AD-763 565

COMPUTER SIMULATION OF HARD ROCK TUNNELING. VOLUME II-ADDENDUM

M. M. McJunkin

General Research Corporation

Prepared for:

Bureau of Mines
Advanced Research Projects Agency

November 1972

CR-2-190

COMPUTER SIMULATION OF HARD ROCK TUNNELING

Final Technical Report
Addendum to Volume II

by
M. M. McJunkin

November 1972

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1579
Amendment 2

GENERAL RESEARCH CORPORATION
WASHINGTON OPERATIONS
1501 WILSON BLVD, SUITE 700, ARLINGTON, VA. 22209

UNCLASSIFIED
Security Classification

AD 763565

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) General Research Corp. 1501 Wilson Boulevard Arlington, Virginia 22209	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP --
3. REPORT TITLE Computer Simulation of Hard Rock Tunneling	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final Technical Report - Addendum to Volume II	
5. AUTHOR(S) (First name, middle initial, last name) M. M. McJunkin	
6. REPORT DATE November 1972	7a. TOTAL NO. OF PAGES ii + 20 7b. NO. OF REFS --
8a. CONTRACT OR GRANT NO. H0110238 b. PROJECT NO. 1578 Amendment No. 2 c. Program Code No. 1F10 d. Work Unit No. F53701	9a. ORIGINATOR'S REPORT NUMBER(S) CR-2-190 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
10. DISTRIBUTION STATEMENT Distribution of this document is unlimited.	
11. SUPPLEMENTARY NOTES Reproduced from best available copy.	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209

13. ABSTRACT

A model of the hard rock tunneling process is developed including a three-dimensional stratified geology, a modular representation of many excavation system possibilities, and a cost-accounting system to facilitate cost-benefit analysis of tunneling system performance.

This supplement to Volume II provides documentation and program guides for a cyclic drill and blast control program and a stochistic boring machine failure model.

The model is implemented in FORTRAN 4 for CDC 6000 Series Computers. A computer source tape is available from NTIS.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Mining						
Excavation						
Tunneling						
Computer Simulation						

CR-2-190

COMPUTER SIMULATION OF HARD ROCK TUNNELING

Final Technical Report
Addendum to Volume II
November 1972

ARPA order No. 1578, Amendment 2	Program Code No. 1F10
Contract No.	H0110238
Effective Date of Contract	26 October 1970
Contract Expiration Date	
Phase I:	30 September 1971
Phase II:	30 April 1972
Amount of Contract	
Phase I:	$124,182
Phase II:	$ 66,288
Principal Investigators	R. R. Hibbard L. M. Pietrzak (703) 524-7206

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Bureau of Mines

GENERAL RESEARCH CORPORATION
Washington Operations
1501 Wilson Boulevard
Arlington, Virginia 22209

PREFACE

This supplement to Volume II provides documentation and program guides for a cyclic drill and blast control program and a stochastic boring machine failure model.

CONTENTS

1 CYCLIC PROGRAM DOCUMENTATION

1.1 CYCLIC CONTROL PROGRAM: CONTR1

The simulation of tunneling using the cyclic processes, drill and blast technique for rock fragmentation, and machine loaders and shovels for short-haul materials handling uses the following routines:

Element and General Process	Activity	Subroutine Entry Initialization/Normal
Rock fragmentation, drill and blast	Move drill jumbo in	MOVEIN/INMOVE
	Set charge	HOLBRN/BRNHOL
	Move drill jumbo out and blast	MOVEUT/OUTMOV
Materials handling, loaders and shovels and rail system	Moving shovels in	MUKIN/INMUK
	Mucking	MUKLOD/LODMUK
	Moving shovels out	MUKOUT/OUTMUK
	Train logistics	RAILHL/RAILTR
	Track laying	RAILEX/RAILXT
	Discharge	RAILDS/RAILDS
	Train maintenance	RAILMT/RAILMT
Ground support	Rock bolts, steel sets, shotcrete or combinations	GRNSUP/GROUN
	Maintenance	GSMAIN/GSMNT
Environmental control	Ventilation, cooling, and water removal	ENVIR/ENVIRC

The control program which has been created for the simulation is very similar to the one created for the Strawberry Aqueduct Layout Tunnel described in Volume II of Appendix IV of the final report. The subroutines associated with the control program for the Layout Tunnel for reading in input parameters (INPUT), and generating output reports (OUTPUT, MAXADV, REPORR, REPORC) are used for this simulation, the only change being substitution of drill and blast variables (input and status) for boring machine variables, and machine loader and shovel variables (input and status) for integrated conveyor loader variables. The description of all activity routines used is in Appendix IV of Volume II.

The control program itself (CONTR1) has the same tasks as the layout tunnel simulation control program had:

a. Reading in and computing input parameters for the processes being modeled

b. Reading in new tunnel geology parameters as they are required

c. Cycling through all the activity subroutines in each time step

d. Updating the face position and time at the end of each time step

e. Determining the status of the rock fragmentation process for the next time step by checking the status of other processes and queuing trains at the loading area if it is required

f. Generating output reports when they are required

g. Making error checks and ending the simulation if an error has occurred

h. Ending the simulation if user input time or distance maximums have been reached, or if the tunnel geology file end has been reached

Two other computations have been put into this control program which were contained in the activity subroutines in the Layout Tunnel simulation.

aa. Computing utilization and availability for each element (rock fragmentation, materials handling, ground support, and environmental control)

bb. Computing labor time (used to compute labor costs in the report subroutine REPORC) for each element

These two calculations were put into the control program for greater user flexibility. Changing the way these are computed is more easily done when the statements are in one routine instead of spread throughout the activity subroutines. The control program tasks will be discussed next, followed by a description of new variables added to the model, and finally a description of changes to the activity subroutines when incorporating will be given.

The control program first handles input by calling INPUT which reads in and prints out parameter cards, initializes the geology file by reading in the first record, and computes some initial values for the run. Further initialization is done by calling each activity subroutine which has an initialization entry point at that entry. The subroutines then set default values for input parameters which were not read in for this run.

After calling each subroutine, all input parameters are again printed out by calling INPUT2. These will reflect any default values applied by the subroutines. The next initialization step is setting all future status variables (COMNEW) equal to the corresponding present status (COMOLD). This is done to initialize all processes for the start of the run. The final initialization step is the printing out of the initial status of all elements by calling OUTPUT.

The control program then begins stepping through the time cycles (DT). In each time step it first checks the face position versus the geology file to determine if a new geology has been reached. If it has, the next geology record is read in. Each tunneling routine is then called in succession at its normal entry point. The order in which the elements are processed is not significant; however, the activity routines for each element should be called in the order shown in the listing. For example, the rail subroutines may be called after the machine loader and shovel routines, but when the rail routines are called, the order should be RAILTR, RAILDS, RAILEX, RAILMT. At each call to a subroutine the subroutine checks the status of the activity it represents by looking at its current status variables, acts upon this information, and finally sets its status for the next time cycle by setting its future status variables. For example, when MUKLOD is called it first checks LOADR, its current status variable. If it is equal to 1, then mucking is not being done. No action is required of the subroutine since its future status variable LLOADR is already 1 for the next time cycle, so a return to the main program is executed. However, if upon entry, LOADR = 0, mucking is being done, and the subroutine computes the amount of muck that is loaded in one time step (DT). It then checks to determine if all muck will have been loaded after DT. If loading will be completed, the future status variable LLOADR is set = 1 indicating to the next time step loading is finished. If there is still muck to be loaded (DV ≠ 0), then LLOADR is set to 0 indicating loading should continue in the next cycle. In either case a return to the main program is executed after setting LLOADR.

In calling the tunneling subroutines, two changes have been made because of the cyclic nature of the drill and blast and mucking processes. The first is that the start of the mucking cycle after the blast and the start of the blasting cycle after mucking are handled in the control program. This allows the user to overlap these processes if he desires or alternate them. In the sample control program the processes are not overlapped but run in alteration. To allow the control program to start the processes, two new parameters which signal the operation of each activity in the drill and blast process and the mucking process, IACT and JACT, have been added. These variables have been placed in common areas and are available to the subroutines and the control program. They have the following possible values:

IACT = 1 - drill jumbo being moved to face
2 - drilling
3 - setting the charge
4 - moving the drill jumbo out, blasting, and smoke time
5 - finished drill and blast cycle

JACT = 1 - moving muckers and shovels to the face
2 - mucking
3 - moving muckers and shovels away from face
4 - mucking cycle finished

The control program starts a process by setting IACT or JACT and the status variables of the lead activity. For example, the drill and blast process is started by setting LJUMIN = 0, IACT = 1. The activity of moving the drill jumbo to the face is to begin. The subroutine MOVEIN representing this activity will begin the move when it is called. The activity subroutines themselves handle the cycling of each process once it has been started by the control program.

The second change due to the cyclic nature of rock fragmentation and short haul materials handling is not calling drill and blast subroutines or mucking routines when these processes are to stop for some reason other than their activities being completed. In the sample control program they are not called when ground support is further from the face than the maximum allowed unsupported length (IEXCA = 1). Not calling them allows whatever process that is occurring to stop immediately and then be resumed in the same place after ground support has caught up.

After all the subroutines which are going to be called in the time step have been called, the control program cumulates operating time, down time, and labor time for each of the elements. As mentioned previously, these computations (aa, bb) were previously done in the activity subroutines. Both of these computations are done on the basis of the status flags in the two common areas, future (NEWCOM) and present (OLDCOM). Also used for the computations is a special status variable, MBLAST, which, as in the sample control program for the boring machine, has been added to keep track of all activities effect on the drill and blast process. It takes the following values:

MBLAST = 0 - rock fragmentation is in progress

1 - rock fragmentation idle due to rail being extended

2 - rock fragmentation idle due to switch being moved

3 - rock fragmentation idle, switch being added

4 - rock fragmentation idle, no train available for loading

5 - rock fragmentation idle, loading muck

6 - rock fragmentation idle, ground support has fallen behind

7 - rock fragmentation is down, drill equipment in maintenance

This variable and the current status variables in common block OLDCOM are used to determine the utilization time and the labor time in the current time step. This time is then added to the already cumulated operating times and subroutines. In the sample control program this is done as shown in Tables 1 and 2. (See Table 3 for definition of status variables.)

The control program now prepares for the next time step by incrementing the face position (X) by DX, the advance during the current time cycle, incrementing time (T) by DT, and setting current status variables (COMOLD) equal to future status variables (COMNEW), thus indicating the status of each activity at the beginning of the new time cycle. Using the new current status variables (COMOLD), the status of rock fragmentation

TABLE 1

Element	Cumulative Operating Time Variable	Cumulative Down Time Variable	Incremented by DT if:
Rock fragmentation	RDUTIM		MBLAST = 0
		RDDTIM	MBLAST = 7
Materials handling	UMHTIM		NUMLD ≠ 0 and MBLAST = 5
		DMHTIM	NUMLD ≠ 0
Ground support	GSUTIM		MGS(IAS) = MGS(IST) = 0
		GSDTIM	MGS(IAS) = 1 or MGS(IST) = 1
Environmental control	ECUTIM		Always
		ECDTIM	Never

TABLE 2

Element	Cumulative Labor Time Variable	Incremented in Time Cycle by DT if:
Rock fragmentation	RDTIME	Always
Materials handling		
Short haul	HTIME(2)	Always
Long haul	HTIME(1)	Always
Ground support	GSTIME	Always
Environmental control	ECTIME	Always

TABLE 3

DEFINITION OF FLAGS SET IN MODEL

Subroutine	Flag	Options
MOVEIN, HOLBRN, MOVOUT, SETCHG	MBLAST	= 0 - rock fragmentation in progress = 1 - rock fragmentation idle, rails are being extended = 2 - rock fragmentation idle, switch being moved = 3 - rock fragmentation idle, switch being added = 4 - rock fragmentation idle, no train available for loading = 5 - rock fragmentation idle, muck being loaded = 6 - rock fragmentation idle, ground support has fallen behind = 7 - rock fragmentation down, drills in maintenance
MOVEIN	JUMIN	= 0 - drill jumbo being moved to face = 1 - drill jumbo move completed
HOLBRN	NDRILL	= 0 - drilling in progress = 1 - drilling completed
HOLBRN	NDRILB	= 0 - burn cutting in progress = 1 - burn cutting finished
SETCHG	NCHRG	= 0 - setting charge in progress = 1 - setting charge completed
MOVOUT	JUMOUT	= 0 - drill jumbo being moved from face = 1 - drill jumbo move completed
MUKIN	MKIN	= 0 - muckers being moved to face = 1 - muckers move completed

MUKLOD	LOADR	= 0 - loading muck in progress = 1 - loading muck completed
MUKOUT	MUKOT	= 0 - muckers being moved from face = 1 - muckers move completed
RAILHL	LSTOP	= 0 - normal operations--continue system = 1 - shut down system--external reasons = 2 - bring up system--external reasons
RAILDS	IFILD	= 0 - normal operation = 1 - discharge area filled
GRNSUP	IEXCA	= 0 - normal operation = 1 - ground support construction rate exceeded by rock fragmentation
GSMAIN	MGS(I) I = 1, 5	= 0 - support type I up = 1 - support type I in maintenance
RAILEX	NSWCH	= 0 - normal operation = 1 - new switch being added = 2 - switch being moved
RAILEX	NBYBY	= 0 - normal operation = 1 - rail extension rate exceeded by rock fragmentation
RAILHL, RAILDS, RAILEX	LTSTAT(I, 1) I = 1, 25	= 0 - train I stopped, empty = 1 - train I stopped, full = 2 - train I accelerating, empty = 3 - train I accelerating, full = 4 - train I going full speed, empty = 5 - train I going full speed, full = 6 - train I decelerating, empty = 7 - train I decelerating, full = 8 - in switch, empty = 9 - in switch, full = 10- in discharge area = 11- in maintenance area

LKSTOP(I) I = 1, 25	= 0 - train I not stopped = 1 - train I at face, being loaded = 2 - train I in switch, train ahead = 3 - train I in switch, train approaching = 4 - train I stopped, external reasons = 5 - train I in switch, waiting to enter discharge area = 6 - train I in switch, waiting to enter load queue = 101 train I in load queue, 1st in line = 102 train I in load queue, 2nd in line = 10N train I in load queue, Nth in line
LLVHCL	= 0 - no train in loading area = n - train n in loading area

(MBLAST) is determined for the new time cycle. This is done in the control program because rock fragmentation is the lead-off, or pivotal, element. The other elements, materials handling, ground support, and environmental control, will automatically follow the rock fragmentation, generally using feet of advance (DX) or volume of muck to be removed (DV), with no interference necessary from the control program. However, the rock fragmentation process must be told what other processes are going on. This could have been included in the rock fragmentation subroutines, but would have made the modeling rigid. It was assumed that the user would prefer the flexibility of being able to vary the interference of say ground support with rock fragmentation, making runs in which these processes go on simultaneously and other runs in which ground support falling behind (further from the face than the maximum unsupported length) idles the rock fragmentation. This can easily be done using the status variable IEXCA which = 1 when ground support falls behind. The user then inserts one statement in the control program bypassing the rock fragmentation subroutines, thus making rock fragmentation idle for the time step. For other processes, similar statements will cause varying relationships between rock fragmentation and materials handling or environmental control; these relationships in turn affect the progress of the tunnel, the utilization reports, and costs.

Besides setting the status of rock fragmentation, status variables for the materials handling system are used to keep track of the trains queuing at the loading area and queue them when a train is released to the discharge area. The queuing of the trains is done in the control program to preserve the independence of the loader subroutines (MUKIN, MUKLOD, MUKOUT) and the rail subroutines (RAILHL, RAILEX, RAILDS, RAILMT). A check is first made to determine whether a train is needed (MBLAST = 4). Another check is made to determine if track is being extended (NBYBY = 1) or a switch is being added or moved (NSWCH = 1 or 2). If a train is needed and neither track nor switches are being moved or added, a final check determines whether a train is already loading. If no train is loading, the train queue is changed to attempt to provide a train for loading. If any of the above conditions was not met, then no queuing is done.

Finally, the control program determines whether reports should be generated and generates them if it is time. It ends the run then if the user time (TMAX) or distance (XMAX) limits have been reached or errors have been detected. If the run is not finished, it starts the next time step by returning to the statements in the control program which check for a change in tunnel geology.

1.2 NEW VARIABLES: DRILL AND BLAST AND SHOVEL ROUTINES

As was stated in Volume II, plant equipment and labor costs were not included in the routines not in the model simulating the Layout Tunnel. In creating the control program for the cyclic processes (drill and blast and shovels), these costs were added in the manner suggested in Volume II, Appendix IV with some slight differences.

Labor costing for this model is all done in the control program instead of in individual routines as in the boring machine model. The variables used to input the labor costs are the same variables that were used in the other model:

BOMEN(I), I = 1, 10 - number of men required for drill and blast in each of ten possible labor categories

BOCST(I), I = 1, 10 - hourly cost for one man in each of above categories

RAMEN(I), I = 1, 10 - number of men required for rail system in each of ten possible labor categories

RACST(I), I = 1, 10 - hourly cost for one man in each of above categories

COMEN(I), I = 1, 10 - number of men required for muck loaders and shovels in each of ten possible labor categories

COCST(I), I = 1, 10 - hourly cost for one man in each of above categories

GSMEN(I), I = 1, 10 - number of men required for ground support in each of ten possible labor categories

GSCST(I), I = 1, 10 - hourly cost for one man in each of above categories

VLMEN(I), I = 1, 10 - number of men required for environmental control in each of ten possible labor categories

VCCST(I), I = 1, 10 - hourly cost for one man in each of above categories

However, instead of equivalencing these variables to other variables for actual computation of costs, no equivalencing was done and these input variables (BOMEN, etc.) are used for computation in subroutines REPORC. Labor costs are input under the NAMELIST of the process they are associated with just as before, so BOMEN, BOCST are input under NAMELIST/BLAST/, COMEN, COCST under NAMELIST/SHOVL/, etc.

The plant and equipment costs which were added for the drill and blast process and the machine loader and shovel will be cumulated in the variables already in the model RDPLAN for drill and blast, and HPLANT for loaders and shovels. The input costs are as follows:

Input in NAMELIST/BLAST/

EBURN = cost per foot of burn drills
EDRIFT = cost per foot of drifter drills
EJUMB = cost per foot of drill jumbo
EJIB = cost per foot of three jibs
EWHAT = cost per foot of three drill positioners

Input in NAMELIST/SHOVL/

ESHOVL = cost per foot of mucker

1.3 CHANGES IN ACTIVITY SUBROUTINES

The basic descriptions for these routines are in Appendix IV, Volume II and the changes below should be studied with those descriptions.

1.3.1 The Drill and Blast Routines

Each of these routines (MOVEIN, MOVOUT, HOLBRN and SETCHG) has been added to so that the cycling of the drill and blast process (moving in, drilling, setting the charge, moving out, and blasting) is now done by the subroutines themselves instead of by the control program. For this change ICYCLE has been put into the common block BLAST and a new variable IACT (described in Sec. 5.1) has also been added to the common area. Each activity initializes the next activity in the cycle by setting ICYCLE to 1, the activities status variable(s) = 0, and by setting IACT for the next activity. For example, when the subroutine which simulates moving in the drill jumbo, MOVEIN, finds that the moving is complete, it sets ICYCLE = 1 for initializing the drilling activity, sets IACT = 2, LNDRILB = 0 and LNDRILL = 0 to indicate that drilling is in progress. These settings will cause the drilling subroutine HOLBRN to begin drilling in the next time step. As a result of ICYCLE being put in a common area, it is no longer a formal parameter appearing in the subroutine call statement or name.

Another change made to each of the activity routines is the addition of a separate initialization entry to be called at the start of each run only. The initialization entry and normal entry for each of the routines are as follows:

Activity	Initialization Entry	Normal Entry
Moving drill jumbo in	MOVEIN	INMOVE
Drilling holes	HOLBRN	BRNHOL
Setting the charge	SETCHG	CHGSET
Moving drill jumbo out and blasting	MOVOUT	OUTMOV

A final change for the drill and blast routines has been to add a plant and equipment cost cumulator in MOVOUT. The new variables involved in costing plant and equipment were given in the previous Sec. 5.2. The cost is accumulated in the variable RDPLAN.

1.3.2 The Machine Loader and Shovel Routines

Each of these routines, MUKIN, MUKLOD, and MUKOUT, has been added to so that the cycling of the mucking process (moving muckers to the face, loading muck, and moving muckers out) is now done by the subroutines themselves, once the cycle is started by the control program. For this change a new variable, JACT, (see previous Sec. 5.1 for description) has been added to the common area CONVEY. Each activity initializes the next one in the cycle by setting JACT equal to the value signaling the next activity's start, and by setting the next activities status variables = 0.

Also, cumulation of plant and equipment costs have been added to subroutine MUKIN, the costs being accumulated in the variable HPLANT.

2 STOCHASTIC MODEL OF BORING MACHINE FAILURE

The REPAIR subroutine has been replaced by three subroutines: REPAIR, OPDSTR, and OPROD. A new common block COMMON/OPSTAT/ has been added to the control program and to the three new subroutines. It contains the following variables:

IWORP = 0 if the Weibull function is to be used to determine hours between maintenance for the boring machine

= 1 if n distribution input by the user is to be used

JWORP = 0 if the Weibull function is to be used to determine how long each maintenance period will be for the boring machine

= 1 if a distribution input by the user is to be used

OWEIBE, OWEIBV, OWEIBK = input variables for the Weibull function, F, defining time between maintenance periods

$$F = 1 - \exp - \left(\frac{x - OWEIBE}{OWEIBV - OWEIBE}\right)^{OWEIBK}$$

for $x \geq OWEIBE$

$$F = 0$$

for $x < OWEIBE$, where x = elapsed operating hours since last maintenance period

RWEIBE, RWEIBV, RWEIBK = input variables for the Weibull function, F, defining hours in maintenance

$$F = 1 - \exp - \left(\frac{x - RWEIBE}{RWEIBV - RWEIBE}\right)^{RWEIBK}$$

for $x \geq RWEIBE$

$$F = 0$$

for $x < RWEIBE$, where x = elapsed maintenance hours

OCP(1), OCPT(1), I = 1,100 = one hundred possible discrete pairs of numbers (input by the user) representing a piece-wise linear distribution function to determine hours between boring machine maintenance periods

OCP(I) = value between 0 and 1 giving the probability of the boring machine going into maintenance

OCPT(I) = elapsed operating time at which the above probability is valid

RCP(I), RCPT(I), I = 1,100 = one hundred possible discrete pairs of numbers (input by the user) representing a piece-wise linear distribution function to determine the length in hours of each maintenance period for the boring machine

RCP(I) = value between 0 and 1 giving the probability that the boring machine maintenance is finished

RCPT(I) = elapsed maintenance hours at which the above probability is valid

Note for the arrays RCP(100) and OCP(100) that

RCP(1) = OCP(1) = 0

and

$$RCP(N) = OCP(N) = 1$$

where N is the number of pairs being used in the distribution ($N \leq 100$).

The variables described above should be input to the program within NAMELIST/BORE/.