Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
2025, vol. 30, no. 1, pp. 133–146
https://doi.org/10.2478/acss-2025-0015
https://content.sciendo.com

1862
RIGA TECHNICAL UNIVERSITY

Accelerated Planar Development of Convex Free-form Mesh Patches Using a Variable Step-size Energy Dissipation Approach

Erdem Yavuz[1*], Rıfat Yazıcı[2]
[1]*Department of Computer Engineering, Bursa Technical University, Bursa, Turkey*
[2]*Department of Computer Engineering, Fenerbahçe University, Istanbul, Turkey*

***Abstract* – Free-form complex surfaces are prevalent in modern graphic applications. With the increasing prevalence of complex 3D surfaces enabled by advances in range scanning and 3D printing technologies, minimising parameterization times for large meshes has become crucial. This paper proposes an efficient approach for the planar development of convex free-form mesh patches using an improved energy-based technique with a variable step-size algorithm. Building upon the energy model of Wang et al., our study addresses the limitations of conventional energy dissipation algorithms, which employ fixed step sizes. The proposed variable step-size method, particularly suitable for convex or disk-shaped mesh surfaces, dynamically adjusts steps, significantly reducing energy dissipation iterations. Leveraging our previous geometric flattening method, we further enhance planar surface development using an advanced mass-spring-based approach. Here, we show that our method accelerates the mechanical flattening process while maintaining high accuracy, achieving a shape error of 0.400 and an area error of 0.147 after 36 iterations for the Surf1 patch, reducing the required iterations by nearly half compared to the fixed step-size method. This study contributes to advancing the field of surface parameterization and flattening, with potential applications in various industries.**

***Keywords* – Convex shape, energy model, free form, surface flattening, surface parameterization.**

I. Introduction

The challenge of translating curved surfaces into their flat, planar counterparts is a longstanding issue in industries such as shipbuilding, toy manufacturing, apparel, footwear, and furniture production. Conventionally, 2D components are reshaped during manufacturing and subsequently assembled to form the final product. A critical factor in this process is the stress experienced during deformation and assembly, as it generates elastic energy that can diminish the quality of the end product and contribute to material fatigue. Historically, these industries have relied on a trial-and-error method for design. Designers typically sketch 2D components manually and produce prototypes that align with the intended product. When the prototype falls short of expectations, the patterns are revised based on the designer's expertise, and another prototype is created. This iterative cycle of prototyping and modification often results in inefficiency, high costs, and considerable labour and time investment [1].

In the current landscape, generating 2D patterns from 3D surface patches through surface flattening techniques often fails to meet the desired production standards, primarily due to material stress. Conversely, surface lattices that employ isometric projections are unaffected by this issue, as the flattening process does not introduce stress [2]. Despite advancements, commercial CAD/CAM (Computer-Aided Design/Manufacturing) software is limited in its capacity to model complex, arbitrarily shaped surfaces. Moreover, existing methodologies for developable surfaces, as presented in the literature, remain inadequate for accurately modelling free-form surfaces [1].

Surface texture characterisation is well-defined for Euclidean planar surfaces, where each point corresponds to a single height value. In contrast, free-form surfaces have more complex structures with non-zero curvature, making them non-Euclidean [3]. Gauss's Theorema Egregium states that surfaces with the same curvature can be projected onto each other without distortion, but this is not possible for surfaces with different curvatures. The Earth, for instance, cannot be projected onto a plane without distortion [4]. Similarly, projecting free-form surfaces onto a plane inevitably distorts or loses surface information, hindering accurate representation on a 2D grid. For example, projecting a hemisphere onto a 2D grid distorts geodesic distances, making them appear shorter than they are [3], underscoring the challenge of preserving non-Euclidean surface integrity in planar representations.

Free-form surfaces are vital in modern graphics applications [5], with tasks like rendering, tessellation, and sampling heavily reliant on high-quality surface parameterization. Conformal parameterization is essential, as it ensures a unit circle in parameter space maps to a unit circle on the surface [6]. While significant progress has been made in conformal mapping for triangular meshes, advancements for more complex free-form surfaces remain limited [6]. This lack of effective conformal

* Corresponding author's e-mail: erdem.yavuz@btu.edu.tr
Article received 2025-04-07; accepted 2025-10-17

©2025 Author(s). This is an open access article licensed under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0).

parameterization poses a major challenge to achieving high-quality results in rendering and tessellation processes [6].

Surface flattening involves various methodologies, including using differential geometry to assess surface developability, affine transformations for approximating isometric projections, minimising distortions in 3D-to-2D projections, and applying shape optimisation techniques for planar representations. Flattening complex surfaces projects a 3D surface onto a 2D plane based on defined principles [7]. Research categorises flattening methods into three main types: geometric flattening, mechanical flattening, and hybrid approaches combining both [8].

Research on geometric flattening [9], [10] has been complemented by significant advancements in mechanical flattening [11]–[14]. McCartney et al. [15] introduced an algorithm to reduce energy variations during flattening, effectively addressing the challenges posed by anisotropic woven fabrics, which differ mechanically from knitted materials. Lee and Huh [16] used the finite element method (FEM) to predict blank shapes and stress distributions in metal sheet forming, addressing issues like finite element alignment, plastic deformation theory, and constrained minimisation of plastic work. Similarly, Lan et al. [17] developed a surface flattening algorithm based on deformation theory and FEM. Further studies by Chung and Richmond [18], [19] and Chung et al. [20] explored designing blank metal sheets using the inverse finite element method, which derives a 2D blank geometry from a given final shape to minimise stress during forming [21].

Zhang et al. [22] developed a universal algorithm for flattening complex surfaces by reframing the geometric challenge of surface flattening as a mechanical structural problem through the consideration of mesh surface edges and nodes. In a separate study, Yoon et al. [23] introduced a computationally efficient model for flat design, applying ideal forming theory to the design of non-planar interfaces in hydroforming processes. Wang and Wang [24] proposed an energy-based algorithm aimed at flattening complex surfaces. Their approach involved constructing a physical model where nodes were interconnected by springs. By resolving the energy function at equilibrium during the flattening process, two-dimensional surfaces were derived from three-dimensional surfaces, with local control over surface accuracy. The following year, Cai et al. [25] formulated a positive definite objective function under two key conditions – uniform deformation distribution and volume conservation. Through iterative minimisation of this function, the planar surface contour was obtained. Similarly, Wang et al. [26] presented a method grounded in a mass-spring model, where nodes were treated as masses and edges as springs [27].

Shin [28] investigated the planar development of free-form surfaces composed of anisotropic materials using a hybrid modelling approach. The free-form surface was initially modelled through triangular elements. After geometric refinement, utilising Euler operators on the triangular representation, an initial projection onto a two-dimensional plane was performed. Optimal 2D contours were derived using the strain energy method with constant strain triangles (CST). A contour alignment technique was introduced for error assessment, enabling comparison of the generated contours. The effectiveness of the method was validated with a specific example object.

Liu et al. [29] proposed an energy-based algorithm for flattening mesh surfaces into planar patterns for manufacturing blank metal components. The method utilised a simplified energy model and introduced a variable step-size energy dissipation process to improve computational efficiency. For complex surfaces, a hierarchical flattening technique reduced distortions. Benchmarking against the classical ARAP (as-rigid-as-possible) mesh parameterization showed that the approach effectively generated flat patterns for blank metal production [27].

Abdul-Rahman et al. [3] proposed a novel model for parameterizing free-form surfaces using a projection algorithm before calculating parameters. Unlike conventional methods, it employs 3D triangular meshes to represent complex geometries and projects surfaces to 2D with minimal distortion, enabling more accurate parameterization of non-Euclidean textures. The study identified four height-related statistical parameters as effective for describing surface texture. Results showed the proposed approach outperformed classical projection methods. While direct surface parameterization is ideal but challenging, this model offers a valuable intermediate step, highlighting the need for further research on optimal projection techniques.

In 2016, Yang et al. [6] introduced hierarchical free-form surfaces and an optimisation algorithm using free-form deformations to improve the conformality of NURBS surfaces. A non-linear energy function measured conformal deviations, with initial deformations derived via the Ricci flow method. The study combined non-linear optimisation with deformation refinement to meet user-defined tolerances for angle preservation. While conformality is critical for surface realisation and coating, equiareality – matching 3D and 2D surface areas – is essential for sampling and coating. The trade-off between conformality and equiareality highlights the potential for future research to enhance area preservation [6].

Yavuz and Yazici [30] developed a recurrent neural network-based technique to generate 2D planar counterparts of arbitrary 3D objects using a geometric flattening method that preserves length and minimises strain energy. This approach simplifies 3D surface parameterization by creating a one-to-one correspondence with a 2D plane, streamlining complex 3D modelling tasks [31]. Recognising the growing importance of reducing parameterization time for large meshes in applications like scanning and 3D printing, the authors also introduced a dynamic neural network model. By combining a feedback neural network with barycentric mapping, this method leverages neural network parallel-distributed processing and the efficiency of barycentric mapping to significantly reduce computational time for large meshes.

In the current study, the initial flattening results from the geometric flattening method were further refined using a mass-spring-based approach. Specifically, the initial projection from the previous study, based on barycentric projection, underwent

an energy dissipation process using a variable step-size algorithm. This process continued until the desired area and edge accuracy criteria were met. The mechanical flattening approach in this study was based on the energy model proposed by Wang et al. [26]. While the original model used a fixed step size for energy dissipation, the current study applied a variable step size to the equations for acceleration, velocity, and position. The results showed that the variable step-size algorithm required fewer iterations than the fixed step-size approach, leading to faster energy dissipation.

The structure of this paper is as follows: Section 2 introduces the foundational concepts; Section 3 outlines the proposed method; Section 4 presents the experimental results, and Section 5 concludes with final remarks.

II. Preliminaries

This section outlines the fundamental concepts and essential conditions for the research, with a focus on the energy-based surface flattening approach proposed by Wang et al [26]. The energy-based surface flattening process can be defined as a deformation process of a planar triangular mesh governed by the energy function of a mass-spring system [26]. A planar triangular mass-spring system is used to control the shape of the triangular mesh and perform simulations. The entire model comprises a collection of virtual masses interconnected by springs [29].

A. Definition of Energy Function

This method uses a mass-spring system to simulate the deformation of a planar triangular Ω-mesh. The Ω-mesh is represented as (K, P), where K defines the node connectivity (topology) and $P = \{P_1, ..., P_m\}$ specifies the node positions in R^2. A 3D mesh surface is represented by (K, Q), where $Q_i \in R^3$ corresponds to P_i in the plane. In the mass-spring system, parameters like forces and masses are derived from geometric properties – forces from elastic deformation energy and masses from node distances and triangle areas. The elastic deformation energy is the difference between the initial and final node positions in the 2D mesh. An example is shown in Fig. 1.

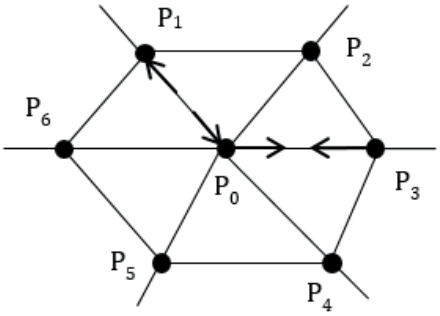


Fig. 1. Mass-spring system [26].

In this system, nodes P_i represent masses, and connections between P_i and P_j act as springs. During deformation, if the distance between P_i and P_j on the planar surface exceeds the corresponding 3D distance, an attractive force is applied (e.g., P_0 and P_3 in Fig. 1). Otherwise, a repulsive force is applied (e.g., P_0 and P_1). The elastic deformation energy and the tension force for each mass P_i are calculated as described by Wang et al. [26] with the following equations:

$$E(P_i) = \sum_{j=1}^{n} \frac{1}{2} C\left(\left|P_iP_j\right| - d_j\right)^2, \tag{1}$$

$$\vec{f}(P_i) = \sum_{j=1}^{n} \frac{1}{2} C\left(\left|P_iP_j\right| - d_j\right)\vec{n}_{p_ip_j}. \tag{2}$$

Here, C is the spring constant, $|P_iP_j|$ is the current distance between P_i and P_j on the planar surface, and d is the distance between Q_i and Q_j. The unit vector $\vec{n}_{p_ip_j}$ points from P_i to P_j, and n represents the number of neighbouring nodes of P_i. Forces are generated by differences in triangle edge lengths between the planar and 3D surfaces. The goal of surface flattening is to find the planar triangular mesh $\Omega = (K, P)$ that minimises the energy function, providing the best fit to the 3D mesh.

$$E(\Omega) = \sum_{i=1}^{m} E(P_i). \tag{3}$$

B. Energy Dissipation

The following Lagrange equation is commonly used to disseminate energy in energy models [32].

$$Mq'' + Dq' + Kq = g_q + f_q. \tag{4}$$

Here, M, D, and K are the mass, damping, and stiffness matrices. g_q represents gravitational force, and f_q denotes the external force linked to the model's degrees of freedom. If both forces are zero and damping is neglected, the Lagrange equation for the mass-spring system simplifies to:

$$Mq'' + Kq = 0. \tag{5}$$

The term Kq equals the force $-\vec{\mathbf{f}}$ in (2). The Euler method is used to solve (5). If Δt is small, the acceleration of mass P_i can be treated as constant. The system's equilibrium is the sum of the equilibria at each node. For each node P_i, (5) can be rewritten as [26]:

$$m_i = \frac{p}{3}\sum A_k, \tag{6}$$

$$q_i''(t) = \frac{f_i(t)}{m_i}, \tag{7}$$

$$q_i'(t+\Delta t) = q_i'(t) + \Delta t q_i''(t), \tag{8}$$

$$q_i(t+\Delta t) = q_i(t) + \Delta t q_i'(t) + \frac{\Delta t^2}{2} q_i''(t). \tag{9}$$

Here, m_i is the mass of P_i, p is the surface area density, and A_k is the area of the k^{th} triangle around node P_i. $q_i''(t)$ represents the acceleration, $f_i(t)$ the tensile force, and $q_i'(t)$ the velocity of node P_i at time t. The position of P_i at time t is also considered. Notably, p does not reflect the actual surface density. Both p

and the spring constant C are experimentally determined to fit the mechanical system. In many physics-based models, p and C act as scaling factors to control deformation. For example, $p = 1/\min\{m_i\}$ and $C = 0.5$ are commonly used. A step size of $\Delta t = 0.01$ is often sufficient to achieve equilibrium at a reasonable rate [26].

C. Energy Dissipation

This surface overlap or intersection often occurs during development. To prevent this, a penalty force is applied to moving masses. Based on Provot's (1995) method, a penalty function for mass P can be defined as follows:

$$\vec{F}_{\text{penalty}} = -\sum_{j=1}^{m} C_{\text{penalty}} |h_j - h_j^*| \vec{n} \begin{cases} C_{\text{penalty}} = 1 \ (h_j \le h_j^*) \\ C_{\text{penalty}} = 0 \ (h_j > h_j^*) \end{cases} \quad (10)$$

Here, h_j denotes the perpendicular distance from point P to its corresponding opposite edge on the planar surface (e.g., in Fig. 2, the distances from P to edge lines $\overline{P_1P_2}$, $\overline{P_2P_3}$,$\overline{P_3P_4}$,$\overline{P_4P_5}$,$\overline{P_5P_6}$,$\overline{P_6P_1}$). Meanwhile, h_j^* refers to the Euclidean distance between Q, the associated corner in the 3D lattice, and the j^{th} opposite edge on the original surface. The unit vector $\vec{n}$ indicates the direction from P toward the opposite edge in the planar representation, and m specifies the total number of opposite edges linked to P.

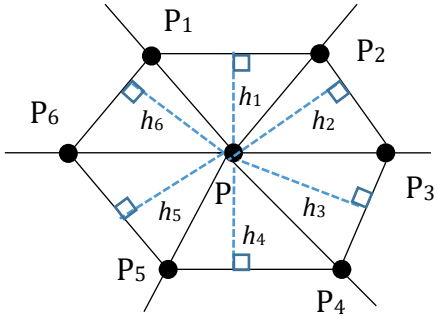


Fig. 2. Penalty functions [26].

The penalty function $\vec{F}_{\text{penalty}}$ is a vector that is applied to the position of mass P_i (node) to prevent overlap when P_i moves too close to its opposite edge.

$$q_{i\,t}^* = q_{i_t} + \overrightarrow{F}_{\text{penalty}}. \quad (11)$$

The algorithm named EnergyDissipation (Ω), detailed in the fundamental study proposed by Wang et al. [26], finds the node corner coordinates, P, of the planar mesh, $\Omega = (K, P)$, that minimise the energy function $E(\Omega)$.

D. Measurement of Accuracy

Two main criteria – area accuracy and edge/shape accuracy – are used to evaluate the precision of a developed planar surface. Surface area may vary during development, and the relative area difference or error can be quantified using the formula proposed by Wang et al. [26]:

$$E_S = \frac{\Sigma|A\text{-}A'|}{\Sigma A}. \quad (12)$$

Here, A is the actual area of a patch on the original surface, and A' is the corresponding area on the developed planar surface. To calculate the area of a surface defined by $S(u, v)$, a double integral is typically used [26]. However, in this approach, a facet model represents the surface, so the analytical form of S cannot be determined. Instead, A is approximated by summing the areas of all triangles in the model, as shown in the following equation [26]:

$$A = \sum_{i=0}^{n-1} A_i. \quad (13)$$

Here, A_i represents the area of the i^{th} triangle in the model, and n denotes the total number of triangles. Area accuracy is calculated as $1 - E_S$, where E_S represents the error. As a result, the smaller the E_S, the higher the area accuracy.

Shape (or edge) error is the difference between the length of a curve on the original surface and the length of the corresponding edge on the developed surface [26]. The final relative shape difference is calculated using the following equation:

$$E_C = \frac{\Sigma|L\text{-}L'|}{\Sigma L}. \quad (14)$$

Let L be the actual length of a curve on the original surface, and L' – its corresponding length on the developed planar surface. If the curve is part of the surface $C = S(u(t), v(t))$, calculating L involves a complex integral [26]. However, since the surface S is represented using a facet model, the analytical form of C is intractable. As a result, L can be approximated by summing the lengths of all triangle edges in the model.

$$L = \sum_{i=0}^{m-1} L_i. \quad (15)$$

Here, L_i represents the length of the i^{th} edge of the triangle in the model, and m denotes the total number of triangles. Shape accuracy is calculated as $1 - E_C$, where E_C represents the error. As a result, the smaller the E_C, the higher the shape accuracy.

III. Proposed Method

The proposed method, which introduces a variable step size, enhances the energy dissipation/release algorithm by Wang et al. [26], improving the determination of step size. The energy dissipation algorithm is essential for achieving the final flattened surface. Following the initial projection obtained through the geometric flattening technique from Yavuz and Yazici's previous study, the result undergoes an energy dissipation process using the energy release algorithm. This process continues until the conditions for allowable area and shape accuracy are satisfied.

$$E_S \leq \emptyset_S, \tag{16}$$

$$E_C \leq \emptyset_C. \tag{17}$$

Building on the energy-based flattening technique and its mathematical foundation outlined in Section 2, this study introduces several modifications to the energy dissipation algorithm proposed by Wang et al. [26]. A key enhancement is the shift from a fixed step size – previously calculated using the Euler method in (7)–(9) – to a variable step size. This variable step size is determined by a function based on the difference between the total area and edge lengths of the deforming surface in 2D space and their original values. The expression for the variable step size function is empirically derived as follows:

$$\Delta t_1 = 0.005 + 0.1e^{-5(1-E_S)}, \tag{18}$$

$$\Delta t_2 = 0.005 + 0.1e^{-5(1-E_C)}. \tag{19}$$

One of these exponential functions generates values based on the percentage of area accuracy, while the other is based on the percentage of shape accuracy. The resulting variations in step size, shown in Figs. 3 and 4, plot step size on the vertical axis, and 1 minus accuracy error on the horizontal axis. The exponential expressions in (8) and (9) yield larger step sizes when the differences in area or shape (edge) accuracy are greater. As the accuracy difference (i.e., error) decreases, the step size produced by these functions decreases exponentially. The graphs indicate that, due to the offset value, the function's output remains above 0.005, even when the accuracy difference is minimal.

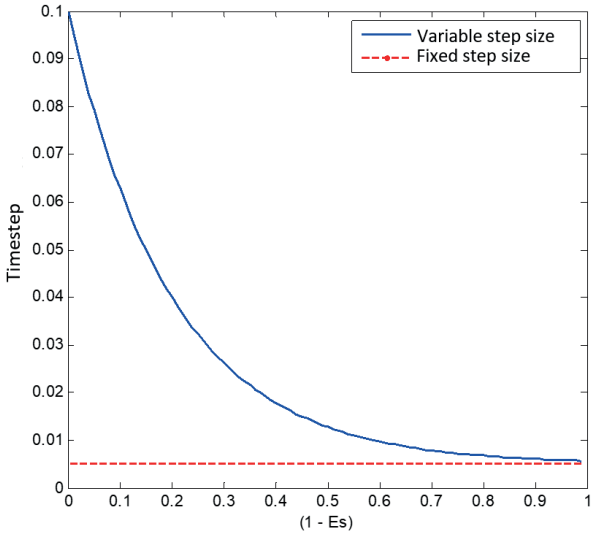


Fig. 3. Variation in area accuracy ($1-E_S$) with variable and fixed step size.

The final variable step size, Δt, was calculated by summing the expressions of the two aforementioned functions:

$$\Delta t = \Delta t_1 + \Delta t_2. \tag{20}$$

Since the constant terms in the two functions based on area and edge accuracy are directly summed in this expression, it is guaranteed that the final step size will not fall below 0.01. This is clearly evident in Fig. 4.

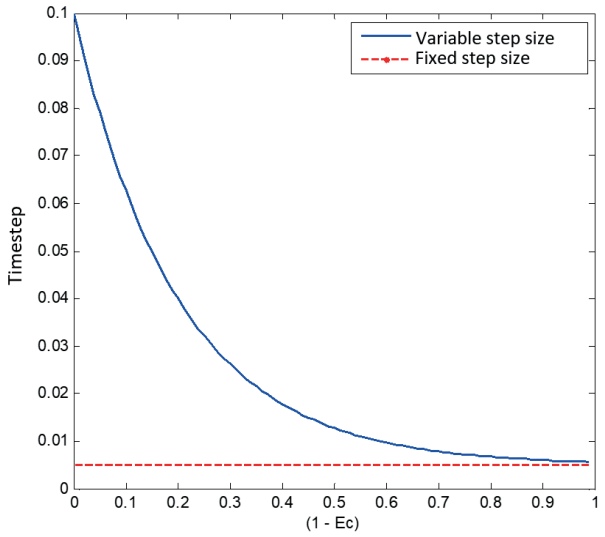


Fig. 4. Variation in shape accuracy ($1-E_C$) with variable and fixed step size.

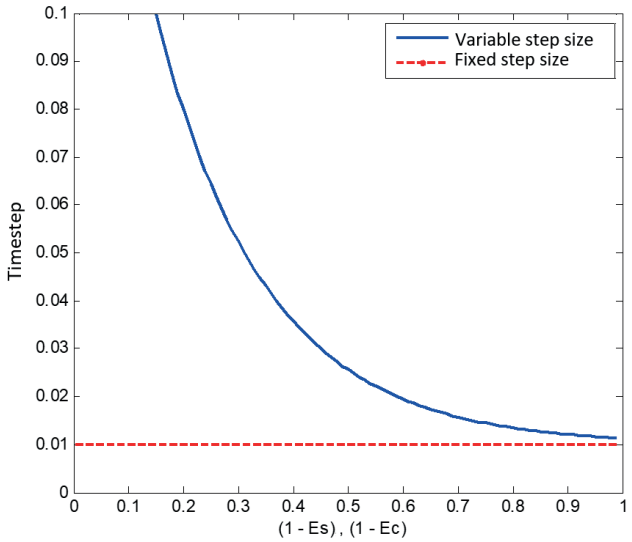


Fig. 5. Variation of the final variable step size according to area and shape accuracies.

The Δt step size in (7)–(9), solved using the Euler method, is calculated using (20). This approach aims to reduce the number of iterations required in the energy dissipation process compared to using a fixed step size. Table I presents the number of iterations, as well as the resulting final relative shape accuracy (E_C) and area (E_S) errors, for a specific surface patch when the energy dissipation process is carried out with a fixed or variable step size. As seen in the table, when the energy dissipation process is operated with a fixed step size ($\Delta t = 0.01$) to meet the conditions of staying below the allowed shape and area error thresholds for a specific surface patch, the desired accuracy rates are achieved in 53 iterations. On the other hand, when the process is operated with a variable step size to meet

the same conditions, approximately the same result is obtained in 18 iterations. The graph of the Δt step size values in (7)–(9) solved by the Euler method, when the process is operated with a variable step size, is plotted in red in Fig. 6.

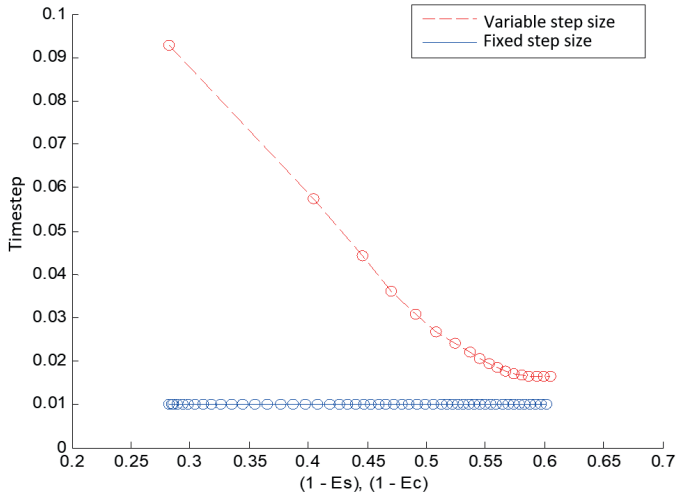


Fig. 6. Set of values obtained for the final variable step size versus the fixed step size.

TABLE I

COMPARISON OF ITERATION COUNTS, SHAPE AND AREA ERRORS IN THE ENERGY DISSIPATION PROCESS EMPLOYING FIXED AND VARIABLE STEP SIZES

Fixed step size ($\Delta t = 0.01$)		**Variable step size**	
Iteration number	**Shape and area errors**	**Iteration number**	**Shape and area errors**
1	$E_C = 0.717$, $E_S = 0.893$	1	$E_C = 0.717$, $E_S = 0.893$
2	$E_C = 0.717$, $E_S = 0.893$	2	$E_C = 0.596$, $E_S = 0.786$
3	$E_C = 0.716$, $E_S = 0.892$	3	$E_C = 0.554$, $E_S = 0.710$
4	$E_C = 0.714$, $E_S = 0.890$	4	$E_C = 0.529$, $E_S = 0.642$
5	$E_C = 0.710$, $E_S = 0.888$	5	$E_C = 0.509$, $E_S = 0.578$
6	$E_C = 0.706$, $E_S = 0.885$	6	$E_C = 0.491$, $E_S = 0.519$
7	$E_C = 0.701$, $E_S = 0.881$	7	$E_C = 0.476$, $E_S = 0.464$
8	$E_C = 0.696$, $E_S = 0.877$	8	$E_C = 0.463$, $E_S = 0.410$
9	$E_C = 0.689$, $E_S = 0.871$	9	$E_C = 0.455$, $E_S = 0.360$
10	$E_C = 0.682$, $E_S = 0.866$	10	$E_C = 0.447$, $E_S = 0.310$
11	$E_C = 0.673$, $E_S = 0.859$	11	$E_C = 0.440$, $E_S = 0.261$
12	$E_C = 0.665$, $E_S = 0.852$	12	$E_C = 0.433$, $E_S = 0.213$
13	$E_C = 0.655$, $E_S = 0.844$	13	$E_C = 0.426$, $E_S = 0.165$
14	$E_C = 0.645$, $E_S = 0.835$	14	$E_C = 0.419$, $E_S = 0.145$
15	$E_C = 0.634$, $E_S = 0.825$	15	$E_C = 0.413$, $E_S = 0.132$
16	$E_C = 0.623$, $E_S = 0.815$	16	$E_C = 0.407$, $E_S = 0.139$
17	$E_C = 0.613$, $E_S = 0.804$	17	$E_C = 0.401$, $E_S = 0.157$
18	$E_C = 0.602$, $E_S = 0.792$	**18**	$E_C =$ **0.395**, $E_S =$ **0.176**
19	$E_C = 0.592$, $E_S = 0.779$		
20	$E_C = 0.582$, $E_S = 0.766$		
21	$E_C = 0.574$, $E_S = 0.752$		
22	$E_C = 0.566$, $E_S = 0.737$		
23	$E_C = 0.559$, $E_S = 0.722$		
24	$E_C = 0.553$, $E_S = 0.705$		
25	$E_C = 0.547$, $E_S = 0.688$		
26	$E_C = 0.540$, $E_S = 0.670$		
27	$E_C = 0.534$, $E_S = 0.651$		
28	$E_C = 0.528$, $E_S = 0.632$		
. . .	. . .		
43	$E_C = 0.445$, $E_S = 0.249$		
44	$E_C = 0.440$, $E_S = 0.217$		
45	$E_C = 0.436$, $E_S = 0.185$		
46	$E_C = 0.431$, $E_S = 0.170$		
47	$E_C = 0.427$, $E_S = 0.159$		
48	$E_C = 0.422$, $E_S = 0.147$		
49	$E_C = 0.417$, $E_S = 0.136$		
50	$E_C = 0.413$, $E_S = 0.133$		
51	$E_C = 0.408$, $E_S = 0.144$		
52	$E_C = 0.403$, $E_S = 0.156$		
53	$E_C =$ **0.399**, $E_S =$ **0.167**		

Initially, the step size takes larger values when the error is high. The horizontal axis of the graph represents the accuracy rate (i.e., 1- error). At the start of the energy dissipation process in surface development, the area or shape error is generally high, resulting in a small difference from 1. For example, in the first iteration, when $E_C = 0.717$, the step size is calculated as approximately $\Delta t = 0.092$ using (20). This step value is represented by a red circular point on the graph in Fig. 6, located at 0.2826 on the horizontal axis, which corresponds to $(1-E_C)$. As the error of the developing surface decreases with each iteration – i.e., as the accuracy rate improves – the graph shows that the step size decreases exponentially. The number of points on the red curve corresponds to the number of iterations in the right column of Table I, which is 18. In contrast, the blue line represents the shape accuracy rate during surface development with a fixed step size. Since the step size remains constant at 0.01 throughout this process, the number of points on the blue line matches the number of iterations in the left column of the table, which is 53.

The details of the enhanced energy-based flattening algorithm, named VarStepSizeBasedEnergyDissipation (Ω), which takes the initial projection of a 2D point set P as input, are provided in Algorithm 1. This algorithm improves the flattening result by moving the coordinates of the surface patch in 2D space (based on energy) with a variable step size.

Algorithm 1: VarStepSizeBasedEnergyDissipation (Ω)

Input:

- A set of 2D projected points $P=\{P_0, P_1,\ldots, P_{n-1}\}$
- Number of nodes n.

Output:

- Final positions of P points that minimise the energy $E(\Omega)$.

1. Define the allowable percentage difference for area error $\emptyset_S$
2. Define the allowable percentage difference for shape error $\emptyset_C$
3. Define the allowable percentage change in energy (ϵ).
4. Set the maximum number of iterations (N_{max})
5. **For** i **from** 0 **to** $n-1$:
   - Calculate the mass of the node P_i using (6).
6. **End for loop**.

   While (Relative area accuracy error ($E_S > \emptyset_S$) **or** relative shape error percentage ($E_L > \emptyset_L$) **or** (change in E_L) > $\emptyset_L$)) **and** iteration count < N_{max}

   For i **from** 0 **to** $n-1$:
   - Calculate the stress force on P_i using (2).
   - Calculate the new position of P_i using (7)–(9).
   - Compute the penalty function to be applied to P_i using (10) and (11).
   - Move P_i to its new position.

   End for loop.
   - Update E_S using (12).
   - Update E_C using (14).
   - Update $E(\Omega)$ using (1) and (3).
   - Calculate Δt_1 using (18).
   - Calculate Δt_2 using (19).
   - Determine the final step size Δt using (20).
7. **End while loop**.
8. **Return** Ω.

IV. Experimental Results and Discussion

This section presents the results of refining the initial projection, derived through geometric flattening, by employing a mass-spring model.

A. Illustrative Examples

To improve the accuracy of the initial projections of sample surface patches (*Surf*1 and *Surf*2, detailed in the Appendix) onto a planar surface – generated using the initial projection algorithm proposed by Yavuz and Yazici [30] – this study employed the variable step-size energy dissipation algorithm described in Section 3. The initial projection results from the previous study were used as input data for the proposed approach in this study.

Intermediate outputs of the energy dissipation algorithm based on variable step size (Algorithm 1), executed with the parameters listed in Table II, are visually presented in Fig. 7, while the corresponding numerical results are provided in Table III. In the figure, green vectors indicate the direction of the force driving the mass, while red vectors represent the direction of the penalty (function) force acting on the mass. For simplicity in visualisation and representation, the forces are depicted with identical magnitudes.

TABLE II
Parameters of the Energy Dissipation Algorithm Based on Variable Step Size for the Surf1 Surface Patch

Parameters		
Description	**Symbol**	**Value**
Spring constant	C	0.5
Area density	p	2
Allowed area error percentage	$\emptyset_S$	0.16
Allowed shape error percentage	$\emptyset_C$	0.40
Allowed energy variation percentage	ϵ	0.05
Maximum allowed iterations	N_{max}	100

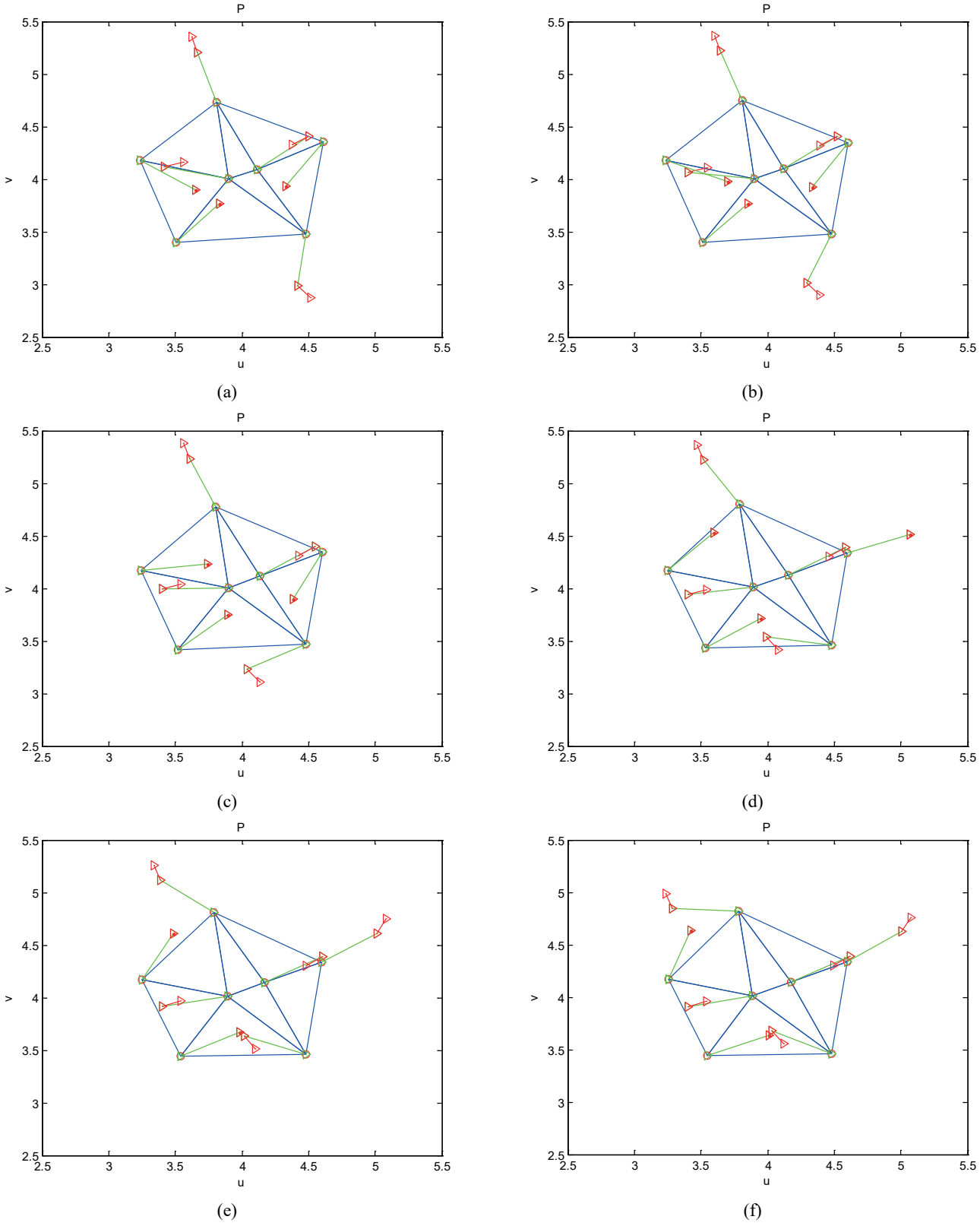


Fig. 7. Outputs of the energy dissipation algorithm at various steps for the *Surf*1 surface: (a) iteration #1; (b) iteration #5; (c) iteration #10; (d) iteration #20; (e) iteration #30; (f) iteration #36.

The intermediate and final results of the energy dissipation algorithm, applied to the initial projection of *Surf*1 under specified conditions, are presented. Using a variable step size, the algorithm was executed with the parameters listed in Table II. The choice of parameters C and p has been discussed in Section 2, while the remaining parameters define the stopping criteria of the algorithm. These criteria can be adjusted based on the type of planar surface and specific requirements.

TABLE III
SHAPE AND AREA ERROR VALUES CORRESPONDING TO THE ENERGY DISSIPATION RESULTS SHOWN IN FIG. 7

Iteration #	Shape error, E_C	Area error, E_S
1	0.419	0.197
5	0.417	0.191
10	0.415	0.185
20	0.411	0.173
30	0.407	0.158
36	0.400	0.147

The initial projection result for the *Surf*1 surface, subjected to the energy dissipation algorithm (executed with the parameters from Table II), was further improved after 36 iterations, with an area error value of $E_S = 0.147$ and an edge error value of $E_C = 0.400$, compared to the initial values of $E_S = 0.204$ and $E_C = 0.420$.

The results of the initial projection algorithm and the improvement achieved by the energy dissipation algorithm for the sample surface are overlaid in the graph in Fig. 8.

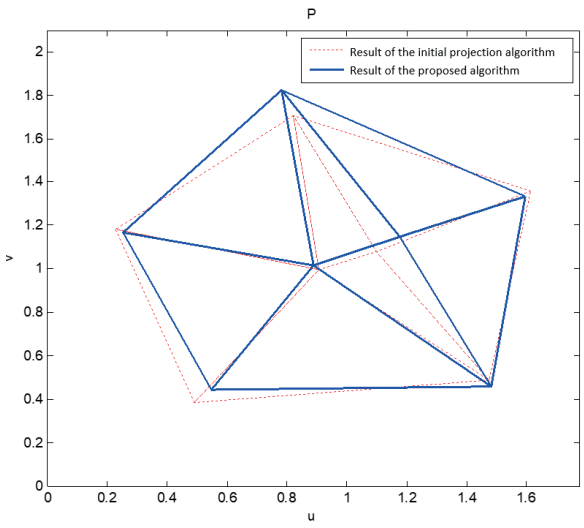


Fig. 8. Results of the initial projection and energy dissipation algorithms for the *Surf*1 surface.

Second, the intermediate step results and final outcome of the energy dissipation algorithm under specific conditions are presented, using the initial projection result generated for the *Surf*2 surface. The energy dissipation algorithm based on variable step size was executed with the parameters provided in Table IV.

Some intermediate outputs of the energy dissipation algorithm (Algorithm 1), executed using the parameters provided in Table IV, are visually presented in Fig. 9, with the corresponding numerical results provided in Table V.

TABLE IV
PARAMETERS OF THE ENERGY DISSIPATION ALGORITHM BASED ON VARIABLE STEP SIZE FOR THE *SURF*1 SURFACE PATCH

Parameters		
Description	**Symbol**	**Value**
Spring constant	C	0.5
Area density	p	2
Allowed area error percentage	$\emptyset_S$	0.30
Allowed shape error percentage	$\emptyset_C$	0.25
Allowed energy variation percentage	ϵ	0.05
Maximum allowed iterations	N_{max}	150

TABLE V
SHAPE AND AREA ERROR VALUES CORRESPONDING TO THE ENERGY DISSIPATION RESULTS SHOWN IN FIG. 9

Iteration #	Shape error, E_C	Area error, E_S
1	0.263	0.479
10	0.261	0.460
20	0.253	0.437
40	0.239	0.386
60	0.242	0.330
70	0.248	0.298

The initial projection result for the *Surf*2 surface, subjected to the energy dissipation algorithm (executed with the parameters from Table IV), was further improved after 70 iterations, with an area error value of $E_S = 0.298$ and a shape error value of $E_C = 0.248$, compared to the initial values of $E_S = 0.490$ and $E_C = 0.268$. The results of the initial projection algorithm and the improvement achieved by the energy dissipation algorithm for the sample surface are overlaid in the graph in Fig. 10.

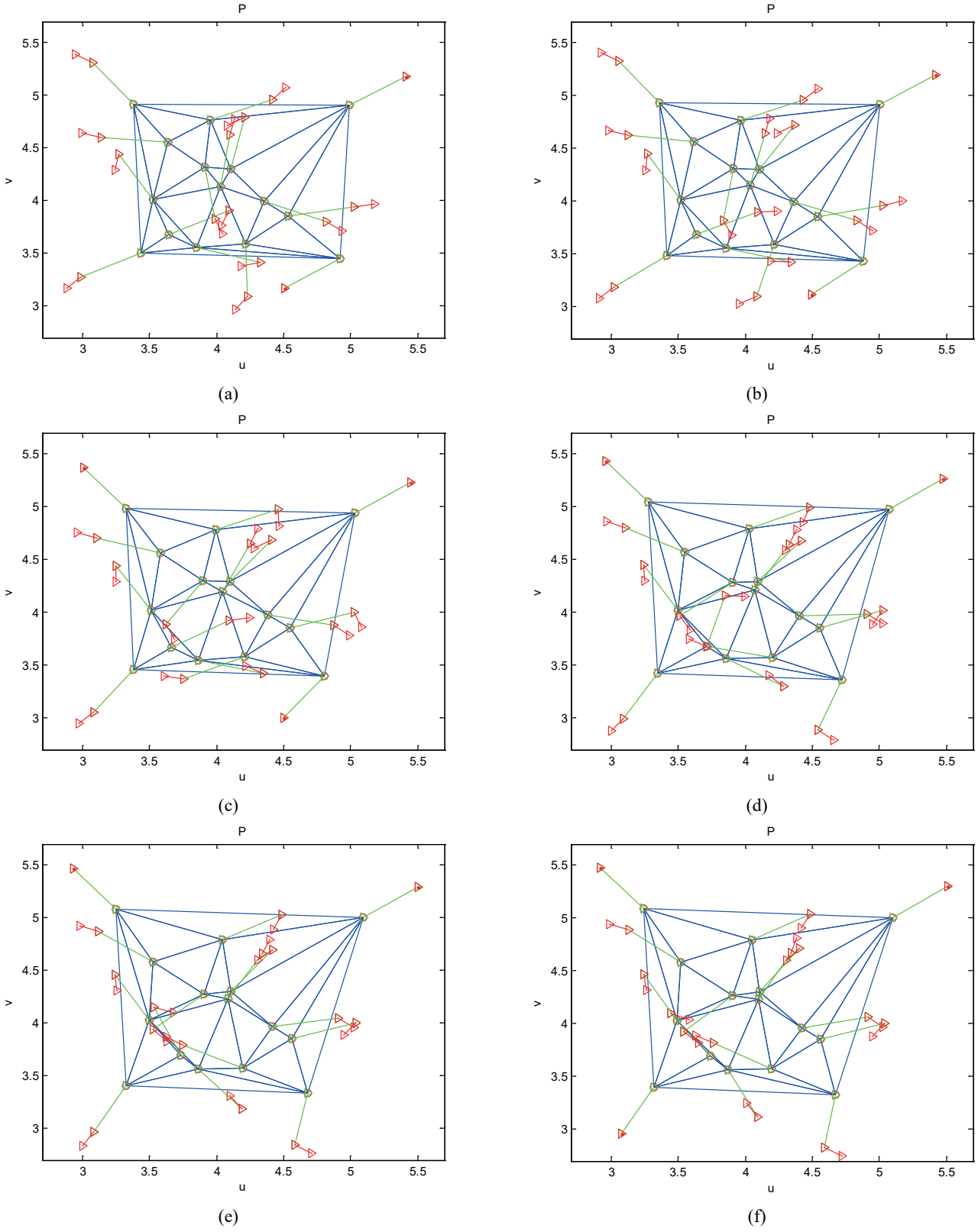
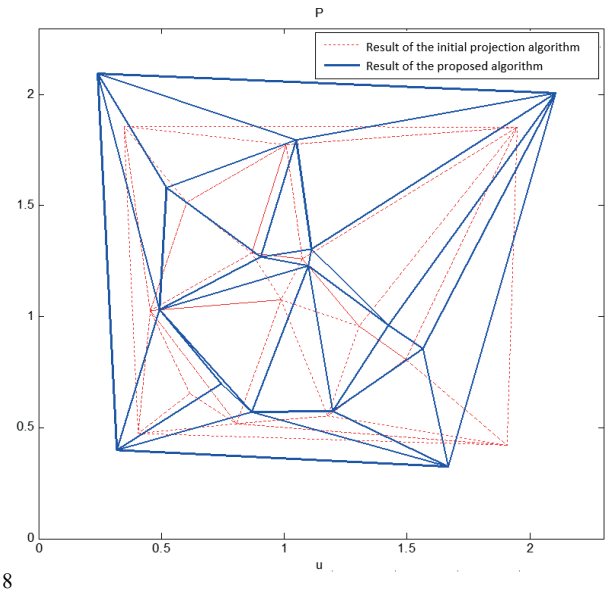


Fig. 9. Outputs of the energy dissipation algorithm at various steps for the *Surf*2 surface: (a) iteration #1; (b) iteration #10; (c) iteration #20; (d) iteration #40; (e) iteration #60; (f) iteration #70.



8

Fig. 10. Results of the initial projection and energy dissipation algorithms for the *Surf*2 surface.

B. Effect of Variable Step Size on Energy Dissipation Time

In the energy-based flattening method, as mentioned earlier in Section 2, the equation solving process using the Euler method was modified to incorporate a variable step size instead of a fixed step size (Δt). This modification enables the energy dissipation process to be completed more quickly (i.e., in fewer iterations) compared to the base model. For the *Surf*1 and *Surf*2 surface patches, the number of iterations required for energy dissipation, whether using a fixed or variable step size, along with the resulting final relative area (E_S) and shape (E_C) errors, are provided in Tables VI and VII. Additionally, the performance of the fixed and variable step energy dissipation methods for the sample surface patches is illustrated in Fig. 11 for *Surf*1 and Fig. 12 for *Surf*2. As shown in the graphs, the proposed variable step-size method reduced the number of iterations from 70 to 36 for the *Surf*1 surface patch and from 112 to 70 for *Surf*2, effectively halving the required iterations.

TABLE VI

ERROR VALUES FOR THE ENERGY DISSIPATION PROCESS BASED ON FIXED AND VARIABLE STEP SIZES FOR THE SURFACE *SURF*1

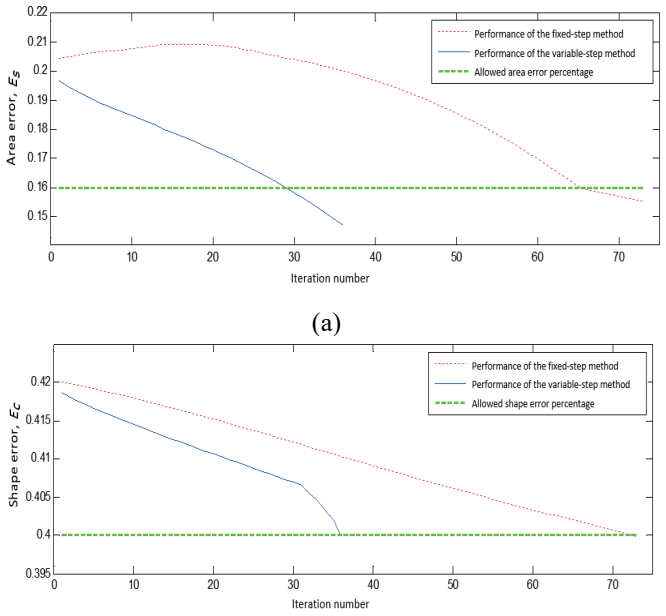
Fixed step size ($\Delta t = 0.01$)		**Variable step size**	
Iteration number	**Shape and area errors**	**Iteration number**	**Shape and area errors**
1	$E_C = 0.420$, $E_S = 0.204$	1	$E_C = 0.419$, $E_S = 0.197$
2	$E_C = 0.420$, $E_S = 0.205$	2	$E_C = 0.418$, $E_S = 0.195$
3	$E_C = 0.420$, $E_S = 0.205$	3	$E_C = 0.417$, $E_S = 0.193$
4	$E_C = 0.419$, $E_S = 0.206$	4	$E_C = 0.417$, $E_S = 0.192$
5	$E_C = 0.419$, $E_S = 0.206$	5	$E_C = 0.417$, $E_S = 0.191$
6	$E_C = 0.419$, $E_S = 0.207$	6	$E_C = 0.416$, $E_S = 0.189$
7	$E_C = 0.419$, $E_S = 0.207$	7	$E_C = 0.416$, $E_S = 0.188$
8	$E_C = 0.419$, $E_S = 0.207$	8	$E_C = 0.415$, $E_S = 0.187$
9	$E_C = 0.418$, $E_S = 0.207$	9	$E_C = 0.415$, $E_S = 0.186$
10	$E_C = 0.418$, $E_S = 0.208$	10	$E_C = 0.415$, $E_S = 0.185$
11	$E_C = 0.418$, $E_S = 0.208$	11	$E_C = 0.414$, $E_S = 0.184$
12	$E_C = 0.417$, $E_S = 0.209$	12	$E_C = 0.414$, $E_S = 0.182$
13	$E_C = 0.417$, $E_S = 0.209$	13	$E_C = 0.413$, $E_S = 0.181$
14	$E_C = 0.417$, $E_S = 0.209$	14	$E_C = 0.413$, $E_S = 0.180$
15	$E_C = 0.417$, $E_S = 0.209$	15	$E_C = 0.413$, $E_S = 0.179$
16	$E_C = 0.416$, $E_S = 0.209$	16	$E_C = 0.412$, $E_S = 0.178$
17	$E_C = 0.416$, $E_S = 0.209$	17	$E_C = 0.412$, $E_S = 0.177$
18	$E_C = 0.416$, $E_S = 0.209$	18	$E_C = 0.411$, $E_S = 0.176$
19	$E_C = 0.416$, $E_S = 0.209$	19	$E_C = 0.411$, $E_S = 0.174$
20	$E_C = 0.415$, $E_S = 0.209$	20	$E_C = 0.411$, $E_S = 0.173$
21	$E_C = 0.415$, $E_S = 0.209$	21	$E_C = 0.410$, $E_S = 0.172$
22	$E_C = 0.414$, $E_S = 0.208$	22	$E_C = 0.410$, $E_S = 0.170$
23	$E_C = 0.414$, $E_S = 0.208$	23	E_C=0.410, E_S=0.169
24	$E_C = 0.414$, $E_S = 0.208$	24	$E_C = 0.409$, $E_S = 0.168$
25	$E_C = 0.414$, $E_S = 0.207$	25	$E_C = 0.409$, $E_S = 0.166$
26	$E_C = 0.413$, $E_S = 0.206$	26	$E_C = 0.409$, $E_S = 0.165$
27	$E_C = 0.413$, $E_S = 0.206$	27	$E_C = 0.408$, $E_S = 0.163$
28	$E_C = 0.413$, $E_S = 0.205$	28	$E_C = 0.408$, $E_S = 0.162$
29	$E_C = 0.413$, $E_S = 0.205$	29	$E_C = 0.407$, $E_S = 0.156$
30	$E_C = 0.412$, $E_S = 0.204$	30	$E_C = 0.407$, $E_S = 0.158$
31	$E_C = 0.412$, $E_S = 0.203$	31	$E_C = 0.407$, $E_S = 0.157$
32	$E_C = 0.412$, $E_S = 0.203$	32	$E_C = 0.406$, $E_S = 0.155$
33	$E_C = 0.411$, $E_S = 0.202$	33	$E_C = 0.405$, $E_S = 0.153$
34	$E_C = 0.411$, $E_S = 0.202$	34	$E_C = 0.403$, $E_S = 0.151$
35	$E_C = 0.411$, $E_S = 0.201$	35	$E_C = 0.402$, $E_S = 0.149$
36	$E_C = 0.410$, $E_S = 0.200$	**36**	$E_C =$ **0.400**, $E_S =$ **0.147**
.	.		
.	.		
.	.		
71	$E_C = 0.401$, $E_S = 0.156$		
72	$E_C = 0.401$, $E_S = 0.156$		
73	$E_C =$ **0.400**, $E_S =$ **0.155**		

TABLE VII

ERROR VALUES FOR THE ENERGY DISSIPATION PROCESS BASED ON FIXED AND VARIABLE STEP SIZES FOR THE SURFACE *SURF*2

Fixed step size ($\Delta t = 0.01$)		**Variable step size**	
Iteration number	**Shape and area errors**	**Iteration number**	**Shape and area errors**
1	$E_C = 0.269$, $E_S = 0.480$	1	$E_C = 0.263$, $E_S = 0.479$
2	$E_C = 0.274$, $E_S = 0.475$	2	$E_C = 0.269$, $E_S = 0.474$
3	$E_C = 0.273$, $E_S = 0.474$	3	$E_C = 0.268$, $E_S = 0.472$
4	$E_C = 0.273$, $E_S = 0.472$	4	$E_C = 0.266$, $E_S = 0.471$
5	$E_C = 0.272$, $E_S = 0.471$	5	$E_C = 0.266$, $E_S = 0.469$
6	$E_C = 0.272$, $E_S = 0.471$	6	$E_C = 0.265$, $E_S = 0.467$
7	$E_C = 0.272$, $E_S = 0.470$	7	$E_C = 0.264$, $E_S = 0.466$
8	$E_C = 0.271$, $E_S = 0.468$	8	$E_C = 0.263$, $E_S = 0.464$
9	$E_C = 0.271$, $E_S = 0.467$	9	$E_C = 0.263$, $E_S = 0.462$
10	$E_C = 0.271$, $E_S = 0.466$	10	$E_C = 0.261$, $E_S = 0.460$
11	$E_C = 0.270$, $E_S = 0.465$	11	$E_C = 0.261$, $E_S = 0.457$
12	$E_C = 0.269$, $E_S = 0.464$	12	$E_C = 0.260$, $E_S = 0.455$
13	$E_C = 0.269$, $E_S = 0.463$	13	$E_C = 0.259$, $E_S = 0.453$
14	$E_C = 0.268$, $E_S = 0.462$	14	$E_C = 0.258$, $E_S = 0.451$
15	$E_C = 0.268$, $E_S = 0.461$	15	$E_C = 0.257$, $E_S = 0.449$
16	$E_C = 0.267$, $E_S = 0.456$	16	$E_C = 0.257$, $E_S = 0.446$
17	$E_C = 0.266$, $E_S = 0.458$	17	$E_C = 0.256$, $E_S = 0.444$

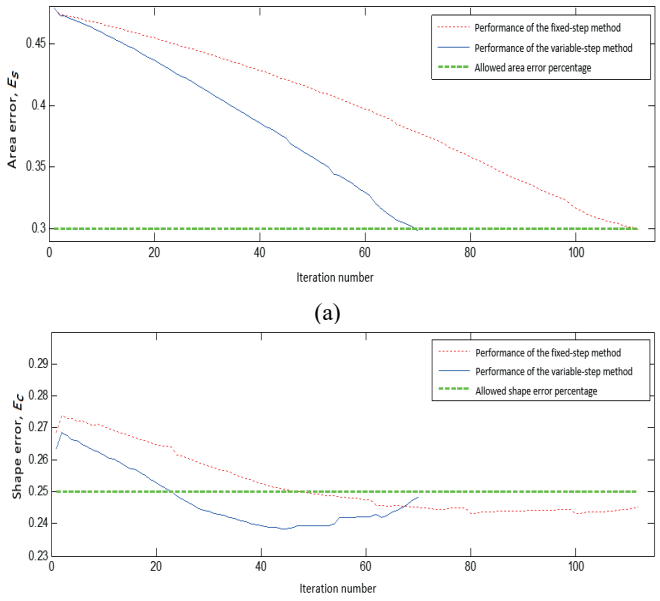
18	E_C = 0.266, E_S = 0.457	18	E_C = 0.255, E_S = 0.442
19	E_C = 0.265, E_S = 0.456	19	E_C = 0.254, E_S = 0.439
20	E_C = 0.265, E_S = 0.455	20	E_C = 0.253, E_S = 0.437
21	E_C = 0.264, E_S = 0.454	21	E_C = 0.252, E_S = 0.434
22	E_C = 0.265, E_S = 0.452	22	E_C = 0.251, E_S = 0432
. . .	. . .	. . .	. . .
		67	E_C = 0.245, E_S = 0.305
		68	E_C = 0.246, E_S = 0.303
		69	E_C = 0.247, E_S = 0.302
		70	E_C = **0.248**, E_S = **0.298**
107	E_C = 0.244, E_S = 0.305		
108	E_C = 0.244, E_S = 0.304		
109	E_C = 0.244, E_S = 0.303		
110	E_C = 0.244, E_S = 0.302		
111	E_C = 0.245, E_S = 0.300		
112	E_C = **0.245**, E_S = **0.299**		



(a)

(b)

Fig. 11. Performance of the fixed-step and variable-step energy dissipation methods for the *Surf*1 surface.



(a)

(b)

Fig. 12. Performance of the fixed-step and variable-step energy dissipation methods for the *Surf*2 surface.

V. Conclusion

This research leverages the energy model proposed by Wang et al. [26] to investigate the mechanical flattening process. Traditional energy dissipation methods use a fixed step size to calculate acceleration, velocity, and position. In contrast, this study employs a variable step-size methodology. A detailed analysis highlights the impact of this improved energy dissipation algorithm on reducing overall dissipation time. The findings demonstrate that the proposed method significantly lowers the number of required iterations, thereby greatly accelerating the energy dissipation process. Future research could explore implementing this algorithm in hardware, such as Field Programmable Gate Arrays (FPGAs) or Very Large Scale Integration (VLSI) circuits. Currently, the method is limited to complex lattice surfaces that can be approximated as convex or disk-shaped. Future efforts could focus on developing an adaptive algorithm capable of dividing intricate geometries into smaller, manageable sub-surfaces, which could then be individually flattened and reassembled to achieve the final flattened surface.

VI. Acknowledgement

This article is a part of the doctoral study titled ‘An Approach to Flattening of Complex Surfaces’, completed in the Department of Computer Engineering, Graduate School of Applied and Natural Sciences at Istanbul Commerce University.

REFERENCES

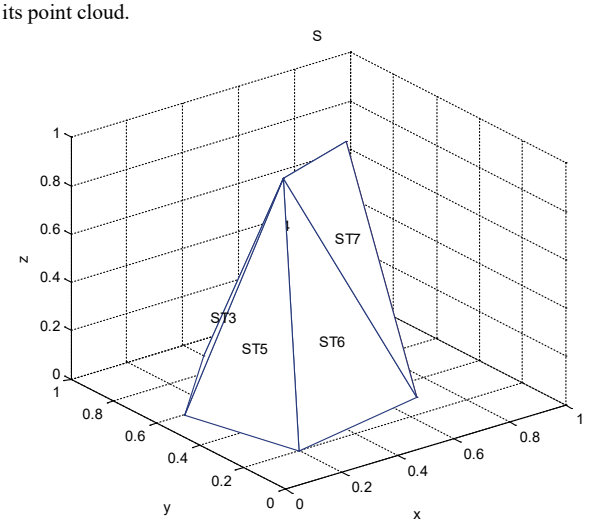
[1] C. C. L. Wang, *Geometric Modeling and Reasoning of Human-Centered Freeform Products*. London: Springer-Verlag, 2013. https://doi.org/10.1007/978-1-4471-4360-4
[2] M. P. Carmo, *Differential Geometry of Curves and Surfaces*. Englewood Cliffs: Prentice-Hall, 1976.
[3] H. S. Abdul-Rahman, S. Lou, W. Zeng, X. Jiang, and P. J. Scott, "Freeform texture representation and characterisation based on triangular mesh projection techniques," *Measurement*, vol. 92, pp. 172–182, Oct. 2016. https://doi.org/10.1016/j.measurement.2016.06.005
[4] P. Azariadis and N. Sapidis, "Planar development of digital free-form surfaces," in *Advances in Geometric Modeling*, M. Sarfraz, Ed. Wiley, Jan. 2004. https://doi.org/10.1002/0470860448.ch2
[5] M. Eigensatz, M. Kilian, A. Schiftner, N. J. Mitra, H. Pottmann, and M. Pauly, "Paneling architectural freeform surfaces," *ACM Trans. Graph.*, vol. 29, no. 4, Jul. 2010, Art. no. 45. https://doi.org/10.1145/1833351.1778782
[6] Y. J. Yang, W. Zeng, and X. X. Meng, "Conformal freeform surfaces," *Computer-Aided Design*, vol. 81, pp. 48–60, Dec. 2016. https://doi.org/10.1016/j.cad.2016.09.003
[7] W. Zhan, "Research on pattern flattening for parts with complex surfaces," M.S. thesis, Nanjing Univ. Aeronautics & Astronautics, Nanjing, China, 2007.
[8] R. Xu, X. Liu, and J. Chen, "Development and status of surfaces flattening method," *Die and Mould Technology*, no. 5, pp. 15–18, 2002.
[9] R. Sawhney and K. Crane, "Boundary first flattening," *ACM Trans. Graph. (ToG)*, vol. 37, no. 1, pp. 1–14, Dec. 2017. https://doi.org/10.1145/3132705
[10] Q. Zhang, J. Hou, Y. Qian, Y. Zeng, J. Zhang, and Y. He, "Flattening-net: Deep regular 2D representation for 3D point cloud analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9726–9742, Feb. 2023. https://doi.org/10.1109/TPAMI.2023.3244828
[11] M. Xiao, Z. Qi, and H. Shi, "The surface flattening based on mechanics revision of the tunnel 3D point cloud data from laser scanner," *Procedia Computer Science*, vol. 131, pp. 1229–1237, 2018. https://doi.org/10.1016/j.procs.2018.04.335
[12] X. Ji, G. Wen, H. Gong, R. Sun, and H. Li, "Three-dimensional wound flattening method for mapping skin mechanical properties based on finite element method," *Comput. Methods Biomech. Biomed. Engin.*, vol. 27, no. 2, pp. 237–250, Feb. 2023. https://doi.org/10.1080/10255842.2023.2183347
[13] P. Zheng, Q. Liu, J. Lou, C. Lian, and D. Lin, "A free-form surface flattening algorithm that minimizes geometric deformation energy," *IET Image Processing*, vol. 16, no. 9, pp. 2544–2556, Jul. 2022. https://doi.org/10.1049/ipr2.12508
[14] B. Yi, Y. Yang, R. Zheng, X. Li, and M. Yi, "Triangulated surface flattening based on the physical shell model," *J. Mech. Sci. Technol.*, vol. 32, pp. 2163–2171, May 2018. https://doi.org/10.1007/s12206-018-0425-0
[15] J. McCartney, B. K. Hinds, B. L. Seow, and D. Gong, "An energy based model for the flattening of woven fabrics," *J. Mater. Process. Technol.*, vol. 107, no. 1–3, pp. 312–318, Nov. 2000. https://doi.org/10.1016/S0924-0136(00)00694-4
[16] C. H. Lee and H. Huh, "Blank design and strain estimates for sheet metal forming processes by a finite element inverse approach with initial guess of linear deformation," *J. Mater. Process. Technol.*, vol. 82, no. 1–3, pp. 145–155, 1998. https://doi.org/10.1016/S0924-0136(98)00034-X
[17] J. Lan, X. Dong, and Z. Li, "Blank and strain estimates for sheet metal forming processes by an inverse approach of FE," *J. Plast. Eng.*, vol. 8, pp. 60–62, 2001.
[18] K. Chung and O. Richmond, "Ideal forming–I. Homogeneous deformation with minimum plastic work," *Int. J. Mech. Sci.*, vol. 34, no. 7, pp. 575–591, Jul. 1992. https://doi.org/10.1016/0020-7403(92)90032-C
[19] K. Chung and O. Richmond, "Ideal forming–II. Sheet forming with optimum deformation," *Int. J. Mech. Sci.*, vol. 34, no. 8, pp. 617–633, Aug. 1992. https://doi.org/10.1016/0020-7403(92)90059-P
[20] K. Chung, J. W. Yoon, and O. Richmond, "Ideal sheet forming with frictional constraints," *Int. J. Plasticity*, vol. 16, no. 6, pp. 595–610, May 2000. https://doi.org/10.1016/S0749-6419(99)00068-6
[21] W. Chung, S. H. Kim, and K. H. Shin, "A method for planar development of 3D surfaces in shoe pattern design," *J. Mech. Sci. Technol.*, vol. 22, no. 8, pp. 1510–1519, 2008. https://doi.org/10.1007/s12206-008-0609-0
[22] Q. L. Zhang and X. Q. Luo, "Finite element method for developing arbitrary surfaces to flattened forms," *Finite Elem. Anal. Des.*, vol. 39, no. 10, pp. 977–984, Jul. 2003. https://doi.org/10.1016/S0168-874X(02)00142-7
[23] J. W. Yoon, K. Chung, F. Pourboghrat, and F. Barlat, "Design optimization of extruded preform for hydroforming processes based on ideal forming design theory," *Int. J. Mech. Sci.*, vol. 48, no. 12, pp. 1416–1428, Dec. 2006. https://doi.org/10.1016/j.ijmecsci.2006.07.003
[24] H. Wang and C. C. Wang, "A general surface development algorithm based on energy model," *J. Comput.-Aided Des. Comput. Graph.*, vol. 13, no. 6, p. 556, 2001.
[25] Z. Cai, M. Li, and W. Guo, "Calculating blank of shell-like parts based on triangular mesh system," *Eng. Technol. Ed.*, vol. 32, no. 2, 2002.
[26] C. C. Wang, S. S. Smith, and M. M. Yuen, "Surface flattening based on energy model," *Comput.-Aided Des.*, vol. 34, no. 11, pp. 823–833, Sep. 2002. https://doi.org/10.1016/S0010-4485(01)00150-6
[27] S. Li, X. Liu, X. Zheng, and H. Lu, "An overview of flattening methods for complex surface," *J. Inf. Comput. Sci.*, vol. 11, no. 1, pp. 323–333, 2014.
[28] K. H. Shin, "A method for planar development of free-form surfaces made of anisotropic materials," *J. Mech. Sci. Technol.*, vol. 25, no. 11, pp. 2817–2825, Nov. 2011. https://doi.org/10.1007/s12206-011-0726-z
[29] Q. Liu, J. Xi, and Z. Wu, "An energy-based surface flattening method for flat pattern development of sheet metal components," *Int. J. Adv. Manuf. Technol.*, vol. 68, no. 5–8, pp. 1155–1166, Apr. 2013. https://doi.org/10.1007/s00170-013-4908-y
[30] E. Yavuz and R. Yazici, "A dynamic neural network model for accelerating preliminary parameterization of 3D triangular mesh surfaces," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3691–3701, Jan. 2018. https://doi.org/10.1007/s00521-017-3332-x
[31] E. Yavuz, R. Yazici, M. C. Kasapbasi, and T. T. Bilgin, "Improving initial flattening of convex-shaped free-form mesh surface patches using a dynamic virtual boundary," *Comput. Syst. Sci. Eng.*, vol. 34, no. 6, pp. 339–355, 2019. https://doi.org/10.32604/csse.2019.34.339
[32] D. N. Metaxas, *Physics-Based Deformable Models*. Dordrecht: Kluwer Academic, 1997. https://doi.org/10.1007/978-1-4615-6335-8

Erdem Yavuz received his PhD degree in Computer Engineering from Istanbul Commerce University in 2018. He is currently an Associate Professor at the Department of Computer Engineering, Bursa Technical University, Türkiye. His research interests include speech processing, biomedical signal processing, machine learning, artificial neural systems, information security and cryptology.
E-mail: erdem.yavuz@btu.edu.tr
ORCID iD: https://orcid.org/0000-0002-3159-2497

Rıfat Yazıcı received his PhD degree from the School of Engineering and Applied Sciences, University of Sussex, United Kingdom, in 1983. He served as Head of the Department of Computer Engineering at Karadeniz Technical University, Türkiye, from 1994 to 1997 and from 2002 to 2010, and as Head of the Department of Computer Engineering at Istanbul Commerce University, Türkiye, from 2010 to 2024. He is currently a Professor at the Department of Computer Engineering at Fenerbahçe University, Türkiye. His research interests include information security and reliability, computer graphics, hardware, artificial intelligence, software, machine learning and pattern recognition.
E-mail: rifat.yazici@fbu.edu.tr
ORCID iD: https://orcid.org/0000-0003-0670-8803

APPENDIX

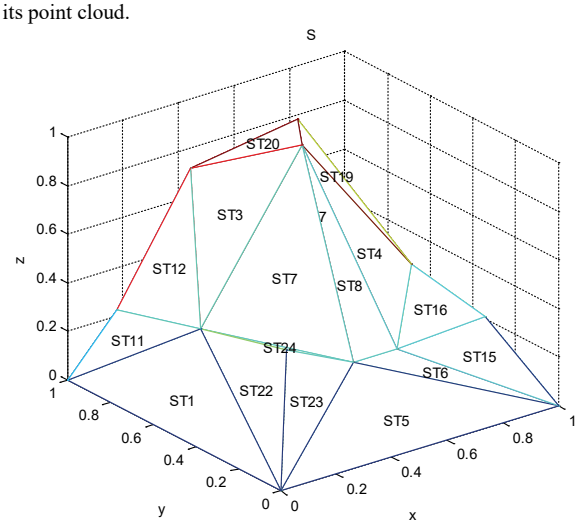
A.1. Plotting of an arbitrary sample surface consisting of 7-vertices (*Surf*1) and its point cloud.



The point cloud representing the vertices of the surface mesh, denoted as X, along with the data defining the triangular faces, referred to as the topology information F, is presented as follows:

$$X=\begin{bmatrix}0.3 & 0.4 & 1\\ 0.6 & 0.5 & 1\\ 0.2 & 0.2 & 0\\ 0.1 & 0.6 & 0\\ 0.4 & 0.9 & 0\\ 0.8 & 0.7 & 0\\ 0.7 & 0.3 & 0\end{bmatrix}, F=\begin{bmatrix}6 & 5 & 2\\ 6 & 2 & 7\\ 1 & 5 & 4\\ 1 & 2 & 5\\ 4 & 3 & 1\\ 3 & 7 & 1\\ 1 & 7 & 2\end{bmatrix}.$$

A.2. Plotting of an arbitrary sample surface consisting of 15-vertices (*Surf*2) and its point cloud.



The point cloud representing the vertices of the surface mesh, denoted as X, along with the data defining the triangular faces, referred to as the topology information F, is presented as follows:

$$X=\begin{bmatrix}0.1 & 0.1 & 0.5\\ 0.1 & 0.9 & 0.3\\ 0.65 & 0.82 & 0.45\\ 0.7 & 0.3 & 0.55\\ 0.3 & 0.05 & 0.4\\ 0.85 & 0.15 & 0.35\\ 0.5 & 0.55 & 1\\ 0.6 & 0.7 & 1\\ 0.08 & 0.48 & 0.42\\ 0.48 & 0.08 & 0.38\\ 0.25 & 0.75 & 0.9\\ 0 & 0 & 0\\ 0 & 1 & 0\\ 1 & 1 & 0\\ 1 & 0 & 0\end{bmatrix}, F=\begin{bmatrix}9 & 13 & 12\\ 14 & 13 & 3\\ 11 & 9 & 7\\ 7 & 10 & 4\\ 12 & 15 & 5\\ 15 & 10 & 5\\ 7 & 9 & 5\\ 5 & 10 & 7\\ 2 & 3 & 13\\ 2 & 11 & 3\\ 13 & 9 & 2\\ 9 & 11 & 2\\ 6 & 15 & 14\\ 14 & 4 & 6\\ 6 & 10 & 15\\ 6 & 4 & 10\\ 14 & 3 & 8\\ 8 & 4 & 14\\ 7 & 4 & 8\\ 8 & 11 & 7\\ 3 & 11 & 8\\ 1 & 9 & 12\\ 12 & 5 & 1\\ 1 & 5 & 9\end{bmatrix}$$