm. By the convolution theorem and the fact that Fourier transforms preserve L^2 norm, one can express this in terms of Fourier transforms, $F = \mathcal{F}\{f\}$, $D_{k,t} = \mathcal{F}\{d_{k,t}\}$, and $S_k[\phi_t] = \mathcal{F}\{s[\phi_t + \theta_k]\}$,

$$J_{data}[\vec{\phi}, F] = \frac{1}{2TK} \sum_{t=1}^T \sum_{k=1}^K \|S_k[\phi_t]F - D_{k,t}\|^2. \quad (3.2)$$

Since deconvolution and phase retrieval are both ill-posed problems, any minimizer of J_{data} is unstable with respect to noise in the data. Hence we add regularization terms to obtain the *full cost functional*,

$$J_{full}[\vec{\phi}, f] = J_{data}[\vec{\phi}, f] + \gamma J_{object}[f] + \alpha J_{phase}[\vec{\phi}]. \quad (3.3)$$

Here the regularization parameters γ and α are positive real numbers, and the regularization functionals J_{object} and J_{phase} provide stability and incorporate prior information.

Because of atmospheric turbulence, variations in the refractive index, and hence the phase itself, can be modeled as a random process [7]. We apply the von Karman turbulence model, which assumes this process is second order, wide sense stationary, and isotropic with zero mean. It can be characterized by its power spectral density,

$$\Phi(\omega) = \frac{C_1}{(C_2 + |\omega|^2)^{11/6}}, \quad (3.4)$$

where $\omega = (\omega_x, \omega_y)$ represents spatial frequency. Corresponding to this stochastic model for phase, we take the phase regularization functional

$$J_{phase}[\vec{\phi}] = \frac{1}{2T} \sum_{t=1}^T \langle \Phi^{-1} \mathcal{F}\{\phi_t\}, \mathcal{F}\{\phi_t\} \rangle, \quad (3.5)$$

where $\langle f, g \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\omega) g^*(\omega) d\omega$, and the superscript $*$ denotes complex conjugate. For regularization of the object, we take the ‘‘minimal information prior’’

$$J_{object}[f] = \frac{1}{2} \|f\|^2 = \frac{1}{2} \|F\|^2. \quad (3.6)$$

Note that the object regularization functional (3.6) is quadratic and the dependence of the fit-to-data functional (3.1) on the object f is quadratic. Moreover, the Hessian with respect to the object of the full cost functional (3.3) is symmetric and positive with eigenvalues bounded below by γ . By setting the gradient with respect to the object equal to zero, one obtains a linear equation whose solution yields the object at a minimizer for J_{full} [6]. From (3.2)-(3.3) and (3.6) one obtains the Fourier representation for the minimizing object,

$$F = \frac{P[\vec{\phi}]^*}{Q[\vec{\phi}]}, \quad (3.7)$$

where

$$P[\vec{\phi}] = \sum_{k,t} D_{k,t}^* S_k[\phi_t], \quad Q[\vec{\phi}] = \gamma + \sum_{k,t} |S_k[\phi_t]|^2. \quad (3.8)$$

Substituting (3.7) back in (3.2)-(3.3), one obtains the reduced cost functional,

$$J_{reduced}[\vec{\phi}] = J_{reduced\ data}[\vec{\phi}] + \alpha J_{phase}[\vec{\phi}], \quad (3.9)$$

where

$$J_{reduced\ data}[\vec{\phi}] = \sum_{k,t} \|D_{k,t}\|^2 - \left\langle \frac{P[\vec{\phi}]}{Q[\vec{\phi}]}, P[\vec{\phi}] \right\rangle. \quad (3.10)$$

See Appendix B of [11] for a detailed derivation.

4. Computational Methods

In the following discussion, x denotes a discretization of the phases $\vec{\phi} = (\phi_1, \dots, \phi_T)$ in the reduced case (3.9), together with a discretization of the object f in the full case (3.3). We assume the resulting cost functional J is a twice continuously differentiable mapping from \mathbb{R}^n into \mathbb{R}^1 with gradient at x denoted by $\text{grad } J(x)$ and Hessian (matrix of second partial derivatives) denoted by $\text{Hess } J(x)$. Given an initial guess x_0 for a minimizer x_* of J , the Newton iteration,

$$x_{\nu+1} = x_{\nu} - \text{Hess } J(x_{\nu})^{-1} \text{grad } J(x_{\nu}), \quad \nu = 0, 1, \dots, \quad (4.1)$$

yields a sequence of approximations x_{ν} to x_* . If $\text{Hess } J(x_*)$ is symmetric positive definite (SPD), J is sufficiently smooth, and x_0 is sufficiently close to x_* , then x_{ν} will converge to x_* at a quadratic rate [2]. If x_0 is not sufficiently close, modifications to the basic Newton iteration called globalization techniques can be incorporated to ensure that $J(x_{\nu})$ decreases in a manner which guarantees convergence to a local minimizer of J . For large scale (large n) problems, an additional shortcoming of Newton's method is the need to assemble and invert the Hessian.

4.1. The Limited Memory BFGS /Line Search Method

The cost of storing and inverting large Hessian matrices motivates quasi-Newton, or secant, methods [2, 5]. These use gradient information to construct an approximation B_ν to $\text{Hess } J(x_\nu)$. If B_ν is SPD, then the quasi-Newton step $d_\nu = -B_\nu^{-1} \text{grad } J(x_\nu)$ is guaranteed to be a descent direction for J at x_ν , i.e., $J(x_\nu + \tau d_\nu) < J(x_\nu)$ whenever τ is sufficiently small. Displayed below is a generic quasi-Newton algorithm with line search globalization.

Quasi-Newton / Line Search Algorithm

```

ν := 0;
x₀ := initial guess for x*;
begin quasi-Newton iterations
    g_ν := grad J(x_ν);           % compute gradient
    B_ν := SPD approximation to Hess J(x_ν);
    d_ν := -B_ν⁻¹ g_ν;           % compute quasi-Newton step
    τ_ν := arg min_{τ>0} J(x_ν + τ d_ν);   % line search
    x_{ν+1} := x_ν + τ_ν d_ν;     % update approximate solution
    ν := ν + 1;
end quasi-Newton iterations

```

In the practice, the line search subproblem is solved inexactly [5].

One of the most popular quasi-Newton techniques is the BFGS method [2, 5], discovered independently by Broyden, Fletcher, Goldfarb, and Shanno. Given B_ν , this method generates a new Hessian approximation $B_{\nu+1}$ in terms of the differences in the successive approximates to the solution and its gradient,

$$s_\nu \stackrel{\text{def}}{=} x_{\nu+1} - x_\nu, \quad (4.2)$$

$$y_\nu \stackrel{\text{def}}{=} \text{grad } J(x_{\nu+1}) - \text{grad } J(x_\nu). \quad (4.3)$$

A variant of BFGS known as limited memory BFGS (L-BFGS) [5] is based on a recursion for the inverse of the B_ν 's,

$$B_{\nu+1}^{-1} = \left(I - \frac{s_\nu y_\nu^T}{y_\nu^T s_\nu} \right) B_\nu^{-1} \left(I - \frac{y_\nu s_\nu^T}{y_\nu^T s_\nu} \right) + \frac{s_\nu s_\nu^T}{y_\nu^T s_\nu}, \quad \nu = 0, 1, \dots \quad (4.4)$$

Given $v \in \mathbb{R}^n$, computation of $B_{\nu+1}^{-1}v$ requires a sequence of inner products involving v and the s_ν 's and y_ν 's, together with the application of B_0^{-1} . If B_0 is SPD and the ‘‘curvature condition’’ $y_\nu^T s_\nu > 0$ holds for each ν , then each of the B_ν 's is also SPD, thereby guaranteeing that $-B_\nu^{-1} \text{grad } J(x_\nu)$ is a descent direction. The curvature condition can be maintained by implementing the line search correctly [5].

‘‘Limited memory’’ means that at most N vector pairs $\{(s_\nu, y_\nu), \dots, (s_{\nu-N+1}, y_{\nu-N+1})\}$ are stored and at most N steps of the recursion are taken, i.e., if $\nu \geq N$, apply the recursion (4.4) for $\nu, \nu-1, \dots, \nu-N$, and set $B_{\nu-N}^{-1}$ equal to an SPD matrix M_ν^{-1} . We will refer to M_ν as the *preconditioning matrix*. In standard implementations, M_ν is taken to

be a multiple of the identity [5]. Choices of M_ν based on structure of the cost functions in Section 3 will be presented in Section 4.3.

Under mild assumptions, the local convergence rate for the BFGS method is superlinear [2, 5]. In the limited memory case, this rate becomes linear.

4.2. The Newton/CG/Trust Region Method

An alternative to a quasi-Newton iteration with line search globalization is the Newton / trust region approach [1, 2, 5]. The key idea is to form at each iteration ν a quadratic approximation to $J(x_\nu + s)$,

$$m_\nu(s) = J(x_\nu) + g_\nu^T s + \frac{1}{2} s^T H_\nu s, \quad (4.5)$$

where $g_\nu = \text{grad } J(x_\nu)$ and $H_\nu = \text{Hess } J(x_\nu)$. Then solve the constrained minimization problem

$$\min_s m_\nu(s) \quad \text{subject to} \quad \|s\|_\nu \leq \Delta_\nu, \quad (4.6)$$

where $\|\cdot\|_\nu$ is a matrix norm which may depend on the Newton iteration ν , and Δ_ν is a positive parameter called the *trust region radius*. This parameter is adjusted at each iteration ν in a manner which guarantees that (i) $m_\nu(s)$ accurately approximates $J(x_\nu + s)$ within the trust region $\{s \mid \|s\|_\nu \leq \Delta_\nu\}$; and (ii) the constrained minimizer s_ν for (4.5)-(4.6) yields a new approximate $x_{\nu+1} = x_\nu + s_\nu$ which sufficiently decreases J in order to guarantee convergence to a minimizer. See [1, 5] for details.

Problem (4.5)-(4.6) has no closed-form solution, and obtaining a highly accurate approximation can be very expensive. For large-scale problems, a very effective approximate solution technique is the Steihaug-Toint algorithm. The key idea is to apply conjugate gradient (CG) iterations to minimize $m_\nu(s)$, or equivalently, to solve the linear system $H_\nu s = -g_\nu$. If the initial guess $s_{\nu,0}$ is zero and H_ν is SPD, the CG iterates $s_{\nu,j}$ monotonically increase in norm and monotonically decrease $m_\nu(s)$. Eventually, either the boundary of the trust region is crossed or the minimizer is attained. If H_ν is indefinite, a check for negative curvature is added. If $d_{\nu,j}^T H_\nu d_{\nu,j} \leq 0$, where $d_{\nu,j}$ denotes the j^{th} CG search direction, then one moves in this direction from the current CG iterate until the trust region boundary is crossed, and the CG iteration is terminated. Preconditioning can also be incorporated to accelerate CG converge. In this case the trust region is determined by the vector norm induced by the SPD preconditioning matrix M_ν ,

$$\|s\|_{M_\nu} \stackrel{\text{def}}{=} \sqrt{s^T M_\nu s}. \quad (4.7)$$

The selection of M_ν is discussed in Section 4.3.

The Steihaug-Toint Truncated CG Algorithm

Given gradient $g = \text{grad } J(x_\nu)$, Hessian $H = \text{Hess } K(x_\nu)$, preconditioning matrix $M = M_\nu$ and trust region radius $\Delta = \Delta_\nu$, to approximately solve the trust region subproblem (4.5)-(4.6):

```

s0 := 0;
r0 := gν;      % Initial CG residual
y0 := M-1r0;    % Apply preconditioner
d0 := -y0;      % Initial CG search direction
j := 0;
Begin CG iterations
  κj := djTHdj;
  If κj ≤ 0,      % Check for indefinite H
    τj := positive root of ||sj + τdj||M = Δ;
    sj+1 := sj + τjdj;    % Move to boundary
    Stop.
  End if
  αj := rjTyj/κj;
  If ||sj + αjdj||M ≥ Δ,    % Boundary crossed
    τj := positive root of ||sj + τdj||M = Δ;
    sj+1 := sj + τjdj;    % Backtrack to boundary
    Stop.
  End if
  sj+1 := sj + αjdj;    % Update solution
  rj+1 := rj + αjHdj;    % Update CG residual
  yj+1 := M-1rj+1;    % Apply preconditioner
  βj+1 := rj+1Tyj+1/rjTyj;
  dj+1 := -rj+1 + βj+1dj;    % Update CG search direction
  j := j + 1;
end CG iterations

```

4.3. Choice of the Preconditioner

Consider the linear system $HS = -g$, where H is SPD. If M is also SPD, then one can solve the transformed system $(M^{-1/2}HM^{-1/2})\tilde{s} = -M^{-1/2}g$ and then backtransform $s = M^{-1/2}\tilde{s}$ to solve the original linear system. In principle one can apply CG iteration to the transformed system to obtain a sequence of approximate solutions to the original system. In practice, this transformed CG iteration can be reformulated to require M^{-1} rather than $M^{-1/2}$. M is called the *preconditioner* and the resulting iteration is called the *preconditioned conjugate gradient* (PCG) algorithm [8]. For this method to be effective, the spectrum of the preconditioned operator $M^{-1/2}HM^{-1/2}$ (or equivalently, the spectrum of $M^{-1}H$) should be more clustered than the spectrum of H itself. In addition, M should be easy to invert.

Now let H_{full} denote the Hessian of the full cost function, cf., (3.3). This Hessian takes the form

$$H_{full}[\vec{\phi}, f] = H_{data}[\vec{\phi}, f] + H_{reg}, \quad (4.8)$$

where, with the regularization functionals (3.6) and (3.5), the Hessian of the

regularization terms is a $(T + 1) \times (T + 1)$ block diagonal matrix,

$$H_{reg} = \begin{bmatrix} \frac{\alpha}{T}L & 0 & \cdots & 0 & 0 \\ 0 & \frac{\alpha}{T}L & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & \frac{\alpha}{T}L & 0 \\ 0 & 0 & \cdots & 0 & \gamma I \end{bmatrix}, \quad (4.9)$$

and the operator L in (4.9) has a Fourier representation,

$$L\psi = \mathcal{F}^{-1} \left\{ \Phi^{-1} \mathcal{F}\{\psi\} \right\}. \quad (4.10)$$

The Hessian of the full fit-to-data functional in (4.8) has a $(T + 1) \times (T + 1)$ block arrow representation,

$$H_{data}[\vec{\phi}, f] = \begin{bmatrix} H_{11} & 0 & \cdots & 0 & H_{1f} \\ 0 & H_{22} & 0 & \ddots & H_{2f} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & H_{TT} & H_{Tf} \\ H_{f1} & H_{f2} & \cdots & H_{fT} & H_{ff} \end{bmatrix}. \quad (4.11)$$

The nonzero off-diagonal terms in (4.11) depend on both $\vec{\phi}$ and f . The diagonal terms for $t = 1, \dots, T$ depend only on ϕ_t . From (3.2), the remaining diagonal term H_{ff} has a Fourier representation

$$H_{ff}[\vec{\phi}]v = \mathcal{F}^{-1} \left\{ \frac{1}{KT} \sum_{k=1}^K \sum_{t=1}^T |S_k[\phi_t]|^2 \mathcal{F}\{v\} \right\}. \quad (4.12)$$

To minimize the full cost functional, we select the preconditioner $M_\nu = M[\vec{\phi}_\nu]$ with the block diagonal representation

$$M[\vec{\phi}] = \begin{bmatrix} \frac{\alpha}{T}L & 0 & \cdots & 0 & 0 \\ 0 & \frac{\alpha}{T}L & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & \frac{\alpha}{T}L & 0 \\ 0 & 0 & \cdots & 0 & H_{ff}[\phi] + \gamma I \end{bmatrix}. \quad (4.13)$$

This incorporates the Hessian of the full cost regularization functional, cf., (4.9). It also includes the term H_{ff} from the Hessian of the full fit-to-data functional, cf., (4.11)-(4.12). From (4.10) and (4.12), one can easily apply M_ν and M_ν^{-1} using Fourier transforms. Numerical experiments indicate that the eigenvalues of H_{data} cluster at zero. Hence, the preconditioned operator should have eigenvalues which cluster at one. For this reason, this preconditioner should improve the performance of the Steihaug-Toint algorithm.

Now consider the reduced cost function in (3.9). Its Hessian has a representation

$$H_{reduced}[\vec{\phi}] = H_{reduced\ data}[\vec{\phi}] + \alpha H_{phase\ reg}. \quad (4.14)$$

Here $H_{\text{phase reg}}$ denotes the Hessian of the phase regularization functional, cf., (3.5). This has a $T \times T$ block diagonal representation, with diagonal components each consisting of $\frac{1}{T}L$. The Hessian of the reduced fit-to-data functional (3.10) also has a $T \times T$ block representation, but because of the coupling introduced by the reduction (3.7)-(3.8), none of the blocks of $H_{\text{reduced data}}$ are zero, and none have a simple representation. For the preconditioner in the reduced case, we take

$$M = \alpha H_{\text{phase}}. \quad (4.15)$$

This is the leading $T \times T$ subblock in (4.9) and is independent of both $\vec{\phi}$ and f . As in the full case, this preconditioner can be easily inverted using its Fourier representation, cf., (4.10), and the eigenvalues of the preconditioned operator appear to cluster at one.

BFGS iteration is closely related to CG iteration. In particular, when BFGS is applied to a quadratic cost function with an SPD Hessian, exact line searches, and initial Hessian $B_0 = M$, one obtains exactly the same iterates as if one had applied PCG iteration with preconditioner M . Consequently, the above preconditioners should also improve the performance of L-BFGS iteration.

5. Numerical Results

In this section we present results obtained when the algorithms of Section 4 were applied to actual data. The data were obtained from a 2-channel phase diversity system incorporated into a 1.6 meter telescope at the US Air Force's Maui Space Surveillance Complex on Mount Haleakala on the island of Maui, Hawaii. These data consists of 4 10-millisecond exposure images of a binary star. Thus $K = 2$ in equation (2.1) and $T = 4$. The in-focus data for diversity channel 1 (corresponding to $c_1 = 0$ in (2.3)) is shown in Fig. 1. Fig. 2 shows the out-of-focus data from phase diversity channel 2. Centroids of the image data were computed and the images were shifted to move these centroids to the centers of the frames. The reconstructed object is shown in Fig. 3. The binary star can clearly be seen in the reconstruction. The corresponding wavefront aberrations are shown in Fig. 4. The annular shape of the telescope's aperture accounts for the presentation of the phase reconstructions in an annular region.

We next discuss numerical performance. Here L-BFGS denotes our implementation of the limited memory BFGS algorithm with line search globalization. $N = 20$ vector pairs s_ν, y_ν were saved. NCGTR denotes our implementation of Newton iteration with trust region globalization obtained from the Steihaug-Toint truncated CG algorithm. Both methods were applied to minimize the full cost functional (3.3) and the reduced cost functional (3.9). It should be noted that numerical performance, as well as the actual object and phase reconstructions, varies with the choice of regularization parameters. In all the cases the regularization parameters were taken to be $\alpha = 5 \times 10^{-1}$ and $\gamma = 5 \times 10^{-3}$. These values were selected by trial and error to provide a reconstructed object that clearly looked like a binary star.

Unfortunately, due to the nonlinearity of the phase-to-observation map, the cost functions need not be convex and may have spurious local minima. Hence, the initial

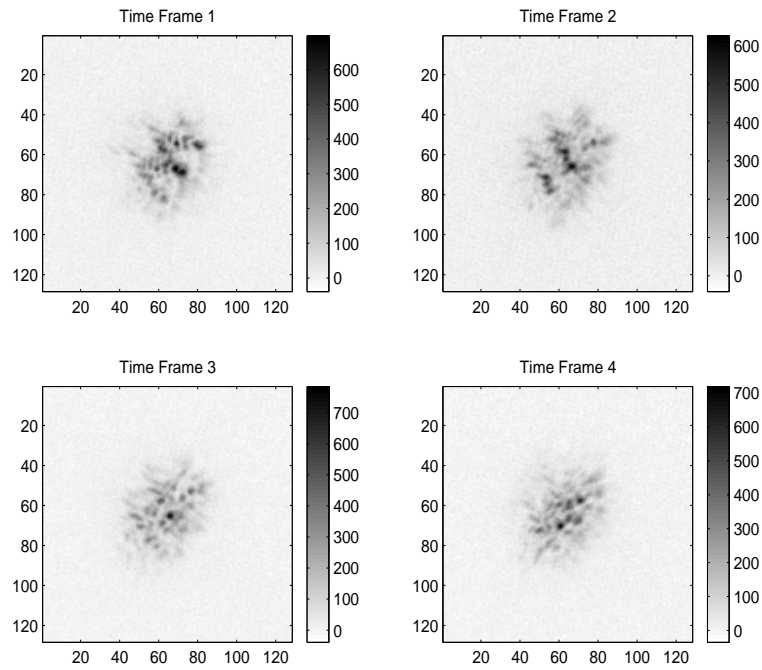


Figure 1. *In-focus (Channel 1) phase diversity image data.*

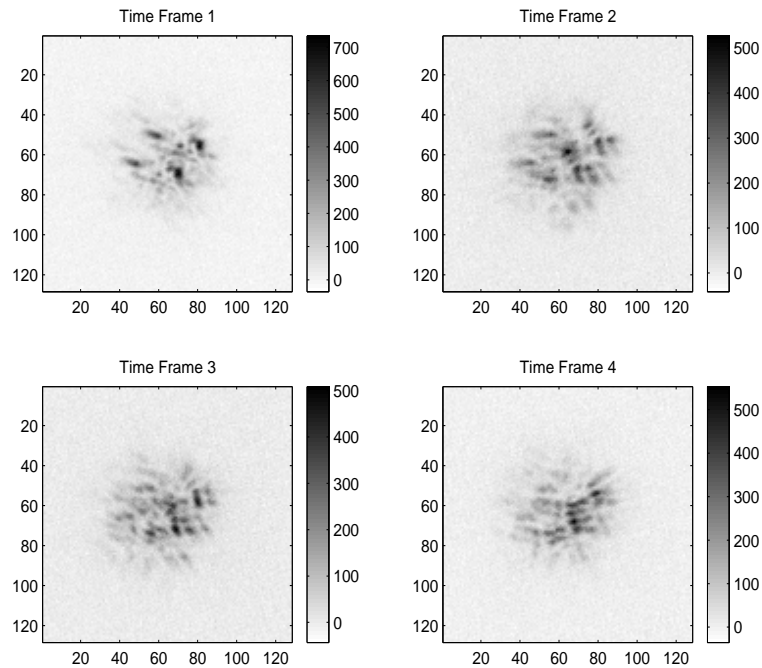


Figure 2. *Out-of-focus (Channel 2) phase diversity image data.*

guess may also influence the reconstructions. The initial guess may also influence the numerical performance. For both methods, the phases were initialized to zero ($\phi_t = 0$ for $t = 1, 2, 3, 4$) in the minimization of the reduced cost functional. For both methods

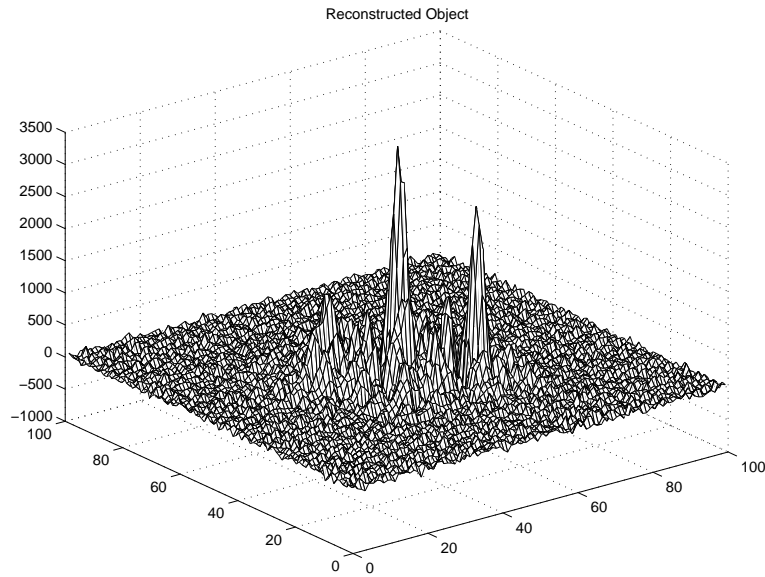


Figure 3. *Reconstruction of the object, a binary star. This reconstruction was generated using L-BFGS with Hessian initialization (4.13) applied to the full cost functional.*

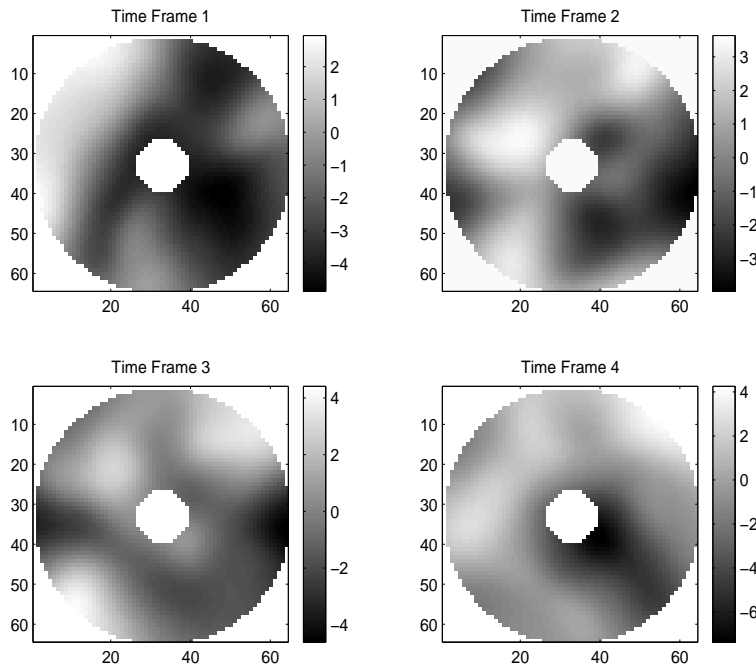


Figure 4. *Reconstructed phases, or wavefront aberrations.*

in the case of the full cost functional, the same phase initialization was used, and the initial object was generated by deconvolving the data using the corresponding PSF's, cf., (3.7)-(3.8).

Fig. 5 shows performance indicators for NCGTR applied to the full cost functional

(3.3). From the step norm (lower left) and gradient norm (upper right) subplots in this figure we see rapid local linear convergence to a minimizer, beginning at about (outer) Newton iteration 35. A quadratic rate of convergence was never attained because at most 50 (inner) PCG iterations were allowed in the inexact solution of the trust region subproblem.

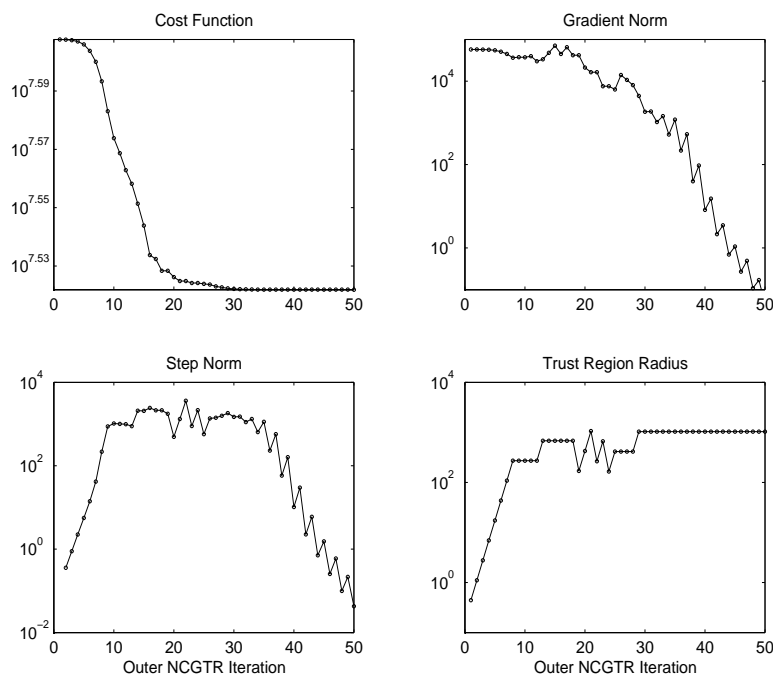


Figure 5. *NCGTR* performance for full cost functional minimization. In each subplot, the horizontal axis represents outer iteration.

In Fig. 6 we display performance indicators for L-BFGS applied to the full cost functional (3.3). The step norm and gradient norm subplots again show linear convergence, now due to the fact that at most $N = 20$ s and y vectors were saved for the recursion (4.4).

A comparison of Fig. 5 and Fig. 6 shows that many more iterations of L-BFGS than outer iterations of NCGTR were required to reach a particular gradient or step tolerance. However, this does not account for the fact that each outer NCGTR iteration may require a number of inner PCG iterations. In Fig. 7 we compare the *total* computational expense of the two methods, measured in terms of the number of forward and inverse fast Fourier transforms (FFT's). For this particular test problem, the computational expense of attaining a particular gradient norm with NCGTR is typically at least twice the expense of L-BFGS.

We next illustrate the effects on numerical performance of reduction in the cost functional by elimination of the object, cf., (3.9)-(3.10). Fig. 8 shows the gradient norm vs NCGTR outer iteration for the full cost functional (solid line) and for the reduced cost functional (dashed line). Corresponding results for L-BFGS are shown in Fig. 9. From

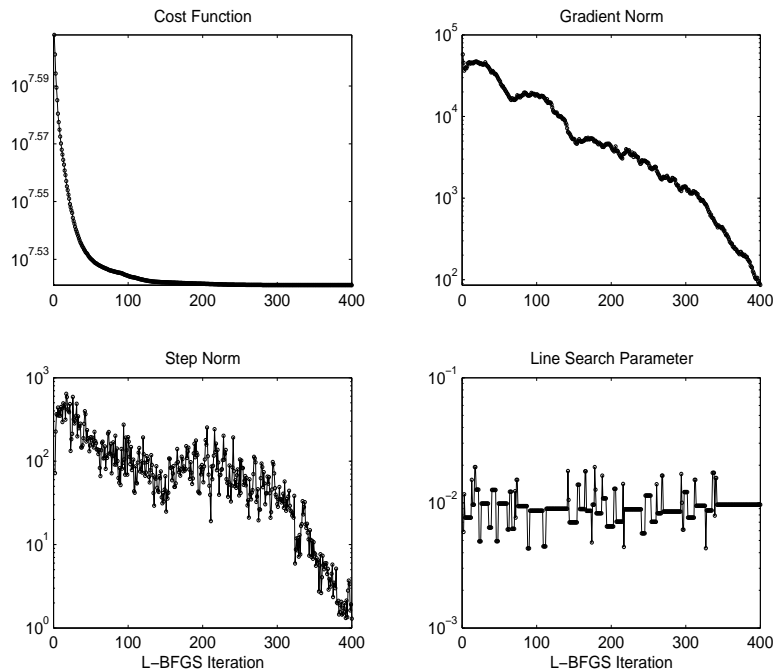


Figure 6. *L-BFGS* performance for full cost functional minimization.

both figures we see that reducing the cost functional substantially increases convergence rates and decreases computational cost. One might conjecture that this performance improvement is associated with the decrease in the number of unknowns that occurs with elimination of the discretized object. However, this decrease becomes relatively small as the number of time frames T increases. A more likely explanation is that the reduced problem is less ill-conditioned than the full minimization problem.

Finally, we consider the effects of the preconditioners / Hessian initializations introduced in Section 4.3. In Fig. 10 we present results for L-BFGS applied to the minimization of the full cost functional. These results are consistent with all the other combinations of methods and full/reduced cost functionals. The top (dashed) line in this figure, representing the gradient norm obtained with standard identity Hessian initialization, initially increases by an order of magnitude. After decreasing slightly in the next 200 iterations, it remains essentially flat. The corresponding reconstructed object, which is presented in Fig. 11, is significantly worse than that in Fig. 3, obtained using L-BFGS with the initial Hessian (4.13). In this test case, preconditioning dramatically improves both the numerical performance and the quality of the reconstructions.

6. Conclusions

Two numerical optimization techniques, limited memory BFGS (L-BFGS) and a Newton/conjugate gradient/trust region method (NCGTR) were applied to large scale

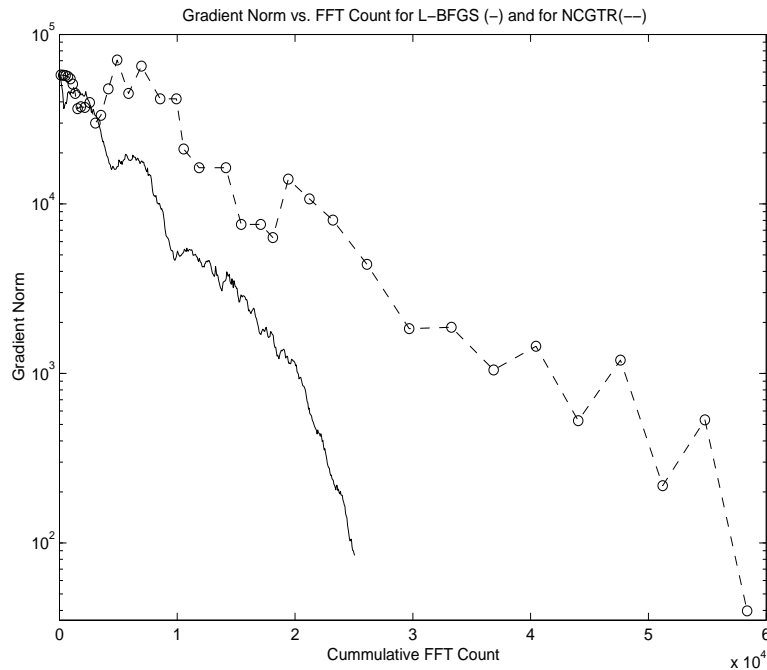


Figure 7. Comparison of computational expense for full cost functional minimization. The horizontal axis denotes cumulative number of forward and inverse FFT's. The vertical axis denotes gradient norm. The solid line represents L-BFGS; the dashed line represents NCGTR; and circles (o) denote outer NCGTR iterations.

minimization problems arising in the reconstruction of phases and object from phase diversity image data. Two separate cost functionals were considered: (i) the full cost functional, which has both phases and the object as unknowns; and (ii) the reduced cost functional, which has only the phases as unknowns. Preconditioners based on the special structure of these cost functionals were introduced. From a numerical study based on observed image data, we draw the following conclusions:

- Preconditioning dramatically improves the numerical performance of both methods, and it dramatically improves the quality of the reconstructed images.
- For our particular test problem, the computational expense of L-BFGS is typically half that of NCGTR.
- With either method, the computational expense of obtaining highly accurate minimizers for the reduced cost functional was less than the corresponding expense for the full cost functional. This difference was more dramatic with L-BFGS than with NCGTR.
- For this particular test problem, in terms of reducing computational expense, the best combination of method and cost functional was L-BFGS applied to the reduced cost functional.

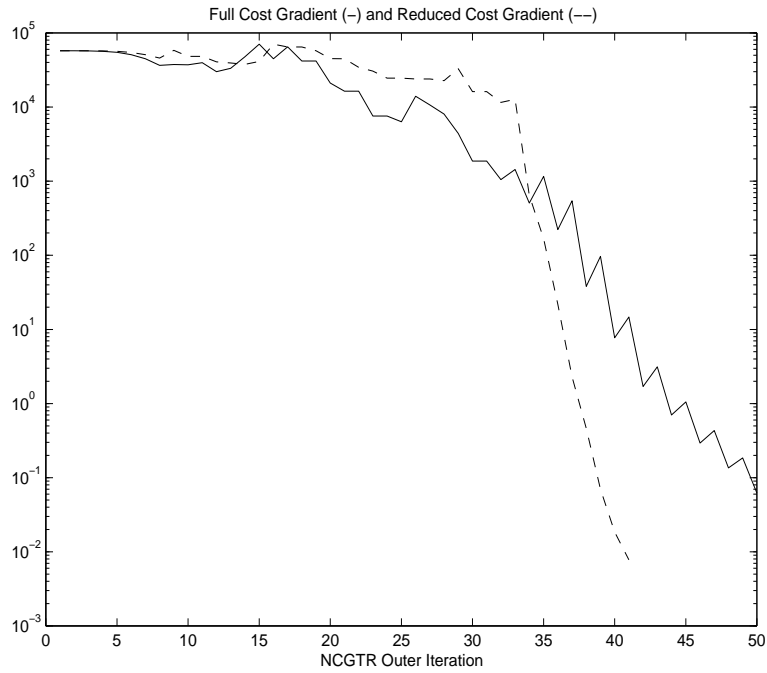


Figure 8. *NCGTR* gradient norm vs outer iteration count for minimization of the full cost functional (solid line) and for the reduced cost functional (dashed line).

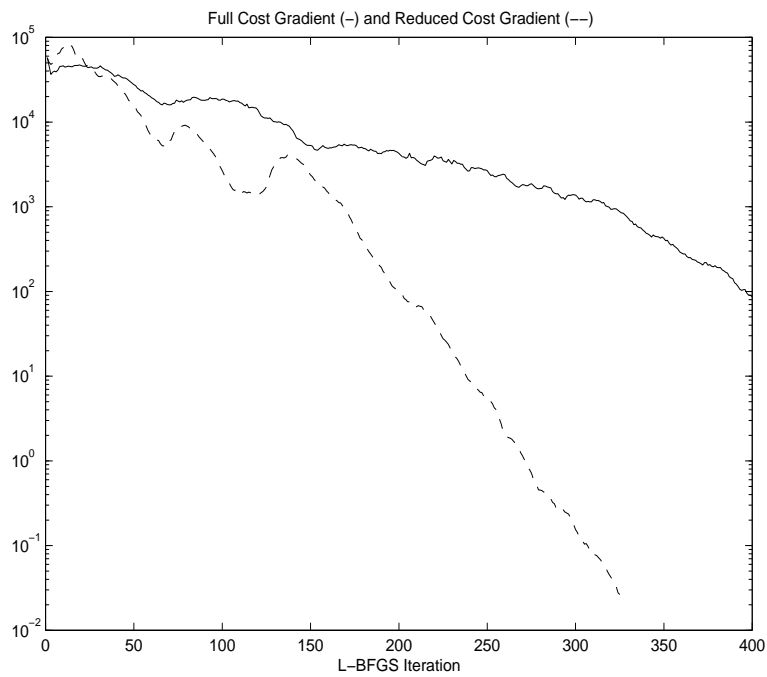


Figure 9. *L-BFGS* gradient norm vs iteration count for minimization of the full cost functional (solid line) and for the reduced cost functional (dashed line).

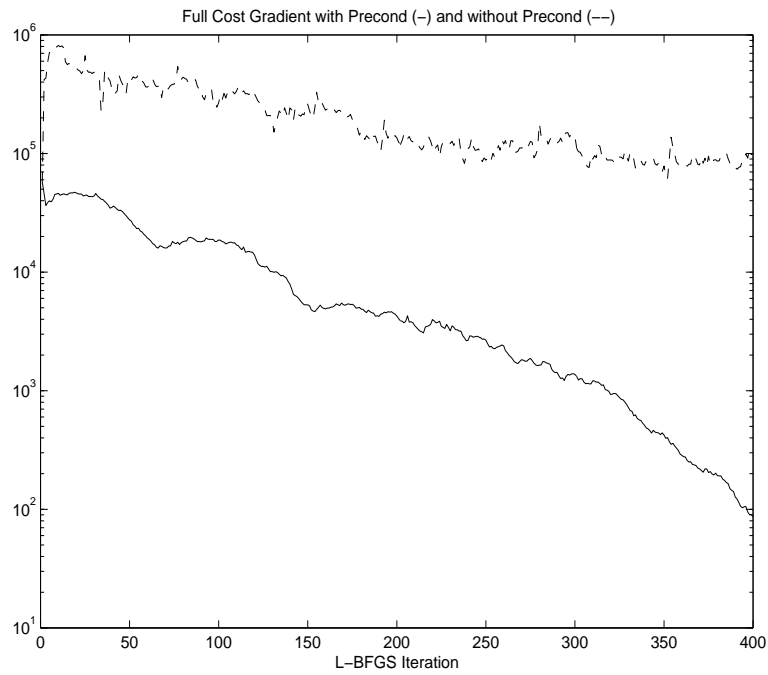


Figure 10. Gradient norm vs iteration count for L-BFGS minimization of the full cost functional using standard identity Hessian initialization (dashed line) and using the initial Hessian presented in Section 4.3 (solid line).

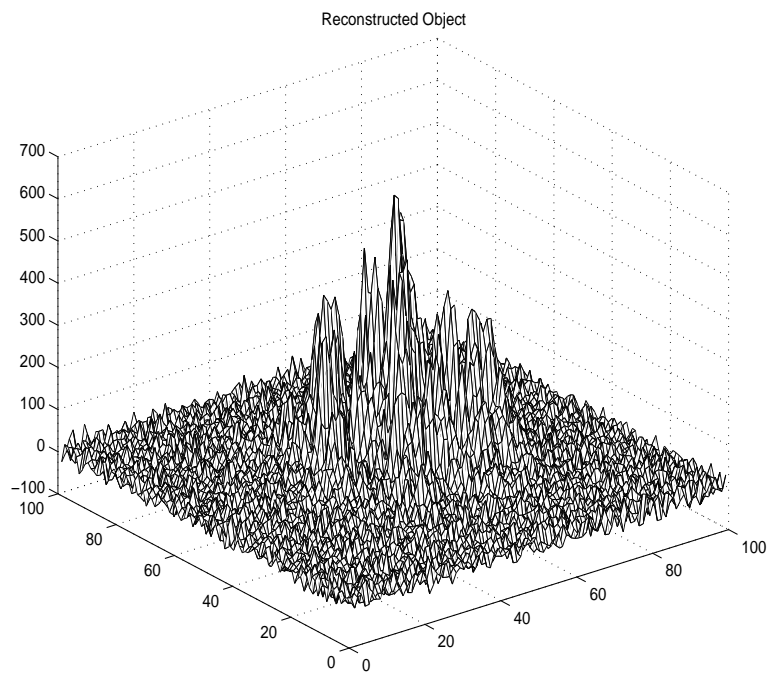


Figure 11. Reconstructed object generated using L-BFGS with standard identity Hessian initialization to minimize the full cost functional.

7. Acknowledgments

The authors wish to express their sincere appreciation to Dr. David Tyler of the Maui High Performance Computing Center and Professor Todd Torgersen of the Department of Computer Science at Wake Forest University. Dr. Tyler and his staff took the actual data observations and made this data available to us. Professor Torgersen provided us with software for preliminary data processing. Finally, we wish to acknowledge the support of the US Air Force Office of Scientific Research and the support and encouragement of the Program Officer for Scientific Computation, Major William Hilbun.

- [1] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, SIAM, 2000.
- [2] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, 1996.
- [3] R. A. Gonsalves, *Phase retrieval and diversity in adaptive optics*, *Opt. Eng.*, **21** (1982), pp. 829–832.
- [4] J. W. Goodman, *Introduction to Fourier Optics, Second Edition*, McGraw-Hill, 1996.
- [5] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, 1999.
- [6] R. Paxman, T. Schulz, and J. Fienup, *Joint estimation of object and aberrations by using phase diversity*, *J. Opt. Soc. Am. a*, **9** (1992), pp. 1072–1085.
- [7] M. Roggemann and B. Welsh, *Imaging Through Turbulence*, CRC Press, 1996.
- [8] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, 1996.
- [9] J. H. Seldin, M. F. Reiley, R. G. Paxman, B. E. Stribling, B. L. Ellerbroek, and D. C. Johnston, *Space object identification using phase-diverse speckle*, in *Mathematical Imaging*, Proceedings of SPIE **3170-01** (1997).
- [10] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, *SIAM Journal on Numerical Analysis*, **20** (1983), pp. 626–637.
- [11] C. R. Vogel, T. Chan, and R. Plemmons, *Fast Algorithms for Phase Diversity-Based Blind Deconvolution*, in *Adaptive Optical System Technologies*, SPIE Proceedings Vol. 3353 (1998).