Image restoration with a new class of forward-backward-forward diffusion equations of Perona-Malik type with Applications to Satellite Image Enhancement

Patrick Guidotti*, Yunho Kim†, and James Lambers‡

Abstract. A new class of anisotropic diffusion models is proposed for image processing which can either be viewed as a novel kind of regularization of the classical Perona-Malik model or, as it is advocated by the authors, as a new independent model. The models are diffusive in nature and are characterized by the presence of both forward and backward regimes. In contrast to the Perona-Malik model, in the proposed model the backward regime is confined to a bounded region and gradients are only allowed to grow up to large but tunable size thus effectively preventing indiscriminate singularity formation, i.e. staircasing. Extensive numerical experiments demonstrate that the method is a viable denoising/deblurring tool. The method is significantly faster than competing state-of-the-art methods and appears to be particularly effective for simultaneous denoising and deblurring (cf. Section 4.1). An application to satellite image enhancement is also presented.

Key words. Nonlinear diffusion, forward-backward diffusion, denoising, deblurring, satellite images

AMS subject classifications. 35K55, 68U10

1. Introduction. The Perona-Malik model, while of considerable theoretical interest, has been limited in its applicability by its strong propensity to singularity formation among other reasons. This problem results in images with strong cartoonish look due to staircasing. TV-based models have experienced tremendous success but typically lack any sharpening capability. The proposed model offers a handle (two regularization parameters) which makes it possible to harvest the desirable sharpening features of Perona-Malik type models in a controlled fashion. This is done by introducing a very mild regularizing term, which still allows gradient growth while effectively introducing a bound for its maximal size. It is obtained as the gradient flow associated to

$$E_{p,\delta}(u) = \int_\Omega \Big[\frac{1}{2}\log(1+|\nabla u|^2) + \delta\frac{|\nabla u|^p}{p}\Big]\, dx\,, \tag{1.1}$$

where $\delta > 0$ and $p \in (1, 2]$. This is an eventually convex functional which exhibits a concave region of the form $[1 \leq |\nabla u| \leq M(p,\delta) < \infty]$ responsible for the controlled growth of gradients. While the functional with $p = 2$ was introduced in [1] for theoretical purposes such as understanding the onset of staircasing for Perona-Malik, it provides, as it is advocated here, a viable tool in applications to image processing, especially for $p \gtrsim 1$. It should be observed that the model obtained by choosing $p = 1$ would effectively lead to (a δ-sloped) TV model. Indeed the relaxation of $E_{1,\delta}$ turns out to be a rescaled TV functional in that case. Numerical

*Department of Mathematics, 340 Rowland Hall, University of California at Irvine, Irvine, CA 92697-3875 (gpatrick@math.uci.edu)

†Department of Diagnostic Radiology, School of Medicine, Yale University, TAC N309, 300 Cedar St., New Haven, CT 06520 (yunho.kim@yale.edu)

‡Department of Mathematics, 118 College Dr 5045, University of Southern Mississippi, Hattiesburg, MS 39406-0001 (James.Lambers@usm.edu)

experiments based on a discretization of the nonlinear PDE (1.2) described later indicate that $p = 1$ is, however, not an optimal choice in applications. Best results are obtained for p close but not equal to one.

Just as for Perona-Malik, the corresponding gradient flow

$$u_t = \nabla \cdot \left[\left(\frac{1}{1 + |\nabla u|^2} + \delta |\nabla u|^{p-2} \right) \nabla u \right], \tag{1.2}$$

is of forward-backward type, at least for δ small enough. As pointed out above, the backward regime is confined to the annulus where

$$[1 < |\nabla u| < M(\delta, p)], \tag{1.3}$$

whereas the backward regime in Perona-Malik is the outer domain where $|\nabla u| > 1$. It is essentially a consequence of (1.3) that solutions of (1.2) enjoy better regularity properties than those of PME, i.e., they never develop any discontinuities. These equations provide a well-posed family of models which are not only amenable to analysis (cf. [1]) but also offer practical benefits such as performance enhancement and robustness in the simultaneous presence of noise and blur. In a way, the new models are an interpolation of two classical models, PM and TV. This is essentially due to the fact that solutions u of (1.2) evolve as solutions of Perona-Malik as long as the gradient is large but not too large, whereas they exhibit a (p-Laplacian type, $p \sim 1$) diffusion for small and very large gradients. It has been proven that the discontinuity set of solutions to TV can not grow [2] and it is well-known that Perona-Malik has strong sharpening features. The tunable parameters δ and p of the new model makes it possible to benefit from the sharpening features of Perona-Malik while still enjoying many of TV's desirable properties. Extensive numerical experiments clearly support this theoretically grounded claim.

It is worthwhile reiterating that (1.2) is not well-posed in the sense of classical solutions but rather in the weaker sense of Young measure valued solutions. Latter are constructed via time discretization and successive minimization (cf. e.g. [3, 1]). They do, however, have center of masses which evolve according to the convexification of the driving energy, which in this case and in contrast to Perona-Malik, is not trivial.

2. Properties of the Model. It is well-known that staircasing is an artifact which appears in discretizations of the Perona-Malik equation. It is a manifestation of the forward-backward nature of the equation and a typical occurrence is depicted in Figure 2.1. Indeed, a solution, in the attempt to flee the backward regime, is forced into developing vanishing or infinite gradients thus generating the staircasing phenomenon. The proposed regularization, on the other hand, reduces the size of the backward region and solutions can escape the backward region merely by developing small and large gradients. Their size is effectively controlled by the parameters δ and p with δ playing the dominant role as explained later. Staircasing is effectively replaced by a form of micro-structured ramping as can be clearly seen in Figure 2.2. In this way it is still possible to exploit the sharpening properties of Perona-Malik without incurring its full cost.

The micro-oscillatory phenomenon, which is akin to a similar one well-known in material science, can be mathematically captured by the use of Young measures (in particular weak Young measure valued solutions in the sense of [3]). This is done in [1] where, for $p = 2$,
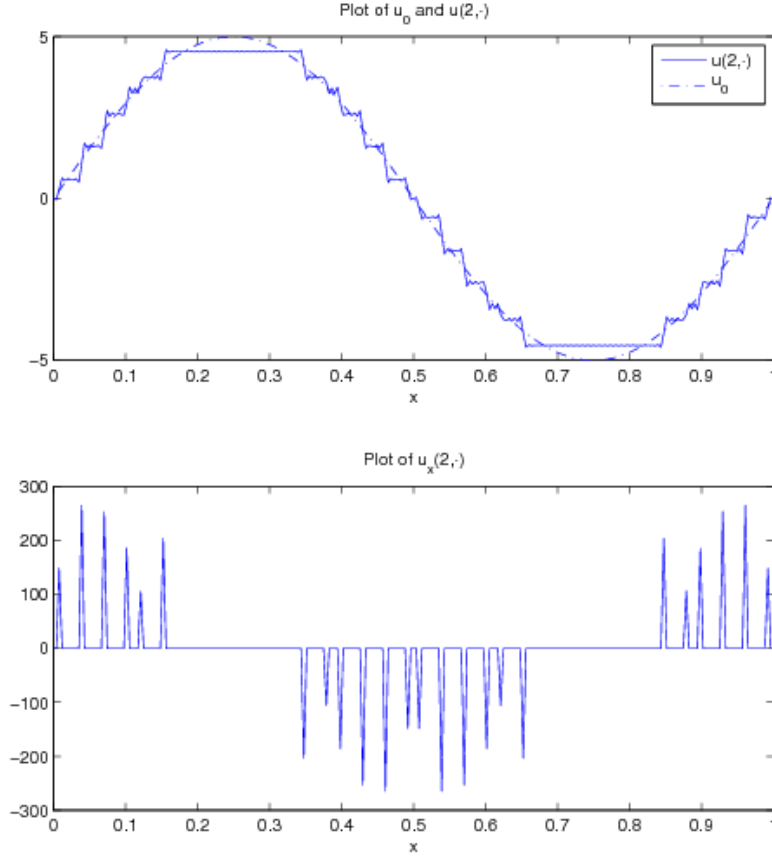
Figure 2.1. *Manifestation of the staircasing phenomenon widely observed in discretizations of the Perona-Malik equation*

equation (1.2) is shown to be globally (in time) well-posed and the micro-structured nature of the gradients of its solutions is fully described by the use of Young measures. This remains
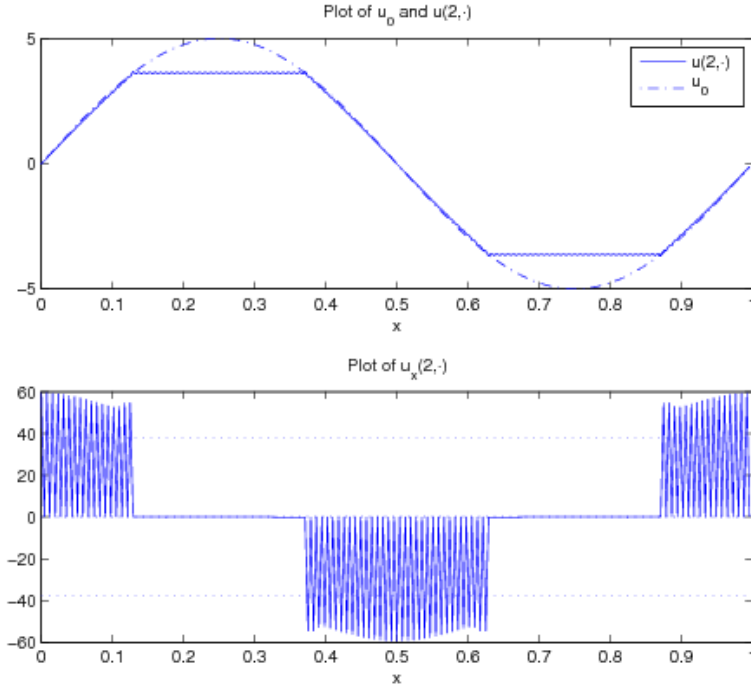
Figure 2.2. *The micro-ramping phenomenon exhibited by* (1.2)

true for values of $p < 2$ and further comprehensive investigation in that direction is being pursued. It is referred to [1] for the details with $p = 2$. It is pointed out that the case $p = 1$ is not of interest in this paper for the reasons cited above. For $p = 1$, $E_{1\delta}$ possesses the same

backward regime as the original Perona-Malik model. Moreover, the convexification $E_{1,\delta}^{**}(u)$ of $E_{1,\delta}(u)$ is given by

$$E_{1,\delta}^{**}(u) = \delta \int_\Omega |\nabla u| dx,$$

and

$$E_{1,\delta}^{**}(u) < E_{1,\delta}(u) \quad \text{unless} \quad \nabla u \equiv 0.$$

Hence $p = 1$ is not an appropriate choice for the development of a confined backward regime as discussed earlier. On the other hand, for $p > 1$, the following minimization problem can be considered

$$\inf_{u \in W^{1,p}(\Omega)} \left\{ F_{p,\delta}(u) = E_{p,\delta}(u) + G(u) \right\}, \tag{2.1}$$

for a convex G. Then, due to the properties of $E_{p,\delta}(u)$ and by convexification[1], one is led to

$$\inf_{u \in W^{1,p}(\Omega)} \left\{ F_{p,\delta}(u) = E_{p,\delta}(u) + G(u) \right\} = \inf_{u \in W^{1,p}(\Omega)} \left\{ F_{p,\delta}^{**}(u) = E_{p,\delta}^{**}(u) + G(u) \right\}. \tag{2.2}$$

Since it is not the purpose of this paper to provide analytical details of this variational model and its associated PDE model (1.2), they will be considered in a subsequent paper. In the experiments performed later in this paper, two distinct functions G are considered

$$G(u) = \begin{cases} 0, & \text{for denoising,} \\ \int_\Omega |u_0 - Ku|^2 dx, & \text{for deblurring or denoising-deblurring,} \end{cases}$$

with the aim to understand how (1.2) affects the recovery from degraded images by noise and blur. Since the images of interest are 2D and boundary effects can be dealt with by introduction of a buffering frame around the image, periodicity will be assumed with periodicity box given by $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$.

Theorem 2.1. *Given a periodic function $u_0 \in L^2(\Omega)$ and $1 < p \leq 2$, the minimization problem (2.1) has a minimizer on Ω in $W_{per}^{1,p}(\Omega)$ regardless of which function G*

$$G \equiv 0 \quad \text{or} \quad G(u) = \int_\Omega |u_0 - Ku|^2 dx$$

is chosen. Here K is a convolution operator with nonnegative kernel $k \in L^1(\Omega)$ satisfying $\int_\Omega k(x)dx = 1$.

Proof. For $G \equiv 0$, it is obvious that any constant is a minimizer. For the other G, choose a minimizing sequence $\{u_n\}_{n \in \mathbb{N}}$. Due to relaxation (2.2), the proof is classical. The uniform boundedness of $\|Ku_n\|_{L^2(\Omega)}$ implies uniform boundedness of $\int_\Omega u_n(x)dx$ since

$$\left| \int_\Omega u_n(x)dx \right| = \left| \int_\Omega k * u_n(x)dx \right| \leq \|Ku_n\|_{L^2(\Omega)},$$

---

[1]While it is not true in general that $F^{**} + G$ coincides with the convexification of $F + G$ for a convex function G, it is true for the functions considered here.

which leads to the uniform boundedness of $\|u_n\|_{W^{1,p}(\Omega)}$ through Poincaré's inequality. Then, Rellich's compactness theorem implies the existence of a weak limit $\tilde{u}$ such that

$$u_n \rightharpoonup \tilde{u} \text{ in } W^{1,p}(\Omega), \text{ and } u_n \to \tilde{u} \text{ in } L^2(\Omega).$$

$W^{1,p}$ sequential lower semicontinuity of $E_{p,\delta}^{**}$ finally implies that $\tilde{u}$ is a minimizer on Ω. ∎

For some details, the reader is referred to [3]. While this variational problem is related to the proposed model (1.2), the interest here is mainly in the behavior of the associated PDE (1.2), which inherits a nice property from the Perona-Malik equation: it does enhance edges. Yet, unlike the Perona-Malik equation, it is well-posed as can be shown using Young measure-valued solutions. It should be stressed that this kind of regularization with a confined backward regime is different from others proposed before. The extensive numerical results about (1.2) presented later will clearly underscore and strongly corroborate this important claim.

We conclude this section with a couple of remarks. First notice that a mathematically rigorous connection between the appropriate gradient flow of (1.1) and the TV flow has been obtained in [30]. In their paper they use Γ-convergence techniques combined with the concept of maximal slope curves to show that time-rescaled gradient flow curves of (1.1) do indeed converge to gradient flow curves to the total variation energy. They obtain their result for $p = 2$. While this paper is motivated by the Perona-Malik model and its subsequent modifications/variations, it should be pointed out that the use of non-convex regularization/penalization was also suggested in [31] in a Bayesian framework. Finally an analytical and numerical study of the well-known staircasing phenomenon was performed by [29] by using a fourth order regularization in a one dimensional setting.

3. Implementation. In this section, the attention is focused on the implementation of an efficient numerical method for nonlinear diffusions of the form

$$u_t = \nabla \cdot (g(\nabla u)\nabla u) \tag{3.1}$$

on $[0,1]^2$, such as (1.2) is, with appropriate initial data and either periodic or Neumann boundary conditions.

3.1. Spatial Discretization. Details about the spatial discretization of (3.1) are given first. The one-dimensional periodic differentiation operator D is discretized spectrally by means of the discrete Fourier transform $\mathcal{F}_N$, which yields the discretization

$$D_N = \mathcal{F}_N^{-1} \Lambda_N \mathcal{F}_N \tag{3.2}$$

where N denotes the number of grid points used and

$$\Lambda_N = 2\pi i \operatorname{diag}\left(-\frac{N}{2}+1, -\frac{N}{2}+2, \ldots, 0, 1, \ldots, \frac{N}{2}\right).$$

Then, the discretization of the two-dimensional differentiation operator D_z, where $z = x, y$, has the form

$$D_{z;N,N} = \mathcal{F}_{N,N}^{-1} \Lambda_{z;N,N} \mathcal{F}_{N,N},$$

where $\mathcal{F}_{N,N}$ is the two-dimensional DFT operator and

$$\Lambda_{x;N,N} = I_N \otimes \Lambda_N \,, \quad \Lambda_{y;N,N} = \Lambda_N \otimes I_N.$$

That is, $\Lambda_{z;N,N}$ is the diagonal matrix of eigenvalues of $D_{z;N,N}$. The resulting system of ODEs is

$$\frac{d\mathbf{u}}{dt} = L_{N,N}\mathbf{u}\,, \tag{3.3}$$

where

$$L_{N,N} = \sum_{z=x,y} D_{z;N,N} G_{N,N} D_{z;N,N},$$

and $G_{N,N}$ is a diagonal matrix consisting of the values of $g(\nabla u)$ at the grid points.

3.2. Time-Stepping. For diffusion equations, it is generally not practical to use explicit time-stepping methods, because of the severe constraints they impose on the time step. A straightforward alternative is to use backward Euler for time-stepping, and an iterative method such as MINRES [25] to solve the associated system of equations.

However, there are two significant drawbacks with this approach. First, the number of iterations required by such methods tends to be very large, unless the time step is prohibitively small. Second, even when convergence to a reasonably small tolerance is achieved, this is not sufficient for image denoising or deblurring, because solutions tend to exhibit staircasing and high-frequency oscillations. In view of the difficulties associated with both explicit and implicit time-stepping methods, it is worthwhile to consider an alternative approach to solving the system of ODEs (3.3).

For problems with periodic or Neumann boundary conditions, spectral methods are an attractive option since the solution can be expressed as a linear combination of basis functions that automatically satisfy the boundary conditions. They are quite effective for problems with constant coefficients, because then these basis functions are eigenfunctions of the spatial differential operator, which allows the Fourier coefficients (or Fourier cosine coefficients, in the case of Neumann boundary conditions) of the solution to be evolved in time, independent of one another.

This property does not extend to variable-coefficient problems, even in the linear case. However, even though the Fourier coefficients are coupled, it is still possible to evolve each of them forward in time using an approximation to the solution operator that, in some sense, is optimal for that coefficient. This approach allows high-order accuracy in time, and favorable stability properties, to be realized with a far lower-dimensional Krylov subspace than is feasible for iterative methods such as MINRES that are used within implicit time-stepping schemes.

This is the basic idea behind *Krylov subspace spectral (KSS) methods*. They were first developed in [20] for the purpose of solving parabolic variable-coefficient problems. Let $S(t) = \exp(Lt)$ represent the exact solution operator of the one-dimensional problem $u_t = Lu$, with periodic boundary conditions, where L is a second-order variable-coefficient differential operator. Functions are discretized on an N-point uniform grid, and $\mathbf{u}^n$ is used to denote the computed solution at time t_n, where $t_n = n\Delta t$ for some time step Δt. Then, each Fourier coefficient of the solution $u(x,t)$ at time t_{n+1} can be approximated by the bilinear form

$$\hat{u}(\omega, t + \Delta t) \approx \sqrt{\Delta x}\hat{\mathbf{e}}_\omega^H S_N(\Delta t)\mathbf{u}^n, \tag{3.4}$$

where $\hat{\mathbf{e}}_\omega$ is a discretization of $\exp[2\pi i \omega x]$, and $S_N = \exp[L_N \Delta t]$, where L_N is an $N \times N$ matrix obtained by spectral discretization of L.

In [10] Golub and Meurant describe a method for computing quantities of the form

$$\mathbf{u}^T f(A) \mathbf{v}, \tag{3.5}$$

where $\mathbf{u}$ and $\mathbf{v}$ are N-vectors, A is an $N \times N$ symmetric matrix, and f is a smooth function. KSS methods, introduced in [18, 20], apply this method with $A = L_N$, $f(\lambda) = \exp(\lambda t)$ for some t, and the vectors $\mathbf{u}$ and $\mathbf{v}$ are equal to $\hat{\mathbf{e}}_\omega$ and $\mathbf{u}^n$, respectively.

The basic idea behind the approach of Golub and Meurant is that the bilinear form (3.5) can be viewed as a Riemann-Stieltjes integral

$$\mathbf{u}^T f(A) \mathbf{v} = I[f] = \int_a^b f(\lambda) \, d\alpha(\lambda),$$

where a and b are the smallest and largest eigenvalues of A, respectively, and the measure $\alpha(\lambda)$ is derived from the components of $\mathbf{u}$ and $\mathbf{v}$ in the basis of eigenvectors of A. As discussed in [5, 9, 8, 10], the integral $I[f]$ can be approximated using Gauss quadrature rules. The nodes and weights can be obtained using the symmetric Lanczos algorithm if $\mathbf{u} = \mathbf{v}$, and the unsymmetric Lanczos algorithm if $\mathbf{u} \neq \mathbf{v}$; see [12]. Alternatively, for the case $\mathbf{u} \neq \mathbf{v}$, a block approach can be used, based on block Lanczos iteration [11]. Applying this approach to implement spectral methods significantly improves accuracy over the unsymmetric Lanczos algorithm [17].

K-node KSS methods applied to diffusion equations achieve $O(\Delta t^{2K-1})$ accuracy in time, and, given sufficient regularity of the coefficients, the 1-node KSS method can be shown to also exhibit unconditional stability [16, 17]. In general, KSS methods provide a "best-of-both worlds" compromise between the efficiency of explicit time-stepping methods and the stability of implicit methods. These desirable properties are due to their use of quadrature rules that are tailored to each Fourier coefficient of the solution, but this adaptivity opens the door to other benefits as well. For example, Fourier coefficients can be computed in parallel, with the only shared data being a low-dimensional Krylov subspace generated by the solution from the previous time step.

KSS methods compute a Jacobi matrix corresponding to *each* Fourier coefficient, in contrast to traditional Krylov subspace methods (see, for example, [13, 14, 23]) that normally use only a single Krylov subspace generated by the initial data or the solution from the previous time step. While it would appear that KSS methods incur a substantial amount of additional computational expense, that is not actually the case, because nearly all of the Krylov subspaces that they compute are closely related by the wave number ω, in the 1-D case, or $\vec{\omega} = (\omega_1, \omega_2, \ldots, \omega_n)$ in the n-D case.

In fact, the only Krylov subspace that is explicitly computed is the one generated by the solution from the previous time step, of dimension $(K+1)$, where K is the number of block Gaussian quadrature nodes (which corresponds to $2K$ scalar nodes). In addition, the averages of the coefficients of L^j, for $j = 0, 1, 2, \ldots, 2K-1$, are required, where L is the spatial differential operator. The overall computation can be carried out in $O(N \log N)$ operations per time step using symbolic calculus [19, 21].

With these considerations, the algorithm for a single time step of a 1-node block KSS method for solving (3.1), with appropriate initial conditions and periodic boundary conditions, is as follows. The average of a function $f(x)$ on $[0, 2\pi]$ is denoted by $\overline{f}$, and, as before, the computed solution at time t_n by u^n. The operator L is a linearization of the operator on the right-hand side of (3.1), defined by $Lu = \nabla \cdot (g(\nabla u^n)\nabla u)$.

$\hat{u}^n = \textbf{fft}(u^n)$, $v = Lu^n$, $\hat{v} = \textbf{fft}(v)$

for each ω **do**

$\quad a_{11} = -\overline{g}(2\pi\omega)^2$

$\quad a_{12} = \hat{v}(\omega) - a_{11}\hat{u}^n(\omega)$

$\quad a_{22} = \langle u^m, v \rangle + a_{11}|\hat{u}^n(\omega)|^2 - 2\,\text{Re}\,[\hat{u}^n(\omega)\overline{v(\omega)}]$

$\quad f_\omega^2 = \|u^n\|_2^2 - |\hat{u}^n(\omega)|^2$

$\quad a_{12} = a_{12}/f_\omega$

$\quad a_{22} = a_{22}/f_\omega^2$

$\quad r = a_{22} - a_{11}$

$\quad b_{12} = |a_{12}|^2$

$\quad d = \sqrt{r^2 + 4b_{12}}$

$\quad s = \text{sgn}(r)$

$\quad t = -2s/(sa_{21} + d)$

$\quad w_1 = 1/\sqrt{1 + b_{12}t^2}$

$\quad w_2 = \overline{a_{12}}tw_1$

$\quad \lambda_1 = a_{11} + b_{12}t$

$\quad \lambda_2 = a_{22} - b_{12}t$

$\quad e_1 = \exp(\lambda_1 \Delta t)$

$\quad e_2 = \exp(\lambda_2 \Delta t)$

$\quad \hat{u}^{n+1}(\omega) = \hat{u}^n(\omega)(|w_1|^2 e_1 + |w_2|^2 e_2) + f_\omega w_1\overline{w_2}(e_1 - e_2)$

end

$u^{n+1} = \textbf{ifft}(\hat{u}^{n+1})$

The quantities a_{11}, a_{12} and a_{22} are entries of a 2×2 Hermitian negative definite matrix that is the $(1, 1)$ block of the block tridiagonal matrix produced by block Lanczos iteration. This block has eigenvalues λ_1 and λ_2, which are the two scalar nodes for 1-node block Gaussian quadrature. The quantities w_1 and w_2 are the first components of the normalized eigenvectors corresponding to λ_1 and λ_2, respectively; they are used to obtain the entries of the single block weight.

The above algorithm is first-order accurate in time, due to the use of one block Gauss quadrature node. Future work involves the generalization of existing higher-order KSS methods for linear PDE to the nonlinear case by combination with methods based on exponential integrators, such as those described in [28].

4. Numerical Experiments. This section is devoted to extensive numerical experimentation. One of the advantages of the proposed model is, unlike other models which are usually specialized for one task (either image denoising or image deblurring), its applicability to both image denoising and deblurring problems. The two parameters δ and p in the model determine the convex and the concave regions. By tuning δ and p, those regions can be enlarged or reduced and this yields control over the forward or backward nature of diffusion. In all experiments, gray-scale images of size 256×256 are used. The emphasis is on the recovery of

noisy blurred images. Therefore, the most important results will be presented first. Later a variety of pure denoising and deblurring experiments will also be considered which illustrate the model's performance as a function of the parameters δ, p.

4.1. Noisy blurred case. Experiments with noisy blurred images, which are the kind of images on which model (1.2) proves to perform the best, are presented in this section. Clearly the smoothing effect sought for when denoising images is not easily reconciled with the need for sharpening in the deblurring process. It follows that image recovery for images which are simultaneously corrupted by noise and blur is a highly ill-posed problem. The parameter $\delta > 0$ of the model makes it, however, possible to walk the fine line between smoothing and sharpening and yields a viable method for a reasonable recovery. Again, the fine tuning of the exact location of the backward regime will enable this.

Blur and noise are assumed to be known. The noise is considered to be Gaussian white noise with variance σ^2. As for blur, it is referred to Table 4.5 below. The exact procedure used to generate noise and blur is described in the following sections. To deal with blur, the model needs to incorporate a priori knowledge about the blur, so an additional term is added to (1.2) yielding

$$u_t = \nabla \cdot \left[\left(\frac{1}{1 + |\nabla u|^2} + \delta |\nabla u|^{p-2} \right) \nabla u \right] + \lambda K'(u_0 - Ku). \tag{4.1}$$

This is the gradient descent flow of the modified functional F,

$$F(u) = E_{p,\delta}(u) + \lambda \int_\Omega |u_0 - Ku|^2(x)dx, \tag{4.2}$$

which includes a blurring operator K. Its adjoint operator, which appears in (4.1), is denoted by K'. Numerically, the equation

$$u_t = \nabla \cdot \left[\left(\frac{1}{1 + |\nabla u|^2} + \delta |\nabla u|^{p-2} \right) \nabla u \right] + \lambda k' * (u_0 - k * u) \tag{4.3}$$

is solved with $u(\cdot, 0) = u_0$. The blurring operator is taken in the form $Ku = k * u$ with a blurring kernel k. The additional forcing term is needed not only in order to boost edge sharpening, but also to recover fine scale texture. While better choices for the forcing term may be found, the decision was made to use the simplest one in this paper. Throughout all experiments $\lambda = 30$. The main advantage of the proposed model is the possibility of tuning the character of the diffusion by varying the two parameters δ and p in (1.2). The first integrand in (1.1) is the energy functional for the Perona-Malik equation and thus introduces the backward regime $[|\nabla u| > 1]$ needed for edge sharpening. The second and regularizing integrand in (1.1), on the other hand, shrinks the backward regime to finally give (1.3). The effects of varying δ and p will be studied in the following sections.

In the nth iteration, one solves (4.3) for one time step with a step-dependent δ_n. It is natural to assume that the values of δ_n need adjustment depending on the amount of noise and blur present in the image. It was experimentally determined that behaviors of δ_n/δ such as the one chosen below would be appropriate. Basically, this choice makes the denoising effect dominant during the initial iterations, when one aims at a significant noise reduction.

The deblurring effect then takes over and becomes more pronounced as more iterations are performed. The starting value of δ is set at 0.02 whereas

$$\delta_n = \begin{cases} (1 + n/10)\,\delta & \text{for } 1 + n/10 < 5\,, \\ \max[0.1, 5 - (n - n_1)/10]\,\delta & \text{for } n \geq n_1\,, \end{cases}$$

where n_1 satisfies $1 + n_1/10 < 5 \leq 1 + (n_1 + 1)/10$.

In numerical experiments $\sigma = 5, 8$ were used. With noise level $\sigma = 5$, δ_n was chosen as indicated above, whereas when $\sigma = 8$ the choice was

$$\delta_n = \begin{cases} (1 + n/10)\,\delta & \text{for } 1 + n/10 < 7\,, \\ \max[0.1, 7 - (n - n_1)/10]\,\delta & \text{for } n \geq n_1, \end{cases}$$

with n_1 satisfying $1 + n_1/10 < 7 \leq 1 + (n_1 + 1)/10$.

In Figure 4.1, the graph of δ_n/δ for the first case is shown. Numerically this graph shape resulted in better recovery than simple use of a fixed high value of δ for a few iterations followed by a lower δ value in the remaining iterations.
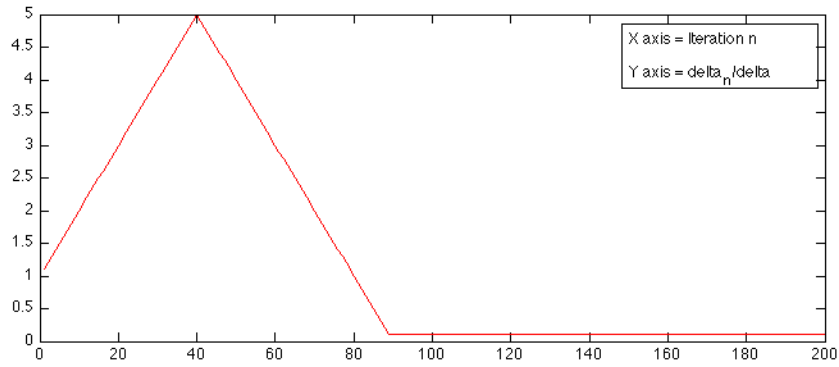

Figure 4.1. *Graph of δ_n/δ.*

Next recovery results obtained for noisy blurred images are presented for various noise levels and blur types (cf. Table 4.5). In all experiments it was observed that both edge sharpening and noise reduction were effectively taking place from the very first iteration. This is somewhat contrary to the expectation that the smoothing effect would prevail while δ_n grows and the sharpening effect would be predominant while δ_n decreases. It was also observed that noise tends to reappear even as images get sharpened, if (4.3) is solved on a longer time interval with a small δ_n.

In Figure 4.2, two types of blur were used, 3×3 average blur of type I and 5×5 out-of-focus blur of type IV with noise level set at $\sigma = 5$, and $\sigma = 8$, respectively.

In Figure 4.3, a 5×5 out-of-focus blur of type IV was used with the noise level set at $\sigma = 5, 8$, respectively.

PSNR values for the experiments shown in Figures 4.2–4.3 are given in Table 4.1 below.

Finally the proposed model is compared with other state-of-the-art algorithms. In Figure 4.4, a 15×15 Gaussian blur of type VI was used to blur the original image and noise was
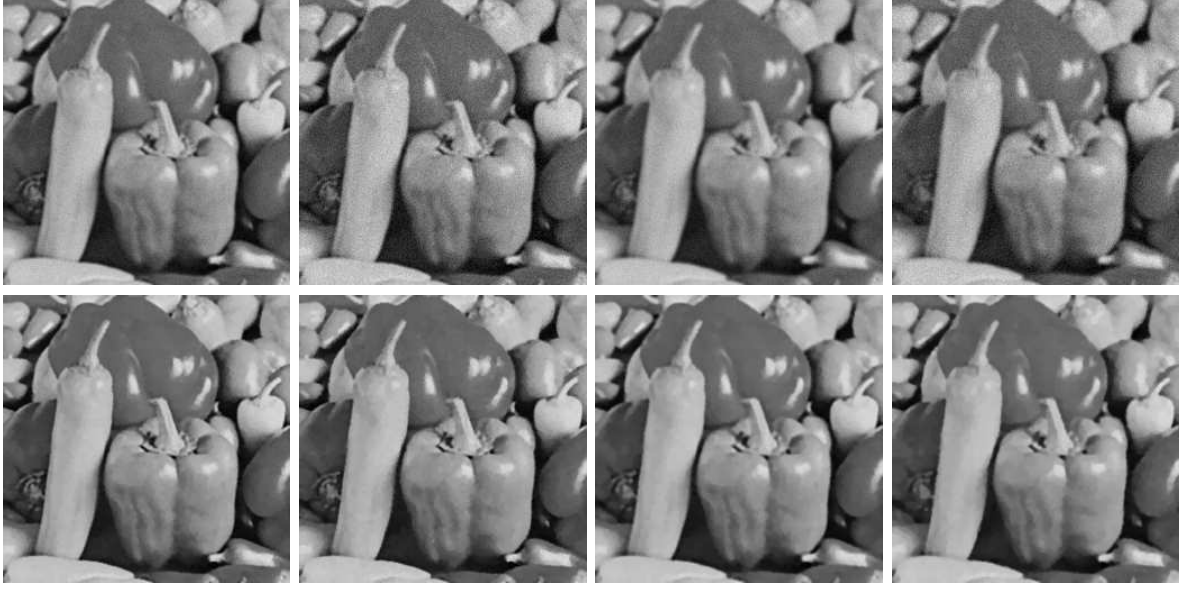
Figure 4.2. **Top Row** - *Noisy blurred images,* **Bottom Row** - *Recovered images corresponding to the above images.* **Left two columns** - 3×3 *average blur of type I and noise levels* $\sigma = 5, 8$, *respectively.* **Right two columns** - 7×7 *Gaussian blur of type V and noise levels* $\sigma = 5, 8$, *respectively.*

Figure 4.3. **Top Row** - *Noisy blurred images with* 5×5 *out-of-focus blur of type IV and noise levels* $\sigma = 5$ *(left two),* $\sigma = 8$ *(right two),* **Bottom Row** - *Recovered images corresponding to the above images.*

added of level $\sigma = 3$. The output of the proposed model is compared with the frame based algorithm presented in [6]. Table 4.2 offers a side-by-side comparison of the corresponding PSNR values. The proposed model delivers acceptable results as compared to the algorithm in [6] in a shorter amount of time. This proves that a PDE based method can indeed be very

			PSNR (noisy)	recovered PSNR (recovered)
Figure 4.2	Left two	$\sigma = 5$	29.74	33.56
		$\sigma = 8$	27.76	32.25
	Right two	$\sigma = 5$	28.52	32.16
		$\sigma = 8$	26.96	31.21
Figure 4.3	Left two	$\sigma = 5$	26.50	30.58
		$\sigma = 8$	26.96	31.21
	Right two	$\sigma = 5$	26.72	29.67
		$\sigma = 8$	25.62	28.90

Table 4.1
PSNR values for the experiments shown in Figures 4.2–4.3.

efficient.

Figure 4.4. *Recovery from a noisy blurred image of size 512×512 with 15×15 Gaussian blur of type VI and noise $\sigma = 3$.* **Left***: Noisy blurred image,* **right***: Recovered image*

	PSNR	Time(s)
Proposed model	24.33	13.2
Single-System ($\kappa = 1, \lambda = 0.005$) in [6]	24.64	20.7
Two-Systems ($\kappa^i = 1, \lambda^i = 0.005$) in [6]	24.75	63.2
Linearized Bregman ($\kappa = 1, \lambda = 0.01$) in [6]	24.67	126.2
Analysis Approach ($\mu = 0.1, \lambda = 0.05$) in [6]	24.58	32.1

Table 4.2
Recovery experiment (15×15 Gaussian blur and additive noise $\sigma = 3$). Figure 4.4 was obtained using the proposed model. It is referred to [6] for further details concerning the results used as a comparison.

In Figure 4.5 another comparison is offered that emphasizes the efficiency of the proposed model as a recovery tool for noisy blurred images.

Figure 4.5. Left to Right: *Noisy blurred image (3×3 average blur of type I and $\sigma = 5$), recovered image from [15], recovered image with proposed model.*

The proposed method is also compared with ROF ([26]), ForWaRD ([24]), AKTV ([27]), BM3D([4]), and DSDM (the direct sparse deblurring method, [22]). The same comparison experiments presented in [22] are borrowed and performed here.

Image	ROF	ForWaRD	Proposed Method	AKTV	BM3D	DSDM
Text1	65.43	62.75	41.85	60.68	69.87	14.60
Rose	6.68	5.75	5.24	4.90	4.83	5.62
Koala	9.80	8.98	8.05	8.45	7.97	8.64
Castle	14.23	12.82	13.79	12.51	11.77	13.62

Table 4.3
RMS Errors for different methods.

Notice that proposed method performs better as compared with ROF and ForWaRD and that it performs as well as the AKTV and DSDM methods but in a much shorter amount of time. In particular, DSDM and AKTV take much longer due to dictionary learning and weight updates, respectively. Figure 4.6 depicts, from left to right, a noisy blurred image, the output image of AKTV, and the output of the proposed method. While AKTV was able to recover a better image, the proposed method computed its output (on the right) in only 26 seconds. This is to be compared with the 6457 seconds it took AKTV to deliver its output image (on the right). In fact, AKTV does not even recover an image of comparable quality to that of the proposed method in hundreds of seconds in the experiment presented in Figure 4.6.

Lastly, a concrete application to satellite images is considered. While we had no access to images taken from satellites, the same procedure described and implemented in [7] was followed in order to simulate the typical degradation of satellite images, in order to appropriately compare degraded and restored images, etc. Firstly the same convolution kernels were used as the ones described in [7] corresponding to two different satellites: SPOT 5 and a specific mode of SPOT 1.

Figure 4.6. **Left to Right***: Noisy blurred image of size 512×512 (5×5 gaussian blur of $\sigma = 1.5$ and noise with standard deviation 8.2), recovered image by AKTV [27] (RMSE = 6.12, PSNR = 32.39, Elapsed time = 6457 seconds), recovered image with proposed model (RMSE = 6.79, PSNR = 31.48, Elapsed time = 26 seconds)*

Figure 4.7. *Ideal reference images of size 256×256*

The two convolution kernels with corresponding levels of Gaussian noise in [7] are

$$\hat{h}_1(\xi, \eta) = e^{-2\gamma_\xi |\xi| - 2\gamma_\eta |\eta|} \Big(\frac{\sin(2\pi\xi)}{2\pi\xi} \Big) \Big(\frac{\sin(2\pi\eta)}{2\pi\eta} \Big) \Big(\frac{\sin(\pi\eta)}{\pi\eta} \Big), \ \xi, \eta \in \Big[-\frac{1}{2}, \frac{1}{2} \Big],$$

where $\gamma_\xi = 1.505, \gamma_\eta = 1.412$ and the standard deviation for the Gaussian noise is $\sigma_1 = 2.4$, and

$$\hat{h}_2(\xi, \eta) = e^{-2\gamma_\xi |\xi| - 2\gamma_\eta |\eta|} \Big(\frac{\sin(4\pi\xi)}{4\pi\xi} \Big) \Big(\frac{\sin(4\pi\eta)}{4\pi\eta} \Big), \ \xi, \eta \in \Big[-\frac{1}{2}, \frac{1}{2} \Big],$$

where γ_ξ, γ_η are the same as before and the standard deviation for the Gaussian noise is $\sigma_2 = 0.5$. Two reference images are shown in Figure 4.7 which will play the role of ideally clean images and to which the restored images are compared. These reference images are obtained by applying a prolate blur function to oversampled sharp images followed by downsampling. The reader is referred to [7] for more details about the experiment. In this section, a comparison is offered between the proposed method and the simple ROF model for image restoration.
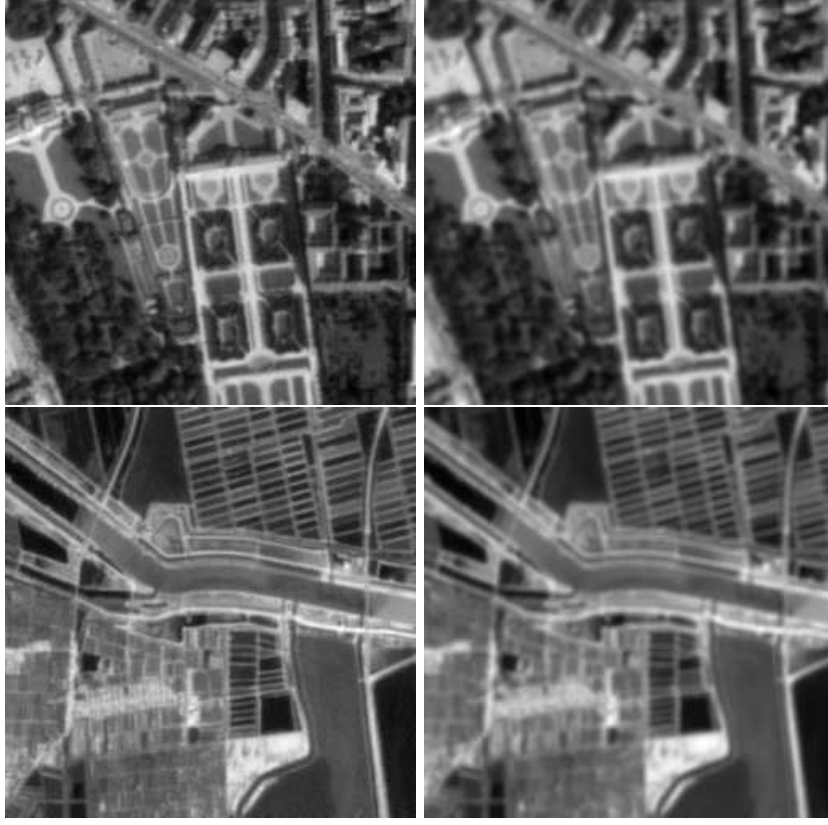
Figure 4.8. *Left column : degraded images of size* 256×256 *with* (h_1, σ_1)*, Right column : degraded images of size* 256×256 *with* (h_2, σ_2)*.*

Figure 4.8 shows two degraded images using (h_1, σ_1) and (h_2, σ_2). Figure 4.9 and Figure 4.10 depict the images restored by the ROF and the proposed model, respectively. For better visual comparisons, a few smaller regions were chosen from the reference and from the restored images. For (h_1, σ_1) the proposed model performs on par with the ROF model. However, for (h_2, σ_2), when there is a significant amount of blur, the proposed model outperforms the ROF model. It is known that the ROF model does not capture small details well and these experiments are a further confirmation of this. The proposed model, on the other hand, is able to recover small details as long as they are characterized by big changes in the modulus of the gradient. This is evident in the last two columns of Figures 4.9 and 4.10. The same columns, however, also point to a potential susceptibility to artifacts of the proposed method.

4.2. Noisy case. In the following experiments, Gaussian white noise with standard deviation $\sigma = 5, 10, 15, 25, 35, 50$ are considered. This and the next sections will illustrate in which way the proposed model's performance is affected by the choice of the two parameters δ, p. Given a clean image $\tilde{f}$, a noisy image f is generated in MATLAB by

$$f = \tilde{f} + \sigma \times \mathrm{randn}(\mathrm{size}(f)).$$
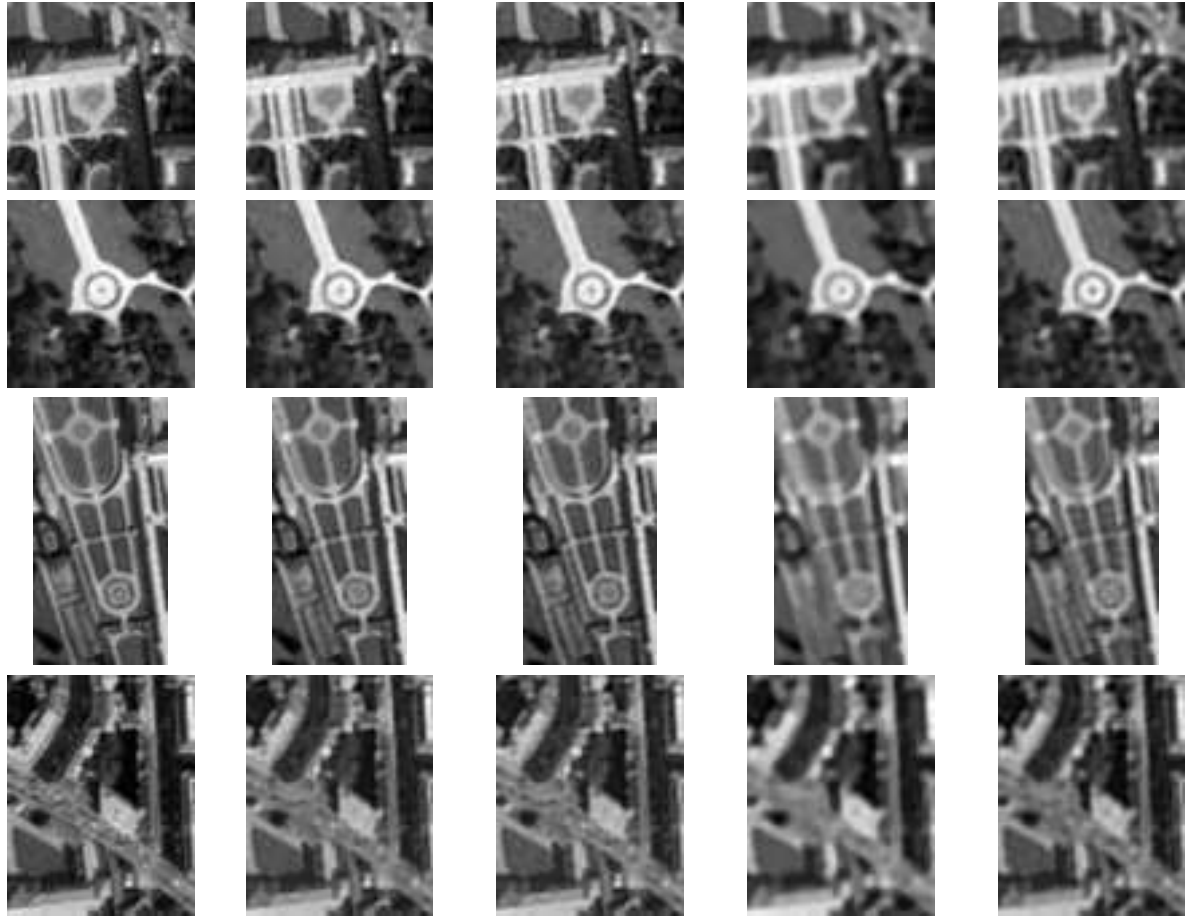
Reference image I   ROF, (h_1, σ_1) Our Model, (h_1, σ_1) ROF, (h_2, σ_2) Our Model, (h_2, σ_2)
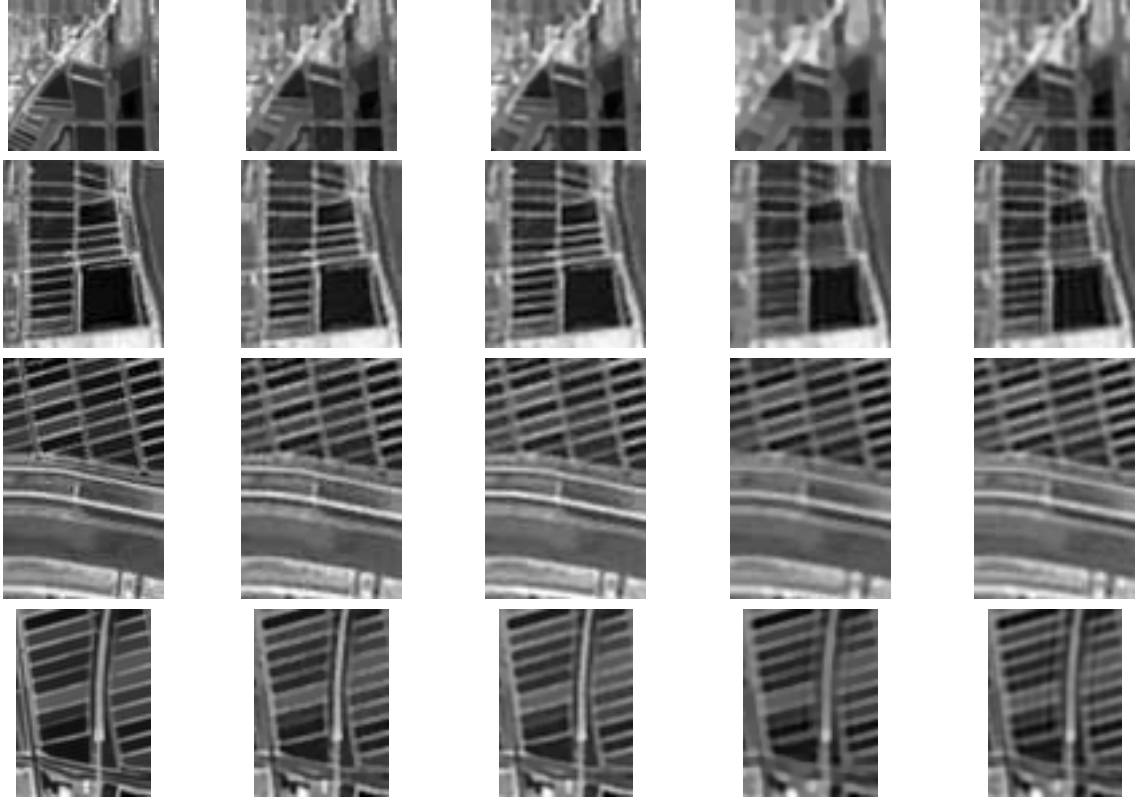
Figure 4.9. *For visual comparisons, we added the method used for recovery and the information of blur and noise at the bottom of each column.*

Then, taking $u_0 = f$, (1.2) is solved. As explained, larger p and larger δ values tend to shrink the backward regime, thus giving way to diffusion.

This is confirmed by numerical experiments. Values of p close to 2 are too strong to preserve edge sharpness while denoising, even for a short amount of time. Hence, preference is given to p values close to 1. As for the other parameters, $\delta = 0.2$ and $dt = 0.007$ are used throughout all those experiments unless otherwise stated. In addition, since solving (1.2) for a long period of time delivers a stationary solution, it is hoped that a significant noise removal effect can be obtained by solving (1.2) for a short amount of time before sharp edges start getting dull. This is exactly what is observed.

Figure 4.11 shows how the quality of the recovered images is affected by different choices of p.

As can easily be expected, numerical experiments (Figure 4.11) do confirm that larger p values require smaller time steps for decent recovery, but still do not preserve edges as well as smaller p values. This is apparent in Figure 4.12.

Reference image II   ROF, (h_1, σ_1) Our Model, (h_1, σ_1) ROF, (h_2, σ_2) Our Model, (h_2, σ_2)

Figure 4.10. *For visual comparisons, we added the method used for recovery and the information of blur and noise at the bottom of each column.*

Figure 4.11. *Noise level is $\sigma = 10$, $\delta = 0.2$, and $dt = 0.005$.* **From left to right** *: Noisy image, recovered images with $p = 1.1$ and $p = 1.3$ after 20 iterations.*

The image obtained with $p = 1.05$ is clearly better than the one obtained with $p = 2$ in that edges are sharper, i.e. better preserved. There does not, however, appear to be monotonicity in p in this sense as smaller p values prove unable to remove noise as compared to larger p values. For this reason a good choice appears to be $p = 1.1$, a value which is used throughout denoising experiments.

Figure 4.12. *Noise level is $\sigma = 10$. Best denoising results with $p = 1.05$ and $p = 2$, respectively.*

Next the noise removal capabilities of the proposed model are tested on images corrupted with noise of varying intensity. In Figure 4.13, the results of experiments with $\sigma = 10, 15, 20, 25, 35, 50$ are depicted which were obtained after 10, 15, 25, 30, 45, 55 iterations, respectively. Computational time for every 10 iterations was approximately 3.6s. In order to better characterize the enhancement process, PSNR values are shown in Table 4.4. It is recalled that

$$\text{PSNR}(f_0, f_1) = 20 \log_{10}\left(\frac{255}{\sqrt{MSE(f_0, f_1)}}\right).$$

Noise level	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$	$\sigma = 35$	$\sigma = 50$
PSNR (noisy)	28.14	24.66	22.12	20.16	17.25	14.15
PSNR (recovered)	33.70	31.79	30.01	29.09	27.44	26.15

Table 4.4
PSNR values for the noisy and recovered images corresponding to the experiments shown in Figure 4.13.

Denoising experiments with various kinds of images follow below. These experiments highlight how well the proposed model preserves sharp edges while removing noise. For comparison purposes, images recovered with a state-of-the-art denoising algorithm, BM3D [4] are also shown in Figure 4.14 and Figure 4.15. Let us emphasize again that BM3D is superior to the proposed model for pure image denoising by nature, but it is not applicable for recovery from noisy blurred images.

4.3. Blurry case. Gray-scale images are considered along with 6 different types of blurring kernels described below.

While it is assumed that $p \in (1, 2]$, the case $p = 1$ was also tested for comparison purposes. It obviously corresponds to the use of total variation as a regularizer for Perona-Malik. In Figure 4.16, a blurring kernel of Type I was used with $\delta = 0.002$ and $dt = 0.06$.

It is expected that larger p values would lead to more diffusion. This is indeed confirmed by the experiment. Table 4.6 contains the PSNR values computed every 50 iterations. This gives some insight into the recovery process for the chosen values of p, i.e. for $p = 1, 1.01, 1.1, 1.3, 1.5, 1.7, 1.9, 2$.

Figure 4.13. $\delta = 0.2$, $p = 1.1$, $dt = 0.007$; **1st row, from left to right**: *Noisy images with $\sigma = 10, 15, 20$,* **3rd row, from left to right**: *Noisy images with $\sigma = 25, 35, 50$,* **2nd and 4th rows**: *Images recovered from the corresponding noisy images right above.*

In the experiment, the nth iteration consists in solving (4.3) for one time step with $\delta_n = \frac{\delta}{n}$ in place of δ. It does not seem to be possible to achieve the same deblurring effect by solving (4.3) alone with fixed δ because of the strong attraction of (1.2) towards constant stationary solutions. In other words, diffusion seems to prevail even if it is quite small. Solving (4.3) using

Figure 4.14. $\delta = 0.2, p = 1.1, dt = 0.007$; **top row, from left to right**: *Noisy images with $\sigma = 10, 15, 25$, **middle row**: Images recovered from the noisy images right above, **bottom row**: Images recovered with a state-of-the-art denoising algorithm, BM3D [4], for comparison.*

a smaller δ in the nth iteration, makes it possible to preserve the edge sharpening property of the Perona-Malik equation. This is true in spite of the fact that the proposed model is well-posed.

Experiments confirm that any value of p close to 1 will be suitable for image deblurring. Values of p larger than 1.5 lead to stronger diffusion and smaller δ values need to be used in this case in order to compensate for it. Loosely speaking, p and δ are inversely proportional. It should be observed, however, that much smaller δ values do not consistently work well all the time.

In Figure 4.17, a closer look is taken at the case $p = 1.01$. The output image after 10 iterations is already a good approximation to the original image and one can see that additional iterations add more texture to the output images.

Next other types of blurring kernels are considered which are found in the list given earlier. A value $p \sim 1$ is fixed for the reason given before. In all experiments, the parameters $\delta = 0.002$
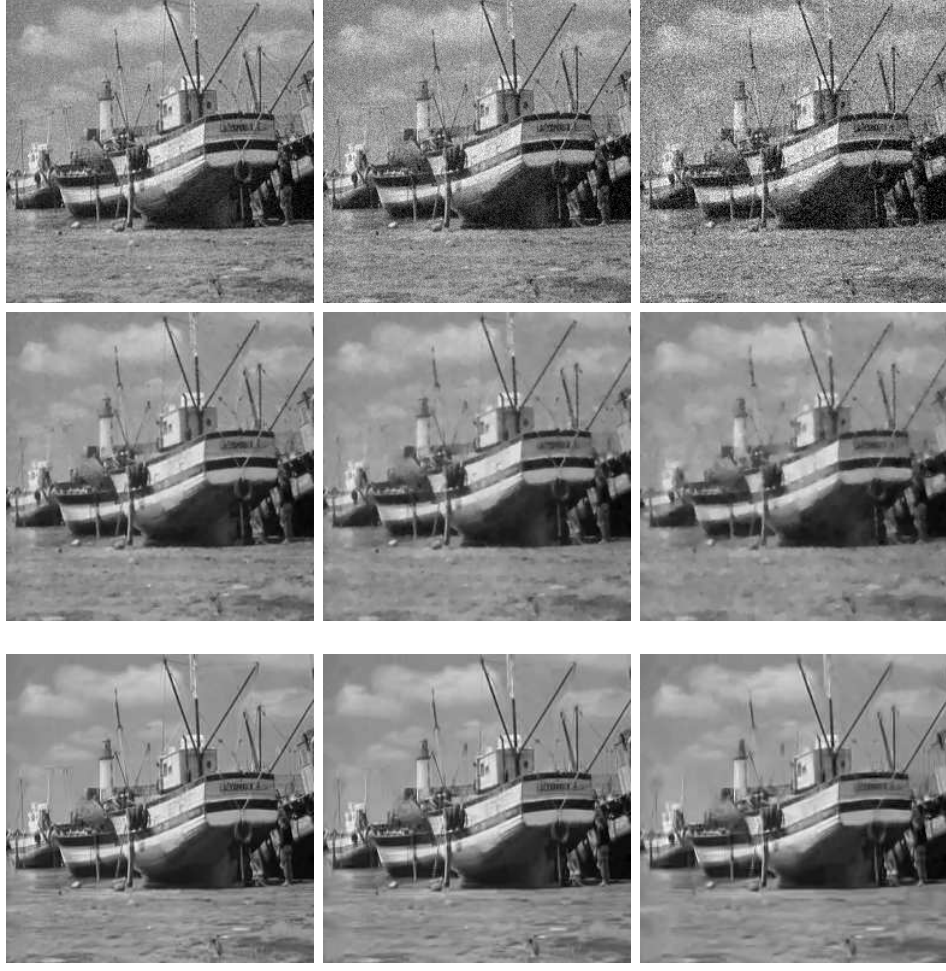
Figure 4.15. $\delta = 0.2$, $p = 1.1$, $dt = 0.005$; **top row, left to right**: *Noisy images with $\sigma = 10, 15, 25$,* **middle row***: Recovered images corresponding to the noisy ones right above,* **bottom row** *: Images recovered with a state-of-the-art denoising algorithm, BM3D [4], for comparison.*

and $dt = 0.06$ are fixed, whereas $\delta_n = \frac{\delta}{n}$ is used in the nth iteration as before. Figure 4.18 shows experiments performed on images blurred with type II kernels, while Figure 4.19 depicts experiments of blur type IV and Figure 4.20 experiments of blur type V. In all these experiments, edges of the blurry images do get sharpened almost instantaneously (within the first few iterations), while texture is recovered subsequently along with still sharper edges.

Furthermore, comparing the output boat images from Figure 4.18, Figure 4.19, and Figure 4.20, it is noticed that there appear more artifacts in case of blur type II. It is checked numerically in Figure 4.21 that size and smoothness of blur are the determining factor for recovery quality. This is to say that less artifacts are observed in smoother blur cases. While edges of the processed boat image (blur type II case) do look sharper, more artifacts can be found in the image than in the other two processed boat images. As for the second woman image, the recovered images (blur type V and VI) look sharper and contain significantly less artifacts than the one with blur type II.
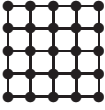
3 × 3 average kernel Type I	5 × 5 average kernel Type II	3 × 3 out-of-focus blur Type III
5 × 5 out-of-focus blur Type IV	fspecial('gaussian',7,1) is used in Matlab 7 × 7 Gaussian kernel Type V	fspecial('gaussian',15,1.5) is used in Matlab 15 × 15 Gaussian kernel Type VI

Table 4.5
The 6 different types of blurring kernels used in the experiments.

Figure 4.16. 3×3 *average kernel, blur type I;* **Left to Right and Top to Bottom**: *Blurry image, Recovered image with* $p = 2$, $p = 1.9$, $p = 1.01$, $p = 1$, *Original image.*

Figure 4.22 depicts the results of a blind deconvolution. This is to determine how much the edge sharpening property of the model helps reconstructing a better image in the absence of any knowledge concerning the type of blur. Two images were used which were taken at night with an ordinary digital camera. A simple guess for the blur present in the blurry images without estimating the actual blur might be an out-of-focus blur or an average blur. As one can notice, the blurry images do become sharper with the help of such a simple guess on the

Iterations	$p = 1$	$p = 1.01$	$p = 1.1$	$p = 1.3$	$p = 1.5$	$p = 1.7$	$p = 1.9$	$p = 2$
50	34.75	34.72	34.65	34.48	33.45	30.99	27.24	25.96
100	36.00	35.97	35.94	35.70	34.95	32.85	28.75	26.98
150	36.67	36.65	36.60	36.39	35.70	33.81	29.77	27.74
200	37.14	37.15	37.06	36.88	36.22	34.46	30.52	28.35
250	37.47	37.49	37.41	37.22	36.60	34.92	31.11	28.86
300	37.73	37.76	37.70	37.50	36.89	35.28	31.58	29.30

Table 4.6

PSNR values corresponding to the enhancement of the blurry image from Figure 4.16 with a PSNR of 26.49 for the blurry image. Computational time for 50 iterations is approximately 17s.

Figure 4.17. *A closer look at the case $p = 1.01$ in Table 4.6.* **First row, from left to right** *: Blurry image, output images after 10 and 20 iterations (PSNR values : 26.49, 31.20, 32.91);* **second row, from left to right***: output image after 30, 40, and 50 iterations (PSNR values : 33.76, 34.30, 34.72). Computational time for 10 iterations amount to approximately 3.6s.*

blur and texture is recovered.

As a final remark it is noted that numerical experiments indicate that better recovery is achieved for blurring kernels with rounder support. The stability properties of the chosen discretization of (1.2) seem to deteriorate if the extra term in (4.3) contains a blurring kernel the support of which significantly deviates from radial symmetry. Figure 4.23 with blurs of motion type is evidence for this. Recovery indeed improves as soon as horizontal motion blur becomes non-horizontal one. Table 4.7 shows the three blurring kernels used in Fig 4.23. However, we cannot determine whether this is due to the model or is a short-coming of the chosen implementation. It seems that this is not only a defect of our proposed model, though,
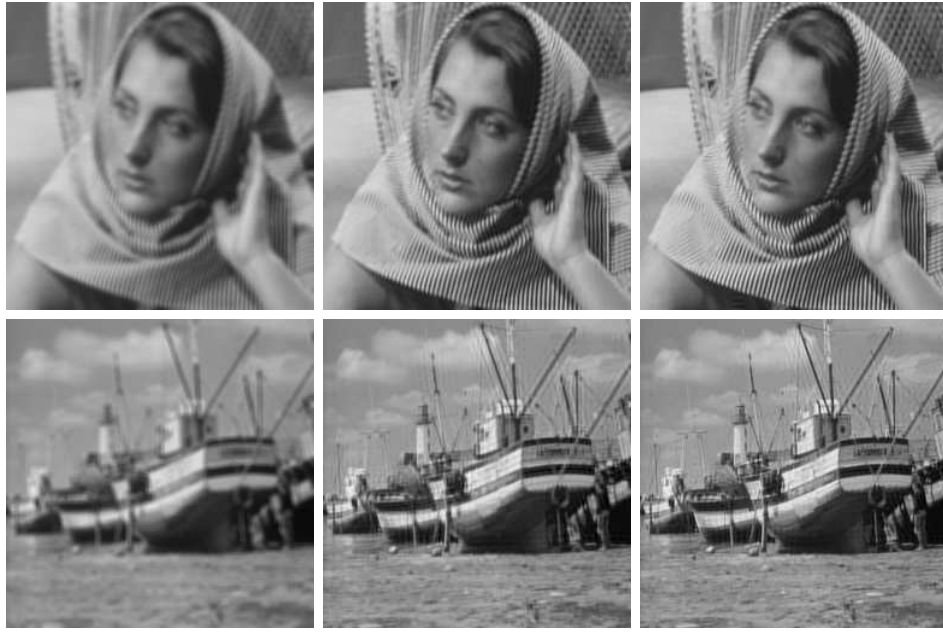
Figure 4.18. 5×5 *average kernel of blur type II.* **Left***: Blurry images, PSNR=22.86(woman) and 24.60 (boat);* **middle***: output images, PSNR=30.03 (woman) and 29.88 (boat) after* 50 *iterations;* **right***:output images, PSNR=32.23 (woman) and 32.49 (boat) after* 300 *iterations.*

Figure 4.19. 5×5 *Out-of-focus blur of type IV.* **Left***: Blurry image, PSNR = 25.01 (woman) and 26.98 (boat);* **middle***: output, PSNR=30.14 (woman)and 32.28 (boat) after* 50 *iterations;* **right***: output PSNR=33.69 (woman) and 35.21 (boat) after* 300 *iterations.*
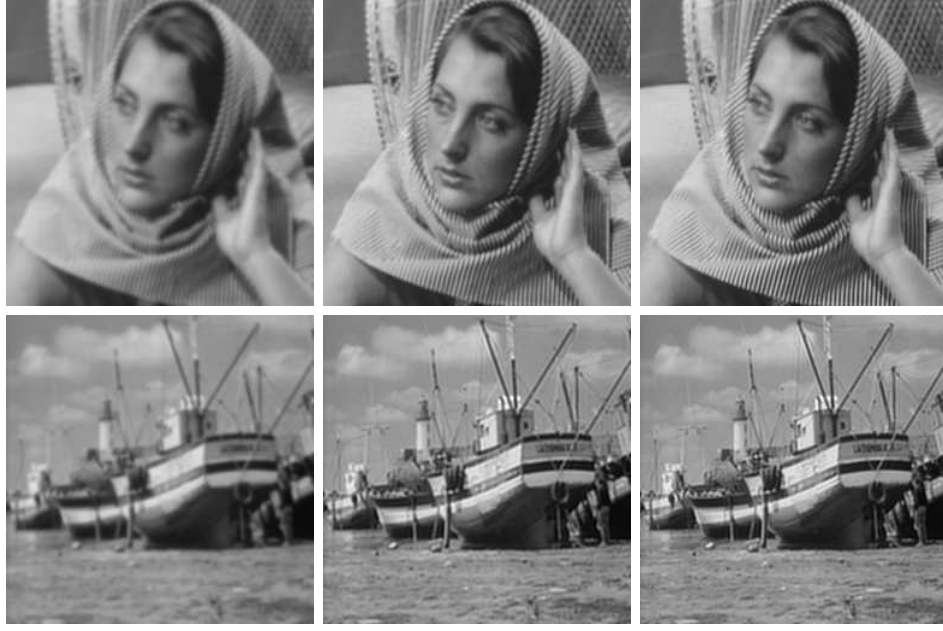
Figure 4.20. 7×7 *Gaussian blur of type V.* **Left***:output PSNR=24.75 (woman)and 26.35 (boat);* **middle***: output, PSNR = 27.68 (woman) and 30.29 (boat) after* 50 *iterations;* **right***: output, PSNR=30.96 (woman) and 31.54 (boat) after* 300 *iterations.*

Figure 4.21. *Comparison between the three blur types, II and V and VI after* 300 *iterations.* **Left (type II)***: PSNR=32.49 (boat) and 31.98 (woman);* **middle (type V)***: PSNR=31.54 (boat) and 32.60 (woman);* **right (type VI)***: PSNR=29.85 (boat) and 31.05 (woman).*

Figure 4.22. *Blind deconvolution of night scenes, size reduced to* 512×512. *Recovered with a simple guess of a radially symmetric blurring kernel.*

since it is also observed in the ROF model, which is shown in the last row of Figure 4.23, where we applied the gradient descent method to compute the images with parameters $\lambda = 3, \Delta t = 0.001$ fixed. All the six images in the last two rows were obtained by stopping at proper iterations before the images get further deteriorated. Nevertheless, the ROF model is not affected as much as our proposed model is by the intermediate blur types A and B in Table 4.7. Especially, the ROF model was stable in the case of the blurring kernel B with much bigger $\Delta t = 0.1$.

5. Conclusion. A new class of anisotropic forward-backward diffusion equations of Perona-Malik type is proposed and a comprehensive set of experiments are designed and run to prove the usefulness of the model. The main feature of the model is that it exhibits both smoothing and/or sharpening depending on the modulus of the solution gradient. In other words, the
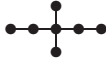
1×5 horizontal blur	3×5 blurring kernel A	3×5 blurring kernel B

Table 4.7

Three blurring kernels corresponding to the experiments in Fig 4.23.

Figure 4.23. *Recovery results.* **Top row** *: Blurry images,* **Middle row** *: Recovered images by our proposed model,* **Bottom row** *: Recovered images by the ROF model.* **Left column** *- 1×5 horizontal motion blur,* **Middle column** *- 3×5 blurring kernel A in Table 4.7,* **Right column** *- 3×5 blurring kernel B in Table 4.7.*

model delivers smoothing in regions of small or large gradient modulus, whereas it delivers sharpening in an intermediate region. This leads to good recovery from noisy blurred images. Unlike the Perona-Malik equation, the model is well-posed and produces good quality images from noisy blurred images faster as compared to other state-of-the-art recovery algorithms. A

noticeable advantage of the model is its ability to allow for control over the desired smoothing and sharpening effect. The parameter δ, in particular, controls the size of regions where smoothing or sharpening takes place, respectively, whereas p controls the overall regularity of the solution and values $p \sim 1$ seem to lead to optimal performance.

REFERENCES

[1] P. GUIDOTTI, *A bakward-forward regularization of the Perona-Malik equation*, J. Differential Equations, 252 (2012), no. 4, 3226–3244.

[2] V. CASELLES, A. CHAMBOLLE, AND M. NOVAGA, *The discontinuity sets of solutions of the TV denoising problem and some extensions*, Multiscale Model. Simul., 6(3), (2007), 879-894.

[3] S. DEMOULINI, *Young Measure Solutions for a Nonlinear Parabolic Equation of Forward-Backward Type*, SIAM J. math. Anal., 27 (1996) 376-403.

[4] K. DABOV, A. FOI, AND K. EGIAZARIAN, *Image Restoration by Sparse 3D Transform-Domain Collaborative Filtering*, Proc. SPIE Electronic Imaging 2008, no. 6812-07, San Jose, CA, Jan. 2008

[5] G. DAHLQUIST, S. C. EISENSTAT, AND G. H. GOLUB, *Bounds for the error of linear systems of equations using the theory of moments*, J. MATH. ANAL. APPL., 37 (1972), P. 151-166.

[6] B. DONG, Z. SHEN, *MRA-Based Wavelet Frames and Applications*, LECTURE NOTE PRESENTED AT PCMI 2010, PARK CITY, UTAH

[7] S. DURAND, F. MALGOUYRES, AND B. ROUGÉ, *Image de-blurring, spectrum interpolation and application to satellite imaging*, CONTROL OPTIMISATION AND CALCULUS OF VARIATION, 5 (2000), P. 445-477.

[8] G. H. GOLUB, *Bounds for matrix moments*, ROCKY MOUNTAIN J. MATH., 4 (1974), P. 207-211.

[9] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Review, 15 (1973), p. 318-334.

[10] G. H. GOLUB, G. MEURANT, *Matrices, Moments and Quadrature*, Proceedings of the 15th Dundee Conference, June-July 1993, D. F. Griffiths, G. A. Watson, (eds.), Longman Scientific & Technical (1994)

[11] G. H. GOLUB, R. UNDERWOOD, *The block Lanczos method for computing eigenvalues*, Mathematical Software III, J. Rice Ed., (1977) 361-377.

[12] G. H. GOLUB, J. WELSCH, *Calculation of Gauss Quadrature Rules*, Math. Comp., 23 (1969) 221-230.

[13] M. HOCHBRUCK, C. LUBICH, *On Krylov Subspace Approximations to the Matrix Exponential Operator*, SIAM J. Numer. Anal., 34 (1996) 1911-1925.

[14] M. HOCHBRUCK, C. LUBICH, AND H. SELHOFER, *Exponential Integrators for Large Systems of Differential Equations*. SIAM J. Sci. Comput., 19 (1998) 1552-1574.

[15] Y. KIM, J. B. GARNETT, AND L. VESE, *A Convex Minimization Model in Image Rrestoration via one-dimensional Sobolev Norm Profiles*, IEEE ICIP 2011.

[16] J. V. LAMBERS, *Derivation of High-Order Spectral Methods for Time-dependent PDE using Modified Moments*, Electronic Transactions on Numerical Analysis, 28 (2008) 114-135.

[17] J. V. LAMBERS, *Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration*, Electronic Transactions on Numerical Analysis, 31 (2008) 86-109.

[18] J. V. LAMBERS, *Krylov Subspace Methods for Variable-Coefficient Initial-Boundary Value Problems*, Ph.D. Thesis, Stanford University SCCM Program, 2003.

[19] J. V. LAMBERS, *Krylov Subspace Spectral Methods for the Time-Dependent Schrödinger Equation with Non-Smooth Potentials*, Numerical Algorithms, 5 (2009) 239-280.

[20] J. V. LAMBERS, *Krylov subspace spectral methods for variable-coefficient initial-boundary value problems*, Electronic Transactions on Numerical Analysis, 20 (2005), 212-234.

[21] J. V. LAMBERS, *Practical Implementation of Krylov Subspace Spectral Methods*, Journal of Scientific Computing, 32 (2007) 449-476.

[22] Y. LOU, A. L. BERTOZZI, AND S. SOATTO, *Direct Sparse Deblurring*, J. Math. Imaging Vis. 39 (2011) 1 - 12.

[23] I. MORET, AND P. NOVATI, *RD-rational approximation of the matrix exponential operator*, BIT, 44 (2004) 595-615.

[24] R. NEELAMANI, H. CHOI, AND R. G. BARANIUK, *ForWaRD : Fourier-wavelet regularized deconvolution for ill-conditioned systems*, IEEE Trans. Signal Process. 52 (2004) 418-433.

[25] C. C. PAIGE, AND M. A. SAUNDERS, *Solution of Sparse Indefinite Systems of Linear Equations*, SIAM J. Numer. Anal., 12 (1975), p. 617-629.

[26] L. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992) 259-268.

[27] H. TAKEDA, S. FARSIU, AND P. MILANFAR, *Deblurring using regularized locally adaptive kernel regression*, IEEE Tans. Image Process. 17 (2008) 550-563.

[28] Tokman, M.: Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. *J. Comput. Phys.* **213** (2006) 748-776.

[29] G. BELLETTINI AND G. FUSCO AND N. GUGLIELMI, *A concept of solution and numerical experiments for forward-backward diffusion equations*, Discrete Cont. Dyn. Syst. vol. 16, no. 4 (2006) 783-842.

[30] M. COLOMBO AND M. GOBBINO, *Passing to the limit in maximal slope curves: from a regularized Perona-Malik equation to the total variation flow*, Math. Models Methods Appl. Sci. vol. 22, no. 8 (2012) 19pp.

[31] T. HEBERT AND R. LEAHY, *A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson dara using Gibbs priors*, IEEE Trans. Med. Imag. vol. 8, no. 2 (1989) 194–202.