geosciences

Article

A Python Application for Visualizing an Imbricate Thrust System: Palomeque Duplex (SE, Spain)

Manuel Bullejos [1] and Manuel Martín-Martín [2,*]

[1] Departamento de Álgebra, University of Granada, 18010 Granada, Spain; bullejos@ugr.es
[2] Departamento de Ciencias de la Tierra y Medio Ambiente, University of Alicante, 03080 Alicante, Spain
* Correspondence: manuel.martin.m3@gmail.com

Abstract: This paper introduces a Python application for visualizing an imbricate thrust system. The application uses the traditional geologic information to create an HTML geological map with real topography and a set of geological cross-sections with the essential structural and stratigraphic elements. On the basis of the high geological knowledge gained during the last three decades, the Palomeque sheets affecting the Cenozoic Malaguide succession in the Internal Betic Zone (SE Spain) were selected to show the application. In this area, a Malaguide Cretaceous to Lower Miocene succession is deformed as an imbricate thrust system, with two thrusts forming a duplex, affected later by a set of faults with a main strike-slip kinematic. The modeled elements match well with the design of the stratigraphic intervals and the structures reported in recent scientific publications. This proves the good performance of this Python application for visualizing the structural and stratigraphic architecture. This kind of application could be a crucial stage for future groundwater, mining, and civil engineering management.

Keywords: Python libraries; 3D geological map; geological cross-sections; imbricate thrust system; Malaguide domain; Internal Betic Cordillera

Citation: Bullejos, M.; Martín-Martín, M. A Python Application for Visualizing an Imbricate Thrust System: Palomeque Duplex (SE, Spain). *Geosciences* **2023**, *13*, 207. https://doi.org/10.3390/geosciences13070207

Academic Editors: Jingshou Liu, Wenlong Ding, Ruyue Wang, Lei Gong, Ke Xu, Ang Li and Jesus Martinez-Frias

Received: 19 May 2023
Revised: 22 June 2023
Accepted: 6 July 2023
Published: 11 July 2023

1. Introduction

Numerical modeling has recently been progressively employed to characterize the geological architecture of structures and sediments. These numerical tools involve a great variety of geospatial and geostatistical techniques, as well as stochastic and deterministic models. The used criteria depend on the input spatial coverage and data typology, and the validation of predictions and capability for calibrating, as well as the purpose of the research. In fact, numerical tools allow measuring the accuracy of the represented parameters and variables interactive in 3D modelling and visualizations [1–6]. As a result, interactive visualizations (more accurate and real than classic 2D representations) are performed. Moreover, modern 3D models allow incorporating new data into the designs during the proper generation. Therefore, numerical modelling tools are very interesting in the different aspect of applied geology (civil engineering, mining, and hydrogeology) for decision-making.

A wide variety of 3D applications for visualizing, based on different programming languages and interpolation algorithms, are available, including commercial tools, such as MOVE [7], 3D Geomodeller [8], Gocad [9], Autocad Civil [10], VOXI [11], ArcGis [12], Geoscene 3D (I-GIS) [13], and PETREL [14], as well as open-source software, such as Gempy [15] and OSGeo [16].

For instance, there are Python libraries with different geological interests, including sedimentology structural geology, hydrogeology, and mining. There are Python libraries [17–19] and post-listing libraries [20,21]. Several computer tools devoted to different fields of geology also appear in scientific documents [22–25]. Additionally, some social media channels have also published analytics data and machine learning educational applications on geological aspects [26,27].

It is noteworthy that commercial applications, which are usually more expensive, have technical support for users and are user-friendly environments. Contrarily, open-source applications are free of cost and have high adaptability (extending or modifying the sources), but there is an absence of technical support for users, and they show low reliability in some cases.

Our previous experience with Python libraries for 3D visualization, including geological data handling [28–31], has been very important to develop new applications for visualizing 3D structural and stratigraphic geological features. These new applications included Jupyter notebooks, describing the methodology, and a Python code operative version, which are downloadable from the GitHub repository (https://github.com/bullejos/visualizing-an-imbricate-thrust-system accessed on 4 July 2023). The created model can be opened with any web browser (without installing additional software) to view different perspectives, enlarge or focus on a single element or specific area, hide elements, and take snapshots in every moment.

This new research aims the application of Python libraries to construct 3D structural and stratigraphic representations of an imbricate thrust system, using, as an example, the Palomeque Duplex from the Malaguide Complex [32–34] in the Mula-Pliego Depression from the Sierra Espuña area [35–40] in the Murcia province (SE, Spain: Figure 1). The methodology presented in this article can be exported to other geological systems. For that purpose, is necessary to know the open-source Python language, but it is not necessary to be an expert in the language. The details of our methodology are presented in a notebook hosted in a GitHub repository. This notebook serves a Python beginner as a step-by-step guide through the model creation, so an interested researcher can follow suit and adapt the inputs and mimic the procedures to create their own model. Our method is mainly based on the matplotlib graphical library, which is a very comprehensive and simple graphical Python library. There is no difficulty to install or set up the Python environment in which our methodology works, since most of the libraries or modules used are very common and are usually installed in the standard Python environments. To set up the environment where a geological open-source tool works is usually difficult, sometimes because the tool is not updated using old Python module versions; in other cases, these tools usually require a big amount of data (usually obtained from geological drilling), which is difficult to obtain in remote locations and without a great economic support. The data used to build our model is obtained by a geologist with classical fieldworks (geological mapping, cross-section performing, and measuring of stratigraphic columns) with no machinery. Basic geologist experience is needed to adapt the model to reality. Finally, we highlight that one of the biggest advantages of our method is the output format provided to the model, as an HTML file, which only requires a web browser to open and manipulate it.

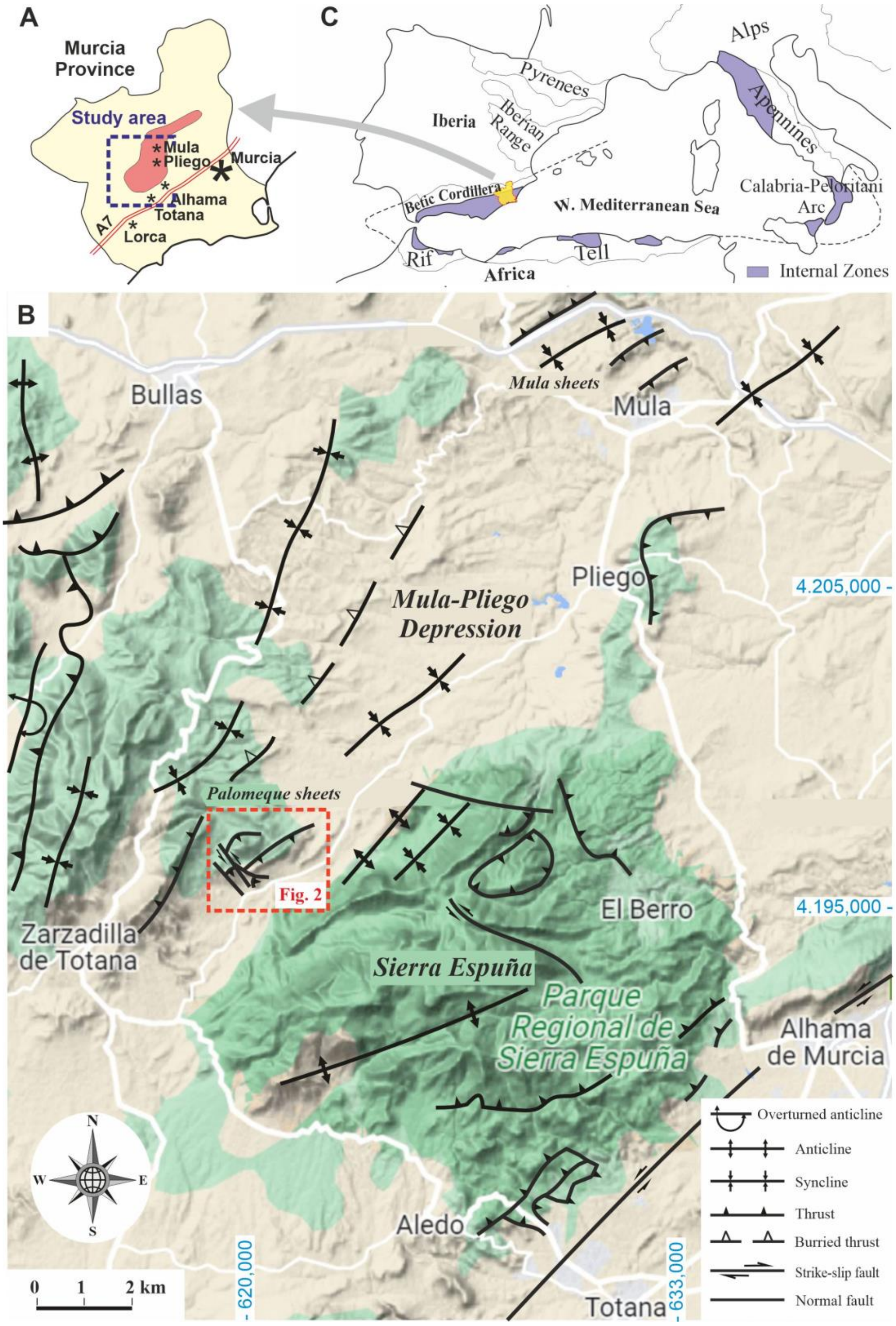

Figure 1. Geographical and geological location of the Sierra Espuña area. (**A**) Geographic locations in the Murcia province (SE Spain) with location of box C; (**B**) Map with the main structural elements of the Sierra Espuña area, including Sierra Espuña s.s. and the Mula-Pliego Depression, with location of box for Figure 2 in the Palomeque sheets area; (**C**) Geographic location of the Betic Cordillera (S Spain) in the framework of the western Mediterranean Alpine chains.

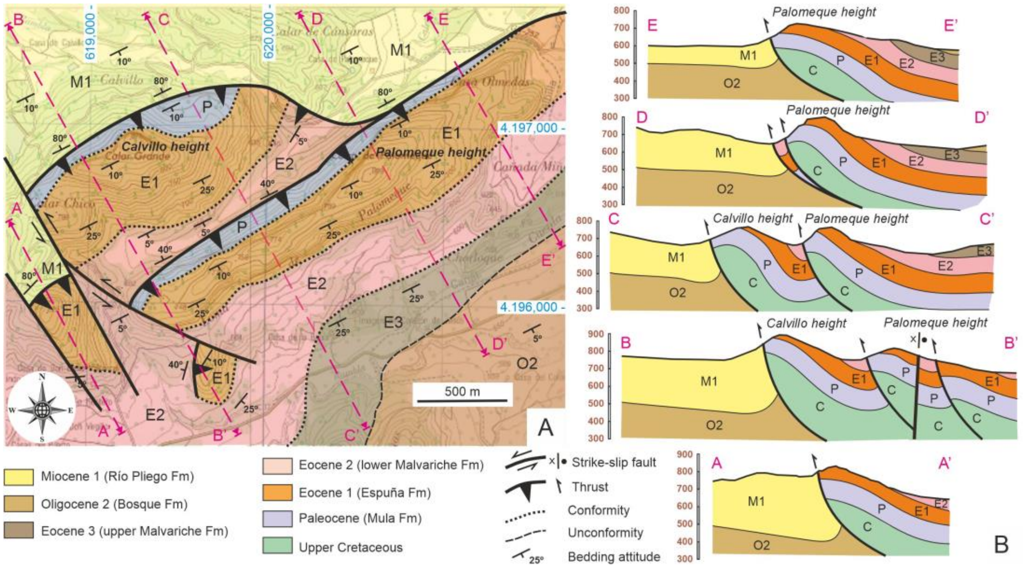

Figure 2. Geological map and cross-sections from the Palomeque sheets (modified from [32–40]). (**A**) left side: geological map of the Palomeque sheets with location of the geological cross-sections; (**B**) right side: a set of five geological cross-sections (A-A′, B-B′, C-C′, D-D′, and E-E′) located in the left map and oriented perpendicular to the main structures.

2. Study Area

The Sierra Espuña area belongs to the Murcia province in SE Spain (Figure 1A). It is surrounded by the towns of Mula and Bullas to the north, Aledo and Totana to the south, Zarzadilla de Totana to the west, and El Berro to the east, with Pliego located in a center position (Figure 1B). Geologically, this area belongs to the Malaguide Complex from the Internal Betic Cordillera [32–40]. The Internal Zone of this cordillera usually shows metamorphic rocks in the lower Nevado-Filabride and Alpujarride complexes [41]. The upper Malaguide Complex is the only one showing a sedimentary basement (sometimes epimetamorphic) and cover [35–42].

The Betic Cordillera belongs to the Alpine chains surrounding the western Mediterranean area from the Gibraltar Arc to the Calabria-Peloritani Arc (Figure 1C) [42–45]. All these chains were formed during the Cretaceous to Miocene era, due to continental collisions with subductions between the Iberia-Europe and Africa plates, and a few collisions of intermediate microplates. This led to a comparable geology in all the branches of this orogenic belt, with intense tectonics in the Internal Zones and metamorphism in the lower units [41,42].

The area mainly shows the Mesozoic-Cenozoic cover (with a thin sheet of Paleozoic basement) from the Malaguide Complex [38–40]. The entire area is structured in the Mula-Pliego Depression (a synclinorium with minor structural highs, related to fold-thrust) to the north (Figure 1B), followed, to the south, by the Sierra Espuña s.s. (an antiformal stack) [32–34].

The area has a quite continuous sedimentary record and outcrops of good quality from the entire succession, so that a large number of papers have been published in the last three decades ([32–40] and references therein). In particular, the Paleocene to Early Miocene tectono-sedimentary evolution has been the subject of numerous research papers, making this basin a key site for understanding the Cenozoic tectono-stratigraphy of the region [38].

In the Mula-Pliego Depression, several structural heights (Figure 1B) crop out as the Mula and Palomeque sheets [32–34].

Concretely, the goal of this paper is the modeling for visualizing one of these structural heights: the Palomeque sheets (Figure 2). These sheets are made of two heights: Palomeque and Calvillo, from S to N. These consist of a Cretaceous to Lower Miocene succession, deformed, in an imbricate thrust system, with two thrusts forming a duplex (Figure 2), related to the Lower Miocene deformation in the area as part of the Malaguide wedge-top basin in the area [32–34]. This structure is affected later by three faults, with main strike-slip kinematics (Figure 2).

The main structural elements are the two thrusts and the three sinistral strike-slip faults (Figure 2), while the stratigraphic succession has been divided into seven homogeneous stratigraphic intervals (Figure 2) [32–34]: Cretaceous (marly limestones and marls), Paleocene (Mula Fm: sands and calcarenites), Eocene 1 (Ypresian Espuña Fm: small-size Nummulite-rich limestones), Eocene 2 (Lutetian Lower Malvariche Fm: marls), Eocene 3 (Bartonian Upper Malvariche Fm: large-size Nummulite-rich limestones), Oligocene 2 (Upper Oligocene Bosque Fm: marls and marly limestones), and Miocene 1 (Aquitanian Río Pliego Fm: turbidite sandstones and silts).

3. Methodology

Python [46] is one of the most popular open-source programming languages at this moment, and it is applied in a wide variety of environmental subjects, including geology. In this work, we have made use of this high-level language to build geological maps and cross-sections, but, also, some tools apart from Python have been used (Table 1). The main Python graphic library used to create the geological models is matplotlib lib [47]. This is a comprehensive library used by many Python programmers which has several active forums where it is possible to ask and consult many questions (https://discourse.matplotlib.org/ accessed on 4 July 2023 or https://community.plotly.com/ accessed on 4 July 2023). Jupyter [48] has been used to create a notebook (https://github.com/bullejos/visualizing-an-imbricate-thrust-system accessed on 4 July 2023), containing all the Python instructions needed to build the geological maps and cross-sections. This notebook is hosted in the GitHub repository (https://github.com/bullejos/visualizing-an-imbricate-thrust-system accessed on 4 July 2023).

Table 1. General tools used, apart from Python.

General Tools	
Google Earth pro	Free software that allows visualization and creation of geospatial data
GPSVisualizer	Free online utility that can be used to handle geographic data
Anaconda	Open-source platform that is very useful for beginners to install and handle Python environments
Jupyter	Web application for creating and sharing computational documents

It is recommended to download the notebook and run it locally in your computer. In the notebook hosted in the GitHub repository (https://github.com/bullejos/visualizing-an-imbricate-thrust-system accessed on 4 July 2023), the needed data to obtain the model can also be found in the data folder. Also, the resulting HTML and PNG files are hosted in the figures folder, but, to open the HTML files, it is necessary to download them, because they are too big to be opened directly in GitHub. GitHub will not show the content of a file with a size of over 1 MB. In the GitHub repository, a readme file is included with the instructions to handle the notebook. A Markdown file (visualizing a thrust system.md) is also included, which can be read directly in the GitHub repository, along with all the instructions in the notebook. To create the geological models, standard mathematics and data processing Python libraries have been used. Those libraries usually come integrated with any standard Python environment (for example, Anaconda). When a library is not included in your environment, the commands “conda install” or “pip install” can be used to install the library. You can consult the library website to see how to install it. In Table 2, the Python libraries used are reported.

Table 2. List of Python modules and libraries used.

Python Modules and Libraries	
Pandas	Python package for data analysis
NumPy	Python library for working with numerical data
SymPy	Python library for symbolic mathematic
Plotly	Python library for data visualization
Matplotlib	Python graphic library
Pillow	Python library for image processing
Bezier.py	Python module to create Bezier curves

Some custom functions have also been defined (Table 3). There are two types of custom functions: (1) those used to create the plot data, and that use plot commands such as "Scatter3d", "Mesh3d", or "Surface". The use of these functions reduces the amount of input lines in the code; (2) those used to deal with geometric objects such as lines, planes, or triangles and help to make some mathematical computations. In the Jupyter notebook, the procedure followed to build the geological model of the Palomeque Duplex is explicitly given, and the methodology given in the notebook can be adapted to other zones or geological structures. The methodology presented here is not automatic; it is not a defined script that allows one to get the model from the geological data without any manipulation. To get a model of another location, the procedures presented here can be mimicked, but these must be adapted to the new data of the new location, and researchers must make their own choices of how the model should be. Python has been used by us to produce other geological models in the recent past, but with a different perspective and different kind of data, as is the case of the Llobregat River Delta (Barcelona, Spain) [28–31]. To build this model in Barcelona, a very complete database of geological drilling was available. In the Palomeque area, the geological data has been obtained by the geologist by classic geological fieldworks without machinery.

Table 3. List of Python custom functions used.

Custom Functions	
surface	Function that interpolates from a grid
contact_tr	Function that prepares data from cvs files to be used for plotting
contact_dat	Function that uses the matplotlib function "Scatter3d" to obtain the plot data that allows to visualize marks or lines
eq and tr	Two functions, used to compute triangles that are uses as geological symbols
fault	Function that prepares data to visualize faults
plane_3d	Function that uses matplotlib command "Mesh3d" and prepares the data to visualize a plane
plane, plane_y, impl_line, para_line, inter_line_plane	Functions that get different mathematical equations of lines, planes, and their intersections
bz	Function that uses Bezier functions to compute Bezier curves from a point, a direction, and lists of intervals and heights
get_cvs	Reads cvs files and prepares the data to be used in the definition of the 3D figure
plane_data	Uses the plot function "Scatter3d" to produce the data used to define the figure for the 3D model

3.1. Data Compilation and The Topographic 3D Model Performing

The starting point to build the 3D model is the geological map, presented in Figure 2A. In that map, tectonic and stratigraphic contacts are represented, as well as dipping values in several points, and also the location (traces) of the geological cross-sections. A flow diagram (Figure 3) is presented, synthetizing the methodology for the 3D model's performance. The complete process of how to get the model is completely described in the Jupyter notebook, visualizing a thrust system, hosted in the GitHub repository: https://github.com/bullejos/visualizing-an-imbricate-thrust-system (accessed on 4 July 2023).

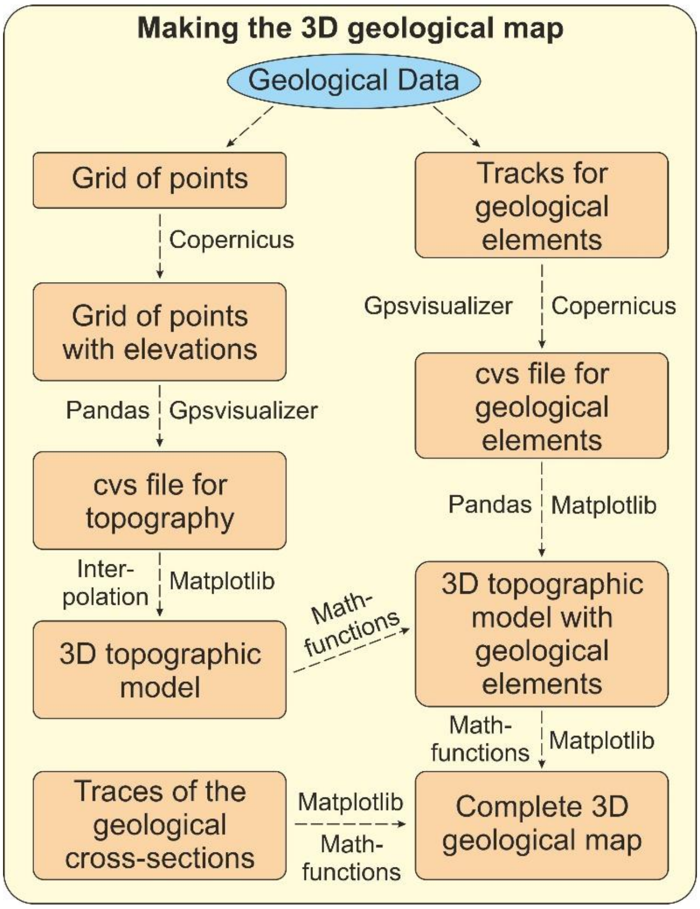
Making the 3D geological map
Geological Data
Grid of points
Copernicus
Grid of points with elevations
Pandas
Gpsvisualizer
cvs file for topography
Inter-polation
Matplotlib
3D topographic model
Math-functions
Tracks for geological elements
Gpsvisualizer
Copernicus
cvs file for geological elements
Pandas
Matplotlib
3D topographic model with geological elements
Math-functions
Matplotlib
Traces of the geological cross-sections
Matplotlib
Math-functions
Complete 3D geological map

Figure 3. Flow diagram summarizing the methodological stages for the creation of the 3D geological map.

The first step to build the 3D model of the study area is to define a grid of points covering the area. Google Earth pro [49] has been used to build a box (a rectangle) enclosing the study area. This rectangle is built as a track by Google Earth and exported as a kml file with the corresponding data. The track (kml file) needs to be converted to a cvs file so that it can be manipulated, and the online resource GPSVisualizer [50] is applied to do this task.

Then, a distance is fixed and a grid covering the area is built using Python language. This grid consists of a list of points with UTM_X and UTM_Y coordinates. The Python package Pandas [17] is used to process this kind of data. With Pandas, the grid data is exported to a cvs file, which has two columns: one each for the UTM_X and UTM_Y coordinates of the points in the grid. Moreover, the elevations of those points are also needed. To get these elevations, the Copernicus Python package [51] is used together with the Copernicus DEM (Digital Elevation Model) [52] and, in particular, the 30-m EEA-10, GLO-30 instance. At this moment, a file (topography.cvs) with the data of a grid of points (UYM_X, UTM_Y, elevation) covering the study area is produced.

Now, to get the topographic model of the area, interpolation is needed. There are some scientific computing packages used to interpolate. The main Python mathematical packages or libraries used here are Numpy [53], Simply [54], and SciPy [18]. At the beginning of the Jupyter notebook, some code lines are included that import the necessary packages to be used (https://github.com/bullejos/visualizing-an-imbricate-thrust-system/blob/main/visualizing%20a%20thrust%20system.ipynb accessed on 4 July 2023). There are three main interpolation methods included in the packages: linear, cubic, and nearest. After analyzing the three methods, the linear method has been chosen, because it is the one that best suits our expectations. The plot command "Surface" is then used to get a 3D model of the topography of the area.

We have to add the rest of the geological elements (thrusts, strike-slip faults, and stratigraphic contacts) to the topographic model. Again, Google Earth is used to create tracks for the contacts (traces) of these geological elements, and GPSVisualizer is used to turn those tracks into cvs files. Finally, Copernicus is used to complete those files with elevations. The "Scatter3d" plot command with mode = lines is then used to add those elements to the model. The last elements we have to add are the traces of the geological cross-sections (intersections of those planes with the topography). For that purpose, lists of points in the directions of the planes are created. With Copernicus, the elevations of those points have been calculated. Again, the "Scatter3d" plot command has been used to represent the traces of the geological cross-sections. Some mathematical computations are also needed to draw the triangles associated to the thrust contacts. To determine each of those triangles, a point U in the corresponding track (which will be the starter vertex), and the length D of the side of the triangle, are fixed. Then, a second point V in the track at a distance D of U is computed. With those two vertices, a third one W is computed to complete the equilateral triangle. Some mathematical functions are defined to do these computations. To complete the model, we add some texts by specifying the positions where the texts must be placed; also, geological symbols for faults (which are just broken lines) are added. The constructed geological map by Python allows viewing the geological contact with real topography.

3.2. Construction of the 2D Geological Cross-Sections with the Essential Geological Elements

The first figure created (sketch_map.png) is a sketch which represents a top view of the area. In this figure, all the geological elements are included, as well as the straight lines representing the geological cross-sections (traces).

The geological cross-sections are rendered as 2D figures that are stored as png files in https://github.com/bullejos/visualizing-an-imbricate-thrust-system/tree/main/figures (accessed on 4 July 2023). The upper curve that appears in any cross-section corresponds to the intersection of the vertical plane that determines the section with the topography. Those intersections were already calculated when the 3D geological map model was created. These calculations are real and performed by us using real elevations, which is one of the advantages of our method. To construct the rest of the curves (geological contacts) inside the geological cross-sections, we must calculate the intersections of the tracks (which determine the faults, thrust, and stratigraphic contacts) with the vertical planes corresponding with the geological cross-sections. Some mathematical functions are implemented to calculate parametric and implicit equations of planes and lines, and also to calculate intersections of lines and planes (Table 3). Three types of curves have been tested: polynomial, periodic, and Bezier curves. Our experience indicates that Bezier curves are the ones that best fit our context. To determine a Bezier curve, a list of control points is needed. In our case, the first and last control points of each curve are usually on the tracks of the geological elements. The rest of the control points are chosen so that the Bezier curve fits with the geological data we have in the geological cross-section from Figure 2. The way in which the control points, and, thus, Bezier curves are constructed, and the way the corresponding data is manipulated to get the geological cross-sections can also be found in the notebook. The flow diagram that summarizes the process is given in Figure 4.

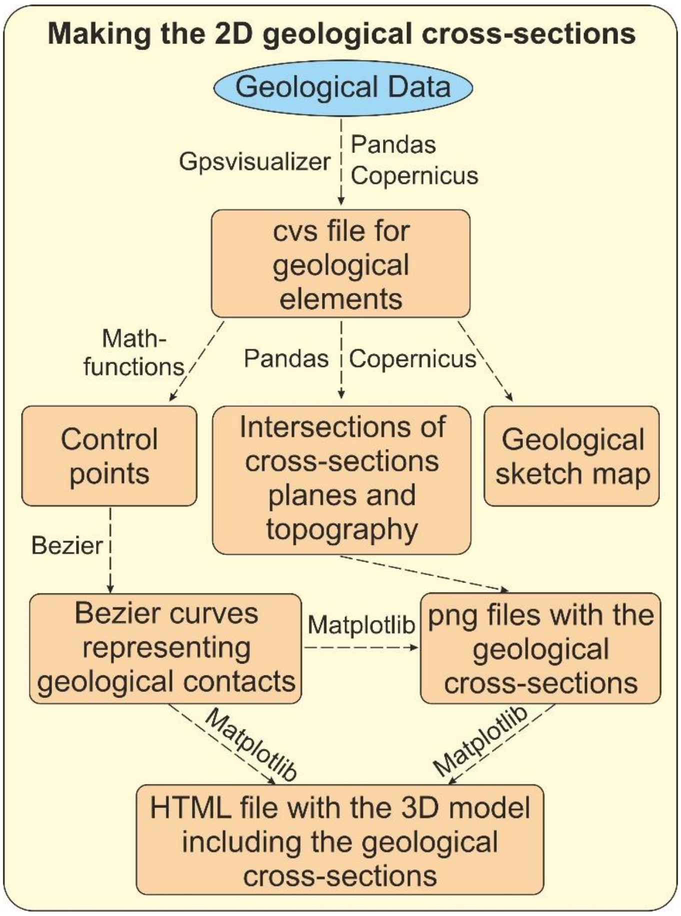
Making the 2D geological cross-sections
Geological Data
Gpsvisualizer
Pandas
Copernicus
cvs file for geological elements
Math-functions
Pandas
Copernicus
Control points
Intersections of cross-sections planes and topography
Geological sketch map
Bezier
Bezier curves representing geological contacts
Matplotlib
png files with the geological cross-sections
Matplotlib
Matplotlib
HTML file with the 3D model including the geological cross-sections

Figure 4. Flow diagram synthetizing the methodological stages for the creation of the 2D geological cross-sections, and the final integration in a 3D HTML model with the geological map.

These geological cross-sections performed by Python are constructed in an exact way; the upper line in each section represents the topography and the positions of the geological contacts, with respect to the topography and other surfaces are mathematically calculated using the custom defined functions (Table 3).

4. Results

4.1. Geological Map of the Palomeque Sheets as an HTML File

The same geological map area presented in Figure 2A was created as an interactive HTML file (3D_Palomeque_map.html) using Python language. Figure 5 shows five screenshots from this 3D HTML model. This map shows the duplex structure and strike-slip lateral faults with the real topography. In the geological map, the seven essential stratigraphic intervals are represented: Cretaceous, Paleocene, Eocene 1 (Ypresian), Eocene 2 (Lutetian), Eocene 3 (Bartonian), Oligocene, and Miocene 1 (Aquitanian). The same vertical vs. horizontal ratio was used to plot the geological map, in order not to alter the geological proportions and dippings. It is necessary to note that only a web browser is needed to open the model and that it allows to view different perspectives, enlarge or focus on a single element or specific area, hide elements, and take snapshots in every moment.

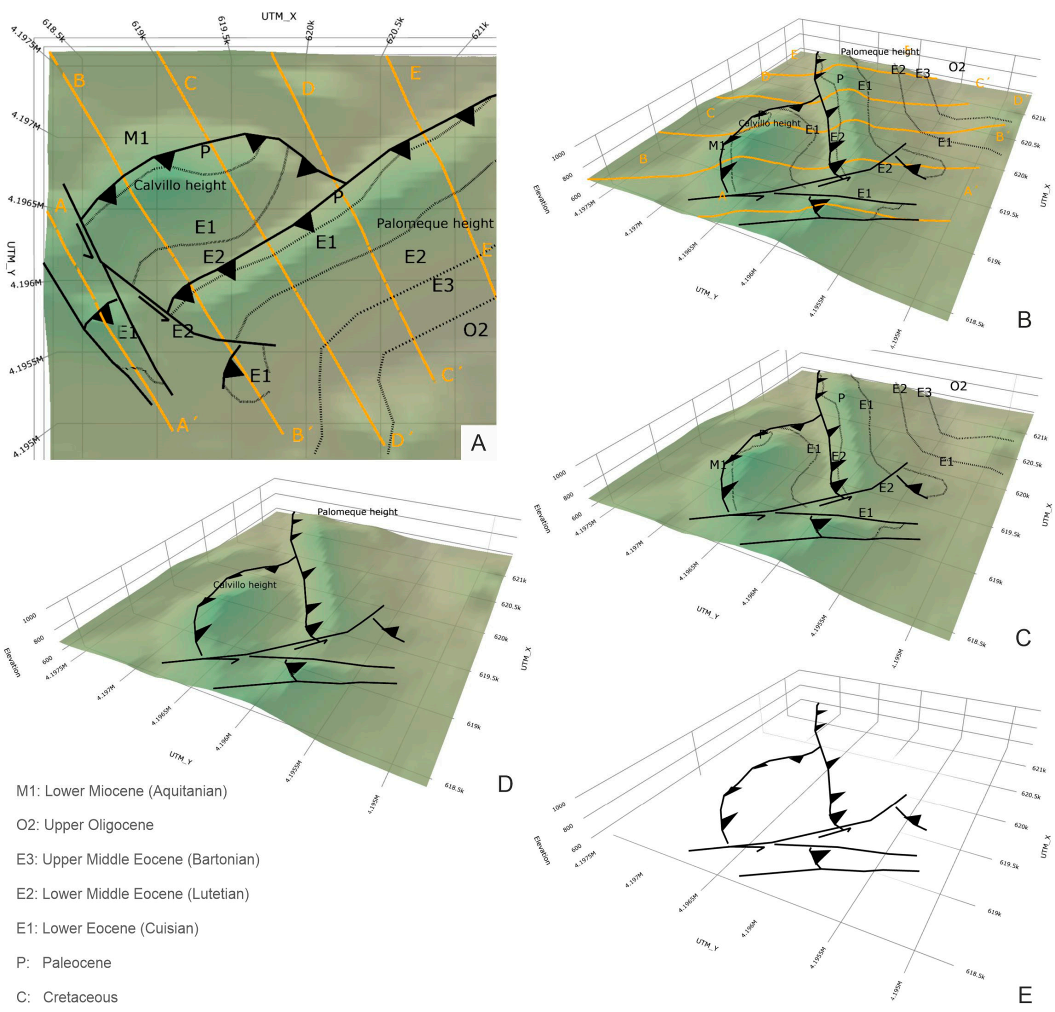

Figure 5. Screenshots with views of the interactive 3D HTML geological map with real topography performed, provided in Supplementary Materials (3D_Palomeque_map.html). The seven essential stratigraphic intervals, the stratigraphic contacts and the mechanic ones (thrusts and strike-slip faults), are represented. (**A**) zenithal view; (**B**) zenithal-oblique view from the SW with all the elements turned on; (**C**) zenithal-oblique view from the SW with height and geological cross-sections turned off; (**D**) zenithal-oblique view from the SW with geological cross-sections and stratigraphic contacts turned off; (**E**) zenithal-oblique view from the SW only with tectonic contacts turned on.

4.2. Geological Cross-Sections of the Palomeque Sheets Using Python Language

The same five vertical geological cross-sections (A-A′, B-B′, C-C′, D-D′, and E-E′) presented in Figure 2B were created using Python language (Figure 6). These are located oriented NW-SE perpendicular to the duplex structure. In the geological cross-sections, the seven essential stratigraphic intervals are also represented: Cretaceous, Paleocene, Eocene 1 (Ypresian), Eocene 2 (Lutetian), Eocene 3 (Bartonian), Oligocene, and Miocene 1 (Aquitanian). Equally, the same vertical vs. horizontal ratio was used to plot all the geological cross-sections in order not to alter the geological proportions and dips.

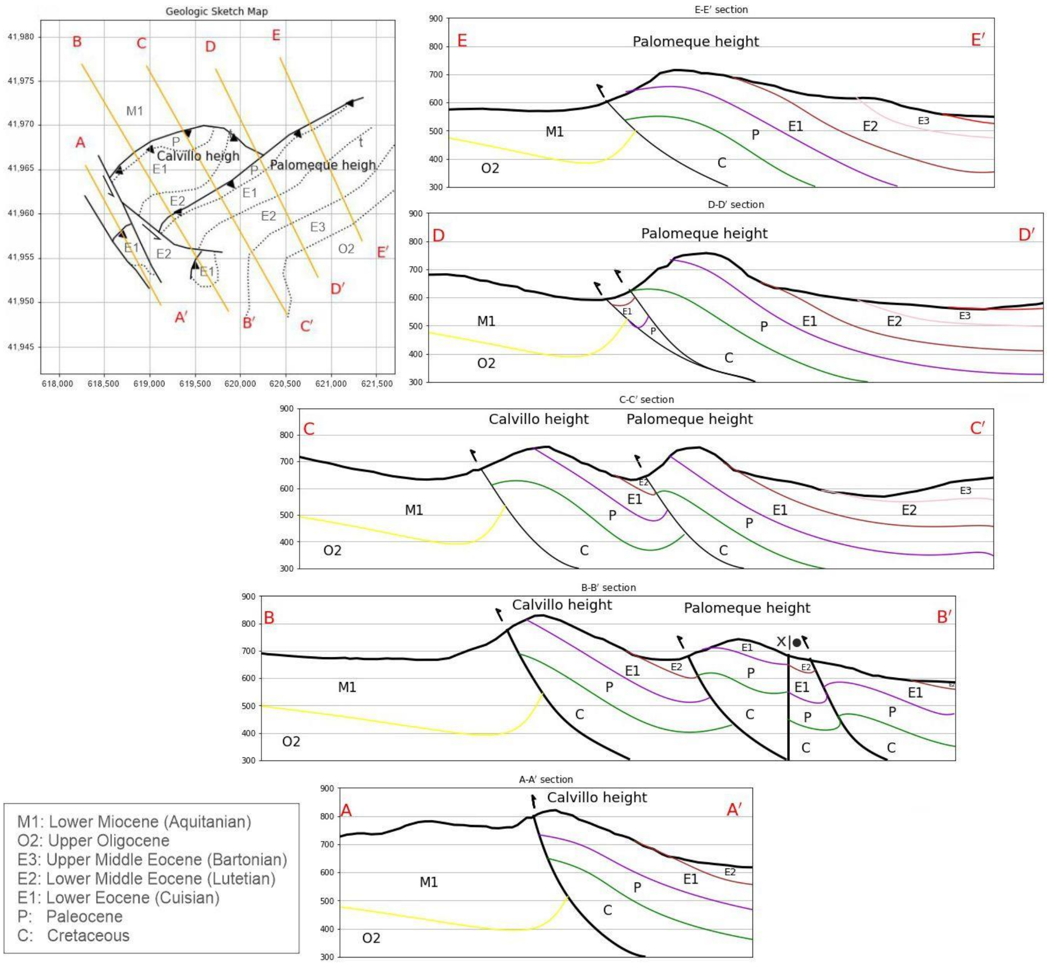

Figure 6. Geological cross-sections performed by Python located in the index sketch structural map (upper-left side): A-A′, B-B′, C-C′, D-D′ and E-E′. The seven essential stratigraphic intervals are represented.

4.3. Integration of the 3D of Geological Map and the Geological Cross-Sections

Finally, the geological cross-sections are integrated with the geological map in a 3D figure (Figure 4). The custom functions “plane_3d” and “plane_data” use the matplotlib functions “Mesh3d” and “Scatter3d” to prepare the data that conform the 3D model (Table 3). The output for this model is the HTML file (3D_Palomeque_map_sections.html) is provided as Supplementary Material. Equally, only a web browser is needed to open the model. This HTML representation allows viewing different perspectives, enlarging or focusing on a single element or specific area, hiding elements by clicking in the lateral legend, and take snapshots in every moment. Figure 7 shows six screenshots from this new 3D HTML model. This figure shows the duplex structure and strike-slip lateral faults with the real topography, but also the underground developing of these structures by means of the representation in five geological vertical cross-sections. In the geological map and cross-sections, the seven essential stratigraphic intervals are represented: Cretaceous, Paleocene, Eocene 1 (Ypresian), Eocene 2 (Lutetian), Eocene 3 (Bartonian), Oligocene, and Miocene 1 (Aquitanian). The same vertical vs. horizontal ratio was used to plot the geological map, in order not to alter the geological proportions and bedding attitude. Each geological cross-section can be turned off separately. Also, the topography or mapped elements in surface can be turned off. Moreover, the structural elements can be hidden separately to the stratigraphic contacts in the geologic map.

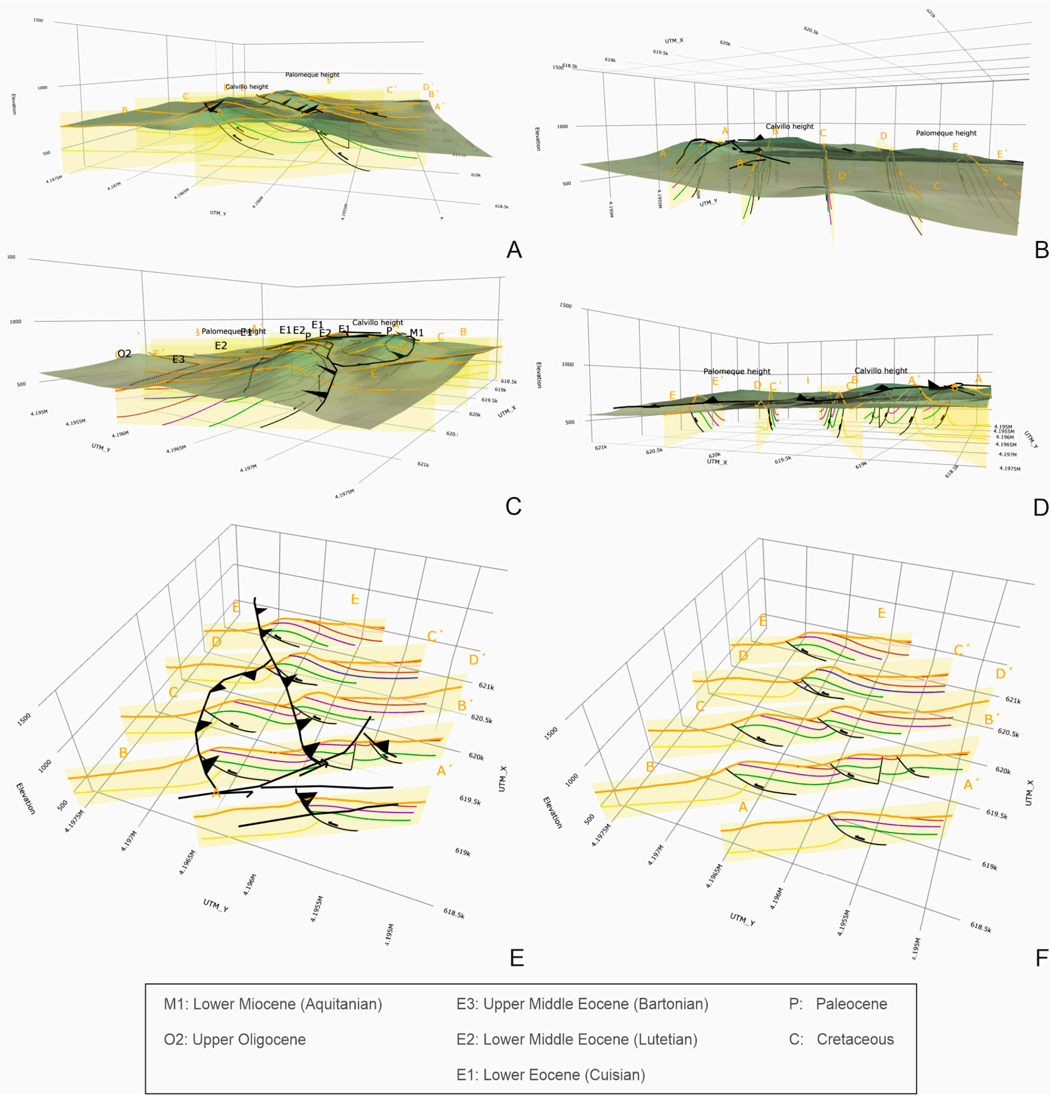
A
B
C
D
E
F
Calvillo height
Palomeque height
Elevation
UTM_X
UTM_Y
M1: Lower Miocene (Aquitanian)
O2: Upper Oligocene
E3: Upper Middle Eocene (Bartonian)
E2: Lower Middle Eocene (Lutetian)
E1: Lower Eocene (Cuisian)
P: Paleocene
C: Cretaceous

Figure 7. Screenshots with views of the interactive geological map with real topography and geological cross-sections performed by Python, provided in Supplementary Material (3D_Palomeque_map_sections.html). (**A**) lateral view from the W; (**B**) lateral view from the S; (**C**) lateral view from the NE; (**D**) lateral view from the N; (**E**) zenithal-oblique view from the SW with geological cross-sections and tectonic contacts, and the rest of the elements turned off; (**F**) zenithal-oblique view from the SW only with geological cross-sections and the rest of the elements turned off.

5. Discussion and Conclusions

The above presented geological map (Figure 5), cross-sections (Figure 6), and the integration of both (Figure 7) performed by Python fit well with the classical hand-made geological map and cross-sections (Figure 2). The geological map allows viewing the geological contacts (mechanic and stratigraphic) with real topography. On the other hand, the geological cross-sections constructed by Python are performed in an exact way, since the position of the geological contacts, with respect to the topography and other surfaces, are mathematically obtained in an exact way.

For the final design of the curves in the geological cross-sections (Figure 8), Bezier curves are used. These curves provide a very useful tool in computer graphics design. A Bezier curve is determined by a list of control points, and the complexity of the curve depends on the number of control points that determine it. For example, the Bezier curve determined by two points is a straight line; by three points, it is a quadratic curve (parabolic); for four points, it is a cubic curve, etc. Looking at the curves in Figure 2B that appear in the cross-sections and that represent thrusts or stratigraphic contacts, it can be seen that they can be constructed as Bezier curves.

Using an example to illustrate the situation (Figure 9), it is needed to construct a curve that represents a stratigraphic contact, such as the Paleocene-Eocene 1 (P-E1) from the section E-E′, shown in Figure 2B (Figure 9A). It is started with the control points F (marked by the geological map and topography) and G (obtained from the general dipping of beds in the geological map). Both points mark the beginning and the end of the curve (Figure 9B). The Bezier curve determined by these two control points is a straight line (Figure 9B). Then, if it is needed to draw an anticline near the origin F and a syncline near the end in G, it is necessary to add a new control point H in the transition anticline-syncline (Figure 9C).

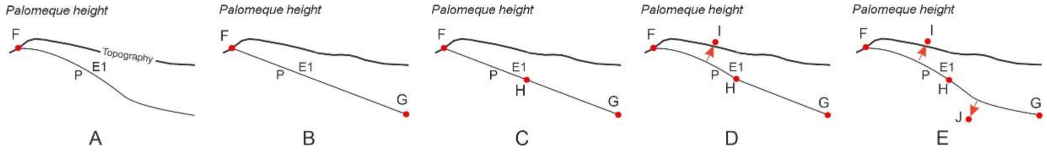

Figure 8. Bezier curves by Python: (**A**) original curve, as an example, in Figure 2B (contact P-E1 and topography from section E-E′); (**B**) Bezier curve determined by two control points [F, G]; (**C**) Bezier curve determined by three control points [F, G, H]; (**D**) Bezier curve determined by four control points [F, G, H, I]; (**E**) Bezier curve determined by five control points [F, G, H, I, J].

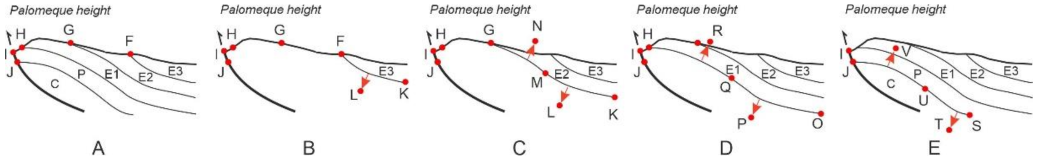

Figure 9. Bezier curves by Python in a geological cross-section with a deformed stratigraphic succession: (**A**) example of section E-E′ from Figure 2B; (**B**) determination of the E2-E3 stratigraphic contact with the control points [F, L, K]; (**C**) determination of the E1-E2 stratigraphic contact with the control points [G, N, M, L, K]; (**D**) determination of the P-E1 stratigraphic contact with the control points [H, R, Q, P, O]; (**E**) determination of the C-P- stratigraphic contact with the control points [J, V, U, T, S].

In order to draw the anticline, a new control point is necessary and is added: point I. Then, the Bezier curve is determined by F, I, H, and G, being the curve composed by a parabolic segment and a straight one (Figure 9D). For the construction of the syncline close to the G point, a new control point J should be added (Figure 9E). Therefore, the Bezier curve determined by the list of control points [F, G, H, I, J] becomes the curve from Figure 9A, and is very similar, or exact, to Figure 2B (section E-E′).

When a deformed stratigraphic succession is represented with several stratigraphic units (Cretaceous, Paleocene, Eocene 1, Eocene 2, and Eocene 3: C, P, E1, E2, and E3, respectively) separated by stratigraphic contacts, as in the case of the same section E-E′ from Figure 2B (Figure 8), the Bezier curves must represent the folds and must also be parallelized to conserve the thicknesses of the stratigraphic units in the whole section and in all the sections. In this case, it is necessary to follow the above-mentioned methodology, starting with the upper contact represented in the geological cross-section by a Bezier curve, and, later, construct lower and lower curves parallelized to the upper one. All the curves have starting control points located in the intersection with the topography and marked by the geological map, or even points of intersection between contacts (control points J, I, H, G, F). The final design depends on the supplementary control points inserted in order to obtain a more similar shape to the geological cross-section from Figure 2B.

In conclusion, from the above-introduced methodology proposed by us and the scope of the achieved structural models, it can be argued that our methodology implemented using Python libraries (matplotlib, numpy, sympy, etc.) seems to be an ergonomic and easy-access tool for visualizing the 3D and 2D structural and stratigraphic architecture of a deformed sector, as is the case of the imbricate thrust system from the Palomeque duplex. The 3D models can be viewed in a web browser, so they are easily accessible to every user. The interactive 3D HTML models of essential geologic elements (structural and stratigraphic) allow making quantitative measures, which is quite valuable to support many geological disciplines, such as the applied hydrogeological, geotechnical, and mining research interests. This methodology can be exported to other geological areas and structures. It is not necessary to be an expert in the Python language. The created methodology is presented in a notebook hosted in a GitHub repository (https://github.com/bullejos/visualizing-an-imbricate-thrust-system accessed on 4 July 2023), so that an interested researcher can follow suit, adapting the inputs and mimicking the procedures to create their own model. Our method is mainly based on the matpoltlib graphical library, which is a very comprehensive and simple graphical Python library. There is no difficulty to install or set up the Python environment in which our mythology works, since most of the libraries or modules used are very common and usually come installed in the standard Python environments. Our methodology has the advantage of not requiring a big amount of data obtained by machinery and not requiring a great economic support. The data used to build our model is obtained by a geologist with classical fieldworks (geological mapping, cross-sections, and measuring of stratigraphic columns). Basic geologist experience is required to adapt the model to reality.

What has been said about the construction of structural models demonstrates the good performance of Python programming language, and shows how packages are a user-friendly software platform for visualizing 3D geological models. For instance, Python application may assist with crucial concerns for structural geology with applications in large civil works and mining, as well as groundwater resources. Currently, additional research on program routines aimed at assessing mapping uncertainty is ongoing. This paper contributes to the field of structural geological tools, which represents poorly documented information in the geology scientific literature.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/geosciences13070207/s1: interactive 3D geological map: 3D_Palomeque_map_sections.html; interactive 3D geological sections: 3D_Palomeque_map_sections.html. Python codes and the detailed instructions to download and run the codes can be found in a GitHub repository at: https://github.com/bullejos/visualizing-an-imbricate-thrust-system (accessed on 4 July 2023).

Author Contributions: M.B. and M.M.-M. contributed to the conceptualization, methodology, formal analysis, data curation, writing, and review of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: Research Project PID2020-114381GB-100 of the Spanish Ministry of Science and Innovation, Research Groups and Projects of the Generalitat Valenciana from University of Alicante (CTMA-IGA), semigrupos.ugr.es from University of Granada and Research Groups FQM-343 of the Junta de Andalucía.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors are grateful to three anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jessell, M. Three-dimensional geological modelling of potential-field data. *Comput. Geosci.* **2001**, *27*, 455–465. [CrossRef]
2. Wycisk, P.; Hubert, T.; Gossel, W.; Neumann, C. High-resolution 3D spatial modelling of complex geological structures for an environmental risk assessment of abundant mining and industrial megasites. *Comput. Geosci.* **2009**, *35*, 165–182. [CrossRef]

3. Ford, J.; Mathers, S.; Royse, K.; Aldiss, D.; Morgan, D.J.R. Geological 3D modelling: Scientific discovery and enhanced understanding of the subsurface, with examples from the UK. *Z. Der Dtsch. Ges. Fur Geowissenschaften* **2010**, *161*, 205–218. [CrossRef]
4. Rohmer, O.; Bertrand, E.; Mercerat, E.D.; Régnier, J.; Pernoud, M.; Langlaude, P.; Alvarez, M. Combining borehole log-stratigraphies and ambient vibration data to build a 3D Model of the Lower Var Valley, Nice (France). *Eng. Geol.* **2020**, *270*, 105588. [CrossRef]
5. Di Bucci, D.; Buttinelli, M.; D'Ambrogi, C.; Scrocca, D.; RETRACE-3D Working Group. RETRACE-3D project: A multidisciplinary collaboration to build a crustal model for the 2016–2018 central Italy seismic sequence. *Boll. Geof. Teor. Appl.* **2021**, *62*, 1–18. [CrossRef]
6. Cirillo, D.; Totaro, C.; Lavecchia, G.; Orecchio, B.; de Nardis, R.; Presti, D.; Ferrarini, F.; Bello, S.; Brozzetti, F. Structural complexities and tectonic barriers controlling recent seismic activity in the Pollino area (Calabria–Lucania, southern Italy)–constraints from stress inversion and 3D fault model building. *Solid Earth* **2022**, *13*, 205–228. [CrossRef]
7. MOVE (Petroleum Experts Ltd.). Available online: https://www.petex.com/products/move-suite/move/ (accessed on 18 June 2023).
8. 3D Geomodeller (Intrepid Geophysics). Available online: https://www.intrepid-geophysics.com/ (accessed on 18 June 2023).
9. Gocad (Emerson Paradigm Roxar). Available online: https://www.emerson.com/es-es/automation/roxar (accessed on 18 June 2023).
10. Autocad Civil (Autodesk, Inc.). Available online: https://www.autodesk.com/ (accessed on 18 June 2023).
11. VOXI (Earth Modeling from Geosoft). Available online: https://www.seequent.com/products-solutions/geosoft-oasis-montaj/voxi-earth-modelling-extension/ (accessed on 18 June 2023).
12. ArcGis. Available online: https://www.esri.es/es-es/arcgis/productos/index (accessed on 18 June 2023).
13. Geoscene 3D (I-GIS). Available online: https://geoscene3d.com/ (accessed on 18 June 2023).
14. PETREL (Geology and Modeling from Schlumberger). Available online: https://www.software.slb.com/products/petrel/petrel-geology-and-modeling (accessed on 18 June 2023).
15. GemPy: Open-source 3D Geological Modeling. Available online: https://www.gempy.org (accessed on 18 June 2023).
16. OSGeo: The Open Source Geospatial Foundation. Available online: https://www.osgeo.org/ (accessed on 18 June 2023).
17. GeoPandas. Available online: https://geopandas.org/en/stable (accessed on 18 June 2023).
18. Scipy. Available online: https://scipy.org (accessed on 18 June 2023).
19. Albion: 3D Geological Models in QGIS. Available online: https://gitlab.com/Oslandia/albion (accessed on 18 June 2023).
20. GISgeography. 15 Python Libraries for GIS and Mapping. Available online: https://gisgeography.com/python-libraries-gis-mapping (accessed on 18 June 2023).
21. Parpoil, B. Open Source and Geology. Available online: https://oslandia.com/en/2020/07/09/geologie-open-source (accessed on 18 June 2023).
22. Hobona, G.; James, P.; Fairbairn, D. Web-based visualization of 3D geospatial data using Java3D. *IEEE Comput. Graph. Appl.* **2006**, *26*, 28–33. Available online: https://ieeexplore.ieee.org/document/1652923 (accessed on 4 July 2023). [CrossRef]
23. Evangelidis, K.; Papadopoulos, T.; Papatheodorou, K.; Mastorokostas, P.; Hilas, C. 3D geospatial visualizations: Animation and motion effects on spatial objects. *Comput. Geosci.* **2018**, *111*, 200–212. [CrossRef]
24. Semmo, A.; Trapp, M.; Jobst, M.; Doellner, J. Cartography-oriented design of 3D geospatial information visualization–overview and techniques. *Cartogr. J.* **2015**, *52*, 95–106. [CrossRef]
25. Miao, R.; Song, J.; Zhu, Y. 3D geographic scenes visualization based on WebGL. In Proceedings of the 6th International Conference on Agro-Geoinformatics, Fairfax, VA, USA, 7–10 August 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 1, pp. 1–6. Available online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8046999 (accessed on 9 June 2022).
26. Husillos, C. Cesarhusrod/sarai_piezo_precip: Mejora de la Documentación (v1.0.1). *Zenodo*. 2022. Available online: https://doi.org/10.5281/zenodo.7197288 (accessed on 4 July 2023).
27. Pyrcz, M. GeostatsGuy Lectures. Available online: https://www.youtube.com/c/GeostatsGuyLectures (accessed on 9 June 2022).
28. Bullejos, M.; Cabezas, D.; Martín-Martín, M.; Alcalá, F.J. A Python Application for Visualizing the 3D Stratigraphic Architecture of the Onshore Llobregat River Delta in NE Spain. *Water* **2022**, *14*, 1882. Available online: https://www.mdpi.com/2073-4441/14/12/1882 (accessed on 4 July 2023). [CrossRef]
29. Bullejos, M.; Cabezas, D.; Martín-Martín, M.; Alcalá, F.J. A K-Nearest Neighbors Algorithm in Python for Visualizing the 3D Stratigraphic Architecture of the Llobregat River Delta in NE Spain. *J. Mar. Sci. Eng.* **2022**, *10*, 986. [CrossRef]
30. Bullejos, M.; Cabezas, D.; Martín-Martín, M.; Alcalá, F.J. Confidence of a k-nearest neighbors Python algorithm for the 3D visualization of sedimentary porous media. *J. Mar. Sci. Eng.* **2023**, *11*, 60. [CrossRef]
31. Martín-Martín, M.; Bullejos, M.; Cabezas, D.; Alcalá, F.J. Using Python libraries and k-Nearest neighbors algorithms to delineate syn-sedimentary faults in sedimentary porous media. *Mar. Pet. Geol.* **2023**, *153*, 106283. [CrossRef]
32. Martín-Martín, M.; Martín-Algarra, A. Thrust sequence and syntectonic sedimentation in a piggy-back basin: The Oligo-Aquitanian Mula-Pliego Basin (Internal Betic Zone, SE Spain). *Comptes Rendus Geosci.* **2002**, *334*, 363–370. [CrossRef]
33. Martín-Martín, M.; Sanz de Galdeano, C.; García-Tortosa, F.J.; Martín-Rojas, I. Tectonic units from the Sierra Espuña–Mula area (SE Spain): Implication on the Triassic paleogeography and the geodynamic evolution of the Betic-Rif Internal Zone. *Geodin. Acta* **2006**, *19*, 1–9. [CrossRef]

34. Martín-Rojas, I.; Sanz de Galdeano, C.; Martín-Martín, M.; García-Tortosa, F.J. Geometry and kinematics of an antiformal stack deduced from brittle structures. Example of the Internal Betic Zone in the Sierra Espuña (province of Murcia, Spain). *Comptes Rendus Geosci.* **2007**, *339*, 506–515. [CrossRef]
35. Tent-Manclús, J.E.; Martín-Martín, M.; Martín-Pérez, J.A.; Serrano, F. Structural evolution of the early Miocene in the eastern Betic Internal-External Zone Boundary (SE Spain). *Bull. Soc. Géol. Fr.* **2001**, *172*, 41–47. [CrossRef]
36. Martín-Martín, M.; Martín-Rojas, I.; Caracuel, J.E.; Estévez-Rubio, A.; Martín-Algarra, A.; Sandoval, J. Tectonic framework and extensional pattern of the Malaguide Complex from Sierra Espuña (Internal Betic Zone) during Jurassic-Cretaceous: Implications for the Westernmost Tethys geodynamic evolution. *Int. J. Earth Sci.* **2006**, *95*, 815–826. [CrossRef]
37. Perri, F.; Critelli, S.; Martín-Algarra, A.; Martín-Martín, M.; Perrone, V.; Mongelli, G.; Zattin, M. Triassic redbeds in the Malaguide Complex (Betic Cordillera–Spain): Petrography, geochemistry and geodynamic implications. *Earth-Sci. Rev.* **2013**, *117*, 1–28. [CrossRef]
38. Perri, F.; Critelli, S.; Martín-Martín, M.; Montone, S.; Amendola, U. Unravelling hinterland and offshore palaeogeography from pre-to-syn-orogenic clastic sequences of the Betic Cordillera (Sierra Espuña), Spain. *Palaeogeogr. Palaeoclimatol. Palaeoecol.* **2017**, *468*, 52–69. [CrossRef]
39. Martín-Martín, M.; Robles-Marín, P. Alternative methods for calculating compaction in sedimentary basins. *Mar. Pet. Geol.* **2020**, *113*, 104132. [CrossRef]
40. Moliner-Aznar, S.; Martín-Martín, M.; Rodríguez-Estrella, T.; Romero-Sánchez, G. The Cenozoic Malaguide Basin from Sierra Espuña (Murcia, S Spain): An Example of Geological Heritage. *Geosciences* **2021**, *11*, 34. [CrossRef]
41. Jabaloy-Sánchez, A.; Martín-Algarra, A.; Padrón-Navarta, J.A.; Martín-Martín, M.; Gómez-Pugnaire, M.T.; Sánchez-Vizcaíno, V.L.; Garrido, C.J. Lithological successions of the internal zones and flysch trough units of the betic chain. In *The Geology of Iberia: A Geodynamic Approach, Regional Geology Reviews*; Quesada, C., Oliveira, J.T., Eds.; Springer Nature: Singapore, 2019; pp. 377–432.
42. Amodio-Morelli, L.; Bonardi, G.; Colonna, V.; Dietrich, D.; Giunta, G.; Ippolito, F.; Zuppetta, A. L'arco calabro-peloritano nell'orogene Appenninico-Maghrebide. *Mem. Soc. Geol. Ital.* **1976**, *17*, 1–60.
43. Filice, F.; Liberi, F.; Cirillo, D.; Pandolfi, L.; Marroni, M.; Piluso, E. Geology map of the central area of Catena Costiera: Insights into the tectono-metamorphic evolution of the Alpine belt in Northern Calabria. *J. Maps* **2015**, *11*, 114–125. [CrossRef]
44. Iannace, A.; Vitale, S.; D'errico, M.; Mazzoli, S.; Di Staso, A.; Macaione, E.; Bonardi, G. The carbonate tectonic units of northern Calabria (Italy): A record of Apulian paleomargin evolution and Miocene convergence, continental crust subduction, and exhumation of HP LT rocks. *J. Geol. Soc.* **2007**, *164*, 1165–1186. [CrossRef]
45. Vitale, S.; Ciarcia, S. Tectono-stratigraphic and kinematic evolution of the southern Apennines/Calabria–Peloritani Terrane system (Italy). *Tectonophysics* **2013**, *583*, 164–182. [CrossRef]
46. Python Programming Language. Available online: https://www.python.org (accessed on 18 June 2023).
47. Matplotlib. Available online: https://pypi.org/project/matplotlib/ (accessed on 18 June 2023).
48. Jupyter. Available online: https://jupyter.org/ (accessed on 18 June 2023).
49. Google Earth Pro. Available online: https://maps.google.com/intl/es/earth/download/gep/agree.html (accessed on 18 June 2023).
50. Gpsvisualizer. Available online: https://www.gpsvisualizer.com/ (accessed on 18 June 2023).
51. Copernicus. Available online: https://copernicus.readthedocs.io/en/latest/getting-started.html (accessed on 18 June 2023).
52. Copernicus DEM. Available online: https://spacedata.copernicus.eu/es/collections/copernicus-digital-elevation-model (accessed on 18 June 2023).
53. Numpy. Available online: https://numpy.org (accessed on 18 June 2023).
54. Simpy. Available online: https://www.sympy.org/en/index.html (accessed on 18 June 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.