Mesh Generation for Folded Airbags

A. Chawla, S. Mukherjee and Aneesh Sharma
Indian Institute of Technology, achawla@mech.iitd.ernet.in

ABSTRACT

This work presents the results of a novel approach to obtaining a geometric mesh of a given airbag after a series of folds are defined on it, for use in crash simulations. The process of airbag folding is simulated as a series of geometric transformations applied on to the airbag mesh, which is modeled as a stack of connected planar layers. Along with these transformations, optimization techniques are used to ensure that there is minimum change in the geometry and the area of the airbag. The results from this approach are compared with several other commercial airbag folding software, and it is observed that the algorithm proves to be very effective in ensuring minimum change in the area and geometry of the airbag during the folding process.

Keywords: Airbag model, cloth model, finite element simulation, folding

1. INTRODUCTION

Airbags are used in vehicles to enhance their safety and have been proven worldwide to be a vital safety device in vehicle crashes. Thus, while modeling vehicle crashes using FE simulations, we also need to look at airbag models carefully as the process of inflation of an airbag can prove to be the determining factor in saving lives. The duration from the initial impact of the crash to the full inflation of an airbag is about 40 milliseconds and during this time, the airbag goes from being in a folded state to a fully inflated state, with a high internal pressure. After achieving this state, the airbag begins to deflate, thus reducing the internal pressure and providing a nice cushion for the body impacting it. Ideally the person in the crash should come into contact with the airbag at this time, but this always does not hold and the contact may take place before the airbag is fully inflated or the body may hit the airbag at its periphery instead of the center of the airbag, which offers maximum protection. In this non-desirable contact position, the airbag unfolding process would determine the extent of safety provided to the passenger, as the state of the airbag at any given time would be different for each kind of folds defined on it. Thus, while studying crashes using FE simulations, we also need to model the behavior of the airbag during the crash. In a vehicle, the airbag is folded and kept in a small inflator, which diffuses gas generated through an explosive chemical reaction to the airbag during a crash. Thus, to study the behavior of the airbag using FE simulations, we need to have an FE model of the airbag in the folded position.

Some of the available software for vehicle crash modeling, such as PAM-CRASH, have modules for folding airbags. The approach taken by PAM-CRASH is to model the airbag as a geometric surface, which is transformed to achieve a folded state of the surface that can then be meshed. PAM-CRASH takes a very simple surface for the airbag surface and in order to keep the geometry simple, it sacrifices the exactness of the folding process. For generating realistic models of the airbag surface, we would have to model the airbag as a set of complex surfaces. Since the airbag is made of cloth, an airbag can be modeled as a cloth and airbag folding can be viewed as a special configuration of generic cloth.

Cloth modeling has been an active area of research for quite some time now as there are numerous applications for cloth models in computer graphics and animation industry. Cloth surfaces have been modeled using the spring-mass models by Zhang [1], self collision modeling, incorporated by Ng [2], and recently wrinkle modeling by Bridson [5]. Ng [3] offers an excellent survey of computer graphics techniques used to model cloth. The emphasis in recent years have been on physically based models of cloth, incorporating cloth dynamics and modeling the cloth as a very fine mesh, whose behavior is governed by self collisions and interaction of cloth with other materials. One such sophisticated technique is presented by Bridson [6], which has been successfully applied to computer animation. Thus, the emphasis in cloth modeling has been on obtaining a realistic looking model of the cloth rather than using it in Finite Element (FE) simulations, as the applications of the model are mostly found in the animation industry. Hence, the models are either too refined to be of use in FE simulations (the computation time increases exponentially with refinement in a FE mesh), or, are unable to handle very tight squeezing of the cloth, which is required in airbag folding. As an illustrative example, in the driver side airbag, used as a case study for this work, the airbag cloth is a 600mm diameter membrane and is packed in a in casing, which is approximately a cuboid of size 130x110x30mm. This results in a very tight squeezing of the airbag surface against itself. The FE model of the folded airbag is inflated using a FE simulation package, like PAM-CRASH and as the computational efficiency of the package is proportional to the size and shape of the FE mesh, the more refined the mesh, the higher the computation time. This work addresses the need for a model of an airbag, which can model the tight folds of the airbag, as well as still be computationally feasible in FE simulations.

The usual approach to airbag folding is a geometric one, consisting of equations of the airbag surface, which is meshed after the completion of the folds. This is a novel airbag folding approach, which folds a given initial 3D mesh of an airbag multiple times and refines the mesh as per the requirements of the fold. Compared with the PAM-SAFE package for airbag folding, it is observed that the change in area and geometry of the airbag is substantially reduced by this technique.

The folding process itself is modeled as a geometric transformation combined with an optimization approach to locate critical points in the fold; the objective of the optimization being to minimize the change in geometry and surface area of the airbag. After the final folds, a FE mesh of the airbag is obtained by assigning physical properties to the airbag material. This FE model can then be imported into any other mesh and inflated using the FE airbag simulator package PAM-CRASH.

2. PROBLEM DEFINITION

Our objective is to generate a well-behaved geometric mesh of a given airbag in a folded position, given the initial geometry of the airbag and the sequence of folds that are defined on it. A well-behaved mesh of an airbag implies a mesh that is amenable to computation using FE simulations. For example, a mesh with very small elements or elements with a very high (or very low) aspect ratio is not well behaved as the time for computation would go up considerably. The initial geometry of the airbag is constrained to be in the form of a set of parallel planar layers, which are connected using a set of elements arranged in a planar configuration (these planes being inclined to the parallel layers and are referred to as inclined layers in this text). This constraint is imposed as initial airbag shapes can easily be modeled using this configuration and the configuration also allows us to model airbag folds. For example, in our case study, we start with an initial geometry of a disc, with a schematic of the sequence of folds that have been followed for folding. The initial geometry is then meshed using shell elements, which are either quadrilateral or triangular. These two kinds of elements are sufficient to model any type of surface and hence we restrict our package to handle only these kinds of elements. This initial mesh is input to our package, and the user is asked to input the values for the parameters, which are used to define the fold. Different packages use their own parameters to define folds and we have also defined a set of parameters that we found to be most intuitive for defining a fold. These parameters take into account the positioning and tightness of the fold, as well as defining which part of the cloth is to be folded. After the fold has been defined, the package determines which planes (among the parallel layers) are to be folded, depending on the folding parameters. Similarly, the inclined layers to be folded are determined. The algorithm then proceeds to refine the mesh as per the requirement of the fold and transforms the elements to their new positions, which are governed by the parameter controlling the tightness of the fold. Thus, the configuration of the mesh after the fold is achieved and output in form of a mesh. This process of defining and executing folds can be continued till the requisite configuration is achieved. The output mesh may have some elements with a non-desirable aspect ratio, and for correcting these, there is a coarsening module in the package, which inputs the mesh along with the required refinement level and outputs the coarsened mesh.

3. EXISTING TOOLS

PAM-CRASH software has a module for folding airbags, called SAFE EDITOR, which inputs the initial geometry of the airbag and generates the geometry resulting after a sequence of folds. One can also define gas properties and other parameters required for airbag definition in SAFE EDITOR, but we will restrict the discussion only to the airbag meshing options in the software, as that is the part our software implements. This module is a geometry-based folder, which takes the initial airbag geometry as input and allows users to define folds on the airbag, giving the mesh of the final geometry as the output. SAFE EDITOR takes an IGES geometry file as an input for the initial geometry. For example, in our case study, the IGES file given as input to the airbag folder contains the geometry of a circle. As the SAFE EDITOR takes only a planar geometry as the initial input, it is constrained to having the lower layer of the airbag identical to the upper one, both of these layers being connected by a set of elements. After giving the initial geometry, one can proceed to define folds on the airbag. The software supports two kinds of folds: the normal roll fold and the tuck fold. We will look at the roll fold in detail in this section, as our tool supports roll folds only. The parameters to be input for defining a roll fold in SAFE EDITOR are location of the folding line, direction of the fold, a fixed point, thickness of the fold and the transient distance for the fold. Thus, one can define a folding line only on one surface of the airbag, which defined the folding location for the rest of the airbag also. The direction of the folding line can be reversed to change the direction of the fold from clockwise to anti-clockwise. The transient distance of an airbag fold corresponds to the width of airbag fabric on both sides of a folding line that will participate to the fold. The fixed point is a point that identifies the airbag patch that remains fixed during a folding sequence. And lastly, thickness of the fold defines the desired distance between innermost layers after the fold. The illustration of these parameters is given in Figure 1. The generation of mesh is a separate process from the folding in SAFE EDITOR, and is detached from the folding sequence. Thus the folding sequence can be changed quickly, without having to mesh the geometry at each stage, as we can mesh only the final state to obtain a "pc" file, which can then be input for FE simulation in PAM-CRASH.

There are some restrictions for defining folds in SAFE EDITOR. The folding lines have to be parallel or orthogonal to each other for defining a fold. Both the folding lines and transient lines have to be defined by points lying outside the airbag and are always applied to the full stack of layers. Also, a transient patch cannot be split by another parallel folding or transient line, even if only the transient line divides a given layer (since folds are applied to the full stack).

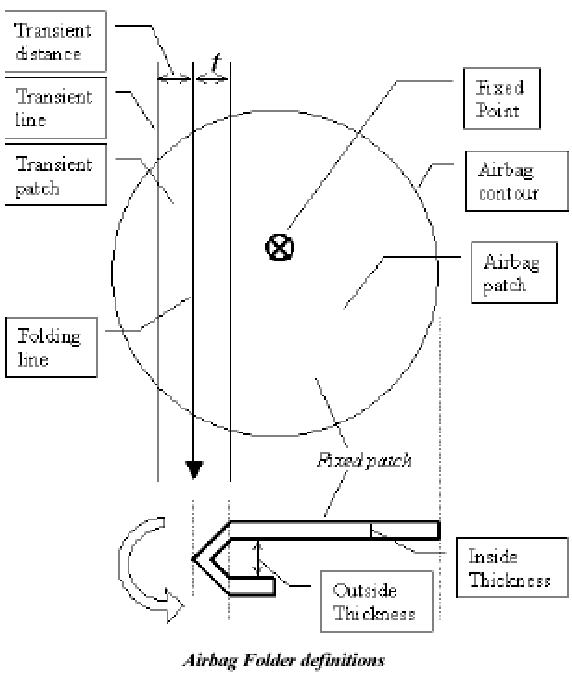

Fig. 1. Illustrations of the parameters for defining a roll fold as required by SAF EDITOR (given in Safe Tutorial [6])

4. DESCRIPTION OF PACKAGE

As stated earlier, this approach views the folds as a series of folds applied on the initial airbag mesh, coupled with an optimization approach to locate critical points in the mesh and generate the geometry of the folded airbag. This model of folding takes its inspiration from a real-life model of folding, wherein the cloth is folded physically by locating a folding line and then rotating a part of the cloth such that it lies on the top of the other. Physically when the cloth is folded, wrinkles are generated on the cloth surface. These are computationally expensive to model. But, if wrinkles are not modeled as a part of the folding process, then there is a loss in the area of the cloth surface, resulting in distortion of the geometry of the airbag. In our folding procedure we have incorporated the effect of wrinkles at the same time we have tried not to sacrifice on the computation time. We start with the initial airbag surface defined as a set of planar connected layers. Thus, prior to being able to define a fold on the airbag surface, it has to be collapsed and flattened into a set of connected planar layers. There are several methods to achieve this flattening, a few of them being: physical measurements of the actual airbag, and simulated flattening of the airbag using flat surfaces or using an approximation mesh to the actual shape. In our case study, we are using an approximation mesh of the shape of a disc of 600mm diameter, which is very close to the actual shape of the airbag. Thus, the flattening of the airbag results in several parallel layers (in our case study, the two planar connected layers), which have to be transformed simultaneously to define folds on the deflated airbag, while taking care to avoid mesh self-penetrations.

Each fold is defined by four input parameters:

1. The choice of a folding plane.
2. The desired width of the fold (which is the desired distance between the innermost layers after the fold).
3. The folding part (whether the mesh lying on the left or the right side of the folding plane is to be folded).
4. The folding direction (whether the mesh is to be folded clockwise or anti-clockwise).

Thus, at each step, we have an airbag mesh, on which the user (using the above parameters) defines a fold, and the algorithm generates an output mesh of the airbag in the folded position. In the process of folding, the mesh often needs to be refined. The algorithm takes this into account.

Thus, after each fold, we have a mesh of the airbag in the desired folded position. Usually, after the fold, the mesh has a higher level of refinement than the initial model. As a refined mesh becomes computationally expensive for FE simulations, we might need to coarsen the mesh at the end of the folding process, while keeping the geometry intact, to make the model amenable to simulation using PAM-CRASH. An algorithm for coarsening the mesh, is also incorporated in the software.

Each fold of the airbag generates one or more planar layers, which are added to the already existing set of parallel layers, and connected to the rest of the mesh by inclined layers. We can keep on folding the mesh as long as the layers are wide enough to support folding, though the time taken for the computation would increase with the number of planar layers. As the folding algorithm is modeled on the real world process of folding, the outer layers in a fold occupy a larger circumference compared to the inner layers in a fold. Thus, each layer that is folded, forks into two layers, both of which are smaller than the original layer, and two inclined layers. Thus, the area of each individual layer keeps on decreasing with each fold, though the sum of all the areas remains almost constant. And after a large number of folds, the layers may become too small to be able to accommodate a transient patch, signaling that no more folds can be carried on it. This process is intuitively similar to the real world folding process, where a cloth can be folded only a certain number of times, till the layers become too small to be folded again. Though, the number of folds that can be carried out are quite large and easily satisfy our requirement for airbag folding. The sequence of folds is also critical, as different sequences may produce distinct folds, thereby affecting the manner in which the airbag inflates.

Our algorithm is capable of handling folds in arbitrary directions, provided they do not intersect some special sections of the mesh called transient patches. Transient patches are created when the folding plane intersects the inclined layers and the creation of these patches is necessary to avoid change in the geometry of the airbag. This is not a serious functional limitation as transient patches created in airbag folding are very small compared to the airbag area. Thus, there is no restriction on the direction of folding planes, as long as they do not intersect the transient patches, giving a lot of flexibility in trying out various folding sequences.

5. CASE STUDY

In this section we illustrate the working of the folder by presenting a case study. This example illustrates the working as well as the efficacy of our algorithm. The generated mesh is compared with the results from one other commercially available software for airbag folding, PAM SAFE EDITOR. The example on which we perform the study is a typical passenger side airbag. The geometric details have been measured from a commercially available airbag. The initial state of the airbag is a closed disc of 640 mm diameter, with two planes of the disc separated by a distance of 0.4 mm, and connected by elements throughout the circumference. This is the initial state of the fabric of the airbag before the start of the folding process and is shown in Figure 1. Here we have taken the geometry of the two layers of the airbag to be identical, which is not a necessity for our algorithm, but is essential for SAFE EDITOR. In fact, SAFE EDITOR asks for a planar geometry as input to the folder and assumes that the second layer is exactly identical to this.

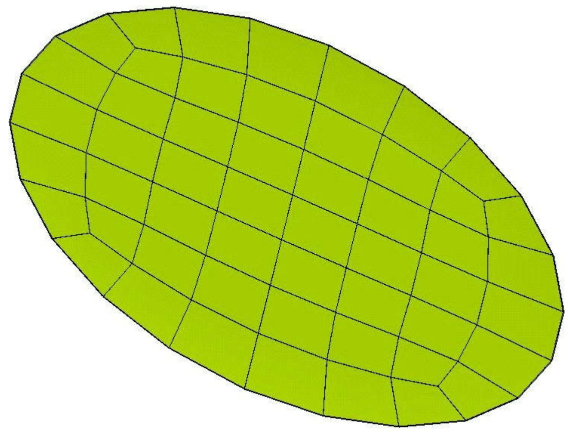

Fig. 2. The initial airbag geometry in the form of a disc

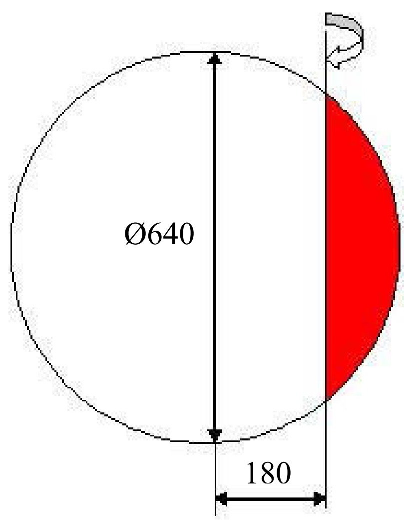

Fig. 3 (a) shows the schematic of the first fold

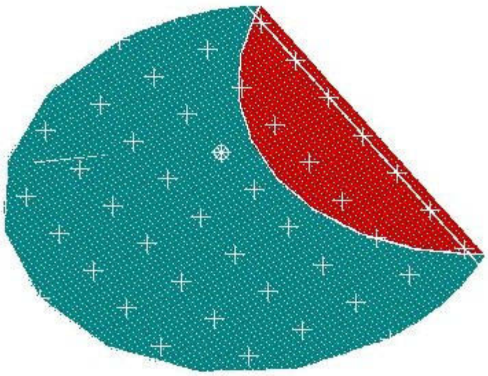

Fig. 3 (b) shows the folded geometry in SAFE EDITOR

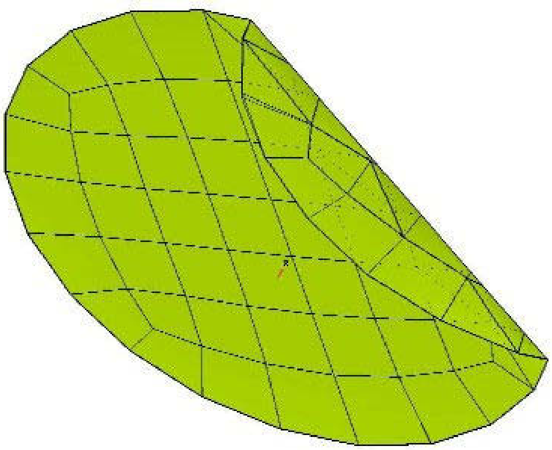

Fig. 3 (c) shows the folded mesh generated by our algorithm

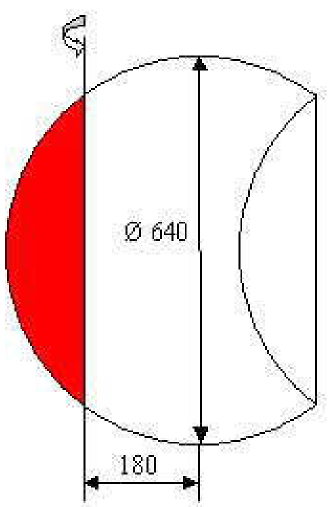

Fig. 4 (a) shows the schematic of the second fold

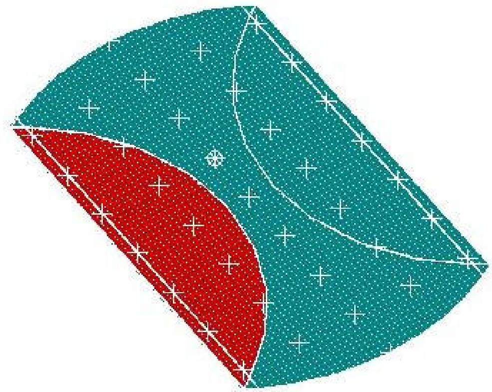

Fig. 4 (b) shows the folded geometry in SAFE EDITOR

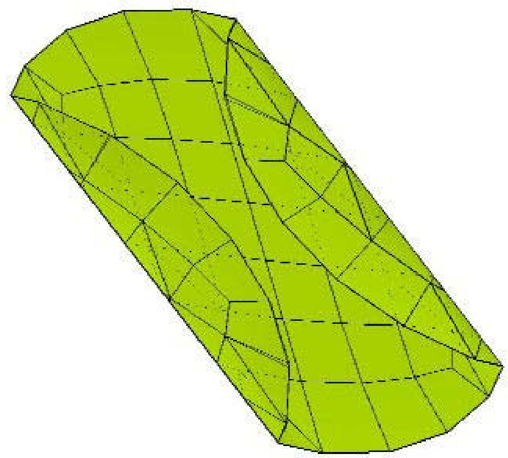

Fig. 4 (c) shows the folded mesh generated by our algorithm

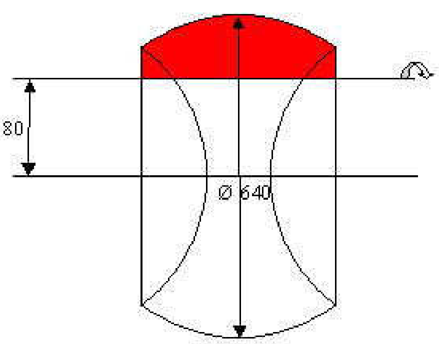

Fig. 5 (a) shows the schematic of the third fold

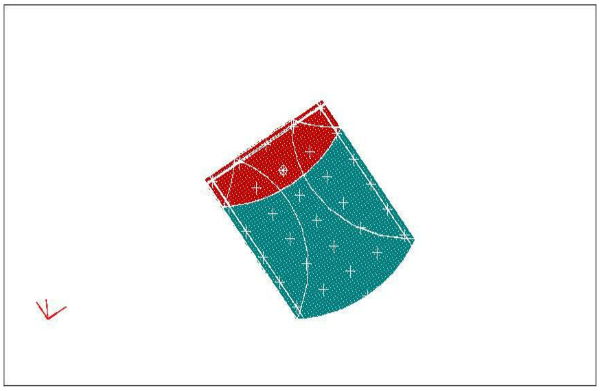

Fig. 5 (b) shows the folded geometry in SAFE EDITOR

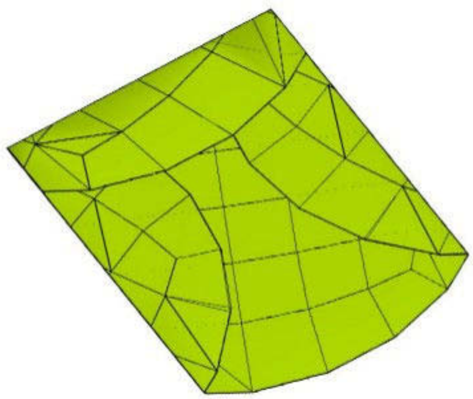

Fig. 5 (c) shows the folded mesh generated by our algorithm.

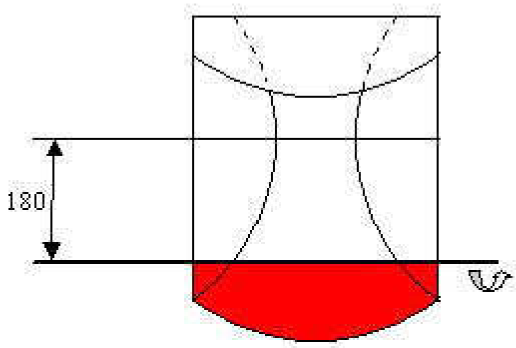

Fig. 6 (a) shows the schematic of the fourth fold

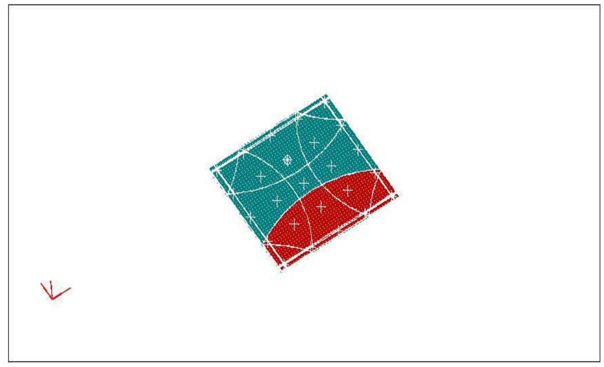

Fig. 6 (b) shows the folded geometry in SAFE EDITOR

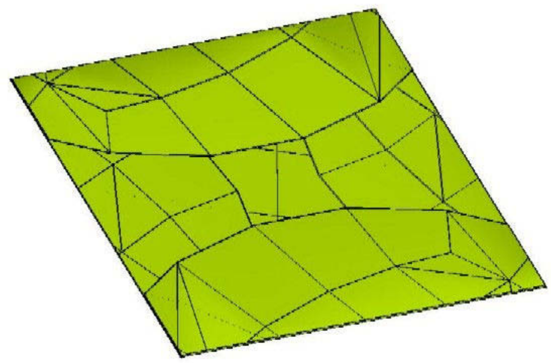

Fig. 6 (c). shows the folded mesh generated by our algorithm.

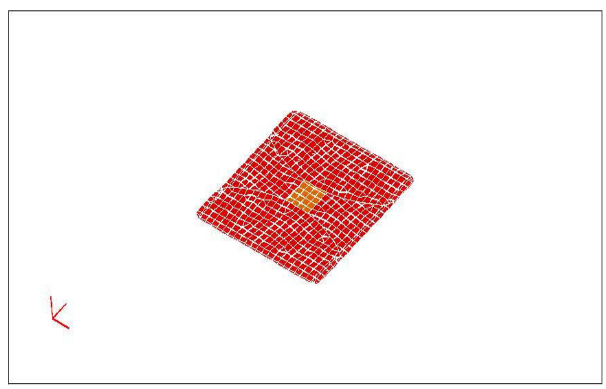

Fig. 7. The generated mesh on the airbag folded with SAFE EDITOR

Some of the images from a particular folding sequence are shown in Figure 1, where a single layer (circular mesh) was folded seven times.

6. RESULTS

The parameters used to measure the performance of the algorithm here are change in surface area of the bag during folding, change in volume of the inflated bag before and after folding, observed time steps and the visual shape of the bag. The algorithm proves to be very efficient in controlling the change in area of the airbag while folding as is demonstrated by fact that the percentage change in surface area with respect to the initial airbag after these four folds is 0.11% (0.637995 m^2 from the initial 0.63874 m^2). Whereas, the surface area in the airbag folded using SAFE EDITOR changes by almost 30% during these four folds to 0.441124 m^2. Volume of the unfolded airbag comes to be 48.8 litres after inflation, which is reduced to 48.1 litres after four folds using our algorithm. The same airbag outputs a volume of 39 litres after being folded using SAFE EDITOR. The reasons for these figures are fairly clear if we compare the shape of the inflated bags after being folded by the two packages. The bag folded using SAFE EDITOR undergoes a visible change in shape, while the bag folded by our package retains its similarity to the original shape.

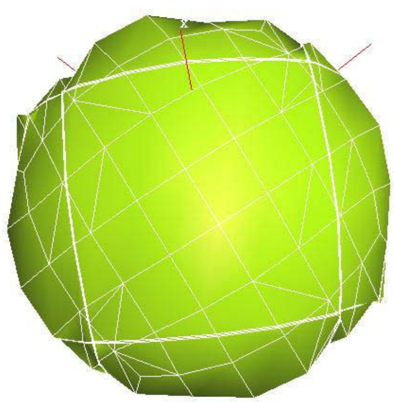

Fig. 8 (a). Top view of the inflated airbag, which was folded with our algorithm

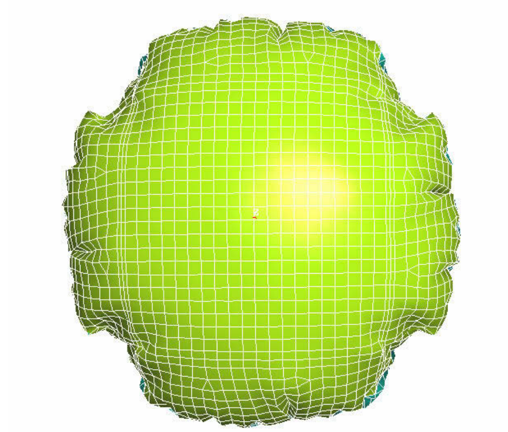

Fig. 8 (b). Top view of the inflated airbag, which was folded with SAFE EDITOR

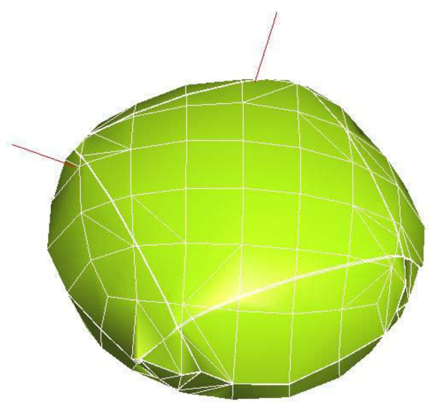

Fig. 9 (a). Isometric view of the inflated airbag, which was folded with our algorithm

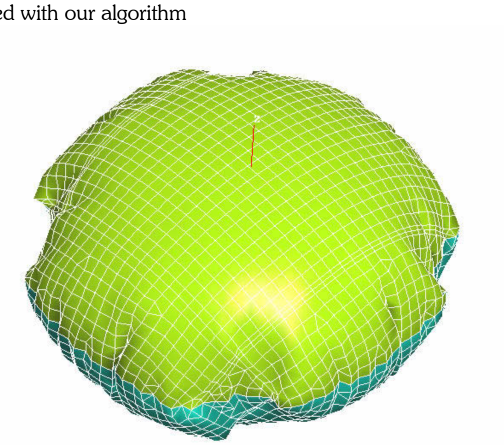

Fig. 9 (b). Isometric view of the inflated airbag, which was folded with SAFE EDITOR

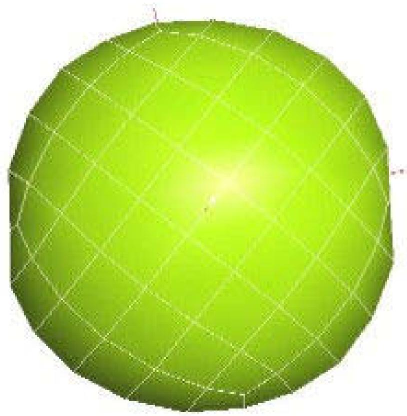

Fig. 10. The original shape of the airbag

7. LIMITATIONS

There are several limitations of this approach if it is viewed as a generic cloth folder, as it has been very highly optimized for one specific application of airbag modeling for crash simulations. Thus, we have not modeled the wrinkles in the cloth explicitly, resulting in a highly smooth appearance of the airbag surface, which does not appear realistic for a cloth surface. Also, as the folds have been modeled geometrically rather than physically, there are certain positions of the folding plane, which are not admissible by the algorithm. This is due to the fact that in certain positions the geometry becomes too complex to be modeled within our limitation of a coarse mesh, resulting in a sacrifice of certain positions of the folding plane for making our model amenable to FE simulations. Although, as airbag models usually have very clean folds, this does not affect the functionality of the algorithm as far as airbag modeling is concerned.

8. CONCLUSIONS

In this paper, we present an approach to airbag modeling for FE simulations, which preserves the geometry and surface area of the airbag during the folding process. We conduct a case study on a typical airbag and find that the said approach compares favorably with existing software and preserves the geometry of the airbag during the folding process. This software finds applications in airbag modeling for FE simulations, as it is optimized for the same.

9. REFERENCES

1. Bridson, R., Fedkiw, R. and Anderson, J., "Robust Treatment of Collisions, Contact and Friction for Cloth Animation", SIGGRAPH 2002, ACM TOG 21, 594-603 (2002).
2. Bridson, R., Marino, S., and Fedkiw, R., "Simulation of Clothing with Folds and Wrinkles", Eurographics/SIGGRAPH Symposium on Computer Animation 2003, ACM TOG 21, 594-603 (2002).
3. Dongliang Zhang, and Matthew M. F. Yuen, "Cloth simulation using multilevel meshes", Computers and Graphics, Volume 25, Issue 3, June 2001, Pages 383 – 389.
4. Liu, J., Ko, M., and Chang, R., "A simple self collision avoidance for cloth Animation", Computers and Graphics, Volume 22, No. 1, pp 117 – 128, 1998.
5. Ng, N., and Grimsdale, R.L., "Computer graphics techniques for modeling cloth", Computers and Graphics, Volume 19, No. 3, pp 423 – 430, 1995.
6. Ng, N., Grimsdale, R.L., Allen, W., "A system for modeling and visualization of cloth material", Computers and Graphics, Volume 25, Issue 3, June 2001, Pages 383 – 389.