THREE DIMENSIONAL MODELLING OF ORE-BODIES
USING INTERGRAPH CAD/CAM SYSTEM

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Ashok A. Narain
November 1984
The author wishes to express his sincere thanks to Professor P. K. Chatterjee, whose valuable guidance and constructive criticisms made this study possible.

The author would also like to thank Professor J. Gunasekera for his help in reading tapes and making available the pertinent Intergraph CAD/CAM manuals.

Finally, the author is grateful to Professor D. E. Scheck for his suggestion and encouragements.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>iv</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Statement of Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Approach</td>
<td>3</td>
</tr>
<tr>
<td>1.4</td>
<td>Significance</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>LITERATURE REVIEW</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>WHY GEOSTATISTICS ?</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>What is Geostatistics ?</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>Variogram</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>Block Variance</td>
<td>21</td>
</tr>
<tr>
<td>3.4</td>
<td>Estimation Variance</td>
<td>22</td>
</tr>
<tr>
<td>3.5</td>
<td>Kriging</td>
<td>23</td>
</tr>
<tr>
<td>3.6</td>
<td>Significance of Geostatistics</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>DESCRIPTION OF THE PACKAGE</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>27</td>
</tr>
<tr>
<td>4.2</td>
<td>Design Criteria</td>
<td>27</td>
</tr>
</tbody>
</table>
4.3 Implementation of Design Criteria

4.3.1 Development of a data base of drilling information

4.3.2 Use of Geostatics to establish tonnage and grade estimation

4.3.3 Use of interactive CAD/CAM facilities for the development of three dimensional model

4.3.3.1 Alternatives

4.3.3.2 Description of 3-D software

4.3.3.3 Subroutine DRAW_CELL

4.3.3.4 Subroutine MAIN_PROGRAM

4.3.3.5 Subroutine INDFPI

4.3.3.6 Subroutine CLDFPI

4.3.3.7 Subroutine DRAW_SKIN

4.3.3.8 Subroutine SHDFPI

4.3.3.9 Subroutine DEDFPI

4.3.3.10 Subroutine CHANGE_SYMBOLOGY

4.3.3.11 Subroutine NEW_FILE

4.4 Model Verification

5 DISCUSSION OF RESULTS

5.1 View Setup

5.2 Hidden line Removal

5.3 Shading

5.4 Coloring
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The Spherical Variogram</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>Flowchart showing various constituent parts of Geostatistical package including THREED</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>Sample of input raw data to routine SURVEY</td>
<td>33</td>
</tr>
<tr>
<td>4.3</td>
<td>Sample output of routine KRI3D for input to routines THREED and ULTPIT.</td>
<td>35</td>
</tr>
<tr>
<td>4.4</td>
<td>Scheme for joining data points using straight lines</td>
<td>37</td>
</tr>
<tr>
<td>4.5</td>
<td>Scheme for joining data points using shapes</td>
<td>37</td>
</tr>
<tr>
<td>4.6</td>
<td>Scheme for joining data points using cubes</td>
<td>37</td>
</tr>
<tr>
<td>4.7</td>
<td>ISO View of the ore-body using 'Shape' approach ( after hidden line removal )</td>
<td>41</td>
</tr>
<tr>
<td>4.8</td>
<td>ISO View of the ore-body using 'cube' approach ( after hidden line removal )</td>
<td>42</td>
</tr>
<tr>
<td>4.9</td>
<td>Block diagram of constituent parts of THREED</td>
<td>43</td>
</tr>
<tr>
<td>4.10</td>
<td>Conceptual data file for constructing 3-D model</td>
<td>54</td>
</tr>
<tr>
<td>4.11</td>
<td>Conceptual data file for Case 1</td>
<td>62</td>
</tr>
<tr>
<td>4.12</td>
<td>Conceptual data file for Case 2</td>
<td>63</td>
</tr>
<tr>
<td>4.13</td>
<td>Conceptual data file for Case 3</td>
<td>64</td>
</tr>
<tr>
<td>4.14</td>
<td>Actual data file for Case 1</td>
<td>66</td>
</tr>
<tr>
<td>4.15</td>
<td>Actual data file for Case 2</td>
<td>67</td>
</tr>
</tbody>
</table>
4.16 Actual data file for Case 3

4.17 THREED output for case 1

4.18 THREED output for case 2

4.19 THREED output for case 3

5.1 Twin screen of Intergraph holding eight different views

5.2 View set-up as used in THREED

5.3 3-D model of the ore-body as seen on left screen of the work station

5.4 Isometric view of ore-body as seen on right screen of the work station

5.5 Drawing coordinate system

5.5a View coordinate system

5.6 Iso view of the ore-body as seen on right screen (after hidden line removal)

5.7 Ore-body from side looking top and south (with shading)

5.8 Ore-body turned upside down and looking from top (with shading)

5.9 Another view of ore-body turned upside down and looking from top (with hidden lines removed)

5.10 Ore-body with a contour slice of richer grade (> 3.00%)

5.11 Ore-body as seen on the left screen of work station developed without kriging the bore-hole raw data (with hidden lines removed)
CHAPTER : 1

Introduction

1.1 Statement of the problem

Sublevel open stoping, sublevel caving and block caving are large scale mining methods that are widely applied to underground hard rock mining. The success of these methods may be attributed to the high degree of mechanization, their adaptability to varying design demands and the low cost of mining that can be achieved through their use. However, as with other large scale mining methods, these methods can give rise to severe ground control problems with increasing mining depths. It is, therefore, important that careful planning and design of these large scale extraction methods be conducted before actual production is initiated. Operational constraints in underground hard rock mining, such as available manpower, time, make it almost impossible to generate alternative design layouts for comparative evaluations. However, with recent advances made in computer technology it is now possible to examine different mine layouts efficiently and swiftly, thus making the selection of a better layout possible.
One of the difficulties associated with planning an efficient excavation system in an underground mine is that of visualizing the complex shape of the ore body. It is generally acknowledged that the design of excavation systems would be much simpler, more effective and less labor intensive if a three-dimensional view of the ore-body could be presented to the design engineer. The desirability of a three-dimensional perspective of the ore-body is indicated by the fact that a number of such drawings are produced manually in spite of the laborious drafting procedures. Such three-dimensional representations by conventional methods cannot be updated easily as the work progresses. Moreover, if another view is required, the entire drafting process has to be repeated. The excessive tedium associated with manual drafting of three-dimensional shapes of the ore-bodies can be overcome with computer-aided drafting.

1.2 Objectives

The objective of this study was to develop a software package for three-dimensional modelling of an ore-body from bore-hole data using Intergraph CAD/CAM techniques. This model can be dynamically rotated for viewing from any angle in three-dimensional space for a subsequent
interactive stope design. This capability enables the design engineer to apply extraction techniques more critically. To obtain a precise definition of the ore-body, the raw bore-hole data was validated by a set of geostatistical routines for variogram calculations and kriging.

1.3 Approach

The college of engineering recently acquired an Intergraph CAD/CAM machine. Since few attempts had been made to interface the graphics capabilities of the workstation with FORTRAN routines, little expertise was available in this area at the time when this study was initiated.

In view of these factors, a significant amount of time was spent trying to overcome machine difficulties and to gain familiarity with the available facilities, especially with regard to the Intergraph software interface. Interactive Intergraph CAD/CAM techniques were then successfully implemented for developing three dimensional models.

There were three other distinct phases in the modelling of the ore-body in this study:
1) **Development of a data base of drilling information**

To ensure the completeness and self supporting nature of the package, the development of an integrated bore-hole database was considered an important feature of the system. Facilities have been included for this database to be created either from the original bore-hole information or from a mine's bore-hole database. Information such as the grades, drill hole direction, and location can be extracted from this database to interface with the geostatistical routines and the CAD/CAM application software.

2) **Use of geostatistics to establish a tonnage and grade estimation procedure**:

The information obtained from the bore-hole data base is used by the geostastical program to obtain a variogram of the grade distributions. The entire ore-body is divided into blocks of a given size and each block is kriged to obtain the kriged estimate of the grade.

3) **Use of interactive CAD/CAM facilities for the development of a three dimensional model**:

Information from the bore-hole database and the geostatistical routines is stored in a data file. The design engineer can access this information and peruse the
drilling data to ascertain whether a drill hole intersects the ore or the waste. Inspection of these grades and drill hole locations provides information to help delineate the ore from the waste zone. The Intergraph CAD/CAM application software is then employed to convert this exploratory drilling information into a three dimensional model of the ore-body. The program uses primitive elements such as shapes (a shape is a multisided polygon) or cubes to construct the model. Once the skin of the ore-body is established, it can be rotated in three dimensional space and viewed from any desired angle (with or without hidden line removal) by using the graphic capability of the Intergraph work station.

1.4 Significance

A volume model, as developed here, will enable the planning engineer to implement the plans as though it were a physical model. The capability to rotate and view the ore-body from a three dimensional perspective will greatly enhance the design engineer's ability to visualize the ore-body and recognize the consequences of design changes. Designing stopes will be less onerous as the engineer will be able to interactively define the outer skin of the stopes and plan their excavation. The inclusion of
Geostatistics will ensure that the estimation of metal values and tonnages inside a three-dimensional space will be calculated with a known degree of accuracy. This integrated software provides the design engineer with a tool to design a more efficient mine in a very cost-effective manner and within a time frame which has not been hitherto possible.
CHAPTER : 2

Literature Review

The mineral industry has been continuously striving to expand the use of the computer to process exploration, geological and mining data. Kaas (1969) emphasized the great potential of computer applications in mining and discussed some of the problem solving approaches of digital plotting and computer graphics. He reported the increasing use of digital plotting programs and computer graphics for lowering the cost and the time required to produce many types of maps and cross sections commonly used by mining companies. He stressed that computer graphics has greatly facilitated the use of computers to solve a large number of problems in the mining industry.

Computer modelling of ore bodies is no longer a novel technique, and two-dimensional models of ore bodies are being widely used. The development of three-dimensional models, however, has been disappointingly slow. With the advent of CAD/CAM and recent advances made in computer graphics, it appears that many of the difficulties encountered in the past are no longer a major obstacle.

Dowd (1975) presented an interactive graphics procedure for the estimation of ore reserves. The method
used is that of kriging which involves finding the minimum variance unbiased linear estimator. The design engineer can interactively select blocks to be estimated, change sampling patterns, add samples or even change the assumed geological characteristics of the deposits. The author identified many other areas in the mining industry, apart from the one discussed above, where graphic display systems could be employed to increase productivity.

Myers, et al. (1980) did a case study of the graphic aspect of an ore reserve system implemented by the management services division of Gold Field Mineral Development Co., South Africa. In this study they discussed the numerous advantages of using computer graphics in actual mining applications.

A slightly different kind of application of computer graphics in the mineral industry was reported by Marlow and Wall (1979). They described an interactive linear regression graphics package and included a practical application. The advantages and the applications of the package to the development of a mathematical model for hot rolling mill sequences have been discussed.

Recently, a number of works have been reported involving the use of interactive graphics for designing, modelling and reserve estimation in the coal mining industries.
Coal face machinery cost many millions of dollars and involves the use of components from many manufacturers. In addition to problems of selecting compatible equipment, meeting mining constraints and the incorporating new production techniques, into the plan, the many available alternatives present opportunities to establish production schedules that will minimize the cost of operations. Ranson and Daniel (1980), in a study conducted by Great Britain's National Coal Board, reported on some of these problems and suggested ways to solve them. The paper included the history and development of mathematical modelling with graphics. The choice of the software, its capabilities and possible the future applications within the industry have been also discussed.

In another recent study conducted within the National Coal Board, Hartley and Ranson (1984) reported significant progress in developing application programs for surveying, geological, modelling, planning and engineering of coal deposits. The coal face design and reserve computations using CAD are among the many other capabilities of the software.

Roselman (1983) reported a mine planning system for coal and base metals which was in use at Rand mines and which was later modernized to include facilities to generate computer graphics. The system provides a
computerized base for geological data that is used for long and short term planning at the existing mines. It is standardized on IBM (or compatible) hardware and resides on a main frame. It uses economical IBM 3279 Graphic screens as the standard terminal. The low cost of these terminals allows a large number of users to be provided with graphics work station. Graphic capabilities are being employed for geostatistical data analysis, contouring and cut-diagram generation.

Ley and Welborn (1982) discussed a system called ENCORE. This package was designed for use by the geologist or mining engineer with little or no computer background. ENCORE is specifically written for the development of coal ore-body models and the reserve estimation. Geological data is stored on a large mainframe computer and is accessed from a local work station for editing, manipulating and producing various reports and high quality engineering maps. ENCORE provides the geologist or mining engineer with the capability to select the appropriate extraction methodology for specific ore-body which is being analyzed.

Barua and Kim (1984) discussed a system in which the STPM (i.e., short term coal quality prediction model) is used in conjunction with an interactive graphics program MPIG (i.e., mine planning with interactive graphics). The STPM model is essentially a gridded coal seam model
consisting of predicted grades at each grid point in the model. The predicted grades are assigned using the geostatistical technique of (linear) kriging. Once a model using STPM is created, the MPIG program displays the model values on a graphics screen as grade contour lines. The interactive feature of MPIG allows its user to mine any portion of the model on the screen by using a terminal cursor or a joystick.

Arcamone, Dejean and Sayettat (1984) reported a system called PLAMIMET. The objective of the PLAMIMET system is to provide mine operators with a CAD system which integrates the geology of the mine block under investigation, its geometry, and existing or future workings. The paper discusses the main stages in the development of PLAMIMET and its practical application in mining coal deposits.

The application of computer graphics is quite common in surface mining and open pit planning. Naude, Heyn and Kotze (1977), reported a comprehensive mine planning system called ISCOR. It is an integrated system which covers geological exploration, ore-body modelling, open pit planning and production scheduling. An important feature of the system is that the user has full interactive control over each function through the use of a graphics terminal.

In an unpublished report Chatterjee and Jalkanen
(1983) described a comprehensive package SURFMIN for surface mining applications. The package consists of a number of interactive computer programs which employ geostatistical techniques to obtain grade estimations and to generate a pit outline. It also includes facilities for scheduling the production within a final mine skin.

Not much work has been reported in the modelling of the ore-body in the area of hard rock mining. Digitizing has been the most common method for developing three dimensional models. One of the earlier works done on three dimensional modelling by computer graphics has been reported by Notley and Wilson (1975). It describes a comprehensive system for producing drawings of mine workings by a computer-directed digital plotter. Standard mine plans with elevation information are digitized, using a special technique to generate an ordered set of three dimensional coordinates and line status codes. The three dimensional structure is projected on a plane from a specified direction and distance. This system has the capabilities to produce any desired type of mine drawing with complete hidden line removal. Although, this system was originally conceived to visualize underground mine workings, it can also be applied to open pit operations.

Allen and Van Barneveld (1977) reported an interactive graphics system that uses CAD/CAM. This
system was developed to assist the mine design engineer to optimize his design. The system can assist the engineer with the ore-body visualization, initial design and optimization phases of his design. At any stage the engineer may, at his discretion, requests a display of gridded mesh data, digitized ore outlines or existing stopes or developments (drives, cross-cuts). It is possible to view this information in plan or any vertical section or in three dimensional that can be rotated. Tonnage and grade calculations can be quickly performed for the initial stope outlines and for each stope modification during the optimization phase. The implementation and utilities of the facilities in the system have also been discussed.

Harris (1976) in an unpublished report, presented the actual implementation of a totally integrated software MES (Mine Engineering System) in underground mining. The system capabilities include modelling of the ore-body, ore reserve estimation at any stage of mining, interactive stope design and other routine mine design activities. MES is built around the concept of pictorial data transfer. An advanced graphics system (LDS-2) forms the core of a graphic work station which also does the dynamic rotation in a three dimensional space.

Cox and Johnson (1983) discussed the problems
associated with technical computing in the mining industry and examined the way in which distributed graphics processor (DGP) could be employed to resolve them. They stress that DGP offers more a natural means of interaction with complex technical models.

Use of color graphics in mining software has been rather limited. Johnson (1983) has given an outline of the current use of graphics in mining software and development of interactive applications using high level color graphics terminals.

The notion that CAD/CAM has great potential to significantly increase the productivity within the minerals industry is further strengthened by a paper presented by Sharp and Douglas (1982). This article describes a graphics station and graphics software designed specifically for use within the minerals industry. By using the graphics station the operator is able to recall maps for modification, display them on a color CRT, send the work to a plotter and restore the work when finished. The engineer may use the system for storing data, performing engineering calculations, word processing, and performing general file maintenance. In addition the system may be used to communicate with a mainframe computer for such purposes as transmission of large data files, graphics, three dimensional rotation of a mineral deposit or mine
design, coloring of special data, and other data processing work.

A review of these papers suggests that while application of computers for modeling of an ore body is commonplace for coal deposits and in surface mining, limited efforts have been made in this area in underground mining, especially with regard to porphyry deposits where the shape of the ore-body is complex. Moreover, not much attention has been paid to the problem of geostatistically validating the bore-hole data and this has restricted the precision of the models of ore-bodies. The proposed package aims to provide the design engineer with such a tool. Moreover, the speed and ease with which an Intergraph system does dynamic rotation, and hidden line removal makes the proposed software package unique.
CHAPTER : 3

Why Geostatistics ?

3.1 What is Geostatistics ?

Once an ore deposit has been located, various decisions must be made regarding its economic viability. Any decision will depend upon a detailed knowledge about quantity (tonnage), quality and location of ore within the reserve. This information has to be inferred from a finite number of observations that are derived from drill hole logs. There is a definite need for some mathematical techniques to enable one to interpret the data, make predictions about the prospects of the deposit, and quantify the reliability of one's prediction in terms of confidence levels.

Geostatistics provides such a tool. In the early 1960's, after much empirical work by authors in South Africa, George Matheron (France) published his treatise on the theory of regionalized variables. The application of this theory to problems in geology and mining has led to the more popular name geostatistics. Geostatistics provides a rational theoretical basis for quantifying many statistical relations (which could be recognized but not quantified by classical statistical techniques and other
mathematical methods such as spatial correlation, range of influence, random behaviour and reliability of estimates.

Classical statistical techniques assume that all of the samples are taken randomly and independently from one simple probability distribution and that all points in the deposit can be thought of as samples from that single distribution. This assumption is called stationarity. It does not take into account the actual sample position or relationships between samples. The assumptions of geostatistics, however, are not so restrictive. While, it accepts the concept that each point in the deposit represents a sample from some distribution, it also recognizes that the distribution at any point may differ completely from other points in the deposit in terms of its mean, variance and form. However geostatistics does assume that difference in grade between two sample points depends only on the distance between, and orientation of, the points. It does not depend on whether ore is in a rich or poor zone or on the actual sample values. In other words, difference in grades must be consistent, not constant over the deposit. This assumption is sometimes referred to as the quasi-stationarity or intrinsic hypothesis.

The main "tools" of geostatistics are the variogram,
block variance, estimation variance, and kriging. Each of them is briefly discussed below.

3.2 Variogram

The variogram is used to describe the spatial correlation between grades or any other characteristics, such as thickness of mineralization within an ore deposit and is the basic tool of geostatistics. It incorporates several geological features that are important in ore deposit evaluation such as the continuity of mineralization, zones of influence and also can be used to assess whether the zones of influence are different in different directions.

Consider two numerical values $Z(x)$ and $Z(x+h)$, which denote the grades at two points $x$ and $x+h$ separated by a distance $h$. The variability between these two quantities is characterized by the variogram function $2\gamma(h)$, which is defined as the expectation of random variable

$$2\gamma(h) = E_{i,j} \left[ Z(x) - Z(x+h) \right]^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[ Z(x_i) - Z(x_i+h) \right]^2$$

where 'n' is the number of experimental pairs $Z(x_i), Z(x_i+h)$ of data separated by the distance 'h'.
While a calculated experimental variogram may help in determining the structure of a deposit and the behavior of grade variations, it is purely a data summary technique which describes the behavior of only sample values. Conclusions about the whole deposit, if they are desired, must be arrived at by a process of inference. The process is analogous to the process of constructing a histogram from sample values and then inferring from the histogram a theoretical distribution for the whole deposit. Thus, if an experimental variogram has been constructed, it must be related to some theoretical model if conclusions are to be drawn or estimation made for unsampled areas in the deposit.

Just as there are few distributions available to the practitioner of classical statistics, there are only a few simplistic models for theoretical variograms such as the linear, spherical and exponential models.

Of these models, only the spherical model will be discussed here for two reasons.

1) The model is applicable to about 80 percent of ore-bodies.

2) The computer programs used in this package assumed a spherical model.

The spherical model shown in Figure 3.1 is characterized by three parameters. The nugget effect, $C_0$, 

$3.1$
Figure 3.1 The Spherical Variogram
if non-zero is an expression of the discontinuity of the model for a distance \( h = 0 \) and originates from two possible sources. One source may be the presence of nested structures on different scales and the other arises from sampling and assessing errors.

The sill, \( C^0 + C^0 \), is the limiting value of the variogram and is equivalent to the sample variance of the grades in the deposit. The distance, at which the variogram reaches the sill, is called the range of influence, \( a \). Points further apart than distance 'a' are unrelated or independent of one another. The variogram is uniquely defined by the nugget effect, \( C^0 \); sill, \( C^0 + C^0 \); and range, \( a \).

\[
\gamma(h) = \begin{cases} 
C^0 + C^0 & \text{if } h < a \\
0 & \text{if } h \geq a
\end{cases}
\]

\[
\gamma(h) = C^0 + C
\]

The spherical variogram model is particularly useful because it incorporates and graphically portrays many of the important geological concepts related to mineralization that have previously been intuitively recognized but not quantified. The homogeneity of the regionalized variables in three dimensional space can be investigated by computing variograms in several directions.

### 3.3 Block variance

If the mineral deposit is split into blocks of
identical size, shape and orientation, the variance of the grades of these blocks in the deposit is called the block variance. As the blocks get larger their variance get smaller because the grade of a larger block is an average of many smaller blocks. If the block size correspond to mining units, the block variance provides information on the variability of each mineable block's grade.

3.4 **Estimation variance**

Estimation variance refers to the error between the true grade and the estimated grade of a block. Its magnitude depends on the characteristics of the mineralization, the variance of the samples, the sampling grid and the estimation method.

Consider the simplified situation shown below, where the sample $S$, is being used to obtain an estimate for the average grade of block $V$.

$$ i.e. \ Z_\mathbf{v} = Z(S) $$

The estimation variance of this estimate can be found in any book on geostatistics and is given by

$$ \sigma_{E}^{2} = 2 \tilde{\gamma}(S,V) - \tilde{\gamma}(V,V) - \tilde{\gamma}(S,S) \quad (1) $$
where,

\[ \gamma(S,V) \] is the average value of the variogram computed between all the points in the sample and all points in the block.

\[ \gamma(V,V) \] is the average variogram value computed between all points in a block otherwise known as block variance.

\[ \gamma(S,S) \] is the average variogram computed between all points in the sample \( S \).

The arithmetic mean is often known in geostatistics as an extension estimator and the above variance is referred to as the extension variance.

It is clear from equation (1) that reliability of the estimator depends on three quantities: the relationship of the samples to the area to be estimated; the relationship amongst the samples; and the variation of grades within the area being estimated.

3.5 Kriging

Kriging is a geostatistical technique of estimating the grade of a block as a linear combination of the available samples in or near the block such that the estimate is unbiased and has minimum variance. Simply stated, kriging is a technique to find a
set of weights that minimizes the estimation variance according to the geometry of the problem and character of mineralization. Kriging not only assigns low weights to distant samples and vice versa, but also takes into account the relative position of the samples with respect to the block and to each other. Kriging becomes particularly attractive when compared to other techniques because it not only provides unbiased and minimum variance estimates but it also provides an estimation variance for each block kriged i.e. confidence limits can be placed on each block grade.

In the more common and complicated situations, where several samples are used to obtain a linear unbiased estimator, which is given by

$$Z^* = \frac{1}{n} \sum_{i=1}^{n} \lambda_i Z(S_i)$$

The associated estimation variance is given by

$$\sigma^2 = 2 \sum_{i=1}^{n} \lambda_i \tilde{r}(S_i, V) - \tilde{r}(V, V) - \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j \tilde{r}(S_i, S_j)$$

Weights can be adjusted in the above equation so as to give minimum estimation variance under the constraint imposed by the non biasing condition,

$$\sum_{i=1}^{n} \lambda_i = 1$$

The above optimization problem is solved by the mathematics of Lagrange multipliers, and the estimate
derived from the set of weights \( \lambda_i \) that gives the minimum estimation variance, \( \sigma_e^2 \), is known as a kriged estimate and is called the kriging variance. Kriging always calculates the best estimate regardless of the relative sample size and block locations. The input requirements are the variogram model parameters, data points and block coordinates.

3.6 **Significance of Geostatistics**

A comparative study of geostatistical estimation techniques using kriging over conventional methods has clearly demonstrated the superiority of the former (Kim & Muller, 1978) in terms of precision. Geostatistics has special appeal to the geologist or to a mining engineer because it explicitly recognizes the existence of spatial relationship and provides an estimate of this relationship in quantitative terms.

The usefulness of geostatistics is not restricted to the ore reserve estimation process. The concept of estimation variance and block variance are also very useful—the former quantifies the errors in any ore reserve estimation procedure, while the latter quantifies the so-called volume variance relationship. Geostatistics also includes some other highly advanced mathematical techniques.
such as conditional simulation and disjunctive kriging. All of these tools are extremely useful for maximizing the information that can be drawn from the drill hole data.
CHAPTER : 4

Description of the package

4.1 Introduction

As discussed earlier, the aim of this study is to provide the design engineer with a software package which will construct a precise three dimensional model of the ore-body from kriged bore-hole data. The model is to be developed using Intergraph CAD/CAM techniques which enable the design engineer to rotate and view the ore-body in three dimensional space for subsequent interactive stope design.

4.2 Design criteria

The design criteria for the proposed package are as follows:

1) A typical user of this package will be familiar with the geostatistical terms such as sill, range, kriging and nugget effects but the software must not require any computing expertise on the part of the user.

2) The software must be user friendly and highly interactive so that the user will have control over the execution of the package including the freedom to
select the output device.

3) As a consequence of the above criteria the user must have the option to review the input data and change it, if necessary, before executing the program.

4) The software must be robust so that any absurd input is ignored and the user is instructed to correct the input.

5) The definition of the ore-body must exclude all zones that are outside the specified cutoff grade(s).

6) Data retrieval must be easy for both the user and the program.

7) The software must be structured so that additional features can be incorporated without revising the entire program. This requirement is only to ensure that additional software for mine design, such as scheduling, ring-design or fan and diamond drilling can be integrated into the existing program.

4.3 Implementation of design criteria

The three distinct phases of modelling an ore-body are: 1) developing a data base 2) interfacing a set of geostatistical subroutines and 3) generating a three dimensional model. The software for the first two phases
was already available (Chatterjee and Jalkanen, 1983) but not compatible with the existing configuration of the VAX computer. This software was modified to make it compatible with the system and was then integrated with the ore-body modelling software that uses CAD/CAM techniques.

The geostatistical package will be discussed very briefly, since it was only modified for compatibility with the system. The third phase of modelling will be described at length. Some of the capabilities of the geostatistical package are outlined below:

--- Edits, lists and sorts the raw data after calculating the three dimensional coordinates of all samples.

--- Carries out statistical analysis on the sample data, including the computation of semi-variograms.

--- Subdivides the deposits into a three dimensional matrix of bench blocks according to the size specified by the user, assigns to each block the value of the relevant parameters such as grades and rock types, and generates a pit outline from the specified parameters.

--- Calculates for each area the tonnage and grade of mineralization within the specified cutoff grades.

--- Permits the scheduling of production.

A Block diagram of the system for three-dimensional modelling of an ore-body is shown in Figure 4.1. ULTPIT and
Fig. 4.1 Flowchart showing constituent parts of Geostatistical package including THREE_D
CASHFLOW in this diagram have been shown in rectangular dotted boxes to indicate that the graphic features of these programs are not compatible with the VAX computer and modifications are needed to make them executable on this machine. Routines inside the dotted oval boxes must be developed to make the package complete. In the following pages, the three phases in the development of the ore-body model are discussed under separate sub-sections.

4.3.1 Development of a data base of drilling information

The objective of delineation drilling is to provide a data base for tonnage and grade estimates. The reliability of the grade estimation increases with the accumulation of drilling information. The creation of a data base not only facilitates the recording of this information, but also ensures the self-supporting nature of the package. The software creates a data base from the original exploratory drilling information. Routine 'SURVEY' is a data base program that classifies DDH (Diamond drilling hole) logs into a form suitable for further processing and usage. The routine considers the X, Y and Z information starting from the collar of the DDH and processes the downhole survey information following the DDH track. Assay values along with various geological classification fields are included
with each sampled interval. The final output of SURVEY is written to two sequential access files. The first file contains information concerning the mid-point sample in a form suitable for analyses by various statistical routines such as MULTX and GAM. The second file contains end-point sample information in a form suitable for other statistical routines such as 'CLAS'. Currently the program 'SURVEY' can handle eight variables simultaneously. Figure 4.2 shows a sample of the raw data (as echoed on the terminal) that is used by routine SURVEY for creating the data base.

4.3.2 Use of geostatistics to establish a tonnage and grade estimation procedure

The information obtained from the bore-hole data base is used by the geostatistical routines to obtain the grade estimation over the entire ore body. These geostatistical routines permit variogram calculations and kriging of block grades. Information obtained from diamond drilling holes for a given area is then employed to compute the variogram. The variogram emphasizes the continuity of mineralization and indicates the variation of the ore-body characteristics in the vertical and horizontal directions. The entire ore body is divided into blocks of a given size and each block is kriged to obtain the block and regional kriged estimate of the grade.
DDH LOG - NARAIN PROJECT
COMPLETE DATABASE LISTING

INPUT SURVEY DATA
=================

ECHO OF INPUTTED SURVEY DATA FOR DOH 1

DATE STARTED: 1-1-82  DATE FINISHED: 1-1-82
CORE SIZE: AQX  SURVEY TYPE: KUSTER
TOTAL BOREHOLE LENGTH: 120.00
COLLAR COORDS. * XC = 20.00 YC = 380.00 ZC = 0.00

SAMPLE #: 1 FROM 0.00 TO 24.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 14.3986: 9.3630

SAMPLE #: 2 FROM 24.00 TO 48.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 11.7376: 8.2412

SAMPLE #: 3 FROM 48.00 TO 72.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 14.1405: 9.2316

SAMPLE #: 4 FROM 72.00 TO 96.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 12.9438: 9.6575

SAMPLE #: 5 FROM 96.00 TO 120.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 12.9438: 9.6575

ECHO OF INPUTTED SURVEY DATA FOR DOH 2

DATE STARTED: 1-2-82  DATE FINISHED: 1-2-82
CORE SIZE: AQX  SURVEY TYPE: KUSTER
TOTAL BOREHOLE LENGTH: 120.00
COLLAR COORDS. * XC = 20.00 YC = 380.00 ZC = 0.00

SAMPLE #: 6 FROM 0.00 TO 24.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 14.3986: 9.4741

SAMPLE #: 7 FROM 24.00 TO 48.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 12.9724: 11.2785

SAMPLE #: 8 FROM 48.00 TO 72.00 BEARING 0.00 DIP 90.00
ALTERATION IS: UNKNOWN LITHOLOGY IS: MASSIVE
ASSAY FIELDS: GOLD: SILVER: ARSENIC
***********: 0.0100: 9.4679: 12.7217

Fig. 4.2 Sample of input raw data to routine SURVEY
Routine CLAS creates three direct access files that are required as input for kriging routine KRI3D. On request from the user, CLAS also creates a listing which can be stored on the disc or displayed on the terminal or line printer.

Program KRI3D computes the kriging estimation of the parallelepipedic units. This program uses the three-dimensional drill-hole data that was previously sorted by the routine CLAS. The output from the KRI3D program contains the mean grade and the variance of each block for each variable. A sample of the output is shown in Figure 4.3. Columns 1, 2 and 3 contain information about the location of the block in terms of its position with respect to the kriged area. Columns 4 through 6 contain the kriged estimate of the grade of that block (the raw data contained only three variables for this run). Columns 7 through 9 contain the block variances for the mean estimates in columns 4 through 6 respectively. For further information on routines CLAS and KRI3D, the reader is referred to 'Mining Geostatistics' by Journel, 1978.

4.3.3 Use of interactive CAD/CAM facilities for the development of three dimensional model

Routine THREE_D accesses the information generated by
Fig 4.3  Sample output of routine KRI3D for input to routines THREED and ULTPIT
KRI3D and scans the grade of each block to isolate the ore zone from the waste zone for a given cutoff grade. Initially, the mean grade of kriged blocks contained in the output file of routine KRI3D is read into a three-dimensional array in which the cell positions have one to one correspondence with the blocks in the kriged area. The user is asked to specify the cutoff grade which is compared with the mean grade of the kriged blocks to delineate the ore zone from the waste.

4.3.3.1 Alternatives

For the construction of the ore-body in three dimensional space, three approaches were considered. In all of these approaches, points lying on the skin of the ore-body are isolated from the ones within. Figures 4.4 through 4.6 are examples of these approaches. These figures have been used only for explaining the three approaches and they do not represent the actual data used in the program. All the data points in these figures are located on the skin of the ore-body. Only two levels with four blocks on each level are shown here.

One approach is to connect these points using straight lines. The arrowheads in the Figure 4.4 show the sequence in which the points can be joined. This method for joining
Fig 4.4 Scheme for joining data points using straight lines

Fig 4.5 Scheme for joining data points using shapes

Fig 4.6 Scheme for joining data points using cubes
the points can be extended to the other levels and then to other faces until the ore-body is completely defined or drawn.

Another method for connecting these points is shown in Fig. 4.5. The difference here is that a four-sided shape instead of a straight line is used to define the ore-body. The shapes are obtained by joining four adjacent points (e.g. 1,1 - 1,2 - 2,2 - 2,1) as shown in Figure 4.5.

The third approach for constructing the ore-body is to draw a cube; each cube is the same size as a kriged block. This cube is then defined as a cell whose origin is located at the centroid of the cube. By placing a cell at every point which lies within the ore-body (Refer to Fig. 4.6), the ore-body can be modelled as a group of 1 elementary cubes. If the display symbology of the cubes is varied according to the high or low values of the associated grades, the viewer can visualize the grade distribution within the ore-body.

---

1 Display symbology refers to the attributes of the element such as its line-style (solid, dotted or dashed), line-weight (i.e. the thickness of the line), and color. For a detailed information on the above, the reader is referred to pg. 6-31 of the Intergraph Graphic Design System (IGDS) Software Interface manual.
The first approach takes the least amount of computer time as it involves drawing only straight lines and storing the fewest number of data points in the graphic design file as compared to other approaches. However, the drawback with this method is that Intergraph system does not recognize lines for the removal of hidden lines. This deficiency becomes a major handicap as the picture complexity increases. The viewer is likely to lose his mental image of "what is behind what" and also his understanding of what he intends to construct. As a result, the straight line or STICK diagram of the ore-body may defeat the objective of the interactive stope design system which relies on the designer's ability to visualize the shape of the ore-body.

In the second approach, hidden line removal is possible and this method needs less memory and computing time than the 'cube' method. However, the algorithm to generate the model with this method becomes complicated.

The method using cubes provides the maximum amount of information to the user. It is also the most accurate of all as it includes a range of influence for each data point. However, the ore-body obtained by using this algorithm requires more core memory than the other methods because the design file has to store information about the x, y and z co-ordinates of all the corners of the cubes. The cube itself is drawn using six shapes corresponding to the six faces of a cube. Each shape needs 15 data points to
be stored, which implies that for each cube 90 data points are required to be stored). Hidden line removal is possible, although the process will be slower than the method using shapes since the computer has to resolve a larger amount of data.

Three-dimensional models using 'Shapes' as well as 'Cubes' were developed. Isometric views of these models are shown in Figures 4.7 and 4.8. A comparison of these figures suggests that the model using cubes is easier for the user to interpret as well as being more accurate. The logic used to generate the 'cube' model is also simpler than the 'shape' model. In view of these advantages, it was decided to use the 'cube approach' for generating the model. This method is discussed below.

4.3.3.2 Description of 3-D software

A block diagram of the THREE_D program is shown in Fig. 4.9. The three-dimensional model of the ore-body is developed in two stages. In the first stage, the user is asked to supply the ore-body network parameters. Using this information, routine DRAW_CELL draws a cube from shapes. At this stage, it is the responsibility of the user to attach a cell-library to the design file and declare this

2 Refer to pg. 2-10 of IGDS software operating manual.
Fig. 4.7 ISO View of the ore-body using 'Shape' approach (after hidden line removal)
Fig. 4.8  ISO View of the ore-body using 'cube' approach (after hidden line removal)
Fig 4.9 Block diagram of constituent parts of THREE_D

* These routines are available in the Customer Support Library of the Intergraph system
cube as a cell. In the second stage, a call to the subroutine MAIN_PROGRAM is made and a 3-D model is generated. All the subroutines involved in the generation of the model are discussed below.

4.3.3.3 Subroutine DRAW CELL

Since a cube is not available as an element for manipulation in the menu of the Intergraph graphics workstation, it is created from shapes. After the block size is specified, the subroutine DRAW_CELL makes six successive calls to the subroutine SHDFPI in order to construct the cube in three dimensions. The software provides the option to create a new graphics design file during the execution by calling a subroutine NEW_FILE. The design file must be opened before elements can be placed in it. A call to subroutine INDFPI initializes or opens the design file and a call to subroutine DEDFPI closes the file.

4.3.3.4 Subroutine MAIN_PROGRAM

This subroutine is the heart of the 3-D software. Before making a call to this subroutine, the user must

3 Refer to pg 4-17 of IGDS software operating manual.
attach a cell library, which contains the cube defined as a normal and also as a point cell, to the design file. The output from KRI3D is read at this stage and the design file is opened. The user is then prompted with a menu that has following options:

Option one, in the menu allows the user to draw the entire ore body on one level. This is accomplished by placing cubes at all points within the data file which contain grade values within the cutoff grades specified by the user.

Option two generates only the skin of the ore-body. A call to subroutine DRAW_SKIN accomplishes this task.

Option three permits the user to draw zones of richer grade inside the ore body using different symbology for the graphic elements. Subroutine CHANGE SYMBOLOGY provides the user with the flexibility to change the display symbology of the graphic element to be placed in the design file.

Option four provides the exit from the program.

A IGDS design file has 63 levels. The user can store parts of his drawings at different levels and work on them separately. A normal cell can be placed only on the level at which it is created. A point cell unlike normal cell can be placed at any level. Ref. pg.4-18 of IGDS oper. manual.
To execute these options the subroutine MAIN_PROGRAM makes calls to several other subroutines which are discussed later in this chapter.

4.3.3.5 Subroutine INDFPI

This subroutine is a part of the Customer Support Library (CSL) available on the Intergraph system which contains a group of routines that can be accessed by the Intergraph Graphics Design System (IGDS) Interface programs. INDFPI initializes or opens a design file that will be accessed by the design file processor interface (DFPI). This call must be successfully completed before any other DFPI call for writing to the design file is issued. The steps that must be followed before calling this subroutine, are listed below:

1) The user must supply the name of the design file which is to be opened. The specification of the file name is read in two variables INBUF and ICHA. INBUF is an INTEGER*2 word array in which the ASCII file specification is stored and ICHA (again an INTEGER*2 word) contains the length in bytes of the file specification block.

2) Since a call to subroutine INDFPI requires file specifications in LKTRAN (LKTRAN is a disk I/O utility provided to minimize the effort required in coding
disk I/O operations), the necessary LKTRAN file specifications is generated from ASCII file specification by making a call to the conversion routine LKTCSI. LKTCSI accepts INBUF and ICHA as its first and third arguments and returns the LKTRAN file specifications as its second argument. The fourth argument of this subroutine is a code which can either be 0 (for outfile) or 1 (for infile). The fifth argument (IRC) contains the completion code for the execution status of this subroutine on return from LKTCSI. A successful completion of a call to this subroutine is indicated by assigning the value 0 to this variable. Any other value of IRC indicates an error condition and the program terminates giving appropriate messages to the user.

3) INDFPI and all other DFPI routines require a name to identify the region attached to each DFPI task. Any six character, RAD-50 name may be used for this purpose. The VAX system does not require that a region name is attached to its DFPI tasks, but this feature has been kept in the DFPI calls in this program for compatibility with

5 RAD-50 is a special character data representation in which up to three characters can be packed in to 16 bits. For more information, the reader is referred to pp 1-107 of IGDS software interface supplementary manual.
the PDP-11 system. The region name is read in ASCII using the two variables ICHA and INBUF. This name is converted to a RAD-50 name using the ASCII to RAD-50 conversion routine 'ASC2RD'.

Once this information is generated, a call to INDFPPI can be made. Subroutine INDFPPI has eight arguments. These are discussed below:

Arg. 1 (REGNSO):

This is an INTEGER*4 variable which contains the region name (in 6 RAD50 characters) generated by subroutine ASC2RD. This feature was retained only for compatibility with the PDP-11 system.

Arg. 2 (IFILE):

IFILE is an INTEGER*2 word array which contains the design file specification in LKTRAN (LKTRAN is a disk I/O utility provided to minimize the computer effort required in coding disk I/O operations). A call to subroutine LKTCSI generates the LKTRAN file specification from the ASCII file specification.

Arg. 3:

This argument can either be 0, 1 or 2. A value of 0 implies that only the design file is to be opened; a value of 2 means only the cell library will be opened; and a value of 1 will open both the design file and the cell library.
Arg. 4:

When a cell library is to be opened, this argument contains the LKTRAN file specification of the cell library name, otherwise the argument must be 0.

Arg. 5:

This argument is used to select the input format for the interface subroutines. For the VAX system, if the input is to be converted to the IGDS format, this argument would be 1. Since the format of the VAX fortran in UORs is the same as the IGDS format, no conversion is required. A value of 0 has been used for this argument, which implies no change in the format.

Arg. 6:

The programmer has the option to work with either a circular or a non-circular region by specifying this argument as 1 or 0 (1 for circular region). However, this specification is significant only in a PDP-11 system. A circular region is strongly recommended. A non-circular region should only be used when the programmer is more concerned about conserving the storage space rather than conserving time. In this program, a value of 1 is used for this argument.

UORs refer to the units in which the distance is measured in the design file. For more information on UORs, refer to pg. 2-14 of IGDS software interface sup. manual.
Arg. 7 (IRC) :

This argument contains the completion code; Any value other than 0 implies an error condition (An explanation of the error codes can be found in Sec. 6.1 of IGDS software interface manual).

Arg. 8 :

This argument sets up the terminal ID. Normally, the value 'EX' is used for this software. However, when the user commands are also being used simultaneously with DFPI, the actual terminal number must be used for this argument.

4.3.3.6 **Subroutine CLDFPI**

This subroutine places a cell in the design file which contains an angle and a scale factor. The cell must be defined in the cell library before a cell can be placed in a file. The cell library, which must be specified during initialization, contains the following ten arguments.

Arg. 1 (LVL):

This argument defines the level at which the point cell is to be placed. For a normal cell this value is meaningless because the cell is placed on the level specified in the cell library irrespective of the value of this argument.
Arg. 2 ( IGG ) :

This argument is a two word array ( INTEGER*2 ). The 7
first element specifies the graphics group code, and the
second element of this array assigns a graphic group 7
number to the graphic group.

Arg. 3 ( ROT ) :

This argument represents the angle through which the
cell will be rotated from its original position ( i.e. the
position at which it is specified in the cell library )
before being placed in the design file. This angle is
measured in degrees as a double precision floating point
number.

Arg. 4 ( SCALE ) :

This argument contains the x, y and z scale factors as
double precision floating point numbers. A value of zero in
the argument will cause default value of 1.0 to be sent to
DFPI.

Arg. 5 ( CLNAME ) :

This argument is a six byte array that contains the
ASCII cell name.

Arg. 6 ( IORGS ) :

This argument defines the coordinate of the cell's
origin. The array contains x, y and z coordinates in UORS.

Elements belonging to the same graphic group can be
manipulated together. Ref. pg. 6-41 of IGDS interface man.
Arg. 7 (IVIEW):

This argument is a two word array in which the first element is the view code and the second element defines the view number in which the cell is to be placed.

Arg. 8 (IRC):

This argument is the return code which is used in error diagnostics.

Arg. 9 (IATLK):

This argument is a variable length array which contains the attribute linkage (see section 6.2 of IGDS Software Interface manual for an explanation). The first two words must be set to 0 if there is no attribute linkage.

Arg. 10 (ISPEC):

This argument defines the symbology of each cell that is placed in the design file. It is meaningful for a point cell only. For normal cells the symbology will be as specified in the attached cell library. This argument is discussed at length under the heading CHANGE SYMBOLOGY.

4.3.3.7 Subroutine DRAW-SKIN

This subroutine generates the skin of the ore-body by constructing one face at a time. As discussed earlier, the data from KRI3D is read into a three dimensional array. The
data file may be visualized as a rectangular block, as shown in Figure 4.10. The position of each data point in the data file corresponds to the position of the centroid of the blocks in the kriged area. The data points contain the estimated grades of kriged blocks. The scan begins at the top left corner of a particular face (say A in Fig. 4.10) in the data file and continues backward (i.e. for \( i=1,I \)) at the same level (i.e. at the same \( k \)) until a block with an estimated grade equal to or greater than the cut-off grade is located. The element\((j,k)\) in a 2-dimensional array \( \text{ITEST} \) of size \((J,K)\) is initialized to 1. The value of \( i \) is stored in an element of another 2-dimensional array \( \text{IK} \) at the same position (i.e. \( j,k \)). Thus, the three coordinates \( i,j \) and \( k \) are saved for that block. If the scan is unsuccessful in locating any element with a grade greater than the cutoff grade on that level, the corresponding elements in the two 2-dimensional arrays are initialized to zero. The search is continued for all levels (i.e. for \( k=1,K \)) in that column and then similarly for other columns (i.e. for \( j=1,J \)). Scan is terminated only when elements of all rows and columns for that particular face have been scanned. This method of scanning fills the array \( \text{ITEST} \) with 0's and 1's to represent the shape of the corresponding wall of the ore-body. Since \( \text{IGDS} \) graphic routines accept coordinates only in UORS, 'x','y' and 'z'
Fig 4.10 Conceptual data file for constructing 3-D model
coordinates of the blocks are multiplied by the working unit of the seed file to get the desired conversion. A call to subroutine CLDFPI is then made to place cubes successively at each point containing a '1', beginning from the top left corner of the face. Once the face 'A' is drawn, the remaining five faces are obtained using the same algorithm.

Once the skin of the ore-body has been obtained, the user has the option to rotate the ore-body model in three dimensional space and to invoke the hidden line removal routines to get a better perspective.

4.3.3.8 Subroutine SHDFPI

This is a standard DFPI interface subroutine. It allows the user to place shapes (a shape is a multisided polygon) in the graphic design file. This subroutine needs seven values to be transferred. Each of them is discussed below.

Arg 1 (IGG):

This argument is a two word array (INTEGER*2). The first element of this array determines whether or not a graphic group is to be started (or cancelled). The second element assigns a number to the graphic group if a group has been started. Both of these elements have been

8 Refer to pg. 2-22 of IGDS software interface sup. manual
initialized to zero since the graphic group need not be initialized for this program.

Arg. 2 (LVL):

This argument determines the level at which the element (in this case, a cube) will be placed in the design file. This data is supplied interactively within the program.

Arg. 3 (ISPEC):

This argument is a two word array that defines the symbology of the element to be placed in the design file.

Arg. 4 (ISHAPE):

This argument is a variable length array in INTEGER*4 Fortran or IGDS format that contains the x, y and z coordinates in UORs which define the shape. The dimension of Arg. 4 should be equal to three times the value of Arg. 5 (i.e. three times the number of coordinate sets needed for the shape) for a three-dimensional file. For example, this dimension is 15 for a four-sided shape which needs 5 sets of coordinate points to be completely defined.

Arg. 5 (NSHAPE):

This INTEGER*2 variable defines the number of coordinate sets in the array represented by Arg. 4 of this subroutine. For a four-sided shape this value is 5.

Arg. 6 (IRC):

This argument contains the return code for completion.
If a call to the subroutine SHDFPI was successfully completed this variable should have a value equal to zero. The value of IRC is checked for errors after a call to SHDFPI has been made. Different values of IRC indicate different kinds of error.

Arg. 7 ( IATLK ):

IATLK is a variable length array that is available to the user for describing the attribute linkage. It has been set equal to zero in this program since there is no special attribute linkage needed for our purpose. For further information on this argument, refer to sub-section 6.2 of software interface IGDS manual.

4.3.3.9 Subroutine DEDFPI

A IGDS file can only be accessed for data transfer by first opening it through a call to subroutine INDFPI, (see section 4.3.3.5). It is imperative that the file should be detached from DFPI when the task of reading or writing in the design is finished. This is accomplished by a call to the subroutine DEDFPI. For normal termination, the argument of subroutine DEDFPI should be one.
4.3.3.10 Subroutine CHANGE SYMBOLOGY

When the user opts for number three in the menu displayed by the MAIN_PROGRAM, he is asked if he wants a change from the existing symbology for the shapes. To get a better appreciation of the richer zones inside the orebody, a different symbology can be attached. On an affirmative reply, the program makes a call to this subroutine. Variable ISPEC is passed on as its argument. ISPEC is a five word array of INTEGER*2 elements. The elements of this array determine the attributes of the element that are placed in the design file. A menu is displayed to the user which gives the user the option to change class number (0-5), status (0 and 1), style (0-4), line weight (0-31), and color code (0-255) of the elements (in this case cubes) that are placed in the design file. For an explanation of the codes for the attributes of an element refer to page 6-31 of the IGDS software interface manual. The fifth element in this array, which determines the color code, is only meaningful on color terminals. The value of elements of ISPEC is changed as desired by the user and then passed to the subroutine CLDFPI as its 8th argument.
4.3.3.11 Subroutine NEW FILE

The program provides the option of drawing the ore-body in an existing file or to a new file. The user can create a new file by making a call to subroutine NEW_FILE. This procedure creates a contiguous design file and copies the contents of the seed file 'SEEDZ.DGN' to the design file. Before making a call to this subroutine, the user must be sure that the three-dimensional seed file is available on the user's disk. When a call to this subroutine is made, the user is prompted for the file name and the number of blocks to be allocated to the file. A 500 block file should provide the user ample space to construct the ore-body and carry out interactive stope design. If the size of the user's design file is inadequate the INCREASE command available on the VAX system may be invoked to increase the size of an existing file. The actual creation of the design file is done by a call to FILALO, which is one of the I/O utilities routines available on the system. The arguments of this subroutine are discussed below.

Arg. 1 (ILUN):

This variable defines the logical unit on which the new file will be created. Units 5 and 6 should not be used because they define the terminal.
Arg. 2 ( IBUFF ) :

Variable IBUFF is a one byte array which contains the ASCII file specification.

Arg. 3 ( KOUNT ) :

It is an INTEGER*2 word that contains the number of characters in IBUFF.

Arg. 4 ( IBLOCKS ) :

This variable is an INTEGER*2 word which specifies the number of blocks to be allocated to the design file. The user has to supply this data interactively.

Arg. 5 ( IRC ) :

This variable contains the code for completion of the call to FILALO.

Once the call to FILALO is successfully, a design file is created in the user's directory. The file is then opened and the contents of the seed file, SEEDZ.DGN, are read, one sector at a time, and written to the newly created file.

The program also makes calls to subroutines ERROR and ERROR9 after calls to the DFPI routines. These subroutines are system utilities to assist the user in properly terminating the program if an error is encountered in a DFPI call. Appropriate messages to the user help him/her diagnose the error and take corrective action.

After a successful execution of the DFPI program, the
user can switch to the graphic mode of the system and open the design file. The ore-body can be then inspected from any angle by invoking commands for three-dimensional rotation. The hidden line removals with or without shading can also be performed if they can help the engineer carry out the interactive stope design.

4.4 Model Verification

The shape of the ore-body generated from the data used in the program was complex and consequently made the actual verification of the model difficult. Since, the shape of the ore body plays a crucial role in mine design, a step to verify the correctness of the model was considered necessary.

The research approach was to apply the software to simpler bodies where the shapes were easy to visualize and compare to the THREE D output. It was assumed that, if the program worked for simpler models it would be valid for models with complex shapes.

Three 3 by 3 matrices with test data containing only 0's and 1's were created for the purposes of verification. It was assumed that 1's defined the ore-zone and 0's denoted a waste zone. Conceptually, the data file would have appeared as shown in Figures 4.11, 4.12 and 4.13. It
Fig. 4.11 Conceptual data file for Case 1
Fig. 4.12 Conceptual data file for Case 2
Fig. 4.13  Conceptual data file for Case 3
was not difficult to visualize the shapes of the ore-bodies represented by these data files. The actual data files for the three cases are shown in Figs. 4.14, 4.15 and 4.16.

The three dimensional model was developed by the program for these three cases and that the computer outputs were in perfect agreement with the expected shapes of these models. The outputs from the computer for these cases are shown in Figure 4.17, 4.18, 4.19. Based on these results, the model was assumed to be valid for larger data files representing complex shapes.
<table>
<thead>
<tr>
<th>Type</th>
<th>Case 1: Dat</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2.0000</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2.0000</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2.0000</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2.0000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2.0000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2.0000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2.0000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3.0000</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3.0000</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3.0000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3.0000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3.0000</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3.0000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3.0000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3.0000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3.0000</td>
</tr>
</tbody>
</table>

Fig. 4.14 Actual data file for Case 1
<table>
<thead>
<tr>
<th>Type</th>
<th>Case 2. DAT</th>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4.15  Actual data file for Case 2
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Fig 4.16  Actual data file for Case 3
Fig 4.17   THREED output for case 1
Fig 4.18  THREED output for case 2
Fig 4.19. THREED output for case 3
5.1 View Setup

Each screen of Intergraph graphic workstation can be split in four windows. Each window can hold a different view as indicated in Figure 5.1. To produce an orthographic or isometric view the user keys the view name using the following format:

VI = (TOP, FRONT, RIGHT, LEFT, BACK, BOTTOM or ISO)

The user then selects the intended view by moving the cursor on the screen to the desired view and pressing the data button available on the hand-cursor. A view set-up similar to the one shown in Figure 5.2 was used in the design file opened by this software.

When the user opens the graphic design file the four views (top, front, right and isometric) of the ore body appear on the left screen of the Intergraph system and the right screen contains an enlarged isometric view. These displays are shown in Figures 5.3 and 5.4. When the user opens a 3-D design file, the window-origins of these views default to a location approximately at the center of the

1 Refer Sec.2 of IGDS software 3-D manual.
Fig 5.1 Twin screen of Intergraph holding eight different views
Fig 5.2 View set-up as used in THREEED
Fig 5.3  3-D model of the ore-body as seen on left screen of the work station
Fig 5.4 Isometric view of ore-body as seen on right screen of the work station
design cube. The global origin (0,0,0) is also set at the center of the design cube provided the seed file in the user's directory is set properly. The coordinate system for this setup is shown in Fig. 5.5. It is important to distinguish between the view axes and the design cube drawing axes (refer Figures 5.5 and 5.5a). The view coordinate system has no real origin. It is used to input directions relative to the screen, where +X axis is horizontal to the right, +Y axis is vertical and up, and +Z axis is directly out from the screen. +X and +Y directions in Figure 5.5 correspond to the east and north directions in the field. This set-up of the views was chosen because only the right screen is controlled by the commands for the three-dimensional rotation. The ability to rotate the pictorial view is absolutely essential for interactive stope design.

The cut-off grade for the ore-body (for gold) was set arbitrarily at 1.41 for this exercise. In an actual mine design this value must be chosen very carefully because it affects the economics of the production system and the profitability of the mine.

5.2 Hidden Line Removal

Hidden line removal is possible on both the screens
Fig 5.5 Drawing coordinate system

Fig 5.5a View coordinate system
and can be applied to any 3-D orthographic, isometric or oblique view. Surface elements are treated as opaque and all elements or parts of elements which should be obscured by a plane are temporarily removed from the graphical display. The output with the hidden line removal can be displayed on the screen or saved on a disk file. The user has the option to display this file on the screen. Fig. 5.6 is a duplication of Fig. 5.4 but with the hidden lines removed.

5.3 Shading

The user has the option to create a shaded view of the rotated surfaces after carrying out the removal of hidden lines. On a black and white work station this option shades surfaces by turning on a random number of pixels within the boundaries of the surface. The number of pixels turned on is determined by the angle which surface normal makes with the line of sight. The light source is assumed to be at the eye. Figures 5.7 to 5.9 contain the shaded views of the ore-body. On a black and white terminal the shading does not seem to help the user comprehend the shape of the ore-body.
Fig. 5.6  Iso view of the ore-body as seen on right screen (after hidden line removal)
Fig. 5.7 Ore-body from side looking top and south (with shading)
Fig. 5.8 Ore-body turned upside down and looking from top (with shading)
Fig. 5.9 Another view of ore-body turned upside down and looking from top (with hidden lines removed)
5.4 Coloring

On a color raster work station, the option to shade with color is available to the user. In this option the visible surface of the drawing are painted with a shade of the color that was selected to draw the original elements. The shading is determined by the angle which the surface normal makes with the line of sight (the light source is assumed to be at the eye, although the user can have different perspectives).

Although a color terminal was not available for this research, the software will not need any modification to exploit the potentials of a color terminal. Provisions have been made in the package to enable the design engineer to change the symbology of the elements including the choice of color.

5.5 Available Options

As has been discussed in chapter 4, there are various options available to the user. Figs. 5.3 and 5.6 correspond to the option no.1 in the menu displayed to the user. In this case, the entire ore-body has been drawn on level 1. Option 2 allows the user to generate only the skin of the ore-body. This information may be helpful in getting a
better appreciation of the distribution of grades inside the ore-body.

Fig. 5.10 corresponds to the option no. 3 of the menu. It is helpful to know where the richer zones lie inside the ore-body and what they look like. This information may be used to select the shape of the stope. Thicker lines in the figure show the zone of richer grades and were obtained by changing the symbology of the shapes. If a colour terminal is available the grades allocation can be emphasized by using different colors.

5.6 Effects of Kriging

Figure 5.11 shows the isometric view of the ore-body constructed with the raw data without applying kriging. A comparison of this model with Figure 5.6 reveals that kriging gives a more regular shape to the ore-body. This results because kriging smoothes or averages the data.
Fig. 5.10  Ore-body with a contour slice of richer grade ( > 3.00% )
Fig. 5.11 Ore-body as seen on the left screen of work station developed without kriging the bore-hole raw data (with hidden lines removed)
CHAPTER : 6

Conclusions and Recommendations

This research demonstrated the feasibility of generating a three-dimensional model of an ore-body directly from the bore-hole data using Intergraph CAD/CAII techniques. Specific conclusions and recommendations are as follows:

1) At the beginning of this study, it was anticipated that the time and effort to generate a three-dimensional model of an ore-body could be greatly reduced by eliminating the digitizing process and interpreting the bore-hole data directly. This result has been confirmed by this research. Three-dimensional representations which took days to construct with the digitizing process are completed in minutes with the 3-D software.

2) This study used two methods to generate the model, the 'cube approach' and the 'shape approach'. The 'cube approach' was preferred because of its greater accuracy and simplicity. Reserve estimation is also simpler using this approach. However, the 'shape approach' required less memory than the 'cube approach' and may be considered under circumstances where the lack of memory space prohibits the use of the 'cube' method.
3) One disadvantage of the 3-D software is that it executes the model building procedure in two stages. Between the stages the user is required to create and attach a cell library to the design file. The cube must also be declared as a cell before executing the second stage. Further studies could explore the possibility of modifying the software so that the task of initializing and attaching the cell library to the design file is performed without intervention by the user.

4) The Intergraph software cannot dynamically rotate the object in three-dimensional space while simultaneously removing hidden lines. This deficiency may prove to be a hindrance to interactive stope design. The problem may be circumvented by storing the outputs from the hidden line removal routine for various positions of the ore-body. The user could refer to the outputs, as and when required, while carrying out interactive stope design.

5) In its present configuration the 3-D software does not permit the program to execute while simultaneously displaying the model on the screen. Visual display of the model is possible only after the entire program has been executed. It should be possible to develop User's commands to execute the program and display the model on the Intergraph work station screen simultaneously. This capability would reduce the number of steps required to
create the design because the need to open the design file after the execution of the software is eliminated.

6) Although the software developed in this study is system dependent, it should be possible to apply the logic of the algorithm to other computer systems by modifying the system dependent commands.

7) The greatest benefit from applying a computer to mining engineering is likely to be realized if the computer applications are totally integrated into the routine engineering design process. This package, in its current state, does not provide such a tool but is a step in that direction. The full potential of this package can only be realized when routines for reserve estimation and scheduling are developed and integrated with the existing software. A routine for reserve estimation will enable the engineer to compute reserves for any mining volume such as block, stope or pillar. A scheduling routine will help management minimize the intervals of non productive time between unit operations that must be done in a sequential pattern (e.g. drilling, charging, blasting, loading, hauling, dumping and filling). Capabilities of this package can be further extended by including routines for generation of drilling patterns and blast designs.
BIBLIOGRAPHY


5. Ranson, M.A.; Daniel, P.T.


6. Hartley, D.; Ranson, M.A.


8. Ley, A.P.; Lawrence, E.L.

"Encore - A computer Program For Drill - Shoe Data Management, Coal Ore Body Modelling and Coal Reserve Estimation" Proceedings of the 17th International


10. Arcamone, J.; Dejean, M.; Sayette, C.


12. Chatterjee, P.K.; Jalkanen, G.J.


13. Notley, K.R.; Wilson, E.B.

"Three Dimensional Mine Drawings by Computer Graphics." CIM Bull. (Canada), Vol. 68, No. 754, pp 60 -


16. Cox, D.J.; Johnson, M.T.


"Development of an Interactive Graphics Station For Mining Application."

USER'S MANUAL

This manual has a limited objective of assisting the user to get started with SURFMIN and software for developing three dimensional model of ore-bodies. It should not be construed as a guide to Intergraph graphic workstation and Vax system. It provides information about only those commands which the user is likely to use to run this package. For clarity user's commands have been underlined in capital letters, whereas user's varying inputs have been shown in lower case within quotes. Computer generated responses are shown in capital letters but not underlined.

To begin, press the Return key. The machine will respond: USER'S NAME:
Type: 'chatterjee' (Always press Return after any input)
Computer will respond with
ENTER PASSWORD
Type: 'apcom'
The machine will respond with logon messages.

At this point you should refer to the user's guide for package SURFMIN to run geostastical routines. To run THREED it is necessary to create the input data file by running routines SURVEY, CLAS and KRI3D in that order. (The input data file using raw data file RAWDATA.DAT and executable images of all geostatistical routines have already been
created on your disc). To run THREED

Type: RUN threed

Computer will respond with

YOU SHOULD ENTER DATA USED TO DEVELOP YOUR ORE-BODY NETWORK MODEL. IT SHOULD BE CONSISTENT WITH YOUR INPUT IN ROUTINE KRI3D. ENTER THE FOLLOWING

NUMBER OF BLOCKS IN X,Y AND Z DIRECTIONS

If you are using a data-file other than already available on the disc, respond appropriately to these prompts by the computer. The input shown here pertains only to the data file already available on the disc.

Type: '10 10 5'

Computer will prompt you with

ENTER THE X,Y AND Z DIMENSIONS OF EACH BLOCK

Type: '40 40 24'

Computer will respond

DO YOU WANT TO CREATE THE CELL? ENTER Y/N.

If you have not created a cube of the size equal to that of kriged block you must enter Y. The computer then prompts you with

DO YOU WANT TO OPEN A NEW FILE?

ENTER N FOR NEW AND 0 FOR OLD

If you enter N then computer will prompt you for file name and file size. An example of file name is 'Threed.dgn'. Explicit version number is not necessarily needed to be
given. A file size of 500 blocks should be adequate for developing the 3-D model and subsequent interactive stope design. Computer should respond with the message

VALID DESIGN FILE CREATED

In case of a reply '0' from you, computer will prompt you for the old file name. Reply appropriately.

Computer will then prompt you with

ENTER THE REGION NAME

Type: '3rnam'  (Any 6 character RAD50 name will do)

Computer will now open the new/old graphic design file for writing. A successful completion of this task is indicated by a message

INDFPI COMPLETED followed by 0 (any other value indicates an error condition.)

The program then draws a cube using 6 shapes. It terminates giving you the following message

GO TO THE NEXT STAGE, RUN THE PROGRAM AGAIN.

You should now go to the utilities and create a 3-D cell file (recommended size -75 blocks). To get into the graphics mode

Type: '@login'

Computer will respond with GRAPHICS along with other messages. Enter the name of the design file that you created while running THREED and hit the Return key.

Once you have opened the design file, you should attach the
newly created cell library to this file.

Type : NC='name of your cell library'/3D

( e.g. THREED.CEL/3D )

To place the cell in the cell library, put a fence around the cube which you drew by running THREED. Place the fence in the ISO view of the work station set-up. Select the 'define cell origin' command from the menu. The computer will prompt you with

IDENTIFY ORIGIN

Type : xy= 20,20,12 ( these coordinates are valid only when the block size is 40,40,24. Appropriate value should be input to get the origin at the centroid of the cube if the block size is different. )

Type : CC='name of the cell' ( not more than 6 characters )

The above command will create a normal cell for you. If you need to change the symbology of the elements to be placed in the design file, you must create a point cell also. To accomplish this, repeat the above procedure and

Type : CC='name of the cell, description, P

( description field is optional )

You must now run THREE D again. When the computer asks for your option for creating the cell you must enter 'N'.

Computer will ask

ENTER FILE NAME

Answer appropriately. Computer will ask for the names of the cell-library and the cell. Reply appropriately to these questions.

Computer will respond

ENTER THE NUMBER OF VARIABLES IN DATA FILE

The maximum number of variables that can be handled by geostatistical routine and 3-D software is 8. The existing data file has 3.

Type :'3'

The machine will respond

ENTER THE VARIABLE # FOR WHICH YOU WANT THE 3-D MODEL

Type :'1' ( 1 corresponds to Gold in the existing data file )

Computer will now read the data file and write it on terminal. It will then prompt you with

ENTER CUTOFF

Type :'1.40' ( this value has been chosen arbitrarily here. However, in actual mine design it depends on the economics of the that mine and the profitabilty criterion. )

You will then be prompted with the following menu for your reply
YOU HAVE FOLLOWING OPTIONS:
1. ENTIRE ORE-BODY ON ONE LEVEL
2. DRAW ONLY THE SKIN OF ORE-BODY
3. DRAW A CONTOUR SLICE OF RICHER GRADE
4. EXIT THE PROGRAM

ENTER THE NUMBER OF YOUR OPTION:

After entering the number of your option you will be prompted with

ENTER LEVEL NUMBER

Enter a number between 1-63 of your choice. It is at this level the figure corresponding to that option will be drawn. If you choose option 3, computer will prompt you with

ENTER THE CUTOFF FOR THE RICHER GRADE

Enter an appropriate value. Computer will ask DO YOU WANT TO CHANGE DISPLAY SYMBOLOGY?

ENTER Y/N

Upon an affirmative reply, a menu will be displayed to you for your choice for the changes in display symbology. For a detailed information on these changes please consult pg. 6-31 of IGDS software interface manual.

After executing an option you will be returned to the menu. Exit from the program is provided through option 4.

To get into the graphic mode

Type: '@LOGIN'
Once you have opened the design file you can go to any level to inspect the ore-body. To do this
Type: 'lv= (the desired number)'
If you find that the figure is not fully contained in a view, execute the FIT VIEW command available on the system. You may have to put data point twice in the desired view.

To invoke hidden line removal routine
Type: 'uc=pro_dd_hlin:hline.ucm'
or
Type: 'at=hline'
If you are working with 3-D menu, it can be invoked also by hidden line command available on the menu. You will then be displayed a tutorial from which you can carry out removal of hidden lines with or without shading in a desired view by just putting a data point at the desired location.

To rotate the ore-body in three dimensional space choose the DYNAMIC VOLUME command. The volume is defined by inputting two data points which define the diagonal of the rectangular volume (Note: Three dimensional dynamic rotation can only be performed on right screen). The right screen should update as the terminal data set is downloaded. Now
Type: 'am=cbmenu,cb' (this will attatch the three dimensional cursor button menu)
For dynamic rotation to occur, position the visual cursor in the desired view on the right screen and press cursor button number 4. The speed of the rotation is determined by the distance of the cursor away from the center of the view. The direction of the rotation is determined by the location of the cursor in the view. The volume will rotate in the direction defined by an imaginary line from the center of the screen to the cursor.

Note: cursor button number 6 is ZOOM-IN

cursor button number 7 is PAN

cursor button number 8 is ZOOM-OUT

To get out of the graphics mode press control and Z simultaneously. To log out

Type: 'lo'
T CUBE1.FOR
PROGRAM THREED

THREED IS AN INTERACTIVE PROGRAM THAT WILL CONSTRUCT
A THREE DIMENSIONAL MODEL OF THE ORE-BODY FROM THE
DATA GENERATED EARLIER BY THE ROUTINE KRI3D. IT
PROMPTS THE USER FOR THE ORE-BODY NETWORK PARAMETERS.
USER'S INPUT FOR THESE PARAMATERS IN THIS PROGRAM
SHOULD BE CONSISTENT WITH HIS INPUTS IN ROUTINES
'CLAS' AND 'KRI3D'. ON REQUEST FROM THE USER, THIS
ROUTINE CREATES A NEW 3D DESIGN FILE OR OPENS AN OLD
DESIGN FILE IN WHICH THE MODEL IS DEVELOPED. THIS
ROUTINE ALSO PROVIDES THE OPTION TO CHANGE THE SYMBO-
LOGY OF THE ELEMENTS TO BE PLACED IN THE DESIGN FILE.
THE SOFTWARE IS COMPATIBLE WITH INTERGRAPH APPLICATION
SOFTWARE.

PARAMETERS

X(I,J,K) KRIGED GRADE OF INDIVIDUAL SUPER
BLOCKS.
NX,NY,NZ NUMBER OF S. BLOCKS DEFINING THE
=NXYZ(3) THREE SIDES OF THE NETWORK.
IDBX,IDBY, IDBZ=IDB(3) DIMENSION OF S. BLOCKS ALONG THREE
AXES.

SUBROUTINES

MAIN_PROGRAM DISPLAYS THE MENU TO THE USER,
OPENS THE NEW/Old GRAPHIC DESIGN FILE AND CALLS OTHER
SUBROUTINES TO DO DESIRED FUNCTIONS.

DIMENSION NXYZ(3),IDB(3),XYZ(3)
COMMON X(20,20,20),IK(20,20)
* ,ITEST(20,20),CUTOFF,IDBX,IDBY,IDBZ,LARGEST

PRINT*, 'PLEASE ENTER DATA USED TO DEVELOP YOUR ORE-BODY'
PRINT*, 'NETWORK MODEL. IT SHOULD BE CONSISTENT WITH'
PRINT*, 'YOUR INPUT IN ROUTINE KRI3D. ENTER FOLLOWINGS'
PRINT*, '
PRINT*, 'NUMBER OF BLOCKS IN X,Y, AND Z DIRECTIONS'
PRINT*, '
READ*, (NXYZ(I),I=1,3)
PRINT 200,(NXYZ(I),I=1,3)
200 FORMAT(' ',NX = ',I5,' NY = ',I5,' NZ = ',I5)
PRINT*, ''
PRINT*, 'ENTER THE X, Y AND Z DIMENSIONS OF EACH BLOCK'
PRINT*, ''
READ*, (IDB(I), I=1,3)
PRINT 205, (IDB(I), I=1,3)
205 FORMAT(' ',', IDBX = ', 110, ' IDBY = ', 110, ' IDBZ = ', 110)
NX = NXYZ(1)
NY = NXYZ(2)
NZ = NXYZ(3)
IDBX = IDB(1)
IDBY = IDB(2)
IDBZ = IDB(3)
TYPE*, ' DO YOU WANT TO CREATE THE CELL. ENTER Y/N.'
READ(S, 206) ANS
206 FORMAT(A1)
IF(ANS.EQ.'N') GO TO 207
CALL DRAW_CELL(IDBX, IDBY, IDBZ)
STOP ' GO TO THE NEXT STAGE 'RUN THE PROGRAM AGAIN'
207 LARGEST=NXYZ(1)
IF(LARGEST.LT.NXYZ(2)) LARGEST=NXYZ(2)
IF(LARGEST.LT.NXYZ(3)) LARGEST=NXYZ(3)
CALL MAIN_PROGRAM(NX, NY, NZ)
STOP ' NORMAL HALT'
END
SUBROUTINE MAIN_PROGRAM(LAST1, LAST2, LAST3)

C
C PARAMETERS NV
C INBUF(15) NO. OF VARIABLES IN THE DATA FILE
C AN INTEGER*2 VARIABLE IN WHICH IS
C STORED THE ASCII FILE SPECIFICATION
C IFILE(7) AN INTEGER*2 WORD ARRAY DEFINING THE
C DESIGN FILE WHERE THE GRAPHIC ELEMENTS
C ARE TO BE PLACED,
C IRC AN INTEGER*2 VARIABLE IN WHICH A RET-
C URN CODE IS STORED,
C ICHA AN INTEGER*2 WORD WHICH CONTAINS THE
C LENGTH IN BYTES OF THE FILE SPECIFI-
C CATION BLOCK,
C IEX AN INTEGER*2 VARIABLE WHICH DEFINES
C THE TERMINAL IDENTIFICATION,
C REGN50 AN INTEGER*4 VARIABLE WHICH CONTAINS
C THE REGION NAME (6-RAD50 CHARACTERS),
C (NOTE: USED FOR COMPATIBILITY WITH
C FDP)
C ISPEC(5) AN INTEGER*2 WORD ARRAY DEFINES DISP-
C PLAY SYMBOLOGY,
C CUTOFF CUTOFF GRADE SUPPLIED BY THE USER
C AGAINST WHICH THE GRADE OF DIFFER-
C ENT BLOCKS WOULD BE COMPARED TO
C CONSTRUCT THE 3-D ORE-BODY MODEL
C
C SUBROUTINES INDFPI THIS IS A SYSTEM'S DFPI INT-
B-3

INTERFACE ROUTINE FOR INITIALIZING OR OPENING A GRAPHIC DESIGN FILE TO BE ACCESSED BY DFPI.

LKTCSI

THIS IS A DISK I/O UTILITY ROUTINE AVAILABLE IN CUSTOMER SUPPORT LIBRARY OF INTERGRAPH SOFTWARE INTERFACE. IT PROVIDES A MEANS OF GENERATING THE FILE SPECIFICATION BLOCK REQUIRED BY LKTRAN.

ASC2RD

CONVERTS A STRING OF ASCII CHARACTERS INTO AN EQUIVALENT STRING OF RAD50 CHARACTERS.

ERROR

THIS ROUTINE TESTS DFPI ERROR RETURN CODES.

DRAW_SKIN

DRAWS WALLS OF THE ORE-BODY INDIVIDUALLY OR ENTIRE ORE-BODY DEPENDING ON USER'S REQUEST.

CHANGE_SYMBOLOGY

ALLOWS THE USER TO CHANGE THE SYMBOLOGY OF THE ELEMENTS PLACED IN THE DESIGN FILE.

NEW_FILE

ALLOWS THE USER TO CREATE NEW DESIGN FILE OF DESIRED SIZE.

DIMENSION V(8)
COMMON X(20,20,20),IK(20,20)
*,TEST(20,20),CUTOFF,IBX,IBY,IBZ,LARGEST
IMPLICIT INTEGER*2(A-Z)
BYTE CLNAME(6)
INTEGER*2 IGG(2), ISPEC(5),IATLK(4),LVL
INTEGER*4 IORGS(3),REGN50
REAL*8 ROT,SCALE(3)
INTEGER*2 IFILE(7),INBUF(15),ICELL(7),IView(2)
DATA IGG/2*O/,ISPEC/S*O/,IEX/'EX'/,IVIEW/O,1/
DATA ROT/O.O/,IORGS/3*O/IATLK/4*O/
DATA SCALE/3*1.0/
OPEN(UNIT=4,FILE='OUTKRI',STATUS='OLD')
PRINT*, ENTER THE NUMBER OF VARIABLES IN DATA FILE
READ(5,**) NV
PRINT*, ENTER THE VARIABLE # FOR WHICH YOU WANT THE 3-D MODEL, IT SHOULD BE CONSISTENT WITH THE ORDER IN WHICH IT APPEARS IN THE DATA FILE
READ(5,**) NO
DO 5 I=1,LAST1*LAST2*LAST3
READ(4,1220)M1,M2,M3,(V(L5),L5=1,NV)
5 CONTINUE
WRITE(6,**)M1,M2,M3,V1
1220 FORMAT(315,3F10.4)
X(M2,M1,M3)=V(NO)
WRITE(6,22)((X(I,J,K),J=1,LAST2),K=1,LAST3)
22 FORMAT(10(F5.2))
CONTINUE
WRITE(6,10)
10 FORMAT(' ENTER FILE NAME,DEV-UIC-NAME-EXT-VER')
READ(5,15) ICHA,INBUF
15 FORMAT(Q,15A2)
CALL LKTC51(INBUF,IFILE,ICHA,0,IRC)
IF (IRC.NE.0) STOP '7-WORD LKTRAN CONV ERROR'
TYPE *, ' ENTER CELL LIBRARY NAME'
READ(5,15) ICHA,INBUF
CALL LKTC51(INBUF,ICELL,ICHA,0,IRC)
IF (IRC.NE.0) STOP '7-WORD LKTRAN CONV ERROR'
WRITE(6,20)
20 FORMAT(' ENTER REGION NAME,6-CHARACTERS')
READ(5,15) ICHA,INBUF
CALL ASC2RD(INBUF,REGN50,ICHA,IRC)
IF (IRC.NE.0) STOP 'ASC2RD CONVERSION ERROR'
CALL INDFFP(REGN50,IFILE,1,ICELL,0,1,IRC,IEX)
WRITE(6,72)IRC
72 FORMAT(' INDFFP COMPLETED',I6)
IF (IRC.NE.0) CALL ERROR
TYPE *, ' ENTER CELL NAME,6 CHARACTERS'
READ(5,30) ICHA,CLNAME
30 FORMAT(Q,6A1)
1200 DO 52 II1=1,5
52 CONTINUE
WRITE(6,801)
801 FORMAT(AI)
TYPE *, ' YOU HAVE FOLLOWING OPTIONS'
TYPE *, ' --------------------------,
TYPE *, /' ENTER THE NO. OF YOUR OPTION:
READ(5,13) A
13 FORMAT(A1)
IF(A.EQ.'1' .OR.A.EQ. '2' .OR.A.EQ. '3' .OR.A.EQ. '4') GO TO 700
WRITE(6,801)
TYPE *, ' PLEASE ENTER A NO. FROM THE MENU'
GO TO 1200
700 IF(A.EQ. '1') IREP=1
IF(A.EQ. '2') IREP=2
IF(A.EQ. '3') IREP=3
IF(A.EQ. '4') IREP=4
IF(I'REP.EQ.4) GO TO 1309
TYPE *, ' ENTER THE LEVEL NO.'
READ(5,*) LVL
IF(I'REP.EQ.1) GO TO 1201
IF(IREP.EQ.2) GO TO 1203

C THE USER HAS OPTED FOR OPTION NO. 3. PROMPT HIM FOR CUTOFF
C GRADE FOR RICHER ZONE.
C
TYPE *, ' ENTER THE CUTOFF FOR RICHER ZONE'
READ(5,*) CUTOFF
TYPE*, ' DO YOU WANT TO CHANGE DISPLAY SYMBOLOGY?'
1651 TYPE*, ' ENTER Y/N'
3 FORMAT(A1)
READ(5,3) ANS
IF(ANS.EQ.‘Y’,OR.ANS.EQ.‘N’) GO TO 1650
GO TO 1651
1650 IF(ANS.EQ.‘N’) GO TO 1202

C PROVIDE OPTIONS FOR CHANGING THE DISPLAY SYMBOLOGY
C
CALL CHANGE_SYMBOLISM(ISPEC)
GO TO 1202
1201 TYPE*, ' ENTER CUTOFF'
READ(5,*) CUTOFF
1202 DO 311 I=1,LAST1
DO 311 J=1,LAST2
DO 311 K=1,LAST3
IF(X(I,J,K).GE.CUTOFF) THEN
IORGS(1)=I*IDBX*1000
IORGS(2)=J*IDBY*1000
IORGS(3)=-K*IDBZ*1000
CALL CLDFPI(LVL,IGG,ROT,SCALE,CLNAME,IORGS,IVIEW,IRC,IATLK,
1 ISPEC)
IF(IRC.NE.0) GO TO 1000
WRITE(6,36)
36 FORMAT(’CLDFPI COMPLETED’)
GO TO 1100
1000 WRITE(6,911)IRC
911 FORMAT(’ERROR IN CLDFPI IRC= ’,I3)
CALL ERROR9(IRC)
CALL ERROR
1100 CONTINUE
END IF
311 CONTINUE
GO TO 1200
1203 TYPE*, ' ENTER CUTOFF'
READ(5,*) CUTOFF
IREP=2

C DRAW THE FRONT WALL
C
CALL DRAW_SKIN(LVL,LAST2,LAST3,LAST1,IREP,CLNAME)
IREP=3

C DRAW THE LEFT WALL
C
CALL DRAW_SKIN(LVL,LAST1,LAST3,LAST2,IREP,CLNAME)
IREP=4
DRAW THE TOP WALL
CALL DRAW_SKIN(LVL, LAST1, LAST2, LAST3, IREP, CLNAME)
IREP=5

DRAW THE BACK WALL
CALL DRAW_SKIN(LVL, LAST2, LAST3, LAST1, IREP, CLNAME)
IREP=6

DRAW THE RIGHT WALL
CALL DRAW_SKIN(LVL, LAST1, LAST3, LAST2, IREP, CLNAME)
IREP=7

DRAW THE BOTTOM WALL
CALL DRAW_SKIN(LVL, LAST1, LAST2, LAST3, IREP, CLNAME)
IREP=2
GO TO 1200

1309  IARG=1
   IF (IRC.NE.0) IARG=0
   CALL DEDFPI(IARG)
   CALL FINODF
   RETURN
END

SUBROUTINE DRAW_CELL(N1,N2,N3)

AN INTEGER*2 ARRAY DESCRIBING ATTRIBUTE LINKAGE.
AN INTEGER*2 ARRAY WHICH DEFINES THE GRAPHIC GROUP NUMBER.
AN INTEGER*4 WORD ARRAY WHICH CONTAINS THE COORDINATES IN UOR'S OF THE SHAPE
AN INTEGER*2 VARIABLE WHICH DEFINES THE LEVEL ON WHICH THE GRAPHIC ELEMENT IS TO BE PLACED.
AN INTEGER*2 VARIABLE SPECIFYING THE NUMBER OF COORDINATE SETS IN A SHAPE.
AN INTEGER*2 WORD ARRAY WHICH DEFINES THE DISPLAY SYMBOLOGY.

THIS IS SYSTEM'S DFPI ROUTINES AVAILABLE IN CUSTOMERS SUPPORT LIBRARY. IT PLACES A SHAPE IN THE DESIGN FILE.

ERROR9 & THESE ROUTINES ARE USED TO TEST
ERROR CONDITIONS. THEY ARE
ALSO PART OF CUSTOMER SUPPORT
LIBRARY OF INTEGRAPH SOFTWARE

DIMENSION IX(8), IY(8), IZ(8)
COMMON X(20,20,20), IK(20,20)
* ITEST(20,20), CUTOFF, IDBX, IDBY, IDBZ, LARGEST
INTEGER*2 IG6(2), ISPEC(5), IATLK(4), LVL, IRC
INTEGER*4 ISHAPE(15), REGN50
INTEGER*2 IFILE(7), INBUF(15), ICHA
DATA IG6/2*O/, ISPEC/5*O/, IEX/EX/, NSHAPE/5/
DATA IATLK/4*O/, IN/5/, IOUT/6/
TYPE*, 'DO YOU WANT TO OPEN A NEW FILE ?'
READ(5,3) RESPONSE
FORMAT(A1)
IF(RESPONSE.EQ.'N'.OR.RESPONSE.EQ.'O') GO TO 1051
GO TO 1052
1051 IF(RESPONSE.EQ.'N') THEN
CALL NEW_FILE(ICHA, INBUF, IRC)
IF(IRC.NE.0) CALL ERROR
GO TO 1053
END IF
WRITE(IOUT,80)
FORMAT('ENTER OLD FILE NAME, DEV-UIC-NAME-EXT-VER')
READ(S,85) ICHA, INBUF
FORMAT(Q,15A2)

CALL LKTC3IS(INBUF, IFILE, ICHA, 0, IRC)
IF (IRC.NE.0) STOP '7-WORD LKTRAN CONV ERROR'
WRITE(IOUT,90)
FORMAT('ENTER REGION NAME, 6-CHARACTERS')
READ(S,85) ICHA, INBUF

CALL ASC2RD(INBUF, REGN50, ICHA, IRC)
IF (IRC.NE.0) STOP 'ASC2RD CONVERSION ERROR'

CALL INDFPI(REGN50, IFILE, 0, 0, 1, IRC, IEX)
WRITE(IOUT,72) IRC
FORMAT('INDFPI COMPLETED', I6)
IF (IRC.NE.0) CALL ERROR
LVL=1
DO 10 I=1,8
M1=0
M2=O 
M3=O 
IF(I.GT.4) M1=1 
IF(I.GE.3.AND.I.LE.6) M3=1 
IF(I.EQ.2.OR.I.EQ.3.OR.I.EQ.6.OR.I.EQ.7) M2=1 
IX(I)=M1*N1 
IY(I)=M2*N2 
IZ(I)=M3*N3 
CONTINUE
DO 11 I1=1,2 
M1=O 
IF(I1.GT.1) HI=I1 
DO 12 J1=1,3 
N=O 
NUM1=M1*N1 
NUM2=M1*N2 
NUM3=M1*N3 
DO 13 K1=1,8 
IF(J1.EQ.1) THEN 
IF(IX(K1).EQ.NUM1) GO TO 15 
END IF 
IF(J1.EQ.2) THEN 
IF(IY(K1).EQ.NUM2) GO TO 15 
END IF 
IF(J1.EQ.3) THEN 
IF(IZ(K1).EQ.NUM3) GO TO 15 
END IF 
GO TO 13 
N=N+1 
ISHAPE(3*N-2)=IX(K1)*1000 
ISHAPE(3*N-1)=IY(K1)*1000 
ISHAPE(3*N)=IZ(K1)*1000 
CONTINUE
ISHAPE(13)=ISHAPE(1) 
ISHAPE(14)=ISHAPE(2) 
ISHAPE(15)=ISHAPE(3) 
CALL SHDFPI(IGG,LVL,ISPEC,ISHAPE,NSHAPE,IRC,IATLK) 
IF(IRC.NE.0) GO TO 1000 
TYPE*,' SHDFPI COMPLETED' 
CONTINUE
GO TO 1209 
CONTINUE
CALL ERROR9(IRC) 
CALL ERROR 
CLOSE THE DESIGN FILE 
C 
IARG=1 
IF(IRC.NE.0) IARG=0 
CALL DEDFPI(IARG) 
CALL FINDDF 
TYPE*,' DO YOU WANT TO CREATE A CELL LIBRARY ALSO ? ENTER Y/N' 
C 
TYPE*,' ENTER N FOR NEW AND 0 FOR OLD' 
READ(5,3) RESPONSE
IF (RESPONSE.EQ. 'Y'.OR.RESPONSE.EQ. 'N') GO TO 1061
GO TO 1062
C1061 IF (RESPONSE.EQ. 'Y') THEN
OPEN A NEW FILE ON USER'S REQUEST
CALL NEW_FILE(ICHAF,INBUF,IRC)
IF (IRC.NE.0) CALL ERROR
END IF
RETURN
END

SUBROUTINE NEW_FILE(KOUNT,IBUFF,IRC)
IMPLICIT INTEGER*2 (A-Z)
CONTAINS THE NUMBER OF CHARACTERS
IN FILE SPECIFICATION
STORES THE FILE SPECIFICATION
INTEGER*2 IOBUFF(256)
BYTE IBUFF(32)

READ(5,2) KOUNT,(IBUFF(I),I=1,KOUNT)
FORMAT(Q,32A1)
WRITE(6,3)
FORMAT( 'HOW MANY BLOCKS OF STORAGE FOR FILE'/
& 'RECOMMENDED SIZE: 500')
READ(5,4) IBLOCKS
FORMAT(I5)

CREATES A NEW GRAPHIC DESIGN FILE
CALL FILALO(ILUN,IBUFF,KOUNT,IBLOCKS,IRC)
IF (IRC.NE.0) THEN
PRINT *, 'ERROR IN CALL FILALO'
RETURN
END IF
CLOSE(ILUN)

OPEN THE NEWLY CREATED DESIGN FILE FOR WRITING THE
THE CONTENTS OF SEEDZ.DGN IN IT SECTOR BY SECTOR
OPEN(UNIT=ILUN,FILE=IBUFF,ACCESS='DIRECT',RECORDTYPE='FIXED',
& RECL=128,STATUS='OLD',IOSTAT=IST)
WRITE(6,1000) IST
1000 FORMAT( 'STATUS OF OPEN OPERATION IST= ',15)
OPEN(UNIT=2,FILE='SEEDZ.DGN',ACCESS='DIRECT',RECORDTYPE='FIXED',
& RECL=128,STATUS='OLD',IOSTAT=IST)
WRITE(6,1000) IST
B-10

DO 5 I=1,5
   READ(2,I) IOBUFF
   WRITE(ILUN+1,I) IOBUFF
CONTINUE
CLOSE(ILUN)
CLOSE(2)
PRINT *,' VALID DESIGN/CEL FILE CREATED'
RETURN
END

SUBROUTINE CHANGE_SYMBOLOLOGY(ISPEC)

C
THIS SUBROUTINE ALLOWS THE USER TO CHANGE THE CLASS, STATUS
LINE STYLE, LINE WEIGHT AND COLOR OF THE ELEMENTS TO BE
PLACED IN THE GRAPHIC DESIGN FILE. THESE ATTRIBUTES OF THE
ELEMENT ARE STORED IN A INTEGER*2, 5 WORD ARRAY (ISPEC)
C

INTEGER*2 ISPEC(5)
DATA IN/5/IOUT/6/
WRITE(IOU,651)
FORMAT(1H1)
TYPE*, '
TYPE*, '
TYPE*, '
TYPE*, ' YOU HAVE FOLLOWING OPTIONS'
TYPE*, ' --------------------------
TYPE*, ' 1: CHANGE CLASS'
TYPE*, ' 2: CHANGE STATUS'
TYPE*, ' 3: CHANGE STYLE'
TYPE*, ' 4: CHANGE LINE WEIGHT'
TYPE*, ' 5: CHANGE COLOR CODE'
TYPE*, ' 6: EXIT THIS MENU'
TYPE*, '
TYPE*, ' ENTER THE NUMBER OF YOUR OPTION :
READ(IN,652) ANS
FORMAT(AI)
IF(ANS.EQ.'1'.OR.ANS.EQ.'2'.OR.ANS.EQ.'3'
  .OR.ANS.EQ.'4'.OR.ANS.EQ.'5'.OR.ANS.EQ.'6')
GO TO 670
WRITE(IOU,651)
TYPE*, ' PLEASE ENTER A NUMBER FROM THE MENU'
GO TO 660
IF(ANS.EQ.'6') RETURN
IF(ANS.EQ.'1') THEN
   WRITE(IOU,653) ISPEC(1)
   FORMAT(' THE EXISTING CLASS CODE IS ',I5)
   TYPE*, ' ENTER THE NEW CLASS CODE ( 0-5 )'
   READ(IN,*) ISPEC(1)
   GO TO 650
END IF
IF(ANS.EQ.'2') THEN

WRITE(IOUT,654) ISPEC(2)
FORMAT(' THE EXISTING STATUS CODE IS ',I5)
TYPE*,' ENTER THE NEW STATUS CODE ( 0 OR 1 )'
READ(IN,*) ISPEC(2)
GO TO 650
END IF
IF(ANS.EQ.'3') THEN
WRITE(IOUT,655) ISPEC(3)
FORMAT(' THE EXISTING STYLE CODE IS ',I5)
TYPE*,' ENTER THE NEW STYLE CODE ( 0-4 )'
READ(IN,*) ISPEC(3)
GO TO 650
END IF
IF(ANS.EQ.'4') THEN
WRITE(IOUT,656) ISPEC(4)
FORMAT(' THE EXISTING LINE WEIGHT CODE IS ',I5)
TYPE*,' ENTER THE NEW LINE WEIGHT CODE ( 0-31 )'
READ(IN,*) ISPEC(4)
GO TO 650
END IF
IF(ANS.EQ.'5') THEN
WRITE(IOUT,657) ISPEC(5)
FORMAT(' THE EXISTING COLOR CODE IS ',15)
TYPE*,' ENTER THE NEW COLOR CODE ( 0-255 )'
READ(IN,*) ISPEC(5)
GO TO 650
END IF
RETURN
END
SUBROUTINE DRAW_SKIN(LVL,N1,N2,N3,IREP,CLNAME)
C
C THIS SUBROUTINE DRAWS ONLY THE SKIN OF THE ORE BODY.
C
COMMON X(20,20,20),IK(20,20)
* ,ITEST(20,20),CUTOFF,IDX,Y,IBY,IBZ,LARGEST
BYTE CLNAME(6)
INTEGER*2 IGG(2), ISPEC(5),IATLK(4),LVL
INTEGER*4 IORGS(3),REGN50
REAL*8 ROT,SCALE(3)
INTEGER*2 IFILE(7),INBUF(15),ICELL(7),IVIEW(2)
DATA IGG/2*0/,ISPEC/5*0/,IATLK/4*1/,LVL
DATA ROT/0.0/,SCALE/3*1.0/
DO 501 J3=1,LARGEST
DO 501 L3=1,LARGEST
ITEST(J3,L3)=0
IK(J3,L3)=0
501 CONTINUE
ISTEP=1
IFIRST=1
LAST=N3
IF(IREP.GT.4) THEN
ISTEP=-1
IFIRST=N3
LAST=1
DO 501 J3=1,LARGEST
DO 501 L3=1,LARGEST
ITEST(J3,L3)=0
IK(J3,L3)=0
501 CONTINUE
ISTEP=1
IFIRST=1
LAST=N3
IF(IREP.GT.4) THEN
ISTEP=-1
IFIRST=N3
LAST=1
END IF
DO 10 J=1,N1
DO 20 K=1,N2
DO 30 I=IFIRST,LAST,ISTEP
IF(IREP.EQ.2.OR.IREP.EQ.5) GO TO 13
IF(IREP.EQ.3.OR.IREP.EQ.6) GO TO 12
IF(X(J,K,I).GT.CUTOFF) GO TO 40
GO TO 30
12 IF(X(J,I,K).GT.CUTOFF) GO TO 40
GO TO 30
13 IF(X(I,J,K).GT.CUTOFF) GO TO 40
GO TO 30
40 IK(K,J)=I
ITEST(K,J)=1
GO TO 20
30 CONTINUE
20 CONTINUE
10 CONTINUE
K1=1
J1=1
100 IF(ITEST(K1,J1).EQ.1) THEN
IA=K1
IB=J1
IC=IK(K1,J1)
GO TO 105
END IF
GO TO 1100
105 IF(IREP.EQ.2.OR.IREP.EQ.5) THEN
N4=IC
N5=IB
N6=IA
END IF
IF(IREP.EQ.3.OR.IREP.EQ.6) THEN
N4=IB
N5=IC
N6=IA
END IF
IF(IREP.EQ.4.OR.IREP.EQ.7) THEN
N4=IB
N5=IA
N6=IC
END IF
IORGS(1)=N4*IDBX*1000
IORGS(2)=N5*IDBY*1000
IORGS(3)=-N6*IDBZ*1000
C
C PLACE THE CELL IN THE DESIGN FILE
C
CALL CLDFPI(LVL,IGG,ROT,SCALE,CLNAME,IORGS,IVIEW,IRC,IATLK,
1 ISPEC)
IF(IRC.NE.0) GO TO 1000
WRITE(6,36)
36 FORMAT(' CLDFPI COMPLETED')
GO TO 1100
1000 WRITE(6,911)IRC
911 FORMAT(' ERROR IN CLDFPI IRC= ',I3)
   CALL ERROR9(IRC)
   CALL ERROR
1100 IF(K1.EQ.N2.AND.J1.EQ.N1) GO TO 210
   IF(J1.EQ.N1) THEN
      K1=K1+1
      J1=0
   END IF
   J1=J1+1
   GO TO 100
210 RETURN
END