

SEMI-IMPLICIT COVOLUME METHOD IN 3D IMAGE SEGMENTATION*

S. CORSARO[†], K. MIKULA[‡], A. SARTI[§], AND F. SGALLARI[¶]

Abstract. We introduce a three-dimensional (3D) semi-implicit complementary volume numerical scheme for solving the level set formulation of (Riemannian) mean curvature flow problem. We apply the scheme to segmentation of objects (with interrupted edges) in 3D images. The method is unconditionally stable and efficient regarding computational times. The study of its experimental order of convergence on 3D examples shows its second order accuracy for smooth solutions and first order accuracy for highly singular solutions with vanishing gradients as arising in image segmentation.

Key words. partial differential equations, numerical methods, image segmentation, semi-implicit scheme, complementary volume method

AMS subject classifications. 65M60, 65M12, 68U10

DOI. 10.1137/060651203

1. Introduction. In this paper we introduce a fast and stable computational method for a three-dimensional (3D) image segmentation based on a solution of the following Riemannian mean curvature flow equation:

$$(1.1) \quad u_t = \sqrt{\varepsilon^2 + |\nabla u|^2} \nabla \cdot \left(g(|\nabla G_\sigma * I^0|) \frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}} \right).$$

Here, $u(t, x)$ is an unknown; we call it *segmentation function* or *subjective surface* [31, 32, 33], defined in $Q_T \equiv [0, T] \times \Omega$. The computational domain $\Omega \subset \mathbb{R}^d$, $d = 3$, is bounded by a Lipschitz continuous boundary $\partial\Omega$. The endpoint of the interval $[0, T]$ represents a time when the final segmentation result is obtained. In practice it is chosen a posteriori; the segmentation is stopped when a change of solution in time (e.g., in L_2 -norm) is less than a prescribed threshold. The Riemannian mean curvature flow (1.1) is weighted by a function g giving edge detection information from the segmented image I^0 . The function $g : \mathbb{R}_0^+ \rightarrow \mathbb{R}^+$ is nonincreasing, $g(0) = 1$, admitting $g(s) \rightarrow 0$ for $s \rightarrow \infty$, e.g., $g(s) = 1/(1 + Ks^2)$, $K \geq 0$ [29]. $G_\sigma \in C^\infty(\mathbb{R}^d)$ is a smoothing kernel, e.g., the Gauss function, which is used in presmoothing of image gradients by the convolution. We will use the notation $g^0 = g(|\nabla G_\sigma * I^0|)$, and, due to properties of convolution, $1 \geq g^0 \geq \delta_\sigma > 0$ [8, 17]. The model parameter $1 \geq \varepsilon > 0$ shifts the model from the mean curvature flow of graphs ($\varepsilon = 1$) to the mean curvature flow of level sets ($\varepsilon = 0$) [15]. For segmentation of objects without corrupted boundaries (e.g., due to a noise or occlusion) the choice $\varepsilon = 1$ is optimal.

*Received by the editors January 30, 2006; accepted for publication (in revised form) June 27, 2006; published electronically December 11, 2006. This work was supported by NATO Collaborative Linkage grant PST.CLG.979123, and by grants VEGA 1/3321/06 and APVT-20-040902.

<http://www.siam.org/journals/sisc/28-6/65120.html>

[†]CNR, Institute for High-Performance Computing and Networking (ICAR), Via P. Castellino 111, 80131 Napoli, Italy (stefania.corsaro@dma.unina.it).

[‡]Department of Mathematics, Slovak University of Technology, Radlinského 11, 813 68 Bratislava, Slovakia (mikula@vox.svf.stuba.sk).

[§]DEIS, University of Bologna, Via Risorgimento 2, 40136 Bologna, Italy (asarti@deis.unibo.it).

[¶]Department of Mathematics, University of Bologna, Piazza di Porta S. Donato 5, 40127 Bologna, Italy (sgallari@dm.unibo.it).

When segmenting an object with gaps in encompassing edges, then smaller ε , close to zero, should be used.

Together with the variational approaches following the seminal idea of Mumford and Shah [25], the curve and surface evolution models are powerful techniques used in image segmentation [1, 19, 5, 22, 6, 7, 20, 24]. Equation (1.1) can be formally understood as an Evans–Spruck ε -regularization [15]

$$(1.2) \quad |\nabla u| \approx |\nabla u|_\varepsilon = \sqrt{\varepsilon^2 + |\nabla u|^2}$$

of the segmentation equation

$$(1.3) \quad u_t = |\nabla u| \nabla \cdot \left(g^0 \frac{\nabla u}{|\nabla u|} \right),$$

which is called geodesic or conformal mean curvature flow equation in the level set formulation [6, 7, 20].

In [15], the ε -regularization (1.2) was used as a tool to prove existence of the viscosity solution [12, 10] of the mean curvature motion in the level set formulation given by (1.3) with $g^0 \equiv 1$ [28]. The basic idea is that if the motion of a graph by mean curvature is rescaled by a factor $\frac{1}{\varepsilon}$, and letting $\varepsilon \rightarrow 0$, it gives the evolution of level sets. Equations (1.3) and (1.1), respectively, were studied analytically in [20, 7] and [32, 11], respectively.

In (1.3), where $0 < g^0 \leq 1$, the motion of level sets is influenced by the image features expressed in g^0 . In the case $\varepsilon = 1$, (1.1) gives the motion of a graph by mean curvature with respect to a specific Riemann metric depending on g^0 . Rescaling this motion by $\frac{1}{\varepsilon}$ (or, correspondingly, rescaling the Riemann metric) helps to complete the interrupted edges. The idea to use such special Riemannian mean curvature flow of graphs to extract the so-called subjective contours [18] and to segment images with interrupted edges originates in [31, 32, 33]; such an approach is called the *subjective surface method*. Unlike in (1.3), the subjective surface method does not move one particular (e.g., zero) level set to the boundary of the segmented object, but it moves all the level sets there. The forming discontinuity (shock) in the evolving graph of solution of (1.1) is used to detect object boundaries. Thus the standard level set methods preventing discontinuity formation by reinitialization of the front and applying, e.g., higher order (ENO or WENO) schemes cannot be used here.

Concerning numerical solution, in all previous approaches, the level set type equation (1.3), respectively (1.1), was rewritten into the advection-diffusion form

$$(1.4) \quad u_t = g^0 |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \nabla g^0 \cdot \nabla u,$$

where $|\nabla u|_\varepsilon$ is used instead of $|\nabla u|$ in the case of (1.1). Then this form of equation was solved by a finite difference scheme using upwind in the advection term and explicit time stepping in the curvature term [6, 7, 20, 31, 32, 33]. As it will be discussed in section 3.3, such an approach is computationally inefficient mainly due to severe stability restrictions both in advection and curvature parts.

In spite of the previous methods based on (1.4), our 3D covolume technique suggested in this paper relies on discretization of (1.3) or, more precisely, of its ε -regularization (1.1). Using formulation (1.1) and employing its variational structure we develop an unconditionally stable semi-implicit time discretization scheme which gives the image segmentation in a computationally fast way. The main goal of this paper is to suggest such a method for high-scale 3D image segmentation problems and

show its stability, computational efficiency, and experimental convergence properties. Since the method can be used for any $\varepsilon > 0$ with $\varepsilon \rightarrow 0$, it can be applied to efficient solution of level set flows also in any other mean curvature driven applications.

The covolume technique (also called complementary volume or finite volume element method) is a combination of finite element (see, e.g., [4]) and finite volume methods; see, e.g., [21, 23]. The discrete equations are derived using the finite volume methodology, i.e., integrating equation in the so-called control (complementary, finite) volume. The control volumes are constructed as elements of a dual (complementary) grid to a primal finite element triangulation (tetrahedral grid in the 3D case). Employing the methodology of the finite element method, the piecewise linear representation of solution on a tetrahedral grid is used to evaluate in a clear and simple way the nonlinear quantities, the absolute value of gradients, in the numerical solution of (1.1).

A two-dimensional (2D) covolume technique for solving the level set formulation of mean curvature flow was first introduced in [35]. There also a particular nonlinear fully implicit time discretization scheme was given. The nonlinear scheme was designed in such a way that the basic properties of a continuous solution, namely L_∞ - and $W^{1,1}$ -stability, are fulfilled also for discrete solutions.

In [16] a linear semi-implicit 2D covolume level set method was given and studied. The same basic stability properties were proven for the semi-implicit scheme. Since it solves in every discrete time step linear (instead of nonlinear) algebraic systems, it gives solution in a more efficient way. In [16], the semi-implicit scheme was applied to 2D image filtering problem represented by the nonlinear diffusion level set type equation [2]

$$(1.5) \quad u_t = g(|\nabla G_\sigma * u|) |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right).$$

In this paper we present for the first time a semi-implicit covolume scheme in three dimensions for solving level set type problems and apply the scheme to a different high-scale application—3D image segmentation. The previous 2D covolume strategy is not straightforwardly extended to the 3D case, but a new primal-dual grid construction is developed respecting discrete voxel structure of 3D images. Our new covolume scheme is related to 3D image data representation known in computer graphics as a centered-cubic-lattice method [9]. From the numerical point of view, it avoids any grid orientation effect, which is important for simulation of curve and surface evolution on a fixed grid; we illustrate this phenomenon in section 3.3. Although (1.5), studied in [16], and (1.1) and (1.3), respectively, look similar, their structure is different. In (1.1) and (1.3) the weighting term $g^0 = g(|\nabla G_\sigma * I^0|)$ is inside the divergence which brings a new advective phenomenon into the model; cf. (1.4). The advective term given by the vector field $-\nabla g^0$ is very important in the segmentation context; it drives all level sets to the image edges. Without the convective term (let us think about (1.5) with g^0 instead of $g(|\nabla G_\sigma * u|)$ in the right-hand side), the model is unable to segment nonconvex objects, as discussed in section 3.3, and so the method from [16] cannot be applied to image segmentation. Fortunately, the semi-implicit covolume strategy handles the weighting term inside divergence naturally, without going to advection-diffusion form (1.4), and allows us to develop a fast and stable numerical scheme for 3D image segmentation.

The paper is organized as follows. In section 2 we introduce in detail our 3D semi-implicit covolume method. In section 3 we study its experimental order of convergence in nontrivial examples of the level set evolution and thus its reliability in

simulations. We also discuss segmentation experiments with artificial and real (medical) 3D images and compare the method with other approaches. In this paper we concentrate mainly on algorithmic and computational aspects of the method. A full theoretical convergence analysis is not a simple task, and it is out of scope of this paper; however, it is a subject of our current research.

2. Computational method. We present our method formally in the discretization of (1.3), although we always use its ε -regularization (1.1) with a specific $\varepsilon > 0$. The notation is simpler in the case of (1.3), and it will be clear where regularization appears in the numerical scheme.

Equation (1.1) is accompanied either with Dirichlet boundary conditions

$$(2.1) \quad u(t, x) = u^D \quad \text{in } [0, T] \times \partial\Omega,$$

or Neumann boundary conditions

$$(2.2) \quad \frac{\partial u}{\partial s}(t, x) = 0 \quad \text{in } [0, T] \times \partial\Omega,$$

where s is unit normal to $\partial\Omega$, and with the initial condition

$$(2.3) \quad u(0, x) = u^0(x) \quad \text{in } \Omega.$$

In the image segmentation we use Dirichlet boundary conditions, and without loss of generality we may assume $u^D = 0$. The zero Neumann boundary conditions are often used in computations of interface motions in free boundary problems (see, e.g., [34, 27]) or in morphological image smoothing (see, e.g., [2, 16]). In this paper, we use Dirichlet boundary conditions in computing 3D image segmentations and in comparisons with known exact solutions u . The zero Neumann boundary conditions are used here in testing the accuracy of the method when dealing with the known motion of a particular level set of function u without knowing the exact evolution of the entire function u . In 3D image segmentation, the computational domain Ω is usually a subdomain of the image domain; it should include the segmented object. Along with the boundary conditions an initial condition (2.3) is also prescribed, and we assume that the initial function is bounded, i.e., $u^0 \in L_\infty(\Omega)$.

2.1. Semi-implicit time discretization. First, we choose a uniform discrete time step τ and a variance σ of the smoothing kernel G_σ . Then we replace the time derivative in (1.3) by backward difference. The nonlinear terms of the equation are treated from the previous time step, while the linear ones are considered on the current time level; this means semi-implicitness of the time discretization. By such an approach we get our semi-discrete in time scheme:

Let τ and σ be fixed numbers, g^0 be given by an image I^0 , and u^0 be a given initial segmentation function. Then, for every discrete time moment $t_n = n\tau$, $n = 1, \dots, N$, we look for a function u^n , solution of the equation

$$(2.4) \quad \frac{1}{|\nabla u^{n-1}|} \frac{u^n - u^{n-1}}{\tau} = \nabla \cdot \left(g^0 \frac{\nabla u^n}{|\nabla u^{n-1}|} \right).$$

2.2. Covolume spatial discretization in three dimensions. A 3D digital image is given on a structure of voxels with cubic shape, in general. Since discrete values of image intensity I^0 are given in voxels and they influence the model, we will

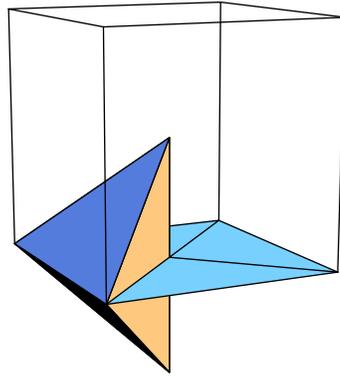


FIG. 2.1. A tetrahedron of our primal grid together with its intersection with the bottom face of voxel covolume is shown.

relate spatially discrete approximations of the segmentation function u also to the voxel structure, more precisely, to voxel centers. In every discrete time step t_n of the method (2.4) we have to evaluate gradient of the segmentation function at the previous step $|\nabla u^{n-1}|$. Toward that goal we put a 3D tetrahedral grid into the voxel structure and take a piecewise linear approximation of the segmentation function on such a grid. Such an approach will give a constant value of gradient in tetrahedra (which is the main feature of the covolume [35, 16] and linear finite element [13, 14] methods in solving mean curvature flow in the level set formulation), allowing simple, clear, and fast construction of a fully discrete system of equations.

The formal construction of our covolumes will be given in the next paragraph, and we will see that the covolume mesh corresponds back to the image voxel structure, which is reasonable in image processing applications. On the other hand, the construction of covolume mesh has to use a 3D tetrahedral finite element grid to which it is complementary. This will be possible using the following approach. First, every cubic voxel is split into six pyramids with vertex given by the voxel center and base surfaces given by the voxel boundary faces. The neighboring pyramids of neighboring voxels are joined together to form an octahedron which is then split into four tetrahedra using diagonals of the voxel boundary face—see Figure 2.1. In such a way we get our *3D tetrahedral grid*. Two nodes of every tetrahedron correspond to centers of neighboring voxels, and further two nodes correspond to voxel boundary vertices; every tetrahedron intersects the common face of neighboring voxels. Let us note that due to our image processing application in mind, we restrict all further considerations only to this type of grid. In our method, only the centers of voxels will represent degree of freedom nodes (DF nodes); i.e., solving the equation at a new time step we update the segmentation function only in these DF nodes. Additional nodes of tetrahedra will not represent degrees of freedom; we will call them nondegree of freedom nodes (NDF nodes), and they will be used in piecewise linear representation of segmentation function. Let a function u be given by discrete values in the voxel centers, i.e., in DF nodes. Then in NDF nodes we take the average value of the neighboring DF nodal values. By such defined values in NDF nodes a piecewise linear approximation u_h of u on the tetrahedral grid is built.

For the tetrahedral grid \mathcal{T}_h , given by the previous construction, we construct a covolume (dual) mesh. We modify the approach given in [35, 16] in such a way that

our covolume mesh will consist of cells p associated only with DF nodes p of \mathcal{T}_h , let us say $p = 1, \dots, M$. Since there will be one-to-one correspondence between covolumes and DF nodes, to avoid any confusion, we use the same notation for them. For each DF node p of \mathcal{T}_h let C_p denote the set of all DF nodes q connected to the node p by an edge. This edge will be denoted by σ_{pq} and its length by h_{pq} . Then every *covolume* p is bounded by the planes e_{pq} that bisect and are perpendicular to the edges $\sigma_{pq}, q \in C_p$. By this construction, if e_{pq} intersects σ_{pq} in its center, the covolume mesh corresponds exactly to the voxel structure of the image inside the computational domain Ω where the segmentation is provided. Then the covolume boundary faces do cross in NDF nodes. So we can also say that NDF nodes correspond to zero-measure covolumes, and thus they do not add additional equations to discrete model, and they do not represent degrees of freedom in the covolume method. We denote by \mathcal{E}_{pq} the set of tetrahedra having σ_{pq} as an edge. In our situation (see Figure 2.1) every \mathcal{E}_{pq} consists of four tetrahedra. For each $T \in \mathcal{E}_{pq}$ let c_{pq}^T be the area of the portion of e_{pq} that is in T , i.e., $c_{pq}^T = m(e_{pq} \cap T)$, where m is the measure in \mathbb{R}^{d-1} . Let \mathcal{N}_p be the set of all tetrahedra that have DF node p as a vertex. Let u_h be a piecewise linear function on \mathcal{T}_h . We will denote a constant value of $|\nabla u_h|$ on $T \in \mathcal{T}_h$ by $|\nabla u_T|$ and define regularized gradients by

$$(2.5) \quad |\nabla u_T|_\varepsilon = \sqrt{\varepsilon^2 + |\nabla u_T|^2}.$$

We will use notation $u_p = u_h(x_p)$, where x_p is the coordinate of DF node p of \mathcal{T}_h .

With these notations, we are ready to derive covolume spatial discretization. As it is usual in finite volume methods [21, 23], we integrate (2.4) over every covolume $p, i = 1, \dots, M$, and using divergence theorem we get an integral formulation

$$(2.6) \quad \int_p \frac{1}{|\nabla u^{n-1}|} \frac{u^n - u^{n-1}}{\tau} dx = \sum_{q \in C_p} \int_{e_{pq}} \frac{g^0}{|\nabla u^{n-1}|} \frac{\partial u^n}{\partial \nu} ds.$$

The exact “fluxes” $\int_{e_{pq}} \frac{g^0}{|\nabla u^{n-1}|} \frac{\partial u^n}{\partial \nu} ds$ in the right-hand side and “capacity function” $\frac{1}{|\nabla u^{n-1}|}$ in the left-hand side (see, e.g., [21]) will be approximated numerically using piecewise linear reconstruction of u^{n-1} on triangulation \mathcal{T}_h . If we denote g_T^0 approximation of g^0 on a tetrahedron $T \in \mathcal{T}_h$, then for the approximation of the right-hand side of (2.6) we get

$$(2.7) \quad \sum_{q \in C_p} \left(\sum_{T \in \mathcal{E}_{pq}} c_{pq}^T \frac{g_T^0}{|\nabla u_T^{n-1}|} \right) \frac{u_q^n - u_p^n}{h_{pq}},$$

and the left-hand side of (2.6) is approximated by

$$(2.8) \quad M_p m(p) \frac{u_p^n - u_p^{n-1}}{\tau},$$

where $m(p)$ is the measure in \mathbb{R}^d of covolume p and M_p is an approximation of the capacity function inside the finite volume p . For that goal we use either the averaging of the gradients proposed by Walkington [35], i.e.,

$$(2.9) \quad M_p = \frac{1}{|\nabla u_p^{n-1}|}, \quad |\nabla u_p^{n-1}| = \sum_{T \in \mathcal{N}_p} \frac{m(T \cap p)}{m(p)} |\nabla u_T^{n-1}|,$$

or

$$(2.10) \quad M_p = \sum_{T \in \mathcal{N}_p} \frac{m(T \cap p)}{m(p)} \frac{1}{|\nabla u_T^{n-1}|},$$

which is close to finite element approximation with the mass lumping. Then the regularization of both approximations of the capacity function is given either by

$$(2.11) \quad M_p^\varepsilon = \frac{1}{|\nabla u_p^{n-1}|_\varepsilon}$$

or by

$$(2.12) \quad M_p^\varepsilon = \sum_{T \in \mathcal{N}_p} \frac{m(T \cap p)}{m(p)} \frac{1}{|\nabla u_T^{n-1}|_\varepsilon},$$

and if we define coefficients (where the ε -regularization is taken into account)

$$(2.13) \quad b_p^{n-1} = M_p^\varepsilon m(p),$$

$$(2.14) \quad a_{pq}^{n-1} = \frac{1}{h_{pq}} \sum_{T \in \mathcal{E}_{pq}} c_{pq}^T \frac{g_T^0}{|\nabla u_T^{n-1}|_\varepsilon},$$

we get from (2.7)-(2.8) our fully discrete semi-implicit covolume scheme:

Let $u_p^0, p = 1, \dots, M$, be given discrete initial values of the segmentation function. Then, for $n = 1, \dots, N$ we look for $u_p^n, p = 1, \dots, M$, satisfying

$$(2.15) \quad b_p^{n-1} u_p^n + \tau \sum_{q \in C_p} a_{pq}^{n-1} (u_p^n - u_q^n) = b_p^{n-1} u_p^{n-1}.$$

Applying either Neumann or Dirichlet boundary conditions, our fully discrete scheme gives the system of linear equations with the matrix which is a symmetric and diagonally dominant M-matrix. Using a similar technique as in [16] we get the following.

THEOREM. *There exists unique solution (u_1^n, \dots, u_M^n) of the scheme (2.15) for any $\tau > 0, \varepsilon > 0$ and for every $n = 1, \dots, N$. The system matrix is a symmetric and diagonally dominant M-matrix. For any $\tau > 0, \varepsilon > 0$ the following L_∞ stability estimate holds:*

$$(2.16) \quad \min_p u_p^0 \leq \min_p u_p^n \leq \max_p u_p^n \leq \max_p u_p^0, \quad 1 \leq n \leq N.$$

One can solve the system (2.15) by any efficient preconditioned linear iterative solver suitable for sparse, symmetric, diagonally dominant M-matrices, e.g., by SOR (successive overrelaxation) or PCG (preconditioned conjugate gradient) methods [30]. In the experiments presented in section 3 we use the SOR method. The iterative process is stopped if $R^{(l)} < \text{TOL} R^{(0)}$, where $R^{(l)}$ represents the squared residual in the l th iteration. Reporting experimental order of convergence we use $\text{TOL}=10^{-6}$. In the segmentation one can increase TOL, e.g., to 10^{-2} , in order to use fewer SOR iterations and consequently to decrease overall segmentation time. No significant influence to a precision of the segmentation result is usually observed.

The evaluation of g_T^0 included in coefficients (2.14) can be done in several ways. First, we may replace the convolution by the weighted average to get $I_\sigma^0 := G_\sigma * I^0$

(see, e.g., [23]) and then relate discrete values of I_σ^0 to voxel centers. Then, as above, we may construct its piecewise linear representation on a grid and get a constant value of $g_T^0 \equiv g(|\nabla I_\sigma^0|)$ on every tetrahedron $T \in \mathcal{T}_h$. Another possibility is to solve a numerically linear heat equation for time t corresponding to variance σ with initial datum given by I^0 (see, e.g., [4]) by the same method as above. The convolution represents a preliminary smoothing of the data. It is also a theoretical tool to have bounded gradients and thus a strictly positive weighting coefficient g^0 . In practice, the evaluation of gradients on a fixed discrete grid (e.g., described above) always gives bounded values. So, working on a fixed grid, one can also avoid the convolution, especially if preliminary denoising is not needed or not desirable. Then it is possible to work directly with gradients of piecewise linear representation of I^0 in evaluation of g_T^0 .

3. Discussion of numerical results. In this section we study experimental order of convergence of the method in case of mean curvature driven level set flow (i.e., for (1.1) with $g \equiv 1$ and $\varepsilon \rightarrow 0$), and then we present some 3D segmentation examples. The comparison with nontrivial exact solutions shows the second order accuracy for smooth (or mildly singular) solutions and the first order accuracy for highly singular solutions (i.e., when gradient is vanishing on a large subset of a domain and discontinuity set of the gradient field is nontrivial). It means that the method is accurate in computing interface motions, and it is reliable also for computing evolutions including flat regions as arising in the subjective surface based segmentation method.

3.1. Experimental order of convergence. In the first example we test the method using the exact solution [26]

$$(3.1) \quad u(x, y, z, t) = (x^2 + y^2 + z^2 - 1)/4 + t$$

of the level set equation

$$(3.2) \quad u_t = |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right)$$

and consider Dirichlet boundary conditions given by this exact solution.

This problem, and all further computed examples in this subsection, are solved in the spatial domain $\Omega = [-1.25, 1.25]^3$ and in the time interval $T = 0.16$. We have taken subsequent grid refinement with $M = n^3$ covolumes (DF nodes), $n = 10, 20, 40, 80, 160$. So the grid size is $h = 2.5/n$. The time step τ is chosen proportionally to h^2 and we measure errors in $L_\infty((0, T), L_2(\Omega))$ -norm (cf. [13]), which is natural for testing the schemes for solving parabolic problems.

Let us assume that the error of the scheme in some norm is proportional to some power of the grid size, i.e., $Error(h) = Ch^\alpha$, with a constant C . Then halving the grid size we have $Error(h/2) = C(h/2)^\alpha$ from where we can simply extract

$$(3.3) \quad \alpha = \log_2(Error(h)/Error(h/2)).$$

The α is called the *experimental order of convergence* (EOC) and can be determined by comparing numerical solutions and exact solutions on subsequently refined grids.

In Table 3.1 we report errors in $L_\infty((0, T), L_2(\Omega))$ -norm for refined grids and for several choices of ε . In all cases we observe $\alpha = 2$, where the coupling $\varepsilon \approx h^2$ seems optimal (see also other examples), but as one can see, choosing ε even smaller can get even smaller errors on coarse grids.

TABLE 3.1
 Errors in $L_\infty((0, T), L_2(\Omega))$ -norm, and EOC comparing numerical and exact solution (3.1).

n	h	$\varepsilon = h^2$	EOC	$\varepsilon = h$	EOC	$\varepsilon = 10^{-6}$	EOC
10	0.25	2.9208e-2		6.5190e-2		2.4936e-2	
20	0.125	5.5133e-3	2.40	2.0893e-2	1.64	5.2251e-3	2.25
40	0.0625	1.3106e-3	2.07	5.4823e-3	1.93	1.2939e-3	2.01
80	0.03125	3.2371e-4	2.01	1.3679e-3	2.00	3.2270e-4	2.00
160	0.015625	8.0548e-5	2.00	3.4086e-4	2.00	8.0485e-5	2.00

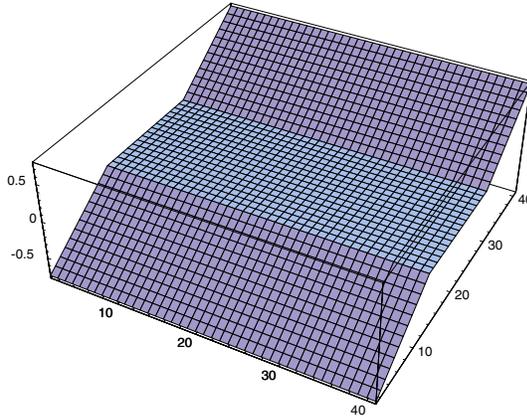


FIG. 3.1. Exact solution (3.4) remaining unchanged under mean curvature flow (plot at $z = 0$).

Next an interesting example comes from [14] and is given by (see Figure 3.1)

$$(3.4) \quad \begin{aligned} u(x, y, z, t) &= x + 0.5, \quad x \leq -0.5 \\ &= 0, \quad -0.5 \leq x \leq 0.5 \\ &= x - 0.5, \quad x \geq 0.5. \end{aligned}$$

The level sets of u are planes, with zero mean curvature, and so the solution remains unaltered by the flow. There is also a large part of the solution where gradient is vanishing; however, the set of discontinuity in gradient is relatively simple (two planes) regarding a possible orientation of our grid (we can make it parallel). Mathematically, it is a trivial example, but it is a good test for a numerical scheme. One can observe (see Table 3.2) that the error of the scheme in this example is proportional just to regularization parameter ε , and thus it can be made as small as desirable. This is a simple consequence of the *consistency* of our scheme in the sense that it gives exact solution for any linear initial function and for any choice of regularization parameter ε , any grid size h , and any size of time step τ . Such a property can be checked by inspection of the scheme (2.15). For the linear function, the gradients in all tetrahedra are the same, and so we get the system with the same structure as given by the backward Euler scheme for solving a linear diffusion equation with constant coefficients, and such a scheme does not alter a steady state.

In the next example we compare our numerical solution with the highly singular solution given by

$$(3.5) \quad u(x, y, z, t) = \min((x^2 + y^2 + z^2 - 1)/4 + t, 0).$$

The initial function and numerical result at $T = 0.16$ are plotted in Figure 3.2. We

TABLE 3.2
 Errors in $L_\infty((0, T), L_2(\Omega))$ -norm, and EOC comparing numerical and exact solution (3.4).

n	h	Error, $\varepsilon = h^2$	EOC	Error, $\varepsilon = 10^{-6}$
10	0.25	5.2252e-2		0.9114e-6
20	0.125	1.5088e-3	1.79	1.0309e-6
40	0.0625	4.0474e-3	1.89	1.0648e-6
80	0.03125	1.0118e-3	2.00	1.0473e-6

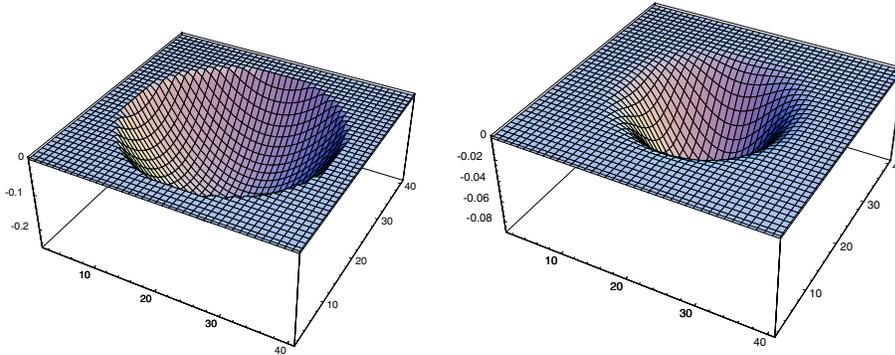


FIG. 3.2. Numerical solution corresponding to exact solution (3.5) plotted at times $t = 0$ (left) and $t = 0.16$ and for $z = 0$.

TABLE 3.3
 Errors in $L_\infty((0, T), L_2(\Omega))$ -norm, and EOC comparing numerical and exact solution (3.5).

n	h	Error, $\varepsilon = h^2$	EOC
10	0.25	6.9571e-2	
20	0.125	4.2686e-2	0.70
40	0.0625	2.2049e-2	0.95
80	0.03125	1.1030e-2	0.99
160	0.015625	5.5544e-3	0.99

see only slight smoothing in numerical solution along singularity; numerical solution converges to viscosity solution with $\alpha = 1$, as reported in Table 3.3.

In the last example we test experimental order of convergence comparing the numerical and exact evolution of one particular level set. Namely, we use the exact solution given by shrinking sphere with exact radius $r(t) = \sqrt{r(0) - 4t}$, starting with $r(0) = 1$. Our initial level set function is a 3D cone (signed distance function) having zero value on the unit sphere. We consider zero Neumann boundary conditions, and so we do not know the exact solution for evolution of the whole level set function. At every time step we evaluate $L_2(S^2)$ -norm of error, where S^2 is the unit sphere, comparing the exact shrinking sphere and the numerical zero level set, and then we take $L_\infty((0, T), L_2(S^2))$ -norm for the overall error in time interval $T = 0.16$. Figure 3.3 represents the initial sphere and the numerical zero level set at $T = 0.16$. Table 3.4 documents the higher order rate of convergence, where again the coupling $\varepsilon = h^2$ is used. Similarly to the first example, different couplings or choice of a fixed small ε give similar errors and convergence rates.

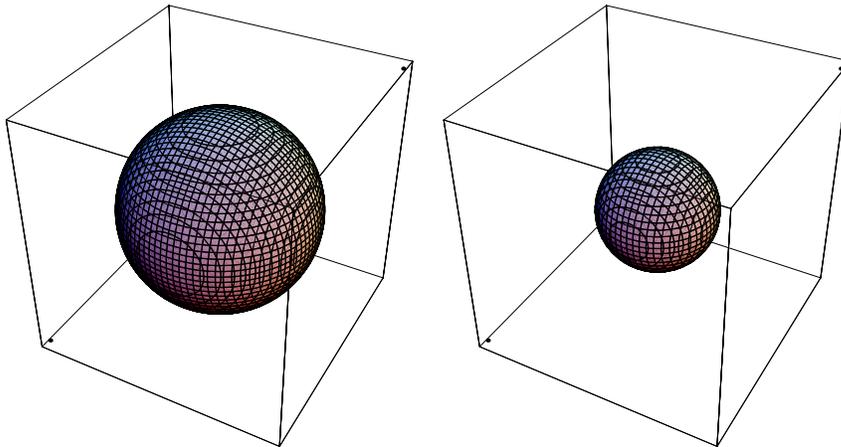


FIG. 3.3. Numerical solution corresponding to exact shrinking sphere with radius $r(t) = \sqrt{r(0) - 4t}$, $r(0) = 1$, plotted at times $t = 0$ (left) and $t = 0.16$.

TABLE 3.4

Errors in $L_\infty((0, T), L_2(S^2))$ -norm, and EOC comparing numerical zero level sets and the exact shrinking sphere.

n	h	Error, $\varepsilon = h^2$	EOC
10	0.25	7.7228e-2	
20	0.125	2.8732e-2	1.42
40	0.0625	7.8269e-3	1.87
80	0.03125	4.7901e-4	4.03
160	0.015625	9.9568e-5	2.26

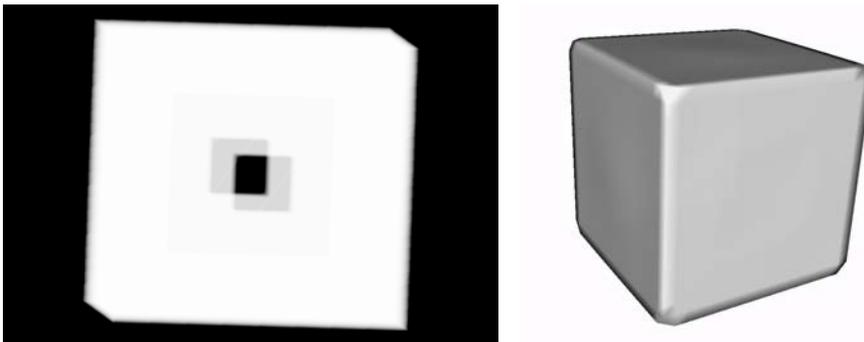


FIG. 3.4. Segmentation (right) of a 3D cube with two holes (left).

3.2. Segmentation examples. Here we present two 3D segmentation examples. The first one is simple; in the image with resolution 40^3 voxels is an object, a cube with two holes (see Figure 3.4(left)); the holes are in front and back faces. Figure 3.4 gives a reconstructed surface with the perfect hole filling (see Figure 3.4(right)). Due to the advection term all level sets are attracted to the faces of the cube. Due to the mean curvature flow type diffusion the level sets continue inside the gaps in the form of planar surfaces. Since in the holes area $g^0 \equiv 1$ and $\nabla g^0 \equiv 0$ just the mean curvature motion of level sets plays the role inside the gaps, and the planar

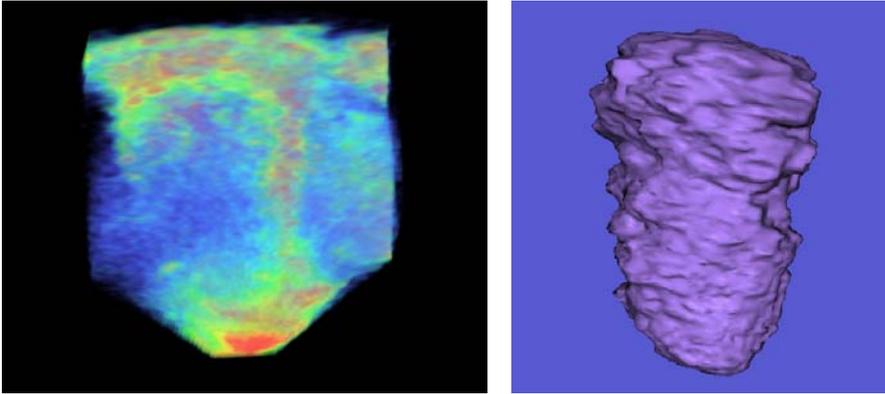


FIG. 3.5. Volume rendering of original 3D data set (left) and segmentation of the ventricle (right).

filling is then a consequence of the basic property of mean curvature flow, namely the minimization of surface area. Segmentation of an object of this size takes few seconds (on a 2.4GHz PC); one time step takes 0.15 sec, and we need few tens of time steps to find the result. A change in L_2 -norm of numerical solutions in subsequent time steps is used to stop the segmentation process. We check whether

$$(3.6) \quad \sqrt{\sum_p m(p) (u_p^n - u_p^{n-1})^2} < \delta$$

with a prescribed threshold δ . In Figure 3.4(right) we visualize isosurface of the segmentation function close to its maximum. In all segmentation examples computed by our semi-implicit scheme we take $K = 1$, $h = \frac{1}{n_1}$, n_1 number of voxels in the horizontal direction, $\tau \approx 10 \times h^2$, and $\delta = 10^{-5}$. We start all computations with the initial function given as a peak centered in a “focus point” inside the segmented object. At a sphere with center s and radius R the initial function can be described by $u^0(x) = \frac{1}{|x-s|+v}$, where s corresponds to the focus point and $\frac{1}{v}$ gives a maximum of u^0 . Outside that sphere we take values of u^0 equal to $\frac{1}{R+v}$. R usually corresponds to the halved inner diameter of the image domain. For small objects a smaller R can be used to speed up computations. In this example the maximum of the initial segmentation function (the focus point s) is taken in the image center.

The second example is given by 3D echocardiography of size $81 \times 87 \times 166$. As one can see from volume rendering visualization (Figure 3.5(left)), the 3D image is very noisy; however, the surface of the ventricle is observable. How noisy the image intensity is can be seen also from Figure 3.6, where one can see 2D image intensity and its graph in one 2D slice. Now we start the segmentation process with few “points of view” inside the object of interest, i.e., the initial segmentation function having several maxima, and let it evolve until the difference in L_2 -norm of two subsequent time steps is less than the prescribed threshold. Then we look to a 2D slice with relatively good boundary edges (Figure 3.6), where we see accumulation of level sets along the inner boundary of the ventricle (Figure 3.6(left)). The largest gap in the histogram (Figure 3.7(left)) indicates the shock in the segmentation function, and so we choose one level inside the gap and plot it inside the slice (Figure 3.7(right)). We can check how this level set looks like in other noisy slices (Figure 3.8) and then make

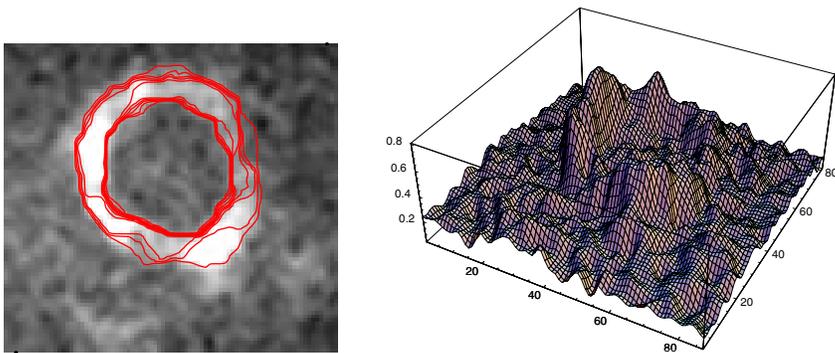


FIG. 3.6. Plot of accumulated level sets in the slice $k = 130$ (left) and 3D graphical view of this slice (right).

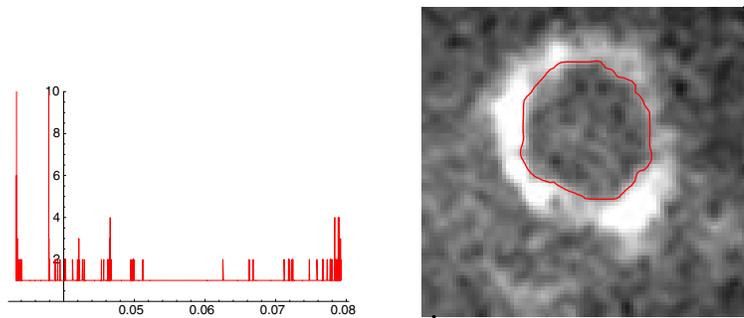


FIG. 3.7. The histogram of the segmentation function in the slice $k = 130$ (middle); image intensity with level set $u = 0.052$ (right). Visualization of the 3D surface in Figure 3.5 is done with the same level set.

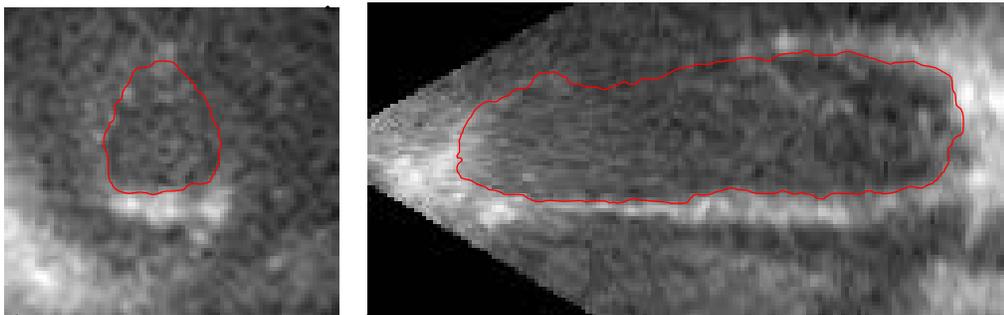


FIG. 3.8. Plot of image intensity together with level line 0.052 in two other slices $k = 100$ (left) and $j = 40$ (right).

a 3D isosurface visualization (Figure 3.5(right)) which gives realistic representation of the left ventricle.

The computation of one time step takes 3 seconds on a 2.4GHz one processor PC and takes 200 steps to finish segmentation, and so the overall computing time for this image size is in a range of few minutes. The MPI parallel implementation [3] of our 3D segmentation scheme has been built under the support of the Project HPC-EUROPA at CINECA SuperComputing Center, Bologna, and so the segmentation

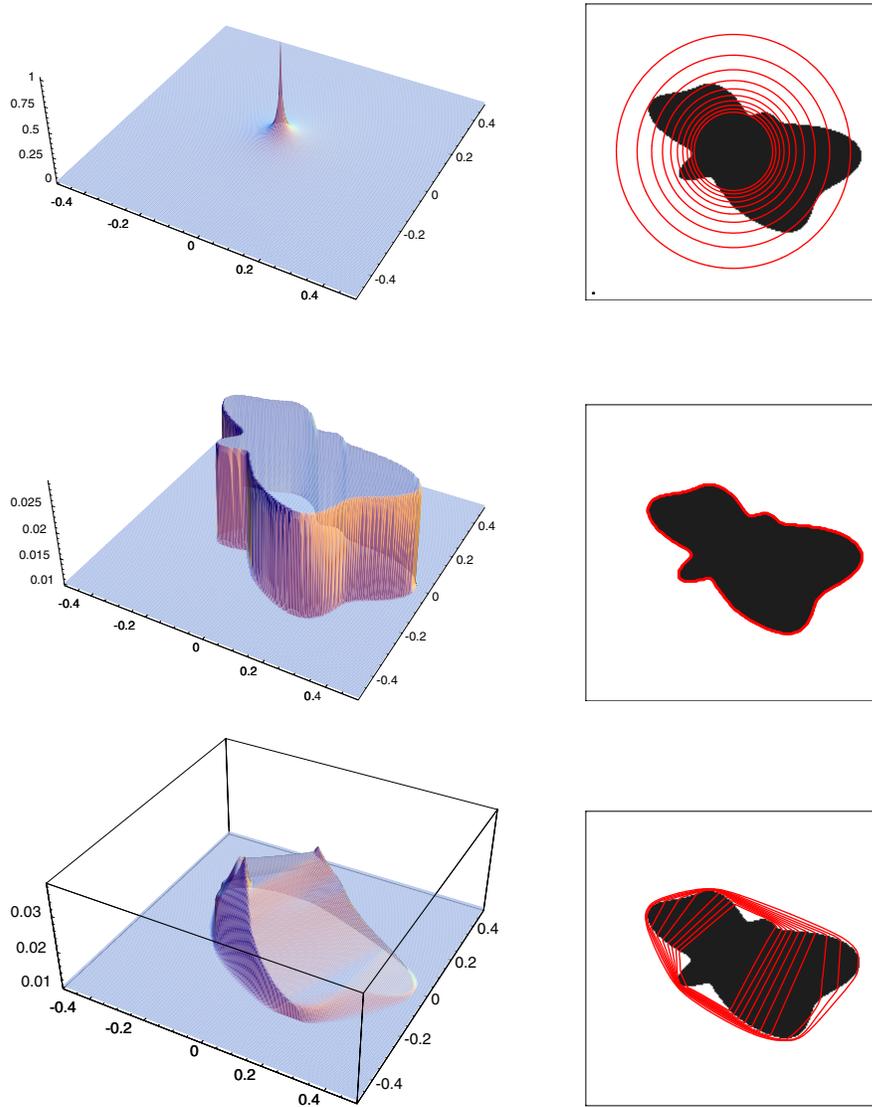


FIG. 3.9. Successful subjective surface based segmentation (1.4) of a “batman” image, in the middle. Nonsuccessful segmentation only by slowed mean curvature motion (1.5), in the bottom.

time is even speeding up depending on the number of available processors (e.g., in Linux cluster). The MPI parallelization of our algorithm is straightforward, because it uses classical structures of scientific computing methodology for solving partial differential equations and linear algebra problems (see, e.g., [3]).

3.3. Comparison with different models and methods. First, we show the difference between segmentation model (1.4) and (1.5) with g^0 instead of $g(|\nabla G_\sigma * u|)$; i.e., we document the role of convective term which appears in the model if g^0 is inside the divergence term. This comparison also shows that the method from [16] cannot be straightforwardly transferred from image smoothing to image segmentation problems.

Let us consider a simple nonconvex black object on a white background as plotted in Figure 3.9. If we start with the initial function plotted in the top left subfigure, isolines of which are plotted in the top right subfigure, and apply model (1.4), we end up with results as in the middle subfigures, while applying (1.5) we end up in the long run with the bottom subfigures. In the middle we see accumulation of all level lines on the object boundary due to the convective term, and any of these level lines can be used to represent segmentation of the object. In the case of (1.5), the motion of level lines is slowed down along the edges; however, level lines shrinking from outside, due to minimizing length (property of mean curvature motion), cannot go inside the nonconvex regions, and they accumulate on a convex envelope of the object.

Next we compare computational efficiency of our semi-implicit scheme with standard methods based on classical level set strategy and explicit time stepping. Such explicit schemes are usually based on formulation (1.4), where, moreover, all derivatives are expanded to get mean curvature and advection terms. Then, in three dimensions, (1.1) is written in the form

$$(3.7) \quad \begin{aligned} u_t = g^0 & ((\varepsilon + u_{x_2}^2 + u_{x_3}^2)u_{x_1x_1} + (\varepsilon + u_{x_1}^2 + u_{x_3}^2)u_{x_2x_2} + (\varepsilon + u_{x_1}^2 + u_{x_2}^2)u_{x_3x_3} \\ & - 2u_{x_1}u_{x_2}u_{x_1x_2} - 2u_{x_1}u_{x_3}u_{x_1x_3} - 2u_{x_2}u_{x_3}u_{x_2x_3}) / \\ & (\varepsilon + u_{x_1}^2 + u_{x_2}^2 + u_{x_3}^2) + g_{x_1}^0 u_{x_1} + g_{x_2}^0 u_{x_2} + g_{x_3}^0 u_{x_3}, \end{aligned}$$

where u_s means partial derivative of a function u with respect to a variable s . The explicit time stepping and central finite difference approximation is used in the mean curvature term, and upwind finite difference approximation is used in convective term [33]. For the computational comparison we use a representative 3D example where we segment a sphere with four holes (white object on black background in 80^3 voxels image); the 2D cut by equatorial plane is given in the upper left of Figure 3.10. In the upper right of Figure 3.10 we see the reconstructed sphere, and in the bottom part of Figure 3.10 we see the 2D cut of the shock profile of the segmentation function which is used to extract the upper sphere. This state of the segmentation function is achieved after 1450 time steps using discretization [33] of (3.7). Although CPU time for one explicit time update is small, due to stability reasons, both in curvature and convective parts, we have to choose a small time step, $\tau = 0.00002$, and need many of them to achieve a piecewise constant profile of the segmentation function. So the overall CPU time is 380 secs. In spite of that our semi-implicit discretization (2.15) fulfills unconditionally the discrete minimum-maximum principle (L_∞ -stability); i.e., no spurious oscillations appear for any length of discrete time step. We get the same 3D segmentation result in 28 steps with $\tau = 0.002$. Although one step of the semi-implicit scheme is more complex, we have to solve the linear system, the overall CPU time is 45 secs, and there is a benefit of almost 10 times speed-up. This speed-up ratio is quite a common observation in many 3D segmentation comparisons of semi-implicit and explicit schemes.

At the end we illustrate removing a grid orientation effect which can be observed in numerical results of previously studied algorithms [35, 16]. There either “left oriented” or “right oriented” triangulations (see Figure 3.11) have been used for solving the level-set-like problems, and, especially on coarse grids, the influence of grid orientation can be observable. To show this phenomenon in a transparent way we present a 2D experiment where the initial unit circle is moving by mean curvature and expanding with unit speed in the outer normal direction at the same time. In such a case, $Fm(p)$, $F = -1$ is added to the right-hand side of the fully discrete scheme (2.15). Since there is a balance between curvature and driving force the unit circle is the

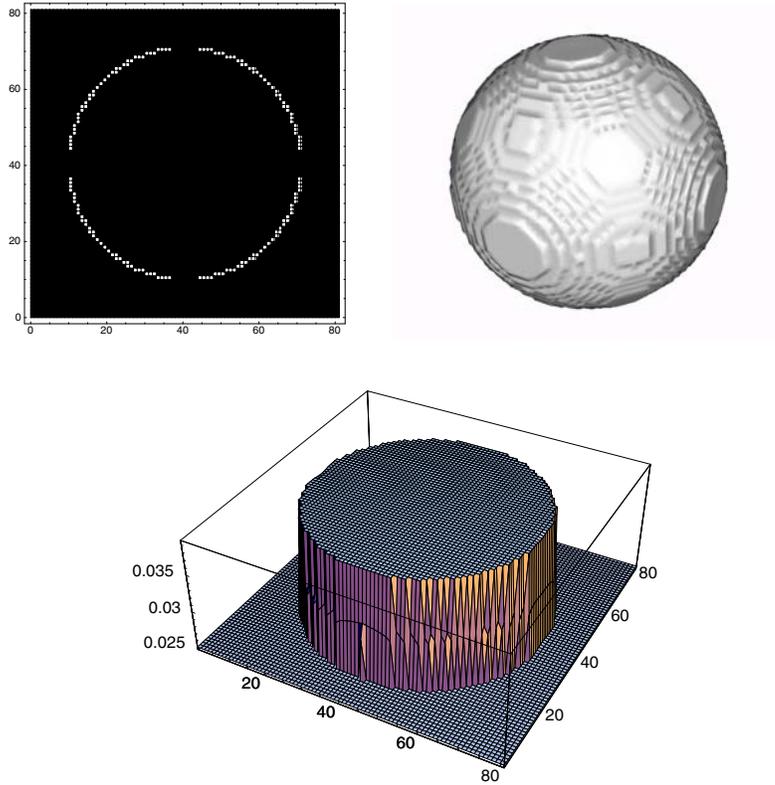


FIG. 3.10. Subjective surface based segmentation of a 3D sphere with four holes.

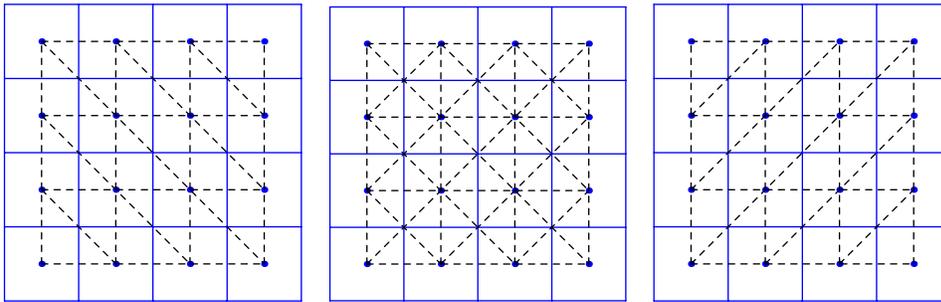


FIG. 3.11. By dashed lines we plot the “left oriented” triangulation (left), “right oriented” triangulation (right), and “symmetric” triangulation corresponding to our method in two dimensions. The image pixels (solid lines) in all cases correspond to the covolume mesh.

exact solution, and so it should remain stable also computationally. Using the “right oriented” triangulation with 20×20 covolumes and the method from [16], at time $T = 1$, we get the result plotted in Figure 3.12(left). We see exact solution plotted by the dashed line and its slight distortion visualizing the numerical solution. On the other hand, using the 2D analogy of the method presented in this paper, i.e., “symmetric” triangulation plotted in the middle part of Figure 3.11, we get the numerical result presented in Figure 3.12(right), where the circular shape is not distorted. Our 3D tetrahedral mesh and covolume grid construction transfer such “symmetry” of the

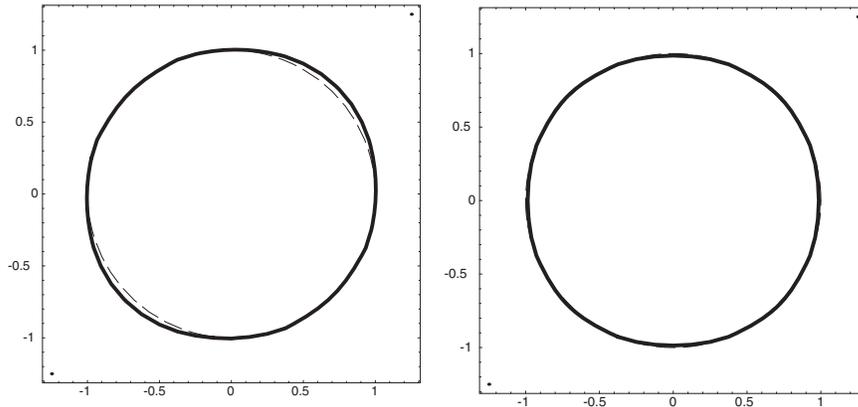


FIG. 3.12. By the dashed line we plot the exact solution (stable unit circle). By the thick solid line we plot the level line obtained from numerical solution using the “right oriented” triangulation (left) and the “symmetric” triangulation corresponding to our method in two dimensions. In the latter case the solutions coincide in this visualization scale.

method into the 3D case, and in this sense the previous covolume techniques are improved.

Acknowledgement. We thank Roberto Gori from CINECA for help with 3D visualizations.

REFERENCES

- [1] L. ALVAREZ, F. GUICHARD, P. L. LIONS, AND J. M. MOREL, *Axioms and fundamental equations of image processing*, Arch. Rational Mech. Anal., 123 (1993), pp. 199–257.
- [2] L. ALVAREZ, P.-L. LIONS, AND J.-M. MOREL, *Image selective smoothing and edge detection by nonlinear diffusion. II*, SIAM J. Numer. Anal., 29 (1992), pp. 845–866.
- [3] Y. AOYAMA AND J. NAKANO, *RS/6000 SP: Practical MPI Programming*, IBM, <http://www.redbooks.ibm.com> (1999).
- [4] E. BÄNSCH AND K. MIKULA, *A coarsening finite element strategy in image selective smoothing*, Comput. Vis. Sci. 1 (1997), pp. 53–61.
- [5] V. CASELLES, F. CATTÉ, T. COLL, AND F. DIBOS, *A geometric model for active contours in image processing*, Numer. Math., 66 (1993), pp. 1–31.
- [6] V. CASELLES, R. KIMMEL, AND G. SAPIRO, *Geodesic active contours*, Int. J. Comput. Vis., 22 (1997), pp. 61–79.
- [7] V. CASELLES, R. KIMMEL, G. SAPIRO, AND C. SBERT, *Minimal surfaces: A geometric three dimensional segmentation approach*, Numer. Math., 77 (1997), pp. 423–451.
- [8] F. CATTÉ, P.-L. LIONS, J.-M. MOREL, AND T. COLL, *Image selective smoothing and edge detection by nonlinear diffusion*, SIAM J. Numer. Anal., 29 (1992), pp. 182–193.
- [9] S. L. CHAN AND E. O. PURISIMA, *A new tetrahedral tessellation scheme for isosurface generation*, Computers and Graphics, 22 (1998), pp. 83–90.
- [10] Y.-G. CHEN, Y. GIGA, AND S. GOTO, *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equation*, J. Differential Geom., 33 (1991), pp. 749–786.
- [11] G. CITTI AND M. MANFREDINI, *Long time behavior of Riemannian mean curvature flow of graphs*, J. Math. Anal. Appl., 273 (2002), pp. 353–369.
- [12] M. G. CRANDALL, H. ISHII, AND P. L. LIONS, *User’s guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc. (N.S.), 27 (1992), pp. 1–67.
- [13] K. DECKELNICK AND G. DZIUK, *Error estimates for a semi implicit fully discrete finite element scheme for the mean curvature flow of graphs*, Interfaces Free Bound., 2 (2000), pp. 341–359.
- [14] K. DECKELNICK AND G. DZIUK, *Numerical approximation of mean curvature flow of graphs and level sets*, in Mathematical Aspects of Evolving Interfaces, L. Ambrosio, K. Deckelnick, G. Dziuk, M. Mimura, V. A. Solonnikov, and H. M. Sonner, eds., Springer, Berlin, Heidelberg, New York, 2003, pp. 53–87.

- [15] L. C. EVANS AND J. SPRUCK, *Motion of level sets by mean curvature I*, J. Differential Geom., 33 (1991), pp. 635–681.
- [16] A. HANDLOVIČOVÁ, K. MIKULA, AND F. SGALLARI, *Semi-implicit complementary volume scheme for solving level set like equations in image processing and curve evolution*, Numer. Math., 93 (2003), pp. 675–695.
- [17] J. KAČUR AND K. MIKULA, *Solution of nonlinear diffusion appearing in image smoothing and edge detection*, Appl. Numer. Math., 17 (1995), pp. 47–59.
- [18] G. KANIZSA, *Organization in Vision*, Praeger, New York, 1979.
- [19] M. KASS, A. WITKIN, D. AND TERZOPULOS, *Snakes: Active contour models*, Int. J. Comput. Vis., 1 (1987), pp. 321–331.
- [20] S. KICHENASSAMY, A. KUMAR, P. OLVER, A. TANNENBAUM, AND A. YEZZI, *Conformal curvature flows: From phase transitions to active vision*, Arch. Rational Mech. Anal., 134 (1996), pp. 275–301.
- [21] R. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, UK, 2002.
- [22] R. MALLADI, J. A. SETHIAN, AND B. VEMURI, *Shape modeling with front propagation: A level set approach*, IEEE Trans. Pattern Anal. Machine Intell., 17 (1995), pp. 158–174.
- [23] K. MIKULA AND N. RAMAROSY, *Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing*, Numer. Math., 89 (2001), pp. 561–590.
- [24] K. MIKULA AND D. ŠEVČOVIČ, *Computational and qualitative aspects of evolution of curves driven by curvature and external force*. Comput. Vis. Sci., 6 (2004), pp. 211–225.
- [25] D. MUMFORD AND J. SHAH, *Optimal approximation by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
- [26] A. M. OBERMAN, *A convergent monotone difference scheme for motion of level sets by mean curvature*, Numer. Math., 99 (2004), pp. 365–379.
- [27] S. OSHER AND R. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2003.
- [28] S. OSHER AND J. A. SETHIAN, *Front propagating with curvature dependent speed: Algorithms based on the Hamilton–Jacobi formulation*, J. Comput. Phys., 79 (1988), pp. 12–49.
- [29] P. PERONA AND J. MALIK, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Machine Intell., 12 (1990), pp. 629–639.
- [30] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, MA, 1996; reprinted, SIAM, Philadelphia, 2003.
- [31] A. SARTI, R. MALLADI, AND J. A. SETHIAN, *Subjective surfaces: A method for completing missing boundaries*, Proc. Natl. Acad. Sci. USA, 97 (2000), pp. 6258–6263.
- [32] A. SARTI AND G. CITTI, *Subjective surfaces and Riemannian mean curvature flow graphs*, Acta Math. Univ. Comenian., (N.S.), 70 (2000), pp. 85–103.
- [33] A. SARTI, R. MALLADI, AND J. A. SETHIAN, *Subjective surfaces: A geometric model for boundary completion*, Int. J. Comput. Vis., 46 (2002), pp. 201–221.
- [34] J. A. SETHIAN, *Level Set Methods and Fast Marching Methods, Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, UK, 1999.
- [35] N. J. WALKINGTON, *Algorithms for computing motion by mean curvature*, SIAM J. Numer. Anal., 33 (1996), pp. 2215–2238.