

ANSYS Composite PrepPost User's Guide



ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
<http://www.ansys.com>
(T) 724-746-3304
(F) 724-514-9494

Release 15.0
November 2013

ANSYS, Inc. is
certified to ISO
9001:2008.

Copyright and Trademark Information

© 2013 SAS IP, Inc. All rights reserved. Unauthorized use, distribution or duplication is prohibited.

ANSYS, ANSYS Workbench, Ansoft, AUTODYN, EKM, Engineering Knowledge Manager, CFX, FLUENT, HFSS and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. is certified to ISO 9001:2008.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the [legal information](#) in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Published in the U.S.A.

Table of Contents

1. Installation and Licensing	1
1.1. Installation	1
1.1.1. Windows	1
1.1.1.1. Interactive Installation	1
1.1.1.2. Silent Installation	4
1.1.2. Linux	4
1.1.3. ANSYS Update (SP and Subversion)	5
1.2. Licensing	5
1.2.1. License for ANSYS Composite PrepPost	5
1.2.2. License for the ANSYS Solver	5
2. Getting Started	7
2.1. Overview of ACP	7
2.1.1. Introduction	7
2.1.2. Principle	7
2.1.3. First Steps	9
2.2. Tutorials and Examples	9
2.2.1. Tutorial 1: First Steps	10
2.2.2. Tutorial 2: Advanced Use of ACP	10
2.2.3. Example: Class40	10
2.2.4. Example: T-Joint	11
2.2.5. Example: Kiteboard	11
2.3. Analysis of a Composite Shell Model	12
2.3.1. Pre-processing	13
2.3.1.1. Workbench Integration	13
2.3.1.2. Adding ACP Components to the Project	13
2.3.1.3. Engineering Data (ED)	14
2.3.1.4. Properties	14
2.3.1.5. Geometry and Units	16
2.3.1.6. Named Selections and Elements/Edge Sets	16
2.3.1.7. Starting and Running ACP	16
2.3.2. Workbench Analysis System	17
2.3.2.1. Adding an Analysis System to the Project	17
2.3.3. Post-processing	18
2.3.3.1. Adding an ACP (Post) Component to the Project	18
2.4. Analysis of a Composite Solid Model	21
2.4.1. Pre-processing	22
2.4.2. Workbench Analysis System	22
2.4.3. Post-processing	26
2.5. WB Workflow Examples	28
2.5.1. Single Analysis Extended	29
2.5.2. Multiple Load-cases and Analyses	30
2.5.3. Shared Composite Definition for Different Models	30
2.6. Stand-Alone Operation	31
2.6.1. Starting ACP	31
2.6.2. Command line options and batch mode	31
2.6.3. Workflow in Stand Alone Operation	32
2.7. Migration from Previous Versions	35
2.7.1. Migrate ACP Projects from v14.5 to 15.0	36
2.7.2. Migrate ACP Projects from v14.0 to 14.5	36
2.7.3. Migrate ACP Projects from v13.0 to 14.0	36

2.8. Graphical User Interface	41
2.8.1. Layout Modification	41
2.8.2. Menu	42
2.8.2.1. File	42
2.8.2.1.1. Workbench Integration	42
2.8.2.1.2. Stand-Alone	42
2.8.2.2. View	43
2.8.2.2.1. Perspectives	44
2.8.2.2.2. View Manager and Other..	44
2.8.2.3. Tools	45
2.8.2.3.1. Logger Level	45
2.8.2.3.2. ACP Preferences	45
2.8.2.3.3. ACP Submenus	46
2.8.2.3.3.1. Scene	47
2.8.2.3.3.2. Appearance	47
2.8.2.3.3.3. Screenshot	48
2.8.2.3.3.4. Interaction	48
2.8.2.4. Help	49
2.8.3. Tree View	49
2.8.4. Scene	51
2.8.5. Toolbar	51
2.8.5.1. Scene Manipulation	51
2.8.5.2. Mesh Appearance	52
2.8.5.3. Orientation Visualization	52
2.8.5.4. Draping and Flat Wrap	54
2.8.5.5. Other Features	56
2.8.5.6. Post-processing	57
2.8.5.7. Updates	57
2.8.6. Shell View	57
2.8.7. History View	57
2.8.8. Logger	57
3. Composite Model Techniques	59
3.1. T-Joint	59
3.2. Local Reinforcements	69
3.3. Ply Tapering and Staggering	71
3.3.1. Ply Tapering	71
3.3.2. Ply Staggering	74
3.4. Variable Core Thickness	75
3.4.1. Solid CAD Geometry	75
3.4.2. Look-Up Table	77
3.4.3. Geometry Cutoff Rule	78
3.4.4. General Application	80
3.5. Draping	80
3.5.1. Internal Draping Algorithm of ACP	80
3.5.2. User-Defined Draping	82
3.5.3. Visualization	84
3.6. Ply Book	85
3.7. Guide to Solid Modeling	87
3.7.1. When to use a Solid Model	88
3.7.2. How to use the Solid Model Feature	88
3.7.3. Principle of the Solid Model Generation	89
3.7.4. Workflow	89

3.7.5. Practical Tips	90
3.7.6. Known Limitations	90
3.8. Guide to Composite Visualizations	91
3.8.1. Model Verification	91
3.8.2. Post-processing visualizations	93
3.9. Element Choice in ACP	99
3.9.1. Introduction	99
3.9.2. Shell Elements	100
3.9.3. Solid Elements	100
3.9.4. Solid Shell Elements	100
4. Usage Reference	101
4.1. Features	101
4.1.1. Model	101
4.1.2. Material Data	108
4.1.3. Element and Edge Sets	125
4.1.4. CAD Geometries	127
4.1.5. Rosettes	129
4.1.6. Look-up Tables	134
4.1.7. Rules	137
4.1.8. Oriented Element Sets (OES)	144
4.1.9. Modeling Ply Groups	149
4.1.9.1. Ply Group Structure	149
4.1.9.2. Modeling Ply Group Context Menu	150
4.1.9.3. Ply Group Context Menu	150
4.1.9.4. Modeling Ply Properties	151
4.1.9.4.1. Draping	152
4.1.9.4.2. Rules	153
4.1.9.4.3. Thickness	154
4.1.9.4.4. Modeling Ply Context Menu	157
4.1.9.5. Interface Layer Properties	159
4.1.9.6. Production Ply	160
4.1.9.7. Analysis Ply	160
4.1.9.8. Import from / Export to CSV Files	161
4.1.9.9. Export Ply Geometry	161
4.1.10. Analysis Ply Groups	163
4.1.11. Sampling Elements	163
4.1.12. Section Cuts	165
4.1.13. Sensors	167
4.1.14. Solid Models	168
4.1.14.1. Solid Model Properties - General	169
4.1.14.1.1. Element Sets	169
4.1.14.1.2. Extrusion Method	170
4.1.14.1.3. Connect Butt-Jointed Plies	171
4.1.14.1.4. Drop-Off Method	171
4.1.14.1.5. Offset Direction	172
4.1.14.1.6. Drop-Off Element Handling	173
4.1.14.1.7. Element Quality	174
4.1.14.2. Solid Model Properties - Export	175
4.1.14.2.1. Use Solsh Elements	175
4.1.14.2.2. Use Solid Model Prefix	175
4.1.14.2.3. Transferred Sets	175
4.1.14.2.4. Numbering Offset	176

4.1.14.3. Extrusion Guides	177
4.1.14.3.1. Mesh Morphing	179
4.1.14.3.2. Curvature Control	180
4.1.14.3.3. Extrusion Guide Examples	181
4.1.14.4. Snap to Geometry	182
4.1.14.5. Export Solid Model...	185
4.1.14.6. Save & reload Solid Models	185
4.1.15. Layup Plots	185
4.1.16. Definitions	188
4.1.17. Solutions	190
4.1.17.1. Solution	191
4.1.17.1.1. Solution Properties	192
4.1.17.1.2. Name	192
4.1.17.1.3. Format	192
4.1.17.1.4. Paths	193
4.1.17.1.5. Data Set	193
4.1.17.1.6. Solid Model Post-Processing	193
4.1.17.2. Envelope Solution	194
4.1.17.3. Solution Plots	195
4.1.18. Scenes	198
4.1.19. Views	200
4.1.20. Ply Book	200
4.1.21. Parameters	202
4.1.22. Material Databank	205
4.2. Postprocessing	206
4.2.1. Failure Criteria	206
4.2.2. Failure Mode Measures	207
4.2.3. Principal Strains and Stresses	207
4.2.4. Limitations & Recommendations	207
4.3. Available Interfaces to FE Packages	207
4.3.1. ANSYS	207
4.3.2. ESAComp	212
4.3.3. CSV Files	213
4.3.4. HDF5	214
4.3.5. LS-Dyna	215
4.4. FAQ	215
5. Theory Documentation	219
5.1. Nomenclature	219
5.2. Draping Simulation	221
5.2.1. Introduction	221
5.2.2. Draping Procedure	222
5.2.3. Implemented Energy Algorithm	223
5.2.4. Limitations of Draping Simulations	225
5.3. Interlaminar Stresses	225
5.3.1. Introduction	225
5.3.2. Interlaminar normal stresses	225
5.3.2.1. Analytical model	225
5.3.2.2. Reference Coordinates	228
5.3.2.3. Numeric solution	228
5.3.3. Transverse shear stresses	229
5.4. Failure Analysis	231
5.4.1. Reserve factor	231

5.4.2. Weighting factors	232
5.4.3. Failure Criterion Function	232
5.4.4. Failure Criteria for Reinforced Materials	233
5.4.4.1. Maximum Strain Criterion	233
5.4.4.2. Maximum Stress Criterion	233
5.4.4.3. Quadratic Failure Criteria	234
5.4.4.3.1. Tsai-Wu Failure Criterion	234
5.4.4.3.2. Tsai-Hill Failure Criterion	236
5.4.4.3.3. Hoffman Failure Criterion	236
5.4.4.4. Hashin Failure Criterion	237
5.4.4.5. Puck Failure Criteria	238
5.4.4.5.1. Simple and Modified Puck Criterion	238
5.4.4.5.2. Puck's action plane strength criterion	238
5.4.4.5.2.1. Fiber Failure (FF)	238
5.4.4.5.2.2. Inter-fiber failure (IFF)	239
5.4.4.6. LaRC Failure Criterion	243
5.4.4.6.1. LaRC03/LaRC04 Constants	243
5.4.4.6.2. General Expressions	244
5.4.4.6.3. LaRC03 (2D)	247
5.4.4.6.4. LaRC04 (3D)	248
5.4.4.7. Cuntze's Failure Criterion	249
5.4.4.7.1. 2D Failures	249
5.4.4.7.2. 3D Failures	250
5.4.5. Sandwich Failure	251
5.4.5.1. Core failure	251
5.4.5.2. Face sheet wrinkling	251
5.4.6. Interlaminar failure	253
5.4.7. Isotropic material failure	253
5.4.8. Failure Criteria vs. Ply Type Table	253
5.5. Classical Laminate Theory	254
5.5.1. Overview	254
5.5.2. Analysis	255
5.5.2.1. Laminate Stiffness and Compliance Matrices	255
5.5.2.2. Normalized Laminate Stiffness and Compliance Matrices	255
5.5.2.3. Laminate Engineering Constants	256
5.5.2.4. Polar Properties	256
5.5.2.5. Analysis Options	257
6. The ACP Python Scripting User Interface	259
6.1. Introduction to ACP Scripting	259
6.2. The Python Object Tree	260
6.3. DB Database	261
6.4. Material Classes	262
6.4.1. MaterialData	262
6.4.2. Materials	266
6.4.2.1. StressLimits	268
6.4.2.2. StrainLimits	268
6.4.2.3. PuckConstants	269
6.4.2.4. WovenCharacterization	269
6.4.2.5. ThermalExpansionCoefficients	271
6.4.3. Fabric	271
6.4.4. Stackup	272
6.4.5. SubLaminate	274

6.5. Model Classes	275
6.5.1. Model	275
6.5.2. Rosette	293
6.5.3. LookUpTable1D	294
6.5.4. LookUpTable3D	294
6.5.5. ElementRule Classes	295
6.5.5.1. ParallelRule	295
6.5.5.2. CylindricalRule	295
6.5.5.3. SphericalRule	296
6.5.5.4. TubeRule	296
6.5.5.5. CutoffRule	296
6.5.6. EntitySet	297
6.5.6.1. ElementSet	297
6.5.6.2. EdgeSet	299
6.5.6.3. CADGeometry	299
6.5.7. OrientedElementSet	300
6.5.8. ModelingPlyGroup	302
6.5.9. ModelingPly	305
6.5.10. ProductionPly	309
6.5.11. AnalysisPly	310
6.5.12. SamplingElement	311
6.5.13. SectionCut	313
6.5.14. Sensor	314
6.5.15. PlyBook	315
6.5.15.1. PlyBook	315
6.5.15.2. Chapter	316
6.6. Solid-model Classes	316
6.6.1. SolidModel	316
6.6.2. ExtrusionGuide	321
6.6.3. SnapToGeometry	321
6.6.4. CutOffGeometry	322
6.7. Solution Classes	322
6.7.1. Solution	322
6.7.2. EnvelopeSolution	325
6.8. Scene Classes	325
6.8.1. Scene	325
6.8.2. View	327
6.9. Postprocessing Definition Classes	327
6.9.1. CombinedFailureCriteria	327
6.9.2. MaxStressCriterion	328
6.9.3. MaxStrainCriterion	329
6.9.4. TsaiWu	330
6.9.5. TsaiHill	330
6.9.6. Hashin	330
6.9.7. Hoffman	331
6.9.8. Puck	331
6.9.9. Wrinkling	333
6.9.10. CoreShear	333
6.9.11. Larc	333
6.9.12. Cuntze	334
6.9.13. VonMises	335
6.10. Plot	336

6.10.1. PlotContainer	336
6.10.1.1. PlotDataDict	336
6.10.1.2. LayupPlotDict	339
6.10.1.3. PostProcessingPlotDict	340
6.10.2. PlotData	343
6.10.2.1. PlotData	343
6.10.2.2. ContourData	346
6.10.2.3. AngleData	347
6.10.2.4. ThicknessData	347
6.10.2.5. DeformationContourData	347
6.10.2.6. StrainData	347
6.10.2.7. StressData	347
6.10.2.8. FailureData	348
6.10.2.9. TemperatureData	349
Bibliography	351
Index	353

List of Figures

2.1.T Joint Lay-up	11
2.2.ACP components	13
2.3.Engineering data sources	14
2.4.Outline of Composite Materials	14
2.5.Material properties for ACP	15
2.6.Definition of Mesh Output Options for ACP	16
2.7.Context menu of ACP (Pre) Setup	17
2.8.Connecting a Static Structural Analysis to ACP (Pre) with a drag-and-drop operation	17
2.9.Transferring an ACP Setup to an Analysis System through the context menu	18
2.10.Adding ACP (Post) component by "Transfer Data..." option	19
2.11.Adding ACP (Post) by drag and drop operation	20
2.12.Complete composite shell analysis model	21
2.13.Workbench workflow for composite solid modeling with Workbench Mechanical	23
2.14.Workbench workflow for composite solid modeling with Mechanical APDL	23
2.15.Analysis of a composite tube with metal inserts modeled with Workbench Mechanical	24
2.16.Connecting the ACP solid model to the Static Structural component	24
2.17.Suppressed Shell in Mechanical Model	25
2.18.Assembly of composite and metal solids	25
2.19.Analysis of composite plate and t-joint modeled with Mechanical APDL	25
2.20.Add reference file	26
2.21.List of used file and their order in Mechanical APDL	26
2.22.Step 1:Drag-and-drop an ACP (Post) system on to an ACP (Pre) system	27
2.23.Step 2:Drag-and-drop the Static Structural Solution cell onto the ACP (Post) Results cell	28
2.24.Single Analysis with ACP (Pre) and ACP (Post)	29
2.25.Project Schematic of a Linear Buckling Analysis	29
2.26.Multiple Load-Cases and Analyses	30
2.27.Two Analyses share the same ACP (Pre) Setup	31
2.28.Write Input File...	33
2.29.Choose Format	33
2.30.Import ANSYS Model	34
2.31.Switch with a simple click	35
2.32.Switch with a drop-down Menu	35
2.33.Restore Archive from v13 Project	36
2.34.Export the materials	37
2.35.Switch to ANSYS Workbench XML file format	37
2.36.Import materials	38
2.37.Select XML format	39
2.38.New materials	39
2.39.Engineering Data Box with question mark	39
2.40.Delete these two components	40
2.41.Create ACP (Pre) cell	40
2.42.Import Composite Definitions from ACP File	40
2.43.ANSYS Composite PrepPost GUI	41
2.44.ANSYS Composite PrepPost menu	42
2.45.File Menu for Workbench Integration	42
2.46.Stand-alone file menu	43
2.47.View Menu	43
2.48.Perspective Submenu	44
2.49.Show View	44
2.50.Logger Preferences	45

2.51. ANSYS Solver Preferences	46
2.52. ACP submenu and Section Generation Preferences	47
2.53. Scene Preferences	47
2.54. Edit Color	48
2.55. Tree View	50
2.56. Locked Rosettes and the update status	51
2.57. Orientation visualizations in the toolbar	52
2.58. Visualization of the element normals	52
2.59. Visualization of the OES normal	53
2.60. Visualization of the OES reference direction	53
2.61. Visualization of the ply angle	54
2.62. Visualization of the transverse ply angle	54
2.63. Draping "distortion" mesh	55
2.64. Ply angle vector (defined and draped)	55
2.65. Flat wrap surface of the ply	56
2.66. Enclosed box and coordinate system	57
3.1. T-joint lay-up	60
3.2. OES for the base plate	61
3.3. OES for the stringer	62
3.4. OES for bonding plies	63
3.5. Reference direction	64
3.6. Laminate of the base plate	65
3.7. Laminate of the base plate and stringer	66
3.8. First bonding laminate	67
3.9. Second bonding laminate	68
3.10. Cover plies	69
3.11. Tube rule	70
3.12. Rule tab of the modeling ply property dialog	70
3.13. Resulting local reinforcements	71
3.14. Tapered edge	72
3.15. Tapering in Ply Definition	73
3.16. Thickness distribution after core tapering	73
3.17. Superposition of Modeling Plies with identical taper angles. Schematic (middle) and Section View illustration (right).	74
3.18. Thickness distribution of a laminate with a cutoff rule	74
3.19. Template rule definition	75
3.20. Imported Core Geometry	76
3.21. Modeling ply thickness definition	76
3.22. Section with variable core thickness	77
3.23. Table definition	77
3.24. Thickness definition through a tabular values	78
3.25. Section cut and thickness contour plot	79
3.26. Imported Cutoff Geometry	79
3.27. Resulting thickness distribution (Ply Tapering activated)	80
3.28. Draping coefficients of a Fabric	81
3.29. Draping definition in OES	82
3.30. Internal draping definition	83
3.31. Tabular values definition of draping	83
3.32. Draping Mesh with Shear Energy	84
3.33. Flatwrap (boundary)	85
3.34. Fiber and draped fiber directions	86
3.35. View definition	86

3.36. Example of a Production Ply representation	87
3.37. Analyzing a Solid Model alongside a shell model	89
3.38. Solid model assembly workflow	90
3.39. T Joint Section cut	92
3.40. Class40 Section Cut	93
3.41. Activate deformed geometry in the solution properties visualizations	94
3.42. Activate the deformation plot for total deformation	94
3.43. Activate the Failure Criteria Plot with failure mode and critical Ply information	95
3.44. IRF value and Text plot for each element (Tutorial 1)	96
3.45. Zoom on critical area (Class 40)	96
3.46. Activate the ply-wise results in the plot properties	97
3.47. Select an Analysis Ply in the Modeling Ply Groups or Sampling Elements	97
3.48. Ply-wise stress (Tutorial 1)	98
3.49. Stress analysis for selected Sampling Element	99
4.1. Model context menu in stand-alone	102
4.2. Model drop-down menu in Workbench mode	102
4.3. Model Properties in stand-alone	104
4.4. Model properties in Workbench integration	105
4.5. Units system	107
4.6. Solver information (solve.out)	107
4.7. Export Composite Definitions window	108
4.8. Import Composite Definitions window	108
4.9. Materials class context menu in Stand-Alone mode	109
4.10. Stackup sequence with even symmetry	121
4.11. Stackup sequence with odd symmetry	122
4.12. Layup information and polar properties	123
4.13. CLT Analysis results	123
4.14. Properties based on the classical laminate theory	124
4.15. Properties based on the classical laminate theory	124
4.16. Element Set Selection	126
4.17. Element Set Context Menu	126
4.18. Edge Set Definition	127
4.19. Import external CAD Geometry	128
4.20. Project Schematic with a CAD Geometry Import in ACP	129
4.21. Property dialog	130
4.22. Oriented Element Set with a Radial Rosette. The yellow arrows indicate the reference direction of each element.	131
4.23. Oriented Element Set with a Cylindrical Rosette	132
4.24. Oriented Element Set with a Spherical Rosette	132
4.25. Edge wise Rosette	133
4.26. Right-click Menu on Look-Up Tables head node	134
4.27. Right-click Menu on Look-Up Tables	134
4.28. Look-up Table Tree	135
4.29. 1D Look-Up Table Properties	135
4.30. Schematic of 1D Look-Up table function	136
4.31. Look-up table edition	136
4.32. Look-up table interpolation parameters	137
4.33. Rules context menu	137
4.34. Definition of a parallel rule	138
4.35. Example of a parallel rule	139
4.36. Example of tube rule	140
4.37. Cutoff Rule Properties	141

4.38. Trailing edge with cutoff plies (ply tapering activated)	142
4.39. Section of the Cutoff Geometry	142
4.40. Core thickness without ply tapering (left) and with ply tapering (right)	143
4.41. Taper cutoff rule definition	143
4.42. Section with the production ply option	144
4.43. Section with the analysis ply with tapering option	144
4.44. Definition	145
4.45. The reference direction of a bonding laminate defined by two Rosettes and a Minimum Angle Selection Method	147
4.46. Rules	148
4.47. Draping	148
4.48. Object tree of a layup definition	150
4.49. Context menu of Modeling Ply Groups	150
4.50. General information	151
4.51. Draping definition	152
4.52. Draping Calculation options	153
4.53. Thickness definition	154
4.54. Thickness definition options	155
4.55. Core geometry	155
4.56. Resulting section cut	156
4.57. Edge tapering	157
4.58. Taper Edge example	157
4.59. Right-click modeling ply Menu	158
4.60. Interface Layer Properties - General	159
4.61. Interface Layer Properties - Open Area	160
4.62. Menu	160
4.63. Context menu production ply	160
4.64. Export Ply Geometry Window	162
4.65. Sections definition from a post-processing model	163
4.66. Definition	164
4.67. Layup sequence and enhanced post-processing	165
4.68. Section Cut definition	166
4.69. Sensor Properties	167
4.70. Solid Model feature in the GUI tree view	169
4.71. "Connect Butt-Jointed Plies" option activated	171
4.72. "Connect Butt-Jointed Plies" option deactivated	171
4.73. Extrusion direction	172
4.74. Solid model with Surface Normal direction	172
4.75. Solid model with Shell Normal direction	173
4.76. Export with drop-off elements	173
4.77. Export without drop-off elements	173
4.78. Disabling the use of global drop-off material option for a core material	174
4.79. Transferred element sets in Workbench Mechanical	176
4.80. Extrusion without and with an Edge Set Guide	178
4.81. Properties for a direction Extrusion Guide	179
4.82. Mesh morphing diagram	180
4.83. Example of a direction-type Extrusion guide with different mesh morphing radii. The location of the edge set is indicated by the circle in the bottom left corner.	181
4.84. Example of a geometry-type Extrusion guide with different mesh morphing depths.	182
4.85. Extrusion without snap operation	183
4.86. Extrusion with snap to geometry at the top (shell geometry also displayed)	184
4.87. Extrusion with snap to geometry at the top and bottom (shell geometry also displayed)	184

4.88. Example of thickness plot (tutorial 2)	186
4.89. Angle plot properties - General tab	187
4.90. Thickness plot properties - Legend tab	188
4.91. Failure Criteria Definition	189
4.92. Puck failure criteria configuration options	190
4.93. Solutions object in the tree view	191
4.94. Solution Properties window showing three solutions on the Data tab	192
4.95. Comparison of imported and recomputed interlaminar stresses (A solid stack is a single layered solid element that represents multiple layers)	194
4.96. Envelope Solution Properties windows	195
4.97. Scene with Failure Mode Plot activated. Critical failure mode, critical layer and critical load case are displayed above the visualization threshold.	198
4.98. Scene Properties	199
4.99. Draping plot for a hemisphere	200
4.100. One page of a ply book	202
4.101. Connection of ACP and Workbench Parameter Interface	203
4.102. Parameter Properties	203
4.103. Setting the maximum IRF as an output parameter	205
4.104. Material databank	205
4.105. Shell 91 keyoptions	209
4.106. Shell 99 keyoptions	210
4.107. Shell 181 keyoptions	210
4.108. Shell 281 keyoptions	211
4.109. Solid 185 keyoptions	211
4.110. Solid 186 keyoptions	212
4.111. ESAComp Options	212
4.112. ESAComp FE import and export units	213
4.113. Regional options	214
4.114. List separator customization	214
5.1. Deformation of the draping unit cell	222
5.2. Draping scheme	222
5.3. Draping modes: mode 0 (left), mode 1 (center) and mode 2 (right).	224
5.4. Angle notation for the draping energy algorithm	224
5.5. Doubly curved FE geometry	226
5.6. Integration scheme	229
5.7. Fracture curve in σ_1, τ_{21} space for $\sigma_1 = 0$. Three different fracture modes A, B, C are distinguished [28].	239

Chapter 1: Installation and Licensing

1.1. Installation

ANSYS Composite PrepPost (ACP) is not installed during the default Workbench installation procedure and has to be installed after ANSYS v15.0. In addition the ACP installation executable must be *run as administrator*.

1.1.1. Windows

ANSYS Composite PrepPost is supported on the following Windows platforms and operating system levels:

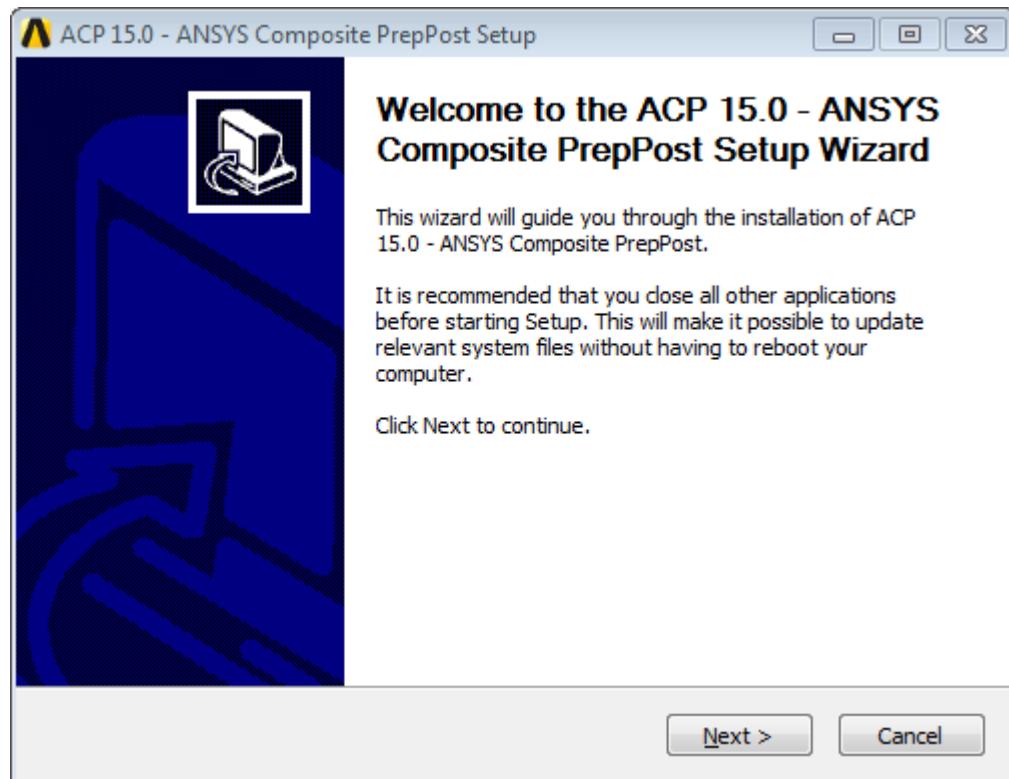
- Windows XP (sp2) 32bit/64bit
- Windows 7 32bit/64bit
- Windows 8 64bit

For Windows 7 and Windows 8, please refer to ANSYS Installation Documentation for Windows (*Chapter 2: Platform Details*) for more information on the necessary administrator rights and the User Account Control (UAC) management.

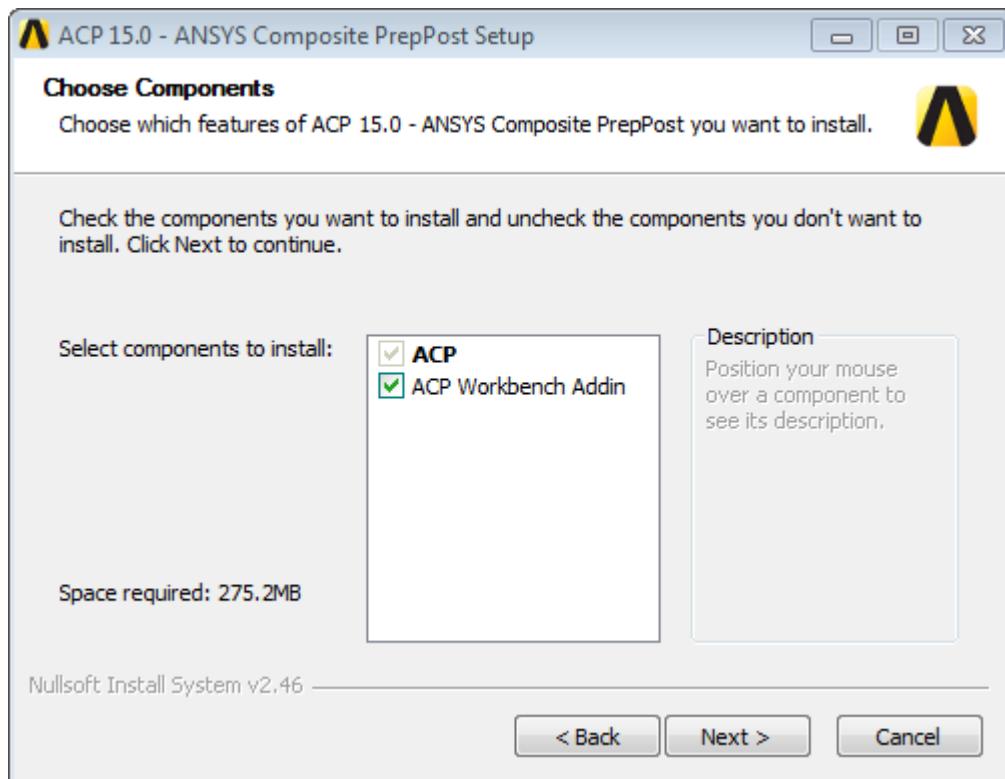
1.1.1.1. Interactive Installation

Follow these steps to install ACP:

- Download the installation executable from the [ANSYS Customer Portal](#).
- To install, double-click the
 `ACP-15.0-RXXXX.exe`
 file.
- Click "Next" on the Welcome page.
- The installation launcher appears. Choose if you want to also install the Workbench Addin of ACP:



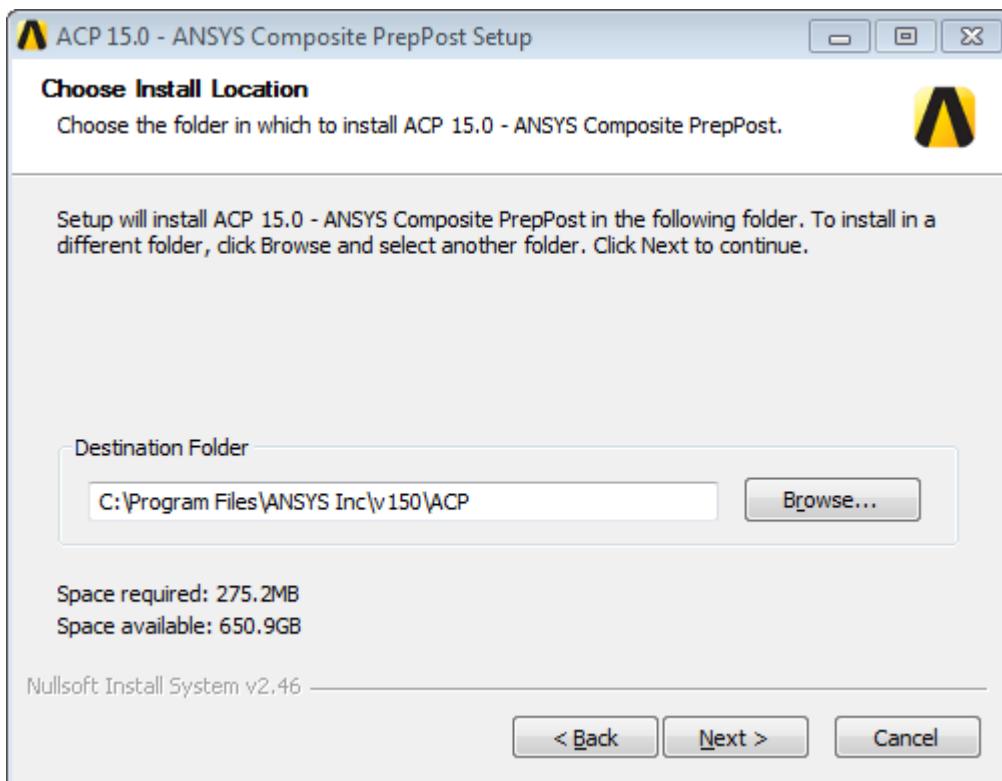
- Click "Next" to continue.



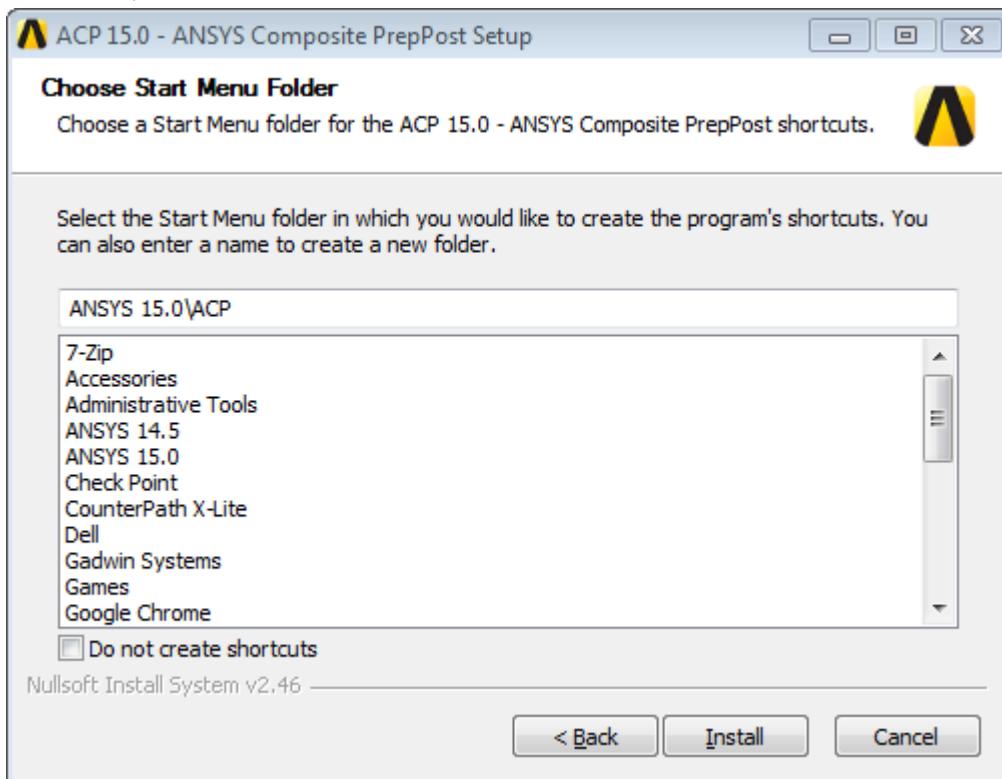
More information about the Workbench Addin is given in the Section [Workbench Integration](#).

- Click "Next" to continue.

- Give the path where ANSYS Composite PrepPost will be installed (the ANSYS installation directory is recommended):



- Click "Next" to continue.
- Choose if you want to create shortcuts in the Windows Start Menu:



- Click "Install" to launch the installation.

- At the end of the installation, click "Next", and then "Finish" to close the installation wizard.

1.1.1.2. Silent Installation

ANSYS Composite PrepPost supports a silent mode installation where the ACP Workbench Addin is installed by default. The arguments for the silent installation are:

/S	: Silent installation (no prerequisites are installed).
/D=path	: Sets the default installation directory to path.
/AWP_ROOT150=path	: Sets the path to the ANSYS Workbench directory for the ACP Workbench Addin..

Example for a silent installation:

```
ACP-15.0-RXXXX.exe /S /AWP_ROOT150=C:\Program Files\ANSYS Inc\v150
```

Either the environment variable or command line option

```
AWP_ROOT150
```

have to be set for a successful installation of the ACP Workbench Addin.

1.1.2. Linux

ANSYS Composite PrepPost is also supported on the operating systems RedHat Enterprise Linux 5 and 6 for 64-bit Platform. Both Linux platforms are supported by the same installer, which is now named

```
ACP-15.0-rXXXX-lin64-rhel.sh
```

To install ACP on Linux, use the self-extracting file

```
ACP-15.0-rXXXX-lin64-rhel.sh
```

The installation command is:

```
sh ACP-15.0-rXXXX-lin64-rhel.sh [options] [target-dir]
```

options

\-h, --help	Show this help message and exit
\-v, --verbose	Display more information during installation
\-t, TEMPDIR, --tempdir=TEMPDIR	Specify the temporary installation directory (default is /tmp)
target-dir	Specify installation directory (default is /usr/ansys_inc/v150)

To uninstall ACP, execute the

```
uninstallACP.py
```

script located in the installation directory.

Workbench Addin: The ACP Workbench Addin is included in the installation. On Linux, the ACP components within ANSYS Workbench are only available if the environment variable ACP150_DIR is correctly set to the actual installation path of ACP. Example for a default installation:

```
export ACP150_DIR=/ansys_inc/v150/ACP
```

Notice: ANSYS licensing might not work if there is no symbolic link called
`/ansys_inc`
 pointing to the ANSYS installation directory.

To run ACP on Linux refer to Section [Starting ACP \(p. 31\)](#)

1.1.3. ANSYS Update (SP and Subversion)

As the installation of ACP modifies some *ANSYS Workbench* files, ACP must be uninstalled before the Service Pack is installed. The operations must be made in following order:

- Uninstall ACP (with Administrator rights)
- Install ANSYS Service Pack
- Reinstall ACP (with Administrator rights)

1.2. Licensing

Layered elements, which are used within ACP, are supported by these solver licenses:

- Prof NLS
- Structural
- Mechanical
- Multiphysics

1.2.1. License for ANSYS Composite PrepPost

ANSYS Composite PrepPost 15.0 is licensed with the ANSYS, Inc. License Manager v15.0. Previous versions of the ANSYS, Inc. License Manager are not supported. The ANSYS Composite PrepPost license does not appear in the ANSYS *License Preferences*.

One ANSYS Composite PrepPost license contains one *ACP Pre* and one *ACP Post* License. This allows to work simultaneously with *ACP Pre* and *ACP Post* without closing the application.

1.2.2. License for the ANSYS Solver

In the Workbench environment, the license used for the solver is defined in the *User Preferences*. In the *stand-alone mode* of ACP, the ANSYS solver license must be specified in the ACP Preferences (see [ACP Preferences](#)).

Chapter 2: Getting Started

The following sections provide information about ACP and how to get started.

- [2.1. Overview of ACP](#)
- [2.2. Tutorials and Examples](#)
- [2.3. Analysis of a Composite Shell Model](#)
- [2.4. Analysis of a Composite Solid Model](#)
- [2.5. WB Workflow Examples](#)
- [2.6. Stand-Alone Operation](#)
- [2.7. Migration from Previous Versions](#)
- [2.8. Graphical User Interface](#)

2.1. Overview of ACP

The following sections provide an overview of ACP.

- [2.1.1. Introduction](#)
- [2.1.2. Principle](#)
- [2.1.3. First Steps](#)

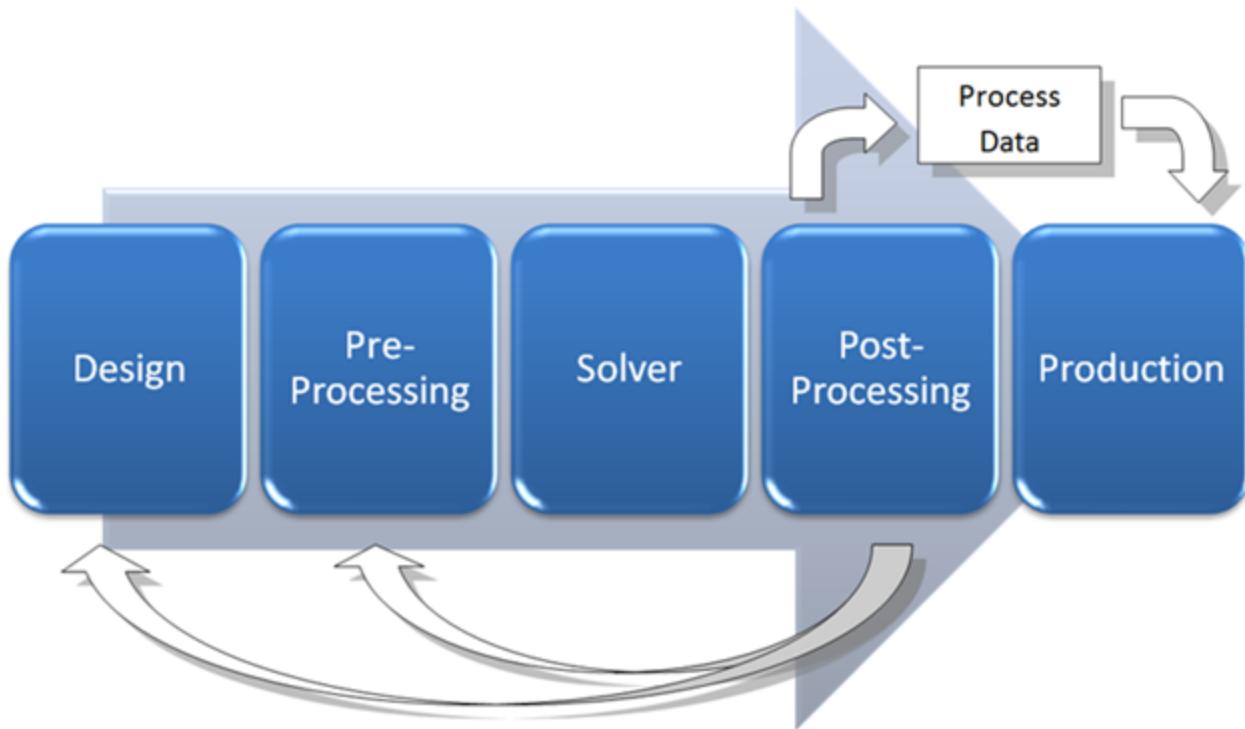
2.1.1. Introduction

Composite materials are created by combining two or more layered materials, each with different properties. These materials have become a standard for products that are both light and strong. Composites provide enough flexibility so products with complex shapes, such as boat hulls and surfboards, can be easily manufactured.

Engineering layered composites involves complex definitions that include numerous layers, materials, thicknesses and orientations. The engineering challenge is to predict how well the finished product will perform under real-world working conditions. This involves considering stresses and deformations as well as a range of failure criteria. ANSYS Composite PrepPost provides all necessary functionalities for the analysis of layered composite structures.

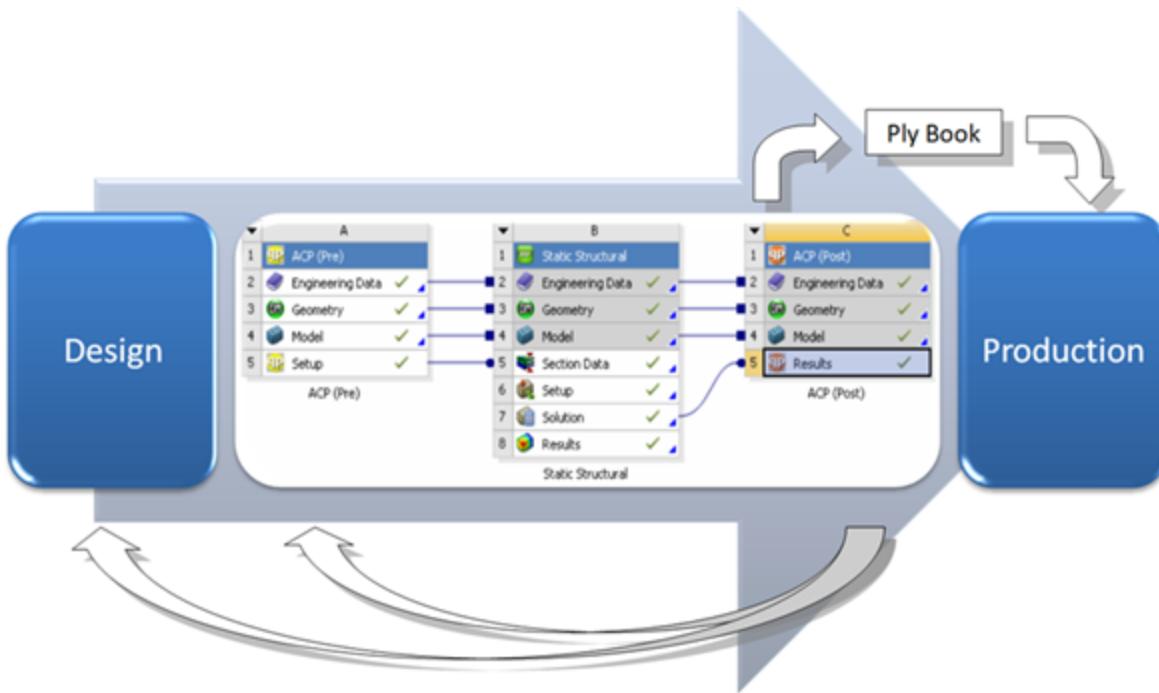
2.1.2. Principle

ANSYS Composite PrepPost (ACP) is an add-in to ANSYS Workbench and is integrated with the standard analysis features. The entire workflow for composite structure can be completed from design to final information production as a result.



The geometry of the tooling surfaces of a composite structure is the basis for analysis and production. Based on this geometry and a FE mesh, the boundary conditions and composite definitions are applied to the structure in the pre-processing stage. After a completed solution, the post-processing is used to evaluate the performance of the design and laminate. In the case of an insufficient design or material failure, the geometry or laminate has to be modified and the evaluation is repeated.

ACP has a pre- and post-processing mode. In the pre-processing mode, all composite definitions can be created and are mapped to the geometry (FE mesh). These composite definitions are transferred to the FE model and the solver input file. In the post-processing mode, after a completed solution and the import of the result file(s), post-processing results (failure, safety, strains and stresses) can be evaluated and visualized.



2.1.3. First Steps

The best way to get to know ACP features is to attempt one of the tutorials. There are two tutorials that explain step-by-step how to define and analyze basic composite structures. The tutorials both start off with existing Workbench projects. These sample projects and more information can be found in section [Tutorials and Examples \(p. 9\)](#). Knowledge of ANSYS Workbench is a prerequisite.

For information on how to build a composite model from new, please see section [Analysis of a Composite Shell Model \(p. 12\)](#).

There are many ways how to implement ACP in Workbench. The workflow for modeling composite solid element models is described in section [Analysis of a Composite Shell Model \(p. 12\)](#). Other examples are shown in section [WB Workflow Examples \(p. 28\)](#).

The section [Composite Model Techniques](#) offers an insight into modeling approaches for common composite problems.

Explanations and specific information of the ACP features can be found in section [Features](#).

Background information on the underlying theory used in ACP is available in section [Theory](#). This is especially of interest for the failure criteria.

2.2. Tutorials and Examples

The tutorials and examples can be found in the folder: <Installation Dir>\ANSYS Inc\<vXY>\ACP\<Release>\examples

[2.2.1. Tutorial 1: First Steps](#)

[2.2.2. Tutorial 2: Advanced Use of ACP](#)

[2.2.3. Example: Class40](#)

[2.2.4. Example: T-Joint](#)

[2.2.5. Example: Kiteboard](#)

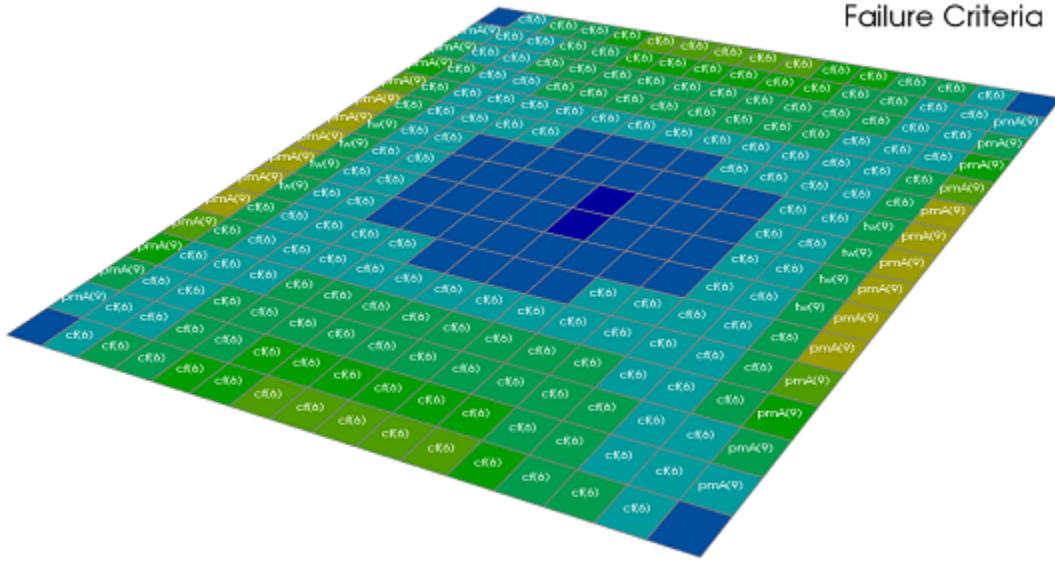
2.2.1.Tutorial 1: First Steps

In this first tutorial, a simple sandwich plate is defined from scratch.

The layup will be defined first and some basic post-processing operations show how efficient composite structures can be analyzed with ACP Post.

ACP Model

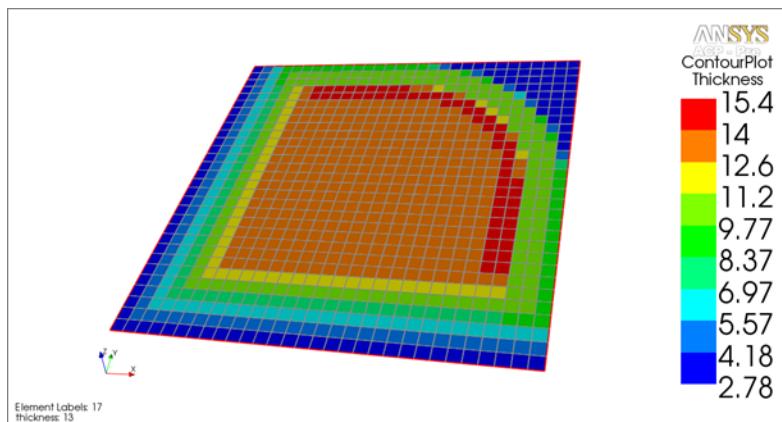
Failure - If
Element-Wise
Max: 0.85115
Min: 0.12445



2.2.2.Tutorial 2: Advanced Use of ACP

The second example Tutorial_2 illustrates some advanced features of ACP Pre.

- Tapered edges
- Core with variable thickness
- Local reinforcements



2.2.3.Example: Class40

Basic and advanced ACP Pre features are used in this example for the layup definition of a sail boat.

- Patches are defined to reinforce the structure locally

- The sandwich core has a variable thickness and is tapered as well
- Draping is also configured

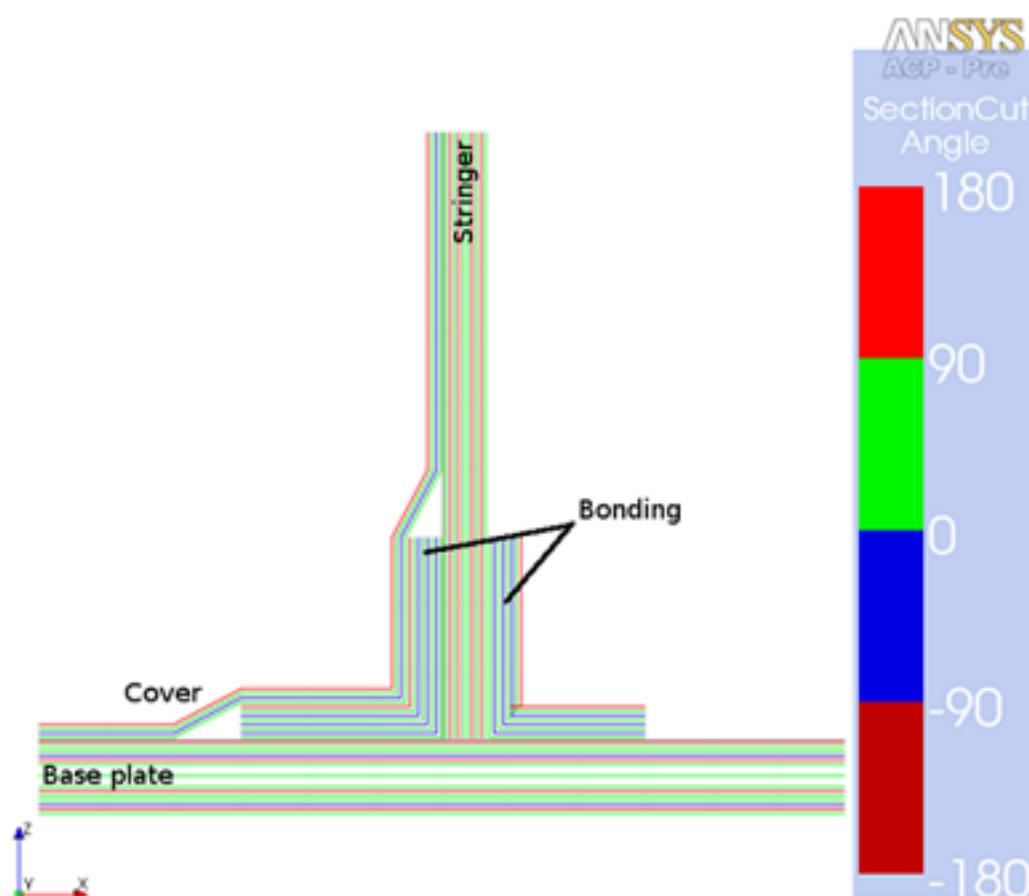
Start ANSYS Workbench and restore the class40.wbpz archive or start ACP as stand alone and open the class40.acp file.

2.2.4. Example: T-Joint

Bonding layers are often used to join different composite parts. The outcome is a complex laminate where offsets and local reinforcements have to be considered. This second example shows how a T-joint can be modeled within ACP.

Start ACP as stand alone program and open t-joint.acp.

Figure 2.1: T Joint Lay-up

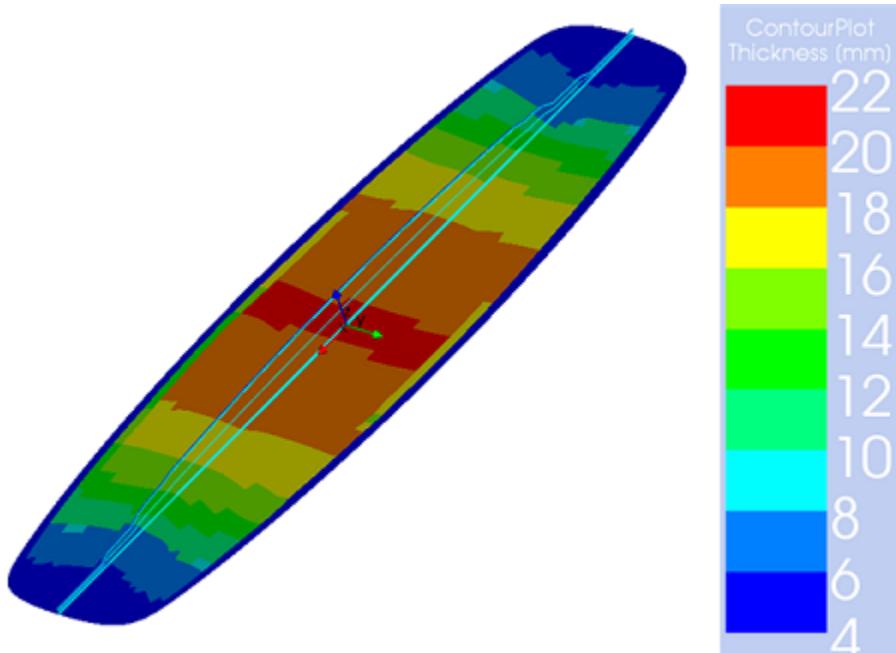


2.2.5. Example: Kiteboard

The ability to parameterize an ACP composite model is a powerful design tool. Parameter studies can be carried out efficiently in the Workbench environment. This example focuses on the parameter functionality for the layup of a kiteboard. Input and output parameters are defined in both ACP (Pre) and ACP (Post):

- Control over material choice, ply stackup and rules
- Readout of weight, Inverse Reserve Factor and deformations

Start ANSYS Workbench and restore the kiteboard.wbpz.



2.3. Analysis of a Composite Shell Model

The tutorials 1 & 2 ([Tutorials and Examples \(p. 9\)](#)) give a good insight into the ACP functionality. While the tutorials always start with an existing model this section outlines the generic build-up of a composite shell model. Selected steps are explained in more detail below and highlighted with a link.

- **Pre-processing**

- Add ACP (Pre) component to the project
- Define Engineering Data
- Import or construct Geometry (Units)
- Open the Model and
 - [Define Named Selections/Element Sets](#)
 - [Generate Mesh](#)
- Open ACP (Pre) and
 - [Define Fabric](#)
 - [Define Rosettes and Oriented Element Sets](#)
 - [Create Modeling Plies](#)

- **Workbench Analysis System**

- Add Analysis System to the project
- Open the Analysis System and

- Define Analysis Settings
- Define Boundary Conditions
- Solve model (update the project)
- **Post-processing**
 - Add ACP (Post) component to the project
 - Open ACP (Post) and run the post-processing

2.3.1. Pre-processing

The steps involved in pre-processing are described in the sections below:

- 2.3.1.1. Workbench Integration
- 2.3.1.2. Adding ACP Components to the Project
- 2.3.1.3. Engineering Data (ED)
- 2.3.1.4. Properties
- 2.3.1.5. Geometry and Units
- 2.3.1.6. Named Selections and Elements/Edge Sets
- 2.3.1.7. Starting and Running ACP

2.3.1.1. Workbench Integration

The Workbench Add-in of ACP installs two additional Component Systems to the Workbench Toolbox: ACP (Pre) and ACP (Post). These systems allow transferring the composite definitions of ACP between ACP and Mechanical on the Workbench schematic level. ACP is now fully integrated in the data structure of ANSYS Workbench and the update and refresh logic.

It is important that the user updates (refresh) the upstream data to pass the modifications to the ACP components. The update symbols can be used to check the up-to-date status of each component.

2.3.1.2. Adding ACP Components to the Project

The components ACP (Pre) and ACP (Post) are available in the Toolbox menu.

Figure 2.2: ACP components



These components are handled in the Project Schematic like the other standard components (drag-and-drop or right mouse-click menu).

2.3.1.3. Engineering Data (ED)

With the installation of ACP a new material catalog named Composite Materials is available in the databank. This catalog contains typical materials used in composite structures like unidirectional and woven carbon and glass, or core materials. Within the Workbench workflow of ACP, the materials have to be defined in the ED and not in ACP (Pre).

Figure 2.3: Engineering data sources

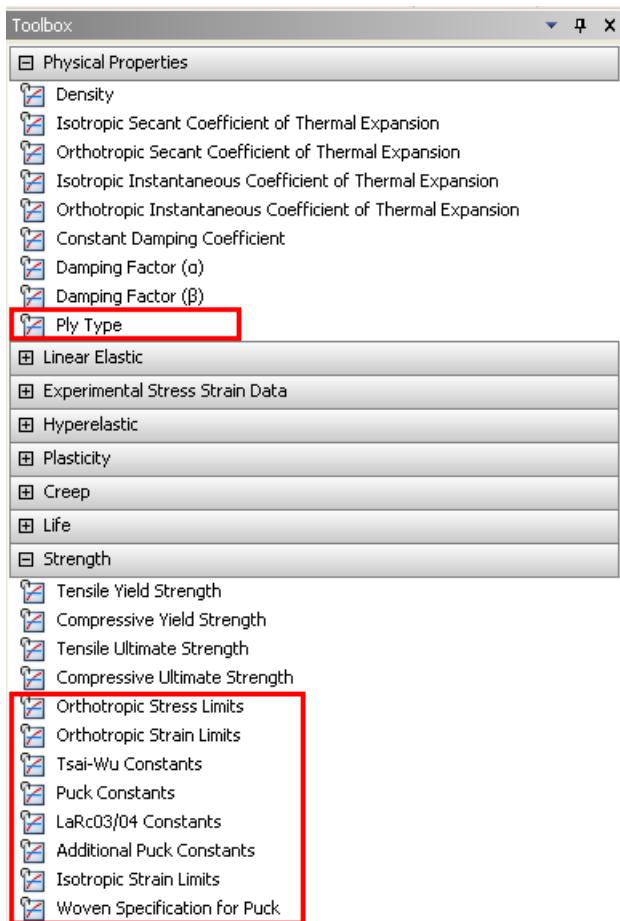
	A	B	C	D
	Data Source		Location	Description
1	Favorites			Quick access list and default items
2	General Materials			General use material samples for use in various analyses.
4	General Non-linear Materials			General use material samples for use in non-linear analyses.
5	Explicit Materials			Material samples for use in an explicit analysis.
6	Hyperelastic Materials			Material stress-strain data samples for curve fitting.
7	Magnetic B-H Curves			B-H Curve samples specific for use in a magnetic analysis.
8	Thermal Materials			Material samples specific for use in a thermal analysis.
9	Fluid Materials			Material samples specific for use in a fluid analysis.
10	Composite Materials			Material samples specific for composite structures.
*	Click here to add a new library			

Figure 2.4: Outline of Composite Materials

	A	B	C	D	E
	Contents of Composite Materials	Add	Source	Description	
1	Material				
3	Epoxy_Carbon_UD_230GPa_Prepreg				
4	Epoxy_Carbon_UD_230GPa_Wet				
5	Epoxy_Carbon_UD_395GPa_Prepreg				
6	Epoxy_Carbon_Woven_230GPa_Prepreg				
7	Epoxy_Carbon_Woven_230GPa_Wet				
8	Epoxy_Carbon_Woven_395GPa_Prepreg				
9	Epoxy-EGlass_Wet				
10	Honeycomb				
11	SAN Foam 103kgm3				

2.3.1.4. Properties

To fulfill the ACP requirements, the materials in ANSYS Workbench have some additional properties which are highlighted below.

Figure 2.5: Material properties for ACP**The new properties are:**

- Ply Type: Physical behavior of the material like core, unidirectional or woven ply.
- Strengths:
 - Orthotropic Stress Limits
 - Orthotropic Strain Limits
 - Isotropic Strain Limits
- Composite Failure Parameters:
 - Tsai-Wu Constants
 - Puck Constants
 - LaRc03/04 Constants
 - Additional Puck Constants
 - Woven Specification for Puck.

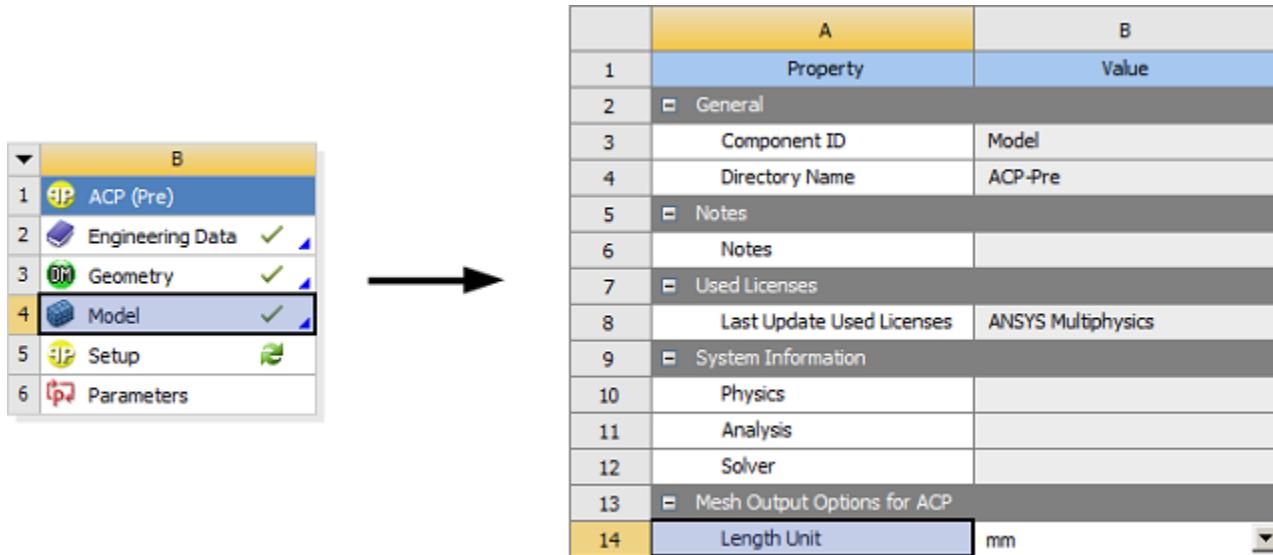
More information about the ACP material definitions are described in Section [Material Data](#).

2.3.1.5. Geometry and Units

A shell geometry is required for building any composite model in ACP. The geometry can either be constructed in the ANSYS Design Modeler or imported as a CAD file.

The unit system in ACP is defined by the length unit in the mesh output options for ACP in the Workbench project schematic. The length unit can be set in the properties of the Model cell that precedes the ACP (Pre) cell. The ACP unit system is independent from the unit system in the Mechanical application (User Interface or Solver). The transfer from the Mechanical application to ACP and vice versa automatically converts the data. The current unit system is displayed in the status bar of ACP at the bottom of the screen

Figure 2.6: Definition of Mesh Output Options for ACP

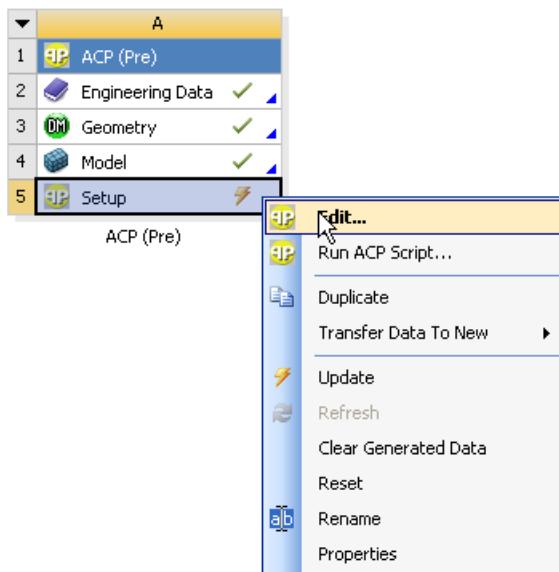


2.3.1.6. Named Selections and Elements/Edge Sets

Named Selections based on bodies, surfaces and edges defined in the Design Modeler or the Mechanical application are transferred to ACP as Element Set and Edge Set, respectively. They are necessary for building a composite model.

2.3.1.7. Starting and Running ACP

First, an ACP (Pre) component has to be defined in the Workbench project. Double-click on Setup to open ACP (Pre). You can also use the context menu and select Edit, or run a Python script in which the ACP commands are included, from the context menu. After defining the composite data in ACP (Pre), the user can return to the Workbench Project to proceed. The ACP data is saved with Save in the Workbench Project or any other Save Project command in the different components.

Figure 2.7: Context menu of ACP (Pre) Setup

2.3.2. Workbench Analysis System

The Workbench analysis system is described in the sections below:

2.3.2.1. Adding an Analysis System to the Project

The ACP components are handled in the Workbench project schematic like any other standard components. The components can be connected by drag-and-drop operations or using the context menu. The Mesh, Engineering Data, Named Selections and Coordinate Systems are transferred to the Analysis System.

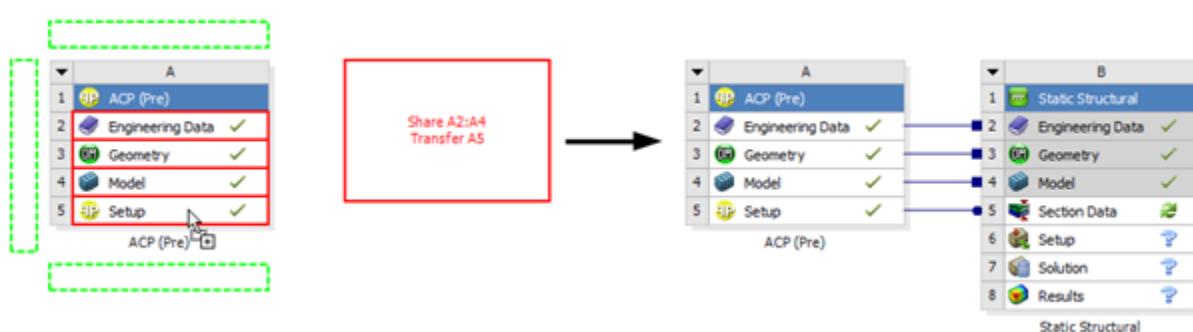
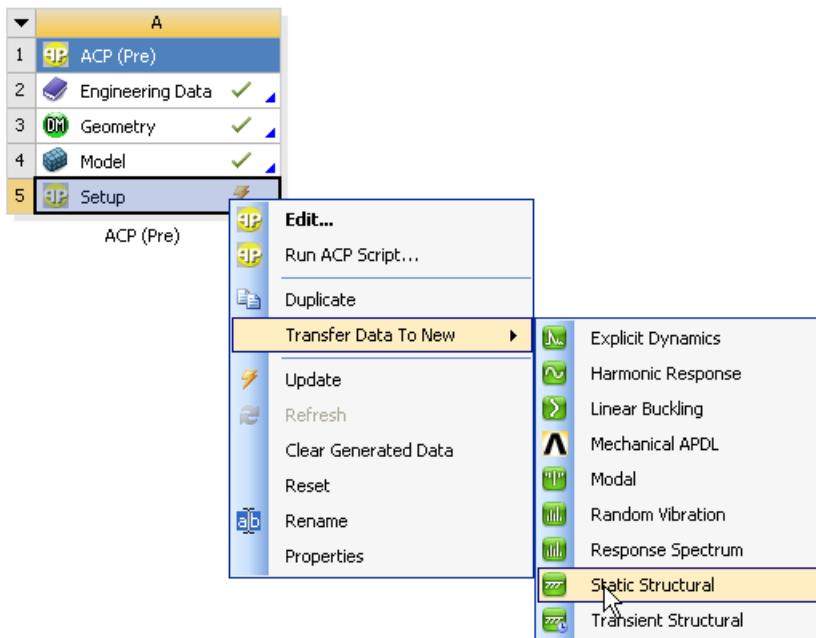
Figure 2.8: Connecting a Static Structural Analysis to ACP (Pre) with a drag-and-drop operation

Figure 2.9: Transferring an ACP Setup to an Analysis System through the context menu

2.3.3. Post-processing

2.3.3.1. Adding an ACP (Post) Component to the Project

The ACP (Post) component can be linked with one or several solutions and allows post-processing of composite structures. Because ACP (Post) is linked with the Engineering Data, Geometry and the Model of the ACP (Pre) component, the composite definitions (Section Data) are transferred automatically to ACP (Post).

As before, the ACP (Post) component can be added to the project by a drag and drop operation or by using the "Transfer Data to New" option in the context menu of the analysis component.

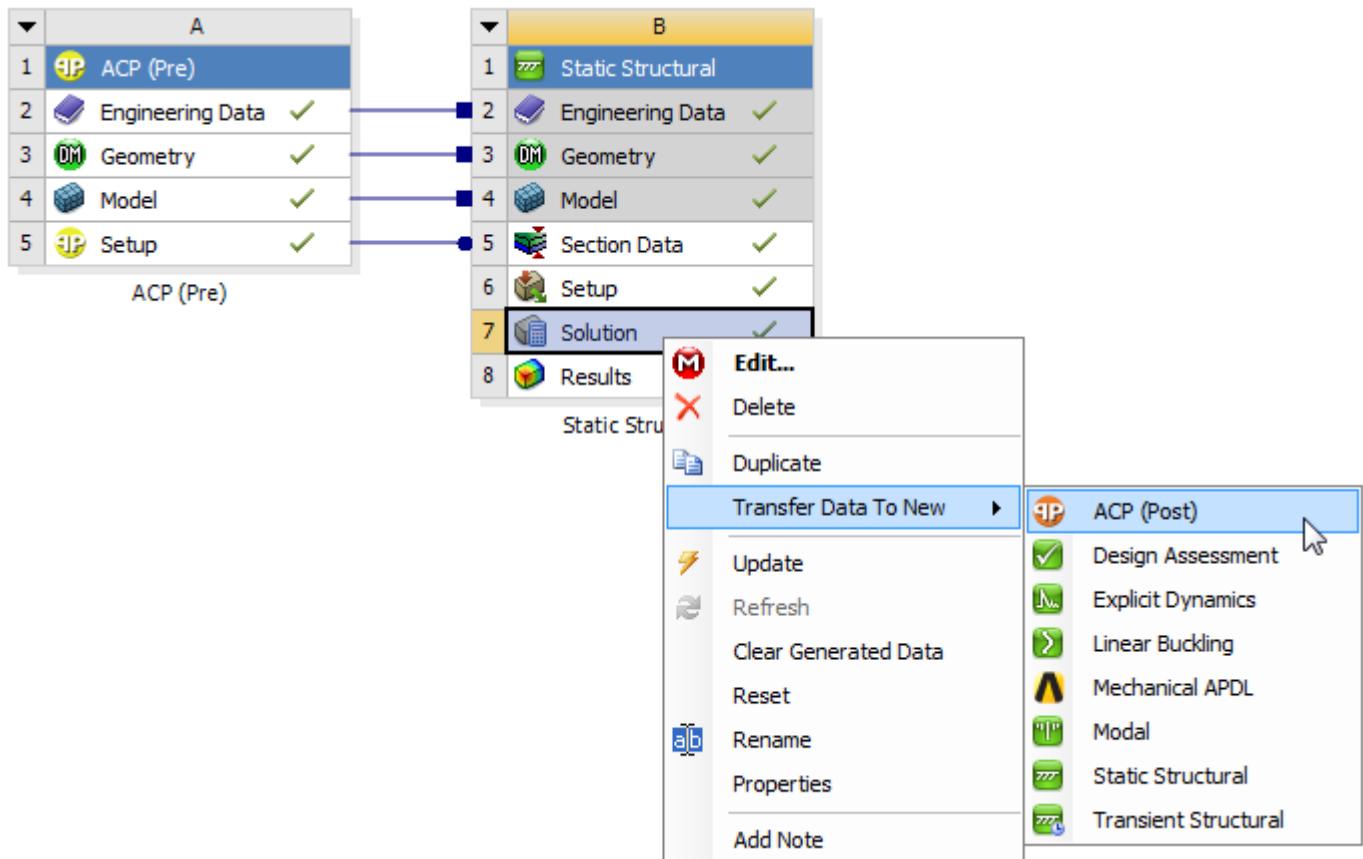
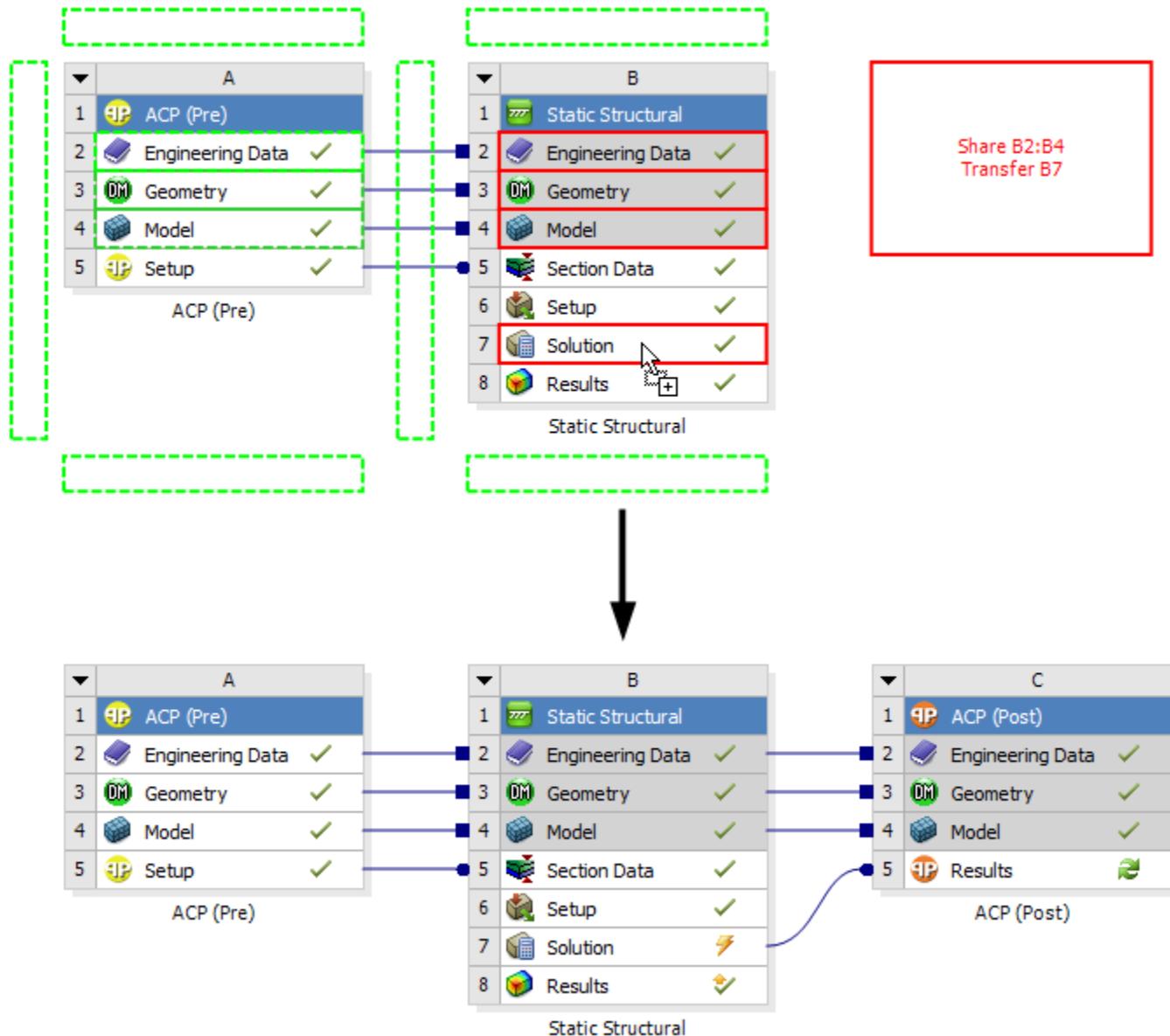
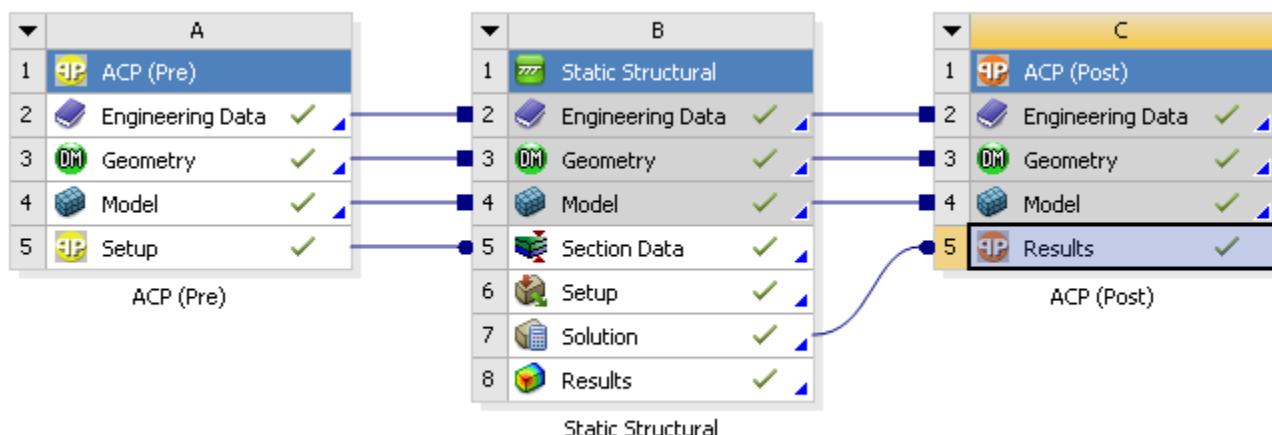
Figure 2.10: Adding ACP (Post) component by "Transfer Data..." option

Figure 2.11: Adding ACP (Post) by drag and drop operation

The complete composite shell model is now ready to be analyzed in ACP (Post).

Figure 2.12: Complete composite shell analysis model

2.4. Analysis of a Composite Solid Model

In the case of thick composites, the layered shell theory can cause significant errors in the obtained results. In some cases, it is necessary to work with 3D models - also referred to as Solid Models. ACP has the unique feature to generate layered solid models based on the shell layup definitions. ACP generates layered solid elements based on the shell mesh and the ACP Composite Definitions thus representing one-to-one the composite part. Drop-offs, staggering and tapering are also considered. In addition, the Solid Model extrusion allows to define extrusion directions and boundary curves.

In this section, the workflow of modeling a composite solid is outlined as it differs to some extent from shell modeling. Selected steps that differ from shell modeling are explained in more detail below and highlighted with a link.

- **Pre-processing**

- Add ACP (Pre) component to the project
- Define Engineering Data
- Import or construct Geometry
- Open the Model and
 - Define Named Selections/Element Sets
 - Generate Mesh
- Open ACP (Pre) and
 - Define Fabric
 - Define Rosettes and Oriented Element Sets
 - Create Modeling Plies
 - Create Solid Model

- **Workbench Analysis System**

- Choose an Analysis System (WB Mechanical/Mechanical APDL)
- Add Analysis System to the project
- Add other systems
- Open the Analysis System and
 - Define Analysis Settings
 - Define Boundary Conditions
- Solve model

- **Post-processing**

- Post-process complete assembly
- Add ACP (Post) component to the project
- Open ACP (Post) and run the post-processing for composite parts

2.4.1. Pre-processing

Creating a Solid Model

The generation of a layered solid element model has to be configured in ACP with the Solid Model feature. See the usage references for details on [Solid Models](#).

In the ACP solid model export settings, the user has to set an individual NUMOFF to avoid node and element numbering conflicts between multiple models (see section [Solid Model Properties– Export](#) for more details). Further, it is recommended to use homogenized drop-off elements with a global drop-off material (for more information see [Drop-Off Element Handling](#)).

Element sets and edge sets can be transferred from ACP (Pre) to the Static Structural component where they appear as named selections. Named Selections from the Mechanical Model are also transferred. This aids the definition of boundary conditions (See section [Solid Model Properties– Export](#) for more information).

2.4.2. Workbench Analysis System

Choice of Analysis System

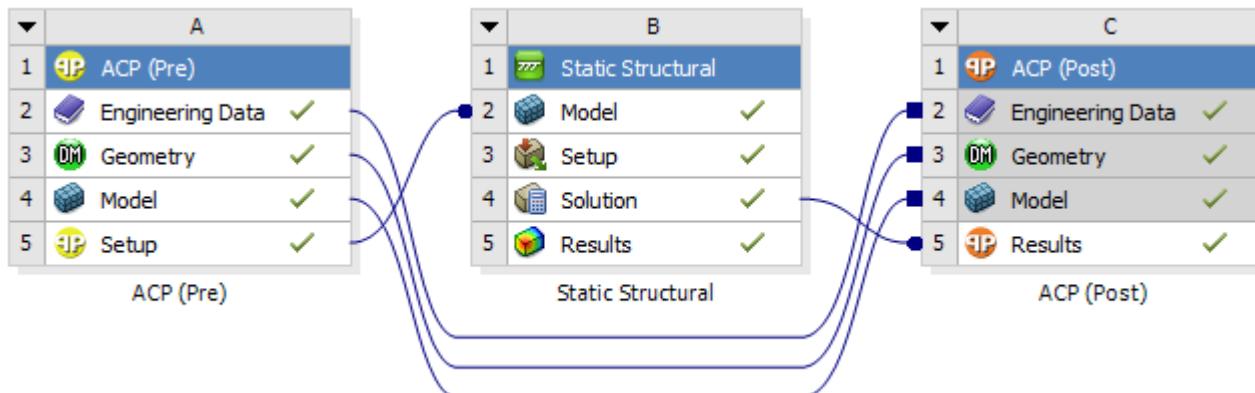
Within ANSYS workbench, there are two ways to analyze composite solid models. On the one hand the analysis can be done in Workbench Mechanical, on the other hand it can be carried out in Mechanical APDL (ANSYS Classic). The functionality is identical, the user interface is very different however. Alternatively, the composite solid models can be exported from ACP for processing outside of ANSYS.

Analysis with Workbench Mechanical

The composite layered solid element model appears in Workbench Mechanical as a meshed body. Any other bodies in the ACP (Pre) component are not carried forward. The user can define loads, boundary

conditions and connections to other parts in the usual Workbench Mechanical fashion. An example of such a solid model workflow is shown below:

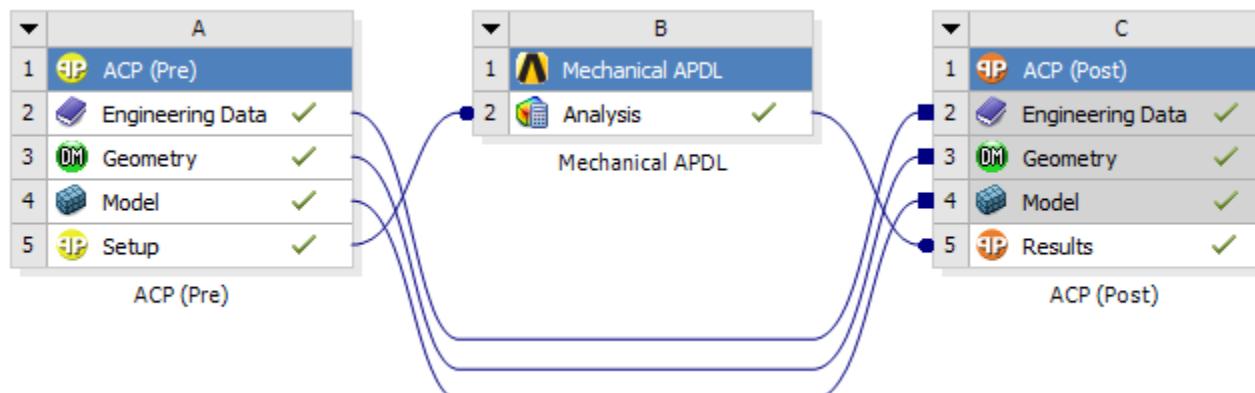
Figure 2.13: Workbench workflow for composite solid modeling with Workbench Mechanical



Analysis with Mechanical APDL

A further option is to link solid models to a Mechanical APDL where the boundary conditions and loads are defined. Typically, an APDL script is used to set boundary conditions and analysis settings for a workflow with Mechanical APDL. An example of a solid model workflow with Mechanical APDL is shown below:

Figure 2.14: Workbench workflow for composite solid modeling with Mechanical APDL



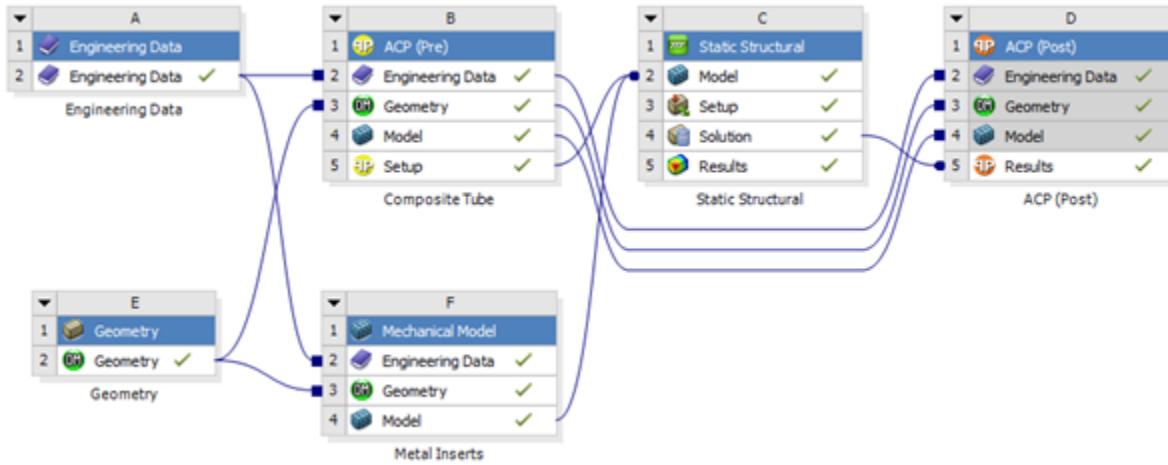
Adding an ACP (Pre) Component to the Project

The solid modeling workflow in ANSYS allows the assembly of many pre-processing components into one Analysis System. In some cases, it may be desirable to analyze thick-walled composites in isolation but, often enough, it is of interest to see the interaction between multiple bodies. The connection procedure is explained with the help of two examples for both analysis methods (WB Mechanical and Mechanical APDL).

Link with Workbench Mechanical

The procedure for building an analysis model is illustrated with a Static Structural Analysis System as an example. A composite tube connected to two metal inserts is subjected to torsion. A project schematic is shown below:

Figure 2.15: Analysis of a composite tube with metal inserts modeled with Workbench Mechanical

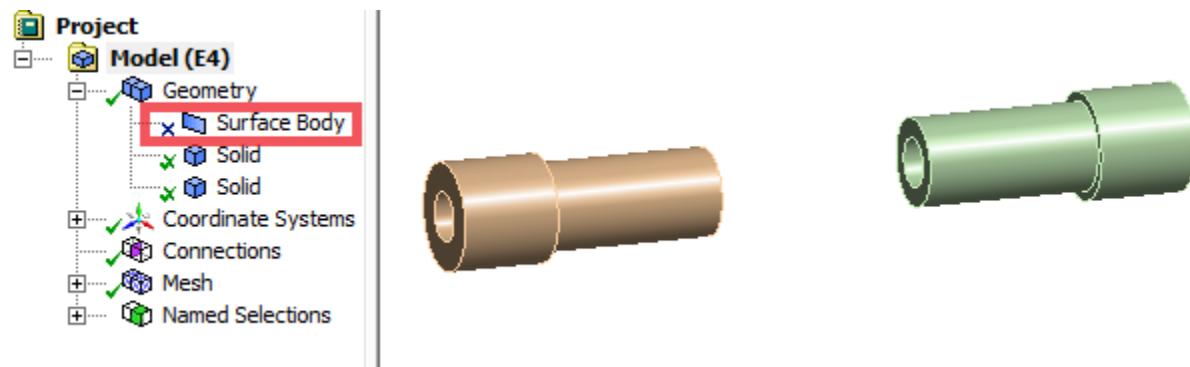
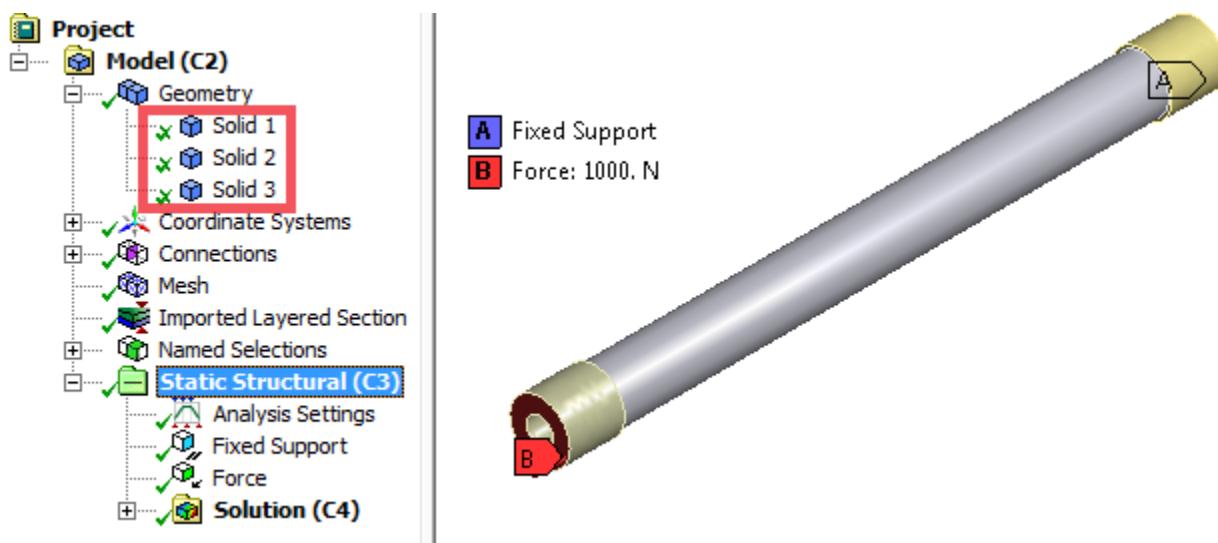


The sequence of connecting both models to the Static Structural component is important: First, a stand-alone Static Structural Analysis System is dragged into the project schematic. Secondly, the ACP (Pre) Setup cell is dragged into the Model cell of the Static Structural analysis. The Engineering Data and Geometry cells disappear as a result of the connection. Once the ACP Setup has been connected multiple other components can be attached to the Model component of the Static Structural analysis.

Figure 2.16: Connecting the ACP solid model to the Static Structural component.



In the case of this example, the geometry consists of one shell and two metal inserts. The link between ACP (Pre) (B5) and the Static Structural (C2) only transfers the generated layered solid element model. The link between the Mechanical Model (F4) and the Static Structural (C2) transfers all active bodies. As such, the shell geometry has to be suppressed. Consequently, all three parts appear as solid bodies in the Static Structural (C2) component. The connections, boundary conditions and all other pre-processing definitions can be defined in the Setup (C3) in the usual fashion. The global solution can be determined and analyzed in Workbench Mechanical while the composite component can be analyzed in detail in ACP (Post) (D5).

Figure 2.17: Suppressed Shell in Mechanical Model**Figure 2.18: Assembly of composite and metal solids**

Link with Mechanical APDL

Two composite components serve as an example for the Mechanical APDL workflow procedure - a plate and a t-joint. A project schematic is shown below:

Figure 2.19: Analysis of composite plate and t-joint modeled with Mechanical APDL

The sequence of connecting the system is to always connect an ACP (Pre) component first. In this case, it is not important because both inputs are ACP (Pre) solid models.

In the Mechanical APDL component the boundary conditions, loads and all other pre-processing definitions can be defined through APDL macros. These macros can be linked with the Mechanical APDL cell which will be integrated in the automatic update functionality of Workbench. A macro file can be added to the component through the right click menu (see figure below). *Add Input File...* appends the APDL running sequence with an additional macro. Check the order of the files of the Mechanical APDL component. The macros should be listed after the Solid Model Process Setup file(s).

Figure 2.20: Add reference file

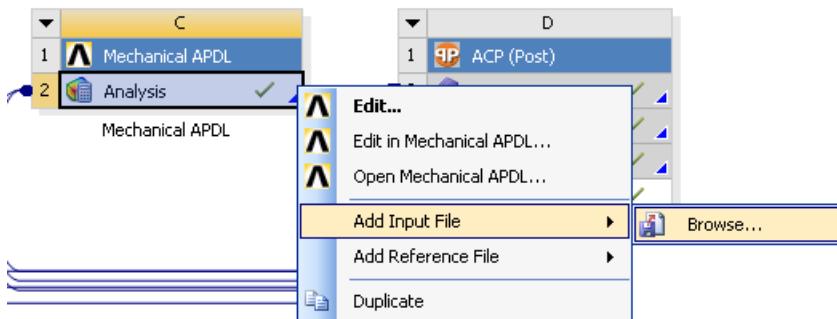


Figure 2.21: List of used file and their order in Mechanical APDL

Outline of Schematic C2: Analysis		
	A	B
1		Step
2	Launch ANSYS	1
3	Process "Setup"	2
4	Process "Setup"	3
5	Process "run_T_joint.mac"	4

Multiple Parts (Adding other Systems)

It is possible to add multiple components to one Analysis System. Two composite parts or a composite part connected to two isotropic parts, for example.

2.4.3. Post-processing

Global Post-processing

In general, the global solution of all parts can be viewed in Workbench Mechanical or in Mechanical APDL. Analyzing the results of a multi-part assembly is not possible in ACP (Post) for Solid Models.

Adding an ACP (Post) component to the Project

The post-processing functionality for Solid Models of ACP allows the mapping of ply wise results on to the reference surface of the solid model. This ensures that also failures occurring inside the laminate can be observed and investigated.

The connection between an Analysis System and ACP (Post) always requires the same two steps regardless of whether the Analysis System is Workbench Mechanical or Mechanical APDL. First of all, the ACP (Post)

system has to be associated with an ACP (Pre) system. Subsequently, a solution from an Analysis System can be linked with the ACP (Post) Results. There are other ways of connecting ACP (Post) with an Analysis System yet they all fall short when it comes to linking the Analysis Solution cell with the ACP (Post) Results cell. This is a known limitation of the Workbench integration.

Figure 2.22: Step 1: Drag-and-drop an ACP (Post) system on to an ACP (Pre) system

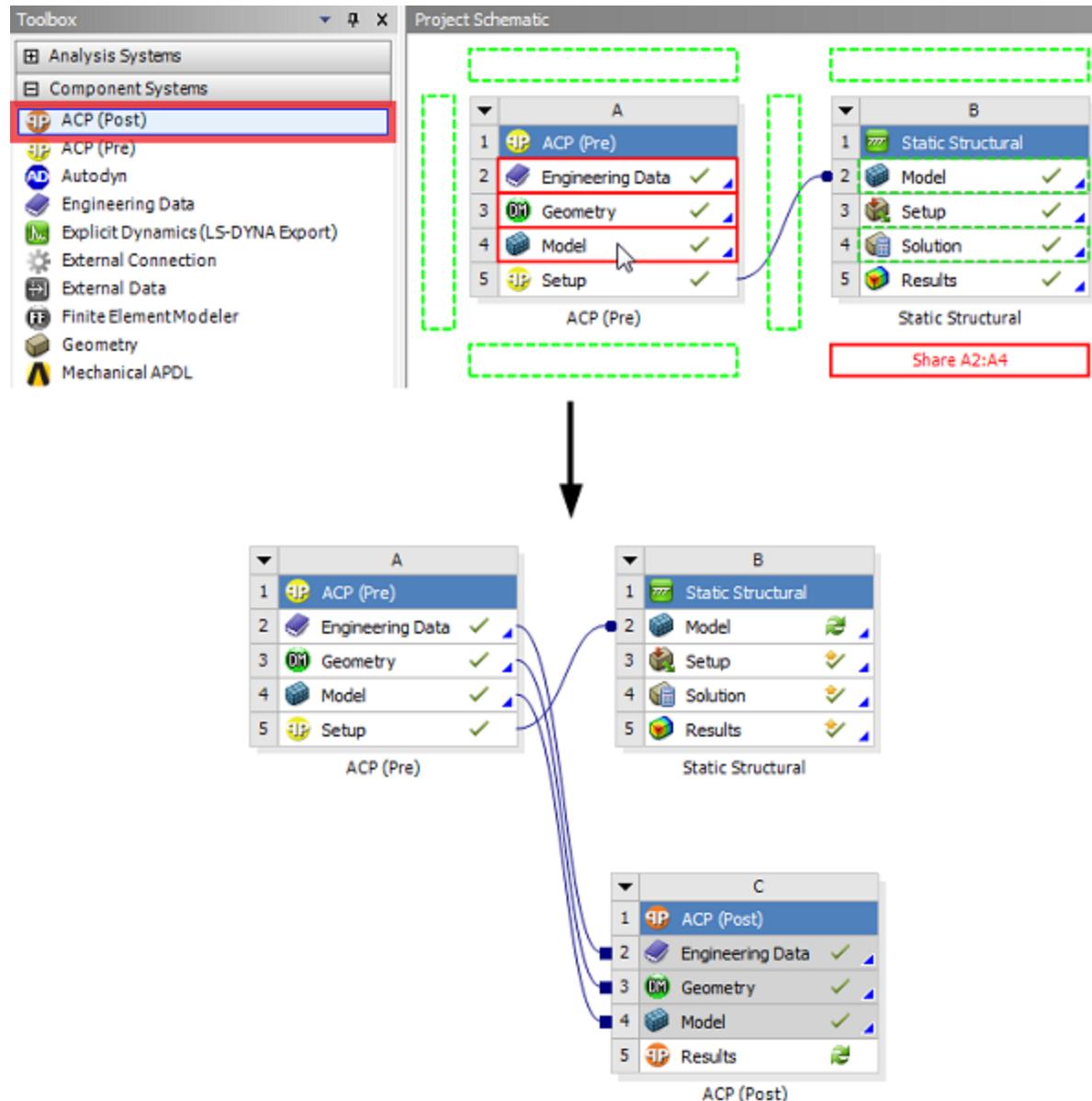
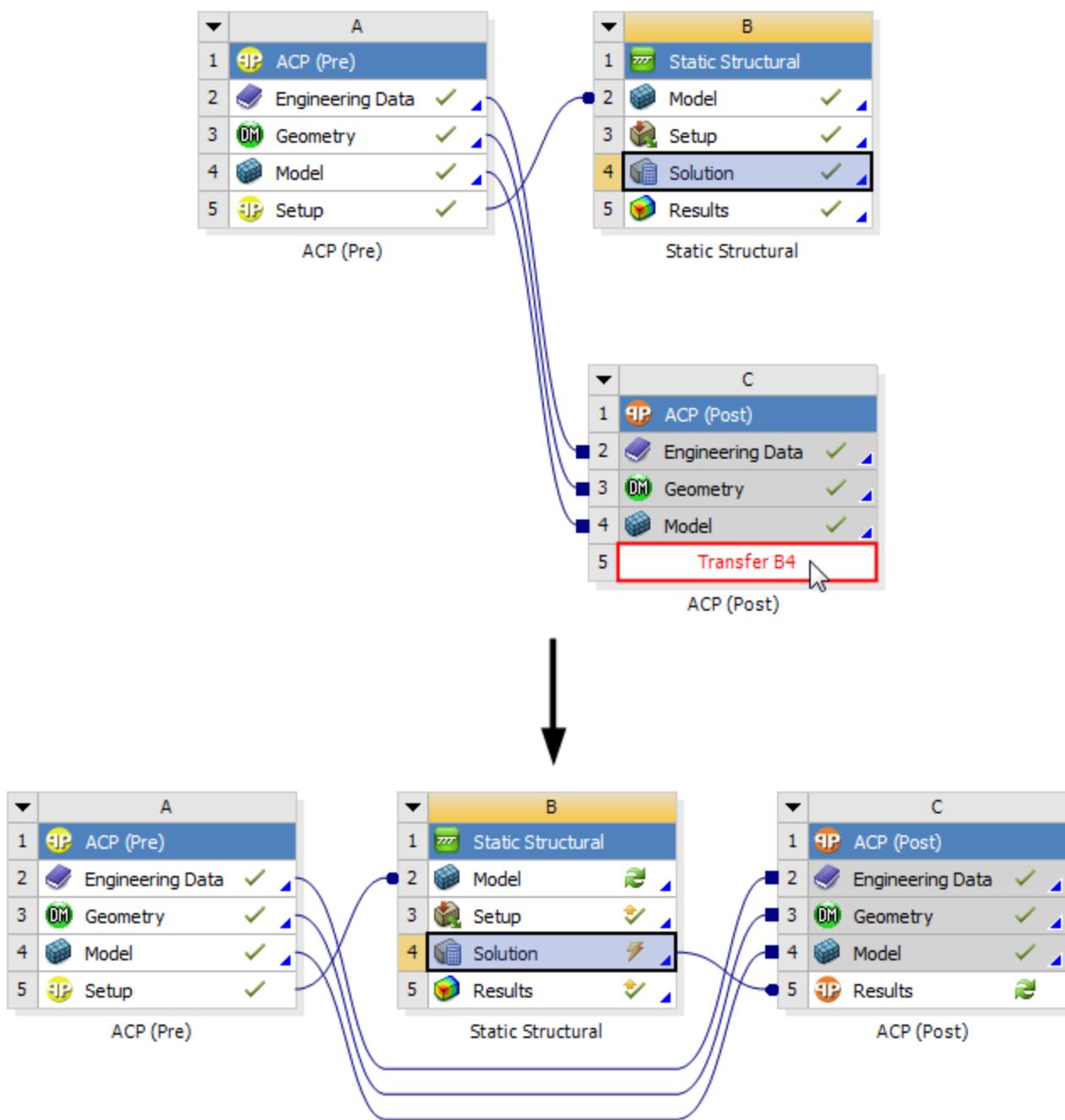


Figure 2.23: Step 2: Drag-and-drop the Static Structural Solution cell onto the ACP (Post) Results cell



When multiple ACP (Pre) solid model systems are linked to an Analysis System then every ACP (Pre) system has to have a corresponding ACP (Post) system. See the [Analysis of composite plate and t-joint modeled with Mechanical APDL](#) for an example workflow.

2.5. WB Workflow Examples

The ACP Module can be used from basic analysis to complex load-cases and analyses systems. Some examples are given here.

2.5.1. Single Analysis Extended

Analyses which require previous results like **linear buckling** or **pre-stress modal** analyses, are also supported and can be transferred from the solution:

Figure 2.24: Single Analysis with ACP (Pre) and ACP (Post)

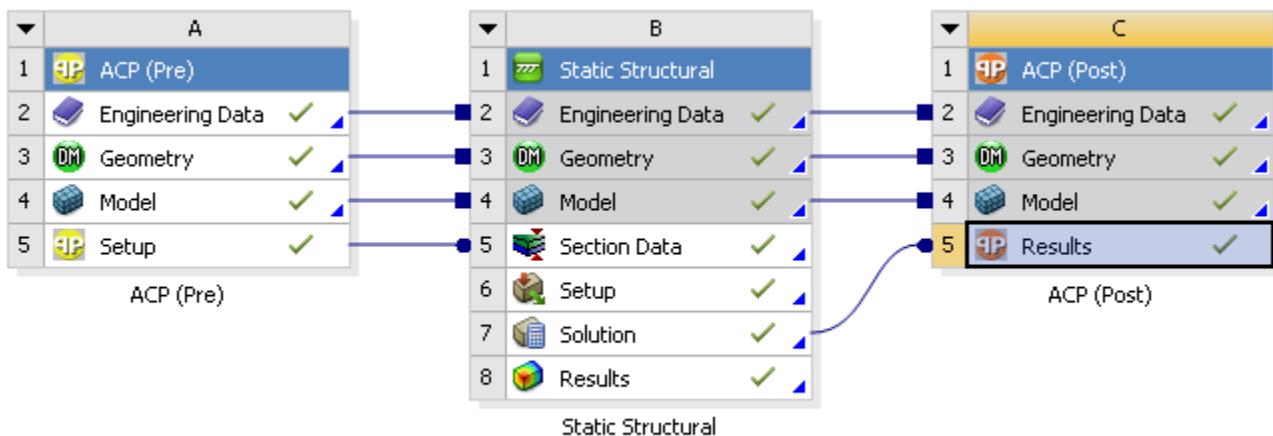
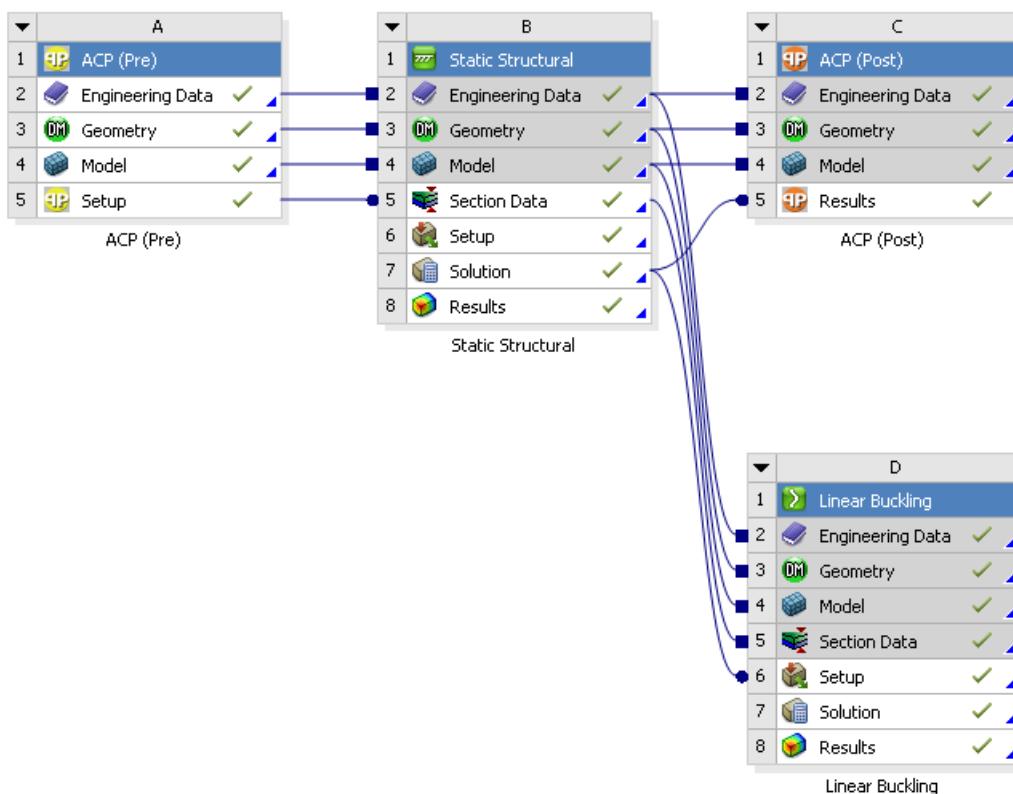


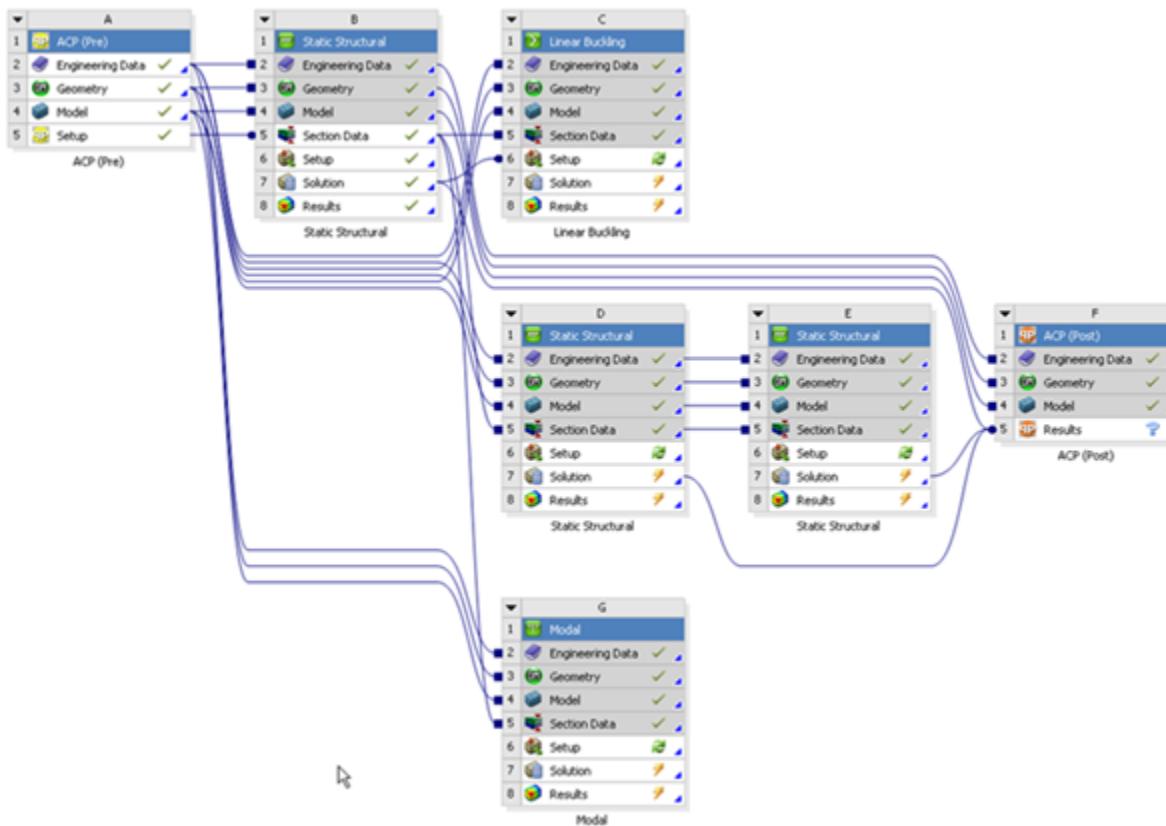
Figure 2.25: Project Schematic of a Linear Buckling Analysis



2.5.2. Multiple Load-cases and Analyses

Complex workflows with multiple load-cases and/or analyses are defined exactly like standard analyses. In most of the cases, the links to share the data are set automatically by Workbench. But some links must be manually added. In the following example, the links from Solution of analyses D and E to ACP (Post) are added manually.

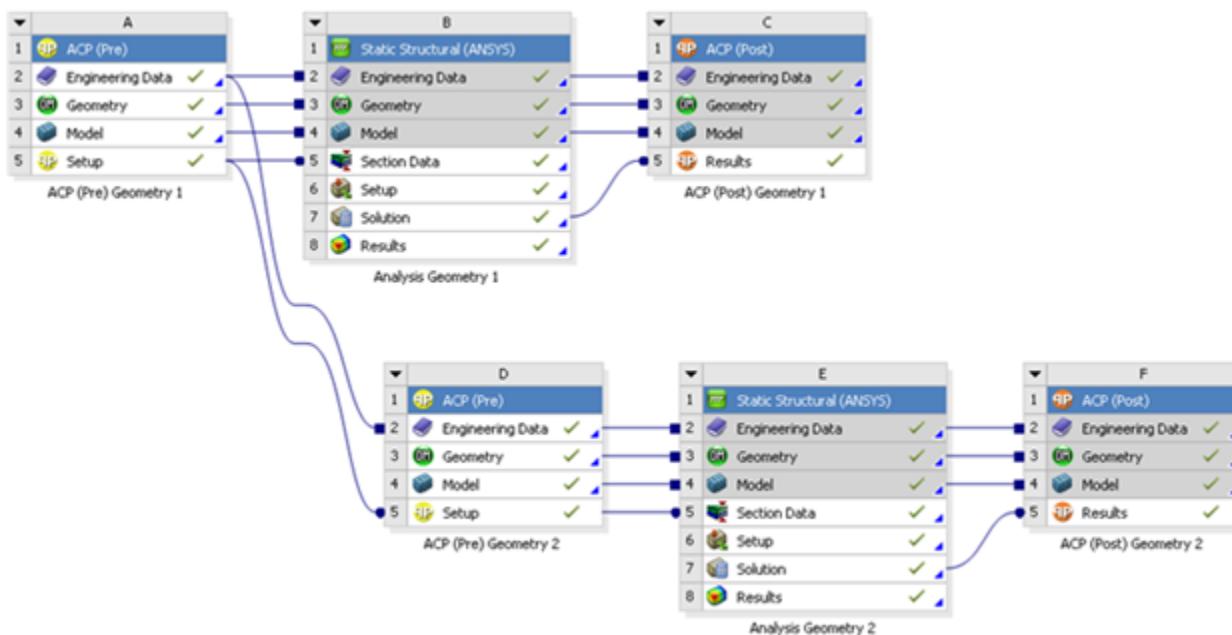
Figure 2.26: Multiple Load-Cases and Analyses



2.5.3. Shared Composite Definition for Different Models

The ACP-Pre Setup can be shared across multiple models. This means the same composite lay-up can be applied to models of different geometries and meshes. This functionality can be used in design studies where the composite definition remains the same but different geometries are evaluated. Sub-modeling is another situation where this functionality can be used. In a scenario where two models are to share the same ACP-Pre Setup, one setup cell is simply dragged on to the other. ACP tries to map as much information onto the second model as possible. It is advised to use common Named Selections as well as the same Engineering Data. The figure below shows an example of a shared ACP-Pre setup.

Figure 2.27: Two Analyses share the same ACP (Pre) Setup



2.6. Stand-Alone Operation

ACP can also be used as stand-alone application decoupled from the Workbench Project logic. In this case some operations, which are normally handled by the Workbench Addin, have to be performed manually.

2.6.1. Starting ACP

Windows

The easiest way to start ACP interactively on Windows is to use the button provided in the Start Menu at

```
Programs\ANSYS XX.X\ACP\ACP XX.X
```

By default the executable file is located at

```
C:\Program Files\ANSYS Inc\vXXX\ACP\ACP.exe
```

Linux

On a standard installation, ACP can then be started with

```
/ansys_inc/vXXX/ACP/ACP.sh
```

2.6.2. Command line options and batch mode

ACP can also be used on a command line level. The general usage of the ACP start script is:

```
ACP.exe [option] [script.py]
```

The supported command line options are:

--help (-h)	Produce this help message.
-------------	----------------------------

--batch (-b)	Run ACP in batch mode. There are three batch mode options available. For the last two, the program exits at the end of the script run.
--batch=0	No batch mode. ACP starts in normal stand-alone mode.(Default)
--batch=1	Batch mode with no graphical functionality.
--batch=2	Batch mode with graphical functionality, i.e. capturing Scene snapshots.
--debug (-d)	Run ACP in debug mode. In debug mode detailed runtime information is printed to the command window.
--num_threads (-t)	Number of threads to be used (ACP post-processing; does not affect the ANSYS solver options).
--logfile (-o)	File to be used to write the log messages.
FILE	Execute the given Python script FILE on Start-Up.

Example:

```
C:\Program Files\ANSYS Inc\vXXX\ACP\xx.x\ACP.exe --batch myACPScript.py
```

2.6.3. Workflow in Stand Alone Operation

The difference between Stand-Alone operation and WB Integration is that the several steps or operations have to be performed manually. An overview of these steps and more information is shown below.

- Generate the ANSYS input file in Mechanical APDL, including the loads and boundary conditions (*.inp, *.dat or *.cdb).
- Start ACP
- Import the ANSYS Model in ACP.
- Define the Materials, or copy the materials from the ACP Material Databank.
- Create laminate sequences as usual.
- Update the model after any change in the ACP definition (layup definitions) or a change of the input model.
- Send the model to the ANSYS solver (Solve Current Model) or export the new analysis file (Save Analysis Model)
- Switch between ACP Pre and ACP Post.
- Import the results.
- Run the composite post-processing.
- Save the Composite Definitions in ACP.

Generate the ANSYS input file

The first step is to generate the ANSYS input file for ACP.

From Workbench

If the analysis is defined in Workbench, there are two ways to create an input file used for ACP (Pre):

- In the Mechanical application, select the analysis and go to **Tools->Write an Input file...** to write a *.dat or a *.inp file.

Figure 2.28: Write Input File...

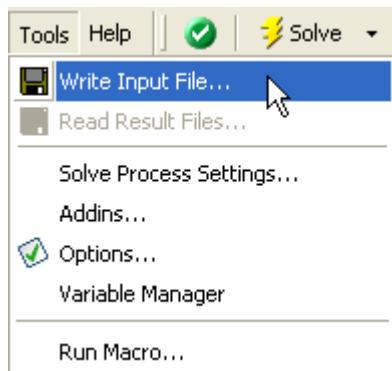
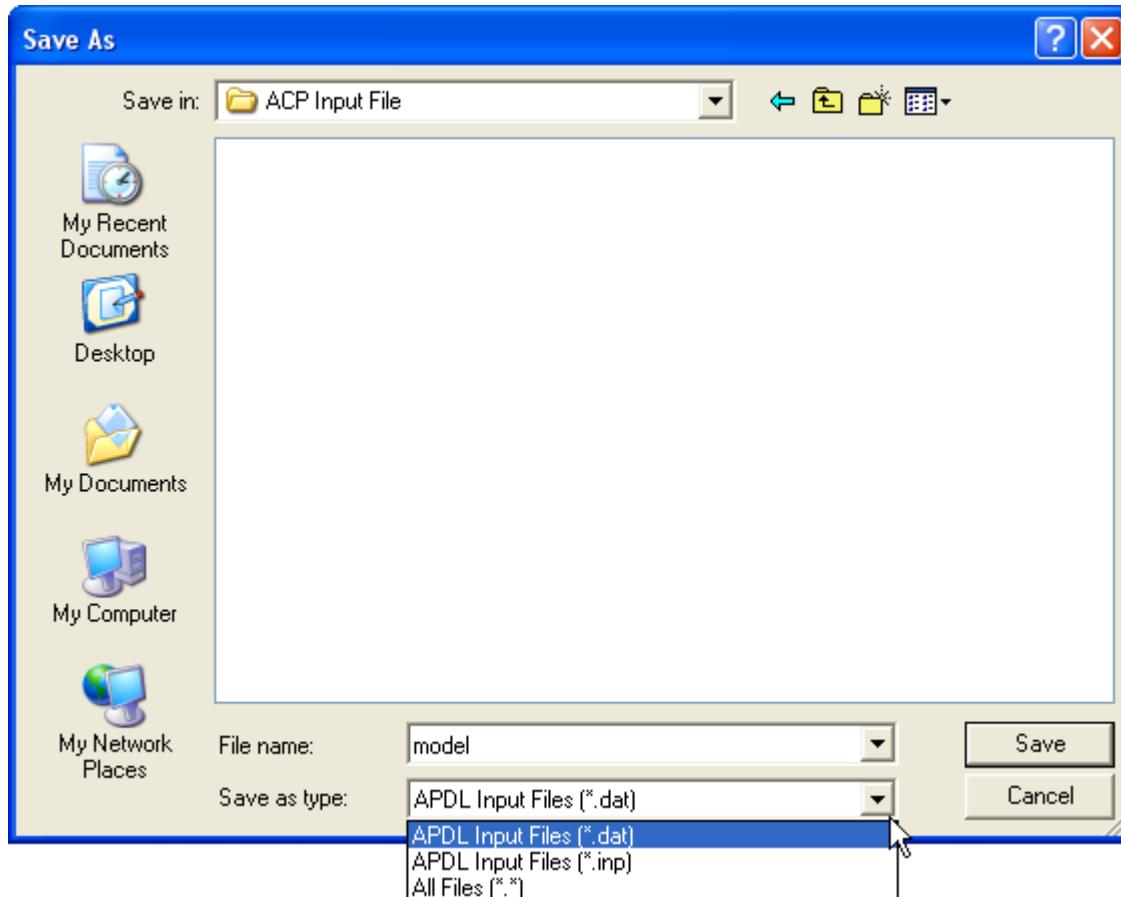


Figure 2.29: Choose Format

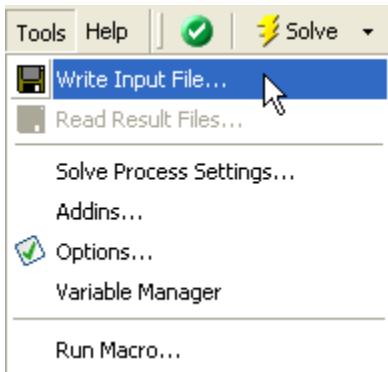


- In the Project Schematic update the Setup status. A ds.dat file is written in the folder SYS-X/MECH. This file can also be used as Pre-processing Model in ACP.

From Mechanical APDL

In Mechanical APDL, use the command CDWRITE to write a *.cdb file, which can be used as Pre-processing Model in ACP.

The command 'cdwrite,db,file,cdb' is a typical example to generate this file.

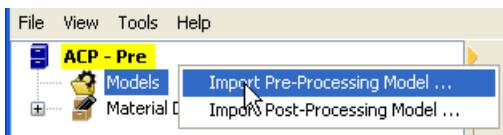


Import the ANSYS Model in ACP

ACP has interfaces for the *.dat, *.inp and *.cdb file format. There are two different but equivalent ways to import the input file:

- From the File menu (refer to [Menu](#))
- With the right click menu of the Model in the tree view

Figure 2.30: Import ANSYS Model



Define the Materials

Define the different materials used in the lay-up definitions. For more details, refer to [Material Data](#).

The material defined in the Databank can also be used through a Copy/Paste operation. For more information about the material databank, refer to [Databank](#).

Create Laminate Sequences

Use the different features of ACP to define the laminate sequences. Refer to [Usage Reference](#) for more details on the different features or to [Composite Model techniques](#) for modeling techniques.

Update

After any modification in the input model, the input file must be reloaded (see [Model](#)).

After this operation or any change in the ACP definition, the ACP Database must be updated. Use the button in the Toolbar (see [Toolbar](#)).

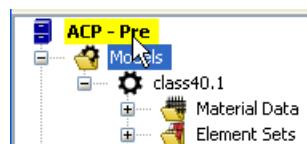
Solve or Export The New Analysis

Send the model to ANSYS Solver (**Solve Current Model**) or export the new analysis file (**Save Analysis Model**) through the drop-down menu of the Model (see [Model](#)). In the first case, the ANSYS Solver is automatically started in batch mode.

Switch between ACP Pre and ACP Post

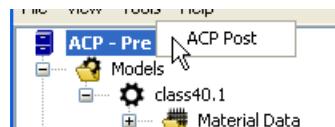
Click on the root of the object tree to switch between ACP Pre and ACP Post.

Figure 2.31: Switch with a simple click



or use the right mouse button.

Figure 2.32: Switch with a drop-down Menu



Import the Results

In the ACP Post mode, solutions can be imported to evaluate the strength of the composite structure. In the Tree view, import the result files through the drop-down menu on Solution. Refer to [Solutions](#) for more details.

Run the Composite Post-Processing

Use the feature [Definitions](#) to define which results are evaluated in the post-processing. Plot these values through the [Scenes](#) for representation on the geometry or use [Sampling Elements](#) for representation through the lay-up.

Save the Composite Definitions in ACP

Save the ACP definition through the drop-down Menu of Model [Model](#) or through the menu File [Menu](#).

2.7. Migration from Previous Versions

All versions between 14.0 and the latest release are compatible.

[2.7.1. Migrate ACP Projects from v14.5 to 15.0](#)

[2.7.2. Migrate ACP Projects from v14.0 to 14.5](#)

[2.7.3. Migrate ACP Projects from v13.0 to 14.0](#)

2.7.1. Migrate ACP Projects from v14.5 to 15.0

ACP v14.5 projects can be opened in v15.0 without any further modifications required. All data and definitions are transferred to the new system.

2.7.2. Migrate ACP Projects from v14.0 to 14.5

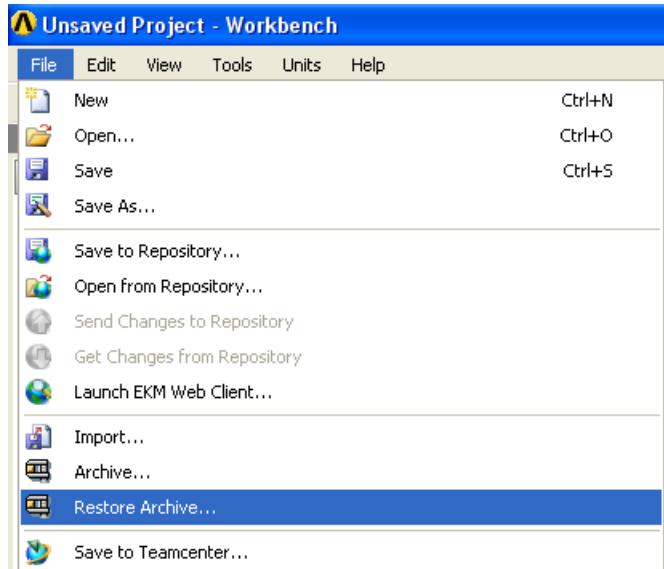
ACP v14.0 projects can be opened in v14.5 without any further modifications required. All data and definitions are transferred to the new system.

2.7.3. Migrate ACP Projects from v13.0 to 14.0

Old ACP projects (ACP Definitions) are not automatically upgraded and included in the project schematic of Workbench 14.0. Follow these steps to import the ACP Composite Definitions to Workbench 14.0:

1. Start Workbench 14.0 and restore the old archive

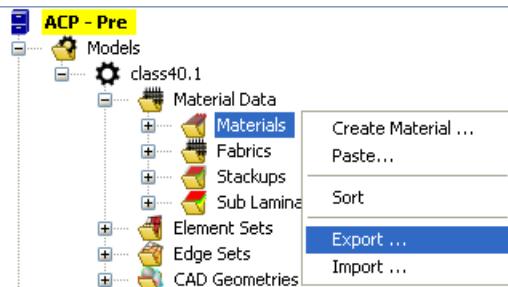
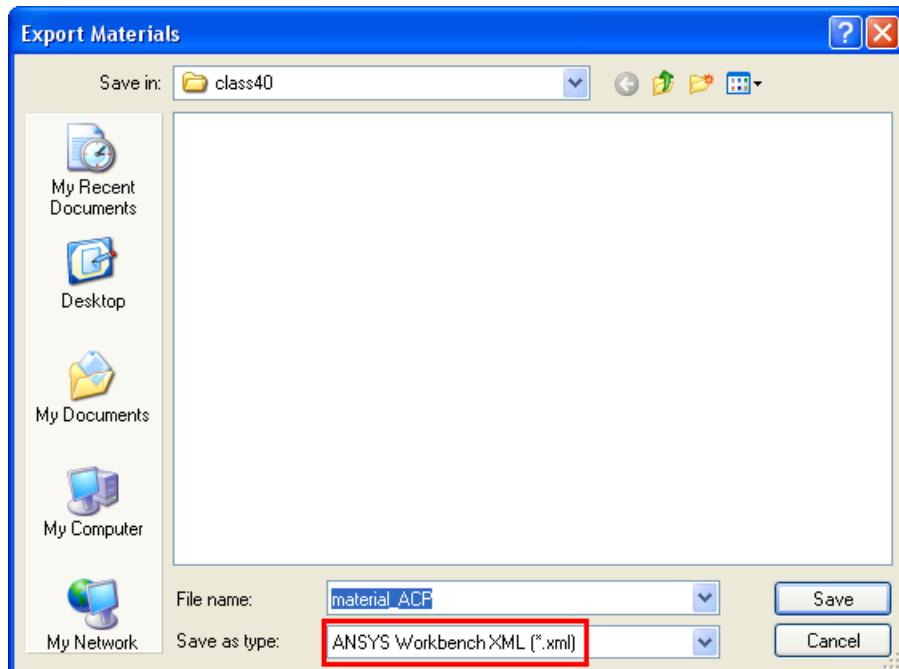
Figure 2.33: Restore Archive from v13 Project



2. Start ACP 14.0 in stand-alone mode and open the old ACP file which is located here:

- <project folder>\user_files\ACP\SYS_XY} folder (basic integration)
- <project folder>\user_files\ACP folder (advanced integration)

3. Update the model.
4. Export the Materials container as ANSYS Workbench XML file. Save it e.g. on the desktop.

Figure 2.34: Export the materials**Figure 2.35: Switch to ANSYS Workbench XML file format**

5. ACP will later convert the units of composite definitions automatically if a unit system is defined (normally done). However check the *Model* properties and define a unit system if needed.
6. Save the model e.g. on the desktop (Use *Save as...*).
7. Switch back to ANSYS Workbench project schematic.
8. Open the Engineering Data component of the **first analysis** and import the materials from the ANSYS Workbench XML file generated at point 4. Deactivate the filter option to list the new material properties. **This step has proven to be unstable.** Often it is easier to enter material properties manually into the Engineering Data component and skip to step 10.

Figure 2.36: Import materials

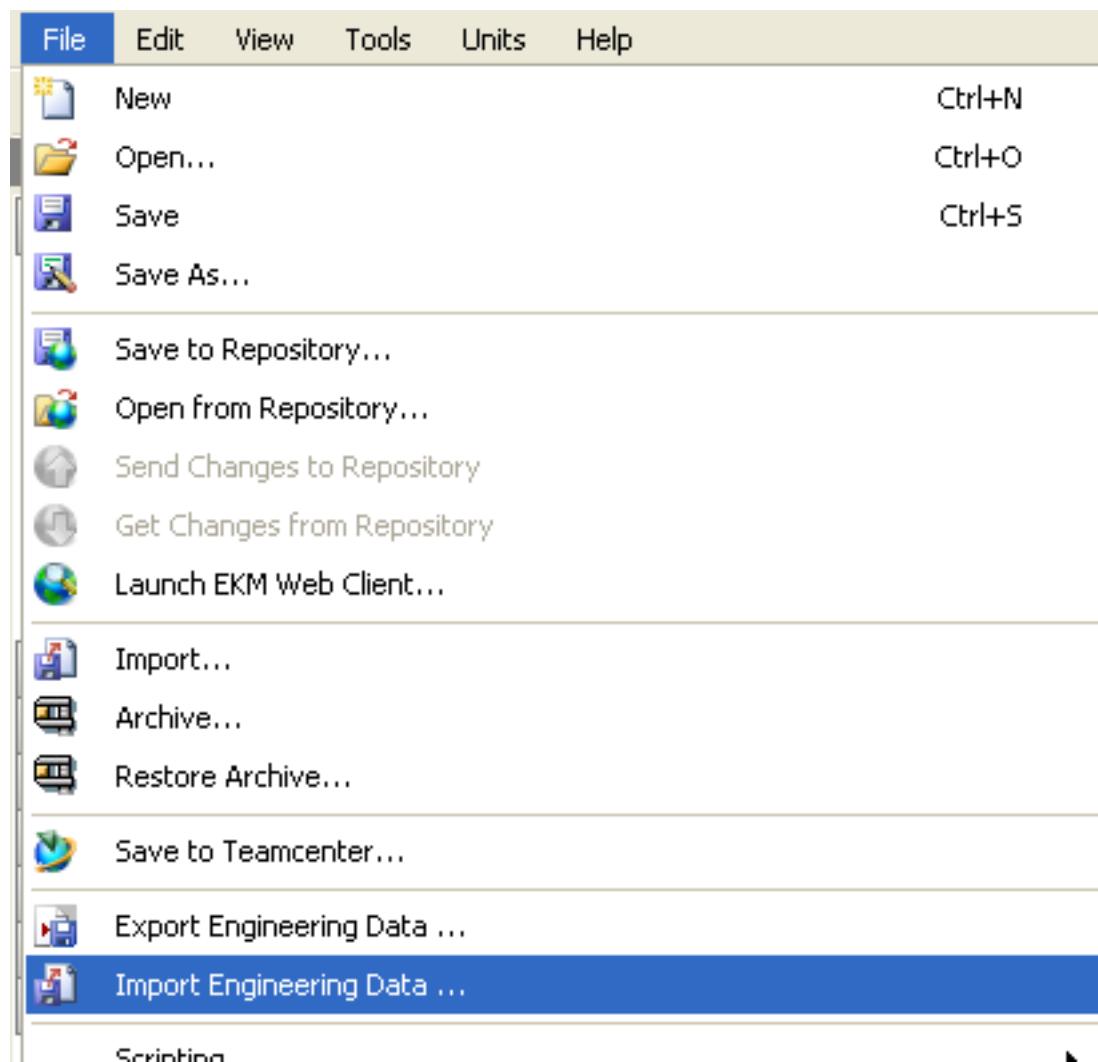
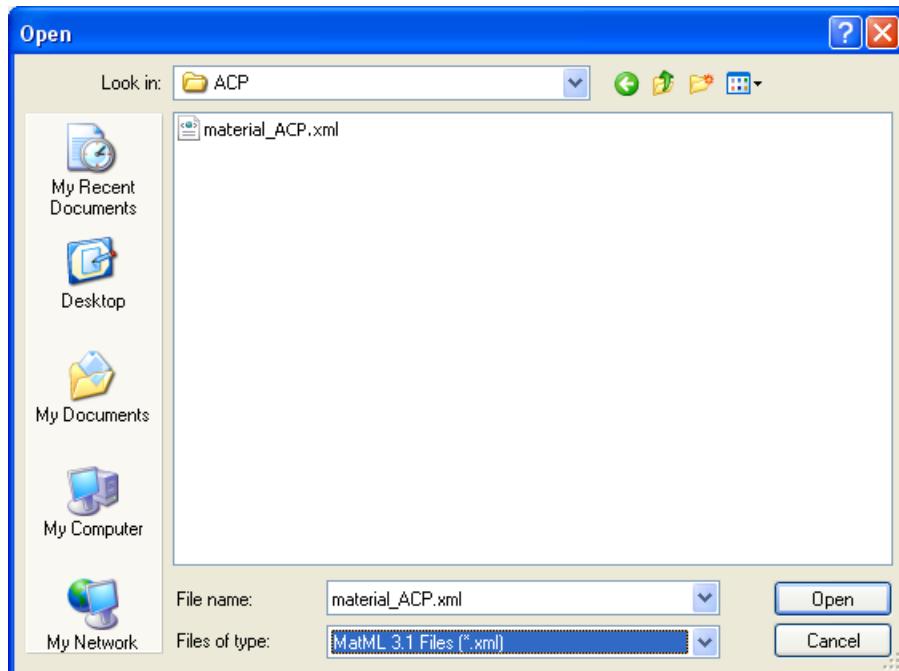
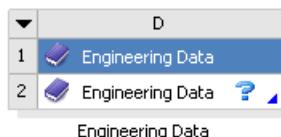


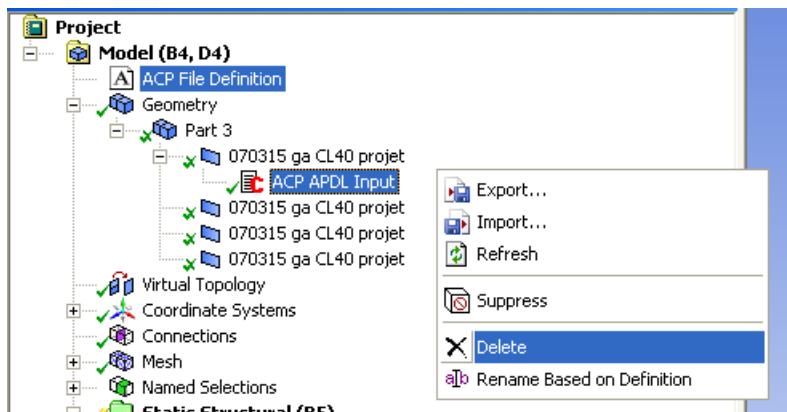
Figure 2.37: Select XML format**Figure 2.38: New materials**

Outline of Schematic, A2, B2, C2: Engineering Data			
	A	B	D
1	Contents of Engineering Data		Description
2	Material		
3	Structural Steel		Fatigue Data at zero mean stress comes from 1998 ASME BPV Code, Section 8, Div 2, Table 5-110.1
4	Corecell_A450		
5	E-Glas		
6	Corecell_A550		
*	Click here to add a new material		

- Check if the materials are completely defined. If the ED component has a question mark some material properties are missing or not defined and have to be entered manually.

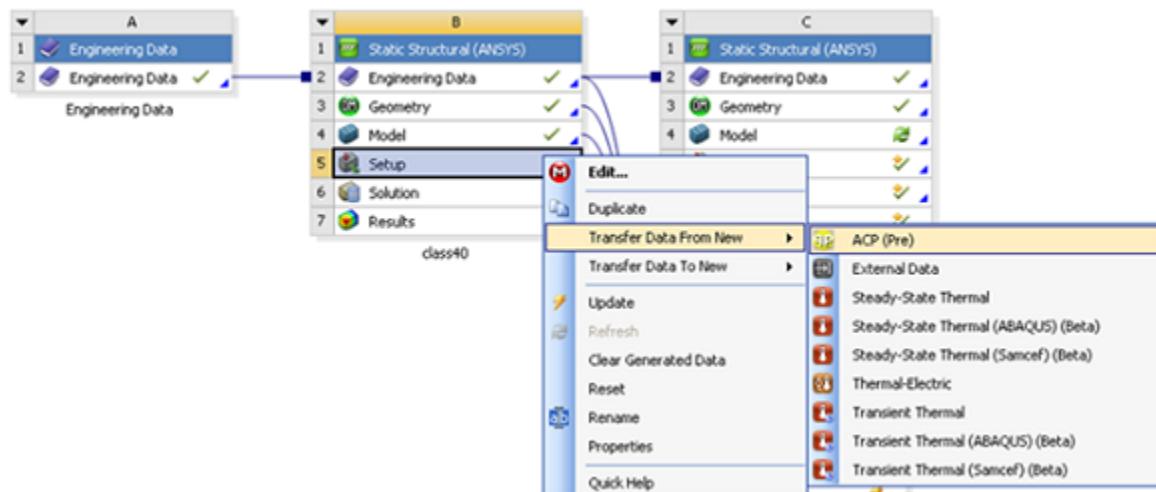
Figure 2.39: Engineering Data Box with question mark

- Update the *Model* cell of the first *Analysis Component*
- Open the Mechanical application. If the *Advanced Integration* of ACP 13.0 has been used, perform the next step. Otherwise skip it.
- Delete the *ACP File Definition* and the *ACP APDL Input* attached to the first active geometry.

Figure 2.40: Delete these two components

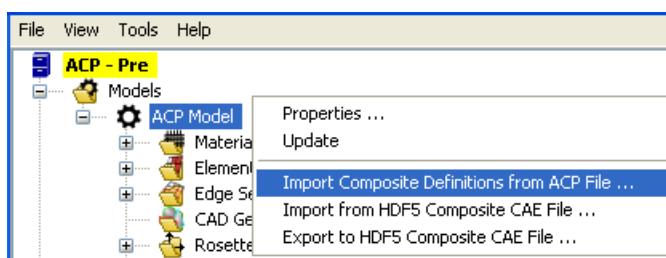
13. Write all *Named Selections* in upper case.

14. In the Project schematic of ANSYS Workbench select *Transfer Data From New* of the right click menu of the *Setup* cell of the first analysis and select ACP (Pre). A new ACP (Pre) cell is generated.

Figure 2.41: Create ACP (Pre) cell

15. Start ACP (Pre). It might be possible that a refresh or update is necessary.

16. Use the *Import Composite Definitions from ACP File...* functionality to load the model saved at step 6.

Figure 2.42: Import Composite Definitions from ACP File

17. Update the model and check the definitions.

The ACP Composite *Definitions* are now loaded in ACP 14 and automatically passed to the ANSYS Workbench 14 project. For post-processing add an ACP (Post) component to your analysis.

The *Import Composite Definitions from ACP File...* functionality can also be used to map *ACP Composite Definitions* from another model \ analysis to another.

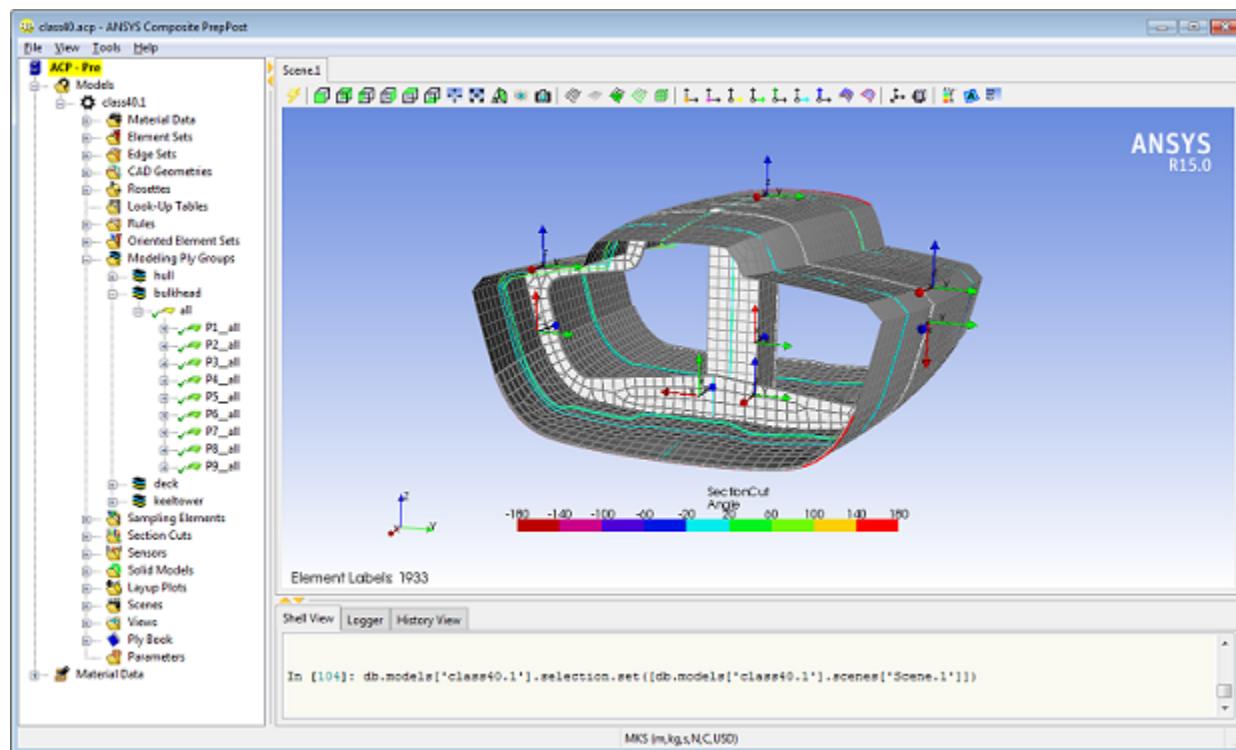
2.8. Graphical User Interface

The User Interface is split in different parts:

- Menu
- Scene and its **Toolbar**
- Tree View
- Shell View, History View, and the **Logger**

The section **Layout Modification** explains how the Scenes, Shell View, History View and Logger can be rearranged.

Figure 2.43: ANSYS Composite PrepPost GUI



2.8.1. Layout Modification

The user can modify the position of each View (managed by the perspective) with a drag and drop action. Select the View's header, and drag it. The future position is indicated by a red square and changes with the mouse position. Drop it when position is found.



Select the View that you want to reposition

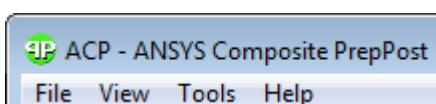


Drag and drop to the desired position

2.8.2. Menu

The menu contains 4 submenus:

Figure 2.44: ANSYS Composite PrepPost menu



2.8.2.1. File

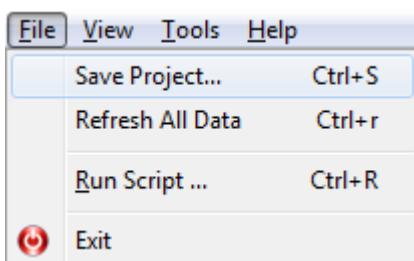
The *File* Menu differs in Workbench Integration and Stand-alone modes.

2.8.2.1.1. Workbench Integration

In Workbench Integration, the *File* Menu contains only 4 options:

- Save Project...: If the project was not already saved, specify the project name and location. It saves not only the ACP database, but the entire Workbench project. ACP will close. If it is an existing project, it will save the entire Workbench project.
- Refresh All Data: same as Refresh in the Workbench Schematic; it reloads the Model in ACP. If the model is not up-to-date in the Schematic, changes on the model (mesh, named selections) are not transferred to ACP. The update of the model must be made in the Workbench Project Schematic.
- Run Script... : Allows the launching of a Python script.
- Exit: Exit from ANSYS Composite PrepPost. The newly defined ACP Features are not deleted.

Figure 2.45: File Menu for Workbench Integration

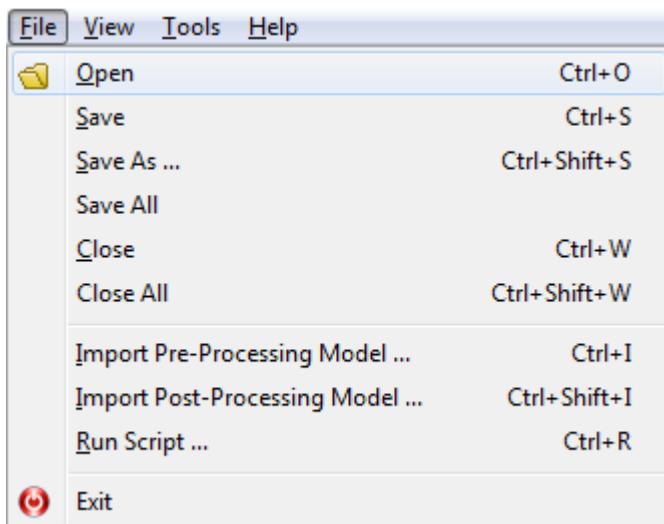


2.8.2.1.2. Stand-Alone

In stand-alone, the following actions are accessible:

- Open: Open an existing ACP Database
- Save: Save the active ACP Database
- Save as...: save the active Database
- Save All: save all opened Databases
- Close: Close the active ACP Database
- Close All: Close the opened ACP Database
- Import Pre-Processing Model...: Import an ANSYS model (mesh, materials, components,...) into ACP Pre.
- Import Post-Processing Model...: Import an ANSYS model (mesh, materials, components,...) into ACP Post for post-processing only.
- Run Script... : Allows the launching of a Python script.
- Exit: Exit from ANSYS Composite PrepPost. The newly defined ACP Features are not deleted.

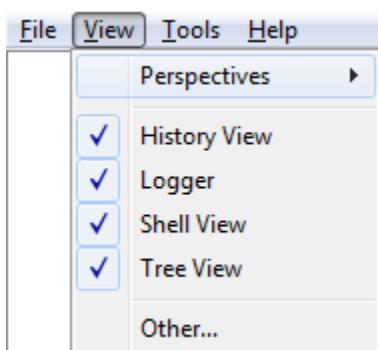
Figure 2.46: Stand-alone file menu



2.8.2.2. View

The layout of the GUI is managed through this Menu.

Figure 2.47: View Menu

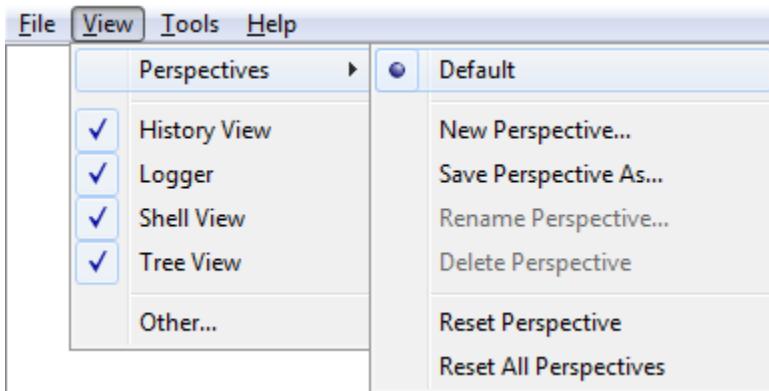


2.8.2.2.1. Perspectives

This submenu helps to manage the different layouts (perspectives):

- Perspective: Activate defined Perspective. The perspectives are listed in a drop-down menu.
- New Perspective: Create a new perspective. Define its name. The new perspective will be empty.
- Save Perspective As... : Save the actual layout into a new perspective.
- Rename Perspective... : Modify the name of the active perspective.
- Delete Perspective: Delete the active Perspective.
- Reset Perspective: The active Perspective is rest to its default setting. In most cases, it resets to an empty perspective.
- Reset All Perspectives: Reset all defined perspectives.

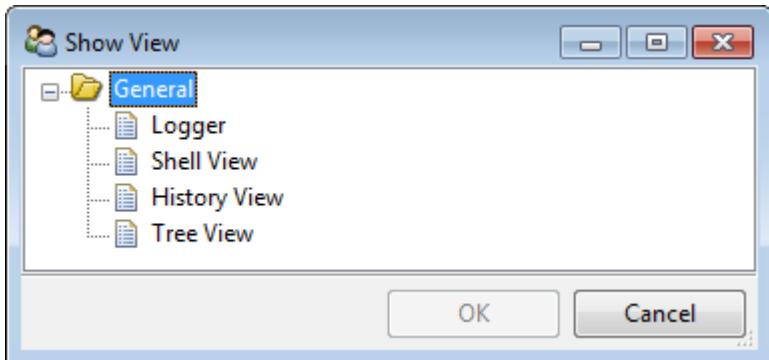
Figure 2.48: Perspective Submenu



2.8.2.2.2. View Manager and Other...

The different parts of the layout can be added to a perspective by clicking on **Other...**. It opens a "Show View" window. Click on one of the views, then click OK. The selected view will be activated and will appear in the View Menu for the selected perspective. The views can be activated or deactivated by a simple click in the View Menu.

Figure 2.49: Show View



2.8.2.3. Tools

In Tools, you can set your *Preferences*:

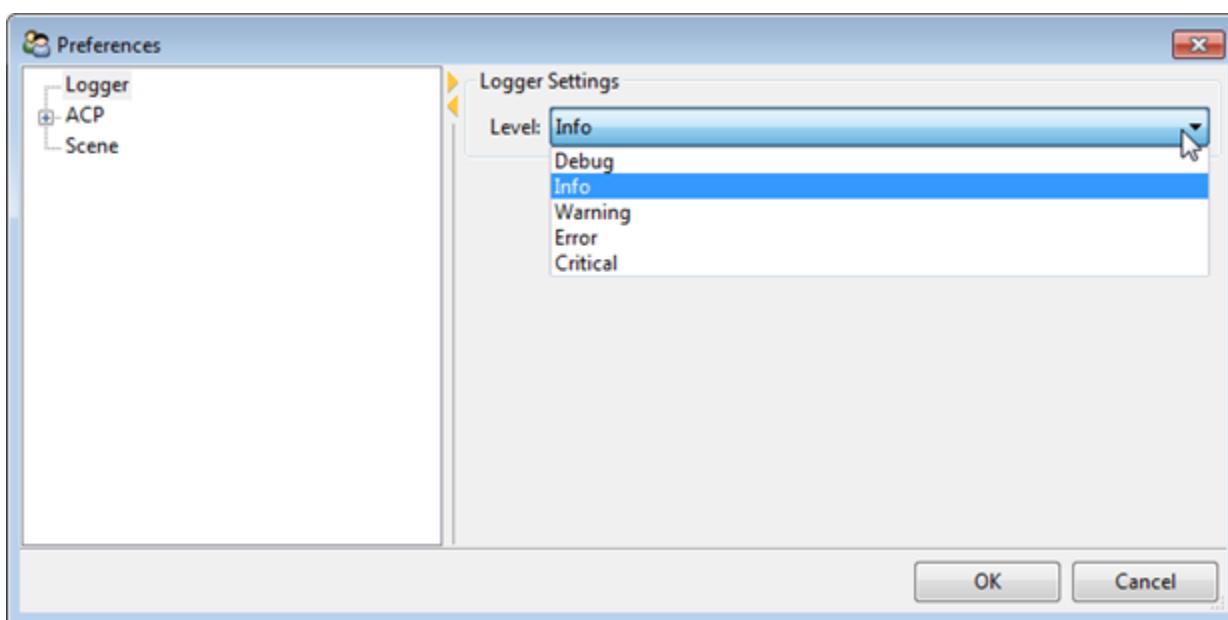
2.8.2.3.1. Logger Level

Choose which level of information should be shown in the Logger window.

The different levels are:

- Debug: Log everything, including debugging information
- Info (default): Log everything, excepted debugging information
- Warning: Log errors and warnings
- Error: Log error only
- Critical: not used

Figure 2.50: Logger Preferences



2.8.2.3.2. ACP Preferences

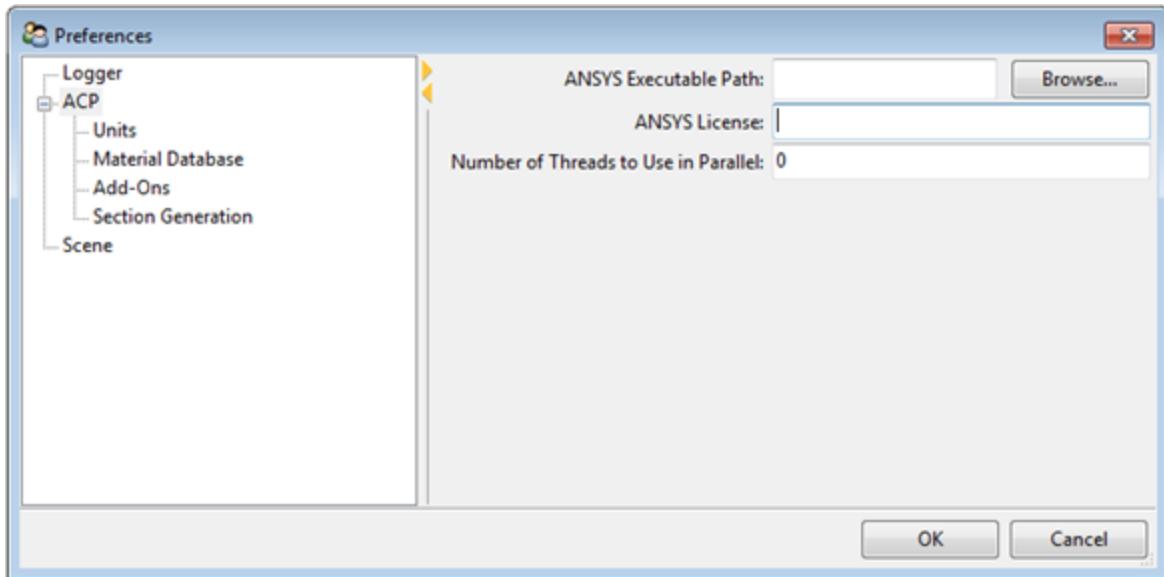
The Solver Path and License are not relevant in Workbench Integration, as the model is solved with Workbench Preferences. It is necessary to verify these options if the model is directly solved from ACP in Batch Mode through *Solve current Model* in *Model*.

Under ACP, the solver properties are defined:

- ANSYS Solver path: path to the ansys.exe file. If empty, ACP uses the default location defined during the installation of ANSYS.
- ANSYS License: defines which license must be used to solve the model. The different licenses are described in the ANSYS Help in the Product Variable Table. The information to give in ACP is the **Feature Name**, not the product. For example, enter ane3fl for an ANSYS Multiphysics License. By default, the defined license

is the ANSYS Structural. The feature names corresponding to the licenses can be found in the license file or in the ANSYS help (\ Installation and Licensing Documentation \ ANSYS, Inc. Licensing Guide \ 6. Product Variable Table). By default the **ANSYS License** field is empty. This means that the specifications defined in the *User License Preferences* of ANSYS are used.

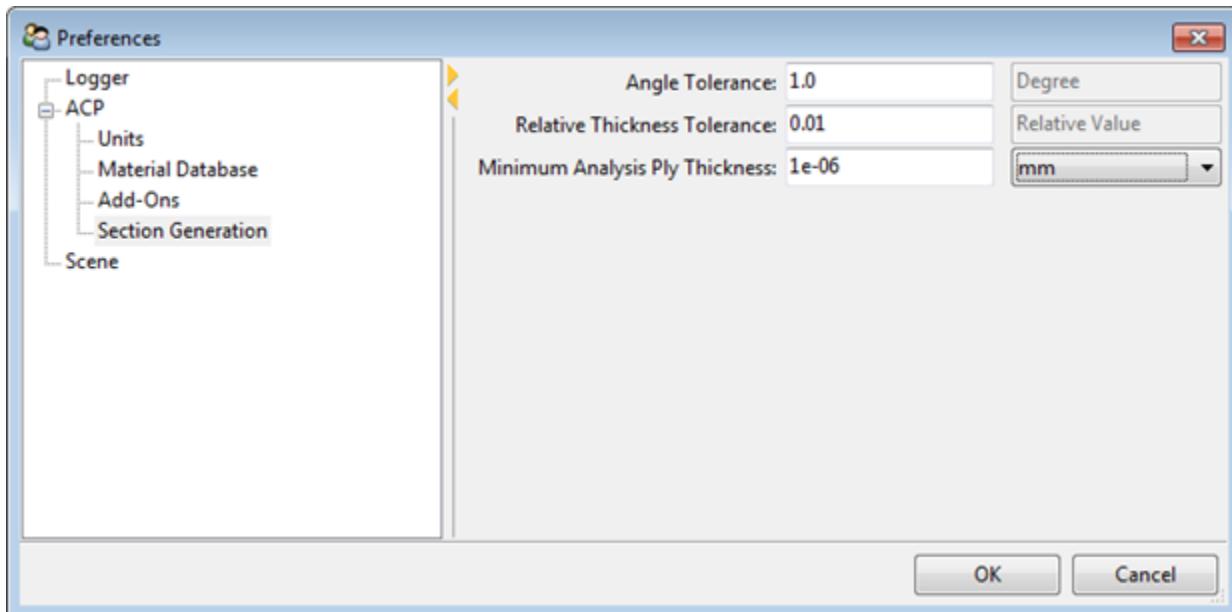
Figure 2.51: ANSYS Solver Preferences



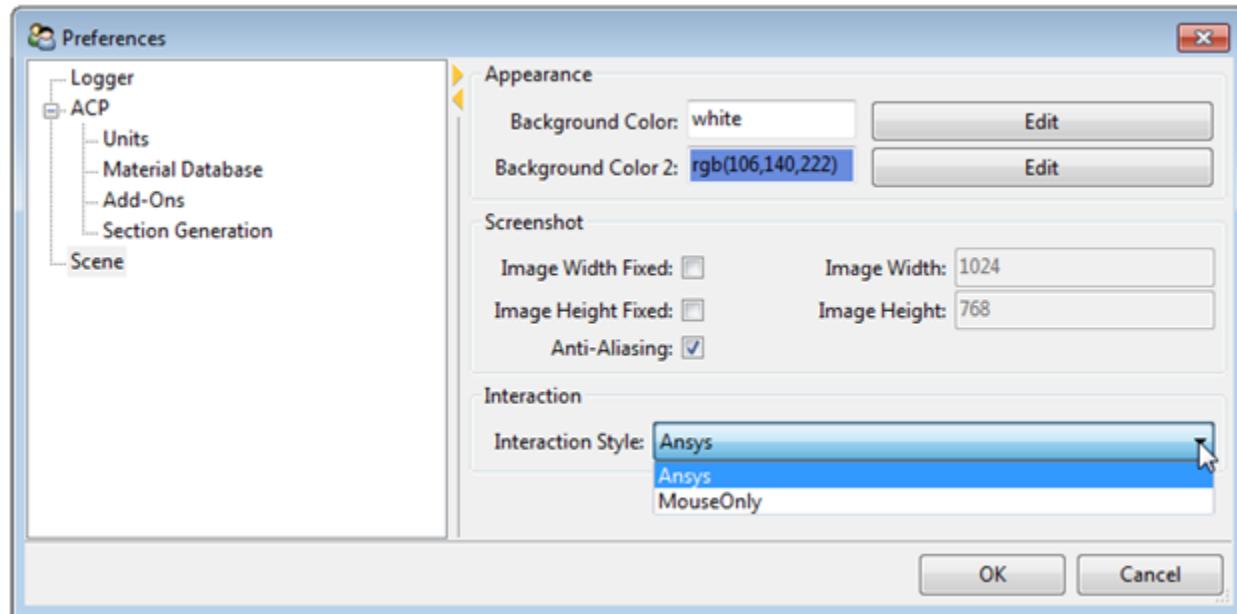
2.8.2.3.3. ACP Submenus

ACP Menu has four sub-levels:

- Units: Define which currency is used for the material cost.
- Material Database: Define the path to the Material Database (.acpMdb file).
- Add-Ons: Activate or deactivate the available Add-Ons.
- Section Generation: Define default tolerance values for the generation of sections as well as the Minimum Analysis Ply Thickness. For more information see [Section Computation \(p. 106\)](#).

Figure 2.52: ACP submenu and Section Generation Preferences**2.8.2.3.3.1. Scene**

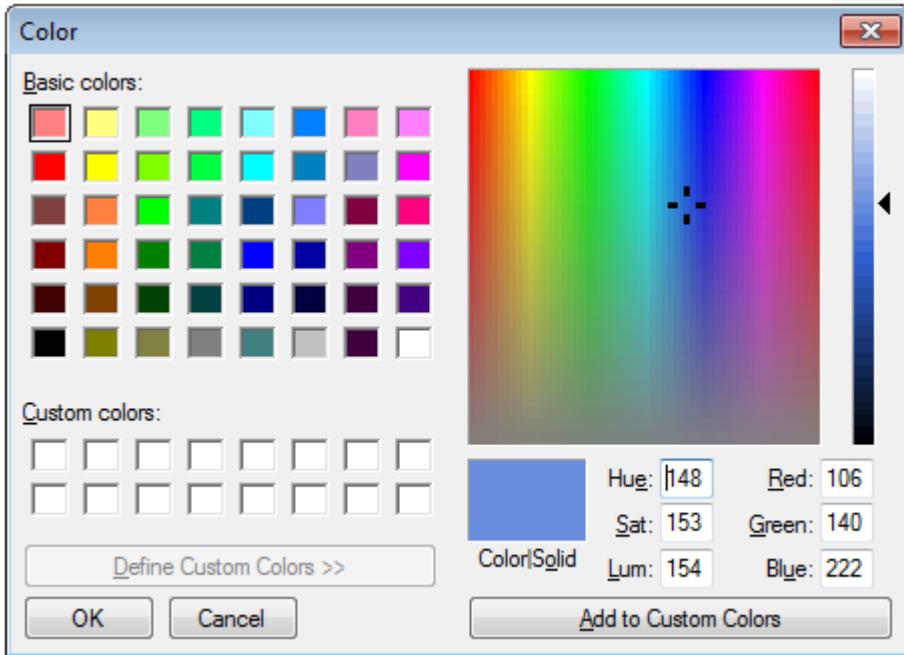
In the Scene Preferences, some graphical properties of the Scene can be modified. These properties are grouped in three parts:

Figure 2.53: Scene Preferences**2.8.2.3.3.2. Appearance**

The Scene Background can be defined as uniform or with a gradient from bottom to top. To obtain a uniform Background, Color and Color 2 must have the same color definition. Color is the bottom color and Color 2 is the top color for a gradient background. To modify these colors click on Edit. Some basic colors are predefined, but additional colors can be customized. To define a custom color, click on the desired position in the color palette. The luminosity can be modified using the slider on the right. An-

other way to define a new custom color is to directly enter its properties in the HSL (Hue, Saturation, Luminosity) system or in RGB (Red, Green, Blue) proportions. Modifying a value in one system interactively modifies the values in the other system.

Figure 2.54: Edit Color



2.8.2.3.3.3. Screenshot

By default, the size of the picture captured by the snapshot has the same size as the Scene size. The size of the captured picture can be fixed in width and height. This option must be used carefully. By default, the anti-aliasing option is also active. On some hardware, it may slow down image creation. In this case, it is recommended to deactivate this option.

2.8.2.3.3.4. Interaction

Two Mouse Interaction Styles are available in ACP. The standard one (ANSYS) is the same as the standard one in ANSYS Workbench. The two Interaction Styles are described in the table below:

Action	Interaction Styles	
	Ansys	Mouse Only
Pan	Ctrl+MB drag	MB drag
Dolly-Zoom	Wheel/Shift+MB drag	MB drag+RB click/MB drag+LB click
Box-Dolly-Zoom	RB drag	
Rotate	MB drag	MB+RB
Spin		MB+RB/MB+LB drag close to border
Pick	LB click	LB click/RB click
Box-Pick	LB drag	LB drag/RB drag

Rotation Point	MB click	MB click
Reset		

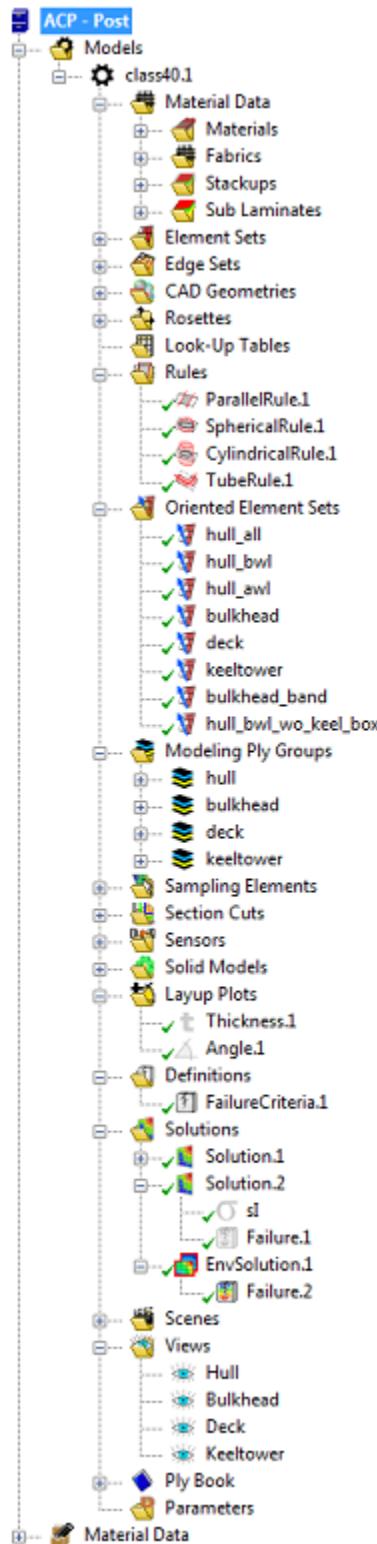
- (L,M,R)B denotes left, middle, right mouse button respectively
- + denotes concurrent execution

2.8.2.4. Help

In the Help Menu, you can access the Documentation, and the information on this version of ACP in About.

2.8.3. Tree View

There are some variations in the tree contents (available feature and feature contents) between ACP - Pre and ACP - Post.

Figure 2.55: Tree View

Descriptions of every part in the Tree View are provided in the Chapter [Usage Reference](#).

A status symbol appears for each item.

Figure 2.56: Locked Rosettes and the update status

In the example above (Rosettes):

- The symbol means that this Rosette is locked (locked as part of the imported model).
- The symbol means that this Rosette is updated.
- The symbol means that this Rosette is hidden in Scene.
- The symbol means that this Rosette is not updated. It can be updated with the general update in Toolbar or alone with the Menu, which appears with the Right Mouse Button.

The symbol indicates that this object is defined, but inactive and therefore not considered in any evaluation. *Modeling Plies, Solid Models and Analysis Plies* can be inactive.

You can move through the tree using the arrow keys. Up / Down moves to next items in the up / down direction respectively. The left / right arrows go to the upper / lower level in the tree structure. The sub-trees are automatically reduced and expanded respectively.

Special shortcuts (see [Ply Groups](#)) exist for the **Modeling Ply Group**.

2.8.4. Scene

The scene allows visualizing 3D representations of the model and all defined entities interactively. There is no limit to the number of Scenes that can be created, and changing from one Scene to another can be done with a single click. The user is able to navigate through the Scene by mouse or keyboard inputs triggering manipulations of the view properties (camera). View manipulations always refer to the rotation point which the user can specify by picking a point on a surface.

The Mouse Interaction in the Scene is described in [Interaction](#).

2.8.5. Toolbar

The Toolbar interacts with the Scene by modifying camera views or displaying or hiding some elements. The different buttons of the Toolbar are described by group below.



Scene Manipulation



2.8.5.1. Scene Manipulation

The first 6 buttons are standard views along each axis (both directions). Click on to fit the zoom to the model dimension. The scene can be viewed as full screen by clicking on .

Activate / Deactivate the perspective with . With save the current view of the scene or capture a snapshot for external applications.

2.8.5.2. Mesh Appearance

The Element Edges can be hidden or shown with . The shaded view is activated with . The Element selection is highlighted by default. It can be deactivated with . The highlighted elements can be switched between shell and solid elements with . The silhouette of selected elements can be highlighted with even when it is hidden by the mesh.

2.8.5.3. Orientation Visualization

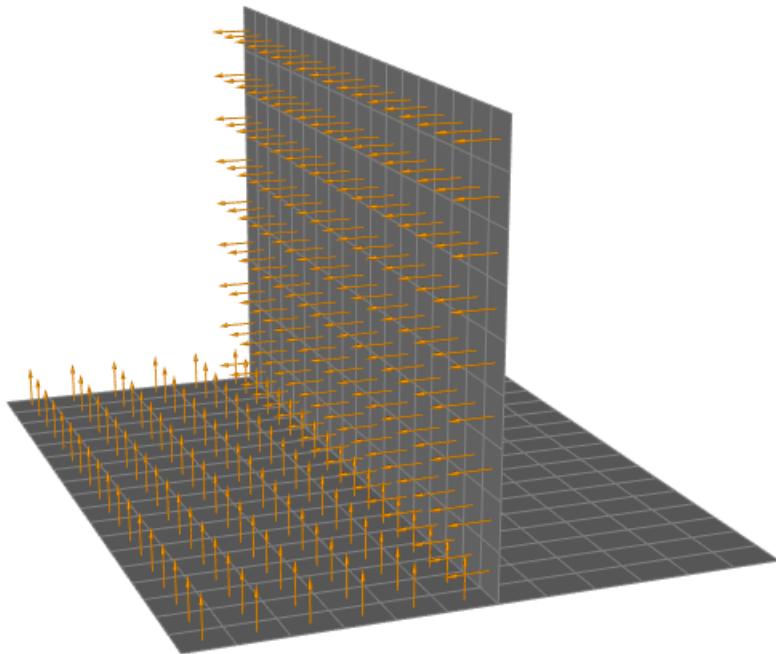
The orientation of surfaces and ply angles can be visualized with the several orientation arrows in the toolbar. The scaling factor for the ply offset visualization can also be controlled via the toolbar.

Figure 2.57: Orientation visualizations in the toolbar

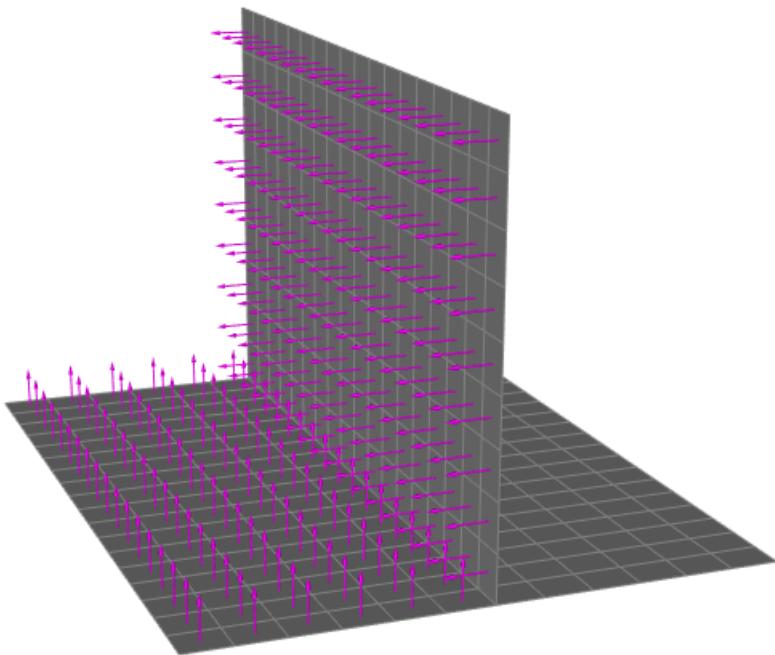
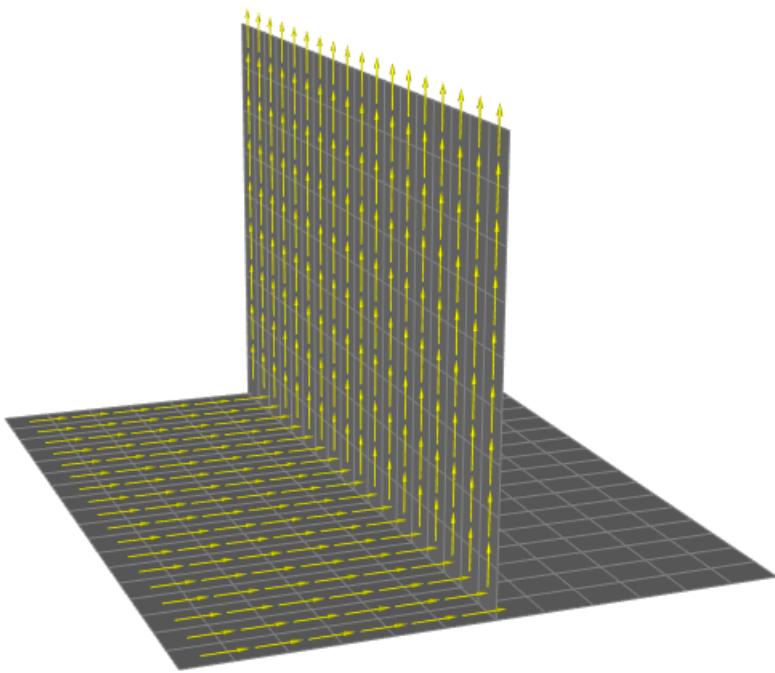


The element normal is displayed with the icon.

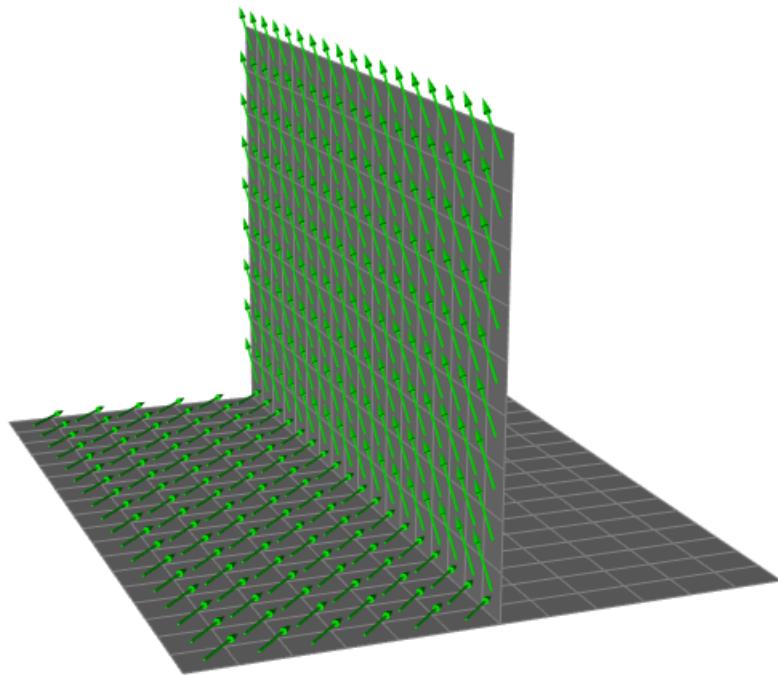
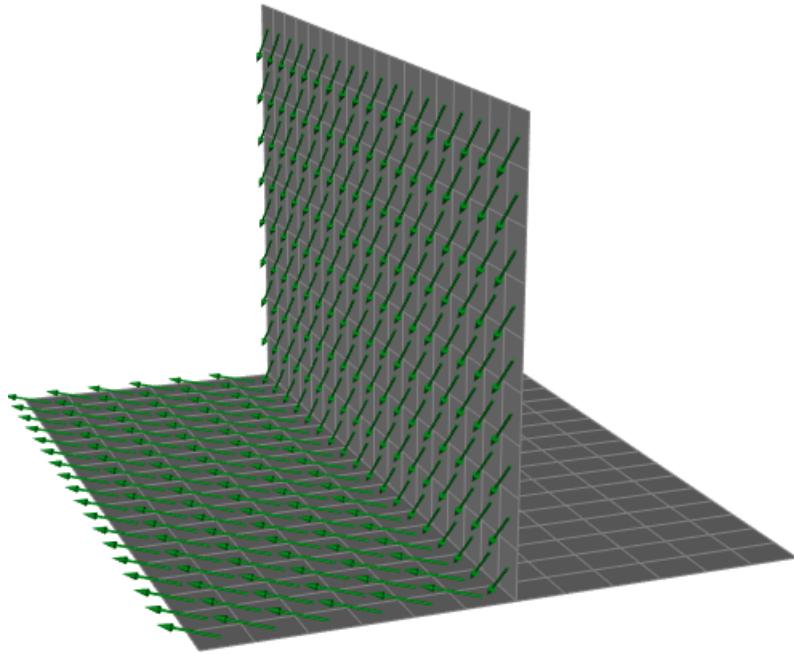
Figure 2.58: Visualization of the element normals



Select the icon to display the orientation of an Oriented element Set (OES). Select the icon to display its reference direction.

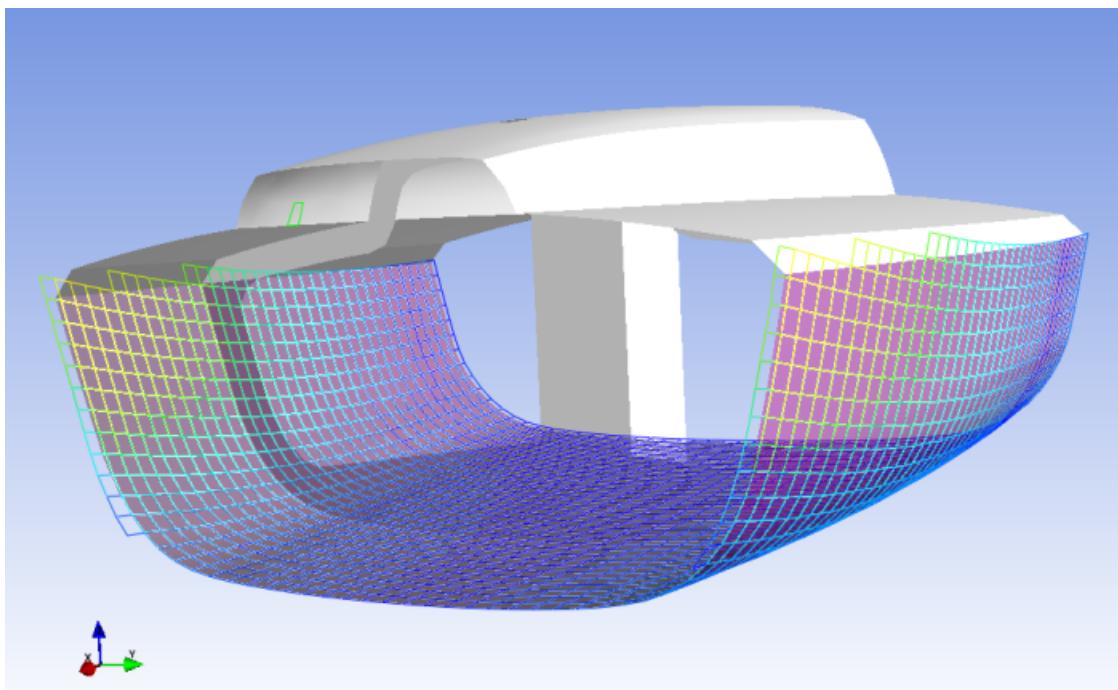
Figure 2.59: Visualization of the OES normal**Figure 2.60: Visualization of the OES reference direction**

Activate the  icon and select a ply to display its angle direction. Activate the  icon to display the transverse ply direction.

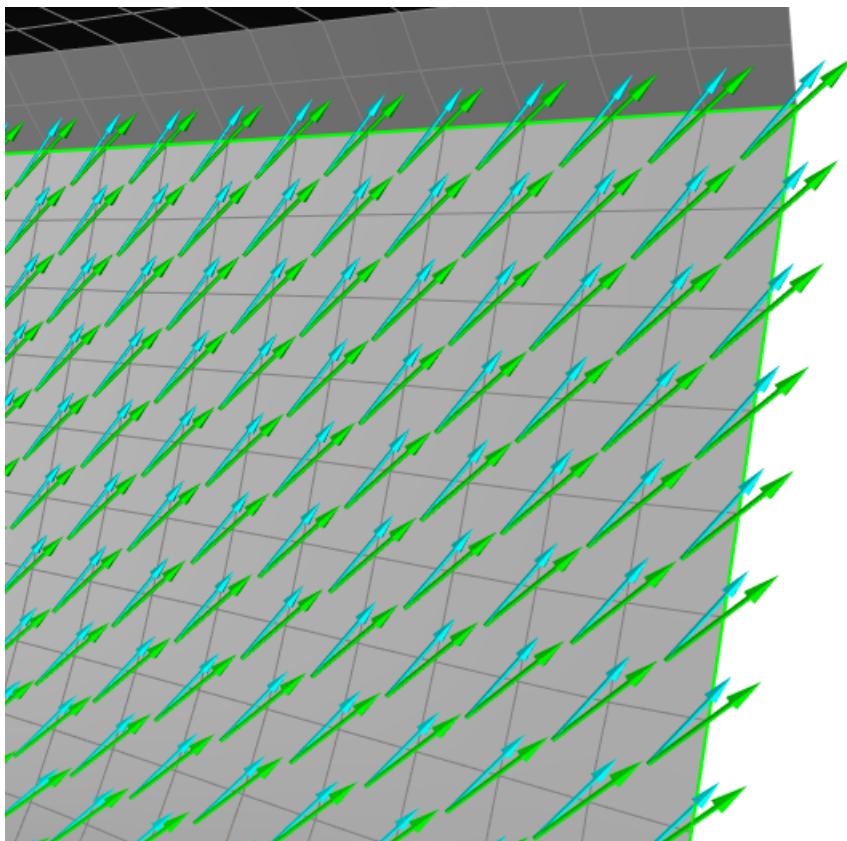
Figure 2.61: Visualization of the ply angle**Figure 2.62: Visualization of the transverse ply angle**

2.8.5.4. Draping and Flat Wrap

If draping is enabled, the draping "distortion" and the original *Flat Wrap* can be visualized. The draping mesh is plotted with . The worst distortion is located in red areas of the draping mesh.

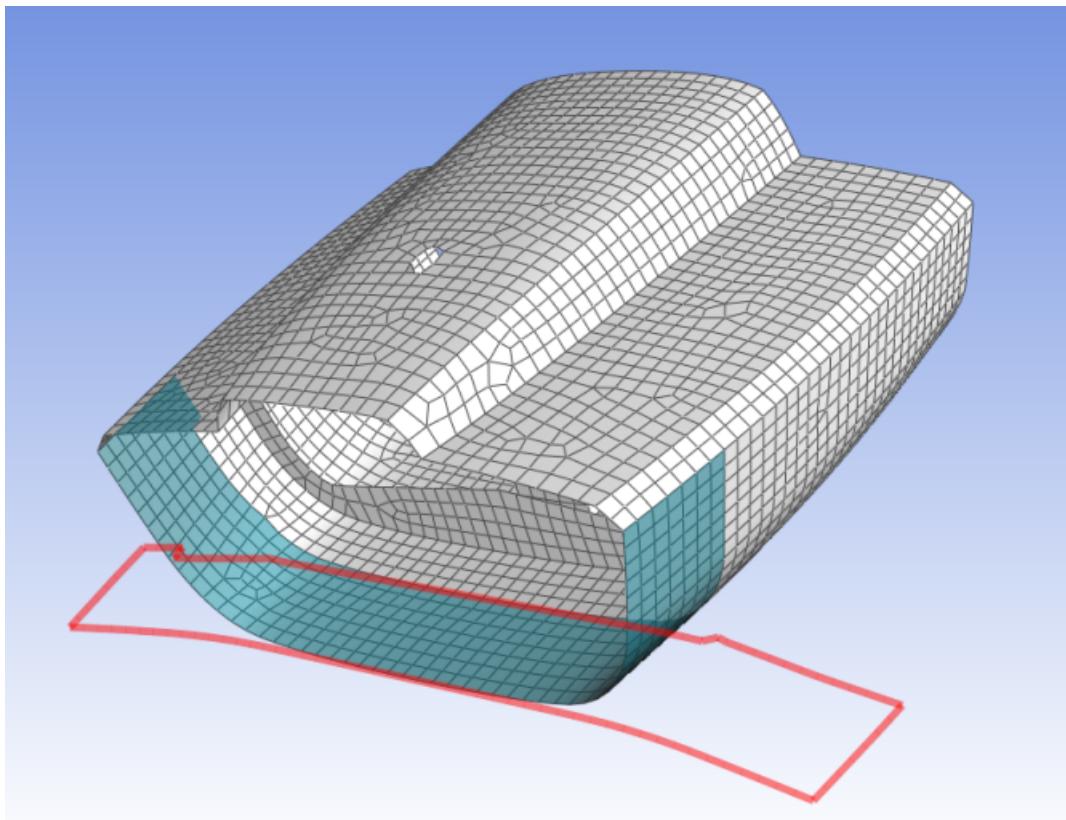
Figure 2.63: Draping "distortion" mesh

The draping effect can be highlighted by plotting the fiber direction and the draped fiber direction together. Equally the transverse fiber direction can be plotted with the transverse draped fiber direction .

Figure 2.64: Ply angle vector (defined and draped)

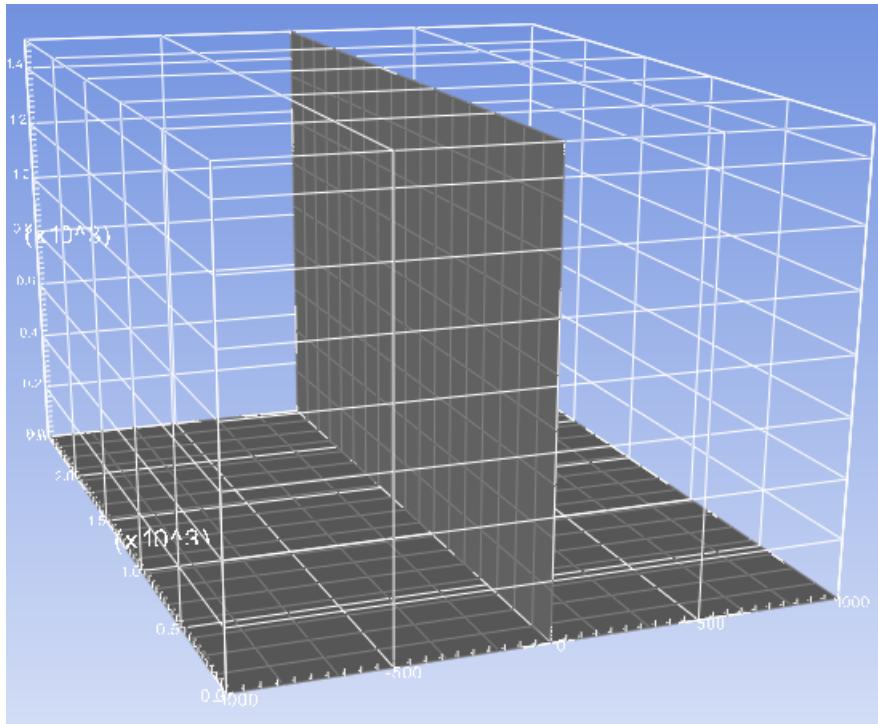
With the draping definition, the original flat surface can be developed and plotted with .

Figure 2.65: Flat wrap surface of the ply



2.8.5.5. Other Features

A coordinate system is present by default in the bottom left corner of the screen for a better 3D orientation. This coordinate system can be deactivated by clicking on . An enclosed box with coordinates can be plotted by clicking on .

Figure 2.66: Enclosed box and coordinate system

2.8.5.6. Post-processing

Display the legend and the Failure Criteria Textplot with and . Show the plot description with .

2.8.5.7. Updates

After some operations (Reload Model, modification of one or more plies, activate post-processing,...), an update of the features and of the Scene is necessary.

2.8.6. Shell View

All commands performed through GUI interaction are executed by the internal Python interpreter. The same commands can be entered manually in the Python Shell View window. The Shell View also provides standard text editing features like Copy/Paste/...

2.8.7. History View

All commands processed during the existing session are stored in the Python command history. They can be inspected in the History View window, where each text line refers to an executed command. The command history is also available in the Shell View by using the Ctrl+Up/Ctrl+Down keys.

2.8.8. Logger

The information saved in the file %app_data%\Ansys\vXXX\acp directory\ACP.log (e.g.: C:\Users\user-name\AppData\Roaming\Ansys\vXXX\acp\ACP.log) is shown in this View. The level of information is defined in the Menu Tools.

Chapter 3: Composite Model Techniques

The composite modeling techniques are described in the sections below:

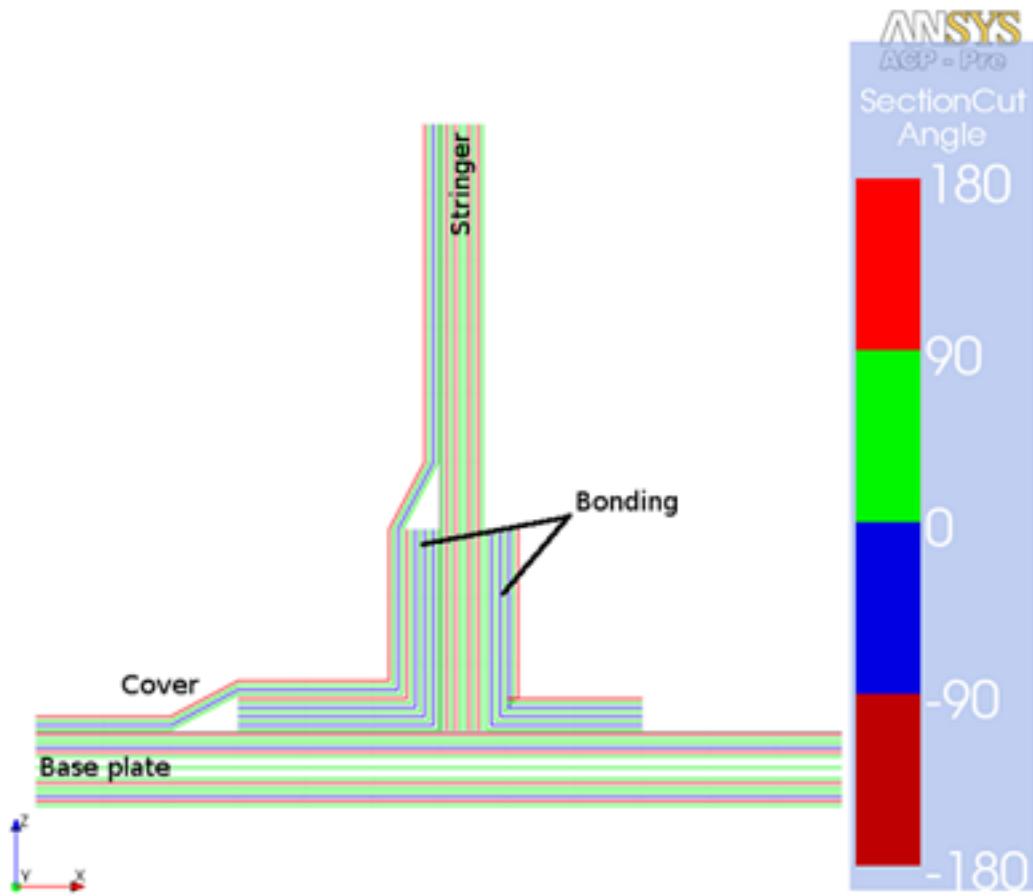
- 3.1.T-Joint
- 3.2.Local Reinforcements
- 3.3.Ply Tapering and Staggering
- 3.4.Variable Core Thickness
- 3.5.Draping
- 3.6.Ply Book
- 3.7.Guide to Solid Modeling
- 3.8.Guide to Composite Visualizations
- 3.9.Element Choice in ACP

3.1.T-Joint

T-joints are widely-used to bond a primary structure to a secondary one. A good example is a frame with a stringer of a boat hull. The *Oriented Element Set* (OES) concept allows to define such complex laminates by an intuitive approach. An example with the complete ACP model is delivered with ACP and can be found in the installation directory of ACP.

The laminate of a T-joint can be split into several sublaminates:

- Base plate (or skin)
- Stringer (or frame)
- Bonding
- Cover

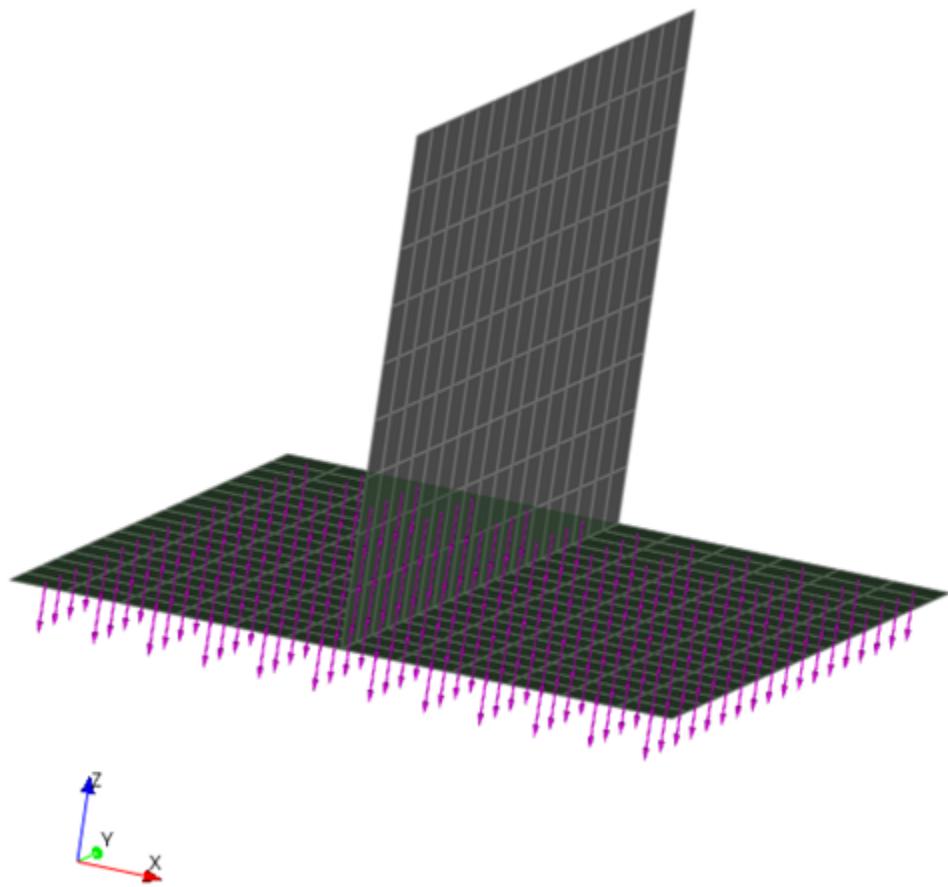
Figure 3.1: T-joint lay-up

The basic idea is to define different *Oriented Element Sets* for the different regions. The modeling plies are then associated with the OES and their order defines the stacking sequence of the laminate.

The first OES is defined for the base. The offset direction of this OES shows from top to bottom as shown in the Figure [OES for the base plate](#).

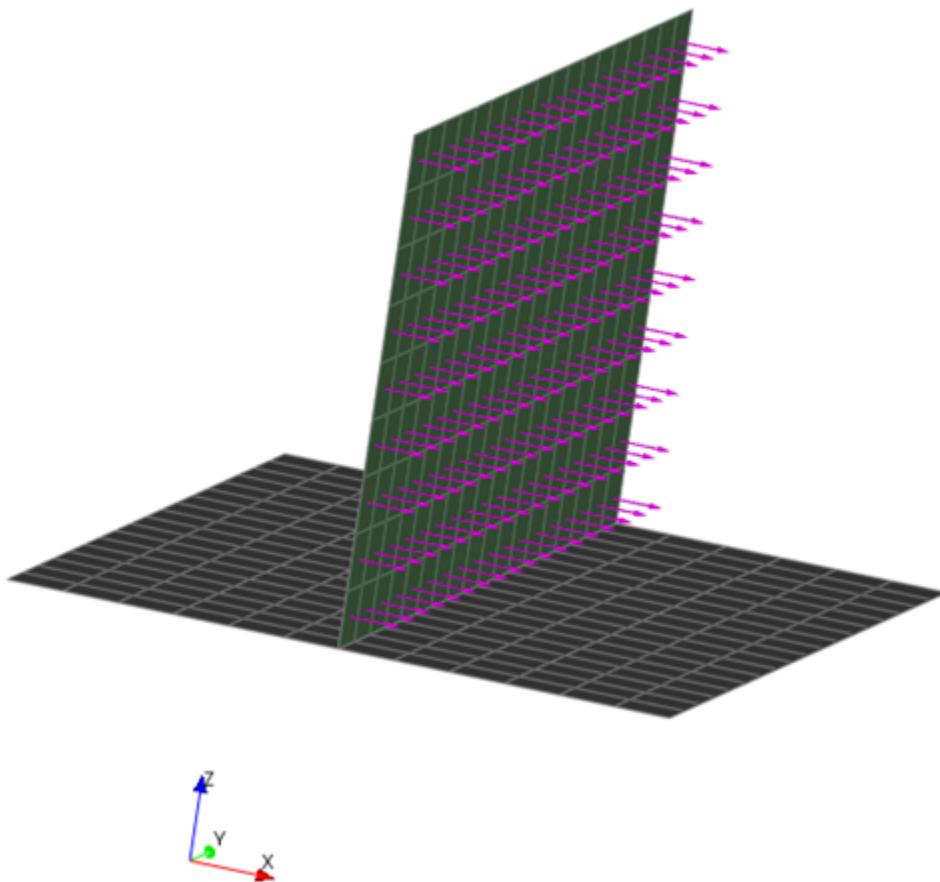
Figure 3.2: OES for the base plate

ANSYS
ACP - Pre



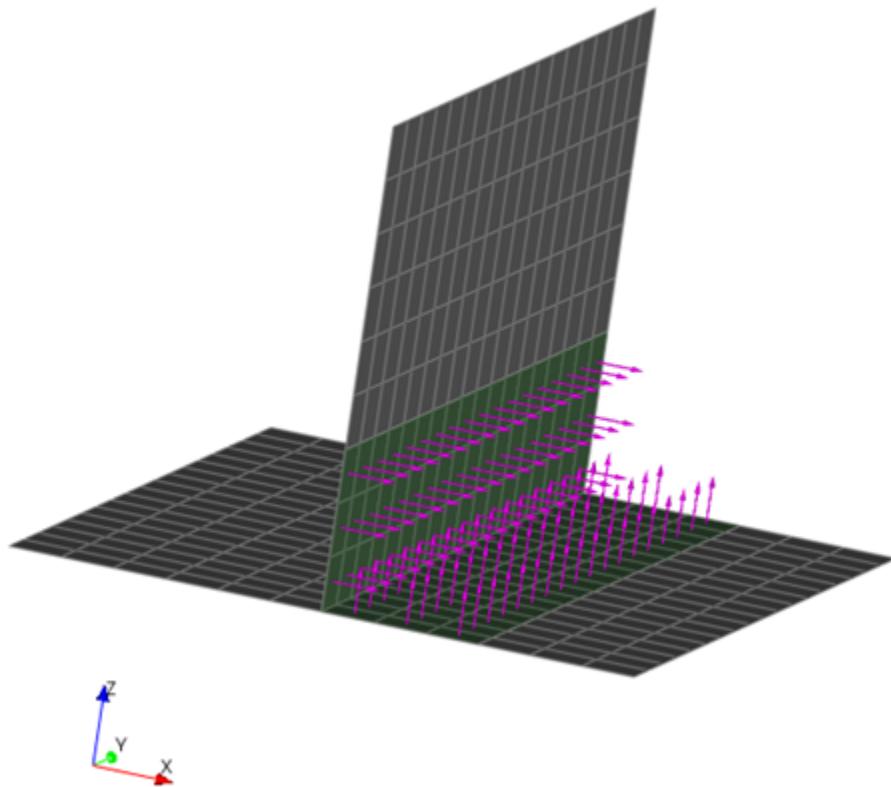
Element Labels: 326

The OES of the string has an orientation parallel to the global x direction as shown in the Figure [OES for the stringer](#)

Figure 3.3: OES for the stringer**ANSYS**
ACP - Pre

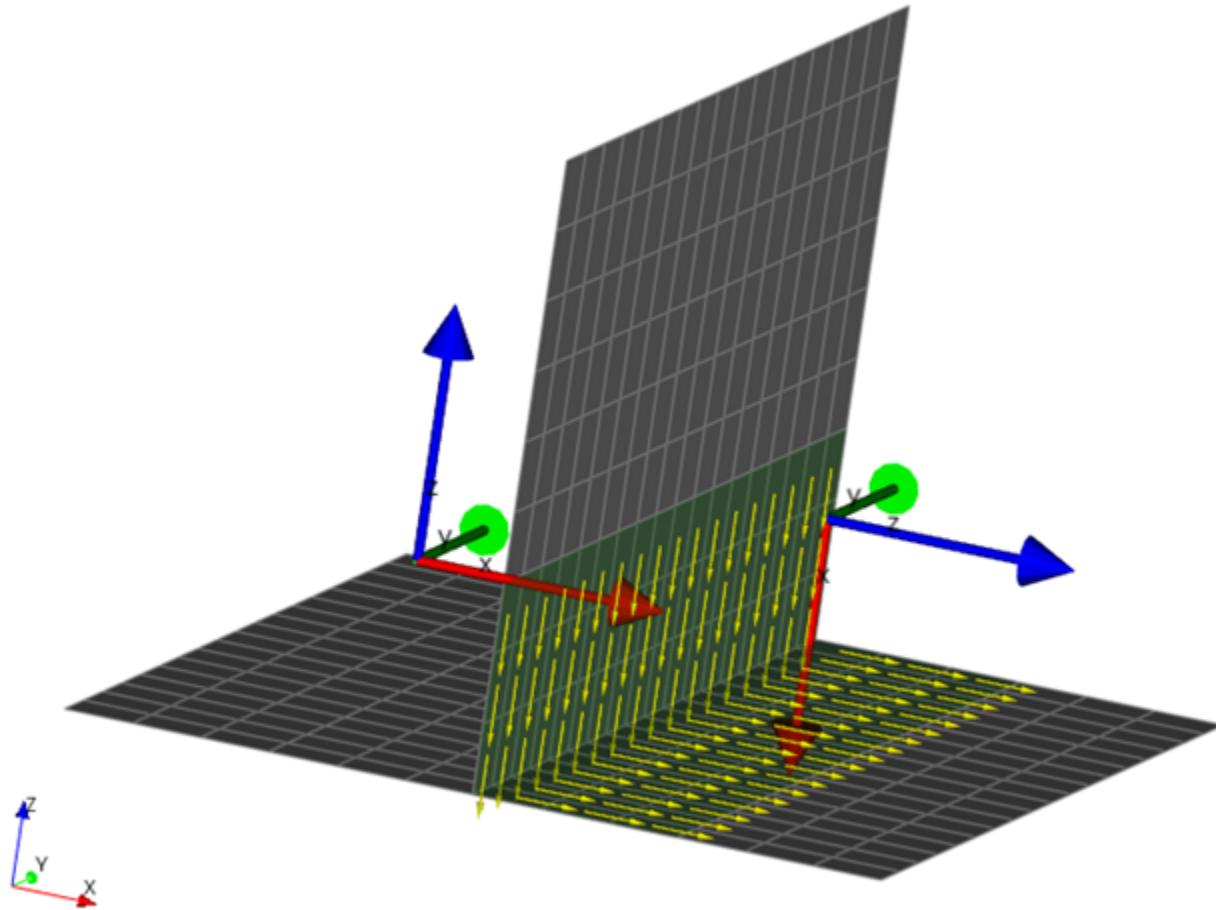
Element Labels: 326

The OES concept allows to define several offset directions for one element, or in other words: OES can overlap and can have different orientations. This functionality is used to define the offset direction for the bonding layers as shown in the Picture [OES for bonding plies](#). The offset direction of the base plate is different if compared with OES for the base plate.

Figure 3.4: OES for bonding plies**ANSYS**
ACP - Pre

Element Labels: 326

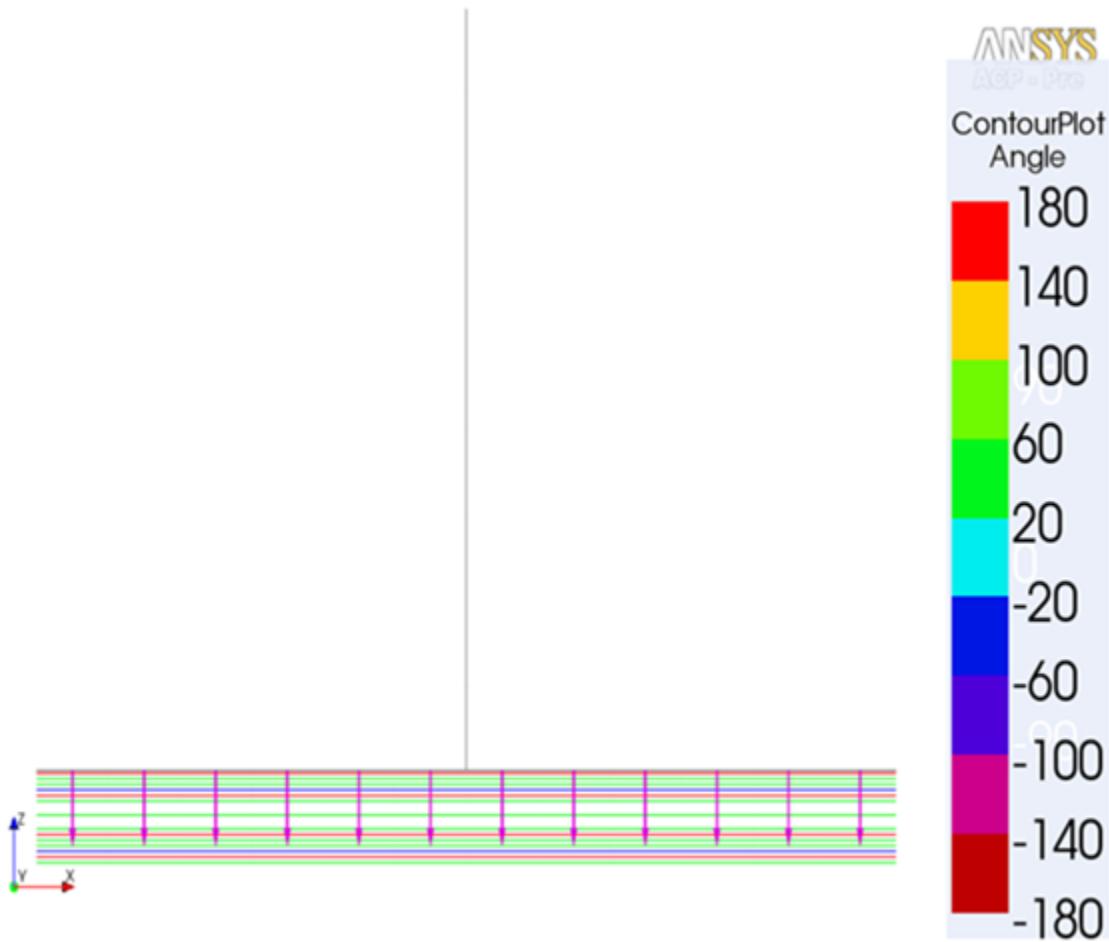
In addition the OES feature allows to define the reference directions for complex shapes (twisted surfaces, right angles). The reference direction is computed from one or several reference coordinate systems (CSYS) as shown in Figure [Reference direction](#). In this case two CSYS are selected.

Figure 3.5: Reference direction**ANSYS**
ACP - Pre

Point Labels: 398

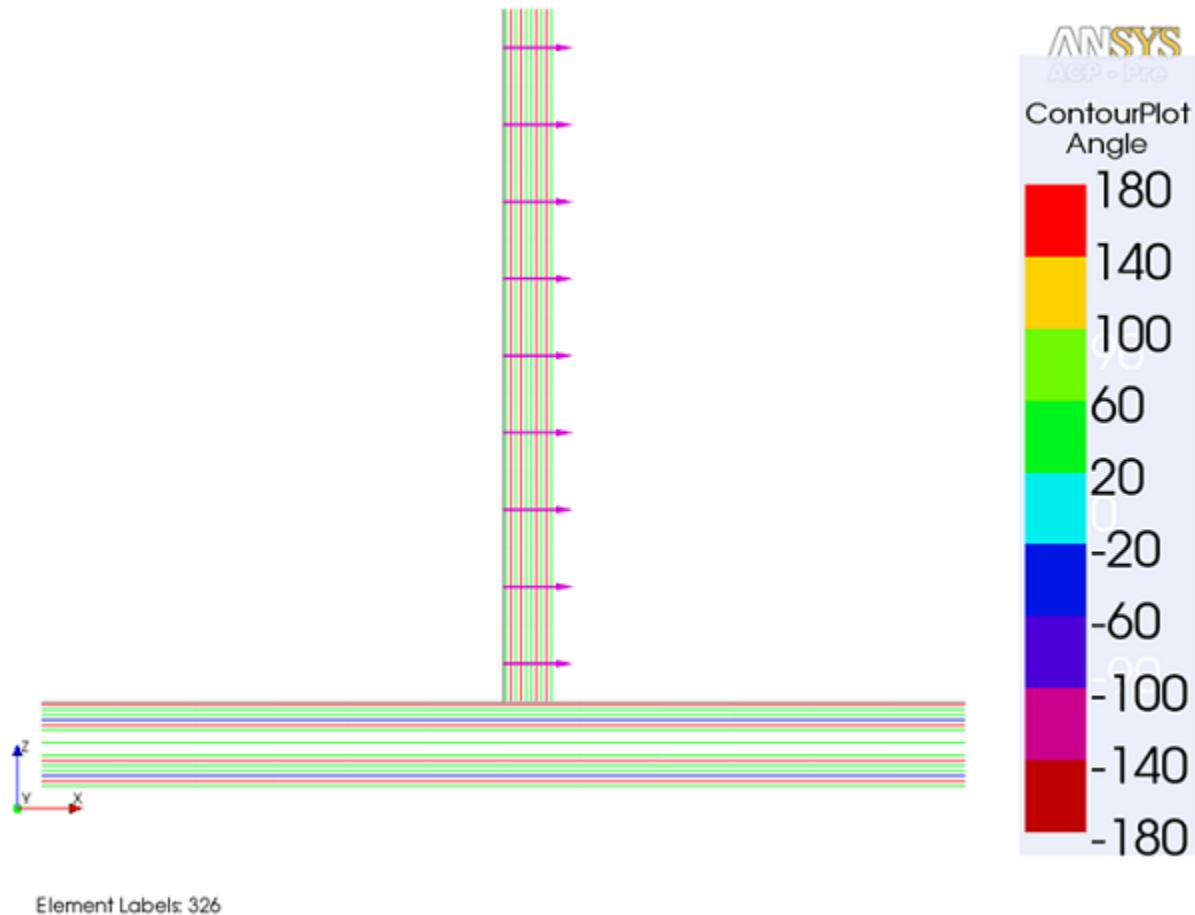
So far all necessary OES are defined. The next step is to define the Modeling Plies in the same order as the structure is produced later. First, the base layup is defined using the OES of the base plate.

Figure 3.6: Laminate of the base plate

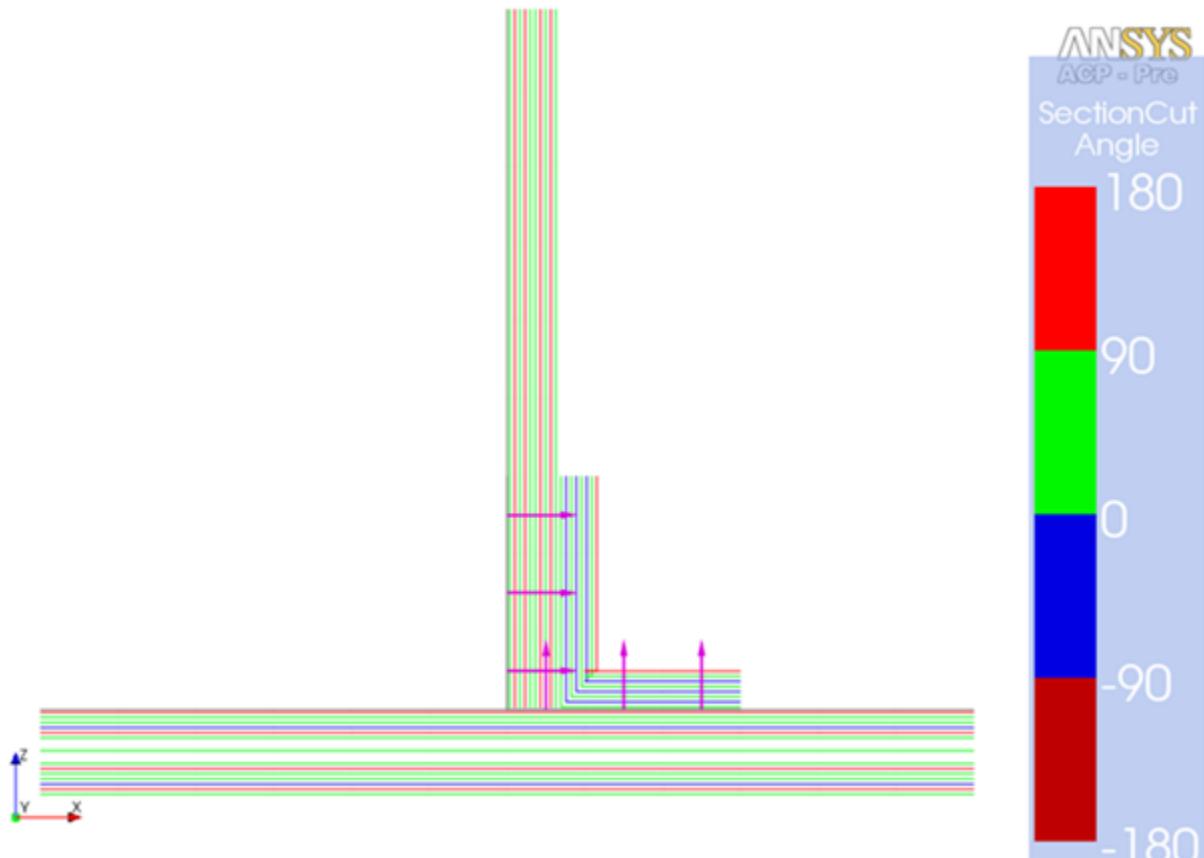


Element Labels: 326

The next plies are added to the stringer.

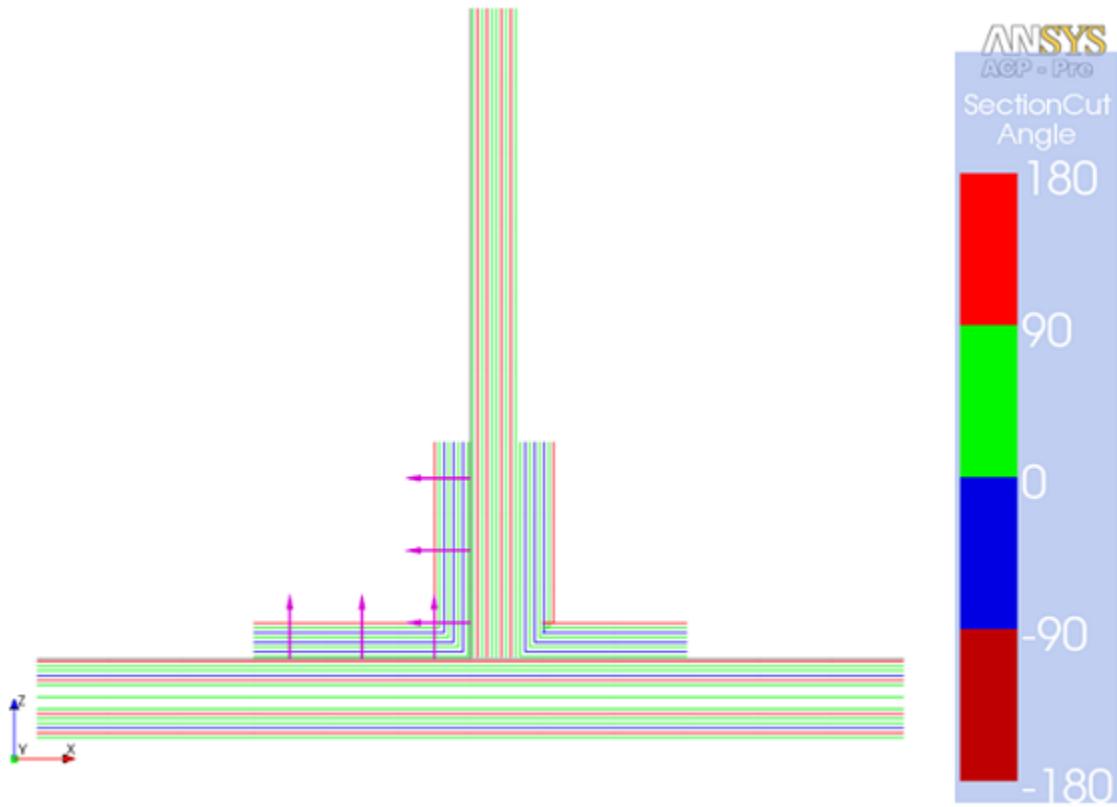
Figure 3.7: Laminate of the base plate and stringer

It is important to define the base plate and stringer laminate before the bonding plies are defined because the order is responsible for the final offset. As shown in Figure [First bonding laminate](#), the bonding layers are applied to the top of the base plate and onto the plies of the stringer.

Figure 3.8: First bonding laminate

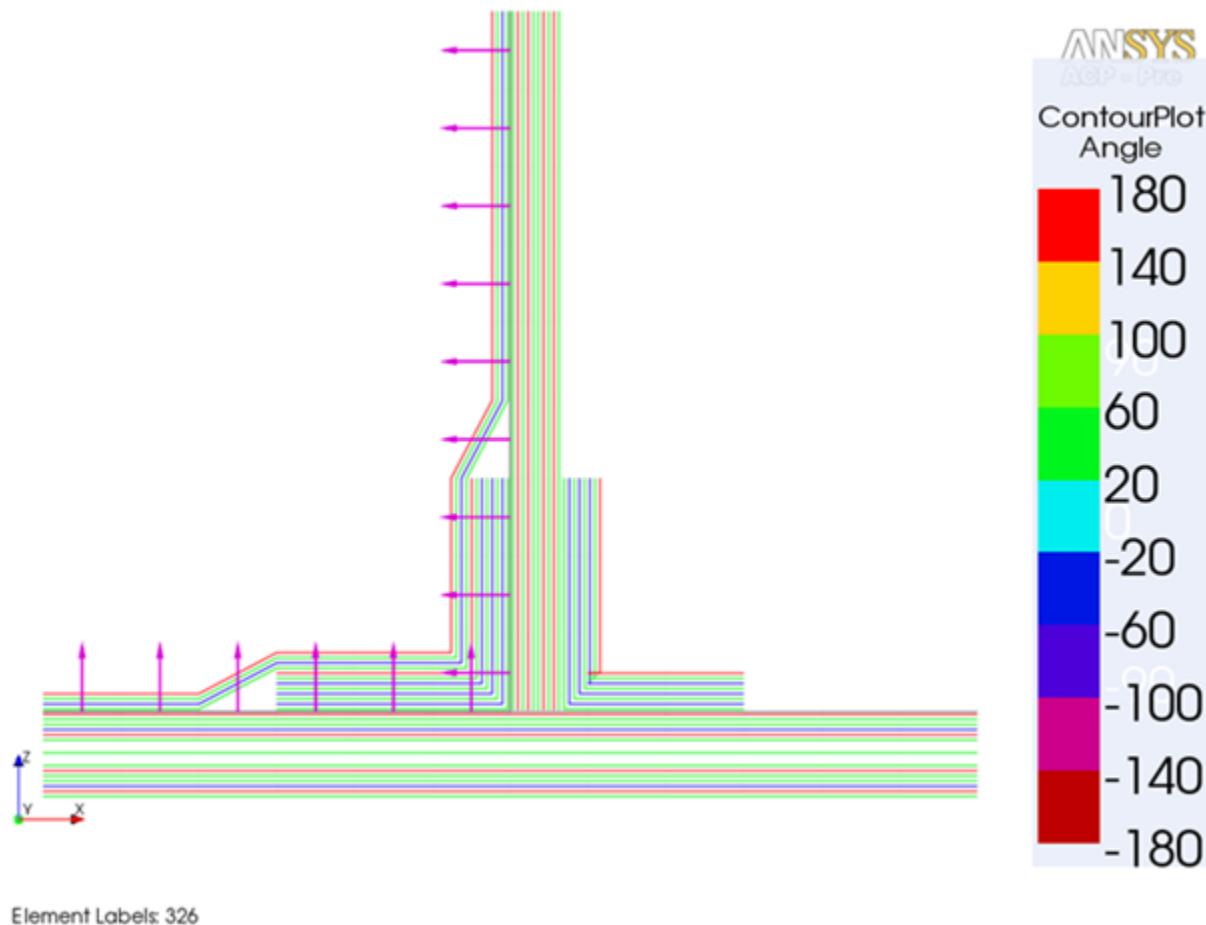
Element Labels: 326

On the other side, the second bonding laminate is offset to the top (base plate) and to the left (stringer).

Figure 3.9: Second bonding laminate

Element Labels: 326

Finally the cover plies finishes the layup definition of the T-joint. The Picture [Cover plies](#) shows that ACP can also handle drop-offs.

Figure 3.10: Cover plies

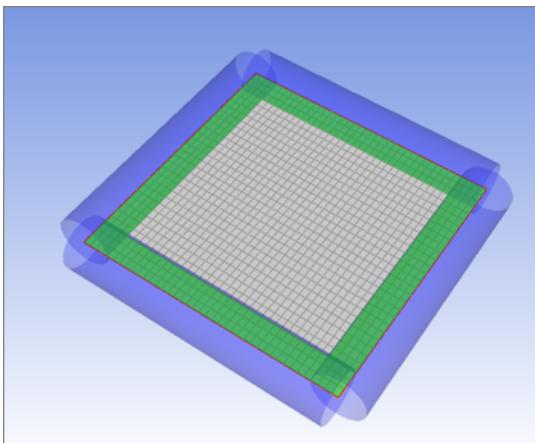
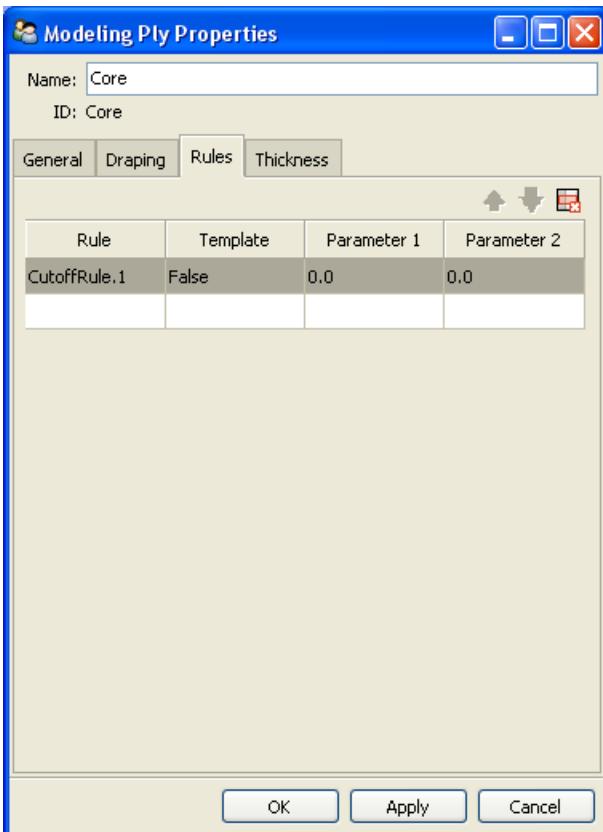
The final model is given in the example folder of the ACP installation ([Tutorials and Examples](#)).

3.2. Local Reinforcements

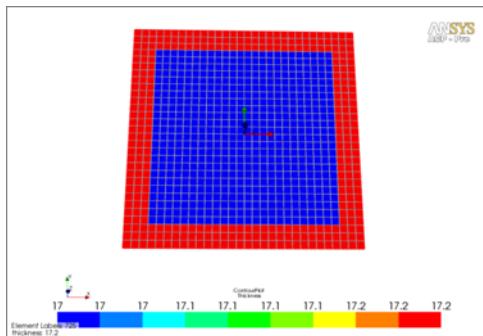
Regions with cut outs, holes or load introduction elements are normally highly stressed and require local reinforcements to prevent failure. ACP offers different ways to define local patches. Rules can be used to apply reinforcements to selected areas of the structure's geometry. The shape of a reinforcing ply is defined by the intersection of an Oriented Element Set and the selected Rules.

The examples class40 and Tutorial 2 use Parallel Rule and Tube Rule to define patches. Tutorial 2 describes how a Tube Rule can be defined to add a ply following an edge. The procedure involves these steps:

- Define an Edge Set from the boundaries of an Element Set
- Create a Tube Rule along the defined Edge Set with a certain inner and outer radius,
- Create a new ply and configure the Rules in the Rule tab of the Modeling Ply property dialog.

Figure 3.11: Tube rule**Figure 3.12: Rule tab of the modeling ply property dialog**

The rule parameters can be modified for each Modeling Ply. This allows the user to work with one *Rule* to define the staggering of a laminate. The user just activates "Template" and sets the new parameters. The final result can be double-checked with Section Cuts or a thickness contour plot as shown in the figure below.

Figure 3.13: Resulting local reinforcements

Rules can also be combined with Oriented Element Sets and other Rule types like Parallel, Spherical or Cylindrical are also implemented in ACP. Any combination of these rules allows to create plies with complex shapes.

3.3. Ply Tapering and Staggering

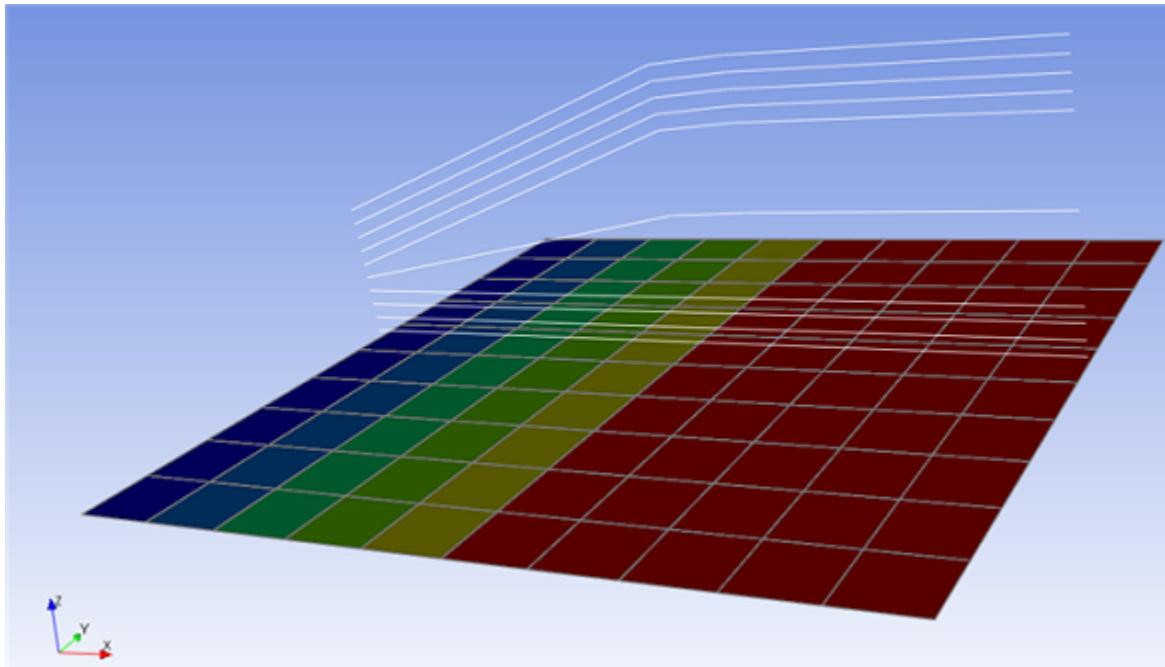
Ply tapering and staggering can be quickly defined within ACP. Several examples are shown below.

3.3.1. Ply Tapering

Core plies are generally much thicker than regular or woven plies. This means that core edges must be tapered for structural and manufacturing reasons. When a taper is applied to an edge of a ply in ACP and the corresponding thicknesses are evaluated and mapped automatically onto the finite elements.

Class 40

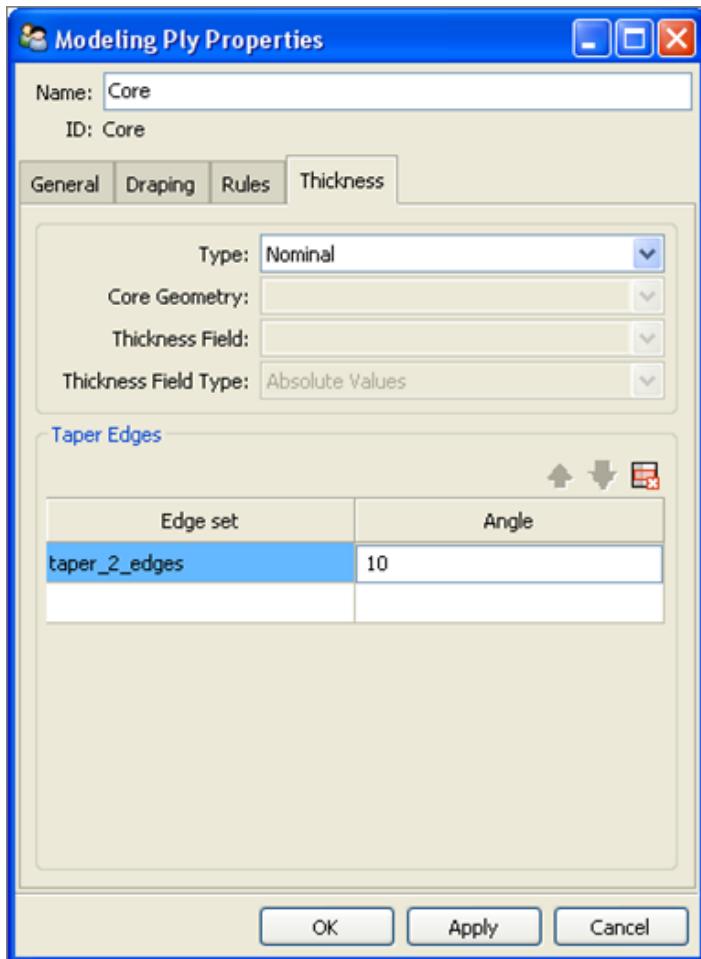
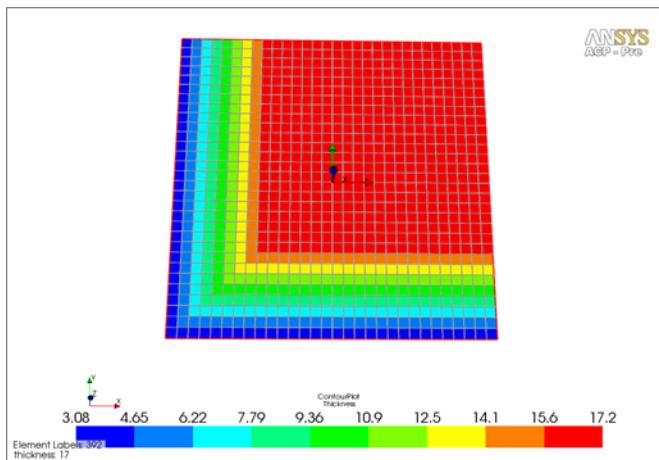
The *class40* example uses a tapered core. Open the model and check the *Modeling Ply* "core_bwl" in the *Modeling Ply Group* "Hull". In the *Thickness* tab of the property dialog a taper angle of 15 degrees is defined for the edge *edgeset.2*. A Section Cut or thickness contour plot illustrates the final result as shown in the next figure.

Figure 3.14: Tapered edge

Tutorial 2

In *Tutorial 2* a ply tapering is defined along 2 edges. The procedure involves these steps:

- Define an *Edge Set*. In this case the *Edge Set* is defined through a *Named Selection* in ANSYS Mechanical.
- Open the *Thickness* tab in the *Modeling Ply* properties dialog.
- Select the edge and define the taper angle.

Figure 3.15: Tapering in Ply Definition**Figure 3.16: Thickness distribution after core tapering**

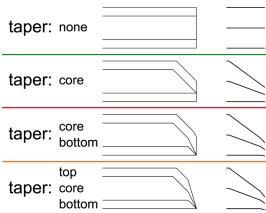
Tapering of Multiple Plies

The Taper Edges option for Modeling Plies is suitable for defining individual taper angles for specific modeling ply. Its intended purpose is the use with a single tapered ply, such as a core, however it can be used for tapering multiple plies. If the Modeling Ply tapering option is used for multiple layers the taper angle is applied to each Modeling Ply and the vertical ply thickness distribution is superposed. In such a case, the total taper angle of the layup is generally higher than the individual taper angles.

The total taper angle scales non-linearly with the number of plies and their thicknesses. Furthermore, the taper size is a dependent on the size of the mesh. For this reason, care should be taken when using the Taper Edges option for multiple plies. When modeling a composite with a defined total taper angle a rule-based definition may be more suitable. The trailing edge of an airfoil blade is an example for such an application.

The example below shows the effects of superposing multiple modeling plies that have the same taper angle. The middle column shows a layup schematic while the right column displays the corresponding representation of a section cut in ACP. The superposition of two different ply thicknesses results in two taper angles of which one is steeper than the nominal angle.

Figure 3.17: Superposition of Modeling Plies with identical taper angles. Schematic (middle) and Section View illustration (right).

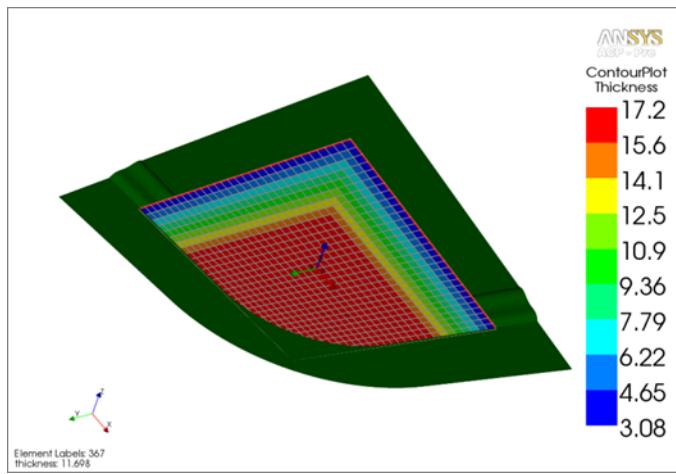


3.3.2. Ply Staggering

Cutoff Rule

A *Cutoff Rule* is used to cut plies and is suitable to define a ply staggering. This feature is not limited to an edge because the staggering is derived from a *CAD Geometry*. The intersection between the ply and geometry defines where the plies are cut. The ply offsets are taken into consideration. This allows the user to define a laminate where the total thickness follows a 3D shape.

Figure 3.18: Thickness distribution of a laminate with a cutoff rule

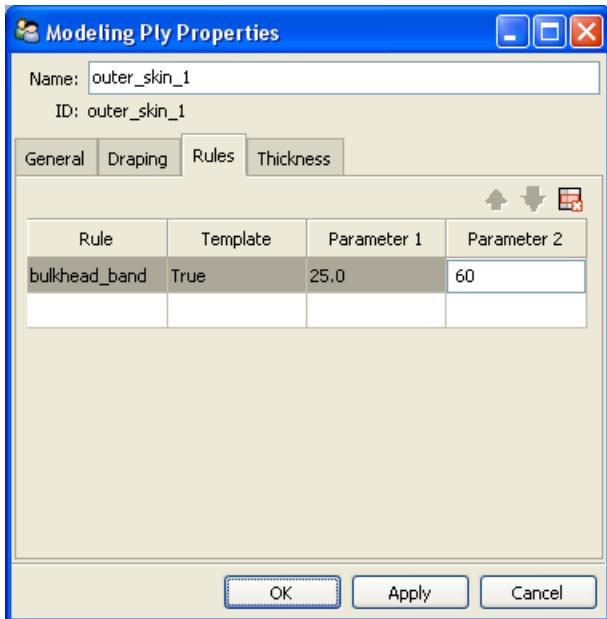


Template Rule

The template rule feature of the *Modeling Ply* allows to use one *Rule* to define plies of different extensions. The user can redefine the parameters of a rule in the property dialog of the *Modeling Ply*. A wind turbine blade, for example, has hundreds of similar plies that only differ in their axial extension. Such plies can be defined with one *Oriented Element Set* and one *Parallel Rule* and the use of template parameters.

The template parameters are easily adjusted using the "Import from / Export to CSV file" feature in the *Modeling Ply Group*.

Figure 3.19: Template rule definition



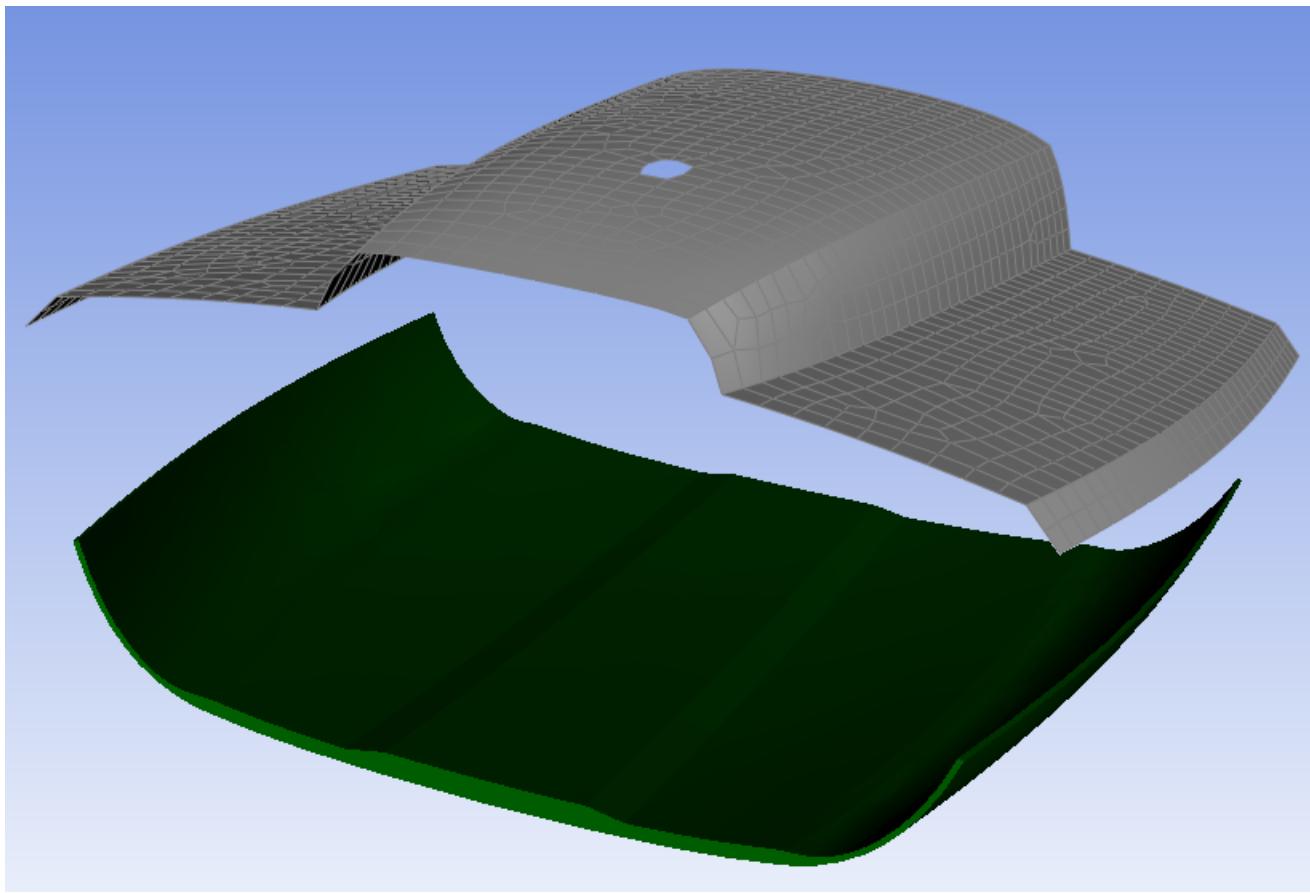
3.4. Variable Core Thickness

In many cases, the thickness of a sandwich panel is constant or at least single plies have a constant thickness. Regarding structural efficiency, cores with variable thickness are used more and more since CNC milling allows the production of core plies with complex shapes.

In ACP there are three different ways to define a laminate with variable thickness:

3.4.1. Solid CAD Geometry

An external core geometry can be used to define the variable core thickness. The 3D shape of the core is modeled in a CAD tool as a 3D solid or a closed shell. This *CAD Geometry* can be imported directly into ACP or via the Workbench Project Schematic.

Figure 3.20: Imported Core Geometry

In the *Thickness* tab of the *Modeling Ply* the thickness definition can be changed from *Nominal* to *from Geometry*. In this case, ACP samples through the geometry in the normal direction and evaluates the thickness of the core for each element. The original thickness defined in the *Fabric* definition becomes obsolete. This method is used in the class 40 example delivered with the ACP Installation.

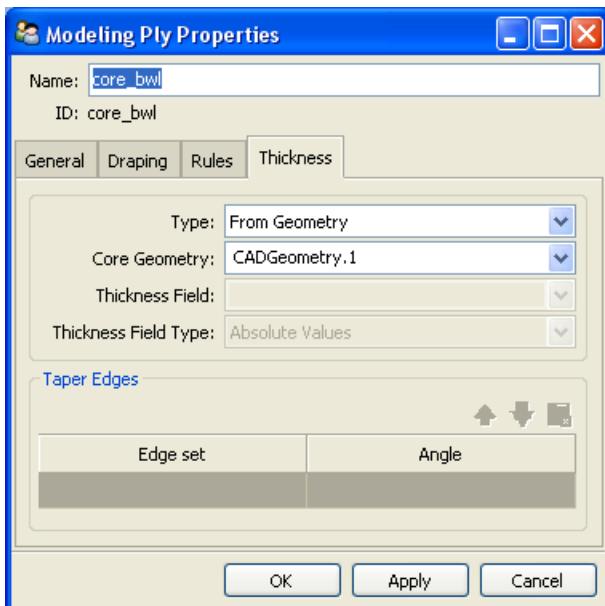
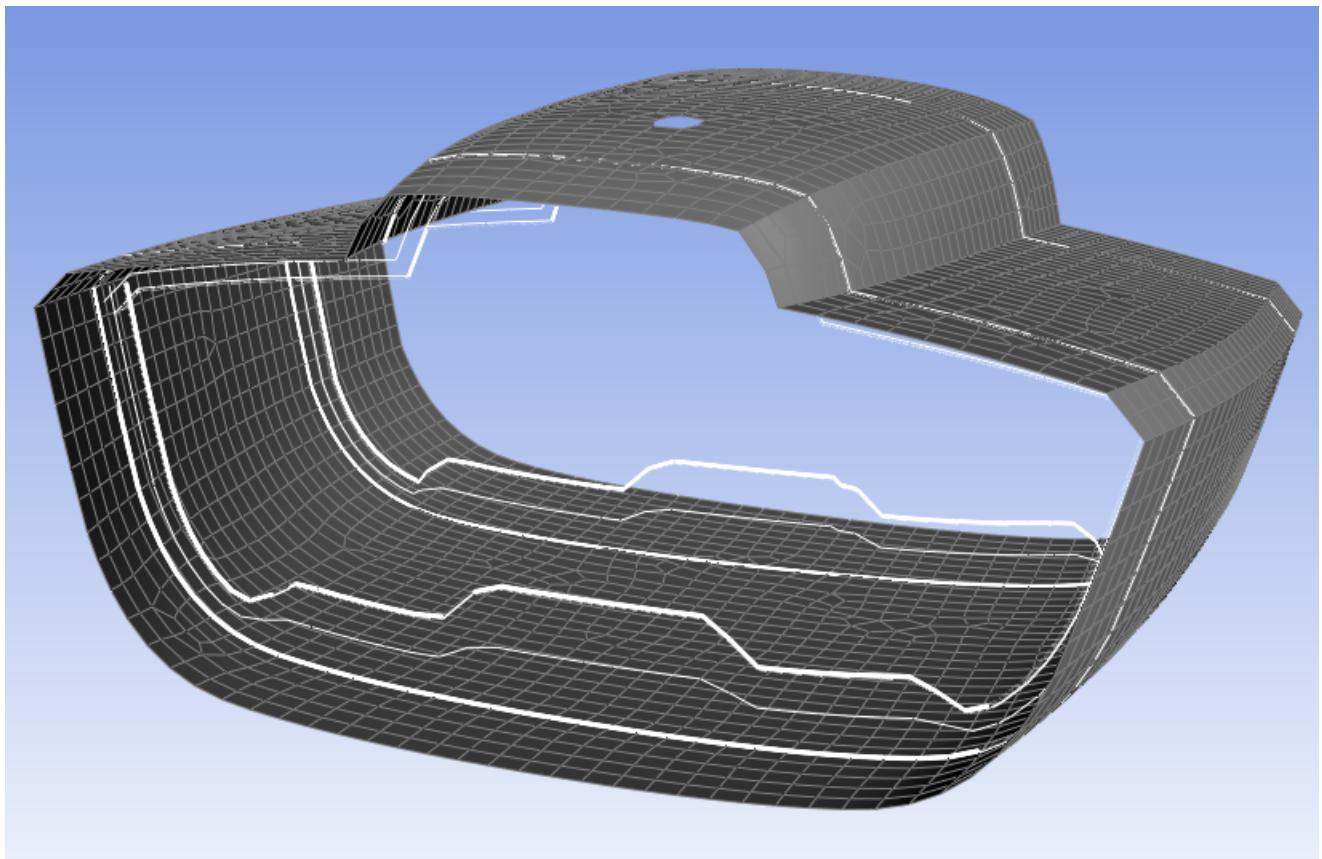
Figure 3.21: Modeling ply thickness definition

Figure 3.22: Section with variable core thickness

3.4.2. Look-Up Table

The variable core thickness can also be defined with a *Look-Up Table*. A *Look-Up Table* is used to define a data field or tabular values. Thicknesses, angles and directions can be defined in a *Look-Up Table* and the 3D mapping function of ACP inter- or extrapolates the values for each element. The user defines the thickness of the ply material for certain support points. The figure below shows a list of different angles and thicknesses for selected data points.

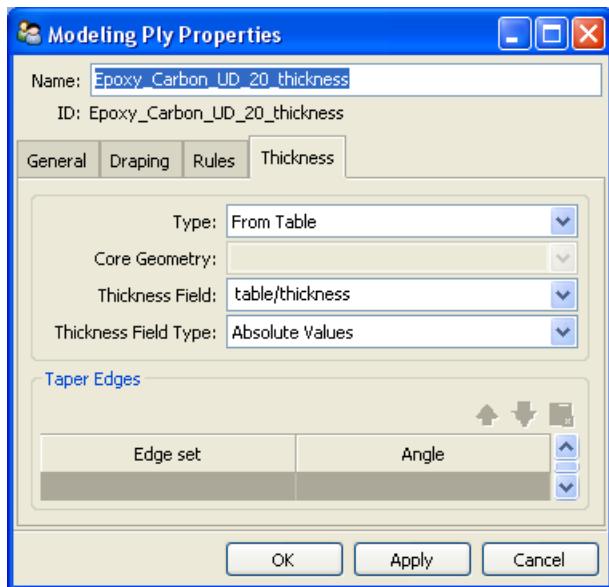
Figure 3.23: Table definition

Look Up Table Properties

Look Up Table Properties											
Name:	table	ID:	table								
Values		Interpolation									
I	Location.x	Location.y	Location.z	-20	-40	20	40	90	Radius	thickness	
0	-28.124110	-18.511083	-270.48	-27.72	-60.971187	27.726121	60.971187	90.000000	33.669359	0.272057	
1	-26.067880	-17.147761	-271.95	-30.13	-70.649864	30.134522	70.649864	90.000000	31.202244	0.293569	
2	-23.943567	-15.741248	-273.28	-33.13	-90.000000	33.138761	90.000000	90.000000	28.654516	0.319570	
3	-21.758262	-14.296281	-274.46	-36.99	-90.000000	36.990215	90.000000	90.000000	26.034700	0.351938	
4	-19.519142	-12.817651	-275.48	-42.13	-90.000000	42.130117	90.000000	90.000000	23.351426	0.392267	
5	-17.233583	-11.310281	-276.35	-49.45	-90.000000	49.456610	90.000000	90.000000	20.613549	0.444368	
6	-14.909218	-9.779162	-277.06	-61.46	-90.000000	61.466476	90.000000	90.000000	17.830221	0.513735	
7	-12.553803	-8.229478	-277.61	90.000000	90.000000	90.000000	90.000000	90.000000	15.010738	0.610230	
8	-10.176601	-6.667310	-277.99	90.000000	90.000000	90.000000	90.000000	90.000000	12.166192	0.752906	
9	-24.707560	-23.870949	-270.48	-27.72	-60.974917	27.727309	60.974917	90.000000	33.660143	0.272067	
10	-22.913148	-21.177828	-271.95	-30.13	-70.655507	30.135672	70.655507	90.000000	31.201165	0.293579	
11	-21.056861	-19.432697	-273.28	-33.14	-90.000000	33.140131	90.000000	90.000000	28.653466	0.319682	
12	-19.144093	-17.641511	-274.46	-36.99	-90.000000	36.991991	90.000000	90.000000	26.033629	0.351853	
13	-17.183554	-15.810295	-275.48	-42.13	-90.000000	42.132452	90.000000	90.000000	23.350374	0.392285	
14	-15.179303	-13.945173	-276.35	-49.45	-90.000000	49.459709	90.000000	90.000000	20.612555	0.444388	
15	-13.138765	-12.052382	-277.06	-61.47	-90.000000	61.471402	90.000000	90.000000	17.829387	0.513759	
16	-11.068702	-10.138251	-277.61	90.000000	90.000000	90.000000	90.000000	90.000000	15.010006	0.610260	
17	-8.977204	-8.210284	-277.99	90.000000	90.000000	90.000000	90.000000	90.000000	12.165482	0.752950	
18	-20.607203	-26.624604	-270.48	-27.72	-60.975714	27.727441	60.975714	90.000000	33.667883	0.272069	
19	-19.120301	-24.655157	-271.95	-30.13	-70.659700	30.136527	70.659700	90.000000	31.200364	0.293586	
20	-17.580155	-22.624084	-273.28	-33.14	-90.000000	33.141045	90.000000	90.000000	28.652153	0.319697	
21	-15.991911	-20.504681	-274.46	-36.99	-90.000000	36.994827	90.000000	90.000000	26.031918	0.351876	
22	-14.360778	-18.409642	-275.48	-42.13	-90.000000	42.136877	90.000000	90.000000	23.348380	0.392318	
23	-12.692104	-16.230061	-276.35	-49.46	-90.000000	49.466724	90.000000	90.000000	20.610437	0.444435	

Following on, the corresponding tabular field can be selected in the *Thickness* tab of the *Modeling Ply* property dialog.

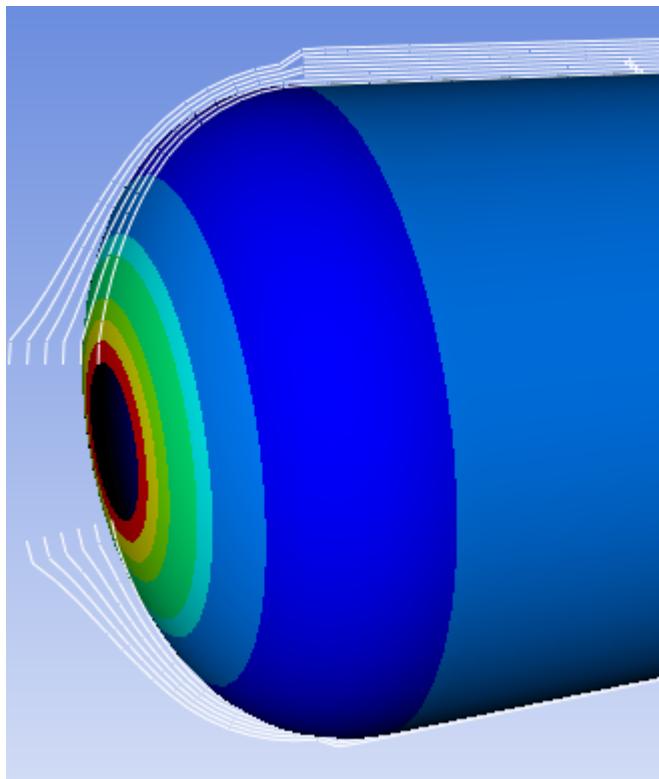
Figure 3.24: Thickness definition through a tabular values



The final result can be investigated with *Section Cuts* or a thickness contour plot as per usual.

3.4.3. Geometry Cutoff Rule

A further possibility of achieving a variable core thickness is use a Cutoff Rule. Even if the cutting operation only applies to a core layer it is dependent on the entire layup. If the thickness of the bottom laminate is changed the thickness of the core is cutoff at a different height. In this way, a laminate thickness limit can be set. This can be a very useful in places where the laminate thickness is limited, near a trailing edge of a blade for example.

Figure 3.25: Section cut and thickness contour plot

The core thickness can be set to be cut in different ways depending on the Ply Tapering option of the Rule. It can either follow the exact intersection with the CAD geometry or can be cutoff to two discrete size - its nominal thickness or no thickness at all.

The cutoff Rule has to be used with precaution as any modification of the underlying plies might modify the core.

An example of a *Cutoff Rule* can be found in *Tutorial 2*.

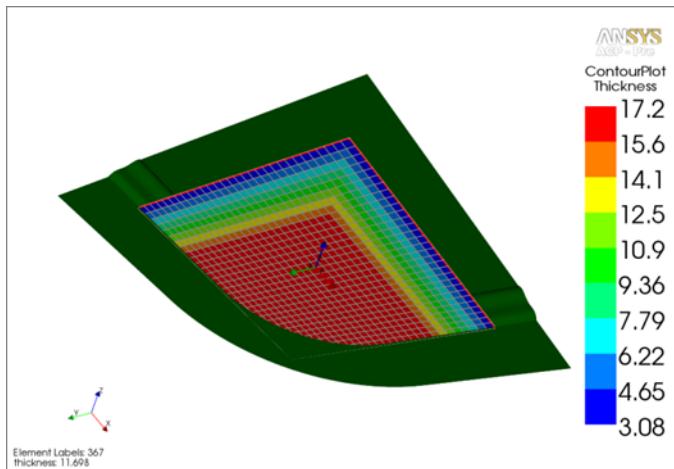
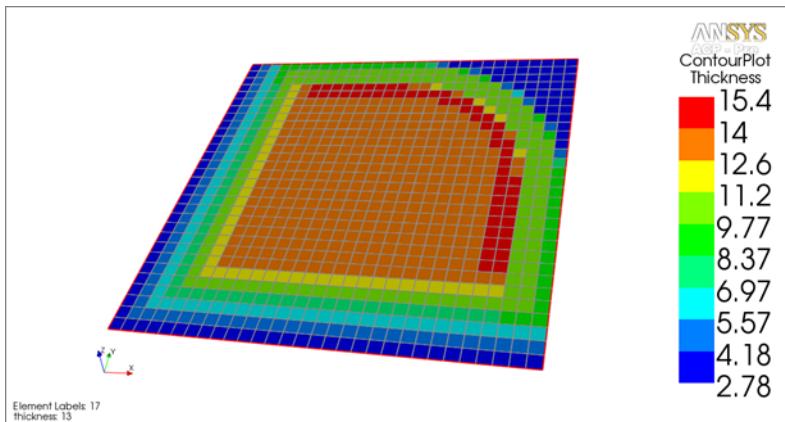
Figure 3.26: Imported Cutoff Geometry

Figure 3.27: Resulting thickness distribution (Ply Tapering activated)

3.4.4. General Application

The described features can also be used in combination with regular or woven materials and are not restricted to core materials. The selection of the method is often derived from the manufacturing process. Tabular values can be used for a winding process and CAD geometries for a CNC milling process. A *Cutoff Rule* is often used in regions of sharp tapered edges (trailing edge of a wind turbine blade).

3.5. Draping

The ply application (draping) on doubly curved surfaces changes the theoretical fiber orientations. In many cases the effect is small and can be neglected. On the other side it is important to know how big this effect can be and if it has to be considered. In that case ACP allows to evaluate the draped fiber directions. These angles can be visualized and are considered in all analysis resulting in more accurate evaluations. The draping is evaluated on *Production Ply* level. In addition the draping algorithm of ACP evaluates the flatwrap of the *Production Plies* which can be exported for manufacturing purposes.

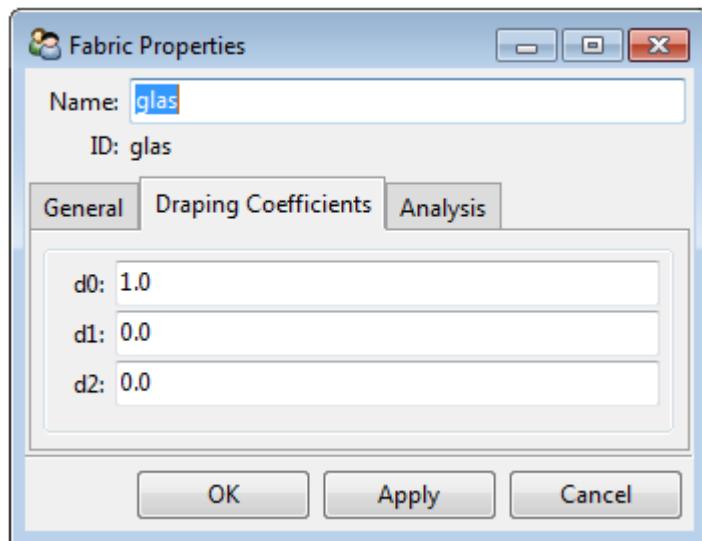
3.5.1. Internal Draping Algorithm of ACP

The draping algorithm and the meaning of the draping coefficients are described in Section [Draping Simulation](#). The required data to define draping functionalities are:

Draping Coefficients

The three draping coefficients are defined in the *Draping Coefficients* tab of the *Fabric and Stackup* property dialog. These coefficients are relative values of different energy modes. More details can be found in the Section [Draping Simulation](#).

Figure 3.28: Draping coefficients of a Fabric



Note: *Fabrics* and *Stackup* have their own draping coefficients because a non-crimp fabric (**Stackup**) behaves different than the sum of its individual layers.

Draping Method Definition

The draping effect highly depends on the manufacturing process. A few process relevant values can be defined in the ACP draping algorithm.

Seed Point

The *Seed Point* is the starting point where the ply is laid into the mold. At this location the fiber direction is unchanged and the draped fiber direction is equal to the theoretical one. The *Seed Point* can have a big influence on the final result of the draped fiber angles. Assuming a half sphere and a *Seed Point* located on the pole, the maximum draped fiber angle is much smaller than the same evaluation with a *Seed Point* on the equator. The seed point corresponds to the first element of the draping mesh (left representation in Figure [Draping scheme](#)).

Draping Direction

After the first point is applied on the mold, the *Draping Direction* defines along which route the ply is laid into the mold. The draping algorithm first walks along the *Draping Direction*, then orthogonal and finally proceeds with the 45-degree zones. Figure [Draping scheme](#) shows the scheme in which the ply is applied.

Draping Mesh

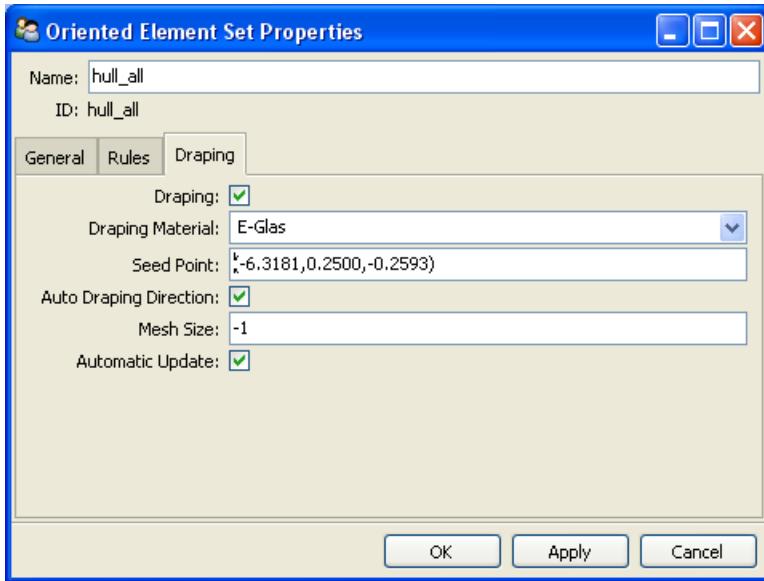
The draping algorithm minimizes the shear energy dissipation where an internal *Draping Mesh* is used for the evaluation. This mesh is independent from the structural mesh and has its own size. Analog to the structural mesh, the optimal *Draping Mesh* size is derived from the balance between the precision of the draping evaluation and the computational cost.

In the case of an incomplete draping, select an other *Seed Point*, define a different *Draping Direction* and/or change the *Draping Mesh* size. The draping mesh is built as shown in Figure [Draping scheme](#)

Draping Definition on OES Level

The draping can be activated in the *Draping* tab of the *Oriented Element Set* property dialog.

Figure 3.29: Draping definition in OES



If draping is activated on the OES level, draping is evaluated for all plies linked with this OES. If the *Modeling Ply* has also an active draping definition, the OES are obsolete for this ply. The draping coefficients are given through a *Draping Material* (Fabrics or Stackup). This means that the *Draping Coefficients* of this material are used and not those of the *Modeling Ply* material. This allows to define a *Draping Material* just used for draping purposes.

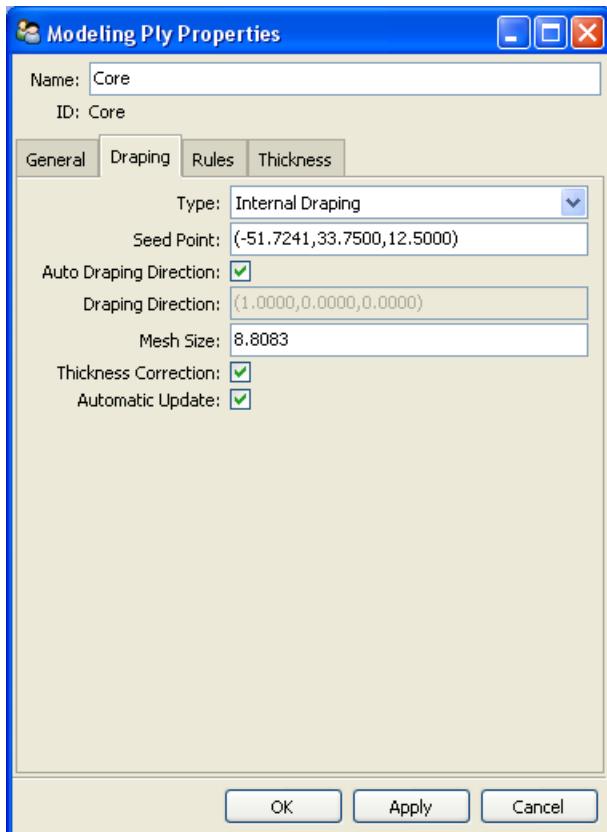
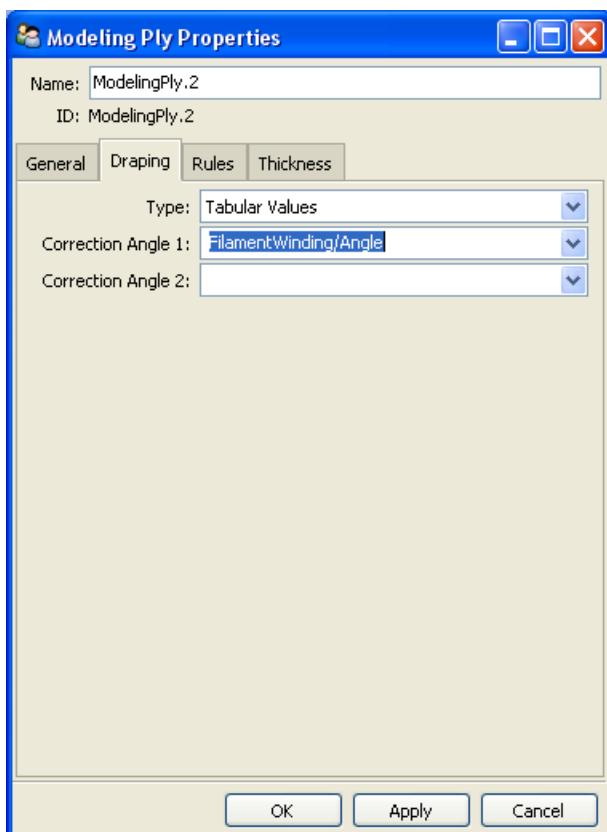
Draping Definition on Modeling Ply Level

The same definition as on the OES level can be done on the *Modeling Ply* (MP) level. The only difference is that the MP material also is the *Draping Material*. To active the draping you toggle the *Draping* check box and define a *Seed Point*. Per default, the *Mesh Size* and *Draping Direction* are evaluated automatically.

An additional feature *Thickness Correction* is implemented in the *Internal Draping* algorithm. Due to the shear deformation the fiber direction and thickness of the ply change. This change can also be considered by activating the *Thickness Correction* option.

3.5.2. User-Defined Draping

ACP can also handle user-defined draping results. The draped fiber directions can be imported as *Look-Up Table* and used instead of *Internal Draping*.

Figure 3.30: Internal draping definition**Figure 3.31: Tabular values definition of draping**

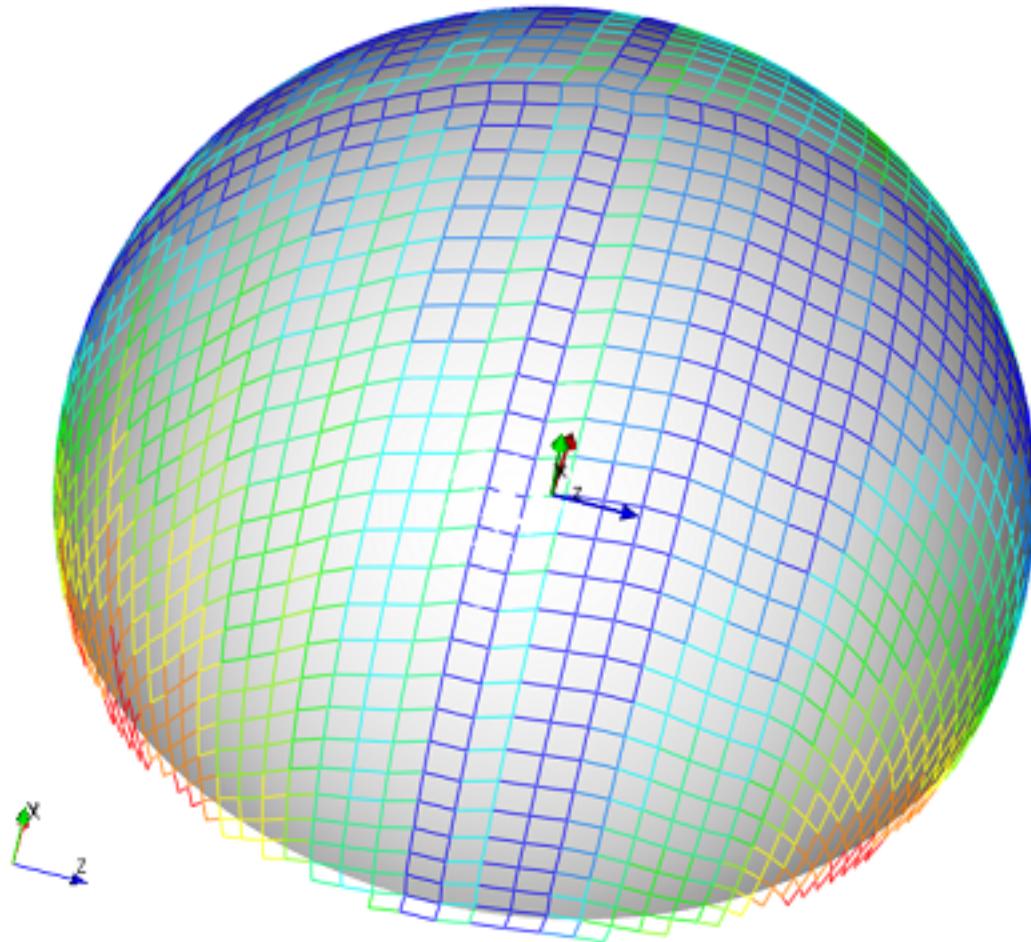
The first angle *Correction Angle 1* defines the correction of the material reference direction and is also considered in the analysis. The second value *Correction Angle 2* can be used to define the correction angle of the material 2 direction for woven materials. In ACP the second correction is not considered in any evaluation and is just for information or third party products.

3.5.3. Visualization

The result of the draping evaluation can be visualized on *Production Ply* and *Analysis Ply* level. The flatwrap and the draping mesh can be visualized. You just activate *Toggle Draping Mesh* and *Toggle Flatwrap* in the toolbar (see [Toolbar](#)). All other configurations are defined by default and can be modified in the *Draping* tab of the active scene. The contour plot of the draping shows the average shear (distortion) angle of each element (in degree). Zero means no shear deformation. Depending on the scene configuration the flatwrap is also available in the *Ply Book* (see [Ply Book](#)). In addition the *Production Ply* functionality allows to export the flatwrap as *.dxf.

The last result of the draping are the draped fiber directions which are considered in the analysis. These directions can be visualized with the button *Show Draped Fiber Directions*. This visualization combined with *Show Fiber Directions* highlights the influence of the draping.

Figure 3.32: Draping Mesh with Shear Energy



3.6. Ply Book

A *Ply Book* is the easiest way to forward the production data to other actors of the project (designers, manufacturers and others). It is a good medium to exchange information generated automatically.

A *Ply Book* is separated in different chapters. In the automatic setup, a chapter is generated for each *Modeling Ply Group*. Each chapter has its own *View*. This allows to predefine a *View* showing the details of this specific section of the model. Therefore the first step is to generate a new *View* through the button in the *Toolbar* or with the right mouse click in the tree-view.

Figure 3.33: Flatwrap (boundary)

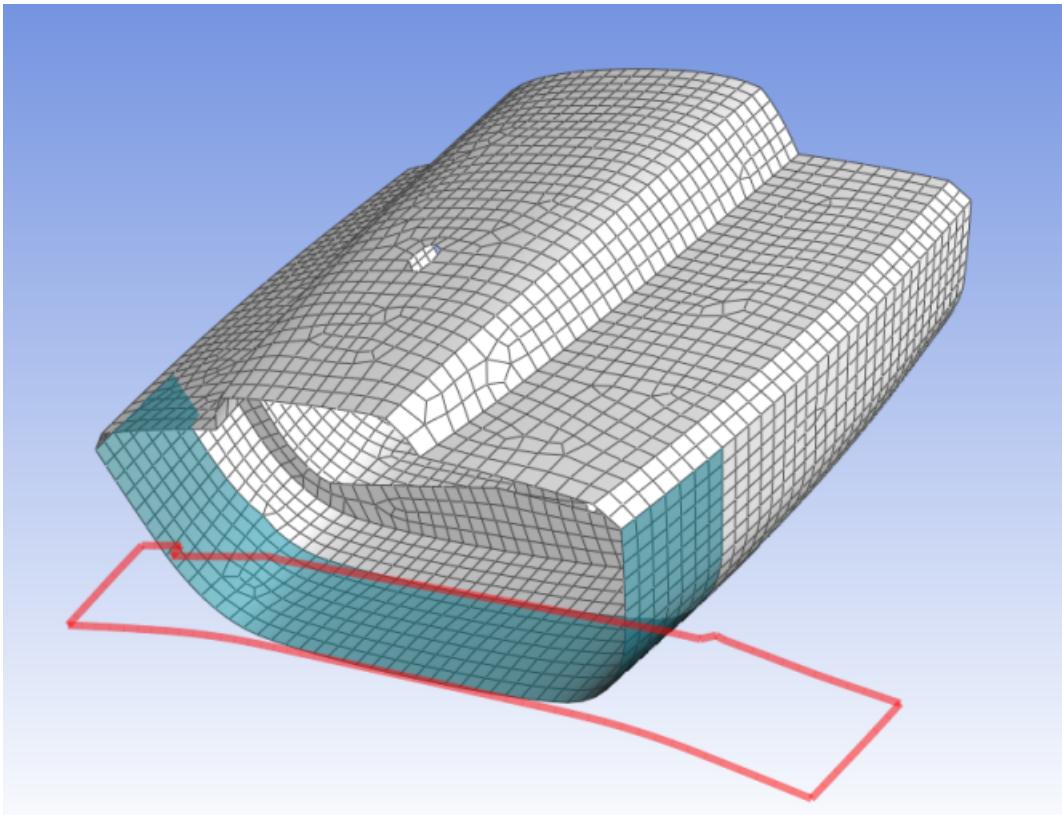
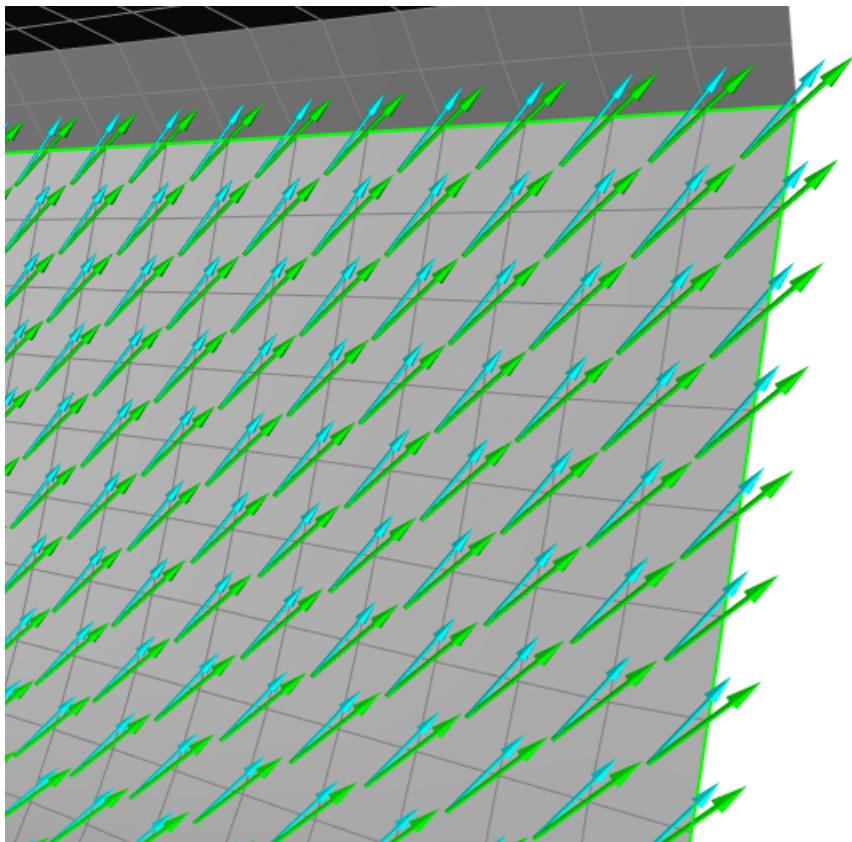
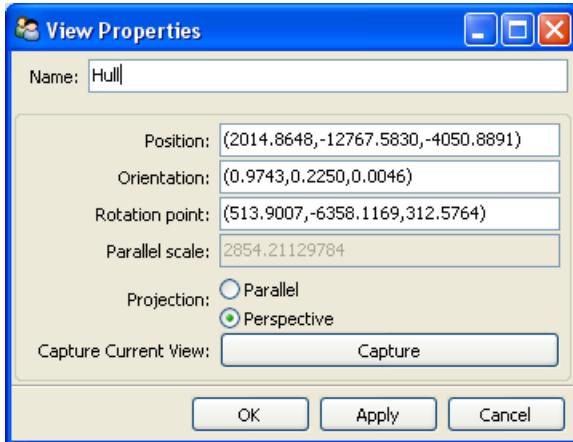
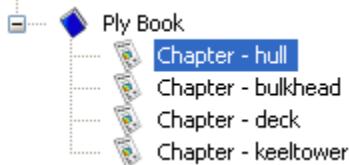
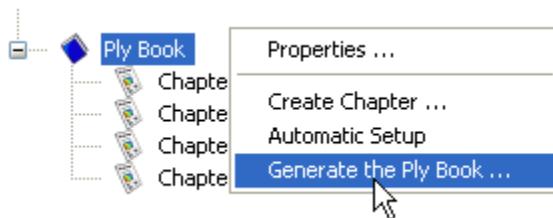


Figure 3.34: Fiber and draped fiber directions**Figure 3.35: View definition**

Create the chapters with *Automatic Setup* from the right click menu, or define your own chapters.



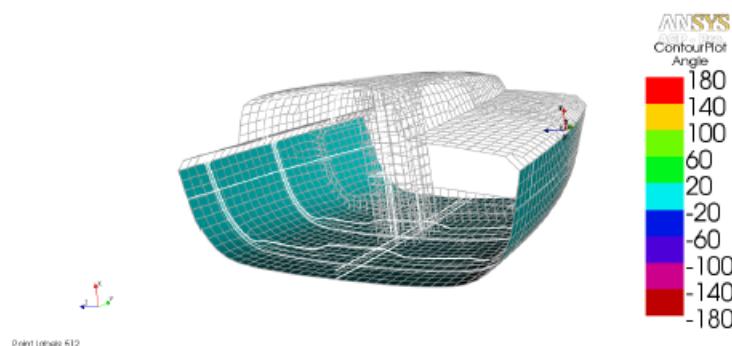
After the chapter definition, use the right click menu to *Generate the Ply Book* in the desired format (*.html, *.pdf, *.odt or *.txt).



In the example class40, delivered with the installation, a *Ply Book* with different chapters and different views is already defined. Check its definition and export the *Ply Book*. Note that the export can take several minutes. The image size used in the *Ply Book* can be defined in the global *Scene Preferences* (Tools - Preferences - Scene).

Figure 3.36: Example of a Production Ply representation

outer_skin_9



- Ply Group Name: hull (hull)
- Modeling Ply Name: outer_skin_9 (outer_skin_9)
- Ply Name: P1_outer_skin_9 (ProductionPly.8)
- Orientation: 0.0
- Material: E-Glas

Parameter	Value
Thickness	0.27
Area	18116375.4818
Cost	0.0
Weight	4891421.38007

3.7. Guide to Solid Modeling

In the case of thick composites, the layered shell theory can cause significant errors in the obtained results. In some cases it is necessary to work with 3D models - *Solid Models*. ACP has the unique feature to generate layered solid models based on the shell layup definitions. Based on the shell mesh and the *ACP Composite Definitions*, ACP generates layered solid elements representing one-to-one the composite part. Drop-offs, staggering and tapering are also considered. In addition the *Solid Model* extrusion allows to define extrusion directions and boundary curves.

The following section aims to give a brief guide to Solid Modeling.

3.7.1. When to use a Solid Model

From an analysis point of view, the choice between a shell or solid element analysis models largely depends on the structure and the type of structural investigation.

Solid models are inherently larger and computationally more expensive than shell models. It is therefore wise to start an analysis with a shell model before moving on to a Solid Model. It provides a basis for comparison but it is also a good check whether the model is solvable.

Typically, a Solid Model describes the behavior of a structure more precisely when its out-of-plane response becomes significant. ACP has the unique feature of representing the three dimensional stress state for a shell model. Shell model stress behavior can therefore be taken as a first indication of the 3D stress state. If the out-of-plane stresses are significant then it may be worthwhile analyzing the structure as a solid element model.

The following list shows cases where a solid elements analysis model can be used. The Solid Modeling feature is, however, by no means limited to these examples.

- Analysis of thick structures
- Investigation into 3D stresses (high element resolution necessary)
- Investigation into debonding
- Investigation into edge effects
- Buckling analysis of sandwich structures

There are no hard rules on this matter. It remains entirely the choice of the designer when to use a Solid Model in addition or instead of a shell model.

3.7.2. How to use the Solid Model Feature

The intention of the Solid Model feature is to generate analysis models for structures that are built in one piece.

The feature itself makes no distinction between generating solid element models that are analyzed in isolation or ones that are analyzed in combination with other components. There is however a distinct difference in how the analysis of a solid element model of multiple components can be approached. The components of an assembly can either be extruded individually or extruded as one assembly. Both approaches are possible within ACP yet they both have advantages and limitations.

The recommended approach is to generate individual components and connect them using contacts in a Workbench Analysis System. This follows the intention that Solid Models are only created for components that are to be built in one piece. Additional connecting structures can be also fully modeled or dimensioned with the help of substitute model.

The other approach is the extrusion of an entire assembly in one Solid Model generation. This method is not only limited by the topological complexity of a geometry but also by a reduced stiffness at transitions as a result of drop-off elements. On the other hand, this approach offers the ability to model all connecting structures in full.

3.7.3. Principle of the Solid Model Generation

The Solid Model feature requires a reference shell geometry and a composite definition to construct a solid element model. The feature has additional ways to enhance the resulting Solid Model to be as detailed as necessary: Ply staggering and tapering are transferred from the composite definitions if they are activated. Extrusion guides add more complex possibilities in shaping the model. The Snap-to functionality makes an alignment with an external CAD geometry possible.

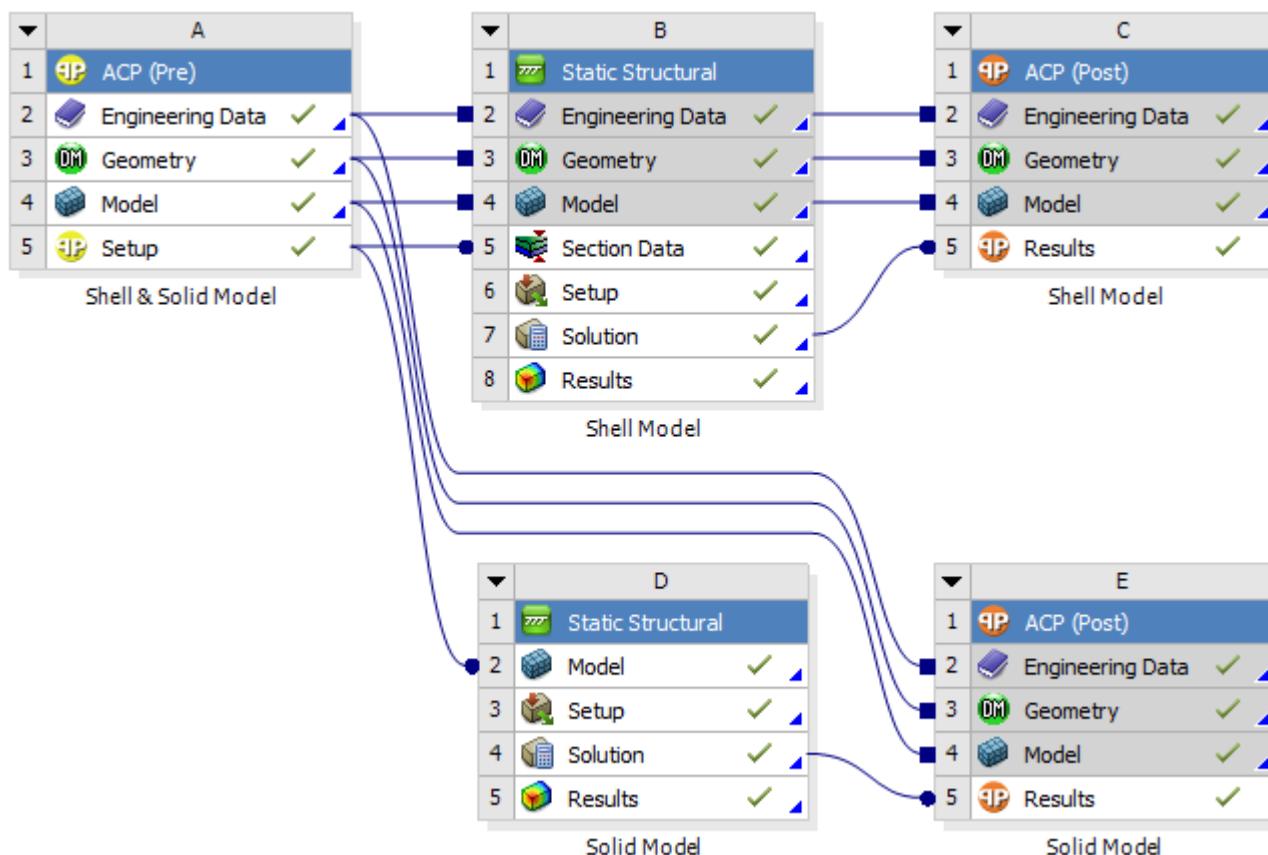
Settings control how the Solid Model is divided into elements in the thickness direction. The level of detail required in the Solid Model depends on how accurate certain features are to be modeled. This is down to the judgment of the designer.

Details of the Solid Model feature are explained in the [Usage Reference](#).

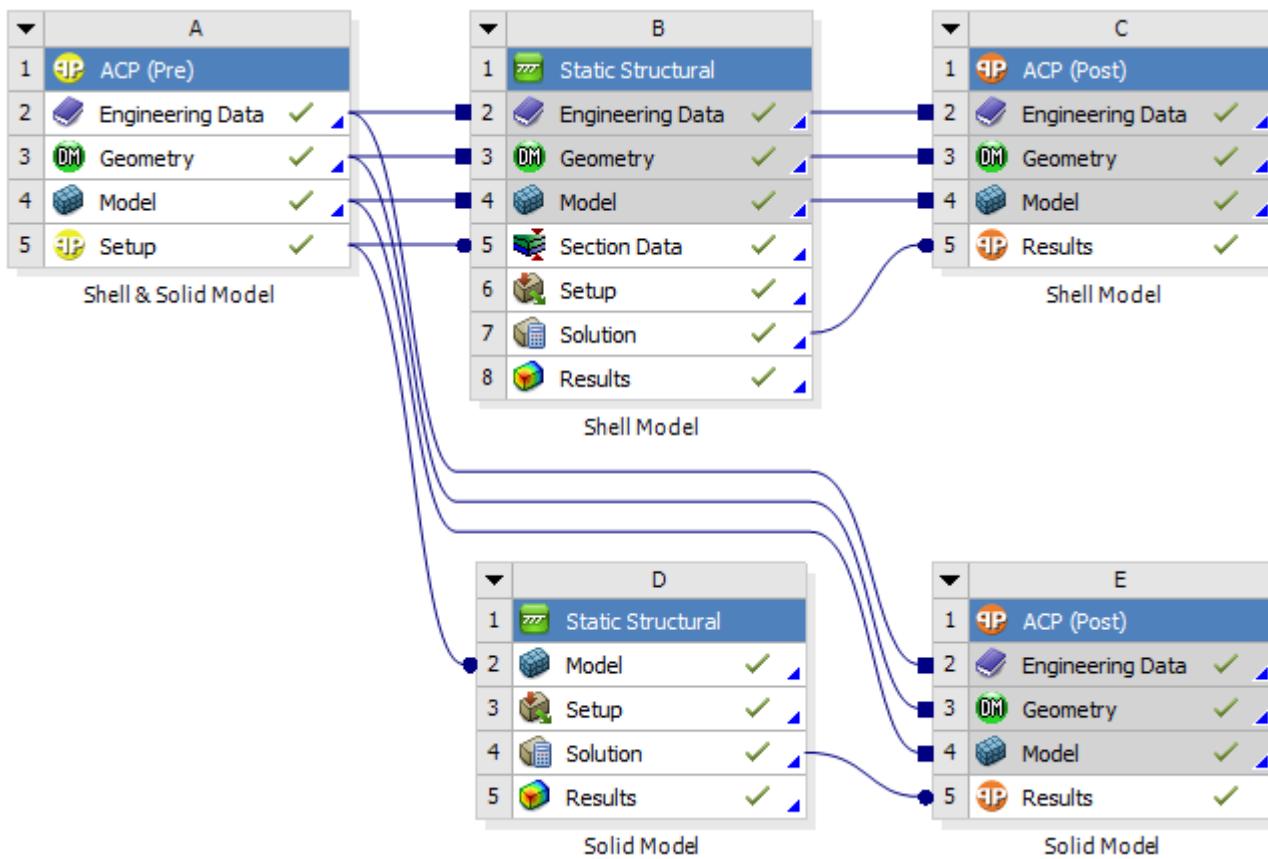
3.7.4. Workflow

A Solid Model is easily created alongside an existing shell model in the Workbench logic. Once a Solid Model has been generated in ACP (Pre) it can be linked to new Analysis System. A single ACP (Pre) system can be used to create a shell and a Solid Model analysis. This workflow is shown below:

Figure 3.37: Analyzing a Solid Model alongside a shell model



The analysis of an assembly requires multiple Solid Model components to be connected in a Workbench Analysis System. The components can be connected with contacts in Workbench Mechanical for example. As a consequence, the connections can be modeled and analyzed in detail. One limitation is that no composite post processing for the complete assembly is available. An example of a Solid Model assembly is shown below.

Figure 3.38: Solid model assembly workflow

3.7.5. Practical Tips

The Solid Modeling process is generally more difficult than shell modeling is. The shape and mesh of a structure have a strong influence on the robustness of the Solid Modeling process. Generally speaking, the less complex the model the more robust the process. The challenges of composite modeling can vary greatly. Following composite design principles for structural concepts will aid the solid modeling process. Abrupt changes in shape and sharp edges are not advisable in composite design and also cause problems in composite modeling.

3.7.6. Known Limitations

While ACP offers many advanced modeling features it also has some limitations. This section aims to give a brief overview:

Mesh Extrusion

Despite many enhancing features it is worth stating that the Solid Model relies on the extrusion of a shell geometry mesh. As the extrusion directions and operations increase the Solid Model generation reaches a limit of how heavily a shell mesh can be distorted. If the topology of the structure is complex then the extrusion operations can result in ill-formed element which are subsequently deleted in the element check.

Drop-Off Elements

There are transition regions where an edge of a Solid Model extrusion is reduced to a series of drop-off elements. This reduction in thickness will result in a local reduction in stiffness that should not be overlooked.

Connect Butt-Jointed Plies

The ability to connect adjacent plies is currently restricted to plies that appear sequentially in the same modeling ply group. Consequently, there a certain arrangement where a ply drop-off cannot be evaded.

Solid Model Extrusion Offset

The Solid Model extrusion starts from a reference shell and the layup definition. An extrusion with an offset to the reference geometry is currently not possible.

3.8. Guide to Composite Visualizations

There are several features available in ACP that help visualizing a composite model. Some features aid in the verification of the layup definition. Others provide an insight into the stresses, strains and failures and can thus help in the optimization of a structure.

This section aims to be a brief guide to the some of the available functionality.

3.8.1. Model Verification

Two features are very useful for checking the layup definition before solving the model. One is the Orientation Visualization, the other is the Section Cut feature.

The Orientation Visualization can display the direction and / or orientation of the elements, Oriented Element Sets and plies. Refer to [Scene Manipulation](#) for more information.

After the whole definition of the lay-up, a visual verification of the lay-up sequence can be very useful. Section Cuts offer a useful visual check once the layup has been determined. By defining one or several Section Cuts, the ply position (number in the sequence) and orientation can easily be verified. Refer to [Section Cuts](#) for more information on Section Cut definition.

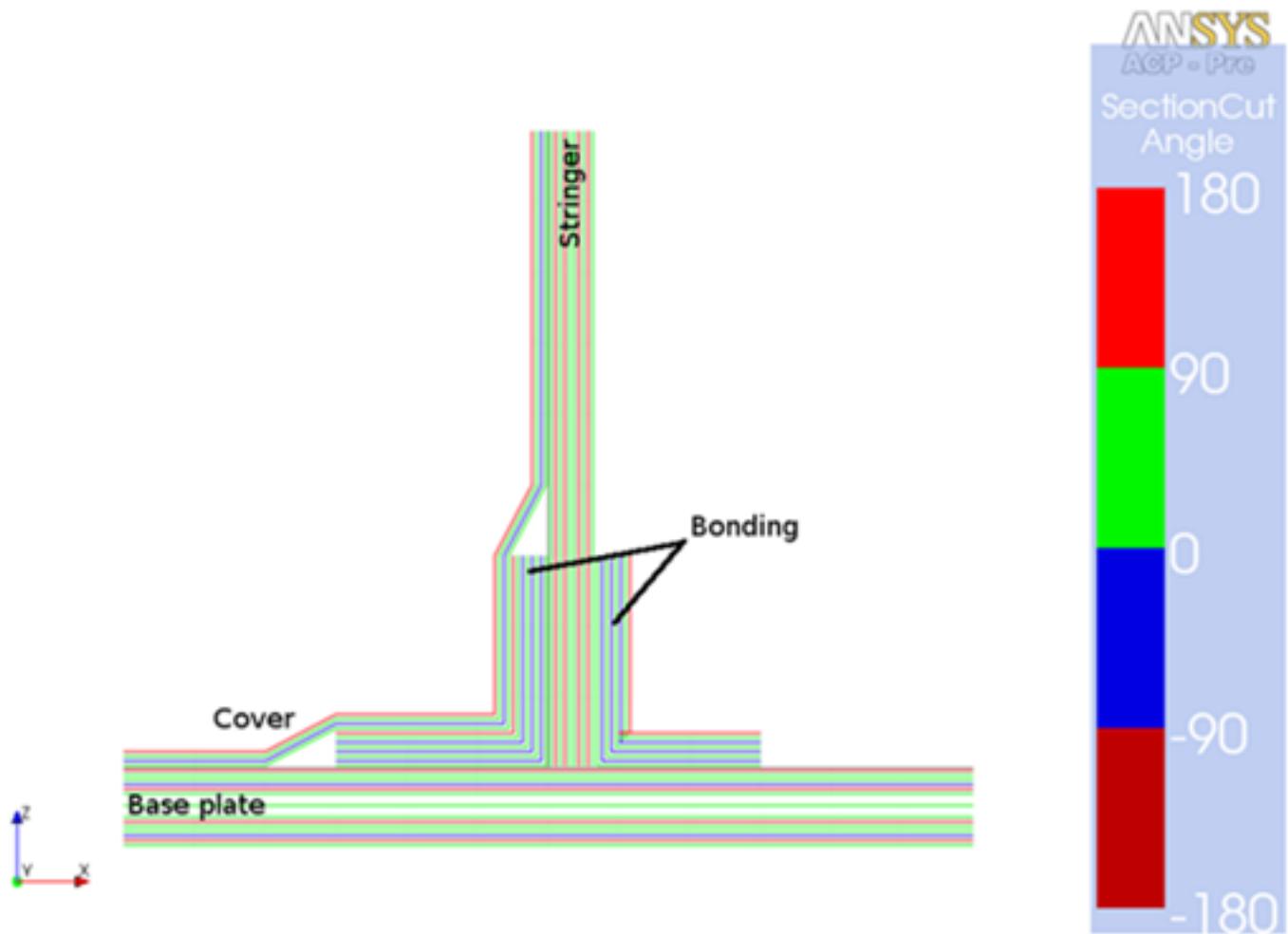
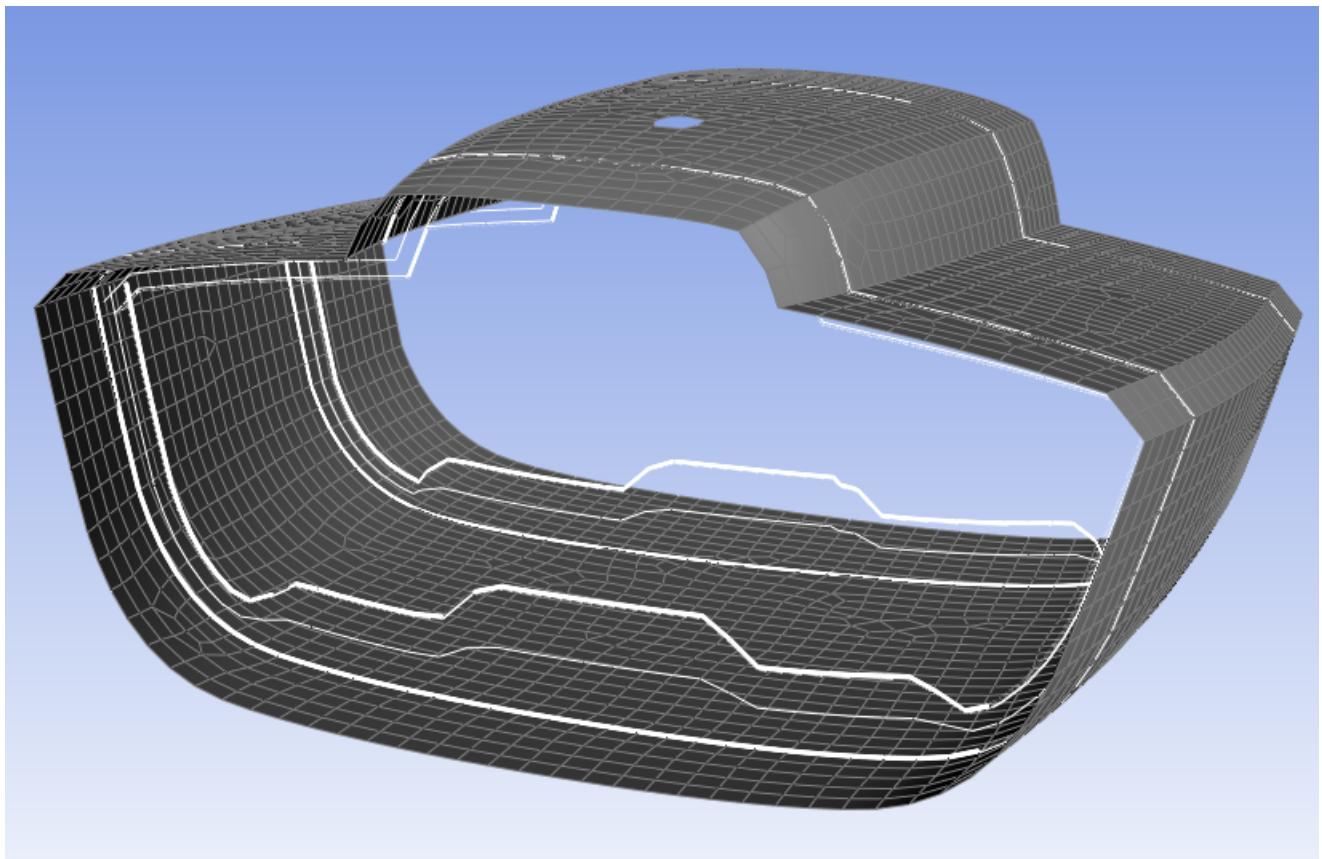
Figure 3.39: T Joint Section cut

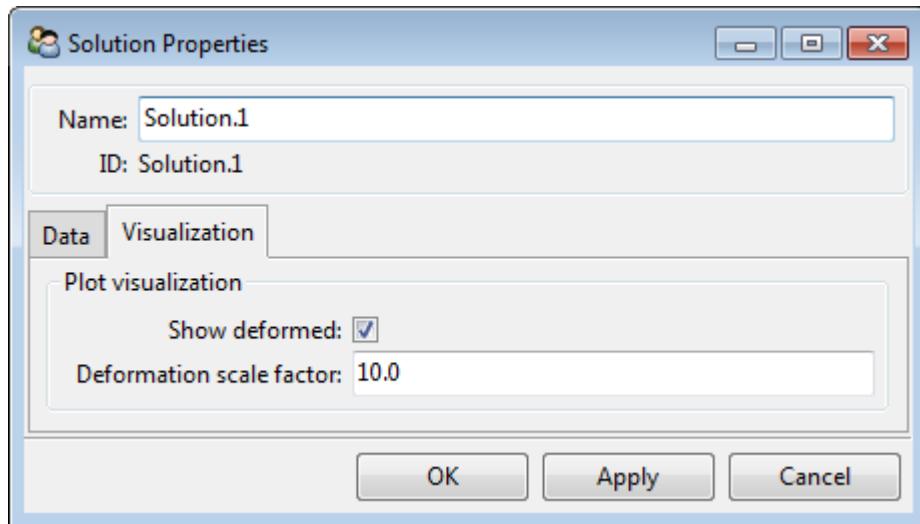
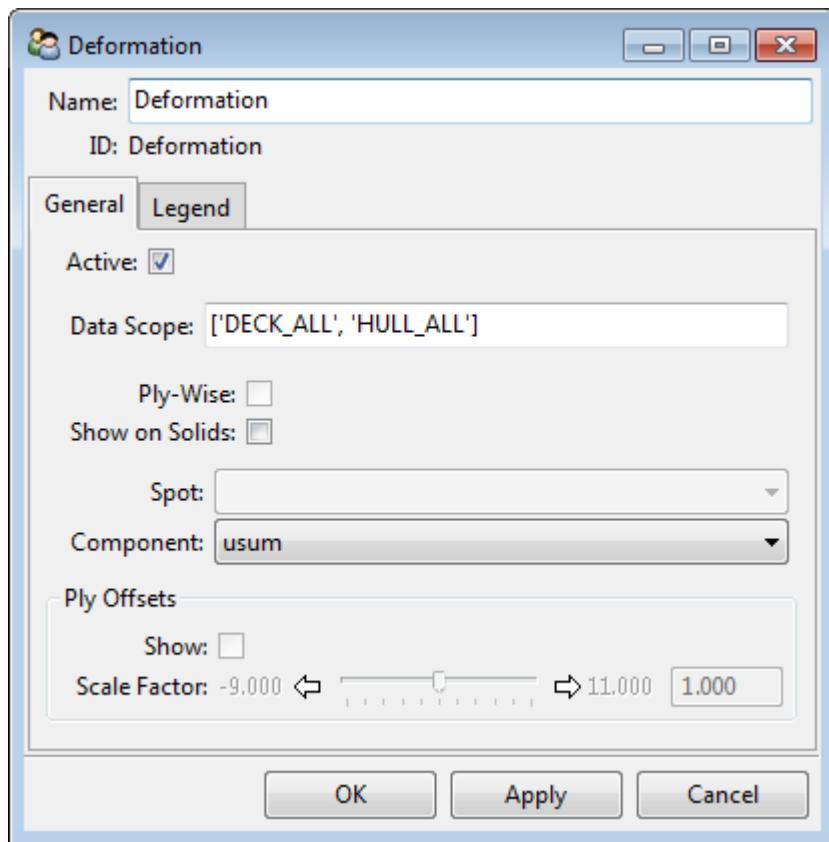
Figure 3.40: Class40 Section Cut

3.8.2. Post-processing visualizations

There are several features that can be used for viewing the results of a simulation. They all can give a different insight into the behavior of the composite structure. The different post-processing visualizations mentioned below are also described in the last part of Tutorial 1.

Deformation

The deformation of a structure can be visualized with deformation plot for a specific solution (see [Solution Plots \(p. 195\)](#)). The plot can be scaled by setting the deformation scale factor in the solution properties.

Figure 3.41: Activate deformed geometry in the solution properties visualizations**Figure 3.42: Activate the deformation plot for total deformation**

Failure Criteria

The failure plot displays the critical safety factors (reserve factors, inverse reserve factors & margin of safety) to first ply failure for a given failure criteria definition.

The safety factors are evaluated for every element and every layer and the critical value through the thickness of the layup is then projected on to the reference shell mesh. A failure plot for an envelope

solution works in the same way and is a superposition of more than one solution failure plots. Alternatively, the safety factors can be displayed ply-wise for each analysis ply.

First of all, a failure criteria definition has to be defined before creating a failure plot (see [Definitions \(p. 188\)](#)). A failure plot can be inserted under a normal solution or an envelope solution and the predefined failure criteria definition can be selected. Additionally, critical failure modes, critical plies and critical load case (in case of solution envelope) can be displayed as element labels.

Figure 3.43: Activate the Failure Criteria Plot with failure mode and critical Ply information

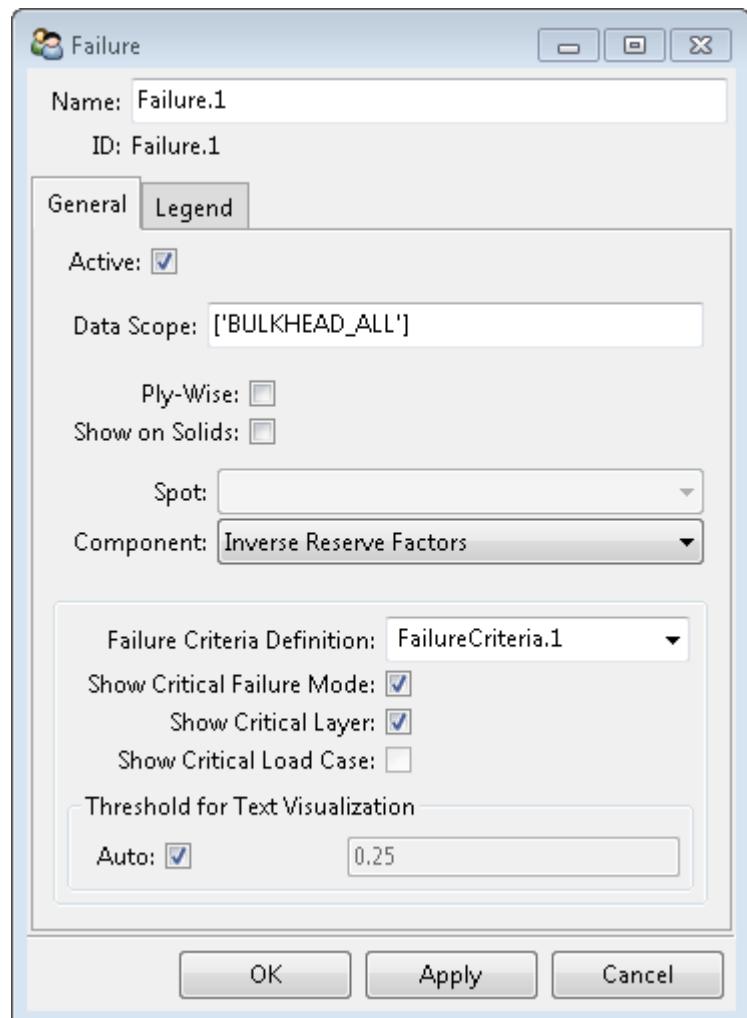
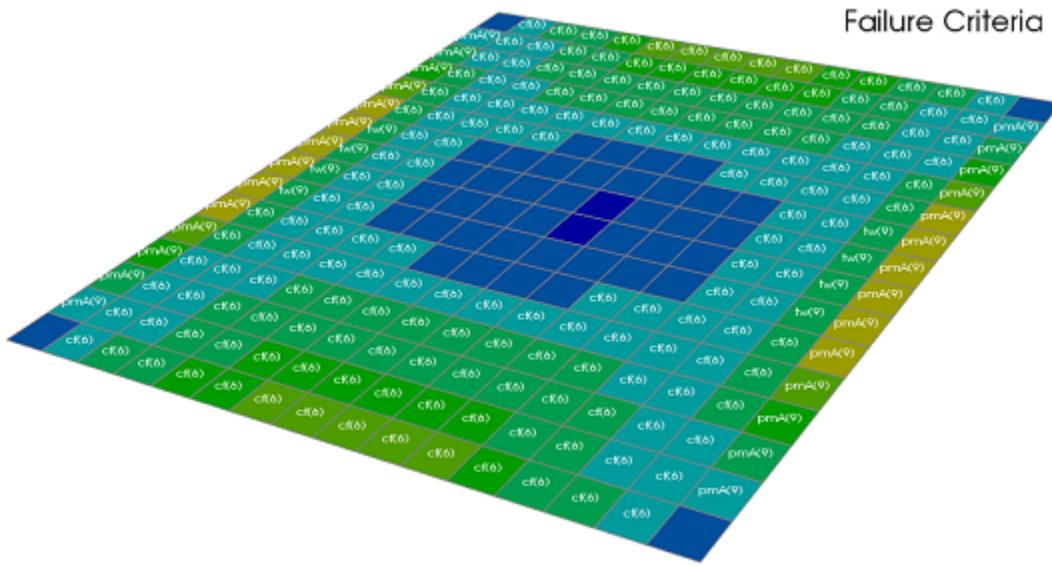
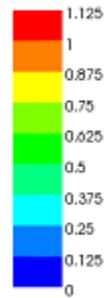
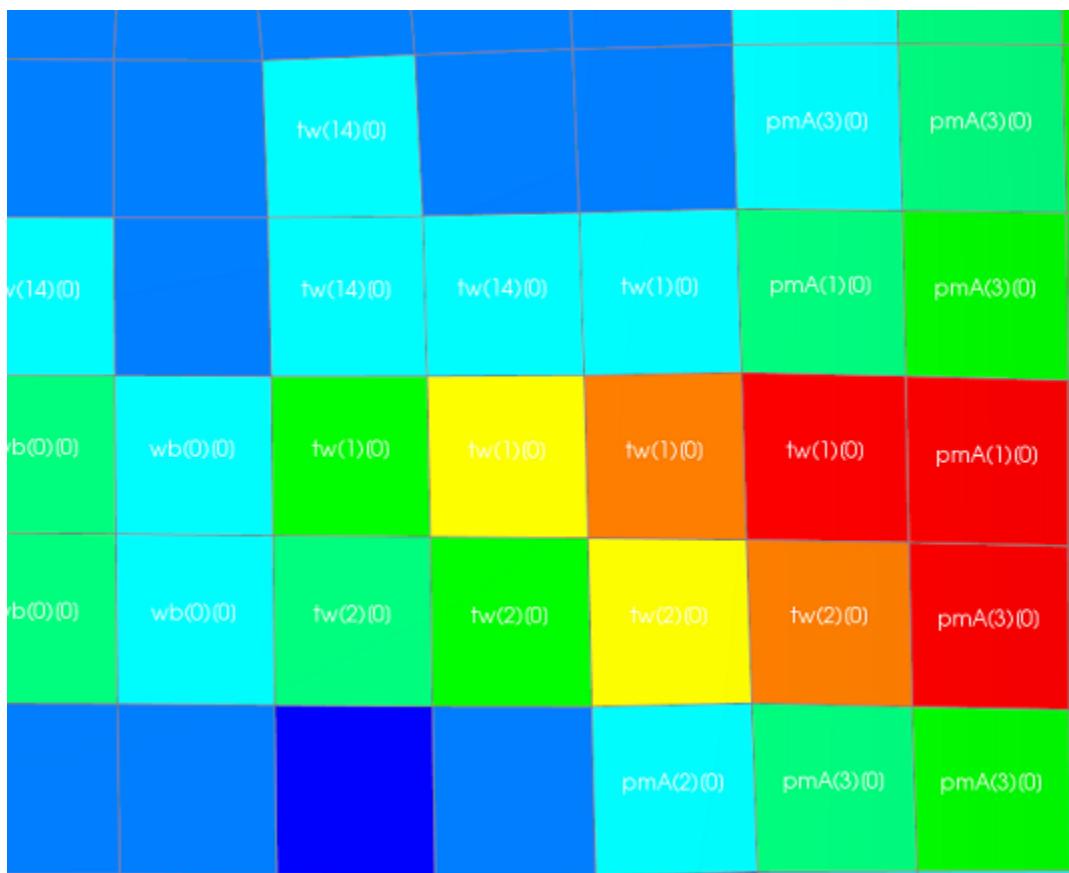


Figure 3.44: IRF value and Text plot for each element (Tutorial 1)

ACP Model

Failure - If
Element-Wise
Max: 0.85115
Min: 0.12445

Failure,1

**Figure 3.45: Zoom on critical area (Class 40)**

Ply Wise Results

The structural behavior throughout the layup at each layer is of great interest in composite design. Ply wise information helps to identify, if not optimize, layers that are critical and ones that are not.

All the solution plots except the deformation plot have the option of displaying results ply-by-ply (see [Solution Plots \(p. 195\)](#)). The plot will only display results if a ply is selected. Plies can be selected in the Modeling Ply Groups, Sampling Element or Solid Model Analysis Plies.

Figure 3.46: Activate the ply-wise results in the plot properties

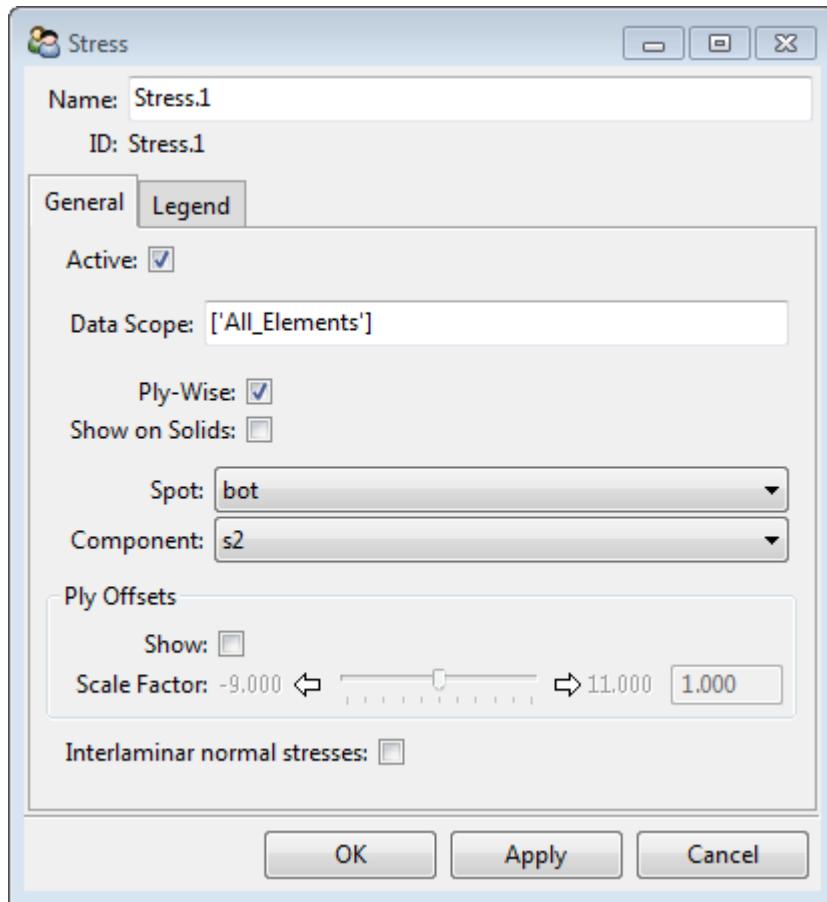


Figure 3.47: Select an Analysis Ply in the Modeling Ply Groups or Sampling Elements

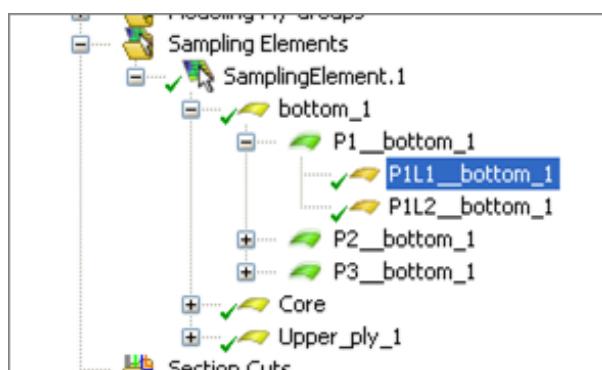
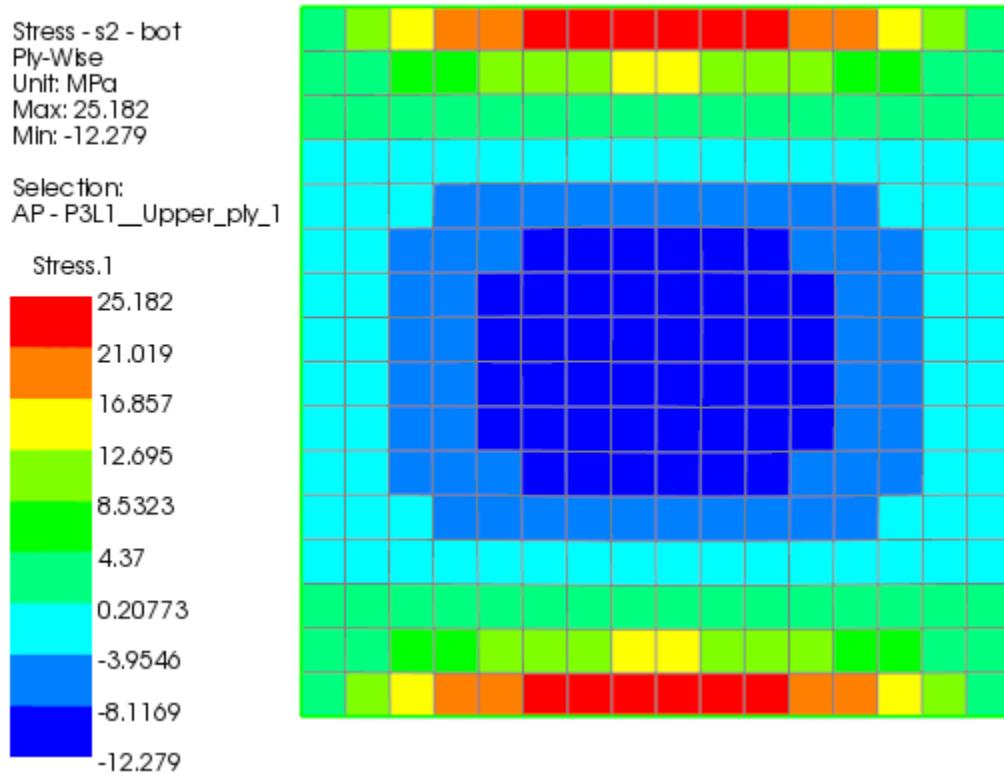


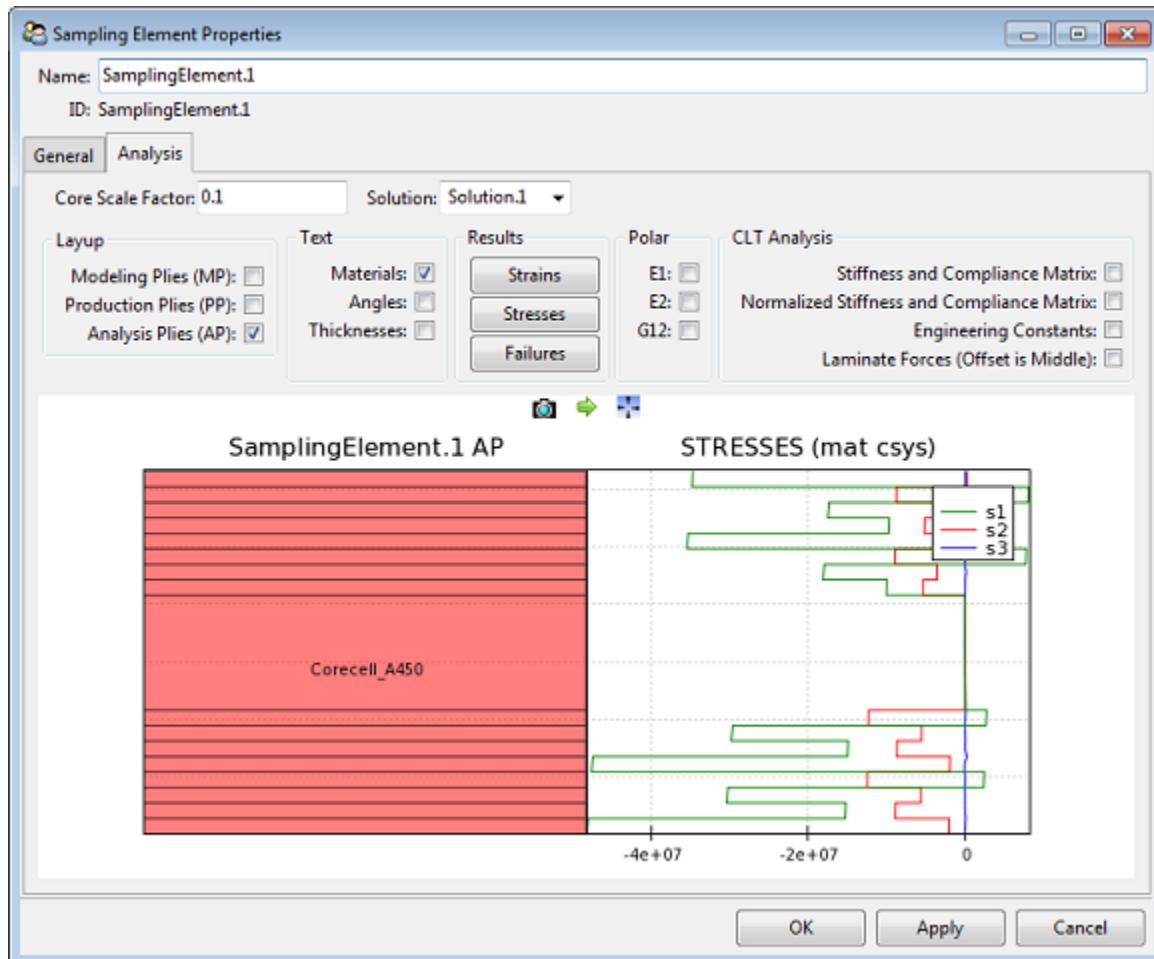
Figure 3.48: Ply-wise stress (Tutorial 1)

ACP Model



The use of [Sampling Elements](#) is an alternative way of analyzing a layup on a ply level. A point of interest on the composite part is selected and its local layup is sampled. The feature can display failure criteria, stresses and strains through the thickness of the laminate for a given solution. In this way, the Sampling Element gives a detailed insight into the laminate behavior ply-by-ply.

The [Sampling Elements](#) is an alternative way of analyzing a layup on a ply level. A point of interest on the composite part is selected and its local layup is sampled. The feature can display failure criteria, stresses and strains through the thickness of the laminate. In this way, the Sampling Element gives a detailed insight into the laminate behavior ply-by-ply.

Figure 3.49: Stress analysis for selected Sampling Element

3.9. Element Choice in ACP

This section describes the composite modeling techniques for the use of shell, solid and solsh elements.

3.9.1. Introduction

3.9.2. Shell Elements

3.9.3. Solid Elements

3.9.4. Solid Shell Elements

3.9.1. Introduction

The underlying principle of ACP is that a composite lay-up is defined on a shell geometry. The model of the lay-up that is passed from the ACP preprocessor to the solver can be a shell element mesh but also a solid or a solid shell element mesh. The solid model mesh is an 'extrusion' of the shell element input mesh. If this input shell mesh uses linear elements (SHELL181) the solid model mesh generated in ACP can have either layered solid elements (SOLID185) or layered solid shell elements (SOLSH190). If it is quadratic (SHELL281) the solid model mesh can only have quadratic layered solid elements (SOLID186).

The geometry and loading of the engineering problem ultimately dictate what element type is best suited for the analysis. Here are a few general considerations about the element types in ACP. Please refer to the [Element Library in the ANSYS Theory Reference](#).

3.9.2. Shell Elements

The shell elements are suited for modeling thin-walled to moderately thick-walled structures. Shell elements are compliant in bending and give good deformation results while being computationally inexpensive.

3.9.3. Solid Elements

The solid elements are aimed at modeling thick walled structures. As laminate thicknesses increase, out-of-plane stresses become more significant and solid elements are better at approximating these thickness effects. Furthermore, the layered solid elements allow the incorporation of composite parts in larger solid model assemblies.

A shortcoming of these element types is that they are typically too stiff in bending when elements are thin. Displacements can be wrong by an order of magnitude as the element undergo a phenomenon called locking. Element technologies such as Enhanced Strain Formulation try to remedy this numerical locking but are not sufficient to do so in linear 3D solid elements. Quadratic solid elements (SOLID186) offer better solutions, however, this comes at an increased computational cost.

3.9.4. Solid Shell Elements

Solid shell elements cover the spectrum between shell and solid elements and are best suited for modeling thin to moderately thick structures. Thin SOLSH elements do not undergo locking yet at the same time they are able to give good results for out-of-plane stresses and strains.

Chapter 4: Usage Reference

This chapter consists of the following sections:

- 4.1. Features
- 4.2. Postprocessing
- 4.3. Available Interfaces to FE Packages
- 4.4. FAQ

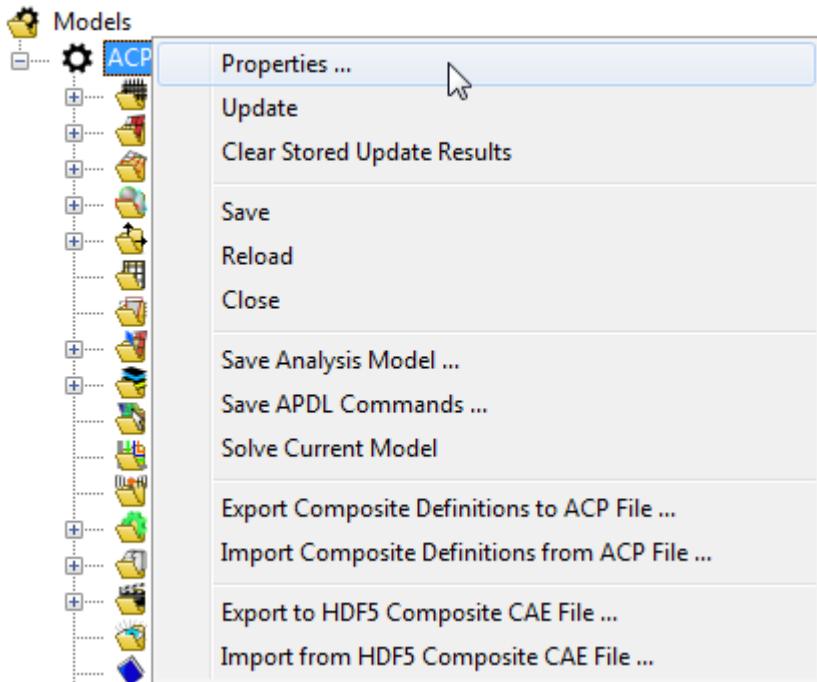
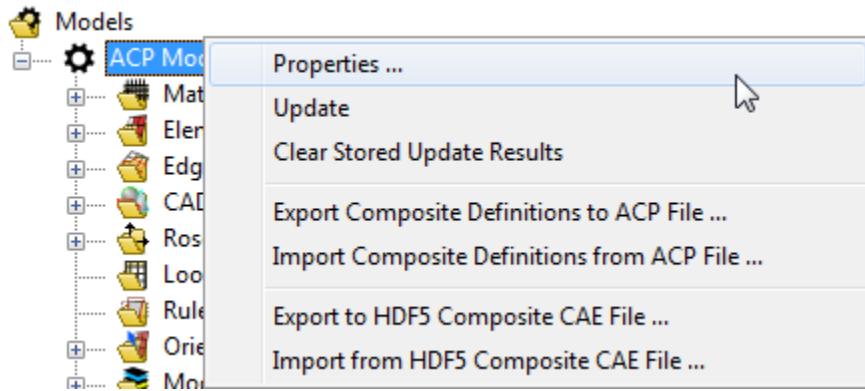
4.1. Features

The features described are in the following sections:

- 4.1.1. Model
- 4.1.2. Material Data
- 4.1.3. Element and Edge Sets
- 4.1.4. CAD Geometries
- 4.1.5. Rosettes
- 4.1.6. Look-up Tables
- 4.1.7. Rules
- 4.1.8. Oriented Element Sets (OES)
- 4.1.9. Modeling Ply Groups
- 4.1.10. Analysis Ply Groups
- 4.1.11. Sampling Elements
- 4.1.12. Section Cuts
- 4.1.13. Sensors
- 4.1.14. Solid Models
- 4.1.15. Layup Plots
- 4.1.16. Definitions
- 4.1.17. Solutions
- 4.1.18. Scenes
- 4.1.19. Views
- 4.1.20. Ply Book
- 4.1.21. Parameters
- 4.1.22. Material Databank

4.1.1. Model

The context menu of the *Model* feature differs between the ACP Workbench mode and the Stand-Alone mode.

Figure 4.1: Model context menu in stand-alone**Figure 4.2: Model drop-down menu in Workbench mode**

This is an overview of the items in the model feature context menu. Selected items are explained in more detail below.

- **Properties:** displays the Model Properties window where information about the model, input file, tolerances and unit system can be found and modified (see [Model Properties](#))
- **Update:** causes an update of the entire model.
- **Clear Stored Update Results:** deletes all results of the previous update.
- **Save** (Stand-Alone only): saves the selected model.
- **Reload** (Stand-Alone only): reloads the input file into the database. This is way to return to the last saved state.
- **Close** (Stand-Alone only): closes the selected model.
- **Save Analysis Model ...** (Stand-Alone only): save the ANSYS input file including the lay-up defined in ACP.

- **Save APDL Commands ...** (Stand-Alone only): Saves the lay-up definition as APDL Command Macro, modifies the model from isotropic material monolithic elements to orthotropic layered composite elements with some adjustments on results save.
- **Solve Current Model** (Stand-Alone only): submits the ANSYS input file including the composite lay-up definition to the ANSYS solver.
- **Export Composite Definitions to ACP File ...**: export the lay-up definitions to a different ACP file (see [Import / Export of ACP Composite Definitions File](#)).
- **Import Composite Definitions from ACP File ...**: import the lay-up definitions from an other ACP file (see [Import / Export of ACP Composite Definitions File](#)).
- **Export to HDF5 Composite CAE File ...**: export the mesh with the composite definitions to a HDF5 file (see [Import from / Export to HDF5 Composite CAE File](#)).
- **Import to HDF5 Composite CAE File ...**: import a mesh with composite definitions from a HDF5 file (see [Import from / Export to HDF5 Composite CAE File](#)).

Model Properties

Figure 4.3: Model Properties in stand-alone

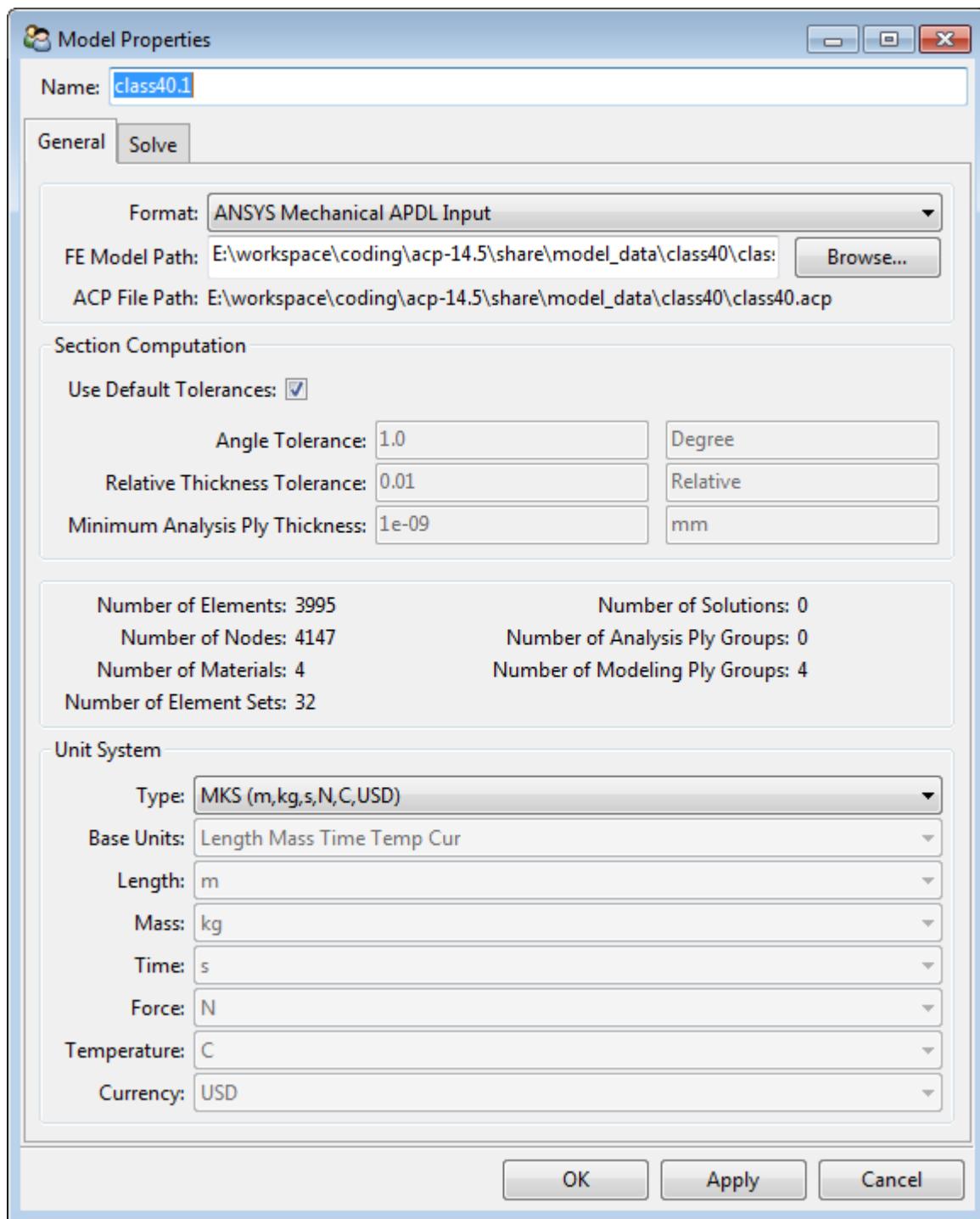
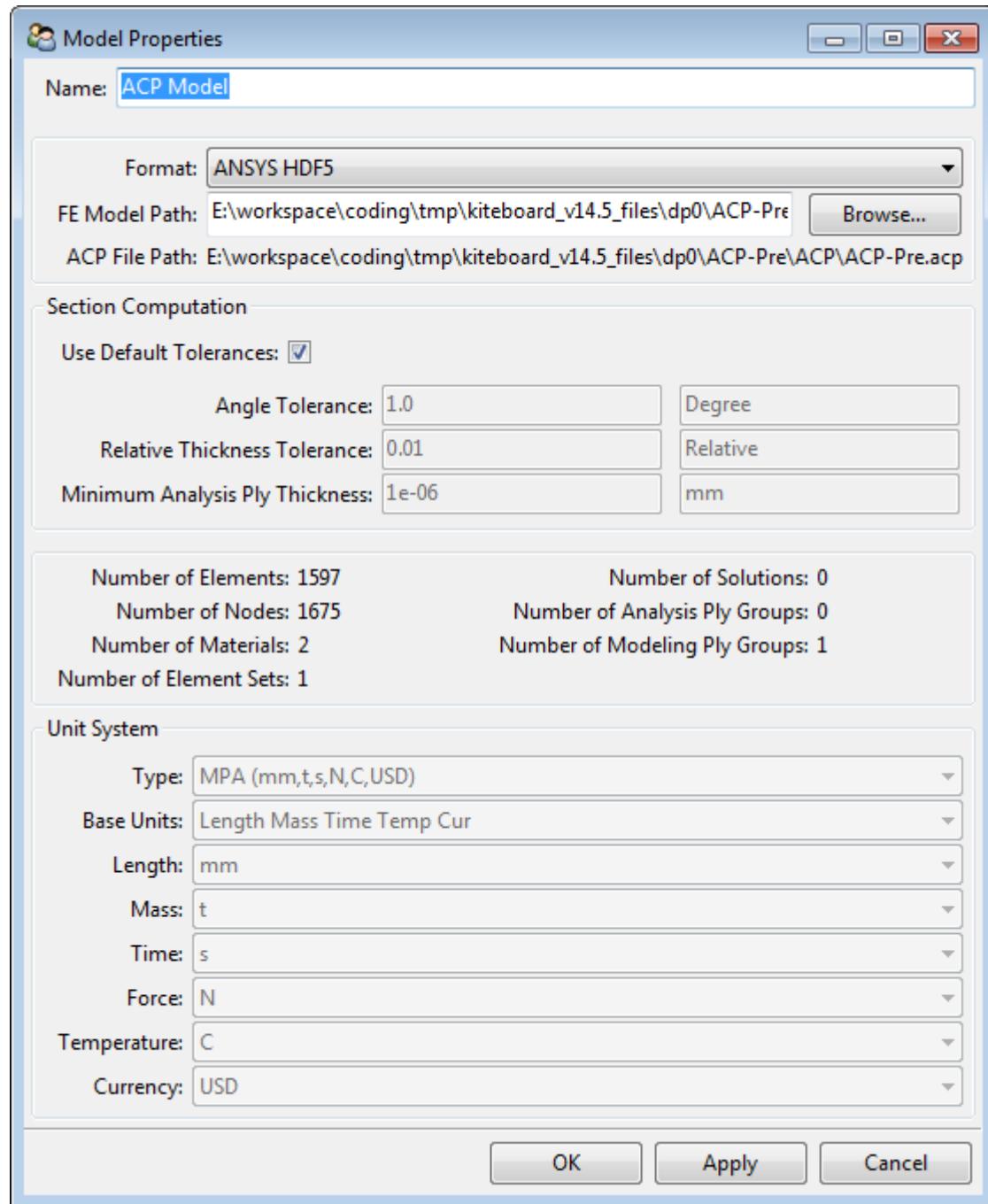


Figure 4.4: Model properties in Workbench integration



File Information

In this first part, the user can change the ANSYS File input file. This is useful in the stand-alone Mode. If the project is managed in Workbench through the Workbench Add-in, the file management must be made in Workbench to avoid any mistake. The input file format for the stand-alone mode can be a *.dat (generated by Workbench), *.inp or *.cdb (generated by Mechanical APDL with CDWRITE). With the Workbench Add-in, the input File is an *.hdf5 file generated by Workbench. More information on the *.hdf5 format can be found in [HDF5](#). The ACP file path is for information only.

Section Computation

Angle and Relative Thickness Tolerance

ACP transfers the composite definitions into Section Data so that they can be interpreted by ANSYS Mechanical. In the case of curved surfaces or draped laminates, sections may change continuously with every element as their orientations change. This generates a large amount of information which can reduce the performance of data transfers and solvers. To avoid this, ACP groups section data of multiple elements together if it lies within a tolerance range.

The Angle Tolerance sets the allowable ply Angle Tolerance between the same layers of neighboring elements. The Relative Thickness Tolerance applies to the individual layer thickness as well as the global layup thickness. For two elements to be included in the same section definition, the difference of the angle and the relative thickness of every single ply must be within the defined tolerance. Core materials are typically a factor ten thicker than a laminate. As such the thickness tolerance is defined as a relative rather than an absolute value. The default tolerance values are very small compared with the manufacturing tolerances of composites. The loss of accuracy is negligible.

Minimum Analysis Ply Thickness

The cutting operations used in ACP can cut an analysis ply to a thickness thinner than the specified ply thickness. If a cutoff geometry intersects a ply at its vertical mid-point it will slice the ply in half for example. When the intersection occurs at ply boundaries extremely thin layer can appear as the result of geometric tolerances of CAD files. These extremely thin layer are of the order of magnitude of 10^{-9} and are purely a results of the numerical imprecision. The Minimum Analysis Ply Thickness sets a thickness threshold below which no such plies can arise.

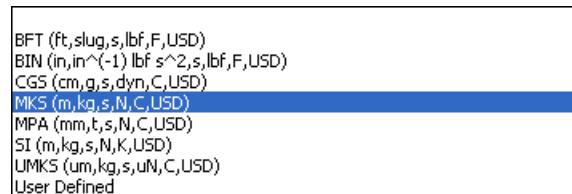
The global default values for the above mentioned tolerances and thickness are set in the section [ACP Submenus](#) but can be overridden in the Model Properties.

Model Summary

In this part global information about the model are given. The number of elements includes the layered shell and layered solid (*Solid Models*) elements.

Units

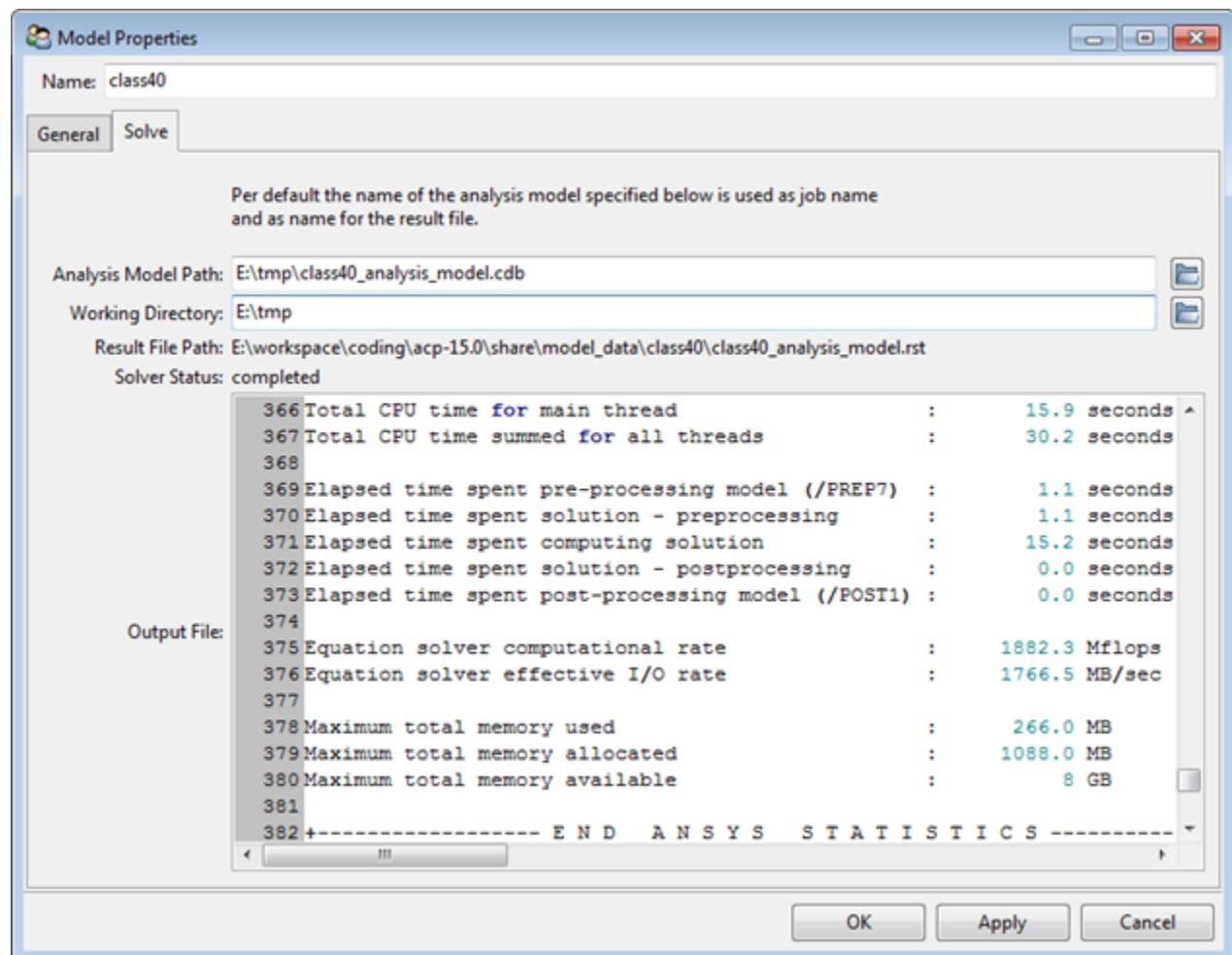
In the stand-alone mode the user can define the *Unit System* of the model if it is not defined in the input file (*.dat file stores the unit system). The user can choose between standard units system or define his own system. If the model is imported directly from Workbench through the Workbench Add-in, the units used in Workbench are automatically transferred and the unit system is frozen. Note: If the unit system changes ACP converts automatically the composite definitions (material data, thicknesses, ...). The mesh of the input file (nodal coordinates) is not converted!

Figure 4.5: Units system**Important**

To import or export model in hdf5 Format, to exchange material data with the database or with ESAComp, the units of the model must be defined.

Solve

In stand-alone mode the *Model* property dialog has a second tab called *Solve*. The user can define the file path of the analysis model and the working directory used by the ANSYS solver. The *Solver Status* and the *Output File* are also given here. In the case of an incomplete run warning and error messages of the ANSYS solver can be found here.

Figure 4.6: Solver information (solve.out)

Import/Export of ACP Composite Definitions File

The ACP Composite Definitions file contains all the information stored in an ACP model. The model is defined in the ACP file in the ACP Python scripting language. For the import, it has to be specified how double entities with the same name are handled.

Figure 4.7: Export Composite Definitions window

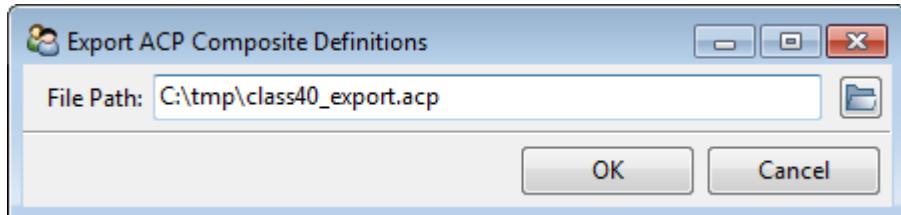
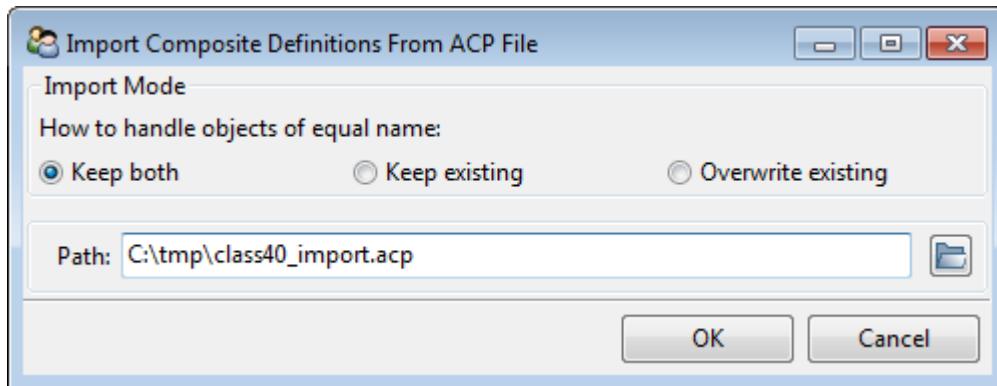


Figure 4.8: Import Composite Definitions window



Import from / Export to HDF5 Composite CAE File

HDF5 is a neutral format, which allows the exchange of composite data between different software (CAE or CAD). For more information, see [HDF5](#).

4.1.2. Material Data

ANSYS Composite PrepPost differentiates between four material classes: Materials, Fabrics, Stackups and Sub Laminates.

- The Materials class is the material database in ACP.
- The Fabric class is where the Materials can be associated with a ply of a set thickness. Draping Coefficient can be added as well as unit price properties.
- The Stackup class is used to combine fabrics into a non-crimp fabric, such as a [0 45 90] combinations.
- The Sublaminates class is used to group fabrics and stackups together for frequently used lay-ups.

A stackup or a sublamine can only be defined if a fabric and material have been defined previously.

Materials

The Materials database is only editable within ACP in the Stand-Alone mode. Otherwise it draws all material properties from the Engineering Data component within Workbench. In this case, the material properties can only be viewed but not altered in ACP.

Materials Context Menu

The context menu of the Materials class has the following options:

- Create Material ... :opens a Material Properties window for creating a new material (only available in Stand-Alone Mode).
- Paste ... :pastes a copied material into the material database (only available in Stand-Alone Mode).
- Sort: sorts the list of materials alphabetically.
- Export ... : export the material database into a CSV file, ESAComp XML file or an ANSYS Workbench XML file.
- Import ...: imports materials from CSV file or ESAComp XML file into the material database (only available in Stand-Alone Mode).

More information on the import and export of ESAComp XML file can be found in the section [ESAComp](#). Stackups and Sublaminates can also be exported to an ESAComo XML file format and are converted to laminates in the process.

Figure 4.9: Materials class context menu in Stand-Alone mode

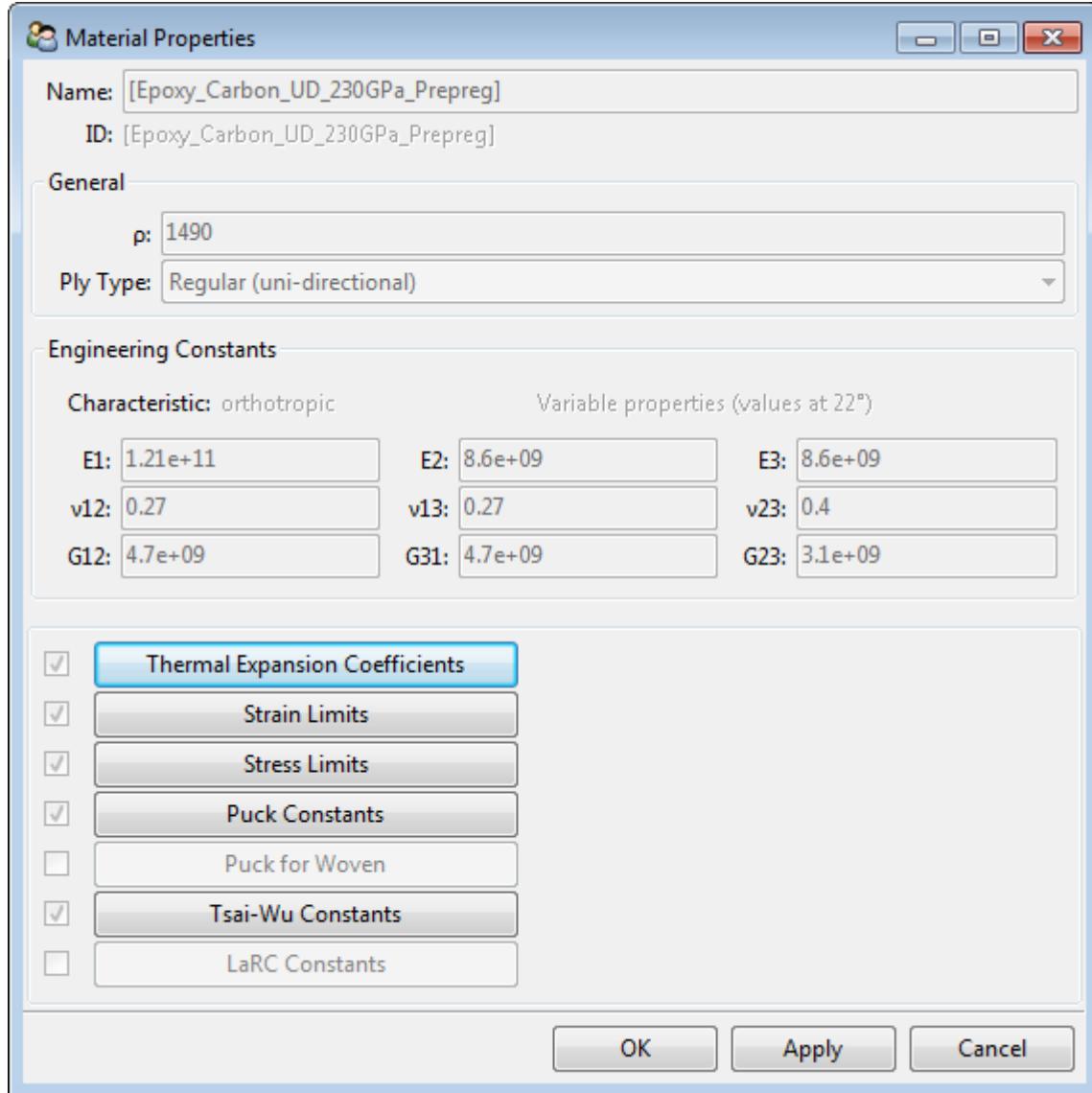


Temperature Dependent Material Properties

Temperature dependent material properties can be used in the failure analysis of a composite if the solution contains a temperature datablock. Temperature dependent properties can be set for Engineering Constants (Young's Modulus, Poisson's Ratio, etc.) as well as Stress and Strain Limits. Other properties used in the failure analysis (Puck's constants, etc.) can not be made temperature dependent and remain constant. The material properties are always shown at the reference temperature in the Materials class. Variable material data can be been displayed in the Engineering Data component in WB as well as in the Python UI. The temperature dependent material properties only come into use in the post-processing mode. For any analysis in the pre-processing, the material properties at the reference temperature are used.

Within the WB workflow variable data is passed to and from ACP. In the Stand-Alone mode, variable material data is passed to and from ACP if the import and export is in an ANSYS file format such as a .cdb file. While it is possible to enter variable material data into ACP Stand-Alone through the Python UI it is cumbersome and it is advised to read in temperature dependent material properties through an ANSYS input file. The export and import of material data within the Material class via the CSV and XML file formats does not support the variable temperature data.

General Properties



In this main window, the standard pre-processing data is required:

- **Name of the material**
- **Density ρ**
- **Orthotropic Young's Modulus:**
 - E1: in-plane, in fiber direction (fiber direction is corresponding to angle 0 for the ply's definition)
 - E2: in-plane, orthogonal to fiber direction
 - E3: out of plane direction
- **Orthotropic Poisson's Ratio:**
 - ν_{12} : in-plane
 - ν_{13} : out of plane, in fiber direction

- ν_{23} : out of plane, normal to fiber direction

- **Orthotropic Shear Modulus:**

- G12: in-plane
- G13: out of plane, in fiber direction
- G23: out of plane, normal to fiber direction

- **Ply type: Defines the type of material which affects the post-processing**

- *regular*: Uni-directional reinforced material
- *woven*: Weave material
- *homogeneous_core*: Sandwich core material e.g. balsa or foam
- *honeycomb_core*: Sandwich core material with a honeycomb pattern
- *isotropic_material*: Isotropic material and post-processed with the Von Mises criterion

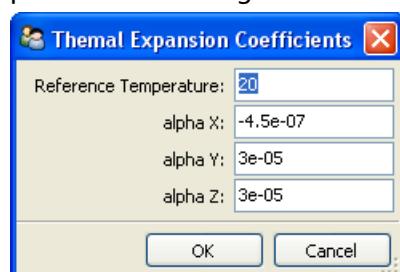


The Engineering Data can only be modified in the Pre Mode. In Post Mode, they are frozen.

In addition further failure properties can be activated. Depending on the ply-type some properties are deactivated automatically.

Thermal Expansion Coefficients

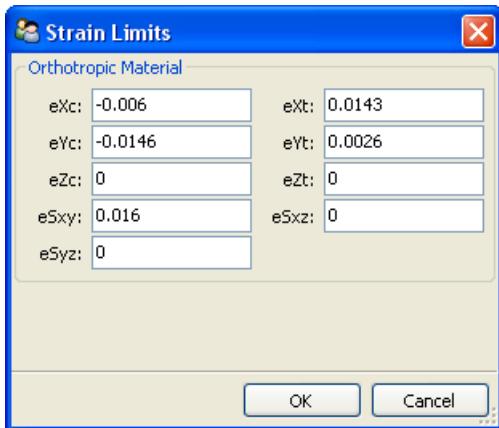
For thermal stress analyses, the thermal expansion coefficients of the material and the reference temperature must be given:



- Reference Temperature: temperature at which strain in the design does not result from thermal expansion or contraction
- alpha X: in-plane, in fiber direction (fiber direction is corresponding to angle 0 for the ply's definition)
- alpha Y: in-plane, orthogonal to fiber direction
- alpha Z: out of plane direction

Strain Limits

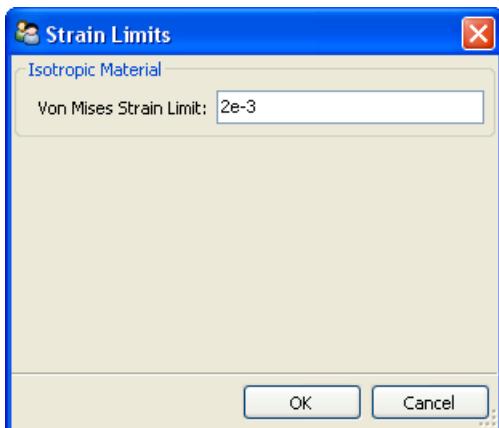
The given limits are used to calculate the IRF if the Criteria *Max Strain* is selected in the Failure Criteria Definition. Compressive strain limits have to be negative.



For orthotropic materials, the 9 strain limits (5 in-plane and 4 out-of-plane strains) can be completed:

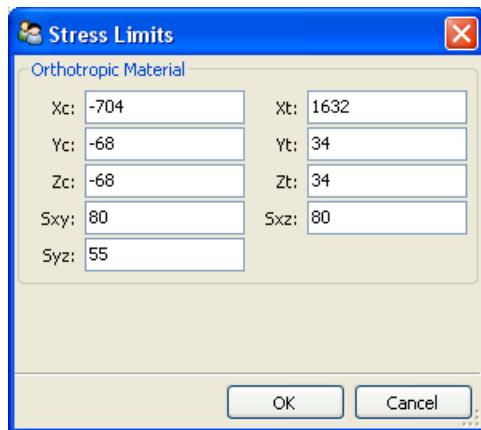
- eXc: normal strain, in-plane, in fiber direction, compression limit
- eXt: normal strain, in-plane, in fiber direction, tension limit
- eYc: normal strain, in-plane, orthogonal to fiber direction, compression limit
- eYt: normal strain, in-plane, orthogonal to fiber direction, tension limit
- eZc: normal strain, out of plane, compression limit
- eZt: normal strain, out of plane, tension limit
- eSxy: in-plane shear strain
- eSxz: transverse (interlaminar) shear strain, plane in fiber direction
- eSyz: transverse (interlaminar) shear strain, plane normal to fiber direction

If the material is defined as isotropic, the Von Mises Strain Limit is active:



Stress Limits

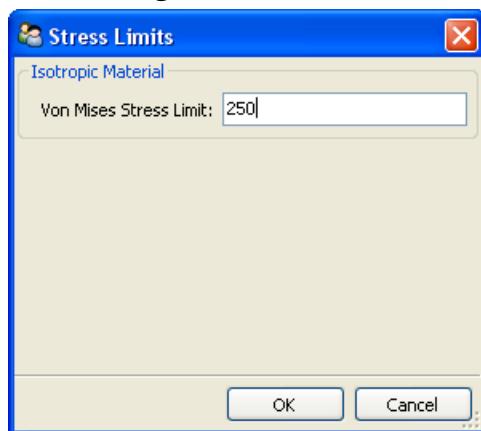
All the other *Failure Criteria Definitions* are based on *Stress Limits* values. Compressive stress limits have to be negative.



For orthotropic materials, the 9 stress limits (5 in-plane and 4 out-of-plane strains) can be completed:

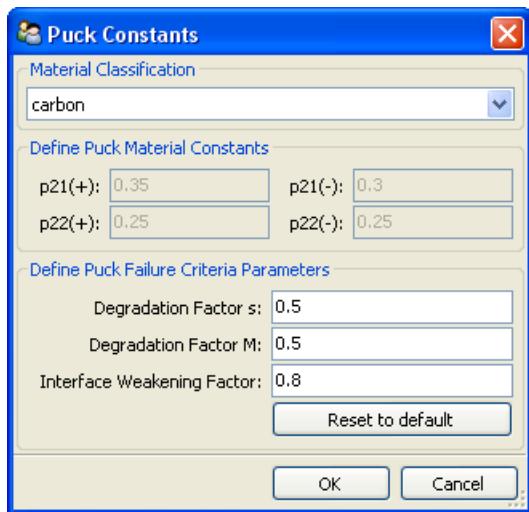
- Xc: normal stress, in-plane, in fiber direction, compression limit
- Xt: normal stress, in-plane, in fiber direction, tension limit
- Yc: normal stress, in-plane, orthogonal to fiber direction, compression limit
- Yt: normal stress, in-plane, orthogonal to fiber direction, tension limit
- Zc: normal stress, out of plane, compression limit
- Zt: normal stress, out of plane, tension limit
- Sxy: in-plane stress strain
- Sxz: transverse (interlaminar) shear stress, plane in fiber direction
- Syz: transverse (interlaminar) shear stress, plane normal to fiber direction

If the material is defined as isotropic the Von Mises Stress Limit is active (equivalent to Tensile Yield Strength in the Workbench ED):



Puck Constants

The Puck Failure Criterion requires internal parameters, which depend on the material. Two default sets of parameters are already defined for carbon and glass fibers. If the parameters are different, define them as *material-specific* constants. If the Puck criterion does not need to be checked for this material, select *ignore Puck criterion*.

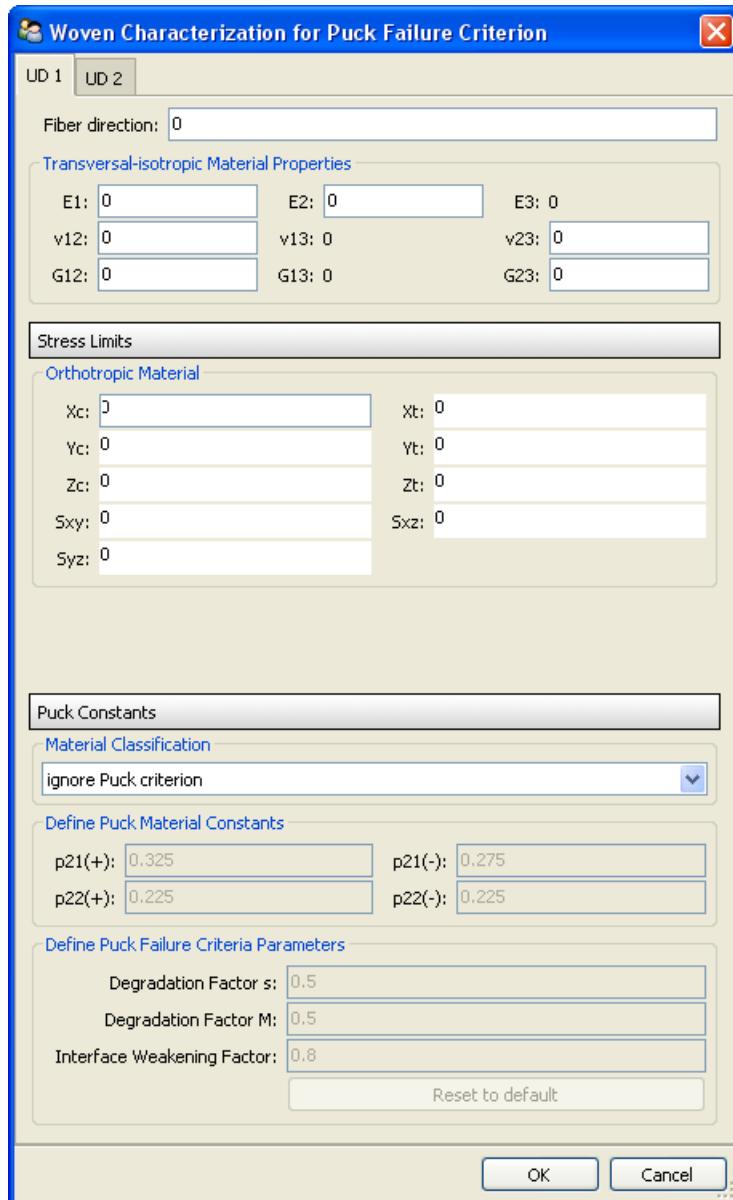


The meaning of the parameters is:

- p21(+) Tensile inclination XZ
- p21(-): Compressive inclination XZ
- p22(+) Tensile inclination YZ
- p22(-): Compressive inclination YZ
- s and M: Degradation parameters
- Interface weakening factor: Scales the interlaminar normal strength

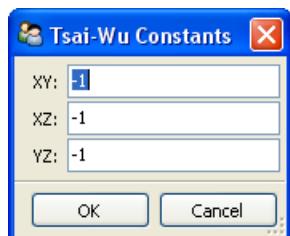
Puck for Woven

The *Puck for Woven* functionality of ACP allows to evaluate the Puck failure criterion for woven materials. Two UD plies can be specified representing the woven ply. During the failure evaluation ACP evaluates the stresses for these plies and computes the Puck failures. The relative ply angles, engineering constants, stress limits and Puck constants have to be defined for both plies. Note: This specification does NOT effect the analysis model and is only considered in the failure analysis for the Puck criterion.



Tsai-Wu Constants

The Tsai-Wu Constants are constants used into the interaction coefficient of the quadratic failure criteria for Tsai-Wu formulation. Refer to [Tsai-Wu Failure Criterion](#) for more details on this formulation.



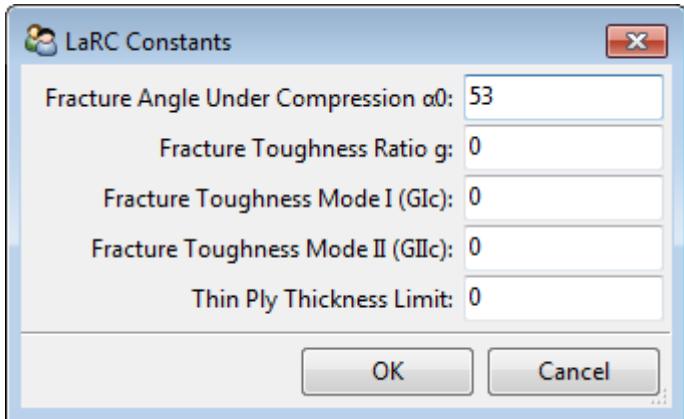
Note

In ACP the Tsai-Wu constants are:

- $2 F_{12} = XY$, default -1
- $2 F_{13} = XZ$, default -1
- $2 F_{23} = YZ$, default -1

LaRC Constants

The LaRC Failure Criteria needs also user's defined parameter to evaluate the failure in matrix and fiber:

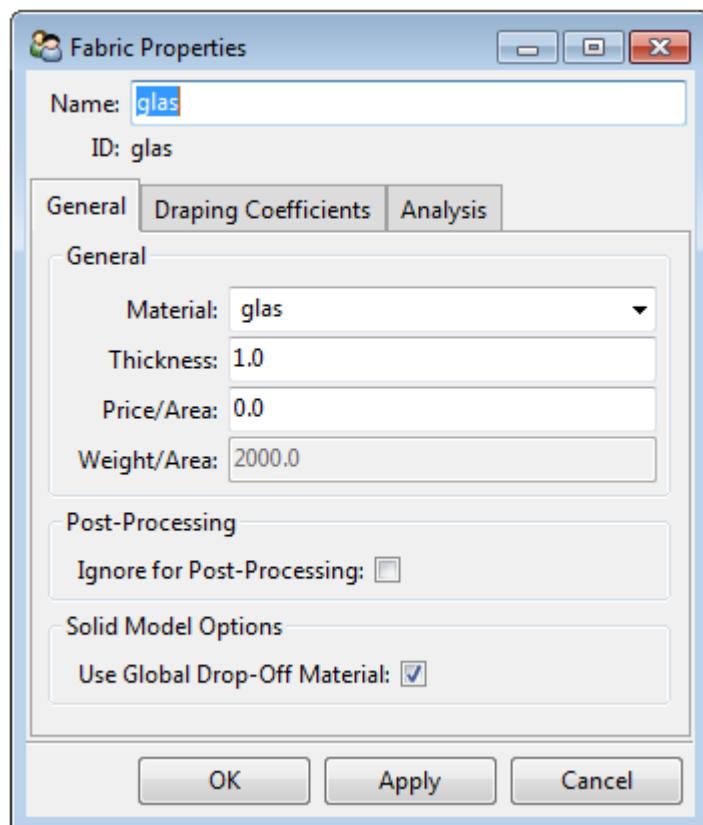


- Fracture Angle under Compression is the value for α_0 used in the LaRC fiber and matrix failure. By default and conformed to literature, the default value is defined to 53° .
- Fracture Toughness Ratio is the ratio of the mode I to mode II fracture toughness, which is used in the fiber failure criteria.
- Fracture Toughness Mode I
- Fracture Toughness Mode II
- Thin Ply Thickness Limit (set to 0.7 mm in WB mode)

More information is given in the section [LaRC03/LaRC04 Constants \(p. 243\)](#).

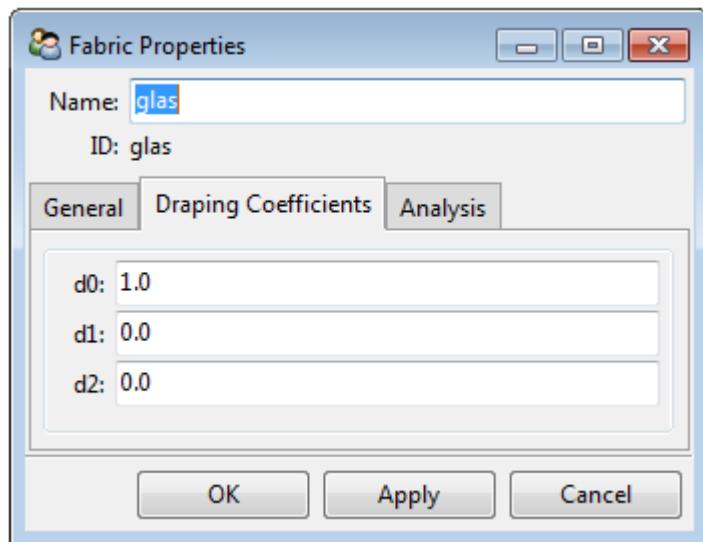
Fabric

The *Fabric* property dialog is shown in the figure below.



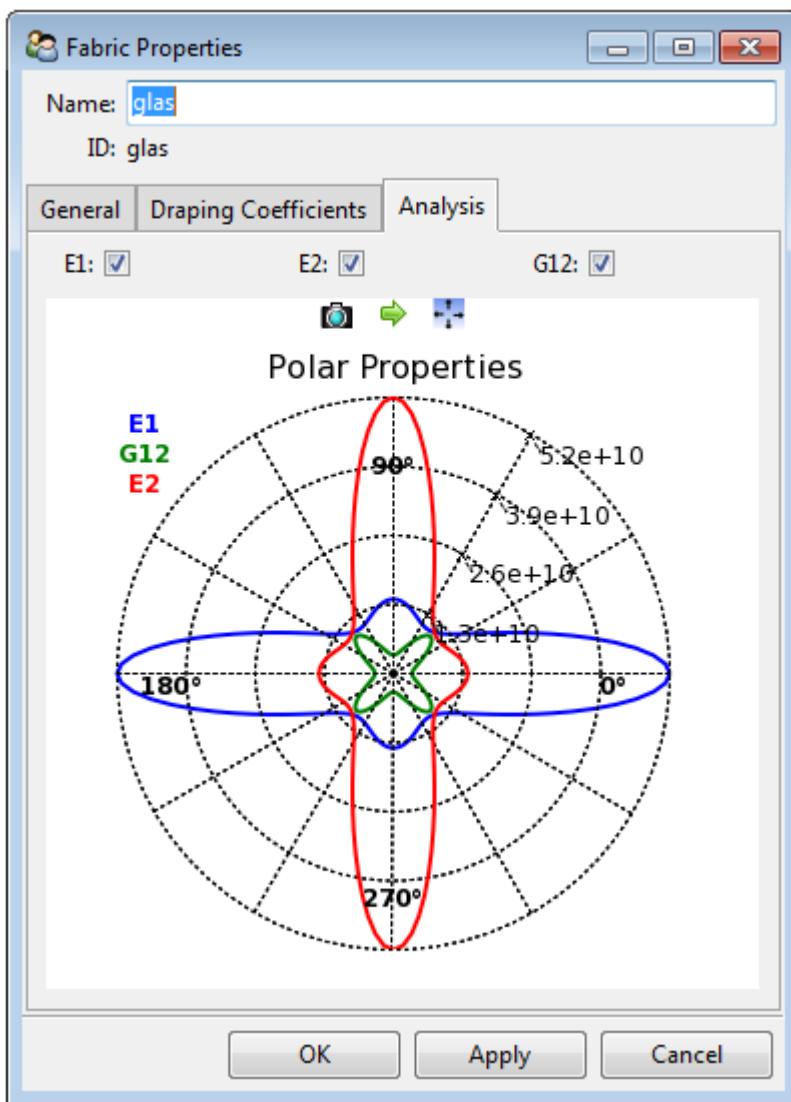
- Material: material of the fabric
- Thickness: ply thickness
- Price /Area: as additional information, the surface price can be given to provide global information thanks to [Sensors \(p. 167\)](#):
- Ignore for Post-Post-processing: if activated all the analysis plies with this fabric will not be considered in the failure criteria analysis in ACP (Post). This does not effect the analysis model.
- Use Global Drop-Off Material: if active the solid model export uses a globally defined homogeneous material such as a resin for drop-off elements. If deactivated the Fabric material is used for drop-off elements which is of interest for sandwich structures. See [Drop-Off Element Handling](#) for details.

The draping coefficients are part of the fabric definition and are used if draping is activated in the *Oriented Element Set* or *Modeling Ply* definition.



The three coefficients d0, d1 and d2 used in the *draping* feature are defined here. For more details about the draping calculation, see the [Draping Simulation](#) in the [Theory Documentation](#) (p. 219).

The *Polar Properties* ([Classical Laminate Theory](#)) of the *Fabric* can be plotted as graphical information.

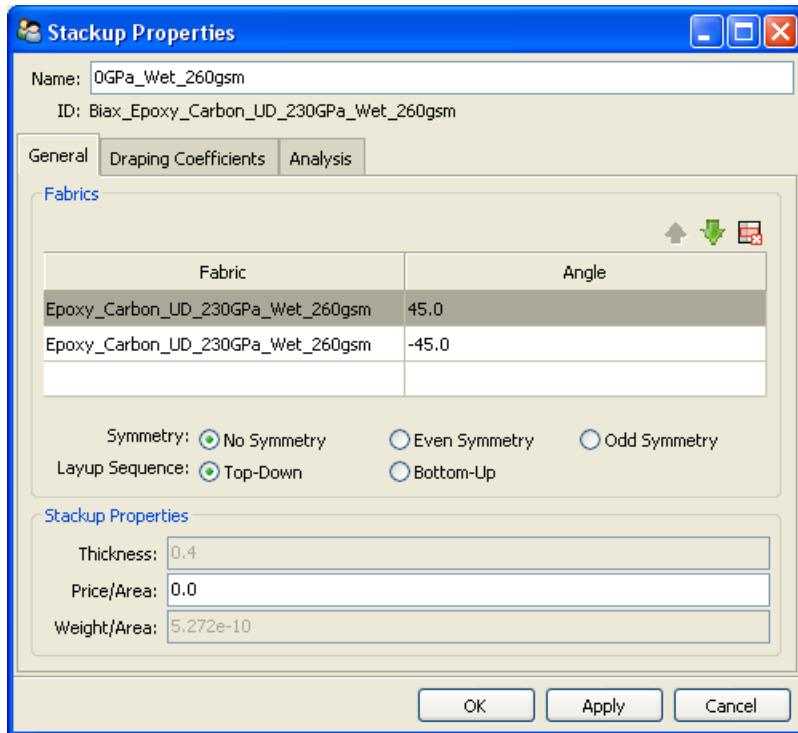


This plot can be exported as a picture (📷) or in a csv file (➡)

Stackups

General

A stackup is a non-crimp fabric with a defined stacking sequence. From a production point of view, it is considered as one ply, which is applied on the form. For the analysis, all plies forming the stackup are considered. For every ply of the stackup, the Fabric and its orientation must be given:



These stackups have different Price/Area and draping properties than a laminate of single plies. That's why Price/Area and draping coefficients can be entered again. Stackups can be exported to an ESACOMP XML file format and are converted to laminates in the process.

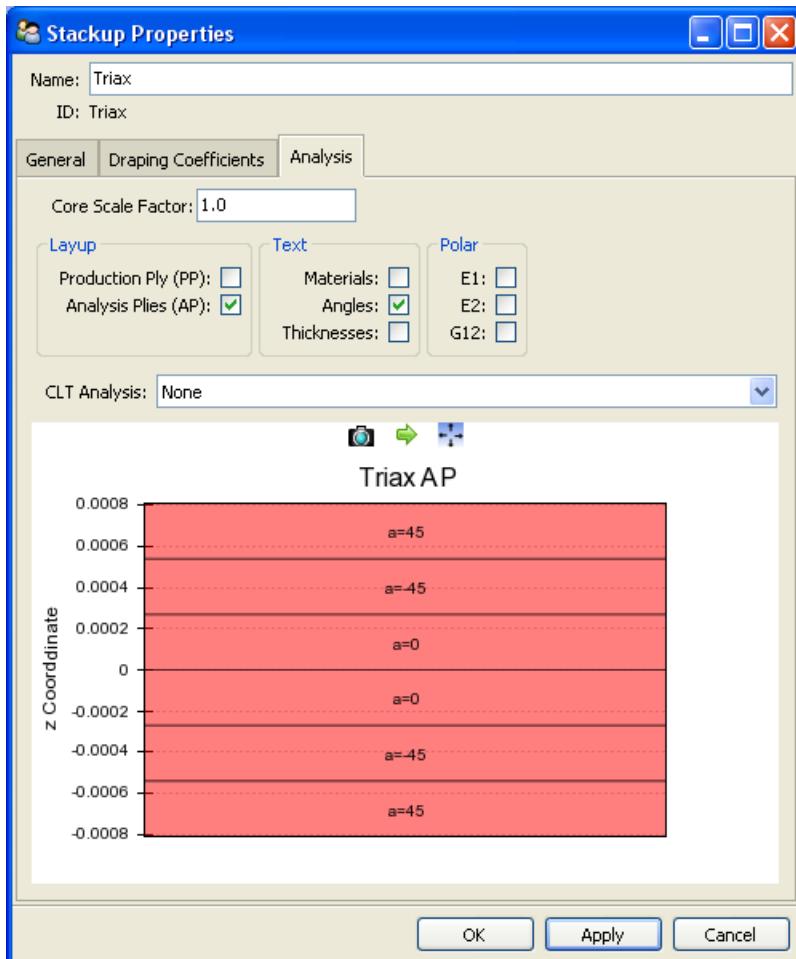
Top-Down or Bottom-Up Sequence

The definition of the Stackup can be given in both directions (Bottom-Up and Top-Down). In the Top-Down sequence, the first defined ply (the first one in the list, in the picture here the 45 direction) is placed first on the mold ad so on the bottom of the stackup and the other plies are placed over it. In Analysis, the sequence can be checked. In this example, the ply -45 direction is on top (see [Layup information and polar properties](#)).

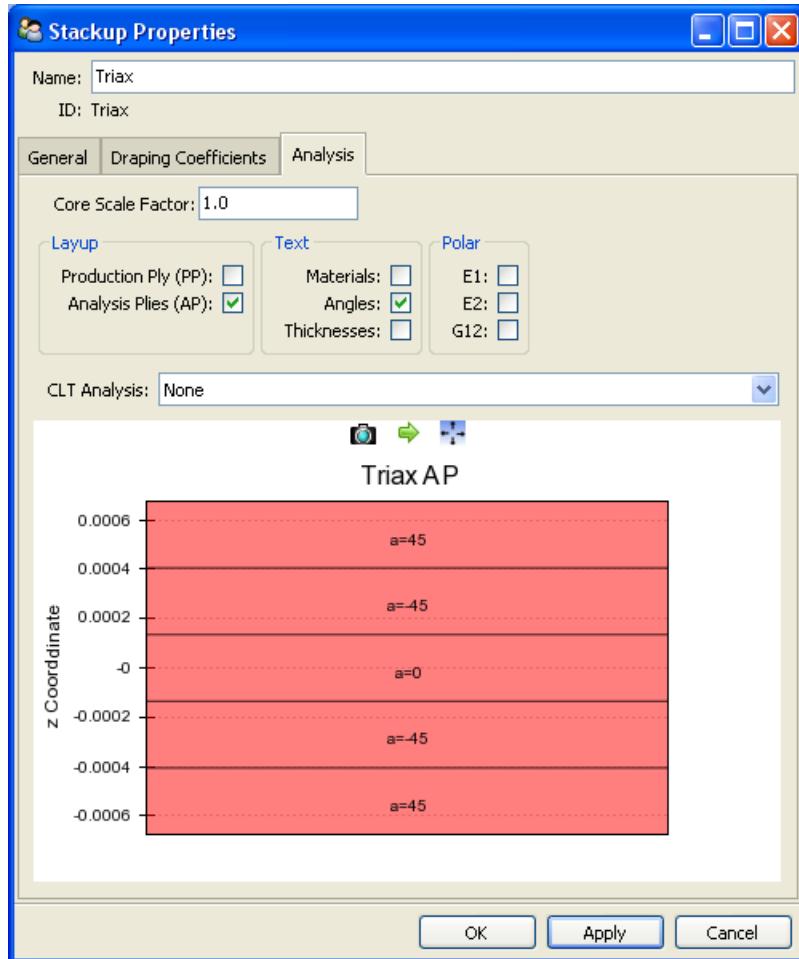
Symmetries

For a quicker definition, the stackup can be defined with symmetries.

The *Even Symmetry* defines the symmetry axis on the top of the sequence, and uses all plies.

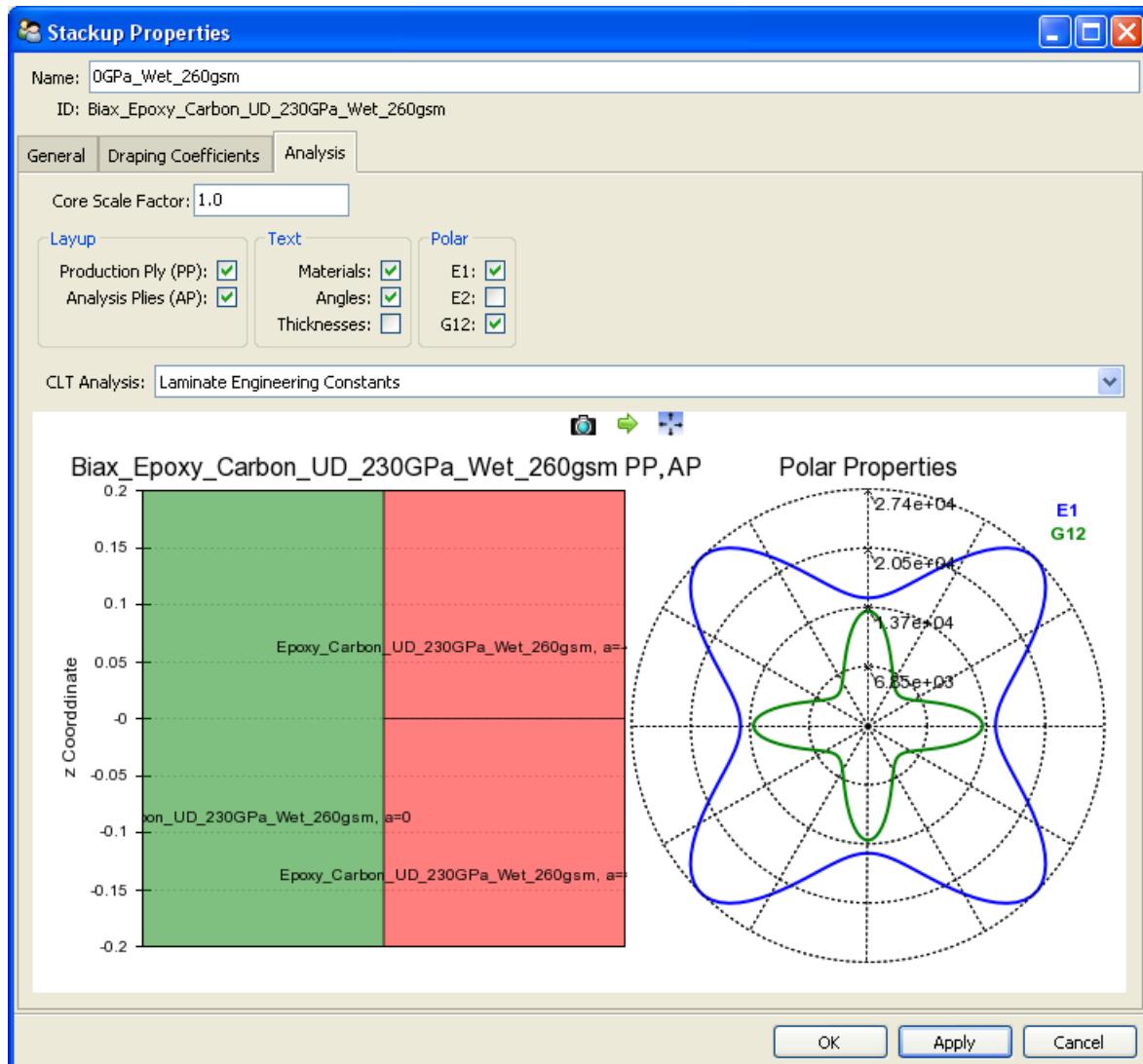
Figure 4.10: Stackup sequence with even symmetry

In the *Odd Symmetry* the ply on the top is not used for the symmetry. So the middle of the top ply is the symmetry axis of the final sequence.

Figure 4.11: Stackup sequence with odd symmetry

Analysis

The *Analysis* tab provides the illustration and evaluation of the laminate properties of the stackup, which can be plotted as graphical information.

Figure 4.12: Layup information and polar properties

This plot can be exported as picture () or in a csv file (). It is possible to translate and zoom into the lay-up distribution with the mouse button. To come back to a fit view, click on . In addition, laminate properties e.g. stiffness matrix or flexural stiffness, which are based on the classical laminate theory (Section [Classical Laminate Theory](#)), can be calculated by ANSYS Composite PrepPost.

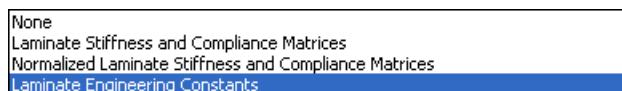
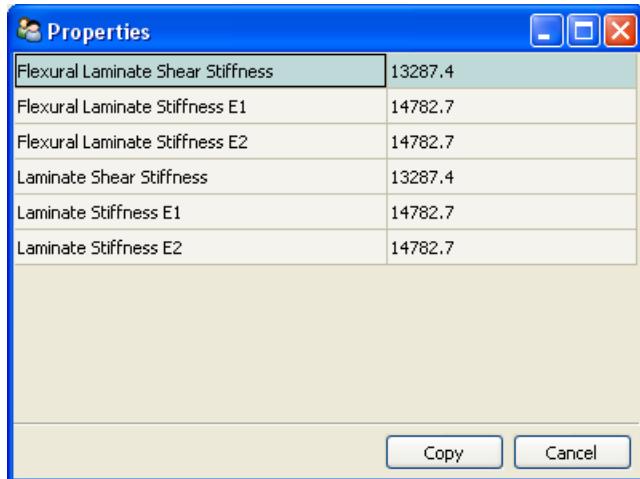
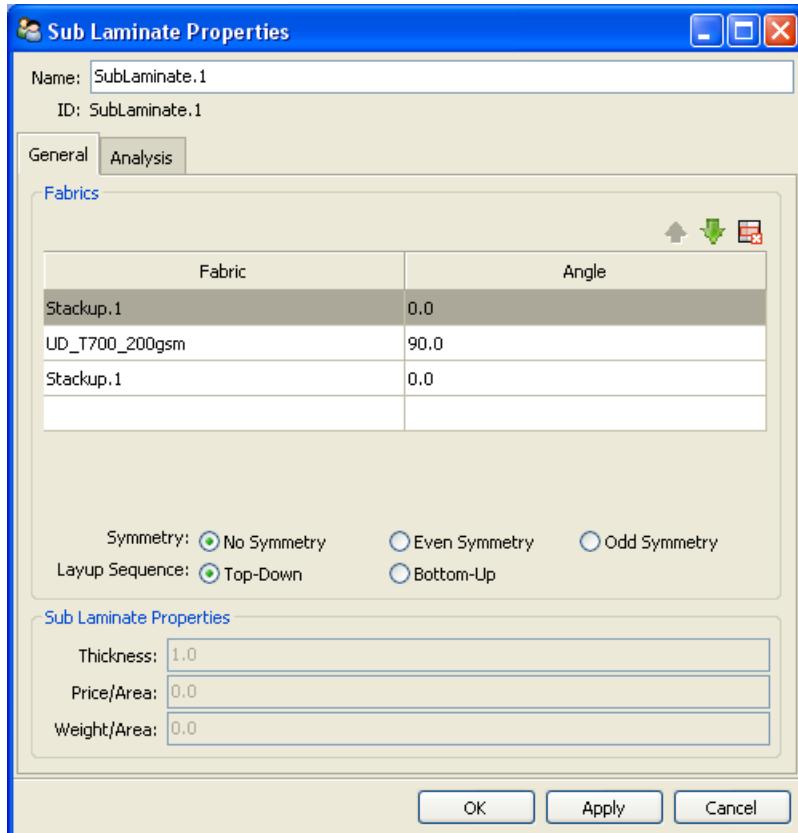
Figure 4.13: CLT Analysis results

Figure 4.14: Properties based on the classical laminate theory

Sub Laminates

A sub laminate is a sequence of plies defined by fabrics and stackups with relative angles. This sub laminate can be used later in the lay-up definition. As with the *Stackups*, the sequence direction and symmetry can be chosen. Refer to [Stackups](#) for the description. Sub laminates can be exported to an ESAComp XML file format and are converted to laminates in the process.

Figure 4.15: Properties based on the classical laminate theory

The *Analysis* tab is exactly the same for a stackup. There is one level (modeling ply) more in the lay-up description.

Important

If stackups and symmetry are used in the sub-laminate definition, the stackup will not be reversed into the ply sequence. As example, define a stackup S1 defined as [45,-45,0], and a sub-laminate defined with even symmetry and the stackup ([S1]s), the sub-laminate sequence will be [45,-45,0,45,-45,0], not [45,-45,0,0,-45,45].

4.1.3. Element and Edge Sets

The *Named Selections* defined in the ANSYS Mechanical application or the components in ANSYS Mechanical APDL are imported into ANSYS Composite PrepPost. These selections are imported with the same names. The Faces (Element Components in ANSYS Mechanical APDL) are imported as [Element Sets](#) and the edges (Node Components in ANSYS Mechanical APDL) as [Edge Sets](#).

Important

After the definition of a new *Named Selection* in ANSYS Mechanical, the *Model* status must be updated.

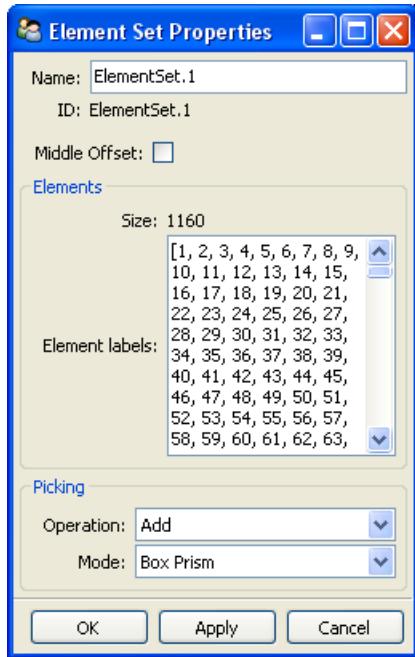
Element Sets

Middle Offset option

If the mesh is generated at the mid-plane surface, the section definition has to be translated so that the middle of the section corresponds to the element.

Manual definition

New *Element Sets* can be manually defined by selecting the elements. The Element Sets store the element labels and are therefore not associative with the geometry. If the mesh changes the element numbering changes and the Element Set has to be redefined to avoid erroneous layups.

Figure 4.16: Element Set Selection

- Operation: select if you want to *Add* or *Remove* elements in the list.
- Mode: define the selection mode by dragging the mouse from one corner to the other one
 - Box on surface: only visible elements are selected
 - Box Prism: all elements included in the box, also in depth
 - Point: element at the picking location

Context Menu

Figure 4.17: Element Set Context Menu

Several options are available in the Element Set context menu:

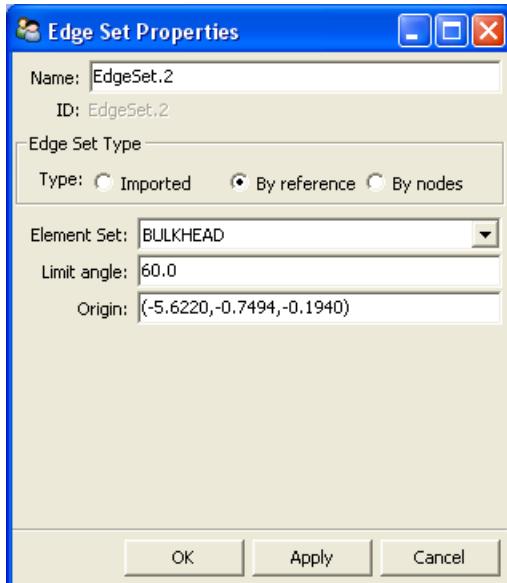
- Properties: open the element set property window.
- Update: update any changes into ACP database
- Hide / Show: hide or show this element set. The elements of the hidden Element Sets are no longer visible in the Scene.

- Copy: copy the selected Element Set
- Paste: paste a copied Element Set
- Delete: delete the selected Element Set
- Export Boundaries...: the boundaries of an element set can be exported to a STEP or an IGES file
- Partition: creates partitioned Element Sets for any Element Set that can be divided in different zones, due to a geometrical separation (three elements share the same edge for example).

Edge Sets

Similar to **Element Sets**, the edge components defined in ANSYS Mechanical using *Named Selections* are imported. New *Edge Sets* can be manually defined by selecting the nodes:

Figure 4.18: Edge Set Definition



- Imported: Edge set imported with the model.
- By reference: an existing *Element Set* is used to define a new *Edge Set* through one boundary limited by an angle diffusion:
 - Element Set: the Edge Set is part of the boundary of the selected Element Set.
 - Limit angle: the Edge Set is extended from the origin in both directions until the angle between two elements is bigger than the *Limit angle*. With a negative value, the *Limit angle* is deactivated; it means the edge set will be the whole boundary of the selected Element Set.
 - Origin: origin to determine the closest boundary
- By nodes: Select manually the nodes

4.1.4. CAD Geometries

External geometry (iges and stp format) can be imported to use them in combination with these features:

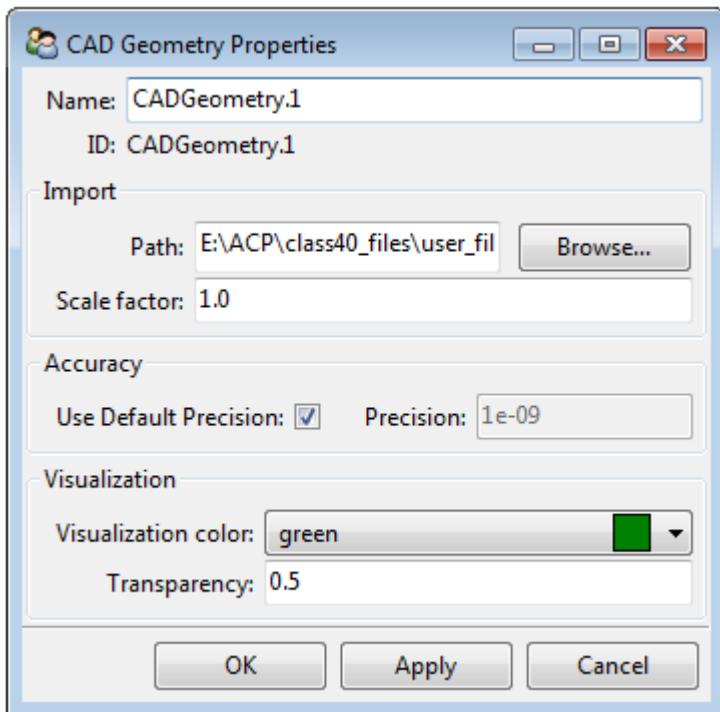
- *Cut off Rules*, see [Cutoff Rules](#).
- Core ply definition (Core geometry), see [Modeling Ply Groups](#).
- *Extrusion Guides* and *Snap to Geometry* in the *Solid Models* definition, see [Solid Models](#).

There are two ways to import a CAD file into ACP - either directly in ACP or through a geometry link in Workbench. The advantage of the Workbench geometry link is that the CAD geometry link remains intact when a project is archived and restored elsewhere.

Direct Import

The geometry import has these options:

Figure 4.19: Import external CAD Geometry



- Name: name of geometry for future use in ACP Database
- Path: location of the geometry file
- Scale factor: scale the geometry in the global coordinate system (useful for change of units).
- Precision: precision of the imported geometry. This value is used to evaluate intersections and other geometrical operations.
- Visualization color: color, in which the imported geometry is plotted
- Transparency: adjust the transparency of the imported geometry between zero and one.

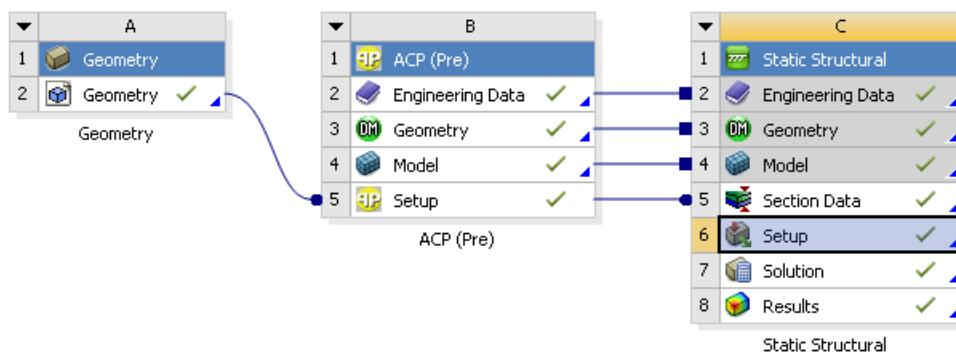
Assemblies should be imported as a single step file (*.stp) rather than individual ones containing links. This has proven to be more robust.

Workbench geometry link

The geometry component in ANSYS Workbench can be used to import (link) CAD geometries with ACP Pre. The linked geometry is scaled automatically if its units do not match those of the project. The required steps for geometry import are:

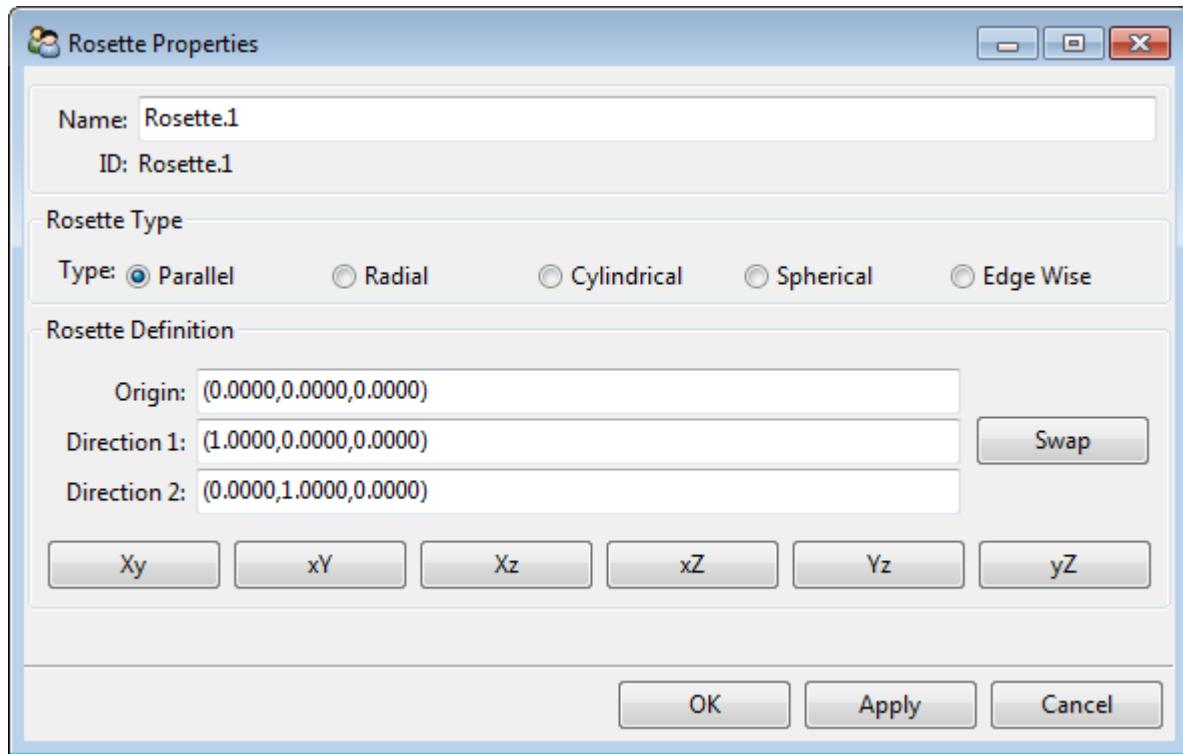
- Add a geometry component to the project
- In the component's context menu select Import Geometry
- Link the geometry cell to the appropriate ACP Setup cell
- The imported geometry will now appear under CAD Geometries in ACP

Figure 4.20: Project Schematic with a CAD Geometry Import in ACP



4.1.5. Rosettes

Rosettes are coordinate systems that are used to set the reference direction of Oriented Element Sets. In other words, Rosettes define the 0° direction for the composite layup. The coordinate systems defined in the Mechanical application are imported by default. Additional Rosettes can be defined where five different types are implemented in ACP. The origin and directions of the rosettes are given by global coordinates. Hence the Rosettes are independent of the mesh even if the user selects nodes and elements to define this properties.

Figure 4.21: Property dialog

Rosette Types

- Parallel: this Rosette type is analogous to a cartesian coordinate system. The reference direction is given by the Rosette's X direction.
- Radial and Cylindrical: both Rosettes are based on a cylindrical coordinate system. For a Radial Rosette, the reference direction is either radially inward or outward and is given by the Rosette's X direction. For a Cylindrical Rosette, the reference direction runs circumferentially around the Rosette's Z direction.

Figure 4.22: Oriented Element Set with a Radial Rosette. The yellow arrows indicate the reference direction of each element.

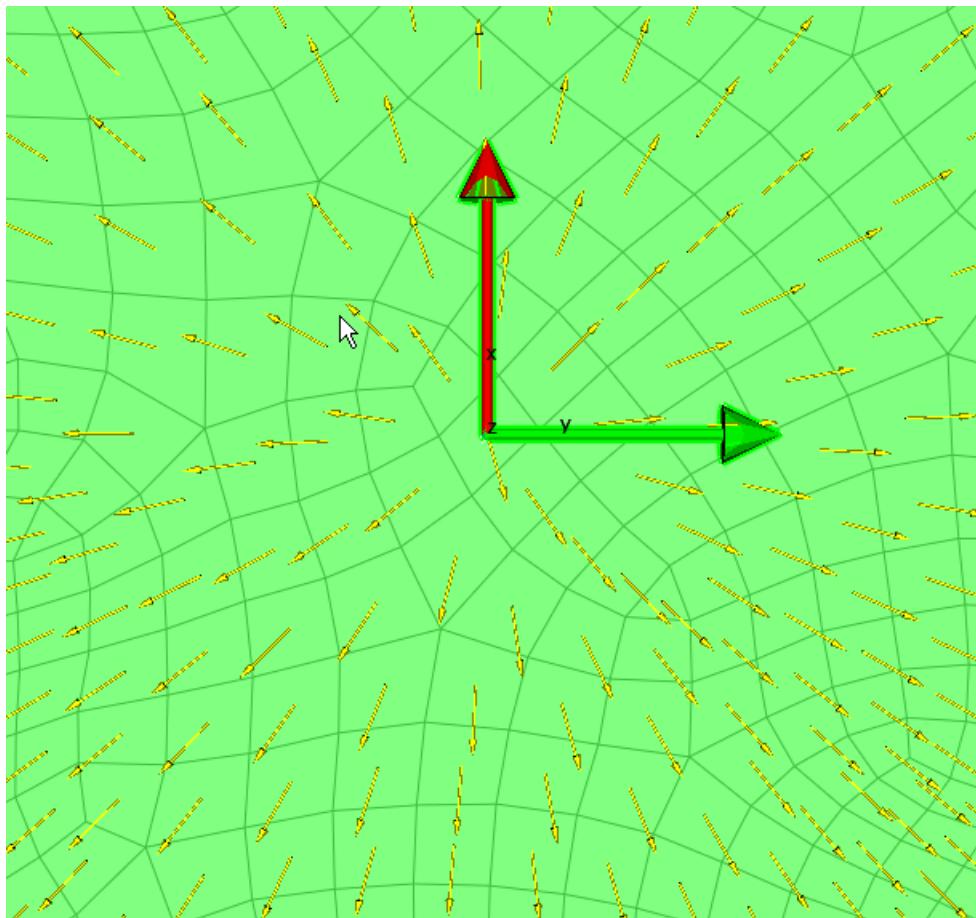
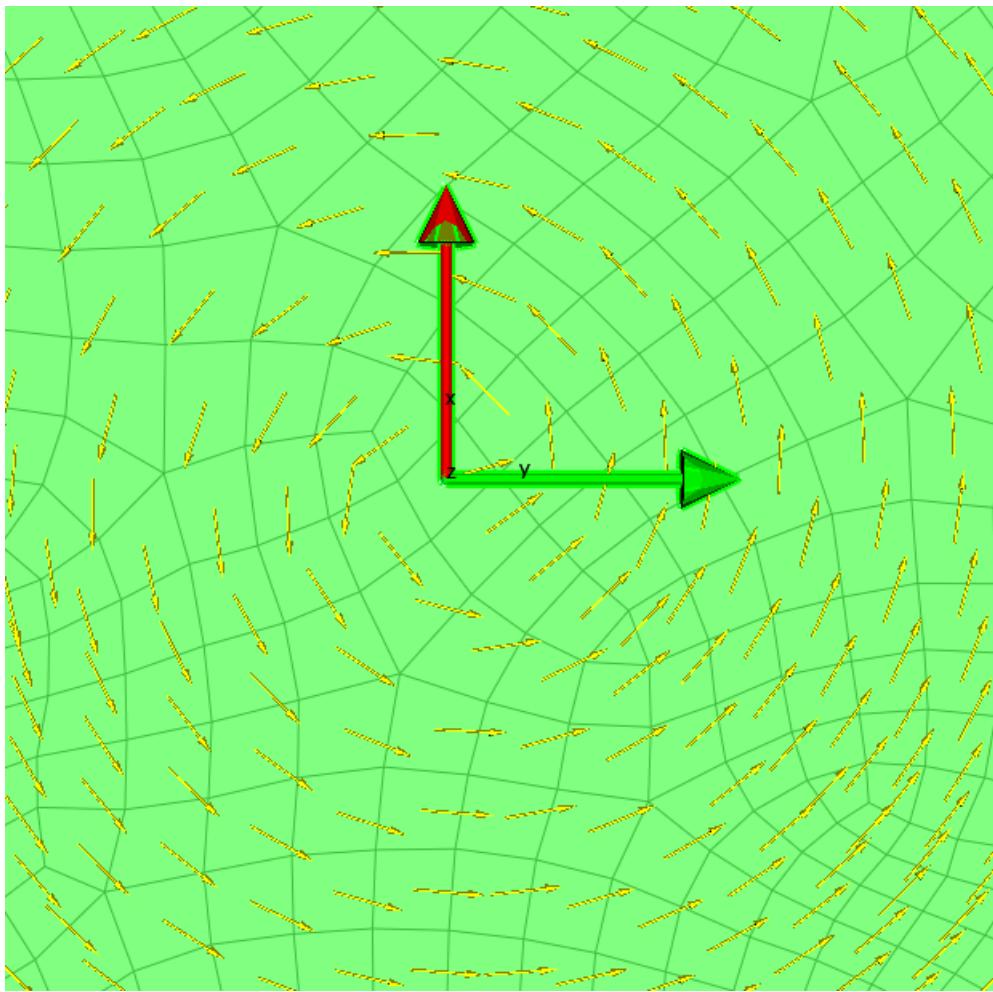
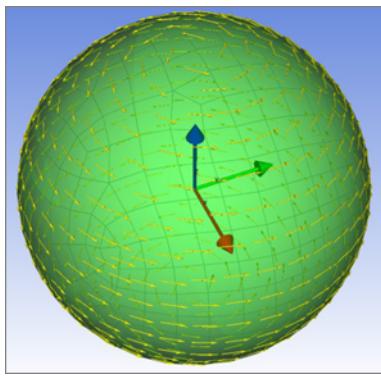


Figure 4.23: Oriented Element Set with a Cylindrical Rosette

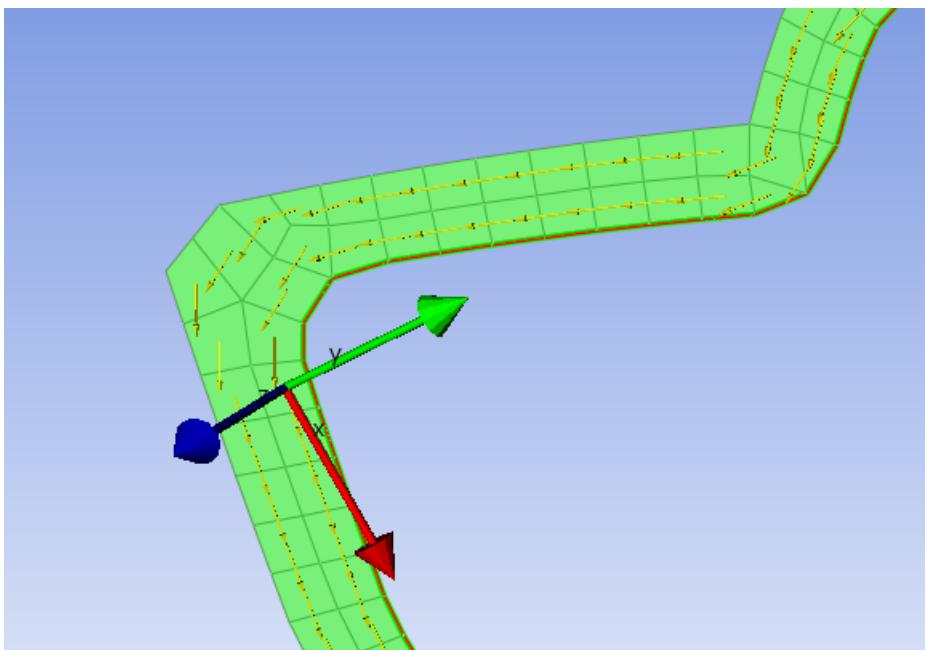
- Spherical: the Spherical Rosette is based on a spherical coordinate system and the reference direction runs circumferentially around the Z axis of the Rosette.

Figure 4.24: Oriented Element Set with a Spherical Rosette

- Edge wise: the Edge Wise Rosette requires the selection of an Edge Set in addition to the usual Rosette definition. The reference direction is given by a projection of the Rosette's X direction and the path of the Edge Set. The X direction of the Rosette coordinate system is projected on to the point on the Edge that is closest to the origin of the Rosette. This determines reference direction along the Edge Set. The

reference direction is reversed by switching the coordinates of the Rosette's X direction. An element within an Oriented Element Set gets its reference direction from the direction of the point on the edge that is closest to the element centroid.

Figure 4.25: Edge wise Rosette



Rosette Definition

Each Rosette is defined by an origin and two vectors. Enter the origin by clicking on an element or a node. Or just type the coordinates. By selecting an element, the coordinates of the center of the element are used.

If a direction field is selected in the dialog, the selection of an element returns the normal direction of the element. By pushing down the CTRL key and clicking on a second element ACP will return the direction defined by the two element centers.

If the two directions are not orthogonal the orientation buttons at the bottom of the [Property dialog](#) can be used to create an orthogonal Rosette definition. Direction 1 translates to the X direction of the Rosette and similarly direction 2 translates to its Y direction. The buttons are able to adjust the two directions by keeping one constant and rotating the other in-plane so that they are perpendicular to each other. The Capital Letter on each button denotes which direction is kept constant while the other direction is adjusted. The third vector is directly calculated from the two first directions. Using the button "yz" as an example, the direction of the vector Z stays as it is. The vector Y remains in the plane defined by the two directions (Y and Z) but is adjusted to be orthogonal to the vector Z. The X vector is calculated from the Y and Z vectors.

Warning

After the operation, the directions 1 and 2 are changed with the values of the vectors X and Y respectively.

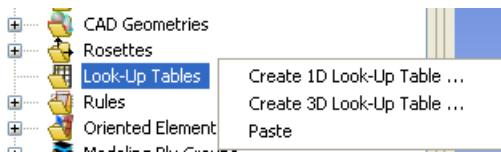
4.1.6. Look-up Tables

Several features in ANSYS Composite PrepPost can be defined as *Tabular Values*.

A Look-up Table can be a 1D (linear interpolation) or 3D (spatial interpolation) Table. It contains at least a *Location* column (1 or 3 columns, one for each coordinate). Afterwards *Direction* columns (3 columns) and/or *Scalar* columns can be added into a Look-up Table. The Look-up Table can also be imported and/or exported from/to **.csv** files (see [CSV Files](#)). The most efficient way to define tables is to generate the columns in ACP, export the table as CSV-file, edit the file in e.g. Excel or OpenOffice and import the file again.

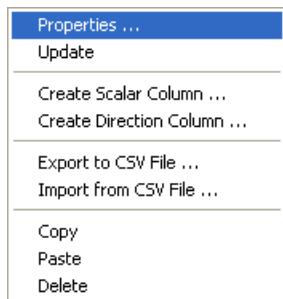
Use the right click menu to define tables.

Figure 4.26: Right-click Menu on Look-Up Tables head node



And the right click menu of the tables can be used to define the columns:

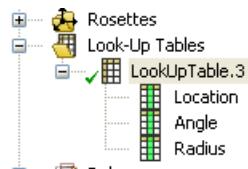
Figure 4.27: Right-click Menu on Look-Up Tables



- Properties: open the properties window of the selected table
- Update: update the selected table
- Create Scalar Column...: define a scalar data (1D array)
- Create Direction Column...: define a vector direction data (3D array)
- Export to CSV File...: export the table to a csv file. Necessary to fill up the table without Python commands.
- Import from CSV File...: read data from a csv file to fill the table.
- Copy: copy the selected table
- Paste: paste a table, which was copied before
- Delete: delete the selected table

For example, the table defined below is a 3D Look-up table with standard 3D array location and two Scalar Column (1D array) Angle and Radius. Each array appears in the table tree.

Figure 4.28: Look-up Table Tree



1D Look Up Table

A 1D Look up table can be used to create a distribution of a scalar or vector in one direction. Such a distribution can be used in:

- the definition of the Oriented Element Set reference direction, see [Oriented Element Sets \(OES\)](#),
 - the definition of the draping angle, see [Draping](#),
 - the definition of the ply thickness, see [Thickness](#).

A 1D Look-Up table is defined by an origin, a direction vector and a table of at least one quantity varying along the direction vector. The values of the defined quantity are inter- or extrapolated to the element centers in the mesh. The 1D Look-Up table uses a scalar product to project the vector between origin and element center on to the Look-Up table direction vector and look-up the desired value at that point.

Figure 4.29: 1D Look-Up Table Properties

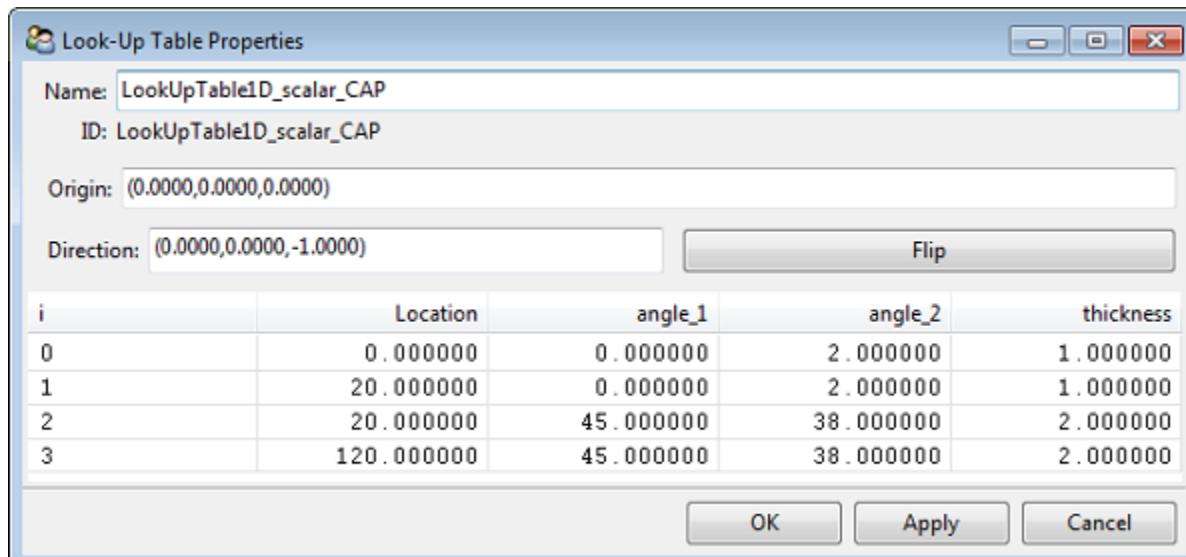
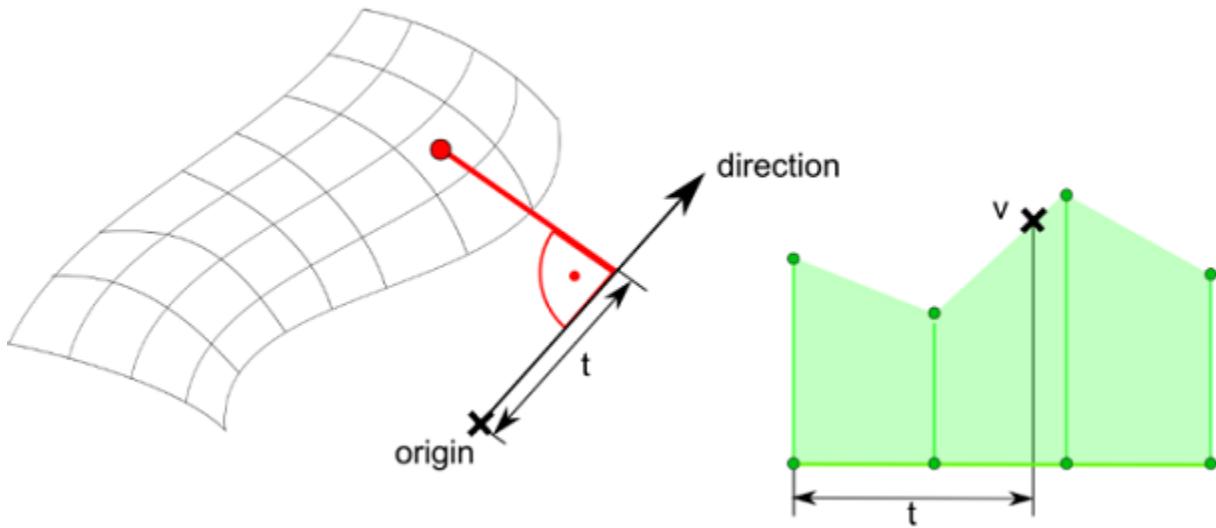


Figure 4.30: Schematic of 1D Look-Up table function

3D Look Up Table

A 3D Look up table can be used in:

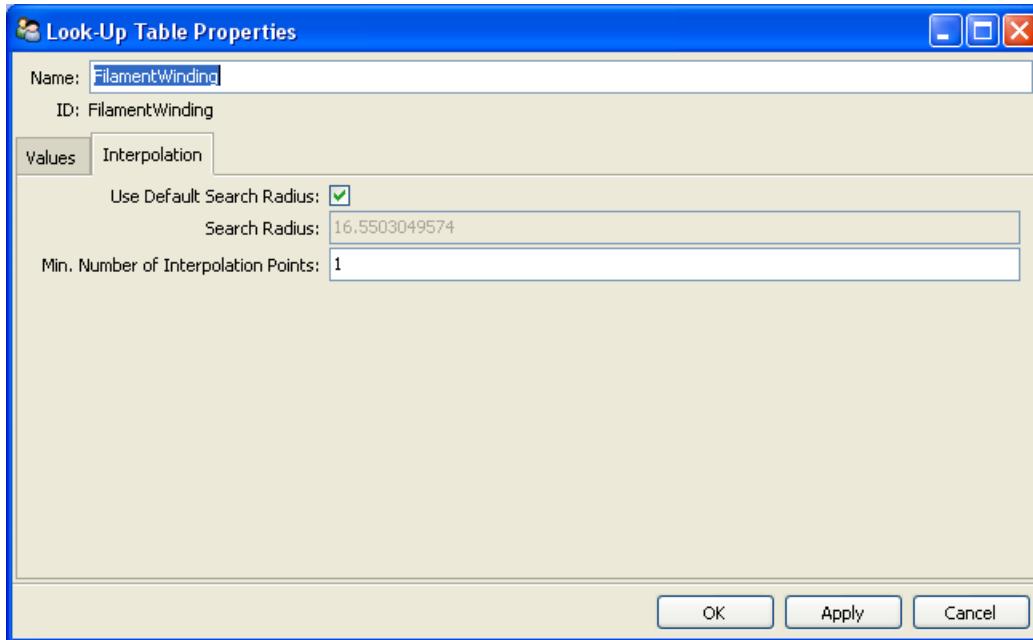
- the definition of the Oriented Element Set reference direction, see [Oriented Element Sets \(OES\)](#),
- the definition of the draping angle, see [Draping](#),
- the definition of the ply thickness, see [Thickness](#).

Figure 4.31: Look-up table edition

Look-Up Table Properties						
Name:	LookUpTable.3					
ID:	LookUpTable.3					
Values	Interpolation					
i	Location.x	Location.y	Location.z	Angle	Radius	
0	-137.565731	408.715816	10.797216	30.826347	137.988804	<input type="button" value="..."/>
1	-130.095930	1573.476249	91.569438	26.389253	159.090896	<input type="button" value="..."/>
2	-82.624354	1431.099409	-19.746520	56.342769	84.951215	<input type="button" value="..."/>
3	97.927396	270.208687	14.537761	45.581346	99.000613	<input type="button" value="..."/>
4	41.078999	1429.095696	-72.433445	58.120592	83.271172	<input type="button" value="..."/>
5	-39.503722	270.494009	91.038467	45.440572	99.239843	<input type="button" value="..."/>
6	91.809301	1017.795433	-48.432448	42.938565	103.801010	<input type="button" value="..."/>
7	117.445883	779.057888	-47.810098	33.892475	126.804341	<input type="button" value="..."/>
8	126.730629	361.860326	-7.311373	33.850940	126.941358	<input type="button" value="..."/>

The values in the Look-up Table are inter- or extrapolated to the model elements position. The interpolation uses the Shepard's method (3D inverse distance weighted interpolation). The *Interpolation* tab allows to enter two parameters. By default ACP evaluates a reasonable search radius and the number of interpolation points is 1.

Figure 4.32: Look-up table interpolation parameters



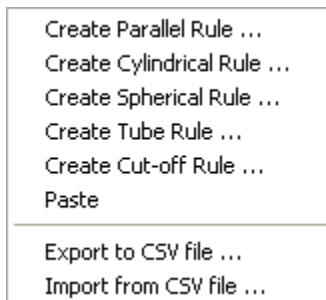
- **Search Radius:** This radius defines a pinball. Only the element centers, which are included in this pinball, are used in the interpolation.
- **Min. Number of Interpolation Points:** If there are no element centers (or not enough if >1) in the pinball, the *Search Radius* is increased until the pinball includes at least the defined numbers of interpolation points (element centers).

The Look-up Tables are also used in the HDF5 Import function. See [HDF5](#).

4.1.7. Rules

A *Rule* allows to select elements through geometrical operations. These selections can be combined with *Oriented Element Sets* or *Modeling Plies* to define plies of arbitrary shape. The final extension of the ply is the intersection of the *Rule* and the selected *Oriented Element Sets*. This feature can be used to define local reinforcements (patches) or staggering. The user can select between different rule types which are explained in the sections below. It is also possible to combine different rules.

Figure 4.33: Rules context menu



When multiple rules are used in combination they behave like sequential filters. If there is no common area of intersection then no elements are selected. Two rules have to overlap to select any elements.

The *.csv file interface of ACP offers to create or modify rules externally (Excel, Open Office), share them with other ACP user working on the same model or even on a different project.

Geometrical Rules

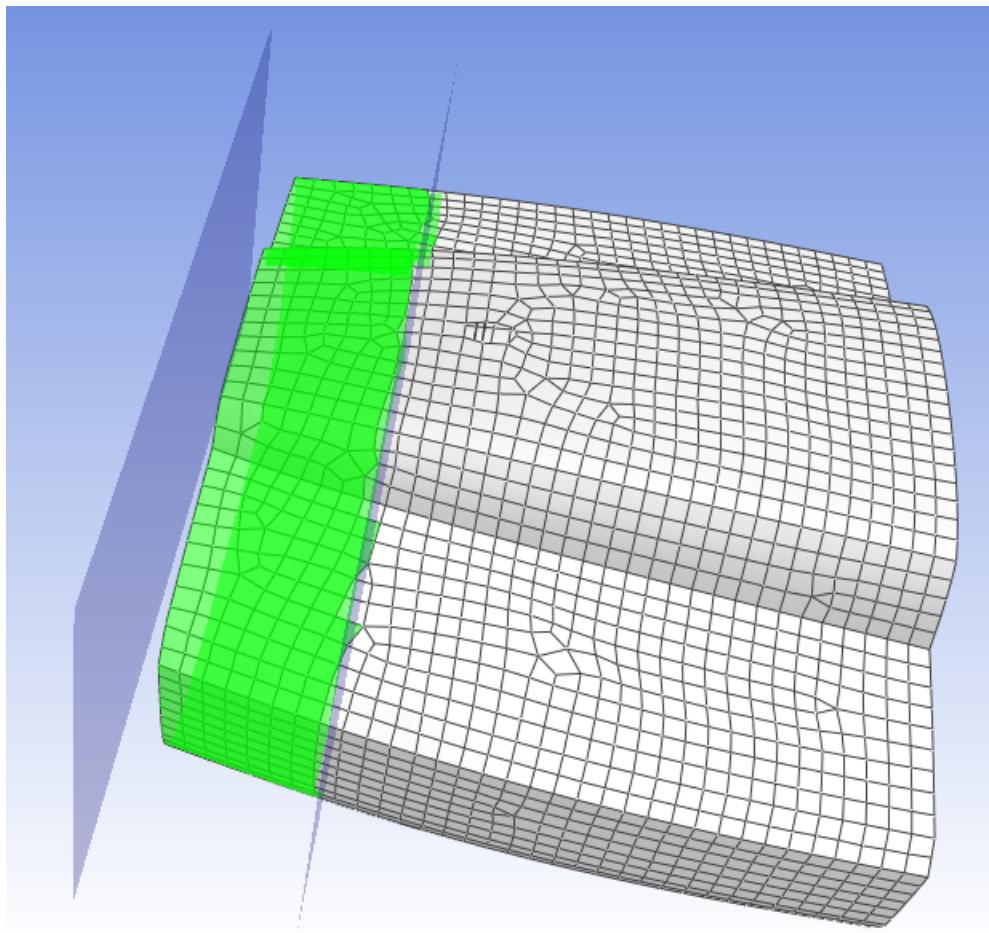
Figure 4.34: Definition of a parallel rule



The rule type Parallel, Cylindrical and Spherical Rule are simple shapes which can be defined by a few parameters:

- Parallel: Defined by two parallel planes. The planes are defined by an origin, a normal vector and two distances (offsets of the planes from the origin along the normal vector).
- Cylindrical: The cylinder is defined by an origin and the vector of the axial direction and the radius. The cylinder has infinite height.
- Spherical: The sphere is defined by the center and the radius.

Figure 4.35: Example of a parallel rule



Relative Rule Type

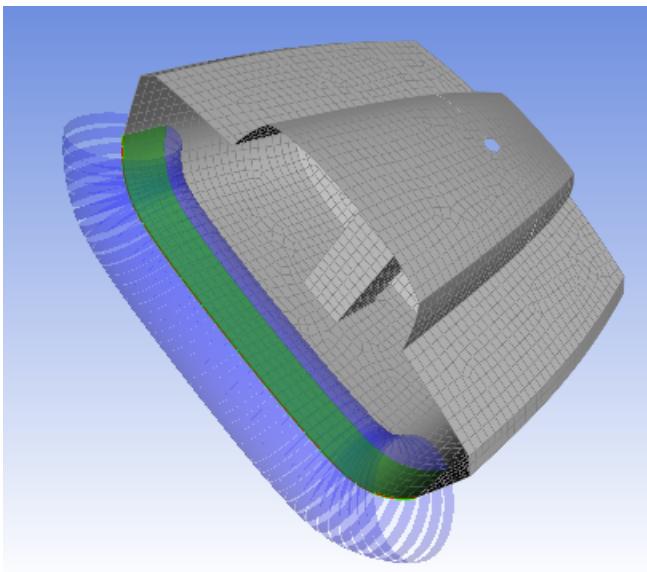
It is possible to define a rule as *relative*. In this case the rule parameters define the ratios of the rule relative to the *Modeling Ply* dimensions. Take care that the plotted rule is evaluated from the global geometry dimensions and that this plot does not represent the final shape of the rule which depends on the modeling ply. Select the ply and check the highlighted elements which represent the final shape of the ply.

Include Rule Type

The *Include Rule Type* option can be used to select the elements inside the geometry or outside. Per default the option is active as shown in the figure above. If the option is inactive, the inverse selection is in effect. E.g plies with a hole can be created through a *Cylindrical Rule* and disabled *Include Rule Type* option. It applies to all geometrical rules as well as the *Tube Rule*.

Tube Rule

A *Tube Rule* is a cylindrical rule of variable axial direction. The longitudinal direction is defined by the *Edge Set* and the radius defines the diameter of the cylinder.

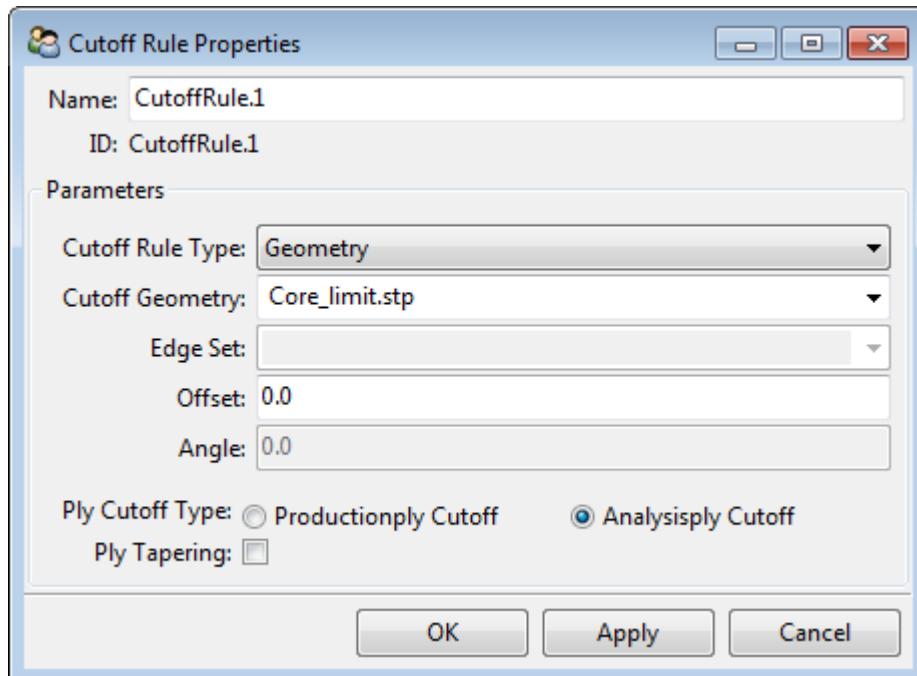
Figure 4.36: Example of tube rule

Cutoff Rules

The *Cutoff Rule* acts as a cutting operation on the composite layup. In contrast to the other rules that affect the in-plane directions of the ply the *Cutoff Rule* also considers the laminate thickness. A Cutoff Rule can be defined by a geometry or a taper. Using a Geometry Cutoff Rule, the ply is cut at the intersection with the *CAD Geometry* taking into account the thickness of the laminate. This means that a skew-whiff surface can be used to define the tapering near a trailing edge of a blade for example. A Taper Cutoff Rule, on the other hand, cuts plies based on an edge and a taper angle. Both are explained in more detail below.

Only Analysis Plies are cut off as a result of the rule. Modeling and Production Plies are not affected. The cutoff rule is similar to a milling operation on built-up structure. In that sense, the full size Modeling and Production Plies are required before the machining operation. The Analysis Plies are the only decisive plies for any structural computation.

Figure 4.37: Cutoff Rule Properties

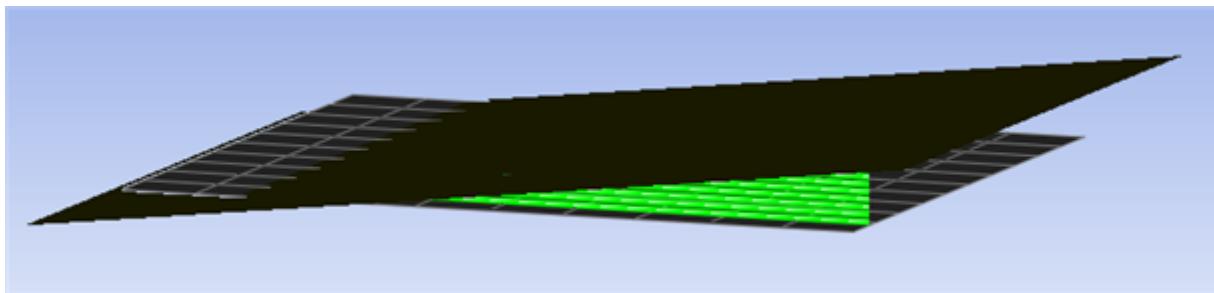


The Cutoff Rule settings are briefly explained here:

- **Name:** name of the rule.
- **Cutoff Geometry:** selection of a CAD geometry (Geometry).
- **Offset:** CAD / taper surface offset. The intersection of the *CAD Geometry/taper edge* and the lay-up can be moved by an offset. The direction and orientation of the offset is defined by the normal direction of the Oriented Element Set.
- **Edge Set:** selection of an edge set (Taper).
- **Angle:** taper angle (Taper).
- **Ply Cutoff Type**
- **Ply Tapering:** control of the cutoff resolution.

Geometry Cutoff Rule

For each element, ANSYS Composite PrepPost determines the position of the ply (including its offset) in relation to the imported surface. There are two possibilities of how the geometry is cut by a CAD geometry - one cutting operation follows the geometry contour, the other divides the ply into either its maximum thickness or zero thickness. This is controlled by the "Ply Tapering" option in the properties tab.

Figure 4.38: Trailing edge with cutoff plies (ply tapering activated)

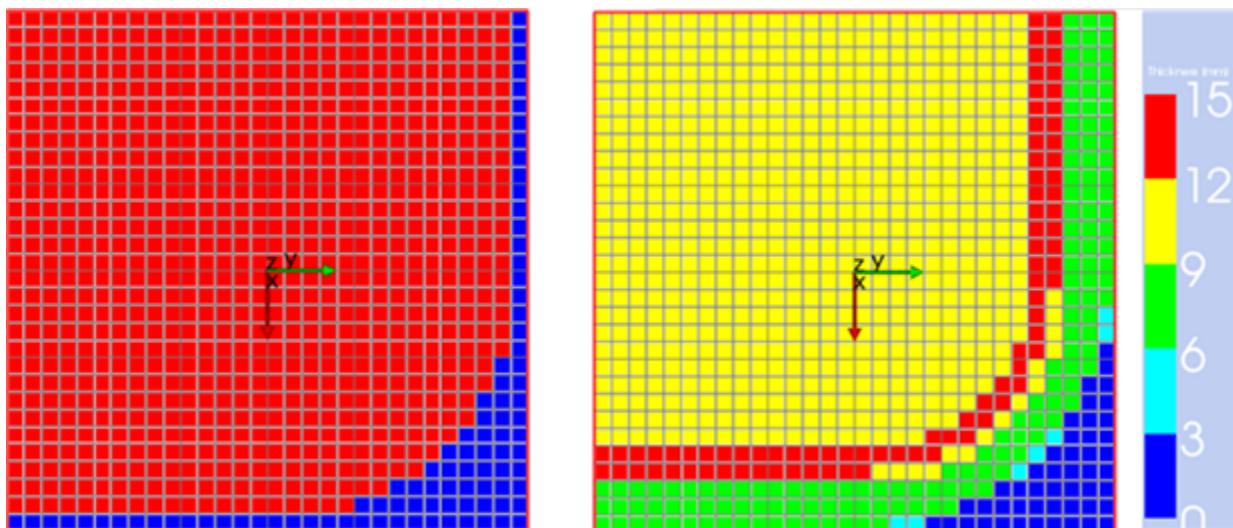
The first method is straightforward in that the ply is cropped if it intersects with the geometry. The ply is cut to match the external geometry. The second method is used to cut the ply at a discrete point. The ply cannot have a varying thickness - it is either at its maximum thickness or it has been entirely cut-off. The ply is cut if the intersection of CAD geometry and ply is less than half of the ply's thickness.

The following figures taken from Tutorial 2 and are presented to explain the concept further. In Tutorial 2, a Geometry Cutoff Rule is applied to the core. A section view of the Cutoff Geometry, applied to two edges, is shown in the figure below. The cutoff has a nominal thickness of approx. two thirds of the core thickness and, towards the outside, it has a multi radii edge. The dashed yellow line in the figure denotes the ply's centerline.

Figure 4.39: Section of the Cutoff Geometry

When Ply Tapering is activated the core is cropped to match the Cutoff Contour, as shown in the right figure below. If Ply Tapering is, however, deactivated then the resulting core thickness is at its full thickness everywhere the intersection of the geometry is above the core centerline. Everywhere else, the core is completely cut-off.

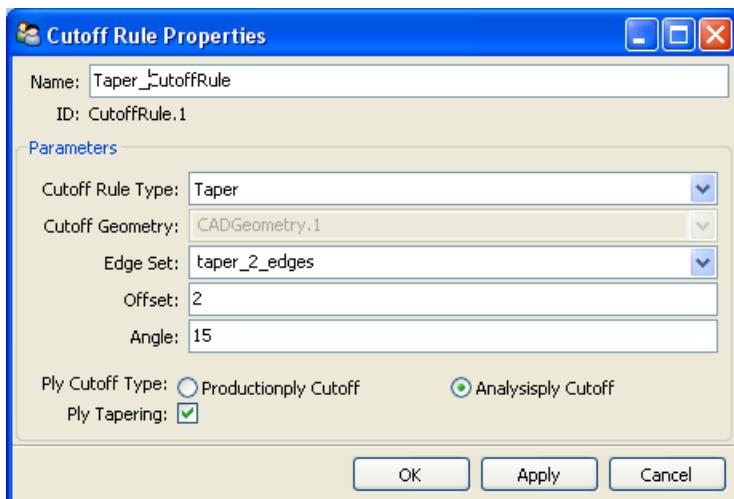
Figure 4.40: Core thickness without ply tapering (left) and with ply tapering (right)



Taper Cutoff Rule

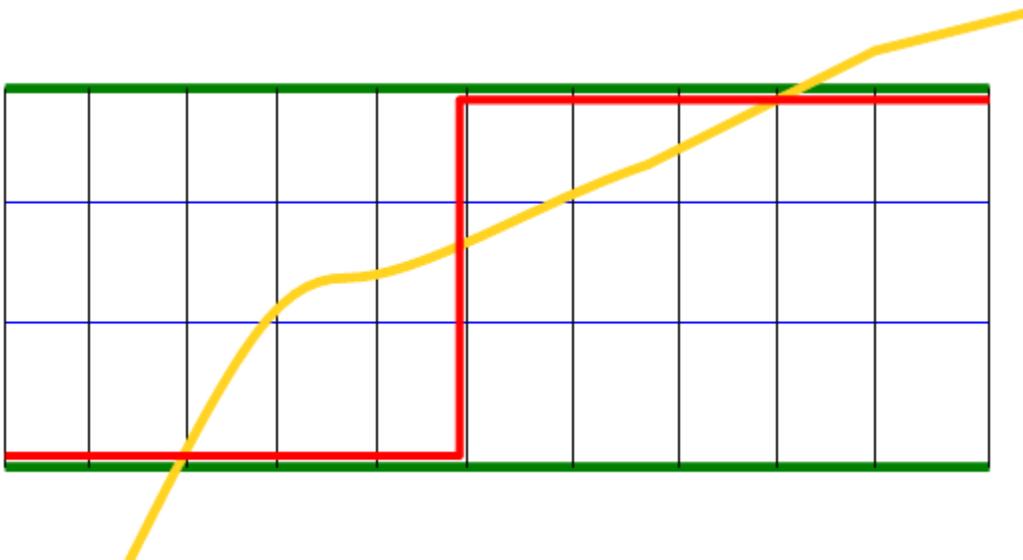
The second way of using the cutoff rule is to define a taper with an edge set and a tapering angle. The area close to the edge has to be sufficiently meshed for the taper cutoff to work. Similar to the Geometry Cutoff, the ply is cutoff in the taper zone. Only if Ply Tapering is selected will there be a gradual taper over thickness.

Figure 4.41: Taper cutoff rule definition



Ply Cutoff Type

Consider a Stackup of three Fabrics for this example. In the pictures below, a laminate of one *Stackup* is shown marked with the green lines. The blue lines highlight the *Analysis Plies* of the *Stackup*. The black lines represent the mesh and the red line indicates the final laminate resulting from the Cutoff rule.

Figure 4.42: Section with the production ply option

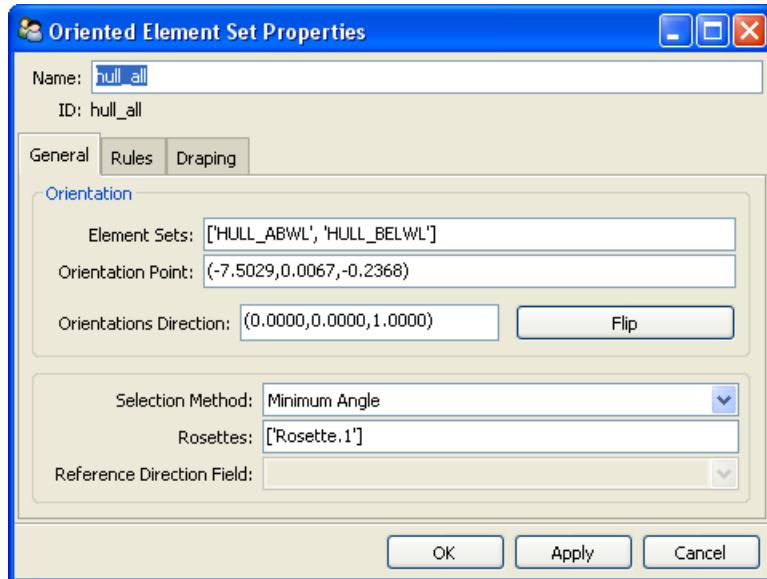
The *Cutoff Rule* with *Ply Tapering* will result in a smoother section. The ratio between the area of the section cut by the *CAD Geometry* and the uncut section is calculated. The same ratio is applied to the ply thickness for the considered element.

Figure 4.43: Section with the analysis ply with tapering option

4.1.8. Oriented Element Sets (OES)

Element Set and Orientations

An *Oriented Element Set* is an *Element Set* with additional information about the element orientations. The orientation direction of an *Element Set* is responsible for setting the stacking direction of the associated layup. The reference direction on the other hand is responsible for setting the 0° direction of the associated layup. These two directions must be defined while other parameters like *Rules* and *Draping* are optional.

Figure 4.44: Definition

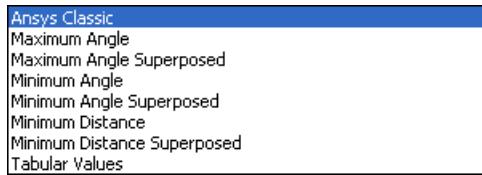
An *Oriented Element Set* is defined by:

- Name: name of the OES as it will be used in further definitions
- Element Sets: underlying elements for the OES definition
- Orientation Point: the offset direction is defined at this point. The point should be inside and close to the reference surface. Otherwise the mapping of the offset direction can result in wrong results.
- Orientation directions: vector defining the offset (normal) direction at the *Orientation Point*.
- **Reference direction: defines the 0° direction of the OES**
 - Selection method: defines the mapping algorithm for the *Rosettes* if more than one Rosette is used. More details are given in the Section [Reference Direction](#).
 - Rosettes: select one or several Rosettes defining the reference direction for each element through the selected method.
 - Reference Direction Field: only applicable to the tabular values method. Defines the direction column of a *3D Look-up Table*.

Use the *Flip* button to reverse the offset direction.

Reference Direction

The reference direction can be defined by *Rosettes* or from tabular values. A single Rosette is sufficient for the definition of the reference direction. Multiple *Rosettes* can be selected for one Oriented Element Set to obtain complex reference direction definition. In this case, a Selection Method must be used to determine which Rosette is applicable to what part of the Element Set. The Selection Method offers several interpolation algorithms listed below:



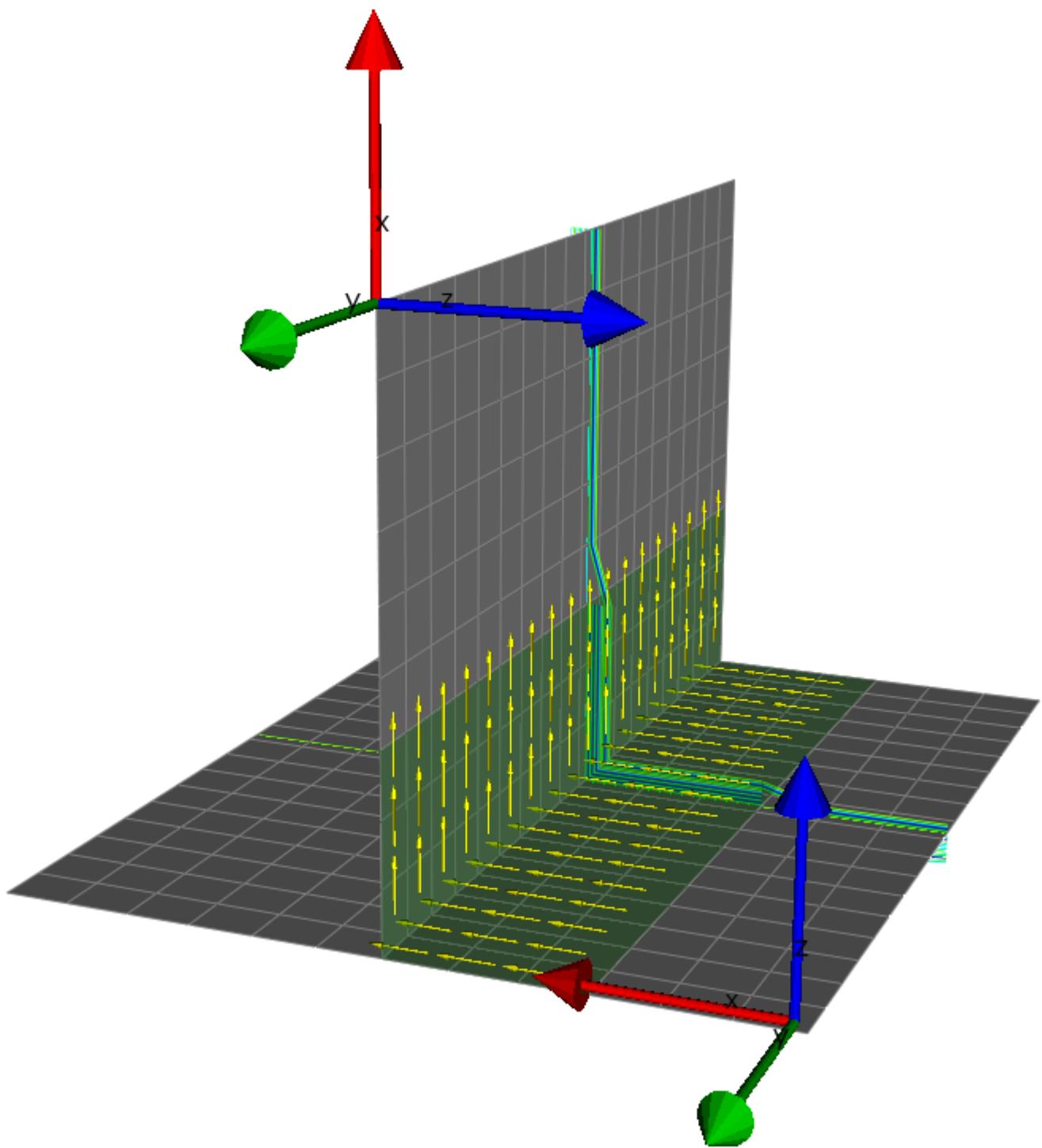
- **Ansys Classic:** the coordinate system is projected on the elements as defined in ANSYS (See *Help/Mechanical APDL /Element Reference/2.3.1. Element Coordinate Systems*)
- **Maximum Angle:** The coordinate system, from the ones selected, in which the Z direction has the maximum angle with the element orientation, is selected to define the reference direction of the Oriented Element Set
- **Maximum Angle Superposed:** Same as Maximum_Angle but all the chosen coordinate systems are considered and weighted by the maximum difference angle direction
- **Minimum Angle:** Default. Same as Maximum_Angle but with the minimum angle
- **Minimum Angle Superposed:** Same as Minimum_Angle but all the chosen coordinate systems are considered and weighted by the minimum difference angle direction
- **Minimum Distance:** Take the nearest coordinate system of the element to define the reference direction of the Oriented Element Set
- **Minimum Distance Superposed:** Same as Minimum_Distance but all the chosen coordinate systems are considered and weighted by the distance to the element
- **Tabular Values:** The orientation definition is interpolated from the values put in a *Look-up Table* (see [Look-up Tables](#)). The table must include the location values and a direction column.

Note

If the determination of an element's reference direction fails, an alternate computation method is used and a warning will be issued. In this case, it is recommended to verify the reference directions of the affected Oriented Element Set.

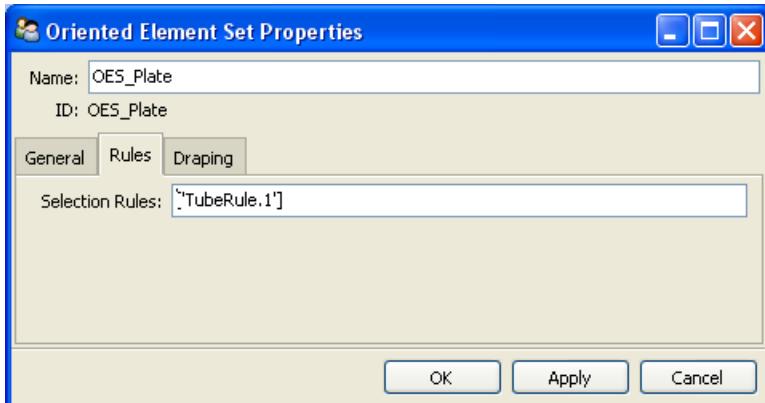
The bonding laminates of the T-Joint Example are a case where a Minimum Angle Selection Method is suitable.

Figure 4.45: The reference direction of a bonding laminate defined by two Rosettes and a Minimum Angle Selection Method



Rules

The OES can be intersected with one or several *Rules*. The intersection of all selected entities ('Element Sets' and 'Rules') defines a new extension of the OES.

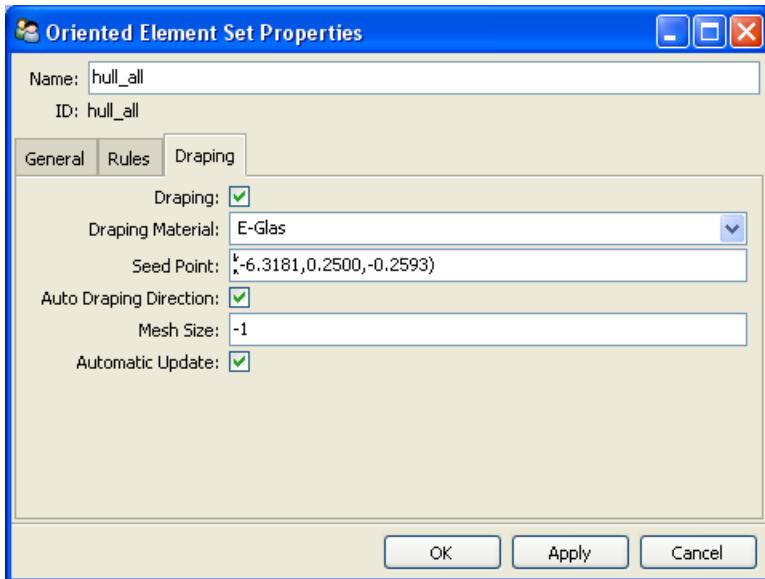
Figure 4.46: Rules

Warning

As the *Cutoff Rules* use the lay-up definition to calculate the cut location, this *Rule* has no influence in this case.

Draping

If the *Draping* option is activated, all plies derived from this *Oriented Element Set* will use this *Draping Data*. If the *Draping* option is also activated in the *Modeling Plydefinition*, the draping data of this OES will be overwritten. More info about draping are given in Section [Draping](#) and in the theory documentation ([Draping Simulation](#)).

Figure 4.47: Draping

- Option Draping: activates / deactivates the draping evaluation
- Draping Material: sets the draping coefficients of the selected material
- Seed point: starting point for the draping evaluation
- **Draping Direction**

- If *Auto Draping Direction* is active, a default draping direction is evaluated
- Mesh size: defines the draping mesh size. If this mesh size is defined as negative, the default mesh size is used.
- Automatic Update: If active, will automatically recalculate the draping correction after any change

4.1.9. Modeling Ply Groups

In the section Modeling Ply Group, the desired composite lay-up can be defined. Beforehand, it is necessary to specify at least an Oriented Element Set and a Material (Fabric, Stackups or Sub Laminate).

The ply definition can be organized into *Ply Groups*. These *Ply Groups* have no influence on the ply-ordering and definition but help to group the composite definitions. It make sense to define one ply group for each substructure (e.g hull, deck, bulkhead for a boat).

Within a *Ply Group* plies can be created. The lay-up is defined as it would be in production. The first ply is also first in the stacking sequence. The lay-up can be tailored by specifying the orientation, layering, geometrical rules, draping settings and edge tapering for each ply.

A lay-up can also include an *Interface Layer* for carrying out a fracture analysis of a composite solid model in Workbench Mechanical. The interface layer is a separation layer in the stacking sequence. It can be used to analyze the crack growth of existing cracks. The crack topology is defined with an interface layer in ACP while all other fracture settings need to be specified in Workbench Mechanical. The interface layers are exported as INTER204 or INTER205 elements and can be used to set up a Cohesive Zone Model (CZM) or a Virtual Crack Closure Technique (VCCT) analysis. They can also be used to define contacts zones between two layers. Please refer to the Workbench Mechanical Help for further information.

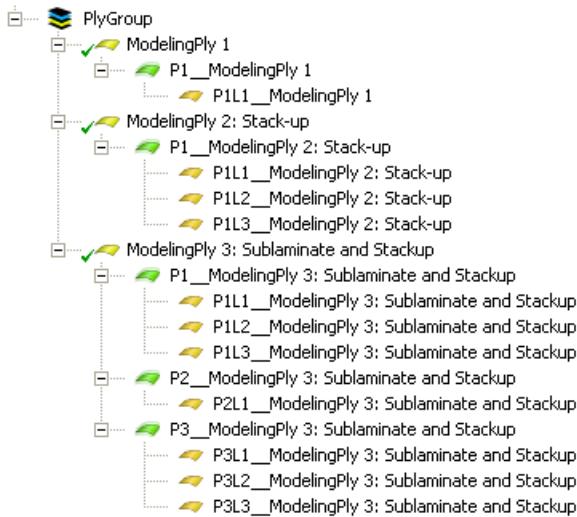
4.1.9.1. Ply Group Structure

The *Ply Group* node has three sub-levels:

- Modeling Ply (MP): the ACP lay-up is defined at this level. From the information given at this level, the other two levels are built automatically.
- Production Ply (PP): derived from the MP definition (*Material* and *Number of Layers*) the PPs are generated. A *Fabric* and *Stackup* is one PP. But a *Sublamine* typically contains more than one PP. In addition the *Number of Layers* option is also propagated to this level.
- Analysis Ply (AP): the analysis plies describe the plies used in the section definition for the ANSYS solver. A *Fabric* results in one AP, a *Stackup* with two *Fabrics* has two AP and so on. A PP without AP indicates that the resulting AP has no elements in is therefore not generated.

In the example below, there are three different MP separated by one interface layer:

- the first one is defined with a single Fabrics,
- the second one with a stackup of two Fabrics,
- the third one with a sub-laminate defined with three production plies (Stackup, Fabric, Stackup), which results in five AP.
- The interface layer lies between the second and third MP.

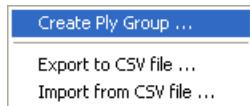
Figure 4.48: Object tree of a layup definition

Shortcuts exist to easily navigate through the ply definition. Use the square brackets keys ([and]) to move up and down through the plies.

4.1.9.2. Modeling Ply Group Context Menu

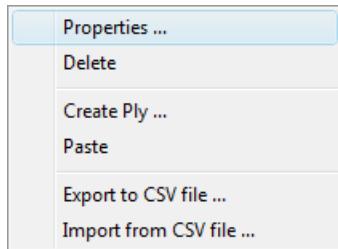
The context menu of Modeling Ply Group has these options:

- *Create Ply Group*: A ply group can be created and a name assigned.
- *Export to CSV file...*: export all plies with all modeling ply definitions to a *.csv file (see [Import from/ Export to CSV Files](#))
- *Import from CSV file...*: import plies from a *.csv file (see [Import from/ Export to CSV Files](#))

Figure 4.49: Context menu of Modeling Ply Groups

4.1.9.3. Ply Group Context Menu

The context menu of a *Ply Group* has these options:



The menu contains:

- *Properties...*: edit the modeling ply properties
- *Delete*: delete the selected modeling ply group

- *Create Ply...*: create and define a new ply in the [Modeling Ply Properties](#)
- *Create Interface Layer...*: create and define an interface layer for fracture analysis in the [Interface Layer Properties](#).
- *Paste*: paste a copied ply
- *Export to CSV file...*: export the whole ply group with all modeling ply definitions (see [Import from / Export to CSV files](#))
- *Import from CSV file...*: import a ply group from CSV file (see [Import from / Export to CSV files](#)).
- *Export Plies...* export the ply offset geometry as a *.stp or *.iges file. (see [Export Ply Geometry](#))

4.1.9.4. Modeling Ply Properties

This is minimal information needed in the *Modeling Ply* definition can be found in the general tab.

Figure 4.50: General information

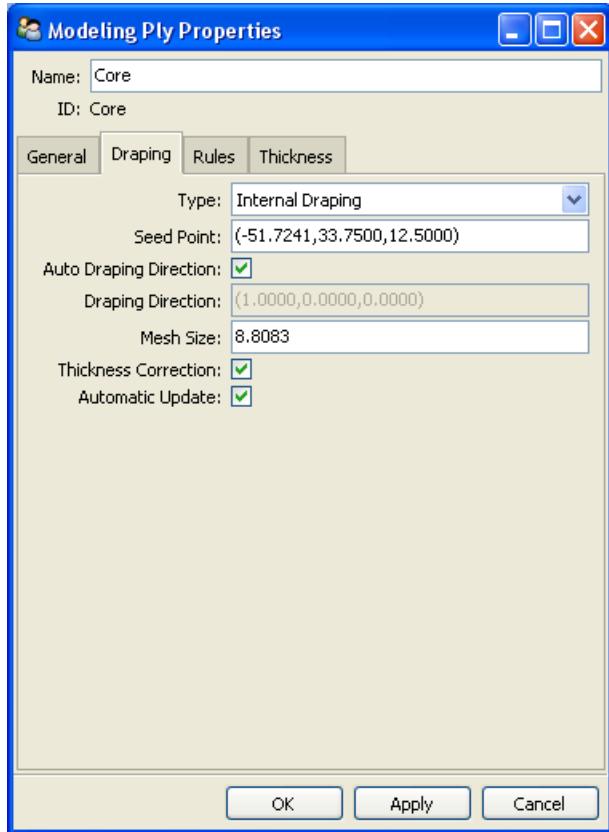


- Name
- *Oriented Element Sets*: defines the offset and material direction
- *Material*: modeling ply material (Fabric, Stackup or Sublaminates)
- *Number of Layers*: the plies are generated X times
- *Active*: if the ply is active or not. Inactive plies are not deleted, but is not considered in any analysis
- *Global Ply Nr*: defines the global ply order. Per default a new MP is added after the last MP of the *Ply Group*. The order of the MPs in the ply groups is equal to the *Global Ply Nr*

4.1.9.4.1. Draping

It is possible to consider the draping effects, which occur during the production process. If defined the draping properties of the assigned *Oriented Element Set* are passed to the MP. If the MP has its own parameters, the draping properties of the OES are overwritten.

Figure 4.51: Draping definition



The draping definition options are:

- **Type:** choose the draping calculation method. See below for more details.
- **Seed point:** starting point of the draping process
- **Draping Direction:**
 - *Auto Draping Direction:* uses default draping direction or a user defined direction
- **Mesh Size:** defines the draping mesh size. If this mesh size is defined as negative, ACP will use the default mesh size
- **Thickness Correction:** the thickness correction due to draping effects are considered in any calculations (analysis model, section cuts, ...)
- **Automatic Update:** if active, the draping will be updated automatically after any modifications

Per default draping is not considered in the analysis. The other two options are *Internal Draping* or *Tabular Values*.

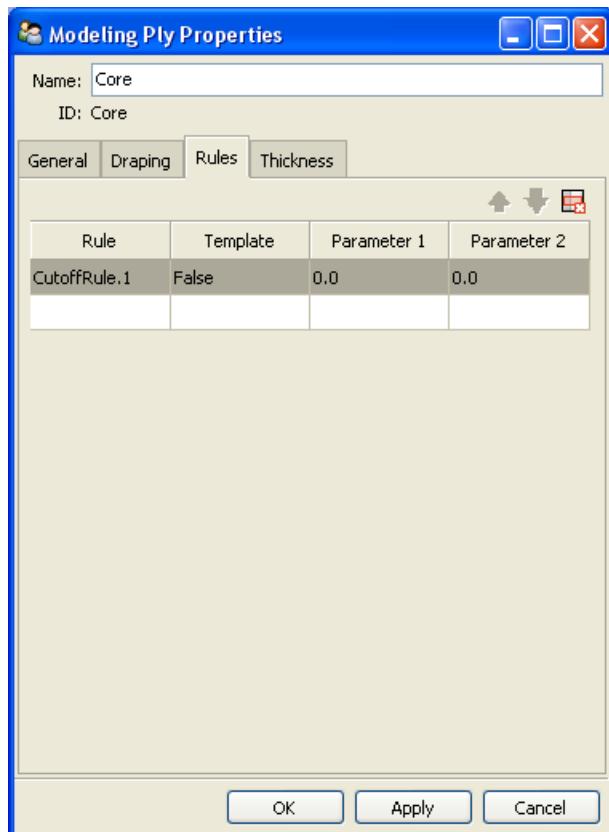
Figure 4.52: Draping Calculation options



More info in [Draping](#) and in the theory documentation ([Draping Simulation](#)).

4.1.9.4.2. Rules

Like in the definition of an *Oriented Element Set*, the *Modeling Plies* can have their own rules. Again, the intersection of all *Element Sets* and active *Rules* defines the extension of the MP. In addition, the rule parameters can be redefined in the MP definition. This prevents the user to redefine the same rule several time and allows to define a staggering with one rule. In *Select Rules...*, select the original Rule and activate *Template* as *True* and give new parameters.



The template parameters for each rule type are given in the table below:

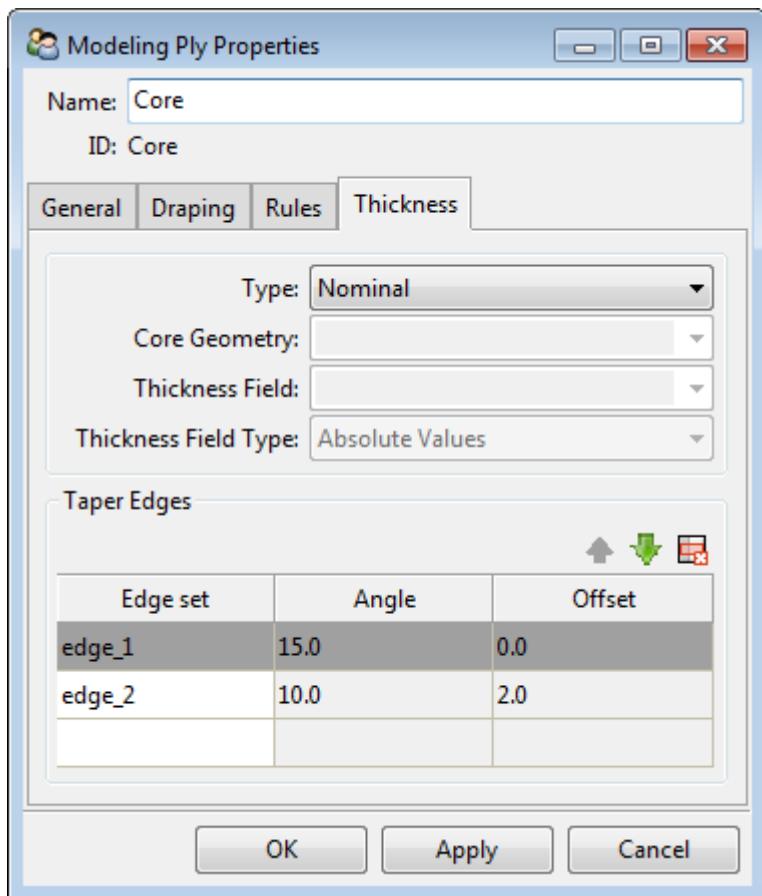
Rule Type	Parameter 1	Parameter 2
Parallel Rule	Lower Limit	Upper Limit
Tube Rule	Outer Radius	Inner Radius
Cylindrical Rule	Radius	/
Spherical Rule	Radius	/

Rule Type	Parameter 1	Parameter 2
Cutoff Rule	/	/

4.1.9.4.3. Thickness

The thickness of the ply is defined by default by the thickness of the ply material.

Figure 4.53: Thickness definition



For *Fabrics* the ply thickness can also be defined by a *CAD Geometry* or *Tabular Values*. The thickness options are:

- *Type*: thickness definition, more info below
- *Core Geometry*: thickness of the CAD geometry is mapped to the FE mesh
- *Thickness Field*: value field is mapped to the FE mesh
- *Thickness Field Type*:
 - *Absolute Values*: values define one-to-one the thickness
 - *Relative Scaling Values*: values in the look-up are scaling factors
- *Taper Edges*: adds an edge tapering to the selected edge set

The ply thickness for each element can be defined by three different ways:

Figure 4.54: Thickness definition options

Nominal
From Geometry
From Table

- Nominal: the thickness defined in Fabrics is used for the thickness definition.
- From Geometry: The thickness is calculated from a CAD Geometry, which is defined in *CAD Geometries*. See more information below. In the case of a complex core ply, it can be helpful to work with a *CAD Geometry* defining the thickness distribution of the core. ACP samples through the *CAD Geometry* for each element and maps the thickness. The thickness is evaluated in the element normal direction.

Figure 4.55: Core geometry

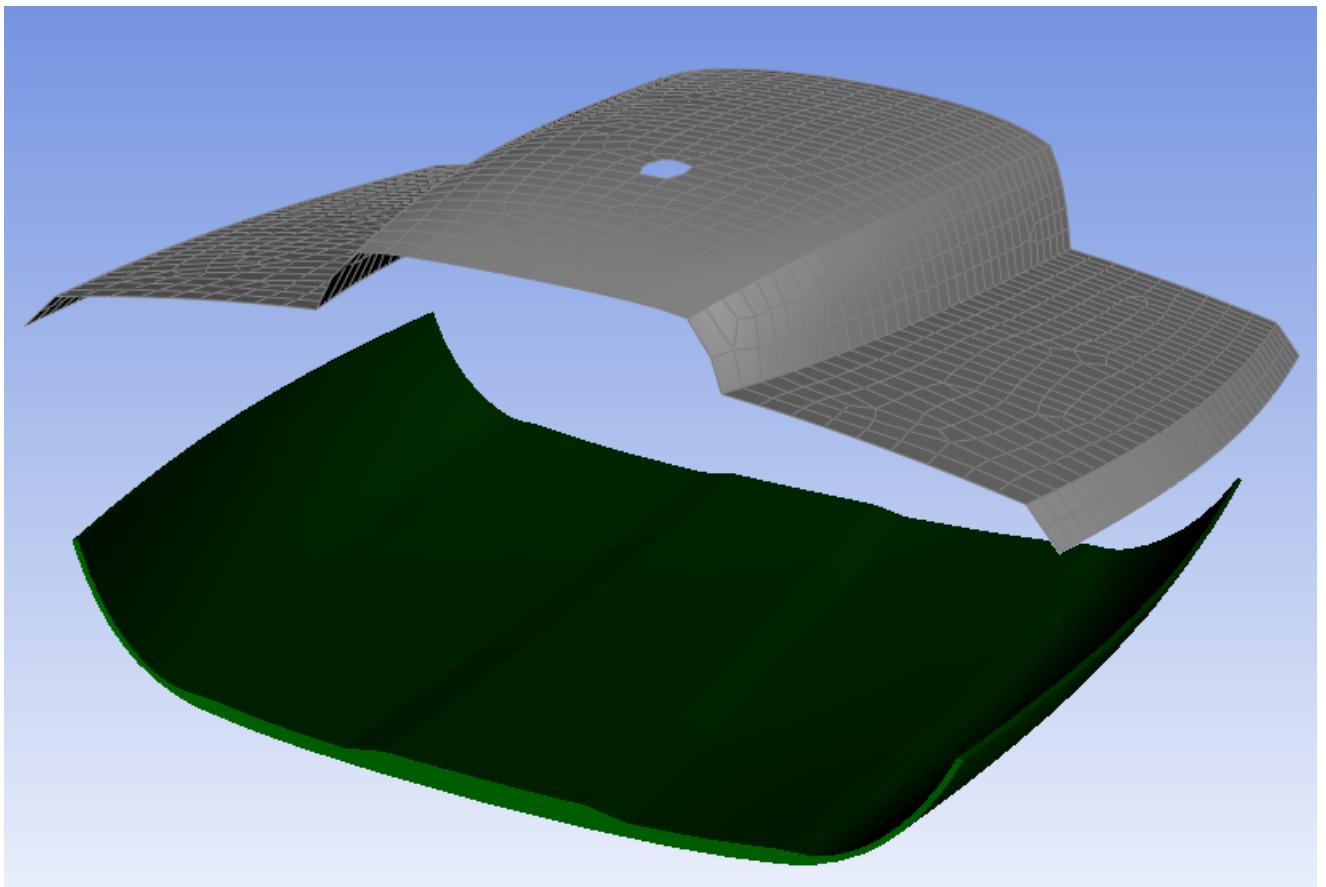
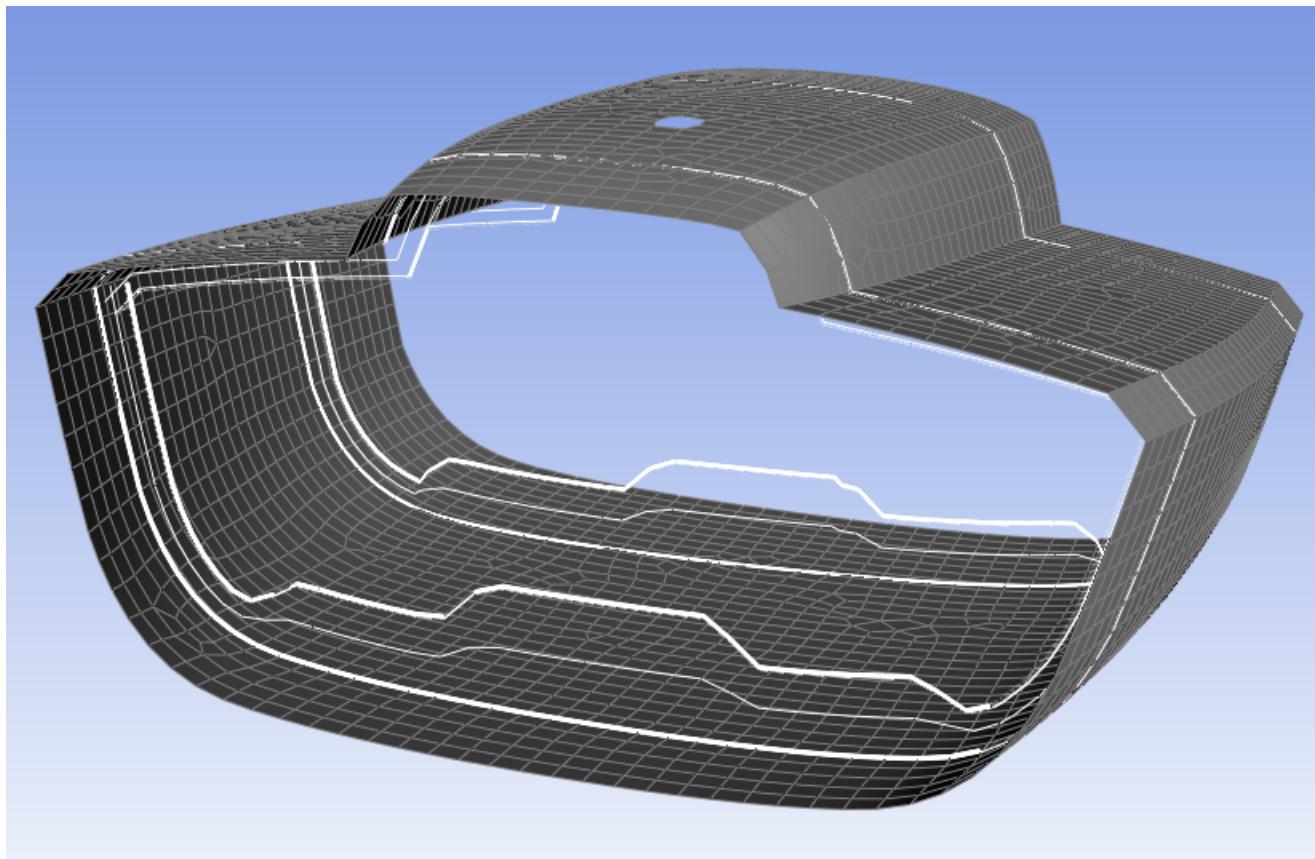
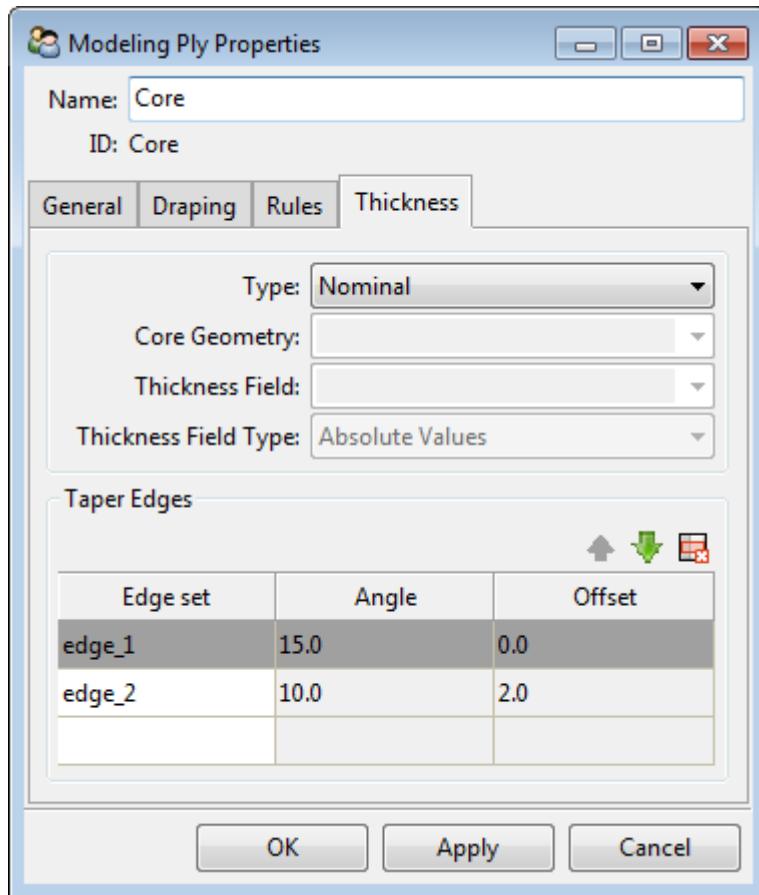
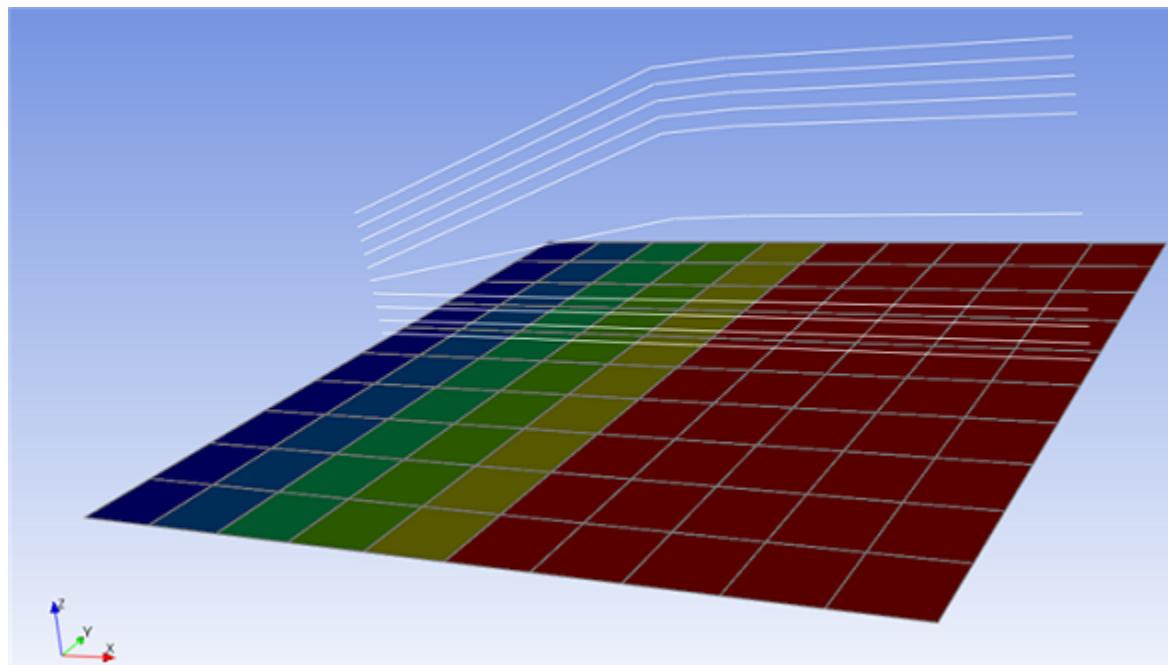


Figure 4.56: Resulting section cut

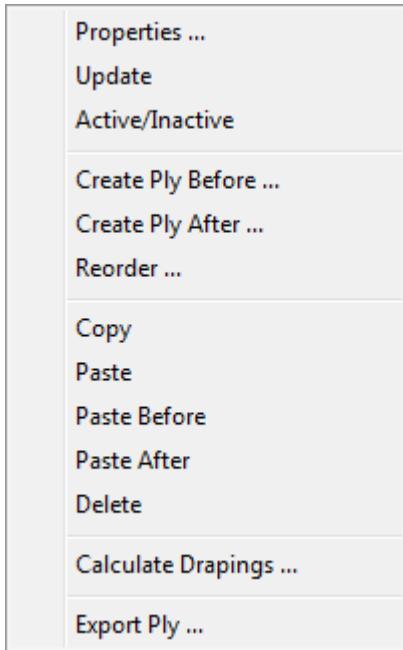
- From Table: The thickness is evaluated from a data field. ACP inter- or extrapolates the thicknesses for each element. One data point contains the global coordinates and the thickness values. The values in the table can be used as absolute or relative thickness. See [Look-up Tables](#) for the definition of the Look-up tables.

It is common that core plies are tapered along the boundary. The taper edge feature allows to define a taper angle and a taper offset for each edge. The figure below shows a 15 degree tapering along the edge on the left. The thickness is 0 at the selected edge and grows with the specified angle. The Taper Edges option is intended for applying a taper angle to a single ply, a core material for example. When applied to multiple Modeling plies the thickness distributions of all plies are superposed. See [Tapering of Multiple Plies](#) for more information.

Figure 4.57: Edge tapering**Figure 4.58: Taper Edge example**

4.1.9.4.4. Modeling Ply Context Menu

The actions of the *Modeling Ply* are quite useful and described below:

Figure 4.59: Right-click modeling ply Menu**The different options in menu are:**

- *Properties...:* opens the modeling ply properties.
- *Update:* updates the selected modeling ply.
- *Active/Inactive:* an inactive ply is still defined in the database, but not considered in the analysis
- *Create Ply Before...:* creates a new ply before the selected one.
- *Create Ply After...:* creates a new ply after the selected one.
- *Reorder...:* allows the user to move the selected ply (or plies if several are selected) before or after another defined ply.



- *Copy:* copies the selected modeling ply.
- *Paste:* pastes a copied modeling ply.
- *Paste Before:* pastes a copied modeling ply before the selected one.
- *Paste After:* pastes a copied modeling ply after the selected one.
- *Delete:* deletes the selected modeling ply.
- *Calculate Drapings...:* allows the user to calculate the draping data when the *automatic update* is not active in the draping ply definition

- *Export Ply...:* see [Export Ply Geometry](#)



4.1.9.5. Interface Layer Properties

The Interface Layer is defined by two sets of surfaces:

- The first set is the total surface of the open interface and the surface along which a crack can propagate. It is defined by the Oriented Element Set in *Interface Layer Properties* tab *General*.
- The second set is the surface of the open interface. It is defined by the (Oriented) Element Set in the *Interface Layer Properties* tab *Open*.

Furthermore, the Interface Layer can be deactivated with a check box and its Global Numbering altered. The Interface Layer is only taken into consideration in the solid model generation and further processes. All shell based analyses ignore any interface layers.

Figure 4.60: Interface Layer Properties - General

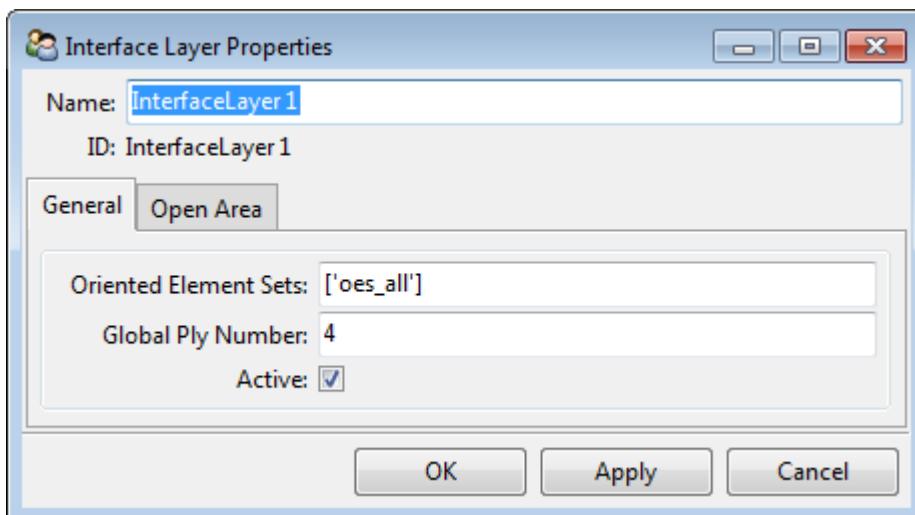
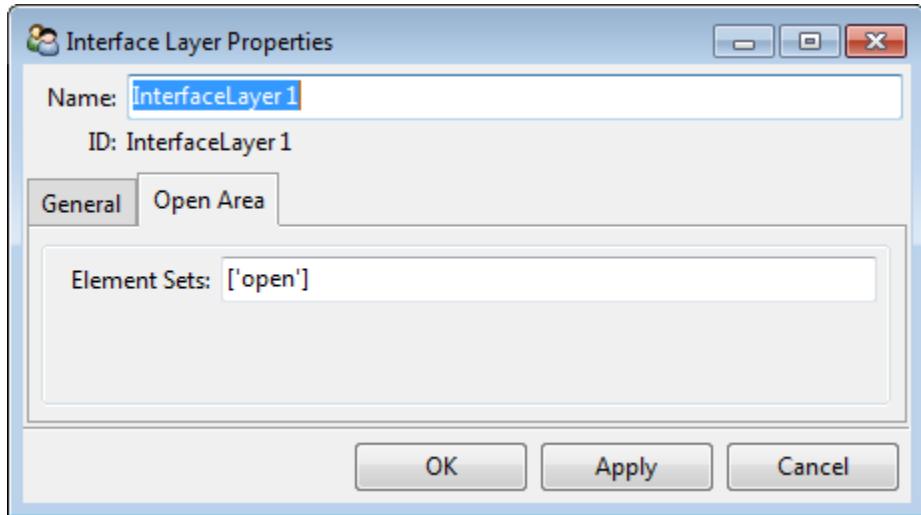
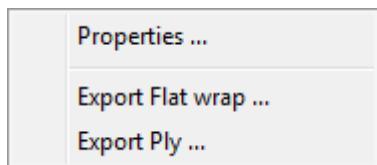


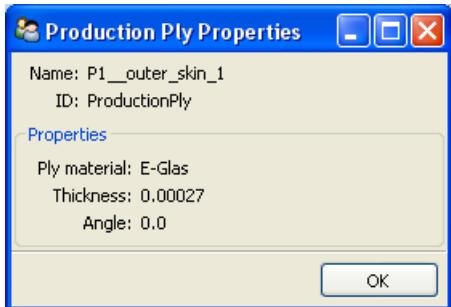
Figure 4.61: Interface Layer Properties - Open Area

4.1.9.6. Production Ply

The context menu of the *Production Ply* is:

Figure 4.62: Menu

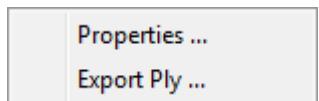
- *Properties...:* the properties of a production ply can't be modified, but can be printed as information.



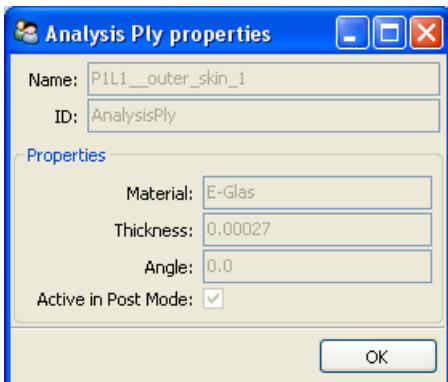
- *Export Flat wrap...:* exports the flat wrap as dxf file for production or design needs. The draping option must be activated to obtain a flat wrap. Refer to [Draping](#) for more information on draping and flat wrap.
- *Export Ply...:* see [Export Ply Geometry](#).

4.1.9.7. Analysis Ply

The context menu of the *Analysis Ply* has these options:

Figure 4.63: Context menu production ply

- *Properties....*: the properties of a production ply can't be modified, but can be printed as information.



- *Export Ply....*: see [Export Ply Geometry](#).

4.1.9.8. Import from / Export to CSV Files

The CSV interface allows to create a spreadsheet with Excel or OpenOffice. This is very efficient if parameters have to be change for many plies. Also Copy and Paste simplifies the modeling process.

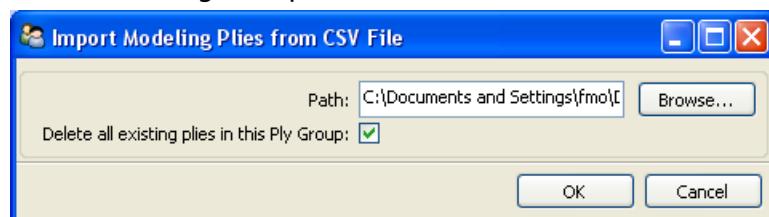
For additional information on the format, see [CSV Files](#).

Export...

All the information are exported to a *.csv file. This file can be used to give the Lay-up information back to a CAD System or can be modified and imported.

Import...

The modified spreadsheet can be imported. The previously defined plies can be deleted or kept by either activating the option or not. The format must be well defined for a clean import.



4.1.9.9. Export Ply Geometry

The ply geometry and fiber directions can be exported to a CAD file format. One possible use of this feature is to check geometry clashes in CAD assemblies. Other uses may include using the data for 3D cutting tool control or for the projection of orientation vectors of the fibers onto tooling surfaces. It is possible to export the surfaces or boundaries with an offset. This give full control over surfaces to be exported.

The Export Ply feature has the following settings:

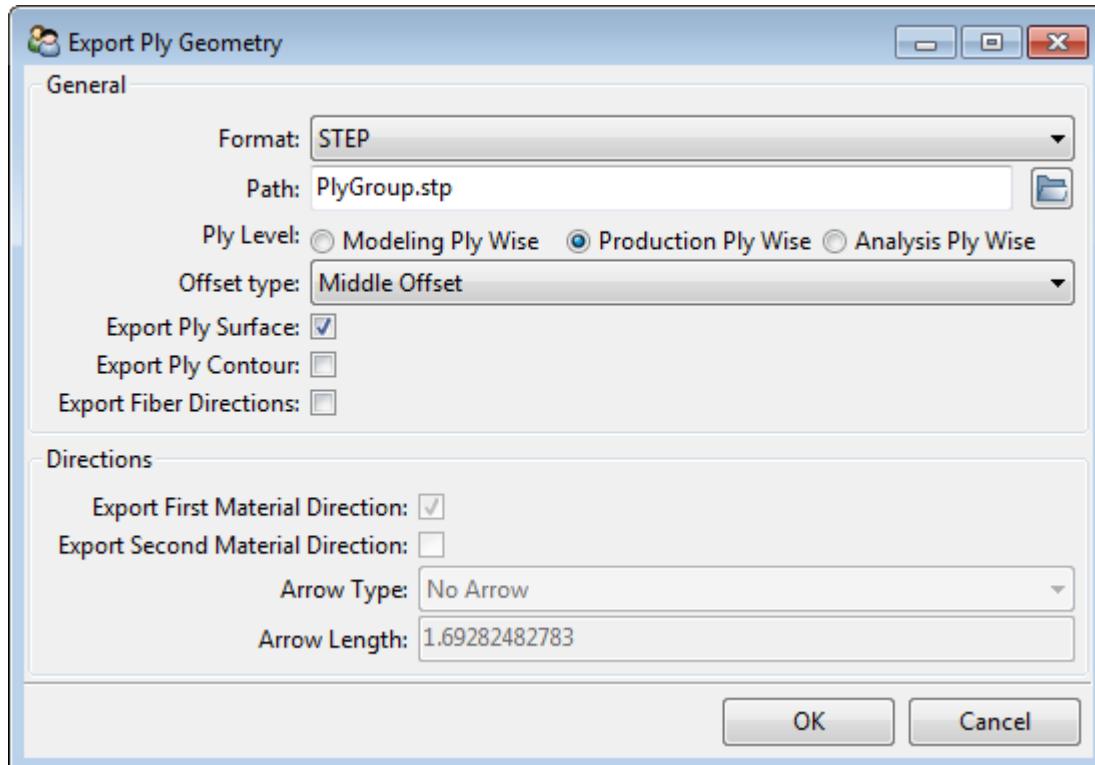
- *Format* : choose between the STEP or IGES CAD geometry file formats.
- *Path* : specify file name and file path.
- *Ply level* : This option is only available for Export Ply Geometry at Ply Group level.

- for Modeling Ply Wise every modeling ply will be exported
- it works likewise for Production and Analysis Ply Wise
- *Offset type*
 - *No Offset* : the ply geometry is exported with no offset to the reference surface (Oriented Element Set).
 - *Bottom Offset* : the bottom surface of the ply relative to the direction of the reference surface is exported.
 - *Middle Offset* the mid-surface of the ply is exported. It is the middle between the bottom and top offset. (default)
 - *Top Offset* : the top surface of the ply relative to the direction of the reference surface is exported.
- *Export Ply Surface* : the ply surface is exported as a shell surface.
- *Export Ply Contour* : the outlining contour of the ply surface is exported as perimeter lines.
- *Export Fiber Directions* : the fiber orientations are exported as orientation vectors.

Settings for Export Fiber Directions

- *Export First Material Direction* : include the first material direction
- *Export Second Material Direction* : include the second material direction
- *Arrow Type* : choose between *No Arrow* (line without an arrowhead), *Standard Arrow* and *Half Arrow*
- *Arrow Length* : specify the arrow length

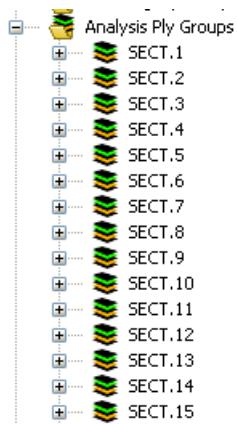
Figure 4.64: Export Ply Geometry Window



4.1.10. Analysis Ply Groups

A model, which already contains lay-up definition, can be imported as post-processing mode. In this case the layup is listed in the *Analysis Ply Groups* folder. Note that no pre-processing features are available if a model is imported as post-processing model. But the post-processing functionality can be used to investigate the strength.

Figure 4.65: Sections definition from a post-processing model

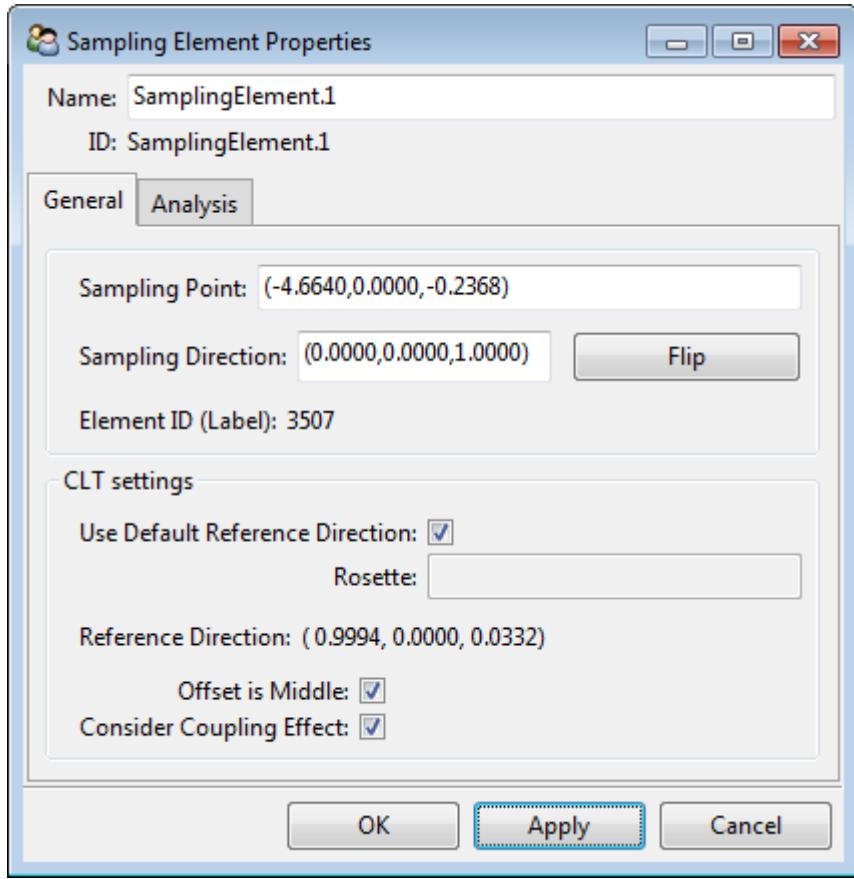


4.1.11. Sampling Elements

Sampling Elements are very useful in the post mode to access to ply-wise results. In addition the *Sampling Element* functionality provides layup plots, through-the-thickness post-processing plots, laminate engineering constants and much more.

General

ACP samples through the element near the given coordinates. After the update all plies (MP, PP and AP) are listed and can be selected for post-processing. In the *General* tab the sampling point and direction can be defined.

Figure 4.66: Definition

- *Sampling Point*: global coordinates; the nearest element will be the sampling element.
- *Sampling Direction*: define a normal direction to the sampling element. The ply sequence will be given in this direction.
- *Element ID (label)*: element number corresponding to the defined sampling point.

A detailed description of the options *Offset is Middle* and *Consider Coupling Effect* can be found in the Section [Analysis Options \(p. 257\)](#).

Use the buttons [and] to navigate easily through the ply definition.

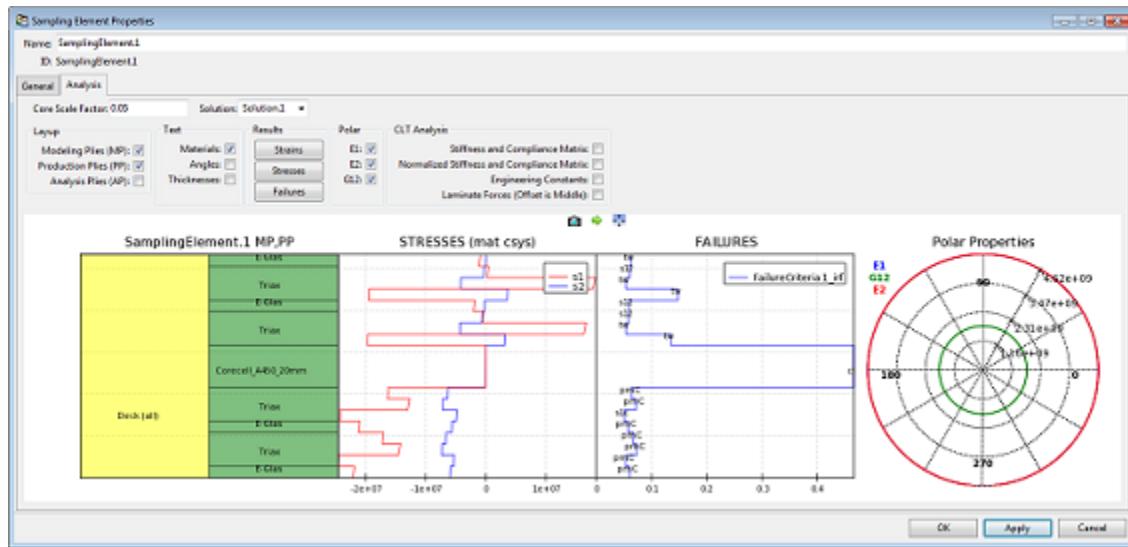
Analysis

The *Analysis* tab provides extended post-processing functionality. First the layup and ply-sequence can be visualized, second based on the classical laminate theory polar properties and laminate stiffnesses can be analyzed and finally the distributions of the post-processing results (strains, stresses and failure criteria) are shown in 2D plots.

Note

Note: The analysis of the Classical Laminate Theory is described in Section [Classical Laminate Theory](#).

Figure 4.67: Layup sequence and enhanced post-processing



Visualization:

The strains and stresses shown in the 2D plot are the values at the element center (interpolated) at the top and bottom of the layer. On the other side the 2D failure plot shows the worst IRF, RF or MoS factor of all failure criteria, failure modes evaluated and integration point level. This can cause that the stresses in the 2D plot are two small if they are compared with the accordant failure plot because the interpolate strain and stress value is smaller or equal than the maximum of the values at the integration point level (graphical inconsistency between the strain/stress and failure plot).

4.1.12. Section Cuts

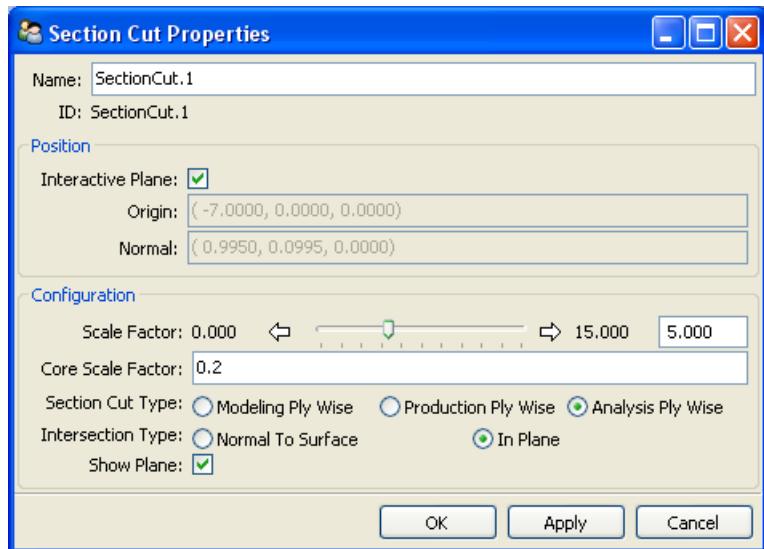
Section Cuts enable a visual verification of the layup definition on an arbitrary section plane through the model.

The section Cut definition contains:

- *Name*: name of the Section Cut.

The section plane is defined with:

- **Interactive Plane**: if the option is active, modify the section plane directly in the Scene. If inactive, the following options are necessary:

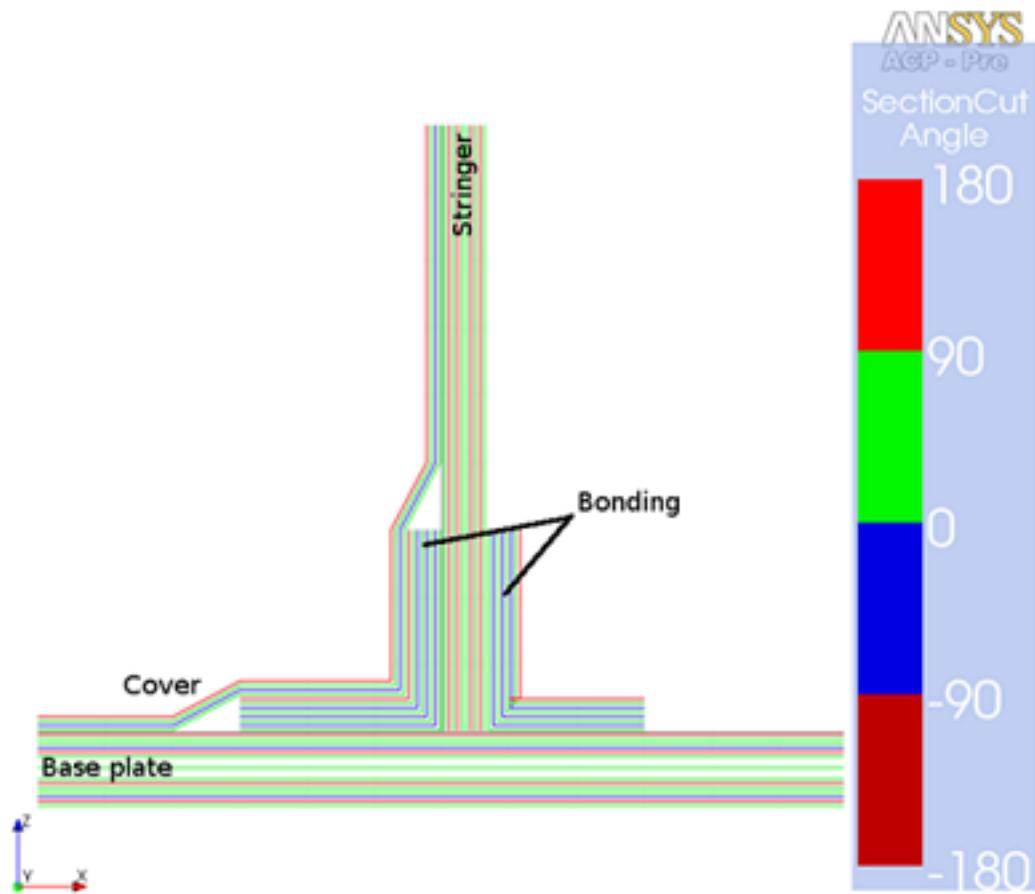
Figure 4.68: Section Cut definition

- *Origin*: origin of the section plane.
- *Normal*: normal direction of the plane.

The plot options are defined in Configuration part:

- **Scale Factor**: scales the offsets of all plies
- **Core Scale Factor**: the thickness of the core plies are scaled by this factor
- **Section cut Type: select which ply type are plotted**
 - *Modeling Ply Wise*: the modeling plies are plotted
 - *Production Ply Wise*: the production plies are plotted
 - *Analysis Ply Wise*: the analysis plies are plotted
- **Interaction Type: define how the intersection of the section plane and the model is defined**
 - *Normal to surface*: the plies are plotted as normal to the intersected elements
 - *In Plane*: the plies are plotted in the section plane
- **Show Plane**: option to plot the section plane or not

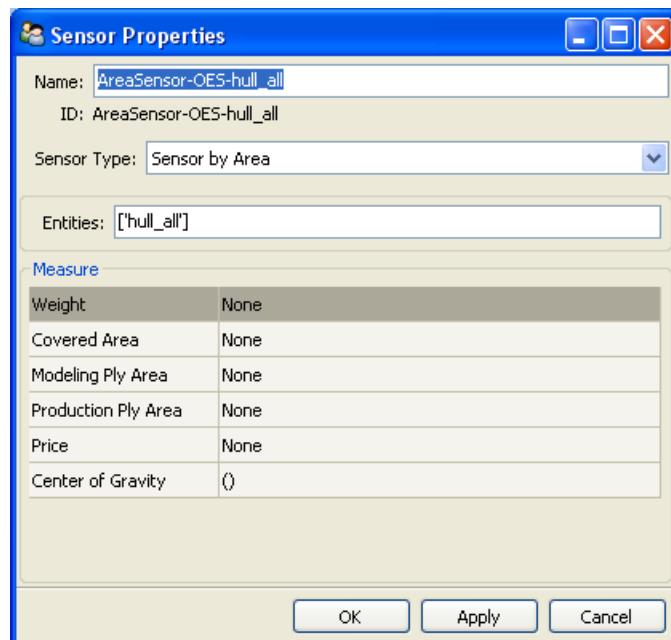
Resulting section cuts are very useful during the [Model Verification](#). The ply angles can be shown on the section with the angle layup plot (see [Layup Plots](#)).



4.1.13. Sensors

A sensor provides the evaluation of global results like price, weight or area. The results can be evaluated for specific parts, materials or plies.

Figure 4.69: Sensor Properties



The Sensor Properties are defined by Name, Sensor Type and Entities. The results are displayed directly in the Properties window.

- **Name:** name of the sensor
- **Sensor Type:** define the evaluation type (see below)
- **Entities:** select the corresponding entities by clicking them in the tree
- **Measure:** displays results of different quantities (see below)

Sensor Type

The sensor type defines what type of entity is quantified. The different sensor types are:

- **Sensor by Area:** select one or several Element Sets or Oriented Element Sets.
- **Sensor by Material:** select Fabric(s), Stackup(s) and / or Sublaminates. If a Fabric is selected, the plies in the sublaminates are also considered in the evaluation; the plies in the stackups not.
- **Sensor by Modeling Ply:** select one or more plies.

Measure

To display the results of the Sensor click the Apply button. The results are displayed in the units defined for the ACP model. The results shown are the following:

- **Weight:** the mass of the selected entity.
- **Covered Area:** the surface area of a selected Element Set / Oriented Element Set or the tooling surface area that is covered by the composite layup of the selected Material or Modeling Ply.
- **Modeling Ply Area:** the surface area of all modeling plies of the selected entity.
- **Production Ply Area:** the surface area of all production plies of the selected entity.
- **Price:** the price for the composite layup of the selected entity. The price per area is set under the feature Material Data > Fabrics or Stackups.
- **Center of Gravity:** the center of gravity of the selected entity in the global coordinate system.

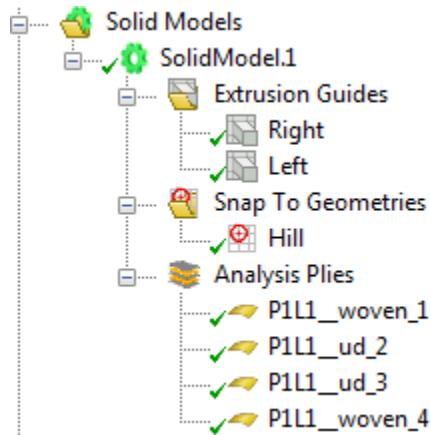
4.1.14. Solid Models

The Solid Model feature creates a layered solid element model from a composite shell model. The solid element model can be integrated into a Workbench workflow or exported for use outside of Workbench. The section [Analysis of a Composite Solid Model](#) explains the Solid Model workflow in Workbench where as the section [Guide to Solid Modeling](#) provides general information on Solid Modeling.

The settings for the solid model generation are adjusted under the Solid Model context menu item **Properties**. It covers what element sets are extruded, the extrusion method, drop-off elements handling and numbering offsets among other things.

Extrusion Guides and Snap To Geometries can be used to shape the Solid Model in a desired way. They are specified in the respective subfolders in the Solid Model tree view. The subfolder Analysis Plies shows which analysis plies are incorporated in the Solid Model.

Figure 4.70: Solid Model feature in the GUI tree view



4.1.14.1. Solid Model Properties - General

This section is divided into the following parts:

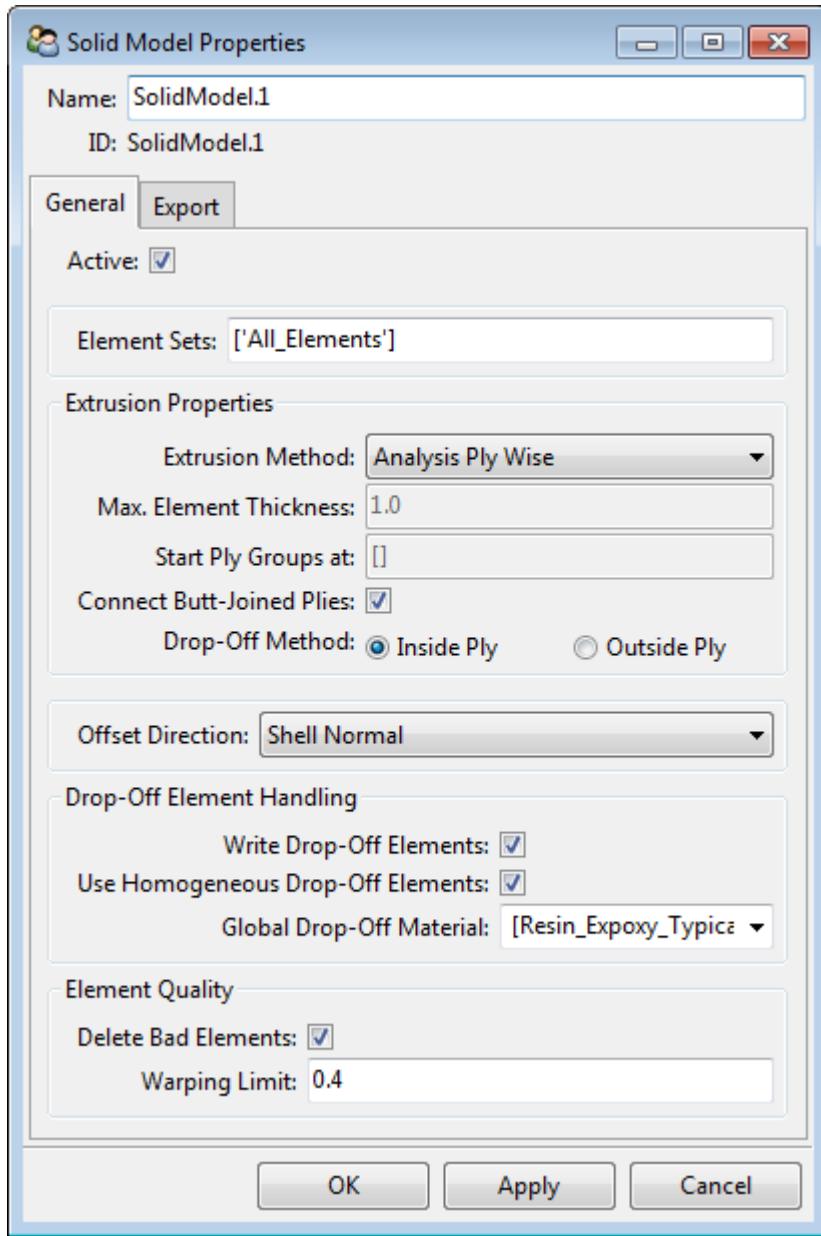
- 4.1.14.1.1. Element Sets
- 4.1.14.1.2. Extrusion Method
- 4.1.14.1.3. Connect Butt-Jointed Plies
- 4.1.14.1.4. Drop-Off Method
- 4.1.14.1.5. Offset Direction
- 4.1.14.1.6. Drop-Off Element Handling
- 4.1.14.1.7. Element Quality

4.1.14.1.1. Element Sets

Starting with a shell model and the lay-up definition, the shell elements are extruded to a layered solid element model. Select an *Element Set* to define the region of the extrusion.

Important

The mid-offset option of Element Sets is not supported for Solid Model extrusion. The ply definition must be defined without this option to obtain the correct solid model position.



4.1.14.1.2. Extrusion Method

The lay-up extrusion can be organized in different ways to merge plies with different criteria:

- *Analysis Ply Wise*: each Analysis Ply is extruded as one solid element layer.
- *Material Wise*: all sequential plies containing the same material are grouped in one solid element layer. A maximum element thickness can be specified that will subdivide the single element layers if necessary.
- *Modeling Ply Wise*: each Modeling Ply is extruded as one solid element layer, i.e. every stackup or sub-laminate is extruded as one solid element layer.
- *Monolithic*: the whole lay-up is extruded in one solid element layer.
- *Production Ply Wise*: each Production Ply is extruded as one solid element layer if possible. Depending on the model topology monolithic ply groups may be split in order to ensure the cohesion of the resulting solid model.

- *Specify Thickness*: plies are grouped by iterating through the laminate from the inside out. A new ply group is introduced if the thickness of the preceding group reaches the specified Max. Element Thickness value. If a single ply (e.g. a sandwich core) is thicker than Max. Element Thickness it will be split to equally thick layers no thicker than the Max. Element Thickness.
- *User Defined*: plies are grouped by iterating through the laminate from the inside out. A new ply group is introduced each time the iteration meets one of the plies specified in *Start Ply Groups at*.
- *Sandwich Wise*: plies either side of a core material are grouped into single element layers. The core material is extruded as one element layer. A maximum element thickness can be specified that will subdivide the single element layers further if necessary.

4.1.14.1.3. Connect Butt-Jointed Plies

If a composite layer ends away from a mesh boundary it tails off with a drop-off element. These drop-off elements are degenerated brick elements that are reduced from bricks into prisms.

The "Connect Butt-Jointed Plies" option can prevent an element drop-off of two adjacent, sequential plies in the same modeling ply group. The option is activated by default.

Currently, the feature is limited to plies that appear sequentially in the same modeling ply group. As a result, it is not possible to connect all butt-jointed plies that are arranged in a circle. This is a known limitation.

The example of a sandwich structure with divided core material is shown below to exemplify the use of a ply connection.

Figure 4.71: "Connect Butt-Jointed Plies" option activated

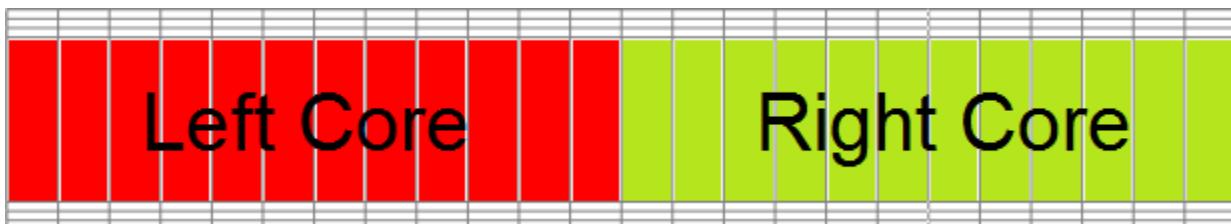
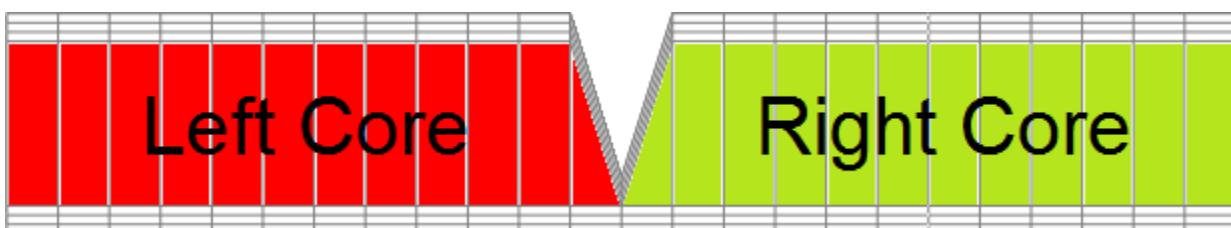
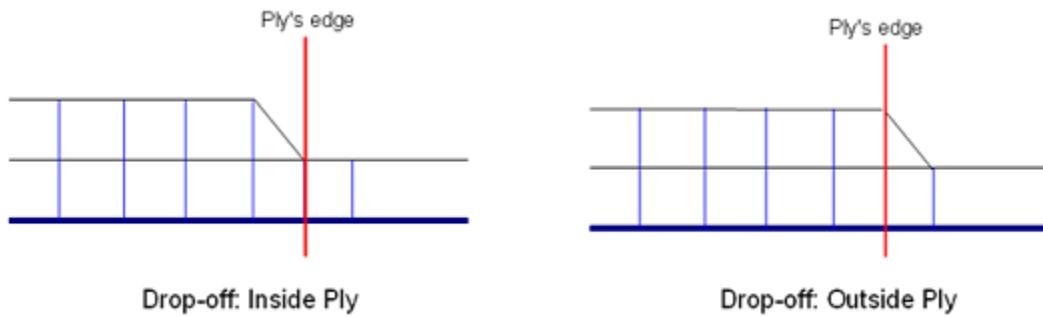


Figure 4.72: "Connect Butt-Jointed Plies" option deactivated



4.1.14.1.4. Drop-Off Method

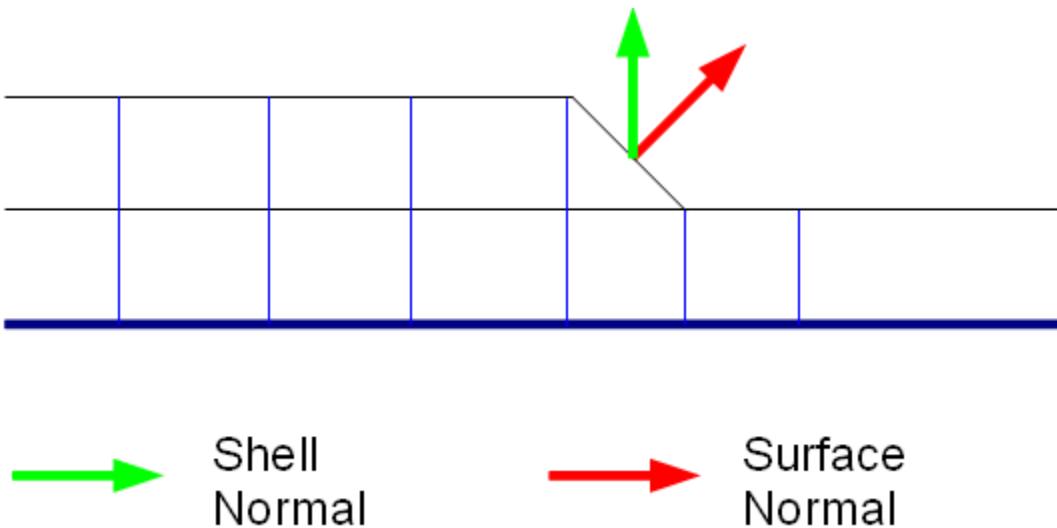
At the end of a ply, the Ply's **Drop-Off** can be generated before or after the ply's edge.



4.1.14.1.5. Offset Direction

The extrusion direction has two modes: Shell and Surface Normal. With the Surface normal, the extrusion direction is re-evaluated after each row of solid elements. With the Shell normal, the extrusion direction stays defined as shell normal.

Figure 4.73: Extrusion direction



An example with both *Offset Directions* is shown below:

Figure 4.74: Solid model with Surface Normal direction

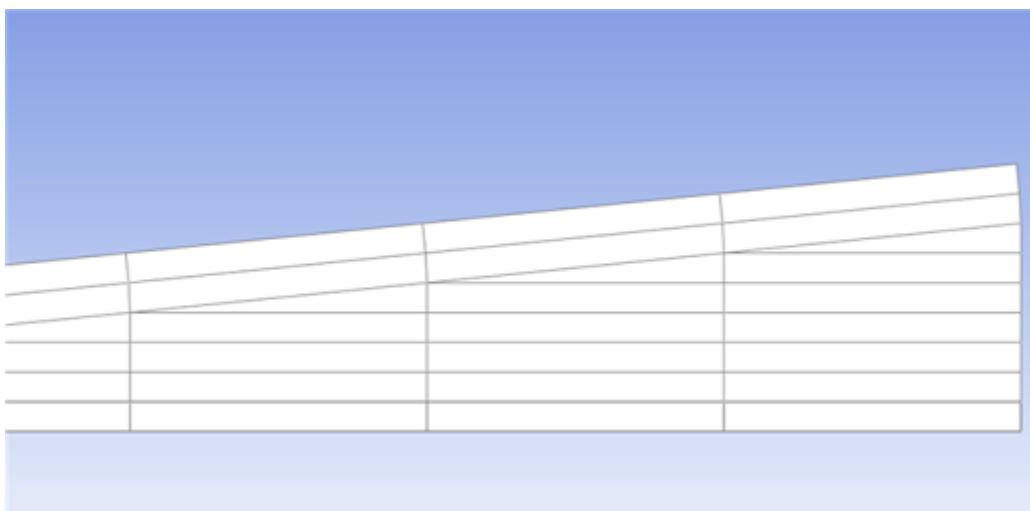
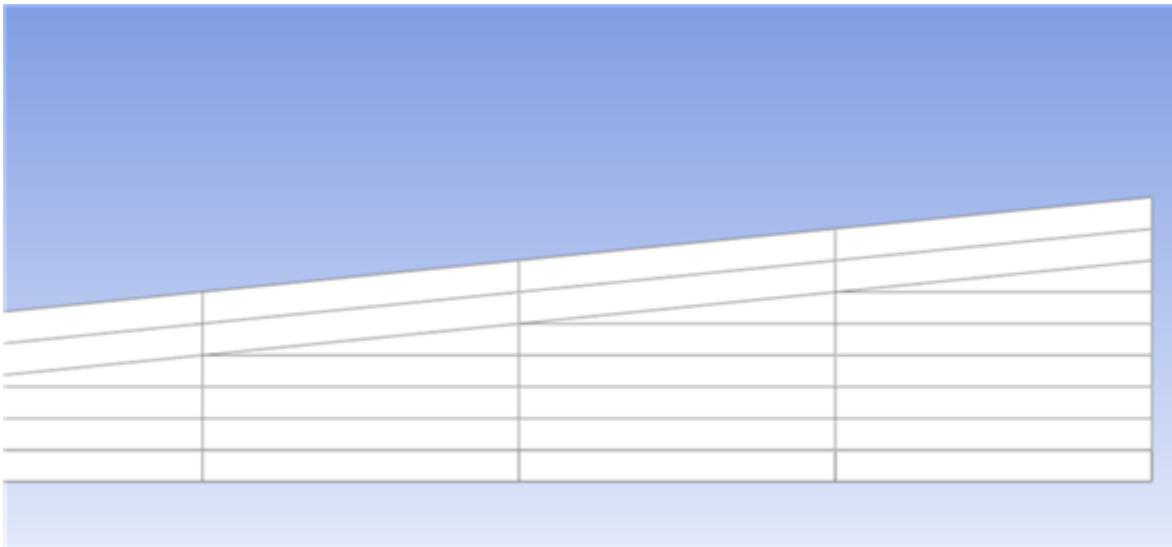


Figure 4.75: Solid model with Shell Normal direction



4.1.14.1.6. Drop-Off Element Handling

Write Drop-Off Elements

The Write Drop-Off Elements option has to be checked to include drop-off elements in the solid model export as exemplified below. If it is not activated the space between the two layers remains empty (void).

Figure 4.76: Export with drop-off elements

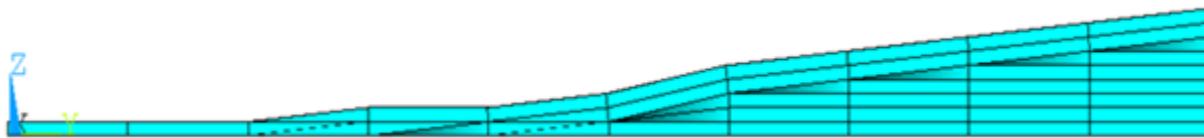


Figure 4.77: Export without drop-off elements



Use Homogeneous Drop-Off Elements

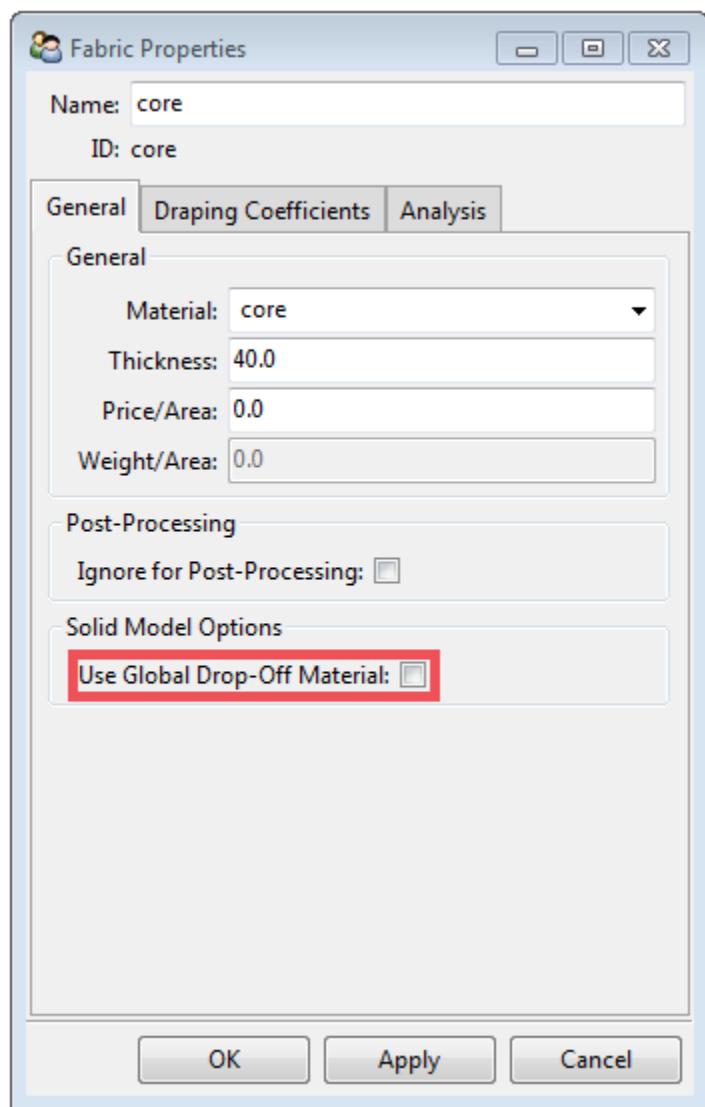
Using homogeneous drop-off elements replaces the layered drop-off elements with homogeneous non-layered solid elements. Layered drop-off elements are degenerated brick elements and cause problems for numerical solvers. Therefore the use of homogeneous drop-off elements is recommended, especially in the Workbench workflow. A global drop-off material must be selected in combination with this option.

Global Drop-Off Material

The global drop-off material is only used for the homogeneous non-layered solids elements. It is recommended to use an isotropic material such as a resin.

In the case of sandwich structures, it is useful to model the drop-element with the core material as opposed to a resin. To override the use of a global homogeneous drop-off material it is necessary to unselect the use of global drop-off material for the core material (Fabric) as shown below. As degenerated elements cannot have layered properties the drop-off zone must consist of only one material type for this function to work properly. If the Extrusion Method is set to **Analysis Ply Wise** it is guaranteed to work. Other extrusion methods such as **User Defined** or **Specific Thickness** can work as well yet require the correct configuration.

Figure 4.78: Disabling the use of global drop-off material option for a core material



4.1.14.1.7. Element Quality

ACP performs a shape check during the solid model generation. The checks are similar to the ANSYS Element Shape Testing (see [Element Shape Testing](#) in the Mechanical APDL Theory Reference for more information). The Solid Model feature has the option to delete elements if they violate the shape checking. Warping is not the only element shape check that is carried out but the warping factor can be adjusted.

4.1.14.2. Solid Model Properties - Export

This section is divided into the following parts.

- 4.1.14.2.1. Use Solsh Elements
- 4.1.14.2.2. Use Solid Model Prefix
- 4.1.14.2.3. Transferred Sets
- 4.1.14.2.4. Numbering Offset

4.1.14.2.1. Use Solsh Elements

Activate this option to create the model with the SolidShell elements SOLSH190. Refer to *ANSYS Help* for more information about this element and its properties.

4.1.14.2.2. Use Solid Model Prefix

Element and edge sets in ACP relate to named selections in Workbench Mechanical and components in the *.cdb file or in Mechanical APDL. The Solid Model Prefix option sets the name of the element components to begin with the name of the defined solid model. For example, if the name of the solid model is *BULKHEAD*, the elements will be grouped into components like below:

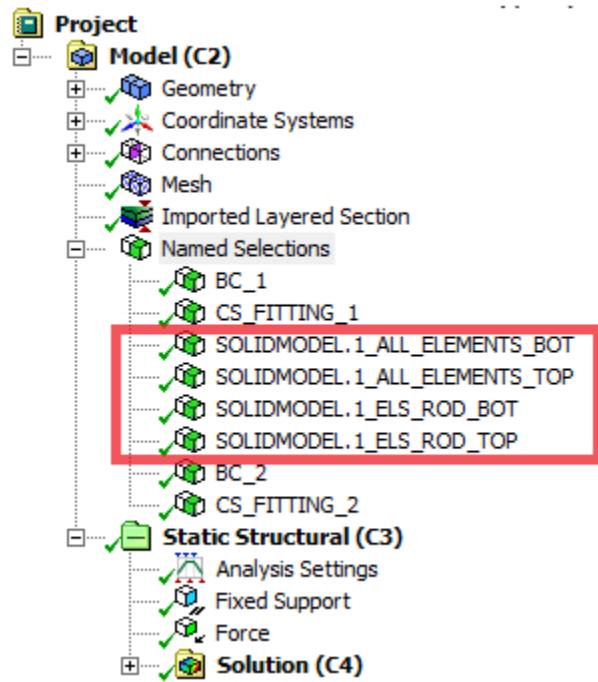
```
CMBLOCK,BULKHEAD_P9L1_Plies_Top,ELEM, 600! users element component definition
```

4.1.14.2.3. Transferred Sets

It can be specified which element sets or edge sets are transferred as element components in the workflow. Both sets are altered as a result of the solid model extrusion.

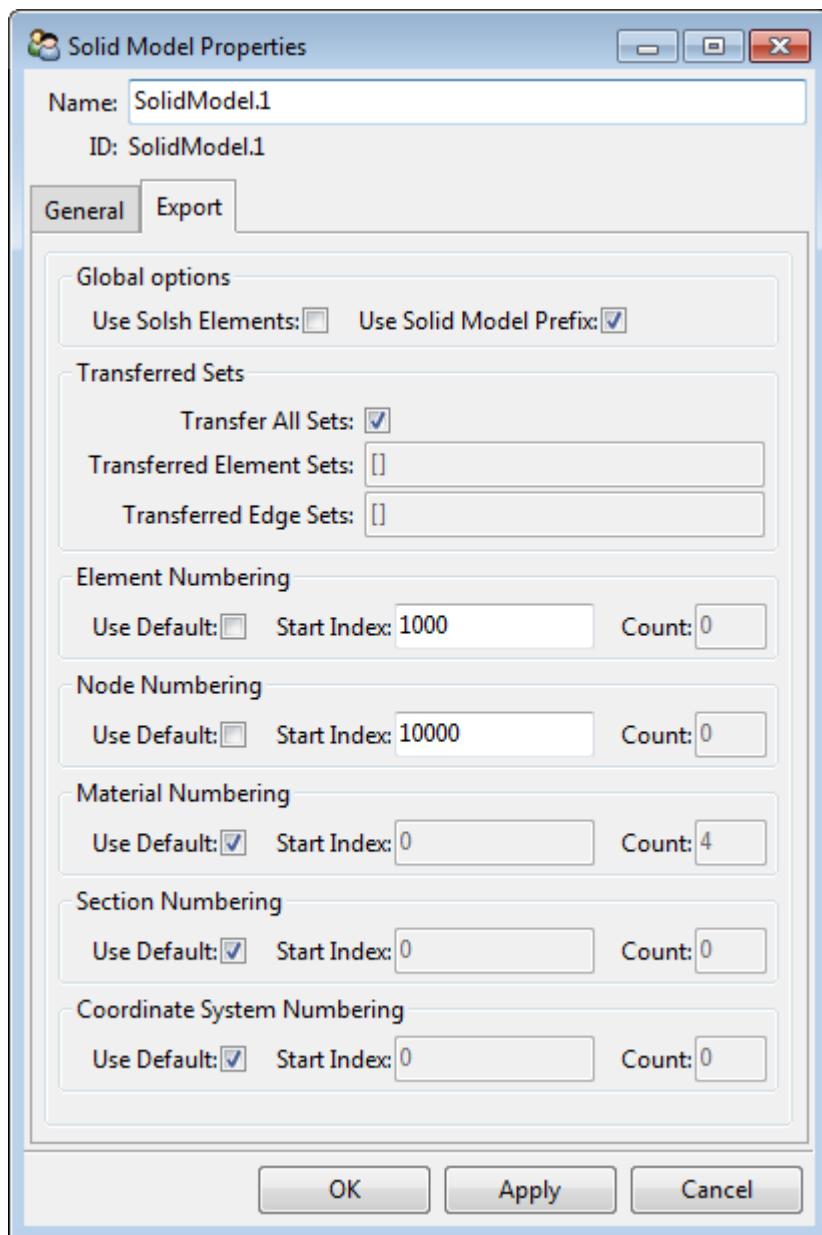
An element set turns into two separate element components with the designation “_TOP” and “_BOT”. One element component coincides with the original element set and the other one lies at the end of the extrusion path.

An edge set is a collection of lines and as part of the solid model extrusion they are extruded to form surfaces. Thus an edge set transfers into a surface element component. The example below shows transferred element sets.

Figure 4.79: Transferred element sets in Workbench Mechanical

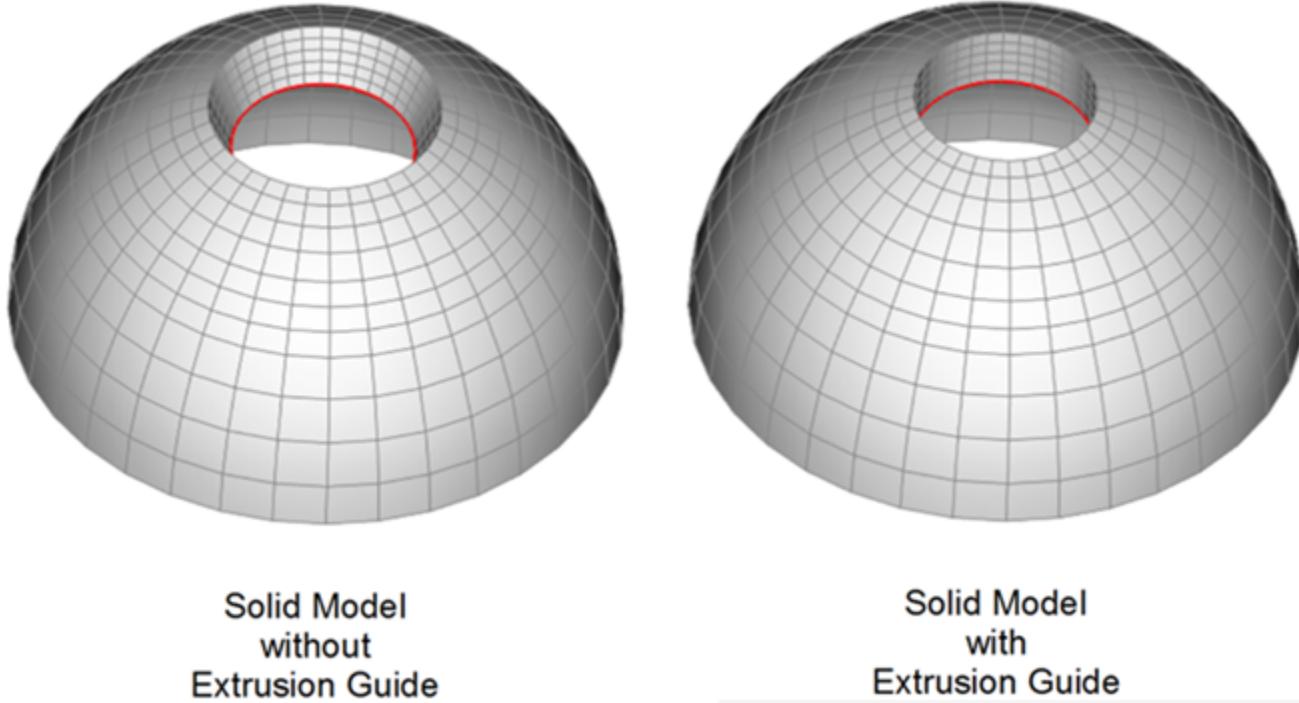
4.1.14.2.4. Numbering Offset

If *Solid Models* from different ACP or Mechanical models are combined in one Analysis System the global numbering has to be unique otherwise the Model import fails ("Failed due to duplicate Nodes/Elements"). If all *Solid Models* are created in the same ACP model the numbering is evaluated automatically. In all other cases, the NUMOFF (offset number) for each entity (materials, section, nodes, elements and coordinate systems) can be defined in the Numbering sections in the *Solid Model Properties*. Note that the numbers need not be continuous. Deactivate the *Default* option to define the offset manually (numbering start at this index and will be incremented).



4.1.14.3. Extrusion Guides

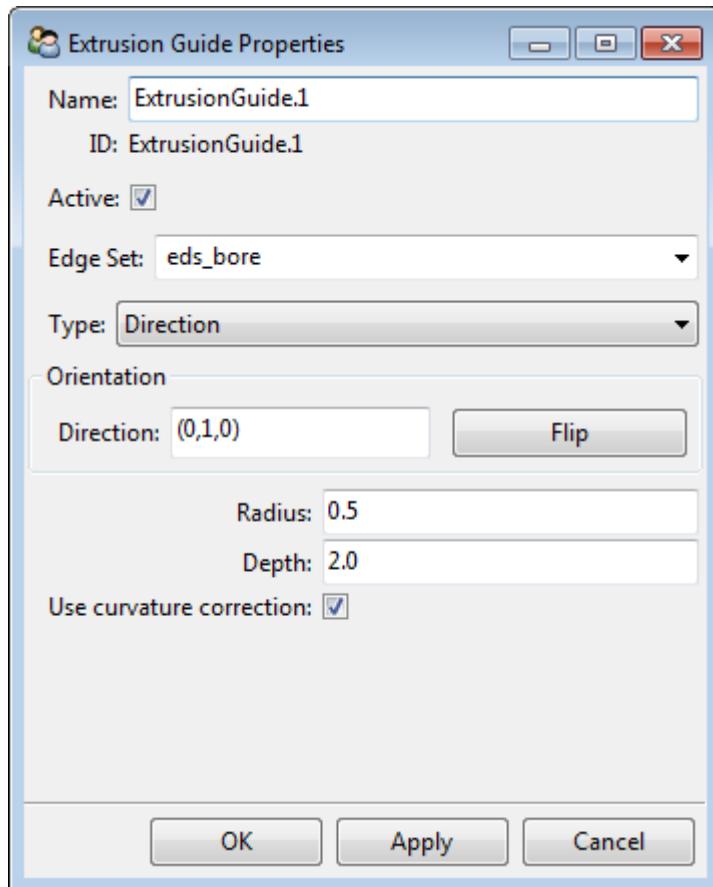
The solid model generation of curved geometries and thick layups can lead to boundary edges being extruded in undesirable directions. The extrusion of a dome with a hole at the top results in a solid model with a hole that is not cylindrical for example. The Extrusion Guide features allow the user to control the extrusion direction of the edges in order to rectify this. The edge of the hole can be used as an Extrusion Guide in the vertical direction to create a cylindrical hole.

Figure 4.80: Extrusion without and with an Edge Set Guide

Multiple Extrusion Guides can be used for one Solid Model. The extrusion itself is controlled with an edge set and a direction vector or with a geometry. The Extrusion Guide feature also controls the Curvature Control. It has the following properties:

- *Edge Set*: the edge set along which the Extrusion Guide acts
- *Type*: 3 different type of Extrusion Guides
 - *Direction*: a direction vector defines the extrusion direction of the edge set.
 - *Orientation Direction*: By default, the normal direction of the edge set is calculated and defined when an edge set is selected. It can also be entered manually.
 - *Geometry*: a CAD file of a boundary surface is used to define the extrusion path.
 - *CAD Geometry*: selection of a previously imported CAD geometry
 - *Free*: no extrusion path is defined but the Curvature Correction can be activated independently.
- *Radius*: controls the sphere of influence of the mesh morphing. More information below.
- *Depth*: controls the bias of the mesh morphing. More information below.

Figure 4.81: Properties for a direction Extrusion Guide



4.1.14.3.1. Mesh Morphing

The explanation of the mesh morphing requires brief recapitulation of the way the solid model generation and extrusion guide works.

The generation of a solid model is the extrusion of a shell mesh. The extrusion is in the direction of the shell normal by default. The 2D shell mesh is used as a base for the 3D solid element mesh which can have one or more element layers depending on the ply thickness and extrusion method.

The extrusion guide only affects the extrusion of the element edges that are part of the guided edge set. It is either defined by an edge set and direction vector or by an edge set and a CAD geometry. While the CAD geometry already is a surface, the edge set and direction vector are used to define a surface. In both cases, these surfaces serve as target surfaces in the extrusion.

The guided edge is initially extruded in the normal direction and then the nodes on the resulting free surface are moved to coincide with the target surface of the extrusion guide.

The mesh morphing is a way to control the propagation of the extrusion guide effect through the entire mesh.

The mesh morphing is governed by the Morphing Law shown in Equation [create link to equation below]. It relates the displacement of internal nodes to the displacement of a node of the guided free surface. It can be controlled with the two parameters:

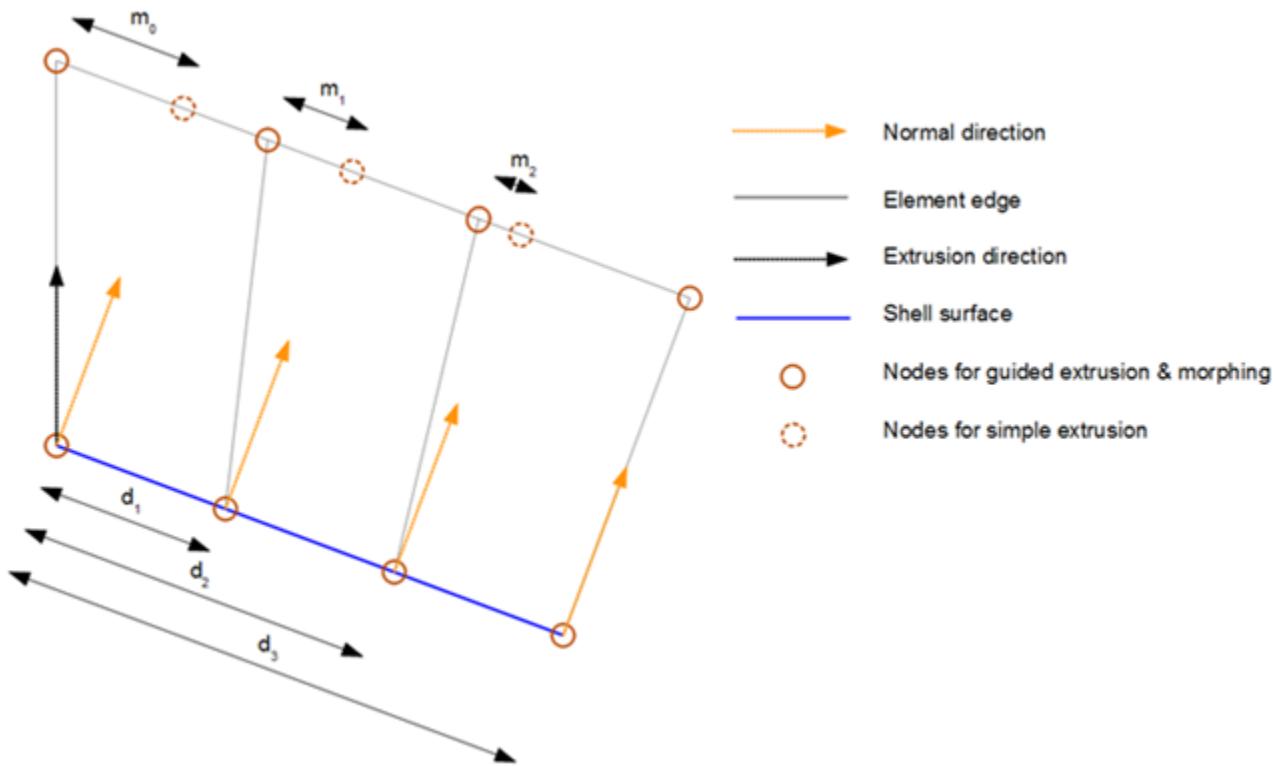
- *Radius*: all elements within the defined Radius from the Edge Set will be extruded with a mesh morphing correction.
- *Depth*: this parameter defines the bias of the mesh morphing (linear with 1, quadratic with 2,...).

$$m_i = m_0 \cdot \left[1 - \left(\frac{d_i}{\text{Radius}} \right)^{\text{Depth}} \right]$$

The other parameters entering the morphing law (m_0 , m_i and d_i) are shown in the figure below and have the following meaning:

- m_0 : the distance a node on the free surface has to move in-plane to coincide with the extrusion guide.
- m_i : the distance the inward node of the i^{th} shell element moves in-plane as a result of the mesh morphing
- d_i : the distance between the node in the guided edge set and the inward node of the i^{th} shell element.

Figure 4.82: Mesh morphing diagram



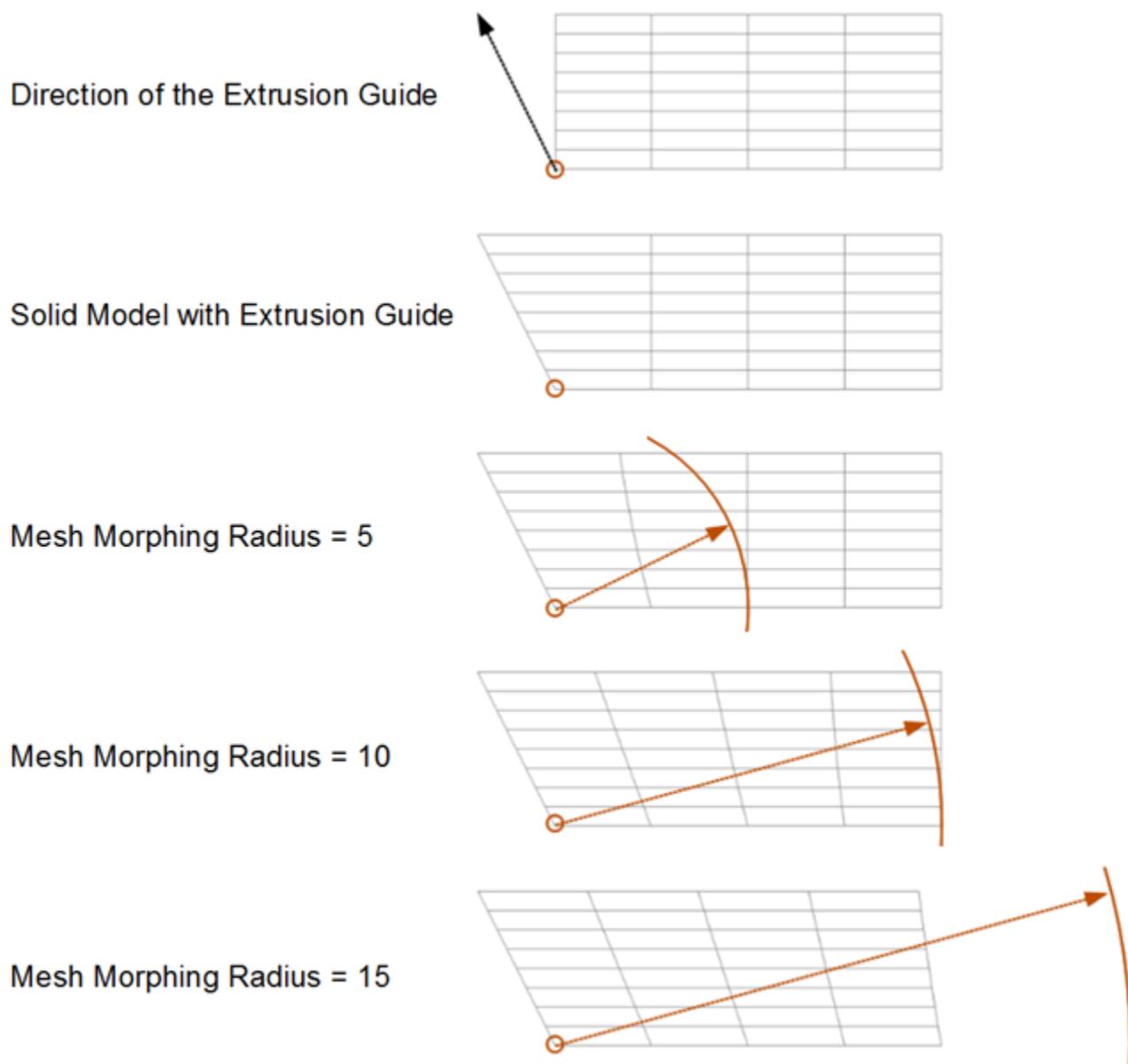
4.1.14.3.2. Curvature Control

A curvature correction can be applied during the solid model extrusion which results in a smoother extruded surface. In previous ACP versions, the curvature correction was automatically applied. It has become an optional feature. Under certain circumstances, a deactivated curvature correction can lead to better extrusion results.

4.1.14.3.3. Extrusion Guide Examples

Example of a direction-type Extrusion guide with different mesh morphing radii. The location of the edge set is indicated by the circle in the bottom left corner. The mesh morphing is only applied to nodes on the shell surface whose distance to the edge set is smaller than the radius.

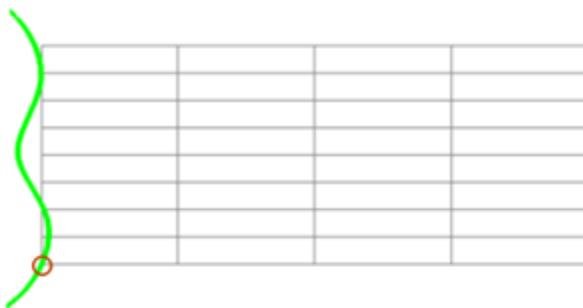
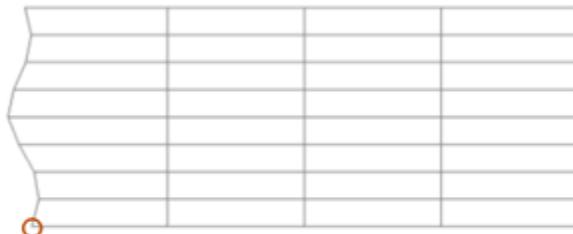
Figure 4.83: Example of a direction-type Extrusion guide with different mesh morphing radii. The location of the edge set is indicated by the circle in the bottom left corner.



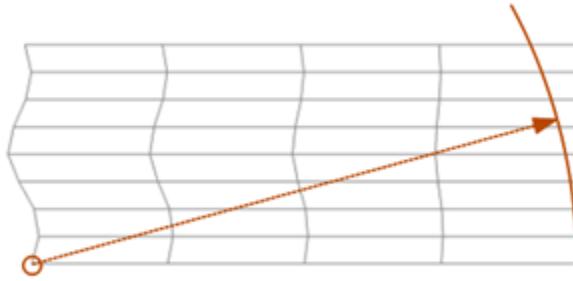
Example of a geometry-type Extrusion guide with different mesh morphing depths.

Figure 4.84: Example of a geometry-type Extrusion guide with different mesh morphing depths.

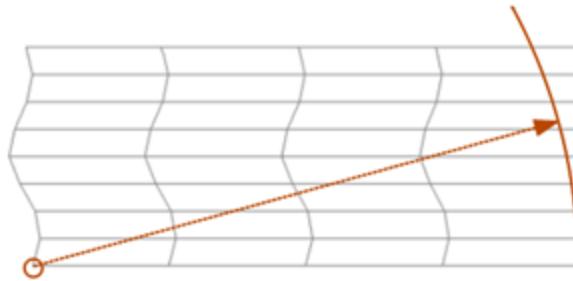
CAD Geometry Contour

Solid Model with Extrusion Guide,
no Mesh Morphing

Mesh Morphing: Radius = 10, Depth = 1



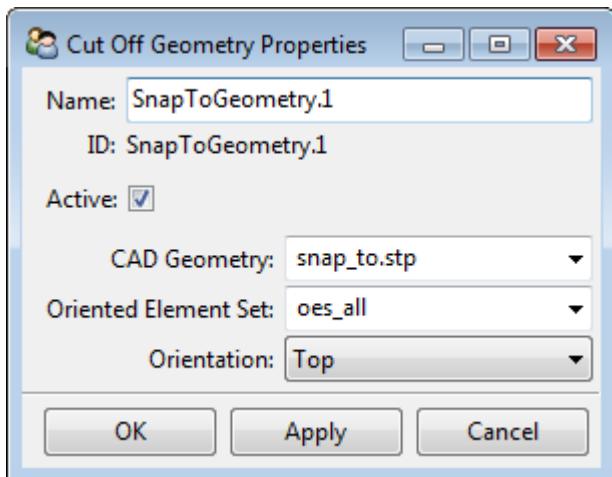
Mesh Morphing: Radius = 10, Depth = 10



4.1.14.4. Snap to Geometry

The Snap to Geometry feature can alter an extruded solid model to align with an imported CAD geometry. The layered solid model is locally stretched or compressed so that its selected faces coincide with the CAD geometry. Multiple Snap To Geometries can be assigned to one Solid Model. The adaption will occur at the first intersection that is found.

The feature is only applied to the selected *Oriented Element Set* and its selected face (top or bottom). Which face is top or bottom is defined by the normal orientation of the *Oriented Element Set*. The height of all the elements through the thickness is altered to an even distribution.



In the example below, the first picture shows the extrusion without any *Snap to* operation. Note that the layup is defined from two *Oriented Element Sets* which point in opposite directions. In the second picture, the first modeling ply (oriented to the top) is defined to be extruded to a *CAD Geometry*. Notice that only the nodes which meet the surface are extruded until the surface. The other nodes are extruded normally. The second modeling ply is extruded to another *CAD Geometry* in the last picture. Note that, for both cases, the orientation of the snap operation must be *top* as the *Oriented Element Sets* both point outward.

Figure 4.85: Extrusion without snap operation

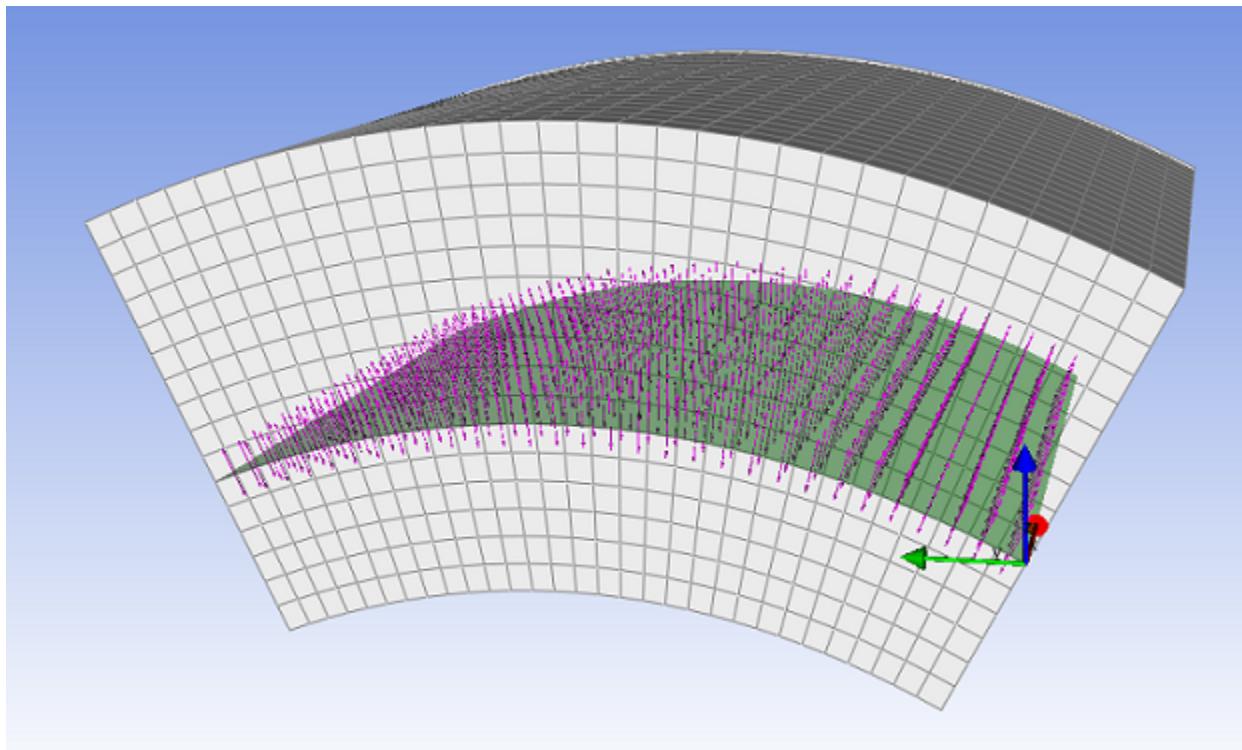


Figure 4.86: Extrusion with snap to geometry at the top (shell geometry also displayed)

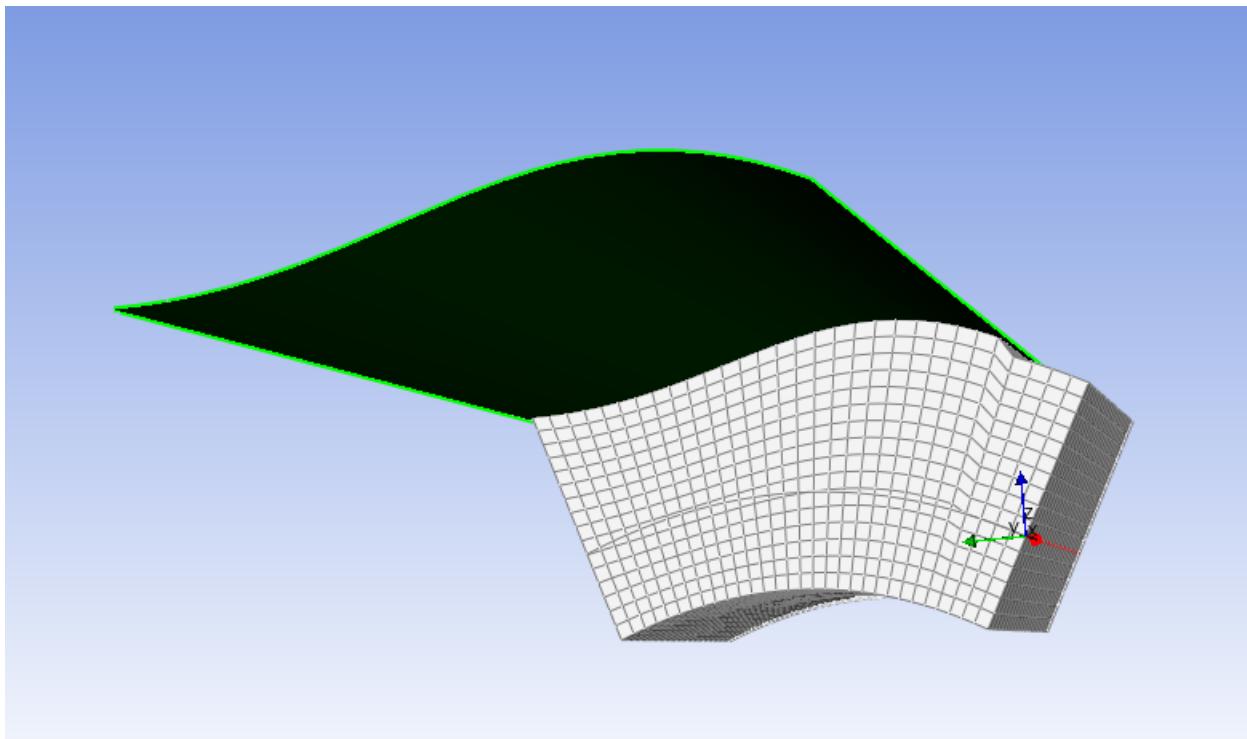
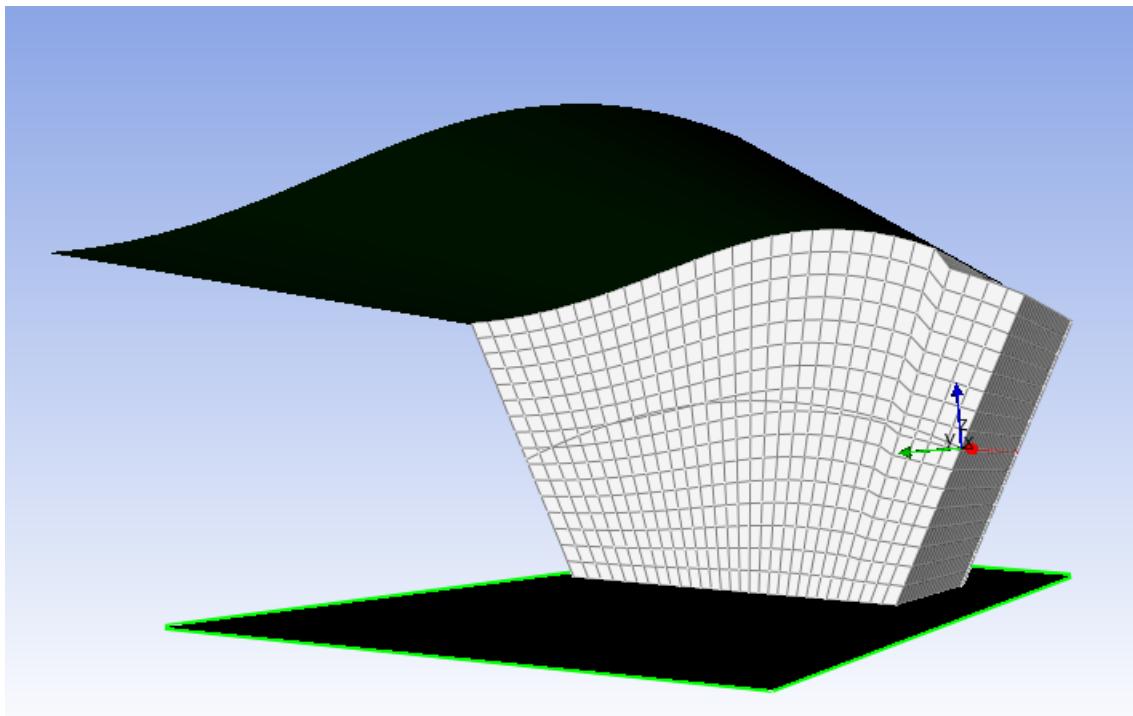


Figure 4.87: Extrusion with snap to geometry at the top and bottom (shell geometry also displayed)

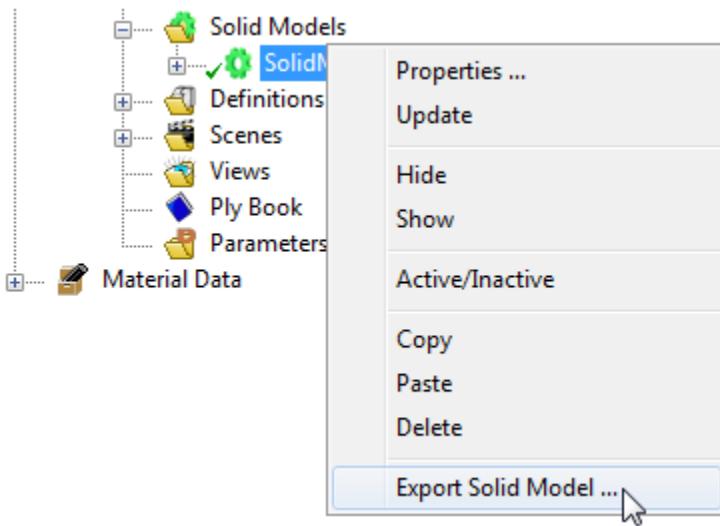


Warning

The *Snap to Geometry* operation occurs after the *Extrusion Guides* operations. So it is possible that the nodes moved during the *Extrusion Guides* operations will be translated again, and do not match with the previous *Extrusion Guides* definition.

4.1.14.5. Export Solid Model...

The context menu of the *Solid Model* allows to export the model as *.cdb file. Element shape checking is deactivated in the exported *.cdb file.

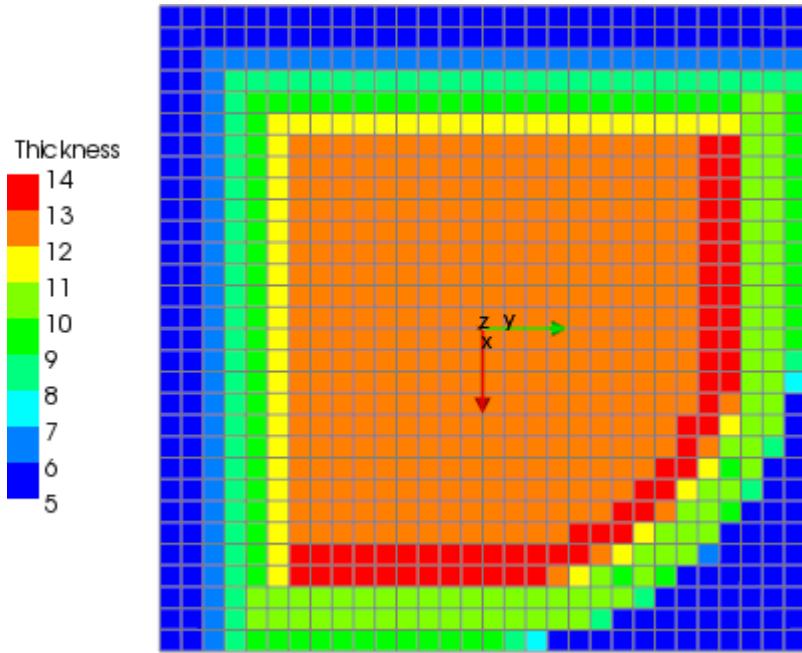


4.1.14.6. Save & reload Solid Models

The ACP solid models are saved as *.h5 files. An Update of ACP (Pre) component initiates a check of the current and previous laminate layup. While the laminate layup does not change no new solid model is generated.

4.1.15. Layup Plots

Layup Plots controls the thickness and ply angle plots in ACP. The thickness plot shows the thickness distribution for an entire layup or single plies. The angle plot is purely a ply-wise plot and shows the orientation angle of a selected ply. Both plots can display the information for all or a selection of elements through the data scope. A thickness and angle plot for all element sets is predefined by default. The settings both plots are similar. The plot definition for layup plots follows the same definition as for solution plots.

Figure 4.88: Example of thickness plot (tutorial 2)

Layup Plot Context Menu

The context menu of Layup Plots has these options:

- **Create Thickness ...** create a thickness plot.
- **Create Design Angle ...** create an angle plot.
- **Paste** paste a layup plot that was previously copied.

Plot Context Menu

The context menu of a layup plot has these options:

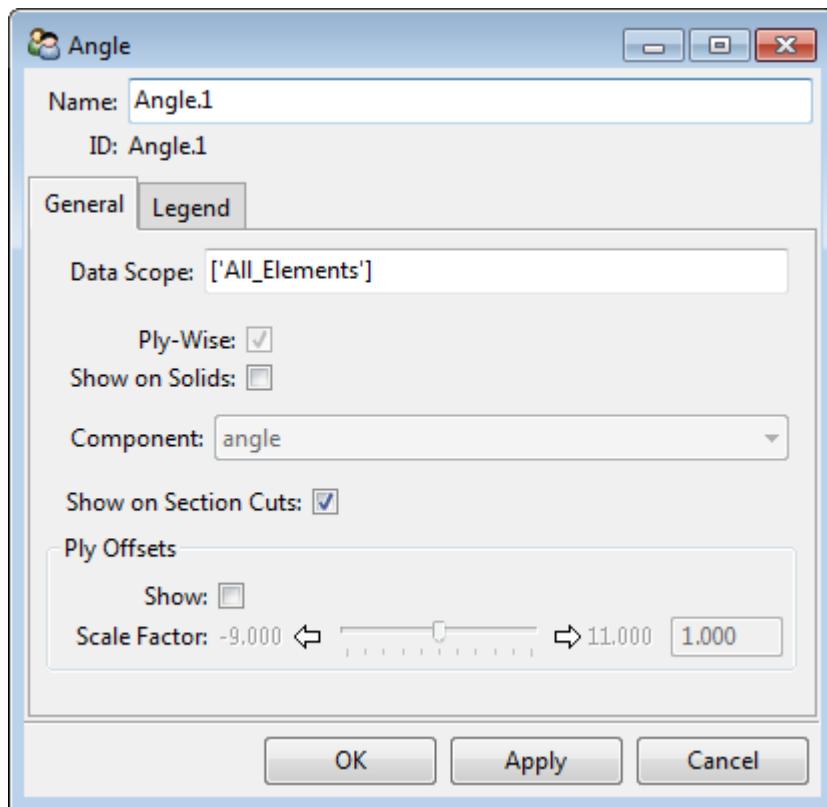
- **Properties ...** view and edit the plot properties.
- **Update** update the selected plot.
- **Copy** copy the plot definition.
- **Paste** paste a layup plot that was previously copied.
- **Delete** delete the selected plot.
- **Hide** hide the selected plot.
- **Show** show the selected plot.

Plot Properties - General

The basic plot settings are configured in the General tab of the properties window. The settings are similar to those for the solution plots:

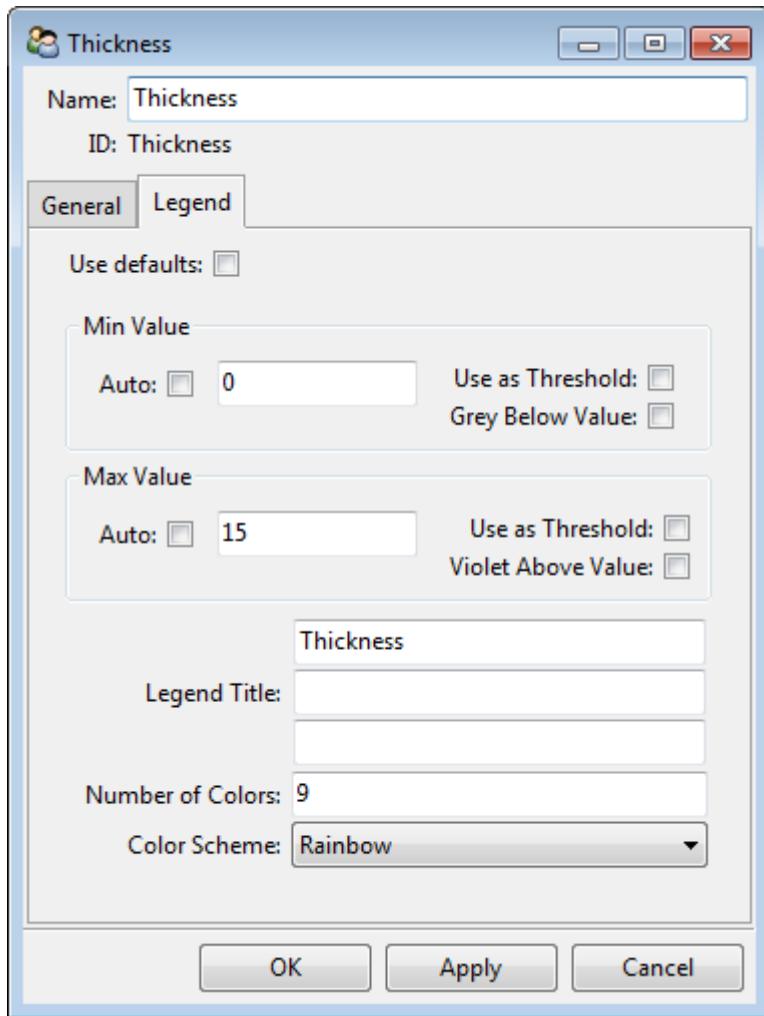
- **Name** set the name of the plot.
- **Data Scope** determines what scope is used in the plot. Element sets, OES, Modeling Plies & Sampling Elements can be selected in the data scope. The data scope of a sampling element covers all plies that are intersected by the sampling element.
- **Ply-wise** activates a ply-wise plot display. Thickness and angle plot are only shown if a ply is selected. Angle plots are automatically set to ply-wise. The plies can be selected from the modeling ply groups, sampling elements or solid model analysis plies.
- **Show on solids** shows plot data on solid elements and for solid elements only.
- **Component** indicates whether the plot is a thickness or angle plot.
- **Show on Section Cuts** additionally shows the ply angle on section cuts in the same color scale (angle plot only).
- **Ply Offsets** visualize the results of a ply-wise plot on the selected plies at their true or scaled offset from the reference surface (angle plot only).

Figure 4.89: Angle plot properties - General tab



Plot Properties - Legend

The legend is configured by standard settings as a default but titles, labels and legend ranges can be customized.

Figure 4.90: Thickness plot properties - Legend tab

4.1.16. Definitions

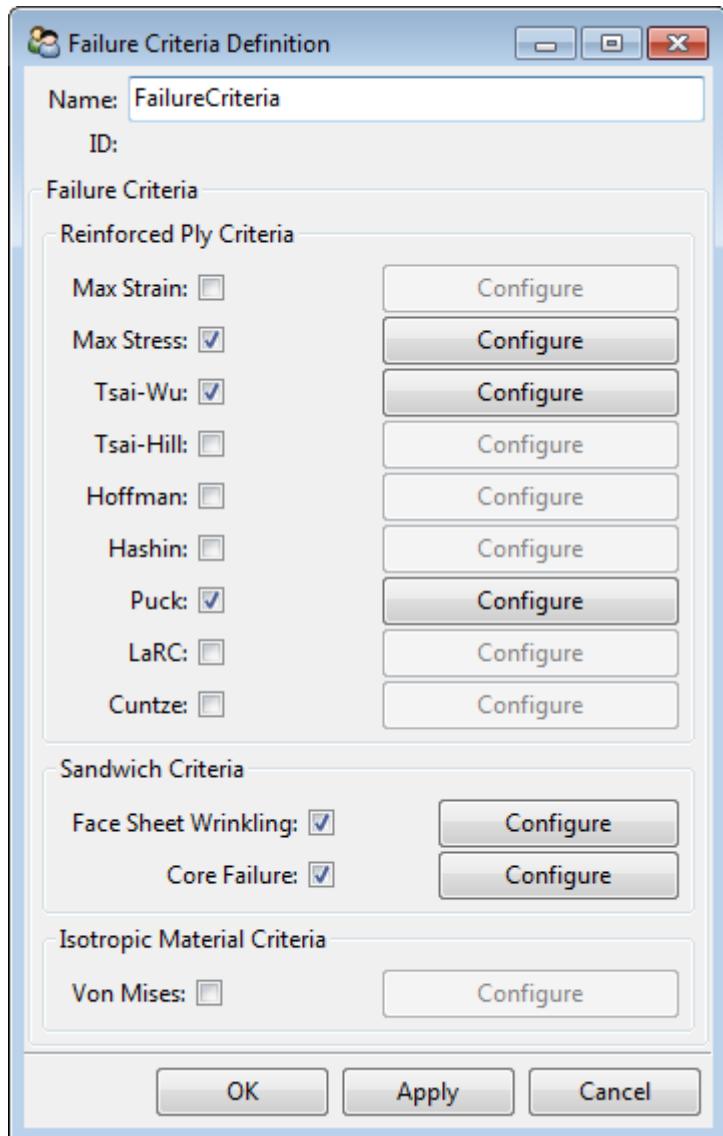
Failure criteria are used to evaluate the strength of a composite structure. Several failure criteria can be defined, combined and configured in the Definitions object. The failure criteria definitions can be used for failure plots and sampling elements. The critical failure mode for an element shown in failure plots and sampling elements is always the one with the lowest reserve factor.

A list of the implemented failure criteria and their failure types is shown in the section [Postprocessing](#). More detailed information about the failure criteria is provided in the theory section [Failure Analysis](#).

New failure criteria definitions can be created by selecting **Create Failure Criteria ...** in the context menu of the Definitions object. The failure criteria definition is subsequently configured in the properties window.

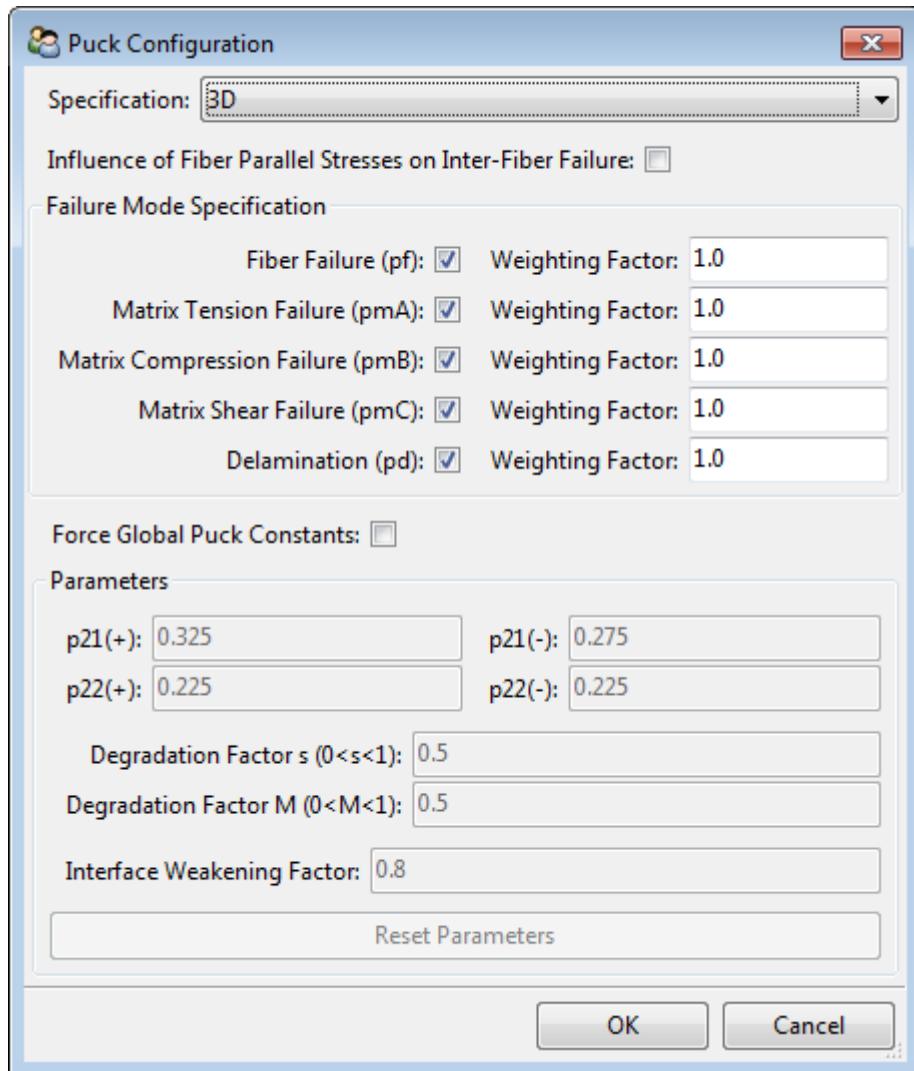
Failure Criteria definition

Each failure criteria definition can be a selection of Reinforced Ply, Sandwich and Isotropic Criteria. The different failure modes are activated via the check-boxes and can be set up in the failure criteria configuration.

Figure 4.91: Failure Criteria Definition

Failure Criteria Configuration

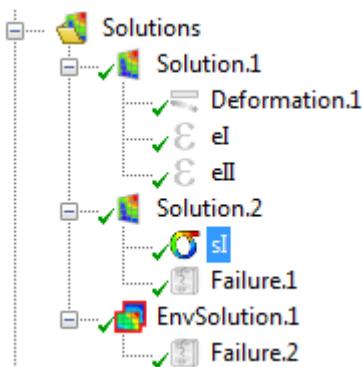
Individual failure modes for each failure criteria can be activated and be associated with a weighting factor. The weighting factors can be used to define different factors of safety for certain failure criteria or specific failure modes. Some criteria also have different levels of complexity. The implemented Puck criterion can be used in its simplified, 2D or 3D option for example.

Figure 4.92: Puck failure criteria configuration options

4.1.17. Solutions

The Solutions object is only available in ACP (Post) mode. The individual solutions under the Solutions objects are used to import and read the solution results into ACP. All postprocessing plots (i.e. deformation, failure, stress, strain & temperature plots) are linked to individual solutions. Several solutions can be combined into one envelope solution to visualize an overlay of failure results. A solution corresponds to a load step in Mechanical APDL and the load step of interest can be selected if the result file contains several load steps. The solution has to be up-to-date and the results need to match the mesh before any post processing can be done.

Figure 4.93: Solutions object in the tree view



The details on how to configure the results import, create an envelope solution and plot results is explained in the following sections:

- 4.1.17.1. [Solution](#)
- 4.1.17.2. [Envelope Solution](#)
- 4.1.17.3. [Solution Plots](#)

4.1.17.1. **Solution**

A solution can be added by selecting **Import Results ...** in the context menu of the Solutions object and this creates a solution and opens the Solution Properties window.

In the Workbench mode, a solution is set up in ACP for every solution component that is linked to the ACP (Post) cell in the project schematic. By default, the solution points to the last load step in the *.rst file. The selection of a different load step and the import settings can be configured in the Solution Properties. They can accessed via the context of the particular solution.

The context menu of every solution has the following options:

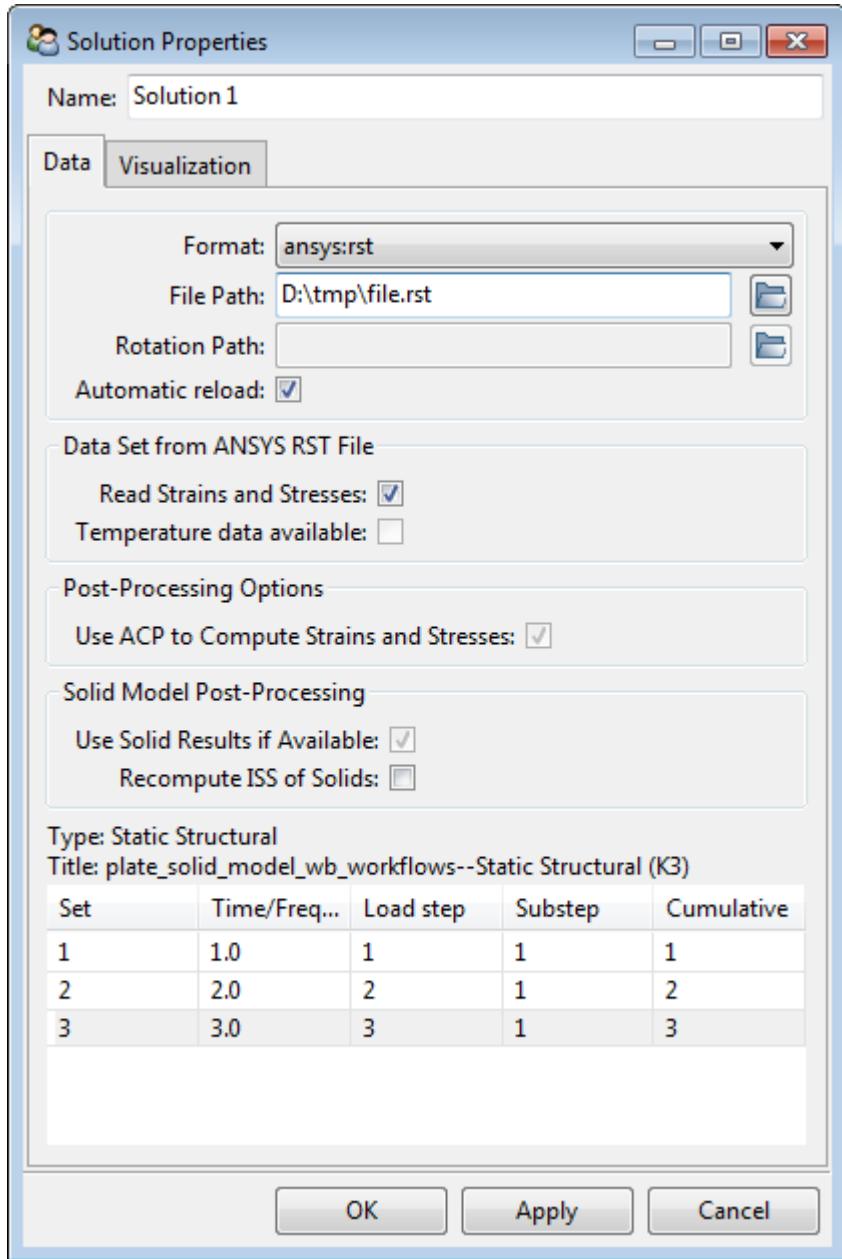
- *Properties ...*: opens the solution properties
- *Update*: updates the specific solution and reloads results if it has changed.
- *Reload*: reloads the results file.
- *Delete*: deletes the selected solution.
- *Delete post-processing results*: deletes all deformations, stresses and strains computed in ACP.
- *Export Results ...*: exports results (deformation, stress, strain, temperature, failure results) as a CSV file for selected element sets.
- *Create Deformation ...*: creates deformation plot for the selected solution. (see [Solution Plots](#) for more information on all plots)
- *Create Strain ...*: creates strain plot.
- *Create Stress ...*: creates stress plot.
- *Create Failure ...*: creates failure plot based on a failure definition.
- *Create Temperature ...*: creates temperature plot if data is available.

- *Paste*: pastes a copied plot.

4.1.17.1.1. Solution Properties

The import settings and load step selection can be done in the Data tab of the Solution Properties. On the Visualization tab, the deformation scale of the plot visualization can be configured.

Figure 4.94: Solution Properties window showing three solutions on the Data tab



4.1.17.1.2. Name

Choose the name of the solution, which is later used in post-processing.

4.1.17.1.3. Format

You can import the results in one of two ways:

- The first one is to import the *.rst file. This is the result file from the ANSYS Solver. All of the information is in the file.
- The second one is to import the deformation and rotational result files. These files must first be generated in ANSYS Mechanical APDL with the command *PRNSOL*.

4.1.17.1.4. Paths

If you choose to import results from a *.rst file, only the *.rst path has to be given. With the PRNSOL command a file for the deformations and one for the rotations will be created and therefore two paths have to be defined.

When the option *Automatic Reload* is active, ANSYS Composite PrepPost checks for any change in the result files. If changes occur, ACP reads and stores the new results automatically.

4.1.17.1.5. Data Set

Stresses and Strains

This option exists only with the import of the *.rst file. Stresses and strains are generally compiled in the *.rst files. By default, the stresses and strains are directly read from there. This option activates the calculation of stresses and strains within ACP from deformation and rotation fields ignoring any stresses or strains in the results file. When several loadsteps and/or substeps are present select which set is to be imported. Of course, this will require additional time for post-processing and is only recommended for a linear analysis. Interlaminar stresses and strains cannot be calculated for linear triangular elements within ACP.

Temperature Data

This check box serves as an indicator when the imported results file includes a data block for a temperature field. In such a case, the temperature field can be visualized or temperature dependent material data (if defined) can be used in the stress and strain analysis.

4.1.17.1.6. Solid Model Post-Processing

Solid Results

This check box serves as an indicator if solid elements, generated by ACP, are present in the *.rst file. In this case, the solid results are mapped on the reference shell elements. The mapped results are visible when the *Solid Models* are hidden. The post-processing functionality can be used for both layered shell and solid elements.

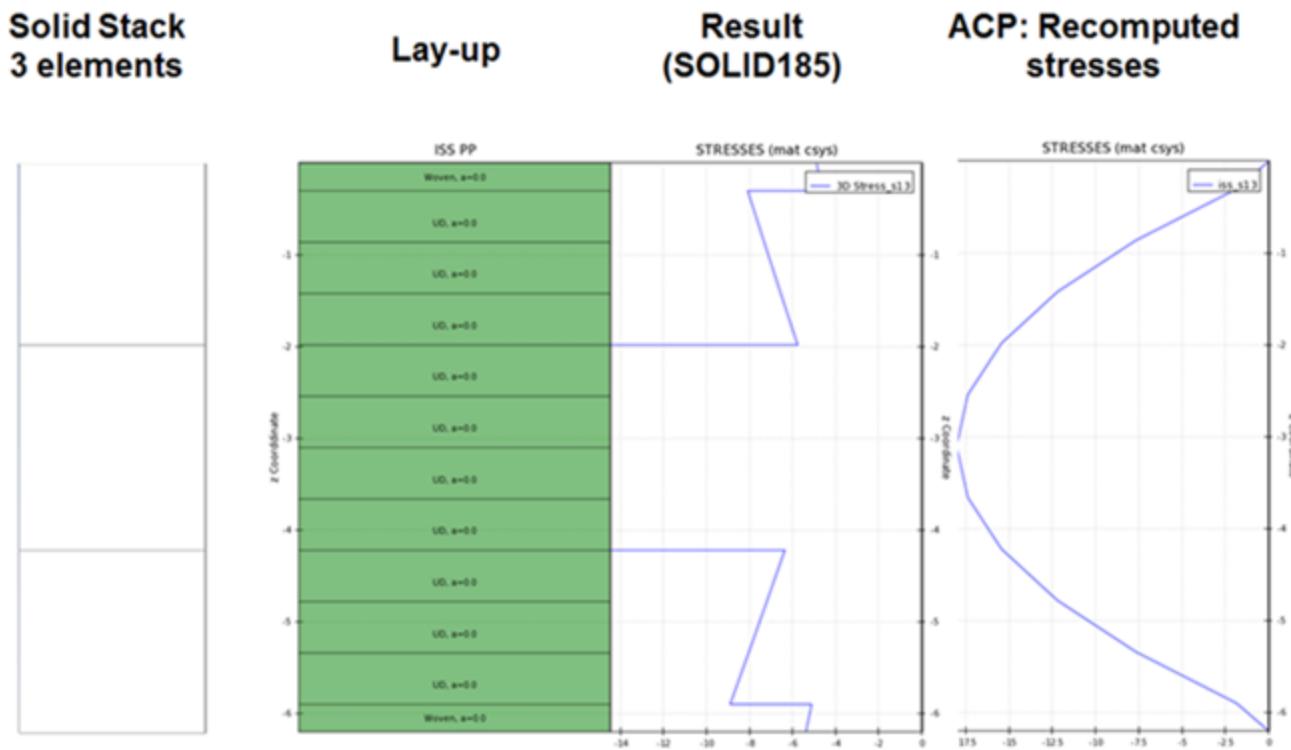
Recomputation of interlaminar shear stress in solids

Interlaminar stresses in solid models do not necessarily fulfill boundary conditions and continuity requirements. In reality, the interlaminar stresses are zero on the free surfaces if no external loads act on them. In addition, the stress distribution through the thickness should be C0 continuous. The check box *Recompute ISS of Solids* activates a calculation in ACP that re-evaluates the interlaminar shear stresses.

The recomputation algorithm undergoes the following two steps:

- Summation of all shear forces per solid stack
- Calculation of interlaminar shear stress on laminate based approach (see [Transverse shear stresses \(p. 229\)](#))

Figure 4.95: Comparison of imported and recomputed interlaminar stresses (A solid stack is a single layered solid element that represents multiple layers)



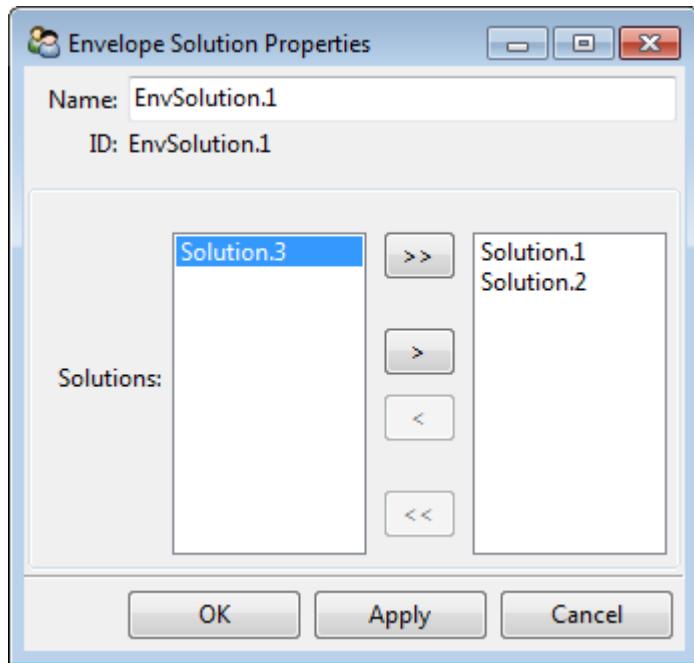
Non-zero boundary conditions are not considered in this recomputation process. The recomputed stresses take the place of the imported ones. The results file itself is, however, not altered and the re-computation can be reversed by un-checking the option and updating the solution.

4.1.17.2. Envelope Solution

The envelope solution feature can be used to combine multiple and compare load cases in failure plots. Thus the critical load case for a structure can be determined. An envelope solution can be added by selecting **Create Envelope Solution ...** from the context menu of the Solutions object. Existing solutions can be added to the envelope solution in the Envelope Solution Properties window.

See [Failure Mode Plot \(p. 197\)](#) for more details on visualizing failure plots for envelope solutions.

Figure 4.96: Envelope Solution Properties windows



4.1.17.3. Solution Plots

All analysis results can be visualized as solution plots in ACP (Post). The solution plots are attached to individual solutions. The available plot types are:

- Deformation Plot
- Strain Plot
- Stress Plot
- Failure Plot
- Temperature Plot

Common Plot Settings

The plot settings are largely similar for all plot types regardless of whether they are solution or layup plots. Each plot can be configured through the plot property window. This property window has typically got two tabs - one General tab and one Legend tab. The General tab is where the results component and geometry scope are defined. It is possible to configure a plot to display only a particular section of a component. The Legend tab controls the format of the plot legend.

The common general settings in the plot properties are the following:

- *Name*: sets the name of the plot.
- *Data Scope*: determines what scope is used in the plot. Element sets, OES, Modeling Plies & Sampling Elements can be selected in the data scope. The data scope of a sampling element covers all plies that are intersected by the sampling element.
- *Ply-Wise*: activates the ply-wise plot display. Data is only displayed when a ply is selected in the Modeling Ply Groups, Sampling Element Plies or Solid Model Analysis Plies.

- *Show on Solids*: shows plot data on solid elements and for solid elements only.
- *Spot*: location where shell results are evaluated (either top or bot).
- *Component*: gives a selection over the results component that is plotted (*uz* = vertical deformation for example).
- *Ply Offsets*: visualize the results of a ply-wise plot on the selected plies at their true or scaled offset from the reference surface.

The legend settings are the same for all plots:

- The legend is formatted automatically by default but can be customized to suit.
- limits can be set to be min/max limits
- limits can be set as thresholds on the penultimate labels on the contour plot scale
- values above limits can be colored in non-rainbow scale colors (grey and pink).

Visualization Mismatch

The failure plots shows the critical values for all defined failure criteria (modes) and integration points. The strain and stress plot, on the other hand, illustrates the values at the element center (interpolation). Therefore the absolute strain and stress peaks are not displayed in the plots and the elements have a constant value. This can cause graphical inconsistency between the strain/stress and failure plot.

Deformation Plot

- *ux*: translation in X direction.
- *uy*: translation in Y direction.
- *uz*: translation in Z direction.
- *rotx*: rotation around the X axis.
- *roty*: rotation around the Y axis.
- *rotz*: rotation around the Z axis.

Strain Plot

Strains can be displayed ply-wise or for an entire laminate. The strain results can be plotted for the following component directions:

- 1 = material 1 direction
- 2 = material 2 direction
- 3 = out-of-plane normal direction
- 12 = in-plane shear
- 13 = out-of-plane shear terms
- 23 = out-of-plane shear terms

- $/$ = 1st principal direction
- $/\!/$ = 2nd principal direction
- $/\!/\!$ = 3rd principal direction

Stress Plot

There is the option to compute interlaminar normal stresses for shell elements. See [Interlaminar Stresses](#) for background information. Stress results can be plotted for the following stress component directions:

- 1 = material 1 direction
- 2 = material 2 direction
- 3 = out-of-plane normal direction
- 12 = in-plane shear
- 13 = out-of-plane shear terms
- 23 = out-of-plane shear terms
- $/$ = 1st principal direction
- $/\!/$ = 2nd principal direction
- $/\!/\!$ = 3rd principal direction

Failure Mode Plot

The failure plot can be used to display the safety factor for first ply failure of a pre-defined failure criteria definition. There are three kinds of safety factors that can be displayed in a failure contour plot. Additionally text labels can be activated to show the critical failure mode and in what layer it occurs. In the case of an envelope solution, the critical load case can also be shown. The toolbar button  switches the display of activated element text labels on and off. For information on Failure Definitions see [Post-processing](#).

Available safety factor components:

- Inverse Reserve Factors (IRF)
- Margins of Safety (MoS/MS)
- Reserve Factors (RF)

The failure plot properties have the following additional options:

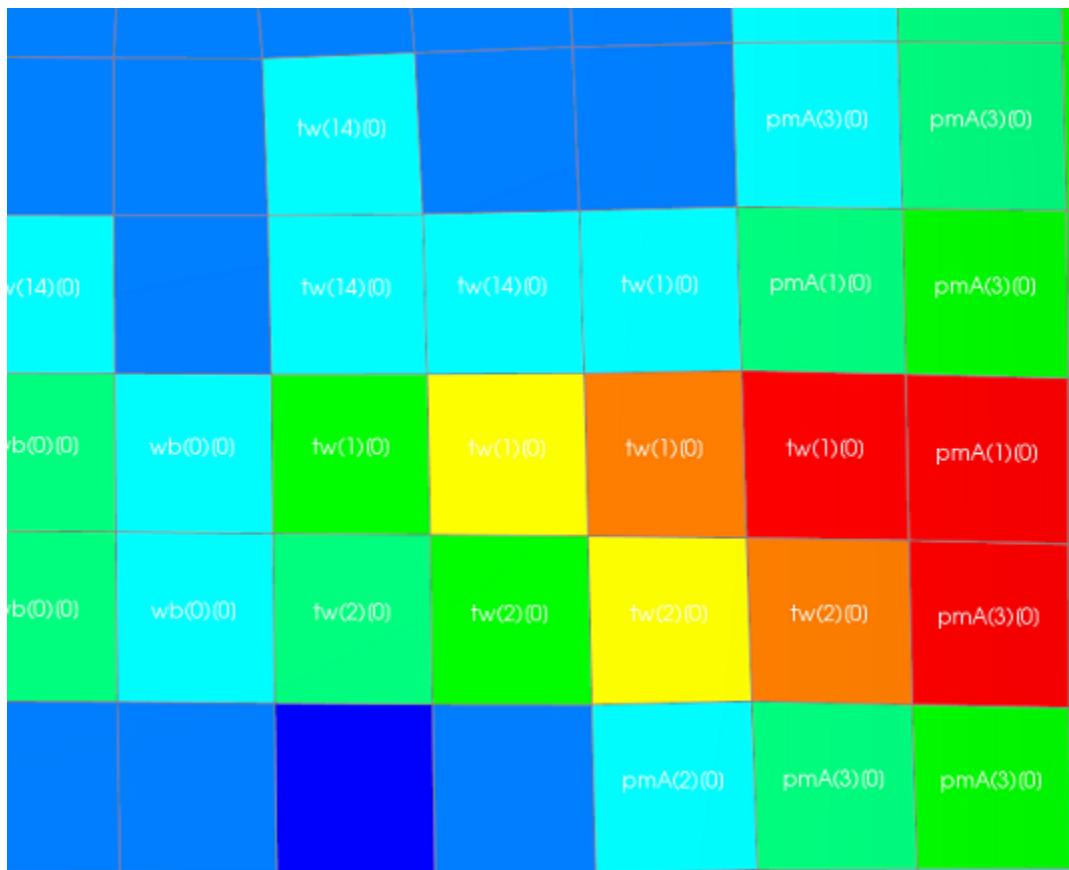
- *Failure Criteria Definition*: drop-menu for selecting the desired failure criteria definition.
- *Show Critical Failure Mode*: activates the critical failure mode as an element text label.
- *Show Critical Layer*: activates the layer index of the critical failure mode as an element text labels.
- *Show Critical Load Case*: activates the solution index of the critical failure mode as an element text labels in the case of envelope solution.

- **Threshold for Text Visualization:** sets the threshold so that element labels are only shown for as of a certain IRF, RF or MoS level.

Note

The critical layer index counts from the reference surface upwards and starts at layer 1. The sandwich failure criteria top and bottom sheet wrinkling are evaluated for a sandwich structure as a whole and cannot be linked to specific layer. The layer index shows 0 in this case. The critical load case index starts at 0. In the **Envelope Solution**, the solution in position n is plotted with number n-1.

Figure 4.97: Scene with Failure Mode Plot activated. Critical failure mode, critical layer and critical load case are displayed above the visualization threshold.



Temperature Plot

The temperature plot can display a temperature field results on solid elements if the temperature data is available in the results file.

4.1.18. Scenes

Scenes are windows that contain the visualization settings of the composite model. New scenes can be added or existing ones can be modified by hiding or showing visualization features. The visualization of the following features is saved in a scene:

- Element Sets

- Edge Sets
- CAD Geometries
- Rosettes
- Section Cuts
- Solid Models

In a new scene, all Element Sets, Section Cuts and Solid Models are shown.

Scene Properties

The scene properties set the scene name and the title which is displayed in the top right corner of the scene. They also control the legend settings for draping plots. The plot can be activated through the Draping Mesh button  or with the 'show' checkbox in the Scene Properties.

The Draping Plot shows the average shear (distortion) angle of each element. The angles are given in degrees and they are the average absolute values of the corner angles differing from 90 degrees. Therefore no distortion is equal to zero degrees. More information on draping can be found in the section Composite Modeling Techniques under [Draping \(p. 80\)](#).

Figure 4.98: Scene Properties

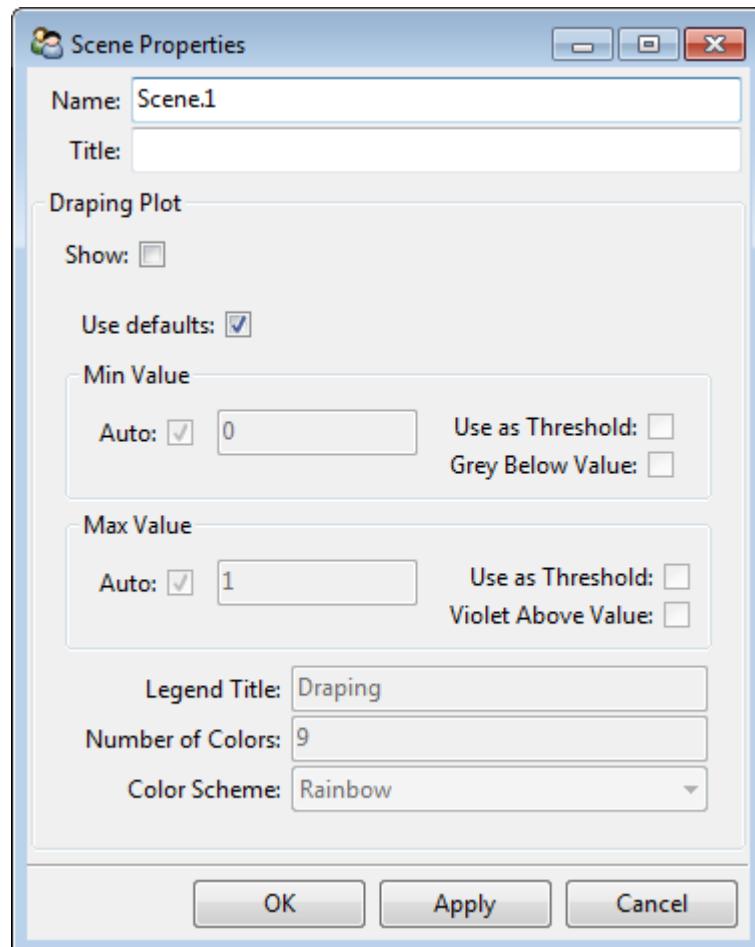
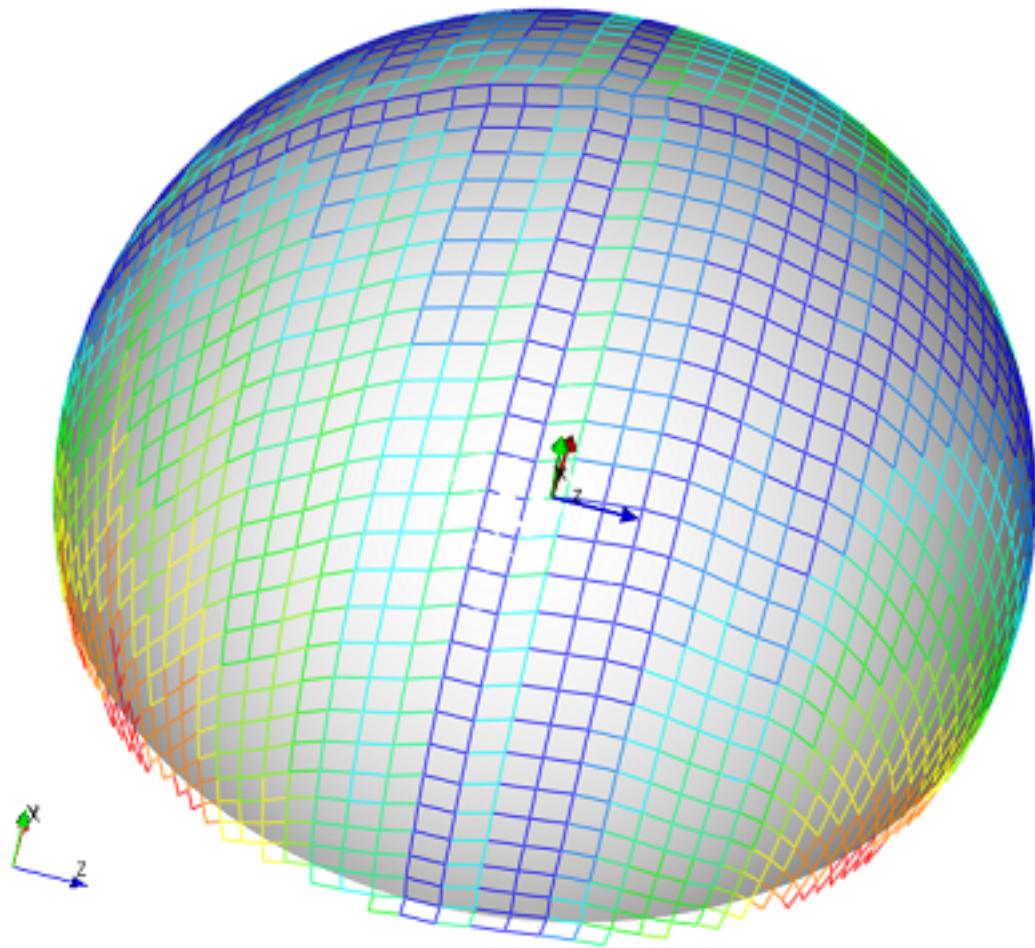


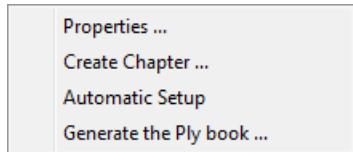
Figure 4.99: Draping plot for a hemisphere

4.1.19. Views

Views can be used to save a certain view. The selection of a view automatically updates the scene and transfer the properties of the View to the active scene. New Views can be created with the button in the toolbar (see [Scene Manipulation \(p. 51\)](#)) or via the object tree. Of course the different parameters can also defined manually.

4.1.20. Ply Book

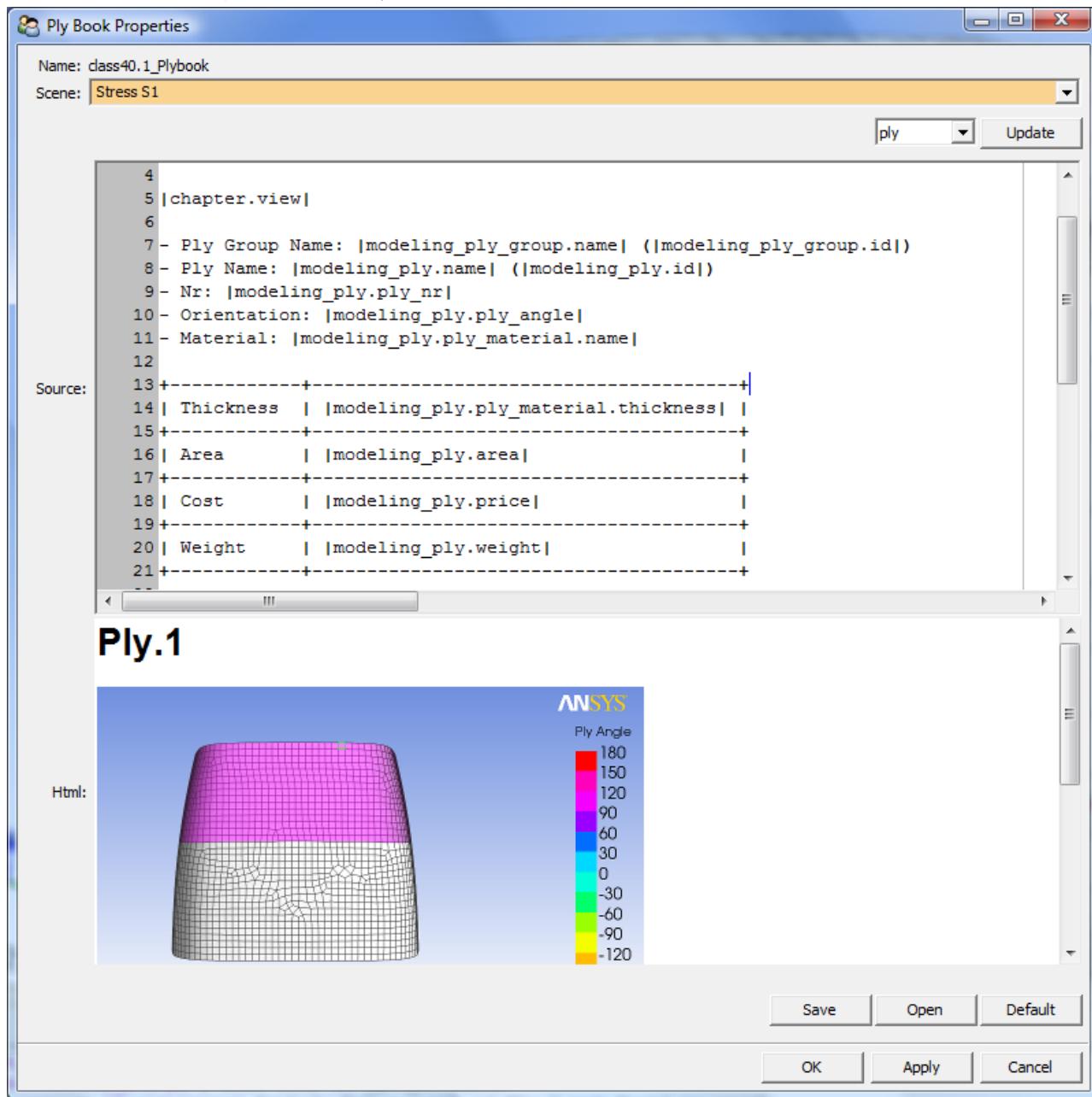
The *Ply Book* feature allows to create a report for production with all relevant information like material, orientation, angle and extension.



Properties...

The *Ply book* is divided in three parts: the *title page*, and for each chapter, the chapter title (*chapter*) and the ply definition (*ply*).

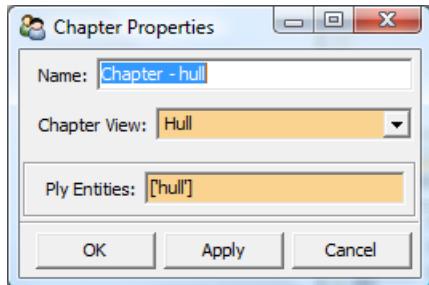
The format used to generate the *Ply book* is the *reStructuredText*.



Any modifications can be saved and opened later to be used as a template. In the second part of the window, a preview of the resulting html file is available.

Create Chapter...

A *Chapter* is defined by a *Ply Group* and certain ([Views](#)) in which the pictures must be made. You can also define the name of this chapter.



Automatic Setup

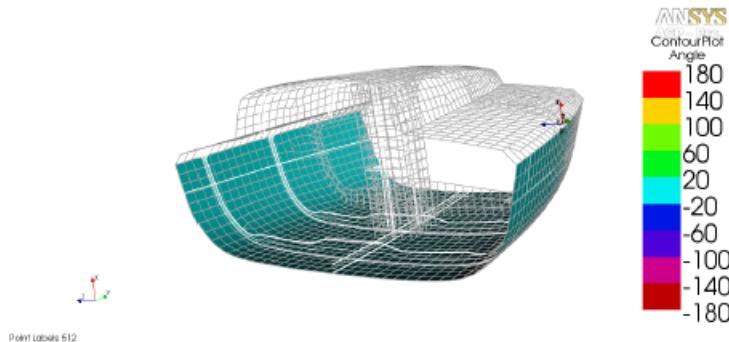
The *Automatic Setup* quickly defines the whole ply book. It defines a Chapter for each *Ply Group* in the actual selected view. The names and views can be changed later.

Generate the Ply book...

The configured *Ply Book* is exported in the html, pdf, Open Document or plain text format.

Figure 4.100: One page of a ply book

outer_skin_9



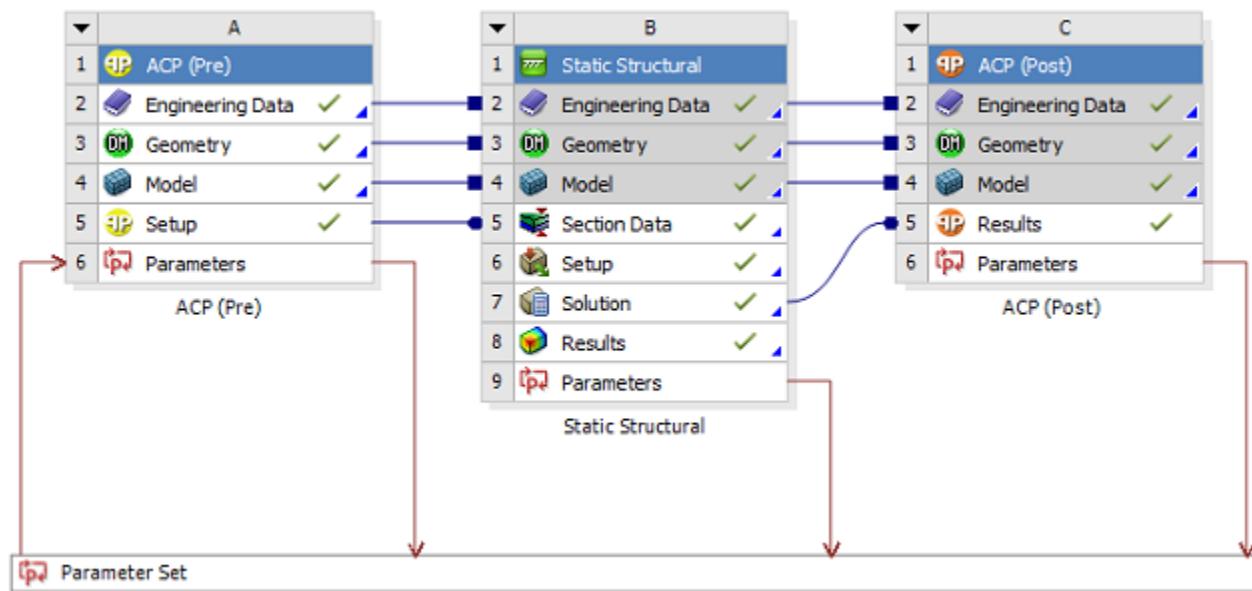
- Ply Group Name: hull (hull)
- Modeling Ply Name: outer_skin_9 (outer_skin_9)
- Ply Name: P1_outer_skin_9 (ProductionPly.8)
- Orientation: 0.0
- Material: E-Glas

Parameter	Value
Thickness	0.27
Area	18116375.4818
Cost	0.0
Weight	4891421.38007

4.1.21. Parameters

The Parameter feature in ACP connects Inputs and Outputs to the Parameter Interface in the Workbench project. The interaction of parameters within a Workbench project provides greater flexibility and capabilities to run parameter studies, what-if scenarios and Workbench driven optimizations.

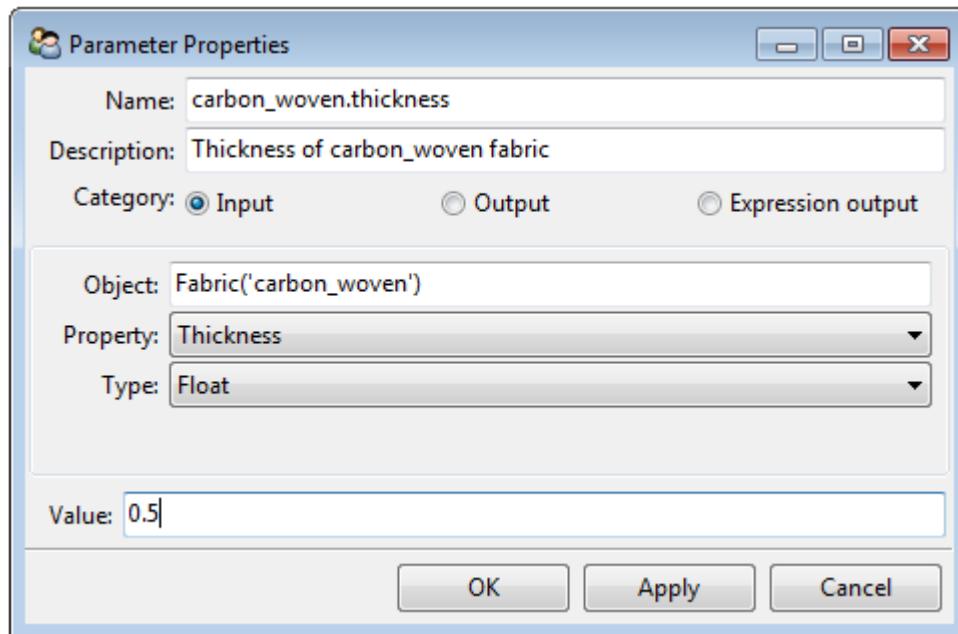
Figure 4.101: Connection of ACP and Workbench Parameter Interface



Parameter Properties

A Parameter is created by selecting *Create Parameter ...* in the context menu of the Parameters icon. The Parameter connection is then defined in the Parameter Properties window shown below:

Figure 4.102: Parameter Properties



Category

There are three different categories of parameters:

- **Input:** an input parameter taken from the Workbench Parameter interface
- **Output:** an output parameter given to the Workbench Parameter interface

- **Expression output:** an output parameter that can be an expression or formula involving multiple parameters

Object & Property

A parameter is associated to a property of an object, the thickness of a Fabric for example. The user has to select an object from the ACP feature tree before he can choose a property from the drop-down selection.

Type

Naturally, not all parameters have the same format type. The parameter format type is determined automatically if there is only one possible option. In certain cases however, more than one format type is available for a given property of an object. A brief description of all format types is given below. Note that units are not transferred from ACP to Workbench. The parameters appear as dimensionless numbers in the Workbench Parameter interface.

- **Bool** is a boolean format. Its value can either be "True" or "False".
- **Float** is the format used for real numbers.
- **Float List** is a beta option that requires the activation of the Advanced Parameters add-on.
- **Int** is the format used for integers.
- **None** is the default selection if multiple options are available. It implies nothing has been selected.
- **String** is the format used when the parameter is a string within a selection of text strings. The values in the string list can be called up via an index number in the Workbench parameter interface. The first entry in the string list has the index value 1. The rest follows sequentially. An example of a string parameter is a ply material for which the selection could be Fabrics, Stackups or Sublaminates.

Value

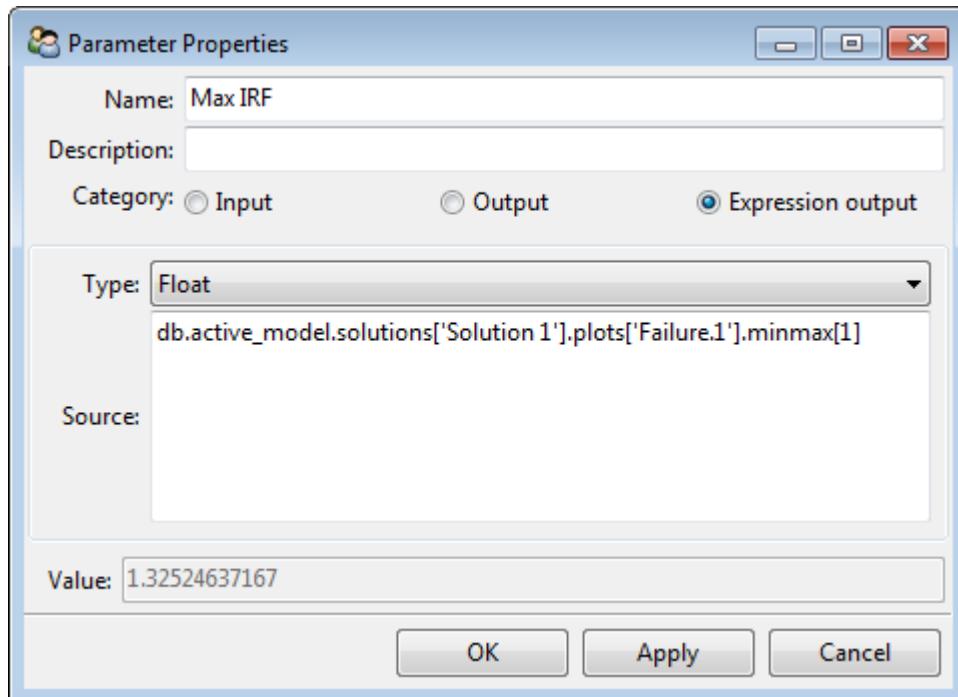
The Value box displays the current value of the parameter. The value can be altered depending on whether the parameter is an Input or an Output.

Settings for an Expression Output

The formulation of an Expression output requires a basic understanding of [The ACP Python Scripting User Interface \(p. 259\)](#).

The Parameter Properties displays a "Source" field that accepts Python code. Various information stored in the ACP database can be accessed. It is possible to enter several expressions and perform basic mathematical operations on them. An example of an expression output is shown below. The maximum inverse reserve factor is retrieved from the active contour plot.

Figure 4.103: Setting the maximum IRF as an output parameter



4.1.22. Material Databank

It is possible to build a *Material Databank* which can be used in different projects. The databank can also be saved on a intranet hard drive that all ACP users have access.

The structure of the *Material Databank* is exactly the same as in the model. For more information, see [Material Data](#). The Databank is stored as `*.acpMcd` and it can be managed through the right click menu:

Figure 4.104: Material databank



The units of the Databank and of the model could be different. Use Copy/Paste to transfer a material from one to the other, the values of the materials are converted automatically.

A default *Material Databank* is installed with ANSYS Composite PrepPost. These values are given as information and their use is under the user's responsibility. ANSYS and EVEN are not responsible of the validity of the values.

File location: <ACP installation folder>/databases

4.2. Postprocessing

More information regarding the background of the ACP postprocessing can be found in the theory sections [Failure Analysis \(p. 231\)](#) and [Interlaminar Stresses \(p. 225\)](#).

- 4.2.1. Failure Criteria
- 4.2.2. Failure Mode Measures
- 4.2.3. Principal Strains and Stresses
- 4.2.4. Limitations & Recommendations

4.2.1. Failure Criteria

In the following all available failure criteria are listed together with their failure mode abbreviations as used in failure mode plots.

Terms:

- e = strain, s = stress
- 1 = material 1 direction, 2 = material 2 direction, 3 = out-of-plane normal direction, 12 = in-plane shear, 13 and 23 = out-of-plane shear terms
- I = principal I direction, II = principal II direction, III = principal III direction
- t = tension, c = compression

Criteria:

- Maximum Strain. Failure modes: e1t, e1c, e2t, e2c, e12
- Maximum Stress: s1t, s1c, s2t, s2c, s3t, s3c, s12, s23, s13
- Tsai-Wu 2D and 3D: tw
- Tsai-Hill 2D and 3D: th
- Hashin: hf (fiber failure), hm (matrix failure), hd (delamination failure)
- Puck (simplified, 2D and 3D Puck implementations are available): pf (fiber failure), pmA (matrix tension failure), pmB (matrix compression failure), pmC (matrix shear failure), pd (delamination)
- LaRC 2D and 3D: lft3 (fiber tension failure), lfc4 (fiber compression failure under transverse compression), lfc6 (fiber compression failure under transverse tension), lmt1 (matrix tension failure), lmc2/5 (matrix compression failure)
- Cuntze 2D and 3D: cft (fiber tension failure), cfc (fiber compression failure), cmA (matrix tension failure), cmB (matrix compression failure), cmC (matrix wedge shape failure)
- Sandwich failure criteria
 - Wrinkling: wb (wrinkling bottom face), wt (wrinkling top face)
 - Core Failure: cf
- Isotropic failure criteria- Von Mises: vMe (strain) and vMs (stress)

Weighting factor:

- The inverse reserve factor of each failure mode is multiplied by the accordant weighting factor.
- 1: no safety
- 2: safety of two

An overview of all available failure criteria is given in the [Section Failure Analysis](#).

4.2.2. Failure Mode Measures

Currently, three failure mode measures are available:

- IRF = Inverse Reserve Factor (IRF) defines the inverse margin to failure. Load divided with IRF is equal to the failure load. $IRF > 1$ discloses failure.
- MoS = Margin of Safety (MoS) defines the margin to failure. MoS is defined as $(1/IRF - 1)$. $MoS < 0$ discloses failure.
- RF = Reserve Factor (RF) defines the margin to failure. Load multiplied with RF is equal to the failure load. $RF < 1$ discloses failure.

4.2.3. Principal Strains and Stresses

For strains, only the first (el) and second (ell) principals are evaluated. The principal strain (e) and stress (s) values are ordered in descendent order.

4.2.4. Limitations & Recommendations

Interlaminar shear strains of linear triangular shell elements can not be evaluated by ANSYS nor by ACP. Interlaminar shear stresses of linear triangular shell elements can be evaluated by ANSYS but not by ACP. By default, the ANSYS *.rst results file contains stress and strain data, however, they may be excluded. In that case, ACP can evaluate stresses and strains on the basis of the deformation and rotation fields in the ANSYS results file. Non-linear effects are not considered by ACP and will induce inaccurate stresses and strains. In general, it is recommended to include the stress and strain data in the *.rst data. More information can be found in the section [Solutions](#).

ACP provides a unique method to evaluate interlaminar normal stresses (INS) for shell elements. This calculation of the INS requires the evaluation of the shell curvature. It is therefore recommended to use quadratic shell elements when INS are of interest. The quadratic elements contain the curvature information per element and offer a better approximation than linear elements. The curvature for a linear shell element is determined from its neighboring elements. This evaluation does not consider INS induced by edge effects or out-of-plane loads (e.g. inserts, pressures, etc.).

4.3. Available Interfaces to FE Packages

4.3.1. ANSYS

Model format: ANSYS CDB files.

Nodal solutions can be loaded from:

- PRNSOL file formats. Export ANSYS results using something like:

```
/FORMAT, 10,G,25,15,1000,1000
```

```
PRNSOL,U
```

```
PRNSOL,ROT
```

- **RST file interface. It allows to load nodal and element results directly from the ANSYS result file.**

Use these options for element result import:

- Shell91: keyopt 5-3 and keyopt 8-1 (See [Shell 91 keyoptions](#))
- Shell99: keyopt 5-2, keyopt 8-1, keyopt 9-0 (See [Shell 99 keyoptions](#))
- Shell181: keyopt 8-2 (See [Shell 181 keyoptions](#))
- Shell281: keyopt 8-2 (See [Shell 281 keyoptions](#))
- Solid46/185/186: keyopt 8-1 (See [Solid 185 keyoptions](#) and [Solid 186 keyoptions](#))

Keyoption 8 must be defined via the command line (keyopt,<et_number>,8,1)

- ERESX,NO (Copy integration point results to nodes)

Supported shell and solid (for post-processing only) element types: SHELL181, SHELL281, SHELL91, SHELL99, SOLID46, SOLID185, SOLID186

Supported element property definition commands: SECTYPE, SECOFFSET, SECBLOCK, SECCONTROL, RLBLOCK

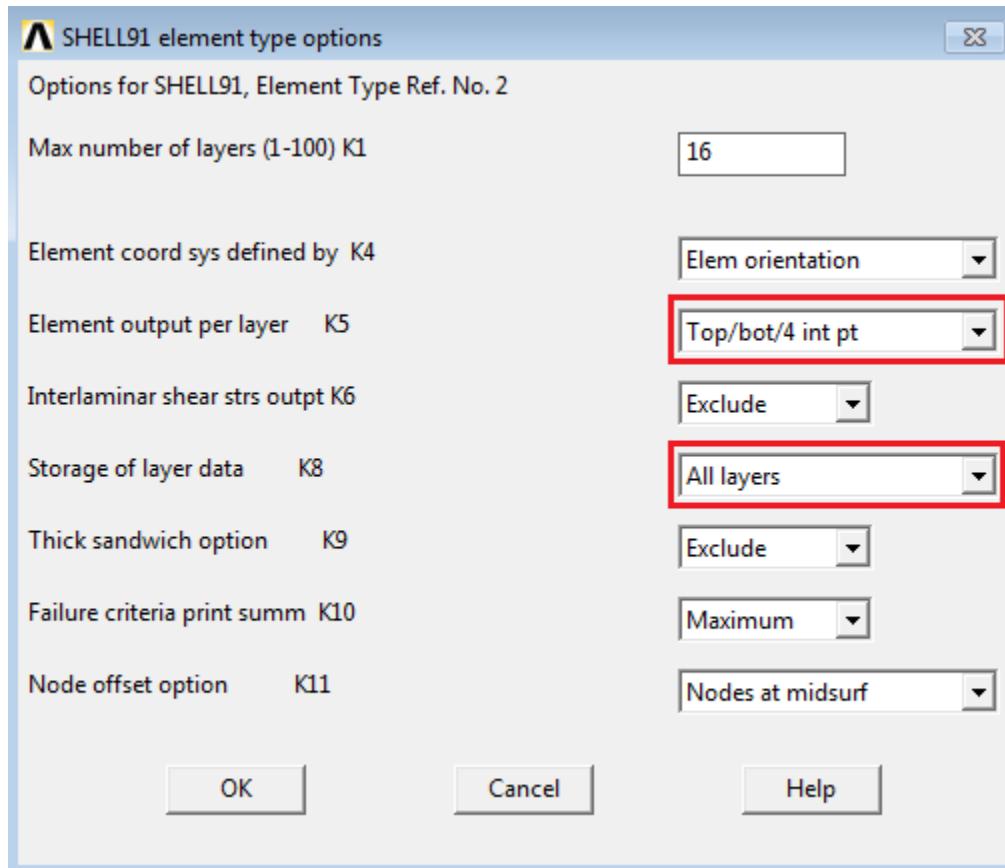
Figure 4.105: Shell 91 keyoptions

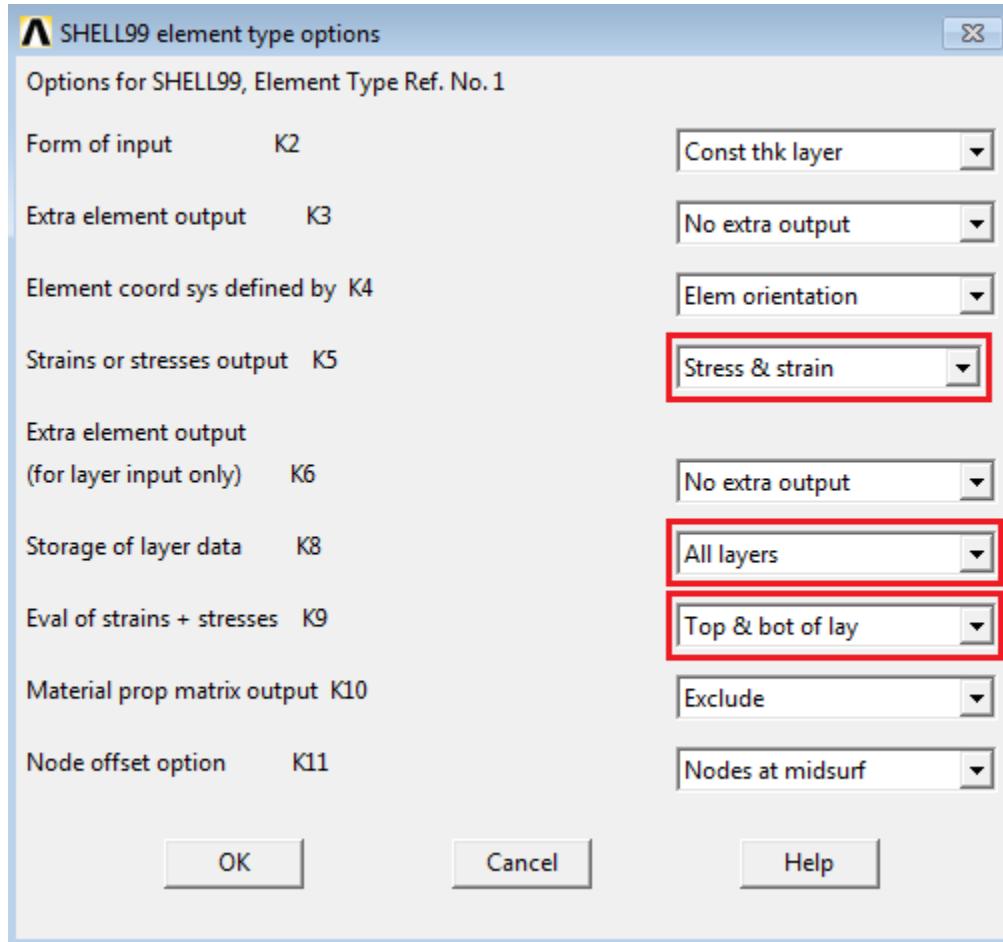
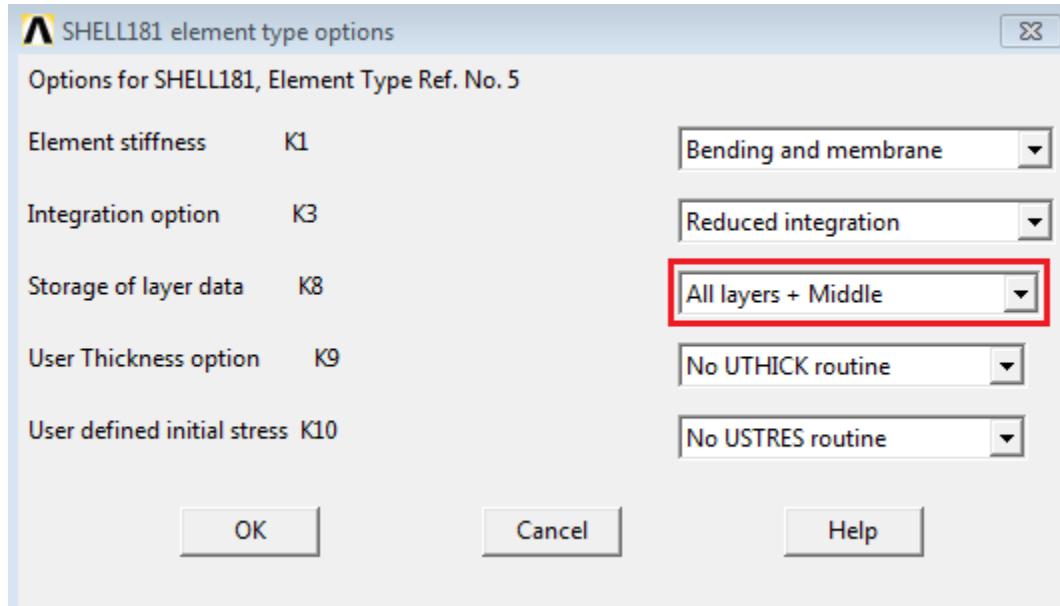
Figure 4.106: Shell 99 keyoptions**Figure 4.107: Shell 181 keyoptions**

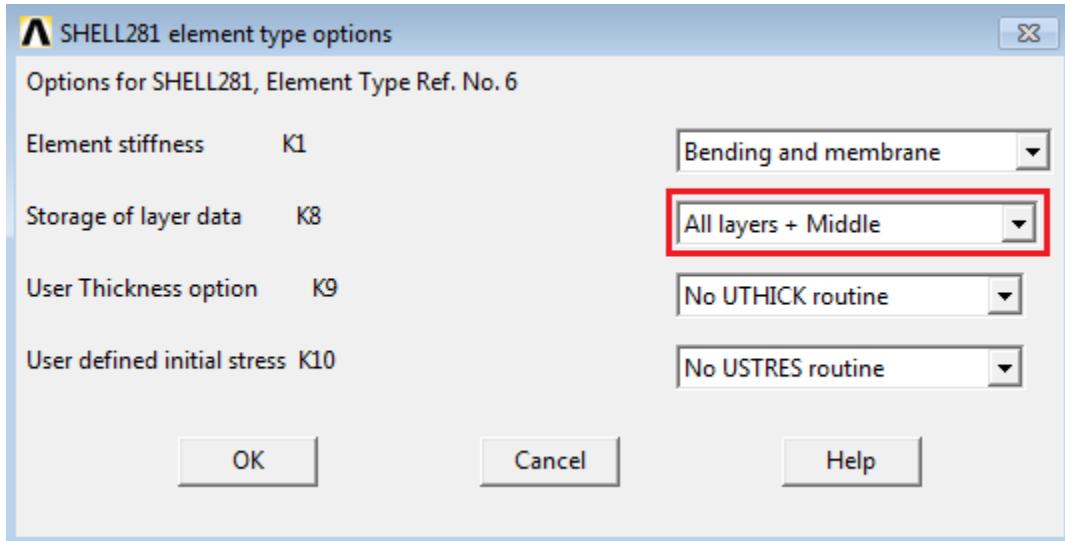
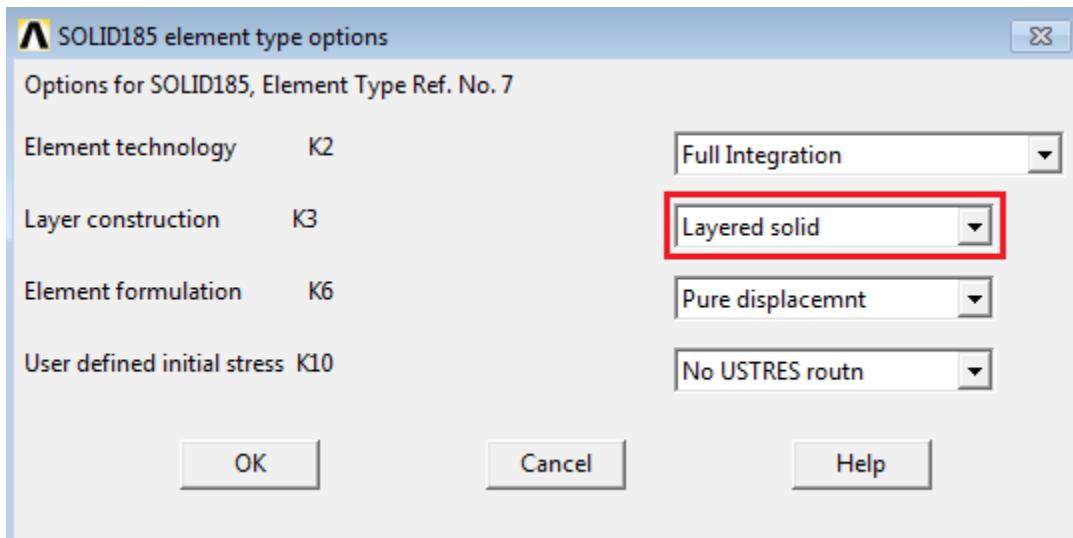
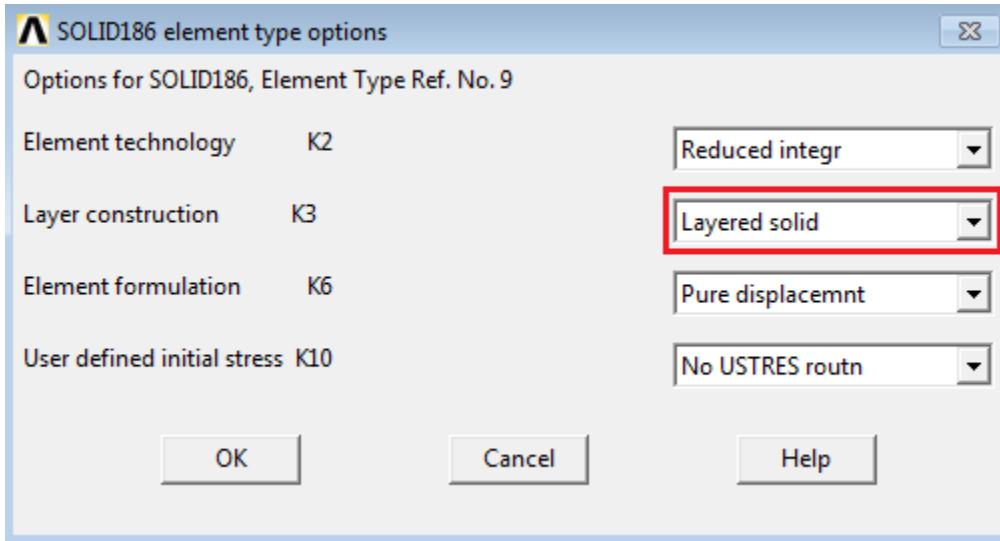
Figure 4.108: Shell 281 keyoptions**Figure 4.109: Solid 185 keyoptions**

Figure 4.110: Solid 186 keyoptions

4.3.2. ESAComp

Model format: ESAComp XML files.

Export

Material data (Fabrics, Stackups and Sub-laminates) and Sampling Elements can be exported to ESAComp XML. A Fabric represents a Ply in ESAComp, where Stackups, Sub-laminates and Sampling Elements are exported as Laminates.

To be sure that the imported values in ESAComp are in accordance to the ACP model, the FE import units in ESAComp have to be checked first.

- Open FE import and export units in ESAComp (See [ESAComp Options](#)).
- Adjust units (See [ESAComp FE import and export units](#)) which can be checked in the Model Properties (see [Units](#)).

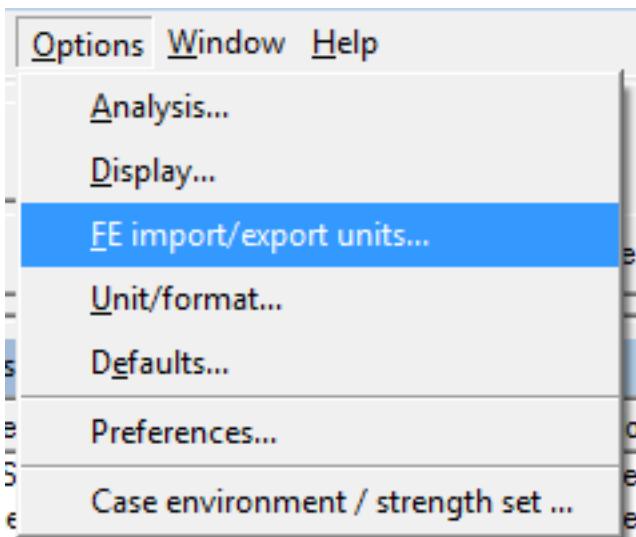
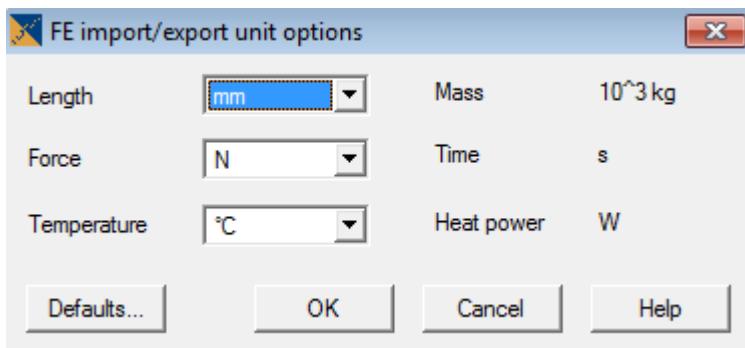
Figure 4.111: ESAComp Options

Figure 4.112: ESAComp FE import and export units

Import

As for the export, check the units used in ESAComp and ANSYS Composite PrepPost.

In Escaomp, there are 2 ways to export the data:

Script Format (recommended)

In ESAComp, you can export Plies and Laminates through the Menu *FE Export / ANSYS ACP*. It generates a Python script, in which all the information is stored in ACP format.

Run the script in File Menu.

XML Format

In ESAComp, export your data as *.xml file in the menu File. Then import this same data in ANSYS Composite PrepPost with the drop down menu in Material (see [Material Data](#))

Only the material data can be imported.

Important

The Import from ESAComp XML does not create the material in ANSYS Composite PrepPost. It only changes the properties. So the material must be created before with the same name as in ESAComp.

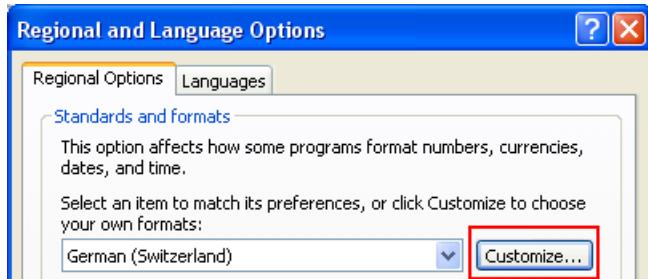
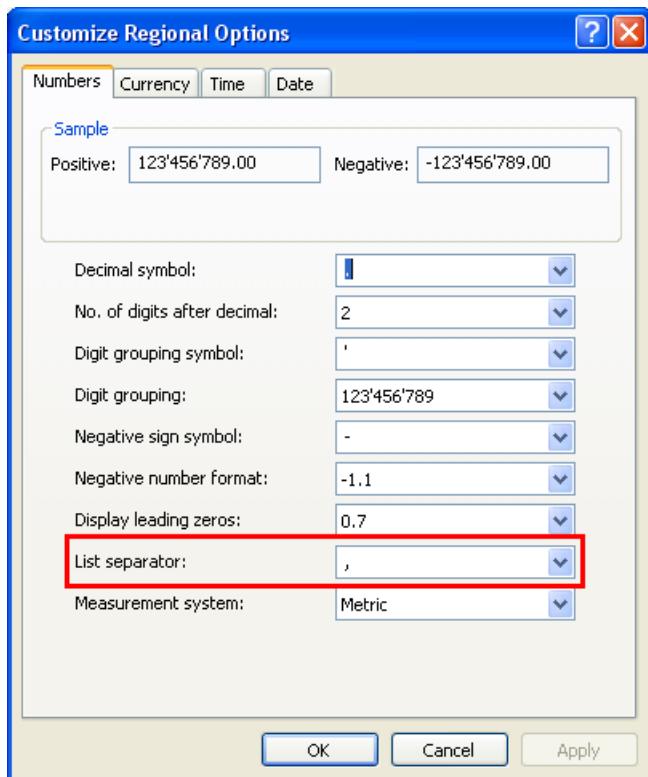
4.3.3. CSV Files

Materials, Look-up Tables, the rules' definitions and the Modeling ply groups can be exported and /or imported with a csv Format (comma separated values).

Is is very useful to edit, change and reload the corresponding definitions in an optimization process.

Warning

The CSV format uses a "," as list separator. In some "Regional Options" in Windows, the list separator is defined by another character (very often ";"). In this case, the *.csv files will not be properly read and written by Excel. In this case, change the list separator under Settings / Control panel

Figure 4.113: Regional options**Figure 4.114: List separator customization**

4.3.4. HDF5

The HDF5 Format is used to exchange data between CAE and CAD softwares. With ANSYS Composite PrepPost, the HDF5 is especially developed to exchange data with the software FiberSIM(R) from Vistagy, Inc. (more info by [Fibersim](#))

Export

The different material and lay-up definitions are exported as material and geometrical information to the *.h5 file. These information can be imported into FiberSIM(R) for further design and production management.

Import

The material information are imported to the material data.

The geometrical information contained in the *.h5 file is imported into several look-up Tables. These tables are used to define the Oriented Element Set reference direction, the draping direction and the ply thickness.

4.3.5. LS-Dyna

The export of the solid model to a LS-Dyna *.k file is available through the Python Scripting Interface. Please consult the [SolidModel \(p. 316\)](#) for more information.

4.4. FAQ

1. Does the surface mesh dictate the final ply location? (i.e. Do you need to know where the different layup sections are going to be before you mesh, then apply the composite directly to the mesh?)

No, the spatial location of plies is linked to the definition of oriented-element-sets and optionally to some rules. The oriented-element-sets themselves are again linked to named selections typically defined in AWB. Therefore the ply-definition is independent from the mesh.

2. What kind of results does the draping tool provide?

The draping tool indicates the shear stress in every element. In a case where a certain amount of shear stress leads to wrinkles or other undesired effects is strongly dependent of the fabric used. Therefore the user (or the manufacturer of the fabric) must have knowledge about shear-limits for a specific fabric that still allows for reasonable draping. The final fiber angle can be nicely visualized and is of course considered in the analysis-model.

3. Can it lay up complex sheets with cut-outs included? (e.g. a conical surface wrapped with 0-degree-to axis-sheet with large "v" cuts to prevent overlap) or even a simple hole in the prepreg sheet to fit around a surface feature like a fixing point.

The handling of holes is not an issue. Holes do not even have to be treated in a special manner for the draping and outline-generation. The case of periodic surfaces like cylinders, spheres or this conical surface is not handled yet in ANSYS Composite PrepPost with respect to draping and outline-generation. The functionality to generate an additional border as needed in this case is not automated. However there is always the manual workaround to split such a surface in two areas which solves the problem at hand.

4. Are parameters exchanged with WB?

Parameters can be passed to ANSYS Workbench with the Parameter Feature. The user can perform parameter studies on the composite layup and use ANSYS Composite PrepPost with the Workbench Parameter Manager and Design Exploration features.

5. Does ANSYS Composite PrepPost allow the user to orient SOLID layered models in a consistent manner?

Once the layers have been defined on a shell model, the user can extrude parts of the surface model to create a solid mesh that will contain all layers. Once the solid model has been created, orientations and layers can be edited further.

6. Does ANSYS Composite PrepPost allow the user to define layers for tetrahedral meshes?

It is not possible to define layers on a tetrahedral mesh with the current version of ANSYS Composite PrepPost.

7. Does ANSYS Composite PrepPost handle layer drop-offs? Does it provide a consistent layer numbering scheme for post-processing layer-by-layer?

ANSYS Composite PrepPost provides a consistent layer-handling between pre- and post-processing. Namely the sampling-elements make it very easy to access the desired layers, both in pre- (view of stacking sequence on the sampling elements) and post-processing (results on specific layers on the actual plies). Layer drop-offs are also supported.

8. Can I check ply orientation, layer-by-layer?

ANSYS Composite PrepPost provides various efficient tools for the visualization of the fiber orientation (both theoretical and draped fiber orientation are available), along with section-cuts as well as ply-offset settings.

9. How can I check the stacking sequence at different locations?

Both 'section-cuts' and sampling elements allow to check the stacking sequence. The tree structure of ANSYS Composite PrepPost also allows the user to view the stacking sequence in a compact form.

10. Can I easily modify ply/element/material orientation to perform design iteration studies?

The user interface provides an easy way to modify all aspects of the composite definitions (material, orientation, thicknesses, ply sequences, ...). Design studies can be efficiently carried out by using Parameter feature of ANSYS Composite PrepPost in combination with Design Points in the Workbench Parameter Manager. Additionally, the powerful scripting capabilities of ANSYS Composite PrepPost provide automation capabilities as well as the ability to perform changes in batch mode. Also, ANSYS Composite PrepPost has the ability to read layer definitions from a spreadsheet.

11. How do I access interlaminar shear stresses and thickness stresses for both shells and solids?

This is possible by choosing ply-wise-visualization in the post-processing. ANSYS Composite PrepPost provides additional through-thickness stresses using internal additional computations.

12. Can I visualize layer-by-layer failure criteria for both shells and solids?

ANSYS Composite PrepPost provides several views of the results. You can visualize all results either as maxima through the laminate or layer-wise.

13. Does ANSYS Composite PrepPost have the ability to get the worst case failure criteria and the layer in which it occurs?

The overlay-text-plot provides exactly this and can be tailored to your individual needs.

14. Does ANSYS Composite PrepPost have the ability to plot stress/strain results in both the element and layer coordinate system, along with any other pre-defined system?

The current version of ANSYS Composite PrepPost provides results in fiber-direction (and orthogonal to it) as well as along the principal directions of the stress-state.

15. Does ANSYS Composite PrepPost handle composite beam elements?

Currently ACP supports only shell as input elements. Link and beam are not visualized in ACP and can not be processed (pre and post) in ACP. The pre- and post-processing of these elements has to be performed in ANSYS Mechanical.

16. Does ANSYS Composite PrepPost compute its own failure and interlaminar shear stresses?

Some of the values are taken directly from the ANSYS results files. Additional computations are performed by ANSYS Composite PrepPost to evaluate interlaminar shear stresses and interlaminar normal stresses. ANSYS Composite PrepPost offers a more comprehensive list of failure criteria than provided by the standard composite capabilities of the ANSYS structural solvers.

17. Does ANSYS Composite PrepPost supports several loadsteps if the input data is a *.cdb?

*ANSYS Composite PrepPost writes a new *.cdb file with the lay-up definitions. Since the solve properties and commands are not stored in the *.cdb files, ACP uses a default solve script to start the ANSYS solver. This script `solve_script.inp` can be found in < Ansys installation dir>ACP<version>shareansys and has to be modified by the user to handle non-linear analysis, multiple load steps, pre-stress analysis, buckling analysis etc. The final input file `<jobname>.inp` can be found in the defined working directory.*

18. Which file formats are supported?

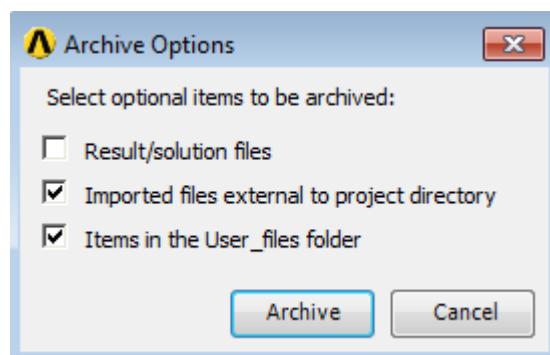
*Currently ACP supports *.cdb, *.inp and *.dat Ansys input files. In addition, a LS-Dyna *.k file interface is available as beta add-on. In addition, a beta Nastran `k-file` interface is available to import Nastran meshes.*

19. Are element and node sets created in ANSYS Composite PrepPost associative?

'Element Sets' and 'Node Sets' created in ACP are based on element and node indices, respectively. That means that changes in the mesh can not be handled. Therefore the sets created in ACP have to be redefined after a model update which includes changes in the mesh or topology. 'Named Selections' are passed to ACP and converted to 'Element Sets' or 'Node Sets'. These sets are updated automatically and are locked (user can not add or remove elements). Do not change the names of these sets to ensure an associative workflow.

20. Are the ACP composite definitions stored within an WB archive?

To ensure that all ACP composite definitions are archived, select 'Items in the User_files` folder.



Chapter 5: Theory Documentation

The theory chapter consists of the following sections:

- 5.1. Nomenclature
- 5.2. Draping Simulation
- 5.3. Interlaminar Stresses
- 5.4. Failure Analysis
- 5.5. Classical Laminate Theory

5.1. Nomenclature

Greek Symbols

α	Coefficient of thermal expansion / Fracture angle in LaRC
β	Coefficient of moisture expansion / Shearing angle in draping
γ	Shear strain, shear angle
η	Coefficient of influence
Δ	Load change
	Normal strain
ϑ	Third coordinate of the modified cylindrical coordinate system
κ	Curvature
ν	Poisson's ratio
σ	Normal stress
τ	Shear stress
ϕ	Misalignment angle
ψ	Second coordinate of the cylindrical coordinate system
ξ, η, ζ	Principal stress/strain coordinate system

Latin Symbols

b	Curve fitting parameter for Cuntze's failure criterion
C	Components of the 3D stiffness matrix
E	Young's modulus
F_i, F_{ij}	Coefficients in quadratic failure criteria, stress space
F_{12}^*	Interaction coefficient in the Tsai-Wu criterion
f	Failure criterion function
G	Shear modulus
g	Toughness ratio
H	Moisture / humidity

h	Thickness
IRF	Inverse Reserve Factor
MoS	Margin of safety
m	Interaction exponent for Cuntze's failure criterion
p	Slope parameter for Puck's action plane criterion
Q	Shear failure stress in the 23-plane / Wrinkling coefficient
R	Shear failure stress in the 13-plane / Midplane curvature radius/ Fracture resistance in Puck's action plane criterion
r	Radial ordinate R+z
r _d	Radius difference
RF	Reserve Factor
S, S	Shear failure stress/strain in the 12-plane
T	Temperature
u,v	In-plane displacement in x, y-direction
vM _σ , vM	Failure stress/strain for isotropic material
w	Through-the-thickness displacement
X, X	Failure stress/strain in the 1-direction
x	Global x-coordinate
Y, Y	Failure stress/strain in the 2-direction
y	Global y- and longitudinal coordinate of the cylindrical coordinate system
z	Thickness coordinate
1, 2, 3	Ply principal coordinate system

Subscripts

C	Core
c	Compressive
d	Delamination
E	Exposure
eff	Effective
F	Face sheet
F _b	Bottom face sheet
F _t	Top face sheet
f	Fiber failure mode
is	In-situ
m	Matrix failure mode
n	Supporting point number
t	Tensile
⊥	Perpendicular to fiber
	Parallel to fiber

<i>w</i>	Wrinkling
,	Derivative

Superscripts

<i>o</i>	Midplane strain
<i>A</i>	Action plane
<i>C</i>	Under compression load
<i>F</i>	Sum of temperature and moisture effects
<i>L</i>	Longitudinal
<i>m</i>	In misalignment frame coordinate system
<i>T</i>	Under tension load / Transverse

Acronyms

FF	Fiber Failure
IFF	Inter-fiber Failure
INS	Interlaminar Normal Stress
ISS	Interlaminar Shear Stress
LSoE	Linear System of Equations
UD	Unidirectional

5.2. Draping Simulation

Draping simulation is explained in the following sections.

- 5.2.1. Introduction
- 5.2.2. Draping Procedure
- 5.2.3. Implemented Energy Algorithm
- 5.2.4. Limitations of Draping Simulations

5.2.1. Introduction

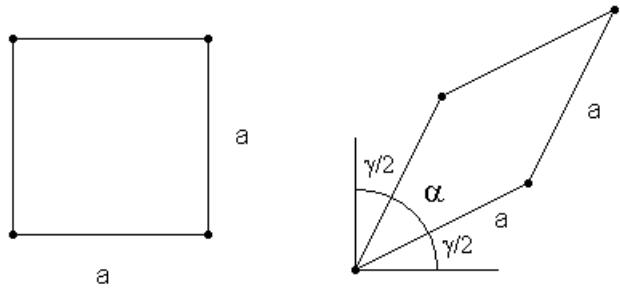
Layered composite structures are typically formed by placing reinforced plies against a mold surface in desired orientations. In the case of flat and singly curved surfaces, the orientation of the ply stays practically unchanged over the whole application area. When it comes to doubly curved surfaces, a ply can follow the surface only by deforming. In particular, dry and pre-impregnated woven fabrics can adapt to the shape of a doubly curved surface without use of excessive force. Deformation occurs with in-plane shear and up to certain deformation level, the shear stiffness of the fabric is insignificantly small. [2 (p. 351)]

When a ply deforms by shearing to follow the surface, the fiber orientation changes. Different approaches have been developed for the simulation of the so-called draping process [3 (p. 351)]. The need for draping simulation is twofold. Firstly, the manufacturability of the composite product can be assessed. Areas where the reinforcement cannot follow the surface are indicated and hence measures can be taken in design to avoid this. Secondly, the draping simulation gives the actual fiber orientations at any location in the model. This information is needed for accurate finite element analysis of the structure.

5.2.2. Draping Procedure

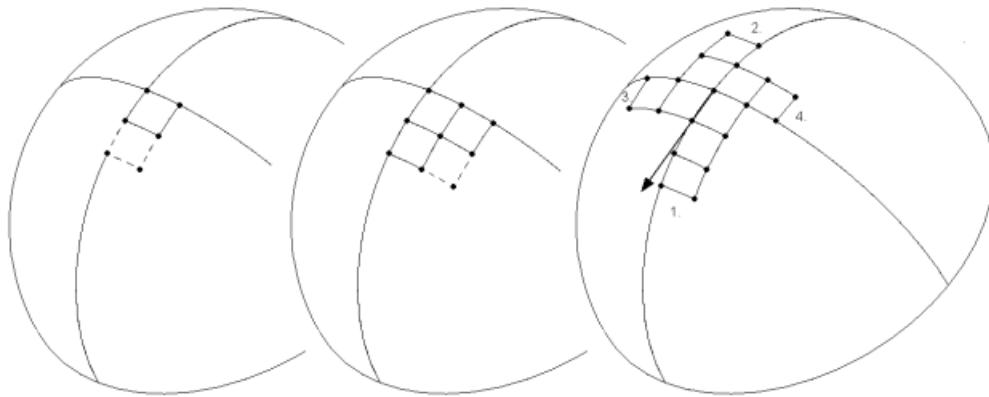
The draping simulation in ACP uses an energy algorithm. In this approach, a reinforced ply is idealized with a pin joint net model [1 (p. 351)] [6 (p. 351)]. The net consists of unit cells, which are constructed of inextensible bi-directional fibers that are pinned together at crossover node points. The deformation of the fabric takes place by pure rotation of the fibers around the pins as illustrated in Figure [Deformation of the draping unit cell](#).

Figure 5.1: Deformation of the draping unit cell



In the draping simulation, draping unit cells are laid one by one on the surface of the model so that they are fully in contact with the surface. The draping procedure involves the search for two types of draping cells: those with two or three known node points as shown in Figure [Draping scheme](#). In case of three known node points, the search algorithm seeks the fourth node point from the surface so that the distances along the surface to the adjacent node points are equal to the unit cell side lengths.

Figure 5.2: Draping scheme



Draping of a cell that has two or three known node points (left/middle) and propagation scheme using orthogonal directions (right).

When two node points are known and the locations of the other two must be determined, the search algorithm is based on the minimization of the shear strain energy [4 (p. 351)] :

$$\min E = \frac{1}{2} G \gamma^2 \quad (5.1)$$

where G is the elastic shear modulus of the uncured reinforcement. The shear deformation is related to the angle α between the originally orthogonal fibers [3 (p. 351)]

$$\gamma \approx \cos \alpha \quad (5.2)$$

The total shear strain energy of the draping cell is defined as the sum of energy computed at the four corners. The two constants can be excluded and the minimization problem becomes:

$$\min E' = \sum_{i=1}^4 \cos^2 \alpha_i \quad (5.3)$$

From this, the locations of the two node points can be determined with an iterative minimization algorithm.

The draping simulation starts from a given seed point and progresses in the given draping direction. In this phase each draping cell has initially two known node points. Draping cells are laid until the model edge is reached. Then, the procedure is repeated in the opposite direction (if applicable) and in the orthogonal directions (Figure [Draping scheme](#)). After the main draping paths have been determined, the cells with three known nodes are populated. The algorithm resolves if the whole model is draped or if there are areas where the draping simulation needs to be restarted.

The simulation determines fiber principal directions 1. These directions are mapped to the finite element model to correct laminate lay-ups accordingly. The shearing angle β is defined as

$$\beta = 90^\circ - \alpha \quad (5.4)$$

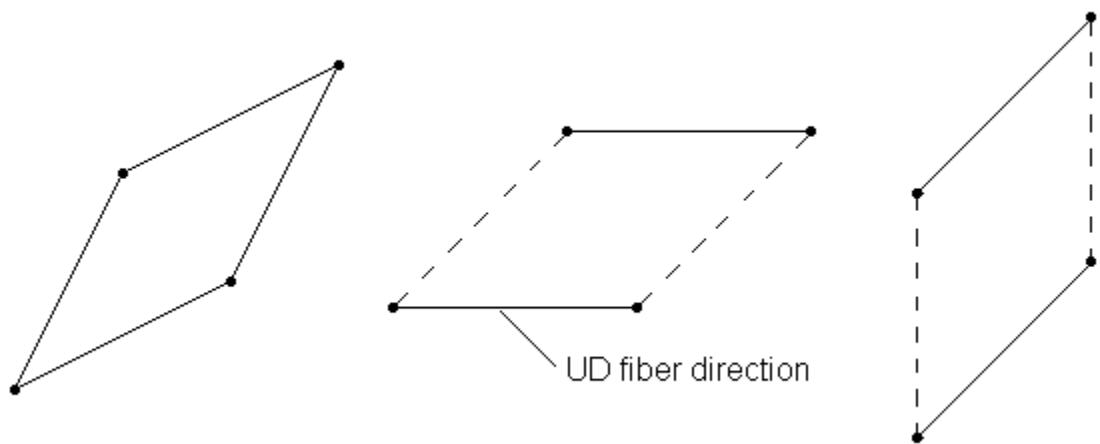
It expresses the deviation from the ideal non-sheared case. The visualization of β values over the model surface is useful for depicting problem areas. For most fabric reinforcements, the maximum deformation angle alpha is 30-40 degrees [7 (p. 351)]. When a fabric is sheared to a specific deformation level, the shear force starts to increase radically with only little increase in the shear deformation. This limit is called the locking angle. Beyond this limit buckling can be observed. The locking angle of a reinforcement can be determined experimentally. [5 (p. 351)]

The pin joint net model is specifically developed for woven fabrics, but it has been proven to work for cross ply prepreg stacks and also for single unidirectional plies when the deformation is moderate.[8 (p. 351)]

5.2.3. Implemented Energy Algorithm

In the adopted software implementation, the material draping behavior is controlled by draping coefficients d_0 , d_1 and d_2 . The coefficients are considered as weighting factors for the different draping modes. The three modes are:

- Pure shear deformation of a regular woven fabric (mode 0).
- Parallel sliding in the fiber direction of a UD ply (mode 1).
- Parallel sliding orthogonal to the fiber direction of a UD ply (mode 2).

Figure 5.3: Draping modes: mode 0 (left), mode 1 (center) and mode 2 (right).

A shear energy minimization routine is used to determine the draping mesh when the shear energy used in the minimization routine is the sum of the shear energy of every mode with the respective weighting:

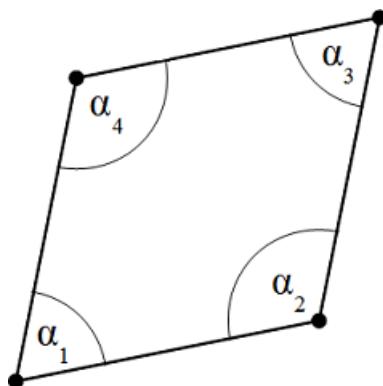
$$E = d_0 \cdot E_0 + d_1 \cdot E_1 + d_2 \cdot E_2 \quad (5.5)$$

The shear energy for the pure shear mode, E_0 , uses the expression in Equation 3. The energy formulations for the parallel sliding modes, E_1 and E_2 are a derivation of this expression. They introduce a directional bias into the draping algorithm:

$$E_0 = \sum_{i=1}^4 \cos^2 \alpha_i \quad (5.6)$$

$$E_1 = [\cos(\alpha_1) + \cos(\alpha_2)]^2 + [\cos(\alpha_3) + \cos(\alpha_4)]^2 \quad (5.7)$$

$$E_2 = [\cos(\alpha_1) + \cos(\alpha_3)]^2 + [\cos(\alpha_2) + \cos(\alpha_4)]^2 \quad (5.8)$$

Figure 5.4: Angle notation for the draping energy algorithm

The pure shear mode is active by default. It is recommended to use the default coefficient values (1, 0, 0) for regular woven fabrics. For UD fabrics or an irregular woven fabric, it is recommended to set the drapability by adjusting d_1 and d_2 in combination and setting d_0 to zero. The draping coefficients can be any positive real number or zero.

5.2.4. Limitations of Draping Simulations

The draping simulation approach has the following limitations:

- The surfaces to be draped must be smooth. No sharp edges are allowed.
- The draping procedure does not change mechanical properties and the thickness of the ply.
- It is assumed that fabric transverse direction is perpendicular to the principal direction 1.
- The fiber slippage is a phenomenon that takes place after the locking limit and is noticeable only at relatively high deformation levels. The fiber slippage is not considered in this draping approach.

5.3. Interlaminar Stresses

This section contains detailed background information on the evaluation of interlaminar stresses in ACP. A short overview of the stress, strain and failure analysis in general is shown in the section [Postprocessing](#).

5.3.1. Introduction

5.3.2. Interlaminar normal stresses

5.3.3. Transverse shear stresses

5.3.1. Introduction

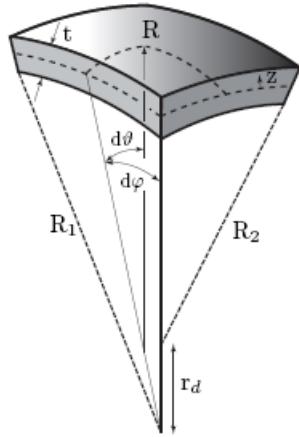
In the analysis layered composite structures, shell elements are widely used to keep the computational effort reasonable. In-plane stresses and even transverse shear stresses can be predicted with good accuracy using shells based on the first-order shear deformation theory (FSDT). However, in the analysis of thick-walled curved structures, interlaminar normal stresses (INS) can play a significant role. The normal stresses may affect the failure mode or even cause delamination failure. INS computation is not commonly available in shell element formulations, which leads to use of computationally expensive solid modeling instead.

The approach by Roos et al [14 (p. 351)] for INS computation of doubly curved laminate structures represents an alternative for solid modeling. The basis for the INS calculation is the displacement solution obtained from a shell based model. In conjunction with the INS approach, transverse shear stresses are computed with the approach presented by Rohwer and Rolfes [11 (p. 351)][12 (p. 351)]. When considered at layer interfaces, transverse shear stresses are referred to as interlaminar shear stresses (ISS).

5.3.2. Interlaminar normal stresses

5.3.2.1. Analytical model

A cylindrical coordinate system is used for describing an arbitrary doubly curved shell [9 (p. 351)]. The curved shell geometry, illustrated in Figure [Doubly curved FE geometry](#), is described by the coordinates (r, φ, ϑ) and it is subdivided into angular segments with the apex angles d_φ and d_ϑ and constant curvature radii of the centerline R_1 and R_2 .

Figure 5.5: Doubly curved FE geometry

The radial equilibrium equation becomes

$$\sigma_{r,r} + \frac{1}{r}\sigma_{\varphi r,\varphi} + \frac{1}{r+r_d}\sigma_{\vartheta r,\vartheta} + \frac{\sigma_r - \sigma_\varphi}{r} + \frac{\sigma_r - \sigma_\vartheta}{r+r_d} = 0 \quad (5.9)$$

where $r=R_1+z$, $r_d=R_2-R_1$ and $z=[-t/2, t/2]$. Each segment is embedded in between four cross-sections which are still assumed to remain straight and perpendicular to the midplane. Studies on singly curved plates [10 (p. 351)], [13 (p. 351)] show that the shear terms have only small effect and are thus neglected here. Equation 9 reduces to

$$\sigma_{r,r} + \frac{\sigma_r - \sigma_\varphi}{r} + \frac{\sigma_r - \sigma_\vartheta}{r+r_d} = 0 \quad (5.10)$$

where only direct stresses appear and the material law reduces to

$$\left\{ \begin{array}{c} \sigma_\varphi \\ \sigma_\vartheta \\ \sigma_r \end{array} \right\} = \left[\begin{array}{ccc} \bar{C}_{11} & \bar{C}_{12} & \bar{C}_{13} \\ \bar{C}_{21} & \bar{C}_{22} & \bar{C}_{23} \\ \bar{C}_{31} & \bar{C}_{32} & \bar{C}_{33} \end{array} \right] \left\{ \begin{array}{c} \varepsilon_\varphi - \varepsilon_\varphi^F \\ \varepsilon_\vartheta - \varepsilon_\vartheta^F \\ \varepsilon_r - \varepsilon_r^F \end{array} \right\} \quad (5.11)$$

where \bar{C}_{ij} are components of the 3D stiffness matrix expressed in reference coordinates which are parallel to the principal direction. The evaluation is described in Section [Reference Coordinates](#). The ε^F indicates free layer strains due to spatially constant changes of temperature T and moisture content H .

$$\varepsilon^F = \alpha \cdot \Delta T + \beta \cdot \Delta H \quad (5.12)$$

The goal is to express the direct strains in Equation 11 through the displacements u , v , and w . The kinematic relations in the modified coordinate system are

$$\begin{aligned} \varepsilon_\varphi &= \frac{1}{r} (w + u_{,\varphi}) \\ \varepsilon_\vartheta &= \frac{1}{r+r_d} (w + v_{,\vartheta}) \\ \varepsilon_r &= w_{,r} \end{aligned} \quad (5.13)$$

The in-plane deformations u and v are expressed by the laminate deformations ε_ϕ^0 and κ_9 which is analogous to the "ref.'CLT'. The through-the-thickness coordinate z is replaced with the radial coordinate r and the curvature radii of the midplane:

$$\begin{aligned} du &= R_1 \cdot d\varphi \left(\varepsilon_\phi^0 + (r - R_1) \kappa_\phi \right) \\ dv &= R_2 \cdot d\vartheta \left(\varepsilon_9^0 + (r - R_2 + r_d) \kappa_9 \right) \end{aligned} \quad (5.14)$$

The direct strains become

$$\begin{aligned} \varepsilon_\phi &= \frac{1}{r} \left(w + R_1 \left(\varepsilon_\phi^0 + (r - R_1) \kappa_\phi \right) \right) \\ \varepsilon_9 &= \frac{1}{r + r_d} \left(w + R_2 \left(\varepsilon_9^0 + (r - R_2 + r_d) \kappa_9 \right) \right) \\ \varepsilon_r &= w_{,r} \end{aligned} \quad (5.15)$$

The combination of the material law of Equation 11 with the kinematic relations of Equation 13 leads to the direct stresses expressed by the deformations:

$$\begin{aligned} \sigma_\phi &= \frac{\bar{C}_{11}}{r} \left(w + R_1 \left(\varepsilon_\phi^0 + (r - R_1) \kappa_\phi \right) \right) + \\ &\quad + \frac{\bar{C}_{12}}{r + r_d} \left(w + R_2 \left(\varepsilon_9^0 + (r - R_2 + r_d) \kappa_9 \right) \right) + \\ &\quad + \bar{C}_{13} \cdot w_{,r} - \bar{C}_{1i} \cdot \varepsilon_i^F \\ \sigma_9 &= \frac{\bar{C}_{21}}{r} \left(w + R_1 \left(\varepsilon_\phi^0 + (r - R_1) \kappa_\phi \right) \right) + \\ &\quad + \frac{\bar{C}_{22}}{r + r_d} \left(w + R_2 \left(\varepsilon_9^0 + (r - R_2 + r_d) \kappa_9 \right) \right) + \\ &\quad + \bar{C}_{23} \cdot w_{,r} - \bar{C}_{2i} \cdot \varepsilon_i^F \\ \sigma_r &= \frac{\bar{C}_{31}}{r} \left(w + R_1 \left(\varepsilon_\phi^0 + (r - R_1) \kappa_\phi \right) \right) + \\ &\quad + \frac{\bar{C}_{32}}{r + r_d} \left(w + R_2 \left(\varepsilon_9^0 + (r - R_2 + r_d) \kappa_9 \right) \right) + \\ &\quad + \bar{C}_{33} \cdot w_{,r} - \bar{C}_{3i} \cdot \varepsilon_i^F \end{aligned} \quad (5.16)$$

Equation 16 is combined with the radial equilibrium Equation 10 and the differential equation of the through-the-thickness displacement is

$$\begin{aligned} 0 &= w_{,rr} + w_{,r} \left(\frac{1}{r} + \frac{1}{r + r_d} \right) + \\ &\quad + w \left(\frac{\hat{C}_{13} + \hat{C}_{23} - 2 \hat{C}_{12}}{r(r + r_d)} - \frac{\hat{C}_{11}}{r^2} - \frac{\hat{C}_{22}}{(r + r_d)^2} \right) + P^* \end{aligned} \quad (5.17)$$

where $\hat{C}_{ij} = \frac{\bar{C}_{ij}}{\bar{C}_{33}}$ and P^* is

$$\begin{aligned} P^* = & u_{,\varphi} \left(\frac{\hat{C}_{13} - \hat{C}_{12}}{r(r+r_d)} - \frac{\hat{C}_{11}}{r^2} \right) + u_{,\varphi r} \frac{\hat{C}_{13}}{r} + \\ & + v_{,\vartheta} \left(\frac{\hat{C}_{23} - \hat{C}_{12}}{r(r+r_d)} - \frac{\hat{C}_{22}}{(r+r_d)^2} \right) + v_{,\vartheta r} \frac{\hat{C}_{23}}{r+r_d} + \\ & + \varepsilon_i^F \left(\frac{\hat{C}_{1i} - \hat{C}_{3i}}{r} + \frac{\hat{C}_{2i} - \hat{C}_{3i}}{r+r_d} \right) \end{aligned} \quad (5.18)$$

5.3.2.2. Reference Coordinates

The reference coordinates, which are needed to evaluate the material stiffness C and strains are evaluated depending on the laminate properties or the curvature. If the laminate is non-isotropic, the reference coordinates are parallel to the principal laminate directions, where the first principal laminate stiffness has its maximum. In the case of a quasi-isotropic laminate, the reference coordinates are parallel to the principal curvature directions.

5.3.2.3. Numeric solution

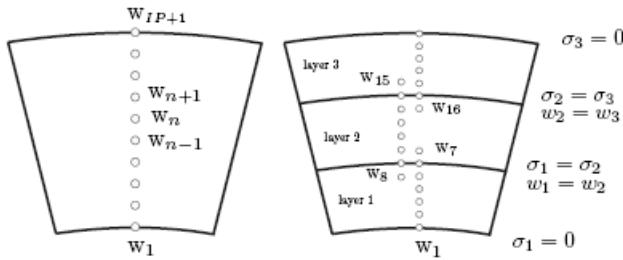
The solution of Equation 17 is found with the finite difference method. The differential equation represents a linear second order boundary value problem.

$$w_{,rr} + p(r) \cdot w_{,r} + q(r) \cdot w(r) = g(r) \quad (5.19)$$

where the derivatives are replaced with

$$\begin{aligned} w_{,r} &= \frac{w_{n+1} - w_{n-1}}{2d} \\ w_{,rr} &= \frac{w_{n+1} - 2w_n + w_{n-1}}{d^2} \end{aligned} \quad (5.20)$$

whereas w_n are the displacements at the supporting points through the thickness and d is the distance between two consecutive supporting points. Their placement scheme for a single-layer laminate is shown on the left of Figure [Integration scheme](#). The boundary conditions lead to a non-singular linear system of equations (LSoE) and are represented by the INS which have to vanish at the top and bottom surfaces of the laminate.

Figure 5.6: Integration scheme

Single layer (left) and multilayer laminate (right) where the indices 1, 2, and 3 count the layers and σ means σ_r .

The through-the-thickness INS distribution is obtained by combining Equations 11 and 13.

$$\sigma_r = \frac{\bar{C}_{13}}{r} (u_{,\varphi} + w) + \frac{\bar{C}_{23}}{r + r_d} (v_{,\theta} + w) + \bar{C}_{33} \cdot w_{,r} \quad (5.21)$$

This equation can be transformed to

$$w_{,r}(r) + \alpha \cdot w(r) = \beta \quad (5.22)$$

and is integrated in the LSoE whereas α and β modify the first and the last row of the left and right side of the LSoE, respectively.

Every additional layer leads to two more interface continuity conditions that have to be fulfilled:

$$\begin{aligned} w_{k+1}(z_k) &= w_k(z_k) \\ \sigma_{k+1}(z_k) &= \sigma_k(z_k) \end{aligned} \quad (5.23)$$

The first derivative of the through-the-thickness displacements w is found in the INS Equation 21. Additional supporting points, which are placed outside the layer, are necessary to evaluate the INS at the layer intersections. An integration scheme for a three layer laminate is plotted on the right of Figure [Integration scheme](#) where the supporting points $n = [7,8,15,16]$ guarantee the through-the-thickness continuity of the INS.

5.3.3. Transverse shear stresses

The method employed for computing transverse (interlaminar) shear stresses of FSDT based shell elements is based on the work by Rohwer and Rolfes [11 (p. 351)] [12 (p. 351)]. The transverse shear stresses are calculated from the three dimensional equilibrium equations of elasticity:

$$\begin{aligned} \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yz}}{\partial z} &= 0 \\ \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} &= 0 \end{aligned} \quad (5.24)$$

In order to calculate the shear stresses from the equilibrium equations, the in-plane stresses need to be derived first and then integrated with respect to the thickness coordinate:

$$\left\{ \begin{array}{c} \tau_{yz} \\ \tau_{zx} \end{array} \right\} = - \int \left\{ \begin{array}{c} \sigma_{y,y} + \tau_{xy,x} \\ \sigma_{x,x} + \tau_{xy,y} \end{array} \right\} dz \quad (5.25)$$

In-plane stresses are piecewise continuous functions, so the integration needs to be done in parts. Applying the stress-strain relation for in-plane stresses, Equation 21 takes the form

$$\begin{Bmatrix} \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = - \int \left\{ B_1 \bar{Q}^{(k)} \left(\varepsilon_{,x}^0 + z^{(k)} \kappa_{,x} \right) + B_2 \bar{Q}^{(k)} \left(\varepsilon_{,y}^0 + z^{(k)} \kappa_{,y} \right) \right\} dz \quad (5.26)$$

where

$$B_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.27)$$

Strain derivatives in Equation 26 can also be transformed to give an expression in terms of force derivatives by applying the constitutive equations. In order to calculate the stresses straight from the shear forces, some additional assumptions have to be made.

The influence of the in-plane force derivatives is neglected, that is

$$\frac{\partial}{\partial x} \{N\} = \{0\} \quad \frac{\partial}{\partial y} \{N\} = \{0\} \quad (5.28)$$

Strain derivatives then reduce to a form

$$\begin{aligned} \frac{\partial}{\partial x} \{\varepsilon^0\} &= [b] \cdot \frac{\partial}{\partial x} \{M\} & \frac{\partial}{\partial x} \{\kappa\} &= [d] \cdot \frac{\partial}{\partial x} \{M\} \\ \frac{\partial}{\partial y} \{\varepsilon^0\} &= [b] \cdot \frac{\partial}{\partial y} \{M\} & \frac{\partial}{\partial y} \{\kappa\} &= [d] \cdot \frac{\partial}{\partial y} \{M\} \end{aligned} \quad (5.29)$$

where [b] and [d] are the laminate compliance matrices.

The actual displacement fields are further simplified by assuming two separate cylindrical bending modes. The moment derivatives then reduce to the simple resultant shear forces:

$$\frac{\partial}{\partial x} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{Bmatrix} Q_x \\ 0 \\ 0 \end{Bmatrix} \quad \frac{\partial}{\partial y} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{Bmatrix} 0 \\ Q_x \\ 0 \end{Bmatrix} \quad (5.30)$$

Applying Equations 28 - 30 to Equation 26 yields

$$\begin{Bmatrix} \tau_{yz} \\ \tau_{zx} \end{Bmatrix}^{(k)} = [B_1] [F(z)]^{(k)} \begin{Bmatrix} Q_x \\ 0 \\ 0 \end{Bmatrix} + [B_2] [F(z)]^{(k)} \begin{Bmatrix} 0 \\ Q_x \\ 0 \end{Bmatrix} \quad (5.31)$$

This simplifies to

$$\begin{Bmatrix} \tau_{yz} \\ \tau_{zx} \end{Bmatrix}^{(k)} = \begin{bmatrix} F_{22} & F_{31} \\ F_{32} & F_{11} \end{bmatrix}^{(k)} \begin{Bmatrix} Q_y \\ Q_x \end{Bmatrix} \quad (5.32)$$

where F_{ij} are the components of the 3×3 matrix $F(z)^k$. The components of the $F(z)$ matrix are piecewise continuous, second order functions of z determined by

$$F(z)^{(k)} = -m(z)[b] - n(z)[d] \quad (5.33)$$

where

$$\begin{aligned} m(z) &= \int [\bar{\mathcal{Q}}]^{(k)} dz = M(k) + [\bar{\mathcal{Q}}]^{(k)} z \\ n(z) &= \int [\bar{\mathcal{Q}}]^{(k)} z dz = N(k) + \frac{1}{2} [\bar{\mathcal{Q}}]^{(k)} z^2 \end{aligned} \quad (5.34)$$

The functions $M(k)$ and $N(k)$ are defined by the continuity of stresses at the layer interfaces:

$$\begin{aligned} M(k) &= \sum_{i=1}^k [\bar{\mathcal{Q}}]^{(i)} (z_i - z_{i-1}) - [\bar{\mathcal{Q}}]^{(k)} z_k \\ N(k) &= \frac{1}{2} \sum_{i=1}^k [\bar{\mathcal{Q}}]^{(i)} \left(z_i^2 - z_{i-1}^2 \right) - \frac{1}{2} [\bar{\mathcal{Q}}]^{(k)} z_k^2 \end{aligned} \quad (5.35)$$

It can be seen that the functions $m(z)$ and $n(z)$ of Equation 34 give the laminate stiffness matrices [A] and [B] at the lower surface of the laminate and zero at the top surface of the laminate:

$$\begin{aligned} m(-h/2) &= n(-h/2) = 0 \\ m(h/2) &= [A] \\ n(h/2) &= [B] \end{aligned} \quad (5.36)$$

Therefore, the transverse shear stresses of Equation 32 become zero for the top and bottom surfaces and fulfill the boundary conditions that transverse shear stresses need to vanish at the laminate surfaces.

In the local coordinate system the transverse shear stresses are

$$\left\{ \begin{array}{c} \tau_{23} \\ \tau_{31} \end{array} \right\} = [T] \left\{ \begin{array}{c} \tau_{yz} \\ \tau_{zx} \end{array} \right\} \quad (5.37)$$

where the 2×2 transformation matrix $[T]$ is

$$[T] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (5.38)$$

5.4. Failure Analysis

This section contains detailed background information on the evaluation of interlaminar stresses in ACP. A short overview of the failure analysis in general is shown in the section [Postprocessing](#).

[5.4.1. Reserve factor](#)

[5.4.2. Weighting factors](#)

[5.4.3. Failure Criterion Function](#)

[5.4.4. Failure Criteria for Reinforced Materials](#)

[5.4.5. Sandwich Failure](#)

[5.4.6. Interlaminar failure](#)

[5.4.7. Isotropic material failure](#)

[5.4.8. Failure Criteria vs. Ply Type Table](#)

5.4.1. Reserve factor

The reserve factor (RF) indicates margin to failure. The applied load multiplied by the reserve factor gives the failure load:

$$RF \times F_{applied} = F_f \quad (5.39)$$

Reserve factor values greater than one indicate positive margin to failure and values less than one indicate negative margin. The values of reserve factors are always greater than zero.

The critical values of reserve factors lie between zero and one, whereas the non-critical values range from one to infinity. Whether the results are shown in numeric form or as contour plots, the non-critical values tend to be emphasized in comparison to critical values. Therefore, the inverse reserve factor (*IRF*) is often preferred in practical use:

$$IRF = \frac{1}{RF} \quad (5.40)$$

The non-critical values of *IRF* range from zero to one and the critical values from that on.

The margin of safety (*MoS*) is an alternative for the reserve factor in indicating margin to failure. The margin of safety is obtained from the corresponding reserve factor with the relation

$$MoS = RF - 1 \quad (5.41)$$

A positive margin of safety indicates the relative amount that the applied load can be increased before reaching failure load. Correspondingly, a negative margin of safety indicates how much the applied load should be decreased. Margins of safety are typically expressed as percentages.

5.4.2. Weighting factors

Several failure criteria implemented in ACP are able to predict the failure mode - whether fiber or matrix failure occurs first for example. There is the possibility to assign weighting factors to individual failure mode criteria and thus create a certain bias towards or against a specific failure mode. As a result, the failure criteria can be tuned to specific requirements. A design may require a higher safety against delamination than against fiber failure and the weighting factor of delamination is increased.

Using the maximum stress criterion given in (44) as an example it is shown how the weighting factors are implemented in ACP.

$$f = \max \left(\left| W_1 \cdot \frac{\sigma_1}{X} \right|, \left| W_2 \cdot \frac{\sigma_2}{Y} \right|, \left| W_3 \cdot \frac{\sigma_3}{Z} \right|, \left| W_{12} \cdot \frac{\tau_{12}}{S} \right|, \left| W_{13} \cdot \frac{\tau_{13}}{R} \right|, \left| W_{23} \cdot \frac{\tau_{23}}{Q} \right| \right) \quad (5.42)$$

5.4.3. Failure Criterion Function

The strength of materials and material systems under multiaxial loads can be predicted based on different failure criteria. Failure criteria relate the material strength allowables, defined for uniaxial tension-compression and shear, to the general stress-strain state due to multiaxial loads. Typically failure criteria are presented as mathematical expressions called failure criterion functions (*f*):

$$f = \text{function(stresses (or strains), material strength)} \quad (5.43)$$

where $f \geq 1$ indicates failure.

The values of failure criterion functions change with load similarly as the inverse reserve factor (values below one are non-critical and one indicates failure). However, the values are generally not equal except at the failure point. Only the reserve factor *RF* (*IRF*, *MoS*) tells the actual distance to the failure point from the point represented by the applied load. Typically a numeric line search method is used for determining the value of *RF* (*IRF*, *MoS*) based on the selected failure criterion, stresses and strains due to the applied load, and material strength allowables.

5.4.4. Failure Criteria for Reinforced Materials

This section contains the following information:

- 5.4.4.1. Maximum Strain Criterion
- 5.4.4.2. Maximum Stress Criterion
- 5.4.4.3. Quadratic Failure Criteria
- 5.4.4.4. Hashin Failure Criterion
- 5.4.4.5. Puck Failure Criteria
- 5.4.4.6. LaRC Failure Criterion
- 5.4.4.7. Cuntze's Failure Criterion

5.4.4.1. Maximum Strain Criterion

In the maximum strain criterion, the ratios of the actual strains to the failure strains are compared in the ply principal coordinate system. The failure criterion function is written as

$$f = \max \left(\left| \frac{\varepsilon_1}{X_\varepsilon} \right|, \left| \frac{\varepsilon_2}{Y_\varepsilon} \right|, \left| \frac{\gamma_{12}}{S_\varepsilon} \right| \right) \quad (5.44)$$

$$\begin{aligned} \varepsilon_1 \geq 0 \Rightarrow X_\varepsilon &= X_{\varepsilon t} & ; \varepsilon_1 < 0 \Rightarrow X_\varepsilon &= X_{\varepsilon c} \\ \varepsilon_2 \geq 0 \Rightarrow Y_\varepsilon &= Y_{\varepsilon t} & ; \varepsilon_2 < 0 \Rightarrow Y_\varepsilon &= Y_{\varepsilon c} \end{aligned} \quad (5.45)$$

For isotropic plies and for transversely isotropic plies with the plane of isotropy 12, such as mat plies, the principal strains and the maximum shear strain have to be determined first. The principal strains are

$$\varepsilon_{\xi, \eta} = \frac{1}{E} \left\{ \frac{1}{2} (1-\nu) (\sigma_1 + \sigma_2) \pm (1+\nu) \left[\frac{1}{4} (\sigma_1 - \sigma_2)^2 + \tau_{12}^2 \right]^{\frac{1}{2}} \right\} \quad (5.46)$$

and the maximum shear strain is

$$\gamma_{\max} = \frac{2(1+\nu)}{E} \left[\frac{1}{4} (\sigma_1 - \sigma_2)^2 + \tau_{12}^2 \right]^{\frac{1}{2}} \quad (5.47)$$

The value of the failure criterion function is then computed according to Equations 44 - 45 by replacing ε_1 , ε_2 and γ_{12} with ε_ξ , ε_η and γ_{\max} , respectively. Due to the transverse isotropy, failure strains X_ε and Y_ε are equal.

5.4.4.2. Maximum Stress Criterion

In the maximum stress criterion, the ratios of the actual stresses to the failure stresses are compared in the ply principal coordinate system. Thus, the failure criterion function is

$$f = \max \left(\left| \frac{\sigma_1}{X} \right|, \left| \frac{\sigma_2}{Y} \right|, \left| \frac{\sigma_3}{Z} \right|, \left| \frac{\tau_{12}}{S} \right|, \left| \frac{\tau_{13}}{R} \right|, \left| \frac{\tau_{23}}{Q} \right| \right) \quad (5.48)$$

$$\begin{aligned} \sigma_1 \geq 0 \Rightarrow X &= X_t & ; \sigma_1 < 0 \Rightarrow X &= X_c \\ \sigma_2 \geq 0 \Rightarrow Y &= Y_t & ; \sigma_2 < 0 \Rightarrow Y &= Y_c \\ \sigma_3 \geq 0 \Rightarrow Z &= Z_t & ; \sigma_3 < 0 \Rightarrow Z &= Z_c \end{aligned} \quad (5.49)$$

As in the maximum strain criterion, isotropic plies and transversely isotropic plies with the plane of isotropy 12 have to be considered separately. In plane stress state the principal stresses are

$$\sigma_{\xi,\eta} = \frac{1}{2}(\sigma_1 + \sigma_2) \pm \left[\frac{1}{4}(\sigma_1 - \sigma_2)^2 + \tau_{12}^2 \right]^{\frac{1}{2}} \quad (5.50)$$

and the maximum shear stress is

$$\tau_{\max} = \left[\frac{1}{4}(\sigma_1 - \sigma_2)^2 + \tau_{12}^2 \right]^{\frac{1}{2}} \quad (5.51)$$

The value of the failure criterion function is then computed according to Equations 44 - 45 by replacing σ_1 , σ_2 and τ_{12} with σ_{ξ} , σ_{η} , and τ_{\max} , respectively. Due to the transverse isotropy, the failure strains X and Y are equal.

When the 3D maximum stress criterion ($\sigma_3 = 0$) is applied to isotropic plies, the principal stresses are solved first and the failure function is

$$f = \max \left(\left| \frac{\sigma_{\xi}}{X_i} \right|, \left| \frac{\sigma_{\eta}}{X_i} \right|, \left| \frac{\sigma_{\zeta}}{X_i} \right|, \left| \frac{\tau_{\max}}{S} \right| \right); \begin{array}{l} i=t \text{ when } \sigma_j \geq 0 \\ i=c \text{ when } \sigma_j < 0 \end{array}, j=\xi, \eta, \zeta \quad (5.52)$$

where

$$\tau_{\max} = \frac{1}{2}(\sigma_{\xi} - \sigma_{\zeta}) \quad (5.53)$$

Here σ_{ξ} is the most positive and σ_{ζ} is the most negative principal stress.

5.4.4.3. Quadratic Failure Criteria

In quadratic criteria all the stress or strain components are combined into one expression. Many commonly used criteria for fiber-reinforced composites belong to a subset of fully interactive criteria called quadratic criteria. The general form of quadratic criteria can be expressed as a second-degree polynomial.

$$f = F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + F_{33}\sigma_3^2 + F_{44}\tau_{23}^2 + F_{55}\tau_{13}^2 + F_{66}\tau_{12}^2 + 2F_{12}\sigma_1\sigma_2 + 2F_{23}\sigma_2\sigma_3 + 2F_{13}\sigma_1\sigma_3 + F_1\sigma_1 + F_2\sigma_2 + F_3\sigma_3 \quad (5.54)$$

In plane stress-state ($s_3=0$), the polynomial reduces to the simpler form

$$f = F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + F_{66}\tau_{12}^2 + 2F_{12}\sigma_1\sigma_2 + F_1\sigma_1 + F_2\sigma_2 \quad (5.55)$$

The quadratic failure criteria in ACP - Tsai-Wu, Tsai-Hill and Hoffman - differ in how the coefficients F_{ii} and F_i are defined. Generally, the coefficients F_{ii} and F_i are determined so that the value of the failure criterion function corresponds to the material strength when a unidirectional stress state is present. However, not all coefficients cannot be determined in this way.

5.4.4.3.1. Tsai-Wu Failure Criterion

For the plane stress-state the Tsai-Wu criterion coefficients F have the values

$$\begin{aligned}
 F_{11} &= \frac{1}{X_t X_c} & F_1 &= \frac{1}{X_t} - \frac{1}{X_c} \\
 F_{22} &= \frac{1}{Y_t Y_c} & F_2 &= \frac{1}{Y_t} - \frac{1}{Y_c} \\
 F_{44} &= \frac{1}{Q^2} & F_{55} &= \frac{1}{R^2} \\
 F_{66} &= \frac{1}{S^2}
 \end{aligned} \tag{5.56}$$

Thus, the criterion can be written as

$$f = \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\tau_{23}^2}{Q^2} + \frac{\tau_{13}^2}{R^2} + \frac{\tau_{12}^2}{S^2} + \left(\frac{1}{X_t} - \frac{1}{X_c} \right) \sigma_1 + \left(\frac{1}{Y_t} - \frac{1}{Y_c} \right) \sigma_2 + 2F_{12}\sigma_1\sigma_2 \tag{5.57}$$

The coefficient F_{12} cannot be obtained directly from the failure stresses of uniaxial load cases. For accurate results it should be determined through biaxial load tests. In practice, it is often given in the form of a non-dimensional interaction coefficient

$$F_{12}^* = \frac{F_{12}}{\sqrt{(F_{11}F_{22})}} \tag{5.58}$$

To insure that the criterion represents a closed conical failure surface, the value of F_{12}^* has to be within the range $-1 < F_{12}^* < 1$. However, the value range for physically meaningful material behavior is more limited. The often used value $-\frac{1}{2}$ corresponds to a "generalized Von Mises criterion". The final Tsai-Wu constant $2 F_{12}^*$ becomes -1 as used in the Equation 61. Similarly $2 F_{13}, 2 F_{23}$ can be dealt with using the corresponding values F_{13}^* and F_{23}^* , which leads to the Tsai-Wu 3D expression:

Similarly $2 F_{13}, 2 F_{23}$ can be dealt with using the corresponding values F_{13}^* and F_{23}^* , which leads to the Tsai-Wu 3D expression:

$$\begin{aligned}
 f &= \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\sigma_3^2}{Z_t Z_c} + \frac{\tau_{12}^2}{S_{xy}^2} + \frac{\tau_{13}^2}{S_{xz}^2} + \frac{\tau_{23}^2}{S_{yz}^2} \\
 &- 1.0 \frac{\sigma_1 \sigma_2}{\sqrt{X_t X_c Y_t Y_c}} - 1.0 \frac{\sigma_2 \sigma_3}{\sqrt{Y_t Y_c Z_t Z_c}} - 1.0 \frac{\sigma_1 \sigma_3}{\sqrt{X_t X_c Z_t Z_c}} \\
 &+ \sigma_1 \left(\frac{1}{X_t} - \frac{1}{X_c} \right) + \sigma_2 \left(\frac{1}{Y_t} - \frac{1}{Y_c} \right) + \sigma_3 \left(\frac{1}{Z_t} - \frac{1}{Z_c} \right).
 \end{aligned} \tag{5.59}$$

Note: In ACP the Tsai-Wu constants are:

- $2 F_{12} = XY$, default -1
- $2 F_{13} = XZ$, default -1
- $2 F_{23} = YZ$, default -1

5.4.4.3.2. Tsai-Hill Failure Criterion

In the Tsai-Hill criterion, either tensile or compressive strengths are used for determining the coefficients F depending on the loading condition. The coefficients are

$$\begin{aligned} F_{11} &= \frac{1}{X^2} & F_1 = 0 & F_{12} = -\frac{1}{2X^2} \\ F_{22} &= \frac{1}{Y^2} & F_2 = 0 & \\ F_{44} &= \frac{1}{Q^2} & F_{55} = \frac{1}{R^2} & F_{66} = \frac{1}{S^2} \end{aligned} \quad (5.60)$$

where the values of X and Y are

$$\begin{aligned} \sigma_1 \geq 0 \Rightarrow X &= X_t \quad ; \sigma_1 < 0 \Rightarrow X = X_c \\ \sigma_2 \geq 0 \Rightarrow Y &= Y_t \quad ; \sigma_2 < 0 \Rightarrow Y = Y_c \end{aligned} \quad (5.61)$$

Hence, the Tsai-Hill failure criterion function can be written in the form

$$f = \left(\frac{\sigma_1}{X} \right)^2 + \left(\frac{\sigma_2}{Y} \right)^2 + \left(\frac{\tau_{23}}{Q} \right)^2 + \left(\frac{\tau_{13}}{R} \right)^2 + \left(\frac{\tau_{12}}{S} \right)^2 - \frac{\sigma_1 \sigma_2}{X^2} \quad (5.62)$$

For the full 3D case another formulation can be used as in [23 (p. 352)].

$$\begin{aligned} (G+H)\sigma_1^2 + (F+H)\sigma_2^2 + (F+G)\sigma_3^2 - 2H\sigma_1\sigma_2 - 2G\sigma_1\sigma_3 - 2F\sigma_2\sigma_3 \dots \\ + 2L\tau_{23}^2 + 2M\tau_{13}^2 + 2N\tau_{12}^2 = 1 \end{aligned} \quad (5.63)$$

where

$$\begin{aligned} F &= \frac{1}{2} \left(-\frac{1}{X^2} + \frac{1}{Y^2} + \frac{1}{Z^2} \right) \\ G &= \frac{1}{2} \left(\frac{1}{X^2} - \frac{1}{Y^2} + \frac{1}{Z^2} \right) \\ H &= \frac{1}{2} \left(\frac{1}{X^2} + \frac{1}{Y^2} - \frac{1}{Z^2} \right) \end{aligned} \quad (5.64)$$

5.4.4.3.3. Hoffman Failure Criterion

The Hoffman criterion defines the biaxial coefficients F_{12} , F_{23} and F_{13} with the following material strength expressions for the 3D stress state:

$$F_{12} = \frac{1}{(X_t X_c)} + \frac{1}{(Y_t Y_c)} - \frac{1}{(Z_t Z_c)} \quad (5.65)$$

$$F_{13} = \frac{1}{(Z_t Z_c)} + \frac{1}{(X_t X_c)} - \frac{1}{(Y_t Y_c)} \quad (5.66)$$

$$F_{23} = \frac{1}{(Y_t Y_c)} + \frac{1}{(Z_t Z_c)} - \frac{1}{(X_t X_c)} \quad (5.67)$$

The Hoffman failure criterion for a 3D stress state can be written as

$$f = \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\sigma_3^2}{Z_t Z_c} + \frac{\tau_{12}^2}{S_{xy}^2} + \frac{\tau_{13}^2}{S_{xz}^2} + \frac{\tau_{23}^2}{S_{yz}^2} - F_{12}\sigma_1\sigma_2 - F_{23}\sigma_2\sigma_3 - F_{13}\sigma_1\sigma_3 + \frac{\sigma_1}{X_t X_c} + \frac{\sigma_2}{Y_t Y_c} + \frac{\sigma_3}{Z_t Z_c} \quad (5.68)$$

The biaxial coefficient F_{12} for the plane stress state reduces to

$$F_{12} = \frac{1}{(X_t X_c)} \quad (5.69)$$

The entire Hoffman criterion in the plane stress case reduces to

$$f = \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\tau_{12}^2}{S_{xy}^2} - F_{12}\sigma_1\sigma_2 + \frac{\sigma_1}{X_t X_c} + \frac{\sigma_2}{Y_t Y_c} \quad (5.70)$$

5.4.4.4. Hashin Failure Criterion

In the Hashin criterion, criticality of tensile loads in the fiber direction is predicted with the expression

$$2D: f_f = \left(\frac{\sigma_1}{X_t} \right)^2 + \left(\frac{\tau_{12}}{S} \right)^2, \quad \sigma_1 \geq 0 \quad (5.71)$$

$$3D: f_f = \left(\frac{\sigma_1}{X_t} \right)^2 + \left(\frac{\tau_{12}}{S} \right)^2 + \left(\frac{\tau_{13}}{R} \right)^2, \quad \sigma_1 \geq 0 \quad (5.72)$$

Under compressive loads in the fiber direction, failure is predicted with an independent stress condition

$$f_f = -\frac{\sigma_1}{X_c}, \quad \sigma_1 < 0 \quad (5.73)$$

In the case of tensile transverse stress, the expression for predicting matrix failure is

$$2D: f_m = \left(\frac{\sigma_2}{Y_t} \right)^2 + \left(\frac{\tau_{12}}{S} \right)^2 +, \quad \sigma_2 \geq 0 \quad (5.74)$$

$$3D: f_m = \left(\frac{\sigma_2}{Y_t} \right)^2 + \left(\frac{\tau_{23}}{Q} \right)^2 + \left(\frac{\tau_{12}}{S} \right)^2 + \left(\frac{\tau_{13}}{R} \right)^2, \quad \sigma_2 \geq 0 \quad (5.75)$$

A more complex expression is used when the transverse stress is compressive:

$$2D: f_m = \left(\frac{\sigma_2}{2S} \right)^2 + \left(\frac{\tau_{12}}{S} \right)^2 + \left[\left(\frac{Y_c}{2S} \right)^2 - 1 \right] \frac{\sigma_2}{Y_c}, \quad \sigma_2 < 0 \quad (5.76)$$

$$3D: f_m = \left(\frac{\sigma_2}{2Q} \right)^2 + \left(\frac{\tau_{23}}{Q} \right)^2 + \frac{\tau_{12}^2}{S^2} + \left[\left(\frac{Y_c}{2Q} \right)^2 - 1 \right] \frac{\sigma_2}{Y_c}, \quad \sigma_2 < 0 \quad (5.77)$$

Delamination (tension and compression) is predicted with this expression:

$$3D: f_d = \left(\frac{\sigma_3}{Z} \right)^2 + \left(\frac{\tau_{13}}{R} \right)^2 + \left(\frac{\tau_{23}}{Q} \right)^2 \quad (5.78)$$

The most critical of the failure modes is selected:

$$f = \max(f_f, f_m, f_d) \quad (5.79)$$

5.4.4.5. Puck Failure Criteria

5.4.4.5.1. Simple and Modified Puck Criterion

The two elder Puck failure criterion formulations are the so called simple Puck and the so called modified Puck. Both criterions considers failure due to longitudinal loads and matrix failure mode due to transverse and shear loads separately [25 (p. 352)] [26 (p. 352)]

For both, simple and modified Puck criterion, failure in fiber direction is calculated the same way as in the maximum stress criterion:

$$f_f = \left| \frac{\sigma_1}{X} \right| \quad (5.80)$$

The matrix failure is calculated differently for each formulation as illustrated in Equation 78 for simple Puck. As shown in Equation 79 tensile or compressive failure stresses are used depending on the stress state.

$$f_m = \left(\frac{\sigma_2}{Y} \right)^2 + \left(\frac{\tau_{12}}{S} \right)^2 \quad (5.81)$$

$$\begin{aligned} \sigma_1 \geq 0 \Rightarrow X &= X_t \quad ; \sigma_1 < 0 \Rightarrow X = X_c \\ \sigma_2 \geq 0 \Rightarrow Y &= Y_t \quad ; \sigma_2 < 0 \Rightarrow Y = Y_c \end{aligned} \quad (5.82)$$

The modified Puck criterion differs from the latter one only in the formulation for matrix failure:

$$f_m = \frac{\sigma_2^2}{Y_t Y_c} + \frac{\tau_{12}^2}{S^2} + \left(\frac{1}{Y_t} + \frac{1}{Y_c} \right) \sigma_2 \quad (5.83)$$

Just like in [Hashin Failure Criterion](#) the failure occurs for either f_f or f_m reaching the value one, so the failure criterion function is

$$f = \max(f_f, f_m) \quad (5.84)$$

Despite being called 'simple' in the failure criteria configuration in the ACP "Failure Criteria Definition"- dialog for Puck the modified version is actually implemented and the name is referring to the simplicity of that criterion in comparison to [Puck's action plane strength criterion](#).

5.4.4.5.2. Puck's action plane strength criterion

5.4.4.5.2.1. Fiber Failure (FF)

As is simple Puck one option for evaluating fiber failure is to use the maximum stress criterion for that case [27 (p. 352)], [28 (p. 352)], [29 (p. 352)]

$$\frac{\sigma_1}{X_t} = 1 \quad \text{for } \sigma_1 > 0 \quad \text{or} \quad \frac{\sigma_1}{X_c} = 1 \quad \text{for } \sigma_1 < 0 \quad (5.85)$$

and similarly a maximum strain criterion.

$$\frac{\varepsilon_1}{X_{et}} = 1 \quad \text{for } \varepsilon_1 > 0 \quad \text{or} \quad \frac{\varepsilon_1}{X_{ec}} = 1 \quad \text{for } \varepsilon_1 < 0 \quad (5.86)$$

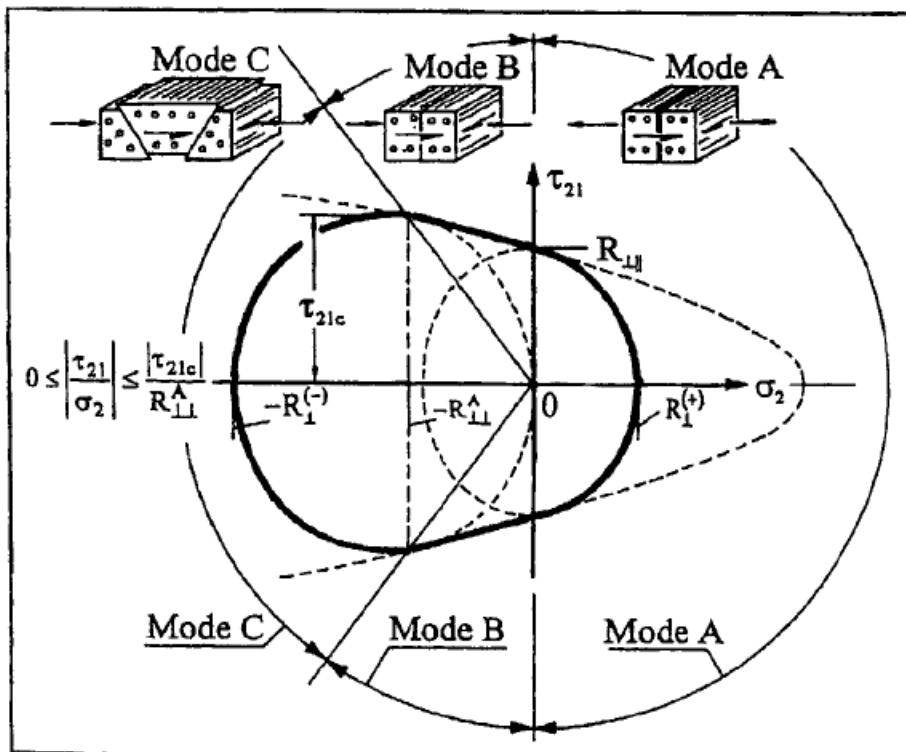
A more complicated version for FF criterion was as well presented by Puck for the World Wide Failure Exercise, but the maximum stress criterion is considered sufficient for the case of FF.

5.4.4.5.2.2. Inter-fiber failure (IFF)

Plane stress-state

Inter-fiber failure, or inter-fiber fracture according to [27 (p. 352)], [28 (p. 352)] can be explained in the cutting plane for which the principal stress σ_1 of an UD layer.

Figure 5.7: Fracture curve in σ_1 , τ_{21} space for $\sigma_1 = 0$. Three different fracture modes A, B, C are distinguished [28 (p. 352)].



The curve consists of two ellipses (modes A and C) and one parabola (mode B). Generally Puck's action plane strength criterion is formed utilizing the following 7 parameters,

$R_{\perp}^{(+)} , R_{\perp \parallel} , R_{\perp}^{(-)} , p_{\perp \parallel}^{(+)} , p_{\perp \parallel}^{(-)} , p_{\perp \perp}^{(+)} ,$ and $p_{\perp \perp}^{(-)}$, where R stands for fracture resistances and p for slope parameters of the fracture curves. The symbols \perp and \parallel denote the reference to direction parallel to the fibers and transverse (perpendicular) to the fibers. The values for $R_{\perp}^{(+)} \text{ and } R_{\perp}^{(-)}$ define the intersections of the curve with σ_2 -axis, as well as $R_{\perp \parallel}$ for the intersection with τ_{21} -axis. The slope parameters $p_{\perp \parallel}^{(+)} \text{ and } p_{\perp \parallel}^{(-)}$ are the inclinations in the latter intersections.

The failure conditions for IFF are:

$$\sqrt{\left(\frac{\tau_{21}}{R_{\perp \parallel}}\right)^2 + \left(1 - p_{\perp \parallel}^{(+)} \frac{R_{\perp}^{(+)}}{R_{\perp \parallel}}\right)^2 \left(\frac{\sigma_2}{R_{\perp}^{(+)}}\right)^2} + p_{\perp \parallel}^{(+)} \frac{\sigma_2}{R_{\perp \parallel}} = 1 \quad (5.87)$$

for mode A ($\sigma_2 \geq 0$)

$$\frac{1}{R_{\perp\parallel}} \left(\sqrt{\tau_{12}^2 + (p_{\perp\parallel}^{(-)} \sigma_2)^2} + p_{\perp\parallel}^{(-)} \sigma_2 \right) = 1 \quad (5.88)$$

for mode B $\left(\sigma_2 < 0 \text{ and } 0 \leq \left| \frac{\sigma_2}{\tau_{21}} \right| \leq \frac{R_{\perp\perp}^A}{|\tau_{21c}|} \right)$

The superscript A denotes that the fracture resistance belongs to the action plane.

$$\left[\left(\frac{\tau_{21}}{2(1+p_{\perp\perp}^{(-)})R_{\perp\parallel}} \right)^2 + \left(\frac{\sigma_2}{R_{\perp}^{(-)}} \right)^2 \right] \frac{R_{\perp}^{(-)}}{(-\sigma_2)} = 1 \quad (5.89)$$

for mode C $\left(\sigma_2 < 0 \text{ and } 0 \leq \left| \frac{\tau_{21}}{\sigma_2} \right| \leq \frac{|\tau_{21c}|}{R_{\perp\perp}^A} \right)$

The assumption

$$p_{\perp\perp}^{(-)} = p_{\perp\parallel}^{(-)} \frac{R_{\perp\perp}^A}{R_{\perp\parallel}} \quad (5.90)$$

is valid here and leads to

$$R_{\perp\perp}^A = \frac{R_{\perp\parallel}}{2p_{\perp\parallel}^{(-)}} \left(\sqrt{1 + 2p_{\perp\parallel}^{(-)} \frac{R_{\perp}^{(-)}}{R_{\perp\parallel}}} - 1 \right) \quad (5.91)$$

Also Equation 88 is valid.

$$\tau_{21c} = R_{\perp\parallel} \sqrt{1 + 2p_{\perp\perp}^{(-)}} \quad (5.92)$$

As the failure criterion functions and the functions for their corresponding stress exposure factors f_E are the same, they can be written as follows given the Equations 87 and 88 .

$$f_{E, \text{Mode A}} = \frac{1}{R_{\perp\parallel}} \left(\sqrt{\left(\frac{R_{\perp\parallel}}{R_{\perp}^{(+)}} - p_{\perp\parallel}^{(+)} \right)^2 \sigma_2^2 + \tau_{21}^2} + p_{\perp\parallel}^{(+)} \sigma_2 \right) \quad (5.93)$$

$$f_{E, \text{Mode B}} = \frac{1}{R_{\perp\parallel}} \left(\sqrt{\tau_{21}^2 + (p_{\perp\parallel}^{(-)} \sigma_2)^2} + p_{\perp\parallel}^{(-)} \sigma_2 \right) \quad (5.94)$$

$$f_{E, \text{Mode C}} = \frac{\tau_{21}^2}{4R_{\perp\parallel}^2 (1 + p_{\perp\parallel}^{(-)})^2} \frac{R_{\perp}^{(-)}}{(-\sigma_2)} + \frac{(-\sigma_2)}{R_{\perp}^{(-)}} \quad (5.95)$$

3D stress-state

While the latter formulations have been a reduced case working in $(\sigma_2 - T_{21})$ -stress space, the 3D stress-state can be described with Equations 92 and 93

$$\sigma_n \geq 0 : f_E = \sqrt{\left[\left(\frac{1}{R_{\perp}^{(+)}} - \frac{p_{\perp\psi}^{(+)}}{R_{\perp\psi}^A} \right) \sigma_n \right]^2 + \left(\frac{\tau_{nt}}{R_{\perp\perp}^A} \right)^2 + \left(\frac{\tau_{n1}}{R_{\perp\parallel}^A} \right)^2 + \frac{p_{\perp\psi}^{(+)}}{R_{\perp\psi}^A} \sigma_n} \quad (5.96)$$

$$\sigma \leq 0 : f_E = \sqrt{\left(\frac{\tau_{nt}}{R_{\perp\perp}^A} \right)^2 + \left(\frac{\tau_{n1}}{R_{\perp\parallel}^A} \right)^2 + \left(\frac{p_{\perp\parallel}^{(-)}}{R_{\perp\psi}^A} \sigma_n \right)^2 + \frac{p_{\perp\psi}^{(-)}}{R_{\perp\psi}^A} \sigma_n} \quad (5.97)$$

where

$$\cos^2 \psi = 1 - \sin^2 \psi = \frac{\tau_{nt}^2}{\tau_{nt}^2 + \tau_{n1}^2} \quad (5.98)$$

$$\frac{p_{\perp\psi}^{(\pm)}}{R_{\perp\psi}^A} = \frac{p_{\perp\perp}^{(\pm)}}{R_{\perp\perp}^A} \cos^2 \psi + \frac{p_{\perp\parallel}^{(\pm)}}{R_{\perp\parallel}^A} \sin^2 \psi \quad (5.99)$$

$$R_{\perp\perp}^A = \frac{R_{\perp}^{(-)}}{2(1 + p_{\perp\perp}^{(-)})} \quad (5.100)$$

As can be seen from the equations above the failure criterion function is formulated in the fracture (action) plane using the corresponding stresses and strains. The formulations for the stresses σ_n , T_{nt} and T_{n1} in an arbitrary plane with the inclination angle θ are

$$\sigma_n = \sigma_2 \cos^2 \theta + \sigma_3 \sin^2 \theta + 2\tau_{23} \sin \theta \cos \theta \quad (5.101)$$

$$\tau_{nt} = (\sigma_3 - \sigma_2) \sin \theta \cos \theta + \tau_{23} (\cos^2 \theta - \sin^2 \theta) \quad (5.102)$$

$$\tau_{n1} = \tau_{31} \sin \theta + \tau_{21} \cos \theta \quad (5.103)$$

To find the stress exposure factor f_E one has to iterate the angle θ to find the global maximum, as the failure will occur for that angle. An analytical solution for the fracture angle is only available for plane stress-state by assuming

$$p_{\perp\perp}^{(-)} / R_{\perp\perp}^A = p_{\perp\parallel}^{(-)} / R_{\perp\parallel}^A, \quad (5.104)$$

which leads to formulations for the exposure factor

$$f_E(\theta) = \begin{cases} \arccos \left(\sqrt{\frac{-R_{\perp\perp}^A}{-\sigma_2}} \right) & , \text{for } \sigma_2 < -R_{\perp\perp}^A \\ 0 & , \text{for } \sigma_2 \geq -R_{\perp\perp}^A \end{cases} \quad (5.105)$$

Puck illustrated in [27 (p. 352)] that the latter criterion can be used as a criterion to determine delamination, if an additional weakening factor for the interface $f_w^{(if)} \approx 0.8 \dots 0.9$ is applied, finally resulting in:

$$\frac{1}{f_w^{(If)}} \sqrt{\left[\left(\frac{1}{R_{\perp}^{(+)}} - \frac{p_{\perp\psi}^{(+)}}{R_{\perp\psi}^A} \right) \sigma_3 \right] + \left(\frac{\tau_{32}}{R_{\perp\perp}^A} \right)^2 + \left(\frac{\tau_{31}}{R_{\perp|}^A} \right)^2 + \frac{p_{\perp\psi}^{(+)}}{R_{\perp\psi}^A} \sigma_3} = 1 \quad \text{for } \sigma_3 \geq 0 \quad (5.106)$$

$$\frac{1}{f_w^{(If)}} \sqrt{\left(\frac{\tau_{32}}{R_{\perp\perp}^A} \right)^2 + \left(\frac{\tau_{31}}{R_{\perp|}^A} \right)^2 + \left(\frac{p_{\perp\psi}^{(-)}}{R_{\perp\psi}^A} \sigma_3 \right)^2 + \frac{p_{\perp\psi}^{(-)}}{R_{\perp\psi}^A} \sigma_3} = 1 \quad \text{for } \sigma_3 < 0$$

The active failure mode depends on the fraction angle θ and the sign of σ_n . Delamination can occur if σ_n is positive and θ is 90 degree. The failure modes *PmB* and *PmC* happen just in combination with negative σ_n .

Puck Constants

Different default values for the coefficients are set for carbon and glass fiber plies to:

Carbon

$$p_{\perp\parallel}^{(+)} = 0.35 \quad p_{\perp\parallel}^{(-)} = 0.3 \quad p_{\perp\perp}^{(+)} = 0.25 \quad p_{\perp\perp}^{(-)} = 0.2$$

Glass

$$p_{\perp\parallel}^{(+)} = 0.3 \quad p_{\perp\parallel}^{(-)} = 0.25 \quad p_{\perp\perp}^{(+)} = 0.2 \quad p_{\perp\perp}^{(-)} = 0.2$$

Those values are compliant with recommendations given in [30 (p. 352)].

Influence of fiber parallel stresses on inter-fiber failure

To take into account that some fibers might break already under uniaxial loads much lower than loads which cause ultimate failure (which can be seen as some kind of "degradation"), one can introduce weakening factors f_w for the strength parameters. Puck has formulated a power law relation in [27 (p. 352)]

$$f_w = 1 - \left(\frac{\sigma_1}{\sigma_{1d}} \right)^n \quad (5.107)$$

where $\sigma_{1d} = \xi X_t$ (or $-\xi X_c$, respectively); ζ and n can be experimentally determined.

Different approaches exist to handle that problem numerically. The function given in Equation 102 can be replaced by an elliptic function.

$$f_w = \begin{cases} \sqrt{1 - c(|\sigma| - sX)^2} & , \text{ for } |\sigma_1| > s \cdot X \\ 1 & , \text{ for } |\sigma_1| \leq s \cdot X \end{cases} \quad (5.108)$$

where

$$c = \frac{1 - M^2}{[(1 - s)X]^2}; \quad 0 \leq M < 1 \text{ and } 0 \leq s < 1. \quad (5.109)$$

where M and s are so called "degradation" parameters.

In ACP the stress exposure factor is calculated by intersecting the weakening factor ellipse with a straight line defined by the stress vector using the parameters

$$c_{\text{Line}} = \frac{f_{E0}}{(|\sigma_1|/X)} \quad a_{\text{Ellipse}} = \frac{(1-s)}{\sqrt{1-M^2}} \quad (5.110)$$

$$q = \frac{s}{(1 + (a_{\text{Ellipse}} c_{\text{Line}})^2)} + \sqrt{\left(\frac{s}{1 + (a_{\text{Ellipse}} c_{\text{Line}})^2} \right)^2 - \frac{s^2 - a_{\text{Ellipse}}^2}{1 + (a_{\text{Ellipse}} c_{\text{Line}})^2}} \quad (5.111)$$

$$f_E = \frac{|\sigma_1|/X}{q_F} \quad \text{for } \frac{|\sigma|}{f_{E0}} > s \cdot X \quad (5.112)$$

Otherwise the fiber failure criterion determines the stress exposure factor f_E .

Default values for the "degradation" parameters are M=0.5 s=0.5.

5.4.4.6. LaRC Failure Criterion

LaRC03 (2D) and LaRC04 (3D) are two sets of failure criteria for laminated fiber-reinforced composites. They are based on physical models for each failure mode and distinguish between fiber and matrix failure for different transverse fiber and matrix tension and compression modes. The LaRC criteria take into account that the apparent (in-situ) strength of an embedded ply, constrained by plies of different fiber orientations, is different compared to the same ply embedded in a UD laminate. Specifically, moderate transverse compression increases the apparent shear strength of a ply. Similarly in-plane shear significantly reduces the compressive strength of a ply. The evaluation of the in-situ strength also makes a distinction between thin and thick plies. The definition for a thick ply is a ply in which the slit crack is much smaller than the ply thickness. For epoxy E-glass and epoxy carbon laminates, the suggested threshold between thin and thick plies is 0.7 mm. [19 (p. 352)]_[24 (p. 352)]

The implemented LaRC04 (3D) failure criterion ACP assumes linear shear behavior and small angle deflection. The abbreviation LaRC stands for Langley Research Center.

5.4.4.6.1. LaRC03/LaRC04 Constants

The required unidirectional properties for the criteria are:

$$E_1, E_{21}, G_{12}, \nu_{12}, X^t, X_c, Y_t, Y_c, S^L, G_{Ic}, G_{IIc}$$

where S^L is the longitudinal shear strength, G_{Ic} & G_{IIc} is the fracture toughness for mode I & II while the other symbols have their usual meaning.

The following LaRC Constants are required for post-processing in ACP:

- Fracture Toughness Ratio: $g = \frac{G_{Ic}}{G_{IIc}}$ (Dimensionless)
- Fracture Toughness Mode I: G_{Ic} (Units: Force / Length)
- Fracture Toughness Mode II: G_{IIc} (Units: Force / Length)

- Fracture Angle under Compression: α_0 (Units: Degrees)
- Thin Ply Thickness Limit: (Units: Length)

The fracture angle can be determined in tests or taken to be $53 \pm 2^\circ$ which has proven to be a good results for carbon/epoxy and glass/epoxy laminates [28 (p. 352)]. The Thin Ply Thickness Limit is the only default value set for the LaRC parameters. The following reference values are drawn from literature [35 (p. 352)]:

Parameter	Typical Values
	Car-bon/epoxy
Elastic Modulus, E_1 [GPa]	128
Elastic Modulus, E_2, E_3 [GPa]	7.63
Fracture Angle α_0 [deg]	53
Fracture Toughness Mode 1 G_{Ic} [N/mm]	0.28
Fracture Toughness Mode 2 G_{IIc} [N/mm]	0.79
Fracture Toughness Ratio g	0.35
Thin Ply Thickness Limit [mm]	0.7

5.4.4.6.2. General Expressions

Several failure functions are involve the friction coefficients, in-situ strengths and fiber misalignment. As result, they are given at this point.

Friction Coefficients

Laminates tend not too fail in the plane of maximum shear stress. This fracture angle is typically. This is attributed to internal friction and consider in the LaRC failure criteria with two friction coefficients:

Transverse Friction Coefficient:

$$\eta^T = \frac{-1}{\tan(2\alpha_0)}$$

Longitudinal Friction Coefficient:

$$\eta^L = -\frac{S^L \cos(2\alpha_0)}{Y_c \cos^2(\alpha_0)}$$

In-Situ Ply Strength

The in-situ transverse direct strength and longitudinal shear strength for a thin ply are:

$$Y_{is}^T = \sqrt{\frac{8 \cdot G_{Ic}}{\pi \cdot t \cdot A_{22}}}$$

$$S_{is}^L = \sqrt{\frac{8G_{IIc}}{\pi t A_{44}}}$$

where (t) is the thickness of an embedded ply and

$$\Lambda_{22} = 2\left(\frac{1}{E_2} - \frac{v_{21}^2}{E_1}\right)$$

$$\Lambda_{44} = \frac{1}{G_{12}}$$

For a thick ply, the in-situ strengths are not a function of the ply thickness:

$$Y_{is}^T = 1.12\sqrt{2} Y_t$$

$$S_{is}^L = \sqrt{2} S_L$$

Fiber Misalignment Frame

Fiber compression, where the plies fail due to fiber kinking, is handled separately for transverse tension and transverse compression. In the model, imperfections in the fiber alignment are represented by regions of waviness, where transformed stresses can be calculated using a misalignment frame transforming the "original stresses". There are two different misalignment frames for LaRC03 (2D) and LaRC04 (3D).

LaRC03

For LaRC03, the stresses in the misaligned frame are computed as follows:

$$\sigma_1^m = \sigma_1 \cos^2(\phi) + \sigma_2 \sin^2(\phi) + 2\tau_{12} \sin(\phi) \cos(\phi) \quad (5.113)$$

$$\sigma_2^m = \sigma_1 \sin^2(\phi) + \sigma_2 \cos^2(\phi) - 2\tau_{12} \sin(\phi) \cos(\phi) \quad (5.114)$$

$$\tau_{12}^m = -\sigma_1 \sin(\phi) \cos(\phi) + \sigma_2 \sin(\phi) \cos(\phi) + \tau_{12} (\cos^2(\phi) - \sin^2(\phi)) \quad (5.115)$$

The misalignment angle for pure compression ϕ^C can be derived to 114 using $\sigma_1 = -X_c$ and $\sigma_2 = \tau_{12} = 0$ in the equations above as well as the stresses σ_2^m and τ_{12}^m the quadratic interaction criterion presented in Equation 123 for matrix compression.

$$\phi^C = \tan^{-1} \left(\frac{1 - \sqrt{1 - 4 \left(\frac{S_{is}^L}{X_c} + \eta_L \right) \left(\frac{S_{is}^L}{X_c} \right)}}{2 \left(\frac{S_{is}^L}{X_c} + \eta^L \right)} \right) \quad (5.116)$$

The total misalignment angle ϕ is calculated from:

$$\phi = \frac{|\tau_{12}| + (G_{12} - X_C) \phi^C}{G_{12} + \sigma_1 - \sigma_2}.$$

LaRC04

The 2D misalignment model assumes that the kinking occurs in the plane of the lamina. LaRC04 incorporates a more complex 3D model for the kink band formation. The kink plane is at an angle ψ to the plane of the lamina. It is assumed to lie at an angle so that $\tau_{2\psi 3\psi} = 0$ and is thus given by:

$$\psi = \frac{1}{2} \operatorname{atan} \left(\frac{2\tau_{23}}{\sigma_2 - \sigma_3} \right)$$

and the stresses rotated in this plane are:

$$\sigma_{2\psi 2\psi} = \frac{(\sigma_2 + \sigma_3)}{2} + \frac{(\sigma_2 + \sigma_3)}{2} \cos(2\psi) + \tau_{23} \sin(2\psi)$$

$$\sigma_{3\psi 3\psi} = \sigma_2 + \sigma_3 - \sigma_{2\psi 2\psi}$$

$$\tau_{12\psi} = \tau_{12} \cos(\psi) + \tau_{31} \sin(\psi)$$

$$\tau_{2\psi 3\psi} = 0$$

$$\tau_{3\psi 1} = \tau_{31} \cos(\psi) - \tau_{12} \sin(\psi)$$

Following the definition of a kink plane, the stresses are rotated into a misaligned frame. This frame defined by evaluating the initial and the misalignment angles for pure compression as well as the shear strain under the assumption of linear shear behavior and small angle approximation:

$$\gamma_{1m2m_c} = -\frac{\phi_c X_c}{G_{12}}$$

where ϕ_c is the misalignment angle for pure compression.

$$\phi_c = \operatorname{atan} \left(\frac{1 - \sqrt{1 - 4 \left(\frac{S^L}{X_c} + \eta^L \right) \frac{S^L}{X_c}}}{2 \left(\frac{S^L}{X_c} + \eta^L \right)} \right)$$

$$\phi_0 = \phi_c - \gamma_{1m2m_c}$$

$$\gamma_{1m2m} = \frac{(\phi_0 * G_{12} + |\tau_{12}|)}{(G_{12} + \sigma_1 - \sigma_2)} - \phi_0$$

$$\phi = \frac{\tau_{12\psi}}{|\tau_{12\psi}|} (\phi_0 + \gamma_{1m2m})$$

Following this, the stresses can be rotated into the misaligned coordinate system:

$$\sigma_{1m1m} = \frac{(\sigma_1 + \sigma_2 \psi_2 \psi)}{2} + \frac{(\sigma_1 - \sigma_2 \psi_2 \psi)}{2} \cos(2\psi) + \tau_{12\psi} \sin(2\psi)$$

$$\sigma_{2m2m} = \sigma_1 + \sigma_2 \psi_2 \psi - \sigma_{1m1m}$$

$$\tau_{1m2m} = -\frac{(\sigma_1 - \sigma_2 \psi_2 \psi)}{2} \sin(2\psi) + \tau_{12\psi} \cos(2\psi)$$

$$\tau_{2m3\psi} = \tau_{2\psi} \psi \cos(\psi) - \tau_{3\psi_1} \sin(\psi)$$

$$\tau_{3\psi 1m} = \tau_{3\psi_1} \cos(\psi)$$

5.4.4.6.3. LaRC03 (2D)

Fiber Failure

Fiber tensile failure

For fiber tension a simple maximum strain approach is applied:

$$f_f = \frac{\varepsilon_1}{T} \text{ for } \sigma_1 \geq 0 \quad (5.117)$$

Fiber compressive failure for transverse compression

Fiber compression failure for matrix compression is calculated as follows:

$$f_f = \frac{|\tau_{12}^m| + \eta_L \sigma_2^m}{S_{is}^L} \text{ for } \sigma_1 < 0 \text{ and } \sigma_{22}^m < 0 \quad (5.118)$$

Fiber compressive failure for transverse tension

For fiber compression failure with matrix tension, the following quadratic equation has to be solved:

$$f_f = g \left(\frac{\sigma_{22}^m}{Y_{is}^T} \right)^2 + \left(\frac{\tau_{12}^m}{S_{is}^L} \right)^2 + (1-g) \left(\frac{\sigma_{22}^m}{Y_{is}^T} \right) \text{ for } \sigma_1 < 0 \text{ and } \sigma_{22}^m \geq 0 \quad (5.119)$$

Matrix Failure

Matrix tensile failure

The formulation for matrix tensile failure is similar to that of fiber compressive failure under transverse compression. The difference is that the stress terms are not in the misaligned frame.

$$f_m = g \left(\frac{\sigma_2}{Y_{is}^T} \right)^2 + \left(\frac{\tau_{12}}{S_{is}^L} \right)^2 + (1-g) \left(\frac{\sigma_2}{Y_{is}^T} \right) \text{ for } \sigma_2 \geq 0 \quad (5.120)$$

Matrix compressive failure for longitudinal loading above transverse stress limit

Matrix compression failure is divided into two separate cases depending on the longitudinal loading. The failure function for the first case $\sigma_1 \geq -Y_c$ is:

$$f_m = \left(\frac{\tau_{eff}^T}{S^T} \right)^2 + \left(\frac{\tau_{eff}^L}{S_{is}^L} \right)^2 \quad \text{for } \sigma_1 \geq -Y_c \text{ and } \sigma_2 < 0, \quad (5.121)$$

where the effective shear stresses for matrix compression are based on the Mohr-Coulomb criterion which relates the effective shear stresses with the stresses of the fracture plane in Mohr's circle.

$$\tau_{eff}^T = -\sigma_2 \cos(\alpha_0) \left(\sin(\alpha_0) - \eta^T \cos(\alpha_0) \right) \quad (5.122)$$

$$\tau_{eff}^L = \cos(\alpha_0) |\tau_{12}| + \eta_L \sigma_2 \cos(\alpha_0) \quad (5.123)$$

The transverse shear strength S^T in terms of the transverse compressive strength and the fracture angle can be written as:

$$S^T = Y_c \cos(\alpha_0) \left(\sin(\alpha_0) + \frac{\cos(\alpha_0)}{\tan(2\alpha_0)} \right). \quad (5.124)$$

Matrix compressive failure for longitudinal loading below transverse stress limit

The failure function for the second case ($\sigma_1 < -Y_c$) is:

$$f_m = \left(\frac{\tau_{eff}^{mT}}{S^T} \right)^2 + \left(\frac{\tau_{eff}^{mL}}{S_{is}^L} \right)^2 \quad \text{for } \sigma_1 < -Y_c \text{ and } \sigma_2 < 0 \quad (5.125)$$

where the effective shear stresses are rotated into the misaligned frame.

$$\tau_{eff}^{mT} = -\sigma_2 \cos \alpha \left(\sin \alpha - \eta_T \cos \alpha \right), \quad (5.126)$$

$$\tau_{eff}^{mL} = \cos \alpha |\tau_{12}| + \eta_L \sigma_2 \cos \alpha \quad (5.127)$$

5.4.4.6.4. LaRC04 (3D)

Fiber Failure

Fiber tensile failure The LaRC04 fiber tensile failure criteria is simply a maximum allowable stress criterion with no interaction of other components:

$$f_f = \frac{\sigma_{11}}{X_t} \quad \text{for } \sigma_1 \geq 0$$

Fiber compressive failure for transverse compression

Fiber compressive failure is divided into two components depending of the direction of the transverse stress. For transverse compression it is:

$$f_f = \left(\frac{\tau_{1m2m}}{S_{is}^L - \eta^L \sigma_{2m2m}} \right)^2 \quad \text{for } \sigma_1 < 0 \text{ and } \sigma_{2m2m} < 0$$

Fiber compressive failure for transverse tension

The failure function for fiber compression and matrix tension is based on the ANSYS Combined Stresses & Strains formulation for the LaRC criteria.

$$f_{mf} = (1-g) \frac{\sigma_{2m2m}}{Y_{is}^T} + g \left(\frac{\sigma_{2m2m}}{Y_{is}^T} \right)^2 + \left(\frac{\tau_{1m2m}}{S^L} \right)^2 + \frac{\Lambda_{23}}{2} \left(\frac{\tau_{2m3\psi}}{S^L} \right)^2 \text{ for } \sigma_1 < 0 \text{ and } \sigma_{2m2m} \geq 0$$

Matrix Failure

Matrix tensile failure

The failure function for matrix tension is based on the ANSYS Combined Stresses & Strains formulation for the LaRC criteria.

$$f_m = (1-g) \frac{\sigma_2}{Y_{is}^T} + g \left(\frac{\sigma_2}{Y_{is}^T} \right)^2 + \left(\frac{\tau_{12}}{S^L} \right)^2 + \frac{\Lambda_{23}}{2} \left(\frac{\tau_{23}}{S^L} \right)^2 \text{ for } \sigma_2 \geq 0$$

Matrix compressive failure, transverse compression

$$f_m = \left(\frac{\tau^{Tm}}{S^T - \eta^T \sigma_n^m} \right)^2 - \left(\frac{\tau^{Lm}}{S_{is}^L - \eta^L \sigma_n^m} \right)^2 \text{ for } \sigma_2 < 0 \text{ and } \sigma_1 < -Y_c$$

where

$$\sigma_{nm} = \frac{\sigma_{2m2m} + \sigma_{3\psi3\psi}}{2} + \frac{\sigma_{2m2m} - \sigma_{3\psi3\psi}}{2} \cos(2\alpha) + \tau_{2m3\psi} \sin(2\alpha)$$

$$\tau_m^T = -\frac{\sigma_{2m2m} - \sigma_{3\psi3\psi}}{2} \sin(2\alpha) + \tau_{2m3\psi} \cos(2\alpha)$$

$$\tau_m^L = \tau_{1m2m} \cos(\alpha) + \tau_{3\psi1m} \sin(\alpha)$$

Matrix compressive failure, transverse tension

$$f_m = \left(\frac{\tau^T}{S^T - \eta^T \sigma_n} \right)^2 - \left(\frac{\tau^L}{S_{is}^L - \eta^L \sigma_n} \right)^2 \text{ for } \sigma_2 < 0 \text{ and } \sigma_1 \geq -Y_c$$

5.4.4.7. Cuntze's Failure Criterion

5.4.4.7.1. 2D Failures

Cuntze's approach is to strictly relate one failure mode to one basic strength [17 (p. 352)] so that $F(\bar{\sigma}, R^{mode}) = 1$. [22 (p. 352)] As for a unidirectional ply five basic strengths exist due to symmetry conditions five failure modes have to be considered related to X^t, X^c, Y^t, Y^c and S . So the criteria can be formulated for two different kinds of fiber-failure (tension and compression) and three different kinds of inter-fiber-failure (due to tension, compression and shear).

These single and rather simple criteria are then combined in an interaction criterion of the form

$$(\text{Eff})^m = \sum_1^5 (\text{Eff}^{(\text{modes})})^m, \quad (5.128)$$

where Eff is the so called effort, which is defined the same way as the inverse reserve factor for linear behavior and no residual stresses. In the current implementation those assumptions are assumed to be valid, so that the results and following formulations are presented in terms of failure functions based on inverse reserve factors.

The failure mode conditions for a UD ply in the 2D case are presented in Equations 148 - 152

$$f_{FF1} = \frac{\sigma_1}{X^t}, \text{ for } \sigma_1 \geq 0 \quad (5.129)$$

$$f_{FF2} = \frac{-\sigma_1}{X^c}, \text{ for } \sigma_1 < 0 \quad (5.130)$$

$$f_{IFF1} = \frac{\sigma_2}{Y^t}, \text{ for } \sigma_2 \geq 0 \quad (5.131)$$

$$f_{IFF2} = \frac{|\tau_{21}|}{S - B_{\perp} \|\sigma_2\|} \quad (5.132)$$

$$f_{IFF3} = \frac{-\sigma_2}{Y^c}, \text{ for } \sigma_2 < 0 \quad (5.133)$$

The parameter B_{\perp} (as illustrated in [18 (p. 352)]) is the friction parameter μ_{\perp} used in the simplified 2D Mohr-Coulomb formula. It can be estimated for a typical fracture point by

$$\mu_{\perp} = \frac{(S - \tau_{21}^{fracture})}{\sigma_2^{fracture}} \quad (5.134)$$

as, for example, presented in [22 (p. 352)] (B_{\perp} of the 2D FC is not b_{\perp} of the 3D FC).

Thus the "global" effort can then be calculated with

$$Eff^m = f_{FF1}^m + f_{FF2}^m + f_{IFF1}^m + f_{IFF2}^m + f_{IFF3}^m, \quad (5.135)$$

where m is the interaction exponent (set to a default value of 3.1). Due to the tension-compression differentiation at most three single failure modes can be active at the same time.

5.4.4.7.2. 3D Failures

For the 3D case of the single criteria are presented in Equations 156 - 160 (see [18 (p. 352)]). The single criteria are formulated in terms of the invariants

$$\begin{aligned} I_1 &= \sigma_1 \\ I_2 &= \sigma_2 + \sigma_3, \\ I_3 &= T_{31}^2 + T_{21}^2, \\ I_4 &= (\sigma_2 - \sigma_3)^2 + 4T_{23}^2, \\ I_5 &= (\sigma_2 - \sigma_3)(T_{31}^2 + T_{21}^2) - 4T_{23}T_{31}T_{21} \end{aligned} \quad (5.136)$$

As the invariant $I_1 = \sigma_1$ the fiber failure is formulated the same way as for the 2D version simply replacing the invariant by the stress.

$$f_{FF1} = \frac{\sigma_1}{X^t}, \text{ for } \sigma_1 \geq 0 \quad (5.137)$$

$$f_{FF2} = \frac{-\sigma_1}{X_c}, \text{ for } \sigma_1 < 0 \quad (5.138)$$

$$f_{IFF1} = \frac{I_2 + \sqrt{I_4}}{2Y_t} \quad (5.139)$$

$$f_{IFF2} = \frac{I^{\frac{3}{2}}}{S^3} + b_{\perp\parallel} \frac{I_2 I_3 - I_5}{S^3} \quad (5.140)$$

$$f_{IFF3} = (b_{\perp\parallel}^{\tau} - 1) \frac{I_2}{Y_c} + \frac{b_{\perp\parallel}^{\tau} I_4 + b_{\perp\parallel}^{\tau} I_3}{Y_c^2} \quad (5.141)$$

According to Cuntze and Freund [17 (p. 352)] safe curve fit parameters (can be determined from multiaxial test data) for glass, carbon and aramid fiber reinforced plastics are in the range of

$$0.05 < b_{\perp\parallel} < 0.15, \quad 1.0 < b_{\perp\parallel}^{\tau} < 1.6, \quad 0 < b_{\perp\parallel}^{\tau} < 0 \quad (5.142)$$

The global failure functions is then again defined as in Equation 147.

5.4.5. Sandwich Failure

5.4.5.1. Core failure

The Core Failure criterion predicts failure in core materials due to interlaminar shear and normal stresses. The normal stresses are only considered if they are enabled (see failure criterion definition). By default the σ_3 stress component is inactive.

In the case σ_3 of inactive, the criterion has this form

$$f = \frac{|\tau_{23}|}{Q} + \frac{|\tau_{13}|}{R}, \quad (5.143)$$

else

$$f = \frac{|\tau_{23}|}{Q} + \frac{|\tau_{13}|}{R}, \text{ for } \sigma_3 \leq 0 \quad (5.144)$$

$$f = \frac{|\tau_{23}|}{Q} + \frac{|\tau_{13}|}{R} + \frac{\sigma_3}{Z_t}, \text{ for } \sigma_3 > 0 \quad (5.145)$$

5.4.5.2. Face sheet wrinkling

Wrinkling of sandwich face sheets is a local instability phenomenon, in which the face sheets can be modeled as plates on an elastic foundation formed by the core. Simple formulas for estimating wrinkling stresses of sandwich face sheets under uniaxial load have been presented, for instance, in [21 (p. 352)] and [31 (p. 352)]. Linear elastic material behavior is assumed. Possible interaction of the top and bottom face sheets is not considered.

In the following, ξ , η , and ζ refer to a coordinate system in which the ξ -axis is in the direction of compression and the ζ -axis is perpendicular to the face sheets. The subscript F and C indicate the face sheet and the core, respectively.

For sandwich laminates with *homogeneous cores*, the wrinkling stress of a face sheet is

$$\sigma_w = -Q \left(\frac{E_\xi, F E_\zeta, C G_{\zeta\xi}, C}{1 - v_{\xi\eta}, F v_{\eta\xi}, F} \right) \quad (5.146)$$

where the theoretical value of the so-called wrinkling coefficient Q is 0.825. The effects of initial waviness and imperfections of the face sheet are normally accounted for by replacing the theoretical value of the wrinkling coefficient with a lower value. [21 (p. 352)] and [31 (p. 352)] recommend to use a value $Q = 0.5$ as a safe design value for homogeneous cores.

The wrinkling stresses for sandwich laminates with *honeycomb cores* are estimated with the expression

$$\sigma_w = -Q \left(\frac{E_\xi, F E_\zeta, C h_F}{(1 - v_{\xi\eta}, F v_{\eta\xi}, F) h_C} \right)^{\frac{1}{2}} \quad (5.147)$$

The theoretical value of Q is 0.816, whereas a safe design value is $Q = 0.33$ [21 (p. 352)] [31 (p. 352)].

The prediction of wrinkling under multiaxial stress state is discussed in [31 (p. 352)]. When in-plane shear stresses exist, it is recommended that the principal stresses are determined first. If the other of the two principal stresses is tensile, it is ignored and the analysis is based on the equations given above. When biaxial compression is applied, wrinkling can be predicted with an interaction formula. The condition for wrinkling is

$$\frac{\sigma_\xi}{\sigma_{\xi,w}} + \left(\frac{\sigma_\eta}{\sigma_{\eta,w}} \right)^3 = 1 \quad (5.148)$$

where ξ is "the direction of maximum compression" [31 (p. 352)]. For orthotropic sandwich face sheets, ξ is more logically interpreted as the most critical of the two directions. The wrinkling stresses $\sigma_{\xi,w}$ and $\sigma_{\eta,w}$ are computed from the formulas for uniaxial compression by considering the compressive stresses in the ξ - and η -directions independently.

The average face sheet stresses σ_x , σ_y , and τ_{xy} are obtained from the layer stresses of the face sheets. The following procedure for the computation of reserve factors is then used independently for the top and bottom face sheets.

If the shear stress τ_{xy} of the face sheet is zero, the normal stresses σ_x and σ_y are used directly in the prediction of wrinkling. Otherwise, the principal stresses are determined first:

$$\sigma_{\xi,\eta} = \frac{1}{2} (\sigma_x - \sigma_y) \pm \left[\frac{1}{4} (\sigma_x - \sigma_y)^2 + \tau_{xy}^2 \right]^{\frac{1}{2}} \quad (\sigma_\xi \geq \sigma_\eta) \quad (5.149)$$

The orientation of the normalized principal stresses with respect to the xy-coordinate system is

$$\varphi = \frac{1}{2} \arctan \left(\frac{2\tau_{xy}}{\sigma_x - \sigma_y} \right) \quad (5.150)$$

$$\left. \begin{array}{l} \sigma_\xi \geq 0 \\ \sigma_\eta \geq 0 \end{array} \right\} \Rightarrow f_w = 0 \quad (5.151)$$

$$\left. \begin{array}{l} \sigma_\xi \geq 0 \\ \sigma_\eta < 0 \end{array} \right\} \Rightarrow f_w = \frac{\sigma_\eta}{\sigma_{\eta,w}} \quad (5.152)$$

$$\left. \begin{array}{l} \sigma_\xi < 0 \\ \sigma_\eta < 0 \end{array} \right\} \Rightarrow \begin{cases} f_w = R_\xi + R_\eta^3 & , \quad R_\xi \geq R_\eta \\ f_w = R_\eta + R_\xi^3 & , \quad R_\xi < R_\eta \end{cases} \quad (5.153)$$

where

$$R_\xi = \frac{\sigma_\xi}{\sigma_{\xi,w}} ; \quad R_\eta = \frac{\sigma_\eta}{\sigma_{\eta,w}} \quad (5.154)$$

$$RF_w = \min(RF_{w,Ft}, RF_{w,Fb}) \quad (5.155)$$

5.4.6. Interlaminar failure

The failure function for interlaminar failure is defined as

$$f = \frac{\tau_{23}^2 + \tau_{13}^2}{\tau_{ISS}^2}, \quad (5.156)$$

where τ_{ISS} represents the lowest of the stress allowables Q and R of the two involved plies.

5.4.7. Isotropic material failure

The failure criteria for isotropic materials is based on the von Mises stress (or equivalent stress in ANSYS):

$$\sigma_{eqv} = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 - \sigma_1\sigma_2 - \sigma_1\sigma_3 - \sigma_2\sigma_3 + 3(\tau_{12}^2 + \tau_{13}^2 + \tau_{23}^2)} \quad (5.157)$$

or on the von Mises strain (or equivalent strain in ANSYS):

$$\varepsilon_{eqv} = \frac{1}{1+\nu_{12}} \sqrt{\frac{(\varepsilon_I - \varepsilon_{II})^2 + \varepsilon_I^2 + \varepsilon_{II}^2}{2}} \quad (5.158)$$

where $/$ and $//$ the first and second principal strains are. For isotropic material, the stress failure function is defined as:

$$f = \frac{vM_\sigma}{\sigma_{eqv}}, \quad (5.159)$$

and the strain failure function is defined as:

$$f = \frac{vM_\varepsilon}{\varepsilon_{eqv}}. \quad (5.160)$$

5.4.8. Failure Criteria vs. Ply Type Table

The table below illustrates which failure criteria evaluates the safety of the different ply types.

	Ply Type					
Failure Criteria	regular (UD)	woven	homogeneous core	honeycomb core	iso-tropic	sandwich
Max Strain	•	•				
Max Stress	•	•				
Tsai-Wu	•	•				
Tsai-Hill	•	•				
Hashin	•					
Puck	•	•1				
Cuntze	•					
LaRC	•					
Wrinkling						•2
Core Failure			•	•		
Von Mises					•	

- 1: In combination with Puck for Woven specifications.

- 2: Sandwich is a laminate with at least one core layer.

5.5. Classical Laminate Theory

5.5.1. Overview

The Classical Laminate Theory provides useful information about the mechanical behavior of a composite structure. The in-plane and flexural laminate stiffness represent characteristic values which can be compared with other laminates. These information can be quite useful since no external loads and boundary conditions are necessary to evaluate these values.

The Classical Laminate Theory is described in many text books [32 (p. 352)] and [33 (p. 352)]. The basic assumptions are

- Layers are perfectly bonded together.
- The material properties of each layer are constant through the thickness.
- Linear-elastic strain-stress behavior.
- Lines originally straight and normal to the mid-plane remain straight and normal in extension and bending.
- Plane stress state.
- In-plane strains and curvature are small compared to all other dimensions.

This requirements are fulfilled in a relatively thin or moderate thick laminate where the thickness is small compared to the in-plane extensions (length and width).

5.5.2. Analysis

5.5.2.1. Laminate Stiffness and Compliance Matrices

Evaluates the classical **ABD** matrix of the laminate which is the stiffness matrix of the laminate. **A** is the in-plane stiffness matrix, **B** describes the coupling between in-plane forces and bending moments and **D** is the flexural stiffness matrix. The inverse of the **ABD** matrix is the compliance matrix also called **abd** matrix (ABD^{-1}).

The **B** matrix becomes 0 in the case of a symmetrically balanced laminate.

In addition to the ABD terms the shear matrix **Q** is evaluated as well. The shear matrix has form

$$\begin{bmatrix} C_{44} & C_{45} \\ C_{54} & C_{55} \end{bmatrix}$$

C_{44} and C_{55} represents the G_{23} and G_{31} stiffness, respectively.

5.5.2.2. Normalized Laminate Stiffness and Compliance Matrices

The stiffness and compliance matrices can also be written in a normalized form where **h** is the laminate thickness. The normalized stiffness matrices **ABD*** are

- In-plane stiffness matrix: $A^* = \frac{A}{h}$
- Coupling matrix: $B^* = \frac{2 \cdot B}{h^2}$
- Flexural stiffness matrix: $D^* = \frac{12 \cdot D}{h^3}$
- Out-of-plane shear matrix: $Q^* = \frac{Q}{h}$

And the normalized compliance matrices **abd*** are

- In-plane compliance matrix: $a^* = a \cdot h$
- Coupling matrix: $b^* = \frac{h^2 \cdot b}{2}$
- Flexural compliance matrix: $d^* = \frac{h^3 \cdot d}{12}$.
- Out-of-plane shear matrix: $q^* = q \cdot h$

5.5.2.3. Laminate Engineering Constants

The laminate engineering constants are derived from the normalized compliance matrices which can be compared with the ply compliance matrix. Therefore the in-plane engineering constants are

- Laminate stiffness $E_1 = \frac{1}{*} \cdot \frac{1}{a_{11}}$
- Laminate stiffness $E_2 = \frac{1}{*} \cdot \frac{1}{a_{22}}$
- Laminate shear stiffness $G_{12} = \frac{1}{*} \cdot \frac{1}{a_{66}}$.

and the out-of-plane shear terms are

- Shear correction factors k_{44} and k_{55}
- Out-of-plane shear stiffness $G_{31} = \frac{1}{*} \cdot k_{55}$
- Out-of-plane shear stiffness $G_{23} = \frac{1}{*} \cdot k_{44}$

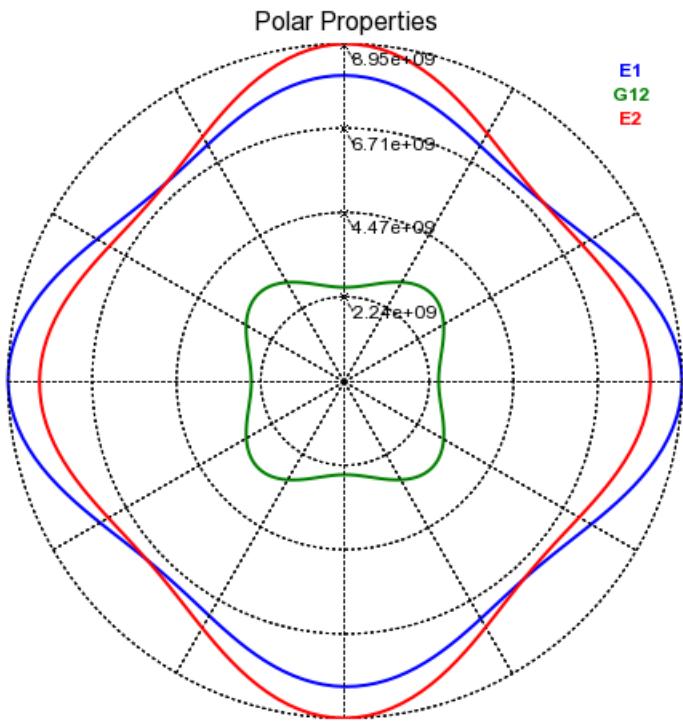
And the flexural constants become

- Flexural laminate stiffness $E_1^f = \frac{1}{*} \cdot \frac{1}{d_{11}}$
- Flexural laminate stiffness $E_2^f = \frac{1}{*} \cdot \frac{1}{d_{22}}$
- Flexural laminate shear stiffness $G_{12}^f = \frac{1}{*} \cdot \frac{1}{d_{66}}$.

In the case of coupling between in-plane and bending forces (B matrix has non-zero elements), the engineering constants represents the case where the laminate is free to curve when loaded with in-plane forces.

5.5.2.4. Polar Properties

The polar plot shows the in-plane laminate engineering constants of the laminate (E_1 , E_2 and G_{12}) rotated by 0 to 360 degree. This plot highlights the anisotropy of the laminate and the influence of the orientation.



5.5.2.5. Analysis Options

The options **Offset is Middle** and **Consider Coupling Effect** allows to change/modify the evaluation of the laminate properties.

- **Offset is Middle:** By default true. This option moves the reference plane of the laminate to the middle. This can be quite useful since the layers in a FE analysis mostly have one offset direction and hence the reference plane is at the top or bottom of the laminate. This option reduces the coupling effect between in-plane and bending forces. This considers the results of the stiffness and compliance matrices, polar properties and laminate engineering constants.
- **Consider Coupling Effect:** By default true. The polar properties and laminate engineering constants are derived from the laminate compliance matrix (inverse ABD matrix) where by default the coupling effect is considered. This option allows to neglect this effect which causes that the polar properties and laminate engineering constants represent the values of a symmetrically balanced laminate. Note that the coupling effect can significantly reduce the polar properties and laminate engineering constants.

Chapter 6: The ACP Python Scripting User Interface

The ACP Python Module provides the scripting user interface of ACP.

- 6.1. Introduction to ACP Scripting
- 6.2. The Python Object Tree
- 6.3. DB Database
- 6.4. Material Classes
- 6.5. Model Classes
- 6.6. Solid-model Classes
- 6.7. Solution Classes
- 6.8. Scene Classes
- 6.9. Postprocessing Definition Classes
- 6.10. Plot

6.1. Introduction to ACP Scripting

Basic Scripting

The scripting language of ACP is Python which is an object oriented programming language. The user should have some basics experience in object oriented programming for the optimal use, but a basic script can easily be written by modifying an existing one or copy and paste the [History View](#) or the [Shell View](#).

An example of one command is given below. In this command, the density of the material Corecell_A450 is defined (or modified if already defined) as 90.

```
db.models['class40.1'].material_data.materials['Corecell_A450'].rho = 90.0
```

The easiest way to use scripting is to generate a script with the Graphical User Interface. Every action performed via the GUI is written to the [History View](#). Use copy and paste to create your own scripts with a text editor. Save the scripts as *.py file and use the **Run Script...** functionality in the File [Menu](#) to run your script. In the case of error(s) check the [Shell View](#) for more details.

The aforementioned approach may be limited when it comes to retrieving solution results or specific model information. The section [Advanced Scripting](#) lists several examples of how to access such information via a script.

Additionally, an example of a python script is given in the Class40 example folder in the ACP installation directory. The update_materials.py file defines stress limits, Puck and Tsai-Wu constants for the UD and core materials. A list of common commands used to retrieve or modify data is shown below.

Advanced Scripting

These examples are provided as a guide to more involved commands in the shell view. They all refer to the example model [Kiteboard](#).

Maximum Layup Thickness

```
# get active model
model = db.active_model
# get total thickness of all elements
thicknesses = list(model.mesh_query(name= thickness ,position= centroid ,selection= all ))
# get element labels
labels = model.mesh_query(name= labels ,position= centroid ,selection= all )
# get maximum thickness
max_thickness = max(thicknesses)
index_of_max = thicknesses.index(max_thickness)
# get element label with max thickness
element_label_with_max_thickness = labels[index_of_max]
```

Maximum Ply Thickness

```
# get active model
model = db.active_model
model = db.active_model
# create new selection of all elements attached to a specific ply
modeling_ply = model.modeling_ply_groups['Core'].plies['mp_4']
model.select_elements(selection='sel0',op='new',attached_to=[modeling_ply])
# get total thickness of the first entity of selection sel0
thicknesses = list(model.mesh_query(name='thickness',position='centroid',selection='sel0', entities=[modeling_ply]))
# get element labels
labels = model.mesh_query(name='labels',position='centroid',selection='sel0')
# get maximum thickness
max_thickness = max(thicknesses)
index_of_max = thicknesses.index(max_thickness)
# get element label with max thickness
element_label_with_max_thickness = labels[index_of_max]
```

Maximum Inverse Reserve Factor and Failure Mode

```
# get active model
model = db.active_model
# get first solution
solution = model.solutions.values()[0]
# get the failure criterion definition
fc_definition = model.definitions[ 'FailureCriteria.MaxStrain_Core' ]
# get element labels
labels = model.mesh_query(name=' labels' ,position= 'centroid ' ,selection= all )
# get inverse reserve factors of all elements
irfs = list(solution.query(definition=fc_definition,position= 'centroid ' ,selection= 'all ' ,component='irf'))
# get failure modes of all elements
failure_modes = solution.query(definition=fc_definition,position= 'centroid ' ,selection= 'all ' ,component='fm')
# get the maximum IRF value
max_irf = max(irfs)
# get the index of maximum IRF
index_of_max = irfs.index(max_irf)
# get failure mode corresponding to maximum IRF
critical_failure_mode = failure_modes[index_of_max]
# get element label where the maximum IRF occurs
element_label_of_max = labels[index_of_max]
```

6.2. The Python Object Tree

The ACP Python interface is organized as a static python object tree. This tree contains all loaded models, solutions, definitions, views and scenes. Access to the actually loaded object tree is always provided through the root object `acp.DB` in the embedded Python shell.

An image of the python object tree appears only in the online help. If you are reading the PDF version of the help and want to see the figure, please access this section in the online help.



6.3. DB Database

`class compolyx.DB`

Class to represent ComPoLyX database

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
```

active_model

Active model

clear()

Clear database

close(model=None)

Close model

Parameters

- model: model to close (optional) if no model is given all models were closed

file_revision

Revision number if the database is read from an ACP File.

import_model (name, path, format, reduced_integration=True, ignored_entities=None, post_processing_model= None, unit_system_type=None)

Create a model from file

Parameters

- name: Custom name of the model
- path: Path to the data file
- format: File format string. Choose one of 'abaqus:inp','ansys:cdb','ansys:dat','nastran:bdf','ansys:h5'
- reduced_integration: Whether to use reduced integration schemes for 4-node layered shells. Default: True
- ignored_entities: Entities to ignore. Can be a subset of the following list: ['mesh','element_sets','materials','coordinate_systems','shell_sections']
- post_processing_model: Whether to handle this model as static and only post-processing functionality can be performed. Default: True
- unit_system_type : Set the unit system of the model to this type. Ignored if a unit system was already defined in the data file.

material_data

Material Data Base

models

Models

open(path, replace_mesh_kwarg=None, replace_workbench_inputs=None, pre_db=None)

Open ACP file and append the model to models container :Parameters: - path: Path to ACP file - replace_mesh_kwarg Optional keyword arguments to replace the mesh to load in db.import_model(...) upfront - replace_workbench_inputs Optional dictionary with Workbench inputs to replace before executing the .acp file

reload(model)

Reloads the model

Parameters

- -model: the model to be reloaded

save(path=None, model=None)

Save active model

Parameters

- path: file path
- model: active model

6.4. Material Classes

This section contains the following information:

6.4.1. [MaterialData](#)

6.4.2. [Materials](#)

6.4.3. [Fabric](#)

6.4.4. [Stackup](#)

6.4.5. [SubLaminate](#)

6.4.1. MaterialData

```
class compolyx.MaterialData(graph, parent=None)
```

MaterialData manages all composite material data.

copy(source, on_duplicate_name='keep_both')

copy a list of material data source, keeps track of all dependencies

Parameters

- source : a list of source of copy
- on_duplicate_name : [action to take if source.name is already contained in self.fabrics] keep_both : create a new instance with the same name (different id) overwrite : replace first instance with equal name in self with source keep_existing : ignore copy action, returns first existing instance in self with equal name

copy_fabric(source, on_duplicate_name='keep_both', memo=None)

Copy a fabric

Parameters

- source: Source object to copy

- `on_duplicate_name` : [action to take if source.name is already contained in self.fabrics] `keep_both` : create a new instance with the same name (different id) `overwrite` : replace first instance with equal name in self with source `keep_existing` : ignore copy action, returns first existing instance in self with equal name

`memo` : a dict to collect copied items (for internal dependency tracking when copying stackups or sub-laminates)

Returns New Instance of Fabric

copy_material(source, on_duplicate_name='keep_both', memo=None)

Copy a material

Parameters

- `source`: Source object to copy
- `on_duplicate_name` : [action to take if source.name is already contained in self.materials] `keep_both` : create a new instance with the same name (different id) `overwrite` : replace first instance with equal name in self with source `keep_existing` : ignore copy action, returns first existing instance in self with equal name

`memo` : a dict to collect copied items (for internal dependency tracking when copying stackups or sub-laminates)

Returns New instance of material

copy_stackup(source, on_duplicate_name='keep_both', memo=None)

Copy a stackup

Parameters

- `source`: Source object to copy
- `on_duplicate_name` : [action to take if source.name is already contained in self.stackups] `keep_both` : create a new instance with the same name (different id) `overwrite` : replace first instance with equal name in self with source `keep_existing` : ignore copy action, returns first existing instance in self with equal name
- `memo` : a dict to collect copied items

Returns New instance of Fabric

copy_sub_laminate(source, on_duplicate_name='keep_both', memo=None)

Copy a sub laminate

Parameters

- `source`: Source object to copy
- `on_duplicate_name` : [action to take if source.name is already contained in self.sub_laminates] `keep_both` : create a new instance with the same name (different id) `overwrite` : replace first instance with equal name in self with source `keep_existing` : ignore copy action, returns first existing instance in self with equal name
- `memo` : a dict to collect copied items

Returns New Instance of sub laminate

create_fabric(name, id=None, material=None, thickness=0.0, draping0=1.0, draping1=0.0, draping2=0.0, area_price=0.0, ignore_for_postprocessing=False, global_dropoff_material=True)

Create a new fabric

Parameters

- name: Name for the Fabric
- material: Material of the Fabric
- thickness: Thickness of the Fabric
- draping0: Draping Parameter0
- draping1: Draping Parameter1
- draping2: Draping Parameter2
- area_price: Area Price of the Fabric
- ignore_for_postprocessing: Flag if this material is post-processed
- global_dropoff_material: Flag if for drop-offs of this material the global drop-off material shall be used

Returns The created Fabric

Examples:

```
>>> material_data = db.models['beam'].material_data
>>> fabric_1 = material_data.create_fabric(name='Fabric.1',
material=material_data.materials['Material.1'], thickness=0.2,
draping1=0.3, draping2=0.7)
```

create_material(name, id=None, ply_type='regular', E1=0.0, E2=0.0, E3=0.0, G12=0.0, G31=0.0, G23=0.0, nu12=0.0, nu13=0.0, nu23=0.0, rho=0.0, locked=False, ext_id=None)

Create a Material

Parameters

- name: Name of the new Material
- E1 - rho: Material parameters

Returns New Instance of Material

create_stackup(name, id=None, fabrics=None, draping0=1.0, draping1=0.0, draping2=0.0, area_price=0.0, symmetry='No Symmetry', layup_sequence='Top-Down')

Create a new Stackup

Parameters

- name: Name for the Stackup
- fabrics: Fabrics of the Stackup
- draping0: Draping Parameter0

- draping1: Draping Parameter1
- draping2: Draping Parameter2
- area_price: Area Price of the Stackup
- symmetry: Symmetry the Stackup can be 'No Symmetry', 'Even Symmetry' or 'Odd Symmetry'
- layup_sequence: Layup sequence of the Stackup can be 'Top-Down' or 'Bottom-Up'

Returns The created Stackup

Examples:

```
>>> material_data = db.models['beam'].material_data
>>> sublamine_1 = material_data.create_sub_laminate
(name='SubLamineate.1', fabrics=(material_data.fabrics['Fabric.1'],
material_data.stackups['Stackup.1']))
```

create_sub_laminate(name, id=None, fabrics=None, symmetry='No Symmetry', layup_sequence='Top-Down')

Create a new SubLamineate

Parameters

- name: Name for the Sub Laminate
- fabrics: Fabrics of the Sub Laminate
- symmetry: Symmetry the Sub Laminate can be 'No Symmetry', 'Even Symmetry' or 'Old Symmetry'
- layup_sequence: Layup sequence of the Sub Laminate can be 'Top-Down' or 'Bottom-Up'

Returns The created SubLamineate

Examples:

```
>>> material_data = db.models['beam'].material_data
>>> sublamine_1 = material_data.create_sub_laminate
(name='SubLamineate.1', fabrics=(material_data.fabrics['Fabric.1'],
material_data.stackups['Stackup.1']))
```

enabled

Whether MaterialData is currently enabled or not.

export_matml(path, unit_system=None)

Export materials to ANSYS Engineering Data MatML format.

Parameters

- path: Path to file to write.
- unit_system : Convert all quantities into this unit system. The units will be stored in the file written.

fabrics

Dictionary with all fabrics defined.

find_materials(properties)**

Find materials with the given properties or property ranges

Parameters

- properties: Arbitrary material properties which must be matched. Note that a single property value can be given as string, number or min-max range

Returns A list with materials which match the given properties. If nothing matches an empty list is returned.

Examples:

```
>>> material_data = db.models['model.1'].material_data
>>> materials = material_data.find_materials(E1=100000.0, nu12=0.3)
>>> materials = material_data.find_materials( name='1')
>>> materials = material_data.find_materials(E1=[200000.0, 220000.0],
    nu12=0.3, G12=[4500.0,5500.0])
```

import_matml(path, unit_system=None, material_apdl_path="")

Import material data from MatML file as provided by Workbench Engineering Data.

Parameters

- path: File to read from.
- unit_system : Created materials will be converted into this unit system.
- material_apdl_path : Specify the APDL file containing the ANSYS Engineering Data material definitions.

material_apdl_path

Optional path to file with APDL material definitions to be used in the CDB export.

materials

Dictionary with all materials defined.

matml_path

Path to MatML file as provided by Workbench EngineeringData

name

Currently a name is needed for every object in the db tree.

serialize()

Serialize to Python string

stackups

Dictionary with all stack ups defined.

sub_laminates

Dictionary with all sub laminates defined.

unit_system

Unit system of material data, propagated from model

6.4.2. Materials

class compolyx.**Material**(graph, obj, parent=None)

ComPoLyX material class.

This class allows to retrieve all material properties defined within the loaded Finite Element model. Additionally this class allows to define the needed material properties for postprocessing:
 - Stress limit values
 - Strain limit values
 - Failure property values
 - Sandwich core properties

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> model = db.models['class40.1']
>>> materials = model.material_data.materials
>>> mat_UD300 = materials['UD300_GLAS']
```

Example for the modification of stress / or strain limits::

```
>>> mat_UD300.stress_limits.Xt = 500.0e8
>>> mat_UD300.strain_limits.eXt = 0.00035
>>> mat_UD300.puck_constants.p_21_pos = 0.25
>>> mat_UD300.thermal_expansion_coefficients.aX = -1.e-6
>>> mat_UD300.tsai_wu_constants.XZ = -1.
>>> mat_UD300.larc_constants.fracture_angle_under_compression = 53.
```

characteristic

Material characteristic. Read only property.

ext_id

Id of corresponding Material in external source.

is_constant

True if all engineering constants are constant.

larc_constants

Larc constants.

link_path

Root path of the current node in the tree for links to this object

locked

Material is generated from an external source and cannot be changed.

ply_type

Ply type. Allowed string values: 'regular', 'woven', 'orthotropic_homogeneous_core', 'isotropic_homogeneous_core', 'honeycomb_core', 'isotropic_material'

puck_constants

Puck constants.

rho

Density

serialize()

Serialize to Python string

strain_limits

Strain limit values.

stress_limits

Stress limit values.

thermal_expansion_coefficients

Coefficients of thermal expansion.

tsai_wu_constants

Tsai-Wu constants.

woven_characterization

Woven characterization.

6.4.2.1. StressLimits

class

compolyx.StressLimits(stress_limits=None, **kwargs)

Compolyx stressLimit class..

This class allows to get and set stress limits of a material within a loaded Finite Element model.

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> model = db.models['class40.1']
>>> materials = model.material_data.materials
>>> mat_UD300 = materials['T700']
>>> mat_UD300.stress_limits.Xt = 500.0e8
>>> mat_UD300.strain_limits.Xc = 450.0e8
```

is_constant

True if all stress limits are constant.

set(kwargs)**

Generic function which allows to set all properties of stress limits >>> material_data.materials['1'].stress_limits.set(Sxy=0.2, Xc=0.1)

6.4.2.2. StrainLimits

class

compolyx.StrainLimits(strain_limits=None, **kwargs)

Compolyx strainLimit class.

This class allows to get and set strain limits of a material within a loaded Finite Element model.

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> model = db.models['class40.1']
>>> mat_UD300 = model.material_data.materials['UD300_GLAS']
>>> mat_UD300.strain_limits.eXt = 0.005
>>> mat_UD300.strain_limits.eXc = 0.003
```

is_constant

True if all strain limits are constant.

set(kwargs)**

Generic function which allows to set all properties of this class

6.4.2.3. PuckConstants

class

compolyx.PuckConstants(puck_constants=None, **kwargs)

Compolyx puckConstants class.

This class allows to set Puck failure properties.

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> model = db.models['class40.1']
>>> materials = model.material_data.materials
>>> mat_UD300 = materials['T700']
>>> mat_UD300.puck_constants.p_21_pos = 0.25
>>> mat_UD300.puck_constants.p_21_neg = 0.3
```

M

Puck effect of fiber parallel stresses on inter-fiber failure

interface_weakening_factor

Puck interface weakening factor for delamination failure

mat_type

Material type definition

p_21_neg

Puck criteria p21 negative

p_21_pos

Puck criteria p21 positive

p_22_neg

Puck criteria p22 negative

p_22_pos

Puck criteria p22 positive

s

Puck effect of fiber parallel stresses on inter-fiber failure

set(kwargs)**

Generic function which allows to set all properties of this class

6.4.2.4. WovenCharacterization

class

compolyx.WovenCharacterization(woven_characterization=None, **kwargs)

Compolyx wovenCharacterization class.

This class allows to define two UD's representing a woven material which is used for the Puck failure criterion evaluation.

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> model = db.models['class40.1']
>>> materials = model.material_data.materials
>>> mat_UD300 = materials['T700']
>>> mat_UD300.woven_characterization.E1_1 = 40000
>>> mat_UD300.woven_characterization.G12_2 = 2400
```

E1_1

Woven characterization Youngs modulus E1 of UD1

E1_2

Woven characterization Youngs modulus E1 of UD2

E2_1

Woven characterization Youngs modulus E2 of UD1

E2_2

Woven characterization Youngs modulus E2 of UD2

G12_1

Woven characterization shear modulus G12 of UD1

G12_2

Woven characterization shear modulus G12 of UD2

G23_1

Woven characterization shear modulus G23 of UD1

G23_2

Woven characterization shear modulus G23 of UD2

nu12_1

Woven characterization Poissons ration nu12 of UD1

nu12_2

Woven characterization Poissons ration nu12 of UD2

orientation_1

Woven characterization fiber orientation of UD1

orientation_2

Woven characterization fiber orientation of UD2

puck_constants1

Woven characterization Puck constants of UD1

puck_constants2

Woven characterization Puck constants of UD2

set(kwargs)**

Generic function which allows to set all properties of this class

stress_limits1

Woven characterization stress limits of UD1

stress_limits2

Woven characterization stress limits of UD2

6.4.2.5. ThermalExpansionCoefficients

class

compolyx.ThermalExpansionCoefficients(thermal_expansion_coefficients=None, **kwargs)

Compolyx ThermalExpansionCoefficients class.

This class allows to get and set the coefficients of thermal expansion of a material within a loaded Finite Element model.

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> model = db.models['class40.1']
>>> mat_UD300 = model.material_data.materials['UD300_GLAS']
>>> mat_UD300.thermal_expansion_coefficients.aX = -1.e-5
>>> mat_UD300.thermal_expansion_coefficients.aY = 1.e-6
```

aX

Coefficient of thermal expansion in material 1 direction.

aY

Coefficient of thermal expansion in material 2 direction.

aZ

Coefficient of thermal expansion in material 3 direction.

reference_temperature

Reference temperature.

set(kwargs)**

Generic function which allows to set all properties of this class

6.4.3. Fabric

class compolyx.Fabric(graph, obj, parent=None)¶

Class to represent fabric

area_price

Area price of fabric

area_weight

Area weight of fabric

create_plot(query={'polar_properties': ['E1', 'G12']})

Generates 2D-plots with the results of interest :Parameters: - query: query arguments

E.g: - layup:['pp'] Production plies - polar_properties:[‘E1’,’E2’,’G12’] polar plot of laminate stiffnesses
- text_plot:[‘materials’,‘angles’,‘thicknesses’] query={‘polar_properties’:[‘E1’,’G12’], layup:[‘pp’],
text_plot:[‘materials’,‘angles’,‘thicknesses’]}

draping0

Draping coefficient 0 for fabric

draping1

Draping coefficient 2 for fabric

draping2

Draping coefficient 0 for fabric

global_dropoff_material

Flag if global drop-off material is used in drop-off areas of this fabric.

graph_plot

Graph Plot object used to configure 2D plots.

ignore_for_postprocessing

Flag if this material is NOT post-processed.

material

Material of the fabric

serialize()

Serialize to Python string

thickness

Thickness of fabric

update_plot()

updates the 2D plot

6.4.4. Stackup

class compolyx.Stackup(graph, obj, parent=None)¶

Class to represent stack-up

add_fabric(fabric, angle=0.0)

Add fabric at end of fabrics list

area_price

Price per area of the Stackup

area_weight

Area weight of the Stackup

clear_fabrics()

Clear all fabrics

clt_query(query='laminate_properties')

Returns the properties of the classical laminate theory: :Parameters: - query: query parameters

Can be: - layup: Return the layup of the laminate (Modeling, Production and Analysis Plies - laminate_properties: Young's, flexural and shear moduli of the laminate - polar_properties: E1, E2 and G12 depending on the laminate orientation - stiffness_matrix: Returns the laminate stiffness matrix (ABD) - compliance_matrix: Returns the laminate compliance matrix (inverse of ABD)

create_plot(query={'layup': ['pp', 'ap'], 'polar_properties': ['E1', 'G12']}, core_scale_factor=None)

Generates 2D-plots with the results of interest Parameters: - query: Query parameters - core_scale_factor: Scale core thickness by this value.

Query parameters can be: - layup:[‘pp’,‘ap’] Production Ply and Analysis Plies - polar_properties:[‘E1’‘E2’‘G12’] polar plot of laminate stiffnesses - text_plot:[‘materials’,‘angles’,‘thicknesses’] property to show as label in the layup plot E.g: query={‘polar_properties’: [‘E1’, ‘G12’], layup:[‘pp’], text_plot:[‘materials’]}

draping0

Draping coefficient 0 for Stackup

draping1

Draping coefficient 1 for Stackup

draping2

Draping coefficient 2 for Stackup

fabrics

Fabrics property of the Stackup

get_all_fabrics()

Returns a list with all fabrics and orientations including symmetry and layup sequence option.

graph_plot

Graph Plot object used to configure 2D plots.

insert_fabric(pos, fabric, angle)

Insert fabric at given position

layup_sequence

Layup Sequence of the Stackup can be ‘Top-Down’ or ‘Bottom-Up’

remove_fabric(pos)

Remove fabric from position

serialize()

updates the 2D plot

symmetry

Symmetry of the Stackup can be ‘No Symmetry’, ‘Even Symmetry’ or ‘Odd Symmetry’

thickness

Thickness of the Stackup

update_plot()

updates the data of the 2D plot

6.4.5. SubLaminate

class compolyx.SubLaminate(graph, obj, parent=None)

Class to represent sub-laminate

add_fabric (fabric, angle=0.0)

Add fabric at end of fabrics list

area_price

Price per area of the Sub Laminate

area_weight

Area weight of the Sub Laminate

clear_fabrics()

Clear all fabrics

clt_query(query='laminate_properties')

Returns the properties of the classical laminate theory: :Parameters: - query: query parameters

Query parameters can be: - layup: Return the layup of the laminate (Modeling, Production and Analysis Plies - laminate_properties: Young's, flexural and shear moduli of the laminate - polar_properties: E1, E2 and G12 depending on the laminate orientation - stiffness_matrix: Returns the laminate stiffness matrix (ABD) - compliance_matrix: Returns the laminate compliance matrix (inverse of ABD)

create_plot(query={'layup': ['mp', 'pp', 'ap'], 'polar_properties': ['E1', 'G12']}, core_scale_factor=None)

Generates 2D-plots with the results of interest Parameters: - query: query parameters - core_scale_factor: Scale core thickness by this value.

Query parameters can be: - layup:[‘mp’,‘pp’,‘ap’] Modeling Ply, Production Plies and Analysis Plies - polar_properties:[‘E1’‘E2’‘G12’] polar plot of laminate stiffnesses - text_plot:[‘materials’,‘thicknesses’,‘angles’] text plot shown in the layup plot E.g.: query={‘polar_properties’:[‘E1’,‘G12’], layup:[‘pp’], text_plot:[‘materials’]}

fabrics

Fabrics property of the Sub Laminate

get_all_fabrics()

Returns a list with all fabrics and orientations including symmetry and layup sequence option.

get_all_sub_materials()

Returns a list with all sub materials (fabrics and stackups) and orientations including symmetry and layup sequence option.

graph_plot

Graph Plot object used to configure 2D plots.

insert_fabric(pos, fabric, angle)

Insert fabric at given position

layup_sequence

Layup Sequence of the Sub Laminate can be ‘Top-Down’ or ‘Bottom-Up’

remove_fabric(pos)

Remove fabric from position

serialize()

Serialize to Python string

symmetry

Symmetry of the Sub Laminate can be 'No Symmetry', 'Even Symmetry' or 'Odd Symmetry'

thickness

Thickness of the Sub Laminate

update_plot()

Updates the data of the 2D plot

6.5. Model Classes

6.5.1. Model

```
class compolyx.Model(name, path=None, format=None, reduced_integration=True, ignored_entities=None, graph=None, parent=None, post_processing_model=True, unit_system_type=None)
```

Class to represent a finite element model

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> model = db.import_model(name='class40.1', path='class40.cdb', format='ansys:cdb')
```

Get existing model:

```
>>> class40_model = db.models['class40.1']
```

active_scene

Active Scene

```
add_solution(name, id="", path=None, path2=None, format='ansys:rst', subcase=(False, 0), set=-1, load_factor=(False, 0.0), read_stresses_strains=True, use_felyx_to_compute_pp_results=True, automatic_reload=True, use_solid_results=True, recompute_iss_of_solids=False, deformation_scale_factor=1.0, show_deformed=False )
```

Load a nodal solution from file(s) and add it to the model

Parameters

- name: Custom name of the solution
- path: Path to the data file
- path2: Optional path to second result file. Useful for ANSYS PRNSOL solution, where nodal deformations and nodal rotations can be exported to different files only.
- format: File format string. Choose one of 'abaqus:fieldreport', 'ansys:prnsol', 'ansys:rst' or 'nastran:f06'
- subcase: Optional subcase to read. Only valid for 'nastran:f06' format..

- `load_factor`: Optional load factor within substep of non-linear solution where the nodal solution should be taken from. Only valid for 'nastran:f06' format
- `set`: Result set for ANSYS RST files, None is last result set
- `automatic_reload`: Reread data on update
- `read_stresses_strains`: Reads strain and stress results from the RST file (necessary to post-process non-linear solutions)
- `use_felyx_to_compute_pp_results`: Use ACP to compute strain and stress data
- `use_solid_results`: Maps solid element solution onto 'Layered Solid Reference Surface'
- `recompute_iss_of_solids`: For solids the interlaminar shear stresses are recalculated considering the laminate stacking
- `show_deformed`: whether to show the deformed mesh calculated in this solution
- `deformation_scale_factor`: The scale factor for visualizing the deformed mesh

Returns The new Solution instance just added to the model

analysis_model_path

Analysis model file path

analysis_ply_groups

Dictionary with all Analysis Ply Groups defined.

angle_tolerance

Section computation thickness tolerance (in length unit of model)

average_element_size()

Average element size of the model

cad_geometries

Dictionary with all CAD geometries defined

clear_stored_update_results()

Function clears the eventually stored update results

copy_cad_geometry(source)

Copy a CAD Geometry

Parameters

- `source`: Source object to copy

Returns New instance of a CAD Geometry

copy_combined_failure_criteria(source)

Copy a Combined Failure Criteria Definition

Parameters

- `source`: Source object to copy

Returns New instance of Combined Failure Criteria Definition

copy_edge_set(source)

Copy an edge set :Parameters: - source: Source object to copy

Returns New instance of edge set

copy_element_set(source)

Copy an element set

Parameters

- source: Source object to copy

Returns New instance of element set

copy_lookup_table(source)

Copy a Look-up Table

Parameters

- source: Source object to copy

Returns New instance of a Look-Up Table

copy_oriented_element_set(source)

Copy an oriented element set

Parameters

- source: Source object to copy

Returns New instance of oriented element set

copy_rosette(source)

Copy a Rosette

Parameters

- source: Source object to copy

Returns New instance of Rosette

copy_rule(source)

Copy a rule

Parameters

- source: Source object to copy

Returns New instance of rule

copy_sampling_element(source)

Copy a sampling element

Parameters

- source: Source object to copy

Returns New instance of a sampling element

copy_section_cut(source)

Copy a section cut

Parameters

- source: Source object to copy

Returns New instance of a section cut

copy_sensor(source)

Copy a sensor

Parameters

- source: Source object to copy

Returns New instance of a sensor

copy_solid_model(source)

Copy a sensor

Parameters

- source: Source object to copy

Returns New instance of a solid model

create_cad_geometry(name, id=None, path=None, scale_factor=1.0, use_default_precision=True, precision=None, locked=False)

Create a new CAD geometry object

Parameters

- name: Name
- id: ID
- path: File path of CAD file to load
- scale_factor: Scale geometry with this factor
- use_default_precision: Use default intersection precision
- precision: Precision used for geometrical operations (intersection points, thickness sampling, ...)
- locked: Whether this CAD geometry should be locked or not, used with geometries imported from Workbench

Returns The created CADGeometry object

create_combined_failure_criteria(name, set=[], id=None)

Create Combined Failure Criteria :Parameters

Parameters

- name: Name for the Combined Failure Criteria
- set: set of Failure Criteria to be assigned
- id: id to be assigned (optional)

Returns The created Combined Failure Criteria

```
create_cutoff_rule(name, id=None, cutoff_rule_type='geometry', offset=0.0, angle=0.0, origin=(0.0, 0.0, 0.0), direction=(1.0, 0.0, 0.0), distance_type='along_direction', ply_cutoff_type='production-ply_cutoff', ply_tapering=False, cutoff_geometry=None, edge_set=None, offset_method='lamin-ate_stack', offset_type='out_of_plane')
```

Create new Cut-off Rule

Parameters

- name: Name of the rule
- cutoff_rule_type: geometry, taper, or variable_taper
- offset: Offset of the rule (float for cutoff_rule_type='geometry' or taper, LookUpTableColumn for cutoff_rule_type='variable_taper')
- angle : Angle of the rule (ignored for cutoff_rule_type='geometry', float for taper, LookUpTableColumn for variable_taper)
- origin : Origin of the offset and angle interpolation for variable_taper
- direction : Direction of the offset and angle interpolation for variable_taper
- distance_type : along_direction or along_edge (only relevant for variable_taper)
- ply_cutoff_type: Determines on which ply level the cutoff is done.
- ply_tapering: Use ply tapering
- cutoff_geometry: CADGeometry for the rule (only relevant for cutoff_rule_type='geometry')
- edge_set: Edge Set for cutoff_rule_type='taper' or variable_taper
- offset_method : Method to compute offset of plies laminate_stack or attached_plies
- offset_type : Measure offset from edge set normal to element reference surface (out_of_plane) or in element reference surface (in_plane)

Returns The created rule

```
create_cylindrical_rule(name, id=None, origin=None, direction=None, radius=None, relat-ive_rule_type=False, include_rule_type=True)
```

Create new Cylindrical Rule

Parameters

- name: Name of the rule

- origin: Origin of the Cylindircal Rule
- direction: Direction of the Cylindircal Rule
- radius: Radius of the Cylindircal Rule

Returns The created Cylindircal Rule

create_edge_set(name, id=None, origin=(0.0, 0.0, 0.0), limit_angle=-1.0, edge_set_type='By Reference', element_set=None, nodes=[])

Create new Edge Set

Parameters

- name: Name of the Edge Set
- edge_set_type: 'By Nodes', 'By Reference', 'Imported' (only for imported Edge Sets)
- origin: Origin
- limit_angle:
- element_set: element set
- nodes: list of nodes

Returns The created Edge Set

create_element_set(name, id=None, element_ids=None, element_sets=None, x=None, y=None, z=None, op='new', middle_offset=False)

Create new element set

Parameters

- name: Name of the Element Set
- element_ids: Indices of elements to be assigned to Element Set
- element_sets: Select elements of these sets
- x,'y','z':
- op:
- middle_offset:

Returns The created Element Set

If element set already exists, it is updated depending on the operation given in op.

create_envelope_solution(name, id=None, solutions=[])

Create Envelope Solution

Parameters

- name: Name for the Envelope Solution

- solutions: list of Solutions that are combined

Returns The created Look-Up Table object

create_lookup_table1d(name, id=”, tabular_data=None)

Create a new 1D Look-Up Table object

Parameters

- name: Name
- id: ID

Returns The created Look-Up Table object

create_lookup_table3d(name, id=”, tabular_data=None, use_default_search_radius=True, search_radius=0.0, num_min_neighbors=1)

Create a new 3D Look-Up Table object

Parameters

- name: Name
- id: ID

Returns The created Look-Up Table object

create_oriented_element_set(name, id=None, orientation_point=(0.0, 0.0, 0.0), orientation_direction=(0.0, 0.0, 0.0), element_sets=None, rosettes='None', rosette_selection_method='minimum_angle', rules=None, draping_material=None, draping=False, draping_seed_point=(0.0, 0.0, 0.0), auto_draping_direction=True, draping_direction=(0.0, 0.0, 1.0), draping_mesh_size=False, reference_direction_field=None)

Create a new Oriented Element Set

Parameters

- name: The name of the oriented element set
- id: The id of the oriented element set.
- orientation_point: Orientation Point for the Oriented Element Set
- orientation_direction: Orientation Direction for the Oriented Element Set
- element_sets: Element Sets
- rosettes: Rosettes for the Oriented Element Set
- rosette_selection_method: Method to calculate element orientation
('minimum_angle', 'maximum_angle', 'minimum_distance', 'minimum_angle_superposed', 'minimum_distance_superposed', 'maximum_angle_superposed', 'ansys_classic', or 'tabular_values')
- reference_direction_field: Table column used to compute reference directions
- draping: Draping enabled

- draping_material: Material for draping
- draping_seed_point: Seed Point used to start draping
- draping_direction: Direction for draping
- auto_draping_direction: Generate direction for draping
- draping_mesh_size: Mesh size for draping

Returns The created Oriented Element Set

create_parallel_rule(

name, id=None, origin=None, direction=None, pos_distance=None, neg_distance=None, relative_rule_type=False, include_rule_type=True)

Create new Parallel Rule

Parameters

- name: Name of the Rule
- origin: Origin of the Parallel Rule
- direction: Direction of the Parallel Rule
- pos_distance: Positive Distance of the Parallel Rule
- neg_distance: Negative Distance of the Parallel Rule

Returns The created Parallel Rule

create_published_parameter(

name, source= , category= input , acp_type= float , description= , lower_limit=None, upper_limit=None, cyclic=False, float_list=[], string_list=[])

Create published parameter

create_rosette(

name, id=None, origin=(0.0, 0.0, 0.0), dir1=(1.0, 0.0, 0.0), dir2=(0.0, 1.0, 0.0), rosette_type='PARALLEL', edge_set=None)

Create a new rosette

Parameters

- name: The name of the Rosette
- id: ID (optional)
- origin: The origin of the Rosette
- dir1: Direction 1 of the Rosette
- dir2: Direction 2 of the Rosette
- rosette_type: Type of the Rosette ('PARALLEL', 'RADIAL', 'CYLINDRICAL', 'SPHERICAL', 'EDGE_WISE')

- edge_set: Edge Set to be used in Rosette

Returns The created Rosette

Example

```
>>>rosette_1 = model.create_rosette('Rosette.1',
                                     origin=(0.,0.,0.),
                                     dir1=(1., 0., 0.),
                                     dir2=(0., 1., 0.),
                                     rosette_type='PARALLEL')
```

create_sampling_element()

(name, id=None, point=(0.0, 0.0, 0.0), direction=(0.0, 0.0, 0.0), locked=False, use_default_reference_direction=True, rosette=None, offset_is_middle=True, consider_coupling_effect=True, solution=None)

Create a new Sampling Element

Parameters

- name: Name of the Sampling Element
- point: Sampling point
- direction: Sampling direction
- use_default_reference_direction: whether to use the default reference direction for the evaluation
- rosette: Rosette used for the evaluation of the reference direction
- offset_is_middle: Specifies the offset of the reference plane used for the CLT analyses
- consider_coupling_effect: Specifies whether the laminate properties are evaluated considering the coupling effect (B matrix) or not
- solution: Specifies the solution of the element-wise post-processing

Returns The created Sampling Element

create_scene()

(name, id=None, title="", active_set=None, projection='perspective', scale_factor=1.0, show_draped_fiber_directions=False, show_draped_transverse_directions=False, show_draping_mesh=False, show_edges=True, show_fiber_directions=False, show_flatwrap=False, show_global_coordinate_system=True, show_legend=True, show_normals=False, show_orientations=False, show_ref_directions=False, show_selected_mesh=False, show_section_cut_plots=False, show_solid_elements=False, show_surface=True, show_transverse_directions=False)

Create a new scene

Parameters

- name: Name of the scene
- show_deformed_mesh: Whether to show the deformed mesh
- show_undeformed_mesh: Whether to show the undeformed mesh
- scale_factor: Scale factor of the deformed mesh

create_sensor()

```
name, id=None, sensor_type='SENSOR_BY_AREA', entities=None, locked=False)
```

Create new Sensor

Parameters

- name: Name of the Rule
- sensor_type: Type of Sensor values are: SENSOR_BY_AREA,, SENSOR_BY_MATERIAL, SENSOR_BY_PLIES
- entities: Entities of the Sensor

Returns The created Sensor

create_solid_model()

```
name, id='', active=True, element_sets=None, ex_type=None, drop_off_type=None, offset_type=None,  
max_thickness=None, ply_group_pointers=None, element_set=None, use_default_element_index=True,  
element_index=0, use_default_node_index=True, node_index=0, use_default_section_index=True, sec-  
tion_index=0, use_default_material_index=True, material_index=0, use_default_coordinate_system_in-  
dex=True, coordinate_system_index=0, connect_butt_joined_plies=True, write_drop_off_elements=True,  
use_solid_elements=False, use_solid_model_prefix=True, use_homogeneous_drop_off_elements=True,  
global_dropoff_material=None, transfer_all_sets=True, transferred_element_sets=[], trans-  
ferred_edge_sets=[], delete_bad_elements=True, warping_limit=0.4, locked=False)
```

Create a new Solid Model

Parameters

- name: The name of the Solid Model
- id: The id of the Solid Model
- active : bool of active or inactive switch
- element_sets: Active status fo the Solid Model
- element_sets: a list of Element Sets
- ex_type: monolithic (1 element through the thickness),

analysis_ply_wise (1 element per layer), modeling_ply_wise (1 element for each modeling ply), production_ply_wise (1 element for each production ply) specify_thickness (1 element per layer, layers thicker than max_thickness are split to several solids of at most max_thickness) user_defined (groups plies by global ply numbers to groups material_wise (groups subsequent plies with equal material)

- drop_off_type: inside ply (one element inside the ply boundary), outside ply (one element outside the ply boundary)
- offset_type : shell normal (offset to the shell normal), surface normal (update normal direction by normal of layered solids),

distortion controlled (surface normal with local corrections)

- max_thickness : maximum thickness for one solid, splits the layer into more solids, if a single layer is thicker than this value (only for ex_type='specify thickness')
- ply_group_pointers : step used to make user defined ply groups
- element_set : (deprecated, use element_sets instead) a single element set
- use_default_element_index : consecutive element numbering if set to true
- element_index : start index for first element (only relevant if use_default_element_index)
- use_default_node_index : consecutive node numbering if set to true
- node_index : start index for first node (only relevant if use_default_node_index)
- use_default_section_index : consecutive section numbering if set to true
- section_index : start index for first element (only relevant if use_default_section_index)
- use_default_material_index : consecutive material numbering if set to true
- material_index : start index for first element (only relevant if use_default_material_index)
- use_default_coordinate_system_index : consecutive coordinate system numbering if set to true
- coordinate_system_index : start index for first coordinate system (only relevant if use_default_coordinate_system_index)
- connect_butt_joined_plies : connect adjacent plies without intermediate drop-offs
- write_drop_off_elements : drop-off elements are written to the *cdb file
- use_solsh_elements : the solid model is created out of solsh elements
- use_solid_model_prefix : the name of the solid model is used as a prefix for all components written to the *cdb file
- use_homogeneous_drop_off_elements: Flag to export the drop-off elements as homogeneous solid elements without layered option.
- global_dropoff_material: defines the global drop-off material
- transfer_all_sets: defines whether all edge and element sets should be transferred to the solid model.
- transferred_element_sets: element sets to transfer to the solid model if transfer_all_sets is set to false.
- transferred_edge_sets: edge sets to transfer to the solid model if transfer_all_sets is set to false.
- delete_bad_elements: Boolean whether to delete the erroneous elements or not
- warping_limit: Warping limit factor used to detect erroneous elements

Returns The created Solid Model

create_spherical_rule(name, id=None, origin=None, radius=None, relative_rule_type=False, include_rule_type=True)

Create new Spherical Rule

Parameters

- name: Name of the Rule
- origin: Origin of the Spherical Rule
- radius: Radius of the Spherical Rule
- include_rule_type: Include or Exclude area in rule
- relative_rule_type: Flag for relative rule

Returns The created Spherical Rule

create_tube_rule(

name, id=None, outer_radius=1.0, inner_radius=0.0, include_rule_type=True, edge_set=None)

Create new Tube Rule

Parameters

- name: Name of the Rule
- radius: Radius of the Spherical Rule
- include_rule_type: Include or Exclude area in rule
- edge_set: Edge Set for the rule

Returns The created rule

definitions

Definitions

edge_sets

Dictionary with all Edge Sets defined.

element_normal (globalID)

Returns the element normal (direction)

Parameters

- *globalID*: Element label

element_sets

Dictionary with all Element Sets defined.

export(path)

Exports all ACP composite definitions.

Parameters

- *path*: File path

```
export_ply_geometries ( filename, plies=[], boundary=True, surface=True, offset_type='middle_offset', direction_arrows=False, first_direction=True, second_direction=False, arrow_length=1.0, arrow_type='no_arrow')
```

Exports the surface, boundary and/or fiber directions of modeling, production and analysis ply to igs or step file.

Parameters

- *filename*: File path (allowed extensions are iges, igs, step and stp).
- *plies*: List of plies (allowed are modeling, production and analysis plies).
- *boundary*: Boolean whether to export the boundary. Default is True.
- *surface*: Boolean whether to export the ply surface. Default is True.
- *offset_type*: Offset type (can be 'no_offset', 'middle_offset', 'top_offset' or 'bottom_offset'). Default is 'middle_offset'
- *direction_arrows*: Boolean whether to export the direction arrows. Default is False.
- *first_direction*: Boolean whether to export the first (main) material direction. Default is True.
- *second_direction*: Boolean whether to export the second material direction. Default is False.
- *arrow_length*: Length of the arrows. Default is 1.
- *arrow_type*: Defines the arrow type (can be 'standard_arrow', 'no_arrow', 'half_arrow'). Default is 'no_arrow'.

find_materials (properties)**

Find materials with the given properties or property ranges

Parameters

- *properties*: Arbitrary material properties which must be matched. Note that a single property value can be given as string, number or min-max range

Returns A list with materials which match the given properties. If nothing matches an empty list is returned.

Examples:

```
>>> materials = model.find_materials(E1=100000.0, nu12=0.3)
>>> materials = model.find_materials( name='1')
>>> materials = model.find_materials(E1=[200000.0, 220000.0], nu12=0.3, G12=[4500.0,5500.0])
```

format

File format string. Choose one of 'abaqus:inp'/'ansys:cdb'/'ansys:dat'/'nastran:bdf' or 'layup'

get_element_by_point (point)

Returns the element label of the closest element with respect to the given point.

Parameters -*point*: Tuple of the global coordinates

get_layup_from_csv_file(path, delete_all_existing_plies=False, modeling_ply_group=None)

Function that reads the layup data from a csv file and adds the data to the graph

Parameters

- *path*: the path to the file
- *delete_all_existing_plies*: if False only the plies that are present in the ply_list are deleted and rebuilt afterwards, if True all plies are deleted and then rebuilt
- *modeling_ply_group*: key of the mpg_collection dict = the id of the mpg. Only plies of this modeling_ply_group will be imported from the file if none is specified all mpgs are read

import_composite_definitions_from_acp_file(path=None, import_mode='keep_both')

This functions loads the ACP file definitions from an other ACP Model. In the Workbench mode everything but the ANSYS input file and the materials is loaded. In the stand-alone mode everything but the ANSYS input file is loaded

Parameters

- *path*: *.acp file path
- *import_mode*: Defines how to solve conflicts of objects of equal name.

Global Resolution Actions

- *keep_both*: Keep target and source. Default.
- *keep_existing*: Imported entities are ignored
- *overwrite*: Overwrite target with source

layup_plots

Plots

lookup_tables

Dictionary with all Look-Up Tables

material_data

Dictionary with all Material Data defined.

mesh

Mesh of this model.

mesh_query (name, position, selection='all', entity=None, entities=None, simulate=False)

Query arbitrary data from the mesh of the model

Parameters

- *name*: Data type to query
- *position*: Position where data is queried:

selection: The selection set determines the selected nodes and elements.

Can be given as string 'sel0' – 'sel5' or 'all' or can be given as ObjectSelection object such as –
model.selection – scene.active_set.

- *entity*: Specialized queries require the specification of an additional associated entity,

e.g. an oriented element set is needed to compute orientations. Entity can be given as NamedGraphObjects or vertex descriptor.

- *entities*: If a list of entities is given, the query will also compute and return a list of results, with one array for each entity.
- *simulate*: Whether the query is only simulated to test if it will return data..

If this flag is set the mesh_query (...) function will only return 0 or 1.

minimum_analysis_ply_thickness

Section computation minimum analysis ply thickness (in length unit of model)

modeling_ply_groups

Dictionary with all Modeling Ply Groups defined

oriented_elements_sets

Dictionary with all oriented element sets defined

parameters

List of parameters visible to the workbench

path

Path to the data file

plot_dependencies(path=None, parent=None, levels=3)

Generates a graph with all dependencies. The output is a dot, png and pdf file

Parameters

- *path*: file path without file extension
- *parent*: Parent object
- *levels*: Depth levels to look for children

Output A *.dot file which can be opened with Graphviz or similar tools.

Usage

```
>>> model.plot_dependencies (r'C:\mp\hull_dependencies', model.element._sets['HULL'])
```

plybook

PlyBook

post_processing_model

Post-processing model

pre-path

Save path of pre database linked to currently loaded post database

reduced_integration

Reduced integration

relative_thickness_tolerance

Section computation relative thickness tolerance

rosettes

Dictionary with all Rosettes defined.

rules

Dictionary with all Element Sets defined.

sampling_elements

Sampling Element Container

save(path=None)

Save model to ACP Database file: Parameters: – *path*: path to write file (optional)

save_analysis_model(path)

Save actual analysis model to disc

Parameters

- *path*: Save path of the cdb file

save_apdl_commands(path)

Save APDL commands for composite definitions of actual model

Parameters

- *path*: Save path of the cdb file

save_h5_model(path)

Save actual model to HDF5 file. Function is mainly used to exchange composite definitions with ANSYS Workbench

Parameters

- *path*: Save path of the h5 file

save_layup_to_csv_file(path, modeling_ply_group=None)

Function that saves the layup data to a csv file

Parameters

- *path*: the path to the file
- *modeling_ply_group*: optional parameter if left the entire layup is written to the file, else only the layup defined within modeling_ply_group

save_solid_models(directory=None, prefix= ACPSolidModel_ , delete_existing=True, formats=[cdb , h5])

Save solid models to files. Function is used within Workbench updates

Parameters

- *directory*: directory to save the models
- *prefix*: Prefix of the model. Default 'ACPSolidModel_'
- *delete_existing*: Flag if existing models should be deleted. Default True

- *formats*: Available file formats are ' cdb' or h5. Default ['cdb', h5']

scenes

Scenes

section_cuts

Section Cuts

select_elements(selection= sel0 , op= new , labels=None, indices=None, attached_ to=None, x=None, y=None, z=None, element_type= all)

Selects element within active model. (Marks the given selection as SELECTED)

Parameters

- *selection*[The selection to update] Can be given as string sel0 - sel5 or all or can be given as ObjectSelection object such as - model.selection -scene.active_set
- *opSelect* operation. Can be all, new (default), add, remove, intersect, inverse or none
- *labelsList* with element labels to select.
- *indicesList* with element indices to select.
- *attached_toElements* attached to entities / vertices in this list will be selected.
- *xX-range* to select.
- *yY-range* to select.
- *zZ-range* to select.
- *element_type*Element type: solid, shell

select_nodes(selection= sel0 , op= new , labels=None, attached_to=None, x=None, y=None, z=None)

Function selects nodes in graph and marks the given selection as SELECTED.

Parameters

- *selection*[The selection to update] Can be given as string sel0 - sel5 or all or can be given as ObjectSelection object such as - model.selection -scene.active_set
- *opSelect* operation. Can be all, new (default), add, remove, intersect, inverse or none
- *labelsList* with element labels to select.
- *indicesList* with element indices to select.
- *attached_toElements* attached to entities / vertices in this list will be selected.
- *xX-range* to select.
- *yY-range* to select.
- *zZ-range* to select.

selection

Selected objects of this model

sensors

Dictionary with all Sensors.

serialize()

Serialize to Python string

set_unit_system(type, locked=False, **kwargs)

Create a unit system and assign it to the model

User defined units are passed as tuple (str,float) where str stands for the unit name, float is a conversion factor to the corresponding SI unit. Temperature units are passed as tuple (str,float,float) where str stands for the unit name, the first float is the conversion factor to Kelvin and the second float is the zero offset to Kelvin.

Parameters

- *type* type string (si,mks,cgs,umks,mpa,bft,bin,or user), see ANSYS documentation
- *locked* boolean to indicate that this unit system is locked (set if the unit system is imported from ANSYS)
- *length_unit* tuple (str,float) as length unit (for type== user only)
- *mass_unit* (str,float) as mass unit (for type== user only, if mass_unit is given, force_unit has to be None)
- *time_unit* (str,float) as time unit (for type== user only)
- *temperature_unit* (str,float,float) as temperature unit (for type== user only)
- *currency_unit* (str,float) as currency unit (for type== user only)

solid_models

SolidModel

solutions

Solutions

solve(wait=False)

Convenience function to directly solve the current model

solver

Solver instance

unit_system

Unit System assigned to this model

update(objects= all , relations_only=False)

Serialize to Python string

update_results_path

Optional path to file storing update results.

use_default_section_tolerances

Uses angle and thickness tolerances from preferences for section computation if set to true

views

Views

6.5.2. Rosette

```
class compolyx.Rosette(graph, obj, parent=None)
```

Rosette class.

Access:

```
>>> import compolyx
>>> model = db.models[ 'class40.1' ]
>>> model = rosette_1 = model.rosettes[ 'Rosette.1' ]
>>> rosette_2 = model.create_rosette(name= 'Rosette.2' , origin=(1.5, 5.75, 7.), dir1=(-0.4, -0.4,
```

dir1

Direction 1 of the Rosette

dir2

Direction 2 of the Rosette

edge_set

Edge Set for Rosette

enabled

Whether this object is currently enabled or not. SamplingElements are always enabled.

get_global_coordinates(x, y, z)

Evaluates the global coordinates of a point given in local coordinates: Rotation from local to global.

CYLINDRICAL, RADIAL and SPERICAL coord sys type: Give phi and theta in RAD

Parameters

- xlocal x direction (x for PARALLEL, r for CYLINDRICAL, RADIAL and SPERICAL)
- ylocal y direction (y for PARALLEL, phi for CYLINDRICAL, RADIAL and SPERICAL)
- zlocal z direction (z for PARALLEL, CYLINDRICAL, RADIAL and theta for SPERICAL)

Usage >>> rosette.get_global_coordinates(1., 3., 4)

local_direction(point, angle)

Get local orientation for a given relative angle and position in space

locked

Rosette is generated from an imported rosette and cannot be changed.

origin

Origin of the Rosette

rosette_type

Rosette Types can be: spherical,edge_wise,cylindrical,parallel,radial

serialize()

Serialize to Python string

set_Xy()

sets dir2 orthogonal to dir1 as y- and x-axis

set_Xz()

sets dir2 orthogonal to dir1 as z- and x-axis

set_Yz()

sets dir2 orthogonal to dir1 as z- and y-axis

set_xY()

sets dir1 orthogonal to dir2 as x- and y-axis

set_xZ()

sets dir1 orthogonal to dir2 as x- and z-axis

set_yZ()

sets dir1 orthogonal to dir2 as y- and z-axis

6.5.3. LookUpTable1D

class compolyx.**LookUpTable1D**(obj, parent=None)

A LookUpTable to associate arbitrary data to a one-dimensional field of Locations

column_factory

alias of LookUpTable1DColumn

direction

The Direction of the Look Up Table

origin

The Origin of the Look Up Table

tabular_data

a tuple containing a list of column labels and a 2d array with floats for all cells. This is a flattened view of all columns.

6.5.4. LookUpTable3D

class compolyx.**LookUpTable1D**(obj, parent=None)

A LookUpTable to associate arbitrary data to a one-dimensional field of Locations

column_factory

alias of LookUpTable3DColumn

num_min_neighbors

Number of neighbors used for interpolation

search_radius

Search Radius used for interpolation

tabular_data

a tuple containing a list of column labels and a 2d array with floats for all cells. This is a flattened view of all columns.

use_default_search_radius

True if the search radius is estimated automatically

6.5.5. ElementRule Classes

```
class compolyx.ElementRule(graph, obj, parent=None)
```

Base class for Rules

extend

Extend of the rule

include_rule_type

include type

relative_rule_type

relative type

6.5.5.1. ParallelRule

```
class compolyx.ParallelRule(graph, obj, parent=None)
```

Bases: compolyx.rule.ElementRule

Parallel rule

direction

Direction of the Parallel Rule.

neg_distance

Negative distance

origin

Origin of the Parallel Rule.

pos_distance

Positive distance

serialize()

Serialize to Python string

6.5.5.2. CylindricalRule

```
class compolyx.CylindricalRule(graph, obj, parent=None)
```

Bases: compolyx.rule.ElementRule

Cylindrical Rule

direction

Direction of the Cylinder.

origin

Origin of the Cylinder.

radius

Radius of the Cylinder

serialize()

Serialize to Python string

6.5.5.3. SphericalRule

```
class compolyx.SphericalRule(graph, obj, parent=None)
```

Bases: compolyx.rule.ElementRule

Spherical Rule

origin

Origin of the Sphere.

radius

Sphere Radius

serialize()

Serialize to Python string

6.5.5.4. TubeRule

```
class compolyx.TubeRule(graph, obj, parent=None)
```

Bases: compolyx.rule.ElementRule

Tube Rule

edge_set

Edge Set for the Tube Rule

inner_radius

Inner tube-radius

outer_radius

Outer tube-radius

serialize()

Serialize to Python string

6.5.5.5. CutoffRule

```
class compolyx.CutoffRule(graph, obj, parent=None)
```

Bases: compolyx.rule.ElementRule

angle

Cut-Off angle

cutoff_geometry

Cut-off Geometry for the Cut-off Rule

cutoff_rule_type

Cutoff rule type, valid values geometry,variable_taper,taper

direction

Direction of the offset and angle interpolation for variable_taper

distance_type

Distance type for offset and angle interpolation for variable_taper

edge_set

Edge Set for cutoff_rule_type= taper or variable_taper

offset

Cut-Off offset

offset_method

Method to compute the offset of a ply.

offset_type

Take offset from edge set perpendicular to element reference surface (out_of_plane) or in element reference surface (in_plane)

origin

Origin of the offset and angle interpolation for variable_taper cutoff rules

ply_cutoff_type

Cutoff Types can be: productionply_cutoff,analysisply_cutoff

ply_tapering

Use Ply Tapering Orientation

6.5.6. EntitySet

```
class compolyx.EntitySet(graph, obj, parent=None)
```

Base class for entity sets

add(entity)

Add entity to the set

remove(entity)

Remove entity from the set

size

Number of entities

6.5.6.1. ElementSet

```
class compolyx.ElementSet(graph, obj, parent=None)
```

Bases: compolyx.entity_set.EntitySet

Element set class

Exemplary usage

```
>>> m=db.models.values()[-1]  
>>> eset=m.element_sets[ DECK ]  
>>> eset.modify(op= none )  
  
>>> eset.modify(op= new , element_ids=[1,2,3,4])  
  
>>> eset.modify(op= add , element_sets=[ m.element_sets[ Deck_layup-1 ] ])  
  
>>> eset.modify(op= intersect , x=[-6.5,-5.5])
```

boundaries

Get the boundaries of the Element Set

locked

Element Set is imported and cannot be changed.

middle_offset

Middle offset flag

modify(op= new , element_ids=None, element_sets=None, x=None, y=None, z=None)

General method to modify the elements in an element set

Parameters

- op: Selection method: new, add, remove, intersect or inverse
- element_ids: List of element ids
- element_sets: List of element sets
- Min and max of x location
- Min and max of y location
- Min and max of z location

normals

Get the Normals of the Element Set

orientable

True if the Element Set has an orientable topology

partition()

Partitions this ElementSet into new ElementSets with an orientable topology if this ElementSet is already orientable, a copy will be created

planar

True if the Element Set has a planar topology

serialize()

Serialize to Python string

write_boundaries(filename, format=None)

Write boundaries in iges/step format :Parameters: - filename: output file - format: iges , step , None (automatic format recognition)

6.5.6.2. EdgeSet**class compolyx.graph, obj=None, parent=None)**

Edge Set class

display_data

The edge set mesh plot

edge_set_type

Edge Set Types can be: By Reference , By Nodes

get_nodes()

Return python list with nodes as objects

is_closed

Edge Set is closed.

is_closed

Edge Set is closed.

limit_angle

Edge Set limit angle for creation of edge set by reference

locked

Edge Set is imported and cannot be changed.

mesh

The edge set mesh

origin

Edge Set origin for creation of edge set by reference

serialize()

Serialize to Python string

6.5.6.3. CADGeometry**class compolyx..CADGeometrygraph, obj, parent=None)****changed**

Status boolean. Set to true if the underlying data has been changed. Write only property

display_data

The edge set mesh plot

is_solid

True if geometry is a solid body.

locked

CAD geometry is generated from an imported geometry and cannot be changed.

path

The file path where the CAD geometry is loaded from.

precision

Precision of geometrical operations (intersection points, thickness sampling, ...).

scale_factor

Geometry is scaled with this factor

shape_type

Topological type of the shape.

show_normals

Visibility of Face Normals.

use_default_precision

Whether to use default precision value or not.

visualization_mesh

Visualization mesh of this geometry

visualization_normals

Visualization normals (point and direction).

6.5.7. OrientedElementSet

class compolyx.**OrientedElementSet**(graph, obj, parent=None)

Class to represent Oriented Element Set

add_element_set(element_set)

Add Element Set to Oriented Element Set

add_rosette(rosette)

Add Rosette to Oriented Element Set

add_rule(rule)

Add Rule to Oriented Element Set

auto_draping_direction

Automatic selection of draping direction

boundaries

Get the boundaries of the Oriented Element Set

clear_element_sets()

Clear Element Sets of Oriented Element Set

clear_roslettes()

Clear Rosettes of Oriented Element Set

clear_rules()

Clear Rule of Oriented Element Set

draping

Flag for using draping or not

draping_direction

The direction in which the draping starts.

draping_mesh_size

The mesh size for draping.

draping_obj

Draping representation

draping_seed_point

The seed point where the draping starts.

element_sets

Element Sets of the oriented element set.

elements

Elements of the Oriented Element Set.

normal_from_id(id)

Returns the element normal

normals

Get the Normals of the Oriented Element Set

orientation_direction

The Orientation Direction of the Oriented Element set.

orientation_point

The Orientation Point of the Oriented Element Set.

orientations

Get the oriented normals of the Oriented Element Set

ref_directions

Get the Reference Directions of the Oriented Element Set

reference_direction_field

a look-up table column or None for external reference directions

remove_element_set(element_set)

Remove Element Set from Oriented Element Set

remove_rosette(rosette)

Remove Rosette from Oriented Element Set

remove_rule(rule)

Remove Rule from Oriented Element Set

rosette_selection_method

Selection Method for Rosettes of the Oriented Element Set.

rosettes

Rosettes of the Oriented Element Set.

rules

Rules of the Oriented Element Set.

save_flat_wrap (filename)

Write the flatwrap to DXF file

Parameters

- *filename*: Path to the file to be written

serialize()

Serialize to Python string

write_boundaries(filename, format=None)

Write boundaries in iges/step format:Parameters: - filename: output file - format: iges , step , None
(automatic format recognition)

6.5.8. ModelingPlyGroup

class compolyx.**ModelingPlyGroup**(graph, obj, parent=None)

Class to manage modeling ply groups.

Access:

```
>>> import compolyx  
>>> db = compolyx.DB()  
>> model = db.models[ class40.1 ]  
>>> mpg = model.modeling_ply_groups[ PlyGroup.1 ]
```

Creation:

```
>>> import compolyx  
>>> db = compolyx.DB()  
>> model = db.models[ class40.1 ]  
>>> mpg_1 = model.modeling_ply_groups[ PlyGroup.1 ]
```

copy_interface_layer(source, global_ply_nr=None, resort=True)

Copy an Interface Layer

Parameters

- *source*: Source object to copy

- *global_ply_nr*: Global ply number to use. If 0 the ply is added at the top.
- *resort*: **Whether to resort all plies of Interface Layer group after copy**. If multiple plies are copied at once it can be useful to resort only once at the end of the copy operation.

Returns New instance of Interface Layer

copy_modeling_ply(source, global_ply_nr=None, resort=True)

Copy a modeling ply

Parameters

- *source*: Source object to copy
- *global_ply_nr*: Global ply number to use. If 0 the ply is added at the top.
- *resort*: **Whether to resort all plies of modeling ply group after copy**. If multiple plies are copied at once it can be useful to resort only once at the end of the copy operation.

Returns New instance of modeling ply

create_interface_layer(name=None, id=None, global_ply_nr=None, oriented_element_sets=None, open_ara_sets=None, active=True)

Create Interface Layer

Parameters

- *name*: Name of the new Interface Layer
- *id*: Optional id of the new Interface Layer.
- *global_ply_nr*: Ply number for stacking sequence.
- *oriented_element_sets*: **Oriented Element Set for the expansion** of the Interface Layer
- *open_area_sets*: Defines the initial crack of a VCCT layer (optional)
- *active*: Interface Layer active. Default True

Returns The created Interface Layer

Example

```
>>> oes_1 = model.oriented_element_sets['OrientedElementSet.1']
>>> mpg = model.modeling_ply_groups['PlyGroup.1']
>>> mp_1 = mpg.create_interface_layer( name='InterfaceLayer.1', global_ply_nr=0, oriented_element_sets=(oes_1))
```

create_modeling_ply(nname=None, id=None, ply_material=None, ply_angle=0.0, number_of_layers=1, global_ply_nr=None, oriented_element_sets=None, rules=None, draping='no_draping', draping_seed_point=None, auto_draping_direction=True, draping_thickness_correction=True, draping_direction=None, draping_mesh_size=None, thickness_definition='nominal', core_geometry=None, active=True, taper_edges=None, thickness_field=None, thickness_field_type='absolute', angle_1_field=None, angle_2_field=None)

Create modeling ply

Parameters

- name: Name of the new Modeling Ply
- id: Optional id of the new Modeling Ply
- ply_material: Ply Material (Fabric, Stackup, SubLaminate)
- ply_angle: Angle of the Ply Material
- number_of_layers: Multiplier of this layer
- global_ply_nr: Ply number for stacking sequence
- oriented_element_sets: Oriented Element Set for the expansion of the Modeling Ply
- rules: Rules for the Modeling Ply
- draping: The type of draping to be used "no_draping", "evaluate_draping", or "tabular_values"
- draping_seed_point: Start/Seed Point for Draping
- auto_draping_direction: Automatically set draping direction (Default: True)
- draping_direction: Direction to go in Draping (Default: None)
- draping_mesh_size: Mesh size used for Draping (Default: Calculated average element size from mesh)
- thickness_definition: Enum that describes the method used for thickness definition (Default: Nominal)
- core_geometry: The assigned core geometry
- active: Modeling Ply active
- taper_edges: Taper Edges for the Modeling Ply
- thickness_field: Look-Up table column with scalar values for thickness sampling (optional)
- thickness_field_type: The type of thickness field 'absolute' or 'relative'
- angle_1_field: Look-Up table column with scalar values for angle 1
- angle_2_field: Look-Up table column with scalar values for angle 2

Returns The created Modeling Ply

Example

```
>>> oes_1 = model.oriented_element_sets[ OrientedElementSet.1 ]
>>> fabric_1 = model.material_data.fabrics[ Fabric.1 ]
>>> mpg = model.modeling_ply_groups[ PlyGroup.1 ]
>>> mp_1 = mpg.create_modeling_ply( name= ModelingPly.1 , ply_angle=0.0,
global_ply_nr=0, number_of_layers=1, ply_material=fabric_1, oriented_element_sets=(oes_1, ),
rules=(rule1,), draping="no_draping", draping_seed_point = (1,0,0),
auto_draping_direction = True)
```

```
export_ply_geometries(filename, ply_level='production_ply', boundary=True, surface=True, offset_type='middle_offset', direction_arrows=False, first_direction=True, second_direction=False, arrow_length=1.0, arrow_type='no_arrow')
```

Exports the surface, boundary and/or fiber directions of modeling, production and analysis ply to igs or step file.

Parameters

- filename: File path (allowed extensions are iges, igs, step and stp).
- ply_level: Defines which plies are exported: modeling_ply_wise, production_ply_wise or analysis_ply_wise. Default is production_ply_wise.
- boundary: Boolean whether to export the boundary. Default is True.
- surface: Boolean whether to export the ply surface. Default is True.
- offset_type: Offset type (can be no_offset, middle_offset, top_offset or bottom_offset). Default is middle_offset.
- direction_arrows: Boolean whether to export the direction arrows. Default is False.
- first_direction: Boolean whether to export the first (main) material direction. Default is True
- second_direction: Boolean whether to export the second material direction. Default is False
- arrow_length: Length of the arrows. Default is 1.
- arrow_type: Defines the arrow type (can be standard_arrow, no_arrow, half_arrow). Default is no_arrow

plies

Modeling Plies of the Modeling Ply Group

reorder_plies(source, target, type= after)

Reorder the ply group. Take source plies and insert before/after target ply. - source: list of plies to insert at new position - target: position to insert plies can be modeling ply or global_ply_nr - type: insert type can be after (default) and before

serialize()

Serialize to Python string

6.5.9. ModelingPly

class compolyx.**ModelingPly**(graph, obj, parent=None, element_vd=None)

Class to represent Modeling Ply

add_oriented_element_set(oriented_element_set)

Add Oriented Element Set

Parameters

- oriented_element_set: The Oriented Element Set to be assigned to ModelingPly

add_rule(rule, template_rule=False, rule_values=())

Add Rule to Modeling Ply

Parameters

- rule: The Rule to be added to the Modeling Ply
- template_rule: Bool
- rule_values: Parameters of the template rule

add_taper_edge(taper_edge, angle, offset=0.0)

Add Taper Edge to Modeling Ply

Parameters

- taper_edge: The Taper Edge to be added to the Modeling Ply
- angle: Angle for tapering
- offset: Offset for tapering

angle_1_field

Angle 1 Correction field

angle_2_field

Angle 2 Correction field

area

Area of the Modeling Ply

auto_draping_direction

Automatic selection of draping direction.

clear_oriented_element_sets()

Clear all Oriented Element Sets of the Modeling Ply

clear_rules()

Clear all Rules assigned to the Modeling Ply

clear_taper_edges()

Clear all taper_edges assigned to the Modeling Ply

core_geometry

Assigned Core Geometry

direction_arrows(arrow_length=None, arrow_type= standard_arrow , offset_type= no_offset)

Direction arrows of the ply

Parameters

- *arrow_length*: length of the arrow
- *arrow_type*: standard_arrow (default), no_arrow , half_arrow
- *offset_type*: no_offset (default), bottom_offset , middle_offset , top_offset

draped_fiber_directions

Get the Draped Fiber Directions of the Modeling Ply

draping

Type of draping to be used

draping_direction

The direction in which the draping starts.

draping_direction_from_calculation(analysis_ply=None)

Draping direction used for draping calculation

draping_mesh_size

The mesh size for draping.

draping_obj

Draping properties of the Modeling Ply

draping_seed_point

The seed point where the draping starts

draping_seed_point_from_calculation(analysis_ply=None)

Draping seed point used for draping calculation

draping_thickness_correction

Thickness correction for draping.

element_normal_is_equal(element_id=None, normal=None)

Returns 1 if the element normal is equal the orientation of the modeling ply, else -1

Parameters

- *element_id*: Element label
- *normal*: Reference normal direction

elements

Elements of the Modeling Ply.

fiber_directions

Get the Fiber Directions of the Modeling Ply

number_of_layers

Number of layers of the Modeling Ply

on_sampling_element

Flag if the modeling ply is on sampling element

orientation_at_element(element_id=None)

Returns the orientation of this modeling ply for a certain element. If the element does not belong to the modeling ply the return value is [0,0,0]

Parameters

element_id: Element label

orientations

Get the oriented normals of the Modeling Ply

oriented_element_sets

Oriented Element Sets of the Modeling Ply

ply_angle

Ply Angle of the Modeling Ply

ply_offsets

Get the offset of the Modeling Ply

price

Price of the Modeling Ply

production_plies

Production Plies of the Modeling Ply

ref_directions

Get the Reference Directions of the Modeling Ply

remove_oriented_element_set(oriented_element_set)

Remove Oriented Element Set from Modeling Ply

Parameters

- *oriented_element_set*: The Oriented Element Set to be removed from ModelingPly

remove_rule(rule)

Remove Rule from Modeling Ply

Parameters

- *rule*: The Rule to be removed from Modeling Ply

remove_taper_edge(taper_edge)

Remove taper_edge from Modeling Ply

Parameters

- *taper_edge*: The taper_edge to be removed from Modeling Ply

rules

Rules of the Modeling Ply.

serialize()

Serialize to Python string

taper_edges

Taper Edges of the Modeling Ply.

thickness_definition

Type of thickness-definition to be used

thickness_field

LookUpTable Column with tabular thicknesses or None

thickness_field_type

The type of the Thickness field absolute or relative

weight

Weight of the Modeling Ply

write_boundaries(filename, format=None, offset_type= no_offset , with_direction_arrows=False, arrow_length=None, arrow_type= standard_arrow)

Write boundaries in iges/step format

Parameters

- *filename*: output file
- *format*: iges , step , None (automatic format recognition)
- *offset_type*: no_offset (default), bottom_offset , middle_offset , top_offset
- *with_direction_arrows*: the element directions should be written to
- *arrow_length*: length of the direction arrows (default is average element edge size)
- *arrow_type*: type to be used as arrows (standard_arrow (default), no_arrow , half_arrow)

6.5.10. ProductionPly

class compolyx.**ProductionPly**(graph, obj, parent=None, element_vd=None)

Class to represent Production Ply

analysis_plies

Analysis Plies of the Production Ply

angle

Ply Angle of the Production Ply

area

Area of the production ply

const_thickness

True if this Production Ply has a constant thickness

direction_arrows(arrow_length=None, arrow_type= standard_arrow , offset_type= no_offset)

Direction arrows of the ply

Parameters

- *arrow_length*: length of the arrow
- *arrow_type*: standard_arrow (default), no_arrow , half_arrow
- *offset_type*: no_offset (default), bottom_offset , middle_offset , top_offset y

draping_obj

Draping representation

ply_material

Ply Material of the Production Ply

price

Price of the production ply

save_draping_input_data(filename)

Writes out Modeling Ply to Draping Interface File

Parameters

- *filename*: Path to the file to be written

save_flat_wrap(filename)

Write the flatwrap to DXF file

Parameters

- *filename*: Path to the file to be written

thickness

Thickness of the Production Ply

weight

Weight of the production ply

write_boundaries(filename, format=None, offset_type= no_offset , with_direction_arrows=False, arrow_length=None, arrow_type= standard_arrow)

Write boundaries in iges/step format

Parameters

- *filename*: output file
- *format*: iges , step , None (automatic format recognition)
- *offset_type*: no_offset (default), bottom_offset , middle_offset , top_offset
- *with_direction_arrows*: the element directions should be written to
- *arrow_length*: length of the direction arrows (default is average element edge size)
- *arrow_type*: type to be used as arrows (standard_arrow (default), no_arrow , half_arrow)

6.5.11. AnalysisPly

```
class compolyx.AnalysisPly(graph, obj, parent=None)
```

ComPoLyX Class to represent Analysis Ply

active_in_post_mode

True if failure criteria will be processed for this ply.

angle

Ply Angle of the Production Ply

direction_arrows(arrow_length=None, arrow_type= standard_arrow , offset_type= no_offset)

Direction arrows of the ply

Parameters

- *arrow_length*: length of the arrow
- *arrow_type*: standard_arrow (default), no_arrow , half_arrow
- *offset_type*: no_offset (default), bottom_offset , middle_offset , top_offset

draping_obj

Get the Fiber Directions of the Analysis Ply

material

Ply Material of the Analysis Ply

weight

Ply Material of the Analysis Ply

thickness

Thickness of the Analysis Ply

write_boundaries(filename, format=None, offset_type= no_offset , with_direction_arrows=False, arrow_length=None, arrow_type= standard_arrow)

Write boundaries in iges/step format

Parameters

- *filename*: output file
- *format*: iges , step , None (automatic format recognition)
- *offset_type*: no_offset (default), bottom_offset , middle_offset , top_offset
- *with_direction_arrows*: the element directions should be written to
- *arrow_length*: length of the direction arrows (default is average element edge size)
- *arrow_type*: type to be used as arrows (standard_arrow (default), no_arrow , half_arrow)

6.5.12. SamplingElement

class compolyx.SamplingElement(graph, obj, parent=None)

The Sampling Element allows to pick through the laminate at a certain point to run detailed analyses.

The key features of the sampling element are:

- Layup in the object tree of the closest element to the selected point
- Sampling direction defines the ply order (bottom-up or top-down)
- Reference direction defines the 0 degree axis of the element used for the evaluations (CLT)
- Optional a user-defined coordinate system can be selected to compute the reference direction. The element normal and reference direction defines the result coordinate system.
- 2D plots showing the layup, stress, strain and failure distribution through the laminate.

- Polar plot of the laminate properties.
- Analysis based on the Classical Laminate Theory (CLT).
- Export to ESAComp, CSV file ...

Usage:

```
>>> model.create_sampling_element(name= Sampling Element )
```

clt_query(query= layup , offset_is_middle=True)

Returns the properties of the classical laminate theory:

Parameters

- *query*: query parameter (see below)
- *offset_is_middle*: Bool to set laminate reference to middle for the laminate stiffness evaluation.
- *consider_coupling_effect*: Bool whether to consider the coupling effect or not

Query parameter can be: - *layup*: Return the layup of the laminate (Modeling, Production and Analysis Plies) - *laminate_properties*: Young s, flexural and shear moduli of the laminate - *polar_properties*: E1, E2 and G12 depending on the laminate orientation - *text_labels*: Returns a list with the material names, angles and thicknesses - *stiffness_matrix*: Returns the laminate stiffness matrix (ABD) - *compliance_matrix*: Returns the laminate compliance matrix (inverse of ABD) - *laminate_forces*: Returns a dict with the laminate forces Nx, Ny, Nxy, Mx, My, Mxy, Qx and Qy. Offset is middle is always true for this evaluation.

Usage:

```
>>> se.clt_query(query= polar_properties )
```

consider_coupling_effect

Specifies the coupling effect is considered or not.

create_plot(query={ layup : [mp], polar_properties : [E1 , G12]}, offset_is_middle=True)

Generates 2D-plots with the results of interest

Parameters

- *query*: query parameter
- *offset_is_middle*: Bool to set laminate reference plane to middle
- *consider_coupling_effect*: Bool whether to consider the coupling effect or not

Query Parameter can be: - *layup*: [mp , pp , ap] Modeling Plies, Production Plies and Analysis Plies - *polar_properties*: [E1 , E2 , G12] polar plot of laminate stiffnesses - *strains*: ['e1' , 'e2' , 'e3' , 'e12' , 'e13' , 'e23' , 'el' , 'ell' , 'elli'] - Strain definition name and component - *stresses*: ['s1' , 's2' , 's3' , 's12' , 's13' , 's23' , 'sl' , 'sll' , 'slll'] - Stress definition name and component - *failures*: ['FailureCriteria.1_irf' , 'FailureCriteria.1_rf' , 'FailureCriteria.1_mos' , 'FailureCriteria.1_fm'] - Name of FC and value - *text_labels*: ['material' , 'angle' , 'thickness']

Usage:

```
>>> se.create_plot (query={layup:[mp]}, failure:[FailureCriteria.1_irf])
```

```
>>> se.graph_plot.x_values
>>> se.graph_plot.layer_thicknesses
```

direction

Sampling Element Direction

element_id

Element ID (label) of the Sampling Element

enabled

Whether this object is currently enabled or not. SamplingElements are always enabled.

graph_plot

Graph Plot object used to configure 2D plots.

locked

Sampling Element is generated from an imported source and cannot be changed.

offset_is_middle

Specifies the offset of the reference plane for the CLT analysis.

plies

Plies of the Sampling Element

point

Sampling Element Point

reference_direction

Reference direction

rosette

Rosette of the Sampling Element

solution

Solution of the Sampling Element

update_plot(offset_is_middle, consider_coupling_effect)

Updates the 2D plot

Parameters

offset_is_middle: Bool to set laminate reference plane to middle

consider_coupling_effect: Bool whether to consider the coupling effect or not

use_default_reference_direction

Flag to use default reference direction

6.5.13. SectionCut

```
class compolyx.SectionCut(graph, obj, parent=None, color_table=None)
```

Section Cut Class showing the lay-up in the cutting plane.

core_scale_factor

Get/set the core scale factor

display_data

Section cut plot

elastic_measures

Cross-sectional Measures of Elasticity

enabled

Whether this object is currently enabled or not. SectionCuts are always enabled.

geometric_measures

Cross-sectional Measures of Geometry

in_plane_reference_direction1

Reference direction for cross-sectional measures

in_plane_reference_direction2

Reference direction for cross-sectional measures

intersection_type

Intersection Types can be: in_plane,normal_to_surface

locked

Section cut was imported and cannot be changed.

mass_measures

Cross-sectional Measures of Mass

mesh

Section cut line mesh

normal

Get/set the plane normal

origin

Get/set the plane origin

scale_factor

Scale factor used for visualization of section cuts

section_cut_type

Section Cut Types can be: analysis_ply_wise,modeling_ply_wise,production_ply_wise

6.5.14. Sensor

```
class compolyx.Sensor(graph, obj, parent=None)
```

Sensor object for measuring areas, prices, weights, and centers of gravity

add_entity(entity)

Add entity to Sensor

area

Area covered by all Entities of the Sensor

center_of_gravity

Center of Gravity over all Entities of the Sensor

clear_entities()

Clear all entities of this Sensor

enabled

Whether this object is currently enabled or not. Sensors are always enabled.

entities

Entities of the Sensor

locked

Sensor cut was imported and cannot be changed.

modeling_ply_area

Cumulated area of all modeling-plies involved

price

Price over all Entities of the Sensor

production_ply_area

Cumulated area of all production-plies involved

remove_entity(entity)

Remove entity from sensor

sensor_type

Sensor type

weight

Weight over all Entities of the Sensor

6.5.15. PlyBook

6.5.15.1. PlyBook

```
class compolyx.PlyBook(name='PlyBook', parent=None, reST_ply='', reST_chapter='',
reST_title_page='', scene=None) (6.1)
```

Class to represent a ply book

(6.2)

chapters

Dictionary with all chapters defined.

create_chapter(name, view=None, ply_entities=[])

Add a chapter to the Ply book

Parameters

- *name*: Name of the chapter to be added

- *view*: The view for snapshots of the chapter
- *ply_entities*: List of modeling plies and modeling ply groups for the chapter

generate(filename, format=None)

Generate the complete plybook

Parameters

- *filename*: Output filename
- *format*: pdf, html, odt, txt

reST_chapter

reST chapter template

reST_ply

reST ply template

reST_title_page

reST title page template

6.5.15.2. Chapter

class class compolyx.Chapter(name, parent, view=None, ply_entities=[], id=0) (6.3)

Class to represent plybook chapter (6.4)

generate(reST_chapter, reST_ply, scene, tmp_dir)

generate the reST file for one single ply

ply_entities

Plies/PlyGroups for the chapter

6.6. Solid-model Classes

6.6.1. SolidModel

class compolyx.SolidModel(obj, parent=None)

Solid Model class

active

Solid-Model active

add_element_set(element_set)

Add Element Set to Solid Model

analysis_plies

Analysis Plies of the Solid Model

clear_element_sets()

Clear Element Sets of Solid Model

clear_generated_data()

Function clears generated solid model but keeps all definitions.

connect_butt_joined_plies

Do not make drop-offs between butt-joined plies if set to True

coordinate_system_index

Coordinate System index

copy_extrusion_guide (source)

Copy an Extrusion Guide

Parameters

- *source*: Source object to copy

Returns New instance of an Extrusion Guide

copy_snap_to_geometry_obj (source)

Copy a Snap to Geometry

Parameters

- *source*: Source object to copy

Returns New instance of a Snap to Geometry

create_extrusion_guide(name, edge_set, id= , cad_geometry=None, direction=(0.0, 0.0, 0.0), radius=0.0, depth=1.0)

Create a new extrusion guide

Parameters

- *name* : the name of the extrusion guide
- *edge_set* : an edge set where this guide applies
- *id* : the id of the extrusion guide
- *cad_geometry* : a cad geometry object
- *direction*: Extrusion direction
- *radius* [distance up to which node translations due to the guide will be propagated through the mesh] 0.0 : only the nodes extruded from edge_set will be shifted onto the guide
- *depth* [intensity for the propagation of mesh corrections] 1.0 : linear decay from guide to radius >1.0 : higher reach <1.0 : more local
- *use_curvature_correction* : whether to use curvature correction algorithm to smooth mesh adapted to extrusion guide. Default is False

create_snap_to_geometry_obj(name, cad_geometry=None, oriented_element_set=None, orientation='top', active=True, id='')

Create a new snap-to-geometry object

Parameters

name : the name of the object

id : the id of the object

cad_geometry : a geometry to snap to

oriented_element_set : oriented element set where this snap to applies

orientation : top or bottom

cut_off_geometry_objs

Cut-Off Geometry objects

delete_bad_elements

Boolean whether to delete the erroneous elements.

disable_numbering_updates

Debug option to disable all numbering updates. Note that the status of this option is not saved.

disable_skin_components

Debug option to disable the creation of skin components. Note that the status of this option is not saved.

drop_off_type

Drop-off type. Allowed string values: [outside_ply , inside_ply]

element_index

Element offset index

element_sets

Element Set of the solid model.

ex_type

Extrusion type. Allowed string values: [specify_thickness , modeling_ply_wise , monolithic , production_ply_wise , analysis_ply_wise , material_wise , user_defined]

extrusion_guides

Extrusion Guides

global_dropoff_material

Element Set of the solid model.

material_index

Material index

max_thickness

Maximum thickness of solid elements.

node_index

Node offset index

num_coordinate_systems

Number of Coordinate Systems

num_materials

Number of Materials

num_solid_elements

Number of Solid Elements

num_solid_nodes

Number of Solid Nodes

num_solid_sections

Number of Solid Sections

offset_type

Offset type. Allowed string values: [surface_normal , shell_normal]

ply_group_pointers

Ply group pointers for user defined extrusion.

remove_element_set(elset)

Remove Element Set from the solid model

remove_global_dropoff_material()

Removes the global dropoff material of the SolidModel

section_index

Section offset index

shell_elements()

Function returns the shell elements where this extrusion operates on

snap_to_geometry_objs

Snap to Geometry objects

solid_elements(include_solver_elements)

Return a list with solid element labels of current solid model. The parameter include_solver_elements controls whether all layered elements are returned or if the solver elements are returned for layered elements which have them.

transfer_all_sets

Whether to transfer all edge and element sets to the solid model or not.

transferred_edge_sets

Edge sets to transfer to solid model. Only processed if transfer_all_sets=False.

transferred_element_sets

Element sets to transfer to solid model. Only processed if transfer_all_sets=False.

update_update_status_from_dependent_objects()

update the update-status from dependent objects of the SolidModel

use_default_coordinate_system_index

Use consecutive coordinate system numbering

use_default_element_index

Use consecutive element numbering

use_default_material_index

Use consecutive material numbering

use_default_node_index

Use consecutive node numbering

use_default_section_index

Use consecutive section numbering

use_homogeneous_drop_off_elements

If set the drop off elements are exported as homogeneous structural element (without layered option) to the *cdb file

use_solid_model_prefix

If set the solid model name is used as a prefix for all components exported to the *cdb file

use_solsh_elements

If selected solid-shell elements are written to the *cdb file

warping_limit

Defines the maximum allowable warping limit

write_drop_off_elements

If set to false no drop-off-elements are written to the *cdb file

write_h5_and_cdb_file(path)

Writes the h5 and cdb file for ANSYS. The file extensions are added automatically.

Usage solid_model.write_h5_and_cdb_file(r C: mphull_solid_model)

Parameters

- path: The file path without extension

write_lsdyna_solid_model(path, part_number=1, mat_type= mat_enhanced_composite_damage)

Export solid model as FE model for LS Dyna

- path: Out file path (file extension must be .k)
- part_number: LS Dyna part number (default is 1)
- mat_type: LS Dyna material card type (default is MAT_ENHANCED_COMPOSITE_DAMAGE)

Supported mat_types:

- MAT_ENHANCED_COMPOSITE_DAMAGE
- MAT_COMPOSITE_DAMAGE
- MAT_COMPOSITE_FAILURE_SOLID_MODEL
- MAT_COMPOSITE_DMG_MSC
- MAT_USER_DEFINED_MATERIAL_MODELS

write_solid_model(path, part_number=1, mat_type= mat_enhanced_composite_damage)

Export solid model as FE model for ANSYS

Parameters

- *path*: Out file path (currently allowed formats are .cdb or .h5)
- *part_number*: LS Dyna part number
- *mat_type*: LS Dyna material card type

6.6.2. ExtrusionGuide

```
class compolyx.ExtrusionGuide(graph, name= , id= , edge_set=None, cad_geometry=None, direction=(0.0, 0.0, 0), radius=0.0, depth=1.0, parent=None)
```

Extrusion guide class

active

Extrusion Guide active

cad_geometry

Associated CADGeometry.

depth

intensity for the propagation of mesh corrections, depth=1 leads to a linear decay from the guide to the radius, depth <1 leads to more local corrections

direction

extrusion direction

edge_set

Associated EdgeSet.

radius

radius up to which nodal guide translations are propagated through the mesh

use_curvature_correction

whether to use curvature correction algorithm to smooth mesh adapted to extrusion guide. Default is False

6.6.3. SnapToGeometry

```
class compolyx.SnapToGeometry(graph, name= , id= , cad_geometry=None, oriented_element_set=None, orientation= top , parent=None)
```

SnapToGeometry guide class

active

Snap-To Geometry active

cad_geometry

Associated CADGeometry.

elements

List of affected shell elements

orientation

Orientation. Allowed string values: [top , bottom]

6.6.4. CutOffGeometry

```
class compolyx.CutOffGeometry(graph, name= , id= , cad_geometry=None, orientation= up , parent=None)
```

CutOffGeometry guide class

active

Cut-Off Geometry active

cad_geometry

Associated CADGeometry.

orientation

Orientation. Allowed string values: [down , up]

6.7. Solution Classes

6.7.1. Solution

```
class compolyx.Solution Classes(obj, parent=None, set=-1, format= ansys:rst , path=None, path2=None)
```

Class to compute postprocessings of a finite element solution.

Access:

```
>>> import compolyx  
>>> db = compolyx.DB()  
>>> model = db.models[ class40.1 ]  
>>> sol = db.models[ class40.1 ].add_solution(name= class40.1 , path= class40.rst , format= ansys
```

ID

Id to be displayed in Envelope solution

automatic_reload

Read result data when updating.

clear()

Clear all result data

clear_element_results()

Resets the post-processing results for each layered element

clear_failure_criteria_results()

Resets the failure criteria results for each layered element

enabled

Whether this object is currently enabled or not. Mainly defined through the current application mode pre or post.

export_results_to_csv(definition=None, entities=[], file_path=None)

Exports the results of the selected entities to a csv file.

Parameters

definitions: Selected definition - Deformations, Strains, Stresses, FailureCriteria

entities: Defines the selection for the export. Can be a list of ElementSets, AnalysisPlies,...

file_path: File name

format

File format string. Choose one of abaqus:inp , ansys:cdb or nastran:f06

has_element_nodal_temperatures

Boolean flag if element nodal temperatures are read from the rst file.

load()

Load result data from file

load_factor

Optional load factor within substep of non-linear solution where the nodal solution should be taken from. Only valid for nastran:f06 format. Becomes (False, 0) if not defined.

path

Path to the data file

path2**Path to the data file****query(definition, position= centroid , selection= all , entity=None, entities=None, spot=None, component=None, rosette=None, simulate=False)**

Query results from the solution

Parameters

- *definition*: **The postproce definition defines what results are evaluated.** For the laminate forces use definition = 'laminate_forces'
- *position* [Position where data is queried:]
- *selection* [The selection set determines the selected nodes and elements.] Can be given as string sel0 - sel5 or all or can be given as ObjectSelection object such as - model.selection - scene.active_set
- *entity* [Entity for which results are evaluated.] Currently supported: Analysis ply or analysis ply vertex
- *entities* : If a list of entities is given, the query will also compute and return a list of results, with one array for each entity.

- *spot*: Used to identify bot, mid or top when querying layered shells
- *component*: Components to query.
 - Valid components for DEFORMATION evaluations: - x, y, z, usum, rotx, roty, rotz - all -> (nx6)
- translations, rotations -> (nx3)
 - Valid components for STRAIN evaluations: - e1, e2, e3, e12, e23, e13, el, ell, ell, von_mises -> (nx1) - all -> (nx6) - principals -> (nx3)
 - Valid components for STRESS evaluations: - s1, s2, s3, s12, s23, s13, sl, sll, sll -> (nx1) - all -> (nx6) - principals -> (nx3)
 - Valid components for FAILURE CRITERIA evaluations: - irf (Inverse reserve factor) ->(nx1) - rf (Reserve factor) ->(nx1) - mos (Margin of safety) ->(nx1) - fm (Failure mode) ->(n x string(size<=4)) - li (Layer index) ->(n x 1) (Only available for element queries where no entity is given.)
 - Valid components for LAMINATE FORCES evaluations: - all -> (nx8)
- *rosette* : If a rosette is given, the results are evaluated with respect to this coordinate system (not recommended for non-linear results)
- *simulate* [Whether the query is only simulated to test if it will return data.] If this flag is set the query(...) function will only return 0 or 1.

Usage

```
>>>solution.query(definition='laminate_forces',position='centroid',selection='sel0'

>>>solution.query(definition=model.definitions['FailureCriteria'], position='centroid

>>>solution.query(definition=model.definitions['Stresses.1'], position='element_results
```

read_stresses_strains

True if the stresses and strains are to be read from rst file. Only valid for ansys:rst format.

recompute_iss_of_solids

Use laminate-based computation method to recalculate the interlaminar shear stress distribution.

reload()

Reload all result data

serialize()

Serialize to Python string

set

Result set to be read. Only valid for ansys:rst format.

subcase

Optional subcase to read. Only valid for nastran:f06 format.

use_felyx_to_compute_pp_results

True if the stresses and strains are to be computed by felyx. If the stresses and strains are read from rst file, nothing is computed.

use_solid_results

Allows to visualize the post-processing results of layered solid models on the Layered Solid Reference Surface .

6.7.2. EnvelopeSolution

```
class compolyx.EnvelopeSolution(name= EnvSolution.1 , solutions=[], ID=0, parent=None)
```

Merge multiple postprocess results into one.

list

List of selected solutions

parent

Parent object

serialize()

Serialize to Python string

status

Status of the object

update()

Update Python-only object scene

6.8. Scene Classes

6.8.1. Scene

```
class compolyx.Scene(Scene(name, model, title= , solution=None, components=[None, None, None], fields=[None, None, None], view=None, parent=None, active_set=None)
```

ComPoLyX scene class.

Access:

```
>>> import compolyx  
>>> db = compolyx.DB()  
>>> model = db.models[ class40.1 ]  
>>> scene = model.scenes[ Scene.1 ]
```

Create new scene:

```
>>> model.create_scene( Scene.2 ,db.models[ class40.1 ].solutions[ class40.1 ])
```

active_set

Set of active entities

background

Background color

background2

Background color 2

camera

Camera settings

color_tables

Collection of color tables

fit_to_window

Reset the zoom of the window

foreground

Foreground color

logo_type

Logo type: default or black

mode

Current ACP mode (pre or post).

model

Service to query for properties

plots

Collection of plot requests

projection

Projection method: parallel or perspective

serialize()

Serialize to Python string

show_global_coordinate_system

Toggle visibility of global coordinate system marker

show_labeled_bounding_box

Toggle visibility of labeled bounding box

show_selected_mesh

Specify whether to show/highlight currently selected Elements

show_solid_elements

Specify whether to highlight Shell or Solid Elements in Selections

snapshot

Current file to save snapshot to. Write only property>

solution

Service to query for properties

status

Status of the object

title

Scene title

update()

Update Python-only object scene

update_direction_plots(entities)

Function synchronizes the following direction plots with the added/removed entities given: - orientations
 - ref_directions - fiber_directions - draped_fiber_directions

view

Apply a view to the scene. Write only

6.8.2. View

```
class compolyx.View(name, position=(0.0, 0.0, 0.0), orientation=(0.0, 0.0, 0.0), rotation_point=(0.0,0.0,0.0), parallel_scale=1.0, projection= perspective , locked=False, parent=None)
```

ComPoLyX class to capture view properties.

Access:

```
>>> import compolyx
>>> db = compolyx.DB()
>>> view1 = db.create_view(name= View.1 , position=[1.5, 5.75, 7.], orientation=[-0.4, -0.4,
```

locked

A View which is imported from an other source can not be modified.

orientation

Get/set the view orientation

parallel_scale

Get/set the view parallel perspective scale factor.

position

Get/set the view position

projection

Get/set the projection method parallel or perspective

rotation_point

Get/set the view rotation point.

serialize()

Serialize to Python string

6.9. Postprocessing Definition Classes

6.9.1. CombinedFailureCriteria

```
class compolyx.CombinedFailureCriteria(graph, obj, failure_criteria=[], parent=None)¶
```

CombinedFailureCriteria class

enabled

Whether this object is currently enabled or not. Mainly defined through the current application mode pre or post.

serialize()

Serialize to Python string

6.9.2. MaxStressCriterion

```
class compolyx.MaxStressCriterion(s1=1, s2=1, s3=0, s12=1, s13=0, s23=0, wf_s1=1.0, wf_s2=1.0,  
wf_s3=1.0, wf_s12=1.0, wf_s13=1.0, wf_s23=1.0)
```

Max stress failure criterion configuration Properties are s1, s2, s3, s12, s13, s23, wf_s1, wf_s2, wf_s3, wf_s12, wf_s13, wf_s23 e.g. MaxStressCriterion(s1=1, s2=1, s3=0, s12=1, s13=0, s23=0, wf_s1=1, wf_s2=1, wf_s3=1, wf_s12=1, wf_s13=1, wf_s23=1)

s1

Specifies whether to compute max stress in 1 direction

s12

Specifies whether to compute max shear stress in 12 direction

s13

Specifies whether to compute max normal stress in 13 direction

s2

Specifies whether to compute max stress in 2 direction

s23

Specifies whether to compute max normal stress in 23 direction

s3

Specifies whether to compute max stress in 3 direction

serialize()

Serialize to Python string

wf_s1

Weighting factor of s1

wf_s12

Weighting factor of s12

wf_s13

Weighting factor of s13

wf_s2

Weighting factor of s2

wf_s23

Weighting factor of s23

wf_s3

Weighting factor of s3

6.9.3. MaxStrainCriterion

```
class compolyx.MaxStrainCriterion(e1=1, e2=1, e3=0, e12=1, e13=0, e23=0, wf_e1=1.0, wf_e2=1.0,
wf_e3=1.0, wf_e12=1.0, wf_e13=1.0, wf_e23=1.0, eXt=0.0, eXc=0.0, eYt=0.0, eYc=0.0, eZt=0.0, eZc=0.0,
eSxy=0.0, eSxz=0.0, eSyz=0.0, force_global_strain_limits=False)
```

Max strain failure criterion configuration

s1

Specifies whether to compute max strain in 1 direction

s12

Specifies whether to compute max shear 12 strain

s13

Specifies whether to compute max shear 13 strain

s2

Specifies whether to compute max strain in 2 direction

s23

Specifies whether to compute max shear 23 strain

s3

Specifies whether to compute max strain in 3 direction

eSxy

Global limit shear strain in material 12 direction

eSxz

Global limit shear strain in material 13 direction

eSyz

Global limit shear strain in material 23 direction

eXc

Global limit compression strain in material 1 direction

eXt

Global limit tension strain in material 1 direction

eYc

Global limit compression strain in material 2 direction

eYt

Global limit tension strain in material 2 direction

eZc

Global limit compression strain in material 3 direction

eZt

Global limit tension strain in material 3 direction

force_global_strain_limits

Force to use global strain limits

serialize()

Serialize to Python string

wf_e1

Weighting factor of e1

wf_e12

Weighting factor of e12

wf_e13

Weighting factor of e13

wf_e2

Weighting factor of e2

wf_e23

Weighting factor of e23

wf_e3

Weighting factor of e3

6.9.4. TsaiWu

```
class compolyx.TsaiWu(dim=2, wf=1.0)
```

Tsai Wu failure criterion configuration

dim

Dimension of the Tsai-Wu failure criterion (2 or 3)

serialize()

Serialize to Python string

wf

Weighting factor

6.9.5. TsaiHill

```
class compolyx.TsaiHill(dim=2, wf=1.0)¶
```

Tsai Hill failure criterion configuration

dim

Dimension of the Tsai-Hill failure criterion (2 or 3)

serialize()

Serialize to Python string

wf

Weighting factor

6.9.6. Hashin

```
class compolyx.Hashin(dim=2, hf=1, hm=1, hd=1, wf_hf=1.0, wf_hm=1.0, wf_hd=1.0)
```

Hashin failure criterion configuration (6.5)

dim

Dimension of the Hashin failure criterion (2 or 3)

hd

Specifies whether to compute delamination

hf

Specifies whether to compute matrix failure

serialize()

Serialize to Python string

wf_hd

Weighting factor

wf_hf

Weighting factor

wf_hm

Weighting factor

6.9.7. Hoffman

```
class compolyx.Hoffman(dim=2, wf=1.0)
```

Hoffman failure criterion configuration

dim

Dimension of the Hoffman failure criterion (2 or 3)

serialize()

Serialize to Python string

wf

Weighting factor

6.9.8. Puck

```
class compolyx.Puck(dim=1, force_global_constants=False, p21_pos=0.325, p21_neg=0.275,
p22_neg=0.225, p22_pos=0.225, s=0.5, M=0.5, interface_weakening_factor=0.8, pf=1, pmA=1, pmB=1,
pmC=1, pd=1, wf_pf=1.0, wf_pmA=1.0, wf_pmB=1.0, wf_pmC=1.0, wf_pd=1.0, cfps=1)
```

Puck failure criterion configuration

M

Degradation factor (Default=0.5)

cfps

Specifies whether to consider the influence of fiber parallel stresses on inter-fiber failure

dim

Dimension of the puck failure criterion (1, 2 or 3)

force_global_constants

Use global Puck constants instead of material specific values.

interface_weakening_factor

Interface weakening factor (Default=0.8)

p21_neg

Inclination of the failure curve for negative normal matrix stresses (Default=0.275)

p21_pos

Inclination of the failure curve for positive normal matrix stresses (Default=0.325)

p22_neg

Inclination of the failure curve for negative normal matrix stresses (Default=0.225)

p22_pos

Inclination of the failure curve for positive normal matrix stresses (Default=0.225)

pd

Specifies whether to compute delamination

pf

Specifies whether to compute fiber failure

pmA

Specifies whether to compute matrix tension failure

pmB

Specifies whether to compute matrix compression failure

pmC

Specifies whether to compute matrix shear failure

s

Degradation factor (Default=0.5)

serialize()

Serialize to Python string

wf_pd

Weighting factor

wf_pf

Weighting factor

wf_pmA

Weighting factor

wf_pmB

Weighting factor

wf_pmC

Weighting factor

6.9.9. Wrinkling

```
class compolyx.Wrinkling(q_homogeneous=0.5, q_honeycomb=0.33, wf_wr=1.0)
```

Configuration of wrinkling failure criterion for sandwich structures

q_homogeneous

Wrinkling coefficient for sandwiches with homogeneous core. Default=0.5

q_honeycomb

Wrinkling coefficient for sandwiches with honeycomb core. Default=0.33

serialize()

Serialize to Python string

wf_wr

Weighting factor

6.9.10. CoreShear

```
class compolyx.CoreShear(ins=False, iss=True, wf_cs=1.0)
```

Configuration of core shear failure criterion for sandwich structures

ins

Whether to compute sandwich core shear criterion under consideration of interlaminar normal stresses.
Default = False

iss

Whether to compute sandwich core shear criterion under consideration of interlaminar shear stresses.
Default = True

serialize()

Serialize to Python string

wf_cs

Weighting factor

6.9.11. Larc

```
class compolyx.Larc(dim=2, lft=1, lfc=1, lmt=1, lmc=1, wf_lft=1.0, wf_lfc=1.0, wf_lmt=1.0, wf_lmc=1.0)
```

LaRC failure criterion configuration

dim

Dimension of the LARC failure criterion (2 or 3)

lfc

Specifies whether to compute fiber compression failure

lft

Specifies whether to compute fiber tension failure

lmc

Specifies whether to compute matrix compression failure

lmt

Specifies whether to compute matrix tension failure

serialize()

Serialize to Python string

wf_lfc

Weighting factor

wf_lft

Weighting factor

wf_lmc

Weighting factor

wf_lmt

Weighting factor

6.9.12. Cuntze

```
class compolyx.Cuntze(dim=2, cft=1, cfc=1, cmA=1, cmB=1, cmC=1, wf_cft=1.0, wf_cfc=1.0, wf_cmA=1.0,  
wf_cmB=1.0, wf_cmC=1.0, b_cross_par=0.1, b_cross_tau=1.0, b_cross_par_tau=0.0, B2D_cross_par=1.1,  
m=3.1)¶
```

Cuntze failure criterion configuration

B2D_cross_par

2D Curve parameter (Default=1.1)

b_cross_par

Curve parameter transverse parallel (between [0.05,0.15], default=0.1)

b_cross_par_tau

Curve parameter transverse parallel shear (between [0.0,0.4], default=0.0)

b_cross_tau

Curve parameter transverse shear (between [1.0,1.6], default=1.0)

cfc

Specifies whether to compute fiber compression failure

cft

Specifies whether to compute fiber tensile failure

cmA

Specifies whether to compute matrix tension failure

cmB

Specifies whether to compute matrix compression failure

cmC

Specifies whether to compute matrix wedge shape failure

dim

Dimension of the cuntze failure criterion (2 or 3)

m

Mode interaction coefficient (between [2.5,4.0], default=3.1)

serialize()

Serialize to Python string

wf_cfc

Weighting factor

wf_cft

Weighting factor

wf_cmA

Weighting factor

wf_cmB

Weighting factor

wf_cmC

Weighting factor

6.9.13. VonMises

```
class compolyx.VonMises(vme=1, vms=1, iss=1, ins=0, wf_vme=1.0, wf_vms=1.0)
```

Von Mises failure criterion configuration

ins

Specifies whether to compute interlaminar normal stresses

iss

Specifies whether to compute interlaminar shear stresses

serialize()

Serialize to Python string

vme

Specifies whether to compute von mises strain criteria

vms

Specifies whether to compute von mises stress criteria

wf_vme

Weighting factor

wf_vms

Weighting factor

6.10. Plot

6.10.1. PlotContainer

6.10.1.1. PlotDataDict

```
class class compolyx.plot_data.PlotDataDict(name, label=None, obj=None, list=[], item_type=<type  
'object'>, parent=None, key_attr='name')
```

copy_plot(source)

Copy a plot object

Parameters

- *source*: Plot object to be copied

Returns Object of the plot duplicate

Example: Make a copy of the thickness plot "My_Thickness_Plot"

```
>>> plot_copy = db.active_model.layup_plots.copy_plot( db.active_model.layup_plots['My_Thickness_Plot'] )
```

```
create_draped_fiber_directions_plot(name=None, id=None, data_scope=[], component=None,  
spot=None, ply_wise=True, title="")
```

Create a Draped Fiber Directions Plot object

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a draped fiber directions plot with Data Scope set to Element Sets "My_ESet1" and "My_ESet2":

```
>>> new_plot = db.active_model.scenes['Scene.1'].static_plots.create_draped_fiber_directions_plot(self, name="MyDr
```

```
create_draped_transverse_directions_plot(name=None, id=None, data_scope=[], component=None,  
spot=None, ply_wise=True, title="")
```

Create a Draped Transverse Fiber Directions Plot object.

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element

- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a draped transverse fiber directions plot with DataScope set to Element Sets "My_ESet1" and "My_ESet2"

```
>>> new_plot = db.active_model.scenes['Scene.1'].static_plots.create_draped_transverse_directions_plot(self,
```

create_fiber_directions_plot(name=None, id=None, data_scope=[], component=None, spot=None, ply_wise=True, title="")

Create a Fiber Directions Plot object.

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a fiber directions plot with Data Scope set to Element Sets "My_ESet1" and "My_ESet2"

```
>>> new_plot = db.active_model.scenes['Scene.1'].static_plots.create_fiber_directions_plot(self, name="MyFiberDirPlot")
```

create_normals_plot(name=None, id=None, data_scope=[], component=None, spot=None, ply_wise=False, title="")

Create a Normals Plot object

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a normals plot with Data Scope set to Element Sets "My_ESet1" and "My_ESet2"

```
>>> new_plot = db.active_model.scenes['Scene.1'].static_plots.create_normals_plot(self, name="MyNormalsPlot")
```

create_orientations_plot(name=None, id=None, data_scope=[], component=None, spot=None, ply_wise=True, title="")

Create a Orientations Plot object

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create an orientations plot with Data Scope set to Element Sets "My_ESet1" and "My_ESet2":

```
>>> new_plot = db.active_model.scenes['Scene.1'].static_plots.create_orientations_plot(self, name="MyOrientationsP
```

create_ref_directions_plot(name=None, id=None, data_scope=[], component=None, spot=None, ply_wise=True, title="")

Create a Reference Directions Plot object.

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a reference directions plot with Data Scope set to Element Sets "My_ESet1" and "My_ESet2":

```
>>> new_plot = db.active_model.scenes['Scene.1'].static_plots.create_ref_directions_plot(self, name="MyRefDirPlot"
```

create_transverse_directions_plot(name=None, id=None, data_scope=[], component=None, spot=None, ply_wise=True, title="")

Create a Transverse Fiber Directions Plot object.

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a transverse fiber directions plot with Data Scope set to Element Sets "My_ESet1" and "My_ESet2"

```
>>> new_plot = db.active_model.scenes['Scene.1'].static_plots.create_transverse_directions_plot(self, name=
```

6.10.1.2. LayupPlotDict

```
class class compolyx.plot_data.LayupPlotDict(name, label=None, obj=None, list=[], item_type=<type  
'object'>, parent=None, key_attr='name')
```

Container for Layup Plots

```
create_angle_plot(name=None, id=None, data_scope=[], show_on_solids=False, show_on_section_cuts=True, show_ply_offsets=False, ply_offset_scale_factor=1.0, title="", add_to_active_set=True, locked=False)
```

Create a Angle Plot object.

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *show_on_solids*: True or False (default). Whether to show the results on the shell or solid model (if present).
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a angle plot with Data Scope set to Element Set "All_Elements"

```
>>> angle_plot = db.active_model.layup_plots.create_angle_plot(self, name="MyPlot", data_scope=db.active_m
```

```
create_thickness_plot(name=None, id=None, data_scope=[], ply_wise=False, show_on_solids=False, title="", add_to_active_set=True, locked=False)
```

Create a Thickness Plot object.

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *ply_wise*: True or False (default). Decides whether to return the thickness of the individual plies or of the whole stack of plies present at an element.
- *show_on_solids*: True or False (default). Whether to show the results on the shell or solid model (if present).
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a thickness plot with Data Scope set to Element Set "All_Elements" in ply_wise mode

```
>>> thick_plot = db.active_model.layup_plots.create_thickness_plot(self, name="MyPlot", data_scope=db.active_model
```

6.10.1.3. PostProcessingPlotDict

```
class compolyx.plot_data.PostProcessingPlotDict(name, label=None, obj=None, list=[], item_type=<type  
'object'>, parent=None, key_attr='name')
```

Container for Post Processing Plots

```
create_deformation_contour_plot(name=None, id=None, active=True, data_scope=[], component=None,  
spot=None, ply_wise=False, show_on_solids=False, title="")
```

Create a Deformation Plot object.

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *component*: 'x', 'y', 'z', 'rotx', 'roty', 'rotz', 'usum'
- *show_on_solids*: True or False (default). Whether to show the results on the shell or solid model (if present).
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a deformation plot with Data Scope set to Element Set "All_Elements" for 'rotx'

```
>>> defo_plot = db.active_model.solutions['Solution.1'].plots.create_deformation_contour_plot(self, name="MyPlot"
```

```
create_failure_plot(name=None, id=None, active=True, data_scope=[], component=None, spot=None,  
ply_wise=False, show_on_solids=False, title="", show_critical_failure_mode=True, show_critical_lay-  
er=False, show_critical_load_case=False, text_threshold=0.0, text_threshold_auto=True, failure_criter-  
ia_definition=None)
```

Create a Failure Criterion Plot object

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *component*: 'irf', 'mos', or 'rf', which are Inverse Reserve Factor, Margin of Safety, and Reserve Factor

- *ply_wise*: True or False (default). Decides whether to evaluate for individual plies or pick the most critical ply per element.
- *show_on_solids*: True or False (default). Whether to show the results on the shell or solid model (if present).
- *title*: The title displayed in the plot legend
- *show_critical_failure_mode*: True (default) or False. Whether to show the text, which indicates the mode of failure
- *show_critical_layer*: True or False (default). Whether to show the text, which indicates the number of the failing ply (only if *ply_wise* = False)
- *show_critical_load_case*: True or False (default). Whether to show the text, which indicates the load case for which failure is predicted (only for Envelope Solutions)
- *text_threshold*: Threshold value above/below which text for an element is hidden. This helps to declutter the text plot.
- *text_threshold_auto*: True (default) or False. If True, suggested *text_threshold* values are used.
- *failure_criteria_definition*: Object of the failure criterion definition to be used for this plot.

Returns The plot object

Example: Create a failure criterion plot with Data Scope set to Element Set "All_Elements", showing the inverse reserve factor with the most critical layer per element for failure criterion "MyFC", and not showing any text

```
>>> fc_plot = db.active_model.solutions['Solution.1'].plots.create_failure_plot(self, name="MyPlot", data_scope=
```

create_strain_plot(name=None, id=None, active=True, data_scope=[], component=None, spot=None, ply_wise=True, show_on_solids=False, title="")

Create a Strain Plot object

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *component*: 'e1', 'e2', 'e3', 'e12', 'e23', 'e13', 'el', 'ell', 'elli'
- *spot*: 'bot', 'top', 'mid' (only if *ply_wise* = True)
- *ply_wise*: True (default) or False. Decides whether to evaluate for individual plies or return the result of the finite element.
- *show_on_solids*: True or False (default). Whether to show the results on the shell or solid model (if present).
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a strain plot with Data Scope set to Element Set "All_Elements" for 'e1' at 'mid' in ply_wise mode:

```
>>> strain_plot = db.active_model.solutions['Solution.1'].plots.create_strain_plot(self, name="MyPlot", data_scope=
```

```
create_stress_plot(name=None, id=None, active=True, data_scope=[], component=None, spot=None,
ply_wise=True, show_on_solids=False, title="", interlaminar_normal_stresses=False)
```

Create a Stress Plot object

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *component*: 's1', 's2', 's3', 's12', 's23', 's13', 'sI', 'sII', 'sIII'
- *spot*: 'bot', 'top', 'mid' (only if *ply_wise* = True)
- *ply_wise*: True (default) or False. Decides whether to evaluate for individual plies or return the result of the finite element.
- *show_on_solids*: True or False (default). Whether to show the results on the shell or solid model (if present).
- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a stress plot with Data Scope set to Element Set "All_Elements" for 's1' at 'mid' in ply_wise mode

```
>>> stress_plot = db.active_model.solutions['Solution.1'].plots.create_stress_plot(self, name="MyPlot", data_scope=
```

```
create_temperature_plot(name=None, id=None, active=True, data_scope=[], component=None,
spot=None, ply_wise=False, show_on_solids=True, title="")
```

Create a Temperature Plot object

Parameters

- *name*: Name of the plot
- *id*: ID for the plot
- *data_scope*: Object(s) defining the scope for which data is computed and returned. Applicable are: Element Set, Oriented Element Set, Modeling Ply, Sampling Element
- *spot*: 'bot', 'top', 'mid' (only if *ply_wise* = True)
- *ply_wise*: True (default) or False. Decides whether to evaluate for individual plies or return the result of the finite element.

- *title*: The title displayed in the plot legend

Returns The plot object

Example: Create a temperature plot with Data Scope set to Element Set “All_Elements” with spot set to ‘mid’ in ply_wise mode

```
>>> temp_plot = db.active_model.solutions['Solution.1'].plots.create_temperature_plot(self, name="MyPlot",
```

6.10.2. PlotData

6.10.2.1. PlotData

```
class compolyx.plot_data.PlotData(graph, obj, parent=None)
```

active

Whether the plot is active or not; an inactive plot is not updated and it does not contain data.

active_display_data

Returns a dictionary containing display data for this plot owned by the active scene

add_data_scope_entity(entity)

Add 1 entity to the Data Scope of a plot.

Parameters

- *entity*: Entity to be added. Applicable are Element Set, Oriented Element Set, Modeling Ply, Production Ply, Analysis Ply, and Sampling Element

Returns The plot object

Example: Add the Element Set “My_ESet” to a my_plot:

```
>>> my_plot.add_data_scope_entity(entity = db.active_model.element_sets['My_ESet'])
```

all_display_data

Returns a list of all display data classes in existence for this object

changed

Status boolean. Set to true if the underlying data has been changed. Write only property

clear_data_scope()

Clears the complete data scope of a plot, i.e. all scope entities are removed at once.

component

The requested result component for a specific plot, e.g. s1 for the stress in the 1-direction

components

Return the possible component set on this plot type

data_scope

Data scope of the plot where results will be evaluated.

display_data_create(parent=None)

Function for internal use, not meant for scripting.

enabled

Whether this object is currently enabled or not. Mainly defined through the current application mode pre or post.

eval_position

The finite element position at which the results are given, e.g. centroid or nodal.

get_data(visible=None, selected=None)

Get data of plot

Parameters

- *visible*: Object(s) defining visible scope
- *selected*: Selected object(s) for ply-wise evaluations.

Returns Resulting n-dimensional array with plot data for each selected object. The scope of the data is evaluated identically as for get_element_xx functions as the intersection of - data_scope of plot - visible_scope defined by visible - selection_scope defined by selected

Data is returned independent of update status of plot, but only if results for the current plot configuration are available, else an empty array is returned.

Examples

Get data for current ply-wise plot with visible scope set to element set "All_Elements" and with ply1 and ply2 selected:

```
>>> data = my_plot.get_data(visible=db.active_model.element_sets['All_Elements'], selected=[ply1, ply2])
```

Get data for current element-wise plot with visible scope equal to all objects visible in current scene:

```
>>> data = my_plot.get_data(visible=db.active_model.active_scene.active_set.entities)
```

get_deformations(visible=None, selected=None)

Function used internally to query matching nodal deformations for a given plot. To query deformations in a script it is recommended to create a separate deformations plot and query its data using normal get_data(...) method.

get_element_indices(visible=None, selected=None)

Get element indices (ACP internal element numbers starting from 0) of plot

Parameters

- *visible*: Object(s) defining visible scope
- *selected*: Selected object(s) for ply-wise evaluations.

Returns Resulting array with element indices from the intersection of - data_scope of plot - visible_scope defined by visible - selection_scope defined by selected

Indices are returned independent of update status of plot, but only if results for the current plot configuration are available, else an empty array is returned.

Examples: Get element indices for current plot with visible scope set to element set "All_Elements" and with ply1 and ply2 selected:

```
>>> eis = my_plot.get_element_indices(visible=db.active_model.element_sets['All_Elements'], selected=[ply1,
```

Get element indices for current plot with visible scope equal to all objects visible in current scene:

```
>>> eis = my_plot.get_element_indices(visible=db.active_model.active_scene.active_set.entities)
```

get_element_labels(visible=None, selected=None)

Get element labels (element numbers as read from / written to mesh files) of plot.

Parameters

- *visible*: Object(s) defining visible scope
- *selected*: Selected object(s) for ply-wise evaluations.

Returns Resulting array with element labels from the intersection of - data_scope of plot - visible_scope defined by visible - selection_scope defined by selected.

Labels are returned independent of update status of plot, but only if results for the current plot configuration are available, else an empty array is returned.

Example: Get element labels for current plot with visible scope set to element set "All_Elements" and with ply1 and ply2 selected:

```
>>> eis = my_plot.get_element_labels(visible=db.active_model.element_sets['All_Elements'], selected=[ply1,
```

Get element labels for current plot with visible scope equal to all objects visible in current scene:

```
>>> eis = my_plot.get_element_labels(visible=db.active_model.active_scene.active_set.entities)
```

get_full_description()

This function returns information displayed together with the legend. Internal use only, not meant for scripting.

has_element_wise

Whether the plot offers element-wise data.

has_ply_wise

Whether the plot offers ply-wise data.

locked

Returns the locked status of the plot

name

Name of object

ply_wise

Whether to plot ply-wise or element-wise.

remove_data_scope_entity(entity)

Remove 1 entity to the Data Scope of a plot.

Parameters

- *entity*: Entity to be removed. Applicable are Element Set, Oriented Element Set, Modeling Ply, Production Ply, Analysis Ply, and Sampling Element

serialize()

Serialize to Python string

serialize_properties()

Serialize to Python string

show_on_solids

Whether to plot results on solid or shell elements.

solution

Solution underlying the plot.

spot

Whether to plot data at bot, mid or top of the layer or laminate.

spots

Return the possible spot set on this plot type

title

The title of the plot.

updated

Status boolean. Set to true if the underlying data has been changed. Write only property

uptodate

Whether the plot is up-to-date.

6.10.2.2. ContourData

```
class compolyx.plot_data.ContourData(graph, obj, parent=None, display_data=None)
```

Bases:

```
compolyx.plot_data.plot_data.PlotData
```

get_ply_offsets(visible=None, selected=None)

Get the offset between plies and reference surface.

Parameters

- *visible*: Object(s) defining visible scope
- *selected*: Selected object(s) for ply-wise evaluations

Returns Resulting n-dimensional array of arrays with the ply-offset vectors for each node within the object/poly-selection: [

```
[array(float, float, float), array(float, float, float), ...]
```

The scope of the data is evaluated as the intersection of - data_scope of plot - visible_scope defined by visible - selection_scope defined by selected

Example: Get the ply-offsets for the current ply-wise plot with visible scope set to element set "All_Elements" and with ply1 and ply2 selected:

```
>>> p_offs = my_contour_plot.get_ply_offsets(visible=db.active_model.element_sets['All_Elements'], selected=[ply1,
```

minmax

Min and max value of current data

ply_offset_scale_factor

Offset scale factor

6.10.2.3. AngleData

```
class compolyx.plot_data.AngleData(graph, obj, parent=None, display_data=None)
```

Bases:

```
compolyx.plot_data.contour_data.ContourData
```

spot

Not applicable to an angle plot

6.10.2.4. ThicknessData

```
class compolyx.plot_data.ThicknessData(graph, obj, parent=None)
```

Bases:

```
compolyx.plot_data.contour_data.ContourData
```

spot

Not applicable to a thickness plot

6.10.2.5. DeformationContourData

```
class compolyx.plot_data.DeformationContourData(graph, obj, parent=None, display_data=None)
```

Bases:

```
compolyx.plot_data.contour_data.ContourData
```

6.10.2.6. StrainData

```
class compolyx.plot_data.StrainData(graph, obj, parent=None, display_data=None)
```

Bases:

```
compolyx.plot_data.contour_data.ContourData
```

6.10.2.7. StressData

```
class compolyx.plot_data.StressData(graph, obj, parent=None, display_data=None)
```

Bases:

```
compolyx.plot_data.contour_data.ContourData
```

interlaminar_normal_stresses

Whether to evaluate normal stresses.

6.10.2.8. FailureData

class compolyx.plot_data.FailureData(graph, obj, parent=None, display_data=None)

Bases:

compolyx.plot_data.contour_data.ContourData

get_text(visible=None, selected=None, str_results=True)

Get text data of failure plot.

Parameters

- *visible*: Object(s) defining visible scope
- *selected*: Selected object(s) for ply-wise evaluations.
- *str_results*: Whether to return failure modes as strings or enums. (default is str)

Returns Resulting n-dimensional array with a dict with data for each selected object/ply:

```
[ dict( "fm" : array(int/str), "li" : array(int), "lc" : array(int) ), dict( "fm" : array(int/str), "li" : array(i
```

Possible keys: - fm: Failure Mode (enum) - li: Layer index - lc: Load case (for envelope solutions only)

The scope of the data is evaluated identically as for get_element_xxx functions as the intersection of:
- data_scope of plot - visible_scope defined by visible - selection_scope defined by selected

Text data is returned independently of the update status of plot, but only if results for the current plot configuration are available, else an empty array is returned.

Example: Get text data for current ply-wise plot with visible scope set to element set "All_Elements" and with ply1 and ply2 selected:

```
>>> text = my_plot.get_text(visible=db.active_model.element_sets['All_Elements'], selected=[ply1, ply2])
```

Get textdata for current element-wise plot with visible scope equal to all objects visible in current scene:

```
>>> text = my_plot.get_text(visible=db.active_model.active_scene.active_set.entities)
```

spot

Not applicable to a thickness plot

show_critical_failure_mode

Whether to evaluate and show the critical failure mode.

show_critical_layer

Whether to evaluate and show the critical layer.

show_critical_load_case

Whether to evaluate and show the critical laod case; available only for Envelope Solutions.

text_threshold

Threshold below/above which failure mode text is shown or not.

text_threshold_auto

Whether the threshold for visualization of failure text is set automatically or not.

6.10.2.9. TemperatureData

class compolyx.plot_data.TemperatureData(graph, obj, parent=None, display_data=None)

Bases:

compolyx.plot_data.contour_data.ContourData

Bibliography

- [1] KD. Potter. *The influence of accurate stretch data for reinforcements on the production of complex structural moldings. Part 1. Deformation of aligned sheets and fabrics.*. Composites. 1979. 10: 161-167.
- [2] Bergsma OK , Huisman J. *Deep drawing of fabric reinforced thermoplastics*. In: Brebbia CA. de Wilde WP. Blain WR editors. *Computer aided design in composite material technology*. New York: Springer. 1988.. p. 323-334..
- [3] Van der Wegen F. *Algorithms for draping fabrics on doubly-curved surfaces*. Int J Numer Meth Eng. 1991., 31:1415-1426.
- [4] Van West BP, Luby SC. *Fabric draping simulation in composites manufacturing Part II. Analytical methods*. J Adv Mater. 1997;. 28(3):36-41.
- [5] AG Prodromou, Chen J. *On the relationship between shear angle and wrinkling of textile composite preforms*,. Composite: Part A 28A. (1997). 491-503.
- [6] Wang J et al.. *The draping of woven fabric preforms and prepgs for production of polymer composite components*,. Composite: Part A 30. (1999);. 757-765..
- [7] Mohammed U. et al. *Experimental studies and analysis of the draping of woven fabrics*,. Composites: Part A 31. (2000). 1409-1420..
- [8] Potter K.. *Bias extension measurements on cross-ply unidirectional prepreg*. Composites: Part A 33. (2002). 63-73.
- [9] Huang NN Tauchert TR. *Thermal stresses in doubly curved cross-ply laminate* Int. J. Solids Structures. 29(8):991-1000,1991.
- [10] Kress G, Roos R, Barbezat M, Dransfeld C, Ermanni P. *Model for interlaminar normal stress in singly curved laminates*. Composite Structures. 69:458-469. 2005.
- [11] Rohwer K. *Improved Transverse Shear Stiffnesses for Layered Finite Elements*. DFVLR-FB 88-32. Braunschweig. 1988.
- [12] Rolfes R Rohwer K. *Improved Transverse Shear Stresses in Composite Finite Elements Based on First Order Shear Deformation Theory* Int. J. for Num. Meth. in Eng. 40:51-60. 1997.
- [13] Roos R, Kress G, Barbezat M, Ermanni P. *Enhanced model for interlaminar normal stress in singly curved laminates*. Composite Structures. October 2007.
- [14] Roos R, Kress G, Ermanni P. *A post-processing method for interlaminar normal stresses in doubly curved laminates*. Composite Structures. 81:463-470. December 2007.
- [15] Camanho P, Lambert L. *A design methodology for mechanically fastened joints in laminated composite materials*. Comp. Sci. Technol.. 66 (2006). pp. 3004-3020.
- [16] Camanho P, Davila C, Pinho S, Iannucci L, Robinson P. *Prediction of in situ strengths and matrix cracking in composites under transverse tension and in-plane shear*. Composites. : Part A 37 (2006). pp. 165-176.

- [17] Cuntze R, Freund A. The predictive capability of failure mode concept-based strength criteria for multidirectional laminates . *Comp. Sci. Technol.* 64 (2004) . 343-377.
- [18] Cuntze R. Efficient 3D and 2D failure conditions for UD laminae and their application within the verification of the laminate design. *Comp. Sci. Technol.*. 66 (2006), No. 7-8. pp 1081-1096.
- [19] Davila C Navin J. Failure Criteria for FRP Laminates in Plane-Stress NASA Langley Research Center. Hampton, 2003.
- [20] Davila C Camanho P, Rose C . Failure Criteria for FRP Laminates, Journal of. *COMPOSITE MATERIALS*. Vol. 39, No. 4/2005.
- [21] *Structural Materials Handbook, Volume 1 -. Polymer Composites*. ESA PSS-03-203, Issue 1. ESA Publications Division, ESTEC. Noordwijk 1994.
- [22] *Structural Materials Handbook* ESA ECSS-HB-304 (Draft 2),ESA publications division, ESTEC. Noordwijk 2009.
- [23] Jones R. *Mechanics of composite materials* Taylor & Francis. Philadelphia 1999. pp. 109-112.
- [24] Pinho S, Davila C, Camanho P, Iannucci L, Robinson P. *Failure models and criteria for FRP under in-plane or three-dimensional stress states including shear non-linearity*. NASA/TM-2005-213530.
- [25] Puck A, Schneider W. On Failure Mechanisms and Failure Criteria of Filament-wound Glass-Fiber/Resin Composites. *Plast Polym.* (Febr. 1969). pp. 33-43.
- [26] Puck A, Festigkeitsberechnung an Glasfaser/Kunststoff-Laminaten bei zusammengesetzter Beanspruchung. *Kunststoffe*. 59 (1969). 11, pp. 780-787.
- [27] Puck A. *Festigkeitsanalyse von Faser-Matrix-Laminaten*. Carl Hanser. Verlag, Munchen Wien 1996.
- [28] Puck A, Schurmann H. Failure analysis of FRP laminates by means of physically based phenomenological models. *Comp. Sci. Technol.* 58 (1998). pp 1045-1067.
- [29] Puck A, Kopp J, Knops . Failure analysis of FRP laminates by means of physically based phenomenological models. *Comp. Sci. Technol.* 62 (2002). pp. 1633-1662.
- [30] Puck A, Kopp J, Knops M. Guidelines for the determination of the parameters in Puck's action plane strength criterion. *Comp. Sci. Technol.* 62 (2002). pp. 371-378.
- [31] Sullins RT et al. *Manual for Structural Stability Analysis of Sandwich Plates and Shells*. NASA CR-145. 1969.
- [32] Theoretical Background of ESAComp Analyses. M. Palantera, Version 1.0. 1998.
- [33] Mechanics of Composite Material. Jones R.M.. Hemisphere, New York, 1975.
- [34] Hoffman O. *The Brittle Strength of Orthotropic Materials*. Journal of Composite Materials, v. 1 1967. pp. 200-206.
- [35] T.A.et al.. *Numerical investigation to prevent crack jumping in Double Cantilever Beam tests of multi-directional composite laminates.* Comp. Sci. and Technol.71 (2011). pp. 1587-1592.

Index

- geometry cutoff, 78
- C**
- composite shell model, 12
- E**
- examples, 9
- F**
- Fabric, 116
features
 usage reference, 101
file information
 usage reference, 105
- G**
- general application, 80
getting started, 7
- H**
- HDF5 Composite CAE File
 usage reference, 108
- I**
- installation, 1
 linux, 4
 sp and subversion, 5
 windows, 1
introduction
 ACP, 7
- L**
- LaRC Constants, 116
license
 ANSYS solver, 5
 composite prepost, 5
licensing, 5
local reinforcements, 69
look-up table, 77
- M**
- Material Data
 usage reference, 108
model
 usage reference, 101
model summary
 usage reference, 106
- O**
- overview
 ACP, 7
- P**
- pre-processing, 13
Puck Constants
 usage reference, 114
Puck for Woven
 usage reference, 114
- R**
- rule, 78
- S**
- section computation
 usage reference, 106
solve
 usage reference, 107
Strain Limits
 usage reference, 112
Stress Limits
 usage reference, 113
- T**
- theory, 219
Tsai-Wu Constants, 115
tutorials, 9
- U**
- units
 usage reference, 106
user reference, 101
- V**
- variable core thickness, 75

