PYRAMP – A platform for resource assessment and mine planning in Jupyter

N. Morales, E. Jélvez, G. Morales, F. Soto, G. Díaz, and F. Navarro

University of Chile, Chile

The increasing availability of tools on the web provides a great opportunity for the research community to boost investigation by accessing computational resources from all over the world. For example, data analytics libraries implemented in languages like Python can be accessed through many open and commercial platforms, therefore providing researchers with a common set of tools or shared datasets.

Unfortunately, that is not the case with research in resource assessment and mine planning, where the most standardised tools are commercial applications designed to work on the desktop, but there are not many standard platforms for research. As a result, researchers from different groups often need to start from scratch in order to develop their work, and rarely share implementations details. Similarly, apart from a few publicly available examples, there is a lack of common datasets that are validated and can be used to compare results between different approaches.

In this paper, we introduce a platform that provides tools for resource assessment and mine planning, which can be accessed remotely utilising Jupyter Notebooks. In its current version, the platform provides (i) definition and implementation of basic structures for managing samples and models (loading, saving, adding or removing columns, etc.), (ii) definition of experimental and modelled variograms for spatial continuity analysis, (iii) definition of standard interpolation methods in resource evaluation (inverse distance and kriging), (iv) definition of standard problems in mine planning (ultimate pit and precedence constrained production scheduling), (v) generation of 2D standard graphs like grade distribution, conditional means, tonnage-grade, and production plans (vi) a set of datasets, which can be used for case studies, and (vii) a set of notebooks with examples and documentation.

In this paper, we introduce the platform and show some examples that illustrate some of the possibilities that it has for research and education in the area of resource assessment and mine planning.

INTRODUCTION

Resource assessment and mine planning are computationally intensive tasks, requiring extensive calculations and 3D visualisation of large datasets. This latter issue implies that most software developed for these tasks are designed to be executed on a desktop computer.

Having the whole application running on a desktop computer allows the software to take advantage of the full potential of the graphical hardware available for visualisation; however, it also has the drawback of limiting the applicability of high-performance computing, and other specialised resources. This is particularly important at the academic level, because it impacts the capability of sharing sets of data and the development and validation of new algorithms. In fact, many researchers often need to start from scratch to develop their algorithms, which requires testing them in a sufficiently large number of experiments to validate their results.

Another issue, related to validation of results, is that desktop software is rarely scriptable; thus, only a limited number of experiments can be run on them for comparison with new approaches. Finally, different software uses different file formats, and datasets have their own 'history' or metadata.

Some of these issues have been partially addressed in the past. For example, (Espinoza *et al.* 2013) provides a set of several datasets which are suitable for ultimate pit and direct block scheduling; the open mining format (OMF) (GMG Group 2017) is a file format standard to store mining data that is open and implemented in several commercial tools; and omfvtk (Sullivan 2017) is a set of python library that permits the visualisation of OMF files.

The current limitations for resource assessement and mine planning contrast, for example, with the case of data analytics, where research has widely benefited from the utilisation of remote computing, permitting many researchers and developers to collaborate, share code and datasets; thus allowing cross-validation of techniques and results. For example, Jupyter Notebooks allows accessing remote machines for development in numerous languages (among them Python), utilising only a web-browser.

In this paper, we introduce PYRAMP, a platform that provides data structures and algorithms for mineral resource assessment and mine planning. The platform, developed at the Advanced Mining Technology Centre, University of Chile, provides a starting point for developing research in both areas. The main aspects included in PYRAMP are summarised now, and explained in detail in the next sections.

- **Data structures.** The library provides several useful data structures to read and save samples and block models, as well as functions to work with this information in memory.
- **Utilities for resource assessment.** PYRAMP provides definition of experimental and modelled variograms for spatial continuity analysis, and definition of standard interpolation methods in resource evaluation, namely inverse distance and kriging methods.
- **Utilities for mine planning.** Definition of standard problems in mine planning: Ultimate and nested pits and precedence constrained production scheduling.
- **Visualisation.** Generation of 2D standard graphs like grade distribution, conditional means, tonnage-grade, and production plans
- **Datasets and examples.** Finally, PYRAMP provides several sets of data, which can be used for testing and validation of new algorithms, as well as many Jupyter Notebooks with examples and documentation.

The rest of the paper is organised as follows: Section 2 presents a general overview of the platform in its current state. Sections 3 and 4 introduce concepts and functionality available on the platform for resource assessment and mine planning, respectively. Section 5 presents other aspects of the library: visualisation utilities, examples and datasets. Finally, Section 6 presents the conclusions of this work, including future directions of development.

DESCRIPTION OF THE PLATFORM

In this section we provide a brief description of the platform as well as the main theoretical concepts from the literature. The section does not pretend to introduce the concepts related to resource modelling and mine planning, nor to go into their details or to be exhaustive. Indeed, we asume that the reader is familiar with these concepts and limit the exposition to provide a reduced background and explain how the concepts apply in the context of the platform.

Technical aspects of the platform

- **Getting PYRAMP**. To access to the platform, you need to request an account to pyramp@delphoslab.cl and access the site http://pyramp.delphoslab.cl.
- **Software/hardware requirements**. PYRAMP runs on a server at the University of Chile, therefore there are not major requirements from the client's side. The user, however, must have a modern web-browser. A dedicated graphic card is also recommended to view large datasets easily.

- **Documentation.** The library is documented using pydoc, which means that you can use the help command in Python to access detailed documentation about the methods in the library. You can also visit http://www.delphoslab.cl/MINELINKDOC for more detailed information about the underlying C++ implementation of the methods available for mine planning.

Organisation

Structure of the library. PYRAMP is organised into four packages: *pyramp.geostats, pyramp.planning, pyramp.utils* and *pyramp.views.*

Structure of files. A user of the platform has access to the following folders:

- *datasets/samples/* Contains sets of data with composite samples.
- *datasets/blockmodels/* Contains block models to be used for mine planning.
- *examples/geostats/* Contains notebooks with examples for resource assessment.
- *examples/mineplanning/* Contains notebooks with examples related to mine planning.

RESOURCE ASSESSMENT IN PYRAMP

This section provides a description of the tools and algorithms available in the *PYRAMP.geostats* package. For detailed information about all the classes and utilities, please refer to the in-line documentation or the documentation of the PYRAMP. (See Section 2.1 for more details.)

Variograms

The goal of variogram analysis is to describe the spatial continuity of the georeferenced variable under study. For this purpose, the aim is to characterise the variation in the values of this variable as a separation occurs between two analysed positions.

The traditional variogram analysis, having previously performed an exploratory analysis of the geolocated variable under study, starts with the calculation of the experimental variogram, and then defines a theoretical variogram model that will be used to continue with the analysis and its future estimation in a block model.

Experimental semivariogram

If we denote by $S = \{x_i \in \mathbb{R}^3, i = 1, ..., n\}$ the data locations and by $\{v(x), x \in D\}$ the regionalised variable over the domain D, the theoretical variogram is defined by:

$$\gamma(\mathbf{h}) = \frac{1}{2}\mathrm{Var}[v(\mathbf{x}+\mathbf{h}) - v(\mathbf{x})] = \frac{1}{2}\mathbb{E}\left\{[v(\mathbf{x}+\mathbf{h}) - v(\mathbf{x})]^2\right\}$$

This function calculates the variance of the differences of the variable values, for all pairs of positions that are separated by the oriented vector h.

The calculation of the experimental semivariogram or variogram for simplicity, consists of an estimator of this theoretical variogram, which measures the mean squared deviation among data values spatially separated by h:

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2|N_\gamma(\mathbf{h})|}\sum_{k\in N_\gamma(\mathbf{h})}[v(\mathbf{x}_k) - v(\mathbf{x}_k+\mathbf{h})]^2$$

In this case, $N\gamma(h) = \{k \in 1, \ldots, n: \{xk, xk + \mathbf{h}\} \subseteq S\}$ is the set of indexes of the data locations found that meet the condition of two samples separated by a given vector h, and $|N\gamma(\mathbf{h})|$ the number of pairs found in this set for that value of **h**.

When the variogram has the same behaviour in all directions, it is said that the phenomenon is isotropic (its continuity depends on separation, and not on direction), otherwise it is said that the phenomenon is anisotropic (Kitanidis 1997). For these analyses, it is usual to calculate the variogram in a specific direction, where the azimuth angle is measured against the north, and the dip angle is the inclination measured respect to the horizontal plane (NS) (Deutsch & Journel 1992).

Since in practice it is difficult to find samples that are exactly separated by a vector **h**, as defined in the set $N\gamma(\mathbf{h})$, certain tolerances are established in the search parameters for these pairs of samples. Thus, by defining these tolerances together with the parameters associated with the directions of analysis of the spatial continuity, the experimental variogram is fully established (Deutch Journel 1992).

Figure 1 shows an example and the list of parameters required to characterise experimental variograms.

```
directions = [{
"label": "dir1",
"azimuth": 0,
"azimuth_tolerance": 90,
"horizontal_bandwidth": 22,
"dip": 0,
"dip_tolerance": 90,
"vertical_bandwidth": 20.0,
"lag_count": 15,
"lag_size": 15,
"lag_tolerance": 7.5
}]
semivariogram(coordinates, grades, directions)
```

Figure 1. Example of a semivariogram using PYRAMP.

Modelled variogram

From the calculation of the experimental variogram, it is possible to fit a modelled variogram to complete the description of the spatial continuity. This modelled variogram is an input to the kriging methodology, where it is necessary to have a complete characterisation of the spatial continuity of the variable of interest, both for all directions and also for all possible separations in space. To fit a modelled variogram, it is necessary to have a characterisation of the spatial continuity on three orthogonal axes, being fully parameterised by the angles to describe the first direction and the three ranges corresponding to these axes (Deutch Journel 1992).

There exist several functions that comply with all the mathematical conditions that a variogram function must satisfy (Chiles and Delfiner 2009). The most used variogram model types are implemented in this library: the gaussian, exponential and spherical variograms. In order to fully describe the modelled variogram as outlined above, the following parameters must be defined for each variogram model: variogram type, sill, ranges (in each of the three directions of analysis) and angles (three angles to orient the first of the perpendicular directions of analysis) (Deutch Journel 1992). Each set of these parameters represents a variogram structure that serves to construct a modelled variogram.

The adjustment of variogram structure parameters is based on a careful analysis of the data, and their spatial continuity. If necessary, it is possible to consider more than one type of variogram structure in the complete modelled variogram, adding another of the mentioned variogram types, as well as a smallscale randomness component, traditionally called the nugget effect (Chiles and Delfiner 2009). To fully define a modelled variogram, PYRAMP requires as inputs the following parameters for each variogram structure: target variable, type of variogram structure (gaussian, exponential or spherical), sill, ranges (in each of the three orthogonal directions), and angles (definition of the first analysis direction).

In addition, it is possible to establish a nugget effect value for the spatial characterisation of the studied variable. An example of use is shown in Figure 2. After creating the variography model, it is necessary to create ad hoc structures to your variogram. The next step is to generate a certain number of 3D points, every metre across the direction azimuth and dip. Finally, a variography model can be calculated.

```
variography_model = VariographyModel()
sill = 0.14
angles = [0, 0, 0]
ranges = [96, 96, 96]
exp_struct = VariogramStructure("Exponential", sill,
angles, ranges)
variography_model.add_structure(exp_struct)
n_lags = 222
m_meters = 1
azimuth, dip = 0, 0
model_lags = generate_lags(azimuth, dip, m_meters, n_lags)
model_values = variography_model(model_lags)
```

Figure 2. Example of a modelled variogram using PYRAMP.

Interpolation methods

To generate the block models with estimations of the variable of interest in all the domain D, different types of methods have been developed (Li and Heap 2008; Chiles and Delfiner 2009; Meng, Liu and Borders 2013). Traditionally, linear interpolations have been widely used due their simplicity and reasonable results. The general equation describing this type of interpolation is given by:

$$\hat{v}(x) = \sum_{i \in N(x)} \omega_i v(x_i)$$

In this way, each position to be analysed in the domain $x \in D$ is estimated by a value $\hat{v}(x)$ based on a weighted sum with weights ω_i of each sample $v(x_i)$ with $x_i \in S$. On the other hand, the set $N(x)$ describes the set of indices of the x_i data locations that are going to be used for the calculation of the estimated value $\hat{v}(x)$. Thus, in the most general case, this set will present some of the indices associated with the available samples $N(x) \subseteq \{1, \ldots, n\}$, having equality in the case where all samples are considered for estimation. Due to practical considerations regarding computational costs necessary to obtain the weights, usually only a subset of the total samples is used to calculate each estimated value (e.g. closest samples to the target location to be estimated).

There are different methods for calculating the weights ω_i to obtain the estimated values at a given position x_i. In this library, two of the most used methods for this purpose are implemented, such as the inverse distance interpolation and the kriging method.

Inverse distance method

This method calculates the weights according to the distance between the data location x_i and the position x to be estimated. Thus, the weight $\omega_i = \omega_i(x, x_i)$ is defined as follows:

$$\omega_i(x, x_i) = \frac{1}{\|x - x_i\|_2^p}$$

For the calculation of this weight $\omega_i(x, x_i)$ we make use of the distance between x and x_i, based on the Euclidean norm $\|*\|_2$. The exponent p associated with this distance is considered as an additional parameter, which modulates the effect of the contribution of these distances in the calculation of the associated weight (Cheng and Liu 2012).

To keep the values of the interpolation near the values of the samples considered in the neighbourhood $N(x)$, a normalisation of the calculated weights is performed, so that the sum of these weights is 1 for each position x to be estimated.

To use this interpolation method in PYRAMP, it is necessary to define the following parameters: target variable, output blockmodel datafile (grid parameter or full coordinates to each point), distance's exponent p, maximum search radius, and minimum number of neighbours. See Figure 3 for an example.

```
search_options = [{
"exp_dist" = 3,
"r_max" = 10,
"n_nbhs" = 20
}]
idw = inverse_distance(coordinates, variable_target,
blockmodel, search_options, miss_value=-99)
```

Figure 3. Example of an inverse distance calculation using PYRAMP.

Kriging methods

Kriging-based interpolation methods make use not only of the spatial configuration of the data, but also consider the spatial continuity of them. Using a function that quantifies the spatial structure of the geolocalised samples (e.g., variogram), and different assumptions made about the variable to be estimated, the basis of the different types of kriging for the calculation of the weights ω_i are constructed.

Kriging methodologies present some important features that have led them to be the traditional technique for the estimation of spatial variables in mining contexts. In particular, it has the properties of being unbiased, the mean error is zero; and optimal, the estimation error variance is minimal (Armstrong 1998; Chiles and Delfiner 2009).

The two most used kriging methods are simple kriging (SK) and ordinary kriging (OK), which are widely used in the estimation of continuous spatial variables. In the case of SK the mean of the estimated variables are considered to be known, while in OK the mean is constant but unknown. Both methodologies provide values for the weights ω_i based on the resolution of a system of equations, according to the considerations of each case (Goeverts 1997). Figure 4 provides an example including the required parameters.

```
ck1 = Kriging({
"samples_coords": coordinates,
"samples_values": grades,
"limits": [0, 1e21],
"top_cut": [1e6],
"missing_value": -99,
"kriging_type": "ordinary",
"nb_radius": [150, 85, 85],
"nb_angles": [0, 0, 0],
"nb_octant": False,
"nb_ndata": [4, 8],
"variogram": variogram_modeled
})
result = ck1({
"target": "scattered",
"target_coords": blockmodel_points
}
```

Figure 4. Example of kriging calculation using PYRAMP.

MINE PLANNING IN PYRAMP

This section provides a short description of some of the problems and algorithms available in the PYRAMP.planning package, however this section is far from exhaustive. For detailed information about all the classes and utilities, please refer to the in-line documentation or the documentation of the PYRAMP (see Section 2.1 for more details).

Ultimate and Nested Pits

PYRAMP provides two classes to address the problem of the ultimate pit problem. FinalPitInstance, that specifies the problem to be solved; and FinalPitSolver abstracts any algorithm able to solve the final pit problem. This class implements a solver based on the pseudo-flow algorithm.

Figure 5 presents a code snippet that solves an instance of the final pit problem. The example creates an object fpi that takes an object slope_prec of type Precedence which has already been set with the slope precedence arcs, and that a column name value that contains the economic values of the blocks. It then sets a solver fps, runs it, and retrieves its solution as an object fs. This solution is then stored in a column named FINALPIT.

```
fpi = FinalPitInstance("value",slope_prec)

fps = FinalPitSolver(fpi)

fps.Run()

fp = fps.FinalPit()

fp.StoreAsAttribute("FINALPIT")
```

Figure 5. Example of computation of an ultimate pit using PYRAMP.

Direct block scheduling

PYRAMP implements Direct Block Scheduling as the Precedence Constrained Production Scheduling Problem. Within this framework, there is a set $\boldsymbol{T} = \{1, 2, \ldots, T\}$ of periods, $\boldsymbol{D} = \{1, 2, \ldots, D\}$ of destinations and each block b has economic values $v(b, d, t)$ corresponding to the net profit perceived if block b is extracted at period t and sent to destination d. For each block the following decisions are made: (i) when to mine each block, and (ii) what fraction of the block must be sent to each potential destination. (PYRAMP also implements a variation which does not allow the block to be split into fractions to be sent to different destinations.) Under this setting, the problem is to maximise the net present value (NPV) given by these decisions, under several constraints, which can be customised or already implemented, like precedence, capacity and blending constraints, among others.

As in the case of the final pit, PYRAMP provides utilities for specifying the instances in terms of periods and constraints, two solvers based on integer programming, and interfaces to several mixed integer programming solvers (CPLEX, Gurobi and CBS). It also provides an implementation of the Bienstock-Zuckerberg algorithm.

Precedences

Precedences are an essential component of open-pit planning, which allow modelling slope angles in the case of open-pit mining. PYRAMP provides several opportunities to work with precedences, the most elementary one being the Precedence class, which is simply a container of arcs. An object of this type is always linked to a block model to which it refers.

SlopePrecedence is a subclass of Precedence with utility functions to compute arcs. The most relevant method of this class is CreateArcs which populates the collection with arcs for a given definition. For a block $b \in B$ with coordinates (x, y, z), its predecessors are those blocks $b' \in B$ with coordinates (x', y', z') that verify

$$z < z' \leq z + h \ \wedge \ \frac{\sqrt{(x - x')^2 + (y - y')^2}}{z' - z} \leq \tan(\alpha) \tag{5}$$

Figure 6 provides an example of the creation of a slope precedence with a constant angle of 50° for a block model with block sizes of 12.5 × 12.5 × 15.0 m. In this case, we provide CreateArcs with the dimensions of the blocks, which allows a significant speed-up of the computation. (In the example, bm is a block model.)

```
prec = SlopePrecedence( bm )

prec.CreateArcs( 50.0, 8, 12.5, 12.5, 15.0 )
```

Figure 6. Example of computation of an ultimate pit using PYRAMP.

Finally, if more complex slopes are required, the class Precedence provides a method AddArc(b, b') which adds the arc (b, b') to the set. (Here b' is a predecessor of b.)

a

Figure 7. Minimal precedence patterns for a 45° slope angle and five levels of precision. (Blocks are displayed smaller than their actual size so it is possible to see them individually.)

OTHER ASPECTS

In this section, we illustrate other relevant aspects of the platform, like the visualisation tools, Jupyter Notebooks with examples and datasets, which are used for the examples, and are available for research and teaching.

Visualisation

The platform provides several visualisation tools that provide a graphical representation of the results.

Exploratory data analysis. PYRAMP provides some graphs that are useful in order to study, in a general way, the behaviour of a geolocalised variable. (See Figure 8 (left) for an example of a histogram.)

Spatial continuity analysis. With the tools presented in this library, it is possible to generate the graphs associated with the spatial continuity of a 3D variable, through the experimental variogram and the modelled variogram (See Figure 8 (right)). Additionally, it is possible to use the conditional means graph to verify the behaviour of a variable in each of the axes in the space.

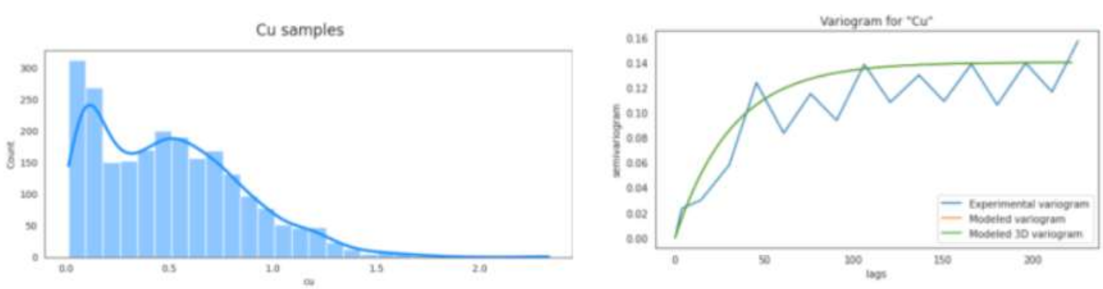


Figure 8. Left: A Cu histogram. Right: Variography.

Tonnage-grade graph. This graph allows for characterising the tonnage and the average grade that would be obtained for a given cut-off grade. Since this graph gives the results for different cut-off grades, the effect of choosing a cut-off grade on the quantity and quality of the ore obtained can be analysed.

Pit by pit graph. This graph permits visualisation of the results of a nested pit instance. It displays the accumulated value, mineral and waste tonnage of the pits generated using the nested pits approach.

Figure 9 (left) shows an example of the pit-by-pit graph for nested pits generated for revenue factors $\lambda = i/20$ for $i = 1,\ldots,20$.

Block model visualisation. Viewing blocks in 3D permits to analyse the spatial distribution of any block attribute. Figures 9 (center and right) shows an example of the grades in a solution of the ultimate pit and block economic values for a selection of specific blocks.

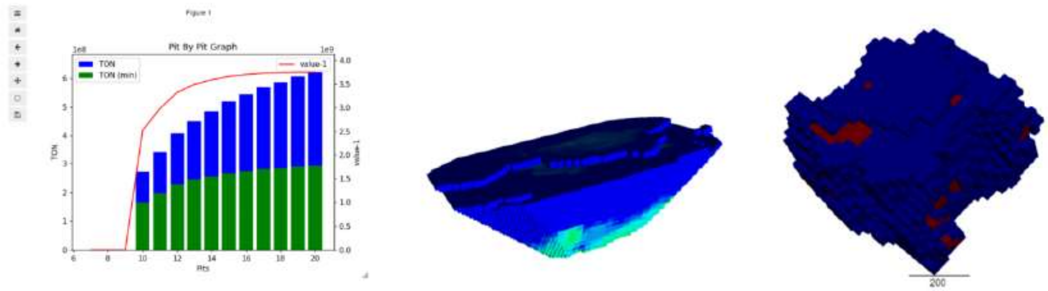


Figure 9. Left: Pit by pit graph. Center: 3D visualisation blocks' grades in an ultimate pit. Right: 3-D visualisation of economic value of some blocks.

The method BMView(block_model, filter_function) produces 3D visualisation of block models. It receives a block model and a function that evaluates blocks and returns True or False, so blocks in the model such that the function return is true are displayed. Other visualisation utilities include the grade-tonnage curve of a block model, pit by pit graphs and a production plan graph.

Examples

The platform provides many Jupyter Notebooks of utilisation, which are summarised in Table 1.

The examples are organised in two folders. (The examples use the datasets presented in Section 5.3.)

- *examples/modelling* contains all the examples related to resource modelling, which include experimental variography, variogram modelling, kriging estimation and inverse distance interpolation.
- *examples/planning* contains the mine planning examples. These examples include simple cases like working with block models, to scheduling using DBS.

Table 1. Examples of utilisation of the platform

Resource Modelling (datasets/examples/modeling)	
Notebook	**Description**
01-samples.ipynb	Show how to handle composites samples. Includes exploratory analysis, experimental variography and variogram modelling.
02-kriging.ipynb	Kriging estimation over a block model datafile. The resulting estimation is saved to a file.
03-inverse-distance.ipynb	An example to inverse distance interpolation over a grid datafile.
Mine Planning (datasets/examples/mineplanning)	
Notebook	**Description**

01-blockmodels.ipynb	Basic functionality: Reading block models, creating new columns, saving to disk.
02-finalpit.ipynb	A full example of computation of an ultimate pit. Valorisation of blocks.
03-nestedpits.ipynb	An example of nested pits.
04-precedences.ipynb	An example illustrating the impact of precedences in the ultimate pit.
05-dbs.ipynb	A direct block scheduling example.

Datasets
The platform provides several datasets available to any potential user, which are available under datasets folder. Tables 2 and 3 describe the sets of data for samples and block models.

Table 2. Datasets of samples (folder datasets/samples)

Dataset name	# Samples	Column names	Description
composites_v3.csv	2,289	dhid, cu, rocktype	Includes drillhole id, copper grades and rocktype to each sample.
babbitt_v1.csv	109,436	dhid, cu, fe	Includes drillhole id, copper and iron grades to each sample.

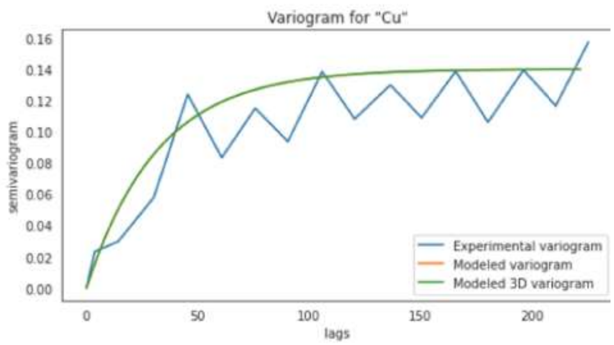
Table 3. Block model datasets (folder datasets/blockmodels)

Block Model	# Blocks	Block Dimensions	Column Names	Description
bm1.txt	497,760	15 × 15 × 15	DENSITY, TON, CU, AU	An open pit block model with copper CU and silver AU grades, tonnage TON and density DENSITY
mine3.txt	26,787	10 × 10 × 10	CU, TON	Block model with copper grade and tonnage.
delphos5.txt	323,200	12.5 × 12.5 × 15.0	Tonnage, Cu	Block model with copper grade and tonnage per block
grid_geology.txt	28,800	10 × 10 × 12	cu, ton, rocktype, density	Blockmodel with geology and density, used as base for estimation along with composites_v3.csv dataset.

A DETAILED EXAMPLE

In this section we present a summary of an example where some utilities in the library are utilised to model a copper deposit and compute an ultimate pit and production plan. The data for the example is available on the platform. The complete version of the Jupyter Notebook that corresponds to this example can be found in https://delphoslab.cl/media/pyramp-notebook-example.pdf.

1. **Experimental variography and variogram model.** In this step, after loading the composite samples, we visualise the grade statistics. An experimental variography is then calculated to obtain the parameters that will be used on the variogram model. The resulting model, along with the experimental variogram, is displayed in Figure 10 (left).
2. **Kriging estimation of block model.** Using the modelled variogram in the previous step, we perform a kriging estimation of the copper variable using a search radius of (150, 85, 85). We use a given grid with density and rock type (See Table 3) to estimate ore grades. The resulting block model (shown in Figure 10 on the right) is used in the next steps, for mine planning purposes.



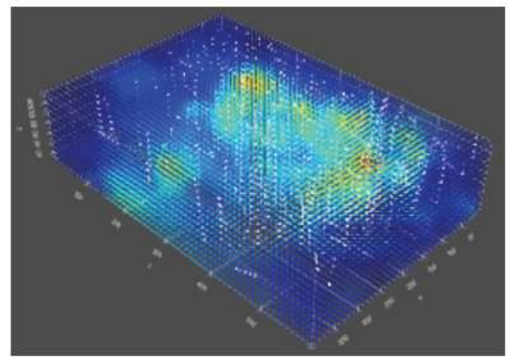


Figure 10. Resource modelling results using PYRAMP. Left: An example of a variogram. Right: Results of the kriging estimation.

3. **Evaluate the block model.** The values of the blocks are computed using the formula $V_1 = (P - C_S)RFgT - (C_M + C_P)T$ for blocks that extracted and processed, and $V_2 = -C_M T$ for blocks that extracted and sent to the waste dump. The value of the block is then $V = max(V_1, V_2)$.
4. **Compute ultimate pit and nested pits.** In this part of the notebook, we calculate the ultimate pit and nested pits using revenue factors $\lambda = k/20$ for $k = 1, 2, \ldots, 20$.
5. **Generate a production plan.** In this section, we select some of the pits and generate a production plan. We use a simple criterion (similar tonnage). To estimate the production and mining capacities, we utilise the best case approach to try several scenarios and choose one with high NPV. Then we present the best and worst case production plans for the chosen scenario and select the best alternative.

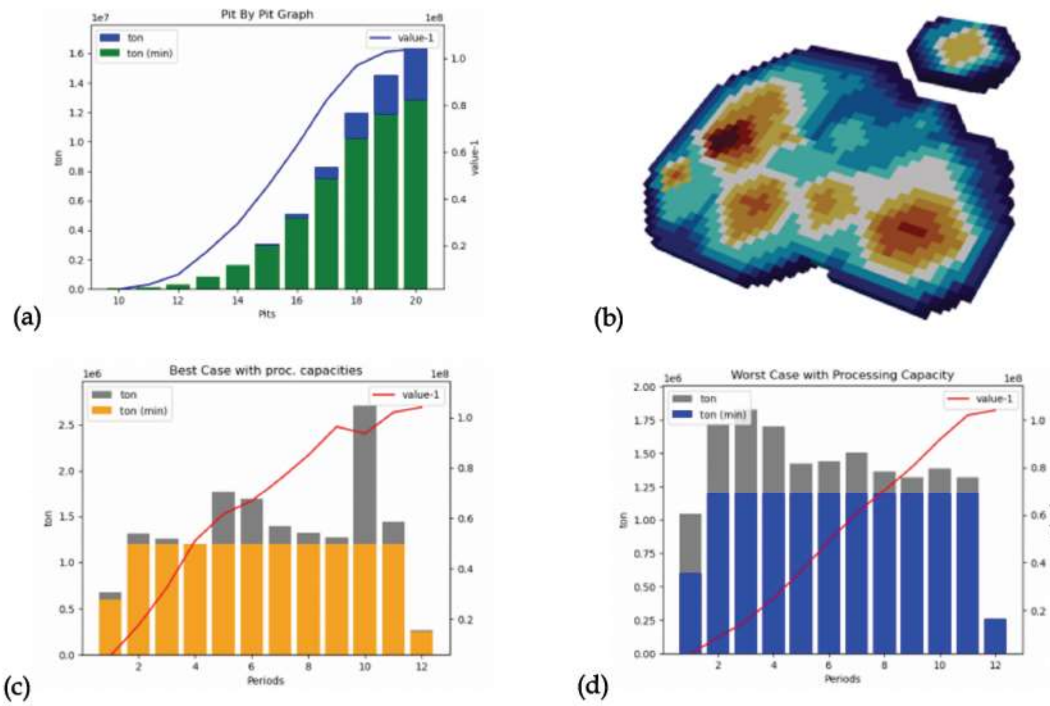


Figure 11. Results from mine planning in PYRAMP. (a) A pit-by-pit graph, (b) Graphical results of nested pits, (c) Production plan (best case), (d) Production plan (worst case).

The results of the mine planning steps are shown in Figure 11. (a) presents the tonnages and cumulated values of the 10 nested pits (from $\lambda = k/20, k = 11, \ldots, 20$) and (b) shows an isometric visualisation of the same pits, represented with coloured blocks. (c) and (d) present two production plans corresponding to the best and worst case. The net present values of these production plans are, respectively, MMUSD 40.6 and MMUSD 32.5.

CONCLUSIONS

In this paper, we have presented PYRAMP, a platform which uses Jupyter Notebooks to access libraries for resource assessment and mine planning. We have presented the main technical aspects and structure of the platform, as well as provided some examples to illustrate its capabilities.

Currently, the library provides a set of data structures, algorithms and utility functions, which allow the modelling of the deposit up to computing production schedules, as well as tools for managing the information and performing advanced 3D visualisation. The library also provides several datasets that can be used for academic purposes, thereby providing a common set of instances to test, validate, and compare research developments.

We expect to extend the library in the future, and for the community to make use of it, extending the datasets and examples that are available.

ACKNOWLEDGMENT

This work was funded by the Cilean National Agency for Research and Development of Chile (ANID) through the basal grant AFB180004 "Advanced Mining Technology Center" and FONDEF grant ID19I10155.

REFERENCES

Margaret Armstrong (1998). Basic linear geostatistics. Springer Science & Business Media, 1998.

D. Bienstock and M. Zuckerberg (2009). A new LP algorithm for precedence constrained production scheduling. *Optimization Online.*

J.P. Chiles and P. Delfiner (2009). Geostatistics: modeling spatial uncertainty, volume 497. John Wiley & Sons.

F.W. Chen and C.W. Liu (2012). Estimation of the spatial rainfall distribution using inverse distance weighting (idw) in the middle of taiwan. *Paddy and Water Environment*, 10(3):209–222.

C.V. Deutsch, A.G. Journel, *et al.* (2012) Geostatistical software library and user's guide. New York, 119(147).

D. Espinoza, M. Goycoolea, E. Moreno, and A. Newman (2013). Minelib: a library of open pit mining problems. *Annals of Operations Research*, 206(1):93–114.

P. Goovaerts et al. (1997). Geostatistics for natural resources evaluation. Oxford University Press on Demand, 1997.

Gemcom (2018). Gemcom Whittle™ Strategic Mine Planning software. *http://www.gemcomsoftware.com/products/whittle*, 2018. Accessed 18 July 2018.

D. S. Hochbaum and A. Chen (2000). Performance analysis and best implementations of old and new algorithms for the open-pit mining problem. *Operations Research*, 48:894–914.

D.S. Hochbaum (2008). The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research, 56*(4), 992-1009.

E. Jélvez, N. Morales, P. Nancel-Penard, J. Peypouquet, and P. Reyes (2016). Aggregation heuristic for the open-pit block scheduling problem. *European Journal of Operational Research*, 249(3):1169–1177.

E. Jélvez, N. Morales, and P. Nancel-Penard (2019). Open-pit mine production scheduling: Improvements to Minelib library problems. In *Proceedings of the 27th International Symposium on Mine Planning and Equipment Selection-MPES 2018*, pages 223–232. Springer.

E. Jélvez, N. Morales, P. Nancel-Penard, and F. Cornillier (2020). A new hybrid heuristic algorithm for the precedence constrained production scheduling problem: A mining application. *Omega*, 94:102046.

T.B. Johnson (1968). Optimum open-pit mine production scheduling. PhD thesis, *Operations Research Department, University of California, Berkeley.*

T.B. Johnson (1969). A Decade of Digital Computing In the Mineral Industry, chapter Optimum open-pit mine production scheduling, pp. 539–562.

A. Kenny, X. Li, A. Ernst, and D. Thiruvady (2017). Towards solving large-scale precedence constrained production scheduling problems in mining. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1137–1144. ACM

P.K. Kitanidis (1997). Introduction to geostatistics: applications in hydrogeology. Cambridge university press.

H. Lerchs and H.C. Grossman (1965). Optimal design of open-pit mines. Transactions C.I.M., 58:47–54.

J. Li and A.D. Heap (2008). *A review of spatial interpolation methods for environmental scientists. Geoscience Australia,* Record 2008/23, 137 pp.

Q. Meng, Z. Liu, and B.E. Borders (2013). Assessment of regression kriging for spatial interpolation–comparisons of seven gis interpolation methods. Cartography and geographic information science, 40(1):28–39.

GMG Group (2021). *https://gmggroup.org/projects/data-exchange-for-mine-software/*, 2017. Accessed January 2021.

M.J. Pyrcz and C.V. Deutsch (2014). Geostatistical reservoir modeling. Oxford university press.

J.C. Picard (1976). Maximal closure of a graph and applications to combinatorial problems. *Management Science*, 22:1268–1272.

O. Rivera, D. Espinoza, M. Goycoolea, E. Moreno, and G. Muñoz (2020). Production scheduling for strategic open pit mine planning: A mixed-integer programming approach. *Operation Research*, 68(2).

B. Sullivan (2017). *https://pypi.org/project/omfvtk/*. Accessed January 2021.

Nelson Victor Morales Varela

Professor
Polytechnique Montreal

Nelson Morales is a Mathematical Engineer with a PhD in Computer Science. For about 20 years has developed research and technology in the area of mine planning. Until recently, he was the director of DELPHOS Mine Planning Laboratory of the Advanced Mining Technology at the University of Chile. Currently, he became an Associate Professor at Civil, Geology and Mining Engineering Department in Polytechnique Montreal, Quebec, Canada.

His interests are in optimization and simulation for modeling mining operations and optimize their plans.