In [1]:
```
%matplotlib inline
```

Structural Analysis of a Lathe Cutter {#ref_lathe_cutter_example}

Summary: Basic walk through PyMAPDL capabilities.

Objective

The objective of this example is to highlight some regularly used PyMAPDL features via a lathe cutter finite element model. Lathe cutters have multiple avenues of wear and failure, and the analyses supporting their design would most often be transient thermal-structural. However, for simplicity, this simulation example uses a non-uniform load.

 **Figure 1: Lathe cutter geometry and load description.**{.align-centeralign-center width="600px"}

Contents

1. **Variables and launch** Define necessary variables and launch MAPDL.
2. **Geometry, mesh, and MAPDL parameters** Import geometry and inspect for MAPDL parameters. Define linear elastic material model with Python variables. Mesh and apply symmetry boundary conditions.
3. **Coordinate system and load** Create a local coordinate system for the applied load and verify with a plot.
4. **Pressure load** Define the pressure load as a sine function of the length of the application area using numpy arrays. Import the pressure array into MAPDL as a table array. Verify the applied load and solve.
5. **Plotting** Show result plotting, plotting with selection, and working with the plot legend.
6. **Postprocessing:** List a result two ways: use PyMAPDL and the Pythonic version of APDL. Demonstrate extended methods and writing a list to a file.
7. **Advanced plotting** Use of mesh.grid for additional postprocessing.

Step 1: Variables and launch

Define variables and launch MAPDL.

In [2]:
```
import os

import numpy as np

from ansys.mapdl.core import launch_mapdl
from ansys.mapdl.core.examples.downloads import download_example_data

# cwd = current working directory
path = os.getcwd()
```

```
PI = np.pi
EXX = 1.0e7
NU = 0.27
```

Often used MAPDL command line options are exposed as Pythonic parameter names in `ansys.mapdl.core.launch_mapdl` {.interpreted-text role="func"}. For example, `-dir` has become `run_location`. You could use `run_location` to specify the MAPDL run location. For example:

..code:: python3

> mapdl = launch_mapdl(run_location=path)

Otherwise, the MAPDL working directory is stored in `mapdl.directory`. In this directory, MAPDL will create some of the images we will show later.

Options without a Pythonic version can be accessed by the `additional_switches` parameter. Here `-smp` is used only to keep the number of solver files to a minimum.

In [3]:
```
license = "mech_2" # Ansys Mechanical Premium
execfile = "C:\\Program Files\\ANSYS Inc\\v212\\ansys\\bin\\winx64\\ANSYS212.exe"
mapdl = launch_mapdl(exec_file=execfile, license_type=license,additional_switches='
```

Step 2: Geometry, mesh, and MAPDL parameters

- Import geometry and inspect for MAPDL parameters.
- Define material and mesh, and then create boundary conditions.

In [4]:
```
# First, reset the MAPDL database.
mapdl.clear()
```

Import the geometry file and list any MAPDL parameters.

In [5]:
```
lathe_cutter_geo = download_example_data("LatheCutter.anf", "geometry")
mapdl.input(lathe_cutter_geo)
mapdl.finish()
print(mapdl.parameters)
```

```
MAPDL Parameters
----------------
PRESS_LENGTH                    : 0.055
UNIT_SYSTEM                     : "bin"
```

Use pressure area per length in the load definition.

In [6]:
```
pressure_length = mapdl.parameters["PRESS_LENGTH"]

print(mapdl.parameters)
```

```
MAPDL Parameters
----------------
PRESS_LENGTH                    : 0.055
UNIT_SYSTEM                     : "bin"
```

Change the units and title.

In [7]:
```
mapdl.units("Bin")
mapdl.title("Lathe Cutter")
```

Out[7]:
```
TITLE=
 Lathe Cutter
```

Set material properties.

In [8]:
```
mapdl.prep7()
mapdl.mp("EX", 1, EXX)
mapdl.mp("NUXY", 1, NU)
```

Out[8]:
```
MATERIAL          1     NUXY =  0.2700000
```

The MAPDL element type `SOLID285` is used for demonstration purposes. Consider using an appropriate element type or mesh density for your actual application.

In [9]:
```
mapdl.et(1, 285)
mapdl.smrtsize(4)
mapdl.aesize(14, 0.0025)
mapdl.vmesh(1)

mapdl.da(11, "symm")
mapdl.da(16, "symm")
mapdl.da(9, "symm")
mapdl.da(10, "symm")
```

Out[9]:
```
CONSTRAINT AT AREA    10
      LOAD LABEL = SYMM
```

Step 3: Coordinate system and load

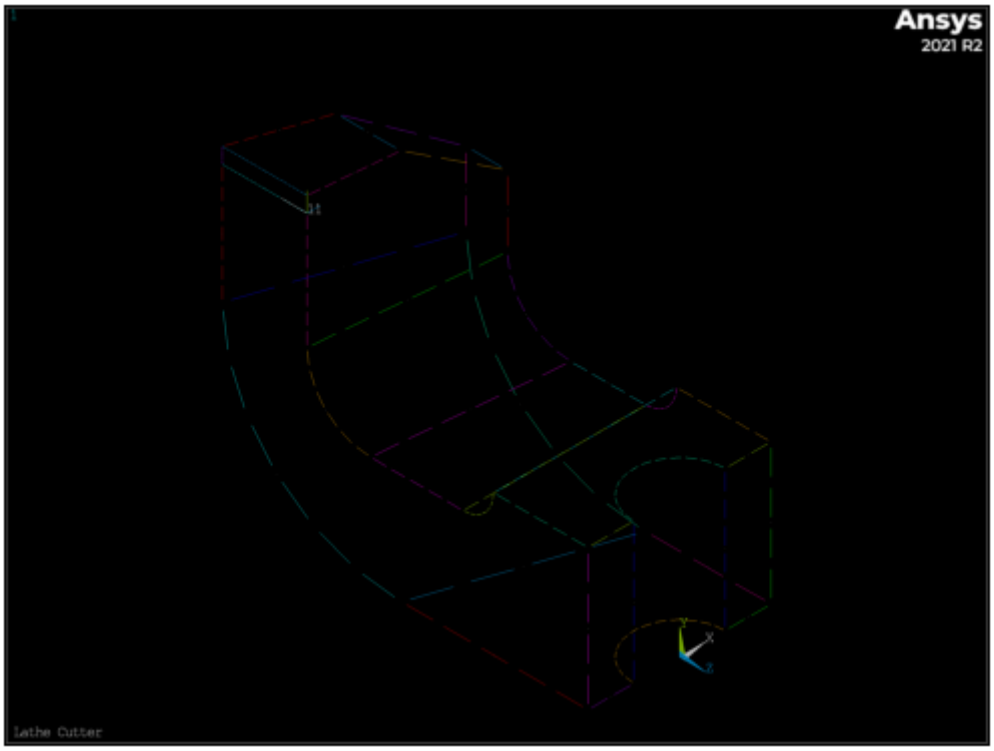
Create a local Coordinate System (CS) for the applied pressure as a function of local X.

Local CS ID is 11

In [10]:
```
mapdl.cskp(11, 0, 2, 1, 13)
mapdl.csys(1)
mapdl.view(1, -1, 1, 1)
mapdl.psymb("CS", 1)
mapdl.vplot(
    color_areas=True,
    show_lines=True,
    cpos=[-1, 1, 1],
    smooth_shading=True,
)
```

VTK plots do not show MAPDL plot symbols. However, to use MAPDL plotting capabilities, you can set the keyword option `vtk` to `False`.

In [11]:
```
mapdl.lplot(vtk=False)
```




```
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
```

Step 4: Pressure load