Simulation-Based Airbag Folding System JFOLD Version 2 -New Capabilities and Folding Examples

Shinya Hayashi[1], Richard Taylor[2]

[1]JSOL Corporation

[2]Arup

2-18-25 Marunouchi, Naka-ku, Nagoya, Aichi, 460-0002, Japan

1 Abstract

Computer simulation is playing an increasingly important role in the design, development and application of airbag safety systems. As folding patterns and airbag structures become more and more complex, users are turning to simulation based folding solutions to generate accurately folded models in a short space of time. To meet this demand, a new software tool called JFOLD has been developed by JSOL Corporation to enable successful airbag folding using LS-DYNA®. JFOLD's intuitive and interactive system guides the user through the folding steps using flow-chart graphics, interactive tool positioning/resizing, tool motion control, animation preview and more. This paper introduces the new capabilities of JFOLD Version 2 and demonstrates various folding examples. JFOLD runs inside the powerful and popular pre-processor Primer.

2 Introduction

The airbag is a vital component in the vehicle safety system to reduce occupant injuries during a vehicle collision. Airbags consist of a fabric bag, tightly folded to fit inside the steering wheel, instrument panel, front seats, roof side rail and so on. The way the airbag is folded can greatly affect its deployment speed and shape, and this can have a large influence on occupant protection performance. Automobile and airbag manufacturers increasingly use computer simulations of airbag deployment to design and enhance more complex folding patterns. However it is not easy to create the computer model of the folded airbag due to the lack of CAD data for the complex folded geometry. A simulation-based folding approach needs preparation and calculation time but can be applied to almost any kind of folding pattern thanks to the physical, realistic nature of the folding process.

A new software tool called JFOLD [1] has been developed by JSOL Corporation to enable successful airbag folding using LS-DYNA. JFOLD Version 1 was released in July 2013 and JSOL Corporation is developing Version 2 to be more user-friendly with new applications to enable various folding patterns. This paper introduces the new capabilities of JFOLD Version 2 and demonstrates various folding examples. JFOLD runs inside the powerful pre-processor Oasys Primer and also uses the post-processor Oasys D3plot.

3 Simulation-based airbag folding & JFOLD general overview

As a simulation-based airbag folding system, JFOLD manages the set-up of various jigs (called tools) that move and deform the airbag mesh in an LS-DYNA analysis so as to achieve a folded shape. The location, shape and motion of each tool must be carefully set up then adjusted over a few analysis iterations to achieve the desired geometry for each fold. The final folded shape is usually obtained over a series of folding analysis steps: typically 5 to 10 for a simple airbag, many more for complex fold patterns. After each successful fold, the geometry of the deformed airbag and some of the tools must be carried over to the next step. To carry out these tasks as quickly as possible, the simulation-based airbag folding system must both manage the series of folding steps and also help the user to easily set up complicated tool motions.

JFOLD has an intuitive and interactive GUI system which guides users through the folding steps using flow-chart graphics and useful functions such as interactive tool positioning/resizing, tool motion control, animation preview and so on. Often the same shape tools are used repeatedly in folding steps. In JFOLD, a tool can be read from tool library as well as from another existing step. Tools in JFOLD have three categories: *tool mesh*, *fabric* and *component*. “Tool mesh” types are typically plates but can be any shape, made from beams, shells or solids. Tool meshes can represent the steel plates and jigs used in the real airbag folding process. “Fabric tool” type describes input data that for example applies pressure or nodal constraints to the airbag fabric. “Component” types represent real parts like the inflator, airbag module case, seat and body structure that can be used to deform the final folded shape.

4 Process management capability of JFOLD

Figure 1 shows the Process Management Panel of JFOLD, the hierarchy flow-chart of a chain of folding simulation steps. The root step contains the definition of the initial airbag mesh model and each folding step after that represents one folding analysis. Figure 1 also shows an example of how the data is carried over between analyses. The final result of the folding analysed in the parent step is used as the input model in the child step.

One of the most important capabilities of the flow-chart system is that it can be forked into several branches. JFOLD can be therefore be used as a design tool to quickly create airbags with various folding patterns. Figure 1 shows a chain of folding simulations that generate three folding patterns A, B and C.

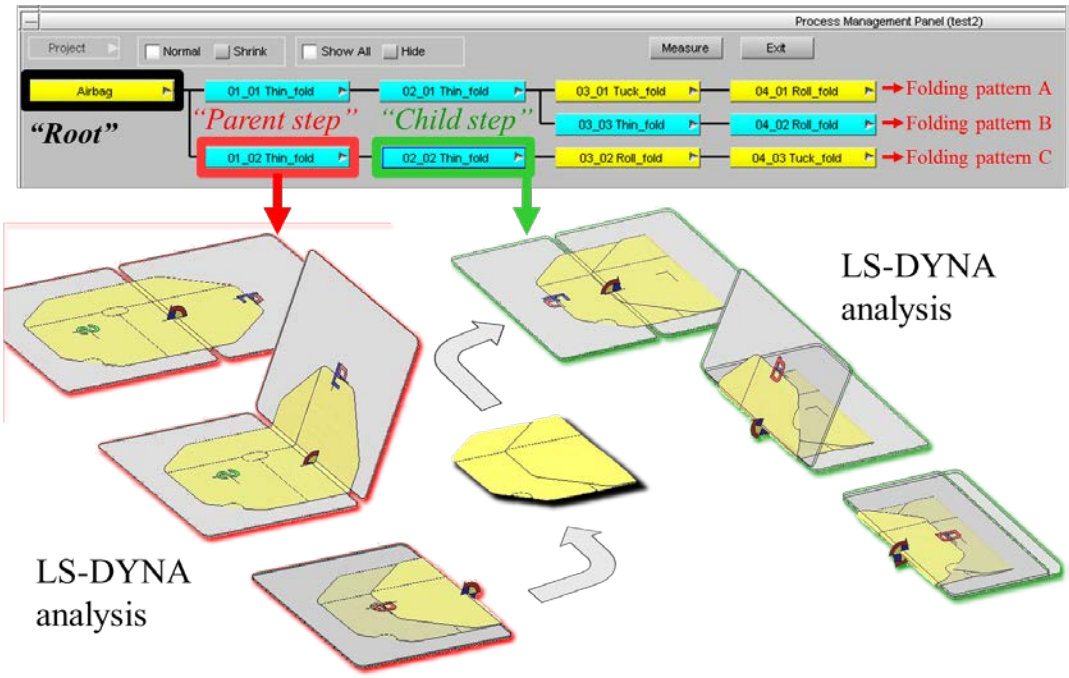

Figure 1: Flow-chart graphics of JFOLD and an example of two folding simulations.

5 Introduction to some of new capabilities of JFOLD Version 2

In JFOLD Version 2 many new capabilities and improvements have been added, some of which are introduced in this next section.

5.1 Tool Assembly assembling multiple tools

Figure 2 shows a typical folding mechanism that can generate a thin-fold, one of the most basic fold types. In JFOLD a thin-fold model is usually made up of four tools: two moving tools shown on the left fold the airbag and two stationary tools on the right clamp the airbag. Even for such a simple mechanism four tools are needed, and more complex folding mechanisms require many more tools. Therefore a new capability to manage folding mechanisms made up of multiple tools has been developed.

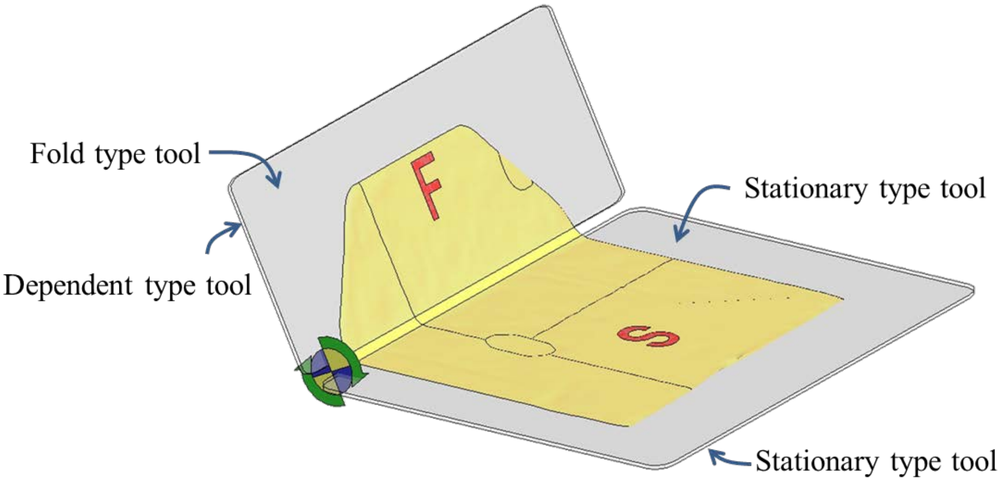

Figure 2: Four tools make up the thin-folding model.

JFOLD Version 2 provides a new capability called “Tool Assy” which can manage a set of tools as one folding mechanism. The above thin-folding mechanism can be defined as one Tool Assy. Users can create or modify an existing Tool Assy interactively using only a few mouse clicks. Tool Assys can be saved into and read from a tool library. The Tool Assy capability also brings GUI enhancements. All tools used in a folding step are listed as tool buttons on Tool Management Panel of JFOLD. When the list of tools becomes long (e.g. ten and over) the user has to scroll down to see all the buttons and it can take some effort to keep track of the whole folding process. In a Tool Assy multiple tool buttons are grouped into one. As a result users can see the whole folding sequence at a glance.

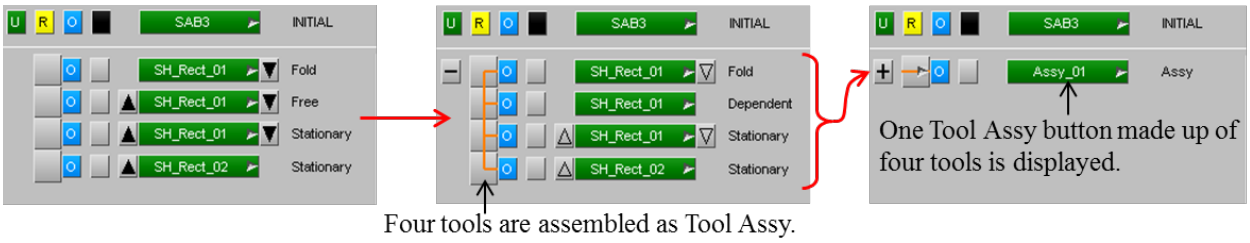

Figure 3: Tool Assy capability.

5.2 Applying load to airbag nodes

A new fabric tool called “LoadNodes” is implemented in Version 2. This new tool can directly apply forces on nodes of an airbag in any or all directions. JFOLD already has similar fabric tool “Pressure” to apply internal pressure inside the airbag, but its loading direction is limited only to normal direction of airbag shells.

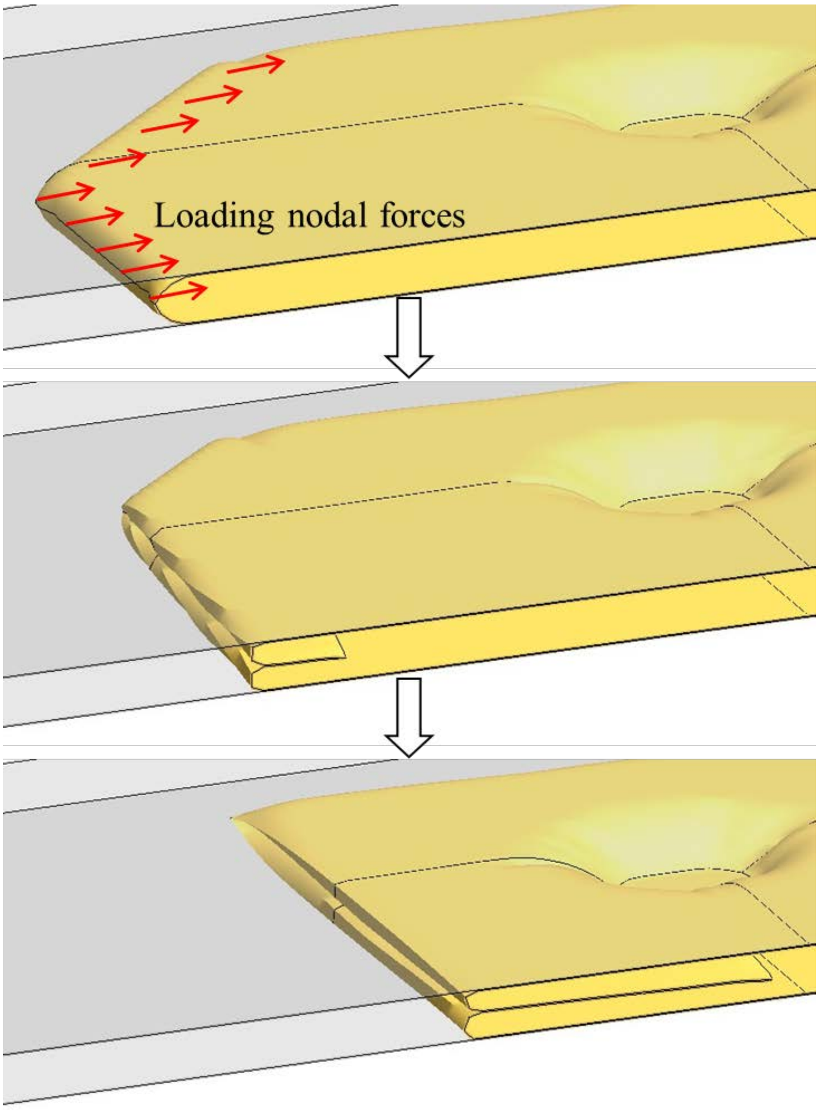

Figure 4: Tuck folding performed by nodal forces

5.3 Tying nodes of airbag to deformable tool

JFOLD Version 1 has the capability to move nodes on the airbag according to the motion of a rigid tool. In this case the airbag nodes are “tied” to the tool using *CONSTRAINED EXTRA NODE and the “Move type” rigid tool motion is controlled by *BOUNDARY PRESCRIBED MOTION RIGID (BPMR).

In JFOLD Version 2 this functionality has been extended to non-rigid tools, which means now airbag nodes can also be tied to the “finalGeom type” tool. The finalGeom type tool uses *BOUNDARY PRESCRIBED FINAL GEOMETRY (BPFG) and this is sometimes more useful than BPMR because only the final position must be set up, rather than translations and rotations for all degrees of freedom. The tied connectivity is realized with high stiffness discrete beams. The beams are automatically created between the airbag and deformable tool by just selecting a pair of nodes. Figure 5 shows an airbag tied to a finalGeom type tool, and deformed according to the motion of the tool.

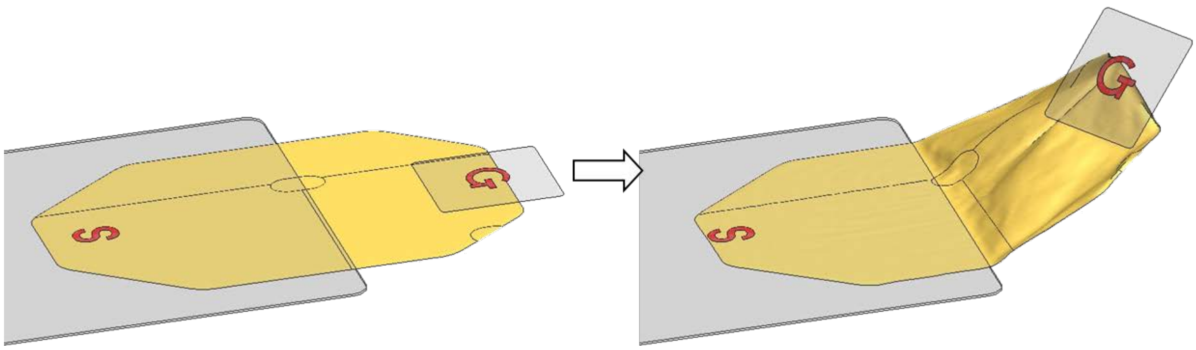

Figure 5: Example of tying airbag nodes to a deformable tool

5.4 Tool library management system

In JFOLD Version 1 only one tool library could be used per folding project. In JFOLD Version 2, multiple (max. 10) tool libraries can be used. Also new functions for managing tools (rename, copy and delete tools) have been added to make this a more user-friendly system.

5.5 Airbag management system

Users can now change material properties of airbag parts in each folding step. JFOLD Version 1 has a simple function to change airbag material properties, but its usability is rather limited. The new Airbag Management System implemented in Version 2 can change airbag material properties in a more user-friendly way. Also a new function to add other airbag parts (like an inner bag) midway through a folding project is provided for more complicated airbags.

In Version 2 the basic functions of the Airbag Management System are introduced. In the next version more advanced functions and capabilities are planned for development.

6 Avoidance of intersection problem during airbag folding

With JFOLD’s default contact settings, example models for reference and tutorial for guidance, users should be able to simulate realistic airbag folding without causing many intersections (crossed-edge penetrations) in the airbag fabric. However in extreme and complex fold cases these can still occur and cause serious problems not only during folding analyses but also final deployment.

In LS-DYNA R6 the contact option Q2TRI=4 can sometimes help reduce intersections, although it adds to computation time and must then be used forever with that airbag. In LS-DYNA R7 another new contact option “depth=45” has been implemented [2] with airbag folding in mind. This option activates a node-to-bilinear patch contact calculation to better cope with the warped quad surfaces.

7 Examples of airbag folding

In this section, various airbag folding examples are demonstrated to introduce the performance of JFOLD.

7.1 Driver airbag folding

Figure 6 shows an example of a typical Driver's airbag folding process. The final folded shape is obtained in just four folding analysis steps. In the final "Z-fold" step, JFOLD performs several actions in just one step.

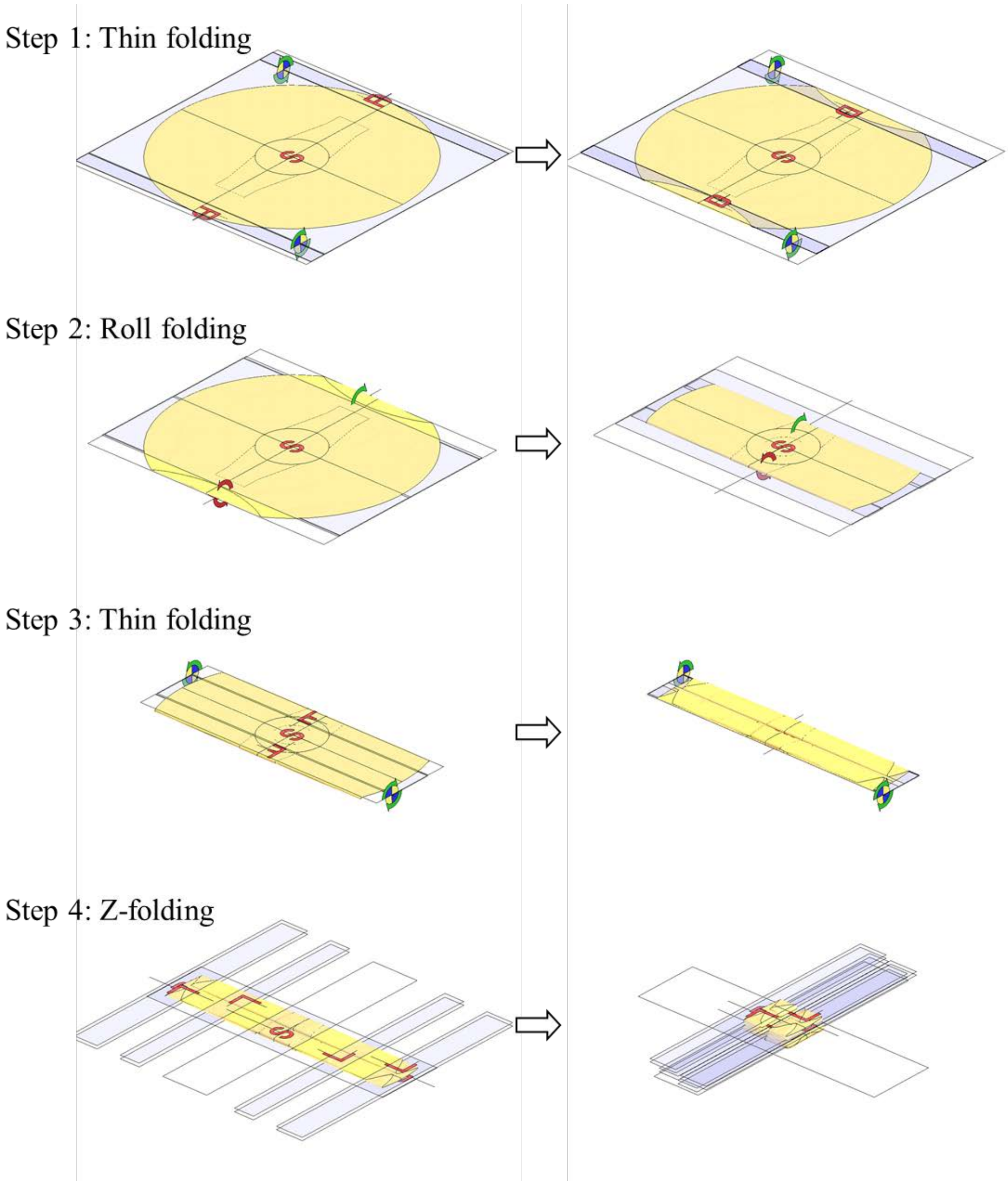

Figure 6: Driver airbag folding.

7.2 Z-folding performed by load type tools

Z-folding is a commonly used folding technique but is not easy to simulate because of the complicated motions required by multiple tools. The exact motion of each tool is hard to predict unless each fold is treated individually, which would then take a lot of time. A new folding method to create Z-folds more efficiently has been developed to cope with this. The new Z-fold method uses “Load type” tools and is presented in Figure 7. Load type tools are moved by *LOAD RIGID BODY. The positions and motions of the tools are driven naturally in the folding simulation according to the balance of applied force on the tool and tension in the airbag. Appropriate forces must be chosen to generate the desired response in the analysis time without causing strange deformation, but the tool library supplied with JFOLD Version 2 will contain a working Z-fold example as a Tool Assy.

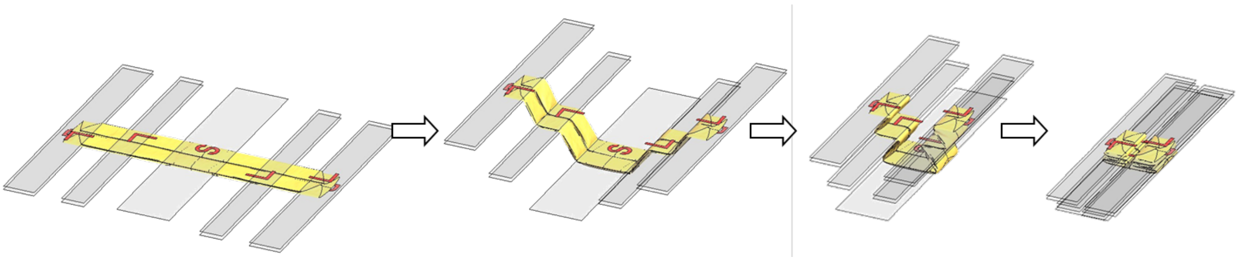

Figure 7: Z-folding performed by load type tools.

7.3 3D shaped airbag folding to flat shape

Figure 8 shows an example of flattening fold step from a 3D passenger airbag model folding process. This type of folding is very difficult because users have to accurately locate the fold lines and set up complicated tool motions to generate the flat condition without wrinkles.

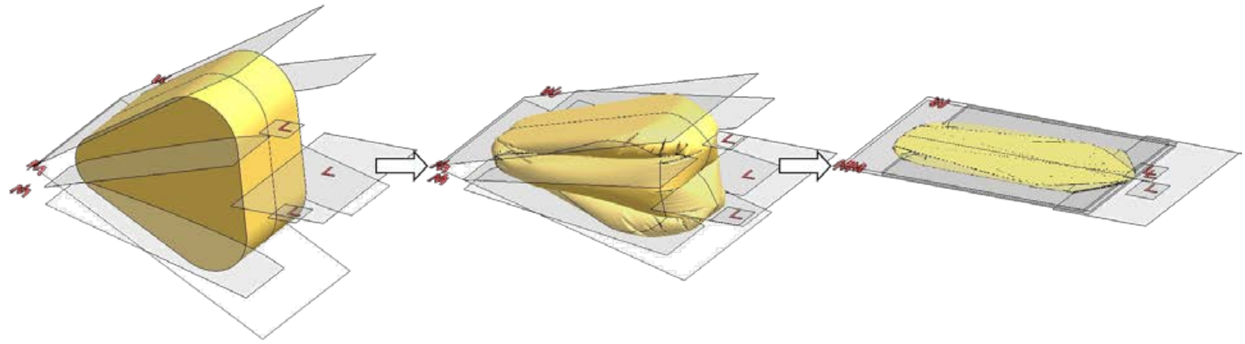

Figure 8: Passenger airbag folding simulation

In the Figure 8 example some Load type tools are used. As seen in the previous Z-fold method, Load type tools can be very effective in performing efficient airbag folding.

Figure 9 shows an example of folding a vertical sheet using a Load type tool. The initial position is not so important – the Load type tool moves to the natural fold line automatically while applying adequate tension to the sheet. The result is a well folded sheet without any wrinkles.

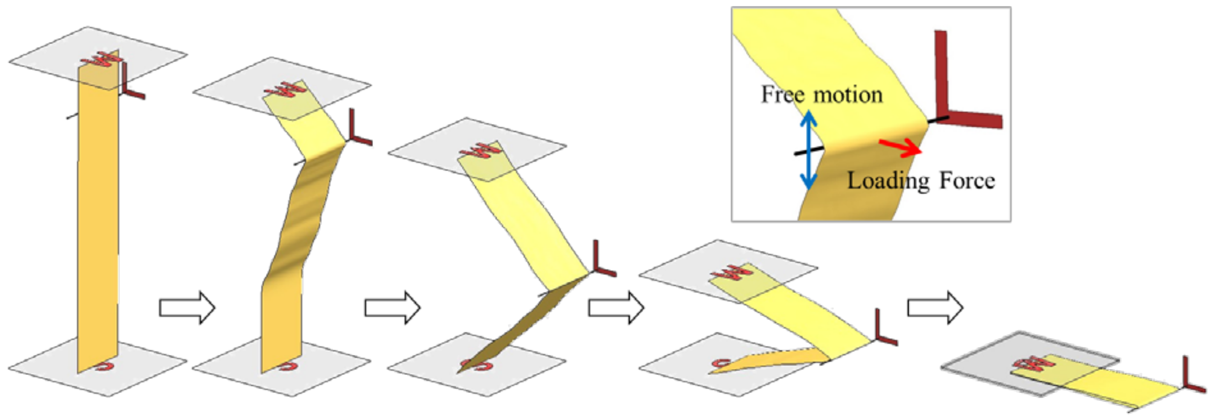

Figure 9: Example of folding executed by Load type tool.

7.4 New method to insert inflator model inside airbag

In JFOLD Version 1, a "flatten inflator model insert method" was implemented [1]. The inflator is first flattened then inserted into a small flat gap inside an airbag. During the simulation the inflator gradually returns to its original shape, pushing open the gap by contacting the airbag. This method can be used successfully for some airbag models, but when the area of airbag in which to insert the inflator is bumpy or curved the method cannot be used.

In JFOLD Version 2, a new insertion method is proposed. Called the "double inflator models insert method" and shown in Figure 10, space for the inflator model can be created even where the airbag is curved or bumpy. The user reads two inflator models and positions one above and one below the airbag. In the simulation these inflator models move to their correct location. The upper inflator contacts only lower layer of the airbag, and the lower inflator only contacts the upper.

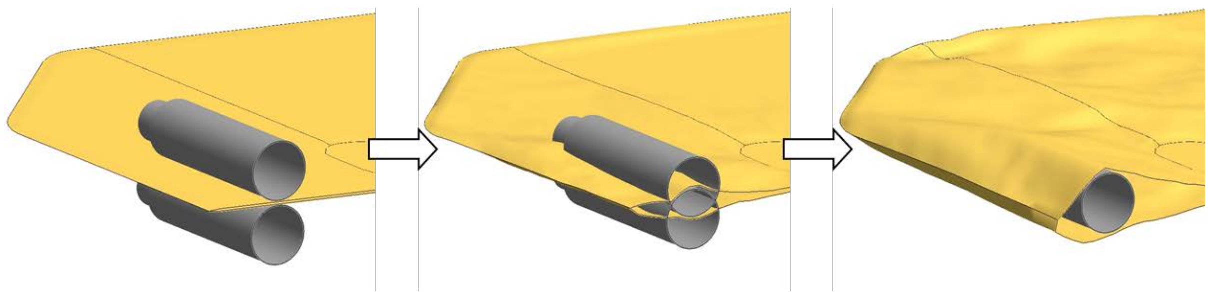

Figure 10: Double inflator models insert method.

In JFOLD Version 3 this new insert method will be improved to be more user-friendly. The preparation of two inflator models, their motion and positioning will be executed automatically using only a few mouse clicks.

7.5 Fitting simulation of curtain airbag to roof side rail

After a curtain airbag is folded into straight roll shape, it needs to be fitted to the vehicle's roof side rail. This fitting procedure can be performed by using finalGeom type tools as shown in Figure 11. Interior trim and bracket components can also be used during this process.

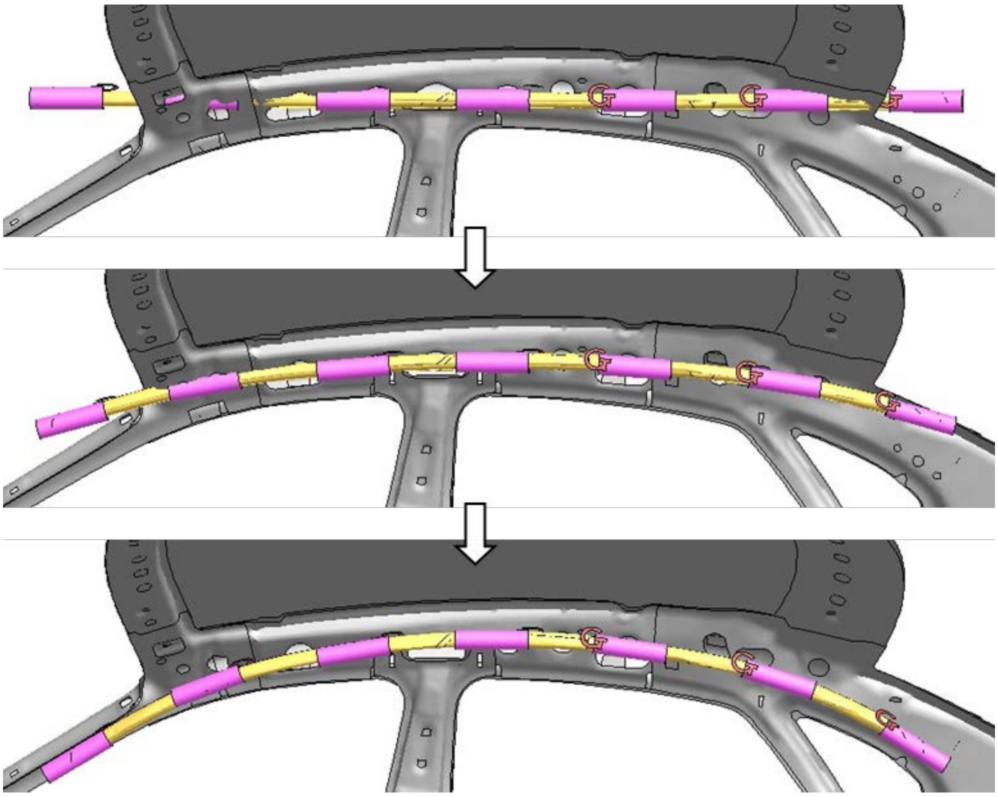

Figure 11: Fitting curtain airbag to roof side rail

8 Investigation of time to complete a folded airbag model

The simulation-based approach can be applied to almost any kind of folding pattern but it requires preparation and calculation time. There is strong demand to speed up the airbag modelling process because it can take up a significant part of the total time to develop a new airbag. The time needed to prepare input models for airbag folding simulation can be reduced by using a specialized pre-processor like JFOLD, thanks to its useful capabilities for setting up complex folding simulations in short space of time.

The calculation time depends on the number of elements in the folding model and the processing power available. As an example, the "flatten" folding step of the 3D shaped airbag shown in Figure 8 is about 2 hours; the airbag model has 180,000 shells with mesh size 2.5mm, termination time is 0.1 second, the calculation used 16 cores (Xeon E5 2.6GHz) on an MPP cluster machine. When the same airbag model has a 5.0mm mesh and 45,000 shells, the calculation time becomes 30 minutes on the same hardware. The number of trial and error simulations required to obtain a desired folding shape is also a critical issue because it often forms a large part of the development time. The more airbag folding experience (know-how) users have, the less trial and error simulations are needed. Such folding experience can be effectively stored within JFOLD as user-defined tools or Tool Assys, like a knowledge database system. Therefore the longer users use JFOLD, the more development time is reduced.

We propose another way to reduce development time. First, use a coarse mesh airbag model (with small number of elements) in JFOLD and set up all the tool shapes, positions and motion definitions. Although some trial and error calculations are needed, the final folded airbag can be created in a relatively short space of time. Some crossed edges and wrinkling will occur due to the coarse mesh size but can be ignored at this stage. Once complete, the coarse mesh airbag model can be replaced with a fine mesh model and the folding simulations re-analysed. JFOLD has the capability to easily replace one initial airbag model with another while keeping all existing tool definitions. Some fine tuning may be required but most of the hard work of tool set up is done using the fast running model. The process of re-running a chain of folding steps with a new airbag will be more automated in future versions.

9 JFOLD execution environment and license

JFOLD runs inside the powerful pre-processor Oasys Primer and also uses the post-processor Oasys D3plot. JFOLD requires a separate license, in addition to the Oasys license. JFOLD licenses are available as node-locked or networked types. JFOLD does not include the LS-DYNA solver – all folding simulations must be performed using the customer's own LS-DYNA licenses.

10 JFOLD Version 2 release plan

JFOLD Version 2 is planned to be released in Japan in December 2014 and worldwide a few months after Japanese release. New tool libraries and folding example models will be included in the JFOLD install package.

11 Future development plan

JSOL will continue to develop and upgrade JFOLD. The following developments are planned in the next or future versions:

- New functions according to customers' requests
- Usability improvements to the GUI system
- Changing the tool shape interactively by users
- System to automatically submit a chain of existing folding simulations
- Continued development of a new type of folding method focusing on 3D to 2D flattening

12 Summary

JSOL Corporation has developed a new simulation-based airbag folding system for LS-DYNA called JFOLD. JFOLD will be continuously enhanced to enable users to develop airbags more efficiently.

13 Acknowledgments

The authors show grateful appreciation to the software developers of Oasys Primer for their continued cooperation, and wish to thank all co-workers of JSOL Corporation in the JFOLD development team.

14 Literature

[1] S. Hayashi: "JFOLD – Introducing A New Simulation-Based Airbag Folding System for LS-DYNA", 9th European LS-DYNA Conference, Manchester, 2013

[2] LSTC: "LS-DYNA General Update", Oasys LS-DYNA UK Users' Meeting 2014