AFRL Morph X

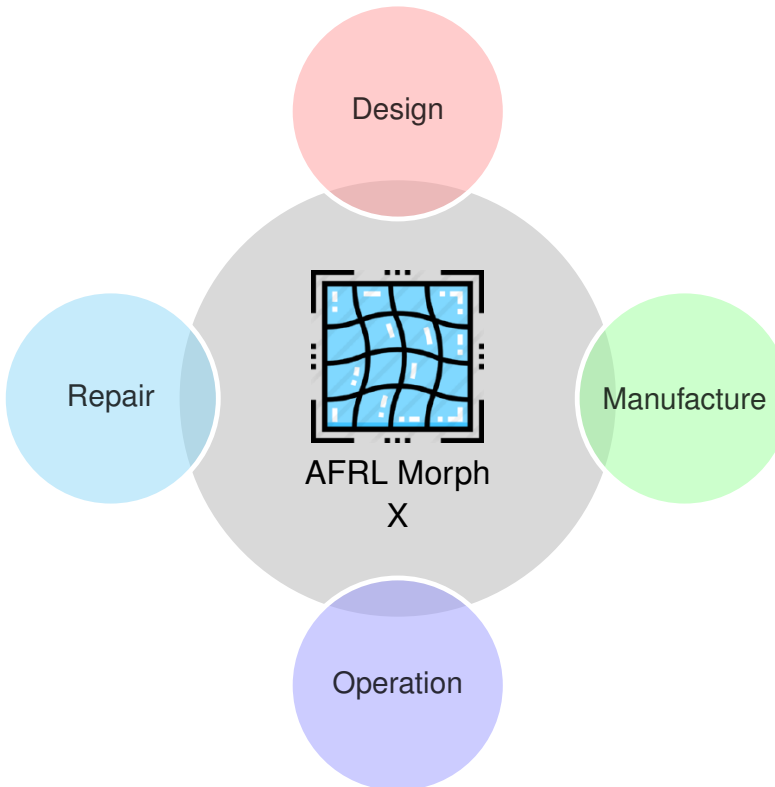
JEFF BROWN
ALEX KASZYNSKI

ENGINE INTEGRITY BRANCH
TURBINE ENGINE DIVSION
AEROSPACE SYSTEMS DIRECTORATE
FEBRUARY 26, 2021

AFRL Morph X

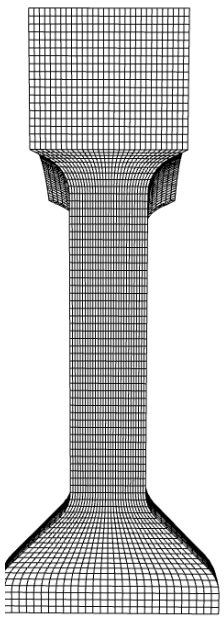
3D Mesh Morphing Software

- Move 3D FEM and CFD meshes to point cloud and surface data
- Save design time by removing mesh generation bottlenecks
- Rapidly assess impacts of as-manufactured geometry on design intent
- Learn the effects of operational damage on continued operation
- Design and validate the suitability of component repair
- And more...

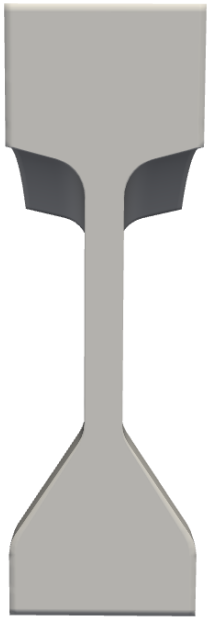


Design with AFRL Morph X

Update FEM to Parametrically Variable CAD Surface

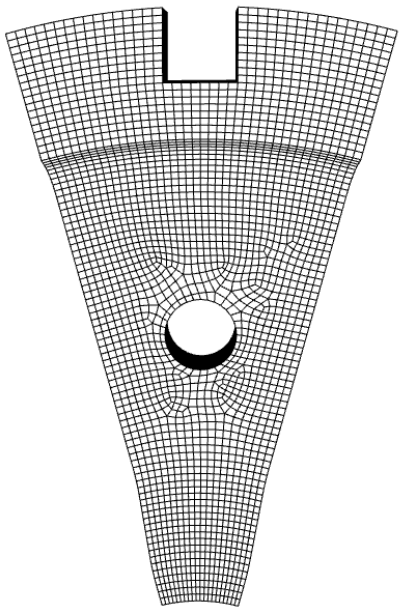


FEM



Target CAD

Side View



FEM

Target CAD

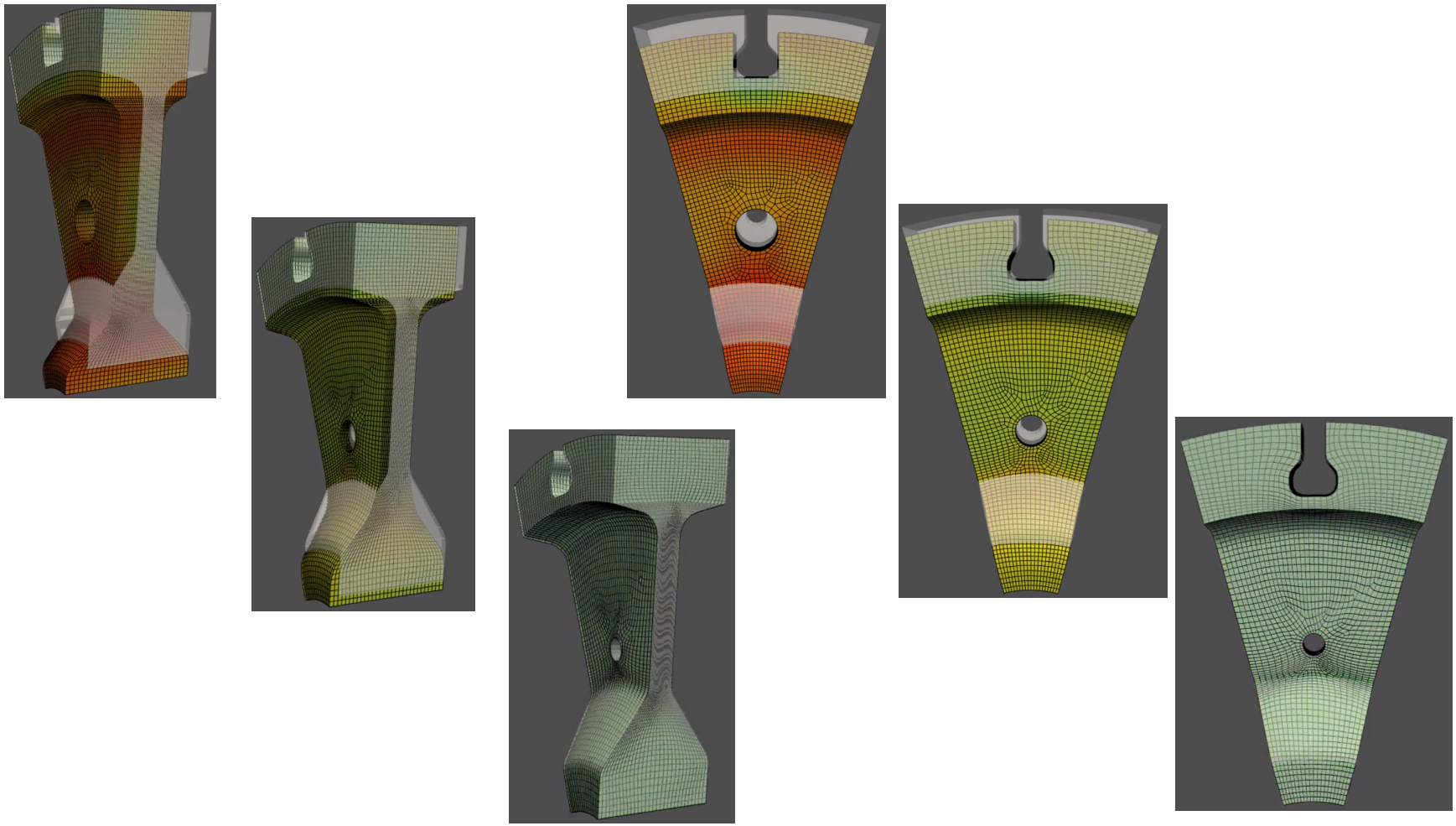
Front View

Models plotted in same reference frame. FEM grows radially.

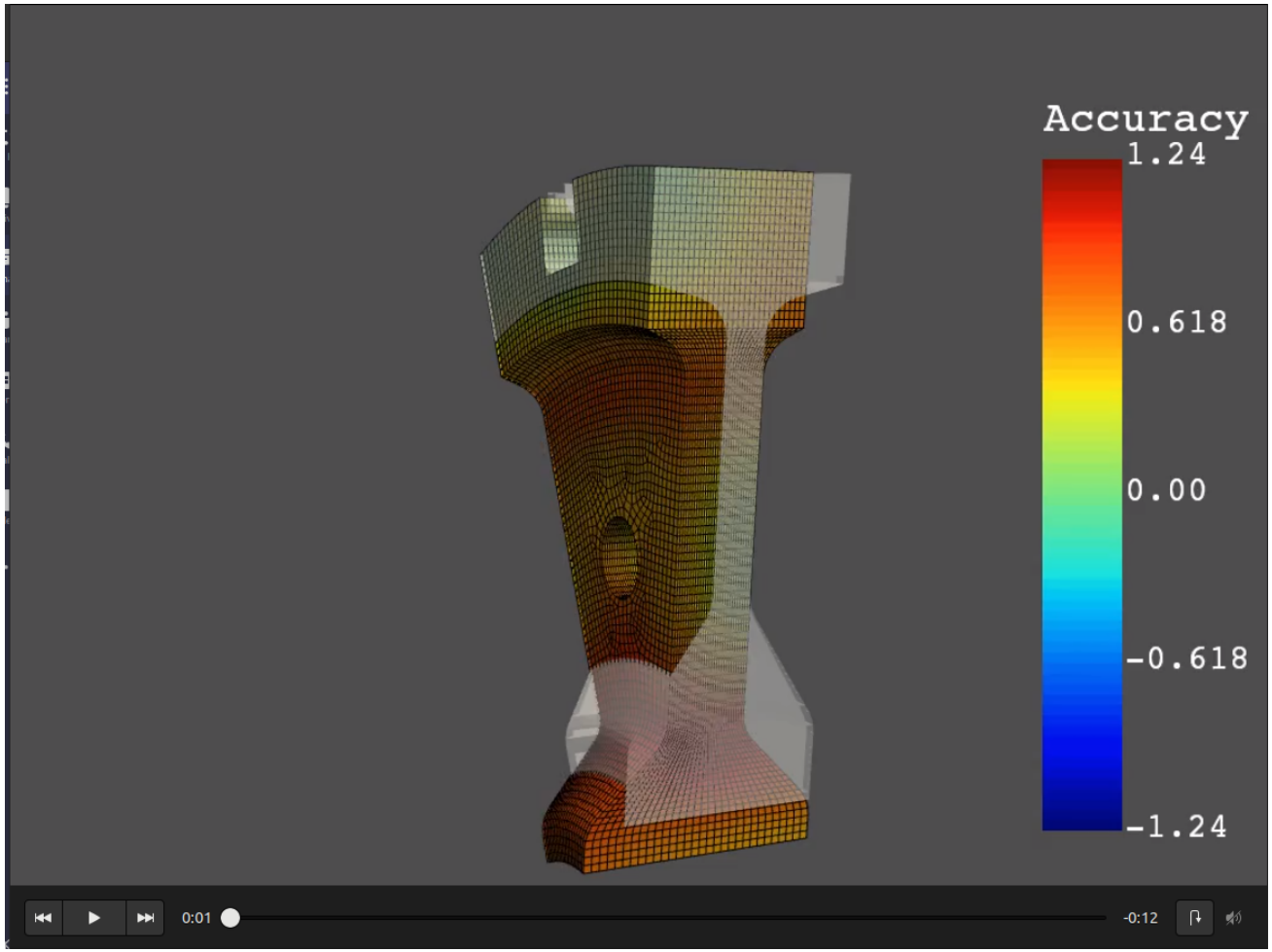
Fourteen variable geometry parameters.

Design with AFRL Morph X

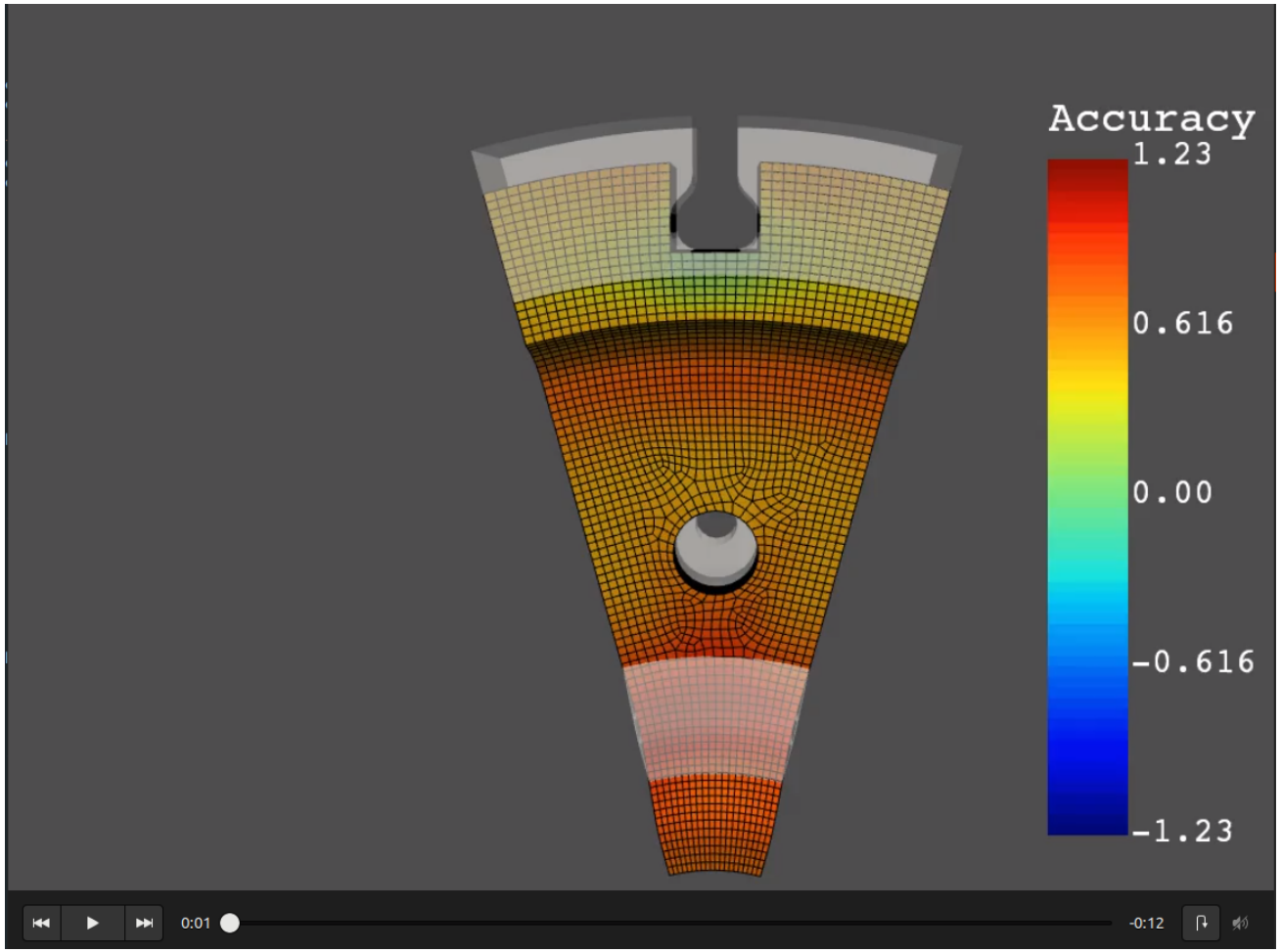
Update FEM to Parametrically Variable CAD Surface



Animation - Click image to Play

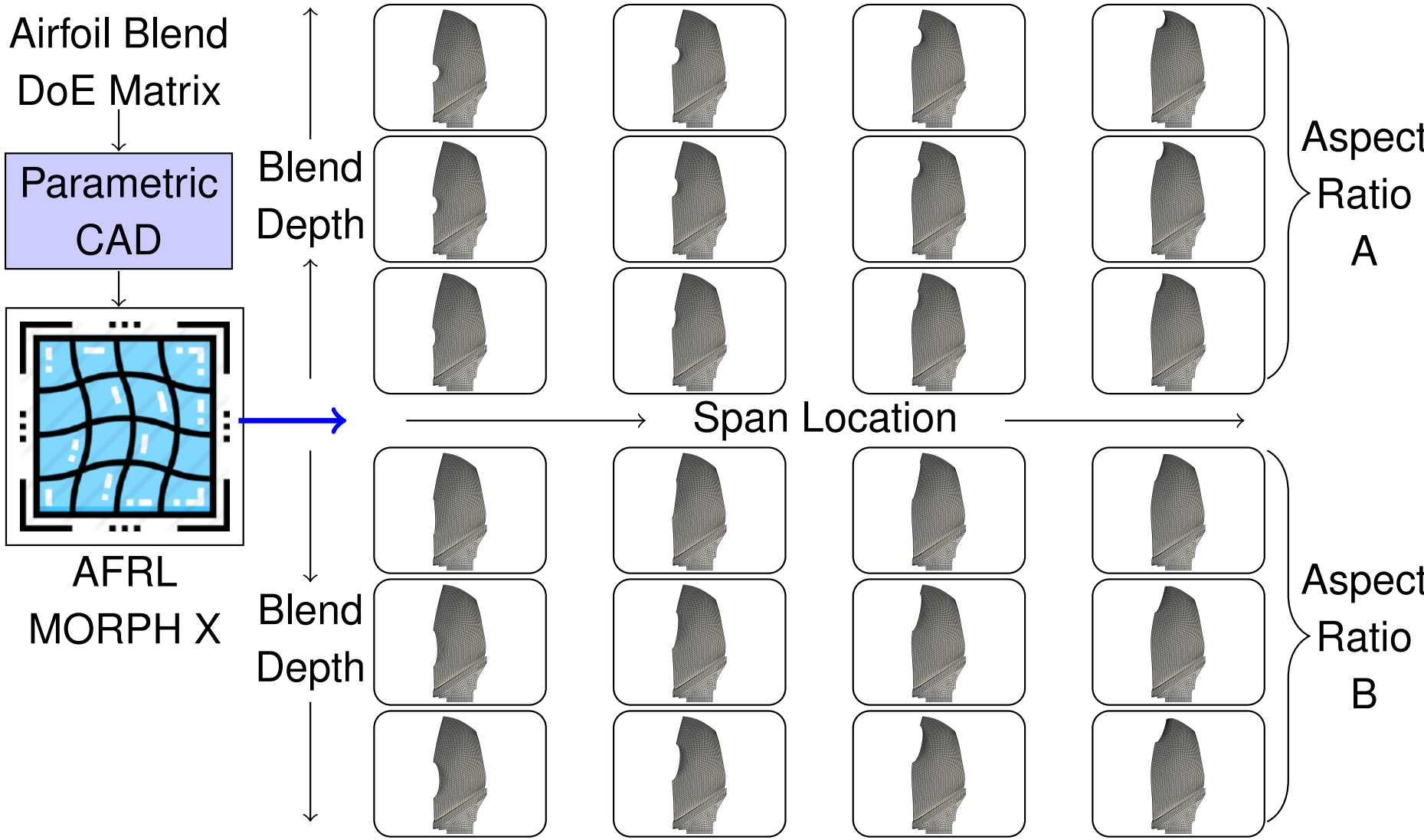


Animation - Click image to Play



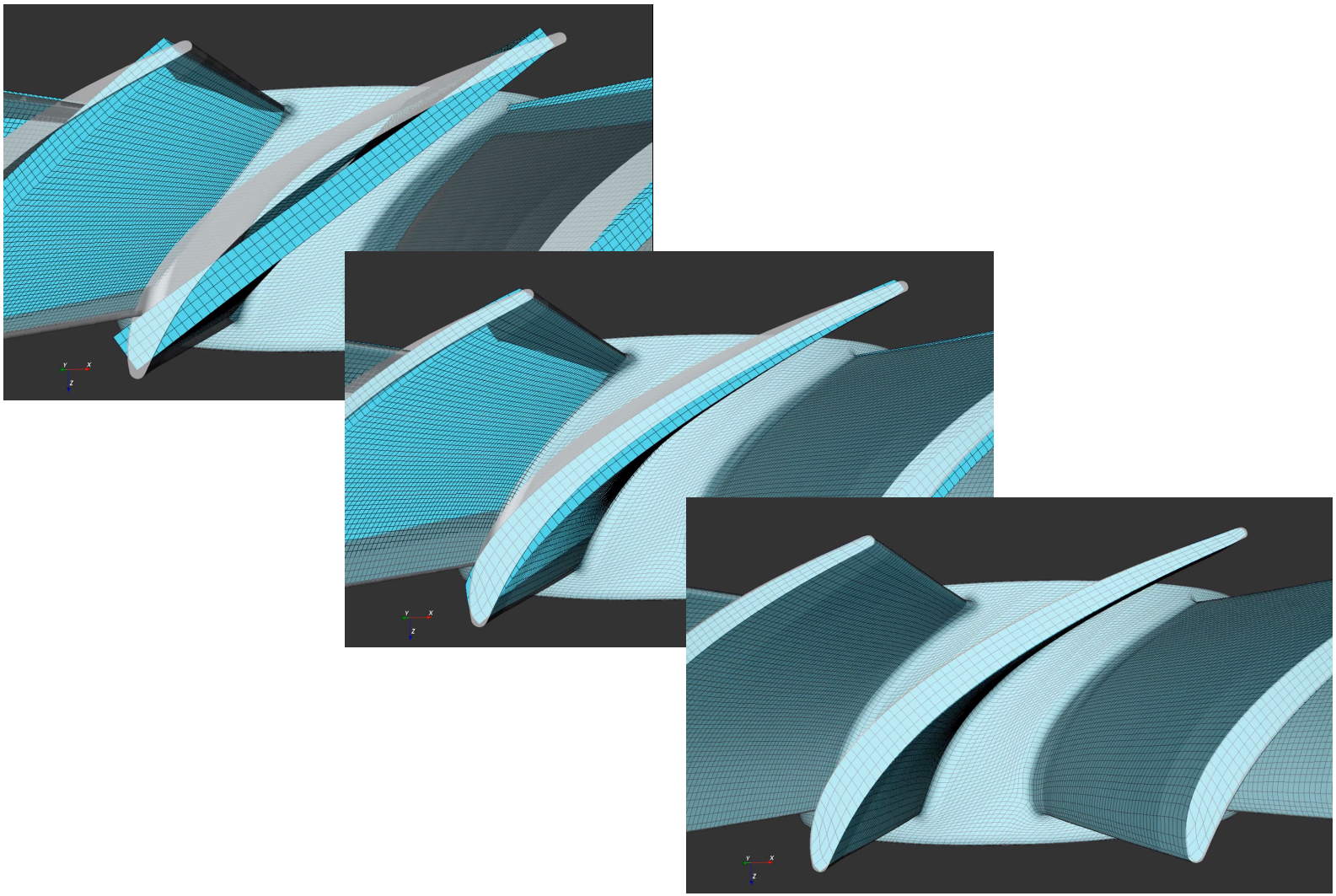
Design with AFRL Morph X

Sample Large Design Spaces Without Remeshing Challenges

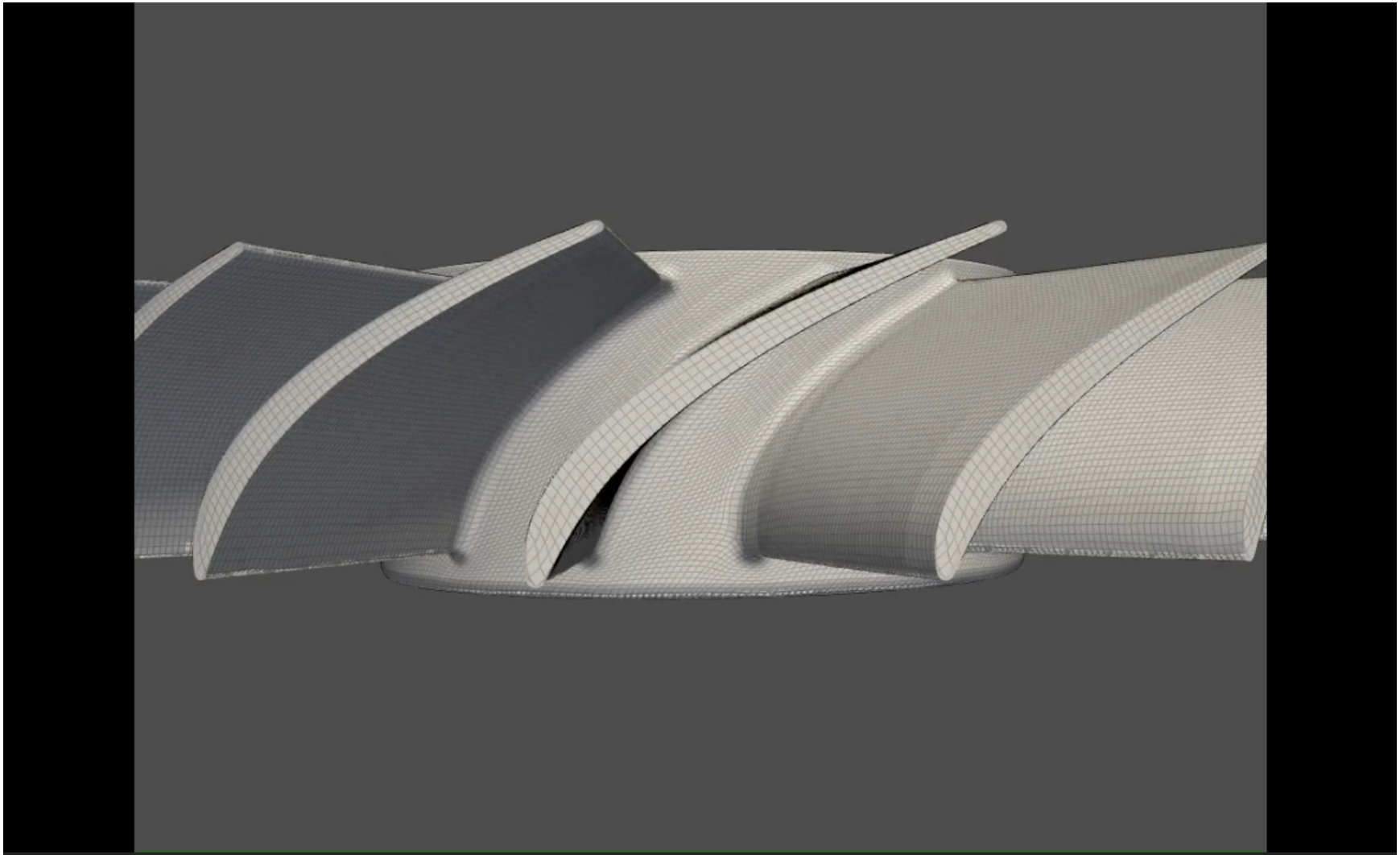


Design with AFRL Morph X

Simplify Hex Mesh Generation - Morph defeatured FEM to Complex Surface

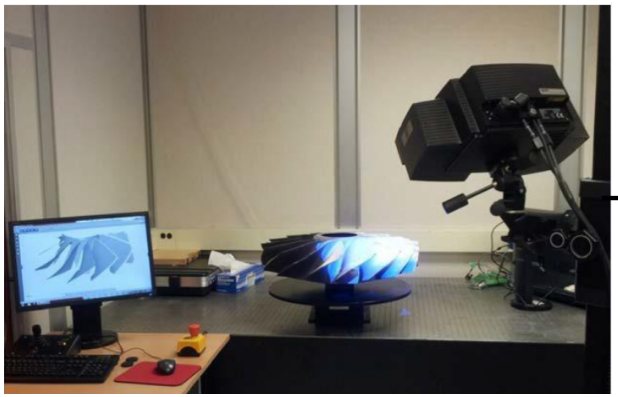


Animation - Click image to Play

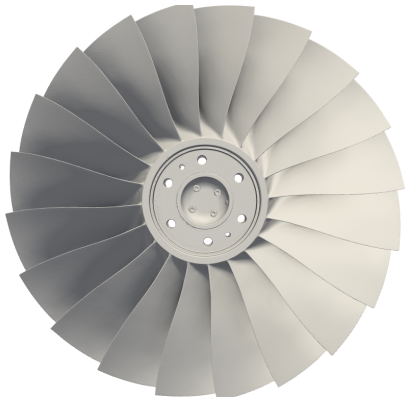


Manufacturing with AFRL Morph X

It starts with measuring variation of a real part and comparing to Design Intent

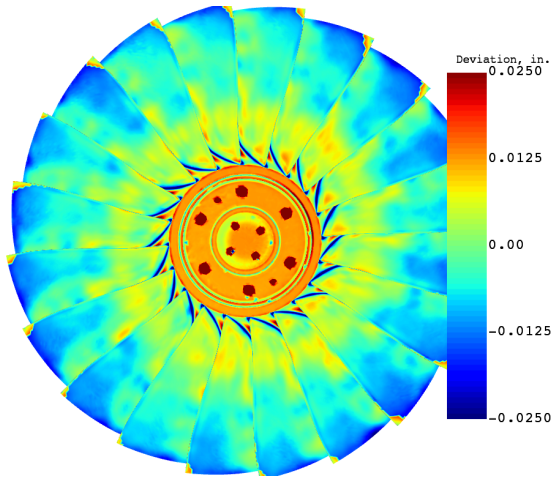


Geometry Measurement



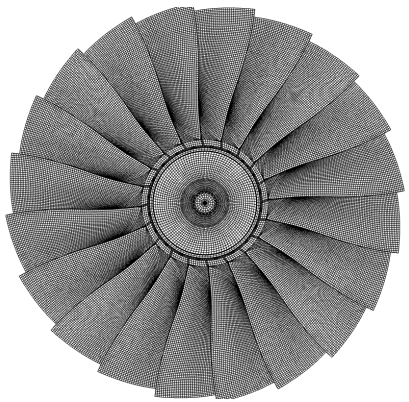
STL Surface

Manufacturing Deviations



Ansys

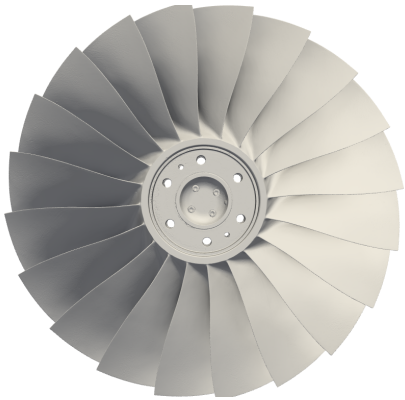
FEM Software



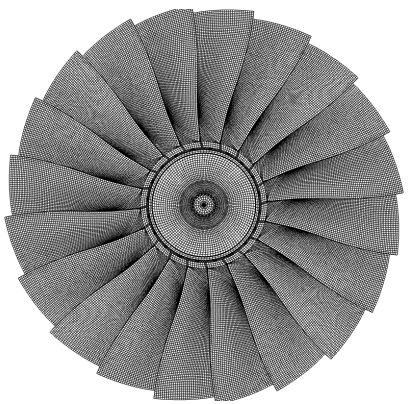
Design FEM

Manufacturing with AFRL Morph X

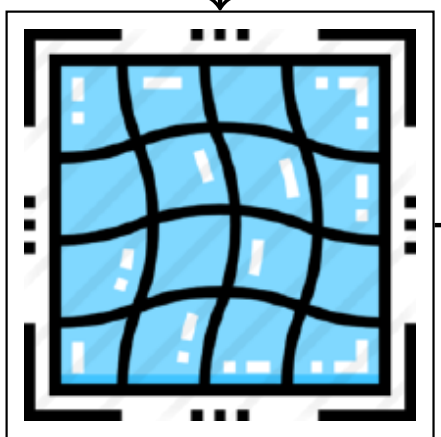
AFRL Morph X updates the Design FEM to match the STL surface within 0.001”



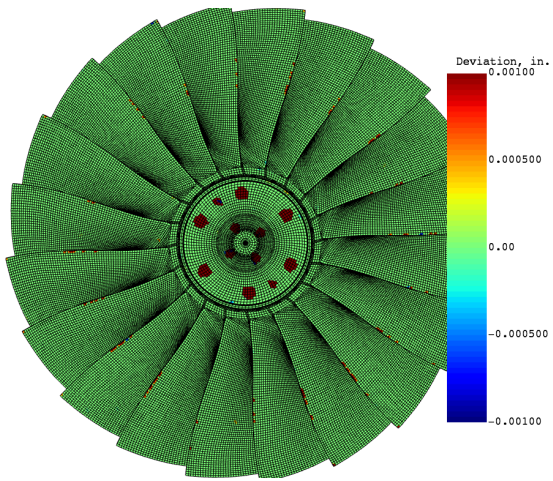
STL Surface



Design FEM

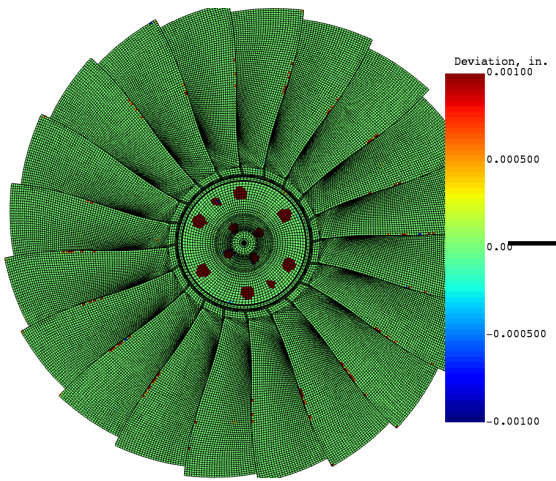


Morphed FEM
Deviations from STL



Manufacturing with AFRL Morph X

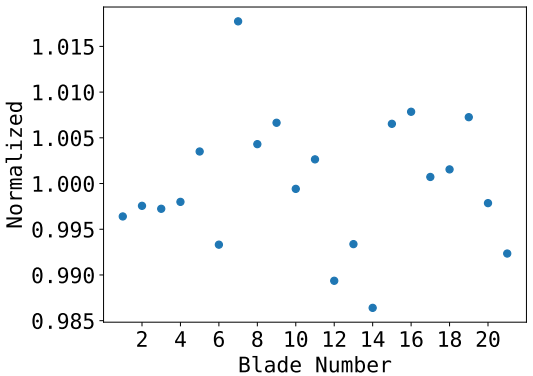
Morphed FEM is a Computational Replica of Physical Part



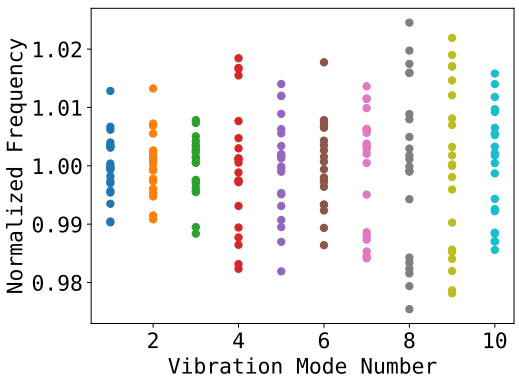
Morphed FEM
Computational Replica

Ansys

FEM Software

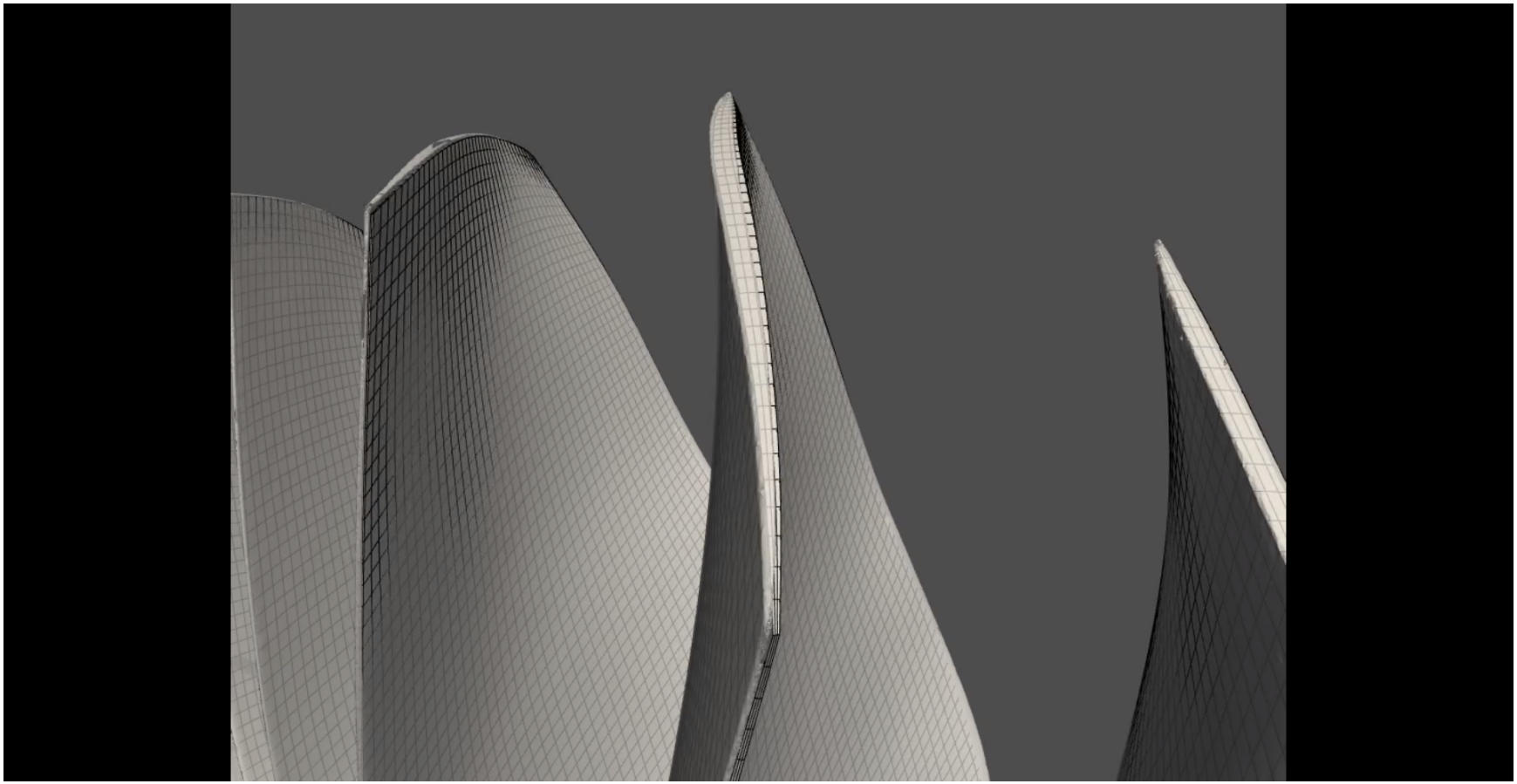


Blade to Blade
Variation



Rotor to Rotor
Variation

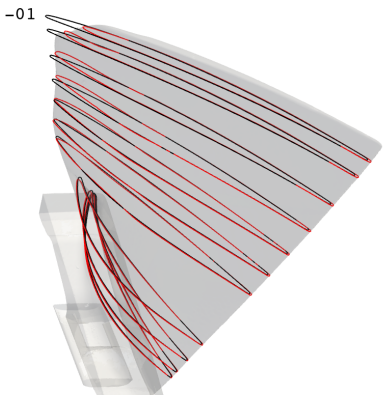
Animation - Click image to Play



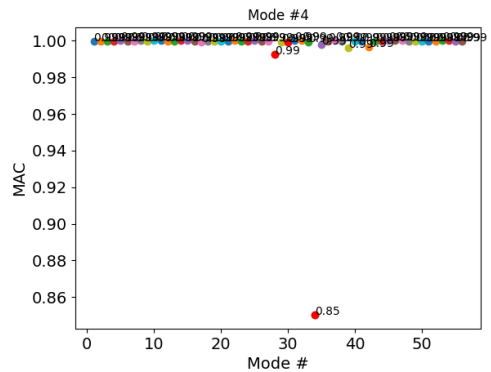
Operations with AFRL Morph X

Integrating with Maintenance, Repair, and Operations with Morph X

Airfoil Leading Edge Blend Repair

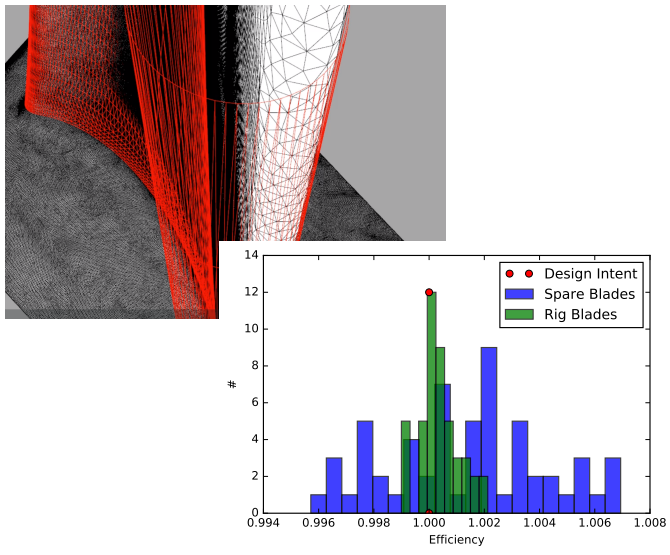


Assess Damage

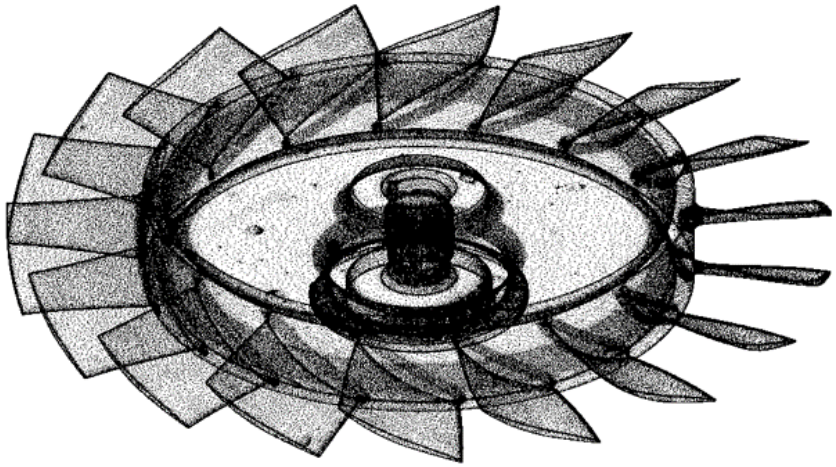


Safety Review Board

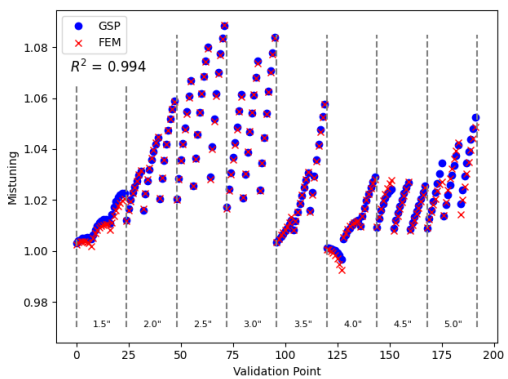
More with AFRL Morph X



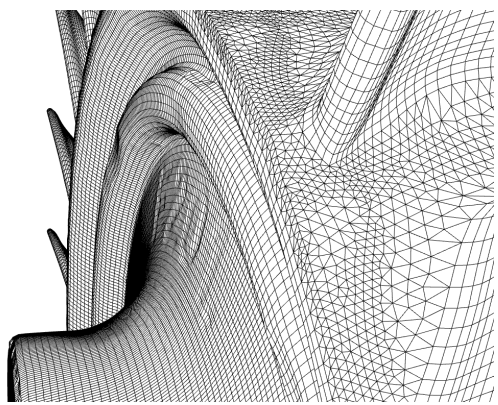
Turbine CFD Mesh Morphing



First Article Analysis

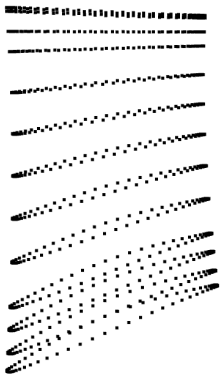


ML/AI Training Data Generator



Additive Part Models

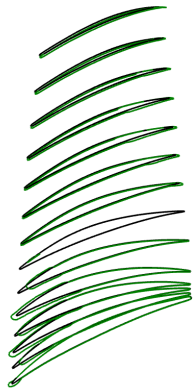
Even more with AFRL Morph X



CMM Data Capable

Improved Instrumentation

Improved Model Validation

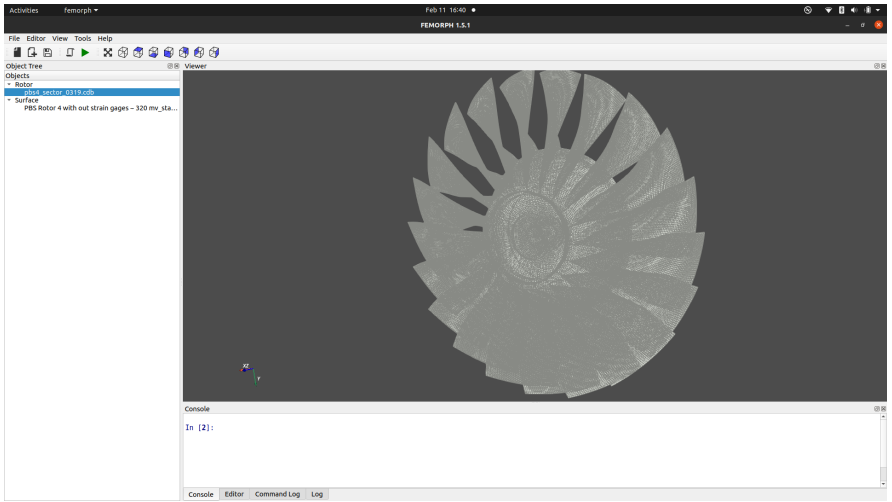


Adaptive Toolpath Generation

AFRL Design X Interface

Scriptable for integration in automated workflows and GUI for user experience

```
File  Edit  Options  Buffers  Tools  Python  Virtual Envs  Elpy  YASnippet
 1 import femorph
 2 import femorph_blender
 3
 4 jetcat_surf = 'jetcat_scan.ply'
 5 jetcat_cdb = 'jetcat_sector.cdb'
 6
 7 surf = femorph.Surface(jetcat_surf)
 8 fem = femorph.Rotor(jetcat_cdb)
 9 fem.replicate_cyclically()
10 fem.align(surf)
11 fem.morph(surf, settings=set_morph)
12 fem.write_cyclic_sectors()
13
14 blend = femorph_blender.Blender(fem)
15 blend.define_edge('LEAD_EDGE_NODE')
16 span = 2.5
17 aspect = 2.0
18 depth = 0.2
19 blend.blend_on_edge(spans, aspect, depth, 0.0, 0.0, 100)
20 blend.write_archive()
```



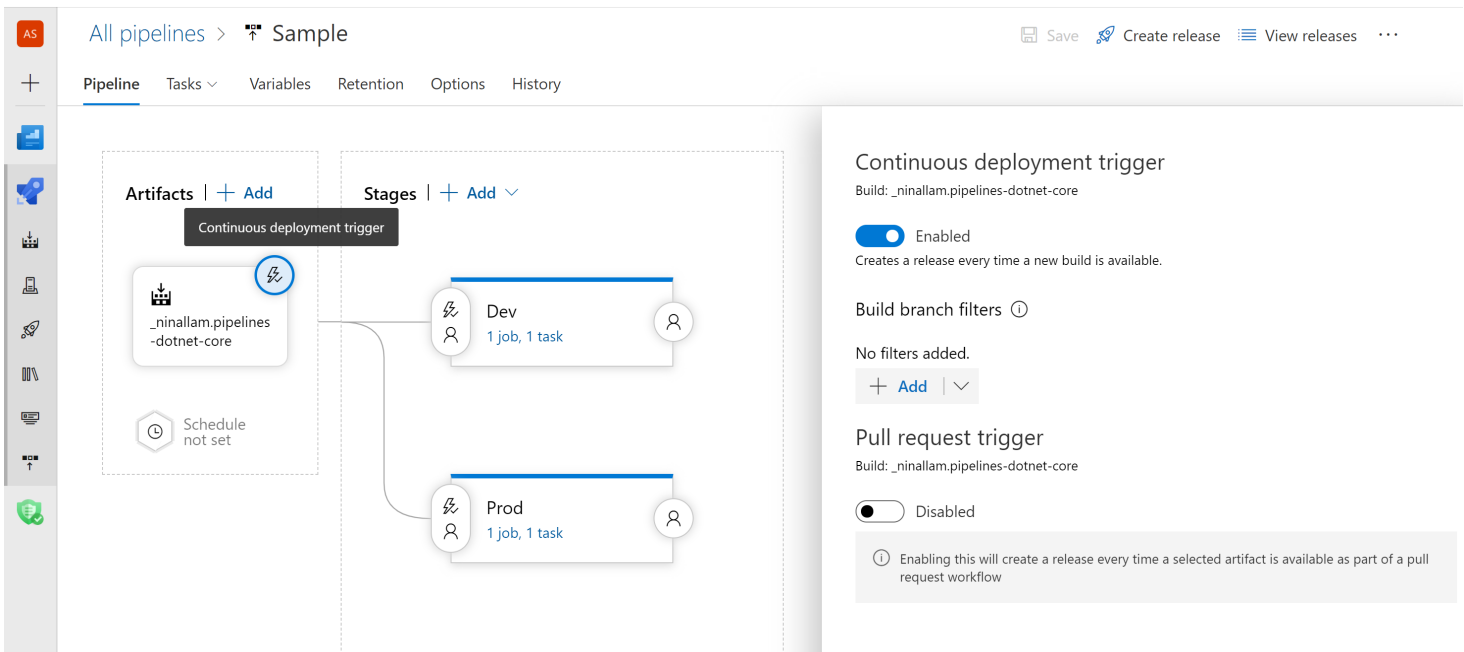
AFRL Design X Background

- Developed in Python, C, and C++ for 8+ years
- Accessible through GUI and Python API, Linux, Windows, and Mac OS
- Beta tested by Pratt & Whitney for two years
- Operating in year 4 of 10 year non-exclusive license with Pratt & Whitney
- P&W provides licensing fees for use and support
- Utilized by large P&W user base for wide range of applications
- Applied pervasively in AFRL research activities

AFRL Design X Background

Software Development

- Software tested rigorously in continuous development and deployment (CI/CD) pipelines
- Leverages automated testing in a secure environment
- Enables high code coverage and stability testing through regression testing.



AFRL Design X Background

Software Testing

- AFRL Design X follows standard software development practices.
- Unit and regression tests ensure new features are added without breaking existing ones.
- Code coverage of $\geq$ 95% required for all modules, and $\geq$ 98% for core modules

```
tests/test_surface.py::test_surface_blend SKIPPED (Requires femorph_blender)    [ 91%]
tests/test_surface.py::test_boolean PASSED                                       [ 92%]
tests/test_surface.py::test_load_g3d PASSED                                      [ 92%]
tests/test_tree.py::test1d PASSED                                                [ 93%]
tests/test_tree.py::test3d PASSED                                                [ 93%]
tests/test_tree.py::test3d_float32 PASSED                                        [ 94%]
tests/test_tree.py::test3d_float32_mismatch PASSED                               [ 94%]
tests/test_tree.py::test3d_float32_mismatch2 PASSED                              [ 95%]
tests/test_tree.py::test3d_8n PASSED                                             [ 95%]
tests/test_tree.py::test3d_8n_ub PASSED                                          [ 96%]
tests/test_tree.py::test3d_8n_ub_leaf20 PASSED                                   [ 96%]
tests/test_tree.py::test3d_8n_ub_eps PASSED                                      [ 97%]
tests/test_tree.py::test3d_large_query PASSED                                    [ 97%]
tests/test_tree.py::test_scipy_comp PASSED                                       [ 98%]
tests/test_tree.py::test1d_mask PASSED                                           [ 98%]
tests/test_tree.py::test1d_all_masked PASSED                                     [ 99%]
tests/test_tree.py::test3d_mask PASSED                                           [ 99%]
tests/test_tree.py::test6d PASSED                                                [100%]

====================== 210 passed, 2 skipped in 88.92s (0:01:28) ======================
```

AFRL Design X Background

Software Design: Modularity

- Software is a non-monolithic, modular, and composed of several packages. Includes both GUI and batch packages.
- Designed to be expandable, adaptive, and compatible with modern Python 3.6 - 3.8 environments on Windows, Mac, and Linux.
- Installable from *pip* via offline wheelhouse from GitHub releases.
- Compatible with standalone virtual environments, Jupyterlab webpages, or standard Python.

Closing Remarks

- AFRL MORPH X adds value across the life cycle, from design, to manufacture, and through operations
- Rigorous software development practices have enabled rapid development and successful deployment
- Windows demo software available and licensing opportunities


AFRL