Numerical Methods for Minimization Problems Constrained to S^1 and S^2

Thomas Cecil*, Stanley Osher†, and Luminita Vese‡

January 21, 2004

Abstract

In this paper we propose numerical methods for minimization problems constrained to S^1 and S^2. By our technique based on the angle formulation, standard numerical difficulties are easily overcome. Applications to computations of harmonic maps, denoising of directional data and of color images are presented, in two and three dimensions.

1 Introduction

When solving constrained minimization problems using partial differential equations one encounters the problem of finding a solution that minimizes some defined energy, E, and also satisfies the constraint condition, C. When a numerical solution must be computed then we are often in a predicament where we can only approximately satisfy both C and E. Sometimes numerical schemes can be designed which implicitly enforce C, but these are not always easy to find, and they can restrict us from using the most advanced numerical methods. Other times we can analytically derive energies for variables ϕ that satisfy C exactly, and then we can numerically minimize these energies in terms of ϕ, not having to worry about trading off the accuracy of solving E for the requirement of satisfying C. While this second method is more ideal in that we have one less problem to solve, minimizing the energies involving ϕ may lead to new difficulties.

In this paper we will introduce numerical methods for minimization problems where the constraint, C, is that the solution U lie on either S^1 or S^2 (the unit circle in two dimensions and the unit sphere in three dimensions). These types of problems arise more recently in computer vision where one studies

*Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA. email: tcecil@math.ucla.edu

†Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA. email: sjo@math.ucla.edu

‡Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA. email: lvese@math.ucla.edu

gradient directions, optical flow directions, surface normals, principle directions and colors [18],[13],[5],[11],[16],[19]. In the field of liquid crystals one studies p-harmonic maps with the above constraints [1],[6], and the general topic of harmonic maps between manifolds is well studied [2],[17],[15],[9],[7].

There are several approaches to minimize the energy E under the constraint C. If we compute the Euler-Lagrange equations associated with the minimization of E, as $\partial E/\partial U = 0$, then a gradient descent method can be applied in order to iteratively minimize E. To enforce C one technique is to normalize the solution after each iteration, i.e. given a computed solution U_*^n at time n, we then replace it by $U^n/|U_*^n|$ [8],[18],[1],[1],[12]. Another method leaves this normalization until after convergence has been reached [20]. In either case the theoretical analysis of convergence is not complete. A second option is to minimize a related unconstrained energy

$$E + \frac{1}{\epsilon}\int (1 - |U|^2)^2 dx$$

as $\epsilon \to 0$, which will asymptotically approach the desired solution [3],[4]. Both of these methods are of the first type mentioned in the initial paragraph: approximately solving C while minimizing E.

Here we will approach the solution using the second idea mentioned in the first paragraph, that of finding variables ϕ that solve C exactly, writing E in terms of ϕ, and minimizing E with respect to ϕ. The PDE formulation of the problem constrained to S^1 or S^2 is not new. However, as the parameterizations of S^1 and S^2 involve multivalued and highly singular variables, previous numerical solutions have avoided these parameterizations or tried to resolve them by slightly changing the formulation of the problem, see [13] for example. The techniques presented here aim to handle the numerical difficulties of these parameterizations, allowing $\partial E/\partial \phi$ to be minimized using straightforward finite differences or any other local evolution procedure.

Thus, our method is advantageous because there is no error at all in the constraint condition. The complexity of our method will be shown to be of the same order as straightforward discretizations of the PDEs resulting from taking the Euler-Lagrange equations of E, C, requiring only a fixed number of extra operations per gridpoint. In fact, since we advance the PDEs in parameter space we reduce the number of PDEs to be solved by one.

Related work in the context of image analysis and computer vision is [13], [5], [18], [11], [16], [20], [19], for directional diffusion, color image denoising, and other applications.

2 The S^1 Case

Consider an open and bounded domain $\Omega \in \mathbb{R}^2$ and an energy to be minimized defined by

$$E(U) = \int_\Omega g(U)dx,\ U = (u, v) \in \mathbb{R}^2 \tag{1}$$

along with the constraint $|U| = 1$, where g may contain differential operators. We change to the orientation formulation by letting $U = (\cos\theta, \sin\theta)$ and by the method of gradient descent we derive the associated time-dependent partial differential equation

$$\frac{\partial\theta}{\partial t} = -\frac{\partial E}{\partial\theta}, \tag{2}$$

for an artificial time, t. For example

$$\inf_{U,|U|=1} \int_\Omega |\nabla U|^2 dx = \inf_\theta \int_\Omega |\nabla\theta|^2 dx \tag{3}$$

yields

$$\theta_t = \Delta\theta. \tag{4}$$

We then solve (2) to steady state starting from initial conditions given by

$$\theta_0 = \tan^{-1}\left(\frac{v_0}{u_0}\right), \theta \in [0, 2\pi), \tag{5}$$

using techniques to be described below. We then use $U = (\cos\theta, \sin\theta)$ to recover our final solution. Note that the constraint, $|U| = 1$, is automatically satisfied as we only evolve θ.

2.1 Adaptive Riemann Surface Method

As noted in [13], the numerical solution of (2) using a straightforward finite difference scheme to approximate derivatives of θ would have difficulties because of the discontinuity at 0 (and any integer multiple of 2π). In that work the author circumvented this problem by approximating the two point finite difference, $D\theta$, by $\sin(D\theta)$. As the Taylor series of $\sin x \approx x + O(x^3)$ we can see that this method loses accuracy when $|D\theta|$ is large. For example, if 2 point finite differences are used and θ is always 0 or π, then all derivatives vanish and the surface will not be processed.

In order that we do not smear the discontinuity or create spurious oscillations, when taking finite differences we use the value of $\theta(s) + 2k\pi$, where $k \in \mathbb{Z}, \theta(s) \equiv \tan^{-1}(U_2(s)/U_1(s))$, at a stencil point s which minimizes $|\theta(s) - \theta(s_0)|$ where s_0 is a chosen point of the stencil.

For example, in a discrete formulation in two dimensions, if we would like to approximate θ_x using

$$\theta_x(x_i, y_j) \approx \frac{\theta_{i+1,j} - \theta_{i-1,j}}{2\triangle x},$$

then we would instead find $k_0 \in \mathbb{Z}$ that minimizes

$$|(\theta_{i+1,j} + 2\pi k_0) - \theta_{i-1,j}|$$

and then take

$$\theta_x(x_i, y_j) \approx \frac{(\theta_{i+1,j} + 2\pi k_0) - \theta_{i-1,j}}{2\triangle x}.$$

Of course for a surface with only one discontinuity we do not need to search over all values in $\mathbb{Z}$, just in the set $\{-1, 0, 1\}$. Using this method the discontinuity has no effect on the approximation of the derivative.

We then evolve the solution to $\theta_t = L(\theta)$, equivalent with (2), using

$$\theta^{n+1} = \theta^n + \triangle t(L(\theta))$$

at each point and then repeat the evolution process after setting $\theta^n = \theta^{n+1}$ in Ω.

We can also say that this method is analogous to the ENO (essentially nonoscillatory) interpolation methods used for conservation laws [10], except that instead of having an adaptive stencil we have an adaptive Riemann surface.

2.2 Multiple Parameterization Method

Another formulation that is similar (and in most cases equivalent), is the method of choosing a parameterization of S^1 such that the discontinuity in θ is moved away from the region where finite differences are being applied. If we are advancing the solution at a point $x_{i,j}$ this could mean creating a parameterization where $\theta_{i,j} = \pi$, assuming U is sufficently smooth and the values of U near $x_{i,j}$ are similar to those at $x_{i,j}$. If the values of U near $x_{i,j}$ are not similar to those at $x_{i,j}$ then we can choose a different parameterization such as one that sets the mean of $\theta_{k,l} = \pi$, where the $x_{k,l}$ range over all the nodes of the finite difference stencil being applied.

This procedure means that for each gridpoint $x_{i,j}$ we are creating a unique parameterization of S^1, and thus when we derive an equivalence similar to the one in (3), we will have a unique formulation

$$E = \inf_{\theta} \int_{\Omega} F_{i,j}(\theta) dx \tag{6}$$

for each $x_{i,j}$, where $F_{i,j}$ may involve differential operations. The subsequent minimization by finding the Euler-Lagrange equations and applying gradient descent will then give the PDE in θ

$$\frac{\partial \theta}{\partial t} = -\frac{\partial E}{\partial \theta}, \tag{7}$$

to be solved at each $x_{i,j}$. Also note that at each time t_n the $U_{i,j}$ are changed, so the parameterization at $x_{i,j}$ will be different at time t_{n+1}.

This use of many parameterizations may seem like a difficulty, but fortunately the PDEs (7) are all very similar, as will be shown below.

Advancing the solution to (7) for one timestep using a forward-Euler discretization in time is as follows:

1. Assume that $(u, v) = U^n \in S^1$ is given at time $t = t_n$ on a two-dimensional discrete grid $\{x_{i,j}\}$ of Ω.

2. When calculating spatial derivatives needed in (7) at the point $x_{i,j}$, find all points that will be used in the stencils, and call this set $S_{i,j}$.

3. Choose a particular vector w that will uniquely determine the parameterization of S^1 when we set the $\theta(w) = 0$, where $\theta(w)$ is the θ corresponding to the vector w (note that there is a difference in the meaning of $\theta(\cdot)$ from section 2.1). Again, this can be chosen as $w = -U_{i,j}$, or as $w = -average_{S_{i,j}}(U_{k,l})$, or in other ways. In practice we have found that $w = -U_{i,j}$ works well, although hypothetically it may encounter problems when the vector field U is highly oscillatory. We then have a perpendicular vector $w^\perp \equiv (-w_2, w_1)$, such that $w, w^\perp$ form the axes of the parameterization where $\theta = 0, \pi/2$ respectively.

4. Create the parameter values $\theta_{k,l}$ for all points $x_{k,l} \in S_{i,j}$ by finding

$$\theta_{k,l} \equiv \tan^{-1}\left(\frac{u_{k,l}w_1^\perp + v_{k,l}w_2^\perp}{u_{k,l}w_1 + v_{k,l}w_2}\right). \tag{8}$$

   Note that this is equivalent to finding θ using the relationship in (5) after a change of basis by multiplying $U_{k,l}$ by

$$M = \begin{bmatrix} w_1 & w_1^\perp \\ w_2 & w_2^\perp \end{bmatrix}^{-1} = \begin{bmatrix} w_1 & w_2 \\ w_1^\perp & w_2^\perp \end{bmatrix},$$

   which is also a rotation of by $U_{k,l}$ by θ_0 by multiplying by the matrix

$$R_{\theta_0} = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{bmatrix},$$

   where $\theta_0 = \pi - \theta(-w)$.

5. Derive the energy functional $E(\theta)$ by inverting the relationship

$$\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} = M \begin{pmatrix} u \\ v \end{pmatrix},$$

   which yields

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} M_{1,1}^{-1}\cos\theta + M_{1,2}^{-1}\sin\theta \\ M_{2,1}^{-1}\cos\theta + M_{2,2}^{-1}\sin\theta \end{pmatrix} = \begin{pmatrix} f_1(\theta) \\ f_2(\theta) \end{pmatrix}, \tag{9}$$

   and then substituting $(f_1(\theta), f_2(\theta))$ into (1) in place of (u, v) to get $E(\theta)$.

6. Find the Euler-Lagrange equation for $E(\theta)$ when minimizing with respect to θ, and apply gradient descent to arrive at the PDE (2) to be solved. Note that for a given energy functional (1) the PDEs that are derived from the different parameterizations of S^1 differ only in the coefficients of $\cos\theta$ and $\sin\theta$, which are determined exactly by the entries of the matrix M^{-1}.

7. Advance the solution to $\theta_t = L(\theta)$ using

$$\theta^{n+1} = \theta^n + \triangle t(L(\theta)),$$

where all spatial derivatives of θ in $L(\theta)$ are approximated by finite differences using the $\theta_{k,l}$ where $x_{k,l} \in S_{i,j}$.

8. Recover the updated $U_{i,j}^{n+1}$ using the relationship (9) with θ^{n+1}.

Note that steps 5 and 6 are done prior to implementing the timestepping routine, so that once we have found M^{-1} we can immediately use its entries to determine $L(\theta)$ in step 7.

Examples Here we show 2 examples of the energies found in step 5. The first example is a p-harmonic flow

$$\inf_{U,|U|=1} \int_\Omega |\nabla U|^p dx, \tag{10}$$

where

$$|\nabla U| \equiv \sqrt{\sum_{i=1}^{2}\sum_{j=1}^{2}\left(\frac{\partial U_i}{\partial x_j}\right)}. \tag{11}$$

If we know that at a given grid point we have chosen $w, w^\perp$ such that

$$M = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \text{ and } M^{-1} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{12}$$

then

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c\cos\theta + s\sin\theta \\ -s\cos\theta + c\sin\theta \end{pmatrix}. \tag{13}$$

Substituting into (11) we get

$$\begin{aligned} |\nabla U|^p = [&(\sin^2\theta)(c^2 + (-s)^2)(\theta_x^2 + \theta_y^2) + \\ &(\cos^2\theta)(s^2 + c^2)(\theta_x^2 + \theta_y^2) \\ &- 2\cos\theta\sin\theta(cs - sc)(\theta_x^2 + \theta_y^2)]^{p/2} \\ = &(\theta_x^2 + \theta_y^2)^{p/2} \end{aligned} \tag{14}$$

as $s^2 + c^2 = 1$. So we see that the energy functional is independent of the parameterization of S^1 that we are using.

A second example is

$$\inf_{U,|U|=1} \int_\Omega u_x^2 + v_y^2 dx. \tag{15}$$

Given that we have the same M, M^{-1} as in (12) we can derive

$$\begin{aligned} u_x^2 + v_y^2 &= [-c\sin(\theta)\theta_x + s\cos(\theta)\theta_x]^2 + \\ &\quad [-s\sin(\theta)\theta_y + c\cos(\theta)\theta_y]^2 \\ &= \sin^2\theta(c^2\theta_x^2 + s^2\theta_y^2) + \cos^2\theta(s^2\theta_x^2 + c^2\theta_y^2) \\ &\quad - 2\sin\theta\cos\theta(cs\theta_x^2 - cs\theta_y^2). \end{aligned} \tag{16}$$

In this case the functional depends on the parameterization of S^1 that we choose.

Again we reiterate that the derivation of the PDE from the Euler-Lagrange equation of the energy functional need only be done once for a general matrix M of the form (12), and then the values of s, c are substituted in the discretization of the PDE when it is advanced at each gridpoint.

3 The S^2 Case

In 3 dimensions we have the constrained minimization problem

$$E(U) = \int_\Omega g(U)dx,\ U = (u, v, w) \in \mathbb{R}^3 \tag{17}$$

along with the constraint $|U| = 1$. On S^2 we have the parameterization

$$U \equiv (\cos\phi\cos\theta, \cos\phi\sin\theta, \sin\phi). \tag{18}$$

After substituting the functions of θ, ϕ from (18) into (17), applying the gradient descent method yields a system of equations

$$\frac{\partial\theta}{\partial t} = -\frac{\partial E}{\partial\theta}, \tag{19}$$

$$\frac{\partial\phi}{\partial t} = -\frac{\partial E}{\partial\phi}. \tag{20}$$

Now we have not only the discontinuity in θ, but also a singularity at $\phi = \pm\pi/2$. We treat the singularity in ϕ by rotating our coordinate system so that when taking finite differences we are as far away from it as possible. Thus our method in 3d is a generalization of the multiple parameterization method given in 2d.

3.1 Multiple Parameterization Method

Advancing the solution to (19), (20) for one timestep using a forward-Euler discretization in time is as follows:

1. Assume that $(u, v, w) = U^n \in S^2$ is given at time $t = t_n$ on a three-dimensional discrete grid $\{x_{i,j,k}\}$ of Ω.

2. When calculating spatial derivatives needed in (19), (20) at the point $x_{i,j,k}$, find all points that will be used in the stencils, and call this set $S_{i,j,k}$.

3. Choose a particular vector a that will uniquely determine the parameterization of S^2. This can be chosen as $a = U_{i,j,k}$, or in other ways such as the mean of the $U_{l,m,n} \in S_{i,j,k}$. In the parameterization this vector will have values $(\theta, \phi) = (\pi, 0)$.

4. Create the parameter values $\theta_{l,m,n}, \phi_{l,m,n}$ for all points $x_{l,m,n} \in S_{i,j,k}$ by first projecting onto a new basis by multiplying $U_{l,m,n}$ by M:

$$W = MU^t_{l,m,n} = \begin{bmatrix} \cos\phi_0 \cos\theta_0 & -\cos\phi_0 \sin\theta_0 & -\sin\phi_0 \\ \sin\theta_0 & \cos\theta_0 & 0 \\ \sin\phi_0 \cos\theta_0 & -\sin\phi_0 \sin\theta_0 & \cos\phi_0 \end{bmatrix} U^t_{l,m,n}$$

where

$$\theta_0 = \pi - \theta(a), \text{ and } \phi_0 = \phi(a).$$

The matrix M is a rotation matrix that is the composition $R_{\phi_0} R_{\theta_0}$ of the two matrices

$$R_{\theta_0} = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$R_{\phi_0} = \begin{bmatrix} \cos\phi_0 & 0 & -\sin\phi_0 \\ 0 & 1 & 0 \\ \sin\phi_0 & 0 & \cos\phi_0 \end{bmatrix}.$$

The columns of $M^{-1} = M^t$ give us the basis of vectors onto which we are projecting U, and which act as the x, y, z vectors do in the standard basis (e_1, e_2, e_3). Note that the first column of M^{-1} is

$$\begin{pmatrix} \cos(\phi(a))\cos(\pi - \theta(a)) \\ -\cos(\phi(a))\sin(\pi - \theta(a)) \\ -\sin(\phi(a)) \end{pmatrix} = \begin{pmatrix} -\cos(\phi(a))\cos(\theta(a)) \\ -\cos(\phi(a))\sin(\theta(a)) \\ -\sin(\phi(a)) \end{pmatrix} = -a^t,$$

so that the discontinuity in θ and the singularity in ϕ have been moved away from the region of computation. We then use the relationships

$$\phi = \tan^{-1}\left(\frac{W_3}{\sqrt{W_1^2 + W_2^2}}\right), \ \theta = \tan^{-1}\left(\frac{W_2}{W_1}\right), \tag{21}$$

to determine the values of $\theta_{l,m,n}, \phi_{l,m,n}$ in our new parameterization.

5. Derive the energy functional $E(\theta, \phi)$ by inverting the relationship

$$\begin{pmatrix} \cos\phi\cos\theta \\ \cos\phi\sin\theta \\ \sin\phi \end{pmatrix} = M \begin{pmatrix} u \\ v \\ w \end{pmatrix},$$

which yields

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = M^{-1} \begin{pmatrix} \cos\phi\cos\theta \\ \cos\phi\sin\theta \\ \sin\phi \end{pmatrix} = \begin{pmatrix} f_1(\theta,\phi) \\ f_2(\theta,\phi) \\ f_3(\theta,\phi) \end{pmatrix}, \tag{22}$$

and then substituting $(f_1(\theta,\phi), f_2(\theta,\phi), f_3(\theta,\phi))$ into (17) in place of (u,v,w) to get $E(\theta,\phi)$.

6. Find the Euler-Lagrange equations for $E(\theta,\phi)$ when minimizing with respect to θ,ϕ, and apply gradient descent to arrive at the PDEs (19), (20) to be solved. Note that as in the 2d case, for a given energy functional (17) the PDEs that are derived from the different parameterizations of S^2 differ only in the coefficients of $\cos\phi\cos\theta, \cos\phi\sin\theta,$ and $\sin\phi$, which are determined exactly by the entries of the matrix M^{-1}.

7. Advance the solution to $\theta_t = L(\theta,\phi), \phi_t = J(\theta,\phi)$ using

$$\theta^{n+1} = \theta^n + \triangle t(L(\theta,\phi)),$$

$$\phi^{n+1} = \phi^n + \triangle t(J(\theta,\phi)),$$

where all spatial derivatives of θ,ϕ in L, J are approximated by finite differences using the $\theta_{l,m,n}, \phi_{l,m,n}$ where $x_{l,m,n} \in S_{i,j,k}$.

8. Recover the updated $U_{i,j,k}^{n+1}$ using the relationships (22) with θ^{n+1}, ϕ^{n+1}.

Note that M and $M^{-1} = M^t$ in both the S^1 and S^2 cases can be applied quickly by using a lookup table or any other fast trigonometric evaluator, so the complexity of the algorithm is about the same as if the change in parameterization was not applied.

Example Here we show an example of the energy found in step 5. As we did for S^1 we examine the p-harmonic flow

$$\inf_{U,|U|=1} \int_\Omega |\nabla U|^p dx, \tag{23}$$

where

$$|\nabla U| \equiv \sqrt{\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{\partial U_i}{\partial x_j}\right)}. \tag{24}$$

If we know that at a given grid point we have chosen M such that

$$M^{-1} = M^t = \begin{bmatrix} c\hat{c} & s & c\hat{s} \\ -s\hat{c} & c & -s\hat{s} \\ -\hat{s} & 0 & \hat{c} \end{bmatrix},$$

then

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} c\hat{c}\cos\phi\cos\theta + s\cos\phi\sin\theta + c\hat{s}\sin\phi \\ -s\hat{c}\cos\phi\cos\theta + c\cos\phi\sin\theta - s\hat{s}\sin\phi \\ -\hat{s}\cos\phi\cos\theta + \hat{c}\sin\phi \end{pmatrix}. \tag{25}$$

Substituting into (24) we get (omitting the intermediate calculations)

$$|\nabla U|^p = (\theta_x^2 + \theta_y^2 + \theta_z^2 + \cos^2\phi(\phi_x^2 + \phi_y^2 + \phi_z^2))^{p/2} \tag{26}$$

as $s^2 + c^2 = 1, \hat{s}^2 + \hat{c}^2 = 1$. So we see that the energy functional is independent of the parameterization of S^2 that we are using, like it was for p-harmonic flows in S^1.

As is the case for S^1, the derivation of the PDE from the Euler-Lagrange equation of the energy functional need only be done once for a general matrix M, and then the values of $s, c, \hat{s}, \hat{c}$ are substituted in the discretization of the PDE when it is advanced at each gridpoint.

4 Numerical Examples

For the S^1 case, we first perform the p-harmonic flow with $p = 2$. This is the problem described in (3) and (4). $U = (u, v)$ is initialized as

$$u(x,y) = \frac{x}{|(x,y)|} + 0.1\eta + 0.6(1 + ((x+1)/2)^2 - ((y+1)/2)^2),$$

$$v(x,y) = \frac{y}{|(x,y)|} + 0.1\eta + 0.6((x+1)/2 - 2(y+1)/2),$$

where η is uniform random noise on $[-1, 1]$ for $(x, y) \in [-1, 1]^2$.

We prescribe Dirichlet boundary conditions of the minimal solution $\frac{x}{|x|}$. Vector plots of the initial and final values of U are shown in Figure 1, and θ plots are shown in Figure 2.

If the Neumann boundary conditions, $\frac{\partial U}{\partial n} = 0$, are used for the same problem, we obtain the results shown in Figures 3 and 4.

For the p-harmonic flow with $p = 1$ in figure 5 we impose Neumann boundary conditions and run the evolution on a set of vectors consisting of 4 distinct homogeneous regions with added noise. Note how the "edges" are preserved while the noise is removed.

For the S^2 case we first perform the p-harmonic flow with $p = 1$ for $(x, y, z) \in [-1, 1]^3$. The energy to be minimized is

$$\inf_{U,|U|=1} \int_\Omega |\nabla U| dx = \inf_{\theta,\phi} \int_\Omega \sqrt{|\nabla\phi|^2 + \cos^2\phi|\nabla\theta|^2} dx, \tag{27}$$

which, by gradient descent, gives the time-dependent partial differential equations

$$\phi_t = \nabla \cdot \left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2 + \cos^2\phi|\nabla\theta|^2}}\right) + \frac{\sin\phi\cos\phi|\nabla\theta|^2}{\sqrt{|\nabla\phi|^2 + \cos^2\phi|\nabla\theta|^2}} \tag{28}$$

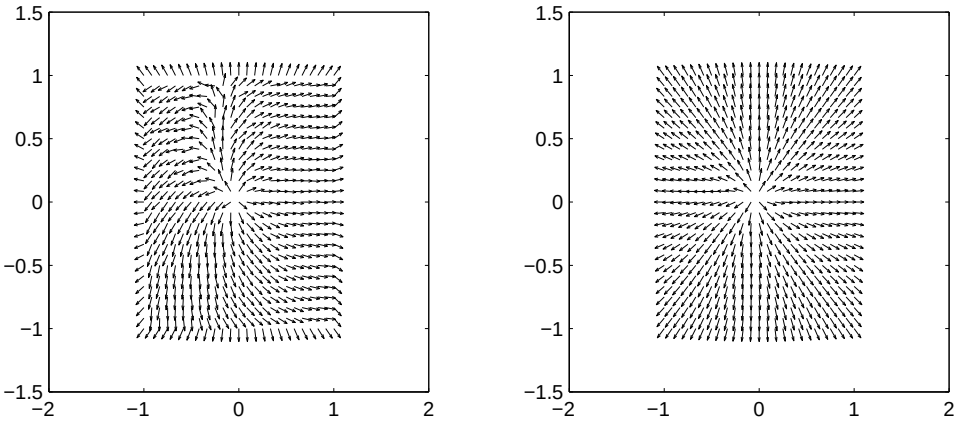

Figure 1: U of p-harmonic flow with $p = 2$, Dirichlet BCs.

$$\theta_t = \nabla \cdot (\cos^2 \phi \frac{\nabla \theta}{\sqrt{|\nabla \phi|^2 + \cos^2 \phi |\nabla \theta|^2}}). \tag{29}$$

Following [1] and [20], U is initialized as $U(\bar{x}) = \frac{\bar{x}-\bar{x}_0}{|\bar{x}-\bar{x}_0|}$, where $\bar{x}_0 = (0.6, 0.6, 0.6)$. Dirichlet boundary conditions of the minimal solution $\frac{\bar{x}}{|\bar{x}|}$ are used. Vector plots are shown at initialization and at steady state in Figure 6.

In the above examples 2^{nd} order central finite differences are used in conjunction with the methods described in sections 2 and 3. Time evolution is done using forward Euler advancement.

We also show an application to RGB color image denoising. A color image I is decomposed into red, green and blue (RGB) channels, $I = (I_R, I_G, I_B) \in \mathbb{R}^3$, from which we can extract the brightness $|I| = \sqrt{I_R^2 + I_G^2 + I_B^2}$, and the chromaticity $I/|I| = (I_R, I_G, I_B)/|I| \in S^2$. In Figure 7 we add noise to the chromaticity vectors only, leaving the brightness unchanged. Gaussian noise is added to 50% of the vectors, and then they are denoised by running a p-harmonic flow with $p = 1$ solving the PDEs given in (28) and (29) for a two-dimensional domain. If noise had been added to the brightness as a grayscale image, then this could have been denoised separately, see [14], [18], [5], [11], [16].

5 Acknowledgements

Tom Cecil and Luminita Vese have been supported in part by NSF grants ITR/AP 0113439 and DMS 0312222. Stanley Osher has been supported in part by ONR grant N00014-02-1-0720. The authors were also supported by a grant from the National Institute of Mental Health and the National Institute of Neurological Disorders and Stroke (Grant #: MH65166).

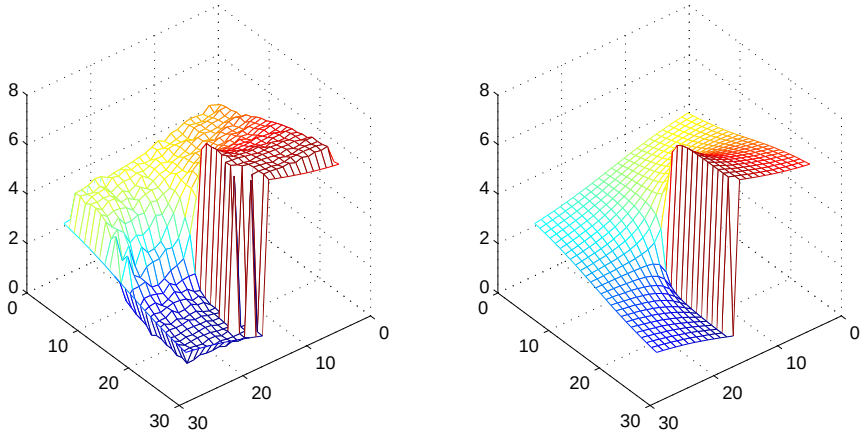

Figure 2: θ of p-harmonic flow with $p = 2$, Dirichlet BCs.

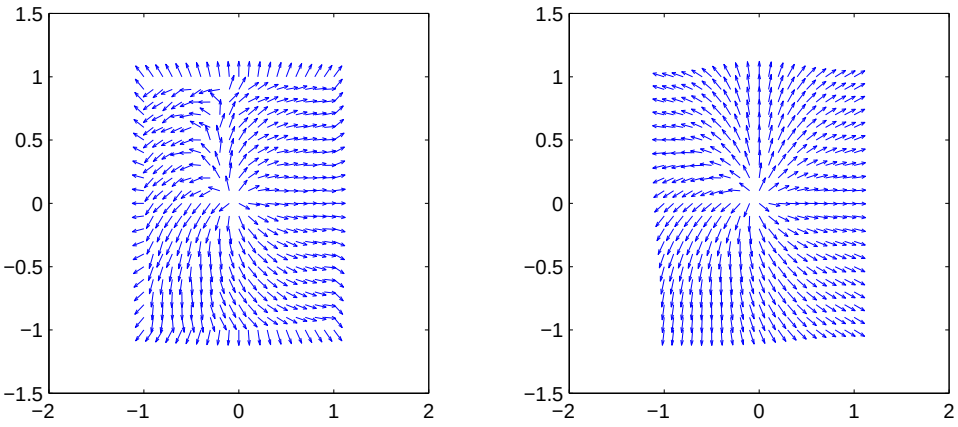

Figure 3: U of p-harmonic flow with $p = 2$, Neumann BCs.

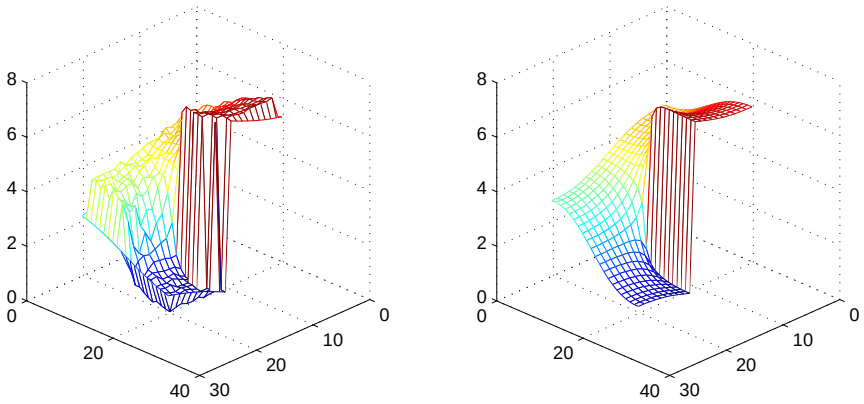

Figure 4: θ of p-harmonic flow with $p = 2$, Neumann BCs.

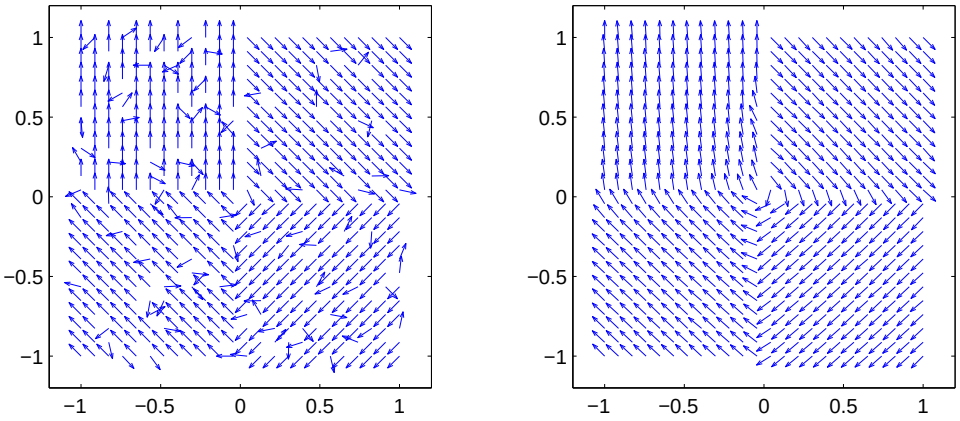

Figure 5: U of p-harmonic flow with $p = 1$, Neumann BCs.

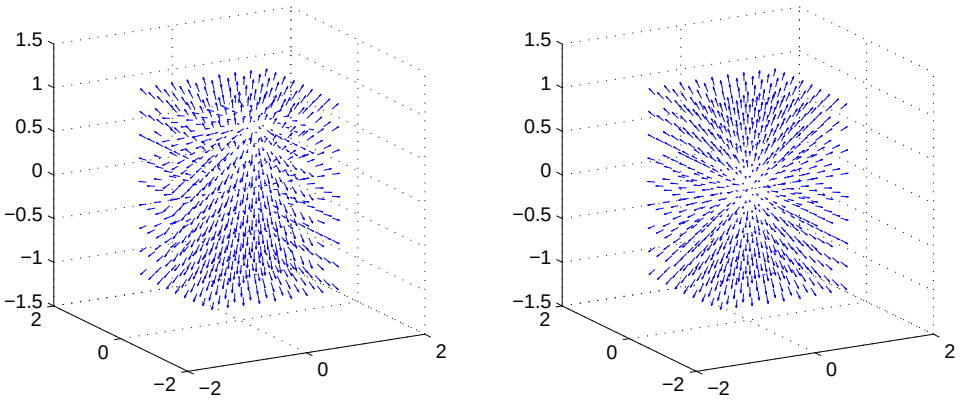

Figure 6: U of p-harmonic flow with $p = 1$, Dirichlet BCs.

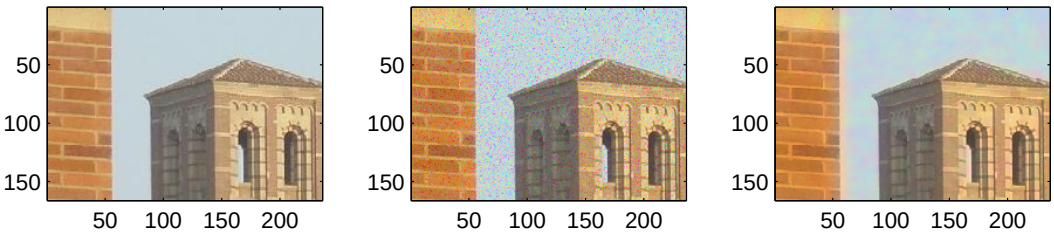

Figure 7: Color denoising, p-harmonic flow with $p = 1$, Neumann BCs. Left-Right: Original, Noisy, Denoised.

References

[1] François Alouges. A new algorithm for computing liquid crystal stable configurations: the harmonic mapping case. *SIAM J. Numer. Anal.*, 34(5):1708–1726, 1997.

[2] F. Bethuel, H. Brezis, and J.-M. Coron. Relaxed energies for harmonic maps. In *Variational methods (Paris, 1988)*, volume 4 of *Progr. Nonlinear Differential Equations Appl.*, pages 37–52. Birkhäuser Boston, Boston, MA, 1990.

[3] Fabrice Bethuel, Haïm Brezis, and Frédéric Hélein. Singular limit for the minimization of Ginzburg-Landau functionals. *C. R. Acad. Sci. Paris Sér. I Math.*, 314(12):891–895, 1992.

[4] Fabrice Bethuel, Haïm Brezis, and Frédéric Hélein. Asymptotics for the minimization of a Ginzburg-Landau functional. *Calc. Var. Partial Differential Equations*, 1(2):123–148, 1993.

[5] Tony Chan and Jianhong Shen. Variational restoration of nonflat image features: models and algorithms. *SIAM J. Appl. Math.*, 61(4):1338–1361 (electronic), 2000/01.

[6] Robert Cohen, Robert Hardt, David Kinderlehrer, San Yih Lin, and Mitchell Luskin. Minimum energy configurations for liquid crystals: computational results. In *Theory and applications of liquid crystals (Minneapolis, Minn., 1985)*, volume 5 of *IMA Vol. Math. Appl.*, pages 99–121. Springer, New York, 1987.

[7] P. Courilleau, S. Dumont, and R. Hadiji. Regularity of minimizing maps with values in S^2 and some numerical simulations. *CMLA-ENS Technical Report*, (9905), 1998.

[8] Weinan E and Xiao-Ping Wang. Numerical methods for the Landau-Lifshitz equation. *SIAM J. Numer. Anal.*, 38(5):1647–1665 (electronic), 2000.

[9] M. Giaquinta, G. Modica, and J. Souček. Variational problems for maps of bounded variation with values in S^1. *Calc. Var. Partial Differential Equations*, 1(1):87–121, 1993.

[10] Ami Harten, Stanley Osher, Björn Engquist, and Sukumar R. Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Appl. Numer. Math.*, 2(3-5):347–377, 1986.

[11] Ron Kimmel and Nir Sochen. Orientation diffusion or how to comb a porcupine. *Journal of Visual Communication and Image Representation*, 13(1-2), 2002.

[12] F. Memoli, G. Sapiro, and S. Osher. Solving variational problems and partial differential equations mapping into general target manifolds. *J. Comput. Phys.*, to appear.

[13] P. Perona. Orientation diffusions. *IEEE Transactions on Image Processing*, 7(3), 1998.

[14] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60, 1992.

[15] Richard Schoen and Karen Uhlenbeck. Regularity of minimizing harmonic maps into the sphere. *Invent. Math.*, 78(1):89–100, 1984.

[16] Nir Sochen, Ron Kimmel, and Ravikanth Malladi. A general framework for low level vision. *IEEE Trans. Image Process.*, 7(3):310–318, 1998.

[17] Michael Struwe. *Variational methods*. Springer-Verlag, Berlin, 1990. Applications to nonlinear partial differential equations and Hamiltonian systems.

[18] B. Tang, G. Sapiro, and V. Caselles. Color image enhancement vis chromaticity diffusion. *IEEE Transactions on Image Processing*, 10(5), 2001.

[19] D. Tschumperle and R. Deriche. Regularization of orthonormal vector sets using coupled pde's. *Proceedings 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 3-10, 2001.

[20] Luminita A. Vese and Stanley J. Osher. Numerical methods for p-harmonic flows and applications to image processing. *SIAM J. Numer. Anal.*, 40(6):2085–2104 (electronic) (2003), 2002.