Cross Sections & Dimensions 1.0

(User Manual)

Dr Joshphar Kunapo and Dr Kathryn Russell

Waterway Ecosystem Research Group

thewerg.org

Melbourne Waterway Research-Practice Partnership

Technical Report 23.4

Cross Sections & Dimensions 1.0 (User Manual)

Report to: **Rhys Coleman, Melbourne Water**

Published: March 2023

Cite as: *Kunapo J and Russell K (2021) Cross Sections & Dimensions 1.0 (User Manual). Melbourne Waterway Research-Practice Partnership Technical Report 23.4, March 2023.*

Contact author: Dr Joshphar Kunapo (jkunapo@unimelb.edu.au)

Reviewed by: Dr Kathryn Russell (klru@unimelb.edu.au)

Cover photo: Yellingbo Creek, Yellingbo Nature Reserve

Version: 01

Version Number	Purpose/change	Authors(s)	Date
01	Draft for circulation	Kunapo, Russell	2 March 2023
02	Final	Kunapo, Russell	31 May 2024

Contents

1 Introduction

This user manual is for a GIS tool that was developed as part of Melbourne Waterway Research-Practice Partnership for automated extraction of channel metrics. The tool runs using ArcGIS Pro functions as a backend. Thus, it requires ArcGIS Pro to be installed and licenced.

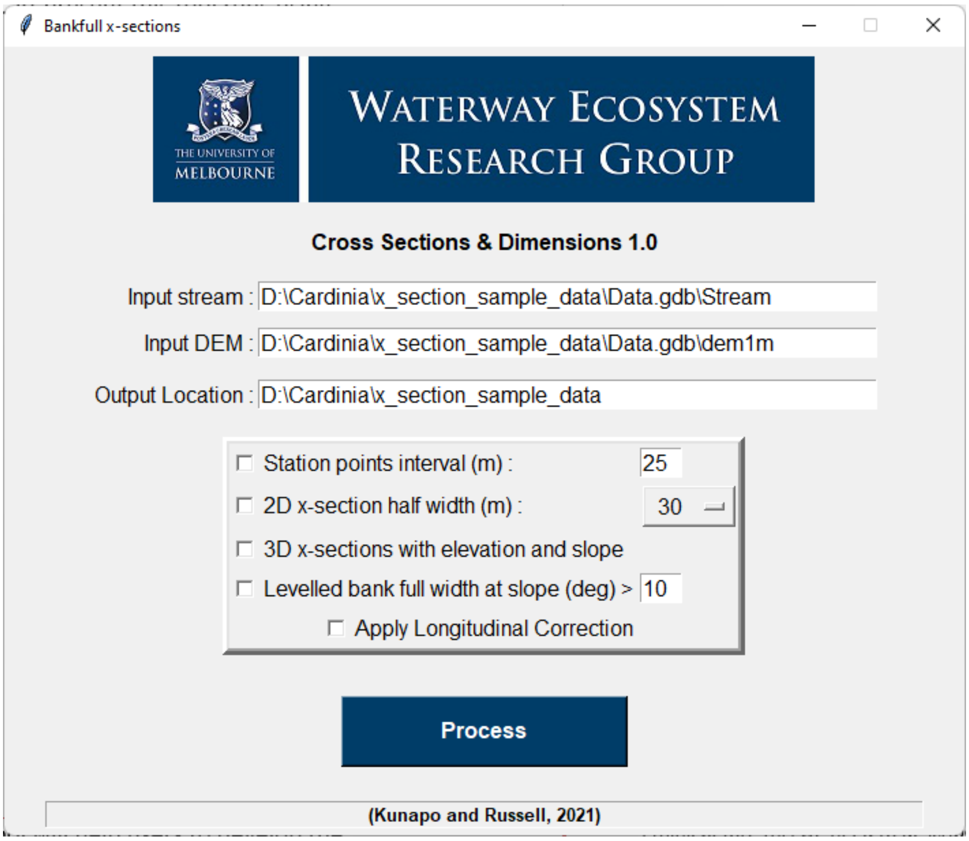
This document covers the following:

- User Interface, and
- Software Installation

The software is available at https://github.com/k-russell/x_sections_tool. It can be downloaded directly or accessed using git.

2 User Interface

Figure 1 below shows the Graphic User Interface (GUI) of the Cross Sections and Dimensions tool. The tool requires only two key datasets as input. They are 1) the stream centreline and, 2) the digital elevation model (DEM). Users of this tool are expected to understand and interpret the DEM-based slope data to identify thresholds that form the toe and top of bank in their study area. At present this tool is limited to work with GDA_1994_MGA_Zone_55 projection. Below sections describe input and output options and the parameters of the tool.


Bankfull x-sections
THE UNIVERSITY OF MELBOURNE
WATERWAY ECOSYSTEM RESEARCH GROUP

Cross Sections & Dimensions 1.0

Input stream : D:\Cardinia\x_section_sample_data\Data.gdb\Stream

Input DEM : D:\Cardinia\x_section_sample_data\Data.gdb\dem1m

Output Location : D:\Cardinia\x_section_sample_data

□ Station points interval (m) : 25

□ 2D x-section half width (m) : 30

□ 3D x-sections with elevation and slope

□ Levelled bank full width at slope (deg) > 10

□ Apply Longitudinal Correction

(Kunapo and Russell, 2021)

Figure 1 Graphic User Interface (GUI) of the tool.

2.1 Inputs

Parameters	Values	File format	Description
Input stream	Path to the stream centreline feature class.	ESRI feature geodatabase	• The tool starts at the centreline location to build various outputs. This stream centreline is crucial for this tool. The centreline must be within 5 metres of the lowest elevation point (i.e., thalweg) of the stream. • Users may use DEM, hillshade and/or slope data to generate and/or correct the centreline before using it with this tool. • The tool will correct the exact lowest point of cross-sections if the centreline is within 5 meters of the thalweg.
Input DEM	Path to the DEM raster.	ESRI grid, ESRI ASCII grid, TIFF or ESRI feature geodatabase	• This tool was developed to work with a 1m resolution DEM.
Output Location	Output folder location.	Directory path	• Please avoid long paths and paths with spaces. GIS functions often fail with such situations, and it varies from function to function. • Directory must exist – it may need to be created in the file system before running the tool.

2.2 Outputs

All outputs are stored in a geodatabase at <**Output location>\x_sec_outputs.gdb\**

IMPORTANT: Once generated, outputs will be re-used at each rerun of the tool (with same parameters and output location) to save time. Please delete manually any outputs you need to recreate.

Station points interval (m):

This option will place station points (Figure 2) along the stream centreline at a defined interval in metres.

- Points are added in the direction of the line for each branch of the stream network.
- First, the stream centreline segments are defined, and a line feature class is created named **mwstr_segments**. Separate segments are defined between confluences (junctions of more than two stream centrelines). Otherwise, centrelines which join end to end are merged together. A field is added named **"seg_id"** which gives each segment a unique ID. This field will be carried across all cross-sectional outputs.
- The process then creates a point feature class named **station_points_<interval>m.**
- The process adds a field named "**x_sec_id**" with unique values for each station point. This field will be carried across all cross-sectional outputs.

Parameters	Values (m)	Default	Description
Station points interval (m)	5 to 1000	25	Spacing of station points (where cross-sections will be placed) along centreline

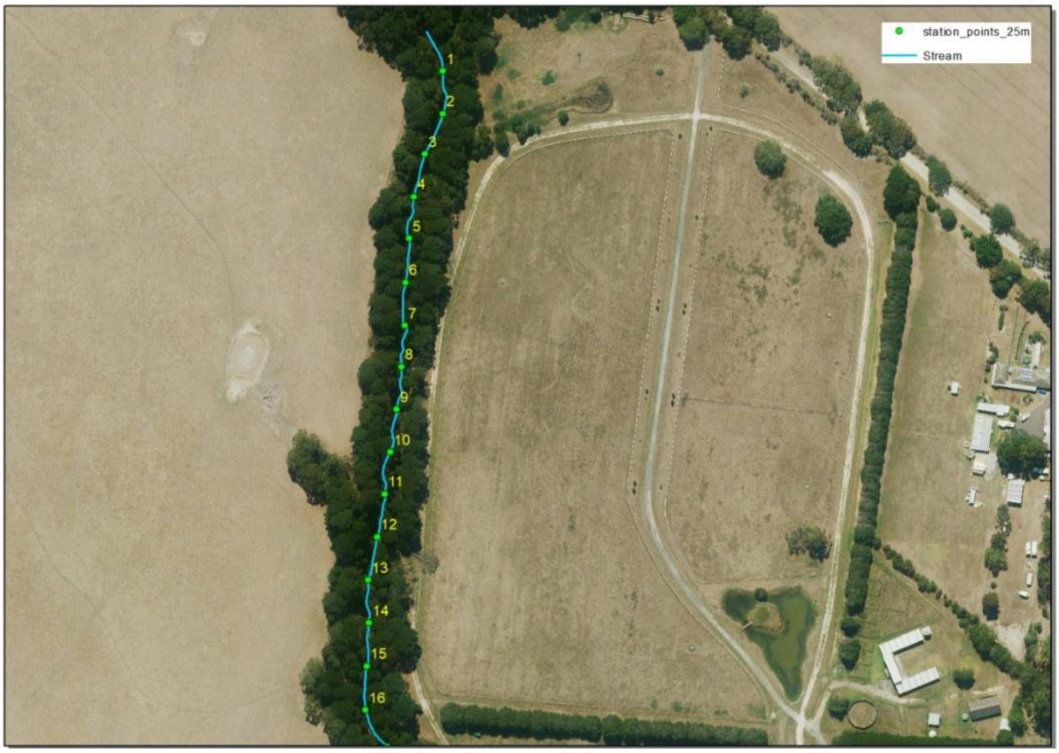


Figure 2 Typical station points generated along the input stream centreline.

2D x-section half width (m):

This option generates cross-sections at station points (Figure 4).

- The process creates a line feature class named **x_sec_i< interval >_w< half width>.**
- Users need to define the half width of the cross sections which will be used as the maximum buffer distance from the stream centreline (see Figure 3) in constructing the cross sections.
- The output cross-sections are not fully perpendicular to the stream network, rather they follow a buffer distance from the centreline (Figure 3). This method avoids intersecting cross sections.
- The cross sections are generated from left to right in the direction of stream. Figure 3 and 4 show typical cross sections developed for 40m half-width (80m total width).
- **Tip**: Estimate maximum width of the top of bank of your study area and provide half width accordingly to get better results rather than leaving the defaults. Providing a larger width than required may cause overlap issues at meandering areas and errors in levelling the cross sections.

Parameters	Values (m)	Default	Output name
2D x-section half width (m)	20 to 100	30	x_sec_i< interval >_w< half width>

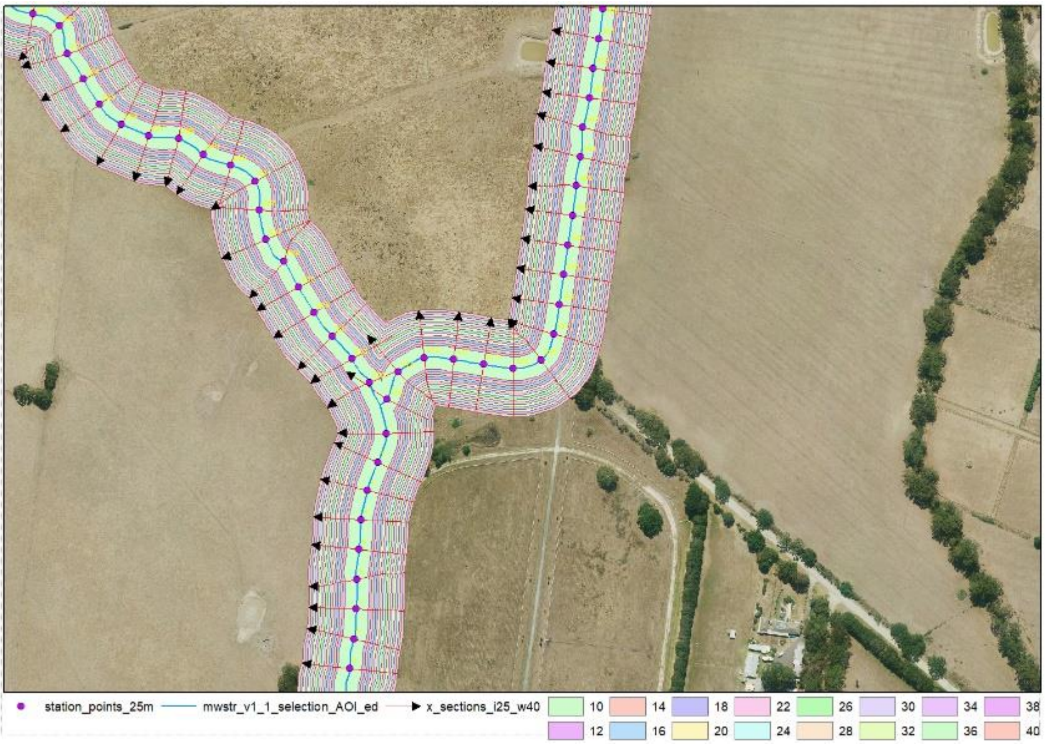


Figure 3: Figure showing how buffer rings are used in constructing cross sections.

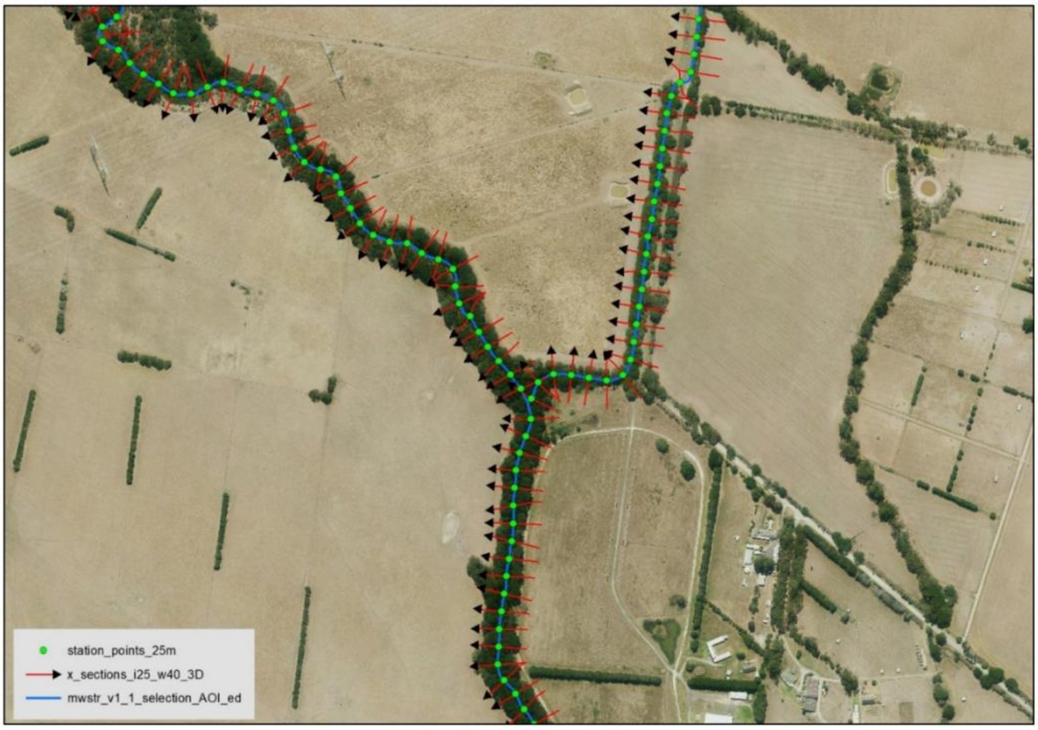


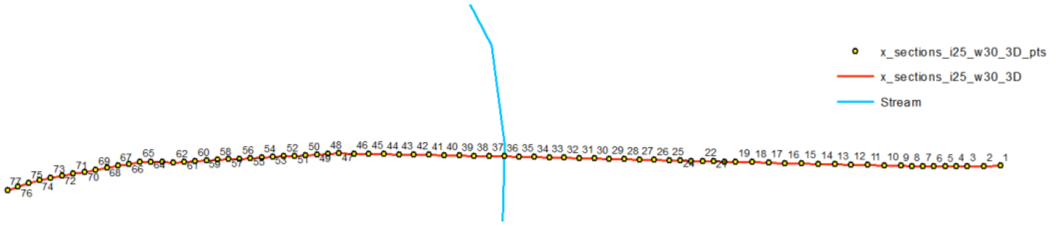
Figure 4: Typical cross-sections generated by the tool.

3D x-sections with elevation and slope:

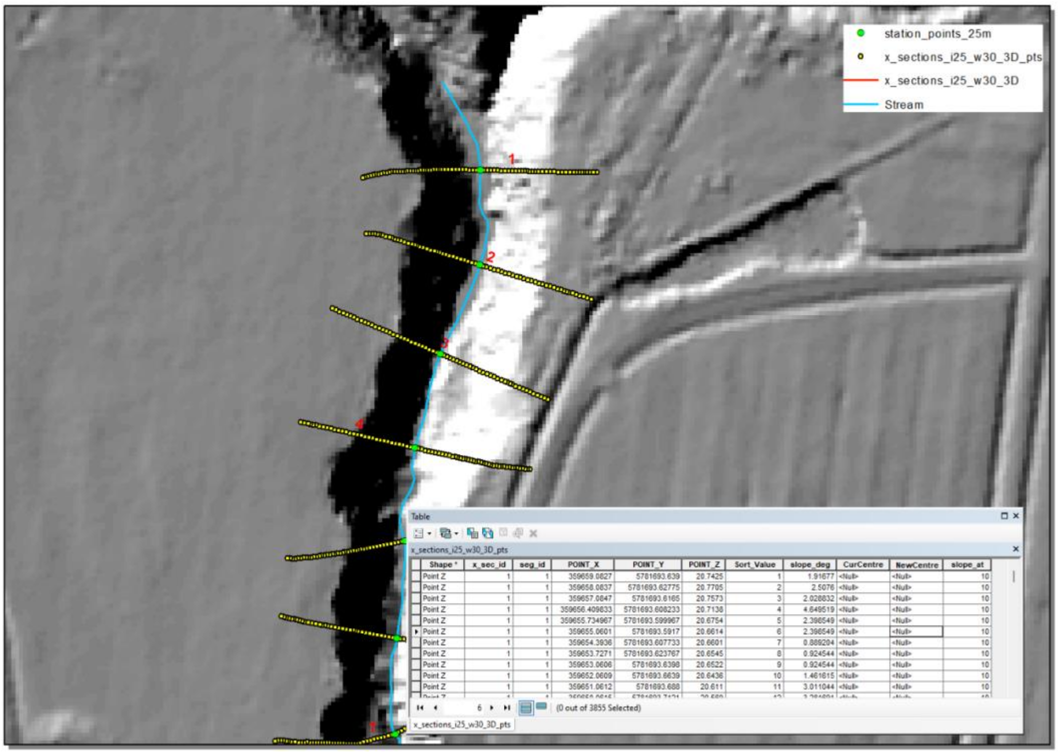
This option will first use the input DEM to develop a slope raster in degrees named **slope_deg**.

It will then use the above cross sections, the input DEM and the slope raster to develop two outputs (Figure 5):

- 3D cross-section lines – **x_sec_i<interval>_w<half width>_3D.** Lines with Z values extracted from the input DEM.
- 3D cross-section points - **x_sec_i<interval>_w< half width>_3D_pts:** Points along each 3D cross-section line. This points output is used to develop levelled bankfull cross sections and levelled bank-full width polygons. The following fields are added:
    - **POINT_X**: X-coordinate of the point.
    - **POINT_Y**: Y-coordinate of the point.
    - **POINT_Z**: Z-coordinate of the point.
    - **Sort_Value**: Point sort order by position along cross-section, starting from left to right to the direction of stream.



    - **slope_deg**: Slope in degrees for each point.
    - **CurCentre**: Point at the centre of the cross-section (= station point location)
    - **NewCentre**: The lowest elevation point in the cross-section (within 5m of CurCentre).



x_sections_i25_w30_3D_pts

Shape *	x_sec_id	seg_id	POINT_X	POINT_Y	POINT_Z	Sort_Value	slope_deg	CurCentre	NewCentre	slope_at
Point Z	1	1	359659.0827	5781693.639	20.7425	1	1.91677	<Null>	<Null>	10
Point Z	1	1	359658.0837	5781693.62775	20.7705	2	2.5076	<Null>	<Null>	10
Point Z	1	1	359657.0847	5781693.6165	20.7573	3	2.028832	<Null>	<Null>	10
Point Z	1	1	359656.409833	5781693.608233	20.7138	4	4.649519	<Null>	<Null>	10
Point Z	1	1	359655.734967	5781693.599967	20.6754	5	2.398549	<Null>	<Null>	10
Point Z	1	1	359655.0601	5781693.5917	20.6614	6	2.398549	<Null>	<Null>	10
Point Z	1	1	359654.3936	5781693.607733	20.6601	7	0.889204	<Null>	<Null>	10
Point Z	1	1	359653.7271	5781693.623767	20.6545	8	0.924544	<Null>	<Null>	10
Point Z	1	1	359653.0606	5781693.6398	20.6522	9	0.924544	<Null>	<Null>	10
Point Z	1	1	359652.0609	5781693.6639	20.6436	10	1.461615	<Null>	<Null>	10
Point Z	1	1	359651.0612	5781693.688	20.611	11	3.011044	<Null>	<Null>	10

6 (0 out of 3855 Selected)

x_sections_i25_w30_3D_pts

Figure 5: Typical 3D cross-sectional information generated by the tool

Levelled bank full width at slope (deg) >:

This option creates levelled bankfull cross-sections and points, and levelled bankfull extent polygon, where the bankfull elevation is based on the top of the lower of the two banks as defined by a slope threshold.

While all the previous steps can be automated without any human interaction, this step requires the user’s judgment and may require rerunning the tool until desired results are achieved.

Users of the tool may use a slope raster to identify a threshold of slope (in degrees) where top of bank can be identified.

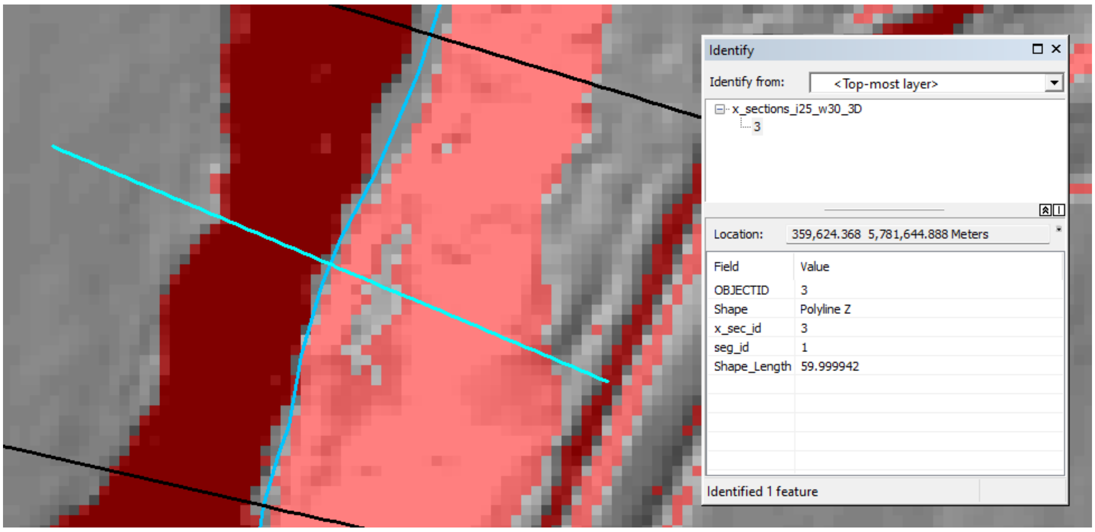
The tool performs the following steps using 3D x-section points and their attributes:

Step 1: Get high slope points (>= defined slope threshold).

Step 2: For each cross-section, use NewCentre, minimum and maximum of sort values (Sort_Value) to separate left and right points.

Step 3: Get max (POINT_Z) for each side to get initial estimate of levelled Z (minimum of left and right max (POINT_Z)). Remove points which are above this levelled Z.

Step 4: For the above refined collection of points, identify any slope gaps. For examples, below selected cross-section (x_sec_id = 3) has a slope gap (flat areas) along the cross-section due to stepped elevations banks. The first gap is acceptable, but not the second gap which is at the top of bank.



Identify from: <Top-most layer>

x_sections_i25_w30_3D
3

Location: 359,624.368 5,781,644.888 Meters

Field	Value
OBJECTID	3
Shape	Polyline Z
x_sec_id	3
seg_id	1
Shape_Length	59.999942

Identified 1 feature

- Thus, logic has been built to accept only the first gap along the cross-section and the gap can't exceed more than 2 sort values.
- If the gap is larger than 2 points it will stop searching further and it recalculates levelled Z.

Step 5: Re-construct levelled cross-section lines and points (Figure 6) using the new bounds and constructs the bank full width polygon.

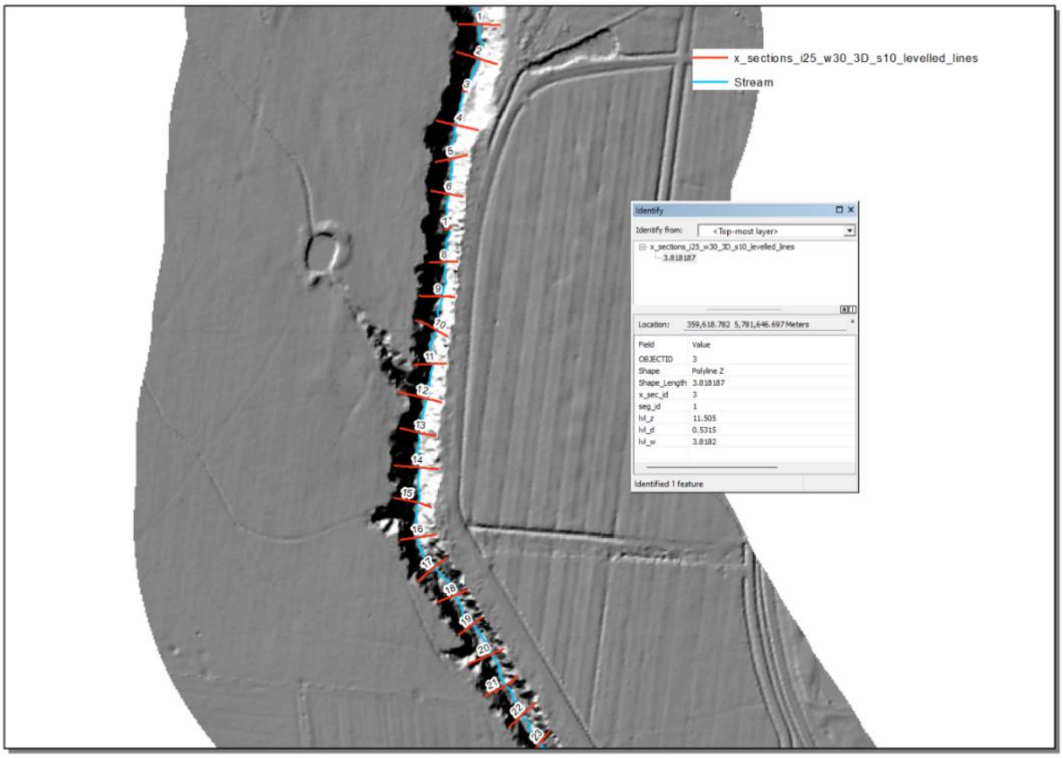
The flowing layers are created:

- Levelled cross-section points – **x_sec_i<interval>_w<half width>_3D_s<slope>_levelled_pts**

  The following fields are added:
  - **Slope_at** – slope threshold used to identify top of bank
  - **Lvl_z** – bankfull elevation
- Levelled cross-section lines – **x_sec_i<interval>_w<half width>_3D_s<slope>_levelled_lines**

  The following fields are added:
  - **lvl_d** – bankfull depth
  - **lvl_w** – bankfull width
- Levelled cross-section polygons - **x_sec_i<interval>_w<half width>_3D_s<slope>_levelled_poly**

Parameters	Values (m)	Default	Description
Levelled bank full width at slope (deg)	5 to 15	10	Slope threshold for identification of top of bank



Identify

Identify from: <Top-most layer>

x_sections_i25_w30_3D_s10_levelled_lines
3.818187

Location: 359,618.782 5,781,646.697 Meters

Field	Value
OBJECTID	3
Shape	Polyline Z
Shape_Length	3.818187
x_sec_id	3
seg_id	1
lvl_z	11.505
lvl_d	0.5315
lvl_w	3.8182

Identified 1 feature

Figure 6: Typical levelled 3D cross-sections with their elevation, depth, and width. Please note 3rd cross-section shows slope-gap threshold application affect.

Apply Longitudinal Correction

This option will enable longitudinal correction to the above levelled cross-sections to smooth irregularities in bankfull width between neighbouring cross-sections. For example, cross-section 3 in figure 6 is unusual when compared to its upstream downstream cross-sections.

- The following logic is applied to correct such irregularities considering 10 cross-sections at a time to look upstream/downstream widths and their coherence:
    - **Step 1**: Remove severe outliers. For each point look at its previous and next bankfull elevations. If its elevation is a sudden change (+/- 3m) from its neighbours, it will be flagged and ignored.
    - **Step 2**: run a Linear Regression (LR) model between bankfull elevation and cross-section ID, for ten cross-sections at a time (ignoring any severe outliers already flagged). Get predicted regression values for each cross-section. If the difference between original and predicted bankfull elevation is more than 1 m, flag as an outlier and use the predicted bankfull elevation.

- **Step 3:** Re-construct levelled cross-section lines and points (Figure 7) using the new bankfull elevation (lvl_z) applied to original cross-sections and also construct the bank full width polygon (Figure 8).

Cross-section 3 in figure 7 below is an example of the effect of longitudinal correction. Longitudinal correction of outliers may or may not be desirable depending on the application. The user's judgment is required here.

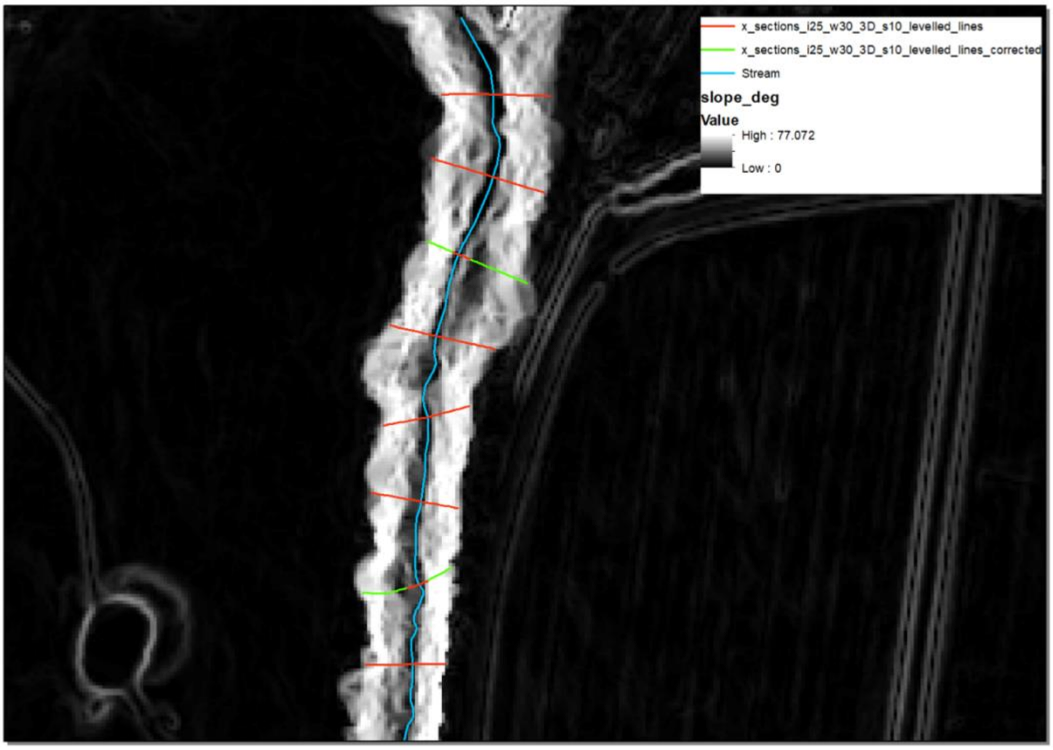

x_sections_i25_w30_3D_s10_levelled_lines
x_sections_i25_w30_3D_s10_levelled_lines_corrected
Stream
slope_deg
Value
High : 77.072
Low : 0

Figure 7: Typical levelled 3D cross-sections with longitudinal correction applied.

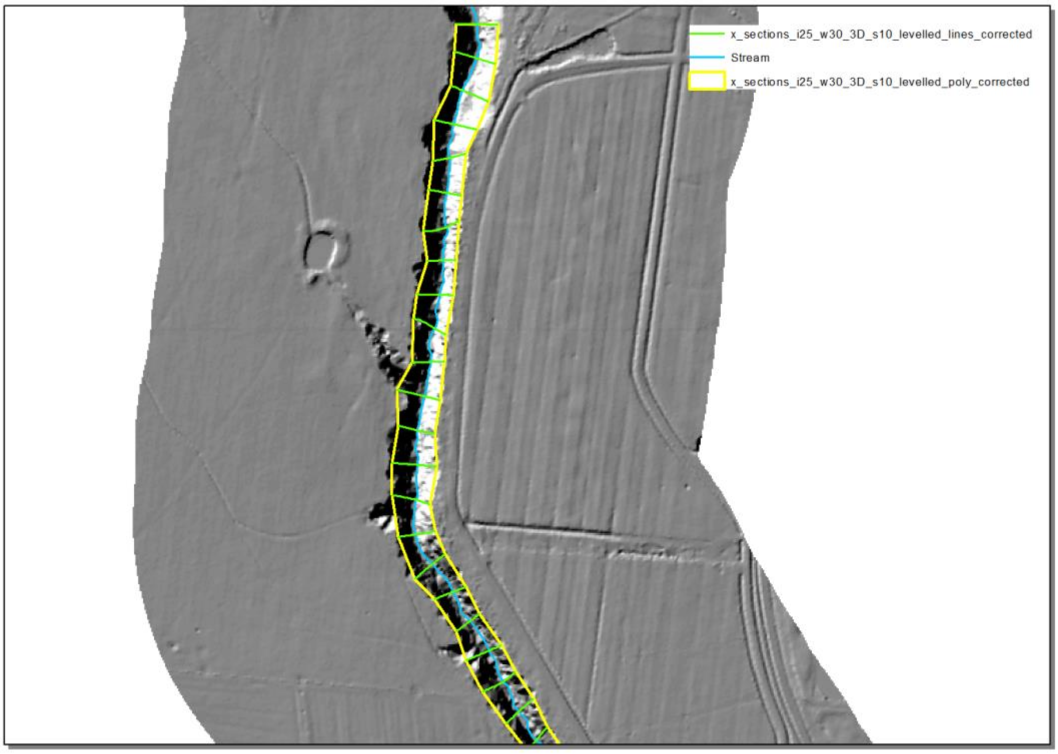

x_sections_i25_w30_3D_s10_levelled_lines_corrected
Stream
x_sections_i25_w30_3D_s10_levelled_poly_corrected

Figure 8: Typical bankfull width polygon constructed using levelled cross-sections.

3 Software installation

3.1 System Requirements:

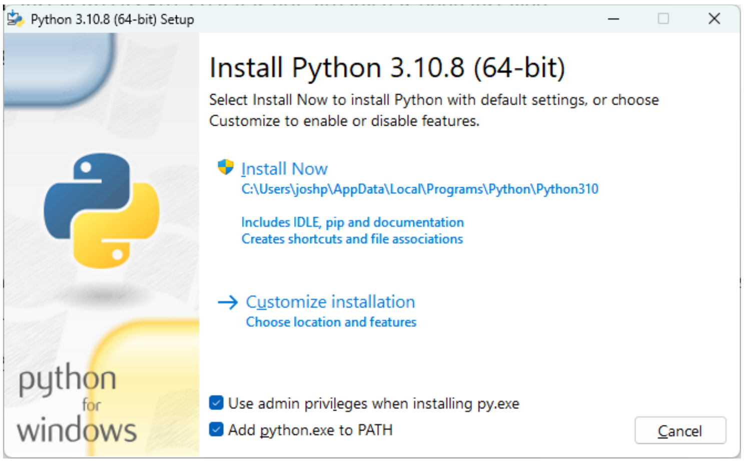
ArcGIS Pro must be installed prior to the installation of this Tool.

3.2 Installation:

Step 1: Install ArcGIS Pro 3.0 if it is not already has been installed.

Step 2: Install Python 3.10 or greater.

https://www.python.org/ftp/python/3.10.8/python-3.10.8-amd64.exe



Python 3.10.8 (64-bit) Setup

Install Python 3.10.8 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\joshp\AppData\Local\Programs\Python\Python310

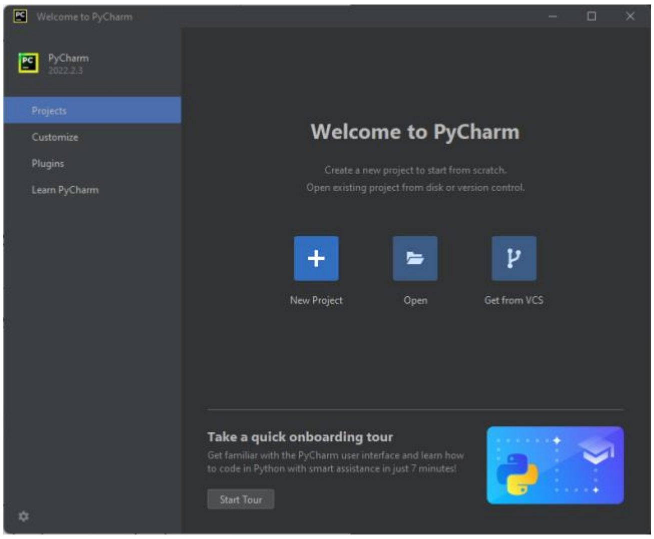
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

Step 3: Install latest Pycharm Community Edition from https://www.jetbrains.com/pycharm/

https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows&code=PCC



Welcome to PyCharm

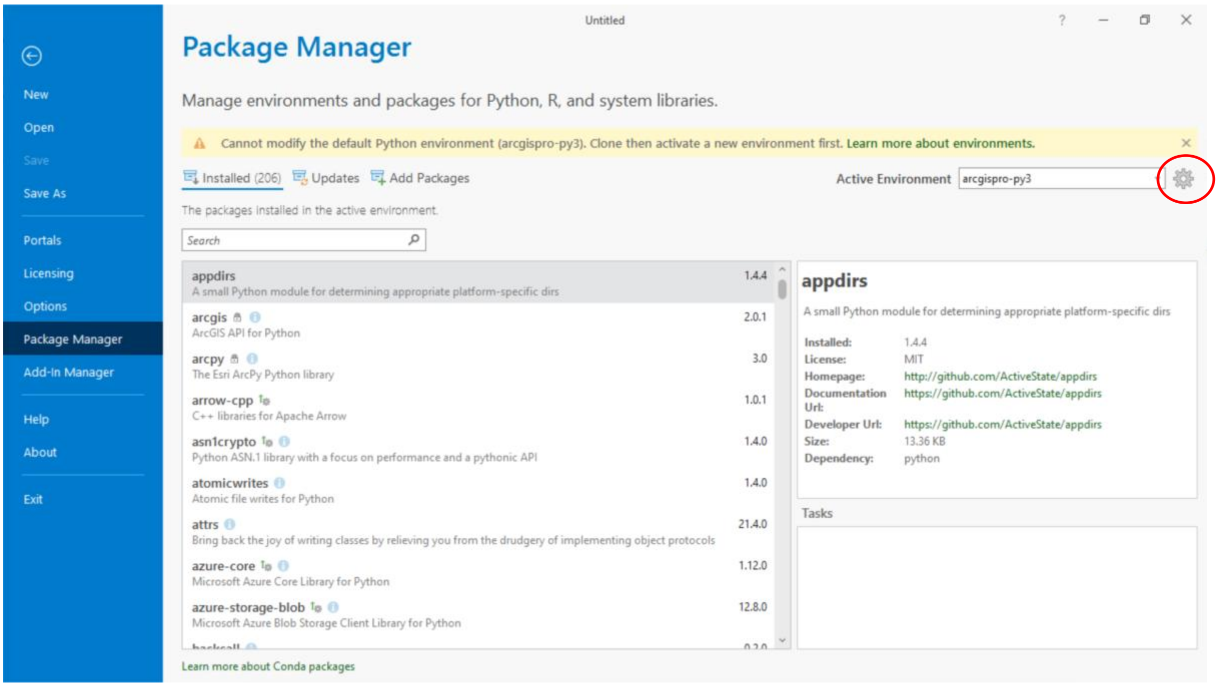
Create a new project to start from scratch.
Open existing project from disk or version control.

Take a quick onboarding tour

Get familiar with the PyCharm user interface and learn how to code in Python with smart assistance in just 7 minutes!

Step 4: In ArcGIS Pro, Clone arcgispro-py3 environment, install Scikit-learn package in clone environment.

Go to Project -> Package Manager, then next to Active Environment click the settings gear wheel.



Package Manager

Manage environments and packages for Python, R, and system libraries.

Cannot modify the default Python environment (arcgispro-py3). Clone then activate a new environment first. Learn more about environments.

Active Environment: arcgispro-py3

The packages installed in the active environment.

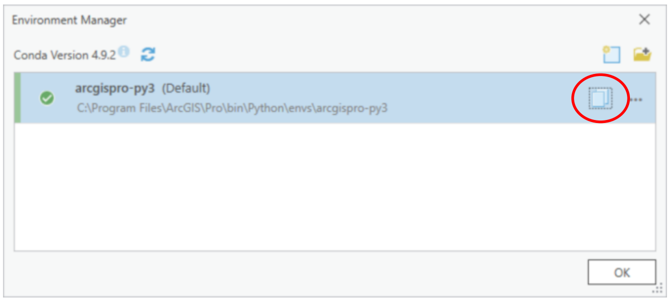
appdirs	A small Python module for determining appropriate platform-specific dirs	1.4.4
arcgis	ArcGIS API for Python	2.0.1
arcpy	The Esri ArcPy Python library	3.0
arrow-cpp	C++ libraries for Apache Arrow	1.0.1
asn1crypto	Python ASN.1 library with a focus on performance and a pythonic API	1.4.0
atomicwrites	Atomic file writes for Python	1.4.0
attrs	Bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols	21.4.0
azure-core	Microsoft Azure Core Library for Python	1.12.0
azure-storage-blob	Microsoft Azure Blob Storage Client Library for Python	12.8.0
backcall		0.2.0

appdirs

A small Python module for determining appropriate platform-specific dirs

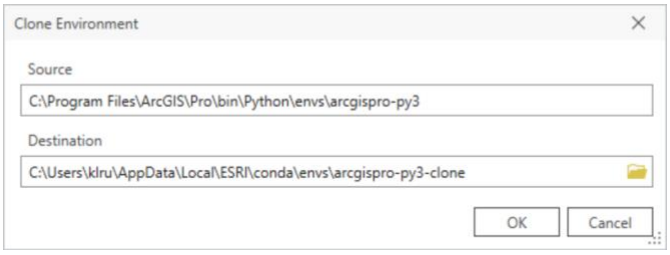
Installed:	1.4.4
License:	MIT
Homepage:	http://github.com/ActiveState/appdirs
Documentation Url:	https://github.com/ActiveState/appdirs
Developer Url:	https://github.com/ActiveState/appdirs
Size:	13.36 KB
Dependency:	python

Learn more about Conda packages



Conda Version 4.9.2

arcgispro-py3 (Default)
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3

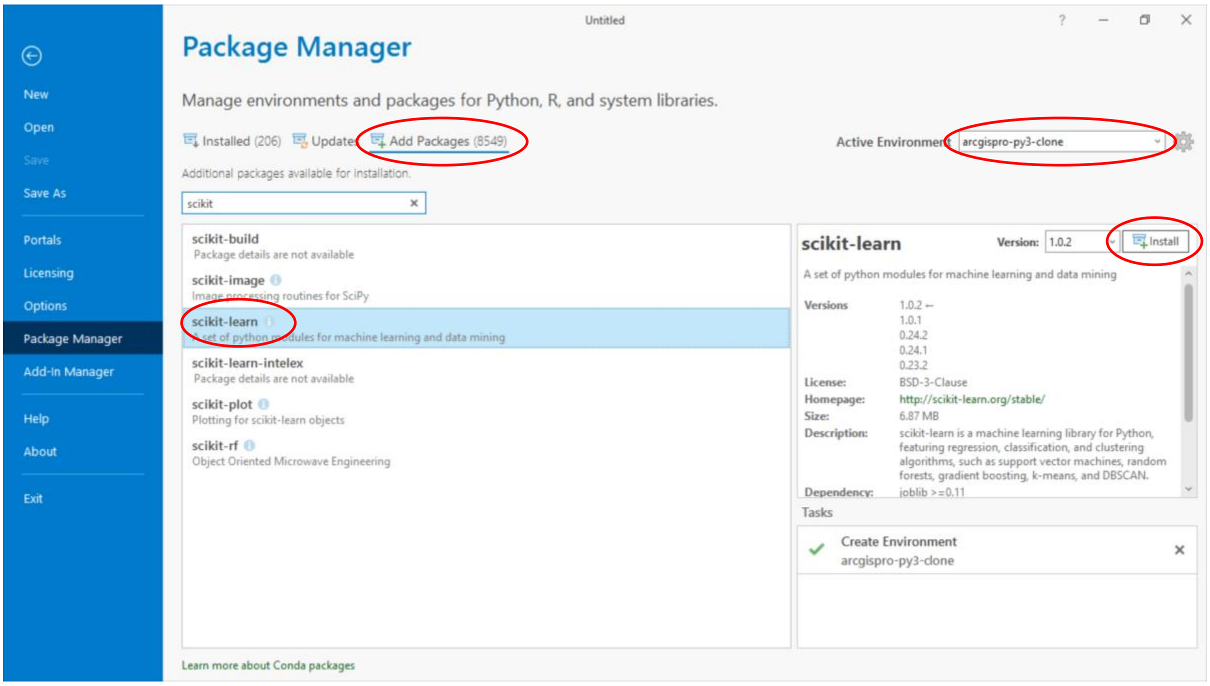


Source: C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3

Destination: C:\Users\klru\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone

Cloning can take a few minutes.

Change Active Environment to the cloned environment, then go to Add Packages and search for scikit-learn and click install.



Package Manager

Manage environments and packages for Python, R, and system libraries.

Installed (206) Updates Add Packages (8549)

Active Environment: arcgispro-py3-clone

Additional packages available for installation.

scikit

scikit-build
Package details are not available

scikit-image
Image processing routines for SciPy

scikit-learn
A set of python modules for machine learning and data mining

scikit-learn-intelex
Package details are not available

scikit-plot
Plotting for scikit-learn objects

scikit-rf
Object Oriented Microwave Engineering

scikit-learn Version: 1.0.2 Install

A set of python modules for machine learning and data mining

Versions: 1.0.2 ←, 1.0.1, 0.24.2, 0.24.1, 0.23.2
License: BSD-3-Clause
Homepage: http://scikit-learn.org/stable/
Size: 6.87 MB
Description: scikit-learn is a machine learning library for Python, featuring regression, classification, and clustering algorithms, such as support vector machines, random forests, gradient boosting, k-means, and DBSCAN.
Dependency: joblib >=0.11

Tasks

✓ Create Environment
arcgispro-py3-clone

Learn more about Conda packages

You can now exit ArcGIS Pro.

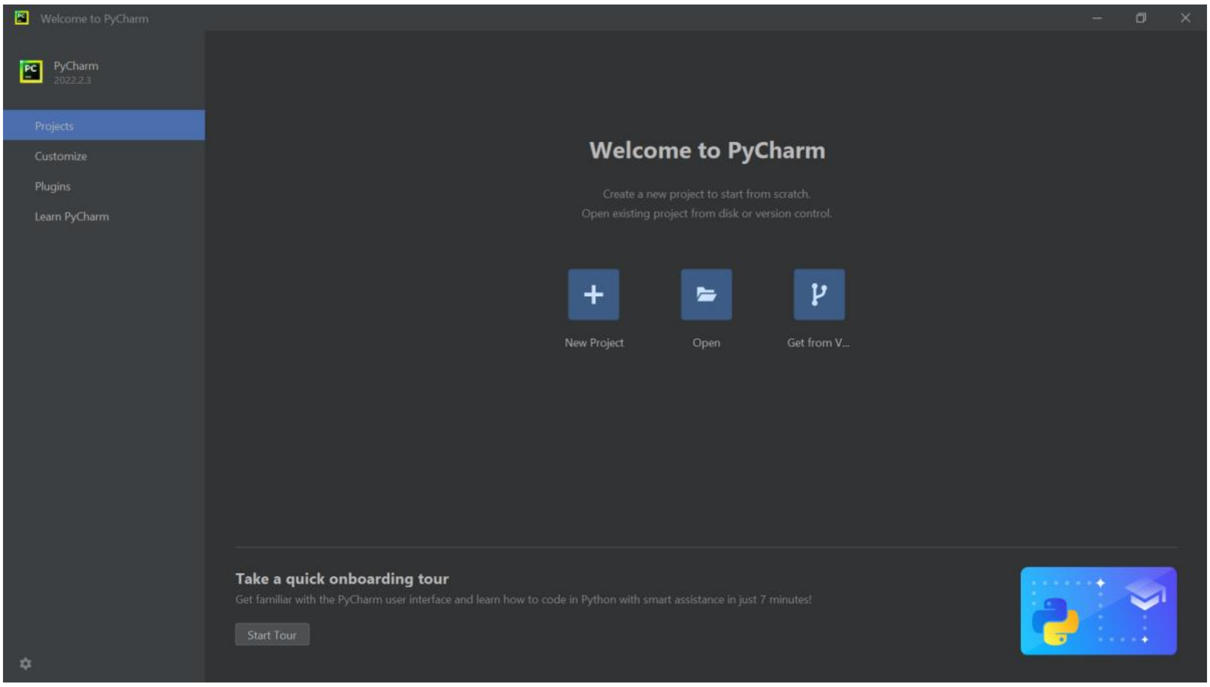
Step 5: In PyCharm, Configure Python interpreter for ArcGIS Pro (arcgispro-py3-clone)

When you first launch PyCharm, click New Project.

Create a new project in whatever location you like.

Under Python Interpreter, select “Previously configured interpreter” then select Environment: “Existing” and navigate to the python.exe file in the cloned environment. The path in our example is:

C:\Users\klru\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\python.exe

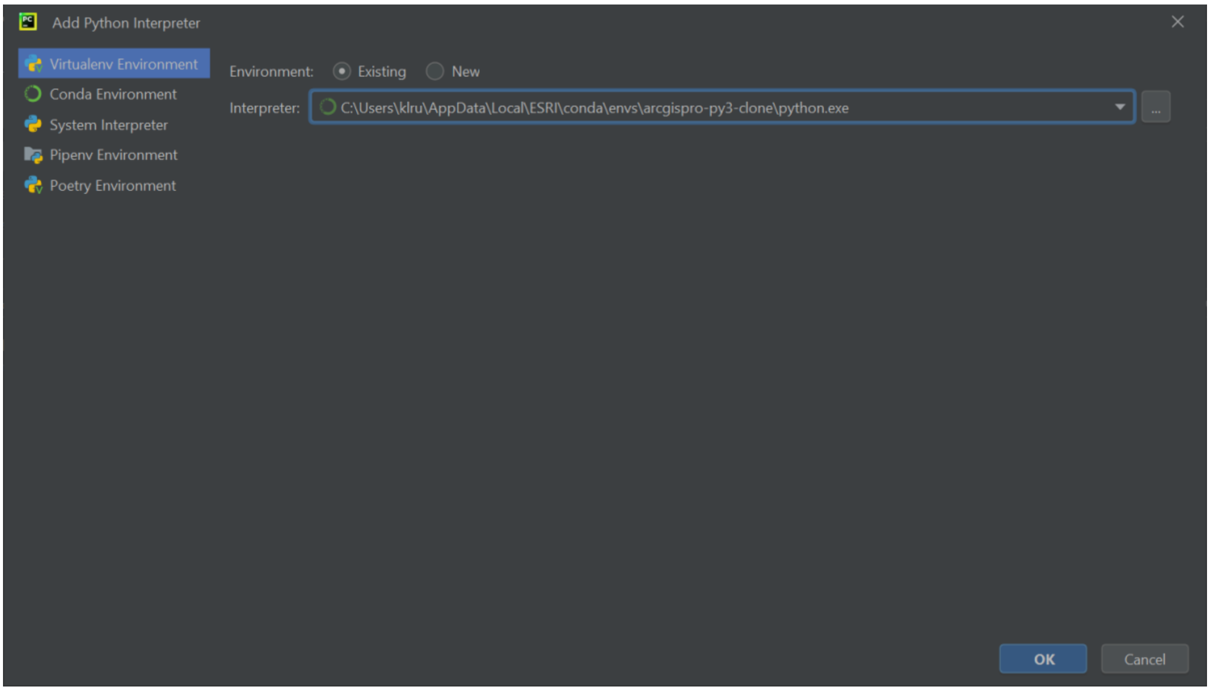


Welcome to PyCharm

Create a new project to start from scratch.
Open existing project from disk or version control.

Take a quick onboarding tour

Get familiar with the PyCharm user interface and learn how to code in Python with smart assistance in just 7 minutes!



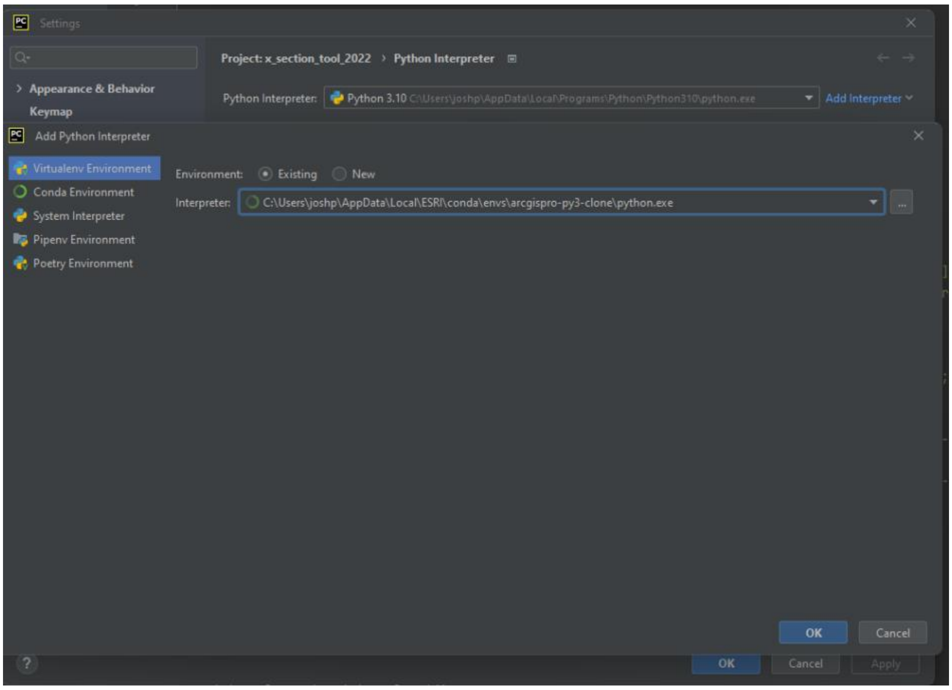
Environment: ● Existing ○ New

Interpreter: C:\Users\klru\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\python.exe

If you have previously configured PyCharm, change the Python interpreter as follows.

After PyCharm launches, click the gear at the upper right corner of the screen (or press Ctrl-Alt-S) to open the settings dialog. You can set up PyCharm with multiple Python interpreters, and we want to make the default setup for ArcGIS Pro (you can choose others though, which is handy!)

1. *Select Project: (project name) and then Python Interpreter at left*
2. *Click Add Interpreter at upper right and select Add Local Interpreter*
3. *Select the Existing environment radio button*
4. *For Interpreter: click the ... icon at right and navigate to and select the python.exe for Pro: C:\Users\klru\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\python.exe*
5. *Click OK.*



Project: x_section_tool_2022 › Python Interpreter

Python Interpreter: Python 3.10 C:\Users\joshp\AppData\Local\Programs\Python\Python310\python.exe

Environment: ● Existing ○ New

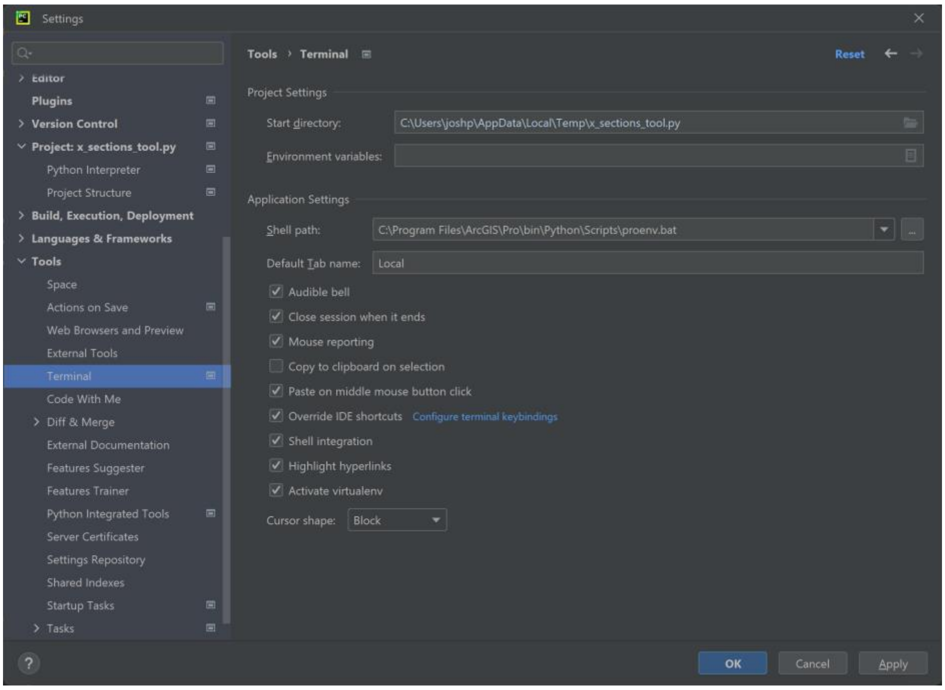
Interpreter: C:\Users\joshp\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\python.exe

Configure PyCharm Terminal for ArcGIS Pro

PyCharm supports a terminal window for use in manipulating the conda environment, running shell tools, editors, or whatever else. The standard shell environment for ArcGIS Pro is the one you launch by with the Windows shortcut Python Command Prompt - which launches a shell with the proenv.bat script.

Bring up to the same settings dialog used above to set the Python interpreters

Still in Settings, at left, find Tools > Terminal



Tools › Terminal

Project Settings

Start directory: C:\Users\joshp\AppData\Local\Temp\x_sections_tool.py

Environment variables:

Application Settings

Shell path: C:\Program Files\ArcGIS\Pro\bin\Python\Scripts\proenv.bat

Default Tab name: Local

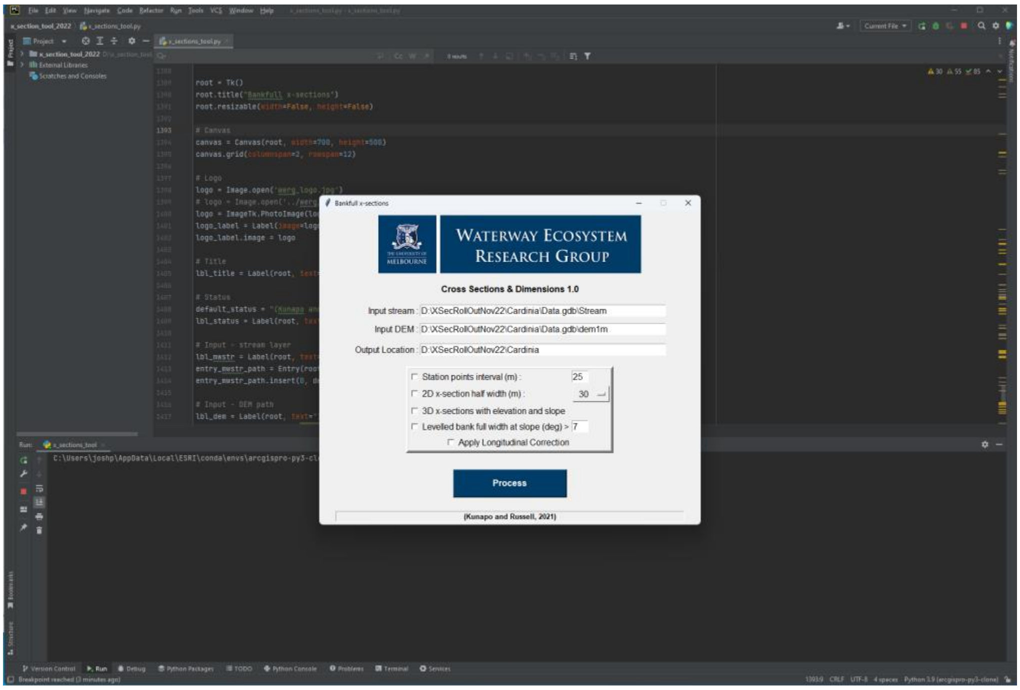
☑ Audible bell
☑ Close session when it ends
☑ Mouse reporting
□ Copy to clipboard on selection
☑ Paste on middle mouse button click
☑ Override IDE shortcuts
☑ Shell integration
☑ Highlight hyperlinks
☑ Activate virtualenv

Cursor shape: Block

For Shell Path: click the ... icon at right and navigate to and select

C:\Program Files\ArcGIS\Pro\bin\Python\Scripts\proenv.bat

Step 6: Open "D:\x_section_tool_2022\x_sections_tool.py" in Pycharm code editor and run.



```
root = Tk()
root.title('Bankfull x-sections')
root.resizable(width=False, height=False)

# Canvas
canvas = Canvas(root, width=700, height=500)
canvas.grid(columnspan=2, rowspan=12)

# Logo
logo = Image.open('werg_logo.jpg')
# logo = Image.open('../werg
logo = ImageTk.PhotoImage(lo
logo_label = Label(image=log
logo_label.image = logo

# Title
lbl_title = Label(root, text

# Status
default_status = "(Kunapo an
lbl_status = Label(root, tex

# Input - stream layer
lbl_mwstr = Label(root, text
entry_mwstr_path = Entry(roo
entry_mwstr_path.insert(0, d

# Input - DEM path
lbl_dem = Label(root, text="
```

```
C:\Users\joshp\AppData\Local\ESRI\conda\envs\arcgispro-py3-cl
```

Bankfull x-sections

WATERWAY ECOSYSTEM RESEARCH GROUP

Cross Sections & Dimensions 1.0

Input stream : D:\XSecRollOutNov22\Cardinia\Data.gdb\Stream

Input DEM : D:\XSecRollOutNov22\Cardinia\Data.gdb\dem1m

Output Location : D:\XSecRollOutNov22\Cardinia

☐ Station points interval (m) : 25

☐ 2D x-section half width (m) : 30

☐ 3D x-sections with elevation and slope

☐ Levelled bank full width at slope (deg) > 7

☐ Apply Longitudinal Correction

3.3 Tutorial Dataset:

A sample dataset (<Drive>:\x_section_tool_2022\x_sec_sample_data) has been provided to test this tool. Run the tool using this sample data at least once before using your own data.