Kotyle

Release 0.1

Stefano Costa

May 18, 2020

CONTENTS

Kotyle (from the ancient greek κοτύλη, “measure of capacity”, “drinking cup”) is a software program for calculating the capacity of a ceramic vessel. The main use of Kotyle is for artifacts studied by archaeologists.

Kotyle is written in the Python programming language and is available under the Apache Software License 2.0.

Kotyle can be downloaded from the Git repository at Codeberg.

For a quick start, look at the *GIMP plugin* and *Μετρω, the Kotyle measurement tool* pages.

CHAPTER

ONE

CONTENTS

1.1 Introduction, or why would you want to know capacity

Measuring capacity of a ceramic vessel is not part of the ordinary archaeological process that starts with the recovering of artifacts during excavation and ends with publication as a drawing of the profile. More recently, photographic pictures of sherds and vessels are included in archaeological publications.

1.1.1 Transport vessels

Capacity is of paramount importance for transport vessels like amphorae. Should you fail to recognize this importance, you may end up comparing numbers that have no meaning. If you have 3 *Africana grande* and 3 *Keay 26* (*spatheion*), you're not dealing with the same amount of 2 different types of vessels, but with a dramatically different quantity of oil or wine.

1.1.2 Eating vessels

Even for the study of tablewares, capacity can be a key factor to distinguish between individual and collective vessels.

1.2 Describing the profile of a vessel in a digital format

The most difficult part of the story is finding an acceptable digital format for describing the profile of a vessel.

For **drawing** purposes, it is common to use vector graphics software or CAD systems. It is clear that it is highly desirable to represent the profile as a vector geometry, that is, a sequence of points and lines in a cartesian space.

However, all those programs are made for drawing, and not for describing geometric entities in a purely **abstract** fashion. We don't need line widths, colors and other such features. Furthermore, we don't want to work in a coordinate system where the origin (the `0,0` point) is in the top left corner, like for example SVG does. We don't need anything that has to do with graphics, just a compact way of representing a (tiny) set of ordered geometric coordinates.

1.2.1 CSV: a basic approach

Until now, the most convenient way for doing so may be a good old CSV file:

```
11.5989847716,11.4316005076
12.4873096447,11.3938563452
13.4263959391,11.387836802
14.6700507614,11.4760149746
16.1675126904,11.6587162437
16.7512690355,11.7511243655
17.9441624365,12.0639779188
18.7563451777,12.6677218274
19.3147208122,13.4012926396
19.6446700508,14.008127665
19.7715736041,14.1675642132
19.7969543147,14.2635515228
19.7969543147,14.3917515228
19.923857868,14.8075880711
```

While it might seem awkward at first sight, it contains all the information we need. The only problem with this approach is that there is no place for metadata like the name and description of the vessel.

1.2.2 GeoJSON: going further

A slighly more advanced encoding of the same data can be expressed as GeoJSON:

```
{ "type": "Feature",
  "geometry": {
    "type": "LineString",
      "coordinates": [
            [11.5989847716,11.4316005076],
            [12.4873096447,11.3938563452],
            [13.4263959391,11.387836802],
            [14.6700507614,11.4760149746],
            [16.1675126904,11.6587162437],
            [16.7512690355,11.7511243655],
            [17.9441624365,12.0639779188],
            [18.7563451777,12.6677218274],
            [19.3147208122,13.4012926396],
            [19.6446700508,14.008127665],
            [19.7715736041,14.1675642132],
            [19.7969543147,14.2635515228],
            [19.7969543147,14.3917515228],
            [19.923857868,14.8075880711]
      ]
    },
  "properties": {
    "name": "Hayes 61 B",
    "description": "A quite small version of 61 B.",
    "author": "Stefano Costa"
  }
}
```

This format is cheap, and it can be read directly by dedicated geometry libraries. It's not easy to type, but it's definitely easy to read. Plus, the idea is that you *don't* have to type it, because a dedicated program can do the work for you.

If you use the GIMP, Kotyle has *GIMP plugin* that export vector paths to GeoJSON and facilitate creating digital profiles.

GeoJSON profiles can be analysed with *Μετρω, the Kotyle measurement tool*.

1.3 GIMP plugin

If you use an image editor like GIMP or Glimpse you can use Kotyle directly with a plugin to measure capacity and save the corresponding *vector path* of the vessel profile as GeoJSON data.

It is very easy to install and use the plugin. The README with installation instructions is here together with the plugin.

Both GIMP and Glimpse are freely distributed programs for image retouching and manipulation, available for all operating systems.

1.3.1 Installation

Installing this plug-in is very easy and requires copying one file in the right directory.

Open the *Edit ▸ Preferences* dialog from the main window.

The dialog has a long list of various preferences. The last item in the list should be "Folders" and among the sub-items there are "Plug-ins". When you enter the preferences for plug-ins folders, you should find some paths listed and you can open the path in your user directory directly from the same dialog.

Copy the `vector-to-geojson.py` file in that directory. On Linux or MacOS, make sure the file is executable.

Restart The GIMP, and you will find the plug-in the *Filters ▸ Kotyle* menu.

1.3.2 Before you start: check image resolution

It is very important that you check the image resolution is the correct one, e.g. if you scanned a drawing at 300 dpi the image file **must** be set at the same resolution, otherwise all measurements will be wrong. This is fairly easy if you're working with your own drawings, but it may require some work in case the drawings are from older digital archives.

You can set the image resolution from the *Image ▸ Print size…* menu item.

You will also need to make sure that the drawing is correcly rotated with respect to the geometric lines, that is the rotation axis must be a vertical straight line. Kotyle had a plugin for this but it became obsolete with GIMP 2.10 that includes the same functionality by default with the "Measure" tool and the associated "Straighten" function.

You can find the detailed manual on the GIMP website at <https://docs.gimp.org/2.10/en/gimp-tool-measure.html>_.

1.3.3 Vector paths in GIMP

Both plugins work with *Paths* in GIMP. Paths are separate from the raster layers and can be drawn on top of the existing layers.

There is a specific panel for Paths where they can be removed and renamed. Renaming is important here because we will need to be able to pick one or more paths and use them with Kotyle.

You can find the most recent documentation about paths in the GIMP Manual.

1.3.4 Drawing a vector path for a vessel profile

The *Path* tool allows you to create vector paths. You should either start from the vessel bottom (foot, base) or the top (rim) taking care to draw along the **inner profile** (assuming you want to calculate the volume of the vessel content).

Make sure you draw only one path for the profile. When you're done, go the *Paths* panel in the main toolbox (close to the *Layers* panel) and rename your path to something useful, like *Bowl Type 23*.

1.3.5 Measuring vessel capacity

Go to the *Plugins* ▸ *Kotyle* menu (it should be close to the bottom of the menu) and click the *Measure vessel capacity* item.

A dialog will open, where you need to choose a few parameters:

- the path (directory) where you want to save the file
- the name of the file (default is `profile.json` but you want to choose a more sensible name if you have more than one profile) where to save the GeoJSON data of the vessel profile
- the vector path you want to export – you will see that the drop-down list shows you the names of paths – you can choose only one path
- title, description and author of the drawing, optional but useful

The computed capacity expressed in liters will be shown in a small window.

1.3.6 What to do with the GeoJSON?

In short, you are going to have one `.json` file for each profile you want to measure. These files are nothing special, you can open them with any text editor and look at their contents.

More details about the GeoJSON format are found in *Describing the profile of a vessel in a digital format*.

The *Measure vessel capacity* plugin will save your vessel profile in a file (e.g. named `profile.json`).

Kotyle has also a separate tool to calculate vessel volume from that file, called *Μετρω, the Kotyle measurement tool*. This tool is useful if you need to process many files at once.

1.4 Μετρω, the Kotyle measurement tool

This is a simple command-line Python script that reads data from a GeoJSON file (see *Describing the profile of a vessel in a digital format*) and outputs the vessel volume.

It is found in the source tree of Kotyle.

1.4.1 Usage

Example usage:

```
$ python metro.py ~/profile.json
The capacity of "GQB 266.231" is 6.04 ℓ
```

In this example, the input file `profile.json` has the following content:

```
{
    "geometry": {
        "type": "LineString",
        "coordinates": [
            [
                0.13614400000000002,
                0.20726400000000003
            ],
            [
                0.13893799999999998,
                0.19431000000000004
            ],
            [
                0.138176,
                0.18669000000000002
            ],
            [
                0.133096,
                0.1778
            ],
            [
                0.120396,
                0.165354
            ],
            [
                0.107696,
                0.15417800000000004
            ],
            [
                0.10033,
                0.13741400000000004
            ],
            [
                0.096012,
                0.11531600000000002
            ],
            [
                0.09982200000000001,
                0.09931400000000001
            ],
            [
                0.10744200000000001,
                0.07594600000000004
            ],
            [
                0.11683999999999999,
                0.05994400000000005
            ],
            [
                0.124714,
                0.043180000000000024
            ],
            [
                0.135128,
                0.02895600000000001
            ],
            [
```


(continues on next page)

(continued from previous page)


```
                0.1397,
                0.02235200000000001
            ],
            [
                0.158496,
                0.014224000000000042
            ],
            [
                0.17653,
                0.005842000000000014
            ],
            [
                0.19583399999999998,
                0.0012700000000000489
            ],
            [
                0.21488400000000002,
                0.0
            ]
        ]
    },
    "type": "Feature",
    "properties": {
        "author": "Stefano Costa",
        "description": "Cooking pot",
        "title": "GQB 266.231"
    }
}
```


1.5 Calculating the volume of a vessel

As we have seen, the simplest method for calculating the volume of a solid of revolution is by dividing the rotating shape into smaller pieces.

This method has however one major flaw that limits its adoption for archaeological purposes, namely it cannot be used to calculate the capacity of a vessel that has a convex interior surface.

For this reason, Kotyle uses a more advanced technique, based on Pappus's centroid theorem.

1.5.1 Pappus's centroid theorem

In mathematics, Pappus's centroid theorem is either of two related theorems dealing with the surface areas and volumes of surfaces and solids of revolution.

The second theorem states that the volume V of a solid of revolution generated by rotating a plane figure F about an external axis is equal to the product of the area A of F and the distance d traveled by its geometric centroid.

$$V = Ad$$

For more details, see Wikipedia (which is also the source for the two paragraphs above).

1.5.2 Implementing Pappus's centroid theorem

According to the above definition, the required data to compute the volume are:

- the area of the plane figure
- the distance of its centroid from the rotation axis

Both geometric operations are not trivial for non-standard polygons, so we use the Shapely Python library to perform them. Shapely can be easily integrated with Matplotlib to plot both the profile and the geometric centroid.

1.5.3 Making sense of different standards

To add more fun, profile drawings can be both right-handed or left-handed. We could force people to mirror their images before feeding them into Kotyle, but why would you do that ? Computers should be working for us, not the opposite !

It's quite easy to manage this disparity, even taking into account that people might start drawing from the rim (top) or from the bottom of the vessel.

	x0 > xN	x0 < xN
y0 > yN	↙	↘
y0 < yN	↖	↗

`x0` is the ascissa of the first point in the sequence, and `xN` the ascissa of the last point. Same for `y0` and `yN`. Arrows indicate in which direction the profile was drawn. "Same sign" combinations give the same result.

The standardized view is with the `0,0` point at the bottom of the profile, where it crosses the rotation axis. But it's easy to `reverse()` those profiles that follow the opposite standard, mirroring them.

1.6 Calculating the surface of a vessel

While certainly less used, it's quite easy to get the surface of a solid of revolution using the Pappus's first centroid theorem.

1.6.1 Pappus's centroid theorem

$$A = sd$$

1.6.2 Why calculating surface ?

One might ask as well: why not ? A large number of different quantification methods have been developed since the 1970s, but none of them have taken into account the surface of the vessel.

It can be difficult to measure the planar surface of a potsherd, but it doesn't mean that surface could not be adopted together with *EVE*, or *MNV*.

1.6.3 How can I do that in practice ?

You could use a large number of small objects, to be temporarily sticked on the surface covering it homogeneously. Then, you just remove them and count / weight. This would work for sherds of small size.

1.7 Dealing with weight and density of ceramic vessels and sherds

Volume and capacity have a relationship with weight. This is especially true for two separate ways of using such measurements:

- calculating the ratio between the weight of the container (either amphora or other container) and the weight of the liquid stored inside;
- using the weight of potsherds as a quantification method.

Suppose you have a complete profile of a vessel, in the form of a drawing. As often happens, only part of the vessel is preserved. Nevertheless, you would like to calculate what is the proportion between the total vessel and the sherds that were found in a specific archaeological context.

This is possible, and not difficult. The envisaged procedure should be as follows:

1. calculate the total volume of the vessel from the drawing – note that this means the volume of the ceramic body and not the capacity of the vessel
2. from the volume, obtain the total weight through density
3. weight your sherds and see the proportion

Knowing the density of a ceramic body might not be immediate, and it's not certainly a good idea to stick with a standard value good for contemporary ceramics (that is around 2 kg/dm^3, by the way).

1.7.1 Measuring the density of a ceramic body

The density and porosity of a ceramic body can be obtained through Archimedes' principle. A formal procedure is described in [ASTM-C373] (not publicly available, unfortunately), and it needs only water and a weight scale. Here follows a basic summary of the procedure.

There are three measurements that need to be taken: - dry weight $W1$ - weight of the ceramic body in water $W2$ - weight of the water-saturated ceramic body $W3$

From these three values, the following properties can be obtained with simple formulas:

- external volume $V = W3–W2$
- bulk density $bD = W1/V$
- apparent density $aD = W1/(W1 - W2)$
- apparent porosity $P = (W3 - W1)/V$
- water absorption $A = (W3 - W1)/W1$

Note: This section was based on content published by Antonio Licciulli on his faculty website. You can find detailed content, mostly in Italian, at this page

A basic example

What follows is a basic procedure that was tested on a limited number of samples. You may want to use a spredsheet software to help with the recording and the quick calculation of the properties derived from the weight measurements.

You will need a container for water that is both wide enough to put ceramic sherds inside it and tall enough to have them stay completely below the water level. You will also need to keep the ceramic body suspended in water, almost certainly in vertical position.

Step by step:

- weight the dry sherd ($W1$);
- put the sherd inside the water, and let it stand on the bottom of the container until you see no air coming out of it: the ceramic body is now saturated with water;
- when the ceramic sherd is water-saturated, move the container with water (and the sherd on the bottom) on the weight scale and measure the weight $p1$
- use a pair of pliers to lift the ceramic sherd from the bottom, and keep it suspended in water and completely below the water level – taking care that the pliers will stay as much as possible out of the water – and measure weight $p2$;
- take the ceramic sherd out of the water, let it quickly drip inside the container and weight the container with only water (the sherd should now be *out* of the container, on the desk), recording $p3$
- optionally weight the ceramic sherd you just took out from water, recording $p4$.

At this point, the *dry weight* is:

$$W1$$

while the weight of the ceramic body in water is:

$$W2 = p2 - p3$$

and the weight of the water-saturated ceramic body is:

$$W3 = p1 - p3 = p4$$

$p1$ ***must*** **always be the largest measurement, and** $p3$ **the** smallest one:

$$p1 > p2 > p3$$

All the required measurements are now available to calculate the density, porosity and water absorption of the ceramic body.

1.8 Other programs that calculate vessel capacity

Inspiration for writing Kotyle derived from other existing tools, that I didn't find satisfactory for my purposes.

1.8.1 VESCAP

VESCAP (VESsel CAPacity) is a standardized procedure based on AutoCAD used together with the CAD Overlay plugin to digitize a raster drawing.

The main advantage of the VESCAP routine lies in avoiding the separation between the drawing of the potsherd and the geometric definition of the vessel profile. Both can be stored in the same CAD file, and the solid of revolution can be easily created. AutoCAD gives directly the volume of the solid, so the results are guaranteed correct.

VESCAP was published by James McCaw in 2007 [Peña07].

1.8.2 ARCANE Pottery Utility

Far from being a magic tool, the Pottery Utility developed by the ARCANE Project is a dedicated stand-alone program devoted to measurements of pottery drawings and profiles of vessels.

However, the method it uses to calculate volume has a major flaw, because it assumes a *monotonic* profile, that is, a profile where the lowest point lies on the rotation axis. There are some cases where such a condition doesn't hold.

Pottery Utility is based on Shockwave Flash.

1.8.3 Amphoralex

There is a volume calculation program available from Amphoralex, the website of the *Centre Alexandrin d'Étude des Amphores*.

It appears to be based on FileMaker 5.

1.8.4 Calcul de capacité d'un récipient à partir de son profil

This is a web service developed by CReA (Centre de Recherches en Archéologie et Patrimoine).

Registration is needed.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

Logo: Kotyle with satyr, from W. Lamb, *Seven vases from the Hope collection*, in *Journal of Hellenic Studies* 38, 1918.

BIBLIOGRAPHY

[ASTM-C373] ASTM Standard C373 - 88(2006) “Standard Test Method for Water Absorption, Bulk Density, Apparent Porosity and Apparent Specific Gravity of Fired Whiteware Products”, ASTM International, West Conshohocken, PA, 2006. DOI: 10.1520/C0373-88R06, http://www.astm.org/

[Peña07] J. Theodore Peña, *The quantitative analysis of Roman pottery: general problems, the methods employed at the Palatine East, and the supply of African Sigillata to Rome*, in E. Papi (ed.) *Supplying Rome and the Empire*, pp. 153-172