Electronic Transactions on Numerical Analysis.
Volume 20, pp. 139-153, 2005.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

A NONNEGATIVELY CONSTRAINED TRUST REGION ALGORITHM FOR THE RESTORATION OF IMAGES WITH AN UNKNOWN BLUR *

JOHNATHAN M. BARDSLEY†

Abstract. We consider a large-scale optimization problem with nonnegativity constraints that arises in an application of phase diversity to astronomical imaging. We develop a cost function that incorporates information about the statistics of atmospheric turbulence, and we use Tikhonov regularization to induce stability. We introduce an efficient and easily implementable algorithm that intersperses gradient projection iterations with iterations from a well-known, unconstrained Newton/trust region method. Due to the large size of our problem and to the fact that our cost function is not convex, we approximately solve the trust region subproblem via the Steihaug-Toint truncated CG iteration. Iterations from the trust region algorithm are restricted to the inactive variables. We also present a highly effective preconditioner that dramatically speeds up the convergence of our algorithm. A numerical comparison using real data between our method and another standard large-scale, bound constrained optimization algorithm is presented.

Key words. constrained optimization, phase diversity, astronomical imaging

AMS subject classifications. 65K10, 65F22

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 the blurring of images of an astronomical object captured with a ground-based telescope [17]. The problem of interest is to reconstruct the object from these blurred images. The blurred, noisy data for such problems takes the form

$$\mathbf{d} = S\mathbf{f}_{\text{true}} + \boldsymbol{\eta}, \tag{1.1}$$

where S is a large, non-sparse, ill-conditioned matrix that characterizes the blur, $\mathbf{f}_{true}$ is the true image, or object, and is unknown, and $\boldsymbol{\eta}$ represents noise.

In the application of interest, the matrix S in (1.1) is also unknown, but can be accurately modelled in terms of the discrete phase, or wavefront aberration, $\boldsymbol{\phi}$, which quantifies the deviation from planarity of the wavefront. The problem of interest is to estimate $\mathbf{f}_{true}$ and the unknown phase $\boldsymbol{\phi}$ from the data $\mathbf{d}$. This problem is underdetermined since if $S[\boldsymbol{\phi}]$ is an $N \times N$ matrix, then both the object $\mathbf{f}_{true}$ and the phase $\boldsymbol{\phi}$ are $N \times 1$ vectors, and hence, we have N equations with $2N$ unknowns. Phase diversity [8, 16, 18] provides additional observable information in the form of auxiliary images produced by introducing known additional wavefront aberrations $\{\boldsymbol{\theta}_k\}$. The problem of interest then becomes the estimation of both $\mathbf{f}_{true}$ and $\boldsymbol{\phi}$ from data $\{\mathbf{d}_k\}$. Allowing, in addition, for data to be collected at various times t, we obtain the following system of equations for data formation:

$$\mathbf{d}_{k,t} = S[\boldsymbol{\phi}_t + \boldsymbol{\theta}_k]\mathbf{f}_{true} + \boldsymbol{\eta}_{k,t}, \qquad k = 1,\dots,K, \quad t = 1,\dots,T, \tag{1.2}$$

where $S[\boldsymbol{\phi}_t + \boldsymbol{\theta}_k]$ is an $N \times N$ matrix for each k and t, and $\boldsymbol{\phi}_t$, $\boldsymbol{\theta}_k$, $\boldsymbol{\eta}_{k,t}$, and $\mathbf{f}_{true}$ are $N \times 1$ vectors. Here $N = n_x \cdot n_y$, where n_x and n_y are the dimensions in numbers of pixels in the x and y directions respectively of the telescope's CCD detector array.

We will assume that the object is incoherent [9]. Then it represents an energy density, or photon density, and hence, is nonnegative. Reconstructing the object and unknown wavefront aberrations given equations (1.2) will then involve solving a nonnegatively constrained

*Received January 13, 2004. Accepted for publication May 9, 2005. Recommended by D. Calvetti.
†Department of Mathematical Sciences, The University of Montana, Missoula, MT 59812-0864. (bardsleyj@mso.umt.edu).

minimization problem of the form

$$\min_{(\vec{\phi},\mathbf{f})} J(\vec{\phi},\mathbf{f}), \qquad \text{subject to} \qquad \mathbf{f} \geq \mathbf{0}, \tag{1.3}$$

where $\vec{\phi} = (\phi_1, \ldots, \phi_T)$, and $\mathbf{f} \geq \mathbf{0}$ means that $f_i \geq 0$ for all i.

Problem (1.3) falls into the category of bound constrained optimization problems. Various algorithms have been developed for general, bound constrained optimization, the most basic of which is the projected gradient method [2, 5, 10]. The projected gradient method can be viewed as an extension of the method of steepest descent to constrained optimization problems. The gradient projection method has many of the same properties as the steepest descent method, including a slow convergence rate.

The projected Newton method [2, 10] is an extension of Newton's method to bound constrained optimization problems. This algorithm is straightforward to implement, and rapid local convergence is obtained when the exact projected Newton step is computed at each iteration. For large-scale problems such computations are typically not possible, and hence, approximations to the Newton step are necessary, in which case a deterioration in the convergence properties of the algorithm are typically noticed.

An extension of the quasi-Newton method LBFGS [13] to bound constrained optimization known as LBFGS-B is given in [4, 21]. The asymptotic convergence rate of LBFGS-B is the same as that of LBFGS. Unfortunately, the FORTRAN implementation of LBFGS discussed in [21] does not allow for preconditioning, i.e. a non-identity initial guess for the inverse Hessian. This severely limits the effectiveness of LBFGS-B for poorly conditioned optimization problems. For the application in this paper, an implementation of LBFGS-B via a MATLAB mex interface with the FORTRAN source code was not effective.

An algorithm that is well-suited for use on the application presented in this paper is the TRON algorithm of [11]. Public FORTRAN source code for this algorithm is available for use. Unfortunately, the function, gradient and Hessian evaluations and the preconditioner implementation for our problem are written in MATLAB and are complex. Consequently, we wanted to avoid having to rewrite these codes in FORTRAN. A MATLAB mex interface with the TRON source code is possible, but difficult. In particular, several mex interfaces are required. In addition, TRON does not allow for a free choice of the preconditioning matrix, and hence, a modification of the source code would be necessary. Finally, TRON is a complex algorithm that is difficult to implement independently (the author attempted this). The algorithm presented in this paper is, therefore, the result of the subsequent attempt to find an algorithm robust enough to solve (1.3) (all of the other algorithms that we had access to did not converge) that is also straightforward to implement.

The algorithm that we introduce in this paper is designed for solving poorly conditioned, large-scale, nonnegatively constrained optimization problems. It intersperses gradient projection iterations for active set identification, with iterations from an existing unconstrained optimization algorithm. Thus, provided that one has an implementation of the unconstrained algorithm, the implementation of this algorithm is very straightforward. We follow the general framework presented in [5]. Several highly effective algorithms have been developed which follow this frame work. For large-scale, bound constrained, quadratic optimization problems, Moré and Toraldo use conjugate gradient (CG) iterations in the unconstrained optimization stage. This algorithm, which we denote GPCG, can be viewed as an enhancement of the active set/CG methods of O'Leary [14, 15]; it improves efficiency by allowing the active set to change dramatically at each iteration. In [1], GPCG is extended for use on problems in which the cost function is convex. CG iterations are again used in the unconstrained optimization stage. This algorithm, which we will denote GPRNCG, can in turn be extended

ETNA
Kent State University
etna@mcs.kent.edu

to non-convex problems. Unfortunately, the resulting algorithm does not converge for the application in this paper, and hence, a more robust algorithm is needed for use in the unconstrained optimization stage. We use, in place of CG, the Newton-CG-trust region algorithm [6, 13] (NCGTR). NCGTR is an unconstrained optimization algorithm that uses the Steihaug-Toint truncated CG iteration to approximately solve the trust region subproblem at each outer NCGTR iteration.

This paper is organized as follows: In Section 2, the mathematical modelling of phase-diverse image formation, and the development of the optimization problem (1.3) are presented. Section 3 contains background material related to constrained optimization. Also in this section we present our computational algorithm and our preconditioner. In Section 4 we present results obtained from observed (not simulated) astronomical image data. Conclusions are presented in Section 5.

2. The Mathematical Model. We model short time exposure phase diversity atmospheric image data by

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

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,

$$(s \star f)(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(x - x', y - y')\, f(x', y')\, dx'\, dy'. \tag{2.2}$$

For details see [17, 8, 16, 18, 20]. The PSF takes the form

$$s[\phi] = C\, |\mathcal{F}^{-1}\{p e^{\imath\phi}\}|^2, \tag{2.3}$$

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,

$$\theta_k = c_k \times (x^2 + y^2), \quad k = 1, \ldots, K. \tag{2.4}$$

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.

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

$$J_{data}[\vec{\phi}, f] = \frac{1}{2TK} \sum_{t=1}^{T} \sum_{k=1}^{K} ||s[\phi_t + \theta_k] \star f - d_{k,t}||^2. \tag{2.5}$$

Here $\vec{\phi} = (\phi_1, \ldots, \phi_T)$ and $||\cdot||$ denotes the standard L^2 norm.

In practice, the data are pixelated images and are therefore discrete. In addition, we discretize the integral in (2.1)-(2.2), e.g., using midpoint quadrature, to obtain a discrete version of (2.1). With lexicographical ordering of unknowns, these discrete equations can in turn be rewritten as equation (1.2). The discrete version of (2.5) is then given by

$$J_{data}[\vec{\phi}, \mathbf{f}] = \frac{1}{2TK} \sum_{t=1}^{T} \sum_{k=1}^{K} ||S[\boldsymbol{\phi}_t + \boldsymbol{\theta}_k]\mathbf{f} - \mathbf{d}_{k,t}||^2. \tag{2.6}$$

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

$$J[\vec{\phi}, \mathbf{f}] = J_{data}[\vec{\phi}, \mathbf{f}] + \gamma J_{object}[\mathbf{f}] + \alpha J_{phase}[\vec{\phi}]. \tag{2.7}$$

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 modelled as a random process [17]. 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,

$$\Phi(\omega) = \frac{C_1}{(C_2 + |\omega|^2)^{11/6}}, \tag{2.8}$$

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

$$J_{phase}[\vec{\phi}] = \frac{1}{2T} \sum_{t=1}^{T} \boldsymbol{\phi}_t^T C \boldsymbol{\phi}_t, \tag{2.9}$$

where C is an $N \times N$ matrix with Fourier representation

$$C = F^* \Phi^{-1} F. \tag{2.10}$$

Here F and F^* are discrete Fourier and inverse Fourier transform matrices, respectively.

For regularization of the object, we take the "minimal information prior"

$$J_{object}[\mathbf{f}] = \frac{1}{2}||\mathbf{f}||^2. \tag{2.11}$$

The optimization problem we wish to solve is then given by

$$\min_{\mathbf{x} \in \Omega} J(\mathbf{x}), \tag{2.12}$$

where $\mathbf{x} = (\vec{\phi}, \mathbf{f})$, J is defined by (2.7), and

$$\Omega = \{(\vec{\phi}, \mathbf{f}) \mid , \mathbf{f} \geq \mathbf{0}\} \tag{2.13}$$

is known as the feasible set.

3. The Optimization Algorithm. In this section, we introduce an iterative scheme for solving (2.12). Each outer iteration of this scheme is comprised of two stages. The first stage consists of gradient projection iterations to identify the active set, while the second stage uses an efficient unconstrained minimization algorithm to compute a search direction in the inactive variables. A projected line search is then done in order to ensure sufficient decrease in function value, and to enforce the nonnegativity constraints. The effectiveness of this approach for large-scale bound-constrained minimization problems has been demonstrated for both quadratic and convex cost functions in [12] and [1] respectively.

3.1. Preliminaries. The projection of a vector $\mathbf{x} \in \mathbb{R}^{(T+1)N}$ onto the feasible set Ω is given by

$$\mathcal{P}_\Omega(\mathbf{x}) = \mathcal{P}_\Omega((\vec{\phi}, \mathbf{f})) \stackrel{\text{def}}{=} \arg\min_{\mathbf{v}\in\Omega} ||\mathbf{v} - (\vec{\phi}, \mathbf{f})|| = (\vec{\phi}, \max\{\mathbf{f}, \mathbf{0}\}),$$

where $\max\{\mathbf{f}, \mathbf{0}\}$ is the vector whose ith component is $\max\{f_i, 0\}$. The active set for a vector $\mathbf{x} \in \Omega$ is given by

$$\mathcal{A}(\mathbf{x}) = \{i \mid x_i = f_j \text{ for some } j, \text{ and } f_j = 0\}. \tag{3.1}$$

The complementary set of indices is called the inactive set and is denoted by $\mathcal{I}(\mathbf{x})$. The inactive, or free, variables consist of the components x_i for which the index i is in the inactive set. Note that, because there are no constraints on the values of phase vector components $[\phi_t]_j$, the corresponding indices of $\mathbf{x}$ will always be contained in $\mathcal{I}(\mathbf{x})$.

We assume that the cost functional J is a twice continuously differentiable function from $\mathbb{R}^{(T+1)N}$ to $\mathbb{R}$. The projected gradient of J at $\mathbf{x} \in \Omega$ is the $(T+1)N$-vector with components

$$[\text{grad}_\mathcal{P}\ J(\mathbf{x})]_i = \begin{cases} \frac{\partial J(\mathbf{x})}{\partial x_i}, & i \in \mathcal{I}(\mathbf{x}) \text{ or } (i \in \mathcal{A}(\mathbf{x}) \text{ and } \frac{\partial J(\mathbf{x})}{\partial x_i} < 0) \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

The reduced Hessian is given by

$$[\text{Hess}_R\ J(\mathbf{x})]_{ij} = \begin{cases} \frac{\partial^2 J(\mathbf{x})}{\partial x_i \partial x_j}, & \text{if } i \in \mathcal{I}(\mathbf{x}) \text{ or } j \in \mathcal{I}(\mathbf{x}) \\ \delta_{ij}, & \text{otherwise.} \end{cases}$$

Let $D_\mathcal{I}$ denote the diagonal matrix with components

$$[D_\mathcal{I}(\mathbf{x})]_{ii} = \begin{cases} 1, & i \in \mathcal{I}(\mathbf{x}) \\ 0, & i \in \mathcal{A}(\mathbf{x}). \end{cases} \tag{3.3}$$

Then

$$\text{Hess}_R\ J(\mathbf{x}) = D_\mathcal{I}(\mathbf{x})\ \text{Hess}\ J(\mathbf{x})\ D_\mathcal{I}(\mathbf{x})\ +\ D_\mathcal{A}(\mathbf{x}), \tag{3.4}$$

where $D_\mathcal{A}(\mathbf{x}) = I - D_\mathcal{I}(\mathbf{x})$.

3.2. The Gradient Projection Iteration. In principle, gradient projection generates a sequence of approximate minimizers $\{\mathbf{x}_k\} \subset \Omega$ via the following iteration:

$$\mathbf{p}_k = -\,\text{grad}\, J(\mathbf{x}_k) \tag{3.5}$$

$$\lambda_k = \arg\min_{\lambda>0} J(P_\Omega(\mathbf{x}_k + \lambda\mathbf{p}_k)) \tag{3.6}$$

$$\mathbf{x}_{k+1} = P_\Omega(\mathbf{x}_k + \lambda_k\mathbf{p}_k) \tag{3.7}$$

In practice, subproblem (3.6) is solved inexactly using a projected backtracking line search. We take the initial step length parameter to be the so-called Cauchy point,

$$\lambda_k^0 = \frac{||\mathbf{p}_k||^2}{\langle \text{Hess}\, J(\mathbf{x}_k)\mathbf{p}_k, \mathbf{p}_k\rangle}. \tag{3.8}$$

This is the minimizer of the quadratic $\hat{q}(\lambda) = q(\mathbf{x}_k + \lambda \mathbf{p}_k)$, where

$$q(\mathbf{x}+\mathbf{s}) = J(\mathbf{x}) + \langle \text{grad}\, J(\mathbf{x}), \mathbf{s}\rangle + \frac{1}{2}\langle \text{Hess}\, J(\mathbf{x})\mathbf{s}, \mathbf{s}\rangle. \tag{3.9}$$

As in [2], step length reduction can be accomplished by taking $\lambda_k^m = \beta^m \lambda_k^0$, $m = 0, 1, \ldots$ for some $\beta \in (0, 1)$. A quadratic interpolation scheme as found in [12] can also be used. We stop at the first m for which the sufficient decrease condition

$$J(\mathbf{x}_k(\lambda_k^m)) \leq J(\mathbf{x}_k) - \frac{\mu}{\lambda_k^m}||\mathbf{x}_k - \mathbf{x}_k(\lambda_k^m)||^2 \tag{3.10}$$

holds, where $\mu \in (0, 1)$ and

$$\mathbf{x}_k(\lambda) = \mathcal{P}_\Omega(\mathbf{x}_k + \lambda \mathbf{p}_k). \tag{3.11}$$

Some important convergence properties of the gradient projection iteration are given in the following theorem.

THEOREM 3.1. *Let $\{\mathbf{x}_k\}$ be a sequence generated by the gradient projection iteration as discussed above. If $\overline{\mathbf{x}}$ is a limit point of $\{\mathbf{x}_k\}$, then grad$_\mathcal{P}$ $J(\overline{\mathbf{x}}) = \mathbf{0}$. If $\{\mathbf{x}_k\}$ converges to $\overline{\mathbf{x}}$, and $\overline{\mathbf{x}}$ is nondegenerate, i.e., $[\text{grad}\, J(\overline{\mathbf{x}})]_i > 0$ for all $i \in \mathcal{A}(\overline{\mathbf{x}})$, then $\mathcal{A}(\overline{\mathbf{x}})$ is identified in finitely many iterations. More precisely, there exists an integer m_0 such that for all $k \geq m_0$, $\mathcal{A}(\mathbf{x}_k) = \mathcal{A}(\overline{\mathbf{x}})$.*

The first half of the theorem is proved in [2]. The second half is proved in [5].

The asymptotic convergence rate for the gradient projection iteration is linear, and hence, the algorithm is typically very slow to converge.

3.3. The Reduced Newton-CG-Trust Region Method. A standard choice for improving the convergence rate of the gradient projection iteration is to implement, instead, the projected Newton iteration [3, 10]. This involves replacing $\mathbf{p}_k$ in (3.5) with the solution to

$$\text{Hess}_R\ J(\mathbf{x}_k)\ \mathbf{p} = -\,\text{grad}\, J(\mathbf{x}_k). \tag{3.12}$$

With a properly implemented line search, this yields a quadratically convergent scheme [3, 10]. However, it lacks robustness if the reduced Hessian is ill-conditioned and system (3.12) is solved inexactly. For this reason, we advocate interspersing gradient projection iterations with iterations from a robust unconstrained minimization algorithm, in which we (approximately) solve

$$\min_{\mathbf{p}\in\mathcal{F}(\mathbf{x}_k)} J(\mathbf{x}_k + \mathbf{p}) \tag{3.13}$$

at each outer iteration. Here

$$\mathcal{F}(\mathbf{x}_k) = \{\mathbf{x} \in \mathbb{R}^{(T+1)N} \mid x_i = 0 \text{ whenever } x_{k,i} = 0\}.$$

Problem (3.13) is an unconstrained minimization problem in the free variables. In [12], the case where J is quadratic is discussed, and CG is used at each outer iteration to approximately

ETNA
Kent State University
etna@mcs.kent.edu

solve (3.13). In [1], J is a convex function, and CG again proves to be an effective algorithm for approximately solving (3.13). The approach in [1] can be extended to nonconvex bound constrained minimization, but the resulting algorithm is not effective for (2.12),(2.7) and hence a more robust choice of unconstrained algorithms for solving (3.13) is required.

One such algorithm is Newton-CG-trust region (NCGTR). NCGTR is a trust region method that uses the Steihaug-Toint truncated CG algorithm to solve the trust region subproblem

$$\min_s m_\nu(\mathbf{s}) \quad \text{subject to} \quad ||\mathbf{s}||_\nu \leq \Delta_\nu \tag{3.14}$$

at each inner iteration ν. In the setting of (3.13), the quadratic function m_ν must be restricted to $\mathcal{F}(\mathbf{x}_k)$. Therefore, we take

$$m_\nu(\mathbf{s}) = J(\mathbf{x}_k + \mathbf{p}_\nu) + \mathbf{s}^T \mathbf{g}_\nu + \frac{1}{2}\mathbf{s}^T H_\nu \mathbf{s}, \tag{3.15}$$

where

$$\mathbf{g}_\nu = D_{\mathcal{I}}(\mathbf{x}_k)\ \operatorname{grad} J(\mathbf{x}_k + \mathbf{p}_\nu), \tag{3.16}$$

$$H_\nu = D_{\mathcal{I}}(\mathbf{x}_k)\ \text{Hess}\, J(\mathbf{x}_k + \mathbf{p}_\nu)\ D_{\mathcal{I}}(\mathbf{x}_k) + D_{\mathcal{A}}(\mathbf{x}_k). \tag{3.17}$$

The matrix norm $||\cdot||_\nu$ in (3.14) depends on the Newton iteration ν, and Δ_ν is a positive parameter called the trust region radius.

Problem (3.15)-(3.14) 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 for subproblem (3.14) is the Steihaug-Toint algorithm. The key idea is to apply CG iterations to minimize $m_\nu(\mathbf{s})$, or equivalently, to solve the linear system

$$H_\nu \mathbf{s} = -\mathbf{g}_\nu. \tag{3.18}$$

(Notice the similarity between (3.18) and (3.12).) If the initial guess $\mathbf{s}_{\nu,0}$ is zero and H_ν is SPD, the CG iterates $\mathbf{s}_{\nu,j}$ monotonically increase in norm and monotonically decrease $m_\nu(\mathbf{s})$ [19]. 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 $\mathbf{d}_{\nu,j}^T H_\nu \mathbf{d}_{\nu,j} \leq 0$, where $\mathbf{d}_{\nu,j}$ 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 be incorporated to accelerate CG convergence. In this case, the trust region is determined by the vector norm induced by the SPD preconditioning matrix M_ν,

$$||\mathbf{s}||_{M_\nu} \stackrel{\text{def}}{=} \sqrt{\mathbf{s}^T M_\nu \mathbf{s}}. \tag{3.19}$$

Given gradient $\mathbf{g} = \mathbf{g}_\nu$, Hessian $H = H_\nu$, preconditioning matrix $M = M_\nu$ and trust region radius $\Delta = \Delta_\nu$, the Steihaug-Toint algorithm for approximately solving the trust region subproblem (3.15)-(3.14) is given by

The Steihaug-Toint Truncated CG Algorithm:
$\mathbf{s}_0 := \mathbf{0}$;
$\mathbf{r}_0 := \mathbf{g}$; % Initial CG residual
$\mathbf{y}_0 := M^{-1}\mathbf{r}_0$; % Apply preconditioner
$\mathbf{d}_0 := -\mathbf{y}_0$; % Initial CG search direction

$j := 0$;
for j=0,1,2,...
  $\kappa_j := \mathbf{d}_j^T H \mathbf{d}_j$;
  If $\kappa_j \leq 0$, *% Check for indefinite H*
    $\tau_j :=$ positive root of $||\mathbf{s}_j + \tau \mathbf{d}_j||_M = \Delta$;
    $\mathbf{s}_{j+1} := \mathbf{s}_j + \tau_j \mathbf{d}_j$; *% Move to boundary*
    Stop.
  End if
  $\alpha_j := \mathbf{r}_j^T \mathbf{y}_j / \kappa_j$;
  If $||\mathbf{s}_j + \alpha_j \mathbf{d}_j||_M \geq \Delta$, *% Boundary crossed*
    $\tau_j :=$ positive root of $||\mathbf{s}_j + \tau \mathbf{d}_j||_M = \Delta$;
    $\mathbf{s}_{j+1} := \mathbf{s}_j + \tau_j \mathbf{d}_j$; *% Backtrack to boundary*
    Stop.
  End if
  $\mathbf{s}_{j+1} := \mathbf{s}_j + \alpha_j \mathbf{d}_j$; *% Update solution*
  $\mathbf{r}_{j+1} := \mathbf{r}_j + \alpha_j H \mathbf{d}_j$; *% Update CG residual*
  $\mathbf{y}_{j+1} := M^{-1} \mathbf{r}_{j+1}$; *% Apply preconditioner*
  $\beta_{j+1} := \mathbf{r}_{j+1}^T \mathbf{y}_{j+1} / \mathbf{r}_j^T \mathbf{y}_j$;
  $\mathbf{d}_{j+1} := -\mathbf{r}_{j+1} + \beta_{j+1} \mathbf{d}_j$; *% Update CG search direction*
  $j := j + 1$;
end (for) CG iterations

Once the Steihaug-Toint algorithm outputs an approximate solution $\mathbf{s}_\nu$ of (3.14)-(3.17), two tests are performed. The first test determines the new iterate $\mathbf{p}_{\nu+1}$, while the second determines the new trust region radius $\Delta_{\nu+1}$. Both tests require the following parameter:

$$\rho_\nu = \frac{J(\mathbf{x}_k + \mathbf{p}_\nu) - J(\mathbf{x}_k + \mathbf{p}_\nu + \mathbf{s}_\nu)}{m_\nu(\mathbf{0}) - m_\nu(\mathbf{s}_\nu)}. \tag{3.20}$$

The reduced Newton-CG-Trust Region (RNCGTR) algorithm is then given by

The RNCGTR Algorithm:
Given $\Delta_{max} > 0$, $\Delta_0 \in (0, \Delta_{max})$, $\eta_1 \in [0, \frac{1}{4})$, $\eta_1 < \eta_2 < 1$, and $\alpha_1 < 1 < \alpha_2$:
for $k = 0, 1, 2, \ldots$
  Obtain $\mathbf{s}_\nu$ by (approximately) solving (3.14)-(3.17)
   via the Steihaug-Toint Algorithm;
  Evaluate ρ_ν from (3.20);
  if $\rho_\nu > \eta_1$
    $\mathbf{p}_{\nu+1} = \mathbf{p}_\nu + \mathbf{s}_\nu$,
  else
    $\mathbf{p}_{\nu+1} = \mathbf{p}_\nu$.
  if $\rho_\nu > \eta_2$
    $\Delta_{\nu+1} = \min(\alpha_1 \cdot ||\mathbf{s}_\nu||, \Delta_{max})$,
  else if $\rho_\nu > \eta_1$
    $\Delta_{\nu+1} = \Delta_\nu$,
  else
    $\Delta_{\nu+1} = \alpha_2 \cdot ||\mathbf{s}_\nu||$.
end (for)

3.3.1. 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 $M^{-1/2}$. (See the Steihaug-Toint algorithm above.) M is called the preconditioner and the resulting iteration is called the preconditioned conjugate gradient (PCG) algorithm [13]. 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 function J in (2.7). This Hessian takes the form

$$H_{full}[\vec{\phi}, f] = H_{data}[\vec{\phi}, f] + H_{reg}, \tag{3.21}$$

where, with the regularization functionals (2.11) and (2.9), the Hessian of the regularization terms is a $(T+1)\times(T+1)$ block diagonal matrix,

$$H_{reg} = \begin{bmatrix} \frac{\alpha}{T}C & 0 & \cdots & 0 & 0 \\ 0 & \frac{\alpha}{T}C & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & \frac{\alpha}{T}C & 0 \\ 0 & 0 & \cdots & 0 & \gamma I \end{bmatrix}, \tag{3.22}$$

where C is given by (2.10). During the Steihaug-Toint truncated CG iterations, we will use the preconditioner $M_\nu = H_{reg}$. Because γI is the preconditioner for the component of the Hessian related to the object, difficulties that arise because of the nonnegativity constraints are avoided.

3.4. The Numerical Algorithm. In the first stage of our algorithm we need stopping criteria for the gradient projection iterations $\{\mathbf{x}_{k,j}\}$. Borrowing from Moré and Toraldo [12], we stop as soon as either

$$J(\mathbf{x}_{k,j-1}) - J(\mathbf{x}_{k,j}) \le \gamma_1 \cdot \max\{J(\mathbf{x}_{k,i-1}) - J(\mathbf{x}_{k,i}) \,|\, i = 1, \dots, k-1\}, \tag{3.23}$$

where $0 < \gamma_1 < 1$, or

$$\mathcal{A}(\mathbf{x}_{k,j}) = \mathcal{A}(\mathbf{x}_{k,j-1}). \tag{3.24}$$

In the second stage, for the RNCGTR iterations, we use the stopping criteria

$$J_k(\mathbf{p}_{\nu-1}) - J_k(\mathbf{p}_\nu) \le \gamma_2 \cdot \max\{J_k(\mathbf{p}_{i-1}) - J_k(\mathbf{p}_i) \,|\, i = 1, \dots, j-1\}, \tag{3.25}$$

where $0 < \gamma_2 < 1$, and $J_k(\mathbf{p}) = J(\mathbf{x}_k + \mathbf{p})$.

Once an approximate solution $\mathbf{p}_k$ of (3.13) is obtained using RNCGTR with stopping criteria (3.25), we apply a projected backtracking line search (see Section 3.2) in order to enforce the nonnegativity constraints, and to ensure sufficient decrease in function value. The stopping criteria that we use for the line search is much less stringent than (3.10), and is given by

$$J(\mathbf{x}_k(\lambda_k^m)) < J(\mathbf{x}_k). \tag{3.26}$$

We will denote the gradient projection-reduced Newton-CG-trust region algorithm by GPRNCGTR.

ETNA
Kent State University
etna@mcs.kent.edu

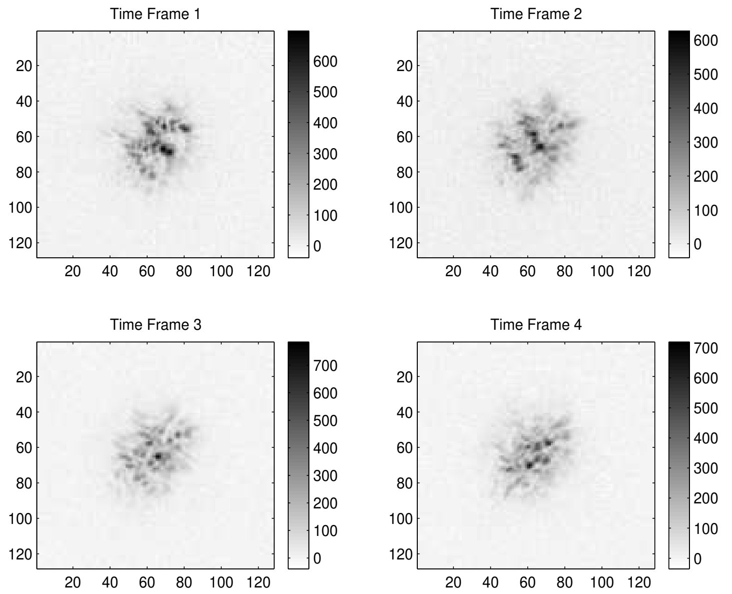

FIG. 4.1. *In-focus (Channel 1) phase diversity image data.*

The GPRNCGTR Algorithm:
Select initial guess $\mathbf{x}_0$, initialize RNCGTR parameters, and set $k = 0$.
Stage 1: Given $\mathbf{x}_k$.
(1) Take gradient projection steps until either (3.23) or (3.24) is satisfied, and return updated $\mathbf{x}_k$.
Stage 2: Given $\mathbf{x}_k$ and Δ_k.
(1) Do RNCGTR iterations on (3.13) until (3.25) is satisfied, and return $\mathbf{p}_k$ and Δ_k.
(2) Find λ_k^m which satisfies (3.26), and return $\mathbf{x}_{k+1} = \mathbf{x}_k(\lambda_k^m)$ (see (3.11)).
(3) Update $k := k + 1$ and return to Step 1.

Remark: The results of Theorem 3.1 will hold for the iterates $\{\mathbf{x}_k\}$ generated by GPRNCGTR, since at each outer iteration at least one gradient projection iteration is taken, and (3.26) holds in Step 2.

4. Numerical Results. In this section we present results obtained when the GPRNCGTR algorithm was 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 (2.4)) is shown in Figure 4.1. Figure 4.2 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 Figure 4.3. The binary star can clearly be seen in the reconstruction. The corresponding wavefront aberrations are shown in Figure 4.4. The annular shape of the telescope's aperture accounts for the presentation of the phase reconstructions in an annular region.

In Figure 4.5 we present a comparison between three algorithms: GPRNCGTR with

ETNA
Kent State University
etna@mcs.kent.edu

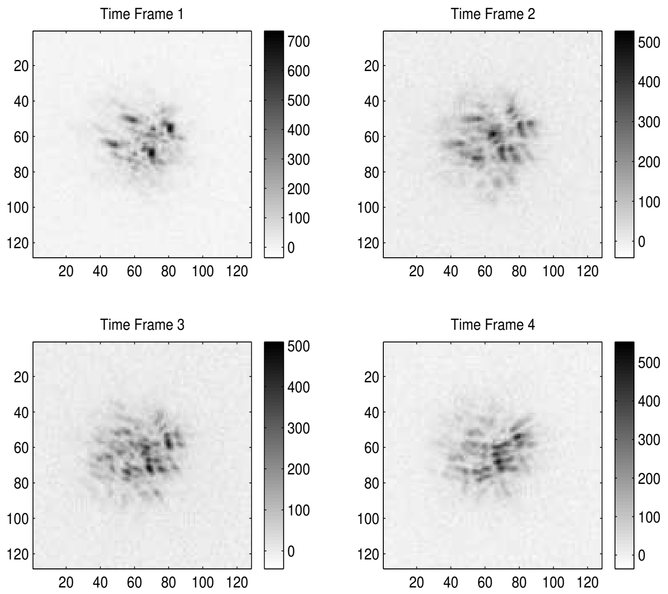

FIG. 4.2. *Out-of-focus (Channel 2) phase diversity image data.*

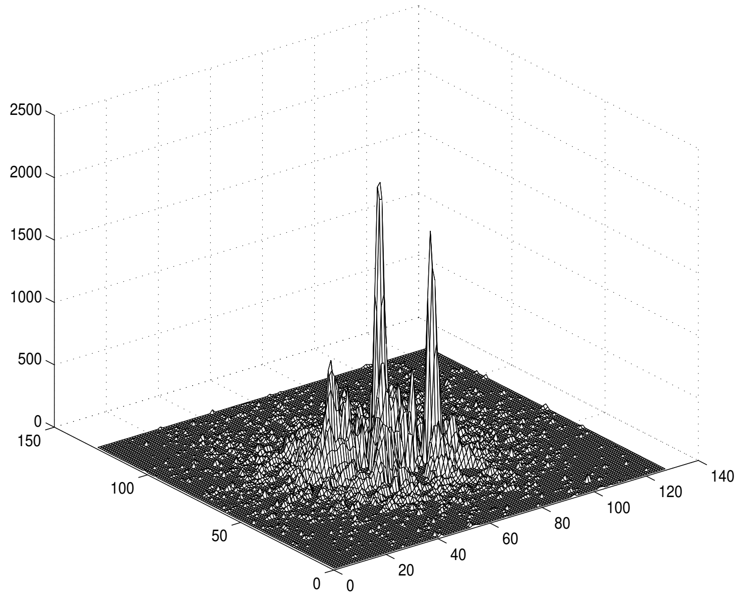

FIG. 4.3. *Reconstruction of the object, a binary star.*

preconditioning, GPRNCGTR without preconditioning, and the projected Newton method of Bertsekas [3] with preconditioning. Each method is used to solve (2.12). The source code for these algorithms was written in MATLAB. In this comparison, it is our objective to show both the effectiveness of the preconditioner and the robustness of the GPRNCGTR algorithm.

In each case, the regularization parameters in (2.7) were taken to be $\alpha = 5 \times 10^{-1}$ and $\gamma = 5 \times 10^{-3}$. It should be noted that numerical performance, as well as the actual object

ETNA
Kent State University
etna@mcs.kent.edu

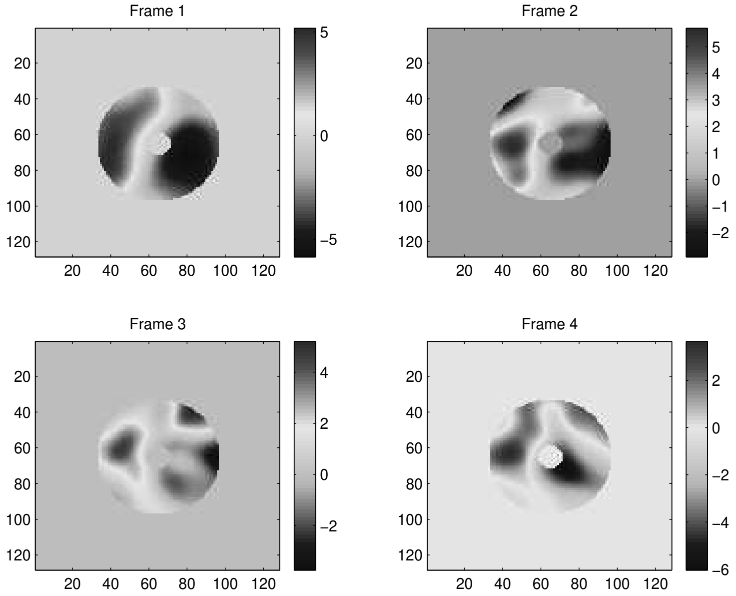

FIG. 4.4. *Reconstructed phases, or wavefront aberrations.*

and phase reconstructions, varies with the choice of regularization parameters.

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 guess may also influence the reconstructions. The initial guess may also influence the numerical performance. For each method, the phase vectors are initialized to zero ($\boldsymbol{\phi}_t = \mathbf{0}$ for $t = 1, 2, 3, 4$), and the initial object is generated by deconvolving the data using the corresponding PSF's.

To obtain a rough comparison of convergence behavior as a function of computational cost for the various methods, we plot the norm of the projected gradient (3.2) versus the cumulative number of two-dimensional fast Fourier transforms (FFTs). The computational cost of function and gradient evaluations, and Hessian matrix-vector multiplications is dominated by FFTs. Hence FFTs constitute a significant cost in our implementations.

The parameters required for RNCGTR iterations are taken to be $\eta_1 = 0.05$, $\eta_2 = 0.9$, $\alpha_1 = 2.5$, $\alpha_2 = 0.25$, $\Delta_{max} = 1 \times 10^7$, and

$$\Delta_0 = \frac{||\,\mathrm{grad}\, J(\mathbf{x}_0)||^3}{|\,\mathrm{grad}\, J(\mathbf{x}_0)^T \mathrm{Hess}\, J(\mathbf{x}_0)\,\mathrm{grad}\, J(\mathbf{x}_0)|}.$$

The stopping parameter for gradient projection iterations (see (3.23)) is given by $\gamma_1 = 0.25$, while the stopping parameter for the RNCGTR iterations (see (3.25)) is given by $\gamma_2 = 1$.

The two implementations of the GPRNCGTR algorithm (preconditioned and unpreconditioned) used in this comparison contain stopping criteria not discussed in the previous section. Both restrict the number of Steihaug-Toint truncated CG iterations allowed, as this is the most expensive part of the GPRNCGTR algorithm, and large numbers of CG iterations are not beneficial until the algorithm is near to convergence.

First, we stop CG iterations once

$$||\mathbf{r}_j|| \le \min(\beta, \sqrt{||\mathbf{r}_0||}) \cdot ||\mathbf{r}_0||, \tag{4.1}$$

where $\{\mathbf{r}_j\}$ are defined in the Steihaug-Toint algorithm above. For the comparisons in Figure 4.5 we take $\beta = 0.01$.

ETNA
Kent State University
etna@mcs.kent.edu

Secondly, we set a maximum CG_{max} on the number of CG iterations allowed during each application of the Steihaug-Toint algorithm. In the implementations of GPRNCGTR used for this comparison, $\mathrm{CG}_{max} = 30$. This accounts for the linear convergence rate exhibited in Figure 4.5 by preconditioned GPRNCGTR in the later iterations. In other numerical experiments, we solved the minimization problem with various values of CG_{max} and found that as CG_{max} increases, the convergence profile tends toward what appears to be a superlinear convergence rate. Unfortunately, the number of FFTs also increases dramatically, as large numbers of CG iterations are taken in early iterations with little, if any, benefit. The choice of $\mathrm{CG}_{max} = 30$ seems to balance rapid convergence with the computational cost of implementation.

Our implementation of the projected Newton algorithm uses a truncated CG iteration to approximately solve (3.12). We stop CG iterations as soon as (i) stopping criteria (4.1) is satisfied, (ii) the number of CG iteration reaches $\mathrm{CG}_{max} = 30$, or (iii) negative curvature is detected in the CG iteration, in which case the previous CG iterate is taken as the approximate solution to (3.12). (Note that each of these stopping criteria is also used within GPRNCGTR.) The choice of $\mathrm{CG}_{max} = 30$ seems, once again, to balance rapid convergence with the computational cost of implementation. We also use (3.22) as a preconditioner.

We make two important observations based on the results of Figure 4.5. First, we note that the preconditioned GPRNCGTR algorithm has converged, and hence, we have achieved our goal of finding an algorithm robust enough to solve (2.12) that is also straightforward to implement. Notice also that, since the projected Newton algorithm with preconditioning did not converge, the algorithmic approach taken in GPRNCGTR is clearly more robust than that of the projected Newton algorithm. Secondly, the effectiveness of preconditioner (3.22) is evident when the convergence of the preconditioned GPRNCGTR is compared with that of GPRNCGTR without preconditioning. It is noted, however, that this preconditioner ignores the H_{data} term in (3.21). This suggests that a more effective preconditioner for this problem is possible. This is an area of current investigation.

5. Conclusions. We present an algorithm for solving large-scale nonnegatively constrained minimization problems. This algorithm, which we denote GPRNCGTR, is comprised of two stages. The first stage consists of gradient projection iterations. The second stage consists of iterations from an unconstrained trust region algorithm applied to the problem of minimizing the cost function restricted to the free variables. The trust region method used approximately solves the (reduced) trust region subproblem using the Steihaug-Toint truncated CG algorithm.

We compare the performance of the preconditioned GPRNCGTR, the unpreconditioned GPRNCGTR, and a preconditioned version of the projected Newton algorithm of Bertsekas [3]. For this test problem, preconditioned GPRNCGTR is the most efficient in terms of FFTs and, in fact, is the only algorithm that converges. The superior performance of preconditioned GPRNCGTR over the preconditioned projected Newton algorithm suggests that the algorithmic approach taken in GPRNCGTR is more robust than that of the projected Newton algorithm. In addition, the ease of implementing GPRNCGTR as compared to the TRON algorithm of [11] suggests that in certain situations, GPRNCGTR may be a viable alternative to TRON, though a direct comparison between GPRNCGTR and TRON is needed.

A preconditioner is presented in the hopes that improved convergence properties of the inner CG iterations can be obtained. A dramatic improvement in the convergence properties of the GPRNCGTR algorithm is noted when preconditioning is used. When preconditioning is not used, robustness is lost and computational cost increases dramatically.

Acknowledgments. A special thanks to Dr. David Tyler of the Maui High Performance Computing Center and Professor Todd Torgersen of the Department of Computer Science

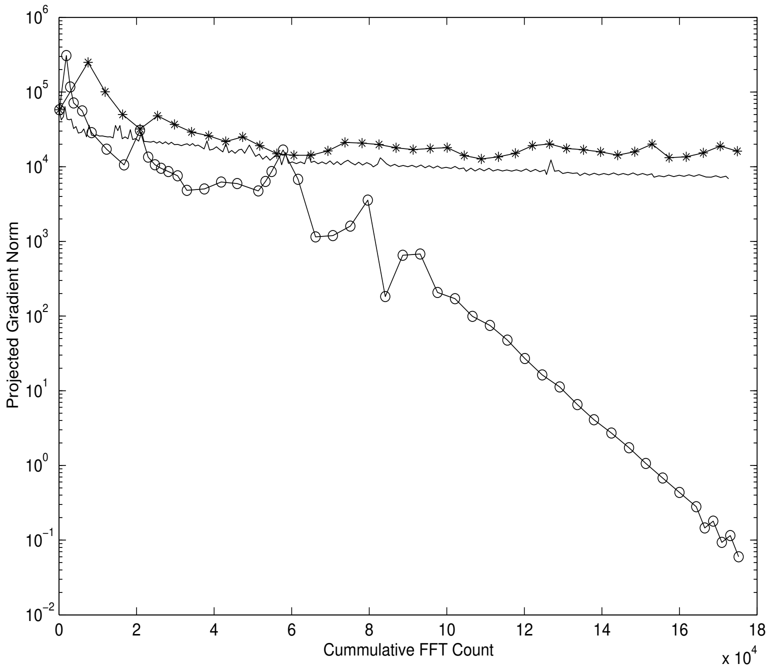

FIG. 4.5. *Performance of Methods for Solving* (2.12). *The horizontal axis represents cumulative FFTs. The vertical axis shows the norm of the projected gradient on a logarithmic scale. The line connected by asterisks (top graph) denotes GPRNCGTR without preconditioning. The solid line (second graph from top) denotes the projected Newton method with preconditioning. The line connected by circles (bottom graph) denotes GPRNCGTR with preconditioning.*

at Wake Forest University. Dr. Tyler and his staff took the actual data observations and made this data available. Professor Torgersen provided us with software for preliminary data processing.

Thanks is also due to Dr. Luc Gilles of Michigan Tech and Dr. Curt Vogel of Montana State University. Dr. Gilles wrote the code for the exact Hessian vector multiplication computations. Dr. Vogel, in addition to introducing me to this problem, wrote the code for the function and gradient evaluations.

REFERENCES

[1] J. M. BARDSLEY AND C. R. VOGEL, *A Nonnnegatively constrained convex programming method for image reconstruction*, SIAM J. Sci. Comput., 25 (2003), pp. 1326–1343.
[2] D. P. BERTSEKAS, *On the Goldstein-Levitin-Poljak gradient projection method*, IEEE Trans. Automat. Control, 21 (1976), pp. 174–184.
[3] D. P. BERTSEKAS, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.
[4] R. H. BYRD, P. LU AND J. NOCEDAL, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190–1208.
[5] P. H. CALAMAI AND J. J. MORÉ, *Projected gradient methods for linearly constrained problems*, Math. Program., 39 (1987), pp. 93–116.
[6] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Trust Region Methods*, SIAM, Philadelphia, 2000.
[7] L. GILLES, C. R. VOGEL, AND J. M. BARDSLEY, *Computational Methods for a Large-Scale Inverse Problem Arising in Atmospheric Optics*, Inverse Problems, 18 (2002), pp. 237-252.
[8] R. A. GONSALVES, *Phase retrieval and diversity in adaptive optics*, Optim. Eng., 21 (1982), pp. 829–832.
[9] J. W. GOODMAN, *Statistical Optics*, Wiley, New York, 1985.

[10] C. T. KELLEY, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.
[11] C.-J. LIN AND J. J. MORÉ, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.
[12] J. J. MORÉ AND G. TORALDO, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.
[13] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer-Verlag, New York, 1999.
[14] D. P. O'LEARY, *Conjugate gradient algorithms in the solution of optimization problems for nonlinear elliptic partial differential equations*, Computing, 22 (1979), pp. 59–77.
[15] D. P. O'LEARY, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.
[16] *Joint estimation of object and aberrations by using phase diversity*, J. Opt. Soc. Am. A, 9 (1992), pp. 1072–1085.
[17] M. ROGGEMANN AND B. WELSH, *Imaging Through Turbulence*, CRC Press, Boca Raton, 1996.
[18] 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, Proc.SPIE, vol. 3170-01, 1997.
[19] T. STEIHAUG, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.
[20] C. R. VOGEL, T. CHAN, AND R. PLEMMONS, *Fast algorithms for phase diversity-based blind deconvolution*, in Adaptive Optical System Technologies, SPIE Proc., vol. 3353, 1998.
[21] C. ZHU, R. H. BYRD AND J. NOCEDAL, *L-BFGS-B: FORTRAN subroutines for large scale bound constrained optimization*, ACM Trans. Math. Software, 23 (1997), pp. 550–560.