Discrete Surface Modeling using Geometric Flows

Guoliang Xu * Qing Pan
Academy of Mathematics and System Science,
Chinese Academy of Sciences, Beijing, China
Email: xuguo@lsec.cc.ac.cn

Chandrajit L. Bajaj †
Center for Computational Visualization and Institute for Computational Engineering & Sciences,
Department of Computer Science, University of Texas, Austin, TX 78712
Email: bajaj@cs.utexas.edu

August 17, 2003

Abstract

We use various nonlinear geometric partial differential equations to efficiently solve several surface modeling problems, including surface blending, N-sided hole filling and free-form surface fitting. The nonlinear equations used include two second order flows (mean curvature flow and average mean curvature flow), one fourth order flow (surface diffusion flow) and a sixth order flow. These nonlinear equations are discretized based on discrete differential geometry operators. The proposed approach is simple, efficient and gives very desirable results, for a range of surface models, possibly having sharp creases and corners.

Key words: Geometric PDE; Discrete Surface; Surface Blending; N-sided hole filling; Free-form Surface Design.

1 Introduction

We use various curvature driven geometric partial differential equations (GPDE) to solve several surface modeling problems. The GPDEs we use include mean curvature flow, averaged mean curvature flow, surface diffusion flow and even higher order flows. All these equations are nonlinear and the geometry intrinsic, i.e., they do not depend upon any particular parameterization. The problems we solve include surface blending, N-sided hole filling and free-form surface fitting with high boundary continuity.

For the problems of surface blending and N-sided hole filling, we are given triangular surface meshes of the surrounding area. Triangular surface patches need to be constructed to fill the openings enclosed by the surrounding surface mesh and interpolate its boundary with some specified order of continuity. For the free-form surface fitting problem, we are possibly given a set of points, or a wire frame of curves that defines an outline of the desired shape, or even some surface patches. We construct a surface which interpolates the points or curves or the boundaries of the patches with specified order of continuities. The free-form surface

*Support in part by NSFC grant 10241004, National Innovation Fund 1770900, Chinese Academy of Sciences.
†Supported in part by NSF grants ACI-9982297 and CCR 9988357.

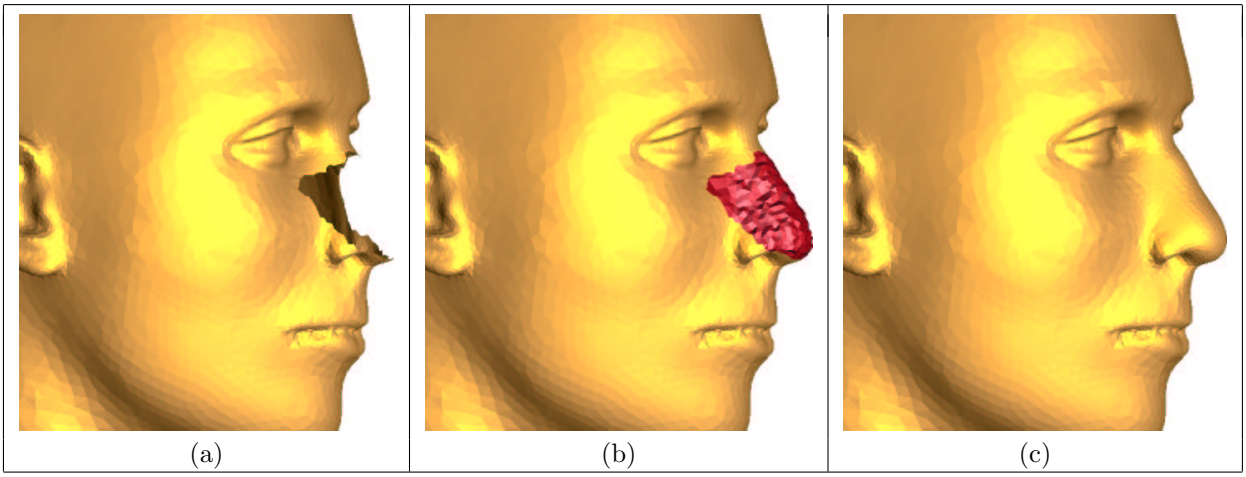


Fig 1.1: (a) shows a head mesh with a hole around the nose. (b) shows an initial filler construction of the brandy nose. (c) the faired filler surface, after 1 iteration, generated using surface diffusion flow. The time step length is chosen to be 0.0001.

fitting problem is the most general and it includes the surface blending and N-sided hole filling problems, as its special cases.

Our twofold strategy for solving these problems is as follows: First we construct an initial triangular surface mesh ("filler") using any of a number of automatic or semi-automatic free-form modeling techniques (see [1, 2, 3, 25, 42, 60]). One may also interactively edit this "filler" to meet the weak assumptions for an initial solution shape. This "filler" may be bumpy or noisy, and in general this "filler" does not satisfy the smoothness boundary conditions, though it may roughly characterize the shape of the surface to be constructed. Second we deform the initial mesh by solving a suitable geometric flow GPDE. Unlike most of the previous free-form modeling techniques, our approach solves high-order boundary continuity constraints without any prior estimation of normals or derivative jets along the boundary. The solution of the GPDE is time dependent. We consider two possibilities for the time span of the evolution. One is a short time evolution, where we require the solution to respect to the initial shape or geometry (see Fig. 1.1). The other is a long time evolution, where the initial filler provides a topological structure, and what we look for is a stable solution state of the geometric flow (see Fig. 4.1, and 4.4). In this paper, we focus our attention on these twofold solutions of GPDE with boundary continuity constraints, rather than the construction of initial filler mesh. In section 3.4, we present automatic approaches for constructing the initial filler mesh, and our preferred choice.

Previous Work. Earlier research on using PDEs to handle surface modeling problems trace back to Bloor et al's works at the end of the 1980s ([9, 10, 11]). The basic idea of these papers are the use of biharmonic equations on a rectangular domain to solve the blending and hole filling problems. One of the advantages of using the biharmonic equation is that it is linear, and therefore easier to solve. However, the equation is not geometry intrinsic and the solution of the equation (the geometry of the surface) depends on the concrete parameterization used. Furthermore, these methods are inappropriate to model surfaces with arbitrary shaped boundaries.

The evolution technique, based on the heat equation $\partial_t x - \Delta x = 0$, has been extensively used in the area of image processing (see [41, 43, 52]. In [52], there are 453 relevant references listed), where Δ is a $2D$ Laplace operator. This was extended lately to smoothing or fairing noisy surfaces (see [16, 18, 19]). For a surface M,

the counterpart of the Laplacian Δ is the Laplace-Beltrami operator Δ_M (see [20]). One then obtains the geometric diffusion equation $\partial_t x - \Delta_M x = 0$ for a surface point $x(t)$ on the surface $M(t)$. Taubin [49] discussed the discretized operator of the Laplacian and related approaches in the context of generalized frequencies on meshes. Kobbelt [28] considered discrete approximations of the Laplacian in the construction of fair interpolatory subdivision schemes. This work was extended in [30] to arbitrary connectivity for purposes of multi-resolution interactive editing. Desbrun et al. [18] used an implicit discretization of geometric diffusion to obtain a strongly stable numerical smoothing scheme. The same strategy of discretization is also adopted and analyzed by Deckelnick and Dziuk [17] with the conclusion that this scheme is unconditionally stable. Clarenz et al. [16] introduced anisotropic geometric diffusion to enhance features while smoothing. Ohtake et al. [39] combined an inner fairness mechanism in their fairing process to increase the mesh regularity. Bajaj and Xu [5] smooth both surfaces and functions on surfaces, in a C^2 smooth function space defined by the limit of triangular subdivision surfaces (quartic Box splines). Similar to surface diffusion using the Laplacian, a more general class of PDE based methods called *flow surface techniques* have been developed which simulate different kinds of flows on surfaces (see [55] for references) using the equation $\partial_t x - v(x,t) = 0$, where $v(x,t)$ represents the instantaneous stationary velocity field.

Level set methods were also used in surface fairing and surface reconstruction (see [7], [8], [14], [38], [40], [56], [61], [62]). In these methods, surfaces are formulated as iso-surfaces (level surfaces) of 3D functions, which are usually defined from the signed distance over Cartesian grids of a volume. An evolution PDE on the volume governs the behavior of the level surface. These level-set methods have several attractive features including, ease of implementation, arbitrary topology (see [12]) and a growing body of theoretical results. Often, fine surface structures are not captured by level sets, although it is possible to use adaptive (see [6], [43]) and triangulated grids as well as Hermite data (see [27],[29]). To reduce the computationally complexity, Bertalmio et al [7, 8] solve the PDE in a narrow band for deforming vectorial functions on surfaces (with a fixed surface represented by the level surface).

Recently, surface diffusion flow has been used to solve the surface blending problem and free-form surface design problem ([46, 47]). In [46], fair meshes with G^1 conditions are created in the special case where the meshes are assumed to have subdivision connectivity. In this work, local surface parameterization is still used to estimate the surface curvatures. The later paper [47] use 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. In a web poster by Clarenz et al [15], a finite element method is used to solve the equation of Willmore flow for the aim of surface restoration.

Another category of surface fairing research is based on utilizing optimization techniques. In this category, one constructs an optimization problem that minimizes certain objective functions [25, 26, 37, 44, 53], such as thin plate energy, membrane energy [30], total curvature [31, 54], or sum of distances [33]. Using local interpolation or fitting, or replacing differential operators with divided difference operators, the optimization problems are discretized to arrive at finite dimensional linear or nonlinear systems. Approximate solutions are then obtained by solving the constructed systems. In general, such an approach is quite computational intensive.

Main Results. We use second order flows (mean curvature flow and averaged mean curvature flow) for G^0 continuity, fourth order flow (surface diffusion flow) for G^1 continuity and sixth order flow for G^2 continuity in each of several surface modeling problems. The proposed approach is simple and easy to implement. It is general, solves several surface modeling problems in the same manner, and gives very desirable results for a range of complicated free-form surface models, possibly having sharp and corners. Furthermore, it avoids estimating normals or tangents or curvatures on the boundaries.

The rest of the paper is organized as follows: Section 2 describes several nonlinear GPDEs used in this

paper. In section 3, we give details of the discretization and the numerical computation for the solution of the GPDEs. Examples to illustrate the different effects achievable from the solution of the GPDEs are given in section 4.

2 Partial Differential Equation Models

We consider several nonlinear GPDE models we use in this paper. More details on the existence and uniqueness of the solutions, the numerical computations of the solutions and evolution behaviors can be found in a series of papers by Mayer, Simonett, Escher [23, 24, 34, 35, 48]. Suppose $M(t)$ is a closed surface. Let M_0 be a compact closed immersed orientable surface in $\mathbb{R}^3$. A curvature driven geometric evolution consists of finding a family $M = \{M(t) : t \geq 0\}$ of smooth closed immersed orientable surfaces in $\mathbb{R}^3$ which evolve according to the flow equation

$$\frac{\partial x}{\partial t} = V_n(k_1, k_2, x)N(x), \quad M(0) = M_0. \tag{2.1}$$

Here $x(t)$ is a surface point on $M(t)$, $V_n(k_1, k_2, x)$ denotes the normal velocity of $M(t)$, which depends on the principal curvatures k_1, k_2 of $M(t)$, $N(x)$ stands for the unit normal of the surface at $x(t)$. Let $A(t)$ denote the area of $M(t)$, $V(t)$ denote the volume of the region enclosed by $M(t)$. Then it has been shown that (see [58], Theorem 4)

$$\frac{dA(t)}{dt} = \int_{M(t)} V_n H d\sigma, \quad \frac{dV(t)}{dt} = \int_{M(t)} V_n d\sigma, \tag{2.2}$$

where $H = \frac{1}{2}(k_1 + k_2)$ is the mean curvature of $M(t)$.

1. Mean Curvature Flow (see [21, 57])

If we take $V_n = -H = -\frac{1}{2}(k_1 + k_2)$, then one obtains the mean curvature flow GPDE:

$$\frac{\partial x}{\partial t} = -HN(x), \quad M(0) = M_0. \tag{2.3}$$

It follows from (2.2) that

$$\frac{dA(t)}{dt} = -\int_{M(t)} H^2 d\sigma, \quad \frac{dV(t)}{dt} = -\int_{M(t)} H d\sigma. \tag{2.4}$$

(2.4) implies that the mean curvature flow is area shrinking. Since the total squared curvature $\int H^2 d\sigma \geq 4\pi$ (see [13], page 182), the surface will shrink rapidly to zero.

2. Average Mean Curvature Flow (see [24, 45])

In (2.1), if we take $V_n = h(t) - H(t)$, where $h(t) = \int_{M(t)} H d\sigma / \int_{M(t)} d\sigma$, then we have the average mean curvature flow GPDE:

$$\frac{\partial x}{\partial t} = [h(t) - H(x)]N(x), \quad M(0) = M_0. \tag{2.5}$$

The existence proof of the global solutions to this flow can be found in Escher and Simonett's paper [24]. It

follows from (2.2) that

$$\begin{aligned}\frac{dA(t)}{dt} &= \int_{M(t)}(hH - H^2)d\sigma \\ &= \int_{M(t)}[hH - H^2 - h(h-H)]d\sigma \\ &= -\int_{M(t)}(h-H)^2 d\sigma \le 0, \end{aligned} \tag{2.6}$$

since obviously $\int_{M(t)} h(h-H) = h(h\int_{M(t)} d\sigma - \int_{M(t)} Hd\sigma) = 0$. On the other hand, the second equation of (2.2) implies that

$$\frac{dV(t)}{dt} = h(t)\int_{M(t)} d\sigma - \int_{M(t)} Hd\sigma = 0.$$

Hence the averaged mean curvature flow is volume preserving and area shrinking. The area shrinking stops if $H \equiv h$.

3. Surface Diffusion Flow (see [34])

If we take $V_n = \Delta H$, we get the so-called surface diffusion flow GPDE:

$$\frac{\partial x}{\partial t} = (\Delta H)N(x), \quad M(0) = M_0, \tag{2.7}$$

where $\Delta := \Delta_M$ is Laplace-Beltrami operator which acts on functions defined on surface $M(t)$. The existence and uniqueness of solutions for this flow is given in [23]. It is also shown in this paper that the solution converges exponentially fast to a sphere if the initial surface is embedded and close to a sphere. From (2.2) we have

$$\begin{aligned}\frac{d}{dt}A(t) &= \int_{M(t)} \Delta H H d\sigma = -\int_{M(t)} |\nabla H|^2 d\sigma \le 0, \\ \frac{d}{dt}V(t) &= \int \mathrm{div}(\nabla H)d\sigma = -\int \nabla H \nabla(1) d\sigma = 0,\end{aligned}$$

where ∇ stands for the gradient operator. Hence, the surface diffusion flow is area shrinking, but volume preserving. The area stops shrinking when the gradient of H is zero. That is, H is a surface with constant mean curvature. We have also implemented another fourth order flow–the Willmore flow, for which $V_n = \Delta H + 2H(H^2 - K)$, the outcome of this flow is very similar to that of surface diffusion flow. Hence, we do not detail the Willmore flow in the current paper.

4. Higher order Flow GPDE

$$\frac{\partial x}{\partial t} = (-1)^{k+1}\Delta^k H N(x), \quad M(0) = M_0. \tag{2.8}$$

It follows from the second equation of (2.2) that this flow is volume preserving if $k \ge 2$. However we do not know if this flow is area shrinking.

The area/volume preserving/shrinking properties for the flows mentioned above are for closed surfaces. In our application of these flows, the surfaces always have fixed boundaries.

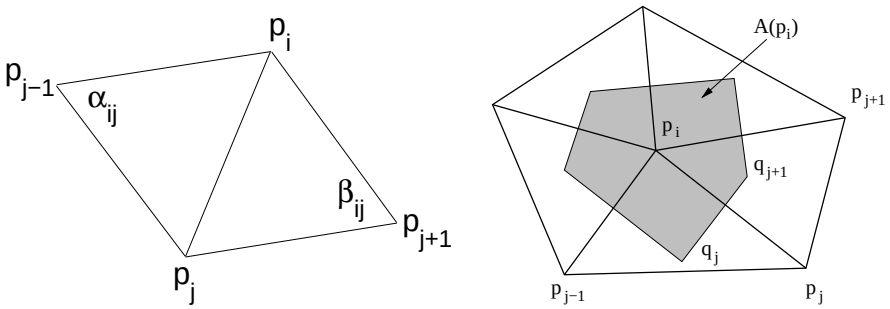


Fig 3.1: Left: The definition of the angles α_{ij} and β_{ij}. Right: The definition of the area $A(p_i)$.

3 Solution of the GPDEs

There are basically two classes of approaches for solving a GPDE on any domain. One approach is based on finite divided differences, the other is based on finite elements (see [5, 17, 21]). The approach we adopt in this paper is based on finite divided differences. Since we are dealing with differential equations over 2-manifolds in $\mathbb{R}^3$, the classical finite divided differences will be replaced by discretized differential geometric operators over surfaces. Section 3.1 deals with discretized geometric operators and shows how the spatial discretization of the GPDE is performed. Next in Section 3.2 we detail how the boundary conditions are respected. Semi-implicit discretization in the time domain is considered in section 3.3. Other issues, such as mesh regularization, initial mesh construction and computational complexity, are addressed in section 3.4.

3.1 Discretized Laplace-Beltrami Operator

One of the fundamental problems in solving GPDEs is the discretization of the Laplace-Beltrami operator. On a triangular surface mesh, several discretized approximations of the operator have been proposed (see [18, 35, 50, 51]). In this paper we adopt the discretization developed by Meyer et al in ([36]). A comparative research about the various discretized Laplace-Beltrami operators is conducted in [59]. It has been shown that Meyer et al's is better for discretizing our GPDEs. Let f be a smooth function on a surface, then Δf is approximated over a triangular mesh by

$$\Delta f(p_i) = \frac{1}{A(p_i)} \sum_{j \in N_1(i)} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} [f(p_j) - f(p_i)], \tag{3.1}$$

where $N_1(i)$ is the index set of 1-ring of neighbor vertices of vertex p_i, α_{ij} and β_{ij} are the triangle angles shown in Fig 3.1 (Left). $A(p_i)$ is the area for vertex p_i as shown in Fig 3.1 (Right), where q_j is the circumcenter point for the triangle $[p_{j-1}p_jp_i]$ if the triangle is non-obtuse. If the triangle is obtuse, q_j is chosen to be the midpoint of the edge opposite to the obtuse angle. Since $\Delta x = -2HN$ (see [58], page 151), we have

$$H(p_i)N(p_i) = \frac{1}{2A(p_i)} \sum_{j \in N_1(i)} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} (p_i - p_j). \tag{3.2}$$

This is an approximation of the mean curvature normal given in [36]. The higher order Laplace-Beltrami operators are discretized recursively as

$$\Delta^k f(p_i) = \Delta(\Delta^{k-1} f)(p_i) = \frac{1}{A(p_i)} \sum_{j \in N_1(i)} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} [\Delta^{k-1} f(p_j) - \Delta^{k-1} f(p_i)] \tag{3.3}$$

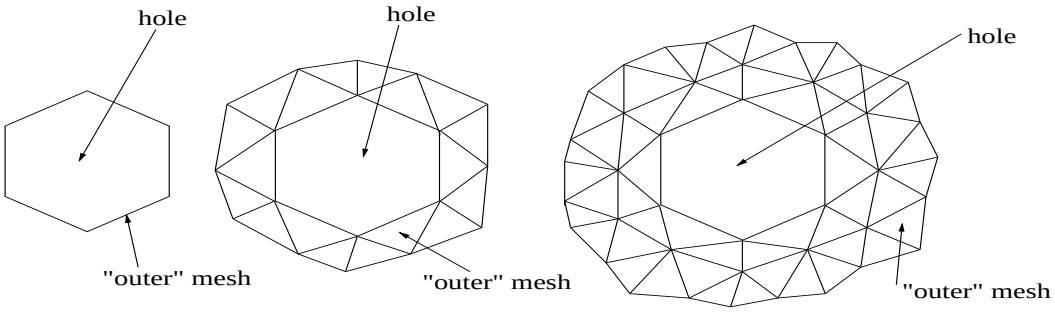


Fig 3.2: Left: The involved vertices of the"outer" mesh for a G^0 boundary condition. The "outer" mesh is just the boundary of the hole. Middle: The involved vertices of the "outer" mesh for a G^1 boundary condition. Right: The involved vertices of the "outer" mesh for a G^2 boundary condition.

with $\Delta^0 f(p_i) = f(p_i)$. Note that $\Delta^k f(p_i)$ involves function values on a k-ring of neighboring vertices of p_i.

Let $F = [f(p_1), \cdots, f(p_m)]^T$, $\Delta F = [\Delta f(p_1), \cdots, \Delta f(p_m)]^T$, where $p_1, \cdots, p_m$ are all the unknown vertices to be determined in each of our modeling problems. Then (3.1) could be written in matrix form:

$$\Delta F = -\mathcal{D}\mathcal{M}F + B \tag{3.4}$$

where $\mathcal{D} = \text{diag}[\frac{1}{2A(p_1)}, \cdots, \frac{1}{2A(p_m)}]$ is a diagonal matrix, $\mathcal{M} = \{m_{ij}\}_{i,j=1}^m$ with

$$m_{ij} = \begin{cases} \sum_{k\in N_1(i)} \cot\alpha_{ik} + \cot\beta_{ik}, & i = j, \\ -(\cot\alpha_{ij} + \cot\beta_{ij}), & i \neq j, \quad i \in N_1(j), \quad j \in N_1(i), \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, $\mathcal{M}$ is a sparse, symmetric and positive definite matrix (see [47]). The constant term B is obtained from the boundary conditions. It follows from (3.4) that

$$\Delta^2 F = \mathcal{D}\mathcal{M}\mathcal{D}\mathcal{M}F - \mathcal{D}\mathcal{M}B + B \tag{3.5}$$

Again, $\mathcal{M}\mathcal{D}\mathcal{M}$ is a sparse, symmetric and positive definite matrix. Furthermore, the coefficient matrix for $-\mathcal{D}^{-1}\Delta^3 F$ is also sparse, symmetric and positive definite.

3.2 Handling of Boundary Conditions

1). Natural Boundary Conditions for Blending and Hole Filling

By the natural boundary conditions, we mean that no continuity conditions are specified at the boundary points, but the continuity is implied by the "outer" mesh incident to the boundary of the hole (see Fig 3.2). Such a treatment for boundary condition is suitable for both the blending problem and the N-sided hole filling problem, since the "outer" mesh always exists in such problems.

Let g_i be the order of continuity at a boundary point p_i, $g = \max g_i$. Then we can use the order $2g$ flow $\frac{\partial x}{\partial t} = (-1)^{g+1}\Delta^g H N(x)$ for constructing the triangular surface patch with G^{g_i} continuity at the boundary vertex p_i. $\Delta^g H$ is discretized recursively: $\Delta^g H = \Delta(\Delta^{g-1} H)$. At a boundary vertex p_i, $\Delta^k H(p_i)$ is evaluated according to the following rule:

Evaluation Rule at Boundary. *$\Delta^k H(p_i)$ is evaluated recursively by formulas 3.2 and 3.3 if $k \leq g_i$, otherwise $\Delta^k H(p_i)$ is set to zero and the recursion stops.*

Note that even for an inner vertex p_j, the recursive definition may make $\Delta^k H(p_j)$ involve the evaluation a lower order Laplace-Beltrami operator on the boundary. In general, the recursive evaluation of $\Delta^k H(p_i)$ at p_i (for either p_i being an inner or an outer vertex) involves k-ring neighbor vertices of p_i. Some of them may be inner vertices, and the remaining are outer vertices. The inner vertices are treated as unknowns in the discretized equations and the outers are incorporated into the left-handed side.

2). Natural Boundary Conditions for Free-Form Surface Filling

In the free-form surface filling problem, suppose we are given a wireframe of curves (edges) and we wish to flesh the wireframe with surface patches that contain the curves as boundary with pre-specified order of continuity. At each of the intersecting points of the patches, an order of continuity is pre-specified and the evaluation rule mentioned above is applied. For each inner point, a discretized linear equation is generated using the operator discretization (3.3). These linear equations for different patches are collected together and solved simultaneously. Note that one linear equation may involve inner vertices of several patches. However, if the continuity order at each boundary point is zero, any equation corresponding to an inner vertex does not involve inner vertices of other patches.

3.3 Time Direction Discretization

Given a time step-length $\tau > 0$, suppose we have an approximate solution $\{p_i^{(n)}\}$ of the order $2k$ GPDE at $t = n\tau$ for all the inner vertices. Then we construct an approximate solution $\{p_i^{(n+1)}\}$ for the next time step $t = (n+1)\tau$ by using a semi-implicit Euler scheme. That is, we replace the derivative $\frac{\partial x}{\partial t}$ by $[x(n\tau+\tau) - x(n\tau)]/\tau$, and the quantities α_{ij}, β_{ij} and $A(p_i)$ in (3.2) are computed using the previous result at $t = n\tau$. Such a treatment yields a linear system of equations with the inner vertices as unknowns. The coefficient matrix of the system can be written as

$$I + (\mathcal{D}\mathcal{M})^{k+1} = \mathcal{D}(\mathcal{D}^{-1} + \mathcal{M}(\mathcal{D}\mathcal{M})^k) = \mathcal{D}M$$

where $M = \mathcal{D}^{-1} + \mathcal{M}(\mathcal{D}\mathcal{M})^k$ is a highly sparse, symmetric and positive definite matrix, and hence an iterative method for solving such a linear system is desirable. We use a conjugate gradient iterative method with diagonal preconditioning. This iteration could be accelerated as usual using multi-grid techniques based on a hierarchical mesh representation (see [32]).

3.4 Other Important Issues

1. Mesh Regularization

The GPDEs, also called geometric evolution equations described in section 2 move the surface in the normal direction. For a discrete surface, this normal direction motion may cause very irregular (nonuniform) distribution of the mesh vertices. Hence, introducing a regularization mechanism in the evolution process is necessary. Since *the tangential displacement does not influence the geometry of the deformation, just its parameterization* (see [22]), we also add a tangential displacement to the normal direction motion. Hence, the general form of our geometric evolution problem could be written as

$$\frac{\partial x}{\partial t} = V_n(k_1, k_2, x)N(x) + V_t(x)T(x), \quad M(0) = M_0, \tag{3.6}$$

where $T(x)$ is a tangent direction at the surface point x, $V_t(x)$ is the tangential velocity. In the process of numerical solution of equation (3.6), $V_t(x)T(x)$ is chosen as

$$\mathcal{U}_0(p_i^{(n)}) - \left(\mathcal{U}_0(p_i^{(n)}), N(p_i^{(n)})\right) N(p_i^{(n)}) \tag{3.7}$$

where $\mathcal{U}_0(p_i^{(n)}) = \frac{1}{card(N_1(i))} \sum_{j \in N_1(i)} (p_j^{(n)} - p_i^{(n)})$, N is the surface normal defined by (3.2). This discretization of $V_t(x)T(x)$ is very similar to the one given by Ohtake et al. [39], which is $\mathcal{U}_0(p_i^{(n)}) - \Big(\mathcal{U}_0(p_i^{(n)}), N(p_i^{(n)})\Big)\, \mathcal{U}_0(p_i^{(n)})$. The difference is that our displacement is in the tangent plane. In (3.7), $\mathcal{U}_0(p_i^{(n)})$ could be replaced by $\mathcal{U}_0(p_i^{(n+1)})$ to use as many of the new values as possible, and still yield a linear system. However, such a treatment destroys the symmetric property of the coefficient matrix.

2). Stopping Criteria

For a given time step-length τ, we need to determine the minimal iteration number n, so that the evolution procedure stops at $n\tau$. The following two criteria are used

$$\|M(n\tau) - M(0)\| \geq \epsilon_1, \tag{3.8}$$

$$\|M(n\tau + \tau) - M(n\tau)\| / \tau \leq \epsilon_2, \tag{3.9}$$

where ϵ_1 and ϵ_2 are given control constants. Criterion (3.8) is for short time evolution, where we require $M(n\tau)$ near $M(0)$. Criterion (3.9) is for long time evolution, where we are looking for a stable status of the solution.

3). Computational Complexity

The computational cost arises from two parts. One is in constructing the coefficient matrix of the linear system via the discretization of the GPDE. The other cost arises in solving the linear system. Let m denote the number of unknowns (only the vertices of the mesh being evolved), and l the average number of nonzero elements of each row of the coefficient matrix. Now for the second, fourth and sixth order flows, the number of nonzero elements l = 7, 19 and 37 approximately. The average costs for forming the system is therefore $O(m)$. The costs of solving the linear system of course depend on the solver used. For the iterative conjugate gradient solver we use, each of iterations requires about $6lm$, and is thus again $O(m)$. The number of iterations required depends on the relative residual error and the condition number of the coefficient matrix. For the examples given in this paper, the relative residual error bound is chosen to be $3 * 10^{-5}$. The required number of iterations varies in the range of 1-100. Table 4.1 gives the running times on a SGI work station Indigo2 for several examples.

4). Construction of Initial Surface Mesh

To provide an initial solution to the geometric evolution problem, we need to construct an initial triangular surface mesh ("filler") for each opening using any of a number of automatic or semi-automatic free-form surface construction techniques [1, 2, 3, 25, 42, 60]. One can also interactively edit this "filler" to meet the weak assumptions for an initial solution shape.

Since the opening to be filled could be topologically complicated, we solve the problem in two steps. In the first step we fit each opening by an implicit algebraic surface or spline which interpolates or approximate the boundary data [3, 4, 10, 42]. The approach we used is the one developed by Bajaj et al [1, 2, 3, 4]. In this approach, the data to be interpolated or approximated could be points or curves (even with normals). For ours, the boundary data are always points. Of course, this approach may not guarantee to produce topologically correct surfaces. If this happens, we break the opening into several parts by inserting a few curves (polygons) and then repeat the surface fitting for each part until we achieve a reasonable shape for the "filler".

After the algebraic surface is obtained, a triangulation step is employed. Since this triangulation should be consistent with the boundary polygon of the opening, we adopted the expansion technique developed in [4]. Using this approach, we triangulate the surfaces starting from the boundary of the opening.

4 Examples

In this section, we give several examples to show how the varied GPDEs are used to solve different problems in a uniform fashion.

1). Comparison of the Flows

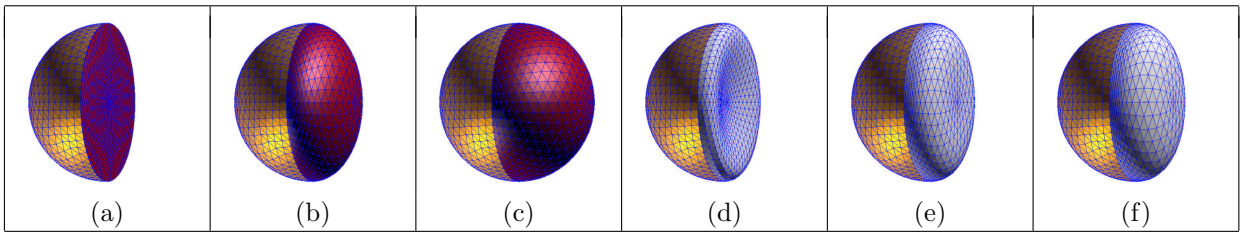
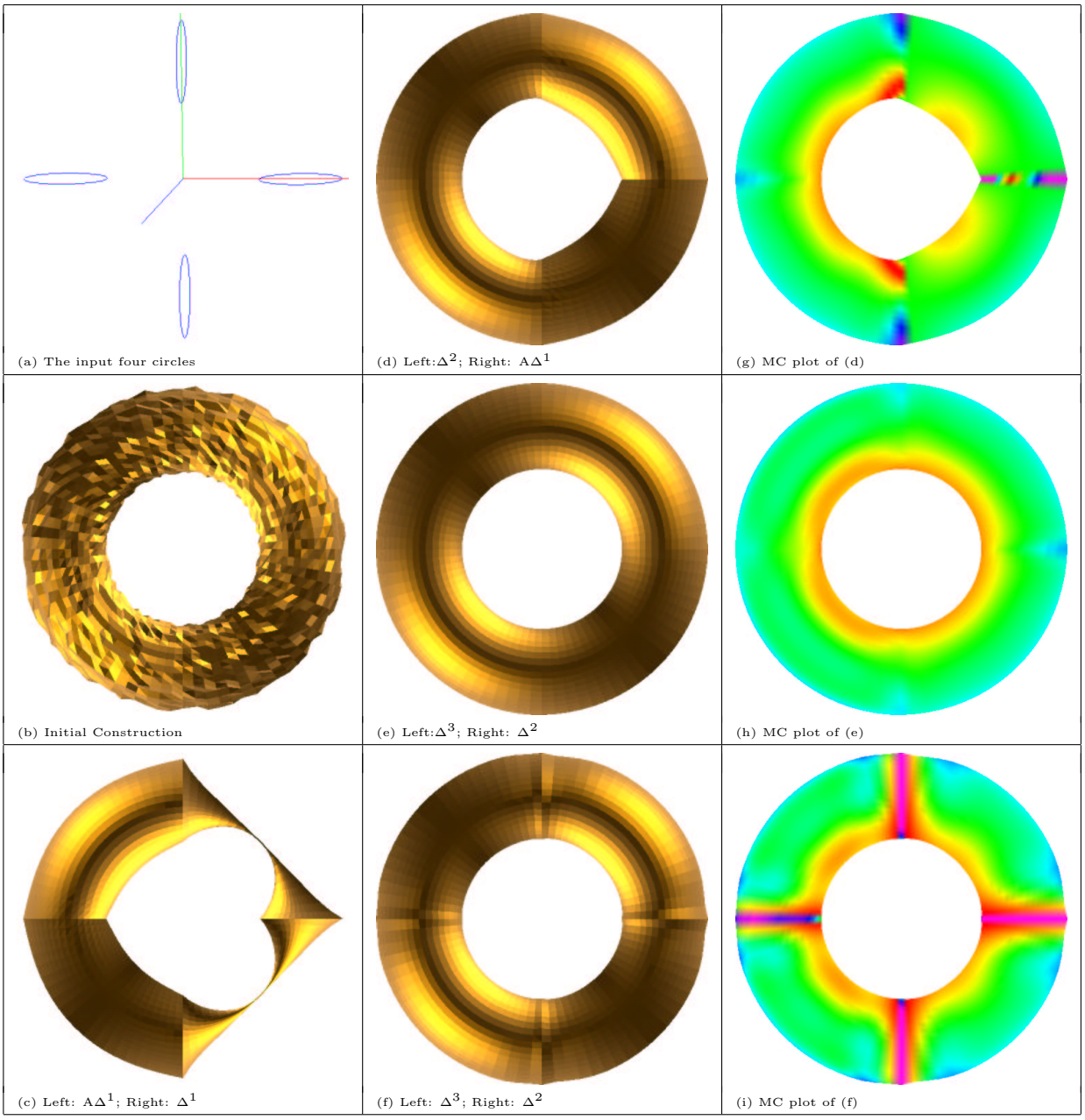


Fig 4.1: (a) The input semi-sphere (left part) with an initial planar triangulation of the disk opening. The mean curvature flow does not change the disk (initial mesh). (b) The result of surface diffusion flow after 10 iteration with $\tau = 0.1$. (c) The result of the sixth order flow after 10 iteration with $\tau = 0.01$. (d), (e) and (f) show three intermediate results of the sixth order flow with $\tau = 0.001$, and 1, 6 and 10 iterations, respectively.

We have pointed that the flows (GPDEs) we use have area/volume preserving properties for a closed surface. However, for an open surface with fixed boundary, the area/volume preserving properties are not guaranteed. The first three figures of Fig. 4.1 show the long time evolution solutions of the mean curvature flow, surface diffusion flow, and the sixth order flow for the input semi-sphere with an initial construction of the opening, a triangulated disk. The mean curvature flow does not change the disk. Figures (b) and (c) are the results after 10 iterations with $\tau = 0.1$ and $\tau = 0.001$, respectively. Further iterations do not have a significant change on the shape of the solution surface. The fourth and sixth order flows yield convex surfaces and the smoothness is clearly observed. Also notice that the sixth order flow recovers the sphere accurately. The last three figures show three intermediate results of the sixth order flow.

Fig. 4.2 shows the combined use of different flows. The aim of this toy example is to illustrate the difference of these flows, especially the continuity on the patch boundaries. Figure (a) shows four circles to be interpolated. Two of the circles are in the xz-plane, the other two are in the yz-plane. (b) shows an initial G^0 surface mesh constructed using [1] with some additional noise added. (c), (d), (e) and (f) are the faired interpolating surfaces after 6 iterations using different combinations of the flows. The step-lengths for the second, fourth and sixth order flows are chosen to be 0.1, 0.0025, and 0.0000625, respectively. Since the higher order flows evolve faster than the lower order flows, we use smaller step-lengths for higher order flows to obtain nearly the same surface evolution speed. Each of the meshes consists of four surface patches. The left two patches are in the regions $R^{-+} := \{(x, y, z) : x \leq 0, y \geq 0\}$ and $R^{--} := \{(x, y, z) : x \leq 0, y \leq 0\}$, respectively, and generated by one type of flow. The right two patches are in the regions $R^{++} := \{(x, y, z) : x \geq 0, y \geq 0\}$ and $R^{+-} := \{(x, y, z) : x \geq 0, y \leq 0\}$, respectively, and generated by a different flow. Figures (g), (h), (i) are the mean curvature plots of figures (d), (e), (f), respectively. The mean curvature at each vertex is computed by (3.2).

The aim of figure (c) is to show the difference between mean curvature flow and averaged mean curvature flow, where the left part is generated by the averaged mean curvature flow and the right part is produced by mean curvature flow. The mean curvature flow shrinks the surface very fast while the averaged mean curvature flow does not. Further evolution using the mean curvature flow will yield a pinch-off of the surface. Therefore, if we model a surface patch using second order flows with G^0 boundary condition, averaged mean curvature flow is more desirable than mean curvature flow.



(a) The input four circles

(d) Left:Δ^2; Right: $A\Delta^1$

(g) MC plot of (d)

(b) Initial Construction

(e) Left:Δ^3; Right: Δ^2

(h) MC plot of (e)

(c) Left: $A\Delta^1$; Right: Δ^1

(f) Left: Δ^3; Right: Δ^2

(i) MC plot of (f)

Fig 4.2: Comparison of different flows. Δ^k represents $2k$ order flow is used. $A\Delta^1$ denote the average mean curvature flow. The step lengths for the second, fourth and sixth order flows are chosen to be 0.1, 0.0025, and 0.0000625, respectively. Figures (c), (d), (e), (f) are the faired interpolating surface meshes after 6 iterations, where the continuities at the boundary curves are set to 0, 1, 2 and 0, respectively. Figures (g), (h), (i) are the mean curvature (MC) plots of figures (d), (e), (f), respectively.

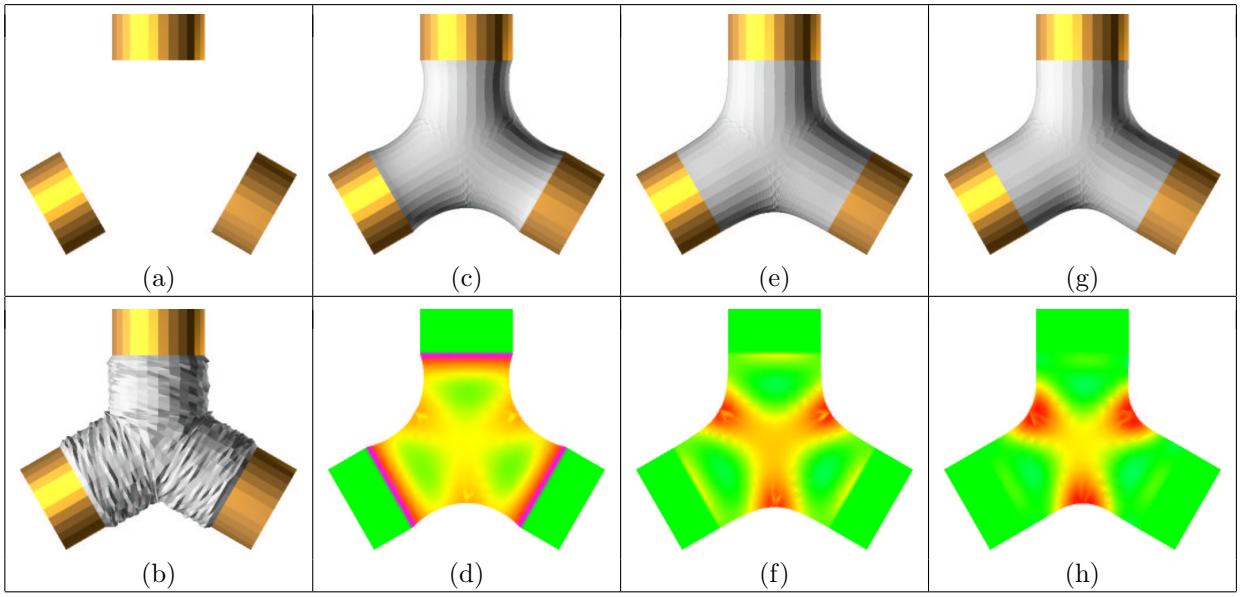


Fig 4.3: (a) shows three cylinders to be blended. (b) shows the initial construction. (c), (e) and (g) are the faired blending meshes generated using the flow (2.8) with $k = 0, 1, 2$, respectively. The time step lengths are chosen to be 0.01, 0.001, and 0.0001, respectively. These figures show the results after 32, 32 and 60 iterations. Figure (d), (f) and (h) show the mean curvature plots correspondingly.

Figure (d) is produced by surface diffusion (left part) and averaged mean curvature flow (right part). At the intersection curves of the right two patches, G^0 continuity is clearly observed. However, at the intersection curves of the left two patches, a smooth join property is exhibited. The smoothness at the intersection curves of the left and right patches is something in between.

The patches in R^{-+} and R^{--} of figure (e) are produced by a sixth order flow (2.8) (with $k = 2$), while the patches in R^{++} and R^{+-} are produced by a surface diffusion flow. As a whole, the surface looks smooth, our curvature plot reveal the smoothness difference at the intersection curves, the sixth order flow gives a smoother result than the fourth order flow.

Figure (f) is produced as (e), but the continuity order at the four circles are set to zero. Hence G^0 continuity is achieved there.

2. Surface Blending

Given a collection surface mesh with boundaries, we construct a fair surface to blend the meshes at the boundaries with specified geometric continuity. Fig 4.3 shows the case, where three cylinders to be blended are given (figure (a)) with an initial G^0 construction (figure (b)) using [1] with some additional noise added. The fair blending surfaces (figures (c), (e) and (g)) are generated using the flow (2.8) with $k = 0, 1, 2$, respectively. Figure (d), (f) and (h) show the mean curvature plot correspondingly. These figures clearly show the difference of smoothness achieved at blending boundaries. The mean curvature flow gives G^0 continuity results. The surface diffusion flow produces smooth surfaces at boundaries. The sixth order flow produces even smoother surfaces as expected.

3. N-sided Hole Filling

Given a surface mesh with a hole, we construct a fair surface to fill the hole with specified geometric

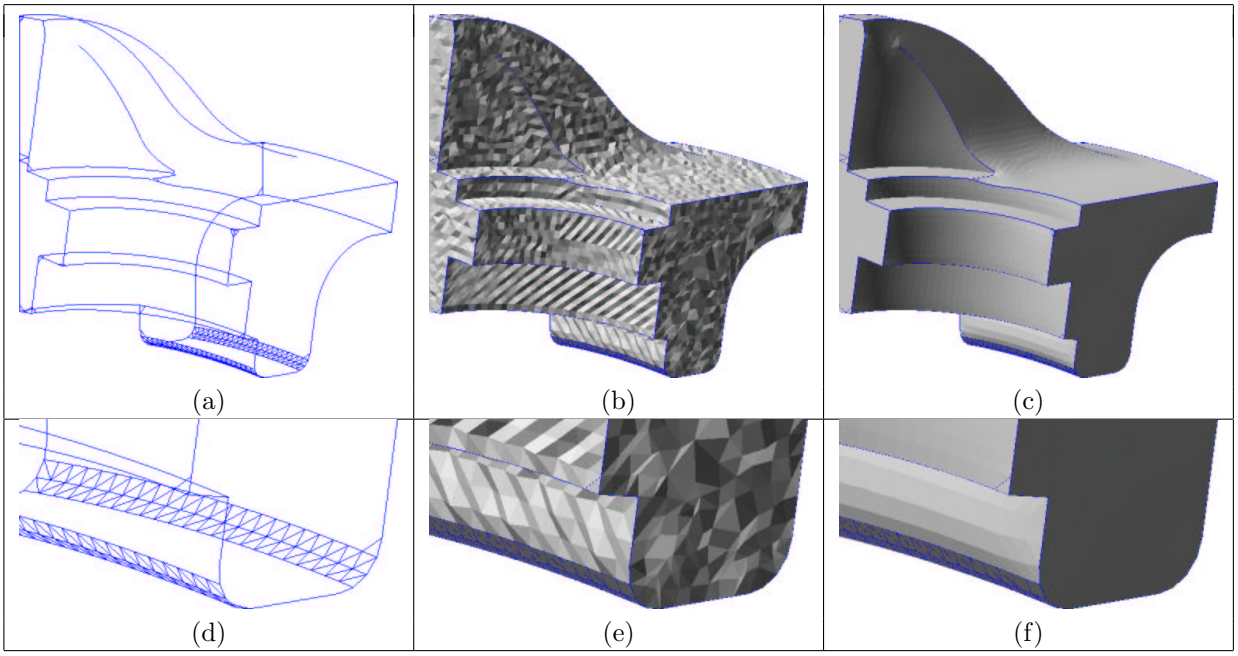


Fig 4.4: Interpolating curves and patches: (a) shows some input curves with G^0 continuity requirement and some bands of mesh with G^1 continuity requirement. (b) shows an initial construction of the surface mesh. (c) is the faired surfaces, after 12 iterations, generated using the the flow (2.8) with $k = 2$. The time step length is chosen to be 0.001. (d), (e) and (f) are the zoom in results of (a), (b) and (c), respectively.

continuity on the boundary. Fig 1.1 shows such an example, where a head mesh with a hole at the nose is given (figure (a)). An initial G^0 construction of the nose is shown in (b) using [1] with some noise added. The fair blending surface (figures (c)) are generated using the flow (2.8) with $k = 2$.

4. Free-Form Surface Construction

For the free-form surface design problem, we are given some curves, or partially patches, or points as inputs, and we wish to construct a fair surface mesh to interpolate this multi-dimensional data. Fig. 4.4 shows the approach of free-form surface construction, where some input curves with G^0 continuity requirement are given, and also given are some surface bands with a G^1 continuity requirement (see (a)). Figure (b) shows an initial construction of the G^0 surface mesh using the patch filling scheme [60] with added noise. (c) is the faired surfaces, after 12 iterations, generated using the the flow (2.8) with $k = 2$. The time step length is chosen to be 0.001. Figures (d), (e) and (f) are zoomed in views of (a), (b) and (c), respectively.

Fig. 4.5 shows the free-form design approach from an input triangular mesh, where (a) show the input surface triangular mesh with a G^1 continuity requirement (see (a)). Figure (b) shows an initial construction of the surface mesh. (c) and (d) are the faired meshes, after 2 iterations with $\tau = 0.01$, generated using the the mean curvature flow and averaged mean curvature flow, respectively. (e) is the faired mesh by surface diffusion flow, after 2 iterations with $\tau = 0.001$. (f) is the mean curvature plot of (e). The area shrinking of the mean curvature flow makes the input vertices to be interpolated become thorns (see (c)), while the area shrinking and the volume preserving of the averaged mean curvature flow make some of input vertices become thorns and some others become pits (see (d)). However, the fourth order flow does not suffer from this problem (see (e)). The obtained surface interpolates the input points and exhibits G^1 smoothness

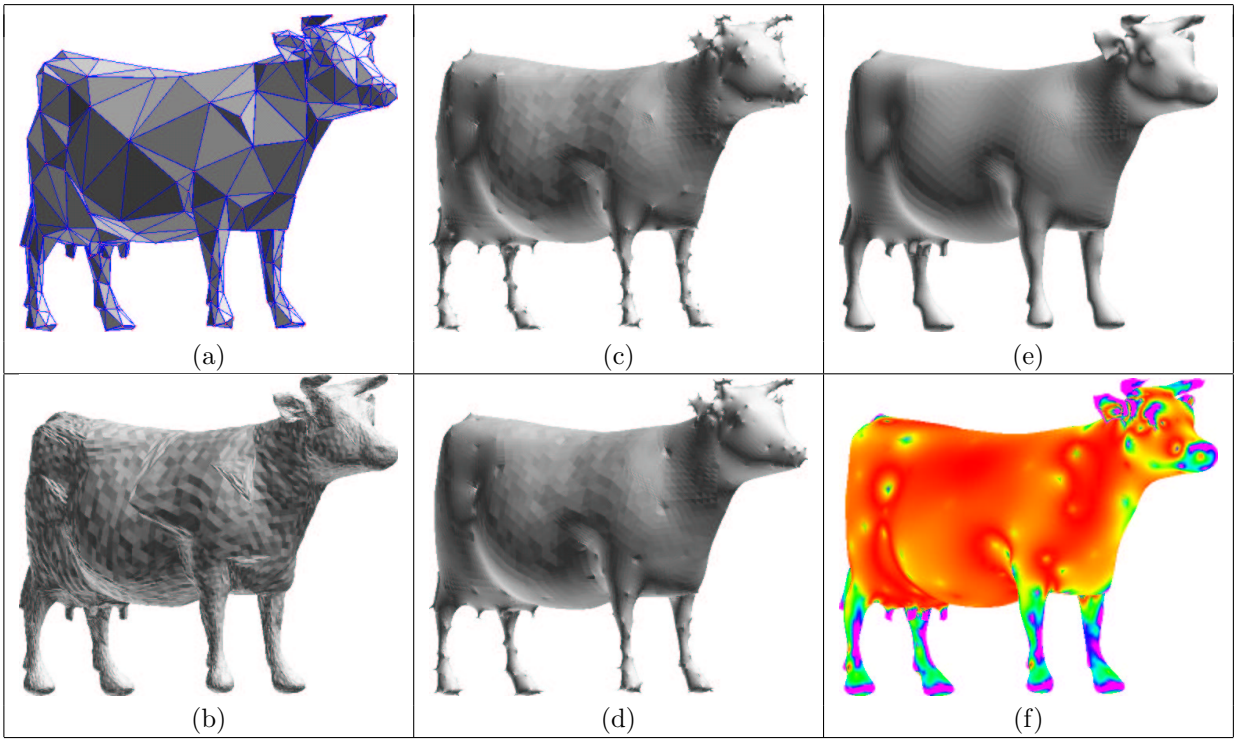


Fig 4.5: Interpolating points: (a) shows some input points and their triangulation. (b) shows an initial construction of the surface mesh. (c) and (d) are the faired surfaces, after 2 iterations with $\tau = 0.01$, using the mean curvature flow and averaged mean curvature flow, respectively. (e) is faired surfaces, after 2 iterations with $\tau = 0.001$, using the surface diffusion flow. (f) is the mean curvature plot of (e).

everywhere as well.

Finally, we summarize, in Table 4.1, the computation time needed by our examples. The algorithm was implemented in C++ running on a SGI graphics work-station Indigo2 with 16MHz. The fifth column in Table 4.1 is the time (in seconds) for forming the coefficient matrix (one time step). The sixth column is the number of iteration. The last column is the total time for solving the linear system. We separate the total time into two parts, because the cost for generating the matrix is fixed, while the time for solving the linear system depends greatly on the used solver.

5 Conclusions

We propose a simple scheme for using GPDEs to solve several surface modeling problems and with high boundary continuity conditions. The scheme has the following features: It produces very fair and desirable solution surfaces. The scheme is simple and very easy to implement. Specifically, it solves the free-form blending problem, the N-sided hole filling problem and free-form surface design problem in a uniform fashion. It solves the high boundary continuity problem in an easy and natural way and avoids prior estimation of normals or derivative jets on the boundaries. The implementation shows that the proposed approach works well for a wide range of surface models. Note that the C^1 or higher continuity interpolatory surface blending

Examples	Flow orders	Inner vertex	τ	Form matrix	Steps	Solving Time
Fig 1.1(c)	4	393	0.001	0.103s	1	0.85s
Fig 4.1(b)	4	481	0.1	0.098s	10	13.24s
Fig 4.1(c)	6	481	0.01	1.12s	10	124.63s
Fig 4.5(c)	2	23,625	0.01	0.64s	2	35.15s
Fig 4.5(d)	2	23,625	0.01	0.65s	2	31.70s
Fig 4.5(e)	4	23,625	0.001	4.45s	2	251.58s

Table 4.1: Firs column: examples. Second column: the order of the flows. Third column: number of inner vertices (unknowns). Fourth column: temporal step-length. Fifth column: time in seconds for computing the coefficient matrix. Sixth column: number of iteration steps. Last column: total times for solving the linear systems. The algorithm was implemented in C++ running on a SGI graphics work-station Indigo2 with 16MHz.

solution produced by e.g. [1, 42] for complicated corners, or holes with many boundary curve segments, are usually of very high algebraic degree and thereby prone to be with unsuitable for applications. The current solution of starting with G^0 low degree blends, coupled with higher order flow evolution, yields in general a much better alternative for very smooth surface solutions.

References

[1] C. Bajaj and I. Ihm. Algebraic surface design with hermite interpolation. *ACM Transactions on Graphics*, 19(1):61–91, 1992.

[2] C. Bajaj and I. Ihm. Smoothing Polyhedra using Implicit Algebraic Splines. *SIGGRAPH'92, Computer Graphics*, 26(2):79–88, 1992.

[3] C. Bajaj, I. Ihm, and J. Warren. Higher order interpolation and least squares approximation using implicit algebraic surfaces. *ACM Transaction on Graphics*, 12(4):327–347, 1993.

[4] C. Bajaj and G. Xu. Rational spline approximations of real algebraic curves and surfaces. In H.P. Dikshit and C. Michelli, editors, Approximations and Decomposition Series, Advances in Computational Mathematics, pages 73–85. World Scientific Publishing Co., 1994.

[5] C. Bajaj and G. Xu. Anisotropic Diffusion of Surface and Functions on Surfaces. *ACM Transaction on Graphics*, 22(1):4–32, 2003.

[6] E. Bansch and K. Mikula. Adaptivity in 3D Image Processing, *Manuscript*, 2001.

[7] M. Bertalmio, L. T. Cheng, and S. Osher. Variational problems and partial differential equations on implicit surfaces. CAM Report 00-23, UCLA, Mathematics Department, 2000.

[8] M. Bertalmio, G. Sapiro, L. T. Cheng, and S. Osher. A framework for solving surface partial differential equations for computer graphics applications. CAM Report 00-43, UCLA, Mathematics Department, 2000.

[9] M. I. G. Bloor and M. J. Wilson. Generating Blend Surfaces Using Partial Differential Equations. *Computer Aided Design*, 21(3):165–171, 1989.

[10] M. I. G. Bloor and M. J. Wilson. Generating N-sided patches with Partial Differential Equations. In *Advances in computer Graphics*, pages 129–145. Springer-Verlag, 1989.

[11] M. I. G. Bloor and M. J. Wilson. Using Partial Differential Equations to Generate Free-Form Surfaces. *Computer Aided Design*, 22(4):221–234, 1990.

[12] D. Breen and R. Whitaker. A Level-Set Approach for the Metamorphosis of Solid Models. *IEEE Transactions of Visualization and Computer Graphics* , 7(2):173–192, 2001.

[13] B. Y. Chen. *Total Mean Curvature and Submanifolds of finite Type*. World Scientific, 1984.

[14] D. L. Chopp and J. A. Sethian. Motion by Intrinsic Laplacian of Curvature. *Interfaces and Free Boundaries*, 1:1–18, 1999.

[15] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration using Willmore flow with boundary conditions, http://numerik.math.uni-duisburg.de/research/poster/surfacerest.pdf, 2002.

[16] U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic Geometric Diffusion in Surface Processing. In *Proceedings of Viz2000, IEEE Visualization*, pages 397–505, Salt Lake City, Utah, 2000.

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

[18] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. *SIGGRAPH99*, pages 317–324, 1999.

[19] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Discrete Differential-Geometry Operators in nD, http://www.multires.caltech.edu/pubs/, 2000.

[20] M. do Carmo. *Riemannian Geometry*. Boston, 1992.

[21] G. Dziuk. An algorithm for evolutionary surfaces. *Mumerische Mathematik*, 58:603–611, 1991.

[22] C. L. Epstein and M. Gage. The curve shortening flow. In A. Chorin and A. Majda, editors, *Wave Motion: Theory, Modeling, and Computation*. Springer-Verlag, New York, 1987.

[23] J. Escher, U. F. Mayer, and G. Simonett. The Surface Diffusion Flow for Immersed Hypersurfaces. *SIAM J. Math. Anal*, 29(6):1419–1433, 1998.

[24] J. Escher and G. Simonett. The Volume Preserving Mean Curvature Flow Near Spheres. *Proceedings of the American Mathematical Society*, 126(9):2789–2796, 1998.

[25] G. Greiner. Variational design and fairing of spline surface. *Computer Graphics Forum*, 13:143–154, 1994.

[26] A. Hubeli and M. Gross. Fairing Of Non-Manifolds For Visualization. In *Proceedings of Viz2000, IEEE Visualization*, pages 407–414, Salt Lake City, Utah, 2000.

[27] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. In Siggraph2002, pages 339–346, 2002.

[28] L. Kobbelt. Discrete Fairing. In Tim Goodman and Ralph Martin, editors, The Mathematics of Surfaces VII, pages 101–129. Information Geometers, 1996.

[29] L. Kobbelt, M. Botsch, U. Schwanecke, and H. P. Seidel. Feature sensitive surface extraction from volume data. In Siggraph2001, pages 51–66, 2001.

[30] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive Muti-Resolution Modeling on Arbitrary Meshes. *SIGGRAPH98*, pages 105–114, 1998.

[31] L. Kobbelt, T. Hesse, H. Prautzsch, and K. Schweizerhof. Iterative Mesh Generation for FE-computation on Free Form Surfaces. *Engng. Comput.*, 14:806–820, 1997.

[32] J. Lang. *Adaptive multilevel solution of nonlinear parabolic PDE systems : theory, algorithm, and applications*. Berlin ; New York : Springer, 2001.

[33] J. L. Mallet. Discrete Smooth Interpolation in Geometric Modelling. *Computer Aided Design*, 24(4):178–191, 1992.

[34] U. F. Mayer. Numerical Solutions for the Surface Diffusion Flow in Three Space Dimensions. *Computational and Applied Mathematics (to appear)*, 2001.

[35] U. F. Mayer and G. Simonett. A Numerical Scheme for Nonsymmetric Solutions of Curvature Driven Free Boundary problems with Applications to the Willmore Flow, *Manuscript*.

[36] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete Differential- Geometry Operator for Triangulated 2-manifolds, *manuscript*.

[37] H. Moreton and C. Sequin. Functional Optimization for Fair Surface Design. *ACM Computer Graphics*, pages 409 – 420, 1992.

[38] K. Museth, D. Breen, R. Whitaker, and A. Barr. Level set surface editing operators. In Siggraph02, pages 330–338, 2002.

[39] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski. Polyhedral surafce smoothing with simultaneous mesh regularization. In Geometric Modeling and Processing Proceedings, pages 229–237, 2000.

[40] S. J. Osher and R. P. Fedkiw. Level set methods. CAM Report 00-07, UCLA, Mathematics Department, 2000.

[41] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In IEEE Computer Society Workshop on Computer Vision, 1987.

[42] J Peters and M. Wittman. Smooth blending of basic surfaces using trivariate box spline. In *IMA 96, The Mathematics of Surfaces*. Dundee, UK, 1996.

[43] T. Preußer and M. Rumpf. An adaptive finite element method for large scale image processing. In *Scale-Space Theories in Computer Vision*, pages 232–234, 1999.

[44] N. Sapidis. *Designing Fair Curves and Surfaces*. SIAM, Philadelphia, 1994.

[45] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge, University Press, 2001.

[46] R. Schneider and L. Kobbelt. Generating Fair Meshes with G^1 Boundary conditions. In *Geometric Modeling and Processing*, pages 251–261. 2000.

[47] R. Schneider and L. Kobbelt. Geometric Fairing of Triangular Meshes for Free-form Surface Design, 2001.

[48] G. Simonett. The Willmore Flow for Near Spheres. *Differential and Integral Equations*, 14(8):1005–1014, 2001.

[49] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH '95 Proceedings*, pages 351–358, 1995.

[50] G. Taubin. Signal processing on polygonal meshes. In *EUROGRPHICS*. 2000.

[51] J. Vollmer, R. Mencl, and H. Miller. Improved Laplacian Smoothing of Noisy Surface Meshes. Research Report 711, 1999.

[52] J. Weickert. *Anisotropic Diffusion in Image Processing*. B. G. Teubner Stuttgart, 1998.

[53] W. Welch and A. Witkin. Variational Surface Modeling. *Computer Graphics*, 26:157–166, 1992.

[54] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. In *SIGGRAPH '94 Proceedings*, volume 28, pages 247–256, July 1994.

[55] R. Westermann, C. Johnson, and T. Ertl. A Level-Set Method for Flow Visualization. In *Proceedings of Viz2000, IEEE Visualization*, pages 147–154, Salt Lake City, Utah, 2000.

[56] R. Whitaker and D. Breen. Level set models for the deformation of solid objects. In *Proceedings of the 3rd International Workshop on Implicit Surfaces, Eurographics Association*, pages 19–35, June 1998.

[57] B. White. Evolution of curves and surfaces by mean curvature. In *Proc. of the Inernational Congress of Mathematicians, Vol. I*, pages 525–538, Beijing, 2002.

[58] T. J. Willmore. *Riemannian Geometry*. Clareden Press, Oxford, 1993.

[59] G. Xu. Convergence of Discrete Laplace-Beltrami Operators over Surfaces, *Manuscript*, 2003.

[60] G. Xu, C. Bajaj, and H. Huang. C^1 Modeling with A-patches from Rational Trivariate Functions. *Computer Aided Geometric Design*, 18(3):221–243, 2001.

[61] H. K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. CAM Report 01-01, UCLA, Mathematics Department, 2001.

[62] H. K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and Non-parametric Shape Reconstruction from Unorgainzed Points Using Variational Level Set Method. *Computer Vision and Image Understanding*, 80(3):295–319, 2000.