

A Cookbook for Variational Subdivision

Joe Warren Henrik Weimer
Department of Computer Science
Rice University

Abstract

These course notes will attempt to answer the following questions: *What is subdivision? How can the rules for subdivision schemes be derived in a systematic manner? How can these rules be extended to handle special topological features such as extraordinary points or creases?* We will argue that most subdivision schemes correspond to a special type of multigrid method that generates shapes which solve (or nearly solve) a variational problem. These subdivision schemes are influenced by two factors: the variational functional and the local grid topology. Using a recipe based on this observation, we will build subdivision schemes for several interesting examples including B-splines, minimum energy curve networks, membrane splines and fluid flow.

1 Introduction

Computer graphics intrinsically depends on the mathematical and algorithmic representation of shape. Traditionally, smooth shapes have been represented using parametric representations such as B-splines or implicit representations such as algebraic surfaces. During the last two decades, subdivision has evolved as a simple yet flexible method for the modeling of geometric shapes. Starting with a coarse polyhedron, a subdivision scheme defines a sequence of increasingly dense polyhedra that converge to a smooth limit shape.

Modeling with subdivision is simple and easy to implement since only discrete geometric entities such as points, edges and polygons are involved. Furthermore, the transformation rules that yield the next finer shape are simple weighted averages of the vertices of the coarser shape. Thus these transformations can be implemented easily and computed very efficiently. Yet, subdivision schemes are flexible since the actual subdivision rules can be chosen very generally. Subdivision schemes can be evaluated locally, that is to say if only some part of the whole model shape is of interest, computational effort has only to be expended on the evaluation of the shape in that localized area. Finally, modeling with subdivision gives us a multiresolution representation of the shapes for granted: The control polyhedra at coarser levels approximate the limit shape well and can be refined arbitrarily. Throughout the last few years, multiresolution surface modeling systems have been presented that exploit this idea (see [5], [13]).

With these systems the user is allowed to modify the polygonal shape at any level of subdivision and therefore has control over the shape of the object at any scale.

Much work has been dedicated to analyzing a subdivision scheme given as a set of known subdivision rules. However, relatively little work has been done on methods for systematically generating interesting subdivision schemes. Two main techniques for generating subdivision schemes are known: Box-splines (see deBoor et. al [2]) are based on the idea of repeated convolution and possess relatively simple subdivision schemes. Unfortunately, box-splines are restricted to uniform grids. Extensions of box-splines to topologically non-uniform grids (i.e. polyhedra) has traditionally been done in a somewhat ad-hoc manner that defies generalization (see Loop [7]).

Variational methods (Kobbelt [4], Mallet [8], Warren and Weimer [10],[11]) generate subdivision schemes as the solution process to certain types of variational problems. Given an initial shape, the subdivision scheme generates a sequence of shapes converging to a limit shape that follows the initial shape and minimizes a functional associated with the variational problem. These notes will show that the actual rules for these variational subdivision schemes are determined by two factors:

- the variational functional, and,
- the local grid topology.

Variational methods have been very successful in Computer Aided Geometric Design because these methods yield fair, smooth shapes. The beauty of the variational approach to subdivision is that the known box-splines can be subsumed as a special case (see Warren and Weimer [11]) with non-uniform grids being handled in a systematic manner. In these notes, we will demonstrate a simple "recipe" that allows one to systematically design customized subdivision schemes. This method can easily handle special topological features such as extraordinary points and creases. First, we review some common methods for solving variational problems. As we shall see, these methods are intrinsically related to variational subdivision.

2 Multigrid

At its most basic level, variational modeling entails finding a shape defined over a given domain that minimizes a given continuous functional. Due to the difficulties of manipulating continuous functionals, a standard approach is to discretize the domain and to convert the problem into one of minimizing a corresponding discrete functional. By splitting the problem domain into a discrete grid T , the corresponding variational solution can be approximated by a discrete coefficient vector p with one value per grid point in T . The beauty of this approach is that many important variational problems can now be expressed as the minimization of a quadratic form $p^T E p$ where E is a symmetric, positive definite matrix whose entries depend on the variational functional.

This matrix E is often referred to as the *energy matrix* associated with the variational problem. E expresses a discrete approximation to the continuous variational functional over T . Using basic calculus, it is easy to show that $p^T E p$ is minimized if and only if $E p = 0$. Based on this observation, the rows of E are often viewed as

discrete differences that approximate the partial differential equation associated with the variational problem (via the Euler-Lagrange equations). Since $p^T E p$ is trivially minimized by $p = 0$, extra conditions are imposed on the minimization to ensure a non-trivial solution. As we shall see, the choice of these extra conditions is very important. For now, we simply consider systems of the form

$$E p = b,$$

where the entries of b capture the associated boundary conditions.

This system of linear equations can be solved using a direct solver, such as Gaussian elimination. However, in many cases, the grid T and its associated energy matrix E is so large that using a direct solver is impractical due to running time and space requirements. Fortunately, the matrices E associated with most variational problems are very sparse. Traditional iterative solution methods such as the Jacobi or Gauss-Seidel iteration can solve such systems of moderate size. Given an initial guess, these methods produce increasingly accurate solutions using localized, iterative update operations. See Varga [9] for an introduction to these techniques.

However, in practice, these iterative solution methods exhibit a very curious behavior: high-frequency errors are eliminated very fast, within a few iterations. However, low-frequency errors, governing the global appearance and nature of the solution, are eliminated only very slowly. In fact, one often observes that iterative solvers of this type "stall" after a certain number of iterations and the solution only improves very marginally once high-frequency errors have been eliminated. Multigrid methods try to circumvent this problem of "stalling". Using a sequence of denser and denser grids, multigrid computes solutions in a hierarchical fashion. If we denote the sequence of domain grids by T_i and the corresponding energy matrices by E_i , then the multigrid method attempts to find a sequence of vectors p_i satisfying the equation

$$E_i p_i = b_i \quad (1)$$

for increasing i , i.e. for increasingly dense grids. Typically, the solution process at level i consists of three steps (see Briggs [1] for more details):

1. **Prediction:** Compute an initial guess p_i^0 to the exact solution p_i . This initial guess is derived from the solution p_{i-1} on the next coarser grid using some type of linear prediction function. In terms of matrix notation, this prediction step can be written as

$$p_i^0 = S_{i-1} p_{i-1}$$

where S_{i-1} is a matrix that maps vectors over T_{i-1} to vectors over T_i . In multigrid terminology, S_{i-1} is a prolongation operator. A common choice for this prediction operation in a typical multigrid method is piecewise linear interpolation.

2. **Smoothing:** Improve the guess p_i^0 using k rounds of an iterative method such as Jacobi or Gauss-Seidel iteration. These iterative methods have the form:

$$p_i^{j+1} = A_i p_i^j + B_i b_i \quad (2)$$

where A_i and B_i are matrices that depend on E_i and the type of smoothing method chosen. If we let r_i^j denote the residual $E_i p_i^j - b_i$, then this residual decreases according to the equation:

$$r_i^{j+1} = A_i r_i^j.$$

3. **Coarse grid correction:** Restrict the residual r_i^k to the next coarser grid T_{i-1} by eliminating all entries in r_i^k which do not correspond to a gridpoint in T_{i-1} . Given this restricted residual r_{i-1} , solve the equation $E_{i-1} e_{i-1} = r_{i-1}$ on T_{i-1} . Finally, add the correction term $e_i = S_{i-1} e_{i-1}$ to the current approximation p_i^k .

Note the interaction of these three steps: At a given level of the multigrid process, high frequency errors are eliminated very rapidly due to the smoothing in step two. Of course, low frequency errors still remain after smoothing. However, restricting these low frequency errors to coarser grids eventually turns the errors into high frequency errors which are quickly eliminated by the smoothing operation on the coarse grid. In practice, multigrid has proven very effective at solving systems of linear equations of this type. As we shall see in the next section, multigrid and subdivision are intrinsically related.

3 Subdivision

For those familiar with subdivision, it is clear that subdivision and multigrid share some striking similarities. Each method produces a sequence of vectors p_i that converge to some "interesting" limit shape. In fact, if we closely examine the structure of multigrid, we can express subdivision as a special case of multigrid. Consider a version of multigrid in which only the prediction step is carried out at each level (i.e. the smoothing and coarse grid correction steps are omitted). This process would define a sequence of vectors p_i^0 satisfying

$$p_i^0 = S_{i-1} p_{i-1}^0.$$

This process is exactly subdivision! The predictor S_{i-1} is simply a subdivision matrix. Of course, the limit shape produced by this process depends on the type of predictor chosen. For a typical multigrid predictor, such as piecewise linear interpolation, the limit of the p_i^0 as $i \rightarrow \infty$ is just a piecewise linear shape. To have any hope that this subdivision process might produce a reasonable solution to the actual variational problem, we must base the predictor S_i very carefully on the particular variational problem. Note that if the predictor S_i happens to produce a solution p_i^0 whose corresponding residual r_i^0 is always zero, then the smoothing and coarse grid correction steps could be omitted without difficulty. Predictors S_i which produce zero residual are *perfect predictors* in the sense that they produce perfectly accurate solutions. Subdivision using perfect predictors produces shapes that exactly satisfy the variational problem. We next give a simple characterization for a general class of perfect predictors S_i in terms of the energy matrices E_i .

3.1 Perfect predictors

By definition, the solution vectors p_i satisfy the equation $E_i p_i = b_i$ where the matrix E_i is a discrete version of the variational functional on T_i . The vector b_i characterized the boundary conditions of the variational problem on T_i . The key in our derivation of perfect predictors is the structure of the right-hand side b_i . In particular, we suggest a specific structure for the b_i that results in a multigrid scheme with perfect predictors whose rows are either locally supported or highly localized. In particular, this choice replicates many known subdivision schemes such as those for B-splines and box-splines.

Let U_i be the *upsampling matrix* that takes a vector over T_i into a vector over the next finer grid T_{i+1} . U_i maps the entries of these vectors as follows: entries corresponding to grid points in T_i are replicated; entries corresponding to grid points in $T_{i+1} - T_i$ are set to zero. Thus, the upsampling matrix U_i consists of rows that are either zero (for new grid points) or a standard unit vector (for old grid points). Now, if we choose b_i to have the form

$$b_i = U_{i-1} U_{i-2} \dots U_0 E_0 p_0,$$

then equation 1 becomes

$$E_i p_i = U_{i-1} \dots U_0 E_0 p_0. \quad (3)$$

Standard interpolatory methods force minimization where the solutions p_i interpolate the values of p_0 on the initial grid T_0 . Instead, equation 3 forces minimization where the differences $E_i p_i$ interpolate the differences $E_0 p_0$ at grid points of T_0 . Repeated upsampling forces the remaining differences of $E_i p_i$ at grid points in $T_i - T_0$ to be zero, thus ensuring minimization of $p^T E p$ (see Warren and Weimer [11] for more details). Given this framework, we can now characterize perfect predictors for the multigrid scheme defined by equation 3.

Theorem: Given a solution p_{i-1} to equation 3, let p_i be the guess produced by the predictor S_{i-1} (i.e. $p_i = S_{i-1} p_{i-1}$). If S_{i-1} satisfies the equation

$$E_i S_{i-1} = U_{i-1} E_{i-1}, \quad (4)$$

then S_{i-1} is a perfect predictor.

Proof: Multiply both sides of equation 4 by p_{i-1} from the right. Next, apply the right-hand side of equation 3 to $E_{i-1} p_{i-1}$. Since $p_i = S_{i-1} p_{i-1}$, we have shown that p_i also satisfies equation 3.

3.2 Example: Minimum energy curves

Historically, splines were a commonly used drafting tool in mechanical and engineering design, before the advent of computer aided design systems. Using a thin, flexible strip of metal or wood a designer could draw smooth curves by first anchoring the strip to a sequence of $n + 1$ points on the drafting table and then letting the strip slide freely into a minimum energy configuration. Mathematically, this strip of metal or wood could be modeled by a curve $\mathbf{p}[t]$ where $t \in [0, n]$ (as a convention throughout these

notes, continuous functions are denoted by bold face letters). The bending energy of $\mathbf{p}[t]$ can be approximated by the continuous functional

$$\mathbf{E}[\mathbf{p}[t]] = \int_0^n \mathbf{p}^{(2)}[t]^2 dt. \quad (5)$$

The associated variational problem is to minimize $\mathbf{E}[\mathbf{p}[t]]$ subject to $\mathbf{p}[j]$ interpolating the j th anchor point. The minimizing functions for this particular problem are well-known: $\mathbf{p}[t]$ is a \mathbf{C}^2 piecewise cubic function known as a *natural cubic spline*. Cubic splines are one of the fundamental tools of geometric design. In particular, they possess a locally supported basis, the B-spline basis, with a number of remarkable properties including a particularly simple subdivision scheme. Our goal in this section is to discretize cubic splines, convert the problem to the multigrid setting and then to systematically derive the subdivision scheme for cubic B-splines as a consequence of equation 3. Later throughout these notes we will generalize this methodology to derive subdivision schemes for other interesting variational problems.

A typical discretization of the problem from equation 5 is to replace $\mathbf{p}[t]$ by a polygon $p = (p_0, p_1, \dots, p_{2n})$. The even index vertices of p are placed at the $n + 1$ anchor points. The remaining, odd index vertices of p are positioned such that the discrete bending energy of p is minimized. Thus, the vertex p_j should approximate $\mathbf{p}\left[\frac{j}{2}\right]$. In this case, the discrete analog of $\mathbf{E}[\mathbf{p}[t]]$ is the functional

$$\mathbf{E}[p] = 8 \sum_{j=1}^{2n-1} (p_{j-1} - 2p_j + p_{j+1})^2. \quad (6)$$

The term $p_{j-1} - 2p_j + p_{j+1}$ is a discrete approximation to the continuous expression $\mathbf{p}^{(2)}[t]$. The summation is the discrete analog of integration. The constant 8 arises due to the half integer grid spacing. This spacing causes the second difference term to be normalized by a factor of 4. Squaring these terms raises this factor to 16. This spacing also introduces an extra factor of $\frac{1}{2}$ into the summation. Thus, the total effect of halving the grid spacing is to multiply the energy functional by a factor of 8.

Moving this process into a multigrid setting, the polygon p is replaced by a sequence of polygons p_i each of whom have $n * 2^i$ vertices. Each polygon p_i has those vertices with indices that are multiples of 2^i fixed at the anchor points. The remaining vertices are positioned to minimize the functional

$$\mathbf{E}[p_i] = 8^i * \sum_{j=1}^{2^i n - 1} ((p_i)_{j-1} - 2(p_i)_j + (p_i)_{j+1})^2.$$

Since these discrete functionals $\mathbf{E}[p_i]$ converge to the continuous functional $\mathbf{E}[\mathbf{p}[t]]$ as $i \rightarrow \infty$, the polygons p_i converge to $\mathbf{p}[t]$ as $i \rightarrow \infty$. Using the energy matrices E_i associated with the discrete functional we can write

$$\mathbf{E}[p_i] = p_i^T E_i p_i$$

where, away from the boundary, the rows of the matrices E_i are all shifts of the single sequence $8^i * (1, -4, 6, -4, 1)$. Note that this sequence encodes the mask for a fourth difference.

To employ equation 3, we distinguish two kinds of local grid configurations. The interior of the polygon p_i is topologically uniform with all grid points looking similar. Near the boundary of the grid, i.e. at the endpoints of p_i , we distinguish among the grid points based on their distance from the actual boundary. Our goal is now to find two perfect predictors, one for the interior of the polygon and one for the endpoints of the polygon. For the interior, we note that upsampling consists in splitting coarse coefficients apart by inserting a 0 inbetween any two adjacent coefficients. In terms of matrices, the condition $E_i S_{i-1} = U_{i-1} E_{i-1}$ can be written as

$$8 \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 \\ \cdot & 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ \cdot & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ \cdot & 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ \cdot & 0 & 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 \\ \cdot & \cdot \end{pmatrix} S_{i-1} = \\ \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & 1 & 0 & 0 & 0 \\ \cdot & 0 & 0 & 0 & 0 & 0 \\ \cdot & 0 & 0 & 1 & 0 & 0 \\ \cdot & 0 & 0 & 0 & 0 & 0 \\ \cdot & 0 & 0 & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 6 & -4 & 1 & 0 & 0 \\ \cdot & -4 & 6 & -4 & 1 & 0 \\ \cdot & 1 & -4 & 6 & -4 & 1 \\ \cdot & 0 & 1 & -4 & 6 & -4 \\ \cdot & 0 & 0 & 1 & -4 & 6 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

Here, the dots indicate that only a small portion of the matrices is shown. However, since the rows in the system repeat, this portion is sufficient to determine a perfect predictor for the interior of the grid. Since this system is rank deficient, there is more than one possible solution to the system above. However, there is one particularly nice solution with minimal support. This predictor has the form

$$S_{i-1} = \frac{1}{8} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 6 & 1 & 0 & 0 & 0 \\ \cdot & 4 & 4 & 0 & 0 & 0 \\ \cdot & 1 & 6 & 1 & 0 & 0 \\ \cdot & 0 & 4 & 4 & 0 & 0 \\ \cdot & 0 & 1 & 6 & 1 & 0 \\ \cdot & 0 & 0 & 4 & 4 & 0 \\ \cdot & 0 & 0 & 1 & 6 & 1 \\ \cdot & 0 & 0 & 0 & 4 & 4 \\ \cdot & 0 & 0 & 0 & 1 & 6 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

This matrix exactly encodes the subdivision scheme for uniform cubic B-splines due to Lane and Riesenfeld [6]. Note the simple structure of S_{i-1} : all columns contain the same sequence of coefficients $(1, 4, 6, 4, 1)$ and the matrix has a 2-to-1-slant, i.e. as we go over one column to the right, the column is shifted down by two. The action of S_{i-1} on a polygon p_{i-1} is as follows: vertices of p_i are positioned to lie either on the midpoint of edges of p_{i-1} or near vertices of p_{i-1} (by taking $\frac{6}{8}$ of a vertex plus $\frac{1}{8}$ of both of its neighbors).

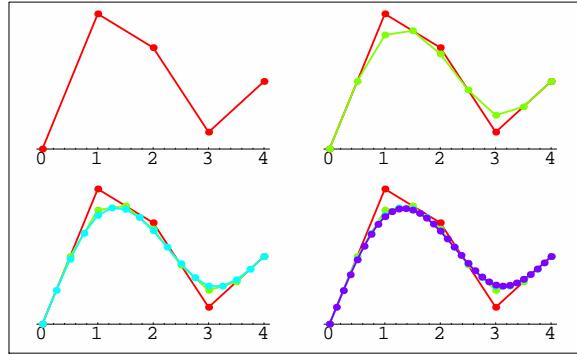


Figure 1: Subdivision for cubic B-splines.

The second type of subdivision rule arises at the endpoints of the polygons. There, the matrices E_i have a slightly more complicated structure. Equation 4 has in this case the form

$$8 \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 & 0 & \dots \\ -2 & 5 & -4 & 1 & 0 & 0 & 0 & \dots \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & \dots \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & \dots \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & \dots \\ \vdots & \ddots \end{pmatrix} S_{i-1} =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \cdot \begin{pmatrix} 1 & -2 & 1 & \dots \\ -2 & 5 & -4 & \dots \\ 1 & -4 & 6 & \dots \\ 0 & 1 & -4 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

It is now easy to verify that S_{i-1} has the form

$$S_{i-1} = \frac{1}{8} \begin{pmatrix} 8 & 0 & 0 & \dots \\ 4 & 4 & 0 & \dots \\ 1 & 6 & 1 & \dots \\ 0 & 4 & 4 & \dots \\ 0 & 1 & 6 & \dots \\ 0 & 0 & 4 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Again, S_{i-1} has a simple structure. Its first row forces the endpoint of p_i to interpolate the endpoint of p_{i-1} . The remaining rows agree with interior subdivision rules. In fact, this rule produces cubic splines satisfying the natural boundary condition.

These predictors (subdivision rules) S_i can easily generate arbitrarily dense polygons p_i . Because these predictors are perfect (i.e. satisfy equation 4), the polygons p_i converge to the natural cubic spline $\mathbf{p}[t]$. Figure 1 shows an example of three rounds

of prediction (i.e. subdivision) applied to the initial polygon $\{(0, 4), (1, 12), (2, 10), (3, 5), (4, 8)\}$.

4 A recipe for variational subdivision

The previous section argued that subdivision can be viewed as a multigrid method that employs a perfect predictor. In every round of the multigrid process, the predictor produces an error free guess for the next solution. Thus, smoothing and coarse grid error correction are rendered unnecessary. As an example, we derived perfect predictors for cubic B-splines and noted that the resulting matrices have a very simple structure with local support. This section presents a general "recipe" for creating a variational subdivision scheme based on these ideas. The recipe is outlined in figure 2 below.

1. Define a discrete variational functional,
2. Generate a catalog of simple mesh types arising in the problem,
3. For each mesh type, solve for the locally supported predictor S_i that minimizes $\|E_i S_{i-1} - U_{i-1} E_{i-1}\|$.

Figure 2: A recipe for custom subdivision schemes.

The minimization process in step 3 is necessary because for many important problems a perfect predictor with local support does not exist. However, by restricting S_{i-1} to have fixed support and minimizing the equation $\|E_i S_{i-1} - U_{i-1} E_{i-1}\|$, we can compute predictors that produce a reasonably accurate approximation to the exact solution. The corresponding subdivision schemes are locally supported and produce limit shapes that are good qualitative approximations to the exact solution. In fact, by increasing the support of S_{i-1} the solution can be made arbitrarily precise. Of course, highly local schemes cannot match the accuracy of a full multigrid method. However, at a minimum, they provide an excellent initial guess for the smoothing and coarse grid correction portions of full multigrid.

4.1 Relation to other work

Subdivision has been studied significantly over the course of the last decades. The specific link between subdivision and variational problems has also been investigated by Kobbelt [4] and Mallet [8]. Each proposes subdivision schemes consisting of a simple predictor (perhaps piecewise linear interpolation or some known subdivision scheme) followed by some number of smoothing steps. (In multigrid terminology, this process is referred to as nested iteration.) Both authors neglect the possibility that a predictor customized to the particular variational problem might produce superior results.

Both Kobbelt [4] and Mallet [8] propose interpolatory schemes. The schemes proposed here are approximating in the sense that the limit shape follows an initial shape while minimizing the variational problem. Traditionally, approximating functions, such as B-splines, have been more successful in geometric design applications than their interpolating counterparts because the resulting solutions typically oscillate less and have more localized support. In some cases (e.g. B-splines and box-splines), there exist exact, approximating schemes with locally supported bases whose interpolating counterparts have globally supported bases. Finally, there exist variational problems whose interpolating schemes do not converge in any reasonable sense, while their approximating counterparts do. One such example are the membrane splines discussed later in the paper.

5 Examples

This section applies the recipe of figure 2 to several interesting example problems. First, we generalize the subdivision scheme for minimum energy curves (B-splines) given in the previous section to a network of curves. Next, we derive a subdivision scheme for surfaces that mimic an elastic membrane. Then, we combine these two problems and produce a subdivision scheme for a minimum energy curve network connected by elastic membranes. Finally, we conclude with some ideas on further applications of the recipe.

5.1 Minimum energy curve networks

In the case of minimum energy curves, our analysis was carried out entirely in the discrete domain using a sequence of polygons. Here, we take a similar approach for curve networks. A *polygon network* consists of a set of vertices connected by a set of edges. Note that more than two edges in the network may be incident on a single vertex of the network. Topologically, subdivision of a polygon network corresponds to splitting each edge into two connected edges. In this framework, a curve network is the limit of a sequence of increasingly dense polygonal networks produced by subdivision. Our goal is to construct subdivision rules that position vertices on the polygon network such that the limit of the process is a minimum energy curve network.

5.1.1 Define the energy functional

We first need to generalize the functional of equation 6 to the case of polygon networks. The crux of the functional in equation 6 was the term $p_{j-1} - 2p_j + p_{j+1}$. This expression measured the amount that the polygon (p_{j-1}, p_j, p_{j+1}) "bends". In the case of polygon networks, several curves may pass through a vertex. Thus, several terms of this form may be necessary for each vertex. We suggest tagging the j th vertex of the network with a list $a[j]$ of neighbor pairs. This tagging scheme allows the user a great deal of flexibility in creating interesting polygon networks. The discrete energy functional for the polygon network can be written as:

$$\mathbf{E}[p_i] = 8^i \sum_{j=1}^{|p_i|} \sum_{k=i}^{|a[j]|} ((p_i)_{a[j,k,1]} - 2(p_i)_k + (p_i)_{a[j,k,2]})^2. \quad (7)$$

Inheritance of these tags during subdivision is straightforward. A new vertex is inserted between a pair of adjacent old vertices and is assigned a single tag pair consisting of the two old vertices. Neighbor tags for old vertices are updated by replacing an old adjacent vertex with the vertex inserted on the corresponding edge.

5.1.2 Generate a catalog of simple mesh types

We next identify the various types of local configurations that are possible in a polygon network. In theory, there are an infinite number of possible configurations. For example, any number of curves can pass through a common vertex! In practice, we focus on a few canonical vertex configurations and analyze those. The simplest two vertex configurations have already been identified and analyzed. For vertices on the interior of a curve and at the isolated endpoint of a curve, the energy functional of equation 7 degenerates into the energy functional of equation 6. Since the analysis of the previous section was entirely local, the subdivision rules for these two vertex configurations remain unchanged. Due to the interpolatory nature of the endpoint rule, the rule for several curves terminating at a common vertex remains simple interpolation. We now focus on three important types of vertex configurations. Other types of vertex configurations can typically be expressed as variants or combinations of these three configurations.

- **X vertex:** Two curves smoothly intersecting at a vertex.
- **Y vertex:** One curve smoothly branching into two curves at a vertex.
- **T vertex:** One curve terminating at a vertex on a smooth curve.

5.1.3 Solve for locally supported predictors

To apply equation 4 for a particular vertex configuration, we first construct the matrices E_i appropriate for the configuration. As in the previous case, all polygons incident on the vertex are assumed to extend infinitely. Note that this assumption causes the topology of the network to be locally invariant under subdivision. As a result, all matrices E_i are multiples of a single common, infinite matrix E . Given this matrix E , we next construct the predictor S defining the subdivision scheme at the vertex. First, we fixed the support of the rows of S to have an appropriate size (in this case, support equivalent to that of cubic splines). Next, we construct the matrix S with non-zero entries expressed as indeterminates. Extra conditions such as constant precision, linear precision or symmetry can also be enforced in terms of constraints on these indeterminates.

Given the matrices E and S , we are now ready to minimize the matrix expression $\|8E S - U E\|$. This expression can be minimized with respect to a variety of norms. In general, we recommend using the ∞ -norm (although we have also used the 2-norm).

Due to the repetition of rows in E , minimizing this norm leads to a linear program in a small number of variables that can be solved in a reasonably efficient manner (see [10] for more details). Unfortunately, these vertex configurations do not admit perfect predictors with local support, i.e. the matrix norm cannot be forced to zero. However, they do have simple, locally supported rules that provide a quite reasonable approximation to the true minimizer. Figure 3 summarizes these rules.

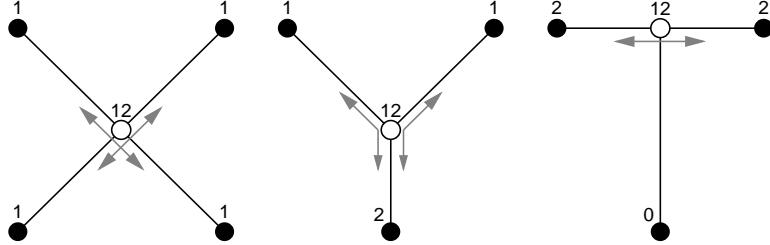


Figure 3: Subdivision rules for various vertex configurations

Figure 4 shows three curve networks produced by these rules and compares them to the true minimum energy curve networks. The top row shows an example using the X rule. The middle row shows an example using the Y rule. Note that the locally supported subdivision rule for a Y vertex fails to produce the smooth branching of the exact minimum energy curve network because the exact solution is globally supported. The bottom row shows an example using the T rule. Based on experimentation with a wide range of vertex configurations, we propose a general rule for subdivision at vertex i of the original network. Let $a[i]$ be the list of neighbor pairs associated with vertex i : weight vertex i by $\frac{3}{4}$; weight the remaining neighbors of vertex i by the number of times that they appear in $a[i]$ divided by $8 * |a[i]|$. Note that this general rule reproduces the special rules given in figure 3.

5.2 Membrane splines

This section develops a subdivision scheme producing spline surfaces that behave like an elastic membrane. Given a triangulated polyhedron, the scheme produces a sequence of increasingly faceted polyhedra using the standard 4-1 triangular face split. The resulting sequence of triangulated polyhedra converges to a surface that follows the initial polyhedron and approximately minimizes a simple notion of discrete elastic energy. This scheme is essentially a generalization of the functional subdivision scheme based on Laplace's equation that appeared in Warren and Weimer [11].

5.2.1 Define the energy functional

Given a polyhedron p with triangular faces, let $a[i]$ denote a list containing the indices of those vertices adjacent to vertex i . One reasonable measure of the elastic energy of

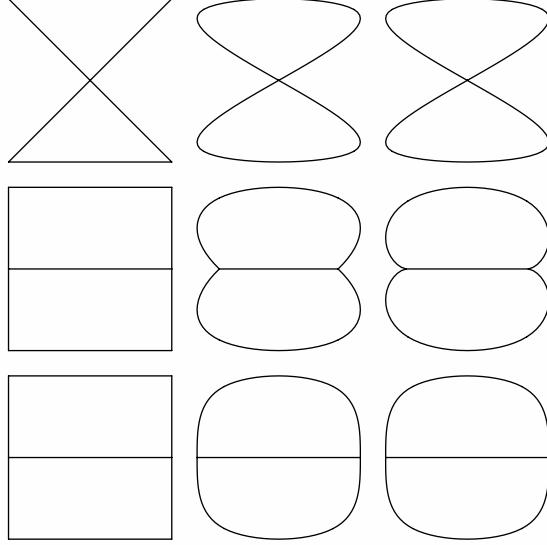


Figure 4: Initial networks (left), minimum energy curve networks via subdivision (middle), and exact solutions (right).

p is the sum of the squares of the edge lengths in p ,

$$\mathbf{E}[p] = \frac{1}{2} \sum_{i=1}^{|p|} \sum_{j=i}^{|q[i]|} (p_i - p_{q[i,j]})^2. \quad (8)$$

If $\mathbf{E}[p]$ is expressed in matrix form as $p^T E p$, then E is simply the Laplacian of the polyhedron p . In the case of polygon networks, a normalizing constant of 8^i was necessary when generalizing the functional to a sequence of polygon networks p_i produced by subdivision. For membrane energy, this constant conveniently works out to be 1: The factor 4 induced by doubling the edge length is canceled by the factor $\frac{1}{4}$ for summing over a two-dimensional mesh. Therefore, the functional of equation 8 can be applied independent of the level of subdivision.

5.2.2 Generate a catalog of simple mesh types

Cataloging different types of local topologies for triangular meshes is straightforward. The standard approach is to classify the mesh locally based on the valence of a vertex. In the uniform case, all vertices of a triangular mesh have valence 6. Vertices with valence other than 6 are classified as *extraordinary points*. Since subdivision leaves the valence of these vertices unchanged while introducing only new vertices of valence six, the meshes in our catalog consist of a single extraordinary point surrounded by an infinite uniform triangular mesh. Note that the infinite energy matrix E associated with such a mesh is again independent of the level of subdivision.

5.2.3 Solve for locally supported predictors

Our approach follows that for polygon networks. For each mesh type, construct the associated energy matrix E and fix a support for the predictor S . (For membranes, we suggest the same support as used in Loop's scheme [7]). Then, minimize the expression $\|E S - U E\|$ where the entries of S are treated as unknowns. Typically, the rules of S are first computed in the uniform case (i.e. valence 6). These uniform rules can then be used to pad the matrix S that expresses the subdivision rules at extraordinary points.

As in the case of minimum energy curve networks, it is impossible to satisfy $E S = U E$ exactly. Therefore, there does not exist a locally supported subdivision scheme that behaves exactly like a membrane. However, there do exist simple, locally supported subdivision rules that provide a reasonable approximation to the behavior of a membrane. Figure 5 summarizes these rules. Note that these rules are not the exact minimizer of $\|E S - U E\|$, but have the property that they are simple and close to the true minimizer.

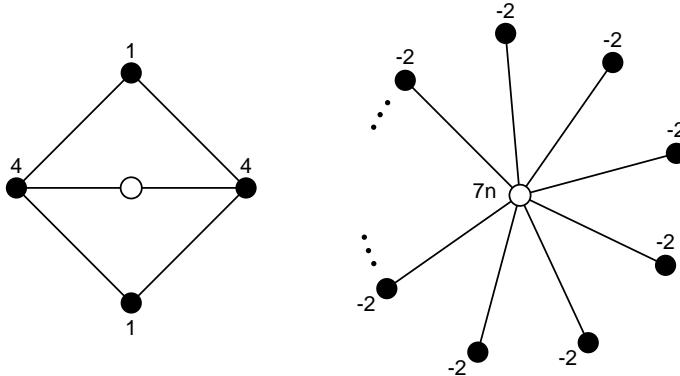


Figure 5: Subdivision rules for membrane splines.

The figure 6 shows an example application of this subdivision scheme. A coarse octahedron is shown on the top with increasing levels of subdivision below. Note that the subdivision scheme produces a very "spiky" shape. This effect models the behavior of an elastic membrane. The surface spikes at coarse vertices and stretches into a narrow tube to produce low elastic energy. In general, the subdivision scheme yields a good low frequency approximation of the exact minimizer, in the sense that the overall shape of the resulting objects is accurate. However, there are also smaller spikes scattered over the surfaces away from the original coarsest vertices. These extra spikes are a result of the fact that the approximate S used in the subdivision scheme does not exactly satisfy $E S = U E$. Specifically, those differences of $E S$ corresponding to new vertices are not exactly zero as required by $U E$.

At this point, we have two options. The first option is to increase the support of the predictor S . This change will produce a better predictor and a more accurate subdivision scheme. However, the resulting shapes will still exhibit the same problem (although at a much reduced scale). Also, manipulating subdivision rules over larger

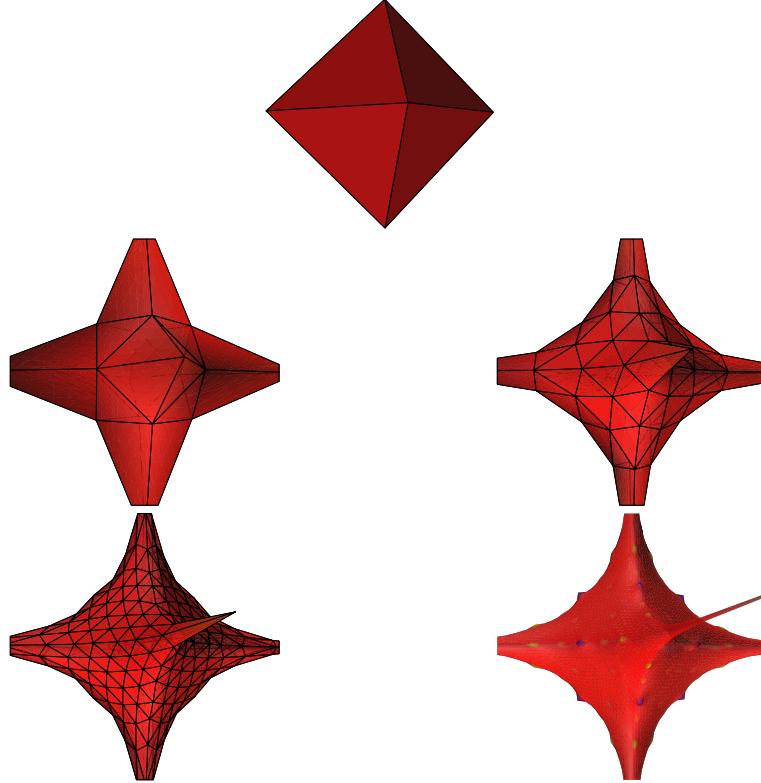


Figure 6: Membrane spline defined via subdivision. Some of the large spikes were truncated to focus on the interesting parts of the figure.

supports can be difficult due to the interactions of extraordinary points on the initial coarse grid.

The second possibility is to employ several rounds of smoothing after each round of subdivision (as proposed by Kobbelt [4] and Mallet [8]). After each round of subdivision, we suggest running several rounds of smoothing *with one crucial modification*. Recalling equation 2 (one round of subdivision with $b_i = U_{i-1}E_{i-1}p_{i-1}$), we can express standard smoothing as

$$p_i^{j+1} = A_i p_i^j + B_i (U_{i-1} E_{i-1} p_{i-1}) \quad (9)$$

where A_i and B_i depend on E_i . Since p_{i-1} has been produced by an inexact subdivision scheme, many of the differences in $E_{i-1}p_{i-1}$ that are zero for the exact scheme are actually non-zero for the inexact scheme. Instead of propagating these non-zero differences, we suggest modifying the upsampling matrix U_{i-1} in equation 9 to upsample only those differences whose corresponding exact differences are non-zero. Typically, this change causes only differences corresponding to a subset of the vertices in p_0 to be upsampled. (Some entries in $E_0 p_0$ may also be exactly zero.)

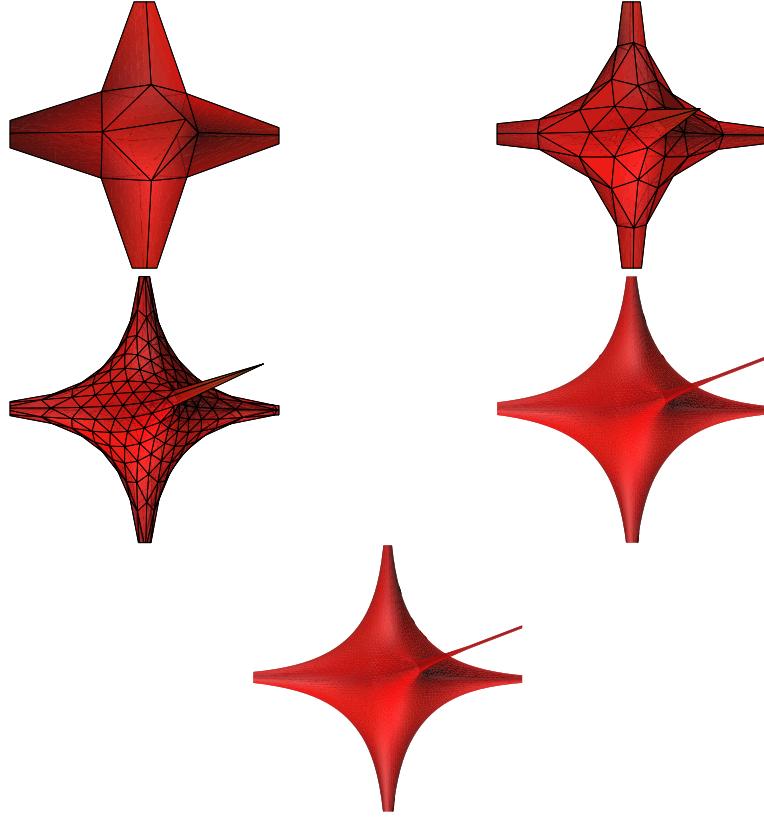


Figure 7: Membrane spline defined via subdivision plus smoothing (3 rounds per level). Some of the large spikes were truncated to focus on the interesting parts of the figure.

This modified smoothing effectively eliminates the extraneous spikes of figure 6 due to their high frequency nature. Combining the low frequency accuracy of S with the high frequency correction of smoothing yields a subdivision scheme that is visually accurate. Note that because the overall, low frequency shape yielded by the local subdivision scheme S is very close to the exact solution, only a few rounds of smoothing are necessary to eliminate all high frequency noise. Figure 7 shows this combined scheme (employing 3 rounds of Jacobi smoothing) applied to the coarse octahedron. The surface at the bottom of figure 7 is the exact solution.

Note that this subdivision scheme for membrane splines is approximating, not interpolating. In particular, the spikes of p_i do not interpolate the vertices of the coarse polyhedron p_0 . Instead, the spikes are determined by the differences of $E_0 p_0$. If one of these initial differences is zero, then the corresponding limit surface is smooth at this vertex. Figure 8 illustrates a modification of the octahedron in which the bottom vertex has been pushed upward to form a square pyramid. Again, the exact solution is on the bottom right. Since the base of this pyramid is now planar, the difference in

$E_0 p_0$ corresponding to this vertex is zero and, thus, the limit surface is smooth at this vertex.

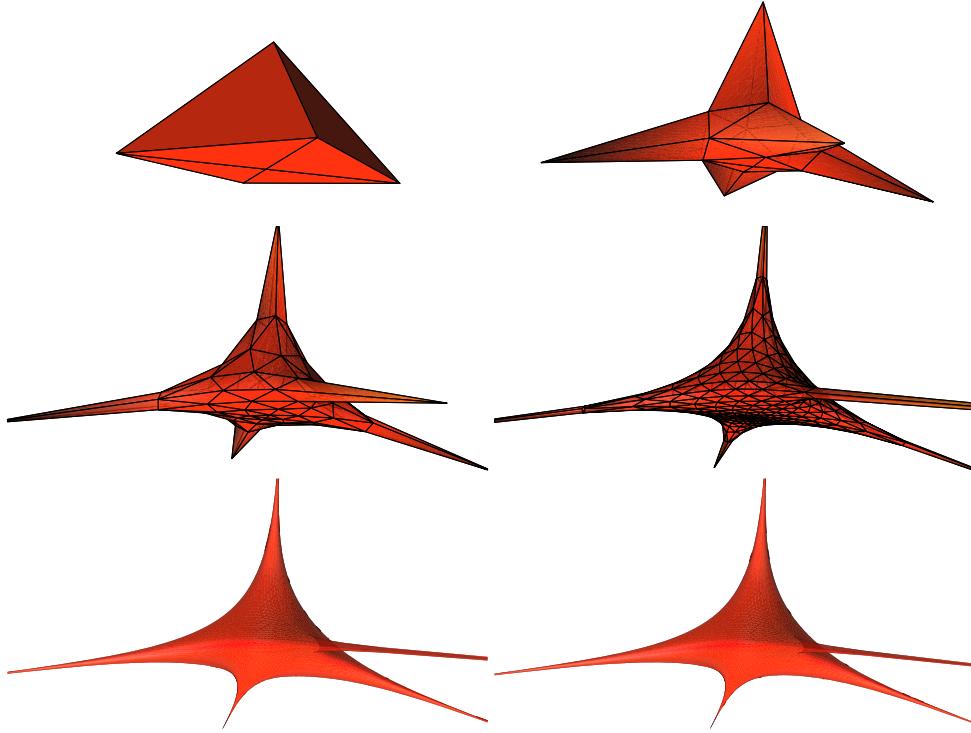


Figure 8: A membrane spline for a square pyramid (3 rounds of smoothing per level).

Finally, note that the lengths of the spikes in figures 7 and 8 diverge very gradually as the number of rounds of subdivision increases. Fortunately, the exact scheme is convergent everywhere except at the tips of these spikes. If instead one attempts to force interpolation at the tips of the spikes, i.e. interpolation of the coarsest control points p_0 , the resulting scheme converges to a limit surface consisting of a collection of infinitesimally thin spikes joined at the centroid of p_0 .

5.3 Membrane splines and minimum energy curve networks

The final example combines the subdivision schemes for minimum energy curve networks and membrane splines. The resulting scheme produces surfaces that mimic an elastic membrane connecting a network of minimum energy curves. The input to the scheme is an initial triangulated polyhedron p_0 and a subset $n[p_0]$ of the edges of p_0 that forms a discrete polygon network. Subsequent polyhedra p_i are defined by 4-1 triangular face splits with the polygon network tags being inherited as usual.

5.3.1 Define the energy functional

If \mathbf{E}_n and \mathbf{E}_m are the energy functionals for curve networks and membranes respectively, then the functional for the combined scheme can be written as:

$$\mathbf{E}[p] = \lambda \mathbf{E}_n[n[p]] + \mathbf{E}_m[p]. \quad (10)$$

As λ increases, $\mathbf{E}[p]$ is dominated by the energy of the curve network $\mathbf{E}_n[n[p]]$. In the limit, as $\lambda \rightarrow \infty$, the surface behaves like a minimum energy curve network on $n[p]$ and an elastic membrane that interpolates the curve network on the rest of p . If $\mathbf{E}[p]$ is written in matrix form, then

$$\mathbf{E}[p] = p^T (\lambda E_n + E_m) p.$$

If the rows of E_n and E_m are viewed as defining two sets of differences, one set characterizing curve networks and the other one characterizing membranes, then these matrices can be merged into a single difference matrix E that captures the behavior of the functional as $\lambda \rightarrow \infty$: For vertices in $n[p]$, place the corresponding rows of E_n into E . For the remaining vertices of p , place the corresponding rows of E_m into E . The resulting non-symmetric difference matrix E characterizes surfaces consisting of a set of elastic membranes interpolating a minimum energy curve network.

5.3.2 Generate a catalog of simple mesh types

For vertices on the curve network, no new cataloging is necessary since the membrane energy has no effect on the curve network. As a first attempt, we distinguish vertices on the interior of the membrane by their valence. Later, we will see that this catalog is insufficient.

5.3.3 Solve for locally supported predictors

Since the catalog of simple mesh types for this combined scheme is simply the union of the two previous catalogs, the subdivision rules for minimum energy curve networks and membrane splines can be reused. The middle portion of figure 9 depicts several rounds of this combined subdivision scheme (without smoothing) applied to an initial octahedron (left) with two red curve loops lying on its surface.

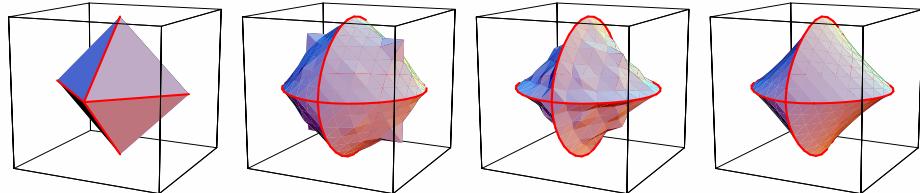


Figure 9: Base octahedron (left) basic subdivision scheme (center left), augmented subdivision scheme (center right) and subdivision with three rounds of smoothing at every level (right).

Note that the resulting surface fails to behave like a membrane away from the curve network. In particular, it bulges outward and exhibits large spikes at grid points that should actually be smooth. The reason for this bad behavior is that the subdivision matrices S_i produced by the combined scheme are a poor solution to the equation $E_i S_{i-1} = U_{i-1} E_{i-1}$. The large residuals that result cause the spikes in the center left of figure 9.

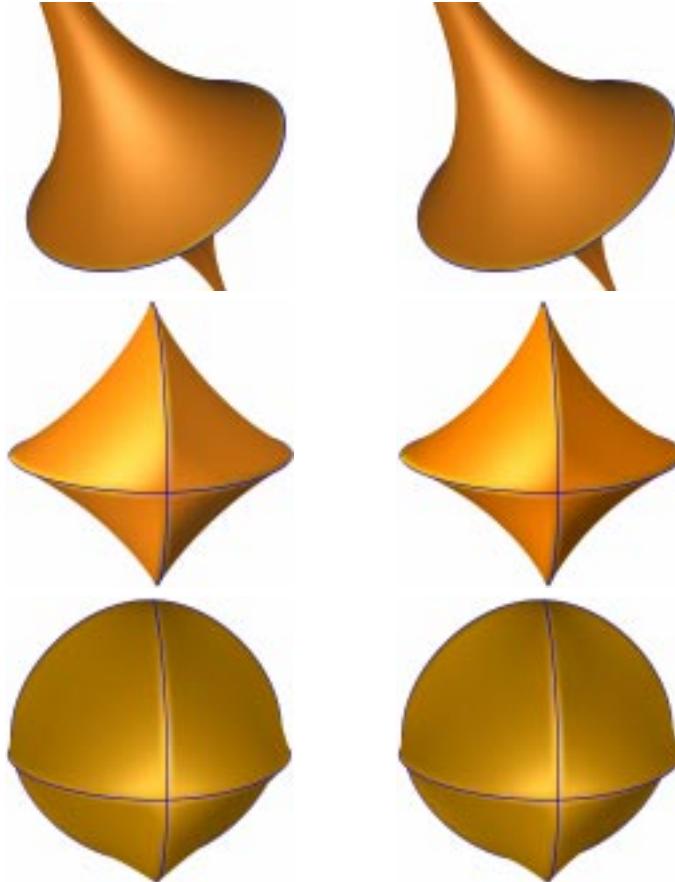


Figure 10: Several curve loops on a octahedron connected by elastic membrane subdivision (left) and exact solution (right).

There are two solutions to this problem. First, we could augment our catalog of local mesh topologies and design specialized subdivision rules for membrane vertices that are employed near the curve network. These rules can be derived using a straightforward application of our standard recipe. The center right portion of figure 9 shows an example application of these custom rules (with the same support as before) applied to the base octahedron. Note that the large spikes are absent and the surface behaves more like a membrane. However, the smaller high frequency spikes that plagued the pure membrane scheme are still present. Another obvious drawback of this solution is

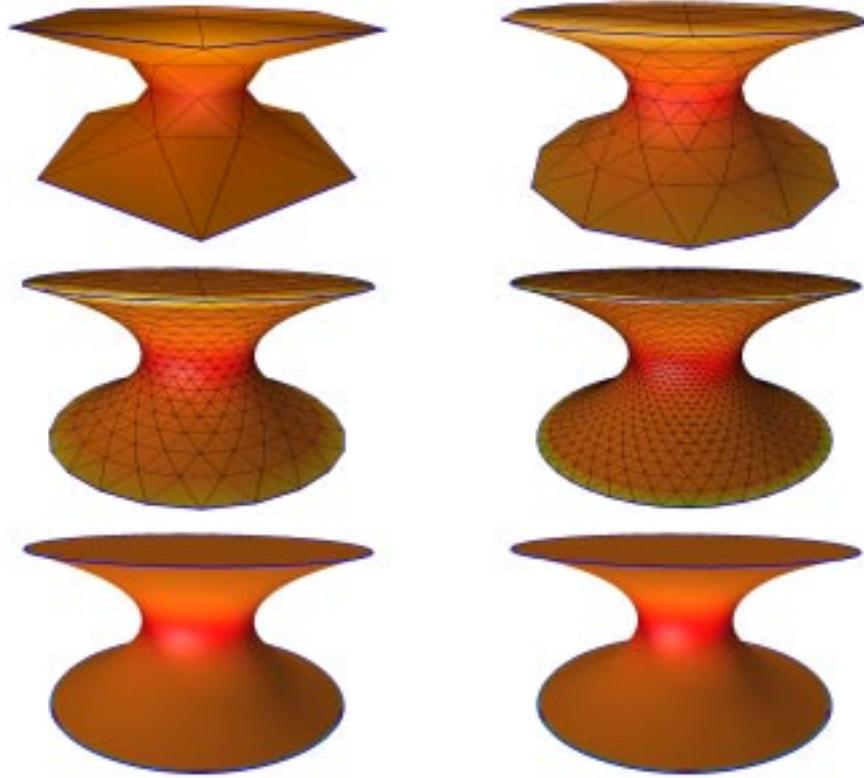


Figure 11: A catenoid defined through subdivision and the exact solution (lower right).

that the resulting catalog of custom subdivision rules would be immense. The second solution is more passable. We simply apply several rounds of the modified smoothing used in conjunction with membrane splines. The smoothing eliminates the high frequency spikes that result from the incompatibility of the membrane rules with the curve network as shown in the right of figure 9.

Figure 10 shows an example of this scheme applied to an initial octahedron with one, two and three minimum energy curve loops on its surface. The left column depicts the result of subdivision combined with three rounds of smoothing per level; the right column depicts the exact solution. Figure 11 shows a classical shape from surface modeling, a catenoid. A catenoid is an elastic membrane that joins two parallel, circular rings. The main characteristic of this shape is an inward pinching of the membrane to minimize its elastic energy. The upper left portion of the figure depicts the initial polygon p_0 for the catenoid. Vertices of p_0 on the interior of the membrane were positioned such that $E_0 p_0$ is zero for these vertices, thus forcing the membrane to be smooth on its interior. The remaining portions of the figure depict subdivision plus smoothing. The resulting smooth subdivision surface is shown in the lower left and the exact solution on the lower right.

5.4 Other applications

5.4.1 Subdivision scheme for lofted curve networks

Given a curve network, one typical problem is lofting: produce a surface that interpolates the curve network smoothly. The beauty of equation 10 is that the surface functional used in place of \mathbf{E}_m can be very general. If this functional penalizes surface bending across edges in the curve network, the resulting surfaces will tend to interpolate the curve network smoothly. The top row of figure 12 depicts an example of a smooth subdivision scheme that interpolates the red minimum energy curve network. Note that the surface has a discontinuous tangent plane across the upper red curve. The bottom row depicts the same scheme adjusted to produce a surface that lofts (i.e. smoothly interpolates) the curve network. The scheme is based on the following discrete functional for triangulated polyhedra: edges of adjacent triangles contribute energy based on their deviation from forming a parallelogram. An interesting problem for future work is developing a set of subdivision rules for a lofted version of the scheme of Loop [7].

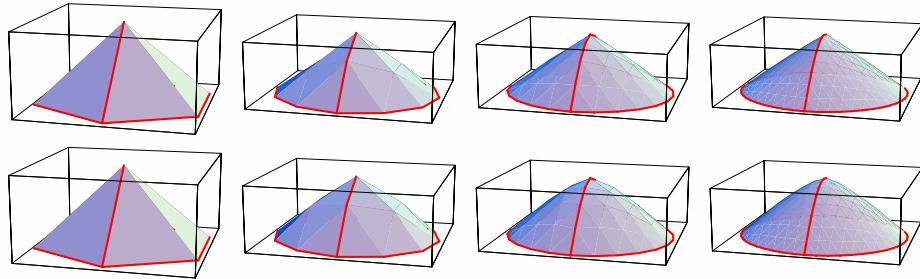


Figure 12: A subdivision surface smoothly interpolating a simple curve network

5.4.2 Subdivision scheme for fluid flow

Our last example does not involve modeling surfaces. Instead, we consider modeling vector fields. In this case, the matrices E_i are derived from the Navier-Stokes equations that characterize the behavior of a fluid. A discrete vector field $\begin{pmatrix} u_i \\ v_i \end{pmatrix}$ satisfies these equations if $E_i \begin{pmatrix} u_i \\ v_i \end{pmatrix} = 0$. Our approach is to use equation 4 to derive a set of subdivision rules S_{i-1} such that $E_i S_{i-1} \simeq U_{i-1} E_{i-1}$. The resulting subdivision scheme for vector fields has the form:

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = S_{i-1} \begin{pmatrix} u_{i-1} \\ v_{i-1} \end{pmatrix}.$$

As $i \rightarrow \infty$, the limit of these discrete fields is a continuous vector field that follows the initial vector field $\begin{pmatrix} u_0 \\ v_0 \end{pmatrix}$ and nearly satisfies the underlying Navier-Stokes equations.

Weimer and Warren [12] describe this scheme in complete detail. Figure 13 gives an example application of this scheme in three dimensions. Shown are a coarse, user-defined vector field around a cylinder (left) and two rounds of subdivision (center left and center right). The right portion of the figure depicts particles being traced through the resulting smooth flow field.

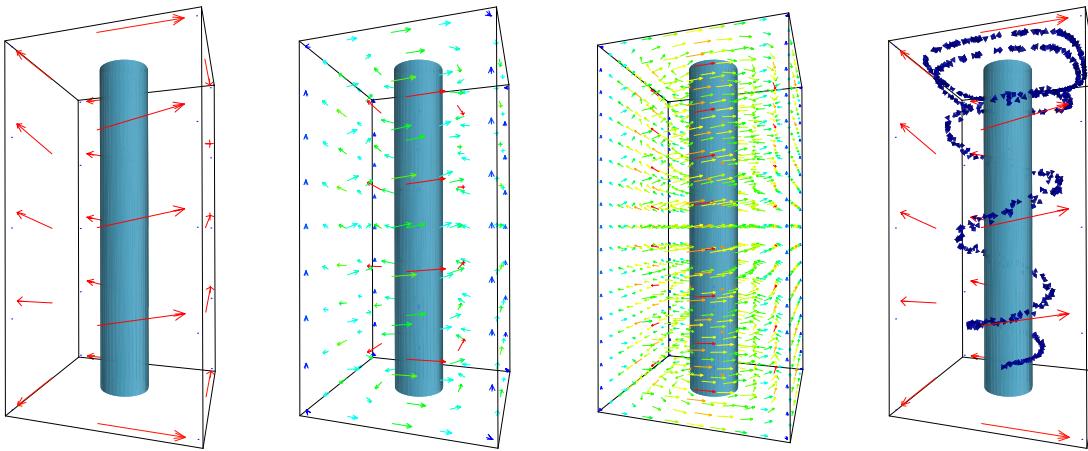


Figure 13: Subdivision for fluid flow.

6 Summary

These course notes presented a recipe for the derivation of custom subdivision schemes. Following a few simple steps, this recipes allows one to derive schemes that model a range of interesting objects described in terms of discrete energy functionals. The examples demonstrated how subdivision rules for cubic B-splines, minimum energy curve networks and membrane splines can be derived using this recipe. Finally, a combination of energy functionals was used to derive a scheme that merges minimum energy curve networks with membranes.

Acknowledgments

This work has been supported in part under National Science Foundation grants CCR-9500572 and CCR-9732344.

References

- [1] W. L. Briggs: *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, 1987.
- [2] C. deBoor, K. Hollig, S. Reimenschneider: *Box splines*. Springer Verlag, 1993.
- [3] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, W. Stuetzle: *Piecewise Smooth Surface Reconstruction*. In SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series, 1994, pp. 295-300.
- [4] L. Kobbelt: *Fairing by Finite Difference Methods*. In: M. Daehlen, T. Lyche, L. L. Schumaker: *Mathematical Methods for Curves and Surfaces II*, Vanderbilt University Press, 1998.
- [5] L. Kobbelt, S. Campagna, J. Vorsatz, H.-P. Seidel: *Interactive multi-resolution modeling on arbitrary meshes*. Proceedings of SIGGRAPH 1998. In *Computer Graphics Proceedings*, Annual Conference Series, pp. 105-114, 1998.
- [6] J. Lane, R. Reisenfeld: *A theoretical development for the computer generation and display of piecewise polynomial surfaces*. IEEE Transactions on Pattern Analysis and Machine Intelligence 2, 1, pp. 35-46, 1980.
- [7] C. T. Loop: *Smooth subdivision surfaces based on triangles*. Master's Thesis, Department of Mathematics, University of Utah, August 1987.
- [8] J.-L. Mallet: *Discrete Smooth Interpolation*. ACM Transactions on Graphics, Vol 4 No. 2, 1985, pp. 74-123.
- [9] R. Varga: *Matrix Iterative Analysis*. Academic Press, New York, 1962.
- [10] H. Weimer, J. Warren: *Subdivision schemes for thin plate splines*. Computer Graphics Forum 17, 3, pp. 303-313 & 392, 1998.
- [11] J. Warren, H. Weimer: *Subdivision schemes for variational problems*. Submitted to Computer-Aided Geometric Design, available online under <http://www.cs.rice.edu/~henrik/publications.html>.
- [12] H. Weimer, J. Warren: *Subdivision schemes for fluid flow*. In SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series, 1999.
- [13] D. Zorin, P. Schröder, W. Sweldens: *Interactive multiresolution mesh editing*, Proceedings of SIGGRAPH 1997. In *Computer Graphics Proceedings*, Annual Conference Series, pp. 259-168, 1997.