A general framework for surface modeling using geometric partial differential equations

Guoliang Xu\textsuperscript{a,\#}, Qin Zhang\textsuperscript{b}

\textsuperscript{a}LSEC, Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China
\textsuperscript{b}Department of Mathematics, Beijing Information Science and Technology University, Beijing 100085, China

Received 8 April 2006; received in revised form 9 June 2007; accepted 11 June 2007

Abstract

In this paper, a general framework for surface modeling using geometric partial differential equations (PDEs) is presented. Starting with a general integral functional, we derive an Euler–Lagrange equation and then a geometric evolution equation (also known as geometric flow). This evolution equation is universal, containing several well-known geometric partial differential equations as its special cases, and is discretized under a uniform framework over surface meshes. The discretization of the equation involves approximations of curvatures and several geometric differential operators which are consistently discretized based on a quadratic fitting scheme. The proposed algorithm can be used to construct surfaces for geometric design as well as simulate the behaviors of various geometric PDEs. Comparative experiments show that the proposed approach can handle a large number of geometric PDEs and the numerical algorithm is efficient.

© 2007 Published by Elsevier B.V.

Keywords: Surface modeling; Geometric PDEs; Triangular surface mesh

1. Introduction

Using geometric partial differential equations (PDEs) in surface modeling and designing has been an interesting research topic (see Xu et al. (2006) for references). The frequently used geometric PDEs include mean curvature flow, averaged mean curvature flow, surface diffusion flow and Willmore flow etc. Some of these equations have solutions with certain properties, for example, the mean curvature flow leads to minimal surfaces and Willmore flow yields surfaces with minimal squared mean curvature. But other equations may not share these kinds of optimal properties.

In surface modeling or shape design, the target surfaces constructed are often required to possess certain optimal properties. In this paper, we present a general framework for surface modeling using geometric PDEs. Starting with a general integral functional, we then derive an Euler–Lagrange equation and a geometric PDE. The geometric PDE is universal, which contains several well-known geometric PDEs as its special cases, but can be discretized under a

\# Project supported in part by NSFC grant 10371130 and National Key Basic Research Project of China (2004CB318000).
\# Corresponding author.

E-mail address: xuguo@lsec.cc.ac.cn (G. Xu).

0167-8396/S – see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.cagd.2007.06.002

Please cite this article in press as: G. Xu, Q. Zhang, A general framework for surface modeling using geometric partial differential equations, Computer Aided Geometric Design (2007), doi:10.1016/j.cagd.2007.06.002
uniform framework over triangular surfaces. The discretization of the equation involves discrete approximations of curvatures and several differential operators, such as mean curvature, Gaussian curvature, Laplace–Beltrami operator etc. All these quantities are consistently discretized based on a quadratic fitting scheme.

Modeling problem. We are given a triangulated surface with some vertices labeled as inner and the others outer. The inner vertices are subject to change while the outer fixed. If an outer vertex is a one-ring neighbor of an inner vertex, we refer to it as a boundary vertex. The aim is moving the inner vertices to satisfy a specified geometric PDE.

Previous work on PDE approaches

1. Using biharmonic equation. Biharmonic equation and its variants on a rectangular domain have been used for solving the blending and hole filling problems (see Bloor et al.’s works at the end of 1980s (Bloor and Wilson, 1989a, 1989b, 1990)). The advantage of using biharmonic equation is that it is linear, and therefore easy to solve. Recently this equation and its generalizations have been used in interactive surface design (see Du and Qin (2005), Ugail et al. (1999)) and interactive sculpting (see Du and Qin (2000)). Lowe et al. (1990) presented a PDE method with certain functional constraints, such as geometric constraints, physical criteria and engineering requirements, which can be incorporated into the shape design process. However, the biharmonic equation is not geometry intrinsic and the solution of the equation (the geometry of the surface) depends on the concrete parametrization used. Furthermore, these methods are inappropriate for modeling surfaces with arbitrary shaped boundaries.

2. Using second order geometric equations. Mean curvature flow (MCF) and its variants have been intensively used for smoothing or fairing noisy surfaces. We just mention a few references among the large number of literatures. Desbrun et al. (1999) employed an implicit discretization of MCF to obtain a strongly stable numerical smoothing scheme. The same strategy of discretization is also adopted and analyzed by Deckelnick and Dziuk (2002) with the conclusion that this scheme is unconditionally stable. Clarenz et al. (2000) introduced an anisotropic geometric diffusion to enhance features as well as smoothing. Ohtake et al. (2000) combined an inner fairness mechanism in their fairing process to increase the mesh regularity. Bajaj and Xu (2003) smoothed both surfaces and functions on surfaces in a C^2 smooth function space defined by the limit of triangular subdivision surfaces (quartic box splines) with sharp features enhanced.

3. Using higher order geometric equations. MCF has been shown to be the most important and efficient flow for fairing or denoising surface meshes. However, for solving the surface modeling and designing problems, MCF cannot achieve G^1 smooth joining of different patches. Recently, fourth order flows have been used to solve the problems of surface blending and free-form surface design. In Schneider and Kobbelt (2000) fair meshes with G^1 boundary conditions are created using surface diffusion flow in a special case where the meshes are assumed to have subdivision connectivity. The later paper (Schneider and Kobbelt, 2001) employed the same equation for smoothing meshes while satisfying G^1 boundary conditions. Outer fairness (the smoothness in the classical sense) and inner fairness (the regularity of the vertex distribution) criteria are used in their fairing process. A finite element method is used in Clarenz et al. (2004) to solve the equation of Willmore flow for surface restoration. Willmore flow is also applied to smooth triangular mesh in Yoshizawa and Belyaev (2002). In Xu et al. (2006) and Xu and Pan (2006) several geometric flows, including the second, fourth and sixth order flows have been employed for surface blending, N-sided hole filling and free-form surface design.

Main contributions of this paper. Traditionally different geometric PDEs are treated separately. In this research, we propose a uniform framework to handle a class of geometric PDEs for solving the surface modeling problems. We derive these PDEs by a complete-variation from a general integral functional. The resulted PDEs, which have some desirable features, are solved using a divided-difference-like method. This divided-difference-like method requires approximations of the involved geometric differential operators. Previous work on the discrete approximations of these geometric differential operators are constructed based on several different theorems from differential geometry. Therefore, they are not consistent in the sense that they may not come from an identical surface. We employ a consistent estimation of these geometric differential operators.

The rest of this paper is organized as follows. Section 2 introduces some basic materials on differential geometry. Section 3 derives the nonlinear geometric PDE used in this paper. Numerical solving of the proposed PDE is discussed in Section 4, where Section 4.1 gives the discretization scheme for the involved differential operators, Section 4.2 outlines the algorithms as a whole for the surface modeling problems. Comparative examples to illustrate the different effects achievable from the solutions of several geometric PDEs are given in Section 5. Section 6 concludes the paper.
2. Notations and differential geometry preliminaries

In this section, we introduce the used notations, curvatures and several geometric differential operators. Some results used in this paper are also presented.

Let \( M = \{ (u, v) \in \mathbb{R}^2 \mid (u, v) \in \Omega \subset \mathbb{R}^2 \} \) be a regular sufficiently smooth parametric surface. To simplify the notation we sometimes write \( w = (u, v) = (u^1, u^2) \). Let \( g_{\alpha \beta} = \langle x_{u^\alpha}, x_{v^\beta} \rangle \) and \( b_{\alpha \beta} = \langle n, x_{u^\alpha} x_{v^\beta} \rangle \) be the coefficients of the first and the second fundamental forms of \( M \) with

\[
x_{u^\alpha} = \frac{\partial x}{\partial u^\alpha}, \quad x_{u^\alpha} x_{v^\beta} = \frac{\partial^2 x}{\partial u^\alpha \partial u^\beta}, \quad \alpha, \beta = 1, 2, \quad n = (x_u \times x_v)/\|x_u \times x_v\|,
\]

where \( \langle \cdot, \cdot \rangle \) and \( \times \) denote the usual scalar and cross products of two vectors, respectively, in Euclidean space \( \mathbb{R}^3 \). Set \( g = \det(g_{\alpha \beta}), \quad [g^{\alpha \beta}] = [g_{\alpha \beta}]^{-1}, \quad b = \det(b_{\alpha \beta}) \).

Curvatures. To introduce the mean curvature and Gaussian curvature, let us first introduce the concept of Weingarten map or shape operator. The shape operator of surface \( M \) is a self-adjoint linear map on the tangent space \( T_X M := \text{span}\{x_u, x_v\} \) defined by

\[
\mathcal{W} : T_X M \rightarrow T_X M,
\]

such that

\[
\mathcal{W}(x_u) = -n_u, \quad \mathcal{W}(x_v) = -n_v.
\]

We can represent this linear map by a \( 2 \times 2 \) matrix \( S = [b_{\alpha \beta}] \). In particular,

\[
[b_u, b_v] = -[x_u, x_v] S^T \tag{2.1}
\]

is valid. The eigenvalues \( k_1, k_2 \) of \( S \) are principal curvatures of \( M \) and their arithmetic average and product are the mean curvature and Gaussian curvature, respectively. That is

\[
H = \frac{k_1 + k_2}{2} = \frac{\text{tr}(S)}{2}, \quad K = k_1 k_2 = \det(S).
\]

Tangential gradient operator. Let \( f \) be a \( C^1 \) smooth function on \( M \). Then the tangential gradient operator \( \nabla \) acting on \( f \) is given by (see do Carmo (1976), p. 102)

\[
\nabla f = [x_u, x_v] [g^{\alpha \beta}] (f_u, f_v)^T \in \mathbb{R}^3, \tag{2.2}
\]

where \( f_u \) and \( f_v \) denote the first order partial derivatives of \( f \) with respect to arguments \( u \) and \( v \). The second order partial derivatives of \( f \) are denoted by \( f_{u u^\alpha} x_{v^\beta} \) below.

Second tangent operator. Let \( f \) be a \( C^1 \) smooth function on \( M \). Then we define the second tangent operator \( \diamond \) acting on \( f \) by

\[
\diamond f = [x_u, x_v] [h^{\alpha \beta}] (f_u, f_v)^T \in \mathbb{R}^3, \tag{2.3}
\]

where

\[
[h^{\alpha \beta}] := \frac{1}{g} \begin{bmatrix} b_{22} & -b_{12} \\ -b_{12} & b_{11} \end{bmatrix}.
\]

To the best of the authors’ knowledge, operator \( \diamond \) is newly introduced. For a vector-valued function \( \mathbf{f} = (f_1, \ldots, f_k)^T \in C^1(M)^k \), we define

\[
\nabla \mathbf{f} = [\nabla f_1, \ldots, \nabla f_k] \in \mathbb{R}^{3 \times k}, \quad \diamond \mathbf{f} = [\diamond f_1, \ldots, \diamond f_k] \in \mathbb{R}^{3 \times k}.
\]

Hence, it is easy to see that

\[
\nabla x = [x_u, x_v] [g^{\alpha \beta}] [x_u, x_v]^T, \tag{2.3}
\]

\[
\nabla n = -[x_u, x_v] [g^{\alpha \beta}] S [x_u, x_v]^T, \tag{2.4}
\]

and both \( \nabla x \) and \( \nabla n \) are symmetric \( 3 \times 3 \) matrices. From the definitions of operators \( \nabla \) and \( \diamond \), we can derive that...
\( \nabla \cdot \mathbf{v} = 0 \).

**Divergence operator.** Let \( \mathbf{v} \) be a \( C^1 \) smooth vector field on \( \mathcal{M} \). Then the divergence of \( \mathbf{v} \) is defined by

\[
\text{div}(\mathbf{v}) = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[ \sqrt{g} g^{\alpha \beta} \left[ x_{u \alpha} x_{v \beta} \right] T \mathbf{v} \right].
\]

(2.6)

Note that if \( \mathbf{v} \) is a normal vector field of \( \mathcal{M} \), \( \text{div}(\mathbf{v}) = 0 \).

For a matrix-valued function \( \mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_k] \in C^1(\mathcal{M})^{3 \times k} \), we define

\[
\text{div}(\mathbf{Q}) = (\text{div}(\mathbf{q}_1), \ldots, \text{div}(\mathbf{q}_k))^T \in \mathbb{R}^k.
\]

**Laplace–Beltrami operator.** Let \( f \) be a \( C^2 \) smooth function on \( \mathcal{M} \). Then \( \nabla f \) is a smooth vector field on \( \mathcal{M} \). The Laplace–Beltrami operator (LBO) \( \Delta \) applying to \( f \) is defined by (see do Carmo (1992), p. 83)

\[
\Delta f = \text{div}(\nabla f).
\]

(2.7)

From the definitions of \( \nabla \) and \( \text{div} \), we can derive that

\[
\Delta f = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[ \sqrt{g} g^{\alpha \beta} (f_{u \alpha}, f_{v \beta})^T \right]
\]

(2.8)

\[
= g^{\alpha \alpha} f_u + g^{\alpha \beta} f_v + g^{u \alpha} f_{u u} + g^{u \beta} f_{u v} + g^{v \alpha} f_{u v} + g^{v \beta} f_{v v},
\]

(2.9)

\[
= \frac{1}{g} \left( g_{22} f_{11} + g_{11} f_{22} - 2 g_{12} f_{12} \right)
\]

(2.10)

\[
= [g^{\alpha \beta}]: [f_{u \beta}] 
\]

(2.11)

where

\[
g^{\alpha \alpha} = -\left( g_{11} (g_{22} g_{12} - g_{12} g_{22}) + 2 g_{12} (g_{12} g_{22} - g_{22} g_{11} - g_{12} g_{11}) \right) / g^2,
\]

\[
g^{\alpha \beta} = -\left( g_{11} (g_{12} g_{22} - g_{22} g_{11} + g_{12} g_{22}) + 2 g_{12} (g_{12} g_{22} - g_{22} g_{11} - g_{12} g_{11}) \right) / g^2,
\]

\[
g^{u \alpha} = g_{22} / g, \quad g^{u \beta} = -2 g_{12} / g, \quad g^{v \alpha} = g_{11} / g,
\]

\[
\text{with } g^{\alpha \beta} = \langle x_{u \alpha}, x_{u \beta} \rangle, \text{ and }
\]

\[
f_{u \alpha} = f_{u u \alpha} - (\nabla f)^T x_{u u \alpha}, \quad \alpha, \beta = 1, 2,
\]

are the second covariant derivatives. The notation \( A:B \) stands for the trace of \( A^T B \). It is easy to see that \( \Delta \) is a second order differential operator.

**Remark 2.1.** Let us explain how (2.8)–(2.11) are derived. First, equality (2.8) is obtained by simply substituting (2.2) into (2.7). (2.9) is derived from (2.8) by a straightforward computation of the first order partial derivative of \( \sqrt{g} g^{\alpha \beta} (f_{u \alpha}, f_{v \beta})^T \) using the product rule for differentiation. Then (2.10) follows easily from (2.9). Writing (2.10) into a matrix form, we obtain the last equality.

\( \square \) operator. Let \( f \) be a \( C^2 \) smooth function on \( \mathcal{M} \). Then the \( \square \) operator acting on \( f \) is given by

\[
\square f = \text{div}(\nabla f).
\]

(2.12)

From the definitions of \( \nabla \) and \( \text{div} \), there is no difficulty to derive that

\[
\square f = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[ \sqrt{g} h^{\alpha \beta} (f_{u \alpha}, f_{v \beta})^T \right]
\]

(2.13)

\[
= g^{\alpha \alpha} f_u + g^{\alpha \beta} f_v + g^{u \alpha} f_{u u} + g^{u \beta} f_{u v} + g^{v \alpha} f_{u v} + g^{v \beta} f_{v v},
\]

(2.14)

\[
= \frac{1}{g} \left( b_{22} f_{11} + b_{11} f_{22} - 2 b_{12} f_{12} \right)
\]

(2.15)

\[
= [h^{\alpha \beta}]: [f_{u \beta}].
\]

(2.16)
Remark 2.2. Equalities (2.13)–(2.16) can be derived in the same way as the equalities (2.8)–(2.11). We ignore the details.

For the operators introduced above, we can prove the following theorems that are used in the sequel. The sketches of the proofs are given in Appendix A. Detailed proofs can be found in our technical report (Zhang and Xu, 2007).

Theorem 2.1. Let \( x \in \mathcal{M} \). Then
\[
\Delta x = 2Hn,
\]
\[
\Delta n = -2\nabla H - 2H\Delta x + \square x.
\]

Theorem 2.2. Let \( x \in \mathcal{M} \). Then
\[
\square x = 2Kn,
\]
\[
\square n = -\nabla K - K\Delta x = -\nabla K - H\square x.
\]

Remark 2.3. The first equality of Theorem 2.1 is well-known. We believe that the second equality and the equalities of Theorem 2.2 are newly established.

Theorem 2.3. If \( v \) is a smooth three-dimensional vector field on \( \mathcal{M} \) and \( f \in C^1(\mathcal{M}) \), then
\[
\text{div}(v) = v^T \Delta x + \nabla x : \nabla v + \nabla n : \nabla v,
\]
\[
\text{div}[f \nabla v] = f \Delta v + [\nabla v]^T \nabla f,
\]
\[
\nabla n : \nabla v = -2H \nabla x : \nabla v + \nabla x : \square v,
\]
where
\[
\nabla x v = [\nabla x v_1, \nabla x v_2, \nabla x v_3], \quad \nabla n v = [\nabla n v_1, \nabla n v_2, \nabla n v_3],
\]
\[
\nabla x v_i = \left( \frac{\partial v_i}{\partial x_1}, \frac{\partial v_i}{\partial x_2}, \frac{\partial v_i}{\partial x_3} \right)^T, \quad \nabla n v_i = \left( \frac{\partial v_i}{\partial n_1}, \frac{\partial v_i}{\partial n_2}, \frac{\partial v_i}{\partial n_3} \right)^T
\]
with \( v = (v_1, v_2, v_3)^T, \ x = (x_1, x_2, x_3)^T \) and \( n = (n_1, n_2, n_3)^T \).

Theorem 2.4 (Green’s formula). Let \( v \) be a smooth three-dimensional vector field on \( \mathcal{M} \) and \( f \in C^1(\mathcal{M}) \) with compact support, then
\[
\int_{\mathcal{M}} \langle v, \nabla f \rangle \, dA = -\int_{\mathcal{M}} f \, \text{div}(v) \, dA.
\]

Note that the Green’s formula could also be regarded as a definition of the divergence operator \( \text{div} \). Since we introduce the divergence operator \( \text{div} \) by (2.6), we therefore treat the Green’s formula as a theorem.
3. Geometric PDEs

First let us examine the energy of a flow. Generally speaking, any flow possesses its own energy because it cannot move without the driving of an energy. For instance, mean curvature flow is the $L^2$ gradient flow of the area functional. Surface diffusion flow comes also from the area functional in the sense of $H^{-1}$ inner product, a notion proposed by Fife (1991). From squared mean curvature functional, we can derive Willmore flow. To generalize the area functional for meeting different requirements in various fields, many energy functionals have been introduced. In chemistry and biology fields such as phase transition, crystal growth (Gurtin and Jabbour, 2002), biologic vesicle (Ou-Yang et al., 1999), various functionals have been proposed. In computer graphics and image analysis such as shape segmentation, boundary detection (Caselles et al., 1997), tracking, different energy models (Mumford and Shah, 1989) have also been used. Having the notion of the flow energy in mind, we construct the geometric flows in this paper from a general energy functional.

3.1. Euler–Lagrange equation derivation

Let $f(H, K) \in C^1(\mathbb{R} \times \mathbb{R})$ be a Lagrange function. Then the Euler–Lagrange equation in the normal direction variation for the curvature function integral

$$\mathcal{F}(\mathcal{M}) = \int_{\mathcal{M}} f(K, H) \, dA = \int \int \int f(K, H) \sqrt{g} \, du \, dv$$

over surface $\mathcal{M} = \{x(u, v); (u, v) \in \Omega \subset \mathbb{R}^2\}$ is given by (see Giaquinta and Hildebrandt (1996), pp. 82–85)

$$\Box f_K + \frac{1}{2} \Delta f_H + 2HKf_K + (2H^2 - K)f_H - 2fH = 0,$$  \hspace{1cm} (3.1)

where $f_K$ and $f_H$ are partial derivatives of $f$ with respect to its arguments $K$ and $H$, respectively. The scalar equation (3.1) is derived by considering a family of normal-variation $x(w, \varepsilon)$ of $\mathcal{M}$ defined by

$$x(w, \varepsilon) = x(w) + \varepsilon \phi(w)n, \quad w \in \overline{\Omega}, \ |\varepsilon| \ll 1, \ \phi \in C^\infty_c(\Omega).$$

In this paper we intend to derive a vector-valued equation from the complete-variation

$$x(w, \varepsilon) = x(w) + \varepsilon \phi(w)n, \quad w \in \overline{\Omega}, \ |\varepsilon| \ll 1, \ \phi \in C^\infty_c(\Omega)^3,$$  \hspace{1cm} (3.2)

for the functional

$$\mathcal{E}(\mathcal{M}) = \int_{\mathcal{M}} f(K, H) \, dA + \int \int h(x, n) \, dA,$$  \hspace{1cm} (3.3)

where $f(H, K) \in C^1(\mathbb{R} \times \mathbb{R})$ and $h(x, n) \in C^2(\mathbb{R}^3 \times \mathbb{R}^3 \setminus \{0\})$. Note that normal-variation deforms the surface point $x$ in the normal direction only (not tangential direction). In contrast, complete-variation allows the surface point $x$ to move in both normal and tangential directions. Thus normal-variation is part of complete-variation. We summarize the obtained result as the following theorem.

**Theorem 3.1.** Let $f(H, K) \in C^1(\mathbb{R} \times \mathbb{R})$ and $h(x, n) \in C^2(\mathbb{R}^3 \times \mathbb{R}^3 \setminus \{0\})$, respectively. Then the Euler–Lagrange equation of the integral $\mathcal{E}(\mathcal{M})$ from the complete-variation (3.2) is

$$\Box(f_K n) + \frac{1}{2} \Delta(f_H n) - \text{div}[f_H \nabla n] - \text{div}[(f + h - 2Kf_K)\nabla x] + \nabla_x h + \text{div}(\nabla_n h)n + \nabla_n \nabla_n h = 0.$$  \hspace{1cm} (3.4)

**Proof.** Suppose that $\mathcal{M}$ is an extremal surface of functional (3.3). Then we obtain

$$0 = \frac{d}{d\varepsilon} \mathcal{E}(\mathcal{M}(\cdot, \varepsilon)) \bigg|_{\varepsilon = 0} = : \delta \mathcal{E}(\mathcal{M}, \phi),$$
where
\[ \delta \mathcal{E}(M, \Phi) = \int \int_{\Omega} \left( f_H \delta(H) + f_K \delta(K) + (\nabla_x h)^T \delta(x) + (\nabla_n h)^T \delta(n) + (f + h)\delta(\sqrt{g})/\sqrt{g} \right) \sqrt{g} \, du \, dv. \] (3.5)

Now we compute \( \delta(H) \), \( \delta(K) \), \( \delta(x) \), \( \delta(n) \) and \( \delta(\sqrt{g}) \) in (3.5) separately. First noticing that
\[ x = x + \epsilon \Phi, \]
\[ x^\alpha = x^\alpha + \epsilon \Phi^\alpha, \]
\[ x^\alpha_{\mu \beta} = x^\alpha_{\mu \beta} + \epsilon \Phi^\alpha_{\mu \beta}, \]
then we obtain
\[ \delta(x) = \Phi, \] (3.6)
\[ \delta(g_{\alpha \beta}) = (\Phi^\alpha_{\mu}, x^\beta_{\mu}) + (\Phi^\beta_{\mu}, x^\alpha_{\mu}), \quad \alpha, \beta = 1, 2, \]
\[ \delta(g) = 2g \nabla x : \nabla \Phi, \]
\[ \delta(\sqrt{g})/\sqrt{g} = \nabla x : \nabla \Phi. \] (3.7)

Since \( \delta(n) \) is a tangent vector to \( M \), we thus have
\[ \delta(n) = -[x^\mu, x^\nu][g^\alpha \beta] \begin{bmatrix} (n, \Phi^\mu) \\ (n, \Phi^\nu) \end{bmatrix} = -\nabla \Phi n, \] (3.8)
and
\[ \delta(b_{\alpha \beta}) = (\delta(n), x^\mu_{\nu \alpha \beta}) + (n, \delta(x^\mu_{\nu \alpha \beta})) \]
\[ = -\langle n, (\nabla \Phi)^T x^\mu_{\nu \alpha \beta} \rangle + \langle n, \Phi^\mu_{\nu \alpha \beta} \rangle \]
\[ = (n, \Phi^\alpha_{\mu \beta}). \]

Therefore,
\[ \delta(H) = \frac{1}{2} (b_{11} g_{22} + b_{22} g_{11} - 2b_{12} g_{12}) - \frac{g_2}{g} + \frac{1}{2g} \left( (n, \Phi_{11} g_{22} + \Phi_{22} g_{11} - 2\Phi_{12} g_{12}) \right. \]
\[ + 2b_{11} (x^\alpha, \Phi^\alpha) + 2b_{22} (x^\mu, \Phi^\mu) - 2b_{12} (x^\alpha, \Phi^\mu + (x^\mu, \Phi^\alpha)) \]
\[ = -2H \nabla x : \nabla \Phi + \frac{1}{2} \langle n, \Delta \Phi \rangle + \nabla x : \phi \]
\[ = \nabla n : \nabla \Phi + \frac{1}{2} \langle n, \Delta \Phi \rangle, \] (3.9)
where relations (2.1) and (2.25) are used.
\[ \delta(K) = \delta \left( \frac{1}{g} \right) (b_{11} b_{22} - b_{12}^2) + \frac{1}{g} \delta(b_{11} b_{22} - b_{12}^2) \]
\[ = -2K \nabla x : \nabla \Phi + \frac{1}{g} \langle n, \Phi_{11} b_{22} + \Phi_{22} b_{11} - 2\Phi_{12} b_{12} \rangle \]
\[ = -2K \nabla x : \nabla \Phi + \langle n, \Box \Phi \rangle. \] (3.10)

Substituting (3.6)–(3.10) into (3.5), we obtain
\[ 0 = \delta \mathcal{E}(M, \Phi) \]
\[ = \int \int_{\Omega} \left( f_H \nabla n : \nabla \Phi + (f + h - 2K f_K) \nabla x : \nabla \Phi + \frac{1}{2} f_H \langle n, \Delta \Phi \rangle + f_K \langle n, \Box \Phi \rangle \right. \]
\[ + (\nabla_x h)^T \Phi - (\nabla_n h)^T \nabla \Phi \n \right) \sqrt{g} \, du \, dv. \]
\[
\int_{\Omega} \left( - \text{div} (f_H \nabla n) - \text{div} [(f + h - 2Kf_K) \nabla x] + \frac{1}{2} \Delta (f_H n) + \Box (f_K n) \\
+ \nabla_x h + \text{div}(\nabla_n h) + \nabla n(\nabla_n h), \Phi \right) \sqrt{H} \, d\nu
\]

for any \( \Phi \in C^\infty_c(\Omega)^3 \). The last equality is valid owing to Green's formula (2.26). Therefore, the Euler–Lagrange equation of functional (3.3) is (3.4) and the theorem is proved. \( \square \)

From the theorem, we can derive Giaquinta and Hildebrandt’s result mentioned above. In fact, we have the following corollary.

**Corollary 3.1.** Let \( f(H, K) \in C^1(\mathbb{R} \times \mathbb{R}) \) and \( h(x, n) \in C^2(\mathbb{R}^3 \times \mathbb{R}^3 \setminus \{0\}) \), respectively. Then

\[
\Box (f_K n) + \frac{1}{2} \Delta (f_H n) - \text{div} [f_H \nabla n] - \text{div} [(f - 2Kf_K) \nabla x] = n \left( \Box f_K + \frac{1}{2} \Delta f_H + 2H Kf_K + (2H^2 - K)f_H - 2H f \right),
\]

\[
\nabla_x h + \text{div}(\nabla_n h) n + \nabla n \nabla_n h - \text{div}[h \nabla x] = n^T \nabla_x h + \text{div}(\nabla_n h) - 2h H).
\]

**Proof.** We first have the following equalities,

\[
\Box (f_K n) \overset{(A.6)}{=} (f_K n) - 2[K \nabla x]^T \nabla f_K + f_K \Box n
\]

\[
\overset{(2.22)}{=} (f_K n) - 2K \nabla f_K + f_K (-\nabla K - 2KH n),
\]

\[
\Delta (f_H n) \overset{(A.3)}{=} (\Delta f_H n) + 2[\nabla n]^T \nabla f_H + f_H \Delta n
\]

\[
\overset{(2.20)}{=} (\Delta f_H n) + 2[\nabla n]^T \nabla f_H + 2f_H (K n - 2H^2 n - \nabla H),
\]

\[
\text{div} [f_H \nabla n] \overset{(A.2)}{=} f_H \Delta n + [\nabla n]^T \nabla f_H = 2f_H (K n - 2H^2 n - \nabla H) + [\nabla n]^T \nabla f_H,
\]

\[
\text{div} [(f - 2Kf_K) \nabla x] \overset{(A.2)}{=} (f - 2Kf_K) \Delta x + [\nabla x]^T \nabla (f - 2Kf_K)
\]

\[
= 2(f - 2Kf_K) H n + f_H \nabla H + f_K \nabla K - 2K \nabla f_K.
\]

After substituting them into the left-hand side of (3.12), we can obtain the right-hand side. Noticing that

\[
\text{div}[h \nabla x] = h \Delta x + (\nabla x, \nabla h) = h \Delta x + \nabla h = h \Delta x + \nabla x \nabla_x h + \nabla n \nabla_n h,
\]

we can derive (3.13). \( \square \)

### 3.2. The geometric flows

For given functions \( f(H, K) \) and \( h(x, n) \), let \( M_0 \) be a given initial surface with boundary \( \Gamma \) (see Fig. 1(a)). Then the geometric flow consists of finding a family \( \{ M(t) : t \geq 0 \} \) of smooth orientable surfaces in \( \mathbb{R}^3 \) which evolve according to the following equation

\[
\begin{cases}
\partial_t x + \Box (f_K n) + \frac{1}{2} \Delta (f_H n) - \text{div} [f_H \nabla n] - \text{div} [(f + h - 2Kf_K) \nabla x] \\
+ \nabla_x h + \text{div}(\nabla_n h) n + \nabla n \nabla_n h = 0,
\end{cases}
\]

\[
M(0) = M_0, \quad \partial M(t) = \Gamma.
\]

Note that \( \Box (f_K n) \) and \( \Delta (f_H n) \) involve the fourth order derivatives of \( M \) in general. The equation is of the fourth order if \( f_H \) or \( f_K \) is not a constant function. Now we reformulate Eq. (3.15). With the help of (2.24), (2.25), (A.10) and Theorem 2.1, we have

\[
\text{div} [f_H \nabla n] = f_H \Box x - 2f_H \Delta x - 2f_H \nabla H + \Box f_H - 2H \nabla f_H.
\]
On the other hand, equality (3.14) yields
\[ \nabla x h + \nabla \mathbf{n} \nabla_n h - \text{div}[h \nabla x] = [I - \nabla x] \nabla x h - h \Delta x. \]
Therefore the first equation of (3.15) becomes
\[
\frac{\partial \mathbf{x}}{\partial t} + \Box(f_2 \mathbf{n}) + \frac{1}{2} \Delta(f_2 \mathbf{n}) - f_2 \Box x + (2Hf_2 + 2Kf_2 - f - h) \Delta x
+ [I - \nabla x] \nabla x h + \text{div}(\nabla_n h) \mathbf{n} + 2f_2 \nabla H - \partial f_H + 2H \nabla f_H - \nabla (f - 2Kf_2) = 0. \tag{3.16}
\]
Suppose that \( f_H \) and \( f_K \) could be represented as
\[ f_H = 2\alpha H + 2\beta K + \mu, \quad f_K = 2\gamma H + 2\delta K + \nu. \tag{3.17} \]
Then (3.16) becomes
\[
\frac{\partial \mathbf{x}}{\partial t} + \Box(\gamma \Delta x + \delta \nabla x + \nu \mathbf{n}) + \frac{1}{2} \Delta(\alpha \Delta x + \beta \nabla x + \mu \mathbf{n}) - f_H \Box x
+ (2Hf_2 + 2Kf_2 - f - h) \Delta x + [I - \nabla x] \nabla x h + \text{div}(\nabla_n h) \mathbf{n}
+ 2f_2 \nabla H - \partial f_H + 2H \nabla f_H - \nabla (f - 2Kf_2) = 0, \tag{3.18}
\]
where \( \text{div}(\nabla_n h) \) is computed, from (2.23), by
\[
\text{div}(\nabla_n h) = (\nabla_n h)^T \Delta x + \nabla x \left[ \nabla^2_n h \right] + \nabla \mathbf{n} \left[ \nabla^2_n h \right] \tag{3.19}
\]
with \( \nabla^2_n h = \nabla_n \nabla_h \mathbf{n} \in \mathbb{R}^{3 \times 3} \) and \( \nabla^2 h = \nabla_n \nabla h \in \mathbb{R}^{3 \times 3} \). The nonlinear equation (3.18) is our starting point of discretization. Advantages of using (3.18) instead of (3.1) is addressed in the next section (see Remark 4.4).

**Remark 3.1.** In the discussion above, we do not impose any restriction on \( f(H, K) \) and \( h(x, n) \). In order to make the generated flow meaningful and well-defined. We first require that \( f \) and \( h \) are smooth functions about their arguments. We also assume that \( f \) and \( h \) are algebraic functions, meaning they do not involve differential and integral operations about their variables. To make the geometric flow independent of the orientation of the surface normal, we further assume that
\[
\begin{align*}
\nabla x h(x, n) &= -\nabla x (-H, K), & \quad \nabla n h(x, n) &= -\nabla n (x, -n), \\
\nabla x h(x, -n) &= -\nabla x (H, K), & \quad \nabla n h(x, -n) &= -\nabla n (x, n).
\end{align*} \tag{3.20}
\]
Finally, we assume that the energy density \( f(H, K) + h(x, n) \geq 0 \) such that the energy functional is non-negative in order to decrease it to a finite limit. Otherwise, an undetermined flow may be obtained. For example, Gauss curvature flow is fallen into this category.
4. Numerical solving of the geometric PDE

Let $M$ be a triangulation of surface $\mathcal{M}$. Let $\{x_i\}_{i=1}^N$ be the vertex set of $M$. For a vertex $x_i$ with valence $l$, denote by $N(i) = \{i_1, i_2, \ldots, i_l\}$ the set of the vertex indices of $x_i$ and its one-ring neighbors. We assume in what follows that these $i_1, \ldots, i_l$ are arranged such that the triangles $[x_ix_{i_1}x_{i_2}]$ and $[x_ix_{i_k}x_{i_{k+1}}]$ are in $M$, and $x_{i-1}, x_{i+1}$ are opposite to the edge $[x_ix_{i_k}]$.

4.1. Discretizations of geometric differential operators

To solve the geometric PDE (3.16) or (3.18) using a divided-difference-like method, discrete approximations of mean curvature, Gaussian curvature, Laplace–Beltrami operator and $\Box$ operator are required. Many discrete schemes have been proposed for geometric differential operators from different points of view (see Langer et al. (2005), Meek and Walton (2000), Meyer et al. (2002), Xu (2004, 2006) for references). Except for the schemes based on the interpolation or fitting, none of these schemes converges without any restriction on the regularity of the meshes considered. In this paper, all the used differential operators are approximated based on a parametric quadratic fitting. In order to use a semi-implicit scheme, the approximations of the above-mentioned differential operators are required to have the following form

$$\Delta f(x_i) = \sum_{j \in N(i)} w_{ij}^\Delta f(x_j), \quad \Box f(x_i) = \sum_{j \in N(i)} w_{ij}^\Box f(x_j), \quad w_{ij}^\Delta, \ w_{ij}^\Box \in \mathbb{R}. \ (4.1)$$

**Definition 4.1.** A set of approximate geometric differential operators is consistent if there exists a smooth surface $S$, such that the approximate operators coincide with the exact counterparts of $S$.

Note that the definition of consistency between approximate differential operators is irrelevant to the consistency between differential equation and difference equation. Since using different approaches to discretize various differential operators may lead to conflicting results, the consistent approximations of differential operators are of great importance. Here we use a biquadratic fitting of the surface data and function data to calculate the approximate differential operators. The algorithm we adopted is from Xu (2004). Let $x_i$ be a vertex of $M$ with valence $l$ and $x_j$ its neighbor vertices for $j \in N(i)$.

**Algorithm 4.1 (Quadratic fitting).**

1. Compute angles $\alpha_k = \cos^{-1}((x_{ik} - x_i, x_{ik+1} - x_i)/\|x_{ik} - x_i\| \times \|x_{ik+1} - x_i\|)$, and then compute the angles $\beta_k = 2\pi \alpha_k / \sum_{j=1}^{l} \alpha_j$ for $k = 1, \ldots, l$. Set $q_0 = (0, 0), \theta_0 = 0$ and $q_k = \|x_{ik} - x_i\|(\cos \theta_k, \sin \theta_k)$, $\theta_k = \beta_1 + \cdots + \beta_k - 1$, for $k = 1, \ldots, l$.
2. Take the basis functions $\{B_j(u, v)\}_{j=0}^5 = \{1, u, v, \frac{1}{2}u^2, uv, \frac{1}{2}v^2\}$, and determine the coefficient $c_j \in \mathbb{R}^3$ of $\sum_{j=0}^5 c_j B_j$ so that

$$\sum_{j=0}^5 c_j B_j(q_k) = x_{i_k}, \quad k = 0, \ldots, l \ (\text{assume } i_0 = i),$$

in the least square sense. This system is solved by solving the normal equation. Let $A = [B_j(q_k)]_{k=0, j=0}^{l, 5} \in \mathbb{R}^{(l+1) \times 6}$, and let

$$C = [A^T A]^{-1} A^T \in \mathbb{R}^{6 \times (l+1)}, \ (4.2)$$

then $[c_0, \ldots, c_5] = [x_{i_0}, \ldots, x_{i_l}] C^T$.  

Please cite this article in press as: G. Xu, Q. Zhang, A general framework for surface modeling using geometric partial differential equations, Computer Aided Geometric Design (2007), doi:10.1016/j.cagd.2007.06.002
Partial derivatives. Let \( (d_0, \ldots, d_5)^T = C(f(x_{i0}), \ldots, f(x_{il}))^T \). Then we compute the partial derivatives up to the second order. Denote the second, third, fourth, fifth and sixth rows of \( C \) as \( C_1, C_2, C_{11}, C_{12} \) and \( C_{22} \), respectively, then we can see that

\[
\begin{align*}
&x_{\alpha} = [x_{i0}, \ldots, x_{il}] C_\alpha^T, \quad \alpha = 1, 2, \\
&\frac{\partial f}{\partial u^\alpha} = (f(x_{i0}), \ldots, f(x_{il})) C_\alpha^T, \quad \alpha = 1, 2, \\
&x_{\alpha\beta} = [x_{i0}, \ldots, x_{il}] C_{\alpha\beta}^T, \quad 1 \leq \alpha \leq \beta \leq 2, \\
&\frac{\partial^2 f}{\partial u^\alpha \partial u^\beta} = (f(x_{i0}), \ldots, f(x_{il})) C_{\alpha\beta}^T, \quad 1 \leq \alpha \leq \beta \leq 2.
\end{align*}
\]

Laplace–Beltrami operator. Substituting (4.3) into (2.9), we get an approximation of LBO as follows:

\[
\Delta f(x_i) \approx \sum_{j \in N(i)} w_{ij}^\Delta f(x_j),
\]

where \( w_{ij}^\Delta = g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) \). Here \( c(j) \) and \( c(j) \) are the \( j \)-th components of \( C_a \) and \( C_{\alpha\beta} \), respectively. Using the relation \( 2 H n = \Delta x \), mean curvature \( H(x_i) \) is easily computed.

\[ \square \operatorname{operator}. \]

Substituting (4.3) into (2.14), we get an approximation of \( \square \) as follows:

\[
\square f(x_i) \approx \sum_{j \in N(i)} w_{ij}^\square f(x_j),
\]

where \( w_{ij}^\square = g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) + g \Delta_{ij} c(j) \). Using the relation \( 2K n = \square x \), Gaussian curvature \( K(x_i) \) can be computed.

Remark 4.1. The quadratic fitting algorithm above may fail if the coefficient matrix of the normal equation is singular or nearly singular. In this case, we look for a least square solution with minimal normal. Let \( \Delta x = \mathbf{b} \) be the linear system in the matrix form. We find a least square solution \( \mathbf{x} \) such that \( \| \mathbf{x} \|_2 \) is minimized. That is, we replace \( [A^T A]^{-1} \) in (4.2) with the Moore–Penrose inverse \( [A^T A]^+ \). It is well known that \( [A^T A]^+ \) could be computed by the SVD decomposition of \( A^T A \) (see Xu (2007) for more details).

Remark 4.2. Now we explain why we derive the used differential operators based on the parametric fitting. The first reason is that this fitting scheme yields a convergent approximation in general as the mesh size (the maximal edge length) \( h \to 0 \) (see Theorem 4.1 in the following). The second reason is that the computation of these operators is consistent. The third reason is that the fitting scheme yields the required form expressions, which are ready for use in the semi-implicit discretization of the PDEs. The last reason is that all the differential operators used in this paper involve the first and second order derivatives of the surface or functions on the surface. Hence, quadratic function is enough to provide these partial derivative data.

Now let us quote a result from Xu (2007). The proof of this result is beyond the scope of this paper.

Theorem 4.1. Let \( f(q) \) be a smooth function around the origin \( q_0 = (0, 0) \), and let \( \{q_i\}_{i=0}^l \) be a well-posed node set. Let \( G(h) = \sum_{i+j \leq 2} a_{ij} \frac{g^i}{\partial u^i} \frac{g^j}{\partial v^j} \) be the quadratic fitting function generated by Algorithm 4.1 for the sampling data \( \{h q_i, f(h q_i)\}^l \in \mathbb{R}^3 \) of function \( f \). Then

\[
\left| a_{ij} \frac{\partial^i+j f(q_0)}{\partial u^i \partial v^j} \right| \leq c_{ij} h^{3-i-j}, \quad i + j \leq 2, \quad \text{as } h \to 0,
\]

where \( c_{ij} \) are constants depending on \( f \).

The node set \( \{q_i\}_{i=0}^l \) is called well-posed, provided that matrix \( A \) in (4.2) is of full rank in row. The theorem implies that the approximate normal from fitting function converges in the rate \( O(h^2) \), the approximate curvatures converge in the rate \( O(h) \).
4.2. Main algorithm

Given are suitable smooth bivariate functions \( f(H, K) \) and \( h(x, n) \). Suppose that we have a subroutine for computing \( f, \alpha, \beta, \mu, \gamma, \delta \) and \( \nu \) at a given point \((H, K)\) and another subroutine for computing \( h, \nabla_xh, \nabla_nh, \nabla_{xx}h \) and \( \nabla_{nn}^2h \). We further assume that an initial surface mesh \( M^{(0)} \) is provided with certain vertices labeled as inner (see Fig. 1(b)).

An explicit Euler scheme for temporal discretization is unstable in general, therefore requires a small temporal step-size. To make the evolution process more efficient, an implicit scheme is more desirable. However, since the PDE is highly nonlinear, a complete implicit scheme is hard to solve. In the following we present a semi-implicit scheme, which leads to a linear system of equations. The basic idea for forming the linear system is to represent \( \nabla f, \alpha, \beta, \mu, \gamma, \delta \) as linear combinations of its neighbor vertices. Starting from \( M^{(0)} \), the following algorithm generates a sequence of triangular surfaces \( \{M^{(m)}\}_{m \geq 0} \).

**Algorithm 4.2 (Semi-implicit scheme).**

1. Set \( m = 0 \), \( x_i^{(0)} = x_i \) and set temporal step-size \( \tau \).
2. For each inner vertex \( x_i^{(m)} \), \( i = 1, \ldots, n \), compute a quadratic fitting function by Algorithm 4.1. Then compute \( n_i^{(m)} = n(x_i^{(m)}) \), \( H_i = H(x_i^{(m)}) \), \( K_i = K(x_i^{(m)}) \), \( w_{ij}^\Delta \) and \( w_{ij}^\nabla \) from the fitting function, and then compute \( f, \alpha, \beta, \mu, \gamma, \delta \) and \( \nu \) at \((H_i, K_i)\) using the provided subroutine. These values are denoted by \( f_i, \alpha_i, \beta_i, \mu_i, \gamma_i, \delta_i \) and \( \nu_i \), respectively. We further compute \( \nabla_xh(x_i^{(m)}, n_i^{(m)}) \) and \( \text{div}(\nabla_nh(x_i^{(m)}, n_i^{(m)})) \) using (3.19).
3. Compute

\[
E(\mathcal{M}^{(m)}) \approx \sum_i A_{i}^{(m)}(x_i^{(m)})(f(H(x_i^{(m)}), K(x_i^{(m)}))) + h(x_i^{(m)}, n_i^{(m)})
\]

where \( A_{i}^{(m)}(x_i^{(m)}) \) is the area of the Voronoi region of \( M^{(m)} \) around \( x_i^{(m)} \). If \( E(\mathcal{M}, m) \) is stable, that is

\[
|E(\mathcal{M}^{(m)}) - E(\mathcal{M}^{(m-1)})| \leq \tau \epsilon,
\]

for a given threshold value \( \epsilon \), or \( m \) is beyond a given number of iteration, then we terminate the computation. Otherwise, go to the next step.
4. For each inner vertex \( x_i^{(m)} \), \( i = 1, \ldots, n \), discretize (3.18) as the following linear system

\[
x_i^{(m+1)} + \tau \left( \sum_{j \in N(i)} c_{ij} x_j^{(m+1)} + b_i \right) = x_i^{(m)}, \quad c_{ij} \in \mathbb{R}, \quad b_i \in \mathbb{R}^3,
\]

where the first term of (3.18) is approximated by \( \frac{x_i^{(m+1)} - x_i^{(m)}}{\tau} \) and the other terms are approximated by \( \sum c_{ij} x_j^{(m+1)} + b_i \).
5. Solve the linear system (4.7) for the unknowns \( x_i^{(m+1)} \), for \( i = 1, \ldots, n \). We use Saad’s iterative method (see Saad (2000)), named GMRES, with incomplete LU decomposition as a pre-conditioner to solve the system. The experiments show that this iterative method works very well.
6. Set \( m \) to be \( m + 1 \) and go back to step 2.

**Spatial discretization.** Now let us explain in detail how Eq. (3.18) is spatially discretized to derive the linear system (4.7). Using (4.5) and (4.4), we have

\[
\square(f \nabla h)|_{x_i} \approx \sum_{j \in N(i)} w_{ij}^\nabla \left( \gamma_j \Delta(x_j^{(m+1)}) + \delta_j \square(x_j^{(m+1)}) + n_j^{(m)} v_j \right)
\approx \sum_{j \in N(i)} w_{ij}^\nabla \sum_{k \in N(j)} \left( \gamma_j w_{jk}^\Delta + \delta_j w_{jk}^\nabla \right) x_k^{(m+1)} + \sum_{j \in N(i)} w_{ij}^\square v_j n_j^{(m)}, \quad (4.8)
\]
\[ \Delta(fHn)|_{x_i} \approx \sum_{j \in N(i)} w_{ij}^\Lambda (\alpha_j \Delta(x_j^{(m+1)}) + \beta_j \square(x_j^{(m+1)}) + n_j^{(m)} \mu_j) \]

\[ \approx \sum_{j \in N(i)} w_{ij}^\Lambda \sum_{k \in N(j)} (\alpha_j w_{jk}^\Lambda + \beta_j w_{jk}^\square) x_k^{(m+1)} + \sum_{j \in N(i)} w_{ij}^\Lambda \mu_j n_j^{(m)}, \]  

(4.9)

\[ fH \square x_i \approx fH \sum_{j \in N(i)} w_{ij}^\square x_j^{(m+1)}, \]

(4.10)

\[ (2HfH + 2KfK - f - h) \Delta x_i \approx (2H_i fH + 2K_i fK - f_i - h_i) \sum_{j \in N(i)} w_{ij}^\Lambda x_j^{(m+1)}, \]  

(4.11)

\[ [I - \nabla x] \nabla x h + \text{div}(\nabla n h) n_i \approx [I - \nabla x_i^{(m)}] \nabla x h (x_i^{(m)}, n_i^{(m)}) + \text{div}(\nabla n h (x_i^{(m)}, n_i^{(m)})) n_i^{(m)}, \]  

(4.12)

\[ 2fH \nabla H - \square fH + 2H \nabla fH - \nabla (f - 2KfK)|_{x_i} \approx 2fH \nabla H - \square fH + 2H \nabla fH - \nabla (f - 2KfK)|_{x_i^{(m)}}. \]  

(4.13)

Plugging (4.8)–(4.13) into (3.18) and adding the coefficient of \( x_k^{(m+1)} \) to \( c_{ij} \) if \( x_k^{(m+1)} \) is an inner vertex, and adding the remaining terms to \( b_i \), we obtain (4.7).

**Remark 4.3.** The coefficient matrix of the system is very sparse. Since each inner vertex involves only its two-ring neighbors, the number of nonzero elements for each row of the coefficient matrix is 19 in average (assume the average of the vertex-valence is 6).

**Remark 4.4.** Now let us explain why we use Euler–Lagrange equation (3.4) instead of (3.1). The main reason is that flow (3.15) constructed from Euler–Lagrange equation (3.4) does not contain the terms \( n \Delta H, n \square H, n \Delta K \) and \( n K \). As a result, the coefficients \( c_{ij} \) in (4.7) are scalars, while the coefficients of the geometric flow constructed from (3.1) are \( 3 \times 3 \) matrices (see (Xu et al., 2006)). For instance, \( n \Delta H \) is discretized as

\[ n \Delta H|x_i \approx n_i^{(m)} \sum_{j \in N(i)} w_{ij}^\Lambda H_j \approx \frac{1}{2} \sum_{j \in N(i)} w_{ij}^\Lambda \sum_{k \in N(j)} n_j^{(m)} (n_j^{(m)})^T w_{jk}^\Lambda x_k, \]

where \( n_i^{(m)} (n_j^{(m)})^T \in \mathbb{R}^{3 \times 3} \). Since \( c_{ij} \) are scalars, system (4.7) can be solved separately by splitting it into three sub-systems with the same coefficient matrix \( (\delta_{ij} + \tau c_{ij}) \). Hence the new formulation not only saves the storage but also accelerates the computation. More importantly, the numerical behavior of the small systems is better than that of the large one, and therefore requires less iteration steps for solving the system. To see this more clearly, let us examine the discretization of (3.1). Suppose that (3.1) is linearized as

\[ \sum_j d_{ij}^T x_j = e_i, \quad d_{ij} \in \mathbb{R}^3, \quad e_i \in \mathbb{R}. \]

Taking an inner product of this equation with \( n_i^{(m)} \), we get

\[ \sum_j c_{ij} x_j = n_i^{(m)} e_i, \quad c_{ij} = n_i^{(m)} d_{ij}^T \in \mathbb{R}^{3 \times 3}, \quad i = 1, \ldots, n. \]

Note that this system is of low rank, since the three equations at vertex \( x_i \) are of rank one.

**5. Examples**

In this section, we give several examples to show the strength of the proposed approach. Considering the approach is general, there are infinitely many possibilities for choosing \( f \) and \( h \) as well as the geometric models. The examples provided here are just a few of them. In the illustrative figures of this section, \( \tau \) and \( T \) stand for the temporal step-size and the number of iteration used.

**5.1. The effects of the geometric PDEs**

To illustrate the effects caused by the various geometric PDEs, we use a few simple surface models, such as cylinder, ring and sphere as the inputs, so that the shape deformations are distinguishable. Most of the figures presented...
Fig. 2. The input is a triangulated cylinder. We evolve the middle part of the cylinder using the geometric flows with different Lagrange functions $f(H, K)$ and $h(x, n) = 0$. Figures (a)–(f) are the evolution results of the corresponding flows. The boundaries are two circles on the cylinder, the initial surface to be evolved is the input cylinder between the two circles.

Fig. 3. The surface to be evolved is defined as a graph of a function $g: x(u, v) = (u, v, g(u, v))^T$, $g(u, v) = e(u, v) + e(u + 1, v) + e(u, v + 1) + e(u + 1, v + 1)$, with $(u, v) \in \Omega := [-1, 1]^2$ and $e(u, v) = \exp [-\frac{1}{16}(u - 0.5)^2 + (v - 0.5)^2]$. This surface is uniformly triangulated using a $60 \times 60$ grid over the domain $\Omega$. We evolve a part of the surface, where $g > 1.5$, with different Lagrange functions $f(H, K)$ and $h(x, n) = 0$. Figures (a)–(f) show the evolution results of the corresponding flows. The boundary curve is defined by $g = 1.5$, the surface $g > 1.5$ is served as the initial surface of the evolutions.

in the following show the steady solutions of the PDEs considered, except for a few of them which show short time solution, since longer time evolution may cause singularities.

**Example 5.1.** Take $f = 1, h = 0$, then (3.4) becomes $H = 0$. The resulted surface is a minimal surface (see Dierkes et al. (1992), Nitsche (1989) for references). Figs. (a), (3(a), 4(c) and 7(a) show four examples of the minimal surface constructions. If $f = H, h = 0$, (3.4) becomes $K = 0$. The resulted surface is a developable surface. From this equation, we get Gauss curvature flow. As we have noted in Remark 3.2, this flow is undetermined because the integrand $H$ is not necessarily positive. Hence all the previous results about this flow require that the surfaces to be evolved are strictly convex (Tso, 1985) or convex (Andrews, 1994). Fig. 8(a) shows a short time evolution result of a triangulated ring. If $f = K, h = 0$, the Euler–Lagrange equation is $0 = 0$, and $E(M)$ is an invariant integral.

**Example 5.2.** Take $f = \alpha |H|^p + \beta |K|^q$, $(p, q > 1, \alpha, \beta \geq 0), h = 0$. (3.4) becomes

$$\beta \left[q \Box |K|^{q-2}K + 2(q - 1)|K|^q H\right] + \alpha (p/2) \Delta (|H|^{p-2} H) + 2(p - 1) H |H|^p - pHK|H|^{p-2} = 0.$$  

In particular, if $\alpha = 1, \beta = 0$ and $p = 2$, the resulted surface is a Willmore surface (see Willmore (1993)), which satisfies the equation $\Delta H + 2(H^2 - K) = 0$. Figs. (b), (3(b), 4(c) and 5(d) show four examples of Willmore surface constructions. If $\alpha = 0, \beta = 1$ and $q = 2$, the resulted surface satisfies the equation $\Box K + HK^2 = 0$. Figs. (c) and 3(c) show two solution surfaces of the equation. In general, the result of this flow is not desirable for surface processing if the considered surface is not convex (see Fig. 8(b)). Figs. 2(d), 3(d) and 8(d) show three examples of the solution surfaces for $\alpha = \beta = 1$ and $p = q = 2$. 

Please cite this article in press as: G. Xu, Q. Zhang, A general framework for surface modeling using geometric partial differential equations, Computer Aided Geometric Design (2007), doi:10.1016/j.cagd.2007.06.002
Example 5.3. Take $f = |H|^p |K|^q$, $(p, q > 1)$, $h = 0$. (3.4) becomes
\[ q \Box (K |H|^p |K|^{q-2} + (p/2) \Delta (H|H|^p |K|^q) + (2p + 2q - 2) |H|^p |K|^q - pHK|H|^p |K|^{q-2} |K|^{q-2} = 0. \]

Figs. 2(e), 3(e) and 8(c) show three examples for $p = q = 2$. The smoothing effect of the flow constructed from the above equation is not desirable if the considered surface is not convex.

Example 5.4. Take $f = (|k_1|^p + |k_2|^p)^q$, $p > 1$, $q > 0$. If $p = 2k$, $q$ is a natural number, then $f$ is a polynomial of $H$ and $K$. For instance, if $p = 2$, $q = 1$, $f = k_1^2 + k_2^2 = 4H^2 - 2K$. But taking Gauss–Bonnet–Chern formula into consideration, we can easily find that this functional will generate Willmore surfaces. Figs. 2(f) and 3(f) show two solution surfaces of the equation for $p = 4$, $q = 1$. Figs. 4(b), (c) and (d) show three examples for $p = 2$ and $q = 1, 1.5, 2$, respectively.

This family of equations generates pleasing surfaces in general.

Example 5.5. Take $f = \frac{1}{2} k_c (2H^2 + c_0)^2 + \lambda$, $h = \frac{1}{2} P_\delta (x, n)$, where $k_c$, $c_0$, $\lambda$ and $P_\delta$ are constants, then (3.3) becomes
\[ E(\mathcal{M}) = \frac{1}{2} k_c \int_{\mathcal{M}} (2H^2 + c_0)^2 dA + P_\delta \int_{\mathcal{V}} dV + \lambda \int_{\mathcal{M}} dA, \]

where $\mathcal{M}$ is a closed surface and $\mathcal{V} \subset \mathbb{R}^3$ is the region enclosed by $\mathcal{M}$. This is the shape energy of biologic vesicle (see Ou-Yang et al. (1999)). Here the constant $k_c$ is the bending rigidity of the vesicle surface, $c_0$ is the spontaneous curvature, $P_\delta = P_{\text{out}} - P_{\text{in}}$ is the difference of the outer pressure and inner pressure of the vesicle, $\lambda$ is the strain of the surface. Then the Euler–Lagrange equation of $E(\mathcal{M})$ is
\[ P_\delta - 2\lambda H + k_c (2H^2 + c_0) (2H^2 - c_0 H - 2K) + 2k_c \Delta H = 0. \]

This is the general equation of the biologic vesicle.

Now let us consider two simple cases. As one may know, the red blood cell and ring-shaped vesicle minimize the energy:
\[ \int_{\mathcal{M}} (2H^2 + c_0)^2 dA, \quad \int_{\mathcal{M}} H^2 dA, \]
Fig. 5. We simulate the shape of two biomembranes—a red blood cell and a ring-shaped vesicle. (a) is an input closed surface served as an initial surface for the evolution, and (b) shows the evolution result for \( f = (H + c_0)^2, h = 0 \). (c) is an input ring, as an initial surface (without boundary). (d) is the evolution result of the input ring for \( f = H^2 \) and \( h = 0 \).

respectively. Hence, we take \( f = (2H + c_0)^2 \) and \( f = H^2, h = 0 \). Fig 5 shows the evolution results, where (a) is an initial approximation of the red blood cell, (b) is the evolution result, a circular biconcave discoid. Here, \( c_0 \) is taken to be \( 1.51/\sqrt{A/(4\pi)} \), where \( A \) is the area of the surface (see Ou-Yang et al. (1999) for more details on the theory of the shape of red blood cell). The second energy leads to Willmore surfaces. The ring type solution of the Willmore equation satisfies \( R/r = \sqrt{2} \), where the ring is formed by rotating a circle of radius \( r \) along another circle of radius \( R \). Figure (c) shows an input ring with \( R/r = 20/7 \), (d) is the evolution result, where \( R/r = 1.419 \approx \sqrt{2} \). Hence, the numerical solution does approach the exact one.

Example 5.6. In physical settings such as phase transitions, epitaxial deposition and grain growth, the following interfacial free energy

\[
\int_M \left( \psi_0(n) + \frac{1}{2} \varepsilon_1 \| \nabla n \|^2 + 2 \varepsilon_2 H^2 \right) dA
\]

is considered (see (Gurtin and Jabbour, 2002)), where \( \psi_0(n) \) is the interfacial free energy density, \( \| \nabla n \|^2 = \nabla n : \nabla n \), \( \varepsilon_1, \varepsilon_2 \) are constant scalar moduli with \( \varepsilon_1 > 0 \) and \( \frac{1}{2} \varepsilon_1 + \varepsilon_2 > 0 \) to ensure that the interfacial energy is positive-definite. This energy can be cast into our general framework with

\[
f(H, K) = 2(\varepsilon_1 + \varepsilon_2)H^2 - \varepsilon_1 K, \quad h(x, n) = \psi_0(n).
\]

For the sake of giving another category of examples that are important in physical and chemical settings, we point out that a function \( f \in C^1(\mathbb{R}^n \setminus \{0\}) \) is positively homogeneous of degree \( t \) (\( t \in \mathbb{R} \)), if \( f(\lambda x) = \lambda^tf(x) \) for all \( x \in \mathbb{R}^n \setminus \{0\} \) and positive real \( \lambda \).

Example 5.7. If we take \( f(H, K) = 0, h(x, n) \) a positive homogeneous function of degree \( t \), then we can write (3.4) as

\[
n^T \nabla_x h + 2H (t - 1) + \nabla x \left[ \nabla^2_{xx} h \right] + \nabla n \left[ \nabla^2_{mn} h \right] = 0.
\]

Please cite this article in press as: G. Xu, Q. Zhang, A general framework for surface modeling using geometric partial differential equations, Computer Aided Geometric Design (2007), doi:10.1016/j.cagd.2007.06.002
The left-hand side of (5.2) can be regarded as the definition of weighted mean curvature. Especially, if \( t = 1 \), Taylor (1992) named it as weighted mean curvature. But considering the first and third terms of the left-hand side of (5.2) relate to the position of the surface, the authors of (Clarenz, 2002; Zhang and Xu, 2005) took the last term, i.e.,

\[
\nabla \cdot (\nabla h) = \left( \sum_{i=1}^{n} n_{i}^{2} + 1 \right) \frac{h_{1}}{2} + \frac{h_{2}}{2} + \frac{h_{3}}{2},
\]

(5.3)

The evolved surface is a triangulated sphere. Fig. 6 shows that \( h_{1} \), \( h_{2} \) and \( h_{3} \) alter the sphere into an ellipsoid, a disk and an octahedron, respectively. \( h_{4} \) also alters the sphere into an octahedron, but speed is anisotropic. Fig. 6(d) shows an intermediate result.

**Example 5.8.** Let \( f = 1, h = \frac{2}{3} H_{0}(x, n) \), then the Euler–Lagrange equation (3.4) becomes \( H = H_{0} \). Obviously, a surface with constant mean curvature \( H_{0} \) satisfies this equation. Fig. 7 shows four constructed constant mean curvature surfaces.

**Remark 5.1.** The graphical examples show that the second order flow can achieve \( G^{0} \) continuity. The fourth order flow can achieve \( G^{1} \) continuity at the boundaries (see Figs. 2(b)-(f), Figs. 3(b)-(f) and Figs. 4(b)-(d), (f)-(h)). For surface processing and modeling, the fourth order flows derived from \( f = H^{2} \), \( f = H^{2} + K^{2} \) and \( f = (|k_{1}|^{p} + |k_{2}|^{p})^{\frac{1}{p}} \) produce desirable shaped smooth surfaces.

**Remark 5.2.** Some flows related to Gaussian curvature do not smooth surface well in general. Fig. 8 exhibits the case, where an input ring is evolved. If \( f = H \), wrinkle shaped bumpy is generated (see figure (a)). Note that \( f = H \) does not satisfy the condition (3.20). Though \( f = K^{2} \) and \( f = H^{2} K^{2} \) satisfy condition (3.20), but figures (b) and (c) show that the result surfaces are still bumpy at the non-convex parts, even for a short period of time. For \( f = H^{2} + K^{2} \), no such
Fig. 9. The input is the bunny model with noise added as shown in (a). We evolve the model using the geometric flows with different Lagrange functions $f(H,K)$ and $h(x,n) = 0$. Figures (b)–(f) are the evolution results of the corresponding flows.

artifacts occur even though $K^2$ is involved. Figure (d) shows a long-standing solution of the flow for $f = H^2 + K^2$, and the obtained surface is smooth.

5.2. Surface smoothing and blending

In this subsection, we present two application examples. One is surface smoothing, the other is surface blending. For surface smoothing, we are given a noisy initial surface, and then use different PDEs to evolve the surface. The aim is to see how well the different PDEs behave. Fig. 9 shows the smoothing results, where (a) is the input surface model. (b)–(f) shows the results for different $f$ and $h = 0$. It is easy to see from these figures that $f = 1$, $f = H^2$ and $f = \sqrt{k_1^2 + k_2^2}$ yields more desirable smoothing results.

For the blending problem, we are given a surface with some parts missing. We want to use different PDEs to reconstruct the surface. The aim is to observe the smoothness at the blending boundary. Fig. 10 shows the blending results, where (a) is the input surface model to be blended. (b) shows the initial blending surface construction. (c)–(h) show the results for different $f$ and $h = 0$. It could be observed that all the used fourth order flows yield smooth joining bending surfaces at the boundaries.

5.3. Running times

We summarize in Table 1 the computation time needed by our examples. The algorithm was implemented in C++ running on a Dell PC with a 3.0 GHz Intel CPU. The second column in Table 1 lists the number of unknowns. These numbers are counted as $3n_0$ (each vertex has $x,y,z$ variables). Here $n_0$ is the number of interior vertices. The third column is the temporal step-sizes used. The fourth column in the table is the time (in seconds) for forming the
Fig. 10. The input is four cylinders to be blended as shown in (a). The initial blending surface is shown in (b). We evolve the initial surface using the geometric flows with different Lagrange functions $f(H,K)$ and $h(x,n) = 0$. Figures (c)–(h) are the evolution results of the corresponding flows, where figure (c) is the result after 135 iterations with temporal step-size 0.01 and figures (d)–(h) show the results after 1000 iterations with temporal step-size 0.00008.

Table 1

<table>
<thead>
<tr>
<th>Examples</th>
<th># unknowns</th>
<th>$\tau^{(k)}$</th>
<th>Form matrix</th>
<th># steps</th>
<th>Solving Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 2(b)</td>
<td>3600</td>
<td>0.00001</td>
<td>0.01</td>
<td>1000</td>
<td>0.14(22)</td>
</tr>
<tr>
<td>Fig. 3(b)</td>
<td>2520</td>
<td>0.000045</td>
<td>0.01</td>
<td>500</td>
<td>0.11(20)</td>
</tr>
<tr>
<td>Fig. 4(b)</td>
<td>9120</td>
<td>0.00002</td>
<td>0.02</td>
<td>1000</td>
<td>0.34(19)</td>
</tr>
<tr>
<td>Fig. 5(b)</td>
<td>12294</td>
<td>0.00001</td>
<td>0.05</td>
<td>1000</td>
<td>0.42(20)</td>
</tr>
<tr>
<td>Fig. 6(a)</td>
<td>9600</td>
<td>0.01</td>
<td>0.04</td>
<td>200</td>
<td>0.40(20)</td>
</tr>
<tr>
<td>Fig. 7(b)</td>
<td>6528</td>
<td>0.01</td>
<td>0.02</td>
<td>400</td>
<td>0.22(20)</td>
</tr>
<tr>
<td>Fig. 8(b)</td>
<td>9600</td>
<td>0.00001</td>
<td>0.03</td>
<td>100</td>
<td>0.36(20)</td>
</tr>
<tr>
<td>Fig. 9(c)</td>
<td>89721</td>
<td>0.0000001</td>
<td>0.49</td>
<td>5</td>
<td>5.36(26)</td>
</tr>
<tr>
<td>Fig. 10(d)</td>
<td>4410</td>
<td>0.000008</td>
<td>0.02</td>
<td>1000</td>
<td>0.15(20)</td>
</tr>
</tbody>
</table>

The coefficient matrix (one time step). The fifth column is the number of evolution steps. The last column is the total time for solving the linear systems. The linear systems are solved by GMRES iterative method. The threshold value of controlling the iteration-stopping is taken to be $10^{-5}$, about the single word-length accuracy. Note that the coefficient matrix of the resulting system is a unit one, if the temporal step-size is zero. Hence smaller temporal step-size leads to fewer number of iterations in using GMRES method. To make the running times comparable for different size problems, we need to select the temporal step-size so that the required numbers of GMRES iterations are about the same (around 20). The numbers are presented in ( ) of the sixth column. In Table 1, we present only one case for each of the nine figures. This is because the coefficient matrices of the linear systems are computed in the same way for the same surface model and different functions $f$ or $h$, and the matrices have the same size and the same sparsity. But the iteration numbers of using GMRES method may be different.

6. Conclusions

We have presented a general framework for surface modeling using a general form geometric PDE. Starting with a general form integral $\int f(H,K) + \int h(x,n)$, we obtain the Euler–Lagrange equation and the geometric PDE.
geometric PDE is general, which includes several known geometric PDEs as its special cases, but is discretized under a uniform framework. The proposed approach is easy to implement, requiring only the discretizations of LBO and operator in the form of (4.1) and evaluating subroutines for computing $f$, $fH$, $fK$, $h$, $\nabla x h$, $\nabla n h$, $\nabla_n h$ and $\nabla^2 h$. The proposed algorithm can be used to simulate the behaviors of various geometric PDEs, such as mean curvature flow, weighted mean curvature flow, Willmore flow, biologic vesicle equations and the equations in chemical and physical settings. The implementation shows that the proposed approach is efficient and works well.

Appendix A. Sketch proofs to theorems in Section 2

To prove the theorems in Section 2, the following lemma is required. This lemma can be proved easily using the definitions of the related differential operators and the product rule for differentiation.

Lemma 6.1. For any function $f \in C^2(\mathcal{M})$ and three-dimensional vector-valued functions $f$, $h \in C^2(\mathcal{M})^3$, we have

\[
\text{div}(\nabla f) = (\Delta f, h) + \nabla f : \nabla h,
\]

(A.1)

\[
\text{div}(f \nabla h) = f \Delta h + [\nabla h]^T \nabla f.
\]

(A.2)

\[
\Delta(f h) = \Delta f h + 2[\nabla h]^T \nabla f + f \Delta h,
\]

(A.3)

\[
\text{div}(\varnothing(f h)) = (\varnothing f, h) + \nabla f : \varnothing h = (\varnothing f, h) + \nabla h : \varnothing f,
\]

(A.4)

\[
\text{div}(f \varnothing h) = f \varnothing h + [\varnothing h]^T \nabla f = f \varnothing h + [\nabla h]^T \varnothing f.
\]

(A.5)

\[
\varnothing(f h) = f \varnothing h + 2[\varnothing h]^T \nabla f + f \varnothing h = f \varnothing h + 2[\nabla h]^T \varnothing f + f \varnothing h.
\]

(A.6)

Proof of Theorem 2.1. Proving (2.19) is equivalent to verifying

\[
(\Delta x, n) = 2H,
\]

(A.7)

\[
(\Delta x, x_u) = 0,
\]

(A.8)

\[
(\Delta x, x_v) = 0,
\]

(A.9)

since $\{x_u, x_v, n\}$ constitutes one frame of surface $\mathcal{M}$. In view of (A.1), we have

\[
(\Delta x, n) = \text{div}(\nabla x n) - \nabla : \nabla n = -\text{tr}[[x_u, x_v][g^{\alpha \beta}][-S][x_u, x_v]^T] = \text{tr}[S] = 2H,
\]

and

\[
\text{div}(x_u) = \text{div}(\nabla x u) = (\Delta x, x_u) + \nabla : \nabla x_u.
\]

Furthermore, we can calculate that

\[
\text{div}(x_u) = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial u} \frac{\partial}{\partial v} \right] \left[ \begin{array}{cc} g_{22} & -g_{12} \\ -g_{12} & g_{11} \end{array} \right] \frac{\partial}{\partial u} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u} \left[ \begin{array}{c} g_{u} \\ 2g \end{array} \right] = \frac{g_u}{2g},
\]

\[
\nabla : \nabla x_u = \text{tr}[[x_u, x_v][g^{\alpha \beta}][x_u, x_v]^T] = \text{tr}[[g^{\alpha \beta}][x_u, x_v]^T] = \frac{g_u}{2g} x_u.
\]

Thus (A.8) is proved. Similar proof to (A.9) can be carried out.

To prove (2.20), we notice that

\[
\nabla n = -[x_u, x_v][g^{\alpha \beta}][S][x_u, x_v]^T = -[x_u, x_v][(2H)g^{\alpha \beta} + Kb^{\alpha \beta}][x_u, x_v]^T = -2H \nabla x + \varnothing x,
\]

(A.10)

where the second equality is true because of the following equality

\[
2H[g^{\alpha \beta}] - [g^{\alpha \beta}]S - K[b^{\alpha \beta}] = 0.
\]

Therefore from (2.17),

\[
\Delta n = \text{div}(\nabla n) = \text{div}[-2H \nabla x + \varnothing x] = -2\nabla H - 2H \Delta x + \varnothing x,
\]

(A.11)
where the last equality is held because of (A.2), (2.18) and the symmetry property (2.3) of $\nabla x$. Thus the correctness of (2.20) yields the completion of the proof. □

Proof of Theorem 2.2. In analogy with the proof of Theorem 2.1, we can calculate, without too much difficulty, the following equalities

\begin{align*}
\langle \Box x, n \rangle &= -\nabla x : n + \text{tr} [K b^{\alpha \beta} g^{\alpha \beta}] = 2K,
\end{align*}

with the help of (A.4) and Mainardi–Codazzi equations (do Carmo, 1976, p. 235), whence (2.21) is valid. From (2.18), (2.5) and (A.5), we obtain

\begin{align*}
\Box n &= \text{div}[\Box n] = -\nabla K - K \Delta x.
\end{align*}

Furthermore, taking (2.19) and (2.21) into account, the last equality of (2.22) can be proved. Thus we complete this proof. □

Proof of Theorem 2.3. (2.23) can be proved with a straightforward calculation without any difficulty from the definition of operator div. (2.24) is (A.2) in Lemma 6.1. We can prove (2.25) by virtue of (A.4) and (A.10). □

References


