Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 190 (2003) 295–310
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Simplicial isosurfacing in arbitrary dimension and codimension ☆

Chohong Min *

Department of Mathematics, UCLA, 3165 Sepulveda Blvd. #304, USA

Received 13 February 2003; received in revised form 7 May 2003; accepted 13 May 2003

Abstract

The level set method has been successfully used for moving interface problems. The final step of the method is to construct and visualize the isosurface of a discrete function $\phi : \{0, \ldots, N\}^n \to \mathbb{R}^m$. There have existed many practical isosurfacing algorithms when $n = 3,\ m = 1$ or $n = 2,\ m = 1$. Recently we have begun to see the development of isosurfacing algorithms for higher dimensions and codimensions. This paper introduces a unified theory and an efficient isosurfacing algorithm that works in arbitrary number of dimensions and codimensions. The isosurface Γ of a discrete function ϕ is defined as the isosurface of its simplicial interpolant $\hat{\phi} : [0, N]^n \to \mathbb{R}^m$. With this simplicial definition, Γ is geometrically a piecewise intersection of a simplex and m hyperplanes. Γ is constructed as the union of simplices. The construction costs $O(N^n)$ with a uniform grid and $O(N^{n-m}\log(N))$ with a dyadic grid in numerical space and time. When $n = m + 1$ or $m + 2$, Γ is projected down into $\mathbb{R}^3$ and can be visualized. For surface visualizations, a simple formula is presented calculating the normal vector field of the projection of Γ into $\mathbb{R}^3$, which gives light shadings.

1. Introduction

The purpose of this paper is to construct and visualize the isosurface of a discrete function in arbitrary dimension and codimension. A discrete function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$ is meant to be a vector valued function defined on a uniform grid in $\mathbb{R}^n$, which is identified as $\mathbb{Z}^n$ by translation and scaling. Geometric objects such as curves and surfaces have been successfully represented as the isosurface of a discrete function. This implicit representation enables us to easily handle a moving geometric object, because updating a discrete function may have the same effect as moving the geometric object. This is the idea of the *level set method* presented by Osher and Sethian in 1989 [5]. They approximated an interface in $\mathbb{R}^n$ moving with curvature dependent speed by simply updating a discrete function $\phi : \mathbb{Z}^n \to \mathbb{R}$. In a recent application of the level set method to geometric optics [10], wave fronts in $\mathbb{R}^3$ were lifted into $\mathbb{R}^5$ to remove singularities and

☆ Research supported by ONR Grants N00014-02-1-0720, UCLA PY-2029 and NSF Grant DMS-0074735.
* Tel.: +1-310-486-4210.
E-mail address: chohong@math.ucla.edu.

0021-9991/$ - see front matter
doi:10.1016/S0021-9991(03)00275-4

represented as the isosurface of $\phi : \mathbb{Z}^5 \to \mathbb{R}^3$. Ambrosio and Soner [17] extended the theory of the level set method to arbitrary dimension. Based on the extended theory, Lorigo and Faugeras et al. [18] used the isosurface of $\phi : \mathbb{Z}^3 \to \mathbb{R}^2$ on the segmentation of blood vessels. Of all the successful methods above, the final step is to construct and visualize the isosurface of a discrete function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$, which is the purpose of this paper.

Before discussing the construction of the isosurface of a discrete function, we must give the definition of the isosurface. Since a discrete function is defined only on grid points, an interpolation should be given to define the isosurface of a discrete function as the isosurface of its interpolant. Historically there have developed two types of interpolations in parallel; one is piecewise interpolation on cubes and the other is on simplices. Lorenson and Cline in 1987 [7] presented a cube-based isosurfacing, *marching cubes* which works in $\mathbb{R}^3$. Bhaniramka et al. [12] extended the marching cubes to arbitrary dimension. On the other hand, Carneiro et al. [8] suggested a simplex-based isosurfacing algorithm, *tetra cubes* which works in $\mathbb{R}^3$. Weigle and Banks [13] proposed a simplex-based isosurfacing algorithm that works in arbitrary dimension and codimension.

The two types of isosurfacing algorithms have very different properties. Simplicial interpolation is naturally defined as the unique linear interpolant on a simplex, but cubic interpolation on a cube is often ambiguous and so is the definition of an isosurface. This is because a cube has 2^n vertices, while a simplex has $(n+1)$ vertices in $\mathbb{R}^n$. For this reason, Montani et al. [15] modified the marching cubes algorithm to remove the ambiguity of cubic interpolation and Bhaniramka et al. [12] in their paper extended their modifications to arbitrary dimension. Given a discrete function on a uniform grid, an isosurfacing algorithm based on cubic interpolation iteratively applies to each grid cell. But for a simplicial isosurfacing to be used, each grid cell should be decomposed into simplices. It is not known yet what is the minimum number of simplices for the decomposition of a cube [14]. Most optimal decomposition algorithms split a cube into $O(n!)$ simplices [1,2]. So the simplicial isosurfacing should iterate $O(n!)$ times more than cubic isosurfacings. If we compare programming complexities, we see that simplicial isosurfacing is much simpler than cubic isosurfacings. In $\mathbb{R}^3$, 2 types of isosurfaces exist on a 3-simplex, but 14 types on a 3-cube [7]. In $\mathbb{R}^4$, 2 types of isosurfaces exist on a 4-simplex, but 222 types on a 4-cube [12]. In $\mathbb{R}^n$, $[(n+1)/2]$ types of isosurfaces exist on a n-simplex, but it is hard even to classify the types of isosurfaces on an n-cube [12].

Weigle and Banks [13] proposed a simplicial isosurfacing algorithm that works in arbitrary dimension and codimension. We followed their framework to define the isosurface of a discrete function as the isosurface of its simplicial interpolant. As they pointed out in their paper [13], the isosurface Γ of a discrete function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$ is then a piecewise intersection of a simplex and m hyperplanes. We introduce a new construction algorithm of Γ that numerically costs $O(1)$ in time and space for each simplex. Our algorithm is explicit, while that of Weigle and Banks is recursive.

In Section 2, the simplicial interpolation algorithm is briefly reviewed. With the simplicial definition, the isosurface of $\phi : \mathbb{Z}^n \to \mathbb{R}^m$ is a piecewise intersection of a simplex and m hyperplanes in $\mathbb{R}^n$. In Section 3, we give the triangulation tables of the intersection of a simplex and a hyperplane. A counting theorem is stated to prove the optimality of the triangulation tables. From the triangulation tables, an intersection of a simplex and any number of hyperplanes can be constructed as the union of simplices. In Section 4, our main isosurfacing algorithm is presented. To visualize it, the isosurface in high dimension is often projected down to $\mathbb{R}^3$. In Section 5, we discuss a visualization of the projected isosurface. In Section 5, a dyadic grid is introduced to reduce the numerical costs. In Section 6, several numerical examples are given. In Section 7, we summarize our algorithms and discuss future work.

2. Simplicial interpolation

We introduce some definitions in computational geometry to review the simplicial interpolation algorithm [3,9]. An *affine set* is a translation of a vector space. A *hyperplane* is an $(n-1)$-dimensional affine set

in $\mathbb{R}^n$. A set of points is called *affinely independent*, if it is a translation of a linearly independent set. An *m-simplex* is the convex hull of $(m+1)$ affinely independent points, which are called *vertices* of the m-simplex. An l-simplex is called a *face* of a m-simplex S, if its vertices are vertices of S. A *triangulation* T of $D \subset \mathbb{R}^n$ is a finite collection of m-simplices such that $\bigcup_{\sigma\in T}\sigma = D$ and $\sigma_1 \cap \sigma_2$ is empty or a common face of σ_1 and σ_2, if $\sigma_1, \sigma_2 \in T$. An *affine map* is the composition of a linear map and a translation. An *n-cube* is the direct product of n intervals. A *polytope* is the convex hull of a set of finite points. Two sets are *affinely isomorphic*, if there exists a bijective affine map between them.

The simplicial interpolation is an operator $\wedge$ from discrete function space $C(\mathbb{Z}^n : \mathbb{R}^m)$ to the continuous function space $C(\mathbb{R}^n : \mathbb{R}^m)$. By a discrete function $\phi \in C(\mathbb{Z}^n : \mathbb{R}^m)$, we mean a vector valued function defined on a uniform grid, which is identified with $\mathbb{Z}^n$ by scaling and translation. To define the simplicial interpolation $\wedge$, an n-cube decomposition algorithm should be given. We choose the canonical Kuhn triangulation algorithm to decompose an n-cube into simplices.

2.1. Kuhn triangulation

Let us define S_n as the set of all permutations of $\{1,\ldots,n\}$, i.e., bijective maps from $\{1,\ldots,n\}$ onto $\{1,\ldots,n\}$. Let us define the standard n-cube $\bar{C} = [0,1]^n$. Given a permutation $J \in S_n$, we define a set $\bar{C}_J$ as

$$\bar{C}_J = \{x \in \bar{C} | 1 \geqslant x_{J(1)} \geqslant \cdots \geqslant x_{J(n)} \geqslant 0\}.$$

Then $\bar{C}_J$ is an n-simplex with the following vertices:

$$\begin{aligned}
&\bar{v}_0 = (0,\ldots,0),\\
&\bar{v}_1 = \bar{v}_0 + e_{J(1)},\\
&\vdots\\
&\bar{v}_n = \bar{v}_{n-1} + e_{J(n)}.
\end{aligned}$$

e_k is the kth canonical base of $\mathbb{R}^n$. $\bar{C} = \bigcup_{J\in S_n}\bar{C}_J$, because for any $x \in \bar{C}$, there is a permutation J such that $x_{J(1)} \geqslant \cdots \geqslant x_{J(n)}$. Since $|S_n| = n!$, $\bar{C}$ is the union of $n!$ simplices.

A general n-cube $C = [a_1, b_1] \times \cdots \times [a_n, b_n]$ is affinely isomorphic to $\bar{C}$ under an affine map $f : \bar{C} \to C$ such that

$$f(x) = \left(\frac{x_1 - a_1}{b_1 - a_1}, \ldots, \frac{x_n - a_n}{b_n - a_n}\right).$$

The decomposition $\bar{C} = \bigcup_{J\in S_n}\bar{C}_J$ is preserved to be a decomposition of C under the affine map f; $C = f^{-1}(\bar{C}) = \bigcup_{J\in S_n} f^{-1}(\bar{C}_J)$. Let us denote the set $f^{-1}(\bar{C}_J)$ as C_J. Then C_J is an n-simplex with the following vertices:

$$\begin{aligned}
&v_0 = (a_1,\ldots,a_n),\\
&v_1 = v_0 + \left(b_{J(1)} - a_{J(1)}\right)\cdot e_{J(1)},\\
&\vdots\\
&v_n = v_{n-1} + \left(b_{J(n)} - a_{J(n)}\right)\cdot e_{J(n)}.
\end{aligned}$$

Given a uniform grid on $\mathbb{R}^n$, we identify the grid as $\mathbb{Z}^n$ by translation and scaling. We define a grid cell $C^a = \{x \in \mathbb{R}^n \mid a_i \leqslant x_i \leqslant a_i + 1\}$ for each $a \in \mathbb{Z}^n$, which is an n-cube. Each grid cell C^a is decomposed into $n!$ simplices C^a_J. Then we have the following decomposition of $\mathbb{R}^n$ into n-simplices;

$$\mathbb{R}^n = \bigcup_{a\in\mathbb{Z}^n} C^a = \bigcup_{a\in\mathbb{Z}^n}\bigcup_{J\in S_n} C^a_J.$$

Furthermore, a set of n-simplices $\{C^a_J \,|\, a \in \mathbb{Z}^n, J \in S_n\}$ is a triangulation of $\mathbb{R}^n$ and called the *Kuhn triangulation* [3].

2.2. Interpolation procedure

Given a discrete function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$, its simplicial interpolant $\hat{\phi} : \mathbb{R}^n \to \mathbb{R}^m$ is piecewise defined on each simplex C^a_J in the Kuhn triangulation. In a simplex $C^a_J = \mathrm{conv}[v_0, \ldots, v_n]$, $\hat{\phi}|_{C^a_J}$ is defined as the linear interpolant of ϕ at $\{v_0, \ldots, v_n\}$. With the barycentric coordinates $\lambda = (\lambda_0, \ldots, \lambda_n)$ s.t. $\lambda_i \geqslant 0, \forall i$ and $\sum_{i=0}^n \lambda_i = 1$, $\hat{\phi}$ is defined as

$$\hat{\phi}\left(\sum_{i=0}^n \lambda_i v_i\right) = \sum_{i=0}^n \lambda_i \phi(v_i).$$

With the standard coordinates, there is a practical procedure to evaluate $\hat{\phi}$, as described by Kuhn [9]. We assume that the uniform grid is $\mathbb{Z}^n$. The following procedure can be applied to a general uniform grid by translation and scaling. Given $x \in \mathbb{R}^n$, $a \in \mathbb{Z}^n$ is defined as $a_i = [x_i]\,\forall i$, and $y \in [0,1)^n$ as $y = x - a$. Let J be a permutation of $\{1, \ldots, n\}$ such that $y_{J(1)} \geqslant \cdots \geqslant y_{J(n)}$, then

$$\begin{aligned}
\hat{\phi}(x) = {} & (1 - y_{J(1)}) \cdot \phi(a) \\
& + (y_{J(1)} - y_{J(2)}) \cdot \phi(a + e_{J(1)}) \\
& \vdots \\
& + (y_{J(n-1)} - y_{J(n)}) \cdot \phi(a + e_{J(1)} + \cdots + e_{J(n-1)}) \\
& + (y_{J(n)}) \cdot \phi(a + e_{J(1)} + \cdots + e_{J(n-1)} + e_{J(n)}).
\end{aligned}$$

If ϕ is scalar valued, $\nabla\hat{\phi}$ on the simplex C^a_J is easily calculated as

$$\begin{aligned}
\nabla\hat{\phi}_{J(1)} &= \frac{\phi(v_1) - \phi(v_0)}{b_{J(1)} - a_{J(1)}}, \\
\nabla\hat{\phi}_{J(2)} &= \frac{\phi(v_2) - \phi(v_1)}{b_{J(2)} - a_{J(2)}}, \\
&\vdots \\
\nabla\hat{\phi}_{J(n)} &= \frac{\phi(v_n) - \phi(v_{n-1})}{b_{J(n)} - a_{J(n)}}.
\end{aligned} \tag{1}$$

Now, we show that the simplicial interpolation operator $\wedge$ is an operator from discrete function space $C(\mathbb{Z}^n : \mathbb{R}^m)$ into continuous function space $C(\mathbb{R}^n : \mathbb{R}^m)$.

Theorem 2.1. *$\hat{\phi} : \mathbb{R}^n \to \mathbb{R}^m$ is a continuous function.*

Proof. $\hat{\phi}$ is continuous on the interior of C^a_J, because it is a polynomial. On $C^a_J \cap C^{a\prime}_{J'}$, $\hat{\phi}$ has two possible definitions $\hat{\phi}|_{C^a_J}$ and $\hat{\phi}|_{C^{a\prime}_{J'}}$. Since $\{C^a_J\}$ is a triangulation, $C^a_J \cap C^{a\prime}_{J'}$ is a common face of C^a_J and $C^{a\prime}_{J'}$. Every face of a simplex is also a simplex of lower dimension. Since there is only one linear interpolant on a simplex, the two definitions must be the same. □

Now, we define the isosurface Γ of a discrete function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$ as the isosurface of its simplicial interpolant $\hat{\phi} : \mathbb{R}^n \to \mathbb{R}^m$. We assume that Γ is the zero isosurface of $\hat{\phi}$ without losing generality. By this simplicial definition,

$$\begin{aligned}\Gamma &= \{x \in \mathbb{R}^n | \hat{\phi}(x) = 0\} \\ &= \bigcup_{a\in\mathbb{Z}^n} \bigcup_{J\in S_n} \left\{ x \in C_J^a | \hat{\phi}(x) = 0 \right\} \\ &= \bigcup_{a\in\mathbb{Z}^n} \bigcup_{J\in S_n} \left[C_J^a \cap \left\{ \hat{\phi}_1|_{C_J^a} = 0 \right\} \cap \cdots \cap \left\{ \hat{\phi}_m|_{C_J^a} = 0 \right\} \right].\end{aligned}$$

Since each component $\hat{\phi}_i$ of $\hat{\phi}$ is a first-order polynomial on each C_J^a, the set $\{\hat{\phi}_i|_{C_J^a} = 0\}$ is geometrically a hyperplane, and Γ is a piecewise intersection of a simplex and m hyperplanes. Γ will be piecewise constructed as the union of $(n-m)$-simplices in Section 4. Before constructing the Γ, the next section introduces a triangulation algorithm of the intersection of a simplex and a hyperplane.

3. Intersection of a simplex and a hyperplane

Let an n-simplex $S = \mathrm{conv}[v_1, \ldots, v_{n+1}]$ and a hyperplane $H = \{x \in \mathbb{R}^n | \psi(x) = 0\}$ be given with a first-order polynomial $\psi : \mathbb{R}^n \to \mathbb{R}$. Let us first assume that H does not include the vertices of S, i.e., $\psi(v_i) \neq 0, \forall i$. This assumption will be removed later in this section. Then $S \cap H$ is a polytope with vertices v_{ij}, where

$$v_{ij} \doteqdot \frac{\psi(v_i)}{\psi(v_i) - \psi(v_j)} \cdot v_j - \frac{\psi(v_j)}{\psi(v_i) - \psi(v_j)} \cdot v_i \quad \text{if } \psi(v_i) < 0,\ \psi(v_j) > 0. \tag{2}$$

v_{ij} is the interpolation point between v_i and v_j such that $\psi(v_{ij}) = 0$. The intersection $S \cap H$ is said to be type (p,q), if $\psi(v_i) < 0$ for p vertices and $\psi(v_j) > 0$ for q vertices. It is worth noting that all sections of the same type are isomorphic to each other [4]. Hence a triangulation of one specific intersection of type (p,q) can be applied to the triangulation of any intersection of type (p,q). An intersection of type (p,q) is naturally isomorphic to an intersection of type (q,p), because $\{\psi = 0\} = \{-\psi = 0\}$. Since H is assumed not to pass through any vertex of S, $p + q = n + 1$ and there exist $[(n+1)/2]$ types of intersection between S and H.

In this section, triangulation tables of the intersection are presented up to dimension five. Triangulation tables of two and three dimensions are easily generated by Figs. 1 and 2. $v_1, \ldots, v_{n+1}$ are vertices of S and $w_1, \ldots, w_n$ are vertices of a simplex in the triangulation of $S \cap H$ (see Tables 1 and 2).

Triangulation tables of four and five dimensions are generated by applying the Delaunay triangulation [6] to a specific section of each type (see Tables 3 and 4). Higher dimensional tables can be also generated in such a way.

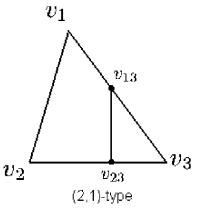

Fig. 1. Intersection of a 2-simplex and a hyperplane.

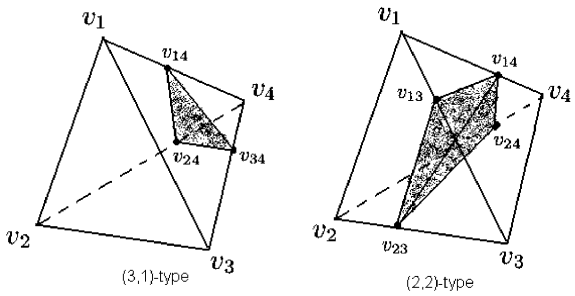

Fig. 2. Intersections of a 3-simplex and a hyperplane.

Table 1
Triangulation of the intersection between a 2-simplex and a hyperplane

v_1	v_2	v_3	w_1	w_2
−	−	+	v_{13}	v_{23}

Table 2
Triangulation of the intersection between a 3-simplex and a hyperplane

v_1	v_2	v_3	v_4	w_1	w_2	w_3
−	−	−	+	v_{14}	v_{24}	v_{34}
−	−	+	+	v_{13}	v_{23}	v_{14}
				v_{14}	v_{23}	v_{24}

Table 3
Triangulation of the intersection between a 4-simplex and a hyperplane

v_1	v_2	v_3	v_4	v_5	w_1	w_2	w_3	w_4
−	−	−	−	+	v_{15}	v_{25}	v_{35}	v_{45}
−	−	−	+	+	v_{14}	v_{15}	v_{34}	v_{24}
					v_{15}	v_{24}	v_{25}	v_{34}
					v_{15}	v_{25}	v_{35}	v_{34}

Table 4
Triangulation of the intersection between a 5-simplex and a hyperplane

v_1	v_2	v_3	v_4	v_5	v_6	w_1	w_2	w_3	w_4	w_5
−	−	−	−	−	+	v_{16}	v_{26}	v_{36}	v_{56}	v_{46}
−	−	−	−	+	+	v_{15}	v_{16}	v_{25}	v_{45}	v_{35}
						v_{16}	v_{25}	v_{26}	v_{35}	v_{45}
						v_{16}	v_{26}	v_{35}	v_{45}	v_{36}
						v_{16}	v_{26}	v_{36}	v_{45}	v_{46}
−	−	−	+	+	+	v_{14}	v_{15}	v_{16}	v_{36}	v_{25}
						v_{14}	v_{15}	v_{25}	v_{36}	v_{35}
						v_{14}	v_{16}	v_{25}	v_{26}	v_{36}
						v_{14}	v_{24}	v_{25}	v_{34}	v_{26}
						v_{14}	v_{25}	v_{26}	v_{36}	v_{34}
						v_{14}	v_{25}	v_{34}	v_{36}	v_{35}

We show that the preceding tables are optimal in a sense that the number of simplices in each triangulation can not be reduced. First we quote a theorem in Haiman's paper [1], then our statement follows as a corollary.

Theorem 3.1. *Let Δ_k be a k-simplex and Δ_l a l-simplex. Every triangulation of $\Delta_k \times \Delta_l$ uses exactly $(k+l)!/k!l!$ simplices.*

Proof. Since every k-simplex is affinely isomorphic to any other, let us pick the standard k-simplex with vertices $e_1, e_2, \ldots, e_{k+1} \in \mathbb{R}^{k+1}$. Every simplex of a triangulation of $\Delta_k \times \Delta_l$ has the volume $1/(k!l!)$. Since the volume of $\Delta_k \times \Delta_l$ is $1/(k+l)!$, every triangulation of $\Delta_k \times \Delta_l$ uses exactly $(k+l)!/k!l!$ simplices. See [1] for details. □

Corollary 3.2. *Every triangulation of (p,q)-type section uses exactly $(p+q-2)!/((p-1)!(q-1)!)$ simplices.*

Proof. Since all sections of the same type are isomorphic to each other, we can choose a specific section of (p,q)-type. Let a simplex $S = \mathrm{conv}[v_1, \ldots, v_{p+q}]$ and a hyperplane $H = \{\psi(x) = 0\}$ be given such that

$$\psi(v_1) = \cdots = \psi(v_p) = -1,$$
$$\psi(v_{p+1}) = \cdots = \psi(v_{p+q}) = 1.$$

With the barycentric coordinates $x(\lambda_1, \ldots, \lambda_{p+q}) = \sum_{k=1}^{p+q} \lambda_k v_k$,

$$\psi(x(\lambda)) = \psi\left(\sum_{k=1}^{p+q} \lambda_k v_k\right) = \sum_{k=1}^{p+q} \lambda_k \psi(v_k) = \sum_{j=p+1}^{p+q} \lambda_j - \sum_{i=1}^{p} \lambda_i.$$

H, S, and $H \cap S$ are algebraically expressed with the barycentric coordinates:

$$x(\lambda) \in H \iff \sum_{i=1}^{p} \lambda_i = \sum_{j=p+1}^{p+q} \lambda_j,$$
$$x(\lambda) \in S \iff \sum_{k=1}^{p+q} \lambda_k = 1, \quad \lambda_k \geqslant 0,$$
$$x(\lambda) \in S \cap H \iff \sum_{i=1}^{p} \lambda_i = \sum_{i=p+1}^{p+q} \lambda_j = \frac{1}{2}, \quad \lambda_i,\ \lambda_j \geqslant 0.$$

Since coordinates λ_i and λ_j are decoupled in the expression of $S \cap H$:

$$S \cap H \simeq \left\{ (\lambda_1, \ldots, \lambda_p) \,\middle|\, \sum_{i=1}^{p} \lambda_i = \frac{1}{2},\ \lambda_i \geqslant 0 \right\} \times \left\{ (\lambda_{p+1}, \ldots, \lambda_{p+q}) \,\middle|\, \sum_{j=p+1}^{p+q} \lambda_j = \frac{1}{2},\ \lambda_j \geqslant 0 \right\}.$$

Hence $S \cap H$ is isomorphic to $\Delta_{p-1} \times \Delta_{q-1}$. By the theorem above, every triangulation of $S \cap H$ should use $(p+q-2)!/((p-1)!(q-1)!)$ simplices. □

By the triangulation tables, we can construct the intersection of an n-simplex and a hyperplane as the union of $(n-1)$-simplices. Here is the construction procedure. Given an n-simplex $S = \mathrm{conv}[v_1, \ldots, v_{n+1}]$ and a hyperplane $H = \{\psi = 0\}$, signums of $\psi(v_1), \ldots, \psi(v_{n+1})$ are counted. If there are more positive signums than negative, ψ is inverted; $\psi = -\psi$. Let p be the number of negative signums and q be the number of positive, then $S \cap H$ is an intersection of type (p,q). The vertices of S are reordered such that $\psi(v_1), \ldots, \psi(v_p) < 0$ and $\psi(v_{p+1}), \ldots, \psi(v_{n+1}) > 0$. Interpolation points v_{ij} (2) are calculated for $i = 1, \ldots, p$

and $j = p+1, \ldots, n+1$. Referencing the triangulation table of (p,q)-type, $S \cap H$ is constructed as the union of some simplices of one lower dimension:

$$S \cap H = S^1 \cup \cdots \cup S^l. \tag{3}$$

Numerically only vertices of simplices are stored. We note that this triangulation procedure is numerically finite in space and time, because the number of simplices is bounded by the tables. Also the procedure is numerically stable, since the only numerical calculation is to interpolate inner points v_{ij} (2) between v_i and v_j. Practically if H passes through a vertex v_i of S, we perturb ψ by $\psi(v_i) = \epsilon$, where $0 < \epsilon \ll 1$.

4. Isosurfacing

We define the isosurface Γ of a discrete function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$ as the isosurface of its simplicial interpolant $\hat{\phi} : \mathbb{R}^n \to \mathbb{R}^m$. We may assume that Γ is the zero isosurface of $\hat{\phi}$ without losing generality. By this simplicial definition,

$$\begin{aligned}
\Gamma &= \{x \in \mathbb{R}^n | \hat{\phi}(x) = 0\}, \\
&= \bigcup_{a \in \mathbb{Z}^n} \bigcup_{J \in S_n} \{x \in C_J^a | \hat{\phi}(x) = 0\}, \\
&= \bigcup_{a \in \mathbb{Z}^n} \bigcup_{J \in S_n} \left[C_J^a \cap \{\hat{\phi}_1|_{C_J^a} = 0\} \cap \cdots \cap \{\hat{\phi}_m|_{C_J^a} = 0\} \right].
\end{aligned}$$

Since each component $\hat{\phi}_i : \mathbb{R}^n \to \mathbb{R}$ of $\hat{\phi}$ is a first-order polynomial on each simplex C_J^a, the set $\{\hat{\phi}_i|_{C_J^a} = 0\}$ is geometrically a hyperplane, and Γ is the intersection of a simplex and m hyperplanes. In Section 3, an algorithm was presented to construct the intersection of a simplex and a hyperplane as the union of simplices of one lower dimension. Weigle and Banks pointed out that the intersection of a simplex and several hyperplanes can be constructed as the union of simplices by successively applying the algorithm [13]. On a simplex C_J^a, let us say $H^1 = \{\hat{\phi}_1|_{C_J^a} = 0\}, \ldots, H^m = \{\hat{\phi}_m|_{C_J^a} = 0\}$. Then $C_J^a \cap H^1 \cap \cdots \cap H^m$ is constructed by the following procedure:

$$\begin{aligned}
&C_J^a \cap H^1 = \bigcup_{j=1}^{n_1} S^{1,j}, \\
&C_J^a \cap H^1 \cap H^2 = \bigcup_{j=1}^{n_1} [S^{1,j} \cap H^2] = \bigcup_{j=1}^{n_2} S^{2,j}, \\
&\vdots \\
&C_J^a \cap H^1 \cap \cdots \cap H^m = \bigcup_{j=1}^{n_{m-1}} [S^{m-1,j} \cap H^m] = \bigcup_{j=1}^{n_m} S^{m,j}.
\end{aligned}$$

We get a set of $(n-m)$-simplices, $T_{a,J} = \{S^{m,j} | j = 1, \ldots, n_m\}$ such that $C_J^a \cap \Gamma = C_J^a \cap H^1 \cap \cdots \cap H^m = \bigcup_{\sigma \in T_{a,J}} \sigma$. We note that it is numerically finite in time and space to construct the set $T_{a,J}$. For example, the intersection of a 5-simplex and 3 hyperplanes is constructed as the union of up to $6 \times 3 \times 2$ number of 2-simplices, where 6, 3, 2 are maximum number of simplices in the triangulation tables of 5, 4, 3 dimensions. That means $n_1 \leqslant 6$, $n_2 \leqslant 6 \times 3$, and $n_3 \leqslant 6 \times 3 \times 2$.

Iterating all the simplices C_J^a in the Kuhn triangulation, Γ is constructed as the union of $(n-m)$-simplices:

$$\begin{aligned}\Gamma &= \bigcup_{a\in\mathbb{Z}^n}\bigcup_{J\in S_n}[\Gamma\cap C_J^a]\\ &= \bigcup_{a\in\mathbb{Z}^n}\bigcup_{J\in S_n}\bigcup_{\sigma\in T_{a,J}}\sigma.\end{aligned} \tag{4}$$

5. Visualization

In Section 4, an algorithm was introduced to construct the isosurface Γ of a discrete function $\phi:\mathbb{Z}^n\to\mathbb{R}^m$ for any dimension n and any codimension m. Γ is an $(n-m)$-dimensional manifold in its smooth region. To visualize it, Γ is often projected down into $\mathbb{R}^3$ under a projection map $P:\mathbb{R}^n\to\mathbb{R}^3$. Since a projection is an affine map, it maps a simplex to a simplex. From the isosurfacing algorithm (4), $P(\Gamma)$ is the union of projected $(n-m)$-simplices;

$$P(\Gamma)=\bigcup_{a\in\mathbb{Z}^n}\bigcup_{J\in S_n}\bigcup_{\sigma\in T_{a,J}}P(\sigma). \tag{5}$$

When $n=m+1$ or $m+2$, $P(\Gamma)$ is the union of line segments or triangles, which can be visualized by usual graphic libraries such as OpenGL. But, for a surface visualization in $\mathbb{R}^3$, the normal vector field is needed for shadowings. Now we present a simple formula calculating the normal vector field of $P(\Gamma)$.

In a general framework, let $P:\mathbb{R}^n\to\mathbb{R}^{n-m+2}$ be a projection such that $P(x_1,\ldots,x_n)=(x_1,\ldots,x_{n-m+2})$. In $\mathbb{R}^n$, $\nabla\hat{\phi}_1,\ldots,\nabla\hat{\phi}_m$ form the normal vector space of Γ in smooth region. Let us define a vector field $\vec{n}:\mathbb{R}^n\to\mathbb{R}^n$ as

$$\vec{n}=\begin{vmatrix}\nabla\hat{\phi}_1 & \nabla\hat{\phi}_1\cdot e_{n-m+2} & \cdots & \nabla\hat{\phi}_1\cdot e_n\\ \vdots & \vdots & & \vdots\\ \nabla\hat{\phi}_m & \nabla\hat{\phi}_m\cdot e_{n-m+2} & \cdots & \nabla\hat{\phi}_m\cdot e_n\end{vmatrix}. \tag{6}$$

$\nabla\hat{\phi}_j$ is calculated by the divided difference formula (1) and the determinant is expanded with cofactors:

$$\vec{n}=\nabla\hat{\phi}_1\cdot\begin{vmatrix}\nabla\hat{\phi}_2\cdot e_{n-m+2} & \cdots & \nabla\hat{\phi}_2\cdot e_n\\ \vdots & & \vdots\\ \nabla\hat{\phi}_m\cdot e_{n-m+2} & \cdots & \nabla\hat{\phi}_m\cdot e_n\end{vmatrix}+\cdots+(-1)^m\nabla\hat{\phi}_m\cdot\begin{vmatrix}\nabla\hat{\phi}_1\cdot e_{n-m+2} & \cdots & \nabla\hat{\phi}_1\cdot e_n\\ \vdots & & \vdots\\ \nabla\hat{\phi}_{m-1}\cdot e_{n-m+2} & \cdots & \nabla\hat{\phi}_{m-1}\cdot e_n\end{vmatrix}.$$

Since $\vec{n}$ is a linear combination of $\nabla\hat{\phi}_1,\ldots,\nabla\hat{\phi}_m$, it is a normal vector field of Γ. By the property of determinant, $\vec{n}\cdot e_j=0$ if $j=n-m+2,\ldots,n$. Given a tangential vector T of Γ at $x\in\Gamma$, $0=\vec{n}(x)\cdot T=\vec{n}(x)\cdot P(T)$. Since the projection π is surjective, it is also surjective in tangential vector space. Therefore $\vec{n}$ is orthogonal to any tangent vector of $\pi(\Gamma)$ and it is the normal vector field of $P(\Gamma)$ in $\mathbb{R}^{n-m+1}$. For example, when $\phi:\mathbb{Z}^5\to\mathbb{R}^3$ and $P:\mathbb{R}^5\to\mathbb{R}^3$, the normal vector $\vec{n}$ is given by

$$\begin{aligned}\vec{n}=&\nabla\hat{\phi}_1((\nabla\hat{\phi}_2\cdot e_4)(\nabla\hat{\phi}_3\cdot e_5)-(\nabla\hat{\phi}_3\cdot e_4)(\nabla\hat{\phi}_2\cdot e_5))\\ &-\nabla\hat{\phi}_2((\nabla\hat{\phi}_1\cdot e_4)(\nabla\hat{\phi}_3\cdot e_5)-(\nabla\hat{\phi}_3\cdot e_4)(\nabla\hat{\phi}_1\cdot e_5))\\ &+\nabla\hat{\phi}_3((\nabla\hat{\phi}_1\cdot e_4)(\nabla\hat{\phi}_2\cdot e_5)-(\nabla\hat{\phi}_2\cdot e_4)(\nabla\hat{\phi}_1\cdot e_5)).\end{aligned}$$

6. Dyadic grid

Let Γ be the isosurface of $\phi : \mathbb{Z}^n \to \mathbb{R}^m$. When Γ is bounded, the domain of ϕ can be reduced, without changing Γ, to $\{-N,\ldots,N\}^n$ for some large N bounding Γ. The isosurfacing in Section 4 numerically costs $O(1)$ in space and time for each simplex. So, the construction of Γ numerically costs $O(N^n)$ in time and space. This is practically too much when $n \geqslant 4$. We can significantly decrease the costs by removing the domain of ϕ, where $\|\phi\|$ is large. Let L be the Lipschitz constant of ϕ defined as

$$L = \max_{a,b\in\{-N,\ldots,N\}^n} \frac{\|\phi(b)-\phi(a)\|_\infty}{\|b-a\|_\infty}.$$

Given an n-cube $C = \{a_i \leqslant x_i \leqslant b_i\}$ with $a, b \in \{-N,\ldots,N\}^n$, $C \cap \mathbb{Z}^n$ can be omitted from the domain of ϕ without changing Γ, if $\|\phi(v)\|_\infty > L \cdot \|b-a\|_\infty$ for some vertex v of C. Based on such an observation, Tsai et al. [19] introduced a multi-scale grid and Strain [11] introduced a dyadic grid to reduce the domain of a scalar valued function. We extend Strain's idea to a vector valued function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$.

Let $D \subset \mathbb{Z}^n$ be the domain of ϕ. First, we set $D = \{-N,N\}^n$, a set of vertices of $[-N,N]^n$. If $\|\phi(v)\|_\infty > L \cdot 2N$ for some $v \in D$, then $\Gamma \cap [-N,N]^n = \emptyset$ and we have nothing to do more. When $\|\phi(D)\|_\infty \leqslant L \cdot 2N$, the n-cube $[-N,N]^n$ is decomposed into 2^n number of n-cubes;

$$[-N,N]^n = \bigcup_{a\in\{0,N\}^n} a + [-N,0]^n.$$

Then, $a + \{-N,0\}^n$ is added to the domain D if $\|\phi(a+\{-N,0\}^n)\|_\infty \leqslant L \cdot N$. This process of $[-N,N]^n$ is recursively applied to every n-cube whose vertices are included in D. Therefore the domain D is numerically implemented as a 2^n branched tree with maximum depth $\log(N)$. Since Γ is $(n-m)$ dimensional, the size of dyadic grid is $O(N^{n-m})$. To maintain dyadic tree structure, optional spaces are required in the dyadic grid. It causes the increase in size from $O(N^{n-m})$ to $O(N^{n-m} \cdot \log(N))$. Hence the isosurfacing algorithm numerically costs $O(N^{n-m} \cdot \log(N))$ with a dyadic grid and $O(N^n)$ with a uniform grid in space and time.

For example, Fig. 3 illustrates dyadic grids, when $\phi : \{-N,\ldots,N\}^2 \to \mathbb{R}$ is taken as $\phi(i,j) = \max(|a_i|,|a_j|)$ with $a_i = -1 + 2 \cdot i/N$.

7. Numerical examples

Every example was implemented in C++ and run on a PC with 2.2 GHz CPU and 512 MB memory. All isosurfaces were constructed by the isosurfacing algorithm in Section 4. 1-simplices or 2-simplices in the

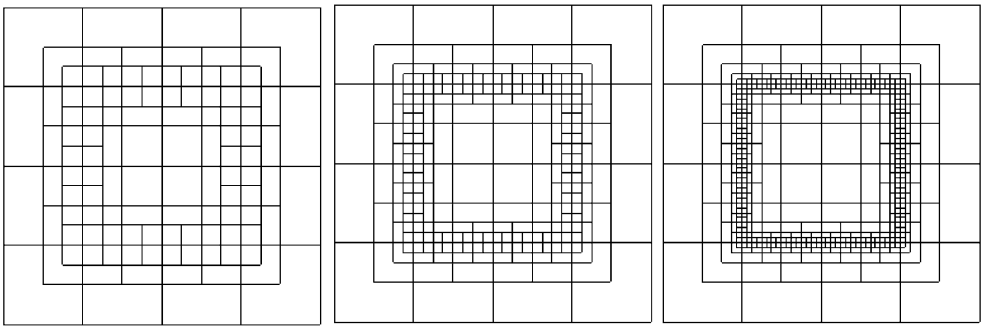

Fig. 3. Dyadic grids when $N = 8, 16, 32$.

construction were visualized by the OpenGL graphic library. For 2-simplices, the normal vector field in Section 5 was used for shadowing.

7.1. Uniform grid vs. dyadic grid

We compare the numerical costs of isosurfacing between a uniform grid and a dyadic grid. On a domain $[-1,1]^3$, We take $x_i = (i-N)/N$, $y_j = (j-N)/N$, $\theta_k = (k-N)/N$, and $\phi : \{0,\ldots,N\}^3 \to \mathbb{R}^2$ as

$$\phi(i,j,k) = \begin{pmatrix} x_i - \cos(\pi\cdot\theta_k) \\ y_j - \sin(\pi\cdot\theta_k) \end{pmatrix}.$$

We can see that the numerical costs are $O(N^3)$ in the uniform grid and $O(N\cdot\log(N)) \approx O(N)$ in the dyadic grid, as expected (see Figs. 4–6 and Tables 5 and 6).

7.2. A singularity resolves in $\mathbb{R}^2$

Let a curve $\Gamma \subset \mathbb{R}^2$ be given by $r = 1+\sin(7\theta)$ in the polar coordinates. Γ is a 7-leafed curve and singular at the origin, where 14 curves meet. In this case, we can resolve the singularity by adding a phase

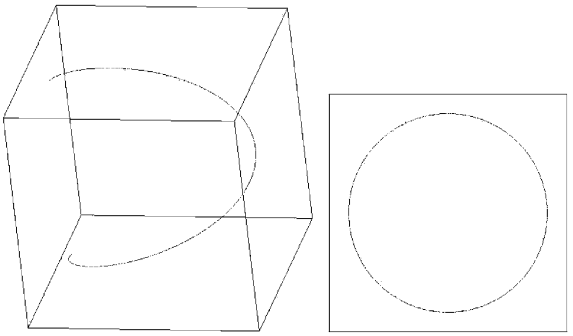

Fig. 4. $\Gamma \subset \mathbb{R}^3$ and its projection on $\mathbb{R}^2$.

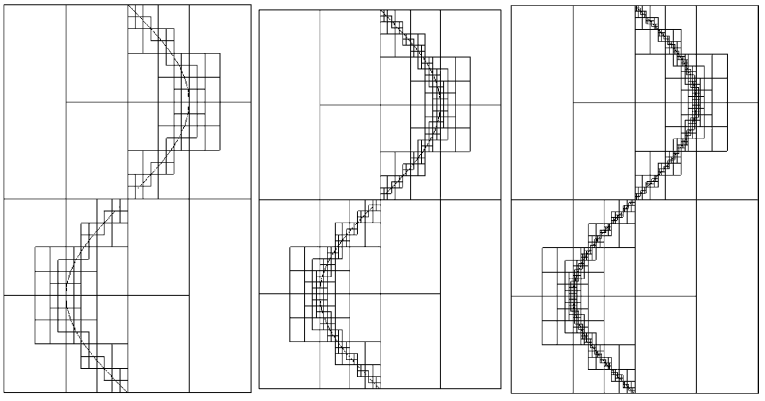

Fig. 5. Dyadic grids (side view) $N = 32, 64, 128$.

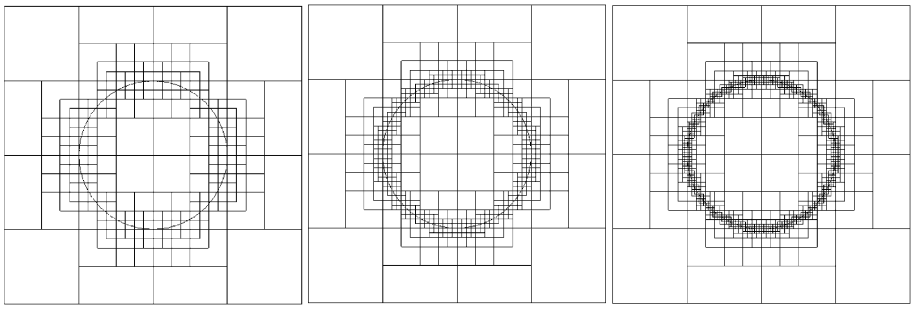

Fig. 6. Dyadic grids (front view) $N = 32, 64, 128$.

Table 5
Numerical costs in uniform grid

N	Time (ms)	Rate	Space for Γ (byte)	Space for ϕ (byte)	Total space	Rate
32	78		11,376	524,288	535,664	
64	609	2.96	24,060	4,194,304	4,218,364	2.98
128	4265	2.81	48,012	33,554,432	33,602,444	2.99
256	34,360	3.01	71,630	268,435,456	268,507,086	3.00

Table 6
Numerical costs in dyadic grid

N	Time (ms)	Rate	Space for Γ (byte)	Space for ϕ (byte)	Total space	Rate
32	5.4		11,952	69,384	81,336	
64	10.8	1.00	25,488	145,992	171,480	1.08
128	21.4	0.99	51,120	293,832	344,952	1.01
256	41.8	0.97	97,776	574,728	672,504	0.96

variable θ, as described by Osher et al. [10] and Engquist et al. [16]. With a new coordinate system (x, y, θ), we define $\Gamma' \subset \mathbb{R}^3$ as the solution of the following equations:

$$\begin{cases} x = (1 + \sin(7\theta)) \cdot \cos(\theta), \\ y = (1 + \sin(7\theta)) \cdot \sin(\theta). \end{cases}$$

Then $\Gamma = \pi(\Gamma')$, where $\pi(x, y, \theta) = (x, y)$. In Figs. 7 and 8, Γ and Γ' are approximated as the isosurfaces of discrete functions $\psi : \{1, \ldots, 128\}^2 \to \mathbb{R}$ and $\phi : \{1, \ldots, 128\}^3 \to \mathbb{R}^2$, respectively, where $a_i = -1 + 2 \cdot i/128$, $\theta_k = 2\pi \cdot k/128$ and

$$\psi(i, j) = \sqrt{a_i^2 + a_j^2} - 1 - \sin\left(7 \cdot \tan^{-1}\left(\frac{a_j}{a_i}\right)\right),$$

$$\phi(i, j, k) = \begin{pmatrix} a_i - (1 + \sin(7\theta_k)) \cdot \cos(\theta_k) \\ a_j - (1 + \sin(7\theta_k)) \cdot \sin(\theta_k) \end{pmatrix}.$$

7.3. A singularity resolves in $\mathbb{R}^3$

Let a surface $\Gamma \subset \mathbb{R}^3$ be given by $r(\theta, \varphi) = 1 + \cos(2\varphi)$ in the spherical coordinates (r, θ, φ). Γ is a surface of revolution obtained by rotating 2-leaf curve around the z-axis and it is singular at the origin. In

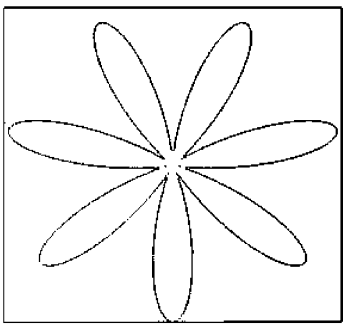

Fig. 7. Approximation of the singular curve $\Gamma \subset \mathbb{R}^2$.

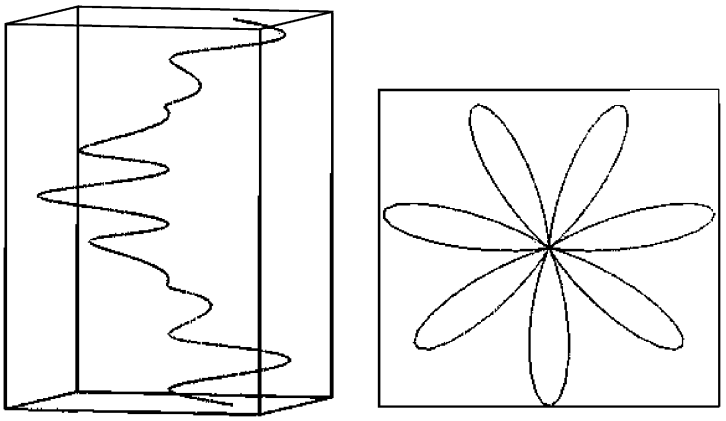

Fig. 8. Approximations of the non-singular curve $\Gamma' \subset \mathbb{R}^3$ and $\pi(\Gamma') \subset \mathbb{R}^2$.

this case, we can resolve the singularity by adding phase variables θ and φ, as described by Osher et al. [10]. This resolution is a direct generalization of the technique used in Section 7.2. With a new coordinate system $(x, y, z, \theta, \varphi)$, we define $\Gamma' \subset \mathbb{R}^5$ as the solution of equations:

$$\begin{cases} x = (1 + \cos(2\varphi)) \cdot \cos(\theta) \cdot \sin(\varphi), \\ y = (1 + \cos(2\varphi)) \cdot \sin(\theta) \cdot \sin(\varphi), \\ z = (1 + \cos(2\varphi)) \cdot \cos(\varphi). \end{cases}$$

Then $\Gamma = \pi(\Gamma')$, where $\pi(x, y, z, \theta, \varphi) = (x, y, z)$. In Fig. 9, Γ and Γ' are approximated as the isosurfaces of discrete functions $\psi : \{1, \ldots, 32\}^3 \to \mathbb{R}$ and $\phi : \{1, \ldots, 32\}^5 \to \mathbb{R}^2$, respectively, where $a_i = -2 + \frac{4}{32}\mathrm{i}$, $\theta_m = 2\pi \cdot \frac{m}{32}$, $\varphi_n = \pi \cdot \frac{n}{32}$ and

$$\psi(i, j, k) = \sqrt{a_i^2 + a_j^2 + a_k^2} - 1 - \cos\left(2\tan^{-1}\left(\frac{a_j}{a_i}\right)\right),$$

$$\phi(i, j, k, m, n) = \begin{pmatrix} a_i - (1 + \cos(2\varphi_n)) \cdot \cos(\theta_m) \cdot \sin(\varphi_n) \\ a_j - (1 + \cos(2\varphi_n)) \cdot \sin(\theta_m) \cdot \sin(\varphi_n) \\ a_k - (1 + \cos(2\varphi_n)) \cdot \cos(\varphi_n) \end{pmatrix}.$$

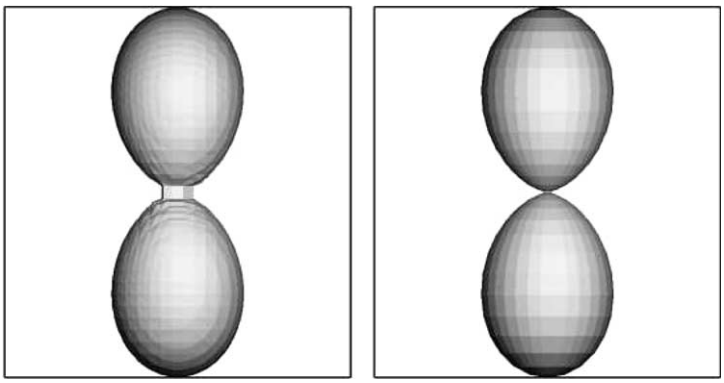

Fig. 9. Approximations of $\Gamma \subset \mathbb{R}^3$ and $\pi(\Gamma') \subset \mathbb{R}^3$.

The dyadic grid in Section 6 is used for the domain of ϕ with $Lip(\phi) = 3.5$. The numerical cost of Γ' is almost $O(N^2) \simeq O(N^2 \log(N))$, as expected (see Table 7).

7.4. Algebraic curves in $\mathbb{C}^2$

Let an algebraic curve $\Gamma \subset \mathbb{C}^2$ is defined by $\{(z_1, z_2) \in \mathbb{C}^2 \mid f(z_1, z_2) = 0\}$, where $f : \mathbb{C}^2 \to \mathbb{C}$ is a polynomial. In their paper [13], Weigle and Banks pointed out that Γ can be represented by the isosurface of $\bar{f} : \mathbb{R}^4 \to \mathbb{R}^2$, where

$$\bar{f}(x, y, p, q) = \begin{pmatrix} \mathrm{Re}\,[f(x + \mathrm{i}y, p + \mathrm{i}q)] \\ \mathrm{Im}\,[f(x + \mathrm{i}y, p + \mathrm{i}q)] \end{pmatrix}.$$

In this example, f is chosen as $f(z_1, z_2) = z_1^2 + z_2^2 - 1$ and $\Gamma \subset \mathbb{R}^4$ is approximated as the isosurface of a sampled discrete function from $\bar{f}$ in 32^4 uniform grid. The approximation of $\Gamma \subset \mathbb{R}^4$ is then projected down into (x, y, p) and (x, y, q) coordinates (see Fig. 10).

7.5. Intersection of two spheres in $\mathbb{R}^4$

We test our algorithms in an example, where the true solution is known. Let two spheres $S_1$and S_2 in $\mathbb{R}^4$ be given as

$$\begin{cases} x^2 + y^2 + z^2 + \left(w - \frac{1}{2}\right)^2 = 1, \\ x^2 + y^2 + z^2 + \left(w + \frac{1}{2}\right)^2 = 1. \end{cases}$$

Table 7
Numerical costs for Γ'

N	Space for ϕ (byte)	Rate	Space for Γ (byte)	Rate
8	2,977,680		2,221,344	
16	11,820,432	1.99	9,398,736	2.08
32	48,300,432	2.03	37,655,568	2.00
64	202,640,016	2.07	148,883,472	1.98

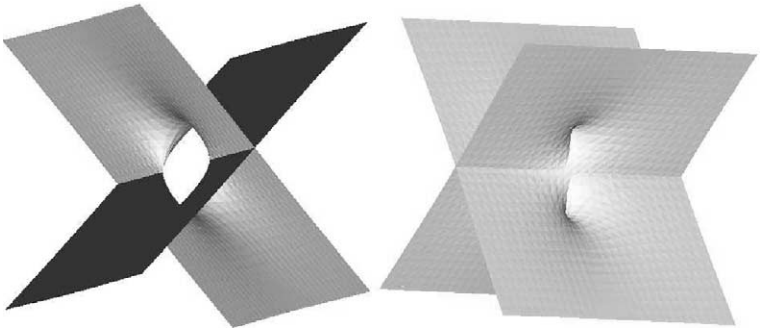

Fig. 10. Projections of an algebraic curve in $\mathbb{C}^2$.

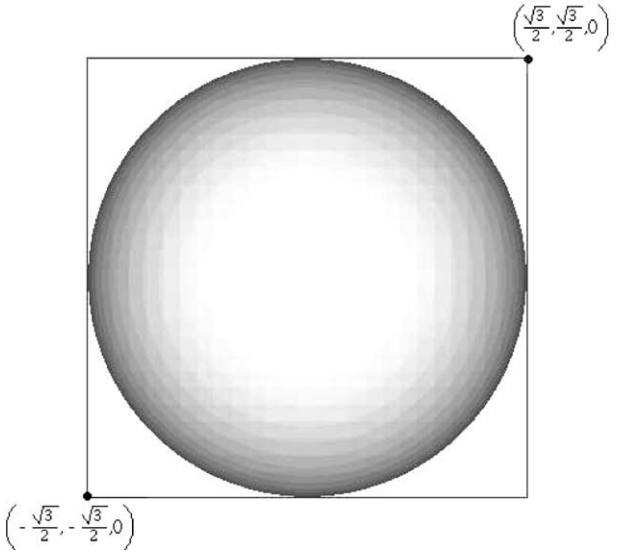

Fig. 11. Projection of the intersection of two spheres in $\mathbb{R}^4$ when $N = 40$.

$S_1 \cap S_2$ is algebraically given as $x^2 + y^2 + z^2 = 3/4$. Numerically $S_1 \cap S_2$ is approximated as the isosurface of $\phi : \{0, \ldots, N\}^4 \to \mathbb{R}^2$ such that

$$\phi(i,j,k,l) = \begin{cases} a_i^2 + a_j^2 + a_k^2 + (a_l - \frac{1}{2})^2 - 1, \\ a_i^2 + a_j^2 + a_k^2 + (a_l + \frac{1}{2})^2 - 1, \end{cases}$$

where $a_i = -1.3 + 2.6 \cdot \mathrm{i}/(N-1)$. The intersection is then projected down into $\mathbb{R}^3$ (see Fig. 11).

8. Conclusion

We have presented an isosurfacing algorithm that works in an arbitrary number of dimensions and codimensions. For a discrete function $\phi : \{0, \ldots, N\}^n \to \mathbb{R}^m$, the isosurface Γ of ϕ is defined as the isosurface of its simplicial interpolant $\hat{\phi} : [0, \ldots, N]^n \to \mathbb{R}^m$. Geometrically Γ is a piecewise intersection of a

simplex and m hyperplanes. Triangulation tables are given of the intersection of a simplex and a hyperplane. A counting theorem was stated to prove the optimality of the tables. Referencing the tables, Γ is constructed as the union of simplices. The construction costs $O(N^n)$ with a uniform grid and $O(N^{n-m}\log(N))$ with a dyadic grid in the numerical space and time. When $n = m + 1$ or $m + 2$, $\Gamma \subset \mathbb{R}^n$ is projected down into $\mathbb{R}^3$ and can be visualized. A simple formula is introduced calculating the normal vector field of the projected surface, when $n = m + 2$.

Acknowledgements

I thank Prof. Stanley Osher for his encouragement and guidance throughout this project. I am indebted to Chris Weigle and Prof. David Ebert for their kind suggestions and comments. I am also grateful to Prof. Mark Green, Thomas Cecil, Yen Hsi Tsai, Yon Seo Kim, and Chiu Yen Kao for their help.

References

[1] M. Haiman, A simple and relatively efficient triangulation of the n-cube, Discrete Comput. Geom. 6 (1991) 287–289.
[2] J.F. Sallee, A triangulation of the n-cube, Discrete Math. 40 (1982) 81–86.
[3] E.L. Allgower, P.H. Schmidt, An algorithm for piecewise-linear approximation of an implicitly defined manifold, SIAM J. Numer. Anal. 22 (1985) 322–346.
[4] D.M.Y. Sommerville, An Introduction to the Geometry of n-Dimensionals, Dover, New York, 1958.
[5] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent spee, J. Comput. Phys. 79 (1988) 12–49.
[6] G.M. Ziegler, Lectures on Polytopes, Springer, Berlin, 1995.
[7] W.E. Lorenson, H.E. Cline, Maching cubes: a high resolution 3d surface reconstruction algorithm, Comput. Graphics 21 (1987) 163–169.
[8] B.P. Carneiro, C. Silva, A.E. Kaufman, Tetra-cubes: an algorithm to generate 3d isosurfaces based upon tetrahedra, Anais do IX SIBGRAPI (1996) 205–210.
[9] H.W. Kuhn, Some combinatorial lemmas in topology, IBM J. Res. Dev. 4 (1960) 518–524.
[10] S. Osher, L.T. Cheng, M. Kang, H. Shim, Y.H. Tsai, Geometric optics in a phase-space-based level set and Eulerian framework, J. Comput. Phys. 179 (2002) 622–648.
[11] J. Strain, Tree methods for moving interfaces, J. Comput. Phys. 151 (1999) 616–648.
[12] P. Bhaniramka, R. Wenger, R. Crawfis, Isosurfacing in higher dimensions, Proc. Visual (2000) 267–273.
[13] C. Weigle, D. Banks, Complex-valued contour meshing, IEEE Visual. (1996) 173–180.
[14] R.B. Hughes, M.R. Anderson, Simplexity of the cube, Discrete Math. 158 (1996) 99–150.
[15] C. Montani, R. Scateni, R. Scopigno, A modified look-up table for implicit disambiguation of marching cubes, Visual Comput. 10 (1994) 353–355.
[16] B. Engquist, O. Runborg, A.-K. Tornberg, High frequency wave propagation by the segment projection method, J. Comput. Phys 178 (2002) 373–390.
[17] L. Ambrosio, H.M. Soner, Level set approach to mean curvature flow in arbitrary dimension, J. Diff. Geom. 43 (1996) 693–737.
[18] L. Lorigo, O. Faugeras, Co-dimension 2 geodesic contours for MRA segmentation, Proc. IPMI (1999).
[19] Y.R. Tsai, L.T. Tien, P. Burchard, S. Osher, G. Sapiro, Dynamic Visibility in an implicit framework, UCLA CAM Report 02–06 (2002).