Proceedings of ALGORITMY 2012
pp. 120–131

A NEW FORM-FINDING METHOD BASED ON MEAN CURVATURE FLOW OF SURFACES*

MARTIN HÚSKA, MATEJ MEDĽA, KAROL MIKULA, PETER NOVYSEDLÁK, MARIANA REMEŠÍKOVÁ†

Abstract. In this paper, we present a new form-finding method for freeform shell structures. For a given boundary curve, we design a construction that is an approximation of the corresponding minimal surface. The surface is represented by a triangular network of curves and its initial shape can be an arbitrary smooth surface with the given boundary. In order to satisfy the minimal surface condition, we apply the mean curvature flow model to the given initial shape and we let it evolve until it reaches the equilibrium. Since it is desirable in practice to have the structure as uniform as possible, we propose a method for tangential redistribution of nodes on the individual curves. The mathematical model is discretized by a flowing control volume scheme and the paper also provides the details of the discretization technique.

Key words. form-finding, shell structure, mean curvature flow, surface evolution, flowing control volume method, tangential redistribution

AMS subject classifications. 53A10, 65D17, 65M08

1. Introduction. Shell structures are light weight constructions that have become very popular in the last decades. They provide a great flexibility in shape design which is very important in modern architecture and they are realizable with a relatively low amount of material. They consist of truss elements usually made of wood, steel or other metals and shell elements for which the typical materials are concrete, glass or plastic. These elements are assembled to large structures that are mostly used as roofs or walls of various types of buildings.

The process of the construction design begins with the design of its shape. Afterwards, a proper geometric configuration of the elements has to be found in order to meet aesthetic criteria and optimize the process of manufacturing. The structure can be viewed as a wireframe or polygonal approximation of a surface so the design procedure – the so called *form-finding* process [8] – consists in choosing the surface itself and the subsequent choosing of a set of vertices and edges in order to represent it.

In our paper, we present a method for generating structures representing minimal surfaces. These surfaces are frequently used as roof constructions because of their appealing aesthetic and practical properties – they have a three-dimensional nature which makes them more interesting than simple flat shapes and in many cases they minimize the consumption of material used for the shell elements. From the mathematical point of view, a minimal surface is a surface of zero mean curvature and its shape depends on its boundary curve. It can be obtained as the solution of the mean curvature flow (MCF) equation with the initial condition being an appropriately differentiable surface with the given boundary. Our method for finding the geometric configuration of the shell structures is based on numerical solution of the

*This work was supported by the grants APVV-0184-10 and VEGA 1/0269/09.

†Slovak University of Technology, Faculty of Civil Engineering, Radlinského 11, 81368 Bratislava, Slovakia (huska.martin2@gmail.com, matej.medla@gmail.com, mikula@math.sk, peter.novysedlak@stuba.sk, remesikova@math.sk).

MCF model. The numerical approach requires a discrete representation of the surface of exactly the same character as the desired structure – the edges of the discretization correspond to the truss elements of the construction while the polygons represent the shell elements. In order to be able to adjust the lengths of the truss elements, the basic MCF model is extended by adding a term representing tangential movement of the node points along the surface. This is important from both aesthetic and manufacturing points of view – the resulting structure should be in some sense uniform or it should contain as many equally sized elements as possible. We propose a method for tangential redistribution of the grid points that produces structures with up to 2/3 of equally sized truss elements.

2. Computing minimal surface shapes. A minimal surface is a surface whose mean curvature is everywhere zero. The existence and uniqueness of a minimal surface for a given boundary curve is still unresolved in the general case, however, a lot is known about particular cases. For example, having in mind the possible use in architecture, we can invoke the result of Nitsche [7] who proved that a regular analytic Jordan curve in $\mathbb{R}^3$ whose total curvature is at most 4π bounds a unique minimal surface which is an embedded disk. This disk is also area-minimizing.

Let us restrict ourselves to the case when the boundary is a simple closed regular curve $\Gamma\colon \mathbb{R} \to \mathbb{R}^3$. The corresponding minimal surface $S^{\Gamma}_{min}\colon \Omega \to \mathbb{R}^3$, Ω being a closed subset of $\mathbb{R}^2$ homeomorphic to a disk, can be obtained as the solution at $t = \infty$ of the mean curvature flow equation

$$\partial_t S = 2HN \tag{2.1}$$

solved in $\mathrm{int}(\Omega)$ where $S\colon \Omega \times \langle 0, \infty) \to \mathbb{R}^3$, $S(\cdot, \cdot, t)$ is an embedding of Ω in R^3, $S(u, v, 0)$ is a smooth regular surface with the boundary given by Γ, $H\colon \Omega \times \langle 0, \infty) \to \mathbb{R}$ and $H(u, v, t)$ is the mean curvature of $S(\cdot, \cdot, t)$ in $S(u, v, t)$, $N\colon \Omega \times \langle 0, \infty) \to \mathbb{R}^3$ and $N(u, v, t)$ is the outward unit normal to $S(\cdot, \cdot, t)$ in $S(u, v, t)$. The boundary condition is given by

$$\partial_t S = 0 \quad \text{on } \partial\Omega. \tag{2.2}$$

Alternatively, the equation (2.1) can be written in terms of the Laplace-Beltrami operator [2] (the symbol Δ stands for the Laplace-Beltrami operator throughout this paper)

$$\partial_t S = \Delta S. \tag{2.3}$$

3. Tangential movement of points on the surface. The equation (2.1) can be seen as a shape evolution equation since the movement in normal direction affects the shape of the evolving surface. However, besides this movement in the surrounding space, there is also an intrinsic evolution that becomes visible once we start following a selected set of (more than one) points on the surface. In such case we might observe changes in configuration of these points – it might be a change of their distances or the ratio of the distances, a change of area in case we follow a whole neighborhood of some point etc. Having in mind a particular application, this evolution of parametrization might be undesirable. In that case we can extend our model and allow the points to move along the surface (without any influence on its geometry) and adjust the parametrization according to our needs. The corresponding equation reads

$$\partial_t S = 2HN + \alpha T \tag{3.1}$$

where $\alpha\colon \Omega \times \langle 0,\infty) \to \mathbb{R}$, $T\colon \Omega \times \langle 0,\infty) \to \mathbb{R}^3$ and $T(\cdot,\cdot,t)$ is a unit vector field on the surface $S(\cdot,\cdot,t)$. Both α and T can be chosen in various ways according to our purposes. The two-dimensionality of the tangent space in each point of the surface gives larger possibilities than in the case of curves [3, 4, 5]. The topic has still not been fully examined though some work has already been done [1, 6].

4. Curves on an evolving surface. The particularity of our situation lies in the fact that we are evolving a surface (in order to be able to guarantee the minimality of the resulting shape) but what we want to obtain is actually a network of one-dimensional line segments. The lengths of these segments are of particular interest from the manufacturing point of view as it is convenient to have as many equally sized elements as possible. One way to redistribute a one-dimensional measure on a two-dimensional manifold is to focus on a selected curve on the surface and redistribute the points on this curve in the course of the surface evolution.

In order to do this, we extend the ideas of Mikula and Ševčovič [3, 4] and Mikula and Urbán [5]. Having an evolving regular space curve $\gamma\colon \mathbb{R} \times \langle 0,\infty) \to \mathbb{R}^3$, we can express its motion as

$$\partial_t \gamma = \beta_1 N_1 + \beta_2 N_2 + aT \tag{4.1}$$

where $\beta_1, \beta_2\colon \Omega \times \langle 0,\infty) \to \mathbb{R}$ are the scalar normal velocities, $a\colon \Omega \times \langle 0,\infty) \to \mathbb{R}$ is the scalar tangential velocity, $T\colon \mathbb{R} \times \langle 0,\infty) \to \mathbb{R}^3$ is the (evolving) unit tangent vector to the curve and $N_1, N_2\colon \mathbb{R} \times \langle 0,\infty) \to \mathbb{R}^3$ form an orthonormal basis of the plane normal to T. There are many possibilities how to choose N_1 and N_2. The Frenet frame cannot be used in this case since we have no guarantee that the curvature vector is non-zero. However, for a regular curve, a non-singular basis which is also smooth along the curve can be easily constructed by rotating the standard basis $\{e_1, e_2, e_3\}$ of $\mathbb{R}^3$ around the axis given by the vector

$$r = T \times e_1 \tag{4.2}$$

by the angle given by

$$\cos\theta = T \cdot e_1. \tag{4.3}$$

That means if $A_{r,\theta}$ is the corresponding rotation matrix, then

$$N1 = A_{r,\theta} e_2, \quad N2 = A_{r,\theta} e_3. \tag{4.4}$$

Now, let $l\colon \mathbb{R} \times \langle 0,\infty) \to \mathbb{R}$ represent the local length of γ ($l(z,t) = ||\partial_z \gamma(z,t)||$) and let $L\colon \mathbb{R} \times \langle 0,\infty) \to \mathbb{R}$ be its global length. The evolution of these quantities is given by [5]

$$\partial_t l = l\partial_s a - l(\beta_1 k_1 + \beta_2 k_2) \tag{4.5}$$

$$\partial_t L = -\int_0^L (\beta_1 k_1 + \beta_2 k_2)\,\mathrm{d}s + a(0,\cdot) - a(L,\cdot) \tag{4.6}$$

where s represents the arc-length of γ and k_1 and k_2 are the projections of the curvature vector $\kappa = \partial_s T$ to N_1 and N_2 ($k_1 = \kappa \cdot N_1$, $k_2 = \kappa \cdot N_2$).

Now we can impose some conditions on the evolution of parametrization of γ. An asymptotically uniform parametrization can be expressed as

$$\lim_{t\to\infty} \frac{l}{L} = 1$$

which can be achieved e.g. by solving the equation

$$\partial_t \left(\frac{l}{L} \right) = \omega \left(1 - \frac{l}{L} \right) \tag{4.7}$$

where $\omega \in \mathbb{R}$ is a parameter regulating the speed of the evolution. This, together with (4.5) and (4.6), determines the arc-length derivative of a

$$\partial_s a = \beta_1 k_1 + \beta_2 k_2 - \frac{1}{L} \int_0^L (\beta_1 k_1 + \beta_2 k_2) \, \mathrm{d}s + \omega \left(\frac{L}{l} - 1 \right) \tag{4.8}$$

where, for simplicity, we suppose that $a(0, \cdot) = a(L, \cdot) = 0.0$ since this will be the only setting that we will use in our experiments.

We might also want to prescribe the length L_∞ of the curve for $t \to \infty$. In that case we have $\partial_t L_\infty = 0$ and thus we get

$$\partial_s a = \beta_1 k_1 + \beta_2 k_2 + \omega \left(\frac{L_\infty}{l} - 1 \right). \tag{4.9}$$

In this case, we do not assume any specific choice of $a(0, \cdot)$.

Now we can apply these ideas to a regular curve γ on the evolving surface S. Such a curve can be obtained as a push-forward of a regular plane curve $\gamma_p \colon \langle 0, 1 \rangle \to \Omega$ along the map S. Its points are evolving according to (2.1) and thus, decomposing the mean curvature vector to the directions N_1, N_2, we have

$$\beta_1 = 2HN \cdot N_1, \quad \beta_2 = 2HN \cdot N_2. \tag{4.10}$$

We can directly use (4.8) or (4.9) in order to redistribute the points on this curve as the surface is evolving. To conserve the smoothness of the surface, we can determine the tangential velocity in the neighborhood of γ e.g. by applying a mollifier to the computed vector field on γ.

Finally, coming back to our architectural application, we have to resolve the situation with a whole network of truss segments. The smooth analogue to such structures are networks of smooth curves. Let us suppose that we have a structure composed of regular curves γ_k, $k = 1 \dots n_c$, with possible intersections but with the conditions that each node of the network is an intersection point of exactly two curves, each pair of curves has at most one common point and the tangent vectors of two intersecting curves at their intersection point are linearly independent. Such networks can often be found in real structures as their subsets; we speak about subsets since, in general, it is not possible to expect that all segments will satisfy some specific condition (i.e. that they will have a given length). Let P_{ijt} be the intersection point of the curves $\gamma_i(\cdot, t)$ and $\gamma_j(\cdot, t)$, i.e. $P_{ijt} = S(u_{ij}, v_{ij}, t) = \gamma_i(z_{ij}, t) = \gamma_j(z_{ji}, t)$. Let the scalar tangential velocities (computed according to (4.8) or (4.9)) at $\gamma_i(z_{ij}, t)$ and $\gamma_j(z_{ji}, t)$ be $a_i(z_{ij}, t)$ and $a_j(z_{ji}, t)$. The corresponding unit tangential vectors are $T_i(z_{ij}, t)$ and $T_j(z_{ji}, t)$. Then we can set the tangential velocity at P_{ijt} to

$$\alpha(u_{ij}, v_{ij}, t) T(u_{ij}, v_{ij}, t) = \frac{a_i(z_{ij}, t) T_i(z_{ij}, t) + a_j(z_{ji}, t) T_j(z_{ji}, t)}{2} \tag{4.11}$$

Since $T_i(z_{ij}, t)$ and $T_j(z_{ji}, t)$ are linearly independent, the left hand side of this equation will be zero only if, at some time point, all of the following conditions are satisfied.

1. $S(\cdot,\cdot,t)$ is a minimal surface, i.e. the normal velocities in (4.1) (and hence in (4.8) and (4.9)) are zero.
2. Both curves $\gamma_i(\cdot,t)$ and $\gamma_j(\cdot,t)$ are uniformly parametrized and, if we use (4.9), they have the prescribed global length. In this case, if also 1 is satisfied, we get $\partial_s a_i(\cdot,t)=0$, $\partial_s a_j(\cdot,t)=0$.
3. $a_i(0,t)=0$, $a_j(0,t)=0$.

These three conditions, if they are satisfied for all curves in the network, characterize the equilibrium state of the whole system.

5. Numerical approximation.

5.1. The flowing control volume method. The surface evolution model that we are going to discretize reads as follows.

$$\partial_t S = \Delta S + \alpha T \tag{5.1}$$

The time discretization is semi-implicit, i.e.

$$\frac{S^n - S^{n-1}}{\tau} = \Delta_{n-1} S^n + \alpha^{n-1} T^{n-1} \tag{5.2}$$

where τ is the time discretization step and $n>0$ denotes the time level. Δ_{n-1} is the Laplace-Beltrami operator corresponding to the surface S^{n-1} (this makes sense since S^n and S^{n-1} are defined over the same domain and there is a one-to-one correspondence between their points).

The key part of the numerical approximation of (5.1) is the discretization of the Laplace-Beltrami operator. We can apply the flowing control volume scheme [3, 9] that we adjust to our situation.

The flowing control volume method belongs to the family of finite volume schemes and as such it is based on polygonal representation of the surface. We suppose that the vertices S_i^n of the polygons are points of the unknown surface $S(\cdot,\cdot,t^n)$. These vertices are moving together with the evolving surface which means that the whole discretization mesh is evolving or "flowing" – that is how the method took its name. We present the version of the method for the most simple case when the surface is represented by a triangular mesh. In order not to get lost in a labyrinth of indices, we omit the time index n in some parts of the explanation and include it only when necessary – though the position of the mesh vertices depends on time, its structure does not. We also use local indexing of the mesh elements since it is sufficient to explain the idea.

The first step is the construction of a co-volume mesh (Fig. 5.1). We will describe the procedure for a chosen inner node S_i. Let us suppose that this node is the common vertex of m mesh triangles $\mathcal{T}_1,\dots,\mathcal{T}_m$. Then it is also the common vertex of m edges $h_1,\dots,h_m$, where h_p connects S_i with S_{i_p}. Let B_p be the barycenter of $\mathcal{T}_p$ and C_p the center of h_p, $p=1\dots m$. The co-volume V_i corresponding to S_i is constructed as the union of the triangles $\mathcal{V}_{p,1}=S_iC_pB_p$ and $\mathcal{V}_{p,2}=S_iB_pC_{p+1}$ for $p=1\dots m$ where we set $C_{m+1}=C_1$. This construction is correct in the sense that the union of all these co-volumes covers the (triangulated) surface and their only intersections are their boundaries. Our approximation will further use the outward unit normals $\mu_{p,1}$, $\mu_{p,2}$, $\mu_{p,3}$ to the sides of $\mathcal{T}_p$ in its plane and the outward unit normals $\nu_{p,1}$, $\nu_{p,2}$ to the co-volume edges $\sigma_{p,1}=C_pB_p$, $\sigma_{p,2}=B_pC_{p+1}$ in the plane of $\mathcal{T}_p$. For the boundary nodes, the only difference is that the node with m neighboring mesh triangles has $m+1$ neighboring edges $h_1,\dots,h_{m+1}$.

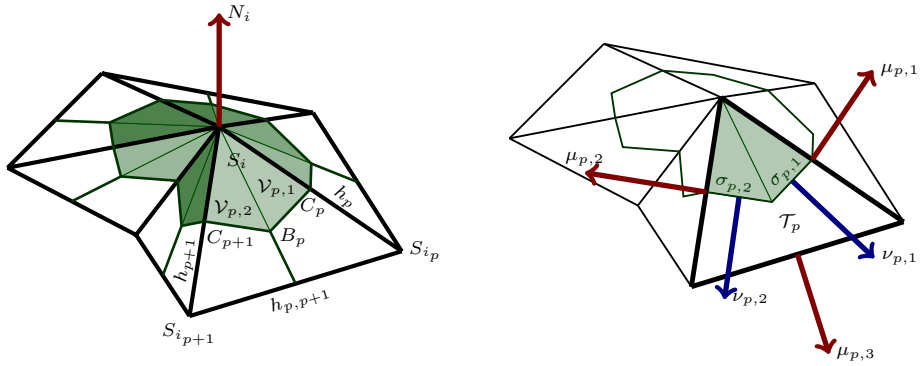

FIG. 5.1. *The discretization mesh.*

Integrating (5.2) over V_i^{n-1}, we get

$$\int_{V_i^{n-1}} \frac{S^n - S^{n-1}}{\tau} \,\mathrm{d}x = \int_{V_i^{n-1}} \Delta_{n-1} S^n \,\mathrm{d}x + \int_{V_i^{n-1}} \alpha^{n-1} T^{n-1} \,\mathrm{d}x. \tag{5.3}$$

For any inner control volume V_i, the Laplace-Beltrami operator term can be rewritten as

$$\int_{V_i} \Delta S \,\mathrm{d}x = \int_{\partial V_i} \nabla S \cdot \nu_i \,\mathrm{d}y = \sum_{p=1}^{m} \left(\int_{\sigma_{p,1}} \nabla S \cdot \nu_{p,1} \,\mathrm{d}y + \int_{\sigma_{p,2}} \nabla S \cdot \nu_{p,2} \,\mathrm{d}y \right) \tag{5.4}$$

where ∇ denotes the tangential gradient. If we assume that $S(\cdot,\cdot,t)$ is linear on $\mathcal{T}_p^n$, we can write

$$D_p := (\nabla S)\restriction_{\mathcal{T}_p} = \frac{1}{|\mathcal{T}_p|} \int_{\mathcal{T}_p} \nabla S \,\mathrm{d}x = \frac{1}{|\mathcal{T}_p|} \int_{\partial \mathcal{T}_p} S \otimes \mu_p \,\mathrm{d}y \tag{5.5}$$

and further, taking in account the semi-implicit time discretization

$$D_p^n = \frac{1}{|\mathcal{T}_p^{n-1}|} \left(|h_p^{n-1}| \bar{S}_{i,i_p}^n \otimes \mu_{p,1}^{n-1} + |h_{p+1}^{n-1}| \bar{S}_{i,i_{p+1}}^n \otimes \mu_{p,2}^{n-1} + |h_{p,p+1}^{n-1}| \bar{S}_{i_p,i_{p+1}}^n \otimes \mu_{p,3}^{n-1} \right). \tag{5.6}$$

where

$$\bar{S}_{i,i_p}^n = \frac{S_i^n + S_{i_p}^n}{2},\ \bar{S}_{i,i_{p+1}}^n = \frac{S_i^n + S_{i_{p+1}}^n}{2},\ \bar{S}_{i_p,i_{p+1}}^n = \frac{S_{i_p}^n + S_{i_{p+1}}^n}{2}$$

Finally, we can write the fully discrete model for the node S_i. At this place, we include the full indexing in order to provide a correct presentation of the resulting linear system.

$$S_i^n - \frac{\tau}{|V_i^{n-1}|} \sum_{p=1}^{m_i} \left(|\sigma_{i,p,1}^{n-1}| D_{i,p}^n \cdot \nu_{i,p,1}^{n-1} + |\sigma_{i,p,2}^{n-1}| D_{i,p}^n \cdot \nu_{i,p,2}^{n-1} \right) = S_i^{n-1} + \tau \alpha_i^{n-1} T_i^{n-1}. \tag{5.7}$$

5.2. Approximation of the tangential velocity. We will describe the approximation for one selected curve γ on the surface. When the surface is approximated by a polygonal mesh, the approximated curve $\hat{\gamma}$ is a polygonal line with vertices $S_{i_0}, S_{i_1}, \ldots, S_{i_p}$ where p is the number of line segments of the curve. Let d_j, $j = 1 \ldots p$, be the length of the segment $S_{i_{j-1}} S_{i_j}$. The unit tangential vector to γ in S_{i_j}, $j = 1, \ldots, p-1$, is approximated by

$$T_j = \frac{1}{2}\left(\frac{S_{i_{j+1}} - S_{i_j}}{d_{j+1}} + \frac{S_{i_j} - S_{i_{j-1}}}{d_j}\right). \tag{5.8}$$

and for the first and the last point of the curve we set

$$T_0 = \frac{S_{i_1} - S_{i_0}}{d_1}, \quad T_p = \frac{S_{i_p} - S_{i_{p-1}}}{d_p} \tag{5.9}$$

Next, we approximate the curvature vector in the inner nodes S_{i_j} by

$$\kappa_j = \frac{2}{d_{j+1} + d_j}\left(\frac{S_{i_{j+1}} - S_{i_j}}{d_{j+1}} - \frac{S_{i_j} - S_{i_{j-1}}}{d_j}\right) \tag{5.10}$$

and in the boundary points as

$$\kappa_0 = \frac{T_1 - T_0}{d_1}, \quad \kappa_p = \frac{T_p - T_{p-1}}{d_p}. \tag{5.11}$$

Having computed κ_j and the normals $N_{1,j}$, $N_{2,j}$ (according to (4.2)–(4.4)), we can compute the projections $k_{1,j} = \kappa_j \cdot N_{1,j}$, $k_{2,j} = \kappa_j \cdot N_{2,j}$. Further, we need the scalar normal velocities $\beta_{1,j}$, $\beta_{2,j}$ which (according to (4.10)) means to approximate the mean curvature vector. Since $2HN = \Delta S$, we can recall (5.7) where we have the integral $\int_{V_{i_j}} \Delta S \, \mathrm{d}x$ approximated by the term

$$I_j = \sum_{k=1}^{m_{i_j}} \left(|\sigma_{i_j,k,1}| D_{i_j,k} \cdot \nu_{i_j,k,1} + |\sigma_{i_j,k,2}| D_{i_j,k} \cdot \nu_{i_j,k,2}\right). \tag{5.12}$$

The velocities $\beta_{1,j}$ and $\beta_{2,j}$ are then obtained as

$$\beta_{1,j} = \frac{I_j \cdot N_{1,j}}{|V_{i_j}|}, \quad \beta_{2,j} = \frac{I_j \cdot N_{2,j}}{|V_{i_j}|} \tag{5.13}$$

The length $\hat{L}$ of the polygonal curve $\hat{\gamma}$ is simply computed as

$$\hat{L} = \sum_{j=1}^{p} d_j. \tag{5.14}$$

Assuming that $\gamma, \hat{\gamma} \colon \langle 0, 1\rangle \to \mathbb{R}^3$ and $S_{i_j} = \hat{\gamma}(j/p)$, we can approximate the local length of γ in S_{i_j} by

$$l_j = \left\| \frac{S_{i_j} - S_{i_{j-1}}}{\frac{1}{p}} \right\| = p d_j \tag{5.15}$$

for $j = 1 \dots p$. Finally, we take the approximation

$$\int_0^L (\beta_1 k_1 + \beta_2 k_2)\,\mathrm{d}s \approx \sum_{j=1}^{p} \left(\frac{\beta_{1,j-1}k_{1,j-1} + \beta_{1,j}k_{1,j}}{2} + \frac{\beta_{2,j-1}k_{2,j-1} + \beta_{2,j}k_{2,j}}{2} \right) d_j =: \bar{\beta}. \tag{5.16}$$

Now, using (5.8)–(5.16), we get the discrete representation of (4.8)

$$a_j^n = a_{j-1}^n + d_j^n(\beta_{1,j}^n k_{1,j}^n + \beta_{2,j}^n k_{2,j}^n) - \frac{d_j^n}{\hat{L}^n}\bar{\beta}^n + \omega\left(\frac{\hat{L}^n}{p} - d_j^n\right) \tag{5.17}$$

and of (4.9)

$$a_j^n = a_{j-1}^n + d_j^n(\beta_{1,j}^n k_{1,j}^n + \beta_{2,j}^n k_{2,j}^n) + \omega\left(\frac{L_\infty}{p} - d_j^n\right) \tag{5.18}$$

for $j = 1 \dots p-1$.

6. Results.

6.1. Experiment 1. In the first experiment, the surface $S(\cdot, t)$ was defined on the quadrilateral Ω with vertices $A_\Omega = (0,0)$, $B_\Omega = (22,0)$, $C_\Omega = (30,25)$, $D_\Omega = (0,25)$. The boundary Γ of the surface was given as the union of four curves, i.e. $\Gamma = \Gamma_W \cup \Gamma_E \cup \Gamma_N \cup \Gamma_S$, where

$$\begin{array}{lcll} \Gamma_W(z) & = & (0.0128z^2 - 0.32z, z, 0.0512z^2 - 1.28z), & z \in \langle 0, 25\rangle \\ \Gamma_E(z) & = & (0.32z + 22.0, z, 0.0384z^2 - 0.96z), & z \in \langle 0, 25\rangle \\ \Gamma_N(z) & = & (z, 25.0, 0.0), & z \in \langle 0, 30\rangle \\ \Gamma_S(z) & = & (z, 0.0, 0.0), & z \in \langle 0, 22\rangle. \end{array}$$

The surface was approximated by a mesh containing 242 triangles, the topology of the grid can be seen in Fig. 6.1. We set $S_i^0 = (0,0,0)$ for all inner nodes of the mesh and the boundary nodes were placed on Γ.

We tested both types of redistribution of the grid points. Looking at the mesh topology, we can identify three sets of curves starting and ending in boundary points – the "horizontal", "vertical" and "diagonal" curves. These are the curves that can be considered for the redistribution of grid points described in Sec. 4. Since in practice the truss element lengths have to be indeed equal in order to optimize the manufacturing process (the allowed tolerance is about 1‰ of the element length), we set $\omega = 800$ in either (4.8) or (4.9). This parameter controls the speed of the redistribution and the higher it is the quicker we approach the desired distribution of the mesh points. The time step was $\tau = 6.25 \times 10^{-3}$, and we stopped the computation after 800 time steps. The linear system (5.7) was solved by the SOR method.

First, we required the uniform distribution of points on the individual curves without any further conditions. Generally, it is not possible to impose this condition on all curves in the network and also the redistribution model described in Sec. 4 does not admit more than two curves intersecting at one point. Therefore we perform the redistribution only on the horizontal and vertical curves (Fig. 6.3). We apply the scheme (5.17). The resulting surface is displayed in Fig. 6.2 and 6.3.

Next, we tested the second type of redistribution by (4.9). This approach can be used if we want to maximize the number of equally sized truss elements in order to minimize the manufacturing expenses – by setting the length of a polygonal curve, we

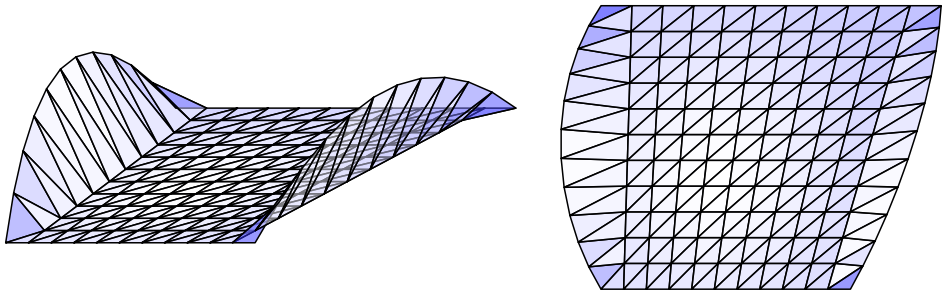

FIG. 6.1. *Two different views of the initial condition for Experiment 1. Left, an axonometric projection, right, a perspective projection.*

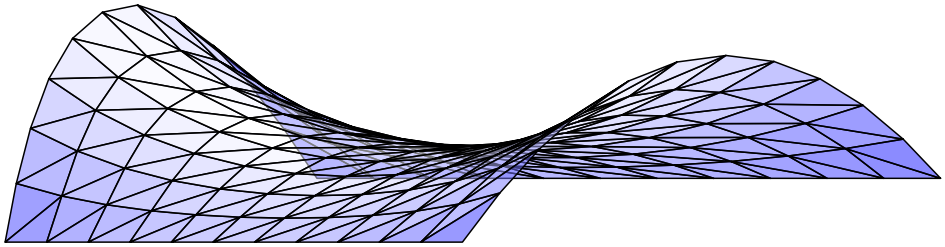

FIG. 6.2. *Experiment 1 – asymptotically uniform redistribution of points on the individual curves. The figure shows the computed triangulated minimal surface in an axonometric projection.*

actually set the length its segments (assuming we want to end up with a uniformly discretized curve). The prescribed boundary and the condition of minimality make it impossible to require the same length for all horizontal and vertical curves considered in the previous case. Therefore we impose the fixed length condition only on the part of the curve between its second and last but one point – the points in this part are allowed to move freely on the surface. We set $L_\infty = 22.0$ and $a_0^n = 0.0$ for each curve. The question is how to choose the velocity a_1^n. We can directly use (5.18) but in order to achieve an aesthetic result, we decided to apply the condition of symmetry and set

$$a_1^n = d_1^n(\beta_{1,1}^n k_{1,1}^n + \beta_{2,1}^n k_{2,1}^n) + \omega\left(d_p^n - d_1^n\right). \tag{6.1}$$

This means that at the end of the evolution (for $t \to \infty$) the first and the last segment of the polygonal curve will have the same length. The result can be seen in Fig. 6.4 and 6.5. We achieved 216 equally sized truss elements out of 385 total, which is 56.1%. For the number of segments tending to infinity, this ratio would approach $\frac{2}{3} = 66.67\%$ (two out of three sides of almost all triangles have the prescribed length). However, we can observe that even though the manufacturing cost would be lower than in the first case, for this particular configuration of the structure we obtain a more aesthetic result by applying the redistribution given by (4.8).

6.2. Experiment 2. Contrarily to Experiment 1, Fig. 6.6 and 6.7 show a case where we obtained a very aesthetic result by applying the second type or tangential redistribution. In this case we set

$$\begin{aligned}
\Gamma_W(z) &= (0.0, z, -(z-0.5)^2 + 0.25), && z \in \langle 0, 1\rangle \\
\Gamma_E(z) &= (1.0, z, -(z-0.5)^2 + 0.25), && z \in \langle 0, 1\rangle \\
\Gamma_N(z) &= (z, 1.0, 0.0), && z \in \langle 0, 1\rangle \\
\Gamma_S(z) &= (z, 0.0, 0.0), && z \in \langle 0, 1\rangle.
\end{aligned}$$

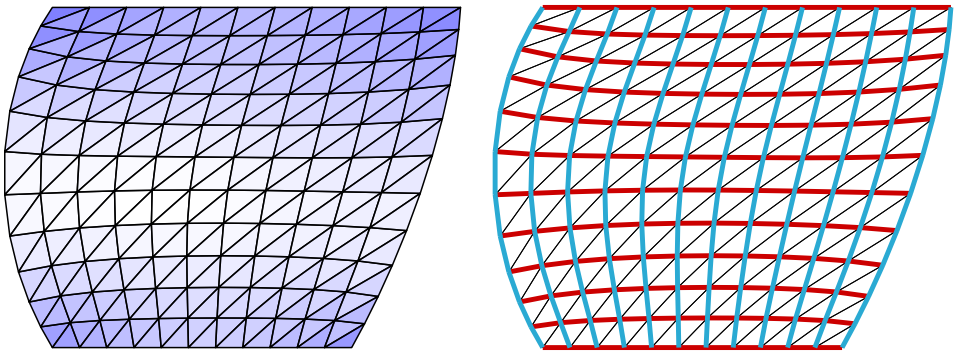

FIG. 6.3. *Experiment 1 – asymptotically uniform redistribution of points on the individual curves. The figure shows the computed triangulated minimal surface in a perspective projection, top view. The curves included in the redistribution process are highlighted on the right picture.*

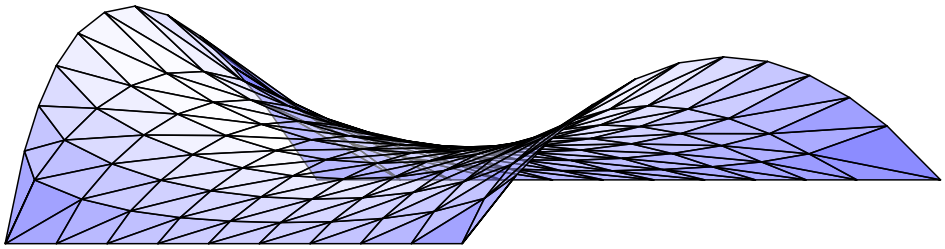

FIG. 6.4. *Experiment 1 – redistribution with prescribed length of the curves. The figure shows the computed triangulated minimal surface in an axonometric projection.*

The mesh was dually hexagonal (each inner node has six neighboring triangles) just as in Experiment 1 but it had a different configuration (Fig. 6.7). In this case we applied the redistribution method to the set of "diagonal" curves and to two boundary curves Γ_W and Γ_E. Instead of setting L_∞, we set the length of a single truss segment to $d_\infty = 0.101$ for the inner curves and $d_\infty = 0.12$ for the boundary curves. All other parameters as well as a_0^n and a_1^n for each curve were set as in Experiment 1. The resulting structure had 180 equally sized truss elements out of 401, i.e. 44.89%. The asymptotic ratio is 66.67% similarly to Experiment 1.

7. Concluding remarks. We described a method for generating triangular approximations of minimal surfaces. We also suggested two possible ways of redistributing the mesh points and illustrated their performance and possible advantages and disadvantages. To conclude, we will mention some possible extensions.

First, the method that we described provides us with (approximate) surfaces of zero mean curvature. However, it can be straightforwardly extended to surfaces of constant curvature or, in fact, any prescribed curvature.

Second, the boundary of the surface need not be a Jordan curve, the method could be applied to any reasonable boundary. We only have to keep in mind that the minimal surface for a given boundary is not always unique so the result will depend on the initial condition. It is also possible to apply the method to closed surfaces evolving by mean curvature flow though this is out of the scope of architectural applications.

Third, it is not complicated to adjust the numerical scheme to the case of a

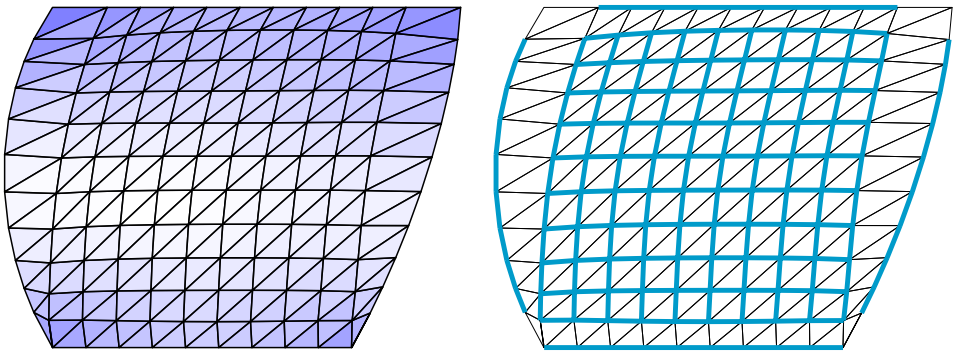

FIG. 6.5. *Experiment 1 – redistribution with prescribed length of the curves. The figure shows the computed triangulated minimal surface in a perspective projection, top view. The equally sized truss elements are highlighted in the right picture.*

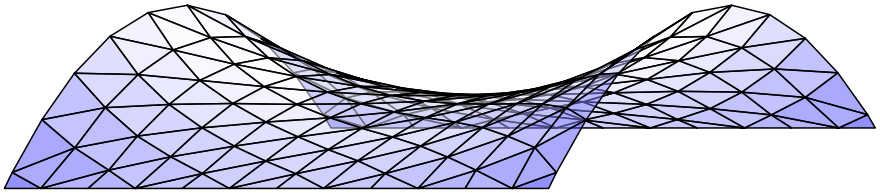

FIG. 6.6. *Experiment 2 – the computed triangulated minimal surface in an axonometric projection.*

general polygonal representation of the surface. In fact, the only problematic thing is that the vertices of the polygons $\mathcal{T}_p$ need not be coplanar in a general situation. This difficulty can be overcome e.g. by triangulating $\mathcal{T}_p$.

REFERENCES

[1] J. W. BARRETT, H. GARCKE, R. NURNBERG, *On the parametric finite element approximation of evolving hypersurfaces in R^3*, Journal of Computational Physics 227 (2008), pp.4281–4307.

[2] G. DZIUK, *An algorithm for evolutionary surfaces*, Numerische Mathematik 58(1) (1991), pp. 603–611.

[3] K. MIKULA, D. ŠEVČOVIČ, *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM Journal on Applied Mathematics, 61 (5) (2001), pp. 1473–1501.

[4] K. MIKULA, D. ŠEVČOVIČ, *A direct method for solving an anisotropic mean curvature flow of planar curve with an external force*, Mathematical Methods in Applied Sciences, 27(13) (2004), pp. 1545–1565.

[5] K. MIKULA, J. URBÁN, *3D curve evolution algorithm with tangential redistribution for a fully automatic finding of an ideal camera path in virtual colonoscopy*, Proceedings of the Third International Conference on Scale Space Methods and Variational Methods in Computer Vision 2011, Lecture Notes in Computer Science Series, Springer, 2011.

[6] S. MORIGI, *Geometric surface evolution with tangential contribution*, J. Computational Applied Mathematics, 233(5) (2010), pp. 1277–1287.

[7] J. C. C. NITSCHE, *A new uniqueness theorem for minimal surfaces*, Arch. Rat. Mech. Anal., 52 (1973), pp. 319–329.

[8] B. H. V. TOPPING, P. IVANYI, *Computer Aided Design of Cable Membrane Structures*, Saxe-Coburg Publications on Computational Engineering (2008).

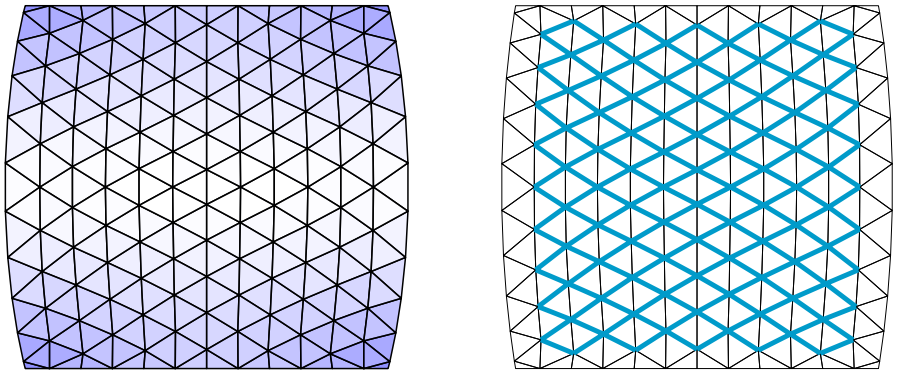

FIG. 6.7. *Experiment 2 – the computed triangulated minimal surface in a perspective projection, top view. The equally sized truss elements are highlighted on the right picture.*

[9] M. TUNEGA, R. ČUNDERLÍK, K. MIKULA, *Filtrácia geodetických dát na povrchu Zeme pomocou nelineárnych difúznych rovníc (Filtration of geodetic data on the Earth surface using nonlinear diffusion equations)*, Proceedings of the conference Mathematics, Geometry and their Applications – MAGIA 2009, Kočovce, Slovakia (2009).