AIMS Mathematics, 7(10): 17989–18009.
DOI: 10.3934/math.2022991
Received: 20 April 2022
Revised: 07 July 2022
Accepted: 21 July 2022
Published: 05 August 2022

http://www.aimspress.com/journal/Math

Research article

Preconditioned augmented Lagrangian method for mean curvature image deblurring

Shahbaz Ahmad[1], **Faisal Fairag**[2], **Adel M. Al-Mahdi**[3,*] **and Jamshaid ul Rahman**[1]

[1] ASSMS, Government College University, Lahore 54000, Pakistan
[2] Department of Mathematics, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[3] PYP-Math, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

* **Correspondence:** Email: almahdi@kfupm.edu.sa.

Abstract: Image deblurring models with a mean curvature functional has been widely used to preserve edges and remove the staircase effect in the resulting images. However, the Euler-Lagrange equations of a mean curvature model can be used to solve fourth-order non-linear integro-differential equations. Furthermore, the discretization of fourth-order non-linear integro-differential equations produces an ill-conditioned system so that the numerical schemes like Krylov subspace methods (conjugate gradient etc.) have slow convergence. In this paper, we propose an augmented Lagrangian method for a mean curvature-based primal form of the image deblurring problem. A new circulant preconditioned matrix is introduced to overcome the problem of slow convergence when employing a conjugate gradient method inside of the augmented Lagrangian method. By using the proposed new preconditioner fast convergence has been observed in the numerical results. Moreover, a comparison with the existing numerical methods further reveal the effectiveness of the preconditioned augmented Lagrangian method.

Keywords: image deblurring; augmented Lagrangian method; mean curvature; ill-posed problem; Krylov subspace methods; preconditioned matrix
Mathematics Subject Classification: 68U10, 94A08, 65K10, 65N12

1. Introduction

Image deblurring or deconvolution is an important topic in image processing due to its wide range of applications in astronomical imaging, robot vision, medical image processing, remote sensing, virtual reality and many others. In the field of image deconvolution, the use of non-linear variational methods have received a great amount of attention in the last few decades. While applying these methods to

blurred and noisy images researchers have to face two main difficulties. One difficulty is non-linearity, and the other is related to solve the underlying large matrix system. For this paper, our primary goal is also to deal with these two computational difficulties for a high-order variational model. The total variation (TV) model [1, 29, 36]

$$\mathfrak{I}(u) = \frac{1}{2}\|\vec{K}u - z\|^2 + \alpha \int_\Omega |\nabla u| \, \mathbf{dx} \tag{1.1}$$

is the most famous non-linear variational model. It has so many desirable properties, but this model transforms smooth functions into piecewise constant functions which generates staircase effects in the deblurred images. That is why the resulting images look blocky. To overcome the problem of staircase effects, one idea is to use mean curvature (MC)-based regularization models [12, 19, 31, 39–41],

$$\mathfrak{I}(u) = \frac{1}{2}\|\vec{K}u - z\|^2 + \frac{\alpha}{2} \int_\Omega (\nabla \cdot \frac{\nabla u}{|\nabla u|})^2 \mathbf{dx}. \tag{1.2}$$

The MC-based regularization models not only remove the staircase effects but they also preserve the edges in the deblurred images. However, the Euler-Lagrange equations of mean curvature models can be used to solve non-linear fourth-order integro-differential equations. This non-linear high order term comes from the MC functional. Furthermore the discretization of an integro-differential equation results in a large ill-conditioned system that causes numerical algorithms like Krylov subspace methods (e.g. conjugate gradient (CG) method) to slowly converge. The Jacobian matrix of the ill-conditioned system has a block-banded structure with a large bandwidth. The efficient and robust numerical solution is a key issue for MC-based regularization methods due to the ill-conditioned system and high non-linearity.

In this paper, we propose an augmented Lagrangian method for an MC-based image deblurring problem. The augmented Lagrangian methods have already been successfully applied to the optimization problems in image processing and computer vision [9, 22, 32, 37]. In these works, the augmented Lagrangian methods achieve much higher speeds than other numerical methods. The augmented Lagrangian methods have been shown to decompose the original nontrivial minimization problem into a number of subproblems that are easy and fast to solve. Some of them can be solved by fast solvers like the fast Fourier transform (FFT), and others have 'closed-form' solutions. Therefore, the construction of an efficient augmented Lagrangian method for the minimization of a given functional depends on whether one can break down the original functional into simple ones. In our proposed augmented Lagrangian method, we use the cell-centered finite difference scheme along with the midpoint quadrature scheme for discretizing the subproblem for the primary variable u (image) of the associated Lagrangian equations. However, this leads to a large non-linear matrix system. The coefficient matrix of this system is symmetric positive definite (SPD). So the CG method is suitable for the solution of this matrix system. But, the SPD matrix is quite dense which causes the CG method to slowly converge. To overcome the problem of slow convergence due to the CG method, we have introduced a new circulant preconditioned matrix. So, for the solution of the system, instead of applying ordinary CG method, we use a preconditioned CG (PCG) method. By using the proposed new preconditioner fast convergence can be observed in the numerical results.

The contributions of the paper are as follows: (i) we present a preconditioned augmented Lagrangian method for an MC-based image deblurring model; (ii) we introduce a new circulant preconditioned

matrix that overcomes the problem of slow convergence due to the CG method inside the augmented Lagrangian scheme and (iii) we present a comparison with existing numerical methods to demonstrate the efficiency of the preconditioned augmented Lagrangian method. The paper includes different sections. The second section explains the image deblurring problem. The third section is about our proposed augmented Lagrangian method for an MC-based image deblurring model. The cell discretization and matrix system are also presented in third section. The fourth section describes the proposed circulant preconditioner, and the fifth section discusses the numerical experiments. We present the conclusions in the last section of the manuscript.

2. Problem description

We start the paper by presenting a concise description of the image debluring problem. The mathematical relationship of u (original or true image) and z (blurred image) is

$$z = \vec{K}u + \epsilon. \tag{2.1}$$

The parameter ϵ represents a noise function which can be Gaussian noise, Brownian noise, salt-and-pepper noise etc. In our experiments, we have only considered Gaussian noise. $\vec{K}$ represents the blurring operator,

$$(\vec{K}u)(x) = \int_{\Omega} k(x, y)u(y)\, dy, \quad \mathbf{x} \in \Omega, \tag{2.2}$$

where $k(x, y) = k(x - y)$ is known as translation-invariant kernel. Figure 1 depicts the process of blurring.

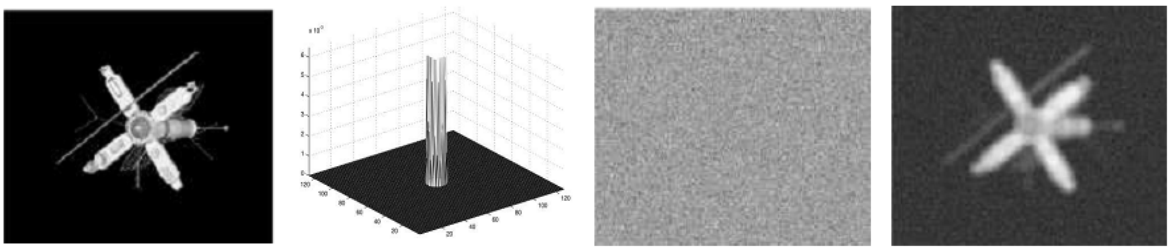

Figure 1. (left) True image, (center-left) blurring kernel, (center-right) noise and (right) the resulting image.

If $\vec{K} = I$ (identity operator), then Problem (2.1) becomes an image denoising problem. If the blurring operator $\vec{K}$ is known then the technique is referred to non-blind deconvolution [14, 34, 38]. However, if the blurring operator is unknown, then it is called blind deconvolution [6, 19, 23]. Our research will focus on non-blind deconvolution. $\vec{K}$ is a Fredholm-integral operator of the first kind, so it is compact. That is why Problem (2.1) becomes ill-posed [1, 35, 36]. The image intensity function u lies in a square $\Omega \subset \mathrm{R}^2$. The position of the pixel in Ω is defined by $\mathbf{x} = (x, y)$. Additionally, $|\mathbf{x}| = \sqrt{x^2 + y^2}$ and $\|\cdot\|$ are the Euclidean norm and $L_2(\Omega)$ norm respectively. The recovery of u from z makes Problem (2.1) an unstable inverse problem [1, 35, 36]. The use of the MC regularization functional [12, 31, 39, 41],

$$J(u) = \int_{\Omega} \kappa(u)^2 \mathbf{dx} = \int_{\Omega} (\bigtriangledown \cdot \frac{\bigtriangledown u}{|\bigtriangledown u|})^2 \mathbf{dx}, \tag{2.3}$$

makes Problem (2.1) a stable problem. The Problem (2.1) is then to find a u that minimizes the functional

$$\Im(u) = \frac{1}{2}\|\vec{K}u - z\|^2 + \frac{\alpha}{2} J(u), \quad \alpha > 0. \tag{2.4}$$

In [41], Zhu and Chan explained the well-posedness of Problem (2.4) for the case of synthetic image denoising. The Euler-Lagrange equations of Problem (2.4) are as follows:

$$\vec{K}^*(\vec{K}u - z) + \alpha \bigtriangledown \cdot (\frac{\bigtriangledown \kappa}{\sqrt{|\bigtriangledown u|^2 + \beta^2}} - \frac{\bigtriangledown \kappa \cdot \bigtriangledown u}{(\sqrt{|\bigtriangledown u|^2 + \beta^2})^3} \bigtriangledown u) = 0 \text{ in } \Omega, \tag{2.5}$$

$$\frac{\partial u}{\partial n} = 0 \text{ in } \partial\Omega, \tag{2.6}$$

$$\kappa(u) = 0 \text{ in } \partial\Omega, \tag{2.7}$$

where $\vec{K}^*$is the adjoint operator of $\vec{K}$ and n is the outward unit normal. $\beta > 0$ is added to make the MC functional differentiable at zero. Equation (2.5) is a non-linear fourth order differential equation.

The MC-based regularization models not only remove the staircase effects but they also preserve edges during the recovery of digital images. However, the Euler-Lagrange equations of the MC model can be used to solve the non-linear fourth order integro-differential equation. The MC functional generates a non-linear high-order term. When developing an efficient numerical method, one key issue is determining a proper approximation of the MC functional. In the next section, the non-linearity of MC is treated by introducing three new variables for the associated Lagrangian.

3. Augmented Lagrangian method

In the given literature [37, 40, 42, 43] one can find lot of work on the augmented Lagrangian method (ALM) for the MC-based image denoising problem but not for the image deblurring problem. In this paper, we have extended the ALM algorithm for application to the image deblurring problem. To develop an ALM for the image deblurring model described by Problem (2.4), we introduce three new variables, i.e., w, $\vec{v}$ and $\vec{t}$, and consider the following constrained minimization problem

$$min_{u,w,\vec{v},\vec{t}}\Big[\frac{1}{2}\int (\vec{K}u - z)^2 + \int_{\Omega} |w|\Big], \quad \text{with} \quad w = \bigtriangledown \cdot \vec{v}, \vec{v} = \frac{\vec{t}}{\sqrt{|\vec{t}|^2 + \beta}}, \vec{t} = \bigtriangledown u. \tag{3.1}$$

The associated augmented Lagrangian functional in Problem (3.1) is

$$\mathbf{L}(u, w, \vec{v}, \vec{t}, \lambda_1, \vec{\lambda}_2, \lambda_3) = \frac{1}{2}\int (\vec{K}u - z)^2 + \alpha \int |w| \tag{3.2}$$

$$+\frac{r_1}{2}\int (w - \bigtriangledown \cdot \vec{v})^2 + \int \lambda_1 (w - \bigtriangledown \cdot \vec{v})$$

$$+\frac{r_2}{2}\int|\vec{v}-\frac{\vec{t}}{\sqrt{|\vec{t}|^2+\beta}}|^2+\int\vec{\lambda}_2\cdot(\vec{v}-\frac{\vec{t}}{\sqrt{|\vec{t}|^2+\beta}})$$

$$+\frac{r_3}{2}\int(\vec{t}-\nabla u)^2+\int\vec{\lambda}_3\cdot(\vec{t}-\nabla u),$$

where r_1, r_2 and r_3 are the penalization parameters. $\lambda_1 \in \mathbb{R}$, $\vec{\lambda}_2, \vec{\lambda}_3 \in \mathbb{R}^3$ are Lagrange multipliers and $\vec{v}, \vec{t} \in \mathbb{R}^3$. One of the key issues with ALMs is related to handling the subproblems involving the term $\frac{\vec{t}}{\sqrt{|\vec{t}|^2+\beta}} = \frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}$. This type of term appears due to the mean curvature functional $\nabla \cdot \frac{\nabla u}{|\nabla u|}$. One remedy is to introduce a variable $\vec{t} = \langle \nabla u, 1\rangle$ instead of $\vec{t} = \nabla u$ for the curvature term. Accordingly, we will use $\vec{v} = \frac{\langle \nabla u, 1\rangle}{\sqrt{|\langle \nabla u, 1\rangle|^2+\beta}}$. Then, our MC model becomes

$$min_{u,w,\vec{v},\vec{t}}\Big[\frac{1}{2}\int(\vec{K}u-z)^2+\int_\Omega|w|\Big], \quad \text{with} \quad w=\nabla\cdot\vec{v}, \vec{v}=\frac{\vec{t}}{\sqrt{|\vec{t}|^2+\beta}}, \vec{t}=\langle\nabla u,1\rangle. \tag{3.3}$$

Then, the associated augmented Lagrangian functional is

$$\mathbf{L}(u,w,\vec{v},\vec{n},\vec{t},\lambda_1,\vec{\lambda}_2,\lambda_3,\vec{\lambda}_4)=\frac{1}{2}\int(\vec{K}u-z)^2+\alpha\int|w| \tag{3.4}$$

$$+r_1\int(|\vec{t}|-\vec{t}\cdot\vec{n})+\int\lambda_1(|\vec{t}|-\vec{t}\cdot\vec{n})$$

$$+\frac{r_2}{2}\int|\vec{t}-\langle\nabla u,1\rangle|^2+\int\vec{\lambda}_2\cdot(\vec{t}-\langle\nabla u,1\rangle)$$

$$+\frac{r_3}{2}\int(w-\partial_x v_1-\partial_y v_2)^2+\int\lambda_3\cdot(w-\partial_x v_1-\partial_y v_2)$$

$$+\frac{r_4}{2}\int|\vec{v}-\vec{n}|^2+\int\vec{\lambda}_4\cdot(\vec{v}-\vec{n})+\delta_R(\vec{n}),$$

where r_1, r_2, r_3 and r_4 are the penalty parameters. $\lambda_1, \lambda_3, \in \mathbb{R}$, $\vec{\lambda}_2, \vec{\lambda}_4 \in \mathbb{R}^3$ are Lagrange multipliers and $\vec{v}, \vec{n}, \vec{t} \in \mathbb{R}^3$. Here, we have introduced $\vec{n}$ to relax the variable $\vec{v}$. $R = \{\vec{n} \in L^2(\Omega) : |\vec{n}| \leq 1 \quad a.e. \quad n \in \Omega\}$ and $\delta_R(\vec{n})$ is a characteristic function on R that can be expressed as

$$\delta_R(\vec{n})=\begin{cases}0, & \vec{n}\in R\\ +\infty, & otherwise.\end{cases} \tag{3.5}$$

To make the term $|\vec{t}| - \vec{t}\cdot\vec{n}$ always non-negative, we need $\vec{n} \in R$. To find the saddle points of the minimization given by Problem (3.3), we need to find the saddle points of the augmented Lagrangian functional given by Eq (3.4). So, we have to fix all variables in Eq (3.4) except one particular variable. Then, we find a critical point of the induced functional to update that particular variable. We repeat the same process for all variables in Eq (3.4). The Lagrangian multipliers will automatically be advanced once all variables are updated. The process will continue until all variables in Eq (3.4) converge to a steady state. The ALM is summarized in the following algorithm.

Algorithm 1: The Augmented Lagrangian method for MC model

Step-I: Initialize $u^0, w^0, \vec{v}^0, \vec{n}^0, \vec{t}^0$, and $\lambda_1^0, \vec{\lambda}_2^0, \lambda_3^0, \vec{\lambda}_4^0$.

Step-II: For $k = 0, 1, 2, ...$: Compute $(u^k, w^k, \vec{v}^k, \vec{n}^k, \vec{t}^k)$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multiplier $\lambda_1^{k-1}, \vec{\lambda}_2^{k-1}, \lambda_3^{k-1}, \vec{\lambda}_4^{k-1}$, i.e.,

$$(u^k, w^k, \vec{v}^k, \vec{n}^k, \vec{t}^k) \approx argmin\mathbf{L}(u, w, \vec{v}, \vec{n}, \vec{t}, \lambda_1^{k-1}, \vec{\lambda}_2^{k-1}, \lambda_3^{k-1}, \vec{\lambda}_4^{k-1}). \tag{3.6}$$

Step-III: Update the Lagrangian multipliers

$$\lambda_1^k = \lambda_1^{k-1} + r_1(|\vec{t}^k| - \vec{t}^k \cdot \vec{n}^k), \tag{3.7}$$

$$\vec{\lambda}_2^k = \vec{\lambda}_2^{k-1} + r_2(|\vec{t}^k| - \langle \nabla u^k, 1\rangle), \tag{3.8}$$

$$\lambda_3^k = \lambda_3^{k-1} + r_3(w^k - \partial v_1^k - \partial v_2^k), \tag{3.9}$$

$$\vec{\lambda}_4^k = \vec{\lambda}_4^{k-1} + r_4(\vec{v}^k - \vec{n}^k), \tag{3.10}$$

where $\vec{n} = \langle n_1, n_2, n_3\rangle$.

Step-IV: Stop the iteration if relative residuals are smaller than a threshold ϵ_r.

Now we consider the following subproblems and find their critical points

$$f_1(u) = \frac{1}{2}\int (\vec{K}u - z)^2 + \frac{r_2}{2}\int |\vec{t} - \langle \nabla u, 1\rangle|^2 + \int \vec{\lambda}_2.(\vec{t} - \langle \nabla u, 1\rangle), \tag{3.11}$$

$$f_2(w) = \alpha \int |w| + \frac{r_3}{2}\int (w - \partial_x v_1 - \partial_y v_2)^2 + \int \lambda_3 \cdot (w - \partial_x v_1 - \partial_y v_2), \tag{3.12}$$

$$f_3(\vec{t}) = r_1 \int (|\vec{t}| - \vec{t}\cdot \vec{n}) + \int \lambda_1(|\vec{t}| - \vec{t}.\vec{n}) + \frac{r_2}{2}\int |\vec{t} - \langle \nabla u, 1\rangle|^2 + \int \vec{\lambda}_2.(\vec{t} - \langle \nabla u, 1\rangle), \tag{3.13}$$

$$f_4(\vec{v}) = \frac{r_3}{2}\int (w - \partial_x v_1 - \partial_y v_2)^2 + \int \lambda_3 \cdot (w - \partial_x v_1 - \partial_y v_2) + \frac{r_4}{2}\int |\vec{v} - \vec{n}|^2 + \int \vec{\lambda}_4.(\vec{v} - \vec{n}), \tag{3.14}$$

$$f_5(\vec{n}) = r_1 \int (|\vec{t}| - \vec{t}\cdot \vec{n}) + \int \lambda_1(|\vec{t}| - \vec{t}\cdot \vec{n}) + \frac{r_4}{2}\int |\vec{v} - \vec{n}|^2 + \int \vec{\lambda}_4 \cdot (\vec{v} - \vec{n}) + \delta_R(\vec{n}). \tag{3.15}$$

The standard procedure gives us the following Euler-Lagrange equations for the functionals $f_1(u)$ and $f_4(\vec{v})$:

$$K^*Ku - \triangle u = K^*z - (r_2 t_1 + \lambda_{21})_x - (r_2 t_2 + \lambda_{22})_y, \tag{3.16}$$

$$r_4 v_1 - r_3(\partial_x v_1 + \partial_y v_2)_x = r_4 n_1 - \lambda_{41} - (r_3 w - \lambda_3)_x, \tag{3.17}$$

$$r_4 v_2 - r_3(\partial_x v_1 + \partial_y v_2)_y = r_4 n_2 - \lambda_{42} - (r_3 w - \lambda_3)_y, \tag{3.18}$$

$$r_4 v_3 = r_4 n_3 - \lambda_{43}, \tag{3.19}$$

respectively, where $\vec{v} = \langle v_1, v_2, v_3\rangle, \vec{t} = \langle t_1, t_2, t_3\rangle, \vec{n} = \langle n_1, n_2, n_3\rangle, \vec{\lambda_2} = \langle \lambda_{21}, \lambda_{22}, \lambda_{23}\rangle$ and $\vec{\lambda_4} = \langle \lambda_{41}, \lambda_{42}, \lambda_{43}\rangle$. To update the functionals $f_2(w)$, $f_3(t)$ and $f_5(\vec{n})$, we need the following equations:

$$argmin_w \quad f_2(w) = max\Big\{0, 1 - \frac{\alpha}{r_3|\tilde{w}|}\Big\}\tilde{w}, \quad \tilde{w} = \partial_x v_1 + \partial_y v_2 - \frac{\lambda_3}{r_3}, \tag{3.20}$$

$$argmin_{\vec{t}} \quad f_3(\vec{t}) = max\Big\{0, 1 - \frac{r_1 + \lambda_1}{r_2|\vec{t}|}\Big\}\tilde{t}, \quad \tilde{t} = \langle \nabla u, 1\rangle - \frac{\vec{\lambda_2}}{r_2} + \frac{r_1 + \lambda_1}{r_2}\vec{n}, \tag{3.21}$$

$$argmin_{\vec{n}} \quad f_5(\vec{n}) = max\begin{cases} \tilde{n}, & |\tilde{n}| \le 1 \\ \tilde{n}/|\tilde{n}|, & otherwise, \end{cases} \quad \tilde{n} = \vec{v} + \frac{\vec{\lambda_4}}{r_4} + \frac{r_1 + \lambda_1}{r_4}\vec{t}. \tag{3.22}$$

To update the Lagrangian multipliers, we have the following equations:

$$\lambda_1^{new} = \lambda_1^{old} + r_1(|\vec{t}| - \vec{t}\cdot\vec{n}), \tag{3.23}$$

$$\vec{\lambda}_2^{new} = \vec{\lambda}_2^{old} + r_2(\vec{t} - \langle \nabla u, 1\rangle), \tag{3.24}$$

$$\lambda_3^{new} = \lambda_3^{old} + r_3(w - \partial v_1 - \partial v_2), \tag{3.25}$$

$$\vec{\lambda}_4^{new} = \vec{\lambda}_4^{old} + r_4(\vec{v} - \vec{n}). \tag{3.26}$$

3.1. Cell discretization

Since the MC regularizer $\kappa(u)^2 = (\nabla \cdot \frac{\nabla u}{|\nabla u|})^2$ of the image deblurring problem is not homogeneous in u, we need a spatial kind of discretization especially for the terms that involve derivatives. This is because the role of the spatial mesh size is quite important to the numerical performance of the MC model. So, we partitioned the domain $\Omega = (0, 1) \times (0, 1)$ by $\delta_x \times \delta_y$. Additionally,

$$\delta_x : 0 = x_{1/2} < x_{3/2} < ... < x_{n_x+1/2} = 1,$$
$$\delta_y : 0 = y_{1/2} < y_{3/2} < ... < y_{n_x+1/2} = 1,$$

where the number of equispaced partitions in the direction of x or y is equal to n_x and (x_i, y_j) represents the centers of the cells. Additionally,

$$x_i = \frac{(2ih - h)}{2} \quad \text{for} \quad i = 1, 2, 3, ..., n_x,$$

$$y_j = \frac{(2jh - h)}{2} \quad \text{for} \quad j = 1, 2, 3, ..., n_x,$$

where $h = \frac{1}{n_x}$; $(x_{i\pm\frac{1}{2}}, y_j)$ and $(x_i, y_{j\pm\frac{1}{2}})$ represent the midpoints of cell edges:

$$x_{i\pm\frac{1}{2}} = x_i \pm \frac{h}{2} \quad \text{for} \quad i = 1, 2, 3, ..., n_x,$$

$$y_{j\pm\frac{1}{2}} = y_j \pm \frac{h}{2} \quad \text{for} \quad j = 1, 2, 3, ..., n_x.$$

For each $i = 1, 2, ..., n_x$, and $j = 1, 2, ..., n_x$, define

$$\Omega_{i,j} = (x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2}).$$

For the function $\theta(x, y)$, let $\theta_{k,l}$ denote $\theta(x_l, y_m)$, where k and l may take values of $i-1, i$, or $i+1$ and $j-1, j$, or $j+1$ respectively, for integers $i, j \geq 0$. For backward and forward discrete functions, we need values at proper discrete points, so we define

$$[d_x^+\theta]_{i,j} = \frac{\theta_{i+1,j} - \theta_{i,j}}{h}, \quad [d_x^-\theta]_{i,j} = \frac{\theta_{i,j} - \theta_{i-1,j}}{h},$$
$$[d_y^+\theta]_{i,j} = \frac{\theta_{i,j+1} - \theta_{i,j}}{h}, \quad [d_y^-\theta]_{i,j} = \frac{\theta_{i,j} - \theta_{i,j-1}}{h}.$$

Then the central difference discrete functions and discrete gradient are

$$[d_x^c\theta]_{i,j} = \frac{[d_x^-\theta]_{i,j} + [d_x^+\theta]_{i,j}}{2}, \quad [d_y^c\theta]_{i,j} = \frac{[d_y^-\theta]_{i,j} + [d_y^+\theta]_{i,j}}{2},$$

and

$$[\nabla^+\theta]_{i,j} = \langle [d_x^+\theta]_{i,j}, [d_y^+\theta]_{i,j} \rangle, \quad [\nabla^-\theta]_{i,j} = \langle [d_x^-\theta]_{i,j}, [d_y^-\theta]_{i,j} \rangle,$$

respectively. By applying midpoint quadrature approximation, we have that

$$(Ku)(x_i, y_j) \cong [K_h U]_{(ij)}.$$

3.2. Matrix system

Here, we consider the cell-centered finite-difference (CCFD) method for the MC-based image deblurring problem. The CCFD approximations $\{U_{i,j}\}$ to $\{u(x_{i,j})\}$ are chosen. So from Eq (3.16) we have that

$$[K^*KU]_{i,j} + r_2[div^- \nabla^+ U]_{i,j} = [K^*Z]_{i,j} - [d_x^-(r_2t_1 + \lambda_{21})]_{i,j} - [d_y^-(r_2t_2 + \lambda_{22})]_{i,j},$$

$$[K^*KU]_{i,j} + r_2[div^- \nabla^+ U]_{i,j} = [K^*Z]_{i,j} + g(i, j), \tag{3.27}$$

where $g(i, j) = -[d_x^-(r_2t_1 + \lambda_{21})]_{i,j} - [d_y^-(r_2t_2 + \lambda_{22})]_{i,j}$. According to the lexicographical ordering of the unknowns, $U = [U_{11} \quad U_{12} \quad ... \quad U_{n_x n_x}]^t$. Then, from Eq (3.27), we have the following matrix system:

$$(K_h^*K_h + r_2B_h^*B_h)U = K_h^*Z + G_h. \tag{3.28}$$

Here K_h is a matrix of size $n_x^2 \times n_x^2$ and B_h is a matrix of size $2n_x(n_x - 1) \times n_x^2$. The matrices K_h^*Z and G_h are of size $n_x^2 \times 1$. $K_h^*K_h$ is symmetric positive semidefinite. The matrix K_h is a block Toeplitz matrix with Toeplitz blocks (BTTB). The structure of the matrix B_h is

$$B_h = \frac{1}{h}\begin{bmatrix} B_1 \\ B_2 \end{bmatrix},$$

where the size of both B_1 and B_2 is $n_x(n_x - 1) \times n_x^2$, and

$$B_1 = C \otimes I, \qquad B_2 = I \otimes C.$$

The size of

$$C = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & -1 & \\ & & & & 1 & -1 \end{bmatrix}$$

is $(n_x - 1) \times n_x$ and I is an identity matrix. To get the value of u, one has to solve the large matrix system (3.28).

Now given Eqs (3.17) and (3.18), after discretizing, we have that

$$r_4 v_1(i,j) - r_3([d_x^+ d_x^- v_1]_{i,j} + [d_x^+ d_y^- v_2]_{i,j}) = r_4 n_1(i,j) - \lambda_{41}(i,j) - [d_x^+ d_x^+ (r_3 w - \lambda_3)]_{i,j}, \tag{3.29}$$

$$r_4 v_2(i,j) - r_3([d_x^+ d_x^- v_1]_{i,j} + [d_x^+ d_y^- v_2]_{i,j}) = r_4 n_2(i,j) - \lambda_{42}(i,j) - [d_x^+ d_y^+ (r_3 w - \lambda_3)]_{i,j}. \tag{3.30}$$

Then, applying a discrete Fourier transformation followed by a discrete inverse Fourier transformation to Eqs (3.29) and (3.30), one can get v_1 and v_2, and v_3 can be obtained directly from Eq (3.19). To update the variables $w, \vec{t}, \vec{n}$ and Lagrange multipliers $\lambda_1, \vec{\lambda}_2, \lambda_3, \vec{\lambda}_4$, we have to discretize Eqs (3.20)–(3.26) on grid points (i, j). So we need the following equations:

$$w(i,j) = max\Big\{0, 1 - \frac{\alpha}{r_3|\tilde{w}(i,j)|}\Big\}\tilde{w}(i,j), \quad \tilde{w}(i,j) = \partial_x v_1(i,j) + \partial_y v_2(i,j) - \frac{\lambda_3}{r_3}, \tag{3.31}$$

$$\vec{t}(i,j) = max\Big\{0, 1 - \frac{r_1 + \lambda_1}{r_2|\tilde{t}(i,j)|}\Big\}\tilde{t}(i,j), \tag{3.32}$$

$$\tilde{t}(i,j) = \langle d_x^+ u(i,j), d_y^+ u(i,j), 1\rangle - \frac{\langle \lambda_{21}, \lambda_{22}, \lambda_{23}\rangle(i,j)}{r_2} + \frac{r_1 + \lambda_1}{r_2}\langle n_1, n_2, n_3\rangle(i,j),$$

$$\vec{n}(i,j) = max\begin{cases} \tilde{n}(i,j), & |\tilde{n}| \le 1 \\ \tilde{n}(i,j)/|\tilde{n}(i,j)|, & otherwise, \end{cases} \tag{3.33}$$

$$\tilde{n}(i,j) = \vec{v}(i,j) + \frac{\vec{\lambda}_4(i,j)}{r_4} + \frac{r_1 + \lambda_1}{r_4}\vec{t}(i,j).$$

To update the Lagrangian multipliers, we have the following equations:

$$\lambda_1^{new}(i,j) = \lambda_1^{old}(i,j) + r_1(|\vec{t}(i,j)| - \vec{t}(i,j).\vec{n}(i,j)), \tag{3.34}$$

$$\vec{\lambda}_{21}^{new}(i,j) = \vec{\lambda}_{21}^{old}(i,j) + r_2(t_1(i,j) - d_x^- u(i,j)), \tag{3.35}$$

$$\vec{\lambda}_{22}^{new}(i,j) = \vec{\lambda}_{22}^{old}(i,j) + r_2(t_2(i,j) - d_y^- u(i,j)), \tag{3.36}$$

$$\vec{\lambda}_{23}^{new}(i,j) = \vec{\lambda}_{23}^{old}(i,j) + r_2(t_3(i,j) - 1), \tag{3.37}$$

$$\lambda_3^{new}(i,j) = \lambda_3^{old}(i,j) + r_3(w(i,j) - d_x^- v_1(i,j) - d_y^- v_2(i,j)), \tag{3.38}$$

$$\vec{\lambda}_{41}^{new}(i,j) = \vec{\lambda}_{41}^{old}(i,j) + r_4(v_1(i,j) - n_1(i,j)) \tag{3.39}$$

$$\vec{\lambda}_{42}^{new}(i,j) = \vec{\lambda}_{42}^{old}(i,j) + r_4(v_2(i,j) - n_2(i,j)) \tag{3.40}$$

$$\vec{\lambda}_{43}^{new}(i,j) = \vec{\lambda}_{43}^{old}(i,j) + r_4(v_3(i,j) - n_3(i,j)). \tag{3.41}$$

4. Preconditioned matrix

To find the value of our main variable u, one has to solve the matrix system given by Eq (3.28). The Hessian matrix $K_h^*K_h + r_2B_h^*B_h$ of the system given by Eq (3.28) is extremely large for practical application and tends to be quite ill-conditioned when r_2 is small. This happens because of the eigenvalues of the blurring operator $\bar{K}$ are very small and close to zero [36]. $K_h^*K_h$ is a full matrix, but an FFT can be used to evaluate $K_h^*K_hu$ in $O(n_xlogn_x)$ operations [36] because the blurring operator $\bar{K}$ is translation-invariant. The good thing is that the Hessian matrix is SPD.

Theorem 4.1. The inner product $\langle \bar{A}U, U\rangle$ is positive for any matrix $U \neq 0$, where $\bar{A} = K_h^*K_h + r_2B_h^*B_h$ is the Hessian matrix of the system given by Eq (3.28).

Proof. For any matrix $U \neq 0$, consider that

$$\begin{aligned}\langle \bar{A}U, U\rangle &= \langle (K_h^*K_h + r_2B_h^*B_h)U, U\rangle \\ &= \langle K_h^*K_hU, U\rangle + r_2\langle B_h^*B_hU, U\rangle \\ &= \langle K_hU, K_hU\rangle + r_2\langle B_hU, B_hU\rangle \\ &= \|K_hU\|^2 + r_2\|B_hU\|^2 > 0.\end{aligned}$$

This completes the proof.

The CG method is suitable for the solution of the system given by Eq (3.28) owing to the above-mentioned properties. But, the CG method have a quite slow convergence rate due to the system being large and ill-conditioned system. So, we use a PCG method [7, 8, 10, 24–26, 30]. The preconditioning matrix P must be SPD so that we get an effective solution for our system [2–4, 11, 36]. Here, we introduce our SPD circulant preconditioned matrix P of the Strang-type [27].

$$P = \tilde{K}_h^*\tilde{K}_h + \gamma I_h, \tag{4.1}$$

where $\tilde{K}_h$ is a circulant approximation of Matrix K_h and $\gamma > 0$. I_h is an identity matrix. While applying the PCG to Eq (3.28), one of the requirements is to take the inverse of the preconditioned matrix P. Since the second term in P is an identity matrix I_h, inversion is not a problem. For the inversion of the first term $\tilde{K}_h^*\tilde{K}_h$, we need to apply FFTs to $O(n_xlogn_x)$ floating-point operations [36].

Now, let $\bar{A} = K_h^*K_h + r_2B_h^*B_h$ be the Hessian matrix of the system given by Eq (3.28). Let the eigenvalues of $K_h^*K_h$ and $B_h^*B_h$ be λ_i^K and λ_i^B respectively such that $\lambda_i^K \to 0$ and $\lambda_i^B \to \infty$. Then the eigenvalues of $P^{-1}\bar{A}$ are

$$\theta_i = \frac{\lambda_i^K + r_2\lambda_i^B}{\lambda_i^K + \gamma}. \tag{4.2}$$

One can notice that clearly $\theta_i \to \frac{\lambda_i^B}{\gamma}$ as $i \to \infty$. Hence, for $\lambda_i^B \leq \gamma$, the spectrum of $P^{-1}\bar{A}$ is more favorable than the Hessian matrix $\bar{A}$. This can also be observed in the numerical examples when we use the PCG algorithm.

5. Numerical experiments

Here, we present numerical examples for the image deblurring problem. In all experiments, we have used different values of n_x, and the resulting matrix system has n_x^2 unknowns. Then the mesh

size is $h = 1/n_x$. The numerical computations were performed using MATLAB software on an Intel$^{(R)}$ Core$^{(TM)}$ i7-4510U CPU @ 2.00 GHz 2.60 GHz. In all experiments, we have taken the initial guess to be the zero vector. The value of the parameters α and β were set by referencing [5, 41]. To observe the optimum value of γ, we have done numerical experiments by using Goldhills image. We found that the value of the peak signal-to-noise ratio (PSNR) does not show much improvement after $\gamma = 1$. So, one can use optimum value of γ close to one. The results of the experiments are given in Figure 2.

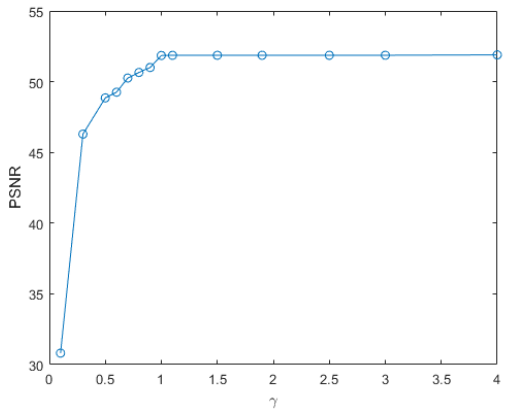

Figure 2. γ vs PSNR.

The PSNR is used to measure the quality of the restored images. For numerical calculations, we have used the $ke\_gen(n_x, r, \sigma)$ kernel [13, 16, 17, 28]. It is a circular Gaussian filter of size $n_x \times n_x$ with a standard deviation σ and radius r. Figure 3 depicts the $ke\_gen(120, 40, 4)$ kernel.

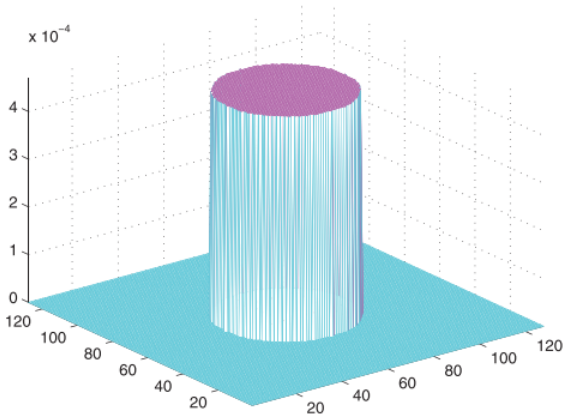

Figure 3. $ke\_gen(120, 40, 4)$ kernel.

Example 1.

For this example, we have used three different types of images. These are Goldhills, Kids and Peppers images (see [33]). The Goldhills image is a real image and Peppers image is a nontexture image. The Kids image is a complicated image, because it contains a small scale texture part (shirt) and a large scale cartoon part (face). The different aspects of all images can be seen in Figure 5. Each

subfigure has a pixel of 256×256. Here, we have also compared our MC-based ALMs, i.e., ALM and preconditioned ALM (PALM) with the TV-based method (TVM) [1, 29, 36]. The Hessian matrix of the TV-based model is SPD, so for the numerical calculation we have used the CG method . The parameters of ALM and PALM were established by referencing [42, 43]. The parameters for the ALM and PALM were $\alpha = 8e-9, r_1 = 9.5e-7, r_2 = 1e-6, r_3 = 1e-8, r_4 = 1e-5, \beta = 1$ and $\gamma = 1$. For the TVM algorithm (CG), we have used $\alpha = 1e-6$ and $\beta = 1$, as according to [36]. For the numerical calculations, the $ke\_gen(n_x, 300, 10)$ kernel was used. The stopping criteria for the numerical methods is based on the tolerance $tol = 1e-7$. The Tables 1–3 contain all of the information of this experiment. A performance comparison of the TVM, ALM and PALM is depicted in Figure 7. The relative residues and eigenvalues of the ALM and PALM are depicted in Figures 4 and 6, respectively.

Remarks:

(1) From Figure 5 one can notice that the deblurred images produced via our methods (ALM and PALM) are much better than those produced via the TVM.

(2) From Tables 1–3, one can observe that the PSNR values of the ALM and PALM methods were quite higher than those for the TVM for all images. Although the TVM generates its PSNR value in much less computational time as compared to the ALM and PALM, its PSNR value is quite low. For example, for the Goldhills image with a pixel size of 128×128, the TVM took 37.1002 seconds to generate its 48.5749 PSNR, but the ALM and PALM generated higher PSNR values of 56.3756 and 56.3654 in 40.8027 seconds and 30.5428 seconds respectively. For the Kids image with a pixel size of 256×256, the TVM took 133.0289 seconds to generate a 40.5482 PSNR, but the ALM and PALM generated higher PSNR values of 46.0595 and 46.9837 in 189.2586 seconds and 141.2561 seconds respectively. Similarly, for the Peppers image with a pixel size of 256×256, the TVM took 118.5327 seconds to generate 49.2863 PSNR, but the ALM and PALM generated higher PSNR values of 51.0284 and 51.4522 in 190.3504 seconds and 146.3177 seconds respectively. Similar behavior was observed for other sizes. So, the proposed ALM and PALM tend to generate high-quality deblurred images as compared to the TVM.

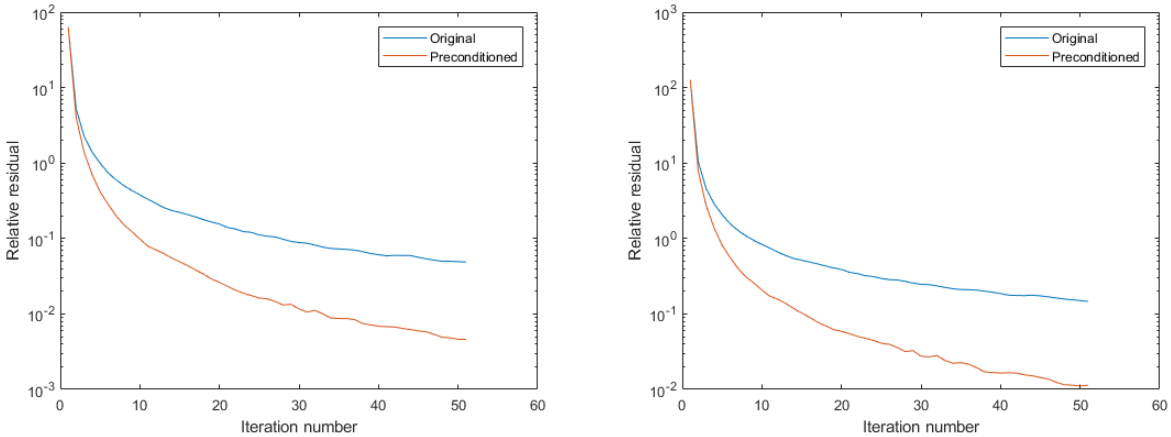

Figure 4. Convergence of CG (original) and PCG (preconditioned) for the inner iteration count for the Goldhills image. Pixel size of (left) 128×128 and (right) 256×256.

Figure 5. (first row) Blurry images, (second row) TVM deblurred images, (third row) ALM deblurred images and (last row) PALM deblurred images.

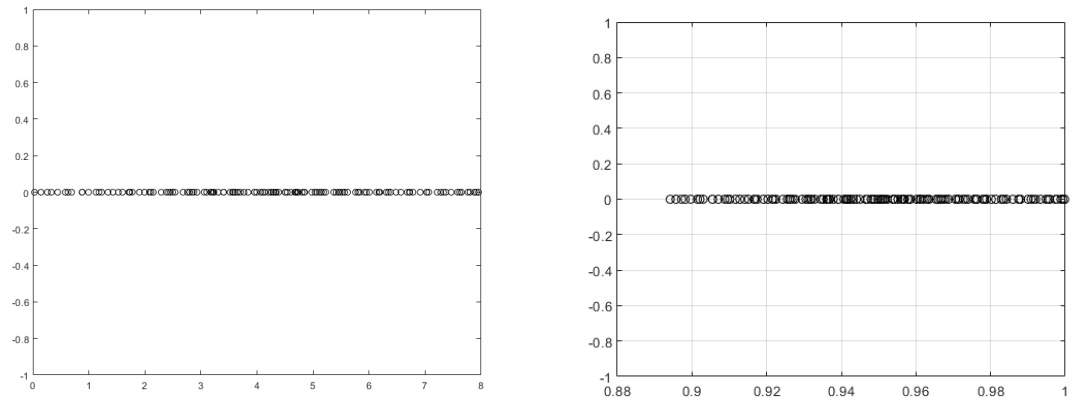

Figure 6. (left) Eigenvalues of $\bar{A}$ and (right) eigenvalues of $P^{-1}\bar{A}$.

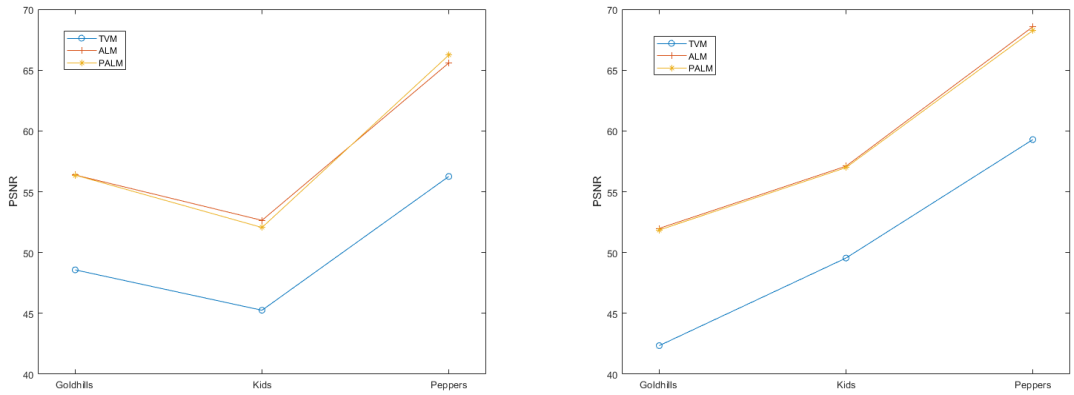

Figure 7. Performance of TVM, ALM and PALM on different images. Pixel size of (left) 128×128 and (right) 256×256.

Table 1. Comparison of TVM, ALM and PALM performance on the Goldhills image.

Image	Pixel size	Blurred PSNR	Method	Deblurred PSNR	CPU time
Goldhills	128×128	22.9784	TVM	48.5749	37.1002
	128×128	22.9784	ALM	56.3756	40.8027
	128×128	22.9784	PALM	56.3654	30.5428
	256×256	22.2335	TVM	42.3448	146.2581
	256×256	22.2335	ALM	51.9872	195.2980
	256×256	22.2335	PALM	51.8512	126.7891

Table 2. Comparison of TVM, ALM and PALM performance on the Kids image.

Image	Pixel size	Blurred PSNR	Method	Deblurred PSNR	CPU time
Kids	128×128	20.4859	TVM	45.2541	21.0522
	128×128	20.4859	ALM	52.1174	40.1252
	128×128	20.4859	PALM	52.0586	23.5469
	256×256	20.4147	TVM	40.5482	133.0289
	256×256	20.4147	ALM	46.0595	189.2586
	256×256	20.4147	PALM	46.9837	141.2561

Table 3. Comparison of TVM, ALM and PALM performance on the Peppers image.

Image	Pixel size	Blurred PSNR	Method	Deblurred PSNR	CPU time
Peppers	128×128	20.5545	TVM	46.2567	30.2596
	128×128	20.5545	ALM	49.7693	37.2904
	128×128	20.5545	PALM	49.1567	32.2563
	256×256	20.5531	TVM	49.2863	118.5327
	256×256	20.5531	ALM	51.0284	190.3504
	256×256	20.5531	PALM	51.4522	146.3177

Example 2.

Here, we have compared our MC-based ALMs, i.e., the ALM and PALM with the MC-based method proposed by Fairag et al. [18], who developed a One-Level method (OLM) and Two-Level method (TLM) for the MC-based image deblurring problem. For this experiment, we used three different types of images, i.e., the images of Goldhills (real), Cameraman (complicated) and Moon (non-texture). The different aspects of all images can be seen in Figure 8. Each subfigure has apixel size of 256×256.

The parameters for ALM and PALM were $\alpha = 1e{-}9, r_1 = 9.5e{-}7, r_2 = 1e{-}6, r_3 = 1e{-}8, r_4 = 1e{-}5$, $\beta = 1$ and $\gamma = 1$. For the MC-based algorithms (OLM and TLM) we used $\alpha = 8e - 9$ and $\beta = 1$. The Level-II parameter $\tilde{\alpha}$ of the TLM was calculated by referencing the formula presented in [18]. For the numerical calculations, the $ke\_gen(n_x, 300, 10)$ kernel was used. The stopping criteria for the numerical methods is based on the tolerance $tol = 1e - 8$. Tables 4–6 contain all of the information on this experiment. A performance comparison of the ALM, PALM, OLM and TLM is depicted in Figure 9.

Remarks:

(1) From Figure 8, one can notice that the ALM and PALM methods both generated slightly higher quality results.

(2) From Tables 4–6, it can be observed that the PSNR values of the ALM and PALM were higher than the OLM and TLM for all images. The TLM generated its PSNR in much less time than the OLM and ALM, but not the PALM. For example, for the Goldhills image with a pixel size of 256×256, the OLM, TLM and ALM took 248.5400, 128.4097 and 195.2980 seconds, respectively, to deblur the image. But, the PALM only required 126.7891 seconds for deblurring. For the Cameraman image with a pixel size of 128×128, the OLM, TLM and ALM took 24.5327, 19.7541 and 38.4161 seconds, respectively, to deblurred image. But, the PALM only required 18.2997 seconds for deblurring. Similarly, for Moon image with a pixel size of 128×128, the OLM, TLM and ALM took 34.3866, 24.8366 and 25.3032 seconds, respectively, to deblur image. But, the PALM only required 16.4356 seconds for deblurring. Similar behavior was observed for other sizes. So, our PALM consumed much less CPU time than other methods, and it also generated high-quality deblurred images.

Figure 8. (first row) Blurry images, (second row) OLM deblurred images, (third row) TLM deblurred images, (fourth row) ALM deblurred images and (last row) PALM deblurred images.

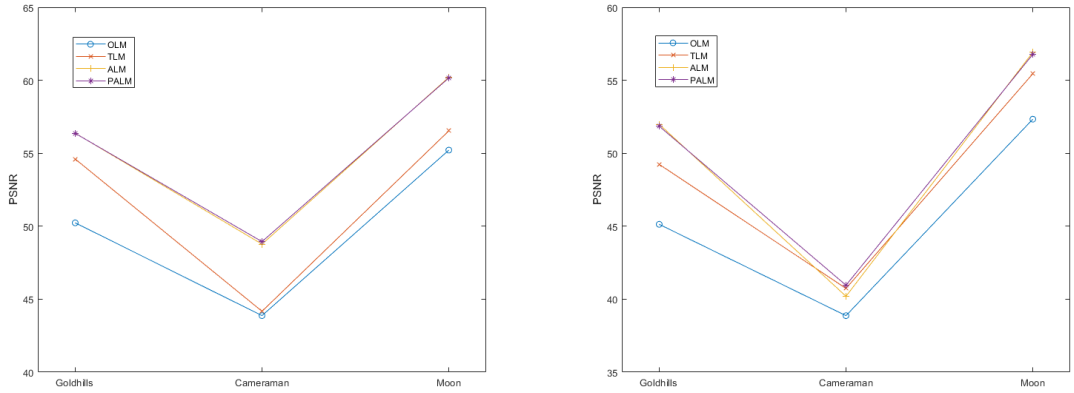

Figure 9. Performance of OLM, TLM, ALM and PALM on different images. Pixel size of (left) 128×128 and (right) 256×256.

Table 4. Comparison of OLM, TLM, ALM and PALM performance on the Goldhills image.

Image	Pixel size	Blurred PSNR	Method	Deblurred PSNR	CPU time
Goldhills	128×128	22.9784	OLM	50.2303	47.9473
	128×128	22.9784	TLM	54.6042	31.0442
	128×128	22.9784	ALM	56.3756	40.8027
	128×128	22.9784	PALM	56.3654	30.5428
	256×256	22.2335	OLM	45.1287	248.5400
	256×256	22.2335	TLM	49.2369	128.4097
	256×256	22.2335	ALM	51.9872	195.2980
	256×256	22.2335	PALM	51.8512	126.7891

Table 5. Comparison of OLM, TLM, ALM and PALM performance on the Cameraman image.

Image	Pixel size	Blurred PSNR	Method	Deblurred PSNR	CPU time
Cameraman	128×128	18.6322	OLM	43.8732	24.5327
	128×128	18.6322	TLM	44.1709	19.7541
	128×128	18.6322	ALM	48.7714	38.4161
	128×128	18.6322	PALM	48.9437	18.2997
	256×256	17.8172	OLM	38.8709	197.7316
	256×256	17.8172	TLM	40.7359	112.5587
	256×256	17.8172	ALM	40.1991	164.2474
	256×256	17.8172	PALM	40.9556	108.2873

Table 6. Comparison of OLM, TLM, ALM and PALM performance on the Moon image.

Image	Pixel size	Blurred PSNR	Method	Deblurred PSNR	CPU time
Moon	128×128	26.1840	OLM	55.2128	34.3866
	128×128	26.1840	TLM	56.5319	24.8366
	128×128	26.1840	ALM	60.2041	25.3032
	128×128	26.1840	PALM	60.1598	16.4356
	256×256	26.4905	OLM	52.3328	179.7899
	256×256	26.4905	TLM	55.4471	125.3007
	256×256	26.4905	ALM	56.9144	148.3155
	256×256	26.4905	PALM	56.7752	119.1213

6. Conclusions

An ALM for the primal form of the image deblurring problem with MC regularizational functional has been presented. A new SPD circulant preconditioned matrix has been introduced to overcome the problem of slow convergence due to the CG method inside of the ALM. Numerical experiments were conducted using different kinds of images (synthetic, real, complicated and nontexture) by our proposed PALM with a new preconditioned matrix. We have compared the TV (total variation) based algorithm with our mean curvature based (ALM and PALM) algorithm. The proposed ALMs, i.e., the ALM and PALM were also compared with the latest MC-based techniques i.e., the OLM and TLM. The numerical experiments showed the effectiveness of our proposed PALM method with a new preconditioned matrix.

In the future, we will work on developing an ALM for other computationally expensive regularization functionals like the fractional TV regularized tensor [21]. Furthermore, the matrix $B_h^* B_h$ was also constructed to have a block Toeplitz matrix structure in the system given by Eq (3.28), so, a single term preconditioner like tau preconditioner [15,20] can be used to approximate both the matrices $B_h^* B_h$ and $K_h^* K_h$ in order to increase the convergence rate of the CG method inside the ALM.

Acknowledgments

The second and third authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for its continuous support. The authors also thank the referees for their very careful reading and valuable comments. This work was funded by KFUPM, Grant No. #SB201012.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. R. Acar, C. R. Vogel, Analysis of bounded variation penalty methods for ill-posed problems, *Inverse Probl.*, **10** (1994), 1217–1229. https://doi.org/10.1088/0266-5611/10/6/003

2. S. Ahmad, A. M. Al-Mahdi, R. Ahmed, Two new preconditioners for mean curvature-based image deblurring problem, *AIMS Math.*, **6** (2021), 13824–13844.
3. S. Ahmad, F. Fairag, Circulant preconditioners for mean curvature-based image deblurring problem, *J. Algorithms Comput.*, **15** (2021).
4. M. Benzi, G. H. Golub, A preconditioner for generalized saddle point problems, *SIAM J. Matrix Anal. A.*, **26** (2004), 20–41.
5. C. Brito-Loeza, K. Chen, V. Uc-Cetina, Image denoising using the gaussian curvature of the image surface, *Numer. Meth. Part. D. E.*, **32** (2016), 1066–1089. https://doi.org/10.1002/num.22042
6. P. Campisi, K. Egiazarian, *Blind image deconvolution: Theory and applications*, CRC press, 2016.
7. R. H. Chan, Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions, *IMA J. Numer. Anal.*, **11** (1991), 333–345.
8. R. H. Chan, K. P. Ng, Toeplitz preconditioners for Hermitian Toeplitz systems, *Linear Algebra Appl.*, **190** (1993), 181–208. https://doi.org/10.1016/0024-3795(93)90226-E
9. S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, T. Q. Nguyen, An augmented lagrangian method for total variation video restoration, *IEEE T. Image Process.,* **20** (2011), 3097–3111. https://doi.org/10.1109/TIP.2011.2158229
10. T. F. Chan, An optimal circulant preconditioner for Toeplitz systems, *SIAM J. Sci. Comput.*, **9** (1988), 766–771. https://doi.org/10.1137/0909051
11. C. Chen, C. Ma, A generalized shift-splitting preconditioner for saddle point problems, *Appl. Math. Lett.*, **43** (2015), 49–55. https://doi.org/10.1016/j.aml.2014.12.001
12. K. Chen, Introduction to variational image-processing models and applications, *Int. J. Comput. Math.*, **90** (2013), 1–8.
13. K. Chen, F. Fairag, A. Al-Mahdi, Preconditioning techniques for an image deblurring problem, *Numer. Linear Algebra,* **23** (2016), 570–584. https://doi.org/10.1002/nla.2040
14. N. R. Choi, *A comparative study of non-blind and blind deconvolution of ultrasound images*, Dissertations Theses Gradworks, University of Southern California, 2014.
15. F. Di Benedetto, Solution of Toeplitz normal equations by sine transform based preconditioning, *Linear Algebra Appl.*, **285** (1998), 229–255. https://doi.org/10.1016/S0024-3795(98)10115-5
16. F. Fairag, S. Ahmad, *A two-level method for image deblurring problem*, in 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), IEEE, 2019, 1–5.
17. F. Fairag, K. Chen, S. Ahmad, Analysis of the ccfd method for mc-based image denoising problems, *Electron. T. Numer. Ana.,* **54** (2021), 108–127. https://doi.org/10.1553/etna_vol54s108
18. F. Fairag, K. Chen, C. Brito-Loeza, S. Ahmad, A two-level method for image denoising and image deblurring models using mean curvature regularization, *Int. J. Comput. Math.*, 2021, 1–21.
19. X. Ge, J. Tan, L. Zhang, Y. Qian, Blind image deconvolution via salient edge selection and mean curvature regularization, *Signal Process.*, **190** (2022), 108336.
20. X. M. Gu, Y. L. Zhao, X. L. Zhao, B. Carpentieri, Y. Y. Huang, A note on parallel preconditioning for the all-at-once solution of Riesz fractional diffusion equations, *Numer. Math. Theor. Meth. Appl.*, **14** (2021), 893–919. https://doi.org/10.4208/nmtma.OA-2020-0020

21. L. Guo, X. L. Zhao, X. M. Gu, Y. L. Zhao, Y. B. Zheng, T. Z. Huang, Three-dimensional fractional total variation regularized tensor optimized model for image deblurring, *Appl. Math. Comput.,* **404** (2021), 126224.
22. C. Li, W. Yin, H. Jiang, Y. Zhang, An efficient augmented lagrangian method with applications to total variation minimization, *Comput. Optim. Appl.*, **56** (2013), 507–530. https://doi.org/10.1007/s10589-013-9576-1
23. L. Li, J. Pan, W. S. Lai, C. Gao, N. Sang, M. H. Yang, *Learning a discriminative prior for blind image deblurring*, IEEE conference on computer vision and pattern recognition, 2018, 6616–6625.
24. F. R. Lin, Preconditioners for block Toeplitz systems based on circulant preconditioners, *Numer. Algorithms*, **26** (2001), 365–379. https://doi.org/10.1023/A:1016674923507
25. F. R. Lin, W. K. Ching, Inverse Toeplitz preconditioners for Hermitian Toeplitz systems, *Numer. Linear Algebra*, **12** (2005), 221–229. https://doi.org/10.1002/nla.397
26. F. R. Lin, C. X. Wang, Bttb preconditioners for Bttb systems, *Numer. Algorithms,* **60** (2012), 153–167. https://doi.org/10.1007/s11075-011-9516-z
27. M. K. Ng, *Iterative methods for Toeplitz systems*, London, U.K.: Oxford University Press, 2004.
28. K. L. Riley, *Two-level preconditioners for regularized ill-posed problems*, PhD Thesis, Montana State University, 1999.
29. L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D,* **60** (1992), 259–268.
30. D. K. Salkuyeh, M. Masoudi, D. Hezari, On the generalized shift-splitting preconditioner for saddle point problems, *Appl. Math. Lett.,* **48** (2015), 55–61. https://doi.org/10.1007/s10986-015-9265-0
31. L. Sun, K. Chen, A new iterative algorithm for mean curvature-based variational image denoising, *BIT Numer. Math.*, **54** (2014), 523–553.
32. X. C. Tai, J. Hahn, G. J. Chung, A fast algorithm for Euler's elastica model using augmented lagrangian method, *SIAM J. Imaging Sci.*, **4** (2011), 313–344. https://doi.org/10.1137/100803730
33. X. C. Tai, K. A. Lie, T. F. Chan, S. Osher, *Image processing based on partial differential equations: Proceedings of the international conference on PDE-based image processing and related inverse problems, CMA, Oslo, August 8–12, 2005*, Springer Science & Business Media, 2006.
34. S. Tao, W. Dong, H. Feng, Z. Xu, Q. Li, Non-blind image deconvolution using natural image gradient prior, *Optik*, **124** (2013), 6599–6605. https://doi.org/10.1016/j.ijleo.2013.05.068
35. A. N. Tikhonov, Regularization of incorrectly posed problems, *Soviet Math. Dokl.,* **4** (1963), 1624–1627.
36. C. R. Vogel, M. E. Oman, Fast, robust total variation-based reconstruction of noisy, blurred images, *IEEE T. Image Process.*, **7** (1998), 813–824. https://doi.org/10.1109/83.679423
37. C. Wu, X. C. Tai, Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models, *SIAM J. Imaging Sci.*, **3** (2010), 300–339. https://doi.org/10.1137/090767558
38. N. Xiong, R. W. Liu, M. Liang, D. Wu, Z. Liu, H. Wu, Effective alternating direction optimization methods for sparsity-constrained blind image deblurring, *Sensors*, **17** (2017), 174. https://doi.org/10.3390/s17010174

39. F. Yang, K. Chen, B. Yu, D. Fang, A relaxed fixed point method for a mean curvature-based denoising model, *Optim. Meth. Softw.*, **29** (2014), 274–285.

40. J. Zhang, C. Deng, Y. Shi, S. Wang, Y. Zhu, A fast linearised augmented lagrangian method for a mean curvature based model, *E. Asian J. Appl. Math.*, **8** (2018), 463–476.

41. W. Zhu, T. Chan, Image denoising using mean curvature of image surface, *SIAM J. Imaging Sci.*, **5** (2012), 1–32.

42. W. Zhu, X. C. Tai, T. Chan, Augmented lagrangian method for a mean curvature based image denoising model, *Inverse Probl. Imag.*, **7** (2013), 1409–1432.

43. W. Zhu, X. C. Tai, T. Chan, *A fast algorithm for a mean curvature based image denoising model using augmented lagrangian method*, in Efficient Algorithms for Global Optimization Methods in Computer Vision, Springer, 2014, 104–118.

AIMS Press