Performance of Block Preconditioners for Additive
Half-Quadratic Image Restoration Problems

Kyoum Sun Kim and Jae Heon Yun

Department of Mathematics, College of Natural Sciences
Chungbuk National University, Cheongju 362-763, Korea

Copyright © 2015 Kyoum Sun Kim and Jae Heon Yun. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Half-quadratic image restoration problems are often solved by minimizing a cost function which consists of a quadratic data-fidelity term and a weighted regularization term. In this paper, we consider the image restoration problem with an additive half-quadratic regularization term. Newton method is usually applied to solve the additive half-quadratic image restoration problem. At each iteration of Newton method, we need to solve a block $2 \times 2$ linear system whose coefficient matrix is a symmetric positive definite matrix. To solve this linear system, the preconditioned conjugate gradient (PCG) method with a suitable block preconditioner is usually used. We first propose a block SAOR (symmetric acceleration overrelaxation) preconditioner for the PCG method when solving the block $2 \times 2$ linear system, and then we study the problem of finding near optimal parameters which provide a fast convergence rate of the PCG method with the block SAOR preconditioner. Lastly, numerical experiments for several image restoration problems are provided to estimate the efficiency of the PCG method with the block SAOR preconditioner.

Mathematics Subject Classification: 49N45, 65F10, 65F22, 65K10, 94A08

Keywords: Image restoration, Deblurring, Half-quadratic regularization. Potential function, PCG method
1 Introduction

In this paper, we consider the image restoration problem of finding a restored image \( \hat{x} \in \mathbb{R}^p \) from a given degraded image \( b \in \mathbb{R}^q \) by minimizing a cost function \( J : \mathbb{R}^p \to \mathbb{R} \)

\[
\hat{x} = \arg \min_{x \in \mathbb{R}^p} J(x), \\
J(x) = \|Ax - b\|_2^2 + \beta \Phi(x),
\]

where \( \beta > 0 \) is a regularization parameter and \( A \in \mathbb{R}^{q \times p} \) is a blurring matrix. The first and second terms of (1) are called the data fidelity term and regularization term, respectively. The value of the parameter \( \beta \) effects the quality of the restored image obtained by minimizing the cost function \( J \). It is assumed that the degraded image \( b \) in the quadratic data-fidelity term is obtained by corrupting an original image \( x \) by a linear transformation, modeled by \( Ax + n = b \), where \( n \in \mathbb{R}^p \) is a noise vector. Using such a data-fidelity term is popular in computation of many inverse problems such as x-ray tomography and non-destructive evaluation [1, 3, 5, 12]. The regularization term \( \Phi \) in (1) is of the form

\[
\Phi(x) = \sum_{i=1}^{r} \phi(g_i^T x),
\]

where \( \phi : \mathbb{R} \to \mathbb{R} \) is a continuously differentiable potential function, and \( g_i^T : \mathbb{R}^p \to \mathbb{R}, i = 1, \ldots, r, \) are difference operators between pixel \((i,j)\) and its neighborhood. Generally, \( g_i^T x \) is the first or the second-order differences between the neighboring samples in \( x \), and \( g_i^T x \) can contain the edge information of the image \( x \). For example, if \( x \) is a one-dimensional signal, then \( g_i^T x = x_i - x_{i+1}, i = 1, \ldots, r \). Theoretical analysis of the image restoration approach using (1) and (2) has been widely studied by many researchers [2, 7, 9, 10].

Let \( G \) denote the \( r \times p \) matrix whose \( i \)th row is \( g_i^T \), that is \( G = [g_1, g_2, \ldots, g_r]^T \). In this paper, we assume that \( G \) is the discretization matrix of the first-order difference operator and

\[
A \neq 0, \ G \neq 0, \ \phi \neq 0, \ \ker(A^T A) \cap \ker(G^T G) = \{0\},
\]

where \( \ker(\cdot) \) denotes the kernel of the corresponding matrix. Clearly, these assumptions guarantee that \( \alpha_1 A^T A + \alpha_2 G^T G \) is a symmetric positive definite matrix when \( \alpha_1 > 0 \) and \( \alpha_2 > 0 \). We focus on convex, edge-preserving potential functions \( \phi : \mathbb{R} \to \mathbb{R} \) in (2) which can yield image estimates of high quality, involving edges and homogeneous regions. Such potential functions \( \phi \) can be found in [2, 4, 11, 10], and some examples of the popular potential functions
are

\[ \phi_1(t) = |t| - \alpha \log \left(1 + \frac{|t|}{\alpha}\right), \]

\[ \phi_2(t) = \sqrt{\alpha + t^2}, \]

\[ \phi_3(t) = \log(\cosh(\alpha t)), \]

\[ \phi_4(t) = |t|^\alpha, \quad 1 < \alpha \leq 2, \]

\[ \phi_5(t) = \begin{cases} \frac{t^2}{2}, & \text{if } |t| \leq \alpha, \\ \alpha|t| - \frac{\alpha^2}{2}, & \text{if } |t| > \alpha, \end{cases} \]

where \( \alpha > 0 \) is a given parameter whose value effects the choice of the parameter \( \beta \). We will consider that \( A^T A \) is invertible and/or \( \phi''(t) > 0, \quad \forall t \in \mathbb{R}, \) (5)

where \( \phi''(t) \) is the second order derivative of \( \phi(t) \) with respect to \( t \). The assumptions in (3) and (5) guarantee that for every \( x \in \mathbb{R}^p \), the function \( J \) has a strict unique minimum point [10].

However, computation of the minimizer \( \hat{x} \) of the cost functions \( J \) in (1) involving edge-preserving regularization term is quite complicated and costly because \( J \) is nonlinear with respect to \( x \). In order to speed up and simplify such computations, an additive or multiplicative half-quadratic reformulation of \( J \) was proposed for image restoration [6, 7]. In this paper, we only consider the additive half-quadratic reformulation of \( J \) whose main idea is to construct an augmented cost function \( \tilde{J} : \mathbb{R}^p \times \mathbb{R}^r \rightarrow \mathbb{R} \), involving additional variables \( z \in \mathbb{R}^r \), which is of the form

\[ \tilde{J}(x, z) = \|Ax - b\|_2^2 + \beta \sum_{i=1}^r \left( \frac{1}{2}(g_i^T x - z_i)^2 + \psi(z_i) \right), \] (6)

where

\[ \phi(t) = \min_{s \in \mathbb{R}} \left\{ \frac{1}{2}(t - s)^2 + \psi(s) \right\}, \quad \forall t \in \mathbb{R}, \] (7)

and \( \psi : \mathbb{R} \rightarrow \mathbb{R} \) is a prescribed dual potential function of the function \( \phi(\cdot) \) in (7). A dual potential function \( \psi(s) \) can be obtained from \( \phi(t) \) by using the convex duality relation. That is, \( \psi(s) \) is determined by

\[ \psi(s) = \sup_{t \in \mathbb{R}} \{-\frac{1}{2}(t - s)^2 + \phi(t)\}. \]

The condition (7) ensures that

\[ J(x) = \min_{z \in \mathbb{R}^r} \tilde{J}(x, z), \quad \forall x \in \mathbb{R}^p. \]
The minimizer of $\tilde{J}$ can be calculated by using an alternating minimization method. The main disadvantage of this method is that its convergence rate is only linear [10]. In order to speed up the convergence of the method, we apply Newton method to minimize the augmented cost function $\tilde{J}$. For this reason, we consider the Hessian of $\tilde{J}(x, z)$ which is given by

$$H(x, z) = \begin{bmatrix} 2A^T A + \beta G^T G & -\beta G^T \\ -\beta G & \beta I + \beta \text{diag}(\psi''(z)) \end{bmatrix} := \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{bmatrix},$$

(8)

where $I$ denotes the identity matrix, $\text{diag}(\cdot)$ is a diagonal matrix, and $\{z_i\}$ are the entries of the vector $z$. At each iteration of Newton method, we need to solve a structured linear system of the form

$$H(x, z) d = \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = c.$$

(9)

It can be easily shown that under the assumptions in (3) and $1 + \psi''(z) > 0$ for any $z \in \mathbb{R}$, the Hessian matrix $H(x, z)$ in (8) is symmetric positive definite for all $x$ and $z$. Hence the PCG (preconditioned conjugate gradient) method with a suitable preconditioners can be used to solve the linear system (9). Notice that the advantage of half-quadratic regularization has the effect of producing smoothing and edge-preserving image. The purpose of this paper is to evaluate performance of the PCG method with a block SAOR (symmetric acceleration overrelaxation) preconditioner when Newton method is applied to the problem of minimizing the augmented cost function $\tilde{J}$ in (6).

This paper is organized as follows. In Section 2, we propose a block SAOR preconditioner with two parameters $\omega$ and $r$ for the linear system (9) and analyze its properties. In Section 3, we study the problem of finding near optimal parameters $\omega$ and $r$ which provide a fast convergence rate of the PCG method with the block SAOR preconditioner when solving the linear system (9). In Section 4, we provide numerical results for several additive half-quadratic image restoration problems to estimate the efficiency of the PCG method with the block SAOR preconditioner. Lastly, some conclusions are drawn.

2 Block SAOR Preconditioner

In this section, we propose a block SAOR preconditioner corresponding to the block $2 \times 2$ linear system $H(x, z) d = c$. We first split the symmetric positive definite matrix $H(x, z)$ into its block lower-triangular part $L$, block diagonal part $D$ and block upper-triangular part $L^T$, i.e.,

$$H = D - L - L^T$$

(10)
with
\[
L = \begin{bmatrix} 0 & 0 \\ -H_{21} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \beta G & 0 \end{bmatrix},
\]
\[
D = \begin{bmatrix} H_{11} & 0 \\ 0 & H_{22} \end{bmatrix} = \begin{bmatrix} 2A^T A + \beta G^T G & 0 \\ 0 & \beta I + \beta \text{diag}(\psi''(z_i)) \end{bmatrix}.
\]

Then, the block SAOR preconditioner \( \tilde{P}(\omega, r) \) for the matrix \( H \) is defined by
\[
\tilde{P}(\omega, r) = \left( \tilde{D} - rL \right) \tilde{C}(\omega, r)^{-1} \left( \tilde{D} - rL \right)^T \tag{11}
\]
where
\[
\tilde{C}(\omega, r) = \omega \left( (2\tilde{D} - \omega D) + (\omega - r)(L + L^T) \right), \quad 0 < \omega, r < 2
\]
and \( \tilde{D} \) is a symmetric matrix which is an approximation to \( D \). When \( \omega = r \), the preconditioner \( \tilde{P}(\omega, r) \) is called the block SSOR preconditioner.

From now on, let \( \langle \cdot, \cdot \rangle \) denote the Euclidean inner product. For two symmetric matrices \( B \) and \( C \), \( C \succeq B \) (\( C \succ B \)) denotes that \( C - B \) is symmetric positive semi-definite (symmetric positive definite).

**Theorem 2.1** If \( \tilde{D} \succeq D \) and \( 0 < \omega \leq r < 2 \), then
\[
\tilde{C}(\omega, r) = \omega \left( (2\tilde{D} - \omega D) + (\omega - r)(L + L^T) \right)
\]
is a symmetric positive definite matrix.

**Proof.** It is clear that \( \tilde{C}(\omega, r) \) is symmetric. Also,
\[
\tilde{C}(\omega, r) = \omega \left( (2\tilde{D} - \omega D) + (\omega - r)(L + L^T) \right)
\geq \omega \left( (2 - \omega)D + (\omega - r)(L + L^T) \right)
\succ \omega \left( (r - \omega)D - (r - \omega)(L + L^T) \right)
= \omega(r - \omega)H
\]
Since \( H \) is symmetric positive definite and \( \omega \leq r \), \( \omega(r - \omega)H \succeq 0 \). Hence \( \tilde{C}(\omega, r) \) is symmetric positive definite.

**Corollary 2.2** If \( \tilde{D} = D \) and \( 0 < \omega \leq r < 2 \), then
\[
C(\omega, r) = \omega \left( (2 - \omega)D + (\omega - r)(L + L^T) \right)
\]
is a symmetric positive definite matrix.
In this paper, we only consider the case for $\tilde{D} = D$. From now on, let
\[ C(\omega, r) = \omega \left( (2 - \omega)D + (\omega - r)(L + L^T) \right) \]
and
\[ P(\omega, r) = (D - rL)C(\omega, r)^{-1}(D - rL)^T. \]

**Lemma 2.3** Let $H = D - L - L^T$ be a splitting of $H$ defined as in (10) and $0 < \omega \leq r < 2$. Let $U(\omega, r) = (1 - \omega)D + (\omega - r)L^T + \omega L$ and $N(\omega, r) = U(\omega, r)C(\omega, r)^{-1}U(\omega, r)^T$. Then $H = P(\omega, r) - N(\omega, r)$ and
\[ \max_{x \neq 0} \frac{\langle x, Hx \rangle}{\langle x, P(\omega, r)x \rangle} \leq 1. \]

**Proof.** By direct calculation, it is easy to show that $H = P(\omega, r) - N(\omega, r)$. Note that $P(\omega, r) \succ 0$ and $N(\omega, r) \succeq 0$. Since $\langle x, P(\omega, r)x \rangle = \langle x, Hx \rangle + \langle x, N(\omega, r)x \rangle \geq \langle x, Hx \rangle > 0$ for any nonzero vector $x$,
\[ \max_{x \neq 0} \frac{\langle x, Hx \rangle}{\langle x, P(\omega, r)x \rangle} \leq 1. \]

### 3 Convergence of the PCG with the block SAOR preconditioner

In this section, we study the problem of finding near optimal parameters $\omega$ and $r$ which provide a fast convergence rate of the PCG method with the block SAOR preconditioner $P(\omega, r)$ proposed in Section 2. For a square matrix $A$ with real eigenvalues, $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote the maximum and minimum eigenvalues of $A$, respectively. For a square matrix $B$ with positive eigenvalues, $\kappa(B) = \lambda_{\max}(B)\lambda_{\min}(B^{-1})$ denotes the condition number of $B$. Since convergence rate of the PCG method for solving the linear system (9) depends on the condition number of $P(\omega, r)^{-1}H$, we provide the following lemma for its condition number.

**Lemma 3.1** Let $0 < \omega \leq r < 2$. Then
\[ \kappa \left( P(\omega, r)^{-1}H \right) = \mu_1 \mu_2 \leq \mu_1, \]
where $\mu_1 = \max_{x \neq 0} \frac{\langle x, P(\omega, r)x \rangle}{\langle x, Hx \rangle}$ and $\mu_2 = \max_{x \neq 0} \frac{\langle x, Hx \rangle}{\langle x, P(\omega, r)x \rangle}$.

**Proof.** Since $H$ and $P(\omega, r)$ are symmetric positive definite,
\[
\mu_1 = \max_{x \neq 0} \frac{\langle x, P(\omega, r)x \rangle}{\langle H^{1/2}x, H^{1/2}x \rangle} = \max_{y \neq 0} \frac{\langle y, H^{-1/2}P(\omega, r)H^{-1/2}y \rangle}{\langle y, y \rangle} = \lambda_{\max} \left( H^{-1/2}P(\omega, r)H^{-1/2} \right) = \lambda_{\max} \left( H^{-1}P(\omega, r) \right),
\]
where \( y = H^{1/2}x \). Similarly, one obtains

\[
\mu_2 = \max_{y \neq 0} \frac{\langle y, P(\omega, r)^{-1/2}HP(\omega, r)^{-1/2}y \rangle}{\langle y, y \rangle} = \lambda_{\max}(P(\omega, r)^{-1}H),
\]

where \( y = P(\omega, r)^{1/2}x \). Hence, we obtain

\[
\kappa(P(\omega, r)^{-1}H) = \lambda_{\max}(P(\omega, r)^{-1}H) \lambda_{\max}(H^{-1}P(\omega, r)) = \mu_1\mu_2.
\]

Since \( \mu_2 \leq 1 \) from Lemma 2.3, \( \kappa(P(\omega, r)^{-1}H) = \mu_1\mu_2 \leq \mu_1 \). \( \square \)

From Lemma 3.1, it can be easily seen that we need to find \( \omega \) and \( r \) which minimize \( \mu_1 = \lambda_{\max}(H^{-1}P(\omega, r)) \) instead of minimizing \( \kappa(P(\omega, r)^{-1}H) \). For a nonzero real vector \( u \), let

\[
\begin{align*}
\beta_u &= \frac{u^TL^Tu}{u^THu}, & \beta &= \max_{u \neq 0} \frac{u^TL^Tu}{u^THu} \geq 0, \\
\gamma_u &= \frac{u^THD^{-1}Hx}{u^THu}, & \gamma &= \min_{u \neq 0} \frac{u^THD^{-1}Hx}{u^THu} > 0, & (12) \\
\delta_u &= \frac{u^THD^{-1}HD^{-1}Hu}{u^THu}, & \delta &= \max_{u \neq 0} \frac{u^THD^{-1}HD^{-1}Hu}{u^THu} > 0.
\end{align*}
\]

**Lemma 3.2** Let \( \beta, \gamma \) and \( \delta \) be defined as in (12). Then

\[
2\gamma\beta + \gamma = 1, \quad \gamma = \lambda_{\min}(D^{-1}H), \quad \delta = \lambda_{\max}^2(D^{-1}H), \quad \text{and} \quad \beta > 0.
\]

**Proof.** Let \( u \) be a nonzero real vector. Since \( u^TL^Tu = \frac{1}{2}u^T(L + L^T)u = \frac{1}{2}u^T(D-H)u \), one obtains

\[
\beta = \max_{u \neq 0} \frac{u^TL^Tu}{u^THu} = \frac{1}{2} \left( \max_{u \neq 0} \frac{u^TDu}{u^THu} - 1 \right) = \frac{1}{2} \left( \lambda_{\max}(H^{-1/2}DH^{-1/2}) - 1 \right), \quad (13)
\]

On the other hand, it can be easily shown that

\[
\gamma = \lambda_{\min}(H^{1/2}D^{-1}H^{1/2}) = \lambda_{\min}(D^{-1}H) = \frac{1}{\lambda_{\max}(H^{-1}D)}. \quad (14)
\]

From (13) and (14), \( \beta = \frac{1}{2}(\frac{1}{\gamma} - 1) \), which is equivalent to \( 2\gamma\beta + \gamma = 1 \). From the definition of \( \delta \) in (12), it can be also shown that

\[
\delta = \lambda_{\max}(H^{1/2}D^{-1}HD^{-1}H^{1/2}) = \lambda_{\max}(D^{-1}HD^{-1}H) = \lambda_{\max}^2(D^{-1}H).
\]
From the splitting of $H$ given in (10), it is easy to choose a nonzero vector $u$ such that $u^T L^T u > 0$. Hence, it is clear that $\beta > 0$.

Let $\lambda_i$ and $u_i$ be an eigenvalue and a corresponding eigenvector of $H^{-1}P(\omega, r)$ for each $i = 1, 2, \ldots, (p + r)$, i.e., $H^{-1}P(\omega, r)u_i = \lambda_i u_i$, where $\lambda_i > 0$ and $u_i \in \mathbb{R}^{p+r}$ is a nonzero vector. Since $P(\omega, r)u_i = \lambda_i H u_i$ and $(D - rL)^{-1} = D^{-1} + rD^{-1}LD^{-1}$, one obtains

\[
(D - rL)^T u_i = \lambda_i \omega \left( (2 - r)D + (r - \omega)H \right)(D - rL)^{-1} H u_i
\]

\[
= \lambda_i \omega \left( (2 - r)I + (r - \omega)HD^{-1} \right)(H + rLD^{-1}H) u_i.
\]

\[
\approx \lambda_i \omega \left( (2 - r)H + (r - \omega)HD^{-1}H + r(r - \omega)HD^{-1}LD^{-1}H \right) u_i.
\]

We first consider the case for $r > \omega$. Premultiplying both sides of the third equation of (15) by $u_i^T$,

\[
u_i^T D u_i - r u_i^T L^T u_i
\]

\[
\approx \lambda_i \omega u_i^T \left( (2 - r)H + (r - \omega)HD^{-1}H + r(r - \omega)HD^{-1}LD^{-1}H \right) u_i.
\]

Since $u_i^T D u_i = u_i^T (H + L + L^T) u_i = u_i^T H u_i + 2u_i^T L^T u_i$, from (16)

\[
u_i^T H u_i + (2 - r)u_i^T L^T u_i
\]

\[
\approx \lambda_i \omega u_i^T \left( (2 - r)H + (r - \omega)HD^{-1}H + r(r - \omega)HD^{-1}LD^{-1}H \right) u_i.
\]

Since $u_i^T H D^{-1} LD^{-1} H u_i = \frac{1}{2}(u_i^T H D^{-1} H u_i - u_i^T H D^{-1} H u_i)$, from (12) and (17) one obtains

\[
1 + (2 - r)\beta_{u_i} \approx \lambda_i \omega \left( (2 - r) + \left( r - \omega \right) \left( 1 + \frac{r}{2} \right) \gamma_{u_i} - \frac{r \left( \omega - \omega \right)}{2} \delta_{u_i} \right)
\]

\[
\lambda_i \approx \frac{1 + (2 - r)\beta_{u_i}}{\omega \left( (2 - r) + \left( r - \omega \right) \left( 1 + \frac{r}{2} \right) \gamma_{u_i} - \frac{r \left( \omega - \omega \right)}{2} \delta_{u_i} \right)}.
\]

Since $\mu_1 = \max_i \lambda_i$, from (12) and (18) we have the following relation

\[
\mu_1 \approx \max_{u_i} \frac{1 + (2 - r)\beta_{u_i}}{\omega \left( (2 - r) + \left( r - \omega \right) \left( 1 + \frac{r}{2} \right) \gamma_{u_i} - \frac{r \left( \omega - \omega \right)}{2} \delta_{u_i} \right)}
\]

\[
\leq \max_{u \neq 0} \frac{1 + (2 - r)\beta_u}{\omega \left( (2 - r) + \left( r - \omega \right) \left( 1 + \frac{r}{2} \right) \gamma_u - \frac{r \left( \omega - \omega \right)}{2} \delta_u \right)} \tag{19}
\]

\[
\leq \frac{1 + (2 - r)\beta}{\omega \left( (2 - r) + \left( r - \omega \right) \left( 1 + \frac{r}{2} \right) \gamma - \frac{r \left( \omega - \omega \right) \delta}{2} \right)} \equiv f(\omega, r).
\]

From Equation (19), the problem of finding optimal parameters $\omega$ and $r$ which minimize the $\mu_1$ may be viewed as that of finding optimal parameters $\omega$ and $r$.
which minimize the upper bound $f(\omega, r)$. Hence we need to find $\omega$ and $r$ such that \[ \frac{\partial f}{\partial \omega} = 0 \] and \[ \frac{\partial f}{\partial r} = 0. \] By direct calculation, \[
\frac{\partial f}{\partial \omega} = -\frac{(1 + (2 - r)\beta) \left( (2\omega - r)((\delta - \gamma)\frac{r}{2} - \gamma) + 2 - r \right)}{\omega^2 \left( (2 - r) + (r - \omega)(\gamma + \frac{\gamma}{2}r - \frac{\gamma}{2}r) \right)^2},
\frac{\partial f}{\partial r} = \frac{(\delta - \gamma)r(1 + 2\beta - \frac{\delta}{2}r) + \omega(-\beta\delta + 2\beta\gamma + \frac{\gamma}{2}r - \frac{\delta}{2}) + 1 - \gamma - 2\beta\gamma}{\omega \left( (2 - r) + (r - \omega)(\gamma + \frac{\gamma}{2}r - \frac{\gamma}{2}r) \right)^2}.\]

Since $0 < r < 2$, \[ \frac{\partial f}{\partial \omega} = 0 \] implies that \[
\omega = r + \frac{r - 2}{(\delta - \gamma)r - 2\gamma}. \quad (20)
\]

Since $\beta = \frac{1 - \gamma}{2\gamma}$ from Lemma 3.2, \[ \frac{\partial f}{\partial r} = 0 \] implies that \[
(\delta - \gamma)r \left( \frac{4 - (1 - \gamma)r}{4\gamma} \right) + \omega \left( -\delta + 2\gamma - \frac{\gamma^2}{2\gamma} \right) = 0 \quad (21)
\]

Substituting (20) into (21), the following cubic equation for $r$ is obtained \[
(\delta - \gamma)^2(\gamma - 1)r^3 + 3(\delta - \gamma)(\delta - \gamma^2)r^2 + 2(\gamma^3 + 2\gamma^2 - 2\gamma - 3\gamma - 3\delta - \delta)r - 4(2\gamma - \gamma^2 - \delta) = 0. \quad (22)
\]

From (20) and (22), we can obtain the following theorem.

**Theorem 3.3** Let $r > \omega$. Then the near optimal parameters $\omega$ and $r$ which provide a fast convergence rate of the PCG method with the block SAOR pre-conditioner $P(\omega, r)$ can be determined as follows: First find a larger solution $r \in (0, 2)$ satisfying the cubic equation (22), and then find $\omega$ using (20).

We next consider the case for $r = \omega$. Since $\omega = r$, from (19) we obtain \[
g(\omega) \equiv f(\omega, \omega) = \frac{1 + (2 - \omega)\beta}{\omega(2 - \omega)}.\]

Now we choose $\omega$ which minimizes $g(\omega)$. The derivative of $g(\omega)$ is given by \[
g'(\omega) = -\frac{\beta\omega^2 - 2(1 + 2\beta)\omega + 4\beta + 2}{\omega^2(2 - \omega)^2}.\]

Hence, $g'(\omega) = 0$ has two roots $\omega = \frac{1 + 2\beta + \sqrt{1 + 2\beta}}{\beta}$. It is easy to show that $g(\omega)$ has a minimum at \[
\omega = 1 + 2\beta - \frac{\sqrt{1 + 2\beta}}{\beta}. \quad (23)
\]

Note that $0 < \frac{1 + 2\beta - \sqrt{1 + 2\beta}}{\beta} < 2$ since $\beta > 0$. From these arguments, we obtain the following theorem.
Theorem 3.4  Let $r = \omega$. Then the near optimal parameter $\omega$ which provides a fast convergence rate of the PCG method with the block SSOR preconditioner $P(\omega, r)$ is given by

$$\omega = \frac{2}{1 + \sqrt{\gamma}} = \frac{2}{1 + \sqrt{\lambda_{\min}(D^{-1}H)}}.$$ 

Proof. From Lemma 3.2, $\gamma = \lambda_{\min}(D^{-1}H)$, $1 + 2\beta = \frac{1}{\gamma}$ and $\beta = \frac{1 - \gamma}{2\gamma}$. Substituting these equations into (23), one obtains $\omega = \frac{2}{1 + \sqrt{\gamma}} = \frac{2}{1 + \sqrt{\lambda_{\min}(D^{-1}H)}}$.

4 Numerical experiments

In this section, we provide numerical experiments to estimate the efficiency of the PCG method with the block SAOR preconditioner when Newton method is applied to the problem of solving the additive half-quadratic image restoration problems. For all test runs, we have used a self-developed PCG method to minimize the use of memory since the built-in function `pcg` in Matlab does not run because of too much memory.

For numerical experiments, we have used the following damped Newton method

$$\begin{bmatrix} x^{(k+1)} \\ z^{(k+1)} \end{bmatrix} = \begin{bmatrix} x^{(k)} \\ z^{(k)} \end{bmatrix} - \gamma_k H(x^{(k)}, z^{(k)})^{-1}\nabla \tilde{J}(x^{(k)}, z^{(k)}), \quad k = 0, 1, 2, \ldots,$$

where $\gamma_k$ is the step size which is determined by the backtracking line search procedure. The PCG method with the block SAOR preconditioner is applied to solve the linear systems $H(x, z)d = c$, and the main computational cost of the PCG method is to solve the linear systems with the coefficient matrix $P(\omega, r)$ which is defined in Section 2.

All numerical tests have been performed using Matlab R2014a on a personal computer with 3.40GHz CPU and 8.00GB memory. For all experiments, the initial vector $x^{(0)}$ is the observed image and $z^{(0)}$ is a vector whose $i$th component is chosen by the formula for $s$ in (25), where $t = g_i^T x^{(0)}$. Average and Gaussian point spread functions (PSF) of size $7 \times 7$ are used to blur images, which can be done using the built-in Matlab function `fspecial`. Also noises are added using Gaussian white noise with standard deviations 0.001 or 0.005, which can be done using Matlab function `randn`, i.e. $E = 0.001 \times \text{randn}(m, n)$ or $E = 0.005 \times \text{randn}(m, n)$, where $(m, n)$ is the size of the original image.

The stopping criterion for the outer Newton iteration is

$$\frac{\| \nabla \tilde{J}(x^{(k)}, z^{(k)}) \|_2}{\| \nabla \tilde{J}(x^{(0)}, z^{(0)}) \|_2} \leq 10^{-9}.$$
and the stopping criterion for the PCG method at the $k$-th Newton iterate is

$$\frac{\|r(k, l)\|_2}{\|r(k, 0)\|_2} \leq 10^{-6},$$

where $r(k, l)$ represents the $l$-th residual vector generated at the $l$-th iteration of PCG method with $r(k, 0)$ the initial residual vector. The accuracy of restored images is measured by the PSNR (peak signal to noise ratio) and the RMSE (root mean square error) which are defined by

$$\text{PSNR} = 10 \log_{10} \left( \frac{\max_{i,j} |f_{ij}|^2 \cdot m \cdot n}{\|f - g\|_F^2} \right), \quad \text{RMSE} = \frac{\|f - g\|_F}{\|f\|_F},$$

$$\text{PSNR}_0 = 10 \log_{10} \left( \frac{\max_{i,j} |f_{ij}|^2 \cdot m \cdot n}{\|f - \hat{f}\|_F^2} \right), \quad \text{RMSE}_0 = \frac{\|f - \hat{f}\|_F}{\|f\|_F},$$

where $f = (f_{ij})$ is a true image, $\hat{f} = (\hat{f}_{ij})$ is a blurred and noisy image of $f$, $g$ is a restored image for $\hat{f}$, and $\| \cdot \|_F$ denotes the Frobenius norm.

We have used the “Cameraman”, “Pepper” and “Lily” images for numerical experiments. The $\phi_1(t)$ in (4) is used as an edge-preserving potential function, and $G$ is chosen as the scaled first order finite difference operator of the form

$$G = \left( \begin{array}{c} g_1^T \\ g_2^T \\ \vdots \\ g_r^T \end{array} \right) = \frac{1}{255} \left( \begin{array}{ccc} 1 & -1 & 0 \\ & 1 & -1 \\ \ddots & \ddots & \ddots \\ 0 & 1 & -1 \end{array} \right)_{r \times p}$$

where $r = p - 1$. Then, the dual potential function $\psi : \mathbb{R} \to \mathbb{R}$ corresponding to $\phi_1(t)$ is given by

$$\psi(s) = |t| - \alpha \log \left( 1 + \frac{|t|}{\alpha} \right) - \frac{1}{2} \left( \frac{t}{|t| + \alpha} \right)^2, \quad s = t - \frac{t}{|t| + \alpha}.$$ (25)

From (25), one easily obtains

$$\psi'(s) = \frac{t}{|t| + \alpha}, \quad \psi''(s) = \frac{\alpha}{(|t| + \alpha)^2 - \alpha}.$$

Notice that $t = g_i^T x$ is used for $i = 1, 2, \ldots, r$, and

$$\nabla \tilde{J}(x, z) = \begin{bmatrix} 2A^T(Ax - b) + \beta G^T(Gx - z) \\ -\beta(g_1^T x - z_1) + \beta(\psi'(z_1)) \\ \vdots \\ -\beta(g_r^T x - z_r) + \beta(\psi'(z_r)) \end{bmatrix},$$ (26)
where $A$ is a blurring matrix in $\mathbb{R}^{q \times p}$, $x \in \mathbb{R}^p$ and $z = (z_i) \in \mathbb{R}^r$. For all test problems, we have used two kinds of boundary conditions which are zero or reflexive, and we have chosen $\alpha = 1.0$ for $\phi_1(t)$ and $\beta = 0.001$ as a regularization parameter.

Tables 1 to 6 provide numerical results for three image restoration problems, where the column labeled with “NIT” represents the number of outer iterations of the damped Newton method and “PIT” represents the total number of iterations used in the PCG method, “CPU(s)” represents the elapsed time in seconds for the overall iteration process. In all Tables and Figures, “σ” and “BC” denote standard deviation of the Gaussian white noise and boundary condition being used, respectively.

In order to evaluate performance of the block SAOR preconditioner, numerical experiments are carried out for many different values of $\omega$ and $r$. We also tried to compute near optimal parameters using the formulas stated in Theorems 3.3 and 3.4, but it takes too much time to compute the largest or smallest eigenvalues of the matrix $D^{-1}H$ because of its large order. Moreover, the formula in Theorem 3.3 requires a solution of a nonlinear cubic equation, which is also time-consuming. For this reason, we have computed only a near optimal parameter $\omega$ in Theorem 3.4 using the leading principal submatrices of $H_{11}$, $H_{12}$ and $H_{22}$ whose row and column sizes are $\frac{1}{64}$ of the original sizes. The $\omega_o$ in Tables 1 to 6 refers to a near optimal parameter computed by the formula in Theorem 3.4 using the leading principal submatrices of $H_{11}$, $H_{12}$ and $H_{22}$. For test problems used in this paper, $\omega_o$ ranges between 1.01 and 1.12.

Tables 1 and 2 provide numerical results for the Cameraman image with size of $256 \times 256$, and Tables 3 and 4 provide numerical results for the Pepper image with size of $256 \times 256$. The original images are blurred by an average blur with size $7 \times 7$ and the Gaussian white noise with $\sigma = 0.001$ being added. The experimentally optimal parameters $\omega$ and $r$ which provide good performance are $(\omega, r) = (1.0, 1.2), (0.9, 0.9), (0.9, 1.1), (1.1, 1.1)$ or $(\omega_o, \omega_o)$. Figure 1 shows the restored images for the Cameraman image, and Figure 2 shows the restored images for the Pepper image.

Tables 5 and 6 provide numerical results for the Lily image with size of $186 \times 230$. The original image is blurred by a Gaussian blur of size $7 \times 7$ with standard deviation 2 and the Gaussian white noise with $\sigma = 0.005$ being added. The experimentally optimal parameters $\omega$ and $r$ which provide good performance are $(\omega, r) = (1.0, 1.2), (0.9, 1.1), (0.9, 0.9), (1.1, 1.1)$ or $(\omega_o, \omega_o)$. Figure 3 shows the restored images for the Lily image.

From Figures 1 to 3, it can be seen that the additive half-quadratic regularization technique tends to sharpen the edges as well as restoring the blurred and noisy image. From Tables 1 to 6, it can be seen that the PCG with the block SAOR preconditioner performs well and the number of outer Newton iterations is 2 for all cases. It is important to find optimal parameters $\omega$ and
Table 1: Numerical results for the image restoration of Cameraman.

<table>
<thead>
<tr>
<th>ω</th>
<th>r</th>
<th>PSNR</th>
<th>RMSE</th>
<th>NIT</th>
<th>PIT</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.8</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>7</td>
<td>35.2</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>6</td>
<td>30.1</td>
</tr>
<tr>
<td>1.0</td>
<td>1.2</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>5</td>
<td>25.6</td>
</tr>
<tr>
<td>1.3</td>
<td>1.5</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>6</td>
<td>30.3</td>
</tr>
<tr>
<td>1.5</td>
<td>1.7</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>7</td>
<td>35.2</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>8</td>
<td>40.2</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>7</td>
<td>35.4</td>
</tr>
<tr>
<td>1.3</td>
<td>1.3</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>7</td>
<td>35.5</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>8</td>
<td>40.4</td>
</tr>
<tr>
<td>1.7</td>
<td>1.7</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>8</td>
<td>40.0</td>
</tr>
<tr>
<td>ω₀</td>
<td>ω₀</td>
<td>50.15</td>
<td>0.0059</td>
<td>2</td>
<td>5</td>
<td>26.2</td>
</tr>
</tbody>
</table>

Average blur, reflexive BC, noise with σ = 0.001
PSNR₀ = 21.1511, RMES₀ = 0.1653

Table 2: Numerical results for the image restoration of Cameraman.

<table>
<thead>
<tr>
<th>ω</th>
<th>r</th>
<th>PSNR</th>
<th>RMSE</th>
<th>NIT</th>
<th>PIT</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.8</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>9</td>
<td>45.4</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>8</td>
<td>40.3</td>
</tr>
<tr>
<td>1.0</td>
<td>1.2</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>6</td>
<td>30.3</td>
</tr>
<tr>
<td>1.3</td>
<td>1.5</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>7</td>
<td>35.2</td>
</tr>
<tr>
<td>1.5</td>
<td>1.7</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>8</td>
<td>40.0</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>9</td>
<td>45.3</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>8</td>
<td>40.2</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>6</td>
<td>29.9</td>
</tr>
<tr>
<td>1.3</td>
<td>1.3</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>8</td>
<td>40.1</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>9</td>
<td>45.1</td>
</tr>
<tr>
<td>ω₀</td>
<td>ω₀</td>
<td>40.70</td>
<td>0.0174</td>
<td>2</td>
<td>6</td>
<td>31.2</td>
</tr>
</tbody>
</table>

Average blur, zero BC, noise with σ = 0.001
PSNR₀ = 20.8302, RMES₀ = 0.1715
Table 3: Numerical results for the image restoration of Pepper.

<table>
<thead>
<tr>
<th>ω</th>
<th>r</th>
<th>PSNR</th>
<th>RMSE</th>
<th>NIT</th>
<th>PIT</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.8</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>7</td>
<td>34.1</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>7</td>
<td>35.0</td>
</tr>
<tr>
<td>0.9</td>
<td>1.1</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>5</td>
<td>25.0</td>
</tr>
<tr>
<td>1.3</td>
<td>1.5</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>6</td>
<td>30.2</td>
</tr>
<tr>
<td>1.5</td>
<td>1.7</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>7</td>
<td>34.5</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>8</td>
<td>40.3</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>7</td>
<td>34.6</td>
</tr>
<tr>
<td>1.1</td>
<td>1.1</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>5</td>
<td>25.5</td>
</tr>
<tr>
<td>1.3</td>
<td>1.3</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>7</td>
<td>34.6</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>51.48</td>
<td>0.0051</td>
<td>2</td>
<td>7</td>
<td>35.0</td>
</tr>
</tbody>
</table>

Average blur, reflexive BC, noise with σ= 0.001
PSNR₀ = 21.9036, RMES₀ = 0.1521

Table 4: Numerical results for the image restoration of Pepper.

<table>
<thead>
<tr>
<th>ω</th>
<th>r</th>
<th>PSNR</th>
<th>RMSE</th>
<th>NIT</th>
<th>PIT</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.8</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>10</td>
<td>49.4</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>9</td>
<td>45.1</td>
</tr>
<tr>
<td>1.0</td>
<td>1.2</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>7</td>
<td>34.7</td>
</tr>
<tr>
<td>1.3</td>
<td>1.5</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>9</td>
<td>44.8</td>
</tr>
<tr>
<td>1.5</td>
<td>1.7</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>9</td>
<td>44.9</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>11</td>
<td>54.1</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>9</td>
<td>44.3</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>7</td>
<td>34.9</td>
</tr>
<tr>
<td>1.2</td>
<td>1.2</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>9</td>
<td>45.1</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>38.84</td>
<td>0.0216</td>
<td>2</td>
<td>11</td>
<td>54.4</td>
</tr>
</tbody>
</table>

Average blur, zero BC, noise with σ=0.001
PSNR₀ = 22.4396, RMES₀ = 0.143
Table 5: Numerical results for the image restoration of Lily.

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$r$</th>
<th>PSNR</th>
<th>RMSE</th>
<th>NIT</th>
<th>PIT</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.7</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>9</td>
<td>24.9</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>8</td>
<td>22.0</td>
</tr>
<tr>
<td>0.9</td>
<td>1.1</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>6</td>
<td>16.6</td>
</tr>
<tr>
<td>1.3</td>
<td>1.5</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>7</td>
<td>19.3</td>
</tr>
<tr>
<td>1.5</td>
<td>1.7</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>8</td>
<td>22.2</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>9</td>
<td>24.6</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>6</td>
<td>16.7</td>
</tr>
<tr>
<td>1.1</td>
<td>1.1</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>6</td>
<td>16.5</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>9</td>
<td>24.7</td>
</tr>
<tr>
<td>1.7</td>
<td>1.7</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>10</td>
<td>27.2</td>
</tr>
<tr>
<td>$\omega_o$</td>
<td>$\omega_o$</td>
<td>32.05</td>
<td>0.0501</td>
<td>2</td>
<td>6</td>
<td>18.1</td>
</tr>
</tbody>
</table>

Gaussian blur, reflexive BC, noise with $\sigma=0.005$
PSNR$_0 = 21.3856$, RMES$_0 = 0.1709$

---

Table 6: Numerical results for the image restoration of Lily.

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$r$</th>
<th>PSNR</th>
<th>RMSE</th>
<th>NIT</th>
<th>PIT</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.7</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>9</td>
<td>24.5</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>8</td>
<td>23.2</td>
</tr>
<tr>
<td>1.0</td>
<td>1.2</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>6</td>
<td>17.0</td>
</tr>
<tr>
<td>1.3</td>
<td>1.5</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>7</td>
<td>19.8</td>
</tr>
<tr>
<td>1.5</td>
<td>1.7</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>8</td>
<td>22.7</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>8</td>
<td>23.0</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>6</td>
<td>17.3</td>
</tr>
<tr>
<td>1.1</td>
<td>1.1</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>6</td>
<td>16.8</td>
</tr>
<tr>
<td>1.3</td>
<td>1.3</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>8</td>
<td>22.5</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>9</td>
<td>25.6</td>
</tr>
<tr>
<td>$\omega_o$</td>
<td>$\omega_o$</td>
<td>31.92</td>
<td>0.0508</td>
<td>2</td>
<td>6</td>
<td>18.0</td>
</tr>
</tbody>
</table>

Gaussian blur, zero BC, noise with $\sigma=0.005$
PSNR$_0 = 21.2187$, RMES$_0 = 0.1743$
of the block SAOR preconditioner which yields the best performance of the PCG. However this problem is too difficult, so we have studied the problem of finding near optimal parameters in this paper. Numerical experiments show that the PCG method with near optimal parameters chosen as in Theorem 3.4 using the leading principal submatrices of $H_{11}$, $H_{12}$ and $H_{22}$ performs as well as that with the experimentally optimal parameters (see Tables 1 to 6).

5 Conclusion

In this paper, we considered the image restoration problem with an additive half-quadratic regularization term. The additive half-quadratic image restoration problem is usually solved by Newton method incorporated with the PCG method. We first proposed a block SAOR (symmetric acceleration overrelaxation) preconditioner for the PCG method, and then we proposed a formula for finding near optimal parameters which provide a fast convergence rate of the PCG method with the block SAOR preconditioner. However, the formula for finding near optimal parameters is not useful in practical applications since it requires too much CPU time. For this reason, we proposed another technique for finding near optimal parameters, which uses the leading principal submatrices of $H_{11}$, $H_{12}$ and $H_{22}$ whose row and column sizes are $\frac{1}{64}$ of the original sizes. Numerical experiments show that the PCG with near optimal parameters which are chosen using the leading principal submatrices performs as well as that with the experimentally optimal parameters (see Tables 1 to 6). It means that the technique using the leading principal submatrices works well. Observe that the number of outer Newton iterations is 2 for all numerical experiments. This means that Newton method performs very well when the PCG method with the block SAOR preconditioner is used at each iteration of Newton method.

Acknowledgements. This work was supported by the research grant of the Agency for Defense Development and the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by Ministry of Education(NRF-2013R1A1A2005722).

References

Figure 1: Example of cameraman image with $256 \times 256$ size (Average blur, noise with $\sigma=0.001$, (b) and (c) : reflexive BC, (d) and (e) : zero BC).
Figure 2: Example of pepper image with $256 \times 256$ size (Average blur, noise with $\sigma=0.001$, (b) and (c) : reflexive BC, (d) and (e) : zero BC).
Figure 3: Example of lily image with $186 \times 230$ size (Gaussian blur, noise with $\sigma=0.005$, (b) and (c) : reflexive BC, (d) and (e) : zero BC).


Received: June 22, 2015; Published: July 29, 2015