Mapmaking with Tsunami Modeling Results and QGIS-Py Tool

Set up and application of QGIS map processing scripts for Palau

Jai Tiarks, PhD

Overview

These set of scripts process the raw QGIS mesh data layer (NetCDF file e.g., tsunami inundation results) into a polygon layer that can be used for a draft of your Tsumani Ready (TR) map.

Overview

1	2	3	4	5	6	7
Mesh Contours	Dissolve Polygons	Extract Boundary	Solid Polygon	Final Dissolve	Layer Cleanup	Add Basemaps

All scripts share: BASE_NAME · OUTPUT_DIR · OUTPUT_FORMAT · LAYER_TRACKER (which enables layer tracking across scripts)

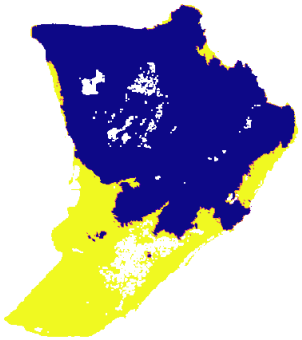
Input: NetCDF mesh file → Output: contour lines + final polygon (GPKG / SHP / GeoJSON)

Two ways to run:

- **Standalone script** — use any individual script in the QGIS Python Console
- **Via Main script** — edit Configure script, then run the Main script

What does each script do?

Modeling tsunami inundation results for Angaur, Palau



(Sannikova and Moore, 2024)

raw mesh layer → polygon layer

Polygon layer in QGIS

1	2	3	4	5	6	7
Mesh Contours	Dissolve Polygons	Extract Boundary	Solid Polygon	Final Dissolve	Layer Cleanup	Add Basemaps

From Mesh to Clean Boundary

1 Mesh Contour Export

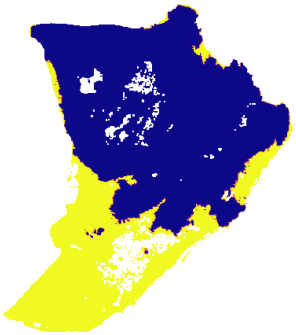
`1-simple_mesh_contours.py`

Reads a QGIS mesh layer, detects the scalar data range (clamping if error: NoData sentinels), then runs native:meshcontours to produce two temporary layers.

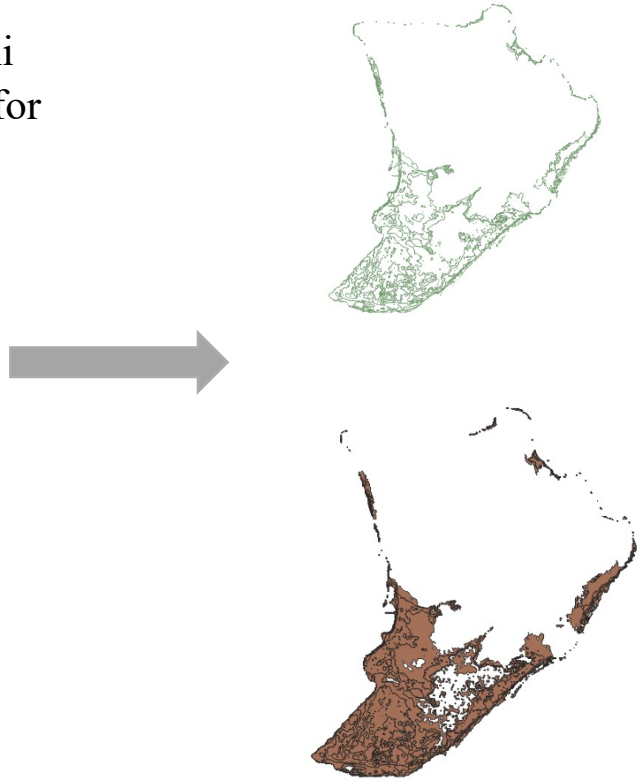
Output:

→ `BASE_NAME_contour_lines`
→ `BASE_NAME_contour_polygons`

Modeling tsunami inundation results for Angaur, Palau



(Sannikova and Moore, 2024)



From Mesh to Clean Boundary

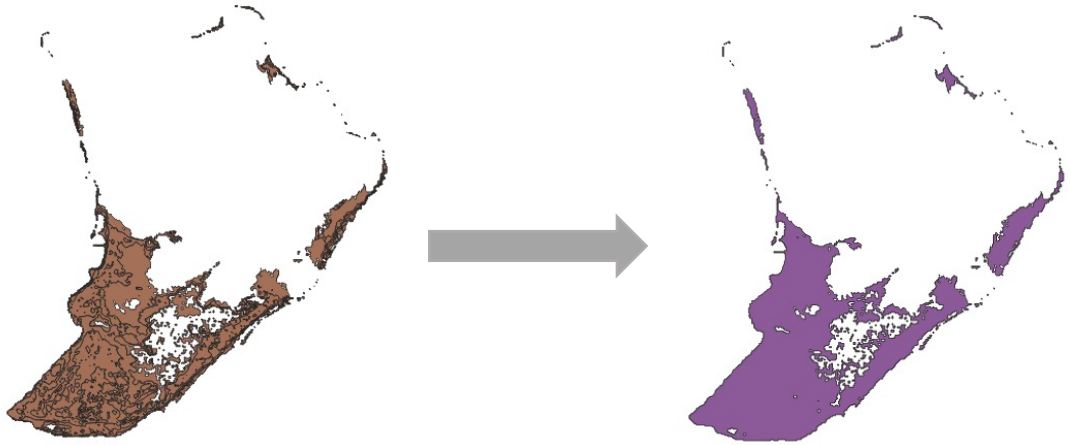
2 Dissolve Contour Polygons

`2-dissolve_function.py`

Takes BASE_NAME_contour_polygons and runs a full dissolve, merging all individual contour bands into one multipart polygon.

Output:

→ `BASE_NAME_dissolved`



From Mesh to Clean Boundary

3 Extract Outer Boundary

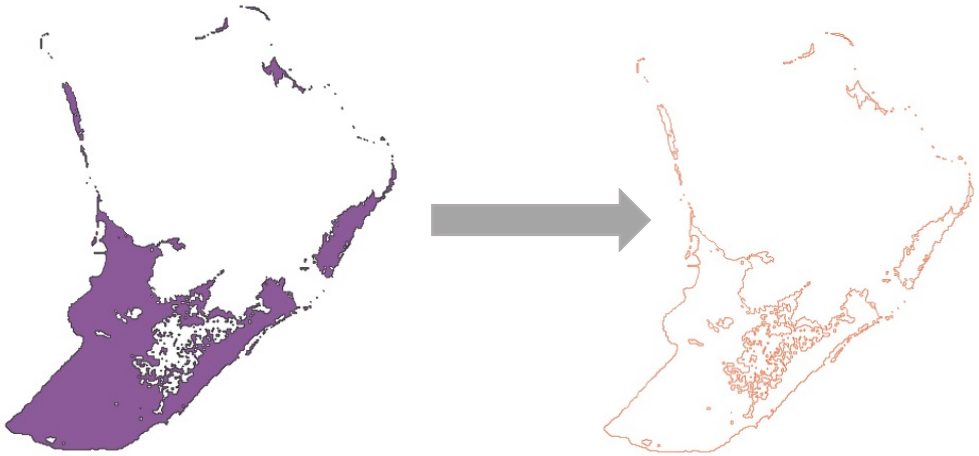
`3-extract_outer_boundary.py`

Runs native:boundary on the dissolved layer to strip away all interior rings, leaving only the outer ring. Saves the results.

Clean boundary for the inundation modeling results.

Output:

→ `BASE_NAME_outer_boundary (.gpkg/.shp/.geojson)`



From Mesh to Clean Boundary– Check In

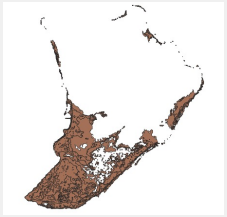
1 Mesh Contour Export

`1-simple_mesh_contours.py`

Reads a QGIS mesh layer, detects the scalar data range (clamping if error: NoData sentinels), then runs native:meshcontours to produce two temporary layers.

Output:

→ `BASE_NAME_contour_lines`
→ `BASE_NAME_contour_polygons`



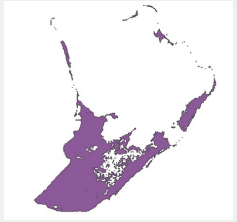
2 Dissolve Contour Polygons

`2-dissolve_function.py`

Takes BASE_NAME_contour_polygons and runs a full dissolve (no field), merging all individual contour bands into one multipart polygon.

Output:

→ `BASE_NAME_dissolved`



3 Extract Outer Boundary

`3-extract_outer_boundary.py`

Runs native:boundary on the dissolved layer to strip away all interior rings, leaving only the outer ring. Saves the results.

Output:

→ `BASE_NAME_outer_boundary (.gpkg/.shp/.geojson)`

Solid Polygon of Results & Final Dissolve

4 Create Solid Filled Polygon

`4-solid_polygon_fixed.py`

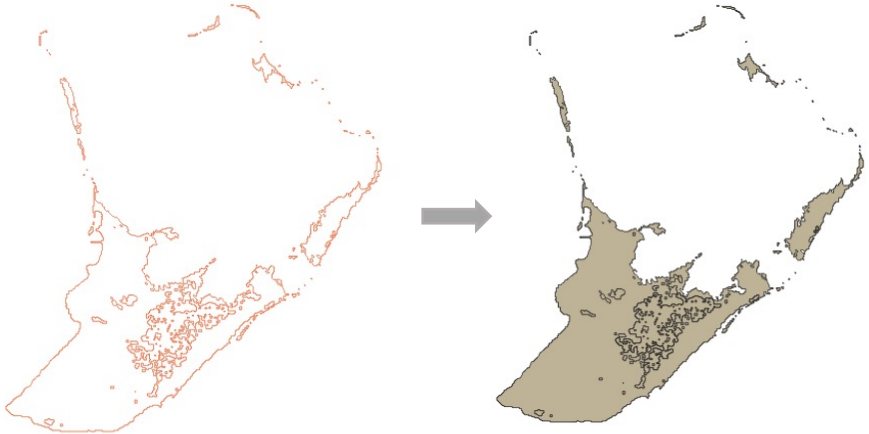
Accepts lines or polygons -- Detects geometry type

Removes interior holes -- Runs native:deleteholes (MIN_AREA = 0) to wipe any remaining interior rings from the polygon.

Saves -- Writes BASE_NAME_filled_polygon to your OUTPUT_DIR in the chosen format.

Output:

→ `BASE_NAME_filled_polygon`



Solid Polygon of Results & Final Dissolve

5 Final Dissolve Polygon

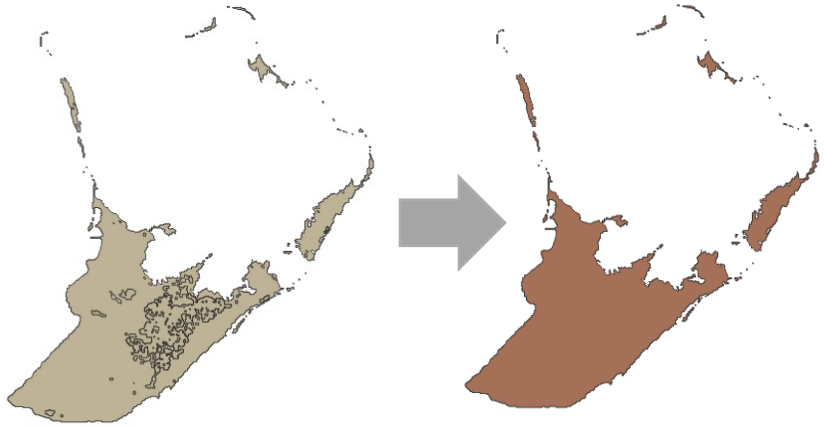
`5-dissolve_polygon.py`

Why a second dissolve?

After polygonizing and deleting holes, the layer may still contain multiple adjacent features or artefact boundaries from the line-to-polygon conversion. A final dissolve (native:dissolve) collapses these into a single clean feature.

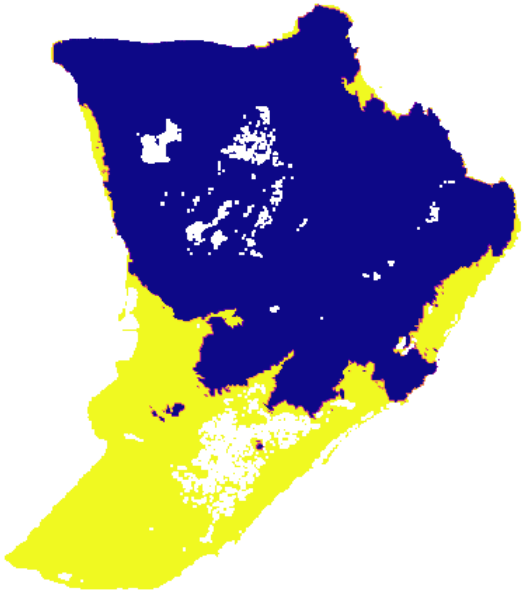
Output saved:

```
BASE_NAME_final_polygon
(.gpkg / .shp / .geojson)
```



From Mesh to Polygon

Modeling tsunami inundation results for Angaur, Palau



(with a lot of behind the scene steps in-between)

(Sannikova and Moore, 2024)

Cleanup & Basemaps– Check In

6 Clean Up Layer Panel

`6-cleanup_layer_panel.py`

Keeps in QGIS:

- Original mesh layer (one copy)
- BASE_NAME_contour_lines
- BASE_NAME_final_polygon
- All basemap / QMS tile layers (e.g., Bing)

Also saves layers and writes the .qgz project file.

Removes:

- _contour_polygons, _dissolved, _outer_boundary, _filled_polygon
- Duplicate mesh layers

7 Add Basemap Layers

`7-add_basemaps.py`

Adds Google Roads and Google Satellite as XYZ tile layers using the 'wms' provider, then moves them to the bottom of the layer panel.

XYZ tiles require an internet connection in QGIS.

Configuration:

```
ADD_GOOGLE_ROADS     = True
ADD_GOOGLE_SATELLITE = True
MOVE_TO_BOTTOM       = True
```

Running Scripts Individually

Each script can be run on its own in the QGIS Python Console.

How to run a single script:

1. Open the script in the QGIS Python Console Editor (Plugins > Python Console > Show Editor).
2. Edit the CONFIGURATION block near the top.
3. Make sure the input layer is already loaded in your QGIS project.
4. Click Run Script. The script will print progress to the console.

Example of editing the Configuration script block in script 2:

```
BASE_NAME     = "Orukei_MaxFlow_Max_of_max"   # must match what script 1 produced
OUTPUT_DIR    = r"C:\Projects\...\Palau\output"
OUTPUT_FORMAT = "GPKG"
```

Running the Full Workflow

Use this script to process one file (SINGLE_FILE_MODE = True) or an entire folder of .nc files.

1. Open QGIS and load your project

File > Open Project or File > New. The QGIS project does not need to be pre-populated — the script will load any mesh layers.

2. Open the Python Console

Plugins > Python Console, then click Show Editor.

3. Open main_batch.py

Click the folder icon in the editor and browse to your scripts folder.

4. Verify config.py is in the same folder

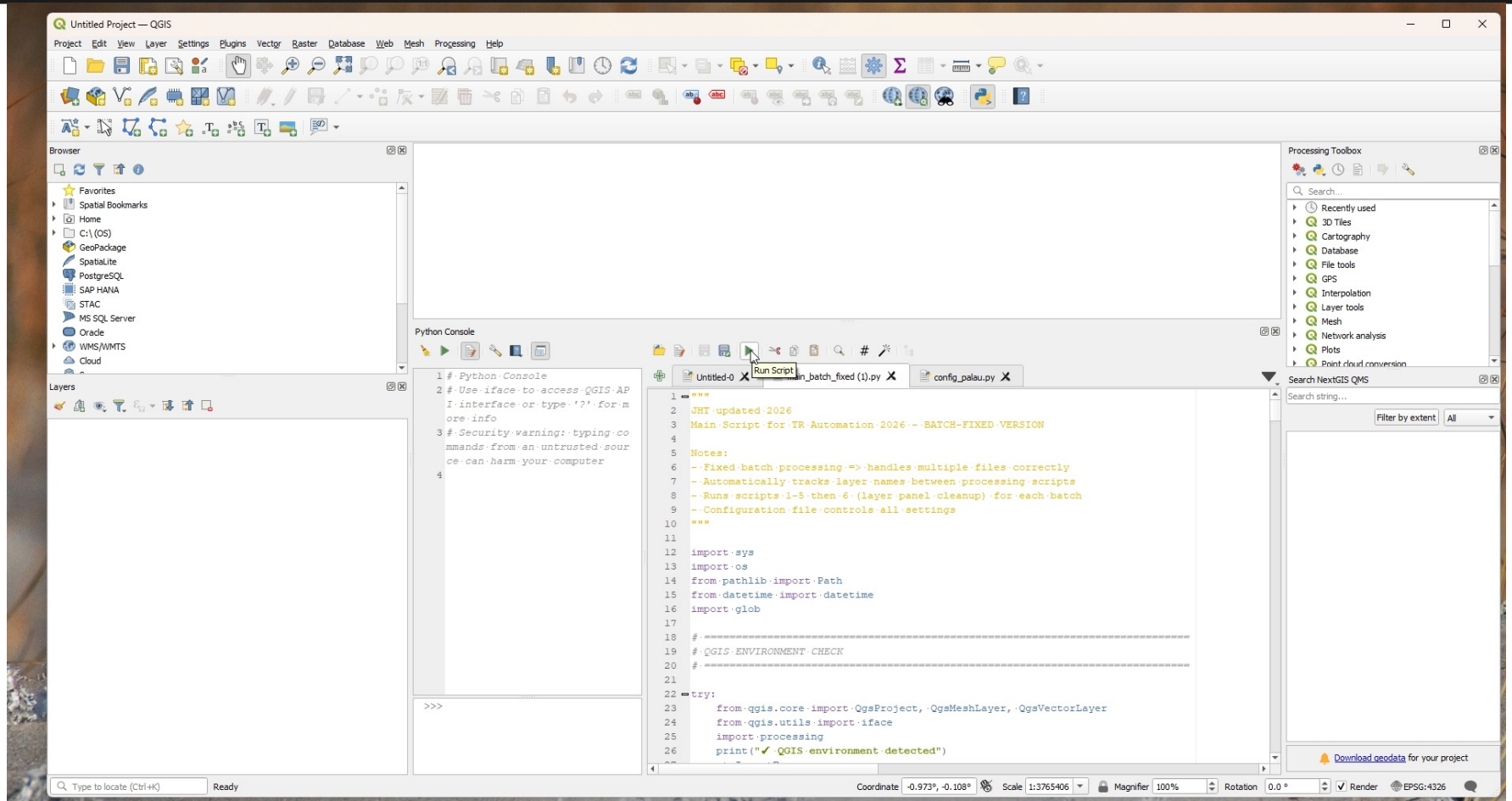
The main script imports it by name. Both files must be in SCRIPTS_DIR.

5. Click Run Script

Watch the console output. Each step prints ✓ on success or ✗ on failure with a traceback.

6. Inspect results and outputs

After completion: contour lines + final polygon remain in the layer panel, basemaps added at bottom, project saved automatically.


Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Browser
Favorites
Spatial Bookmarks
Home
C:\ (OS)
GeoPackage
SpatiaLite
PostgreSQL
SAP HANA
STAC
MS SQL Server
Oracle
WMS/WMTS
Cloud
Layers
Python Console
Python Console
Use iface to access QGIS API interface or type '?' for more info
Security warning: typing commands from an untrusted source can harm your computer
>>>
Run Script
Untitled-0
config_palau.py
"""
JHT updated 2026
Main Script for TR Automation 2026 - BATCH-FIXED VERSION
Notes:
- Fixed batch processing => handles multiple files correctly
- Automatically tracks layer names between processing scripts
- Runs scripts 1-5 then 6 (layer panel cleanup) for each batch
- Configuration file controls all settings
"""
import sys
import os
from pathlib import Path
from datetime import datetime
import glob
# ==
# QGIS ENVIRONMENT CHECK
# ==
try:
from qgis.core import QgsProject, QgsMeshLayer, QgsVectorLayer
from qgis.utils import iface
import processing
print("✓ QGIS environment detected")
Processing Toolbox
Search...
Recently used
3D Tiles
Cartography
Database
File tools
GPS
Interpolation
Layer tools
Mesh
Network analysis
Plots
Point cloud conversion
Search NextGIS QMS
Search string...
Filter by extent
All
Download geodata for your project
Type to locate (Ctrl+K)
Ready
Coordinate -0.973°, -0.108°
Scale 1:3765406
Magnifier 100%
Rotation 0.0 °
Render
EPSG:4326

Implementing Workflow

Requirements

QGIS LTR 3.40 (v3.33 or higher)

Scripts use native:meshcontours and other built-in processing algorithms to quickly digitize modeling results.

Python Console

Access via Plugins > Python Console > Show Editor.

Data – Modeling Results

Tsunami inundation modeling results.

Recommended Folder Structure

```
YourProject/
   ├── scripts/
   │      ├── 1-simple_mesh_contours.py
   │      ├── 2-dissolve_function.py
   │      ├── 3-extract_outer_boundary.py
   │      ├── 4-solid_polygon_fixed.py
   │      ├── 5-dissolve_polygon.py
   │      ├── 6-cleanup_layer_panel.py
   │      └── 7-add_basemaps.py
   ├── data/
   │      ├── input/   ← Put Modeling results here
   │      └── output/  ← GPKG/SHP files saved here
   ├── [TR_map].qgz
   ├── config_[country].py
   └── main_[country].py
```

Editing the Configuration Script for Your Project

This is the single file you edit before each project.
The Main script runs all other scripts in the script directory.

File / Mode

```
SINGLE_FILE_MODE = True   # True = one file,
False = all files in folder
SINGLE_MESH_PATH = r"C:\...\[file name].nc"
SINGLE_MESH_NAME = "[file name]"
```

Output

```
OUTPUT_DIR    = r"C:\...\output"
OUTPUT_FORMAT = "GPKG"     # or "SHP" or
"GEOJSON"
```

Processing

```
CONTOUR_INTERVAL    = 1.0   # meteres between
contours
DATASET_GROUP_INDEX = 0     # usually 0; check
mesh properties
```

Scripts list

```
SCRIPTS_TO_RUN = [
    "1-simple_mesh_contours.py",
    "2-dissolve_function.py", ... ]  # up to 5
scripts
```