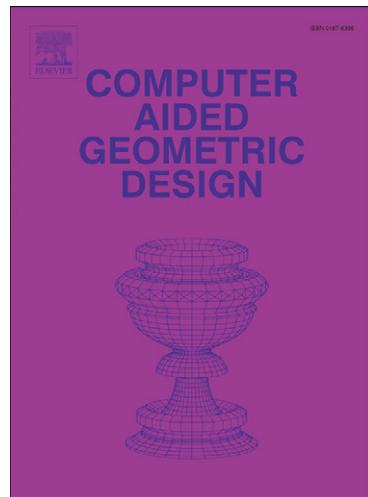


Accepted Manuscript

Regularization of B-spline objects

Guoliang Xu, Chandrajit Bajaj

PII: S0167-8396(10)00102-0
DOI: [10.1016/j.cagd.2010.09.008](https://doi.org/10.1016/j.cagd.2010.09.008)
Reference: COMAID 1245



To appear in: *Computer Aided Geometric Design*

Received date: 1 August 2008
Revised date: 16 March 2010
Accepted date: 27 September 2010

Please cite this article in press as: Xu, G., Bajaj, C. Regularization of B-spline objects. Computer Aided Geometric Design (2010), doi:10.1016/j.cagd.2010.09.008

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Regularization of B-Spline Objects *

Guoliang Xu ¹⁾ Chandrajit Bajaj ²⁾

¹⁾ State Key Laboratory of Scientific and Engineering Computing
Institute of Computational Mathematics, Academy of Mathematics and
System Sciences, Chinese Academy of Sciences, Beijing 100190, China

²⁾ Department of Computer Science and Institute of Computational
Engineering & Sciences, University of Texas at Austin, Austin TX 78712

Abstract

By a d-dimensional B-spline object (denoted as \mathcal{O}^d), we mean a B-spline curve ($d = 1$), a B-spline surface ($d = 2$) or a B-spline volume ($d = 3$). By regularization of a B-spline object \mathcal{O}^d we mean the process of relocating the control points of \mathcal{O}^d such that it approximates an isometric map of its definition domain in certain directions and is shape preserving. In this paper we develop an efficient regularization method for \mathcal{O}^d , $d = 1, 2, 3$ based on solving weak form L^2 -gradient flows constructed from the minimization of certain regularizing energy functionals. These flows are integrated via the finite element method using B-spline basis functions. Our experimental results demonstrate that our new regularization methods are very effective.

Key words: Spline objects, Regularization, L^2 -gradient flow, Finite element method.

MR (2000) Classification: 65D17

1 Introduction

In this paper we use the term a *d-dimensional B-spline object* \mathcal{O}^d to imply a B-spline curve (\mathcal{O}^1), a B-spline surface (\mathcal{O}^2) or a B-spline volume (\mathcal{O}^3). Since B-spline basis functions are linearly independent, many people believe that the parametric B-spline representation of a B-spline object is unique. However, this is not true. Two different sets of control points, with the same number of entries, may represent the same \mathcal{O}^d . To see this explicitly, let us consider a Bézier curve $\mathbf{x}(t) = \sum_{i=0}^n \mathbf{p}_i B_i^n(t) \in \mathbb{R}^2$, which is a special case of a B-spline curve, on the interval $[0, 1]$. Let $t = t(s) = \alpha s(1 - s) + s^2$, $s \in [0, 1]$, $\alpha \in [0, 2]$. Then $t(0) = 0$, $t(1) = 1$ with $t(s)$ increasing. Substituting $t(s)$ into $\mathbf{x}(t)$, we obtain another representation $\mathbf{y}(s) := \mathbf{x}(t(s))$ for the same curve. In this new representation, there is a free parameter $\alpha \in [0, 2]$, which shows

*G. Xu was supported in part by NSFC grant 60773165 and NSFC key project under the grant 10990013. C. Bajaj was supported in part by NSF grant CNS-0540033 and NIH contracts R01-EB00487, R01GM074258, R01-GM07308.

that the representation is not unique. Though the changing of α does not change the shape of the curve \mathcal{O}^1 , whenever $y(s)$ is sampled on a uniform grid of the interval $[0, 1]$, different α 's yield very different distributions of points on the curve. For instance, for the plane Bézier curve

$$\mathbf{x}(t) = [0, 1]^T B_1^3(t) + [2, 1]^T B_2^3(t) + [2, 0]^T B_3^3(t), \quad t \in [0, 1], \quad (1.1)$$

it is easy to obtain

$$\begin{aligned} \mathbf{y}(s) &= \frac{1}{2} \begin{bmatrix} 0 \\ \alpha \end{bmatrix} B_1^6(s) + \frac{1}{5} \begin{bmatrix} 2\alpha^2 \\ -\alpha^2 + 4\alpha + 1 \end{bmatrix} B_2^6(s) + \frac{1}{20} \begin{bmatrix} 6\alpha^4 - 22\alpha^3 + 24\alpha^2 + 12\alpha \\ 3\alpha^4 - 9\alpha^3 + 12\alpha + 12 \end{bmatrix} B_3^6(s) \\ &+ \frac{1}{5} \begin{bmatrix} -2\alpha^2 + 8\alpha + 2 \\ -\alpha^2 + 5 \end{bmatrix} B_4^6(s) + \frac{1}{2} \begin{bmatrix} -2\alpha^2 + 8\alpha + 2 \\ -\alpha^2 + 5 \end{bmatrix} B_5^6(s) + \begin{bmatrix} 4 \\ -\alpha + 2 \end{bmatrix} B_5^6(s) + \begin{bmatrix} 2 \\ 0 \end{bmatrix} B_6^6(s). \end{aligned}$$

Fig 1.1 shows the distributions of the sampling points and control polygons for three different α 's, where the sub-figures (a), (b) and (c) are generated using α values 0, 1 and 2, respectively. It is easy to observe that different α 's yield different distributions of the sampling points and the uniformness of the control polygon reflects the uniformness of the sampled points.

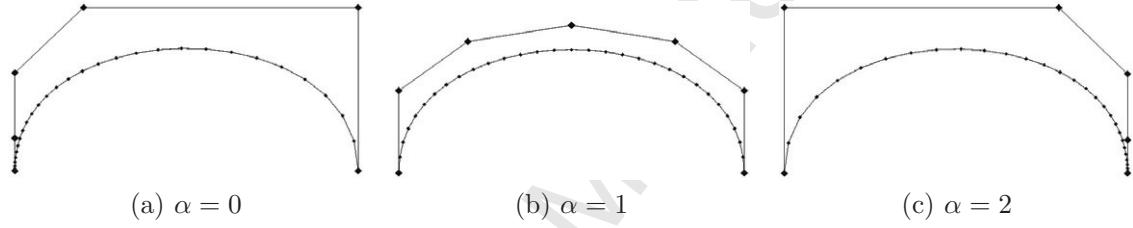


Fig 1.1: (a), (b) and (c) are the sampling points and control polygons of the curve $\mathbf{y}(s) = \mathbf{x}(t(s))$ defined by (1.1), using α values 0, 1 and 2, respectively. The curve points are sampled at the equal-spaced parameter values on $[0, 1]$.

The aim of “regularization of a \mathcal{O}^d ” is to relocate the control points of \mathcal{O}^d such that the object shape is not changed and a uniform sampling on the parameter domain leads to \mathcal{O}^d with an approximate uniform sampling. The main idea of the regularization is to construct several L^2 -gradient flows that move the control points of \mathcal{O}^d in the tangential directions. The foundation of our regularization method is based on a result of Epstein and Gage (see [6]), which states that the tangential motion of a manifold does not change its shape.

The regularization of \mathcal{O}^d is an interesting problem that has not been directly studied in the past. A direct application of the regularization is to generate a high quality discrete approximation (quality mesh) of \mathcal{O}^d . Another application perhaps is in the area of computer aided manufacturing (CAM). Quite often, the machine parts to be manufactured are represented by B-splines in a CAM system. When a machine part is manufactured using cutting tools, one often requires the surface motion of the tools or the curve motion relative the tools, be at a uniform rate with respect to the domain parameters.

For the triangular mesh fairing, the regularization problem has been considered in the past in computer graphics literature (see [11, 12, 17, 19]). In these papers, a tangential displacement is imposed at each of the vertices to move it towards its geometric or mass center of the surrounding

vertices. In Du et al's publications [4, 5], *Spherical Centroidal Voronoi Tessellation* (SCVT) have been introduced. A Voronoi tessellation $\{V_i\}_{i=1}^N$ of the sphere S from the generators $\{\mathbf{p}_i\}_{i=1}^N \subset S$ is called SCVT if and only if \mathbf{p}_i is the constrained mass centroid of the Voronoi region V_i for $i = 1, \dots, N$. Recently, a superlinear convergent Centroidal Voronoi Tessellation method for volumes has also been presented by Liu, Wang et al. [8]. In [18], a sequence of spherical triangulations are constructed which are optimal in certain sense, leading to smaller truncation error of the discrete Laplace-Beltrami operator.

Another subarea of research that is related to but different from the mesh regularization is mesh re-parametrization, which computes a bijective mapping between a triangular surface mesh and another surface with the same or similar topology. There are several excellent surveys [7, 9, 16] on this topic. For volume data re-parametrization, Martin et al [10] present a method based on discrete volumetric harmonic functions to parameterize a volumetric model in a way that it can be used to fit a single trivariate B-spline to data. We could not find any other prior work that directly addressed the regularization of parametric Bernstein-Bézier or B-spline surfaces.

The rest of our paper is organized as follows: Section 2 introduces some background material on \mathcal{O}^d . Sections 3 discuss the regularization methods of \mathcal{O}^d , $d = 1, 2, 3$, respectively. Several experimental examples to illustrate the dramatic effects of \mathcal{O}^d regularization are given in section 4.

2 B-Spline Objects

B-spline representations of curves and surfaces are well-known (see [1, 2, 3, 13, 15]). For easy of understanding and sake of completeness in this paper, we briefly introduce their definition and establish notation, necessary for subsequent use. There are several equivalent approaches to define B-spline functions, including divided differences of truncated power function (see [2, 15]), the blossoming method (see [13]) and the recursive formulas (see [1, 3]). We adopt the approach of recursive formulas which additionally facilitates computer programming.

Definition 2.1 Given a positive integer m , nonnegative integer k and a knot sequence

$$u_0 \leq \dots \leq u_i \leq u_{i+1} \leq u_{i+2} \leq \dots \leq u_{m+2k}.$$

$U = \{u_0, \dots, u_{m+2k}\}$ is referred to as a knot vector. Then B-splines basis functions are defined as follows

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & \text{for } u \in [u_i, u_{i+1}), \\ 0, & \text{otherwise,} \end{cases} & i = 0, 1, \dots, m + 2k - 1, \\ N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u), & i = 0, 1, \dots, m + k - 1, \\ \text{Assume } \frac{0}{0} = 0, & \end{cases} \quad (2.1)$$

where i is the index of $N_{i,k}(u)$, k is the degree.

In this paper, we assume that

$$u_0 = u_1 = \cdots = u_k = 0, \quad u_k < u_{k+1} < \cdots < u_{m+k}, \quad u_{m+k} = \cdots = u_{m+2k} = 1.$$

In order to have the B-spline objects to be at least C^1 smooth, we take $k \geq 2$.

B-Spline objects. Our B-spline objects are located in the n -dimensional Euclidean space \mathbb{R}^n and defined in the d -dimensional Euclidean space \mathbb{R}^d with $n \geq d$. For given positive integers m_1, m_2, \dots, m_d , a nonnegative integer k , and knot vectors

$$U^{(l)} = \{u_0^{(l)}, \dots, u_{m_l+2k}^{(l)}\}, \quad l = 1, \dots, d,$$

the 2d-sided degree k spline object \mathcal{O}^d is defined as

$$\mathcal{O}^d = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \mathbf{x}(\mathbf{u}), \mathbf{u} \in [0, 1]^d\},$$

with

$$\mathbf{x}(\mathbf{u}) = \sum_{i_1=0}^{m_1+k-1} \cdots \sum_{i_d=0}^{m_d+k-1} \mathbf{p}_{i_1 \dots i_d} \prod_{j=1}^d N_{i_j, k}(u^{(j)}),$$

where

$$\mathbf{p}_{i_1 \dots i_d} \in \mathbb{R}^n, \quad \mathbf{u} = [u^{(1)}, \dots, u^{(d)}]^T \in [0, 1]^d \subset \mathbb{R}^d.$$

The vectors $\mathbf{p}_{i_1 \dots i_d}$ are called control points of the object $\mathbf{x}(\mathbf{u})$. If $i_l = 0$ or m_l+k-1 , $l = 1, \dots, d$, $\mathbf{p}_{i_1 \dots i_d}$ are called boundary control points. Other control points are called inner control points. Let $D_l = \frac{\partial}{\partial u^{(l)}}$ be the first order partial derivative operators, $D_{ij} = D_i D_j$ the second order partial derivative operators.

For a space curve $C(t) \in \mathbb{R}^3$, $t \in [0, 1]$, the arc-length of the curve is

$$L = \int_0^1 \|C'(t)\| dt.$$

If $\|C'(t)\|$ is a constant, then $\|C'(t)\| = L$. Hence, we introduce the following definition.

Definition 2.2 Let $\mathbf{u}_l = [u^{(1)}, \dots, u^{(l-1)}, u^{(l+1)}, \dots, u^{(d)}]^T$ be a point in $[0, 1]^{d-1}$, and $L^{(l)}(\mathbf{u}_l)$ the arc length of the space curve $C(t) := \mathbf{x}(\mathbf{u})$, $t \in [0, 1]$, with $\mathbf{u} = [u^{(1)}, \dots, u^{(l-1)}, t, u^{(l+1)}, \dots, u^{(d)}]^T$. If $\|C'(t)\| = L^{(l)}(\mathbf{u}_l)$ for any $t \in [0, 1]$ and any $\mathbf{u}_l \in [0, 1]^{d-1}$, then we say the object \mathcal{O}^d is isometric in the $u^{(l)}$ direction.

In this paper, our attention is focused on the cases $n = 3$, $d = 1, 2, 3$, where the B-spline objects \mathcal{O}^d are space curves \mathcal{O}^1 , surfaces \mathcal{O}^2 and volumes \mathcal{O}^3 in \mathbb{R}^3 , respectively.

3 Regularization of B-Spline Objects Using L^2 -Gradient Flows

In this section, we present a set of L^2 -gradient flows, a method for discretization, and a solution via a linear system solver. The full derivation of these flows starting from an energy functional is given in Appendix A.

3.1 L^2 -Gradient Flows

For a given B-spline object \mathcal{O}^d defined by $\mathbf{x}(\mathbf{u})$, let

$$L^{(l)}(\mathbf{u}_l) = \int_0^1 \|D_l \mathbf{x}(\mathbf{u})\| d u^{(l)}, \quad \mathbf{u} \in [0, 1]^d, \quad l = 1, \dots, d.$$

For a fixed $\mathbf{u}_l \in [0, 1]^{d-1}$, $L^{(l)}(\mathbf{u}_l)$ is the arc length of the space curve $\mathbf{x}(\mathbf{u})$, $u^{(l)} \in [0, 1]$, in the $u^{(l)}$ direction. Let

$$\mathcal{E}^{(l)}(\mathcal{O}^d) = \int_{\mathcal{O}^d} \left[\|D_l \mathbf{x}(\mathbf{u})\| - L^{(l)}(\mathbf{u}_l) \right]^2 dV, \quad l = 1, \dots, d. \quad (3.1)$$

To regularize the B-spline object $\mathbf{x}(\mathbf{u})$, we minimize the energies $\mathcal{E}^{(l)}(\mathcal{O}^d)$ by moving the object point in the $D_l \mathbf{x}$ directions, respectively. This goal is achieved by solving d L^2 -gradient flows derived from (3.1). Consider the general form energy functional

$$\mathcal{E}(\mathcal{O}^d) = \int_{\mathcal{O}^d} f(D_1 \mathbf{x}, \dots, D_d \mathbf{x}) dV, \quad (3.2)$$

where f is a given C^1 smooth function with respect to its arguments.

Theorem 3.1 *Let $\mathcal{E}(\mathcal{O}^d)$ be an energy functional defined by (3.2) for the B-spline object \mathcal{O}^d . Then the L^2 -gradient flow of $\mathcal{E}(\mathcal{O}^d)$ in the direction $D_l \mathbf{x}$ is given by*

$$\int_{\mathcal{O}^d} \frac{\partial \mathbf{x}}{\partial t} \phi dV = - \int_{\mathcal{O}^d} \sum_{k=1}^d \left[\left((D_{kl} \mathbf{x})(\alpha_k^d)^T + (D_{kl} \mathbf{x})^T \alpha_k^d \mathbf{I}_n \right) (D_l \mathbf{x}) \phi + (D_l \mathbf{x})^T \alpha_k^d \mathbf{I}_n (D_l \mathbf{x}) D_k \phi \right] dV, \quad (3.3)$$

where \mathbf{I}_n stands for the $n \times n$ unit matrix, for $d = 1, 2, 3$, α_k^d are defined by

$$\begin{aligned} \alpha_1^1 &= \nabla_{D_1 \mathbf{x}} f + \frac{D_1 \mathbf{x}}{\|D_1 \mathbf{x}\|}, \\ \alpha_1^2 &= \nabla_{D_1 \mathbf{x}} f + \frac{1}{\sqrt{g}} [g_{22}(D_1 \mathbf{x} - g_{12} D_2 \mathbf{x})], \quad \alpha_2^2 = \nabla_{D_2 \mathbf{x}} f + \frac{1}{\sqrt{g}} [g_{11}(D_2 \mathbf{x} - g_{12} D_1 \mathbf{x})]. \\ \alpha_1^3 &= \nabla_{D_1 \mathbf{x}} f + D_2 \mathbf{x} \times D_3 \mathbf{x}, \quad \alpha_2^3 = \nabla_{D_2 \mathbf{x}} f + D_3 \mathbf{x} \times D_1 \mathbf{x}, \quad \alpha_3^3 = \nabla_{D_3 \mathbf{x}} f + D_1 \mathbf{x} \times D_2 \mathbf{x}. \end{aligned}$$

Taking $l = 1, \dots, d$ in (3.3), we obtain a sequence of flows. For $n = 3$ and $d = 1, 2, 3$, these flows are for B-spline curves, surfaces and volumes, respectively.

3.2 Numerical Solutions

To regularize the B-spline object \mathcal{O}^d in the $D_l \mathbf{x}$ directions, we solve (3.3) interchangeably using finite element method for the spatial discretization and a semi-implicit Euler scheme for temporal discretization. More specifically, the terms

$$(D_{kl}\mathbf{x})(\alpha_k^d)^T + (D_{kl}\mathbf{x})^T \alpha_k^d \mathbf{I}_n, \quad (D_l \mathbf{x})^T \alpha_k^d \mathbf{I}_n, \quad (3.4)$$

in (3.3) are treated as known quantities which are computed from the previous step of the approximation. The term $D_l \mathbf{x}$ is treated as an unknown.

For ease of description, we reorder the control points $\mathbf{p}_{i_1 \dots i_d}$ of the B-spline object into a 1-dimensional array and represent them as:

$$\mathbf{x}_0, \dots, \mathbf{x}_{n_0}, \mathbf{x}_{n_0+1}, \dots, \mathbf{x}_{n_1},$$

where $\mathbf{x}_0, \dots, \mathbf{x}_{n_0}$ are inner control points, $\mathbf{x}_{n_0+1}, \dots, \mathbf{x}_{n_1}$ are boundary control points, and

$$n_0 = \prod_{i=1}^d (m_i + k - 2) - 1, \quad n_1 = \prod_{i=1}^d (m_i + k) - 1.$$

The B-spline basis functions $\prod_{j=1}^d N_{i_j, k}(u^{(j)})$ are correspondingly reordered and represented as

$$\phi_0, \dots, \phi_{n_0}, \phi_{n_0+1}, \dots, \phi_{n_1}.$$

Using this ordering of the basis functions and control points, \mathcal{O}^d can be represented as

$$\mathbf{x}(\mathbf{u}) = \sum_{j=0}^{n_0} \mathbf{x}_j \phi_j(\mathbf{u}) + \sum_{j=n_0+1}^{n_1} \mathbf{x}_j \phi_j(\mathbf{u}). \quad (3.5)$$

Substituting (3.5) into (3.3), and taking the test function ϕ as ϕ_i , for $i = 0, \dots, n_0$, we can discretize (3.3) as a set of linear systems of ordinary differential equations (ODE) with the inner control points \mathbf{x}_i , $i = 0, \dots, n_0$, as unknowns.

$$\sum_{j=0}^{n_0} m_{ij}^{(l)} \frac{d\mathbf{x}_j(t)}{dt} + \sum_{j=0}^{n_0} q_{ij}^{(l)} \mathbf{x}_j = \sum_{j=n_0+1}^{n_1} q_{ij}^{(l)} \mathbf{x}_j, \quad i = 0, \dots, n_0, \quad l = 1, \dots, d,$$

where

$$\begin{aligned} m_{ij}^{(l)} &= \int_{\mathcal{O}^d} \phi_i \phi_j \mathbf{I}_n \, dV, \\ q_{ij}^{(l)} &= \int_{\mathcal{O}^d} \sum_{k=1}^d \left[\left((D_{kl}\mathbf{x})(\alpha_k^d)^T + (D_{kl}\mathbf{x})^T \alpha_k^d \mathbf{I}_n \right) (D_l \phi_j) \phi_i + (D_l \mathbf{x})^T \alpha_k^d \mathbf{I}_n (D_l \phi_j) D_k \phi_i \right] \, dV. \end{aligned}$$

For the temporal discretization of these ODE systems, we use a forward Euler scheme which finally leads to a set of linear systems of algebraic equations.

$$\sum_{j=0}^{n_0} \left(m_{ij}^{(l)} + \tau q_{ij}^{(l)} \right) \mathbf{x}_j^{(s)} = \sum_{j=0}^{n_0} m_{ij}^{(l)} \mathbf{x}_j^{(s-1)} + \sum_{j=n_0+1}^{n_1} \tau q_{ij}^{(l)} \mathbf{x}_j^{(0)}, \quad i = 0, \dots, n_0, \quad l = 1, \dots, d, \quad (3.6)$$

where τ is a temporal step-size. Let $S_i \subset \mathbb{R}^d$ be the support of ϕ_i . Then

$$m_{ij}^{(l)} = q_{ij}^{(l)} = \mathbf{0} \quad \text{if} \quad S_i \cap S_j = \emptyset.$$

Hence, the coefficient matrices of system (3.6) are sparse. Solving these linear systems via GMRES iterative method (see [14]) for $l = 1, \dots, d$, we obtained the new inner control points. In the GMRES iteration, we impose a limitation on the number of iterations, if the iteration does not converge within the limitation (we take it to be 300), we reduce the temporal step-size, and restart the iteration.

4 Algorithm Steps and Illustrative Examples

This section outline the steps of regularization algorithms for B-spline curves, surfaces and volumes. Illustrative examples are also given.

4.1 Curve Regularization

Let \mathcal{O}^1 be the B-spline curve to be regularized and defined by $\mathbf{x}(u) = \sum_{i_1=0}^{m_1+k-1} \mathbf{p}_{i_1} N_{i_1,k}(u)$.

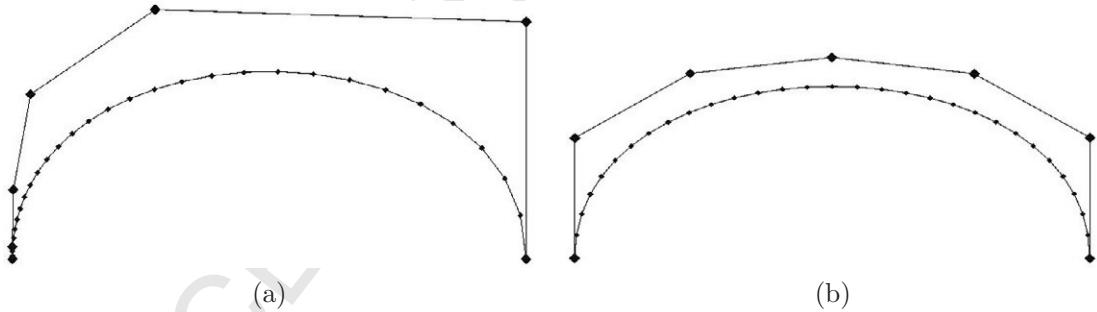


Fig 4.1: (a) and (b) are control polygons as well as the sampled curves of a spline curve before and after regularization. The larger and smaller diamonds represent the control points and sampled curve points, respectively.

Algorithm 4.1. Curve Regularization

1. Set the temporal step-size τ , and set $s = 0$, $\mathbf{p}_{i_1}^{(0)} = \mathbf{p}_{i_1}$.
2. For $n = 3$ and $d = 1$, form the linear system (3.6) and solve it.

3. Check the stopping conditions if $s \geq 1$

$$\begin{aligned} \max_{i_1} \|\mathbf{p}_{i_1}^{(s)} - \mathbf{p}_{i_1}^{(s-1)}\| &\leq \varepsilon_1 \tau, \\ \text{dis}((\mathcal{O}^1)^{(s)}, (\mathcal{O}^1)^{(0)}) &\geq \varepsilon_2, \end{aligned}$$

where $\text{dis}(\cdot, \cdot)$ stands for the Hausdorff distance between two curves. If one of the two conditions is satisfied, stop the iteration and return $\mathbf{x}^{(s)}(u) = \sum_{i_1=0}^{m_1+k-1} \mathbf{p}_{i_1}^{(s)} N_{i_1,k}(u)$ as the final result, otherwise set s to be $s + 1$, go back to the previous step.

It is well-known that the Hausdorff distance for two sets S and S' is defined as

$$\text{dis}(S, S') = \max \{d(S, S'), d(S', S)\},$$

where

$$d(S, S') = \max_{p \in S} d(p, S') \quad \text{with} \quad d(p, S') = \min_{p' \in S'} \|p - p'\|.$$

Let $\mathbf{x}^{(s)}(u)$ and $\mathbf{x}^{(0)}(u) = \mathbf{x}(u)$ be the B-spline representations of $(\mathcal{O}^1)^{(s)}$ and $(\mathcal{O}^1)^{(0)}$, respectively. Then the Hausdorff distance between $(\mathcal{O}^1)^{(s)}$ and $(\mathcal{O}^1)^{(0)}$ is computed approximately as follows.

Algorithm 4.2. Hausdorff distance Computation

1. Sample $\mathbf{x}^{(s)}(u)$ at equal-spaced parameter values $u_i = \frac{i}{K}$, $i = 1, 2, \dots, K-1$, obtain a set of points $\{\mathbf{x}_i^{(s)}\}$, where K is a given integer (we take it as $4m_1$).
2. For each $\mathbf{x}_i^{(s)}$, let $\mathbf{P}_i = \{\mathbf{x} \in \mathbb{R}^3 : (\mathbf{x} - \mathbf{x}_i^{(s)})^T D_1 \mathbf{x}^{(s)}(u_i) = 0\}$ be the plane passing through the point $\mathbf{x}_i^{(s)}$ and perpendicular to the tangent vector $D_1 \mathbf{x}^{(s)}(u_i)$. Compute the intersection point $\mathbf{y}_i^{(s)} = \mathbf{x}^{(0)}(u_i^{(s)})$ between the curve $\mathbf{x}^{(0)}(u)$ and the plane \mathbf{P}_i . We solve the intersection equation

$$(\mathbf{x}^{(0)}(u) - \mathbf{x}_i^{(s)})^T D_1 \mathbf{x}^{(s)}(u_i) = 0,$$

for the unknown u , using the Newton-Raphson iteration method with previous data $u_i^{(s-1)}$ as initial value ($u_i^{(0)}$ is taken as u_i). Then $d((\mathcal{O}^1)^{(0)}, (\mathcal{O}^1)^{(s)})$ is approximated as

$$d((\mathcal{O}^1)^{(0)}, (\mathcal{O}^1)^{(s)}) \approx \max_i \|\mathbf{x}_i^{(s)} - \mathbf{y}_i^{(s)}\|.$$

3. Compute $d((\mathcal{O}^1)^{(s)}, (\mathcal{O}^1)^{(0)})$ in the same way as computing $d((\mathcal{O}^1)^{(0)}, (\mathcal{O}^1)^{(s)})$. Finally compute $\text{dis}((\mathcal{O}^1)^{(s)}, (\mathcal{O}^1)^{(0)})$.

Note that if τ is small, $u_i^{(s-1)}$ and $u_i^{(s)}$ are close. Hence, $u_i^{(s-1)}$ is a good initial value for the Newton-Raphson iteration method for computing $u_i^{(s)}$. Two or three Newton-Raphson iterations are usually sufficient to obtain the required result.

Computational complexity. Since the B-spline basis has compact support, the coefficient matrix of system (3.6) is sparse with $O(m_1)$ nonzero elements. Hence, the computation costs of forming and solving system (3.6) are both of the order $O(m_1)$. Obviously, the cost for computing $\max_{i_1} \|\mathbf{p}_{i_1}^{(s)} - \mathbf{p}_{i_1}^{(s-1)}\|$ is $O(m_1)$. The cost for computing the Hausdorff distance is also $O(m_1)$. Hence, the total cost is of the order $O(m_1)$, for each iteration.

Figure 4.1 shows an regularization example of a B-spline curve, which is defined on the equal-spaced knots with $k = 3$, $m_1 = 6$.

4.2 Surface Regularization

Let \mathcal{O}^2 be a given B-spline surface to be regularized and defined by

$$\mathbf{x}(u, v) = \sum_{i_1=0}^{m_1+k-1} \sum_{i_2=0}^{m_2+k-1} \mathbf{p}_{i_1 i_2} N_{i_1, k}(u) N_{i_2, k}(v).$$

Algorithm 4.3. Surface Regularization

1. Regularize each of the four boundary curves using the curve regularization algorithm.
2. Regularize the interior of the surface as follows
 - (a) Set the temporal step-size τ , and set $s = 0$, $\mathbf{p}_{i_1 i_2}^{(0)} = \mathbf{p}_{i_1 i_2}$.
 - (b) For $n = 3$ and $d = 2$, form the systems (3.6) and solve them for $l = 1, 2$.
 - (c) Check the stopping conditions

$$\max \|\mathbf{p}_{i_1 i_2}^{(s)} - \mathbf{p}_{i_1 i_2}^{(s-1)}\| \leq \varepsilon_1 \tau, \quad (4.1)$$

$$\text{dis}((\mathcal{O}^2)^{(s)}, (\mathcal{O}^2)^{(0)}) \geq \varepsilon_2. \quad (4.2)$$

If one of the two conditions is satisfied, stop the iteration and return $\{\mathbf{p}_{i_1 i_2}^{(s)}\}$ as the final result of control points, otherwise set s to be $s + 1$, return to step (b).

Let $\mathbf{x}^{(s)}(u, v)$ and $\mathbf{x}^{(0)}(u, v) = \mathbf{x}(u, v)$ be the B-spline representations of $(\mathcal{O}^2)^{(s)}$ and $(\mathcal{O}^2)^{(0)}$, respectively. The computation of the Hausdorff distance between two surfaces is computed in a way similar to the curve case. The only difference is the computation of the intersection points. Let $\mathbf{x}_{ij}^{(s)}$ be a sampling point of $\mathbf{x}^{(s)}(u, v)$ at $[u_i, v_j]^T = [\frac{i}{K}, \frac{j}{K}]^T$. Let $\mathbf{n}_{ij}^{(s)}$ be the surface normal at $\mathbf{x}_{ij}^{(s)}$. Then the intersection point $\mathbf{y}_{ij}^{(s)} = \mathbf{x}^{(0)}(u_i^{(s)}, v_j^{(s)})$ is defined as the intersection of the line $\mathbf{x}_{ij}^{(s)} + t\mathbf{n}_{ij}^{(s)}$ and surface $\mathbf{x}^{(0)}(u, v)$. To solve the equation $\mathbf{x}_{ij}^{(s)} + t\mathbf{n}_{ij}^{(s)} = \mathbf{x}^{(0)}(u, v)$, for the

unknowns t , u and v , we take inner products of both sides with $D_1\mathbf{x}^{(s)}(u_i, v_j)$ and $D_2\mathbf{x}^{(s)}(u_i, v_j)$ and obtain the following equations for the unknown u, v :

$$\begin{cases} [\mathbf{x}_{ij}^{(s)} - \mathbf{x}^{(0)}(u, v)]^T D_1\mathbf{x}^{(s)}(u_i, v_j) = 0, \\ [\mathbf{x}_{ij}^{(s)} - \mathbf{x}^{(0)}(u, v)]^T D_2\mathbf{x}^{(s)}(u_i, v_j) = 0. \end{cases} \quad (4.3)$$

System (4.3) is solved using the Newton-Raphson method with $u_i^{(s-1)}$ and $v_j^{(s-1)}$ as initial values (we take $u_i^{(0)}$ and $v_j^{(0)}$ as u_i and v_j , respectively). The solutions of the system are the required $u_i^{(s)}$ and $v_j^{(s)}$.

Remark 4.1 *Though the tangential motion does not change the shape of the surface, but since the PDEs to be solved are nonlinear, the semi-implicit discretization yields a certain amount of normal motion. This motion is small relative to the tangential motion. However the accumulation over long time evolution could make it significant. The stopping condition (4.2) is used to control this movement within a given allowable bound.*

Remark 4.2 *If a B-spline surface representation is not unique, then there are degrees of freedom to relocate the control points to regularize the surface. If the B-spline surface representation is unique, then there will essentially be no freedom for the tangential motion. However we can still regularize the surface within a given error bound ε_2 .*

Computational complexity. Similar to the curve case, the computation costs of forming and solving system (3.6), and for computing $\max_{i_1} \|\mathbf{p}_{i_1}^{(s)} - \mathbf{p}_{i_1}^{(s-1)}\|$ are $O(m_1 m_2)$. The cost for computing the Hausdorff distance is also $O(m_1 m_2)$. Hence, the total cost is of the order $O(m_1 m_2)$.

Now we give a few examples to illustrate that the surface regularization method proposed is efficient and gives good results. Fig 4.2–4.4 show the regularization effects for B-spline surfaces with different shaped boundaries. The B-spline basis are defined on equal-spaced knots with $k = 3$ and $m_1 = m_2 = 6$. Sub-figures (a) and (b) show the control meshes of the spline surfaces on the domain $[0, 1]^2$ without and with using the regularization step, respectively. Sub-figures (c) and (d) show the uniform sampling of the spline surfaces on the domain $[0, 1]^2$ without and with using the regularization step, respectively. It is easy to observe that the surface shapes are unchanged after the regularization, but the quality of the sampling surface meshes are significantly improved.

4.3 Volume Regularization

Let \mathcal{O}^3 be a given B-spline volume to be regularized which is defined by

$$\mathbf{x}(u, v, w) = \sum_{i_1=0}^{m_1+k-1} \sum_{i_2=0}^{m_2+k-1} \sum_{i_3=0}^{m_3+k-1} \mathbf{p}_{i_1 i_2 i_3} N_{i_1, k}(u) N_{i_2, k}(v) N_{i_3, k}(w).$$

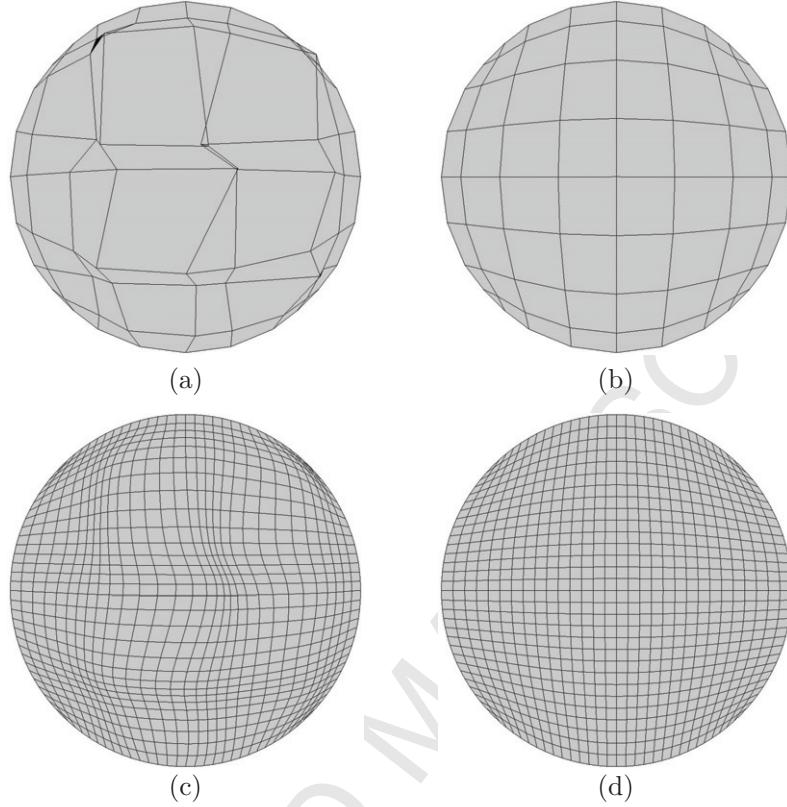


Fig 4.2: (a) Control mesh of a spline surface without regularization. (b) Control mesh of the same spline surface after regularization. (c) Spline surface mesh without regularization. (d) Spline surface mesh after regularization.

Algorithm 4.4. Volume Regularization

1. Regularize each of 12 boundary curves of the volume using the curve regularization algorithm.
2. Regularize each of 6 boundary faces of the volume using the surface regularization algorithm.
3. Regularize the interior of the volume as follows:
 - (a) Set the temporal step-size τ , and set $s = 0$, $\mathbf{p}_{i_1 i_2 i_3}^{(0)} = \mathbf{p}_{i_1 i_2 i_3}$.
 - (b) For $n = 3$ and $d = 3$, form the systems (3.6) and then solve them for $l = 1, 2, 3$.
 - (c) Check the stopping conditions if $s \geq 1$

$$\max \| \mathbf{p}_{i_1 i_2 i_3}^{(s)} - \mathbf{p}_{i_1 i_2 i_3}^{(s-1)} \| \leq \varepsilon_1 \tau, \quad (4.4)$$

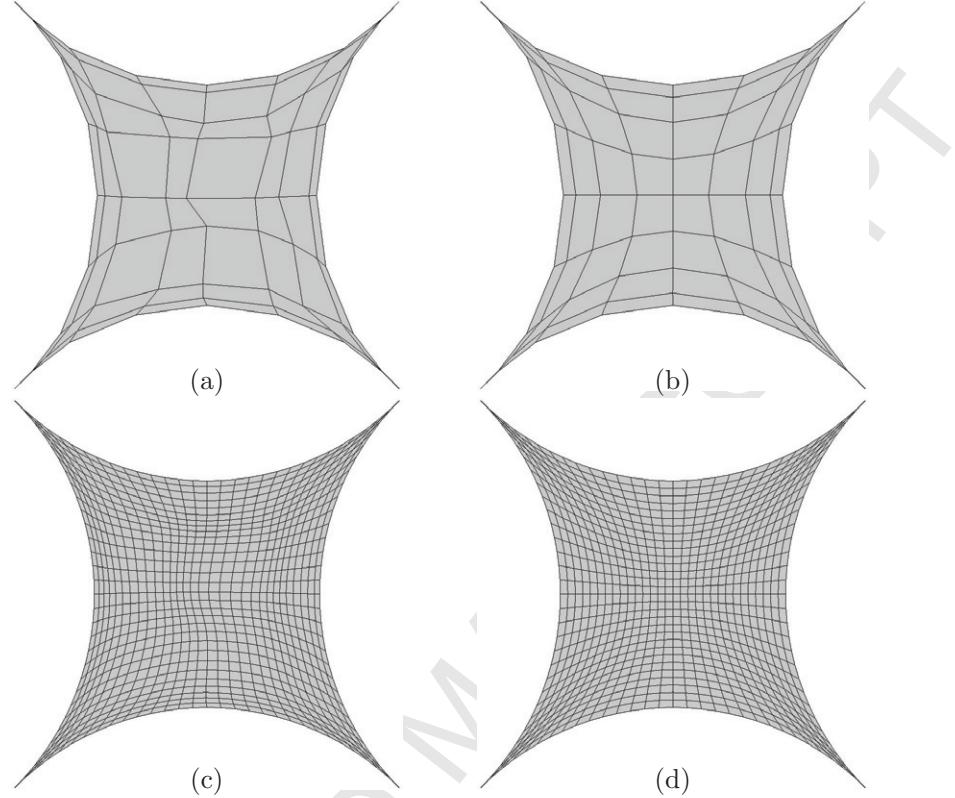


Fig 4.3: (a) Control mesh of a spline surface without regularization. (b) Control mesh of the same spline surface after regularization. (c) Spline surface mesh without regularization. (d) Spline surface mesh after regularization.

If this condition is satisfied, stop the iteration and return $\{\mathbf{p}_{i_1 i_2 i_3}^{(s)}\}$ as the final control point set, otherwise set s to be $s + 1$, return to step (b).

Remark 4.3 Since the space $\text{span}\{D_1 \mathbf{x}, D_2 \mathbf{x}, D_3 \mathbf{x}\}$ for the B-spline volume is three dimensional, the normal space is null. Hence there is no normal motion problem as mentioned in Remark 4.1. For this reason, we do not need to control normal motion. We further point out that the regularization for B-spline volume can always be performed. This is different from the cases of curves and surfaces.

Computational complexity. The computation costs of forming and solving the system (3.6), and for computing $\max_{i_1} \|\mathbf{p}_{i_1}^{(s)} - \mathbf{p}_{i_1}^{(s-1)}\|$ are all $O(m_1 m_2 m_3)$.

Next we present a few test examples to illustrate the regularization effect of B-spline volumes. To see the insides of the sampled volumes, we use both spacing and mesh slicing display techniques. Figure 4.5 shows the regularization of a cube, where the initial inner control points

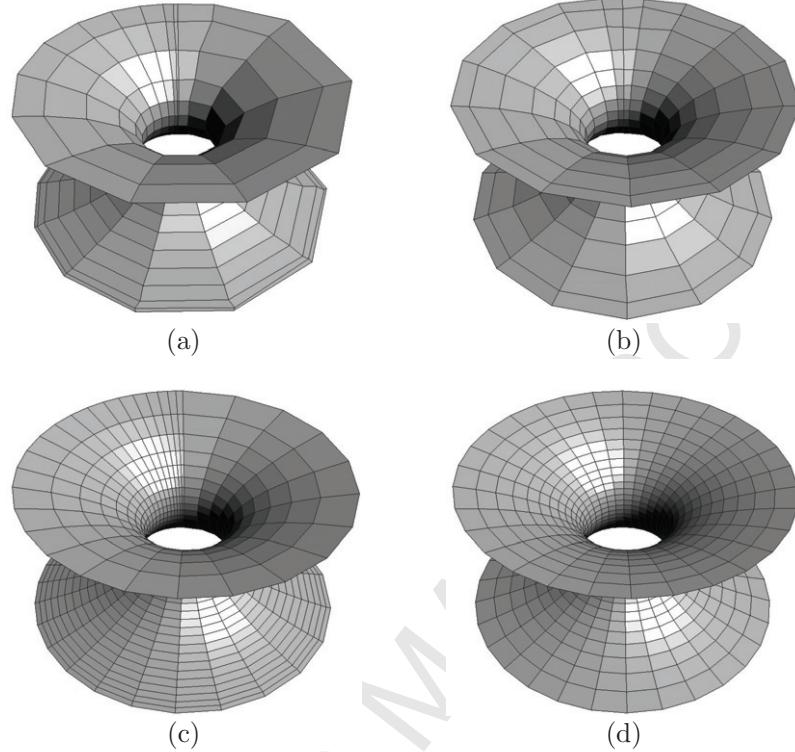


Fig 4.4: Convex B-spline surface: (a) Control mesh of a spline surface without regularization. (b) Control mesh of the same spline surface after regularization. (c) Spline surface mesh without regularization. (d) Spline surface mesh after regularization.

are not regularly distributed. The left figure shows these irregularities. The right figure is the result after regularization. Figure 4.6 show the regularizing results for a volume with curved boundary faces. These figures clearly exhibit the post regularization effects. In these examples, the B-spline volume are defined on equal-spaced knots with $m_1 = m_2 = m_3 = 6$ and $k = 3$. The sampling is on an equal-spaced grid of the domain $[0, 1]^3$ with sampling rate 16^3 .

5 Conclusions and Extensions

We have presented a novel and efficient regularization technique for multi-dimensional B-spline objects. Our new technique consists of first devising appropriate energy functionals, then constructing variational L^2 -gradient flows and finally solving the flows using the finite element method. Our regularization method can be used to generate uniform polygonal approximations for curves, near regular and quality quadrilateral surface meshes for surfaces and near regular and quality hexahedral volumetric meshes for volumes. The implementation results show that

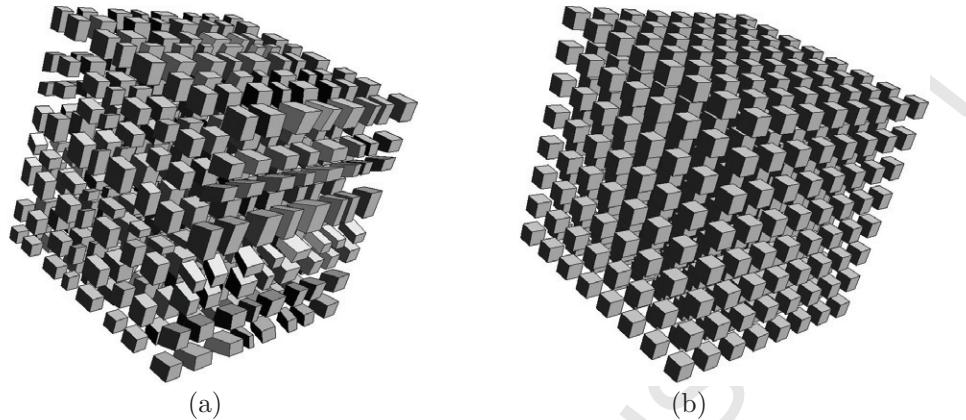


Fig 4.5: Regularization of a cube: (a) Sampling of the initial B-spline volume with irregularly distributed control points. (b) Sampling of the B-spline volume after regularization.

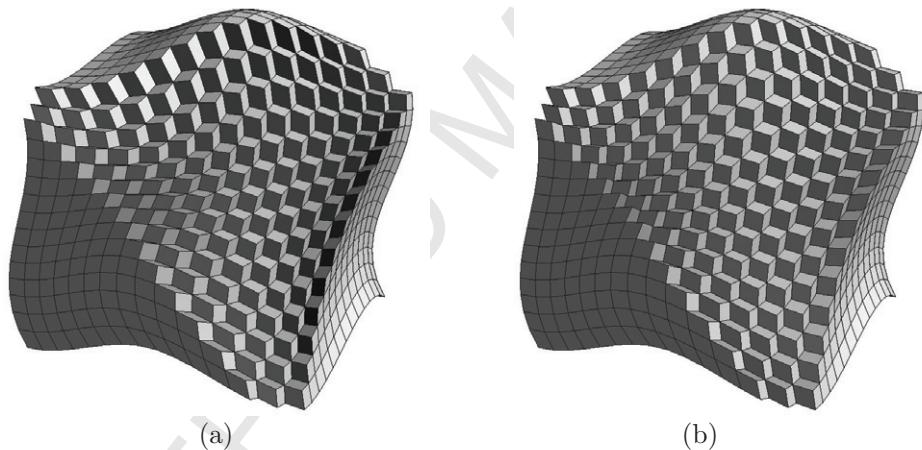


Fig 4.6: Regularizing a B-spline volume with curved boundary faces: (a) Sampling of the initial B-spline volume. (b) Sampling of the B-spline volume after regularization.

the proposed method is effective and yields desired results. This research could be extended in several directions. Here we list a few possibilities.

1. Application the regularizing method to generate regular curve, surface and volume meshes. This could be done by first interpolating the mesh with the B-spline objects, and then regularizing the B-spline objects and finally resampling them to obtain regularized meshes.
2. The regularization method for B-spline objects could be applied to other type objects,

such as Bézier objects, NURBS objects as well as subdivision objects (subdivision curves, surfaces and volumes) and so on.

3. Replacement of the current regularizing energy functional with other type of functionals to meet different requirement, such as the energy functional measuring the angle distribution of the tangent vectors aiming at to achieve uniform distribution of angles. Since the flows for surface and volume are derived for a general form functional, these extensions are straightforward.

Acknowledgment. Work in this paper was completed when Guoliang Xu was visiting Chandrajit Bajaj at UT-CVC. His visit was additionally supported by the J. T. Oden Faculty Fellowship Research Program at ICES.

References

- [1] M. G. Cox. The numerical evaluation of B-splines. *Jour. Inst. Math. Applic.*, 10:134–149, 1972.
- [2] H. B. Curry and I. J. Schoenberg. On spline distribution and their limits: the Pólya distribution functions. *Bull. Amer. Math. Soc.*, 53:109, 1947.
- [3] C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62, 1972.
- [4] Q. Du and M. D. Gunzburger L. Ju. Constrained centroidal Voronoi tessellations for surfaces. *SIAM J. Sci. Comput.*, 24(5):1488–1506, 2003.
- [5] Q. Du and M. D. Gunzburger L. Ju. Voronoi-based finite volume methods, optimal Voronoi meshes, and PDEs on the sphere. *Computer methods in applied mechanics and engineering*, 192:3933–3957, 2003.
- [6] C. L. Epstein and M. Gage. The curve shortening flow. In A. Chorin and A. Majda, editors, *Wave Motion: Theory, Modeling, and Computation*, pages 15–59. Springer-Verlag, New York, 1987.
- [7] B. Lévy. *Parameterization and deformation analysis on a manifold*. Technical Report, ALICE, <http://alice.loria.fr/publications>, 2007.
- [8] Y. Liu, W. Wang, B. Levy, F. Sun, D. Yan, L. Lu, and C. Yang. On Centroidal Voronoi Tessellation , Energy Smoothness and Fast Computation. *ACM Transactions on Graphics*, 28(4):1–17, 2008.
- [9] Floater M. and Hormann K. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometry Modelling, Mathematics and Visualization*, pages 157–186. Springer, 2005.
- [10] T. Martin, E. Cohen, and M. Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In *ACM Symposium on Solid and Physical Modeling*, pages 269–280, Stony Brook, New York, 2008. ACM.
- [11] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski. Polyhedral surface smoothing with simultaneous mesh regularization. In *Geometric Modeling and Processing Proceedings*, pages 229–237, 2000.
- [12] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski. Mesh regularization and adaptive smoothing. *Computer-Aided Design*, 33:789–800, 2001.

- [13] L. Ramshaw. Blossoming; a connect-the-dots approach to spline. Report 19, Digital, System Research Center, Palo Alto, CA, 1987.
- [14] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM Philadelphia, PA, 2 edition, 2003.
- [15] I. J. Schoenberg. Contributions to the problem of approximation of equidistance data by analytic functions. *Quart. Appl. Math.*, 4:45–99, 1946.
- [16] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and Trends in Computer and Vision*, 2(2):105–171, 2006.
- [17] Z. J. Wood, D. Breen, and M. Desbrun. Semi-regular mesh extraction from volumes. In IEEE Visualization Proceedings of the conference on Visualization '00, pages 275–282, Salt Lake City, Utah, United States, 2000.
- [18] G. Xu. Discrete Laplace-Beltrami Operator on Sphere and Optimal Spherical Triangulations. *International Journal of Computational Geometry & Applications*, 16(1):75–93, 2006.
- [19] G. Xu, Q. Pan, and C. Bajaj. Discrete surface modelling using partial differential equations. *Computer Aided Geometric Design*, 23(2):125–145, 2006.

A The Derivation of L^2 -gradient Flows

We construct an L^2 -gradient flow for a general form energy functional

$$\mathcal{E}(\mathcal{O}^d) = \int_{\mathcal{O}^d} f(D_1\mathbf{x}, \dots, D_d\mathbf{x}) \, dV = \int_0^1 \cdots \int_0^1 f(D_1\mathbf{x}, \dots, D_d\mathbf{x}) \, J^d \, d\mathbf{u}, \quad (\text{A.1})$$

where f is a given C^1 smooth function with respect to its arguments, J^d is the metric of the object. For $n = 3$ and $d = 1, 2, 3$,

$$\begin{aligned} J^1 &= \|D_1\mathbf{x}\|, \\ J^2 &= \sqrt{g}, \quad g = g_{11}g_{22} - g_{12}^2, \\ &\quad g_{11} = (D_1\mathbf{x})^T D_1\mathbf{x}, \quad g_{12} = (D_1\mathbf{x})^T D_2\mathbf{x}, \quad g_{22} = (D_2\mathbf{x})^T D_2\mathbf{x}, \\ J^3 &= \det[D_1\mathbf{x}, D_2\mathbf{x}, D_3\mathbf{x}], \end{aligned}$$

and $\det[\cdot]$ stands for the determinant of a matrix. Let

$$\underline{\mathcal{O}}^d(\Phi, \varepsilon)) = \{\underline{\mathbf{x}}(\mathbf{u}, \varepsilon) = \mathbf{x} + \varepsilon\Phi(\mathbf{u}) : \mathbf{u} \in [0, 1]^d\}, \quad \Phi \in C_0^1([0, 1]^d)^n.$$

Then we have

$$\delta(\mathcal{E}(\mathcal{O}^d), \Phi) = \frac{d}{d\varepsilon} \mathcal{E}(\underline{\mathcal{O}}^d(\cdot, \varepsilon)) \Big|_{\varepsilon=0},$$

where

$$\delta(\mathcal{E}(\mathcal{O}^d), \Phi) = \int_0^1 \cdots \int_0^1 \left[\sum_{k=1}^d (\nabla_{D_k \mathbf{x}} f)^T \delta(D_k \mathbf{x}) + \frac{f \delta(J^d)}{J^d} \right] J^d \, d\mathbf{u}. \quad (\text{A.2})$$

Now we compute $\delta(D_k \mathbf{x})$ and $\delta(J^d)$ in (A.2). It follows from

$$\underline{\mathbf{x}} = \mathbf{x} + \varepsilon \Phi, \quad (\text{A.3})$$

$$D_k \underline{\mathbf{x}} = D_k \mathbf{x} + \varepsilon D_k \Phi, \quad (\text{A.4})$$

we have

$$\delta(D_k \mathbf{x}) = D_k \Phi, \quad k = 1, \dots, d,$$

and

$$\delta(J^1) = \frac{(D_1 \Phi)^T D_1 \mathbf{x}}{\|D_1 \mathbf{x}\|},$$

$$\delta(J^2) = \frac{1}{\sqrt{g}} [g_{22}(D_1 \Phi)^T D_1 \mathbf{x} + g_{11}(D_2 \Phi)^T D_2 \mathbf{x} - g_{12}[(D_1 \Phi)^T D_2 \mathbf{x} + (D_2 \Phi)^T D_1 \mathbf{x}]].$$

$$\begin{aligned} \delta(J^3) &= \det[D_1 \Phi, D_2 \mathbf{x}, D_3 \mathbf{x}] + \det[D_1 \mathbf{x}, D_2 \Phi, D_3 \mathbf{x}] + \det[D_1 \mathbf{x}, D_2 \mathbf{x}, D_3 \Phi] \\ &= (D_1 \Phi)^T (D_2 \mathbf{x} \times D_3 \mathbf{x}) + (D_2 \Phi)^T (D_3 \mathbf{x} \times D_1 \mathbf{x}) + (D_3 \Phi)^T (D_1 \mathbf{x} \times D_2 \mathbf{x}), \end{aligned}$$

where \times denotes cross product of two vectors in \mathbb{R}^3 . Hence $\delta(J^d)$ could be written as

$$\delta(J^d) = \sum_{k=1}^d (D_k \Phi)^T \beta_k^d, \quad \beta_k^d \in \mathbb{R}^n.$$

Substituting these into (A.2), we have

$$\begin{aligned} \delta(\mathcal{E}(\mathcal{O}^d), \Phi) &= \int_{\mathcal{O}^d} \sum_{k=1}^d \left[(D_k \Phi)^T \left(\nabla_{D_k \mathbf{x}} f + \beta_k^d \right) \right] dV \\ &= \int_{\mathcal{O}^d} \sum_{k=1}^d \left((D_k \Phi)^T \alpha_k^d \right) dV, \end{aligned} \quad (\text{A.5})$$

where

$$\alpha_k^d = \nabla_{D_k \mathbf{x}} f + \beta_k^d \in \mathbb{R}^n,$$

To construct L^2 -gradient flows moving the volume in the direction $D_l \mathbf{x}$, $l = 1, \dots, d$, we take

$$\Phi = (D_l \mathbf{x})(D_l \mathbf{x})^T \phi, \quad \phi \in C_0^1([0, 1]^d).$$

Then

$$D_k \Phi = (D_{kl} \mathbf{x})(D_l \mathbf{x})^T \phi + (D_l \mathbf{x})(D_{kl} \mathbf{x})^T \phi + (D_l \mathbf{x})(D_l \mathbf{x})^T D_k \phi.$$

Therefore, we construct the following weak form L^2 -gradient flows moving the surface in the $D_l \mathbf{x}$ direction

$$\begin{aligned} \int_{\mathcal{O}^d} \frac{\partial \mathbf{x}}{\partial t} \phi \, dV &= - \int_{\mathcal{O}^d} \sum_{k=1}^d \left[\left((D_{kl} \mathbf{x})(\alpha_k^d)^T + (D_{kl} \mathbf{x})^T \alpha_k^d \mathbf{I}_n \right) (D_l \mathbf{x}) \phi + (D_l \mathbf{x})^T \alpha_k^d \mathbf{I}_n (D_l \mathbf{x}) D_k \phi \right] dV, \\ l &= 1, \dots, d. \end{aligned} \quad (\text{A.6})$$

For the energy functional defined by (3.1), $f = [\|D_l \mathbf{x}(\mathbf{u})\| - L^l(\mathbf{u}_l)]^2$, it is easy to derive that

$$\nabla_{D_l \mathbf{x}} f = 2 \left[1 - \frac{L^l(\mathbf{u}_l)}{\|D_l \mathbf{x}(\mathbf{u})\|} \right] D_l \mathbf{x}(\mathbf{u}), \quad \nabla_{D_k \mathbf{x}} f = \mathbf{0}, \quad \text{for } k \neq l.$$