J. Appl. Comput. Mech., 12(4) (2026) 1330-1347
DOI: 10.22055/jacm.2025.48560.5325

ISSN: 2383-4536
jacm.scu.ac.ir

Shahid Chamran University of Ahvaz

Journal of Applied and Computational Mechanics

Research Paper

# Finite Element Algorithm for Surface Flattening

**Ahmad Kamal, Sayed Hossein Dibajian**

Department of Mechanical and Energy Engineering, Shahid Beheshti University, Tehran, Iran,
Email: a_kamal@sbu.ac.ir (A.K.); h_dibajian@sbu.ac.ir (S.H.D.)

Received July 01 2025; Revised September 27 2025; Accepted for publication November 07 2025.
✉ Corresponding author: S.H. Dibajian (h_dibajian@sbu.ac.ir)
© 2025 Published by Shahid Chamran University of Ahvaz

**Abstract.** Membrane and inflatable structures are generally composed of non-developable geometries that must be approximated from planar surfaces. This study proposes a novel finite element–based algorithm for surface flattening, specifically tailored for structural form-finding and analysis. The method regulates edge length and curvature through truss and beam elements, ensuring boundary compatibility during surface flattening. Unlike conventional geometric or mesh-parameterization techniques, the proposed approach is fully integrated into the finite element framework, enabling direct structural simulation. The algorithm has been implemented as an Abaqus plugin, which automatically generates flattened surfaces and reconstructs the final geometry from assembled planar patches. Beyond geometric accuracy, the plugin leverages Abaqus' nonlinear capabilities to evaluate stresses under applied pressure, providing a unique and practical tool for analyzing the strength and stability of inflatable structures.

**Keywords:** Abaqus plugin, Surface flattening, Finite element method, Fabric structures, Inflatable structures.

## 1. Introduction

Structures like ship hulls, clothing, inflatable structures, and fabrics are constructed by joining planar parts. In their design, it is crucial to determine the optimal size and shape of these parts. This consideration also applies in reverse, where converting a 3D surface into a 2D one is known as surface flattening. After identifying the 2D surfaces that make up the structure, the parts must be connected and the overall structure mechanically analyzed. A developable surface can be transformed into a two-dimensional form without distortion if its Gaussian curvature is zero at all points, as stated in Eq. (1) [1]:

$$K = \kappa_1 \kappa_2 = 0 \tag{1}$$

In this equation, $K$ represents the Gaussian curvature, while $\kappa_1$ and $\kappa_2$ denote the surface's principal curvatures, which are the maximum and minimum curvatures at a point. If one of the principal curvatures of a surface is zero at all points, the Gaussian curvature is also zero, allowing the surface to be mapped to a two-dimensional plane without distortion, shrinkage, or cuts. For instance, a cylinder has zero Gaussian curvature and can be flattened into a two-dimensional surface, whereas a sphere, with non-zero curvature, cannot be mapped without distortion. Distortion or shrinkage occurs when the lengths of curves on the surface change during the mapping process.

The main goal of flattening a three-dimensional shell surface is to minimize distortion and shrinkage, ensure compliance with the target geometry, and reduce residual stresses during production. This process has various applications in mechanical engineering. For example, in sheet metal processes, achieving an optimal two-dimensional surface is essential for deformation with minimal waste and residual stress. In the shipbuilding industry, the required two-dimensional geometry to achieve the curvature of the ship's hull must also be considered. Surface flattening in fabric structures plays a significant role in inflatable structures, which are increasingly used in various applications. Their design and analysis have gained attention due to their incorporation in new architecture, inflatable toys, advertising devices, and military decoys. Among the earliest uses of inflatable structures are airships, employed for weather forecasting, ground observation, and data transmission [2]. Increased attention to the equipment and its diverse applications underscores the importance of designing and constructing its various parts.

Various methods have been proposed to design three-dimensional objects from planer plates made of different materials or to create flattened surfaces of three-dimensional shells. Common examples include origami objects [3], flexible sculptures [4], shoe uppers [1, 5] and ship hulls [6]. These studies focus on mapping the two-dimensional reference configuration to the final three-dimensional configuration, with the primary challenge being to identify the relationship between the two configurations to maximize similarity to the desired outcome. As the three-dimensional shape of the target becomes more intricate, the deformation from two-dimensional to three-dimensional becomes increasingly complex [7].

As authors knowledge, surface flattening methods fall into four categories: surface parameterization, mass and spring methods, finite element methods, and origami-based methods. In surface parameterization, a 3D surface is segmented into discrete parts, establishing a relationship with a planar 2D surface. By defining and minimizing a distortion criterion, the optimal mapping is achieved [8, 9].

Published online: November 10 2025

Triangular meshes are transformed into mass and spring systems in mass and spring methods. With appropriate damping and spring coefficients, this model can represent a two-dimensional surface equivalent to a three-dimensional one. Key advantages include ease of implementation, high speed, and low computational costs [10, 11]. Callens and Zadpoor [3] have scientifically explored the origami method for creating three-dimensional objects from surfaces, highlighting its applications and benefits in achieving curved geometries. While this technique can assist in manufacturing certain medical devices and satellite solar receivers, it is not suitable for objects that lack surface shrinkage.

Finite element techniques enable the conversion of 3D to 2D in finite element-based methods [12, 13]. Many researchers have studied the distortion criterion and the use of boundary conditions to manage boundary distortion [14-17]. Tam et al. [1] proposed a method specifically for producing shoe uppers, comparing the results with previous techniques. While this method is effective for shoe production, its limitations restrict its use in other applications. Shen et al. [18] proposed a surface flattening algorithm that minimizes the strain energy needed to convert plates into common shell shapes. Their article uses both shell and solid elements for finite element analysis, showing that while both yield similar results for spherical patterns, the outcomes are more accurate for three-dimensional cylindrical patterns.

Sheet metalworking is a crucial process in producing curved ship hulls from flat sheets, leading to numerous studies in this area. For instance, some research has focused on forming the shells through localized heating [6].

Identifying the basic shape required in the sheet-metal process involves flattening surfaces. A common yet imprecise method is graphical techniques, particularly for free-form metal sheets. To overcome this limitation, Liu et al. [19] have proposed a method to enhance and optimize the initial shell in the free formation of metal sheets.

The design of cable-membrane structures is another application of surface flattening that is used in architecture and civil engineering. One of the methods of analysis of these structures is dynamic relaxation [20]. This method is a step-by-step method for tracing the motion of a structure from the time of loading until it reaches equilibrium by considering damping effects. The main feature of this method is its efficiency and rapid convergence, especially in the analysis of large structures. Therefore, it is much more useful for large structures compared to the finite element method.

Despite numerous studies on transforming two-dimensional plates into three-dimensional shells, research on fabric structures like hot air balloons and airships is scarce, especially concerning materials with low flexural strength such as elastomers and fabrics. Skouras et al. [21] offer an interactive tool for designing and optimizing inflatable structures, allowing users to position joints and generate patterns while visualizing the final simulated shape. These studies assume high membrane strength and minimal strain, an assumption suitable only for small structures and not applicable to larger ones. Sosa et al. [22] used the finite element method to model the geometric shape of inflatable structures employed to block tunnels during inflation.

Based on the authors' examination, no comprehensive software exists for calculating expanded surfaces, final shape analysis, and stress analysis of fabric inflatable structures with complex loads and nonlinear materials. Although commercial software like SolidWorks is somewhat suitable for surface flattening, commercial finite element software is inadequate for analyzing fabric structures. This is due to their inability to flatten surfaces, despite having a good library for orthotropic, hyperelastic materials and elements for shells and membranes.

Among these, the use of finite element method and especially Abaqus software that utilizes this method has a special place. Ability to define and produce plugin in Abaqus software enables users to develop specialized plugin according to their requirements. Accordingly, various plugins have been prepared and presented in areas such as generating geometry, creating boundary conditions, analyzing or the finite element method, particularly through Abaqus software, plays a significant role due to its capability for plugin development tailored to user needs. Various plugins have been created for tasks like geometry generation, boundary condition setup, data analysis, and output visualization. Researchers have developed specialized plugins for specific purposes. For instance, Nesládek [23] introduced the Fatigue Prediction Utility, which uses another software to process data and predict component fatigue. Riaño developed a plugin for modeling composite microstructures with random fiber distributions [24]. Omairey [25] presented EasyPBC, a tool for estimating the elastic properties of composite materials directly within Abaqus without additional software. Another plugin detailed in [26], models and analyzes auxetic materials with diverse geometries.

For constructing fabric or inflatable structures, it's not enough to just flatten the surface; the compatibility of part boundaries is also crucial. This compatibility should be seen as a constraint when expanding the surfaces. During the second phase of fabric structure analysis, structures created using planar parts must undergo evaluation under loading and boundary conditions. While specialized software like Marvelous Designer and Gemini exists for clothing design, none are dedicated to inflatable structure analysis. This study introduces a new finite element method for controlling boundary distortion and deformation.

For this purpose, the first part of this paper examines the ability of finite element equations to model surface development. Although the finite element method has been applied by other researchers for this purpose, in the proposed method, the flattening process is accomplished solely by applying appropriate boundary conditions on the non-developable surface. By discretizing a 3D surface with membrane or shell elements and applying displacement conditions, the surface is transformed into 2D. Boundary conditions apply to only one degree of freedom, positioning nodes based on minimal strain energy. The comparison of surface flattening results obtained through this method with those from SolidWorks confirms its capability and accuracy. In this approach, even the imposition of boundary compatibility constraints is performed using a finite element–based technique. Truss and beam elements help adjust and control boundary lengths and distortion, akin to the penalty method. The proposed method, in addition to ensuring boundary compatibility, can be implemented in all finite element–based software such as Abaqus. By applying this method within software like Abaqus, one can take advantage of built-in features such as automatic time-step calculation, parallel processing, and others. Moreover, the approach can also be extended to other numerical techniques, including meshless methods and NURBS [27, 28].

In the second part of this paper, the possibility of analyzing inflatable structures reconstructed with developed surfaces using the finite element method is presented. Given the strong capability of the finite element method in analyzing membrane structures, accurate analysis of inflatable structures becomes feasible. In other words, the surface flattening technique employed in this study has been tailored for use within the finite element framework, enabling evaluation of the final form and structural strength of inflatable structures. Clearly, previous methods lacked the ability to analyze both the geometry and the strength of the final structure. Consequently, the proposed method makes it possible to conduct highly precise structural analysis. Furthermore, due to the application of appropriate essential boundary conditions, the analysis is computationally efficient and exhibits rapid convergence with respect to meshing.

This approach, implemented in Abaqus with an ISA (Inflatable Structure Analysis) plugin, leverages commercial software capabilities. The analysis of structures formed by joining 2D parts is crucial, yet specialized software for inflatable structure analysis is lacking. This article presents innovations in reconstructing and analyzing structures by connecting parts, ensuring boundary integrity, final shape, and stress distribution are achieved.

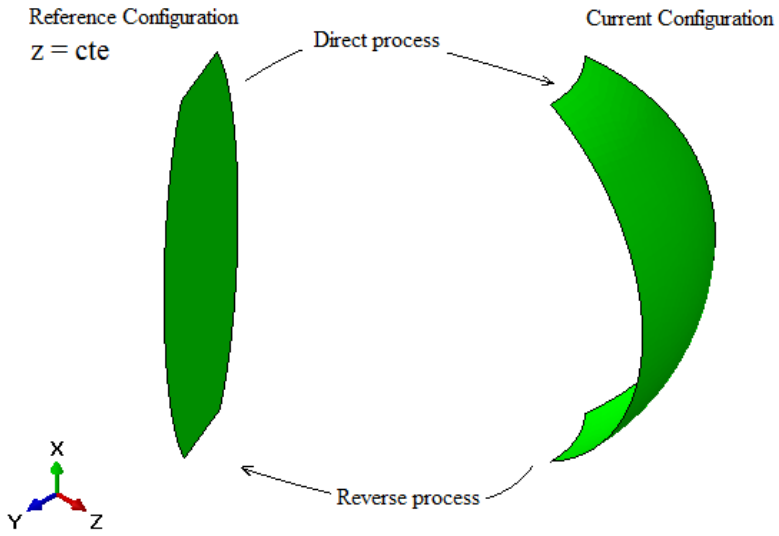



**Fig. 1.** Mapping from the reference configuration (two-dimensional) to the current configuration (three-dimensional).

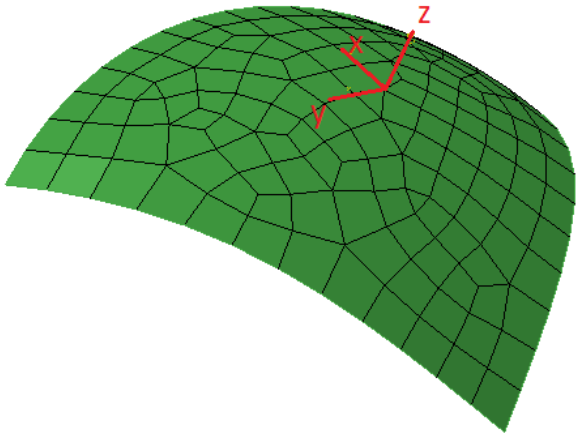



**Fig. 2.** Finding middle element in a part.

## 2. Concepts and Plugin Algorithm

### 2.1. Theory

Converting a two-dimensional plane into a three-dimensional shell, as illustrated in Fig. 1, represents a deformation from the reference configuration (A) to the current configuration (B). External forces and boundary conditions transform plane A into surface B. The process of flattening is contrary to Fig. 1. In the designed algorithm for obtaining flattened surfaces, in the first step, the middle element of the model is identified. The middle element is selected so that the total distance between its center and the center of the other elements of a part is minimal. Therefore, by calculating of Eq. (2) for all elements of one part, the element with the minimum value is selected as the part's middle element:

$$norm_e = \sum_i \left( (x_i - x_e)^2 + (y_i - y_e)^2 + (z_i - z_e)^2 \right)$$
$$\min(norm_e) \rightarrow e \tag{2}$$

At this step, a new local coordinate system is established at the center point of the middle element based on Eq. (3), where the Z-axis is perpendicular to the element's two sides (Fig. 3):

$$\vec{r}_1 = \begin{bmatrix} x_j - x_i \\ y_j - y_i \\ z_j - z_i \end{bmatrix}, \quad \vec{r}_2 = \begin{bmatrix} x_k - x_i \\ y_k - y_i \\ z_k - z_i \end{bmatrix}$$
$$\hat{e}_1 = \frac{\vec{r}_1}{|\vec{r}_1|}, \quad \hat{e}_3 = \frac{\hat{e}_1 \times \vec{r}_2}{|\vec{r}_2|} \tag{3}$$
$$\hat{e}_2 = \hat{e}_3 \times \hat{e}_1$$

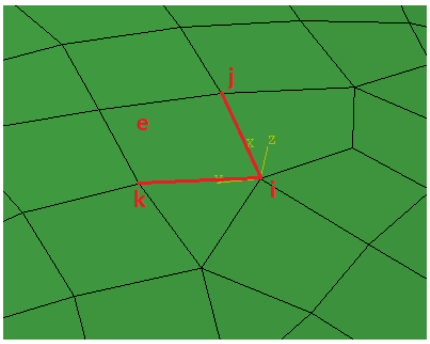
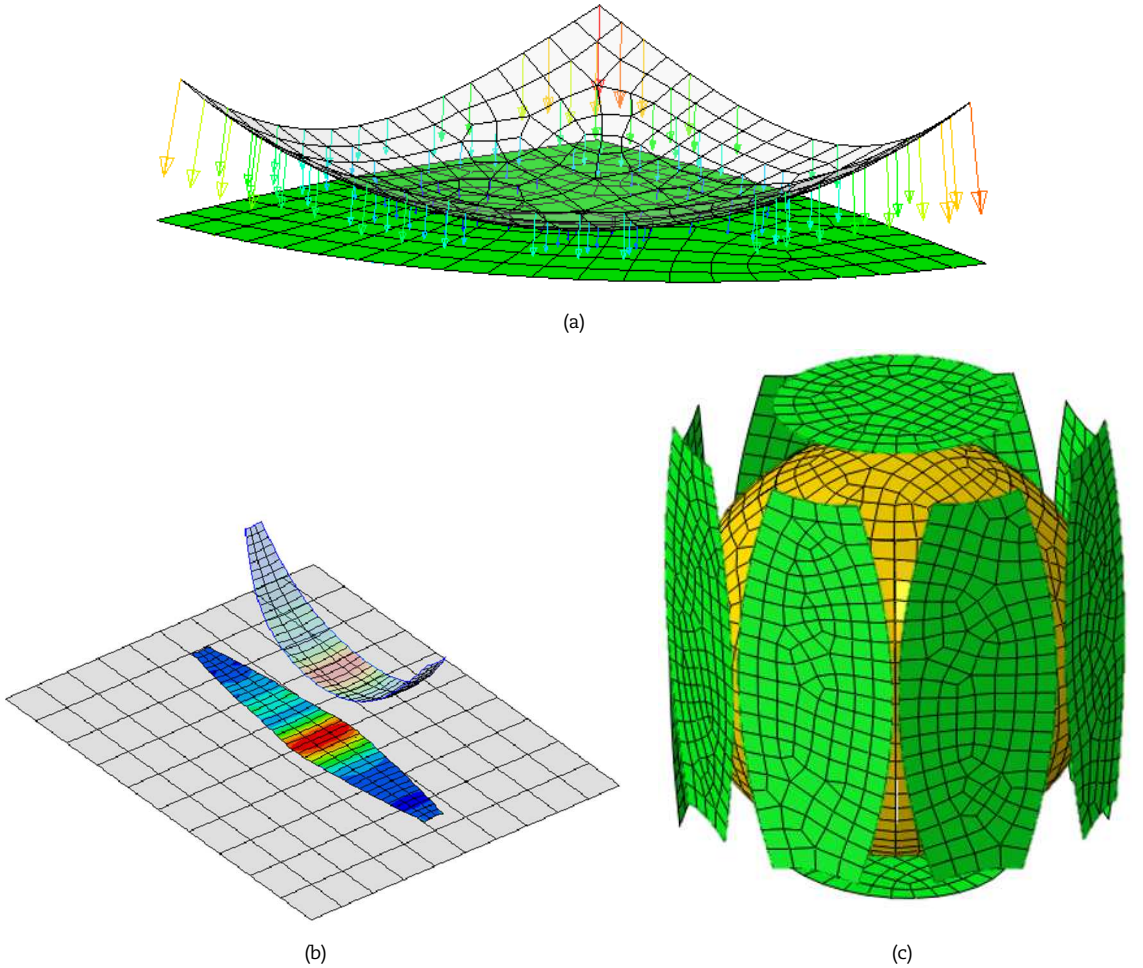



**Fig. 3.** Local coordinate system at the center point of the middle element.



(a)

(b) (c)

**Fig. 4.** (a) Shell transformation to the X-Y plane of the local coordinates and mapping to a two-dimensional plane, (b) Transformation of a 3D shell to a 2D plane of: a sector of a sphere, (c) A partitioned sphere.

The displacement boundary condition for each node in this local system is defined such that all Z coordinates become zero after deformation (Eq. (4)). Consequently, the primary shell is transformed to the X-Y plane of the local coordinates and mapped to a two-dimensional plane (Fig. 4). This two-dimensional surface is the flattened area to be cut for the final fabric structure:

$$U_x = UNSET$$

$$U_y = UNSET \tag{4}$$

$$U_z = -\bar{Z} \rightarrow Z \text{ Coordinate in the local coordinate system}$$

It should be emphasized that the boundary conditions presented in Eq. (4) are applied to every node. Therefore, due to the constraint of all nodes and the nature of the boundary conditions, buckling will not occur. In finite element problems, the goal is typically to transition from geometry A to B. In these cases, the necessary finite element equations are derived by minimizing the total strain energy ($\int [\sigma]^T [D][\varepsilon] d\Omega$).

The principle of virtual work for variables in the current configuration is expressed by [29]:

$$\int_V \sigma : \delta d dV + \int_V \rho \ddot{u} : \delta \dot{u} dV = \int_S t : \delta \dot{u} dS + \int_V f : \delta \dot{u} dV \tag{5}$$

where $\ddot{u}$ is the acceleration, $\dot{u}$ is the velocity, t is the surface traction, and $f$ is the body load. Integration is performed over current volume $V$ and surface area $S$.

In a Lagrangian formulation the transformation of Eq. (5) can be shown to give:

$$\int_{V_0} S : \delta \dot{E} dV_0 + \int_{V_0} \rho_0 \ddot{u} : \delta \dot{u} dV_0 = \int_{S_0} t^0 : \delta \dot{u} dS_0 + \int_{V_0} f^0 : \delta \dot{u} dV_0 \tag{6}$$

In this equation 0 index has been assigned to variables measured in the reference configuration. The variables $t^0$ and $f^0$ are pseudo forces per unit area and volume, respectively, in the reference configuration.

The strain rate and virtual strain rate in matrix notation can be expressed as follows:

$$\begin{gathered} \{\dot{E}\} = [\hat{B}]\{\dot{u}\} \\ \{\delta \dot{E}\} = [\hat{B}]\{\delta \dot{u}\} \end{gathered} \tag{7}$$

The strain matrix $[\hat{B}]$ can be divided in two parts of linear and nonlinear part:

$$[\hat{B}] = [B_L] + [B_{NL}] \tag{8}$$

In accordance with the small deformation theory the discretized dynamic equilibrium equations can be expressed as:

$$[f^{\text{int}}] + [M]\{\ddot{u}\} = [f^{ext}] \tag{9}$$

In this equation, the consistent mass matrix is:

$$[M] = \int_{V_0} \rho_0 [N]^T [N] dV_0 \tag{10}$$

and the external force vector is:

$$[f^{ext}] = \int_{V_0} [N]^T \{b_0\} dV_0 + \int_{S_0} [N]^T \{t_0\} dS_0 \tag{11}$$

and the internal force vector is:

$$[f^{\text{int}}] = \int_{V_0} [\hat{B}]^T \{S\} dV_0 \tag{12}$$

In quasi static problem, the inertia term is neglected. So, the simplified equations can be written:

$$\{\Psi\} = \int_{V_0} [\hat{B}]^T \{S\} dV_0 - \{f^{ext}\} = 0 \tag{13}$$

In this equation, $\{\Psi\}$ is a residual vector. To solve the resulting nonlinear equations, the Newton-Raphson iterative method is commonly employed. This involves linearizing Eq. (13) around the most recent solution. Taking the time derivative of the equilibrium Eq. (13) yields:

$$\{\dot{\Psi}\} = \int_{V_0} \left( [\hat{B}]^T \{\dot{S}\} + [\dot{\hat{B}}]^T \{S\} \right) dV_0 - \{\dot{f}^{ext}\} = 0 \tag{14}$$

By assuming the constitutive equations in the form:

$$\{\dot{S}\} = [D_T]\{\dot{E}\} \tag{15}$$

where $[D_T]$ is a matrix form of the constitutive tensor. It can then be shown that Eq. (14) can be rewritten as:

$$[\dot{\Psi}] = [K_T]\{\dot{u}\} - \{\dot{f}^{ext}\} = 0 \tag{16}$$

where,

$$[K_T] = [K_M] + [K_G] + [K_S] \tag{17}$$

In this equation, $[K_M]$ is called material stiffness matrix and is defined by:

$$[K_M] = \int_{V_0} \{B_L\}^T [D_T] \{B_L\} dV_0 \tag{18}$$

The second matrix, $[K_G]$, is the geometric stiffness matrix, arising from the nonlinear terms in the strain-displacement relation. The third matrix, $[K_S]$, is the initial stress stiffness matrix, dependent on the stress state in the current configuration [29].

For complex, nonlinear equilibrium equations, the relationships can be expressed as in Eq. (19), allowing the solution to be approximated using an explicit dynamic method, provided that the term $M\ddot{u}$ is sufficiently small:

$$[M]\{\ddot{u}\} = \{f^{ext}\} - \{f^{int}\} \tag{19}$$

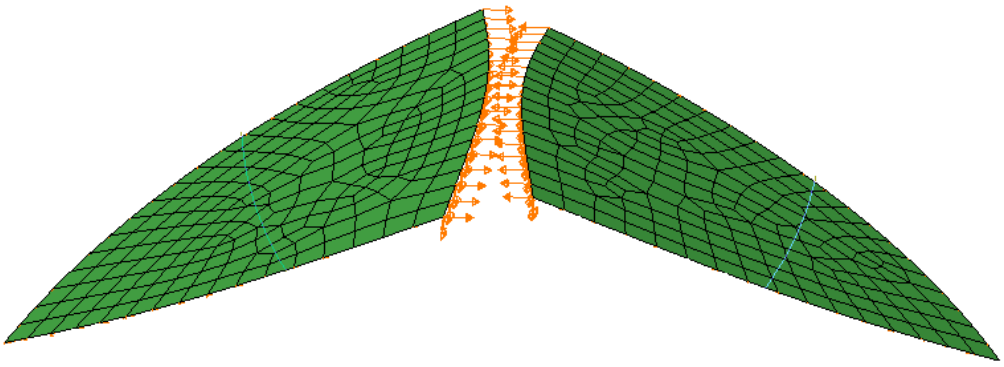
By obtaining $u$, the new location of the points in the current configuration can be obtained from the Eq. (20):

$$x = X + u \tag{20}$$

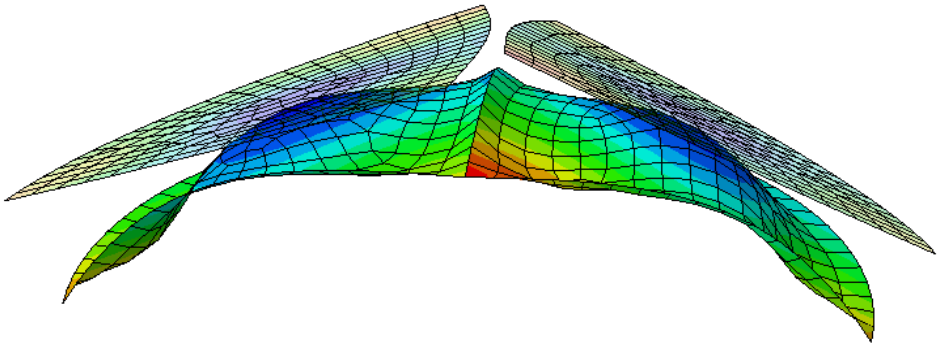
This article aims to establish the positions of points on a shell's reference configuration before loading, to define the two-dimensional geometry of a three-dimensional shell (current configuration) in a fabric structure. Unlike conventional finite element methods that seek to determine x this study focuses on obtaining $X$.

In linear materials and small strain conditions, the above process can be reversed. Specifically, the value of u in converting the final non-developable surface to a planar surface closely matches the displacement in converting the planar surface to the final surface. In fabric structures, despite large displacements, the expansion or strain is minimal. When boundary conditions are reversed in the nonlinear mode, the displacement can be approximated well from Eq. (14) due to the small expansions.

After finding the flattened surfaces, they should be connected and the actual geometry of the object should be determined. For this purpose, adjacent borders are defined before the flattening of the surfaces, so that they can be connected after surface flattening. In this algorithm, the nodes placed on the edges of the flattened surfaces are moved to the position before expansion and are released after tying the nodes to each other (Fig. 5). Thus, the final geometry will be reproduced. With the help of the obtained geometry, accurate stress-strain analyses can be performed using real loads. In addition, the quality of the algorithm can be evaluated.



(a)



(b)

**Fig. 5.** (a) Placing nodes to the position before flattening, (b) Tying nodes.

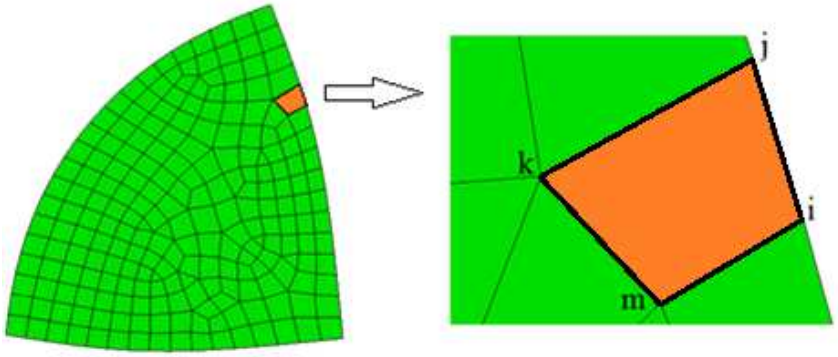



**Fig. 6.** A boundary element.

It is clear that, if a special condition is not imposed on the boundaries, rejoining the surfaces without stretching or wrinkling is impossible. Therefore, when the 3D surface is mapped to the 2D surface, in addition to minimizing the strain energy, the length of the boundaries and their curvatures must be controlled. Lagrange's method or the penalty method can be used to impose constraints on the length of the border. In this algorithm, a method similar to the penalty method is used. For a two-dimensional boundary element (Fig. 6), the stiffness matrix of the element in the finite elements is as Eq. (21):

$$K_e = \begin{pmatrix} K_{11} & \cdots & K_{18} \\ \vdots & \ddots & \vdots \\ K_{81} & \cdots & K_{88} \end{pmatrix} \tag{21}$$

Truss elements are used on the borders to control the distance between the border nodes. In the truss element, there is stiffness between boundary nodes $i$ and $j$ in direction $ij$ as Eq. (22). In this algorithm, the penalty coefficient is adjusted to the strain energy of the plate by controlling the value of A:

$$K_e = \frac{AE}{L}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \tag{22}$$

The same process is used for curvature control. A beam element is used to control the curvature at the boundaries. This element applies the following stiffness to the movement of nodes perpendicular to the boundary as well as changing the angle between the nodes. Similar to the above process, in this algorithm, the desired penalty coefficient is applied to the surface by setting the $I$ value:

$$K_e = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{23}$$

The described process is applied to all boundaries as shown in Figure 9. Therefore, the length and amount of curvature will be controlled. Based on the above, in the proposed method, the change in boundary lengths is indirectly controlled using a penalty approach. Beam and truss elements are employed to implement a finite element–based technique for enforcing boundary compatibility. Consequently, in accordance with the governing principles of the finite element method, convergence is guaranteed, and instabilities are prevented. The penalty coefficients are determined according to the material properties. Small values of these coefficients allow for controlled strain at the boundaries and can be adjusted depending on the user's requirements and the type of material. Conversely, large values ensure full boundary compatibility.

### 2.2. Surface flattening in ISA

The ISA plugin was created to analyze the geometry, flattening surfaces, and stress characteristics of a three-dimensional shell for fabric structures, utilizing Python within the Abaqus environment. Its graphical interface collects essential data from users, including the desired geometry, cutting lines, and material properties, then analyzes and presents the results accordingly.

The ISA plugin automates pre-processing and post-processing at various stages. To identify the two-dimensional flattened surfaces of an inflatable structure, the plugin first collects basic raw data from the user, including the target shape, material properties, solution time, mesh size, final pressure, and type of solution.

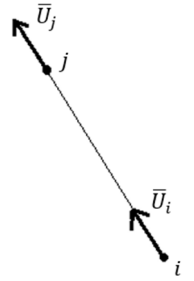



**Fig. 7.** A truss element.

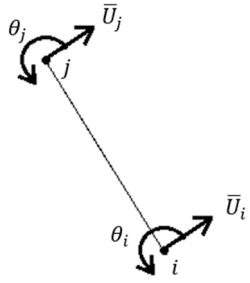



**Fig. 8.** A beam element.

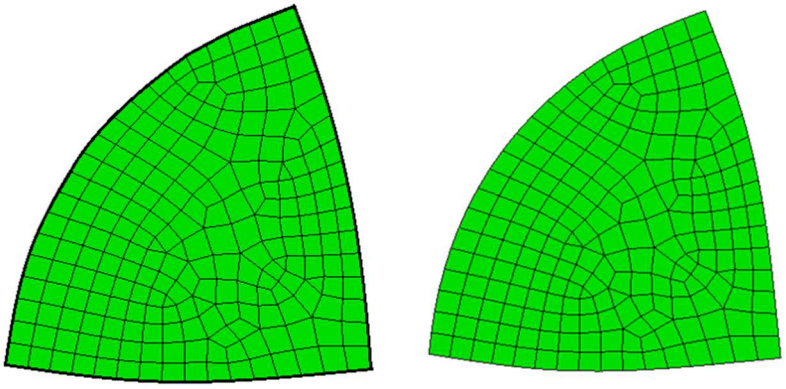


**Fig. 9.** Using truss and beam element to control boundaries length and curvature.

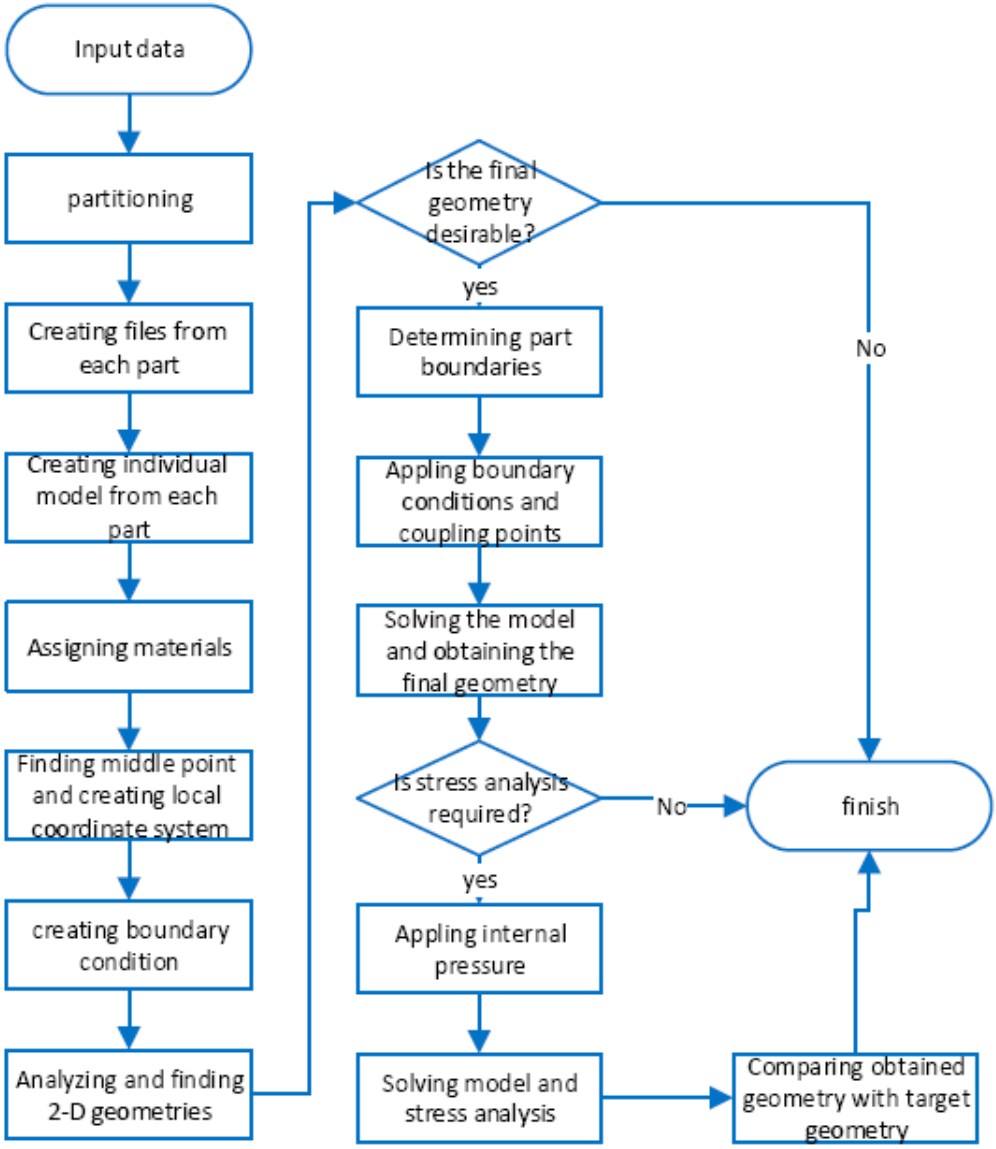



**Fig. 10.** ISA plugin operation flowchart.

Five initial automated processes are required to obtain the geometry of the flattened surfaces based on the user input: 1) creating a separate model for each partition, 2) assigning material properties, 3) defining solution steps, 4) determining boundary conditions, and 5) solving the model and saving it. During these processes, data related to boundary elements is stored for later re-coupling. The output of this analysis phase is the user-specified flattened surfaces.

### 2.3. Geometry modeling and stress analysis

After completing the analysis of all built models and obtaining the required two-dimensional geometries from the ISA plugin, the next step is to evaluate the three-dimensional structure formed from these surfaces. This involves simulating the final design target shell using the flattened surfaces and existing boundary points. First, a list of boundary points for each component and their neighboring points, along with their coordinates, must be prepared. This list is essential for defining boundary conditions during the re-modeling process, where the ISA plugin generates new parts from the previous output to create a unified model. Once material properties are assigned, boundary conditions are established by transferring the boundary points to their original coordinates and coupling adjacent points. After finalizing other stages, such as defining solution time and meshing, the model is submitted for analysis in Abaqus, ensuring that all parts are correctly positioned. Notably, the truss element defined at the boundaries is not utilized at this stage.

The third process in the ISA plugin involves applying internal pressure based on user input to shape the inflatable structure and simulate the stresses on the shell. This stage uses the final geometry from the previous stage as the initial model. After defining the model and assigning material properties, the user-defined pressure is applied to the inside of the shell. This process determines the stresses on the shell and its final geometry. Figure 10 illustrates the plugin's operation.

## 3. Plugin Introduction and How It Operates

The ISA plugin includes four main outputs as shown in Fig. 11. The first step is to identify the necessary flattened surfaces. The user defines the target geometry, cutting lines, and fabric mechanical properties for the inflatable structure. The plugin then generates the two-dimensional flattened surfaces based on this information. Utilizing the finite element method, which minimizes strain energy, the resulting surfaces are optimized for both energy and displacement.

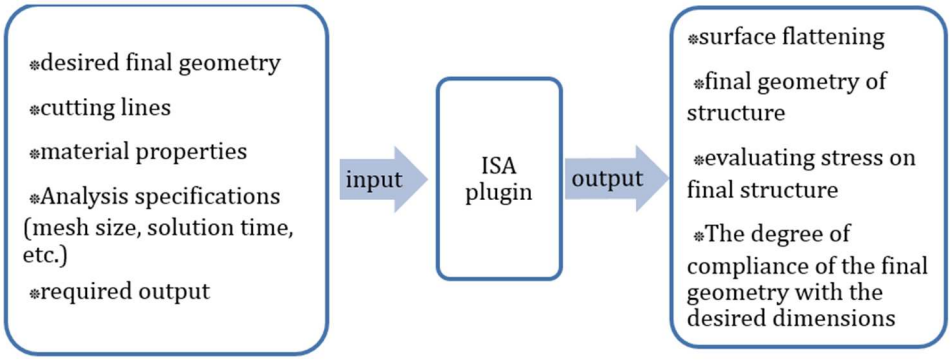



**Fig. 11.** Input and output data of plugin.

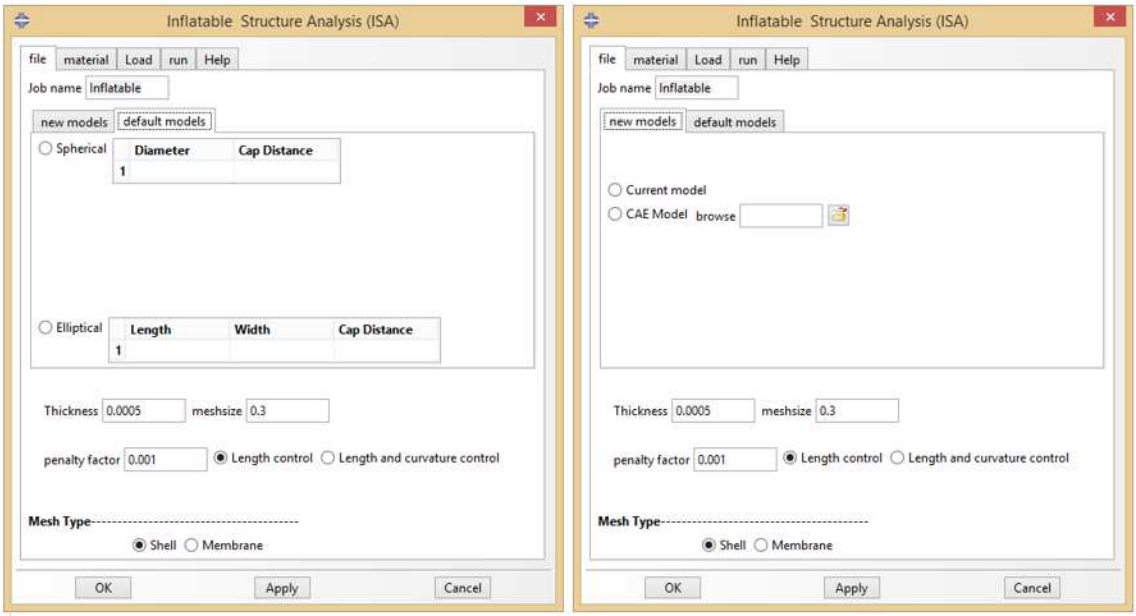



**Fig. 12.** Plugin file menu for entering geometric information.

In the second and third steps, the plugin defines the final shape of the structure and the stresses on various parts of the shell, using the flattened surfaces from the first step and applying boundary conditions to the edges and the user-specified internal pressure. Most studies simulating stresses from static or dynamic loads in inflatable structures, like airships, treat the hull as a complete shell, often overlooking the impact of seams. The stress analysis in the algorithm underlying the ISA plugin accounts for the cutting lines of different parts, enabling users to design suitable connection seams. The fourth part of the ISA plugin is to assessment of conformity the final shape obtained and the optimal geometry initially set as the objective.

### 3.1. Inputting geometric data

In the file menu, parameters for defining the model, mesh size, and shell thickness are provided. The model definition section includes two subsections: one for default models and another for new models (Fig. 12). The default plugin models are a sphere and an ellipse, each comprising eight segments and two end caps. This section specifies their geometric characteristics, including the sphere's diameter, the ellipse's length and width, and the distance between the end caps. In the section for new models, the user can use the model created in the current file or a separately defined CAE file. The penalty factor to control length and curvature boundary elements is defined in this section. The plugin menu allows user to specify the element type for analysis as either membrane or shell.

### 3.2. Defining the properties of utilized materials

In material menu as shown in Fig. 13, it is possible to define two different types of materials commonly used for fabric structures. In surface flattening analyses, material properties are generally neglected. If the boundaries are compatible or the material is defined as isotropic, the material characteristics do not affect the final surface. In this extension, material definition is primarily intended for evaluating the structural performance of inflatable structures after reconstruction with developed surfaces and subsequent strength analysis. However, in physical inflatable models, boundary lengths can be modified through heating. Using an industrial heat gun, the model boundaries are adjusted accordingly. Therefore, in certain cases, it is necessary to weaken boundary compatibility constraints in a controlled manner. In such situations, the material properties—particularly anisotropy—become important in surface flattening. In other words, because the flattening criterion is defined as $\varepsilon^T D\varepsilon$, it is possible to adjust strain in different directions. This capability enhances the effectiveness of the current surface flattening method. Nevertheless, the material strain should not exceed its allowable limit, which can be assessed in analyses using this method. The proposed penalty coefficients allow for precise control of the permissible strain at the boundaries, which can be increased or decreased as needed.

It should be noted that the influence of material type and its effectiveness depends on the specific application and material, and, in general, cannot be universally addressed. Since a detailed study of this issue requires extensive experimental investigation, it is beyond the scope of this paper.

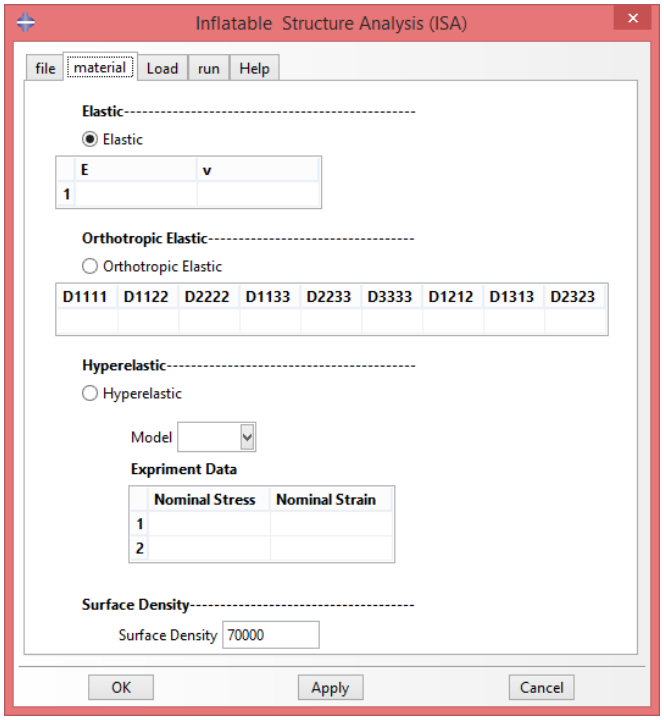



**Fig. 13.** Definition of utilized materials properties menu.

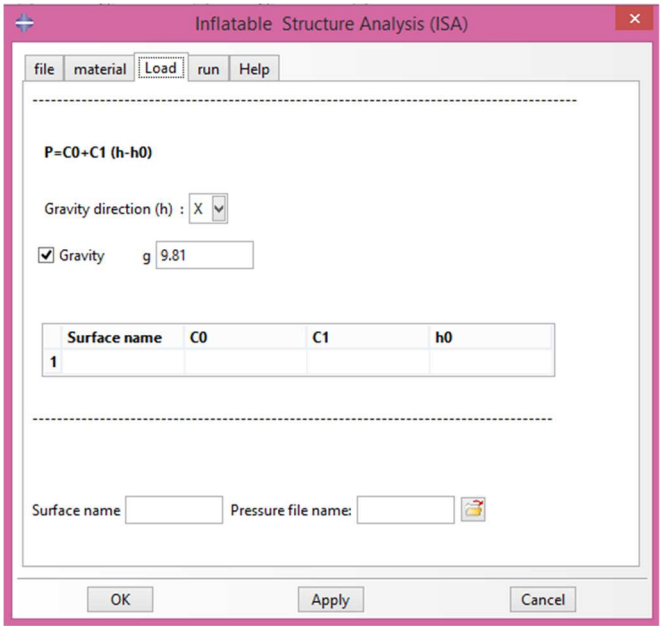



**Fig. 14.** Load definition menu.

In this section, the user can either select a type from orthotropic or hyperelastic materials and define the desired properties for this material. Data on material properties should be extracted from the test results and entered into the program. If the material is selected as orthotropic, 9 parameters of the orthotropic stiffness matrix are defined for the plugin compliant with Eq. (24):

$$\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{13}\\ \sigma_{23}\end{Bmatrix}=\begin{bmatrix}D_{1111} & D_{1122} & D_{1133} & 0 & 0 & 0\\ & D_{2222} & D_{2233} & 0 & 0 & 0\\ & & D_{3333} & 0 & 0 & 0\\ & & & D_{1212} & 0 & 0\\ & Sym & & & D_{1313} & 0\\ & & & & & D_{2323}\end{bmatrix}\begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{12}\\ \varepsilon_{13}\\ \varepsilon_{23}\end{Bmatrix} \tag{24}$$

Regarding hyperelastic materials, it is possible for the user to utilize two models, Marlow and Ogden. In order to define the properties of hyperelastic materials, the findings of the simple tensile test are specified in the form of a table called nominal stress and nominal strain. Abaqus uses the input data for the tensile test to calculate the potential energy relationship between the materials and includes it in the stress and strain calculations. The potential Marlow and Ogden energy relationship is consistent with functions (25) and (26), respectively [30, 31]:

$$W = \int_0^{\lambda_T - 1} T(\varepsilon) d\varepsilon \tag{25}$$

$$W = \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} \left( \lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3 \right) \tag{26}$$

In this section, the surface density of the utilized materials is also determined.

### 3.3. Loads

The third part of the plugin is focused on the definition of loads. (Fig. 14). This section is designed so that pressure loads for the model can be defined in two different ways. In the first loading method, the applied pressure on the defined surfaces is characterized as a linear function that actually represents the buoyancy force due to the difference between internal and external density. The relationship of the buoyancy force definition is per the following Eq. (27):

$$P = C_0 + C_1 (h - h_0) \tag{27}$$

Regarding the above, the $C_0$ is the initial pressure with pressure dimension, $C_1$ is the elevation coefficient with $\rho g$ dimension. Moreover, $h_0$ is the height of the inflatable structure's base. To utilize this method, the surfaces under this pressure must first be defined by the user in the model and then in this section, the name of the surface and the parameters related to the pressure must be defined for it. In addition, the direction and amount of gravity in the model must be specified in this section for the plugin. The plugin employs VDLOAD subroutine to load the model. The second method to apply pressure is to utilize a text file to define pressure on points on the surface. This type of load definition enables the user to utilize the output of CFD software as a pressure for the plugin.

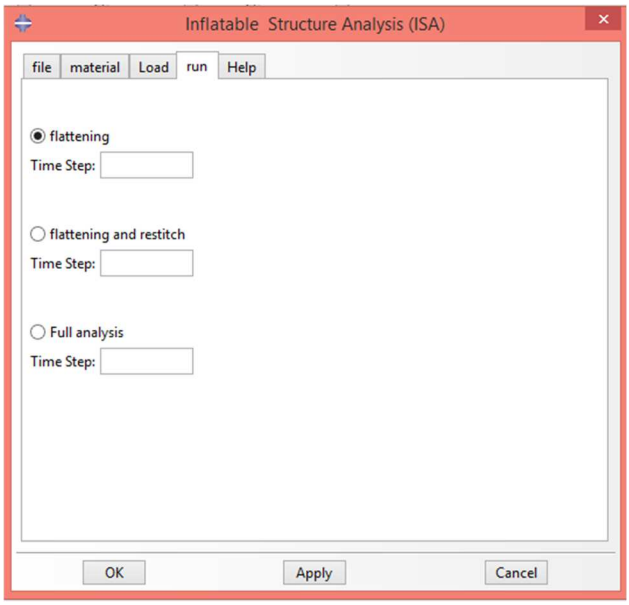



**Fig. 15.** Plugin run menu.

**Table 1.** Plugin input data for each section.

| Section | Sub-Section | Input Data |
|---|---|---|
| File | | Model Name |
| | | Shell Thickness |
| | | Mesh Size |
| | | Mesh Type (membrane/shell) |
| | | Penalty Factor |
| | | Boundary Control Type |
| | New Model | Current Model |
| | | CAE File |
| | Default Models (Sphere, ellipse) | Sphere (Diameter of Sphere, Distance of Caps) |
| | | ellipse (Elliptical Length, Elliptical Width, Distance of Caps) |
| Material | Elastic | Elastic coefficient and Poisson ratio |
| | Orthotropic Elastic | Stiffness Matrix Coefficients |
| | Hyperelastic Material | Marlow Model / Ogden Model |
| | | Surface Density |
| Load | Buoyant Force | Definition of Buoyancy Coefficients |
| | Utilization of CFD Analysis Output | Data file |
| Run | Flattening | Time step |
| | Flattening and restitch | Time step |
| | Full Analysis | Time step |

### 3.4. Run menu

In the fourth section, related to plugin implementation, the user determines the type of solution by selecting the options of flattening, flattening and restitch as well as full analysis. In addition, the time period of each analysis type is inputted in this section (Fig. 15). The finite element method employed in this study is the explicit dynamic approach. Clearly, by selecting a density higher than the actual value and a lower Young's modulus, the solution speed can be significantly increased. By default, a Young's modulus of 200 and a density of 700,000 are assigned for the structure with metric dimensions, which yields acceptable results. To further accelerate the analysis, multiple surfaces can be developed simultaneously. Additionally, the parallel processing capabilities of Abaqus can be utilized. Other solver settings, such as the time-step, are determined according to the software defaults. In summary, Table 1 introduces the various sections of plugin and input segments of each section.

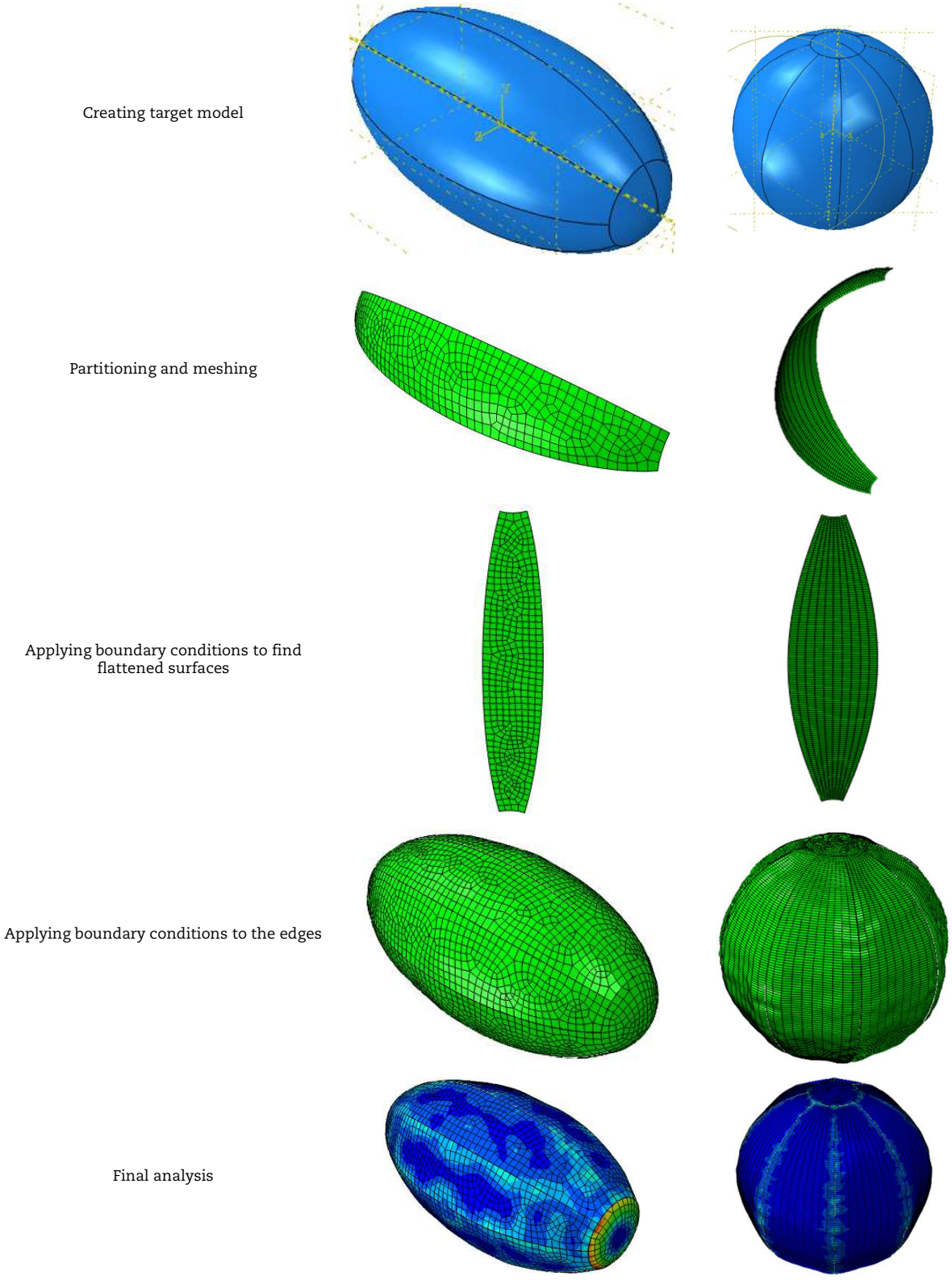



**Fig. 16.** Different steps of the algorithm to determine the surfaces making up the shells.

## 4. Plugin Analysis and Modeling

In this section, in order to ensure the plugin's performance and to validate the results it provides to the user, two experimental examples made of inflatable structures are examined. The two examples selected are based on the default models predicted in the plugin, namely sphere and ellipse. Figure 16 shows the different stages of plugin modeling based on the algorithm. The material selected for the fabrication of these two models are made of a hyperelastic material. However, because the strains applied in this model are insignificant, the behavior of the material is assumed to be linear.

## 5. Fabrication and Testing

Several CAD software options, including SolidWorks, are available for surface flattening. To assess the accuracy of the plugin's output, the surfaces created were compared to those produced by SolidWorks. Figure 17 shows the flattening surfaces from plugin and SolidWorks. The results for spheres of identical diameters are almost similar in plugin and SolidWorks, but the Abaqus solver with the ISA plugin offers enhanced capabilities for simulating the final shape and conducting stress analysis. A key advantage of the ISA plugin is its algorithm, which ensures that boundaries for different components using truss and beam elements are fully compatible. Figure 18 illustrates a comparison between two simulations, one with compatible boundaries and one without the compatibility condition. A 10-part sphere was created using the ISA plugin's two-dimensional model. As shown in Fig. 19, the projected two-dimensional plates closely align with the intended sphere's construction. Another aspect to consider in surface flattening models is the mesh size. Figure 20 shows that the mesh size only affects the accuracy of the stress field and does not affect the size of the resulting two-dimensional surfaces.

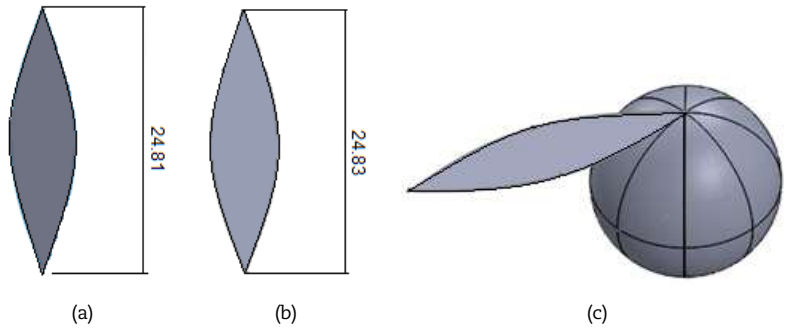



**Fig. 17.** Comparison of ISA plugin output and SolidWorks for surface flattening of a sphere, (a) ISA output, (b) SolidWorks output, (c) SolidWorks model.

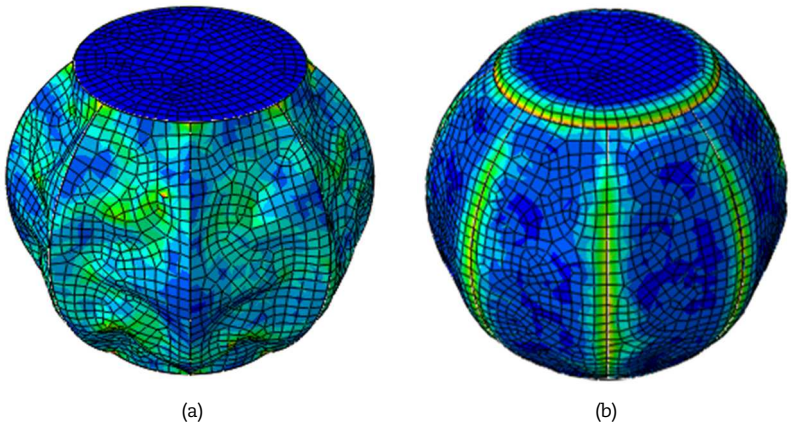



**Fig. 18.** Comparison of two simulations, (a) With compatible condition, (b) Without compatible condition.

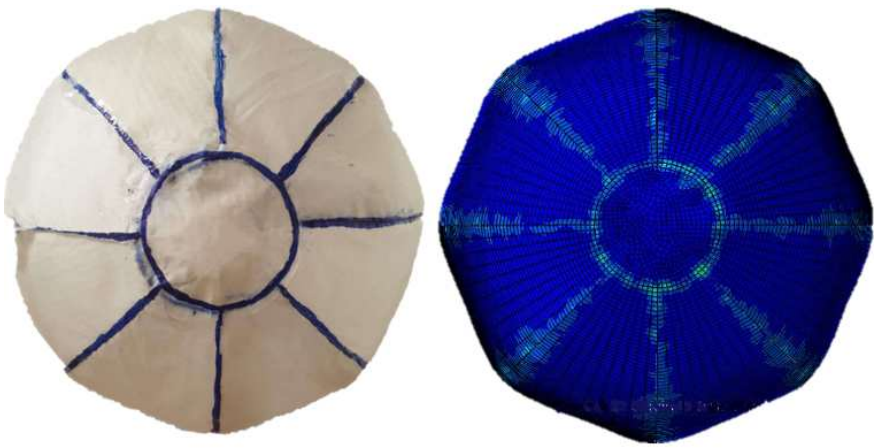


**Fig. 19.** Comparison of a spherical model (made by plugin) and a sphere (created by flattened surfaces).

## 6. Application

This section discusses three applications of the introduced plugin. The first example covers the design steps for an airbag to protect a falling object. The second and third examples focus on the design and analysis of an aerostat for carrying a load at height.

### 6.1. Airbag design

The airbag is designed to protect a falling object, with its dimensions depicted in Fig. 21. In the initial design, the flattened surfaces are calculated solely based on the desired external geometry. The ISA plugin was used to simulate and analyze the flattened surfaces and final structure. As shown in Fig. 22, the final shape did not meet expectations, exhibiting an unacceptable convexity that fails to adequately protect against falling objects. The modeled part was adjusted as shown in Fig. 23(b) to optimize the final shape, which closely resembles the optimal geometry.

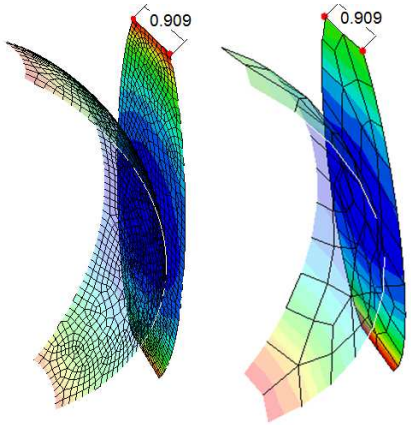



**Fig. 20.** Effect of mesh size of surface flattening.

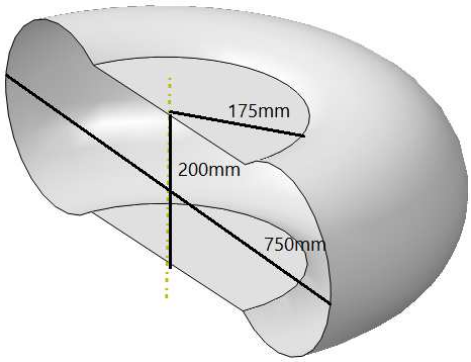



**Fig. 21.** Airbag dimensions.

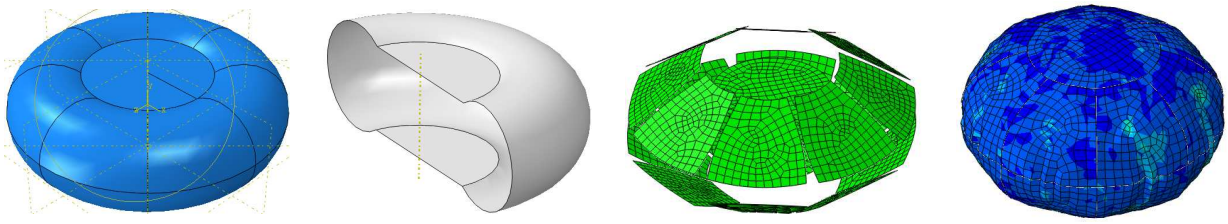



**Fig. 22.** Initial design of airbag by ISA, (a) Optimal geometry, (b) Modeled part, (c) Two-dimensional surfaces, (d) Final simulated geometry.

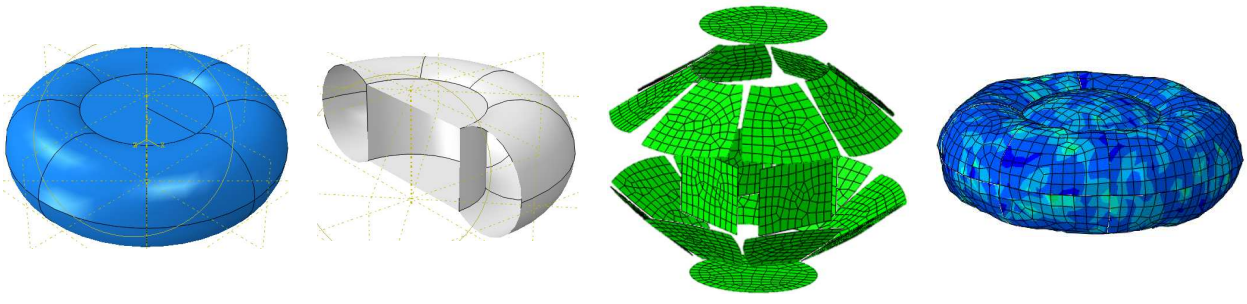



**Fig. 23.** Optimized design of airbag by ISA, (a) Optimal geometry, (b) Modeled part, (c) Two-dimensional surfaces, (d) Final simulated geometry.

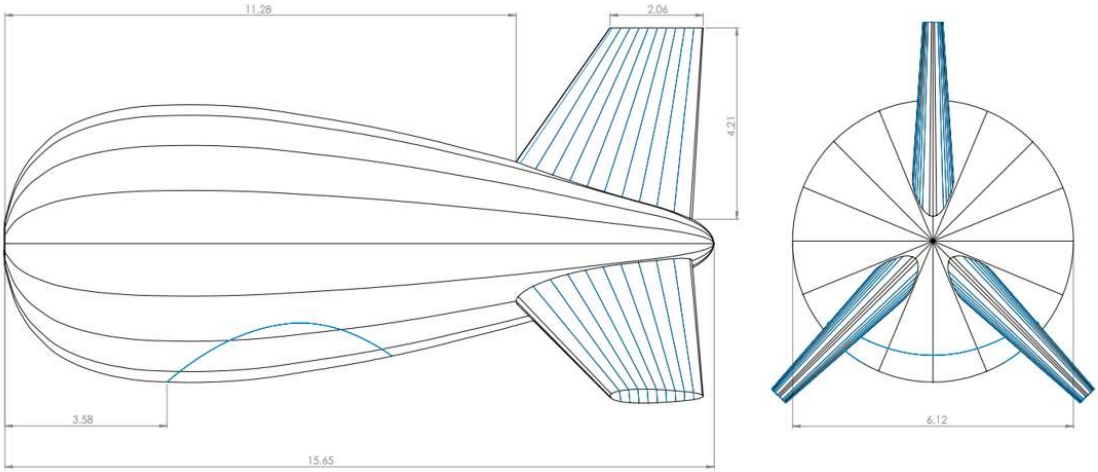



**Fig. 24.** The geometric shape and dimensions of the aerostat.

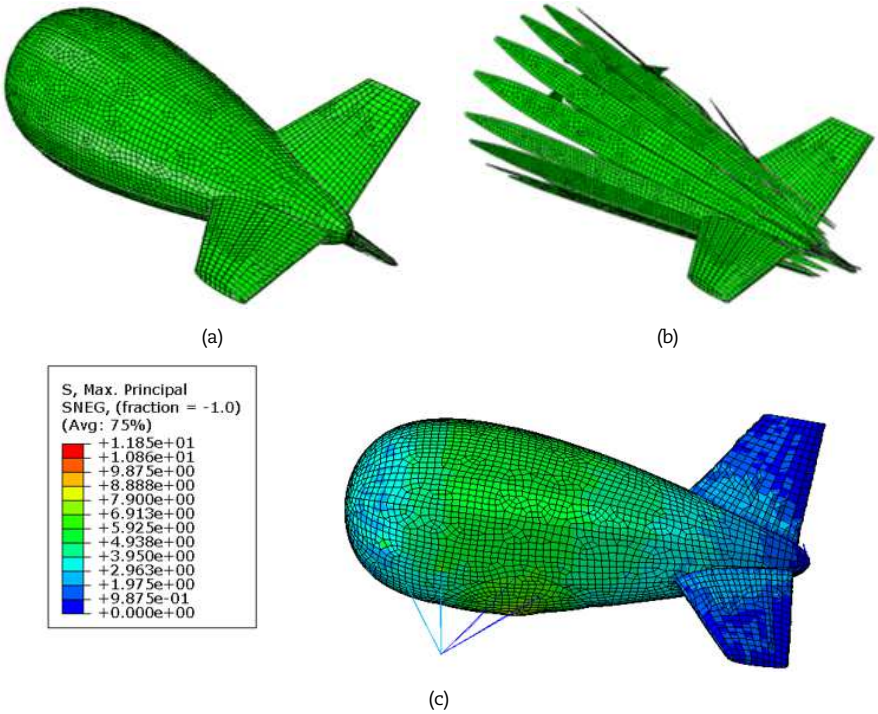



**Fig. 25.** Aerostat analysis results by ISA, (a) Initial model, (b) Flattened surfaces, (c) Stress on the balloon shell.

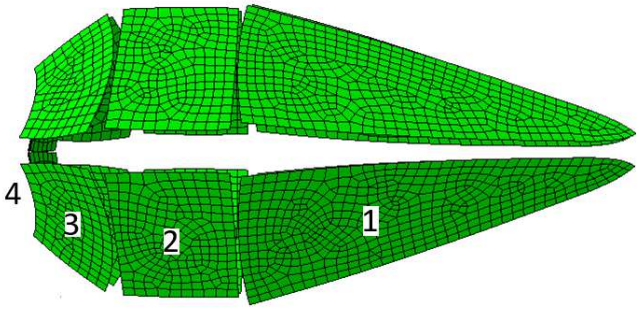



**Fig. 26.** A balloon surface flattening without compatible boundaries.

### 6.2. Aerostat design and simulation

Aerostat balloons consist of interconnected planer fabric parts, typically cut lengthwise along the balloon's axis. As illustrated in Fig. 24, the uniformity of the fabric sections ensures their borders are compatible. The ISA plugin was utilized to create an initial model for determining the flattened surfaces of the aerostat. This model informed subsequent analyses, revealing the applied stress on the balloon shell, as shown in Fig. 25. If the balloon parts differ, surface flattening requires controlling the boundary conditions to ensure compatibility. The resulting two-dimensional components cannot connect without strain. Figure 26 illustrates a special case of surface flattening for a balloon divided into nonuniform parts.

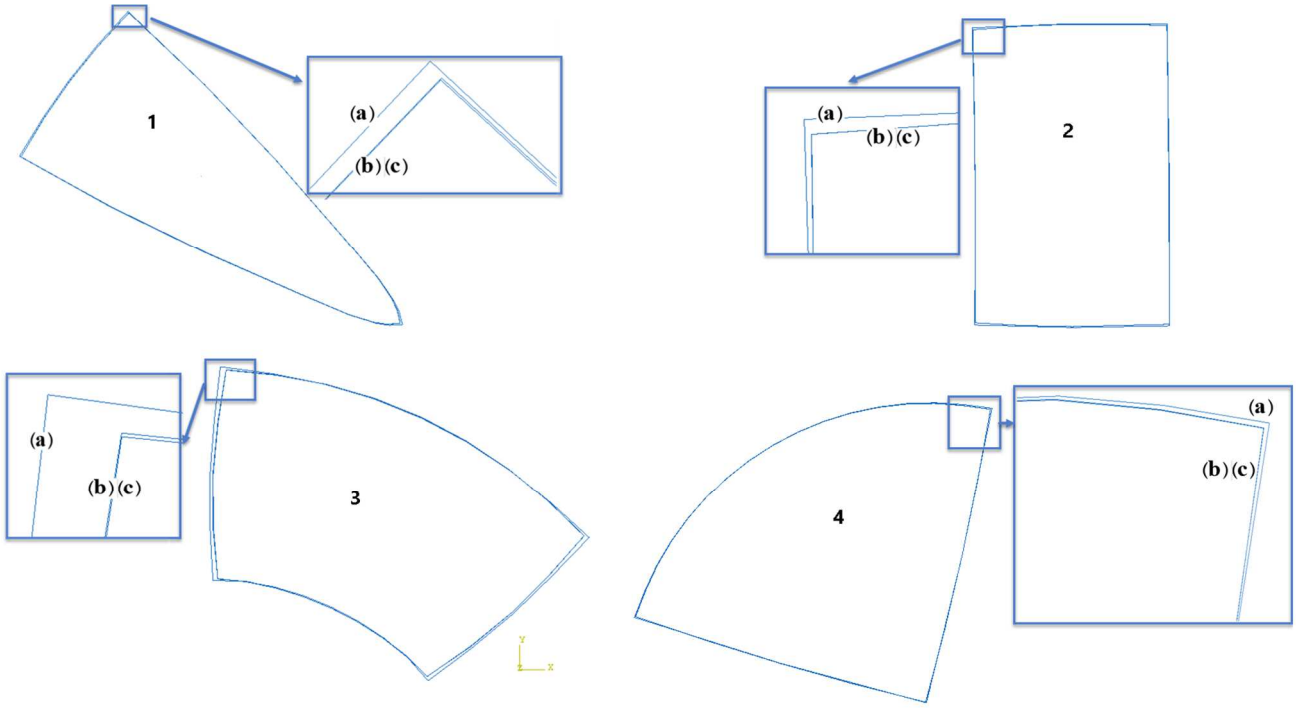



**Fig. 27.** Planer surfaces 1, 2, 3, and 4 assuming a free boundary (a), compatible boundary (b) and compatible boundary with curvature control (c).

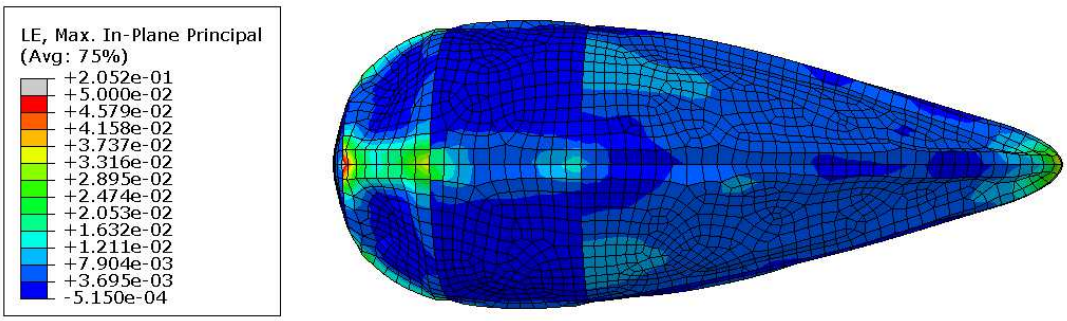



**Fig. 28.** Strain values at the boundaries after joining the planer surfaces without constraints at the boundary.

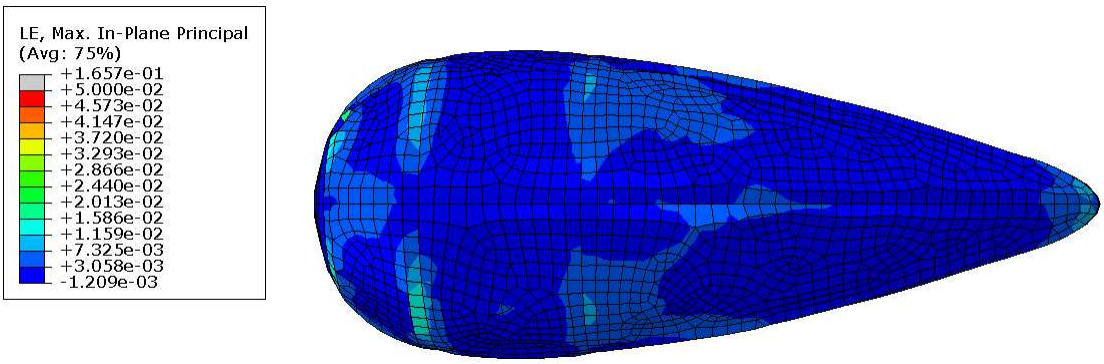



**Fig. 29.** Strain values in the boundaries after joining the planer surfaces with the condition of controlling length and curvature of the boundaries.

Applying the consistency boundary condition is crucial for identifying planer surfaces. Figure 27 illustrates the distinction between unconditional planer surfaces and those with consistent boundary conditions. Moreover, the difference between planer surfaces controlled for length at the boundaries and those controlled for both length and curvature is minimal.

If the boundaries are incompatible, reattaching the parts without straining them is impossible. Figure 28 illustrates the strain at the boundaries of expanding surfaces lacking boundary control. However, controlling the border length and using truss or beam elements significantly reduces strain. Figure 29 demonstrates a substantial reduction in boundary strain. In some cases, the length of the parts at the boundary could be increased by heating. Therefore, in some cases, complete compatibility at the boundary is not expected to be achieved. In such cases, the maximum strain rate is a fundamental criterion for determining the suitability of surface expansion methods. In the proposed method, the strain at the boundaries can be controlled by adjusting the values of $A$ in the truss element or $I$ in the beam element. Given that the amount of length change at the boundary is not the same, the strain field is a suitable criterion for demonstrating the degree of compatibility at the boundaries.

## 7. Conclusion

This paper presented a method for surface flattening based on finite element analysis. Using explicit finite element techniques and shell elements, surfaces were mapped to planer plates. To ensure compatibility between the boundaries of the flattened surfaces, truss elements used as penalty functions to control boundary lengths, while beam elements limited boundary curvature. These processes were implemented via a script in Abaqus software. To validate the method, results were compared with SolidWorks, yielding a discrepancy of less than 0.1% when a sphere was divided into petal-like sections. Additionally, the script in Abaqus facilitated the connection of flattened surfaces to create a three-dimensional object. Strain analysis from the reconnection process assessed the method's effectiveness in producing surfaces with compatible boundaries. The example of a balloon structure demonstrated significant improvements, with strains in certain areas reduced from 0.02 to below 0.007, and maximum strain values lowering from 0.2 to less than 0.05, showcasing the penalty method's efficacy in optimizing 2D surfaces. The described algorithms have been integrated into a new plugin for Abaqus, allowing for the calculation of flattened surfaces and the analysis of inflatable structures, similar to clothing analysis software.

## Author Contributions

A. Kamal planned the scheme and suggested the experiments; S.H. Dibajian conducted the experiments developed the mathematical modeling and examined the theory validation and analyzed the empirical results; The manuscript was written through the contribution of all authors. All authors discussed the results, reviewed, and approved the final version of the manuscript.

## Acknowledgments

Not applicable.

## Conflict of Interest

The authors declared no potential conflicts of interest concerning the research, authorship, and publication of this article.

## Funding

The authors received no financial support for the research, authorship, and publication of this article.

## Data Availability Statements

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

## References

[1] Tam, A., Joneja, A., Tang, K., Yao, Z., A Surface Development Method with Application in Footwear CAD/CAM, *Computer-Aided Design and Applications*, 4, 2007, 67–77.
[2] Ceruti, A., Marzocca, P., Airship and Hot Air Balloon Real Time Envelope Shape Prediction through a Cloth Simulation Technique, *SAE Technical Paper Series*, 2015.
[3] Callens, S.J.P., Zadpoor, A.A., From flat sheets to curved geometries: Origami and kirigami approaches, *Materials Today*, 21(3), 2018, 241–264.
[4] Malomo, L., Migliore, E., Panozzo, D., Cignoni, P., Pietroni, N., FlexMaps: Computational design of flat flexible shells for shaping 3D objects, *ACM Transactions on Graphics*, 37(6), 2019, 1–14.
[5] Azariadis, P.N., Nearchou, A.C., Aspragathos, N.A., An evolutionary algorithm for generating planar developments of arbitrarily curved surfaces, *Computers in Industry*, 47, 2002, 357–368.
[6] Zhang, S., Liu, C., Wang, X., An optimal flattening algorithm for ship hull plate fabricated by thermal forming, *Journal of Marine Science and Technology*, 24(1), 2019, 134–151.
[7] Guseinov, R., Miguel, E., Bickel, B., CurveUps: Shaping Objects from Flat Plates with Tension-Actuated Curvature, *ACM Transactions on Graphics*, 36(4), 2017, 1–12.
[8] Lévy, B., Petitjean, S., Ray, N., Maillot, J., Least squares conformal maps for automatic texture atlas generation, *Seminal Graphics Papers: Pushing the Boundaries*, 2, 2023, 193–202.
[9] Floater, M.S., Hormann, K., Surface parameterization: a tutorial and survey, *Advances in Multiresolution for Geometric Modelling*, 2005, 157–186.
[10] Choi, K.-J., Ko, H.-S., Research problems in clothing simulation, *Computer-Aided Design*, 37(6), 2005, 585–592.
[11] Ng, H.N., Grimsdale, R.L., Computer graphics techniques for modeling cloth, *IEEE Computer Graphics and Applications*, 16(5), 1996, 28–41.
[12] Chang, L., Zhao, Y., Yuan, H., Wei, Z., A Curved Plate-Flattening Method to Construct the Membrane Strain Distribution, *Journal of Marine Science and Engineering*, 11(2), 2023, 439.
[13] Zheng, P., Liu, Q., Lou, J., Lian, C., Lin, D., A free-form surface flattening algorithm that minimizes geometric deformation energy, *IET Image Processing*, 16(9), 2022, 2544–2556.
[14] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W., Multiresolution analysis of arbitrary meshes, *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 1995, 173–182.
[15] Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R., Sapiro, G., Halle, M., Conformal surface parameterization for texture mapping, *IEEE Transactions on Visualization and Computer Graphics*, 6(2), 2000, 181–189.
[16] Desbrun, M., Meyer, M., Alliez, P., Intrinsic parameterizations of surface meshes, *Computer Graphics Forum*, 21, 2002, 209–218.
[17] Floater, M.S., Parametrization and smooth approximation of surface triangulations, *Computer-Aided Geometric Design*, 14(3), 1997, 231–250.
[18] Shen, H., Wang, H., Hu, J., Yao, Z., An elastic approach for developing non-developable sheets, *International Journal of Mechanical Sciences*, 106, 2016, 138–146.
[19] Liu, X., Li, S., Zheng, X., Lin, M., Development of a flattening system for sheet metal with free-form surface, *Advances in Mechanical Engineering*, 8(2), 2016, 168781401663051.
[20] Topping, P.I.B.H.V., *Computer Aided Design of Cable Membrane Structures*, Saxe-Coburg Publications, 2000.
[21] Skouras, M., Thomaszewski, B., Kaufmann, P., Garg, A., Bickel, B., Grinspun, E., Gross, M., Designing inflatable structures, *ACM Transactions on Graphics*, 33(4), 2014, 1–10.
[22] Sosa, E.M., Wong, J.C.-S., Adumitroaie, A., Barbero, E.J., Thompson, G.J., Finite element simulation of deployment of large-scale confined inflatable structures, *Thin-Walled Structures*, 104, 2016, 152–167.
[23] Nesládek, M., Španiel, M., An Abaqus plugin for fatigue predictions, *Advances in Engineering Software*, 103(1), 2017, 1–11.

[24] Riaño, L., Joliff, Y., An Abaqus™ plug-in for the geometry generation of Representative Volume Elements with randomly distributed fibers and interphases, *Composite Structures*, 209(1), 2019, 644–651.
[25] Omairey, S.L., Dunning, P.D., Sriramula, S., Development of an ABAQUS plugin tool for periodic RVE homogenisation, *Engineering with Computers*, 35(2), 2019, 567–577.
[26] Sangsefidi, A.R., Dibajian, S.H., Kadkhodapour, J., Anaraki, A.P., Schmauder, S., Schneider, Y., An Abaqus plugin for evaluation of the Auxetic structure performance, *Engineering with Computers*, 3(5), 2021, 371.
[27] Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V.M., Hamdia, K.M., Zhuang, X., Rabczuk, T., An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, *Computer Methods in Applied Mechanics and Engineering*, 362, 2020, 112790.
[28] Nguyen-Thanh, V.M., Anitescu, C., Alajlan, N., Rabczuk, T., Zhuang, X., Parametric deep energy approach for elasticity accounting for strain gradient effects, *Computer Methods in Applied Mechanics and Engineering*, 386, 2021, 114096.
[29] Banabic, D., *Sheet Metal Forming Processes: Constitutive Modelling and Numerical Simulation*, Springer, 2010.
[30] Ogden, R.W., Large Deformation Isotropic Elasticity – On the Correlation of Theory and Experiment for Incompressible Rubberlike Solids, *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 326, 1972, 565–584.
[31] Marlow, R.S., *Constitutive Models for Rubber III: A General First-Invariant Hyperelastic Constitutive Model*, CRC Press, 2003.

## ORCID iD

Ahmad Kamal https://orcid.org/0009-0008-5636-6016
Sayed Hossein Dibajian https://orcid.org/0000-0002-6165-2374

© 2025 Shahid Chamran University of Ahvaz, Ahvaz, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0 license) (http://creativecommons.org/licenses/by-nc/4.0/).

**How to cite this article:** Kamal A., Dibajian S.H., Finite Element Algorithm for Surface Flattening, *J. Appl. Comput. Mech.*, 12(4), 2026, 1330-1347. https://doi.org/10.22055/jacm.2025.48560.5325

**Publisher's Note** Shahid Chamran University of Ahvaz remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.