Matplotlib in the geospatial analyzes

G. Nagy*

*Óbuda University, Alba Regia Technical Faculty, Institute of Geoinformatics
nagy.gabor@amk.uni-obuda.hu
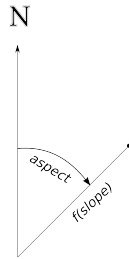
Figure 1. The polar coordinate system of the diagram.

Abstract—**The diagrams are useful tools to illustrate the result of an analysis. In many cases the classical tools (the diagram creator modules in the spreadsheet softwares) can not make the imagined diagrams. For example diagram which was suggested in my previous article to illustrate the distribution of slope and aspect of an area. The Matplotlib is an Open-source plotting library, which can make varies diagrams with varied settings. This tool can be used well in spatial analyzes. Data processing and charting can be done in an Python program.**

Index Terms—**Matplotlib, Python, GIS**

I. INTRODUCTION

The figures are very important parts of scientific articles and education materials. A lot of application can create various diagrams. Probably, the most famous and easy choice is the chart creator function of the spreadsheet softwares, but these tools provides only the basic chart types, and the customization is also limited.

This paper present an open source tool for this task with practical examples in the geospatial analyzes.

II. A VISUALIZATION TASK

The distribution of slope and aspect can be shown an polar diagram, where the angle is the aspect and the distance depends from the slope (in the following examples the distance is the square root of the slope). (See in Figure 1.) The distribution of the different areas are shown an scatter chart in this polar coordinate system. ([1].)

The distribution of the points in these scatter diagrams represents the distribution of the slope and aspect in an area. The [1] uses a program, which creates SVG files from the datasets.
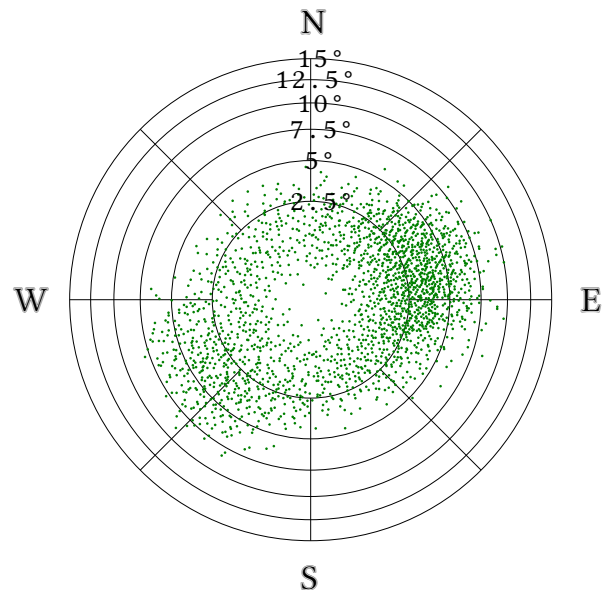

Figure 2. The scatter diagram created by an SVG based program.

III. THE MATPLOTLIB

The Python language [2] is very popular in the scientific projects [3], [4], many Python tools aid research works. The Matplotlib [5] is an open source Python module for creating diagrams.

The diagrams presented in the Section II can be created by the Matplotlib tools easier than the SVG-based solution. (See in Figure 3. and Figure 4.)

The Matplotlib based program is more clearly, because it need not create the XML text description of a lot of required parts of the SVG files.

IV. SURFACE VISUALIZATION IN MATPLOTLIB

Matplotlib can show surfaces by different methods. The surface data is stored in a 2 dimensional NumPy [6] array. This array follows the Matlab convention: the X coordinate depends from the column number and the Y coordinate depends from the row number. The Gdal [7] module can read raster data from varied file formats to a NumPy array.

The direction of the rows are reverse in the Gdal read arrays than the upper convention. The rows of the array can be reversed by the `flipud` function of the NumPy.

A. Colored maps

The Matplotlib can create colored map representation of the surfaces. The simplest way for this is
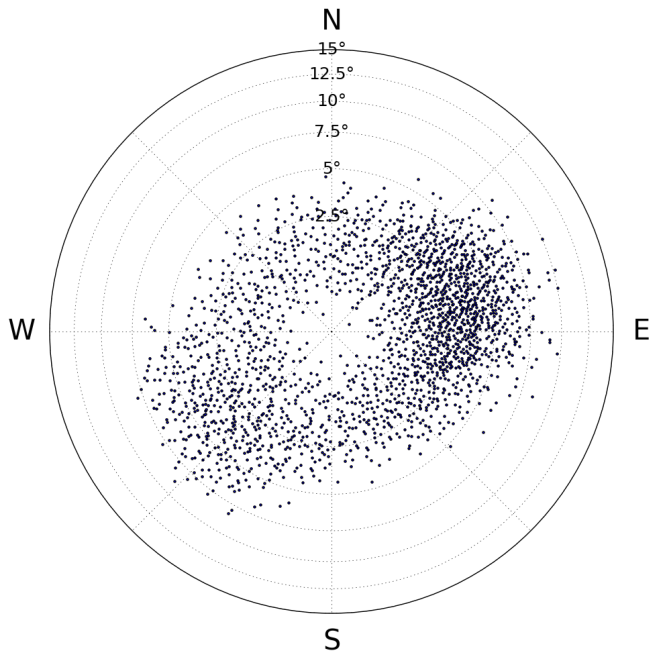
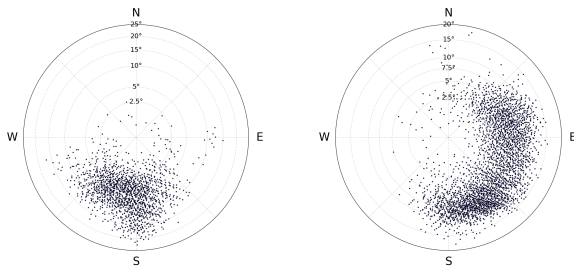Figure 3. The scatter diagram created by an Matplotlib based program.

Figure 5. The surface of Velence Hills with the standard colormap of the Matplotlib

Figure 4. Other scatter diagrams created by the Matplotlib based program.

Figure 6. The surface of Velence Hills with the `gist_earth` colormap of the Matplotlib

the `matplotlib.pyplot.pcolor()` function. (The `matplotlib.pyplot` usually is used by the `import matplotlib.pyplot as plt` command in the Python programs, and in this case this function can be used by the `plt.pcolor()` form. In the following, I use this form in the sample codes.) The `plt.pcolor(z)` creates a colored map from the `z` two-dimensional NumPy array by the default colormap, where `z` is an two-dimensional array with the elevations.

Other colormap can be used by an optional argument of the `plt.pcolor()` function. For example: `plt.pcolor(z, cmap=plt.cm.gist_earth)`. (See in the Figure 6. and the Figure 7.)

The Matplotlib can create shaded surface map. (See in the Figure 8. and the Figure 9.) An `LightSource` object configures the azimuth and the altitude of the light source of the shading.

B. Contour maps

The `plt.contour()` function creates contour map from the surface data. For example: `plt.contour(z)`. (See in the Figure 10.)
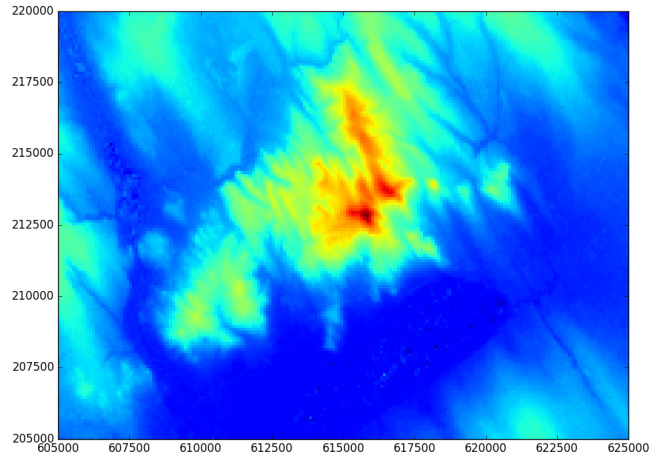

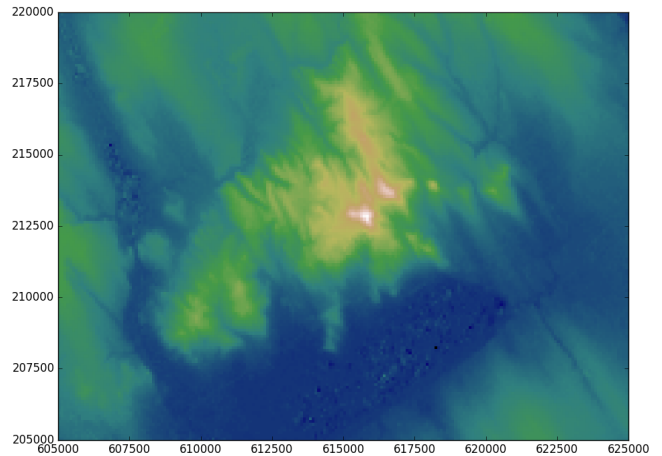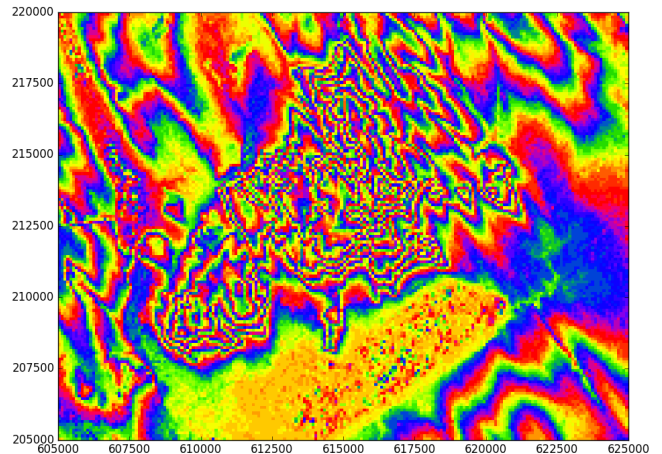Figure 7. The surface of Velence Hills with the `prism` colormap of the Matplotlib

Figure 8. The surface of Velence Hills with shaded map by Matplotlib

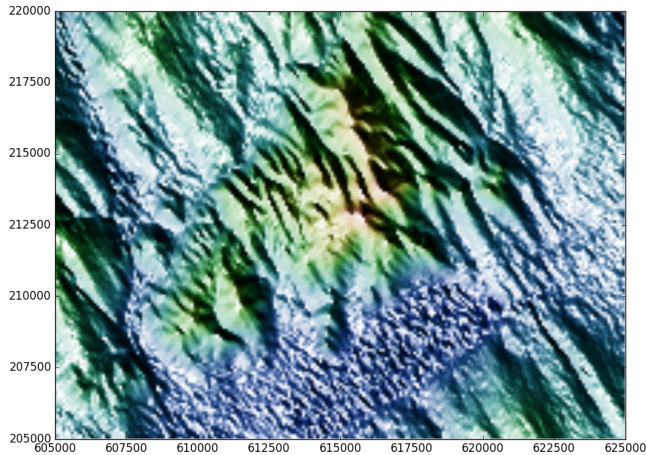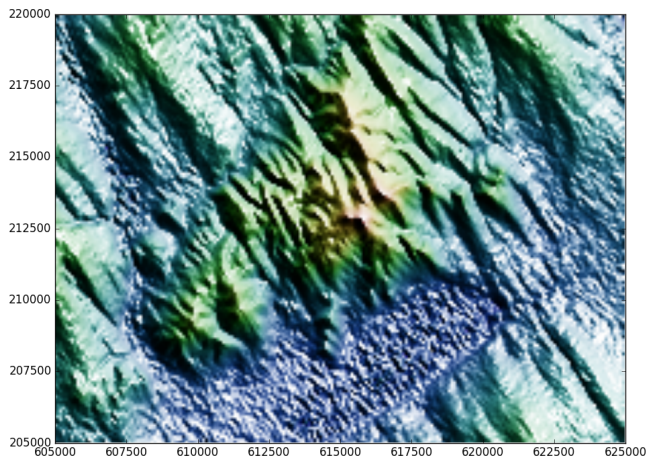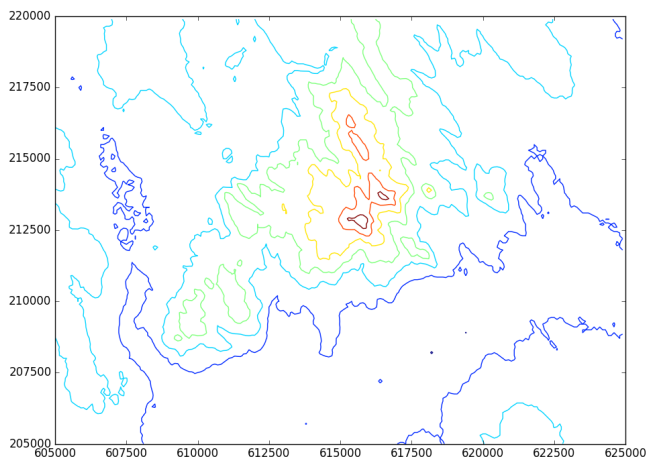Figure 11. Violin plots of the distribution of LiDAR-based surcaces compared to geodetic survey.

Figure 9. Shaded map with opposite light direction than the Figure 8.

Figure 10. The surface of Velence Hills with contour lines

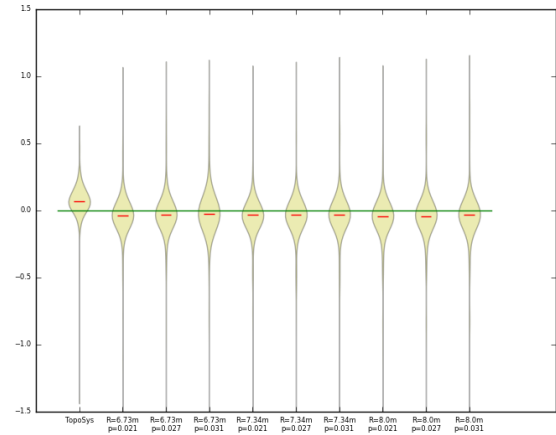The Matplotlib contour charts are not the typical contour maps. The program uses different colors for the different elevations. (The colormap can be changed.)

V. OTHER MATPLOTLIB APPLICATIONS

Matplotlib provides a lot of chart types. Any of them may be useful in a research.

For example LiDAR-based surfaces are comparable with geodetic surveys, the difference between the elevations from the surfaces and the geodetic survey can be calculated each point of the survey. The distribution of these values characterizes the method which created the surface from the LiDAR point cloud.

The violin plots [8] are good tools for representing distributions. The Matplotlib can create violin plot from a dataset.

A Matplotlib plot can contains more violins from different datasets. For example the first element of the Figure 11. and Figure 12. are the distribution of the errors (differences between the surface and the geodetic survey) in a surface created by an commercial LiDAR processing software (TopoSys), and the other violins shown same results by a new suggested LiDAR processing methods with different parameters (Denoted R and q).

Matplotlib based Python programs can create animated violin plots, one of the parameters can be changed in the time. This possibility are very useful, because an animation can show a lot of combinations of the R and q values.

VI. CONCLUSION

Matplotlib is an useful tool for creating scientific and educational charts. The diagrams can be created by Python programs. These programs also can be calculated the data of the generated figures.

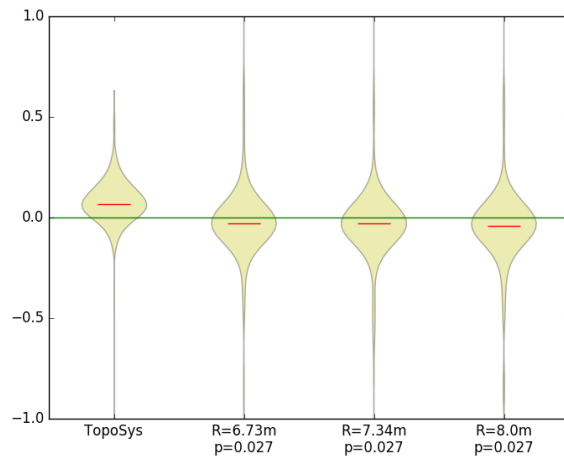These tools may be useful in any project.

Figure 12. Another violin plots of the distribution of LiDAR-based surcaces compared to geodetic survey.

REFERENCES

[1] G. Nagy, *A diagram to illustrate the distribution of slope and aspect of an area*. Óbudai Egyetem, 2016, pp. 24–27.

[2] G. Van Rossum *et al.*, "Python Programming Language." in *USENIX Annual Technical Conference*, vol. 41, 2007.

[3] K. J. Millman and M. Aivazis, "Python for scientists and engineers," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 9–12, 2011.

[4] T. E. Oliphant, "Python for scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, 2007.

[5] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, May-Jun 2007.

[6] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[7] F. Warmerdam, "The geospatial data abstraction library," *Open source approaches in spatial data handling*, pp. 87–104, 2008.

[8] J. L. Hintze and R. D. Nelson, "Violin plots: a box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.