

Development of a Customized Beam-to-Shell Element Model Mapping Tool

Miro Duhovic, Pratik Patil, David Scheliga, Dominic Schommer, Lukas Münch, Joachim Hausmann

Institut für Verbundwerkstoffe GmbH, Erwin-Schrödinger-Str., Gebäude 58, 67663 Kaiserslautern, Germany

Abstract

A customized solution enabling the mapping of fiber orientations represented by beam elements in organic sheet materials from one simulation phase to another of the product development cycle has been developed using python scripting language. The strategy implemented for the mapping of the fiber orientations is based on the modeling approaches used for the input models in both types of simulation. The thermoforming simulation model consists of beam and shell elements representing the fiber and polymer layers in an organic sheet respectively while in the structural simulation, the component is usually modeled using only shell elements. The thermoforming simulation results (a .d3plot mesh) and structural simulation model input mesh are provided as inputs to the so called "*BETA Mapper script*". The script segregates elements from each model into discrete volumes enabling parallel processing of the mapping procedure. The centroid coordinates of the elements from each matching cuboid are used to identify element pairs by finding the shortest distance between two element centroids. During the thermoforming process, the fibers, in the warp and weft directions of an organic sheet undergo a relative scissoring motion. In order to take this effect into consideration and to capture non-orthogonal fiber directions, the script is developed to produce three solutions and provides the possibility for mapping various types of part geometries. Developable geometries, which can be unfolded as a flat surface and do not exhibit any relative fiber scissoring, are mapped according to "*Solution 1*" and the part can then be simulated using assigned orthotropic material properties. "*Solutions 2*" and "*3*" implemented in the script, provide a methodology to enable the mapping of fiber orientations in two non-orthotropic directions in a single mesh model, which is not feasible with the generic approach using only one definition of the keyword (**PART_COMPOSITE*). Additionally, "*Solution 2*" implemented in the script provides the user with the flexibility to choose the number of individual parts to be generated during the mapping. The method facilitates the realization of correctly mapped fiber orientations of the warp and weft yarns of an organic sheet in a single mesh model. With its three solutions, the "*BETA Mapper script*" provides the required data integrity between the different phases of organic sheet virtual product development and enables overall improvement in product design.

1 Introduction

In existing FEM solvers, simulating the forming manufacturing process of fabric reinforced thermoplastic composites is still considered to be a challenge. In the past, only a purely isothermal forming process could be simulated using explicit FEM solvers. Now, with the extended multiphysics capabilities of the FEM solver LS-DYNA®, it is possible to simulate the shear and bending material behavior for a non-isothermal process and achieve an accurate prediction of fiber orientations and arising defects such as wrinkling. Woven fabric reinforced thermoplastic composites, commonly known as organic sheets, are widely used materials in aerospace as well as the automotive industry. High specific energy absorption capacity, strength and high stiffness make organic sheets suitable materials for lightweight structures. Unlike steel, fabric reinforced thermoplastic composites are anisotropic in nature i.e. the mechanical properties of organic sheets are dependent on the fiber directions. In general practice, composites are implemented using the stacking of layers exhibiting different fiber orientations. With this implementation, material properties of the resulting part are quasi-isotropic in nature. Nonetheless, in organic sheets, the yarns are interlocked with each other by a weaving process. During manufacturing of the parts, these fiber orientations determine the drapability and formation of wrinkling defects in the parts. Furthermore, parts which have undergone a forming process, local and overall final mechanical properties are extensively influenced by the resulting fiber orientations.

In case of a structural simulation, the fiber orientations in the finite element mesh of the model can be given a default orientation i.e. 0° , 90° etc. according to a global or local coordinate system for each layer of the composite stack. As stated above, the part properties obtained from the initial fiber

orientation will differ from the properties exhibited by the part after forming. In any product development cycle, the manufacturing processes through which the product has to go, can bring certain changes in material characteristics which greatly influences the performance of resulting part in real application. In current virtual product development cycles, there is a lot of emphasis on validation of different simulations to meet the product requirements by considering manufacturing effects on part performance. In order to bring robustness in the product development cycle and to reduce the impact of undesired effects of the manufacturing processes, the output parameters of one simulation result are often taken as input parameters for next simulation in the development cycle. This ensures implementation of all parameters that may affect the performance of the resulting product and enables better prediction of the final product behavior.

Product development in the composite industry faces different challenges related to simulating different material behaviors. The current FEM solvers are not capable of simulating all failure scenarios and element behaviors. Thus the industry implements different methodologies in development stages. In such cases to transfer the parameters, which influence the product performance is a supplementary challenge and cannot be completely addressed using existing FEM pre and post processors due to the large number of user defined modeling possibilities. It is therefore inevitable to find workaround methodologies to link between the different modeling strategies for closely coupling of the various stages of product development.

In present day composite product development, different simulations of composite parts involve different approaches for definition of laminas and the part structure [1]. The transfer or mapping of influencing parameters is largely dependent on the definition of laminas, types of element and modeling techniques [2] used to define the parts mesh geometry in the simulation model.

The following work gives an overview of existing mapping methodologies for transferring the fiber orientations and describes the development of a new mapping algorithm which can be used to combine different modeling approaches used in the product development cycle. The purpose of this work is to enable the accurate prediction of the material as well as component behavior in structural simulations while taking into account the fiber orientations which have evolved during non-isothermal forming simulation [3] and influence the properties of organic sheets.

2 Problem definition

Figure 1 exemplifies a typical product development and validation cycle for products made of continuous fiber reinforced thermoplastic composites.

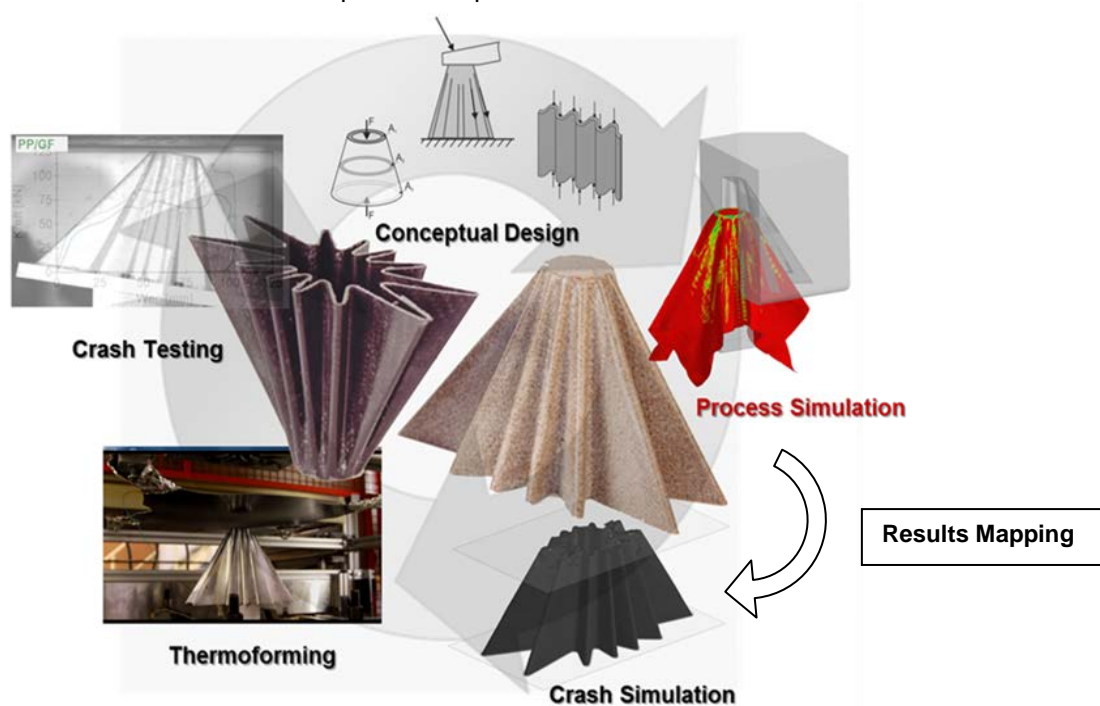


Fig.1: Typical product development and validation cycle for continuous fiber reinforced thermoplastic composites

The prediction of component behavior in real application can be accomplished by taking into consideration all performance influencing factors. To achieve more realistic solutions through

simulations, it becomes necessary to contemplate the changes introduced into the component due to manufacturing processes. These changes are governed by certain parameters which act as a coupling between two phases of the product development. In order to implement the effects of one process into the next process of the product development cycle, an understanding of parameters that affect the component and its material behavior, plays an important role. The product development and validation cycle involves various steps from product concept to product testing. This thesis work concentrates on the information transfer between two phases of virtual product development. Phase one represents the forming simulation while the subsequent phase is the structural or crash simulation.

During the manufacturing of a part, the fiber orientations have significant influence on drapability and on formation of wrinkles in the resulting part. Furthermore fiber orientations govern local and mechanical properties of the manufactured part. Hence fiber orientations prove to be an important parameter to be transferred to enable accurate prediction of the material properties as well as part behavior. Additionally to introduce the post effects of forming processes in product development, fiber orientations have to be determined and given as initial conditions for the subsequent development phase.

Within the scope of this work, fiber orientations of organic sheets are to be determined and provided as an input parameter for structural simulations. In order to enable this transfer or mapping of fiber orientations from one simulation activity to another, it is imperative to understand the modeling methodology used for each particular simulation. The following sections explain the methodologies used to create both the forming and structural simulation models in LS-DYNA®.

2.1 Organic sheet thermoforming modeling technique

The forming simulation model employed here allows simulation of a non-isothermal forming process. The methodology of simulation is based on a “hybrid unit-cell” modeling approach, which incorporates a combination of shell and beam elements. This modeling approach is developed on the idea from Duhovic et al. [2] and was implemented for a forming simulation method to predict resulting fiber orientations and the influence of stitches on a dry textile reinforced structure. A standard organic sheet thermoforming process is non-isothermal, as the organic sheet cools down rapidly after coming into contact with the tooling. Hence the method explained in [2] has been extended further [3] and applied to a non-isothermal organic sheet forming process.

Here, the shear and bending behavior of the fabric reinforcement along with the polymer resin is simulated by using a macro scale model. This model is created by using shell and beam elements which share specific nodes. The beam elements represent the bending stiffness of the composite material governed by orientations and properties of fibers. Hence in this model the beam elements are oriented in actual fiber directions and represent actual yarns of the fabric. Additionally, the shear stiffness of the composite, determined by the shear behavior of the reinforcement and resin properties, is represented using shell elements. Figure 2 illustrates the repeating unit cell used to build up a part scale macro scale model.

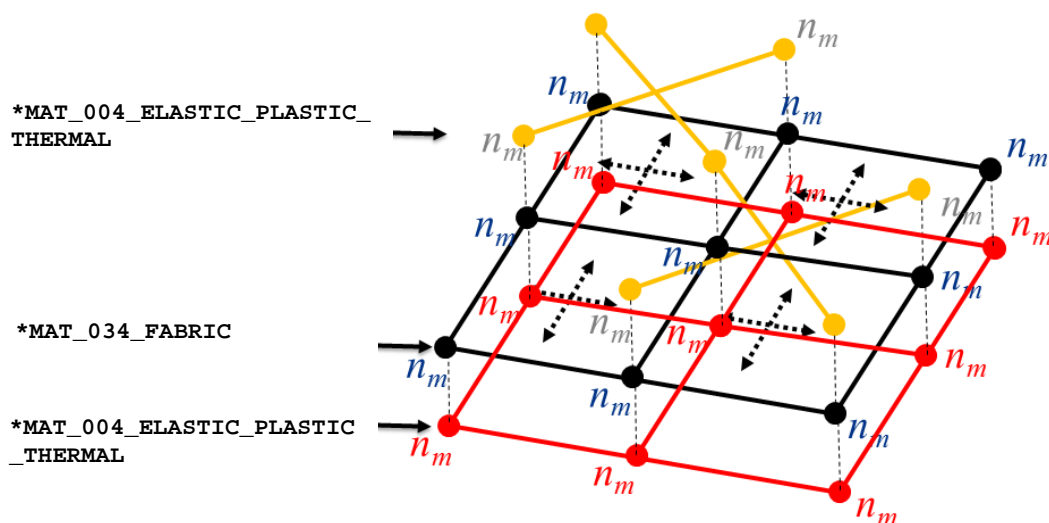


Fig.2: Organic sheet thermoforming modeling technique: Unit cell macro scale model [2] [3]

The crossing beam elements in Figure 2 are connected to the four nodes of an alternate shell element and have no independent nodes among each other. Such an arrangement of beam elements guarantees the free relative sliding motion during the shearing of organic sheets.

2.2 Organic sheet structural modeling technique

The next process that follows the forming simulation in the product development cycle is the structural simulation. For creating composite part models, LS-DYNA® provides various approaches through which the composite part modeling can be accomplished. In this case, the model used for structural simulation consists of generic shell elements using the keyword ***ELEMENT_SHELL** along with BETA options and the ***PART_COMPOSITE** definition. Furthermore the material model ***MAT_LAMINATED_COMPOSITE_FABRIC** is employed. In order to model different laminas of the composite part, LS-DYNA® enables a layer definition with initial fiber orientations, part thickness and MID for each integration point of the shell elements. The initial fiber orientations can be set in the ***ELEMENT_SHELL** keyword with the BETA or MCID definition and ***PART_COMPOSITE**. The ***ELEMENT_SHELL** definition along with the BETA option can only be used to create parts with orthotropic or anisotropic materials. As shown in Figure 3, the initial fiber orientations can be defined using local element coordinate axes. In case of a local axes, the fiber orientation β is an offset angle measured by considering the local X axis (N1 to N2) of the element definition. In case of an additional reference coordinate system, the angle between the element local X axis (N1 to N2) and X axis of the reference coordinate system defines the fiber orientation. Additionally in ***PART_COMPOSITE**, it is possible to define different layers with different fiber orientation and material models.

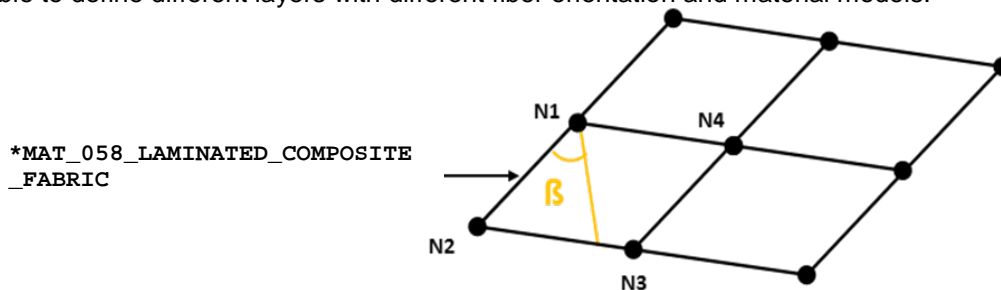


Fig.3: Organic sheet structural modeling technique: Structural model mesh created using ***ELEMENT_SHELL_BETA** and ***PART_COMPOSITE** keywords

2.3 Types of solutions

In order to take actual effects of thermoforming into consideration, it is a significant activity to calculate the fiber orientations from the forming simulation and map them onto the mesh geometry of a structural model. The mapping of parameters from one process to another is primarily governed by the type of modeling techniques used in the respective simulation models. In this case, the fibers in the forming simulation are modeled using beam elements, while the structural model consists only of shell elements. Thus actual mapping involves, finding the orientations of beam elements with respect to the coordinate system in which the fiber angles can be transformed and written as element data in LS-DYNA® keyword format. The actual implementation of the fiber orientation in a structural model is discussed in the following section by providing a few possibilities for modeling composite parts using currently available material and element cards. After the fiber orientations have been calculated during mapping, there are various approaches through which they can be implemented into a structural model. These approaches depend on the type of material model used for the simulation. The following solution types exemplify the possibilities of modeling orthotropic and anisotropic material models through which the structural simulation mesh model can be setup.

2.3.1 Solution 1

Solution 1 illustrates one of the simplest methodologies for modeling a composite part using LS-DYNA®. This solution implements an orthotropic material model. For example, Solution 1 represents a composite part which is made of laminas with fiber orientations of 0° and 90°. In this case, the first layer fiber orientations i.e. 0° are represented in a structural model using the ***ELEMENT_SHELL_BETA** keyword, where the actual fiber orientations from forming results are mapped. However the fiber orientations of the second layer with initial orientation 90° are considered to be default and always perpendicular to the first layer. After the forming simulation of organic sheet, it is expected that the

fibers undergo a relative scissoring motion and that the initial angles will lie in the range of 0° to 180°. Nonetheless in Solution 1, the composite part is modeled by assigning the initial reference fiber direction with the fiber orientations mapped from forming simulation while other fiber direction layer will always be perpendicular to the first. Figure 4 illustrates the implementation of Solution 1.

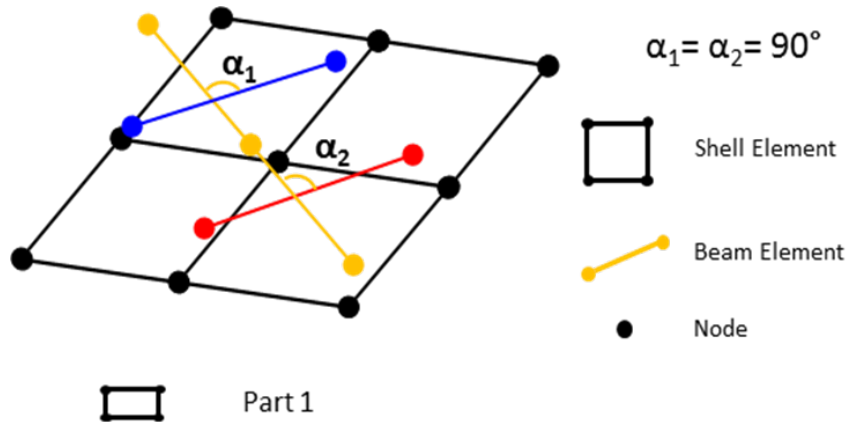


Fig.4: Composite modeling scheme according to “Solution 1” for structural model fiber direction mapping

Solution 1 realizes the composite part using only one part definition in an LS-DYNA® structural model. As shown in Figure 4, the beam elements blue and red represent the second layer in the composite part definition. The fiber orientations for this layer are set perpendicular to the first layer. Therefore, Solution 1 does not implement the effects of the forming simulation completely and the properties of resulting part will be different than that in a realistic scenario if yarn scissoring occurs. The consequences and relevance of this type of solution result is discussed later in Section 5.1.1.

2.3.2 Solution 2

In order to map fiber orientations to a structural model in a more realistic way, Solution 2 employs a second fiber direction from the organic sheet. Similar to Solution 1, the first fiber directions are mapped onto the structural model using the `*ELEMENT_SHELL_BETA` keyword. Furthermore, for realizing the second direction fibers in the organic sheet, the original part composite definition is split into different parts.

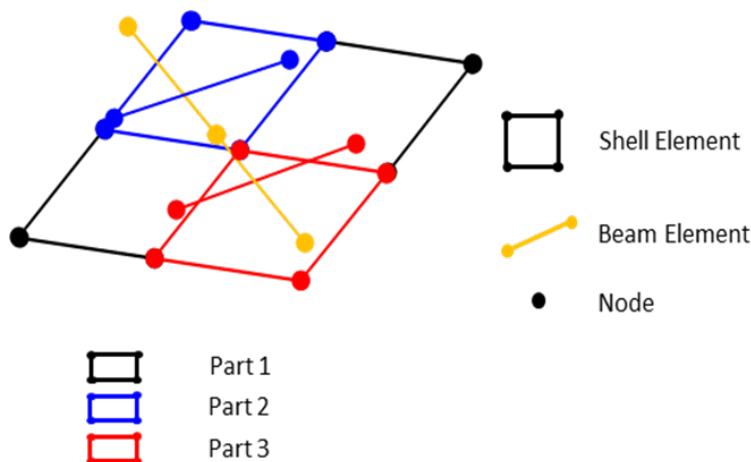


Fig.5: Composite modeling scheme according to “Solution 2” for structural model fiber direction mapping

Figure 5 illustrates the new composite part definition used to incorporate second direction fiber orientations of the organic sheet. After a forming simulation, the fibers in both the first and second directions are oriented differently and to take that effect into consideration, the shell elements are modeled using new composite part definitions. The range in which second direction fibers can be oriented after a forming simulation is considered to be between 0° and 180°. In order to generate the

new parts, the fiber orientations calculated from the forming results should be split further into groups defined by fiber angle ranges and shell elements should be assigned new parts according to these ranges. Here the user has the flexibility to define the number of parts which are assigned to all the shell elements. Depending on the number of parts defined by the user, the range 0° to 180° is divided into that many numbers of parts. Subsequently, shell elements are moved into a part definition that represents the fiber orientations to which the shell elements are to be mapped. For example, a shell element needs to be mapped to a fiber with orientation 75° after the forming simulation and if the number of parts defined by the user is assumed to be 6, then the range 0° to 180° for second fiber direction is split into 6 parts i.e. Part 1, Part 2 and so on, which represent the fiber orientations in the ranges $0^\circ - 30^\circ$, $31^\circ - 60^\circ$, $61^\circ - 90^\circ$ etc. respectively. In this case, the shell elements in the third range will be moved to Part 3 which represents the second layer fiber orientations in the range of $61^\circ - 90^\circ$.

2.3.3 Solution 3

Solution 3 is an approach that utilizes all fiber orientations from the forming simulation and brings the structural model closer to a model with the highest level of data integrity. It tries to map all fiber orientations to shell elements of the structural model mesh. This is achieved by assigning individual parts to each shell element from the structural model. Similar to the previous case, Solution 3 represents the first fiber directions with the `*ELEMENT_SHELL_BETA` keyword and the second with the `*PART_COMPOSITE` definition, see Figure 6.

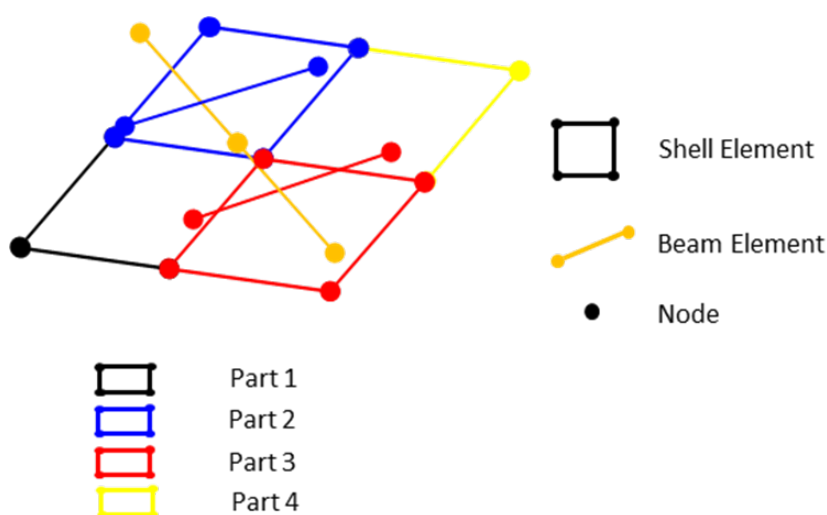


Fig.6: Composite modeling scheme according to "Solution 3" for structural model fiber direction mapping

Therefore, if the structural model consists of 1000 shell elements, the mapped structural model will also contain 1000 PIDs representing the second fiber directions from the organic sheet. Figure 6 demonstrates, on a unit cell level, a structural macro scale model with different `*PART_COMPOSITE` definitions. As Solution 3 takes all the fiber orientations into consideration, it represents the most realistic model which can be created using existing LS-DYNA® keywords for composite part modeling. LS-DYNA® provides limited capability to model anisotropic materials in a realistic way. Solution 3 defines a workaround approach to surpass this limitation. However Solution 3 also involves many part definitions which makes further preprocessing of the structural model a tedious activity. The three solutions described above, define the main objectives of this work. In the following sections, the possibility of achieving these objectives through existing algorithms / tools is discussed in detail.

3 Existing mapping algorithms / tools

The activity of mapping results from one phase of an FEA simulation to another to consider effects of parameters has been implemented in one way or another by many groups in the automotive and the aerospace industry in recent years. Mapping of forming results to structural models for isotropic material such as steel, aluminum, etc. is part of a standard product development cycle in certain organizations. However in the case of composites, the modeling techniques vary depending on the behavior of the material to be simulated. In some cases, the composite mesh models are implemented at the macro-level while in other cases at meso-level. Therefore it is extremely difficult to create a

single software tool which covers the entire spectrum of mapping possibilities. The following sections give an overview of some of the commercially available tools for mapping different parameters from one simulation stage to another and their capabilities. Nevertheless, in this work, a customized solution or so called “*BETA Mapper*” python based mapping tool was seen as the best way forward to provide the exact requirements necessary for the work considered here.

3.1 DIGIMAT

DIGIMAT [4] is a suite of software, developed and commercialized by e-Xstream engineering. DIGIMAT provides multi-scale material modeling technology that can be used in the development of optimal composite materials and parts. Among all other software from the DIGIMAT Platform, DIGIMAT-MAP provides the capability to transfer various parameters of a composite material from one simulation to another. MAP is a mapping tool that can be utilized to transfer fiber orientations, residual stresses, temperatures and weld lines between dissimilar injection molding and structural models. This tool also helps in the assessment of data integrity of the final mapped solution with respect to the original model. Moreover MAP is compatible with PAM-FORM and Simulayt for the transfer of data from forming simulation models. However, this tool supports the mapping of mesh models for an incomplete combination of element types i.e. 3D to 3D, 3D to shell or shell to shell. Therefore, it lacks in providing the required solution of mapping fibers modeled using beam elements to shell element as required in this work.

3.2 MPCCI

MPCCI [5] [6] is a tool developed by Fraunhofer SCAI, Germany and commonly known as MPCCI Metal Mapper. This tool provides a GUI along with different solver formats for mapping simulation data. The capabilities delivered by the mapper involves mapping the results from shell to shell, shell to volume and vice versa. However this tool does not provide complete functionalities for composite structures as well as mapping of beam elements to a structural shell mesh model.

3.3 Results mapping – Beta CAE [7]

Beta CAE provides mapping functionality with its preprocessor ANSA through a Results Mapping utility. The mapping is done by ANSA using FiberSIM in *.fsim2ansa and .hdf5 formats as well as Simulayt layup file, which acts as an interface between ANSA and the input simulation model for collecting composite part geometry and material orientation information. Although it is supported for various solver formats, this tool can only be implemented for shell or solid mesh models.

3.4 Envyo®

Envyo® [8] is a multi-purpose results mapping tool developed by DYNAmore GmbH designed to transfer information from a wide variety of process to structural analyses types. The tool has been developed within the framework of the EUREKA Cluster ITEA 3 project VMAP (led by Fraunhofer SCAI) whose aim is to define the standards for material data transfer in manufacturing virtual simulation. The project runs from September 2017 until September 2020. The software can transfer simulation data between different meshes having different discretization, element-types, material models etc. Furthermore, it is able to consider different material models available within LS-DYNA® and properly deal with their respective history variables and perform required homogenizations when necessary. The software is planned to be commercialized within 2019 and test versions can already be requested. The software works using a geometry matching point cloud based mapping technique and can deal with a long list of mapping scenarios including the beam – shell case required in this work.

3.5 Kinematic draping and mapping – ANSA/PAM-QUICKFORM

Among a large number of kinematic based draping and mapping tools the preprocessing functionality of ANSA from BETA CAE provides an additional utility for draping. This kinematic draping tool can assign shell elements with the required fiber orientations and writes this information into *ELEMENT_SHELL_BETA keywords for an LS-DYNA® input deck. In order to execute this utility the user has to provide a draping direction and a starting point for assigning the fiber orientations. A similar tool (PAM-QUICKFORM) is also provided by ESI-Group in their forming simulation software PAM-FORM. Here, the user also needs to choose a starting point for the kinematic draping and the resolution of the elements used in the procedure. The output is a mesh in draped or undraped form which can be used for mapping fiber orientations and optimizing blank geometries.

4 BETA Mapper – Script implementation

4.1 Basic process flow

Figure 7 explains the basic workflow of the developed script. The *BETA Mapper script* consists of two major modules which can be executed independently via command line by passing some input parameters as arguments. The first module *BeamtoShellPre* is a preprocessing module which processes input models taken from the user along with some other input parameters. These inputs include forming results i.e. .d3plot files, the structural model to which the fiber orientations are to be mapped and some parameters such as the number of CPUs to be used in the calculation, etc. Once all the inputs are given to the module and executed, the script invokes an LS-PrePost® session without the GUI in the background and reads in the input models one by one. These models are converted into keyword format by LS-PrePost® and are then passed on as inputs to the *BeamtoShellMap* module.

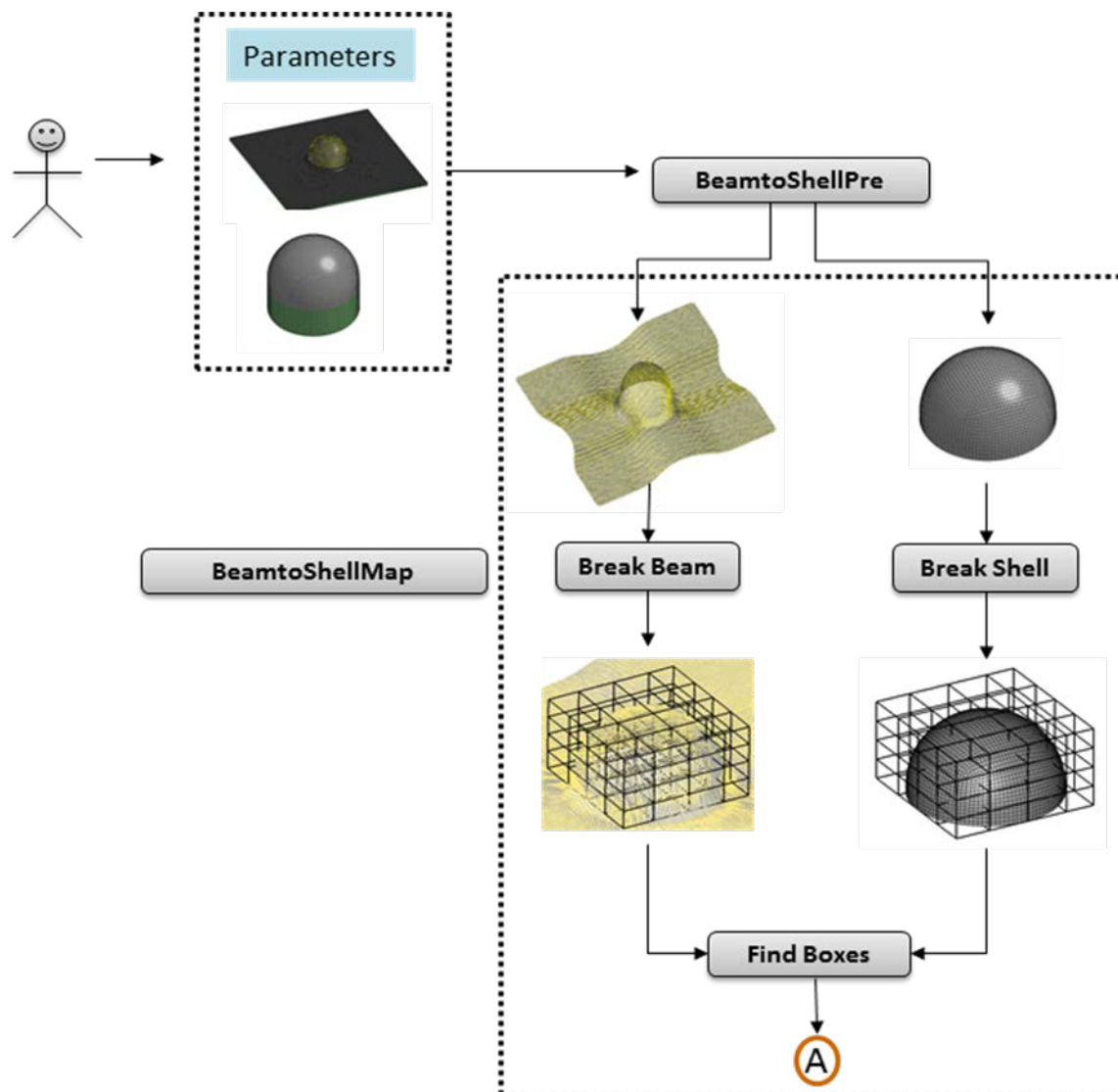


Fig.7: Basic workflow of the developed BETA Mapper script

BeamtoShellMap is a module that performs the actual mapping of fiber orientations. This module consists of several small submodules which perform different tasks. The first activity of *BeamtoShellMap* is to break up the data generated by LS-PrePost® from the forming results and structural model. This is achieved by invoking two different submodules; first the “Break Beam” submodule which discretizes the forming results while the second submodule called “Break Shell” discretizes the structural model into small cuboids. These submodules use only those PIDs, which are necessary for the mapping activity. After this volumetric discretization, the “Find Boxes” submodule

finds the pair of cuboids such that for each cuboid containing shell elements, there is a corresponding cuboid for beam elements from the forming results. These cuboids are then used as inputs for finding the pairs of shell and beam elements by the “Find Beam-Shell Pair” submodule (see Figure 8). The beam – shell pairs are written in different .csv files depending on the choice of solution desired. These .csv files containing pairs are then processed by the “BETA Calculate” submodule to find the final fiber orientations and are written into the BETA_Solution-X.key file which can be used as an include file for a further simulation input deck.

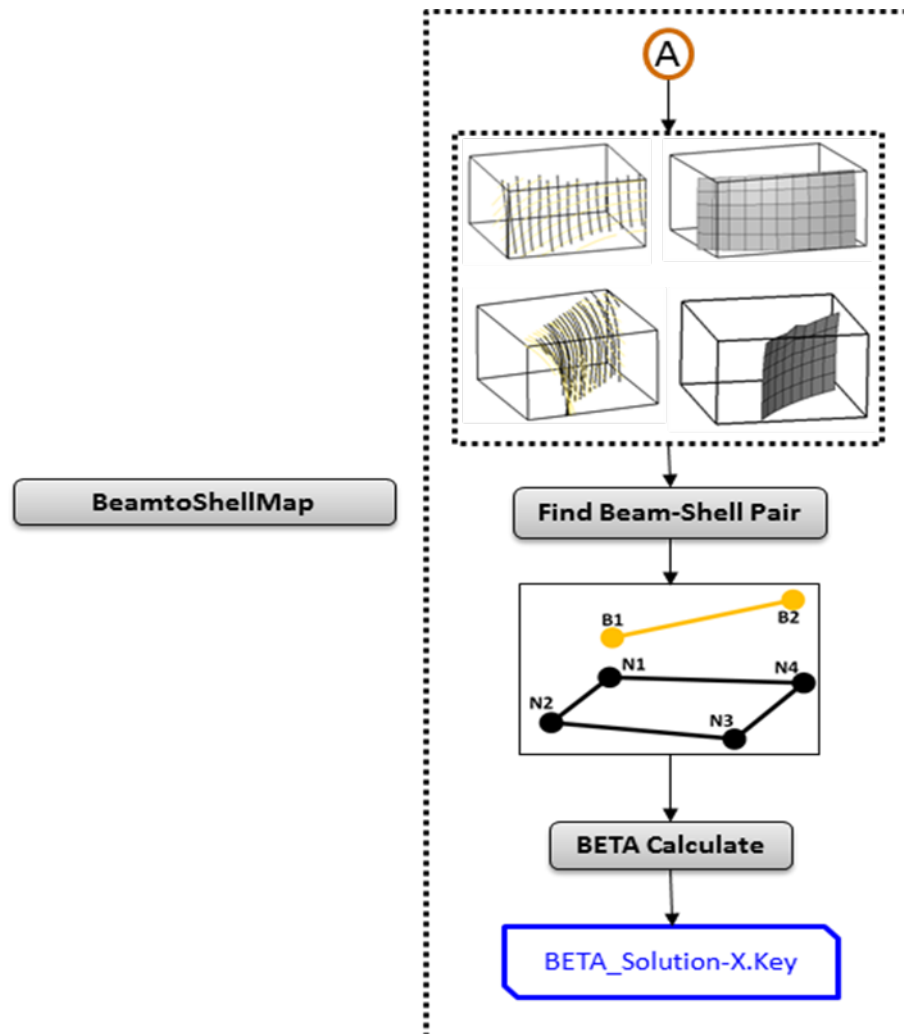


Fig.8: Basic workflow of the BETA Mapper script (continued)

4.2 Implementation of solution

Based on the user input for a type of solution, the BeamtoShellMap module has to write mapped fiber orientations into LS-DYNA® keywords. To achieve this goal, BeamtoShellMap writes these angles along with further keyword data into a .key file. This gives flexibility to the user to include a mapped solution .key file directly into an input deck. But to use the solution .key file as an include file, it is necessary to delete redundant element and nodal data from the previous structural model in the case of a desired Solution 1 while for Solutions 2 and 3 in addition to elements and nodal data, redundant parts should also be deleted. As discussed in Section 2.3, the solution types are implemented differently depending on the type of data the BeamtoShellMap module has to write into the BETA-Solution-X.key file. In the case of Solution 1, the script will write a BETA-Solution-1.key file, in which all elements are written under the keyword ***ELEMENT_SHELL_BETA** along with nodal data. In Solution 2, the module writes the number of parts defined by the user, the updated element cards with new PIDs and nodal data. Thus for Solution 2, the script writes all elements similar to Solution 1 and newly generated parts in the BETA-Solution-2.key file. Similarly for Solution 3, every element is represented

by an individual part as ***PART_COMPOSITE** and is written into BETA-Solution-3.key along with elemental and nodal data.

5 BETA Mapper – “Hemispherical Dome” case study

The previous section has given an overview about the implementation of various functions and modules to address different activities. In this section the functionality of all modules from the script is verified by executing them on a prototype component. The prototype component is a hemispherical dome geometry, which is modeled using the same modeling methodology explained in Section 2 for the forming simulation and structural model. The hemispherical dome is made of two parts consisting of beam elements representing initial fiber directions as 0° and 90° while the resin is represented by shell elements. This model is simulated considering non-isothermal forming conditions. The structural model has a standard mesh consisting of shell elements defined using the ***ELEMENT_SHELL** keyword while the composite part definition is created by using ***PART_COMPOSITE**.

5.1 Results

In this section, different solution types generated by executing the *BETA Mapper script* are compared. The purpose of this comparative study is to develop an understanding about the use of the different solutions for implementation of fiber orientations from non-isothermal forming simulations to structural simulations.

5.1.1 Solution 1

Figure 9 (left) illustrates the mapped structural model mesh for Solution 1. The figure also displays the fibers represented by beam elements from the forming simulation which are overlaid onto the structural model. This superimposition helps evaluate the correctness of the mapping through visualization. As it can be seen from the figure, the fibers are represented in the structural mesh as arrows and are oriented correctly along the first beam element direction from the forming results shown as yellow colored curves.

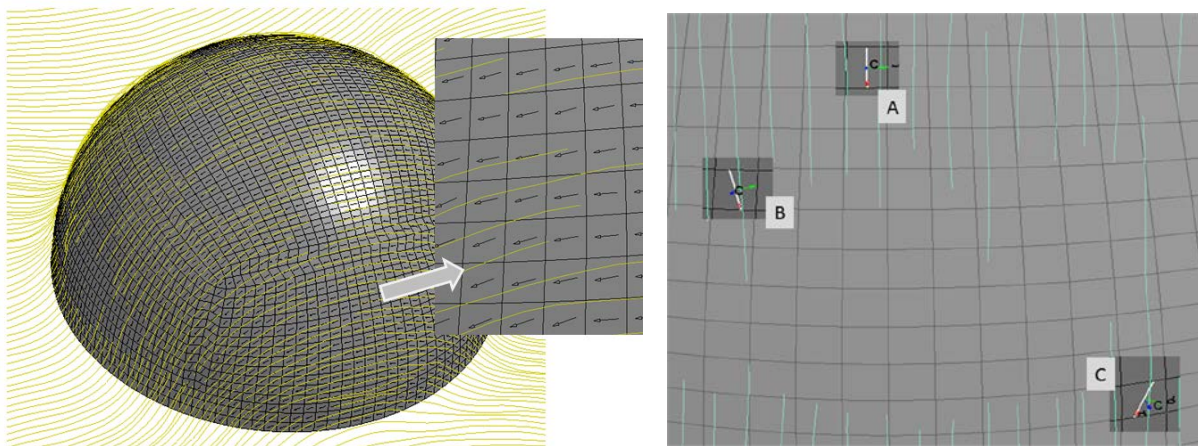


Fig.9: First (left) and second (right) direction fiber orientations resulting from Solution 1

These arrows demonstrate the first material directions of the shell elements as visualized in LS-PrePost®. As Solution 1 maps only one fiber direction, the second fiber directions are set to the default value of 90° representing an orthotropic material. The second directions are demonstrated in Figure 9 (right). Here the beam elements in cyan represent second fiber directions from the forming simulation while the white lines on the shell elements represent the actual fiber orientations assigned to the shell elements. These fiber orientations in the shell elements are dependent on the first fiber orientations mapped from forming results and are oriented at 90° to them. In Figure 10, it can be seen from three different locations A, B and C, that the second fiber direction in the sampled shell element is along the beam element direction at location A but are offset from the actual fiber orientations at locations B and C. These misalignments in the second fiber directions have emerged due to local shearing of the organic sheet. Therefore Solution 1 should be implemented only in geometries where these in-plane shearing effects are negligible (e.g. developable surface geometries).

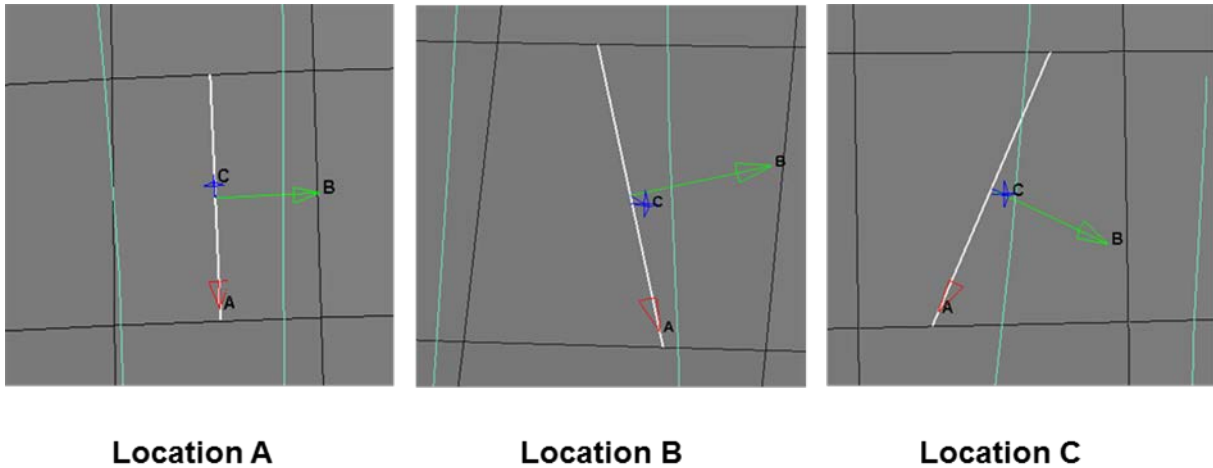


Fig.10: Second direction fiber orientations in Solution 1 as per different locations in the mapped dome model

To illustrate the above scenario, the mapping is carried out on the “Crash Muffin” model geometry [9]. The *Crash Muffin* model geometry has been designed on the basis of a “developable geometry” i.e. the entire surface of the *Crash Muffin* can be unfolded flat and does not exhibit any in-plane shearing effects. In this case, initial fiber directions which are oriented at 90° offset, maintain their relative orientations and the resulting part theoretically does not involve any in-plane shearing effect. Therefore a structural model can be mapped only for the first direction and the second direction can be assigned the default orientation of 90° offset with respect to the first direction. Figure 11 demonstrates the mapped *Crash Muffin* model. Here the beam elements from forming results are superimposed onto the *Crash Muffin* structural model mesh. The figure shows the first fiber directions indicated by arrows in the shell elements mapped according to fibers represented by beam elements in yellow while in Figure 12 the second direction fibers are represented by white lines along the beam elements represented by cyan colored curves. Here negligible shearing, results in better fiber orientations in the mapped mesh model even for the second fiber direction.

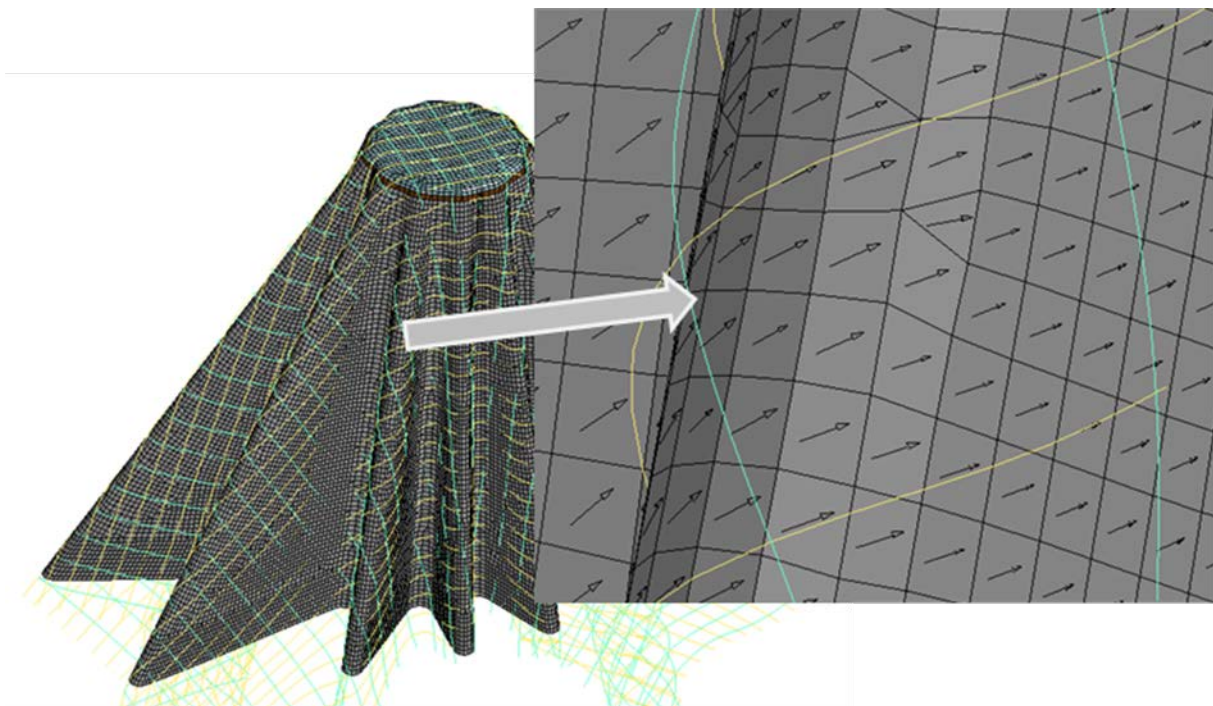


Fig.11: First direction fiber orientations in Solution 1 for the crash muffin geometry

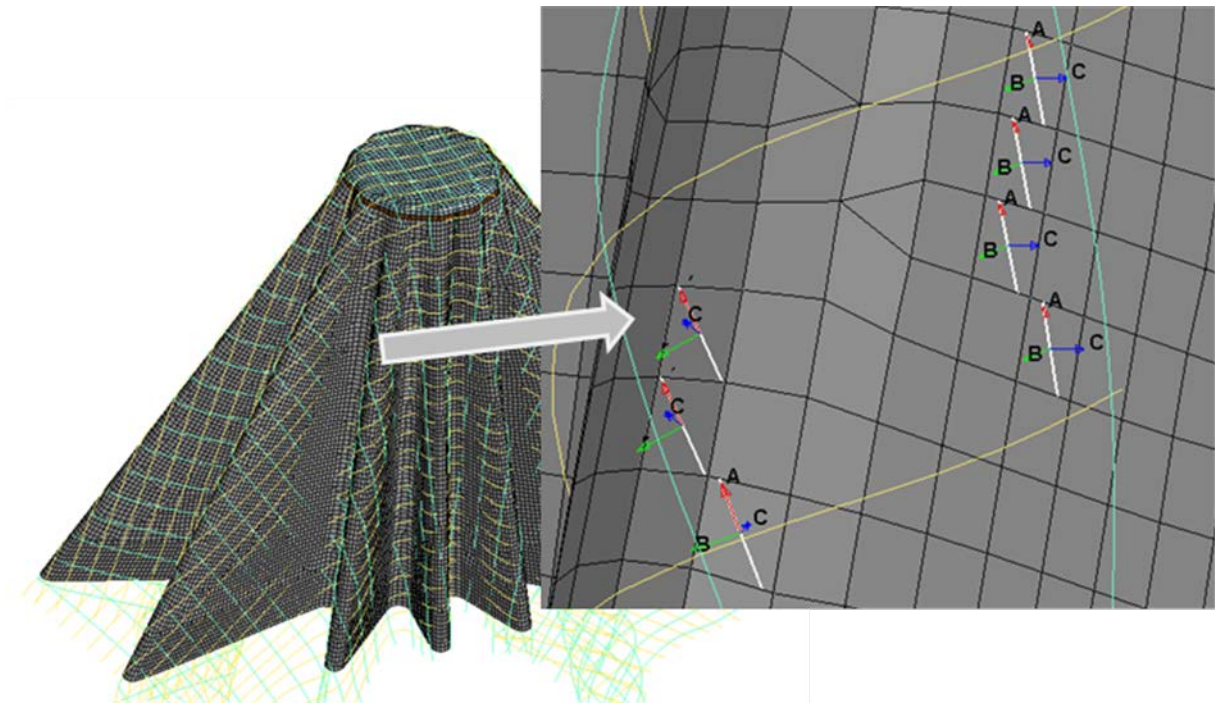


Fig. 12: Second direction fiber orientations in Solution 1 for the crash muffin geometry

5.1.2 Solution 2

Solution 2 implements both fiber directions by generating new PIDs defined by the user. As discussed in Section 2.3.2, the elements are segregated into new parts according to their second fiber orientation defined within a particular range. Figure 13 illustrates the shell elements with new PIDs based on the second direction fiber orientations. Here the first fiber orientations remain identical to the ones from Solution 1 but the shell elements are assigned to new parts.

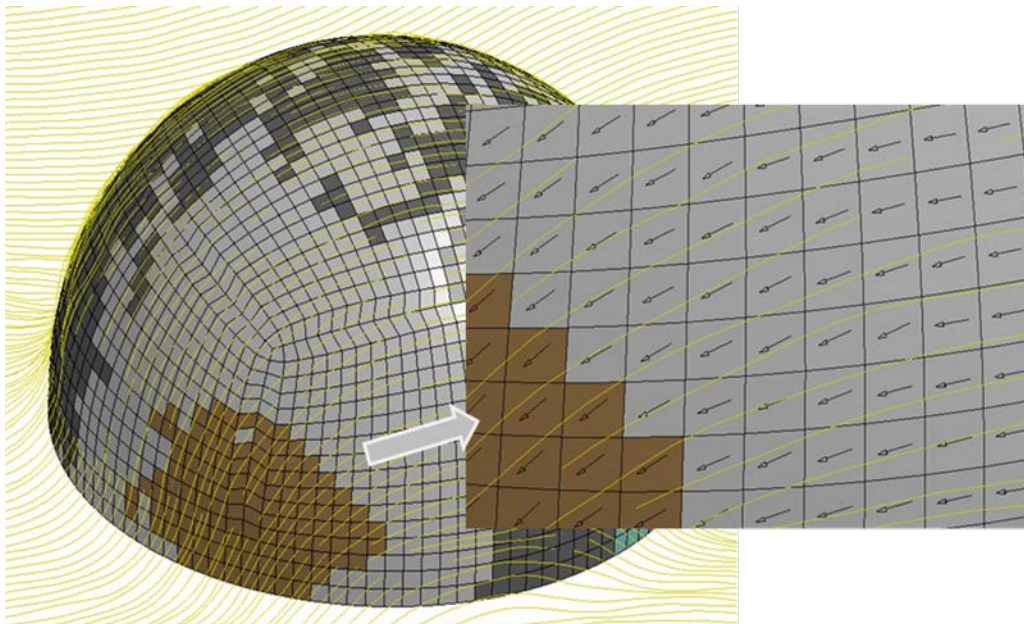


Fig. 13: First direction fiber orientations resulting from Solution 2

As defined in the LS-DYNA® ***PART_COMPOSITE** keyword, the first direction fiber orientations β_1 are taken as the offset angles from the element directions and are written in the ***ELEMENT_SHELL_BETA** keyword while β_2 angles are the offset angles from β_1 .

Eventually the parts with different values of β_2 represent different regions with shearing effects. It is important to note that β_2 represents the offset angle between two fiber directions from different layers of ***PART_COMPOSITE**. Figure 14 illustrates a shell element S1 with its element direction defined by nodes N1 and N2, the first fiber direction according to beam element B1 is used to calculate β_1 and second fiber direction according to element B2 is used to calculate β_2 . After determining $\beta_{2absolute}$ from the element co-ordinate system, β_2 is evaluated by considering the β_1 angle made by the first fiber direction B1 with respect to the element co-ordinate system. In LS-PrePost® the beam directions are considered along the vector from beam node 1 to node 2. In order to calculate the orientations β_1 and $\beta_{2absolute}$ these directions are used as reference and based on their relative positions, the position vector defining β_2 with respect to shell node N1 is derived.

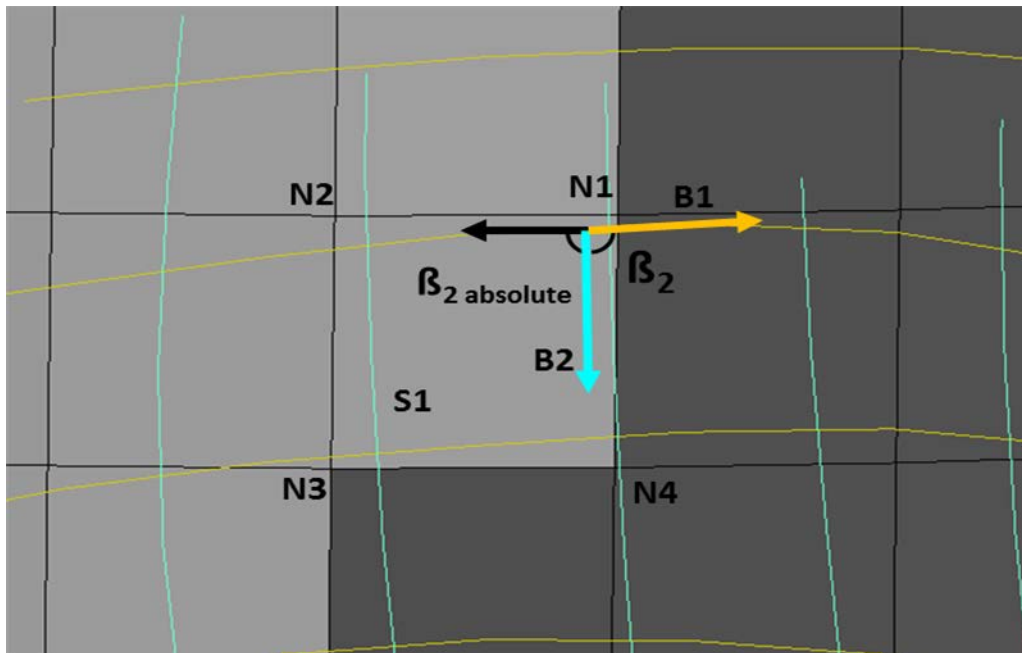


Fig.14: β_2 calculation for Solution Types 2 and 3

Therefore in some cases the vector defining β_2 lies ahead of β_1 i.e. β_2 is positive while in other cases when β_2 is behind β_1 then it is taken as negative. This convention is based on the shell normal or top surface definition derived from counterclockwise node numbering in LS-DYNA® [16]. Nonetheless the sign convention for β_2 is applicable only for Solution 3 and for Solution 2 β_2 is defined in ranges between 0° and 180° . Different boundary conditions used for identifying the PID for any shell element are dependent on its mapped β_2 value and are explained below.

```

 $\beta_2 = \beta_{2absolute} - \beta_1$ 
if  $0 < \beta_2 < 180$ , then  $\beta_2$  is +ve
if  $180 < \beta_2 < 360$ , then  $\beta_2 = \beta_2 - 180$ 
if  $-180 < \beta_2 < 0$ , then  $\beta_2 = \beta_2 + 180$ 
if  $-360 < \beta_2 < -180$ , then  $\beta_2 = \beta_2 + 360$ 

```

From above discussion it is clear that the second fiber orientation of the mapped shell elements depends on the β_2 values assigned to the corresponding ***PART_COMPOSITE** keyword. As new parts are assigned different β_2 values depending on the ranges in which the parts are defined, some parts have appropriate values for β_2 representing the actual fiber orientation from the forming simulation. However some parts might have β_2 values that do not match with actual fiber orientations. This approximation has emerged due to the division of the range $0^\circ - 180^\circ$ equally based on the number of parts given by the user. In order to minimize this approximation, the user can choose to generate a larger number of parts to have more flexibility to assign more β_2 values or assign the parts minimum, average or a maximum limit of the range in which β_2 is defined. For example, if elements which have β_2 values in the range $90^\circ - 120^\circ$ will be segregated into PID 100. This newly generated PID 100 can be assigned a minimum of the range $90^\circ - 120^\circ$ i.e. 90° or an average value 105° etc. After assigning

different β_2 values i.e. minimum, average or maximum limit of ranges to ***PART_COMPOSITE**, the fiber orientations are modified accordingly. Therefore, after mapping, the user can decide on the appropriate values of β_2 to be assigned for the newly generated parts in order to represent different regions with shearing effects. These shearing effects are illustrated in Figure 15. Here, the *BETA Mapper script* was executed on the dome geometry to produce Solution 2 with different parts and β_2 values.

In the following solutions, parts representing some ranges were observed to be empty, as no single shell element had a β_2 value which could be represented by those parts. For example, when the number of parts generated is 6, PID 1000 and 1005 were observed to be empty. These parts represent ranges of $0^\circ - 30^\circ$ and $150^\circ - 180^\circ$ and there was no single shell element with β_2 in the range $0^\circ - 30^\circ$ or $150^\circ - 180^\circ$.

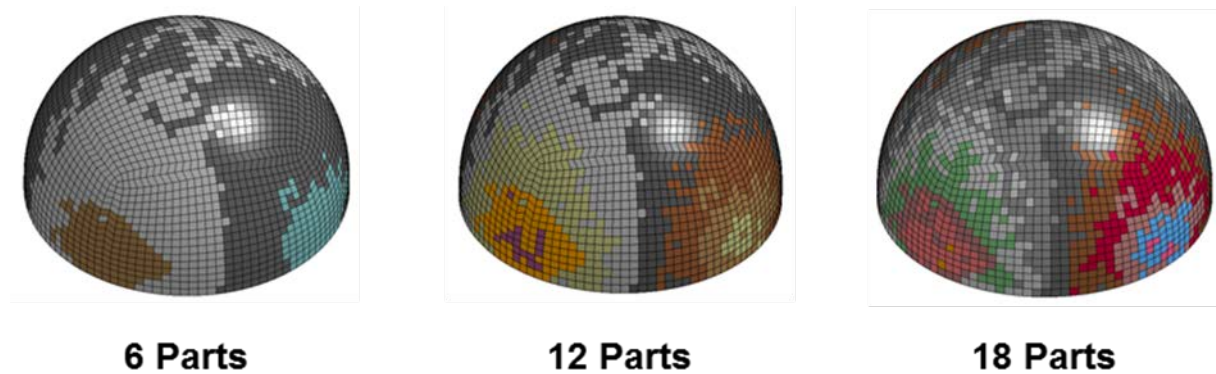


Fig. 15: Solution 2 generated with different number of user defined parts

As discussed above, when a newly generated part is assigned, different values of β_2 within the range used to define that part arise, then the second fiber directions defined for the shell elements modify their orientations. Figure 16 demonstrates the influence of assigning different β_2 values as per the ranges. Here the arrows in black represent first fiber directions in the mapped shell elements according to actual fibers shown as yellow curves. The second directions of shell elements are represented by white lines while the actual fibers from the forming results are represented by cyan curves. For Solution 2 with 6 parts, when β_2 is assigned the maximum or minimum limits of the range, the shell element fibers are oriented at a certain offset angle to the actual fibers while average values of β_2 give sufficient mapping in certain parts. Nonetheless, this assumption of assigning average β_2 values cannot be generalized and the user has to make a judgment based on number of parts/elements that are not correctly mapped according to actual fiber orientations from the forming results.

When the numbers of parts to be generated are increased, the ranges for which new parts are generated become narrow and the error in β_2 values is reduced further. This can be illustrated in Figure 16 when the numbers of parts are increased to 12 parts or 18 parts; the deviation of β_2 from the actual fiber directions is reduced. With the number of parts increased to 18, the deviation of β_2 from actual fibers is observed to be the minimum among all three scenarios when different values of β_2 according to the range limits are assigned. Eventually, the selection of different β_2 values and number parts depend on the type of geometry being mapped. If the resulting geometry from the forming simulation has more regions with shearing effects, the user has the flexibility to increase the number of parts given to the script accordingly and then modify the β_2 values in ***PART_COMPOSITE** keywords in the .key files.

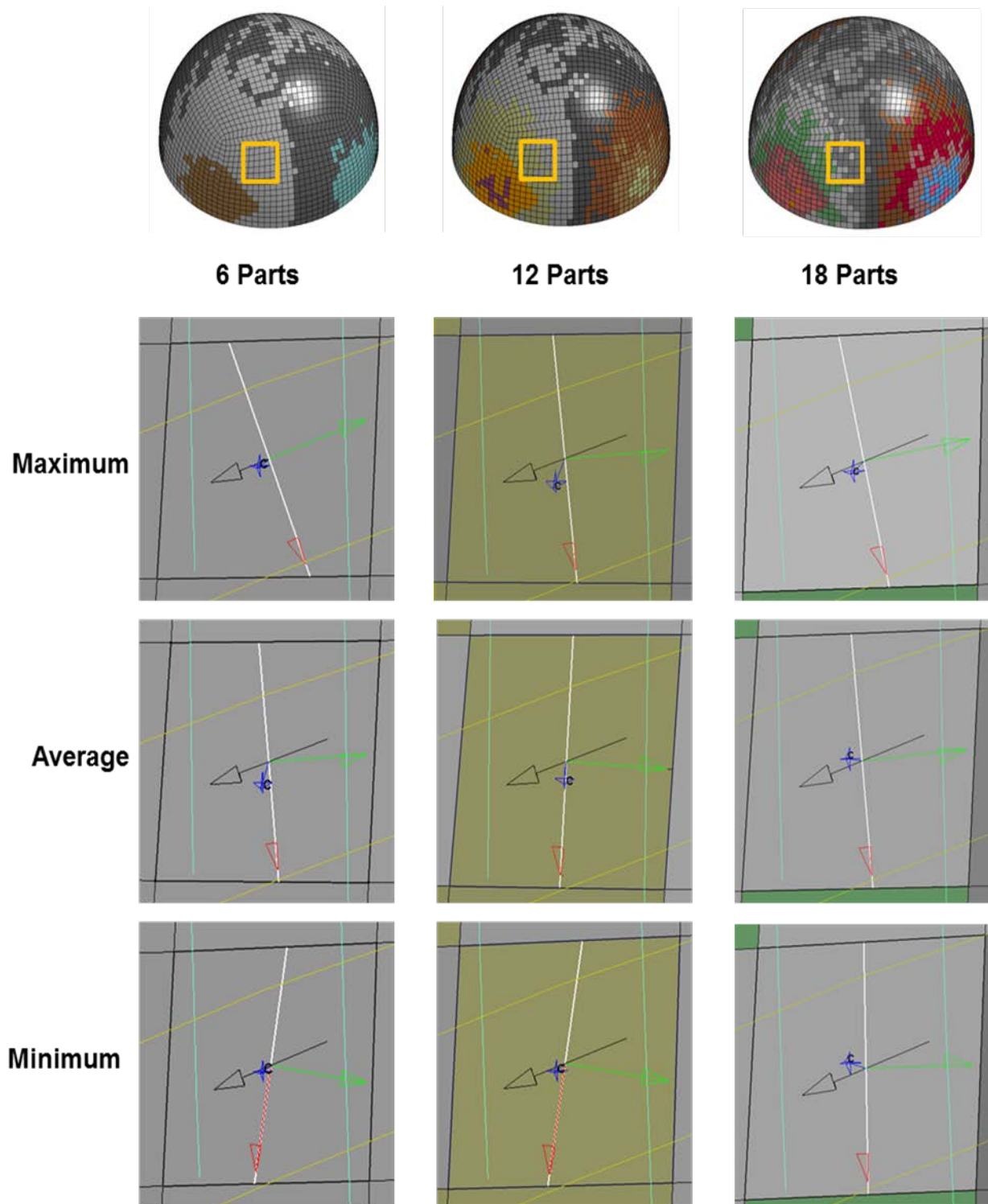


Fig. 16: Solution 2 – Influence of number of user defined parts and β_2 values

5.1.3 Solution 3

Among all three solutions, Solution 3 generates the structural mesh model with maximum data integrity with respect to the forming results. In the case of Solution 3, all elements are represented by individual PIDs which have a β_2 orientation derived from each pairing fiber from the forming results. As shown in Figure 17, all elements are assigned to individual PIDs. As discussed previously, this methodology makes further preprocessing of the generated structural model quite cumbersome.

To implement both fiber directions, using a single ***ELEMENT_SHELL_BETA** keyword for each shell element is not possible. Hence Solution 3 suggests a workaround technique to consider both fiber orientations using the existing capabilities available in LS-DYNA®. Figure 18 shows the second fiber directions where each element has individual ***PART_COMPOSITE** properties assigned and hence an individual β_2 value. Here every single shell element is mapped according to the beam elements and therefore Solution 3 has the highest level of data integrity among all solutions showing accurate mapping for both the first and second fiber directions.

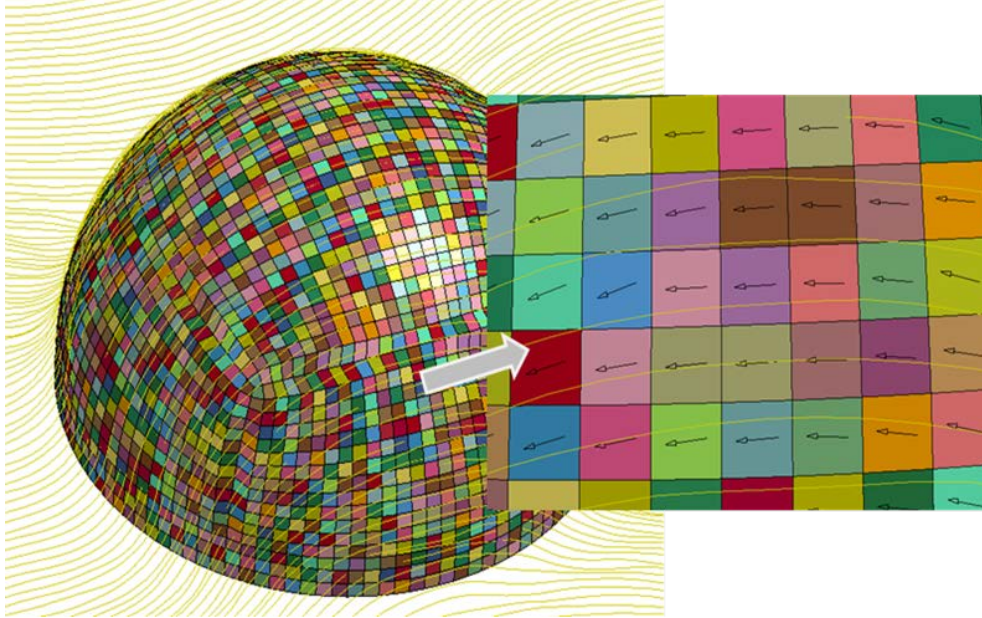


Fig. 17: First direction fiber orientations resulting from Solution 3

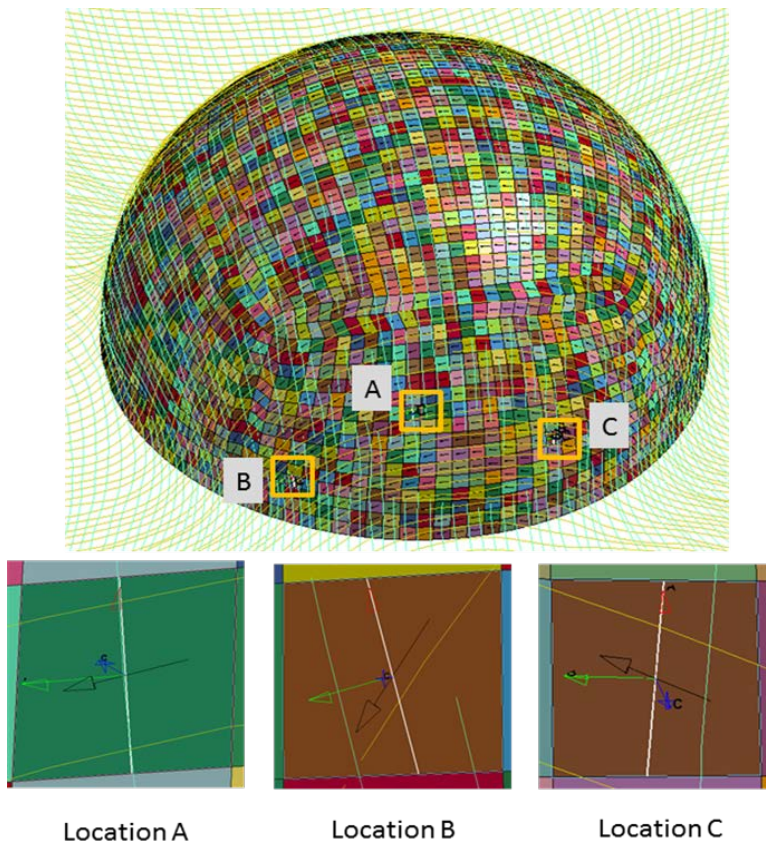


Fig. 18: Second direction fiber orientations resulting from Solution 3

6 Conclusions

6.1 Summary

As per the problem definition, the purpose of this work was to design a workflow through which the fiber orientations could be transferred from one simulation to another. Section 3 gave an overview of the different commercial tools available for simulation results mapping. Existing tools deliver some solutions for the fiber mapping but they cannot exactly provide the required functionality for the modeling methodology implemented for non-isothermal forming simulation [3]. Some tools only implement shell to shell, shell to solid or solid to solid element types for mapping. Furthermore some of these tools can only be used for particular solver formats. Some kinematic draping utilities which also provide mapping functionality are not accurate enough to match the results obtained by mapping fiber orientations extracted from a forming simulation. Kinematic draping considers only part geometry and draping direction while in the real scenario the part behavior during the actual forming process is highly governed by the material properties, thermal behavior etc. effects which are not included when executing a kinematic draping analysis.

With development of a customized script, the fiber orientation represented by the beam elements in a forming simulation can be now mapped to shell elements. Here with non-isothermal forming simulation, material properties, thermal behavior and other parameters affecting the fiber orientations are all taken into consideration. The difference in results can be seen by comparing kinematic draping results with the mapping results obtained by executing the *BETA Mapper script*.

The *BETA Mapper script* provides user flexibility to map fiber directions onto the structural model based on the shearing behavior observed in the forming simulation. When the resulting geometry after the forming simulation exhibits more regions with shearing, the solution to be implemented for mapping can be adapted as per the resulting geometry. To explain this in more detail, if a resulting part from forming process does not exhibit any regions with shearing effects, then the mapping can be carried out solely by employing Solution 1. As discussed previously, Solution 1 represents an orthotropic material behavior and is the simplest of all solutions whereby only one fiber direction is used for mapping. With less shearing effects, only the first direction fibers are necessary for mapping and Solution 1 is adequate in this case.

In the case of Solution 2, the script implements both fiber directions. Here, based on the number of parts given by the user, new parts are generated and assigned to the shell elements. After mapping, the parts can be assigned minimum, average or maximum angle values of the ranges that are defined when generating the parts. Similar to the previous example, the judgment of selecting the number of parts and the β_2 values should be based on the resulting geometry and shearing effects observed after the forming simulation. The *BETA Mapper script* divides the range 0° - 180° equally which is dependent on the number of parts given by user and cannot be altered. Therefore, Solution 2 involves some approximation regarding the second fiber directions and the angles assigned to newly generated parts should be chosen appropriately to take shearing effects into consideration. Moreover, the user has the flexibility to choose a greater number of parts for mapping to minimize this approximation and then assign angles independently to new parts. But it is important to note that with more parts, Solution 2 moves closer to Solution 3 which needs more time for mapping and probably for simulation as well. Therefore selection of the number of parts should be carried out carefully to balance the model approximation, time needed for mapping as well as for carrying out the structural simulation.

Existing keywords in LS-DYNA® allow limited flexibility for the implementation of different fiber orientations in one keyword. Thus Solution 3 gives a workaround method to implement two fiber orientations using the ***ELEMENT_SHELL_BETA** und ***PART_COMPOSITE** keywords. Even though Solution 3 focuses on maximum data integrity from a forming simulation, it is important to consider the time needed to generate the mapping solution as well as for simulation. Nonetheless, with higher computational power the time required for both mapping and simulation can be reduced.

The existing *BETA Mapper script* is developed for the LS-DYNA® solver and can be extended for other solvers used for FEM analysis. In addition, the workflow implemented for the development of the *BETA Mapper script* is based on sequential execution and can be enhanced further. These topics related to future scope are discussed in the following section

6.2 Future scope

In the near future, depending on the element types, solvers or for other requirements the *BETA Mapper script* can be upgraded further. In this section some of the possibilities for future developments are discussed.

6.2.1 Mapping of different element types

The existing *BeamtoShellMap* module can be modified to incorporate future needs. Currently this module is used to implement mapping of fiber orientations from beam elements to the shell elements. By modifying certain modules and functions the same work flow can be implemented to map fiber orientation from beam elements to solid elements (hexa, tetra, penta or wedge). To achieve this, the "Shell Function" module which processes all data related to shell elements needs to be updated. The "Shell Function" module consists of functions necessary to extract the element, nodal and centroid data used to perform different activities in other modules. Most of the modules such as "Break Shell", "Find Beam-Shell Pair" and "Find Beam-Shell Pair Failure" process centroid data. Hence once the functions that carry the centroid data are modified these modules should be able to do their tasks in a predefined way. Furthermore the module "BETA Calculate" which finds actual orientations has to be modified, as it processes element and nodal data. When mapping is desired to be carried out on a Hexagonal element, it is obvious that the element cards and nodal information for that type of element are different from shell elements. Hence to implement a new type of element it becomes necessary to change this module.

6.2.2 Solver independent mapping

The actual mapping workflow has been designed for the LS-DYNA® solver. The basic modules "Shell Function" and "Beam Function" extract data from input files based on LS-DYNA® keyword information. In order to make the mapping algorithm work independently of the solver, the "Shell Function" and "Beam Function" modules have to be updated to recognize the solver of the input models. Further different keywords used in various solver formats can be stored in a keyword library. Once the solver is recognized, the keyword library associated with a particular solver can be used to extract the required element, nodal and centroid data for further execution of subsequent modules. In this script the "BETA Calculate" module writes all output fiber orientations, element and part data in LS-DYNA® format. Once the solver is changed, modifying all functions processing this information from the "BETA Calculate" module would be essential.

6.2.3 Workflow optimization

The existing workflow implemented in the script has been explained in Section 4.1. This workflow is based on a sequential execution of individual modules. As the modules are executed sequentially, every next module has to wait until the previous module has finished its execution. Hence to optimize the existing workflow in order to reduce the time of execution, modules can be executed simultaneously using operating system protocols of write, read and wait for different processes running parallel on several processors. Furthermore, every module uses some files as inputs and writes .csv files which are used as input for subsequent modules. This involves a lot of input-output processing and can be optimized by using system memory to store all necessary data.

7 Literature

- [1] Reddy, J.N.: Mechanics of Laminated Composite Plates Theory and Analysis – ISBN 0-8493-3101-3.
- [2] Duhovic, M.; Mitschang, P.; Bhattacharyya, D.: Modeling approach for the prediction of stitch influence of woven fabric draping. Composites. Part A, Applied Science and Manufacturing. Volume 42A, Issue 8, 2011, pp.968-978.
- [3] Schommer, D.; Duhovic, M.; Hausmann, J.: Modeling non-isothermal thermoforming of fabric-reinforced thermoplastic composites, 10th European LS-DYNA Conference 2015, Würzburg, Germany.
- [4] DIGIMAT MAP, <http://www.e-xstream.com/products/tools/DIGIMAT-map>
- [5] MPCCI, <http://www.mpcci.de/fileadmin/mpcci/download/MpCCI-SCAIMapper/v4.3/SCAIMapper-430.pdf>
- [6] MPCCI, <http://www.scapos.com/products/cae-tools/mapper.html>
- [7] Results Mapping, https://www.beta-cae.com/pdf/results_mapping.pdf

- [8] Workshop Envyo®: Mapping and data management along the simulation process chain
<https://www.dynamore.de/de/download/papers/2016-ls-dyna-forum/Papers%202016/mittwoch-12.10.16/workshops/mapping-tool-envyo>
- [9] Crash-Muffin: <https://www.kunststoffe.de/produkte/uebersicht/beitrag/thermoplastische-crash-muffins-crashabsorber-aus-faser-kunststoff-verbunden-fkv-945632.html>