

Feasibility Grade Control (FGC): Simulation of Grade Control on Geostatistical Realizations

Brandon J. Wilde and Clayton V. Deutsch

Centre for Computational Geostatistics
School of Mining and Petroleum Engineering
University of Alberta

Predicting recoverable reserves before mining commences is an important problem in ore reserve estimation. The conventional approach of categorizing block estimates as ore or waste does not always fairly represent dilution, lost ore and the anticipated grade control data. Simulating short term planning, that is, drawing dig limits is tedious and impractical on many benches and simulated grade realizations. A new method is proposed in this paper. A high resolution simulated realization is processed to establish dilution and lost ore. The high resolution blocks are amalgamated into practical mining units by an optimization technique. The number of blocks to amalgamate and controls on their geometry are controlled to mimic anticipated selectivity. Multiple benches and multiple realizations are processed quickly and efficiently. The methodology is developed and examples are presented.

Introduction

Mining is the process of removing useful and valuable minerals and commodities from the earth. There are many mines around the world. Before any mine is constructed, there must be adequate reason to justify construction of the mine. This is determined by conducting a thorough study of the area, including drilling holes to get an idea of the geological resource. With this information, a block model of the underground formation is built. This block model can be used to calculate the extent of profitable resource.

Geostatistical methods are often used to characterize the resource using exploration data. The total resource is delineated by characterizing many small blocks within the volume of interest. Each block's profitability is calculated based on the amount of valuable mineral it contains and its location within the mine. A block is considered profitable if the revenue from processing it as ore exceeds the cost of mining it as waste. A cutoff grade may be established based on milling cost, ore mining cost, waste mining cost, recovery, and the metal price. A grade above the cutoff grade is considered profitable, below cutoff grade is not. Profit-based grade control uses multiple simulated realizations of the ore grade. These realizations are passed through a transfer function based on the same factors used to calculate a cutoff grade. The transfer function produces a model of the expected profit if called ore for each block. Blocks with profits above zero are profitable to mine; blocks with profits below zero are costly to mine and may be wasted.

Classifying profitable blocks as ore and unprofitable blocks as waste is simplistic. There are a number of factors that must be considered. Firstly, more information will be available at the time of mining; selectivity will be better than implied by a kriged estimate based on widely spaced exploration data. Secondly, the mining equipment will select according to the chosen benches and with a relatively large scale; selectivity will not be at the scale of drillhole data. There are other factors including the spacing of the blastholes, the blasthole sampling error and visual controls on ore and waste.

Delimiting the ore and waste during mining is referred to as grade control. The most common approach to grade control consists of mapping the grades, drawing dig limits and transferring them to the field for mining equipment. The dig limit algorithm was developed to mimic this operation (see previous CCG papers by Deutsch and Norrena). Dig limit optimization starts with an initial hand-drawn dig limit and optimizes the limit based on an objective function that maximizes profitability subject to the limitations of the mining equipment. This approach is recommended for short term planning at mines; however, it is impossible to apply on feasibility models. There are many benches, pit stages, ore/waste units, and multiple realizations. The professional time to implement the algorithm is prohibitively large. Even if it

were accepted, it would be inflexible and difficult to ensure quality control throughout the exercise. Attempts to automate initial dig limit selection have failed in the vicinity of erratic ore/waste interfaces – the logic to merge/split polygons and explore the optimality of including isolated waste in ore (and ore in waste) is difficult to automate.

A simple alternative consists of choosing a reasonable Selective Mining Unit (SMU) size, scaling the feasibility model to the chosen SMU size, then categorizing as ore/waste. This approach is reasonable and should be tested on virtually every feasibility model; however, the precise SMU size is difficult to establish. It could be calibrated with production data or small-scale grade control studies. An alternative approach is desired for checking and validation.

This note outlines a new method called feasibility grade control (FGC) because of its applicability with models at the feasibility scale. The basic idea is to consider optimal grade or profit values at a small scale. The smallest scale could be the amount of material in one pass of loading the equipment. In practice, we may consider a larger scale to mimic the anticipated information available for grade control. The algorithm optimizes how the elementary block units are accumulated to larger mining units. Some waste blocks will be taken with ore units and some ore blocks are lost in waste units. It is precisely this dilution and lost ore that is of interest at the time of a feasibility study.

Feasibility Grade Control

Geostatistics is also used as the basis for the grade control model. As with dig limit-based grade control, an expected profit if called ore model is required as input. Each block of the model is considered an element in the final mining units, as mentioned, at the smallest scale each block could represent a pass by the loading equipment and will be part of a certain truck load or mining unit. The size of the equipment and other factors will determine the number of blocks assigned to each mining unit. Typically, 2-4 passes are needed to load a truck, thus 2-4 blocks will constitute a mining unit. The goal of feasibility grade control is to arrange the blocks such that every mining unit consists of all ore or all waste. This will be accomplished with varying success depending on how erratic the deposit is. A small example is shown to illustrate the principle.

This example consists of an aerial grid with 100 blocks (10 x 10). For simplicity, we will first assume that each mining unit consists of four passes, or blocks. The majority of the blocks are ore, but there is a strip of waste running diagonally across the area of interest as shown below:

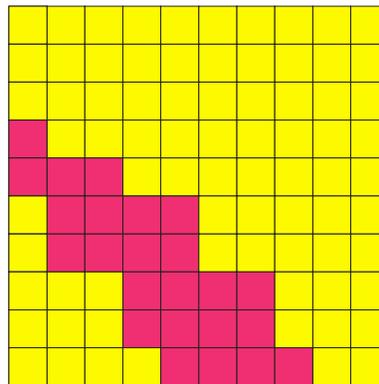


Figure 1: Layout of small example.

The blocks are assigned to mining units with each unit containing four blocks. We start with each being accumulated into square units. Mining progresses from the lower left to the upper right. The mining units are numbered from 1 to 25. This initial numbering is of no consequence. It is not meant to resemble mining progression. It is simply necessary to associate each block with a mining unit.

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 15 | 15 | 16 | 16 | 22 | 22 | 23 | 23 | 25 | 25 |
| 15 | 15 | 16 | 16 | 22 | 22 | 23 | 23 | 25 | 25 |
| 7 | 7 | 14 | 14 | 17 | 17 | 21 | 21 | 24 | 24 |
| 7 | 7 | 14 | 14 | 17 | 17 | 21 | 21 | 24 | 24 |
| 6 | 6 | 8 | 8 | 13 | 13 | 18 | 18 | 20 | 20 |
| 6 | 6 | 8 | 8 | 13 | 13 | 18 | 18 | 20 | 20 |
| 2 | 2 | 5 | 5 | 9 | 9 | 12 | 12 | 19 | 19 |
| 2 | 2 | 5 | 5 | 9 | 9 | 12 | 12 | 19 | 19 |
| 1 | 1 | 3 | 3 | 4 | 4 | 10 | 10 | 11 | 11 |
| 1 | 1 | 3 | 3 | 4 | 4 | 10 | 10 | 11 | 11 |

Figure 2: Initial unit numbering sequence following mine progression, from bottom left to top right.

Mining unit or load 2 consists of ore with one block of waste; unit 6 is waste with one block of ore. We can see that there is dilution of the ore with the waste and that ore is being lost as waste. Each of these situations lowers profit. We will assume that each ore blocks profitability is +\$10 and each waste block's profitability is -\$2. The total profit of mining the area is calculated as follows:

$$\text{Profit} = \sum_{\substack{\text{ore} \\ \text{loads} \\ (l)}} \sum_{\text{blocks}} P_{l,k} + \sum_{\substack{\text{waste} \\ \text{loads}}} n_p \cdot P_w$$

Where $P_{l,k}$ is the profit of each block, n_p is the number of passes or blocks in a mining unit, and P_w is the cost of mining a waste block and is a constant. This form of the profit calculation assumes that the mining units are shaped in a desirable configuration. For this example, the total profit with this configuration is \$652. It is apparent that this situation is not optimal as there are many units that contain both ore and waste. The best results would come from having units of all waste or all ore. This is shown in the configuration below. Using the same parameters, we now have a profit of \$712, a 9.2% increase.

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 15 | 15 | 16 | 16 | 22 | 22 | 23 | 23 | 25 | 25 |
| 15 | 15 | 16 | 16 | 22 | 22 | 23 | 23 | 25 | 25 |
| 8 | 8 | 9 | 9 | 17 | 17 | 21 | 21 | 24 | 24 |
| 7 | 8 | 8 | 9 | 17 | 17 | 21 | 21 | 24 | 24 |
| 7 | 7 | 7 | 9 | 14 | 14 | 18 | 18 | 20 | 20 |
| 2 | 6 | 6 | 10 | 10 | 14 | 18 | 18 | 20 | 20 |
| 2 | 6 | 6 | 10 | 10 | 14 | 12 | 12 | 19 | 19 |
| 2 | 2 | 3 | 5 | 5 | 11 | 11 | 12 | 19 | 19 |
| 1 | 1 | 3 | 5 | 5 | 11 | 11 | 12 | 13 | 13 |
| 1 | 1 | 3 | 3 | 4 | 4 | 4 | 4 | 13 | 13 |

Figure 3: Modified unit sequence. Each unit carries only ore or waste.

In reality, the profit model is more complex, that is, not all ore/waste blocks are adjacent and dilution can never be completely removed. But the concept remains the same. The goal is to maximize profit by maximizing the value of each unit of ore and by minimizing the amount of ore lost in waste units.

Although the algorithm applies to arbitrarily elementary blocks and larger mining units, there is an underlying model of loader passes filling a truck. The assumption that blocks will be loaded with uniformly shaped blocks is also less than ideal. The selection abilities of the loading device (usually shovel or front end loader) would have to be much better than they are now. The loading device would also have to be extremely mobile which is not the case with today's equipment. A loading scenario is likely to follow one of the following configurations.

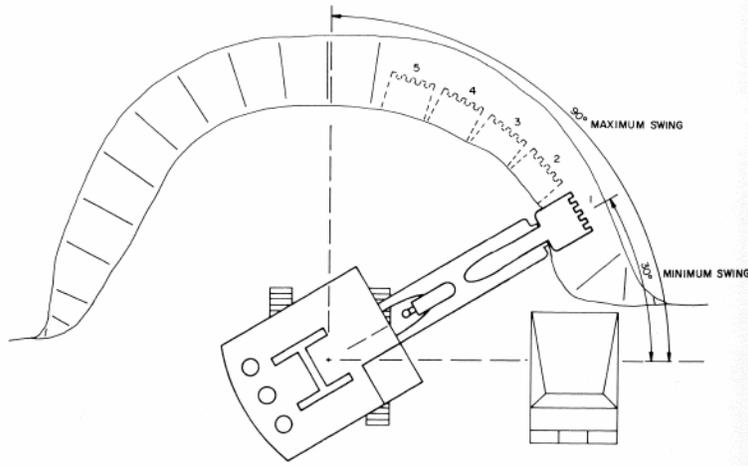


Figure 4: Pass configuration for double-backup operation.

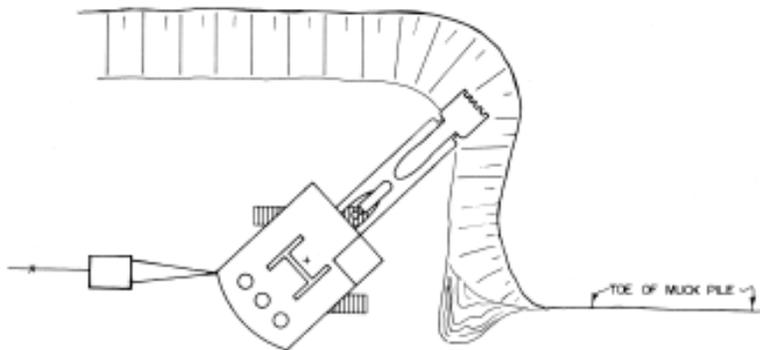


Figure 5: Pass configuration for drive-by operation.

We do not, of course, recommend or suggest that the mine plan on taking waste with ore or ore with waste. The procedure devised here aims to predict the inevitable dilution and lost ore. During mining, all practical steps should be taken to correctly identify, then mine ore and waste.

More Details of Methodology

Feasibility grade control requires a profit model setup as benches as input. The Center for Computational Geostatistics (CCG) has a guidebook titled 'Guide to Geostatistical Grade Control and Dig Limit Determination.' It includes detailed instructions for creating the profit model.

FGC also has an option of inputting a categorical pit stage model. If chosen, this model is used for two purposes: the first is to determine a starting point for mining to begin; the second is to create a mining schedule subsequent to the optimization process. Whether a categorical model is input or not, a starting point must be determined as mining direction is relevant to the optimization process. If no categorical model is used, the starting point is assumed to be the centroid of the top layer of the profit model; if a categorical model is used, the starting point is the centroid of the top layer of the first pit stage. This starting point is necessary for determining the mining direction at each mining unit. Four quadrants are created with their origin at this centroid. Mining units in the upper right quadrant have a mining direction towards the upper right, units in the lower left quadrant have a mining direction towards the lower left and so on (see Figure 6). This principle becomes necessary for index calculation later on in the methodology.

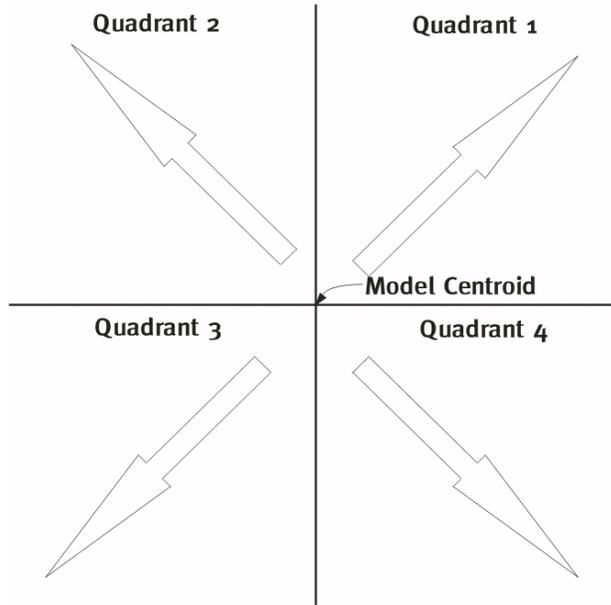


Figure 6: Mining directions move outwards away from the model centroid.

Ideally, the block size of the profit model would correspond to an elementary SMU size for the given mine. Each block is assigned to a specific mining unit, as shown previously in Figure 2. This is done differently depending on the number of passes the loading device must make to fill the truck. The beginning initial configurations are shown for two- and three-block mining units in Figure 7 and Figure 8 respectively. Four-block mining units are initialized as squares as shown in Figure 2. Again, the actual numbering sequence is inconsequential at this stage.

| | | | | | | | |
|---|---|---|---|---|---|----|----|
| 7 | 7 | | | | | | |
| 6 | 6 | 8 | 8 | | | | |
| 2 | 2 | 5 | 5 | 9 | 9 | | |
| 1 | 1 | 3 | 3 | 4 | 4 | 10 | 10 |

Figure 7: Initial unit numbering sequence for 2 block units.

| | | | | | | | | |
|----|----|----|---|----|----|---|---|---|
| 11 | 12 | 12 | | | | | | |
| 11 | 11 | 12 | | | | | | |
| 3 | 4 | 4 | 9 | 10 | 10 | | | |
| 3 | 3 | 4 | 9 | 9 | 10 | | | |
| 1 | 2 | 2 | 5 | 6 | 6 | 7 | 8 | 8 |
| 1 | 1 | 2 | 5 | 5 | 6 | 7 | 7 | 8 |

Figure 8: Initial unit numbering sequence for 3 block units.

Once every block is assigned to a mining unit, the initial profit of each unit is calculated. Unit profit is simply the sum of the expected profit for each block in the unit:

$$P_{u_j} = \sum_{i=1}^{n_p} P_{b_i}$$

Where P_u is unit profit, P_b is block profit, and n_p is the number of blocks per unit. If the unit profit is greater than zero, every block within the mining unit is considered ore. If the unit profit is less than zero, every block in the mining unit is considered to be waste. The sum of all the ore mining unit profits is calculated as well as the sum of all the waste mining unit profits.

The objective function which we aim to optimize by maximizing its value is defined as the sum of all the positively valued (ore) mining units minus the sum of all of the negatively valued (waste) mining units:

$$P_T = \sum_1^{N_O} P_O - \sum_1^{N_W} P_W$$

Where P_T is the value of the objective function, N_O is the number of ore units, P_O is the ore unit's profit, N_W is the number of waste units, and P_W is the waste unit's profit.

The initial block assignment is now optimized. Clear ore/waste contacts will permit a configuration of blocks that have little dilution or lost ore. Erratic contacts will lead to inevitable dilution. An optimization process is considered:

- Generate a random path through the model
- At each block:
 - Choose a random neighbor from the eight surrounding blocks
 - Switch the unit numbers for those blocks
 - Check the viability of the new configuration, if doable apply the appropriate penalty
 - Recalculate the objective function based on the new unit configuration
 - Compare this value:
 - If it is higher, keep the switch and move to a new block
 - If it makes no improvement, switch back the unit numbers and move to a new block

This process is repeated as many times as has been specified by the user, typically from 5-8 times.

Within the process there are some constraints that must be adhered to. The mining unit configuration, or shape, is the most important constraint. Mining units cannot be composed of any blocks located anywhere. Their shape must fit some specific limitations. The first shape limitation is that every block within a unit must be adjacent to a minimum of one other block in that same unit. There are eight possibilities for considering a block as adjacent: above, below, left, right, above-right, above-left, below-right, and below-left. The second limitation is that the unit must be in an easily extractable shape, or it will be penalized. Figure 9 illustrates. The penalty is a fractional reduction in unit profitability.

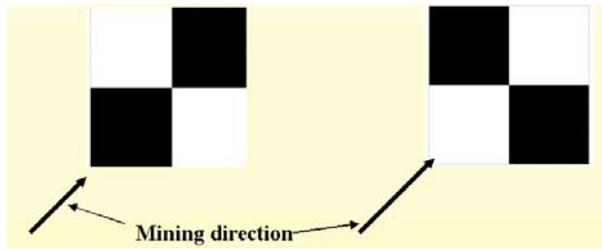


Figure 9: Mining units must be in an easily extractable configuration, otherwise they are penalized. The left figure shows a shape which is not easily extractable as the blocks below and left of the top-right block have not yet been mined. The right figure shows a shape which is easily extractable as there are no unmined blocks obstructing it.

It is apparent that there are only a few allowable geometric configurations, some being better than others. Each geometric possibility has been found for two-, three-, and four-block mining units and they are shown in Figures 10, 11, and 12 respectively. Each has its own particular index which is also shown. This index makes it simple to reference and catalog each unit's configuration.

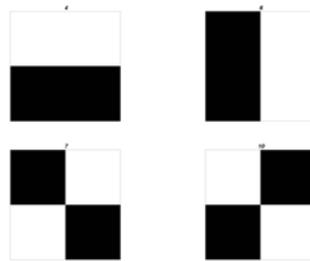


Figure 10: Geometric configurations for two-block mining units.

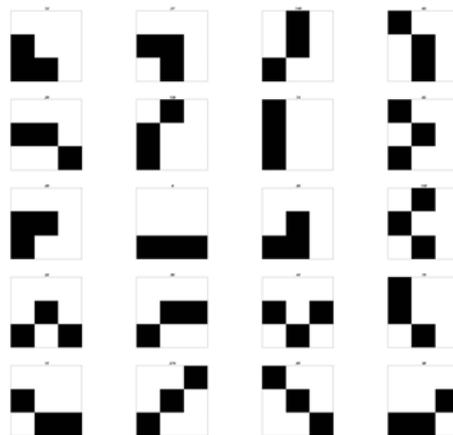


Figure 11: Geometric configurations for three-block mining units.

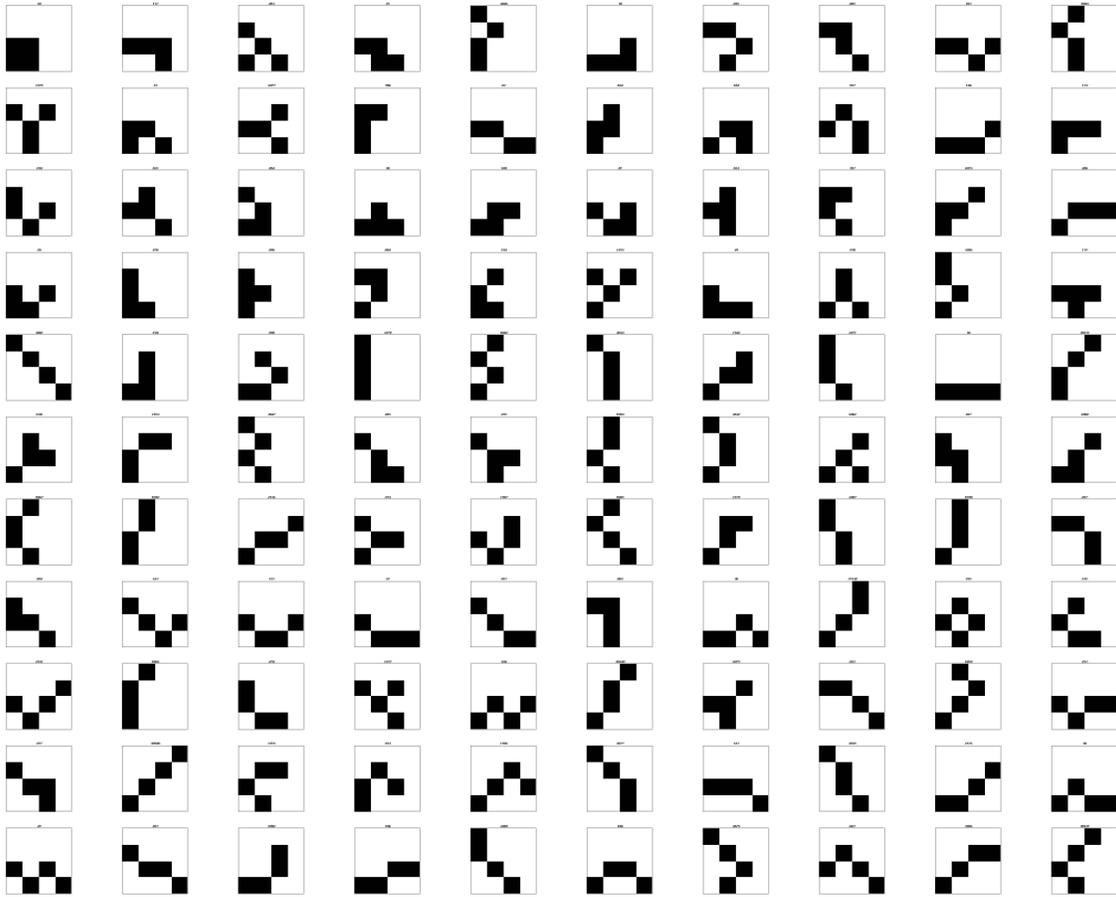


Figure 12: Geometric configurations for four-block mining units.

The index is calculated by generating a square grid that is $n_{\text{blocks/unit}} \times n_{\text{blocks/unit}}$ large. The blocks in the grid are numbered sequentially as shown in Figure 13. We will demonstrate an example of calculating an index for a four-block unit. The same principles apply to any other number of passes. The mining unit configuration we will consider for the example is shown in Figure 14. The first step is to code the grid using 1's and 2's. Blocks belonging to the mining unit (black) are coded as 2's while blocks not belonging (white) are coded as 1's as shown in Figure 15. The index is then calculated according to the following formula:

$$index = 1 + \sum_{i=1}^n (c_i - 1) \cdot 2^{n-1}$$

Where c_i is the grid code (either 1 or 2) and n is the block number. In the case of our example, the above equation works out to be:

$$\begin{aligned} index &= 1 + [2^2 + 2^4 + 2^5 + 2^6] \\ &= 1 + 4 + 16 + 32 + 64 \\ &= 117 \end{aligned}$$

| | | | |
|----|----|----|----|
| 13 | 14 | 15 | 16 |
| 9 | 10 | 11 | 12 |
| 5 | 6 | 7 | 8 |
| 1 | 2 | 3 | 4 |

Figure 13: Numbering of blocks in the grid for index calculation.

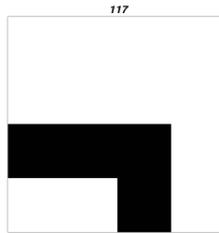


Figure 14: Four-block mining unit configuration used for index calculation example.

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 1 |
| 1 | 1 | 2 | 1 |

Figure 15: The grid is coded according to block color, 1's for white blocks and 2's for black blocks.

The unit index makes it easy to decide if newly created configurations are acceptable. After each random switch, the new unit configuration is compared to the library of possible configurations. If the new layout is found to be allowable, the new value of the objective function is calculated and, if higher than before, the new layout is kept.

Note that every index is calculated assuming a mining direction from lower left to upper right. As previously mentioned, this is not always the direction in which mining is progressing. There are four possible mining directions as shown in Figure 6. To account for this fact, the mining units in quadrants 2-4 must be rotated either 90 or 180 degrees in order to calculate the correct unit index. This is illustrated in Figure 16.

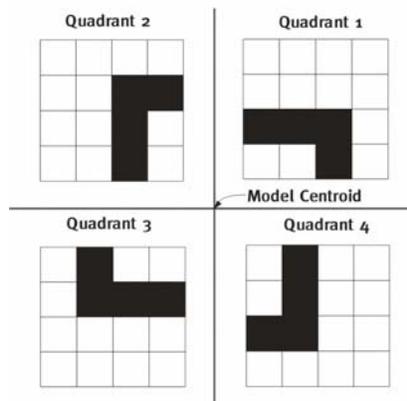


Figure 16: Mining unit indexes are calculated based on the mining direction of the quadrant in which the unit resides. As such, each of these four units has the same index.

As mentioned previously, difficult-to-extract mining units are assigned a penalty. The undesirable configurations have a high penalty; the desirable configurations have a very low or no penalty. The penalties are assigned through the use of a penalty file. There are three separate penalty files, one for each of 2, 3 and 4 block mining units. Each penalty file contains two columns: the first consists of unit shape indexes and the second contains the penalty associated with that index value. The penalty is a fractional reduction in the unit's profit. A portion of the 4-block penalty file is shown in Figure 17. These penalty files can be manipulated by the user by cross-referencing with Figures 10-12 to best determine the penalty each configuration should receive.

```

Valid Combinations for Npass = 4
2
Index
Penalty (fractional decrease in profit)
52      0.0
117     0.05
55      0.05
4626    0.05
72      0.05
835     0.05
805     0.0
181     0.05
8483    0.05
1315    0.05
54      0.05
1077    0.05
786     0.05
61      0.05
562     0.05
102     0.05

```

Figure 17: Penalty file sample from the 4-block penalty file. The first column consists of unit shape indexes and the second column contains the fractional profit reduction penalty associated with each index.

When a unit number switch is made, the two new unit configurations are checked for viability. If the configurations are allowable, the units are assigned the corresponding penalties and the profit is recalculated, adjusting the unit profits by their associated penalties. If the profit is higher, the switch is kept and the process begins again. The objective function for calculating the profit and accounting for the penalty is:

$$P_T = \sum_{i=1} [(1 - C_i) \cdot P_i] - \sum_{j=1} [(1 - C_j) \cdot P_j]$$

where C represents the unit complexity penalty. It is a fraction and decreases the profit of the mining unit.

The process of choosing a random location, switching unit numbers with a random neighbor, and recalculating the profit based on the change will lead to increased delineation of the recoverable reserves. The degree of improvement depends on how many random switches are made and tested. The number times that every block is visited is specified by the user. For our previous example with 100 blocks, we visited every block 10 times, that is, we tried 1000 random switches to optimize the profit. It is recommended that every block in the model be visited at least 5 times. This will give optimal results.

Post-Processing

With an optimized expected profit achieved, the program will proceed in one of two directions, depending on whether a category model was input or not. If a category model was input, the program scans through the optimized profit model and splits any units that cross category boundaries and then proceeds to the classification step. If no category model was used, the program proceeds directly to the classification step. In the classification step, each block in the model is classified as ore or waste. If a block has a positive expected profit and is part of a mining unit that has a positive expected profit, the block is classified as ore. If a block has a negative expected profit and is part of a mining unit with a negative expected profit, the block is classified as waste. There is an option in the program to report lost ore and dilution as well. If a block has a positive expected profit, but is part of a mining unit with a negative expected profit, it is classified as lost ore. If a block has a negative expected profit and is part of a mining unit with a positive expected profit, the block is classified as dilution. The following table summarizes:

| Block Value | Mining Unit Value | Classification |
|-------------|-------------------|----------------|
| > 0 | > 0 | Ore |
| < 0 | < 0 | Waste |
| < 0 | > 0 | Dilution |
| > 0 | < 0 | Lost Ore |

There is also an option to input a schedule model. This schedule model must have been generated previously. It is used to renumber the mining units according to the schedule, typically in the order in which they will be mined.

Real-Life Example

The following shows a profit map generated by post-processing the results of simulation. As previously mentioned, you are encouraged to refer to the guidebook titled 'Guide to Geostatistical Grade Control and Dig Limit Determination' for instructions in producing such a model.

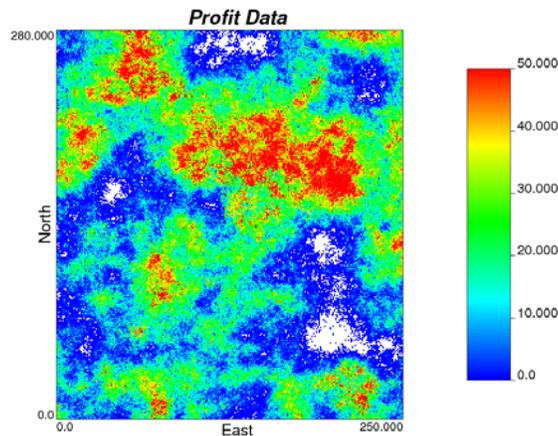


Figure 18: Example profit model. The white blocks are those with profit value below waste mining cost.

The area shown on the profit map contains 280x250 1x1m blocks for a total of 70,000 blocks. The first step in the process is to assign every block a unit number. For this example we will use 4 blocks per mining unit meaning there are 17,500 units.

Once every block has been assigned a unit number, the initial profit can be calculated. The value of each unit is calculated and the sum of all of the positive mining unit values minus the sum of all of the negative unit values is the initial profit. The process of randomly visiting blocks and choosing neighbors can now begin. A block is randomly visited, one of its eight neighbors is randomly selected, and the unit numbers for each are switched. The unit configurations are checked against the index of allowable configurations. If each unit's configuration is acceptable, the objective function is recalculated. If the switch has increased this value, the change is kept and a new block is visited. If the switch has made no improvement to the objective function, the unit numbers are switched back and a new block is visited. As mentioned previously, it is recommended that every block be visited at least 5 times. Different numbers of iterations were experimented with to ensure that optimal results were being achieved. The results of this experimentation are shown below:

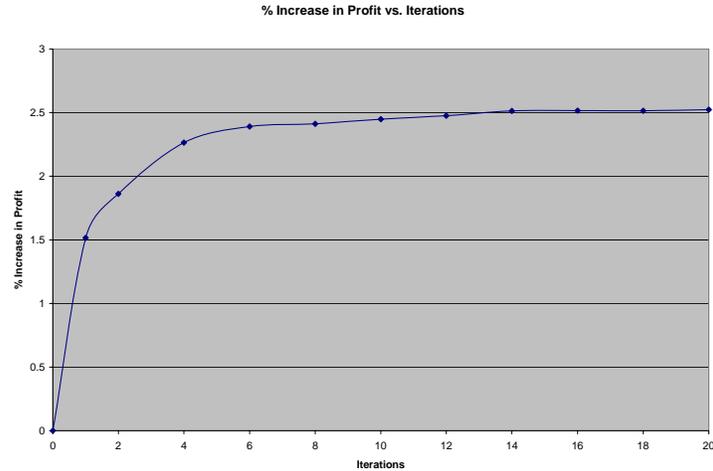


Figure 19: Graph of % increase in expected profit vs. iterations. After approximately 5 iterations, the increase in expected profit becomes small.

After approximately 5 iterations, the increase in expected profit begins to taper off and no significant gains are realized.

Each block can now be classified as ore, waste, lost ore, or dilution by using the units' expected profit. If both the mining unit expected profit and the block profit are greater than zero then the unit is classified as ore. If the expected profit for the mining unit is less than zero and the block profit is less than zero, the unit is classified as waste. The block is classified as dilution when the mining unit value is positive, but the block value is negative. Lost ore is the classification used when the mining unit profit is negative and the block profit is positive. This classification scheme produced the results shown in Figure 20.

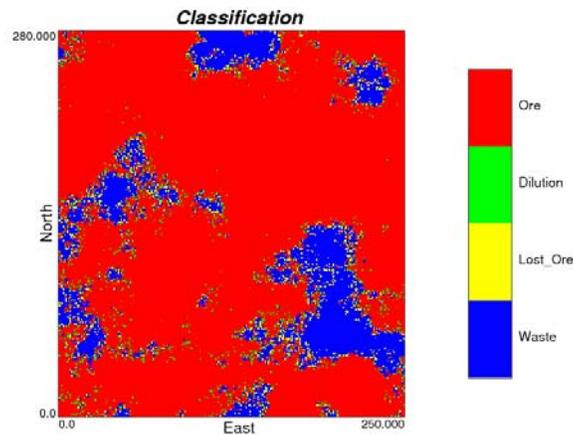


Figure 20: Blocks are classified according to their own expected profit and the expected profit of the unit they fall into.

Example with Categories

An example of FGC utilizing categories is now presented. This is a three-dimensional example with a grid of 250x236x10 and cell size of 10x10x15. A profit model was created following the guidelines presented in the CCG guidebook. Profit values range from -\$2564 to \$272410. This model is shown in Figure 21. The categorical pit stage model has 12 categories or stages and is shown in Figure 22.

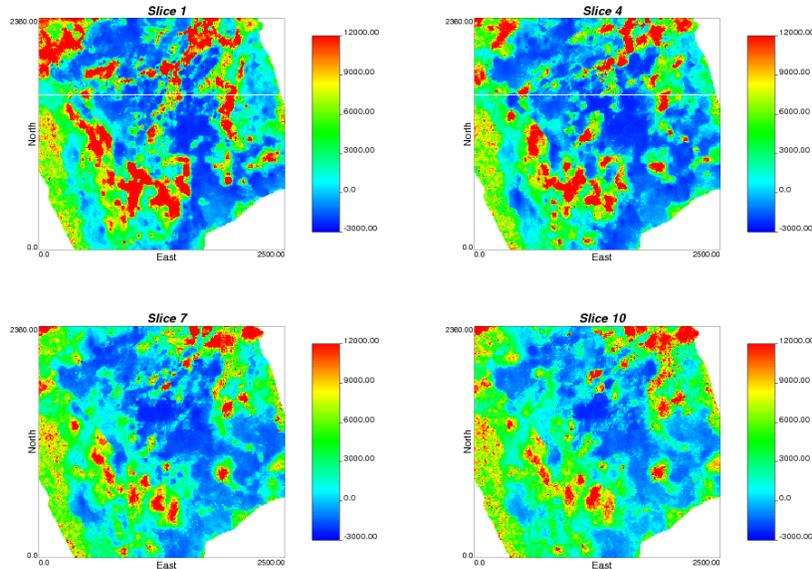


Figure 21: Multiple slices of the 3-D profit model.

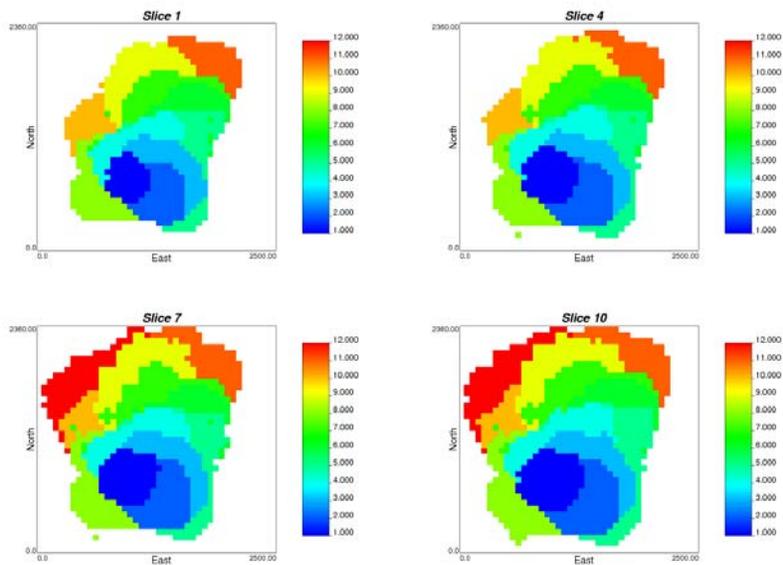


Figure 22: Categorical pit stage model. The centroid of the model is at the centroid of the stage 1 category.

The first step is to group the blocks into their initial mining units. As mentioned, the numbering of the mining units is inconsequential. The initial profit can then be calculated. The optimization process is then begun by randomly visiting the blocks in the model, choosing a random neighbor, switching the unit numbers, and recalculating the profit. This is done until every block in the model is visited at least 5 times, or as specified by the user. In this way the optimum accumulation of blocks to mining units is found.

The optimized mining unit model is then checked to see if any of the mining units straddle a category boundary. If such a situation is found, the mining unit is split so as to be kept within a single category.

The final step is to classify every block in the model according to its profit and the profit of the mining unit it falls within. If both profit values are positive the block is ore, if the unit value is positive and the block value is negative the block is dilution, if the unit value is negative and the block value positive the block is

lost ore, and if both profit values are negative the block is waste. The classifications for this example are shown in Figure 23.

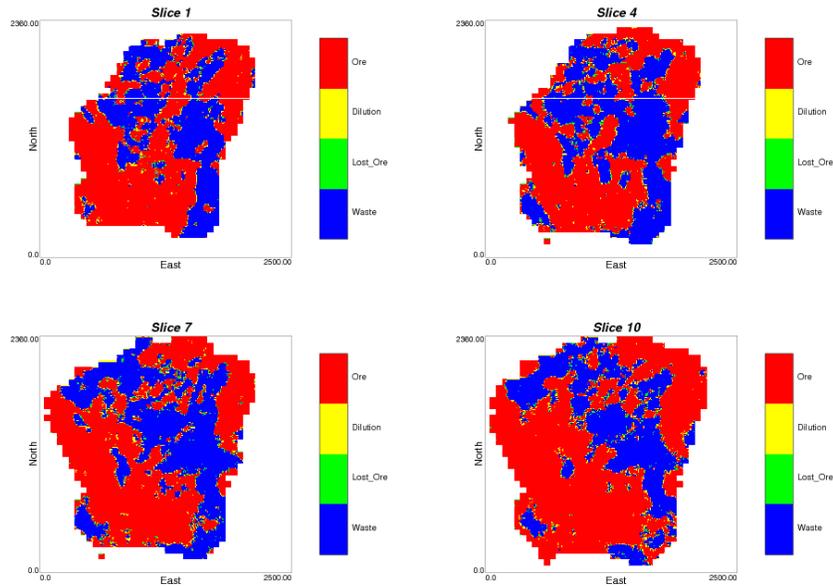


Figure 23: Final classifications produced by FGC.

Conclusions

The common practice of manually creating dig limits for multiple benches and multiple realizations cannot be realistically applied to feasibility models. The professional time involved is prohibitively large. Performing Feasibility Grade Control (FGC) is a practical alternative. The absence of time consuming manual input and the efficiency with which multiple realizations and benches can be processed make it a reasonable solution for performing grade control at the feasibility stage.

References

- Deutsch, C.V. and Journel, A.G. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, New York, 2nd Edition, 1998.
- Neufeld, C.T., Norrena, K.P. and Deutsch, C.V. *Guide to Geostatistical Grade Control and Dig Limit Determination*. Centre for Computational Geostatistics, Edmonton, Guidebook Series Vol. 1, 2005.