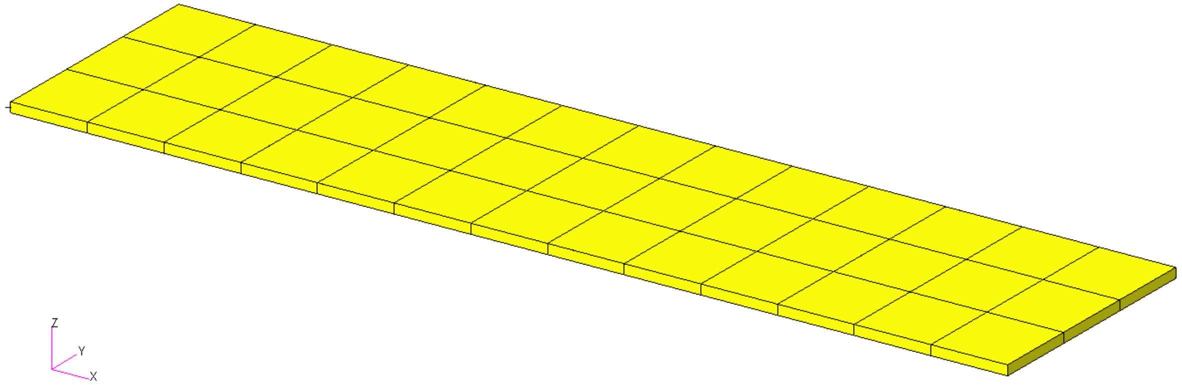
Workshop - MSC Nastran Topometry Optimization of a Cantilever Plate

AN MSC NASTRAN SOL 200 TUTORIAL

Goal: Use Nastran SOL 200 Optimization

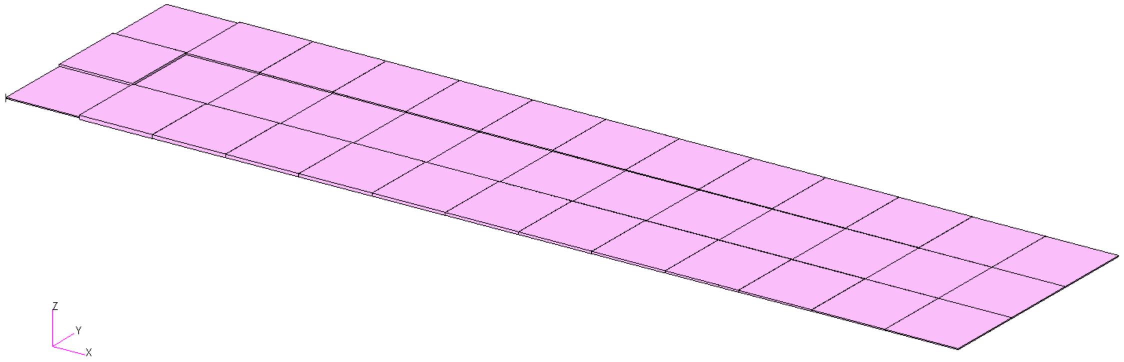
Before Optimization

- Mass: 19.5 kg



After Optimization

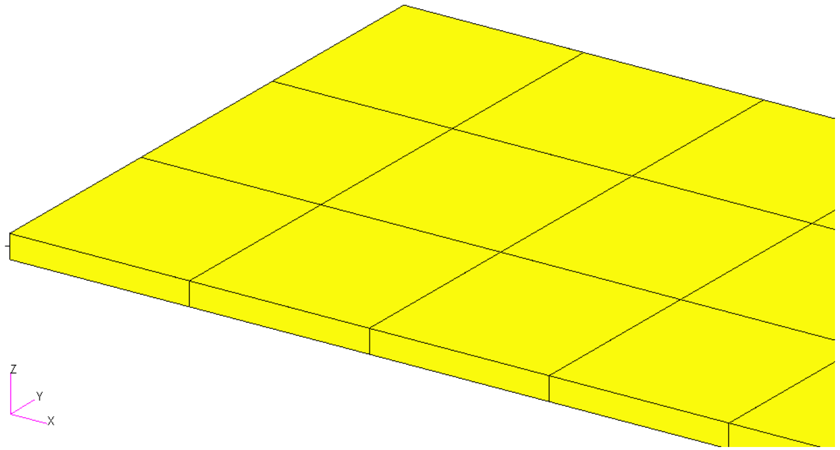
- Mass: 3.97 kg
- Vary the thickness of each element



Goal: Use Nastran SOL 200 Optimization

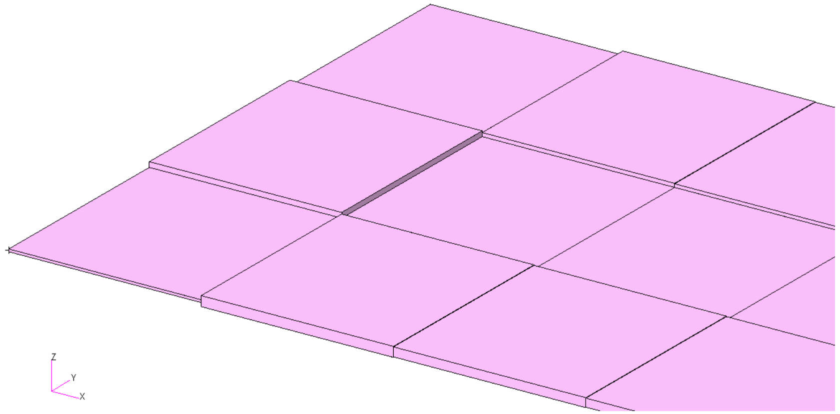
Before Optimization

- Mass: 19.5 kg

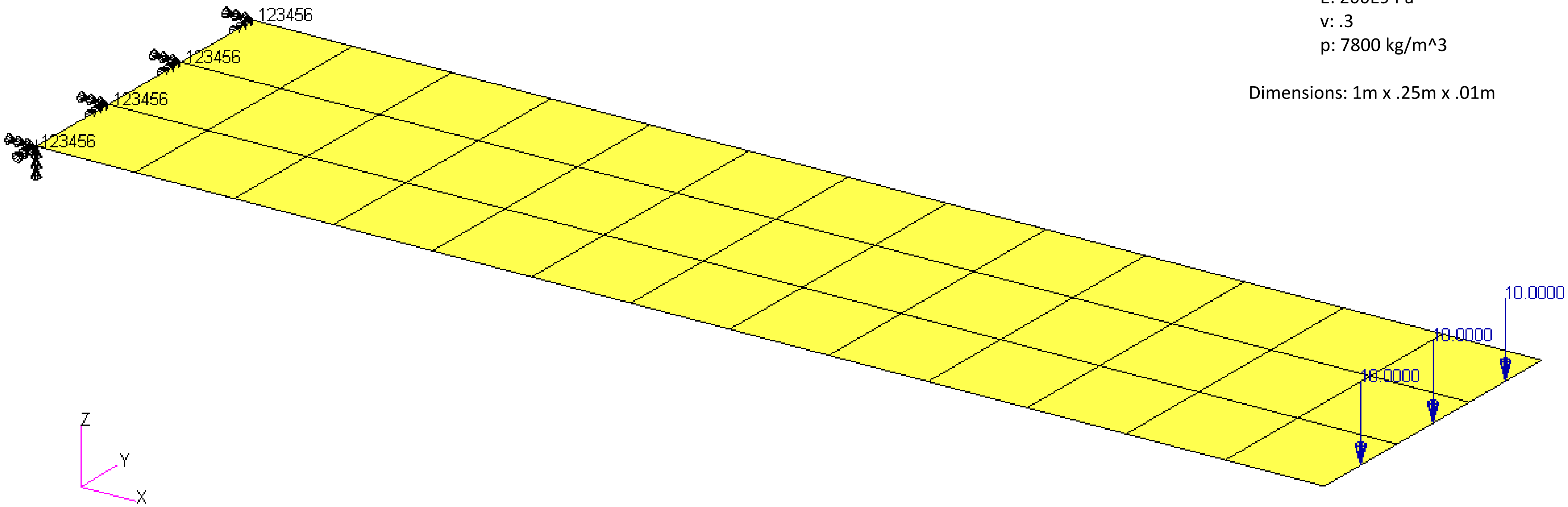


After Optimization

- Mass: 3.97 kg
- Vary the thickness of each element



Details of the structural model



Units: m, N, MPa

Material:

E: 200E9 Pa

v: .3

p: 7800 kg/m^3

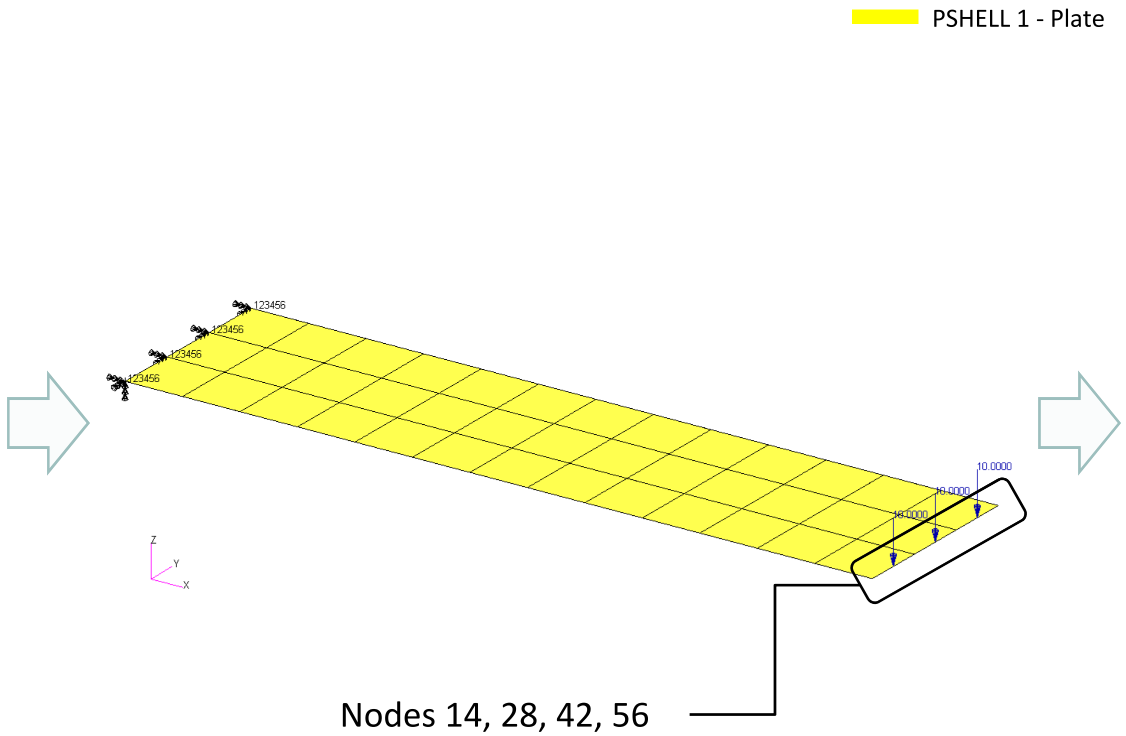
Dimensions: 1m x .25m x .01m

Optimization Problem Statement

Design Region/Variables

z1: Thickness (T) of PSHELL 1

.001 < z1



Design Objective

r0: Minimize weight

Design Constraints

r1: von Mises stress of PSHELL 1

r1 < 250E6

r2: Z Displacement of nodes 14, 28, 42, 56 (GRID IDs: 14, 28, 42, 56)

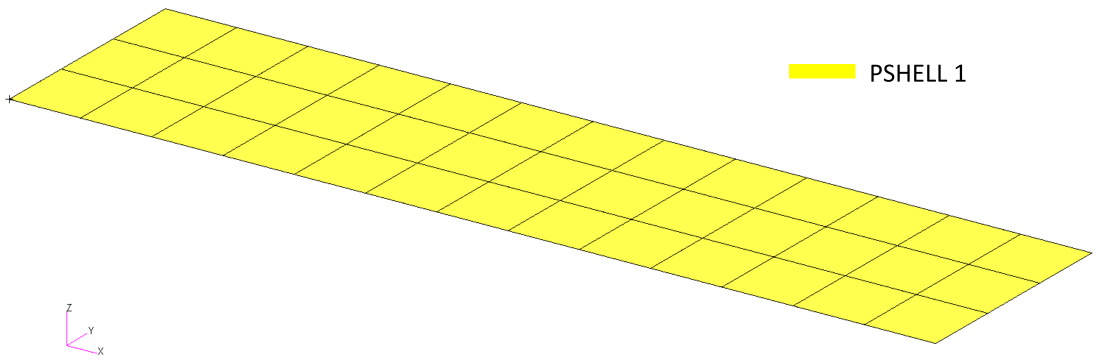
-.01 < r2

Difference Between Size and Topometry Optimization

Size Optimization

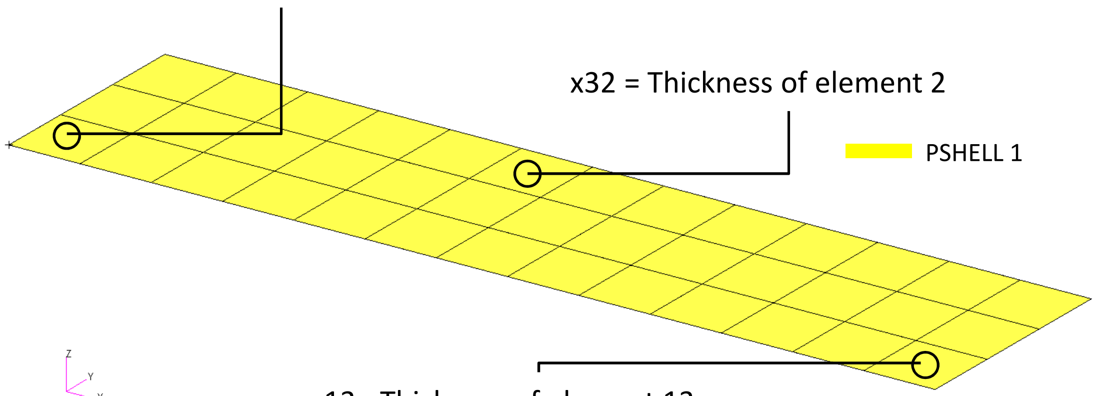
1. Select the parameter to optimize
2. One design variable (x1) is created and applies to all the elements

x1 = Thickness of every element related to PSHELL 1



Topometry Optimization

1. Select the parameter and design region
2. In the background, one design variable is automatically created for each element => Element-by-element optimization



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

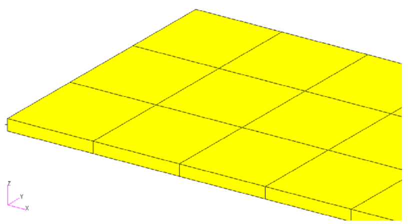
Tutorial Overview

1. Start with a .bdf or .dat file
2. Use the SOL 200 Web App to:
   - Convert the .bdf file to SOL 200
     - Design 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

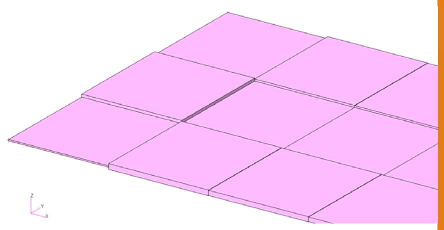
Topometry Optimization - This type of optimization assigns a design variable for each individual element.

Before



After

Topometry Optimization (Element-by-element size optimization)



SOL 200 Web App Capabilities

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

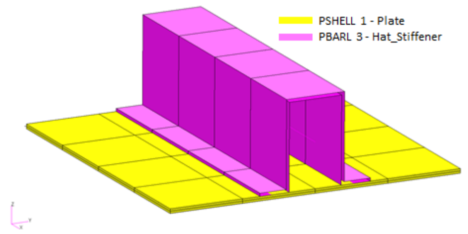
Compatibility

- 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.

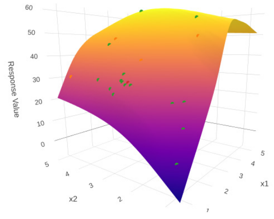
Benefits

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

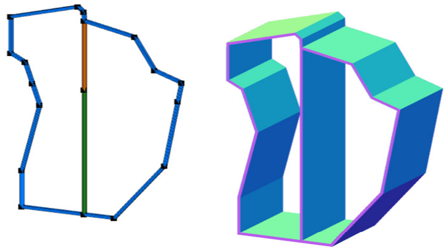
Web Apps



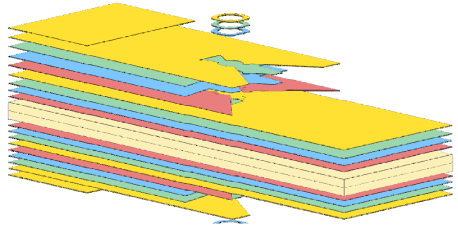
Web Apps for MSC Nastran SOL 200
Pre/post for MSC Nastran SOL 200. Support for size, topology, topometry, topography, multi-model optimization.



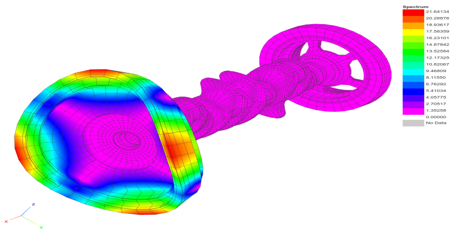
Machine Learning Web App
Bayesian Optimization for nonlinear response optimization (SOL 400)



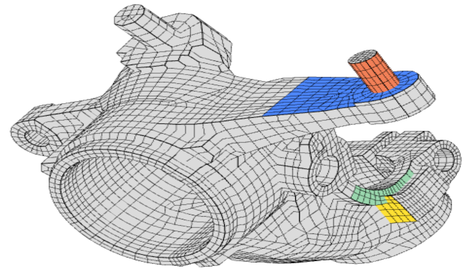
PBMSECT Web App
Generate PBMSECT and PBRSECT entries graphically



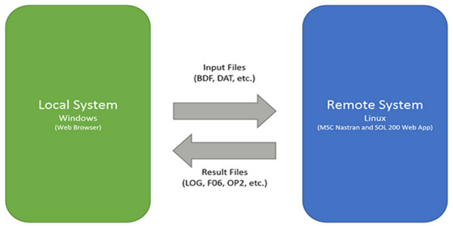
Ply Shape Optimization Web App
Optimize composite ply drop-off locations, and generate new PCOMPG entries



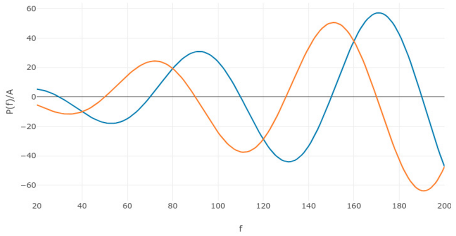
Post-processor Web App
View MSC Nastran results in a web browser on Windows and Linux



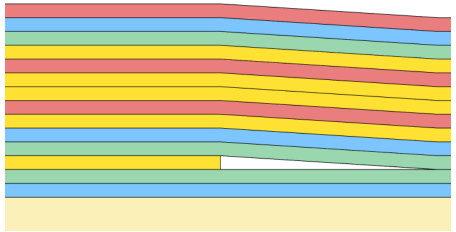
Shape Optimization Web App
Use a web application to configure and perform shape optimization.



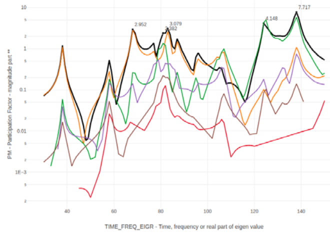
Remote Execution Web App
Run MSC Nastran jobs on remote Linux or Windows systems available on the local network



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



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

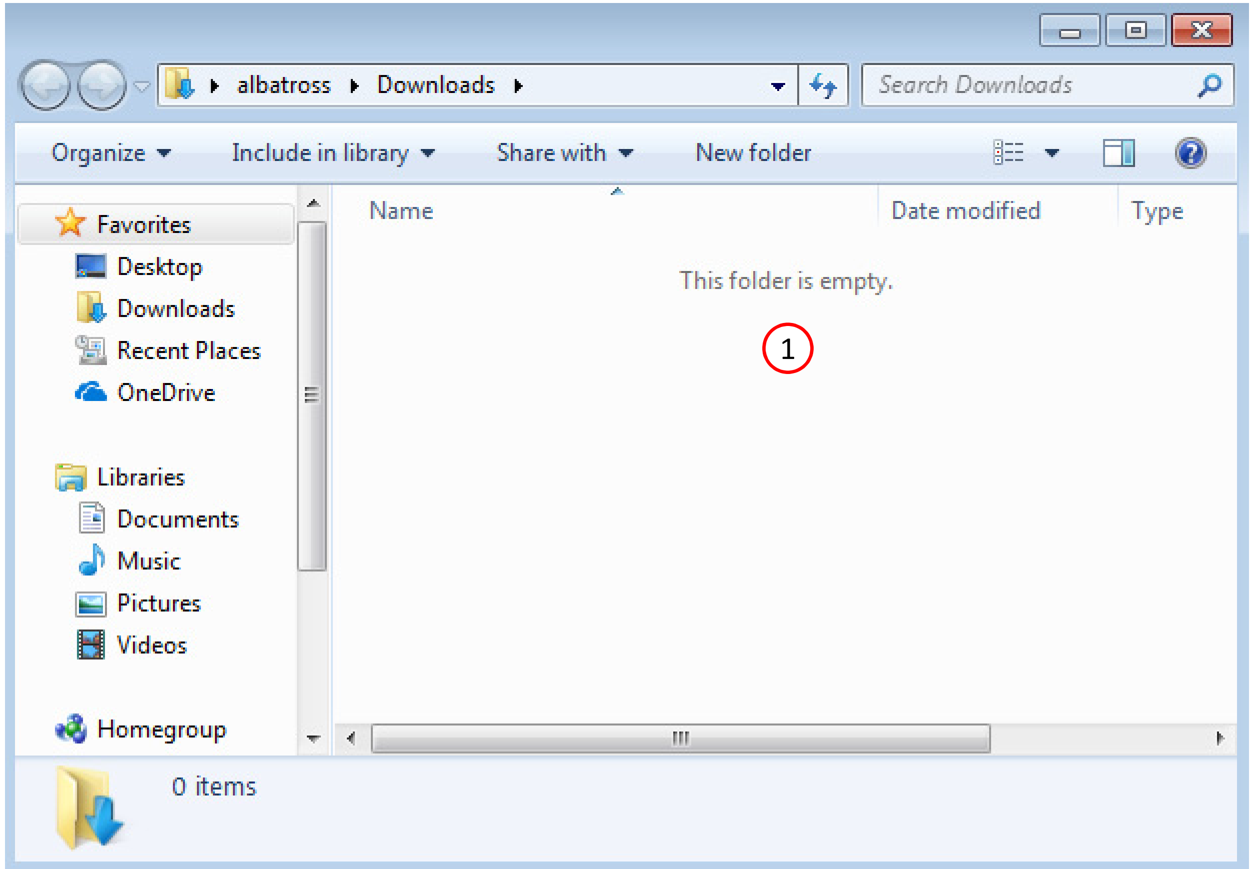


HDF5 Explorer Web App
Create graphs (XY plots) using data from the H5 file

Before Starting

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

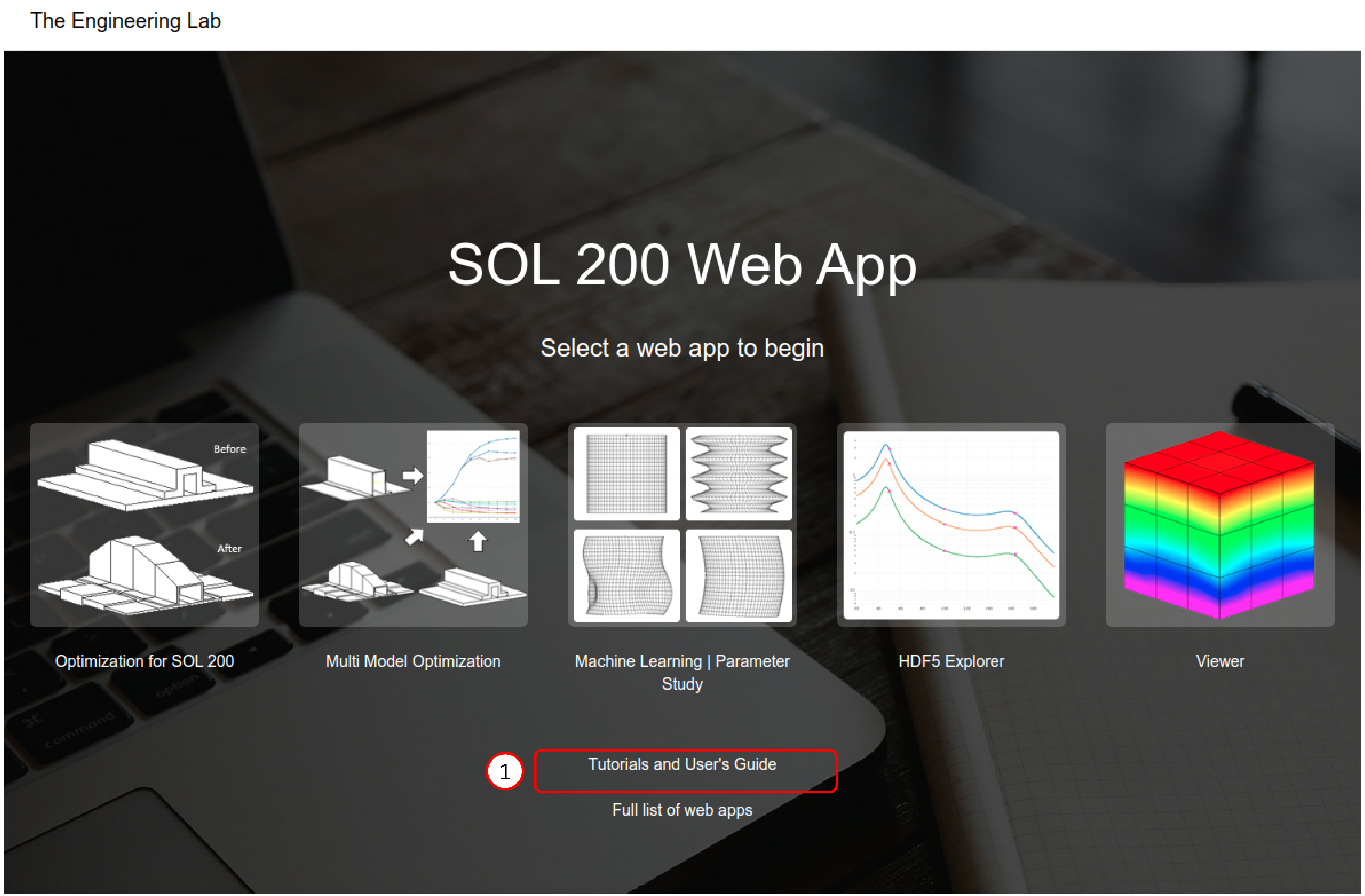
- Throughout this workshop, you will be working with multiple file types and directories such as:
    - .bdf/.dat
    - nastran_working_directory
    - .f06, .log, .pch, .h5, etc.
- To minimize confusion with files and folders, it is encouraged to start with a clean directory.



Go to the User's Guide

1. Click on the indicated link

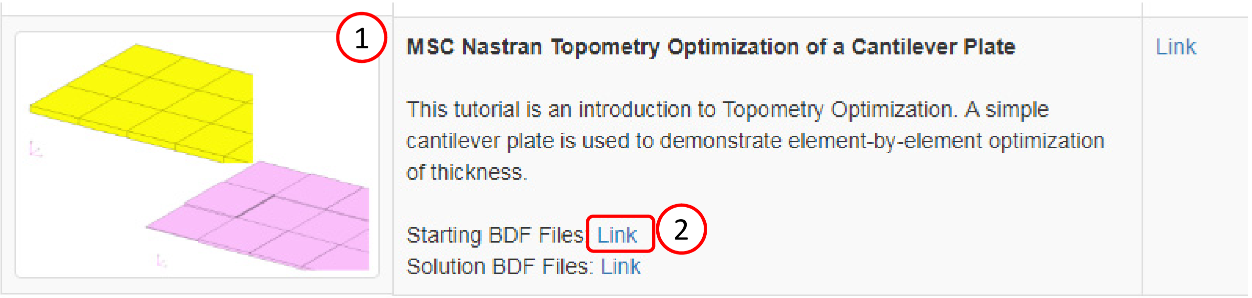
- The necessary BDF files for this tutorial are available in the Tutorials section of the User's Guide.

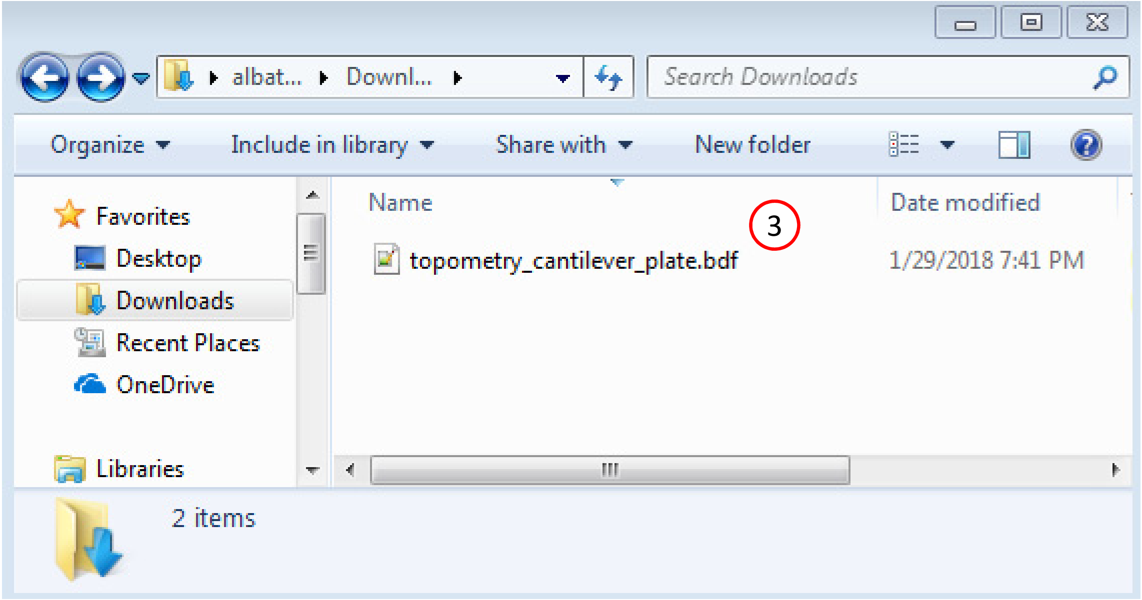


Obtain Starting Files

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.





Open the Correct Page

1. Click on the indicated link

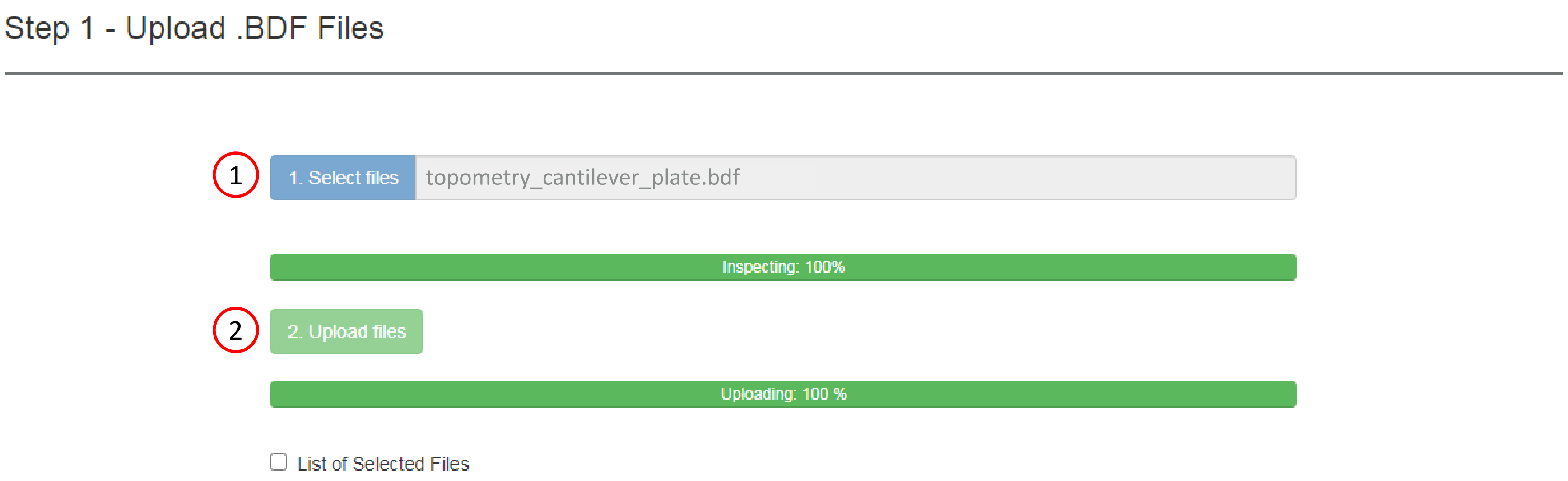
- MSC Nastran can perform many optimization types. The SOL 200 Web App includes dedicated web apps for the following:
    - Optimization for SOL 200 (Size, Topology, Topometry, Topography, Local Optimization, Sensitivity Analysis and Global Optimization)
    - Multi Model Optimization
    - Machine Learning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

The Engineering Lab

Upload BDF Files

1. Click 1. Select Files and select topometry_cantilever_plate.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.



Create Design Region

1. Click Topometry
2. Click on the plus (+) icon to set the thickness (T) of PSHELL 1 as a Design Region
3. 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 (1)   Topography

Step 1 - Select design properties

+ Options

Create TOMVAR	Property	Property Description	Entry	Entry ID	Current Value
	Search	Search	Search	Search	Search
(2) +	T	Thickness	PSHELL	1	.01
+	E	Young's modulus	MAT1	1	2.+11
+	NU	Poisson's ratio	MAT1	1	.3
+	RHO	Mass density	MAT1	1	7800.

5 10 20 30 40 50

Number of Visible Rows 5

Step 2 - Adjust TOMVAR Entries

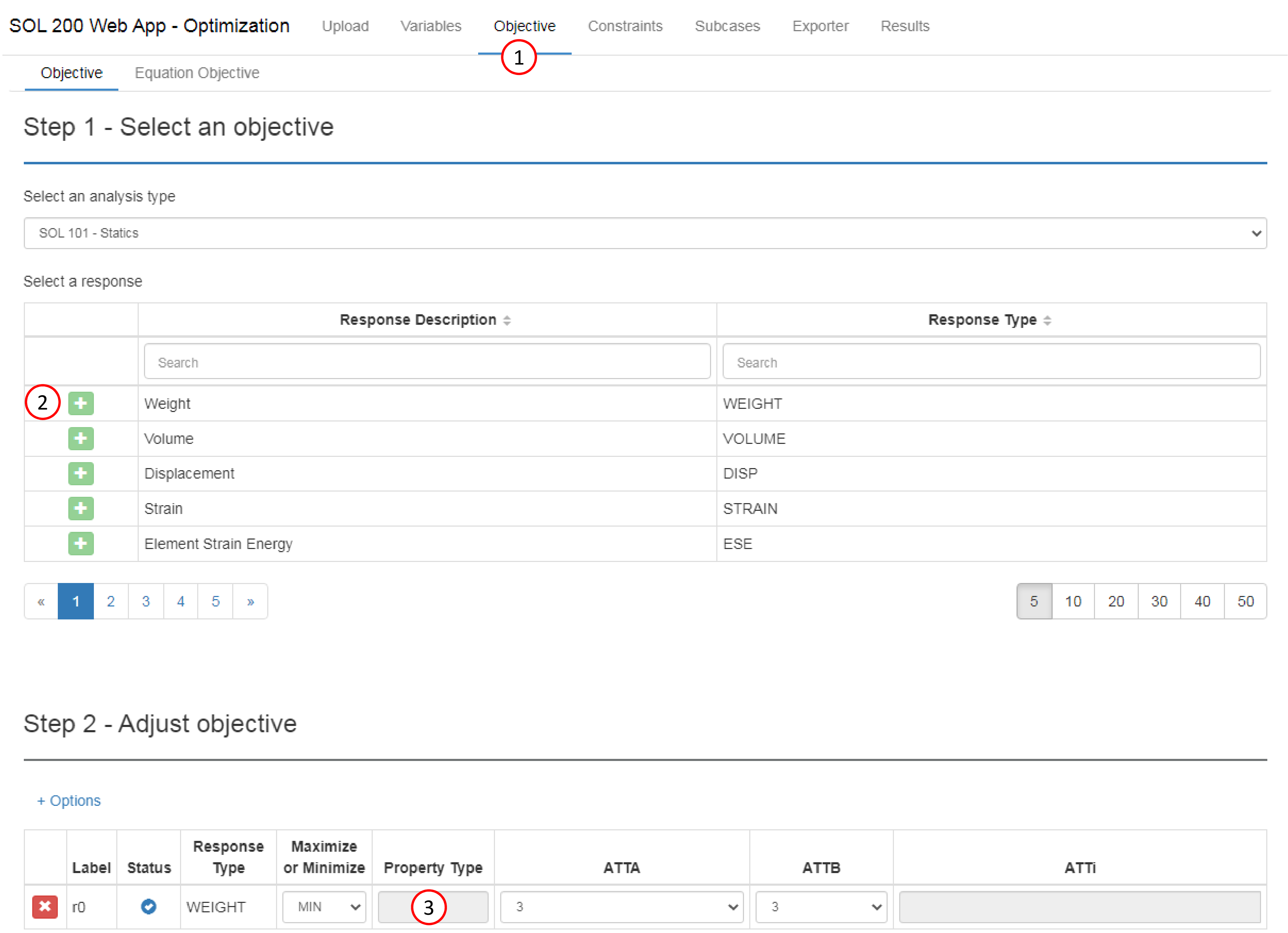
+ Options   ✖ Delete Visible Rows

	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 (3)	✔	T	Thickness	PSHELL	1	.01	.001	Upper	Examples: -2.0, 1.0, THRU, 10.0, B'

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

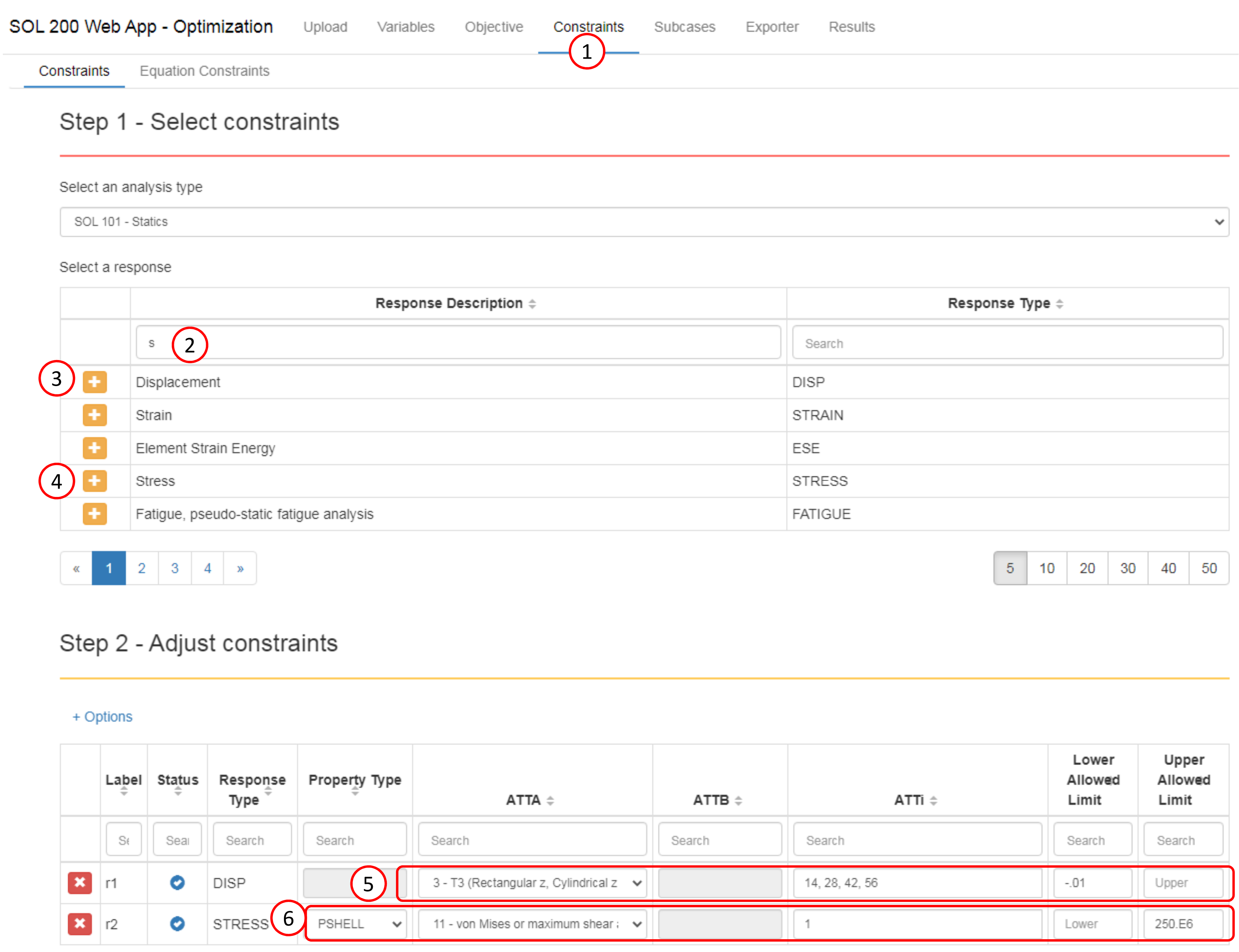
- The objective must always be a single scalar response. A response such as weight and volume are single responses and can be used as an objective. Other responses require special care when set as an objective. For example, if the objective is stress, only the stress of a single component, e.g. von Mises,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



Create Design Constraints

1. Click Constraints
2. Type ‘s’ into the search bar
3. Click on the plus (+) icon for Displacement
4. Click on the plus (+) icon for Stress
5. Configure the following for r1:
   - ATTA: 3 - T3 (Z component)
   - ATTi: 14, 28, 42, 56 (Nodes 14, 28, 42 and 56)
   - Lower Allowed Limit: -.01
6. Configure the following for r2:
   - Property Type: PSHELL
   - ATTA: 11 - von Mises
   - ATTi: 1 (PSHELL 1)
   - Upper Allowed Limit: 250.E6

- 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 | Match | Other

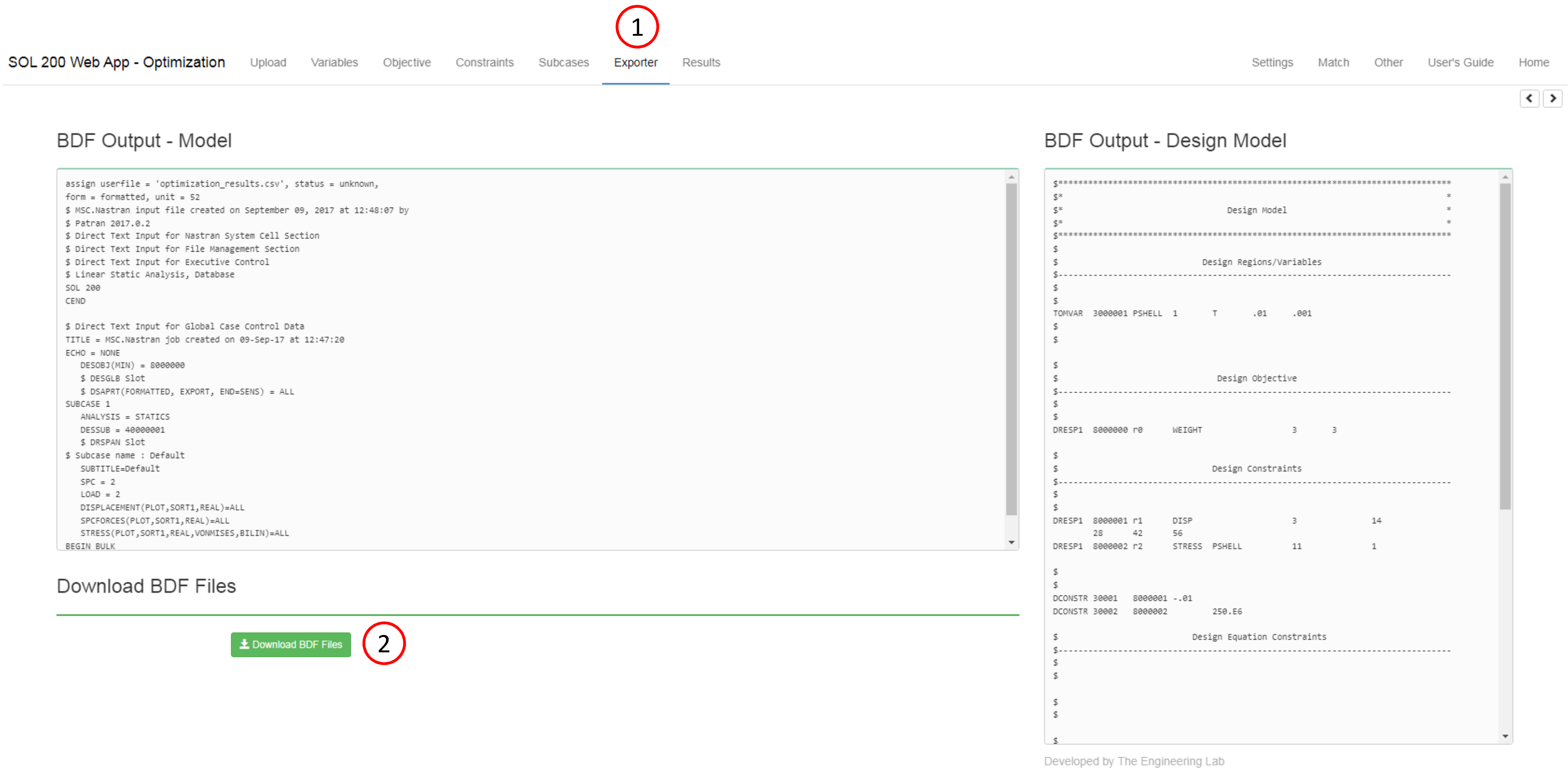
1

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	☑ 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

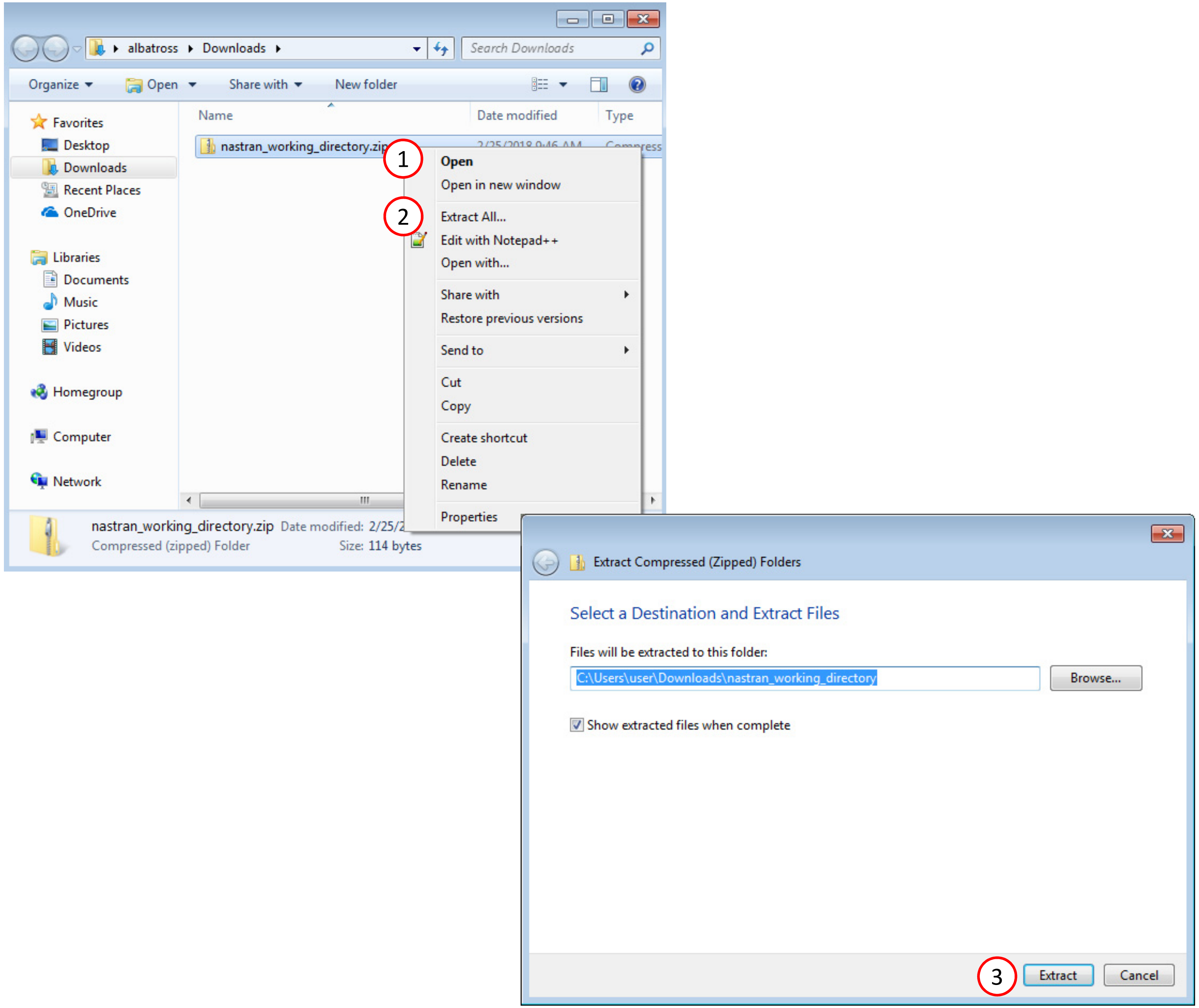


- When the download button is clicked a new file named “nastran_working_directory” is downloaded. If the file already exists in your local folder, the folder name is appended with a number, e.g. “nastran_working_directory (1).zip”

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

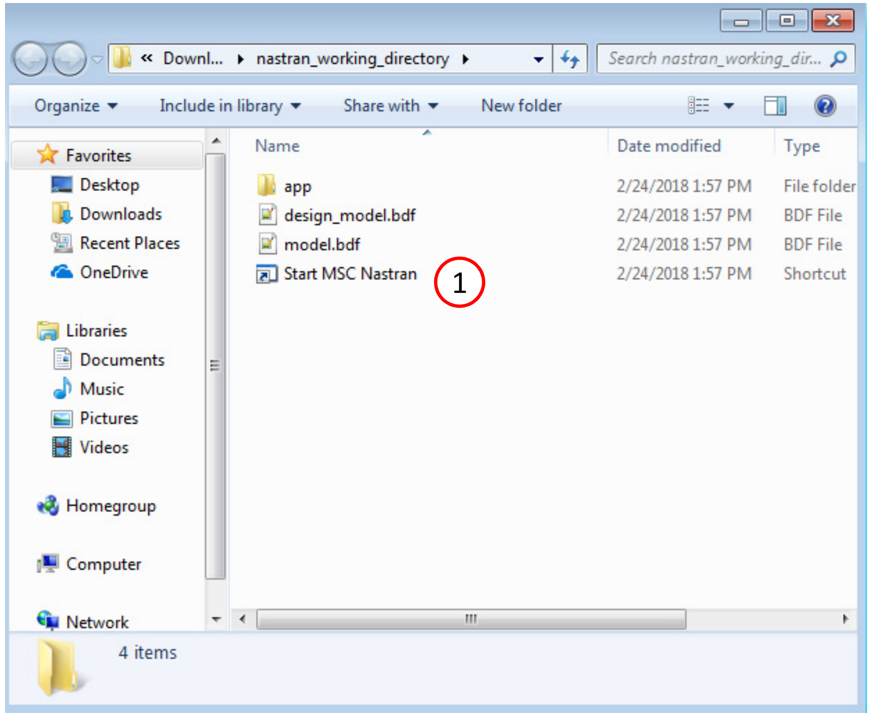
- After a successful optimization, the results will be automatically displayed as long as the following files are present: BDF, F06 and LOG.
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the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

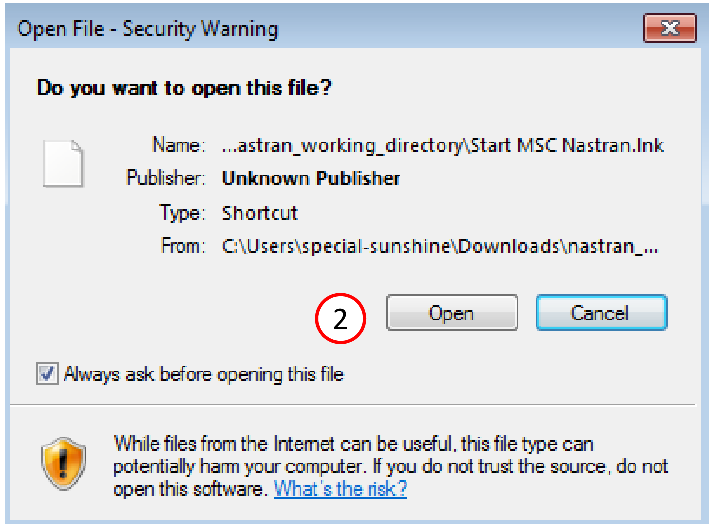
Using Linux?

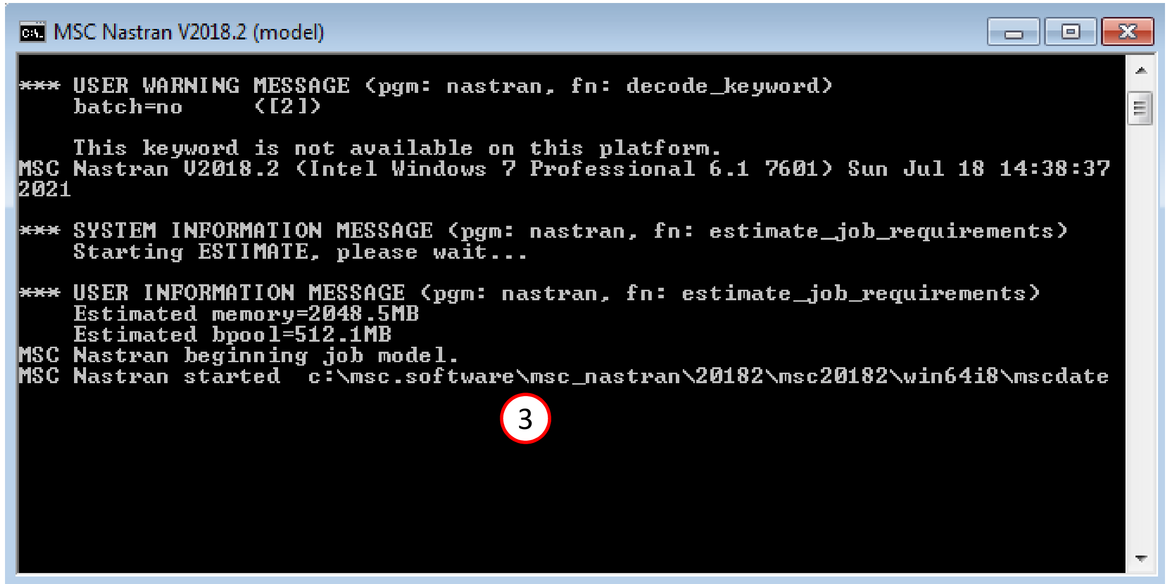
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

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

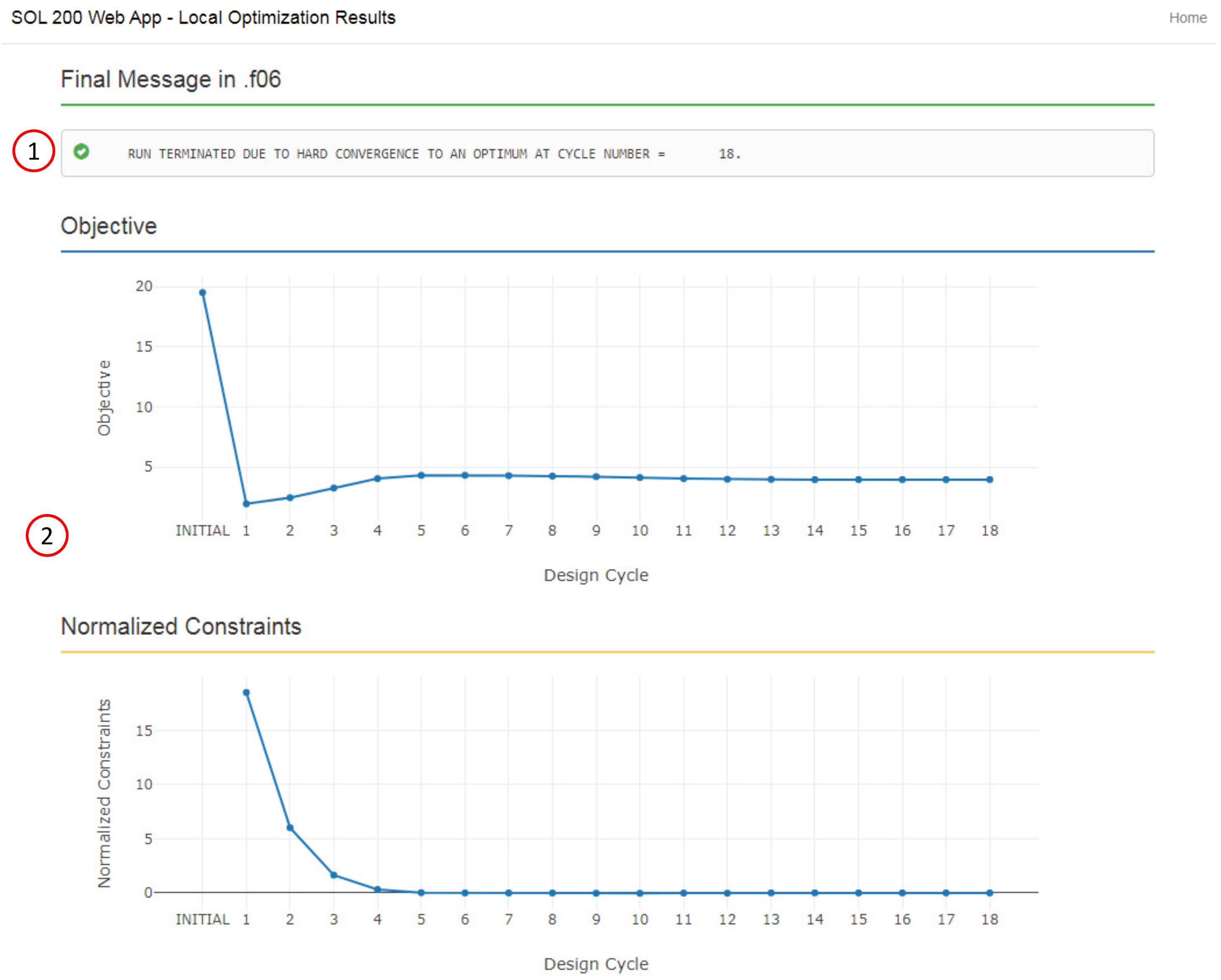
Name	Status of Job	Design Cycle	RUN TERMINATED DUE TO
model.bdf	Running	None	

Review Optimization Results

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

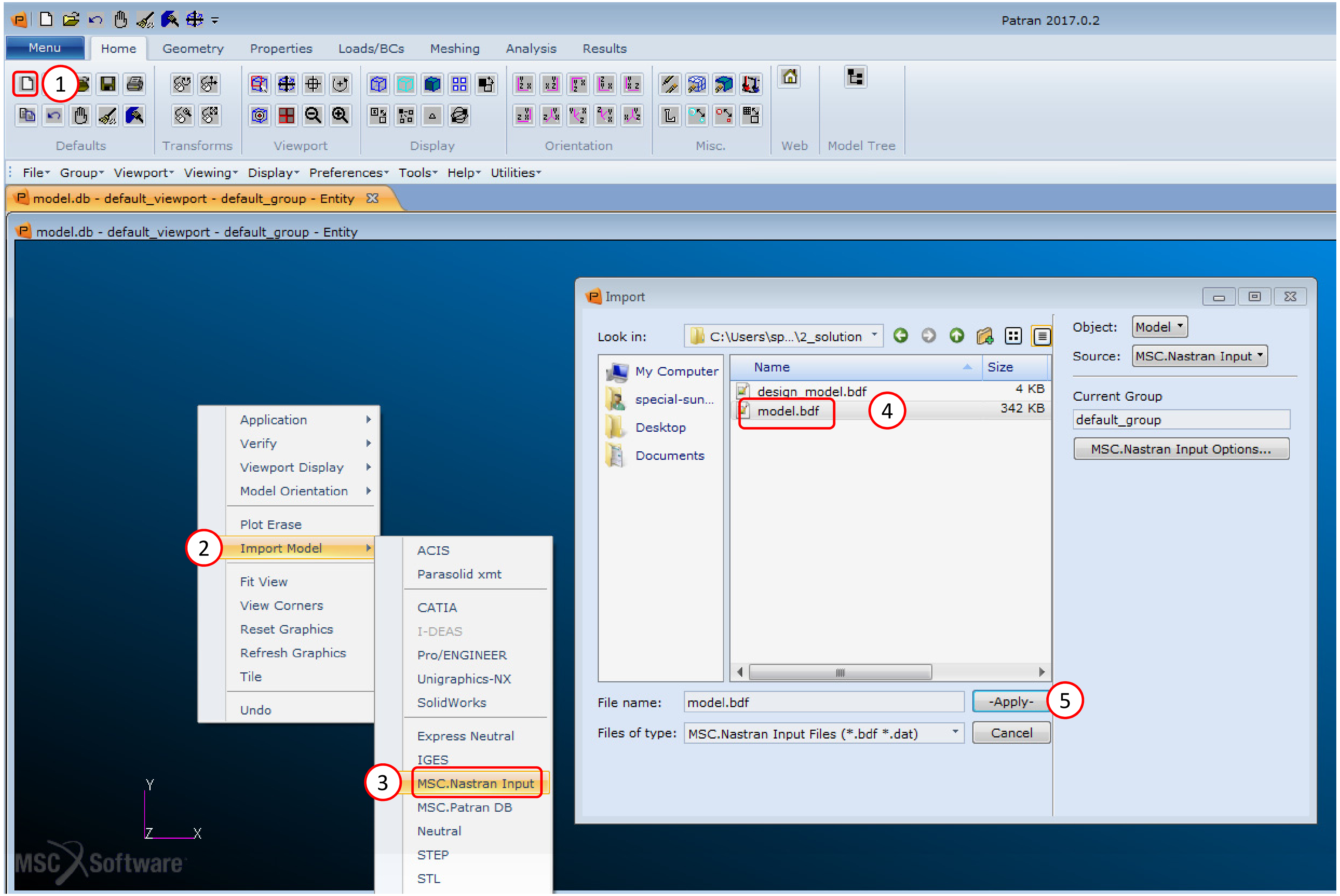
1. Ensure the messages shown have green checkmarks. This is indication of success. Any red icons indicate challenges.
2. The final value of objective and normalized constraints can be reviewed.

- 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.



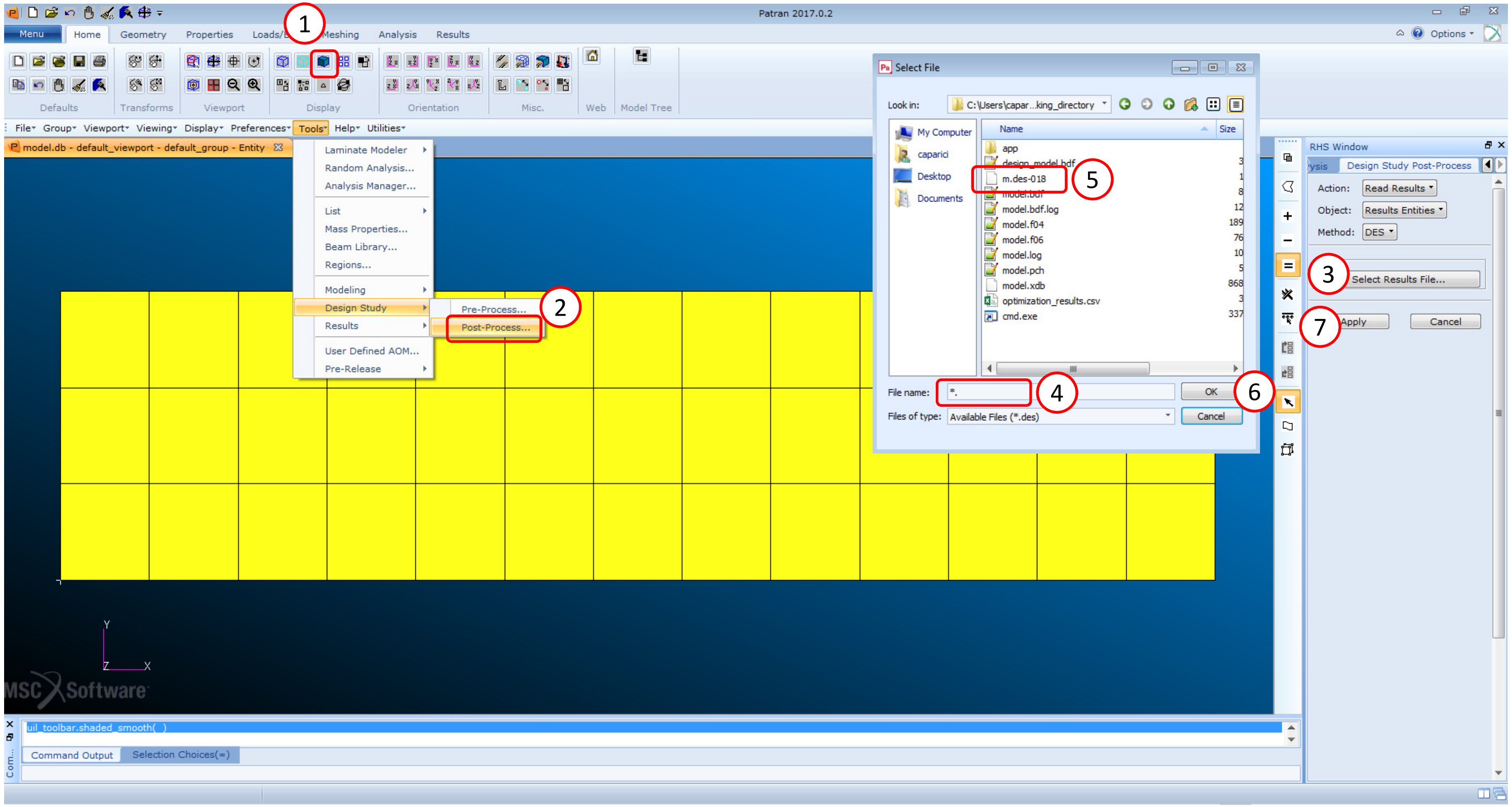
Review Optimization Results in Patran

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



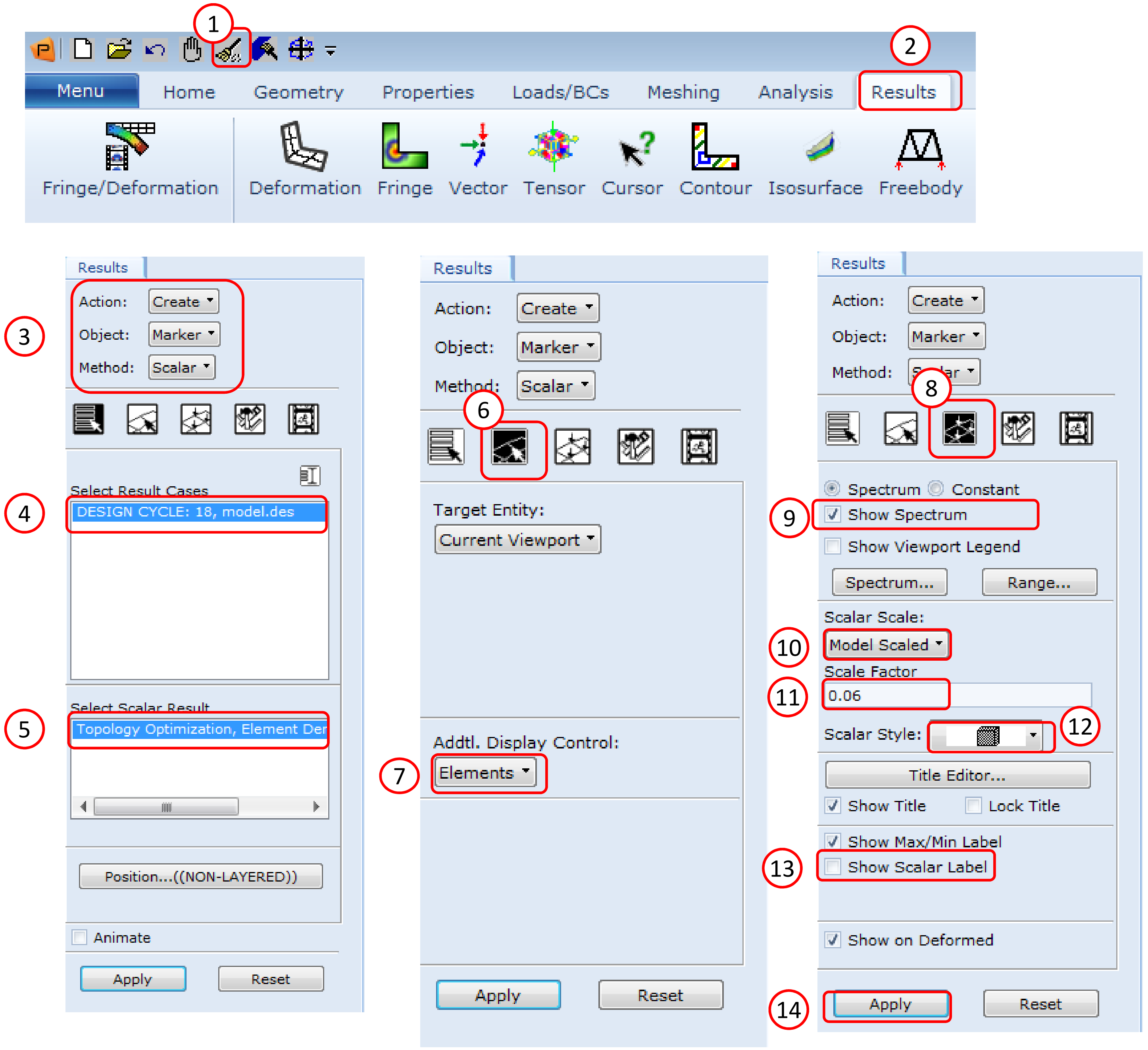
Review Optimization Results in Patran

1. Click Smooth Shading
2. Go to Tools > Design Study and click on Post-Process
3. Click Select Results File
4. Type this in the File name input box and press the Enter key: *.
   - This will display all the files in the directory
5. Select one of the following result files:
   - m.des-000i
     - This file is created when this entry is in the BDF file: PARAM DESPCH1 -1
   - model.des
     - This file is created when this entry is not in the BDF file: PARAM DESPCH1 -1
6. Click OK
7. Click Apply



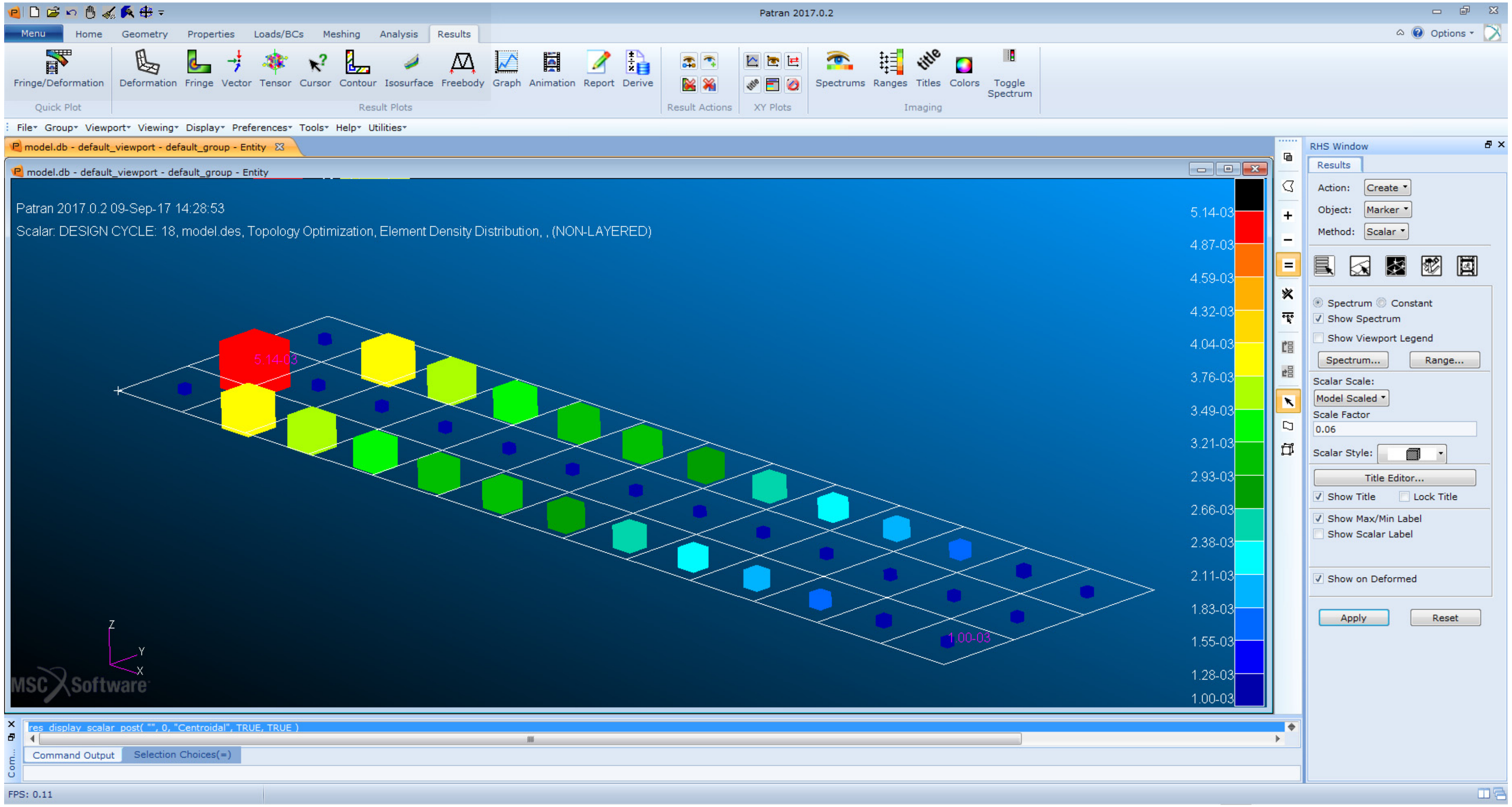
Review Optimization Results in Patran

1. Click the clear icon
2. Click Results
3. Set the following:
    1. Action: Create
    2. Object: Marker
    3. Method: Scalar
4. Select: DESIGN CYCLE: 18, model.des (The row should be highlighted blue)
5. Select Topology Optimization, Element D… (The row should be highlighted blue)
6. Click Target Entities
7. Set to Elements
8. Click Display Attributes
9. Mark the checkbox for Show Spectrum
10. Change to Model Scaled
11. Set Scale Factor to .06
12. Change Scalar Style to the shaded cube
13. Uncheck the box for Show Scalar Label
14. Click Apply



Review Optimization Results in Patran

1. The plot shows the new thickness distribution of each element. Plot not to scale.



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.

model.pch when DESPCH=-1

```
$
$          *************************************************************
$          *                                                           *
$          *      CONTINUOUS  DESIGN  CYCLE  NUMBER  =        18       *
$          *                                                           *
$          *************************************************************
$
$
$          *************************************************************
$          *                                                           *
$          *      CONTINUOUS  DESIGN  CYCLE  NUMBER  =        18       *
$          *                                                           *
$          *************************************************************
$
$
$   UPDATED ANALYSIS MODEL DATA ENTRIES
$
PSHELL  ********       1 .001           1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .003932        1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .003583        1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .00326         1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .003114        1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .002941        1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .00274         1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .002506        1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .002254        1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .001968        1    1.0           1 .833333   0.0
                               0
PSHELL  ********       1 .001628        1    1.0           1 .833333   0.0
                               0
```

model.pch when DESPCH=1

```
$
$          *************************************************************
$          *                                                           *
$          *      CONTINUOUS  DESIGN  CYCLE  NUMBER  =        18       *
$          *                                                           *
$          *************************************************************
$
$
$          *************************************************************
$          *                                                           *
$          *      CONTINUOUS  DESIGN  CYCLE  NUMBER  =        18       *
$          *                                                           *
$          *************************************************************
$
$
$   UPDATED ANALYSIS MODEL DATA ENTRIES
$
PSHELL*       1000000001                      1  1.00007309E-03               1*
*         1.00000000E+00                      1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000002                      1  3.93152203E-03               1*
*         1.00000000E+00                      1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000003                      1  3.58257980E-03               1*
*         1.00000000E+00                      1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000004                      1  3.26027502E-03               1*
*         1.00000000E+00                      1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000005                      1  3.11369399E-03               1*
*         1.00000000E+00                      1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000006                      1  2.94087051E-03               1*
*         1.00000000E+00                      1  8.33333333E-01  0.00000000E+00*
```

Update the Original Model

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

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

new_2D_elements.tmp

new_2D_elements.tmp

```
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
CQUAD4,8,1000000008,8,9,23,22
CQUAD4,9,1000000009,9,10,24,23
CQUAD4,10,1000000010,10,11,25,24
```

Original BDF File

topometry_cantilever_plate.bdf

```
CQUAD4   1    1    1    2    16   15
CQUAD4   2    1    2    3    17   16
CQUAD4   3    1    3    4    18   17
CQUAD4   4    1    4    5    19   18
CQUAD4   5    1    5    6    20   19
CQUAD4   6    1    6    7    21   20
CQUAD4   7    1    7    8    22   21
CQUAD4   8    1    8    9    23   22
CQUAD4   9    1    9    10   24   23
CQUAD4   10   1    10   11   25   24
```

2

model.pch

```
$
$        ************************************************************
$        *                                                          *
$        *     CONTINUOUS  DESIGN  CYCLE  NUMBER  =         18      *
$        *                                                          *
$        ************************************************************
$
$
$        ************************************************************
$        *                                                          *
$        *     CONTINUOUS  DESIGN  CYCLE  NUMBER  =         18      *
$        *                                                          *
$        ************************************************************
$
$
$  UPDATED ANALYSIS MODEL DATA ENTRIES
$
PSHELL*       1000000001              1  1.00007309E-03              1*
*        1.00000000E+00               1  8.33333333E-01  0.00000000E+00*
*                                                      0               *
*
PSHELL*       1000000002              1  3.93152203E-03              1*
*        1.00000000E+00               1  8.33333333E-01  0.00000000E+00*
*                                                      0               *
*
PSHELL*       1000000003              1  3.58257980E-03              1*
*        1.00000000E+00               1  8.33333333E-01  0.00000000E+00*
*                                                      0               *
```

1

PCH

topometry_cantilever_plate.bdf

```
$ MSC.Nastran input file created on September 09, 2017 at
$ Patran 2017.0.2
$ Direct Text Input for Nastran System Cell Section
$ Direct Text Input for File Management Section
$ Direct Text Input for Executive Control
$ Linear Static Analysis, Database
SOL 101
CEND
$ Direct Text Input for Global Case Control Data
TITLE = MSC.Nastran job created on 09-Sep-17 at 12:47:20
ECHO = NONE
SUBCASE 1
$ Subcase name : Default
   SUBTITLE=Default
   SPC = 2
   LOAD = 2
   DISPLACEMENT(PLOT,SORT1,REAL)=ALL
   SPCFORCES(PLOT,SORT1,REAL)=ALL
   STRESS(PLOT,SORT1,REAL,VONMISES,BILIN)=ALL
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM    POST    0
PARAM   PRTMAXIM YES
$ Elements and Element Properties for region : Plate
PSHELL   1       1      .01      1               1
$ Pset: "plate" will be imported as: "pshell.1"
CQUAD4   1       1       1       2       16      15
```

Original BDF 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/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, 'PSHELL', 'new_pshell_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

```
CQUAD4,1,1000000001,1,2,16,15
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
CQUAD4,8,1000000008,8,9,23,22
CQUAD4,9,1000000009,9,10,24,23
CQUAD4,10,1000000010,10,11,25,24
CQUAD4,11,1000000011,11,12,26,25
CQUAD4,12,1000000012,12,13,27,26
CQUAD4,13,1000000013,13,14,28,27
CQUAD4,14,1000000014,15,16,30,29
CQUAD4,15,1000000015,16,17,31,30
CQUAD4,16,1000000016,17,18,32,31
CQUAD4,17,1000000017,18,19,33,32
CQUAD4,18,1000000018,19,20,34,33
CQUAD4,19,1000000019,20,21,35,34
CQUAD4,20,1000000020,21,22,36,35
CQUAD4,21,1000000021,22,23,37,36
CQUAD4,22,1000000022,23,24,38,37
CQUAD4,23,1000000023,24,25,39,38
CQUAD4,24,1000000024,25,26,40,39
CQUAD4,25,1000000025,26,27,41,40
CQUAD4,26,1000000026,27,28,42,41
CQUAD4,27,1000000027,29,30,44,43
CQUAD4,28,1000000028,30,31,45,44
CQUAD4,29,1000000029,31,32,46,45
```

Original BDF File

topometry_cantilever_plate.bdf

```
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
CQUAD4,8,1000000008,8,9,23,22
CQUAD4,9,1000000009,9,10,24,23
CQUAD4,10,1000000010,10,11,25,24
CQUAD4,11,1000000011,11,12,26,25
CQUAD4,12,1000000012,12,13,27,26
CQUAD4,13,1000000013,13,14,28,27
CQUAD4,14,1000000014,15,16,30,29
CQUAD4,15,1000000015,16,17,31,30
CQUAD4,16,1000000016,17,18,32,31
CQUAD4,17,1000000017,18,19,33,32
CQUAD4,18,1000000018,19,20,34,33
CQUAD4,19,1000000019,20,21,35,34
CQUAD4,20,1000000020,21,22,36,35
CQUAD4,21,1000000021,22,23,37,36
CQUAD4,22,1000000022,23,24,38,37
CQUAD4,23,1000000023,24,25,39,38
CQUAD4,24,1000000024,25,26,40,39
CQUAD4,25,1000000025,26,27,41,40
CQUAD4,26,1000000026,27,28,42,41
CQUAD4,27,1000000027,29,30,44,43
CQUAD4,28,1000000028,30,31,45,44
CQUAD4,29,1000000029,31,32,46,45
```

1

Update the Original Model

The Python script generates a new TMP file.

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

new_pshell_entries.tmp

new_pshell_entries.tmp

```
PSHELL*       1000000001                       1  1.00007309E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000002                       1  3.93152203E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000003                       1  3.58257980E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000004                       1  3.26027502E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000005                       1  3.11369399E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000006                       1  2.94087051E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000007                       1  2.74009123E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000008                       1  2.50630222E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000009                       1  2.25367849E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000010                       1  1.96759530E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
```

Original BDF File

topometry_cantilever_plate.bdf

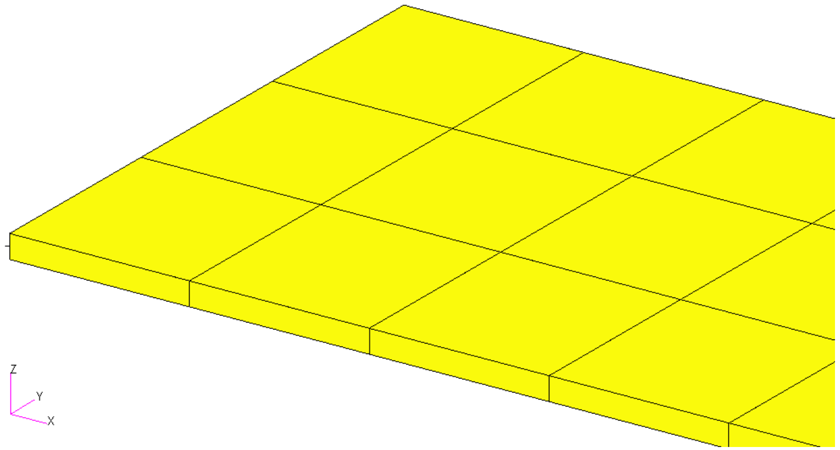
```
    STRESS(PLOT,SORT1,REAL,VONMISES,BILIN)=ALL
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM    POST    0
PARAM   PRTMAXIM YES
$ Elements and Element Properties for region : Plate
PSHELL*       1000000001                       1  1.00007309E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000002                       1  3.93152203E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000003                       1  3.58257980E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000004                       1  3.26027502E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000005                       1  3.11369399E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000006                       1  2.94087051E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000007                       1  2.74009123E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
PSHELL*       1000000008                       1  2.50630222E-03              1*
*         1.00000000E+00                       1  8.33333333E-01  0.00000000E+00*
*                                                              0               *
*
```

1

Results

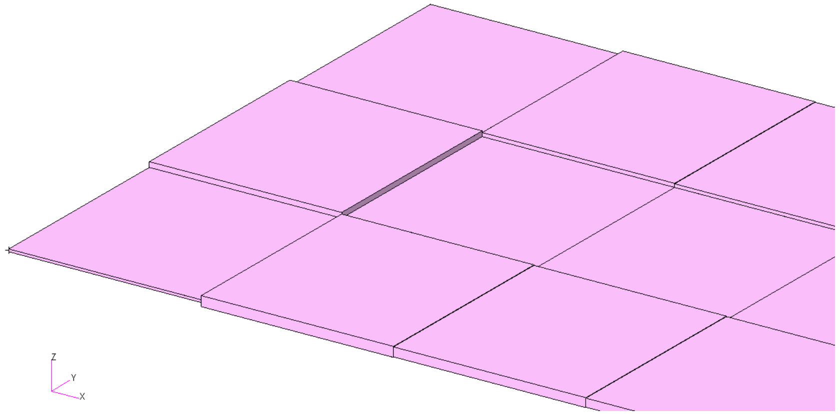
Before Optimization

- Mass: 19.5 kg



After Optimization

- Mass: 3.97 kg
- Vary the thickness of each element



End of Tutorial