

ESA STUDY CONTRACT REPORT –

**Deliverable D 5.4
Virtual simulation tool documentation**

<p>ESA Contract No: 400114452/15/ NL/NDe</p>	<p>SUBJECT: Final, D 5.4 Virtual simulation tool documentation</p>	<p>CONTRACTOR: Riga Technical University, Institute of Materials and Structures</p>
<p>* ESA CR()No:</p>	<p>No. of Volumes:1 This is Volume No:1</p>	<p>CONTRACTOR'S REFERENCE: 62518 (bidder code)</p>
<p>ABSTRACT:</p> <p>The current report presents documentation for all developed software during “Development and validation of methodology for assessment of damage resistance properties of sandwich structures for European space sector” project. Including Automated Surface Image Analysis (ASIA) toolbox for MATLAB, two version of Analytical Indentation Model (AIM) function integrated in MATLAB, Surface Inspection Viewer (SIV), Compression After Impact Viewer (CAIV), Modal assurance criteria MAC script also integrated in MATLAB, as well as ColorThick for digitalising of thickness distribution from NDE, and finally ANSYS Model Configuration Tool (AMCT) for pre-processing of numerical simulation example.</p>		
<p>The work described in this report was done under ESA Contract. Responsibility for the contents resides in the author or organisation that prepared it.</p>		
<p>Names of authors: Gints Jēkabsons; Jānis Andersons; Eduards Skuķis.</p>		
<p>** NAME OF ESA STUDY MANAGER: DIV: DIRECTORATE:</p>	<p>** ESA BUDGET HEADING:</p>	

Table of Contents

1	Introduction	3
2	Automated Surface Image Analysis (ASIA) toolbox for MATLAB.....	3
2.1	Introduction.....	3
2.2	Usage.....	3
2.3	Implementation details	8
2.3.1	General structure	8
2.3.2	Main functions.....	9
3	Analytical Indentation Model (AIM) function for MATLAB.....	17
3.1	Introduction.....	17
3.2	Computations done by the function AIM v1	17
3.3	Computations done by the function AIM v2: differences from AIM v1	19
3.4	Usage.....	21
4	Surface Inspection Viewer (SIV)	23
4.1	Introduction.....	23
4.2	User interface.....	23
5	Compression After Impact Viewer (CAIV)	25
5.1	Introduction.....	25
5.2	User interface.....	26
6	MAC script for MATLAB	27
6.1	Introduction.....	27
6.2	Usage.....	28
7	ColorThick.....	28
7.1	Introduction.....	28
7.2	User interface.....	29
8	ANSYS Model Configuration Tool (AMCT).....	30
	References.....	31

1 Introduction

The present report presents software developed during the “Development and validation of methodology for assessment of damage resistance properties of sandwich structures for European space sector” project. This includes:

1. Automated Surface Image Analysis (ASIA) toolbox for MATLAB
2. Analytical Indentation Model (AIM) function for MATLAB
3. Surface Inspection Viewer (SIV)
4. Compression After Impact Viewer (CAIV)
5. Modal assurance criteria MAC script for MATLAB
6. ColorThick for digitalising of thickness distribution from NDE
7. ANSYS Model Configuration Tool (AMCT)

All developed software is provided on the project disc. Where applicable, the software is also available for immediate use on the project website (see individual sections below for more details).

2 Automated Surface Image Analysis (ASIA) toolbox for MATLAB

2.1 Introduction

The ASIA toolbox is intended for detection, analysis, and reporting of the extent of damage caused by the low-velocity drop weight impact event or quasi-static indentation on honeycomb sandwich panels. The task is performed using xyz point cloud digital data from a laser-based xyz scan of the specimen surface. The software uses automated algorithms to detect and measure geometry of the damaged area – outline, depth, diameters, area, volume. It also returns surface plots and cross-section plots of the area.

2.2 Usage

The ASIA toolbox is developed for MATLAB version 2015b [5]. On the project disc all the ASIA toolbox files can be found in the directory " ASIA toolbox".

The software is intended for use with xyz point cloud data from a laser-based xyz scanner of a 460 x 210 mm panel cut to six individual samples 150 x 100 mm. While the indentation is done on each sample separately, the scanning is performed for all six samples put back together. The scan is done before and after indentation. Scanning before indentation is optional but it can make the necessary further computations more accurate. Additionally the software can also make use of data from FEM simulations and results from analytical computations (by using the functionality of the AIM function – see Section 3 for details) for the same samples to compare real physical experiments, FEM simulation results, and analytical results.

While the toolbox contains many functions there is available a convenience script that is already made to call all the necessary functions in the right order. This script file is called `processpanels.m`. It also already contains all the separate

configurations (sets of preset parameters) used for all panels considered during the project.

The script `processpanels.m` contains the following main parameters:

- `filePathInput` (string) – Path to the directory for input files – xyz point cloud data before indentation, xyz point cloud data after indentation, and the corresponding xyz point cloud data from a FEM model.
- `filePathOutput` (string) – Path to the directory of all output files – generated plots and cross-section coordinate files.
- `flipXAfter` (boolean) – Whether to flip the loaded "after" data on the X coordinate. Default value = `false`.
- `isOneColumnBefore` (boolean) – Whether the "before" data has only one column. Default value = `false`.
- `isOneColumnAfter` (boolean) – Whether the "after" data has only one column. Default value = `false`.
- `isDataBeforeUsed` (3 by 2 matrix of boolean) – Each element indicates whether the "before" data of the corresponding specimen must actually be processed. Default value = 3 by 2 matrix of `true`.
- `isDataAfterUsed` (3 by 2 matrix of boolean) – Each element indicates whether the "after" data of the corresponding specimen must actually be processed. Default value = 3 by 2 matrix of `true`.
- `maskIgnore` (optional 3 by 2 cell array with elements containing masks) – Defines masks of areas of a specimen to be ignored while determining area of damage. Default value = a 3 by 2 cell array of empty elements indicating that no masks should be employed.
- `name` (3 by 2 cell array of string) – Names for specimens. Default value = a 3 by 2 cell array of empty elements.
- `step` (double type) – The loaded irregularly scattered measurement data of arbitrary resolution is transformed into a regular grid. This parameter specifies the step size of the grid. Default value = 0.1.
- `smoothingParameter` (double) – Smoothing parameter of the smoothing procedure (see next subsection for details). The higher the value the more smoothed will be the surface of the specimen – undersmoothing and oversmoothing both cause inaccuracies in results. The parameter is independent from `step` but dependent on specimen physical size. Default value = 35 (this is a good general value chosen empirically for the panels considered in the project).
- `maxFloodDiameter` (double) – Maximum diameter around the lowest point of the indentation inside which the flood fill algorithm must find the geometry of the damage (see next subsection for details). Default value = 35.
- `ignoredDiameter` (double) – Approximate diameter around the lowest point of the indentation that must be ignored when estimating zero level of the Z coordinate of the surface. This value should always be a little bigger than the actual diameter of the indentation. Default value = 25;
- `panelParameters` (a 3 by 2 cell array of parameters for analytical calculations using the AIM function – see Section 3 for details) – The

containing parameters are the following: ply angles, core crushing stress, diameter of indenter tip, and indentation force. This is required only if analytical calculations are to be made. Default value = `{0 0; 0 0; 0 0}`.

- `isFEThresholdZ` (boolean) – Whether to determine area and volume of the indentation in the FEM data by thresholding surface depth (`true`) or thresholding surface local angle (`false`). Default value = `true`.
- `filesIndex` (integer) – Index of a prepared configuration (a set of preset parameters) for a specific panel. In the script file additional configurations can be freely added.

When the script is started, it loads all the given data, does the necessary pre-processing steps and analyzes the data (note that it may take a while) to report the following results for each specimen and for each method (analysis of physical experiment data, analysis of FEM method data, and analytics):

- Indentation depth (mm);
- Indentation diameters (mm) – diameters at four angles, maximum diameter, and average diameter – this is in accordance with the standard D7136/D7136M [3] (see Figure 2.1.);
- Indentation area (mm²);
- Indentation volume (mm³);
- Surface plot of indentation area that showing the outline of the damage (see Figure 2.2. for an example);
- Four plots of cross-sections through the four diameters of the indentation (see Figure 2.3. for an example);
- Indentation cross-section shape files – the files are simple tab-separated text files containing two columns: 1) position in respect to the lowest point of the indentation; 2) surface depth at that position.

See Table 2.1. for an example of the estimated measurements printed by the program.

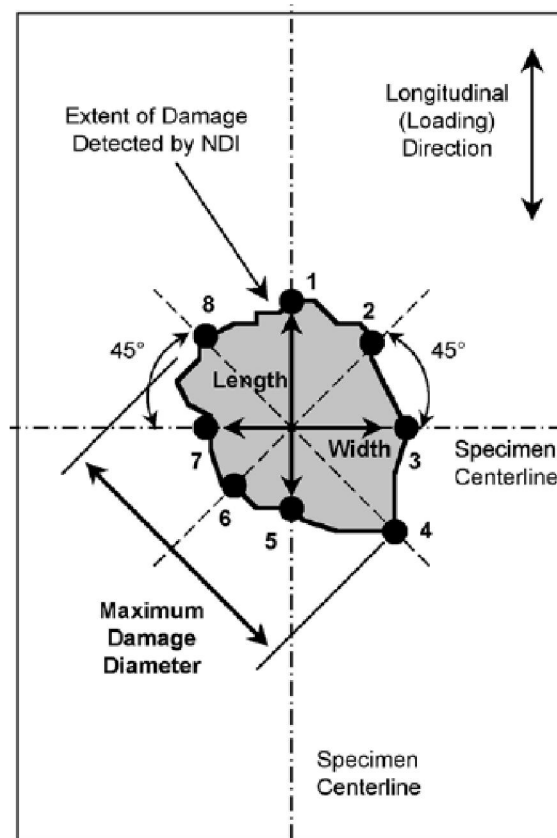


Figure 2.1. Measurement of extend of damage according to the standard D7136/D7136M [3]

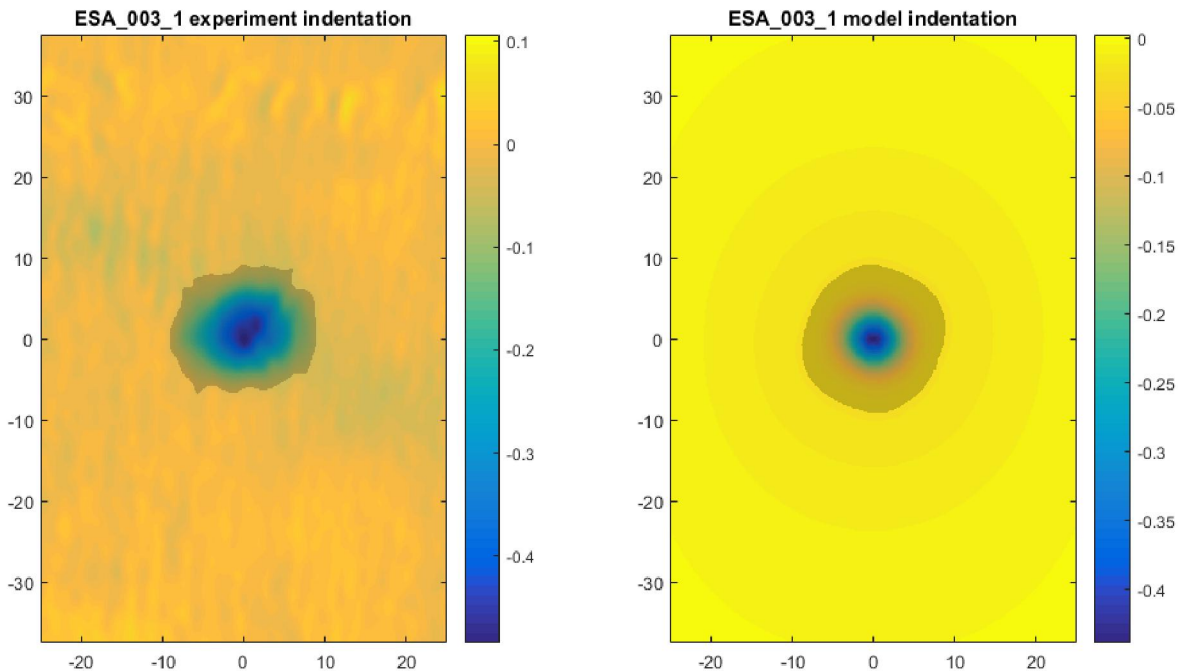


Figure 2.2. Example of surface plots (all measurements in mm) of the damaged area of a specimen. The left surface plot is from a physical experiment, the right surface plot is from a FEM model. The views are zoomed-in. The lowest point of the indentation is at the centre of the plot.

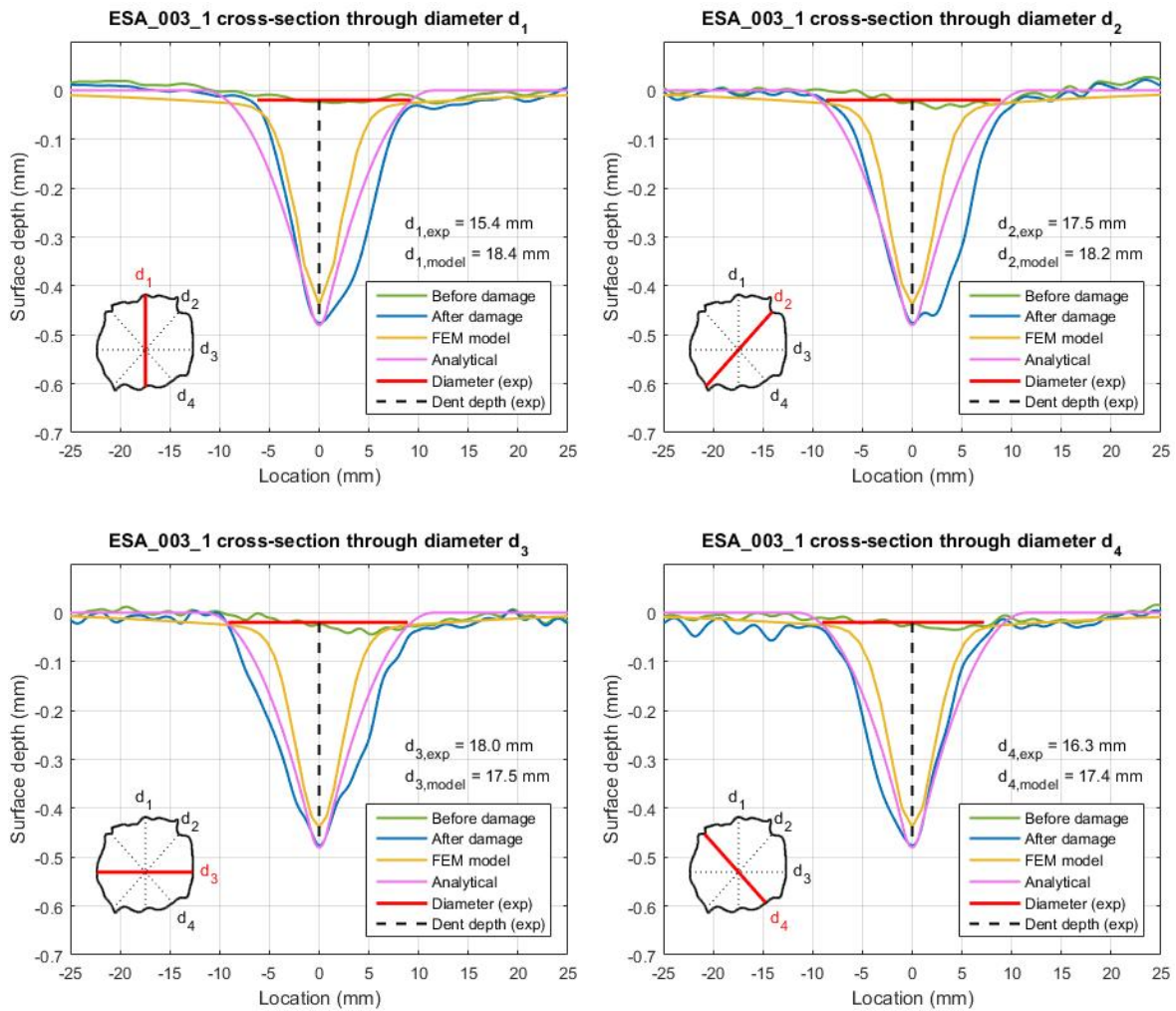


Figure 2.3. Example plots of cross-sections through the four diameters of the indentation. The plots contain cross-sections from physical experiments (before and after the damage), FEM model, and analytical calculations, as well as indicate the depth and the diameter of the indentation. The lowest point of the indentation is at the centre of the x axis. All measurements are in mm.

Table 2.1.

Example of estimated measurements of the damaged area of a specimen

Specimen	Depth	d_1	d_2	d_3	d_4	Average diameter	Max diameter	Area	Volume
ESA_003_1	0.46	15.4	17.5	18.0	16.3	16.8	18.0	231	35
ESA_003_2	0.46	16.5	18.0	19.3	15.6	17.3	19.3	248	38
ESA_003_3	0.41	17.0	18.8	19.7	18.1	18.4	19.7	287	36
ESA_003_4	0.48	14.8	16.7	18.7	15.1	16.3	18.7	230	38
ESA_003_5	0.41	15.7	16.5	18.9	15.4	16.6	18.9	242	27
ESA_003_6	0.44	13.9	17.0	18.4	15.0	16.1	18.4	230	35

2.3 Implementation details

2.3.1 General structure

Figure 2.4. depicts the general data flow in the ASIA toolbox starting from the loading of the xyz point cloud data to the returning of the results which include indentation depth, diameters, are, and volume as well as surface plots and cross-section plots.

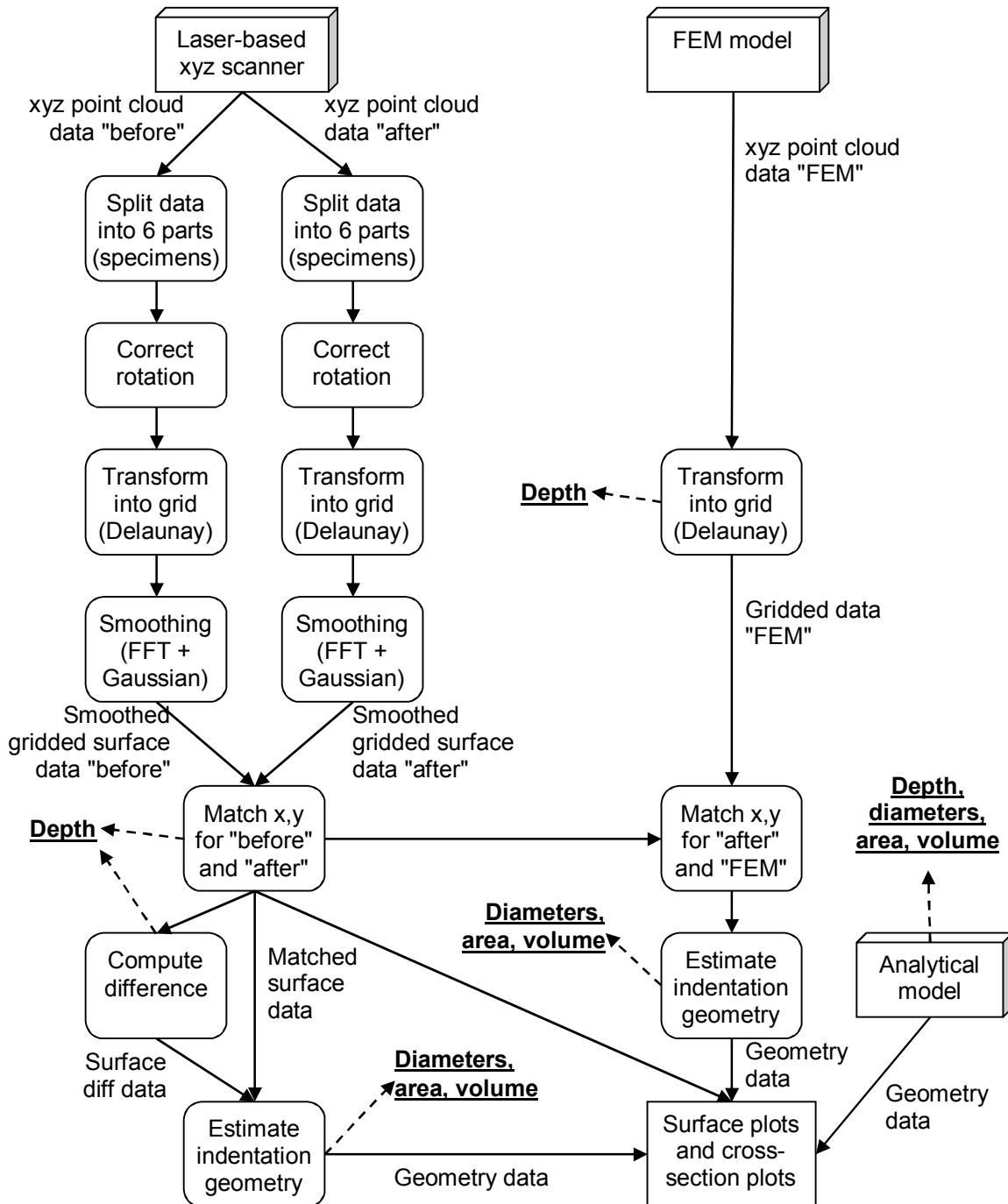


Figure 2.4. Data flow diagram of the ASIA toolbox

The ASIA toolbox consists of 25 functions (not including the private local functions and not including the script described in the previous subsection). 13 of them can be considered main functions and the rest are auxiliary functions. See next subsection for descriptions of the main functions. For brevity, only the main functions are described. The auxiliary functions that are not described here are the following: `calcangles`, `calcdiameters`, `correctrotation`, `floodrestricted`, `floodfe`, `loaddata`, `saveimage`, `smoothpanels`, `smoothfft`, `splitpanels`, `subaxis`, `parseargs`.

2.3.2 Main functions

The main functions are listed roughly in the order of usage in a full analysis process as depicted in Figure 2.4.

loadandpreprocess

The function loads and pre-processes the panel data. It loads the xyz cloud data (by calling function `loaddata`), optionally flips and/or crops it, then splits into six individual parts – specimens (by calling function `splitpanels`), corrects rotation of the data (by calling function `correctrotation`), transforms the data into a grid, and smoothes the z values (by calling function `smoothpanels`).

The loaded irregularly scattered measurement data of arbitrary resolution is transformed into a regular grid (having the defined step) using Delaunay triangulation and linear interpolation. The z values of the data are then smoothed using Fast Fourier Transform with Gaussian filter (size of the filter is defined by the input argument `smoothingParameter`).

Call:

```
panelsSmooth = loadandpreprocess(filename, flipX, flipY,
cropOnZero, isOneColumn, step, smoothingParameter, isDataUsed)
```

Input arguments:

`filename` (string) – Filename for the xyz point cloud file to be loaded.

`flipX` (boolean) – Whether to flip the data on the x axis.

`flipY` (boolean) – Whether to flip the data on the y axis.

`cropOnZero` (boolean) – Whether to delete all measurements for which x or y is smaller than zero.

`isOneColumn` (boolean) – Whether the data has only one column.

`step` (double) – step size for the regular grid to be generated from the original data.

`smoothingParameter` (double) – Smoothing parameter of the smoothing procedure (see function `smoothpanels`). The higher the value the more smoothed will be the surface of the specimen – undersmoothing and oversmoothing both cause inaccuracies in results. The parameter is independent from `step` but dependent on specimen physical size.

`isDataUsed` (3 by 2 matrix of boolean) – A matrix where each element indicates whether the data of the corresponding specimen must actually be processed.

Output arguments:

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing smoothed data for each of the six specimens.

loaddatafe

The function loads FE model xyz point cloud data and transforms it into a regular grid of the defined step using Delaunay triangulation and linear interpolation.

Call:

```
panelsFE = loaddatafe(filename1, filename2, step,
calcSurfaceAngles)
```

Input arguments:

`filename1` (string) – Filename for file containing x and y data of the FEM model.

`filename2` (string) – Filename for file containing z data of the FEM model.

`step` (double) – `step` size for the regular grid to be generated from the original data.

`calcSurfaceAngles` (boolean) – Whether to calculate surface angles at each grid point.

Output arguments:

`panelsFE` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

findlowest

The function finds the lowest point of the surface for each specimen (doesn't look at the edges).

Call:

```
[idxLowestX, idxLowestY] = findlowest(panelsSmooth,
isDataOK)
```

Input arguments:

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

`isDataOK` (3 by 2 matrix of boolean) – A matrix indicating whether the data for each specimen is usable and therefore lowest value really should be found.

Output arguments:

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

moveto0

The function moves z of the surface to 0, ignoring the inner region around the lowest point of the data.

Call:

```
panelsSmooth = moveto0(panelsSmooth, idxLowestX,
idxLowestY, step, isDataUsed, ignoredDiameter)
```

Input arguments:

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

`step` (double) – step size for the regular grid of the data.

`isDataUsed` (3 by 2 matrix of boolean) – A matrix where each element indicates whether the data of the corresponding specimen must actually be processed.

`ignoredDiameter` (double) – Approximate diameter around the lowest point of the indentation that must be ignored when estimating zero level of the z coordinate of the surface.

Output arguments:

`panelsSmooth` (3 by 2 matrix of structure) – The same as the input argument of the same name except that the z values are moved accordingly.

matchpanels

The function matches x and y offsets of "before" and "after" data.

Call:

```
[panelsSmoothBefore, panelsSmooth, panelsSmoothDiff,
idxLowestX, idxLowestY, offsetX, offsetY] =
matchpanels(panelsSmoothBefore, panelsSmooth, step,
idxLowestX, idxLowestY, isDataBeforeUsed, ignoredDiameter)
```

Input arguments:

`panelsSmoothBefore` (3 by 2 matrix of structure) – A matrix containing "before" data for each of the six specimens.

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing "after" data for each of the six specimens.

`step` (double) – step size for the regular grid of the data.

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

`isDataBeforeUsed` (3 by 2 matrix of boolean) – A matrix where each element indicates whether the "before" data of the corresponding specimen must actually be processed.

`ignoredDiameter` (double) – Approximate diameter around the lowest point of the indentation that must be ignored when matching the data.

Output arguments:

`panelsSmoothBefore` (3 by 2 matrix of structure) – The same as the input argument of the same name except that it has the necessary offsets applied for x and y coordinates so that "before" data matches "after" data.

`panelsSmooth` (3 by 2 matrix of structure) – The same as the input argument of the same name except that it has the necessary offsets applied for x and y coordinates so that "before" data matches "after" data.

`panelsSmoothDiff` (3 by 2 matrix of structure) – A matrix containing difference between `panelsSmoothBefore` and `panelsSmooth` for each of the six specimens.

matchpanelsfe

The function matches the data from the experiments and the data from the FE model so that the lowest points have the same x, y coordinates.

Call:

```
panelsFE = matchpanelsfe(panelsFE, panelsSmooth,
idxLowestX, idxLowestY)
```

Input arguments:

`panelsFE` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens from FEM model.

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

Output arguments:

`panelsFE` (3 by 2 matrix of structure) – The same as the input argument of the same name except that it has the necessary offsets applied for x and y coordinates so that its data matches the data in `panelsSmooth`.

analyzearea

The function analyzes the area of the indentation in experiment data. Estimates the damaged zone and its outline (using function `floodrestricted`), its diameters (using function `calcdiameters`), and its area. Note that the damaged area is estimated using a modified flood fill algorithm (see function `floodrestricted`) together with a modified z-score outlier detection algorithm [2] that detects the sudden significant increase in the standard deviation of radiuses of contours of the flooded area when the flood fills large area outside of the actual indentation zone.

Call:

```
results = analyzearea(panelsSmooth, step, idxLowestX,
idxLowestY, maxFloodDiameter, isDataUsed, maskIgnore,
panelsFE)
```

Input arguments:

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

`step` (double) – step size for the regular grid of the data.

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

`maxFloodDiameter` (double) – Maximum diameter around the lowest point of the indentation inside which the flood fill algorithm must find the geometry/outline of the damage (see function `floodrestricted` for details).

`isDataUsed` (3 by 2 matrix of boolean) – A matrix where each element indicates whether the data of the corresponding specimen must actually be processed.

`maskIgnore` (optional 3 by 2 cell array with elements containing masks) – Defines masks of areas of a specimen to be ignored while determining geometry/outline of the damage.

`panelsFE` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens from FEM model.

Output arguments:

`results` (structure) – A structure containing fields that define the detected outline of the damage, its area (in squared millimetres), its diameters (in millimetres), and some additional information.

analyzeareafe

The function analyzes the area of the indentation in FE data. Estimates the damaged zone (using function `floodfe`), its diameters (using function `calcdiameters`), and its area.

Call:

```
results = analyzeareafe(panelsFE, panelsSmooth, step,  
idxLowestX, idxLowestY, threshold, isThresholdZ, isDataUsed)
```

Input arguments:

`panelsFE` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens from FEM model.

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

`step` (double) – step size for the regular grid of the data.

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

`threshold` (double) – A threshold value for the z axis that the flood fill algorithm should not exceed.

`isThresholdZ` (boolean) – Whether to determine area and volume of the indentation in the FEM data by thresholding surface depth (`true`) or thresholding surface local angle (`false`).

`isDataUsed` (3 by 2 matrix of boolean) – A matrix where each element indicates whether the data of the corresponding specimen must actually be processed.

Output arguments:

`results` (structure) – A structure containing fields that define the detected outline of the damage, its area (in squared millimetres), its diameters (in millimetres), and some additional information.

calcdepth

The function computes the depth of the indentation for each specimen.

Call:

```
depth = calcdepth(panelsSmooth, idxLowestX, idxLowestY)
```

Input arguments:

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

Output arguments:

`depth` (double) – Computed depth of the indentation (in millimetres)

calcvolume

The function computes volume of the indentation for each specimen.

Call:

```
volume = calcvolume(panelsSmooth, panelsSmoothBefore,  
zqFilled, unit)
```

Input arguments:

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing "after" data for each of the six specimens.

`panelsSmoothBefore` (3 by 2 matrix of structure) – A matrix containing "before" data for each of the six specimens.

Output arguments:

`volume` (double) – Computed volume of the damaged area (in cubic millimetres).

plotpanels

The function plots the set of processed surfaces of the specimens, showing the outline of the damaged area.

Call:

```
[minZ, maxZ] = plotpanels(panelsSmooth, analysisResults,  
isDataUsed, equalScaleZ, forceMinZ, forceMaxZ, strictLook,  
allInOne, maskIgnore, filePathOutput, name)
```

Input arguments:

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing data for each of the six specimens.

`analysisResults` (structure) – results from function `analyzearea` augmented with results from function `calcdepth` and `calcvolume`.

`isDataUsed` (3 by 2 matrix of `boolean`) – A matrix where each element indicates whether the data of the corresponding specimen must actually be processed.

`equalScaleZ` (`boolean`) – Indicates whether to use equal z scale for all six specimens (`true`) or each specimen can have its own z scale (`false`).

`forceMinZ` (`double`) – Forced minimum z value for all specimens. If empty, each specimen can have its own z scale.

`forceMaxZ` (`double`) – Forced maximum z value for all specimens. If empty, each specimen can have its own z scale.

`strictLook` (`boolean`) – Whether to create the plots to be with a strict predefined zoomed-in x, y scale (`true`) or the plots may automatically adjust x, y scales to fit all data (`false`).

`AllInOne` (`boolean`) – Whether to put all six plots in a single image (`true`) or in six separate images (`false`).

`maskIgnore` (optional 3 by 2 cell array with elements containing masks) – Defines masks of areas of a specimen to be ignored while determining area of damage.

`filePathOutput` (`string`) – Path indicating where to save all the generated files.

`name` (3 by 2 cell array of `string`) – Names for specimens (used in image captions as well as in filenames of the saved files).

Output arguments:

`minZ` (`double`) – Smallest z value across all the specimens. Can be used to create a common scale for the z axis.

`maxZ` (`double`) – Largest z value across all the specimens. Can be used to create a common scale for the z axis.

calcanalytcs

The function calls analytical calculations done by the AIM function (see Section 3). The function makes sure the input and output data are in the right units of measurement.

Call:

```
[depth, diameter, area, volume, surfaceDepth] =  
calcanalytcs(panelParameters, locations)
```

Input arguments:

`panelParameters` (3 by 2 cell array of parameters for analytical calculations) – The structure contains the following parameters: ply angles, core crushing stress, diameter of indenter tip, and indentation force (see Section 3 for details).

`locations` is a vector of `double` containing all distances (in meters) from the lowest point where surface depth should be calculated.

Output arguments:

`depth` (`double`) – Calculated indentation depth (in millimetres).

`diameter` (`double`) – Calculated indentation diameter (in millimetres).

`area` (`double`) – Calculated indentation area (in millimetres).

`volume` (`double`) – Calculated indentation volume (in millimetres).

`surfaceDepth` (vector of double) – This vector contains the calculated surface depths at the distances contained in `locations`.

plotcrosssection

The function plots cross-sections through diameter for "before", "after", FE, and analytical data. In addition it shows the diameters and the depths of the indentation.

Call:

```
plotcrosssection(i, j, panelsSmooth, panelsSmoothBefore,  
analysisResultsDiff, analysisResultsFE, idxLowestX,  
idxLowestY, step, panelsFE, filePathOutput, name,  
panelParameters)
```

Input arguments:

`i` (integer) – Row index for the specimen in all of the 3 by 2 data matrices.

`j` (integer) – Column index for the specimen in all of the 3 by 2 data matrices.

`panelsSmooth` (3 by 2 matrix of structure) – A matrix containing "after" data for each of the six specimens.

`panelsSmoothBefore` (3 by 2 matrix of structure) – A matrix containing "before" data for each of the six specimens.

`analysisResultsDiff` (structure) – results from function `analyzearea` augmented with results from function `calcdepth` and `calcvolume` for the difference data between "before" data and "after" data.

`analysisResultsFE` (structure) – results from function `analyzearea` augmented with results from function `calcdepth` and `calcvolume` for the FEM model data.

`idxLowestX` (3 by 2 matrix of integer) – A matrix of indices along the x axis for the lowest point for each specimen.

`idxLowestY` (3 by 2 matrix of integer) – A matrix of indices along the y axis for the lowest point for each specimen.

`step` (double) – step size for the regular grid of the data.

`panelsFE` (3 by 2 matrix of structure) – A matrix containing FEM model data for each of the six specimens.

`filePathOutput` (string) – Path indicating where to save all the generated files.

`name` (3 by 2 cell array of string) – Names for specimens (used in image captions as well as in filenames of the saved files).

`panelParameters` (3 by 2 cell array of parameters for analytical calculations) – The structure contains the following parameters: ply angles, core crushing stress, diameter of indenter tip, and indentation force (see Section 3 for details).

3 Analytical Indentation Model (AIM) function for MATLAB

3.1 Introduction

The AIM v1 function estimates dent depth, diameter, area, and volume using an analytical model of quasi-static indentation described in detail in Section 3 of previous deliverable [4].

The AIM v2 function has two modifications compared to AIM v1, namely, it is applicable both for quasi-static indentation and impact loading cases, and also it applies minimization of the potential energy when estimating the residual dent depth in order to improve the accuracy of the dent depth prediction.

3.2 Computations done by the function AIM v1

Input

Facesheet layup:

Number of plies: n

Ply angles: $[\theta_1, \theta_2, \dots, \theta_n]$ (radians)

Ply thicknesses: h_1, h_2, \dots, h_n (m)

Ply stiffness characteristics: E_1 (Pa), E_2 (Pa), G_{12} (Pa), ν_{12}

Core crushing stress: q (Pa)

Radius of indenter tip: R (m)

Indentation force: P (N)

1) Calculate parameter L :

$$L = 3A_{11} + 2A_{12} + 3A_{22} + 4A_{66}$$

$$A_{ij} = \sum_{k=1}^n \bar{Q}_{ij}^{(k)} h_k$$

where $\bar{Q}_{ij}^{(k)}$ is calculated as follows:

$$\begin{aligned} \bar{Q}_{11} &= m^4 Q_{11} + 2m^2 n^2 (Q_{12} + 2Q_{66}) + n^4 Q_{22} \\ \bar{Q}_{12} &= n^2 m^2 (Q_{11} + Q_{22} - 4Q_{66}) + (n^4 + m^4) Q_{12} \\ \bar{Q}_{22} &= n^4 Q_{11} + 2n^2 m^2 (Q_{12} + 2Q_{66}) + m^4 Q_{22} \\ \bar{Q}_{66} &= m^2 n^2 (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66}) \\ &\quad + (m^4 + n^4) Q_{66} \end{aligned}$$

where $m = \cos \theta_k$ and $n = \sin \theta_k$ and

$$Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}}$$

$$Q_{12} = \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}}$$

$$Q_{22} = \frac{E_2}{1 - \nu_{12}\nu_{21}}$$

$$Q_{66} = G_{12}$$

where $\nu_{21} = \nu_{12} \frac{E_2}{E_1}$

2) Find non-negative δ and ρ by minimizing $\Pi(\delta, \rho)$:

$$\begin{aligned} \Pi(\delta, \rho) = & -P\delta + \frac{1}{64} \pi L \left(\frac{R^2 \rho^2}{R^2 - \rho^2} + \rho^2 + 2R^2 \log \left(1 - \frac{\rho^2}{R^2} \right) \right) + \\ & \frac{1}{240} \pi L \frac{\rho^2 \left(\sqrt{R^2 - \rho^2} - R + \delta \right)^4 \left(R^2 + 2\rho^2 - (R - \delta) \sqrt{R^2 - \rho^2} \right)}{\left(R^2 - \rho^2 - (R - \delta) \sqrt{R^2 - \rho^2} \right)^3} + \\ & \frac{1}{3} \pi q \left(2R^3 - 3(R - \delta) \rho^2 - 2(R^2 - \rho^2)^{3/2} \right) + \\ & \frac{2}{3} \pi q \frac{\left(\left(R^2 - (R - \delta) \sqrt{R^2 - \rho^2} \right)^2 - \rho^4 \right) \left(\sqrt{R^2 - \rho^2} - R + \delta \right)}{\rho^2} \end{aligned}$$

3) Calculate:

$$a = \rho + \frac{2}{\rho} \left(R^2 - \rho^2 - (R - \delta) \sqrt{R^2 - \rho^2} \right)$$

4) Find smallest positive root ρ_r ($0 \leq \rho_r \leq \rho$):

$$\begin{aligned} & \frac{1}{64} L \left(\frac{R^2 \rho^2}{R^2 - \rho^2} + \rho^2 + 2R^2 \log \left(1 - \frac{\rho^2}{R^2} \right) \right) + \frac{1}{960} L \frac{\rho^4 (a - \rho) (a + 5\rho)}{(R^2 - \rho^2)^2} = \\ & \frac{1}{64} L \left(\frac{R^2 \rho_r^2}{R^2 - \rho_r^2} + \rho_r^2 + 2R^2 \log \left(1 - \frac{\rho_r^2}{R^2} \right) \right) + \frac{1}{960} L \frac{\rho_r^4 (a - \rho_r) (a + 5\rho_r)}{(R^2 - \rho_r^2)^2} + \\ & \frac{1}{12} q \frac{-8R^4 + 4R^2 \rho^2 + 8R^3 \sqrt{R^2 - \rho^2} + \rho (a + \rho) (a^2 + \rho^2)}{\sqrt{R^2 - \rho^2}} - \\ & \frac{1}{12} q \frac{-8R^4 + 4R^2 \rho_r^2 + 8R^3 \sqrt{R^2 - \rho_r^2} + \rho_r (a + \rho_r) (a^2 + \rho_r^2)}{\sqrt{R^2 - \rho_r^2}} \end{aligned}$$

5) Calculate:

$$\delta_r = \frac{2R\sqrt{R^2 - \rho_r^2} - 2R^2 + \rho_r(a + \rho_r)}{2\sqrt{R^2 - \rho_r^2}}$$

Result

Indentation depth: δ_r (m)

Indentation diameter: $2a$ (m)

Indentation area: πa^2 (m²)

Indentation volume (m³):

$$\begin{aligned} & \frac{1}{3} \pi \left(-3R^3 + 2(R^2)^{3/2} + 3R^2 \delta_r + (R - \rho_r)(R + \rho_r) \left(3R - 3\delta_r - 2\sqrt{R^2 - \rho_r^2} \right) + \right. \\ & \left. \frac{1}{2} (a^2 + 2a\rho_r - 3\rho_r^2) \left(-R + \delta_r + \sqrt{R^2 - \rho_r^2} \right) \right) \end{aligned}$$

Predicted indentation shape:

$$w(r) = \begin{cases} \sqrt{R^2 - r^2} + \delta_r - R & 0 \leq r \leq \rho_r \\ \left(\sqrt{R^2 - \rho_r^2} + \delta_r - R \right) \left(1 - \frac{r - \rho_r}{a - \rho_r} \right)^2 & \rho_r \leq r \leq a \end{cases}$$

3.3 Computations done by the function AIM v2: differences from AIM v1

AIM v2 has one additional input parameter compared to AIM v1, namely, core unloading stress:

Core unloading stress: q_{ct} (Pa).

Core unloading stress (also called core restoring pressure [6]) is core yield strength in unloading, and it characterises the energy dissipation in the core during unloading of the sandwich. q_{ct} can be determined experimentally as the plateau level of the tensile stress in flatwise tension-after-compression test of the honeycomb core, while the compressive plateau stress in this test is the core crushing stress q .

When applied for modelling quasi-static indentation, the rest of AIM v2 input parameters are identical to those described in Section 3.2 of this deliverable and also computation steps 1, 2, 3, and 5 are the same as described above. By contrast, the residual indentation depth ρ_r upon unloading is evaluated in AIM v2 by minimizing the potential energy at zero indentation load as suggested in [7] and allowing for the work of core unloading stress q_{ct} as described in [6]. Hence computation step 4 in AIM v2 is as follows:

4) Determine ρ_r ($0 \leq \rho_r \leq \rho$) by minimizing $\Pi_{ct}(\rho_r)$:

$$\begin{aligned} \Pi_{ct}(\rho_r) = & \\ & \frac{1}{64} L \left(\frac{R^2 \rho_r^2}{R^2 - \rho_r^2} + \rho_r^2 + 2R^2 \log \left(1 - \frac{\rho_r^2}{R^2} \right) \right) + \frac{1}{960} L \frac{\rho_r^4 (a - \rho_r)(a + 5\rho_r)}{(R^2 - \rho_r^2)^2} - \\ & \frac{1}{12} q_{ct} \frac{-8R^4 + 4R^2 \rho_r^2 + 8R^3 \sqrt{R^2 - \rho_r^2} + \rho_r (a + \rho_r)(a^2 + \rho_r^2)}{\sqrt{R^2 - \rho_r^2}} \end{aligned}$$

When applied for modelling of impact indentation, the input parameters of AIM v2 remain the same as for quasi-static indentation described above, except for “Indentation force: P (N)” which is replaced by impact energy:

Impact energy: W (J)

Estimation of the maximum dent depth δ during impact in computation step 2 is based on the following simplifying assumptions. First, the kinetic energy W of the impactor is assumed to be spent only on denting the sandwich; then it is equal to the work of indentation $\int_0^\delta P(x) dx$ where $P(x)$ is the indentation force as a function of indentation depth x . (Neglecting the part of impact energy spent on global bending of the sandwich panel during impact results in some overestimation of dent size, which should lead to conservative CAI strength estimate.) Further, indentation force-depth relation is linearized, $P(x) = kx$ (such an assumption is used for simplification in the sandwich impact analysis to enable analytical solution, see e.g. [8]). This leads to $\int_0^\delta P(x) dx = P(\delta)\delta/2$. Hence the relation of impact energy and maximum indentation depth $W = P(\delta)\delta/2$ is used in computation step 2) of AIM v2 for impact loading case. It is implemented as follows:

2) Find non-negative δ and ρ by minimizing $\Pi(\delta, \rho)$ at such a positive value of P that the condition $W = \frac{P\delta}{2}$ is fulfilled:

$$\begin{aligned} \Pi(\delta, \rho) = & -P\delta + \frac{1}{64}\pi L \left(\frac{R^2\rho^2}{R^2 - \rho^2} + \rho^2 + 2R^2 \log \left(1 - \frac{\rho^2}{R^2} \right) \right) + \\ & \frac{1}{240}\pi L \frac{\rho^2 \left(\sqrt{R^2 - \rho^2} - R + \delta \right)^4 \left(R^2 + 2\rho^2 - (R - \delta)\sqrt{R^2 - \rho^2} \right)}{\left(R^2 - \rho^2 - (R - \delta)\sqrt{R^2 - \rho^2} \right)^3} + \\ & \frac{1}{3}\pi q \left(2R^3 - 3(R - \delta)\rho^2 - 2(R^2 - \rho^2)^{3/2} \right) + \\ & \frac{2}{3}\pi q \frac{\left(\left(R^2 - (R - \delta)\sqrt{R^2 - \rho^2} \right)^2 - \rho^4 \right) \left(\sqrt{R^2 - \rho^2} - R + \delta \right)}{\rho^2} \end{aligned}$$

3.4 Usage

The AIM functions implementing the computations given in the previous subsection is developed for MATLAB version 2015b. On the project disc the AIM functions can be found in the directory "AIM".

Function call for AIM version 1:

```
[depth, diameter, area, volume, surfaceDepth] =
analytics(theta, h, E1, E2, G12, v12, q, R, P, locations)
```

Function call for AIM version 2:

```
[depth, diameter, area, volume, surfaceDepth] =
analytics2(theta, h, E1, E2, G12, v12, q, qct, R, PorW,
impact, locations)
```

All the input arguments except `locations` and `impact` are defined in the previous subsection. `locations` is a vector of double containing all distances (in meters) from the lowest point where surface depth should be calculated. `theta` is a vector of double containing angles in radians. `h` is a vector of double of the same length as `theta` containing ply thicknesses in meters. `PorW` is either P or W from the previous subsection depending on the value of `impact` which is a boolean and should be set to `true` if the type of indentation is impact.

All the output arguments except `surfaceDepth` are defined in the previous subsection. `surfaceDepth` is a vector of double containing predicted surface depth at the given locations in `locations`.

Example of usage for AIM version 1 (version 2 is used very similarly):

```

% ESA_003
theta = [60 0 -60]; % ply angles (degrees)
theta = deg2rad(theta); % convert to radians
q = 1.6 * 1e6; % core crushing stress (Pa)
R = 10 * 1e-3; % radius of indenter tip (m)
P = 500; % indentation force (N)

h = repmat(8.667*1e-5, 1, length(theta)); % ply thickness (m)

% ply stiffness characteristics
E1 = 122.69*1e9; % (Pa)
E2 = 8.39*1e9; % (Pa)
G12 = 6.07*1e9; % (Pa)
v12 = 0.27;

locations = (-25:0.1:25) * 1e-3; % locations for plotting (m)

[depth, diameter, area, volume, surfaceDepth] =
analytics(theta, h, E1, E2, G12, v12, q, R, P, locations);

fprintf('depth = %0.5g, diameter = %0.5g, area = %0.5g, volume
= %0.5g\n', depth, diameter, area, volume);

% In millimetres
fprintf('In millimetres: depth = %0.5g, diameter = %0.5g, area
= %0.5g, volume = %0.5g\n', depth * 1e3, diameter * 1e3, area * 1e6,
volume * 1e9);
figure;
plot(-25:0.1:25, -surfaceDepth * 1e3);
xlabel('Location (mm)');
ylabel('Surface depth (mm)');

```

As a result MATLAB prints the following output as well as draws a plot of the indentation (see Figure 3.1).

```

depth = 0.00048009, diameter = 0.023094, area = 0.00041889,
volume = 3.6101e-08
In millimetres: depth = 0.48009, diameter = 23.094, area =
418.89, volume = 36.101

```

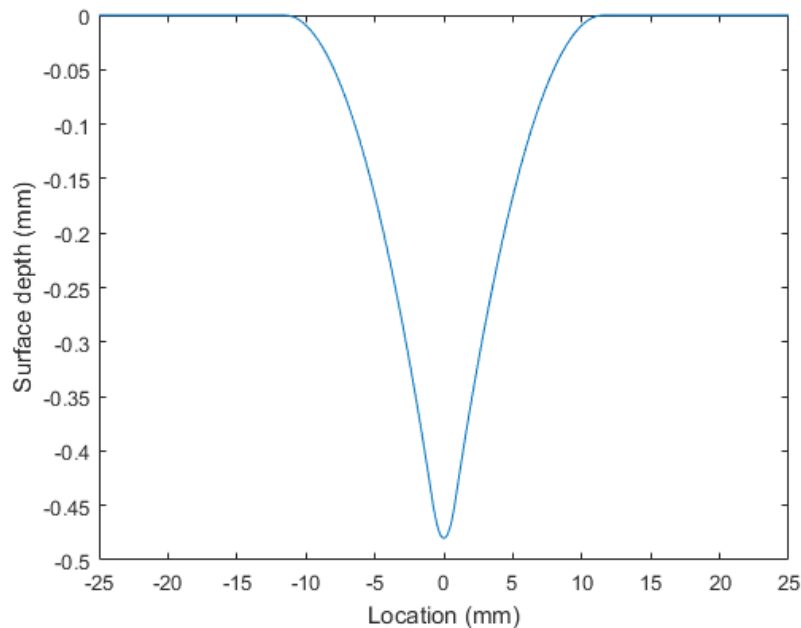


Figure 3.1. Example indentation plot from the analytical model

4 Surface Inspection Viewer (SIV)

4.1 Introduction

The SIV software tool is intended for viewing of surface inspection results after low-velocity drop weight impact or quasi-static indentation as well as for side-by-side viewing of results from FEM model and analytical calculations. Essentially this software tool enables convenient viewing of results from the ASIA toolbox and AIM function (see Sections 2 and 3).

The SIV software tool is developed as a part of the project website with reported data stored in .js and .png files and with additional code developed in JavaScript. The tool is available on the project website under the "SIV" section: http://bnm4eks.rtu.lv/tools_en.html

On the project disc the SIV tool can be started by double-clicking the file "tools_en.html" in the "Website" directory.

4.2 User interface

SIV user interface (UI) example is shown in Figure 4.1.

"1" marks the part where the user is briefly informed about how the artificial damage in the sandwich panels is performed using the two alternative methods (low-velocity drop weight impact and quasi-static indentation). Also two images are shown depicting the used low-velocity impact setup and quasi-static indentation setup.

"2" marks the UI element where the user can select a specimen for which the surface inspection results should be shown. All the UI elements marked with "2" and

above change their content dynamically depending on the choice made here. Additionally reported here are panel layup, honeycomb properties, indentation type (either Impact or Quasi-static indentation), indenter diameter (mm), as well as load (N).

"3" marks the table of indentation properties automatically computed using the ASIA toolbox and AIM function (see Sections 2 and 3) for the selected specimen. The properties are Depth (mm), Average diameter (mm), Area (mm²), and Volume (mm³). Row "Experiment" shows results from the physical experiment, row "Model" shows results from FEM model, and row "Analytical" shows results from analytical calculations.

"4" marks two surface plots of the damaged area of the specimen. The left surface plot is from a physical experiment while the right surface plot is from a FEM model. The views are slightly zoomed-in and smoothed. The lowest point of the indentation is at the centre of the plot. All measurements are in mm. Both plots are automatically generated by ASIA toolbox.

"5" marks four cross-section plots through the four measured diameters of the damaged area of the specimen. The plots contain cross-sections from physical experiments (before and after the damage), FEM model, and analytical calculations, as well as indicate the depth and the diameter of the indent. The lowest point of the indentation is at the centre of the plot. All measurements are in mm. All four plots are automatically generated by ASIA toolbox.



Surface inspection results

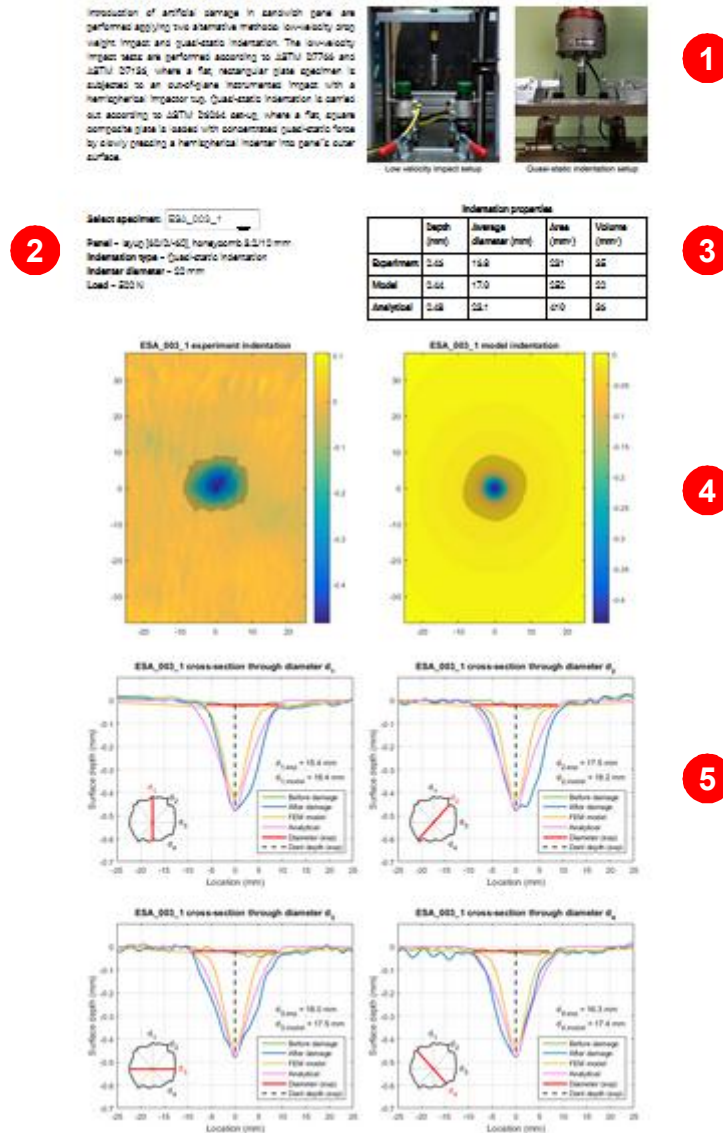


Figure 4.1. User interface of the SIV software tool

5 Compression After Impact Viewer (CAIV)

5.1 Introduction

The CAIV software tool is intended for viewing of videos of Compression After Impact (CAI) tests and the results of the tests for panels with various layouts and indentation or impact damage.

The CAIV software tool is developed as a part of the project website with reported data stored in .js and .png files as well as YouTube videos and with additional code developed in JavaScript. The tool is available on the project website under the "CAIV" section: http://bnm4eks.rtu.lv/cai_en.html

On the project disc the CAIV tool can be started by double-clicking the file "cai_en.html" in the "Website" directory.

5.2 User interface

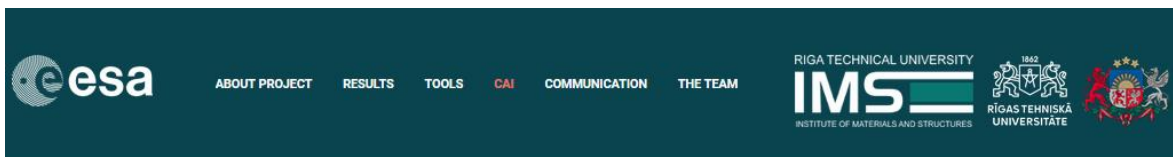
CAIV user interface (UI) example is shown in Figure 5.1.

"1" marks the UI element where the user can select a specimen for which the CAI results should be shown. All the marked UI elements change their content dynamically depending on the choice made here. Additionally reported here are panel layup, honeycomb properties, indentation type (either Impact or Quasi-static indentation), indenter diameter (mm), as well as load (N).

"2" marks table of CAI results for the specimen – Load (kN), Deflection (mm), σ_{skins} (MPa), E_{skins} (GPa).

"3" marks YouTube video of the front view of the specimen during CAI test.

"4" marks the plot of Load (N) versus Deflection (mm) for the specimen.



Compression After Impact

Compression after impact (CAI) tests on panels with various layups and various damage.

1

Select specimen:

Panel = layup [0/60/-60/0], honeycomb 6.4/20 mm (thin foil)

Indentation type = Impact

Indenter diameter = 20 mm

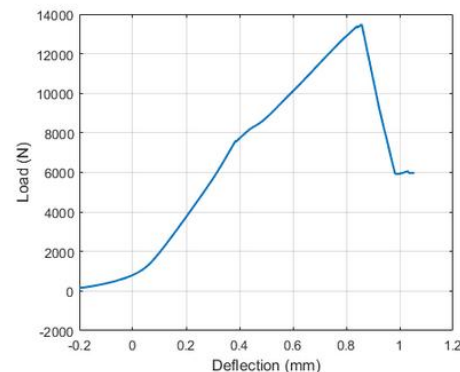
Impact energy = 0.33 J

Compression After Impact

Load (kN)	Deflection (mm)	σ_{skins} (MPa)	E_{skins} (GPa)
13.46	0.857	176.0	26.32

2

3



4

Figure 5.1. User interface of the CAIV software tool

6 Modal Assurance Criteria MAC script for MATLAB

6.1 Introduction

One of the most efficient means for the quantitative comparison of modal vectors is the Modal Assurance Criterion (MAC). The development of the MAC was modelled after the development of the ordinary coherence calculation associated with computation of the frequency response function. The MAC is a statistical indicator, just like ordinary coherence. This least squares based form of linear regression analysis yields an indicator that is most sensitive to the largest difference between comparative values and results in modal assurance criterion that is insensitive to small changes or small magnitudes. The MAC was originally introduced in modal testing in connection with The Modal Scale Factor, as an additional confidence factor in the evaluation of modal vector from different excitation locations. When an FRF matrix is expressed in the partial fraction expansion form, the numerator of each term represents the matrix of residues or modal constants [1].

The MAC is calculated as the normalized scalar product of the two sets of vectors $\{\varphi_A\}$ and $\{\varphi_X\}$. The resulting scalars are arranged into the MAC matrix:

$$MAC(r, q) = \frac{|\{\varphi_A\}_r^T \{\varphi_X\}_q|^2}{(\{\varphi_A\}_r^T \{\varphi_A\}_r) (\{\varphi_X\}_q^T \{\varphi_X\}_q)}$$

where the form of coherence function can be recognized, indicating the casual relationship between $\{\varphi_A\}$ and $\{\varphi_X\}$.

One of the big changes in the application of the MAC over the last years in the way the information is presented. Today most computer systems routinely utilize colour to present magnitude data like MAC using 2D and 3D plot, see Figure 6.1. It is important to remember that MAC is a discrete calculation and what appears as a colour contour plot really only represents the discrete mode-to-mode comparison. Nevertheless, a colour plot does allow more data to be presented in an understandable form in minimum space.

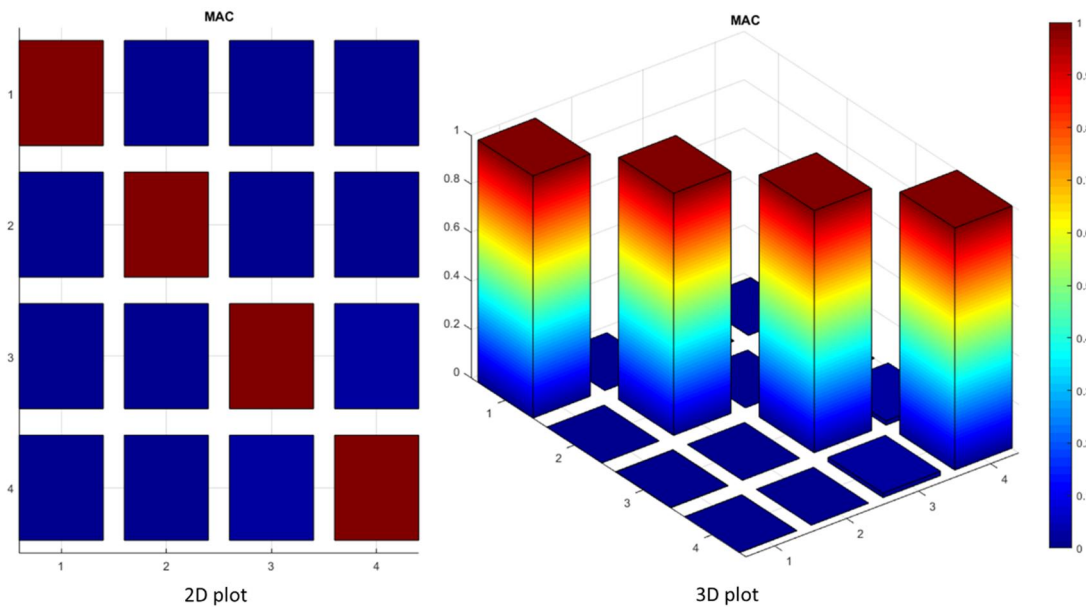


Figure 6.1. MAC plot example

6.2 Usage

The MAC script is developed for MATLAB version 2015b. On the project disc the MAC script can be found in the directory "MAC".

Before running the script the following parameters must be set:

- `filenamePrefix` (string): A path to the input files together with a prefix for the common name prefix of the input files.
- `viewSelection` (string of length 2): Selection of view combination – 'AA', 'AB', or 'BB'.
- `n` (integer): The number of eigenmodes.
- `xEigenFreq` (vector of integer): Eigenfrequency values as X axis labels.
- `yEigenFreq` (vector of integer): Eigenfrequency values as Y axis labels.

As a result a MAC plot is generated (see Figure 6.1. for an example of such plot).

7 ColorThick

7.1 Introduction

ColorThick is a program for measured thickness imperfection data processing and integration. ColorThick can be used for processing any kind ultrasound or other source bitmap image to the numerical thickness imperfection file. The main features included in the program are: ability to process data for flat or circular shells, possibility to adjust output bitmap orientation (rotate, flip), possibility to adjust field of interest, output field grid and thickness range.

ColorThick is developed using Borland Delphi version 7. On the project disc the ColorThick program can be found in the directory "ColorThick".

7.2 User interface

Figure 7.1. shows user interface (UI) of ColorThick with already loaded bitmap file. The coloured numbers "1" through "4" mark the four main input fields of the UI.

UI element 1 ("Field") defines ignorable margins of the image to exclude all unnecessary data around the ultrasound image (axis labels and values, colour scale, etc.).

UI element 2 ("Grid") controls the number of output values along each axis, as well as number of pixels used to calculate average thickness value.

UI element 3 ("Color scale") controls range of the measured thickness values and recalculates colour scheme according to input values. It should be noted that thickness range values should be transferred from the original thickness measurement bitmap colour scale.

UI element 4 ("Options") controls setting of bitmap manipulation before numerical values are calculated. Here the user must select what will be the origin of the output (plate or cylinder), physical dimensions of the object to which the bitmap will be calculated (for plate – beginning and end for x and y coordinates; for shell – beginning and end values for radius, theta, and height). Transformations panel offers rotation and flipping possibilities for the raw bitmap file as well as final data storing option (one column is one point per row or original storing sequence as per Grid field settings). Rotation and flip of the raw data is needed to align bitmap to the outer surface of the shell and initial geometrical imperfection pattern.

UI element 5 ("Calculate field") is a button for saving calculation results to a file.

Final processed file (see Figure 7.2.) has three columns – x, y, thickness.

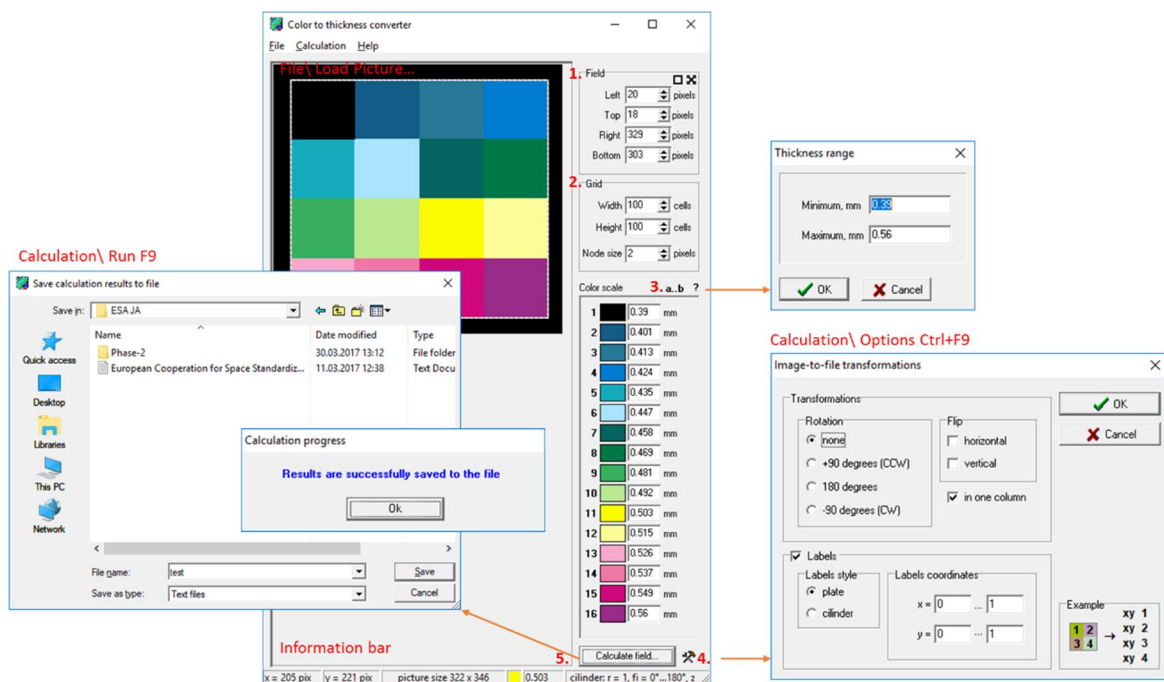


Figure 7.1. ColorThick user interface

```

0.0032 0.0035 0.3900
0.013 0.0035 0.3900
0.023 0.0035 0.3900
0.032 0.0035 0.3900
0.042 0.0035 0.3900
0.055 0.0035 0.3900
0.065 0.0035 0.3900
0.074 0.0035 0.3900
0.084 0.0035 0.3900
0.094 0.0035 0.3900
0.1 0.0035 0.3900
0.11 0.0035 0.3900
0.12 0.0035 0.3900
0.13 0.0035 0.3900
0.14 0.0035 0.3900
0.15 0.0035 0.3900
0.16 0.0035 0.3900
0.17 0.0035 0.3900
0.18 0.0035 0.3900
0.19 0.0035 0.3900
0.2 0.0035 0.3900
0.21 0.0035 0.3900
0.22 0.0035 0.3900
0.23 0.0035 0.3900
0.24 0.0035 0.3900
0.25 0.0035 0.4010
0.26 0.0035 0.4010
0.27 0.0035 0.4010

```

Figure 7.2. Example of contents of a ColorThick output file

8 ANSYS Model Configuration Tool (AMCT)

AMCT tool is intended for ANSYS APDL user configurable Finite Element Model (FEM) log file generation for analyses of Indentation (IND) / Compression after impact (CAI) simulation. Finite element model considers the substitution of the actual honeycomb structure with the solid volume. Hill plasticity was considered for core material properties. MaxStress failure criteria was considered for composite face sheets with progressive damage propagation mechanism. Cohesive zone model with contact separation was considered for adhesive layer material.

Finite element model consists of two load step non-linear analyses. Load step 1 for indentation analysis and Load step 2 for CAI analysis. Interaction between indenter and indentation frame realized by the use of surface-to-surface contact. Interaction between face sheets and core was realized as surface-to-surface contact with opening option (Cohesive zone model).

Based on standard 150 x 100 mm specimen size, standard CAI indentation frame (125 x 75 mm opening) in combination of quasi-static indentation. Model can be accustomed for the user dimensions within reasonable range, where applicable. CAI based on free side CAI setup, with 20 mm resin casting of both loading edges.

All basic input dimensions and quantities are in: mm, N, MPa.

All basic ANSYS output dimensions and quantities are in: m, N, Pa.

The user is given a form with the according input fields to define the parameters of the simulation and as a result is given a ready-to-use ANSYS APDL FEM log file for analyses of Indentation / Compression after impact simulation.

The AMCT software tool is developed as a part of the project website with code developed in JavaScript. The tool is available on the project website under the "AMCT" section: http://bnm4eks.rtu.lv/amct_en.html

On the project disc the AMCT tool can be started by double-clicking the file "amct_en.html" in the "Website" directory.

References

1. Avilés, R. 2009, The Modal Assurance Criterion (MAC). Bilbao.
2. Boris Iglewicz and David Hoaglin (1993), "Volume 16: How to Detect and Handle Outliers", The ASQC Basic References in Quality Control: Statistical Techniques, Edward F. Mykytka (ed.)
3. D7136 A. Standard Test Method for Measuring the Damage Resistance of a Fiber-Reinforced Polymer Matrix Composite to a Drop-Weight Impact Event. Analysis. 2005;i:1-16. doi:10.1520/D7136
4. Deliverable D 2.2 Numerical and analytical sandwich structure analysis-technical note, ESA Contract No. 400114452/15/NL/NDe, 2017.
5. MATLAB Release 2015b, The MathWorks, Inc., Natick, Massachusetts, United States.
6. Singh A.K., Davidson B.D., Zehnder A.T., Hasseldine B.P.J. An Analytical Model for the Response of Carbon/Epoxy-Aluminum Honeycomb Core Sandwich Structures Under Quasi-Static Indentation Loading. In: *Proceedings of the American Society for Composites - 31st Technical Conference, ASC, 2016.*
7. Zenkert D., Shipsha A., Persson K.. Static Indentation and Unloading Response of Sandwich Beams. *Composites: Part B* 2004; 35:511–522.
8. Malekzadeh Fard K., Khalili S.M.R., Forooghy S.H., Hosseini M. Low velocity transverse impact response of a composite sandwich plate subjected to a rigid blunted cylindrical impactor. *Composites: Part B* 2014; 63:111–122.