Simulation-Based Design Optimization

PRESENTED FOR

SMASH Program
September 30, 2024

PRESENTED BY

Miguel A. Aguilo, PhD
maguilo@morphorm.com
www.morphorm.com

Agenda

- Introduction
- Examples
  - Shape Matching
  - Shape Matching + Force Calibration
  - Thermomechanical Stress
  - Large Deformations
- Enhancements

Introduction

Simulation-Based Design Optimization

Project Overview

<u>Technical Objective</u>: Design the internal topology of a finite-thickness, self-actuating surface to optimally match multiple target configurations while undergoing thermal and mechanical loading

<u>Technical Approach</u>: (1) Apply topology optimization to find a configuration capable of matching a desired deformation profile, (2) Apply multi-level optimization to codesign the actuating forces and the material layout

<u>Deliverables</u>: Software source code and executables plus documentation

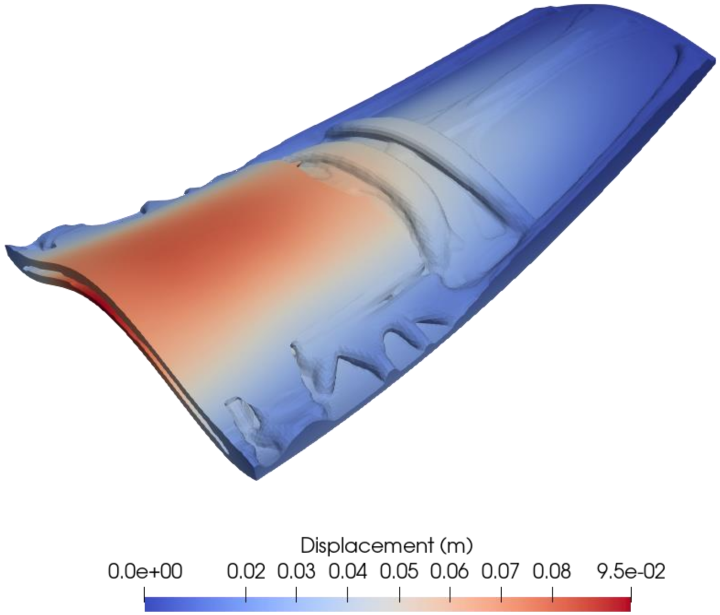


Figure 1. Displacement field for topology optimized design. The design was optimized for thermal and mechanical environments.

Morphorm History

Genesis SNL Plato	DARPA TRADES	SNL DfAM LDRD	SPP SNL-AFRL	SNL NNSA	SNL ASC	Morphorm Spinoff
2015 →	2017 →	2018 →	2019 →	2021 →	2021 →	2022

- Software foundations developed at Sandia National Laboratories (SNL)
- Modern software quality and development practices
- Ongoing partnership with SNL to integrate the latest simulation-based optimization research
- New major release every March 15th and October 15th
    - New minor release every three weeks
- Partners: SNL, DOE, AFRL, JHAPL, Private Industry

Computing & Parallelism

- Runs in various computing environments
  - Desktop: Linux, Mac, and Windows (Experimental)
  - HPC: Linux Clusters
  - Cloud Platforms: GCP, AWS
- Exploits concurrency at multiple levels
  - Multiprocessor and multi-GPU simulations
  - Multiple simulations per response
  - Samples in a parameter study
- File management features, including
  - Work directories to partition analysis files

Many Numerical Methods in One

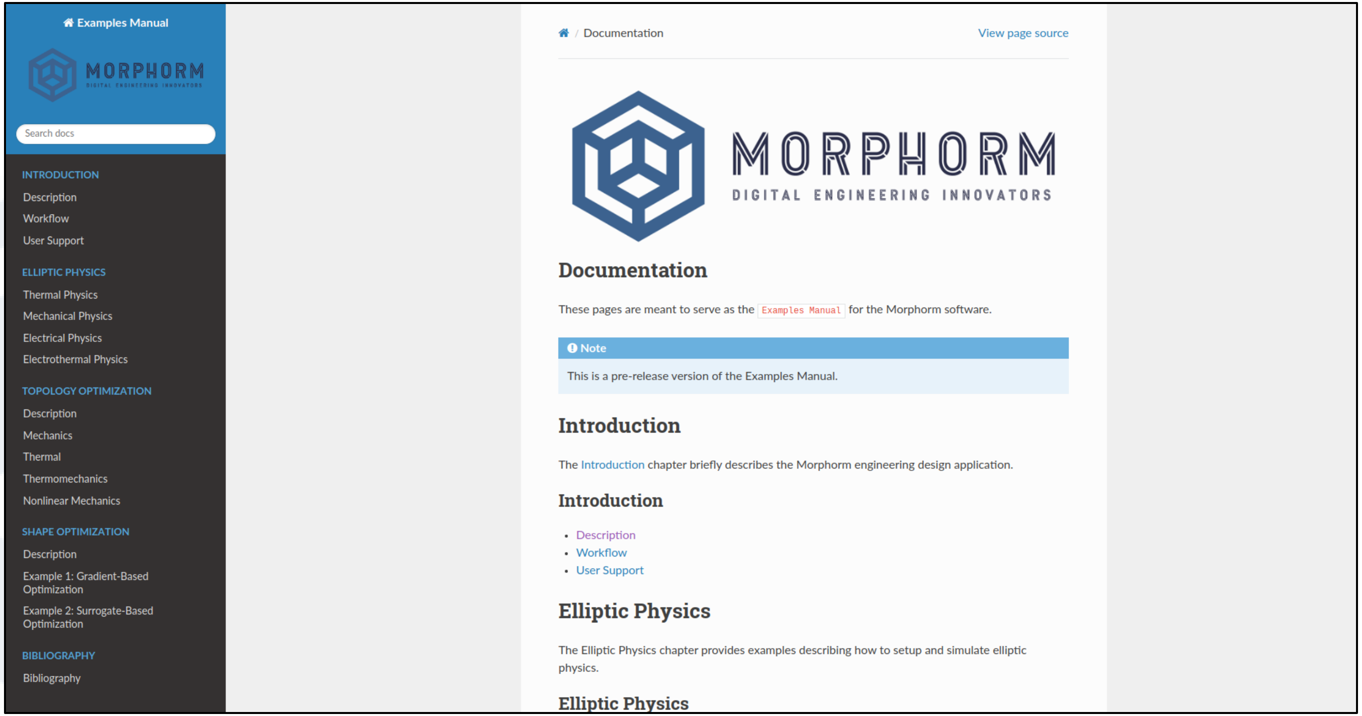
Geometry Optimization	**Optimization**

• Topology Optimization
  o Density Methods
  o Level Set (Under Development)
• Shape Optimization
  o ESP Integration, Cubit Journal File[1]
	• Gradient-based
	• Derivative-free
	• Global Optimization
	• Surrogate-Based Global
	• Automatic Differentiation

Uncertainty Quantification	**Finite Element Analysis**

• SNL Dakota Integration
  o Monte Carlo, Latin Hypercube Sampling, Reliability,
  o Design of Experiments
  o Sensitivity Analysis
 	
[1] https://cubit.sandia.gov/
	• Elliptic Physics: Thermal, Linear/Nonlinear Mechanical & Thermal-Mechanical, Electrical, Electrothermal
	• Parabolic Physics: Heat Equation
	• Hyperbolic Physics: Mechanical, Thermal-Mechanical

Resources


Examples Manual

MORPHORM
DIGITAL ENGINEERING INNOVATORS

Search docs

INTRODUCTION
- Description
- Workflow
- User Support

ELLIPTIC PHYSICS
- Thermal Physics
- Mechanical Physics
- Electrical Physics
- Electrothermal Physics

TOPOLOGY OPTIMIZATION
- Description
- Mechanics
- Thermal
- Thermomechanics
- Nonlinear Mechanics

SHAPE OPTIMIZATION
- Description
- Example 1: Gradient-Based Optimization
- Example 2: Surrogate-Based Optimization

BIBLIOGRAPHY
- Bibliography

/ Documentation

View page source

MORPHORM
DIGITAL ENGINEERING INNOVATORS

Documentation

These pages are meant to serve as the `Examples Manual` for the Morphorm software.

Note

This is a pre-release version of the Examples Manual.

Introduction

The Introduction chapter briefly describes the Morphorm engineering design application.

Introduction

- Description
- Workflow
- User Support

Elliptic Physics

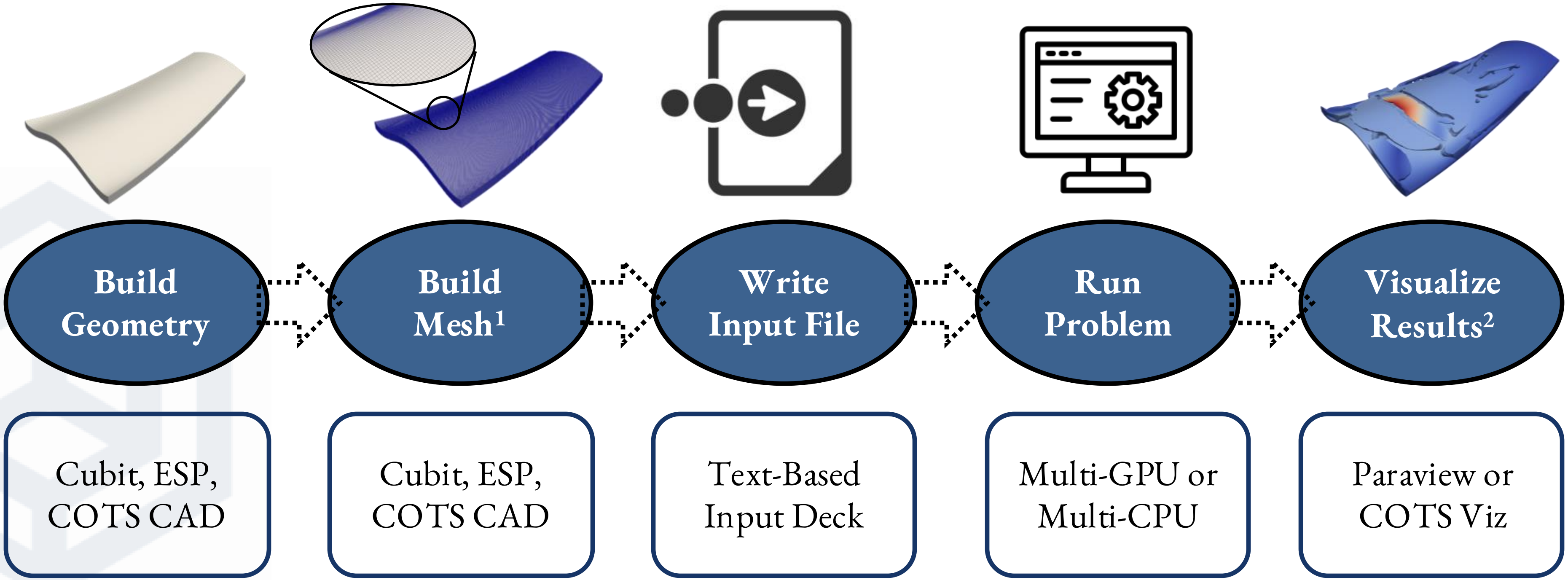
The Elliptic Physics chapter provides examples describing how to setup and simulate elliptic physics.

Elliptic Physics

HTML and PDF documentation:

- Examples Manual
- Theory Manual
- User Manual

Workflow



[1] Auto-mesh generation tools exist for shape optimization when using ESP and Cubit
[2] Morphorm uses the Exodus file format for the finite element database

Input Deck*

The input deck is broken down into several building blocks that are combined to define a study:

- **Scenario:** Setup for the physics-based simulation
- **Service:** Setup for an application performing computational tasks
- **Load:** Setup for external forces
- **Boundary Condition:** Setup for Dirichlet boundary conditions
- **Material:** Setup for material constitutive models
- **Block:** Setup for assigning a material to a region
- **Criterion:** Setup for defining a performance criterion
- **Objective:** Setup for a real-valued function that is minimized/maximized
- **Constraint:** Setup for a real-valued function defining a limit on a performance criterion
- **Study:** Setup for the numerical methods used in a design study

Run Job: python runmorphorm.py inputfile.i

* See User Manual for an in-depth discussion about the input deck building blocks.

Example 1

Simulation-Based Design Optimization

Shape Matching

Example 1: Shape Matching*

Problem Description. Find a lightweight structure capable of matching a desired deformation profile. The actuating forces are considered known.

Material. The structure is made of an elastic material:

- Young's Modulus: 325.0 GPa
- Poisson's Ratio: 0.33

Model. The model is symmetric about the plane at X=0.0, only half of the model is considered for optimization.

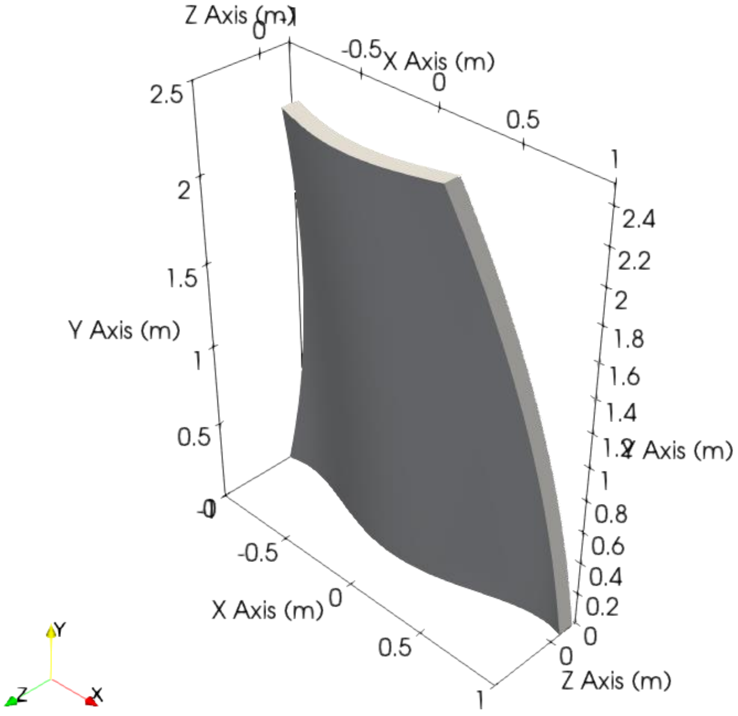


Figure 2. Initial design domain for the topology optimization study.

* See Example 3.2.5 in the Examples Manual for details.

Example 1: Boundary Conditions

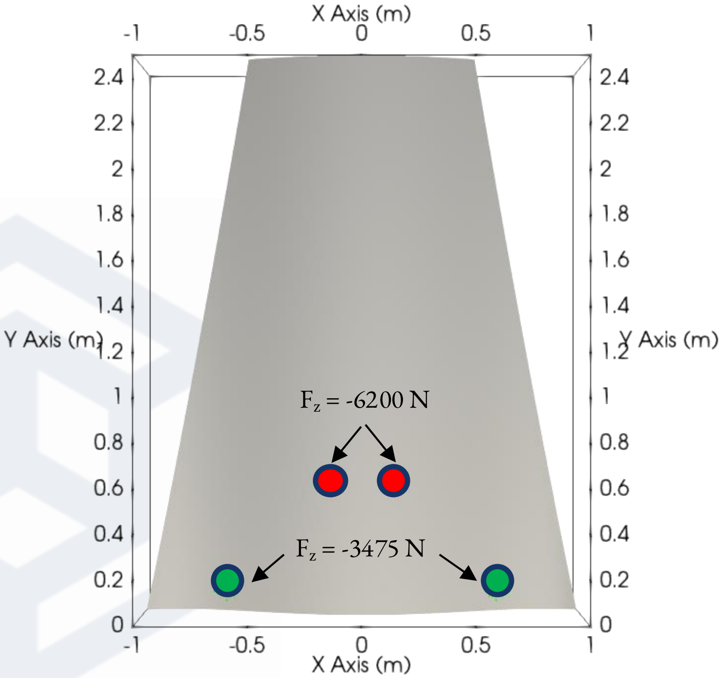


*Figure 3. Locations of the actuating forces on the top surface. Regions with matching colors have the same force components. The **red** and **green** regions are non-optimizable regions.*

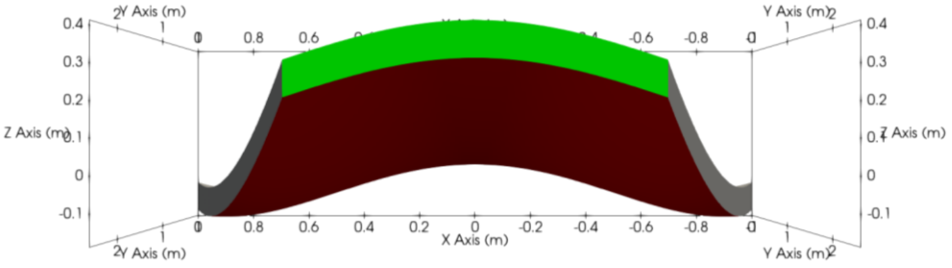


*Figure 4. Bottom **red surface** represents the target surface, i.e., deformation profile will be matched at this surface. X, Y, and Z displacements on the **green surface** are fixed. The **red surface** is a non-optimizable region.*

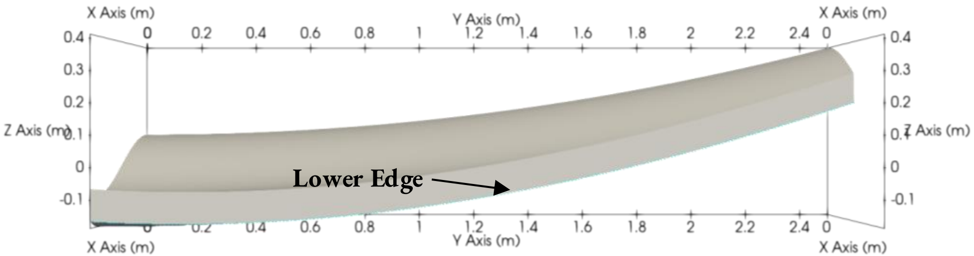


Figure 5. X, Y, and Z displacements along the lower edge are fixed.

Example 1: Formulation Overview

Structural Topology Optimization

Goal: Shape Matching

$$
\begin{aligned}
\min_{\mathbf{z}\in\mathbb{R}^{N_\mathbf{z}}} \quad & f\left(\mathbf{u}(\mathbf{z})\right) \longleftarrow \boxed{\text{Shape Matching Criterion}} \\
\text{s.t.} \quad & \mathbf{R}\left(\mathbf{u}(\mathbf{z}), \mathbf{z}\right) = 0 \longleftarrow \boxed{\text{Residual Equation}} \\
\boxed{\text{Mass Constraint}} \longrightarrow \quad & h_1\left(\mathbf{z}\right) \leq 0 \\
& h_2\left(\mathbf{u}(\mathbf{z}), \mathbf{z}\right) \leq 0 \longleftarrow \boxed{\text{Structural Stiffness Constraint}} \\
\boxed{\text{Geometry parameters}} \longrightarrow \quad & \underline{z}_i \leq z_i \leq \bar{z}_i, \; i = 1, \ldots, N_z,
\end{aligned}
$$

Example 1: Input Deck

State Misfit Criterion. Measures mismatch between the desired deformation profile and the actual (simulated) deformation profile

Inverse Criterion. Setup for an inverse design criterion. The state misfit criterion must be assigned to an inverse design criterion. The following inputs must be provided:

- Target data: Specifies the csv file containing the target value(s) at the measurement nodes
- Target degree of freedom(s): Specifies the degrees of freedom(s) to be matched

```
begin service 1
  code engine
  number_processors 1
  number_ranks 1
end service

begin service 2
  code analyze
  number_processors 1
  number_ranks 1
end service

begin criterion 1
  type mechanical_compliance
end criterion

begin criterion 2
  type inverse
  criterion_ids 3
  criterion_weights 1
  target_data 3d_pwr_v1_1_1D_surf.csv
  degree_of_freedom dispz
end criterion

begin criterion 3
  type state_misfit
  transform exponential
end criterion

begin criterion 4
  type volume
end criterion

begin scenario 1
  physics steady_state_mechanics
  service 2
  dimensions 3
  loads 1 2
  boundary_conditions 1 2 3
  linear_solver 1
  blocks 1
  output 1
  material_penalty_exponent 4.0
end scenario

begin linear_solver 1
```

Inverse Criterion

State Misfit Criterion

Figure 6. Snippet from the input deck use for the inverse topology optimization study.

Example 1: CSV File Format

EXODUS_GLOBAL_NODE_ID_1	TARGET_VALUE_QOI_1	...	TARGET_VALUE_QOI_N
:	:		:
:	:		:
EXODUS_GLOBAL_NODE_ID_N	TARGET_VALUE_QOI_1	...	TARGET_VALUE_QOI_N

- **EXODUS_GLOBAL_NODE_ID_{integer}** is the exodus global node id
  - {integer} = 1, ..., N_{node}, where N_{node} is the number of target global node ids
- **TARGET_VALUE_QOI_{dof}** is the value for the quantity of interest
  - {dof} = 1, ..., N_{dofs}, where N_{dofs} is the number of degrees of freedom

* See User Manual for an in-depth discussion about the CSV file format (Section 2.9.3 - Inverse Criterion).

Example 1: CSV File Format (cont.)

Approach 1. Matlab File

Create a Matlab file using the exomat application, e.g.,

- **Command.** exomat mesh.exo
- **Input.** exodus mesh file, e.g., mesh.exo
- **Output. mesh.mat** file, saves finite element mesh data
  - Upload in Matlab or Python
  - Default behavior is to use the prefix for the exodus file, e.g., mesh

Approach 2: Text File

Morphorm outputs a text file with the exodus global node ids

- Requires an input deck
  - Must define the writevars block and set the study method to **preprocess**

Example 1: Access Exodus Global Node IDs

Approach 1: Matlab File

Loads Matlab File & Creates a Python Dictionary

Python Dictionary Keys

Simple python script

```
>>> from mat4py import loadmat
>>> data = loadmat('3d_pwr_v1_1.mat')
>>> data.keys()
dict_keys(['naxes', 'nnodes', 'nelems', 'nblks', 'nnsets', 'nssets', 'nsteps', 'ngvars', 'nnvars', 'nevars', 'nnsvars', 'nssvars', 'Title', 'x0', 'y0', 'z0', 'ssids', 'ssnum
01', 'ssnod01', 'ssfac01', 'ssside01', 'sselem01', 'ssnum02', 'ssnod02', 'ssfac02', 'ssside02', 'sselem02', 'ssusernames', 'nsssides', 'nssdfac', 'nsids', 'nsnod01', 'nsfac0
1', 'nsnod02', 'nsfac02', 'nsnod03', 'nsfac03', 'nsnod04', 'nsfac04', 'nsnod05', 'nsfac05', 'nsnod06', 'nsfac06', 'nsusernames', 'nnsnodes', 'nnsdfac', 'blkids', 'blk01', 'b
lk01_nattr', 'blkusernames', 'blknames', 'node_num_map', 'elem_num_map'])
>>> data.get('ssids')
[[1], [2]]
>>> data.get('nsids')
[[1], [2], [3], [4], [5], [6]]
>>> data.get('nsnod05')
[[15847], [16703], [17993], [17995], [17997], [17999], [18001], [18003], [17151], [16723], [14997], [14999], [15001], [15003], [15421], [16275], [15419], [17131], [17133], [
17561], [17563], [17991], [18005], [17577], [17579], [16295], [16293], [15865], [15863], [15435], [15433], [15005], [14995], [15423], [16273], [16705], [16707], [16709], [16
711], [16713], [16715], [16717], [16719], [16721], [17135], [17137], [17139], [17141], [17143], [17145], [17147], [17149], [17565], [17567], [17569], [17571], [17573], [1757
5], [15425], [15427], [15429], [15431], [15849], [15851], [15853], [15855], [15857], [15859], [15861], [16277], [16279], [16281], [16283], [16285], [16287], [16289], [16291]
]
```

Exodus Side- & Node-Set IDs

Global Exodus Node IDs for Node Set 5

Note: Custom exodus node-set and side-set ids can be assigned when creating the exodus mesh file

Example 1: Access Exodus Global Node IDs

Approach 2: Text File

```
begin service 1
  code analyze
  number_processors 1
end variables

begin writevars 1
  variable gnode_id
  entity_name ss_1 ss_2
end writevars

begin output 1
  service 1
  writevars 1
end output
```

method for requesting the exodus global node ids

identifier for the exodus global node ids

exodus node set id from where the ids are extracted

```
begin study
  method preprocess
end study

begin mesh
  name mesh.exo
end mesh
```

Example 1: Main Output Data Files*

File	Description
morphorm_{type}_optimizer_diagnostics.txt • The {type} key is replaced with the optimizer name id	Diagnostics report for gradient-based optimizers. Holds the objective, constraint(s), and convergence criteria histories. The file is updated at each major optimization iteration. The file is written in the run directory.
opt_criteria_history.csv	Holds the objective and constraint(s) histories. The file is written at the end of the job. The file is written in the run directory.
iteration{itr}_criteria_history.csv • {itr} is replaced with the major optimization iteration • The {svid} and {scid} keys are replaced with the service id and scenario id, respectively	Holds the objective and constraint(s) histories up to the current major optimization iteration. The file is created at each major optimization iteration. The file is written inside the **analyze_id_{svid}_output_scenario_id_{scid}** directory, which is created in the run directory.

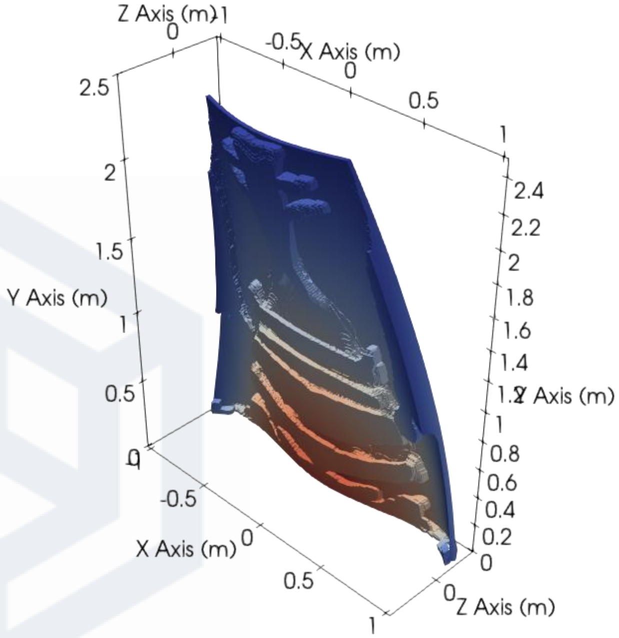
* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

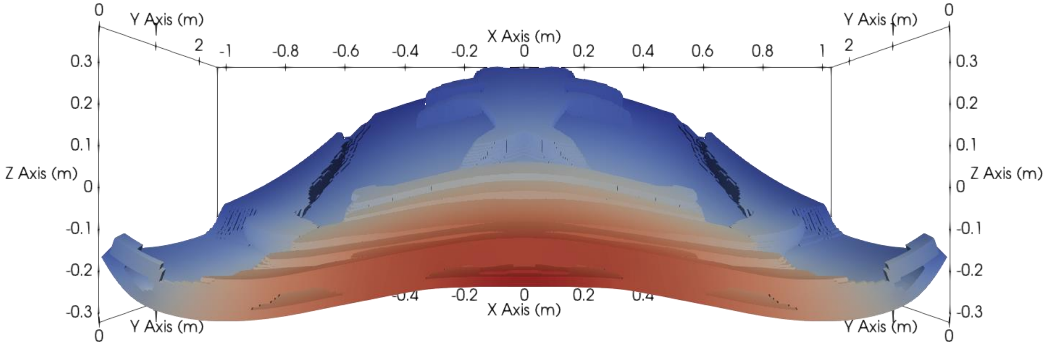
Example 1: Main Output Exodus Files*

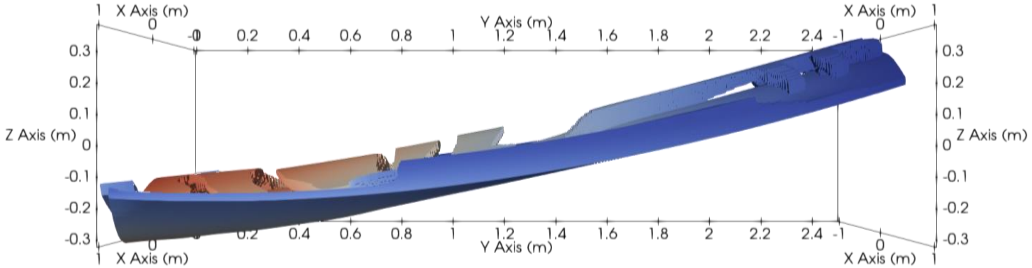
File	Description
engine_output.exo	Holds the optimized material layout history. The file is updated at each major optimization iteration.
iteration{itr}.exo • The {itr} key is replaced with the major optimization number • The {svid} and {scid} keys are replaced with the service id and scenario id, respectively	Holds the material layout and quantities of interest (e.g., displacement) fields for a major optimization iteration. The file is created at each major optimization iteration by default. An output frequency can be specified to manage when the file is written. The file is written in the **analyze_id_{svid}_output_scenario_id_{scid}** directory, which is created in the run directory.

* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

Example 1: Results







Mass Reduction: 75%

Deformation Mismatch: 3%

Example 2

Simulation-Based Design Optimization

Shape Matching + Force Calibration

Example 2: Shape Matching + Force Calibration*

Problem Description. Find a lightweight structure and the actuating forces needed to match a desired deformation profile. The locations of the actuating forces are assumed known.

Material. The structure is made of an elastic material:

- Young's Modulus: 325.0 GPa
- Poisson's Ratio: 0.33

Model. The model is symmetric about the plane at X=0.0, only half of the model is considered for optimization.

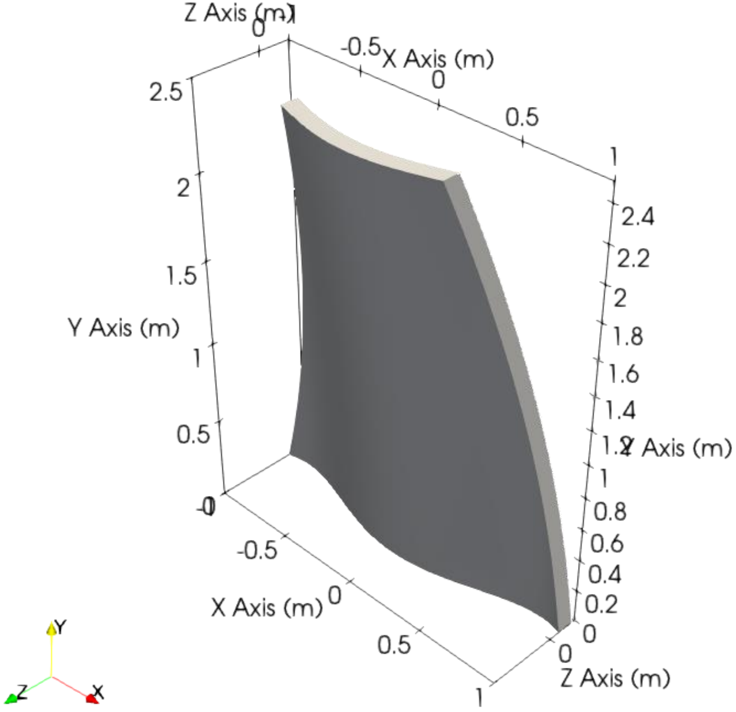


Figure 7. Initial design domain for the topology optimization study.

* See Example 3.2.6 in the Examples Manual for details.

Example 2: Mechanical Boundary Conditions

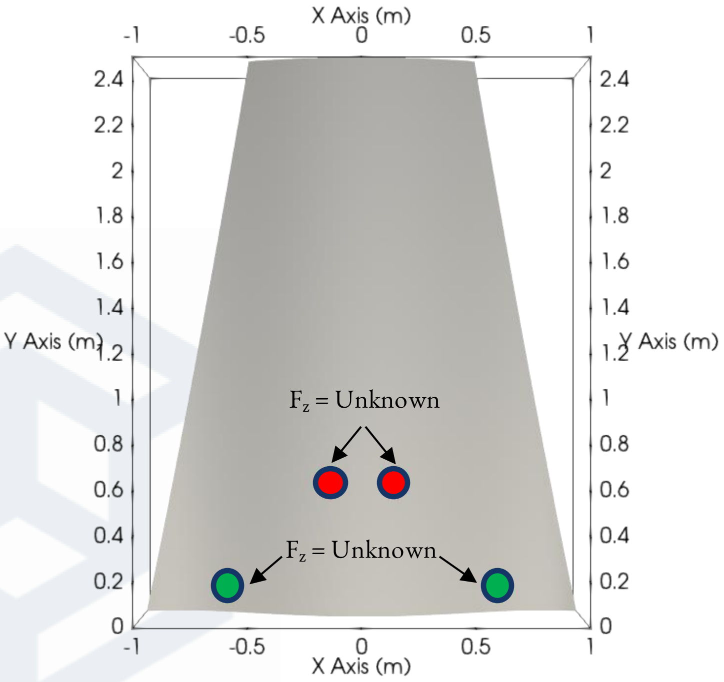


Figure 8. Locations of the actuating forces on the top surface. Regions with matching colors have the same force components. The ***red*** *and* ***green*** *regions are non-optimizable regions.*

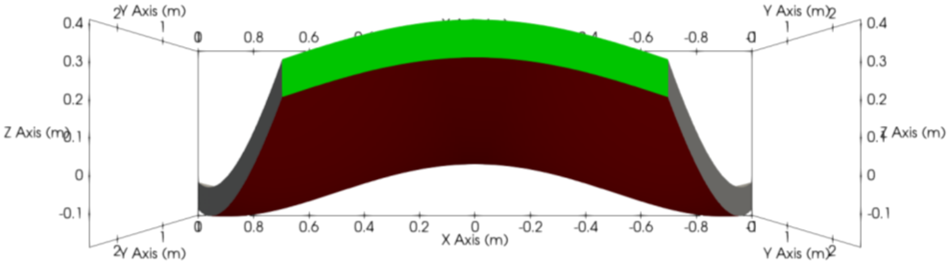


Figure 9. Bottom ***red surface*** *represents the target surface, i.e., deformation profile will be matched at this surface. X, Y, and Z displacements on the* ***green surface*** *are fixed. The* ***red surface*** *is a non-optimizable region.*

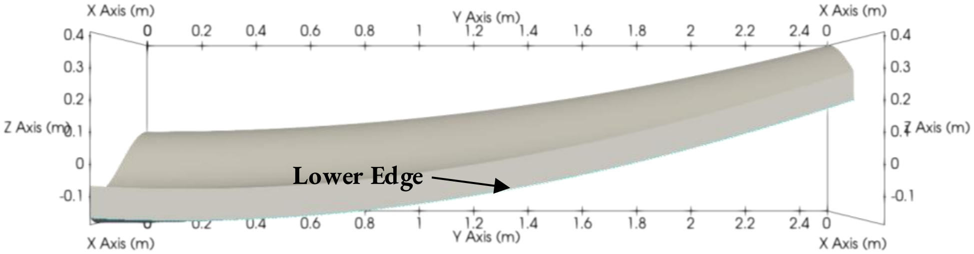


Figure 10. X, Y, and Z displacements along the lower edge are fixed.

Example 2: Formulation Overview

Multi-Level Optimization

Goal: Shape Matching & Force Identification

$$\min_{\mathbf{p}\in\mathbb{R}^{N_\mathbf{p}},\mathbf{z}\in\mathbb{R}^{N_\mathbf{z}}} \sum_{M=1}^{N_F} \alpha_M F_M(\mathbf{p},\mathbf{z})$$

Outer-level objective functions

Force parameters

$$\text{s.t.}\quad \underline{p}_i \le p_i \le \bar{p}_i \text{ for } i = 1,\dots,N_\mathbf{p}$$

$$\underline{z}_j \le z_j \le \bar{z}_j \text{ for } j = 1,\dots,N_\mathbf{z}$$

Geometry parameters

Outer-level constraint functions

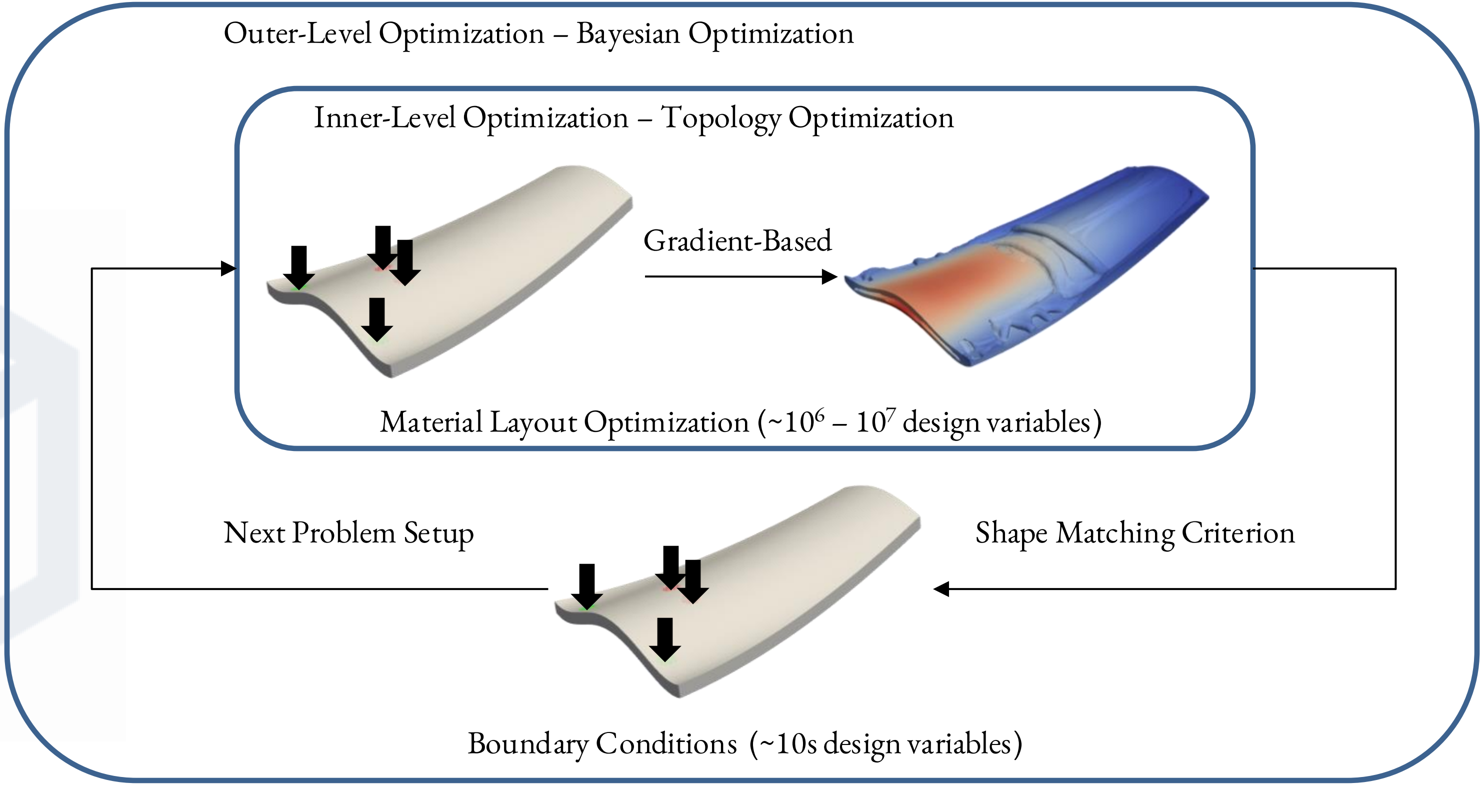
$$H_I(\mathbf{p},\mathbf{z}) \le 0 \text{ for } I = 1,\dots,N_H$$

$$\boxed{\mathbf{z} \in \arg\min_{\mathbf{z}\in\mathbb{R}^{N_\mathbf{z}}} \left\{\sum_{N=1}^{N_f} \beta_N f_N(\mathbf{p},\mathbf{z}) : \mathbf{R}(\mathbf{p},\mathbf{z}) = 0,\ h_J(\mathbf{p},\mathbf{z}) \le 0 \text{ for } J = 1,\dots,N_h\right\}}$$

Inner-level optimization problem

Surrogate-based optimization is applied to accelerate the design study

Example 2: Algorithm



Example 2: Input Decks

Requirements. The multi-level optimization approach requires two input decks. The <u>first input deck</u> is for the outer optimization problem (boundary conditions) and the <u>second input deck</u> is for inner optimization problem (material layout).

Additional Scripts. The multi-level optimization requires two additional scripts. The scripts are called by the outer-level optimizer to (1) update the input deck for the inner-level optimization problem, (2) launch the inner-level optimization problem, (3) and pass the design criteria values (objective and constraint function(s) values) to the outer level optimizer.

Parameter Substitutions. The built-in aprepro* application is used to perform hands-off parameter substitutions at runtime

*Sjaardema, Gregory, and USDOE. *Aprepro - Algebraic Preprocessor*. Computer software. *https://www.osti.gov//servlets/purl/2403459*. Vers. 2.0. USDOE. 7 Feb. 2011. Web. doi:10.11578/dc.20171025.1039.

Example 2: Input Decks (cont.)

Outer Level-Optimization

```
begin variables 1
  numvars 2
  type continuous_design
  initial_points    6e3    4e3
  lower_bounds      1e3    1e3
  upper_bounds      5e4    5e4
  descriptors       load1  load2  ←
end variables

begin study
  method surrogate_based_global_optimization
  variables 1
  num_samples 8
  num_criteria 1
  concurrent_evaluations 4  ←
  max_iterations 25
  link_files 3d_pwr_v1_1.exo
  copy_files target_data.csv analysis_driver.sh runmorphorm.py analysis.temp
postprocess.py
end study
```

Descriptors enable parameter substitutions

Outer-level optimizer runs four concurrent inner-level optimization problems

Notes

- The outer level optimizer creates a directory (**workdir**) where inner-level results are saved
- The **link_files** keyword creates a static link for large files, which are linked to the work directory
- The **copy_files** keyword copies files to the work directories

Example 2: Input Decks (cont.)

File	Description
3d_pwr_v1_1.exo	Exodus mesh file. Exodus mesh file are large files. The best practice is to create a static link.
runmorphorm.py	The run script contains the sequence of instructions to launch a Morphorm job. The script is standard and requires no alterations.
analysis_driver.sh	The analysis driver script launches the inner level optimization problem. The analysis driver script is lightweight and typically has a few instructions.
postprocess.py	The postprocessing script writes a text file with the best objective function(s) and constraint function(s) values. The outer level optimizer reads the text file and updates the outer level design variables based on these criteria values.
target_data.csv	The csv file contains the deformation profile to be matched by the optimizer.
analysis.temp	The input deck template for the inner optimization problem. The actuating force descriptors will be replaced with values at runtime.

Example 2: Input Decks (cont.)

begin load 1
type concentrated_load
location_type nodeset
location_name actuator1
direction z
value {**load1**} ← Replaced with scalar values at runtime
end load

begin load 2
type concentrated_load
location_type nodeset
location_name actuator2
direction z
value {**load2**} ← Replaced with scalar values at runtime
end load

Input Deck – Inner Level Optimization

- The input deck for the inner-level optimization problem (material layout optimization) is like the input deck in Example 1, but with a minor modification to support parameter substitutions
  - The descriptors, e.g., **load1** and **load2**, are used to enable parameter substitutions
  - The descriptors are defined in the input deck for the outer-level optimization

Example 2: Analysis Driver Script

```
#!/bin/bash
dprepro $1 analysis.temp analysis.i
python runmorphorm.py analysis.i
python postprocess.py
mv results.out $2
```

- `dprepro $1 analysis.temp analysis.i` ← Performs parameter substitutions, i.e., creates input deck for the inner optimization problem based on the new actuating forces
- `python runmorphorm.py analysis.i` ← Runs inner optimization problem, i.e., material layout optimization
- `python postprocess.py` ← Extracts simulation objective and constraint values from the simulation output text file and writes the postprocessed values into the **results.out** file
- `mv results.out $2` ← Renames the **results.out** file to the name expected by the outer optimizer (surrogate-based optimization algorithm)

Note. The analysis driver script is standard for a broad class of optimization problems. Thus, its creation will be integrated into the Morphorm software in a future release.

Example 2: Postprocess Script

```
import itertools
tMyFileList=["opt_criteria_history.csv"]

tOutFile = open("results.out", "w+")
for tMyFile in tMyFileList:
  with open(tMyFile) as tFile:
    tLines = tFile.readlines()
    for tLine in tLines:
      tMyList = tLine.split(', ')
      if(tMyList[0] == "My Objective"):
        tOutFile.write(tMyList[len(tMyList)-2])
    tFile.close()
tOutFile.close();
```

Simulation output text file with the objective function(s) and constraint function(s) evaluation histories

Output text file containing the postprocessed objective function(s) and constraint function(s) values for the last inner optimization iteration

Note. The postprocess python script is standard for a broad class of optimization problems. Thus, its creation will be integrated into the Morphorm software in a future release.

Example 2: Main Output Data Files*

File	Description
morphorm_{type}_optimizer_diagnostics.txt	Description presented in Example 1
opt_criteria_history.csv	Description presented in Example 1
iteration{itr}_criteria_history.csv	Description presented in Example 1
{type}_tabular_results.dat • The {type} key is replaced with the name id for the optimizer	Writes a tabular results file with the design variable and response histories

* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

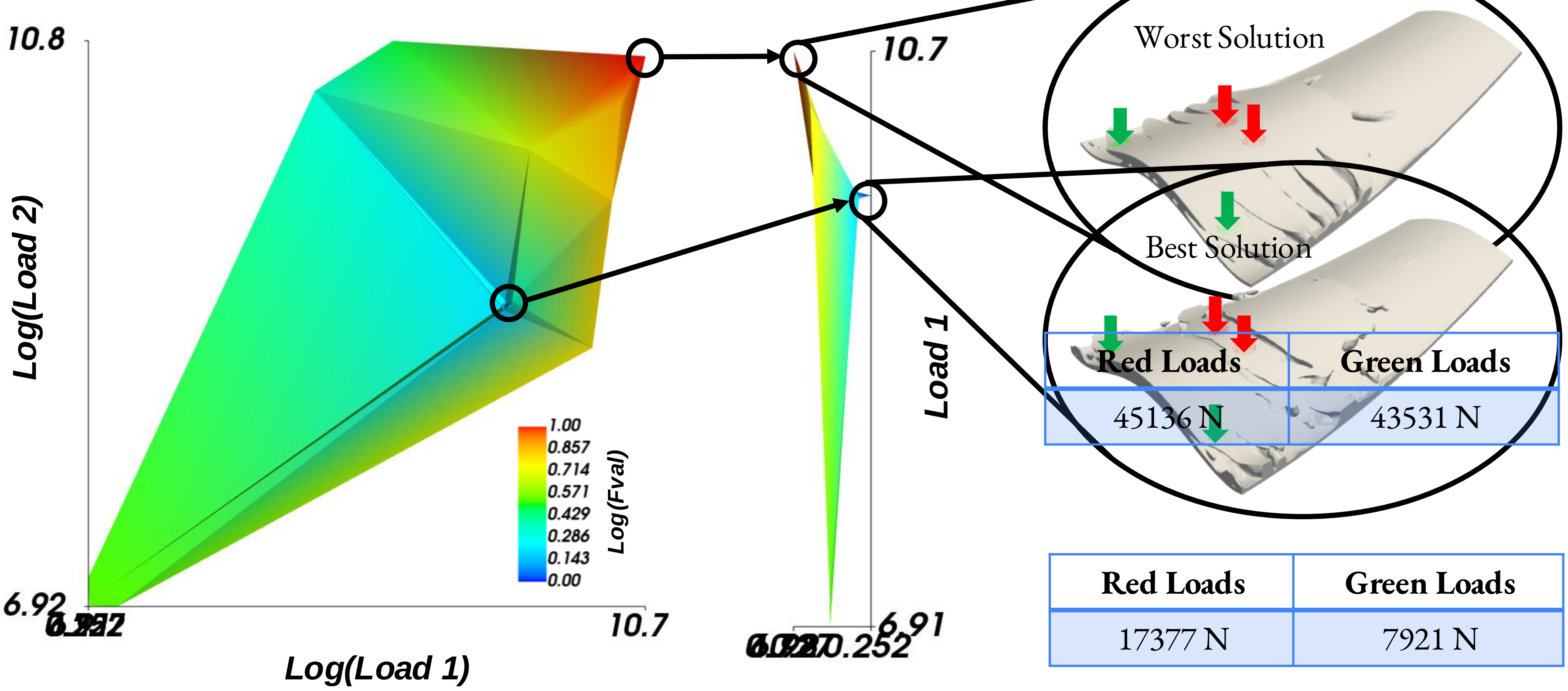
Note. The *morphorm_{type}_optimizer_diagnostics.txt*, *opt_criteria_history.csv*, and *iteration {itr}_criteria_history.csv* files will be located inside the work directory (**workdir**). The **workdir** is created in the run directory. The **workdir** will have the results for each inner optimization problem.

Example 2: Main Output Exodus Files*

File	Description
engine_output.exo	Description presented in Example 1
iteration{itr}.exo	Description presented in Example 1

* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

Example 2: Outer-Level Error Surface

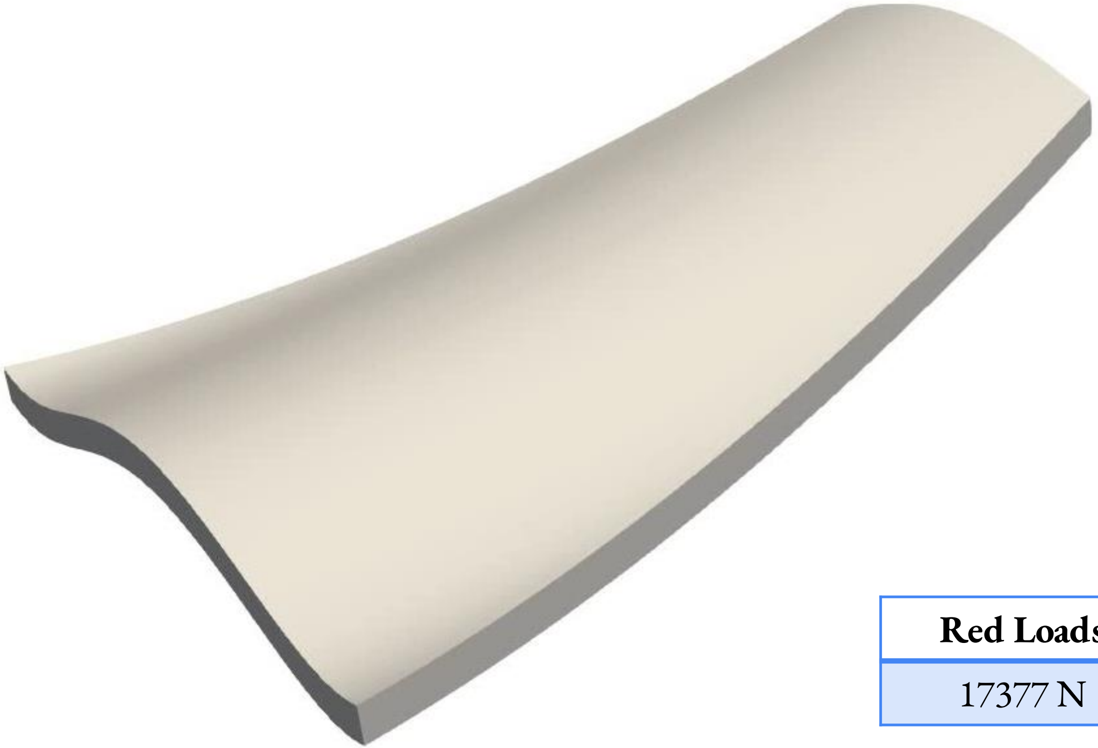


Red Loads	Green Loads
45136 N	43531 N

Red Loads	Green Loads
17377 N	7921 N

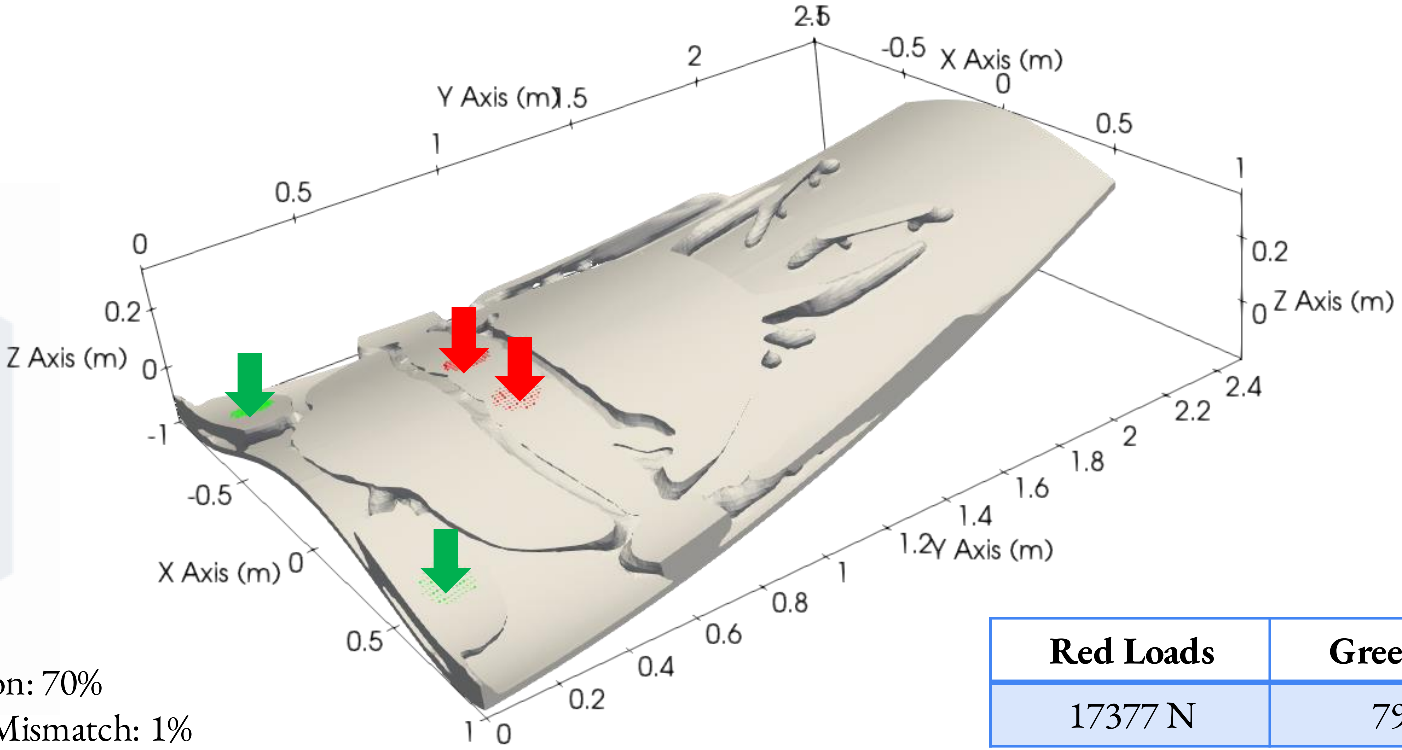
Note. The initial optimization study produced 35 design candidates.

Example 2: Optimized Design



Red Loads	Green Loads
17377 N	7921 N

Example 2: Optimized Design (cont.)

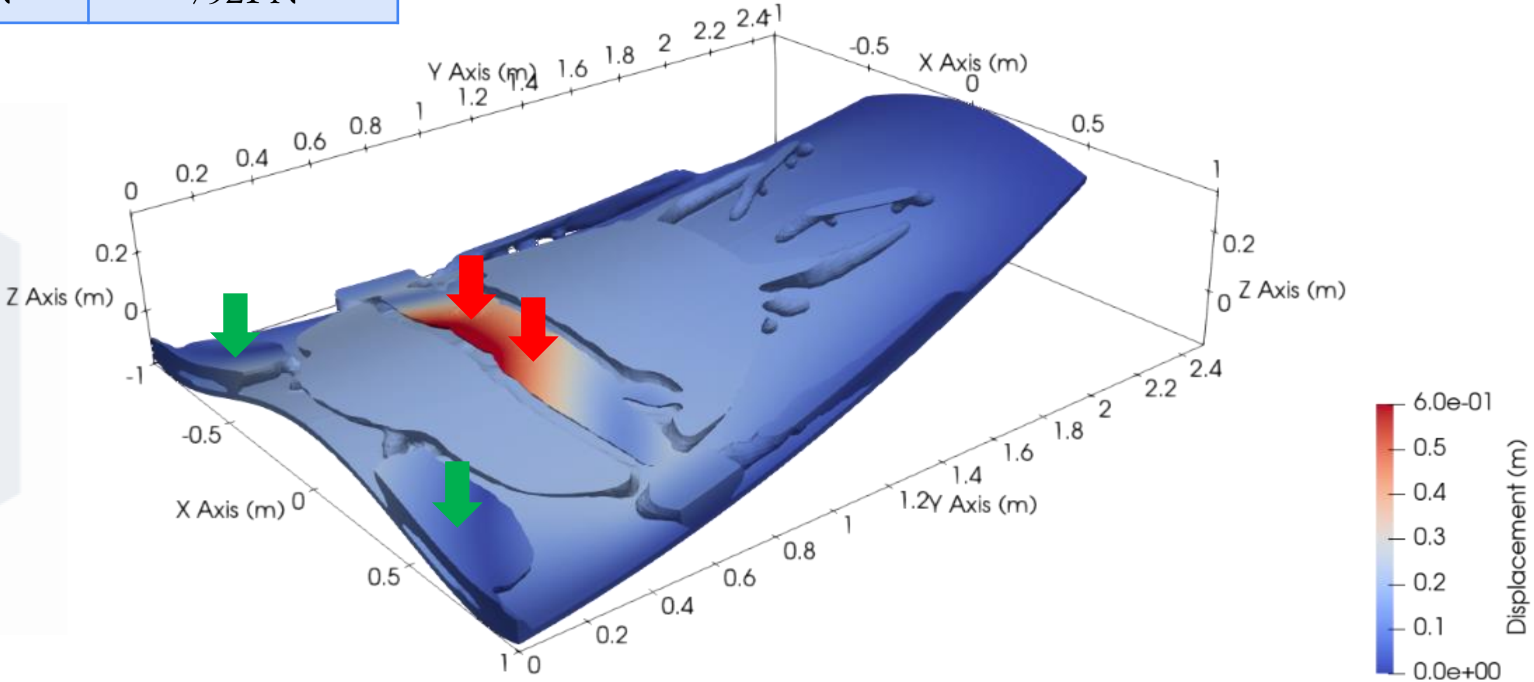


Mass Reduction: 70%
Deformation Mismatch: 1%

Red Loads	Green Loads
17377 N	7921 N

Example 2: Optimized Design (cont.)

Red Loads	Green Loads
17377 N	7921 N



Example 3

Simulation-Based Design Optimization

Design for Thermal & Mechanical Environments

Example 3: Thermomechanical Stress*

Problem Description. Find a lightweight structure that meets a von Mises stress limit at every material point while reducing the peak displacement. The actuating forces are considered known.

Material. The structure is made of an elastic material:

- Young's Modulus: 325.0 GPa
- Poisson's Ratio: 0.33
- von Mises Stress Limit: 200 MPa

Model. The model is symmetric about the plane at X=0.0, only half of the model is considered for optimization.

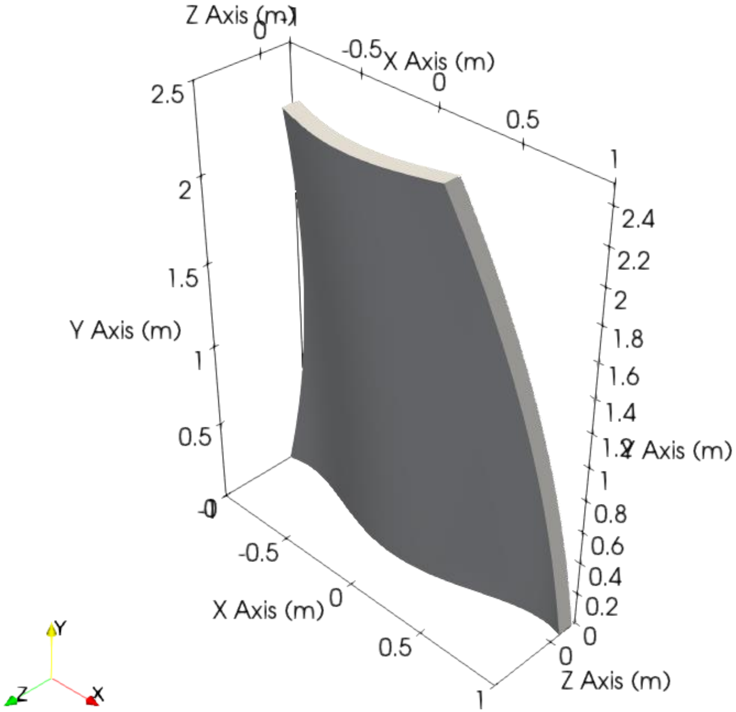


Figure 11. Initial design domain for the topology optimization study.

* See Example 3.4.3 in the Examples Manual for details.

Example 3: Mechanical Boundary Conditions

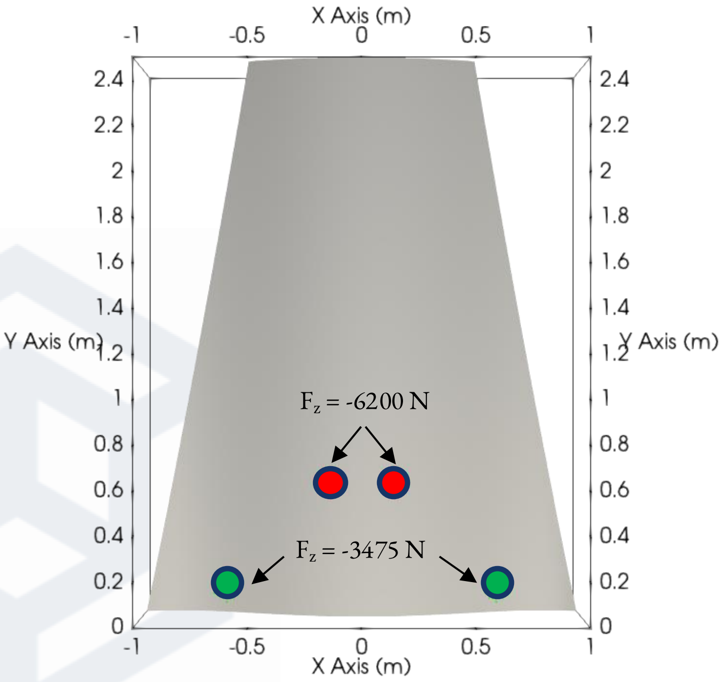


Figure 12. Locations of the actuating forces on the top surface. Regions with matching colors have the same force components. The ***red*** *and* ***green*** *regions are non-optimizable regions.*

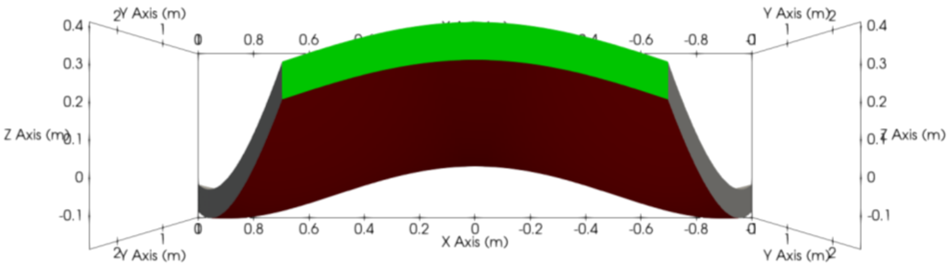


Figure 13. Bottom ***red surface*** *represents the target surface, i.e., deformation profile will be matched at this surface. X, Y, and Z displacements on the* ***green surface*** *are fixed. The* ***red surface*** *is a non-optimizable region.*

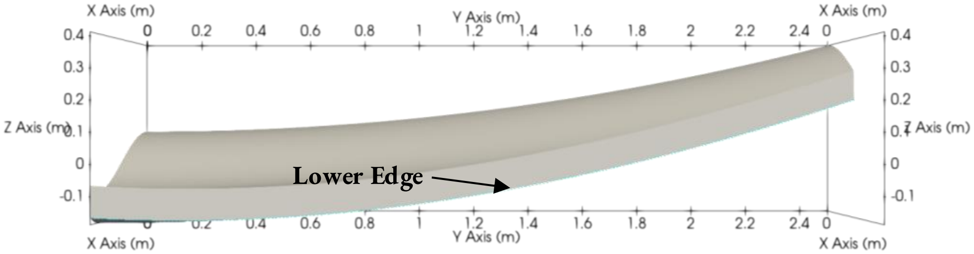


Figure 14. X, Y, and Z displacements along the lower edge are fixed.

Example 2: Thermal Boundary Conditions

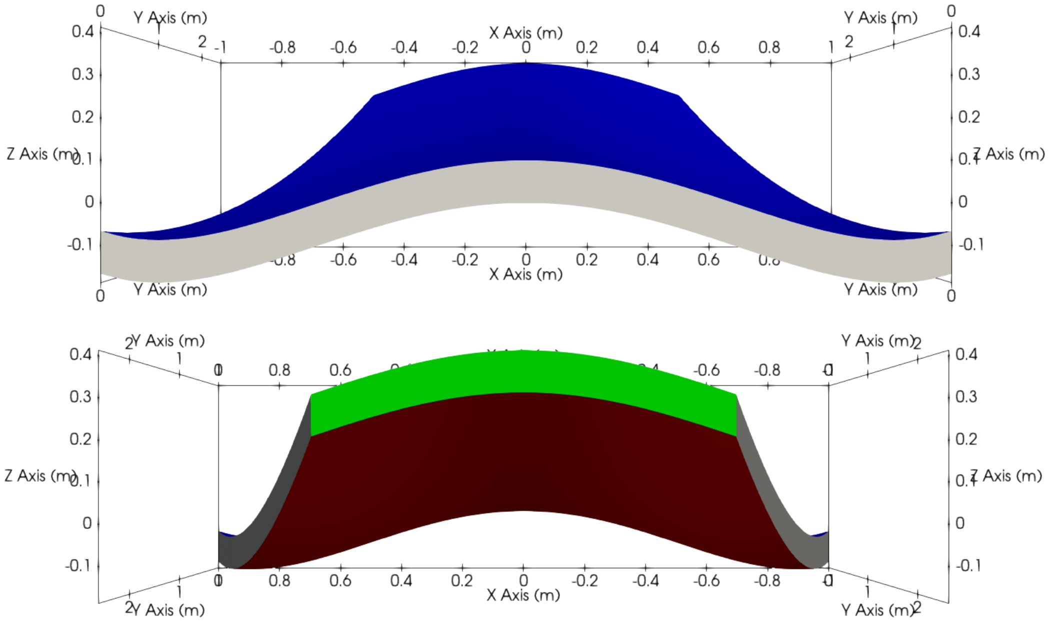


*Figure 15. The top and lower panes show the back and the front views for the initial design domain. The temperature at the top **blue surface** is kept at 700°C. The temperature at the bottom **red surface** is kept at 1200℃. The temperature at the front **green surface** is kept at 3000℃.*

Example 3: Formulation Overview

Standard Approach

$$
\begin{aligned}
\min_{\mathbf{z}\in\mathbb{R}^{N_{\mathbf{z}}}} \quad & \sum_{l=1}^{N_f} \alpha_l f_l\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right) \\
\text{s.t.} \quad & \mathbf{R}\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right) = 0 \\
& g_j\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right) \leq 0,\ j = 1, \ldots, N_g \\
& h(\mathbf{z}) \leq 0 \\
& \underline{z}_i \leq z_i \leq \bar{z}_i,\ i = 1, \ldots, N_z,
\end{aligned}
$$

Challenges

- Computationally intractable due to the many nonlinear constraints
- Requires 1+ N_g finite element solves at each major optimization iteration

Augmented Lagrangian Approach

$$
\begin{aligned}
\min_{\mathbf{z}\in\mathbb{R}^{N_{\mathbf{z}}}} \quad & \sum_{l=1}^{N_f} \alpha_l f_l\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right) + \frac{\beta}{N_g}\sum_{j=1}^{N_g}\left[\gamma_j^{(k)}\hat{g}_j\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right) + \frac{\mu_j^{(k)}}{2}\hat{g}_j\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right)^2\right] \\
\text{s.t.} \quad & \mathbf{R}\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right) = 0 \\
& \underline{z}_i \leq z_i \leq \bar{z}_i,\ i = 1, \ldots, N_z,
\end{aligned}
$$

where

$$
\hat{g}_j\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right) = \left(g_j\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right), -\frac{\gamma_j^{(k)}}{\mu_j^{(k)}}\right)
$$

$$
\gamma_j^{(k+1)} = \gamma_j^{(k)} + \mu_j^{(k)}\hat{g}_j\left(\mathbf{u}(\mathbf{z}), \theta(\mathbf{z}), \mathbf{z}\right)
$$

$$
\mu_j^{(k+1)} = \left(\alpha\mu_j^{(k)}, \mu_{max}\right),\ \alpha > 1
$$

Computationally tractable, requires 2 finite element solves at each major optimization iteration *k*

Example 3: Input Deck

Thermomechanical Compliance Criterion. Measures the weighted mechanical and thermal compliance. The optimizer must balance heat transfer performance and structural stiffness.

Local Constraint Criterion. Measures the local condition that the optimizer must satisfied. An augmented Lagrangian method is applied to handle many nonlinear constraints.

- Requires only one adjoint solve regardless of the number of nonlinear constraints

```
begin service 1
  code engine
  number_processors 1
  number_ranks 1
  update_problem true
  update_problem_frequency 1
end service

begin service 2
  code analyze
  number_processors 1
  number_ranks 1
  update_problem true
  update_problem_frequency 10
end service

begin criterion 1
  type thermomechanical_compliance
  thermal_weighting_factor 1.0
  mechanical_weighting_factor 1.0
end criterion

begin criterion 2
  type volume
end criterion

begin criterion 3
  type local_constraint
  limits 1e8
  local_measures vonmises
  initial_penalty 0.1
  minimum_ersatz_material_value 1e-6
end criterion
```

Thermomechanical Compliance Criterion

Local Constraint Criterion

Figure 3. Text snippet from the input deck used for the topology optimization study with local von Mises constraints.

Example 3: Main Output Data Files*

File	Description
morphorm_{type}_optimizer_diagnostics.txt	Description presented in Example 1
opt_criteria_history.csv	Description presented in Example 1
iteration{itr}_criteria_history.csv	Description presented in Example 1

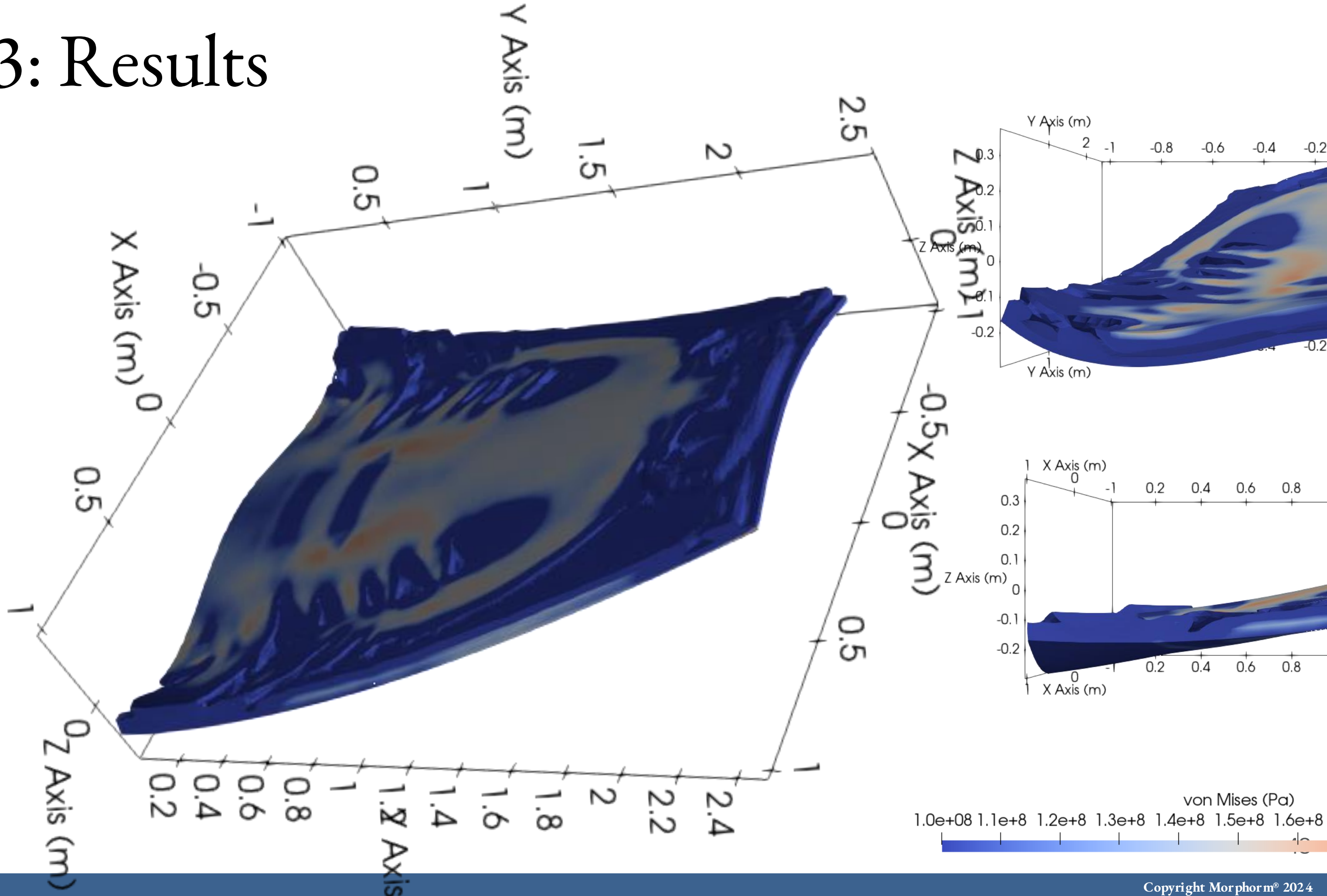
* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

Example 3: Main Output Exodus Files

File	Description
engine_output.exo	Description presented in Example 1
iteration{itr}.exo	Description presented in Example 1

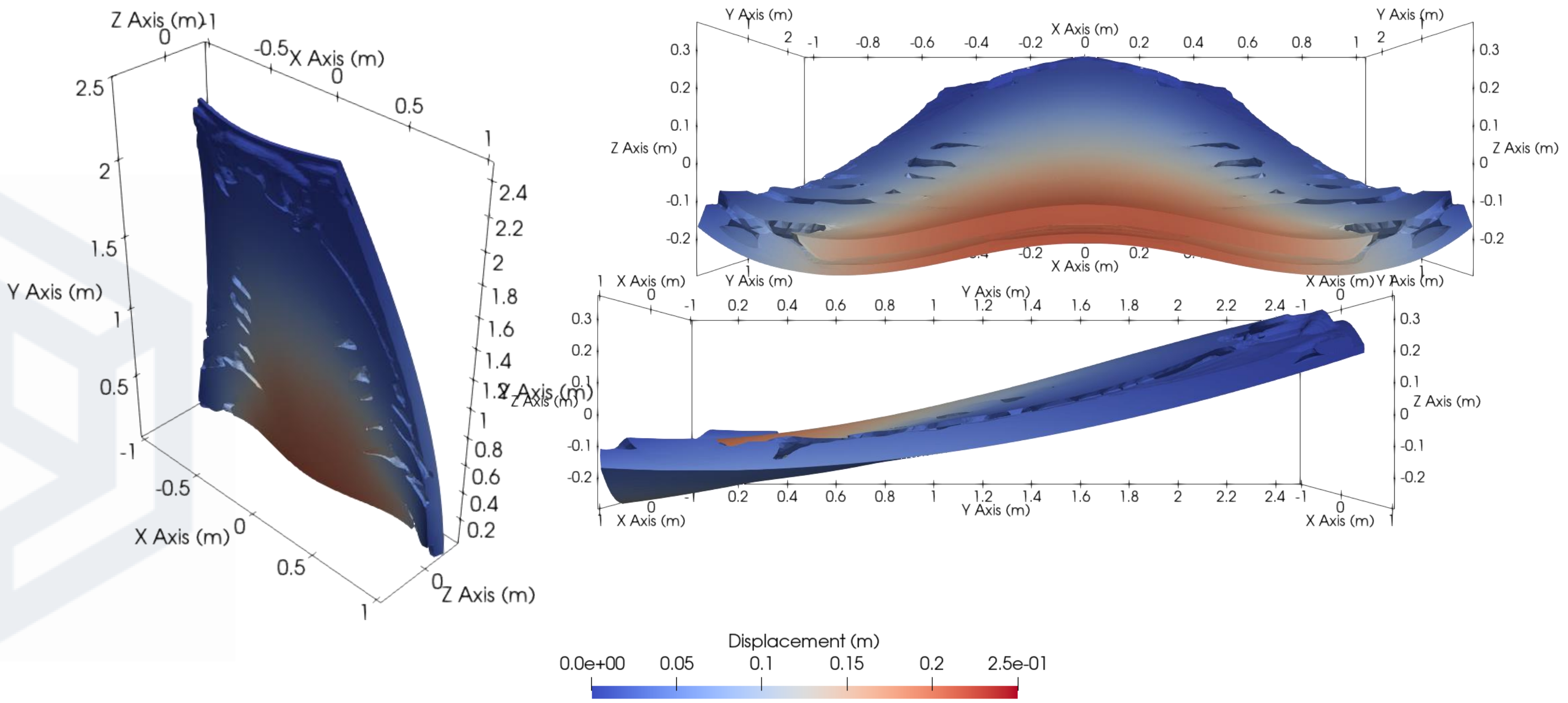
* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

Example 3: Results



Mass Reduction: 71%

Example 3: Results (cont.)



Example 4

Simulation-Based Design Optimization
Large Elastic Deformations

Example 4: Nonlinear Mechanical Physics*

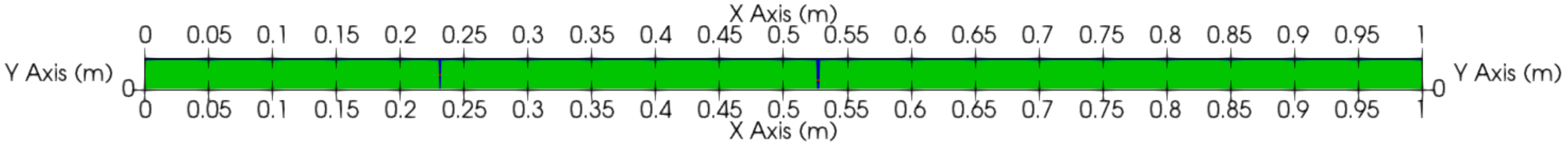


Figure 16. Initial design domain for the topology optimization study.

Problem Description. Find a lightweight structural frame such that the peak displacement is maximize and the mass budget is satisfied. The actuating forces and locations are considered known.

Material. The structure is made of a hyperelastic material:

- Shear Modulus: 122.18 GPa
- Bulk Modulus: 318.63 GPa

Model. The model is symmetric about the plane at X=1.0, only half of the model is considered for optimization.

* See Example 3.5.1 in the Examples Manual for details.

Example 4: Boundary Conditions

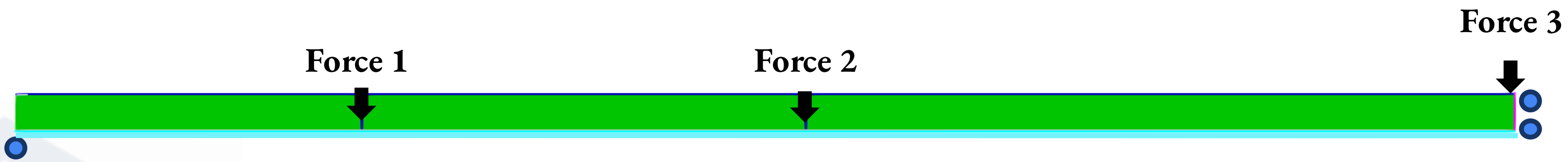


*Figure 17. Initial design domain for the topology optimization study. The Y displacement lower left-most corner (pinned node) is fixed. The X displacements on the edge described by the plane at X=1.0 (**pink surface**) are fixed. The bottom edge (**cyan edge**) and the top edge (**blue edge**) are a non-optimizable regions. The optimizer will seek to match the deformations at the bottom edge (**cyan edge**). The location of the actuating forces are described by the **black arrows**. The actuating forces are assumed known for the topology optimization study.*

Force 1	Force 2	Force 3
F_y = -500 N	F_y = -1000 N	F_y = -2500 N

Example 4: Formulation Overview

Topology Optimization

Goal: Maximize Elastic Deformations

$$\min_{\mathbf{z}\in\mathbb{R}^{N_z}} \quad \alpha f(\mathbf{u}(\mathbf{z}),\mathbf{z}) \longleftarrow \boxed{\text{Elastic Energy Criterion}}$$

$$\text{s.t.} \quad \mathbf{R}(\mathbf{u}(\mathbf{z}),\mathbf{z}) = 0 \longleftarrow \boxed{\text{Nonlinear Residual Equation}}$$

$$\boxed{\text{Mass Constraint}} \longrightarrow h(\mathbf{z}) \leq 0$$

$$\underline{z}_i \leq z_i \leq \bar{z}_i,\ i = 1,\dots,N_z \longleftarrow \boxed{\text{Geometry parameters}}$$

Example 4: Input Deck*

Objective: Elastic Energy Criterion. Measures the internal elastic energy stored in a body due to external forces.

Constraint: Mass Budget. Measures the satisfaction of the mass budget requirement.

```
begin scenario 1
   physics steady_state_nonlinear_mechanics
   service 2
   dimensions 2
   material 1
   loads 1 2 3
   blocks 1 2 3 4
   boundary_conditions 1 2
   newton_raphson 1
   linear_solver 1
   output 1
   material_penalty_exponent 2.0
   minimum_ersatz_material_value 1e-3
end scenario
```

* See Examples Manual for an in-depth discussion about Example 4 (Nonlinear Mechanics Section - 3.5.1)

Example 4: Main Output Data Files*

File	Description
morphorm_{type}_optimizer_diagnostics.txt	Description presented in Example 1
opt_criteria_history.csv	Description presented in Example 1
iteration{itr}_criteria_history.csv	Description presented in Example 1

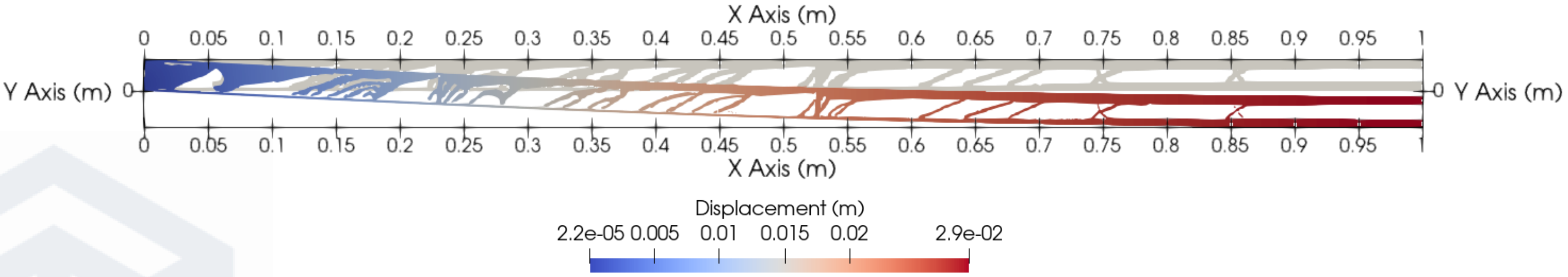
* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

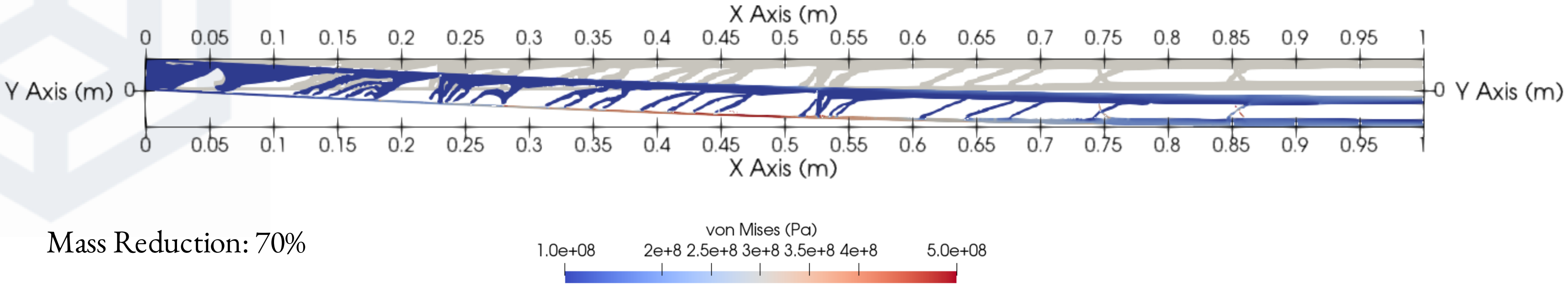
Example 4: Main Output Exodus Files

File	Description
engine_output.exo	Description presented in Example 1
iteration{itr}.exo	Description presented in Example 1

* See User Manual for an in-depth discussion about the output data files (Chapter 3 - Diagnostics).

Example 4: Results





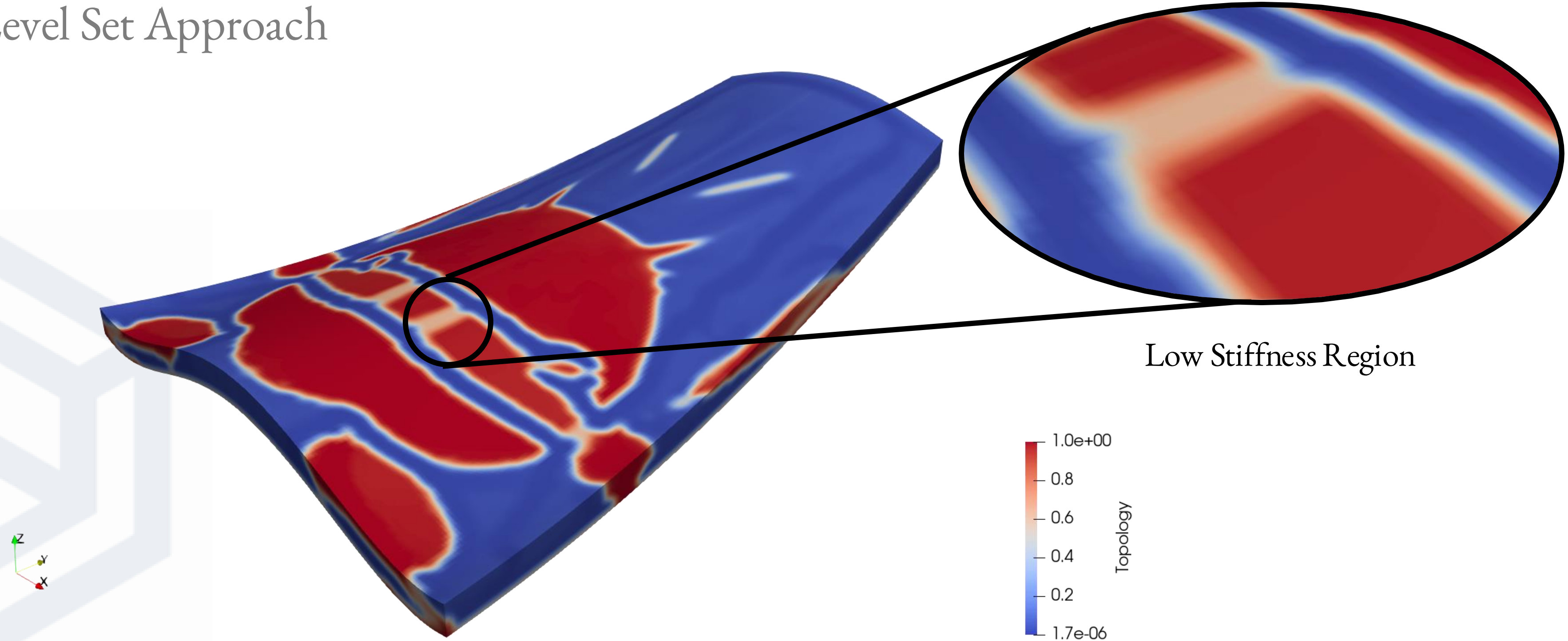
Mass Reduction: 70%

Technology Enhancements

Simulation-Based Design Optimization

Topology Optimization

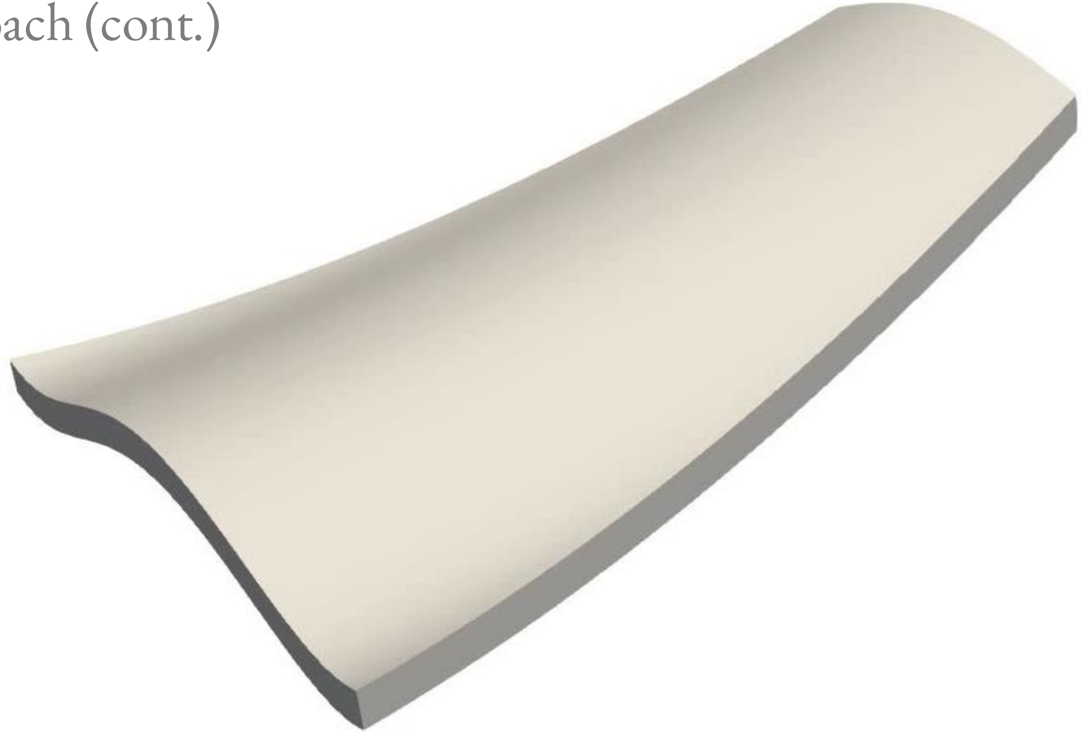
Level Set Approach



Density-based topology optimization methods can cause large spurious deformations due to low stiffness regions. These spurious deformations may trigger premature solver failures.

Topology Optimization

Level Set Approach (cont.)

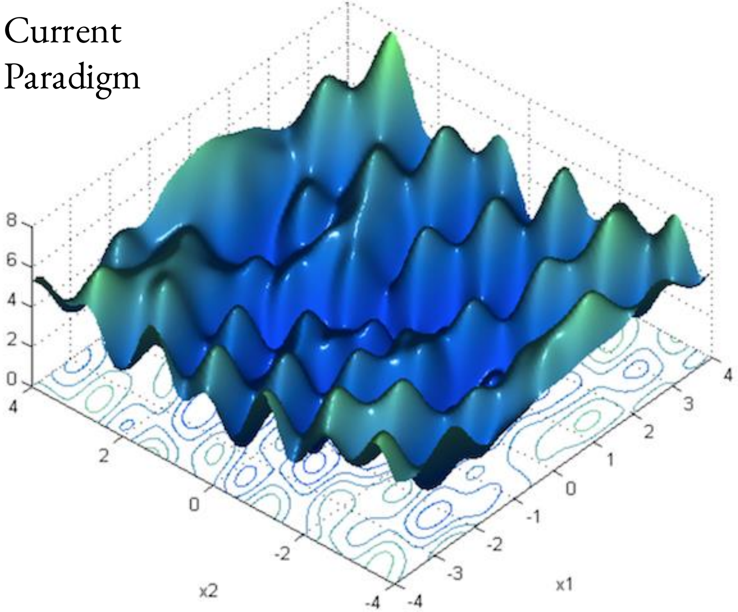


Level-set optimization methods always work with the true geometry at each optimization iteration.
Thus, bypassing the spurious deformations caused by low stiffness regions.

Shape Matching

Formulation Enhancements

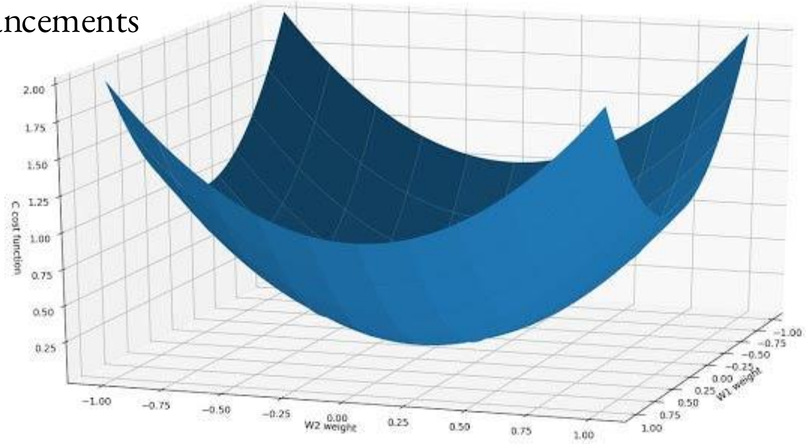
Current Paradigm



Shape matching criterion is highly nonlinear

- Data transforms can smooth oscillations and improve performance

Proposed Enhancements



Apply Dirichlet-Neumann inverse formulation to smooth oscillations*

- Dirichlet-Neumann criterion is convex

*Aguiló, M.A., 2011. Inverse strategies for characterization of material properties. Cornell University.

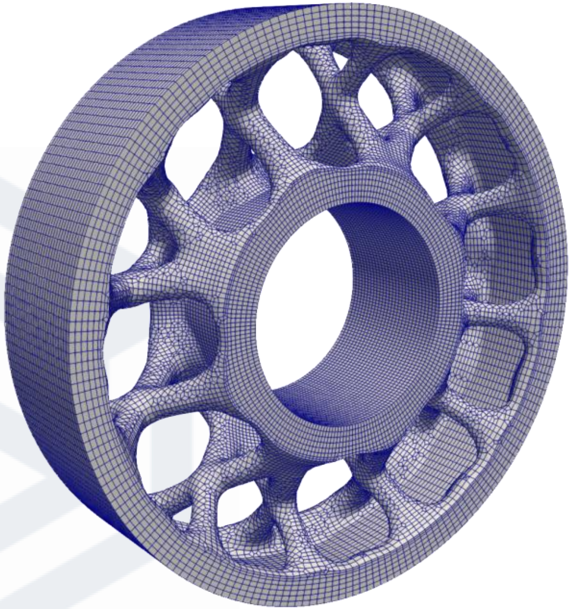
Maturation

Usability Enhancements

- Support import and export operations for commercial models
  - Example: Abaqus input files
- Enhancements to multi-level optimizer
  - Support surrogate-based optimization with constraints
  - Remove script generation burden from user
- Expose more Dakota methods through the Morphorm input deck
  - Example: Design under uncertainty
- Graphical User Interface
  - Update: Development in progress
- Parallel GPU solver
  - Optimize settings to improve solver performance (Working with SNL)

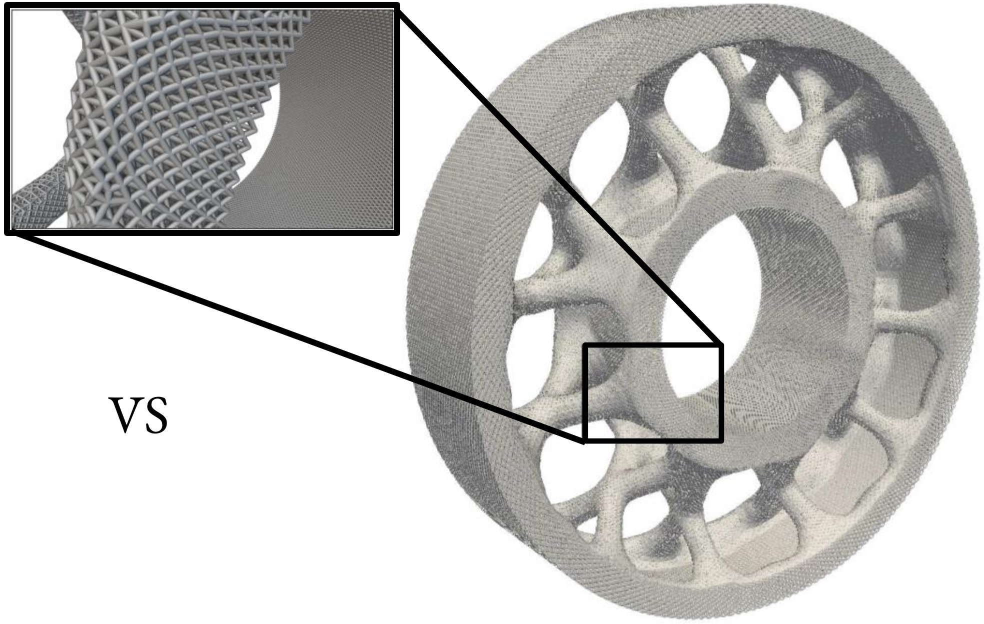
Topology Optimization

Spatially-Varying Lattices



Conformal
Topology Optimization

VS



Lattice-Based
Topology Optimization

Thank you

Simulation-Based Design Optimization