# Standard Operating Procedure (SOP) for True\_Curve.py

## Purpose

The True\_Curve.py script and its derivatives (QGIS script and ArcGIS toolbox) are used to convert complex curved geometries (e.g., CircularString, CurvePolygon, MultiCurve, MultiSurface) into linear representations (e.g., LineString, Polygon). This standardization ensures compatibility with a wider range of GIS software and workflows that may not support non-linear geometry types.

## Supported Platforms

This toolset includes three implementations:

* **Standalone Python Script (True\_Curve.py)**
* **ArcGIS Python Toolbox (ArcGIS\_curves.pyt)**
* **QGIS Processing Script (QGIS\_Remove\_curved\_geometry.py)**

Each version performs the same core functionality: detecting, segmentizing & linearizing curve-based geometries.

## Procedure

Before running the tool, ensure your geospatial dataset meets the following conditions:

**Supported Formats**

* File Geodatabase (.gdb)
* GeoPackage (.gpkg)
* Shapefile (.shp)
* ESRI JSON (.json)
* GeoJSON (.geojson)
* DXF (.dxf)
* DWG (.dwg) *(limited support based on dependencies)*

**Best Practices**

* Place input files in a readable directory (avoid special characters or spaces in path names for CLI use).
* Ensure that the file is not open in another application.

# Usage

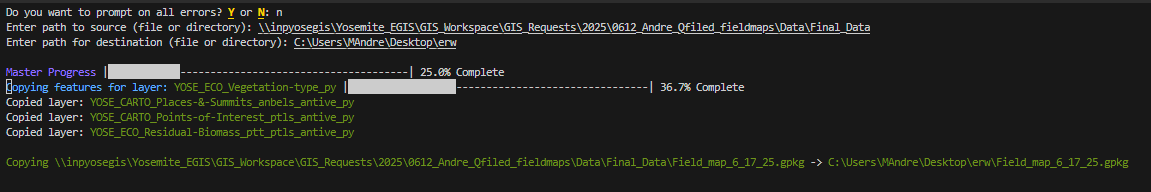
**Standalone Script Mode (True\_Curve.py)**

**Option 1: Terminal (Headless/Batch Mode)**

1. Open a terminal or command prompt.
2. Navigate to the script’s directory.
3. Run the script with CLI:

python True\_Curve.py

Inputs:

* + –error prompt: Notifies if there are any error in the data
  + --input: Path to your input dataset or directory
  + --output: Path to save the output location
  + –segmenting tolerance: if curves cannot be directly linearized the length of curve in meters before each cut to linearize

You may see what looks like unintelligible scripts and character in the terminal if so your terminal doesn’t support escape character so the output will be mangled but conversion will continue properly

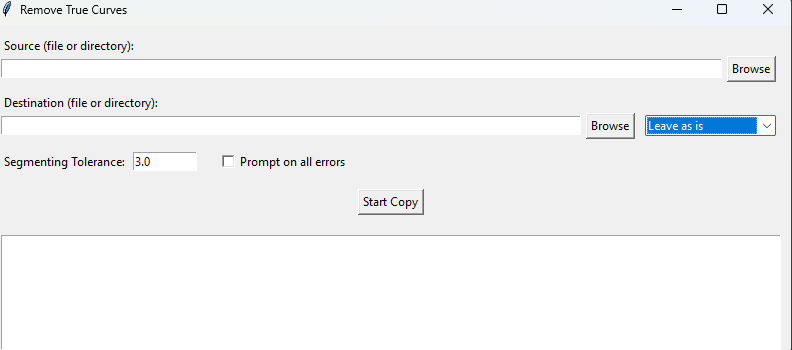
**Option 2: GUI Mode**

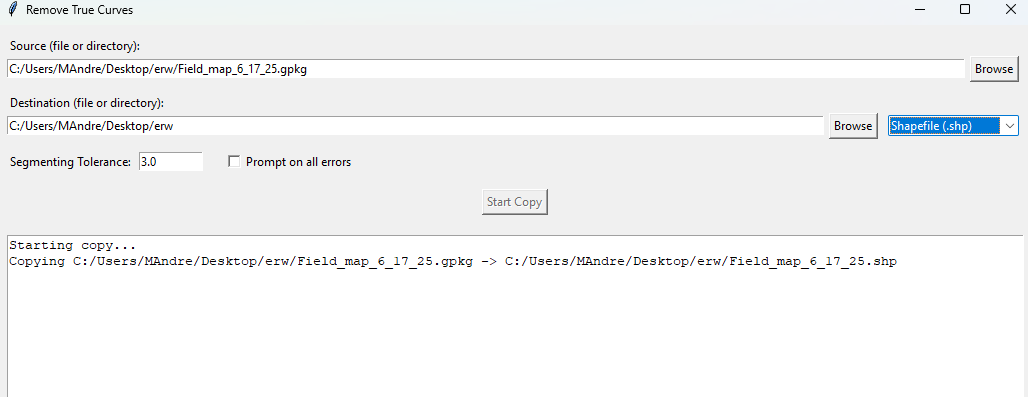
1. In terminal:

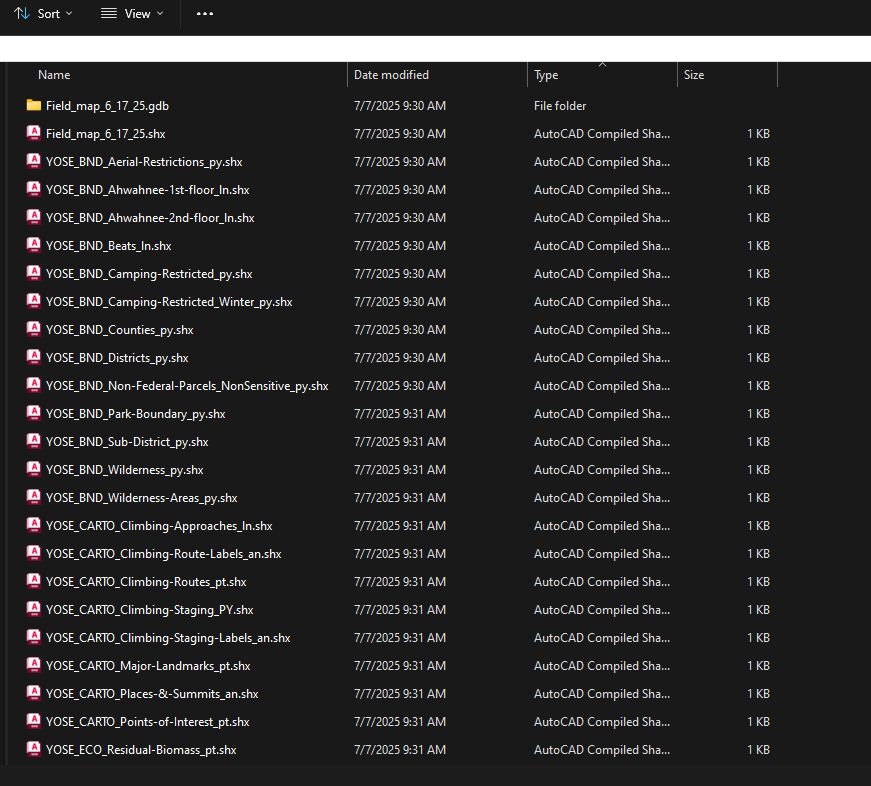
python True\_Curve.py -gui

**Or** double click  
  
Remove True Curves.vbs

1. A graphical interface will launch, allowing you to:
   * Browse and select your **input file** or **directory**
     1. If you wish to select a directory with the browser, you may need to close the first prompt, and a second will open allowing you to select entire folders
   * Choose an **output location & type**
   * Run the tool and monitor progress
   * Review any logged outputs or warnings





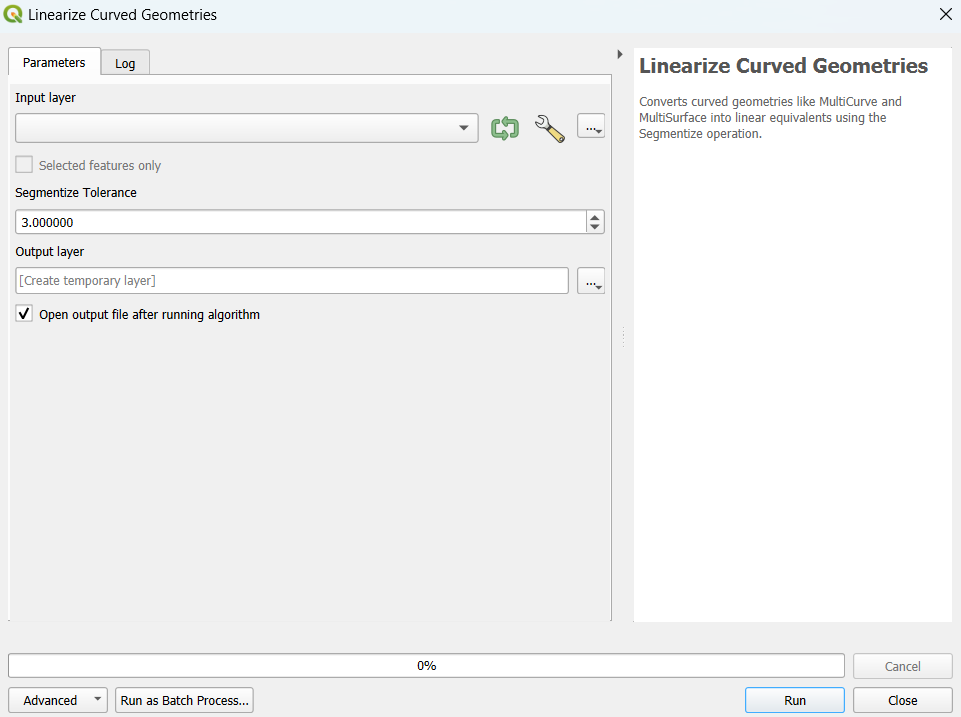
**QGIS Version**

**Installation**

1. Copy QGIS\_Remove\_curved\_geometry.py to your QGIS plugin or processing scripts folder.
2. Restart QGIS if needed.
3. The tool will appear under **Processing Toolbox → Geometry Tools → Linearize Curved Geometries**.

**Usage**

1. Open **Processing Toolbox**.
2. Search for **"Linearize Curved Geometries"**.
3. Select your input layer.
4. Set the [**segmentation tolerance**](#Segemting_tolerance) (e.g., 3.0).
5. Run the algorithm.
6. Output will be saved as a new vector layer.



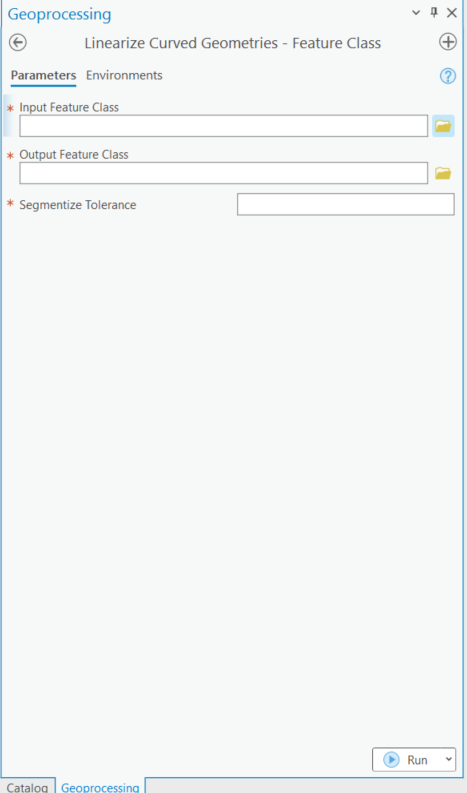
**C. ArcGIS Version**

**Installation**

1. Add ArcGIS\_curves.pyt as a toolbox in ArcGIS Pro:
   * Right-click **Toolboxes** in the Catalog pane.
   * Select **Add Toolbox** → navigate to ArcGIS\_curves.pyt.

**Usage**

1. Open the toolbox from Catalog.
2. Run the tool and fill in:
   * Input feature class
   * Output location
   * [Segmentize tolerance](#Segemting_tolerance) (default: 3.0)

****

**Output**

* Geometry types will be converted to **linear equivalents**:
  + CircularString → LineString
  + CurvePolygon → Polygon
  + MultiCurve, MultiSurface → respective multi-line or multi-polygon
* Output data will conform to **OGR-compliant geometries**.
* Attributes and coordinate systems will be preserved unless altered manually.