Ansys

Powering Innovation That Drives Human Advancement

Adaptive Geometry Templates in Ansys Motor-CAD

Jonathan Godbehere, PhD

©2024 ANSYS, Inc.

Agenda

- Motor-CAD Templates and geometry customization
- Introduction to Ansys Motor-CAD Adaptive Templates
- Adaptive Template Creation
- Workflows into other Ansys products
- Getting started
- Summary

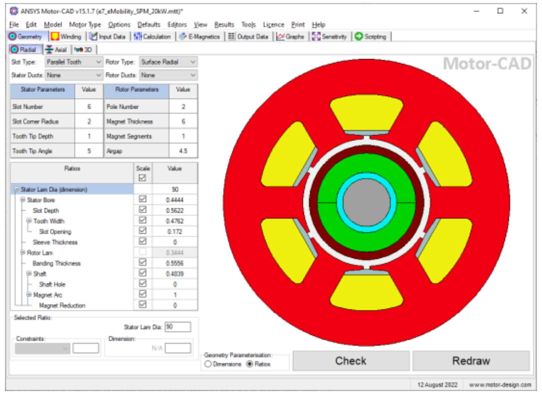
©2024 ANSYS, Inc.
Powering Innovation That Drives Human Advancement
Ansys

Ansys Motor-CAD Geometry Templates and Customizations

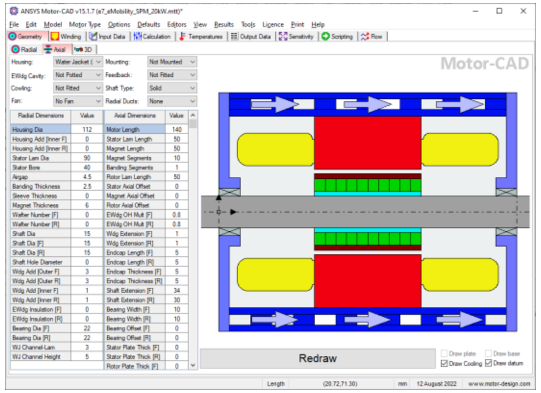
©2024 ANSYS, Inc.

Geometry Templates – Advantages

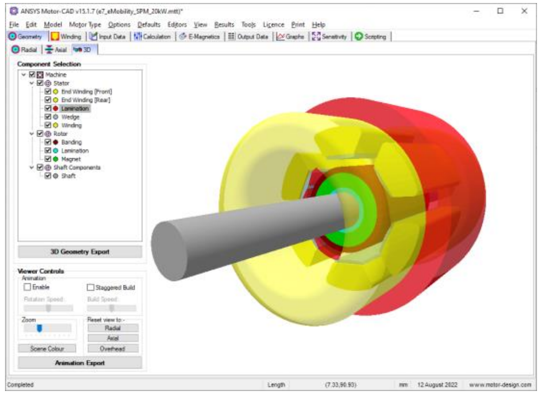
- Large range of parameterized templates: motor/rotor/slot type, rotor/stator ducts…
- Parameterization by dimensions and or by ratio for efficient design space exploration.
- Comprehensive model setup:
- Automated export into Ansys optiSLang for optimization



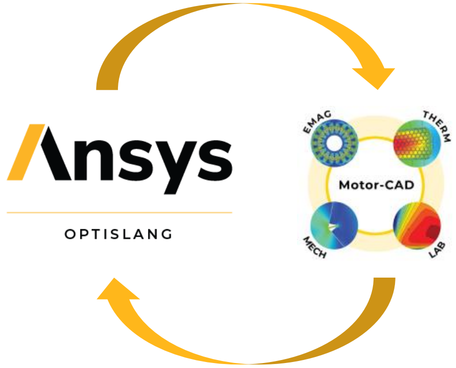
Radial cross section



Cooling Geometry



3D visualization

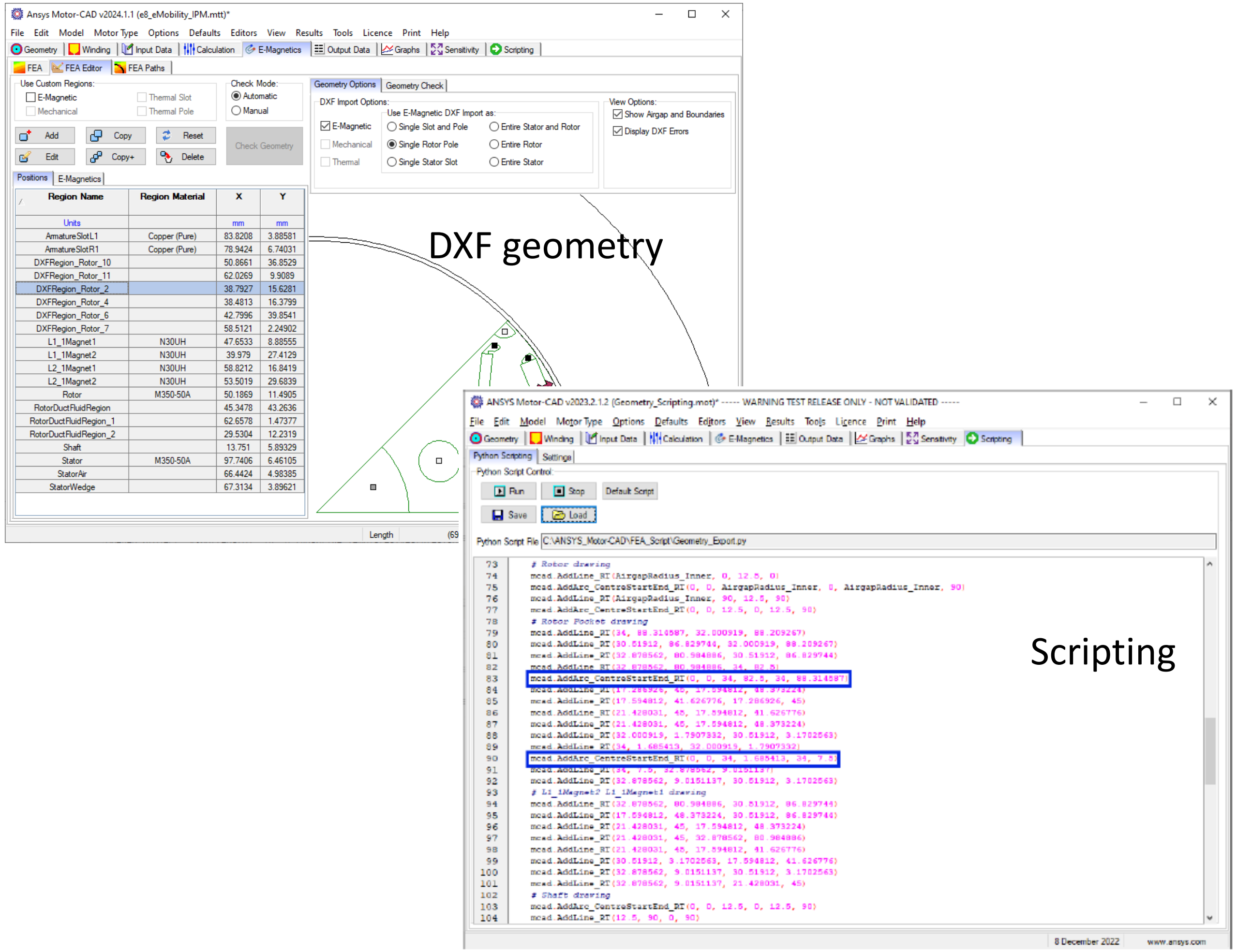


Optimization

©2024 ANSYS, Inc.

Options for Customization in Ansys Motor-CAD 2023 R2 – Part 1

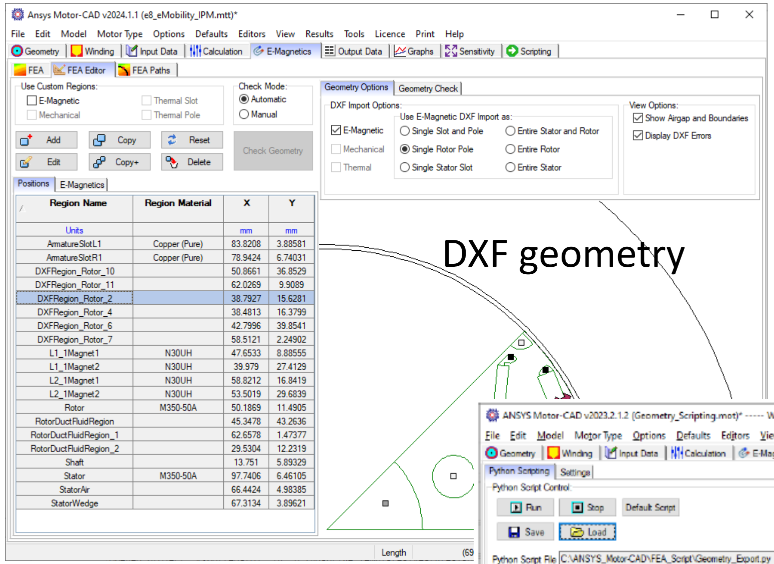
- Engineers wish to customize the geometry as a design progresses:
  - To maximize multiphysics performance
  - To capture manufacturing requirements
  - To optimize the design
- Two options were possible in previous versions of Motor-CAD:
  - DXF drawing import
  - Limited internal/external scripting.



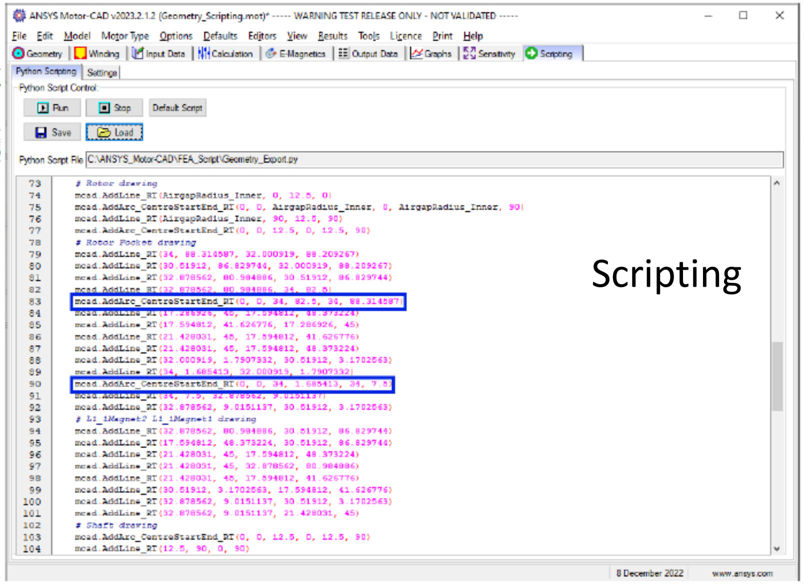
©2024 ANSYS, Inc.

Options for Customization in Ansys Motor-CAD 2023 R2 – Part 2

- Some engineers have created automated workflows in the past:
  - Create a .dxf drawing in a 3rd party tool
  - Automating the creation of a drawing file was often very complex
  - New geometry templates would need to be created from first principles
  - Old internal scripting was very limited
- Therefore, this kind of workflow was rare with many E-machine designers instead:
  - Relying on drawing software to create their customizations
  - No automation means no scalability!



DXF geometry



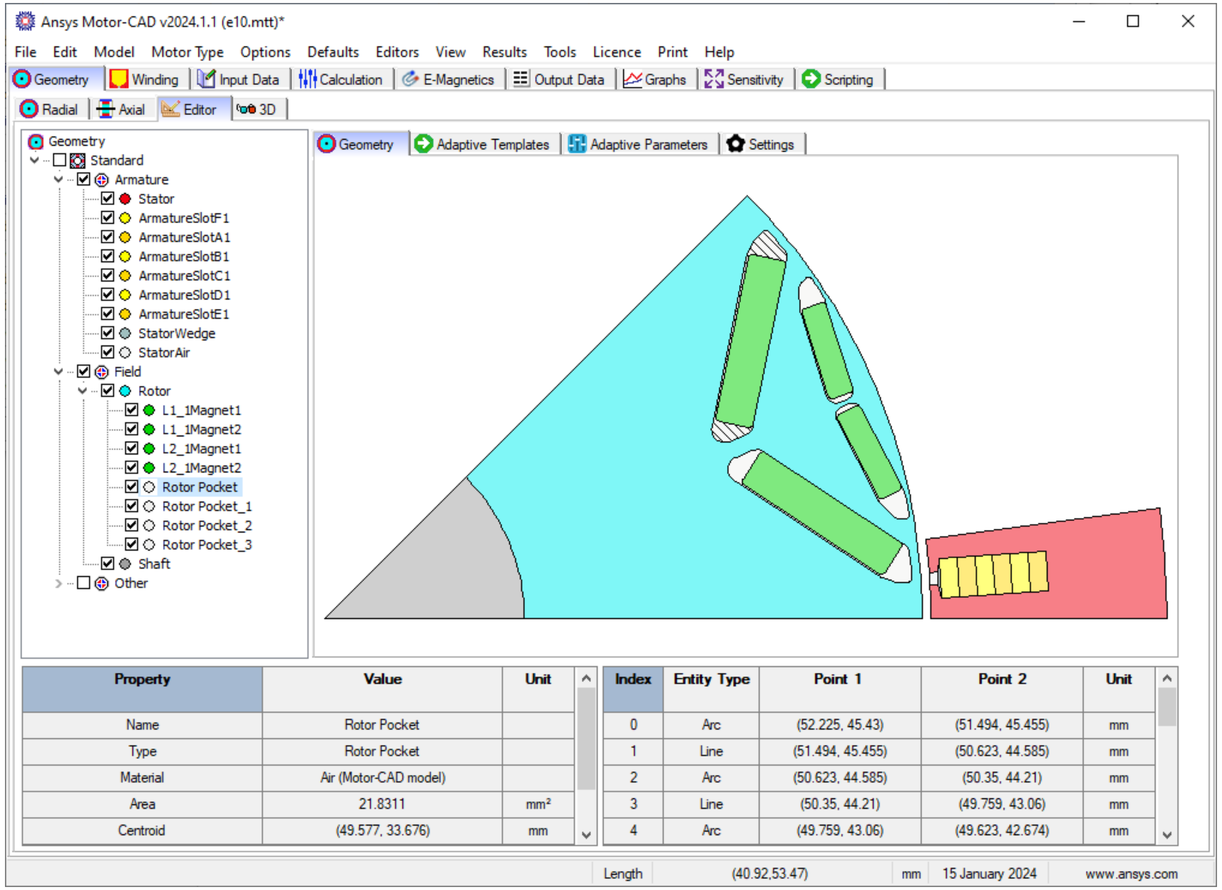
Scripting

©2024 ANSYS, Inc.

Enter: Ansys Motor-CAD 2024 R1 – Adaptive Templates

- The aim:
  - Provide engineers more freedom with geometry creation
  - Provide tools to enable this customization
  - Maintain the same ease-of-use seen with standard templates

- The benefits:
  - Automation and scalability
  - Faster motor design
  - New opportunities for optimization
  - Better Motor performance



©2024 ANSYS, Inc.

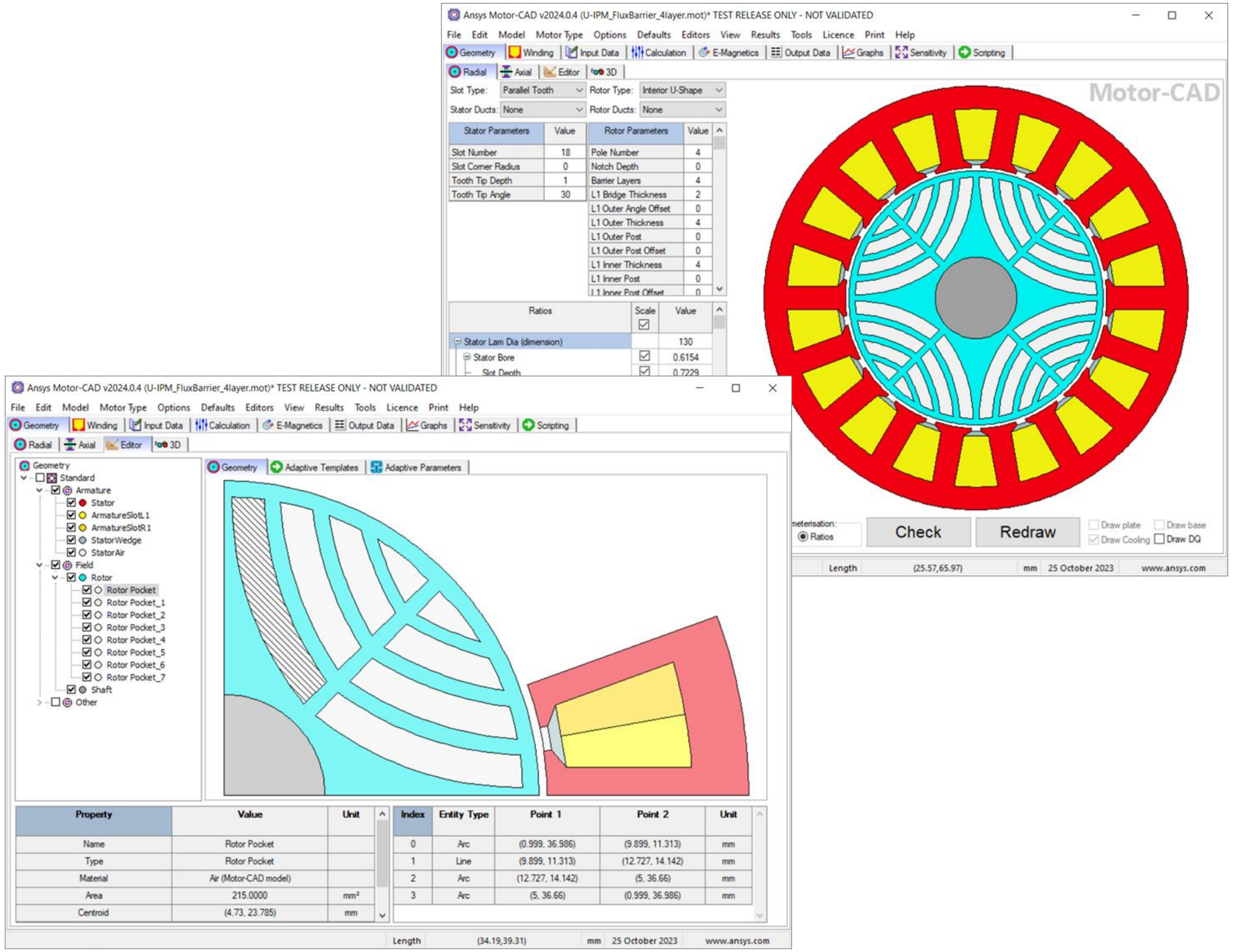
Ansys

Overview of Ansys Motor-CAD 2024 R1 Adaptive Templates

©2024 ANSYS, Inc.

Ansys Motor-CAD 2024 R1 – Adaptive Templates Overview

- Released February 6th, 2024
- Embed Python commands to reparametrize and customize the in-built template geometry
- Add custom geometry parameters
- Flexibility to innovate with the speed and ease of the templated geometry
- Enables IP library to be built up

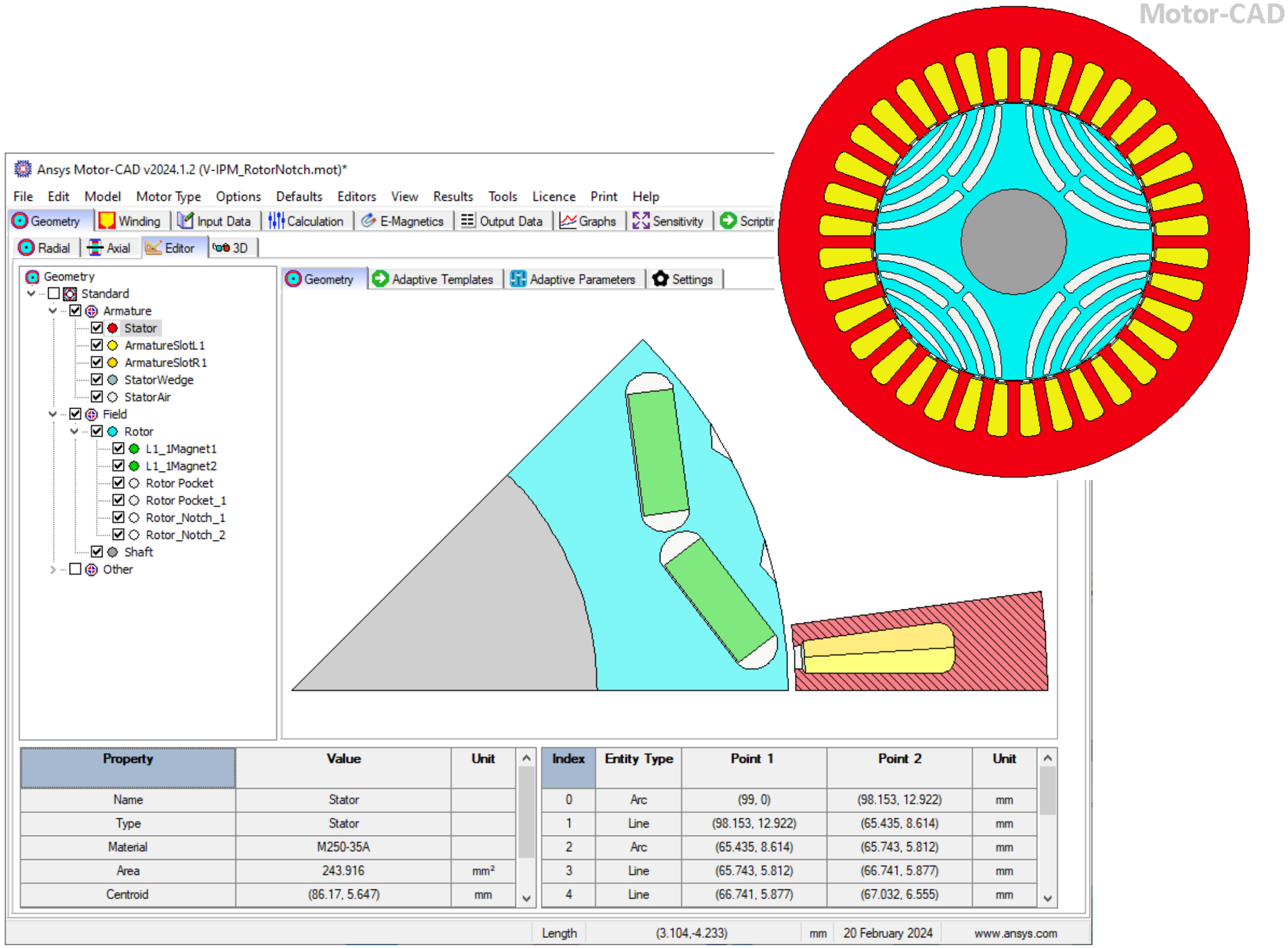


©2024 ANSYS, Inc.

Ansys

Adaptive Templates Demonstration

- **LIVE DEMO**
- New Geometry Editor Window
- Load in a Python script
- Define Adaptive Parameters
- See the geometry update



©2024 ANSYS, Inc.

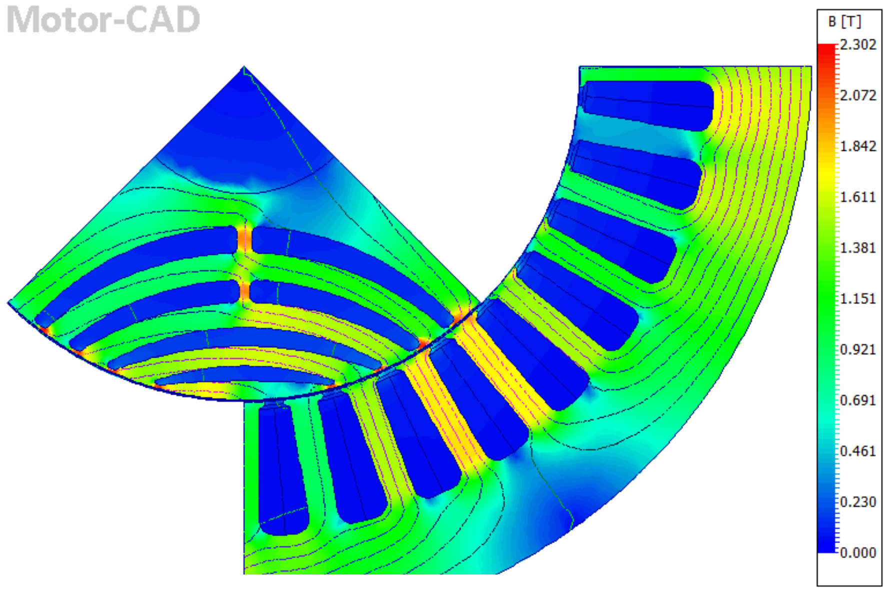
Ansys

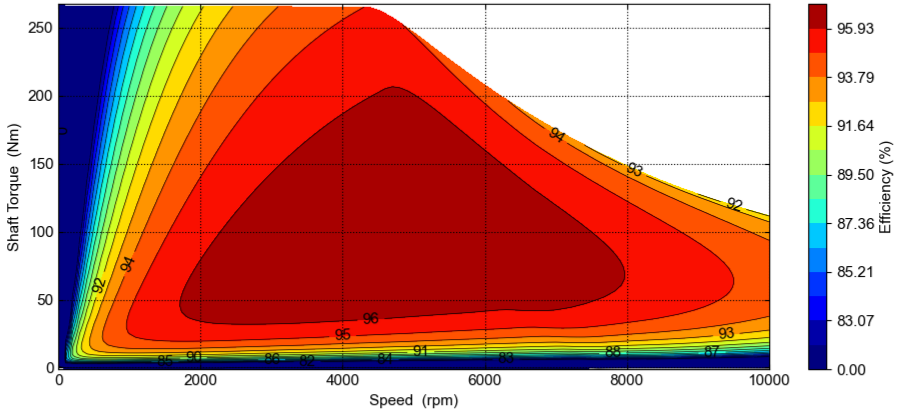
The Adaptive Templates and the Physics Modules of Ansys Motor-CAD

©2024 ANSYS, Inc.

Adaptive Templates & Electromagnetics

- The electromagnetic module is the starting point
- Available to all motor types
- 2D FEA will solve the adapted geometry directly
- Automation of the model is maintained:
  - Meshing, materials, calculation, winding setup, etc.
- Make use of adaptive templates anywhere the FEA is used:
  - Lab module (efficiency mapping and duty cycles)
  - Saturation map export
  - Noise Vibration Harshness (NVH)



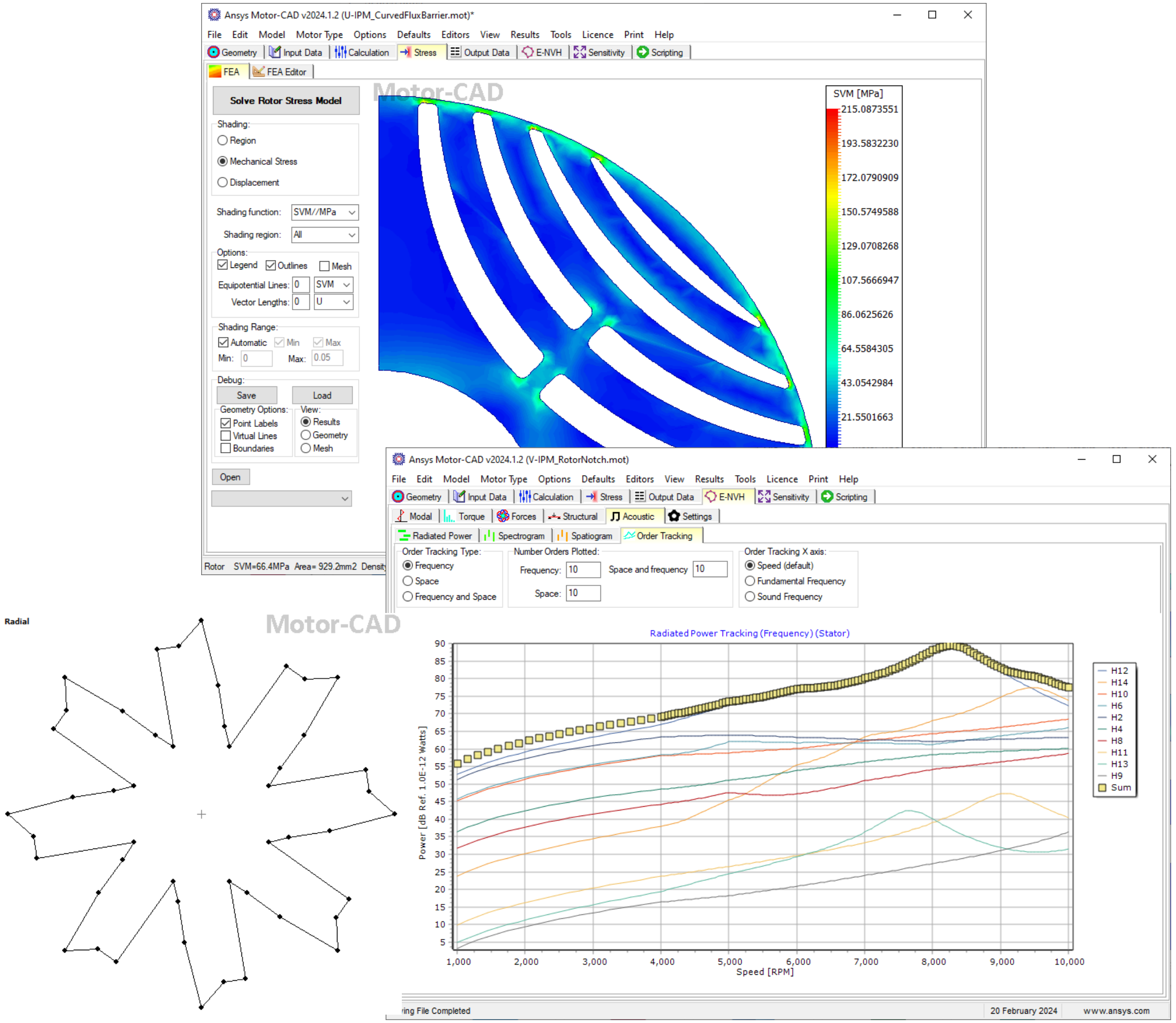


©2024 ANSYS, Inc.

Ansys

Adaptive Templates & Mechanics

- Rotor stress analysis:
  - 2D FEA will solve the adapted geometry directly
  - No additional setup necessary
- Noise & Vibration Analysis:
  - Force calculation uses adaptive template (E-mag)
  - Operating point calculation uses adaptive template (Lab)

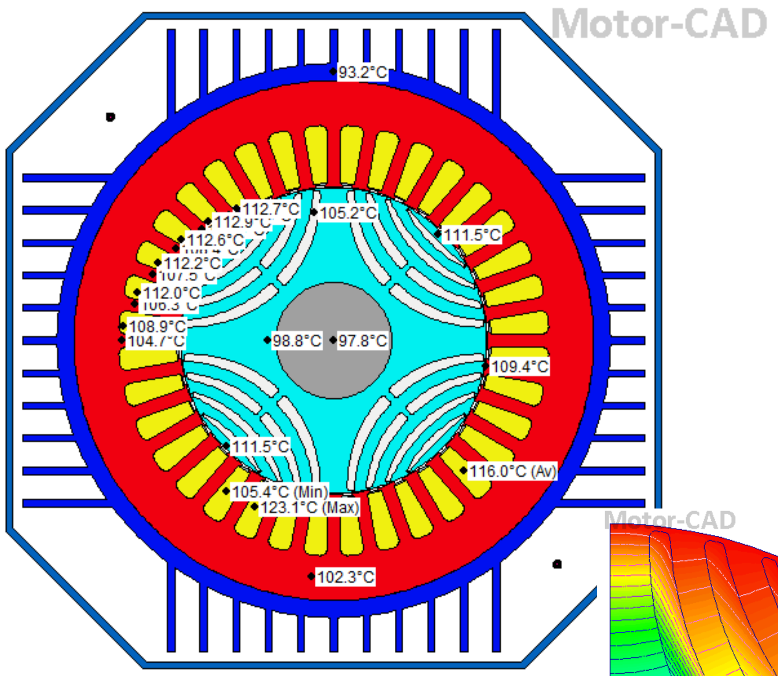


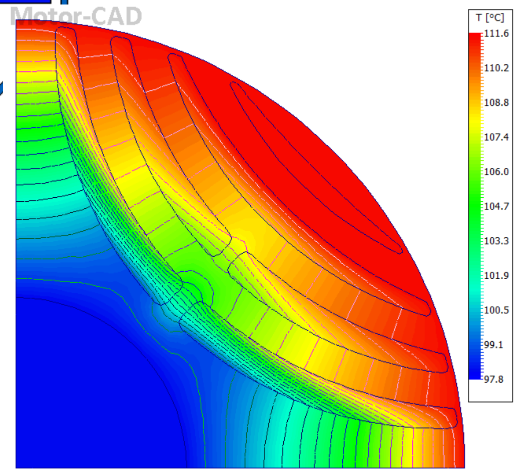
©2024 ANSYS, Inc.

Ansys

Adaptive Templates & Thermals

- Adaptive templates can only be applied to:
  - Active components (stator, rotor, magnets, banding/sleeve and shaft)
  - Radial cross-section
- Thermal model setup primarily follows the standard cooling templates
- Major customizations on the cooling are best served via changes to the thermal circuit
- However, cross-sectional area and the mass of components updates with the adaptive template:
  - Thermal masses & capacitances
  - 2D thermal FEA for calibration





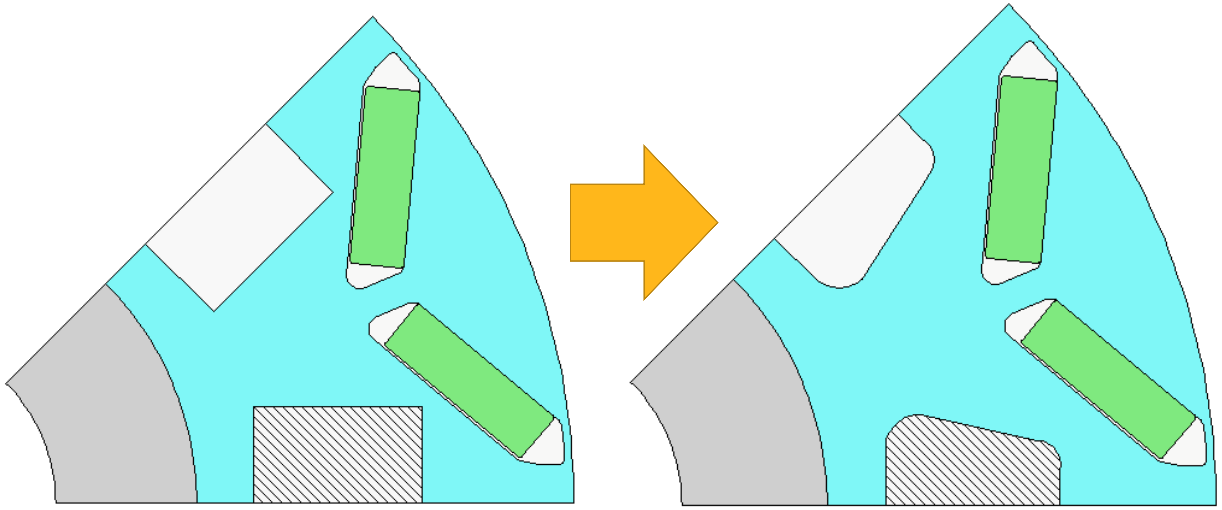
©2024 ANSYS, Inc.

Adaptive Template Creation

©2024 ANSYS, Inc.

Adaptive Template Motor Examples

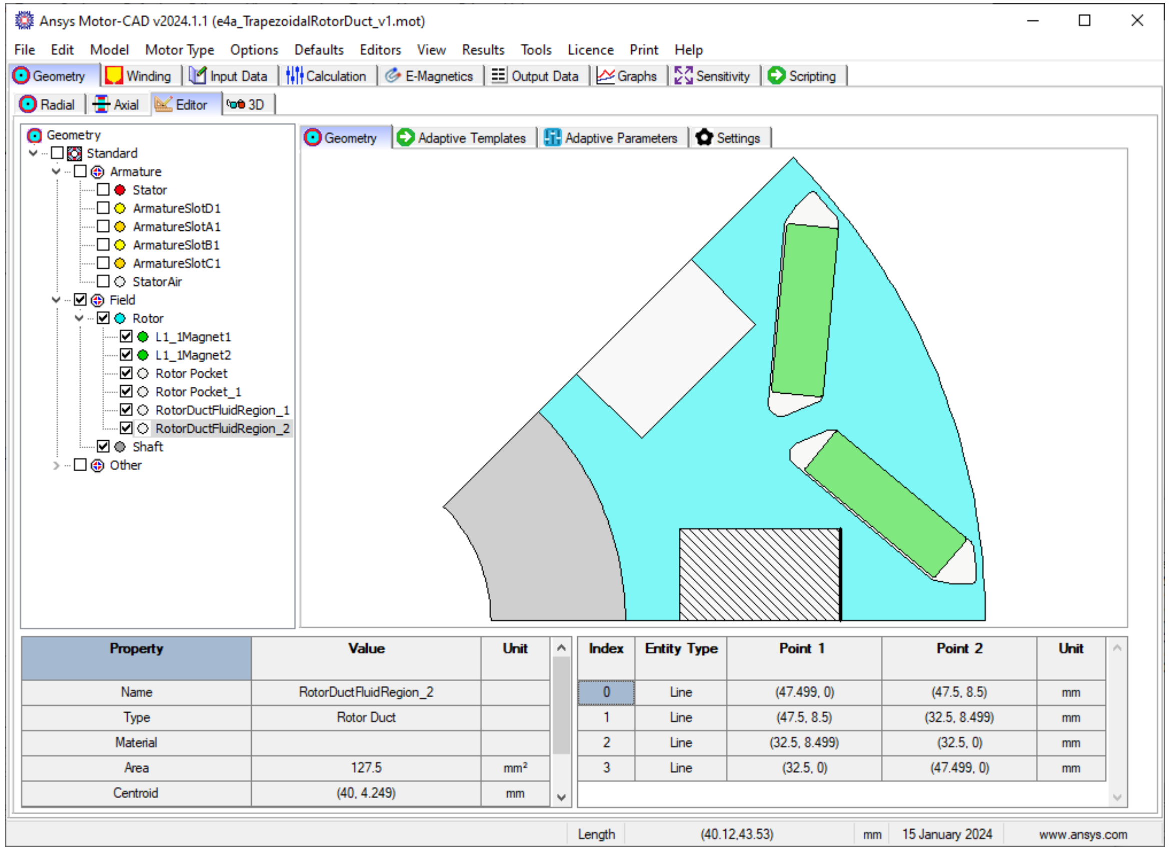
- Example 1: Creating a trapezoid rotor duct
  - Convert a square "standard" rotor duct into a trapezoid
  - We will introduce new parameters to define it
  - We'll show the typical workflow for creating a new adaptive template



©2024 ANSYS, Inc.
Powering Innovation That Drives Human Advancement

Trapezoid Rotor Duct, the Editor Window

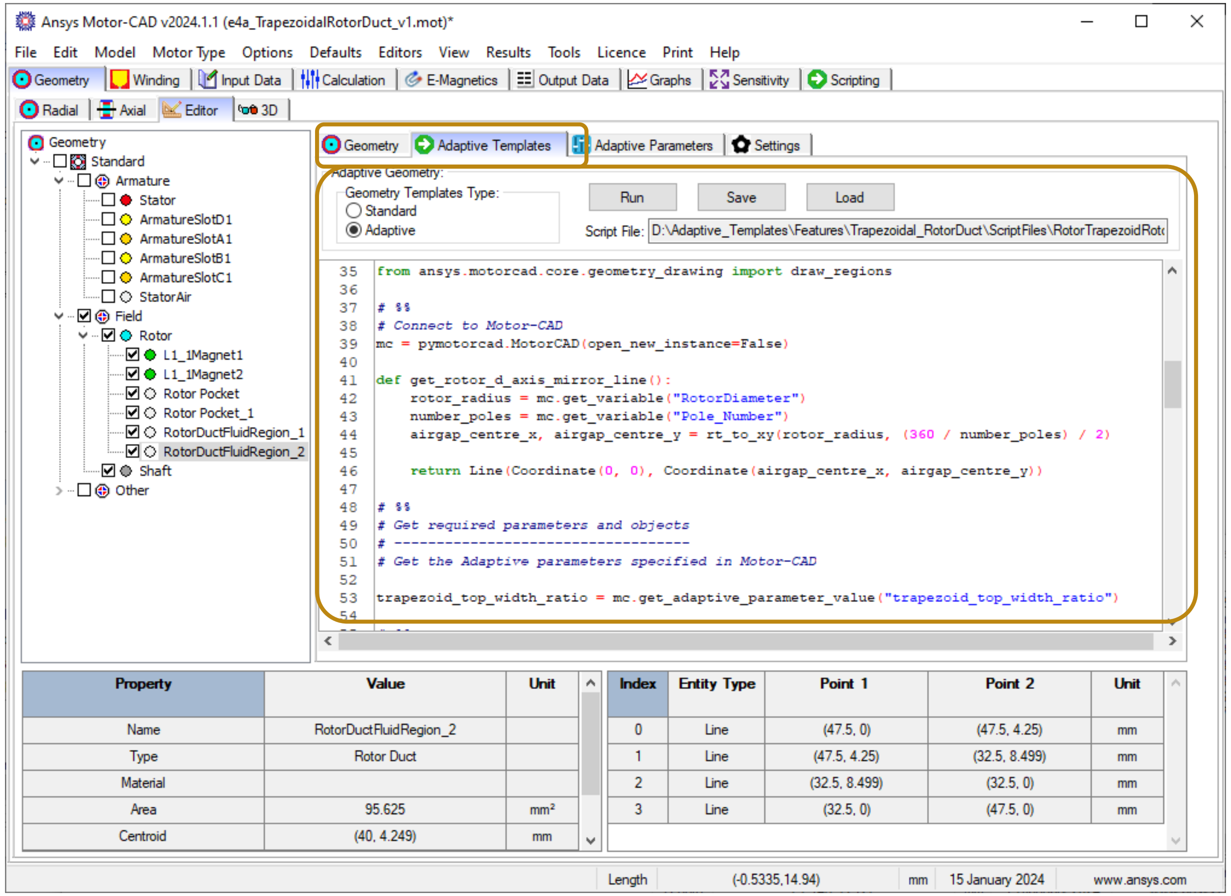
- Dedicated **Geometry** -> **Editor** window:
  - Geometry defined into groups
  - Sub-groups have **regions**
  - **Regions** have **entities** (lines, arcs), **points** (co-ordinates in polar or cartesian) and material definitions
- The Graphical User Interface helps with the customization process



©2024 ANSYS, Inc.

Trapezoid Rotor Duct, Python Scripting

- We can create new regions or make modifications to existing ones.
- However, the main principle of the scripting is to make changes on top of the standard geometry:
  - Significant reduction in the effort required
  - Most customized geometries share many base dimensions with a template: position of magnets, slot, conductors, inner & outer diameter
  - Customizations will adapt to large changes in the motor geometry – perfect for optimizations
  - Automatic setup of the thermal model, conductors, etc.

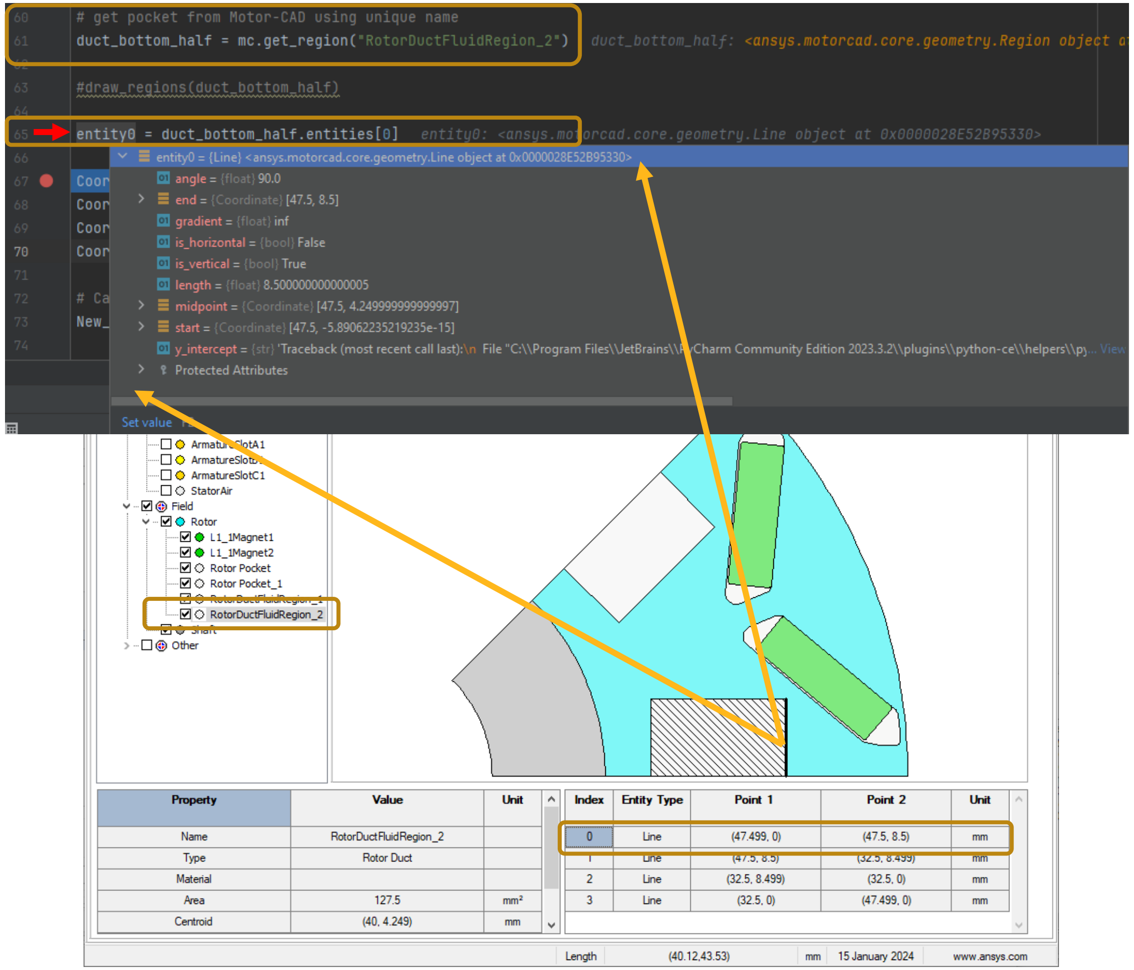


©2024 ANSYS, Inc.

Ansys

Trapezoid Rotor Duct, Example Scripting Workflow – Part 1

- Start with a rectangular rotor duct from the standard geometry templates:
  - Minimize the scripting required
  - Keep the thermal model setup
- From the Motor-CAD editor window find the relevant region name and the Index number.
- Via Python script, grab the entity and a huge amount of information is automatically calculated:
  - Start/end co-ordinates (xy or rt)
  - Length, midpoint, gradient, angle from x/y-axes
  - These can be used for geometry manipulation



©2024 ANSYS, Inc.

Trapezoid Rotor Duct, Example Scripting Workflow – Part 2

- A PyMotorCAD function can plot regions to help with scripting and debugging:

```
draw_regions(duct_bottom_half)
```

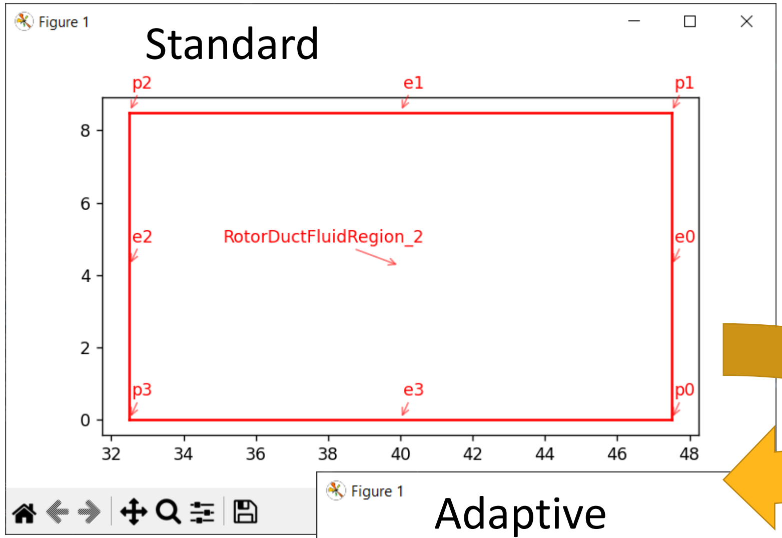
- Define a new adaptive template parameter to create a trapezoid:

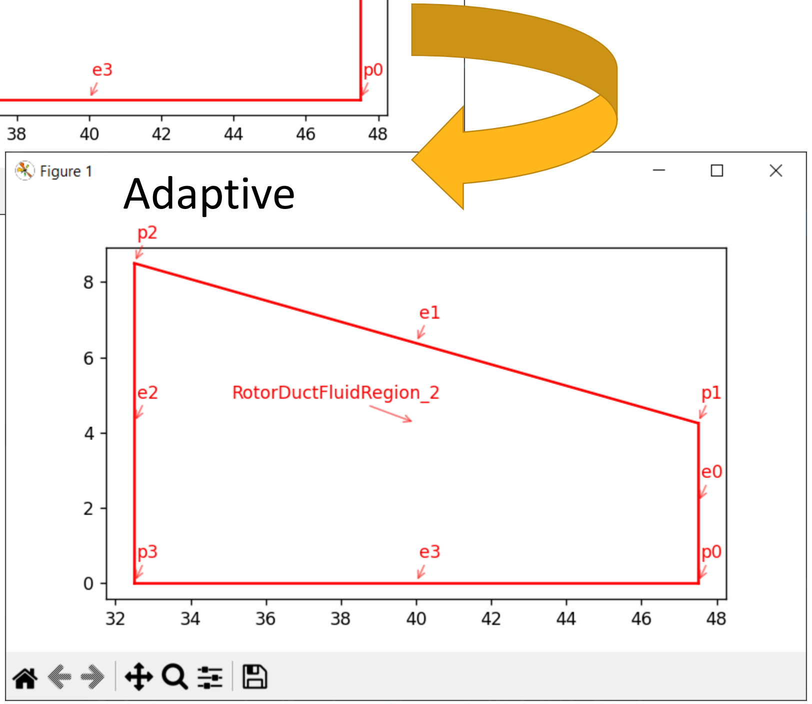
Name	Value	Description
trapezoid_top_width_ratio	0.5	The proportion of the top of the trapezoid compared to the bottom

- Calculate a new Y value and edit p1

```
# Calculate the new y value
New_y = (trapezoid_duct_width / 2.0) * trapezoid_top_width_ratio   New_y: 4.25
Coord1_New = Coordinate(Coord1.x, New_y)   Coord1_New: [47.5, 4.25]
# Method 1: edit the existing point
duct_bottom_half.edit_point(Coord1, Coord1_New)
```

- Update the region and the lines have automatically shifted





©2024 ANSYS, Inc.

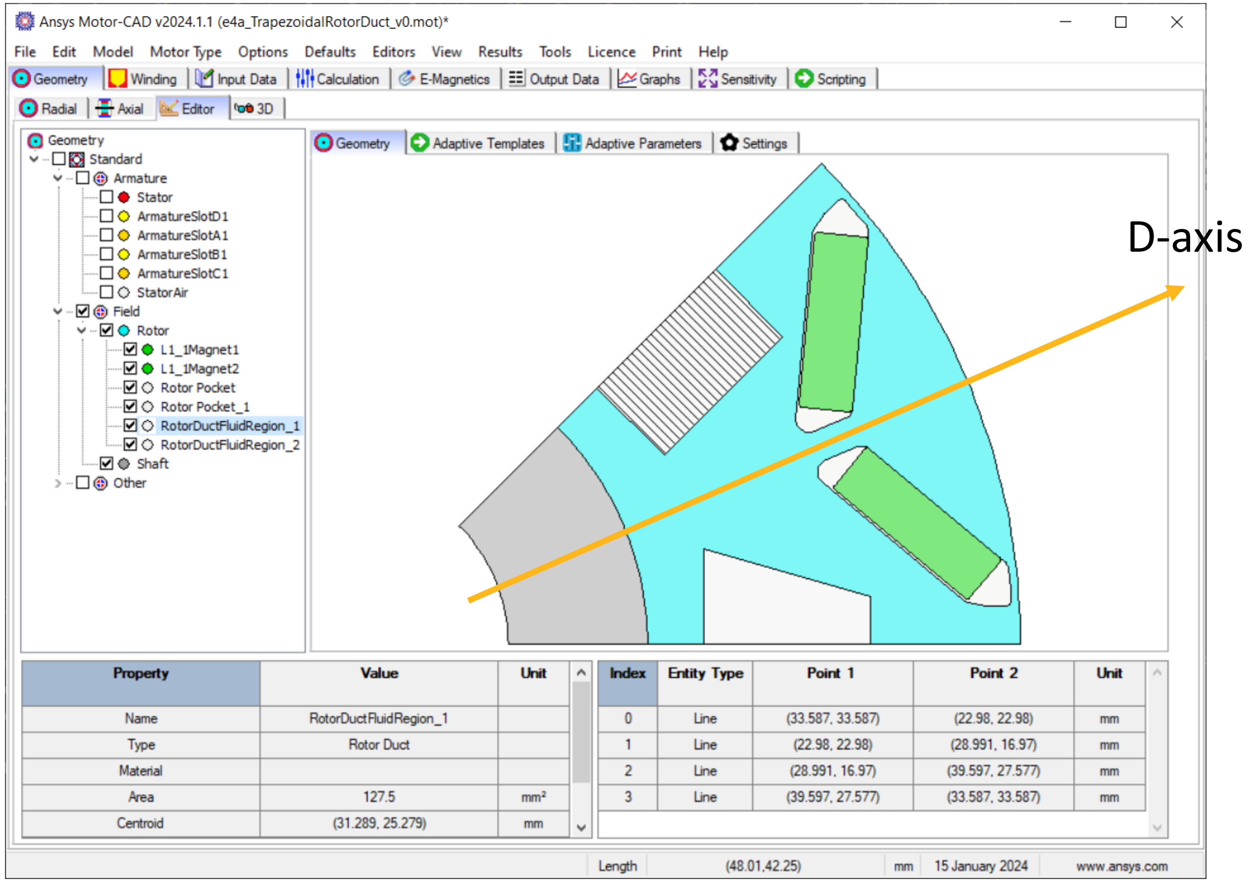
Ansys

Trapezoid Rotor Duct, Example Scripting Workflow – Part 3

- Half of the region has been reshaped from a square to a trapezoid
- Use the PyMotorCAD mirror command to easily generate the co-ordinate on the opposite side of the D-axis:

```
d_axis_mirror_line = get_rotor_d_axis_mirror_line()

MirroredCoord = Coordinate.mirror(Coord1_New, d_axis_mirror_line)
```

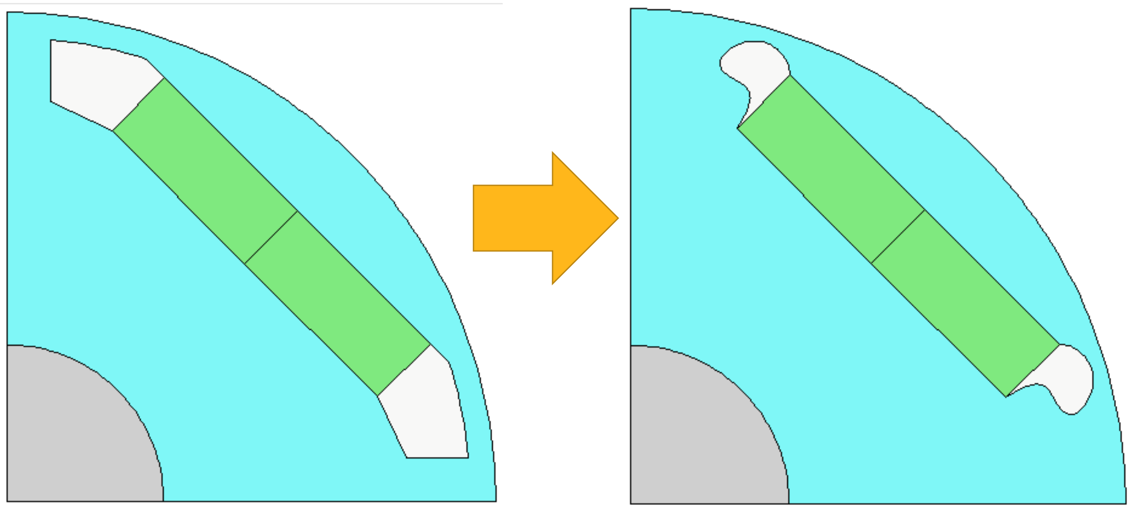
- Update the region and the adaptive template is complete!



©2024 ANSYS, Inc.

Adaptive Template Motor Examples

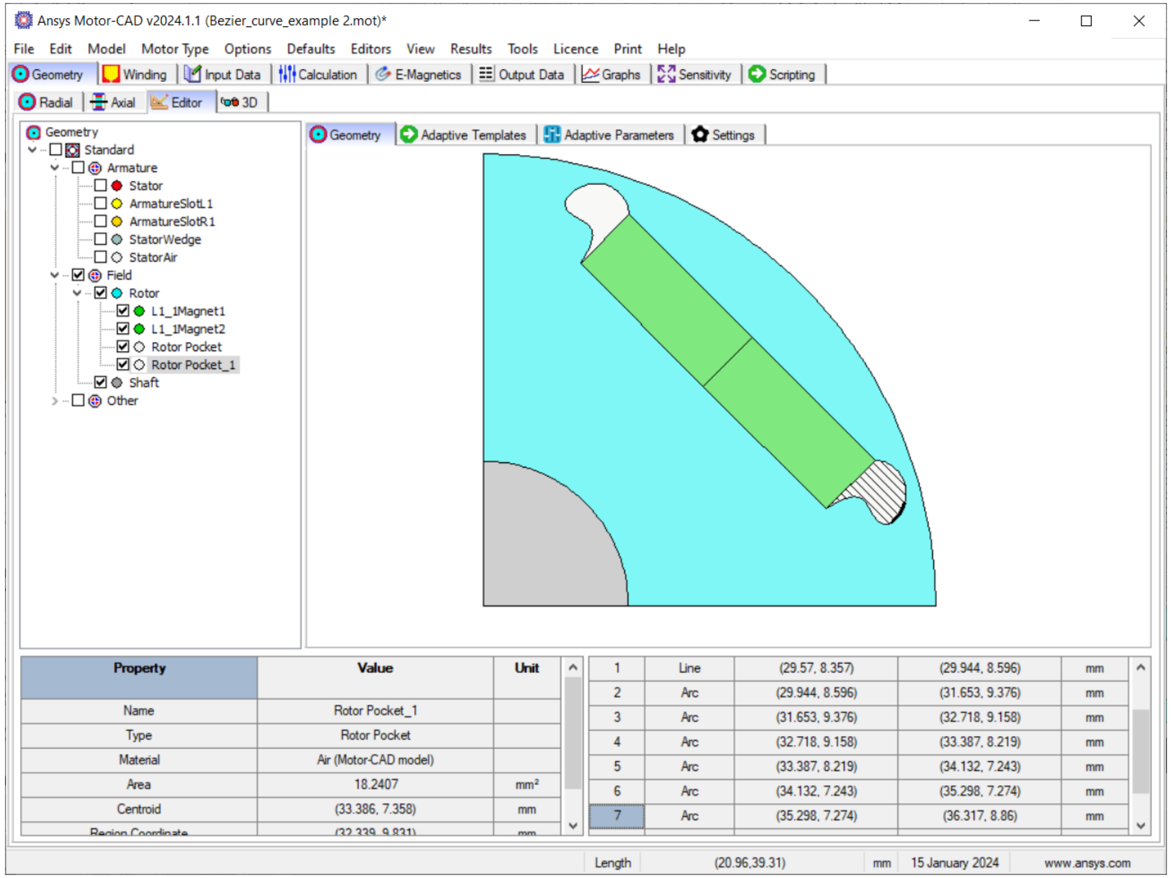
- Example 2: Advanced rotor pocket shaping with Bezier curves
  - More advanced geometry definition to show what is possible
  - Introduction to alternative methods of defining complex shapes



©2024 ANSYS, Inc.

Rotor Pocket Shaping with Bezier Curves – Part 1

- The previous examples were all created using two types of geometry: line and arc
- More freeform shapes may be made up of many arcs
  - 4 parameters needed for every arc
  - Start, end, centre of arc and radius
- Instead, we can create shapes from several co-ordinates + a function for a curve
- Bezier curves are a common mathematical function used for this purpose (polyline is another).

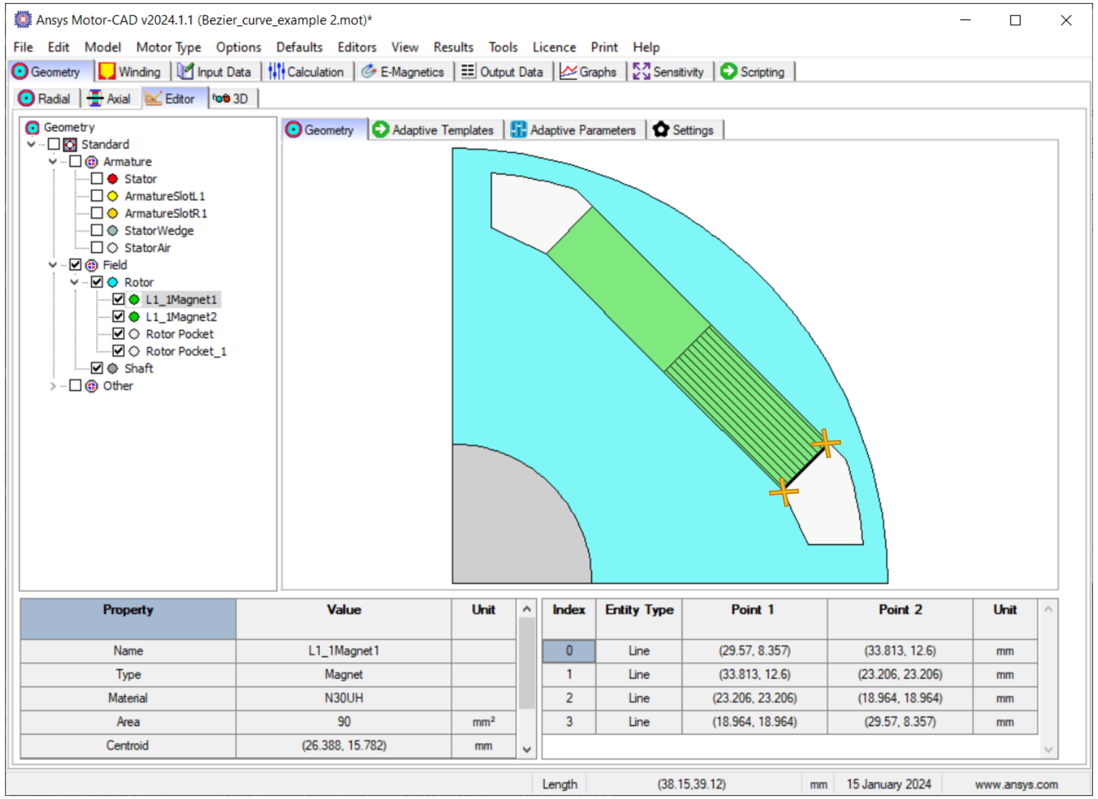


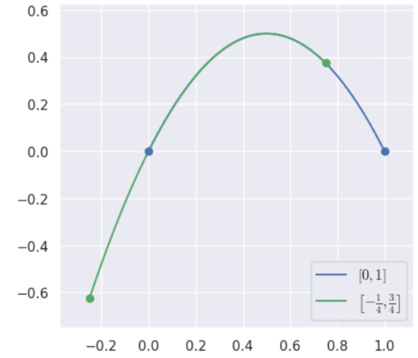
©2024 ANSYS, Inc.

Rotor Pocket Shaping with Bezier Curves – Part 2

- Starting with a standard template, the start and end-coordinate for the rocket pocket is found.
- Several adaptive parameters are created and combined with a python package for Bezier curves.
- Aim: gather a set of co-ordinates along the geometry

Name	Value	Description
bez_curve_projection	13	How far from the magnet the rotor pocket extends.
upperconvex	0.2	Magnitude and direction of the curvature on the top arc, closest to airgap
lowerconcave	-0.2	Magnitude and direction of the curvature on the bottom arc, closer to the shaft





©2024 ANSYS, Inc.

Ansys

Rotor Pocket Shaping with Bezier Curves – Part 3

- A PyMotorCAD command can convert a set of co-ordinates automatically into a list of lines and arcs.
- The functions fits lines and arcs to the provided co-ordinates, with a given tolerance.
- The output is a minimized set of arcs and lines, which can be used to create a new adaptive template region.
- **LIVE DEMO**

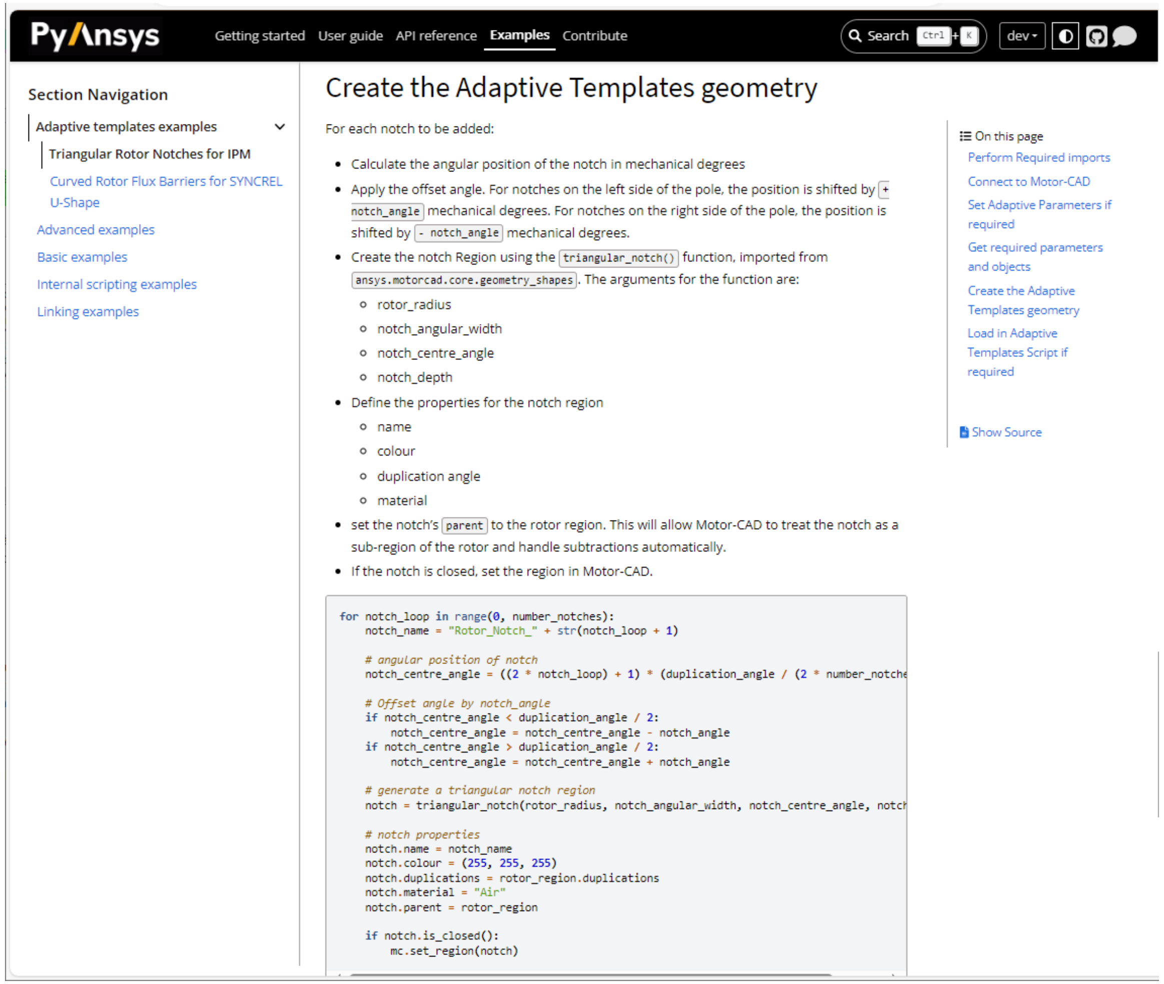
```
def return_entity_list(coordinates, line_tolerance, arc_tolerance):
    """Get list of entities from a list of coordinates.

    Parameters
    ----------
    coordinates : List of ansys.motorcad.core.geometry.Coordinate
        coordinates from which to generate the geometry
    line_tolerance : float
        maximum allowed distance of point away from generated line
    arc_tolerance : float
        maximum allowed distance of point away from generated arc

    Returns
    -------
    ansys.motorcad.core.geometry.EntityList

    """
    p = _PointFitting()
    return p.return_entity_list(coordinates, line_tolerance, arc_tolerance)
```

```
bez_curve_entities = return_entity_list(xylist, linetolerance, arctolerance)
```


©2024 ANSYS, Inc.

Python Scripting, PyMotorCAD and GitHub

- PyMotorCAD in 2024 R1 now contains:
  - Many new and powerful functions to aid custom geometry creation
  - Expanded guides to help users get started
  - Regularly updated throughout the year!
- GitHub is an open-source community
  - Download, use and alter existing examples
  - Ansys created as well as user submitted
- Engineers can build up their own internal libraries of motor geometries, easily scalable to any Motor design



©2024 ANSYS, Inc.

Workflows in Other Ansys Products

©2024 ANSYS, Inc.

Ansys Tools and the Adaptive Templates of Ansys Motor-CAD

Ansys MAXWELL

Ansys OPTISLANG

Ansys DISCOVERY

©2024 ANSYS, Inc.

Ansys Motor-CAD & Ansys optiSLang – Motor Optimization

- Any adaptive template parameters created by a user, can be found within the Motor-CAD to optiSLang export window.
- The adaptive template script will move automatically with the export into optiSLang.
- A PyMotorCAD function can be used to check if geometry is valid (overlapping)

collides

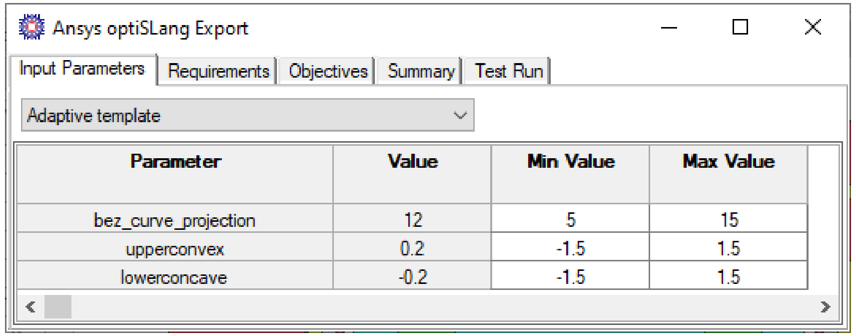
`Region.collides(regions)`

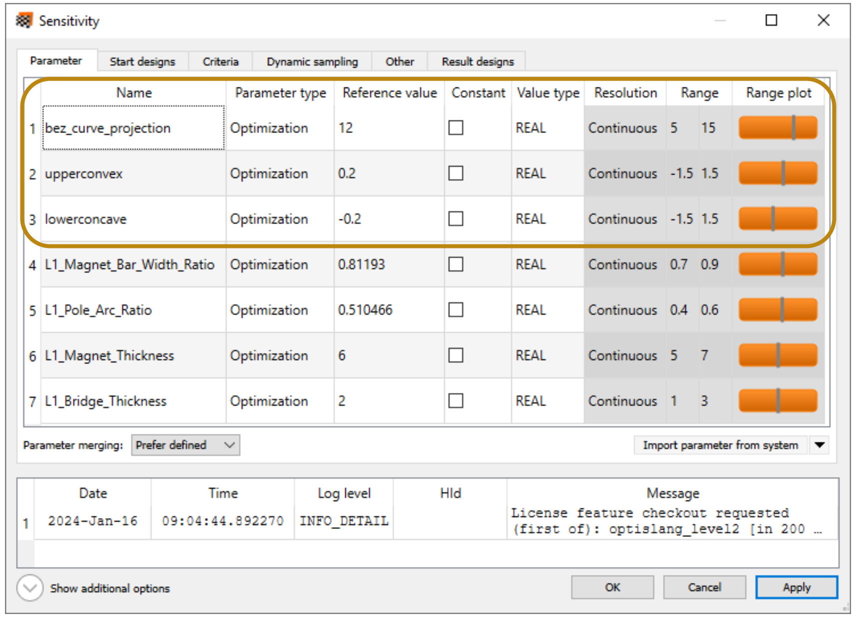
Check whether any of the specified regions collide with self.

Parameters:

regions : `ansys.motorcad.core.geometry.Region` or `list` of `ansys.motorcad.core.geometry.Region`

Motor-CAD region object/list of objects

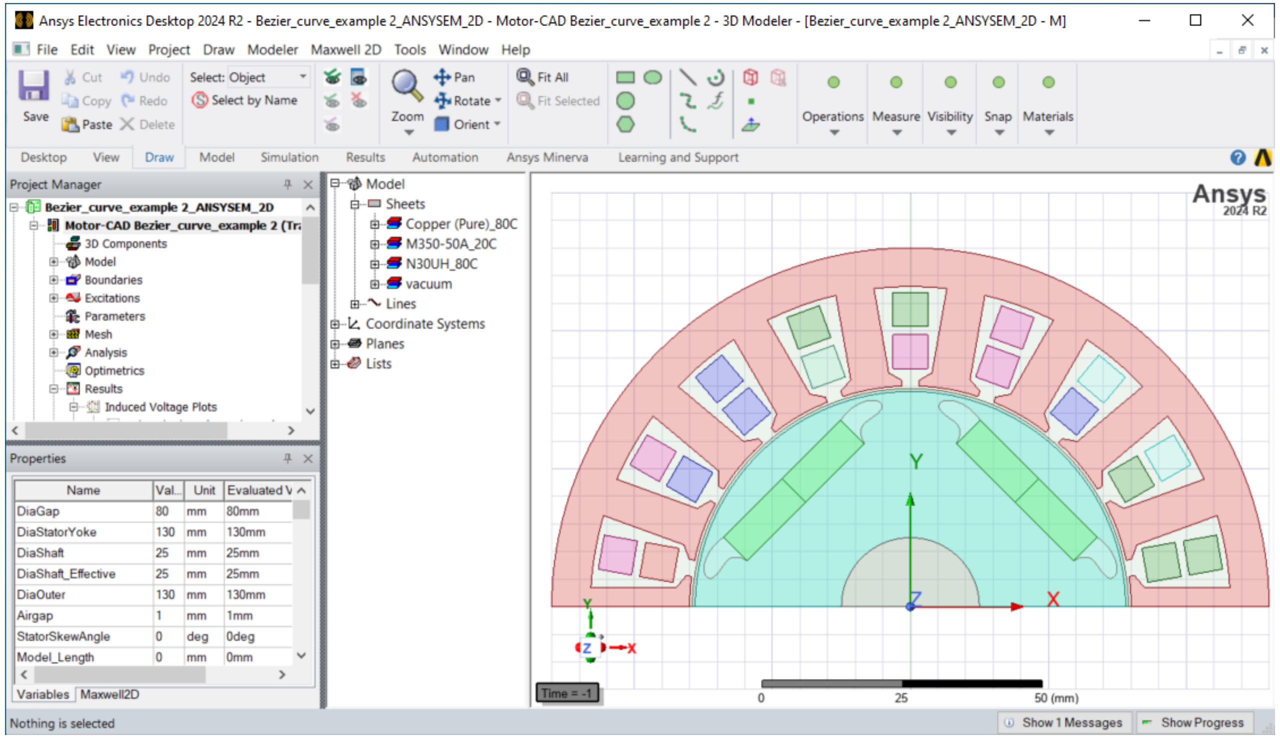




©2024 ANSYS, Inc.

Ansys Motor-CAD & Ansys Maxwell – Advanced Electromagnetics

- Exporting into Maxwell will carry across the adaptive template geometry!
- However, currently the geometry is fixed and will not carry across the parameterization.
- Bringing the parameterization across is on the roadmap.
- The Motor-CAD to Discovery export follows a similar process.



©2024 ANSYS, Inc.

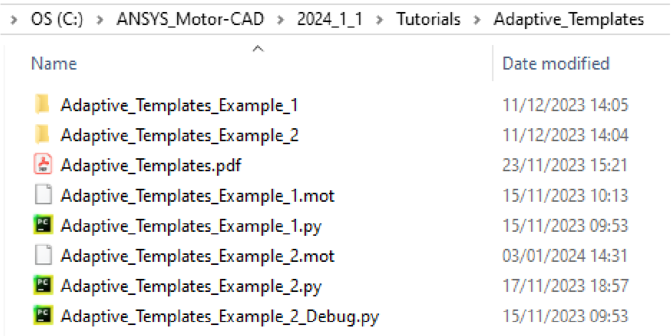
Ansys

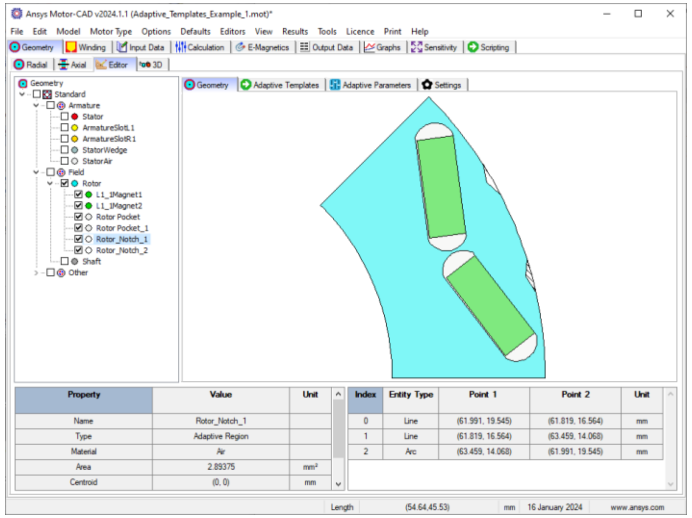
Getting Started

©2024 ANSYS, Inc.

Ansys Motor-CAD 2024 R1 – Adaptive Templates Tutorial

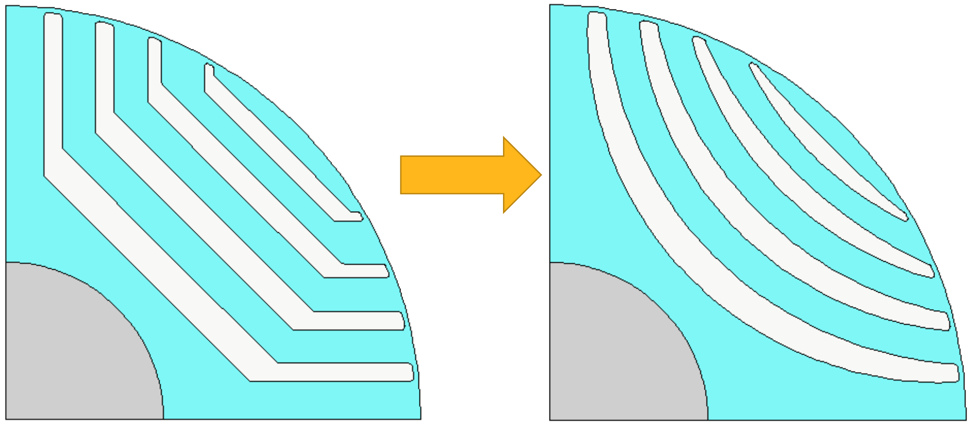
- A tutorial is installed locally with 2024 R1:





- The tutorial goes through the fundamentals of the feature: the Motor-CAD GUI, scripting etc.

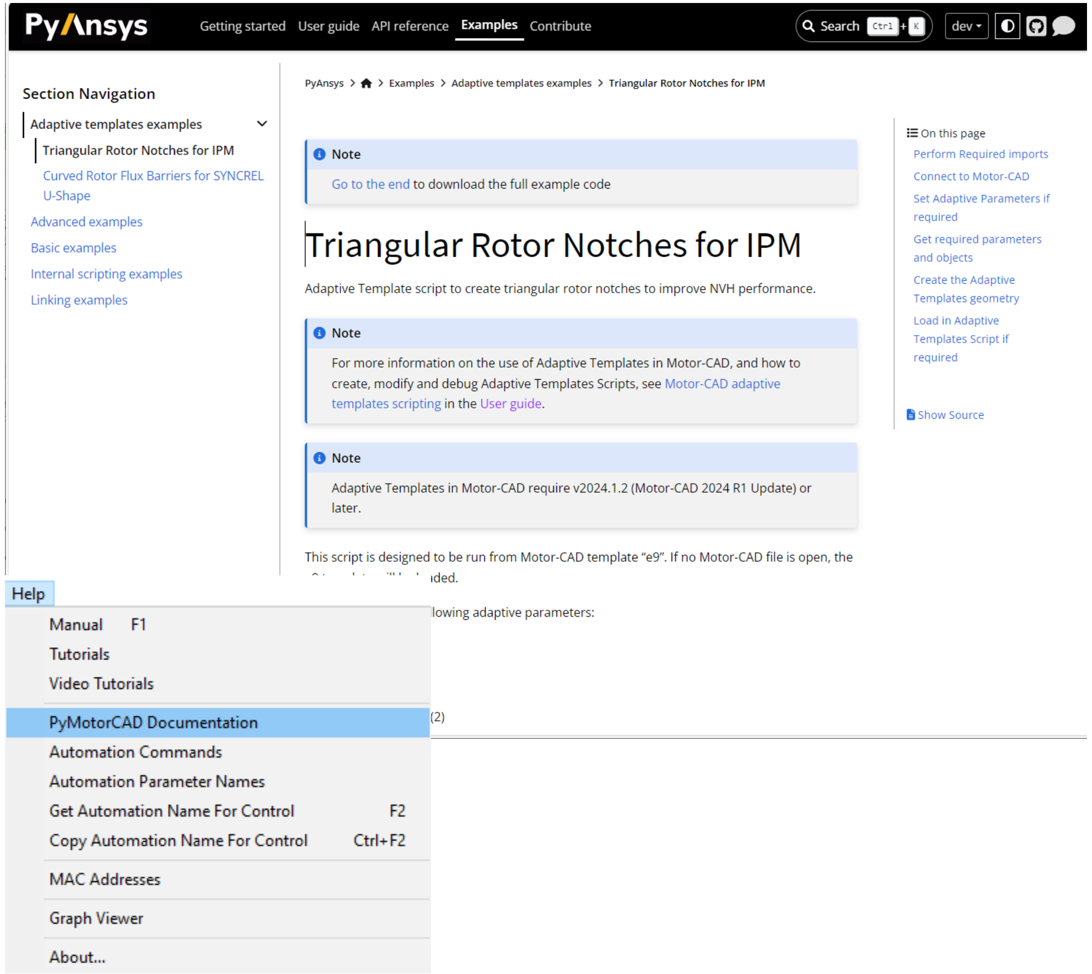
- The tutorial comes with two examples:
  - Adding a V-shape rotor notch
  - Converting the Synchronous reluctance U-IPM template into a curved barrier



©2024 ANSYS, Inc.

Ansys Motor-CAD 2024 R1 – PyMotorCAD & GitHub

- Check in on PyMotorCAD for regular updates
  - More detailed guides to help with python scripting
  - New adaptive template examples
- Find the link from within Ansys Motor-CAD, Help -> PyMotorCAD documentation



©2024 ANSYS, Inc.

Summary

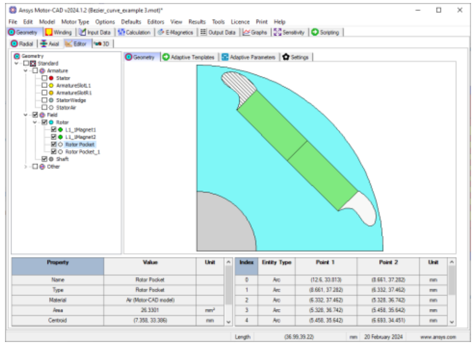
©2024 ANSYS, Inc.

Ansys Motor-CAD 2024 R1 – Adaptive Templates

1

Enable Engineers to:

- Easily create customized shapes
- Maximize multiphysics motor performance
- Build up IP library



2

Major Benefits:

- Flexibility to innovate with ease
- Faster motor design
- Increased automation & scalability
- New opportunities for optimization
- Better motor performance

©2024 ANSYS, Inc.

Ansys