Far East Journal of Mathematical Sciences (FJMS)

http://www.pphmj.com
http://dx.doi.org/10.17654/MS102061115
Volume 102, Number 6, 2017, Pages 1115-1138 **ISSN: 0972-0871**

PARALLEL IMAGE DENOISING USING Gℓ-PCG METHOD WITH KRONECKER PRODUCT PRECONDITIONER

Jae Heon Yun

Department of Mathematics
College of Natural Sciences
Chungbuk National University
Cheongju 28644, Korea

Abstract

We first show how to construct the linear operator equations corresponding to Tikhonov regularization problems for solving image denoising problem, and then we propose how to apply the global preconditioned conjugate gradient (Gℓ-PCG) method with Kronecker product preconditioners to the linear operator equations. Next, we propose a coarse-grained parallel image denoising algorithm using the Gℓ-PCG with Kronecker product preconditioner. Finally, we provide numerical experiments for several image denoising problems to evaluate the effectiveness of the Gℓ-PCG with Kronecker product preconditioner.

Received: April 25, 2017; Accepted: June 25, 2017
2010 Mathematics Subject Classification: 94A08, 65F22, 65K10, 65F10, 68W10.
Keywords and phrases: Gℓ-PCG method, Tikhonov regularization, parallel image denoising, Kronecker product preconditioner, linear operator equation.
This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1A09917364).

1. Introduction

Digital images generated by various digital devices or used in practical applications usually contain some noises, so denoising is necessary before analyzing the images. Over the last decade, a number of image denoising techniques preserving important image information such as edges have been proposed to obtain the true image from the noisy (known) image [1-3, 8, 9].

Let $B \in \mathbb{R}^{m\times n}$ be an observed noisy image and $Z \in \mathbb{R}^{m\times n}$ be an unknown noise array. Then, we would like to construct the true image $X \in \mathbb{R}^{m\times n}$ from the observed noisy image B such that

$$B = X + Z.$$

Image denoising problem is to reconstruct X from B which contains the additive noise Z. It is well-known that image denoising problem is an ill-conditioned problem, so that regularization techniques should be used to approximate X from B. The problem of image denoising problem usually reduces to solving the following *Tikhonov regularization problem*:

$$\min_{X\in\mathbb{R}^{m\times n}} \{\| X - B \|_F^2 + \lambda^2 \mathcal{R}(X)\}, \tag{1.1}$$

where $\lambda > 0$ is a regularization parameter, $\|\cdot\|_F$ denotes the Frobenius norm, the first term $\| X - B \|_F^2$ is called the *data fitting term*, and the second term $\mathcal{R}(X)$ is called a *regularization term*. If $\mathcal{R}(X)$ is chosen to be a discrete total variation (TV) semi-norm of X, then the problem (1.1) is called the *discrete TV-based image denoising problem* [1-3]. Notice that original version of the TV-based image denoising model has been introduced by Rudin, Osher and Fatemi (ROF) in [9] as a regularization approach for preserving edges while removing noises. Another common choice of $\mathcal{R}(X)$ is $\| Dx \|_2^2$, where $x = vec(X) \in \mathbb{R}^{mn}$ denotes a column vector obtained by stacking the columns of X into a long vector x, and $D \in \mathbb{R}^{mn\times mn}$ is a discrete approximation of the first or second order partial derivative

operators [4]. In this paper, we are interested in solving the following discrete image denoising problem:

$$\min_{X \in \mathbb{R}^N} \{\| X - B \|_F^2 + \lambda^2 \| Dx \|_2^2\}. \tag{1.2}$$

In 2006, the global conjugate gradient (Gℓ-CG) method for solving a linear system with multiple right hand sides of the form $AX = B$ has been proposed by Salkuyeh [10], where A is a symmetric positive definite matrix. Big advantage of the Gℓ-CG method is that they have a rich parallelism which is very suitable for advanced parallel supercomputers. Recently, Yun [11] proposed an application of the global preconditioned conjugate gradient (Gℓ-PCG) method to the discrete image denoising problem (1.2).

This paper is organized as follows. In Section 2, we introduce some definitions and properties which are used in this paper. In Section 3, we first show how to construct the linear operator equations for the discrete image denoising problem (1.2) corresponding to two cases of regularization matrices D, and then we propose how to choose *Kronecker product preconditioners* which are required for accelerating the Gℓ-PCG method. In Section 4, we provide how to apply the Gℓ-PCG method with Kronecker product preconditioner to the linear operator equations. In Section 5, we propose a coarse-grained parallel image denoising algorithm using the Gℓ-PCG with Kronecker product preconditioner that is suitable for personal computers with multiple cores which need a lot of communication time among the cores and overhead (or startup) time. In Section 6, we provide numerical experiments for several image denoising problems to evaluate the effectiveness of the Gℓ-PCG with Kronecker product preconditioner. Lastly, some conclusions are drawn.

2. Preliminaries

We need to represent images as arrays of numbers in order to use mathematical techniques for denoising. Let $X \in \mathbb{R}^{m\times n}$ represent the desired true image, and let $B \in \mathbb{R}^{m\times n}$ denote the noisy image or the observed image. Using the *vec* operator, x and b are defined by

$$x = vec(X) = (x_1^T, x_2^T, ..., x_n^T)^T \in \mathbb{R}^N,$$

$$b = vec(B) = (b_1^T, b_2^T, ..., b_n^T)^T \in \mathbb{R}^N,$$

where $N = mn$, $x_i \in \mathbb{R}^m$ and $b_i \in \mathbb{R}^m$ are the ith columns of X and B, respectively. Then it is easy to show that the discrete image denoising problem (1.2) is mathematically equivalent to solving the following linear system:

$$(I_N + \lambda^2 D^T D)x = b, \tag{2.1}$$

where I_N denotes an identity matrix of order N.

If the size of the original image X is $m \times n$ and, m and n are large, then the regularization matrix D is a very large and sparse matrix of order mn. Since the coefficient matrix $I_N + \lambda^2 D^T D$ is symmetric positive definite, the linear system (2.1) can be solved using the PCG (preconditioned conjugate gradient) method [5]. If D can be represented by the Kronecker product of D_r and D_c, i.e., $D = D_r \otimes D_c$, where $D_r \in \mathbb{R}^{n\times n}$ and $D_c \in \mathbb{R}^{m\times m}$, then the large sparse linear system (2.1) can be transformed into the small size of matrix equations which are generated from D_r and D_c. For this reason, we want to study how to solve the small size of matrix equations using the Gℓ-PCG method instead of solving the large sparse linear system (2.1) using the PCG method. Constructing such matrix equations is discussed in the next section.

For matrices X and $Y \in \mathbb{R}^{m\times n}$, the *Frobenius inner product* of X and Y is defined by $\langle X, Y\rangle_F = tr(X^T Y)$, and the corresponding *Frobenius norm* of $X \in \mathbb{R}^{m\times n}$ is defined by $\| X \|_F = \sqrt{\langle X, X\rangle_F}$, where $tr(X^T Y)$ denotes the *trace* of $X^T Y$ which is the sum of its main diagonal entries. It is well-known that if $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{n\times n}$, then $tr(AB) = tr(BA)$ and $tr(C) = \sum_{i=1}^{n} \lambda_i$, where $\lambda_1, \lambda_2, ..., \lambda_n$ are the eigenvalues of C [7].

A bounded linear operator $\mathcal{T} : H \to H$ is called *self-adjoint* if $\mathcal{T}^* = \mathcal{T}$, where H is a Hilbert space and $\mathcal{T}^*$ is the *adjoint operator* of $\mathcal{T}$. That is, $\mathcal{T}$ is self-adjoint if and only if $\langle \mathcal{T}x, y\rangle = \langle x, \mathcal{T}y\rangle$ for all $x, y \in H$, where $\langle \cdot, \cdot \rangle$ is an inner product on H. A self-adjoint operator $\mathcal{T} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is called *positive definite* if $\langle X, \mathcal{T}(X)\rangle_F > 0$ for all $X \neq O$ in $\mathbb{R}^{m\times n}$ [6].

In this paper, we have used two boundary conditions which are zero and reflexive boundary conditions. The zero boundary condition is to assume that the exact image is black outside the boundary. The reflexive boundary condition implies that the scene outside the image boundaries is a mirror image of the scene inside the image boundaries.

3. Construction of Linear Operator Equation

In this section, we study the construction of linear operator equations for the discrete image denoising problem (1.2) corresponding to two cases of matrices D.

3.1. Operator equation for D corresponding to $\| x_s \|_2^2 + \| x_t \|_2^2$ (Case 1)

We consider the discrete image denoising problem (1.2) for the case where D is an approximate matrix corresponding to $\| x_s \|_2^2 + \| x_t \|_2^2$, where

x_s and x_t are the first order partial derivative operators of the image $X \in \mathrm{R}^{m\times n}$, and s and t denote the variables in the vertical direction and the horizontal direction, respectively. Consider the regularization matrix D such that $\| Dx \|_2^2 = \| x_s \|_2^2 + \| x_t \|_2^2$. Let $D_{1,m} \in \mathbb{R}^{m\times m}$ be an approximate matrix obtained by a finite difference approximation to the first order derivative operator [4]. More specifically, the matrix $D_{1,m} \in \mathbb{R}^{m\times m}$ for zero boundary condition is given by

$$D_{1,m} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ 0 & \cdots & 0 & 0 & -1 \end{bmatrix},$$

and $D_{1,m} \in \mathbb{R}^{m\times m}$ for reflexive boundary condition is given by

$$D_{1,m} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

Then we can easily obtain

$$\| Dx \|_2^2 = \| (I_n \otimes D_{1,m})x \|_2^2 + \| (D_{1,n} \otimes I_m)x \|_2^2$$

$$= \left\| \begin{pmatrix} (I_n \otimes D_{1,m})x \\ (D_{1,n} \otimes I_m)x \end{pmatrix} \right\|_2^2,$$

where $D_{1,m} \in \mathbb{R}^{m\times m}$ and $D_{1,n} \in \mathbb{R}^{n\times n}$. Thus, (1.2) can be transformed into the following form:

$$\min_{x\in\mathbb{R}^N}\{\| x-b\|_2^2+\lambda^2\| I_n\otimes D_{1,m}x\|_2^2+\lambda^2\| D_{1,n}\otimes I_m x\|_2^2\}$$

$$=\min_{x\in\mathbb{R}^N}\left\{\left\|\begin{pmatrix} I_N\\ \lambda D_s\\ \lambda D_t\end{pmatrix}x-\begin{pmatrix} b\\ 0\\ 0\end{pmatrix}\right\|_2^2\right\}, \tag{3.1}$$

where $D_s = I_n \otimes D_{1,m}$ and $D_t = D_{1,n} \otimes I_m$. It is easy to show that the minimization problem (3.1) is equivalent to solving the following equation:

$$(I_N + \lambda^2 D_s^T D_s + \lambda^2 D_t^T D_t)x = b. \tag{3.2}$$

Since $D_s = I_n \otimes D_{1,m}$ and $D_t = D_{1,n} \otimes I_m$, the linear system (3.2) can be rewritten as

$$\{I_n \otimes I_m + \lambda^2(I_n \otimes D_{1,m}^T D_{1,m}) + \lambda^2(D_{1,n}^T D_{1,n} \otimes I_m)\}x = b. \tag{3.3}$$

From (3.3), the following matrix equation can be obtained:

$$X + \lambda^2(D_{1,m}^T D_{1,m}X + XD_{1,n}^T D_{1,n}) = B, \tag{3.4}$$

where $B \in \mathbb{R}^{m\times n}$ is a matrix such that $b = vec(B)$. Let us define the linear operator $\mathcal{A}_1 : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ given by

$$\mathcal{A}_1(X) = X + \lambda^2(D_{1,m}^T D_{1,m}X + XD_{1,n}^T D_{1,n}). \tag{3.5}$$

Then (3.4) can be expressed as the following linear operator equation:

$$\mathcal{A}_1(X) = B. \tag{3.6}$$

The following theorem shows that $\mathcal{A}_2$ is self-adjoint and positive definite.

Theorem 3.1. *The operator $\mathcal{A}_1$ is self-adjoint and positive definite.*

Proof. For all X and Y in $\mathbb{R}^{m\times n}$,

$$\begin{aligned}\langle \mathcal{A}_1(X),\, Y\rangle_F &= \langle X + \lambda^2(D_{l,m}^T D_{l,m} X + X D_{l,n}^T D_{l,n}),\, Y\rangle_F \\ &= tr(X^T Y + \lambda^2(X^T D_{l,m}^T D_{l,m} Y + D_{l,n}^T D_{l,n} X^T Y)) \\ &= tr(X^T Y) + \lambda^2(tr(X^T D_{l,m}^T D_{l,m} Y) + tr(X^T Y D_{l,n}^T D_{l,n})) \\ &= tr(X^T Y + \lambda^2 X^T (D_{l,m}^T D_{l,m} Y + Y D_{l,n}^T D_{l,n})) \\ &= \langle X,\, Y + \lambda^2(D_{l,m}^T D_{l,m} Y + Y D_{l,n}^T D_{l,n})\rangle_F \\ &= \langle X,\, \mathcal{A}_1(Y)\rangle_F .\end{aligned}$$

Hence, the operator $\mathcal{A}_1$ is self-adjoint. For each $X \in \mathbb{R}^{m\times n}$ and $X \neq O$,

$$\begin{aligned}&\langle X,\, \mathcal{A}_1(X)\rangle_F \\ &= tr(X^T X + \lambda^2(X^T D_{l,m}^T D_{l,m} X + D_{l,n}^T D_{l,n} X^T X)) \\ &= tr(X^T X) + \lambda^2(tr(X^T D_{l,m}^T D_{l,m} X) + tr(X D_{l,n}^T D_{l,n} X^T)) \\ &= tr(X^T X) + \lambda^2\{tr((D_{l,m} X)^T (D_{l,m} X)) + tr((D_{l,n} X^T)^T (D_{l,n} X^T))\} \\ &= \| X \|_F^2 + \lambda^2(\| D_{l,m} X \|_F^2 + \| D_{l,n} X^T \|_F^2) > 0.\end{aligned}$$

Hence, the operator $\mathcal{A}_1$ is positive definite. □

From the left side of the linear system (3.3), one can obtain the following approximate relation:

$$\begin{aligned}&(I_n \otimes I_m + \lambda^2(I_n \otimes D_{l,m}^T D_{l,m}) + \lambda^2(D_{l,n}^T D_{l,n} \otimes I_m))x \\ &\approx (I_n + \lambda^2 D_{l,n}^T D_{l,n}) \otimes (I_m + \lambda^2 D_{l,m}^T D_{l,m})x. \qquad (3.7)\end{aligned}$$

From (3.7), we can choose a preconditioner of the form

$$M_1 = M_r \otimes M_c,$$

where

$$M_r = I_n + \lambda^2 D_{l,n}^T D_{l,n},$$

$$M_c = I_m + \lambda^2 D_{l,m}^T D_{l,m}.$$

Clearly, $M_r \in \mathbb{R}^{n\times n}$ and $M_c \in \mathbb{R}^{m\times m}$ are tridiagonal matrices. Now we define a preconditioner operator $\mathcal{M}_1 : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ by

$$\mathcal{M}_1(X) = M_c X M_r^T. \tag{3.8}$$

Theorem 3.2. *The preconditioner operator* $\mathcal{M}_1$ *in* (3.8) *is self-adjoint and positive definite*.

Proof. It is clear that M_r and M_c are symmetric positive definite. For all $X, Y \in \mathbb{R}^{m\times n}$,

$$\langle \mathcal{M}_1(X), Y\rangle_F = \langle M_c X M_r^T, Y\rangle_F = tr(M_r X^T M_c^T Y)$$

$$= tr(X^T M_c^T Y M_r) = \langle X, \mathcal{M}_2(Y)\rangle_F.$$

Hence, the preconditioner operator $\mathcal{M}_1$ is self-adjoint. Next, we show that $\mathcal{M}_2$ is positive definite. For each $X \in \mathbb{R}^{m\times n}$ and $X \neq O$, one obtains

$$\langle \mathcal{M}_1(X), X\rangle_F = tr(M_r X^T M_c^T X) = tr(X^T M_c^T X M_r) = tr(X^T M_c X M_r).$$

Since $X^T M_c X M_r$ is similar to $M_r^{\frac{1}{2}} X^T M_c X M_r^{\frac{1}{2}}$ which is symmetric positive semidefinite, all eigenvalues of $X^T M_c X M_r$ are nonnegative and so $tr(X^T M_c X M_r) \geq 0$. Since M_r and M_c are symmetric positive definite,

$XM_r^{\frac{1}{2}} \neq 0$ and so $M_r^{\frac{1}{2}} X^T M_c X M_r^{\frac{1}{2}} = (XM_r^{\frac{1}{2}})^T M_c (XM_r^{\frac{1}{2}})$ is a non-zero symmetric matrix. Thus, all eigenvalues of $X^T M_c X M_r$ are not zero, which implies $tr(X^T M_c X M_r) > 0$. Hence, $\mathcal{M}_1$ is positive definite. □

3.2. Operator equation for D corresponding to $\| x_{ss} \|_2^2 + \| x_{tt} \|_2^2$ (Case 2)

We consider the discrete image denoising problem (1.2) for the case where D is an approximate matrix corresponding to $\| x_{ss} \|_2^2 + \| x_{tt} \|_2^2$. Consider the regularization matrix D such that $\| Dx \|_2^2 = \| x_{ss} \|_2^2 + \| x_{tt} \|_2^2$, where x_{ss} and x_{tt} are the second order partial derivative operators of the image $X \in \mathbb{R}^{m \times n}$. Let $D_{2,m} \in \mathbb{R}^{m \times m}$ be an approximate matrix obtained by a finite difference approximation to the second order derivative operator [4]. More specifically, the matrix $D_{2,m} \in \mathbb{R}^{m \times m}$ for zero boundary condition is given by

$$D_{2,m} = \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -1 & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix},$$

and $D_{2,m} \in \mathbb{R}^{m \times m}$ for reflexive boundary condition is given by

$$D_{2,m} = \begin{bmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -1 & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{bmatrix}.$$

Then we can easily obtain

$$\| Dx \|_2^2 = \| (I_n \otimes D_{2,m})x \|_2^2 + \| (D_{2,n} \otimes I_m)x \|_2^2$$

$$= \left\| \begin{pmatrix} (I_n \otimes D_{2,m})x \\ (D_{2,n} \otimes I_m)x \end{pmatrix} \right\|_2^2,$$

where $D_{2,m} \in \mathrm{R}^{m \times m}$ and $D_{2,n} \in \mathrm{R}^{n \times n}$. Thus, (1.2) can be transformed into the following form:

$$\min_{x \in \mathbb{R}^N} \{ \| x - b \|_2^2 + \lambda^2 \| I_n \otimes D_{2,m} x \|_2^2 + \lambda^2 \| D_{2,n} \otimes I_m x \|_2^2 \}$$

$$= \min_{x \in \mathbb{R}^N} \left\{ \left\| \begin{pmatrix} I_N \\ \lambda D_s \\ \lambda D_t \end{pmatrix} x - \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix} \right\|_2^2 \right\}, \tag{3.9}$$

where $D_s = I_n \otimes D_{2,m}$ and $D_t = D_{2,n} \otimes I_m$. It is easy to show that the minimization problem (3.9) is equivalent to solving the following equation:

$$(I_N + \lambda^2 D_s^T D_s + \lambda^2 D_t^T D_t)x = b. \tag{3.10}$$

Since $D_s = I_n \otimes D_{2,m}$ and $D_t = D_{2,n} \otimes I_m$, the linear system (3.10) can be rewritten as

$$\{ I_n \otimes I_m + \lambda^2 (I_n \otimes D_{2,m}^T D_{2,m}) + \lambda^2 (D_{2,n}^T D_{2,n} \otimes I_m) \} x = b. \tag{3.11}$$

From (3.11), the following matrix equation can be obtained:

$$X + \lambda^2 (D_{2,m}^T D_{2,m} X + X D_{2,n}^T D_{2,n}) = B, \tag{3.12}$$

where $B \in \mathbb{R}^{m \times n}$ is a matrix such that $b = vec(B)$. Let us define the linear operator $\mathcal{A}_2 : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ given by

$$\mathcal{A}_2(X) = X + \lambda^2 (D_{2,m}^T D_{2,m} X + X D_{2,n}^T D_{2,n}). \tag{3.13}$$

Then (3.12) can be expressed as the following linear operator equation

$$\mathcal{A}_2(X) = B. \tag{3.14}$$

In the similar way as was done in the proof of Theorem 3.1, the following theorem can be easily obtained.

Theorem 3.3. *The operator $\mathcal{A}_2$ is self-adjoint and positive definite*.

From the left side of the linear system (3.11), one can obtain the following approximate relation:

$$\begin{aligned}&(I_n \otimes I_m + \lambda^2(I_n \otimes D_{2,m}^T D_{2,m}) + \lambda^2(D_{2,n}^T D_{2,n} \otimes I_m))x \\ &\approx (I_n + \lambda^2 D_{2,n}^T D_{2,n}) \otimes (I_m + \lambda^2 D_{2,m}^T D_{2,m})x. \end{aligned} \tag{3.15}$$

From (3.15), we can choose a preconditioner of the form $M_2 = M_r \otimes M_c$, where

$$M_r = I_n + \lambda^2 D_{2,n}^T D_{2,n} \quad \text{and} \quad M_c = I_m + \lambda^2 D_{2,m}^T D_{2,m}.$$

Now we define a preconditioner operator $\mathcal{M}_2 : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ by $\mathcal{M}_2(X) = M_c X M_r^T$. Then the following theorem can be easily obtained as was done in the proof of Theorem 3.2.

Theorem 3.4. *The preconditioner operator $\mathcal{M}_2$ is self-adjoint and positive definite*.

Remark 3.1. We have shown that the linear operators $\mathcal{A}_i$ $(i = 1, 2)$ and the preconditioner operators $\mathcal{M}_i$ $(i = 1, 2)$ are self-adjoint and positive definite since these conditions are required for applying the Gℓ-PCG method to the linear operator equations.

4. Gℓ-PCG Method for Linear Operator Equations

In this section, we propose the global preconditioned conjugate gradient (Gℓ-PCG) algorithm for solving the linear operator equations $\mathcal{A}_i(X) = B$ $(i = 1, 2)$, where $\mathcal{A}_i$'s are the linear operators defined in Section 3.

By combining the ideas of the PCG method [5] and the Gℓ-CG method [10], we can obtain the following Gℓ-PCG algorithm applied to the linear operator equations $\mathcal{A}_i(X) = B$ $(i = 1, 2)$.

Algorithm 1. Gℓ-PCG for solving linear operator equation $\mathcal{A}_i(X) = B$

---

1. Compute $R_0 \coloneqq B - \mathcal{A}_i(X_0)$, $Z_0 = \mathcal{M}_i^{-1}(R_0)$, $P_0 = Z_0$.

2. For $j = 0, 1, ...,$ until convergence Do :

$$\alpha_j \coloneqq \langle R_j, Z_j \rangle_F / \langle \mathcal{A}_i(P_j), P_j \rangle_F$$

$$X_{j+1} \coloneqq X_j + \alpha_j P_j$$

$$R_{j+1} \coloneqq R_j - \alpha_j \mathcal{A}_i(P_j)$$

$$Z_{j+1} \coloneqq \mathcal{M}_i^{-1}(R_{j+1})$$

$$\beta_j \coloneqq \langle R_{j+1}, Z_{j+1} \rangle_F / \langle R_j, Z_j \rangle_F$$

$$P_{j+1} \coloneqq Z_{j+1} + \beta_j P_j.$$

3. End

Notice that $\mathcal{A}_i$ and $\mathcal{M}_i$ are self-adjoint and positive definite operators defined in Section 3. If the operators $\mathcal{A}_i$ and $\mathcal{M}_i$ in Algorithm 1 are replaced by symmetric positive definite matrices A and M respectively, then Algorithm 1 reduces to the Gℓ-PCG algorithm for solving a linear system with multiple right hand sides of the form $AX = B$.

5. Parallel Image Denoising Using the Gℓ-PCG Method

In this section, we propose a parallel image denoising algorithm using the Gℓ-PCG method with Kronecker product preconditioners when the pixel size of the noisy image B is large. Let ℓ denote the number of processors (or cores) to be used. For simplicity of exposition, suppose that n is divisible by ℓ. Then the noisy image $B \in \mathbb{R}^{m\times n}$ and the true image $X \in \mathbb{R}^{m\times n}$ are partitioned into ℓ equal column blocks of the form

$$B = (B_1 \quad B_2 \quad \cdots \quad B_\ell), \quad X = (X_1 \quad X_2 \quad \cdots \quad X_\ell),$$

where B_i and X_i are arrays of the equal size $m \times \frac{n}{\ell}$ which is required for load-balancing of parallel computing. Then each processor k needs to execute the following operations:

Construct the regularization matrix $D^{(k)}$ corresponding to B_k

Construct Kronecker preconditioner M_{r_k} and M_{c_k} from $D^{(k)}$

Compute X_k by applying the Gℓ-PCG to the linear operator equation generated from $D^{(k)}$.

Finally, the true image X can be approximated by collecting X_k from each processor k. The parallel algorithm corresponding to the above operations can be written using the Matlab *parfor* statement as follows:

parfor $k = 1$ to ℓ

Construct $D^{(k)}$ corresponding to B_k

Construct M_{r_k} and M_{c_k} from $D^{(k)}$

Apply the Gℓ-PCG to the linear operator equation generated from $D^{(k)}$ to compute X_k

end

Since the true image X is formed by collecting X_k from each processor k, the reflexive boundary condition should be used to improve the continuity of the image X at the boundary of X_k.

Since the Gℓ-PCG has a rich parallelism, we can easily parallelize the Gℓ-PCG algorithm itself, which is called a *fine-grained parallelization* that is suitable for advanced parallel supercomputers. In this section, we provided a coarse-grained parallel algorithm that is suitable for personal computers with multiple cores which need a lot of communication time among the cores and overhead (or startup) time. That is, we provided a coarse-grained parallel denoising algorithm based on matrix blocks of B_k instead of parallelizing the Gℓ-PCG working on the total image of B.

6. Numerical Experiments

In this section, we provide numerical experiments for several image denoising problems to estimate the efficiency of the Gℓ-PCG method with Kronecker product preconditioners which is applied to the linear operator equations $\mathcal{A}_i(X) = B$ $(i = 1, 2)$ discussed in Section 3.

The noisy images B are generated by adding Gaussian white noise to the clean images using the built-in MATLAB function *randn*. For example, adding 10% of Gaussian white noise to the true image X can be done as follows;

$$E = \text{randn}(m, n); \quad \widetilde{E} = \frac{E}{\| E \|_F}; \quad B = X + 0.10 \cdot \| X \|_F \cdot \widetilde{E},$$

where $\| \cdot \|_F$ refers to the Frobenius norm.

All numerical tests have been performed using Matlab R2016a on a personal computer, which has 4 Cores, equipped with Intel Core i5-4570

3.2GHz CPU and 8GB RAM. The initial image X_0 is set to the noisy image B, and the stopping criterion for the Gℓ-PCG at the kth iterate is

$$\frac{\| R_k \|_F}{\| R_0 \|_F} \leq 10^{-3},$$

where R_k represents the kth residual matrix corresponding to the kth iteration matrix X_k of the Gℓ-PCG with R_0 the initial residual matrix corresponding to X_0.

The restored images is measured by the PSNR (peak signal to noise ratio) and ISNR (improvement in signal to noise ratio) which are defined by

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

where $f = (f_{ij})$ is the true image, $\hat{f} = (\hat{f}_{ij})$ is a noisy image of f and g is a restored image for $\hat{f}$. PSNR_0 in Tables 1 and 2 denotes the PSNR for the noisy image $\hat{f}$ which is defined by

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

Table 1. Numerical results for image denoising problem (Case 1)

BC	Image	Noise	$PSNR_0$	Gℓ-PCG				
				ISNR	PSNR	λ	Itime	IT
Z	Pepper	10%	25.55	3.07	28.62	0.50	0.024	3
		15%	22.03	4.50	26.52	0.65	0.030	4
	Cameraman	10%	25.51	2.57	28.08	0.45	0.024	3
		15%	21.99	3.96	25.96	0.60	0.030	4
	Joomaks	10%	25.03	3.68	28.71	0.55	2.68	3
		15%	21.51	5.21	26.72	0.75	3.40	4
R	Pepper	10%	25.55	3.46	29.01	0.55	0.024	3
		15%	22.03	5.01	27.04	0.75	0.030	4
	Cameraman	10%	25.51	2.68	28.20	0.47	0.024	3
		15%	21.99	4.15	26.14	0.65	0.030	4
	Joomaks	10%	25.03	3.71	28.74	0.55	2.68	3
		15%	21.51	5.25	26.75	0.75	3.40	4

Table 2. Numerical results for image denoising problem (Case 2)

BC	Image	Noise	$PSNR_0$	Gℓ-PCG				
				ISNR	PSNR	λ	Itime	IT
Z	Pepper	10%	25.55	3.31	28.86	0.35	0.028	4
		15%	22.03	4.79	26.82	0.50	0.037	6
	Cameraman	10%	25.51	2.61	28.12	0.30	0.028	4
		15%	21.99	3.94	25.93	0.40	0.033	5
	Joomaks	10%	25.03	4.23	29.26	0.40	4.23	5
		15%	21.51	5.63	27.15	0.60	5.75	7
R	Pepper	10%	25.55	3.68	29.23	0.40	0.033	5
		15%	22.03	5.21	27.23	0.55	0.037	6
	Cameraman	10%	25.51	2.81	28.32	0.30	0.028	4
		15%	21.99	4.25	26.24	0.45	0.033	5
	Joomaks	10%	25.03	4.32	29.35	0.45	4.25	5
		15%	21.51	5.74	27.25	0.60	5.77	7

Table 3. Parallel performance results of Gℓ-PCG for Joomaks image

D	ℓ	Noise = 10%					Noise = 15%				
		PSNR	λ	Itime	S_ℓ	IT	PSNR	λ	Itime	S_ℓ	IT
Case 1	1	28.74	0.55	2.68	1	3	26.75	0.75	3.40	1	4
	2	28.74		2.27	1.18	3	26.75		2.83	1.20	4
	3	28.73		1.71	1.57	3	26.75		1.98	1.72	4
	4	28.73		1.67	1.60	3	26.75		1.86	1.83	4
Case 2	1	29.35	0.45	4.25	1	5	27.25	0.60	5.77	1	7
	2	29.35		3.39	1.25	5	27.25		4.38	1.32	7
	3	29.34		2.42	1.76	5	27.25		3.01	1.92	7
	4	29.34		2.09	2.03	5	27.35		2.70	2.14	7

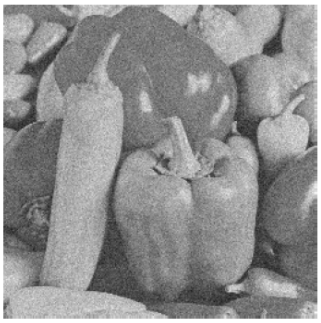
We have used 3 test images such as Pepper, Cameraman and Joomaks for numerical experiments. The pixel size of Pepper and Cameraman images is 256×256, and the pixel size of Joomaks image is 2200×2200. For all test problems, we have used two boundary conditions - one is the zero boundary condition and the other is the reflexive boundary condition. Numerical experiments have been carried out for noisy images that are generated by adding 10% or 15% of Gaussian white noise to the true image. Numerical experiments for the parallel image denoising algorithm using the Gℓ-PCG have been carried out only for Joomaks image of large size (see Table 3). For small size of images, we do not have performance gains from parallel execution since personal computer needs a lot of overhead time and communication time among the processors (or cores). Thus, we do not provide parallel performance results for small size of images.

In all tables, the column labeled with "Noise" represents the percentage of Gaussian white noise, "Itime" represents the elapsed CPU time in seconds required for iteration steps of Gℓ-PCG which includes the construction time for M_r and M_c, "IT" represents the number of iterations for the Gℓ-PCG method, "BC" represents boundary condition being used, "*Z*" and "*R*" stand

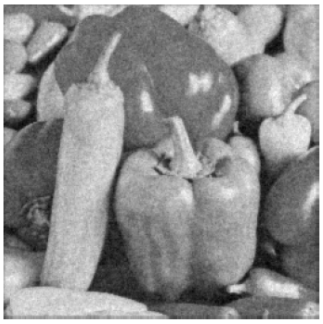
for zero and reflexive boundary conditions, respectively, and "λ" represents a near optimal regularization parameter for the Gℓ-PCG method which is chosen by numerical tries. That is, the Gℓ-PCG uses an experimentally chosen regularization parameter λ. In Table 3, "ℓ" represents the number of cores to be used, and S_ℓ stands for the speedup of parallel execution on ℓ processors (or cores).

Numerical results show that Case 2 of D denoises the image better than Case 1 of D and the reflexive boundary condition performs better than the zero boundary condition for the image denoising problem (see Tables 1-2 and Figures 1-3). As can be seen in Table 3, the coarse-grained parallel denoising algorithm proposed in Section 5 performs quite efficiently on a personal computer with large parallel overhead time. Speedups of the personal computer with 4 cores range from 1.6 to 2.1 depending upon the amount of total computational time. If the coarse-grained parallel algorithm is performed on advanced parallel supercomputers with 4 processors, then its parallel speedup may be very close to 4. Notice that IT (i.e., the number of iterations) does not vary as ℓ (i.e., the number of cores) increases. Also, notice that PSNR values are almost the same from parallel execution on the personal computer with 4 cores. This means that parallel execution does not deteriorate the quality of image denoising.

(a) True image

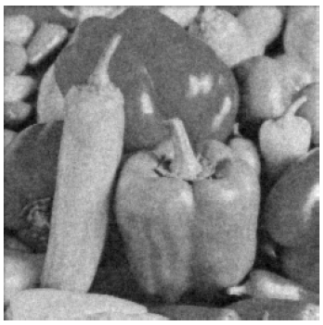


(b) Noisy image



(c) Denoised image for Case 1

(d) Denoised image for Case 2



(e) Denoised image for Case 1

(f) Denoised image for Case 2

Figure 1. Performance of Gℓ-PCG for Pepper image with 15% of noise ((c) and (d) : zero BC, (e) and (f) : reflexive BC).

(a) True image

(b) Noisy image

(c) Denoised image for Case 1

(d) Denoised image for Case 2

(e) Denoised image for Case 1

(f) Denoised image for Case 2

Figure 2. Performance of Gℓ-PCG for Cameraman image with 15% of noise ((c) and (d) : zero BC, (e) and (f) : reflexive BC).

(a) True image

(b) Noisy image

(c) Denoised image for Case 1

(d) Denoised image for Case 2

(e) Denoised image for Case 1

(f) Denoised image for Case 2

Figure 3. Performance of Gℓ-PCG for Joomaks image with 15% of noise ((c) and (d) : zero BC, (e) and (f) : reflexive BC).

7. Conclusions

Numerical results show that Case 2 of D denoises the image better than Case 1 of D and the reflexive boundary condition performs better than the zero boundary condition for the image denoising problem (see Tables 1-2 and Figures 1-3). Hence, Gℓ-PCG with Kronecker product preconditioner corresponding to Case 2 of D using the reflexive boundary condition is recommended for the image denoising problem.

As can be seen in Table 3, the coarse-grained parallel denoising algorithm proposed in Section 5 performs quite efficiently on a personal computer with large parallel overhead time. Speedups of the personal computer with 4 cores range from 1.6 to 2.1 depending upon the amount of total computational time. If the coarse-grained parallel algorithm is performed on advanced parallel supercomputers with 4 processors, then its parallel speedup may be very close to 4. Notice that IT (i.e., the number of iterations) does not vary as ℓ (i.e., the number of cores) increases and PSNR values are almost the same from parallel execution on the personal computer with 4 cores. This means that parallel execution does not deteriorate the quality of image denoising. Since Gℓ-PCG contains a rich parallelism, future work will study development of a fine-grained parallel algorithm for Gℓ-PCG which is suitable for the advanced parallel supercomputers.

References

[1] R. Acar and C. Vogel, Analysis of bounded variation penalty methods for ill-posed problems, IEEE Trans. Image Process. 10 (1994), 1217-1229.

[2] A. Beck and M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, IEEE Trans. Image Process. 18 (2009), 2419-2434.

[3] A. Chambolle, An algorithm for total variation minimization and applications, J. Math. Imaging Vision 20 (2004), 89-97.

[4] P. C. Hansen, J. G. Nagy and D. P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, Philadelphia, 2006.

[5] M. R. Hestense and E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Standards 49 (1952), 409-436.

[6] E. Kreyszig, Introductory Functional Analysis with Applications, John Wiley & Sons, Inc., New York, 1978.

[7] A. J. Laub, Matrix Analysis for Scientists and Engineers, SIAM, Philadelphia, 2005.

[8] A. Matakos, S. Ramani and J. A. Fessler, Accelerated edge-preserving image restoration without boundary artifacts, IEEE Trans. Image Process. 22(5) (2013), 2019-2029.

[9] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, Phys. D 60 (1992), 259-268.

[10] D. K. Salkuyeh, CG-type algorithms to solve symmetrics matrix equations, Appl. Math. Comput. 172 (2006), 985-999.

[11] J. H. Yun, Performance of Gℓ-PCG method for image denoising problems, J. Appl. Math. Informatics 35 (2017), 399-411.