Recent Advances in Image Analysis and Computer Vision

A Four-Pixel Scheme for Singular Differential Equations

Sergej Kosov

Contents

1. Introduction

1.1 Motivation

Singular diffusion equations such as total variation and balanced forward-backward diffusion are appealing: they have a finite extinction time, and experiments show that piecewise constant structures evolve. Unfortunately, their implementation is awkward. The goal is to introduce a novel class of numerical methods for these equations in the 2D case. They should be simple to implement, absolutely stable and do not require any regularization in order to make the diffusivity bounded. The presented schemes are based on analytical solutions for 2 x 2-pixel images which are combined by means of an additive operator splitting. It has been shown that they may also be regarded as iterated 2D Haar wavelet shrinkage. Experiments demonstrate the favorable performance of the numerical algorithm.

1.2 Diffusion process basics

Diffusion is the physical term: *diffusion* is the net action of matter (particles or molecules), heat, momentum, or light whose end is to minimize a concentration gradient. Diffusion process is characterized by two standings. The first one is that the diffusion process always preserves mass of matter [4]. And the second is that diffusion process equlibrates differences of matter concentration. These standings are easy to describe with two formulas:

1. *Fick's law* describe the equilibration of concentration differences: $j = -g \cdot \nabla u$
   Concentration gradient ∇u creates flux j , and g is a diffusion tensor.
2. *Continuity equation* describes conservation of mass: $\partial_t u = div(g \cdot \nabla u)$

In such a way, we have derived the diffusion equation:

$$\partial_t u = div(g \cdot \nabla u)$$

In a case of linear diffusion (when we do not consider local strucrure of matter , i.e. the diffusion tensor is equivalent to identity) we obtain the simpliest diffusion process: $\partial_t u = \Delta u$

To understand how it works, imagine a cap of milk. You can shake this cap somehow to create wawes on the milk surface. But as soon as you put this cup on a table, the milk will start settling down and in some seconds you will have no waves anymore:

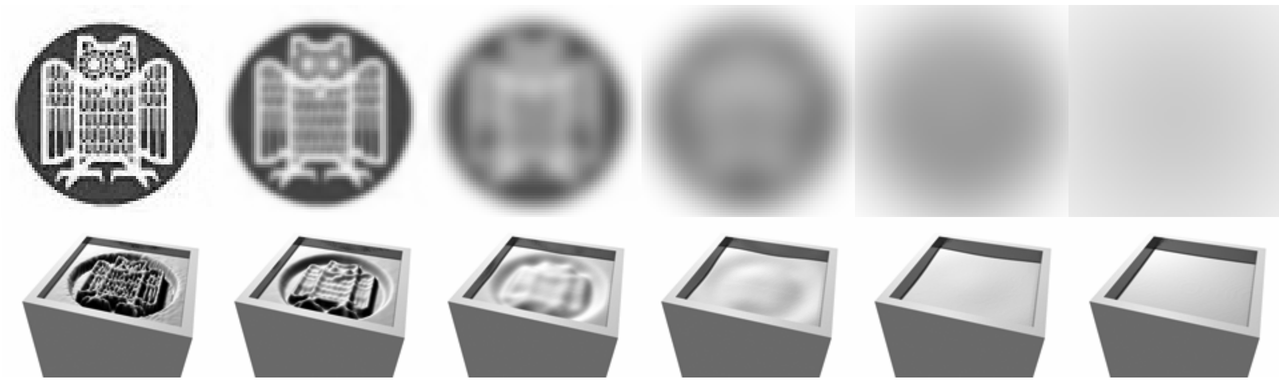


Fig. 1. UdS logo. Linear diffusion example.

You still have the same amount of milk (mass preservation), but the difference between any two drops of milk at the surface is minimal – you have flat surface (equilibrium of concentration differences). In image processing, diffusion is very useful for image enhacement. For example – for denoising an image. When we want to get rid of high frequences (noise or some small details) but preserve edges from blurring. It is pity, but it is almost impossible using linear diffusion.

So, let us concider nonlinear diffusion. This process avoids delocatisation and blurring of edges. It is described with the following equation:

$$\partial_t u = div(g(|\nabla u|)\cdot\nabla u) \quad on \quad \Omega\times(0,\infty)$$

with stable initial and boundary conditions:

$$u(x,0) = f(x) \quad on \quad \Omega$$
$$\partial_n u = 0 \quad on \quad \partial\Omega\times(0,\infty)$$

Function $g()$ takes as an argument a *fuzzy edge detector* $|\nabla u|$ and should be chosen as a decreasing nonnegative function. It means that on the edges, where image derivatives are high, low values of function $g()$ will embarrass diffusion. Now let us suppose that we want to get rid of noise (or high frequencies) with help of diffusion process. And let us, with help of the following illustration, compare the best result of eliminating noise with linear diffusion process and, nonlinear diffusion process:

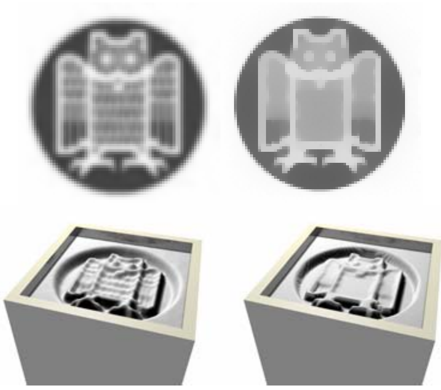


Fig. 2. UdS logo. **Left:** Linear diffusion. **Right:** Nonlinear diffusion.

We can clearly see that in the right column, our owl has no plumage anymore but the edges are very good preserved.

Singular diffusion filters lead to a piecewise constant images. As function $g()$, diffusivity, we use singular diffusivities. As a prototype for a class of singular diffusivities we consider the family:

$$g(|\nabla u|) = \frac{1}{|\nabla u|^p}; \quad p \geq 1.$$

In this paper we shall consider two singular diffusivities. Total variation (TV) diffusivity, that has interesting properties such as finite extinction time and shape-preserving qualities – the case when $p = 1$: $g(|\nabla u|) = \frac{1}{|\nabla u|}$. For $p > 1$ the diffusion not only preserves edges but even enhances them. Balanced forward-backward (BFB) diffusivity – the case when $p = 2$: $g(|\nabla u|) = \frac{1}{|\nabla u|^2}$. The most problems with singular diffusivities are that it is possible to have very small values of $|\nabla u|$ and in such cases our function $g()$ becomes unbounded. That implies numerical instability, and failure of solver. As a result, iterative numerical schemes may reveal slow convergence, and in general numerical errors can be amplified. In order to eliminate all these problems, it is common to regularize the diffusivity function by replacing it by the bounded diffusivity:

$$g(|\nabla u|)=\frac{1}{\left(|\nabla u|^2+\varepsilon^2\right)^{p/2}};\quad p\geq 1.$$

In this case, however, one observes that blurring artifacts are introduced and some of the nice theoretical properties of singular nonlinear diffusion filters do no longer hold. The goal of the present paper is to address these problems by introducing a novel class of numerical schemes for singular diffusion equations.

2. Scheme for 2 x 2 pixel images

2.1 Nonlinear diffusion

Let us consider diffusion equations on 4-pixel image with periodic boundary condition:

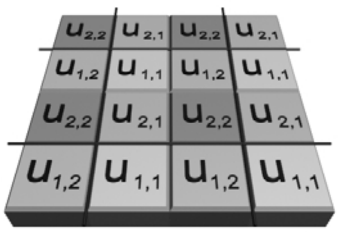


Fig. 3. Sketch of the 4-pixel image with periodic boundary condition.

with initial image data $f=\left(f_{i,j}\right)_{i,j=1}^{2}$

Now, we would like to build a numerical approximation of diffusivity in the middle of this picture. Let the grid size in both directions will be $h=1$; and $D(u)$ denotes the discretization of $|\nabla u|=\left|\begin{pmatrix}\partial_x u, & \partial_y u\end{pmatrix}^T\right|$. Than the approximation of midpoint diffusivity can be written very simple: $g:=g(D(u))$. To build the approximation of $D(u)$ we shall use all the 4 pixels and the following directions: $\bar{x}=(1,0)^T$, $\bar{y}=(0,1)^T$, $\bar{\xi}=\frac{1}{\sqrt{2}}(1,1)^T$ and $\bar{\eta}=\frac{1}{\sqrt{2}}(1,-1)^T$:

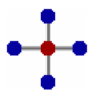
$$D(u):=\left(\frac{1}{2}\left(\frac{1}{2}\left(u_{1,1}-u_{1,2}\right)^2+\frac{1}{2}\left(u_{2,1}-u_{2,2}\right)^2\right)+\frac{1}{2}\left(\frac{1}{2}\left(u_{1,1}-u_{2,1}\right)^2+\frac{1}{2}\left(u_{1,2}-u_{2,2}\right)^2\right)+\right.$$

$$\left.\frac{1}{2}\left(\left(\frac{u_{1,1}-u_{2,2}}{\sqrt{2}}\right)^2+\left(\frac{u_{1,2}-u_{2,1}}{\sqrt{2}}\right)^2\right)\right)^{1/2}=$$

$$\frac{1}{2}\left(\left(u_{1,1}-u_{1,2}\right)^2+\left(u_{2,1}-u_{2,2}\right)^2+\left(u_{1,1}-u_{2,1}\right)^2+\left(u_{1,2}-u_{2,2}\right)^2+\left(u_{1,1}-u_{2,2}\right)^2+\left(u_{1,2}-u_{2,1}\right)^2\right)^{1/2}$$

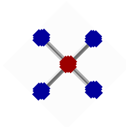
Now let us build two types of the following step of discretization. We shall use the notation $\frac{\partial u}{\partial t}=\dot{u}$. So, the first type is the usual discretization with respect to the directions $\bar{x}=(1,0)^T$ and $\bar{y}=(0,1)^T$:

$$\left(\dot{u}_{i,j}\right)_{xy}=g\cdot\left(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}-4u_{i,j}\right)$$



and the second type with respect to the directions $\bar{\xi} = \frac{1}{\sqrt{2}}(1,1)^T$ and $\bar{\eta} = \frac{1}{\sqrt{2}}(1,-1)^T$:

$$\left(\dot{u}_{i,j}\right)_{\xi\eta} = g \cdot \frac{1}{2}(u_{i+1,j+1} + u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} - 4u_{i,j})$$



Combining these two types of discretizations with coefficient $\alpha \in [0,1]$ we get:

$$\dot{u}_{i,j} = \alpha\left(\dot{u}_{i,j}\right)_{xy} + (1-\alpha)\left(\dot{u}_{i,j}\right)_{\varepsilon\eta}$$

And the following system of equations:

$$\begin{aligned}
\dot{u}_{1,1} &= 2g \cdot (-(1+\alpha)u_{1,1} + \alpha u_{1,2} + \alpha u_{2,1} + (1-\alpha)u_{2,2}),\\
\dot{u}_{1,2} &= 2g \cdot (\alpha u_{1,1} - (1+\alpha)u_{1,2} + (1-\alpha)u_{2,1} + \alpha u_{2,2}),\\
\dot{u}_{2,1} &= 2g \cdot (\alpha u_{1,1} + (1-\alpha)u_{1,2} - (1+\alpha)u_{2,1} + \alpha u_{2,2}),\\
\dot{u}_{2,2} &= 2g \cdot ((1-\alpha)u_{1,1} + \alpha u_{1,2} + \alpha u_{2,1} - (1+\alpha)u_{2,2})
\end{aligned}$$

with initial conditions $u_{i,j}(0) = f_{i,j}, \quad i,j = 1,2$. From $\dot{u}_{1,1} + \dot{u}_{1,2} + \dot{u}_{2,1} + \dot{u}_{2,2} = 0$ we see that the average grey value $\mu := \frac{1}{4}(f_{1,1} + f_{1,2} + f_{2,1} + f_{2,2})$ is preserved during the diffusion process.

We are mainly interested in case $\alpha = \frac{1}{2}$, where the above system of equations further simplifies to

$$\dot{u}_{i,j} = 4g \cdot (\mu - u_{i,j}), \quad i,j = 1,2,$$

which is a dynamical system with discontinuous right hand side. It is not difficult to verify that this system possesses the unique analytical solution:

$$u_{i,j}(t) = \begin{cases} \mu + (1 - 4p(D(f))^{-p}t)^{1/p}(f_{i,j} - \mu), & 0 \le t < (D(f))^p/(4p) \\ \mu, & t \ge (D(f))^p/(4p) \end{cases}$$

For total variation diffusion we have:

$$u_{i,j}(t) = \begin{cases} \mu + (1 - 4t/D(f))(f_{i,j} - \mu), & 0 \le t < D(f)/4 \\ \mu, & t \ge D(f)/4 \end{cases}$$

For balanced forward – backward diffusion we have:

$$u_{i,j}(t) = \begin{cases} \mu + \sqrt{1 - 8t/(D(f))^2}(f_{i,j} - \mu), & 0 \le t < D(f)^2/8 \\ \mu, & t \ge D(f)^2/8 \end{cases}$$

2.2 Outlook

We derived a simple scheme for 2x2 pixel images, which is very easy for numerical implementation. This scheme does not require regularization. And it is clearly seen that if $t \to \infty$ we get linear diffusion process. This happens in regions where the gradient is already close to zero. Also the same solutions we can be achieved with Haar wavelet shrinkage.

Now let us proceed with N x M pixel images.

3. Extension for N x M pixel images

3.1 Naïve numerical scheme

Now let us consider images with arbitrary size and reflecting boundary conditions. To solve the diffusion equation $\partial_t u = div(g \cdot \nabla u)$ with diffusivity $g(|\nabla u|) = \frac{1}{|\nabla u|^p}$ we shall discretize this equation in space with respect to $\overline{x}$, $\overline{y}$ and $\overline{\xi}$, $\overline{\eta}$ directions, than combine both discretizations via weighted coefficient α. Now we chose $\alpha = \frac{1}{2}$ and apply the discretization via an explicit Euler scheme. We can write the time derivative as $\dot{u}_{i,j} = \frac{u_{i,j}^{k+1} - u_{i,j}^k}{\tau}$. Moreover, let us introduce the following notations for average grey value:

$$\mu_{i,j,++}^k := \frac{1}{4}\left(u_{i+1,j}^k + u_{i,j+1}^k + u_{i+1,j+1}^k + u_{i,j}^k\right),$$
$$\mu_{i,j,+-}^k := \frac{1}{4}\left(u_{i+1,j}^k + u_{i,j-1}^k + u_{i+1,j-1}^k + u_{i,j}^k\right),$$
$$\mu_{i,j,-+}^k := \frac{1}{4}\left(u_{i-1,j}^k + u_{i,j+1}^k + u_{i-1,j+1}^k + u_{i,j}^k\right),$$
$$\mu_{i,j,--}^k := \frac{1}{4}\left(u_{i-1,j}^k + u_{i,j-1}^k + u_{i-1,j-1}^k + u_{i,j}^k\right).$$

Using these notations, we can now write down the naïve scheme for solving diffusion equation:

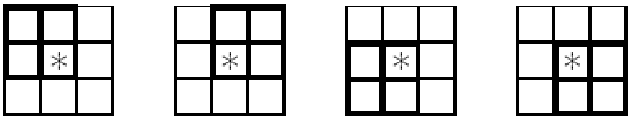
$$u_{i,j}^{k+1} = u_{i,j}^k + \tau g_{i+\frac{1}{2},j+\frac{1}{2}} \cdot \left(\mu_{i,j,++}^k - u_{i,j}^k\right) + \tau g_{i+\frac{1}{2},j-\frac{1}{2}} \cdot \left(\mu_{i,j,+-}^k - u_{i,j}^k\right) +$$
$$+ \tau g_{i-\frac{1}{2},j+\frac{1}{2}} \cdot \left(\mu_{i,j,-+}^k - u_{i,j}^k\right) + \tau g_{i-\frac{1}{2},j-\frac{1}{2}} \cdot \left(\mu_{i,j,--}^k - u_{i,j}^k\right).$$

Here τ denotes the time step size and $u_{i,j}^k$ the approximate solution at pixel (i, j) and time $k\tau$. Unfortunately, due to the singularity of g at zero, this scheme becomes unstable with respect to the maximum-minimum principle for arbitrary small time steps if neighboring pixel values become arbitrary close. We use therefore a different approximation.

3.2 The “four – pixel” scheme

For each pixel (*) with coordinates (i, j):

- Consider the four cells



(−−) (+−) (−+) (++)

- Compute the analytical solutions
  This gives four approximations

$$u_{i,j,++}^{k+1},\ u_{i,j,+-}^{k+1},\ u_{i,j,-+}^{k+1},\ u_{i,j,--}^{k+1}.$$

- Average:

$$u_{i,j}^{k+1} = \frac{1}{4}\left(u_{i,j,++}^{k+1} + u_{i,j,+-}^{k+1} + u_{i,j,-+}^{k+1} + u_{i,j,--}^{k+1}\right).$$

3.3 Stability and consistency

The values of the analytical solution at arbitrary times are convex combinations of its initial values. The novel scheme therefore satisfies the maximum – minimum principle. Consequently, it is absolutely stable for each τ. The scheme is conditionally consistent. Also we have equivalence to shift and rotation invariant wavelet shrinkage.

4. Results

In Figure 4, we contrast the regularization-free scheme based on the analytical 2x2-pixel solution for TV diffusion with a standard explicit discretization. In this scheme, BFB diffusivity is approximated by the regularized BFB diffusivity $1/\sqrt{|\nabla u|^2 + \varepsilon^2}$. Since the stability condition for explicit schemes imposes to the time step size a bound which is inversely proportional to the upper bound of the diffusivity, a high number of iterations is needed for reasonable ε. It can be seen that the 2x2-pixel scheme and the unregularized BFB diffusivity which cannot be used in the explicit scheme considerably reduce blurring effects caused by the discretization.

Fig. 4. Left: Original image (source: www.fishki.net)**. Middle:** 2 x 2 pixel scheme, $\tau = 4\times10^{-5}$, 135 iterations. **Right:** Balanced forward –backward diffusion with standard explicit scheme, $\varepsilon = 0{,}1$, $\tau = 2{,}5\times10^{-4}$, 150000 iterations. Author: S. Kosov (2006)

Figure 5 and figure 6 demonstrate total variation and balanced forward–backward diffusion accordingly. The elimination of noise with edge preservation (and even enhancement in case of BFB diffusion) is very good seen on bottom couple of pictures. Gradient $|\nabla u|$ is very sensitive to high frequencies, and the bottom right pictures show only contours of the objects in scene. The bottom left images are the results of fuzzy edge detector processor with original image as initial data. There we can observe not only contours but also a lot of high frequencies where spatial derivatives are high. Therefore, new 2 x 2 pixel scheme could be very helpful for pattern recognition and segmentation.

Fig. 5. Top left: Original image (source: www.fishki.net)**. Top right** 2 x 2 pixel scheme for total variation diffusion, $\tau = 5 \times 10^{-4}$, 100 iterations. **Bottom left:** Fuzzy edge detector. Original image. **Bottom left:** Fuzzy edge detector. Filtered image. Author: S. Kosov (2006)

Fig. 6. Top left: Original image**. Top right** 2 x 2 pixel scheme for balanced forward – backward diffusion, $\tau = 4{,}5\times10^{-5}$, 120 iterations. **Bottom left:** Fuzzy edge detector. Original image. **Bottom left:** Fuzzy edge detector. Filtered image. Author: S. Kosov (2006)

5. Summary

Novel numerical schemes for a favorable class of singular nonlinear diffusion equations that includes TV and BFB diffusion have been introduced. These schemes can be distinguished from other schemes by the fact that they do not require to regularize the diffusivities. They are based on analytical solutions for 4-pixel images. These solutions create extremely simple algorithms that are absolutely stable in the maximum norm, conditionally consistent and reveal good rotation invariance. The experiments have shown that they give sharper results at edges than traditional schemes with regularized diffusivities, even for significantly larger time steps. This more pronounced tendency to create piecewise constant images is particularly suited for singular nonlinear PDEs.

References

1. Martin Welk, Joachim Weickert, Gabriele Steidl. A Four-Pixel Scheme for Singular Differential Equations. http://www.mia.uni-saarland.de/publications.shtml
2. V. I. Tsurkov. An analytical model of edge protection under noise suppression by anisotropic diffusion. Journal of Computer and Systems Sciences International, 39(3):437-440, 2000.
3. I. Pollak, A. S. Willsky, and Y. Huang. Nonlinear evolution equations as fast and exact solvers of estimation problems. IEEE Transactions on Signal Processing, 2004.
4. Diffusion process description: http://en.wikipedia.org/wiki/Diffusion
5. Basics of differential equations discretization: http://en.wikipedia.org/wiki/Forward_difference