Using JFOLD & LS-DYNA to Study the Effects of Folding on Airbag Deployment

Richard Taylor[1], Shinya Hayashi[2]

[1]Ove Arup & Partners International Ltd.
[2]JSOL Corporation

1 Abstract

Today's engineers and designers need accurately folded airbags in their deployment simulations. Minor changes in the way the real airbag is folded can cause different deployment behaviour, which can lead to trim fracture, hang-ups and unintended injury. There is a growing demand for robust, accurate analysis to study and mitigate these problems, and this requires a good prediction of gas dynamics through an accurately folded airbag. The corpuscular particle method (CPM) is being continuously improved to simulate the gas flow, but creating an accurate model of a folded airbag is still a challenge.

JFOLD is a software tool developed to meet the growing demand for fast and easy simulation based airbag folding. Running inside the pre-processor Oasys PRIMER®, JFOLD helps the user to quickly set up folding simulations that run in LS-DYNA®. Once the initial folding process is complete, the user can easily generate many different fold patterns by only making small adjustments to the set up.

This paper introduces some examples of using JFOLD Ver.2.1 to create different fold patterns and compares deployment characteristics using the corpuscular particle method in LS-DYNA.

2 Introduction

Automotive airbags exist to protect the occupants and reduce injury in a vehicle collision. The airbag design must meet a range of performance requirements at the system and component level:

- Sufficient energy absorption and cushioning to reduce occupant injury
- Gentle enough deployment to not injure out-of-position occupants
- Robust deployment from a tightly folded package into the right place at the right time
- Safe interaction with the vehicle interior trim, seat etc. during deployment
- Low weight, low cost, low environmental impact, etc.

Computer simulation is a proven aid for the design of airbags as cushioning restraint device. However there is still a pressing need to identify and solve component performance issues early on in the design phase [1]. These problems include trim fracture, airbag hang-ups or other malfunction during deployment, and are very disruptive to a vehicle development program and costly to solve at such a late stage. There is a demand for these failure modes to be detected and solved using computer simulation during the earlier design phase. This need along with out-of-position studies requires more accurate finite element models than those traditionally used for in-position occupant injury.

The airbag model must accurately simulate the same deployment shape, timing and power as the real device. This requires a good model of the gas dynamics, a realistic folded condition and correct fabric behaviour, along with robust contacts and realistic material models for the surrounding parts [1].

In addition to the demand for improved performance, other factors are driving folder development. Modern airbag designs are increasing in complexity: 3D shapes, detailed tucks and inner parts can only be folded by simulation based techniques and not by more traditional geometric based folding. Airbag suppliers are facing an increasing burden from car makers who want to move the analysis to the earlier product development phases. Manpower costs must be reduced by improving folder efficiency and reducing the required knowledge and experience. Car makers want to do their own simulations that require re-folding airbag models but this may not be possible if the supplier performed the original work.

In summary, there is a need for an airbag folding tool with the following characteristics:

- produces high quality folded models to predict accurate deployment behaviour
- enables the folding of complex airbags that could not be attempted with pre-processors
- easy to use, contains built-in tools and folding know-how
- speeds up overall modelling time
- allows the rapid investigation of various folding patterns
- enables sharing of folding data between supplier and vehicle manufacturer
- creates unencrypted input data allowing users full flexibility of access
- uses existing site solver and pre-processor to minimise software costs

3 Introducing JFOLD

JFOLD is a software tool developed by JSOL Corporation that helps the user perform simulation based folding on an airbag model [2][3]. It runs inside Oasys PRIMER as a JavaScript, and uses LS-DYNA to simulate each folding step. The JFOLD graphics interface is designed to be easy to use and intuitive, so only a basic knowledge of LS-DYNA or PRIMER is needed.

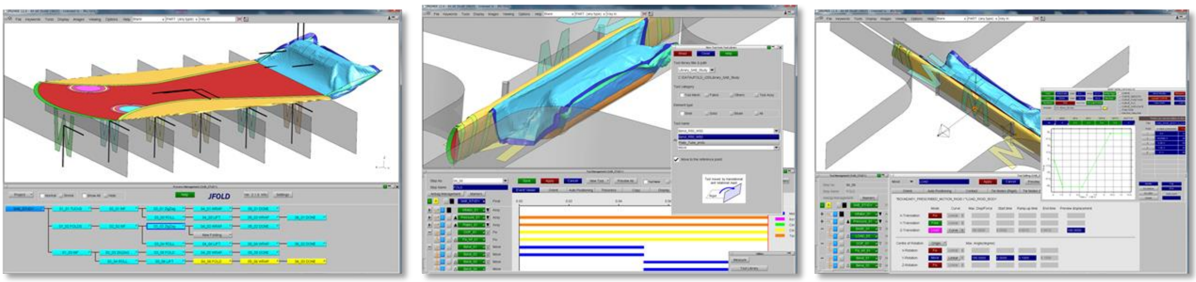

Fig.1: JFOLD graphics interface: process management, tool management & tool setting panels

3.1 How it works

JFOLD manages the folding processes in a series of "steps". Each step requires one LS-DYNA analysis to deform the model like a real fold, stitch panels or relax fabric. The airbag model is passed from step to step, using the deformed shape from the previous analysis. Folding steps can be modified, copied and branched off at any stage to investigate different folding patterns. “Tools” are used to deform the airbag and these can be copied across steps, imported from the built-in library or the user’s own.

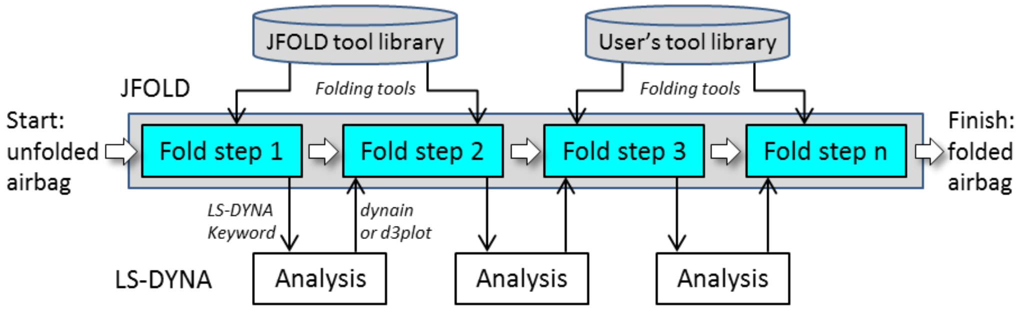

Fig.2: Schematic of one folding process using JFOLD

3.2 Project Sharing Capability

JFOLD data for each airbag project can be easily shared between car maker and supplier, enabling new research and greatly reducing time and costs.

A typical scenario might exist as shown below:

- The airbag model development is done by supplier
  - Includes folding and validation to impactor tests
- The car maker receives model and wants to study changes in trim design, or package, or the influence of the final folds on deployment direction etc.
  - OEM has to ask supplier to refold airbag adding cost and delays

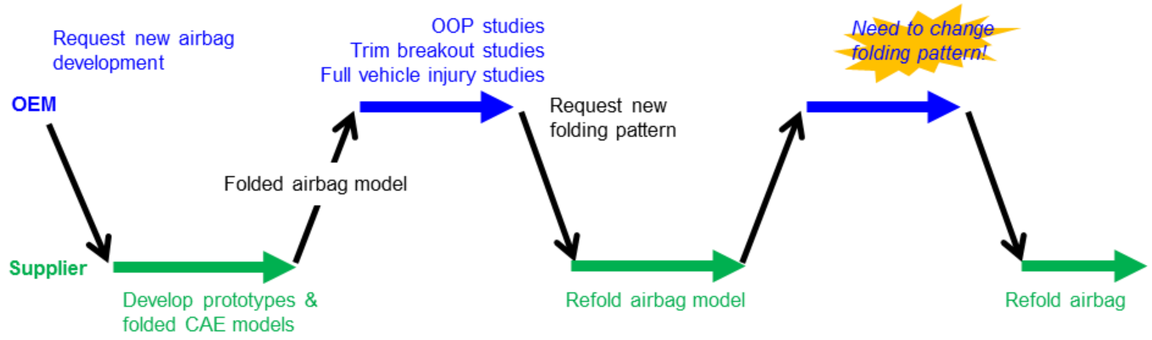

Fig.3: The problem: example of current airbag development between car maker (OEM) and supplier

The solution: share folding capability with JFOLD:

- Supplier and OEM both use JFOLD
  - Supplier shares the JFOLD project data with OEM
  - Includes all steps needed to repeat the folding process.
- OEM can easily modify the fold pattern to do own their studies
  - Complete folding know-how not required (supplier does it the first time)
  - Free to investigate package changes, effect of final folds etc.
- Less burden on supplier – less project delays
  - Saves cost and time

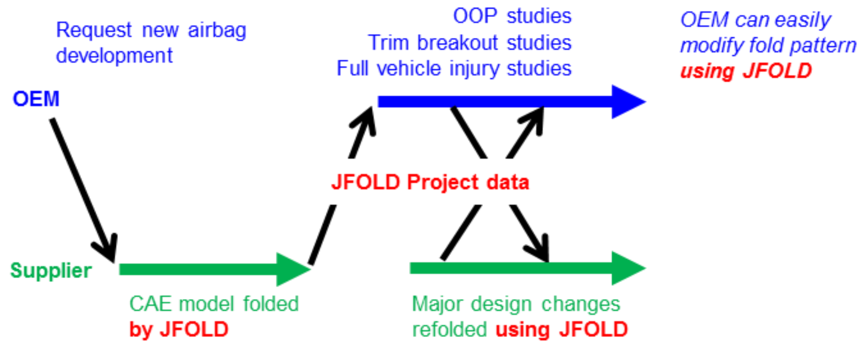

Fig.4: The solution: sharing the folding process using JFOLD

4 Example: Side Airbag Folding and Deployment Study

4.1 Study Objectives:

1. Demonstrate JFOLD using a simple but realistic airbag and common folding techniques
2. Compare the deployment performance of each folded design using CPM in LS-DYNA

JFOLD data and models created in this study will be made freely available to JFOLD customers.

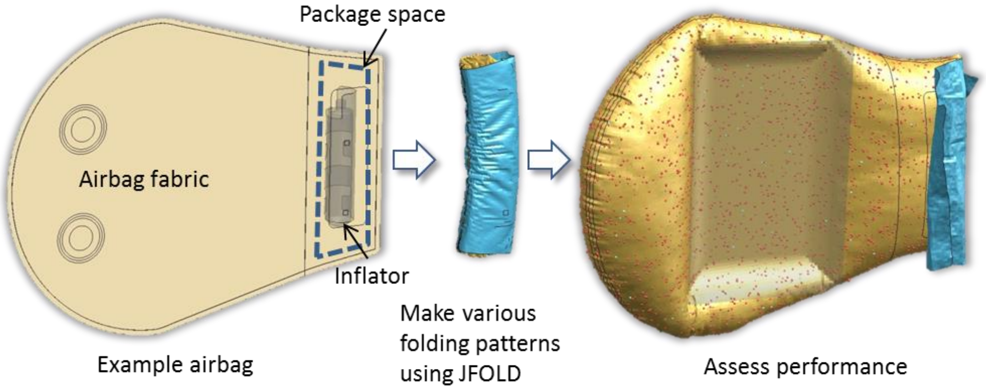

Fig.5: Side airbag example model for study

4.2 Folding Patterns:

For this study a simple single-chamber thorax side airbag was created and folded into six different patterns using JFOLD Ver.2.1. The unfolded geometry and folded package space are shown above. The top and bottom sections of the bag were folded into the vertical package space using the three different methods shown below:

1. Tucking in between the main panels
2. Folding on top of the main panels
3. Bending over using "thick folds" after the main panels have been folded

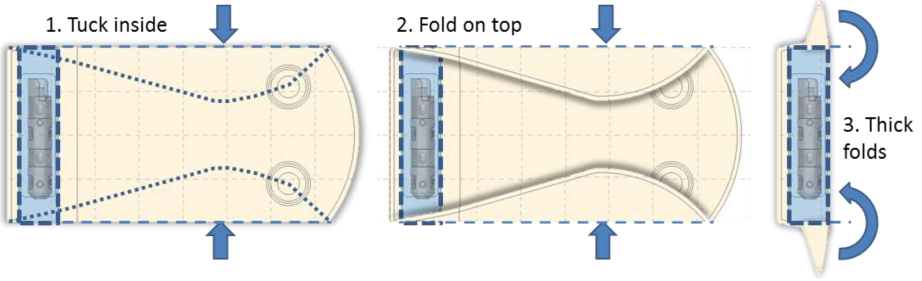

Fig.6: Three ways to fold top and bottom sections of fabric material

The main section of fabric material was folded using two common methods:

1. A zig-zag approach (also known as leporello, accordion or concertina pattern)
2. Rolling up the material around a flat batten

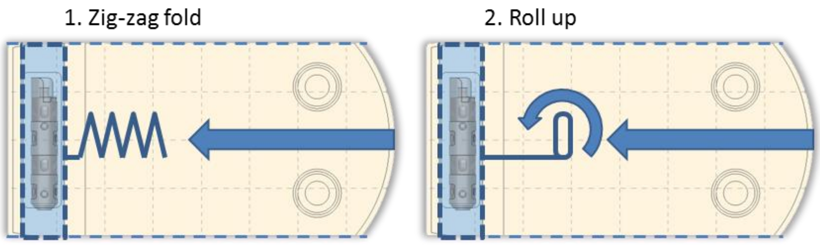

Fig.7: Two ways to fold up the main section of fabric material

The two methods were combined to make a total of six folded airbags. Finally in all cases a wrapper was fitted at the same time as fabric relaxation, to leave a realistic folded condition.

The whole process with all folding steps is shown below, in the JFOLD Process Management panel of this folding project. In the "root step" on the very left the unfolded airbag is imported and set up. Each button in the flowchart is a folding step that manages the input for an LS-DYNA analysis. Once a set of tools is defined for one step it can be copied to any other step in the same project with just a few mouse clicks.

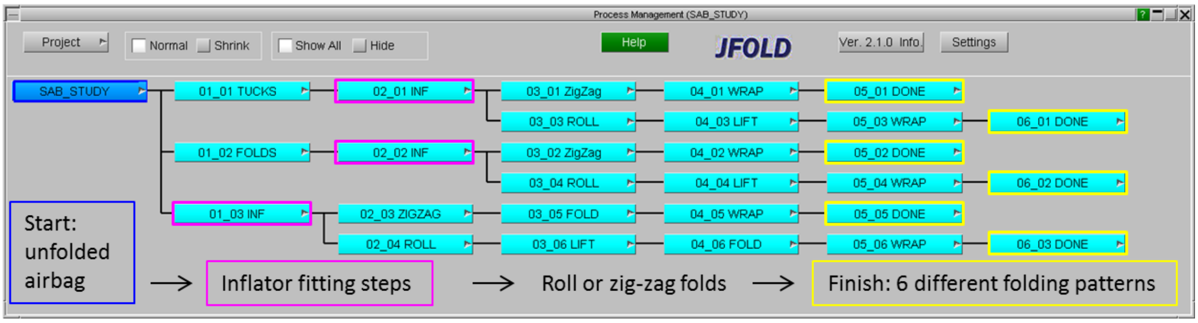

Fig.8: JFOLD Ver.2.1 Menu showing the six fold variants

4.3 Objectives of each folding analysis

An accurate folded model requires the careful set up of each folding step and assessment of each LS-DYNA result. Objectives to consider during set-up and assessment include:

- Accurate representation of the real folded fabric (shape, number of layers or rolls, overall thickness, etc.)
- No (or as few as possible) crossed edges and minimum contact surface penetrations
- Minimal change in fabric shell area (stretch or shrinkage) compared to unfolded condition

The default material and contact settings in JFOLD are designed to give the best result for a range of mesh sizes, although experience suggests that fabric elements of average length 2~3mm are much easier to fold than those 5mm or larger. In this study a 2.5mm mesh was used and this also enabled more realistic deployment behaviour than is often seen with 5mm mesh size.

4.4 Techniques for simulation based folding

The following section discusses some folding steps in more detail. Calculation time for each step averaged from 10 to 20 minutes using 16cpu of a typical MPP cluster. Excluding "thinking time", the whole task took just over one day to produce six models.

4.4.1 Tuck Folding

In this example tucks are made using a simple combination of nodal constraint and varying pressure. Pressures are defined to open up the airbag, push the tucks inside then close the airbag. If the mesh is fine enough this simple approach avoids the need for plates to provide a straight edge.

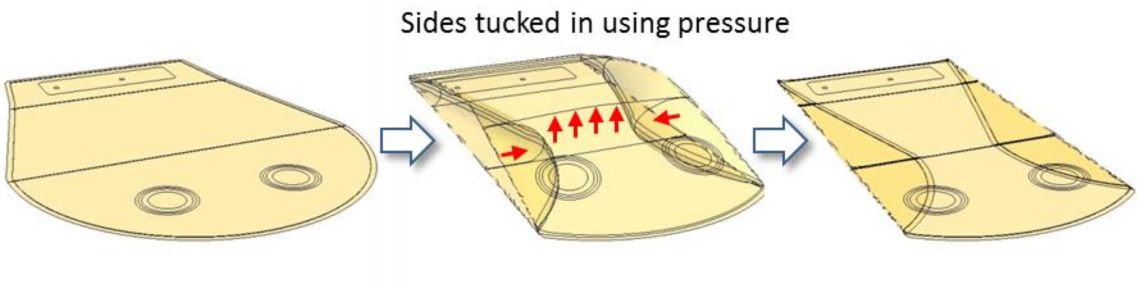

Fig.9: Tuck fold simulations are very easy to set up in JFOLD

4.4.2 Thin Folding

Two airbags in this study were folded using JFOLD's "Fold type tool assembly", which is available in different sizes from the built-in tool library. The tool assembly comprises four plates: two stationary and two that provide the folding action. Once scaled to fit and positioned astride the target fold line, JFOLD's "Auto-Move" function was used to position all plates just slightly away from the airbag fabric automatically. In this model the mesh did not align with the fold lines but even so, a clean fold was generated with no penetrations. After one side was set up, the tool assembly was easily copied and rotated to fold the other side in the same analysis.

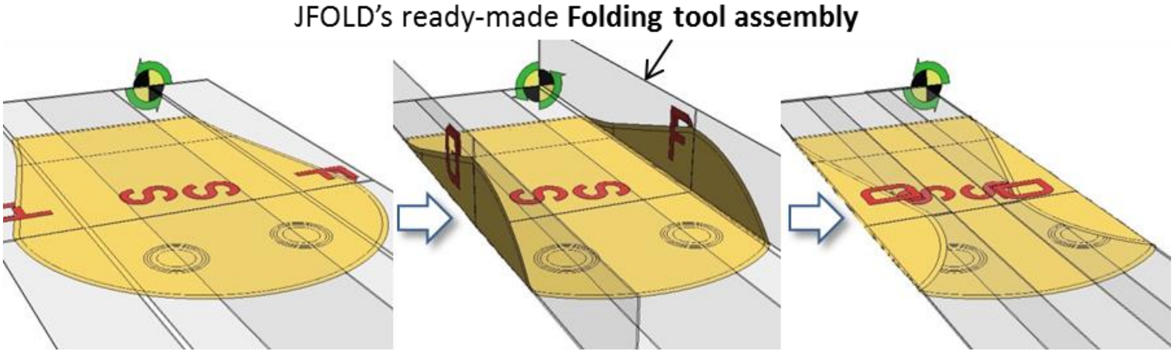

Fig.10: Thin folds created using two of JFOLD's Folding Tool Assemblies

4.4.3 Inflator Fitting

The inflator was fitted inside each model by using two copies of the inflator mesh to push open a space between the fabric layers. The thin-fold step described above required a completely flat bag, so for that airbag variant the inflator was added after the thin fold step.

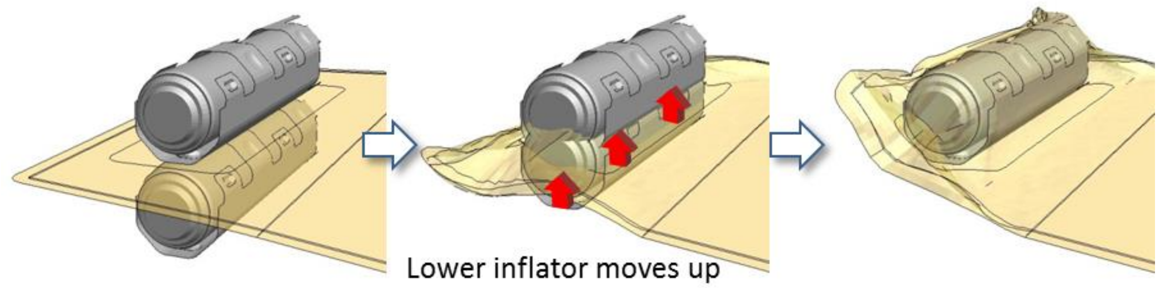

Fig.11: Fitting the inflator

4.4.4 Zig-Zag or Concertina Folding

There are many techniques to achieve this kind of fold pattern, and new ways are being continuously researched for JFOLD's example library. This example uses pressure on the top and several plates under the airbag to bend the fabric – a method well suited to JFOLD's ability to quickly replicate existing tools and change tool actions. This approach is very robust and seems to work well across different mesh sizes, widths of fold and layers of fabric.

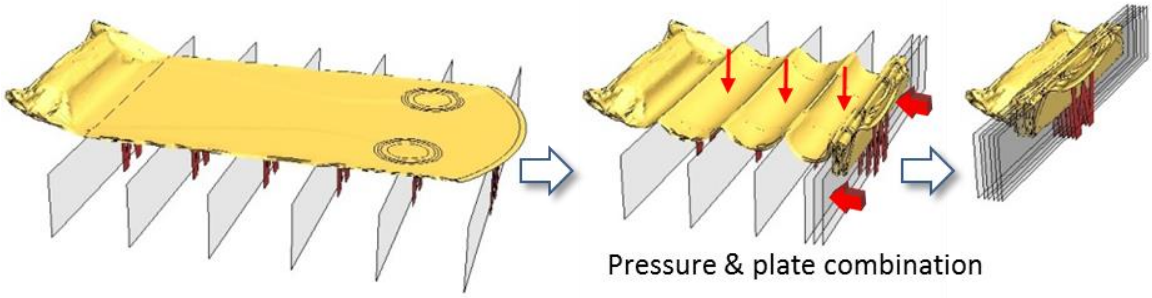

Fig.12: Zig-Zag Folding using pressure and moving plates

4.4.5 Roll Folding

This is the “Roll type tool” from JFOLD’s tool library. The rotating tool can be a tube or flat batten of shell or solid elements. A discrete beam keeps a constant tension in vertical and horizontal directions. The tension needs to be adjusted according to roll speed and fabric thickness etc. A preliminary analysis was needed to tune the tool width to achieve the target number of turns with the material available.

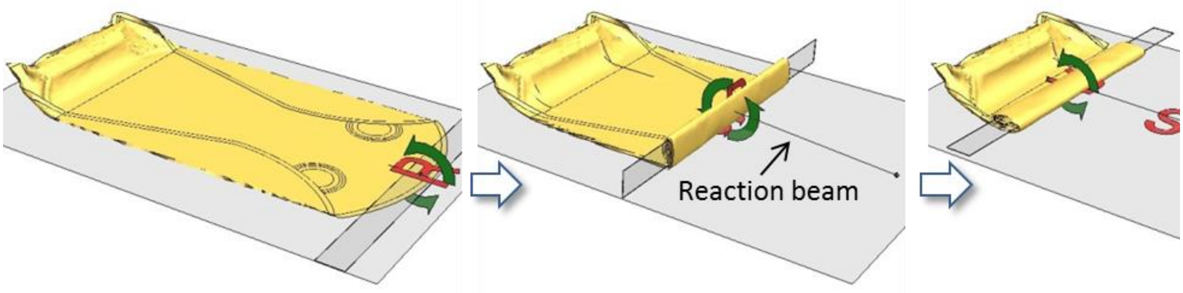

Fig.13: Roll Folding using a batten and reaction beam

4.4.6 Lift Folding after Roll

In some airbags a fold or two is added at the root of the roll to “launch” the rolled section forward. This is demonstrated using a Move type tool to lift the material. The batten from the previous roll fold was carried over in JFOLD and its “action” changed to rotate just 90^0, bringing it flat up against the inflator.

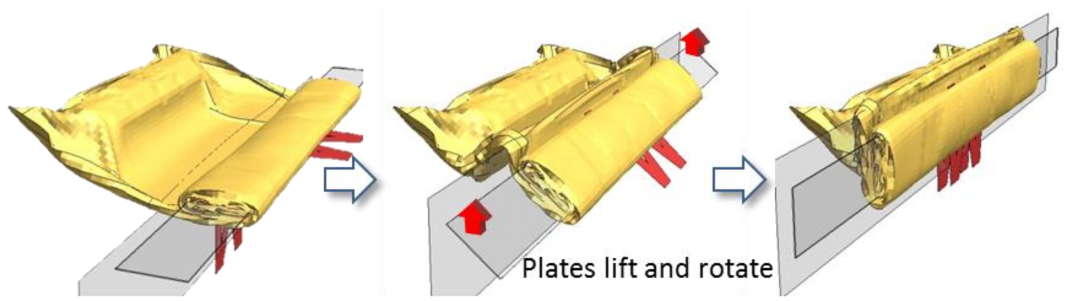

Fig.14: Lift Folding After a Roll Fold

4.4.7 Thick Folding (bending)

In this study two models were not given an initial tuck or thin fold at the start. Instead a “thick fold” was added at the end to bend the fabric material down into the package space. This is a challenging fold to simulate because elements on the inside of the bend feel high compressive forces, which generate contact penetrations and shrinkage. A compromise was achieved by allowing some shrinkage while not overloading the fabric. A 2.5mm mesh and the new DEPTH=45 contact setting also helped. Elements distorted during this step returned to their reference size in the next relaxation step.

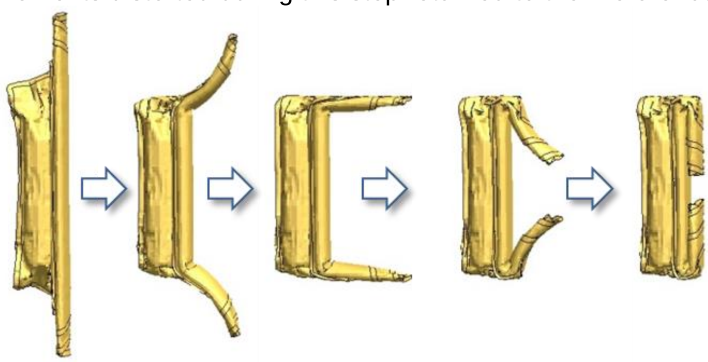

Fig.15: An example of thick folding or bending (tools not shown)

4.4.8 Wrapper Fitting and Fabric Relaxation

The final step was to "shrink-fit" a wrapper around the folded airbag, and relax the elements in the fabric material. In JFOLD other airbag parts like inner bags, wrappers and tethers can be imported at any step. Element relaxation is induced gradually by restoring strains using the TSRFAC loadcurve in ***MAT_FABRIC**. In this example a restraining plate stopped the relaxing bag from unfolding before tension in the wrapper pulled it tight.

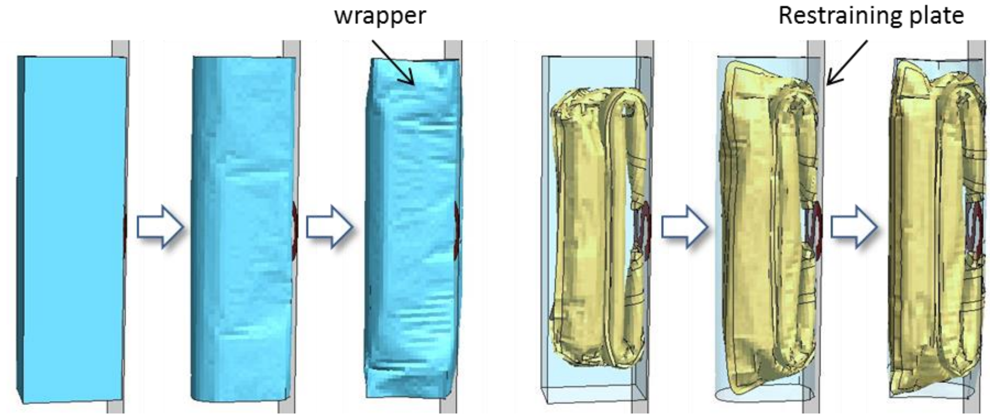

Fig.16: Wrapper fitting and fabric relaxation

4.4.9 Final step

A final step for each of the six models was created to import the nodes from the previous step, then all six airbag models were exported from JFOLD in preparation for the CPM deployment analyses. Because node and element IDs were identical for all models, the folded ***NODE** data was copied to separate ***INCLUDE** files, and all airbag parts, materials, sections, contacts etc. were defined in just one ***INCLUDE** file. A strip of elements in the wrapper was removed to create a perforated tear line. Failure properties were added to the wrapper fabric using ***MAT_ADD_EROSION**.

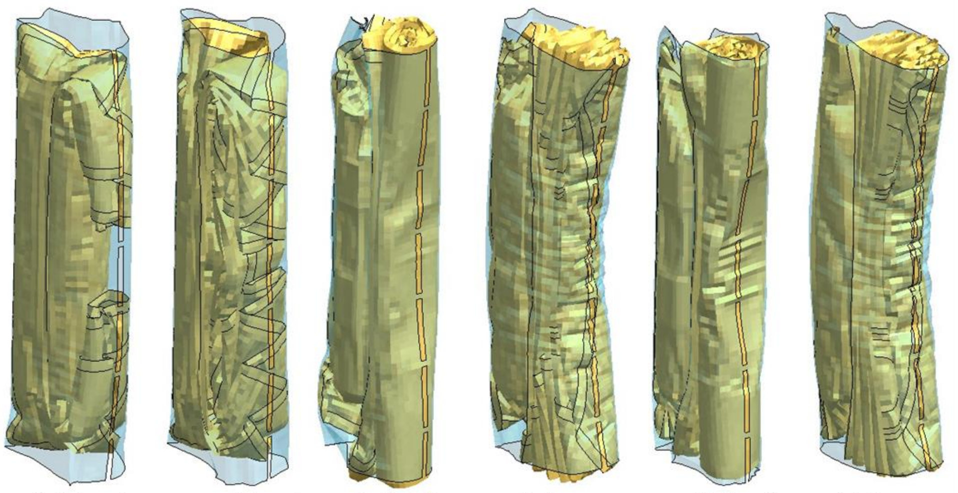

Fig.17: Six complete folded models with perforated tear line added

4.5 Deployment Results

Two loadcases were analysed for each of the six models:

1. A 20kg, 6m/s impactor analysis to measure force response, energy absorption and compare deployment shape.
2. A 10kg stationary “reverse-impactor” analysis to compare the “agressivity” of each design in out-of-position cases.

Three positions of impactor and reverse impactor were studied: “close range” (50mm to inflator), “medium” (100mm) and “far” (150mm). 100,000 particles were used in the 6-liter airbag. DEPTH=45 was used for the airbag self contact to ensure clean deployment and consistency across models. Average calculation time for 50ms analysis was 30 minutes on 16cpu of Xeon E5 (2.6GHz) cluster using MPP R7.1.2 (s).

4.5.1 Impactor tests

Very little difference was seen between the models that have a tuck initially or a thin fold. In a larger or more complex airbag design it is expected that the tuck would inflate quicker than a fold due to internal pressures being applied to more of the folded material.

In this example the rolled airbags inflated slightly slower and with a narrower profile in all directions than the zig-zag designs, which expanded quickly in all directions. The zig-zag-then-bend (thick folded) design was the quickest to deploy, and also the widest early on. The roll-then-bend model took longer to inflate because the roll was impeded by being bent in two places – this seems the design most susceptible to poor deployment.

All bags were fully deployed by 10ms and due to the small single chamber design the pressure/force applied to the impactor was very similar in all cases.

For occupant protection, good coverage is best achieved by a fast but narrow deployment profile, to enter the rapidly closing gap between occupant and door trim. The rolled designed seemed to perform best in this regard.

Apart from the initial peak in the 50mm case, forces on the impactor and overall energy absorption were very similar for all folding designs.

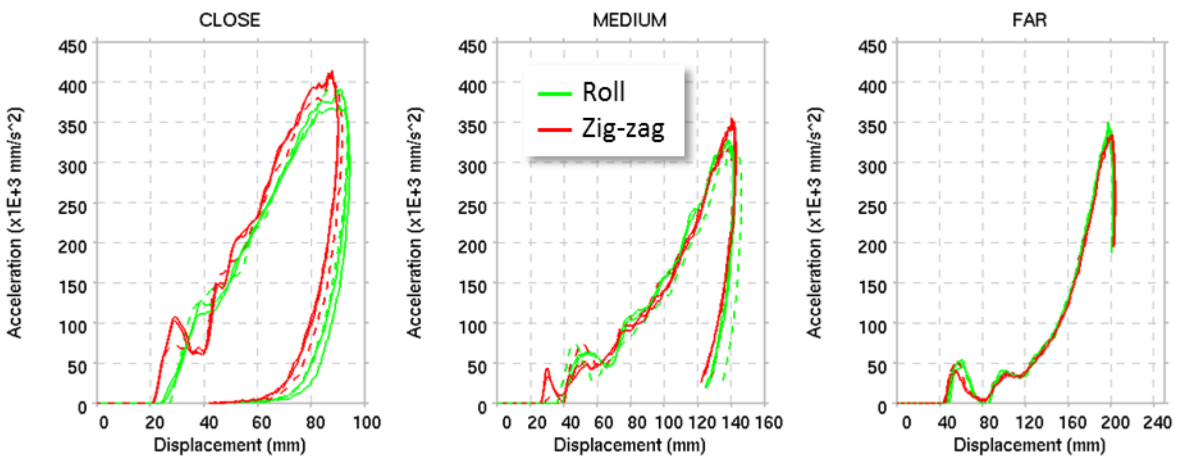

Fig.18: Moving impactor analyses: accel. vs. displacement for six airbags at three distances

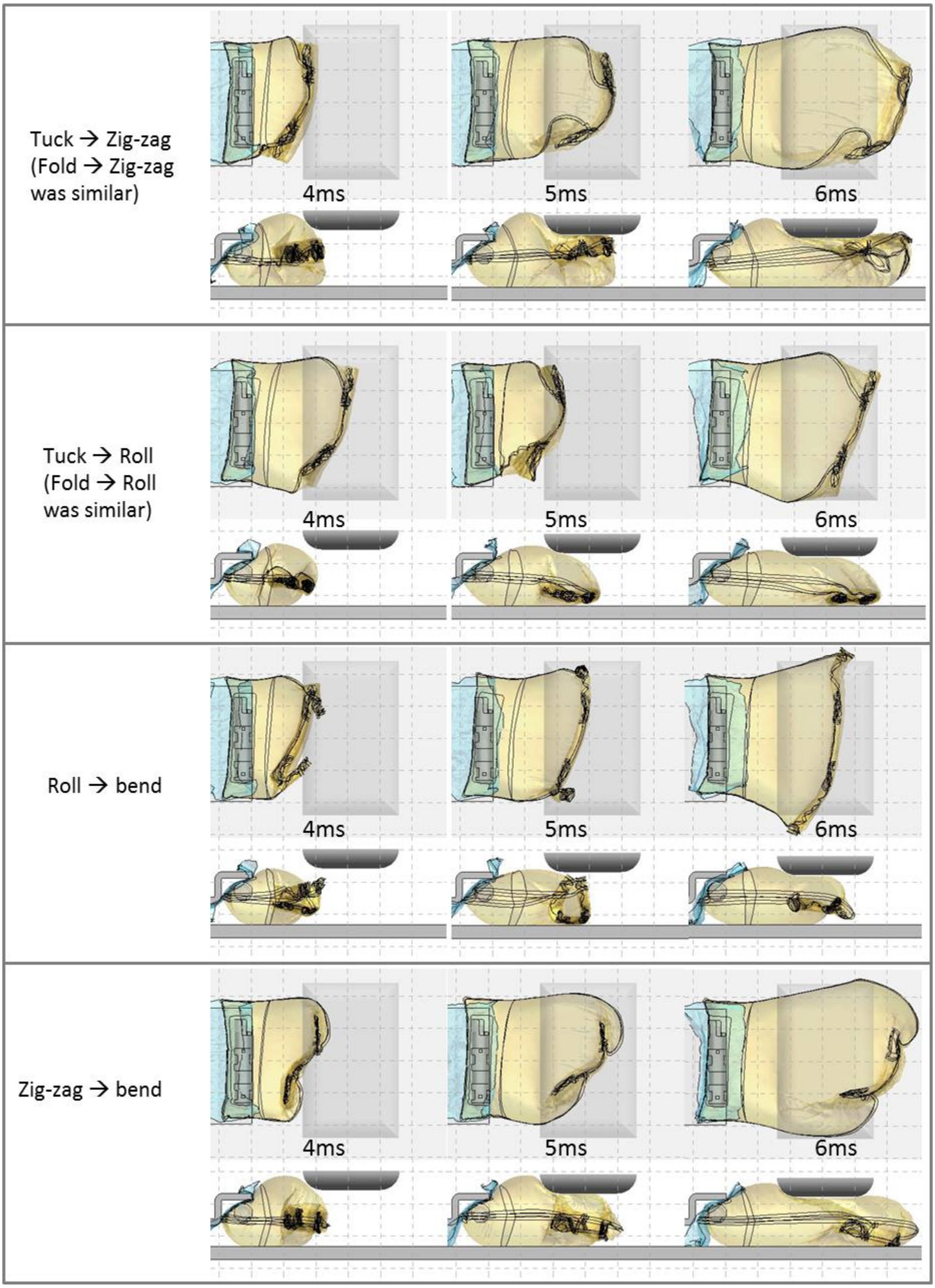

Fig.19: Moving impactor analyses (close distance): deployment from side & top view

4.5.2 Reverse Impactor tests

Again, little difference was seen between the initial tucked and folded airbag designs, however other differences were very apparent. The zig-zag models quickly inflated sideways when forward motion was impeded by the impactor. In contrast the rolled-up airbags became stuck up against the impactor and side wall, and applied higher forces on the impactor for a longer duration. Although the set up was much simpler than a true out-of-position occupant case, the results suggest that a zig-zag folded would be less aggressive than a rolled up one in this case.

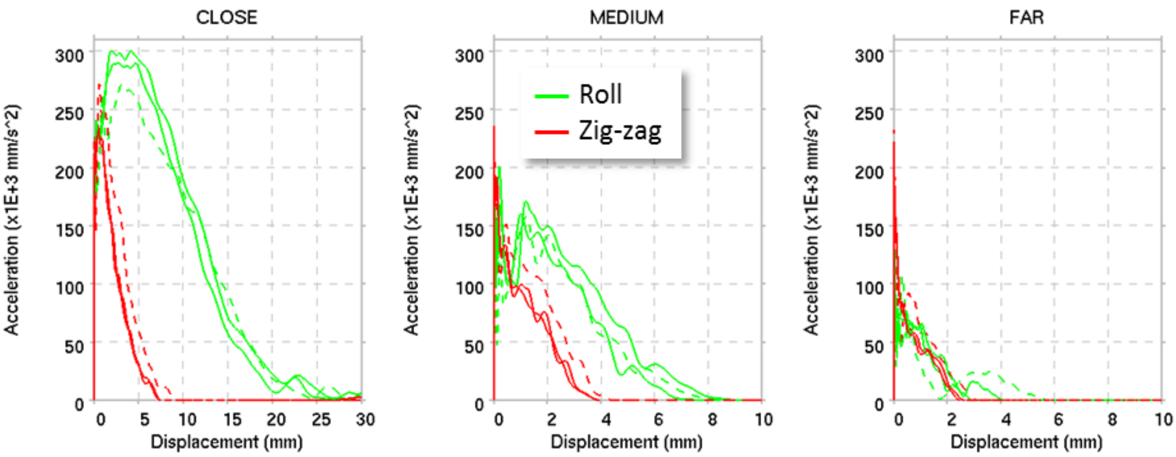

Fig.20: Reverse impactor analyses: accel. vs. displ. for six airbags at three distances

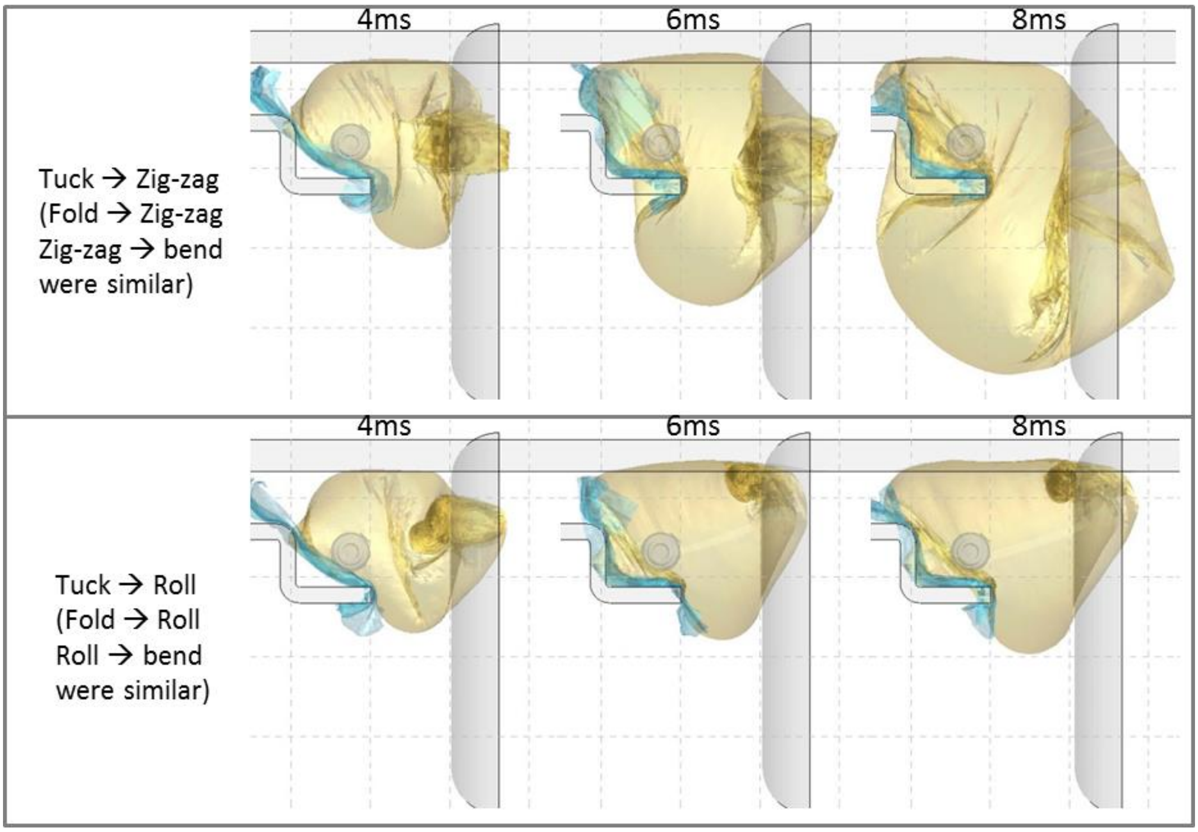

Fig.21: Reverse impactor analyses (close distance) view from below

4.5.3 Conclusions from side airbag study

Using JFOLD, six different realistic fold patterns were generated much quicker than could be done using just a pre-processor alone. This was thanks to the built-in library of folding tools, the graphical interface giving instant access to what needs to be set up and the default contact and material parameters which gave good results first time in LS-DYNA.

The simple design of airbag chosen for this study made folding and analysis quick and easy, but unfortunately the small, single-chamber design greatly reduced the effect that different fold patterns might have on deployment shape and timing. From experience it is expected that a larger, more complex airbag would show more deployment sensitivity to the different folding patterns.

5 Other Examples

5.1 Curtain airbag

In the example below two models of the same curtain airbag design have been rolled around a 15mm diameter tube and in this step, extra folds are added to the root of the roll to aid deployment. The top model in the diagram below has 5mm mesh and the bottom model 2.5mm mesh. Using the same contact thickness and materials the 5mm mesh model forms a larger diameter roll than the 2.5mm model due to geometric constraints. This is important to consider when choosing mesh size and target outer diameter. In both cases JFOLD successfully added the zig-zag folds as shown, but again the 5mm mesh models ends up slightly wider than the 2.5mm due to the larger mesh.

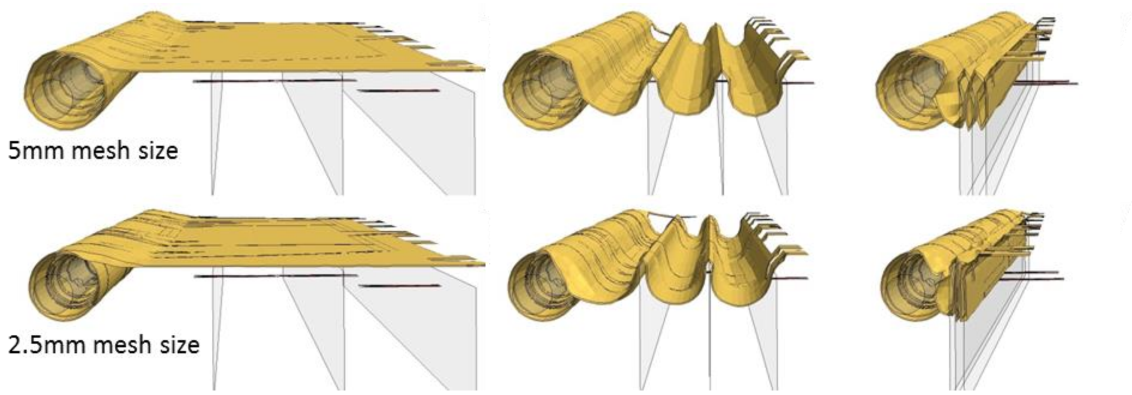

Fig.22: Folding curtain airbags of different mesh size

Curtain fitting techniques are being continuously developed and improved, with new tools/functions added to JFOLD with every release. Some curtain airbag folded designs require a twist at certain sections to ensure a clean deployment over the b-pillar or c-pillar trim. JFOLD can be easily used to tune the twist angle and area using a simple Move tool attached to a section of the roll.

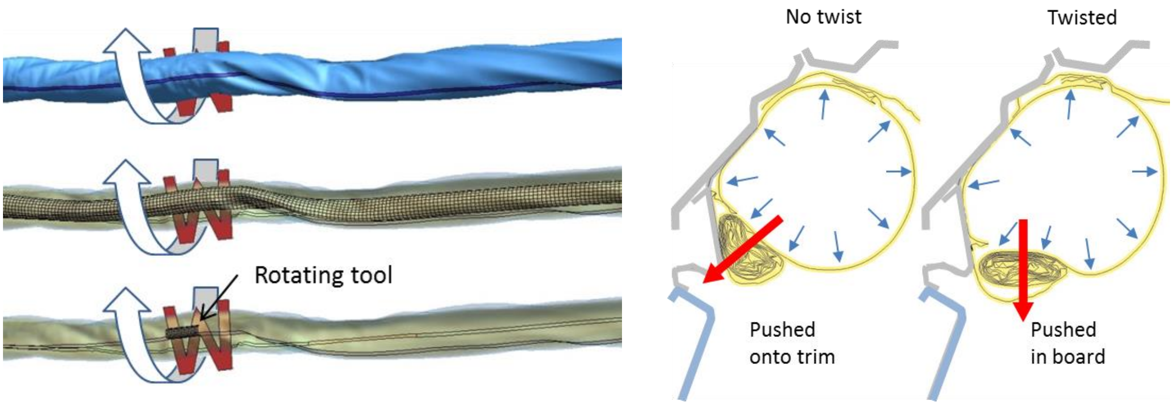

Fig.23: A section of airbag twisted inboard using JFOLD's Move tool deploys away from trim

5.2 Unrolling airbag deployment using CPM

Many users experience slow unrolling airbag models using CPM. This can be sometimes mitigated by choosing a smaller mesh size [4], although this increases run times and is still not always sufficient. LSTC report that the cause is linked to (i) initial air particles prematurely pressurising the rolled section and causing blockage, (ii) atmospheric pressure not being sufficiently countered by pressure from initial air causing a vacuum and (iii) fabric elements at the gas front not opening up sufficiently to let the particles enter the narrow gap in the rolled area.

A new setting IAIR=4 is being developed to counter these issues and preliminary results are very promising. Based on IAIR=2, initial air particles are still generated but only used to track the gas front. This is used to determine which segments should be given the initial air pressure contribution, which is applied using a method similar to LOAD SEGMENT.

Preliminary results are shown below. Two models each containing four rolled airbags were analysed. Each airbag is identical except for mesh size (2mm, 3mm, 4mm and 5mm left to right) and number of particles, NP (set to the same number as shell elements in the airbag part set: 56422, 24688, 14204 & 9012). NP_AIR was set to 10% of NP. Each airbag is 500mm long and 250mm wide and split simply into two chambered areas. All were rolled up using the same tube roll tool in JFOLD.

The model on the left uses IAIR=2: the unrolling speed can be seen to vary according to mesh size. The model on the right has IAIR=4: the unrolling speed is faster than the IAIR=2 model and much closer for all mesh size airbags. It is important to note that the IAIR=4 speed-up affect is significantly reduced if a larger number of particles is used. DEPTH=45 was also found to enable faster unrolling than DEPTH=5. Research and development is still on-going.

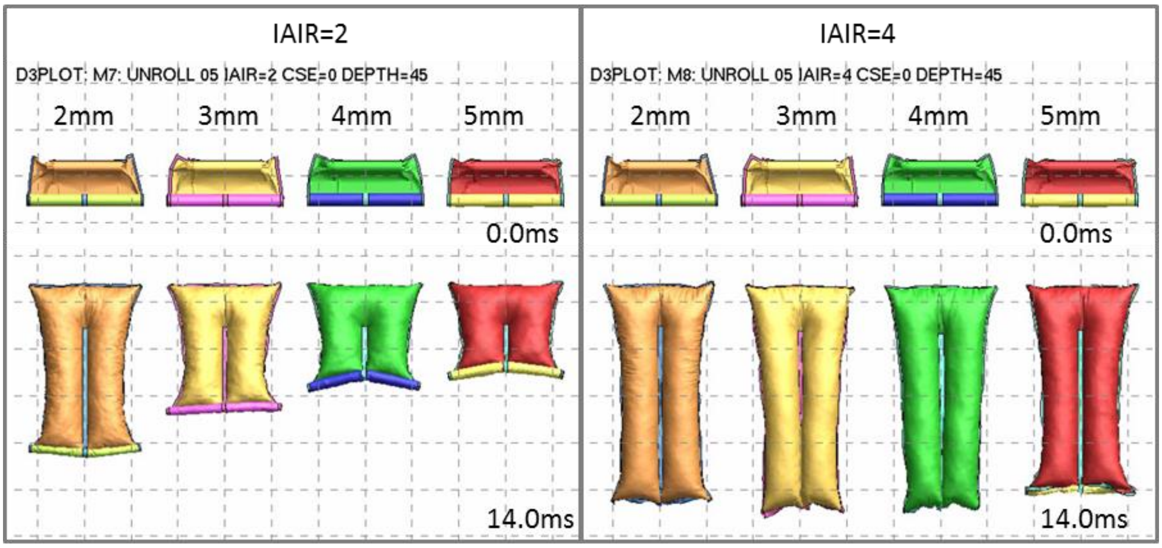

Fig.24: Comparing unrolling speed between different mesh size, IAIR=2 and IAIR=4 models

6 Summary and Conclusions

In this paper JFOLD Ver. 2.1 and LS-DYNA were used to study the deployment behaviour of a simple side airbag with six different folding patterns. It was found that the rolled type deployed into a narrower space, but the zig-zag pattern was less aggressive in an out-of-position scenario.

Various JFOLD tools and techniques were used to quickly generate the models. The conclusion is that JFOLD is a proven enabler for fast and realistic simulation based folding.

User feedback, requests and continuous research into quicker and easier folding techniques drives the development of the software. The next version Ver. 3.0 will have even more capabilities.

7 JFOLD Information

JFOLD Ver.2.1 is a JavaScript based software tool that runs inside Oasys PRIMER v12. It has the option to integrate with Oasys D3PLOT v12 but this is not essential to perform all functions. JFOLD requires an additional license in addition to those needed for Oasys PRIMER and LS-DYNA. Licenses are available as node-locked or networked, using the same FlexLM system as Oasys PRIMER. For the latest Ver.2.1 release English and Japanese language installations are available. The product includes an extensive built-in help manual and also a self-learning based tutorial with example model. A variety of folding example models will be available from JSOL Corporation for JFOLD customers. For more details please see your local JFOLD distributor.

8 Literature

[1] Feng, B., Coleman, D.,: "Gas Dynamic Simulation of Curtain Airbag Deployment through Interior Trims", 10th International LS-DYNA Users Conference, Detroit, 2008

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

[3] Hayashi, S., Taylor, R.,: "Simulation-Based Airbag Folding System JFOLD Version 2 -New Capabilities and Folding Examples", 13th International LS-DYNA Users Conference, Detroit, 2014

[4] Sugaya, H., Imura, K., Mae, H.: "Recent Investigations of Side Curtain Airbag Deployment Simulation Using CPM", 9th European LS-DYNA Users Conference, Manchester, 2013