PyAnsys Geometry cheat sheet

Version: 0.15.dev0

Verify your installation

Check your PyAnsys Geometry version

```
from ansys.geometry.core import __version__
print(f"PyAnsys Geometry version: {__version__}")
```

```
PyAnsys Geometry version: 0.15.dev0
```

Sketching

There are two ways of creating 2D sketches in PyAnsys Geometry.

```
from ansys.geometry.core.sketch import Sketch
from ansys.geometry.core.math import Point2D
```

Functional-style sketching

```
sketch = Sketch()

(
  sketch
    .segment_to_point(Point2D([3, 3]), "Segment2")
    .segment_to_point(Point2D([3, 2]), "Segment3")
    .segment_to_point(Point2D([0, 0]), "Segment4")
)
```

You can visualize the sketch by calling the `plot` method.

```
sketch.plot()
```

Object-oriented sketching

```
sketch = Sketch()

sketch.triangle(
    Point2D([-10, 10]),
    Point2D([5, 6]),
    Point2D([-10, -10]),
)
```

Modeling

Launch a modeling session

```
from ansys.geometry.core import launch_modeler
modeler = launch_modeler()
print(modeler)
```

```
WARNING -  -  docker_instance -
    _check_port_availability - Service is already
    running at port 700...
/home/runner/work/pyansys-geometry/pyansys-geometry/.
    UserWarning: Starting gRPC client without TLS
    on localhost:700. This is INSECURE. Consider
    using a secure connection.
  warn(f"Starting gRPC client without TLS on
      {target}. This is INSECURE. Consider using
      a secure connection.")
```

```
Ansys Geometry Modeler (0x7fc38f53b380)

Ansys Geometry Modeler Client (0x7fc38f53bcb0)
  Target:     localhost:700
  Connection: Healthy
  Backend info:
    Version:            27.1.0
    Backend type:       CORE_LINUX
    Backend number:     20260130.2
    API server number:  1984
    CADIntegration:     27.1.0.7
```

By default, it will detect which modeling service is available on your system and launch it. If you have multiple modeling services installed, you can specify which one to use by passing the `mode` argument.

```
modeler = launch_modeler(mode='spaceclaim')
modeler = launch_modeler(mode='discovery')
modeler = launch_modeler(mode='geometry_service')
```

Connect to an existing modeler

```
from ansys.geometry.core import Modeler
modeler = Modeler()
print(modeler)
```

Create a design

```
design = modeler.create_design("MyDesign")
print(design)
```

```
ansys.geometry.core.designer.Design 0x7fc38dc3e510
  Name                : MyDesign
  Is active?          : True
  N Bodies            : 0
  N Components        : 0
  N Coordinate Systems : 0
  N Named Selections  : 0
  N Materials         : 0
  N Beam Profiles     : 0
  N Design Points     : 0
  N Datum Planes      : 0
```

Create a body by extruding a sketch

```
body = design.extrude_sketch("MyBody", sketch, 2)
print(body)
```

```
ansys.geometry.core.designer.Body 0x7fc38c1b8590
  Name                : MyBody
  Exists              : True
  Parent component    : MyDesign
  MasterBody          : 0:22
  Surface body        : False
  Color               : #D6F7D1
```

Plot the design

```
design.plot()
```

Export the design to a file

```
scdocx_path = design.export_to_scdocx()
pmdb_path = design.export_to_pmdb()
para_txt_path = design.export_to_parasolid_text()
para_bin_path = design.export_to_parasolid_bin()
fmd_path = design.export_to_fmd()
disco_path = design.export_to_disco()
```

Extra: Product scripting

Ansys SpaceClaim and Ansys Discovery support product scripting, and so does the Ansys Geometry service. If you have a product script you want to run, you can use the `run_discovery_script_file` method available on the `Modeler` object. The `script_args` parameter is optional and they will be made available to the script inside a dictionary called `argsDict`.

```
result = modeler.run_discovery_script_file(
  file_path="path/to/script.py",
  script_args={"arg1": "value1", "arg2": "value2"},
)
```