Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **200X**

doi:10.3934/xx.xx.xx.xx

pp. **X–XX**

TRUSS STRUCTURE DESIGN USING A LENGTH-ORIENTED SURFACE REMESHING TECHNIQUE

Karol Mikula, Mariana Remešíková

Department of Mathematics and Descriptive Geometry
Faculty of Civil Engineering, Slovak University of Technology
Radlinského 11, 81368 Bratislava, Slovakia

Peter Novysedlák

Department of Steel and Timber Constructions
Faculty of Civil Engineering, Slovak University of Technology
Radlinského 11, 81368 Bratislava, Slovakia

(Communicated by the associate editor name)

Abstract. We present a method that can be used for designing truss structures representing either minimal surface shapes or general free-form shapes. The structures are designed so that they meet some specific criteria concerning their aesthetic properties and especially the lengths of the truss elements. We explain a technique for tangential redistribution of points on evolving surfaces that allows to obtain equally sized truss elements in selected subsets of the structure. This technique is applied to surfaces evolving by their mean curvature yielding constructions that approximate minimal surface shapes. Afterwards, we show how to remesh static free-form surfaces.

1. **Introduction.** Truss structures have become an important element in modern architecture. They appear either as stand-alone constructions or as a part of combined structures such as thin-shell structures. Very often, their role is not purely mechanical but they also contribute to the visual impression of the whole construction. Designing them requires meeting a number of criteria such as mechanical stability, good aesthetic properties and optimal cost of manufacturing [16].

Algorithmic design of truss structures has become one of the most interesting challenges of modern architecture. Some of the most used methods are force density method [15] and its various extensions and modifications, surface stress density method [5] or dynamic relaxation method [1].

Our paper proposes a novel method for triangular truss structure design that provides results of a good aesthetic quality and, moreover, it allows to optimize the manufacturing expenses. The main idea lies in viewing the truss structure as a triangulation of a surface and using a surface remeshing technique. From the manufacturing point of view, the main object of interest are the lengths of the trusses. In order to minimize the costs, the construction should contain as many equally sized truss elements as possible. Moreover, the designer should have the

2010 *Mathematics Subject Classification.* Primary: 65M08, 65M50; Secondary: 53A05.

Key words and phrases. truss structure design, surface remeshing, surface evolution, minimal surfaces, tangential redistribution.

This work was supported by the grants APVV-0184-10 and VEGA 1/1137/12.

possibility to set the lengths of the trusses to minimize the amount of waste. Though some related work has been done [10, 12], algorithms focused on the lengths of truss elements are not numerous. Our method is based on the length-oriented tangential redistribution of points on evolving surfaces in $\mathbb{R}^3$ [3, 6] that allows to control distances between selected points on a surface as it evolves by a given vector field. This is achieved by letting the points move along the surface by an appropriately chosen tangential velocity field. After a simple adjustment, this technique can be used also for remeshing of discretized surfaces that do not evolve.

Besides that, we show how we can obtain optimized truss structures representing minimal surfaces given only the boundary curve of the construction. So far, there are only a few algorithms able to produce structures that provably consistently approximate minimal surfaces [2, 13]. In our work, we apply the mean curvature flow model enriched with a tangential movement term. To our knowledge, this procedure has not been previously applied to the problem of truss structure design. The mean curvature flow equation is solved numerically by a finite volume scheme that uses the cotangent discretization of the Laplace-Beltrami operator [4, 14]. The tangential velocity is obtained by solving an ordinary differential equation that is approximated by the finite difference approach.

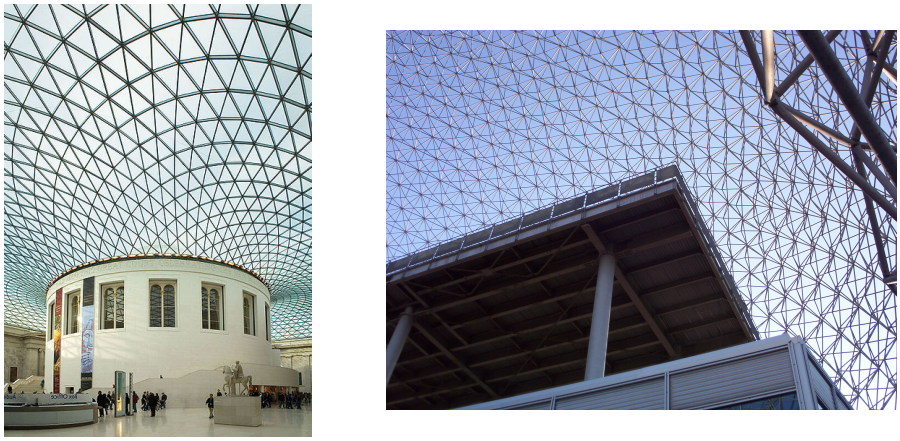

FIGURE 1. Examples of truss structures. Left, a truss structure as a part of a free-form shell structure (The Great Court of the British Museum, photo by Andrew Dunn, http://www.andrewdunnphoto.com/). Right, a truss structure as a stand-alone architectural element (Montréal Biosphère, photo by Colocho, http://fr.wikipedia.org/wiki/Utilisateur:Colocho).

2. **The truss structure design.**

From now on, we will consider only truss structures that can be viewed as a wireframe of a smooth surface $\Sigma \subset \mathbb{R}^3$. This is a common procedure in modern architecture – first, a smooth free-form shape is designed and only afterwards its possible realization using truss and shell elements is sought. Let Σ be obtained by an embedding $S\colon M \to \mathbb{R}^3$, where M is a two-dimensional Riemannian manifold with or without boundary. In the following sections, we show how to deal with two types of situations.

1. We are given a discretized surface – a discrete representation of a shape designed by an architect. Our task is to adjust the mesh in order to meet some specific criteria while keeping its original number of nodes and connectivity. Though moving the node points will modify the wireframe, we want to keep this error as small as possible – we are supposed to represent the same original surface. In the continuous setting, such movement of points can be modelled by the equation
$$\partial_t S = v_T,$$
where S is now considered to be a time dependent embedding, $S\colon M \times \langle 0, T_s\rangle \to \mathbb{R}^3$, $v_T\colon M \times \langle 0, T_s\rangle \to T\Sigma$ is a vector field tangential to Σ ($T\Sigma$ represents the tangential space of Σ) and $v_T\restriction_{\partial M}$ is tangential to $\partial\Sigma$.
2. We do not explicitly know the shape of the surface but we know it has to satisfy some specific condition. The corresponding shape can be found as a solution of an evolution equation of the form
$$\partial_t S = v,$$
where the smooth map $v\colon M \times \langle 0, T_s\rangle \to \mathbb{R}^3$ represents a general velocity in $\mathbb{R}^3$. Our goal is to find the shape of the surface and an appropriate discrete representation at the same time. Though it is possible to modify the mesh once the evolution has finished, in practice it is often more convenient to adjust the mesh on the run as thus we can also improve the quality of the mesh with respect to our numerical method. In order to achieve this, we can add a specifically designed tangential vector field to our evolution model, that means we obtain a modified model,
$$\partial_t S = v + v_T.$$
A typical example of surfaces that can be found by solving an evolution equation are minimal surfaces. These are popular due to their interesting shapes, practical properties (roofs that do not hold water) and minimization of material consumption. Mathematically, minimal surfaces are surfaces of zero mean curvature and thus can be obtained, for a given boundary curve $\Gamma = S(\partial M)$, by solving the equation
$$\partial_t S = HN,$$
where H is the mean curvature of S and N is a unit normal to Σ. This equation has to be accompanied by the boundary condition
$$\partial_t S = 0 \quad \text{on } \partial M.$$

3. **Tangential redistribution of points along curves on evolving surfaces.** We will now explain how to deal with the more general case of truss structure design, that means the case when we design the shape of the construction by solving an evolution equation. We will formulate all our results in a continuous setting and afterwards we will show how to apply them to the discrete case.

Since our object of interest are the lengths of the truss elements, we want to be able to control distances between selected points on the evolving surface. In the continuous setting, we will focus on selected curves on the surface and redistribute points along these curves in the course of the evolution.

We will extend the ideas of Mikula and Ševčovič [7, 8] and Mikula and Urbán [9] concerning redistribution of points on a single evolving 2D or 3D curve. Let $\delta\colon \mathbb{R} \supset I \to M$ be a smooth parametric curve on M. Push-forwarding δ along

S, we obtain a smooth time dependent parametric curve $\gamma\colon I \times \langle 0, T_s\rangle \to \Sigma$. Its evolution is governed by the same vector field as the evolution of S, that means

$$\partial_t \gamma = v.$$

If we want to achieve some specific distribution of points on the curve, we have to add a velocity term w that is tangential to the curve and appropriately designed according to our goal. Thus we have

$$\partial_t \gamma = v + w.$$

The overall velocity $v + w$ can be decomposed in three orthogonal components, namely

$$\partial_t \gamma = \beta_1 N_1 + \beta_2 N_2 + \alpha T, \tag{1}$$

where the vectors N_1 and N_2 are two orthogonal unit normals to γ and T is the unit tangent vector to γ,

$$T = \frac{\partial_u \gamma}{\|\partial_u \gamma\|} = \partial_s \gamma,$$

with s denoting the arc-length of γ and $u \in I$ representing the parameter corresponding to the parametric curve $\gamma(\cdot, t)$. The parameters $\beta_1, \beta_2\colon I \times \langle 0, T_s\rangle \to \mathbb{R}$ are the normal speeds and $\alpha\colon I \times \langle 0, T_s\rangle \to \mathbb{R}$ is the tangential speed,

$$\begin{aligned} \beta_1 &= v \cdot N_1, \\ \beta_2 &= v \cdot N_2, \\ \alpha &= v \cdot T + w \cdot T. \end{aligned}$$

The normal vectors can be chosen as, for example, $N_1 = N$, N being a unit normal to S, and $N_2 = T \times N_1$.

Equally sized line elements in a discrete representation of S can be easily obtained once we have a uniform parametrization of γ. That means we want to achieve

$$l = c,$$

where $l = \|\gamma_u\|$ is the length density of γ and c is a positive constant. A dimensionless equivalent of this condition is

$$\frac{l}{L} = c,$$

where L is the length of γ,

$$L = \int_I l \,\mathrm{d}u.$$

If we assume that the domain of γ is $I = \langle 0, 1\rangle$, we get $c = 1$ and we can write the condition for our evolution process as

$$\lim_{t\to\infty} \frac{l}{L} = 1. \tag{2}$$

This condition is satisfied if $\frac{l}{L}$ is a solution of the equation

$$\partial_t \left(\frac{l}{L}\right) = 1 - \frac{l}{L}.$$

Since in applications it is convenient to be able to control the speed of redistribution, we will use the equation enriched with the redistribution speed parameter $\omega\colon I \times \langle 0, T_s\rangle \to \mathbb{R}_+$,

$$\partial_t \left(\frac{l}{L}\right) = \omega \left(1 - \frac{l}{L}\right). \tag{3}$$

In order to be able to extract some condition for the tangential speed α from (3), we need to know how the length density and the global length evolve in the evolution process given by (1). We can deduce [3]

$$\partial_t l = l\partial_s \alpha - l(\beta_1 k_1 + \beta_2 k_2), \tag{4}$$

$$\partial_t L = -\int_0^L (\beta_1 k_1 + \beta_2 k_2)\,\mathrm{d}s - \alpha(0,\cdot) - \alpha(L,\cdot), \tag{5}$$

where k_1 and k_2 are the projections of the curvature vector $\kappa = \partial_s T$ to the directions of N_1 and N_2 ($k_1 = \kappa{\cdot}N_1$, $k_2 = \kappa{\cdot}N_2$). Let us now suppose that $\alpha(0,\cdot) = \alpha(L,\cdot) = 0$; this setting will be used in all experiments employing this type of redistribution. Equations (3), (4) and (5) lead to a condition for the arc-length derivative of α

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

Now, α is actually the sum

$$\alpha = \alpha_v + \alpha_w,$$

where α_v represents the tangential movement induced by the given velocity v and α_w represents our intentional tangential redistribution. Therefore, the speed of our redistribution will satisfy

$$\partial_s \alpha_w = -\partial_s \alpha_v + \beta_1 k_1 + \beta_2 k_2 - \frac{1}{L}\int_0^L (\beta_1 k_1 + \beta_2 k_2)\,\mathrm{d}s + \omega\left(\frac{L}{l} - 1\right). \tag{6}$$

It is often not enough just to obtain a group of truss elements of the same length but actually we want to set the length to some specific value. Therefore it can be useful if we have the possibility to prescribe also the limit length of the curve γ, L_∞. In that case, since $\partial_t L_\infty = 0$, we get a more simple condition

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

and further

$$\partial_s \alpha_w = -\partial_s \alpha_v + \beta_1 k_1 + \beta_2 k_2 + \omega\left(\frac{L_\infty}{l} - 1\right). \tag{7}$$

In this case, we do not assume $\alpha(0,\cdot) = 0$ or $\alpha(L,\cdot) = 0$, so the whole curve can stretch or shrink in the tangential direction. In the part devoted to results and experiments, we will show how to take advantage of this possibility.

Being able to redistribute points on a single curve, we can apply our technique to the truss structure design. The truss structure is a network of truss elements that is much more complicated than a single discrete curve. One way to deal with this situation is to identify a network of discrete curves in the structure and move the node points along these curves. In the smooth setting, let us imagine a set of smooth curves γ_k, $k = 1 \dots n_c$ on our surface. These curves can have intersections but with the limitation that each node point must be the intersection point of exactly two curves and the tangent vectors to the two intersecting curves must be linearly independent. Networks with these properties can be usually found in truss structures as their subsets. Restricting ourselves to a subset of our structure with the given properties is usually not too limiting since, many times, it is anyway impossible to impose some specific condition on all truss elements in the structure (for example, it is generally not possible to require all trusses to have the same length).

Now, if we compute the tangential velocity w_k for each curve γ_k, the question is how to set the velocity at the intersection points. Let P_{ijt} be an intersection point of the curves $\gamma_i(\cdot,t)$ and $\gamma_j(\cdot,t)$, i.e. $P_{ijt} = S(X_{ij},t) = \gamma_i(z_{ip},t) = \gamma_j(z_{jq},t)$, $X_{ij} \in M$. Let the tangential velocities at $\gamma_i(z_{ip},t)$ and $\gamma_j(z_{jq},t)$ be $w_i(z_{ip},t) = \alpha_{w,i}(z_{ip},t)T_i(z_{ip},t)$ and $w_j(z_{jq},t) = \alpha_{w,j}(z_{jq},t)T_j(z_{jq},t)$. Then we set the surface tangential velocity v_T at P_{ijt} to

$$v_T(X_{ij},t) = \frac{w_i(z_{ip},t) + w_j(z_{jq},t)}{2} \tag{8}$$

Since $T_i(z_{ip},t)$ and $T_j(z_{jq},t)$ are linearly independent, the left hand side of this equation will be zero only if the tangential velocities corresponding to the intersecting curves will be both zero. This will happen once all of the following conditions are satisfied.

1. S is not evolving anymore, that means $v = 0$. For example, if $v = HN$, the evolution stops when S is a minimal surface.
2. Both curves $\gamma_i(\cdot,t)$ and $\gamma_j(\cdot,t)$ are uniformly parametrized and, if we use (7), they have the prescribed global length.
3. $\alpha_{w,i}(0,t) = 0$, $\alpha_{w,j}(0,t) = 0$, that means the curves are not stretching or moving around the surface.

The equilibrium state of the whole system can be characterized analogously.

1. S is not evolving by the velocity field v, that means $v = 0$.
2. All curves $\gamma_i(\cdot,t)$ are uniformly parametrized and, if required, they have the prescribed global length.
3. $\alpha_{w,i}(0,t) = 0$.

Remark 1. For the sake of correctness, let us remark that we have defined the tangential velocity v_T in the node points of the network and in the other points of the curves γ_k, we take $v_T = w_k$. To make the velocity field smooth, we can apply, for example, a mollifier. Also, we did not specify what the value of v_T will be in the points of the surface that do not belong to any of the curves. Since this does not really matter once we apply our method to the discrete case, we can consider an arbitrary smooth extension in our continuous formulation.

4. **Constructions representing minimal surfaces.** As we have already mentioned in Section 2, a minimal surface is defined as a surface of zero mean curvature and as such it is the steady state of the evolution equation

$$\partial_t S = HN. \tag{9}$$

Here we consider

$$H = \kappa_1 + \kappa_2,$$

where κ_1 and κ_2 are the principal curvatures of S. The equation (9) can be alternatively rewritten in terms of the Laplace-Beltrami operator

$$\partial_t S = \Delta_S S, \tag{10}$$

where $\Delta_S S$ represents the Laplace-Beltrami operator with respect to the metric g_S induced by the map S as the pull-back of the Euclidean metric in $\mathbb{R}^3$ along S. If the equations (9) or (10) are applied to a closed surface, they lead to shrinking of the surface and smoothing of its mean curvature function. If they are accompanied by the boundary condition

$$\partial_t S = 0 \quad \text{on } \partial M, \tag{11}$$

the evolution converges to a minimal surface corresponding to the prescribed boundary curve $\Gamma = S(\partial M)$. This surface is generally not unique but it was proved, for example, that if Γ is a regular analytic Jordan curve in $\mathbb{R}^3$ whose total curvature is no more than 4π, then there is a unique minimal surface that has Γ as its boundary. This surface is an embedded disk and it is area-minimizing [11]. Many constructions have a boundary that satisfies this condition. Since we want to obtain a construction with optimized truss lengths, we will use the mean curvature flow model enriched with the tangential velocity term

$$\partial_t S = \Delta_S S + v_T. \tag{12}$$

We chose the version with the Laplace-Beltrami operator due to the discretization method that we are going to use.

4.1. **Time discretization.** The time discretization is semi-implicit, since it allows us to keep the favorable stability properties of the implicit approach while having to solve only a linear system in each time step. The model (12) is approximated as

$$\frac{S^n - S^{n-1}}{\tau} = \Delta_{S^{n-1}} S^n + v_T^{n-1}, \tag{13}$$

where τ is the time discretization interval and the index n represents the n-th time slice of a map. In case of the Laplace-Beltrami operator, the index S^{n-1} indicates that we consider the operator corresponding to the metric $g_{S^{n-1}}$ induced by the map S^{n-1}.

4.2. **Space discretization.** The space discretization of (13) is based on a triangular approximation of Σ. In order to obtain such an approximation, we consider a triangulation of M – a simplicial complex homeomorphic to M. The corresponding homeomorphism induces a triangular structure on M consisting of vertices M_i, $i = 1 \dots n_v$, edges e_j, $j = 1 \dots n_e$, and triangles $\mathcal{T}_k$, $k = 1 \dots n_t$; these elements are obtained as the images of the 0,1 and 2-simplices, respectively. Now we apply the finite volume approach. To this end, we decompose M into n_v polygonal control volumes V_i with the point M_i lying inside V_i. This decomposition is based on the barycentric subdivision of the triangles $\mathcal{T}_p$.

First, let M_i be an inner node of the triangulation (Figure 2). Then it is the common vertex of m mesh triangles $\mathcal{T}_1, \dots, \mathcal{T}_m$ and m edges $e_1, \dots, e_m$, where e_p connects M_i with its neighbor M_{i_p} (for simplicity, we temporarily use local indexing). The triangle $\mathcal{T}_p$ admits a barycentric coordinate system – each point of the triangle can be expressed as $P = \lambda_1 M_i + \lambda_2 M_{i_p} + \lambda_3 M_{i_{p+1}}$, where $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Let B_p be the barycenter of $\mathcal{T}_p$ and C_p the center of e_p, $p = 1 \dots m$, and let the barycentric subdivision of $\mathcal{T}_p$ be constructed using these points. The control volume V_i corresponding to M_i is then constructed as the union of the triangles $\mathcal{V}_{p,1} = M_i C_p B_p$ and $\mathcal{V}_{p,2} = M_i B_p C_{p+1}$ for $p = 1 \dots m$, where we set $C_{m+1} = C_1$. Each triangle contains two control volume edges $\sigma_{p,1} = C_p B_p$, $\sigma_{p,2} = B_p C_{p+1}$.

The manifold M can be embedded in $\mathbb{R}^3$ in a way that respects its triangular structure. Such an embedding $\bar{S}^n$ will be a piecewise linear approximation of S^n. To define it, we set $\bar{S}^n(M_i) = S^n(M_i)$. Then, for any triangle $\mathcal{T}_p$ with vertices M_i, M_{i_p}, $M_{i_{p+1}}$, we set

$$\bar{S}^n(\lambda_1 M_i + \lambda_2 M_{i_p} + \lambda_3 M_{i_{p+1}}) = \lambda_1 S^n(M_i) + \lambda_2 S^n(M_{i_p}) + \lambda_3 S^n(M_{i_{p+1}}).$$

Defining $\bar{S}^n$ in this way, $\bar{\Sigma}^n = \bar{S}^n(M)$ is a polyhedron with vertices $\bar{S}^n(M_i) = S^n(M_i) = S_i^n$, edges $\bar{e}_j^n = \bar{S}^n(e_j)$ and triangular faces $\bar{\mathcal{T}}_p^n = \bar{S}^n(\mathcal{T}_p)$. The embedding

$\bar{S}^n$ induces a metric g^n on M which, in turn, induces a measure μ^n on M. We will use the notation $\nu^n_{p,1}$, $\nu^n_{p,2}$ for the outward unit normals to $\bar{S}^n(\sigma_{p,1})$ and $\bar{S}^n(\sigma_{p,2})$ in the plane of $\bar{\mathcal{T}}^n_p$. Further, $\theta^n_{p,1}$ and $\theta^n_{p,2}$ will denote the angles of $\mathcal{T}_p$ adjacent to M_{i_p} and $M_{i_{p+1}}$, respectively, measured in the metric g^n.

If M_i is a boundary node with m neighboring mesh triangles, it has $m+1$ neighboring edges $e_1, \ldots, e_{m+1}$ and the point M_i is one of the vertices of the control volume V_i (Figure 3). However, the discretization of the Laplace-Beltrami operator is not necessary in the boundary points due to the boundary condition. Actually, we need only the area of the corresponding control volume.

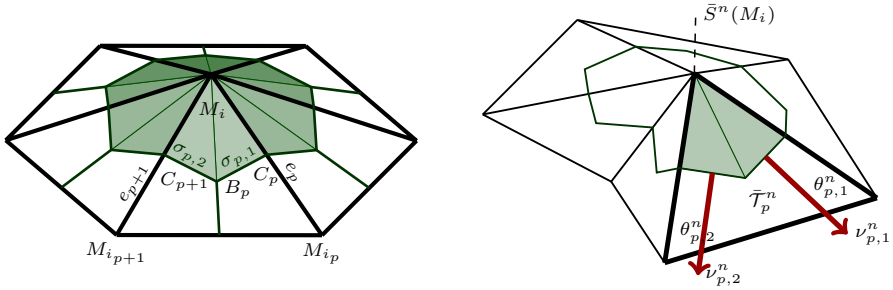

FIGURE 2. The discretization mesh – a sample control volume corresponding to an inner node. Left, discretization of the abstract manifold M. Right, discretization of the embedded surface $\Sigma^n = S^n(M)$.

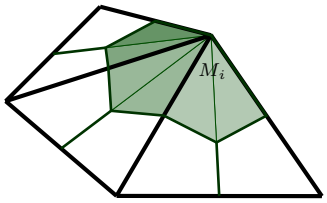

FIGURE 3. The control volume corresponding to a boundary node.

Now, we integrate (13) over V_i with respect to the measure $\mu_{S^{n-1}}$ induced by the metric $g_{S^{n-1}}$. We get

$$\int_{V_i} \frac{S^n - S^{n-1}}{\tau} \,\mathrm{d}\mu_{S^{n-1}} = \int_{V_i} \Delta_{S^{n-1}} S^n \,\mathrm{d}\mu_{S^{n-1}} + \int_{V_i} v_T^{n-1} \,\mathrm{d}\mu_{S^{n-1}}. \tag{14}$$

The time derivative term is simply approximated as

$$\int_{V_i} \frac{S^n - S^{n-1}}{\tau} \,\mathrm{d}\mu_{S^{n-1}} \approx \mu^{n-1}(V_i) \frac{S^n_i - S^{n-1}_i}{\tau}. \tag{15}$$

Next, we will explain the approximation of the two terms on the right hand side of (14).

4.2.1. *Discretization of the Laplace-Beltrami operator.* The Laplace-Beltrami operator is approximated by the cotangent scheme [4, 14]. Since it is a widely used method, we will just sketch the first fundamental steps of the procedure.

First, we apply the divergence theorem to obtain

$$\int_{V_i} \Delta_{S^{n-1}} S^n \,\mathrm{d}\mu_{S^{n-1}} = \int_{\partial V_i} \nabla_{S^{n-1}} S^n \cdot \nu_i^{n-1} \,\mathrm{d}H_{\mu_{S^{n-1}}},$$

where ν_i^{n-1} represents the outward unit normal to $S^{n-1}(\partial V_i)$ and $\nabla_{S^{n-1}}$ is the gradient corresponding to the metric $g_{S^{n-1}}$. Now, we use the approximation $S^n \approx \bar{S}^n$ and hence

$$\int_{\partial V_i} \nabla_{S^{n-1}} S^n \cdot \nu_i^{n-1} \,\mathrm{d}H_{\mu_{S^{n-1}}} \approx \sum_{p=1}^{m} \sum_{q=1,2} \int_{\sigma_{p,q}} \nabla_{n-1}\ \bar{S}^n \cdot \nu_{p,q}^{n-1} \,\mathrm{d}H_{\mu^{n-1}}.$$

Here, ∇_{n-1} denotes the gradient corresponding to the metric g^{n-1}. The fact that the gradient of $\bar{S}^n$ is constant on $\mathcal{T}_p$ implies that [4]

$$\sum_{q=1,2} \int_{\sigma_{p,q}} \nabla_{n-1} \bar{S}^n \cdot \nu_{p,q}^{n-1} \,\mathrm{d}H_{\mu^{n-1}} = \frac{1}{2}\left(\cot\theta_{p,2}^{n-1}(S_{i_p}^n - S_i^n) + \cot\theta_{p,1}^{n-1}(S_{i_{p+1}}^n - S_i^n)\right).$$

Finally, this leads to the approximation

$$\int_{V_i} \Delta_{S^{n-1}} S^n \,\mathrm{d}\mu^{n-1} \approx \frac{1}{2}\sum_{p=1}^{m} \left(\cot\theta_{i,p-1,1}^{n-1} + \cot\theta_{i,p,2}^{n-1}\right)(S_i^n - S_{i_p}^n), \tag{16}$$

where $\theta_{i,0,1}^{n-1} = \theta_{i,m,1}^{n-1}$. Now we included the full global indexing, since this approximation is directly used for composing the linear system representing the discretized problem.

4.2.2. *Discretization of the tangential velocity term.* Let us suppose that we have a set of discrete curves – polygonal lines – consisting of edges of the triangulated surface $\bar{\Sigma}^n$. We will describe the procedure for one chosen polygonal line $\bar{\Gamma}$, the discretization of a smooth curve $\Gamma \subset \Sigma^n$, $\Gamma = \mathrm{Im}(\gamma)$, γ being a parametric curve, $\gamma\colon \langle 0,1\rangle \to \Sigma^n$. The node points of $\bar{\Gamma}$ will be denoted by Γ_j, $j = 0 \ldots n_p$. We have $\Gamma_j = S_{i_j}^n = \gamma(u_j)$ for some $i_j \in \{1,\ldots,n_v\}$. Since $\bar{\Gamma}$ is one-dimensional, its node points can be ordered, for example, by the number of segments separating them from a boundary point Γ_0. Let us suppose that the index j respects this order.

The redistribution velocity $w(u_j)$ at Γ_j is computed as

$$w(u_j) = \alpha_w(u_j) T(u_j). \tag{17}$$

Now, $T(u_j)$ is approximated as

$$\begin{aligned}
T(0) &\approx T_0 = \frac{\Gamma_1 - \Gamma_0}{h_0}, \\
T(u_j) &\approx T_j = \frac{\frac{\Gamma_{j+1}-\Gamma_j}{h_j} + \frac{\Gamma_j - \Gamma_{j-1}}{h_{j-1}}}{2}, \quad j = 1 \ldots n_p - 1, \\
T(1) &\approx T_{n_p} = \frac{\Gamma_{n_p} - \Gamma_{n_p-1}}{h_{n_p-1}},
\end{aligned} \tag{18}$$

where

$$h_j = \|\Gamma_{j+1} - \Gamma_j\|.$$

The approximation $\alpha_{w,j}$ of the tangential speed $\alpha_w(u_j)$ is obtained by discretizing (6) or (7). We use

$$\partial_s \alpha_w(u_j) \approx \frac{\alpha_{w,j+1} - \alpha_{w,j}}{h_j} \tag{19}$$

and $\alpha_{w,0} = \alpha_w(0)$. The arc-length derivative $\partial_s \alpha_v$ can be approximated in the same way, however, in this case we have $\alpha_v = 0$. Further, in order to discretize k_1 and k_2, we need to approximate the curvature vector $\kappa = \partial_s T$ and the normal vectors N_1 and N_2. Here, we use

$$\kappa(u_j) \approx \kappa_j = \frac{T_{j+1} - T_j}{\frac{h_j + h_{j-1}}{2}} \tag{20}$$

for $j = 1 \dots n_p - 1$. In the first and the last point, we can set

$$\begin{aligned} \kappa_0 &= \frac{T_1 - T_0}{h_0}, \\ \kappa_{n_p} &= \frac{T_{n_p} - T_{n_p - 1}}{h_{n_p - 1}}, \end{aligned}$$

if we need to approximate the curvature there. The normal vector N_1 is set to be equal to a unit normal N to the surface. The point Γ_j is the common vertex of m triangular faces $\bar{\mathcal{T}}_1^n, \dots, \bar{\mathcal{T}}_m^n$. If $N_{\mathcal{T},p}$ is the outward unit normal to the p-th triangle, then the outward unit normal to Σ at Γ_j is approximated as

$$N_1(u_j) \approx N_{1,j} = \frac{1}{m} \sum_{p=1}^{m} N_{\mathcal{T},p}. \tag{21}$$

Then we set

$$N_2(u_j) \approx N_{2,j} = T_j \times N_{1,j} \tag{22}$$

and

$$\begin{aligned} k_1(u_j) &\approx k_{1,j} = \kappa_j \cdot N_{1,j}, \\ k_2(u_j) &\approx k_{2,j} = \kappa_j \cdot N_{2,j}. \end{aligned} \tag{23}$$

The length L of Γ is approximated straightforwardly by the length of $\bar{\Gamma}$,

$$L \approx \bar{L} = \sum_{j=0}^{n_p - 1} h_j. \tag{24}$$

From this follows the approximation of the average value of $\tilde{\beta} = \beta_1 k_1 + \beta_2 k_2$,

$$\frac{1}{L} \int_0^L \tilde{\beta} \, \mathrm{d}s \approx \frac{1}{\bar{L}} \sum_{j=0}^{n_p - 1} \frac{\tilde{\beta}_j + \tilde{\beta}_{j+1}}{2} h_j. \tag{25}$$

Here, since $\beta = \Delta_S S$, we have

$$\begin{aligned} \beta_{1,j} &= I_j \cdot N_{1,j}, \\ \beta_{2,j} &= I_j \cdot N_{2,j}, \end{aligned} \tag{26}$$

where

$$I_j = \frac{1}{2\mu^n(V_{i_j})} \sum_{p=1}^{m} \left(\cot \theta^n_{i_j,p-1,1} + \cot \theta^n_{i_j,p,2} \right) (S^n_{i_j} - S^n_{i_{j,p}}).$$

Finally, assuming $u_j = j \frac{1}{n_p}$, the length density $l(u_j) = \|\gamma_u(u_j)\|$ can be discretized as

$$l(u_j) \approx l_j = \frac{h_j}{\frac{1}{n_p}} = n_p h_j. \tag{27}$$

Summarizing (18)–(27), we obtain the approximation of (6),

$$\alpha^n_{w,j} = \alpha^n_{w,j-1} + h^n_j \tilde{\beta}^n_j - \frac{h^n_j}{\bar{L}} \sum_{j=0}^{n_p-1} \frac{\tilde{\beta}^n_j + \tilde{\beta}^n_{j+1}}{2} h^n_j + \omega^n_j \left(\frac{\bar{L}}{n_p} - h^n_j \right) \tag{28}$$

as well as the approximation of (7),

$$\alpha^n_{w,j} = \alpha^n_{w,j-1} + h^n_j \tilde{\beta}^n_j + \omega^n_j \left(\frac{L_\infty}{n_p} - h^n_j \right). \tag{29}$$

Finally, the integral of the tangential velocity term in (14) is approximated as

$$\int_{V_i} v^{n-1}_T \, \mathrm{d}\mu_{S^{n-1}} \approx \mu^{n-1}(V_i) v^{n-1}_{T,i}, \tag{30}$$

where $v^{n-1}_{T,i}$ is obtained from (17) or (8).

4.3. **Results.** Now we are ready to test our method and try to generate some truss structures. However, first of all, we present a test example intended to examine the experimental order of convergence of the proposed numerical scheme. The evolving surface is a sphere with the initial radius $r(0) = 1$. In this case, we know that the analytical solution of (9) is a sphere with radius

$$r(t) = \sqrt{1 - 4t}.$$

Our sphere was discretized non-uniformly and we used four levels of discretization with respect to the size of the grid elements (Figure 4). First, we ran a test with no tangential movement of points (Table 1). Then we investigated how the tangential redistribution affects the error and order of convergence (Table 2 and 3). We used the redistribution given by (6) with $\omega = 10.0$ and $\omega = 100.0$. The curves that were used for redistribution were the parallels and the meridians of the sphere. A few steps of the evolution process with redistribution are shown in Figure 5. The error in the sphere radius was measured in the L_2 norm (over space and time) and we used the coupling $\tau \sim h^2$, where h characterizes the size of the mesh elements. The stopping time was $T_s = 0.06$. We can see that the model with redistribution gives a slightly smaller error than the model with no redistribution for $\omega = 10.0$. For $\omega = 100.0$, the error is bigger due to the approximation of the tangential movement direction. For such a fast redistribution, the approximation error starts to manifest itself with a certain deviation of the points from the surface, where they were originally situated. However, this error is not large and the second order of convergence is preserved.

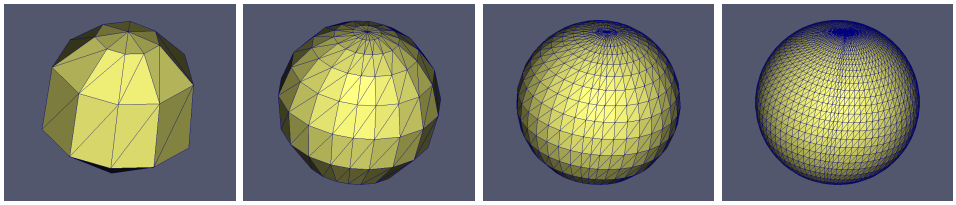

FIGURE 4. The discretization of the sphere, from left to right $n_v = 26, 114, 482, 1986$.

n_V	τ	L_2 error	EOC
26	0.01	3.6622e-3	
114	0.0025	1.9804e-3	0.8870
482	0.000625	6.9700e-4	1.5065
1986	0.00015625	1.7627e-4	1.9834

TABLE 1. The experimental order of convergence for the case with no tangential redistribution.

n_V	τ	L_2 error	EOC
26	0.01	4.0553e-3	
114	0.0025	1.9045e-3	1.0904
482	0.000625	6.4756e-4	1.5563
1986	0.00015625	1.6564e-4	1.9670

TABLE 2. The experimental order of convergence for the case with tangential redistribution given by (6), $\omega = 10.0$.

n_V	τ	L_2 error	EOC
26	0.01	1.8484e-2	
114	0.0025	3.6808e-3	2.3282
482	0.000625	8.8305e-4	2.0594
1986	0.00015625	2.1603e-5	2.0314

TABLE 3. The experimental order of convergence for the case with tangential redistribution given by (6), $\omega = 100.0$.

Now we can demonstrate how our method works when used for truss structure design. In the first example, we generated a structure representing the minimal surface for the boundary curve $\Gamma = \mathrm{Im}(\gamma)$ consisting of four segments,

$$\begin{aligned}
\gamma_1(z) &= (0.1(2z-1)^2 - 0.1, z, 0.05\sin(2\pi z) + 0.1\pi z), \quad z \in \langle 0,1\rangle \\
\gamma_2(z) &= (z, 0, 0), \quad z \in \langle 0,1\rangle \\
\gamma_3(z) &= (1, z, 0.3\sin(\pi z)), \quad z \in \langle 0,1\rangle \\
\gamma_4(z) &= (z, 1, 0.1\pi - 0.1\pi z), \quad z \in \langle 0,1\rangle.
\end{aligned}$$

The initial condition was set to $S(\cdot, 0) = (0,0,0)$ everywhere except the boundary (Figure 6). The computation was run for 800 time steps with $\tau = 6.25 \cdot 10^{-3}$. As the surface was evolving, the node points were redistributed according to (6). This means that at the end of the evolution, we wanted each curve from the selected network of curves to have equally distributed node points. In reality, due to the numerical error and since the equal node distribution is a limit state for $t \to \infty$, the line segments may differ in length. In order to minimize the error, we set the redistribution speed to $\omega = 800.0$, which means that the redistribution was very fast. In this way, we achieved equal distribution of node points up to an error no larger than the admissible tolerance $\approx 1‰$ of the truss length. The network of curves that were used for redistribution is highlighted in Figure 7. The same

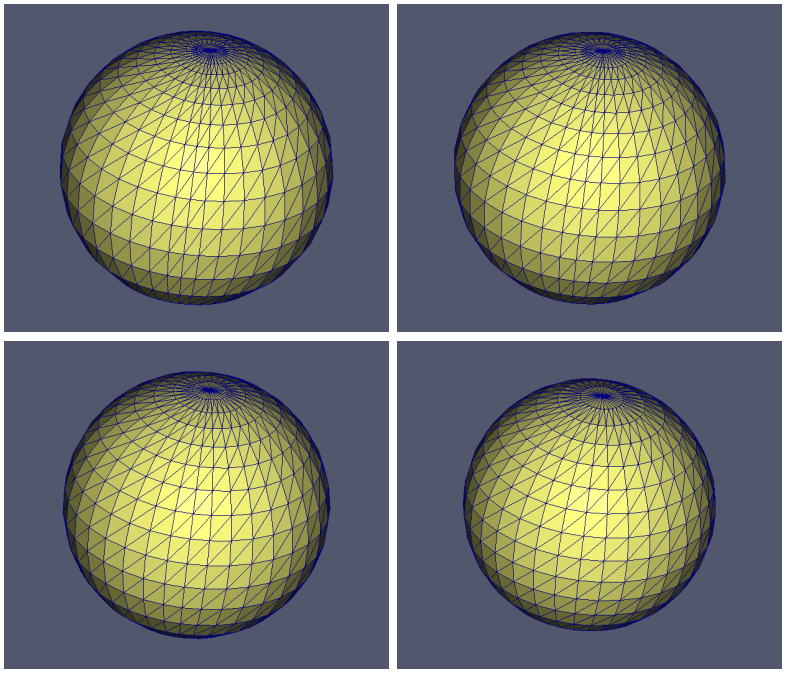

FIGURE 5. Evolution of the sphere with tangential redistribution given by (6), $\omega = 100.0$. The pictures show the sphere at time steps 0, 5, 20, 60. The points are redistributed along the parallels and the meridians.

figure shows also the resulting truss structure from a top view. Another view of the structure is shown in Figure 8. We can see that we obtained an aesthetic structure. From the manufacturing point of view, the type of redistribution that we used can be suitable for long narrow structures (for example roofs of airports) that contain long curves consisting of many truss elements. In such cases, the equal distribution of node points on the individual curves can help to optimize the manufacturing process.

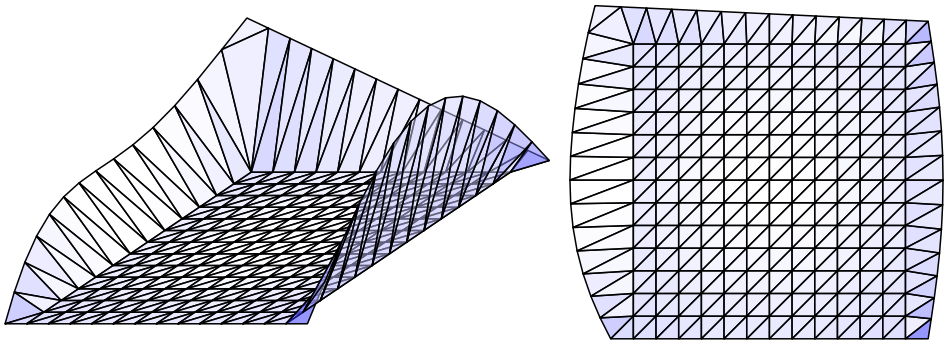

FIGURE 6. Example 1. Two different views of the initial condition. Left, an axonometric projection, right, a perspective projection.

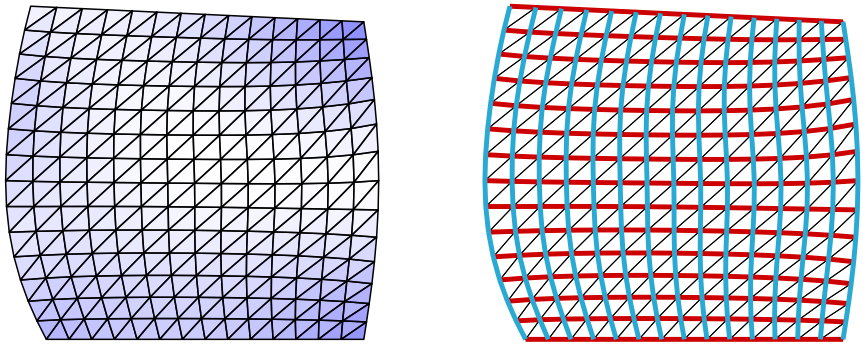

FIGURE 7. Example1. Asymptotically uniform redistribution of points on the individual curves. The figure shows the computed triangulated minimal surface in a perspective projection, top view. The curves included in the redistribution process are highlighted on the right picture.

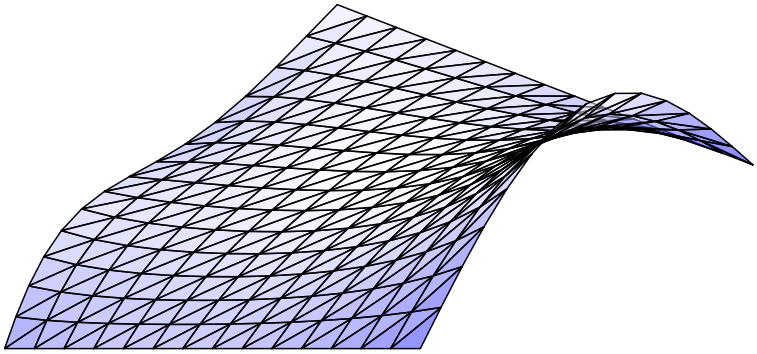

FIGURE 8. Example 1. The computed triangulated minimal surface in an axonometric projection.

The second example is intended to test the other type of redistribution given by (7). This time, the boundary of the structure was given by

$$\begin{aligned}
\gamma_1(z) &= (0.05(2z-1)^2 - 0.05, z, 0.25\sin(\pi z)), \quad z \in \langle 0,1\rangle \\
\gamma_2(z) &= (z, 0, 0), \quad z \in \langle 0,1\rangle \\
\gamma_3(z) &= (1 - 0.05(2z-1)^2 + 0.05, z, 0.25\sin(\pi z)), \quad z \in \langle 0,1\rangle \\
\gamma_4(z) &= (z, 1, 0), \quad z \in \langle 0,1\rangle.
\end{aligned}$$

In this case, we want the structure to contain as many equally sized truss elements as possible. Since now we are allowed to set the overall length of each curve in the network, in the discrete case it means that we are free to set the length of a single truss element. The curves that were used for redistribution are the "diagonal" curves, more precisely their parts between the second and the last but one point. The points were also redistributed on the boundary curves. Again, the redistribution speed was set to $\omega = 800.0$ and we performed 800 time steps with $\tau = 6.25 \cdot 10^{-3}$. In order to compute α_w for one of the curves, we need to set its value in the second point of the curve. For the sake of symmetry, we allowed the

second point to move – at the end, we want the first and the last segment of the curve to have the same length. This requirement implies

$$\alpha^n_{w,1} = h^n_0(\beta^n_{1,1}k^n_{1,1} + \beta^n_{2,1}k^n_{2,1}) + \omega^n_1\left(h^n_{n_p-1} - h^n_0\right). \tag{31}$$

The resulting structure will depend on how we choose the limit lengths of the truss elements. Figure 9 and 10 show four structures obtained by four different length settings. Figure 9 also depicts the truss elements that are equally sized at the end of the evolution. Though the node points are also redistributed on the boundary curves, the lengths of the elements there are different from the lengths in the highlighted part (they are only equal within the corresponding curve). The structures all contain 438 elements and among them 200 equally sized, which means we have 45.66% equally sized trusses. This ratio would grow with increasing number of elements; the limit ratio is 2/3 of the total number of elements (two out of three triangle sides would be equal for almost all triangles). To provide some quantitative evaluation of the achieved results, we complete the presented output by Table 4. This table concerns the 200 (approximately) equally sized truss elements. For each structure from Figure 9 and 10, it lists the prescribed truss length and then the maximal, minimal and average length at the end of the computation.

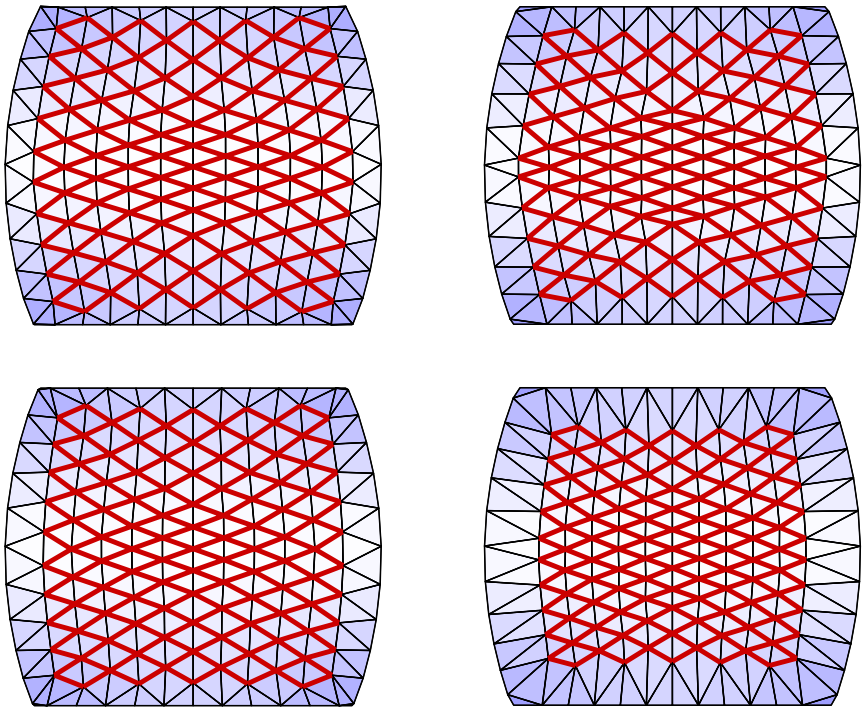

FIGURE 9. Example 2. The pictures represent four different structures approximating the same minimal surface. The structures differ in the length prescribed for the truss elements. The highlighted red elements are all equally sized.

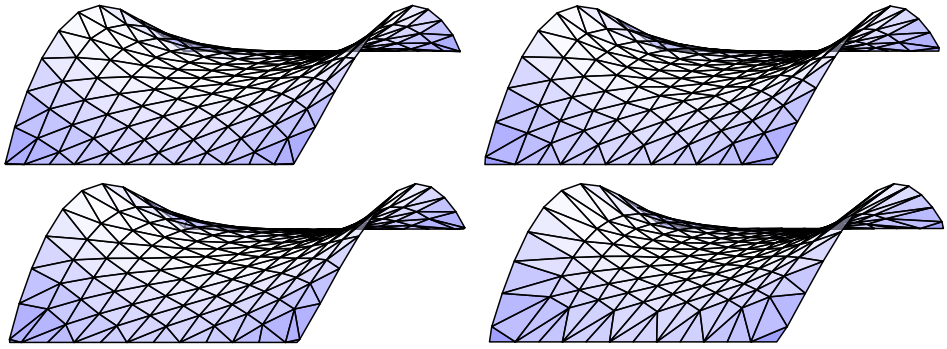

FIGURE 10. Example 2. The structures from Figure 9 in an axonometric view.

structure No.	prescribed	maximal	minimal	average
1	0.105	0.105079	0.104889	0.105001
2	0.1	0.100124	0.099969	0.100010
3	0.1	0.100167	0.099883	0.100010
4	0.088	0.088266	0.087968	0.088023

TABLE 4. Basic quantitative characteristics of the truss lengths for the four structures shown in Figure 9 and 10. The structures are numbered in the order top-left, top-right, bottom-left, bottom-right.

5. **General truss structures.** The redistribution method described in Section 3 does not necessarily require an evolution of the surface. In case when $\beta_1 = \beta_2 = \alpha_v = 0$, the equations (6) and (7) are reduced to simple relations

$$\partial_s \alpha_w = \omega \left(\frac{L}{l} - 1 \right) \tag{32}$$

and

$$\partial_s \alpha_w = \omega \left(\frac{L_\infty}{l} - 1 \right). \tag{33}$$

These can be used to adjust the lengths of truss elements in a general free-form structure with one additional step added to our procedure. As we can see in Table 3, the tangential redistribution can cause deviation of node points from the original surface. While the evolution by mean curvature tends to somewhat correct this error – the surface modified by redistribution is still dragged to the minimal surface shape – when we deal with a static surface, we have no mechanism to compensate the error in the tangential direction approximation. Therefore, in each time step, we perform the redistribution followed by a projection of the shifted points to the original triangulated surface. The projection is done simply by finding the nearest point on the original surface for each of the updated node points. Of course, this could cause degeneration of some triangles, since the projection is not injective. However, in most reasonable structures, this will not be the case.

To illustrate the performance of our method, we present two examples of free-form truss structures that we generated. The initial surfaces were constructed as

Bézier surfaces given by 9×5 control points, the triangulation was obtained simply from a uniform triangulation of the parameter domain. In both examples, we use the redistribution given by (33) with $\omega = 1500.0$. This time it was enough to perform 150 time steps with $\tau = 3.125 \cdot 10^{-3}$. The results can be seen in Figures 11, 12, 13 and 14.

Finally, we would like to mention that in all examples dealing with truss structures, the evolution straightforwardly converged to the equilibrium state described in Section 3.

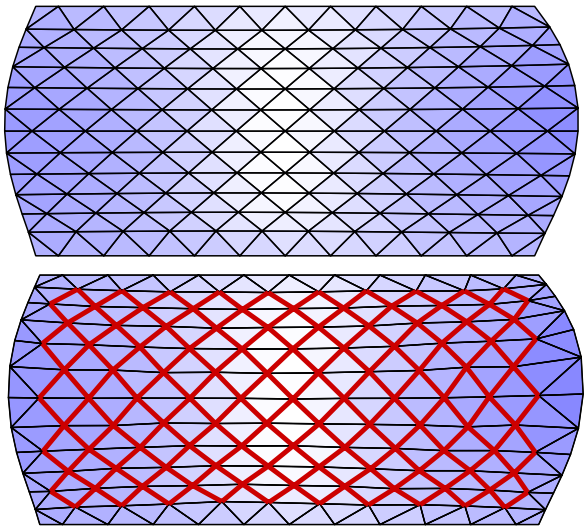

FIGURE 11. Remeshing of a static surface. Top, the initial surface. Bottom, the surface after remeshing. The segments with the same given length are highlighted.

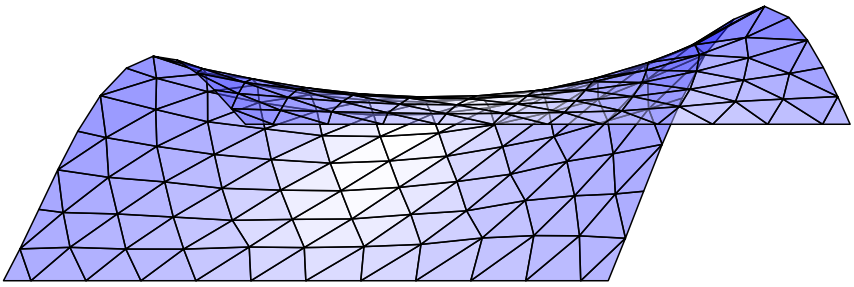

FIGURE 12. Remeshing of a static surface, an axonometric view of the surface from Figure 11, bottom.

REFERENCES

[1] M. R. Barnes, *Form Finding and Analysis of Tension Structures by Dynamic Relaxation*, International Journal of Space Structures, **14 (2)**, (1999), 89–104.

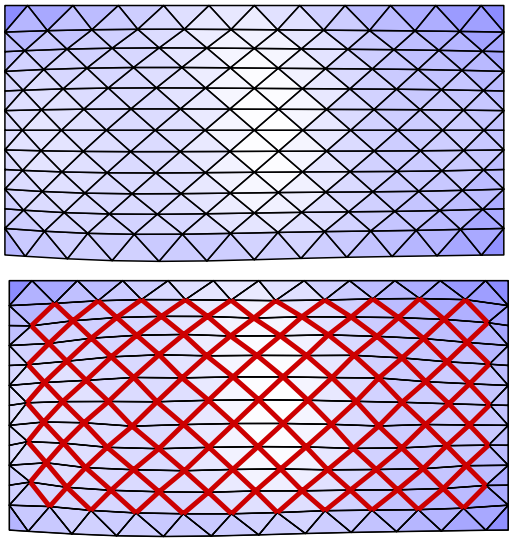

FIGURE 13. Remeshing of a static surface. Top, the initial surface. Bottom, the surface after remeshing. The segments with the same given length are highlighted.

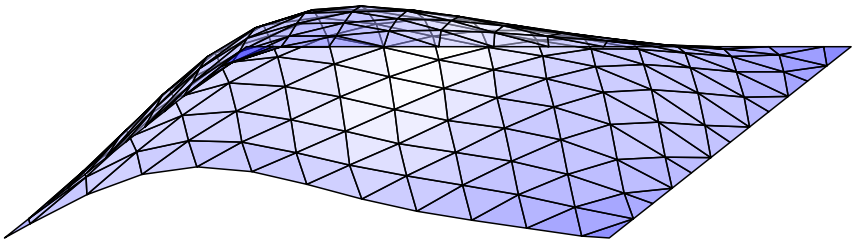

FIGURE 14. Remeshing of a static surface, an axonometric view of the surface from Figure 13, bottom.

[2] K. U. Bletzinger, M. Firl, J. Linhard, R. Wüchner, *Optimal shapes of mechanically motivated surfaces*, Computer Methods in Applied Mechanics and Engineering, **199 (5–8)** (2010), 324–333.

[3] M. Húska, M. Medľa, K. Mikula, P. Novysedlák, M. Remešíková, *A new form-finding method based on mean curvature flow of surfaces*, Proceedings of ALGORITMY 2012, 19th Conference on Scientific Computing, Podbanské, Slovakia, September 9-14, 2012, Publishing House of STU (2012), 120–131.

[4] M. Meyer, M. Desbrun, P. Schroeder, A. H. Barr, *Discrete differential geometry operators for triangulated 2manifolds*, Visualization and Mathematics III (Hans-Christian Hege and Konrad Polthier, eds.) (2003), 35-57.

[5] B. Maurin, R. Motro, *The surface stress density method as a form-finding tool for tensile membranes*, Engineering Structures, **20 (8)** (1998), 712–719.

[6] K. Mikula, M. Remešíková, P. Sarkoci, D. Ševčovič, *Surface evolution with tangential redistribution of points*, to appear in SIAM Journal of Scientific Computing.

[7] 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), 1473–1501.

[8] 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), 1545–1565.

[9] 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.

[10] A. Mottaghi Rad, H. Jamili, S. A. Behnejad, *Length Equalization of Elements in Single Layer Lattice Spatial Structures*, Abstract Book of the IASS-APCS 2012 Conference, Seoul, Korea, (2012), 266–273.

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

[12] F. Pantano, H. Tamai, *Geometric Multi-objective Optimization of Free-form Grid Shell Structures*, Abstract Book of the IASS-APCS 2012 Conference, Seoul, Korea, (2012), 85–93.

[13] R. M. O. Pauletti, P. M. Pimenta, *The natural force density method for the shape finding of taut structures*, Computer Methods in Applied Mechanics and Engineering, **197 (49–50)** (2008), 4419–4428.

[14] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Experim. Math. **2** (1993), 15-36.

[15] H. J. Scheck, *The force density method for form finding and computation of general networks*, Computer Methods in Applied Mechanics and Engineering **3 (1)** (1974), 115–134.

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

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: mikula@math.sk
E-mail address: peter.novysedlak@stuba.sk
E-mail address: remesikova@math.sk