J. Appl. Math. & Informatics Vol. **36***(2018), No. 3 - 4, pp. 317 - 330*
https://doi.org/10.14317/jami.2018.317

PARALLEL PERFORMANCE OF THE Gℓ-PCG METHOD FOR IMAGE DEBLURRING PROBLEMS[†]

JAE HEON YUN

Abstract. We first provide how to apply the global preconditioned conjugate gradient (Gℓ-PCG) method with Kronecker product preconditioners to image deblurring problems with nearly separable point spread functions. We next provide a coarse-grained parallel image deblurring algorithm using the Gℓ-PCG. Lastly, we provide numerical experiments for image deblurring problems to evaluate the effectiveness of the Gℓ-PCG with Kronecker product preconditioner by comparing its performance with those of the Gℓ-CG, CGLS and preconditioned CGLS (PCGLS) methods.

AMS Mathematics Subject Classification : 94A08, 65F10, 68W10.
Key word and phrases : Image deblurring, Gℓ-PCG method, Kronecker product preconditioner, Point spread function.

1. Introduction

Image deblurring is the process of restoring or estimating the true image from the observed blurred and noisy image. The blurred image is arising from the physical process of image devices that can be expressed by a linear mathematical model. The problem of image deblurring usually reduces to the following *Tikhonov regularization problem*

$$\min_{x\in\mathbb{R}^N} \left\{ \|Ax-b\|_2^2 + \lambda^2 \|Dx\|_2^2 \right\}, \tag{1}$$

where $\lambda > 0$ is a regularization parameter, $A \in \mathbb{R}^{N\times N}$ is a blurring matrix which is very ill-conditioned, $D \in \mathbb{R}^{N\times N}$ is an identity matrix or a discrete approximation of the first or second order partial derivative operators (see [2, 4, 5]), the first term $\|Ax-b\|_2^2$ is called the data-fitting term, the second term $\|Dx\|_2^2$ is a regularization (or penalty) term, $x \in \mathbb{R}^N$ and $b \in \mathbb{R}^N$ represent the original and observed images respectively. In this paper, we are interested in solving the

Received January 25, 2018. Revised March 20, 2018. Accepted April 5, 2018.
[†]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).

Tikhonov regularization problem in which D is a discrete approximation of the first or second order partial derivative operators,

In 2006, Salkuyeh [13] proposed the global conjugate gradient (Gℓ-CG) method based on matrix Krylov subspace for solving a linear system with multiple right hand sides of the form $AX = B$, where A is a symmetric positive definite matrix. The Gℓ-CG method has a rich parallelism, so that it is very suitable for advanced parallel supercomputers. For this reason, the purpose of this paper is to study applications of the *global preconditioned conjugate gradient (Gℓ-PCG) method* to the Tikhonov regularization problem (1).

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 introduce two linear operator equations and *Kronecker product preconditioners* corresponding to two choices of regularization matrices D in the Tikhonov regularization problem (1) when A and D can be represented or well approximated by Kronecker products [8, 9]. In Section 4, we provide how to apply the Gℓ-PCG method with Kronecker product preconditioners to the linear operator equations. In Section 5, we propose a coarse-grained parallel image deblurring algorithm using the Gℓ-PCG 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, the effectiveness of the Gℓ-PCG with Kronecker product preconditioners is evaluated by comparing numerical results of the Gℓ-PCG method with those of the Gℓ-CG, CGLS and preconditioned CGLS (PCGLS) methods [1, 10] for image deblurring problems. Lastly, some conclusions are drawn.

2. Preliminaries

We first introduce the *vec* operator which transforms a matrix $C \in \mathbb{R}^{m\times n}$ into a column vector $c \in \mathbb{R}^N$ by stacking the columns of C, i.e.

$$c = vec(C) = (c_1^T, c_2^T, \ldots, c_n^T)^T$$

where $N = mn$ and c_i denotes the ith column of C. Let $X \in \mathbb{R}^{m\times n}$ represent the original or true image, and let $B \in \mathbb{R}^{m\times n}$ denote the the observed blurred and noisy image. Then there exists a large sparse matrix A such that $Ax = b$, where $x = vec(X)$, $b = vec(B)$, and the matrix A represents the blurring operator which transforms the original image into the blurred image. Notice that the blurring matrix A is determined by the point spread function (PSF) and the boundary condition (BC) imposed outside of the image. In this paper, we only consider the cases for zero and reflexive boundary conditions [5].

The Tikhonov regularization problem (1) is mathematically equivalent to solving the following equation:

$$\left(A^TA + \lambda^2 D^TD\right) x = A^Tb. \tag{2}$$

If the size of the original image X is $m \times n$, then the size of blurring matrix A is $mn \times mn$, which is very large and sparse when m and n are large. So, the linear

system (2) is usually solved using iterative methods such as CGLS, LSQR, and so on [1, 3, 10, 11].

If the PSF is separable (i.e., the PSF can be expressed as an outer product of two vectors), then A can be represented by the Kronecker product of A_r and A_c, i.e. $A = A_r \otimes A_c$, where $A_r \in \mathbb{R}^{n\times n}$ and $A_c \in \mathbb{R}^{m\times m}$. Here, the matrix A satisfying $A = A_r \otimes A_c$ is also called separable. If A and D are separable, then the large sparse linear system (2) can be transformed into small size of matrix equations. 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). Constructing such matrix equations will be 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 $\langle X,\ Y\rangle_F = tr(X^TY)$, where $tr(X^TY)$ denotes the *trace* of X^TY which is the sum of its main diagonal entries. Let H be a Hilbert space. A bounded linear operator $\mathcal{T} : H \to H$ is called *self-adjoint* if $\mathcal{T}^* = \mathcal{T}$, where $\mathcal{T}^*$ is the *adjoint operator* of $\mathcal{T}$ [7]. A self-adjoint operator $\mathcal{T} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is *positive definite* if

$$\langle X,\ \mathcal{T}(X)\rangle_F > 0 \ \text{ for all } \ X \neq O \ \text{ in } \ \mathbb{R}^{m\times n}.$$

A self-adjoint operator $\mathcal{T} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is said to be *positive definite on a subset* S *of* $\mathbb{R}^{m\times n}$ if $\langle X, \mathcal{T}(X)\rangle_F > 0$ for all $X \neq O$ in S.

3. Construction of linear operator equations

We first show how to construct linear operator equations corresponding to the Tikhonov regularization problems (1), where D is a discrete approximation of the first or second order partial derivative operators, and then we propose *Kronecker product preconditioners* which are required for the global preconditioned conjugate gradient (Gℓ-PCG) method. We assume that the blurring matrix A is nearly separable, that is, A can be represented or well approximated by Kronecker product of A_r and A_c. More specifically, $A = A_r \otimes A_c$ or $A \approx A_r \otimes A_c$, where $A_r \in \mathbb{R}^{n\times n}$ and $A_c \in \mathbb{R}^{m\times m}$. In this section, we only consider the case of $A = A_r \otimes A_c$ since the other case of $A \approx A_r \otimes A_c$ can be explained similarly.

3.1. Operator equation for D corresponding to the Laplacian (Case 1). We consider the Tikhonov regularization problem (1) for the case where D is an approximate matrix corresponding to the Laplacian $x_{ss} + x_{tt}$ of the image $X \in \mathrm{R}^{m\times n}$, where s and t denote the variable in the vertical direction and the horizontal direction, respectively. Then the matrix D can be expressed as

$$Dx = -(x_{ss} + x_{tt}) = I_n \otimes D_{2,m}\, x + D_{2,n} \otimes I_m\, x,$$

where $x = vec(X)$, I_n and I_m are the identity matrix of order n and m respectively, and $D_{2,m}$ and $D_{2,n}$ are $m \times m$ and $n \times n$ matrices obtained by finite difference approximations to the second order partial derivatives x_{ss} and x_{tt} [5].

More specifically, when $m = 4$, the matrix $D_{2,m}$ for zero boundary condition is given by

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

and the matrix $D_{2,m}$ for reflexive boundary condition is given by

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

Since $A = A_r \otimes A_c$ and $D = I_n \otimes D_{2,m} + D_{2,n} \otimes I_m$, the linear system (2) can be transformed into

$$\begin{aligned} \left\{ A_r^T A_r \otimes A_c^T A_c + \lambda^2 (I_n \otimes D_{2,m}^T D_{2,m} + D_{2,n} \otimes D_{2,m}^T \right. \\ \left. + D_{2,n}^T \otimes D_{2,m} + D_{2,n}^T D_{2,n} \otimes I_m) \right\} x = (A_r \otimes A_c)^T b. \end{aligned} \tag{3}$$

Let $X \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times n}$ be such that $x = vec(X)$ and $b = vec(B)$, and let a linear operator $\mathcal{A}_1 : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ be defined by

$$\mathcal{A}_1(X) = A_c^T A_c X A_r^T A_r + \lambda^2 (D_{2,m}^T D_{2,m} X + D_{2,m}^T X D_{2,n}^T + D_{2,m} X D_{2,n} + X D_{2,n}^T D_{2,n}).$$

Then (3) can be expressed as the following operator equation

$$\mathcal{A}_1(X) = A_c^T B A_r. \tag{4}$$

It can be shown that the linear operator $\mathcal{A}_1 : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ is self-adjoint and positive definite on a subset of $\mathbb{R}^{m \times n}$, i.e., $\langle X, \mathcal{A}_1(X) \rangle_F > 0$ for all $X \in \mathbb{R}^{m \times n}$ which is not a constant image, which is true for most of practical images X.

In order to accelerate the convergence of the Gℓ-CG, a good choice of preconditioner corresponding to the operator equation (4) is required. From the left side of the linear system (3), one can obtain the following approximate relation

$$\begin{aligned} &\left\{ A_r^T A_r \otimes A_c^T A_c + \lambda^2 (I_n \otimes D_{2,m}^T D_{2,m} + D_{2,n} \otimes D_{2,m}^T \right. \\ &\qquad \left. + D_{2,n}^T \otimes D_{2,m} + D_{2,n}^T D_{2,n} \otimes I_m) \right\} x \\ &\approx \left\{ \left(A_r^T A_r + \lambda (I_n + D_{2,n})^T (I_n + D_{2,n}) \right) \right. \\ &\qquad \left. \otimes \left(A_c^T A_c + \lambda (I_m + D_{2,m})^T (I_m + D_{2,m}) \right) \right\} x. \end{aligned} \tag{5}$$

From (5), we can choose a Kronecker product preconditioner of the form $M_1 = M_r \otimes M_c$, where

$$\begin{aligned} M_r &= A_r^T A_r + \lambda (I_n + D_{2,n})^T (I_n + D_{2,n}), \\ M_c &= A_c^T A_c + \lambda (I_m + D_{2,m})^T (I_m + D_{2,m}). \end{aligned}$$

Then it is clear that $M_r \in \mathbb{R}^{n\times n}$ and $M_c \in \mathbb{R}^{m\times m}$. Now we define a preconditioner operator $\mathcal{M}_1 : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ by

$$\mathcal{M}_1(X) = M_c X M_r^T. \tag{6}$$

It can be easily shown that the preconditioner operator $\mathcal{M}_1$ is self-adjoint and positive definite.

3.2. Operator equation for D corresponding to $\|x_s\|_2^2 + \|x_t\|_2^2$ (Case 2). We consider the Tikhonov regularization problem (1) for the case where D is an approximate matrix corresponding to $\|x_s\|_2^2 + \|x_t\|_2^2$, where s and t denote the variables in the vertical direction and the horizontal direction, respectively.

Let $D_{1,m}$ and $D_{1,n}$ be $m \times m$ and $n \times n$ matrices obtained by finite difference approximations to the first order partial derivatives x_s and x_t [5]. That is, when $m = 4$, the matrix $D_{1,m}$ for zero boundary condition is given by

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

and the matrix $D_{1,m}$ for reflexive boundary condition is given by

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

Consider the matrix D in the Tikhonov regularization problem (1) such that $\|Dx\|_2^2 = \|x_s\|_2^2 + \|x_t\|_2^2$. Then we can easily obtain

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

where $D_s = I_n \otimes D_{1,m}$ and $D_t = D_{1,n} \otimes I_m$. Thus, (1) can be transformed into the following form

$$\min_{x\in\mathbb{R}^N} \left\{ \left\| \begin{pmatrix} A \\ \lambda D_s \\ \lambda D_t \end{pmatrix} x - \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix} \right\|_2^2 \right\}. \tag{7}$$

It is easy to show that the minimization problem (7) is equivalent to solving the following equation

$$\left(A^T A + \lambda^2 D_s^T D_s + \lambda^2 D_t^T D_t\right) x = A^T b. \tag{8}$$

Since $A = A_r \otimes A_c$, $D_s = I_n \otimes D_{1,m}$ and $D_t = D_{1,n} \otimes I_m$, the linear system (8) can be rewritten as

$$\left\{ A_r^T A_r \otimes A_c^T A_c + \lambda^2 (I_n \otimes D_{1,m}^T D_{1,m}) + \lambda^2 (D_{1,n}^T D_{1,n} \otimes I_m) \right\} x = (A_r^T \otimes A_c^T) b. \tag{9}$$

Let $X \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times n}$ be such that $x = vec(X)$ and $b = vec(B)$, and let a linear operator $\mathcal{A}_2 : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be defined by

$$\mathcal{A}_2(X) = (A_c^T A_c)X(A_r^T A_r) + \lambda^2 \left((D_{1,m}^T D_{1,m})X + X(D_{1,n}^T D_{1,n})\right).$$

Then (9) can be expressed as the following operator equation

$$\mathcal{A}_2(X) = A_c^T B A_r. \tag{10}$$

It can be shown that the linear operator $\mathcal{A}_2 : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is self-adjoint and positive definite on a subset of $\mathbb{R}^{m\times n}$, i.e., $\langle X, \mathcal{A}_2(X)\rangle_F > 0$ for all $X \in \mathbb{R}^{m\times n}$ when at least one column of $X \in \mathbb{R}^{m\times n}$ is not a constant vector. Notice that most cases of practical images X satisfy that at least one column of $X \in \mathbb{R}^{m\times n}$ is not a constant vector.

From the left side of the linear system (9), one can obtain the following approximate relation

$$\begin{aligned}&\left(A_r^T A_r \otimes A_c^T A_c + \lambda^2(I \otimes D_{1,m}^T D_{1,m}) + \lambda^2(D_{1,n}^T D_{1,n} \otimes I)\right) x \\ &\approx \left(A_r^T A_r + \lambda(I + D_{1,n}^T D_{1,n})\right) \otimes \left(A_c^T A_c + \lambda(D_{1,m}^T D_{1,m} + I)\right) x. \end{aligned} \tag{11}$$

From (11), we can choose a Kronecker product preconditioner of the form $M_2 = M_r \otimes M_c$, where

$$M_r = A_r^T A_r + \lambda(I + D_{1,n}^T D_{1,n}) \text{ and } M_c = A_c^T A_c + \lambda(I + D_{1,m}^T D_{1,m}).$$

Now we define a preconditioner operator $\mathcal{M}_2 : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ by

$$\mathcal{M}_2(X) = M_c X M_r^T. \tag{12}$$

It can be easily shown that the preconditioner operator $\mathcal{M}_2$ is self-adjoint and positive definite.

4. Gℓ-PCG algorithm for solving the linear operator equation

By combining the ideas of the Gℓ-CG method [13] and the PCG method [6, 12], the following Gℓ-PCG algorithm for solving the linear operator equation $\mathcal{A}_i(X) = \mathcal{B}$ with the preconditioner operator $\mathcal{M}_i$ $(1 \le i \le 2)$ can be easily obtained, where $\mathcal{A}_i$ and $\mathcal{M}_i$ are the linear operators defined in Section 3 and $\mathcal{B} = A_c^T B A_r$.

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

1. Compute $R_0 := \mathcal{B} - \mathcal{A}_i(X_0), Z_0 = \mathcal{M}_i^{-1}(R_0), P_0 = Z_0$
2. For $j = 0, 1, \cdots$, until convergence Do :
   $\alpha_j := \langle R_j, Z_j\rangle_F / \langle \mathcal{A}_i(P_j), P_j\rangle_F$
   $X_{j+1} := X_j + \alpha_j P_j$
   $R_{j+1} := R_j - \alpha_j \mathcal{A}_i(P_j)$
   $Z_{j+1} := \mathcal{M}_i^{-1}(R_{j+1})$
   $\beta_j := \langle R_{j+1}, Z_{j+1}\rangle_F / \langle R_j, Z_j\rangle_F$
   $P_{j+1} := Z_{j+1} + \beta_j P_j$
3. End

Notice that $\mathcal{A}_i$ and $\mathcal{M}_i$ in Algorithm 1 are self-adjoint and positive definite operators. Moreover, if $\mathcal{M}_i$ in Algorithm 1 is chosen as an identity operator,

then Algorithm 1 reduces to the Gℓ-CG algorithm for solving the linear operator equation $\mathcal{A}_i(X) = \mathcal{B}$.

5. Parallel image deblurring using the Gℓ-PCG method

In this section, we propose a parallel image deblurring algorithm using the Gℓ-PCG method with Kronecker product preconditioners when the pixel size of the blurred and noisy image B is large. Assume that the PSF (point spread function) P is spatially invariant and separable. 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 blurred and 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

$$B = \begin{pmatrix} B_1 & B_2 & \cdots & B_\ell \end{pmatrix}, \quad X = \begin{pmatrix} X_1 & X_2 & \cdots & X_\ell \end{pmatrix},$$

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 small size of blurring matrices A_{r_k} and A_{c_k} corresponding to B_k from the PSF array P
Construct the regularization matrix $D^{(k)}$ corresponding to B_k
Construct Kronecker preconditioner M_{r_k} and M_{c_k} from A_{r_k}, A_{c_k} and $D^{(k)}$
Compute X_k by applying the Gℓ-PCG to the linear operator equation generated from A_{r_k}, A_{c_k} and $D^{(k)}$.

Finally, the true image X can be formed 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:

Algorithm 2 : Parallel algorithm using Gℓ-PCG method
parfor $k = 1$ to ℓ
Construct A_{r_k} and A_{c_k} corresponding to B_k from the PSF array P
Construct $D^{(k)}$ corresponding to B_k
Construct M_{r_k} and M_{c_k} from A_{r_k}, A_{c_k} and $D^{(k)}$
Apply the Gℓ-PCG to the linear operator equation generated from A_{r_k}, A_{c_k} and $D^{(k)}$ 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. If M_{r_k} and M_{c_k} in Algorithm 2 are chosen as identity matrices, then Algorithm 2 reduces to Parallel algorithm using Gℓ-CG method.

Since the Gℓ-PCG has a rich parallelism, we can easily parallelize the Gℓ-PCG algorithm itself, which is called a fine-grained parallel algorithm that is suitable for advanced parallel supercomputers. The algorithm provided in this section is a coarse-grained parallel algorithm suitable for personal computers with multiple cores which need a lot of communication time among the cores and overhead time.

6. Numerical experiments

In this section, we provide numerical experiments for several image deblurring problems to estimate the efficiency of the Gℓ-PCG method with Kronecker product preconditioners for solving the linear operator equation $\mathcal{A}_i(X) = \mathcal{B}$ with the preconditioner operator $\mathcal{M}_i$ $(1 \leq i \leq 2)$, where $\mathcal{A}_i$ and $\mathcal{M}_i$ are linear operators discussed in Section 3. We evaluate the effectiveness of the Gℓ-PCG by comparing its performance with those of the Gℓ-CG, CGLS and PCGLS methods (see Tables 1 to 4). We also provide parallel performance results for parallel algorithm proposed in Section 5 to evaluate its efficiency on a personal computer with 4 cores (see Tables 5 and 6).

All numerical tests have been performed using Matlab R2016b on a personal computer, which has 4 cores, equipped with Intel Core i5-4570 3.2GHz CPU and 8GB RAM. For numerical experiments, we have used 3 types of PSFs (point spread functions) which are Gaussian blur, Motion blur and Disk blur of size 7×7. The PSF array P for Gaussian blur of size 7×7 is generated by the Matlab function $fspecial('gaussian', [7, 7], 2)$. The PSF array P for Disk blur of size 7×7 is generated by the Matlab function $fspecial('disk', 3)$, and the PSF array P for Motion blur of size 7×7 is generated by the Matlab function

$$P = zeros(7);\ P(3:5, :) = fspecial('motion', 7, 1).$$

Notice that Gaussian blur are separable, but Disk blur and Motion blur are nonseparable. For a nonseparable PSF, we have used a separable PSF which is a rank-1 approximation to the nonseparable PSF using Kronecker product approximation techniques proposed in [8, 9]. So A can be expressed or approximated as $A_r \otimes A_c$ for all PSFs.

The blurred and noisy image B is generated by

$$vec(B) = A \cdot vec(X) + vec(E),$$

where A stands for the blurring matrix which can be generated by the original PSF array P according to the boundary condition to be used, and the noise E is a Gaussian white noise with mean 0 and standard deviation 0.75 which can be generated using Matlab function *randn*. That is,

$$E = 0.75 \times randn(m, n)$$

where (m, n) denotes the size of the true image X.

The initial image X_0 is set to the blurred and noisy image B. The stopping criterion for iterative methods at the k-th iterate is

$$\frac{\|R_k\|_F}{\|R_0\|_F} \leq 10^{-2}$$

where R_k represents the k-th residual matrix corresponding to the k-th iteration matrix X_k of iterative methods with R_0 the initial residual matrix corresponding to X_0. A restored image G is measured by the PSNR (Peak Signal to Noise

Ratio) which is defined by

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\max\limits_{i,j} |x_{ij}|^2 \cdot m \cdot n}{\|X - G\|_F^2} \right)$$

where $X = (x_{ij})$ represents the true image.

We have used 2 test images Lena and Joomaks for numerical experiments. The pixel size of Lena image is 512×512, and the pixel size of Joomaks image is 2200×2200. We have used two boundary conditions which are zero boundary condition and reflexive boundary condition. The preconditioners for the PCGLS method were chosen as follows: For the zero boundary condition we chose the BCCB (Block circulant with circulant blocks) approximation matrix which can be easily obtained using the DFT2 (2-dimensional discrete Fourier transform), and for the reflexive boundary condition we chose the symmetric approximation matrix which can be easily obtained using the DCT2 (2-dimensional discrete Cosine transform) (see [5] for details).

For the CGLS and PCGLS methods, the blurring matrix A whose size is large is not constructed for both zero and reflexive boundary conditions since its construction is very time-consuming and matrix-vector multiplication with A can be performed without constructing A (see [4] for details).

For the Gℓ-PCG method, the matrices A_r and A_c whose size is very small compared to the size of A are constructed for both zero and reflexive boundary conditions. Numerical experiments for parallel image deblurring algorithm using the Gℓ-PCG have been carried out only for Joomaks image of large size (see Tables 5 and 6). For Lena image of small size, we do not have performance gains from parallel execution since personal computer needs a lot of overhead time and communication time among the cores. We only provide parallel performance results for Gaussian and Motion PSFs since parallel performance behaviors for Disk PSF are similar.

In all Tables, "PSNR" represents the PSNR values for the restored images, PSNR_0 represents the PSNR values for the blurred and noisy images, "Itime" represents the elapsed CPU time in seconds required for iteration steps of Gℓ-CG, Gℓ-PCG, CGLS and PCGLS methods, "IT" represents the number of iterations required for the iterative methods, and "λ" represents a near optimal regularization parameter which is chosen by numerical tries.

In Tables 5 and 6, "ℓ" represents the number of Cores to be used, S_ℓ stands for the speedup of parallel execution on ℓ processors (or cores), and "Ttime" represents the elapsed total CPU time in seconds which is the sum of *Itime* and construction time for A_{r_k}, A_{c_k}, $D^{(k)}$, M_{r_k} and M_{c_k} in Algorithm 2. Notice that the construction time is much less than *Itime*. The notation $(k_1, k_2, \ldots, k_\ell)$ under the column labeled with *IT* refers to a collection of the number of iterations required for every core. That is, k_i indicates the number of iterations required for Core i.

As can be seen in Tables 1 to 4, Gℓ-CG and Gℓ-PCG with Kronecker product preconditioners restore the true image as well as CGLS and PCGLS except for the nonseparable Disk blur (see also Figure 1). The reason for worse performance for Disk blur is that Gℓ-CG and Gℓ-PCG use a rank-1 approximation to the Disk blur which is not a good approximation to the Disk blur. For all test problems, Gℓ-PCG with Kronecker product preconditioners yields a superior performance in terms of both convergence rate and execution time. For reflexive boundary condition, PCGLS has extremely faster convergence rate since the DCT2 type of preconditioner is a very good approximation to the original matrix. As compared with the Gℓ-CG, Kronecker product preconditioners for Gℓ-PCG proposed in this paper work extremely well in terms of convergence rate. This means that the Kronecker product preconditioner is a good approximation to the original matrix.

As can be seen in Tables 5 and 6, the speedup of parallel execution on 4 processors (or cores) ranges from 1.30 to 1.88 depending upon the amount of serial execution time (i.e., execution time on $\ell = 1$). This means that the more the serial execution time, the higher the speedup of parallel execution on $\ell = 4$. Since personal computers with multiple cores need a lot of communication time among the cores and overhead time, parallel speedup on personal computers is low. If the coarse-grained parallel algorithm is performed on advanced parallel supercomputers with 4 processors, then its parallel speedup may be close to 4. Notice that for parallel execution the number of iterations (i.e., IT) varies depending upon cores, but the differences among the cores are at most 2. Since computational amount of each core per iteration decreases as ℓ increases, small difference of IT does not affect overall parallel performance much. Also notice that PSNR values remain almost the same from parallel execution on 4 processors. This means that parallel execution on 4 processors does not deteriorate the quality of image deblurring (see Figure 2).

Table 1. Numerical results for Lena image (Case 1)

PSF	Method	Zero boundary condition					Reflexive boundary condition				
		$PSNR_0$	PSNR	λ	Itime	IT	$PSNR_0$	PSNR	λ	Itime	IT
Gaussian	PCGLS	27.57	31.39	0.025	1.93	13	28.78	32.47	0.03	0.47	1
	CGLS		31.91	0.02	2.27	19		32.39	0.025	1.43	20
	Gℓ-PCG		31.63	0.03	0.26	5		32.34	0.035	0.29	6
	Gℓ-CG		31.91	0.02	0.46	19		32.39	0.025	0.47	20
Motion	PCGLS	27.42	34.05	0.03	2.06	14	28.16	35.47	0.035	0.60	2
	CGLS		34.32	0.025	2.31	19		34.96	0.03	1.38	19
	Gℓ-PCG		34.40	0.025	0.26	5		35.20	0.03	0.29	6
	Gℓ-CG		34.22	0.025	0.46	19		34.86	0.03	0.45	19
Disk	PCGLS	27.59	32.18	0.03	1.69	11	28.77	33.50	0.03	0.47	1
	CGLS		32.56	0.025	2.20	18		33.30	0.035	1.27	17
	Gℓ-PCG		31.24	0.035	0.33	7		32.08	0.04	0.29	6
	Gℓ-CG		31.50	0.025	0.45	18		32.14	0.035	0.39	16

7. Conclusions

In this paper, we have studied performance of the Gℓ-PCG method with Kronecker product preconditioners for image deblurring problems with nearly

Table 2. Numerical results for Joomaks image (Case 1)

PSF	Method	Zero boundary condition					Reflexive boundary condition				
		$PSNR_0$	PSNR	λ	Itime	IT	$PSNR_0$	PSNR	λ	Itime	IT
Gaussian	PCGLS	23.83	28.06	0.015	44.6	16	23.98	28.21	0.015	5.70	1
	CGLS		27.79	0.015	60.9	27		27.87	0.015	35.0	28
	Gℓ-PCG		27.79	0.015	9.70	7		27.86	0.015	9.68	7
	Gℓ-CG		27.79	0.015	21.3	27		27.87	0.015	22.2	28
Motion	PCGLS	24.60	33.00	0.025	39.1	14	24.70	33.46	0.025	8.50	2
	CGLS		32.64	0.02	52.2	23		32.69	0.025	28.2	22
	Gℓ-PCG		32.90	0.02	8.52	6		32.94	0.025	8.51	6
	Gℓ-CG		32.46	0.02	19.0	23		32.58	0.025	18.4	23
Disk	PCGLS	23.73	30.26	0.02	31.7	11	23.87	30.59	0.02	5.73	1
	CGLS		29.88	0.02	53.9	24		30.03	0.02	31.6	25
	Gℓ-PCG		27.84	0.025	9.67	7		27.89	0.03	9.64	7
	Gℓ-CG		28.05	0.02	19.1	24		28.12	0.025	18.4	23

Table 3. Numerical results for Lena image (Case 2)

PSF	Method	Zero boundary condition					Reflexive boundary condition				
		$PSNR_0$	PSNR	λ	Itime	IT	$PSNR_0$	PSNR	λ	Itime	IT
Gaussian	PCGLS	27.57	31.95	0.045	1.66	11	28.78	32.39	0.05	0.62	2
	CGLS		32.05	0.035	2.16	18		32.34	0.04	1.38	19
	Gℓ-PCG		31.92	0.045	0.18	4		32.33	0.05	0.21	5
	Gℓ-CG		32.05	0.035	0.31	18		32.34	0.04	0.33	19
Motion	PCGLS	27.42	34.22	0.05	1.43	9	28.16	34.79	0.055	0.75	3
	CGLS		33.92	0.05	1.96	16		34.44	0.055	1.20	16
	Gℓ-PCG		34.15	0.05	0.18	4		34.65	0.05	0.17	4
	Gℓ-CG		33.89	0.055	0.28	16		34.37	0.055	0.28	16
Disk	PCGLS	27.59	32.69	0.055	1.31	8	28.77	33.20	0.055	0.62	2
	CGLS		32.72	0.045	2.06	17		33.18	0.05	1.26	17
	Gℓ-PCG		31.67	0.05	0.22	5		32.11	0.06	0.22	5
	Gℓ-CG		31.72	0.045	0.30	17		32.11	0.05	0.29	16

Table 4. Numerical results for Joomaks image (Case 2)

PSF	Method	Zero boundary condition					Reflexive boundary condition				
		$PSNR_0$	PSNR	λ	Itime	IT	$PSNR_0$	PSNR	λ	Itime	IT
Gaussian	PCGLS	23.83	27.93	0.025	39.8	14	23.98	28.12	0.03	5.56	1
	CGLS		27.80	0.02	65.8	29		27.82	0.02	37.0	30
	Gℓ-PCG		27.82	0.02	7.75	6		27.83	0.02	7.77	6
	Gℓ-CG		27.80	0.02	19.2	29		27.82	0.02	19.8	30
Motion	PCGLS	24.60	32.70	0.04	28.7	10	24.70	32.79	0.04	13.5	4
	CGLS		32.15	0.035	47.4	21		32.25	0.04	26.4	21
	Gℓ-PCG		32.32	0.035	5.81	4		32.33	0.04	5.65	4
	Gℓ-CG		32.08	0.04	14.1	21		32.13	0.04	14.1	21
Disk	PCGLS	23.73	29.90	0.035	26.2	9	23.87	29.96	0.03	8.23	2
	CGLS		29.72	0.03	53.7	24		29.68	0.035	28.7	23
	Gℓ-PCG		27.92	0.035	6.70	5		28.05	0.04	7.83	6
	Gℓ-CG		28.07	0.035	14.8	22		28.11	0.035	15.5	23

Table 5. Parallel performance results of Gℓ-CG for Joomaks image

D	PSF	ℓ	PSNR	λ	Ttime	IT	S_ℓ
Case 1	Gaussian	1	27.87	0.015	22.6	(28)	1
		2	27.86		23.2	(29,28)	0.97
		4	27.82		12.0	(29,29,28,28)	1.88
	Motion	1	32.58	0.025	18.8	(23)	1
		2	32.44		18.4	(23,23)	1.02
		4	32.19		10.6	(23,25,23,23)	1.77
Case 2	Gaussian	1	27.82	0.02	20.5	(30)	1
		2	27.80		20.7	(30,29)	0.99
		4	27.74		11.2	(31,30,30,30)	1.80
	Motion	1	32.13	0.04	14.7	(21)	1
		2	32.02		15.5	(21,21)	0.95
		4	31.82		8.53	(21,22,21,21)	1.72

separable PSFs. Gℓ-CG and Gℓ-PCG with Kronecker product preconditioners restore the true image as well as CGLS and PCGLS when the PSF is well

Table 6. Parallel performance results of Gℓ-PCG for Joomaks image

D	PSF	ℓ	PSNR	λ	Ttime	IT	S_ℓ
Case 1	Gaussian	1	27.86	0.015	10.27	(7)	1
		2	27.91		10.65	(8,7)	0.96
		4	27.85		6.59	(8,8,7,7)	1.56
	Motion	1	32.94	0.025	9.11	(6)	1
		2	32.74		9.05	(6,6)	1.01
		4	32.37		6.20	(6,7,6,6)	1.47
Case 2	Gaussian	1	27.83	0.02	7.97	(6)	1
		2	27.80		8.13	(6,6)	0.98
		4	27.74		5.37	(6,6,6,6)	1.48
	Motion	1	32.33	0.04	5.99	(4)	1
		2	32.28		6.92	(4,4)	0.87
		4	32.01		4.62	(5,5,4,4)	1.30

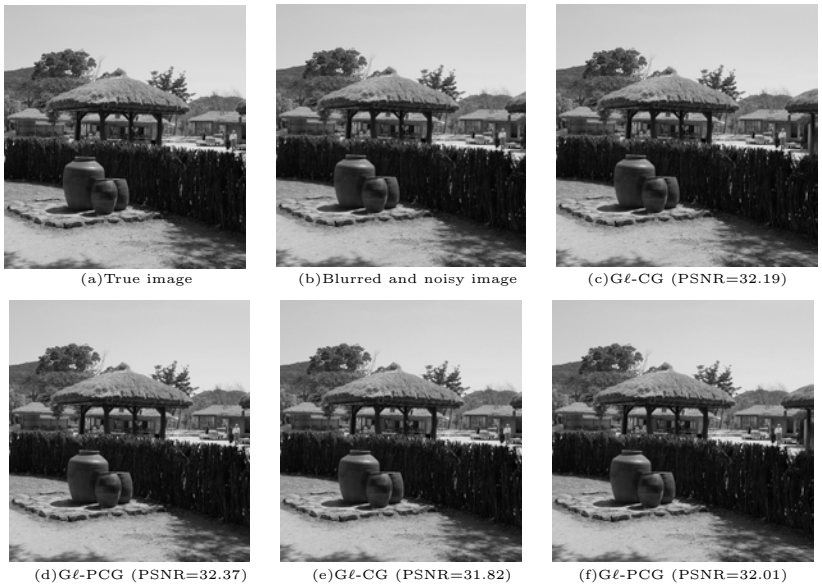
(a)True image (b)Blurred and noisy image (c)CGLS (PSNR=33.18)

(d)PCGLS (PSNR=33.20) (e)Gℓ-CG (PSNR=32.11) (f)Gℓ-PCG (PSNR=32.11)

Fig. 1. Lena image for Disk blur with reflexive BC ((c, d, e, f): restored images for Case 2 of D).

approximated by a rank-1 approximation to the PSF. For all test problems, Gℓ-PCG with Kronecker product preconditioners yields a superior performance in terms of both convergence rate and execution time.

The proposed coarse-grained parallel deblurring algorithm using Gℓ-PCG with Kronecker product preconditioners performs quite efficiently on a personal computer with large parallel overhead time. The speedup of parallel execution on 4 cores ranges from 1.30 to 1.88 depending upon the amount of serial execution time. If the coarse-grained parallel algorithm is performed on advanced parallel supercomputers with 4 processors, then its parallel speedup may be close to 4. Notice that parallel execution does not deteriorate the quality of image deblurring (see Tables 5 and 6).



(a)True image (b)Blurred and noisy image (c)Gℓ-CG (PSNR=32.19)

(d)Gℓ-PCG (PSNR=32.37) (e)Gℓ-CG (PSNR=31.82) (f)Gℓ-PCG (PSNR=32.01)

Fig. 2. Joomaks image for Motion blur with reflexive BC ((c, d): restored images on $\ell = 4$ for Case 1 of D, (e, f): restored images on $\ell = 4$ for Case 2 of D).

References

1. A. Bjorck, *Numerical methods for least squares problems*, SIAM, Philadelphia, 1996.
2. M. Donatelli, D. Martin and L. Reichel, *Arnoldi methods for image deblurring with anti-reflextive boundary conditions*, Appl. Math. Comput. **253** (2015), 135-150.
3. R.W. Freund, G.H. Golub and N.M. Nachtigal, *Iterative solutions of linear systems*, Acta Numerica **1** (1991), 57-100.
4. M. Hankey and J.G. Nagy, *Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient*, Inverse Problems **12** (1996), 157-173.
5. P.C. Hansen, J.G. Nagy and D.P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, Philadelphia, 2006.
6. M.R. Hestenes, E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards **49** (1952), 409-436.
7. E. Kreyszig, Introductory functional analysis with applications, John Wiley & Sons. Inc., New York, 1978.
8. J. Kamn and J.G. Nagy, *Optimal kronecker product approximation of block Toeplitz matrices*, SIAM J. Matrix Anal. Appl. **22** (2000), 155-172.
9. J.G. Nagy, M.N. Ng and L. Perrone, *Kronecker product approximations for image restoration with reflexive boundary conditions*, SIAM J. Matrix Anal. Appl. **25** (2004), 829-841.
10. J.G. Nagy, K.M. Palmer and L. Perrone, *Iterative methods for image deblurring: A MATLAB object oriented approach*, Numerical Algorithms **36** (2004), 73-93.
11. C.C. Paige and M.A. Saunders, *LSQR. An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software **8** (1982), 43-71.

12. Y. Saad, *Iterative methods for sparse linear systems*, PWS Publishing Company, Boston, 1996.
13. D.K. Salkuyeh, *CG-type algorithms to solve symmetrics matrix equations*, Appl. Math. Comput. **172** (2006), 985-999.

Jae Heon Yun received M.Sc. from Kyungpook National University, and Ph.D. from Iowa State University. He is currently a professor at Chungbuk National University since 1991. His research interests are scientific computing, iterative method and image processing.

Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju 28644, Korea
e-mail: gmjae@chungbuk.ac.kr