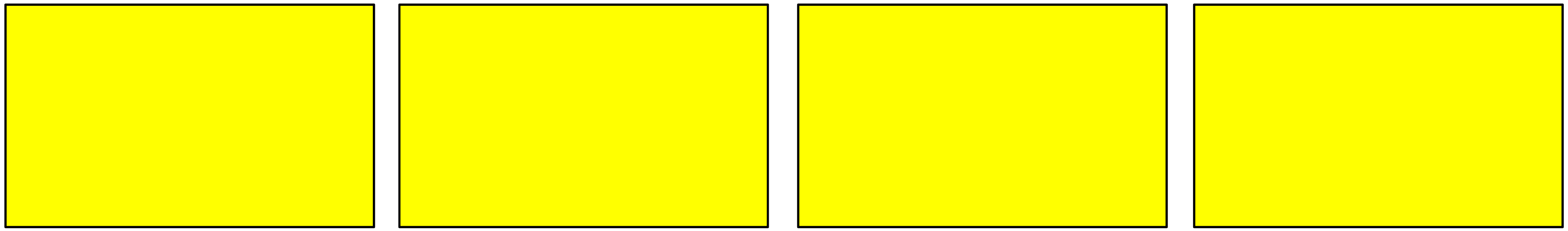
Workshop - MSC Nastran Topometry Optimization of a Composite Panel

AN MSC NASTRAN SOL 200 TUTORIAL

Goal: Use Nastran SOL 200 Optimization

Use Topometry optimization to determine ply shapes

Initial Ply Shapes



Ply 1 and 8
80°

Ply 2 and 7
-65°

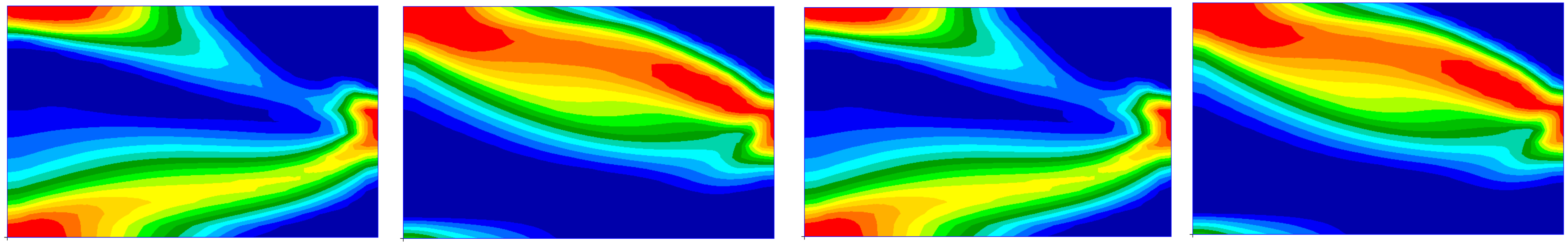
Ply 3 and 6
80°

Ply 4 and 5
-65°

Goal: Use Nastran SOL 200 Optimization

Use Topometry optimization to determine optimal ply shapes

Topometry Results



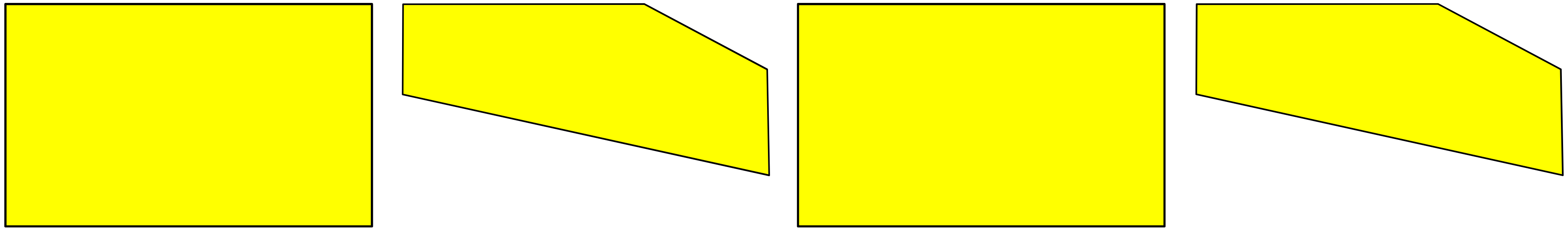
Ply 1 and 8
80°

Ply 2 and 7
-65°

Ply 3 and 6
80°

Ply 4 and 5
-65°

Proposed Ply Shapes



Ply 1 and 8
80°

Ply 2 and 7
-65°

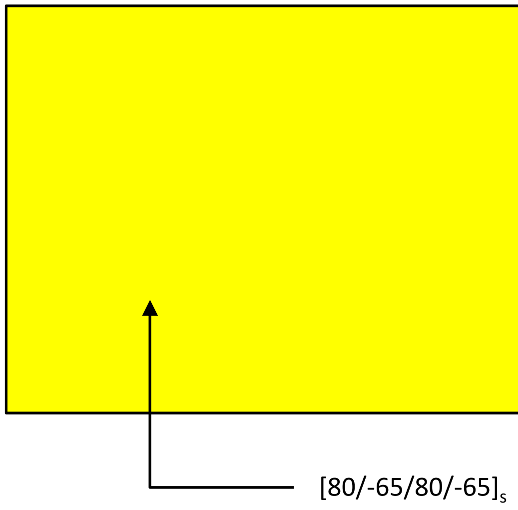
Ply 3 and 6
80°

Ply 4 and 5
-65°

Goal: Use Nastran SOL 200 Optimization

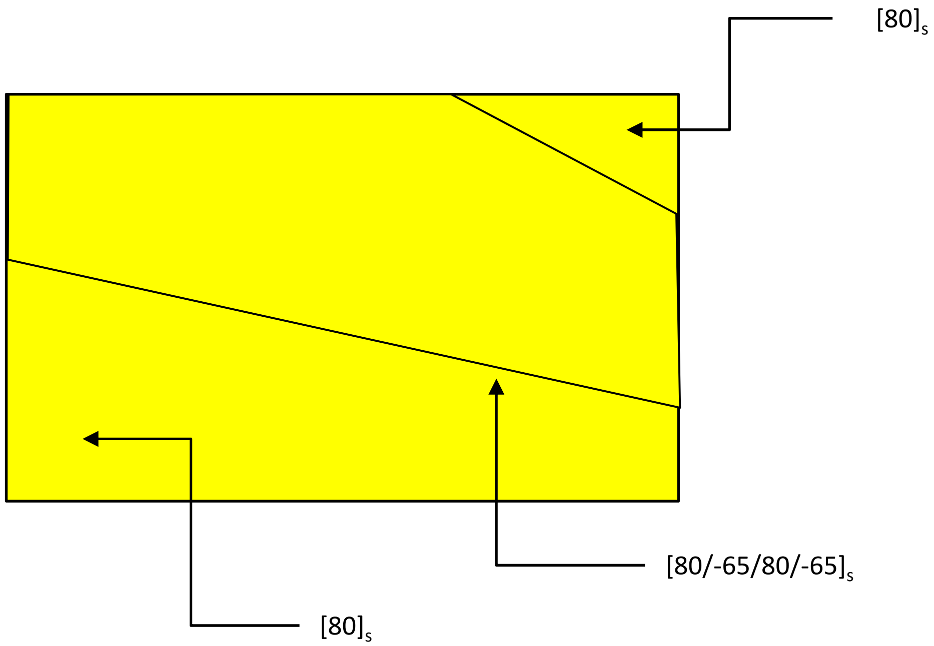
Before Optimization

- Panel of initial layup

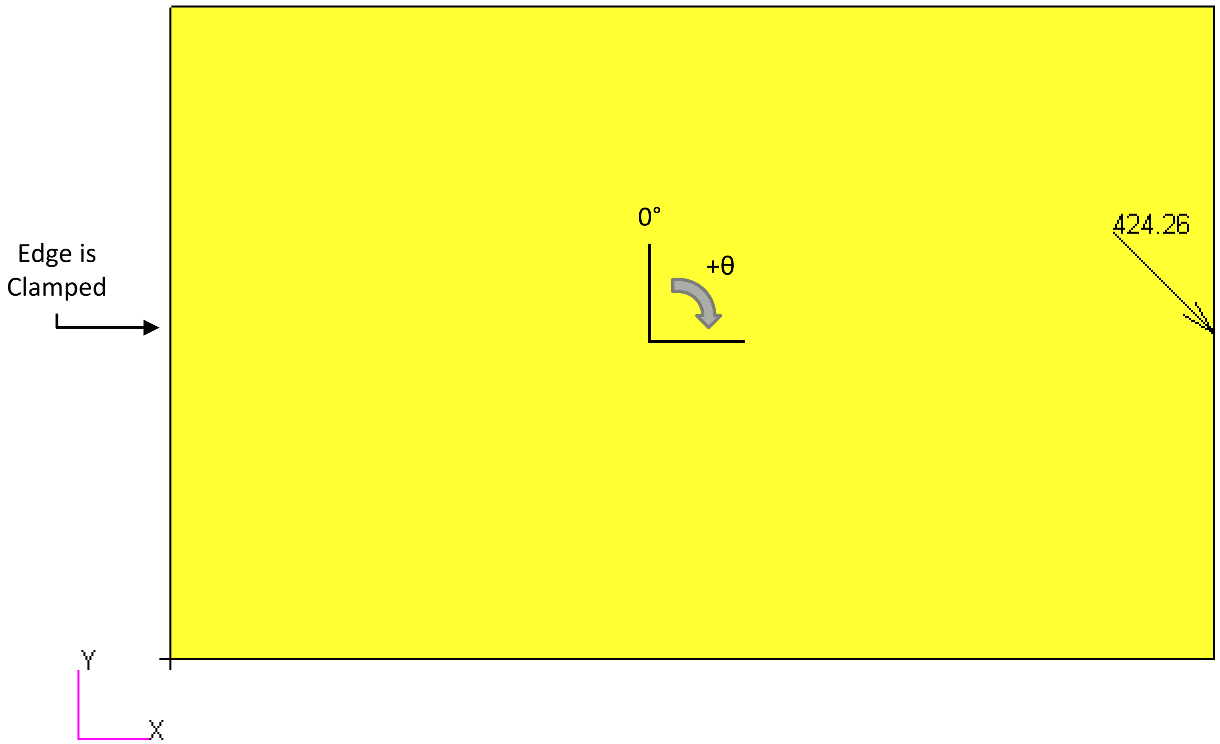


After Optimization

- Panel of updated layups
- The weight is reduced



Details of the structural model



Panel: 16in. x 10in.
Layup: $[80/-65/80/-65]_s$

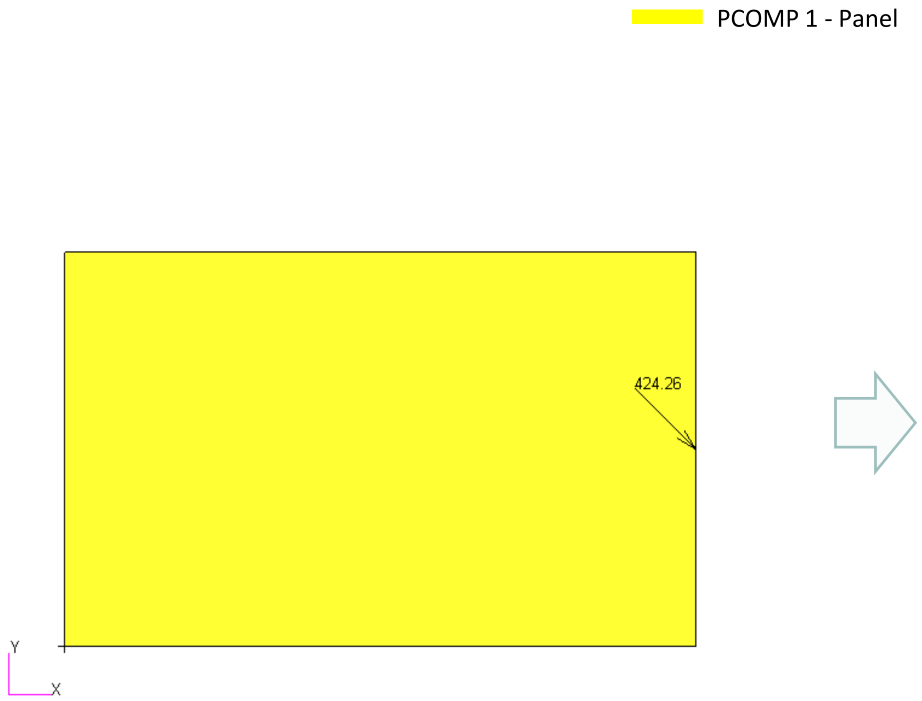
Optimization Problem Statement

Design Region (Topometry)

z1: Thickness of ply 1 and 8
z2: Thickness of ply 2 and 7
z3: Thickness of ply 3 and 6
z4: Thickness of ply 4 and 5

.001 < z1, z2, z3, z4

Variable Linking
z3 = z1
z4 = z2



Design Objective

r0: Minimize compliance
This is equivalent to maximizing stiffness

Design Constraints

r1: Weight

r1 < 38.0 (~50% of original weight 75.0)

Contact me

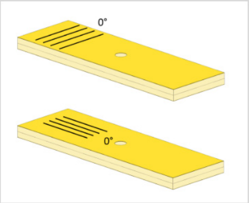
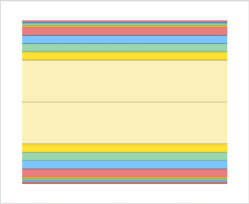
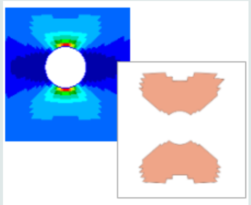
- Nastran SOL 200 training
- Nastran SOL 200 questions
- Structural or mechanical optimization questions
- Access to the SOL 200 Web App

christian@ the-engineering-lab.com

Before Continuing

Consider the New Composite Laminate Optimization Tutorials – Composite Coupon

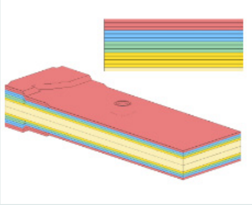
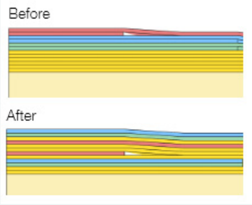
Visit the User's Guide to access the newest tutorials.

	Title and Description	PDF Tutorial	YouTube Tutorial
	**Composite Coupon – Phase A – Determination of the optimal 0° direction of a composite** The goal of this 5-phase tutorial series is to optimize a composite coupon, with a core, and produce a lightweight composite that satisfies failure index constraints. The optimal ply shapes (ply drop-offs) and ply numbers are determined for 0°, ±45°, and 90° plies. A stacking sequence optimization is performed to satisfy manufacturing requirements. One important part of optimizing composites is visualizing the composite plies. This tutorial series also demonstrates the visualization of ply drop-offs, tapered plies and core layers. This first phase involves determining the optimal 0° direction of a composite. It is best practice to align the 0° plies in the direction of the load. Not doing so will more than likely produce a suboptimal composite that is heavier than necessary. This tutorial demonstrates the use of MSC Nastran's optimizer to determine the optimal 0° direction of a composite. An optimization is performed to maximize the stiffness of the composite for multiple load cases and while varying the angle of the 0° plies. Ultimately, the best 0° direction is determined. This is the first phase in a 5-phase tutorial series.	Link	Link
	**Composite Coupon – Phase B – Baseline Ply Number Optimization** This tutorial demonstrates how to configure a basic ply number optimization of continuous plies that span the entire model. The goal of this tutorial is to demonstrate basic actions such as creating variables, a weight objective and constraints on failure index. The results of this ply number optimization serve as a baseline for future comparisons. In a subsequent tutorial, the ply shapes will be optimized to minimize weight. This is the second phase in a 5-phase tutorial series.	Link	Link
	**Composite Coupon – Phase C – Data Preparation for Ply Shape Optimization** This tutorial is a guide to preparing data for ply shape optimization in a subsequent tutorial. The maximum failure index values of the outer plies of the composite are determined and saved to specially formatted PLY000i files. The PLY000i files will be used to construct optimal ply shapes in a subsequent tutorial. This is the third phase in a 5-phase tutorial series.	Link	Link

Before Continuing

Consider the New Composite Laminate Optimization Tutorials – Composite Coupon, Continued

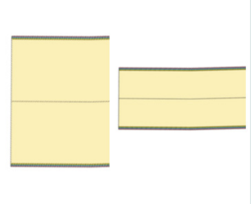
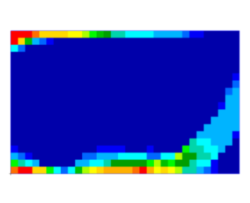
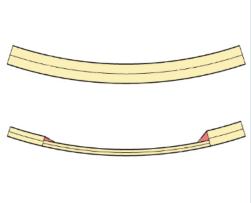
Visit the User’s Guide to access the newest tutorials.

	Title and Description	PDF Tutorial	YouTube Tutorial
	**Composite Coupon – Phase D – Ply Shape and Ply Number Optimization** This tutorial details the process to build optimal ply shapes and perform a ply number optimization. The optimal ply shapes are constructed to follow the contours of the failure indices. The ply number optimization involves minimizing weight and constraining the failure indices of plies. The PLY000i files and BDF files from the previous tutorial, phase C, are used in this tutorial. This is the fourth phase in a 5-phase tutorial series.	Link	Link
	**Composite Coupon – Phase E – Stacking Sequence Optimization** This tutorial involves performing a stacking sequence optimization and is a continuation of the previous tutorial, phase D. A final statics analysis is performed to confirm the optimized composite satisfies failure index constraints. This is the fifth phase in a 5-phase tutorial series.	Link	Link

Before Continuing

Consider the New Composite Laminate Optimization Tutorials – Sandwich Composite Panel

Visit the User's Guide to access the newest tutorials.

	Title and Description	PDF Tutorial	YouTube Tutorial
	**Sandwich Composite Panel – Phase B – Baseline Core Thickness Optimization** The goal of this 3-phase tutorial series is to optimize a curved composite panel, with a core, and produce a lightweight composite that satisfies constraints on the buckling load factor. This tutorial series focuses exclusively on optimizing the thickness of the core. The methods detailed in the tutorial series are applicable to both foam and honeycomb cores. This tutorial demonstrates how to configure a basic core thickness optimization where the core has a constant thickness throughout the entire model. The goal of this tutorial is to demonstrate basic actions such as creating variables, a weight objective and constraints on the buckling load factor. The results of this core thickness optimization serve as a baseline for future comparisons. In a subsequent tutorial, the core will be allowed to have a variable thickness throughout the model and will be optimized to minimize weight. This is the first phase in a 3-phase tutorial series.	Link	Link
	**Sandwich Composite Panel – Phase C – Topometry Optimization to Determine Optimal Core Shape** This tutorial is a guide to preparing data for core shape and core thickness optimization in a subsequent tutorial. A topometry optimization is performed in this tutorial to determine the ideal thickness distribution of the core throughout the entire composite panel while satisfying constraints on the buckling load factor and minimizing weight. The results of a topometry optimization are contained in the PLY000i files and will be used to construct optimal core shapes in a subsequent tutorial. This is the second phase in a 3-phase tutorial series.	Link	Link
	**Sandwich Composite Panel – Phase D – Core Shape and Core Thickness Optimization** This tutorial details the process to build optimal core shapes and perform a core thickness optimization. The optimal core shapes are constructed to follow the contours of thickness results generated by a topometry optimization. The core thickness optimization involves minimizing weight and constraining the buckling load factor. The PLY000i files and BDF files from the previous tutorial, phase C, are used in this tutorial. Comparisons are made between this optimization in phase D and the baseline optimization performed in phase B. This is the third phase in a 3-phase tutorial series.	Link	Link

Tutorial

Tutorial Overview

1. Start with a .bdf or .dat file
2. Use the SOL 200 Web App to:
   - Convert the .bdf file to SOL 200
     - Design Regions/Variables
     - Design Objective
     - Design Constraints
   - Perform optimization with Nastran SOL 200
3. Review optimization results
   - .f06
   - Topometry Optimization and Structural Results

Special Topics Covered

Linked Properties in Topometry Optimization – When configuring a size optimization, there is an option to link variables, i.e. one variable is dependent on the change of another variable. A similar option exists for the thickness properties on the PCOMP entry, enabling users to link ply thicknesses of a composite laminate during a topometry optimization.

SOL 200 Web App Capabilities

The Post-processor Web App and HDF5 Explorer are free to MSC Nastran users.

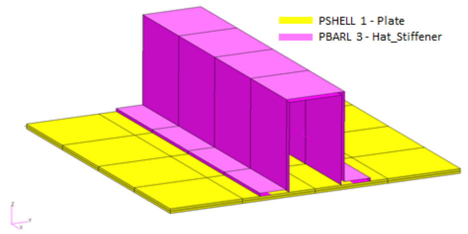
Compatibility

- Google Chrome, Mozilla Firefox or Microsoft Edge
- Windows and Red Hat Linux
- Installable on a company laptop, workstation or server. All data remains within your company.

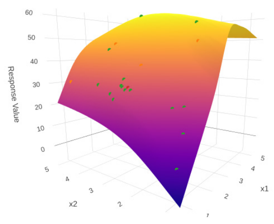
Benefits

- REAL TIME error detection. 200+ error validations.
- REALT TIME creation of bulk data entries.
- Web browser accessible
- Free Post-processor web apps
- +80 tutorials

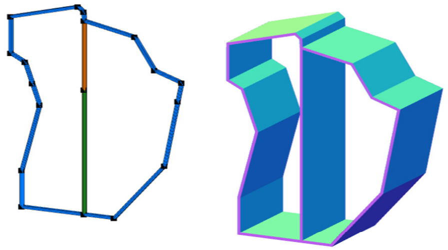
Web Apps



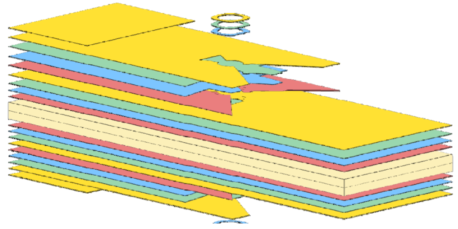
Web Apps for MSC Nastran SOL 200
Pre/post for MSC Nastran SOL 200. Support for size, topology, topometry, topography, multi-model optimization.



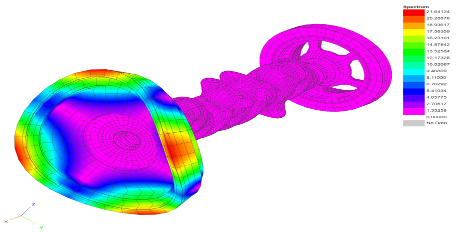
Machine Learning Web App
Bayesian Optimization for nonlinear response optimization (SOL 400)



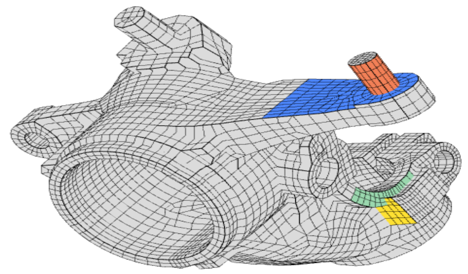
PBMSECT Web App
Generate PBMSECT and PBRSECT entries graphically



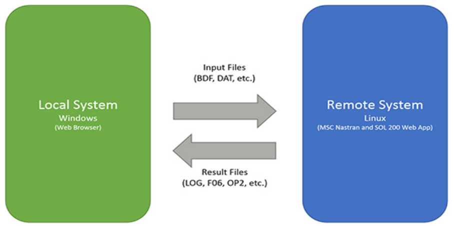
Ply Shape Optimization Web App
Optimize composite ply drop-off locations, and generate new PCOMPG entries



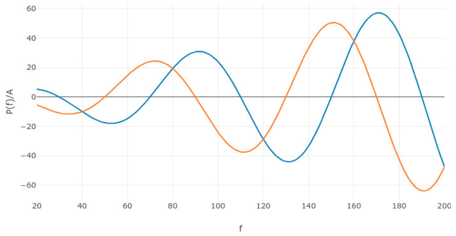
Post-processor Web App
View MSC Nastran results in a web browser on Windows and Linux



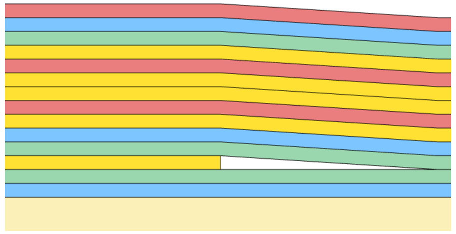
Shape Optimization Web App
Use a web application to configure and perform shape optimization.



Remote Execution Web App
Run MSC Nastran jobs on remote Linux or Windows systems available on the local network



Dynamic Loads Web App
Generate RLOAD1, RLOAD2 and DLOAD entries graphically



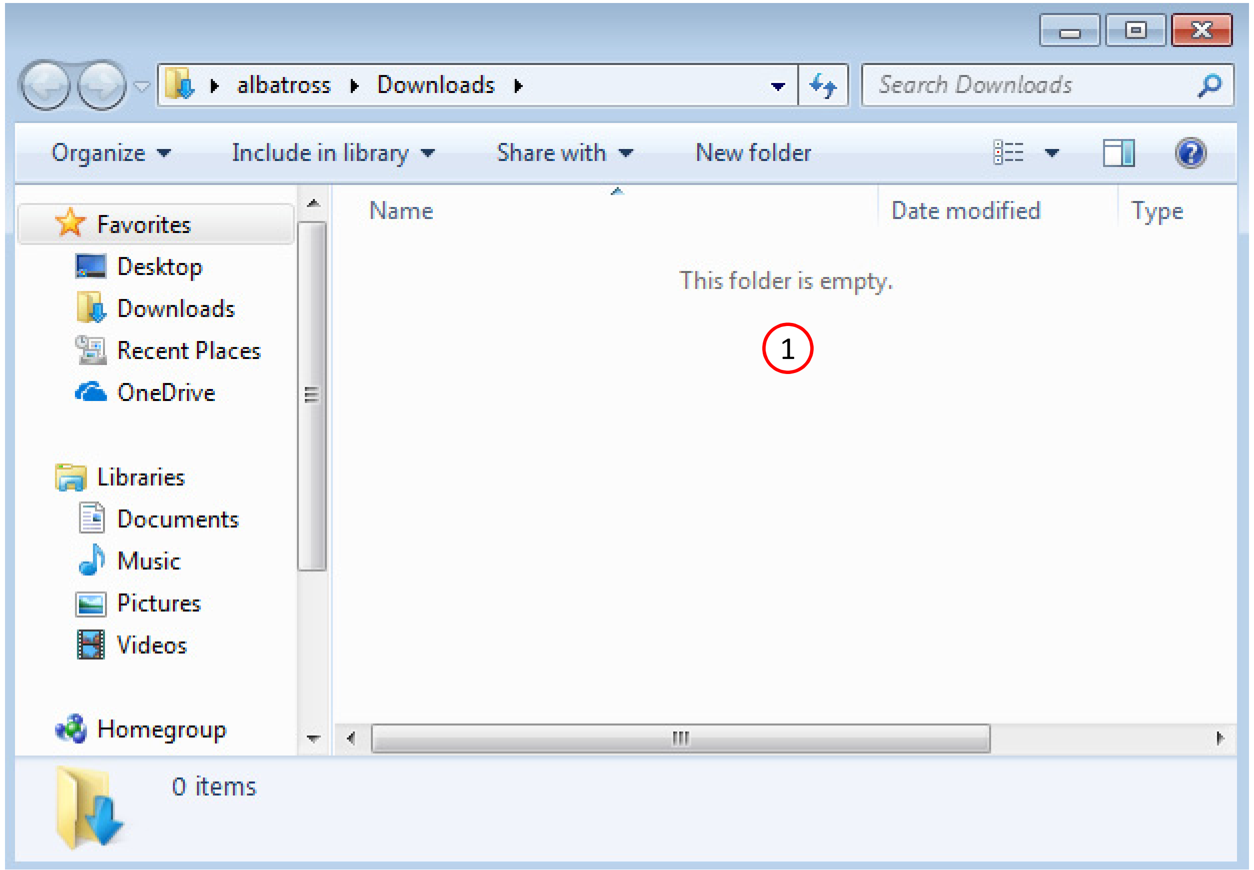
Stacking Sequence Web App
Optimize the stacking sequence of composite laminate plies

HDF5 Explorer Web App
Create graphs (XY plots) using data from the H5 file

Before Starting

1. Ensure the Downloads directory is empty in order to prevent confusion with other files

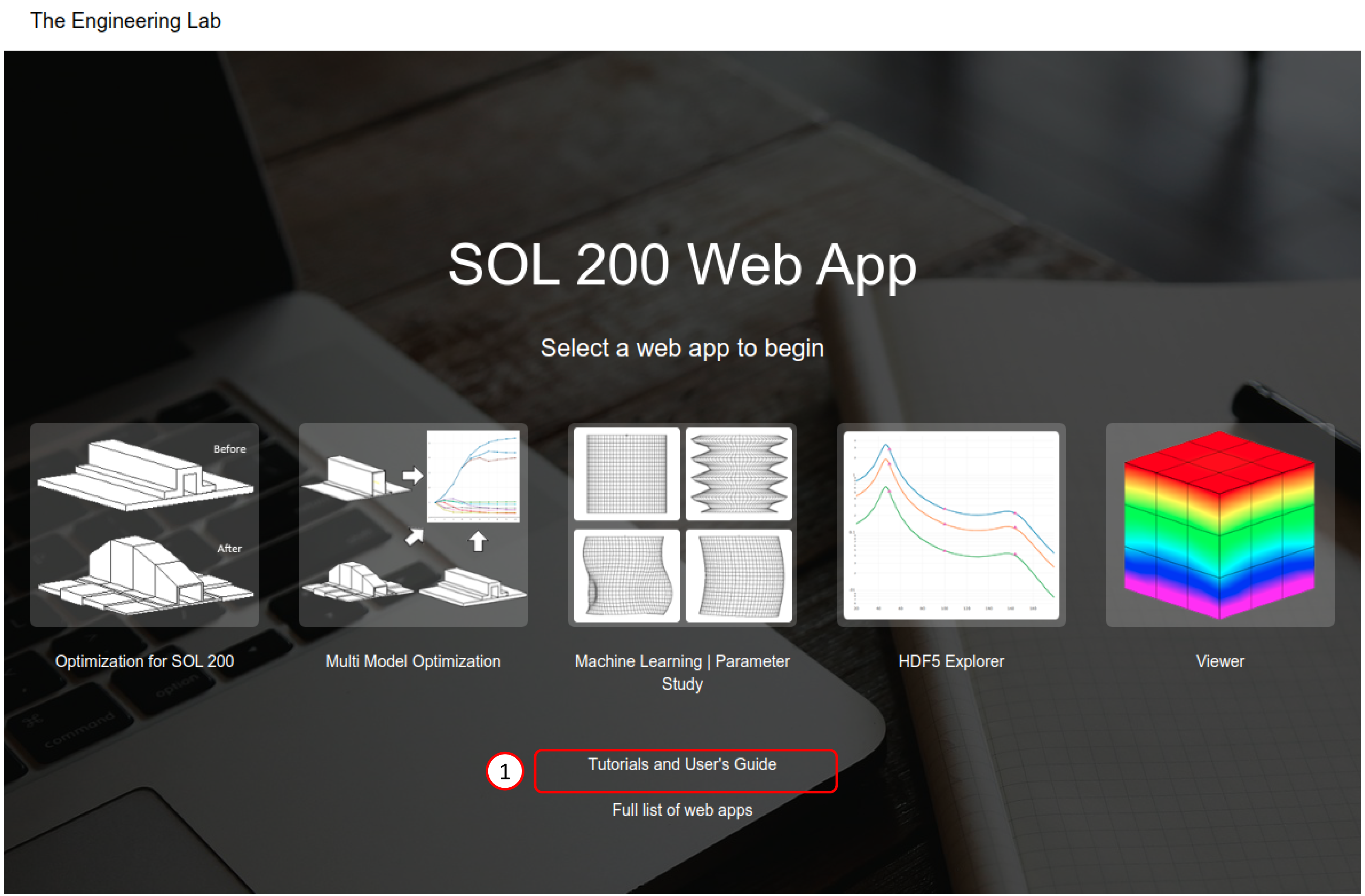
- Throughout this workshop, you will be working with multiple file types and directories such as:
  - .bdf/.dat
  - nastran_working_directory
  - .f06, .log, .pch, .h5, etc.
- To minimize confusion with files and folders, it is encouraged to start with a clean directory.



Go to the User's Guide

1. Click on the indicated link

- The necessary BDF files for this tutorial are available in the Tutorials section of the User's Guide.

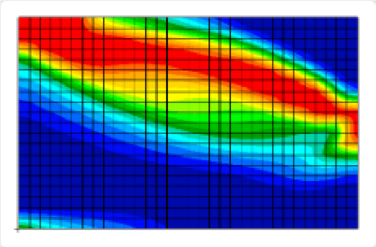


Obtain Starting Files

1. Find the indicated example
2. Click Link
3. The starting file has been downloaded

- When starting the procedure, all the necessary BDF files must be collected together.

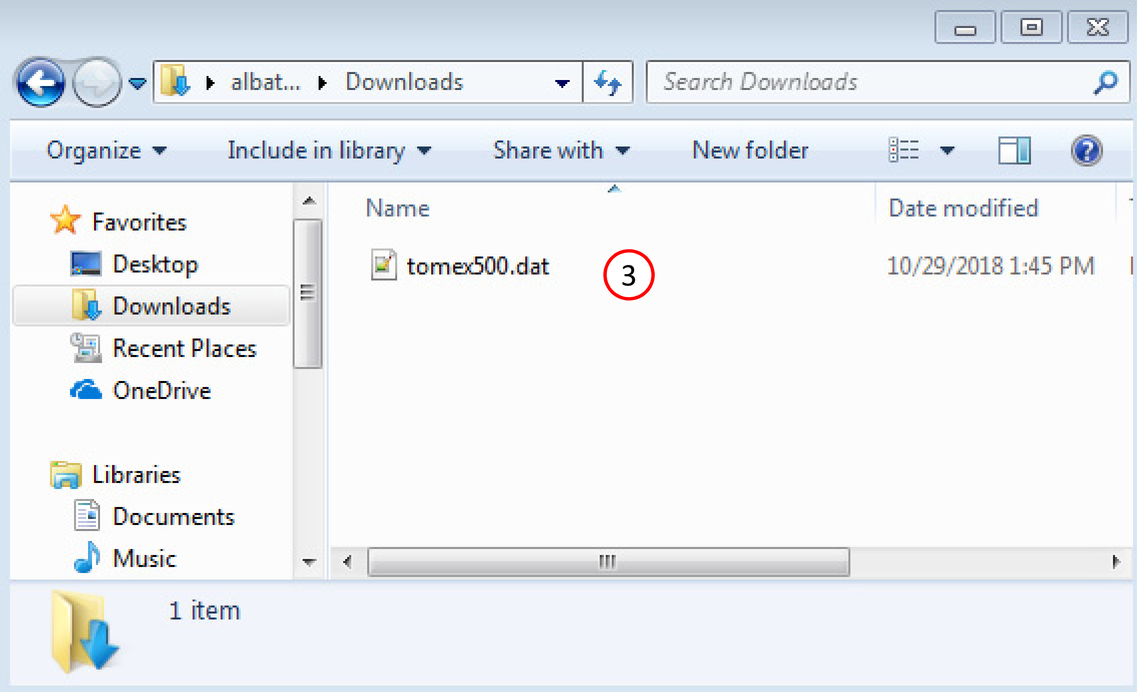
1



MSC Nastran Topometry Optimization of a Composite Panel

This tutorial covers the use of Topometry Optimization to determine ply shapes.

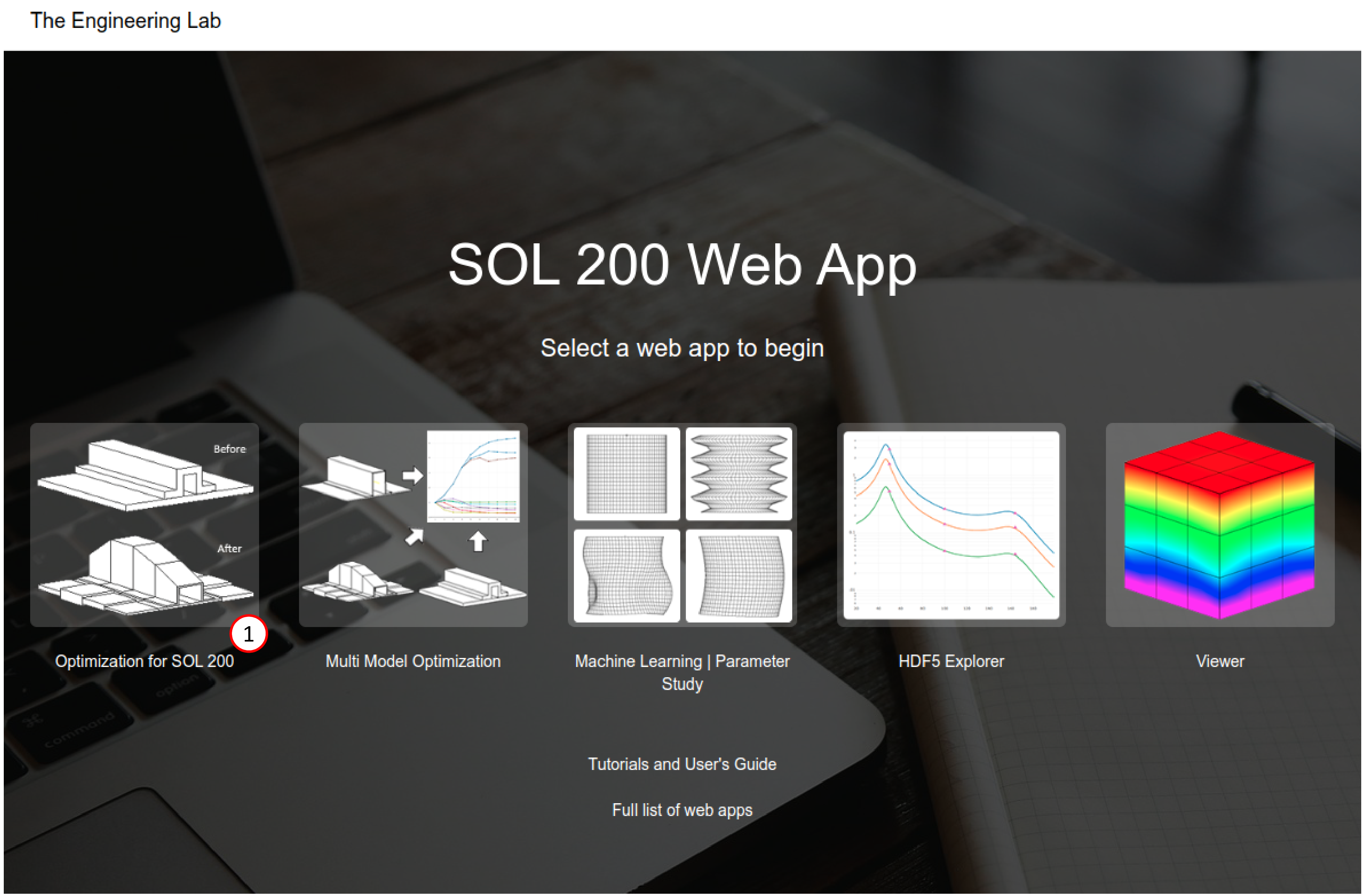
Starting BDF Files: Link 2
Solution BDF Files: Link



Open the Correct Page

1. Click on the indicated link

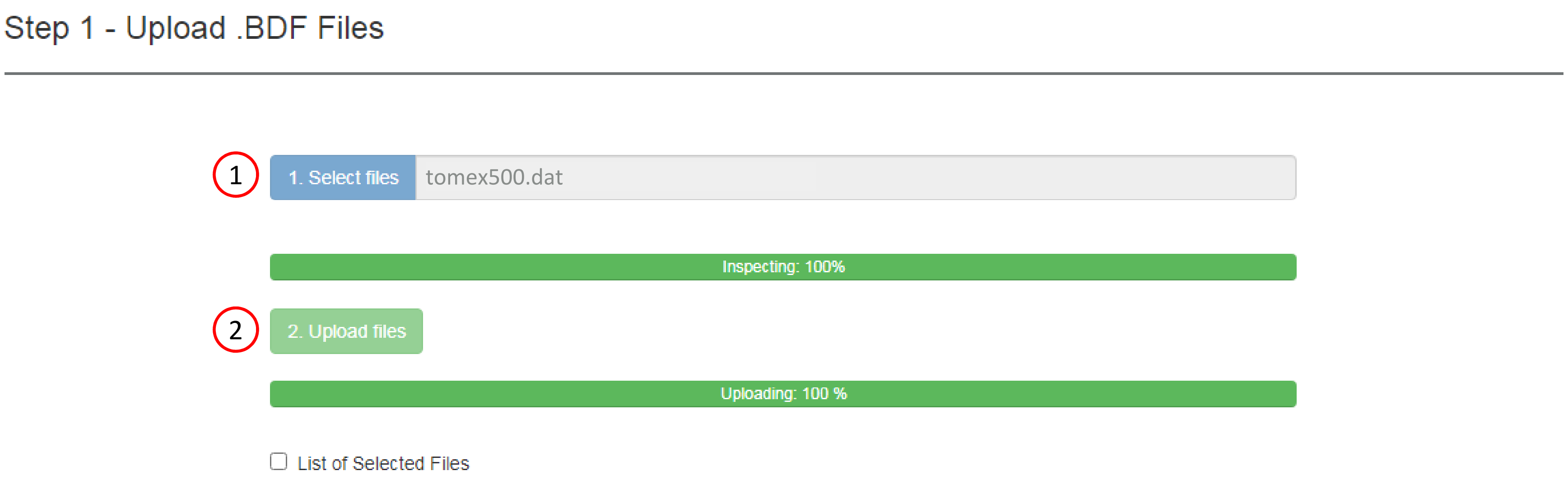
- MSC Nastran can perform many optimization types. The SOL 200 Web App includes dedicated web apps for the following:
    - Optimization for SOL 200 (Size, Topology, Topometry, Topography, Local Optimization, Sensitivity Analysis and Global Optimization)
    - Multi Model Optimization
    - Machine Learning
- The web app also features the HDF5 Explorer, a web application to extract results from the H5 file type.



Upload BDF Files

1. Click 1. Select Files and select tomex500.dat
2. Click Upload Files

- The process starts by uploading all the necessary BDF files. The BDF files can be files of your own or files found in the Tutorials section of the User's Guide.



Create Design Region

1. Click Topometry
2. In the search box, type ‘thickness’
3. Click on the plus (+) icons to set the thickness (Ti) of Plies 1-4 as Design Regions
4. Scroll to the section titled Step 2 – Adjust TOMVAR entries
5. The new Design Region is added to the table, no further edit is necessary

- Suppose the goal is to vary the thickness. In traditional Size optimization, the thickness can be a set a single design variable. With Topometry optimization, when the design region is set, each element in the region is given its own independent thickness design variable.
- If PSHELL 1 has 500 elements associated and is configured as a design region, then there will be 500 design variables created.
- Each step has hidden functionality for advanced users. The visibility is controlled by clicking +Options .
- If the property entry, e.g. PSHELL, was given a name in Patran, e.g. Car Door, the name can be shown by marking the checkbox titled Entry Name.

SOL 200 Web App - Optimization — Upload | Variables | Objective | Constraints | Subcases | Exporter | Results

Size | Topology | Topometry | Topography

Step 1 - Select design properties

+ Options

Create TOMVAR	Property	Property Description	Entry	Entry ID	Current Value
	Search	thickness	Search	Search	Search
+	T1	Thickness of ply	PCOMP	1	1.000
+	T2	Thickness of ply	PCOMP	1	1.000
+	T3	Thickness of ply	PCOMP	1	1.000
+	T4	Thickness of ply	PCOMP	1	1.000

5 | 10 | 20 | 30 | 40 | 50

Number of Visible Rows 5

Step 2 - Adjust TOMVAR Entries

+ Options

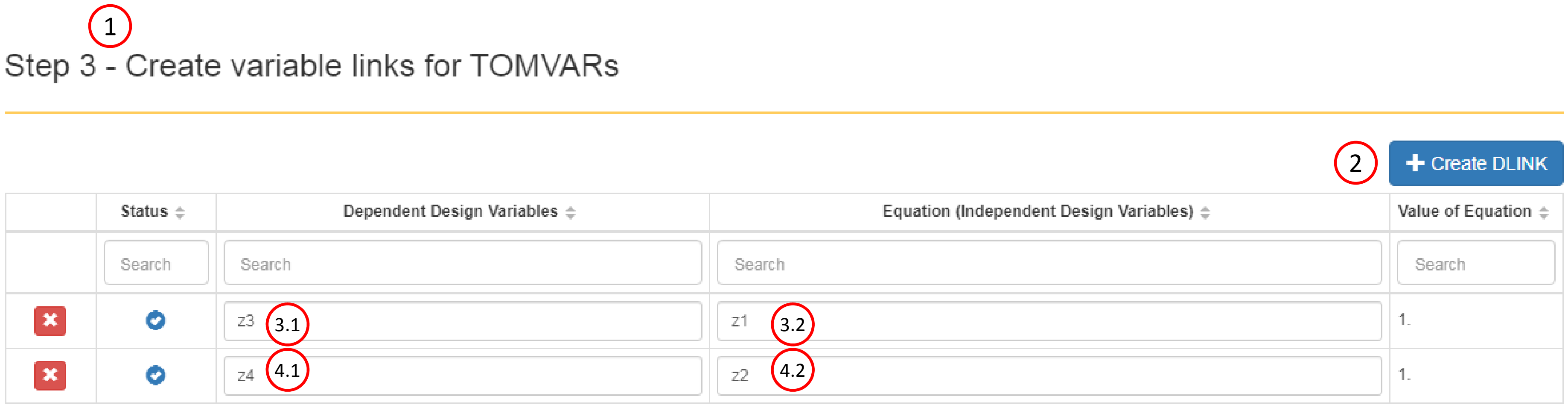
✖ Delete Visible Rows

	Label	Status	Property	Property Description	Entry	Entry ID	Initial Value	Lower Bound	Upper Bound	Allowed Discrete Values
	Search	Search	Search	Search	Search	Search	Search	Search	Search	Search
✖	z1	✔	T1	Thickness of ply	PCOMP	1	1.000	.001	Upper	Examples: -2.0, 1.0, THRU, 10.0, B'
✖	z2	✔	T2	Thickness of ply	PCOMP	1	1.000	.001	Upper	Examples: -2.0, 1.0, THRU, 10.0, B'
✖	z3	✔	T3	Thickness of ply	PCOMP	1	1.000	.001	Upper	Examples: -2.0, 1.0, THRU, 10.0, B'
✖	z4	✔	T4	Thickness of ply	PCOMP	1	1.000	.001	Upper	Examples: -2.0, 1.0, THRU, 10.0, B'

Create Design Region

1. Scroll to the section titled Step 3 – Optional – Create variable links for TOMVARs
2. Click 2 times on +Create DLINK
3. Create design variables links for the thickness of ply 1 and 3
    1. Dependent Design Variables: z3
    2. Equation: z1
4. Create design variables links for the thickness of ply 2 and 4
    1. Dependent Design Variables: z4
    2. Equation: z2

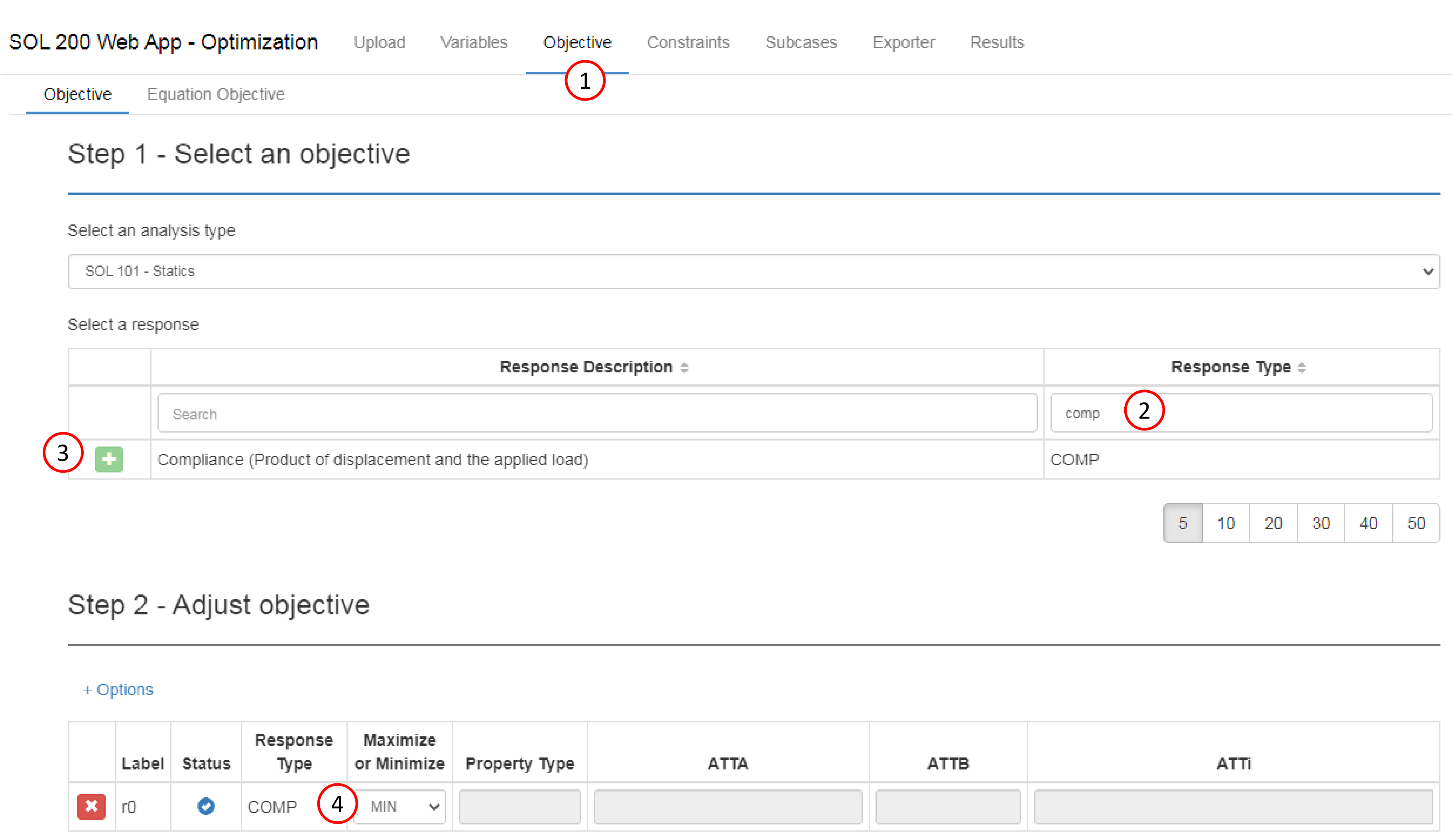
- It is important to verify the Equation is configured properly. For example, the variable z3 is initially equal to 1.0. When the Equation is configured, it should also produce an initial value of 1.0. The resulting value of the Equation is displayed on the column titled Value of Equation and can be used to validate the Equation is configured properly.



Create Design Objective

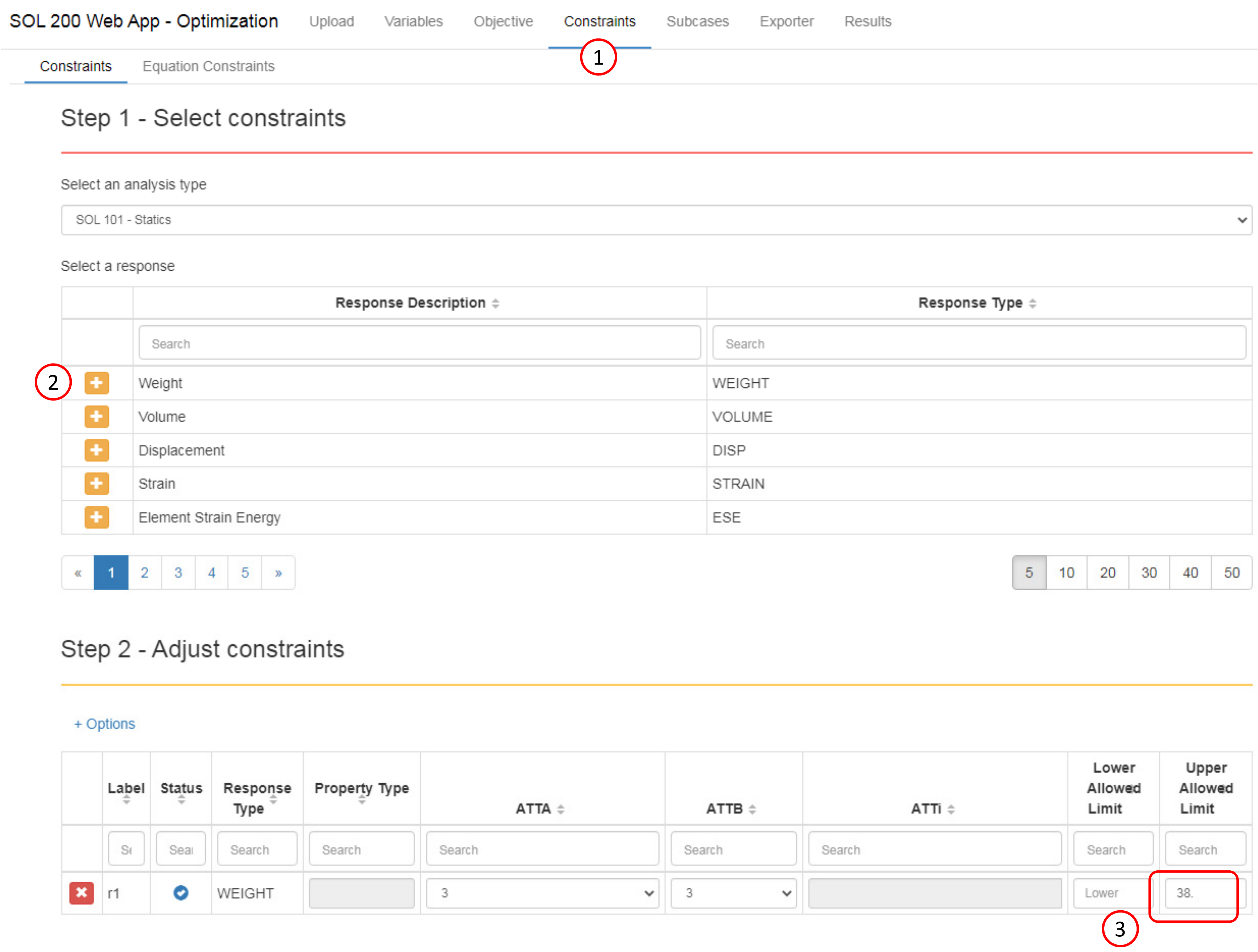
1. Click on Objective
2. Type ‘comp’ in the search box
3. Select the plus(+) icon for Compliance
4. The objective with label r0 is created. The objective is to minimize (MIN)

- The objective must always be a single scalar response. A response such as weight and volume are single responses and can be used as an objective. Other responses require special care when set as an objective. For example, if the objective is stress, only the stress of a single component, e.g. von Mises, of a single element, of a single load case may be used.



Create Design Constraints

1. Click Constraints
2. Click on the plus (+) icon for Weight
3. Configure the following for r1:
    - Upper Allowed Limit: 38.0



- The constraints are defined normally as would be done in a size optimization.

Configure Optimization Settings

1. Click Settings
2. Set P2 to 12 – Print constraints and responses

- The P2 setting controls the output of the following information to the F06 file: objective, constraints, responses, properties and design variables.
- This is a topometry optimization and will generate a large amount of property and design variable data in the F06 file. To make the F06 file size manageable, the design variable information is omitted by using the P2=12 option. When the results are viewed, note that the objective and constraint information is plotted, but the design variable history is not plotted due to the P2=12 option.
- If this is a combined size and topometry optimization, P2 should be set to 15. If this is a pure size optimization, P2 should be set to 15.

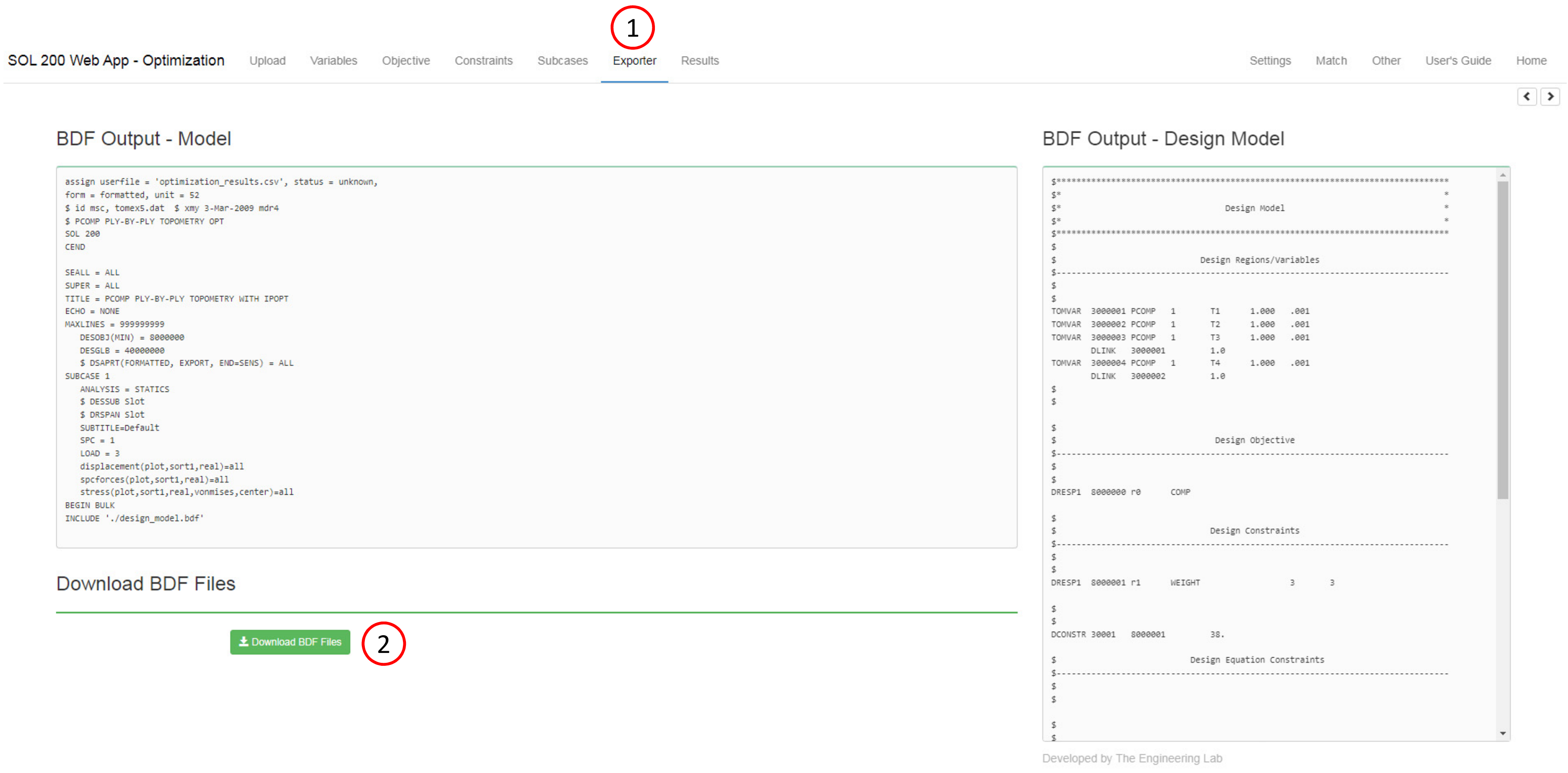
SOL 200 Web App - Optimization | Upload | Variables | Objective | Constraints | Subcases | Exporter | Results | Settings (1) | Match | Other

Optimization Settings

Parameter	Description	Configure
Search	Search	Search
APRCOD	Approximation method to be used	☐ 2 - Mixed Method
CONV1	Relative criterion to detect convergence	☐ Enter a positive real number
CONV2	Absolute criterion to detect convergence	☐ Enter a positive real number
DELX	Fractional change allowed in each design variable during any optimization cycle	☐ Enter a positive real number
DESMAX	Maximum number of design cycles to be performed	☑ 20
DISBEG	Design cycle number for discrete variable processing initiation	☐ Enter a positive integer
GMAX	Maximum constraint violation allowed at the converged optimum	☐ Enter a positive real number
P1	Print items, e.g. objective, design variables, at every n-th design cycle to the .f06 file	☑ 1
P2	Items to be printed to the .f06 file (2)	☑ 12 - Print constraints and respons
TCHECK	Topology Checkerboarding	☐ -1 - Automatic selection (Default)
TDMIN	Minimum diameter of members in topology optimization	☐ Enter a positive real number
TREGION	Trust Region	☐ 1 - Trust Region On

Export New BDF Files

1. Click on Exporter
2. Click on Download BDF Files

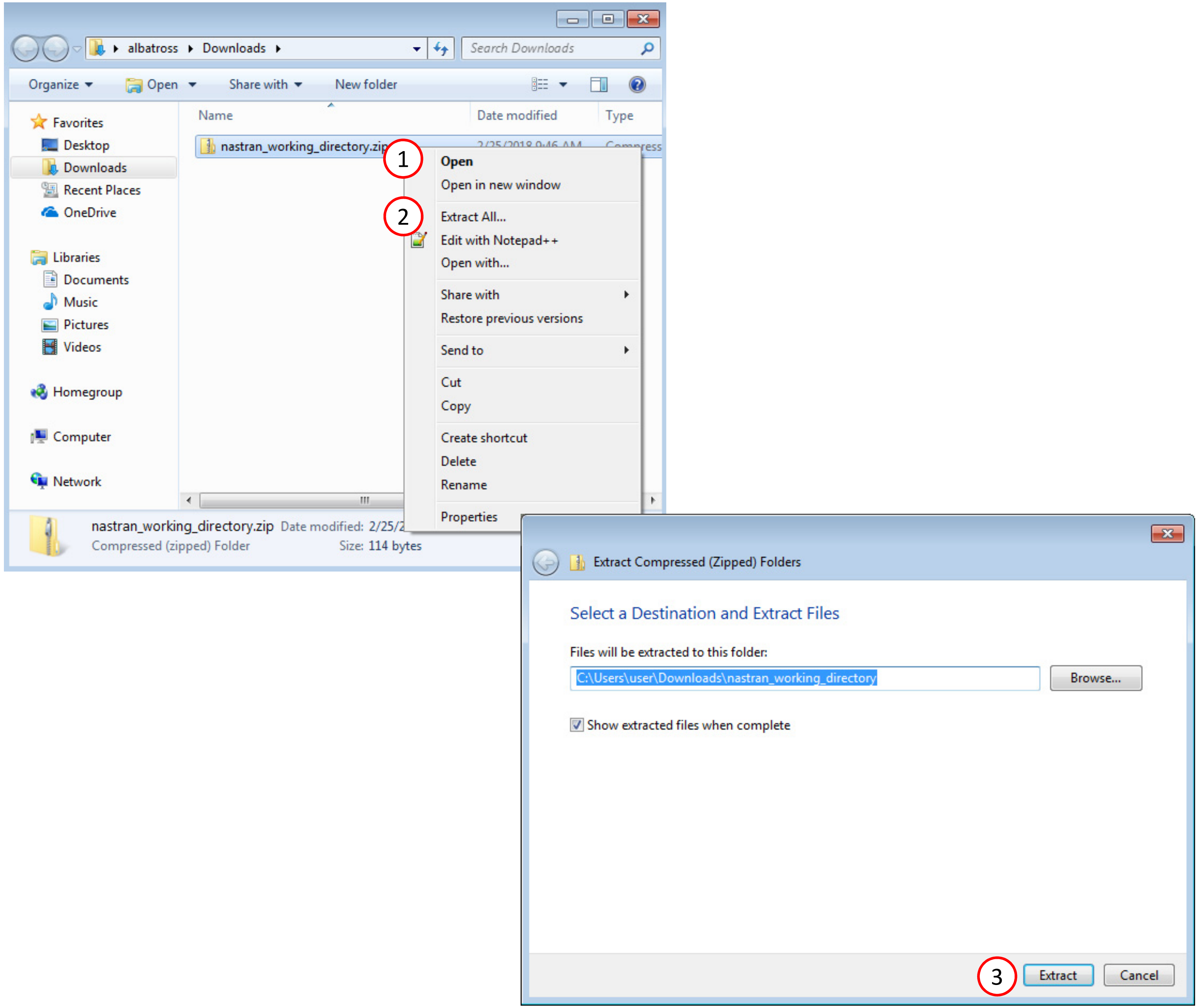


- When the download button is clicked a new file named “nastran_working_directory” is downloaded. If the file already exists in your local folder, the folder name is appended with a number, e.g. “nastran_working_directory (1).zip”

Perform the Optimization with Nastran SOL 200

1. A new .zip file has been downloaded
2. Right click on the file
3. Click Extract All
4. Click Extract on the following window

- Always extract the contents of the ZIP file to a new, empty folder.



Perform the Optimization with Nastran SOL 200

1. Inside of the new folder, double click on Start MSC Nastran
2. Click Open, Run or Allow Access on any subsequent windows
3. MSC Nastran will now start

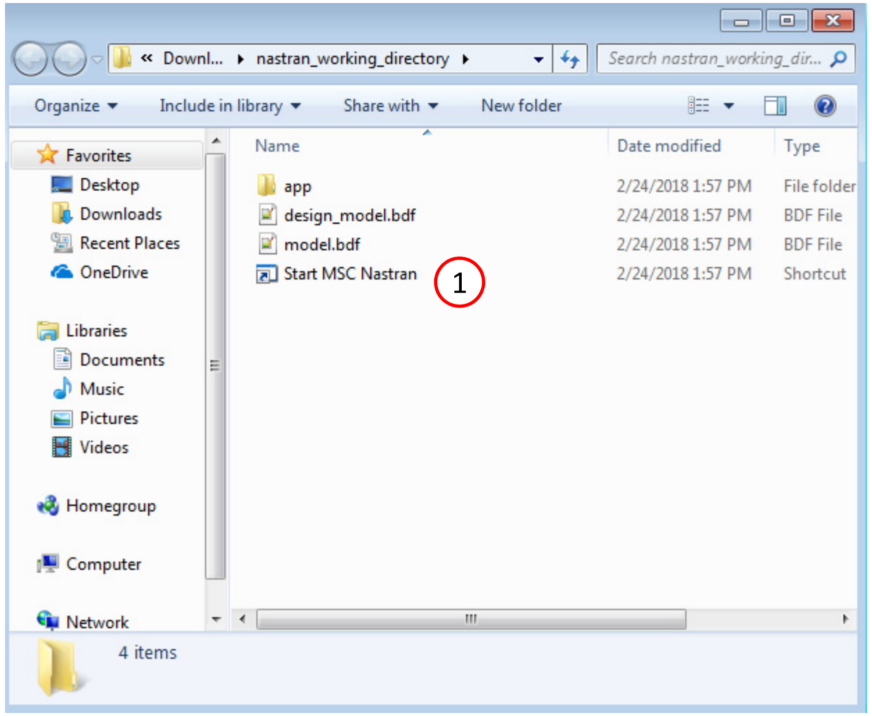
- After a successful optimization, the results will be automatically displayed as long as the following files are present: BDF, F06 and LOG.
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click “Start MSC Nastran” to display the results.

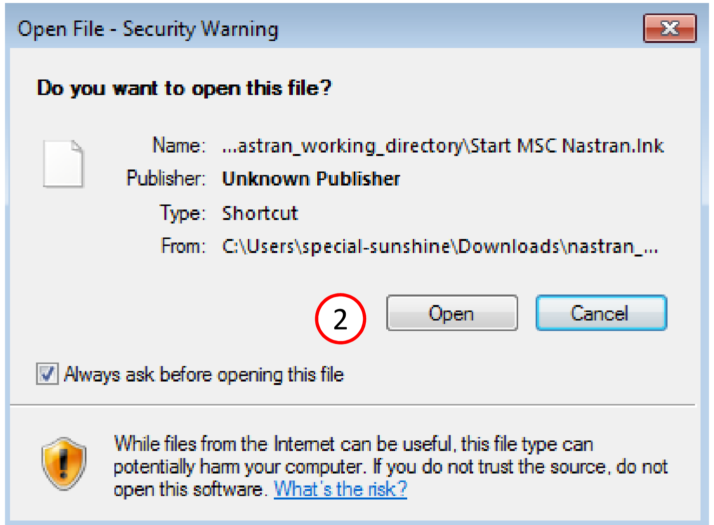
Using Linux?

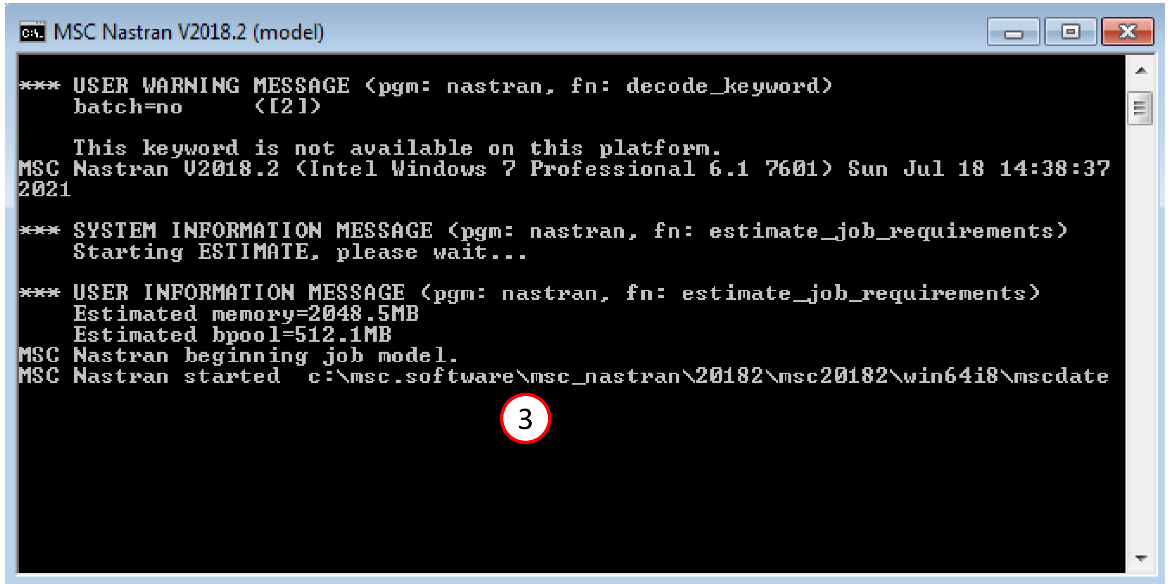
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

In some instances, execute permission must be granted to the directory. Use this command. This command assumes you are one folder level up.

sudo chmod -R u+x ./nastran_working_directory







Status

1. While MSC Nastran is running, a status page will show the current state of MSC Nastran

- The status of the MSC Nastran job is reported on the Status page. Note that Windows 7 users will experience a delay in the status updates. All other users of Windows 10 and Red Hat Linux will see immediate status updates.

SOL 200 Web App - Status

Python MSC Nastran

Status

Name	Status of Job	Design Cycle	RUN TERMINATED DUE TO
model.bdf	Running	None	

Review Optimization Results

After MSC Nastran is finished, the results will be automatically uploaded.

1. Ensure the messages shown have green checkmarks. This is indication of success. Any red icons indicate challenges.
2. The final value of objective, normalized constraints (not shown here) and design variables can be reviewed.

- Note that in a Topometry optimization, hundreds or thousands of design variables can be created. In this situation, the Design Variables are not plotted and displayed. Instead, the Objective and Normalized Constraints are displayed. It is recommended that a traditional post-processor be used to review the design variable results.

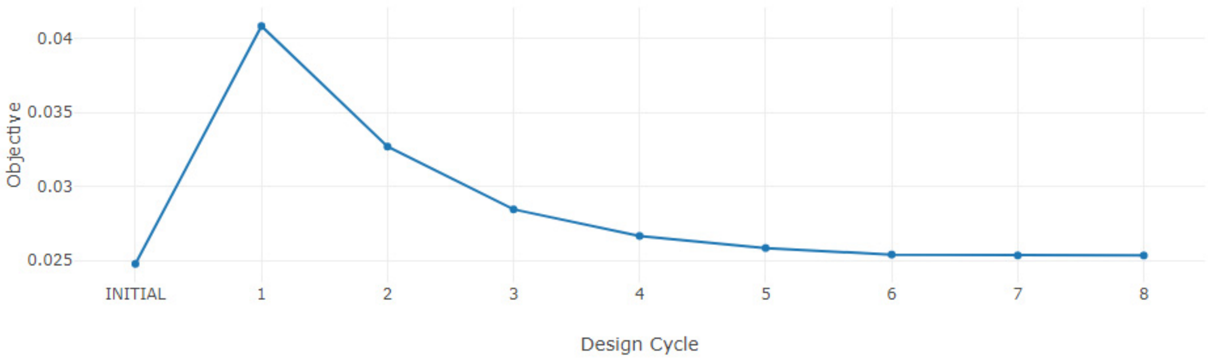
SOL 200 Web App - Local Optimization Results

Home

1 Final Message in .f06

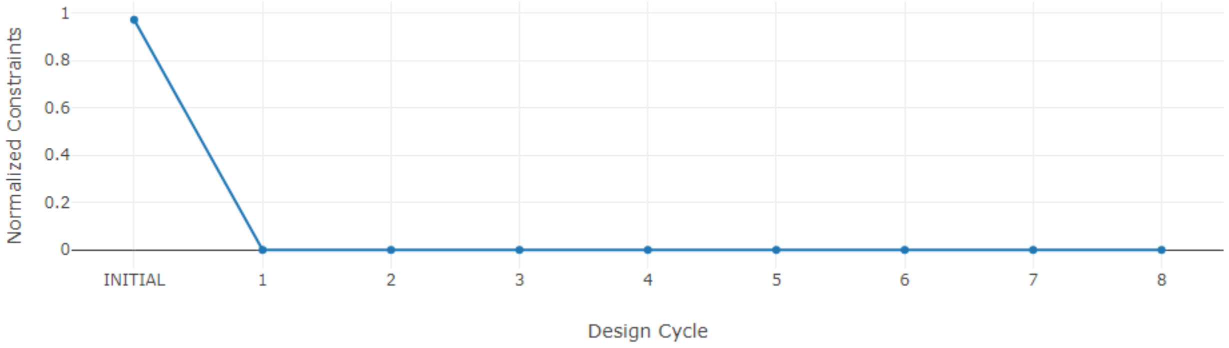
RUN TERMINATED DUE TO HARD CONVERGENCE TO AN OPTIMUM AT CYCLE NUMBER = 8.

Objective



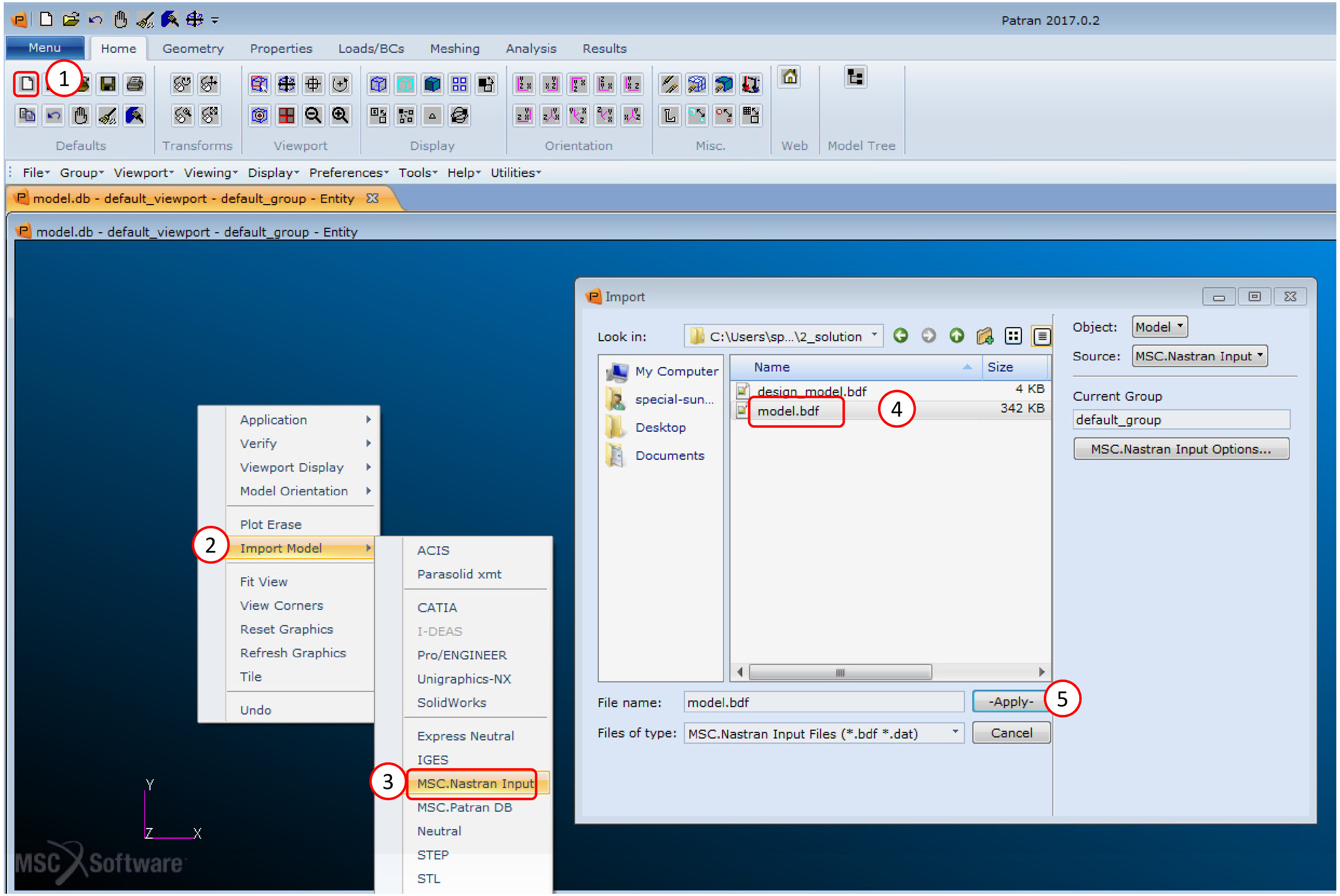
2 Normalized Constraints

+ Info



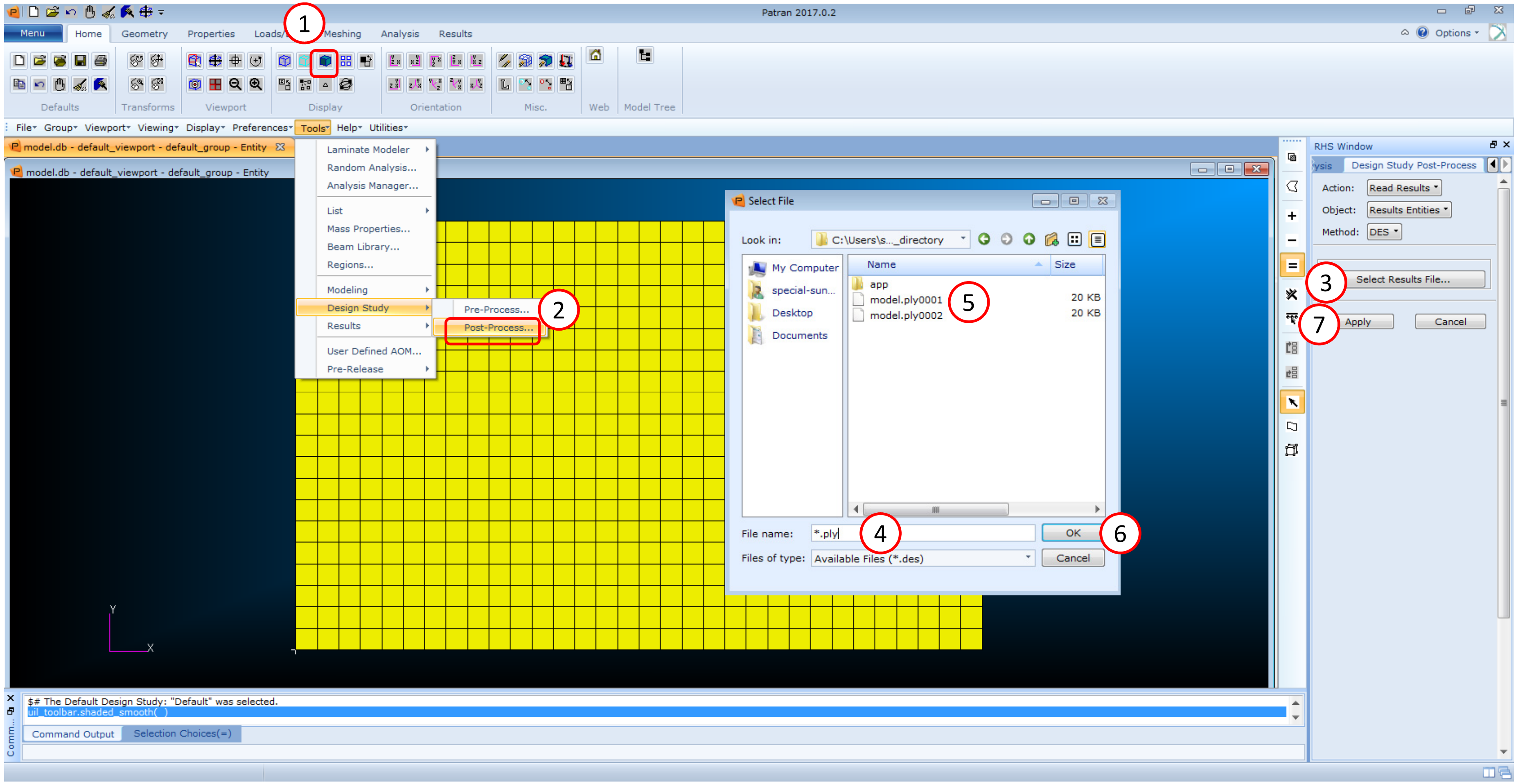
Review Optimization Results in Patran

1. Start a new Patran session
2. Right click to open a menu
3. Go to Import Model and click on MSC.Nastran Input
4. Select model.bdf (This file was used for the optimization)
5. Click Apply



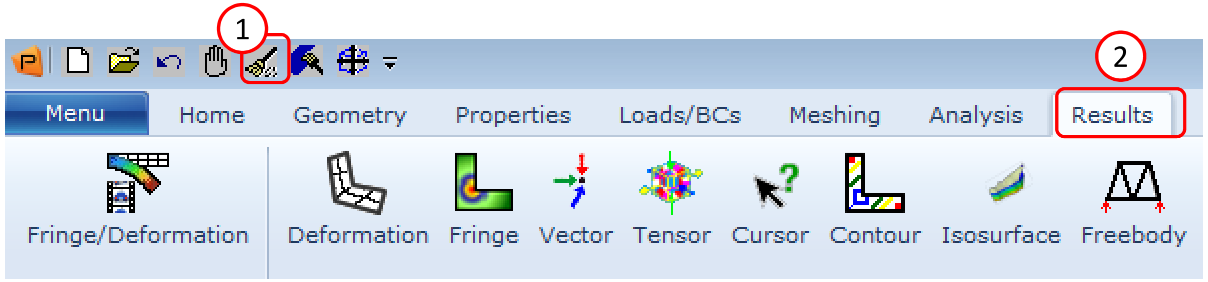
Review Optimization Results in Patran

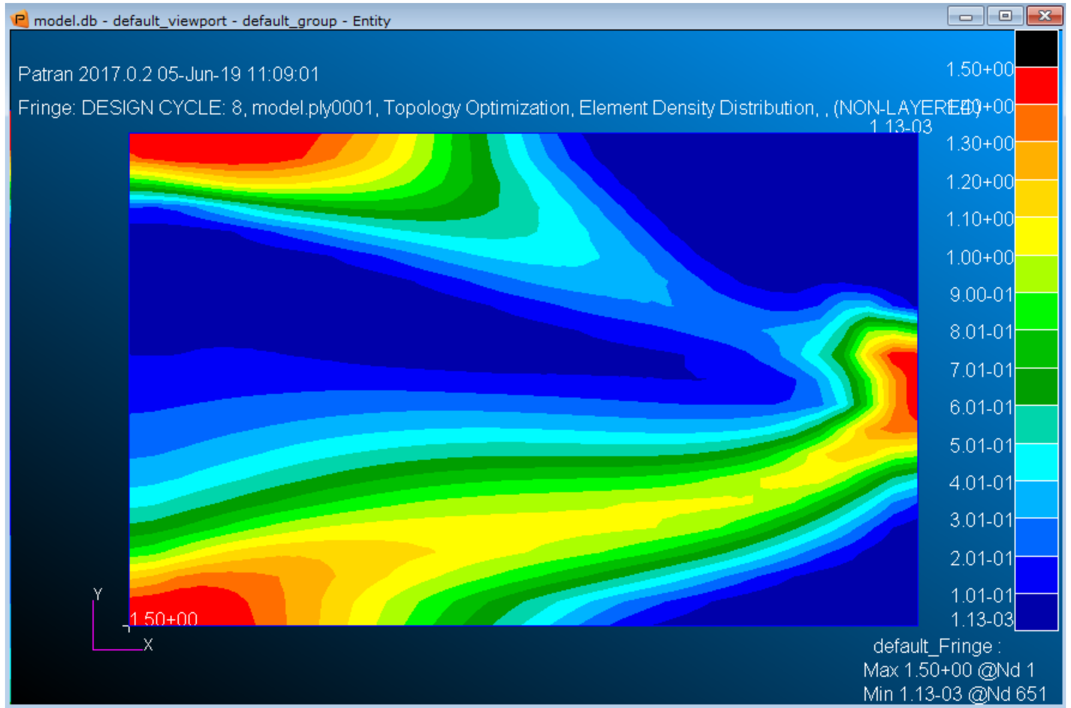
1. Click Smooth Shading
2. Go to Tools > Design Study and click on Post-Process
3. Click Select Results File
4. Type “*.ply” and press the enter key to show all the files in the folder
5. Select model.ply0001 (This file was created during the optimization)
6. Click OK
7. Click Apply
8. Repeat steps 3-7 for the following files:
    1. model.ply0002

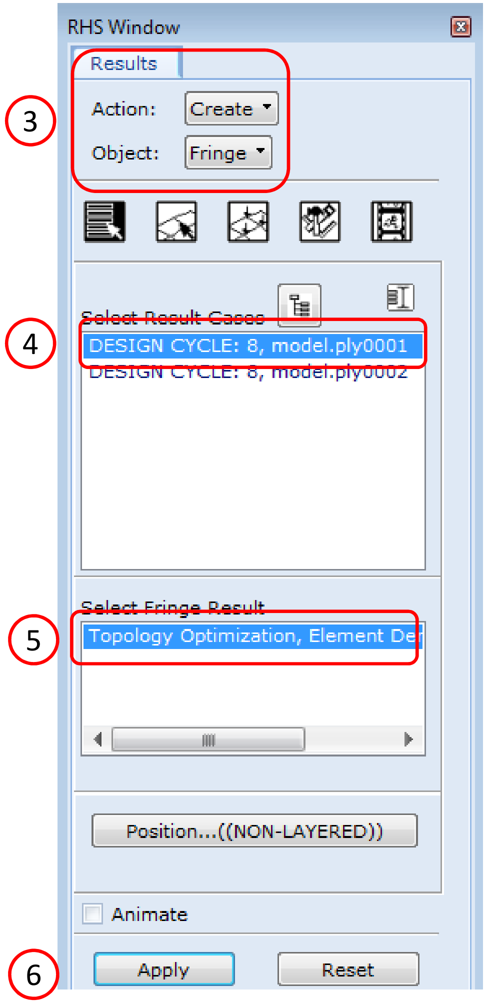


Review Optimization Results in Patran

1. Click the clear icon
2. Click Results
3. Set the following:
    - Action: Create
    - Object: Fringe
4. Select: DESIGN CYCLE: 8, model.ply0001 (The row should be highlighted blue)
5. Select Topology Optimization, Element D… (The row should be highlighted blue)
6. Click Apply
7. Repeat steps 4-6 for the following:
    - DESIGN CYCLE: 8, model.ply0002







Update the Original Model

Ensure the BDF files prior to optimization have one of these entries:

- H5 Output
  - `MDLPRM   HDF5     1`
  - `HDF5OUT  INPUT    YES`

  MDLPRM HDF5 is supported in MSC Nastran 2016.1 and newer. HDF5OUT is supported in MSC Nastran 2022.2 and newer.

The following applies to MSC Nastran 2023.4 and older. For MSC Nastran 2024.1, this is not needed.

Change DESPCH1 to 1

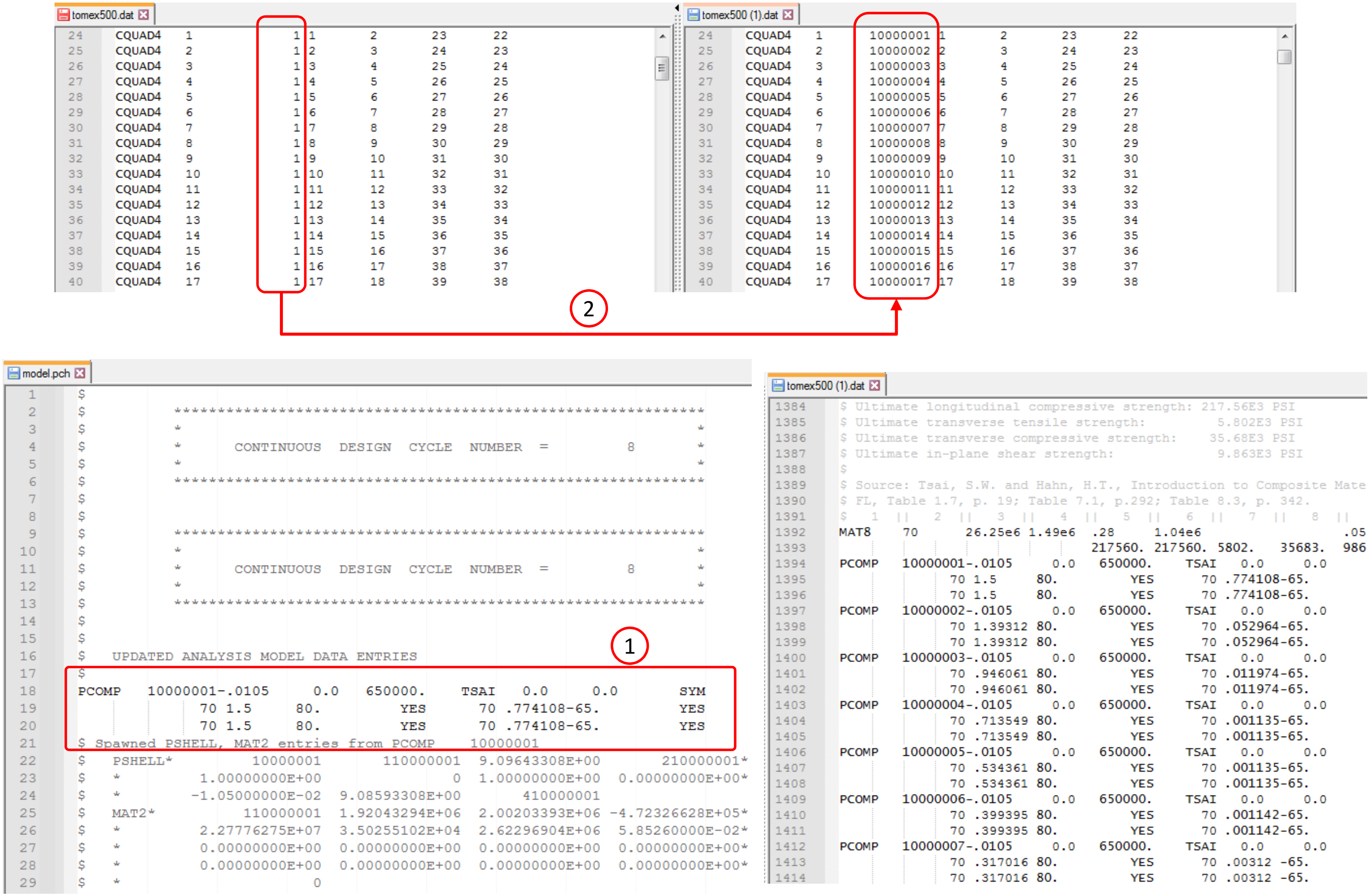
- DESPCH
  - Before:

    PARAM   DESPCH1 -1

  - After:

    PARAM   DESPCH1 1

  DESPCH1 -1 outputs entries to the PCH file in the small field format. Since the PSHELL IDs are longer than 8 characters, the IDs appear as asterisk characters, e.g. *******. DESPCH1 1 outputs the entries in the large field format, so the IDs are fully visible.



PCH

Downloaded BDF/DAT File

Update the Original Model

1. This Python script is used to automate the update the process.

```
import h5py
import hdf5plugin  # This library is necessary when HDF5OUT is used (Approximately MSC Nastran 2021 and newer)
import re


def get_dataset_cquad4(path_of_h5_file):
    file = h5py.File(path_of_h5_file, 'r')
    dataset1 = file['/NASTRAN/INPUT/ELEMENT/CQUAD4']
    dataset_original = dataset1[...]

    list_of_objects = []

    for element in dataset_original:
        # Store the following fields EID, PID, G1, G2, G3, G4
        list_of_objects.append(
            {
                'eid': element[0],
                'pid': element[1],
                'g1': element[2][0],
                'g2': element[2][1],
                'g3': element[2][2],
                'g4': element[2][3]
            }
        )

    return list_of_objects


def read_cquad4_entries_from_h5_and_write_to_bdf(path_a, path_of_new_bdf_file):
    objects_a = get_dataset_cquad4(path_a)
    list_of_strings = []

    for element_i in objects_a:
        # Write the fields to an array/list
        # Ensure all array elements are strings so ','.join() works properly
        array_of_fields = [
            'CQUAD4',
            str(element_i['eid']),
            str(element_i['pid']),
            str(element_i['g1']),
            str(element_i['g2']),
            str(element_i['g3']),
            str(element_i['g4'])
        ]

        # Create the entry with comma delimiters, which is the free field format
        list_of_strings.append(','.join(array_of_fields))

    # Write the strings to a text file
    file = open(path_of_new_bdf_file, 'w')

    for item in list_of_strings:
        file.write(item + '\n')

    file.close()


def filter_entries_from_pch(path_of_pch_file, name_of_entry, path_of_new_bdf_file):
    # This function reads a PCH file and keeps specific entries
    # Before (PCH File):
    # PCOMP   10000001-.0105     0.0   650000.    TSAI   0.0     0.0       SYM
    #               70 1.5     80.         YES      70 .774108-65.         YES
    #               70 1.5     80.         YES      70 .774108-65.         YES
    # $ Spawned PSHELL, MAT2 entries from PCOMP    10000001
    # $   PSHELL*          10000001       110000001  9.09643308E+00       210000001*
    # $   *         1.00000000E+00                0  1.00000000E+00  0.00000000E+00*
    # After (new_entries.bdf):
    # PCOMP   10000001-.0105     0.0   650000.    TSAI   0.0     0.0       SYM
    #               70 1.5     80.         YES      70 .774108-65.         YES
    #               70 1.5     80.         YES      70 .774108-65.         YES

    file = open(path_of_pch_file, 'r')
    file_b = open(path_of_new_bdf_file, 'w')
    keep_line = False
    keep_continuation_line = False

    # Example: Suppose you only want to read PCOMP entries
    # 1   PCOMP   10000001-.0105     0.0   650000.    TSAI   0.0     0.0       SYM
    # 2                 70 1.5     80.         YES      70 .774108-65.         YES
    # 3                 70 1.5     80.         YES      70 .774108-65.         YES
    # 4   MAT1    101
    # 5   PSHELL*          10000001       110000001  9.09643308E+00       210000001*
    # 6   *          1.00000000E+00                0  1.00000000E+00  0.00000000E+00*
    # 7   *         -1.05000000E-02  9.08593308E+00       410000001
    # 8   $MAT2*          110000001  1.92043294E+06  2.00203393E+06 -4.72326628E+05*
    # 9   $*          2.27776275E+07  3.50255102E+04  2.62296904E+06  5.85260000E-02*
    # 10  $*          0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00*
    # 11  $*          0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00*
    # 12  $*                       0
    # 13  PCOMP   10000002-.0105     0.0   650000.    TSAI   0.0     0.0       SYM
    # 14                70 1.5     80.         YES      70 .774108-65.         YES
    # 15                70 1.5     80.         YES      70 .774108-65.         YES
    for line in file:
        if re.match(r'^' + name_of_entry, line):
            # This detects lines 1 and 13 which is the first line of the entry
            keep_line = True
        elif re.match(r'^(\*|\s+)', line) is None:
            # This detects lines 4, 5, 6, 7, 8, 9, 10, 11, 12 which are other entries not to keep
            keep_line = False
            keep_continuation_line = False

        if keep_line is True:
            if re.match(r'^(\*|\s+)', line):
                # This detects lines 2, 3, 14, 15 which are continuation lines of the entry
                keep_continuation_line = True

        if re.match(r'\$', line) is None:
            # This detects all lines, except lines 8, 9, 10, 11, 12 which are commented with $
            if keep_line is True or keep_continuation_line is True:
                # Write the line to a new file
                file_b.write(line)

    file.close()
    file_b.close()


if __name__ == '__main__':

    # Comments
    # 1. This python script outputs updated PCOMP and CQUAD4 elements after an MSC Nastran topometry optimization.
    # 2. Modify path_a and path_b, then run this script
    # 3. This works as long MDLPRM,HDF5,1 is used, which triggers the output of the
    #    INPUT datasets to the H5 file. The INPUT datasets are the bulk data entries: GRIDs, CQUAD4s, PSHELLs,
etc.

    path_a = '/home/usera/Downloads/nastran_working_directory/model.h5'
    path_b = '/home/usera/Downloads/nastran_working_directory/model.pch'

    # Output New QUAD4 Elements After Topometry Optimization
    # ##############################################################################
    # Output updated CQUAD4 entries
    read_cquad4_entries_from_h5_and_write_to_bdf(path_a, 'new_2D_elements.tmp')

    # Output
    # CQUAD4*,1,1000000001,1,2,16,15
    # CQUAD4*,2,1000000002,2,3,17,16
    # CQUAD4*,3,1000000003,3,4,18,17
    # CQUAD4*,4,1000000004,4,5,19,18
    # CQUAD4*,5,1000000005,5,6,20,19
    # CQUAD4*,6,1000000006,6,7,21,20
    # CQUAD4*,7,1000000007,7,8,22,21
    # [...]

    # Output New PCOMP Entries After Topometry Optimization
    # ##############################################################################
    filter_entries_from_pch(path_b, 'PCOMP', 'new_pcomp_entries.tmp')
    # Outputs
    # PCOMP   10000001-.0105     0.0   650000.    TSAI   0.0     0.0       SYM
    #               70 1.5     80.         YES      70 .774108-65.         YES
    #               70 1.5     80.         YES      70 .774108-65.         YES
    # PCOMP   10000002-.0105     0.0   650000.    TSAI   0.0     0.0       SYM
    #               70 1.39312 80.         YES      70 .052964-65.         YES
    #               70 1.39312 80.         YES      70 .052964-65.         YES
```

Update the Original Model

The Python script generates a new TMP file.

1. Copy and paste the CQUAD4 elements to the original BDF file.

new_2D_elements.tmp

new_2D_elements.tmp

```
CQUAD4,1,10000001,1,2,23,22
CQUAD4,2,10000002,2,3,24,23
CQUAD4,3,10000003,3,4,25,24
CQUAD4,4,10000004,4,5,26,25
CQUAD4,5,10000005,5,6,27,26
CQUAD4,6,10000006,6,7,28,27
CQUAD4,7,10000007,7,8,29,28
CQUAD4,8,10000008,8,9,30,29
CQUAD4,9,10000009,9,10,31,30
CQUAD4,10,10000010,10,11,32,31
CQUAD4,11,10000011,11,12,33,32
CQUAD4,12,10000012,12,13,34,33
CQUAD4,13,10000013,13,14,35,34
CQUAD4,14,10000014,14,15,36,35
CQUAD4,15,10000015,15,16,37,36
CQUAD4,16,10000016,16,17,38,37
CQUAD4,17,10000017,17,18,39,38
CQUAD4,18,10000018,18,19,40,39
CQUAD4,19,10000019,19,20,41,40
CQUAD4,20,10000020,20,21,42,41
CQUAD4,21,10000021,22,23,44,43
CQUAD4,22,10000022,23,24,45,44
CQUAD4,23,10000023,24,25,46,45
CQUAD4,24,10000024,25,26,47,46
CQUAD4,25,10000025,26,27,48,47
CQUAD4,26,10000026,27,28,49,48
CQUAD4,27,10000027,28,29,50,49
CQUAD4,28,10000028,29,30,51,50
CQUAD4,29,10000029,30,31,52,51
CQUAD4,30,10000030,31,32,53,52
CQUAD4,31,10000031,32,33,54,53
CQUAD4,32,10000032,33,34,55,54
CQUAD4,33,10000033,34,35,56,55
CQUAD4,34,10000034,35,36,57,56
```

Original BDF File

tomex500.dat

```
CQUAD4  1              1 1       2       23      22
CQUAD4  2              1 2       3       24      23
CQUAD4  3              1 3       4       25      24
CQUAD4  4              1 4       5       26      25
CQUAD4  5              1 5       6       27      26
CQUAD4  6              1 6       7       28      27
CQUAD4  7              1 7       8       29      28
CQUAD4  8              1 8       9       30      29
CQUAD4  9              1 9       10      31      30
CQUAD4  10             1 10      11      32      31
CQUAD4  11             1 11      12      33      32
CQUAD4  12             1 12      13      34      33
CQUAD4  13             1 13      14      35      34
CQUAD4  14             1 14      15      36      35
CQUAD4  15             1 15      16      37      36
CQUAD4  16             1 16      17      38      37
CQUAD4  17             1 17      18      39      38
CQUAD4  18             1 18      19      40      39
CQUAD4  19             1 19      20      41      40
CQUAD4  20             1 20      21      42      41
CQUAD4  21             1 22      23      44      43
CQUAD4  22             1 23      24      45      44
CQUAD4  23             1 24      25      46      45
CQUAD4  24             1 25      26      47      46
CQUAD4  25             1 26      27      48      47
CQUAD4  26             1 27      28      49      48
CQUAD4  27             1 28      29      50      49
CQUAD4  28             1 29      30      51      50
CQUAD4  29             1 30      31      52      51
CQUAD4  30             1 31      32      53      52
CQUAD4  31             1 32      33      54      53
CQUAD4  32             1 33      34      55      54
CQUAD4  33             1 34      35      56      55
CQUAD4  34             1 35      36      57      56
```

1

Update the Original Model

The Python script generates a new TMP file.

1. Copy and paste the PCOMP entries to the original BDF file.

new_pcomp_entries.tmp

new_pcomp_entries.tmp

```
PCOMP   10000001-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 1.5      80.          YES        70 .774108-65.                YES
              70 1.5      80.          YES        70 .774108-65.                YES
PCOMP   10000002-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 1.39312 80.           YES        70 .052964-65.                YES
              70 1.39312 80.           YES        70 .052964-65.                YES
PCOMP   10000003-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .946061 80.           YES        70 .011974-65.                YES
              70 .946061 80.           YES        70 .011974-65.                YES
PCOMP   10000004-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .713549 80.           YES        70 .001135-65.                YES
              70 .713549 80.           YES        70 .001135-65.                YES
PCOMP   10000005-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .534361 80.           YES        70 .001135-65.                YES
              70 .534361 80.           YES        70 .001135-65.                YES
PCOMP   10000006-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .399395 80.           YES        70 .001142-65.                YES
              70 .399395 80.           YES        70 .001142-65.                YES
PCOMP   10000007-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .317016 80.           YES        70 .00312 -65.                YES
              70 .317016 80.           YES        70 .00312 -65.                YES
PCOMP   10000008-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .25365  80.           YES        70 .015178-65.                YES
              70 .25365  80.           YES        70 .015178-65.                YES
PCOMP   10000009-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .205456 80.           YES        70 .020367-65.                YES
              70 .205456 80.           YES        70 .020367-65.                YES
PCOMP   10000010-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .161272 80.           YES        70 .021622-65.                YES
              70 .161272 80.           YES        70 .021622-65.                YES
PCOMP   10000011-.0105      0.0     650000.     TSAI    0.0     0.0             SYM
              70 .116866 80.           YES        70 .055676-65.                YES
```

Original BDF File

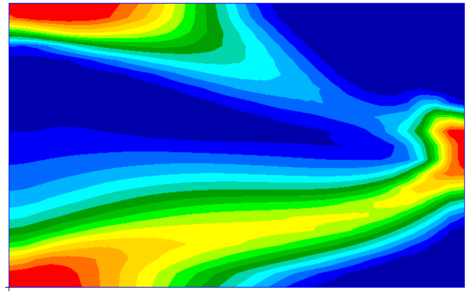
tomex500.dat

```
SPC1    1       123456  1       THRU    21
LOAD    3       1.      1.      1
$ Nodal Forces of Load Set : force
FORCE   1       683     0       300.    1.      -1.     0.
$ Referenced Coordinate Frames
mat1           1  2.07+5                .3  7.93
PCOMP   1       -.0105  0.0     0.65E6  TSAI                            SYM
        70      1.000   80.     YES     70      1.000   -65.    YES
        70      1.000   80.     YES     70      1.000   -65.    YES
$   1  ||   2  ||   3  ||   4  ||   5  ||   6  ||   7  ||   8  ||   9  ||
$ Composite Type: T300/5208 (Graphite/Expoxy)
$
$ E1: 26.25E6 PSI
$ E2:  1.49E6 PSI
$ v12: .28
$ G12: 1.04E6 PSI
$
$ Density: 5.8526E-2 lbm/in^3
$
$ Ultimate longitudinal tensile strength:     217.56E3 PSI
$ Ultimate longitudinal compressive strength: 217.56E3 PSI
$ Ultimate transverse tensile strength:         5.802E3 PSI
$ Ultimate transverse compressive strength:    35.68E3 PSI
$ Ultimate in-plane shear strength:             9.863E3 PSI
$
$ Source: Tsai, S.W. and Hahn, H.T., Introduction to Composite Materials,
$ FL, Table 1.7, p. 19; Table 7.1, p.292; Table 8.3, p. 342.
$   1  ||   2  ||   3  ||   4  ||   5  ||   6  ||   7  ||   8  ||   9  ||
MAT8    70      26.25e6 1.49e6  .28     1.04e6                  .058526
                                217560. 217560. 5802.   35683.  9863.
ENDDATA 5c8b7a30
```

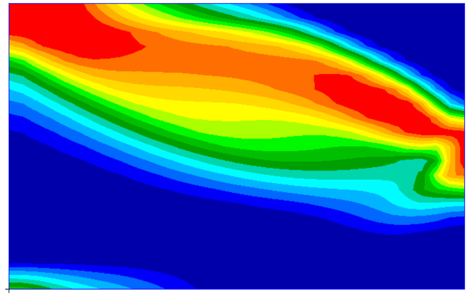
1

Results

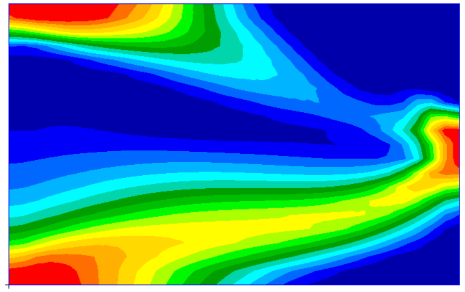
Topometry Results



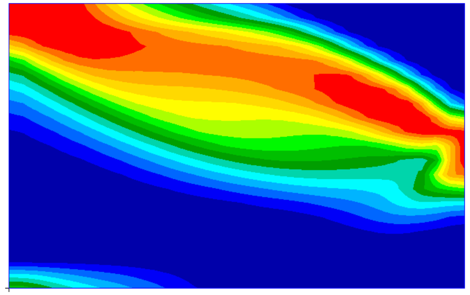
Ply 1 and 8
80°



Ply 2 and 7
-65°

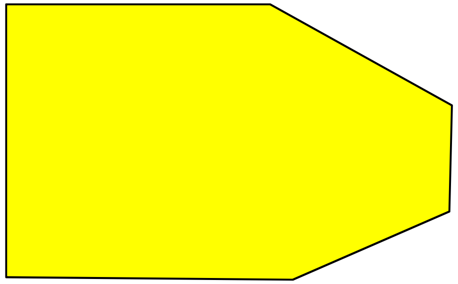


Ply 3 and 6
80°

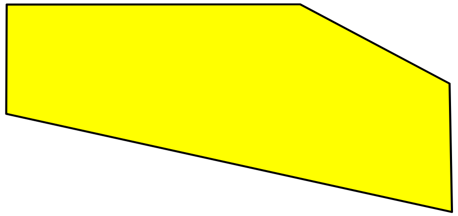


Ply 4 and 5
-65°

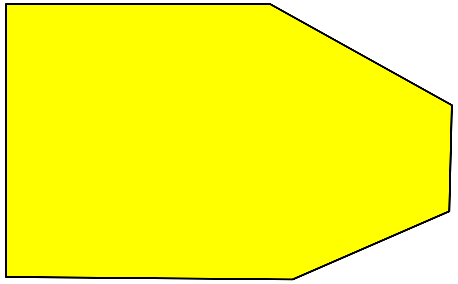
Possible Ply Shapes



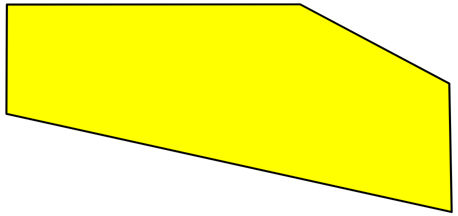
Ply 1 and 8
80°



Ply 2 and 7
-65°



Ply 3 and 6
80°

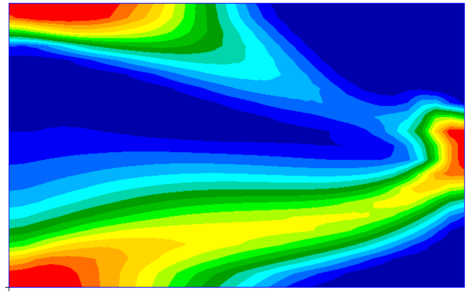


Ply 4 and 5
-65°

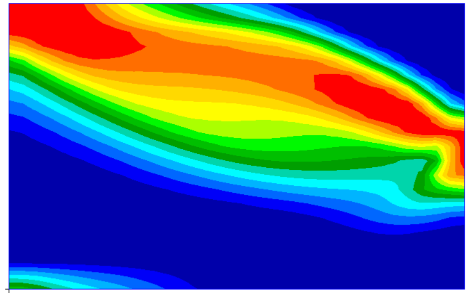
Results

Ply drop offs are employed for the -65 degree plies. In the interest of manufacturability, ply drop offs are not employed for the 80 degree plies.

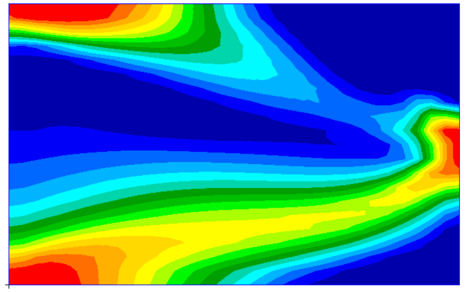
Topometry Results



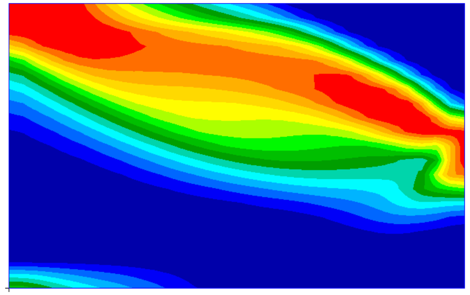
Ply 1 and 8
80°



Ply 2 and 7
-65°

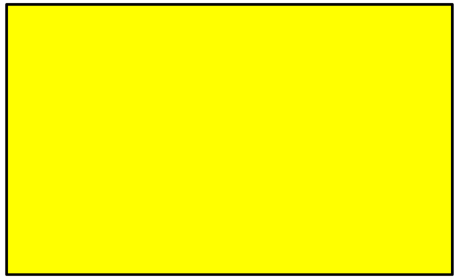


Ply 3 and 6
80°

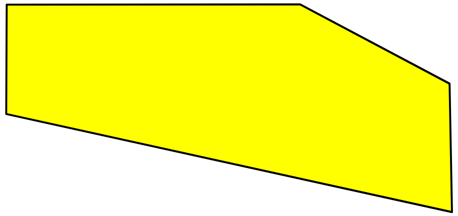


Ply 4 and 5
-65°

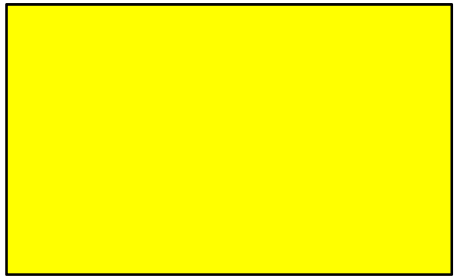
Proposed Ply Shapes



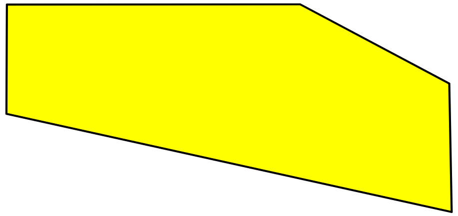
Ply 1 and 8
80°



Ply 2 and 7
-65°

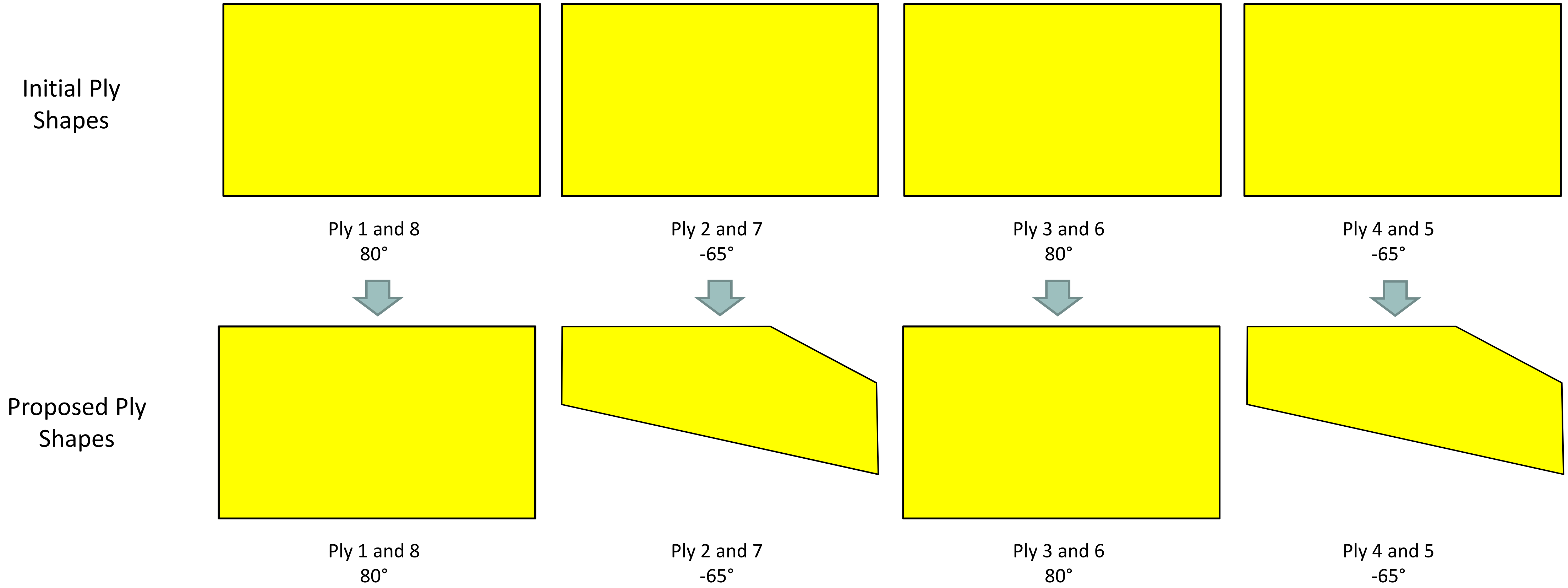


Ply 3 and 6
80°



Ply 4 and 5
-65°

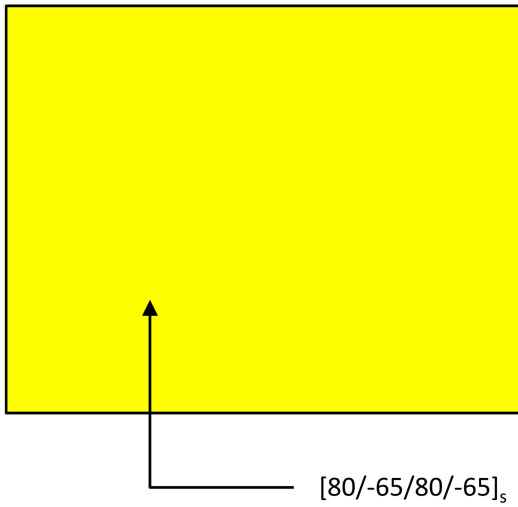
Results


Initial Ply Shapes
Ply 1 and 8
80°
Ply 2 and 7
-65°
Ply 3 and 6
80°
Ply 4 and 5
-65°
Proposed Ply Shapes
Ply 1 and 8
80°
Ply 2 and 7
-65°
Ply 3 and 6
80°
Ply 4 and 5
-65°

Results

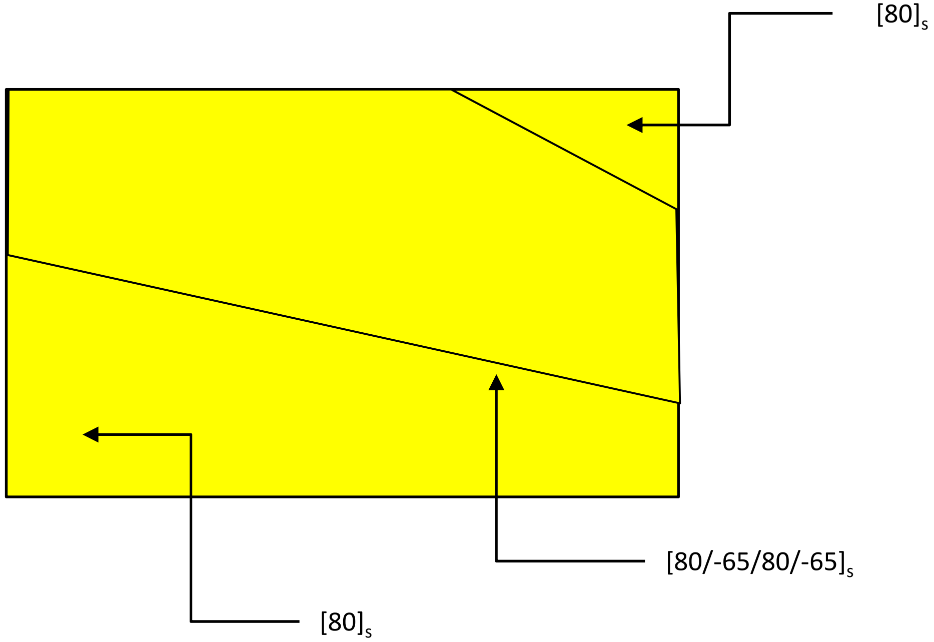
Before Optimization

- Panel of initial layup



After Optimization

- Panel of updated layups
- The weight is reduced



End of Tutorial