UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE GEOCIÊNCIAS

METHODS AND APPLICATIONS FOR GEOLOGICAL DIRECTIONAL DATA ANALYSIS

Arthur Endlein Correia

Orientador: Prof. Dr. Ginaldo Ademar da Cruz Campanha

DISSERTAÇÃO DE MESTRADO

Programa de Pós-Graduação em Geoquímica e Geotectônica

Versão Corrigida
SÃO PAULO
2017

Ficha catalográfica preparada pelo Serviço de Biblioteca e Documentação
do Instituto de Geociências da Universidade de São Paulo

Correia, Arthur Endlein
Methods and Applications for Geological Directional Data Analysis / Arthur Endlein Correia. – São Paulo, 2017.
92 p. + anexos + CD-ROM

Dissertação (Mestrado) : IGc/USP
Orient.: Campanha, Ginaldo Ademar da Cruz

1. Geologia estrutural 2. Dados direcionais 3. Software I. Título

Arthur Endlein Correia

Methods and Applications for Geological Directional Data Analysis

Dissertação apresentada ao Programa de Pós-Graduação em Geoquímica e Geotectônica do Instituto de Geociências da Universidade de São Paulo como requisito parcial para a obtenção do título de Mestre em Ciências.

Área de Concentração: Geotectônica
Orientador(a): Prof. Dr. Ginaldo Ademar da Cruz Campanha

São Paulo

2017

para Francisco Janssen

Acknowledgements

Aos meus pais, minha irmã e minha família, pelo apoio e paciência;

À Lilian;

Ao Ginaldo Campanha, pela orientação e paciência infindável;

Ao Carlos Grohmann, pelas conversas e paciência durante o longo desenvolvimento do OpenStereo;

À Camila Viana, cuja ajuda tornou este trabalho possível;

Ao Eduardo Ribeiro, Eduardo Takafuji, Gustavo Ramos, João Paulo Abujamra, Luiz Urtiga, Matheus Andrade e Mayko Oliveira, que também foram fundamentais ao processo;

Ao Ricardo Francesconi, Rafael Campanhã e Natalia Santiago;

Ao Gustavo Abreu, Marcos Egydio, Teodoro Isnard, Caetano Juliani, Marcelo Rocha, Jorge Kazuo Yamamoto e demais professores do Instituto de Geociências;

Às equipes da Seção de Pós-Graduação, Graduação, Biblioteca, Gráfica e demais funcionários do Instituto de Geociências;

À CAPES pelo apoio financeiro através de bolsa de mestrado;

Esta dissertação foi escrita em LaTeX com a classe IAGTESE.

"*We have no idea, now, of who or what the inhabitants of our future might be. In that sense, we have no future. Not in the sense that our grandparents had a future, or thought they did. Fully imagined cultural futures were the luxury of another day, one in which 'now' was of some greater duration. For us, of course, things can change so abruptly, so violently, so profoundly, that futures like our grandparents' have insufficient 'now' to stand on. We have no future because our present is too volatile. ... We have only risk management. The spinning of the given moment's scenarios. Pattern recognition*"

William Gibson, Pattern Recognition

Resumo

OpenStereo foi desenvolvido originalmente para preencher uma lacuna entre aplicativos de análise para geologia estrutural, como um software livre, gratuito e multi-plataforma. Ao longo dos anos ele adquiriu um grande número de usuários, com citações regulares. Este trabalho objetivou a reestruturação do OpenStereo como um todo, mudando-o para uma nova estrutura de interface gráfica e construíndo-o do zero visando desempenho, estabilidade e facilidade de manutenção e extensão. Diversas novas funcionalidades foram incluídas tais como projetos, conversão de notação de atitudes, ajuste de pequenos círculos, extração de atitudes de modelos tridimensionais e conversão de shapefiles de linhas para dados circulares. A pesquisa gerou dois subprodutos principais: um novo método gráfico para ajuste de pequenos círculos e a biblioteca de análise de dados estruturais Auttitude.

Palavras chave: Geologia Estrutural, Dados Direcionais, Software

Abstract

OpenStereo was originally developed to fill a gap among software packages for structural geology analysis, as a free open source cross-platform software. Over the years it has acquired a great number of users, with regular citations. This work aimed to restructure OpenStereo as a whole, changing to a new graphical interface framework and building it from the ground up for speed, stability, ease of maintenance and extension. Many new functionalities were also included, such as project management, structural attitudes notation handling, small circle fitting, extractions of attitudes from three-dimensional models and conversion of lines shapefiles to circular data. The research involved had two main byproducts, a new graphical method for small circle data fitting and a directional data analysis library, Auttitude.

Key words: Structural Geology, Directional Data, Software

Summary

Chapter 1

Introduction

1.1 Structure of the dissertation

This dissertation is organized into six chapters. The current chapter introduces the topic, outlines the goals of the work, and provides an overview and basics to the OpenStereo software.

The second chapter describe the computer language and libraries used throughout the research.

The third, fourth and fifth chapters make up the bulk of the developed work and present PDFs of research articles that have been prepared to submission to international journals. The first one is dedicated to the construction of a Python directional data analysis library, with emphasis on Tectonics and Structural Geology related data and presents a review about directional data. The second presents a numerical and a graphical methods for small circle data fitting. Both papers were the foundation to rebuild OpenStereo, that is presented in the fifth chapter.

The sixth chapter concludes the dissertation. It summarizes the key achievements and findings.

Source code and example files can be found at https://git.io/vMjma and on the attached CD.

1.2 An introduction to OpenStereo

OpenStereo (Grohmann e Campanha, 2010; Grohmann et al., 2011) is an open source, cross-platform software for directional data projection and structural geology that was released to fill a gap between software packages for structural geology analysis. The soft-

ware was written in Python, a high-level, cross-platform programming language and the GUI was designed with wxPython. Numeric operations were performed with the Numpy module and all graphic capabilities were provided by the Matplolib library.

The original version had data input with simple ASCII text files, with values of dip direction/trend and dip/plunge separated by spaces, tabs or commas, supporting planes in direction notation using the Right Hand Rule. Multiple files can be opened at the same time (or the same file more than once), and different elements of each dataset (poles, great circles etc) can be overlaid. The GUI shows the opened files in a tree structure, similar to "layers" of many illustration software, where the vertical order of the files in the tree reflects the drawing order of the selected elements (Fig. 1.1). At that stage, the software performed plotting operations of poles to planes, lineations, great circles, density contours and rose diagrams. A set of statistics was calculated for each file and its eigenvalues and eigenvectors were used to suggest if the data is clustered about a mean value or distributed along a girdle. Modified Flinn, Triangular and histograms plots were also available.

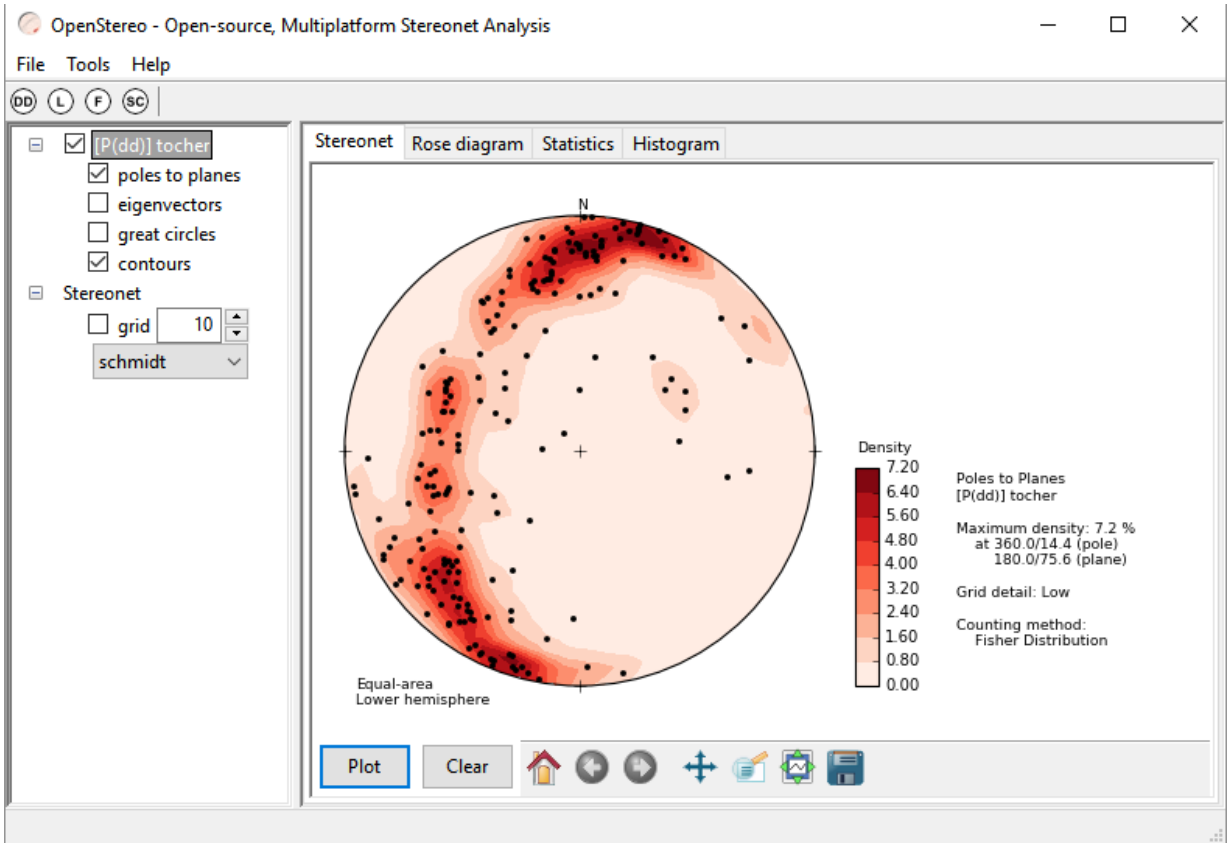

Figure 1.1: Image of the main window on the Stereonet tab, showing a contoured dataset.

Over the years OpenStereo have acquire a great number of users, being cited in several

articles such as the most recent Gehrmann et al. (2016),Moreno-Sánchez et al. (2016) and Phillips et al. (2016). Even back in its first release, the authors predicted a development that would focus on building additional tools such as merging and rotation of datasets, possibility to save projects and paleostress analysis.

Unfortunately, the original graphical interface toolkit used, WxPython, ceased to be developed. Also, while the original code was made to be extensible, it had some structural elements that would make deep changes difficult. As such, a complete rewrite was needed.

1.3 Objectives

This work pursued several objectives. The foremost and key to the others was the rebuilding of OpenStereo. This would happen in two steps, first the building of a directional data analysis library, Auttitude, which would be responsible for data loading, preprocessing and numerical analysis, and second OpenStereo itself, restructured as mainly an interface for Auttitude, focusing on data objects management and user interaction.

This separation of concerns was a strong requirement for the project, to allow it to be more extensible and easily maintainable, while also preventing at least some of the damage where the new graphical toolkit used also cease to be maintained and developed.

A second objective was the building and adaptation of new tools to OpenStereo, such as small circles fitting, project management, projection rotation, new methods to calculate contours and rose diagrams, among others.

Chapter 2

Methods

During the restructuring OpenStereo was effectively divided in three parts: a data model and processing system, an abstraction layer and a graphical interface. The purpose of this division was to make each part as independent as possible, allowing for almost separate development and strongly simplifying management.

The first part, the data model and processing system, came to be a full fledged directional data analysis library, Auttitude. Though mainly built for OpenStereo, it is generic, and may both be used on its own on a Python interactive shell or used to build other software. It contains functions to load datasets, translate notation and classes to store directional data, related statistics and counting grids. It was developed using the Python standard library, Numpy (Oliphant, 2006) and xlrd (Python Software Foundation, 2016).

Both the abstraction layer and graphical interface reside in OpenStereo. The graphical toolkit used was Qt, through the pyside library (https://wiki.qt.io/PySide). The main window and dialogs were created in Qt Designer, and compiled to Python using the pyside-uic routine, part of the pyside library.

The abstraction layer was then responsible for connecting the Auttitude Directional-Data objects to the graphical interface, handling calls to the Matplotlib plotting library (Hunter, 2007), storing plot settings, populating and parsing the properties dialogs and building programmatically the graphical interface elements that couldn't be created on Qt Designer.

To accomplish these tasks it heavily relies on reflection, which is the ability of a computer program to examine, introspect and modify its own structure at runtime. The items that represent loaded data on the layer tree on OpenStereo are derived from an abstract class called DataItem. When a DataItem derived object is created, it introspects and find

all its methods whose name starts with "plot_", creating a checkbox as a subitem using the remaining of its name as the checkbox label.

When the plot method of the main window is called, either meaning that the plot button was clicked or some other method called it directly, the abstraction layer searches for all instantiated DataItems that are checked and asks for their plot elements. In turn, each DataItem search itself for its subitems that are checked and calls their respective "plot_" methods, which will return plot elements that will be dispatched to one of the panels.

In concrete terms, this means that derived classes such as PlanarData can be considerably simple, only concerning itself with providing methods that will return the needed data and plot parameters, as color and symbol. It will then be possible to focus future developments on numerical problems, as for example paleotension calculation for faults, as the graphical interface will effectively build itself and connect the data to the plots.

The properties dialogs for the data types are connected in a similar but simpler way. By following a naming convention for the widgets assembled on Qt Designer the abstraction layer is able to find where in the DataItem object should the related property be retrieved and stored.

Three main additional tools were included in OpenStereo, ply2atti, which is a routine to extract attitudes from three-dimensional models (Viana et al., 2016), a shapefile to circular data conversion tool and a Small Circles Fitting method. Ply2atti uses numpy and NetworkX (Hagberg et al., 2008) to read and extract information from the models, while the shapefile tool uses the pyshp library (Ian Bicking and Michael Foord, 2005) to parse shapefiles. The Small Circles Fitting method uses Numpy for vectors and linear algebra.

Chapter 3

AUTTITUDE: Object Oriented Computational Model for Directional Data

AUTTITUDE: object oriented computational model for directional data with emphasis on tectonics and structural geology

Arthur Endlein[a,*], Ginaldo Ademar da Cruz Campanha[a], Camila Duelis Viana[a], Carlos Henrique Grohmann[b]

[a] *Institute of Geosciences, University of São Paulo, São Paulo, 05508-080, Brazil*
[b] *Institute of Energy and Environment, University of São Paulo, São Paulo, 05508-010, Brazil*

Abstract

Directional data (that is, data that may be represented as unit vectors) have many different uses in the geological sciences, either, for example, on structural geology, representing the attitudes of planar or linear structures or in sedimentology, used for paleocurrent direction analysis. They are usually represented graphically, but translating them into vectors has many advantages. This work aims to create an object oriented directional data analysis library, with emphasis on Tectonics and structural geology related data, ease of use, and as a mathematical engine for OpenStereo, a directional data analysis software. Considering that it is a Python library, it may both be used for creating new directional data analysis software or in a interactive shell as an analysis framework. Its integration to OpenStereo allows complex data sets to be easily visualized, at the same time giving it speed, stability and ease of maintenance.

Keywords: Directional data, Python.

*Corresponding author
Email addresses: arthur.correia@usp.br (Arthur Endlein), ginaldo@usp.br (Ginaldo Ademar da Cruz Campanha), camila.viana@usp.br; camila.duelis@gmail.com (Camila Duelis Viana), guano@usp.br (Carlos Henrique Grohmann)

1. Introduction

Directional data are those that can be represented as unit vectors, typically in two or three dimensions, representing points on a spherical surface (3D) or a circle (2D), line orientations or rotations. They are also referred as orientation, angular or spherical data, and can be vectors, if they have direction and sense, or axial, if they have only direction (Fisher et al., 1987; Jupp and Mardia, 1989).

There are multiple examples of their application within geology: on crystallography, representing the relative orientation of crystal faces; on structural geology, representing the orientation of planar and linear structures; on paleomagnetism, as remaining magnetization poles; and on Tectonics, both on plate tectonic kinematics, representing hot spot traces, and transform faults.

Their representation in Geology is usually made in projections for three-dimensional data (Equal-Area and Equal-Angle mostly, ortographic and gnomonic rarely), or as rose or kite diagrams for two-dimensional data. Also, distributions and numerical statistical methods have been known for quite some time and are available in most software for the treatment of structural orientation data. The representation of these data in the form of vectors allows for greater clarity and ease of operation, facilitating their manipulation in software.

The main goal of this work was the production of an easy to use and extensible object-oriented directional data analysis library, with emphasis on geotectonics and structural geology data processing. The main use of the library was in the remodeling of OpenStereo, as its numerical engine, but it can also be used directly in a interactive Python shell.

2. Directional data analysis

Statistical analysis of directional data in geology can be traced to Schmidt (1917), while studying structures in slates, developing both the use of the rose diagram and the normal distribution enveloped in the circle. Most of the advances in circular data processing at the time occurred in geology, with the notable exception of von Mises (1918), who applied it in error modeling in the

determination of atomic weights, developing the statistical distribution that bears his name, and that is still the most used in this type of data (Fisher, 1993).

Three-dimensional directional data statistical analysis was started by Fisher (1953), when defining a distribution for vectors concentrating around an average vector equivalent to the normal distribution, developed to treat paleomagnetism data. Known as Fisher distribution, it was latter joined by a number of other distributions on the sphere, such as Watson (Bingham, 1964; Watson, 1965), for the concentration of axial data around a mode or in great circles, Bingham-Mardia (Mardia and Gadsden, 1977; Bingham and Mardia, 1978), for concentrations in small circle girdles and the generic distributions of Bingham for axial data (Bingham, 1964, 1974) and Kent for vector data (Kent, 1982), that can describe a number of possible distribution shapes depending on their parameters.

In addition to these more formal statistical distributions and their associated parameters, a number of other ad-hoc numerical methods have been created, many of which are very useful for the analysis of geological data. Kamb (1959) is an example, defining criteria for detecting uniformity deviation in count meshes, later revisited and expanded by Robin and Jowett (1986). In addition, Woodcock (1977) and later Vollmer (1990) developed methods to classify the general data form from the eigenvalues of their orientation matrix, serving as an initial non-parametric basis for data treatment. Circular data are generally represented in the form of rose or kite diagrams built as circular histograms, binning the data in sectors around the circle. However, some studies (e.g. Fisher (1993), Munro and Blenkinsop (2012)) propose a continuous curve based on moving averages, similar to the methods applied on spherical data with counting meshes.

2.1. Attitude notation

In geology there is a great variety of methods for recording measured attitudes, specially for planar data. This is an additional obstacle to the numerical treatment of this type of data, as its manual conversion is generally necessary.

With careful parsing, though, it is possible to read and analyze most of these different notations.

The attitude of a typical planar layer is recorded as a horizontal orientation and a vertical inclination, the first as a direction or a dip direction and the second as a dip. Using dip direction notation is usually unambiguous, whereas using direction has the problem that it may mean either of two opposite dipping planes. To solve that, either additional information on to which quadrant does the layer dips to should be appended to the data, usually after the dip angle, or a convention has to be used as to which direction to record, such as the right-hand rule that states that the recorded direction should always be the one that is 90 degrees counter-clockwise to the dip direction.

These horizontal orientations can be either presented as an azimuth, ranging from 0 to 360 (or 0 to 180, or 270 to 90, for directions), or as a quadrant, ranging from N0E to N90E or N0W to N90W or S0E to S90E or S0W to S90W, depending on the quadrant it is in and whether direction or dip direction is used. Measurements are sometimes recorded in unusual ways, such as direction measures counted from the south, either as a particular method of the geologist who collected the data or simply by inexperience, or approximate measures such as NS or EW, or only N, S, E or W.

It is not possible in general to determine whether a particular measurement was taken using direction or dip direction, but from this basic information the rest can be converted without great difficulties by software. The following regular expressions are the basis of this analysis:

$$([NSEW]\{0,2\})(\backslash d*)([NSEW]\{0,2\}) \tag{1}$$

$$(\backslash d+)([NSEW]\{0,2\}) \tag{2}$$

Regular expressions (Kleene, 1951; Pilgrim, 2009) are a method to validate and analyze texts in a robust and fast way, allowing to define which variations are acceptable and extracting specific pieces of the analyzed text. In the above

cases each parenthesis indicates one of the analyzed pieces, being for Eq. 1, ([NSEW] $\{0, 2\}$) the first letter of the azimuth, which must be one of N, S, E and W, NE, NW, SE, SW or empty; ($\backslash d^*$), the azimuth itself, which must be a number or empty; ([NSEW] $\{0, 2\}$), the second letter of the azimuth, which must also be one between N, S, E and W, NE, NW, SE, SW, or empty; and for Eq. 2, ($\backslash$ d +), the dip, which must be a number; and [NSEW] $\{0, 2\}$), the dip quadrant, which must be one of N, S, E, W, NE, NW, SE, SW, or empty.

From the presence (some value) or absence (empty) of each of these parts (except for the separator and the dip value) the truth table Tab. 1) shows acceptable patterns for attitude data, omitted cases being considered errors.

From this information and the fact that it is a direction or dip direction, it becomes possible to convert these attitudes to a single format, which is then used for the rest of the analysis. AUTTITUDE uses dip direction / dip notation as this single format.

2.2. Vectors and algebra

Although most operations between directional data can be done by other non-vector methods (e.g. graphical methods such as projections or numerical ones through flat or spherical trigonometry), it is difficult to outgrow vectors in terms of syntax clarity and ease of programming. Vectors are sets of numbers, one for each dimension represented, sorted as a single row or column. They are a special case of matrices, which have m rows by n columns (Fergusson 1994).

To represent directional data as vectors it is necessary to convert the entry attitudes (or coordinates, or angles in general) to directional cosines, which are numerically the cosine of the angle between the attitude and x, y and z axes (typically North, East and upwards, for geological attitudes).

Considering a lineation as an example, We first divide the vector resulting from the attitude into its vertical and horizontal components (Kim, 2005). As the plunge is the angle between the line and the horizontal plane, its vertical component will be the sine of this angle, leaving its cosine as its horizontal component. Since the trend is the angle between the North (X) axis and the

horizontal component of the vector, this will be further divided between the cosine of the trend angle for the X axis and the sine of the plunge for the Y axis. In short, the following formula holds:

$$u = \begin{bmatrix} \sin(trend)\cos(plunge) \\ \cos(trend)\cos(plunge) \\ -\sin(plunge) \end{bmatrix} \tag{3}$$

An equivalent formula for plane attitudes is constructed in a similar way using the plane normal vector as its vector representation.

The inverse conversion, from direction cosines to attitudes, is typically made calculating the arcsin (or arccos for planes) of the vertical component and the arctan of the ratio between y and x components (or x and y for planes), and comparing the signs to determine the trend quadrant.

Direction cosines of circular data have a much simpler construction, as their x component will just be the cosine of the azimuth and the y component its sine, and the inverse conversion is the same as for the horizontal components of line data. For axial directional data it is recommended (Fisher, 1993) to double the azimuth before the calculation of direction cosines and then dividing the result by two when performing the inverse conversion, to distribute the vectors along the whole circle and avoid problems such as described in Section 2.3 for axial spherical data.

2.3. Descriptive statistics

These direction cosines can then be used to obtain some descriptive statistics for spherical directional data (Fisher et al., 1987). Being $x_i = \{x_i, y_i, z_i\}$ the direction cosine of each data point, the resulting vector and its length are calculated as,

$$r = \sum x_i, \quad \bar{x} = \frac{r}{n}, \quad R = |r|, \quad \bar{R} = \frac{R}{n} \tag{4}$$

The resulting vector r gives a general sense of the direction of points. The average length $\bar{R}$ will be closer to 1 as the more concentrated the points are

around the resulting vector.

For directional data, average lengths close to zero may either mean that the vectors have no preferred orientation or that they are clustered around two opposing direction with a similar density of points.

As for axial data, it may also mean that the points have no preferred orientation, but it could be an artifact of the vector representation. Points that are highly concentrated around a direction that is nearly horizontal will be represented as vectors on opposing sides of the sphere, resembling a situation similar to the second one for directional data.

A reasonably simple correction is to compare the data points with the first eigenvector of their dispersion matrix (Section 2.4), which will also be close to their mean direction but will not be affected by this phenomenon, as the calculation of the dispersion matrix is insensitive to inversion of points. From this, it is possible to concentrate the vectors in a single hemisphere relative to this eigenvector, avoiding some problems with the calculation of these parameters and other treatments. This method is particularly useful for simple concentrations of points, but may not be adequate for data around small circle girdles with large semi-apical angles.

The same statistics can be calculated for circular data, being also possible to define the circular variance V and circular standard deviation v (Fisher, 1993) as:

$$V = 1 - R, \quad v = \sqrt{2\log(1 - V)} \tag{5}$$

2.4. Eigen analysis and shape parameters

These basic parameters of dispersion around the mean are then complemented by the shape parameters, extracted from the dispersion matrix T (also called the orientation matrix), defined as follows:

$$T = \frac{1}{n} \sum x_i^T \cdot x_i \tag{6}$$

The eigenvectors (u_i) and eigenvalues (λ_i) of this matrix represent the axes and respective lengths of the data best fit ellipsoid (Pearson, 1901). Considering the eigenvalues in decreasing order and their corresponding eigenvectors, the points distribution can be described with relation to extreme ellipsoidal shapes (Table 2). Woodcock (1977) and Vollmer (1990) define ratios that clarify the concept of big and small eigenvalues. For Woodcock's method, the ratios

$$x = \ln(\frac{\lambda_1}{\lambda_2}) \tag{7}$$

$$y = \ln(\frac{\lambda_2}{\lambda_3}) \tag{8}$$

are used to analyze the data points distribution shape through Fig. 1.

However, this graph is not of simple interpretation. Vollmer (1990), on the other hand, proposes three equations that describe the dataset proximity to: point distribution (Eq. 9), in which the eigenvalues would be 1, 0 and 0; girdle (Eq. 10), with eigenvalues 0.5, 0.5 and 0; or random (Eq. 11), with eigenvalues 1/3, 1/3 and 1/3. He also defines a cylindricity parameter (Eq. 12), based on the logarithmic ratio between the first and third eingenvectors. These coordinates can then be plotted to the triangular diagram on Fig. 2, being easier to understand.

$$P = \frac{\lambda_1 - \lambda_2}{N} \tag{9}$$

$$G = \frac{2(\lambda_2 - \lambda_3)}{N} \tag{10}$$

$$R = \frac{3\lambda_3}{N} \tag{11}$$

$$C = \ln(\frac{\lambda_1}{\lambda_3}) \tag{12}$$

2.5. Counting grids

Counting grids are comprised of roughly regularly spaced nodes on the sphere or circle, where each node is calculated either by the fraction of points that are at an angular distance smaller than a chosen value or by a total value that depends on the distance to each data point. Either way, it represents an estimate of the probability density function for the data set.

There are several different methods of producing a regular grid for calculating this estimate (Diggle and Fisher, 1985; Kamb, 1959). A flexible form to build the grid is suggested by Robin and Jowett (1986), in which the only parameter is the great circle distance of the nodes, t. This value is directly used as a regular spacing of latitude circles. For each circle with an angle to the vertical direction of v, nodes are spaced longitudinally along the circle with separation according to Eq. 13.

From the direction cosines of the grid nodes the angle to each point is then calculated by Eq. 14, and the value for each node can either be calculated as in Robin and Jowett (1986) through Eq. 15 or counting the number of points with angles within a limit. The constant K at Eq. 15 or the limit angle for counting has the fundamental effect of smoothing the data points along the sphere allowing to emphasize large tendencies and reduce noise. The choice of this factor must be made experimentally, but both Robin and Jowett (1986) and Diggle and Fisher (1985) suggest formulas to estimate optimal smoothing values and, based on the first, AUTTITUDE calculates these values if not given by the user, allowing Diggle and Fisher (1985) as an optional.

For circular data it is a much simpler operation, only dividing the circle by the appropriate angular spacing to obtain the nodes, typically 1 degree for geological data, and applying either Eqs. 16 or 18 for each counting node, being h a smoothing factor. From these nodes it is possible to produce a continuous rose or kite diagram, which presents itself as a very interesting tool for visualizing circular data (Fisher, 1993; Munro and Blenkinsop, 2012). Wells (2000) suggests another method for smoothing circular data, which is effectively an average of all

possible rose diagrams considering different petal sizes and offsets from north.

$$a = \frac{2(\sin(\frac{t}{2}))}{\sin(v)} \tag{13}$$

$$\theta = a\cos(\frac{v \cdot u}{|v||u|}) \tag{14}$$

$$w = \sum e^{k(\cos\theta - 1)} \tag{15}$$

$$w = \frac{0.9375}{n * h} \sum (1 - \frac{\theta^2}{h^2})^2 \tag{16}$$

$$w = \frac{0.9375}{n * h} \sum (1 - \frac{\theta^2}{h^2})^2 \tag{17}$$

$$Mw_a = \frac{1}{A} \sum_{i=\alpha-\frac{A-1}{2}}^{n=\alpha+\frac{A-1}{2}} F_i \cdot w^{|\alpha-\frac{A-1}{2}|} \tag{18}$$

2.6. Rotations and Change of Axes

It is often convenient or necessary to change the coordinate axes or rotate the data. In the first case, the new coordinate system is defined by three orthogonal axes x', y' and z' which must be of unit length to avoid deforming the spatial distribution of the data, making the points no longer belong to the surface of a unit sphere. In order to project data to this system, the product expressed on Eq. 19 is calculated. As an example, the projection of a data set using its eigenvectors as new axes may be useful for symmetry check. Rotation, on the other hand, is made from a unit axis v and the anti-clockwise rotation angle around it θ from the matrix equation 20. Rotations can be used for restitution, for example, trying to reestablish the sedimentary transport trend of a cross

bedding stratification that was tectonically tilted.

$$u' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \cdot u = \begin{bmatrix} x'_x u_x + x'_y u_y + x'_z u_z \\ y'_x u_x + y'_y u_y + y'_z u_z \\ z'_x u_x + z'_y u_y + z'_z u_z \end{bmatrix} \tag{19}$$

$$u' = R{\cdot}u = \begin{bmatrix} \cos\theta + v_x^2(1-\cos\theta) & v_x v_y(1-\cos\theta) - v_x \sin\theta & v_x v_z(1-\cos\theta) + v_y \sin\theta \\ v_x v_y(1-\cos\theta) - v_z \sin\theta & \cos\theta + v_y^2(1-\cos\theta) & v_y v_z(1-\cos\theta) - v_x \sin\theta \\ v_x v_z(1-\cos\theta) + v_y \sin\theta & v_y v_z(1-\cos\theta) - v_x \sin\theta & \cos\theta + v_z^2(1-\cos\theta) \end{bmatrix} \cdot u \tag{20}$$

2.7. Projections

The most typical form of directional data representation in geology is definitely the projection (Fisher et al., 1987), be it of equal-angle (stereographic) or equal-area (Schmidt-Lambert) (Fig. 3). The first has its main use in crystallography, where it is necessary to correctly represent the angles between crystal faces without distortion. However, for visualization of the distribution of attitudes only the second one lends itself adequately. While an equal-angle projection will preserve the shapes, since circles on the sphere will remain as circles on the plane of projection, the density of points around the sphere will be excessively distorted, as showed in the case of a regular grid on Fig. 4.

Considering the point $v = x, y, z$ belonging to the sphere, its coordinates X and Y on the equal angle projection plane are given by Eq. 21. For equal area projection Eq. 22 is applied.

$$X = \frac{x}{1-z}, Y = \frac{y}{1-z} \tag{21}$$

$$X = x\sqrt{\frac{2}{1-z}}, Y = y\sqrt{\frac{2}{1-z}} \tag{22}$$

3. Development

AUTTITUDE was originally developed in Python series 2.7.x (Python Software Foundation, 2013), latter made compatible with series 3.x. Many of the used modules come from the Python Standard Library, such as regular expressions and CSV parsing. Besides that, xlrd (Python Software Foundation, 2016) is used for reading data from Excel spreadsheets and Numpy (Oliphant, 2006) provides support for vectors, matrices and linear algebra.

The use of Numpy greatly facilitates development and speeds up processing, as its numerical routines are written in C and Fortran and highly optimized. This alleviates many problems inherent to interpreted languages such as Python, allowing for fast processing of large datasets. Some of AUTTITUDE's routines use functions that came originally from Scipy, a scientific numerical data processing library that is a superset of Numpy.

Fortran routines were written in Fortran 90, and compiled and linked as Python modules using the script f2py (Peterson, 2009), that automatically wraps the given code and allows it to be called seamlessly from Python.

The final result was a Python library that can be used both for application development and as a platform for directional data analysis on a interactive environment. The source code of the developed packages and the data files used for testing are available at Section 5.

4. Application of AUTTITUDE

The center piece of the library is the *DirectionalData* class. It serves as a container for directional data and provides a series of automatically calculated statistical parameters for both circular and spherical data. It can be used by providing direction cosines as data along with any other additional parameters available, which when needed will be used in its submodules.

The first step in producing an object of the *DirectionalData* class is to load the input data. For this it is either possible to use the *universal_loader*, that handles file loading, *universal_translator*, to convert attitude notations, and finally

pass the data obtained to the *DirectionalData* class, or use the load function, which will call each of these components in sequence.

Universal_loader is responsible for automatically loading the data, dealing with different file formats. In this release it is able to handle CSV, Numpy and Excel data types. From this, the *universal_translator* translates the data from their notation to dip direction/dip format, as explained in Section 2.1. The file in Ex. 1, frat.dat, shows a sample of the variety of notations that can be found. This sequence of methods allows for greater flexibility in loading the data, using, for example, other external functions among the steps. The *load* function follows this basic sequence automatically, converting the data into direction cosines and loading them into an object of the *DirectionalData* class, which is returned by it. It works transparently to the user, using default values or trying to extract them from the data as shown in Ex. 1.

The last parameter called in the example, *mode*, which contains the counting node with highest value, depends on a counting grid. By default, an axial count mesh is created with average spacing between nodes of 2.5 degrees and K is estimated using the method in Robin and Jowett (1986). Since grid analysis can be time-consuming, the counting grid is not created by *DirectionalData* until it is needed, storing the obtained result.

The counting grid is generically an object of the *SphericalGrid* class, which internally constructs the regular grid and has methods for averaging by Fisher Distribution (Robin and Jowett, 1986) or counting points within a threshold angle. Additionally, it is possible to operate it with any function whose parameters are a grid where it will be calculated and data to do the calculation.

Fortran versions of the couting methods and functions to automatically parallelize counting where also developed. They are not used by default because the additional time spent preparing for parallel processing or importing the Fortran modules is longer than the speedup in processing time for the typical amount of data in geological problems. They become more interesting for very high resolution grids, such as an average separation of one second of arc or very large datasets.

The following examples are available as jupyter notebooks on the project github page (Section 5).

Ex. 1. `>>>entry_file= universal_loader("frat.dat")`

```
>>>entry_data = universal_translator(entry_file, dip\direction=False)
>>>frat = DirectionalData(dcos(entry_data))
>>>b = auttitude.load(b.csv)
>>>data = load("tocher.txt")
```

From a *DirectionalData* a series of statistical parameters are available:

Ex. 2. `>>>data.fisher_k`

```
2.0994328692611806
>>>data.eigenvalue[0]
109.57337905751763
>>>data.vollmer_C
0.79271596847405035
>>>print data
tocher.txt
n = 200
Expected Distribution:
Girdle
Eigenvectors:
1: 204.5 / 1.0
2: 295.4 / 42.2
3: 113.4 / 47.8
Shape parameter
K = 0.21
Strength parameter
C = 2.07
Normalized Eigenvalues:
S1: 0.548
S2: 0.383
```

```
S3: 0.069
Fabric (triangular diag.):
Point = 0.165
Girdle = 0.628
Random = 0.207
>>>spherical(data.mode)
(16.6, 3.8)
```

Some operations with *DirectionalData* objects are also possible, such as concatenate, which combines two *DirectionalData*, and the product, which calculates all intersections (*intersect*) between two data sets (Fig. 5, Ex. 3). The *DirectionalData* resulting from these operations inherit the additional parameters from the first *DirectionalData*.

Ex. 3. (see Fig. 5)

```
>>>a = load("a.xlsx")
>>>concatenate(a, b)
<__main__.DirectionalData at 0xa828080>
>>>d = intersect(a, b)
```

Finally, data can be projected onto another projection system (*project*) or be rotated (*rotate*) by a user-defined axis and angle (Fig. 6). The resulting *DirectionalData* also inherits additional parameters from the original data.

Ex. 4. (see Fig. 6)

```
>>>project(data, (x_line, y_line, z_line))
<__main__.DirectionalData at 0xa828668>
>>>data = load("tocher.txt")
>>>rotated_data = rotate(data, axis, angle)
```

Considering its use at interactive environments auxiliary classes where included that use the matplotlib library (Hunter, 2007), allowing its visualization (Ex.5).

Ex. 5. (see Fig. 7)

```
>>> import matplotlib.pyplot as plt

>>> fig, ax = plt.subplots()
>>> projection = EqualAreaProjection(ax)
>>> projection.plot\_poles(data, "bo")
>>> plt.show()
```

It is also possible to obtain only the coordinates of the projected data, instead of directly plotting, using the *equal_angle* and *equal_area* functions.

5. Conclusions

The creation of the AUTTITUDE library allowed the restructuring of the OpenStereo directional data analysis software. Additionally, it can be used for the treatment of directional data both directly and integrated in other applications. Its integration with OpenStereo brings advantages to both software, since the use of AUTTITUDE in a graphical interface facilitates the visualization and handling of complex data sets while at the same time it brings to OpenStereo speed, robustness and ease of implementation and maintenance. The tests performed had positive results.

The library uses a permissive MIT license, and is available at the project github page. Suggestions, adaptations and contributions are very welcome, as are reports on any issues or bugs.

Supplementary Material

Supplementary data related to this article can be found at https://git.io/vMjmu.

References

Bingham, C., 1964. Distributions on the sphere and on the projective plane. Ph.D. thesis. Yale University.

Bingham, C., 1974. An antipodally symmetric distribution on the sphere. The Annals of Statistics , 1201–1225.

Bingham, C., Mardia, K., 1978. A small circle distribution on the sphere. Biometrika 65, 379–389.

Diggle, P.J., Fisher, N.I., 1985. SPHERE: A contouring program for spherical data. Computers & Geosciences 11, 725–766. URL: https://doi.org/10.1016%2F0098-3004%2885%2990015-9, doi:10.1016/0098-3004(85)90015-9.

Fisher, N.I., 1993. Statistical Analysis of Circular Data. Cambridge University Press (CUP). URL: http://dx.doi.org/10.1017/cbo9780511564345, doi:10.1017/cbo9780511564345.

Fisher, N.I., Lewis, T., Embleton, B.J.J., 1987. Statistical analysis of spherical data. Cambridge University Press (CUP). URL: http://dx.doi.org/10.1017/cbo9780511623059, doi:10.1017/cbo9780511623059.

Fisher, R., 1953. Dispersion on a sphere, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, The Royal Society. pp. 295–305.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. Computing In Science & Engineering 9, 90–95. doi:10.1109/MCSE.2007.55.

Jupp, P.E., Mardia, K.V., 1989. A unified view of the theory of directional statistics, 1975-1988. International Statistical Review / Revue Internationale de Statistique 57, 261. URL: http://dx.doi.org/10.2307/1403799, doi:10.2307/1403799.

Kamb, W.B., 1959. Ice petrofabric observations from blue glacier, washington, in relation to theory and experiment. J. Geophys. Res. 64, 1891–1909. URL: http://dx.doi.org/10.1029/jz064i011p01891, doi:10.1029/jz064i011p01891.

Kent, J.T., 1982. The fisher-bingham distribution on the sphere. Journal of the Royal Statistical Society. Series B (Methodological) , 71–80.

Kim, J.M., 2005. Vectorial formulation of direction cosines for anisotropic geologic structures from their geologic angle measurements. Mathematical Geology 37, 929–941. URL: http://dx.doi.org/10.1007/s11004-005-9224-0, doi:10.1007/s11004-005-9224-0.

Kleene, S.C., 1951. Representation of events in nerve nets and finite automata. Technical Report. DTIC Document.

Mardia, K., Gadsden, R., 1977. A small circle of best fit for spherical data and areas of vulcanism. Applied Statistics , 238–245.

Mardia, K.V., Jupp, P.E. (Eds.), 1999. Directional Statistics. John Wiley & Sons, Inc. URL: https://doi.org/10.1002%2F9780470316979, doi:10.1002/9780470316979.

von Mises, R., 1918. Über die ganzzahligkeit der atomgewichte und verwandte fragen. Phys. z 19, 490–500.

Munro, M.A., Blenkinsop, T.G., 2012. MARD — a moving average rose diagram application for the geosciences. Computers & Geosciences 49, 112–120.

Oliphant, T., 2006. Guide to NumPy. Trelgol Publishing. URL: http://www.tramy.us/.

Pearson, K., 1901. LIII. on lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6 2, 559–572. URL: http://dx.doi.org/10.1080/14786440109462720, doi:10.1080/14786440109462720.

Peterson, P., 2009. F2py: a tool for connecting fortran and python programs. International Journal of Computational Science and Engineering 4, 296. URL: https://doi.org/10.1504%2Fijcse.2009.029165, doi:10.1504/ijcse.2009.029165.

Pilgrim, M., 2009. Dive Into Python. Createspace.

Python Software Foundation, 2013. Python Programming Language, version 2.7. Available at http://www.python.org/, last access 15/Aug/2013.

Python Software Foundation, 2016. Python Package Index : xlrd 1.0.0. Available at https://pypi.python.org/pypi/xlrd, last access 25/jan/2017.

Robin, P.Y.F., Jowett, E.C., 1986. Computerized density contouring and statistical evaluation of orientation data using counting circles and continuous weighting functions. Tectonophysics 121, 207–223. URL: http://dx.doi.org/10.1016/0040-1951(86)90044-2, doi:10.1016/0040-1951(86)90044-2.

Schmidt, W., 1917. Statistische methoden beim gefgestudium kristalliner schiefer. Sitzungsberichte, Mathematisch-Naturwissenschaftliche Klasse 126, 515–538.

Vollmer, F.W., 1990. An application of eigenvalue methods to structural domain analysis. Geological Society of America Bulletin 102, 786–791.

Watson, G.S., 1965. Equatorial distributions on a sphere. Biometrika 52, 193. URL: http://dx.doi.org/10.2307/2333824, doi:10.2307/2333824.

Wells, N.A., 2000. Are there better alternatives to standard rose diagrams? Journal of Sedimentary Research 70, 37–46. URL: http://jsedres.geoscienceworld.org/content/70/1/37, doi:10.1306/2DC408FC-0E47-11D7-8643000102C1865D, arXiv:http://jsedres.geoscienceworld.org/content/70/1/37.full.pdf.

Woodcock, N., 1977. Specification of fabric shapes using an eigenvalue method. Geological Society of America Bulletin 88, 1231–1236.

Figure captions

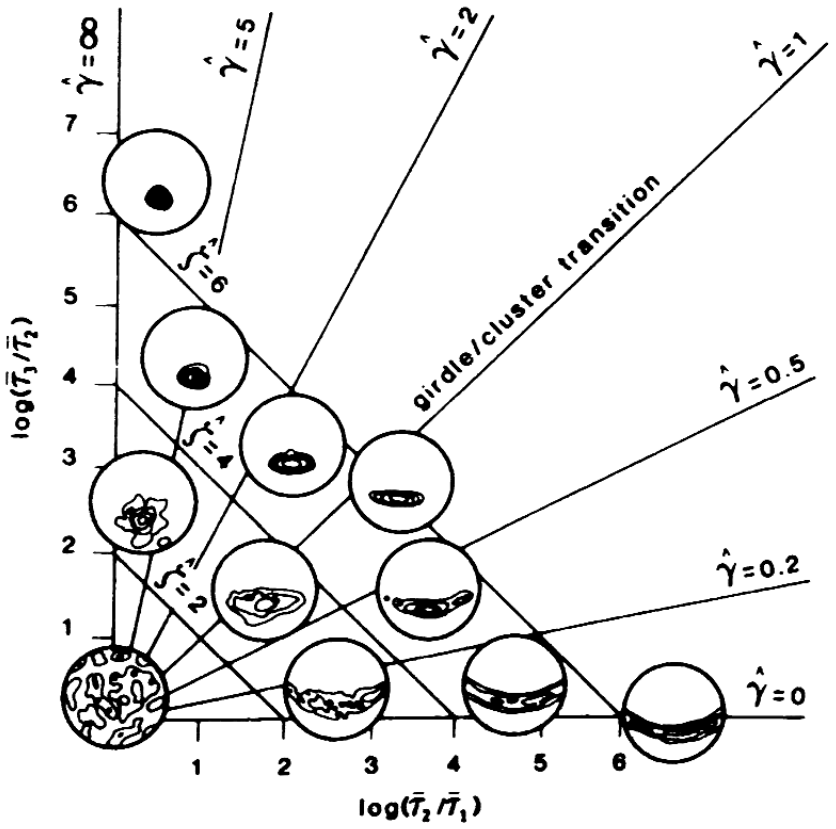

Figure 1: Adapted Flinn diagram ((Woodcock, 1977). Extracted from Fisher et al. (1987)

.

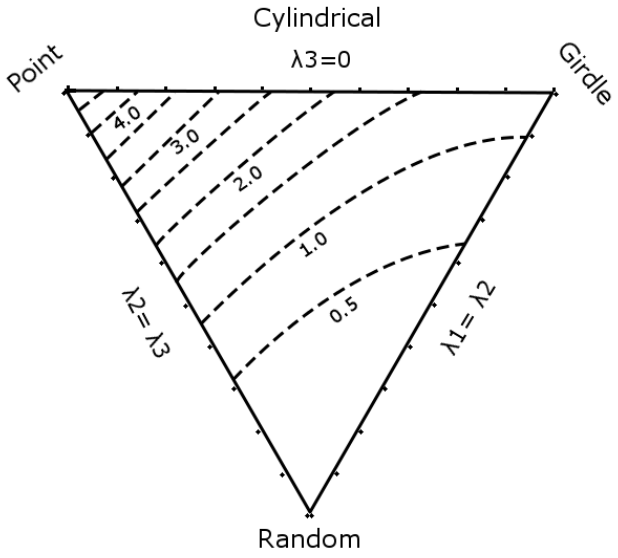

Figure 2: Triangular diagram for shape classification. Extracted from Vollmer (1990)

.

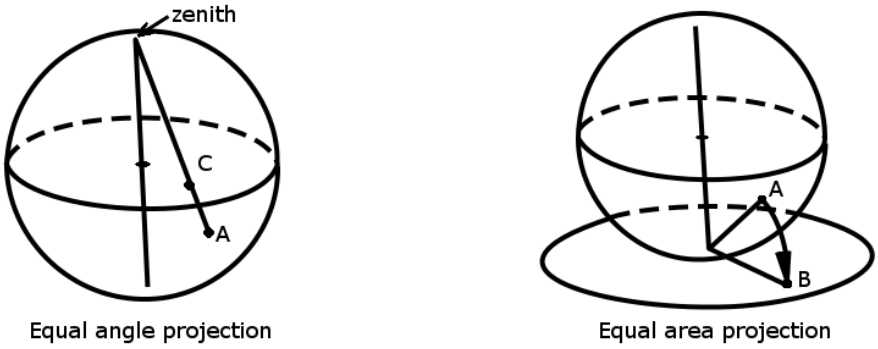

Figure 3: Representation of equal angle and equal area projections, showing the relationship between the points on the sphere and the ones projected on the plane. Extracted from Fisher et al. (1987)

.

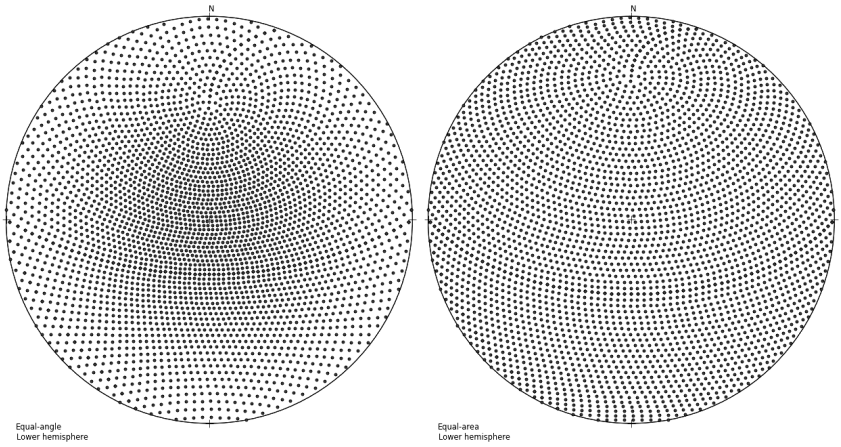

Figure 4: Representation of equal angle (left) and equal area (right) projections for the same 2.5 degrees regular spaced grid, rotated 60 degrees to north. Notice that the latitude circles where the regular grid nodes are distributed remain as circles in the first case, although their regular distribution is only observed in the equal area projection.

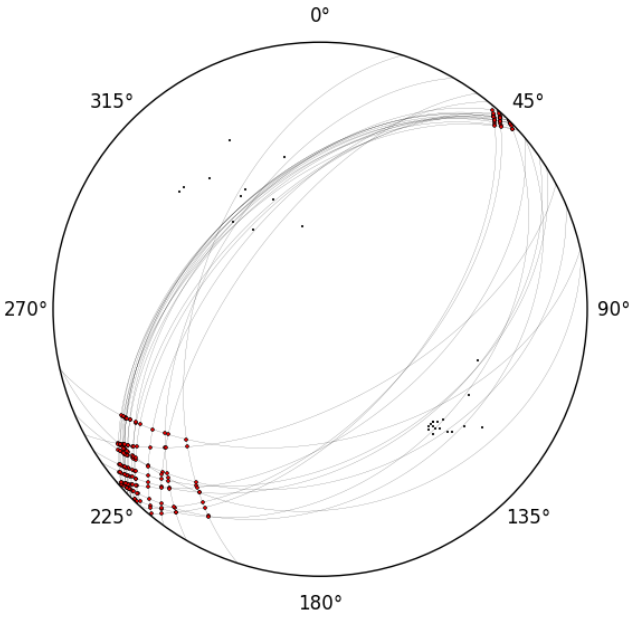

Figure 5: Result of the intersection of data presented in Section 5, with entry data and their maximum circles.

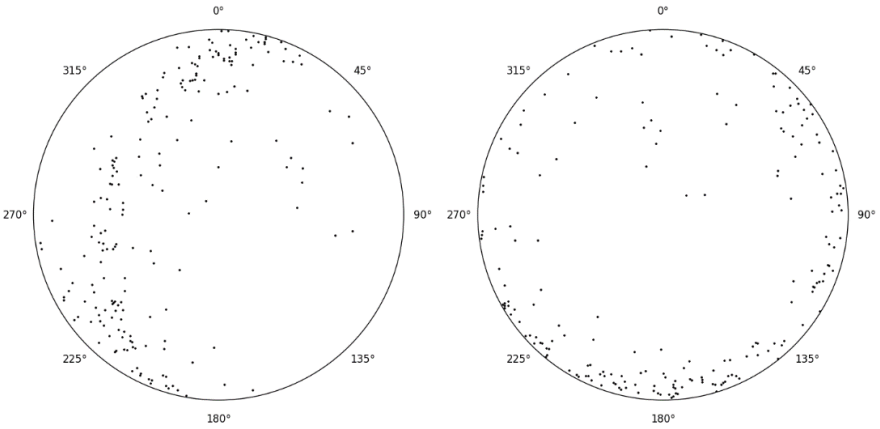

Figure 6: Examples of rotation using the axis 204/01 with 42.2 degrees. Data used for rotation are available at Section 5.

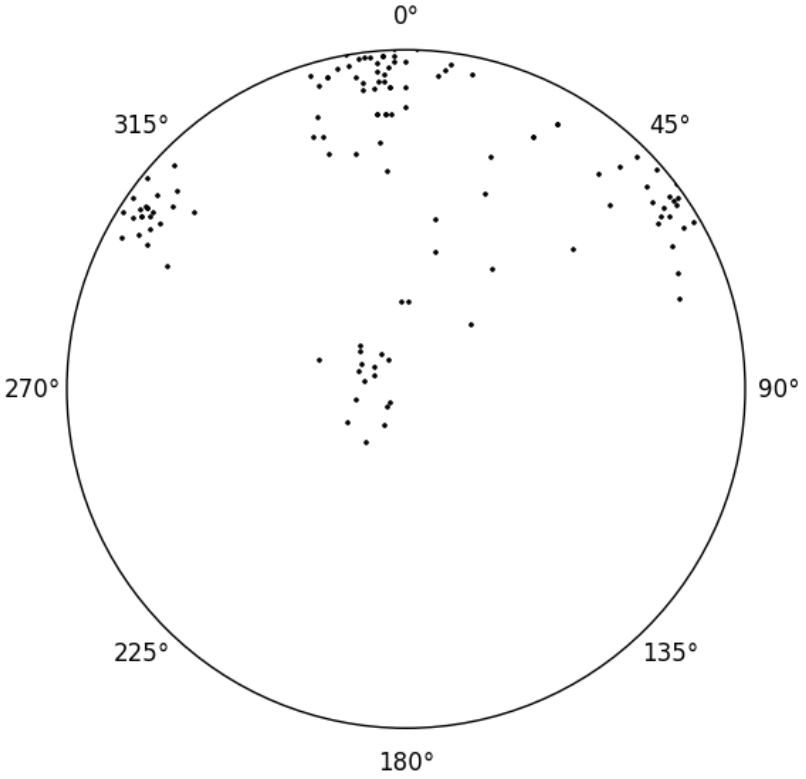

Figure 7: Examples of rotation using the axis 204/01 with 42.2 degrees. Data used for rotation are available at Section 5.

Table captions

Example	[NSEW]{0, 2}	\d*	[NSEW]{0, 2}	[NSEW]{0, 2}	Type
N30E/50NW	Y	Y	Y	Y	Direction, dip, quadrant
140/50	N	Y	N	N	Right hand rule
140/50NE	N	Y	N	N	Direction, dip, quadrant
NW/50	Y	N	N	N	Right hand rule
NS/50E	Y	N	Y	Y	Approximated direction, dip, quadrant

Table 1: Truth table of regular expression for determination of geological attitude notation. Y indicates group presence and N indicates group absence.

Eigenvalues relative magnitude	Distribution	Other characteristics
$\lambda_1 \simeq \lambda_2 \simeq \lambda_3$	Uniform	
λ_1 **big,** λ_2, λ_3 **small**		
(I) $\lambda_2 \neq \lambda_3$	Unimodal if $\bar{R} \simeq 1$, bimodal otherwise	Concentrated at a λ_1 end
(II) $\lambda_2 \simeq \lambda_3$	Unimodal if $\bar{R} \simeq 1$, bipolar otherwise	Rotational symmetry around λ_1
λ_3 **small,** λ_1, λ_2 **big**		
(I) $\lambda_1 \neq \lambda_2$	Girdle	Concentrated in a big circle on plane λ_1, λ_2
(II) $\lambda_1 \simeq \lambda_2$	Symmetric girdle	Rotational symmetry around λ_3

Table 2: Descriptive interpretation of spherical distribution shapes relative to eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of $\bar{T}$ and the resulting length $\bar{R}$. Extracted from Mardia and Jupp (1999).

Chapter 4

Graphical and Numerical Methods for Small Circle Data Fitting

Graphical and Numerical Methods for Small Circle Data Fitting

Arthur Endlein[a,*], Ginaldo Ademar da Cruz Campanha[a], Camila Duelis Viana[a]

[a]*Institute of Geosciences, University of São Paulo, São Paulo, 05508-080, Brazil*

Abstract

A small circle is a set of vectors on a sphere that have a constant angular distance to a vector. They arise on many different situations on geological data and yet most popular free available directional data analysis applications either do not offer the functionality of small circle fitting or provide methods which may struggle to estimate the best fit cone when dealing with complex distributions.This work both reviews a numerical method and proposes a new graphical method. To test the methods, sample datasets for small circles were generated. Python implementations of the methods used are available at Appendix B.

Keywords: Small circles, Conical Folds, Directional Data Fitting.

1. Introduction

A small circle is a set of vectors $x_i = \{x_1, x_2...x_n\}$ on a sphere that have a constant angular distance α to a vector μ. They arise on many different situations on geological data such as planar attitudes measured on certain superimposed folding systems (Ramsay, 1962), conical folds at tappering ends of cylindrical folding systems (Cruden and Charlesworth, 1972), paleomagnetic data directions restoration (Pueyo et al., 2003; Mochales et al., 2016), linear

*Corresponding author

Email addresses: arthur.correia@usp.br; endarthur@gmail.com (Arthur Endlein), ginaldo@usp.br (Ginaldo Ademar da Cruz Campanha), camila.viana@usp.br (Camila Duelis Viana)

attitudes measured on folding systems that condition them to a constant angular displacement from the fold axis or segments of hotspot traces (Mardia and Gadsden, 1977) and transform faults. As noticed by Mulchrone et al. (2013), the most popular free available directional data analysis applications either do not offer the functionality of small circle fitting or provide methods which may struggle to estimate the best fit cone when dealing with complex distributions. So far the axes for these small circles have been adjusted either by visual inspection using π-diagrams or by iterative numerical methods (Ramsay, 1962; Mardia and Gadsden, 1977; Bingham and Mardia, 1978; Mulchrone et al., 2013). Until present, there is no reliable graphical method for estimating best fit small circles.

This work both uses the numerical method detailed on Bingham and Mardia (1978) and proposes a new graphical method, that reduces the problem to an analog of the π or β diagrams for great circles (Ramsay, 1967).

2. Method

A small circle with axis μ and half-apical angle α can be defined either as the intersection of a unit sphere and a cone with axis μ and half-apical angle α or a plane with normal vector μ and distance to origin equal to $sin(\alpha)$.

Being $x_i = \{x_1, x_2...x_n\}$ column vectors that represents data points along a small circle, we define the average vector $\bar{x}$, orientation matrix S and covariance matrix $\tilde{S}$ as

$$\bar{x} = \sum_i \frac{x_i}{n} \tag{1}$$

$$S = \frac{1}{n-1} \sum_i x_i \cdot x_i' \tag{2}$$

$$\tilde{S} = \frac{1}{n-1} \sum_i (x_i - \bar{x}) \cdot (x_i - \bar{x})' \tag{3}$$

Bingham and Mardia (1978) proposes that the eigenvector associated with the smallest eigenvalue of $\tilde{S}$ as a good estimator for the small circle's axis, and the *arcsin* of the dot product between the estimated axis and the data's average vector $\bar{x}$ as an estimator for the half-apical angle.

This follows from the fact that the eigenvector associated with the smallest eigenvalue is parallel to the direction of most flattening of the data, and as such a good estimate of the normal vector of a plane (Pearson, 1901). Watson (1966) provides a physical interpretation for the smallest and largest eigenvalues and related eigenvectors of matrix S as the rotation axes through the origin with respectively maximum and minimum moments of inertia, considering a system with particles of unit weight at the end of each vector. A similar analogy can be made for matrix $\tilde{S}$, with the difference that the axes are through the center of mass of the system, instead of the origin.

2.1. Correction for Axial Data

Given the typical representation of axial data as vectors on the lower hemisphere, the previous method may not work, as it strongly depends on the correct estimation of the average vector $\bar{x}$. For datasets that contains points that cross the primitive (Fig. 1), the direct calculation of the average vector usually results in a subvertical vector with low magnitude, even though the data may be highly concentrated in a preferential direction.

(figure 1 around here)

As a correction, it is possible to concentrate the data around either the largest or the smallest eigenvector of matrix S, depending on the semi-apical angle of the small circle. Considering the components of a data point as $x_i = \{l_i, m_i, n_i\}$ and combining with the definition of matrix S results in

$$S = \frac{1}{n-1}\sum_i x_i \cdot x_i' = \frac{1}{n-1}\sum_i \begin{bmatrix} l_i l_i & l_i m_i & l_i n_i \\ m_i l_i & m_i m_i & m_i n_i \\ n_i l_i & n_i m_i & n_i n_i \end{bmatrix} \tag{4}$$

As during the calculation of Eq. 4 each cell consist on the product of two

components, multiplying any data points by -1 will result in the same matrix. Hence, both matrix S and its eigenvectors and eigenvalues are insensitive to the inversion of points. The eigenvector v_1 associated with its largest eigenvalue will point to the axis of largest axial data concentration, as the eingenvector v_3 asssociated with its smallest eigenvalue will point as far as possible from the data points.

By then comparing each vector x_i and its inverse $-x_i$ to both these eigenvectors and in each case keeping whichever is closest, we may approximate an average vector that better represents the real mean axis, allowing the calculation of matrix $\tilde{S}$.

For axial datasets that consist of a concentration along a preferred axis, v_1 is generally a good choice. In the context of small circles, though, concentrating along this eigenvector may distort the circle, if the semi-apical angle is greater than 45 degrees. In this case it is better to concentrate along v_3. As a heuristic, both small circle axes can be calculated, keeping whichever has the smallest associated standard deviation of angles to the data points.

Another possible solution is through visual inspection. By rotating the dataset so it doesn't cross the primitive, the average vector can be calculated accurately and the numerical method will then work in most situations.

2.2. Graphical Method

The covariance matrix can also be defined on the difference of pairs of data (Barnes (1991), adapted proof in Appendix A), through the equation

$$\tilde{S} = \frac{1}{2n(n-1)} \sum_i \sum_j (x_i - x_j) \cdot (x_i - x_j)' \tag{5}$$

From this, an approximate graphical solution to the problem can be realized. Though obtaining the difference vectors graphically would be cumbersome, a unit vector (d) parallel to the difference vector of two points a and b can be easily constructed:

1. Find the great circle that contains both poles a and b. Mark its pole (c) and the midpoint (m) contained in the smallest angle along the great circle between a and b (Fig. 2A).
2. Point d is the pole to the great circle that contains c and m (Fig. 2B)

(figure 2 around here)

The pole to the great circle adjusted to these points is then an estimate of the small circle axis, as an analog to the π diagram (Fig. 2C) of a great circle girdle, as is the intersection of their respective great circles, comparable to the β diagram (Fig. 2D).

Though being a less robust estimate of the axis, given that each vector will be given equal weight regardless of its magnitude, this is a simple method that can produce useful results.

3. Examples

To illustrate the method and challenges involved, we take as example points on circles with horizontal axis due east and semi-apical angles of 30 and 60 degrees, adding noise to both samples (Fig. 3).

(figure 3 around here)

The first case, with angle of 30 degrees, Fig. 3A shows fitted axes for $\bar{x}$, the non-corrected mean vector, with respective small circle dashed, and $\bar{x}'$, the mean vector calculated with data points concentrated around v_1, with solid circle.

Fig. 3B shows the point cloud resultant from the graphical method, applied exhaustively to all possible pairs of points. The best fit great circle girdle to this point cloud is plotted with solid circle.

The second case is meant to demonstrate the issue with large semi-apical angles. In this example (Fig. 4A), given the distance from the points to the axis, concentrating along v_1 distorts the circle and gives a wrong axis (dashed circle). The standard deviation of angles of the solution calculated using v_3 (solid circle) is then compared to the former, and v_3 is kept as the answer.

(figure 4 around here)

In this case, the graphical method result in a more dispersed point cloud, when applied to all possible combinations of the dataset points and their inverses (Fig. 4B). Either through visual inspection, when applied computationally, or careful selection of points pairs when manually, a girdle may then be fitted to the resulting points.

4. Conclusions

This work reviews and implement a numerical non-iterative method for small circle data fitting, both robust and fast, that may be added to directional data analysis software. It also proposes a graphical solution, that both allows for data fitting on paper but also may be implemented numerically and used as input for visual inspection and fitting of complex datasets.

We have also explored a potential issue with large semi-apical angles, and studied a method that in many cases corrects these problems. Further developments may include non-linear optimization methods, such as simulated annealing, as to search for the best combination of inversion of points for the fitting, avoiding distortions.

The numerical method has been implemented in the new version of OpenStereo (Grohmann and Campanha, 2010), allowing for both the automatic or manual correction methods for datasets that cross the primitive.

References

Barnes, R.J., 1991. The variogram sill and the sample variance. Mathematical Geology 23, 673–678. URL: http://dx.doi.org/10.1007/BF02065813, doi:10.1007/BF02065813.

Bingham, C., Mardia, K.V., 1978. A small circle distribution on the sphere. Biometrika 65, 379. URL: http://dx.doi.org/10.2307/2335218, doi:10.2307/2335218.

Cruden, D.M., Charlesworth, H., 1972. Observations on the numerical determination of axes of cylindrical and conical folds. GSA Bulletin 83, 2019–2024. URL: http://bulletin.geoscienceworld.org/content/83/7/2019, doi:10.1130/0016-7606(1972)83[2019:OOTNDO]2.0.CO;2, arXiv:http://bulletin.geoscienceworld.org/content/83/7/2019.full.pdf.

Grohmann, C.H., Campanha, G., 2010. Openstereo: open source, cross-platform software for structural geology analysis, in: AGU Fall Meeting abstracts.

Mardia, K.V., Gadsden, R.J., 1977. A small circle of best fit for spherical data and areas of vulcanism. Applied Statistics 26, 238. URL: http://dx.doi.org/10.2307/2346963, doi:10.2307/2346963.

Mochales, T., Pueyo, E., Casas, A., Barnolas, A., 2016. Restoring paleomagnetic data in complex superposed folding settings: The boltaña anticline (southern pyrenees). Tectonophysics 671, 281–298. URL: http://dx.doi.org/10.1016/j.tecto.2016.01.008, doi:10.1016/j.tecto.2016.01.008.

Mulchrone, K.F., Pastor-Galán, D., Gutiérrez-Alonso, G., 2013. Mathematica code for least-squares cone fitting and equal-area stereonet representation. Computers & Geosciences 54, 203–210. URL: http://dx.doi.org/10.1016/j.cageo.2013.01.005, doi:10.1016/j.cageo.2013.01.005.

Pearson, K., 1901. LIII. on lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6 2, 559–572. URL: http://dx.doi.org/10.1080/14786440109462720, doi:10.1080/14786440109462720.

Pueyo, E., Parés, J., Millán, H., Pocoví, A., 2003. Conical folds and apparent rotations in paleomagnetism (a case study in the southern pyrenees). Tectonophysics 362, 345–366. URL: http://dx.doi.org/10.1016/s0040-1951(02)00645-5, doi:10.1016/s0040-1951(02)00645-5.

Ramsay, J., 1967. Folding and fracturing of rocks. International series in the earth and planetary sciences, McGraw-Hill.

165 Ramsay, J.G., 1962. Interference patterns produced by the superposition of folds of similar type. The Journal of Geology 70, 466–481. URL: http://dx.doi.org/10.1086/626837, doi:10.1086/626837.

Watson, G.S., 1966. The statistics of orientation data. The Journal of Geology , 786–797.

Figure captions

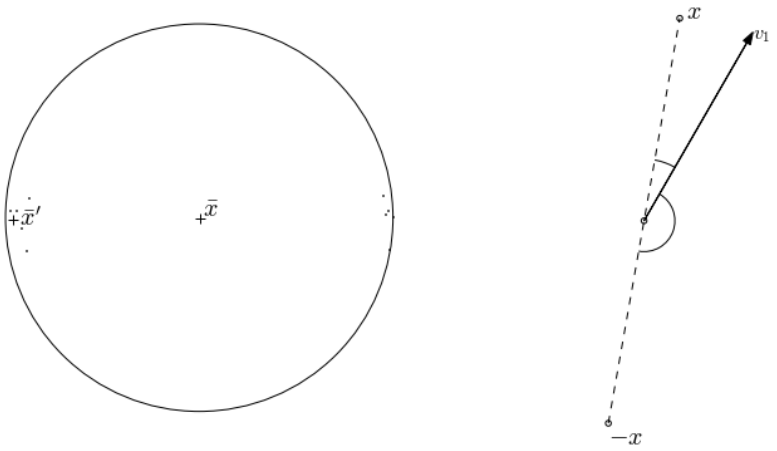

Figure 1: On the left, example of data distribution that cross the primitive. The direct calculation of the average vector ($\bar{x}$) usually results in subvertical vector with low magnitude. Concentrating the data around the first eigenvector results in a average vector($\bar{x}'$) closer to the data concentration. On the right, the comparison of angles between a given vector v_1 and a data point x and its inverse $-x$.

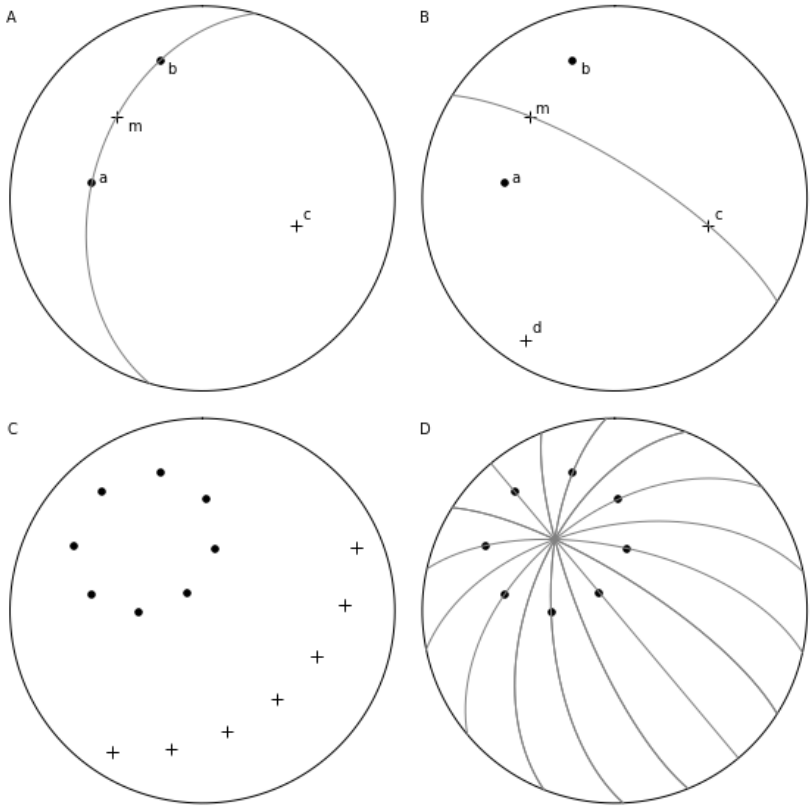

Figure 2: Representation of the construction of a unit vector d parallel to the difference vector of vectors of a and b on small circle using the graphical method: (A) Find the great circle that contains both poles a and b, mark its pole c and the midpoint m contained in the smallest angle along the great circle between a and b; (B) Point d is the pole to the great circle that contains c and m; (C) The pole to the great circle adjusted to these points is then an estimate of the small circle axis, as an analog to the π diagram of a great circle girdle, as is the intersection of their respective great circles, comparable to the β diagram (D)

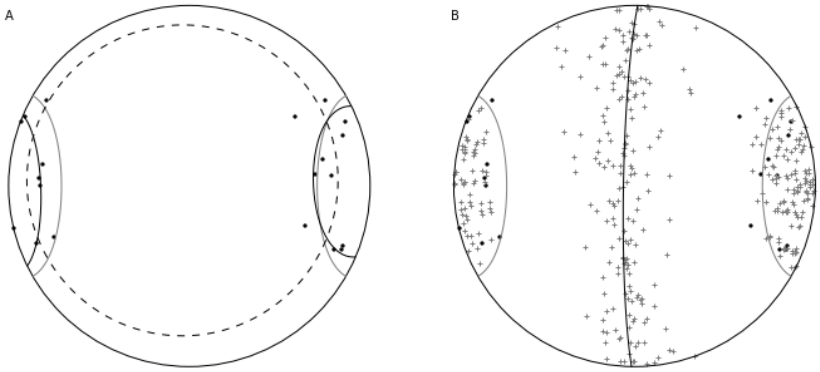

Figure 3: For a 30 degrees semi-apical angle: (A) the numerical method shows fitted axes for $\bar{x}$ (non-corrected mean vector) with respective small circle dashed and $\bar{x}'$ (mean vector with points around v_1) with solid circle graphical; (B) the point cloud resultant from the graphical method applied exhaustively to all pairs of points shows its best fit great circle girdle as a solid line circle.

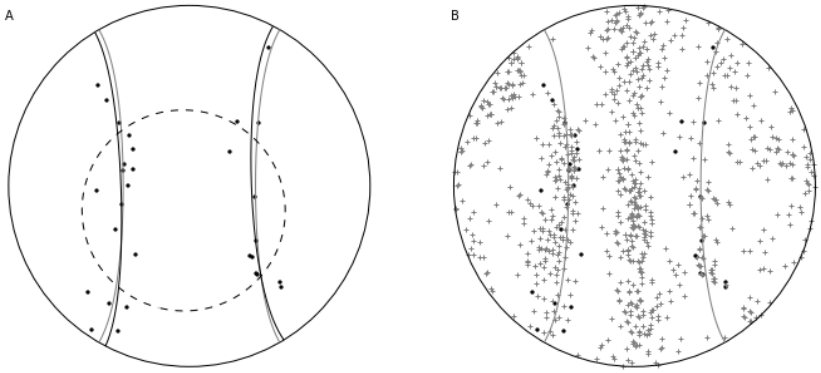

Figure 4: For a 60 degrees semi-apical angle: (A) the numerical method shows that given the distance from the points to the axis concentrating along v_1 distorts the circle and gives a wrong axis (dashed line). The angular variance of the solution calculated using v_3 (solid line) is then compared to the former and v_3 is kept as the answer; (B) the point cloud resultant from the graphical method is more dispersed.

Appendix A. Covariance matrix on differences

This proof is a redevelopment of Barnes (1991), adapted to vectors to produce a covariance matrix. Starting from the equation

$$D = \frac{1}{2n(n-1)} \sum_i \sum_j (x_i - x_j) \cdot (x_i - x_j)' \tag{A.1}$$

Adding and subtracting the average vector $\bar{x}$ from both sides of the dot product and distributing the dot product and sums we arrive in

$$D = \frac{1}{2n(n-1)} \sum_i \sum_j ((x_i - \bar{x}) - (x_j - \bar{x})) \cdot ((x_i - \bar{x}) - (x_j - \bar{x}))' \tag{A.2}$$

$$\begin{aligned} \tilde{S} = & \frac{1}{2n(n-1)} \sum_i \sum_j (x_i - \bar{x}) \cdot (x_i - \bar{x})' - \\ & \frac{1}{2n(n-1)} \sum_i \sum_j 2(x_j - \bar{x}) \cdot (x_i - \bar{x})' + \\ & \frac{1}{2n(n-1)} \sum_i \sum_j (x_j - \bar{x}) \cdot (x_j - \bar{x})' \end{aligned} \tag{A.3}$$

The first component of equation A.3 can be simplified to $\frac{\tilde{S}}{2}$. As the terms inside the parenthesis doesn't depend on j, they can be factored out of the summation:

$$\begin{aligned} & \frac{1}{2n(n-1)} \sum_i \sum_j (x_i - \bar{x}) \cdot (x_i - \bar{x})' \\ & \frac{1}{2(n-1)} \sum_i (x_i - \bar{x}) \cdot (x_i - \bar{x})' \sum_j \frac{1}{n} \\ & \frac{1}{2(n-1)} \sum_i (x_i - \bar{x}) \cdot (x_i - \bar{x})' = \frac{\tilde{S}}{2} \end{aligned}$$

The third component can be simplified in a similar way, by factoring the terms inside the parenthesis out of the summation over i.

The second component may then be reduced to the zero tensor. Factoring out of the sum over i the first term of the dot product results in a sum of

the differences between each vector and their mean vector, resulting in the null vector.

$$\frac{1}{2n(n-1)}\sum_i\sum_j 2(x_j-\bar{x})\cdot(x_i-\bar{x})'$$
$$\frac{1}{2n(n-1)}\sum_j 2(x_j-\bar{x})\cdot\sum_i(x_i-\bar{x})'$$
$$\frac{1}{2n(n-1)}\sum_j 2(x_j-\bar{x})\cdot(\sum_i x_i-\sum_i\sum_k\frac{x_k}{n})'$$
$$\frac{1}{2n(n-1)}\sum_j 2(x_j-\bar{x})\cdot(\sum_i x_i-\sum_k x_k\sum_i\frac{1}{n})'$$
$$\frac{1}{2n(n-1)}\sum_j 2(x_j-\bar{x})\cdot(\sum_i x_i-\sum_k x_k)'$$
$$\frac{1}{2n(n-1)}\sum_j 2(x_j-\bar{x})\cdot(x_1-x_1+x_2-x_2\ldots+x_n-x_n)'$$
$$\frac{1}{2n(n-1)}\sum_j 2(x_j-\bar{x})\cdot 0'=0$$

Finally, combining the previous results, we arrive in

$$D=\frac{\tilde{S}}{2}-0+\frac{\tilde{S}}{2}=\tilde{S} \tag{A.4}$$

Showing that the covariance matrix can be obtained from the differences between vectors.

Appendix B. Sample Python code for small circle axis

```
import numpy as np

def small_circle(x, n=None):
    n = x.shape[0]
    #find the largest eigenvector of matrix S
    S = x.dot(x.T)/(n-1)
    #if n is given, concentrate data around the Nth eigenvector
```

```
    if n is not None:
      eig, eiv = np.linalg.eig(S)
      eig_order = np.argsort(eig, reverse=True)[::-1]
      eiv = eiv[:,eig_order]
      n_eiv = eiv[:, n]
      #concentrate the data
      x = x*np.where(x.dot(n_eiv) > 0, 1, -1)
    #calculate the mean and S_ matrix
    x_ = x.mean(axis=0)
    S_ = S - x_.dot(x_.T)
    #calculate the small circle axis and half-apical angle
    eig, eiv = np.linalg.eig(S_)
    s_eig = np.argmin(eig)
    s_eiv = eiv[:, s_eig]
    theta = asin(abs(x_.dot(s_eiv)))
    return s_eiv, theta
```

Chapter 5

OpenStereo 1.0

OpenStereo 1.0

Arthur Endlein[a,*], Carlos Henrique Grohmann[b], Ginaldo Ademar da Cruz Campanha[a], Camila Duelis Viana[a]

[a] *Institute of Geosciences, University of São Paulo, São Paulo, 05508-080, Brazil*
[b] *Institute of Energy and Environment, University of São Paulo, São Paulo, 05508-010, Brazil*

Abstract

OpenStereo was first released in 2010 as an open source, cross-platform software for directional data projection and structural geology. After a full restructuring its 1.0 version is presented as a faster and more user-friendly program which includes new features such as small circle fitting, conversion of shapefiles to azimuth data, conversion of sampled three-dimensional meshes to plane data, continuous weighted rose diagrams, angle measurement tool and project management support. This release also provides more customization options, and was structured as to facilitate further extensions to the software.

Keywords: Spherical projection, Python, Stereonet.

1. Introduction

Many different directional data analysis software have been developed over the decades, both commercial, such as DIPS (Rocscience, 2013) and freeware, as Orient (Frederick W. Vollmer, 2017) and Stereonet (Cardozo and Allmendinger, 2013). OpenStereo was originally built as a free, Open Source and cross-platform alternative, aiming for a clean interface, simple data input and extensibility.

Over the years OpenStereo have acquire a great number of users, being cited in several articles such as the most recent Gehrmann et al. (2016),Moreno-

*Corresponding author
Email addresses: arthur.correia@usp.br (Arthur Endlein), guano@usp.br (Carlos Henrique Grohmann), ginaldo@usp.br (Ginaldo Ademar da Cruz Campanha), camila.viana@usp.br; camila.duelis@gmail.com (Camila Duelis Viana)

Sánchez et al. (2016) and Phillips et al. (2016). From its first release, the authors set a development path that would focus on tools as merging and rotation of datasets, possibility to save projects and paleostress analysis.

Focusing on the same objectives, this work is the result of a restructuring of the original software in two parts. First the creation of a directional data analysis library, AUTTITUDE, that encapsulates data input, pre-processing and numerical analysis (Endlein et al., a). From this framework it was possible to concentrate the development of OpenStereo on a better user interface, offering additional plot options, analysis methods and data management capabilities.

Both existing tools were expanded, such as the added option to calculate continuous rose diagrams (Munro and Blenkinsop, 2012), the inclusion of rotation and angle measurement to the projection plot, and new tools were added, as projects, extracting planar attitudes from three-dimensional models and azimuths and lengths from shapefiles.

2. Functionality

OpenStereo tries to simplify as much as possible loading data sets. In this release it supports loading data from Excel Spreadsheets and text files, either in CSV or GeoEAS formats. If one of the specific import menus is used, such as *Import Planar Data (Dip Direction)* the software will handle the file format, detect if it has a header and make a best guess on which columns should be used, defaulting to the first for dip direction and the second for dip. In case there is any need of a finer control on the data loading, it is possible to open the import dialog (Fig. 1) and change any settings on the import process.

(figure 1 around here)

OpenStereo's plotting area is organized in tabs (Fig. 2): *Projection*, showing spherical data projected in either Equal-Area or Equal-Angle projections, *Rose Diagram*, with either binned or moving average counts of circular data and *Classification*, which includes both Woodcock (1977) and Vollmer (1990)

methods of distribution shape classification.

(figure 2 around here)

The Projection tab generically allows the plotting of vectors as points, circles as lines and counting grids as contours, whenever adequate to each data set type. It also allows the rotation of the whole plot and the inclusion of grids with user defined spacing of great and small circles (Fig. 3). The plot is also interactive. Inspired by Stereo32 (Röller and Trepmann, 2008) it is possible to measure angles between points on the projection by click and drag.

(figure 3 around here)

The *Rose Diagram* tab serves for plotting circular counts, either as petals, kites or lines (Fig. 4). The scale of the plot can be either set by the user or obtained automatically from the given data. The underlying grid can also be configured, both the spacing of the grid lines and its span (Fig. 5).

(figure 4 around here)

(figure 5 around here)

The *classification* tab allows choosing between Woodcock's adapted Flinn diagram or Vollmer's triangular plot (Fig. 6). The points that are plotted on the diagrams use the same settings as the vectors for the *Projection* tab.

(figure 6 around here)

A major change relative to the original OpenStereo was the inclusion of projects. OpenStereo project files are zip compressed folders with extension "*.openstereo*". They include a *json* (http://www.json.org/) file containing the data sets list and information relative to the project itself, such as title, authorship, description and specific parameters of the plotting tabs. For each data set another *json* file containing its plotting information is kept. These plot settings can also be exported and imported individually as "*.os_lyr*" files.

If desired by the user, the data files themselves are included inside the project

file, being then decompressed on the fly for loading. If not, the data file paths relative to the project file are saved, allowing the user to easily work on multiple computers.

2.1. Data Types and Properties

The software currently supports loading planes, lines, small circles and circular data. After loading, the data set will be included in the items tree, showing as options its plot elements (Fig. 7). With the exception of "Rose", these refer to plotting on the *Projection* tab.

(figure 7 around here)

Each supported data type has its own properties window, allowing the definition of the plot options of each possible element. The *Properties* window (Fig. 8) of a planar dataset can be taken as a comprehensive example, being the other data types effectively subsets of this one.

(figure 8 around here)

The *Projection* tab houses options for plotting projections of vectors and circles, being them original from the input data, as poles to planes or great circles or derived, as poles and circles of the dataset eigenvectors, average vector and best fit small circle.

OpenStereo offer two ways to calculate the average vector. By default, the vector will be calculated directly from the direction cosines of the dataset. Alternatively the data can be first oriented so it occupies the hemisphere defined by the first eigenvector, which may give better results for sub-horizontal datasets that cross the primitive (Endlein et al., a).

The best fit small circle is calculated as suggested by Bingham and Mardia (1978), with options for re-orientating the vectors to the hemisphere of one of the eigenvectors, as it heavily depends on a good estimate of the average vector (Endlein et al., b). This should be set to the first eigenvector for circles with semi-apical angle of less than 45 degrees, otherwise to the third eigenvector.

Another option for troublesome datasets is to rotate the projection until the circle does not cross the primitive. This can be done on the *Projection* tab of the settings dialog, as explained above.

The second tab, *Contours* (Fig. 9), defines options for the calculation and graphical display of counting grids. The grid may either be used to count points that fall within a given θ angle to each node (Kamb, 1959), or sum for each node a smoothing function of its angle to each point, controlled by a factor K (Robin and Jowett, 1986). Both θ and K may be defined by the user or calculated automatically by two different methods. The first one, from Robin and Jowett (1986), depends only on the number of points while the second (Diggle and Fisher, 1985) performs an optimization based on the cross validation of the smoothing function.

(figure 9 around here)

Plotting circular data and horizontal components of spherical data is controlled by the *Rose* tab on the properties dialog (Fig. 10). Here, it is possible to define whether dip directions or directions are going to be used and if these should be considered vectors, with direction and sense, or axes, with only direction (Fisher, 1993). The actual counting can be made as a standard rose diagram, effectively a circular histogram, or as a moving average, weighted or not (Munro and Blenkinsop, 2012). The results may be plotted as petals, kites or lines, the last from either the center of the diagram or as deviations from the average frequency (as suggested in Wells (2000)).

(figure 10 around here)

The last three tabs, *Statistics*, *General* and *Data Source* contain respectively the statistics for the dataset calculated automatically by auttitude, legend options for the plot elements and information on the import process of the data. This last may either be used for solving problems on the data import or used by OpenStereo for reloading the data, as lines can be added or edited in the original source.

2.2. Mesh analysis

The emergence of methods for easily building three-dimensional digital models, either with terrestrial laser scanning or digital photogrammetry, have opened a new venue for geological attitudes measurement. Ply2atti is a tool developed in a previous work that uses a combination of graph analysis and linear algebra to calculate the attitudes of selected planar features on a digital model of an outcrop (Viana et al., 2016). It has been integrated in this version of OpenStereo as a tool.

An external software must be used to digitally paint the surfaces to be measured on the three-dimensional model, for example Meshlab (Cignoni et al., 2008), an open source model visualization and editing software. It is important to use colors that do not appear naturally in the model, such as pure basic colors, and as many planes as needed can be painted with the same color, as long as they do not share a vertex. The model should then be exported in binary Stanford Polygon format (*.PLY*), and the colors used noted (Fig. 11).

(figure 11 around here)

When opening the file for conversion in OpenStereo a dialog will appear where the colors used for painting the model can be input. By default, red (255, 0, 0), green (0, 255, 0) and blue (0, 0, 255) will be suggested, but other colors may be used (Fig. 12).

(figure 12 around here)

The internal processing of the dataset consists in building graphs using the vertices of the three-dimensional mesh that have the given colors, splitting the graphs in their connected components to separate different planes, and for each plane calculating the eigenvector associated with the smallest eigenvalue of the covariance matrix of its vertices. This eigenvector is kept as the best fit normal vector to the plane (Pearson, 1901). A CSV file containing the calculated attitudes and some additional parameters is generated for each given color and loaded on OpenStereo, automatically changing the poles to planes color to

match. Additional details on the method and procedure can be found at Viana et al. (2016).

2.3. Shapefile analysis

An experimental tool to convert lines shapefiles to circular data was also included in this release. The pyshp (Ian Bicking and Michael Foord, 2005) library is used to parse the shapefile, which must be in geographical coordinates. Being r the average radius of Earth (6,371km), the length of each segment of each line in the file is calculated by taking the latitudes (ϕ_1 and ϕ_2) and longitudes (λ_1 and λ_2) in radians of their endpoints and applying the haversine formula,

$$\begin{aligned} a =& \sin^2\left(\frac{\phi_2-\phi_1}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right) \\ c =& atan2\left(\sqrt{a},\sqrt{(1-a)}\right) \\ d =& 2rc \end{aligned} \tag{1}$$

Similarly the forward azimuth formula is used to calculate the bearing θ of the segment,

$$\theta = atan2\left(\sin\left(\lambda_2-\lambda_1\right)\cos\phi_2, \cos\phi_1\sin\phi_2 - \sin\phi_1\cos\phi_2\cos\left(\lambda_2-\lambda_1\right)\right) \tag{2}$$

The resulting lengths and azimuths are exported to a CSV file and automatically imported to OpenStereo as a circular dataset (Figs. 13 and 14).

(figure 13 around here)

(figure 14 around here)

Both formulas where adapted from Chris Veness (2016). $atan2$ is a function available on most programming languages that calculates the *arctangent* of the ratio of its two parameters while comparing their signs to determine the correct quadrant of the angle and handling cases in which one of them is equal to zero.

3. Development

OpenStereo was developed in Python series 2.7.x (Python Software Foundation, 2013), though also being compatible with Python Series 3.x. Most calculations are actually performed by the AUTTITUDE library (Endlein et al., a), which also uses the Numpy (Oliphant, 2006) library for vectors and linear algebra and xlrd (Python Software Foundation, 2016) for excel spreadsheets parsing.

Plotting is done using the Matplotlib (Hunter, 2007) library, which provides methods for producing high quality figures and exporting the results in many different formats. An important change from the original implementation of OpenStereo was the switch from Wx (http://www.wxwidgets.org/) to Qt (https://www.qt.io/) as its graphical interface toolkit. This greatly facilitated the development of the interface while also making it possible to internationalize the software in the future. The specific Python implementation of Qt used was the PySide library (https://wiki.qt.io/PySide).

4. Conclusion

This restructuring of OpenStereo greatly expands its capabilities, not only directly, by offering new tools and enhancing the existing ones, but also by focusing from the ground up on making it easy to maintain and improve.

Future developments will include paleostress analysis of faults, kinematic criteria analysis of rock slopes, clustering datasets and expanding interactivity with the plot area, such as allowing direct drawing of new data.

Supplementary Material

Supplementary data related to this article can be found at https://git.io/vMjma.

References

Bingham, C., Mardia, K., 1978. A small circle distribution on the sphere. Biometrika 65, 379–389.

Cardozo, N., Allmendinger, R.W., 2013. Spherical projections with OSXStereonet. Computers & Geosciences 51, 193–205. URL: https://doi.org/10.1016%2Fj.cageo.2012.07.021, doi:10.1016/j.cageo.2012.07.021.

Chris Veness, 2016. Movable Type Scripts. URL: http://www.movable-type.co.uk/scripts/latlong.html.

Cignoni, P., Corsini, M., Ranzuglia, G., 2008. Meshlab: An open-source 3d mesh processing system. ERCIM News 73, 45–46. URL: http://ercim-news.ercim.eu/en73/rd/meshlab-an-open-source-3d-mesh-processing-system.

Diggle, P.J., Fisher, N.I., 1985. SPHERE: A contouring program for spherical data. Computers & Geosciences 11, 725–766. URL: https://doi.org/10.1016%2F0098-3004%2885%2990015-9, doi:10.1016/0098-3004(85)90015-9.

Endlein, A., da Cruz Campanha, G.A., Grohmann, C.H., Viana, C.D., a. AUTTITUDE: object oriented computational model for directional data with emphasis on tectonics and structural geology .

Endlein, A., da Cruz Campanha, G.A., Viana, C.D., b. Graphical and Numerical Methods for Small Circle Data Fitting .

Fisher, N.I., 1993. Statistical Analysis of Circular Data. Cambridge University Press (CUP). URL: http://dx.doi.org/10.1017/cbo9780511564345, doi:10.1017/cbo9780511564345.

Frederick W. Vollmer, 2017. Orient 3.6.2 Spherical Projection and Orientation Data Analysis Software User Manual . URL: http://www.frederickvollmer.com/orient/download/Orient_3.6.2_User_Manual.pdf.

Gehrmann, A., Hneke, H., Meschede, M., Phillips, E., 2016. 3d microstructural architecture of deformed glacigenic sediments associated with large-

scale glacitectonism, jasmund peninsula (ne rgen), germany. Journal of Quaternary Science , n/a–n/aURL: http://dx.doi.org/10.1002/jqs.2843, doi:10.1002/jqs.2843.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. Computing In Science & Engineering 9, 90–95. doi:10.1109/MCSE.2007.55.

Ian Bicking and Michael Foord, 2005. pyshp 0.1.0.

Kamb, W.B., 1959. Ice petrofabric observations from blue glacier, washington, in relation to theory and experiment. J. Geophys. Res. 64, 1891–1909. URL: http://dx.doi.org/10.1029/jz064i011p01891, doi:10.1029/jz064i011p01891.

Moreno-Sánchez, M., Hincapié J, G., Ossa M, C.A., Toro Toro, L.M., 2016. CARACTERIZACIÓN GEOLÓGICO-ESTRUCTURAL DE ALGUNAS ZONAS DE CIZALLA EN EL COMPLEJO QUEBRADAGRANDE EN LOS ALREDEDORES DE MANIZALES Y VILLAMARíA. Boletin de Geología 38, 15 – 27. URL: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-02832016000400001&nrm=iso.

Munro, M.A., Blenkinsop, T.G., 2012. MARD — a moving average rose diagram application for the geosciences. Computers & Geosciences 49, 112–120.

Oliphant, T., 2006. Guide to NumPy. Trelgol Publishing. URL: http://www.tramy.us/.

Pearson, K., 1901. LIII. on lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6 2, 559–572. URL: http://dx.doi.org/10.1080/14786440109462720, doi:10.1080/14786440109462720.

Phillips, M., Haberkorn, A., Draebing, D., Krautblatter, M., Rhyner, H., Kenner, R., 2016. Seasonally intermittent water flow through deep fractures in an alpine rock ridge: Gemsstock, central swiss alps. Cold Regions Science and Technology 125, 117–127. URL: https://doi.org/10.1016%2Fj.coldregions.2016.02.010, doi:10.1016/j.coldregions.2016.02.010.

Python Software Foundation, 2013. Python Programming Language, version 2.7. Available at http://www.python.org/, last access 15/Aug/2013.

Python Software Foundation, 2016. Python Package Index : xlrd 1.0.0. Available at https://pypi.python.org/pypi/xlrd, last access 25/jan/2017.

Robin, P.Y.F., Jowett, E.C., 1986. Computerized density contouring and statistical evaluation of orientation data using counting circles and continuous weighting functions. Tectonophysics 121, 207–223. URL: http://dx.doi.org/10.1016/0040-1951(86)90044-2, doi:10.1016/0040-1951(86)90044-2.

Rocscience, 2013. Dips 6.0.

Röller, K., Trepmann, C., 2008. Stereo32 version 1.01. Ruhr-Universität Bochum, Institut für Geologie, Mineralogie & Geophysik, Germany .

Viana, C.D., Endlein, A., da Cruz Campanha, G.A., Grohmann, C.H., 2016. Algorithms for extraction of structural attitudes from 3d outcrop models. Computers & Geosciences 90, 112–122. URL: https://doi.org/10.1016%2Fj.cageo.2016.02.017, doi:10.1016/j.cageo.2016.02.017.

Vollmer, F.W., 1990. An application of eigenvalue methods to structural domain analysis. Geological Society of America Bulletin 102, 786–791.

Wells, N.A., 2000. Are there better alternatives to standard rose diagrams? Journal of Sedimentary Research 70, 37–46. URL: http://jsedres.geoscienceworld.org/content/70/1/37, doi:10.1306/2DC408FC-0E47-11D7-8643000102C1865D, arXiv:http://jsedres.geoscienceworld.org/content/70/1/37.full.pdf.

Woodcock, N., 1977. Specification of fabric shapes using an eigenvalue method. Geological Society of America Bulletin 88, 1231–1236.

Figure captions

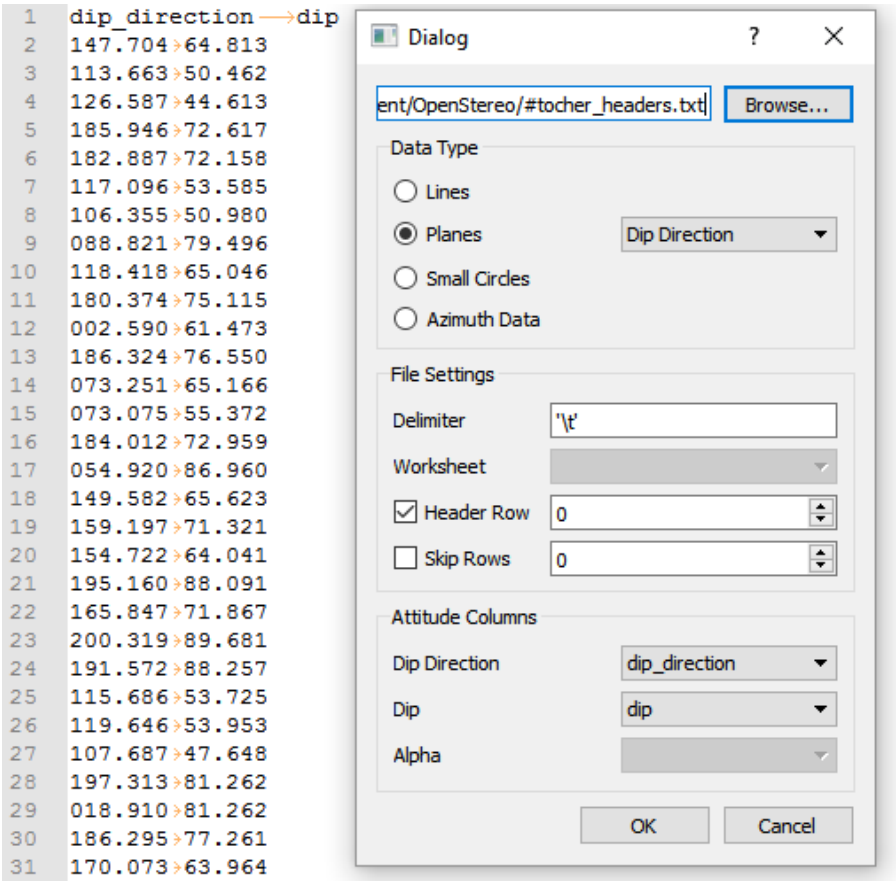

Figure 1: Image of the import dialog, demonstrating the automatic detection of delimiter and headers.

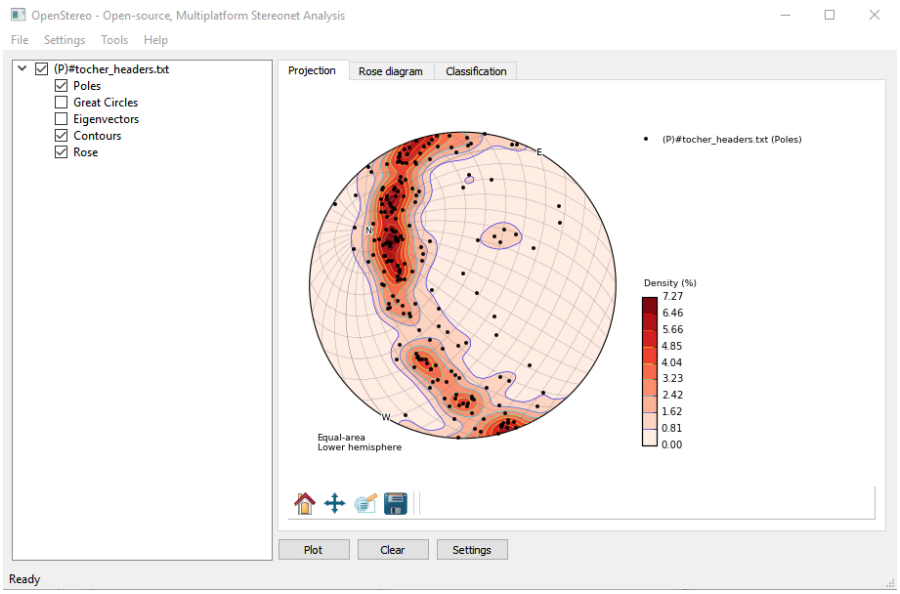

Figure 2: Image of the main window on the Project tab, showing a rotated contoured dataset with grid.

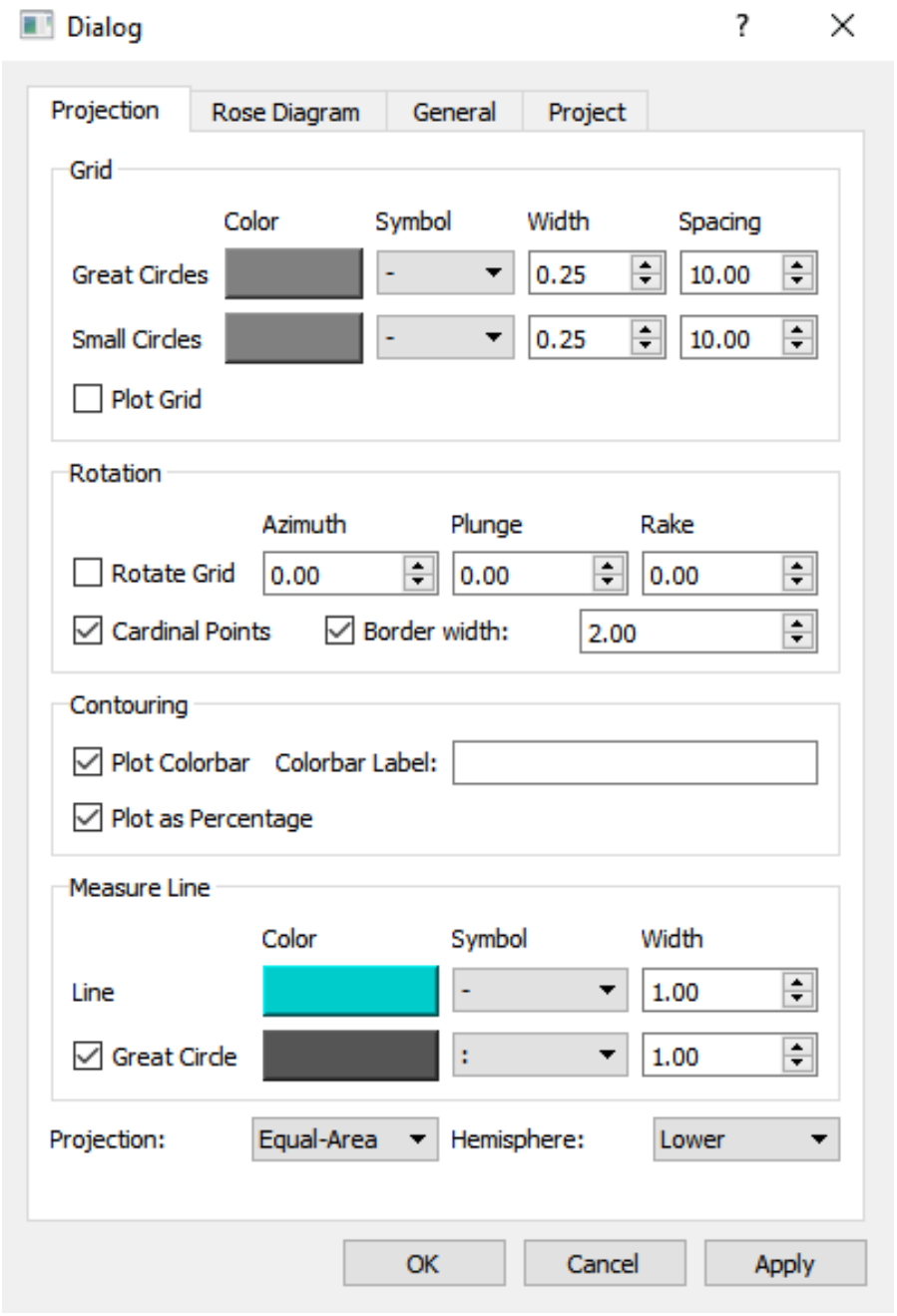

Figure 3: Image of the settings dialog for the Projection tab, including setting the projection, hemisphere, rotation of the plot and grid overlay.

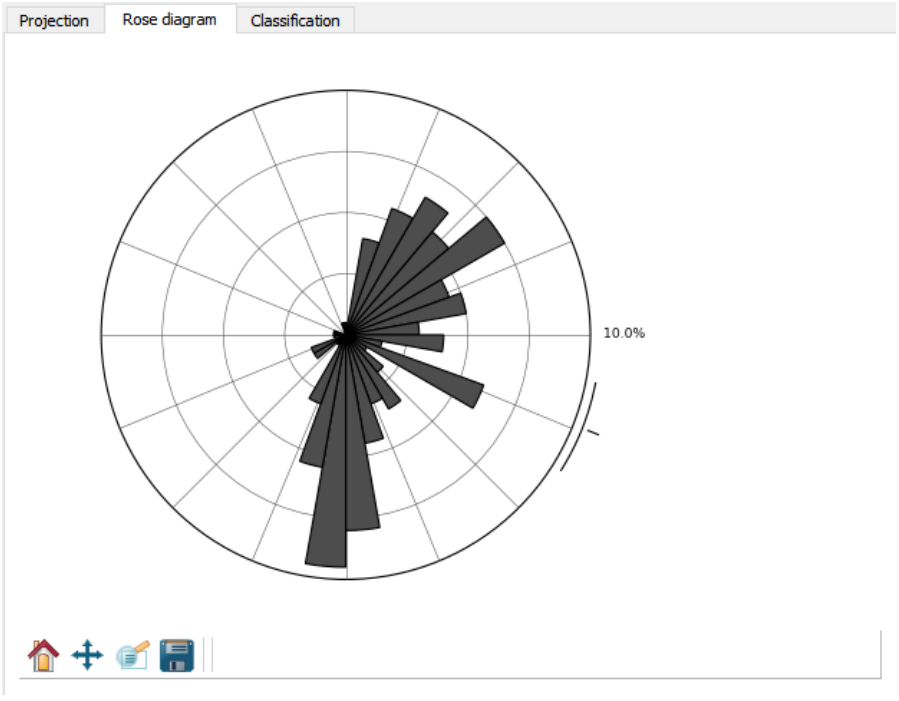

Figure 4: Image of Rose Diagram tab, showing a standard rose diagram for an example dataset.

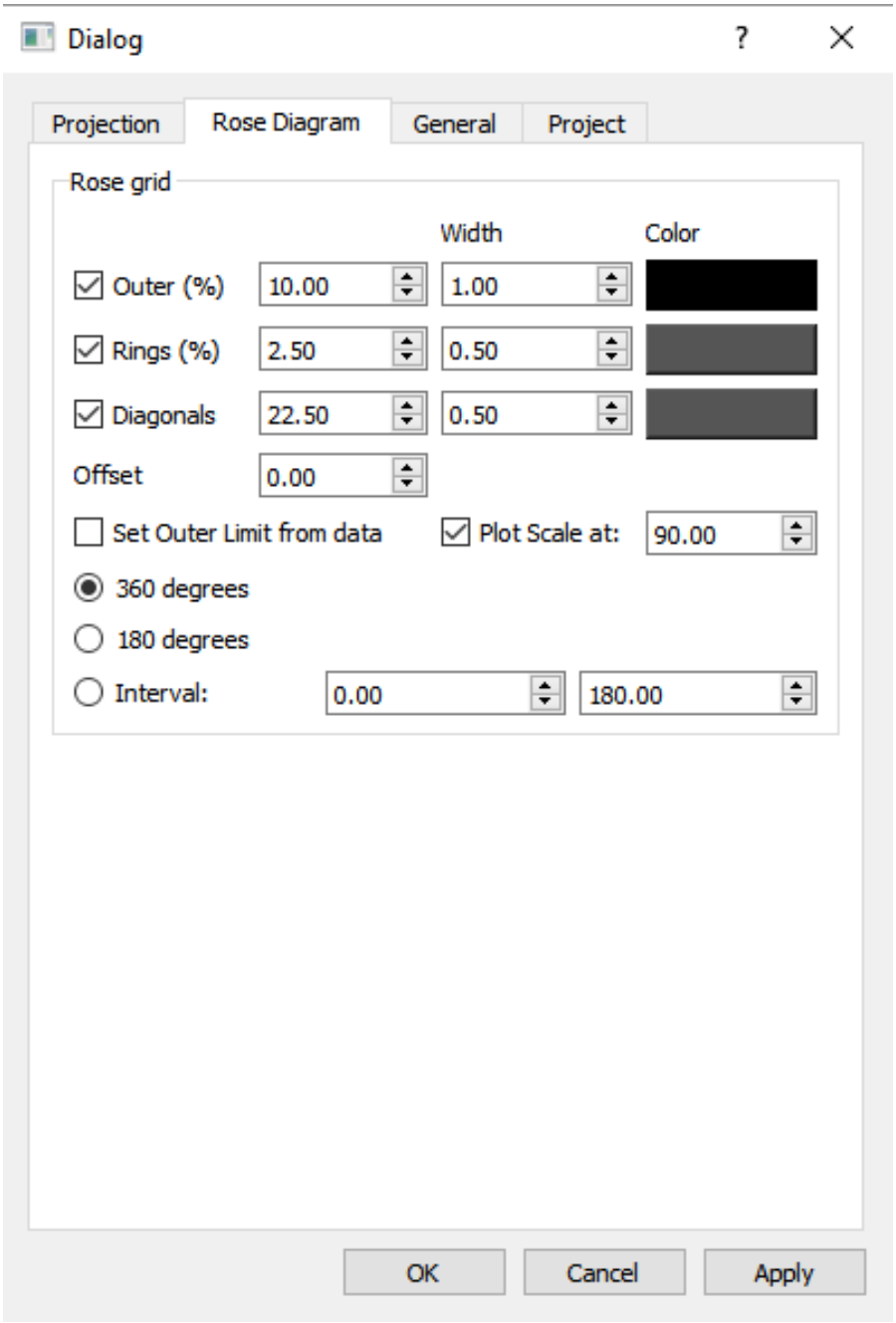

Figure 5: Image of the settings dialog for the Rose Diagram tab, including plot scale and grid options.

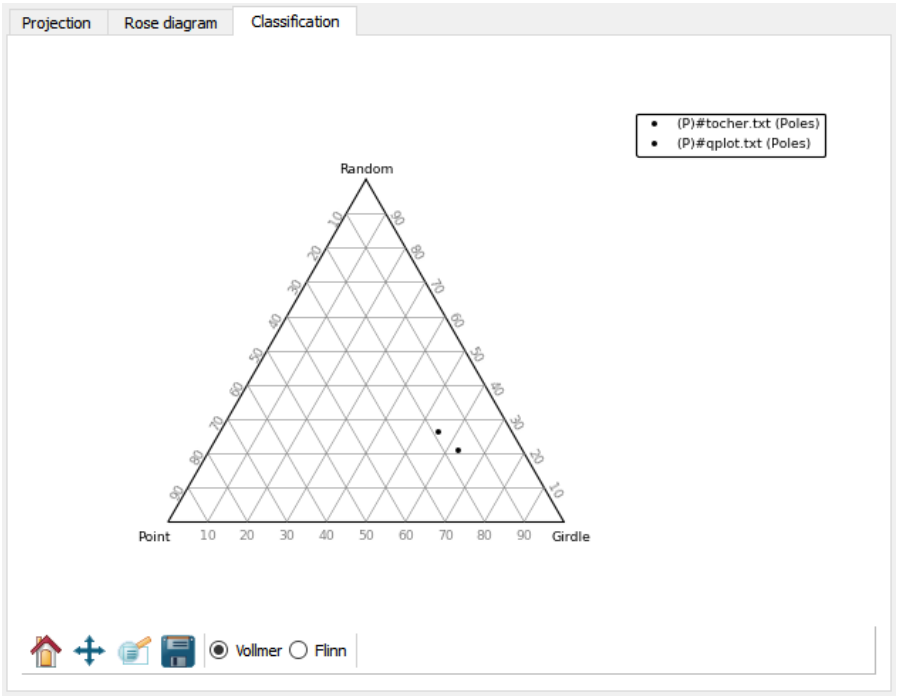

Figure 6: Image of the Classification tab, showing a Vollmer Triangular plot.

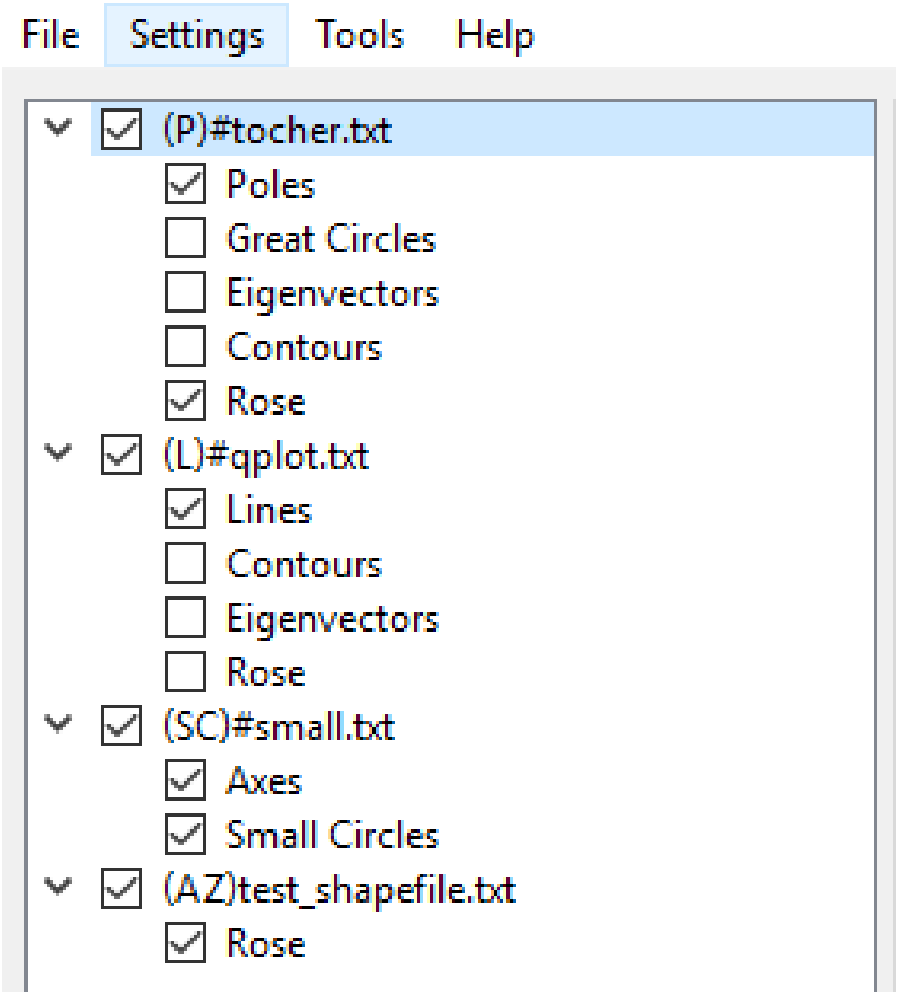

Figure 7: Example of datasets loaded on the item tree, showing the currently supported data types.

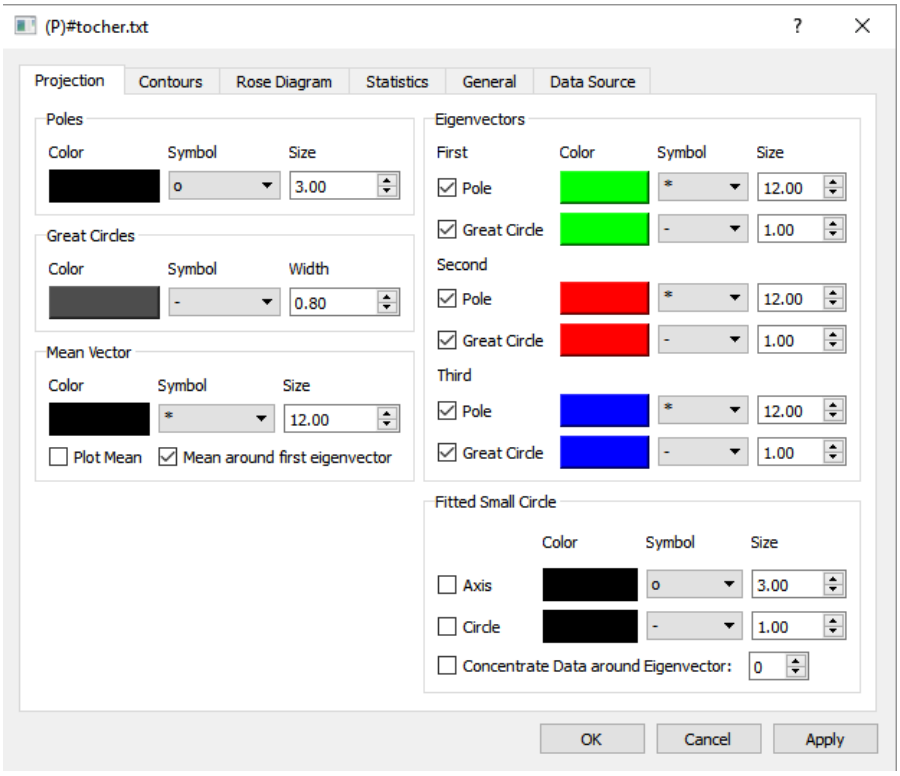

Figure 8: Projection tab on properties window for planar dataset, showing options for graphical display of elements and calculation of average vector and small circle fit.

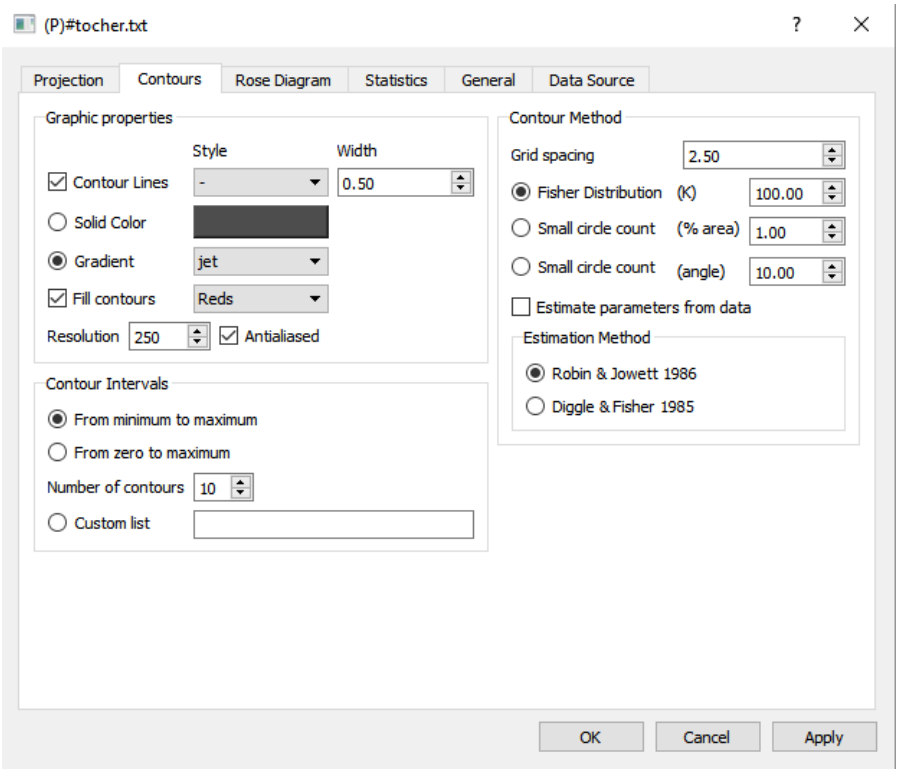

Figure 9: Contours tab on properties window for planar dataset, showing options for graphical display, calculation and parameters estimation.

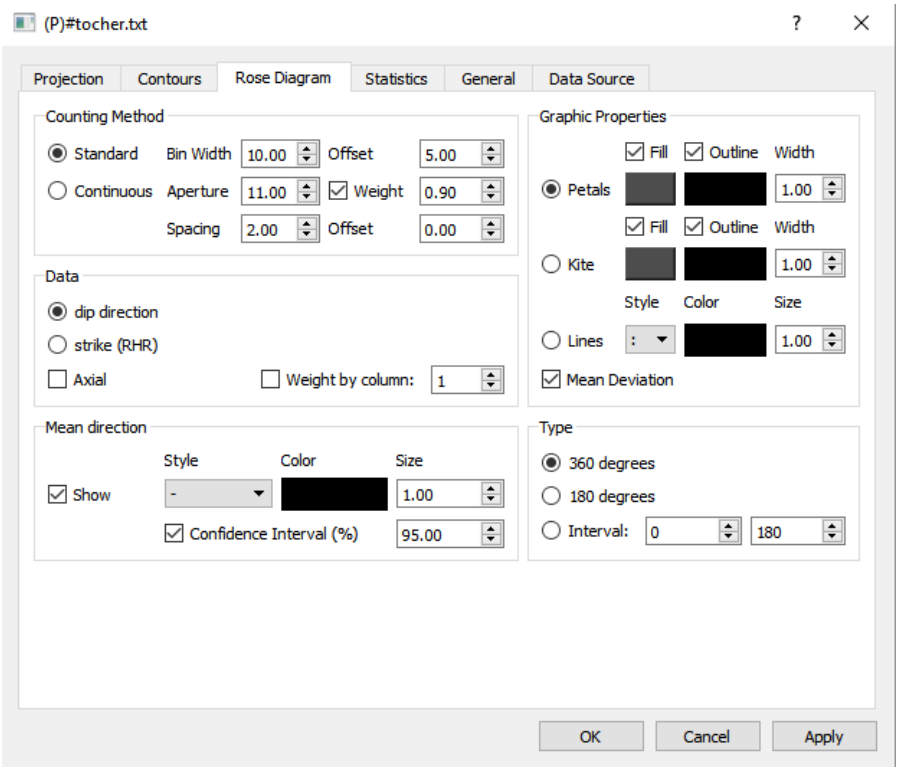

Figure 10: Rose tab on properties window for planar dataset, showing options for graphical display and calculation.

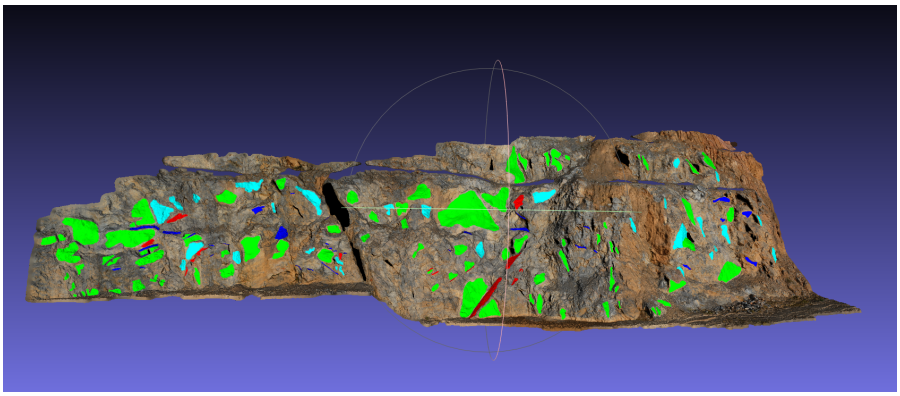

Figure 11: Example of three-dimensional outcrop model with planes digitally painted for extraction of planar attitudes. Original data from Viana et al. (2016).

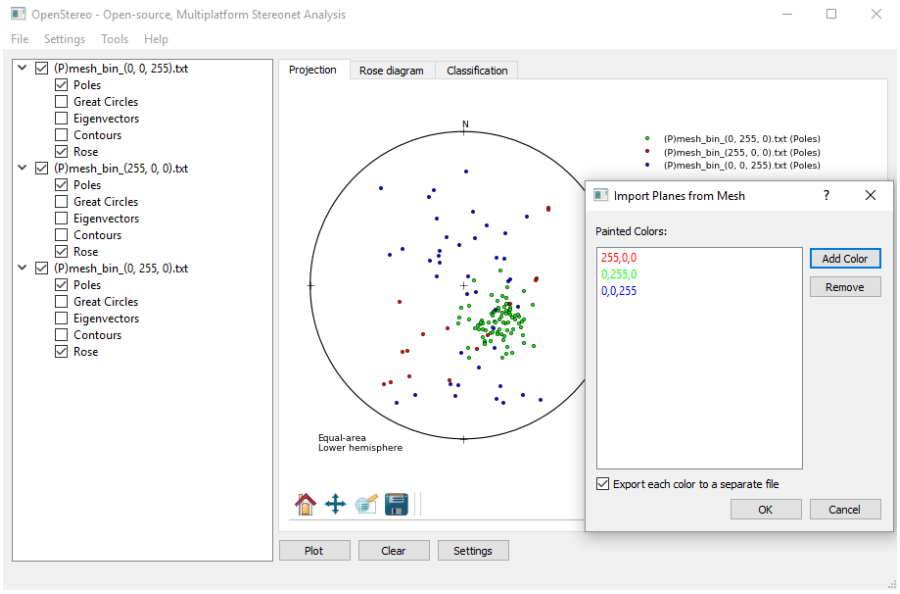

Figure 12: Example of datasets extracted using ply2atti loaded on openstereo. Original data from Viana et al. (2016).

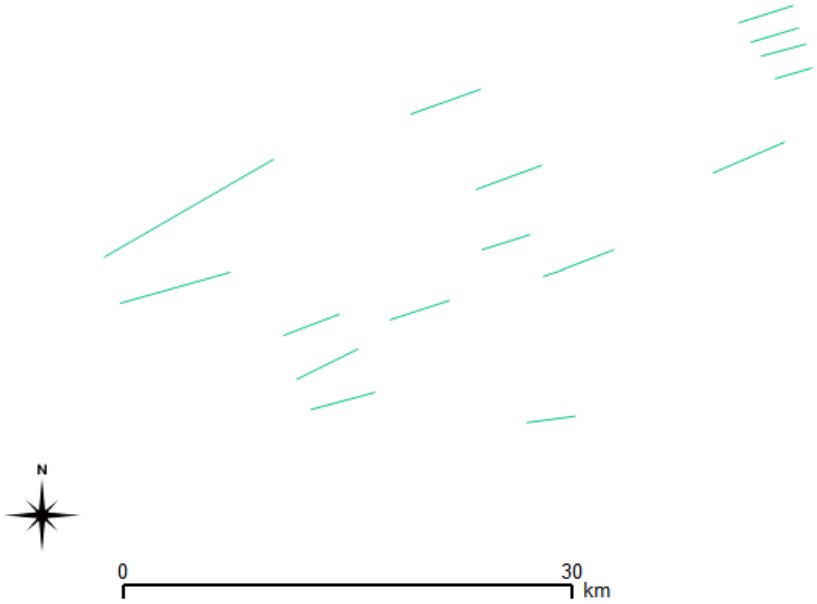

Figure 13: Example of a lines shapefile to be processed on OpenStereo.

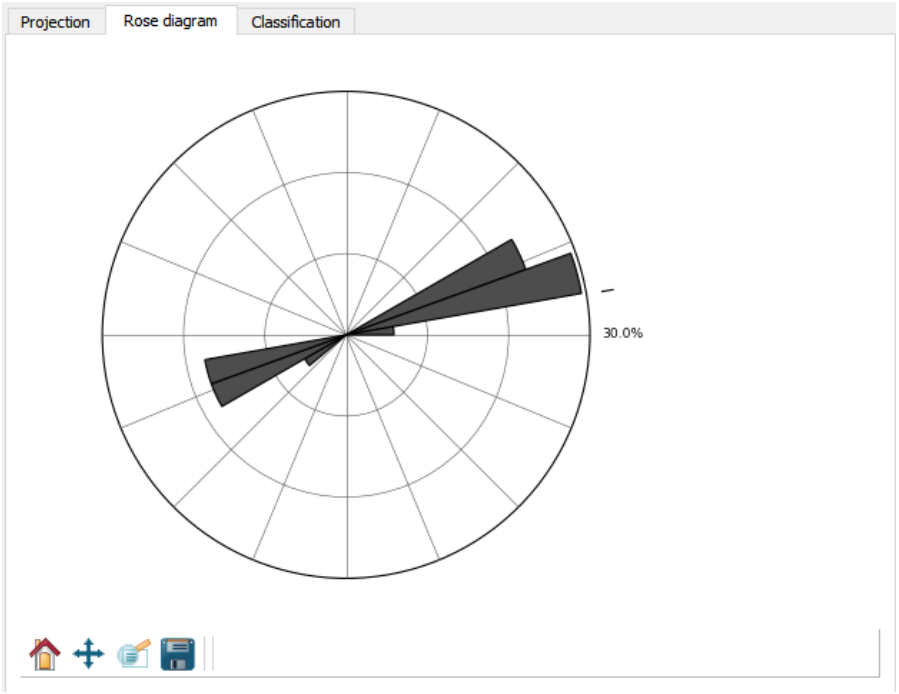

Figure 14: Rose plot of the example shapefile.

Chapter 6

Conclusions

The restructuring of OpenStereo was a long but successful process. The resulting software is robust and fast, able to work with many different data sources, while helping users manage their data and hopefully obtain new insights. It is much easier to maintain and extend, providing a strong foundation over which future projects may be developed.The Auttitude library is also an interesting subproduct, should it be used in other new software of on its own.

The research involved also generated a new graphical method for small circle data fitting, that though not perfect in all situations, can and will be further developed and may result in other methods or applications of existing methods.

Future developments over both OpenStereo and Auttitude are planned, such as including new data types, plotting methods and workflows. Being open source projects, contributions by other researchers and users will be invaluable, and it is hoped that this new version will spark the growth of a larger community around OpenStereo.

References

Gehrmann A., Hüneke H., Meschede M., Phillips E., 3D microstructural architecture of deformed glacigenic sediments associated with large-scale glacitectonism, Jasmund Peninsula (NE Rügen), Germany, Journal of Quaternary Science, 2016, pp n/a–n/a

Grohmann C. H., Campanha G., OpenStereo: open source, cross-platform software for structural geology analysis. In AGU Fall Meeting abstracts , 2010

Grohmann C. H., Campanha G. A., Soares Junior A., OpenStereo: um programa Livre e multiplataforma para análise de dados estruturais, XIII Simpósio Nacional de Estudos Tectônicos, 2011

Hagberg A., Swart P., Schult D., Exploring network structure, dynamics, and function using NetworkX. In Proceedings of the 7th Python in Science Conferences (SciPy 2008) , vol. 2008, 2008, p. 11

Hunter J. D., Matplotlib: A 2D graphics environment, Computing In Science & Engineering, 2007, vol. 9, p. 90

Ian Bicking and Michael Foord, 2005 pyshp 0.1.0

Moreno-Sánchez M., Hincapié J G., Ossa M C. A., Toro Toro L. M., CARACTERIZACIÓN GEOLÓGICO-ESTRUCTURAL DE ALGUNAS ZONAS DE CIZALLA EN EL COMPLEJO QUEBRADAGRANDE EN LOS ALREDEDORES DE MANIZALES Y VILLAMARíA, Boletin de Geología, 2016, vol. 38, p. 15

Oliphant T., Guide to NumPy. Trelgol Publishing, 2006

Phillips M., Haberkorn A., Draebing D., Krautblatter M., Rhyner H., Kenner R., Seasonally intermittent water flow through deep fractures in an Alpine Rock Ridge: Gemsstock, Central Swiss Alps, Cold Regions Science and Technology, 2016, vol. 125, p. 117

Python Software Foundation, 2016 Python Package Index : xlrd 1.0.0

Viana C. D., Endlein A., da Cruz Campanha G. A., Grohmann C. H., Algorithms for extraction of structural attitudes from 3D outcrop models, Computers & Geosciences, 2016, vol. 90, p. 112

Appendix