

ANALYSIS OF THE CCFD METHOD FOR MC-BASED IMAGE DENOISING PROBLEMS*

FAISAL FAIRAG[†], KE CHEN[‡], AND SHAHBAZ AHMAD[†]

Abstract. Image denoising using mean curvature leads to the problem of solving a nonlinear fourth-order integro-differential equation. The nonlinear fourth-order term comes from the mean curvature regularization functional. In this paper, we treat this high-order nonlinearity by reducing the nonlinear fourth-order integro-differential equation to a system of first-order equations. Then a cell-centered finite difference scheme is applied to this system. With a lexicographical ordering of the unknowns, the discretization of the mean curvature functional leads to a block pentadiagonal matrix. Our contributions are fourfold: (i) we give a new method for treating the high-order nonlinearity term; (ii) we express the discretization of this term in terms of simple matrices; (iii) we give an analysis for this new method and establish that the error is of first order; and (iv) we verify this theoretical result by illustrating the convergence rates in numerical experiments.

Key words. image denoising, mean curvature, cell-centered finite difference method, numerical analysis

AMS subject classifications. 68U10, 94A08, 65N06, 65N12

1. Introduction. As the focus of the paper is on image denoising problems, we start by presenting its concise description. Although there exist different types of noise, here we only study algorithms for removing additive, zero-mean Gaussian noise. Mathematically, the relationship between the original image u and a recorded image z is as follows:

$$(1.1) \quad z = u + \varepsilon,$$

where ε is the noise function. Let Ω denote a square in \mathbb{R}^2 , and let $u \in \Omega$ be an image intensity function. Additive noise in images is apparent in the form of random high-frequency oscillations. Therefore, energy minimization-based denoising techniques, represented in the form of the optimization problem

$$\min_u \left\{ T(u) = \frac{1}{2} \|u - z\|^2 + \alpha J(u) \right\},$$

attempt to damp such oscillations by the regularization term $J(u)$, and the key for noise removal is to select a suitable $J(u)$ capable of efficiently measuring oscillations. Different choices for $J(u)$ have been proposed, for example, the well-known total variation (TV) model [1, 6, 10]. The major drawback of this model is that it converts smooth functions into piecewise constant functions resulting in a phenomenon known as the staircase effect, which causes images to look blocky. To reduce the staircase effects, one remedy is to use mean curvature-based (MC-based) regularization functionals [4, 8, 11, 12, 13], which are very effective. These MC-based models not only remove staircase effects but also preserve edges in the recovery of digital images. However, fourth-order derivatives appear in the Euler-Lagrange equations of these models, which create problems in developing an efficient numerical algorithm. The nonlinear fourth-order term comes from the mean curvature functional. Furthermore, MC-based models produce complicated nonlinear systems of equations.

In the literature, one can find many numerical techniques that have been applied to the nonlinear minimization of mean curvature-based image denoising problems. Among them are

*Received May, 1, 2020. Accepted September 29, 2020. Published online on January 8, 2021. Recommended by Fiorella Sgallari.

[†]Department of Mathematics and Statistics, KFUPM, Saudi Arabia
({ffairag, shahbazahmad}@kfupm.edu.sa).

[‡]Center for Mathematical Imaging Techniques and Department of Mathematical Sciences, The University of Liverpool, Liverpool L69 7ZL, UK (k.chen@liv.ac.uk).

augmented Lagrangian methods [13], time marching schemes [12], and multigrid methods [2], etc. Almost all of these numerical techniques deal with the complexity of the MC functional by adding an artificial time step. In this paper we present a cell-centered finite difference (CCFD) method for mean curvature-based image denoising problems. One key issue in the scheme is to provide a proper approximation of the nonlinear mean curvature functional. We have treated this difficulty by reducing the nonlinear fourth-order integro-differential equation to a system of first-order equations. Then the CCFD scheme is applied to this first-order system. With lexicographical ordering of the unknowns, the discretization of the mean curvature functional leads to a block pentadiagonal matrix. We demonstrate that the proposed scheme is of first order in the discrete L_2 -norm on a uniform rectangular grid. We also carried out some numerical experiments using the proposed CCFD scheme, and the numerical results show that our method is robust.

The paper is organized in different sections. Section 1 is an introduction, while Section 2 includes a problem description of image denoising models. In Section 3, we present a nonlinear system of first-order equations for an MC-based image denoising model. The cell discretization and the CCFD method are also presented in this section. In Section 4, we provide an error analysis for our scheme. The algorithm and numerical results are given in Section 5. Conclusions about the proposed scheme are discussed in Section 6. The last section includes appendices, where we collect supplementary material for the proof of Theorem 4.1, which demonstrates that the proposed scheme is of first order in the discrete L_2 -norm.

2. Mean curvature-based denoising models. Problem (1.1) is an inverse problem. The recovery of u from z makes (1.1) an ill-posed problem [1, 9, 10]. To turn it into a well-posed one, a remedy is to use the mean curvature (MC) regularization functional [3, 5, 12, 13],

$$J(u) = \int_{\Omega} \kappa(u)^2 dx = \int_{\Omega} (\nabla \cdot \frac{\nabla u}{|\nabla u|})^2 dx.$$

Then the regularized problem for (1.1) takes the form: find u that minimizes the functional

$$(2.1) \quad T(u) = \frac{1}{2} \|u - z\|^2 + \frac{\alpha}{2} J(u),$$

where $\alpha > 0$ is a regularization parameter. The well-posedness of problem (2.1) for a particular case (synthetic image) is explained in [12]. The Euler-Lagrange equations for (2.1) are

$$(2.2) \quad (u - z) + \alpha \nabla \cdot \left[\frac{\nabla \kappa}{\sqrt{|\nabla u|^2 + \beta^2}} - \frac{\nabla \kappa \cdot \nabla u}{(\sqrt{|\nabla u|^2 + \beta^2})^3} \nabla u \right] = 0 \quad \text{in } \Omega,$$

with

$$\frac{\partial u}{\partial n} = 0 \quad \text{and} \quad \kappa(u) = 0 \quad \text{on } \partial\Omega,$$

where $\beta > 0$ is used to avoid non-differentiability at zero. Equation (2.2) is a nonlinear fourth-order differential equation.

Similarly, we can define a corresponding mean curvature-based one-dimensional signal denoising problem. For convenience, we still denote the original signal and the noisy signal by u and z , respectively. Both are defined on a closed interval $I = [a, b]$. In this case, the mean curvature functional becomes

$$\kappa(u) = \left(\frac{u_x}{\sqrt{u_x^2 + \beta^2}} \right)_x,$$

and the Euler-Lagrange equation is

$$(2.3) \quad (u - z) + \alpha \left(\kappa_x \frac{1}{(\sqrt{u_x^2 + \beta^2})^3} \right)_x = 0,$$

with

$$u_x(0) = u_x(1) = 0 \quad \text{and} \quad \kappa(0) = \kappa(1) = 0.$$

The MC-based model has nice properties. However, fourth-order derivatives appear in the Euler-Lagrange equations, which create problems when developing an efficient numerical algorithm. In the literature [2, 12, 13], one can find many numerical techniques that have been applied to the problem of image denoising by mean curvature-based nonlinear minimization. Almost all of the previous numerical techniques deal with the complexity of the MC functional by adding an artificial time step. In this paper we present a cell-centered finite difference (CCFD) method for mean curvature-based image denoising that does not require artificial time steps. One key problem in developing the scheme is to give a proper approximation of the nonlinear mean curvature functional. We treat this difficulty by reducing the nonlinear fourth-order differential equation to a system of first-order equations.

3. The first-order nonlinear system. We introduce the following four new variables

$$\vec{v} = \frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta^2}}, \quad w = \nabla \cdot \vec{v}, \quad \vec{p} = \frac{\nabla w}{\sqrt{|\nabla w|^2 + \beta^2}}, \quad \text{and} \quad \vec{t} = \frac{(\nabla w \cdot \vec{v}) \vec{v}}{\sqrt{|\nabla w|^2 + \beta^2}}.$$

Hence, equation (2.2) can be expressed as the first-order nonlinear system,

$$(3.1) \quad u + \alpha \nabla \cdot \vec{p} - \alpha \nabla \cdot \vec{t} = z,$$

$$(3.2) \quad -w + \nabla \cdot \vec{v} = 0,$$

$$(3.3) \quad \sqrt{|\nabla u|^2 + \beta^2} \vec{v} - \nabla u = 0,$$

$$(3.4) \quad \sqrt{|\nabla w|^2 + \beta^2} \vec{p} - \nabla w = 0,$$

$$(3.5) \quad \sqrt{|\nabla u|^2 + \beta^2} \vec{t} - (\nabla w \cdot \vec{v}) \vec{v} = 0.$$

In the one-dimensional case, we have to introduce fewer unknowns to transform the problem into a first-order nonlinear system because (2.3) is simpler than (2.2). Hence, from (2.3), we obtain the following system of four equations:

$$(3.6) \quad u + \alpha p_x = z,$$

$$(3.7) \quad -w + v_x = 0,$$

$$(3.8) \quad \sqrt{u_x^2 + \beta^2} v - u_x = 0,$$

$$(3.9) \quad (\sqrt{u_x^2 + \beta^2})^3 p - w_x = 0.$$

3.1. Cell discretization. For the two-dimensional problem, the domain $\Omega = (0, 1) \times (0, 1)$ is discretized into a rectangular grid by the partition $\delta_x \times \delta_y$, [7], where

$$\delta_x : 0 = x_{1/2} < x_{3/2} < x_{5/2} < \dots < x_{n_x-1/2} < x_{n_x+1/2} = 1,$$

$$\delta_y : 0 = y_{1/2} < y_{3/2} < y_{5/2} < \dots < y_{n_y-1/2} < y_{n_y+1/2} = 1.$$

Here n_x denotes the number of equispaced partitions in the x - or y -directions. Defining

$$x_i = (i - \frac{1}{2})h, \quad i = 1, 2, 3, \dots, n_x, \quad \text{and} \quad y_j = (j - \frac{1}{2})h, \quad j = 1, 2, 3, \dots, n_x,$$

and

$$x_{i \pm \frac{1}{2}} = x_i \pm \frac{h}{2}, \quad i = 1, 2, 3, \dots, n_x, \quad \text{and} \quad y_{j \pm \frac{1}{2}} = y_j \pm \frac{h}{2}, \quad j = 1, 2, 3, \dots, n_x,$$

where $h = \frac{1}{n_x}$, we obtain the points (x_i, y_j) , which are the centers of the cells, and the points $(x_{i \pm \frac{1}{2}}, y_j)$ and $(x_i, y_{j \pm \frac{1}{2}})$, which represent the midpoints of the cell edges.

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

$$\begin{aligned} \Omega_{i,j} &= (x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2}), \\ \Omega_{i+1/2,j} &= (x_i, x_{i+1}) \times (y_{j-1/2}, y_{j+1/2}), \\ \Omega_{i,j+1/2} &= (x_{i-1/2}, x_{i+1/2}) \times (y_j, y_{j+1}). \end{aligned}$$

For a function $\theta(x, y)$, let $\theta_{l,m}$ denote the function value $\theta(x_l, y_m)$, where l and m may take the values $i, i + 1/2$ and $j, j + 1/2$, respectively, for integers $i, j \geq 0$. For discrete functions with values defined at proper discrete points, we set

$$\begin{aligned} [d_x \theta]_{i+1/2,j} &= \frac{\theta_{i+1,j} - \theta_{i,j}}{h}, & [D_x \theta]_{i,j} &= \frac{\theta_{i+1/2,j} - \theta_{i-1/2,j}}{h}, \\ [d_y \theta]_{i,j+1/2} &= \frac{\theta_{i,j+1} - \theta_{i,j}}{h}, & [D_y \theta]_{i,j} &= \frac{\theta_{i,j+1/2} - \theta_{i,j-1/2}}{h}. \end{aligned}$$

For the one-dimensional problem we have to restrict the above definitions to the x -direction only.

3.2. The CCFD method. In this section we consider the cell-centered finite difference (CCFD) method for a mean curvature-based image denoising problem. We use a lexicographical ordering of the unknowns,

$$\begin{aligned} U &= [\bar{U}_{11} \quad \bar{U}_{12} \quad \dots \quad \bar{U}_{n_x n_x}]^t, & W &= [\bar{W}_{11} \quad \bar{W}_{12} \quad \dots \quad \bar{W}_{n_x n_x}]^t, \\ V &= [\bar{V}_{11}^x \quad \bar{V}_{12}^x \quad \dots \quad \bar{V}_{n_x-1 n_x-1}^x \quad \bar{V}_{11}^y \quad \bar{V}_{12}^y \quad \dots \quad \bar{V}_{n_x-1 n_x-1}^y]^t, \\ P &= [\bar{P}_{11}^x \quad \bar{P}_{12}^x \quad \dots \quad \bar{P}_{n_x-1 n_x-1}^x \quad \bar{P}_{11}^y \quad \bar{P}_{12}^y \quad \dots \quad \bar{P}_{n_x-1 n_x-1}^y]^t, & \text{and} \\ T &= [\bar{T}_{11}^x \quad \bar{T}_{12}^x \quad \dots \quad \bar{T}_{n_x-1 n_x-1}^x \quad \bar{T}_{11}^y \quad \bar{T}_{12}^y \quad \dots \quad \bar{T}_{n_x-1 n_x-1}^y]^t. \end{aligned}$$

Now by applying the CCFD method to (3.1)–(3.5), one obtains the following equations:

$$(3.10) \quad I_h U - \alpha A_h W + \alpha B_h^* P - \alpha B_h^* T = Z_h,$$

$$(3.11) \quad -I_h W + B_h^* V = 0,$$

$$(3.12) \quad D_h V + B_h U = 0,$$

$$(3.13) \quad D_h P + B_h W = 0,$$

$$(3.14) \quad D_h T - C_h V = 0.$$

Here A_h and I_h are matrices of size $n_x^2 \times n_x^2$, and B_h is a matrix of size $2n_x(n_x - 1) \times n_x^2$. C_h and D_h are matrices of size $2n_x(n_x - 1) \times 2n_x(n_x - 1)$. Thus, we arrive at the following

system

$$\begin{bmatrix} I_h & -\alpha A_h & 0 & \alpha B_h^* & -\alpha B_h^* \\ 0 & -I_h & B_h^* & 0 & 0 \\ B_h & 0 & D_h & 0 & 0 \\ 0 & B_h & 0 & D_h & 0 \\ 0 & 0 & -C_h & 0 & D_h \end{bmatrix} \begin{bmatrix} U \\ W \\ V \\ P \\ T \end{bmatrix} = \begin{bmatrix} Z_h \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The matrix A_h is diagonal having the structure

$$A_h = \frac{2}{\beta h} (A_1 + A_2),$$

where both A_1 and A_2 are of size $n_x^2 \times n_x^2$ defined as

$$A_1 = \tilde{I} \otimes E \quad \text{and} \quad A_2 = E \otimes \tilde{I},$$

where \otimes is the tensor product and \tilde{I} is the identity matrix of size $n_x \times n_x$. The matrix

$$E = \begin{bmatrix} 1 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & 1 \end{bmatrix}$$

is of size $n_x \times n_x$. The matrix B_h has the structure

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

where both B_1 and B_2 are of size $n_x(n_x - 1) \times n_x^2$ with

$$B_1 = F \otimes \tilde{I} \quad \text{and} \quad B_2 = \tilde{I} \otimes F.$$

Here,

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

is a matrix of size $(n_x - 1) \times n_x$. The matrix

$$C_h = \begin{bmatrix} C^x & 0 \\ 0 & C^y \end{bmatrix}$$

is a diagonal matrix, and the diagonal entries are obtained by a discretization of the expression $(\nabla w \cdot \vec{v})$. The matrix C^x is of size $(n_x - 1) \times n_x$, and the matrix C^y is of size $n_x \times (n_x - 1)$. The matrix D_h is also diagonal with positive entries, and the diagonal entries are obtained by discretization of the expression $\sqrt{|\nabla u|^2 + \beta^2}$. The matrix D_h has the structure

$$D_h = \begin{bmatrix} D^x & 0 \\ 0 & D^y \end{bmatrix},$$

where D^x is of size $(n_x - 1) \times n_x$ and D^y is of size $n_x \times (n_x - 1)$. Note that on the horizontal and vertical edges of each cell e_{ij} , the values of all unknowns are not available, so average operators can be used to calculate their values.

Now if we eliminate W, V, P , and T from (3.10)–(3.14), then we have the primal system

$$(3.15) \quad (I_h + \alpha L_h(U))U = Z_h,$$

where

$$L_h = (B_h^* D_h^{-1} B_h)^2 + A_h (B_h^* D_h^{-1} B_h) + B_h^* D_h^{-1} C_h D_h^{-1} B_h.$$

The first and the last term in L_h is symmetric positive semidefinite [10], but the middle term is not symmetric. With lexicographical ordering of the unknowns, L_h is block pentadiagonal. The diagonal blocks are pentadiagonal matrices, while the off-diagonal blocks just below and above the main diagonal blocks are tridiagonal matrices. The remaining blocks are diagonal matrices. A similar block matrix structure exists for the one-dimensional MC-based signal denoising problem.

4. Error analysis. We follow the error analysis procedure of Rui and Pan [7], who considered a Darcy-Forchheimer model using a block-centered finite difference method. For simplicity, we consider the one-dimensional MC-based signal denoising problem (3.6)–(3.9). The cell-centered finite difference approximations $U_i, W_i, V_{i+\frac{1}{2}}$, and $P_{i+\frac{1}{2}}$ of $u(x_i), w(x_i), v(x_{i+\frac{1}{2}})$, and $p(x_{i+\frac{1}{2}})$, respectively, are chosen such that

$$(4.1) \quad U_i + \alpha [D_x P]_i = Z_i,$$

$$(4.2) \quad -W_i + [D_x V]_i = 0,$$

$$(4.3) \quad \sqrt{[d_x U]_{i+\frac{1}{2}}^2 + \beta^2 V_{i+\frac{1}{2}}} - [d_x U]_{i+\frac{1}{2}} = 0,$$

$$(4.4) \quad (\sqrt{[d_x U]_{i+\frac{1}{2}}^2 + \beta^2})^3 P_{i+\frac{1}{2}} - [d_x W]_{i+\frac{1}{2}} = 0,$$

with $V_{\frac{1}{2}} = V_{n+\frac{1}{2}} = 0$ and $W_{\frac{1}{2}} = W_{n+\frac{1}{2}} = 0$. Now we discretize the interval $[0, 1]$ using $n = n_x$ distinct points. With $h = \frac{1}{n}$, the discretization is illustrated in Figure 4.1.

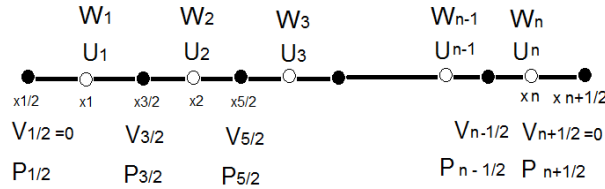


FIG. 4.1. Discretization of the one-dimensional MC-based signal denoising problem.

Error estimates (i). We consider the points $x = x_i, i = 1, 2, \dots, (n - 1)$. From (3.6) we obtain by integration that

$$p_{i+\frac{1}{2}} = p_{i-\frac{1}{2}} + \frac{1}{\alpha} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (z - u) dx.$$

By recursion, we get

$$(4.5) \quad p_{i+\frac{1}{2}} = p_{\frac{1}{2}} + \sum_{l=1}^i \frac{1}{\alpha} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (z - u) dx.$$

From (4.1), we have

$$P_{i+\frac{1}{2}} = P_{i-\frac{1}{2}} + \frac{1}{\alpha} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (Z_i - U_i) dx,$$

which yields again by recursion that

$$(4.6) \quad P_{i+\frac{1}{2}} = P_{\frac{1}{2}} + \sum_{l=1}^i \frac{1}{\alpha} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (Z_l - U_l) dx.$$

Subtracting (4.6) from (4.5), gives

$$p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}} = p_{\frac{1}{2}} - P_{\frac{1}{2}} - \sum_{l=1}^i \frac{1}{\alpha} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (u - U_l) dx + \sum_{l=1}^i \frac{1}{\alpha} \int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (z - Z_l) dx.$$

Applying a mid-point quadrature rule leads to

$$\begin{aligned} p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}} &= p_{\frac{1}{2}} - P_{\frac{1}{2}} - \sum_{l=1}^i \left(\frac{1}{\alpha} h(u_l - U_l) + \frac{1}{\alpha} C_{u_l} h^3 + O(h^4) \right) \\ &\quad + \sum_{l=1}^i \left(\frac{1}{\alpha} h(z_l - Z_l) + \frac{1}{\alpha} C_{z_l} h^3 + O(h^4) \right). \end{aligned}$$

Since z is a known function, we have $z_l = Z_l$. Thus,

$$\begin{aligned} p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}} &= p_{\frac{1}{2}} - P_{\frac{1}{2}} - \sum_{l=1}^i \left(\frac{1}{\alpha} h(u_l - U_l) + \frac{1}{\alpha} C_{u_l} h^3 + O(h^4) \right) \\ &\quad + \sum_{l=1}^i \left(\frac{1}{\alpha} C_{z_l} h^3 + O(h^4) \right) \\ &= C_{w_{\frac{1}{2}}} h^2 + O(h^4) - \sum_{l=1}^i \left(\frac{1}{\alpha} h(u_l - U_l) + \frac{1}{\alpha} C_{u_l} h^3 + O(h^4) \right) \\ &\quad + \sum_{l=1}^i \left(\frac{1}{\alpha} C_{z_l} h^3 + O(h^4) \right), \end{aligned}$$

where we used Appendix A in the last step. So, for $i = 1, 2, \dots, n-1$, we obtain

$$(4.7) \quad \begin{aligned} e_{i+\frac{1}{2}}^p &= C_{w_{\frac{1}{2}}} h^2 + O(h^4) - \sum_{l=1}^i \left(\frac{h}{\alpha} e_l^u + \frac{1}{\alpha} C_{u_l} h^3 + O(h^4) \right) \\ &\quad + \sum_{l=1}^i \left(\frac{1}{\alpha} C_{z_l} h^3 + O(h^4) \right), \end{aligned}$$

where $e_{i+\frac{1}{2}}^p = p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}}$ and $e_l^u = u_l - U_l$. Similarly, from (3.7) and (4.2), we have for $i = 1, 2, \dots, n-1$,

$$(4.8) \quad e_{i+\frac{1}{2}}^v = \sum_{l=1}^i \left(h e_l^w + C_{w_l} h^3 + O(h^4) \right),$$

where $e_{i+\frac{1}{2}}^v = v_{i+\frac{1}{2}} - V_{i+\frac{1}{2}}$ and $e_l^w = w_l - W_l$.

Error estimates (ii). We now consider $x = x_n$. From (3.6) it follows by integration that

$$(4.9) \quad p_{n-\frac{1}{2}} = p_{n+\frac{1}{2}} - \frac{1}{\alpha} \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (z - u) dx.$$

From (4.1), we have

$$(4.10) \quad P_{n-\frac{1}{2}} = P_{n+\frac{1}{2}} - \frac{1}{\alpha} \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (Z_n - U_n).$$

Subtracting (4.10) from (4.9), we find

$$p_{n-\frac{1}{2}} - P_{n-\frac{1}{2}} = p_{n+\frac{1}{2}} - P_{n+\frac{1}{2}} + \frac{1}{\alpha} \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (u - U_n) dx - \frac{1}{\alpha} \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (z - Z_n).$$

As before, using a mid-point quadrature rule, the fact that z is a known function with $z_n = Z_n$, and Appendix A, we have

$$\begin{aligned} p_{n-\frac{1}{2}} - P_{n-\frac{1}{2}} &= p_{n+\frac{1}{2}} - P_{n+\frac{1}{2}} + \frac{1}{\alpha} h(u_n - U_n) + \frac{1}{\alpha} C_{u_n} h^3 + O(h^4) \\ &\quad + \frac{1}{\alpha} h(z_n - Z_n) - \frac{1}{\alpha} C_{z_n} h^3 + O(h^4) \\ &= p_{n+\frac{1}{2}} - P_{n+\frac{1}{2}} + \frac{1}{\alpha} h(u_n - U_n) + \frac{1}{\alpha} C_{u_n} h^3 - \frac{1}{\alpha} C_{z_n} h^3 + O(h^4) \\ &= C_{w_{n+\frac{1}{2}}}''' h^2 + O(h^4) + \frac{1}{\alpha} h(u_n - U_n) + \frac{1}{\alpha} C_{u_n} h^3 - \frac{1}{\alpha} C_{z_n} h^3 + O(h^4). \end{aligned}$$

Hence, we arrive at

$$e_{n-\frac{1}{2}}^p = C_{w_{n+\frac{1}{2}}}''' h^2 + \frac{h}{\alpha} e_n^u + \frac{1}{\alpha} C_{u_n} h^3 - \frac{1}{\alpha} C_{z_n} h^3 + O(h^4),$$

where $e_{n-\frac{1}{2}}^p = p_{n-\frac{1}{2}} - P_{n-\frac{1}{2}}$ and $e_n^u = u_n - U_n$. By using (4.7), the above equation becomes

$$\begin{aligned} &C_{w_{\frac{1}{2}}}''' h^2 + O(h^4) - \sum_{l=1}^{n-1} \left(\frac{h}{\alpha} e_l^u + \frac{1}{\alpha} C_{u_l} h^3 + O(h^4) \right) + \sum_{l=1}^{n-1} \left(\frac{1}{\alpha} C_{z_l} h^3 + O(h^4) \right) \\ &= C_{w_{n+\frac{1}{2}}}''' h^2 + \frac{h}{\alpha} e_n^u + \frac{1}{\alpha} C_{u_n} h^3 - \frac{1}{\alpha} C_{z_n} h^3 + O(h^4). \end{aligned}$$

Taking a common $\frac{h}{\alpha}$ -factor from both sides, we obtain

$$\begin{aligned} &\alpha C_{w_{\frac{1}{2}}}''' h + O(h^3) - \sum_{l=1}^{n-1} \left(e_l^u + C_{u_l} h^2 + O(h^3) \right) + \sum_{l=1}^{n-1} \left(C_{z_l} h^2 + O(h^3) \right) \\ &= \alpha C_{w_{n+\frac{1}{2}}}''' h + e_n^u + C_{u_n} h^2 - C_{z_n} h^2 + O(h^3). \end{aligned}$$

After rearranging, we arrive at

$$(4.11) \quad \begin{aligned} e_1^u &= \alpha C_{w_{\frac{1}{2}}}''' h - \alpha C_{w_{n+\frac{1}{2}}}''' h + O(h^3) - \sum_{l=2}^n e_l^u \\ &\quad - \sum_{l=1}^n \left(C_{u_l} h^2 + O(h^3) \right) + \sum_{l=1}^n \left(C_{z_l} h^2 + O(h^3) \right). \end{aligned}$$

Similarly, from (3.7) and (4.2), we have

$$(4.12) \quad e_1^w = - \sum_{l=2}^n e_l^w - \sum_{l=1}^n \left(C_{w_l} h^2 + O(h^3) \right),$$

where $e_n^w = w_n - W_n$.

Error estimates (iii). We consider $x = x_{i-\frac{1}{2}}, i = 2, 3, \dots, n$. From (3.8), by integration, we have

$$u_i = u_{i-1} - \int_{x_{i-1}}^{x_i} \sqrt{u_x^2 + \beta^2 v} dx,$$

which by recursion leads to

$$(4.13) \quad u_i = u_1 + \sum_{l=2}^i \int_{x_{l-1}}^{x_l} \sqrt{u_x^2 + \beta^2 v} dx.$$

From (4.3), we have

$$U_i = U_{i-1} + \int_{x_{i-1}}^{x_i} \sqrt{[d_x U]_{i-\frac{1}{2}}^2 + \beta^2 V_{i-\frac{1}{2}}} dx.$$

By recursion, we obtain

$$(4.14) \quad U_i = U_1 + \sum_{l=2}^i \int_{x_{l-1}}^{x_l} \sqrt{[d_x U]_{l-\frac{1}{2}}^2 + \beta^2 V_{l-\frac{1}{2}}} dx.$$

Subtracting (4.14) from (4.13) yields

$$u_i - U_i = [u_1 - U_1] + \sum_{l=2}^i \int_{x_{l-1}}^{x_l} \sqrt{u_x^2 + \beta^2 v} dx - \sum_{l=2}^i \int_{x_{l-1}}^{x_l} \sqrt{[d_x U]_{l-\frac{1}{2}}^2 + \beta^2 V_{l-\frac{1}{2}}} dx.$$

Adding and subtracting the term $\sum_{l=2}^i \int_{x_{l-1}}^{x_l} \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}} dx$ on the right-hand side of the above equation gives

$$(4.15) \quad \begin{aligned} u_i - U_i &= [u_1 - U_1] \\ &+ \sum_{l=2}^i \int_{x_{l-1}}^{x_l} [\sqrt{u_x^2 + \beta^2 v} - \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}}] dx \\ &+ \sum_{l=2}^i \int_{x_{l-1}}^{x_l} [\sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}} - \sqrt{[d_x U]_{l-\frac{1}{2}}^2 + \beta^2 V_{l-\frac{1}{2}}}] dx, \end{aligned}$$

from which it follows that

$$e_i^u = e_1^u + I_1 + I_2,$$

with I_1 and I_2 defined as the last two terms in (4.15). So, by Appendix B, for $i = 2, 3, \dots, n$, we have

$$(4.16) \quad \begin{aligned} e_i^u &= e_1^u + \sum_{l=2}^i \left(C_{u_{l-\frac{1}{2}}}''' h^3 + O(h^5) \right) + \sum_{l=2}^i h \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}} \\ &- \sum_{l=2}^i h \sqrt{(u_x(x_l) - \frac{h}{2} u_{xx}(x_l) + O(h^2))^2 + \beta^2 \left(v(x_l) - \frac{h}{2} v_x(x_l) + O(h^2) \right)}, \end{aligned}$$

where $e_i^u = u_i - U_i$. Similarly, from (3.9) and (4.4), we have for $i = 2, 3, \dots, n$,

(4.17)

$$e_i^w = e_1^w + \sum_{l=2}^i \left(C_{w_{l-\frac{1}{2}}}''' h^3 + O(h^5) \right) + \sum_{l=2}^i h \left(\sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2} \right)^3 p_{l-\frac{1}{2}} \\ - \sum_{l=2}^i h \left(\sqrt{(u_x(x_l) - \frac{h}{2} u_{xx}(x_l) + O(h^2))^2 + \beta^2} \right)^3 \left(p(x_l) - \frac{h}{2} p_x(x_l) + O(h^2) \right),$$

where $e_i^w = w_i - W_i$. Now with the help of the above analysis, we can present the following theorem:

THEOREM 4.1. *Let u, w, v , and p be the solutions of the system (3.6)–(3.9) on the mesh Ω_h , and let U, W, V , and P be the computed solutions of the system (4.1)–(4.4) on the mesh Ω_h . Then there exists a positive constant C independent of h such that, for $i = 1, 2, \dots, n$,*

$$|u_i - U_i| \leq Ch, \quad |w_i - W_i| \leq Ch,$$

and, for $i = 2, 3, \dots, n-1$,

$$|v_{i+\frac{1}{2}} - V_{i+\frac{1}{2}}| \leq Ch^2, \quad |p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}}| \leq Ch^2.$$

Proof. For $i = 1, 2, 3, \dots, n$, define

$$e^u = u_i - U_i \quad \text{and} \quad e^w = w_i - W_i,$$

and, for $i = 1, 2, 3, \dots, n-1$, define

$$e^v = v_{i+\frac{1}{2}} - V_{i+\frac{1}{2}} \quad \text{and} \quad e^p = p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}}.$$

Now let

$$E^u = [e_1^u \ e_2^u \ \dots \ e_n^u]^t, \quad E^w = [e_1^w \ e_2^w \ \dots \ e_n^w]^t, \\ E^v = [e_{\frac{3}{2}}^v \ e_{\frac{5}{2}}^v \ \dots \ e_{n-\frac{1}{2}}^v]^t, \quad E^p = [e_{\frac{3}{2}}^p \ e_{\frac{5}{2}}^p \ \dots \ e_{n-\frac{1}{2}}^p]^t.$$

Then, from (4.7), (4.8), (4.11), (4.12), (4.16), and (4.17), we obtain the following system:

$$E^u = A_{n \times n} E^u + \tilde{T}_{n \times 1}^u, \quad E^w = h B_{(n-1) \times n} E^w + \tilde{T}_{n \times 1}^w, \\ E^v = A_{n \times n} E^v + \tilde{T}_{(n-1) \times 1}^v, \quad E^p = -\frac{h}{\alpha} B_{(n-1) \times n} E^u + \tilde{T}_{(n-1) \times 1}^p,$$

where

$$A_{n \times n} = \begin{bmatrix} 0 & -1 & -1 & & \dots & -1 \\ 1 & 0 & 0 & & \dots & 0 \\ 1 & 0 & 0 & & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ 1 & 0 & 0 & \dots & \dots & 0 \end{bmatrix}, \\ B_{(n-1) \times n} = \begin{bmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ 1 & 1 & 0 & \dots & \dots & 0 \\ & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & & \\ 1 & 1 & 1 & \dots & 1 & 0 \end{bmatrix},$$

and the structure of the truncation vectors is as follows:

$$\tilde{T}_{n \times 1}^u = \left[\begin{array}{c} \left\{ \begin{array}{l} \alpha C_{w_{\frac{1}{2}}}''' h - \alpha C_{w_{n+\frac{1}{2}}}''' h + O(h^3) \\ - \sum_{l=1}^n (C_{u_l} h^2 + O(h^3)) + \sum_{l=1}^n (C_{z_l} h^2 + O(h^3)) \end{array} \right. \\ \left\{ \begin{array}{l} C_{u_{\frac{3}{2}}}''' h^3 + O(h^5) + h \sqrt{(u_x(x_{\frac{3}{2}}))^2 + \beta^2} v(x_{\frac{3}{2}}) \\ - h \sqrt{(u_x(x_2) + O(h))^2 + \beta^2} (v(x_2) + O(h)) \end{array} \right. \\ \left\{ \begin{array}{l} \sum_{l=2}^3 (C_{u_{l-\frac{1}{2}}}''' h^3 + O(h^5)) + \sum_{l=2}^3 h \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2} v(x_{l-\frac{1}{2}}) \\ - \sum_{l=2}^3 h \sqrt{(u_x(x_l) + O(h))^2 + \beta^2} (v(x_l) + O(h)) \end{array} \right. \\ \vdots \\ \left\{ \begin{array}{l} \sum_{l=2}^n (C_{u_{l-\frac{1}{2}}}''' h^3 + O(h^5)) + \sum_{l=2}^n h \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2} v(x_{l-\frac{1}{2}}) \\ - \sum_{l=2}^n h \sqrt{(u_x(x_l) + O(h))^2 + \beta^2} (v(x_l) + O(h)) \end{array} \right. \end{array} \right],$$

$$\tilde{T}_{n \times 1}^w = \left[\begin{array}{c} - \sum_{l=1}^n (C_{w_l} h^2 + O(h^3)) \\ \left\{ \begin{array}{l} C_{w_{\frac{3}{2}}}''' h^3 + O(h^5) + h \left(\sqrt{(u_x(x_{\frac{3}{2}}))^2 + \beta^2} \right)^3 p(x_{\frac{3}{2}}) \\ - h \left(\sqrt{(u_x(x_2) + O(h))^2 + \beta^2} \right)^3 (p(x_2) + O(h)) \end{array} \right. \\ \left\{ \begin{array}{l} \sum_{l=2}^3 (C_{w_{l-\frac{1}{2}}}''' h^3 + O(h^5)) + \sum_{l=2}^3 h \left(\sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2} \right)^3 p(x_{l-\frac{1}{2}}) \\ - \sum_{l=2}^3 h \left(\sqrt{(u_x(x_l) + O(h))^2 + \beta^2} \right)^3 (p(x_l) + O(h)) \end{array} \right. \\ \vdots \\ \left\{ \begin{array}{l} \sum_{l=2}^n (C_{w_{l-\frac{1}{2}}}''' h^3 + O(h^5)) + \sum_{l=2}^n h \left(\sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2} \right)^3 p(x_{l-\frac{1}{2}}) \\ - \sum_{l=2}^n h \left(\sqrt{(u_x(x_l) + O(h))^2 + \beta^2} \right)^3 (p(x_l) + O(h)) \end{array} \right. \end{array} \right],$$

$$\tilde{T}_{(n-1) \times 1}^v = \left[\begin{array}{c} C_{w_1} h^3 + O(h^4) \\ C_{w_1} h^3 + C_{w_2} h^3 + O(h^4) \\ C_{w_1} h^3 + C_{w_2} h^3 + C_{w_3} h^3 + O(h^4) \\ \vdots \\ \vdots \\ \sum_{l=1}^{n-1} (C_{w_l} h^3 + O(h^4)) \end{array} \right],$$

and

$$\tilde{T}_{(n-1) \times 1}^p = \begin{bmatrix} C_{w_{\frac{1}{2}}}''' h^2 - \frac{1}{\alpha} C_{u_1} h^3 + \frac{1}{\alpha} C_{z_1} h^3 + O(h^4) \\ \left\{ \begin{array}{l} C_{w_{\frac{1}{2}}}''' h^2 - \frac{1}{\alpha} C_{u_1} h^3 - \frac{1}{\alpha} C_{u_2} h^3 \\ + \frac{1}{\alpha} C_{z_1} h^3 + \frac{1}{\alpha} C_{z_2} h^3 + O(h^4) \end{array} \right. \\ \left\{ \begin{array}{l} C_{w_{\frac{1}{2}}}''' h^2 - \frac{1}{\alpha} C_{u_1} h^3 - \frac{1}{\alpha} C_{u_2} h^3 - \frac{1}{\alpha} C_{u_3} h^3 \\ + \frac{1}{\alpha} C_{z_1} h^3 + \frac{1}{\alpha} C_{z_2} h^3 + \frac{1}{\alpha} C_{z_3} h^3 + O(h^4) \end{array} \right. \\ \vdots \\ \left\{ \begin{array}{l} C_{w_{\frac{1}{2}}}''' h^2 + O(h^4) - \sum_{l=1}^{n-1} \left(\frac{1}{\alpha} C_{u_l} h^3 + O(h^4) \right) \\ + \sum_{l=1}^{n-1} \left(\frac{1}{\alpha} C_{z_l} h^3 + O(h^4) \right) \end{array} \right. \end{bmatrix}.$$

So, for simplicity dropping the subscript, by solving the above system, we obtain

$$\begin{aligned} E^u &= (I - A)^{-1} \tilde{T}^u, & E^w &= (I - A)^{-1} \tilde{T}^w, \\ E^v &= hB(I - A)^{-1} \tilde{T}^w + \tilde{T}^v, & E^p &= -\frac{h}{\alpha} B(I - A)^{-1} \tilde{T}^u + \tilde{T}^p. \end{aligned}$$

By simple column operations, one can see that $\det((I - A)_{n \times n}) = n$, hence $(I - A)_{n \times n}$ is a nonsingular matrix. Now by expansion into a Taylor series, the errors E^u, E^w, E^v , and E^p have the form

$$\begin{aligned} E^u &= (I - A)^{-1} d\tilde{T}^u h + \dots, \\ E^w &= (I - A)^{-1} d\tilde{T}^w h + \dots, \\ E^v &= 2B(I - A)^{-1} d\tilde{T}^w h^2 + \dots, \\ E^p &= -\frac{2}{\alpha} \left(B(I - A)^{-1} d\tilde{T}^u + dd\tilde{T}^p \right) h^2 + \dots \end{aligned}$$

This completes the proof. \square

(For the derivatives of the truncation vectors and error vectors, see Appendix C and Appendix D, respectively).

REMARK 4.2. Along the same lines, we can carry out a similar error analysis for the two-dimensional MC-based image denoising problem.

5. Numerical implementation. Here, we introduce an algorithm to solve the primal form (3.15) of the MC-based nonlinear image denoising problem. Then we present numerical examples based on that algorithm.

At first, we apply a discrete version of the fixed point iteration to (3.15), which leads to the linear system

$$(5.1) \quad (I_h + \alpha L_h(U^m))U^{m+1} = Z_h.$$

Indeed, (5.1) is a fixed point method but its convergence is not yet proved. However, it is known that for the MC model, many well-known and simple fixed point methods are not converging numerically [11, 13]. The properties of our system (3.15), mentioned in Section 3, suggest that an iterative method like a Generalized Minimal Residual (GMRES) method may be suitable for (5.1). Unfortunately, the GMRES method can exhibit quite slow convergence

due to the system being ill-conditioned. One remedy for this problem is preconditioning, which is the reason why we employ a Preconditioned Generalized Minimal Residual (PGMRES) method. In order to make the PGMRES method effective, the preconditioning matrix P must be symmetric positive definite. Therefore we use the following simple preconditioning matrix \bar{P} ,

$$\bar{P} = I_h + \alpha \text{diag}(L_h),$$

where I_h is the identity matrix and $\text{diag}(L_h)$ is the diagonal matrix, whose entries are the diagonal entries of the matrix L_h . When applying the \bar{P} -PGMRES method to (5.1), the inversion of \bar{P} is required. Since our preconditioning matrix \bar{P} is a diagonal matrix, the inversion can be done easily. Rapid convergence is observed in the numerical results below for the \bar{P} -PGMRES method due to our preconditioning matrix \bar{P} . The same algorithm is applied to a one-dimensional MC-based signal denoising problem. The above discussion is summarized in Algorithm 1.

Algorithm 1: The \bar{P} -PGMRES method.

Given: mesh: Ω_h ; initial iteration: U^0 .

for $m = 1$; max

$$A^m = I_h + \alpha L_h(U^m), \quad b^m = Z_h,$$

Use PGMRES method to solve

$$A^m U^{m+1} = b^m, \quad \text{with} \quad \bar{P}_h = I_h + \alpha \text{diag}(L_h(U^m)),$$

end

We present two numerical examples for the MC-based denoising problem. Example 5.1 concerns a one-dimensional model and Example 5.3 a two-dimensional model. In both examples, the value of α is used according to [2, 12, 13], and for the stopping criteria of a numerical method we set a tolerance $tol = 1e-3$. For numerical computations, we used MATLAB, and the results are obtained with an Intel Core i7-4510U CPU @2.00 GHz 2.60 GHz computer. To measure the quality of the restored signals or images, we calculate the signal to noise ratio (SNR) [2].

EXAMPLE 5.1. In this example we use the \bar{P} -PGMRES method for a one-dimensional MC-based signal denoising problem. Here we have used as exact solution a *sine-curve*. The data and results, namely (a) the original (exact) curve, (b) the noisy data, and (c) the curve denoised by the \bar{P} -PGMRES method, are displayed in Figure 5.1. Here, we solved the problem with Algorithm 1 with $\beta = 1$ and α varying from $1e-7$ to $1e-3$ [2]. We tested with four values of n_x , which are $n_x = 64, 128, 256, \text{ and } 512$. All the noisy data have an SNR close to 13. In Table 5.1, we summarize all the information about this experiment.

REMARK 5.2.

1. Table 5.1 shows that the \bar{P} -PGMRES method has first-order accuracy in the discrete L_2 -norm. These results are consistent with the error estimates in Theorem 4.1.
2. From Figure 5.1, one can notice the quality of the denoised signal produced by the \bar{P} -PGMRES method. The graph in Figure 5.1(c) is almost similar to the original one in Figure 5.1(a) with all the noise removed. This means that the \bar{P} -PGMRES method generates a high-quality denoised function by efficiently removing the noise.

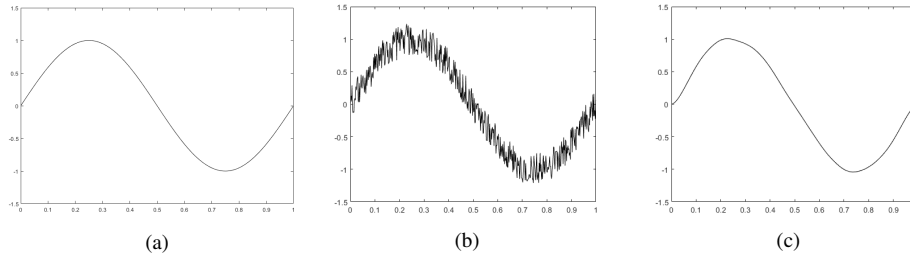


FIG. 5.1. *sine-curve: (a) original signal, (b) noisy signal, and (c) denoised signal by the \bar{P} -PGMRES method.*

TABLE 5.1
Errors and convergence rates for Example 5.1

n_x	Mesh Size (h)	Error	Rate
64	1/64	6.8506e-03	–
128	1/128	3.6680e-03	-0.9011
256	1/256	1.7550e-03	-1.0635
512	1/512	7.5058e-04	-1.2254

EXAMPLE 5.3. In this example we use the \bar{P} -PGMRES method for a two-dimensional MC-based image denoising problem. The example includes the “brain” image and the “peppers” image, which are shown in Figure 5.2. The size of each figure is 512×512 , and (a) and (c) represent noisy images, while (b) and (d) are denoised images using the \bar{P} -PGMRES method. For both noisy images we have an SNR of 3.5. In this experiment, we use $\alpha = 1e-3$ and β is varied from $1e-1$ to 1 [12, 13]. For the analysis of the convergence rate we take four values of n_x , 64, 128, 256, and 512. In Table 5.2 we summarize the information about the errors and convergence rates of this experiment.

REMARK 5.4.

1. From Figure 5.2, one can notice the quality of the denoised images produce by the \bar{P} -PGMRES method. Most of the noise is removed, which means that the \bar{P} -PGMRES method generates high-quality results.
2. Table 5.2 shows that the \bar{P} -PGMRES method has first-order accuracy in the discrete L_2 -norm for all values of n_x .

TABLE 5.2
Errors and convergence rates for Example 5.3.

n_x	Mesh Size (h)	Brain image		Peppers image	
		Error	Rate	Error	Rate
64	1/64	5.8465e-04	–	4.4559e-04	–
128	1/128	1.6012e-04	-1.8303	1.9648e-04	-1.1813
256	1/252	4.6449e-05	-1.6192	8.7649e-05	-1.1646
512	1/512	1.6120e-05	-1.3157	3.6438e-05	-1.2663

EXAMPLE 5.5. In this example we present a comparison of our method (\bar{P} -PGMRES method) with the multigrid method of C. Brito-Loeza and K. Chen [2]. For this we use two images (brain and peppers), displayed in Figure 5.3. The size of each figure is 512×512 and (a) and (d) represent noisy images, while (b) and (e) are the denoised images using the

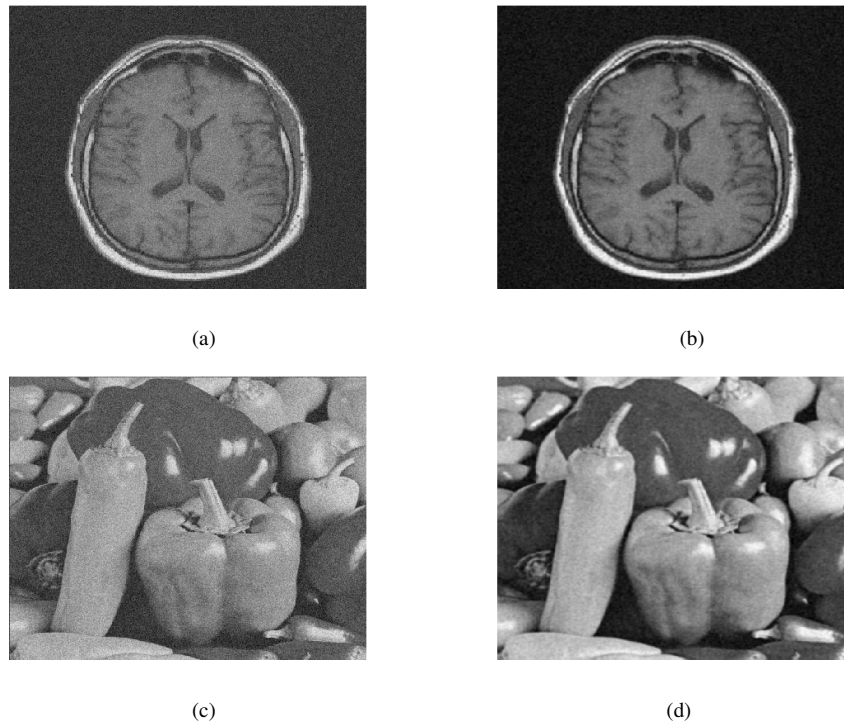


FIG. 5.2. Brain image: (a) noisy image and (b) denoised image by the \bar{P} -PGMRES method. Peppers image: (c) noisy image and (d) denoised image by the \bar{P} -PGMRES method.

multigrid method, and (c) and (f) are the denoised images using the \bar{P} -PGMRES method.

In the multigrid method we use the parameters [2] $\alpha = \frac{1}{200}$ and $\beta = 1e-2$. For the \bar{P} -PGMRES method, we use $\alpha = 1e-3$, and β is varied from $1e-1$ to 1 [12, 13]. In this experiment, all the noisy images have an SNR of 3.5. Here, we take two values of n_x for each image, namely, 256 and 512. In Table 5.3, we summarize the information about this experiment.

REMARK 5.6.

1. Table 5.3 shows that the CPU-time taken for the \bar{P} -PGMRES method is less than that for the multigrid method for all values of n_x . For $n_x = 256$, our method saves more than 25% CPU-time for each image, and for $n_x = 512$, our method saves more than 40%. We expect even more saving in time as the mesh size is further refined.
2. From Figure 5.3, one can notice that the denoised images are almost similar, and most of the noise has been removed. This means that both methods generate the same quality, but the \bar{P} -PGMRES method does this in less CPU-time. This means that our method is a more robust method.

6. Conclusion. A cell-centered finite difference method for a mean curvature-based image denoising problem is discussed. First-order error estimates are established on a uniform rectangular mesh. A numerical algorithm is also presented to solve the primal form of the mean curvature-based nonlinear image denoising problem. Three examples are tested using our algorithm (\bar{P} -PGMRES method). In Example 5.1, we applied the \bar{P} -PGMRES method to a one-dimensional signal denoising problem and in Example 5.3 to a two-dimensional image denoising problem. Numerical experiments using this scheme show consistency of the

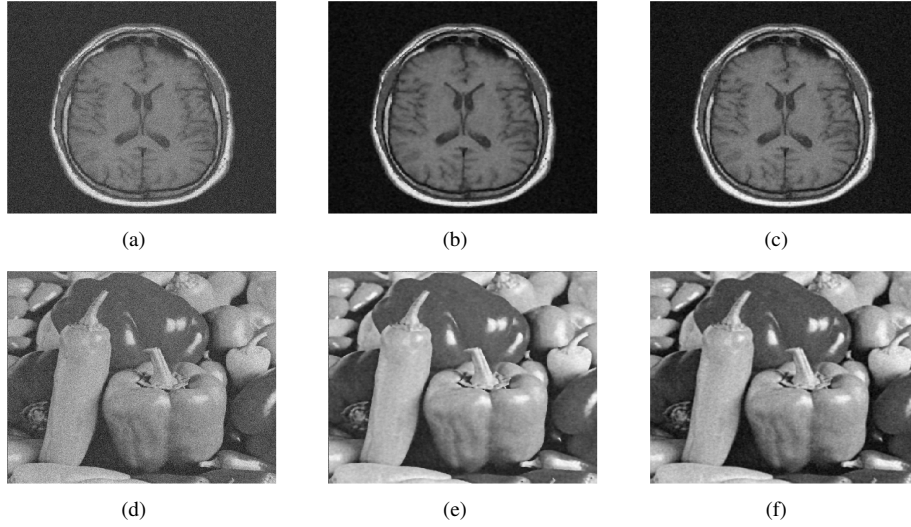


FIG. 5.3. Brain image: (a) noisy image, (b) denoised image by the multigrid method, and (c) denoised image by the \bar{P} -PGMRES method. Peppers image: (d) noisy image, (e) denoised image by the multigrid method, and (f) denoised image by the \bar{P} -PGMRES method.

TABLE 5.3
 CPU-time comparison of the multigrid method and the \bar{P} -PGMRES method for Example 5.5.

Image	n_x	Mesh Size (h)	Multigrid	\bar{P} -PGMRES
			CPU-time	CPU-time
Brain	256	1/252	33.0477	14.6527
	512	1/512	139.9676	81.1921
Peppers	256	1/252	35.3301	13.3242
	512	1/512	140.0890	74.9122

convergence rates of our method with the theoretical analysis. In Example 5.5, we compared the \bar{P} -PGMRES method with a multigrid method, and the numerical results indicate that our method is a more robust method.

Acknowledgments. The first and the last author would like to acknowledge the support provided by the Deanship of Scientific Research at KFUPM for funding this work through project no. SB181013.

Appendix A. In this section we present a collection of supplementary materials. Consider,

$$p(x_{\frac{1}{2}}) - P_{\frac{1}{2}} = \frac{1}{(\sqrt{u_x(x_{\frac{1}{2}})^2 + \beta^2})^3} w_x(x_{\frac{1}{2}}) - \frac{1}{(\sqrt{\frac{1}{h}[U_1 - U_0]^2 + \beta^2})^3} \frac{1}{h} [W_1 - W_0].$$

Since $u_x(x_{\frac{1}{2}}) = 0$ and $U_0 = U_1$, we have

$$\begin{aligned} p(x_{\frac{1}{2}}) - P_{\frac{1}{2}} &= \frac{1}{\beta^3} \left\{ w_x(x_{\frac{1}{2}}) - \frac{1}{h} [W_1 - W_0] \right\} \\ &= \frac{1}{\beta^3} \left\{ w_x(x_{\frac{1}{2}}) - \frac{1}{h} [w(x_{\frac{1}{2}} + \frac{h}{2}) - w(x_{\frac{1}{2}} - \frac{h}{2})] \right\}. \end{aligned}$$

Applying a Taylor expansion on the right-hand side, we get

$$p(x_{\frac{1}{2}}) - P_{\frac{1}{2}} = -\frac{h^2}{24\beta^3} w_{xxx}(x_{\frac{1}{2}}) + O(h^4).$$

Hence, $p(x_{\frac{1}{2}}) - P_{\frac{1}{2}} = C_{w_{\frac{1}{2}}}''' h^2 + O(h^4)$. Similarly, $p(x_{n+\frac{1}{2}}) - P_{n+\frac{1}{2}} = C_{w_{n+\frac{1}{2}}}''' h^2 + O(h^4)$.

Appendix B. Consider

$$I_1 = \sum_{l=2}^i \int_{x_{l-1}}^{x_l} [\sqrt{u_x^2 + \beta^2 v} - \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}}] dx, \quad \text{where } i = 2, 3, \dots, n.$$

Using (3.8), we have $I_1 = \sum_{l=2}^i \int_{x_{l-1}}^{x_l} [u_x(x) - u_x(x_{l-\frac{1}{2}})] dx$. By a Taylor expansion of $u_x(x)$ and then integrating, we have

$$\begin{aligned} I_1 &= \sum_{l=2}^i \left[\frac{(x - x_{l-\frac{1}{2}})^2}{2} u_{xx}(x_{l-\frac{1}{2}}) + \frac{(x - x_{l-\frac{1}{2}})^3}{6} u_{xxx}(x_{l-\frac{1}{2}}) + O((x - x_{l-\frac{1}{2}})^4) \right]_{x_{l-1}}^{x_l} \\ &= \sum_{l=2}^i \left[\frac{h^3}{24} u_{xxx}(x_{l-\frac{1}{2}}) + O(h^5) \right] = \sum_{l=2}^i C_{u_{l-\frac{1}{2}}}''' h^3 + O(h^5). \end{aligned}$$

Now consider

$$\begin{aligned} I_2 &= \sum_{l=2}^i \int_{x_{l-1}}^{x_l} [\sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}} - \sqrt{[d_x U]_{l-\frac{1}{2}}^2 + \beta^2 V_{l-\frac{1}{2}}}] dx \\ &= \sum_{l=2}^i h \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}} - \sum_{l=2}^i h \sqrt{[d_x U]_{l-\frac{1}{2}}^2 + \beta^2 V_{l-\frac{1}{2}}}. \end{aligned}$$

Expanding into a Taylor series, we have

$$V_{l-\frac{1}{2}} = v(x_l) - \frac{h}{2} v_x(x_l) + O(h^2) \quad \text{and} \quad [d_x U]_{l-\frac{1}{2}} = u_x(x_l) - \frac{h}{2} u_{xx}(x_l) + O(h^2),$$

hence,

$$\begin{aligned} I_2 &= \sum_{l=2}^i h \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v_{l-\frac{1}{2}}} \\ &\quad - \sum_{l=2}^i h \sqrt{(u_x(x_l) - \frac{h}{2} u_{xx}(x_l) + O(h^2))^2 + \beta^2 (v(x_l) - \frac{h}{2} v_x(x_l) + O(h^2))}. \end{aligned}$$

Appendix C. Here, we find the first and the second derivatives of the truncation vectors $\tilde{T}_{n \times 1}^u$, $\tilde{T}_{n \times 1}^w$, $\tilde{T}_{n \times 1}^v$, and $\tilde{T}_{n \times 1}^p$ at $h = 0$. Since each element of the truncation vectors involve h , it follows that

$$\tilde{T}_{n \times 1}^u|_{h=0} = 0_{n \times 1}, \quad \tilde{T}_{n \times 1}^w|_{h=0} = 0_{n \times 1}, \quad \tilde{T}_{n \times 1}^v|_{h=0} = 0_{n \times 1}, \quad \text{and} \quad \tilde{T}_{n \times 1}^p|_{h=0} = 0_{n \times 1}.$$

Since $\tilde{T}_{n \times 1}^u$, $\tilde{T}_{n \times 1}^w$, $\tilde{T}_{n \times 1}^v$, and $\tilde{T}_{n \times 1}^p$ all are column vectors, differentiating with respect to h (row-wise) and setting $h = 0$ results in the following first derivatives:

$$\begin{aligned} \frac{\partial \tilde{T}_{n \times 1}^u}{\partial h} \Big|_{h=0} &= \begin{bmatrix} -\frac{\alpha}{24\beta^3} w_{xxx}(x_{\frac{1}{2}}) + \frac{\alpha}{24\beta^3} w_{xxx}(x_{n+\frac{1}{2}}) \\ \sqrt{(u_x(x_{\frac{3}{2}}))^2 + \beta^2 v(x_{\frac{3}{2}})} - \sqrt{u_x(x_2) + \beta^2 v(x_2)} \\ \sum_{l=2}^3 \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v(x_{l-\frac{1}{2}})} - \sum_{l=2}^3 \sqrt{u_x(x_l) + \beta^2 v(x_l)} \\ \vdots \\ \vdots \\ \sum_{l=2}^n \sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2 v(x_{l-\frac{1}{2}})} - \sum_{l=2}^n \sqrt{u_x(x_l) + \beta^2 v(x_l)} \end{bmatrix} \\ &= d\tilde{T}_{n \times 1}^u, \end{aligned}$$

$$\begin{aligned} \frac{\partial \tilde{T}_{n \times 1}^w}{\partial h} \Big|_{h=0} &= \begin{bmatrix} 0 \\ \left(\sqrt{(u_x(x_{\frac{3}{2}}))^2 + \beta^2}\right)^3 p(x_{\frac{3}{2}}) - \left(\sqrt{u_x(x_2) + \beta^2}\right)^3 p(x_2) \\ \sum_{l=2}^3 \left(\sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2}\right)^3 p(x_{l-\frac{1}{2}}) - \sum_{l=2}^3 \left(\sqrt{u_x(x_l) + \beta^2}\right)^3 p(x_l) \\ \vdots \\ \vdots \\ \sum_{l=2}^n \left(\sqrt{(u_x(x_{l-\frac{1}{2}}))^2 + \beta^2}\right)^3 p(x_{l-\frac{1}{2}}) - \sum_{l=2}^n \left(\sqrt{u_x(x_l) + \beta^2}\right)^3 p(x_l) \end{bmatrix} \\ &= d\tilde{T}_{n \times 1}^w, \end{aligned}$$

$$\frac{\partial \tilde{T}_{(n-1) \times 1}^v}{\partial h} \Big|_{h=0} = 0_{n \times 1}, \quad \text{and} \quad \frac{\partial \tilde{T}_{(n-1) \times 1}^p}{\partial h} \Big|_{h=0} = 0_{n \times 1}.$$

Repeating the procedure yields the following second derivatives:

$$\begin{aligned} \frac{\partial^2 \tilde{T}_{n \times 1}^u}{\partial h^2} \Big|_{h=0} &= \begin{bmatrix} -\sum_{l=1}^n \frac{1}{12} u_{xx}(x_l) + \sum_{l=1}^n \frac{1}{12} z''(x_l) \\ -2 \left(\frac{u_x(x_2) u_{xx}(x_2) v(x_2)}{\sqrt{u_x^2(x_2) + \beta^2}} + \sqrt{u_x^2(x_2) + \beta^2} v_x(x_2) \right) \\ -2 \sum_{l=2}^3 \left(\frac{u_x(x_l) u_{xx}(x_l) v(x_l)}{\sqrt{u_x^2(x_l) + \beta^2}} + \sqrt{u_x^2(x_l) + \beta^2} v_x(x_l) \right) \\ \vdots \\ \vdots \\ -2 \sum_{l=2}^n \left(\frac{u_x(x_l) u_{xx}(x_l) v(x_l)}{\sqrt{u_x^2(x_l) + \beta^2}} + \sqrt{u_x^2(x_l) + \beta^2} v_x(x_l) \right) \end{bmatrix} = dd\tilde{T}_{n \times 1}^u, \end{aligned}$$

$$\begin{aligned}
 \frac{\partial^2 \tilde{T}_{n \times 1}^w}{\partial h^2} \Big|_{h=0} &= \begin{bmatrix} -\sum_{l=1}^n \frac{1}{12} w_{xxx}(x_l) \\ -2 \left(3u_x(x_2)u_{xx}(x_2)p(x_2)\sqrt{u_x^2(x_2) + \beta^2} + \left(\sqrt{u_x^2(x_2) + \beta^2}\right)^3 p_x(x_2) \right) \\ -2 \sum_{l=2}^3 \left(3u_x(x_l)u_{xx}(x_l)p(x_l)\sqrt{u_x^2(x_l) + \beta^2} + \left(\sqrt{u_x^2(x_l) + \beta^2}\right)^3 p_x(x_l) \right) \\ \vdots \\ -2 \sum_{l=2}^n \left(3u_x(x_l)u_{xx}(x_l)p(x_l)\sqrt{u_x^2(x_l) + \beta^2} + \left(\sqrt{u_x^2(x_l) + \beta^2}\right)^3 p_x(x_l) \right) \end{bmatrix} \\
 &= dd\tilde{T}_{n \times 1}^w,
 \end{aligned}$$

$$\frac{\partial^2 \tilde{T}_{(n-1) \times 1}^v}{\partial h^2} \Big|_{h=0} = 0_{n \times 1},$$

$$\frac{\partial^2 \tilde{T}_{(n-1) \times 1}^p}{\partial h^2} \Big|_{h=0} = \begin{bmatrix} -\frac{1}{12\beta^3} w_{xxx}(x_{\frac{1}{2}}) \\ -\frac{1}{12\beta^3} w_{xxx}(x_{\frac{1}{2}}) \\ -\frac{1}{12\beta^3} w_{xxx}(x_{\frac{1}{2}}) \\ \vdots \\ -\frac{1}{12\beta^3} w_{xxx}(x_{\frac{1}{2}}) \end{bmatrix} = dd\tilde{T}^p.$$

Appendix D. Here, we find the first and second derivatives of the error vectors E^u , E^w , E^v , and E^p at $h = 0$. For simplicity, we ignore the subscripts. Consider

$$E^u = (I - A)^{-1} \tilde{T}^u$$

so that

$$\frac{\partial E^u}{\partial h} = (I - A)^{-1} \frac{\partial \tilde{T}^u}{\partial h}.$$

From Appendix C, we have

$$\frac{\partial E^u}{\partial h} \Big|_{h=0} = (I - A)^{-1} d\tilde{T}^u,$$

and similarly,

$$\frac{\partial E^w}{\partial h} \Big|_{h=0} = (I - A)^{-1} d\tilde{T}^w.$$

Now consider

$$E^v = hB(I - A)^{-1} \tilde{T}^w + \tilde{T}^v.$$

Hence,

$$\frac{\partial E^v}{\partial h} = B(I - A)^{-1} \tilde{T}^w + hB(I - A)^{-1} \frac{\partial \tilde{T}^w}{\partial h} + \frac{\partial \tilde{T}^v}{\partial h}.$$

From Appendix C, we have

$$\frac{\partial E^v}{\partial h} \Big|_{h=0} = B(I - A)^{-1} 0_{n \times 1} + (0)B(I - A)^{-1} d\tilde{T}^w + 0_{n \times 1}.$$

Thus,

$$\frac{\partial E^v}{\partial h} \Big|_{h=0} = 0_{n \times 1},$$

and similarly,

$$\frac{\partial E^p}{\partial h} \Big|_{h=0} = 0_{n \times 1}.$$

Now, consider

$$\frac{\partial^2 E^v}{\partial h^2} = B(I - A)^{-1} \frac{\partial \tilde{T}^w}{\partial h} + B(I - A)^{-1} \frac{\partial \tilde{T}^w}{\partial h} + hB(I - A)^{-1} \frac{\partial^2 \tilde{T}^w}{\partial h^2} + \frac{\partial^2 \tilde{T}^v}{\partial h^2}.$$

From Appendix C, we obtain

$$\frac{\partial^2 E^v}{\partial h^2} \Big|_{h=0} = B(I - A)^{-1} d\tilde{T}^w + B(I - A)^{-1} d\tilde{T}^w + (0)B(I - A)^{-1} dd\tilde{T}^w + 0_{n \times 1},$$

from which it follows that

$$\frac{\partial^2 E^v}{\partial h^2} \Big|_{h=0} = 2B(I - A)^{-1} d\tilde{T}^w.$$

Similarly,

$$\frac{\partial^2 E^p}{\partial h^2} \Big|_{h=0} = -\frac{2}{\alpha} B(I - A)^{-1} d\tilde{T}^u + dd\tilde{T}^p.$$

REFERENCES

- [1] R. ACAR AND C. R. VOGEL, *Analysis of bounded variation penalty methods for ill-posed problems*, Inverse Problems, 10 (1994), pp. 1217–1229.
- [2] C. BRITO-LOEZA AND K. CHEN, *Multigrid algorithm for high order denoising*, SIAM J. Imaging Sci., 3 (2010), pp. 363–389.
- [3] C. BRITO-LOEZA, K. CHEN, AND V. UC-CETINA, *Image denoising using the Gaussian curvature of the image surface*, Numer. Methods Partial Differential Equations, 32 (2016), pp. 1066–1089.
- [4] K. CHEN, *Introduction to variational image-processing models and applications*, Int. J. Comput. Math., 90 (2013), pp. 1–8.
- [5] F. REN, T. QIU, AND H. LIU, *Mean curvature regularization-based poisson image restoration*, J. Electron. Imag., 24 (2015), Art. ID 033025, 15 pages.
- [6] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D 60 (1992), pp. 259–268.
- [7] H. RUI AND H. PAN, *A block-centered finite difference method for the Darcy-Forchheimer model*, SIAM J. Numer. Anal., 50 (2012), pp. 2612–2631.
- [8] L. SUN AND K. CHEN, *A new iterative algorithm for mean curvature-based variational image denoising*, BIT Numer. Math., 54 (2014), pp. 523–553.
- [9] A. N. TIKHONOV, *Regularization of incorrectly posed problems*, Soviet Math. Dokl., 4 (1963), pp. 1624–1627.
- [10] C. R. VOGEL AND M. E. OMAN, *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE Trans. Image Process., 7 (1998), pp. 813–824.
- [11] F. YANG, K. CHEN, B. YU, AND D. FANG, *A relaxed fixed point method for a mean curvature-based denoising model*, Optim. Methods Softw., 29 (2014), pp. 274–285.
- [12] W. ZHU AND T. CHAN, *Image denoising using mean curvature of image surface*, SIAM J. Imaging Sci., 5 (2012), pp. 1–32.
- [13] W. ZHU, X.-C. TAI, AND T. CHAN, *Augmented Lagrangian method for a mean curvature based image denoising model*, Inverse Probl. Imaging, 7 (2013), pp. 1409–1432.