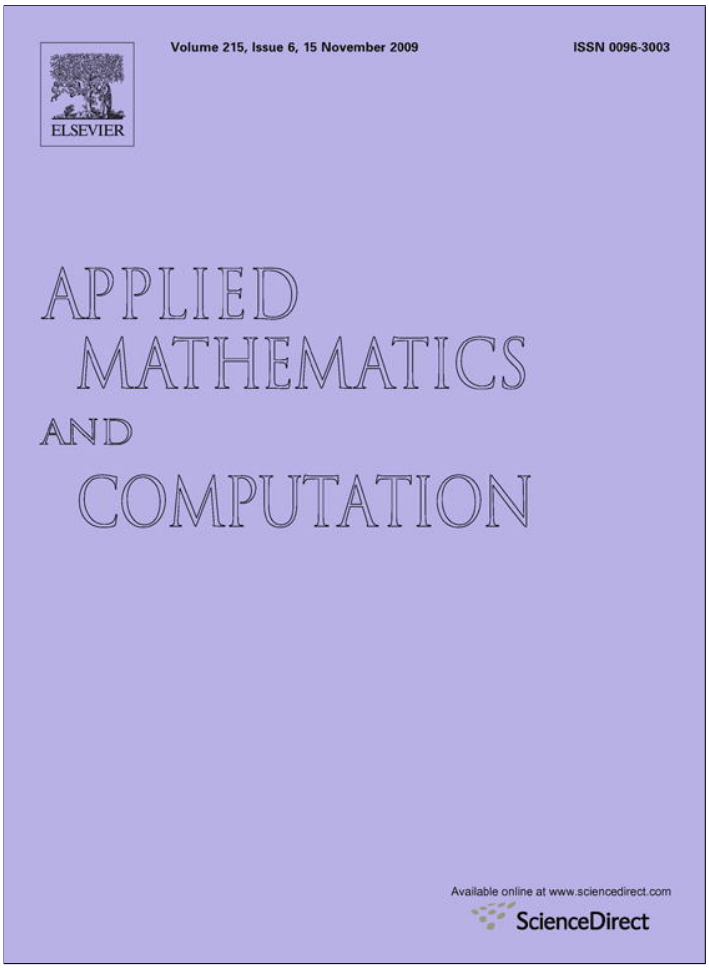

Applied Mathematics and Computation 215 (2009) 2359–2377

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A fast subspace method for image deblurring

Yanfei Wang [a], Shiqian Ma [b,*]

[a] Key Laboratory of Petroleum Geophysics, Institute of Geology and Geophysics, Chinese Academy of Sciences, P.O. Box 9825, Beijing 100029, PR China
[b] Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027-6902, USA

ARTICLE INFO

Keywords:
Image deblurring
Subspace
Interior-point method
Trust region

ABSTRACT

Image deblurring problems appear frequently in astronomical image analysis. For image deblurring problems, it is reasonable to add a non-negativity constraint because of the physical meaning of the image. Previous research works are mainly full-space methods, i.e., solving a regularized optimization problem in a full space. To solve the problem more efficiently, we propose a subspace method. We first formulate the problem from full space to subspace and then use an interior-point trust-region method to solve it. The numerical experiments show that this method is suitable for ill-posed image deblurring problems.

1. Introduction

Astronomical images obtained are usually corrupted or distorted by blurring and noise [1,2]. The image may be degraded by sensor noise, misfocus of CCD camera, non-uniform motion, atmospheric aerosols and random atmospheric turbulence. A key problem in image deblurring is to restore the image by solving a blurring model and removing noise. It indicates that the power distribution in the image plane due to a point source in the object plane can be formulated as

$$h_n(i,j) = h(i,j) + n(i,j) = k(i,j)\bigstar f(i,j) + n(i,j), \tag{1}$$

where, $h(i,j)$ denotes the recorded blurred image, $f(i,j)$ denotes the original object. Their unique relation in the spatial domain is given by the two-dimensional point spread function (PSF) $k(i,j)$. $\bigstar$ is the convolution operator, i and j are the spatial coordinates and $n(i,j)$ denotes an additional noise term. The above expression is commonly modeled as a first kind integral equation of the form

$$h_n(x,y) = h(x,y) + n(x,y) = \int\int_{\mathbb{R}^2} k(x-\xi, y-\eta) f(\xi,\eta) d\xi d\eta + n(x,y) = Kf(x,y) + n(x,y). \tag{2}$$

The image restoration problem is recovering f from h and the kernel k.

In digital image restoration, a discrete model of (2) is usually formulated by the linear system

$$\mathbf{h_n} = \mathbf{h} + \mathbf{n} = \mathscr{K}\mathbf{f} + \mathbf{n}, \tag{3}$$

where $\mathscr{K} \in \mathbb{R}^{N^2 \times N^2}$, $\mathbf{f}, \mathbf{h}, \mathbf{n} \in \mathbb{R}^{N^2}$. But we cannot expect to solve this linear system by algebraic strategy easily, because the matrix $\mathscr{K}$ is usually badly conditioned. In addition, the noise cannot be ignored when the observed image $\mathbf{h_n}$ is recorded.

Due to the ill-posedness of (2) and the discrete ill-posedness of (3) regularization is needed in order for the establishment of well-posedness to be satisfied. For us, this will require a variational formulation of the problem. The most common example of a variational formulation of (2) or (3) is the least squares problem

* Corresponding author.
E-mail addresses: yfwang@mail.iggcas.ac.an (Y. Wang), sm2756@columbia.edu (S. Ma).

0096-3003/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2009.08.033

$$\arg\min_f \Psi(f) := \frac{1}{2}\|Kf - h_n\|^2 \tag{4}$$

or

$$\arg\min_{\mathbf{f}} \Psi(\mathbf{f}) := \frac{1}{2}\|\mathscr{K}\mathbf{f} - \mathbf{h_n}\|^2. \tag{5}$$

But because of the compactness of the operator, this formulation is also a (discrete) ill-posed problem. Regularization techniques for least squares problem have been extensively studied [3,4]. For our problem, we only consider the discrete least squares problem (5). The standard Tikhonov regularization form is given by

$$\min \|\mathscr{K}\mathbf{f} - \mathbf{h_n}\|^2 + \alpha\Gamma(\mathbf{f}), \tag{6}$$

where $\Gamma(\cdot)$ is a function whose role is to give some penalization to the unknown $\mathbf{f}$. There are a lot of tricks for choosing $\Gamma(\cdot)$ (see [5,6] for smooth regularization, [7–9] for nonsmooth regularization and their applications and [10,11] for sparse regularization and their applications). For example, $\Gamma(\mathbf{f})$ can be defined as $(L\mathbf{f}, \mathbf{f})$, where L is a scale operator which can be chosen as a positive definite or positive semi-definite matrix. Usually L is chosen as a sparse matrix for changing little of the spectrum of the original kernel. In (6), $\alpha > 0$ is the so-called regularization parameter which plays a major role in regularizing the ill-posedness. The value of α is positive and will be typically small. But the choice of an appropriate α is a difficult task, which is usually related to the spectrum of the discrete kernel $\mathscr{K}$ and the unknown noise level in $\mathbf{h}$ [5]. Methods for solving ill-posed image deblurring problems include SVD-based (singular value decomposition) direct methods [2], Ricardson-Lucy (EM) approach and a generalization [12–14], projected Landweber iterative method [1], simple least squares method [15] and total least squares method [16], Newton's and quasi-Newton's methods [4], gradient methods and various preconditioning technique and different applications [17,4,18]. The conjugate gradient method has proved to be an efficient iterative regularization method for recovering the correct image from its degradation [19]. Trust-region methods have recently shown another useful regularization tool for image restoration [20–22] and have proved to be a kind of regularization method [20,23]. These methods require solving a trust-region subproblem in each inner iteration and accepting a new trial step within its trust region. Recently, subspace trust-region methods were developed for solving large scale nonlinear programming problems (e.g. [24,25]), but attentions are paid mainly on well-posed problems. It is clear that these methods have potential applications in inverse problems. Iterative methods like steepest descent method and Landweber method, though applicable, are quite expensive, more advanced gradient methods need to be investigated [26].

In practice, the vector $\mathbf{f}$ records the image pixel values, so the component of $\mathbf{f}$ must be nonnegative. Therefore, we reformulate the problem by imposing the non-negativity constraints. This is already considered by several researchers [26–31,11]. The non-negativity constrained image deblurring problem is described as

$$\begin{aligned} \min \quad & \Psi(\mathbf{f}) := \tfrac{1}{2}\|\mathscr{K}\mathbf{f} - \mathbf{h_n}\|^2, \\ \text{s.t.} \quad & \mathbf{f} \geqslant 0. \end{aligned} \tag{7}$$

We denote the feasible set of (7) by $\mathscr{F} = \{\mathbf{f} : \mathbf{f} \geqslant 0\}$, and its interior by $int(\mathscr{F}) = \{\mathbf{f} : \mathbf{f} > 0\}$. The remaining task is how to solve (7) efficiently and accurately.

The research reported in the literature on ill-posed image deblurring is based on solving the inverse model in a full space, i.e., the iterative steps are in $\mathbb{R}^{N^2}$, not in a lower dimensional space $\mathbb{R}^p$, $p < N^2$. Therefore, a question arises of whether we can reduce the original problem from full space to a lower dimensional space (subspace). This paper will address this problem. The contribution of the paper lies in that we propose to apply a subspace interior-point trust-region method to solve (7). Though the idea is originated from optimization theory, it is the first time to implant this idea into image deblurring problem. In addition, for the linear inverse problem oriented quadratic programming model, we propose a new trust-region form for modifying trust-region radius and accepting trust region trial step. The great advantage of the method is that users can develop different kinds of subspace methods to carry out more efficient scientific computing.

This paper is organized as follows: in Section 2.4, a subspace interior-point trust-region algorithm for the non-negative image restoration is presented. The details of the construction of the subspace are described in Section 2.5. In Section 3, the numerical experiments based on the proposed method are reported. Some concluding remarks are given in Section 4. Finally, appendix is given to prove the convergence of the proposed method.

Throughout the paper, we use the following notations: "$:=$" denotes "defined as", "arg min" denotes "minimization for an argument", "max" and "min" denote "maximizing" and "minimizing" some functional, respectively, "A^T" denotes the transpose of matrix A, "s.t." denotes "subject to", "int($\mathscr{F}$)" denotes the interior of $\mathscr{F}$, " diag$(\cdot)$" denotes a diagonal matrix and "lim inf" of a sequence is the infimum of the set of all subsequence limits of the sequence.

2. Subspace method

2.1. Subspace trust-region model

The problem (7) can be regarded as a special case of nonlinear programming with simple bounds. The interior-point trust-region methods for this type problem were investigated by many authors [25,24,32–34] and proved to be robust and effi-

cient. But for the large scale problems, subspace techniques should be involved. In [24], comprehensive study and numerical experiments were given for well-posed nonlinear programming problems. In this paper, we study applying it to ill-posed problems.

The first order Karush–Kuhn–Tucker optimality condition for (7) can be expressed as

$$\begin{cases} \nabla\Psi(\mathbf{f}) - \lambda = 0, \\ \lambda_i \mathbf{f}_i = 0, \qquad i = 1, \ldots, N^2 \\ \lambda \geqslant 0, \mathbf{f} \geqslant 0, \end{cases}$$

where $\nabla\Psi(\mathbf{f}) = \mathscr{K}^T(\mathscr{K}\mathbf{f} - \mathbf{h_n})$ is the gradient of $\Psi(\mathbf{f})$ in (7). This system can be reduced to

$$D(\mathbf{f})\nabla\Psi(\mathbf{f}) = 0, \tag{8}$$

where $D(\mathbf{f})$ is a diagonal matrix whose diagonal elements are given by

$$(D(\mathbf{f}))_{ii} = \begin{cases} \mathbf{f}_i & \text{if } (\nabla\Psi(\mathbf{f}))_i > 0, \\ 1 & \text{if } (\nabla\Psi(\mathbf{f}))_i \leqslant 0. \end{cases} \tag{9}$$

Our subspace trust-region model for (7) is

$$\begin{aligned} \min \quad & \Phi(\mathbf{s}) := g_k^T\mathbf{s} + \tfrac{1}{2}\mathbf{s}^T\mathscr{K}^T\mathscr{K}\mathbf{s}, \\ \text{s.t.} \quad & \|D_k^{-1}\mathbf{s}\| \leqslant \Delta_k, \\ & \mathbf{s} \in S_k, \end{aligned} \tag{10}$$

where $g_k = \mathscr{K}^T(\mathscr{K}\mathbf{f}_k - \mathbf{h_n})$, D_k is defined as $D(\mathbf{f}_k)$ which is a scaling matrix to restrict the step $\mathbf{s}$, Δ_k is the radius of the trust region and S_k is a subspace which is chosen so that (10) can be solved cheaply.

2.2. Computation of a trial step

We denote the solution of (10) as $\mathbf{s}_k^{tr}$ and define $\bar{\mathbf{s}}_k^{tr}$ as

$$(\bar{\mathbf{s}}_k^{tr})_i = \begin{cases} -\sigma_k(\mathbf{f}_k)_i & \text{if } (\mathbf{s}_k^{tr})_i \leqslant -\sigma_k(\mathbf{f}_k)_i, \\ (\mathbf{s}_k^{tr})_i & \text{if } (\mathbf{s}_k^{tr})_i > -\sigma_k(\mathbf{f}_k)_i, \end{cases} \tag{11}$$

where $\sigma_k(\mathbf{f}_k) := \max\{\sigma, 1 - \|\mathbf{s}_k^{tr}\|\}$ and $\sigma \in (0,1)$. Thus $\bar{\mathbf{s}}_k^{tr}$ can be regarded as a truncated projection of $\mathbf{s}_k^{tr}$ onto $\mathrm{int}(\mathscr{F})$ with a small step back in order to make the step feasible and in the interior.

Except for $\mathbf{s}_k^{tr}$, the scaled gradient $-D_k^2 g_k$ is also a good direction which can be used to generate the trial step. The reason for using $-D_k^2 g_k$ instead of $-g_k$ is that the affine scaling matrix D_k takes a crucial role in the simple bound constrained problems. For the components $\mathbf{f}_i$ which are approaching the "correct" bounds, the sequence vectors $\{-D_k^2 g_k\}$ become increasingly tangential to these bounds. Hence, the bounds will not prevent a large stepsize along $\{-D_k^2 g_k\}$. For the components $\mathbf{f}_i$ which are approaching the "incorrect" bounds, $\{-D_k^2 g_k\}$ goes away from these bounds in relatively large angles so that we can get a sufficient descent along the direction $\{-D_k^2 g_k\}$. More details about the affine scaling matrix can be found in [24].

We define the minimizer of (10) in $\mathscr{F}$ along $-D_k^2 g_k$ as $\mathbf{s}_k^c$, i.e.,

$$\mathbf{s}_k^c = \arg\min\{\Phi(\mathbf{s}) : \mathbf{s} = -\tau D_k^2 g_k, \|D_k^{-1}\mathbf{s}\| \leqslant \Delta_k, \mathbf{f}_k + \mathbf{s} \in \mathscr{F}\}. \tag{12}$$

In practical implement, the step may need to be reduced in order to guarantee all $\mathbf{f}_k$ staying in $\mathrm{int}(\mathscr{F})$. So we define the Cauchy step $\bar{\mathbf{s}}_k^c$ as follows:

$$\bar{\mathbf{s}}_k^c := \theta_k \mathbf{s}_k^c, \tag{13}$$

where $\theta_k = 1$ when $\mathbf{f}_k + \mathbf{s}_k^c \in \mathrm{int}(\mathscr{F})$ and $\theta_k = \max\{\theta, 1 - \|\mathbf{s}_k^c\|\}, \theta \in (0,1)$ otherwise.

Then we can generate the trial step $\mathbf{s}_k$ by $\bar{\mathbf{s}}_k^{tr}$ and $\bar{\mathbf{s}}_k^c$. The following condition should be imposed to $\mathbf{s}_k$ to guarantee the global convergence of the algorithm:

$$\rho_k^c := \frac{\Phi(\mathbf{s}_k)}{\Phi(\bar{\mathbf{s}}_k^c)} \geqslant \beta_c, \tag{14}$$

where $\beta_c \in (0,1)$.

We use a line search technique to find $\mathbf{s}_k$ satisfying (14). The line segment is denoted by $\phi(t)$. Because $\bar{\mathbf{s}}_k^{tr}$ and $\bar{\mathbf{s}}_k^c$ are all good directions for (7), we can apply the line search along the line segment which is the convex combination of $\bar{\mathbf{s}}_k^{tr}$ and $\bar{\mathbf{s}}_k^c$, i.e.,

$$\phi(t) = t\bar{\mathbf{s}}_k^{tr} + (1-t)\bar{\mathbf{s}}_k^c. \tag{15}$$

The line search algorithm for finding the trial step can be presented as Algorithm 2.1.

Algorithm 2.1. Compute the trial step

[**Step 1**] Compute $\mathbf{s}_k^{tr}$ for the subproblem (10) and set $\bar{\mathbf{s}}_k^{tr}$ and $\bar{\mathbf{s}}_k^c$ by (11) and (13), respectively. Set $t = 1, iter = 1$. Give $\beta_c, \gamma \in (0,1)$ and integer *Maxiter*;
[**Step 2**] For $iter = 1 : Maxiter$

- Compute $\mathbf{s}_k = t\bar{\mathbf{s}}_k^{tr} + (1-t)\bar{\mathbf{s}}_k^c$ and ρ_k^c by (14);
- If $\rho_k^c < \beta_c$, set $t := \gamma t, iter := iter + 1$;
- Otherwise stop;

[**Step 3**] Set $\mathbf{s}_k = \bar{\mathbf{s}}_k^c$ and stop.

We give some explanations to this algorithm. First, we test whether (14) holds when $\mathbf{s}_k = \bar{\mathbf{s}}_k^{tr}$, i.e., $t = 1$. If (14) does not hold, we reduce t so that the step $\mathbf{s}_k(t) = t\bar{\mathbf{s}}_k^{tr} + (1-t)\bar{\mathbf{s}}_k^c$ is more and more close to $\bar{\mathbf{s}}_k^c$. Finally, if Step 3 is implemented, which shows that (14) does not hold when reduce t in certain times, then we set $\mathbf{s}_k = \bar{\mathbf{s}}_k^c$ so that (14) holds naturally. In addition, in Step 1, the cost of computing $\mathbf{s}_k^{tr}$ must be accounted. For large scale problem, we solve it by converting a full space problem to a lower dimensional space. Details are given in Algorithm 2.3.

2.3. A new form of trust-region scheme for quadratic programming

Trust-region methods are usually formulated for non-quadratic nonlinear programming problem. Consider, for example, an unconstrained non-quadratic minimization problem $\min_{\mathbf{f}\in\mathbb{R}^n} \Gamma(\mathbf{f})$. The trust-region method requires solving a trust region subproblem

$$\begin{aligned} \min_s \quad & \Upsilon(s) := (g(\mathbf{f}), s) + \frac{1}{2}(H(\mathbf{f})s, s), \\ \text{s.t.} \quad & \|s\| \leqslant \Delta, \end{aligned}$$

where $g(\mathbf{f})$ and $H(\mathbf{f})$ denote the gradient and Hessian of $\Gamma(\mathbf{f})$, respectively. In each step, a trial step s is computed and decided whether it is acceptable or not. The decision rule is based on the ratio ρ between the actual reduction in the objective functional and the predicted reduction in the approximate model. And the trust-region iterative step remains unchanged if $\rho \leqslant 0$, where $\rho = \frac{Ared(\mathbf{f})}{Pred(\mathbf{f})}$, and $Ared(\mathbf{f})$ and $Pred(\mathbf{f})$ are defined by $\Gamma(\mathbf{f}) - \Gamma(\mathbf{f}+s)$ and $\Upsilon(0) - \Upsilon(s)$, respectively.

For the model (5), since it is in a quadratic form, the ratio ρ is always equal to 1. This means the trial step s, no matter it is good or not, will be always accepted. But the model is ill-posed, this seems to be unreasonable. To overcome this shortcoming, we propose the following modified trust-region scheme when the model is a quadratic programming problem. We note that the approximate accuracy is characterized by the discrepancy between the observation and the true data; therefore variations of the norm of the discrepancy may reflect the degree of approximation. Based on these considerations, we propose to accept or reject the trial step s_k at the kth step by the ratio $\rho_k = \frac{\Psi(\mathbf{f}_{k+1})}{\Psi(\mathbf{f}_k)} = \frac{\Psi(\mathbf{f}_k+\mathbf{s}_k)}{\Psi(\mathbf{f}_k)}$, where $\Psi(\mathbf{f}_{k+1})$ and $\Psi(\mathbf{f}_k)$ are the reductions in norm of the discrepancy at $(k+1)$st and kth steps, respectively.

2.4. A subspace interior-point trust-region method

After getting a trial step $\mathbf{s}_k$ satisfying (14), we decide to accept or reject $\mathbf{s}_k$. Our rule is whether the function value $\Psi(\mathbf{f}_k + \mathbf{s}_k)$ has some reductions compared to $\Psi(\mathbf{f}_k)$. We denote the ratio of $\Psi(\mathbf{f}_k + \mathbf{s}_k)$ and $\Psi(\mathbf{f}_k)$ as ρ_k^f:

$$\rho_k^f = \frac{\Psi(\mathbf{f}_k + \mathbf{s}_k)}{\Psi(\mathbf{f}_k)}. \tag{16}$$

In our algorithm, we will accept $\mathbf{s}_k$ if $\rho_k^f < \eta$ and reject it otherwise, where $\eta \in (0,1)$. The reason we adopt this stopping rule is that (16) uses discrepancy between the observation and the model, which is more physically meaningful [18].

It is worth noting that $\bar{\mathbf{s}}_k^{tr}$ will tend to the scaled gradient direction when the radius of the trust region Δ_k approaching zero. So the update of Δ_k is also important when generate the good trial step. But Δ_k should not be expected to restrict the point stay interior. In fact, it is important to allow the trial step past the boundary of the trust region as long as it is still satisfying the bound constraints. In our algorithm, we use a similar approach proposed in [34] to update Δ_k.

Based on the above discussions, we describe our interior-point trust-region algorithm as Algorithm 2.2.

Algorithm 2.2. A subspace interior-point trust-region algorithm

[**Step 1**] Give $\epsilon, \beta_c, \gamma_0 \in (0,1), 1 = \mu_2 \geqslant \mu_1 \geqslant \eta > 0, \gamma_2 \geqslant \gamma_1 > 1, x_0 \in \int(\mathscr{F}), \Delta_0 > 0$ and set $k := 0$.
[**Step 2**] If $\|D_k g_k\| < \epsilon$, stop.
[**Step 3**] Compute $\Psi(\mathbf{f}_k)$ and g_k; Define the quadratic model Φ as (10).
[**Step 4**] Compute the trial step $\mathbf{s}_k$ by Algorithm 2.1.
[**Step 5**] Compute ρ_k^f by (16). If $\rho_k^f < \eta$ then set $\mathbf{f}_{k+1} := \mathbf{f}_k + \mathbf{s}_k$. Otherwise $\mathbf{f}_{k+1} := \mathbf{f}_k$.

[**Step 6**] Set $k := k + 1$; Update Δ_k as follows and then go to Step 2.

- If $\rho_k^f \geqslant \mu_2$ then $\Delta_{k+1} = \gamma_0 \Delta_k$.
- If $\mu_1 \leqslant \rho_k^f < \mu_2$ then $\Delta_{k+1} \in [\Delta_k, \gamma_1 \Delta_k]$.
- Otherwise, $\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k]$.

It deserves attention that in Algorithm 2.2, adjusting the trust-region radius Δ_k requires $1 = \mu_2 \geqslant \mu_1 \geqslant \eta > 0$ and $\gamma_2 \geqslant \gamma_1 > 1$. These requirements are necessary to guarantee convergence. In practical simulations, the requirements can be relax. For example, we can let μ_2 less than but toward 1 and $\mu_1 < \mu_2$. Typical values are: $\eta = \mu_1 = 0.95, \mu_2 = 0.9995, \gamma_0 = 0.8, \gamma_1 = 1.5$ and $\gamma_2 = 2.0$.

It is necessary to establish the convergence property of the proposed trust-region algorithm to the ill-posed quadratic programming problem. Before we get the main convergence result, some important assumptions should be declared as [34]:

(AS.1) *Given an initial point* $\mathbf{f}_0 \in \mathscr{F}$, *it is assumed that* $\mathscr{L}$ *is compact, where* $\mathscr{L}$ *is the level set, i.e.,* $\mathscr{L} = \{\mathbf{f} : \mathbf{f} \in \mathscr{F} \text{ and } \Psi(\mathbf{f}) \leqslant \Psi(\mathbf{f}_0)\}$.

(AS.2) *There exists a positive scalar* χ_g *such that for* $\mathbf{f} \in \mathscr{L}, \|g(\mathbf{f})\|_\infty < \chi_g$.

We will prove the following theorem:

Theorem 2.1. *Assume that (AS.1), (AS.2) hold and* $\{\mathbf{s}_k\}$ *satisfies (14), and since* $\Psi : \mathbb{R}^{N^2} \to \mathbb{R}$ *is continuously differentiable on* $\mathscr{F}$, *then*

$$\liminf_{k\to\infty} \|\hat{g}_k\| = 0. \tag{17}$$

Proof. The proof is shown in the Appendix. □

The next theorem establishes that $\{D_k g_k\}$ converges to zero.

Theorem 2.2. *Assume (AS.1) and (AS.2) hold and* $\{\mathbf{f}_k\}$ *is generated by* Algorithm 2, *and since* $g(\mathbf{f}) = \nabla\Psi(\mathbf{f})$ *is continuous on* $\mathscr{F}$, *then*

$$\lim_{k\to\infty} \|D_k g_k\| = 0.$$

Proof. The proof is shown in the Appendix. □

2.5. Solution in two-dimensional subspace trust region

In this subsection, we will give the choice of subspace S_k and the algorithm to solve (10) cheaply. Firstly, the scaled gradient $-D_k^2 g_k$ should be included in the subspace. Secondly, S_k should contain a sufficiently accurate approximation to the Newton direction $-(\mathscr{K}^T\mathscr{K})^{-1} g_k$. Due to the large scale of the matrix, we should get an inexact Newton step $\mathbf{s}_k^N$ by approximately solving the equation

$$\mathscr{K}^T\mathscr{K}\mathbf{s} = -g_k \tag{18}$$

with accuracy η_k:

$$\mathscr{K}^T\mathscr{K}\mathbf{s}_k^N = -g_k + r_k, \quad \text{such that } \|r_k\|/\|g_k\| \leqslant \eta_k. \tag{19}$$

Hence we can select S_k as the two-dimensional subspace

$$S_k = \text{span}\{\mathbf{s}_k^N, -D_k^2 g_k\}. \tag{20}$$

Eq. (18) can be solved by conjugate gradient method so that all the cost of the algorithm is matrix–vector multiplication, which can be operated efficiently using the technique given in Section 3.2.1. It deserves attention that the step $\mathbf{s}_k^N$ lies in a subspace spanned by $\mathscr{S}_r := \{g_k, Hg_k, H^2 g_k, \ldots, H^r g_k\}$ for some r, where $H = \mathscr{K}^T\mathscr{K}$. Here r is unknown when formulating subspace $\mathscr{S}_r$ in each iteration. While by using conjugate gradient method to approximately solve Eq. (18), we do not need to know r and we always get an inexact Newton direction which lies in some subspace $\mathscr{S}_r$.

The remaining work is how to solve the subspace trust region problem (10) efficiently. We define

$$\mathbf{s}_1 = \mathbf{s}_k^N / \|\mathbf{s}_k^N\| \tag{21}$$

and compute

$$\mathbf{s}_2 = -D_k^2 g_k + ((D_k^2 g_k)^T \mathbf{s}_1)\mathbf{s}_1 \tag{22}$$

and then we set $\mathbf{s}_2 = \mathbf{s}_2/\|\mathbf{s}_2\|$ and $S = [\mathbf{s}_1, \mathbf{s}_2] \in \mathbb{R}^{N^2\times 2}$. Then for any vector $\mathbf{s} \in S_k$, there exists a two-dimensional vector $\alpha = [\alpha_1, \alpha_2]^T \in \mathbb{R}^2$ satisfy $\mathbf{s} = S\alpha$. Hence, problem (10) can be formulated as a two-dimensional trust-region subproblem:

$$\begin{aligned} \min \quad & \Phi(\alpha) := g_k^T S\alpha + \tfrac{1}{2}\alpha^T S^T \mathscr{K}^T \mathscr{K} S\alpha \\ \text{s.t.} \quad & \|D_k^{-1} S\alpha\| \leqslant \Delta_k. \end{aligned} \tag{23}$$

With the above description, we can present the following algorithm for solving the subspace trust-region model (10) in detail:

Algorithm 2.3. Solving the subspace trust-region problem

[**Step 1**] Compute $\mathbf{s}_1$ and $\mathbf{s}_2$ by (21) and (22), respectively;
[**Step 2**] Set $\mathbf{s}_2 = \mathbf{s}_2/\|\mathbf{s}_2\|$ and $S = [\mathbf{s}_1, \mathbf{s}_2] \in \mathbb{R}^{N^2\times 2}$.
[**Step 3**] Solve the two-dimensional trust-region subproblem (23) to get α.
[**Step 4**] Set $\mathbf{s}_k^{tr} = S\alpha$.

3. Numerical experiments

3.1. Long slit PSF blurred one-dimensional ill-conditioning problem

The infinite long slit PSF can be written as the following function [35]

$$k(\theta, \phi) = (\cos\theta + \cos\phi)\left(\frac{\sin r}{r}\right)^2, \tag{24}$$

where

$$r = \frac{\pi w}{\lambda}(\sin\theta + \sin\phi),$$

θ is the angle of emergence or observation which specifies the location of the image point, ϕ is the angle of incidence which specifies the location of the source, w is the width of the slit, λ is the wavelength. The illustration of the discrete PSF is given in Fig. 1. The input signal is assumed to be the superimposed Gaussian

$$f(\phi) = \exp(-c_1(\phi - \phi_a)^2) + c(-c_2(\phi - \phi_b)^2).$$

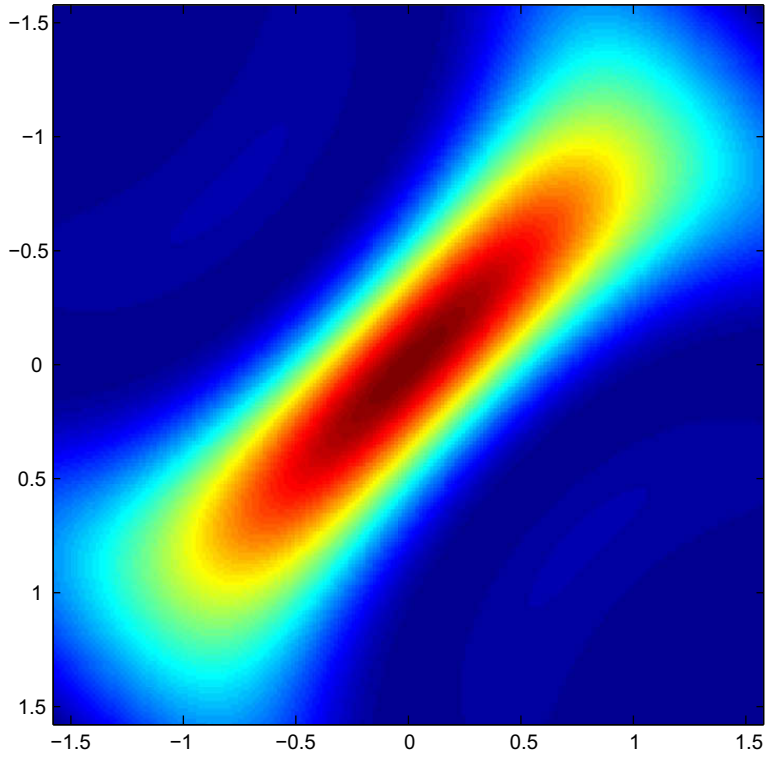

Fig. 1. The long slit PSF image.

Assuming that $\theta, \phi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, we choose $c = 1, c_1$ and c_2 as high as 4, and ϕ_a and ϕ_b as small as $\pm\frac{1}{2}$. The true right-hand side h is obtained by integral:

$$h(\theta) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} k(\theta, \phi) f(\phi) d\phi.$$

After discretization, we obtain the discrete matrix–vector multiplication: $\mathcal{K}\mathbf{f} = \mathbf{h}$, where $\mathcal{K} \in \mathbb{R}^{M\times N}, \mathbf{f} \in \mathbb{R}^N$ and $\mathbf{h} \in \mathbb{R}^M$. We choose $M = N = 200$, which leads to the condition number of the matrix $\mathcal{K}$ is 2.5631×10^{19}. Hence the problem is severely ill-conditioning. To test the robustness of our method, we add noise to $\mathbf{h}$. And the noise level is denoted by level, i.e.,

$$\mathbf{n} = \frac{\text{level}}{M}\|\mathbf{h}\| \times randn(M, 1),$$

where M is the size of the image, $randn(M, 1)$ is a Gaussian normal distributed random vector, and we set $randn(\text{'}state\text{'}, 0)$ in our Matlab codes to ensure the same random vector is generated every time.

We set the parameters in Algorithm 2.1 and Algorithm 2.2 as follows: $\sigma = 0.9995, \theta = 0.9995, \gamma = 0.9$, Maxiter=10, $\beta_c = 0.1, \mu_1 = \eta = 0.95, \mu_2 = 0.9995, \gamma_0 = 0.8, \gamma_1 = 1.5, \gamma_2 = 2.0, \mathbf{f}_0$ is the initial guess value with components all 0.5, and $\Delta_0 = \|\mathbf{f}_0 . * g_0\|$.

The noise added to $\mathbf{h}$ is with levels: level = 0.005, 0.01 and 0.05. The input, blurred and restored images are shown, respectively, in Figs. 2–8. It indicates from these figures that the unknown signal is well recovered for not too large noise level. For large noise level, there are tails on the boundary. In such case, other regularizers, say, nonsmooth regularizer [4] for controlling jump on the boundary, may be incorporated into the model, but this beyond our study in this paper. The precision of the approximation is characterized by the root mean-square error (rmse)

$$\text{rmse} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\frac{(\mathbf{h}_{\mathbf{n}}(\theta_i) - \mathbf{h}(\theta_i))^2}{(\mathbf{h}_{\mathbf{n}}(\theta_i))^2}}, \tag{25}$$

which describes the average relative deviation of the recovered signals from the true signals. For the first case, it needs six steps to reach the convergence with rmse = 0.0016. For the second, it needs six steps to reach the convergence with rmse = 0.0013. For the third, it needs four steps to reach the convergence with rmse = 0.0053. Therefore, we conclude that our proposed method is applicable for ill-conditioned deblurring problems.

3.2. Single channel image deblurring

For astronomical image deblurring problems, the reason for causing blur is various [1,2]. To show the efficiency of our method, we only consider a simple case, i.e. the PSF is modeled by Gaussian and the image is formed in a single channel.

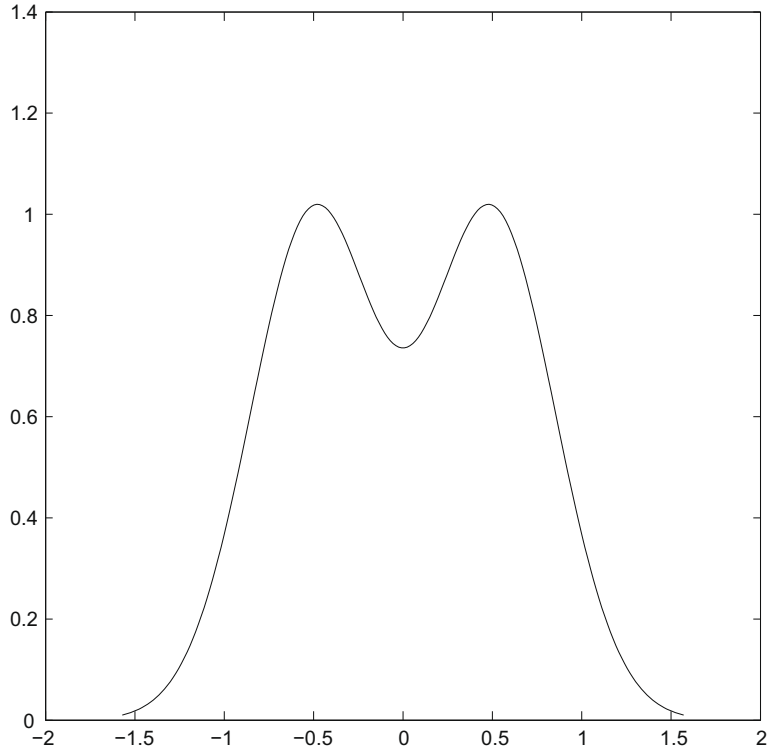

Fig. 2. The noise-free input one-dimensional image.

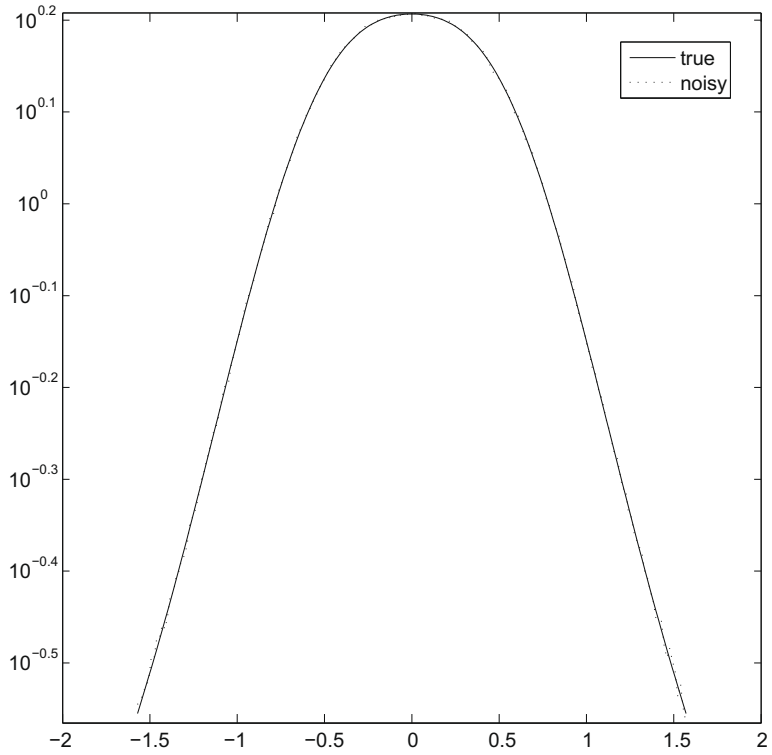

Fig. 3. The true and blurred images for noise level equaling to 0.005.

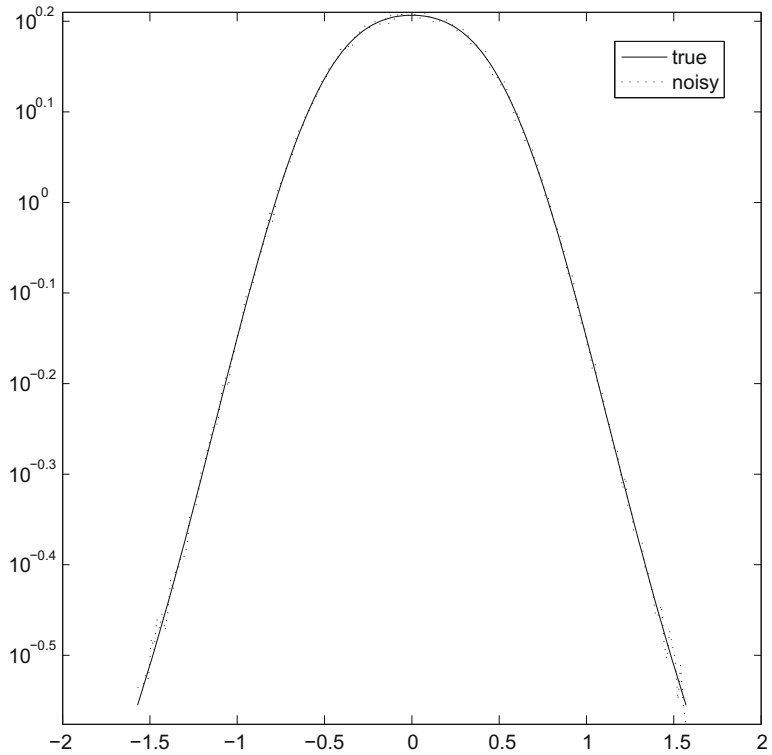

Fig. 4. The true and blurred images for noise level equaling to 0.01.

Actually, the Gaussian function simulates well about the convolution process of the true signal with the PSF operator. Both the blurring by aerosols and turbulence can be taken as Gaussian. The forward model is in the form

$$k(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{1}{2}\left(\frac{x^2+y^2}{\sigma^2}\right)\right), \tag{26}$$

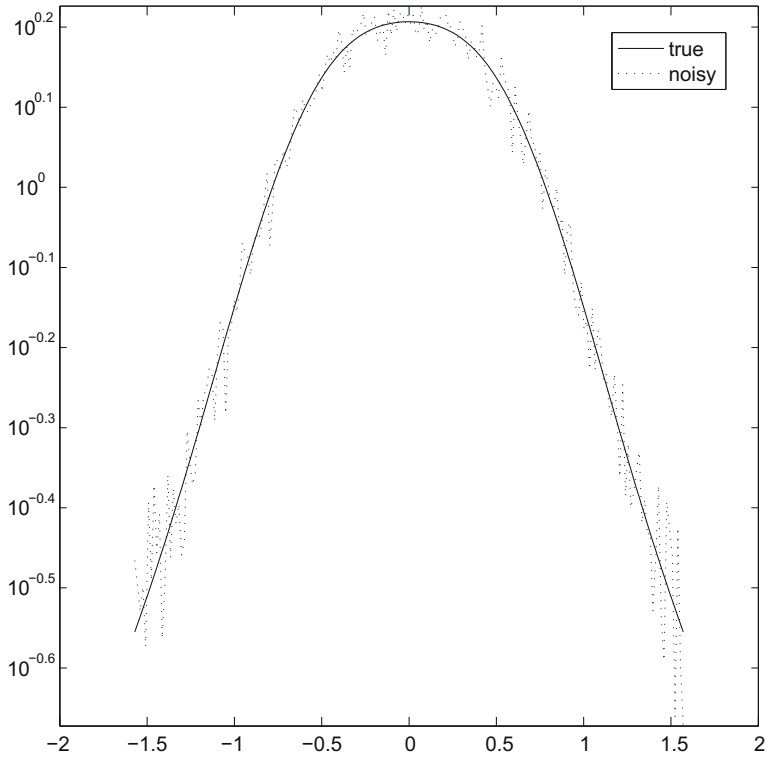

Fig. 5. The true and blurred images for noise level equaling to 0.05.

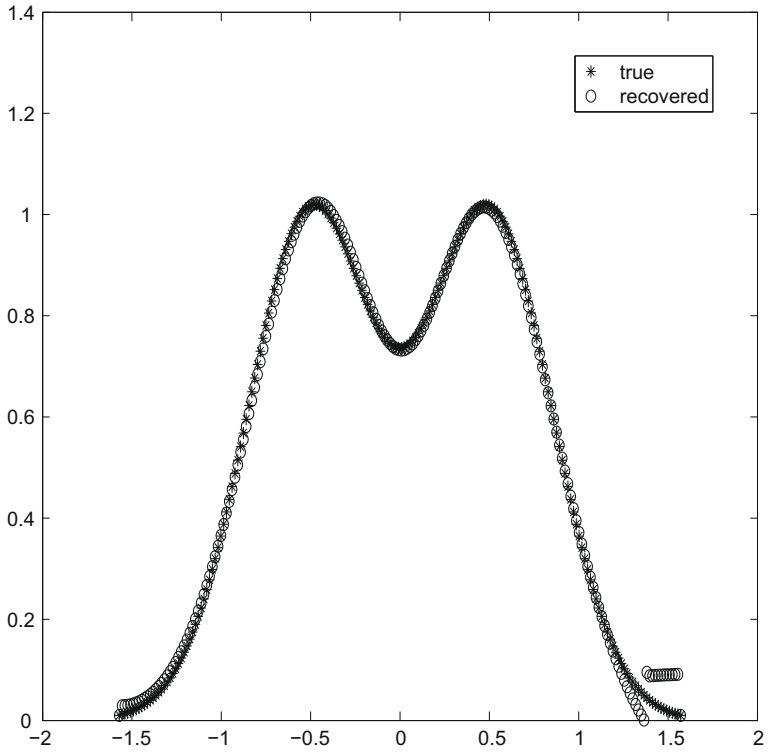

Fig. 6. Restored images for $\epsilon = 10^{-3}$ and level = 0.005.

where σ is a positive constants. The larger we choose σ, the more f gets smoothed. So by the same argument, the smaller we choose σ, the more the convolution result resembles f.

3.2.1. MVM: matrix–vector multiplication

The main cost of our algorithm is the matrix–vector multiplication, so an efficient algorithm to compute the matrix–vector multiplication should be investigated.

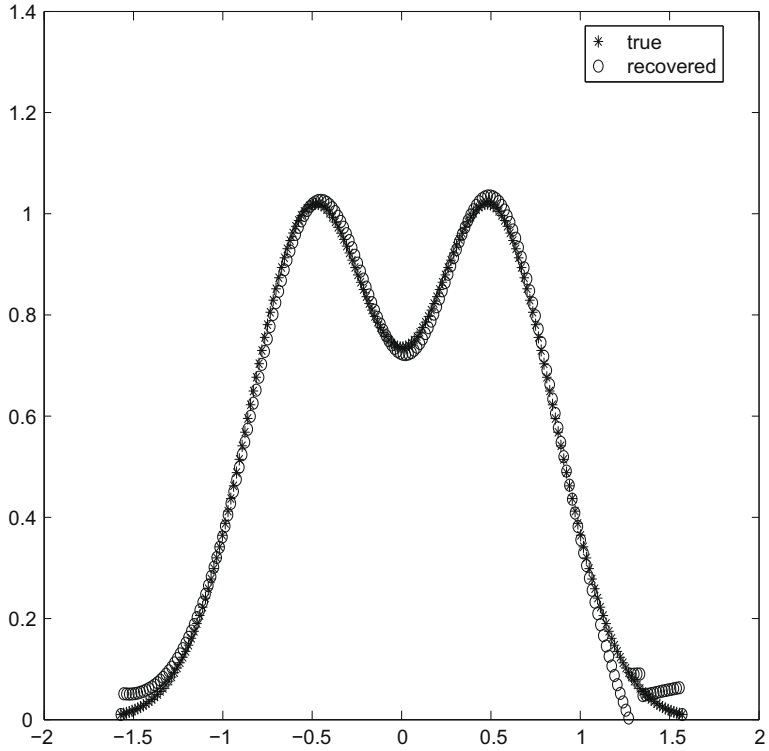

Fig. 7. Restored images for $\epsilon = 10^{-3}$ and level = 0.01.

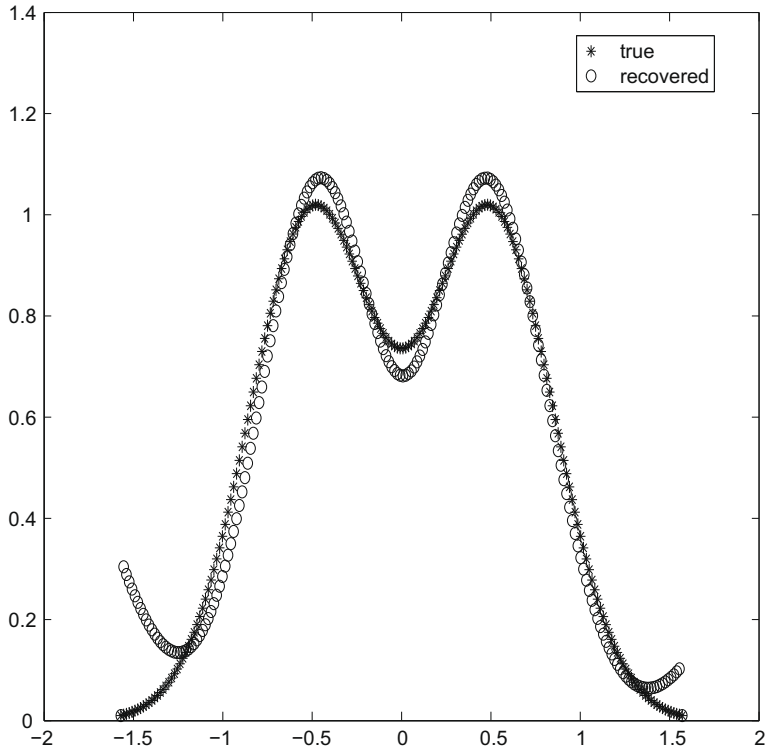

Fig. 8. Restored images for $\epsilon = 10^{-3}$ and level = 0.05.

We suppose the PSF kernel function in (2) is spatially invariant, i.e., the kernel is separable and can be reformulated as

$$k(x-\xi, y-\eta) = k_x(x-\xi)k_y(y-\eta). \tag{27}$$

Numerically, assume that the discretization of k_x and k_y are $\mathscr{K}_x$ and $\mathscr{K}_y$ respectively, then the matrix $\mathscr{K}$ is a tensor of $\mathscr{K}_x$ and $\mathscr{K}_y$, i.e., the Kronecker product of $\mathscr{K}_x$ and $\mathscr{K}_y$,

$$\mathscr{K} = \mathscr{K}_x \otimes \mathscr{K}_y. \tag{28}$$

$\mathscr{K}_x$ and $\mathscr{K}_y$ are all Toeplitz, so $\mathscr{K}$ is a block Toeplitz matrix with Toeplitz blocks (BTTB), i.e., the $N^2 \times N^2$ matrix $\mathscr{K}$ has the block form

$$\mathscr{K} = \begin{pmatrix} \mathscr{K}_0 & \mathscr{K}_{-1} & \cdots & \mathscr{K}_{1-N} \\ \mathscr{K}_1 & \mathscr{K}_0 & \mathscr{K}_{-1} & \vdots \\ \vdots & \ddots & \ddots & \mathscr{K}_{-1} \\ \mathscr{K}_{N-1} & \cdots & \mathscr{K}_1 & \mathscr{K}_0 \end{pmatrix}, \tag{29}$$

where each block $\mathscr{K}_j$ is an $N \times N$ Toeplitz matrix. For a Toeplitz matrix, it can be determined by its first row and first column elements. By extending the BTTB into a block circulant with circulant blocks matrix (BCCB), we can use the two dimensional discrete Fourier transform to compute the matrix vector multiplication [4,18].

Since $\mathscr{K}$ can be represented by a Kronecker product of two low order matrices as $\mathscr{K} = A \otimes B$ with $A \in \mathbb{R}^{m\times m}, B \in \mathbb{R}^{n\times n}$. For the blurring process, A and B are usually taken as sparse banded matrices [2], which means only pixels within a distance band -1 contribute to the blurring. Hence we can use a very economic algorithm proposed in [21,22]. In these papers, the authors only considered the band which is equal to 2 and 3. In this paper, we extend their results to any band value.

Suppose band $= p$, then A, B are $2p-1$ diagonal matrices. Pixels within a distance $p-1$ of A and B contribute to the blurring. The different elements of $\mathscr{K}$ are only $C = A(1:p,1) \otimes B(1:p,1)$. The resulting matrix $\mathscr{K}$ is a sparse BTTB with each block a $2p-1$ diagonal matrix. If we define

$$A = \begin{pmatrix} a_0 & a_1 & \cdots & a_{p-1} & \cdots & 0 \\ a_1 & a_0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_{p-1} \\ a_{p-1} & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_0 & a_1 \\ 0 & 0 & a_{p-1} & \cdots & a_1 & a_0 \end{pmatrix}, \quad B = \begin{pmatrix} b_0 & b_1 & \cdots & b_{p-1} & \cdots & 0 \\ b_1 & b_0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & b_{p-1} \\ b_{p-1} & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & b_0 & b_1 \\ 0 & 0 & b_{p-1} & \cdots & b_1 & b_0 \end{pmatrix},$$

then $C := (a_0b_0, a_0b_1, \ldots, a_0b_{p-1}, \ldots, a_{p-1}b_0, a_{p-1}b_1, \ldots, a_{p-1}b_{p-1})^T$. Thus, we can write the matrix–vector multiplication $y = (A \otimes B)x$ as follows:

$$y = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{p-1}Bx_1 \\ a_{p-1}Bx_2 \\ \vdots \\ a_{p-1}Bx_{m-p+1} \end{pmatrix} + \cdots + \begin{pmatrix} 0 \\ a_1Bx_1 \\ \vdots \\ a_1Bx_{m-1} \end{pmatrix} + \begin{pmatrix} a_0Bx_1 \\ a_0Bx_2 \\ \vdots \\ a_0Bx_m \end{pmatrix} + \begin{pmatrix} a_1Bx_2 \\ \vdots \\ a_1Bx_m \\ 0 \end{pmatrix} + \cdots + \begin{pmatrix} a_{p-1}Bx_p \\ a_{p-1}Bx_{p+1} \\ \vdots \\ a_{p-1}Bx_{2p-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \end{pmatrix}.$$

Then each component of y, for example a_0Bx_1, can be evaluated as,

$$a_0b_{p-1}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ x_{1p} \\ \vdots \\ x_{1,n-p+1} \end{pmatrix} + \cdots + a_0b_1\begin{pmatrix} 0 \\ x_{11} \\ \vdots \\ x_{1,n-1} \end{pmatrix} + a_0b_0\begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{pmatrix} + a_0b_1\begin{pmatrix} x_{12} \\ \vdots \\ x_{1n} \\ 0 \end{pmatrix} + \cdots + a_0b_{p-1}\begin{pmatrix} x_{1p} \\ \vdots \\ x_{1n} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

so to others. Thus the matrix–vector multiplication reduces a lot.

Suppose band = 5, then A, B are nine diagonal matrices. Pixels within a distance 4 of A and B contribute to the blurring. The different elements of $\mathscr{K}$ are only $C = A(1:5,1) \otimes B(1:5,1)$. The resulting matrix $\mathscr{K}$ is a sparse BTTB with each block a five diagonal matrix. For the nine diagonal matrices, the cost of the MVM computation would be $25mn$. Whereas for FFT based matrix–vector computation, the cost is $O(5mn\log_2 mn)$, which is greater than $25mn$ for banded matrix–vector multiplication for large m and n, say, $m = n = 256,\ 512$ or more.

3.2.2. Numerical simulation

In this section, we give examples on the restoration of atmospheric image. The blurring process is modeled by a Gaussian point spread function:

$$k(x-\xi, y-\eta) = \frac{1}{2\pi\rho\bar{\rho}} \exp\left(-\frac{1}{2}\left(\frac{x-\xi}{\rho}\right)^2 - \frac{1}{2}\left(\frac{y-\eta}{\bar{\rho}}\right)^2\right). \tag{30}$$

In our test, we choose $\rho = \bar{\rho} = 0.7$. And the noise level is denoted by level, i.e.,

$$\mathbf{n} = \frac{\text{level}}{N}\|\mathbf{h}\| \times randn(N^2, 1),$$

where N is the size of the image, $randn(N^2, 1)$ is a Gaussian normal distributed random vector, and we set $randn(\text{'}state\text{'}, 0)$ in our Matlab codes to insure the same random vector is generated every time. It is clear that our method is also applicable for other image restoration problems induced by other blurring operator. Users may readily make codes according to our description of the algorithms.

The parameters in Algorithm 2.1 and Algorithm 2.2 for two-dimensional case are the same as that in one-dimensional numerical case except that $\gamma_0 = 0.6, \gamma_1 = 1.1, \gamma_2 = 1.5, \mathbf{f}_0$ is the initial guess value with components all ones, and $\Delta_0 = \|\mathbf{f}_0 . * g_0\|$.

The experiments were performed on a personal Pentium (R) computer with CPU 2.80 GHz. The image for testing is a single channel image, cropland, with size 256×256. Hence the resulting PSF matrix is a BTTB with size equaling $65,536 \times 65,536$. The true image is plotted in Fig. 9. To simulate the blurring, we choose the band which is equal to 5. This induces a severe atmospheric or turbulence blurring. The condition number of the discrete Kronecker kernel $\mathscr{K}$ is equal to 1.2985×10^{30}. Therefore, the matrix $\mathscr{K}$ is very badly conditioned. On occasion that the weather is not too bad, one may choose a small band value. The noisy blurred images for different noise levels are plotted in Figs. 10 and 11. The deblurred restored images by fast subspace algorithms are illustrated in Figs. 12 and 13.

As in the one-dimensional simulation, we also compute the root mean-square error (rmse)

$$\text{rmse} = \sqrt{\frac{1}{N^2}\sum \frac{(\mathbf{h}(:) - \mathbf{h_n}(:))^2}{(\mathbf{h_n}(:))^2}},$$

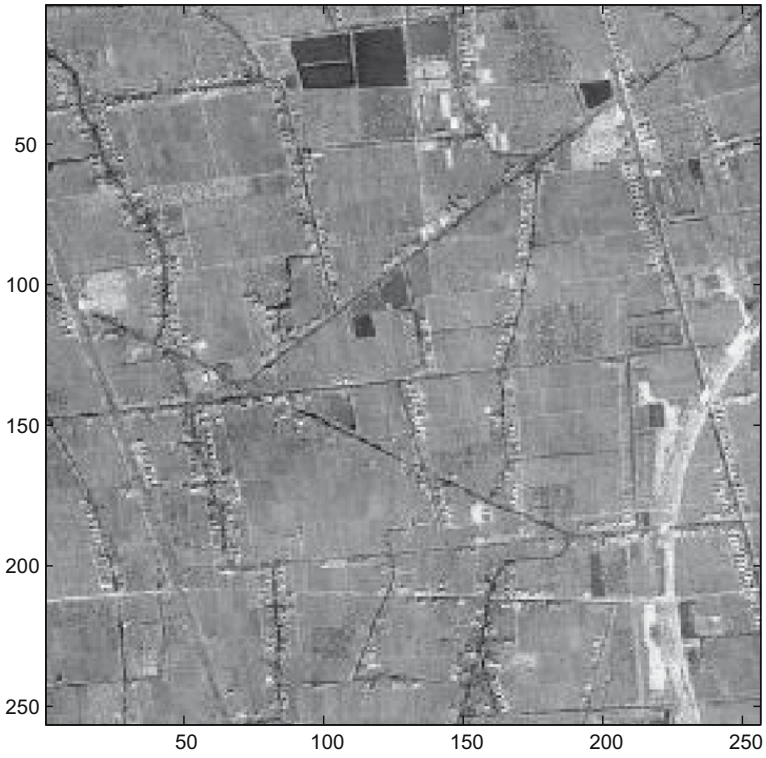

Fig. 9. The noise-free remotely sensed image.

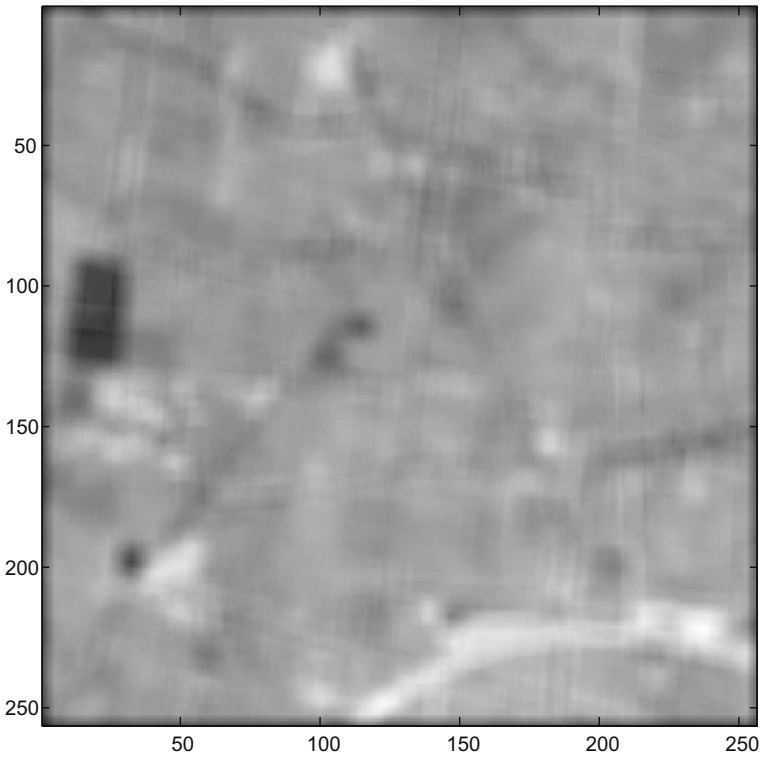

Fig. 10. The blurred images for noise level level = 0.005.

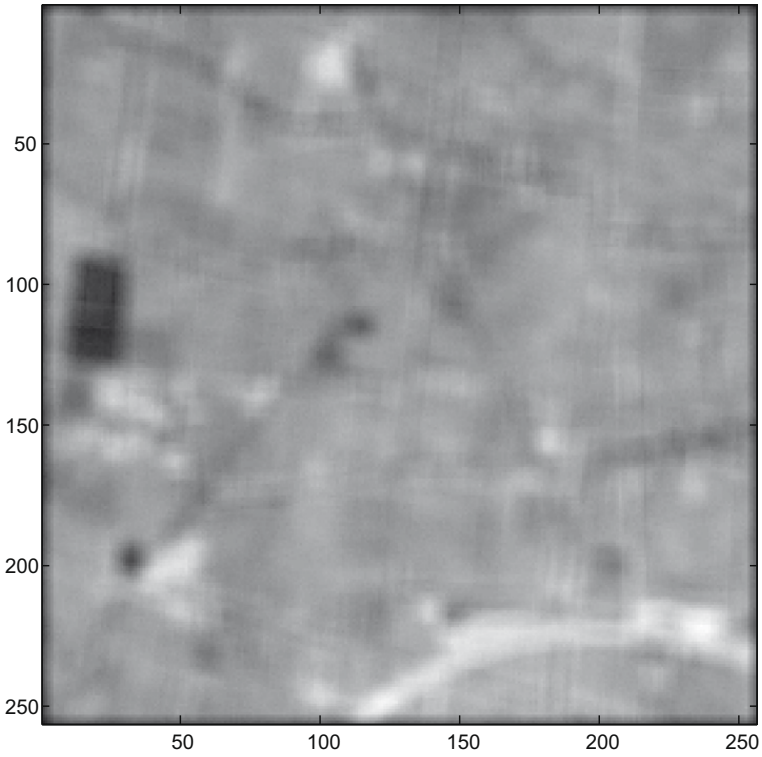

Fig. 11. The blurred images for noise level level = 0.01.

where $\mathbf{h}(:)$ refers that the operation is performed point to point. The numerical results by our subspace interior-point trust-region algorithm for different noise levels with different accuracy are listed in Table 1. In this table, the iterative steps refer to the recorded steps in Algorithm 2.2.

To show the efficiency of our proposed method, we make a comparison of this method with the projected BB (PBB) method which was developed recently for ill-posed problems in [26]. The PBB method is reported that it is more efficient than the well-established GPCG method. Therefore, this is a fair comparison. The results are listed in Table 1.

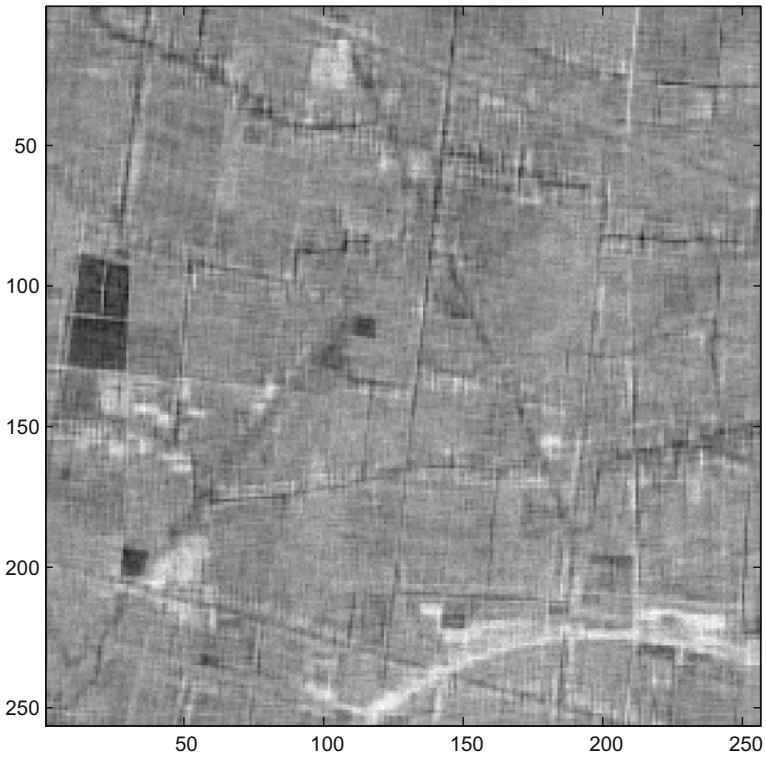

Fig. 12. Restored images for $\epsilon = 10^{-3}$ and level = 0.005.

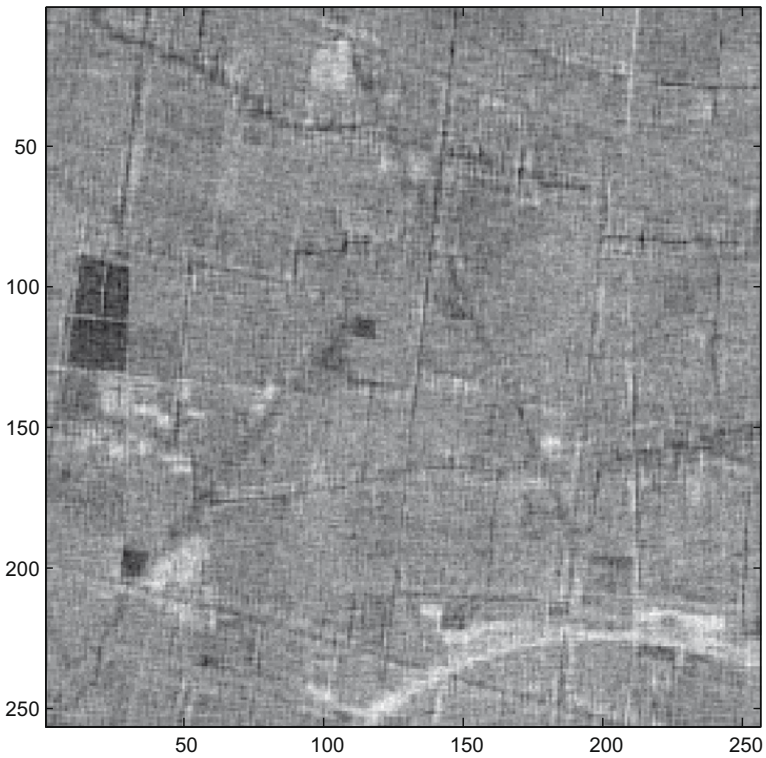

Fig. 13. Restored images for $\epsilon = 10^{-3}$ and level = 0.01.

It indicates from the computational results that our subspace method with the fast MVM algorithm is very fast for image deblurring problem. It generates comparable results as in PBB method. Certainly, our method is suitable for other blurring problems which are in the form of *banded* and *sparse* matrix–vector multiplication. Therefore, we conclude that our subspace method may be a proper choice for large scale image restoration with spatially invariant response.

Table 1
Comparison of subspace method with PBB method for different noise levels.

	Noise levels: level	Iterative steps	rmse	CPU (s)
Subspace method	0.005	2	0.0034	7.0469
	0.01	2	0.0062	7.5938
PBB method	0.005	16	0.0053	7.1400
	0.01	21	0.0058	8.5940

Table 2
Typical values of the parameters.

Parameters for Algorithm 2.1	ϵ	1.0×10^{-3}
	β_c	0.5
	γ	0.9
Parameters for Algorithm 2.2	γ_0	0.8
	γ_1	1.5
	γ_2	2.0
	η	0.95
	μ_1	0.95
	μ_2	0.9995
	$\mathbf{f}_0$	$[1,\ldots,1]^T$
	Δ_0	$\|\mathbf{f}_0 \cdot * g_0\|$

Remark 3.1. For any iterative solution methods, the proper stopping rule is a critical choice. Especially for image deblurring problems, the ill-posed nature indicates that there exists a saturation state for iterations [6], which means, at such a state, further iterations are unnecessary. In our algorithm, the parameter ϵ plays such a role. A smaller ϵ yields more accurate results, however brings more steps of computations; a larger ϵ makes the convergence quickly but with larger accurate results. In our example, we choose $\epsilon = 1.0 \times 10^{-3}$. Empirically, we recommend to choose $\epsilon = 1.0 \times 10^{-3}$ or $\epsilon = 1.0 \times 10^{-2}$, but less than or equal to 1.0×10^{-4}.

3.3. Discussions

One may raise a question that there are about 10 parameters need to be chosen before using the algorithm, which may reduce the applicability of the algorithm. We want to point out that the values of these parameters are mainly used for prove the convergence of the proposed algorithm. Practically, the choice can be loose. We give a table for typical choices of these parameters: These parameters work efficiently for most of ill-posed inverse problems. Therefore, the algorithm developed in this paper is readily used for practical applications.

We want to point out that the numerical examples given in this paper are the long slit PSF and the Gaussian PSF. It is clear that the proposed algorithm is also suitable for other image recovering problems induced by other ways of blurring as long as we recall that the formation of the optimization model is a nonnegative quadratic programming problem (see Table 2).

4. Conclusion

We investigated a new method to solve ill-posed image deblurring problems by applying trust-region technique. The method applies subspace technique to the trust-region subproblem, which possesses not only the efficiency but also the global convergence.

The numerical experiments on the restoration of the degraded image showed that the restoration methods proposed in this paper can restore the original input with relatively small root mean-square errors. Therefore, we concluded that our proposed algorithm is applicable for astronomical image deblurring problems.

Acknowledgements

We would like to thank referees and Dr. Melvin Scott for their insightful comments and suggestions. This work is supported by the National Natural Science Foundation of China under Grant No. 10871191 and National "973" Key Basic Research Developments Program of China under Grant Nos. 2005CB422104 and 2007CB714400.

Appendix A. Global convergence result

To analyze the convergence properties of Algorithm 2.2, we need to prove the following lemma. The lemma follows from a well known theorem in the trust-region method (see Theorem 4 of [36] and Lemma 4.8 of [37]).

Lemma A.1. *If $\mathbf{s}_k$ satisfies (14) then*

$$\Phi(0)-\Phi(\mathbf{s}_k)\geqslant \beta_c\theta\left(1-\frac{1}{2}\theta\right)\|\hat{g}_k\|\min\left\{\Delta_k,\frac{\|\hat{g}_k\|}{\|h_k\|_\infty},\frac{\|\hat{g}_k\|}{\|\widehat{B}_k\|}\right\}, \tag{31}$$

where θ is the parameter in (13), β_c is the parameter in (14), $\hat{g}_k=D_kg_k$, $\widehat{B}_k=D_k\mathscr{K}^T\mathscr{K}D_k$, and $h_k\in\mathbb{R}^{N^2}$ is defined by

$$(h_k)_i=\frac{|(D_k^2g_k)_i|}{(\mathbf{f}_k)_i},\quad i=1,\ldots,N^2. \tag{32}$$

Proof. From

$$\Phi(-\tau D_k^2g_k)=-\tau g_k^TD_k^2g_k+\frac{1}{2}\tau^2g_k^TD_k^2\mathscr{K}^T\mathscr{K}D_k^2g_k,$$

we get that

$$\tau_k^\star=\frac{g_k^TD_k^2g_k}{g_k^TD_k^2\mathscr{K}^T\mathscr{K}D_k^2g_k}$$

is the minimizer of Φ along $-D_k^2g_k$. Because $\mathbf{s}_k^c$ is the minimizer of Φ along the direction $-D_k^2g_k$ in the trust region and feasible set, we consider the constraints $\|\tau D_kg_k\|\leqslant\Delta_k$ and $\mathbf{f}_k-\tau D_k^2g_k\geqslant 0$, which imply

$$0<\tau\leqslant T_k,$$

where T_k is defined by

$$T_k=\min\left\{\frac{\Delta_k}{\|D_kg_k\|},\min\left\{\frac{(\mathbf{f}_k)_i}{(D_k^2g_k)_i}:(g_k)_i>0\right\}\right\}. \tag{33}$$

It can be seen that

$$T_k\geqslant\min\left\{\frac{\Delta_k}{\|\hat{g}_k\|},\frac{1}{\|h_k\|_\infty}\right\}.$$

If $\tau_k^\star<T_k$, then

$$\Phi(\bar{\mathbf{s}}_k^c)=\Phi(\theta_k\mathbf{s}_k^c)=\Phi(-\theta_k\tau_k^\star D_k^2g_k)=-\theta_k\left(1-\frac{1}{2}\theta_k\right)\frac{(g_k^TD_k^2g_k)^2}{g_k^TD_k^2\mathscr{K}^T\mathscr{K}D_k^2g_k}.$$

If $\tau_k^\star\geqslant T_k$, i.e.,

$$g_k^TD_k^2\mathscr{K}^T\mathscr{K}D_k^2g_k\leqslant\frac{g_k^TD_k^2g_k}{T_k},$$

then

$$\begin{aligned}\Phi(\bar{\mathbf{s}}_k^c)=\Phi(\theta_k\mathbf{s}_k^c)=\Phi(-\theta_kT_kD_k^2g_k)&=-\theta_kT_kg_k^TD_k^2g_k+\frac{1}{2}\theta_k^2T_k^2g_k^TD_k^2\mathscr{K}^T\mathscr{K}D_k^2g_k\leqslant\theta_kT_k\left(-g_k^TD_k^2g_k+\frac{1}{2}\theta_kT_k\cdot\frac{g_k^TD_k^2g_k}{T_k}\right)\\&=-\theta_k\left(1-\frac{1}{2}\theta_k\right)T_kg_k^TD_k^2g_k.\end{aligned}$$

Thus we get that

$$\Phi(\bar{\mathbf{s}}_k^c)\leqslant-\theta_k\left(1-\frac{1}{2}\theta_k\right)g_k^TD_k^2g_k\min\left\{T_k,\frac{g_k^TD_k^2g_k}{g_k^TD_k^2\mathscr{K}^T\mathscr{K}D_k^2g_k}\right\}.$$

Because $\theta\leqslant\theta_k\leqslant 1$, we can get

$$\theta_k\left(1-\frac{1}{2}\theta_k\right)\geqslant\theta\left(1-\frac{1}{2}\theta\right).$$

Now from (14) we can obtain

$$\begin{aligned}\Phi(0)-\Phi(\mathbf{s}_k)&\geqslant\beta_c(\Phi(0)-\Phi(\bar{\mathbf{s}}_k^c))\geqslant\beta_c\theta_k\left(1-\frac{1}{2}\theta_k\right)g_k^TD_k^2g_k\min\left\{T_k,\frac{g_k^TD_k^2g_k}{g_k^TD_k^2\mathscr{K}^T\mathscr{K}D_k^2g_k}\right\}\\&\geqslant\beta_c\theta\left(1-\frac{1}{2}\theta\right)\|\hat{g}_k\|\min\left\{\Delta_k,\frac{\|\hat{g}_k\|}{\|h_k\|_\infty},\frac{\|\hat{g}_k\|}{\|\widehat{B}_k\|}\right\}.\end{aligned}$$

Hence we complete the proof. □

A.1. Proof of Theorem 2.1

Proof. We prove this theorem by contradiction. Assume that there is a $\epsilon > 0$ such that $\|\hat{g}_k\| \geqslant \epsilon$ for all sufficiently large k.

If the trial step $\mathbf{s}_k$ is accepted in Algorithm 2, we call the kth iteration a successful iteration, otherwise an unsuccessful iteration. So if there are only a finite number of successful iterations, $\Delta_{k+1} = \gamma_0 \Delta_k$ for all sufficiently large k and

$$\sum_{k=1}^{\infty} \Delta_k < +\infty \tag{34}$$

holds because $\gamma_0 \in (0,1)$.

If there is an infinite sequence $\{k_i\}$ of successful iterations, since $\{\Psi(\mathbf{f}_k)\}$ is non-increasing and bounded below,

$$0 \leqslant \sum_{k=0}^{\infty} (\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}_{k+1})) < +\infty.$$

From Lemma A.1 we can get for successful iteration k

$$\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}_{k+1}) = \Phi(0) - \Phi(\mathbf{s}_k) \geqslant \beta_c \theta \left(1 - \frac{1}{2}\theta\right) \|\hat{g}_k\| \min\left\{\Delta_k, \frac{\|\hat{g}_k\|}{\chi_h}, \frac{\|\hat{g}_k\|}{\chi_{\widehat{B}}}\right\}. \tag{35}$$

Hence we can obtain

$$\sum_{i=1}^{\infty} \Delta_{k_i} < +\infty.$$

Because $\Delta_{k+1} = \gamma_0 \Delta_k$ for an unsuccessful iteration and $\Delta_{k+1} \leqslant \gamma_2 \Delta_k$ for a successful iteration, we can get

$$\sum_{k=1}^{\infty} \Delta_k = \sum_{i=1}^{\infty} \Delta_{k_i} + \sum_{i=1}^{\infty} \sum_{j=1}^{k_{i+1}-k_i-1} \Delta_{k_i+j} \leqslant \sum_{i=1}^{\infty} \Delta_{k_i} + \sum_{i=1}^{\infty} (\gamma_2 + \gamma_0\gamma_2 + \gamma_0^2\gamma_2 + \cdots + \gamma_0^{k_{i+1}-k_i-2}\gamma_2)\Delta_{k_i} \leqslant \sum_{i=1}^{\infty} \left(1 + \frac{\gamma_2}{1-\gamma_0}\right)\Delta_{k_i} < +\infty.$$

Therefore,

$$\Delta_k \to 0, \quad \text{as } k \to \infty. \tag{36}$$

But from (16), (13) and

$$\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}_k + \mathbf{s}_k) = \Phi(0) - \Phi(\mathbf{s}_k),$$

we can get $\rho_k^f < 1 \leqslant \mu_2$ for sufficiently large k. Hence we can conclude from Algorithm 2 that $\{\Delta_k\}$ cannot converge to 0. It contradicts (36) and establishes the result. □

It should be noted that (AS.1) and (AS.2) imply that there exist positive scalars $\chi_D, \chi_{\widehat{B}}, \chi_{\hat{g}}, \chi_h$ such that

$$\|D_k\| \leqslant \chi_D, \quad \|\widehat{B}_k\| \leqslant \chi_{\widehat{B}}, \quad \|\hat{g}_k\| \leqslant \chi_{\hat{g}}, \quad \|h_k\|_\infty \leqslant \chi_h.$$

A.2. Proof of Theorem 2.2

Proof. The proof is by contradiction. Let $\epsilon_1 \in (0,1)$ be given and assume that there is a sequence $\{m_i\}$ such that $\|\hat{g}_{m_i}\| \geqslant \epsilon_1$. Theorem 2.1 guarantees that for any $\epsilon_2 \in (0, \epsilon_1)$ there is a subsequence of $\{m_i\}$ (without loss of generality we assume that it is the full sequence) and a sequence $\{l_i\}$ such that

$$\|\hat{g}_k\| \geqslant \epsilon_2, \quad m_i \leqslant k < l_i, \quad \|\hat{g}_{l_i}\| < \epsilon_2. \tag{37}$$

If the kth iteration is successful, then according to (35),

$$\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}_{k+1}) > \beta_c \theta \left(1 - \frac{1}{2}\theta\right) \epsilon_2 \min\left\{\Delta_k, \frac{\epsilon_2}{\chi_h}, \frac{\epsilon_2}{\chi_{\widehat{B}}}\right\}, \quad m_i \leqslant k < l_i.$$

Since $\Psi(\mathbf{f})$ is bounded below on $\mathscr{L}$ and the sequence $\{\Psi(\mathbf{f}_k)\}$ is non-increasing, $\{\Psi(\mathbf{f}_k)\}$ converges and $\{\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}_{k+1})\}$ converges to zero. From $\|\mathbf{f}_{k+1} - \mathbf{f}_k\| \leqslant \chi_D \Delta_k$, it follows that, for sufficiently large i,

$$\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}_{k+1}) \geqslant \epsilon_3 \|\mathbf{f}_{k+1} - \mathbf{f}_k\|, \quad m_i \leqslant k < l_i, \tag{38}$$

where $\epsilon_3 = \beta_c \theta (1 - \frac{1}{2}\theta)\epsilon_2 / \chi_D$. Using (38) and the triangle inequality,

$$\Psi(\mathbf{f}_{m_i}) - \Psi(\mathbf{f}_{k_i}) \geqslant \epsilon_3 \|\mathbf{f}_{k_i} - \mathbf{f}_{m_i}\|, \quad m_i \leqslant k_i \leqslant l_i.$$

The uniform continuity of $g(\mathbf{f})$ (since $\mathscr{L}$ is compact) and the convergence of $\{\Psi(\mathbf{f}_k)\}$ can now be used to deduce that

$$\|g_{m_i} - g_{l_i}\| \leqslant \epsilon_2 \tag{39}$$

for i sufficiently large.

Consider a subsequence of l_i (without loss of generality assume that it is the full sequence) such that $\{\mathbf{f}_{l_i}\}$ converges to $\mathbf{f}_\star$. Then $\{\mathbf{f}_{m_i}\}$ converges to $\mathbf{f}_\star$. We define $v(\mathbf{f})$ as the diagonal vector of $D(\mathbf{f})$. If the jth component of $g_\star$ is non-zero, then from the KKT condition (8), the jth component of $\mathbf{f}_{l_i}$ and $\mathbf{f}_{m_i}$ will approach to the bound. Assume that the upper bound is approached, then

$$(\mathbf{f}_{l_i})_j \to 0 \quad \text{and } (\mathbf{f}_{m_i})_j \to 0, \quad \text{as } i \to \infty.$$

Thus $\{\mathrm{diag}(v_{m_i} - v_{l_i})g_{l_i}\}$ converges to zero. Therefore, for i sufficiently large,

$$\|(D_{m_i} - D_{l_i})g_{l_i}\| = \|\mathrm{diag}(v_{m_i} - v_{l_i})g_{l_i}\| \leqslant \epsilon_2. \tag{40}$$

Using the triangle inequality for any m and l,

$$\|\hat{g}_m\| \leqslant \|D_m\|\|g_m - g_l\| + \|(D_m - D_l)g_l\| + \|\hat{g}_l\|. \tag{41}$$

Now combining (41) with (37), (39), and (40), we obtain that

$$\epsilon_1 \leqslant (\chi_D + 2)\epsilon_2.$$

Since ϵ_2 can be any number in $(0, \epsilon_1)$, this is a contradiction. □

References

[1] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, Institute of Physics Publishing, Bristol and Philadelphia, 1998.
[2] P.C. Hansen, Rank-deficient and Discrete Ill-posed Problems: Numerical Aspects of Linear Inversion, SIAM, Philadelphia, 1998.
[3] A.N. Tikhonov, V.Y. Arsenin, Solutions of Ill-posed Problems, John Wiley and Sons, New York, 1977.
[4] C.R. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, 2002.
[5] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer, Dordrecht, 1996.
[6] T.Y. Xiao, S.G. Yu, Y.F. Wang, Numerical Methods for Inverse Problems, Science Press, Beijing, 2003.
[7] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.
[8] C.R. Vogel, M.E. Oman, A fast, robust algorithm for total variation based reconstruction of noisy, blurred images, IEEE Transactions on Image Processing 7 (1998) 813–824.
[9] Y.W. Wang, Y.F. Wang, Y. Xue, W. Gao, A new algorithm for remotely sensed image texture classification and segmentation, International Journal of Remote Sensing 25 (19) (2004) 4043–4050.
[10] Y.F. Wang, S. Ma, H. Yang, J.D. Wang, X.W. Li, On the effective inversion by imposing a priori information for retrieval of land surface parameters, Science in China D 52 (2009) 540–549.
[11] Y.F. Wang, J.J. Cao, Y.X. Yuan, C.C. Yang, N.H. Xiu, Regularizing active set method for nonnegatively constrained ill-posed multichannel image restoration problem, Applied Optics 48 (2009) 1389–1401.
[12] M. Bertero, P. Boccacci, A simple method for the reduction of boundary effects in the Ricardson–Lucy approach to image deconvolution, Astronomy and Astrophysics 437 (2005) 369–374.
[13] M. Bertero, P. Boccacci, Application of the os-em method to the restoration of lbt images, Astronomy and Astrophysics Supplement Series 144 (2000) 181–186.
[14] M. Bertero, P. Boccacci, Image restoration methods for the large binocular telescope, Astronomy and Astrophysics Supplement Series 147 (2000) 323–332.
[15] R. Vio, J. Nagy, L. Tenorio, W. Wamsteker, A simple by efficient algorithm for multiple-image deblurring, Astronomy and Astrophysics 416 (2004) 403–410.
[16] M.K. Ng, R. Plemmons, A new approach to constrained total least squares image restoration, Linear Algebra and its Applications Volume 316 (1999) 237–258.
[17] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, 1994.
[18] Y.F. Wang, Computational Methods for Inverse Problems and Their Applications, Higher Education Press, Beijing, 2007.
[19] M. Hanke, J. Nagy, Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques, Inverse Problems 12 (1996) 157–173.
[20] Y.F. Wang, On the regularity of trust region-CG algorithm: with application to deconvolution problem, Science in China A 46 (2003) 312–325.
[21] Z.W. Wen, Y.F. Wang, A trust region method for large scale inverse problems in atmospheric image restoration, in: Y. Yuan (Ed.), Proceedings of the 2003' International Conference on Numerical Optimization and Numerical Linear Algebra, 2004, pp. 275–287.
[22] Z.W. Wen, Y.F. Wang, A new trust region algorithm for image restoration, Science in China A 48 (2005) 169–184.
[23] Y.F. Wang, Y.X. Yuan, Convergence and regularity of trust region methods for nonlinear ill-posed inverse problems, Inverse Problems 21 (2005) 821–838.
[24] M.A. Branch, T.F. Coleman, Y. Li, A subspace interior, and conjugate gradient method for large-scale bound-constrained minimization problems, SIAM Journal on Scientific Computing 21 (1) (1999) 1–23.
[25] S. Bellavia, B. Morini, Subspace trust-region methods for large bound-constrained nonlinear equations, SIAM Journal of Numerical Analysis 44 (2006) 1535–1555.
[26] Y.F. Wang, S. Ma, Projected Barzilai–Borwein method for large scale nonnegative image restoration, Inverse Problems in Science and Engineering 15 (6) (2007) 559–583.
[27] J. Bardsley, C.R. Vogel, A nonnegatively constrained convex programming method for image reconstruction, SIAM Journal on Scientific Computing 25 (4) (2003) 1326–1343.
[28] M. Hanke, J. Nagy, C.R. Vogel, Quasi-newton approach to nonnegative image restorations, Linear Algebra and its Applications 316 (2000) 223–236.
[29] J. Nagy, Z. Strakos, Enforcing nonnegativity in image reconstruction algorithms, in: David C. Wilson, Hemant D. Tagare, Fred L. Bookstein, Francoise J. Preteux, Edward R. Dougherty (Eds.), Mathematical Modeling, Estimation, and Imaging, vol. 4121, 2000, pp. 182–190.
[30] M. Rojas, T. Steihaug, An interior-point trust-region-based method for large-scale nonnegative regularization, Inverse Problems 18 (2002) 1291–1307.

[31] G. Landi, F. Zama, The active-set method for nonnegative regularization of linear ill-posed problems, Applied Mathematics and Computation 175 (2006) 715–729.
[32] S. Bellavia, M. Macconi, B. Morini, An affine scaling trust-region approach to bound-constrained nonlinear systems, Applied Numerical Mathematics 44 (2003) 257–280.
[33] S. Bellavia, B. Morini, An interior global method for nonlinear systems with simple bounds, Optimization Methods and Software 20 (2005) 1–22.
[34] T.F. Coleman, Y. Li, An interior trust region approach for nonlinear minimization subject to bounds, SIAM Journal on Optimization 6 (2) (1996) 418–445.
[35] C.B. Shaw Jr., Improvement of the resolution of an instrument by numerical solution of an integral equation, Journal of Mathematical Analysis and Applications 37 (1972) 83–112.
[36] M.J.D. Powell, Convergence properties of a class of minimization algorithms, in: O.L. Mangasarian, R.R. Meyer, S.M. Robinson (Eds.), Nonlinear Programming, Academic Press, New York, 1975, pp. 1–27.
[37] J.J. Moré, Recent developments in algorithms and software for trust region methods, in: A. Bachem, M. Grötschel, B. Korte (Eds.), Mathematical Programming: The State of Art, Springer, Berlin, 1983, pp. 258–287.