PIECEWISE CONSTANT LEVEL SET METHOD FOR MULTIPHASE MOTION

HONGWEI LI AND XUE-CHENG TAI

Abstract. We apply the Piecewise Constant Level Set Method (PCLSM) to a multiphase motion problem, especially the pure mean curvature motion. We use one level set function to represent multiple regions, and by associating an energy functional which consists of surface tension (proportional to length), we formulate a variational approach for the mean curvature motion problem. Some operator-splitting schemes are used to solve the problem efficiently. Numerical experiments are supplied to show the efficiency for motion by mean curvature for multiphase problems.

Key Words. multiphase motion, level set method, piecewise constant, operator splitting

1. Introduction

Several approaches for multiphase motion have been developed, such as front tracking methods [4], Monte-Carlo methods, phase field methods, diffusion generated MBO-like method and variational level set approach etc, see [37, 24, 22, 23, 18]. Usually, the model problem involves curves meeting at a point with prescribed angles, See Fig.(1). Each interface Γ_{ij}, separates regions Ω_i and Ω_j and moves with a normal velocity

$$(1) \qquad v_{ij} = f_{ij}\kappa_{ij} + (e_i - e_j).$$

where κ_{ij} is the local curvature, f_{ij} is the constant surface tension of Γ_{ij}, and e_i corresponds to the bulk energy. This model problem can be obtained by associating an energy functional E to the motion, which involves the length of each interface and the area of each subregion, i.e.

$$(2) \qquad \begin{aligned} E &= E_1 + E_2 \\ E_1 &= \sum_{1 \leqslant i < j \leqslant n} f_{ij} Length(\Gamma_{ij}) \\ E_2 &= \sum_{1 \leqslant i \leqslant n} e_i Area(\Omega_i). \end{aligned}$$

By minimizing this energy functional, the internal interfaces are driven to equilibrium. Examples of such kind of motion include solid, liquid, grain or multiphase boundaries [37].

Since complicated topological changes can occur for the model problem, front tracking methods are generally hard to implement, especially for more than two dimensions. Monte-Carlo methods are usually too slow to find good approximations. Phase field methods may also have difficulties to resolve the interface layers [24].

MBO method was proposed by Merriman, Bence and Osher [18]. This method naturally handles complicated topological changes with junctions. MBO method is based on the diffusion of characteristic functions of each region, followed by a certain reassignment step. And it has been shown that MBO is appropriate for pure mean curvature flow, in which the bulk energies are zero and the f_{ij} are all equal to the same positive constant. A rigorous convergence proof for two phase mean curvature motion has been given in [3].

Our work is especially inspired by [18] and [37]. In [37], a variational approach and a practical method was proposed for treating junctions even when topological mergings and breakings occur. Their method also limits angles to the classical condition [33]

$$(3) \qquad \frac{\sin\theta_1}{f_{23}} = \frac{\sin\theta_2}{f_{31}} = \frac{\sin\theta_3}{f_{12}},$$
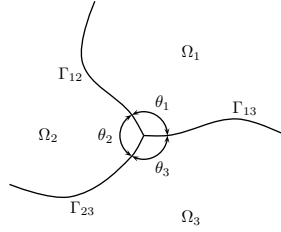
at triple junction points.

FIGURE 1. The interfaces Γ_{ij}, subjected to angles $\theta_1,\theta_2,\theta_3$

In addition to [18] and [37], the traditional level set idea of Osher and Sethian [19] has been used for motion by mean curvature in [14, 6, 8, 26, 29, 5, 20]. For traditional level set methods, one needs to reinitialize the level set function to be a signed distance function during the iterations, and cautions must be taken with respect to the discretization of Heaviside and Dirac functions [7]. The piecewise constant level set method (PCLSM) [13, 12, 11] is trying to use some alternatives for the traditional level set idea, and one doesn't need to care about these issues [13]. Moreover, the formulation is truly variational and just one level set function is needed to identify arbitrary number of phases [34, 37]. Another advantage is that the constraint and energy functionals for our approach are all smooth functionals which do not involve Heaviside and Dirac functions.

In this paper, we apply the piecewise constant level set method [11, 13, 12] to multiple phase motion problems. The remainder of this paper is organized as follows: In section 2, we introduce the piecewise constant level set method (PCLSM), and formulate the mean curvature motion problem into a variational problem by PCLSM. In section 3, two algorithms will be proposed using some operator-splitting techniques. Section 4 goes for numerical experiments.

2. The piecewise constant level set formulation

We shall use the PCLSM of [13] to solve the mean curvature motion problem. The essential idea of the PCLSM of [13] is to use a piecewise constant level set function to identify the subdomains. Assume that we need to partition the domain

Ω into subdomains Ω_i, $i = 1, 2, \ldots, n$ and the number of subdomains is a priori known. In order to identify the subdomains, we try to identify a piecewise constant level set function ϕ such that

$$(4) \qquad \phi = i, \quad \text{in } \Omega_i, \quad i = 1, 2, \ldots, n.$$

Thus, for any given partition $\{\Omega_i\}_{i=1}^n$ of the domain Ω, it corresponds to a unique PCLS function ϕ which takes the values $1, 2, \cdots, n$. Associated with such a level set function ϕ, the characteristic functions of the subdomains are given as

$$(5) \qquad \psi_i = \frac{1}{\alpha_i} \prod_{j=1, j \neq i}^n (\phi - j), \quad \alpha_i = \prod_{k=1, k \neq i}^n (i - k).$$

If ϕ is given as in (4), then we have $\psi_i(x) = 1$ for $x \in \Omega_i$, and $\psi_i(x) = 0$ elsewhere. We can use the characteristic functions to extract geometrical information for the subdomains and the interfaces between the subdomains. For example,

$$(6) \qquad \text{Length}(\partial \Omega_i) = \int_\Omega |\nabla \psi_i| dx, \quad \text{Area}(\Omega_i) = \int_\Omega \psi_i dx.$$

In fact, the level set function also satisfies the relation $\phi = \sum i \psi_i$. Define

$$(7) \qquad K(\phi) = (\phi - 1)(\phi - 2) \cdots (\phi - n) = \prod_{i=1}^n (\phi - i).$$

At every point in Ω, the level set function ϕ should satisfy

$$(8) \qquad K(\phi) = 0.$$

This level set idea has been used for image segmentation in [13, 30, 31] and inverse problems involving shape identification in [32]. Fast algorithms have also been developed for this method for image segmentation in [30, 31].

In the following, the PCLSM will be used to solve the motion problem by mean curvature. For simplicity, let us consider problem (2) with

$$(9) \qquad e_i = 0, \quad f_{ij} = 1.$$

We want to emphasize that there is no problem to apply PCLSM for general setting for (2). Under condition (9), the problem (2) is reduced to the model problem:

$$(10) \qquad \min_{\Gamma_{ij}} \sum_{1 \leqslant i < j \leqslant n} Length(\Gamma_{ij}).$$

There are different ways to find the curves that minimize the above energy functional. Under the condition that Γ_{ij} is the interface between Ω_i and Ω_j, and $\{\Omega_i\}_{i=1}^n$ are represented by (4), we see that

$$\sum_{i=1}^n \int_\Omega |\nabla \psi_i| dx = 2 \sum_{1 \leqslant i < j \leqslant n} Length(\Gamma_{ij}).$$

Thus, If we use our PCLSM for (10), then we need to find a function ϕ that solves the following constrained minimization problem:

$$(11) \qquad \min F, \quad F = \sum_{i=1}^n \int_\Omega |\nabla \psi_i| dx, \quad \text{subject to } K(\phi) = 0.$$

In (11), we have not addressed the boundary conditions for ϕ, which should be necessary when solving the problem. We can take Dirichlet condition or Neumann

boundary condition. Usually, Neumann condition is assumed, just as in many papers in the literature. However, in this paper, we would like try Dirichlet boundary conditions, which should produce a constrained motion. So our problem is

(12) $\min F$, subject to $K(\phi) = 0$ and $\phi|_{\partial\Omega} = g$.

Because we have assumed that $K(\phi) = 0$, g should also satisfy $K(g) = 0$, which means that $g(\cdot) = 1 \vee 2 \vee \ldots \vee n$. Notice that this doesn't specialize or constrain the problem. Once we have transformed the problem into a minimization problem under a constraint, there are efficient methods to solve the minimization problem. Penalization method and operator splitting algorithm will be used in the next section to compute the solutions.

3. PCLSM for curvature motion problems

We can use the penalization technique to deal with the constraint $K(\phi) = 0$ in (12). The penalized problem can be written as

(13) $\min L, \quad L = F + \dfrac{1}{2\mu} \displaystyle\int_\Omega W(\phi)dx,$

where $W(\phi) = K^2(\phi)$ and $\mu > 0$ is the penalization parameter. Gradient type methods will be used to find the minimizers of (13) with respect to ϕ. So we need to calculate the differential $\dfrac{\partial F}{\partial \phi}$. According to the definition of Gateaux differential, we have

$$
\begin{aligned}
\frac{\partial F}{\partial \phi} &= -\sum_{i=1}^{n} \nabla \cdot \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right) \psi_i'(\phi), \\
W'(\phi) &= 2K(\phi)K'(\phi).
\end{aligned}
$$

To find a minimizer of (12), we need to solve

(14) $\dfrac{\partial L}{\partial \phi} = \dfrac{\partial F}{\partial \phi} + \dfrac{1}{2\mu}W'(\phi) = 0,$

with boundary conditions

(15) $\phi|_{\partial\Omega} = g.$

The equation (14) is nonlinear, so we would like to use a standard technique to solve it – instead of solving (14) directly, we can solve the following equation to steady state

(16) $\phi_t + \dfrac{\partial F}{\partial \phi} + \dfrac{1}{2\mu}W'(\phi) = 0.$

We will use some operator-splitting schemes to solve (16). More precisely, we solve the following equations alternatively on $t \in [t_j, t_{j+1}]$ where $t_j = j\tau$ and τ is the time step size:

(17) $\phi_t + \dfrac{\partial F}{\partial \phi}(\phi) = 0, \ t \in [t_j, t_{j+1}],$

(18) $\phi_t + \dfrac{1}{2\mu}W'(\phi) = 0, \ t \in [t_j, t_{j+1}].$

The first equation is trying to minimize the energy functional and the second equation is trying to enforce that the minimizer is taking the values $1, 2, \cdots, n$. In the following, we shall give some details about how to solve these two equations.

3.1. Gradient and operator splitting schemes. The first approach we have tried to solve (17) is the explicit gradient method. The iteration is simply:

$$(19) \qquad \phi^{k+1/2} = \phi^k - \tau \sum_{i=1}^{n} \nabla \cdot \left(\frac{\nabla \psi_i^k}{|\nabla \psi_i^k|} \right) \psi_i'(\phi^k).$$

We may do many iterations of the above updating before we solve (18). The above gradient iteration is easy to implement and very stable if the time step τ is properly chosen. However, it is normally very slow. The second approach we have tried to solve this equation is the Additive Operator Splitting (AOS) scheme of [15, 16, 35]. The AOS scheme was first proposed by Lu, Neittaanmaki and Tai [15, 16]. It was discovered independently later in [35] and used in a different context for image processing [2, 1, 36, 28]. The cost of the AOS for solving (17) is not much higher than the gradient iteration, but it can use large time steps and converge to a steady state much faster.

Notice that

$$(20) \qquad \nabla \psi_i = \psi_i'(\phi) \nabla \phi.$$

and

$$(21) \qquad \frac{\partial F}{\partial \phi} = -\sum_{i=1}^{n} \nabla \cdot \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right) \psi_i' = -\sum_{i=1}^{n} \nabla \cdot \left(\frac{\psi_i' \nabla \phi}{|\psi_i' \nabla \phi|} \right) \psi_i'$$

$$(22) \qquad = -\sum_{i=1}^{n} \nabla \cdot \left(\frac{\psi_i' \nabla \phi}{|\psi_i'| |\nabla \phi|} \right) \psi_i'$$

$$(23) \qquad = -\sum_{i=1}^{n} \nabla \cdot \left(sign(\psi_i') \frac{\nabla \phi}{|\nabla \phi|} \right) \psi_i'.$$

For two dimensional problems, we have

$$(24) \qquad \frac{\partial F}{\partial \phi} = -\sum_{i=1}^{n} \psi_i' \left(sign(\psi_i') \frac{\phi_x}{|\nabla \phi|} \right)_x - \sum_{i=1}^{n} \psi_i' \left(sign(\psi_i') \frac{\phi_y}{|\nabla \phi|} \right)_y.$$

If we apply the AOS to (17) and do some standard linearization, then we need to solve

$$(25) \qquad \frac{\tilde{\phi}^{k+1/4} - \phi^k}{\tau} - \sum_{i=1}^{n} \psi_i'(\phi^k) \left(sign(\psi_i'(\phi^k)) \frac{\tilde{\phi}_x^{k+1/4}}{|\nabla \phi^k|} \right)_x = 0,$$

$$(26) \qquad \frac{\tilde{\phi}^{k+1/2} - \phi^k}{\tau} - \sum_{i=1}^{n} \psi_i'(\phi^k) \left(sign(\psi_i'(\phi^k)) \frac{\tilde{\phi}_y^{k+1/2}}{|\nabla \phi^k|} \right)_y = 0.$$

Then, set

$$(27) \qquad \phi^{k+1/2} = \frac{1}{2}(\tilde{\phi}^{k+1/4} + \tilde{\phi}^{k+1/2}).$$

The two equations (25)–(26) can be solved efficiently on lines parallel to the x and y-axes. We will refer to the above algorithm as semi-implicit AOS scheme hereafter.

3.2. Newton scheme for the constraint equation. It is not a good strategy to solve equation (18) by an explicit scheme. The scheme we have used for it is to solve ϕ^{k+1} from

$$(28) \qquad \frac{\phi^{k+1} - \phi^{k+1/2}}{\tau} + \frac{1}{2\mu} W'(\phi^{k+1}) = 0,$$

n	$\tau/\mu \leq$
2	2
3	0.67
4	0.08
...	...

TABLE 1.   The upper bounds for τ/μ to guarantee the unique solution for the equation (28).

and $\phi^{k+1/2}$ is the solution given by (27) or (19). To solve the above equation is to find the roots of a polynomial of order $2n - 1$. There are many efficient algorithms to find the roots for polynomials. As the solution is not unique, it is necessary to find a way to select the desired solution. Define

$$G(\phi) = \phi + \frac{\tau}{2\mu} W'(\phi) - \phi^{k+1/2}.$$

it is easy to see that

$$G'(\phi) = 1 + \frac{\tau}{\mu} |K'(\phi)|^2 + \frac{\tau}{\mu} K(\phi) K''(\phi).$$

By choosing τ and μ satisfying

$$(29) \qquad \frac{\tau}{\mu} \min_{x \in [0, n+1]} K(x) K''(x) + 1 > 0,$$

it can be guaranteed that (28) has a unique (real) solution which means all the other solutions are complex (non-real) roots. To find this unique solution, we normally use Newton method to find the root. Some simple calculation indicates that we need to use the upper bound as given in Table 1 in order to guarantee the uniqueness of the solutions of (28). In what follows, we shall denote

$$\alpha = \frac{\tau}{2\mu}.$$

It is easy to see that this is a "soft" version of the MBO projection which was used in [18] for the constraint. Similar projection idea has been used in [10] for liquid crystal model.

There are two parameters that we need to choose for actual computing, i.e. τ and μ. In order to have better convergence, it is preferred to choose τ as big as possible. For (19), the CFL condition normally implies that we need to choose the time step below a (small) constant. Due to the semi-implicit nature of (25)–(26), it is also true that the time step needs to be below a constant. The penalization parameter μ needs to be chosen very small to guarantee that the constraint $K(\phi) = 0$ is satisfied with good accuracy. However, in order to guarantee a unique real solution of (28), we need to have $\alpha = \frac{\tau}{2\mu}$ below a constant.

In order to satisfy all these requirements, the following strategy has been used in our simulations. We skip the solving of (28) for the first k_0 iterations, and then switch it on. We also use a fixed value for $\frac{\tau}{2\mu}$ for solving (28) after it has been "switched on". The theoretical values for the upper bound for $\alpha = \frac{\tau}{2\mu}$ are given in table 1. If the algorithm is becoming unstable after equation (28) is "switched on", we reduce the value of τ to get the algorithm to be stable.

3.3. The algorithms. Integrating all the above ingredients together, we get basically two algorithms: explicit scheme, which involves the computation of local curvature, and semi-implicit AOS scheme. We outline these two algorithms as follows:

Algorithm 1 (Explicit scheme) For k = 0, 1, 2, . . .

    a) Solve $\phi^{k+1/2}$ from

$$(30) \qquad \phi^{k+1/2} = \phi^k - \tau \sum_{i=1}^{n} \nabla \cdot \left(\frac{\nabla \psi_i^k}{|\nabla \psi_i^k|} \right) \psi_i'(\phi^k).$$

    b) Set $\phi^{k+1} = \phi^{k+1/2}$ if $k \leq k_0$. Otherwise, solve ϕ^{k+1} from

$$(31) \qquad \phi^{k+1} + \alpha W'(\phi^{k+1}) = \phi^{k+1/2}.$$

    c) Check convergence, if not convergent, *goto* a), else, stop.

Algorithm 2 (semi-implicit **AOS** scheme) For k = 0, 1, 2, . . .

    a) Solve

$$(32) \qquad \frac{\widetilde{\phi}^{k+1/4} - \phi^k}{\tau} - \sum_{i=1}^{n} \psi_i'(\phi^k) \left(sign(\psi_i'(\phi^k)) \frac{\widetilde{\phi}_x^{k+1/4}}{|\nabla \phi^k|} \right)_x = 0,$$

$$(33) \qquad \frac{\widetilde{\phi}^{k+1/2} - \phi^k}{\tau} - \sum_{i=1}^{n} \psi_i'(\phi^k) \left(sign(\psi_i'(\phi^k)) \frac{\widetilde{\phi}_y^{k+1/2}}{|\nabla \phi^k|} \right)_y = 0.$$

    b) Set

$$\phi^{k+1/2} = \frac{1}{2}(\widetilde{\phi}^{k+1/4} + \widetilde{\phi}^{k+1/2}).$$

    c) Set $\phi^{k+1} = \phi^{k+1/2}$ if $k \leq k_0$. Otherwise, solve ϕ^{k+1} from

$$(34) \qquad \phi^{k+1} + \alpha W'(\phi^{k+1}) = \phi^{k+1/2}.$$

    d) Check convergence, if not convergent, *goto* a), else, stop.
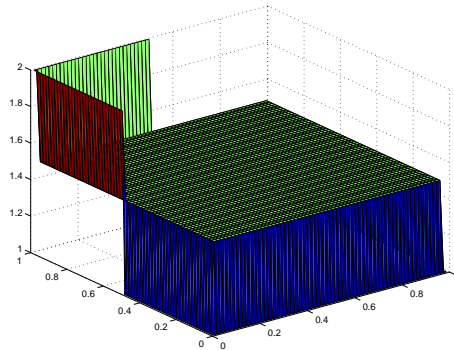
4. Numerical Experiments

In all the experiments, we take $\Omega = (0,1) \times (0,1)$ and use Dirichlet boundary conditions. The domain Ω is divided into square elements with uniform mesh size $h = hx = hy = 1/64$. For Algorithm 1, we set $\tau/h^2 = 5$ at beginning, and set $\tau/h^2 = 0.001$ after k_0 iterations. For Algorithm 2, we set $\tau/h^2 = 50$ at the beginning, and set $\tau/h^2 = 0.1$ after k_0 iterations.

Finite difference is used to approximate the differential operators. To avoid dividing by zeros, we replaces

$$\nabla \cdot \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right), \quad \frac{\psi_i'}{|\psi_i'|}, \quad \frac{1}{|\nabla \phi|}$$

by the following approximations with $\epsilon_i, i = 1, 2, 3$ properly chosen:

$$\nabla \cdot \left(\frac{\nabla \psi_i}{\sqrt{|\nabla \psi_i|^2 + \epsilon_1}} \right), \quad \frac{\psi_i'}{|\psi_i'| + \epsilon_2}, \quad \frac{1}{\sqrt{|\nabla \phi|^2 + \epsilon_3}}.$$

FIGURE 2. Example 1, 2 phases, initial ϕ. Dirichlet boundary conditions

In our simulations, we have taken $\epsilon_1 = \epsilon_3 \approx 10^{-4}$ or 10^{-5} and $\epsilon_2 = 0.1$. The curvature term is approximated by the standard finite difference, see [21] and [17, p.1349]. We have also used the same techniques to evaluate the term $1/|\nabla\phi^k|$ for equations (32) and (33). Due to the linearization techniques used for equations (32) and (33), they are reduced to the solving of some tridiagonal matrix equations, see [15, 16, 35].

4.1. Example 1: a two-phase problem. For this first example, we test on a simple problem. The boundary and initial values are illustrated in Fig.(2). We have $g(x,y)|_{\partial\Omega} = 1$, except $g(0, [1/2, 1]) = g([0, 1/2], 1) = 2$. The initial values of ϕ are $\phi^0|_\Omega = 1.5$ and $\phi^0|_{\partial\Omega} = g$. Because we try to minimize the length of the interface curves, the real solution should consist of straight line segments. We have tested both Algorithm 1 and Algorithm 2 for this problem, and the computed results are displayed in Fig.(3) and Fig.(4). The results produced by both of the algorithms are nearly the same. The computed interface is a straight line. For Algorithm 1, it needs about 11000 iterations to get the computed solution, while for Algorithm 2, we just need about 2200 iterations.

4.2. Example 2: a three-phase problem. In this example, we test our algorithm on the well-known triple-junction problem which involves three phases. The boundary and initial values are illustrated in Fig.(5), i.e. $\phi^0|_\Omega = 1.0$, $g(0, [0, 1/2]) = g([0, 1], 0) = 1$, $g(0, [1/2, 1]) = g([0, 1], 1) = 3$, $g(1, [0, 1]) = 2$.

We have tested both Algorithm 1 and Algorithm 2 for this example. See Fig.(6) and Fig.(7) for the results. For this test problem, we know the real solution. The real triple junction point should be at $(1 - 1/2\sqrt{3}, 1/2/)$ which is approximately $(0.7118, 0.5)$. The three interface curves should be straight lines and the three angles at the triple junction point should satisfy the classical angle condition, i.e. $(\frac{2\pi}{3}, \frac{2\pi}{3}, \frac{2\pi}{3})$ in this case. Both algorithms can approximate the real solution accurately, while Algorithm 1 needs about 60,000 iterations, Algorithm 2 needs only about 2100 iterations.

4.3. Example 3: problem with multiple connected phases. In this last example, we try to use our PCLSM to solve a problem with multiple connected phases. In all the examples we have tested, each phase contains a simple connected region. In this example, we shall test our method for problems that a phase can contain multiple connected regions.

(a) 300 iterations

(b) 1000 iterations

(c) 5000 iterations

(d) 10000 iterations

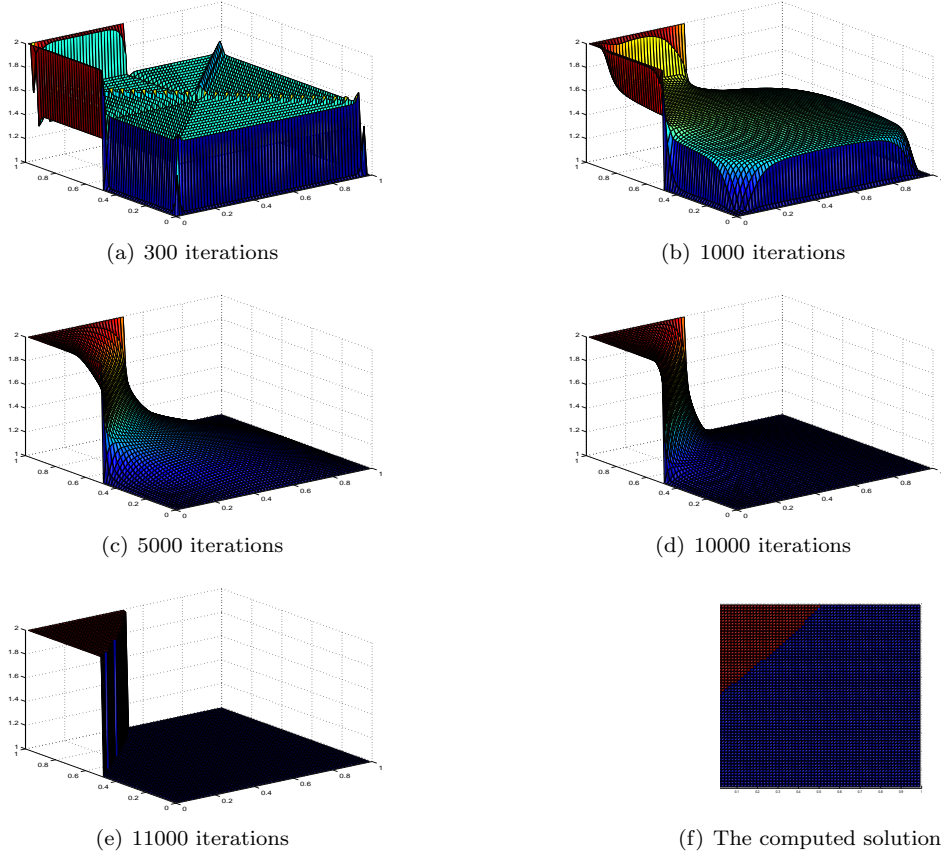(e) 11000 iterations

(f) The computed solution

FIGURE 3. Example 1: A two-phase problem. Computed results by Algorithm 1 with $k_0 = 10000$ and $\alpha = 0.1$.

For this example, the boundary and initial values are illustrated in Fig.(8), i.e. $\phi^0|_\Omega = 1.0$, $g(0, [0, 1/3]) = g([0, 1/3], 0) = 3$, $g([1/3, 2/3], 0) = g(0, [1/3, 2/3]) = 2$, $g([2/3, 1], 0) = g(0, [2/3, 1]) = 1$, $g(1, [1/3, 2/3]) = g([1/3, 2/3], 1) = 2$, $g([2/3, 1], 1) = g(1, [2/3, 1]) = 3$, $g([0, 1/3, 1], 0) = g(1, [1, 1/3]) = 1$. This problem has three phases. We have tested Algorithm 2 for this problem. See Fig.(9) for the results. In this case, both phase 1 and phase 3 contain two distinct regions, while phase 2 has a simple connected region. We see that Algorithm 2 can also compute the solution efficiently and needs only about 2500 iterations.

5. Conclusion

The purpose of this work is to show that the PCLSM of [13] can be used for interface problems coming from mean curvature motion. The method is able to use just one level set function to capture mean curvature motion as well as triple junctions. Moreover, we could use the AOS method to get efficient algorithms. The experiments given here plus the tests done in [13, 12, 11] give a good indication that the PCLSM can be used for a large class of interface problems.

In this work, we have just tested the PCLSM of [13]. In fact, the same technique can be used for the binary level set method of [12] which extends the ideas of [27, 9]

(a) 10 iterations

(b) 200 iterations

(c) 1000 iterations

(d) 2000 iterations

(e) 2200 iterations
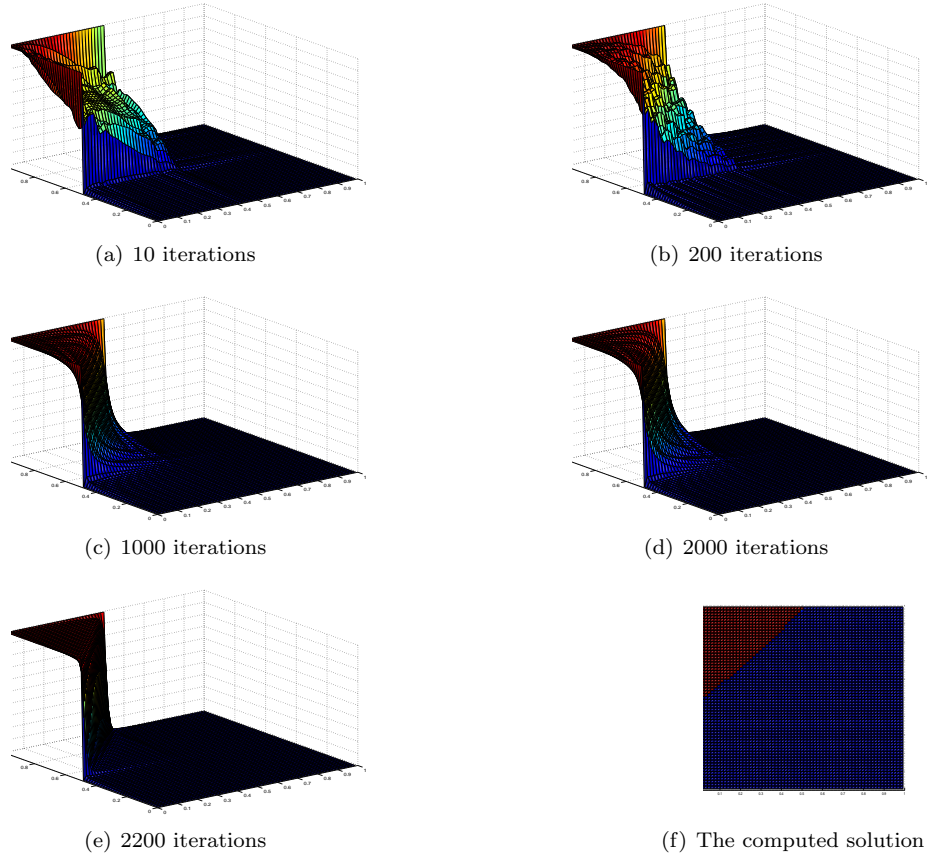
(f) The computed solution

FIGURE 4. Example 1: A two-phase problem. Computed results by Algorithm 2 with $k_0 = 2000$ and $\alpha = 0.1$.
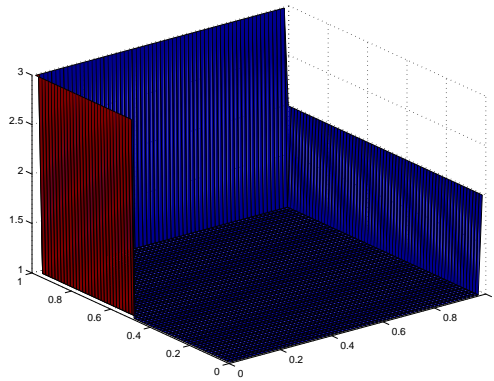

FIGURE 5. Example 2: 3 phases, initial ϕ. Dirichlet boundary conditions

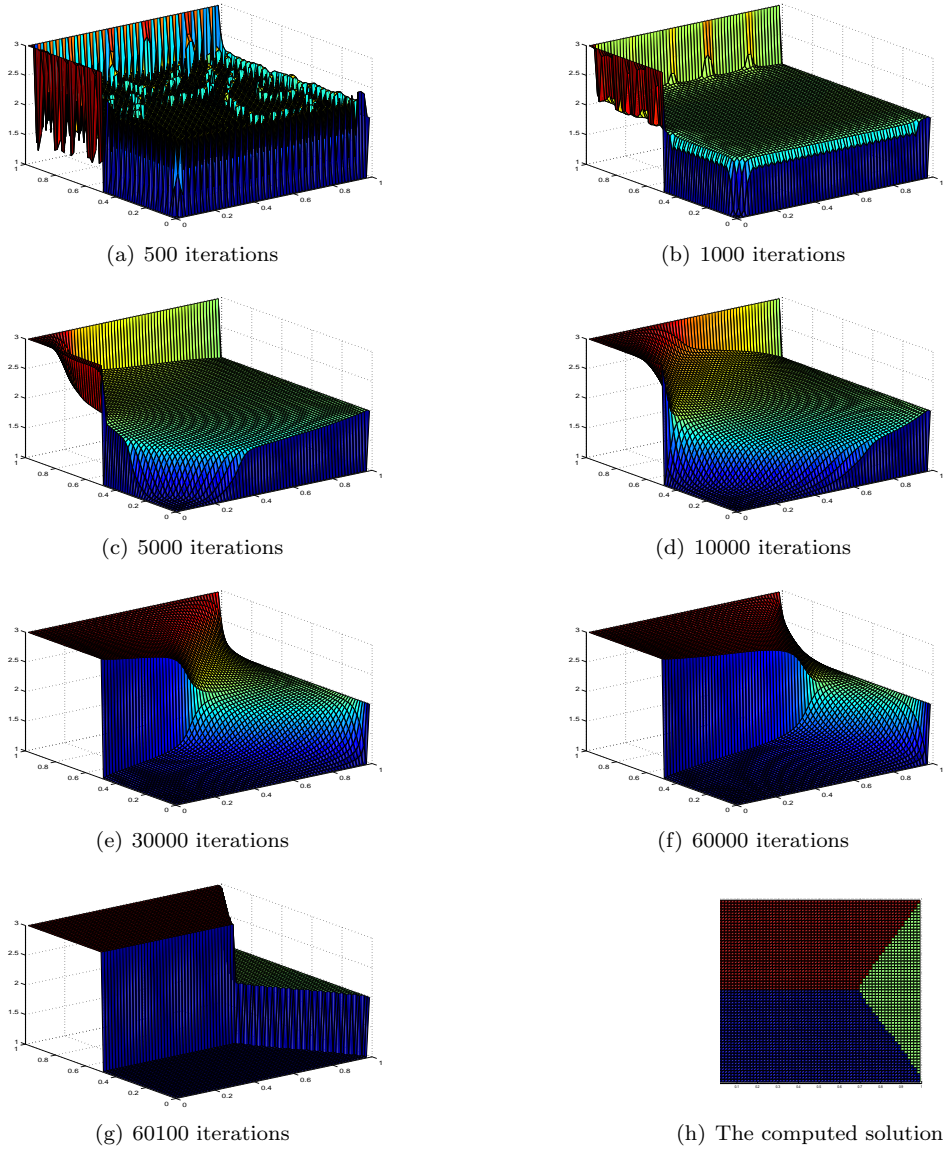and the phase field model [25] . It is very easy to extend our algorithms to three and higher dimensional problems.

(a) 500 iterations

(b) 1000 iterations

(c) 5000 iterations

(d) 10000 iterations

(e) 30000 iterations

(f) 60000 iterations

(g) 60100 iterations

(h) The computed solution

FIGURE 6. Example 2: Computed solution by Algorithm 1 with $k_0 = 60000$ and $\alpha = 0.1$.

References

[1] Danny Barash and. Multiplicative operator splittings in nonlinear diffusion: from spatial splitting to multiple timesteps. *J. Math. Imaging Vision*, 19(1):33–48, 2003.
[2] Danny Barash. Nonlinear diffusion filtering on extended neighborhood. *Appl. Numer. Math.*, 52(1):1–11, 2005.
[3] G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM J. Numer. Anal.*, 32(2):484, 1995.
[4] L. Bronsard and B.T. R. Wetton. A numerical method for tracking curve networks moving with curvature motion. *J. Comput. Phys.*, 120(1):66, 1995.
[5] Paul Burchard, Li-Tien Cheng, Barry Merriman, and Stanley Osher. Motion of curves in three spatial dimensions using a level set approach. *J. Comput. Phys.*, 170(2):720–741, 2001.
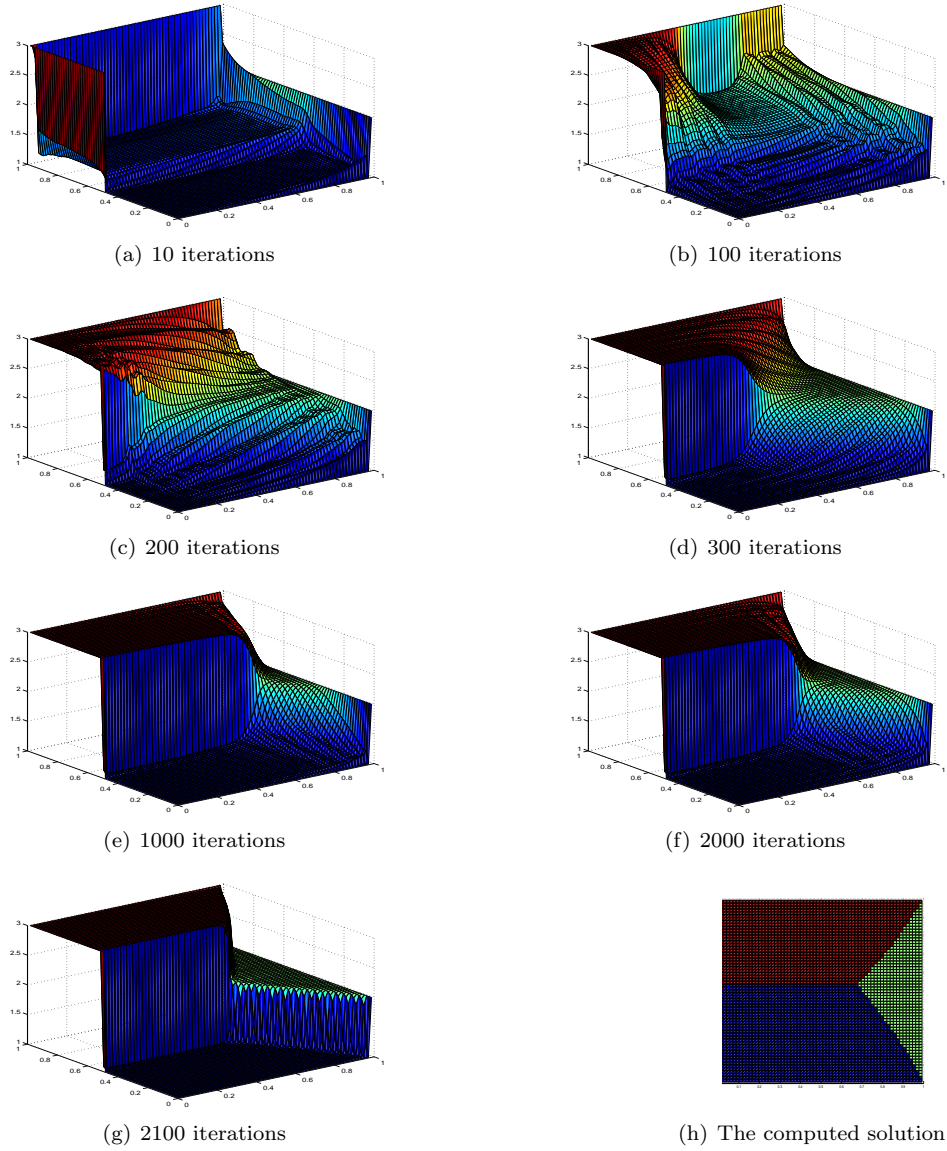
(a) 10 iterations

(b) 100 iterations

(c) 200 iterations

(d) 300 iterations

(e) 1000 iterations

(f) 2000 iterations

(g) 2100 iterations

(h) The computed solution

FIGURE 7. Example 2: Computed solution by Algorithm 2 with $k_0 = 2000$ and $\alpha = 0.1$.

[6] Li-Tien Cheng. Construction of shapes arising from the Minkowski problem using a level set approach. *J. Sci. Comput.*, 19(1-3):123–138, 2003. Special issue in honor of the sixtieth birthday of Stanley Osher.

[7] Björn Engquist, Anna-Karin Tornberg, and Richard Tsai. Discretization of Dirac delta functions in level set methods. *J. Comput. Phys.*, 207(1):28–51, 2005.

[8] Ronald P. Fedkiw, Guillermo Sapiro, and Chi-Wang Shu. Shock capturing, level sets, and PDE based methods in computer vision and image processing: a review of Osher's contributions. *J. Comput. Phys.*, 185(2):309–341, 2003.

[9] Frédéric Gibou and Ronald Fedkiw. A fast hybrid k-means level set algorithm for segmentation. *Stanford Technical Report*, 2002.
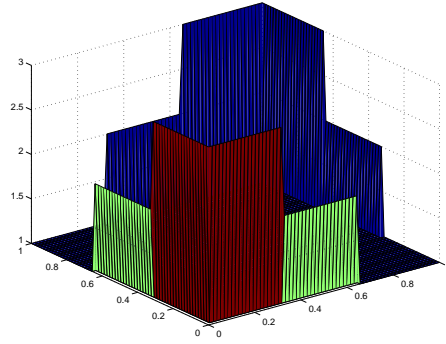
FIGURE 8. Example 3, 3 multiple connected phases, initial g. Dirichlet boundary conditions

(a) 10 iterations

(b) 200 iterations

(c) 1000 iterations

(d) 2000 iterations
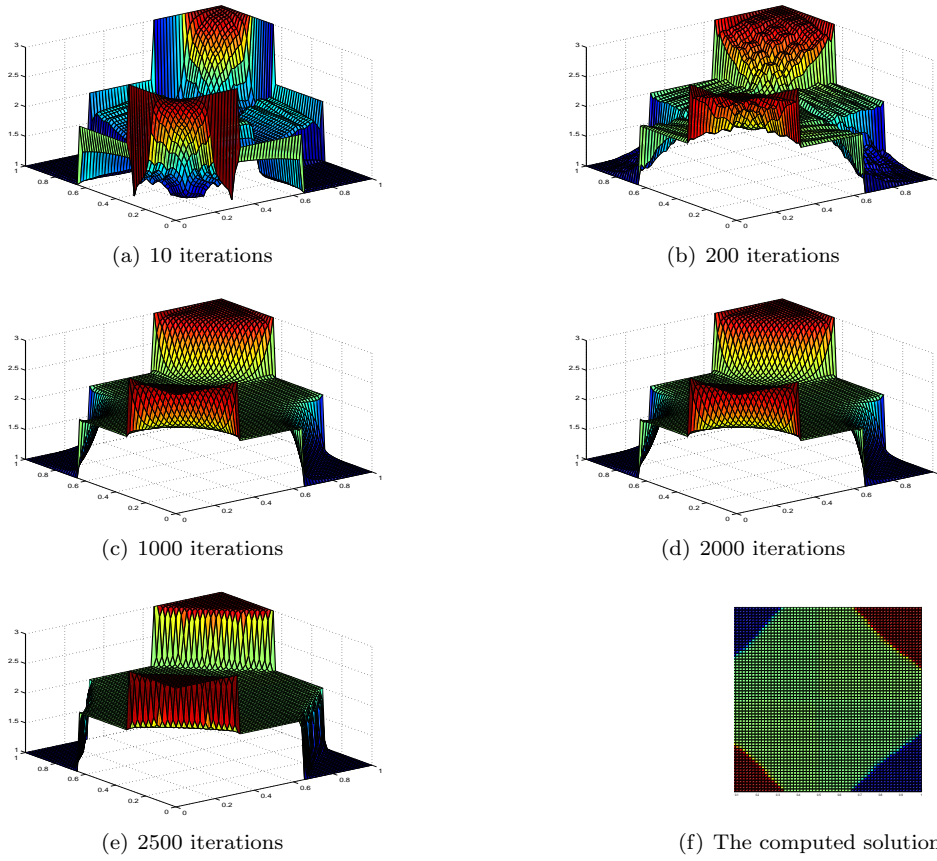

(e) 2500 iterations

(f) The computed solution

FIGURE 9. Example 3: Computed result by Algorithm 2 with $k_0 = 2000$ and $\alpha = 0.1$.

[10] Roland Glowinski, Ping Lin, and Xingbin Pan. An operator-splitting method for a liquid crystal model. *Comput. Physics Comm.*, 152:242–252, 2003.
[11] M. Lysaker J. Lie and X.-C. Tai. Piecewise constant level set methods and image segmentation. In Ron Kimmel, Nir Sochen, and Joachim Weickert, editors, *Scale Space and PDE*

Methods in Computer Vision: 5th International Conference, Scale-Space 2005, volume 3459, pages 573–584. Springer-Verlag, Heidelberg, April 2005.

[12] J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications to image processing. *IEEE Trans. Image Process.*, to appear. Also as UCLA, Applied Math., CAM04-31, 2004.

[13] J. Lie, M. Lysaker, and X.-C. Tai. A variant of the levelset method and applications to image segmentation. *Math. Comp.*, to appear. Also as UCLA, Applied Math. CAM-03-50, 2003.

[14] Zhihong Liu, Richard Wan, Ken Muldrew, Stephen Sawchuk, and John Rewcastle. A level set variational formulation for coupled phase change/mass transfer problems: application to freezing of biological systems. *Finite Elem. Anal. Des.*, 40(12):1641–1663, 2004.

[15] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Appl. Math. Lett.*, 4(2):25–29, 1991.

[16] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting-up method for partial differential equations and its applications to Navier-Stokes equations. *RAIRO Modél. Math. Anal. Numér.*, 26(6):673–708, 1992.

[17] M. Lysaker, Stanley Osher, and Xue-Cheng Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Processing*, 13:1345–1457, 2004.

[18] B. Merriman, J. Bence, and S. Osher. Motion of multiple junctions: A level set approach. *J. Comput. Phys.*, 112(2):334, 1994.

[19] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algrithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[20] Stanley Osher and Ronald P. Fedkiw. Level set methods: an overview and some recent results. *J. Comput. Phys.*, 169(2):463–502, 2001.

[21] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithm. *Physica D.*, 60:259–268, 1992.

[22] S. J. Ruuth. An algorithm for generating motion by mean curvature. In *12th International Conferene on Analysis and Optimization of Systems Images, Wavelets and PDE's*, page 82, Paris, France, 1996.

[23] S. J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. Technical Report 97-33, University of California, Los Angeles, 1997.

[24] S. J. Ruuth. A diffusion-generated approach to multiphase motion. *J. Comput. Phys.*, 145(CP986028):166–192, 1998.

[25] J. Shen. Gamma-convergence approximation to piecewise constant Mumford-Shah segmentation. Tech. Rep. CAM05-16, UCLA Dep. Math, 2005.

[26] Kurt A. Smith, Francisco J. Solis, and David L. Chopp. A projection method for motion of triple junctions by levels sets. *Interfaces Free Bound.*, 4(3):263–276, 2002.

[27] B. Song and T. Chan. A fast algorithm for level set based optimization. *CAM-UCLA*, 68, 2002. Under revision for publication in SIAM J. Sci. Comput.

[28] Gabriele Steidl, Joachim Weickert, Thomas Brox, Pavel Mrázek, and Martin Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variatioin regularization, and slides. *SIAM J. Numer. Anal.*, 42(2):686–713, 2004.

[29] JS Suri, KC Liu, Singh S, SN Laxminarayan, XL Zeng, and L Reden. Shape recovery algorithms using level sets in 2-d/3-d medical imagery: A state-of-the-art review. *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, 6(1):8–28, 2002.

[30] X.-C. Tai and C. Yao. Fast piecewise constant level set method with newton updating. Technical report, to appear.

[31] Xue-Cheng Tai, Oddvar Christiansen, Ping Lin, and Inge Skjaelaaen. A remark on the mbo scheme and some piecewise constant level set methods. Cam-report-05-24, UCLA, Applied Mathematics, 2005.

[32] Xue-Cheng Tai and Hongwei Li. Dynamic constraint and regularization for applying piecewise constant level set method to elliptic inverse problems. (to appear), UCLA, Applied Mathematics, 2005.

[33] Jean E. Taylor. A variational approach to crystalline triple-junction motion. *J. Statist. Phys.*, 95(5-6):1221–1244, 1999.

[34] L. A. Vese and T. F. Chan. A multiphase level set framevork for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[35] J. Weickert, B. H. Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Process.*, 7:398–409, 1998.

[36] Joachim Weickert and Gerald Kühne. Fast methods for implicit active contour models. In *Geometric level set methods in imaging, vision and graphics*, pages 43–57, New York, 2003. Springer.

[37] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:179, 1996.

Center for Integrated Petroleum Research, University of Bergen, CIPR room 4103, Allégaten 41, N-5007 Bergen, Norway
E-mail: `hongwei.li@cipr.uib.no`

Institute of Mathematics, University of Bergen, Joh. Brunsgate 12, N-5008 Bergen, Norway
E-mail: `tai@mi.uib.no`
URL: `http://www.mi.uib.no/~tai`