A COMPUTER MODEL FOR TUNNELING COSTS

by

Thomas J. Lamb

S.B., Massachusetts Institute of Technology
1969

Submitted in partial fulfillment
of the requirements for the degree of
Master of Science
at the
Massachusetts Institute of Technology
June 1971

Signature of Author. . . . . . . . . .
Department of Civil Engineering, May 14, 1971

Certified by . . . . . . . . . . . . . . . . . . .
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . .
Chairman, Departmental Committee on Graduate Students
ABSTRACT

A COMPUTER MODEL FOR TUNNELING COSTS

by

THOMAS JAMES LAMB

Submitted to the Department of Civil Engineering on May 14, 1971, in partial fulfillment of the requirements for the degree of Master of Science.

Tunnels are being used more and more to solve the pressing environmental and social problems of our society. They are very expensive, but little is known concerning the influence of various factors upon the cost. An approach has been developed which uses a computer to model tunnel construction. This model is detailed enough to permit cost estimation, sensitivity analysis, and the evaluation of new construction techniques.

The model consists of four stages: setting up the data, determining the cost of each segment, determining the cost of each reach, and finding the minimum cost. Event simulation is used to model the excavation process. Simulation or network techniques are used to model particular operations. Decision C.P.M. is used to find the cost of the reach. The first stage has been implemented on the MULTICS time sharing system at M.I.T. It is written in the PL/1 programming language.

Thesis Supervisor: Professor Fred Moavenzadeh
Title: Associate Professor of Civil Engineering
ACKNOWLEDGEMENT

I wish to thank my advisor, Professor Fred Moavenzadeh, for his constant encouragement of this work and much helpful advice. Professor Frederick J. McGarry and Professor Ronald C. Hirschfeld of M.I.T. and Mr. Frank Wheby of the Harza Engineering Company also provided many helpful discussions. Special thanks is due to Christopher Herot who wrote the programs and whose knowledge of programming was an invaluable aid. Debbie Allinson provided much needed encouragement and moral support. Rosemary Dujsik typed the thesis quickly and accurately.

Many other persons, too numerous to identify, lent encouragement and provided information, for which I am very grateful.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Title Page</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>2</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>3</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>4</td>
</tr>
<tr>
<td>List of Figures</td>
<td>7</td>
</tr>
<tr>
<td>List of Tables</td>
<td>8</td>
</tr>
<tr>
<td>Chapter One</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>9</td>
</tr>
<tr>
<td>Chapter Two</td>
<td></td>
</tr>
<tr>
<td>Tunnel Cost Estimation</td>
<td>16</td>
</tr>
<tr>
<td>A. Estimation Procedures</td>
<td>17</td>
</tr>
<tr>
<td>B. Tunnel Estimation Procedures</td>
<td>18</td>
</tr>
<tr>
<td>Chapter Three</td>
<td></td>
</tr>
<tr>
<td>The Model Framework</td>
<td>25</td>
</tr>
<tr>
<td>A. Approach</td>
<td>26</td>
</tr>
<tr>
<td>B. Logic</td>
<td>28</td>
</tr>
<tr>
<td>C. Framework</td>
<td>42</td>
</tr>
<tr>
<td>D. Tunnel Definition Language</td>
<td>45</td>
</tr>
<tr>
<td>E. Conflict Analyzer</td>
<td>46</td>
</tr>
<tr>
<td>F. Physical Quantities Analyzer</td>
<td>52</td>
</tr>
<tr>
<td>G. Prescheduler</td>
<td>55</td>
</tr>
<tr>
<td>H. Time Analyzer</td>
<td>55</td>
</tr>
<tr>
<td>I. Scheduler for Segments</td>
<td>56</td>
</tr>
<tr>
<td>J. Preliminary Cost Analyzer</td>
<td>57</td>
</tr>
</tbody>
</table>
Chapter Four

Conclusions and Discussion

References

Appendix A
Tunnel Construction

Part I: Components of the Tunneling Cycle
A. Excavation
B. Mucking
C. Support
D. Lining
E. Groundwater Control
F. Ventilation
G. Surface Plant
H. Safety

Part II: Tunnel Costs
A. Tunnel Cost Components
B. Influence of Geology and Groundwater
C. The Effect of Geometry
D. The Effect of Excavation Method
E. The Effect of Mucking Method
F. The Effect of Support and Lining Systems
LIST OF FIGURES

Figure 1: Projected Tunneling Activity 11
Figure 2: Cost Increases in Tunnels and Hydroelectric Projects 13
Figure 3: Excavation Cost vs. Diameter 20
Figure 4: Harza Cost Equation 23
Figure 5: Bledsoe's Approach 24
Figure 6: Network for Reach Construction 30
Figure 7: Model for the Drilling Event 40
Figure 8: Drilling Submodel 41
Figure 9: Logical Setup of Model 43
Figure 10: Flow Chart of Proposed Program 44
Figure 11: Tunnel Definition Language Example 49
Figure 12: Tunnel Size Definition 51
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D.C.P.M. Operations</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>D.C.P.M. Constraints</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td>Events for the Excavation Simulation</td>
<td>34</td>
</tr>
<tr>
<td>4</td>
<td>Drilling Crew</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>Events in Simulation for Drilling Crew</td>
<td>37</td>
</tr>
<tr>
<td>6</td>
<td>Events in Simulation for Surface Crew</td>
<td>38</td>
</tr>
<tr>
<td>7</td>
<td>Construction Methods</td>
<td>47</td>
</tr>
<tr>
<td>8</td>
<td>Conflict Criteria</td>
<td>48</td>
</tr>
</tbody>
</table>
CHAPTER ONE

Introduction
INTRODUCTION

The space beneath our cities can be used for transportation, service, and possibly living space. The interior of the earth has been utilized to a very limited extent for subways, deep basements, and depressed highways, but the potential exists for a much greater utilization of underground space. It is expected that efforts to better utilize this space will be substantially increased during the next several years. The facilities to provide water and power services frequently make use of underground chambers or conduits, and will do so even more in the future, to avoid surface congestion.

Several studies have been made in an attempt to predict future tunneling activity. Figure 1 shows the results of two such studies, one by the Committee on Rapid Excavation of the National Research Council (1968) and one by the U.S. delegation to the Organization for Economic Cooperation and Development (1970). The two studies indicate a growth in demand of more than 100% every ten years, but the absolute predicted demand, in dollars, varies widely. This figure indicates that demand will be large, will grow rapidly, and is extraordinarily difficult to predict. In another study, Operations Research, Inc. (Lago et al, 1967) indicated that the demand for tunnels is a function of their cost.

Even if demand could be predicted, the costs associated with these tunnels would not be well known. They cannot
OECD

COMM. ON RAPID EXCAVATION

OECD

COMM. ON RAPID EXCAVATION

1960-1970
1970-1980
1980-1990

BILIONS OF DOLLARS

FIGURE 1 PROJECTED TUNNELING ACTIVITY
be extrapolated using a cost index such as that given in Engineering News-Record. Figure 2 shows past costs of tunnels and of hydroelectric projects, which are more or less typical of heavy construction. Tunnel costs seem at present to be increasing less rapidly than costs for heavy construction in general. If this trend continues, tunneling will become more and more attractive when compared with surface construction.

The use of a cost index implies a base cost to which the index can be applied, and this cost must be estimated. Until very recently, satisfactory methods for predicting the cost of a tunnel quickly and accurately did not exist, and producing the base estimate was a major task. Even very careful estimates, such as a contractors bidding estimate, or the engineers estimate, frequently disagree with each other and with the final cost. There are two major reasons for this. The cost of a tunnel is highly dependent upon the geologic conditions encountered, and geologic conditions are rarely well known. It is desirable to estimate the cost of a tunnel for several geologies, so that a range of costs is available and construction strategies for use in emergencies can be worked out, but this is rarely done, because of the time required to make accurate estimates.

The second reason for the inaccuracy of tunnel cost estimates in the dependence of cost upon the method used and upon the interactions between the operations occurring under-
Source:
Engineering News-Record
December 17, 1970

**Figure 2** Cost Increases in Tunnels and Hydroelectric Projects
ground. The various components of a tunnel (excavation, mucking, support, etc.) are usually given separate costs, but this is somewhat misleading, since they are really interdependent. Alternate construction methods are rarely investigated in detail, and the simplified estimating procedures that are used to investigate alternatives can be inaccurate. Appendix A discusses the tunneling process and the factors which influence cost.

Considerable work is being done in an attempt to determine geologic conditions with more accuracy. Research is also in progress evaluating new construction methods. Some procedure is needed to evaluate these new methods of excavation with respect to cost. Also, an improved method of estimating costs is needed, so that costs can be determined more rapidly and with greater accuracy. A computer program which can estimate costs in great detail for various construction methods fits these requirements. The program should take into account what is happening underground minute by minute. The large number of calculations necessary can be done quickly and accurately, so more alternatives can be investigated.

It is the purpose of this thesis to develop the logic for such a program. A complete implementation of the detailed approach outlined herein is beyond the scope of this work, but a considerably more limited version, which illustrates the basic approach taken, has been implemented.
Suggestions are presented for the complete implementation, but the details are lacking. The program has been written in PL/1 on the MULTICS system (time sharing) at M.I.T. This language and system were used because of the relative ease with which they can handle large quantities of data.
TUNNEL COST ESTIMATION

Estimation Procedures

There are several methods of cost estimation currently in use, and probably as many variations as there are estimators. Three basic approaches will be discussed below. Each has its advantages and disadvantages, and which is the "best" depends upon the purpose for which the estimate is made.

Costs are sometimes collected from previous projects and used as a basis for predicting future costs, often coupled with a cost index. In order for the estimate to be reasonable, the project to be estimated must closely match the projects from which data is obtained, both in design and in construction procedure. This type of estimate is fairly simple to make and does not require a detailed knowledge of design or construction on the part of the estimator. Thus, it is useful for preliminary planning. It is easy for an estimate of this type to be misleading, however, especially if significantly different construction techniques are used.

The second type of estimate is based upon unit costs. These can be either calculated, based upon the operations necessary to do the job, or they can be based on costs from previous jobs. This type of estimate requires more knowledge of design and construction than the previous type, and takes longer to do, but it is more detailed and more reliable. The so-called "law of averages" usually insures that even if the exact construction or design details are not known, the
The total cost will be reasonable. This is the type of estimate used most often in feasibility studies. The reliability of this type of estimate is quite variable, depending upon how the unit costs are obtained.

The third type of estimate is a very detailed calculation based upon the exact knowledge of how an operation is to be performed, who is to do it, how long it will take, and the labor, equipment, and material prices. This type of estimate requires a detailed knowledge of the project design and elaborate preplanning of the work. Hence, it is a difficult and time consuming type of estimate to make. On the other hand, it is probably the most reliable estimate possible. The basic cost data (wages and equipment and material prices) are fairly easy to obtain, but productivity of men and machines is difficult to evaluate. New techniques can be incorporated into the estimate if their characteristics are known. Deathridge (1965) and Parker (1970) give excellent treatments of this type of estimate.

Contractors often use this type of estimate on big jobs, since it is rather easy to adapt to C.P.M. or PERT scheduling techniques. This method has the additional advantage of specifying the details of each operation, so that if costs are excessive, the operations departing from assumptions can be pinpointed and perhaps rectified.

**Tunnel Estimation Procedures**

The first fairly detailed and systematic procedure for
the estimation of tunnel costs was published by the California Department of Water Resources (1959). The Department was concerned with various alternative aqueduct systems to serve southern California, and they developed a systematic cost estimation procedure for planning purposes.

Using tunnel diameter and geologic conditions as their basic variables, the Department evaluated many previous tunnel jobs and constructed charts giving the costs of the various components versus these variables. These charts provide a quick and fairly accurate estimate of the cost of constructing a non-pressure water tunnel in California in 1957. This procedure is a vast improvement over a completely separate estimate for each tunnel. It is also instructive to see graphically how various factors influence cost. Figure 3 shows one of their charts, plotting cost vs. diameter for various geologies.

The procedure developed in California has many drawbacks which make it difficult to use in general. It gives accurate results only for a specific area of the U.S. at a specific time. Hirschfeld (1965) has updated the costs in this report to 1964 using the Engineering News-Record Cost Index, but the applicability of this Index to tunnels is somewhat questionable, and no provision is made for costs in different regions of the country. It may well be easier to derive new cost curves than to compute an accurate cost index.
FIGURE 3  EXCAVATION COST VS. DIAMETER
The California method takes account only of drilled and blasted tunnels, driven from portals. Extensive additional computations are required if tunneling machines are used or if the tunnel is driven from a shaft. Also, the design of the supports and lining and the advance rates are fixed, and there is no easy way to incorporate different designs into the charts. In spite of these drawbacks, the California charts are useful in obtaining a first approximation of tunnel costs, if care is taken in projecting prices to today's levels.

More recently, (1968, 1970) the Harza Engineering Company has developed cost estimation procedures for the U.S. Department of Transportation. The aim of these studies was to provide cost estimates for the tunnels in a high speed rail transportation system in the Northeast Corridor. Their first effort, (March, 1968) was set up much like the California estimation procedure, except that the breakdown into components was a bit different and tunneling machines were considered. Their geological classification system is difficult to use and the tunnel design is fixed. The prices given are for Chicago in 1967, and no criteria are given for modification for different times and places.

The second effort of this company utilized a computer to estimate the costs. The computer program is considerably more flexible than the previous procedure, and various combinations of excavation, mucking, support, and lining
methods can be evaluated. The geological classification system is much better, and attempts have been made to extend the estimates to other parts of the country.

The equations which the computer uses to estimate costs (see Figure 4) were derived by estimating costs for various cases (in detail), deciding what factors influence the cost, and plotting these costs versus the factors to obtain a cost equation. The design details are built in, as are the details of all construction procedures. Nevertheless, this program represents a great improvement over previous procedures.

There are many other cost estimation procedures used by individuals and agencies which have not been published. It is probably safe to say that each estimator has his own more or less systematic procedure. These procedures are usually quite detailed, cumbersome, and difficult to explain. Bledsoe (1970) suggests that a much better approach would be to calculate the cost of a tunnel by knowing what is happening at the face in great detail and using a computer to calculate the cost directly. This "model" of underground construction could then be modified, and the effect upon cost noted.

Bledsoe's procedure is outlined in Figure 5. The approach suggested in the next chapter is similar to Bledsoe's approach in that both consider the tunneling operation in great detail, but the techniques used are quite different.
COST EQUATIONS - RAIL MUCKING

LABOR (%/hr.) \( UCL = 15E + 21(DM)^{0.82} + \frac{A}{12} \)

EQUIPMENT (%/hr.) \( UCL = 5.8E + C + 1.4DM \)

MATERIAL (%/y-mi) \( UCM = 0.03 \)

\( E = \) NUMBER OF ENGINES
\( C = \) NUMBER OF CARS
\( A = \) ADVANCE RATE (ft/day)
\( DM = \) HAUL DISTANCE (miles)

AFTER HARZA (1970)

FIGURE 4 HARZA COST EQUATION
FIGURE 5 BLEDSOE'S APPROACH
CHAPTER THREE

The Model Framework
Tunnel construction today is largely a trial and error process. Very little analysis of the actual construction process has been performed, the excuse usually being that geology is too variable. A computer program capable of analyzing construction should be fast enough to calculate results for various geologic conditions and overcome this difficulty. The framework for such a model is presented in this chapter. The goals of such a program are first discussed and the approach taken to meet these goals is outlined. The logic of the program and the method of analysis is then described. Finally, a framework for implementation and partial results are presented.

Approach

Three goals should be met by the program:

1) Accurate Cost Estimation.
2) The ability to perform sensitivity analysis and to evaluate new construction methods.
3) Convenience and ease of use.

These goals determined the approach taken.

Cost estimates which are reasonably accurate can be obtained by using COHART (the program developed by Harza Engineering) or the modification of this program presented in Appendix B. If construction cost is all that is desired, and if current methods are to be used, it is not necessary to use a more detailed program. A detailed program will
give accurate cost estimates also, but improvement will be marginal.

A detailed approach is necessary for sensitivity analysis. It is impossible to determine the effect of, for example, drilling crew size upon cost unless the effect of a change in drilling time upon the mucking or support operations can be specified. The relation is not simple, because crews must be transported to the face, work areas are crowded and operations interfere, and there are many alternate ways of scheduling the operations underground. It is extremely simplistic to look at tunnel construction as a collection of independent events, and any sensitivity analysis based upon this assumption may be misleading. Much the same thing can be said for the evaluation of a new construction technique; interactions cannot be ignored.

A detailed approach has the advantage that the cost information required (wages, equipment operating costs, and material costs) is much easier to obtain than the cost per foot for some operation. Cost per foot depends upon geology, geometry, method used, schedule, equipment, etc., and the relationship is anything but clear. (It should be noted that a cost model capable of sensitivity analysis can help clarify these relationships).

Defining exactly what is happening during construction is very difficult because activities vary from tunnel to tunnel. The construction details are set by the contractor based
largely upon his feeling for what is most effective. A de-
tailed program must either contain assumptions as to the
schedule of allow the user to specify it. The most convenient
way for a user to specify the schedule is through the use of
time sharing. With this type of system, the user can inter-
act with the model and modify his assumptions based upon
results.

The amount of time a given operation takes is not well
defined. It is possible to derive time distributions for
certain operations, either based upon past data or assumed,
and use simulation to determine the time for the entire
process. Simulation can also be used when productivities or
efficiencies are not known exactly

A detailed program should model the tunnel construction
process operation by operation, determining progress, the
amount of time each man works, and the amount of time each
piece of equipment is in operation. A full day (three shifts)
must be modeled to take care of variations in scheduling and
transportation. The user must define how things are to be done.

Tunnel construction must be analyzed at three levels of
detail. The reach level is the most general level. A reach
is that portion of the tunnel excavated from one shaft or
portal (c.f., Harza, 1970). The cost of the reach includes
the cost of tunnel construction, the cost of the plant and
surface equipment, the cost of the shaft or portal, and
overhead, profit, and interest. The reach can be modeled using a decision C.P.M. network as shown in Figure 6. Table 1 lists the operations and Table 2 lists the constraints. This network can be solved using integer linear programming (Crowston & Thompson, 1965).

Costs and times for each reach operation can be determined by modeling it. Clearing operations, shaft construction, and finishing operations will not be considered further, but they could be modeled in a manner similar to excavation. The determination of excavation cost requires that the segments within the reach be modeled.

A segment is a portion of the tunnel in which everything is constant. After a certain number of segments have been constructed, some of the plant may be moved underground. If the cost of construction is computed both with the move and without the move, the D.C.P.M. reach network will tell which is the best solution.

Segment construction can be modeled using event simulation. Several time periods are of interest. The cycle is basic to tunneling and must be kept in mind. A shift must be modeled in order to determine transportation costs. A day, or three shifts, may have to be modeled, depending upon scheduling. Table 3 lists the events which might be used in a simulation of excavation. The events can be ordered by the user. Defining the sequence of events will determine the schedule underground.
FIGURE 6 NETWORK FOR REACH CONSTRUCTION
1. Build Site Access
2. Clear Site
3. Dummy
4. Setup for Shaft or Portal Construction
5. Build Surface Plant
6. Build Shaft or Portal
7-1. Underground Setup
7-2. Dummy
8-1. Excavate and Line Tunnel
8-2. Excavate Tunnel
9-1. Excavate and Line Tunnel
9-2. Excavate Tunnel
10-1. Underground Setup
10-2. Excavate and Line Tunnel
11-1. Underground Setup
11-2. Excavate Tunnel
12-1. Underground Setup
12-2. Excavate and Line Tunnel
13-1. Underground Setup
13-2. Excavate and Line Tunnel
14. Excavate and Line Tunnel
15. Excavate Tunnel
16. Excavate and Line Tunnel
17. Excavate Tunnel

**TABLE 1. D.C.P.M. Operations**
18-1. Line Tunnel
18-2. Dummy
19-1. Cleanup
19-2. Finish Shaft or Portal
19-3. Dummy
20-1. Finish Shaft or Portal
20-2. Finish Shaft or Portal
20-3. Line Tunnel
21-1. Line Tunnel
21-2. Cleanup
21-3. Cleanup
22. Finish Shaft or Portal
23. Cleanup
24-1. Line Tunnel
24-2. Dummy
25. Line Tunnel
26. Finish Shaft or Portal
27. Cleanup
28. Line Tunnel
29. Final Cleanup

TABLE 1 (continued)
\begin{align*}
\text{TABLE 2: D.C.P.M. Constraints} \\
\end{align*}
1. Day Start
2. Drilling Start
3. Drilling Finish
4. Grouting Start
5. Grouting Finish
6. Support Start
7. Support Finish
8. Supply Start
9. Supply Finish
10. Utility Extension Starts
11. Utility Extension Finish
12. Scaling Start
13. Scaling Finish
14. Lining Start
15. Lining Finish
16. Blasting Starts
17. Blasting Finish
18. Surface Work Starts
19. Surface Work Finish
20. End of Shift
21. Start of Shift
22. Lunch
23. Cycle Delimeter
24. Dummy Event
25. End of Day

TABLE 3: Events for the Excavation Simulation
Each event in Table 3 represents the start or completion of an operation. Each operation can in turn be modeled by a simulation, and the time of occurrence of the start of the next operation or the finish of the operation in question can be determined. The solution of the main simulation will give construction time. (Cost is determined by the operation simulations). The schedule can be changed and the simulation run again to determine the sensitivity of cost and completion time to scheduling.

Interactions between operations can be modeled by interactions among the operations simulations. For this purpose, and for cost computations, the groups of men performing different operations must be separated. Crews can be defined for this purpose. Table 4 lists the members of the drilling crew. The user must specify the size and composition of each crew and their wages. The actions of each crew can be simulated separately, with the simulations interacting. An example will clarify this.

The day start event will initiate the two sequences of events listed in Tables 5 and 6. (Additional simulations will be initiated for the other crews). The drilling crew cannot descend the shaft until the surface crew has readied the hoist. There may be additional interactions with other crews. The user must specify these relationships. Simulation languages such as SIMSCRIPT II make this relatively easy.
1 Foreman
N Drillers
N Helpers
1 Supply Man

NOTE: Crew Size must be Defined by the User

TABLE 4: Drilling Crew
1. Clock Start
2. Arrive at Tool Crib
3. Depart Tool Crib
4. Arrive Shaft - Join Queue
5. Start Descent
6. End Descent
7. Arrive Tool Area
8. Depart Tool Area
9. Join Transportation Queue
10. Begin Boarding Transportation
11. Leave Shaft
12. Arrive at Face
13. Walk to Work Area
14. Begin Drilling Operations

TABLE 5. Events in Simulation for Drilling Crew
1. Clock Start
2. Open Crib
3. Hoist Ready
4. Ventilation System Started
5. Compressors Started
6. Generators Started
7. Begin Readying Muck Disposal System
8. Begin Startup of Concrete Batch Plant
9. Begin Surface Operation Model

TABLE 6. Events in Simulation for Surface Crew
Each event in Tables 5 and 6 can be modeled in as much detail as desired. Sufficient accuracy can probably be obtained by simply specifying a time distribution for each event. Costs for the men can be determined by multiplying times by wages. Time can be further broken down into productive time and lost time if desired. The equipment usage time can also be collected and equipment cost calculated.

Event 14 in Table 5 triggers the start of the drilling event in Table 3. If actual drilling is unaffected by other events, techniques more efficient than simulation can be used. Figure 7 shows a GERT (Graphical Evaluation and Review Technique, Pritsker, 1966) network which can be used to simulate drilling. Programs exist for the solution of GERT networks. The solution of this network will give the time and cost for drilling. For each branch, crew wages and equipment operating costs must be specified. Material cost (bits and power) can be computed separately. The probability of completing the drilling on schedule can also be obtained from the GERT network.

It may be desirable to model certain branches of this network in more detail. Figure 8 shows a submodel for the drill branch. This GERT network can be solved for the time and cost of actual drilling. The user must specify time and cost information for each branch and must supply the parameters used to determine n.

Each event in Table 3 will have a simulation associated
1-2a. Advance Sliding Floor
1-2b. Lay Track
1-2c. Bring up Drills
1-2d. Bring up Jumbo
2-3a. Bring up Jumbo
2-3b. Dummy Branch
2-4a Bring up Drills
2-4b Dummy Branch
3-4 Position Jumbo
4-5 Position Drills
5-6 Connect Utilities
6-7 Drill
7-8 Disconnect Utilities

One Branch Chosen by User
One Branch Chosen by User
One Branch Chosen by User

FIGURE 7 MODEL FOR THE DRILLING EVENT
1-2 Dummy Branch
2-3 Drill, Repeat n Times
3-2a. Change Steel
3-2b. Change Bit
3-2c. Reposition Drill
3-4 Blow Out Holes

\[ n = N \times C + N + W \text{ (integer)} \]
\[ N = \text{Number of Holes} \]
\[ C = \text{Steel Changes/Hole} \]
\[ W = \frac{N \times \text{Depth}}{\text{Bit Life}} \quad \text{or} \quad C = 0 \]

Prob. 3-2a = \( \frac{N \times C}{n} \)
Prob. 3-2b = \( \frac{W}{n} \)
Prob. 3-2c = \( \frac{N}{n} \)

FIGURE 8 DRILLING SUBMODEL
with it. Where interactions can be ignored, efficient computational methods such as GERT can be used. Where events interact, event simulation is probably a better procedure. Interactions and sequences must be defined by the user. Once the main simulation of excavation has been constructed it should be run several times to obtain a reasonable cost-advance relationship. The simulations can be constructed to reflect delays due to unexpected geologic conditions or equipment breakdowns by adding a breakdown or delay event and specifying its interactions with other events. In addition, more detail can be added to the analysis. Running time on the computer may be a limitation in this respect. Figure 9 summarizes the logic for tunnel construction modeling. Several reach models can be combined to form a model for the entire tunnel, as shown here.

The Framework

A framework for the implementation of the model has been developed and is shown in Figure 10. Four stages are used: defining the tunnel, modeling segment construction, and cost minimization.

The first stage consists of the tunnel definition language and the conflict analyzer. These two processors define the geologic and geometric properties of the tunnel and the construction methods to be used. The tunnel definition language has been implemented on the Multics time sharing system at M.I.T. It is written in the PL/l
FIGURE 9 LOGICAL SETUP OF MODEL
FIGURE 10 FLOW CHART OF PROPOSED PROGRAM.
programming language. A listing and users manual is presented in Appendix C.

**Tunnel Definition Language**

The tunnel definition language consists of 6 programs: initiate-tunnel, initiate-symtab, initiate-methods, modify-methods, modify-symtab, and tdl. Initiate-tunnel names the tunnel, defines the maximum number of reaches, shafts, and segments in the tunnel, and sets up a data structure to hold geologic, geometric, and method data for each segment. tdl can be called either automatically by initiate-tunnel or, if the information about an existing tunnel is to be changed, directly by the user. tdl asks for the shaft where each reach originates and its destination shaft. It also requests names for the segments in each reach and geologic and geometric information for each segment. The number of shafts, reaches, and segments named in tdl can be less than the number specified in initiate-tunnel.

Initiate-symtab tells the computer where the symbol table is. The symbol table contains the names given to each item of geologic and geometric data and its location in the array where it is stored. Modify-symtab can be used to add or delete names.

Initiate-methods identifies a table of methods to the computer. The user can then describe what construction methods he wants to consider through the use of modify-methods. Methods which might be considered are listed in
Table 7. Initiate-methods also identifies a table of criteria used by the conflict analyzer to eliminate methods. Table 8 lists some sample criteria. Modify-methods can be used to add, delete, or change criteria. Figure 11 shows an example of the use of the tunnel definition language. For more details, refer to Appendix C.

The geologic information needed is R.Q.D. (see Deere et al, 1969), rock compressive strength (psi), initial groundwater inflow (gallons per minute at the face), steady state groundwater inflow (g.p.m./foot), and rock temperature. R.Q.D. or rock strength may be specified as a probability distribution with a given mean and standard deviation. The geologic information must be obtained from a geologic exploration of the tunnel site. Great accuracy cannot be expected.

Geometric information required is tunnel size and shape, slope, depth, and length. These parameters are determined by the tunnel design. The size and shape are specified by a radius and center co-ordinate for circular tunnels and by four co-ordinates and four radii for non-circular tunnels. Refer to Figure 12. A radius of zero indicates a flat surface. The size specified is finished size. The use of this type of specification permits any size or shape of tunnel to be analyzed.

**Conflict Analyzer**

The conflict analyzer will eliminate construction
<table>
<thead>
<tr>
<th>Excavation Methods</th>
<th>Grouting Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drill and Blast--Conventional</td>
<td>Pregrouting</td>
</tr>
<tr>
<td>Heading and Bench</td>
<td>Chemical</td>
</tr>
<tr>
<td>Multiple Drift</td>
<td>Concrete</td>
</tr>
<tr>
<td>Mechanical Mole</td>
<td>Grouting</td>
</tr>
<tr>
<td></td>
<td>Chemical</td>
</tr>
<tr>
<td></td>
<td>Concrete</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Support Methods</th>
<th>Pumping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steel Supports</td>
<td>Pumps</td>
</tr>
<tr>
<td>Rock Bolts</td>
<td>No Pumps</td>
</tr>
<tr>
<td>Shotcrete or Gunnite</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lining Methods</th>
<th>Mucking Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Electric Train</td>
</tr>
<tr>
<td>Concrete</td>
<td>Diesel Train</td>
</tr>
<tr>
<td>Shotcrete</td>
<td>Electric Truck</td>
</tr>
<tr>
<td></td>
<td>Diesel Truck</td>
</tr>
<tr>
<td></td>
<td>Belt Conveyor</td>
</tr>
</tbody>
</table>

**TABLE 7. Construction Methods**
I. Geology

if strength ≥ 32,000 psi, mole eliminated
if RQD < 90, some support necessary
if RQD ≤ 25, steel sets necessary

II. Groundwater

if sustained inflow > 15 gpm, mole eliminated
if sustained inflow ≥ 0, some grouting or pumping needed
if sustained inflow > 100 gpm, grouting needed
if sustained inflow ≥ 500 gpm, pumps needed also
if initial inflow < 1000 gpm, no pregrouting
if initial inflow ≥ 50 gpm, mole eliminated
if sustained (after grouting) ≥ 5 gpm, lining must be used

III. Geometry

if shape not circular, mole eliminated
if diameter 15', truck mucking eliminated

IV. Method

if drill and blast used, no conveyor mucking
if pregrouting used, eliminate mole

V. Combinations

if RQD > 50, and if diameter ≤ 15', eliminate heading and bench
if RQD > 25, and if diameter ≤ 15', eliminate multiple drift
if RQD < 95, and if no lining used, support must be used
if RQD < 95, and if no support used, lining must be used
if inflow (sustained) ≥ 25 gpm, and if RQD ≤ 50, grouting required

TABLE 8: Conflict Criteria
tunnel test
syritab found
pdb found
new tunnel
how many reaches? 2
how many shafts? 3
how many segments? 4
TUL
specify components:
reach       shaft1       shaft2       segments
A           por1         sss          a b
B           por2         sss          c d
modify
>seg a
prompt
mean_rq 90.0
sigma_rq 10.0
mean_strength 25000.
sigma_strength 5000.
x_coordinate_1 0.0
x_coordinate_2 20.0
x_coordinate_3 20.0
x_coordinate_4 0.0
y_coordinate_1 20.0
y_coordinate_2 20.0
y_coordinate_3 0.0
y_coordinate_4 0.0
radius_1 20.0
radius_2 0.0
radius_3 0.0
radius_4 0.0
length 17500.
slope 0.01

,FIGURE II TUNNEL DEFINITION LANGUAGE EXAMPLE
(SEE APPENDIX B FOR EXPLANATION)
depth 3000.
init_gwat_inflow 0.0
sustained_gwat_i 0.0
rock_temp 50.0
shape 2

m: sup rblt shot
n: gap none

rch A
m: exc dabl mole
n: muck rail trck
n: lin none

1q

r 2120 11.484 500+254

FIGURE II (CONTINUED)
FIGURE 12 TUNNEL SIZE DEFINITION
methods which cannot be used because of geologic or geometric conditions. The purpose of the conflict analyzer is to reduce the number of computations required. If all possible combinations of the methods listed in Table 7 were used, almost 2000 cases would have to be considered. This would lead to the use of incredible amounts of computer time. The conflict analyzer applied criteria similar to those in Table 8 and can reduce substantially the number of cases considered. After the program has been used for a while, the criteria can be made more restrictive and computations further reduced. These changes would be based upon the program results. The conflict analyzer has not been implemented, but it is a fairly simple program.

The second stage of the analysis, the analysis of segments, consists of the physical quantities analyzer, the prescheduler, the time analyzer, the scheduler for segments, and the preliminary cost analyzer with their associated data. This stage designs the tunnel, sets up the construction schedule in each segment, and computes the cost of the segment.

**Physical Quantities Analyzer**

The physical quantities analyzer designs the support and lining system for the tunnel, calculates the amount of powder and number of holes needed (if drill and blast excavation is used), calculates overbreak and quantity of muck, and computes the amount of concrete and steel needed for the support and lining system. A rather simplified
analysis is presented below, but more sophisticated techniques would require more computer time and the improvement in design to be gained is dubious.

Deere's criteria can be used for design of support and lining (Deere et al, 1969). These criteria give support spacing and weight and lining thickness as a function of R.Q.D. The criteria will be stored in the table of design parameters. Criteria other than Deere's can be used, if it seems desirable. These criteria will be stored in much the same way as the criteria for the conflict analyzer. The physical quantities analyzer will determine spacing and thickness. Weight can then be computed from geometry.

A program has been written to perform powder calculations for blasting, based upon the results of Langeufors and Kihlström (1963). This program, presented in Appendix D, determines the length of holes, number of holes, size of holes, stemming, degree of packing, and quantity of explosive. Some of these parameters, in addition to area and advance per round, must be specified and the program will compute the others. The physical quantities analyzer can use this program to determine quantities of explosive, primers, and caps. It should be used twice, once for the cut and once for the rest of the face.

The area of the tunnel is equal to

\[ ((X_3 - X_4) + (X_2 - X_1))(Y_1 - Y_3) + 1/2R_1(\theta_1 - \sin \theta_1) + R_2(\theta_2 - \sin \theta_2) \]

53
\[ + R_3(\theta_3 - \sin \theta_3) + R_4(\theta_4 - \sin \theta_4) \]

where

\[ \theta_n = 2 \sin^{-1} \frac{D_n}{2R_n} \]

\[ D_n = \text{length of chord } n \quad (D_1 = [(X_3 - X_1)^2 + (Y_1 - Y_3)^2]^{1/2}) \]

Refer to Figure 12. An effective radius can now be computed and the tunnel calculations performed as if the tunnel were circular.

\[ R_{eff} = (\text{AREA/\pi})^{1/2} \]

This analysis assumes that the area/effective radius ratio does not change as tunnel size is increased. While not strictly true, this approximation is fairly accurate for small size changes.

Excavated diameter can be computed once the lining thickness is known.

\[ D_{exc} = 2(\text{lining thickness} + R_{eff}) \]

True excavated diameter must include overbreak

\[ D_{ext} = D_{ex}(1 + O.B.) \]

where

\[ O.B. = \% \text{ overbreak/100} \]

the quantity of muck produced can be expressed as:

\[ \pi(D_{ext}^2/4)(1 + B.F.) \]
where B.F. is the bulking factor. It may be desirable to make O.B. and B.F. random variables.

**Prescheduler**

The prescheduler sets up the construction sequence in the segment for a given construction method. The method procedure table contains lists of events similar to those in Table 3 for each construction method. The user must specify the order of the events. The construction description input is used to indicate user interaction with the program.

The events in the method procedure table can be identified by referring to the references on tunnel construction. The main events for mole excavation are the same as those in Table 3, but the subevents are quite different. The main events must have linkages to the approximate sub simulations for each method. These linkages can be programmed by using pointers.

**Time Analyzer**

The time analyzer is responsible for assigning times to the events in the lower level simulations, or to the branches of the lower level networks. These networks or simulations are linked to the main events as mentioned above. A few times can be specified as constant, such as "smoke time" or lunch time. Others must be computed. The construction data base contains the information needed to compute these times, such as drill penetration rate or equipment
productivity. The user must specify how much and what type of equipment is to be used. With this information and the output of the physical quantities analyzer, times can be assigned to the simulations. In many cases, a time distribution rather than a single time should be used. The user must also specify crew size and composition.

The computation of times is relatively straightforward. The time required to descend the shaft, for example, can be computed from shaft depth and hoist speed:

\[
\text{Time (sec)} = \frac{\text{Depth (Ft)}}{\text{Speed (f.p.s.)}}
\]

The constants (or distributions) required for the calculations can be obtained from equipment manufacturers. The California Department of Water Resources and the U.S. Bureau of Reclamation also have large stores of data. The construction data base and the table of standard times should have values in them. The user can change these values if he prefers his own or wishes to check sensitivity. The user must also tell which branches, in the case of networks, are to be deleted.

Scheduler for Segments

The entire simulation is set up and times assigned for all events by the time analyzer. The scheduler defines interactions and solves the simulation. The user must define the way events interact, as mentioned in the section on logic. Some knowledge of tunnel construction is required.
do this. This knowledge can be gained from the references. Many of the relationships are based on common sense.

After interactions have been specified, the segment simulation can be solved. Each crew's time is kept in an account, and can be separated into productive time, travel time, and lost time. Equipment operating time is also noted. The simulations must be written to collect these figures. The entire simulation should be run several times so that cost time curves can be developed for use in the reach D.C.P.M. network.

**Preliminary Cost Analyzer**

The cost analyzer determines the cost of constructing the segment using a given method. The times collected when the segment simulation is solved are multiplied by unit costs and summed to give total cost.

Crew cost is determined by adding up the wage rates of the members of the crew. These wage rates can be obtained from the U.S. Department of Labor. Provision must be made to pay overtime rates for hours worked in excess of 8 (or some other number).

Equipment operating cost is somewhat more difficult to obtain. Some information can be obtained from manufacturers or public agencies. Contractors are the best source of such data, but they are usually reluctant to share it. Operating costs can be calculated, based upon equipment cost and expected service life (available from manufacturers),
fuel and power consumption, and expected maintenance. The unit cost table should contain equipment operating costs for various pieces of equipment. The user can change the value if he desires.

When cost has been computed for one construction method in the segment, the program returns to the physical quantities analyzer and begins the analysis of the next method. When all methods have been considered, the next segment is analyzed. Finally, cost and completion time data are available for all methods in all segments, and the reach can be analyzed.

Conflict Analyzer

It is unlikely that construction methods will vary radically from segment to segment when the reach is constructed. The conflict analyzer forms combinations of segments with compatible (or the same) construction methods and the cost analysis of the reach is done for these combinations. For example, mole tunneling and drill and blast tunneling would not be used in alternating segments. Similarly, the same mucking method would be used throughout the reach.

Computation could be significantly reduced if this conflict analyzer were incorporated into the conflict analyzer for segments. It is not, in this framework, because the user may be interested in analyzing a particular segment using all methods feasible there, rather than being constrained by the other segments.
Main Scheduler

The main scheduler sets up and assigns times to the reach D.C.P.M. network. Setup times are the times required to construct the access roads and surface plant. Shaft or portal construction time and cleanup time is included here. The plant and equipment definitions tell what equipment is included in the surface plant. This information is stored for later use. The reach construction description input indicates that the user must interact with the program during the setup. He must specify any additional constraints upon the network and can change setup times or plant and equipment definitions.

Unit Cost Calculator

The unit cost calculator calculates operating costs for the surface plant. The user must supply the purchase price of the experiment, salvage value, discount rate, fuel and power consumption, etc. The program will convert this into a unit cost, which it will assign to the appropriate branches of the network.

Cost Analyzer

The cost analyzer solves the D.C.P.M. network for lowest cost. Overhead and profit are then added in and total reach cost computed. The externals are costs for branches not analyzed in detail and interest, penalties, or bonuses. These costs will be rather difficult to obtain. They can be obtained from past jobs or through crude cost
estimates. The best way is to perform a detailed analysis of every branch, but this is probably not feasible. It is also not strictly necessary for sensitivity analysis of construction.

Excavation cost is required for several branches of the D.C.P.M. network shown in Figure 6. It is necessary to return to the segment level to obtain these costs and solve a new excavation simulation for each branch. The user must specify which branch is being computed.

After the network has been solved for one construction method, the main scheduler sets up another network using a different method. This process is continued until all methods have been examined.

Cost Minimizer

The cost minimizer performs the very simple function of finding the lowest cost of those which have been computed and its associated construction method. This can be printed out and the next reach analyzed. Considering each reach separately prevents the user from modeling the transfer of men and resources from one reach to another. Another level of modeling can be added if this is needed.

This is not the only framework which can be used to implement detailed cost estimation program. It is intended primarily as a guide and may require modification. It appears to be reasonable, however, both from a logic standpoint and from the standpoint of programming.
CHAPTER FOUR

Conclusions and Discussion
Much work remains to be done on the model before conclusions can be drawn concerning the tunneling operation. Several man-years will be necessary to complete development. Conclusions can be drawn concerning the approach taken, based upon the work done in developing the logic.

1) A very detailed cost model is necessary for sensitivity analysis of tunneling or for the evaluation of new construction methods. All other methods proposed so far, with the possible exception of Bledsoe's, are neither designed for this purpose nor capable of performing such an analysis. Some insight will undoubtedly be gained from the construction of the model itself.

2) Event simulation is necessary for a large portion of the model. Very little is known, quantitatively, about the influence of construction techniques upon cost, and what little is known is based mostly upon empirical evidence. Simulation models can provide data so that more efficient but less general modeling techniques can be used in the future. At this stage of development, it is important to be as general as possible.

3) It is important not to attempt to do too much in one model. Simulation programs take a long time to run on a computer. The model described in this thesis should not require undue amounts of time, if only excavation is modeled in detail. The conflict analyzers reduce computations. A further reduction may
necessary. Brute force is used for optimization in this model; more sophisticated and efficient techniques can be used when more is known about tunneling.

4) A program using the approach described in Chapter 3 can be used for construction control, as well as for planning. If conditions are not as anticipated, the model can provide a revised schedule and cost estimate. Contingency plans can also be prepared with little effort.

5) Time sharing is the best computer environment for the model. The user must interact with the model in setting up the schedule. In addition, he may want results at any stage of the model and may wish to change one portion based upon the results. A time sharing system makes this relatively easy to do. Command interpreters can be written to minimize the amount of programming required of the user.

6) FORTRAN is a very bad language to use for this model. The limited data handling ability of FORTRAN leads to very severe restrictions. Several languages will be needed for various parts of the model.

7) Special models may be desirable for certain tunnels. A savings of one percent, a very reasonable expectation, on a tunnel costing tens or hundreds of millions of dollars will pay for a great deal of analysis. Far too little is done at the present time.

8) This model should be completed as soon as possible, both to check the approach and to
provide a means of analysis. Far too little is known about the factors affecting tunneling cost, and the economic evaluation of new tunneling techniques has been sketchy at best. It is deplorable that so much money is spent on tunnels without better means of analysis.

9) Better methods of specifying rock properties are urgently needed. The parameters used in this model do not adequately describe rock's response to tunneling. Models of this sort, although not this particular model, may be useful in this work.

10) The stochastic nature of geology can be incorporated into the model by defining small segments and generating random numbers for rock properties. Bledsoe (1970) first suggested this.

11) It should be noted that the framework for reach cost calculation is not as well developed as the framework for segment cost calculations. This portion of the framework may need revision before implementation can be accomplished. Nevertheless, it is felt that this framework will provide a useful guide in implementing a program. The logic itself seems to be valid, and the implementation of the tunnel definition language shows that programming a highly interactive system such as this is feasible.
References


17) National Research Council (1968); Panel Reports of the Committee on Rapid Excavation, September, 1968.


Additional References on Rock Mechanics and Tunneling


39) Hughes Tool Company (1964); Scientific and Technical Applications Forecast on Excavation. Prepared for
the U.S. Dept. of the Army, 1964.


PART I
COMPONENTS OF THE TUNNELING CYCLE

Hard rock tunneling consists of three basic operations: 1) breaking the rock from the tunnel face, 2) disposing of the broken rock and 3) supporting the opening thus made. All three operations are influenced by the geology and groundwater conditions at the tunnel site. Geometric constraints also influence the ease of construction and the particular methods used. There is considerable interaction among the operations; the excavation method used, for example, may influence the support required to maintain the opening. Thus, the tunneling process should be viewed as a system of smaller processes or components. These components include the three operations mentioned above plus several others which are equally essential, though less obvious.

Excavation

There are two excavation methods in general use today: drilling and blasting and the mechanical mole. Each method has many variations.

In a drill and blast cycle, holes are first drilled in the face to receive the explosive. The drills are then pulled back and the holes are loaded, tamped, primed, and fired. After allowing sufficient time for the ventilation system to remove the noxious gases generated by the explosive, often referred to as "smoke time", the men return
to the face, bar down loose rock, load the broken rock onto
the mucking system, and haul it away. Support is installed
to protect the opening, if necessary. The drills are then
moved back and the cycle is repeated.

Mechanical moles use cutters to break rock from the
face. They advance more or less continuously while the
broken rock is collected by buckets on the cutter head and
hauled away on a conveyor. Unfortunately, moles are not
economical in very hard rock.

Drill and blast tunneling is discussed by Mayo et al
(1968) and Richardson and Mayo (1941). Langevors and
Kihlstrom (1963) discuss blasting in detail. Hill (1968) and
Hirschfeld (1965) discuss mechanical moles.

Mucking

Muck removal involves loading the broken rock and
hauling it away. Muck loading is occasionally still done
by hand, usually in small tunnels. Hand mucking requires
that men lift the broken rock, either by hand or with
shovels, and load it onto the transportation system. This
method is slow and inefficient, as well as very hand physical
labor, and is only used when it cannot be avoided.* In
blasted tunnels, muck loading is usually done with a mucking
machine. There are several machine designs in use, but
they can be grouped into two basic types: 1) those which
scoop up the broken rock in a bucket and dump it (often onto

* Hand mucking is sometimes used in underdeveloped countries,
where equipment is very expensive compared to labor.
a conveyor built into the machine) and 2) those which use hydraulic (or air) powered arms to scrape the muck onto a conveyor. Tunneling machines scoop up muck as it is produced.

Muck transportation within the tunnel involves hauling the muck from the face to the shaft or portal which is the tunnel entrance. Rail transportation is the most widely used; locomotives powered by batteries, diesel engines, or compressed air haul strings of cars to and from the face. Rail haulage is quite safe, has large capacity, and the procedures are well developed. With conventional excavation, a train is often sized to hold all of the muck from one round, and the system is quite efficient. Switching problems and the necessity for laying track are the two biggest drawbacks.

Trucks are more flexible than trains since no track is required. If the entrance to the tunnel is a portal, the trucks can haul the muck directly to the disposal site, saving a handling operation. Trucks can often operate at higher speeds than trains, since the track in tunnels is usually in poor condition.

Trucks are usually diesel powered and tend to pollute the tunnel atmosphere. They are also dangerous to men working in the tunnel, since their path is not fixed, and illumination and noise levels in tunnels leave much to be desired. Trucks are not used in tunnels with diameters less than about 20 feet, since two trucks must be able to
pass in the tunnel for efficient operation.

Belt conveyors are sometimes used to transport muck, especially in mines. Such conveyors cannot handle large muck blocks, so secondary breakage is sometimes necessary. Unlike rail cars or trucks, conveyor belts are designed to move muck continuously. However, a breakdown anywhere along the conveyor will put the entire system out of operation, and these systems are still relatively unreliable.

Conveyors cannot be installed too close to a face where blasting is performed due to possible damage. Hence, haulage from the face to the conveyor is often required. Conveyors are much more suited for use with tunneling machines than with drill and blast tunneling, since moles and conveyors operate continuously, and the muck produced by the machine is usually composed of small pieces.

Muck hauling is occasionally done by hand or by draft animals. With this system, men or animals usually push muck cars, though muck is occasionally carried by hand. These methods were popular in the past, but are rare today.

If excavation proceeds from a portal, the muck transportation system may haul the muck directly to a stockpile or disposal area. When excavation proceeds from a shaft, however, the muck must be hoisted from the bottom of the shaft, where it is left by the muck transportation system, to the ground surface for disposal. A mine hoist is usually used for this purpose. This consists of a winch to raise
and lower a bucket, or "skip", into which the muck is loaded for hoisting.

Men and material must somehow be brought into the tunnel and to the working face. The mine hoist has a "cage" for men and materials, either connected to the skip or, on some large jobs, on a separate cable. Inside the tunnel, the muck transportation system hauls men and materials to the face, if rail or truck muck haulage is used. If conveyors are used, a separate system must be installed to transport men and materials to the face.

Support

Supports are placed in a tunnel to keep it open and to protect persons and equipment from rockfalls. Historically, temporary support was used to hold the tunnel open and to protect the workers during construction. Permanent support was later installed to perform the same function during the tunnels service life. Today, temporary support is incorporated into the permanent support system. The tunnel lining, considered in the next section, is often confused with support because their functions are quite similar.

Steel supports, called "sets", are steel I-beams or other sections bent to fit the contour of the tunnel. Wooden blocking is used to transfer the load from the irregular rock wall to the steel sets. Wooden or steel lagging is used to prevent rockfalls from between the sets.
This blocking and lagging is wedged tightly behind the sets, and is usually left in place when the lining is poured.

Steel supports have a definite thickness, so the tunnel must be excavated somewhat oversize to accommodate them. They are difficult to install and hard to handle, but they are strong and quite safe. They also have psychological advantages; men feel secure working under them.

Rock bolts have recently come into use for tunnel support. Holes are drilled radially from the tunnel, and bolts are installed and tightened, tying the rock mass together. The length of these bolts is usually five to ten feet.

Rock bolts must be installed as soon after excavation as possible, since loosening is a time dependent process. Drilling the holes and installing the rock bolts takes some time, but they are easier to handle than steel sets. Rock bolts do not protrude significantly inside the excavation line, so excess excavation is avoided.

Shotcrete, a pneumatically applied concrete, has been used in recent years for tunnel support. The concrete used has a high early strength, and the idea is to prevent loosening. A thin (1-4 inches) coat of shotcrete should be sprayed on immediately after excavation for the best results. A thin coat of shotcrete acts as a shell, and assists the rock in supporting itself.

Shotcrete has the advantage of coating the entire inner
surface of the tunnel, thus preventing rockfalls. Like rock bolts, shotcrete leaves the interior of the tunnel relatively uncluttered. It also is light colored and aids in tunnel illumination. Shotcrete is often used in conjunction with rock bolts for tunnel support. It has been widely used in Europe, but has only recently been introduced in the United States.

In any given tunnel, it is likely that a variety of rock conditions will be encountered. It is very common for one tunnel to have support consisting of steel sets, rock bolts, and shotcrete. A combination of these methods is often used in the same section of a tunnel, steel sets covered with shotcrete, for example.

The design of a support system involves the estimation of rock loads and the structural design of supports to carry these loads. The structural design is quite straightforward, but the estimation of loads is still an art. Proctor and White (1946) describe steel support systems and give empirical rules for rock loads. Deere et al. (1969) give more recent rules and include rock bolts and shotcrete. Szechy (1966) describes support systems and solutions based upon elasticity. Rabcewicz (1964-1965) discusses shotcrete.

**Lining**

A lining is placed in a tunnel to make it more suitable for its final use or to prevent deterioration of the supports, rather than to stabilize the opening. In pressure tunnels,
the lining resists the internal pressure. The main difference between lining and support seems to be that support is installed immediately after excavation, while lining is installed at some later time.

Linings for hardrock tunnels are usually cast-in-place concrete, although shotcrete is sometimes used. They are sometimes reinforced, but usually they are not. Forms are erected, and the concrete is poured behind them. Grout is then forced behind the lining, to insure good contact with the rock. Movable, collapsible forms are often used to make placement of the lining a more or less continuous operation. It is common to install linings after excavation is completed, to avoid interfering with the mucking operations.

There are two philosophies of lining design. One states that the lining should be capable of stabilizing the tunnel by itself, neglecting the support or the rock's ability to carry load. This type of design is very conservative and was developed when rock mechanics was in its infancy. It is still used in cases where the consequences of failure would be very serious.

The other philosophy of lining design states that the supports will carry all of the load, and any lining beyond what is necessary to protect the supports simply increases the factor of safety (which is at least 1.0 without the lining). This approach is relatively new, and is discussed
Groundwater Control

Large inflows of groundwater into a tunnel hamper construction severely, and may even make further progress impossible. Several methods have been developed to control groundwater. They involve either removing the water or preventing flow.

Grouting, the injection of cement, mortar, or chemicals (usually epoxy resins) into the rock under high pressure, is frequently used to control groundwater inflows. For pre-grouting, holes are drilled ahead of the face and grout is injected. The grout fills the crevices in the rock and hardens, greatly reducing the water inflow when the face is advanced. If flow from the face is small, but the combined leakage along the length of the tunnel is significant, grouting is performed in holes drilled radially outward around the tunnel perimeter.

Grouting usually cannot stop all water inflow, so pumps are required to remove the water from the tunnel. The water is collected in sumps and pumped either to the surface or through the portal. If inflows are small, grouting may not be required, and the flow may be controlled only through the use of pumps. If the tunnel has some slope, it is desirable to drive the tunnel up slope so that water will drain away from the face and out of the tunnel. Pumping water out of a deep shaft can be difficult, and a
series of booster pumps may be required.

Dewatering is sometimes carried out ahead of the face to reduce the groundwater inflow. If the tunnel is close to the surface, wells can be used to lower the water table so that groundwater inflows are not a problem. In deeper tunnels, a small drainage adit is sometimes driven to drain the rock mass ahead of the main excavation. This technique is especially useful where the overall permeability of the rock mass is low, but there is a large volume of stored water. Discussions of groundwater control can be found in Richardson & Mayo (1941), Mayo et al. (1968) and Szechy (1966).

Ventilation

Ventilation systems are designed to meet the oxygen demand, to keep concentrations of dust and gases at safe levels, to control temperature and humidity, and to keep the air flow turbulent so that stratification of gases does not occur. The laws of fluid mechanics for incompressible flow are used to determine air flows and power requirements. The laws of thermodynamics and heat transfer are used to determine cooling requirements.

Ventilation systems can be blower types, exhaust types, or combinations. Blower types deliver fresh air directly to the tunnel face, from which it flows back along the tunnel to the shaft or portal. This carries the dust and gases back through the tunnel. Exhaust systems suck air from the face
and exhaust it at the surface. The fresh air comes through the tunnel and collects gases and dust before it reaches the face. Blower systems are usually preferable because they deliver fresh air to the face, where most of the men are.

In tunnels where drilling and blasting is performed, there is a high concentration of fumes and dust at the face right after the blast. Ventilation systems are often designed to be reversible, so that these fumes and dust can be exhausted quickly, without contaminating the rest of the tunnel, but the benefits of a blowing system can be realized the rest of the time. Tunneling machines are often equipped with an independent dust removal system.

The design of ventilation systems is fairly straightforward compared with the other phases of tunneling. Hartman (1961) gives a good treatment of ventilation problems. United Aircraft Research Laboratories (1968) also discuss ventilation in some detail.

Surface Plant

The surface plant consists of office facilities, warehouses, concrete plants, stockpiles, construction camps, shops, laboratories, and all other facilities erected at the site for the support of the tunneling operation, but not directly a part of it. Some of these facilities may be constructed underground, especially on large jobs, but they are still usually referred to as surface plant. The extent of these facilities depends upon the size of the job.
and the contractor's preferences. It is important to arrange the surface plant so that it is used efficiently and does not delay the tunneling operation. There are no standardized procedures for surface plant operation, however.

**Safety**

Tunnels are probably the most hazardous of all construction jobs. Many of these hazards can be traced to unforeseen geologic conditions, but some are due to the crowded working space, the lack of fresh air, and the explosives used. Legislation has been enacted to deal with the latter causes, but geologic dangers are still largely unsolved problems.

Rockfalls, rockbursts, and water inflows can be attributed mainly to geologic conditions. These dangers can be predicted to a certain extent, but there is always an element of guesswork involved. Rock mechanics is still in its infancy, and cheap, accurate methods of predicting geologic details have not been developed.

Rockfalls occur in rock with unfavorable joint patterns or in rock which has been heavily damaged by blasting. They can be controlled with adequate support, but support requirements are difficult to estimate.

Rockbursts are likely to occur in areas of high tectonic stress. In a rockburst, a chunk of rock is thrown from the side of the tunnel with explosive force. Light support can prevent rockbursts, but it is difficult to
predict areas where they will occur.

Groundwater inflows can sometimes be predicted, but large, unexpected inflows can be disastrous. A pilot hole is often drilled in advance of the face to warn of large inflows.

Improper use of explosives can be a serious hazard in tunneling. The dust and fumes produced during blasting place a heavy demand on ventilation systems. Misfires may result in explosives being mixed with the muck of left in drill holes. The mucking equipment or a drill may detonate this explosive, causing a disaster. Thunderstorms and two-way radios have been known to detonate explosives unexpectedly. Many precautions must be taken when explosives are used, but hazards will still exist.
There are few things which do not affect the cost of tunnel construction. The geologic and groundwater conditions at the site are by far the most important, but geometry, the local and national economic outlook, and many other factors also influence cost. The details of the construction methods used have an influence which cannot be neglected.

Tunnel Cost Components

Tunnel costs are usually expressed in dollars per linear foot of tunnel. This is done for convenience only, not because the cost per foot is constant. The costs are often further broken down into components analogous to the construction components discussed in Part I. Thus, the excavation cost component is the cost of performing the excavation only. The advantage of this type of breakdown is that it shows where the largest costs are, and which procedures need the most improvement. Its disadvantages is that it tends to mask the interrelationships between the cost components.

No cost component can be considered to contribute to the cost of the tunnel completely separately from all other cost components. Sometimes the interrelations are minor and can be ignored, such as the interaction between ventilation and muck hoisting, while at other times, the
interdependence is very pronounced, such as the dependence of support upon excavation method. The components must be considered together if a reasonable cost is to be derived. Costs are usually computed separately for each component, taking the interactions into account.

The excavation component includes all of the labor, equipment, and materials required to remove the rock from the face and to break it into pieces small enough to be loaded on a muck car. When a tunneling machine is used, the muck loading is also included in excavation. The labor subcomponent includes only those men working directly upon excavation, even though they may also do other jobs. The labor component is computed by adding up the products of each man's wage and the time he works. The time worked, or, alternately, the number of men, can be determined from a knowledge of the excavation process (i.e., cycle time, number of holes,) and the productivity of the men.

The equipment subcomponent includes all of the equipment used for excavation, such as a tunneling machine, drills, the jumbo, etc. The decision whether or not to include the auxiliary components of this equipment, such as the transformers and cables for the machine power supply or the compressors and hoses for the drills, is somewhat arbitrary, (since they may also be included under surface plant). Equipment costs are calculated by adding purchase and maintenance costs and dividing by service life. This gives a cost
analagous to a wage, which can be used in essentially the same way for cost calculations.

The material subcomponent includes all expendable items used for excavation, such as explosives, bits, tunneling machine cutters, power, etc. The distinction between material and equipment is somewhat hazy in some areas, such as drill steels or hand tools, and the category in which these are placed is somewhat arbitrary.

Material costs are computed by determining the material cost and life, and converting to some unit cost (such as dollars/foot).

The mucking component consists of all the costs associated with removing the broken rock from the tunnel, including the loading, hauling, hoisting, and disposal of the muck. The labor associated with mucking includes the mucking machine operator, the operators of the muck haulage system, the track layers, the hoist operator, and the drivers who haul the muck above ground.

The equipment required for mucking includes mucking machines, locomotives and muck cars, (or trucks, or conveyors), the hoist, (if any), the trucks or trains used above ground, and the disposal site (if any).

The materials used in mucking include track, oil and grease, fuel, and power. The muck itself is often a valuable product and can be used for aggregate or fill. This will constitute a material credit. The muck is often used
on another portion of the job of which the tunnel is a part, and this must be considered.

Support consists of all those operations connected with the stabilization of the tunnel opening. These include installing steel sets, rock bolting, applying shotcrete, installing blocking and lagging, etc. The labor utilized in support installation includes the erectors, drillers for the rock bolt holes, the shotcrete machine operator, and any labor required to bring the material into the tunnel.

There is not very much equipment associated with support installation. The shotcrete machine and the drills for rock bolting, (which may also be used for excavation) are about the only things not classified as hand tools. Tunneling machines often have erector arms attached, (to help with support installation) but these are considered part of the machine.

The materials used for support include the steel supports, the rock bolts, the shotcrete, the nuts, bolts, wooden wedges and lagging, and any bits or power used for installation.

The lining component of tunnel cost consists of all those costs associated with the installation of the permanent lining. The labor subcomponent includes the men who erect the lining forms, those who place the concrete, those who strip the forms, and all the people involved in
the transportation of the concrete.

The equipment used consists of the concrete pump, the lining forms, vibrators, concrete cars, etc. A concrete batch plant is often constructed on the surface to provide concrete for the tunnel lining. This is usually considered a part of the surface plant for cost purposes, or else a unit cost for concrete is computed and entered under the material subcomponent.

The material used in a lining is basically the concrete itself. Any grout used behind the lining is also a material cost. (This grouting has labor and equipment subcomponents also).

If a shotcrete lining is used, the cost components will be somewhat different. All of the costs associated with the formwork will be eliminated, and the labor, equipment, and materials will change.

Groundwater control includes grouting and pumping. The labor required includes the pump operators, the operators who drill the grout holes, and the persons who place the grout.

Equipment used in groundwater control includes the grouting machine and the drills to drill the holes. Materials used are the grout itself (chemical or concrete), and the bits for hole drilling, as well as fittings for the grout holes.

Ventilation requires fans and pipes to deliver fresh
air to the face. The labor involved includes the operators and the persons involved in erecting the pipe. The equipment is the fans. The pipe is usually considered material, as is the power required for operation. Any air conditioning devices required are also equipment.

Setup includes all those operations which must be performed before construction can begin. As work progresses, some of these facilities may be moved underground, especially the transformers and shops. The costs associated with this move should be included in setup. The setup has labor and material subcomponents. The equipment subcomponent can be included either under the appropriate operation or under surface plant operation. The equipment used solely for setup must be charged to setup, however.

The surface plant operation component consists of the costs involved in keeping the plant in operation. Operators and clerks, as well as many of the supervisors, make up the labor subcomponent. The equipment is charged as mentioned above. Materials used in this operation include not only such obvious things as power and fuel, but also things like paper, stamps, pencils, water, sewage, etc. Care must be taken to include all of these items which contribute to the cost.

Overhead is the contractor's cost of doing business. It includes such things as home office expenses, advertising, insurance, interest, etc. Overhead is usually computed as a
percentage of the total job cost. Roughly 20% is an average figure. Profit is also a percentage of total cost, varying from 0% to 10% or more, depending upon the risks taken by the contractor and various other factors having little or no connection with the tunneling operation.

Certain costs connected with tunneling do not fit into any of these categories. The costs of exploration and design are not considered at all, since they are the responsibility of the engineer rather than the contractor, and are, in theory, completed before construction is begun.

These cost components are useful for comparison purposes to see which areas of construction should be improved to reduce costs. The costs which enter these components can be quite different on different jobs, depending upon the personal preferences of the person calculating the costs. For this reason, components on different jobs must be compared with great care, if at all. The same costs must go into the components if valid comparisons are to be made.

Influence of Geology and Groundwater

Rock type and hardness influence the amount of explosive used and have a great effect upon bit or cutter penetration rate and upon bit or cutter wear. Bits and especially tunneling machine cutters are very expensive, and excessive wear can make costs prohibitive.

Rock type can influence the safety hazards present
in the tunnel, such as gas. The control of these hazards can be quite expensive. Rock type will also affect the value of the muck produced.

The R.Q.D. of the rock (actually the geometry of the imperfections) has an overriding effect upon the number and type of supports, directly affecting costs. The R.Q.D. also affects the amount of overbreak and, to some extent, the ease of excavation, thus affecting excavation and mucking cost. The amount of support required determines how much larger than the nominal diameter the excavated diameter must be, thus affecting excavation and mucking cost. Support requirements also dictate how soon after excavation the supports must be installed, thus determining how much interference occurs between support, excavation, and mucking. This interference can slow down the operation and increase costs.

Groundwater makes all operations in the tunnel more messy and less efficient. Large inflows can stop work altogether. The grouting and pumping which must be done are costs directly chargeable to groundwater control. In addition, the grouting and pumping operations interfere with other work and significantly reduce the advance rate, thus increasing costs. Underpinning or lawsuits necessitated by subsidence at the ground surface can often be attributed to groundwater. If groundwater inflows are very large, special excavation methods, such as hand mining or the
driving of drainage adits may be necessary, at a very great cost.

The Effect of Geometry

The size of the tunnel has a very important influence on cost. In general, as the tunnel diameter increases, the cost per cubic foot of space created decreases but, naturally, the cost per linear foot increases. More efficient methods of excavation and mucking can be used in large tunnels, so a decreased unit cost is to be expected. Tunneling machines, however, become extremely expensive for large diameter tunnels because of the large thrust requirements. In general, the heading and bench method of excavation is more efficient than full face operations in large tunnels, and trucks are the preferred muck transportation system, unless the tunnel length is very great.

Shape has some influence on cost, due mainly to its effect upon the excavation procedures and the support requirements. Shape, taken together with size, can influence the amount of congestion in the tunnel and affect productivity. Unusual shapes may be difficult to construct, and may slow down the tunneling operation. It is difficult to construct a transportation system in tunnels with circular inverts, and additional costs for ballasting will be incurred. Today's tunneling machines cannot excavate non-circular shapes, and in some cases a requirement calling for a non-circular shape may eliminate the potentially cheapest construction method.
A long tunnel gives more length and time over which to amortize initial expenditures for equipment and reduces equipment unit costs (assuming that the equipment is written off on one job and that equipment lasts for the whole job). On the other hand, a long tunnel will often require higher capacity equipment. The tradeoffs vary with the equipment used.

The longer the tunnel, the more time is lost due to transportation from the entrance to the face. This increases all labor costs, since men are usually paid from the time they enter the tunnel until the time they leave. It also increases muck and material transportation costs and power and air transmission costs.

Long tunnels usually have several shafts constructed along their length so that several faces can be worked simultaneously. There is a tradeoff associated with the cost of shaft sinking and the benefits of decreasing tunnel length and shortening construction time.

Long tunnels are necessary to justify the expense of a tunneling machine, since the machine cost is usually amortized against one job. Thus, the length can affect the excavation method chosen. Rail transportation of muck becomes more and more favorable compared to other methods as tunnel length increases, so length may affect the mucking method chosen. More accurate cost estimates can also be made for long tunnels, since conditions will
more nearly approximate the average conditions assumed in the estimate.

The magnitude of the slope can affect the economy of the muck hauling method chosen. Rail cars cannot be operated economically on slopes greater than about 10%, so other, perhaps less desirable, forms of transportation must be used.

The depth below the surface of the tunnel adds a cost to every component, unless the method of entry is through a portal. Every man and every piece of equipment or material must be hoisted up and down. The deeper the tunnel is, the longer this takes and the larger the capacity and the higher the cost of the mine hoist must be. In addition, the groundwater encountered must be pumped to the surface for disposal. This is very expensive in deep tunnels.

The high temperatures encountered at great depth, coupled with the high humidity usually characterizing a tunnel environment, make necessary the installation and operation of a very expensive air conditioning plant, unless an extremely low productivity and horrible working conditions can be tolerated.

The method of entry into a tunnel has a great influence upon cost. If a portal is used, the costs incurred in excavating the shaft, equipping it with a hoist, and the hoisting process itself are eliminated, as is the necessity
of pumping against large heads. However, the portal must be constructed, and the cost of this can be significant. Portal entry also eliminates the costs associated with transferring men, equipment and materials between the surface transportation system, the hoist, and the underground transportation system.

There are essentially no geometric standards for tunnels. This lack of standards is especially unfortunate for moled tunnels. The tunneling machine must be designed and built for each particular job, and usually cannot be used on other jobs, even if it is not worn out. This leads to long delivery times and high capital costs. The lack of geometric standards is one important reason for the lack of standardized tunneling procedures.

The Effect of Excavation Method

Excavation is the first operation performed in tunnel construction (after setup), and has an influence on just about every other operation. The success of the excavation process in breaking up the rock will in large part determine the efficiency of the muck loading, hauling, and hoisting. It will also partially determine the value of the muck.

The damage done to the surrounding rock by excavation determines to some extent the amount of scaling necessary, the overbreak, and the support requirements. The overbreak is an important factor in the cost of a concrete lining. Tunneling machines do much less damage to the rock
than drilling and blasting.

The excavation method also determines to a large degree the ventilation requirements for the tunnel. Explosives produce gases which must be removed, while tunneling machines and drills produce large amounts of dust. The larger the quantity of fresh air required, the higher the cost of the ventilation system.

The groundwater control system which can be used is influenced by the excavation method. Pregrouting is difficult when using tunneling machines. Making exploratory borings ahead of the face to detect groundwater is also difficult in machined tunnels. These factors add to the difficulty and hence to the cost of groundwater control.

**The Effect of the Mucking Method**

The muck loading and hauling methods chosen will determine the degree of fragmentation to be achieved by the excavation process. The capacity of the mucking machine and haulage system will influence the advance rate and hence the cost.

The type of power used for the muck haulage system may play a large role in determining the capacity and, hence, the cost of the ventilation system. The haulage system may also influence safety requirements and costs.

The muck hoisting system is often a bottleneck in the muck removal operation. The capacity of the rest of the mucking system, coupled with the requirements for hoisting
men, equipment, and other materials, will determine the capacity and cost of the hoisting system.

The Effect of the Support and Lining Systems

Support and lining are very closely linked. The type and amount of support used will have a major influence on the cost of the lining. Conversely, the type of lining decided upon will influence the support design and cost.

The installation of supports can cause interference with other tunneling operations. Thus, the details of the support installation process and the type and number of supports will affect the efficiency of the other operations underground. The allowable time lag between excavation and support can have a large effect upon the cost of mucking, as well as the care with which excavation must be carried out.

The Effect of Groundwater Control and Ventilation

The groundwater control system can interfere with other tunneling operations, slow down progress, and raise costs. It is very difficult to perform grouting operations while other processes are going on, so much time is lost due to this operation.

Chemical grout is more expensive than concrete grout, but it can penetrate smaller cracks. Grout can stabilize the rock mass somewhat, but if too great pressures are used, loosening can occur. Grouting can affect support either favorably or unfavorably.
Ventilation affects the efficiency of the men working in the tunnel. The greater the supply of free air, the more efficient the work is. Thus, there is some kind of tradeoff between the cost of ventilation and increased cost due to inefficiency.

The surface plant must operate smoothly for the underground operations to be efficient. The equipment at the surface, however, is dependent upon the underground operations.

**Regional Consideration**

The part of the country (or world) where the tunnel is built can have a large effect upon the cost. Prices vary widely in different areas, so material costs can be significantly different.

The degree of unionization of the work force and the attitude of the union leaders can have a very strong influence on the cost of labor. Where unions are strong and non-cooperative, the price of labor may be very high and the productivity low.

In regions where unions are weak, the cost of labor may be low, but the productivity may well be low also. This is due to the lack of skilled workers. Tunnel construction calls for considerable skill, and, in many locations, workers possessing this skill are not present. Workers must then be either imported or trained, both of which are expensive. In remote areas, the cost of operating
a construction camp must not be neglected when computing a cost.

**Time Considerations**

As a general rule, costs increase with time, so it is usually desirable to start a job as soon as possible and to finish it in the shortest possible time. Completing a job in a short time not only saves on cost increases; it also reduces interest payments. In addition, a premium is often paid for early completion and a penalty imposed for late completion. This reflects the saving to the owner on interest payments and his increased revenue from having the facility completed sooner, as well as any social benefits of early completion.

Unlike most construction jobs, a tunnel, once the initial setup has been performed, is a closed environment, and independent of the weather. The supply of materials may be adversely affected by the weather, however, especially in remote areas with unfavorable climates.

Further discussions of tunneling economics are given by Hill (1968), the California Department of Water Resources (1959), the Harza Engineering Company (1968, 1970) and Norman and Stier (1967).
APPENDIX B

TUNNEL DEFINITION LANGUAGE
Introduction

The MULTICS system is a fairly powerful, user oriented time sharing system. It was developed at MIT by Project MAC, using a G.E. 645 computer. Several languages are available on MULTICS; PL/1 was used for this program because of its many capabilities for character string manipulation, data structure manipulation, and compatibility with the more advanced features of the MULTICS file system. The version of PL/1 available on MULTICS is described by the Cambridge Information Systems Laboratory (1968).

Care must be taken not to confuse the term segment as used by MULTICS with the term segment used in reference to a portion of a tunnel. A segment in MULTICS is a group of memory locations, much like a FORTRAN array. Detailed information on the MULTICS system is given in the MULTICS Programmers Manual (MIT, 1971). Additional information is available through Project MAC.

The Program

The basic function of Tunnel Definition Language is data manipulation. It provides a convenient way for data input and data modification. It was written to check the feasibility of allowing the user to set up his own model of
tunneling, and to see how convenient this would be for him. The processor appears to be successful in this, although it is complicated. Fortunately, the MULTICS operating system contains subprograms to perform many of the required functions.

Tunnel Definition Language itself consists of six programs, a command interpreter which is invisible to the user, and three tables, symlab, methods, and name.DESCR. The description of its operation is keyed to the example given in the next section. A listing follows the example.

The processor operates at command level; the programs at request level. Command level supervises the entire operation and calls the appropriate programs. Request level provides input to a specific program. Any program can be called by the user, using a command level command, or by another program as a subroutine. For example, tdl can be called directly by the user by typing "tdl". It is also called automatically by initiate-tunnel, after the number of reaches, shafts, and segments have been specified and names have been assigned. Each program accepts several requests. All programs have a request called "quit", which exists from the program and returns control to command level.
Users Manual

Refer to the example following as each of the programs is described. Arrowheads indicate that the user typed the text.

Initiate-Symtab and Modify-Symtab

These two programs set up the symbol table. In the example, initiate-symtab was typed by the user. The program returned the empty message and asked for the number of symbols. The user supplied the numbers. The program then called modify-symtab. Modify-symtab typed out MODIFY-SYMTAB to indicate this. If the symbol table already existed, initiate symtab would simply supply the system with a pointer to it and return to command level.

The user typed "help" so that the program would print out a message telling the form of the input. The program typed the next two lines. Requests are add (or a), list (or l), change (or c) and delete (d). Type tells whether a variable is used for segments and shafts (s), reaches (r) or tunnels (t). Name is the name of the variable (1-4 characters). The indicies are the locations in the two symbol tables. These numbers will be used by the program to refer to the data. Lname is the longer, more descriptive variable
name (up to 16 characters). In the example, the user added four variable names to the segment and shaft table and two to the reach and tunnel table (one for reaches, one for tunnels) the "#" in the last line erases the previous letter. (The user made a mistake.)

The user then typed list with no argument. This caused the program to tell how many variables in each table had been created. The word segment (or s) or reach (or r) following list (or l) causes the program to print the contents of the table. If the list request is followed by a variable name (see example), the information for that variable name is listed. New names can be added with the add request. The @ on the next line "erased" the line because a mistake was made. It can be seen that the variable "extr" was added to the table. The change (c) request was used to modify the subscript and lname of the variable extr. Finally, extr was deleted. The request quit returned to command level for more commands.

Initiate-methods and Modify-Methods

These two programs identify the construction methods to be considered and the criteria for their elimination. The methods are stored in a segment called methods. The user
typed initiate-methods. The program returned an empty methods message and called modify-methods. If methods existed, the program would return a pointer to it and return to command level.

The user typed help so that the program would tell him the form of the input. The program printed the next two lines. Requests are add (a) list (l), and delete (d). Phases are excavation (e), mucking (m), lining (l), support (s), and grouting and pumping (g). The identifier is the variable name used for the method (1-4 characters). The description is the long form of the name (1-16 characters). Ntests is the number of criteria with which the method is to be tested.

After each method, the program asks for the feasibility test. Item is the item number of the variable in the symbol table which is to be tested. The relation can be greater than (>), less than (<), equal (=), or not equal (≠). The quantity is the magnitude which is checked. For example, the user input no test for drilling and blasting. The test input for mole excavation is "delete mole excavation if quantity 2 (rock strength) is greater than 30,000". After all methods and tests had been input, the user listed them. He also listed the methods for the excavation and support
phases and the tests for steel-sets. The user then tried to delete stel, but he forgot to specify the phase and got an error message. He then successfully deleted stel, listed the remaining methods, and returned to common level.

_Initiate-Tunnel and tdl_

These two programs define the tunnel and create the segment name.descr. The user first typed initiate-tunnel followed by the tunnel name. The system replied that this is a new tunnel asked for the number of shafts, reaches, and segments. It then called tdl. If the tunnel already existed, initiate-tunnel would return a pointer to it and return to command level. The user could enter tdl himself by typing tdl.

Tdl asks for the reach name, the entry shaft, the shaft in the direction of excavation, and the names of the segments in the reach. The period (.) indicates that input is finished. Note that more space was reserved by initiate-tunnel than was actually used. _|list_ lists the reaches and their components.

A greater than sign (> ) sets a pointer to the next segment. Data for it can then be input. When all segments have been defined, the pointer moves to shafts, then reaches,
then the tunnel. In the example, se7 was the first segment, so it was pointed to first. The request prompt causes the program to type out the long names of the symtab variables (p uses the short names). The user can simply type in numbers if he desires. In any event, the user must input the numbers after the variable name. These numbers are stored in name.descr. The user then listed the data for this segment. List uses long names, l uses short names.

The user then specified methods, using the request m:. He wanted support methods (sup) and wished to include all of the methods in the segment "methods". The methods specified are stored in fsbl, a part of name.dscr. He then listed the methods (lm). The program asked for the phase. After specifying grouting methods (none), he typed a > and moved to the next segment.

An "=" tells the program to set the variables for this segment (se8) equal to the values for the previous segment (se7). The user then changed the diameter to 25.0 and listed the values. He specified methods and moved to the next segment, where he tried to change the length. He used the wrong variable name, and got an error message. He then changed the diameter, specified methods, and moved to the next segment.
In segment se2, the user set all values equal to the values used in segment sel. He then typed p which causes the variable names to be printed out. A comma (,) typed after the variable causes the value to be printed. A number sets the variable to that value, and an = segment name sets the variable equal to the value for that segment. The user typed diam , , which caused the value of diam to be printed. He then specified methods and moved to the next segment, where he set the values of the variables equal to the values in segment sel (=sel). The user continued through all of the segments until he got to a shaft (porl). He then specified data for all shafts, reaches, and the tunnel (making mistakes which he corrected with #'s) and finally returned to command level by typing |quit. After all of these operations, the tunnel data was stored and ready for use by the next processor.

Tunnel

If symtab and methods already exist, the user can use the command tunnel name. Refer to Figure 11, Chapter 3 for this example. The system tells the user that it found symtab and pdb (permanent data base; i.e. methods), and that this is a new tunnel. The user specified the reaches, shafts, and segments. The program then called tdl and the user
input information for one segment and one reach. This is the way Tunnel Definition Language would normally be used.
EXAMPLE
*initiate_symtab
  empty symbol table
  how many shaft and segment symbols? 8
  how many reach and tunnel symbols? 4

MODIFY_SYMTAB

*help
  standard input line
  request type name seg or rch idx sft or tun idx lname
  *add segment rqd 1 1 rock_qual_dsgn
  *a s strn 2 2 comp_strength
  *a s diam 3 3 diameter
  *a s lgth 4 4 length
  *a reach otom 1 0 outdoor_temp
  *a tunnel 0#dnif 0 1 design_life

*list
  segments and shafts: 8 dimensioned 4 used
  reaches and tunnel: 4 dimensioned 2 used

*list segment
  name seg idx sft idx long_name
  rqd 1 1 rock_qual_dsgn
  strn 2 2 comp_strength
  diam 3 3 diameter
  lgth 4 4 length

*list reach
  name rch idx tun idx long_name
  otom 1 0 outdoor_temp
  dnif 0 1 design_life
<table>
<thead>
<tr>
<th>name</th>
<th>seg idx</th>
<th>sft idx</th>
<th>long name</th>
</tr>
</thead>
<tbody>
<tr>
<td>rqd</td>
<td>1</td>
<td>1</td>
<td>rock_qual_dsgn</td>
</tr>
<tr>
<td>strn</td>
<td>2</td>
<td>2</td>
<td>comp_strength</td>
</tr>
<tr>
<td>diam</td>
<td>3</td>
<td>3</td>
<td>diameter</td>
</tr>
<tr>
<td>lgth</td>
<td>4</td>
<td>4</td>
<td>length</td>
</tr>
<tr>
<td>extr</td>
<td>5</td>
<td>0</td>
<td>extra_variable</td>
</tr>
</tbody>
</table>

- l s seg
  - name   seg idx | sft idx | long name    |
  - rqd    1       | 1       | rock_qual_dsgn |
  - strn   2       | 2       | comp_strength |
  - diam   3       | 3       | diameter     |
  - lgth   4       | 4       | length       |

- c s extr 8 8  extra_mod
- l s extr extr 8 8 extra_mod
- delete s extr
- l s name   seg idx | sft idx | long name    |
  - rqd    1       | 1       | rock_qual_dsgn |
  - strn   2       | 2       | comp_strength |
  - diam   3       | 3       | diameter     |
- quit
  - r 1655 7.661 316+232
\texttt{initiate_methods}
\texttt{empty methods segment}
\texttt{MODIFY METHODS}

\texttt{help}
\texttt{standard input line}
\texttt{request phase identifier description ntests}

\texttt{add item exc relation quantity}
\texttt{drill\_and\_blast 0}

\texttt{add item exc relation quantity}
\texttt{mechanical\_mole 1}

\texttt{add item sup relation quantity}
\texttt{rockbolts 1}

\texttt{add item s relation quantity}
\texttt{shotcrete 1}

\texttt{add item s relation quantity}
\texttt{steel\_sets 1}

\texttt{list}
\texttt{phase number of methods}
\texttt{excavation 2}
\texttt{mucking 0}
\texttt{lining 0}
\texttt{support 3}
\texttt{grouting and pumping 0}
list excavation
code number of tests
drbl 0 drill_and_blast
mole 1 mechanical_mole

1 s
code number of tests description
rblt 1 rockbolts
shot 1 shotcrete
stel 1 steel_sets

1 s stel
1 tests for steel_sets
item relation quantity
1 > 00.00000000

delete stel
identifier not found, request ignored

delete sup stel

1 s
code number of tests description
rblt 1 rockbolts
shot 1 shotcrete

quit
r 1854 0.607 230+201
-> initiate_tunnel testA
new tunnel
how many reaches? 3
how many shafts? 5
how many segments? 15
TUL
specify components:
reach shaft1 shaft2 segments
> C por1 sh1 se7 se8
> A sh1 por1 se1 se2 se3
> B sh1 por2 se4 se5 se6
> D por2 sh1 se9 se10 se11 se12

modify
list

<table>
<thead>
<tr>
<th>reach</th>
<th>shaft1</th>
<th>shaft2</th>
<th>segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>por1</td>
<td>sh1</td>
<td>se7</td>
</tr>
<tr>
<td>A</td>
<td>sh1</td>
<td>por1</td>
<td>se1</td>
</tr>
<tr>
<td>B</td>
<td>sh1</td>
<td>por2</td>
<td>se4</td>
</tr>
<tr>
<td>D</td>
<td>por2</td>
<td>sh1</td>
<td>se9</td>
</tr>
</tbody>
</table>

->
> promt
rock_qual_dsgn 80.0
comp_strength 2500.0
diameter 20.0
length 6000.0

-> list
list
rock_qual_dsgn 80.00000000
comp_strength 25000.0000
length 6000.000000

m: sup all

1m
phase?sup
rblt shot

m: grouting none

1m g
none

> 
>seg se8
= 
diam 25.0

1
rqd 80.00000000
strn 25000.0000

m: s all
m: g none

>
>seg se1
=
length 8000.0

token length not found

lgth 8000.0
diam 20.0
m: s all
m: g none
seg se2
p
rqd
80.0000000
strn 20000.0
diam =se6
lgth =se8
3000.00000

\[ \text{diam} = 25.0000000 \]
m: s all
m: g none
seg se3
=se1
diam 20.
ln#gth =se7

\[ \text{seg se4} \]
=\
\[ \text{seg se5} \]
\[ \text{seg se6} \]
> s se5

> seg se6

> seg se9

> seg se10

> seg se11

> seg se12

> sft por1

> rqd 80.0

> strn 25000.0

> diam 20.0

> lgth 0.0

> sft sh1
strn 20000.0
m: e all
lm
phase?e
dbl1 hole
m: e dbl1
lm
phase?exc
dbl1
m: m all
m: l all
m: s all
m: g all
>
sft por2
=por1
1
rgd 80.000000
strn 25000.0000
diam 20.000000
lght 0.
>
rch 0
prompt
outdoor_temp 70.0
m: e db1
m: m all
m: l none
> rch A
> p
tem, 70.000000

tem = 00.0
list
outdoor_temp = 00.000000

> rch B
> tem = A
> tem, 00.000000

> rch D
> tem = 70.0

> tun
prompt
design_life = 20.0

| quit
r 1954 0.660 416+273
LISTING
initiate_symtab: procedure;
declare sym_ptr ptr external static,
  1 symtab based (sym_ptr),
  2 s_dim fixed bin,
  2 s_used fixed bin,
  2 s(s_dim),
  3 sym_name char(4), /*short form name*/
  3 symbolic_name char(16), /*long form name*/
  3 idx(2) fixed bin, /*1=segment, 2=shaft*/
  2 r_dim fixed bin,
  2 r_used fixed bin,
  2 r(r_dim),
  3 sym_name char(4),
  3 symbolic_name char(16),
  3 idx(2) fixed bin; /*1=reach, 2=tunnel*/

 declare hcs_make_seg external entry(char(*),char(*),char(*),
  fixed bin(5),ptr, fixed bin(17)),
con_err_ external entry,
wdir_name char (168) aligned,
get_wdir_ external entry returns(char(168) aligned),
(error_table_$segknown,error_table_$namedup) external fixed bin(17),
ioa_ external entry,
code fixed bin(17),
ask$ask_int external entry;
wdir_name=get_wdir_;
call hcs_make_seg(wdir_name,"symtab","symtab",11011b,sym_ptr,code);
if code=0
  then do;
call ioa_("empty symbol table");
call ask$ask_int("how many shaft and segment symbols? ",s_dim);
call ask$ask_int("how many reach and tunnel symbols? ",r_dim);
s_used=0;
  r_used=0;
call modify_symtab;
end;
else if (code=error_table_$namedup | code=error_table_$segknown)
    then call ioa_("symtab found");
else call com_err_(code,"initiate_symtab");
end initiate_symtab;
modify_symtab: procedure;
declare. sym_ptr ptr external static,
  1 symtab based (sym_ptr),
    2 s_dim fixed bin,
    2 s_used fixed bin,
    2 s(s_dim),
      3 sym_name char(4), /*short form name*/
      3 symbolic_name char(16), /*long form name*/
      3 idx(2) fixed bin, /*1=segment, 2=shaft*/
  2 r_dim fixed bin,
  2 r_used fixed bin,
  2 r(r_dim),
    3 sym_name char(4),
    3 symbolic_name char(16),
    3 idx(2) fixed bin, /*1=reach, 2=tunnel*/
  1 structure based(P),
    2 dim fixed bin,
    2 used fixed bin,
    2 table(dim),
      3 sym_name char(4),
      3 symbolic_name char(16),
      3 idx(2) fixed bin,
  P pointer;
declare (ask_, ask_$ask_clr, ask_$ask_c, ask_$ask_cint) external entry,
  (request, type) char(1),
  name char(4),
  long_name char(16),
  (idx1, idx2, flag, i, j) fixed bin,
  search entry;
on condition (program_interrupt) go to get_request;
call ioa ("MODIFY_SYMTAB");
get_request: call ask_$ask_clr;
call ask_("", request);
type, name, long_name=" ";
idx1, idx2=0;
call ask_$ask_c(type, flag);
if flag=1 then do; /*if character returned for type, look for name*/
call ask$_ask_c(name,flag);
if flag=1 then do;
call ask$_ask_cint(idxl,flag);
if flag=1 then do;
call ask$_ask_cint(idx2,flag);
if flag=1 then call ask$_ask_c(long_name,flag);
end;
end;
P=null;
get_ptr: if type="s" then P=addr(sym_ptr->symtab.s_dim);
else if(type="r" | type="t") then P=addr(sym_ptr->symtab.r_dim);
else if type="" then do;
call ioa_("invalid type "a",type);
go to get_request;
end;
if request="l" then do; /*list request*/
if type="" then do; /*with no arguments*/
call ioa_("segments and shafts: "d dimensioned "d used",s_dim,s_used);
call ioa_("reaches and tunnel: "d dimensioned "d used",r_dim,r_used);
end;
else if name="" then do; /*with one argument*/
if type="s" then call ioa_  
("name"-seg idx"-sft idx"-long_name");  
else call ioa_  
("name"-rch idx"-tun idx"-long name");
do j=1 to used;
call ioa_("-a"-d"-d"-a",table(j).sym_name,  
  table(j).idx(1),table(j).idx(2),  
  table(j).symbolic_name);
end;
end;
else do; /*three arguments*/
call search;
call ioa_("-a"-d"-d"-a",
table(i).sym_name, table(i).idx(1), table(i).idx(2)  

end;  
go to get_request;  
end;  
if request="h" then do;  
call loa_("standard input line"/request"-type"-name"-seg or rch idx"-sft or tun idx"-lnam  

type");  
go to get_request;  
end;  
if request="q" then return;  
if type="" then do;  
call ask_("type?",type);  
go to get_ptr;  
end;  
if name="" then call ask_("name?",name);  
if (request="c" | request="a") /*add or change request*/ then do;  
call search;  
if request="a" then do;  
if i=0 then do;  
call loa_("name duplication "a",name);  
go to get_request;  
end;  
used=used+1;  
i=used;  
end;  
else if i=0 then do;  
call loa_("a not found",name);  
go to get_request;  
end;  

end;  
table(i).sym_name=name;  
table(i).symbolic_name=long_name;  
table(i).idx(1)=idx1;  
table(i).idx(2)=idx2;  
go to get_request;  
end;
if request="d" then do; /*delete request*/
call search;
  if i=0 then do;
    call ioa_("a not found",name);
    go to get_request;
  end;
  used=used-1;
  do j=i to used;
    table(j).sym_name=table(j+1).sym_name;
    table(j).symbolic_name=table(j+1).symbolic_name;
    table(j).idx(1)=table(j+1).idx(1);
    table(j).idx(2)=table(j+1).idx(2);
  end;
  go to get_request;
end;
call ioa_("a not a valid request",request);
go to get_request;
end modify_symtab;
initiate_methods: procedure;
dcl 1 methods based (meth_ptr),
   2 nmeths(5) fixed bin,
   2 meth(5,10),
   3 id char(4),
   3 confli_offset offset(confli_area),
   2 confli_area area(900),
   1 confli_criteria based(confli_crit_ptr),
   2 description char(16),
   2 ntests fixed bin,
   2 tests(new_ntests refer(ntests)),
   3 item fixed bin,
   3 quant float bin,
   3 rel char(2);

declare meth_ptr ptr external static;

declare hcs_$make_seg external entry(char(*),char(*),char(*),
   fixed bin(5),ptr,fixed bin(17)),
area_external entry(fixed bin(17),ptr),
(loa_,com_err_) external entry,
     code fixed bin,
(error_table_$namedup,error_table_$segknown) external fixed bin(17),
     wdir_name char(168) aligned,
get_wdir external entry returns(char(168) aligned);
     wdir_name=get_wdir_;
call hcs_$make_seg(wdir_name,"methods","methods",11011b,meth_ptr,code);
if code=0 then do; /*no segment exists*/
call loa_("empty methods segment");
call area_(900,addr(confli_area));
do n=1 to 5;
   nmeths(n)=0;
end;
call modify_methods;
end;
else if (code=error_table$_segknown | code=error_table$_namedup)
  then call ioa("pdb found");
  else call com_err_(code,"initiate_methods");

end initiate_methods;
modify_methods: procedure;

dcl 1 methods based (meth_ptr),
  2 nmeths(5) fixed bin,
  2 meth(5,10),
    3 id char(4),
    3 confli_offset offset(confli_area),
  2 confli_area area(900),
  1 confli_criteria based(confli_crit_ptr),
  2 description char(16),
  2 ntests fixed bin,
  2 tests(new_ntests refer(ntests)),
    3 item fixed bin,
    3 quant float bin,
    3 rel char(2);

declare meth_ptr pointer external static;

declare (ioa_,ioa_$_nni,ask_,ask_$_ask_clr,ask_$_ask_c,ask_$_ask_cint,ask_$_ask_int) external entry,
  (flag,nphase,n,m,new_ntests) fixed bin,
  phase_num entry (char(1)) returns(fixed bin),
  meth_num entry(fixed bin,char(4)) returns(fixed bin),
  p ptr,
  new_description char(16),
  iden char(4),
  phase char(1),
  request char(1);

on condition (program_interrupt) go to get_request;
  call ioa_("MODIFY METHODS");

get_request:call ask_$_ask_clr;
  call ask_("\",request);
  phase=" ";iden=" ";new_description=" ";new_ntests=0;
  call ask_$_ask_c(phase,flag);
  if flag=1 then do;
    call ask_$_ask_c(iden,flag);
    if flag=1 then do;
      call ask_$_ask_c(new_description,flag);
      if flag=1 then call ask_$_ask_cint(new_ntests,flag);
    end;
nphase=phase_num(phase);
if nphase=0 then do;
call ioa_("phase designation "a not recognized",phase);
go to get_request;
end;
if request="a" then do; /*add a new method*/
if new_description="" then do;
call ask_("description?",new_description);
call ask_"$ask_int("number of tests?",new_ntests);
end;
allocate confi_criteria in (confl_area);
nmeths(nphase)=nmeths(nphase)+1;
id(nphase,nmeths(nphase))=iden;
confli_offset(nphase,nmeths(nphase))=confli_crit_ptr;
description=new_description;
call ioa_("item-relation-quantity");
do n=1 to ntests;
call ask_"$ask_int("item(n));
call ask_("relation",rel(n));
call ask_"$ask_flo("quantity",quant(n));
end;
go to get_request;
end;
if request="d" then do; /*delete a method*/
m=meth_num(nphase,iden);
if m=0 then do;
call ioa_("identifier not found, request ignored");
go to get_request;
end;
p=confli_offset(nphase,m);
free p->confli_criteria;
confli_offset(nphase,m)=null;
do n=m to nmeths(nphase)-1;
id(nphase,m)=id(nphase,m+1); /*pack the remaining methods*/
confli_offset(nphase,m)=confli_offset(nphase,m+1);
if request="l" then do; /*list*/
    if nphase=6 then do; /*for "phase specification"*/
        call ioa_("phase number of methods/excavation"="d"/mucking="d"/lining="d"/support="d"/grouting and pumping
        nmeths(1),nmeths(2),nmeths(3),nmeths(4),nmeths(5));
        go to get_request;
    end;
    if iden="" then do;
        call ioa_("code"="number of tests"="description");
        do n=1 to nmeths(nphase);
            p=confli_offset(nphase,n);
            call ioa_(""a"="d"="d",id(nphase,n),p->ntests,p->description);
        end;
        go to get_request;
    end;
    m=meth_num(nphase,iden);
    if m=0 then do;
        call ioa_("method "="a not found",iden);
        go to get_request;
    end;
    /*otherwise list contents of one confli criteria cell*/
    p=confli_offset(nphase,m);
    call ioa_("d"="tests for "="a",p->ntests,p->description);
    call ioa_("item"="relation"="quantity");
    do n=1 to p->ntests;
        call ioa_("d"="a"="f",p->item(n),p->rel(n),p->quant(n));
    end;
    go to get_request;
end;

phase_num: procedure(phase) returns(fixed bin); /*returns the number of the phase where
1=excavation
2=mucking
**3=lining**

**4=support**

**5=grouting and pumping */

dcl table(6) char(1) init('e','m','1','s','g',' '),
phase char(1),
i fixed bin;
do i=1 to 6;
if table(i)=phase then go to ret;
end;
i=0;
ret:return(i);
end phase_num;

meth_num:procedure(nphase, iden) returns(fixed bin); /*returns the subscript for a given identifier*/
dcl iden char(4),
(nphase,i) fixed bin;
do i=1 to nmeths(nphase);
if iden=id(nphase,i) then go to ret;
end;
i=0;
ret:return(i);
end meth_num;

if request = "h" then do; /*help request*/
call ioa_("standard input line"/request"-phase"-identifier"-description"-ntests");
go to get_request;
end;

if request="q" then return;
if request="c" then call ioa_("request not implemented as of this version");
else call ioa_("'a not a valid request",request);
go to get_request;
end modify_methods;
initiate_tunnel: procedure(arg_name);

dcl 1 descr based(descr_ptr),
2 ns fixed bin, /*number of segments dimensioned*/
2 nsegs fixed bin, /*number of segments actually described*/
2 segment (ns),
  3 id char(4),
  3 data(30) float bin,
  3 fsbl,
    4 sup(5) char(4),
    4 gap(5) char(4),
2 nsft fixed bin, /*number of shafts dimensioned*/
2 nshafts fixed bin, /*number of shafts actually described*/
2 shaft (nsft),
  3 id char(4),
  3 data(30) float bin,
  3 fsbl,
    4 exc(3) char(4),
    4 muck(5) char(4),
    4 lin(5) char(4),
    4 sup(5) char(4),
    4 gap(5) char(4),
2 nr fixed bin, /*dimensioned*/
2 nreaches fixed bin, /*used*/
2 reach(nr),
  3 id char(4),
  3 data(5) float bin,
  3 shaft1 fixed bin,
  3 shaft2 fixed bin,
  3 nsegments fixed bin,
  3 segments (10) fixed bin,
  3 fsbl,
    4 exc(3) char(4),
    4 muck(5) char(4),
    4 lin(5) char(4),
2 data(5) float bin,
2 name char(10);

dcl (descr_ptr, meth_ptr) pointer external static;

dcl hcs$make_seg external entry(char(*),char(*),char(*),
                   fixed bin(5),ptr,fixed bin(17)),
    seg_name char(32),
    arg_name char(*),
    (ask$_ask_clr,ask$_ask_int) external entry,
    com$_err_ external entry,
    wdir_name char(168) aligned,
    get_wdir_ external entry returns(char(168) aligned),
    (error_table$_segknown, error_table$_namedup) external fixed bin(17),
    loa_ external entry,
    (nr_in,nsft_in,ns_in) fixed bin,
    code fixed bin(17);

w segname=arg name 1". descr";

wdir_name=get_wdir_; 

call hcs$make_seg(wdir_name,seg_name,seg_name,11011b,descr_ptr,code);
if code=0
  then do; /*if segment has to be created*/
    call ask$_ask_clr;
    call ask$_ask_int("new tunnel"/how many reaches?",nr_in);
    call ask$_ask_int("how many shafts?",nsft_in);
    call ask$_ask_int("how many segments?",ns_in);
    ns=ns_in;
    nsft=nsft_in;
    nr=nr_in; /*necessary because ns and nsft define size of structure*/
    descr, namenaraname;
    call tdl;
  end;
else if (code=error_table$_namedup | code=error_table$_segknown)
  then call loa_("a found",arg_name);
else call com$_err_(code,"initiate$segment");

end initiate$segment;
tdl: procedure;
dcl 1 descr based(descr_ptr),
   2 ns fixed bin, /*number of segments dimensioned*/
   2 nsegs fixed bin, /*number of segments actually described*/
   2 segment (ns),
      3 id char(4),
      3 data(30) float bin,
      3 fsbl,
         4 sup(5) char(4),
         4 gap(5) char(4),
   2 nsft fixed bin, /*number of shafts dimensioned*/
   2 nshafts fixed bin, /*number of shafts actually described*/
   2 shaft (nsft),
      3 id char(4),
      3 data(30) float bin,
      3 fsbl,
         4 exc(3) char(4),
         4 muck(5) char(4),
         4 lin(5) char(4),
         4 sup(5) char(4),
         4 gap(5) char(4),
   2 nr fixed bin, /*dimensioned*/
   2 nreaches fixed bin, /*used*/
   2 reach(nr),
      3 id char(4),
      3 data(5) float bin,
      3 shaft1 fixed bin,
      3 shaft2 fixed bin,
      3 nsegments fixed bin,
      3 segments (10) fixed bin,
      3 fsbi,
         4 exc(3) char(4),
         4 muck(5) char(4),
         4 lin(5) char(4),
   2 data(5) float bin,
   2 name char(10);
dcl (descr_ptr,
meth_ptr) pointer external static;
declare sym_ptr ptr external static,
  1 symtab based (sym_ptr),
  2 s_dim fixed bin,
  2 s_used fixed bin,
  2 s(s_dim),
    3 sym_name char(4), /*short form name*/
    3 symbolic_name char(16), /*long form name*/
    3 idx(2) fixed bin,  /*1=segment, 2=shaft*/
  2 r_dim fixed bin,
  2 r_used fixed bin,
  2 r(r_dim),
    3 sym_name char(4),
    3 symbolic_name char(16),
    3 idx(2) fixed bin,  /*1=reach, 2=tunnel*/
  1 structure based(P),
  2 dim fixed bin,
  2 used fixed bin,
  2 table(dim),
    3 sym_name char(4),
    3 symbolic_name char(16),
    3 idx(2) fixed bin,
  P pointer;
declare (ioa_,ioa_$nnl) external entry,
  j fixed bin,
  new_name char(4),
  shaft_num entry returns(fixed bin),
  seg_num entry(char(4)) returns(fixed bin); 
declare get_meth external entry(fixed bin,fixed bin),
  list_meth external entry(fixed bin,fixed bin);
  dcl (substr,addr,length,null,index,divide) builtin,
  (ask_,ask_$ask_clr,ask_$ask_c,ask_$ask_line,ask_$ask_setline,ask_$ask_cline) external entry,
  (cv_float_) external entry(char(*),fixed bin,float bin),
  (ass_read,get_line,set_DP,mod_with) entry,
  (1,ii,type_num,cur_name_num,flag,code,eq_source_num,reserved,prev,nidx,idx_type) fixed bin,
key_word_num(9) fixed bin init(1,3,4,1,2,3,4,2,3) internal static,
flo float bin,
token char(8),
input_line char(80),
line char(80),
toke char(3),
eq_source char(4),
WP_array(4) ptr,
key_word(9) char(3) init("s","r","t","seg","sft","rch","tun","sha","rea") static,
cur_name char(4),
t char(1),
request char(1),
long bit(1),
WP ptr,
1 typical based(WP),  /*image of descr part*/
  2 ndim fixed bin,
  2 nused fixed bin,
  2 part(ndim),
    3 id char(4),
    3 data(ddim) float bin,
ddim fixed bin,
 ddim_array(4) fixed bin init(40,53,31,0) internal static,
DP ptr,
tdata(ndim) based (DP);
shaft_num:procedure returns(fixed bin); /*reads in name from console, if it is in shaft
  id table, returns index, if not it is added to table*/
declare 1 fixed bin;
call ask_("shaftname?",new_name);
do i=1 to nshafts;
  if new_name=shaft(i).id then return(i);
end;
if nshafts=nsft then do;
call ioa_("shaft table exceeded: ",new_name);
return(0);
end;
nshafts=nshafts+1;
shaft(nshafts).id=new_name;
return(nshafts);
end shaft_num;
seg_num: procedure(name) returns(fixed bin);
dcl i fixed bin, name char(4);
do i=1 to nsegs;
if name=segment(i).id then return(i);
end;
return(0);
end seg_num;

/* START OF THE PROGRAM */
call ioa_("TDL");
if nsegs>0 then go to requests;

/* Entry for initialization of the tunnel description */
call ioa_("specify components:"/reach"=shaft1"=shaft2"=segments");
call ask_$ask clr;
nshafts=0;
nsegs=0;
do nreaches=1 to nr;
reach(nreaches).nsegments=0;
call ask_("",new_name);
if new_name="" then do; /* to end description without describing all reaches */
nreaches=nreaches-1;
go to requests;
end;
reach(nreaches).id=new_name;
reach(nreaches).shaft1=shaft_num; /* shaft_num() for Version II PL/1 */
reach(nreaches).shaft2=shaft_num;
segl: call ask_("segment?",new_name);
add: if seg_num(new_name) ^= 0 then do;
call ioa_("segment name "a not unique, retype line starting with that value",
   new_name);
call ask_$ask clr;
go to seg1;
end;
if nsegs\textless{}ns then do; /*if more segments input than dimensioned*/
  call ioa_("more segment names input than dimensioned: "a",new_name);
  go to requests;
end;

nsegs=nsegs+1;
segment(nsegs).id=new_name;
reach(nreaches).nsegments=reach(nreaches).nsegments+1;
reach(nreaches).segments(reach(nreaches).nsegments)\textast{}nsegs;
call ask_Sask_c(new_name,flag);
if flag=1 then go to add;
end;
nreaches=nr;
/* beginning of request section */

requests:on condition (program_interrupt) go to new_line;
  WP,WP_array(1)=addr(ns);
  WP_array(2)=addr(nsft);
  WP_array(3)=addr(nr);
  WP_array(4)=null;
  ddim dim array(1);
  cur name num=0;
  type_num=1;
/*THIS SECTION OF THE PROGRAM MODIFIES DATA IN THE DLSER SEGMENT*/
call ioa_("modify");

new_line:call get_line;
get_line:procedure;
  call ask_$ask_clr;
  call ask_$ask_line("",input_line);
  i=index(Input_line,"=");
  if i=0
    then if i=1 then line="|| "||substr(input_line,2);
    else line=substr(input_line,1,i-1)||"||"||substr(input_line,i+1);
  else line=input_line;
  call ask_$ask_setline(line);
end get_line;
call ask_("",token);
t=substr(token,1,1);
if t="l" then do;
request=substr(token,2,1);
   if request="" then call ask_("request?",request);
   else;
   if request="q" then return;
   if request="!
      then do;
      /*print out results of test of above*/
      call ioa_("/");
      call ioa_("reach~shaft1~shaft2~segments");
      do i=1 to nreaches;
         call ioa_$nnI("/"a~a~a~a",reach(i).id,shaft(reach(i).shaft1).id,shaft(reach(i).shaft2).id);
         do j=1 to reach(i).nsegments;
            call ioa_$nnI("~a",segment(reach(i).segments(j)).id);
         end;
      end;
      call ioa_("/");
      go to new_line;
   end;
   call ioa_("/");
   go to new_line;
   end;
   call ioa_("request "a not found",request);
   go to new_line;
   end;
if t=">
   then do; /*change type and name*/
      if substr(token,2,1)="" "
      then do;
         call ask_$ask_c(toke,flag);
         if flag=0
            then do;
               if type_num=4 then go to requests; /*nothing after tunnels*/
               if cur_name_num<typical.nused
                  then do;
                     prev=cur_name_num;
                     cur_name_num=cur_name_num+1;
                  end;
               else do;
cur_name_num=1;
type_num=type_num+1;
WP=WP_array(type_num);
dim=ddim_array(type_num);
end;
if type_num=4
then do;
  DP=addr(descr.data);
curname="";
end;
else do;
  DP=addr(typical.part(cur_name_num).data(1));
curname=typical.part(cur_name_num).id;
end;
call ioa_(">",a", key_word(type_num+3),cur_name);
go to set_P;
end;
else;

else toke=substr(token,2);
do i=1 to 9;
if toke=key_word(i) then go to found_kw;
end;
call ioa_("a not a valid type",toke);
go to new_line;
found_kw:
  type_num=key_word_num(i);
WP=WP_array(type_num);
dim=ddim_array(type_num);
if type_num=4
then do;
  DP=addr(descr.data);
curname_num=0;
curname=descr.name;
end;
else do;
call ask_("name?",cur_name);
call set_DP;
end;

set_P: if type_num>2
then do;
    idx_type=divide(type_num,2,17,0); /*for subscript for idx()*/
P=addr(symtab.r_dim); /*ptr to rch or tun symtab*/
end;
else do;
P=addr(symtab.s_dim); /*ptr to seg or sft symtab*/
    idx_type=type_num;
end;
    go to new_line;
end;

if token="="
then do;
    if type_num=4 then do;
        call ioa_("equals not valid for type tunnel");
go to new_line;
end;
    call ask_$ask_c(eq_source,flag); /*for ",name"*/
if flag=0 then do;
    if cur_name_num=1
then do;
        call ioa_("this is the first ",a no source",key_word
(type_num+3));
go to new_line;
end;
eq_source_num=prev;
end;
else do; /*eq source given*/
do eq_source_num=1 to WP->typical.nused;
if eq_source=WP->typical.part(eq_source_num).id
    then go to assign;
end;
call ioa_("a not found in name table",eq_source);
go to new_line;
assign: do i=1 to structure.used;
    nidx=table(i).idx(idx_type);
    if nidx=0 then
        DP->tdata(nidx)=typical.part(eq_source_num).data(nidx);
    end assign;
go to new_line;
end;

if (token="p" | token="prompt")
then do;
    if token="p" then long="0"b; /*short form*/
        else long="1"b; /*long form*/
do ii=1 to structure.used;
nidx=table(ii).idx(idx_type);
    if nidx=0
then do;
    if long then call ioa_$nnl("a ",table(ii).symbolic_name);
        else call ioa_$nnl("a ",table(ii).sym_name);
call ass_read;
    end;
end;
call ioa(""");
go to new_line;
end;

if (token="l" | token="list")
then do;
    if token="l" then long="0"b;
        else long="1"b;
pr_l:do i=1 to structure.used;
nidx=table(i).idx(idx_type);
    if nidx=0 then
    if long then call ioa("a -f",table(i).symbolic_name,tdata(nidx));
        else call ioa("a -f",table(i).sym_name,tdata(nidx));
end pr_l;
go to new_line;
end;
if token="m:
  then do;
    call get_meth(type_num, cur_name_num);
    go to new_line;
  end;
if token ="lm"
  then do;
    call list_meth(type_num, cur_name_num);
    go to new_line;
  end;
/*IF WE HAVE GOTTEN THIS FAR IT MUST BE A VARIABLE NAME*/
do_short: do i=1 to structure.used;
if token=table(i).sym_name
  then do;
    nidx=table(i).idx(idx_type);
    call ass_read_nnl;
    go to new_line;
  end;
end do_short;
/*BY THIS POINT THE TOKEN IS NOT RECOGNIZED*/
call ioa_("token "a not found",token);
go to new_line;
/* internal procedures */
set_DP: procedure;
do cur_name_num=1 to WP->typical.nused;
if cur_name=WP->typical.part(cur_name_num).id
  then do;
    DP=addr(WP->typical.part(cur_name_num).data(1));
    prev=cur_name_num-1;
    return;
  end; /*end of loop*/
end; /*end of loop*/
DP=null;
call ioa_("a not found in name table",cur_name);
go to new_line;
end set_DP;
ass_read: procedure;
new_get: call get_line;
ass_read_nnl:entry;
call ask_("value?",token);
  if token="" then do; /*print out value*/
    call ioa_("'f",tdata(nidx));
    return;
  end;
  if token="=
    then do;
      call ask_"ask_c(token,flag);
      if flag=0
        then do;
          if cur_name_num=1 then do; /*error message*/
            call ioa_("no prev data base, retype value");
            go to new_get;
          end;
          eq_source_num=prev;
        end;
      else do;
        do eq_source_num=1 to typical.nused;
        if token=part(eq_source_num).id
          then go to hell;
        end;
      end;
      hell:tdata(nidx)=typical.part(eq_source_num).data(nidx);
    end;
  else do;
    call cv_float_(token,code,flo);
    if code=0 then tdata(nidx)=flo;
    else do;
      call ioa_("'a not a number or a tunnel section, retype value",token);
      go to new_get;
    end;
  end;
end ass_read;
end tdl;
get_meth: procedure(type_num, cur_name_num);
declare (ask_, ask$_ask_clr, ask$_ask_c, ioa_, ioa$_nnl) external entry,
addr builtin,
phase char(1),
phase_array(5) char(1) init("e", "m", "l", "s", "g") static,
(1, phase_num, type_num, cur_name_num, lim, flag) fixed bin,
fp ptr,
ids(5) char(4) based(fp), /*image of data to be accessed*/
list bit(1);
def 1 descr based(descr_ptr),
 2 ns fixed bin, /*number of segments dimensioned*/
 2 nsegs fixed bin, /*number of segments actually described*/
 2 segment (ns),
    3 id char(4),
    3 data(30) float bin,
    3 fsb1,
        4 sup(5) char(4),
        4 gap(5) char(4),
 2 nsft fixed bin, /*number of shafts dimensioned*/
 2 nshafts fixed bin, /*number of shafts actually described*/
 2 shaft (nsft),
    3 id char(4),
    3 data(30) float bin,
    3 fsb1,
        4 exc(3) char(4),
        4 muck(5) char(4),
        4 lin(5) char(4),
        4 sup(5) char(4),
        4 gap(5) char(4),
 2 nr fixed bin, /*dimensioned*/
 2 nreaches fixed bin, /*used*/
 2 reach(nr),
    3 id char(4),
    3 data(5) float bin,
    3 shaft1 fixed bin,
    3 shaft2 fixed bin,
3 nsegments fixed bin,
3 segments (10) fixed bin,
3 fsbl,
  4 exc(3) char(4),
  4 nuck(5) char(4),
  4 lin(5) char(4),
2 data(5) float bin,
2 name char(10);
dcl (descr_ptr,
  meth_ptr) pointer external static;
on condition(program_interrupt) go to ret;
dcl 1 methods based (meth_ptr),
  2 nmeths(5) fixed bin,
  2 meth(5,10),
  3 id char(4),
  3 confl_offset offset(confl_area),
  2 confl_area area(900),
  1 confl_criteria based(confl_crit_ptr),
  2 description char(16),
  2 ntests fixed bin,
  2 tests(new_ntests refer(ntests)),
  3 item fixed bin,
  3 quant float bin,
  3 rel char(2);
dcl confl_crit_ptr ptr,
  new_ntests fixed bin;
list="O"b;
go to get_phase;
list_num:entry(type_num,cur_name_num);
list="1"b;
get_phase:call ask_("phase?",phase);
do phase_num=1 to 5;
  if phase=phase_array(phase_num)
    then go to get_fp;
  end;
call ioa_("invalid phase "a",phase);
go to get_phase;

/*Set fp, the pointer to the based array ids, which will be used to set*/
/*the method array for each part*/

get_fp:
if phase_num=1 then lim=3; /*set limit on subscript for fsbl array*/
else lim=5;
if type_num=1 /*segment*/
then do;
if phase_num<4 then go to error;
if phase_num=4
then fp=addr(segment(cur_name_num).fsbl.sup(1));
else fp=addr(segment(cur_name_num).fsbl.gap(1));
end;
if type_num=2 /*shafts*/
then do;
if phase_num=1
then fp=addr(shaft(cur_name_num).fsbl.exc(1));
if phase_num=2
then fp=addr(shaft(cur_name_num).fsbl.muck(1));
if phase_num=3
then fp=addr(shaft(cur_name_num).fsbl.lin(1));
if phase_num=4
then fp=addr(shaft(cur_name_num).fsbl.sup(1));
else fp=addr(shaft(cur_name_num).fsbl.gap(1));
end;
if type_num=3 /*reaches*/
then do;
if phase_num=1
then fp=addr(reach(cur_name_num).fsbl.exc(1));
if phase_num=2
then fp=addr(reach(cur_name_num).fsbl.muck(1));
if phase_num=3
then fp=addr(reach(cur_name_num).fsbl.lin(1));
else if phase_num>3 then go to error;
end;
if type_num=4 then do;
call ioa_("no methods for tunnel part");
go to ret;
end;
if list then go to out;
else go to in;
error: call ioa_("phase "a not valid for this part",phase);
go to ret;
in:
    do i=1 to lim;
    ids(i)=" "; /*set method id array to blanks*/
end;
call ask_("meth id?",ids(1));
if ids(1)="all"
    then do ;
        do i=1 to methods.nmets(phase_num);
        if i > lim then go to ret;
        ids(i)=methods.meth(phase_num,i).id;
    end;
else do;
    do i=2 to 5;
    call ask_$(ask_c(ids(i),flag);]
    if flag=0 then go to ret;
end;
go to ret;
out:
    do i=1 to lim;
    call ioa_$(nnl("-a ",ids(i));
end;
    call ioa_(" ");
ret: call ask_$(ask_clr;
end get_meth;
tunnel: procedure(name);
declare name char(*),
    (initiate_symtab, initiate_methods) entry,
    initiate_tunnel entry(char(*));

call initiate_symtab;
call initiate_methods;
call initiate_tunnel(name);
end tunnel;
APPENDIX C

A COST ESTIMATION PROGRAM
Description

This program estimates the construction cost of tunnels and shafts. Overhead and profit are not included. Cost indices may be specified to indicate variations in labor, equipment, or material prices with time and place. Base costs are for Chicago in 1969.

Cost equations and design methods developed by the Harza Engineering Co. (Harza, 1970) are used for calculations. Reference should be made to this report for derivations of the equations. The program will try various construction methods and determine the cheapest. Optionally, costs for all construction methods can be printed out. The program is written in FORTRAN IV and was run at the MIT Information Processing Center on an IBM 360/65 using the OS/MVT operating system.

There are three significant differences between this program and Harza's program. This program tries several construction methods, rather than one. A conflict analyzer is used to eliminate infeasible methods and reduce computations. The input scheme is also different. Free format input of the form: 'variable name' = 'value(s)' is allowed. Default values are assigned to variables if no data is provided.
The third difference is one that can be resolved quite easily. At present, the program assumes that everything along the tunnel (portion excavated from one shaft) is constant. Segments could be introduced to correct this. The same thing can be accomplished with the present program by defining each segment as a separate tunnel and manually adding the results.

Structure

The program consists of a main program and eight subroutines. The main program calls a system subroutine called ERRSET to prevent errors when a "/*" is read. It then calls subroutines to initialize the variables, eliminate infeasible methods, and form combinations. The subroutines are described below. The main program also prints out the cost in dollars per linear foot and dollars per cubic foot.

Subroutine INITIAL assigns default values to the variables used for cost computations. The variable names and an explanation are included in the listing of this program. A subroutine called INPUT is called to read in the data. This data is then substituted for the appropriate default values. INITIAL also computes area, overbreak, excavated diameter, advance rate, quantity of muck, muck loading rate,
grout hole length, and grout take, unless these are specified as input. Geometric relationships or Harza's equations are used for these calculations, whichever is appropriate. The subroutine then returns control to the main program. Subroutine INITSH is actually another entry point to INITAL, the only difference being that INITSH, used for shafts, sets the default values for the variables to the values specified for the previous tunnel.

Subroutine INPUT reads in the data and returns the values to INITAL. The form of the input is exactly that shown in the output listing that follows. Data items must be separated by either a space or a comma. The codes used to specify methods are shown in the main program listing. (It should be noted that cost equations do not exist for some of these methods.) Real (as opposed to integer) data items must contain a decimal point. INPUT calls LPACK, an assembly language subroutine which converts the input name from one character per computer word to four characters per word.

Subroutine CONFLI is called by the main program after the variables have been initialized. It eliminates infeasible methods. The criteria used are shown as comments in
the listing of CONFLI. Subroutine DELETE is called by CONFLI to delete the methods. Control is then returned to the main program.

Subroutine COMBO forms feasible combinations of methods and calls subroutine COSTS (or COSTSH) to compute the cost of the combination. Some method combinations are not feasible, even though the individual methods are. COMBO eliminates such methods. The criteria used are:

1. If the tunnel design life is greater than five years, either support or lining (or both) must be used.
2. If design life is greater than 10 years, some sort of lining must be used.
3. If liner plate support is used, no lining is necessary.
4. Pregrouting cannot be performed with mole excavation.

COSTS computes the cost of a tunnel; COSTSH computes the cost of a shaft. Both use of the equations developed by Harza. Labor, material, and equipment costs are computed, and the minimum cost is determined. If costs for all combinations are to be printed, COSTS prints them. Control is
returned to COMBO, which computes the next combination. Finally, control is returned to the main program, which begins on the next tunnel or shaft.

Output and Listing

The output following is the result of cost calculations for one tunnel and one shaft. An echo-print of the input is first produced. A *EOF card serves as a delimeter to separate tunnels and shafts. Any number of tunnels and shafts can be estimated provided these cards are used to separate them.

The next item of output is a list of all variables and their values. *DEFAULT proceeding the variable indicates that the program assumed a value. Finally, the cost are printed. This portion of the output has been retyped, because the real output format is not convenient for publication purposes.

Following the output is a listing of the main program and subroutines. They appear in the same order in which they were described above.
EXCAVATE 10,000
WICKING 12,000,000
LINING 0
SUPPORT 20,000
GRAVITING 0
DIAMETER=30, FEET
STRENGTH=2,000
RON=40
OWNEC=4, YEARS
SETC费=0
LENGTH=7,500, FEET
DEPTH=600, FEET

CASE II: SWISS
INITIAL VALUES OF VARIABLES

EXCAVATE = 12.5
SUPPORT = 29.41
LINING = 0
GROUTING = 0
JACKING = 12.22.3

*DEFAULT SHAPE CODE = 1
SHARE CODE = 2

*DEFAULT LINING THICKNESS = 8.6 INCHES
DESIGN LIFE = 4.5 YEARS
DEPTH = 4000 FEET
DIAMETER NOT INCLUDING OVERBREAK = 21.7 FEET

*DEFAULT AREA = 314.5 SQUARE FEET
*DEFAULT EXCAVATED DIAMETER INCLUDING OVERBREAK = 21.2 FEET
LENGTH OF TUNNEL = 2644 FEET

*DEFAULT REACH LENGTH = 2444 FEET

*DEFAULT SLOPE = 14

*DEFAULT DISTANCE FROM BASE OF SHAFT TO WORKING FACE = 5.0 MILES
*DEFAULT DISTANCE TO DISPOSAL AREA = 1.0 MILES

*DEFAULT LABOR COST INDEX = 16.50
*DEFAULT EQUIPMENT COST INDEX = 15.50

*DEFAULT MATERIALS COST INDEX = 12.50

*DEFAULT INITIAL GROUND WATER INFLOW = 0.1 GPW/MILE

*DEFAULT SUSTAINED GROUND WATER INFLOW = 0.6 GPW/MILE

*DEFAULT GROUND WATER PRESSURE = 1.0 ATM

ROCK STRENGTH = 3000 - 6000 PSI

ROCK QUALITY DESIGNATION = 60 - PER CENT

*DEFAULT BULKING FACTOR = 0.40

*DEFAULT CONVENTIONAL OVERBREAK = 3.22

*DEFAULT WIDE OVERBREAK = 1.9
*DEFAULT CONVENTIONAL ADVANCE RATE= 42.7 FEET PER DAY

*DEFAULT HOLE ADVANCE RATE= 19.5 FEET PER DAY

*DEFAULT CONVENTIONAL QUANTITY OF MUCK= 511.2 CUBIC FEET PER FOOT OF LENGTH

*DEFAULT HOLE QUANTITY OF MUCK= 494.8 CUBIC FEET PER UNIT LENGTH

*DEFAULT RATE USED TO SIZE MUCKING EQUIPMENT (CONVENTIONAL)= 33.2 CUBIC YARDS PER HOUR

*DEFAULT RATE USED TO SIZE MUCKING EQUIPMENT (HOLE)= 14.6 CUBIC YARDS PER HOUR

*DEFAULT LENGTH OF DRILL HOLES FOR GROUTING= 15.71 FEET PER FOOT

*DEFAULT GROUT TAKE= 0.31 CUBIC FEET PER FOOT

*DEFAULT ROCK TEMPERATURE= 65° DEGREES F

*DEFAULT OUTSIDE TEMPERATURE= 50° DEGREES F
<table>
<thead>
<tr>
<th>METHOD TO BE TAKEN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>THE 2 EXCAVATION METHODS ARE</td>
<td>12 50</td>
</tr>
<tr>
<td>THE 4 SUPPORT METHODS ARE</td>
<td>41</td>
</tr>
<tr>
<td>THE 1 LIVING METHODS ARE</td>
<td>0</td>
</tr>
<tr>
<td>THE 1 GROUTING AND MIDDING METHODS ARE</td>
<td>110</td>
</tr>
<tr>
<td>THE 2 WICKING METHODS ARE</td>
<td>12 22 36</td>
</tr>
</tbody>
</table>
Tunnel #1

THE RESULTS OF THE MINIMUM COSTS OF 6 COMBINATIONS

All costs are in dollars per linear foot.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>For Excavation:</td>
<td>10.</td>
</tr>
<tr>
<td>Support:</td>
<td>41.</td>
</tr>
<tr>
<td>Lining:</td>
<td>00.</td>
</tr>
<tr>
<td>Grouting and pumping:</td>
<td>110.</td>
</tr>
<tr>
<td>Mucking:</td>
<td>22.</td>
</tr>
</tbody>
</table>

The costs are:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Excavation:</td>
<td>161.75</td>
</tr>
<tr>
<td>Support:</td>
<td>28.18</td>
</tr>
<tr>
<td>Lining:</td>
<td>00.00</td>
</tr>
<tr>
<td>Grouting and pumping:</td>
<td>15.43</td>
</tr>
<tr>
<td>Mucking:</td>
<td>86.83</td>
</tr>
<tr>
<td>Air conditioning:</td>
<td>18.11</td>
</tr>
</tbody>
</table>

Total: 310.30

Same costs in dollars per cubic foot

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Excavation:</td>
<td>0.52</td>
</tr>
<tr>
<td>Support:</td>
<td>0.09</td>
</tr>
<tr>
<td>Lining:</td>
<td>0.00</td>
</tr>
<tr>
<td>Grouting and pumping:</td>
<td>0.05</td>
</tr>
<tr>
<td>Mucking:</td>
<td>0.28</td>
</tr>
<tr>
<td>Air conditioning:</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Total: 0.99
EXCAVATE 1095
MUCKING 2
LINING 10 20
SUPPORT 2 41 00
GROUTING 2
DIAMETER=9, FEET
END
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCAVATE</td>
<td>10 50</td>
</tr>
<tr>
<td>SUPPORT</td>
<td>20 41</td>
</tr>
<tr>
<td>LINING</td>
<td>10 21</td>
</tr>
<tr>
<td>GROUTING</td>
<td>0</td>
</tr>
<tr>
<td>DUNCING</td>
<td>0</td>
</tr>
</tbody>
</table>

*Default Shape Code = 1
*Default Shaft Code = 2
*Default Lining Thickness = 8.0 Inches
*Default Design Life = 45 Years
*Default Depth = 400 Feet

- Diameter not including overbreak = 8.0 Feet
- Default Area = 50.2 Square Feet
- Default Excavated Diameter Including Overbreak = 9.4 Feet
- Default Length of Tunnel = 2440 Feet
- Default Reach Length = 2640 Feet
- Default Slope = 0.5
- Default Distance from Base of Shaft to Working Face = 5.0 Miles
- Default Distance to Disposal Area = 1.7 Miles
- Default Labor Cost Index = 1.5
- Default Equipment Cost Index = 1.4
- Default Material Cost Index = 1.6
- Default Initial Ground Water Inflow = 0.5 GPM/Mile
- Default Sustained Ground Water Inflow = 0.7 GPM/Mile
- Default Ground Water Pressure = 150 PSI
- Default Rock Strength = 7750.0 PSI
- Default Rock Quality Designation = 50.0 Percent
- Default Arling Factor = 0.4
- Default Conventional Overbreak = 0.02
- Default Hole Overbreak = 0.1
- Default Conventional Advance Rate = 3.2 Feet per Day
DEFAULT HOLE ADVANCE RATE = 15.4 FEET PER DAY

DEFAULT CONVENTIONAL QUANTITY OF MUCK = 87.9 CUBIC FEET PER FOOT OF LENGTH

DEFAULT HOLE QUANTITY OF MUCK = 77.9 CUBIC FEET PER UNIT LENGTH

DEFAULT RATE USED TO SIZE MUCKING EQUIPMENT (CONVENTIONAL) = 7.6 CUBIC YARDS PER HOUR

DEFAULT RATE USED TO SIZE MUCKING EQUIPMENT (HOLE) = 1.9 CUBIC YARDS PER HOUR

DEFAULT LENGTH OF DRILL HOLES FOR GRouting = 6.2 FEET PER FOOT

DEFAULT GROUT TAKE = 0.13 CUBIC FEET PER FOOT

DEFAULT ROCK TEMPERATURE = 50.0 DEGREES F

DEFAULT OUTDOOR TEMPERATURE = 50.0 DEGREES F
THE RESULTS OF THE MINIMUM COSTS OF 12 COMBINATIONS

All costs are in dollars per linear foot.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>For Excavation:</td>
<td>50</td>
</tr>
<tr>
<td>Support:</td>
<td>00</td>
</tr>
<tr>
<td>Lining:</td>
<td>10</td>
</tr>
<tr>
<td>Grouting and pumping</td>
<td>00</td>
</tr>
<tr>
<td>Mucking:</td>
<td>00</td>
</tr>
</tbody>
</table>

The costs are:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Excavation:</td>
<td>186.00</td>
</tr>
<tr>
<td>Support:</td>
<td>00.00</td>
</tr>
<tr>
<td>Lining:</td>
<td>143.11</td>
</tr>
<tr>
<td>Grouting and pumping:</td>
<td>00.00</td>
</tr>
<tr>
<td>Mucking:</td>
<td>80.57</td>
</tr>
<tr>
<td>Air conditioning:</td>
<td>00.00</td>
</tr>
<tr>
<td>Total:</td>
<td>409.69</td>
</tr>
</tbody>
</table>

Same costs in dollars per cubic foot

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Excavation:</td>
<td>3.70</td>
</tr>
<tr>
<td>Support:</td>
<td>0.00</td>
</tr>
<tr>
<td>Lining:</td>
<td>2.85</td>
</tr>
<tr>
<td>Grouting and pumping:</td>
<td>0.00</td>
</tr>
<tr>
<td>Mucking:</td>
<td>1.60</td>
</tr>
<tr>
<td>Air conditioning:</td>
<td>0.00</td>
</tr>
<tr>
<td>Total:</td>
<td>8.15</td>
</tr>
</tbody>
</table>
---- FOR THE METHOD VARIABLES THE 'M' PREFIX INDICATES THE ARRAY WHICH CONTAINS THE METHOD CODES. THE 'N' PREFIX INDICATES THE VARIABLE WHICH CONTAINS THE NUMBER OF ELEMENTS IN THE ARRAY. FOR EXAMPLE, 'NEXCAV' CONTAINS THE NUMBER OF ELEMENTS IN 'MEXCAV'.

MEANINGS OF METHOD CODES

EXCAVATION METHODS
17 DRILL AND BLAST - CONVENTIONAL
20 DRILL AND BLAST - SMOOTHWALL
30 HEADING AND BENCH
40 MULTIPLE DRIFT
50 MECHANICAL MOLE
60 MOLE WITH LASER
70 MOLE WITH SURFACTANTS
80 MOLE WITH SHAPED CHARGES
90 COMPRESSED AIR
100 CONVENTIONAL - SHIELD
110 MOLE - SHIELD

SUPPORT METHODS
20 STEEL SUPPORTS
31 LINER PLATES - STEEL
32 LINER PLATES - CAST IRON
33 LINER PLATES - PRECAST CONCRETE
40 ROCK BOLTS WITH WIRE MESH
41 ROCK BOLTS WITHOUT WIRE MESH
50 SHOTCRETE OR GUNNITE
51 SHOTCRETE AND ROCK BOLTS
52 NONE
LINING METHODS

10 CONCRETE - NOMINAL
11 CONCRETE - STRUCTURAL
20 SHOTCRETE - NOMINAL
21 SHOTCRETE - STRUCTURAL

GROUTING AND PUMPING METHODS
FIRST DIGIT -> 1. GROUT, 2. PREGROUT, 0. NO GROUT
SECOND DIGIT -> 1. CHEMICAL, 2. CONCRETE, 0. NONE
THIRD DIGIT -> 1. PUMPS, 0. NO PUMPS

MUCKING METHODS
11 ELECTRIC TRAIN
12 DIESEL TRAIN
21 ELECTRIC TRUCK
22 DIESEL TRUCK
30 BELT CONVEYOR
40 SLURRY PUMP

IMPLICIT REAL (L,M)
INTEGER MEXCAV, MSHPR, MLNING, MGTAPP, MMUCKG, METH
INTEGER COUNT, SHAPE, PRINT, SFTCDE, LIMIT, SHSWCH
COMMON/ MINIM/ STORE, METH(5), PRICE(6)
COMMON/ DATA/ SHAPE, SFTCDE, LTHICK, DNLIFE, DEPTH, DIAM, AREA, EXDIAM,
LENGTH, FL, SLOPE, DOSTW, DSS, LABOR, EQUIP, MALS, GWINFI, GWINFS,
2GPPR, SFT, ROD, RULKF, DVBRRK(2), ADVANC(2), OMUCK(2), RML(2), HL, GT,
3OUT, TEMP, TEMPD
COMMON/ CONTROL, INOUT, SHSWCH
C----- SUPPRESS END OF FILE ERROR MESSAGE AND TRACERBACK
CALL ERRSET(217, -1, 1)
C----- SET NUMBER OF TUNNELS AND NUMBER OF SHAFTS TO ZERO
NTUN=1
ISFT=0
TOUIJT=6

C----PRINT HEADER
10 WRITE(IOUT,9999)
9999 FORMAT(1HI,T4T,'MASSACHUSETTS INSTITUTE OF TECHNOLOGY',/,
1T52, 'CIVIL ENGINEERING DEPARTMENT',/,'T51,
1MATERIALS RESEARCH LABORATORY',/,'T57, 'TUNNEL COST MODEL',/,
3T64, 'VERSION 2.3',//)

C----TUNNEL COMPUTATIONS ARE FIRST: SET SWITCH
SHSWCH=1
NTUN=NTUN+1

C----INITIALIZE ALL VARIABLES
CALL INITIAL
STORE=9.,E7.

C----ELIMINATE METHODS WHICH CONFLICT WITH PHYSICAL PARAMETERS

C----CHANGE ROD FROM PER CENT TO DECIMAL FOR CONFLICT
ROD=ROD/100.
CALL CONFLICT

C----CHANGE ROD BACK TO PER CENT
ROD=ROD*100.

C----FIND COMPATIBLE COMBINATIONS OF METHODS AND CALCULATE COSTS
CALL COMBO
WRITE(IOUT,498) NTUN
498 FORMAT(1H+,T123,'TUNNEL#',I3)
PRINT 590,METH,PRICE,STORE

500 FORMAT(1H+'FOR EXCAVATION':',I3,', SUPPORT':',I3,', LINING':',I3',
1', 'GRouting AND PUMPING':',I4,', MUCKING':',I3',/
2', THE COSTS ARE: EXCAVATION':',F7.2,', SUPPORT':',F7.2,', LINING',',
3F7.2,', 'GRouting AND PUMPING':',F7.2,', MUCKING':',F7.2',',/
4', 'AIR CONDITIONING':',F7.2',', TOTAL':',F7.2',//)
DO 100 I=1,6
100 PRICE(I)=PRICE(I)/AREA
STORE=STORE/AREA
WRITE(IOUT,501)
501 FORMAT(1H2,T4T,'SAME COSTS IN DOLLARS PER CURRIC FOOT')
WRITE(IOUT,502) PRICE,STORE
502 FORMAT(1H' EXCAVATION':',F7.2,', SUPPORT':',F7.2',', LINING':',
IF7.2, 'GROUTING AND PUMPING:' , F7.2, ' , MUCKING:' , F7.2, ' ,/
2' AIR CONDITIONING: ' , F7.2, ' , TOTAL:' , F7.2, ' ,/
C-----PUT PAGE
WRITE(IOUT,1000)
C-----IF THERE IS A SHAFT, SET SWITCH AND COMPUTE COST
IF(SFTCDE.EQ.2) SHSWCH=2
GO TO (20,15),SHSWCH
15 CALL INITSH
STOR9=96(87)
ISFT=ISFT+1
CALL COMBO
WRITE(IOUT,499) ISFT
499 FORMAT(1H+,T124,SHAFT#,13)
WRITE(IOUT,500) METH,PRICE,STORE
DO 200 J=1,6
200 PRICE(J)=PRICE(J)/AREA
STORE=STORE/AREA
WRITE(IOUT,501)
WRITE(IOUT,502) PRICE,STORE
C-----PUT PAGE
200 WRITE(IOUT,1000)
1000 FORMAT(IHI)
GO TO 1
END
SUBROUTINE INITAL

--- THIS SUBROUTINE Initializes VARIABLES FOR THE PROGRAM

<table>
<thead>
<tr>
<th>FORTRAN NAME</th>
<th>INPUT NAME</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEXCAV</td>
<td>EXCAVATE</td>
<td></td>
<td>EXCAVATION METHOD CODE</td>
</tr>
<tr>
<td>NEXCAV</td>
<td>NUMBER OF EXCAVATION METHODS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSUPRT</td>
<td>SUPPORT</td>
<td></td>
<td>SUPPORT METHOD CODE</td>
</tr>
<tr>
<td>NSUPRT</td>
<td>NUMBER OF SUPPORT METHODS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLNING</td>
<td>LINING METHOD CODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLNING</td>
<td>NUMBER OF LINING METHODS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MGTAPP</td>
<td>GROUTING</td>
<td></td>
<td>GROUTING AND PUMPING METHOD</td>
</tr>
<tr>
<td>NGTAPP</td>
<td>NUMBER OF GROUTING AND PUMPING METHODS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMUCKG</td>
<td>MUCKING</td>
<td></td>
<td>MUCKING METHOD CODE</td>
</tr>
<tr>
<td>NMUCKG</td>
<td>NUMBER OF MUCKING METHODS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHAPE</td>
<td>SHAPECOD</td>
<td></td>
<td>SHAPE CODE 1=CIRCULAR, 2=ARCH, 3=BASKETHANDLE</td>
</tr>
<tr>
<td>SFTCDE</td>
<td>SHAFTCOD</td>
<td></td>
<td>1-&gt;TUNNEL BUILT FROM PORTAL</td>
</tr>
<tr>
<td></td>
<td>2-&gt;TUNNEL BUILT FROM SHAFT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTHICK</td>
<td>THICKNES INCHES</td>
<td>DESIGN LIFE</td>
<td></td>
</tr>
<tr>
<td>DNLIFE</td>
<td>DESIGN YEARS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEPTH</td>
<td>DEPTH FEET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAM</td>
<td>DIAMETER FEET</td>
<td>TUNNEL DIAMETER</td>
<td></td>
</tr>
<tr>
<td>AREA</td>
<td>AREA FT²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXDIAM</td>
<td>EXDIAMETER FEET</td>
<td>EXCAVATED DIAMETER INCLUDING OVERBREAK</td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td>LENGTH FEET</td>
<td>LENGTH OF TUNNEL</td>
<td></td>
</tr>
<tr>
<td>RL</td>
<td>REACH LENGTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLOPE</td>
<td>SLOPE SEGMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBSTWF</td>
<td>DISTANCE MILES</td>
<td>DISTANCE FROM BASE OF SHAFT TO FACE</td>
<td></td>
</tr>
<tr>
<td>DDS</td>
<td>DISPOSAL MILES</td>
<td>DISTANCE TO DISPOSAL AREA</td>
<td></td>
</tr>
<tr>
<td>LABORI</td>
<td>LABOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EQUIPI</td>
<td>EQUIPMENT COST INDEX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATLSI</td>
<td>MATERIAL</td>
<td>MATERIALS COST INDEX</td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C 22</td>
<td>GWINF1</td>
<td>Initial GPM/MILE Initial Ground Water Inflow</td>
<td></td>
</tr>
<tr>
<td>C 23</td>
<td>GWINFS</td>
<td>Sustained GPM/MILE Sustained Ground Water Inflow</td>
<td></td>
</tr>
<tr>
<td>C 24</td>
<td>GRWPRS</td>
<td>Ground ATM Ground Water Pressure</td>
<td></td>
</tr>
<tr>
<td>C 25</td>
<td>STRNTH</td>
<td>Strength PSI Rock Strength</td>
<td></td>
</tr>
<tr>
<td>C 26</td>
<td>ROD</td>
<td>Rock % Rock Quality Designation</td>
<td></td>
</tr>
<tr>
<td>C 27</td>
<td>BULKF</td>
<td>Bulking Factor</td>
<td></td>
</tr>
<tr>
<td>C 28</td>
<td>OVRB(1)</td>
<td>Overbreak for Conventional Excavation Method</td>
<td></td>
</tr>
<tr>
<td>C 29</td>
<td>OVRB(2)</td>
<td>Overbreak for Mole</td>
<td></td>
</tr>
<tr>
<td>C 31</td>
<td>ADVANC(1)</td>
<td>Advanc FT/24HR Advance Rate for Conventional</td>
<td></td>
</tr>
<tr>
<td>C 32</td>
<td>ADVANC(2)</td>
<td>Advanc FT/24HR Advance Rate for Mole</td>
<td></td>
</tr>
<tr>
<td>C 33</td>
<td>OMUCC(1)</td>
<td>OMUCC FT**3/FT Quant. Muck/Length - Convo.</td>
<td></td>
</tr>
<tr>
<td>C 34</td>
<td>OMUCC(2)</td>
<td>OMUCCM FT**3/FT Quant. Muck/Length - Mole</td>
<td></td>
</tr>
<tr>
<td>C 35</td>
<td>RML(1)</td>
<td>RML YD**3/HR Rate Used to Size Mucking Equipment - Conventional</td>
<td></td>
</tr>
<tr>
<td>C 36</td>
<td>RML(2)</td>
<td>RML YD**3/HR Rate Used to Size Mucking Equipment - Mole</td>
<td></td>
</tr>
<tr>
<td>C 37</td>
<td>HOLE</td>
<td>Hole FT/FT Length of Grouting Holes</td>
<td></td>
</tr>
<tr>
<td>C 38</td>
<td>OUT</td>
<td>Out FT**3/FT Grout Take</td>
<td></td>
</tr>
<tr>
<td>C 39</td>
<td>TEMPR</td>
<td>Rock Temp Deg F Temperature of Rock</td>
<td></td>
</tr>
<tr>
<td>C 40</td>
<td>TEMPD</td>
<td>Out Temp Deg F Temperature of Outdoors</td>
<td></td>
</tr>
</tbody>
</table>

SUBROUTINE INITIAL
IMPLICIT REAL (L,M)
REAL*4 NAME, LABEL, MESSAG, MES1, MES2, SNT
INTEGER COUNT, SHAPE, PRINT, SET0DE, LIMIT, SHSWCH
INTEGER MEXCAV, MSUPRT, MLNINQ, MGAPP, MMUCCG, M ETH
DIMENSION IDATA(12), RDATA(12), ITYPE(12), NAME(50), MESSAG(50),
SNT(5), ALPHA(3), ARRAY(43)
MLNING. MGTPPR(12). NGTAPP. MMUCKG(12). NMUCKG
COMMON/DATA/SHAPE.SFTCDE. LTHICK. DLNLF. DEPTH. DIAM. AREA. EXDIAM.
LENGTH. RL. SLOPE. DBSTWF. DDS. LABORI. EQUIPI. MATLSI. GWINFI. GWINFS.
2GRWPRS. STRNTH. ROCK. BULKF. OVRBRK(2). ADVANC(2). QMUCK(2). RML(2). HL. GT.
3OUT. TEMPR. TEMPO
COMMON/CONTOL/ IOUT. SSHWCH
EQUIVALENCE (ARRAY(1), LTHICK)
DATA ALPHA/10/ .9.1.85/
DATA MES1. MES2/ 'DEFAULT',
DATA NAME/ 'EXCAVATE', 'SUPPORT', 'LINING', 'GROUTING', 'MUCKING',
1' SHAPE', 'SHAFTCOD', 'THICKNES', 'DESIGN', 'DEPTH', 'DIAMETER',
2' AREA', 'EXDIAMET', 'LENGTH', 'RL', 'SLOPE', 'DISTANCE', 'DISPOSAL',
3' LABOR', 'EQUIPMENT'. 'MATERIAL'. 'INITIAL'. 'SUSTAIN', 'GROUND',
4' STRENGTH', 'ROD', 'BULKING', 'OVERBRK', 'LOVERBRK'. 'ADVANC',
5' MLADVANC'. 'QMUCK'. 'QMUCKM'. 'RMLC'. 'RMLM'. 'HCLE'. 'GROUT'. 'OUT',
6' ROCKTEMP'. 'OUTDTEMP'
DATA SNT/MEXCAV. MSUPRT. MLNING. MGTPPR. MMUCKG. 'SHAPE',
1'SFTCDE'. 'LTHICK'. 'DLNLF'. 'DEPTH'. 'DIAM'. 'AREA'. 'EXDIAM',
2'LENGTH'. 'RL'. 'SLOPE'. 'DBSTWF'. 'DDS'. 'LABORI'. 'EQUIPI'. 'MATLSI',
3'GWINFI'. 'GWINFS'. 'GRWPRS'. 'STRNTH'. 'ROD'. 'BULKF'. 'OVRBRK(1)',
4'OVRBRK(2). ADVANC(1'. 'ADVANC(2'. 'QMUCK(1'. 'QMUCK(2). 'RML(1',
5'RML(2). 'HL'. 'GT'. 'OUT'. 'TEMPR'. 'TEMPO/
C-----SPECIFY DATA SET REFERENCE NUMBER
IOUT=6
C-----ASSIGN NUMBER OF INPUT NAMES
LIMIT=60
C-----ASSIGN DEFAULT VALUES
MEXCAV( 1)=10
MEXCAV( 2)=10
MEXCAV( 3)=10
MEXCAV( 4)=10
MEXCAV( 5)=10
MEXCAV( 6)=10
MEXCAV( 7)=10
MEXCAV( 8)=10
C------THIS IS THE ENTRY POINT FOR SHAFT DATA INPUT, FROM THIS POINT, THE
C------VALUES LEFT FROM THE TUNNEL OPERATIONS BECOME THE NEW DEFAULT
C------VALUES
ENTRY INITSH
C
DO 1 I=1,LIMIT
1 MESSAG(I)=MFS1
C
PRINT 2000
PRINT 2000
PRINT 2000
PRINT 2000
10 CALL INPUT(LABEL, I TYPE, IDATA, RDATA, COUNT)
15 IF (LABEL) 15. 900. 15
15 IF (LABEL .NE. 1.0) GO TO 16
PRINT 3000
3000 FORMAT ( 'END OF FILE (/) FOUND AND EXECUTION TERMINATED')
STOP
16 DO 20 INDEX=1. LIMIT
16 IF (LABEL .EQ. NAME(INDEX)) GO TO 30
20 CONTINUE
DO 21 INDEX=1. LIMIT
21 CONTINUE
PRINT 25. LABEL
25 FORMAT ( 'NAME,' IS AN INVALID DATA NAME')
GO TO 10
30 MESSAG (INDEX) = MES2
30 IF (INDEX .GE. 8) GO TO 40
30 GO TO (1.1, 102.103.104.105.106.107). INDEX
40 ISUBR = INDEX - 7
40 ARRAY(ISUBR) = RDATA(1)
GO TO 10
101 DO 100 I=1. COUNT
100 MEXCAV(I) = IDATA(I)
100 MEXCAV = COUNT
GO TO 10
102 DO 200 I=1. COUNT
200 MNSUPRT(I) = IDATA(I)
200 MNSUPRT = COUNT
GO TO 10
103 DO 300 I=1. COUNT
300 MLNING(I) = IDATA(I)
300 MLNING = COUNT
GO TO 10
104 DO 400 I=1,COUNT
400 MGTAPP(I)=IDATA(I)
NGTAPP=COUNT
GO TO 10
105 DO 500 I=1,COUNT
500 MMUCKG(I)=IDATA(I)
NMUCKG=COUNT
GO TO 10
106 SHAPE=IDATA(1)
GO TO 10
107 SFTCDE=IDATA(1)
GO TO 10
C
100 DO 199 I=1,4
199 PRINT 2000
2000 FORMAT(180)
C
PRINT 1000
1000 FORMAT(1H1,52X,'INITIAL VALUES OF VARIABLES')
IF(.NOT. shswch) WRITE(IOUT,10000)
10000 FORMAT(62X,'FOR SHAFTS')
C
C-----IF NO AREA IS GIVEN, COMPUTE IT
   IF(MESSAG(12).EQ.MES1) AREA=3.14*((DIAM+.5)**2)
C-----IF NO DIAMETER IS GIVEN, COMPUTE IT
   IF(MESSAG(11).EQ.MES1) DIAM=2*SQRT(AREA/3.1416)
C-----IF NO OVERBREAK FOR CONVENTIONAL EXCAVATION IS GIVEN, COMPUTE IT
   IF(MESSAG(28).EQ.MES1) OVRBRK(1)=(1+1-.5*RQD)/12
C-----IF EXCAVATED DIAMETER (INCLUDING OVERBREAK) IS NOT INCLUDED, COMPUTE IT
   IF(MESSAG(13).EQ.MES1) EXDIAM=.167*DIAM*(1-.5*.01*RQD)
C
C-----IF NO ADVANCE RATE IS GIVEN, COMPUTE IT
   GO TO (10001,10002) shswch
TUNNELS
CONVENTIONAL
10001 IF (MESSAG(35).EQ, MES1) ADVANC(1) = ALPHA(SHAPE) * (((0.1*EXDIAM**2.25* 
1ROD+1500)/(EXDIAM**1.5*EXP(0.35*EXDIAM)*(1+2.5*STRNTH/10000000)) 
2)) *((GWINFO+1500)/(5.0*GWINFO+1500))

MOLES
IF (MESSAG(31).EQ, MES1) ADVANC(2) = EXP(5.35-EXDIAM/58.1-0.66*((10.0*39% 
1STRNTH/EXDIAM**2-12))/100000000) *(GWINFO+1500)/(5.0*GWINFO+1500)
GO TO 10003

SHAFTS
10002 IF (GWINFO=1) 
CONVENTIONAL
IF (MESSAG(30).EQ, MES1) ADVANC(1) = 0.05*GWINFO*ROD**1/10.0-

10.0-DEPTH/60000)

MOLF
IF (MESSAG(31).EQ, MES1) ADVANC(2) = 3.0*GWINFO*ROD**2/(EXDIAM*SQRT(STRNTH 
1+5.0*GWINFO))

C----- IF NO REACH LENGTH IS GIVEN, ASSUME IT IS THE LENGTH
10003 IF (MESSAG(15).EQ, MES1) RL=LENGTH
C----- IF NO DISTANCE FROM BASE OF SHAFT TO WORKING FACE IS GIVEN, ASSUME 
C IT IS THE SAME AS THE REACH LENGTH
C----- IF NO QUANTITY OF MUCK IS GIVEN, COMPUTE IT
C CONVENTIONAL
IF (MESSAG(17).EQ, MES1) DBSTWF=RL/5280
C----- IF MUCK RATE IS NOT GIVEN, COMPUTE IT
C CONVENTIONAL
IF (MESSAG(32).EQ, MES1) QMUCK(1) = C.785*((1+OVRBRK(1))*EXDIAM)**2 
1*(1+RULKF)
C MOLF
IF (MESSAG(33).EQ, MES1) QMUCK(2) = C.785*((1+OVRBRK(2))*EXDIAM)**2 
1*(1+RULKF)
C----- IF HOLE LENGTH IS NOT GIVEN, COMPUTE IT
C CONVENTIONAL
IF (MESSAG(34).EQ, MES1) RML(1) = QMUCK(1)*ADVANC(1)/648.
C MOLF
IF (MESSAG(35).EQ, MES1) RML(2) = QMUCK(2)*ADVANC(2)/648.
IF (MESSAG(36) .EQ. MESS1) HLF=(3.1416*DIAM)**(0.25)
C-----IF CORET TAKE IS NOT GIVEN, COMPUTE IT
IF (MESSAG(37) .EQ. MESS1) GT=(3.1416*DIAM)**(1.0+0.01*GWINFS)/206.0
C
PRINT 1001, MESSAG(1), NAME(1), (MEXCAV(I), I=1, NEXCAV)
PRINT 1001, MESSAG(2), NAME(2), (MSUPRT(I), I=1, NSUPRT)
PRINT 1001, MESSAG(3), NAME(3), (MLNING(I), I=1, NLNING)
PRINT 1001, MESSAG(4), NAME(4), (MGTAPP(I), I=1, NGTAPP)
PRINT 1001, MESSAG(5), NAME(5), (MMUCKG(I), I=1, NMUCKG)
1001 FORMAT (1H0, A8, 1H, A8, '=I', 12I4)
PRINT 1006, MESSAG(6), SHAPE
1006 FORMAT (1H0, A8, 1X, 'SHAPE CODE=', 12)
PRINT 1007, MESSAG(7), SFTCDE
1007 FORMAT (1H0, A8, 1X, 'SHAFT CODE=', 12)
PRINT 1008, MESSAG(8), LTHICK
1008 FORMAT (1H0, A8, 1X, 'LINING THICKNESS=', F4.1, ' INCHES')
PRINT 1009, MESSAG(9), DNLIFE
1009 FORMAT (1H0, A8, 1X, 'DESIGN LIFE=', F5.1, ' YEARS')
PRINT 1010, MESSAG(10), DEPTH
1010 FORMAT (1H0, A8, 1X, 'DEPTH=', F6.1, ' FEET')
PRINT 1011, MESSAG(11), DIAM
1011 FORMAT (1H0, A8, 1X, 'DIAMETER NOT INCLUDING OVERBREAK=', F5.1, ' FEET')
PRINT 1012, MESSAG(12), AREA
1012 FORMAT (1H0, A8, 1X, 'AREA=', F6.1, ' SQUARE FEET')
PRINT 1013, MESSAG(13), EXDIAM
1013 FORMAT (1H0, A8, 1X, 'EXCAVATED DIAMETER INCLUDING OVERBREAK=', F5.1,
1F12.0, ' FEET')
PRINT 1014, MESSAG(14), LENGTH
1014 FORMAT (1H0, A8, 1X, 'LENGTH OF TUNNEL=', F6.1, ' FEET')
PRINT 1015, MESSAG(15), KL
1015 FORMAT (1H0, A8, 1X, 'REACH LENGTH=', F6.1, ' FEET')
PRINT 1016, MESSAG(16), SLOPE
1016 FORMAT (1H0, A8, 1X, 'SLOPE=', F4.2)
PRINT 1017, MESSAG(17), DRSTWF
1017 FORMAT (1H0, A8, 1X, 'DISTANCE FROM BASE OF SHAFT TO WORKING FACE=',
  F12.0)
1F4.1, 'MILES')
PRINT 108,MESG(18),DOS
1018 FORMAT(1H1,AB,1X,'DISTANCE TO DISPOSAL AREA=',F5.1, 'MILES')
PRINT 1019,MESG(19),LARO
1019 FORMAT(1H1,AB,1X,'LABOR COST INDEX=',F4.2)
PRINT 1020,MESG(20),EQUIP
1020 FORMAT(1H1,AB,1X,'EQUIPMENT COST INDEX=',F4.2)
PRINT 1021,MESG(21),MATLS
1021 FORMAT(1H1,AB,1X,'MATERIALS COST INDEX=',F4.2)
PRINT 1022,MESG(22),GWINF
1022 FORMAT(1H1,AB,1X,'INITIAL GROUND WATER INFLOW=',F6.1, 'GPM/MILE')
PRINT 1023,MESG(23),GWINF
1023 FORMAT(1H1,AB,1X,'SUSTAINED GROUND WATER INFLOW=',F6.1, 'GPM/MILE')
PRINT 1024,MESG(24),GWRPS
1024 FORMAT(1H1,AB,1X,'GROUND WATER PRESSURE=',F4.1, 'ATM')
PRINT 1025,MESG(25),STRNTH
1025 FORMAT(1H1,AB,1X,'ROCK STRENGTH=',F7.1, 'PSI')
PRINT 1026,MESG(26),ROD
1026 FORMAT(1H1,AB,1X,'ROCK QUALITY DESIGNATION=',F4.1, 'PER CENT')
PRINT 1027,MESG(27),BULKF
1027 FORMAT(1H1,AB,1X,'BULKING FACTOR=',F4.2)
PRINT 1028,MESG(28),OVBRSK(1)
1028 FORMAT(1H1,AB,1X,'CONVENTIONAL OVERBREAK=',F4.2)
PRINT 1029,MESG(29),OVBRSK(2)
1029 FORMAT(1H1,AB,1X,'MOLE OVERBREAK=',F4.2)
PRINT 1030,MESG(30),ADVANC(1)
1030 FORMAT(1H1,AB,1X,'CONVENTIONAL ADVANCE RATE=',F5.1, 'FEET PER DAY')
1030 PRINT 1031,MESG(31),ADVANC(2)
1031 FORMAT(1H1,AB,1X,'MOLE ADVANCE RATE=',F5.1, 'FEET PER DAY')
1031 PRINT 1032,MESG(32),OMUCK(1)
1032 FORMAT(1H1,AB,1X,'CONVENTIONAL QUANTITY OF MUCK=',F6.1, 'CUBIC FEET PER UNIT LENGTH')
1032 PRINT 1033,MESG(33),OMUCK(2)
1033 FORMAT(1H1,AB,1X,'MOLE QUANTITY OF MUCK=',F6.1, 'CUBIC FEET PER UNIT LENGTH')
PRINT 1634,MESAG(34),RML(1)
1634 FORMAT(1H4,A8,1X,'RATE USED TO SIZE MUCKING EQUIPMENT (CONVENTIONAL
1)=',F5.1,' CUBIC YARDS PER HOUR')
PRINT 1635,MESAG(35),RML(2)
1635 FORMAT(1H4,A8,1X,'RATE USED TO SIZE MUCKING EQUIPMENT (MOLE)=',
1F4.1,' CUBIC YARDS PER HOUR')
PRINT 1636,MESAG(36),HL
1636 FORMAT(1H4,A8,1X,'LENGTH OF DRILL HOLES FOR GROUTING=',F5.2,
1' FEET PER FOOT')
PRINT 1637,MESAG(37),GT
1637 FORMAT(1H4,A8,1X,'GROUT TAKE=',F4.2,' CUBIC FEET PER FOOT')
WRITE(INOUT,139) MESAG(39),TEMPR
139 FORMAT(1H4,A8,1X,'ROCK TEMPERATURE=',F5.1,' DEGREES F')
WRITE(INOUT,140) MESAG(40),TEMPO
140 FORMAT(1H4,A8,1X,'OUTDOOR TEMPERATURE=',F5.1,' DEGREES F')
RETURN
END
SUBROUTINE INPUT(LABEL, ITYPE, IDATA, RDATA, COUNT)
C----- IF TYPE=0, THEN THE ORIGINAL INPUT WAS IN INTEGER FORM
C----- IF TYPE=1 THEN THE ORIGINAL INPUT WAS IN FLOATING POINT.
C----- IF THE PROGRAM READS A ' ' THEN IT RETURNS LABEL=0.
C----- THE ARRAY CONTAINS BOTH FIXED AND FLOAT VALUES OF EACH NUMBER
C----- COUNT GIVES THE LENGTH OF THE ARRAY
REAL L8 LABEL
INTEGER I8 LETTER, CARD, AST, EQU, BLANK, DPT, NUMBER
INTEGER COUNT, CC, POS, DIGIT
REAL MULT
DIMENSION ITYPE(12), IDATA(12), RDATA(12), CARD(8), LETTER(8)
DIMENSION NUMBER(10)
EQUIVALENCE (ITYPE(12), IDATA(12), RDATA(12), CARD(8), LETTER(8))
DATA AST, EQU, BLANK, DPT, ' ', ' ', ' ', ' ', ' ', ' ', ' '/
DATA NUMBER/0,1,2,3,4,5,6,7,8,9/
POS=1
DO 10 I=1, 8
10 LETTER(I)=BLANK
COUNT=0
DO 12 I=1, 10
12 IDATA(I)=0
RDATA(I)=0
10 ITYPE(I)=0
15 CC=0
READ(5, 1) CC, END=9, ERR=15) CARD
1001 FORMAT (8X, 1)
PRINT 1001, CARD
1001 FORMAT (1H, 8X, 1)
20 CC=CC+1
IF (CC .GT. 8) GO TO 15
IF (CARD(CC) .EQ. AST) GO TO 901
IF (CARD(CC) .EQ. BLANK) GO TO 20
25 LETTER(POS)=CARD(CC)
CC=CC+1
POS=POS+1
IF (POS GT 8) GO TO 35
IF (CARD(CC) EQ BLANK) GO TO 36
IF (CARD(CC) EQ EQU) GO TO 30
GO TO 25
30 LETTER(POS) = BLANK
35 CONTINUE
C----- LOOK FOR FIRST NUMERIC CHARACTER
36 DO 42 CC = CC + 1
   IF (CARD(CC) EQ BLANK) GO TO 42
   DO 40 J = 1, 10
      IF (CARD(CC) EQ NUMBER(J)) GO TO 45
40 CONTINUE
42 CONTINUE
C----- RETURN IF NO NUMBER IS FOUND
GO TO 910
45 COUNT = COUNT + 1
INUMBR = 0
RNUMBR = 0.0
MULT = 1.0
50 INUMBR = INUMBR X 10 + J - 1
CC = CC + 1
IF (CC GT 80) GO TO 902
C----- BRANCH TO FLOATING POINT ROUTINE IF A '.' IS FOUND
IF (CARD(CC) EQ DPT) GO TO 60
IF (CARD(CC) EQ BLANK) GO TO 56
DO 55 J = 1, 10
   IF (CARD(CC) EQ NUMBER(J)) GO TO 56
55 CONTINUE
56 IDATA(COUNT) = INUMBR
RDATA(COUNT) = FLOAT(INUMBR)
GO TO 36
C----- FLOATING POINT ROUTINE
60 RNUMBR = FLOAT(INUMBR)
1 TYPE(COUNT) = 1
65 CC = CC + 1
MULT = MULT + 0.1

IF(CARD(CC).EQ.BLANK) GO TO 71
DO 71 J=1,10
IF(CARD(CC).EQ.NUMBER(J)) GO TO 75
70 CONTINUE
C------EXIT FROM FLOAT ROUTINE
71 RDATA(COUNT)=RNUMBR
I DATA(COUNT)=INUMBR
GO TO 36
75 RNUMBR=RNUMBR+MULT*FLOAT(J-1)
GO TO 65
C------END OF FILE EXIT (/) 0)
901 LABEL=991
. GO TO 999
C------END OF FILE EXIT (8)"
902 LABEL=992
GO TO 999
C------EXIT FROM FIXED ROUTINE
903 LABEL=993
GO TO 999
C------CALL LPACK(LABEL,LETTER)
999 RETURN
END
CALL LPACK(NAME, LETTER)

LPACK

START

BC 15.15(15)  
DC X'5'

DC CL5'LPACK'

STM 14.12.12(13)  
BALR 1(6)

USING 8.1(13)

L  2.5(1)  
L  3.4(1)

MVC 6(1.2), 0(3)
MVC 1(1.2), 2(3)
MVC 2(1.2), 4(3)
MVC 3(1.2), 6(3)
MVC 4(1.2), 8(3)
MVC 5(1.2), 10(3)
MVC 6(1.2), 12(3)
MVC 7(1.2), 14(3)

LM 2.12.28(13)  
MVI 12(13), X'FF'

RCR 15.14  

END
SUBROUTINE CONFLI

THIS PROGRAM ELIMINATES THOSE METHODS WHICH ARE INCOMPATIBLE WITH THE

PHYSICAL AND SPECIFIED PARAMETERS.

SUBROUTINE CONFLI

IMPLICIT REAL (L,M)

INTEGER MEXCAV, MSUPRT, ML ING, MGTAPP, MMUCKG, METH

INTEGER SHAPE, PRINT, SFTCOD, SHWWCH

COMMON/MMMM/MMEXCAV(12), NEXCAV, MSUPRT(12), NSUPRT, ML ING(12),

1NL ING, MGTAPP(12), MMUCKG(12), NMUCKG

COMMON/SHAPE, SFTCOD, LTHICK, DNLIFE, DEPTH, DIAM, AREA, EXDIAM,

LENGTH, L, SLOPE, DBSTWF, DDO, LABOKI, EQUIPI, MATLIS, GWINFO, GWINFO,

2GRWPRS, STRNTH, ROD, RULKF, OVRBRK(2), ADVANC(2), QMUCK(2), RML(2), HL, GT,

3OUT, TEMPR, TEMPD

COMMON/CONTROL/IOUT, SHWWCH

PRINT 5000

5000 FORMAT(1H1,T66,'METHODS TO BE TRIED',:///)

C---THIS SECTION OF THE PROGRAM ELIMINATES UNFEASABLE METHODS

C---EXCAVATION

IF((GWINFS>GT.10000.) AND (GRWPRS>GT.3)) GO TO 90

C---IF GROUND WATER INFLOW >10000 AND PRESSURE >3 USE COMPRESSED AIR

GO TO 99

99 DO 11 I=1,NEXCAV

IF((MEXCAV(I)>EQ.60)) CALL DELETE(MEXCAV,NEXCAV,I)

CONTINUE

11 IF((STRNTH>GT.20000.) OR (GWINFS>GT.10000.)) GO TO 100

C---IF STRENGTH OF ROCK >20000 OR RATE OF GROUND WATER INFLOW>10000

C---ELIMINATE MECHANICAL MOLE.

GO TO 120

100 DO 110 I=1,NEXCAV

IF(((MEXCAV(I)/10)>EQ.5.) OR (MEXCAV(I)>EQ.95)) CALL DELETE(MEXCAV,

1NFXCAV,I)

CONTINUE

120 IF (ROD-EQ.25) 150,150,130

C---IF ROD > 25% SHIELD SHOULD NOT BE USED
130 DO 14 I=1,NEXCAV
   IF (((MEXCAV(I),EQ,7) .OR. (MEXCAV(I),EQ,8))) CALL DELETE(MEXCAV,NEXCAV,I)
14    CONTINUE
150 IF (SHAPE,1) 180,180,160
C------IF SHAPE IS NOT CIRCULAR MOLES AND SHIELDS CAN NOT BE USED
160 DO 175 I=1,NEXCAV
   IF (MEXCAV(I),GE,50) CALL DELETE(MEXCAV,NEXCAV,I)
175 CONTINUE
180 IF ((DIAM-4),210,210,190
C------IF DIAMETER > 4 FEET MOLE CAN NOT BE USED
190 DO 295 I=1,NEXCAV
   IF (((MEXCAV(I)/1),EQ,51) .OR. (MEXCAV(I),EQ,80)) CALL DELETE(MEXCAV,NEXCAV,I)
295 CONTINUE
210 IF ((DIAM-1),220,215,215
C------IF DIAMETER < 1 FEET HEADING AND BENCH AND MULTIPLE DRIFT ARE OUT
215 IF (DIAM-15),230,240,240
C------IF DIAMETER < 15 FEET MULTIPLE DRIFT IS OUT
220 DO 225 I=1,NEXCAV
   IF (MEXCAV(I),EQ,30) CALL DELETE(MEXCAV,NEXCAV,I)
225 CONTINUE
230 DO 235 I=1,NEXCAV
   IF (MEXCAV(I),EQ,40) CALL DELETE(MEXCAV,NEXCAV,I)
235 CONTINUE
240 PRINT 5001, NEXCAV,(MEXCAV(N),N=1,NEXCAV)
5001 FORMAT(12, EXCAVATION METHODS ARE',T42,1214)
C
C------SUPPORT
   IF (ROD,<95) 250,250,245
C------IF ROD > 95% NO SUPPORTS ARE NEEDED
245 MSUPRT(1)=0
   NSUPRT=1
   GO TO 270
250 IF (ROD,>60) 260,260,255
C------IF ROD > 60% DO NOT USE STEEL SUPPORTS OR LINER PLATES
DO 256 I=1,NSUPRT
MTEST=MSUPRT(I)/1
IF((MTEST.EQ.2).OR.(MTEST.EQ.3)) CALL DELETE(MSUPRT,NSUPRT,I)
CONTINUE
GO TO 260
260 IF (ROD<40) 265,265,267
C-----IF ROD<40% USE STEEL SETS WITH BLOCKING AND LAGGING OR LINER PLATES
C (OR SHOTCRETE AND ROCKBOLTS)
DO 266 I=1,NSUPRT
MTEST=MSUPRT(I)/1
IF((MTEST.EQ.2).OR.(MTEST.EQ.3).OR.(MSUPRT(I).NE.51)) CALL DELETE(MSUPRT,NSUPRT,I)
CONTINUE
267 IF (ROD<.25) 1267,11267,11267
C-----IF ROD<.25% USE STEEL SETS WITH BLOCKING AND LAGGING OR LINER PLATES
DO 1268 I=1,NSUPRT
MTEST=MSUPRT(I)
IF((MTEST.EQ.2).OR.(MTEST.EQ.3)) CALL DELETE(MSUPRT,NSUPRT,I)
CONTINUE
11267 IF (DIAM<30.) 273,270,268
DO 268 I=1,NSUPRT
IF(MSUPRT(I)/10.EQ.3) CALL DELETE(MSUPRT,NSUPRT,I)
CONTINUE
270 PRINT 5602,NSUPRT,(MSUPRT(N),N=1,NSUPRT)
5602 FORMAT('THE',I2,' SUPPORT METHODS ARE',T42,12I4)
C
C-----LINING
IF((ROD<.5).OR.(GWINFL.GT.100000)) GO TO 275
C-----IF ROD<.5% OR GROWING WATER INFLOW > 100000 ELIMINATE NONE
GO TO 280
275 DO 276 I=1,MLNING
IF (MLNING(I).EQ.0) CALL DELETE (MLNING,NLNING,I)
CONTINUE
GO TO 278
278 IF (DNLIFE<5) 285,285,290
C-----IF DESIGN LIFE < 5 YEARS, USE NO LINING
285  MLNING(I)=0,
     NLNING=1
     GO TO 299
C-----IF ROD > 9.9% ELIMINATE STRUCTURAL LINING
295  DO 296 I=1,NLNING
     IF((MLNING(I)-(MLNING(I)/10)**10).EQ.1) CALL DELETE (MLNING,NLNING,
     I)
296  CONTINUE
299  PRINT 5003.  NLNING,(MLNING(I),I=1,NLNING)
5003  FORMAT('THE',12,' LINING METHODS ARE',T42,12I4)
C
C-------GROUTING AND PUMPING
C--------SET INDICATOR
     INDIC=1
C--------CALCULATE TOLERABLE LEAKAGES
C--------INITIAL TOLERABLE IS GWTOLI IN Gp0.0/MILE
     GWTOLI=3000.
C--------SUSTAINED TOLERABLE LEAKAGE IN Gp0.0/MILE
     GWTOLI=EXP(11.85-ALOG(DEPTH))-20.
     IF((GWINFS,LT,1.).AND.(SFCD3,LT,NE,2)) GO TO 303
C--------PUMPING MUST BE USED IF TUNNEL IS STARTED FROM SHAFT
     DO 310 I=1,NGTAPP
         IF(MGTAPP(I).EQ.0) CALL DELETE(MGTAPP,NGTAPP,I)
310  CONTINUE
     GO TO 313
313  IF((ROD,GT,8.9),OR,(GWINFS,LT,100.)) GO TO 304
C--------IF ROD > 89% OR GROUND WATER INFLOW < 100 GPM, USE NO GROUT
     GO TO 310
314  DO 315 I=1,NGTAPP
         IF((MGTAPP(I)-(MGTAPP(I)/10)**10).NE.0) CALL DELETE (MGTAPP,NGTAPP,
         I)
315  CONTINUE
316  IF((GWINFI,GWTOLI)) 315,315,312
C--------IF INITIAL INFLOW IS TO GREAT USE PREGROUTING
312  DO 313 I=1,NGTAPP
IF((MGTAPP(I)/100).NE.2) CALL DELETE(MGTAPP,NGTAPP,I)
CONTINUE
INDIC=2
IF(GWINS-GWTOLS) STOP,STOP,STOP
C-----IF SUSTAINED INFLOW IS TO RIG, USE POST GROUTING
DO 317 I=1,NGTAPP
IF((MGTAPP(I)/100).NE.1) CALL DELETE(MGTAPP,NGTAPP,I)
CONTINUE
INDIC=3
IPUMP=1
IF(GRWP5-3.) STOP,STOP,STOP
C-----IF GROUND WATER PRESSURE > 3 ATM PUMPS ARE REQUIRED
DO 322 I=1,NGTAPP
IF((MGTAPP(I)-(MGTAPP(I)/10)).EQ.0) CALL DELETE(MGTAPP,NGTAPP,I)
CONTINUE
IPUMP=2
IF(NGTAPP) STOP,STOP,STOP
C-----IF WE HAVE ELIMINATED EVERYTHING WE MUST MAKE ASSUMPTIONS
GO TO (325,328,331),INDIC
NGTAPP=1
IF(SFTCDE.EQ.2) GO TO 332
GO TO (326,327),IPUMP
MGTAPP(1)=001
GO TO 334
MGTAPP(1)=011
GO TO 334
NGTAPP=2
GO TO (329,332),IPUMP
MGTAPP(I)=211
MGTAPP(?)=221
GO TO 334
MGTAPP(I)=211
MGTAPP(?)=221
GO TO 334
NGTAPP=2
GO TO (332,333), IPUMP
332 MGTAPP(1)=110
MGTAPP(2)=120
GO TO 334
333 MGTAPP(1)=111
MGTAPP(2)=121
334 PRINT 5004, NGTAPP,(MGTAPP(I),I=1,NGTAPP)
5004 FORMAT ('THE',I2,' GRouting AND PumpING METHODS ARE',T42,1214)

C------Mucking
340 IF (DIAM-20) 343,350,350
343 DO 344 I=1,NMUCKG
     IF (MMUCKG(I)/10),EQ,2) CALL DELETE (MMUCKG,NMUCKG,I)
344 CONTINUE
350 PRINT 5005, NMUCKG,(MMUCKG(I),I=1,NMUCKG)
5005 FORMAT ('THE',I2,' Mucking METHODS ARE',T42,1214)
999 RETURN
END
SUBROUTINE DELETE(ARRAY, NUMBER, ELEMNT)
C------THIS SUBROUTINE DELETES ELEMENT 'ELEMNT' FROM AN INTEGER ARRAY OF
C------DIMENSION 'NUMBER'. THE DIMENSION IS THEN DECREMENTED.
INTEGER ARRAY, ELEMNT, CHRIS
DIMENSION ARRAY(12)
CHRIS=NUMBER-1
DO 1 I=ELEMNT, CHRIS
  1 ARRAY(I)=ARRAY(I+1)
ARRAY(NUMBER)=0
NUMBER=CHRIS
RETURN
END
SUBROUTINE COMBO

SECTION WHICH PRODUCES COMPATIBLE COMBINATIONS OF METHODS AND

COMPUTES THE COST OF EACH METHOD

IMPLICIT REAL (L,M)
INTEGER SHAPE,PRINT,SFTCDE,SHSWCH
INTEGER MEXCAV,MSUPRT,MLNING,MGTAPP,MMUCKG,METH
COMMON/MM/MM/MM/MEXCAV(12),NEXCAV,MSUPRT(12),NSUPRT,MLNING(12),
INLING,MGTPP(12),MGTPP,MMUCKG(12),NMUCKG
COMMON/DATA/SHAPE,SFTCDE,LTHICK,DLNIFE,DEPTH,DIAM,AREA,EXDIAM,
1 LENGTH,RL,SLOPE,DRSTWF,DDS,LABORI,EQUIPI,MATLSI,GWINFI,GWINFS,
2GRWPRS,STRNTH,ROD,BULKF,OVBRK(2),ADVAN(2),OMUX(2),RML(2),HL,GT,
3OUT,TMPR,TEMPD
COMMON/CTRL/10UT,SHSWCH

IFXC=ESCAVATION
ISUP=SUPPORT
ILIN=LINING
IGRD=GRATING AND PUMPING
TMUC=MUCKING

IF (SHSWCH.EQ.2) NMUCKG=1

5100 FORMAT(1H1,53X,"PRICES OF VARIOUS METHODS*/
1T46,",'ALL COSTS ARE IN DOLLARS PER LINEAR FOOT'*/,
IIF (OUT.EQ.2) AND (SHSWCH.EQ.2) WRITE (10UT,5101)

5101 FORMAT(62X,'FOR SHAFTS')
TCOUNT=1
DO 5 IDX5=1,NEXCAV
IFXC=MEXCAV(IDX5)
DO 4 IDX4=1,NSUPRT
ISUP=MSUPRT(IDX4)
DO 3 IDX3=1,MLNING
IF ((ISUP.EQ.0) .AND. (DLNIFE.GE.5) .AND. (MLNING(IDX3).EQ.0))
GO TO 3
IF ((DLNIFE.GE.15) .AND. (MLNING(IDX3).EQ.0)) GO TO 3
ILIN=MINING(IDX3)
IF(((ISUP/LI).EQ.3) ILIN=0
DO 2 IDX2=1,NGTAPP
IF(((IEXC/LI).EQ.5).OR.(IEXC.EQ.86)).AND.((MTAPP(IDX2)/190).EQ.2
1)) GO TO 2
IGRO=MGTAPP(IDX2)
DO 1 IDX1=1,NMUCKG
IMUC=MUACKG(IDX1)
ICOUNT=ICOUNT+1
1 CONTINUE
GO TO (10,11),SHSWCH
10 CALL COSTS(IEXC,ISUP,ILIN,IGRO,IMUC)
GO TO 1
11 CALL COSTSH(IEXC,ISUP,ILIN,IGRO)
C
1 CONTINUE
2 CONTINUE
3 CONTINUE
4 CONTINUE
5 CONTINUE
PRINT 5200,ICOUNT
5200 FORMAT(1H1,T4).THE RESULTS OF THE MINIMUM COSTS OF',14,
1. COMBinations',/T46.'ALL COSTS ARE IN DOLLARS PER LINEAR FOOT',
2//' IF(SHSWCH.EQ.2) WRITE(IOUT,5101)
RETURN
END
SUBROUTINE COSTS

This program computes the actual costs per linear foot of a given tunnel.

SUBROUTINE COSTS(IEXC, ISUP, ILIN, IGRO, IMUC)
IMPLICIT REAL (1,M)
INTEGER SHAPE, PRINT, SFTCDE, ORANGE, SHSWCH
INTEGER MEXCAV, MSUPRT, MLNING, MGTTAPP, MMUCKG, METH
DIMENSION FVCOST(4), VVCOST(4), CDIAM(4), EOTIME(2), LBTIME(2)
COMMON/MMPMAM/MEXCAV(12), NEXCAV, MSUPRT(12), NSUPRT, MLNING(12),
1NLNING, MGTTAPP(12), MGTTAPP, MMUCKG(12), MMUCKG
COMMON/DATA/SHAPE, SFTCDE, LTHICK, DNLIFE, DEPTH, DIAM, AREA, EXDIAM,
1LENGTH, RL, SLOPE, DBSTWF, DDS, LABOR, EQUIP, MATSL, GWINF, GWINFS,
2GRWP, SRTNTH, POD, BULKF, OVRBRK(2), ADVANC(2), OMUCK(2), RML(2), HL, GT,
3OUT, TEMPR, TEMPE
COMMON/CTRL/IOUT, SHSWCH
COMMON/INPUT/STORE, METH, PRICE

--- ROCKBOLT METHOD FACTOR
REAL RBMF(2) / 8.56, 4.81 /

DO 100 I=1,2
LBTIME(I) = (24 + 0.99*DBSTWF) / ADVANC(I)
100 CONTINUE

EOTIME(I) = 24 / ADVANC(I)
IF (IEXC EQ 1) I = 1
IF (IEXC EQ 5) I = 2
RS = STRNTH
IF ((SHAPE EQ 1) OR (SHAPE EQ 1)) Z = 5*(7.0 - DIAM)**2
IF(SHAPE EQ 3) Z = 4.5 - (DIAM - 1)**4/4400.

--- COMPUTE VOLUME IN CUBIC YARDS
VOLUME = AREA/27.

SUFFIX L AND L8K SIGNIFY LABOR COST
SUFFIX M AND MAT SIGNIFY MATERIALS COST
SUFFIX E AND EQP SIGNIFY EQUIPMENT COST
SUFFIX T AND TOT SIGNIFY TOTAL COST

C-----IF NOT MINE GO TO CONVENTIONAL
IF(IEXCNEF.50) GO TO 110

C-----SETUP (SETUP FOR CONVENTIONAL = 0)
C LABOR
SETUPL=(5000.0+15.*DEPTH+152.*(DIAM+12.))**2/LENGTH
C EQUIPMENT
SETUPE=(25.*DEPTH+6.*DEPTH+234.*(DIAM-10.)**2)/LENGTH
C MATERIALS
SETUPM=(25.*DIAM**2)/LENGTH
C TOTAL
SETUPT=SETUPL*LABORI+SETUPE*EQUIPI+SETUPM*MATLSI

C-----EXCAVATION
C LABOR
EXCSTL =(.021*DIAM**2+55.)*LBTIME(2)
C EQUIPMENT
EXCSTE =(.048*DIAM**2+24.)/ADVANC(2)
C MATERIALS
EXCSTM =((5000.0+RS+120000.)/ADVANC(2))**2*DIAM**2/220004.0
C TOTAL
EXCSTT=EXCSTL*LABORI+EXCSTE*EQUIPI+EXCSTM*MATLSI
EXTOT=SETUPT+EXCSTT
GO TO 200

C-----IF NOT CONVENTIONAL GO TO ERROR TRAP
110 IF(IEXCNEF.1) GO TO 9999

C-----COMPUTE COSTS FOR CONVENTIONAL EXCAVATION
GO TO (120,130,140), SHAPE

C-----CIRCULAR TUNNEL COST COMPONENT IN $/HR
C LABOR
120 CSTCPL =.07*(DIAM+4.0)**2-100.
C EQUIPMENT
CSTCPF =.046*(DIAM+15.)**2
C MATERIALS
CSTCPM=0.785*SORT(RS)*DIAM**2/2000+(0.11*(DIAM+10)*2-25)/ADVAN
1C(1)
GO TO 150

C-------HORSEHOE
130 CSTCPL=0.08*(DIAM+40)**2-110
CSTCP=0.5*(DIAM+15)**2+4
CSTCPM=0.893*SORT(RS)*DIAM**2/2000+(0.12*(DIAM+10)**2-22)/ADVAN
1C(1)
GO TO 150

C-------BASKETHANDLE
140 CSTCPL=0.13*DIAM**2+60
CSTCP=0.14*(DIAM+5)**2+27
CSTCPM=0.425*SORT(RS)*DIAM**2/2000+(0.13*(DIAM-5)**2+14)/ADVANC
1C(1)

C-------COMPUTE EXCAVATION COST
150 EXCSTL=CSTCP+LABTIME(1)*LABORI
EXCSTE=CSTCP-EQTIME(1)*EQUIPI
EXCSTM=CSTCPM+MATLSI
EXCSTT=EXCSTL+EXCSTE+EXCSTM

C-------MUCK LOADING (MUCKLOADING INCLUDED IN MULE COST)
C LABOR
MUCKLL=9.30*BTIME(1)/ADVANC(1)
C EQUIPMENT
IF ( RML(1),LE,100 ) CC=3.50
IF ( ( RML(1),GT,100 ) AND ( RML(1),LE,300 ) ) CC=6.60
IF ( ( RML(1),GT,300 ) ) CC=9.90
MUCKLE=CC*24/ADVANC(1)
C MATERIALS
MUCKLM=0.28*VOLUME
MUCKLT=MUCKLL+MUCKLE+EQIP+MUCKLM+MATLSI
C
EXTOT=MUCKLT+EXCSTT
C
C-----MUCKING

C

C-----MUCK TRANSPORT

C

C-----CONVEYOR (MMUCKG=32)
200 IF(IMUCANE=32) GO TO 230
MKTRNL =((15+13*ADVANC(I)+206*DBSTWF)*SORT(RML(I)+40)+0.06*ADVANC(I)-1.0*DBSTWF)*LBTIME(I)*LABORI
IF (SLOPE) 210:211:220
210 MKTRNE =((SORT(RML(I)+40)+60*SLOPE-1.5)/5280)*EOTIME(I)*
(LENGTH/2)*EQUIPI
MKTRNM =((81+1.23*SLOPE)*VOLUME*DBSTWF*MATLSI
GO TO 250
220 MKTRNE =((1.0+40*SLOPE)*SORT(RML(I)+40)-260*SLOPE-1.5)/5280)
1.5*EOTIME(I).getBean(LENGTH/2)*EQUIPI
MKTRNM =((21)*VOLUME*DBSTWF*MATLSI
GO TO 250
C

C-----RAIL (MMUCKG=12)
230 IF(IMUCANE=12) GO TO 240
X=ALOG(DBSTWF)
Y=1.2*EXP(X/30)
CARS=((1.6+Y)*RML(I)+ADVANC(I)*(0.6+X*23*DBSTWF-Y)*ADVANC(I)/300
1.5
CARS=AIMT(CARS+0.5)
ENGINS=CARS/1.5
ENGINS=AIMT(ENGINS+1.5)
IF (ENGINS<1.0) ENGINS=1.0
MKTRNL =((5.0+ENGINS+21.0*EXP(0.82*X)+ADVANC(I)/12)*LBTIME(I)*LABORI
1.5
MKTRNE =((5.8+ENGINS+1.4+DBSTWF)*EOTIME(I)*EQUIPI
MKTRNM =((0.3)*VOLUME*DBSTWF*MATLSI
GO TO 250
C

C-----TRUCK
240 IF(IMUCANE=22) GO TO 9999
CARS = \( \{ \sqrt{6} : \exp(5.7x) \} / z \)
CARS = AINT(CARS + 15)
IF (CARS LT 10) CARS = 10
MKTRNL = (7.7 * CARS + 7.1) * LBTIME(I) * LABOR
MKTRNE = (1.5 + 3.1 * Z) * CARS * EOTIME(I) * EQUIP
MKTRNM = (0.55 + 0.77 * ARS(SLOPE)) * VOLUME * DBSTWF * MATLSI

250 MKTRNT = MKTRNL + MKTRNE + MKTRNM

C
C MUCK HOISTING
C INSTALLATION
C LABOR
MKHSTI = ((500000 + 0.5 * (9000 + DEPTH)) * RML(I) / 27.9) / RL
MKHSTL = (350 + 0.3 * DEPTH * (1000 + 0.2 * RML(I)) + 0.16 * RML(I)) * EOTIME(1) * LABOR
C EQUIPMENT
MKHSTF = (150 + 0.33 * RML(I)) * EOTIME(I) * EQUIP
C MATERIALS
MKHSTM = (5 * VOLUME * DEPTH / 10000) * MATLSI
C TOTAL HOISTING COST
MKHSTT = MKHSTI + MKHSTL + MKHSTF + MKHSTM

C
C MUCK DISPOSAL
C LABOR
MKDSPPL = (10 + (15.5 + 4.7 * DDS) * VOLUME * ADVANC(I) / 10000) * EOTIME(I) * LABOR
MKDSPPE = (60 + (12.5 + 2.1 * DDS) * VOLUME * ADVANC(I) / 10000) * EOTIME(I) * EQUIP
C MATERIALS
MKDSPM = 0.55 * DDS * VOLUME * MATLSI
C TOTAL
MKDSP = MKDSPPL + MKDSPPE + MKDSPM
MUCTOT=MKTNT+MKHSST+MKDSPT

C

C

C-----SUPPORTS
C-----ROCKBOLT
C
IF (ISUPNE.41) GO TO 325
C
SHAPE FACTOR SF=1.0 FOR CIRCULAR AND HORSHOE, SF=1.6 FOR BASKETHANDLE
SF=1.0
IF (SHAPE.EQ.3) SF=1.6
C
COMPUTE WEIGHT OF ROCKBOLTS IN POUNDS PER LINEAR FOOT
WPR=(RBMF(I)*DIAM*SF*EXP(0.026*DIAM)*(100.0-RQD)**2)/(152.0-RQD)**2
C
LABOR
SUPPL=(.02*WPR*ADVANC(I)) * LBTIME(I) * LABOR
C
EQUIPMENT
IF (DIAM.GT.16.0) GO TO 316
CC=1.76
GO TO 320
316 TEMP=WPR*ADVANC(I)-1000
IF (TEMP) 315, 315, 316
315 CC=9.35
GO TO 320
316 CC=9.35+9.03*2*TEMP
320 SUPPF=CC+EOTIME(I) * EQUIPI
C
MATERIALS
C
WEIGHT OF WIRE MESH AND STRAPS
SF=1.0
IF (SHAPE.EQ.3) SF=.74
WWM=5.3*DIAM*SF*(100.0-RQD/100.0)
SUPPM=(.032*WWM+(.23+RQ/530000)*WPR)*MATERIAL
GO TO 360
C
325 IF (ISUPNE.20) GO TO 370
C-----STEEL SFTS
C
QUANTITIES
GO TO (330, 340), I
C CONVENTIONAL
330 YY=1.0
GO TO (331, 332, 333), SHAPE
C CIRCULAR
331 WST=EXP(SORT(0.7*DIAM-6.0)+(3.7-2)*((ROD-2.0)/10.0)**2)
BL=1.056*DIAM+1.056*DIAM*((117.0-ROD)/10.0)**2
GO TO 350
C HORSHOE
332 WST=EXP(SORT(0.85*DIAM-7.0)+0.8+0.3*SORT(100.0-ROD))
BL=1.06+1.012*DIAM+(0.018*DIAM*((117.0-ROD)/100.0)**2)
GO TO 350
C BASKETHANDLE
333 WST=EXP(SORT(0.92*DIAM-6.0)+0.82+0.22*SORT(100.0-ROD))
BL=1.055+1.033*DIAM+(0.0615*DIAM+0.045*((117.0-ROD)/100.0)**2)
GO TO 350
C MOLE (CIRCULAR OF COURSE)
340 YY=1.0
WST=EXP(SORT(0.7*DIAM-6.0)+(3.7-2)*((ROD+2.0)/10.0)**2)
BL=1.056*DIAM+1.056*DIAM*((117.0-ROD)/10.0)**2
C COSTS
C LABOR
350 SUPPL=YY*(0.001*WST+ADVANC(I)+0.4*DIAM)*LTIME(I)*LABOR
C EQUIPMENT
SUPPE=YY*(6.0+0.82)*WST+ADVANC(I)*EQTIME(I)*EQUIPI
C MATERIALS
SUPPM=(36.0*YY/ADVANC(I)+0.096*WST+15.0)*BL*MATLSI
C TOTAL
360 SUPST=SUPPL+SUPPE+SUPPM
GO TO 380
370 IF(ISUPPE>0) GO TO 9999
SUPST=9999
380 SUPTOT=SUPST
C C------LINING
VOLUME OF LINING
VL = \pi \cdot 1416 \cdot ((EXDIAM/2a) + 2 - ((DIAM-LTHICK/6a)/2a) \cdot 2) / 27a
IF(ILIN\_NE=20) GO TO 410

SHOTCRETE
LINPL = (5.5 \cdot (VL + 3a) + 2 + 10a) \cdot LBTIME(I) \cdot LABORI
LINPE = (0.625 \cdot (VL + 8a) + 2 - 23a) \cdot EOTIME(I) \cdot EQUIP
LINPM = 14a \cdot VL \cdot MAT\_LSI
LINFT = 0.6

TRANSPORTATION PARAMETERS
CARS = (0.84 \cdot DBSTWF + 3a) \cdot VL
TR = (0.56 \cdot DBSTWF + 0.95) \cdot VL / 2
GO TO 415

CONCRETE
410 IF (ILIN\_NF=10) GO TO 460
LINPL = (5.5 \cdot (VL + 26a) + 2 - 12a) \cdot LBTIME(I) \cdot LABORI
LINPE = (5a + 1.9 \cdot VL) \cdot EOTIME(I) \cdot EQUIP
LINPM = 13a \cdot VL \cdot MAT\_LSI
LINFT = 0.6

TRANSPORTATION PARAMETERS
CARS = (0.42 \cdot DBSTWF + 1.5) \cdot VL
TR = (0.28 \cdot DBSTWF + 0.48) \cdot VL / 2

LINING FORMWORK (FOR CONCRETE)
LABOR
LINFL = (9a \cdot (DIAM-LTHICK/12a)) \cdot LBTIME(I) \cdot LABORI
EQUIPMENT
BF = DIAM-LTHICK/12a
IF (SHAPE=EQ.1) CC = 15 + 2a \cdot BF
IF (SHAPE=EQ.2) CC = 15a + 2a \cdot BF
IF (SHAPE=EQ.3) CC = 15a + 1.6 \cdot BF
LINFE = CC \cdot EOTIME(I) \cdot EQUIP
MATERIALS
LINFM = 0.18 \cdot BF \cdot MAT\_LSI
TOTAL
LINFT = LINFL + LINFE + LINFM

415 LINPT = LINPL + LINPE + LINPM
TRANSPORTATION OF LINING MATERIALS

CARS = AINT (CARS + 0.5)
TRANS = CARS / 1.5
TRANS = AINT (TRANS + 0.5)
IF (TRANS > EO) TRANS = 1
TR = AINT (TR + 0.5)
IF (TR > EO) TR = 1
IF (IMUC = 3) GO TO 430

CONVEYOR
LINL = [(0.6 * SORT(RML(I) + 0.1) - 1.1) * DBST * LBTIME(I)] * LABOR

C. ASSUMING THAT THE LINING ADVANCE RATE IS EQUAL TO THE TUNNELING ADVANCE RATE, THE TWO EQUIPMENT COSTS WILL BE THE SAME
LINTE = MKTRNE
LINTM = 0
GO TO 450

RAIL
LINL = (15 * TRAINS + 21 * DBST * EO/82) * LBTIME(I) * LABOR
LINTE = (5 * TRAINS + CARS + 1.4 * DBST) * EQTIME(I) * EQUIP
LINTM = 0
GO TO 450

IF (IMUC = 12) GO TO 435

C. TRUCK
LINL = (7.7 * TR + 7.1) * LBTIME(I) * LABOR
LINTE = (1.5 + 3.17) * TR * EQTIME(I) * EQUIP
LINTM = 0

C. TOTAL
LINT = LINL + LINTE + LINTM
LINTOT = LINP + LINT + LINFT
GO TO 500

IF (LINT = 0) GO TO 9999
LINTOT = 0
C-----GROUTING

500  GRRT=0.0
    GRTE=0.0
    GRTM=0.0
    IF((IGRT/10).EQ.0) GO TO 550
    GRTE=(2.3-IGRT/10)*(DIAM-46.6)**2*EQTME(I)*EQUIP
    GRTM=(1.25*GRTE*(RS*SHL/20000)+0.75*SHL*(DIAM-46.6)**2-100)*
    1MATLSI/ADVANC(I)
    GRRT=GRTE+GRTM

C

C-----PUMPING

550  PUMT=15.0
    IF((IGRT-(IGRT/10)*10).EQ.0) GO TO 599
    PUMPL=(5.5+11*DEPTH/3000+16*DBSTWF)*EQTME(I)*LABOR
    FLOWRT=1.0
    IF(GWINFS.LT.100.0) FLOWRT=GWINFS/100.0
    PUMP=((1000.*DEPTH)*FLOWRT/1000000)*EQTME(I)*EQUIP
    PUMP=5*(DEPTH*FLOWRT/200000)*EQTME(I)*MATLSI
    PUMPT=PUMPL+PUMP+PUMP

599  GROTT=GRRT+PUMPT

C

C-----AIR CONDITIONING

C VENTILATING

ACVL=3.0*DBSTWF*LATME(I)*LABOR
    IF(SHAPE.EQ.1) SF=0.785
    IF(SHAPE.EQ.2) SF=0.893
    IF(SHAPE.EQ.3) SF=0.425
    ACVE=(2.5*6.25*SF*(DIAM**2/1000000))EQTME(I)*EQUIP
    ACVM=(3.5*SF*(DIAM**2/200000)+(3.5*SF*DBSTWF*(DIAM**2/1000000)))*EQTME(I)*
    1MATLSI
    ACVT=ACVL+ACVE+ACVM
OT = AREA * (15 * DEPTH) / (TEMPD + 46) + O.3 * RL + 54 * (TEMPD - 40) * 90 + 1.25 * SQRT(AREA) * (2 * 35 * TEMPR - 28)

C------IF THERE IS NO HEAT GAIN, NO COOLING IS REQUIRED
IF(QT) 610, 612
610 ACCT=ACCT + GO TO 615
612 ACCL=((QT/48)/RL+8)*EQTIME(I)"LABORI
ACCE=ACCT + E-7*EQTIME(I)"EQUIP
ACCM=ACCL + E-7*EQTIME(I)"MATERIALS
ACCT=ACCL + ACCE + ACCM
615 ACTOT=ACVT + ACCT
C C
C------FINAL TOTAL
TOTAL=EXTOT + MUCTOT + SUPTOT + LINTOT + GROTOT + ACTOT
IF(QT.GE.200) WRITE(IOUT,2501) IEXC,ISUP,ILIN,IGRO,IMUC,EXTOT,
1 SUPTOT,LINTOT,GROTOT,MUCTOT,ACTOT,TOTAL
2 FORMAT('1 FOR EXCAVATION:',13,' SUPPORT:',13,' LINING:',13,
1 ' GROUTING AND PUMPING:',14,' MUCKING:',13,'/
2 THE COSTS ARE: EXCAVATION:',F7.2,,' SUPPORT:',F7.2,,' LINING:',
3 F7.2, ' GROUTING AND PUMPING:',F7.2, ' MUCKING:',F7.2,,'/
4 AIR CONDITIONING:',F7.2,,' TOTAL:',F7.2,'/
IF(TOTAL.GE.STORE) GO TO 999
C------IF WE HAVE A NEW MINIMUM, STORE THE TOTAL COST AND THE METHOD CODES
STORE=TOTAL
METH(1)=IEXC
METH(2)=ISUP
METH(3)=ILIN
METH(4)=IGRO
METH(5)=IMUC
C------ALSO STORE COMPONENT COSTS
PRICE(1)=EXTOT
PRICE(2)=SUPTOT
PRICE(3)=LINTOT
PRICE(4)=GROTOT
PRICE(5)=MUCTOT
PRICE(6)=ACTOT
GO TO 999

C---ERROR TRAP
9999 WRITE(IOUT,5000) IEXC,ISUP,ILIN,IGRO,IMUC
5000 FORMAT(1X,INVALID COMBINATION:,514)
999 RETURN
END
C SUBROUTINE COSTSH(IEXC,ISUP,ILIN,IGRO)
C------THIS PROGRAM COMPUTES THE ACTUAL COST IN DOLLARS PER LINEAR FOOT OF A GIVEN SHAFT
SUBROUTINE COSTSH(IEXC,ISUP,ILIN,IGRO)
IMPLICIT REAL(L,M)
INTEGER METH,SHAPE,SFTCDE,SHSWCH
COMMON/DATA/SHAPE,SFTCDE,LTHICK,ONLIFE,DEPTH,DIAM,AREA,EXDIAM,
1. LENGTH,RL,SLP,MAG,DDS,LABOR,EQUIPI,MATLSI,GWINFI,GWINFS,
2. GPWPRS,STRNTH,ROD,BULKF,OVBRK(2),ADVANC(2),QMUCK(2),RML(2),HL,GT,
3. OUT,TEMPR,TEMPD
COMMON/CONTL/OUT,SHSWCH
COMMON/MINIM/STORE,METH(5),PRICE(6)
C-----ROCKBOLT METHOD FACTOR
REAL RBMF(2)/8.56,4.81/
RS=STRNTH
C-----COMPUTE VOLUME IN CUBIC YARDS
VOLUME=AREA/27*
C
C SUFFIX L AND LBR SIGNIFY LABOR COST
SUFFIX E AND EOP SIGNIFY EQUIPMENT COST
SUFFIX M AND MAT SIGNIFY MATERIALS COST
SUFFIX T AND TOT SIGNIFY TOTAL COST
C
I=1
IF(IEXC.GT.45) I=2
EO TIME=24./ADVANC(I)
GO TO (100,200),1
C
C------CONVENTIONAL EXCAVATION INCLUDING MUCK LOADING
100 EXCSTL=(3.*a+4.*25.*EXDIAM)*EO TIME*LABOR
EXCSTE=(8.*a+2.*25.*EXDIAM)*EO TIME*EQUIP
EXCSTM=(1.*a+RS/2.*OUT(1.))*EXDIAM*2+5.*a+0.9+RS/4000.*3.4/ADVANC(1)
111:EXDIAM=MATLSI
EXCST=T=EXCSTL+EXCSTE+EXCSTM
MUCK=0.33*EO TIME*LABOR
MC K I_ T I ME, F Q U I P I M U I C K LM= 5' V OL Ut .?MATL SI MUC KLT = Mi IC K L+ MUCK LM EXC ST= EXC STT+MUCK LT
GO TO 25:

C C-----MOLE EXCAVATION INCLUDING SET UP
205: SETUPL = (130. * (FXDIAM+5.0)/(2+65000) ) * LABORI/DEPTH
SETUPW = (165. * (EXDIAM-1.0)/(2+20000) ) * EQUIPI/DEPTH
SETUPB = (300. * (EXDIAM-1.0)/(2+32000) ) * MATLSI/DEPTH
SETUP = SETUPL + SETUPW + SETUPB
EXC STL = (150. + XDIAM*2 + 84) * EQTIME*LABORI
EXC STE = (100. + XDIAM*2) * EQTIME*EQUIPI
EXC STM = (150. + XDIAM*2 + 84) / ADVANC(I) + EXDIAM*2 / 150000 * MATLSI
EXC STT = EXC STL + EXC STE + EXC STM
EXTOT = EXC STT + SETUP

C C-----MUCKING
C HOISTING
C 256: MKH STL = (29.0 + XDIAM*2) + DEPTH * (1.0 + XDIAM*8 * RML(I)) * EQTIME*LABORI
MKH STE = (3.0 + XDIAM*4 * RML(I)) * EQTIME*EQUIPI
MKH STM = (0.5 + VOLUME/1000) * MATLSI
MKH STT = MKH STL + MKH STE + MKH STM

C DISPOSAL
MKDSPL = (15.0 + 4.7 * DOS) + VOLUME * ADVANC(I)/1000 * EQTIME*LABORI
MKDSPE = (6.0 + 12.5 + 2.1 * DOS) + VOLUME * ADVANC(I)/1000 * EQTIME*EQUIPI
MKDSPM = (350 * DOS) * VOLUME * MATLSI
MKD S PT = MKDSPL + MKDSPE + MKDS PM
MKC TOT = MKH STT + MKD S PT

C C-----SUPPORT
C EXCEPT FOR MOLE EXCAVATION WITH ROCKBOLTS, THE ONLY SUPPORT COST IS
C FOR MATERIALS
SUPTOT = 0
IF (ISUP EQ 20) GO TO 300
IF (ISUP EQ 41) GO TO 310
IF(ISUP, RQD) GO TO 350
GO TO 9999
C STEEL SETS
300 TEMP = 0.7 * DIAM - 6.
IF (TEMP.LT.0.0) TEMP = 0.0
GO TO (301, 3.2, 1)
301 WST = 3.56 * ((RQD - 20.) / 100.) * 2 + 1.22 * EXP(SORT(TEMP) + 3.7 - 2.)*
1 * (RQD + 25.0) / 100.) ** 2
RL = 0.67 * 18. * DIAM + 0.56 * DIAM * ((117.0 - RQD) / 100.) ** 2
GO TO 313
302 WST = EXP(SORT(TEMP) + 3.7 - 2. * (RQD + 20.0) / 100.) ** 2
RL = 0.67 * 15. * DIAM + 0.45 * ((117.0 - RQD) / 100.) ** 2
303 SUPTOT = (1.95 * WST + 15.0 ** RL) * MATLSI
GO TO 350
C C ROCK BOLTS
310 SUPE = 0.1
SF = 1.67
C COMPUTE WEIGHT OF ROCK BOLTS IN POUNDS PER LINEAR FOOT
WRB = (RMPF(I) * DIAM ** SF) * EXP(26.26 * DIAM) * (150.0 - RQD) ** 2 / (152.0 - RQD) ** 2
C WEIGHT OF WIRE MESH AND STRAPS
WWM = 3.3 * DIAM ** SF * 1.0 - RQD / 100.0
IF (I.EQ.2) SUPE = 1.75 * EQTIME * EQUIPI
SUPM = (0.32 * WWM + (0.23 + RS / 530000) * WRB) * MATLSI
SUPTOT = SUPE + SUPM
C C Lining
350 LINTOT = 0.0
INDIC = ILIN / 10
IF (INDIC .EQ. 0) GO TO 370
IF (INDIC .GT. 2) GO TO 9999
VL = 3.1416 * ((EXDIAM / 2.) ** 2 - ((DIAM - LTHICK / 6.) / 2.) ** 2) / 27.
GO TO (360, 3.65, INDIC)
C CONCRETE
360 DF = DIAM - LTHICK / 6.
LINL = (125. + 3.1 * VL + 25.0 * 4.0 * BF) * EQTIME * LABORI
LINE = (7o + 42 * VL + 1o + 1o * 5 * RF) * EOTIME * EQUIPI
LINM = (13o * VL + 7 * RF) * MATLSI
GO TO 369

C
C SHOTCRETE
365 LINL = (5.5 * (VL + 3 * 2) * 2 + 1o) * EOTIME * LABORI + MKHSTL
LINL = (1.625 * (VL + 8o) * 2 - 23) * EOTIME * LABORI + MKHSTL
LINM = 14o * VL * MATLSI
369 LINTOT = LINT + LINES + LINM

C
C ----- GROUTING
370 GROTOT = '

C ----- FINAL TOTAL
400 TOTAL = EXTOT + MUCTOT + Suptot + LINTOT + GROTOT
IMUC = '
ACTOT = '
IF (OUT, EQ, 2, 0) WRITE (IOUT, 2001) IEXC, ISUP, ILIN, IGRO, IMUC, EXTOT,
1 SUPTOT, LINTOT, GROTOT, MUCTOT, ACTOT, TOTAL
2001 FORMAT (*) FOR EXCAVATION: 'I3', SUPPORT: 'I3', LINING: 'I3',
1', GROUTING AND PUMPING: 'I4', MUCKING: 'I3',/
2 THE COSTS ARE: EXCAVATION: 'F7.2', SUPPORT: 'F7.2', LINING: 'F7.2',
3F7.2', GROUTING AND PUMPING: 'F7.2', MUCKING: 'F7.2', /
4 AIR CONDITIONING: 'F7.2', TOTAL: 'F7.2', /
IF (TOTAL, GE, STORE) GO TO 999
C ----- IF WE HAVE A NEW MINIMUM, STORE THE TOTAL COST AND THE METHOD CODES
STORE = TOTAL
METH (1) = IEXC
METH (2) = ISUP
METH (3) = ILIN
METH(4)=IGRO
METH(5)=0
C-----ALSO STORE COMPONENT COSTS
PRICE(1)=EXTOT
PRICE(2)=SUPTOT
PRICE(3)=LINTOT
PRICE(4)=GRVTOT
PRICE(5)=MUTOT
PRICE(6)=CF6
GO TO 999
C
C-----ERROR TRAP
9999 WRITE(UNIT,5999) IEXC,ISUP,ILIN,IGRO
999 FORMAT('INVALID COMBINATION:',514)
999 RETURN
END
APPENDIX D

PROGRAM FOR POWDER CALCULATIONS
This program is based upon the work of Langefors and Kihlstrom (1963). Their empirical equations have been modified slightly to give consistent results. The user must provide the area of the face and the advance per round. He also must supply:

1) Either number of holes or stemming, or both.
2) Either degree of packing or hole diameter, or both.
3) Specific charge.
4) Strength of explosive.

Either 1 or 2 must specify both. The charge per foot of hole is also calculated. All computations are done in metric units.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Variable</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advance Per Round</td>
<td>APR</td>
<td>Meters</td>
</tr>
<tr>
<td>Area</td>
<td>AREA</td>
<td>Meters$^2$</td>
</tr>
<tr>
<td>Hole Diameter</td>
<td>DIAM</td>
<td>mm</td>
</tr>
<tr>
<td>Stemming</td>
<td>STEM</td>
<td>Meters</td>
</tr>
<tr>
<td>Number of Holes</td>
<td>N</td>
<td>-</td>
</tr>
<tr>
<td>Strength of Explosive</td>
<td>S</td>
<td>(Relative, Dynamite=1.0)</td>
</tr>
<tr>
<td>Specific Charge</td>
<td>SC</td>
<td>kg/m$^3$</td>
</tr>
<tr>
<td>Degree of Packing</td>
<td>PACK</td>
<td>kg/dm$^3$</td>
</tr>
<tr>
<td>Charge Per Foot</td>
<td>L</td>
<td>kg/m</td>
</tr>
</tbody>
</table>

The program is written in FORTRAN, input format is F11.1, one value per card. If a value is not specified, a large negative number (-1 x 10$^{50}$) must be entered in its place.
The order of input is as shown in the example.
EXAMPLE
INPUT DATA:

APR = 2.70
AREA = 100.00
DIAM = 36.00
L =
N = 100.89
PACK =
S = 1.00
SC = 0.60
STEM = 0.50

RESULTS OF CALCULATIONS

APR = 2.70
AREA = 100.00
DIAM = 36.00
L = 0.64
N = 100.89
PACK = 0.76
S = 1.00
SC = 0.60
STEM = 0.50
Q = 162.00
NC = 100.89
NP = 100.89
EXPLOSIVE CALCULATIONS

This program calculates the amount of explosive material

APR = ADVANCE PER ROUND
LEN= LENGTH OF HOLES
AREA = AREA OF FACE TO BE BLASTED
N = NUMBER OF HOLES TO BE DRILLED
STEM = STEMMING
Q = QUANTITY OF EXPLOSIVE
NC = NUMBER OF CAPS
NP = NUMBER OF PRIMERS
IMPLICIT REAL (J, K, L, M, N)

SET INITIAL TEST CONSTANT
TESTIN = -1.592

SET DATA SET REFERENCE NUMBERS
IIN = 5
ICOUT = 6

READ IN DATA
READ (IIN, 101) APR
READ (IIN, 101, END=999) AREA, DIAM, L, N, PACK, S, SC, STEM
1001 FORMAT (F11.2)
WRITE (ICOUT, 102)
1002 FORMAT ('INPUT DATA: ')
WRITE (ICOUT, 2001) APR
2001 FORMAT ('APR=', F11.2)
WRITE (ICOUT, 3001) AREA, DIAM, L, N, PACK, S, SC, STEM
LEN=15, APR/9.

MAKE SURE EITHER N OR STEM HAS BEEN INITIALIZED
IF (N .LT. TESTIN) AND (STEM .LT. TESTIN) GO TO 900

CHECK TO SEE IF PACK WAS INITIALIZED
C----- OUTPUT OF RESULTS
800 WRITE(IOUT,1004)
1004 FORMAT ('RESULTS OF CALCULATIONS:',/)
WRITE(IOUT,2001) APR
WRITE(IOUT,1003) AREA,DIAM,L,N,PACK,S,SC,STEM
Q=L+N*(LENGT-STEM)
NC=N
NP=N
WRITE(IOUT,1005) Q,NC,NP
1005 FORMAT (' Q=',F10.2,' NC=',F10.2,' NP=',F10.2)
GO TO 999

C----- ERROR TRAP
950 WRITE(IOUT,950)
950 FORMAT (' INSUFFICIENT DATA')
999 CONTINUE
999 STOP
C END