Workshop – Composite Panel – Phase C – Topometry Optimization to Determine Optimal Core Shape

AN MSC NASTRAN SOL 200 TUTORIAL

Composite Workshop

This workshop is phase C of a 3-phase workshop.

Phase B

Workshop – Composite Panel – Phase B – Baseline Core Thickness Optimization

- Perform a core thickness optimization with a constant thickness core
- Tools Used: SOL 200 Web App (Viewer and Optimization web apps) and MSC Nastran

Phase C

Workshop – Composite Panel – Phase C – Topometry Optimization to Determine Optimal Core Shape

- Generate PLY000i Files via Topometry Optimization
- Tools Used: Patran, MSC Nastran and SOL 200 Web App

Phase D

Workshop – Composite Panel – Phase D – Core Shape and Core Thickness Optimization

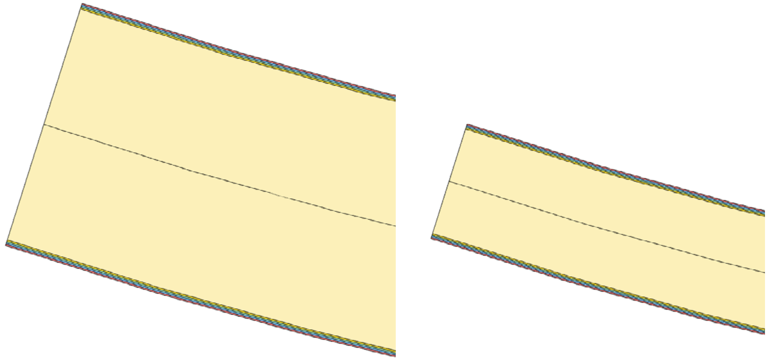
- Input BDF and PLY000i Files
- Create Core Shapes
- Perform Core Thickness Optimization
- Inspect Core
- Tools Used: SOL 200 Web App (Viewer and Optimization web apps) and MSC Nastran

Baseline Core Thickness Optimization → Core Shape Optimization → Core Thickness Optimization

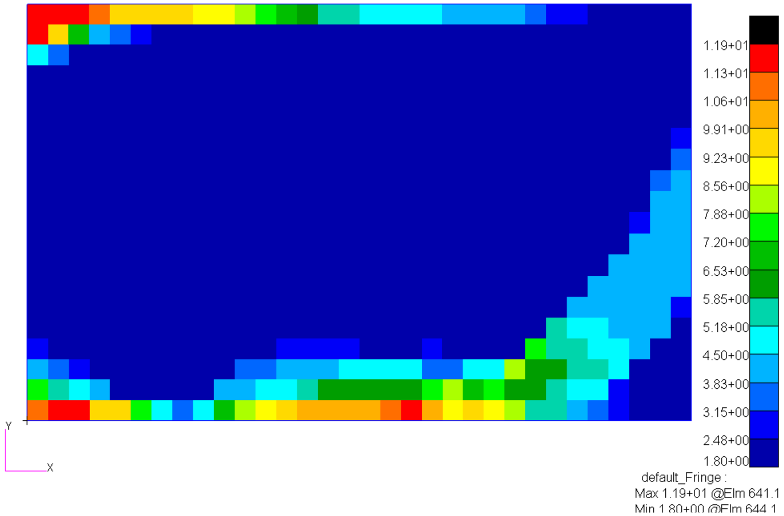
Composite Workshop

This workshop is phase C of a 3-phase workshop.

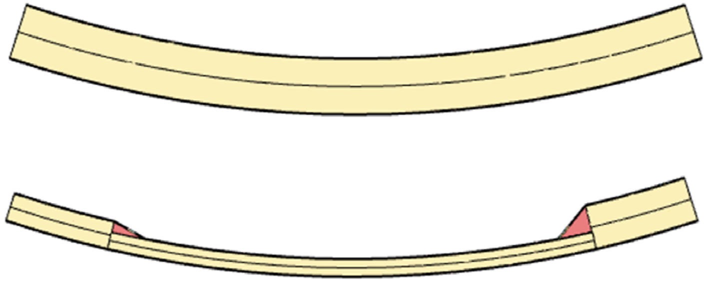
Phase B



Phase C



Phase D



Baseline Core Thickness Optimization

Core Shape Optimization

Core Thickness Optimization

Summary of Optimized Designs

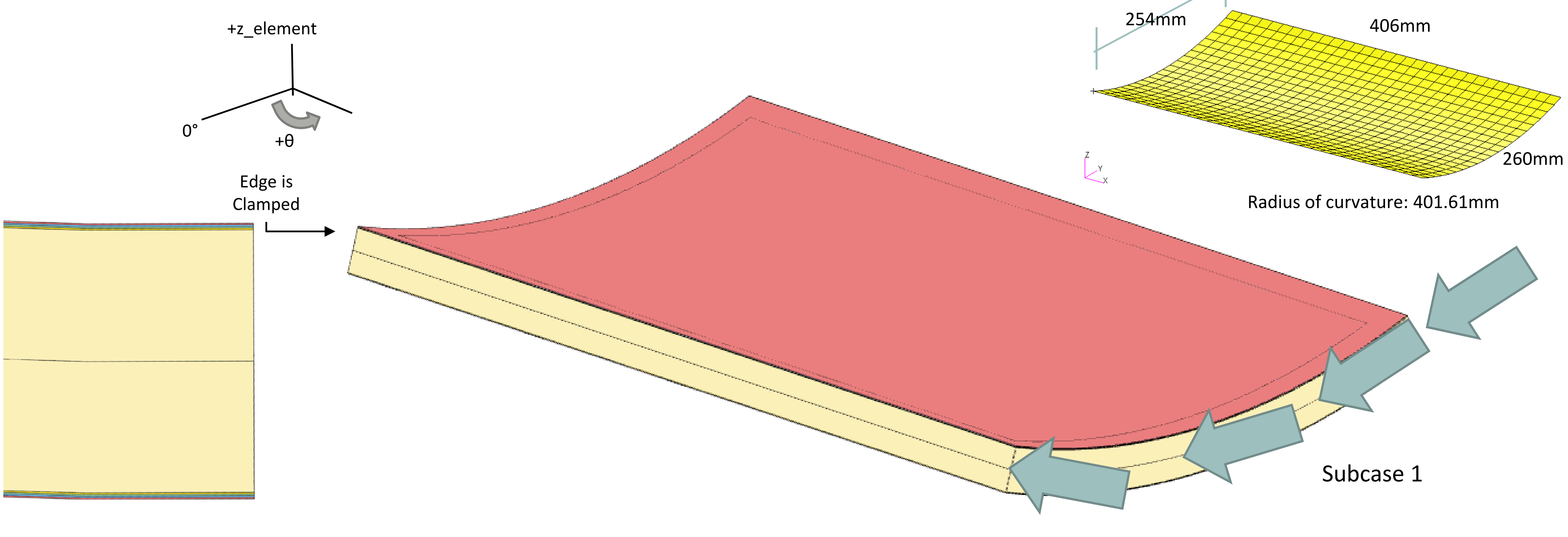
This tutorial is part of a 3-phase tutorial. Phase C and D yield optimized composites. Often, the results of a topometry optimization, produced in phase C, are difficult to manufacture, but the results are refined in phase D.

A comparison is made between the starting and final composite designs from phase B and D. Observe the following:

1. ~23% mass savings. The mass of the core was reduced from 2.203330E-04 to 1.70E-04.
2. In both designs, the buckling load factor is greater than 1.0, so both designs are feasible.

	Starting Design	**Design After Topometry Optimization**	**Design After Core Shape and Core Number Optimization**
	Tutorial Phase B	Tutorial Phase C	Tutorial Phase D
Total Mass	3.9503E-04	2.97E-4	3.444094E-04
Mass of Non-design Region (Plies)	1.746926E-04	1.746926E-04	1.746926E-04
Mass of Design Region (Core)	**2.203330E-04**	**1.22E-04**	**1.70E-04**
Buckling Load Factor, Subcase 2	1.064771 (OK)	9.9758E-01 (NOT OK)	1.013359 (OK)

Details of the structural model



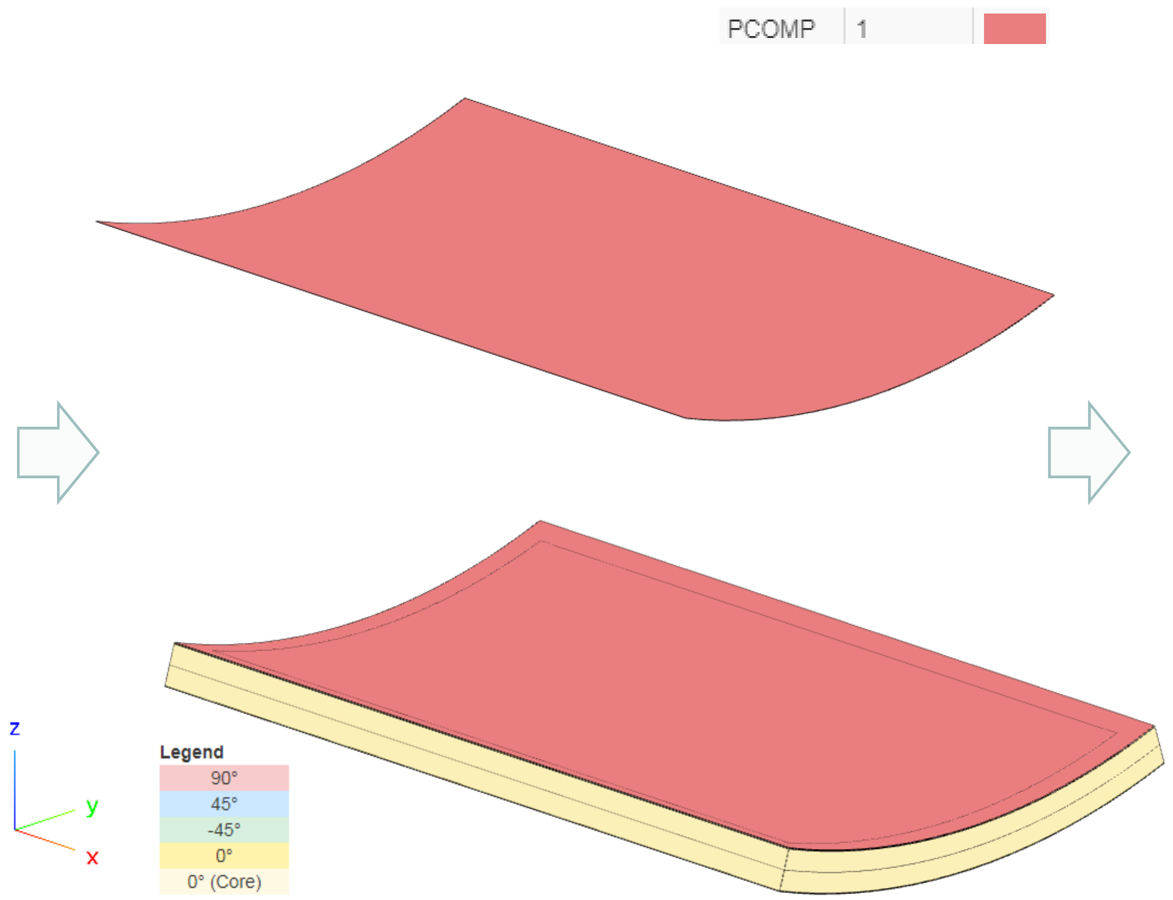
[90/+45/-45/0/ $0_{core}]_s$

Optimization Problem Statement

Design Region (Topometry)

z1: Thickness of layer 5 and 6

$.001 < z1 < 25.0$



Design Objective

Minimize r0: weight

Design Constraints

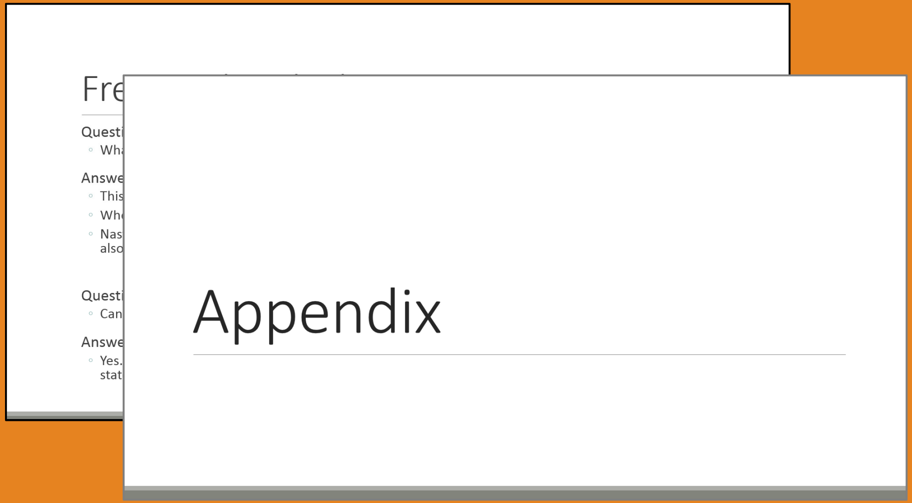
r1: 1st buckling load factor

$1.0 < r1$

More Information Available in the Appendix

The Appendix includes information regarding the following:

- Why Use the Trust Region?
- Using the Topometry Optimization Results as Is



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

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 Variables
     - Design Objective
     - Design Constraint
3. Use Topometry Optimization to create PLY000i files

Special Topics Covered

PLY000i Files - The PLY000i files contain information that is used to construct ply shapes. This tutorial discusses how to manually create PLY000i files.

SOL 200 Web App Capabilities

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

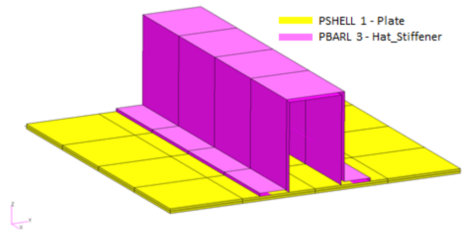
Compatibility

- 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.

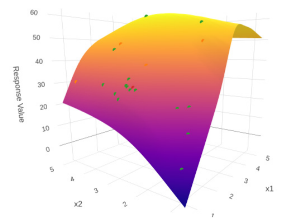
Benefits

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

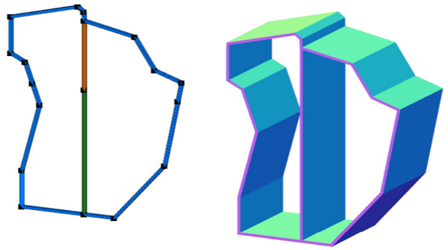
Web Apps



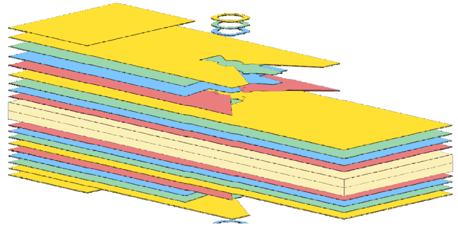
Web Apps for MSC Nastran SOL 200
Pre/post for MSC Nastran SOL 200. Support for size, topology, topometry, topography, multi-model optimization.



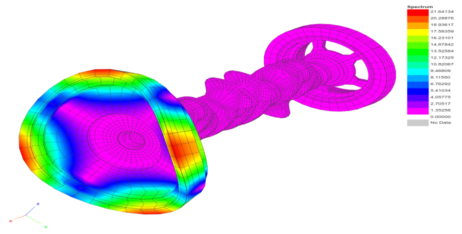
Machine Learning Web App
Bayesian Optimization for nonlinear response optimization (SOL 400)



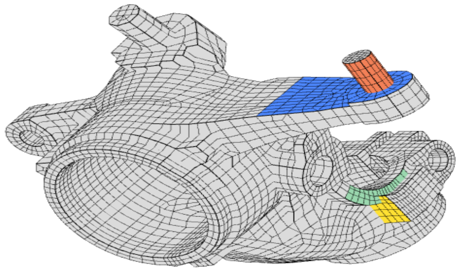
PBMSECT Web App
Generate PBMSECT and PBRSECT entries graphically



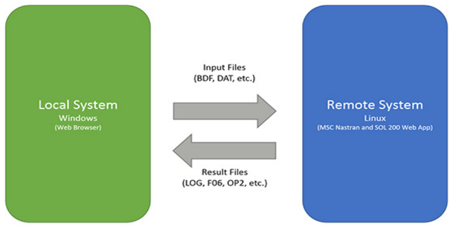
Ply Shape Optimization Web App
Optimize composite ply drop-off locations, and generate new PCOMPG entries



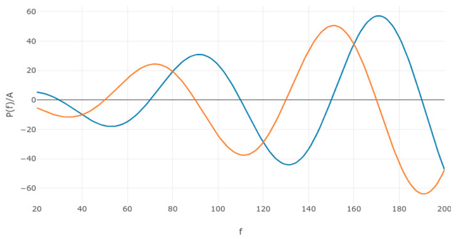
Post-processor Web App
View MSC Nastran results in a web browser on Windows and Linux



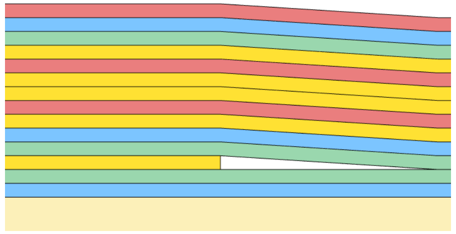
Shape Optimization Web App
Use a web application to configure and perform shape optimization.



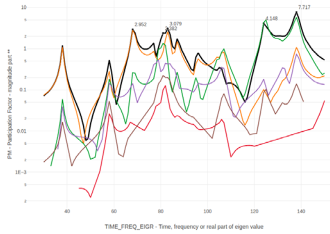
Remote Execution Web App
Run MSC Nastran jobs on remote Linux or Windows systems available on the local network



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



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

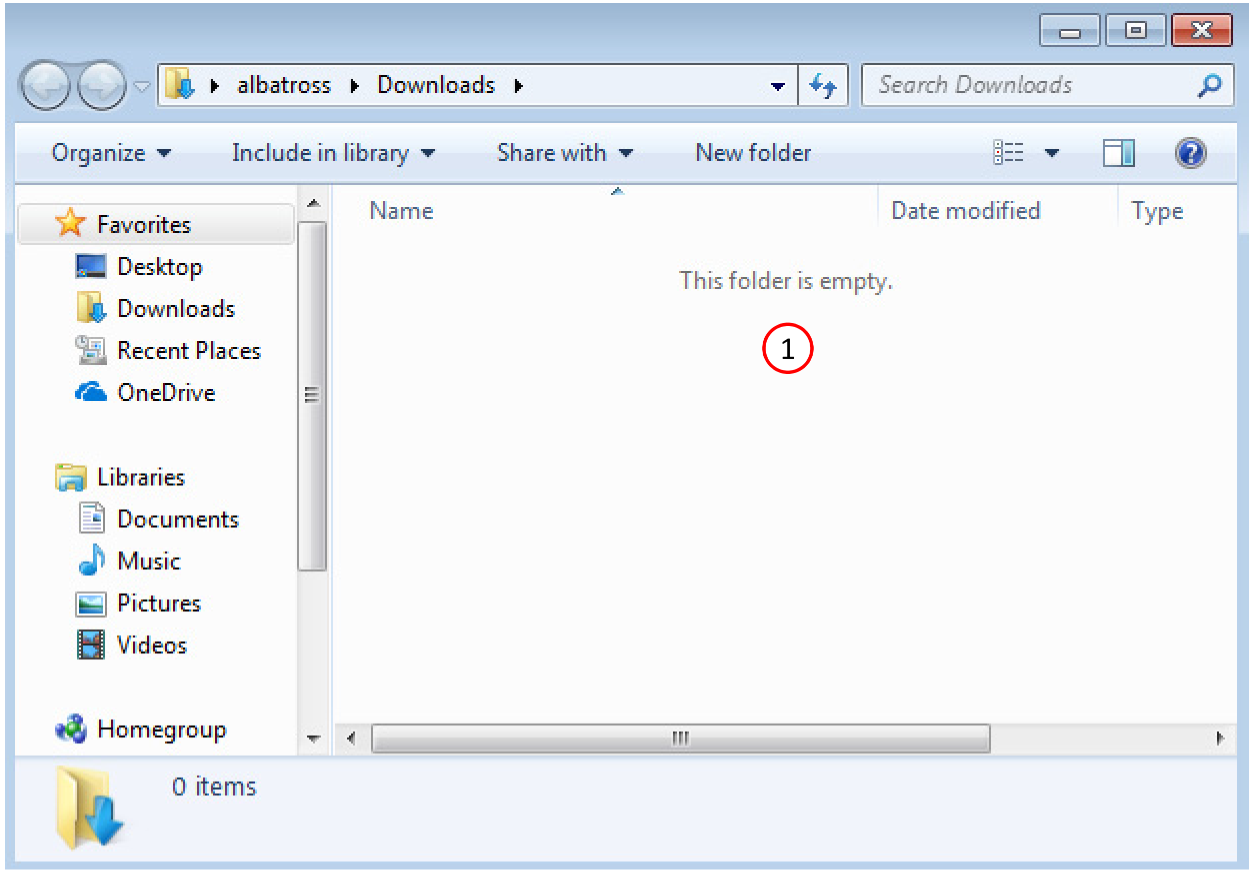


HDF5 Explorer Web App
Create graphs (XY plots) using data from the H5 file

Before Starting

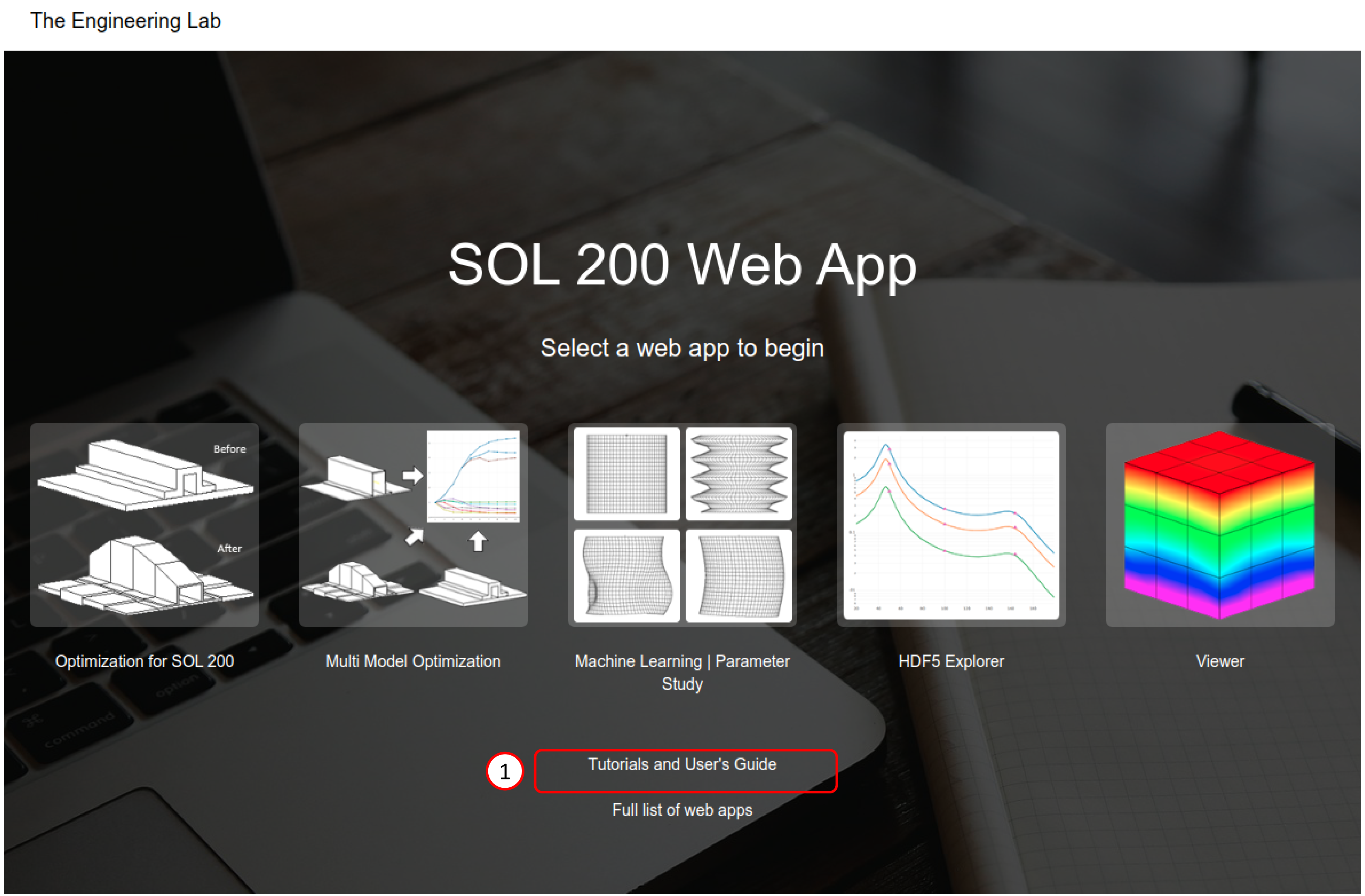
1. Ensure the Downloads directory is empty in order to prevent confusion with other files

- 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

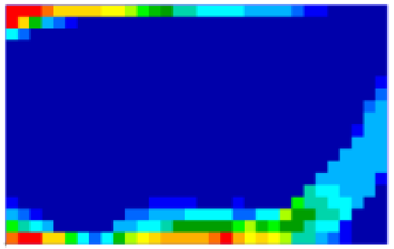


- The necessary BDF files for this tutorial are available in the Tutorials section of the User's Guide.

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.

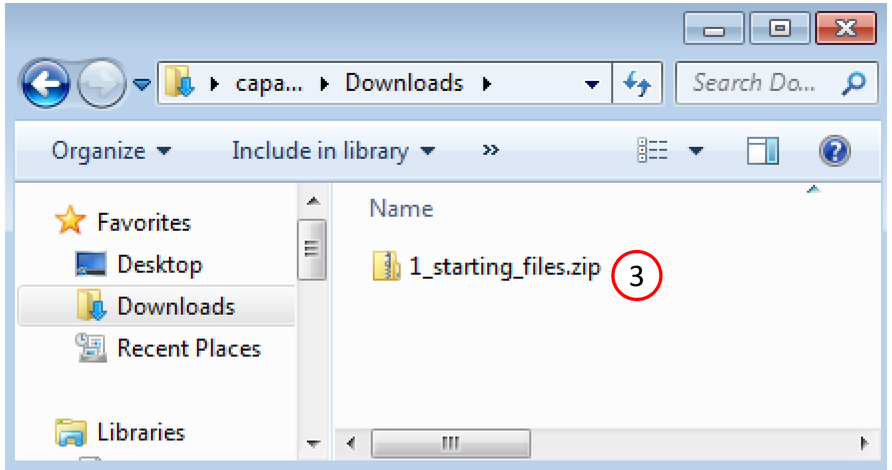


1 **Composite Panel – Phase C – Topometry Optimization to Determine Optimal Core Shape**

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.

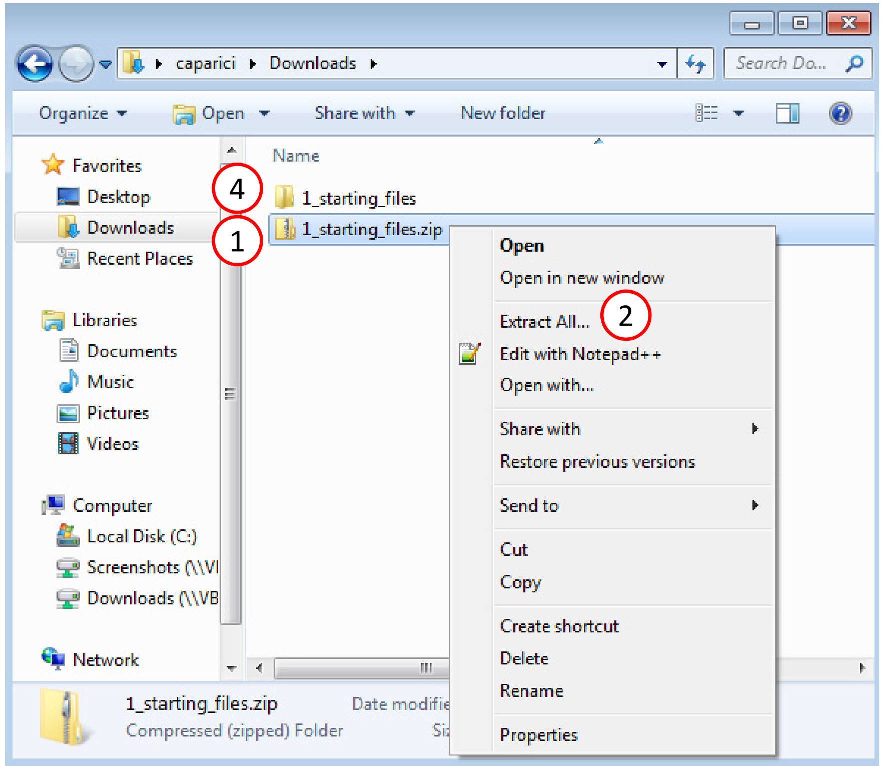
Starting BDF Files: Link 2
Solution BDF Files: Link

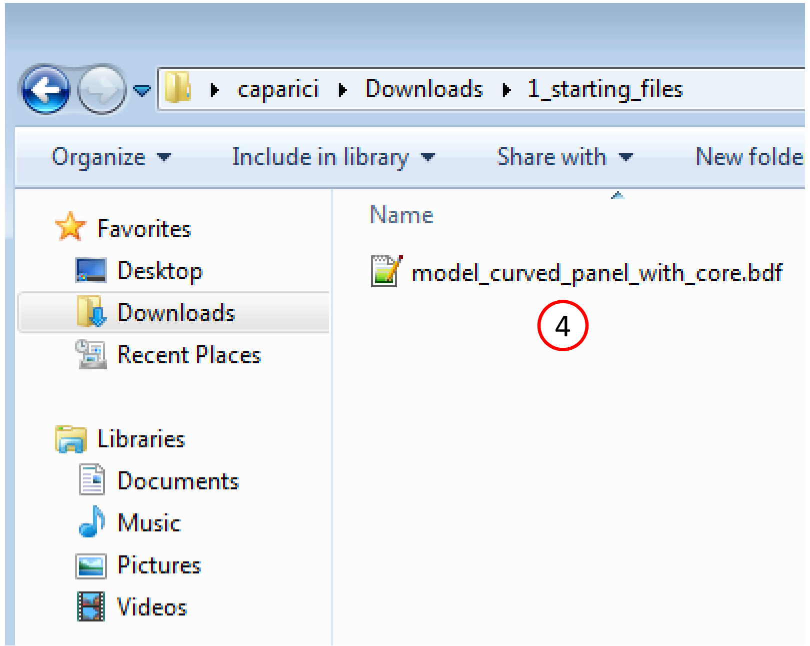


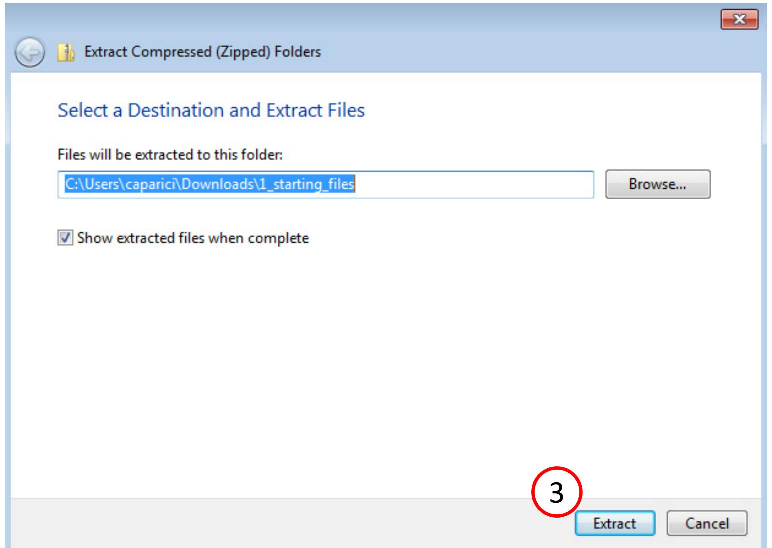
Obtain Starting Files

1. Right click on the zip file
2. Select Extract All…
3. Click Extract
4. The starting files are now available in a folder

- This workflow works best when the BDF has the following configuration:
  - PCOMP entries are used
  - The Ti fields on the 2D element entries, e.g. CQUAD4 and CTRIA3, are NOT used
- See the appendix for more information



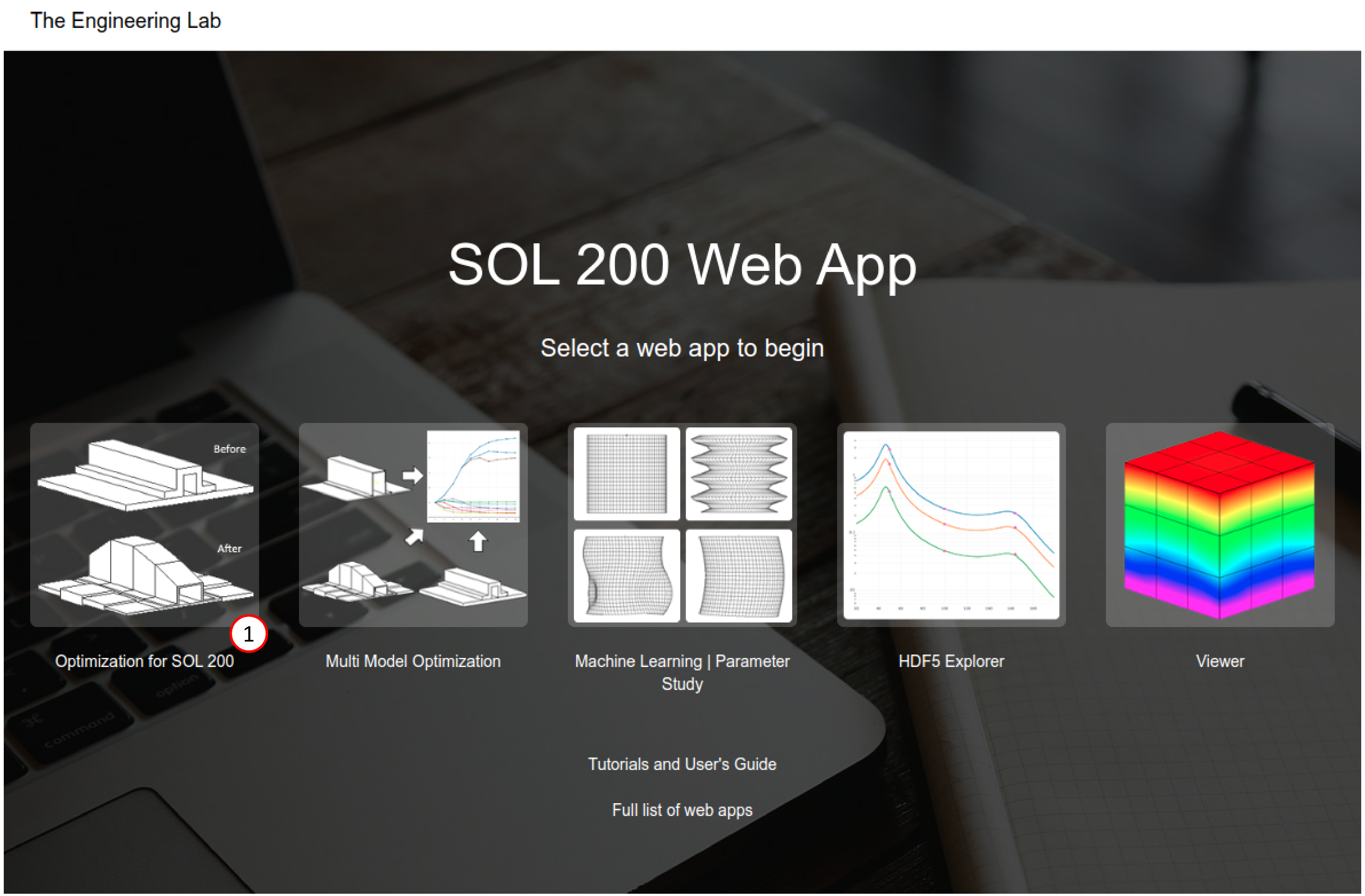




Open the Correct Page

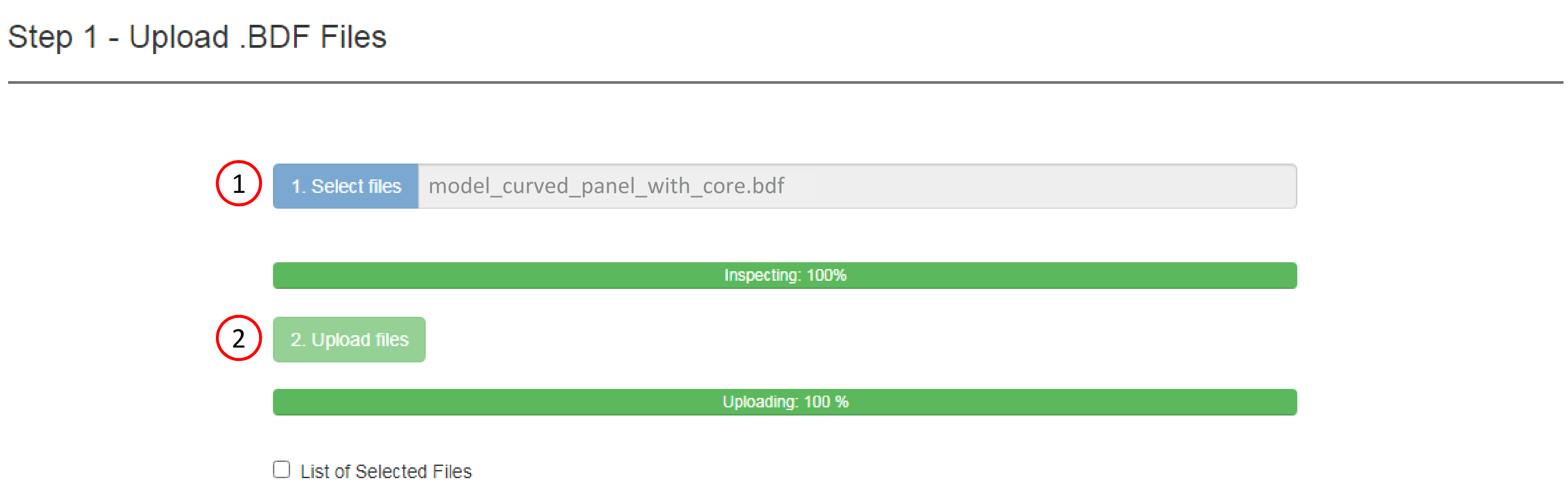
1. Click on the indicated link

- 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 model_curved_panel_with_core.bdf
2. Click Upload Files

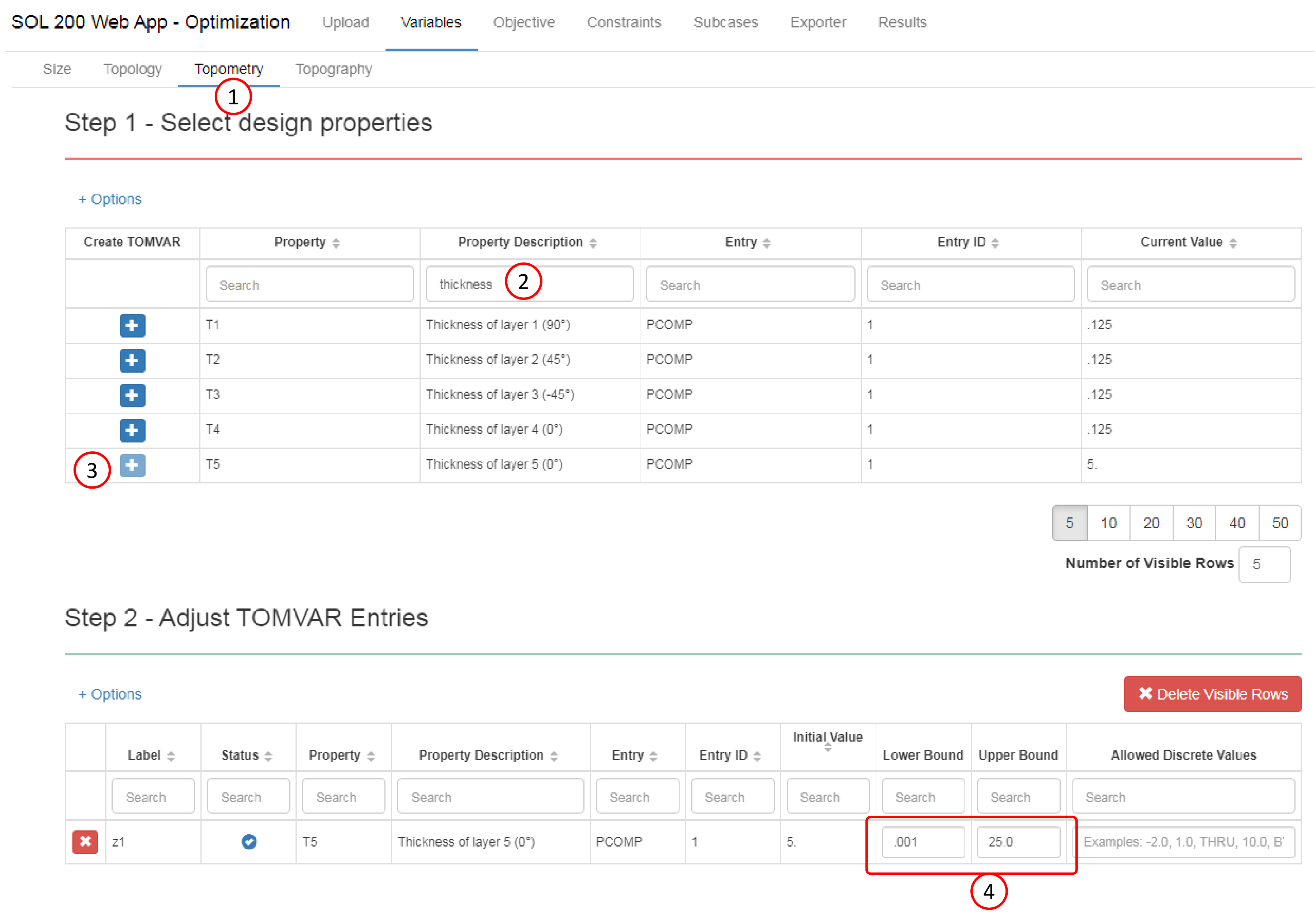


- The process starts by uploading all the necessary BDF files. The BDF files can be files of your own or files found in the Tutorials section of the User's Guide.

Variables

1. Click Topometry
2. In the search box, search for: thickness
3. Click the indicated plus (+) icon to set the 5th layer as a design variable
4. Configure the following values:
   - Lower Bound: .001
   - Upper Bound: 25.0

- The core's thickness may be manufactured in the range of 6mm to 50mm. Since the PCOMP entry is configured with the LAM=SYM option, only half the core thickness needs to be specified. The upper bound is set to 50/2=25mm. The lower bound could be set to 6/2=3mm, but a very small value of .001 is used instead. This is done to see if there is a region of the model that takes on very small thickness values of the core. Very small core thickness values could be indication the core is negligible and could lead to additional mass reduction.



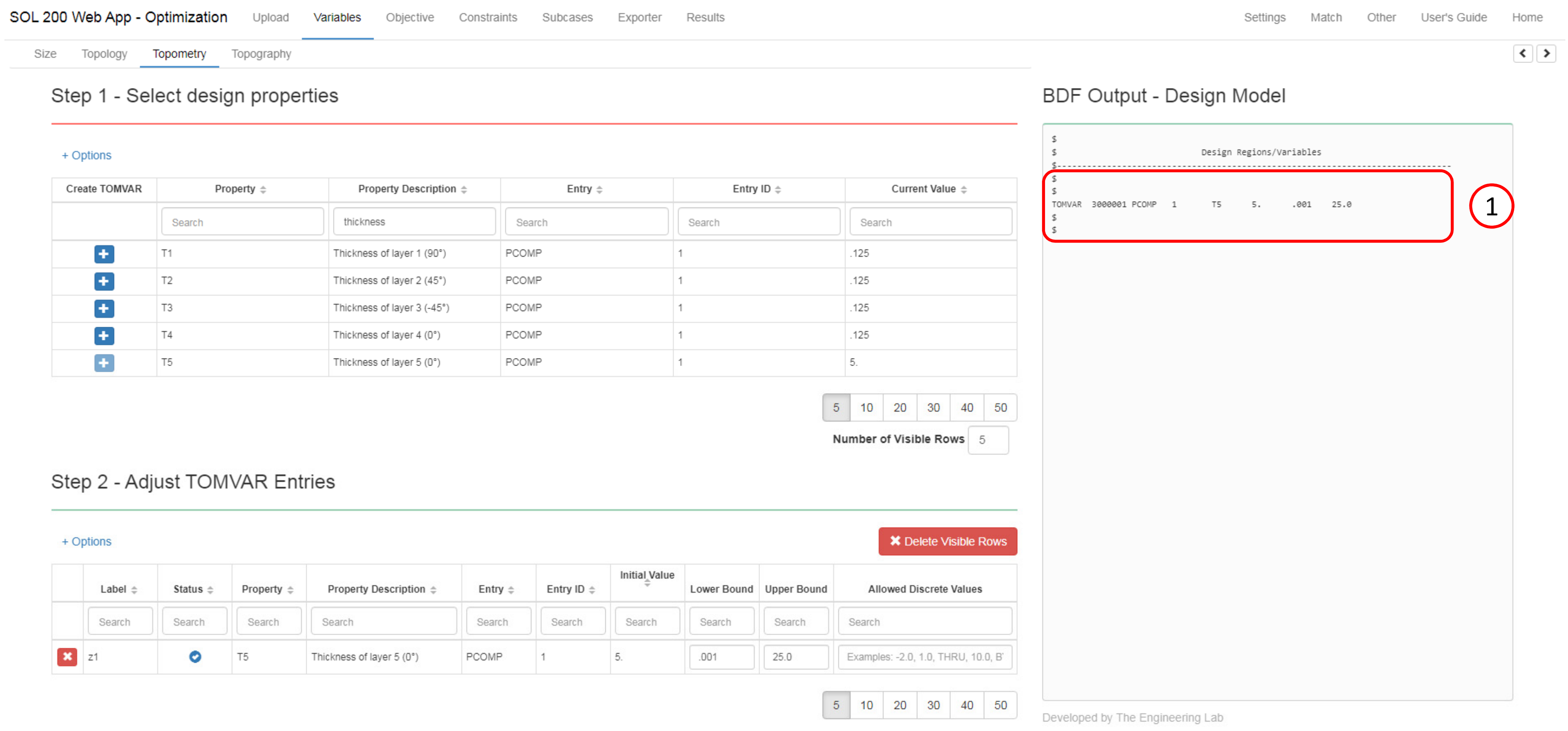
TOMVAR Entries

1. The indicated TOMVAR entries have been created for the thicknesses of layers 5. Layer 5 is the core.

It is expected that files model.ply0005 are available to construct optimal core shapes. The topometry optimization will output a model.ply0005 file.

1

```
$TOMVAR ID      TYPE    PID     PNAME   XINIT   XLB     XUB     DELXV
TOMVAR  3000001 PCOMP   1       T5      5.      .001    25.0
```



Create Design Objective

1. Click Objective
2. Select the plus (+) icon for weight
3. The objective has been set to minimize the weight, no further modification is necessary

- The objective must always be a single and global response. A response such as weight and volume are single responses, are independent of load case, 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.

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

1

Objective Equation Objective

Step 1 - Select an objective

Select an analysis type

SOL 105 - Buckling

Select a response

	Response Description	Response Type
	Search	Search
2 +	Weight	WEIGHT
+	Volume	VOLUME
+	Buckling Eigenvalue/Factor	LAMA
+	Weight from Particular Material or Property ID	WMPID
+	Fractional Mass	FRMASS

5 10 20 30 40 50

Step 2 - Adjust objective

+ Options

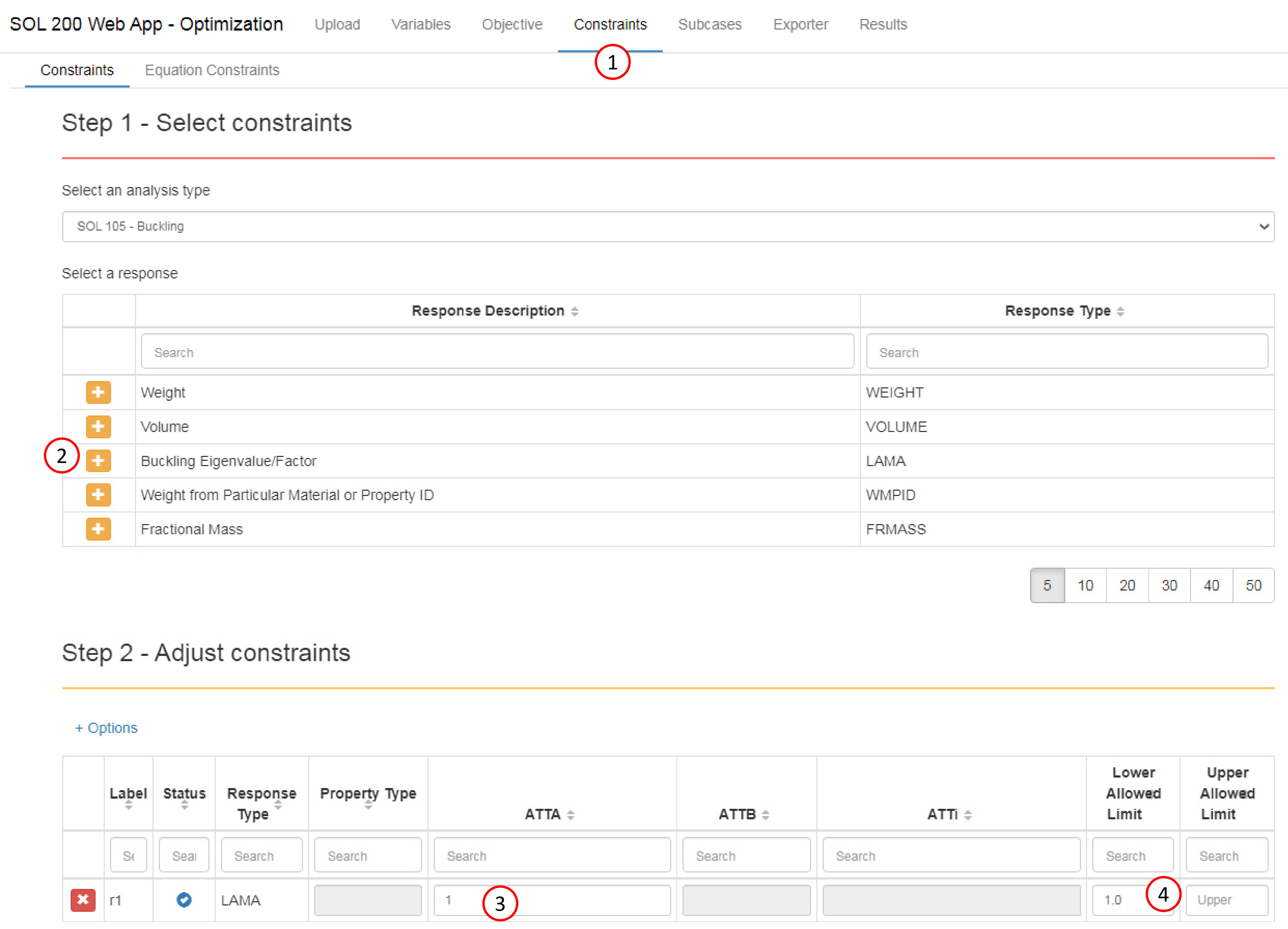
	Label	Status	Response Type	Maximize or Minimize	Property Type	ATTA	ATTB	ATTi
✖	r0	✔	WEIGHT	MIN	3	3	3	

Questions? Email: christian@ the-engineering-lab.com

HEXAGON Technology Partner

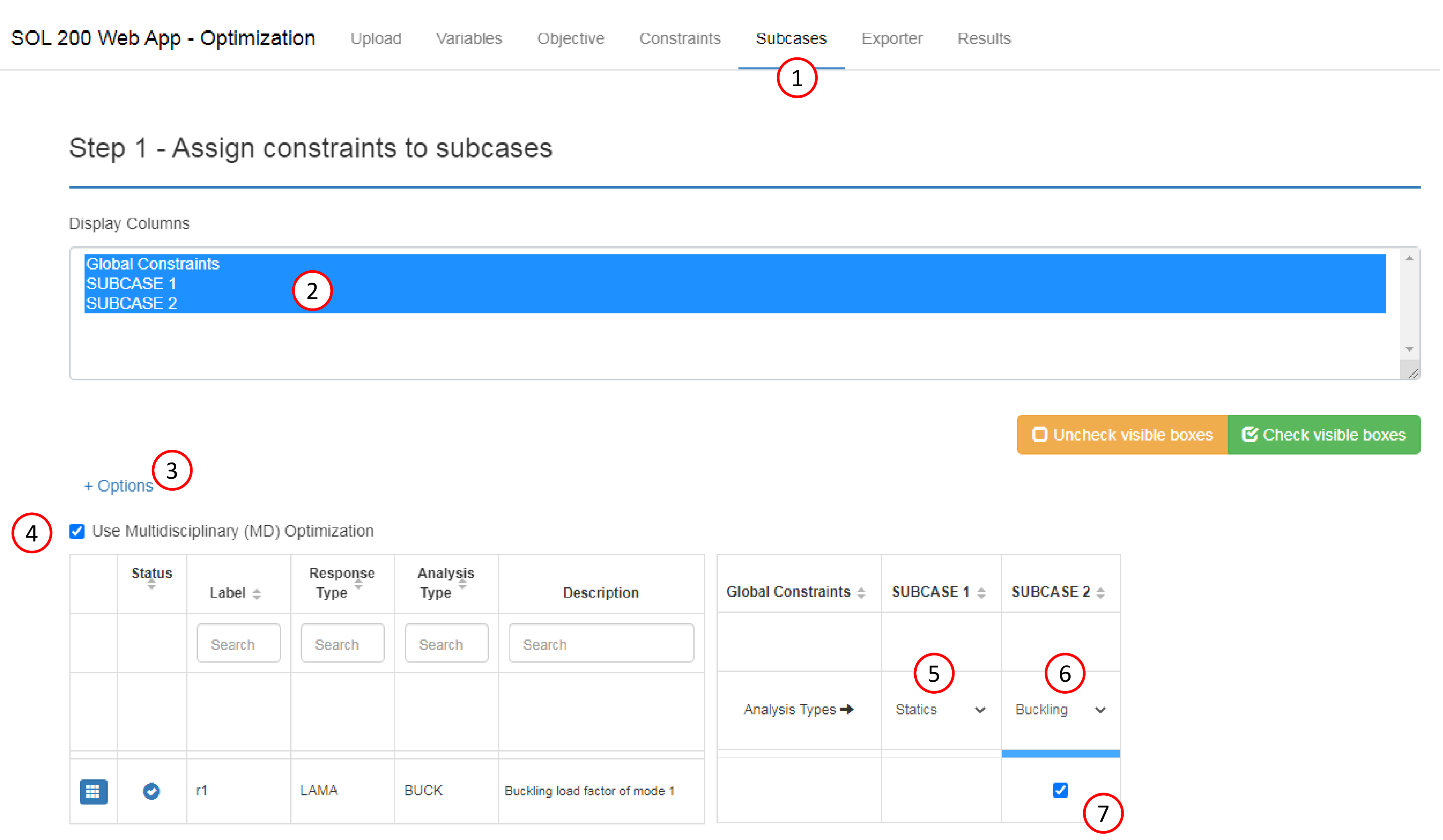
Create Design Constraints

1. Click Constraints
2. Click the plus (+) icon for Buckling Eigenvalue/Factor to create one constraint
3. Set the Buckling Mode Number (ATTA) to 1
4. Set the Lower Allowed Limit to 1.0



Assign Constraints to Load Cases (SUBCASES)

1. Click Subcases
2. Select all the subcases
3. Click +Options
4. Mark the checkbox for Use Multidisciplinary (MD) Optimization
5. For subcase 1, set the Analysis Type as Statics
6. For subcase 2, set the Analysis Type as Buckling



- The r1 constraint has been assigned to SUBCASE 2
- When hundreds of SUBCASEs must be configured, the following options expedite the process:

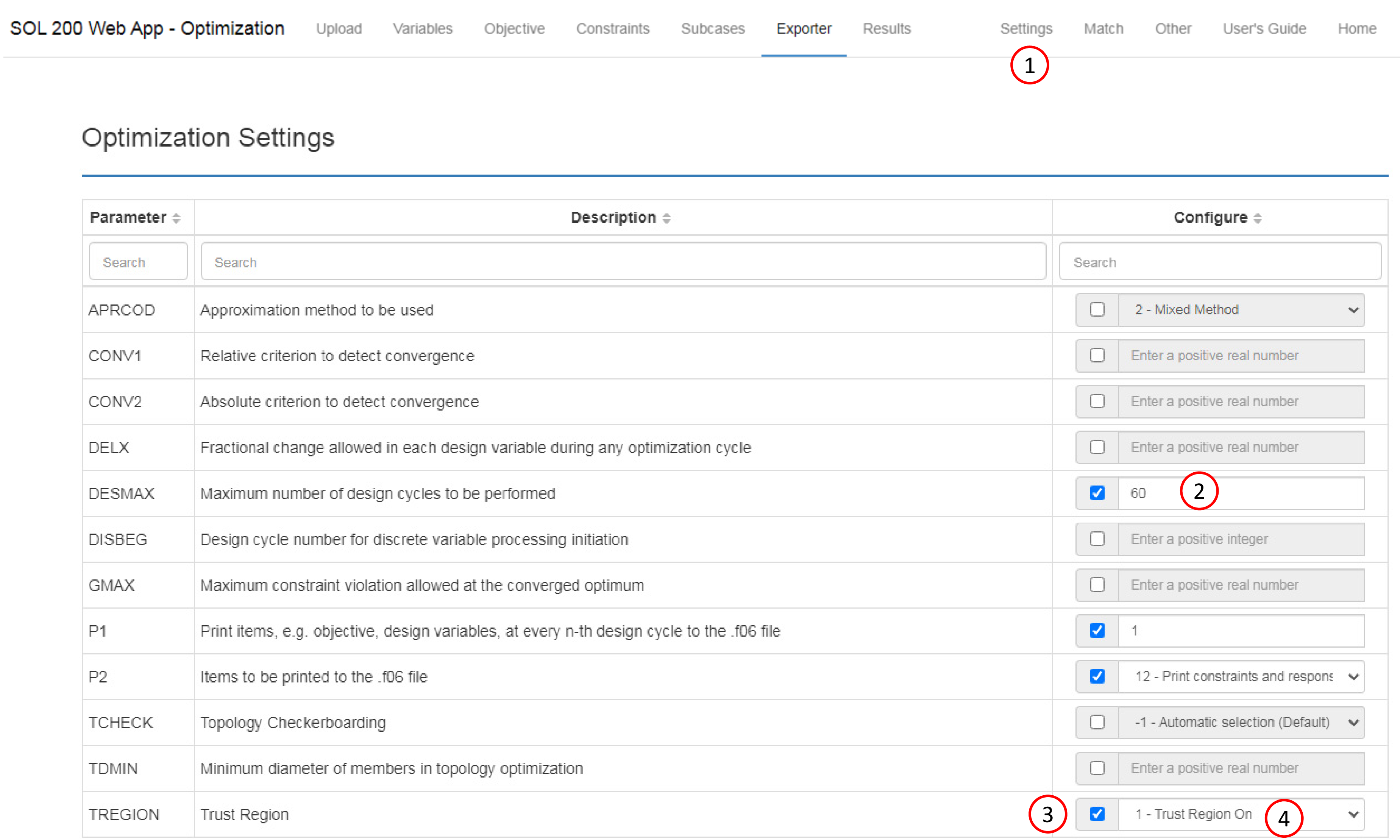
Uncheck visible boxes

Check visible boxes

Settings

1. Click Settings
2. Set the maximum number of design cycles to 60. This is done because when there are thousands of design variables, it takes longer to converge.
3. Mark the indicated checkbox
4. Turn on the trust region by using value 1 – Trust Region On

- Why is the trust region setting used? Refer to the Appendix to answer this question.



SOL 200 Web App - Optimization | Upload | Variables | Objective | Constraints | Subcases | Exporter | Results | Settings | Match | Other | User's Guide | Home

1

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	☑ 60 (2)
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	☑ 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	(3) ☑ 1 - Trust Region On (4)

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”

1

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

BDF Output - Model

```
assign userfile = 'optimization_results.csv', status = unknown,
form = formatted, unit = 52
$ MSC.Nastran input file created on May       23, 2023 at 07:05:29 by
$ Patran 2022.2
$ Direct Text Input for Nastran System Cell Section
$
SOL 200
CEND

ECHO = PUNCH(NEWBULK)
TITLE = MSC.NASTRAN JOB CREATED ON 22-MAY-23 AT 09:49:34
   DESOBJ(MIN) = 8000000
   $ DESGLB Slot
   $ DSAPRT(FORMATTED, EXPORT, END=SENS) = ALL
SUBCASE 1
   ANALYSIS = STATICS
   $ DESSUB Slot
   $ DRSPAN Slot
$ Subcase name : Default
   SUBTITLE=Default
   SPC = 2
   LOAD = 5
   DISPLACEMENT(PLOT,SORT1,REAL)=ALL
   SPCFORCES(PLOT,SORT1,REAL)=ALL
SUBCASE 2
   ANALYSIS = BUCK
   DESSUB = 40000002
   $ DRSPAN Slot
$ Subcase name : Default
```

Download BDF Files

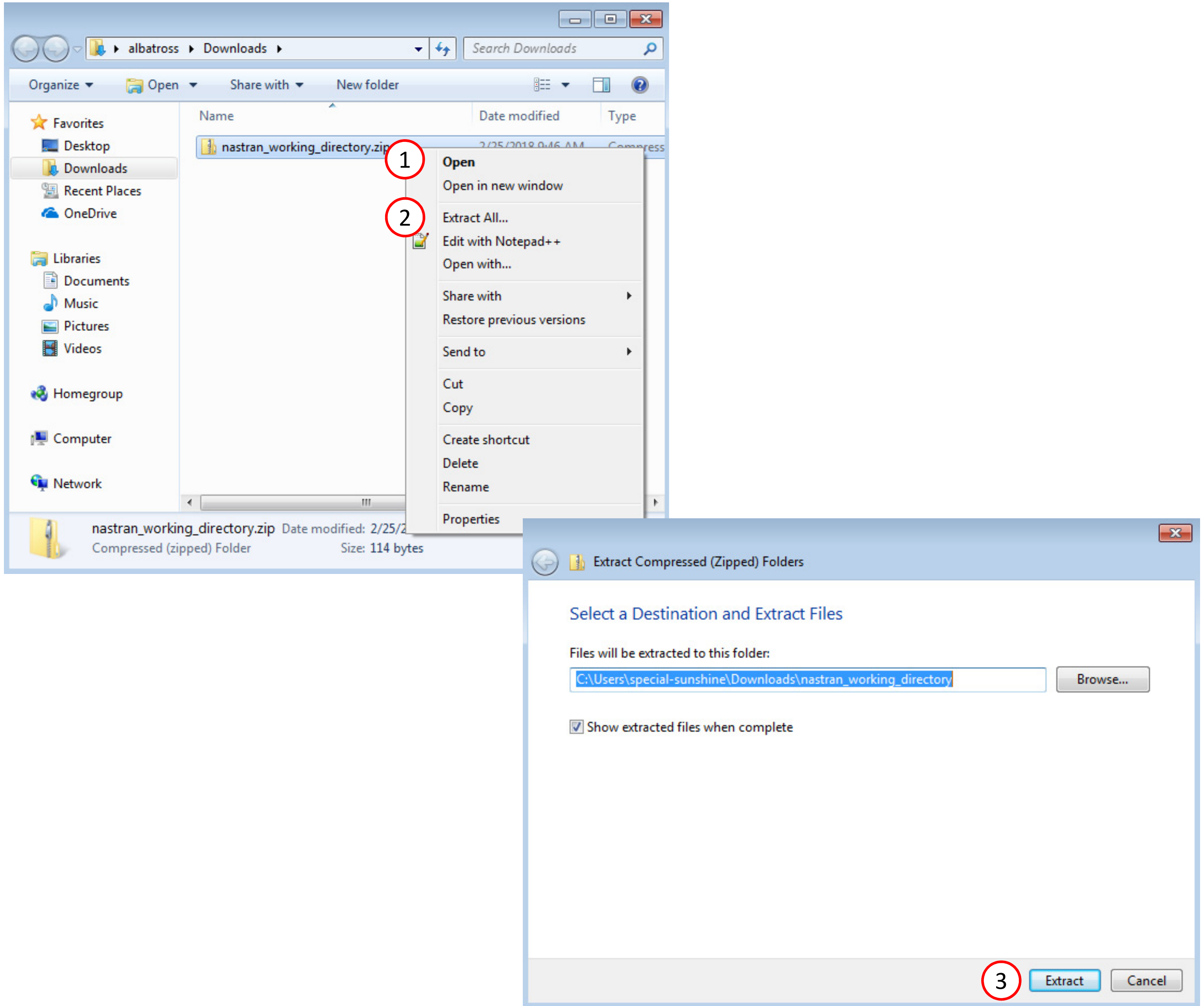
Download BDF Files

2

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

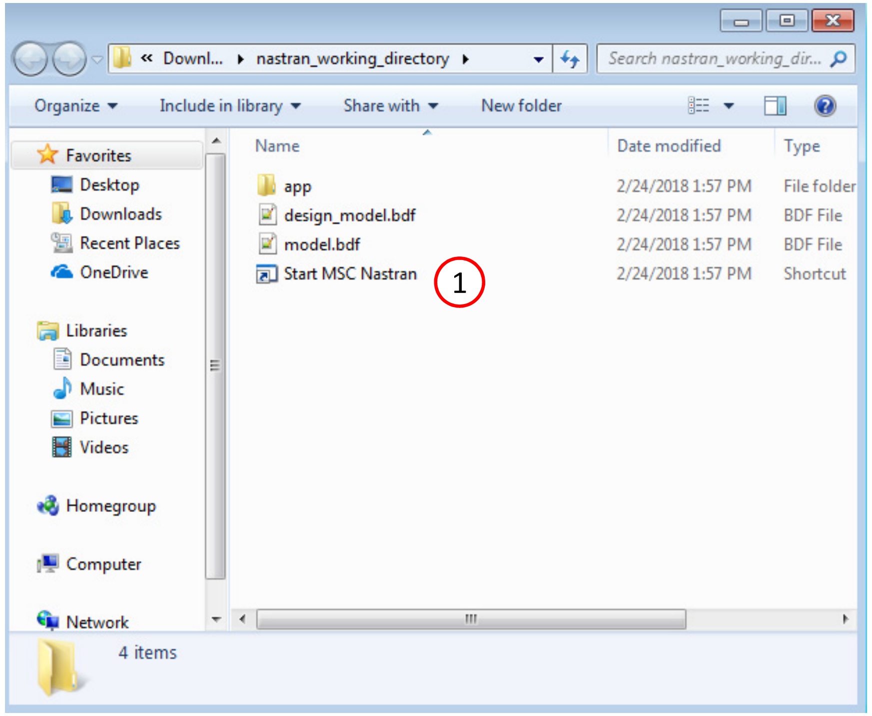
- After a successful optimization, the results will be automatically displayed as long as the following files are present: BDF, F06 and LOG.
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the 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.

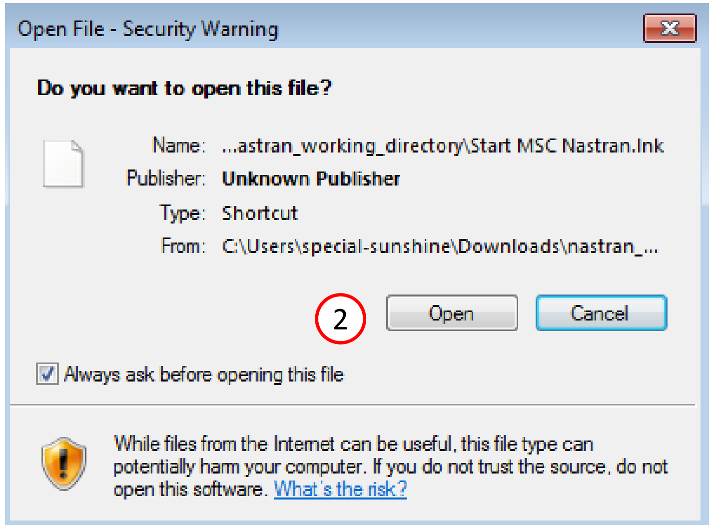
Using Linux?

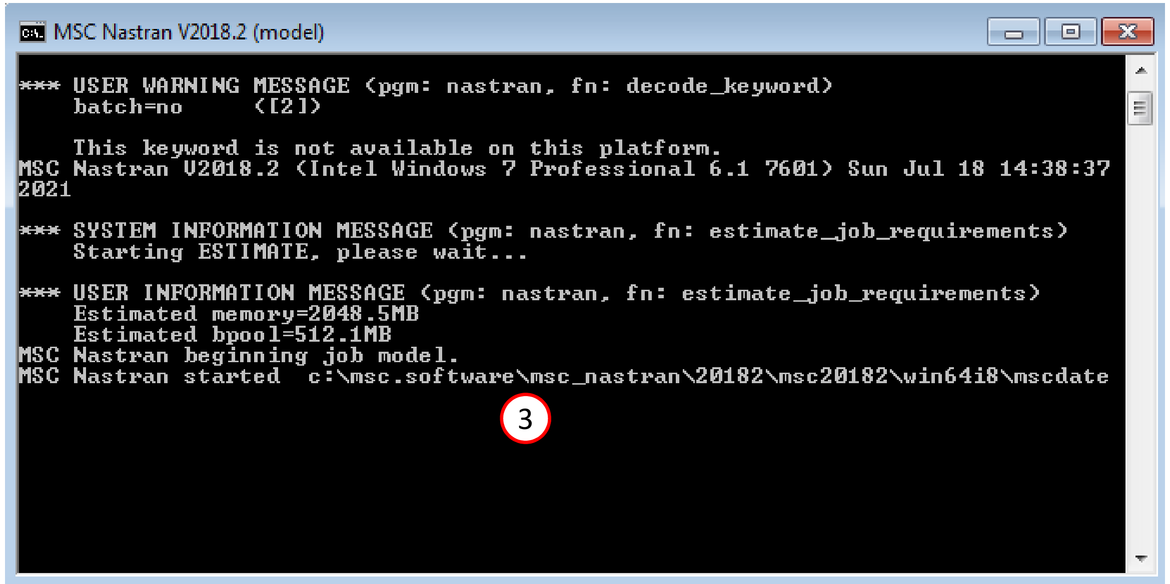
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
cd ./nastran_working_directory
3) Use this command to start the process
./Start_MSC_Nastran.sh

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 and normalized constraints can be reviewed.

- The final max normalized constraint is negative, indicating the design is feasible. The fact the objective was minimized and the final design is feasible indicates this has been a successful optimization.

SOL 200 Web App - Local Optimization Results

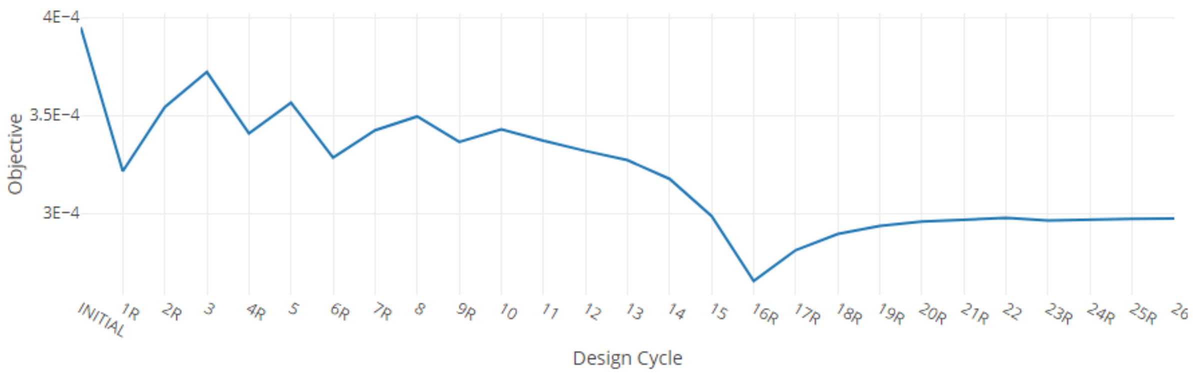
Home

Final Message in .f06

1

RUN TERMINATED DUE TO HARD CONVERGENCE TO AN OPTIMUM AT CYCLE NUMBER = 26.

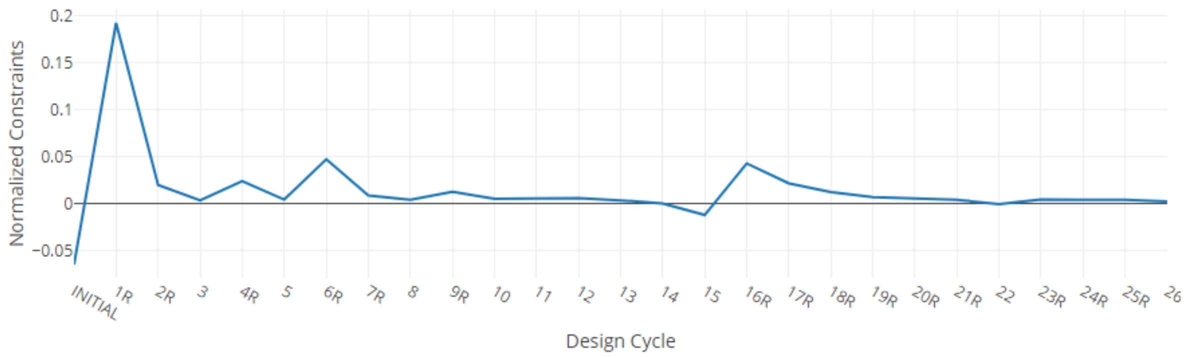
Objective



2

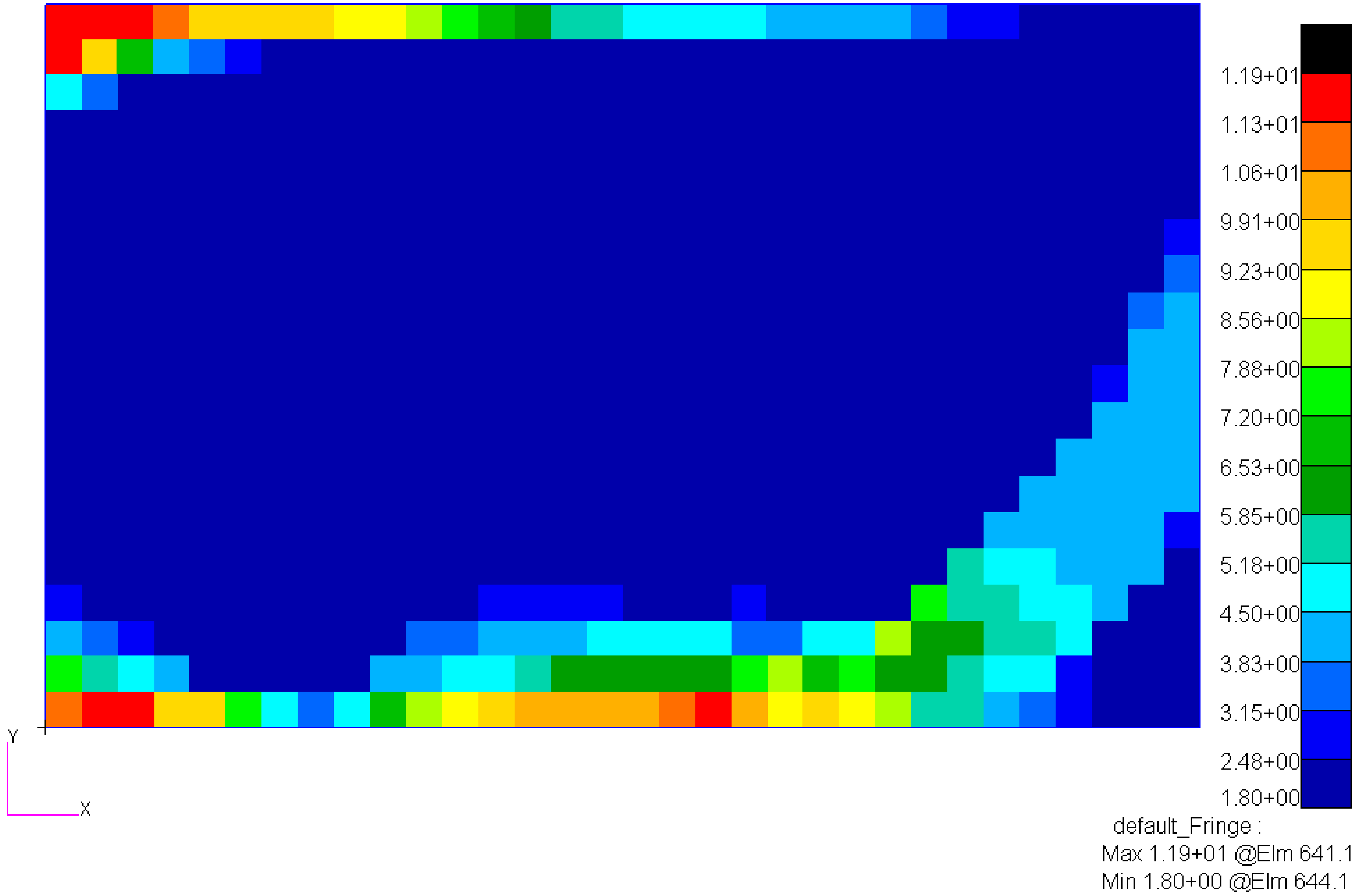
Normalized Constraints

+ Info



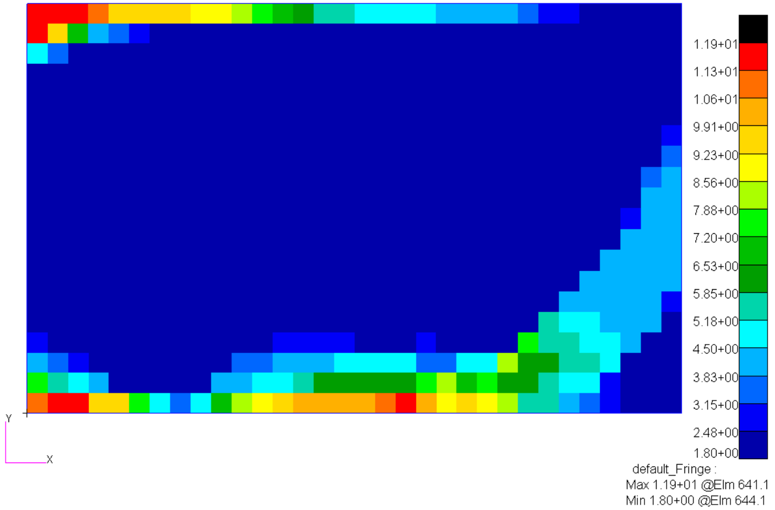
View Thickness Distribution

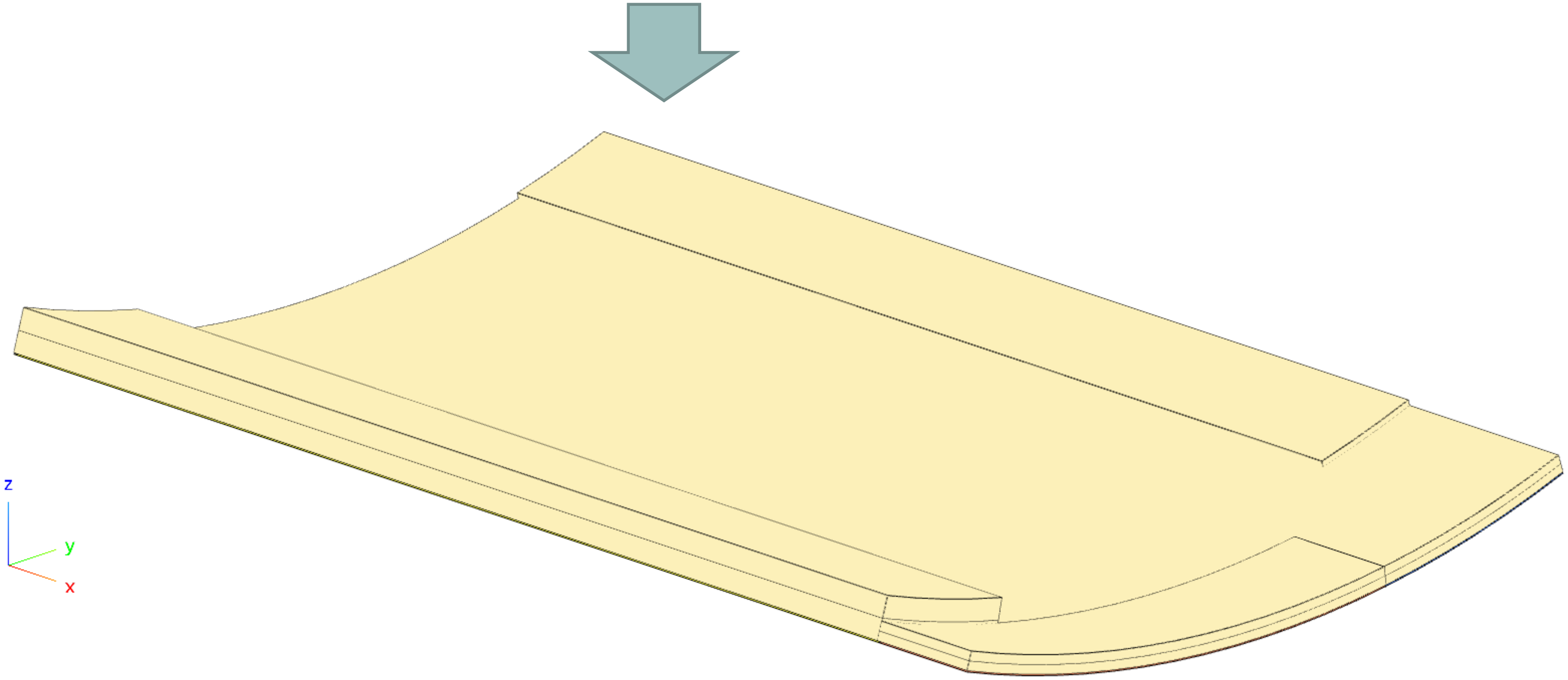
Patran is used to display the thickness distribution of the 5th layer of the composite, which is the core layer.



Next Steps

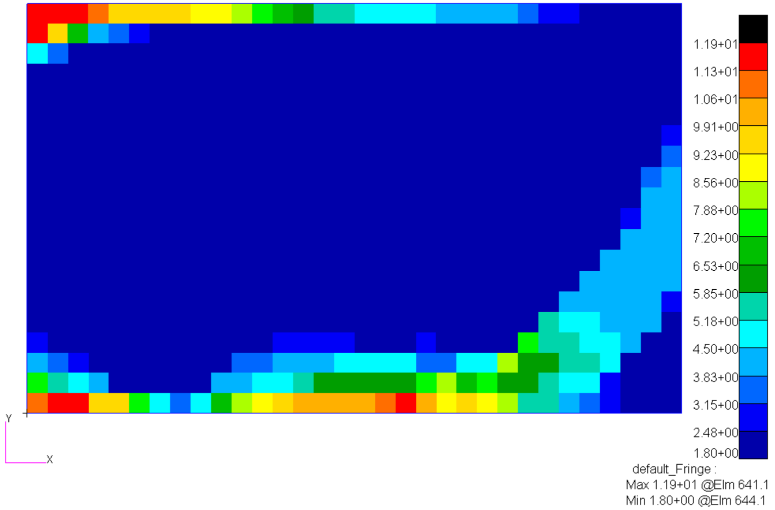
- In a subsequent tutorial, the results of the topometry optimization will be used to define multiple core sections, each with their own thickness.

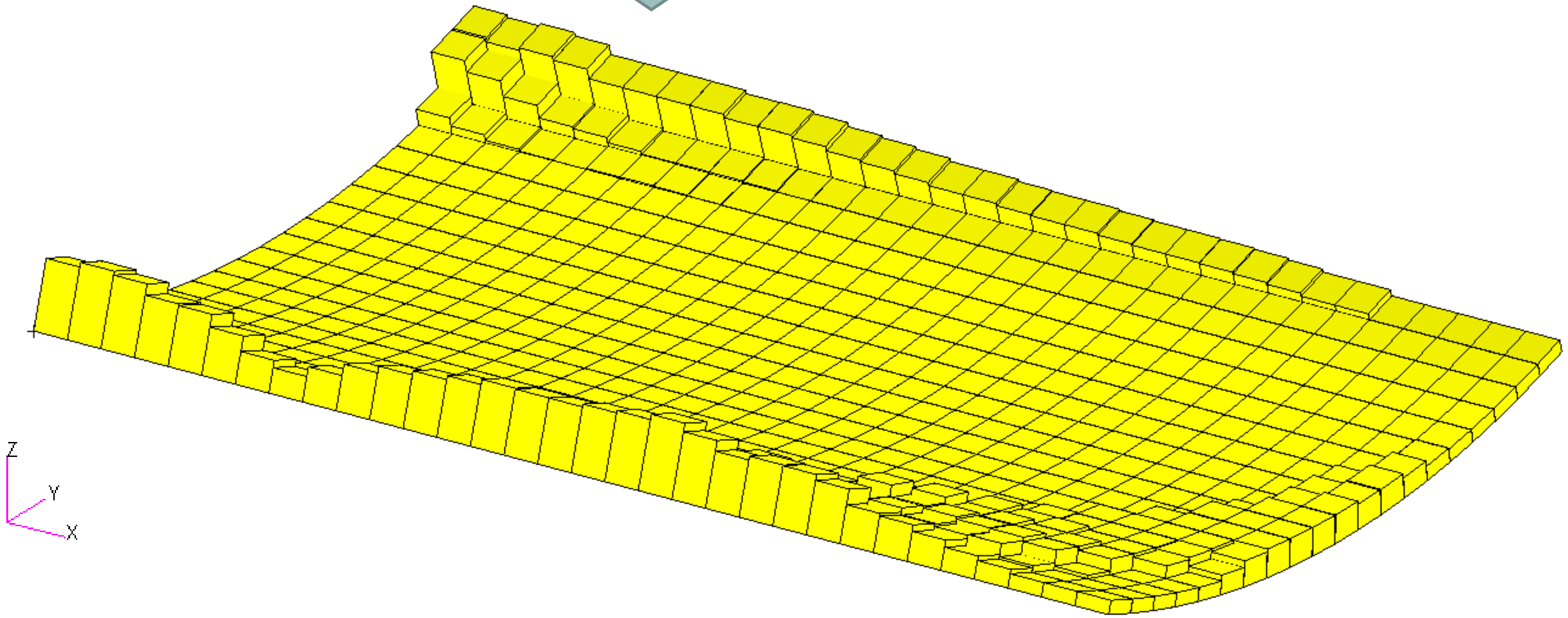




Using the Topometry Optimization Results as Is

- If, instead, you would like to use the topometry optimization results as is, the steps detailed in the appendix, section *Using the Topometry Optimization Results as Is*, show how to update the BDF file with the results from the topometry optimization.





End of Tutorial

Appendix Contents

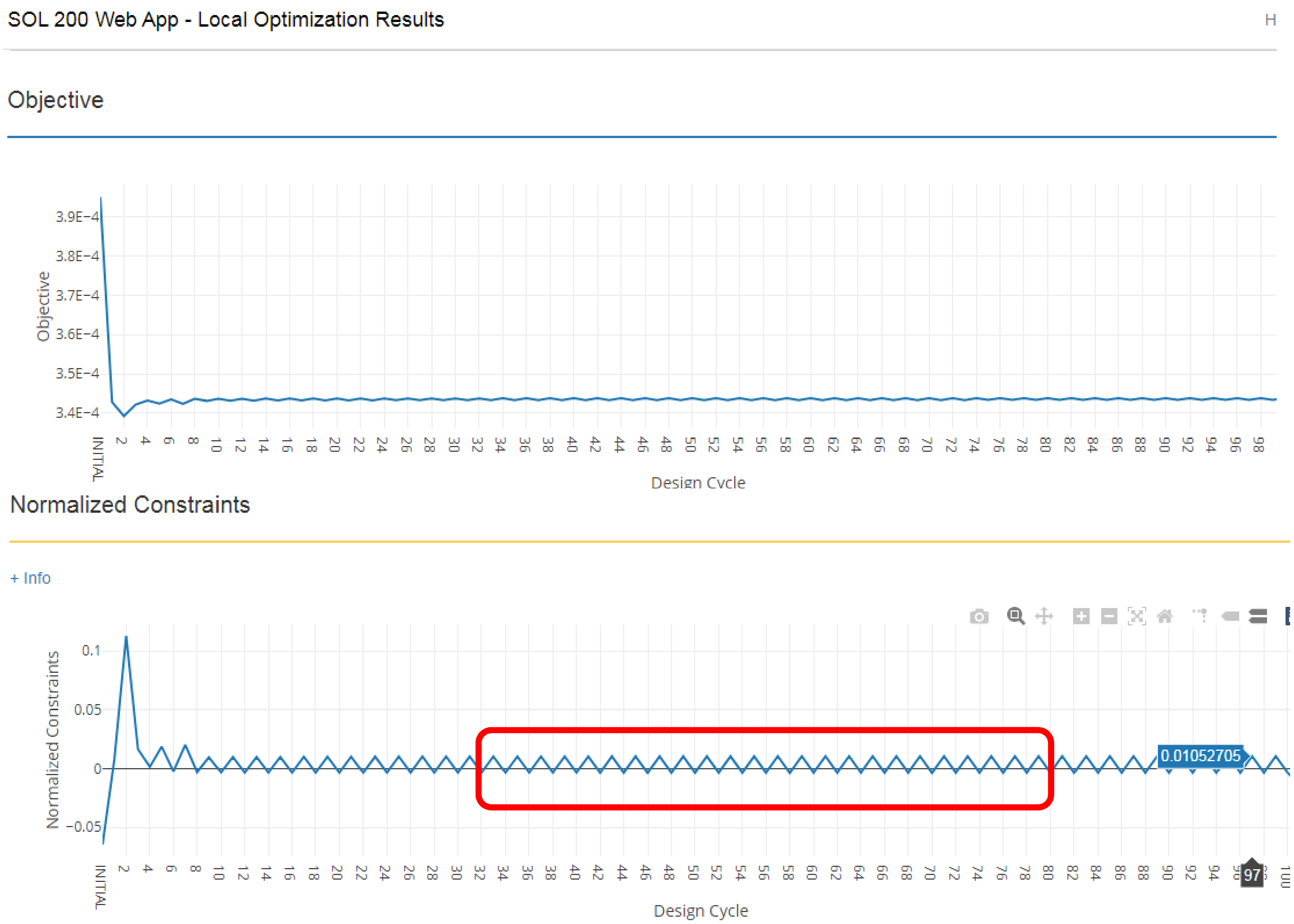
- Why Use the Trust Region?
- Using the Topometry Optimization Results as Is

Why Use the Trust Region?

Why Use the Trust Region?

A rare number of optimizations will fail to converge because the move limits are not reduced and remain constant near the optimum point.

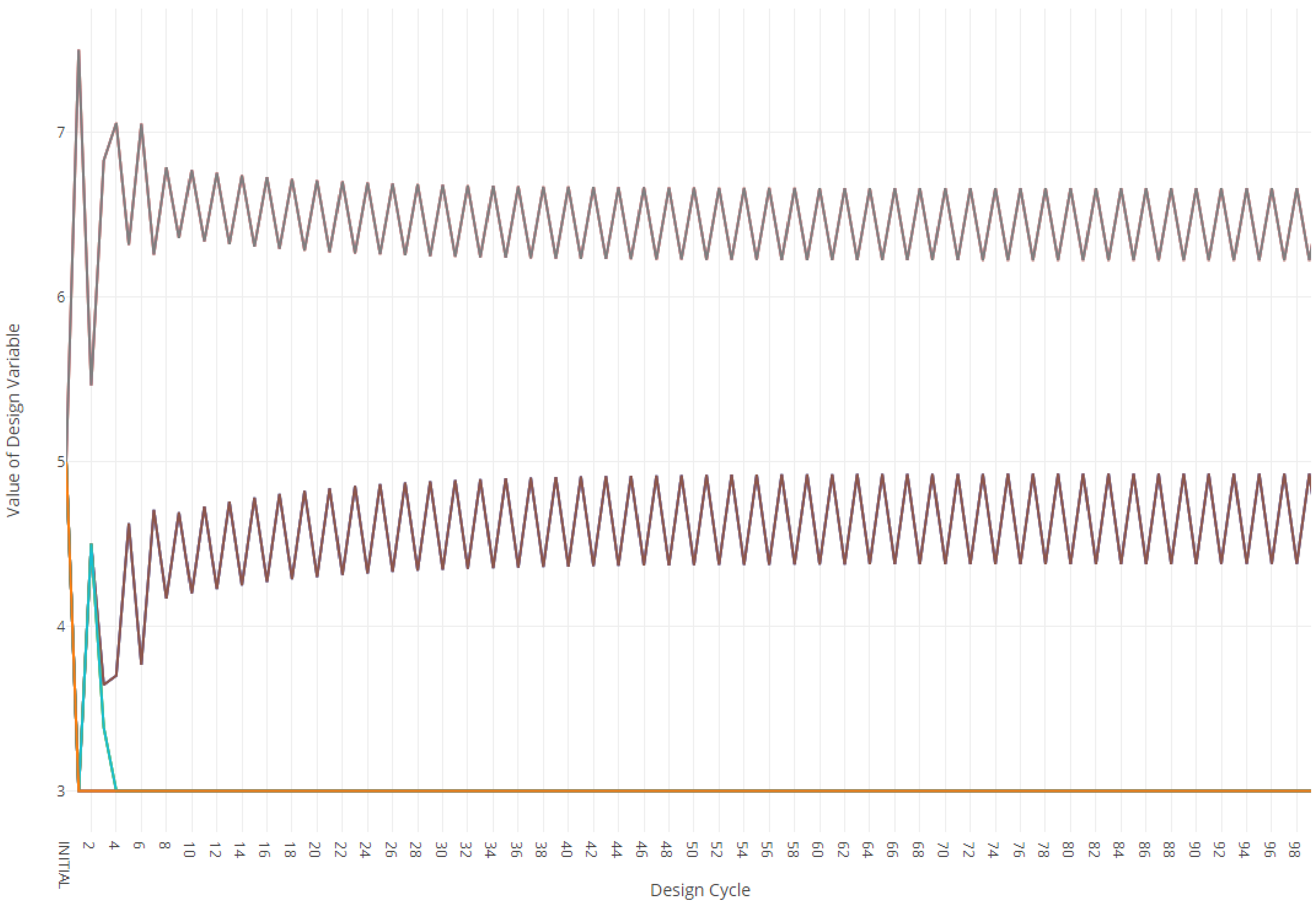
Evidence of this is when the optimization history plots have a zig zag pattern. On this page, the max normalized constraint plot has a zig zag pattern.



Why Use the Trust Region?

On this page, the variable plot also has a zig zag pattern. The optimizer endlessly selects the same variables.

Design Variables



Why Use the Trust Region?

This issue is due to 2 reasons:

1. The move limits remain constant, which means the approximate model remains constant.
2. The optimizer believes it is selecting feasible designs/variables, but the designs are actually infeasible.

Why are the move limits not being changed?

The MSC Nastran Design Sensitivity and Optimization User's Guide provides information regarding the move limits.

The move limits are reduced when the max normalized constraint is greater than +.02 (2%) and continues to increase.

MSC Nastran Design Sensitivity and Optimization User's Guide

Automatic Updates of Move Limits

Parameters related to design variables can be changed using the DOPTPRM entry (DELX and DXMIN) and the DESVAR entry (DELXV, XLB, XUB). The designed property limits can be controlled using the DOPTPRM entry (DELP, DPMIN) and the DVPREL1 or DVPREL2 entries (PMIN, PMAX). This set of constants is used to recompute the move limits for each design cycle.

At times, the code may automatically adjust these move limits if the design task is performing poorly. The situation might arise as follows: an approximate problem is constructed, from which the optimizer determines a corresponding approximate optimum. Perhaps some of the approximate constraints are critical for this design. The responses are now evaluated by a finite element analysis, and it is determined that rather than just critical, these constraints are actually violated. Discrepancies have thus been detected between the approximate and the true responses.

If these discrepancies continue from one design cycle to the next, it can be taken as an indication that the move limits are probably too wide. Continued constraint violations have an adverse effect on overall convergence. The move limit-controlling parameters are updated automatically in MSC Nastran if the following criteria are satisfied:

1. The current design cycle number is greater than or equal to three.

2. There is at least one violated constraint (violated by more than 2%), and the level of constraint violation is increasing.

Under these conditions, DELP, DPMIN, DELX, and DXMIN are reduced by one-half of their current values. The reason for the first condition is that frequently the optimizer may violate the constraints somewhat as it makes favorable gains in the objective function in the first few cycles. However, if this condition continues, it may indicate that the problem is becoming ill-conditioned as a result of excessive move limits. A corresponding User Warning Message is printed as notification that this update has occurred (see Modification of Move Limit Parameters). If the job is to be restarted, it is recommended that an updated DOPTPRM entry with the new move limits be included in the restart file.

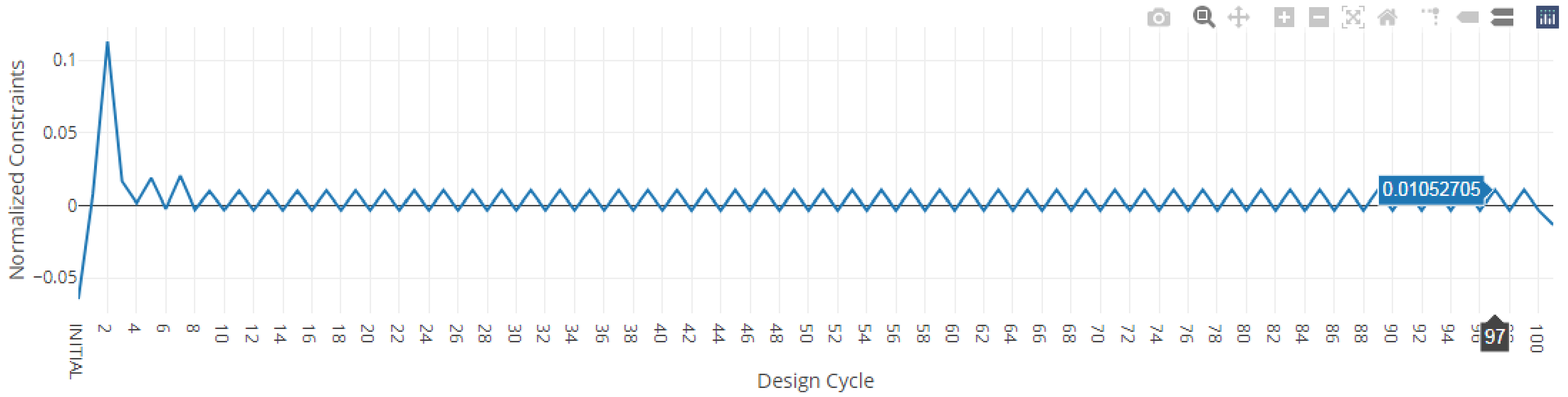
Why Use the Trust Region?

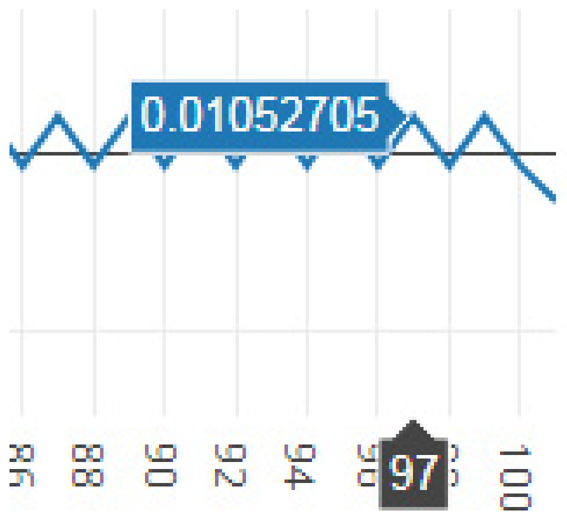
An inspection of the max normalized constraint plot shows a value of +.01052705 (~1.1%) and the violation does not increase, which does not meet the criteria to reduce the move limits.

Since the move limits are not changed, the approximate model remains constant in later design cycles.

Normalized Constraints

+ Info





Why Use the Trust Region?

To keep the number of expensive FE evaluations to a minimum, the optimizer employs an inexpensive approximate model to predict responses during the design cycle. If the true response surface is characterized by a very rough surface, the response is approximated locally by a Taylor series expansion.

At the start of the next design cycle, an FE analysis is performed which helps gauge the accuracy of the approximate model from the previous design cycle. After this FE analysis, a new approximate model is constructed, which the optimizer uses to select the next set of variables. This sequence is repeated.

In the image to the right, the boxes represent the move limits. Variables are allowed to change within these boxes.

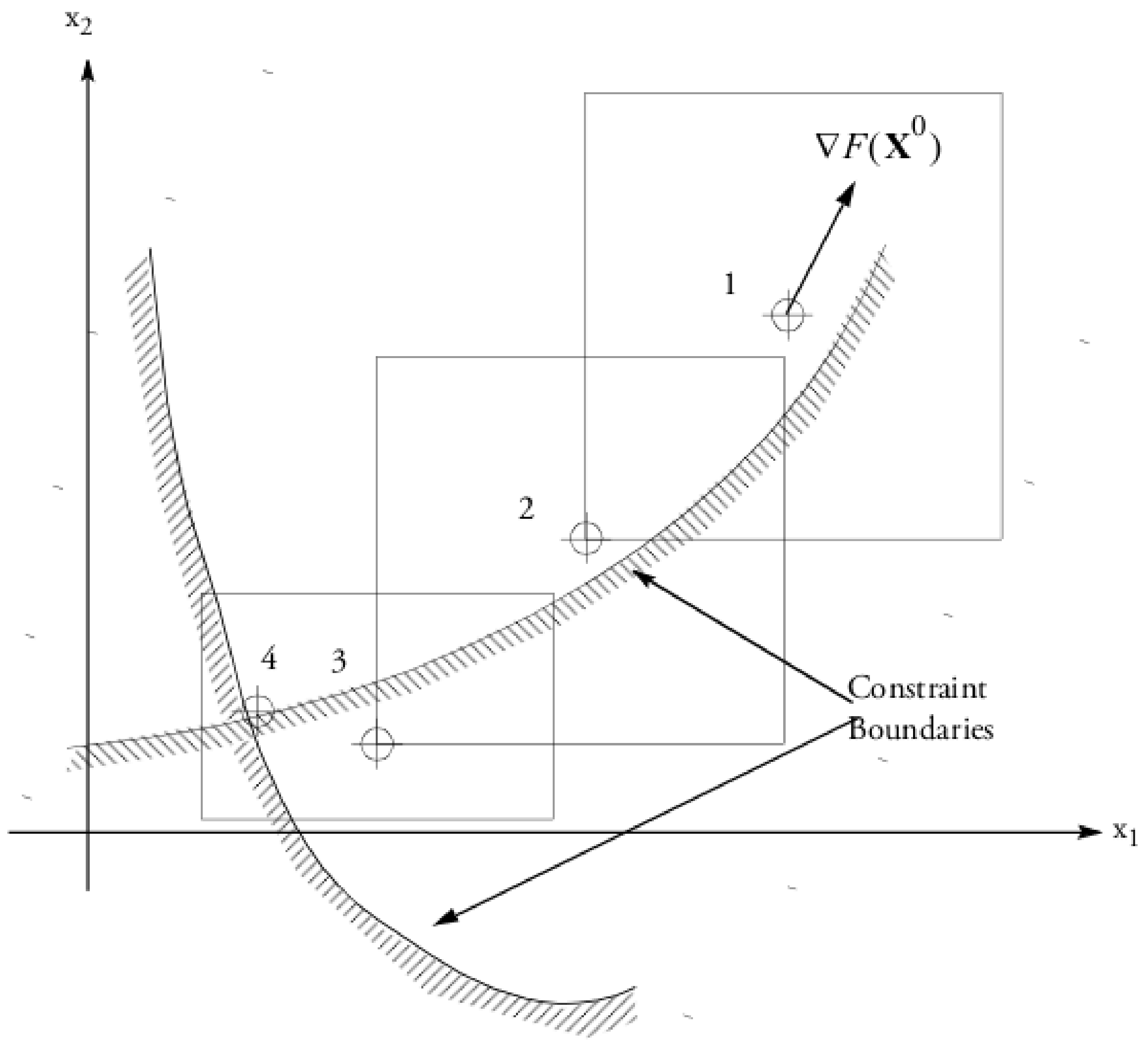


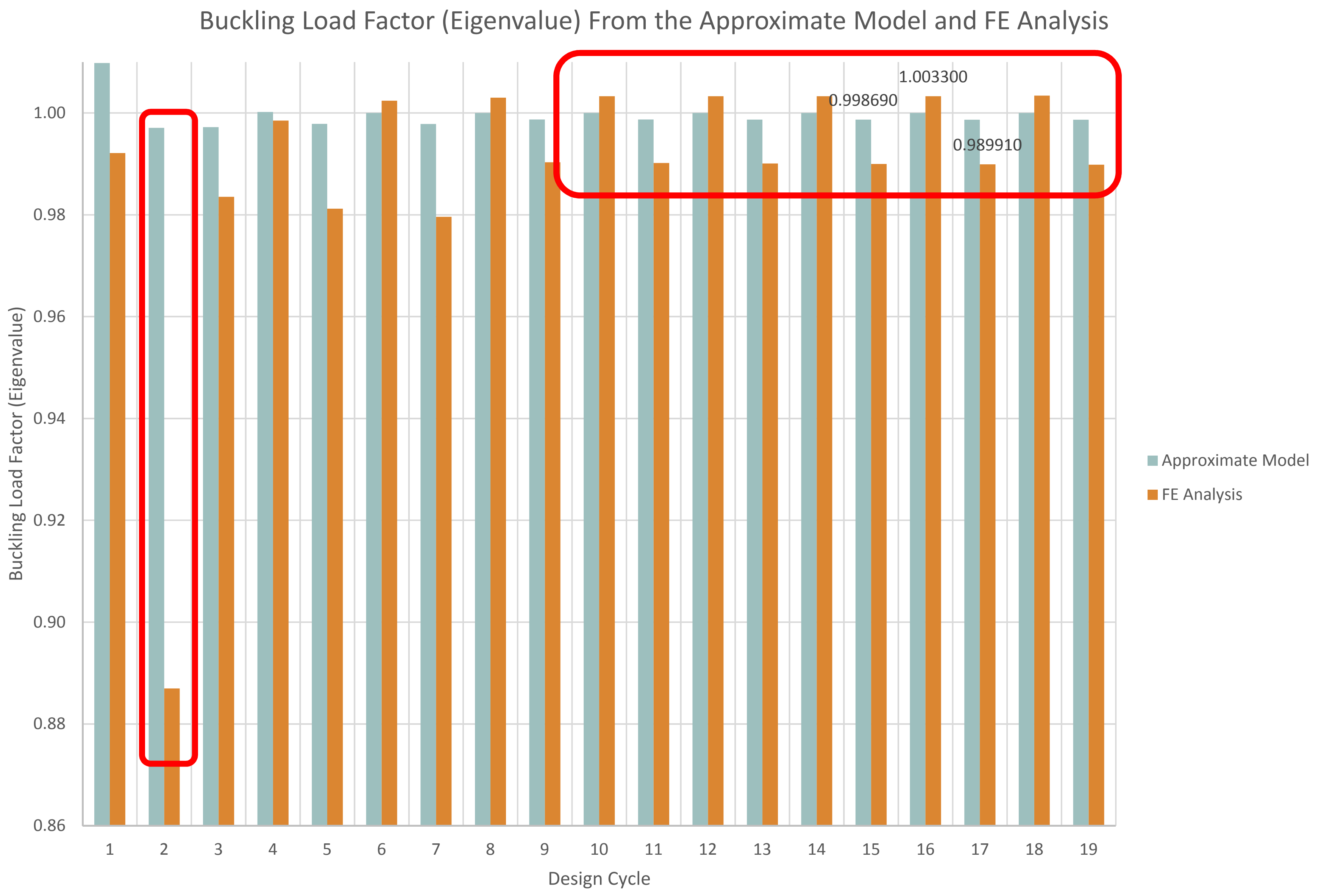
Figure 2-13 Sequence of Approximations

Why Use the Trust Region?

A comparison is made between the buckling load factor predicted by the approximate model and the actual value from an FE analysis. Only the first 20 design cycles are considered.

- After design cycle 2, the biggest discrepancy between the approximate model and the FE model is observed. The actual buckling load factor (BLF) is approximately 8.90E-1, which violates the constraint. The optimizer reduces the move limits, which improves the accuracy of the approximate model. Notice that after design cycle 3, the BLF from the approximate model and FE analysis better align, which indicates the accuracy of the approximate model has been improved after reducing the move limits.

- As discussed on the previous page, certain conditions are not met, so the move limits remain constant, which means the approximate model remains constant in design cycles 10, 11, 12, …, 100. The optimizer is now stuck in a perpetual cycle where the selected designs, according to the approximate model are feasible (BLF~=1.0), but are in actuality infeasible for some design cycles. A new design cycle is performed, but the same points are selected since the approximate model is constant.

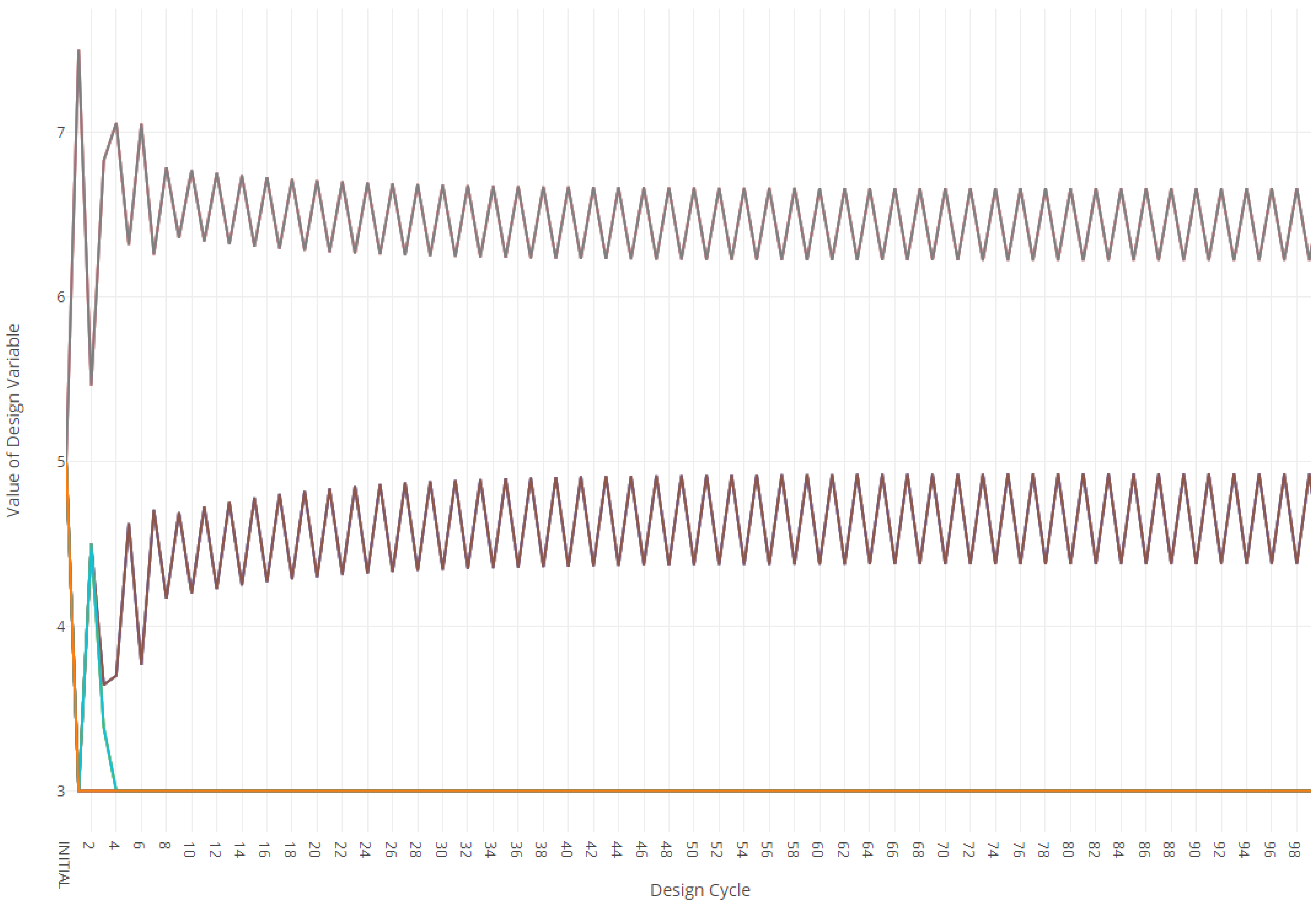


Why Use the Trust Region?

Since the approximate model is constant and predicting feasible designs for the same points, the optimizer perpetually selects the same points.

The only stop criteria to terminate this optimization is the DESMAX setting, which is the maximum number of design cycles.

Design Variables



Why Use the Trust Region?

An alternative is to use the trust region setting which provides an alternative adjustment to the move limits.

To the right is an example of the same optimization that previously had the zig zag pattern, but with trust region setting on. This optimization now converges to a feasible design.

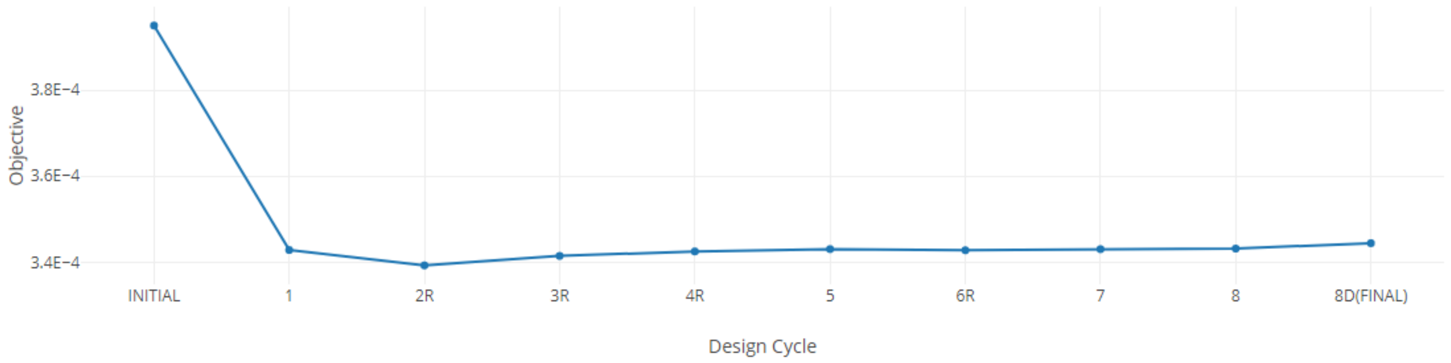
SOL 200 Web App - Local Optimization Results

Home

Final Message in .f06

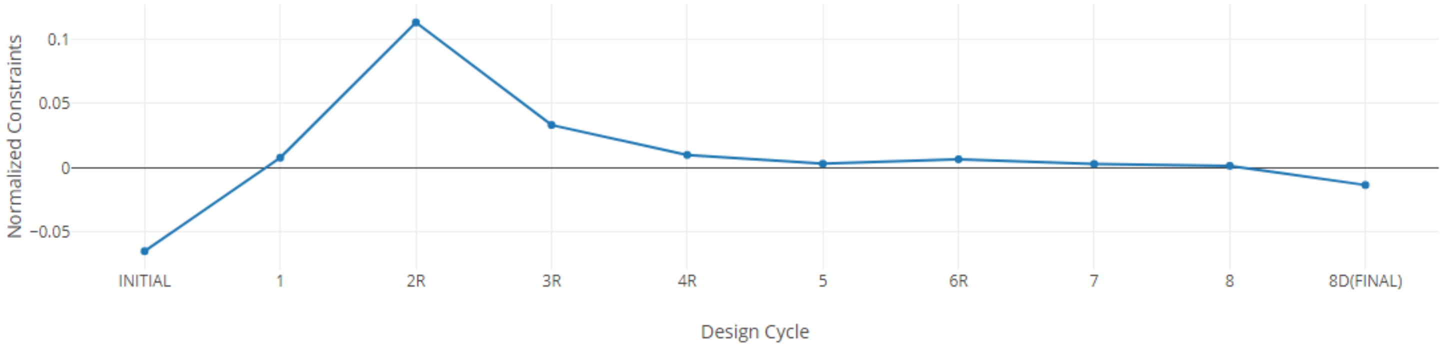
```
RUN TERMINATED DUE TO HARD CONVERGENCE TO AN OPTIMUM AT CYCLE NUMBER =        8.
AND HARD FEASIBLE DISCRETE DESIGN OBTAINED
```

Objective



Normalized Constraints

+ Info



Why Use the Trust Region?

The variable history also no longer shows the zig zag pattern.

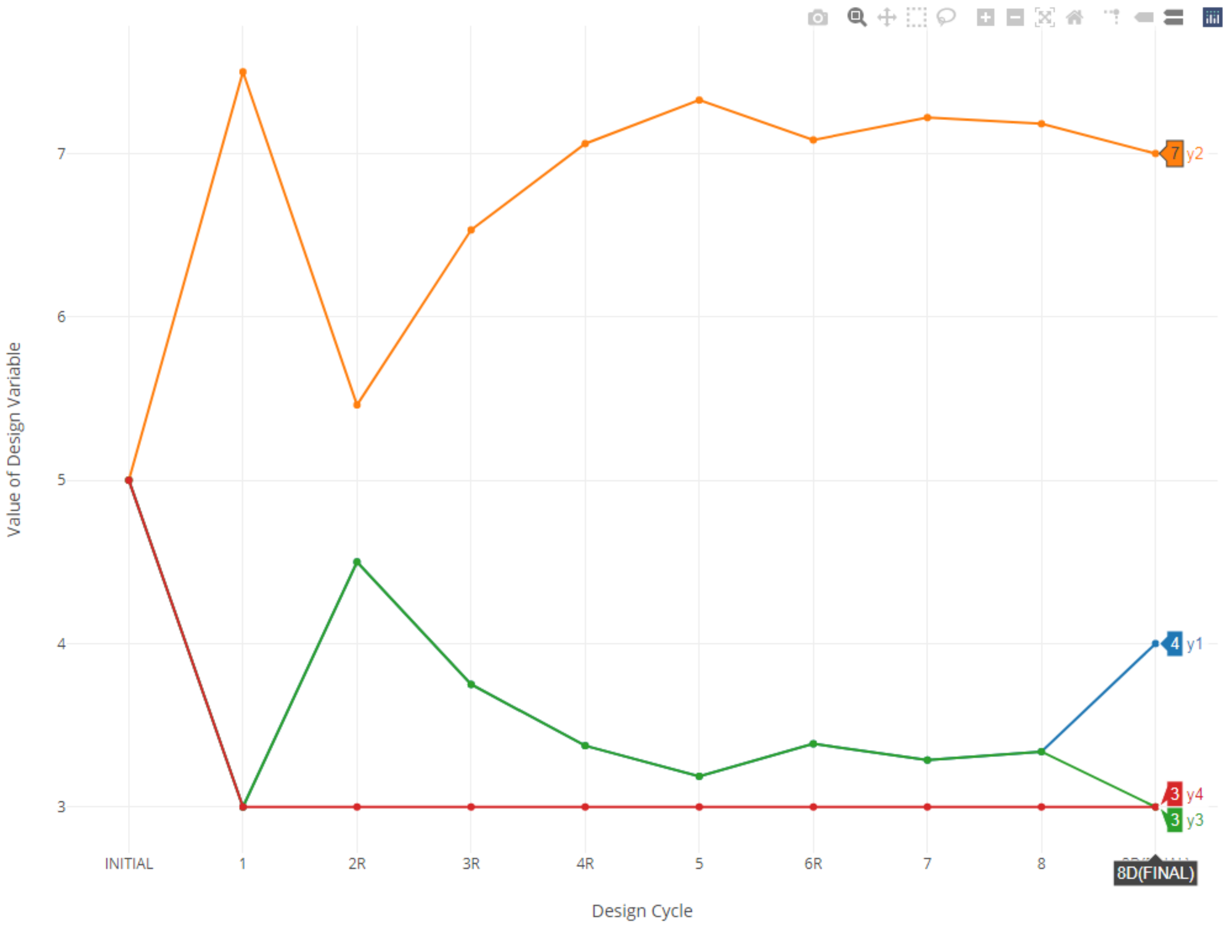
It is recommended that after a first optimization, the trust region should be used in a second optimization for the following reasons:

- The trust region might help achieve a feasible and converged solution
- The trust region might allow the optimizer to converge faster

The trust region setting is recommended only for size and topometry optimization.

SOL 200 Web App - Local Optimization Results

Design Variables



Reset Table

Display None | Display All

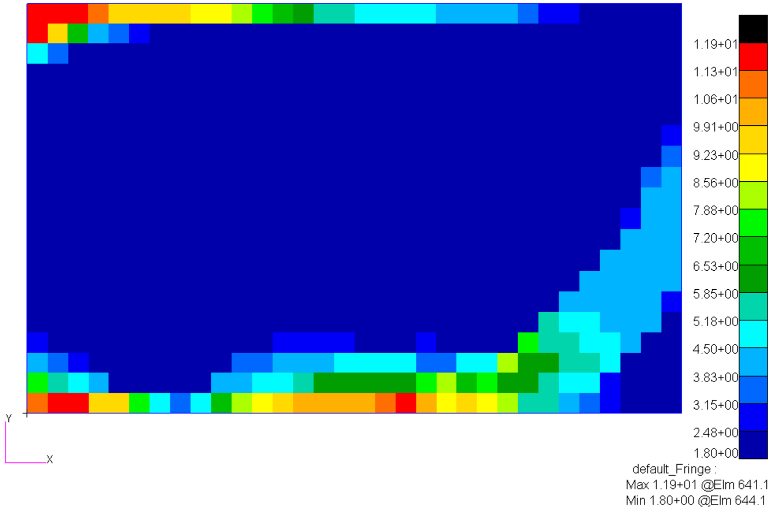
Display	Color	Label	Label Comments
		y	Search
☑		y1	Parent PCOMP 1 - Number of plies for 0°, GPLY IDs: 152000, 2152000
☑		y2	Parent PCOMP 1 - Number of plies for 0°, GPLY IDs: 153000, 2153000
☑		y3	Parent PCOMP 1 - Number of plies for 0°, GPLY IDs: 154000, 2154000
☑		y4	Parent PCOMP 1 - Number of plies for 0°, GPLY IDs: 155000, 2155000

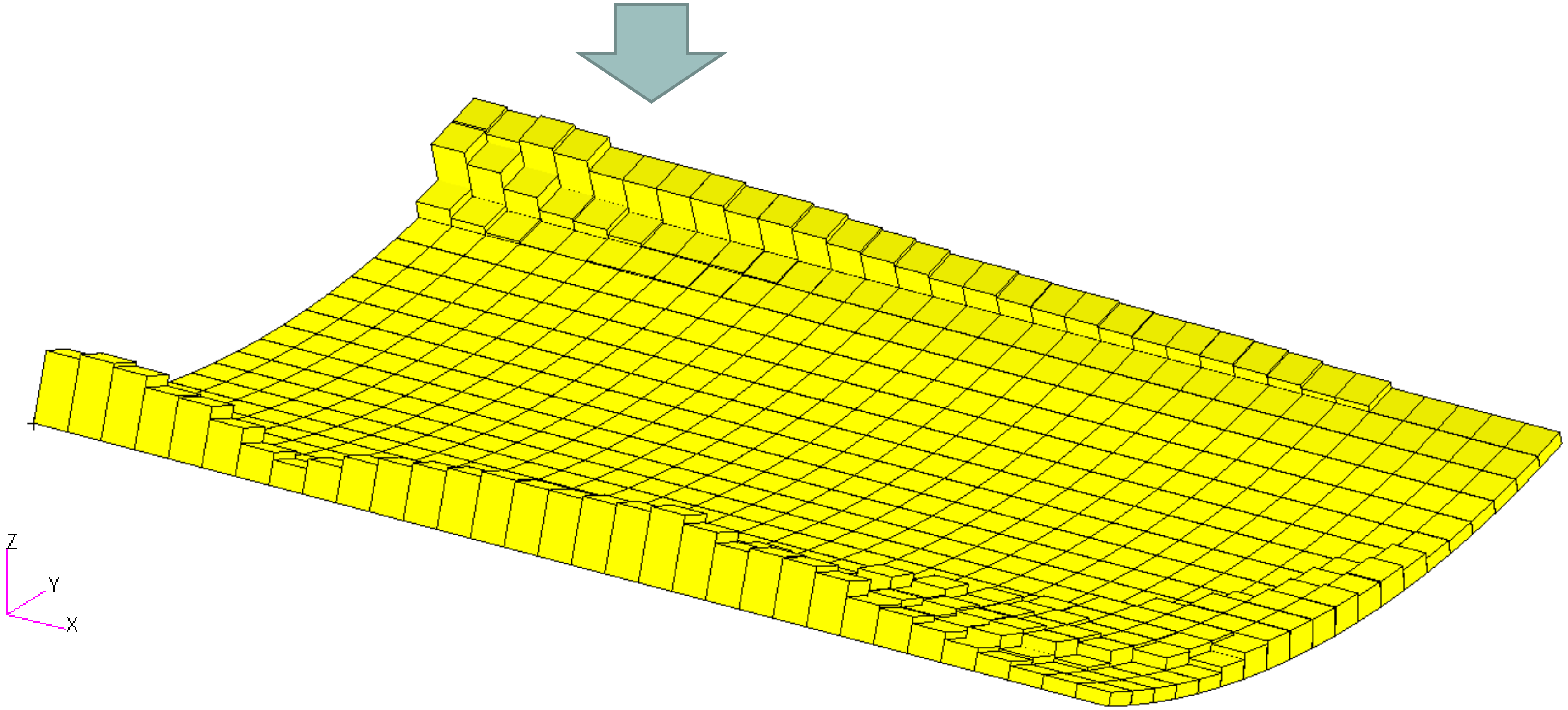
5 | 10 | 20 | 50 | 100 | 200

Using the Topometry Optimization Results as Is

Using the Topometry Optimization Results as Is

- The results of a topometry optimization correspond to a new element-by-element thickness of the composite.
- The original BDF file only included 1 PCOMP entry and was assigned to 640 CQUAD4 elements.
- During the topometry optimization, internally 640 PCOMP entries were created, and the core thickness in each of the 640 PCOMP entries was varied.
- It may be desired to use the topometry optimization results in the BDF file as is without a smoothing procedure. The following steps details how to use the topometry optimization results as is.





Update the Original Model

Before this, ensure the BDF files prior to optimization have one of these entries:

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

The original BDF files are updated to use the new thickness distributions after a topometry optimization.

1. After a topometry optimization, new PCOMP entries are output to the PCH file.
2. Also, the 2D element entries must be updated to use the new PCOMP entry IDs.

new_2D_elements.tmp

```
CQUAD4,641,10000001,725,726,798,724
CQUAD4,642,10000002,726,727,799,798
CQUAD4,643,10000003,727,728,800,799
CQUAD4,644,10000004,728,729,801,800
CQUAD4,645,10000005,729,730,802,801
CQUAD4,646,10000006,730,731,803,802
CQUAD4,647,10000007,731,732,804,803
CQUAD4,648,10000008,732,733,805,804
```

model_curved_panel_with_core.bdf

```
CQUAD4   641     1       725     726     798     724
CQUAD4   642     1       726     727     799     798
CQUAD4   643     1       727     728     800     799
CQUAD4   644     1       728     729     801     800
CQUAD4   645     1       729     730     802     801
CQUAD4   646     1       730     731     803     802
CQUAD4   647     1       731     732     804     803
CQUAD4   648     1       732     733     805     804
```

2

model.pch

```
$
$          *************************************************************
$          *                                                           *
$          *      CONTINUOUS  DESIGN  CYCLE  NUMBER  =        26       *
$          *                                                           *
$          *************************************************************
$
$
$          *************************************************************
$          *                                                           *
$          *      CONTINUOUS  DESIGN  CYCLE  NUMBER  =        26       *
$          *                                                           *
$          *************************************************************
$
$
$   UPDATED ANALYSIS MODEL DATA ENTRIES
$
PCOMP   10000001   0.0     0.0    90.        HILL    0.0     0.0             SYM
                 101 .125     90.         YES     101 .125     45.           YES
                 101 .125    -45.         YES     101 .125      0.0          YES
                 501 11.9324   0.0        YES
$ Spawned PSHELL, MAT2 entries from PCOMP    10000001
$   PSHELL*          10000001        110000001  2.48647373E+01        210000001*
$   *          1.00000000E+00                0  1.00000000E+00  0.00000000E+00*
$   *          0.00000000E+00  2.48647373E+01        410000001
$   MAT2*           110000001  1.18076869E+03  3.11226920E+02  6.17248552E-12*
$   *          1.18076869E+03  8.56375213E-11  4.44368711E+02  2.68516605E-10*
$   *          1.51168938E+01  1.51168938E+01 -1.90930257E-12  0.00000000E+00*
$   *          0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00*
$   *                       0
```

1

PCH

model_curved_panel_with_core.bdf

```
    SPCFORCES(PLOT,SORT1,REAL)=ALL
    STATSUB = 1
BEGIN BULK
param   grdpnt  0
MDLPRM  HDF5    2
PARAM   PRTMAXIM YES
EIGRL    1       0.0                     1       0
$ Elements and Element Properties for region : pcomp.1
$ Composite Property Reference Material: pcomp.502
$ Composite Material Description :
PCOMP    1       0.0                     90.     HILL                    SYM
         101     .125    90.     YES
         101     .125    45.     YES
         101     .125    -45.    YES
         101     .125    0.      YES
         501     5.      0.      YES
$ Pset: "pcomp.1" will be imported as: "pcomp.1"
CQUAD4   641     1       725     726     798     724     0.      0.
CQUAD4   642     1       726     727     799     798     0.      0.
CQUAD4   643     1       727     728     800     799     0.      0.
CQUAD4   644     1       728     729     801     800     0.      0.
CQUAD4   645     1       729     730     802     801     0.      0.
CQUAD4   646     1       730     731     803     802     0.      0.
CQUAD4   647     1       731     732     804     803     0.      0.
CQUAD4   648     1       732     733     805     804     0.      0.
CQUAD4   649     1       733     734     806     805     0.      0.
CQUAD4   650     1       734     735     807     806     0.      0.
CQUAD4   651     1       735     736     808     807     0.      0.
CQUAD4   652     1       736     737     809     808     0.      0.
CQUAD4   653     1       737     738     810     809     0.      0.
CQUAD4   654     1       738     739     811     810     0.      0.
CQUAD4   655     1       739     740     812     811     0.      0.
```

Downloaded BDF/DAT File

Update the Original Model

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

```python
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/apricot/Downloads/trsh/model.h5'
    path_b = '/home/apricot/Downloads/trsh/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
```


HEXAGON Technology Partner

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

```
CQUAD4,641,10000001,725,726,798,724
CQUAD4,642,10000002,726,727,799,798
CQUAD4,643,10000003,727,728,800,799
CQUAD4,644,10000004,728,729,801,800
CQUAD4,645,10000005,729,730,802,801
CQUAD4,646,10000006,730,731,803,802
CQUAD4,647,10000007,731,732,804,803
CQUAD4,648,10000008,732,733,805,804
CQUAD4,649,10000009,733,734,806,805
CQUAD4,650,10000010,734,735,807,806
CQUAD4,651,10000011,735,736,808,807
CQUAD4,652,10000012,736,737,809,808
CQUAD4,653,10000013,737,738,810,809
CQUAD4,654,10000014,738,739,811,810
CQUAD4,655,10000015,739,740,812,811
CQUAD4,656,10000016,740,741,813,812
CQUAD4,657,10000017,741,742,814,813
CQUAD4,658,10000018,742,743,815,814
CQUAD4,659,10000019,743,744,816,815
CQUAD4,660,10000020,744,745,746,816
CQUAD4,661,10000021,724,798,817,723
CQUAD4,662,10000022,798,799,818,817
CQUAD4,663,10000023,799,800,819,818
CQUAD4,664,10000024,800,801,820,819
CQUAD4,665,10000025,801,802,821,820
CQUAD4,666,10000026,802,803,822,821
```

Original BDF File

model_curved_panel_with_core.bdf

```
$ Pset: "pcomp.1" will be imported as: "pcomp.1"
CQUAD4,641,10000001,725,726,798,724
CQUAD4,642,10000002,726,727,799,798
CQUAD4,643,10000003,727,728,800,799
CQUAD4,644,10000004,728,729,801,800
CQUAD4,645,10000005,729,730,802,801
CQUAD4,646,10000006,730,731,803,802
CQUAD4,647,10000007,731,732,804,803
CQUAD4,648,10000008,732,733,805,804
CQUAD4,649,10000009,733,734,806,805
CQUAD4,650,10000010,734,735,807,806
CQUAD4,651,10000011,735,736,808,807
CQUAD4,652,10000012,736,737,809,808
CQUAD4,653,10000013,737,738,810,809
CQUAD4,654,10000014,738,739,811,810
CQUAD4,655,10000015,739,740,812,811
CQUAD4,656,10000016,740,741,813,812
CQUAD4,657,10000017,741,742,814,813
CQUAD4,658,10000018,742,743,815,814
CQUAD4,659,10000019,743,744,816,815
CQUAD4,660,10000020,744,745,746,816
CQUAD4,661,10000021,724,798,817,723
CQUAD4,662,10000022,798,799,818,817
CQUAD4,663,10000023,799,800,819,818
CQUAD4,664,10000024,800,801,820,819
CQUAD4,665,10000025,801,802,821,820
CQUAD4,666,10000026,802,803,822,821
CQUAD4,667,10000027,803,804,823,822
```

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     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501 11.9324     0.0             YES
PCOMP   10000002     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501 11.9324     0.0             YES
PCOMP   10000003     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501 4.91334     0.0             YES
PCOMP   10000004     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000005     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000006     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000007     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000008     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000009     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000010     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
```

Original BDF File

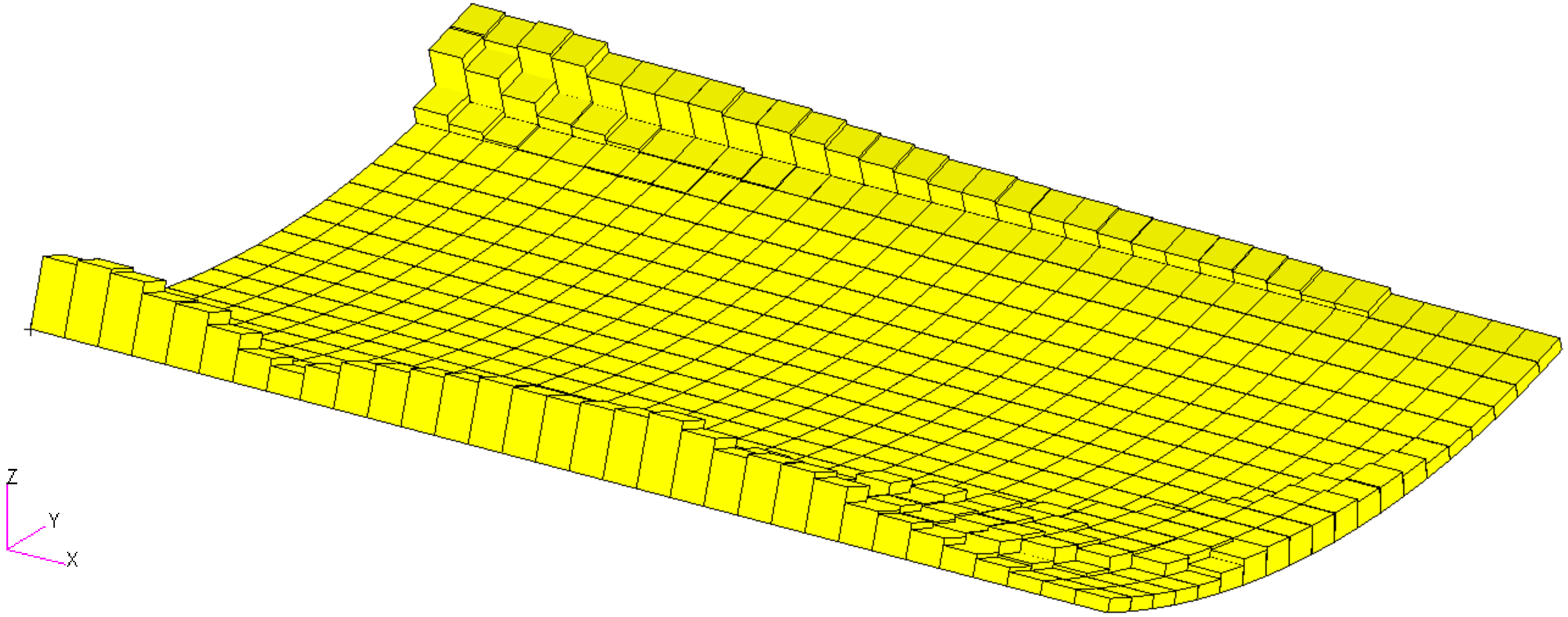
model_curved_panel_with_core.bdf

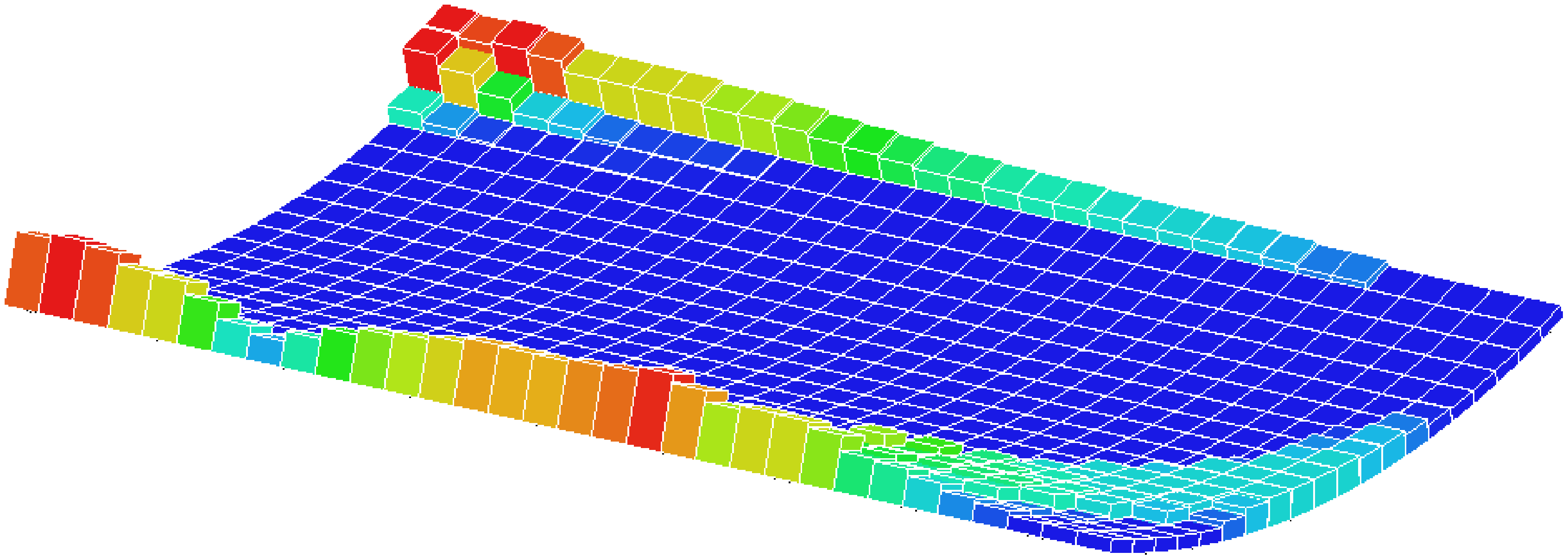
```
$ Composite Material Description :
PCOMP   10000001     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501 11.9324     0.0             YES
PCOMP   10000002     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501 11.9324     0.0             YES
PCOMP   10000003     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501 4.91334     0.0             YES
PCOMP   10000004     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000005     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000006     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000007     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000008     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
PCOMP   10000009     0.0     0.0     90.             HILL     0.0     0.0             SYM
             101    .125     90.             YES      101    .125     45.             YES
             101    .125    -45.             YES      101    .125     0.0             YES
             501  1.8016     0.0             YES
```

1

Confirm the Composite Thicknesses

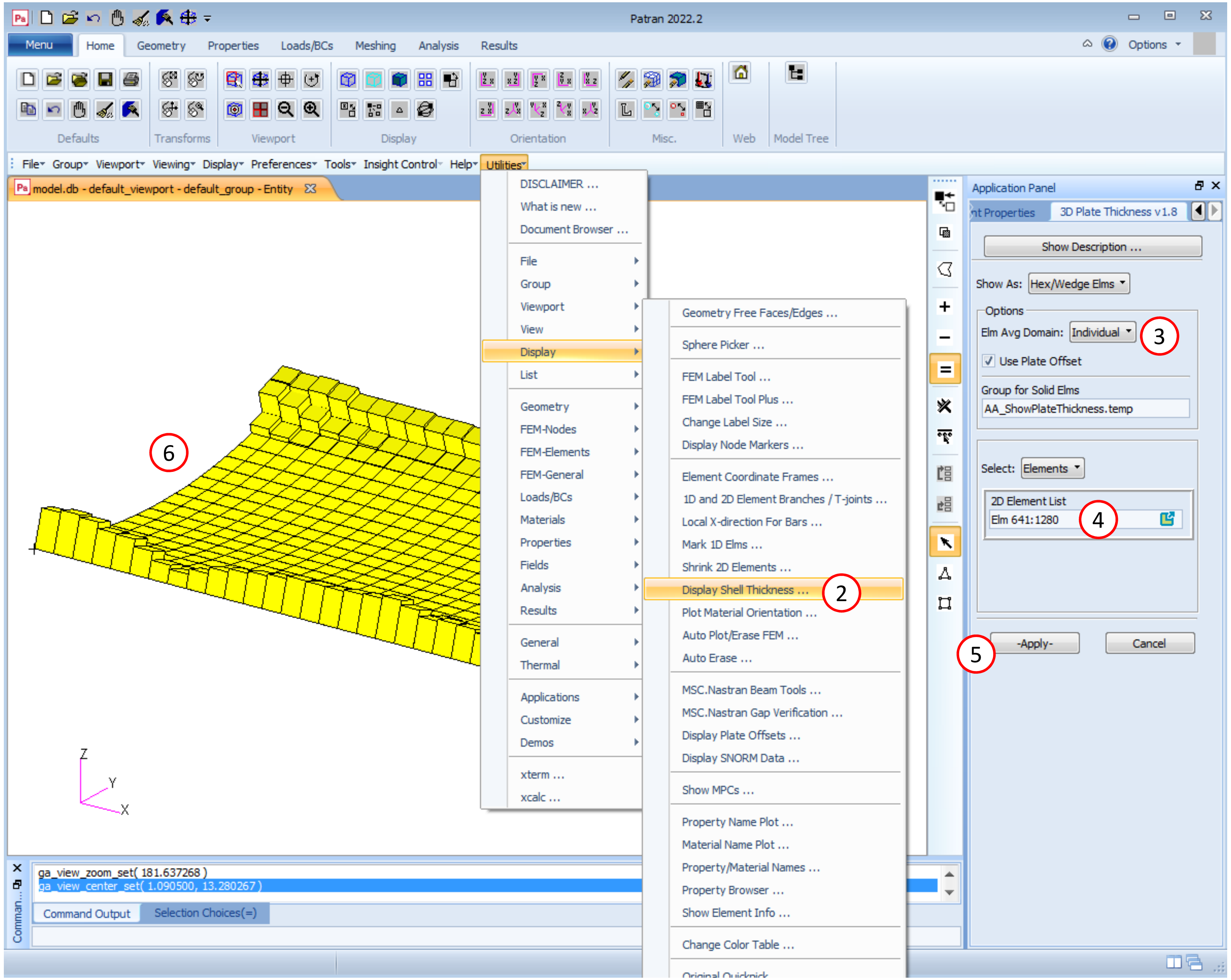
1. Patran or MSC Apex may be used to confirm the total thickness of the composite.





Composite Thickness - Patran

1. Import the BDF file (not shown)
2. Go to Utilities > Display > Display Shell Thickness …
3. Set Elm Avg Domain: Individual
4. Select all the 2D element visible in the viewport
5. Click Apply
6. The total composite thickness is now displayed



Composite Thickness – MSC Apex

1. Import the BDF file (not shown)
2. Click the arrow next to the Shell Thickness button
3. Click 3D Thickness and Offset. The button should become blue in color, indicating the option is on.
4. Click Shell Thickness to display the total composite thickness. The button should become blue in color, indicating the option is on.
5. The total composite thickness is now displayed