FINITE ELEMENT METHODS ON VERY LARGE, DYNAMIC TUBULAR GRID ENCODED IMPLICIT SURFACES

OLIVER NEMITZ*, MICHAEL BANG NIELSEN†, MARTIN RUMPF‡, AND ROSS WHITAKER§

Abstract. The simulation of physical processes on interfaces and a variety of applications in geometry processing and geometric modeling are based on the solution of partial differential equations on curved and evolving surfaces. Frequently, an implicit level set type representation of these surfaces is the most effective and computationally advantageous approach. This paper addresses the computational problem of how to solve partial differential equations on highly resolved level sets with an underlying very high-resolution discrete grid. These high-resolution grids are represented in a very efficient Dynamic Tubular Grid encoding format for a narrow band. A reaction diffusion model on a fixed surface and surface evolution driven by a nonlinear geometric diffusion approach, by isotropic, or truly anisotropic curvature motion are investigated as characteristic model problems. The proposed methods are based on semi-implicit finite element discretizations directly on these narrow bands, require only standard numerical quadrature and allow for large time steps. To combine large time steps with a very thin and thus storage inexpensive narrow band, suitable transparent boundary conditions on the boundary of the narrow band and a nested iteration scheme in each time step are investigated. This nested iteration scheme enables the discrete interfaces to move in a single time step significantly beyond the domain of the narrow band of the previous time step. Furthermore, algorithmic tools are provided to assemble finite element matrices and to apply matrix vector operators via fast, cache-coherent access to the Dynamic Tubular Grid encoded data structure. The consistency of the presented approach is evaluated and various numerical examples show its application potential.

Key words. level set methods, narrow band approach, partial differential equations on surfaces, curvature motion

AMS subject classifications. 65D18, 65D10, 65M50 65M60 65N30, 65N50, 68P05, 68P20

1. Introduction.

This paper addresses the computational problem of how to solve partial differential equations (PDEs) on the highly resolved level sets of smooth scalar functions [1]. The context for this work is the growing interest in computing PDEs on surfaces that are represented implicitly as level sets

$$[\phi = c] := \{x \in \Omega \,|\, \phi(x) = c\}$$

of a smooth scalar function ϕ on a domain Ω. Starting with the pioneering paper by Osher and Sethian [49] this approach has become increasingly important in a variety of fields such as computational physics [3,5,8,12,30], scientific visualization [38], image analysis [6,14], and computer graphics [43,48]. Most of these applications rely on the efficient computation of partial differential equations on curves or surfaces implicitly represented by a level set function ϕ. Thereby, ϕ is discretized on a discrete, usually structured, grid. The attraction of solving problems with discretely sampled implicit surfaces is the relatively large number of degrees of freedom provided by the grid and the freedom of not having to choose an explicit surface parametrization, which often limits shape and topology.

*University of Bonn, Germany, (oliver.nemitz@ins.uni-bonn.de).

†University of Århus, Denmark, (bang@daimi.au.dk).

‡University of Bonn, Germany, (martin.rumpf@ins.uni-bonn.de).

§University of Utah, U.S.A., (whitaker@cs.utah.edu).

[1] A preliminary form of this paper was published in an 8 page workshop proceedings publication: *O. Nemitz, M. B. Nielsen, M. Rumpf, and R. Whitaker*, "Narrow band methods for pdes on very large implicit surfaces", *Vision, Modeling and Visualization Proceedings*, 2007.

There are in particular two scenarios in which such surface-based PDEs are interesting. The first is when the implicit surface serves as the domain and one would like to solve a PDE for a function u intrinsic on the surface. Examples are the spreading of thin liquid films or coatings on surfaces [53], and reaction diffusion equations for texture generation on surfaces [62]. Projections of the derivatives in the ambient space onto the surface provide a mechanism for computing differential operators that live on the implicit surface [6]. A corresponding finite element approach is discussed in [9]. Finite elements on narrow bands are investigated in [18] and in [29] an improved approximation of tangential differential operators is presented. Furthermore, in [25] a finite element level set method is introduced for the solution of parabolic PDEs on moving surfaces. The other scenario is when the surface itself evolves according to a geometric PDE that depends on the shape. The most prominent example is motion by mean curvature [27]. For the discretization in space either finite difference [49, 55] or finite element schemes [19] are considered. Semi-implicit time discretizations are suitable due to their stability properties for large time steps. Thus they outperform explicit time discretization for diffusion type problems which require much smaller time steps. This is particularly important when one is considering higher order PDEs [23, 30].

In many applications in material science, biology and in geometric modeling PDE models are considered on curved surfaces, which frequently evolve themselves in time. Here, examples are transport and diffusion of a surfactant on interfaces in multiphase flow [36], surfactant driven thin film flow [32] coupled with surface evolution [57], e.g. a lubrication model on the enclosed membrane of lung alveoli coupled with the expansion or contraction of the alveoli [57] and diffusion induced grain boundary motion [10].

Perhaps the greatest promise of level-set methods, for both moving interfaces and PDEs defined on static surfaces (codimension one), is their ability to deal with a wide variety of complicated shapes in an elegant manner within a single computational framework. However, the computation and memory requirements on the discrete grid that represents ϕ become prohibitive as the grid resolution increases. The complexity of the surface increases (roughly) as the grid resolution squared, but the overall grid size increases with the cube of the resolution.

Several technical advances have addressed different aspects of the problem associated with storing level sets and computing level-set equations at high resolutions. The introduction of methods that solve PDEs on a narrow band around the surface [2, 18, 50, 65] provided significant advantages in computation time. As grid sizes become progressively larger the number of computations in the narrow band is not the limiting factor on performance. Rather, the performance of computations is limited by the very small fraction of the narrow band values that can fit simulataneously into cache or random-access memory as well as the number and pattern of accesses to these values. To address this issue several authors have proposed memory-efficient data structures for storing narrow bands associated with level sets.

The use of such narrow bands, which can encode many millions of degrees of freedom, gives the level-set approach to surface representation a *distinct computational advantage* relative to parametric representations, such as triangle meshes. The reason is that with careful attention to how grid points are stored and accessed, the grid-based, implicit method for processing surfaces provides regular, predictable access to memory in a way that allows for cache coherency (on conventional processors) and data streaming on more advanced architectures.

This very thin computational domain presents a challenge for numerical schemes, however, because one must introduce a solution for the PDE along the boundary domain, whose shape can be quite irregular. As the resolution increases the boundaries of the computational domain come progressively closer to the level set of interest, and the so called *natural* boundary conditions allow artifacts from the grid (whose faces are aligned with the cardinal directions) to propagate into the PDE on the surface. Furthermore, when solving free boundary problems by a standard approach, the time steps must be limited so that at each iteration the moving interface (level set of interest) is neither impeded by the boundary conditions nor allowed to pass outside of the computational domain (at which point its shape is lost).

This work also builds on the research in computer science on efficient data structures for storing sparse computational domains associated with level sets. In recent years quadtrees (2D) and octrees (3D) [15] have been applied to level sets in numerous papers [22, 26, 33, 40–42, 58–61]. The pointer-based quadtree and octree data structures reduce the storage requirements of level sets to $O((d+1)n)$, but also introduce an $O(d)$ access time, where d is the depth of the quadtree or octree and n is the number of grid points in the narrow band. Note that it may be the case that $d >> \log n$. The octree data structure can be modified to reduce storage requirements to $O(n)$ and access time to $O(\log n)$ (see [15]). This access time is nevertheless still penalizing in the context of the level set method, and state-of-the-art octree-traversal and search methods [28, 56] utilize bit-arithmetic that cannot immediately be used in conjunction with these modifications. The method of Losasso et al. [40] addresses some of the performance issues associated with octrees. Instead of using a traditional octree they propose to use a coarse uniform grid in which each grid cell stores an octree of its own. This decouples the depth of the octree from the size of the computational domain and hereby lowers the depth d. In addition they introduce an iterator construct that speeds up access locally during interpolation for semi-Lagrangian advection. Unfortunately a comparative study of the practical performance of this method has not been documented. Furthermore, no method has been published on how to ensure cache coherency in the octree storage format as the narrow band changes due to the temporal evolution of the level set. Cache coherency becomes increasingly important in high resolution. The Dynamic Tubular Grid (DT-Grid) [46] employs a hierarchical encoding of the topology of the narrow band, inspired by the storage-format of sparse matrices. Similar and derivative works employ a run-length encoding, and focus either on flexibility [34] or are tailored for a specific application in fluid simulation [35]. All of these data structures require $O(n)$ storage and have $O(1)$ access time to grid points in a local stencil during the sequential access typically required by level set methods. Furthermore these data structures have been shown to perform faster in practice than recent narrow band and octree approaches due both to the lower memory footprint and the more cache coherent memory layout and access patterns [34]. In this work we utilize the DT-Grid since it has been shown to perform slightly faster and require less memory than the run-length encoding alternatives.

In contrast to the finite difference schemes already implemented in this context we consider here finite element methods with a semi-implicit discretization approach in time and introduce the required suitable DT-Grid based linear algebra operations on finite element matrices. We introduce transparent boundary conditions together with nested iterations in time that decrease the impact of the irregular narrow band boundary on the quality of the discrete solution. In the case of moving interfaces, this enables semi-implicit updates with large time steps that do not restrict the updated solution to the thin computational domain from the previous time step.

As applications we consider texture synthesis via a reaction diffusion model, anisotropic fairing of surfaces, and the evolution of surfaces by mean curvature motion or by truly anisotropic curvature motion. These models are applied to very large data sets that are appropriate for state-of-the-art applications in surface processing.

2. Finite Element Discretization on Narrow Bands.

In this section we will treat a simple scalar reaction diffusion equation on a fixed surface and classical curvature motion as a role model for a geometric evolution problem. We will recall in both cases a finite element discretization in the level set context and introduce the fundamental notation involved then restricting the computation to a narrow band around the surface of interest. We will rely on the notation introduced here in the later description of our improved narrow band algorithms. Several difficulties immediately present themselves with this formulation. The first is the choice of proper boundary conditions on the boundary of the narrow band which do not interfere too much with the solution on the actual surface. The second issue is the manipulation of the narrow band itself, which must be rebuilt as the actual surface gets too close to the boundary. Addressing these problems, which we do in the next sections, is one of the main contributions of the paper.

2.1. A Reaction Diffusion Model on Level Sets.

As a model for a reaction diffusion process on a fixed, closed surface $\mathcal{M} \subset \mathbb{R}^3$ we consider the following scalar initial value problem: Find a function $u : \mathbb{R}^+ \times \mathcal{M} \to \mathbb{R}$, such that

$$\partial_t u - \Delta_{\mathcal{M}} u = f(u) \tag{2.1}$$

with initial condition $u(0) = u^0$, where u^0 is some initial value function on the surface $\mathcal{M}$. Here $\Delta_{\mathcal{M}}$ is the Laplace Beltrami operator on $\mathcal{M}$. The case of multiple species and applications to texture generation are detailed on Section 4.1. Let us suppose that $\mathcal{M}$ can be represented as the zero level set of a function $\phi : \Omega \to \mathbb{R}$, where Ω is a box domain enclosing $\mathcal{M}$. Then the Laplace Beltrami operator can be written in level set form and we obtain $\Delta_{\mathcal{M}} u = |\nabla\phi|^{-1}\mathrm{div}(|\nabla\phi| P[\phi]\nabla u)$ [6], where $P[\phi] = \mathbb{1} - \frac{\nabla\phi}{|\nabla\phi|} \otimes \frac{\nabla\phi}{|\nabla\phi|}$ at a point x on $\mathcal{M}$ is the projection onto the tangent space $T_x\mathcal{M}$. Now, we first discretize in time and introduce a time derivative $\frac{u^{k+1}-u^k}{\tau}$ for functions u^k at time $k\,\tau$. Testing equation (2.1) with a smooth function ϑ and applying integration by parts we derive the following time discrete weak formulation:

$$\int_\Omega |\nabla\phi| \frac{u^{k+1} - u^k}{\tau}\, \vartheta + |\nabla\phi| P[\phi] \nabla u^{k+1} \cdot \nabla\vartheta \;\mathrm{d}x = \int_\Omega |\nabla\phi| f(u^k)\, \vartheta \;\mathrm{d}x \tag{2.2}$$

for all test functions $\vartheta \in C^1$. Here the nonlinear right hand side f is evaluated on the old time step. The operator $P[\phi]$ ensures a decoupling of the reaction diffusion process on different level sets $[\phi = c]$, which reflects the geometric nature of the problem (2.1). This weak formulation directly implies natural boundary conditions $P[\phi]\nabla u \cdot \nu = 0$ on $\partial\Omega$, where ν is the outer normal on $\partial\Omega$.

A convergence theory for narrow band level set methods based on unfitted Finite Elements is given in [18]. For the numerical solution of general ellipic equations on implicit surfaces with Finite Elements see also [9]. To identify the solution on $\mathcal{M}$ it suffices to consider the weak formulation restricted to a small band around $\mathcal{M}$. In case this band consists of level sets $[\phi = c]$ for level set values c from an interval $(-\delta, \delta)$ the above boundary condition is fulfilled for any u because $P[\phi]\nu$ vanishes on the narrow band boundary.

Next, we discretize in space based on a finite element approximation. At first, let us fix some notation. In what follows, we denote discrete quantities with upper case letters to distinguish them from continuous quantities in lower case letters. The domain Ω is supposed to be covered by a regular hexahedral grid and we denote the corresponding space of continuous, piecewise tri-linear functions by $\mathcal{V}^h$, where h indicates the grid size. Let $\{\Theta_i\}_{i\in\mathbf{I}}$ be the canonical nodal basis of this finite element space for an index set $\mathbf{I}$ corresponding to all grid nodes. A discrete function U is represented as a nodal vector $\bar{U} = (U_i)_{i\in\mathbf{I}}$, where $U = \sum_{i\in\mathbf{I}} U_i\Theta_i$ is the corresponding discrete function. Now, given an approximation $\Phi \in \mathcal{V}_h$ of the level set function ϕ, we obtain an approximation $\mathcal{M}^h := [\Phi = 0]$ of the continuous surface $\mathcal{M}$ as one particular discrete level set represented by the function Φ. Concerning the reaction diffusion model, we replace all continuous quantities in (2.2) by their discrete counterparts and introduce mass lumping. We define the weighted lumped mass and stiffness matrix explicitly as

$$\mathbf{M}[\Phi] = \left(\int_\Omega \mathcal{I}_h^0(|\nabla\Phi|)\mathcal{I}_h^1(\Theta_i\Theta_j)\ \mathrm{d}x\right)_{i,j\in\mathbf{I}}, \quad \mathbf{L}[\Phi] = \left(\int_\Omega |\nabla\Phi| P[\Phi]\nabla\Theta_i\cdot\nabla\Theta_j\ \mathrm{d}x\right)_{i,j\in\mathbf{I}},$$

where $\mathcal{I}_h^0$, $\mathcal{I}_h^1$ denote the piecewise constant and the piecewise multilinear Lagrangian projection, respectively. Furthermore, we introduce the right hand side vector $\bar{F}[U] = (f(U_i))_{i\in\mathbf{I}}$ and end up with the system of linear equations

$$\left(\mathbf{M}[\Phi] + \tau\mathbf{L}[\Phi]\right)\bar{U}^{k+1} = \mathbf{M}[\Phi]\left(\tau\bar{F}[U^k] + \bar{U}^k\right).$$

Solving this system allows us to iteratively compute a discrete solution U^k at times $t_k = k\tau$ for $k \geq 1$ given an approximation U^0 of u^0. For a narrow band solution strategy to be discussed here we observe in the discrete model different from the continuous counterpart a coupling of the solution on different nearby level sets due to the overlapping support of the basis functions. This leads to an interference of the solution on $\mathcal{M}_h$ with the boundary condition on the boundary of the narrow band and is in particular crucial for a jaggy narrow band boundary consisting of facets of grid cells. To resolve this difficulty, one either can modify the narrow band domain and the corresponding basis functions, such that the narrow band turns into a union of discrete level sets, which has recently been proposed by Deckelnick et al. [18], or one keeps the rectangular cell geometry and the corresponding basis function, but introduces suitable "transparent" boundary conditions - an approach to be investigated in this paper.

2.2. Curvature Motion of Level Sets. The second model problem is the evolution of surfaces under mean curvature motion. Given an initial surface $\mathcal{M}_0$ we ask for a family of surfaces $\{\mathcal{M}(t)\}_{t\geq 0}$ generated from the motion of points $x(t)$ under the evolution

$$\dot{x}(t) = -h(t)n(t) \tag{2.3}$$

with initial condition $x(0) = x_0$ with $x_0 \in \mathcal{M}_0$. Here $n(t)$ is the normal and $h(t)$ the mean curvature on $\mathcal{M}(t)$. Given the normal velocity v of a moving surface the level set equation $\partial_t\phi + |\nabla\phi|v = 0$ [49] allows to encode the surface motion implicitly in a family of level sets $\mathcal{M}(t) = [\phi(t,\cdot) = 0]$ for a level set function $\{\phi(t)\}_{t\geq 0}$ in space and time. An investigation of the viscosity solution approach to mean curvature motion can be found in [27]. The convergence of finite element solutions to the

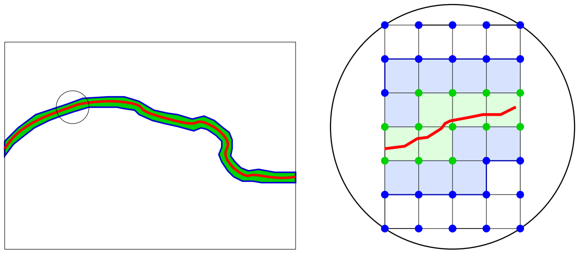


Fig. 2.1: A sketch of a narrow band domain Ω_n corresponding to a level set (plotted in red) is shown. In the blowup on the right interior nodes are indicated by green dots, whereas the boundary $\partial\Omega_\mathrm{n}$ is represented by blue lines.

viscosity solution for mean curvature motion is discussed in [16]. In our case, $v = h = \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$. Hence, we end up with the level set equation of mean curvature motion $\partial_t\phi - |\nabla\phi|\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) = 0$ on $\mathbb{R}^+ \times \Omega$ with initial data ϕ_0. Again discretizing in time and applying integration by parts we obtain the weak formulation

$$\int_\Omega \frac{\phi^{k+1} - \phi^k}{\tau|\nabla\phi^k|}\vartheta + \frac{\nabla\phi^{k+1}}{|\nabla\phi^k|}\cdot\nabla\vartheta \,\mathrm{d}x = 0 \tag{2.4}$$

for sufficiently regular test functions ϑ, where we take into account the old time step solution for the weight $|\nabla\phi^k|^{-1}$. As the resulting natural boundary condition on $\partial\Omega$ we obtain $\nabla\phi\cdot\nu = 0$. Thus, the level sets are forced to be perpendicular on $\partial\Omega$. Now, aiming for a fully practical numerical algorithm we discretize in space and introduce the usual regularization $|x|_\epsilon = \sqrt{\epsilon^2 + |x|^2}$ for the norm $|\nabla\phi^k|$ in the denominator and replace it by $|\nabla\phi^k|_\epsilon$ [27]. Thus, we end up with a sequence of linear systems of equations

$$\left(\mathbf{M}[\Phi^k] + \tau\mathbf{L}[\Phi^k]\right)\bar{\Phi}^{k+1} = \mathbf{M}[\Phi^k]\bar{\Phi}^k \tag{2.5}$$

for the nodal vector $\bar{\Phi}^{k+1}$ of the discrete level set function at time $t_{k+1} = \tau(k+1)$. Here, the involved lumped mass and stiffness matrices are given by

$$\mathbf{M}[\Phi] = \left(\int_\Omega \mathcal{I}_h^0(|\nabla\Phi|_\epsilon^{-1})\mathcal{I}_h^1(\Theta_i\Theta_j)\,\mathrm{d}x\right)_{i,j\in\mathbf{I}}, \tag{2.6}$$

$$\mathbf{L}[\Phi] = \left(\int_\Omega |\nabla\Phi|_\epsilon^{-1}\nabla\Theta_i\cdot\nabla\Theta_j\,\mathrm{d}x\right)_{i,j\in\mathbf{I}}. \tag{2.7}$$

Solving these systems iteratively for a given approximation Φ^0 of ϕ^0 we obtain a sequence of discrete surfaces $\mathcal{M}_h^k = [\Phi^k = 0]$. As before, we might restrict the computational domain to a - in this case moving - narrow band around the moving family of surfaces. Now we have to carefully update this computational domain in every time step and to resolve the issue of boundary conditions.

2.3. Transparent Neumann Boundary. The continuous formulation operates on the solution on each level-set separately. Thus, solutions from different level sets do not interact and we can truncate the computational domain to a narrow band around the zero set without affecting the solution on this zero level set. However,

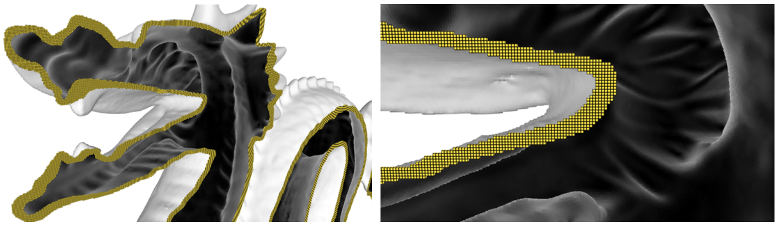


Fig. 2.2: Visualization of the minimal narrow band: We intersect the dragon surface with a plane and draw all those elements that touch this plane. At some points the narrow band seems to be rather thick, but appearances are deceiving. Such an impression is created if the surface is nearly parallel to the clip-plane, such that more cells intersect the plane.

the discrete formulation introduces a coupling of nearby level sets through a coupling of the contributions of different discrete basis functions. Hence, in case of a very thin band the natural boundary conditions induced by the weak formulation interfere strongly with the solution on $\mathcal{M}_h$. This interaction undermines the numerical convergence of the scheme (cf. Table 2.1 below). In this section we will describe boundary conditions which try to avoid this interference.

Given a level set surface $\mathcal{M}_h$ for a level set function $\Phi \in \mathcal{V}^h$ we define discrete narrow bands of varying width as a union of supports of discrete basis functions. Hence, for the thinnest band we consider a corresponding index set

$$\mathbf{I}^1_{\text{int}} := \{ i \in \mathbf{I} \,|\, \operatorname{supp}\Theta_i \cap \mathcal{M}^h \neq \emptyset \}$$

and the resulting narrow band domain $\Omega^1_{\text{n}} = \bigcup_{i\in\mathbf{I}_{\text{int}}} \operatorname{supp}\Theta_i$. A sketch of this narrow band is given in Figure 2.1. This is the smallest possible band which allows to resolve the discrete surface $\mathcal{M}_h$ in the sense that the description of the surface requires nodal values of all basis functions considered in the definition of the band.

Given this narrow band we can successively increase the width of the band and iteratively define

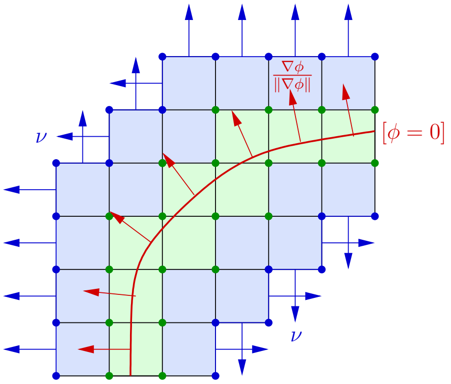
$$\mathbf{I}^{j+1}_{\text{int}} := \{ i \in \mathbf{I} \,|\, \exists k \in \mathbf{I}^j \text{with} \operatorname{supp}\Theta_i \cap \operatorname{supp}\Theta_k \neq \emptyset \}\,.$$

The corresponding narrow band domain is given by

$$\Omega^j_{\text{n}} = \bigcup_{i\in\mathbf{I}^j_{\text{int}}} \operatorname{supp}\Theta_i\,. \tag{2.8}$$

In what follows, we will skip the index indicating the width of the narrow band if it is clear from the context. Let us denote by $\mathcal{V}^h_{\text{int}} = \operatorname{span}\{\Theta_i \,|\, i \in \mathbf{I}_{\text{int}}\}$ the space of discrete functions on the narrow band domain Ω_{n} which vanish on the boundary $\partial\Omega_{\text{n}}$ and by $\mathcal{V}_{\text{bd}} = \operatorname{span}\{\Theta_i \,|\, i \in \mathbf{I} \setminus \mathbf{I}_{\text{int}}\,,\ \operatorname{supp}\Theta_i \cap \Omega_{\text{n}} \neq \emptyset\}$ the discrete function space corresponding to boundary values on $\partial\Omega_{\text{n}}$. Hence, the direct sum $\mathcal{V}^h_{\text{n}} = \mathcal{V}^h_{\text{int}} \oplus \mathcal{V}^h_{\text{bd}}$ represents the discrete finite element space corresponding to the narrow band domain Ω_{n}. Now, we replace the domain of integration in the weak formulations by the narrow band domain.

For the *reaction diffusion model* integration by parts leads to the boundary integral $\int_{\partial\Omega_{\text{n}}} |\nabla\phi| P[\phi] \nabla u^{k+1} \cdot \nu \vartheta \, \mathrm{d}a$ on $\partial\Omega_{\text{n}}$, which gives rise to the Neumann boundary condition $P[\phi]\nabla u^{k+1} \cdot \nu = 0$.



This condition is automatically fulfilled in the continuous case where the boundary of the narrow band consists of level-sets that are parallel to the level set of interest. But in the discrete case the boundary of the narrow band consists of the grid aligned boundary segments of single grid cells. Prescribing natural boundary conditions $P[\phi]\nu \neq 0$ would consequently couple the gradient of the solution u^{k+1} with the grid-aligned, faceted boundary of the narrow band domain, which is obviously meaningless (cf. Fig. 2.1, 2.2 and the sketch on the right). Let us suppose that a good estimate u^{k+1}_{approx} of the solution u^{k+1} is given. Then, by adding the boundary integral $\int_{\partial\Omega_{\text{n}}} |\nabla\phi| P[\phi]\nabla u^{k+1}_{\text{approx}} \cdot \nu\vartheta \,\mathrm{d}a$ on the right hand side of the weak formulation in (2.2) we could compensate for this defect (cf. [21]). Hence, we obtain a modified time discrete weak formulation, namely

$$\int\limits_{\Omega_{\text{n}}} |\nabla\phi| \left(\frac{u^{k+1}-u^k}{\tau}\,\vartheta + P[\phi]\nabla u^{k+1}\cdot\nabla\vartheta - f(u^k)\vartheta \right) = \int\limits_{\partial\Omega_{\text{n}}} |\nabla\phi| P[\phi]\nabla u^{k+1}_{\text{approx}} \cdot \nu\vartheta \,\mathrm{d}a\,.$$

In case of the finite element discretization we add a corresponding correction vector

$$\bar\Gamma[U_{\text{approx}}] = \left(\int_{\partial\Omega_{\text{n}}} |\nabla\Phi| P[\Phi]\nabla U^{k+1}_{\text{approx}} \cdot \nu\Theta_j \,\mathrm{d}a \right)_{j\in\mathbf{I}} \tag{2.9}$$

for a given fully discrete approximation U^{k+1}_{approx} of U^{k+1} on the right hand side of the modified system of linear equations. In explicit, we get

$$\left(\mathbf{M}[\Phi] + \tau\mathbf{L}[\Phi]\right)\bar U^{k+1} = \mathbf{M}[\Phi]\left(\tau\bar F[U^k] + \bar U^k\right) + \tau\bar\Gamma[U^{k+1}_{\text{approx}}]\,. \tag{2.10}$$

Here $\bar F$ is the nodal vector in $\mathbb{R}^{\mathbf{I}_{\text{ext}}}$ corresponding to the right hand side f. A suitable choice for the approximation is given by the discrete solution at the last time step, i. e. we might set $U^{k+1}_{\text{approx}} = U^k$. A motivation for this correction is the following. If $\|\frac{\nabla u^{k+1} - \nabla u^k}{\tau}\|_{L^2(\partial\Omega_{\text{n}})}$ is uniformly bounded, the discussed defect of the Neumann boundary condition on the narrow band boundary for the modified model in (2.9) is bounded by $O(\tau)$. Table 2.1 compares the impact of this modified boundary condition on the convergence properties for the following test case with the explicitly known solution. We consider as a model problem the exponential decay of the amplitude of the second spherical harmonic eigenfunction on a sphere with radius $\frac{1}{3}$ if considered as initial data for the equation $\partial_t u - \Delta_{\mathcal{M}} u = 0$. We compute the L^2 difference between the exact solution and the discrete narrow band solution for different grid resolutions $(33^3, 65^3, 129^3 ...)$ for the computational domain $\Omega = (-\frac{1}{2}, \frac{1}{2})^3$ on the discrete surface at a fixed time $t_0 = 0.01$. To conclude, the straightforward approximation of PDEs on level set surfaces with thin, grid aligned and thus faceted bands and natural boundary conditions does not lead to consistent schemes. Transparent boundary conditions are an effective way to resolve this shortcoming and lead at least experimentally to a consistent numerical scheme. Already for the second thinnest possible band, we observed optimal, second order consistency. One observed in Table 2.1 that the error might stay nearly constant or even slightly increases from one grid resolution to the

grid size	17^3	33^3	65^3	129^3	257^3	513^3
	natural boundary conditions					
$j = 1$	0.008630	0.003052	0.002378	0.004136	0.004901	0.004986
$j = 2$	0.006465	0.001884	0.000911	0.000408	0.001375	0.001108
$j = 4$	–	0.001763	0.000407	0.000179	0.000181	0.000285
$j = 8$	–	–	0.000432	0.000109	1.98e-05	2.77e-05
	transparent boundary conditions					
$j = 1$	0.005976	0.000775	0.000320	0.000124	0.000134	7.91e-05
$j = 2$	0.006791	0.001828	0.000388	8.24e-05	0.000132	4.99e-05
$j = 4$	–	0.001745	0.000428	0.000128	1.86e-05	5.31e-05
$j = 8$	–	–	0.000432	0.000109	2.66e-05	8.73e-06

Table 2.1: We compare the L^2 error of the narrow band solution on the discrete interface for the heat equation on the sphere at a fixed time in case of natural boundary conditions (top rows) and the proposed transparent boundary conditions (bottom rows). This comparison is done for different grid size listed in the columns and different width of the involved narrow band (listed in the rows).

next finer one. This is possibly due to a particular alignment or misalignment of the discrete interface and the grid cells, which frequently will change while moving from a coarser to the next finer mesh.

For the discrete *mean curvature motion* we might proceed similarly. From (2.4) we deduce the natural boundary condition $\nabla\phi^{k+1}\cdot\nu = 0$ on Ω_{n}. Hence, level lines meet the boundary at right angles, which contradicts our intention to obtain bundles of nearly parallel level set surfaces on the narrow band. Given an approximation $\phi^{k+1}_{\text{approx}}$ of the time discrete solution ϕ^{k+1}, we again compensate for this defect adding a corresponding boundary integral. Indeed, for the spatial finite element discretization we consider $\bar{\Gamma}[\Phi^{k+1}_{\text{approx}}, \Omega_{\text{n}}] = \left(\int_{\partial\Omega_{\text{n}}} |\nabla\Phi^k|_\epsilon^{-1}\nabla\Phi^{k+1}_{\text{approx}}\cdot\nu\Theta_j \,\mathrm{d}a\right)_{j\in\mathbf{I}}$ as a correction vector in (2.5) and obtain the modified system

$$\left(\mathbf{M}[\Phi^k] + \tau\mathbf{L}[\Phi^k]\right)\bar{\Phi}^{k+1} = \mathbf{M}[\Phi^k]\bar{\Phi}^k + \tau\bar{\Gamma}[\Phi^{k+1}_{\text{approx}}, \Omega_{\text{n}}] \tag{2.11}$$

to be solved in each time step. If the time step is large compared to the width of the narrow band, it may happen that the new zero level set is no longer completely inside the narrow band and our time stepping scheme breaks down. To overcome this difficulty would require sophisticated a priori estimates for the expected position of the zero level set and correspondingly thicker narrow bands. Alternatively, we can apply an appropriate thin narrow band approach based on Dirichlet boundary conditions and an adaptation of the narrow band in an inner loop for each time step. The next section is devoted to this approach.

2.4. Transparent Dirichlet Boundary Conditions and Nested Iterations. If Dirichlet boundary conditions are considered for a discrete time step of mean curvature motion on a narrow band domain, the proper selection of the Dirichlet data has to anticipate the new position of the zero level set. Otherwise the interface motion might be hampered (cf. Table 2.2). Indeed, due to the maximum principle the new zero level set will stay in the narrow band as long as we assume different signs of the boundary values on both sides of the interface. Hence, we ask for a suitable approximation $\phi^{k+1}_{\text{approx}}$ of the unknown level set function ϕ^{k+1} defined on the boundary of the narrow band.

Our ansatz is to improve this approximation in an inner iteration for each time step.

In this inner iteration we also adjust the narrow band domain itself, which allows the evolving surface to move beyond the boundary of the current narrow band. These are the two key ingredients for an efficient narrow band approach, which confines with thin bands even in case of highly resolved level set surfaces.
Before we detail this inner iteration let us briefly comment on some basic implementation issues of the Dirichlet boundary condition. To unroll the discrete scheme in matrix vector notation, we exploit the introduced splitting of the finite element space $\mathcal{V}^h_{\mathrm{n}} = \mathcal{V}^h_{\mathrm{int}} \oplus \mathcal{V}^h_{\mathrm{bd}}$. Thus, reordering degrees of freedom we obtain a decomposition $\bar{U}_{\mathrm{n}} = (\bar{U}_{\mathrm{int}}, \bar{U}_{\mathrm{bd}})$, where $U_{\mathrm{int}} \in \mathcal{V}^h_{\mathrm{int}}$ and $U_{\mathrm{bd}} \in \mathcal{V}^h_{\mathrm{bd}}$. Correspondingly, we obtain a splitting of the stiffness matrix with respect to V_{int} and V_{bd} by

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{\mathrm{int,int}} & \mathbf{L}_{\mathrm{bd,int}} \\ \mathbf{L}_{\mathrm{int,bd}} & \mathbf{L}_{\mathrm{bd,bd}} \end{pmatrix}.$$

Here $\mathbf{L}_{\mathrm{int,int}}$ is the actual stiffness matrix on $\mathcal{V}_{\mathrm{int}}$. Furthermore, let us introduce a trivial extension operator $\mathbf{E} : \mathbb{R}^{\mathbf{I}_{\mathrm{int}}} \to \mathbb{R}^{\mathbf{I}_{\mathrm{n}}}$; $\bar{U}_{\mathrm{n}} \mapsto (\bar{U}_{\mathrm{int}}, 0)$ and the corresponding restriction operator $\mathbf{R} : \mathbb{R}^{\mathbf{I}_{\mathrm{n}}} \to \mathbb{R}^{\mathbf{I}_{\mathrm{int}}}$; $\bar{U} \mapsto \bar{U}_{\mathrm{int}}$. Based on this notation we can rewrite $L_{\mathrm{int,int}} = RLE$ and obtain the linear system

$$\mathbf{R}(M[\Phi^k] + \tau\mathbf{L}[\Phi^k])\mathbf{E}\bar{\Phi}^{k+1}_{\mathrm{int}} = \mathbf{R}(M[\Phi^k]\bar{\Phi}^k - \tau\mathbf{L}[\Phi^k]\bar{\Phi}^{k+1}_{\mathrm{approx}}). \tag{2.12}$$

to be solved for prescribed Dirichlet data $\Phi^{k+1}_{\mathrm{approx}} \in V_{\mathrm{bd}}$. The practical consequence is that we always work with the full matrix $\mathbf{L}$ and do not explicitly extract $\mathbf{L}_{\mathrm{int,int}}$ from it. Boundary data and solution vector are stored in one vector in $\mathbb{R}^{\mathbf{I}_{\mathrm{n}}}$. In the linear solver (e.g. a CG method) we only update values on interior nodes and keep the boundary nodes fixed.

Now, we have all the notation at hand to construct the inner iteration to be performed in each time step. In this inner iteration we modify the boundary conditions *and* the computational domain, to reflect the actual position of the evolving surface at the new time. This will in particular allow the evolving surface to leave the initially selected narrow band in a single time step. Inner iterations in one time step of an interface evolution – even though not in the context of a narrow band method – have recently been investigated for fluid structure interaction in [11] and for two-phase flow in [31].
At first, we initialize the inner iteration with data from the previous time step setting $\Phi^{k+1}_{\mathrm{approx}} = \Phi^k$ on $\mathcal{V}^h_{\mathrm{n}}$. Next, we extend Φ^k via a discrete signed distance transform from the interior of the current narrow band onto the narrow band boundary.
In the actual inner iteration, we solve (2.12) for $\Phi^{k+1}_{\mathrm{int}}$. Next, a new discrete signed distance function is computed for the discrete level set surface $[\Phi^{k+1}_{\mathrm{int}} = 0]$. Simultaneously, for a given width parameter j of the narrow band (cf. (2.8)), the corresponding narrow band domain $\Omega_{\mathrm{n}} = \Omega^j_{\mathrm{n}}$ is recomputed. This requires a rebuilding of the DT-Grid encoded data structure, which will be discussed in Section 3. The recomputed signed distance function on this new band is then considered as the new approximation $\Phi^{k+1}_{\mathrm{approx}}$. To solve (2.12) in the next inner iteration, we in addition need an extension of Φ^k onto the new narrow band domain. This update scheme is iterated until the narrow band domain is no longer changing and the resulting solution update on the boundary of the narrow band is sufficiently small.
The resulting scheme allows for large time steps and discrete surfaces propagating significantly outside the initial narrow band in one time step. If we use an index m to identify the inner iteration, we thus compute in each time step intermediate solutions $\Phi^{k+1,m}$ and intermediate narrow band domains denoted by $\Omega^{k+1,m}_{\mathrm{n}}$. In pseudo code

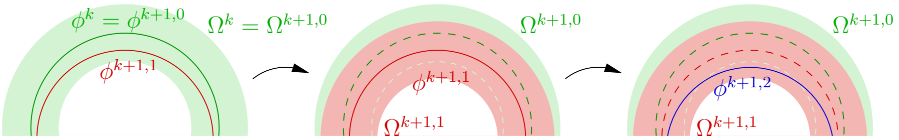


Fig. 2.3: A sketch of the inner iteration scheme: In the kth time step we start with the narrow band $\Omega^{k+1,0} = \Omega^k$ defined by the old time step data Φ^k (green) and compute the new solution time step $\Phi^{k+1,1}$ (red) using Eq. 2.12 (left); we define the new narrow band $\Omega^{k+1,1}$ around $\Phi^{k+1,1}$ (red) and extend Φ^k to this new domain (middle); based on the extended Φ^k and the boundary data imposed by $\Phi^{k+1,1}$ we then compute the next iterate $\Phi^{k+1,2}$ (blue) solving (2.12) this time on the already updated narrow band domain $\Omega^{k+1,1}$.

notation the resulting scheme for the computation of K time steps of mean curvature motion looks as follows (cf. also Figure 2.3):

```
**MeanCurvatureMotion**($\bar{\Phi}^0$) {
    initialize $\Omega_{\mathrm{n}}^0$;
    for ($k = 1$; $k \leq K$; $k++$) {
        $\bar{\Phi}^{k+1,0} = \Phi^k$; $m = 0$; $\Omega_{\mathrm{n}}^{k+1,0} = \Omega_{\mathrm{n}}^k$;
        do {
            compute $\bar{\Phi}^{k+1,m+1}$ on $\Omega_{\mathrm{n}}^{k+1,m}$ solving
                $\mathbf{R}(M[\Phi^k] + \tau\mathbf{L}[\Phi^k])\mathbf{E}\bar{\Phi}_{\mathrm{int}}^{k+1,m+1} = \mathbf{R}(M[\Phi^k]\bar{\Phi}^k - \tau\mathbf{L}[\Phi^k]\bar{\Phi}^{k+1,m})$;
            Define new band $\Omega_{\mathrm{n}}^{k+1,m+1}$ for $\bar{\Phi}^{k+1,m+1}$;
            Apply $\mathcal{E}[\Omega_{\mathrm{n}}^{k+1,m}, \Omega_{\mathrm{n}}^{k+1,m+1}]$ to $\bar{\Phi}^{k+1,m+1}$ and $\bar{\Phi}^k$;
            $m = m + 1$;
        } while($|\bar{\Phi}_{\mathrm{bd}}^{k+1,m} - \bar{\Phi}_{\mathrm{bd}}^{k+1,m-1}| \geq \delta$ or $\Omega_{\mathrm{n}}^{k+1,m} \neq \Omega_{\mathrm{n}}^{k+1,m-1}$)
        $\bar{\Phi}^{k+1} = \Phi^{k+1,m}$; $\Omega_{\mathrm{n}}^{k+1} = \Omega_{\mathrm{n}}^{k+1,m}$;
    }
}
```

Here, the operator $\mathcal{E}[\Omega_{\mathrm{n}}^{k+1,m}, \Omega_{\mathrm{n}}^{k+1,m+1}]$ represents the extension based on the discrete signed distance transform described above. In fact, for given data it computes a discrete solution of the Eikonal equation $|\nabla\phi| = 1$, which takes as input the given data values on the discrete zero level set $[\Phi^{k+1,m+1} = 0]$ of the current updated iterate $\Phi^{k+1,m+1}$. For the thinnest narrow band with $j = 1$ we use a first order PDE scheme [7]. For thicker narrow band we use a second order Weno PDE scheme [47]. Figure 2.3 sketches the first two update steps in the inner iteration of a single time step. Figure 2.4 demonstrates the difference between a straightforward implementation of a narrow band algorithm with the extended solution from the previous time step as Dirichlet data on the narrow band boundary and the proposed algorithm with the additionally built in inner iteration. Table 2.2 exemplifies the convergence properties of the proposed method depending on the width of the narrow band and the grid size in the case of spheres contracting under mean curvature motion. Here the exact solution is analytically given. Without inner iteration, we do not observe any order of consistency. With the inner iteration already on the thinnest possible narrow band we observe almost first order consistency in space and time. Furthermore, optimal second order consistency can be observed as long as the width of the narrow band is not too small and the time step is sufficiently small. This is due to the fact that the

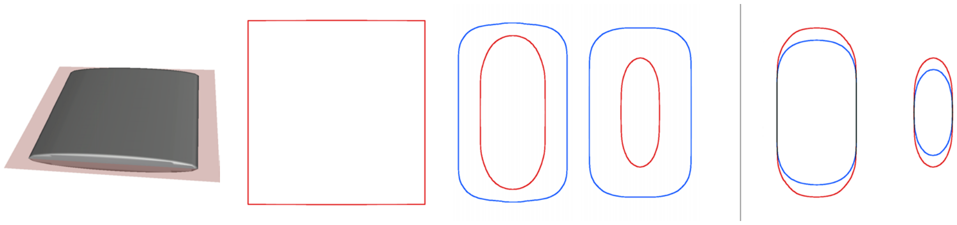


Fig. 2.4: Mean curvature motion applied to a cylindrical shape with rounded edges (left). On the planar cross section (sketched in red on the left) the discrete solution for the scheme with (in red) and without (in blue) inner iterations is compared at time step 0, 6 and 8 for a time step size $\tau = 0.04$ and a grid size $h = 0.008$ (middle). On the right we depict for two different time steps the cross section of the discrete solution based on the new algorithm (again in red), with the cross section corresponding to the method without inner iteration (in blue) at a different time but with comparable width in x direction. Here, we have taken into account a narrow band with one layer of cells on each side of the evolving surface.

grid size	17^3	33^3	65^3	129^3	257^3
	scheme with inner iterations $\tau = h^2$				
$j = 1$	0.018982	0.006987	0.004533	0.002694	0.001267
$j = 2$	0.012818	0.002848	0.001277	0.000883	0.000785
$j = 4$	0.007579	0.001830	0.000492	0.000144	6.16e-05
$j = 8$	0.006822	0.001629	0.000411	9.65e-05	2.33e-05
	scheme with inner iterations. $\tau = \frac{1}{8}h$				
$j = 1$	0.024438	0.015436	0.005113	0.000767	0.008048
$j = 2$	0.024601	0.014252	0.003951	0.002080	0.000305
$j = 4$	0.010950	0.008937	0.002940	0.001370	0.000603
$j = 8$	0.008948	0.004677	0.001996	0.000910	0.000393
	scheme with without inner iterations. $\tau = \frac{1}{8}h$				
$j = 1$	0.026056	0.050502	0.042877	0.047269	0.049729
$j = 2$	0.006402	0.014512	0.024515	0.036311	0.043778
$j = 4$	0.006223	0.002222	0.007890	0.019629	0.030859
$j = 8$	0.176777	0.003851	0.000525	0.004257	0.012067

Table 2.2: For a sphere of radius 0.25 around the origin as initial surface, we compute the mean error of the shrinking radius on the discrete zero level set compared to the analytical radius at time 0.0059. In the rows we compare narrow bands of different width measured in cells. The columns refer to different underlying full grids with a grid size h ranging from 16^{-1} to 256^{-1}. In the upper third, the scheme with inner iterations is applied for time step size $\tau = h^2$. This enables us to experimentally verify the (optimal) second order consistency in space for a not too thin narrow band. In the second third, for a fairly large time step $\tau = \frac{h}{8}$ and with the inner iterations we observe the expected first order consistency dominated by the time discretization error, whereas in the third part without inner iterations consistency completely breaks down.

scheme is only first order accurate in time.

Finally, Figure 2.5 shows the application of the discrete mean curvature motion algorithm on a highly detailed level set surface on an underlying 1986 × 1323 × 1104 grid.

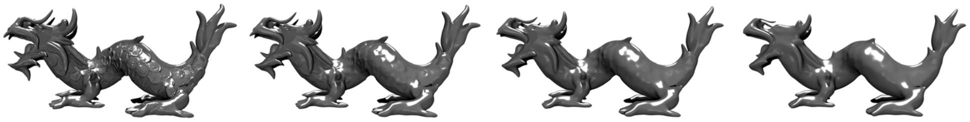


Fig. 2.5: The mean curvature motion scheme with Dirichlet boundary conditions is applied to an asian dragon surface ($1986 \times 1323 \times 1104$) [1] at times 0, 0.01, 0.06, and 0.23 (from left to right). The initial time step size is $\tau = h \approx 0.0005$. In later stages of the evolution a time step $\tau = 8\,h$ is applied. In each time step 1 up to 3 inner iterations are performed. Inner iterations are in particular needed in the initial phase of the evolution.

3. Narrow Band Implementation on the Dynamic Tubular Grid. In this section we describe how the proposed narrow band algorithms can be implemented on the Dynamic Tubular Grid (DT-Grid) data structure [46] in order to obtain a framework that is efficient both with regard to space- and time-utilization.

The DT-Grid is a data structure and a set of algorithms designed for storing data of a subset of nodes or elements defined on a regular grid. In particular the DT-Grid is well suited for storing data defined in a narrow band.

Asymptotically, the storage requirement of the DT-Grid is linear in the number of nodes in the narrow band. The asymptotic computational complexity of finite element method with its sequential data access to sparse, locally assembled matrix stencils, is also linear when implemented on the DT-Grid. Furthermore, the DT-Grid has been shown to perform faster and require less storage in practice than recent narrow band algorithms and sparse data structures, such as octrees and run-length encoded volumes [34, 45, 46]. Performance analysis indicates that the faster run-times of the DT-Grid are due both to the low memory footprint as well as the cache-coherency of the storage format, which results in low L1 and L2 cache miss rates [45]. Constant time access and cache performance for neighborhood operations are achieved through the careful use of *iterators* used to build, store, and manipulate the solution and all of the associated matrices/vectors. We first give a brief overview of the DT-Grid terminology required to comprehend the exposition of the implementation issues. Next, we describe how to implement the proposed finite element narrow band algorithms in the DT-Grid framework.

3.1. Dynamic Tubular Grid Terminology. The DT-Grid [46] separately stores the *topology* and *data* of the narrow band nodes in a compact form convenient for fast manipulation and access. In particular the nodes in the narrow band are stored in the DT-Grid in (x, y, z) lexicographic order which allows for a number of specific algorithmic constructs.

In order to represent the topology of the narrow band, a 3D DT-Grid consists of 1D, 2D and 3D grid components as shown leftmost in Figure 3.1. The 3D grid component consists of the nodes in the narrow band, the 2D grid component is the projection of the narrow band onto the XY-plane, and the 1D grid component is the projection of the 2D grid component onto the X-axis. For a full explanation of the DT-Grid we refer the reader to [46]. Here it is sufficient to say that each grid component has two constituents: *data* and *coord*. The *coord* constituent in the *nD* grid component stores the nth coordinate of the first and last node in each topologically connected component of grid points in a column of the *nD* grid component. These are colored red in Figure 3.1, left. As also depicted leftmost in Figure 3.1, the $data_{1D}$

and $data_{2D}$ constituents link the 1D, 2D and 3D grid components together by storing indices that point to the first coordinate in a column in the *coord* constituent of the 2D and 3D grid components respectively.

We denote the $coord_{1D}$, $coord_{2D}$, $coord_{3D}$, $data_{1D}$ and $data_{2D}$ constituents the *topology* since they specify the topology of the narrow band. The $data_{3D}$ constituent contains the actual data values, e.g. a level set function, and is stored separately from the topology in a flat data vector of length equal to the number of nodes in the narrow band. Since a total (lexicographic) ordering, starting from zero, is imposed on the nodes in the narrow band, entry i in the data vector corresponds uniquely to node i in the narrow band. In fact traversing the entries in the data vector sequentially from start to end corresponds to accessing the data of all nodes in the narrow band in lexicographic order. This also means that storing multiple data items at each node in the narrow band can be done by allocating multiple separate data vectors. Entry i in each of these data vectors then identify the data stored at node i in the narrow band.

The DT-Grid utilizes the concept of *stencil iterators* to sequentially access each individual node of the narrow band and provide constant time access to the node's neighbors as defined by a stencil suited for some computational task. In particular a specific stencil iterator consists of M individual iterators, where M is the number of nodes in the stencil, and an iterator is simply a construct that sequentially visits all grid points of the narrow band in lexicographic order. The center iterator of the stencil iterator dictates the movement of the entire stencil at each increment. Given the new (x, y, z) position of the center iterator during an increment of the stencil iterator, the remainder of iterators are positioned correctly relative to this position facilitated by the lexicographic storage order of the nodes. Details are given in [46].

3.2. Implementation. The applications of the narrow band framework proposed in this paper require the definition of a narrow band level set function as well as a number of vectors and matrices defined over this narrow band. The vectors contain an entry for each node in the narrow band, and the matrices are defined over the cardinal product of the narrow band with itself. However, the matrices are sparse and banded due to the limited support of the nodal basis functions employed in the finite element method. The narrow band, vectors and sparse matrices can be represented as a single instance of a DT-Grid topology and a collection of flat data vectors. Furthermore we create a number of customized stencil iterators that compute boundary face integration on the narrow band, matrix-vector multiplication as well as mass and stiffness matrix assembly with the stencil iterator framework. In the following subsections we describe the implementation of each of these components.

Iterating over Elements and Boundary Faces. The narrow band mesh used in our proposed framework is defined in terms of a finite element approach. Assembling the mass and stiffness matrices requires an iteration over these elements, whereas the stencil iterators of the DT-Grid visit nodes. However, an iterator that sequentially visits all elements of the narrow band can be phrased as a stencil iterator with a stencil of eight nodes as depicted in Figure 3.1.a. The iterator of the lexicographically smallest node in the stencil (marked in red in Figure 3.1.a) dictates the movement of the stencil, and the stencil iterator skips a node whenever at least one of the seven remaining nodes in the stencil are outside the narrow band. This procedure enumerates all elements of the narrow band in lexicographic order.

Similarly an iterator that sequentially visits all boundary faces of the narrow band can be phrased as a stencil iterator. Such a boundary face iterator is required

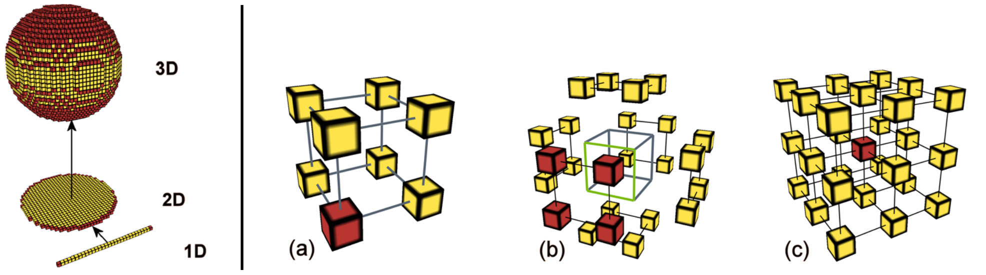


Fig. 3.1: Left: The 1D, 2D and 3D components of the DT-Grid encoding of a sphere. Right: a) The stencil of an iterator that visits elements. b) The stencil of an iterator that visits boundary faces. The element is shown as a wire frame cube in the center. c) The stencil of an iterator with access to neighboring nodes for a multi-linear nodal basis function (the center node is colored red).

in the computation of the boundary integrals arising from the transparent boundary conditions. A boundary face iterator can be implemented as an iterator that visits all elements (as described above), augmented with 24 nodes in the stencil as shown in Figure 3.1.b. Consider the element-face outlined in green. This face constitutes a boundary face if at least one of the nodes colored red is outside the narrow band. Similar configurations exist for the remaining faces.

Vectors and Sparse Matrices. As mentioned above, data items associated to each node in the narrow band can be stored as flat vectors of length equal to the number of nodes in the narrow band. Simple vector operations that operate on vectors locally on an entry-by-entry basis, such as addition, subtraction and dot-product, utilize no information about the topology of the narrow band and hence require no special treatment in our framework.

The mass and stiffness matrices arising from the finite element machinery on the narrow band are sparse banded matrices due to the limited support of the nodal basis functions. In fact in case of multi–linear finite elements the number of non-zero entries in a row of a matrix used in our framework will be at most $N = 27$. Equivalently to the situation arising with vectors, simple matrix operations that operate on matrices locally on an entry-by-entry basis can be done without knowledge of the narrow band topology.

To facilitate matrix-vector multiplications, $c = \mathbf{A}b$, a coupling with the narrow band topology is required. This is done by means of a stencil iterator and a stencil defined by the support of a nodal basis function. In particular, the stencil at a particular node, i, must include all neighboring nodes (cf. Figure 3.1.c). The matrix-vector product is formed by sequential iteration of the stencil iterator over the narrow band topology. At each node, i, entry c_i of the resulting vector is computed as $c_i = \sum_{j=0}^{N-1} A_{i,S(j)} b_{S(j)}$, where S is the stencil iterator, and $S(j)$ is the mapping from node j in the stencil to the corresponding number in the total lexicographic ordering of nodes in the narrow band. Note that if a node in the stencil is outside the narrow band, its contribution in the above product is set to zero.

The above matrix storage has the advantage that a single DT-Grid topology instance is used to define both the topology of the narrow band as well as of the location of non-zero entries in matrices associated with this narrow band. The restriction, $\mathbf{R}$, and extension, $\mathbf{E}$, operators on vectors arising in connection with Dirichlet boundary conditions can be integrated into the above matrix-vector implementation in a straightforward manner. The implementation of these operators requires the construction of a boundary-mask vector of bits, stored along with the topology in lexicographic order, which distinguishes between internal and boundary nodes by storing a 1 for each internal node and a 0 for each boundary node. Hence internal and boundary nodes are stored intermixed in order to retain the lexicographic order of the nodes in the narrow band. The boundary-mask is incorporated into the matrix-vector multiplication to construct products of the form $\mathbf{RAE}\bar{U}_{\mathrm{n}}$ and $\mathbf{RA}\bar{U}_{\mathrm{bd}}$. In both of these cases an entry in the resulting vector is computed only if it is an internal node (this is due to the trailing restriction operator). In the case of the product $\mathbf{RAE}\bar{U}_{\mathrm{n}}$, the computation of the i'th entry in the resulting vector is modified to $\sum_{j=0}^{N-1} \bar{B}_{S(j)} A_{i,S(j)} \bar{U}_{\mathrm{n}S(j)}$, where $\bar{B}_{S(j)}$ is the $S(j)$'th entry of the boundary-mask vector. In the case of the product $\mathbf{RA}\bar{U}_{\mathrm{bd}}$, the computation is modified to $\sum_{j=0}^{N-1} (\neg \bar{B}_{S(j)}) A_{i,S(j)} \bar{U}_{\mathrm{bd}S(j)}$, where $\neg$ implies a negation of the boundary mask bit.

Concerning the storage requirements, the matrices usually constitute the bottleneck in practical implementations. In the case of multi-linear basis functions for example, a stiffness matrix will require 27 times the storage required by the level set values in the narrow band. To reduce the storage requirements we keep as few matrices in memory as possible during computations. If necessary we lower the storage requirements of each matrix by performing a uniform quantization of the individual matrix entries whenever possible. The uniform quantization requires the numerical range of matrix entries to be known, and these can be determined prior to the simulation based on the properties of the particular matrix.

Rebuilding the Dynamic Tubular Grid and Extending Data. The simulation of evolving surfaces requires an iterative tracking of the narrow band domain as defined in Section 2. This involves rebuilding the narrow band while extending level set data (e.g. the discrete signed distance functions $\Phi^{k+1,m+1}$ and Φ^{k} from Section 2.4) onto the new band. Rebuilding the narrow band domain is facilitated by the DT-Grid's *dilation* operator [46]. In particular dilating a 3D DT-Grid by j nodes corresponds to iterating a cube-shaped stencil of dimensions $(2j+1)^3$ over the nodes in the narrow band and adding to the narrow band all nodes outside that pass under the support of the stencil. Using the dilation operator, rebuilding the narrow band domain according to an updated level set function entails first the identification of narrow band elements that the discrete interface passes through. Secondly this set of nodes is dilated to define a new narrow band domain in accordance with the definition in Section 2. Many algorithms exist for computing and re-initializing signed distance values, and we proceed as follows. The iterate $\Phi^{k+1,m+1}$ is re-initialized to a discrete signed distance function on the updated narrow band domain by the algorithm in [51]. The values of the previous solution time step Φ^{k} at nodes of the previous narrow band domain $\bar{\Omega}^{k+1,m}$ are kept fixed, whereas discrete signed distance values at the new nodes are computed in an iterative manner applying the method in [37].

4. Applications.

4.1. Reaction-Diffusion Textures on Implicit Surfaces. As an important application of PDEs on a fixed surface we consider the generation of textures based on a reaction diffusion model. Reaction diffusion equations describe a variety of biological and chemical phenomenon, and have been used in 3D graphics for the generation of interesting, natural-looking textures on surfaces [63]. Here, we consider the system of partial differential equations proposed by Turing [62]

$$\begin{aligned}\partial_t a &= c_s(\alpha - ab) + c_a \Delta_{\mathcal{M}} a\\ \partial_t b &= c_s(ab - b - \beta) + c_b \Delta_{\mathcal{M}} b\end{aligned}$$

on a level set surface $\mathcal{M} = [\phi = 0]$, where c_s, c_a, c_b, and α are parameters that determine the shapes, their scale and frequency in the steady state limit and $\beta : \mathcal{M} \to \mathbb{R}$ is a stochastic function, that creates a degree of randomness in the texture. The tradeoffs and choices for these parameters are nicely described in [54]—we list specific choices for these parameters with the figures in this paper. As discussed in the scalar case in Section 2.1 we discretize the reaction term explicitly and the diffusion term implicitly in time and make use of the transparent Neumann boundary conditions. Figure 4.1 shows the solution of a reaction-diffusion equation on a 3D dragon model [1], which has been scan-converted to a DT-Grid with 982 × 695 × 442 grid point of the underlying mesh using the method in [34]. In the literature, anisotropic reaction-diffusion has been proposed to generate anisotropic texture pattern. For that purpose we replace the isotropic diffusion operator $\Delta_{\mathcal{M}} u$ by an anisotropic one. Given a tangent, unit length vector field v we define

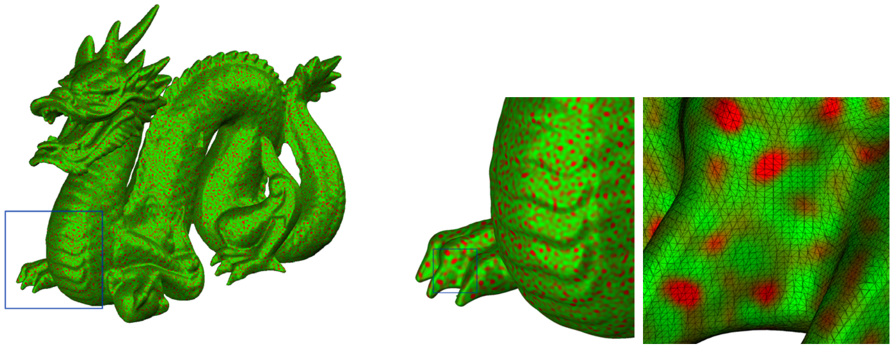


Fig. 4.1: Reaction diffusion textures generated on a dragon model are depicted on different scales from left to right. On the right, in addition the edges of a corresponding marching cube mesh are shown to represent the underlying grid visually. The pattern is generated in 150 time steps for the parameters $c_s = 0.05$, $c_a = 2.5\ 10^{-7}$, $c_b = 6.3\ 10^{-8}$, $\alpha = 16$, and $\beta = 12 \pm 0.4$ and an effective grid spacing $h = 0.00102$.

$$\Delta_{\mathcal{M},v} u = |\nabla\phi|^{-1} \mathrm{div}\left(|\nabla\phi| \left(g_1 vv^T + g_2(\mathbb{1} - vv^T)\right) P[\phi]\nabla u\right) ,$$

where diffusion in v direction is weighted by g_1 and in the perpendicular direction by g_2. Now, we choose v such that the texture aligns approximately with the subdominant principal curvature direction on the surface. Therefore, we pick up the structure tensor approach from [64] and define the 3 × 3 tensor

$$J(x) = \left(\nabla\phi_\sigma(x) \otimes \nabla\phi_\sigma(x)\right)_\sigma , \tag{4.1}$$

where the index σ indicates an approximate Gaussian filtering via the application of the discrete heat equation semigroup with time step $\frac{\sigma^2}{2}$. Let us suppose that

$\lambda_1 \leq \lambda_2 \leq \lambda_3$ are the eigenvalues corresponding to eigenvectors v_1, v_2, and v_3. Then we choose $v = v_1$, $g_1 = g^2(\lambda_1)$, and $g_2 = g^2(\lambda_2)$, where $g(s) = (1 + \frac{s^2}{\gamma^2})^{-1}$. Examples of such anisotropic textures are shown in Figure 4.2.

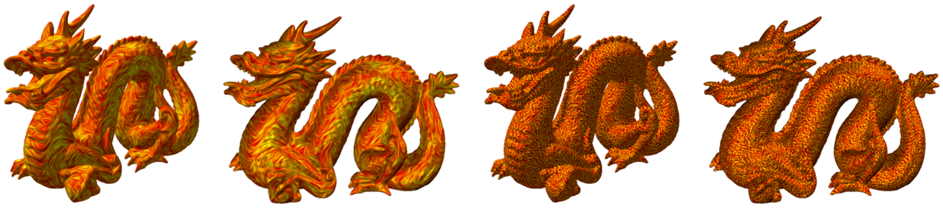


Fig. 4.2: An anisotropic reaction-diffusion is applied to the dragon model, resulting in anisotropic spot texture on two different scales. For each scale we show a side view and a diagonal view after 450 iterations with parameters $\gamma = \frac{1}{2}, c_s = 0.05$, $\alpha = 16$, and $\beta = 12 \pm 0.4$. To control the size of the spots the relation between c_a and c_b to c_s has to be changed. We take into account $c_a = 2.5\ 10^{-7}$ and $c_b = 6.3\ 10^{-8}$ (left), and $c_a = 1.5625\ 10^{-8}$ and $c_b = 3.9375\ 10^{-9}$ (right).

4.2. Surface Fairing based on Curvature Motion. Curvature motion models have been considered for the fairing of scanned surface models [20, 65]. Figure 4.3 shows results for the standard mean curvature motion model for a scan converted model (DT-grid of size 2471 × 1439 × 827) of the *Lucy Statue* [1]. The full 3D grid with level-set data stored as floats, would require almost 11 gigabytes of data, whereas the DT-Grid representation of the model requires roughly 159MB. Because of the very large sizes of such models, the run times in the current version are still significant. The mean-curvature results required roughly eight minutes per time step on an Intel Pentium 3.6 GHz processor for the Lucy Statue [1] in Figure 4.3 and about 5.5 minutes per time step for the Asian Dragon [1] in Figure 4.4. However, these are very large models, which are not manageable without the very thin band approach offered by the DT-grid. We investigated several runs of the mean curvature code by a profiler, and found that the overhead of employing the DT-Grid data structure is relatively small. In particular the most time-intensive part is independent of the DT-Grid and consists of the preparation of the matrices which comprises about 62% of the total run-time. The remaining 38% is spent iterating stencil-iterators over the DT-Grid, solving the linear systems, computing the signed distance function, and rebuilding the DT-Grid during the inner iterations.

In Figure 4.5 the mean curvature motion is applied to a dumbbell which breaks into two pieces. This demonstrates the advantage of level set methods to change the topology of the surface and it illustrates the very accurate resolution of singularities via a level set approach on very large grids. The fairing results of the standard mean curvature motion model can be improved incorporating the anisotropic geometric diffusion model (cf. [13, 52])

$$\partial_t \phi - |\nabla \phi| \mathrm{div}(A \nabla \phi) = 0\,. \tag{4.2}$$

Given the diagonalization $Q\,\mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)Q^T$ of the regularized geometric structure tensor J_σ from (4.1), we choose the anisotropy tensor $A(x) = Q\,\mathrm{diag}(g(\lambda_1), g(\lambda_2), 0)Q^T$. This schemes preserves edges on the surface indicated by a large principal curvature across the edge. Notice that due to the motion of the surface the structure tensor has

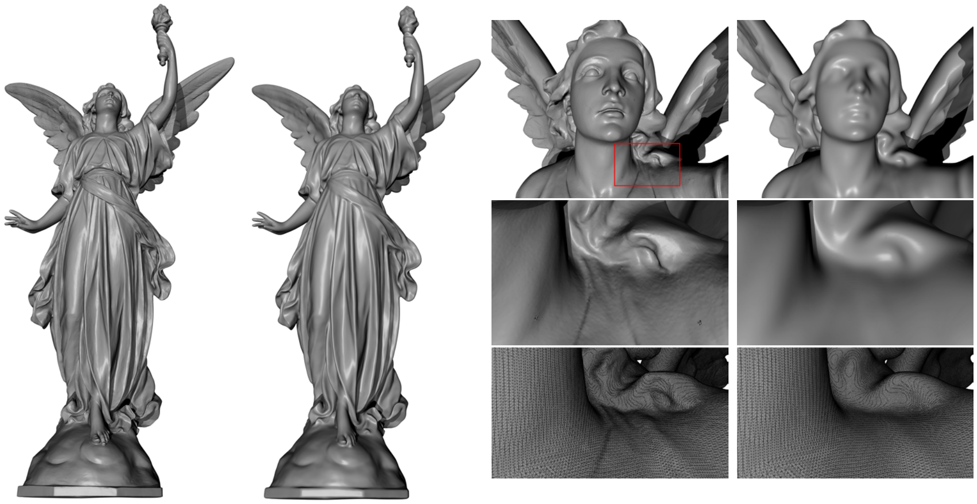


Fig. 4.3: Mean curvature motion is applied to the Lucy statue. On the left the initial surface and the result after 19 time steps with time step size $\tau = 0.0004$ are shown. On the right, we render corresponding blow ups at two different scales and overlay the finest scale blow up with the mesh generated using the marching cubes algorithm [39].

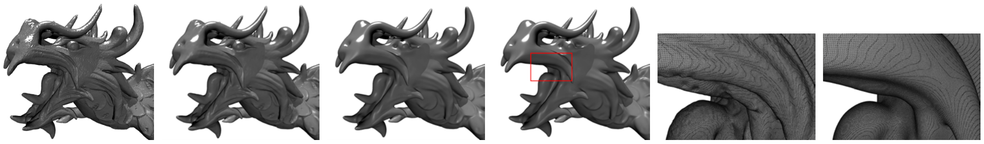


Fig. 4.4: Evolution by mean curvature motion for the asian dragon model. Here from left to right a closeup to the head is shown at different time steps 0, 6, 30, 44 of the evolution ($\tau = 10^{-6}$). The last two images show the mesh, generated using the marching cubes algorithm [39], of an even closer blowup of the dragon tongue at time step 0 and 44.

to be recomputed in each step. We observed that the structure tensor computation requires a narrow band width of at least 2 to enable a reliable estimate of the principal curvatures. In Figure 4.6 this method is applied to smooth the Asian dragon model. Here, we choose $\tau = 0.000015$ and $\lambda = 0.1$. A narrow band with width parameter $j = 2$ is used throughout in the algorithm.

4.3. Prior Driven Denoising with Anisotropic Mean Curvature Motion. The usual isotropic mean curvature motion model is known as the gradient flow of the usual surface area $E = \int_{\mathcal{M}} da$. Hence, given the enclosed volume, rounds spheres are the preferred shape. For specific applications it might be advantageous to consider different convex shapes as asymptotically optimal. This can be achieved via a weighting of surface area depending on the surface orientation given by the normal direction n. Thus, one considers the gradient flow with respect to the anisotropic surface energy $E_\gamma = \int_{\mathcal{M}} \gamma(n)\, da$, where $\gamma : \mathcal{M} \to \mathbb{R}$ is a positive, convex and 1-homogeneous function. When minimizing this energy, certain directions are preferred and the so called Wulff shapes [4] are minimizers up to scaling. Such an anisotropic mean cur-

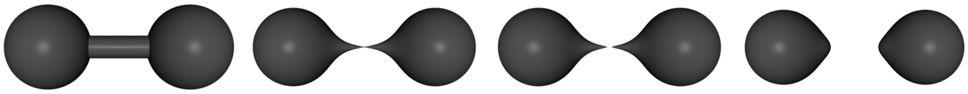


Fig. 4.5: The evolution of a dumbbell under mean curvature motion is shown based on a grid resolution of 1017 × 387 × 387 to demonstrate that topological changes are recovered correctly at high resolution in this framework. From left to right the solution is rendered at time 0, $1.288067\ 10^{-3}$, $1.290934\ 10^{-3}$, and $2.278962\ 10^{-3}$. The initial radius of the spheres of the dumbell is 0.4971 and computations are performed with a time step $\tau = 0.95554\ 10^{-6}$.

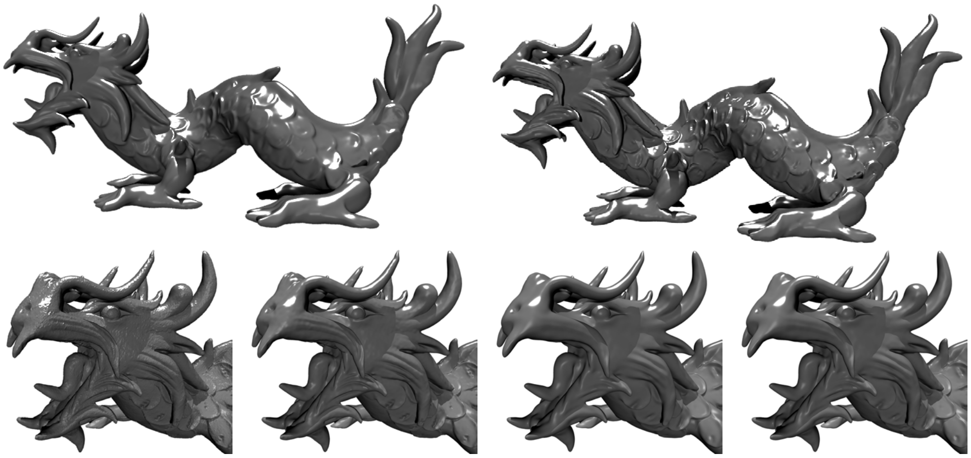


Fig. 4.6: The mean curvature motion model is compared with the anisotropic geometric diffusion after 30 time steps (top row). Furthermore, blow ups at the time steps 0, 7, 30, and 47 are shown for the anisotropic geometric diffusion model (bottom row).

vature motion model has been applied to the smoothing of tubular vessel structures in [44]. The level set formulation of the continuous anisotropic mean curvature motion is given by

$$\partial_t \varphi - |\nabla \varphi| \mathrm{div}\left(\gamma_z(\nabla \varphi)\right) = 0.$$

For the discrete flow Dziuk and Deckelnick proposed a stabilization term [17] which leads to the following weak formulation:

$$\int_\Omega \frac{\Phi^{k+1} - \Phi^k}{\tau |\nabla \Phi^k|_\epsilon} \Theta\, dx + \int_\Omega \gamma_z(\nabla \Phi^k) \cdot \nabla \Theta\, dx = -\lambda \int_\Omega \frac{\gamma(\nabla \Phi^k)}{|\nabla \Phi^k|_\epsilon} (\nabla \Phi^{k+1} - \nabla \Phi^k) \cdot \nabla \Theta\, dx$$

To derive a matrix vector formulation of this equation we define in addition a vector $\bar{F}_\gamma[\Phi]$ and a nonlinear stiffness matrix $L_\gamma[\Phi]$ given by

$$F_\gamma[\Phi] = \left(\int_\Omega I_h^3 \left(\gamma_z(\Phi) \nabla \Theta_i\right)\, dx \right)_{i \in I}, L_\gamma[\Phi] = \left(\int_\Omega I_h^1 \left(\frac{\gamma(\nabla \Phi) \nabla \Theta_i \cdot \nabla \Theta_j}{|\nabla \Phi|_\varepsilon} \right) dx \right)_{i,j \in I},$$

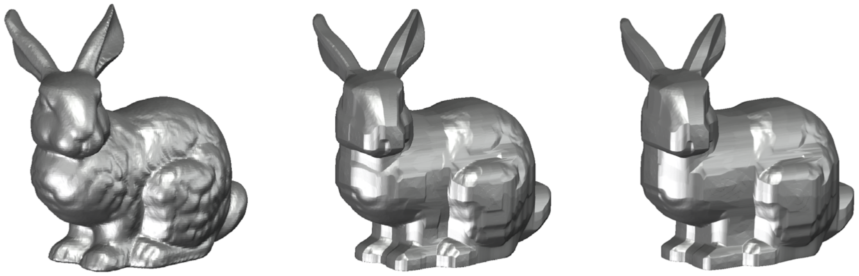


Fig. 4.7: Anisotropic mean curvature motion applied to the Stanford Bunny of resolution $595 \times 590 \times 462$ with a cube as Wulff shape. Time steps 0, 2000 and 4000 are depicted, where $\tau = 5.7\,10^{-8}$ and $\lambda = 2.5$.

where I_h^3 represents a third order exact Gaussian quadrature on the hexahedral cells. We end up with the semi implicit time discretization

$$(M[\Phi^k] + \tau\lambda L_\gamma[\Phi^k])\bar{\Phi}^{k+1} = (M[\Phi^k] + \tau\lambda L_\gamma[\Phi^k])\bar{\Phi}^k - \tau\bar{F}_\gamma[\Phi^k]\,.$$

Transparent boundaries are handled as in the linear case in Section 2.4. In Fig. 4.7 we applied the anisotropic curvature motion model to the Stanford bunny leading to successively more "cubistic" representations of the initial surface. Here the anisotropy is given by $\gamma(z) = \|z\|_1 = |z_1| + |z_2| + |z_3|$. In the numerical experiment we observed that, different from the isotropic case, we have to choose a narrow band which is at least two cells wide to avoid small oscillatory artifacts.

4.4. Comparison to parametric surface processing. The level set method is renown for its significant advantages over parametric surface approaches: in many applications the raw data is already given in implicit form, a level set evolution can easily cope with topological transitions, and the interplay between processes on the surfaces and in the bulk regions can be integrated in a straightforward way into a level set method. Nethertheless, the computational overhead seems to be significant. Hence, in that respect we have explicitly compared our narrow band level set scheme with a corresponding scheme for the evolution of parametric surfaces for a simple test problem. We have considered a single iteration of the isotropic mean curvature motion applied to a sphere. We have generated an implicit representation of the sphere with radius $\frac{1}{3}$ and center in $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ on an underlying grid with resolution 513^3 and a triangulated sphere whose grid size given by the maximum length of edge in the approximately uniform triangulation is approximately the same as the grid size $h = 512^{-1}$ of the above regular grid. In the level set case we define our computation domain as the thinnest possible narrow band, which consists of an average thickness of about 5 cells. We have executed one time step either of our level set or of a parametric curvature motion scheme [24]. In both cases the time step size is chosen as $\tau = h^2$. We have separately measured the time needed for the assembly of the matrix, for the solution of the corresponding linear system of equations, and in the implicit case for the rebuilding of the narrow band including the time for the level set redistancing with a first order PDE scheme [7] (cf. Table 4.1). A single iteration in the CG algorithm is about 12.5 times faster for the parametric grid than for the level set narrow band. This is not surprising because the narrow band scheme requires about 5.5 times the

	rebuilding	assembly	solving	degrees of freedom	number iterations
narrow band	10.202 s	14.133 s	5.864 s	1 098 484	5
parametric		0.780 s	4.296 s	196 614	84

Table 4.1: Comparison of mcm on parametric and on implicit surfaces.

number of effective degrees of freedom. Despite of this the observed overhead for the data access on the narrow band of about a factor 2 - measured in computing time per degree of freedom - is strikingly small. Here the cache coherence of the narrow band scheme pays out. The number of CG iterations is 17 times smaller for the level set approach, which we expect to be due to the Dirichlet boundary conditions considered on the boundary of the narrow band. Altogether, this results in a similar computing time for the actual solution of the linear system. In addition, one has to count for the obviously much higher assembly time and the time for the additional data structure rebuilder.

5. Conclusions. This paper presented a novel solution strategy for geometric PDEs on implicit surfaces, represented via a dynamic tubular grid approach on extremely large underlying grids. The method comes along with boundary conditions which reduce the effects of the narrow band on the solution and allow the use of semi-implicit finite-element schemes in the context of very thin bands. The results show that it is feasible to solve PDEs on very large implicit surfaces (with resolutions comparable to state of the art triangular surface meshes) in reasonable amounts of time. Future work will be devoted to the transfer of further PDE based models already studied on triangular surface meshes to highly detailed implicit surface representations. So far we have either confined to evolution problems on fixed surfaces or the evolution of the geometry itself obeying a geometric PDE. Surely, one can couple both and consider a density u on a moving surface, which is governed by an evolution problem and acting as a parameter for an evolution problem of the surface itself. Indeed, the following system of reaction diffusion equations describes such a coupled system:

$$\begin{aligned}\partial_t u(t) - \Delta_{\mathcal{M}(t)} u(t) &= f(u(t), x(t)),\\ \partial_t x(t) - \epsilon \Delta_{\mathcal{M}(t)} x(t) &= g(u(t), x(t))\end{aligned}$$

for a small parameter ϵ. There are plenty of applications in geometric modeling or geometric texture generation, which lead to such coupled problems.

Acknowledgments. The narrow band approach for higher order geometric evolution problems in the PhD thesis of Marc Droske [21] inspired this work. We thank Marc Droske and Martin Burger for many discussions and fruitful remarks on narrow band methods. We also thank Ken Museth and Ola Nilsson for permission to use their software and Nadine Olischläger for her help with the comparison of parametric and implicit surface models. This work was partially supported by the Deutsche Forschungsgemeinschaft through the Collaborative Research Center 611 *Singular phenomena and scaling in mathematical models*, the Hausdorff Center for Mathematics at Bonn University and the Danish Agency for Science, Technology and Innovation.

REFERENCES

[1] Stanford scanning repository. http://graphics.stanford.edu/data/3Dscanrep/.

[2] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[3] D. Adalsteinsson and J. A. Sethian. A level set approach to a unified model for etching, deposition, and lithography III: Re-deposition, re-emission, surface diffusion, and complex simulations. *Journal of Computational Physics*, Volume 138, Issue 1:193–223, 1997.

[4] B. Andrews. Volume-preserving anisotropic mean curvature flow. *Indiana University Mathematics Journal*, 50:783–827, 1991.

[5] H. Ben Ameur, M. Burger, and B. Hackl. Level set methods for geometric inverse problems in linear elasticity. *Inverse Problems*, 20(3):673–696, 2004.

[6] M. Bertalmío, F. Mémili, L. T. Cheng, G. Sapiro, and S. Osher. *Geometric level set methods in imaging, vision, and graphics*, chapter Variational Problems and Partial Differential Equations on Implicit Surfaces: Bye Bye Triangulated Sufaces?, pages 381–397. Springer, New York, 2003.

[7] F. Bornemann and C. Rasch. Finite-element discretization of static hamilton-jacobi equations based on a local variation principle. *Comput. Visual Sci.*, (9):57–69, 2006. E-print available at arXiv:math.NA/0403517.

[8] M. Burger. A framework for the construction of level set methods for shape optimization and reconstruction. *Interfaces and Free Boundaries*, 5:301–329, 2003.

[9] M. Burger. Finite element approximation of elliptic partial differential equations on implicit surfaces. *Computing and Visualization in Science*, 2007. to appear.

[10] J. Cahn, P. Fife, and O. Penrose. A phase-field model for diffusion-induced grain-boundary motion. *Acta Materiala*, 45(10):4397–4413, 1997.

[11] P. Causin, J. F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comput. Methods Appl. Mech. Engrg.*, 194(42-44):4506–4527, 2005.

[12] S. Chen, B. Merriman, M. Kang, R. E. Caflisch, C. Ratsch, C. L.-T., M. Gyure, R. P. Fedkiw, and S. Osher. A level set method for thin film epitaxial growth. *Journal of Computational Physics*, 167:475 – 500, January 2001.

[13] U. Clarenz, M. Rumpf, and A. Telea. Robust feature detection and local classification for surfaces based on moment analysis. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):516–524, 2004.

[14] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.

[15] M. de Berg. *Computational Geometry.* Springer, January 2000.

[16] K. Deckelnick and Dziuk. Convergence of numerical schemes for the approximation of level set solutions to mean curvature flow. In M. Falcone and C. Makridakis, editors, *Numerical Methods for Viscosity Solutions and Applications*, volume 59 of *Series Adv. Math. Appl. Sciences*, pages 77 – 93, 2001.

[17] K. Deckelnick and G. Dziuk. A fully discrete numerical scheme for weighted mean curvature flow. *Numerische Mathematik*, 91(3):423–452, 2002.

[18] K. Deckelnick, G. Dziuk, C. Elliott, and C. Heine. An h-narrow band finite element method for elliptic equations on implicit surfaces. Preprint 18-07, Fakultät für Mathematik und Physik, Universität Freiburg, 2007.

[19] K. Deckelnick, G. Dziuk, and C. M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232, 2005.

[20] M. Desbrun, M. Meyer, P. Schröder, and A. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Computer Graphics (SIGGRAPH '99 Proceedings)*, pages 317–324, 1999.

[21] M. Droske. *On Variational Problems and Gradient Flows in Image Processing.* Dissertation, University Duisburg, 2005.

[22] M. Droske, B. Meyer, M. Rumpf, and C. Schaller. An adaptive level set method for medical image segmentation. *Lecture Notes in Computer Science*, pages 412–422, 2001.

[23] M. Droske and M. Rumpf. A level set formulation for willmore flow. *Interfaces and Free Boundaries*, 6(3):361–378, 2004.

[24] G. Dziuk. An algorithm for evolutionary surfaces. *Numer. Math.*, 58:603–611, 1991.

[25] G. Dziuk and C. M. Elliott. Surface finite elements for parabolic equations. *Journal of Computational Mathematics*, 25:385–407, 2007.

[26] D. Enright, F. Losasso, and R. Fedkiw. A fast and accurate semi-lagrangian particle level set method. *Computers and Structures*, 83:479–490, Feb. 2005.

[27] L. Evans and J. Spruck. Motion of level sets by mean curvature I. *Journal of Differential*

Geometry, 33(3):635–681, 1991.

[28] S. Frisken and R. Perry. Simple and efficient traversal methods for quadtrees and octrees. *Journal of Graphics Tools*, 7(3), 2003.

[29] J. B. Greer. An improvement of a recent eulerian method for solving pdes on general geometries. *J. Scientific Computing*, 29, No. 3:321–352, 2006.

[30] J. B. Greer, A. L. Bertozzi, and G. Sapiro. Fourth order partial differential equations on general geometries. Ucla computational and applied mathematics reports, University of California Los Angeles, 2005. Journal of Computational Physics, Volume 216 , Issue 1, Pages: 216 - 246, Year of Publication: 2006.

[31] S. Groß, V. Reichelt, and A. Reusken. A finite element based level set method for two-phase incompressible flows. *Comput. Visual Sci.*, 9:239–257, 2006.

[32] G. Grün, M. Lenz, and M. Rumpf. A finite volume scheme for surfactant driven thin film flow. In R. Herbin and D. Kröner, editors, *Finite volumes for complex applications III*, pages 567–574. Hermes Penton Sciences, 2002.

[33] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. *ACM Trans. Graph.*, 22(3):871–878, 2003.

[34] B. Houston, M. Nielsen, C. Batty, O. Nilsson, and K. Museth. Hierarchical RLE Level Set: A Compact and Versatile Deformable Surface Representation. *ACM Transactions on Graphics*, 25(1):1–24, 2006.

[35] G. Irving, E. Guendelman, F. Losasso, and R. Fedkiw. Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Transactions on Graphics (SIGGRAPH)*, Aug. 2006.

[36] A. James and J. Lowengrue. A surfactant conserving volume of fluid method for interfacial flows with insoluble surfactant. *J. Comput. Phys.*, 201(2):685–722, 2004.

[37] W.-K. Jeong and R. Whitaker. A fast iterative method for eikonal equations. *SIAM Journal on Scientific Computing (in review)*.

[38] A. E. Lefohn, J. M. Kniss, C. D. Hansen, and R. T. Whitaker. A streaming narrow-band algorithm: Interactive computation and vusualization of level sets. *IEEE Transactions on Visualization and Computer Graphics, Juli-August 2004*, 10 (4):422–433, 2004.

[39] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.

[40] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*, 35:995–1010, 2006.

[41] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics*, 23(3), Aug. 2004.

[42] C. Min. Local level set method in high dimension and codimension. *Journal of Computational Physics*, 200:368–382, 2004.

[43] K. Museth, D. Breen, R. Whitaker, S. Mauch, and D. Johnson. Algorithms for interactive editing of level set models. *Computer Graphics Forum*, 24 (4):821–841, 2005.

[44] O. Nemitz, M. Rumpf, T. Tasdizen, and R. Whitaker. Anisotropic curvature motion for structure enhancing smoothing of 3D MR angiography data. *Journal of Mathematical Imaging and Vision*, 27(3):217–229, 2007.

[45] M. B. Nielsen. *Efficient and High Resolution Level Set Simulations*. PhD thesis, Aarhus University, 2006.

[46] M. B. Nielsen and K. Museth. Dynamic Tubular Grid: An efficient data structure and algorithms for high resolution level sets. *Journal of Scientific Computing*, 26(3):261–299, 2006. (submitted November, 2004; accepted January, 2005).

[47] O. Nilsson and A. Söderström. Euclidian distance transform algorithms: A comparative study. Technical Report 2, Linköping University, 2007.

[48] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.

[49] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[50] D. Peng, B. Merriman, S. Osher, H. Yhao, and M. Kang. A pde-based fast local level set method. *Journal of Computational Physics*, 155(2):410–438, 1999.

[51] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155(2):410–438, 1999.

[52] T. Preußer and M. Rumpf. A level set method for anisotropic geometric diffusion in 3D image processing. *SIAM J. Appl. Math.*, 62(5):1772–1793, 2002.

[53] R. V. Roy, A. J. Roberts, and M. E. Simpson. A lubrication model of coating flows over a

curved substrate in space. *J. Fluid Mech.*, 454:235–261, 2002.

[54] A. R. Sanderson, R. M. Kirby, C. R. Johnson, and L. Yang. Advanced reaction-diffusion models for texture synthesis. *journal of graphics tools*, 11(3):47–71, 2006.

[55] P. Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. *Journal of Scientific Computing*, 19 (1 – 3):439 – 456, 2003.

[56] N. Stolte and A. Kaufman. Parallel spatial enumeration of implicit surfaces using interval arithmetic for octree generation and its direct visualization. In *Implicit Surfaces '98*, pages 81–87, 1998.

[57] H. A. Stone. A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface. *Physics of Fluids*, 2:111–112, Jan. 1990.

[58] J. Strain. Tree methods for moving interfaces. *Journal of Computational Physics*, 151(2):616–648, 1999.

[59] J. A. Strain. Fast tree-based redistancing for level set computations. *Journal of Computational Physics*, 152, 1999.

[60] J. A. Strain. Semi-lagrangian methods for level set equations. *Journal of Computational Physics*, 151, 1999.

[61] J. A. Strain. A fast modular semi-lagrangian method for moving interfaces. *Journal of Computational Physics*, 161, 2000.

[62] A. M. Turing. The chemical basis of morphogenesis. *Phil. Trans. Roy. Soc. London*, B 237:37–72, 1952.

[63] G. Turk. Re-tiling polygonal surfaces. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 25, No. 4, 1991.

[64] J. Weickert. *Anisotropic diffusion in image processing.* Teubner, 1998.

[65] R. T. Whitaker. A level-set approach to 3D reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998.