

https://dspace.org/

Digital computer solution of special electrostatic and elastostatic problems with applications to the mining of tabular deposits

2015-02-09
http://hdl.handle.net/10539/16950

Downloaded from a DSpace repository. https://dspace.org/

in fact, resides in the Fortran routine CONTRL which awaits the next command.

As the program starts, its first function is to call CONTRL with zeroed parameters. This causes CONTRL to read a parameter card from the file, FT05F001. It is followed optionally by the mining plan on one card per each of its lines, which is then kept in a file on disk ready for use. (See Appendix X for rules of the use of the system).

The first card contains, in the first 4 columns, the character CARD or TERM. If "CARD", further input to the system is via cards from file FT05F002. If "TERM" further communications with the system are via an IBM 2741 remote terminal.

To facilitate coding, the Stanford University terminal software support available at the University of the Witwatersrand Computer Centre has been used. Once the routine TMNLIO has been called, the standard Fortran IV READ (5,....) and WRITE (6,....) refer to a 2741 terminal, accessed through file LINE001. This effect is terminated when CALL DISAB. is executed.

Further information on the same card are two switches, the first referring to input and the second to output. These are the first three characters of METRIC (Default) or IMPERIAL.

5.3.2 Configuration Specification Modes

5.3.2.1 Create Mode

This is the first command required in order to enter a configuration for the first time. The condensed flow-chart of TANKSYS illustrating the sequence of routines for this operation is shown Figure 5.2

In CReate Mode, the routine CONTRL is entered thrice, so that switches may be set and routines used in common with other modes of operation. For example, as will be seen later, the sequence CRETID - TSTLST is replaced by SCALPROC - TSTLST in scale mode.

The routine BASINF reads in the system information about the problem as opposed to the physical information. For a full-scale problem TANKSIZE = 64, WINDOWSIZE = 56. The user specifies the problem inside the window. The rim between the window and the tanksize

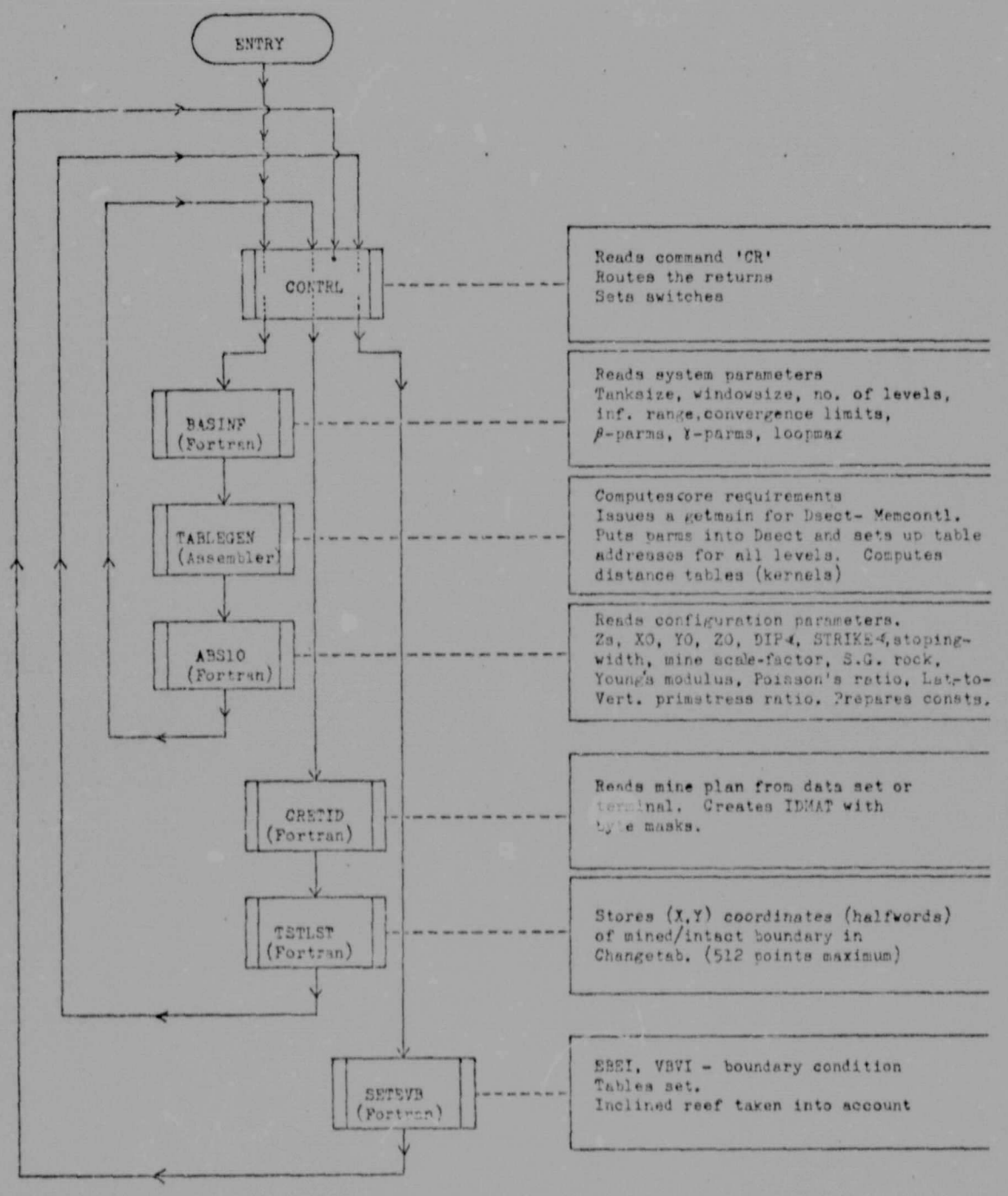


Figure 5.2 Flowchart of Create-Mode Routines

is given as either all intact (usually the case) or all mined. The user is therefore forced to have an untouched rim around the problem, as this is a region where the results of the program are a little unstable and unreliable for the unaware user. Some tests were performed to illustrate this instability and to show that a rim of 4 units is

sufficient for the accuracy desired, (see Appendix IX).

No - of - levels=3 (i.e. 0, 1, 2, 3), has always been chosen since, with an INFLUENCE RANGE of 8, chosen for reasons given in Chapter 4(§4.2.1), this is the minimum number which will always cover the full size tank, as seen in Figure 4.5.

The remaining parameters will be discussed in the section on the solution phase.

The routine, TABLEGEN, computes the parameters associated with each table at each level, and stores them in the section of MEMCONTL, headed 'TABLE ADDRESSING INFORMATION'. As indicated in Figure 5.2, it also issues a GETMAIN to request the core needed for the tables dynamically (i.e. a problem smaller than 64 x 64 will need less core in which to run). Finally, TABLEGEN computes the two kernel tables and their self-effect constants.

The physical constants of the problem are entered via the routine ABSIO.

The co-ordinates of the problem are entered relative to some fixed geographical point, such as the mine co-ordinate system. In this way the 'Z' co-ordinate of the surface is entered, Zs, followed by the co-ordinates of the top left-hand corner of the reef, i.e. point 1,1 of the 56 x 56 window for a full size problem, as illustrated in Figure 5.3. Also on the diagram are shown the angle of dip and angle of strike, as the x-axis is specified along strike.

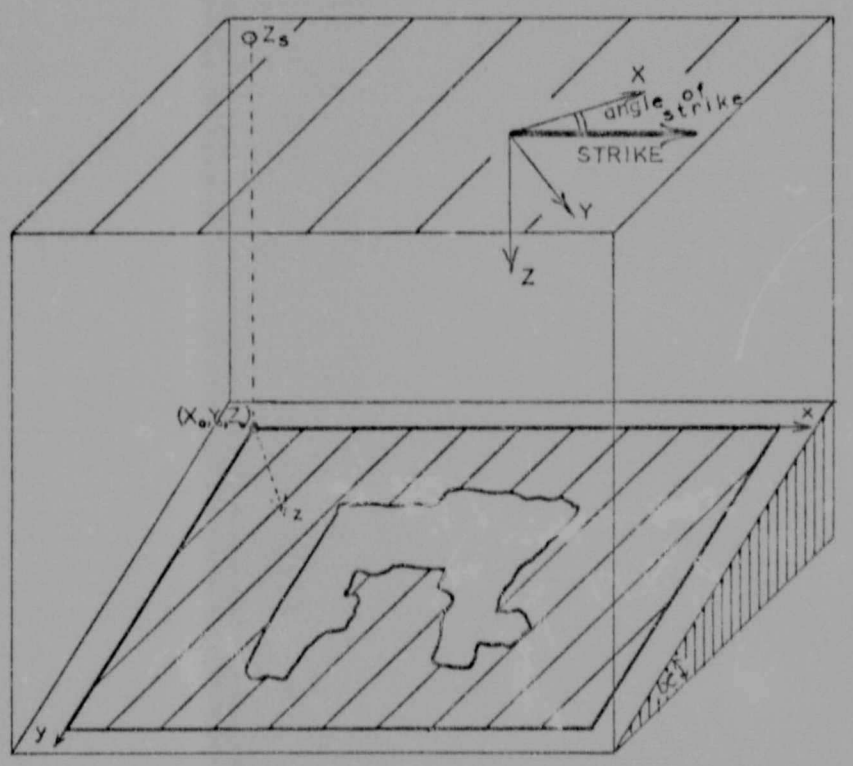


Figure 5.3 Model Co-Ordinates

The SCALE FACTOR is the number of feet per block or metres per block of the tank.

Note that the Zo-Zs must always be a positive quantity, i.e. the mine plan is always below the surface.

Making use of the physical parameters, ABSIO also calculates constants for later use by the output routine on-reef and by the off-reef integrator.

The next operation is using CRETID to form IDMAT, followed by TSTLST, as shown in Figure 5.2. TSTLST stores the co-ordinates of points on the borders between mined and intact regions. The solution values at these critical points are used in the convergence tests of the solution phase.

The last function invoked by 'CREATE' is the setting of the boundary condition tables.

In Chapter 2, it has been shown that the boundary conditions required for the solution of horizontal reefs are given by $\sigma_z = -q_z$ in the mined region. For inclined reefs, however, this gives only the conditions along the top edge, i.e. where y = 0. To compensate for the added primitive stress down the slope of the reef, a factor $\frac{y}{H} \sin \alpha^3$, where $H = Z_o - Z_s$, must be added. Thus, the boundary conditions, assuming that the primitive stress normal to the plane of the ore-body is in a principal stress direction, are :-

$$\sigma_z = -q_z \left(1 + \frac{y}{H} \sin \alpha\right) \text{ inside A} \qquad \text{.....(5.1)}$$

$$S_z = 0 \qquad \text{outside A}$$

The computations are performed using normalized constants, i.e. the multiplier $-q_z$ is removed from (5.1) i.e. replaced by unity and re-applied at the time of printing the final outputs.

Instead of applying equation (5.1) for each region as described, separate boundary condition tables for stresses, EBEI, and for convergences, VBVI, are calculated for the whole tank. This is to avoid complications when a point is updated and changes its type, particularly with regard to the operation of the scaling phase.

This completes the description of the create mode. The tables are now ready for computation.

5.3.2.2 Update Mode

Updating, i.e. changing a configuration, may only be performed once a created configuration has completed a basic solution. The UPDATE command, followed by the list of change-points, should be followed immediately by the C[illegible]N command.

The changes are [illegible] via the Fortran routine UPDID called by TANKSYS. UPDID checks if the change is valid, i.e. if a point is to be mined, it must previously have been intact. Repeated entries are also trapped as errors. The change co-ordinates are stored in pairs of half-words in the table CHANGTAB, with the mask,that IDMAT must be changed to,in byte one of each word. (This is the normal arrangement for IDMAT's main information). In addition, UPDID flags byte 2 of the changed words in IDMAT. A group-number, entered into UPDID by the user, associated with a group of updates, is stored in byte 3 of IDMAT. This number will be explained in the section on energy release.

The maximum number of changes allowed per update-solution combination is 750, as this is the limit set to the length of CHANGTAB.

5.3.2.3 Scale Mode

This mode of operation allows the user to specify the top left corner of a more detailed configuration somewhere inside one that has already been solved. For instance, in a solved problem having a window of 56 x 56, a 28 x 28 area is chosen as specified by its top left co-ordinate, and SCALING implies that this be expanded to the full 56 x 56 window, i.e. the linear dimension has been multiplied by two. This coarse scaling, where each square of the original becomes a group of 4 squares in the scaled version, can now be updated to refine the configuration. Note that this is not the same as starting with the more detailed problem, because the latter would carry no information concerning the surrounding area, as is the case of the scaled problem. The mechanism will be explained later.

The power of this feature becomes clear from the following example, as illustrated in Figure 5.4. The object was to study the shaft pillar in the centre in detail. If the configuration of Figure 5.4 had been entered, initially all the information concerning the surrounding un-mined areas would have been lost. The solution, scale,

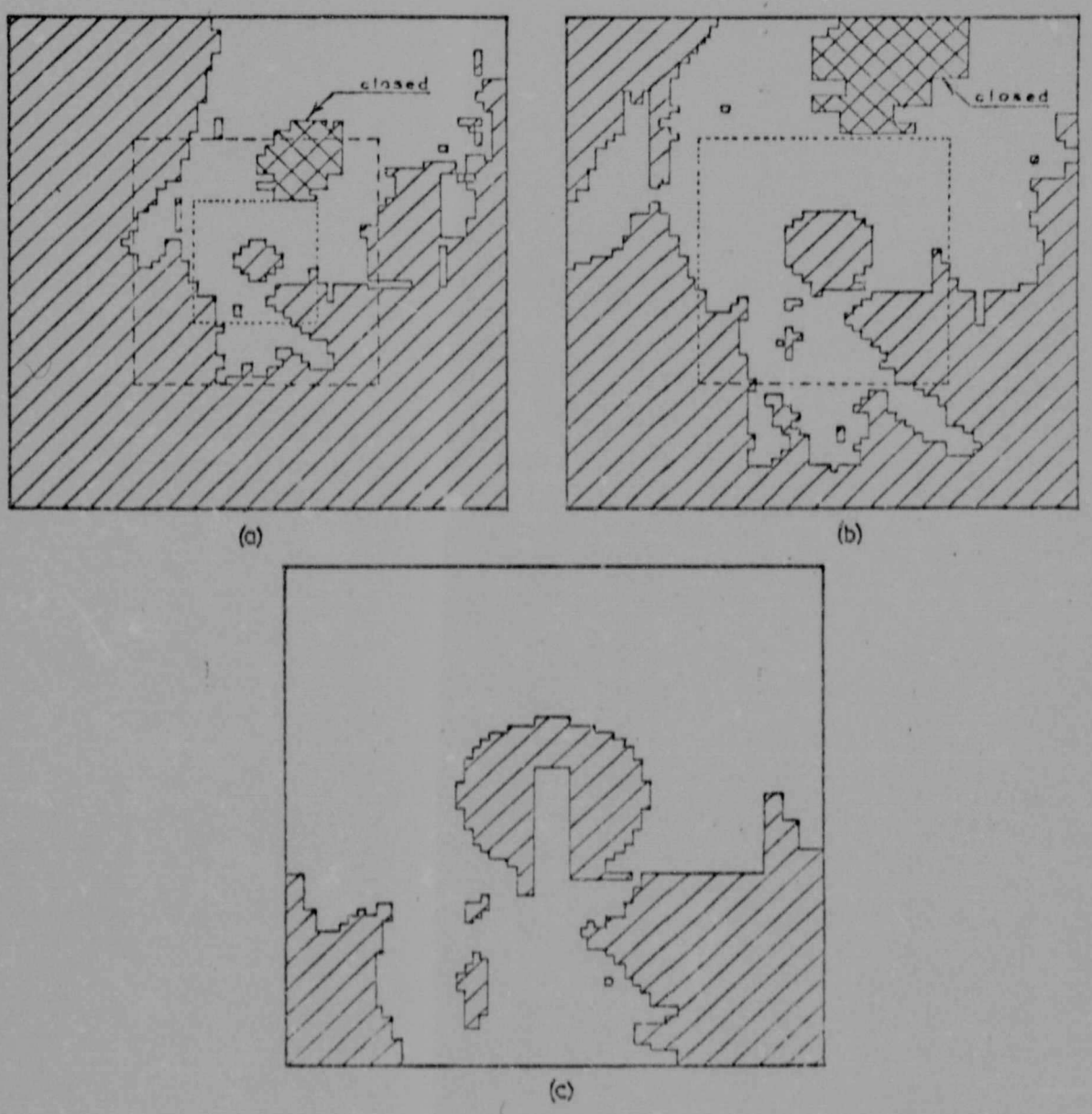


Figure 5.4 Successive Scales-by-2 of Original to Finest Diagram

solution, scale..... sequence would be performed and, on the most detailed solution, mining updates and further studies can be performed, assuming there has been no significant change in the mining outside the window.

Note that the solutions to the configuration of Figure 5.4(a), and 5.4(b), show a totally closed region which will influence the solution of Figure 5.4(c).

To explain the scaling feature, consider the general equations employed in the iterative technique. These are from Chapter 3 (§3.1 equations 3.4).

$$V_{i+1} = V_b - K_1 E_i$$

$$E_{i+1} = E_b - K_2 V_{i+1}$$

summed over the whole tank.

If scaling is introduced, these will now apply in the scaled window and the area outside it, since the purpose of scaling is not to lose the information of the area beyond the scaled window. The above equations may be re-written to take this effect into account:-

$$\left.\begin{aligned} V_{i+1} &= V_b - K_1 E_i - K_1 \overline{E_i} \\ E_{i+1} &= E_b - K_2 V_{i+1} - K_2 \overline{V_{i+1}} \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.2)$$

Where V_i, E_i refer to values inside the scaled window and $\overline{V_i}$, $\overline{E_i}$ refer to those outside.

No charges are assumed to occur beyond the scaled window. It is further assumed that updates inside it will have a negligible effect outside the window. Therefore, E_i and V_i may be taken as constants. Defining new pseudo-boundary conditions

$$\left.\begin{aligned} V_b' &= V_b - K_1 \overline{E_i} \\ E_b' &= E_b - K_2 \overline{V_{i+1}} \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.3)$$

equations (5.2) may be re-written

$$\left.\begin{aligned} V_{i+1} &= V_b' - K_1 E_i \\ E_{i+1} &= E_b' - K_2 V_{i+1} \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.4)$$

which are of the same form as equations (3.4), Chapter 3.

The routine SCALPROC, in addition to changing certain scale factors effectively performs the computation indicated by equations (5.3), i.e. modifies the boundary condition tables.

On issuing the command, SC, TANKSYS calls the Fortran routine SCALE2, whose sole purpose is to read the x,y co-ordinates of the current solution, which are to become the top left corner of the scaled problem. TANKSYS then LINKS to SCALPROC. The general flow-chart of the latter routine is given in Figure 5.5, and needs no further explanation. When SCALPROC has completed its task, TANKSYS calls the Fortran routine TSTLST, as was the case in CREATE mode.

5.3.3 Solution Phase

The same basic solution routine, TANKSIMO is used for the solution after CREATE, UPDATE, SCALE and FINISH. Suitable switches set in TANKSYS, after interlocks have been tested in CONTRL, differentiate between the various solution modes.

As mentioned before, the solution routine, TANKSIMO, and its service sub-routines, are 'ATTACHed' to TANKSYS as a multiprogrammed task, running at a lower priority than the rest of the program, so that graphic facilities, which use very little processor time, may continue independently. A particular word in core called the EVENT CONTROL BLOCK may be interrogated to ascertain whether the attached task is complete.

The basic block-diagram of TANKSIMO appears in Figure 5.6. The comments are supplemented by the description below.

In create mode, TANKSIMO is entered with IDMAT defined only at the zero level. Thus, one of the first tasks is to prepare the lower level IDMAT patterns. The intact/mined masks in IDMAT are in the first byte of each word. A different bit is selected for the two conditions; for example :-

B'10000000' = X '80' represents mined, and
B'01000000' = X '40' represents intact

Since 4 points at a higher level, e.g. level zero, are represented by 1 point at a lower level, e.g. level 1, the lower level may, in general, represent a mixture of some mined and some intact points. Since different bits are used as indicators, as described

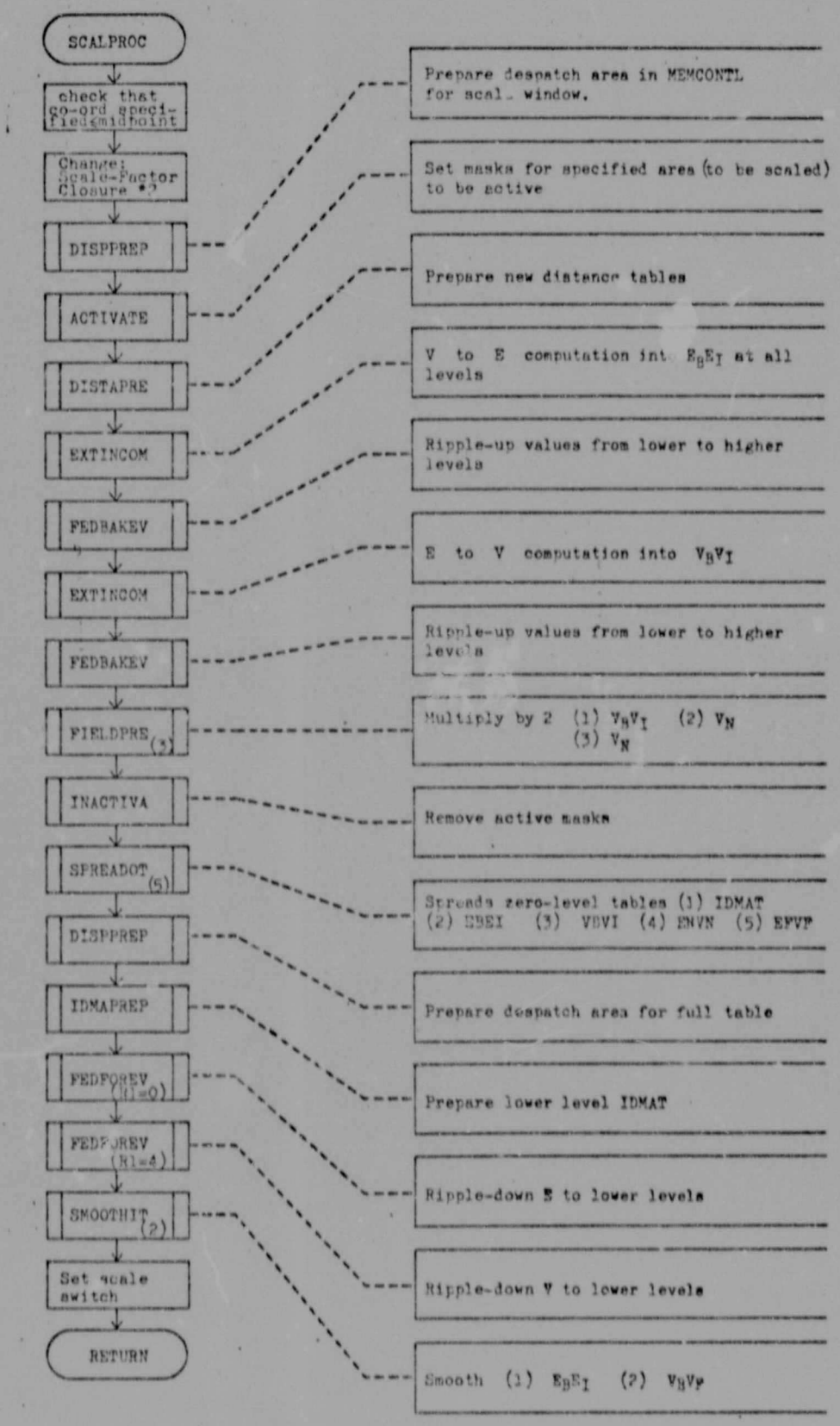


Figure 5.5 Flow Chart of SCALPROC

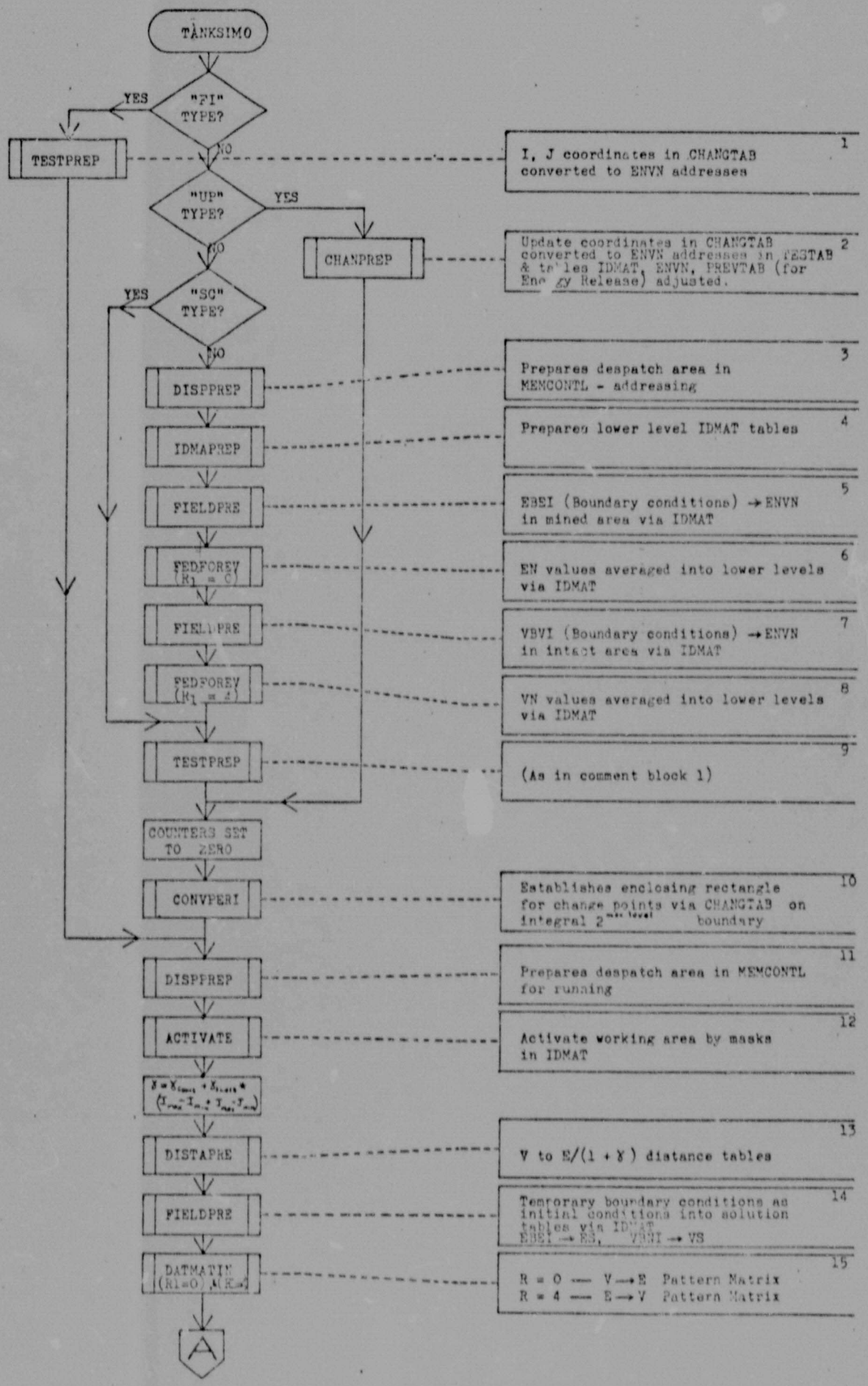


Figure 5.6 Flow Chart of TANKSIMO

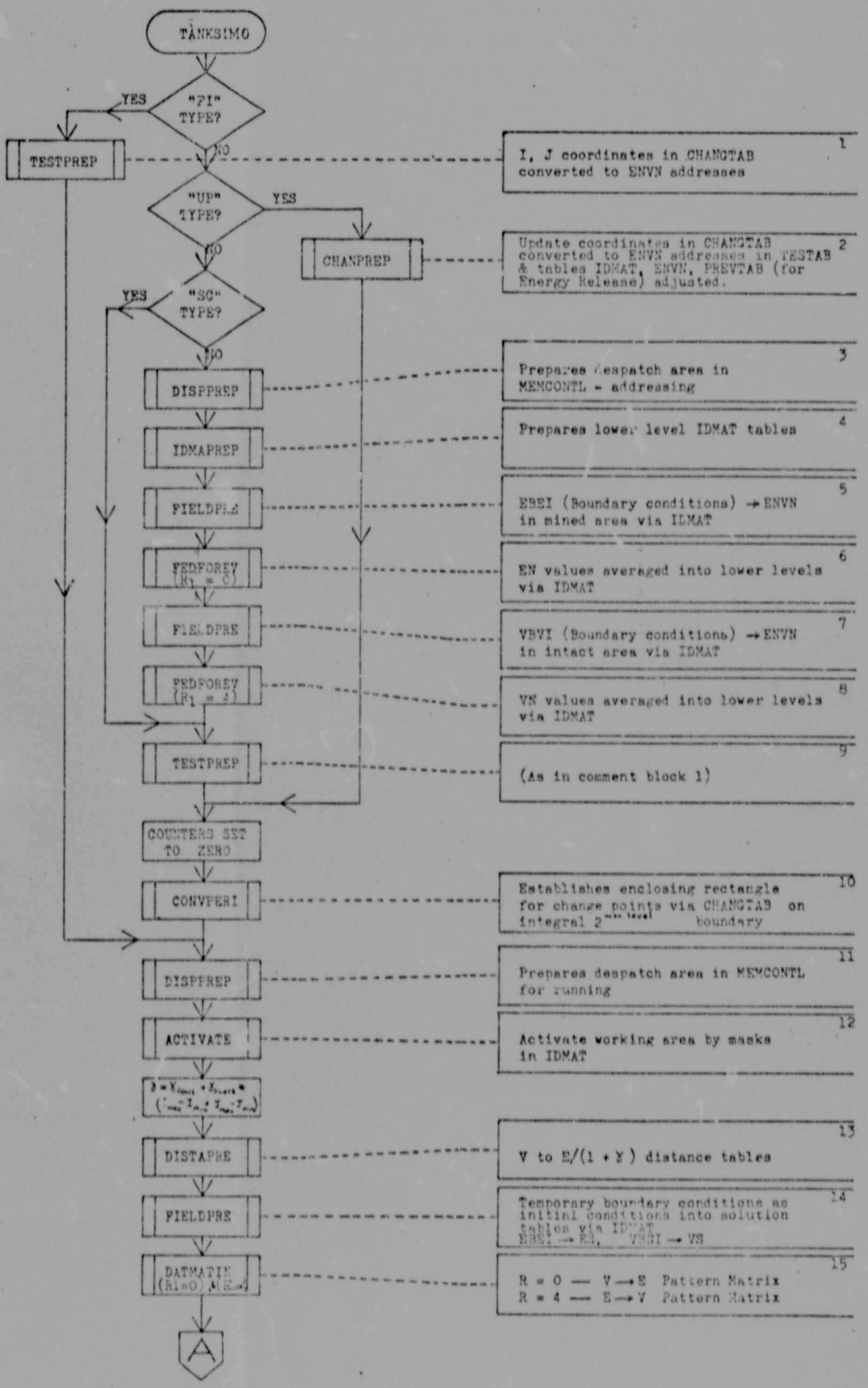


Figure 5.6 Flow Chart of TANKSIM

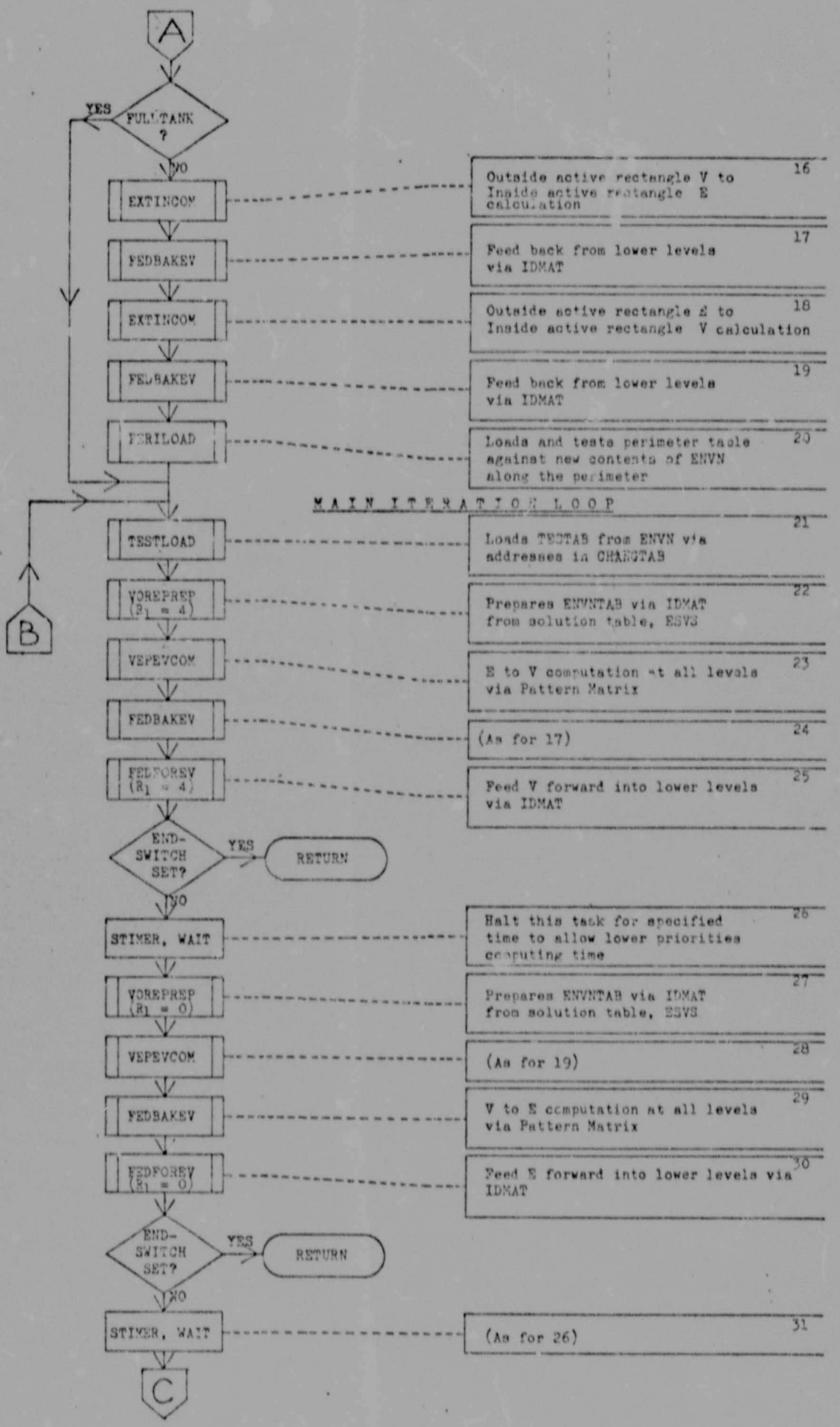

A
YES
FULLTANK ?
NO
EXTINCOM
FEDBAKEV
EXTINCOM
FEDBAKEV
PERILOAD
MAIN ITERATION LOOP
TESTLOAD
VOREPREP (R1 = 4)
VEPEVCOM
FEDBAKEV
FEDFOREV (R1 = 4)
END-SWITCH SET?
YES
RETURN
NO
STIMER, WAIT
VOREPREP (R1 = 0)
VEPEVCOM
FEDBAKEV
FEDFOREV (R1 = 0)
END-SWITCH SET?
YES
RETURN
NO
STIMER, WAIT
B
C
16 Outside active rectangle V to Inside active rectangle E calculation
17 Feed back from lower levels via IDMAT
18 Outside active rectangle E to Inside active rectangle V calculation
19 Feed back from lower levels via IDMAT
20 Loads and tests perimeter table against new contents of ENVN along the perimeter
21 Loads TESTAB from ENVN via addresses in CHANGTAB
22 Prepares ENVNTAB via IDMAT from solution table, ESV3
23 E to V computation at all levels via Pattern Matrix
24 (As for 17)
25 Feed V forward into lower levels via IDMAT
26 Halt this task for specified time to allow lower priorities computing time
27 Prepares ENVNTAB via IDMAT from solution table, ESVS
28 (As for 19)
29 V to E computation at all levels via Pattern Matrix
30 Feed E forward into lower levels via IDMAT
31 (As for 26)

Figure [illegible]

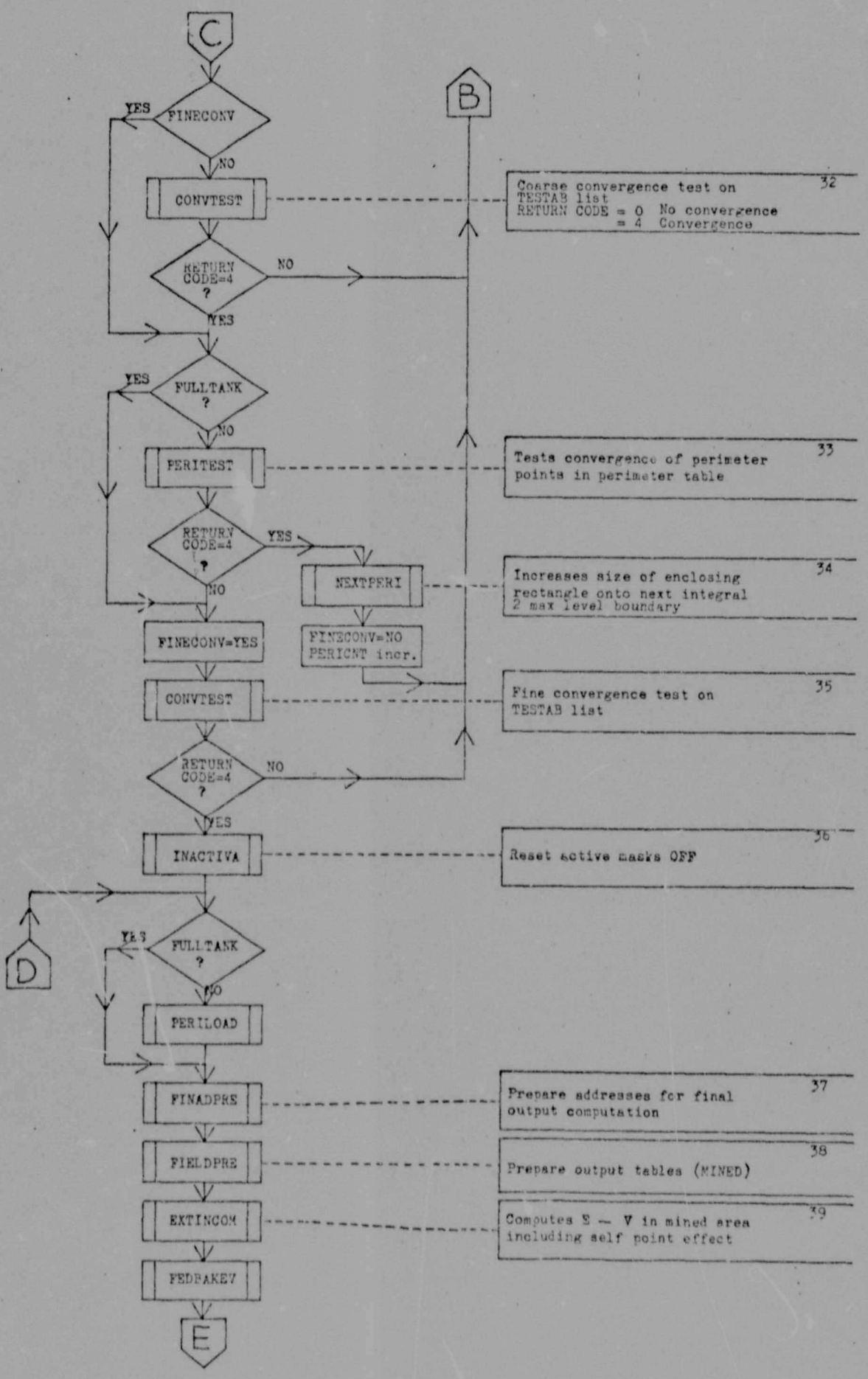


Figure 5.6 (continued)

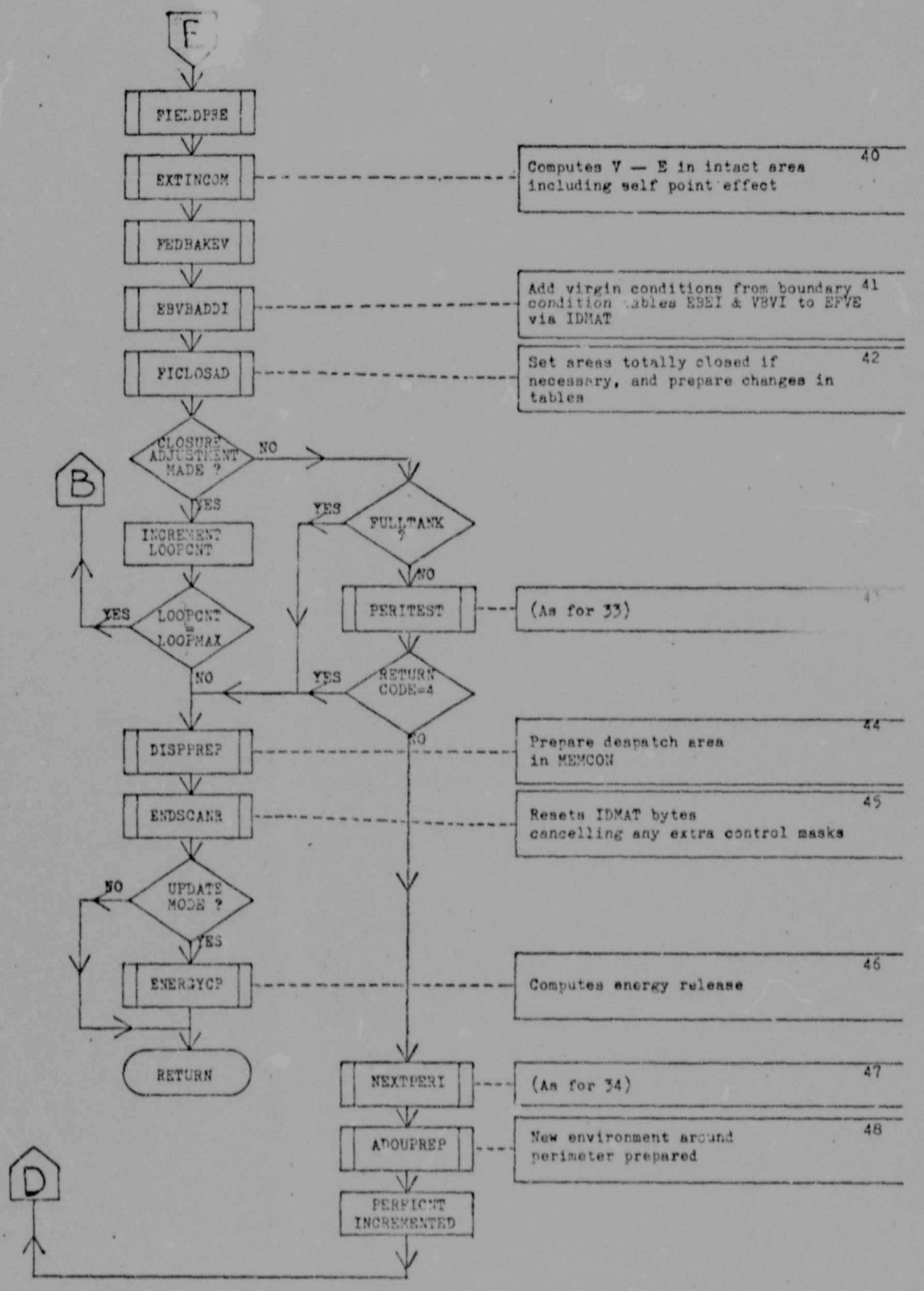


Figure 5.8 (concluded)

above, bits one and two represent a mixture of mined and intact conditions:-

e.g. B'11000000' = X'C0'

The above discussion also shows that the solution tables EN & VN may be combined into one table for level zero, since the mined and intact values are mutually exclusive. This will result in considerable saving in core since the zero level tables are 16 K-bytes each. Note that, at the lower levels, this does not apply, since mined and intact points may overlap. Thus, separate tables must be created at the lower levels. From the memory point of view, this is not so serious as the total core for the lower level tables is a maximum of (4K + 1K + 256 bytes) per table.

The boundary condition tables, VBVI and EBEI, are inserted into the EN and VN tables via IDMAT at all levels. In scale mode this has already been performed by the routine SCALPROC.

TESTPREP is then called to prepare TESTAB, the list of addresses of ENVNTAB level zero, corresponding to the co-ordinates in CHANGTAB set by TSTLST.

In update mode, CHANPREP performs the function of TESTPREP, but, in addition, updates and adjusts IDMAT and inserts boundary condition values into ENVN at the update addresses.

The next step is to establish an enclosing rectangle on an 8 boundary, (in general a $2^{\ell-1}$ boundary, where ℓ = maximum number of levels).

In FINISH mode all of the above is carried over from the previous state of the solution.

From this point onwards TANKSIMO performs a general solution inside the window created by CONVPERI. There is the same outside-upon-inside effect for the window, as discussed in the previous section, §5.3.2.3, using equations of the form (5.3) and (5.4).

Note that there are two convergence tests - FINE convergence, usually <4% change per iteration, and COARSE convergence, usually < 10% change. Once the solution has reached coarse convergence, the fine convergence switch is turned ON, and a test for fine convergence is performed on the perimeter of the window by PERITEST. If the test fails, the window is enlarged by NEXTPERI, PERICNT is incremented, and the solution is repeated to coarse convergence in the new window. If the perimeter test is successful, the 'test-points' are again scanned, this time for fine convergence, and, if successful, the final internal solution has been reached and the active masks are set to OFF.

Still working in the window (which in general may be the full tank), the final output calculation is performed by EXTINCOM. At this stage, with the first estimate of the final external solution computed, total closure is tested by FICLOSAD. The policy is that there is closure if the displacement has reached or exceeded the stoping width.

If there is no closure and the perimeter test of the external solution is passed, no further computing is required or, if the perimeter test fails, a new window is chosen for the external solution, etc.

If there is total closure, then formerly mined points are effectively treated as intact, with boundary conditions equal to the stoping width. These must then be set and IDMAT adjusted and a whole new solution re-started.

The routine ENERGYCP will be discussed in a later section.

5.3.4 Output Functions

5.3.4.1 On-Reef Output

Upon receiving the instruction "OU", CONTRL checks that there has been a solution of this configuration, or else the request is invalid. TANKSYS then calls the Fortran routine, FILDOP, and passes it to the addresses of the variables IDMAT, EFVF and some constants.

It may be shown that the principal stress component normal to the plane of the reef dipping at an angle α to the horizontal is

$$q_{Zo} = -\frac{q_o}{2}\left(1 + \frac{y}{H}\sin\alpha\right)\left((1 + k) + (1 - k)\cos 2\alpha\right) \quad \ldots\ldots\ldots (5.5)$$

where

$q_o = \gamma H = \gamma(Z_o - Z_s)/144$, (in imperial units),

γ = the specific weight of rock,

H = the depth of the top-left corner of the region under consideration, and

k = the lateral-to-vertical primitive stress ratio.

Note that the solution tables in MINSIM are in imperial units. The input and output routines adjust the input and/or output values to be in the requested units.

As mentioned in § 5.3.2.1, the boundary conditions were normalised to unity along the top left corner of the region; thus, all the stresses in the output solution must be multiplied by

$$q'_{Zo} = -\frac{q_o}{2}((1 + k) + (1 - k)\cos 2\alpha)$$

Similarly, it may be shown[7] that the scale factor for the convergence in the mined area is

$$K = -\frac{4(1 - \nu^2)}{E} q'_{Zo} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.6)$$

where

ν = Poisson's ratio, and

E = Young's elastic modulus

In addition, the distances used in the kernel tables assume a mine scale factor, (i.e. the scale of the real situation to the tank block), of unity. Thus, the convergences $S_z(x,y)$ are multiplied by K and the scale factor.

These constants are prepared in ABSIO during create mode. They are merely passed as parameters here.

The user may specify a list of discrete points by their window co-ordinates, for which this routine states whether this point is mined, intact or totally closed, and gives the corresponding total normal stress or convergence results. Note that a totally closed

point now has a stress component. Alternatively the user may request the results for the whole window, which are then tabulated row-by-row.

The total number of iterations, the number of perimeter increases during solution, the number of its increases during final output and the number of closure-estimate loops are also printed.

5.3.4.2 Graphic Display

The instruction "VE" may be given at any time, once the system has a configuration in core. TANKSYS calls the Fortran subroutine IDFULO which prints out a graphic display of the configuration.

5.3.4.3 Energy Release

It can be shown that the energy released due to mining may be calculated from the sum of the products of the stress components normal to the reef at a point before mining and the convergence due to mining.

As mentioned in § 5.3.2.2, the user is permitted to associate a group number with a group of updates. Immediately before TANKSIMO completes a solution, it calls the routine ENERGYCP to calculate the energy release per point.

Entering the command, "EN" into CONTRL causes TANKSYS to call the Fortran routine, ENEROP. This routine is passed a double word table, PREVTAB. These double words consist of two single ones, the first containing the group-number and the second containing the energy release for that point. ENEROP summates the total energy release per group and the number of points per group-update. It then publishes the total energy release, as well as the total area per group number. In addition, it prints the energy release rate. The units of the latter in the imperial system are ft. lbs./fathom, where the fathom is 36 sq.ft. (metric units are Joules/Centare). Sum-totals are also printed.

5.3.5 The Off-Reef Integrator

On the command DIsplacements and Stresses being given to CONTRL and suitable checks being made that a solution does exist,

TANKSYS calls the Fortran routine, REMOTE. This routine was adapted from an existing off-reef integrator program.

5.3.6 Adjustments

It is desirable to be able to alter various parameters in the system under certain conditions. This is performed by issuing the command PAtch. TANKSYS calls the Fortran routine - PATCH. A list of the constants the user is permitted to alter is published with the current value. The user may then change a particular parameter or parameters and the list is repeated with the new values.

The parameter names are :- FINCONLM, COSCONLM, GAMSLOP, GAMCONS, BETASLOP, BETACONS, LOOPMAX, GAMMAROK, EYOUNG, POISSON, PRIMSTRS, TOTCLOSE, ZTOPLEFT.

The original need for this facility arose out of the introduction of total closure to the system. Since, during solution, the estimate of the region of total closure is difficult and changed the whole configuration, the overall solution took many more iterations to converge than without closure. The parameter, LOOPMAX, forces a solution to stop after a certain number of closure perimeter adjustments. After examining the on-reef solution, patch could be invoked to change LOOPMAX and the FInish command issued, which would continue the solution where it had left off.

The user may also experiment with the rate of convergence and final accuracy by adjusting the first 4 parameters. As mentioned earlier, BETASLOP and BETACONS are not incorporated into the system.

The physical constants GAMMAROK to PRIMSTRS and ZTOPLEFT may be altered at this stage, as they affect only the output calculations - on-and-off reef.

5.3.7 The Storage Facility

It was considered desirable to be able to store all solution tables and constants of a problem on disk. The STore command followed by a unique 8 character name achieves this. TANKSYS calls the Assembler routine DMPRSTOR to do so.

The name of a stored solution appears in the directory of the partitioned data-set, whose members are these solutions. Additional information concerning the size of MEMCONTL for the solution is stored in the "user data" area of the directory.

On retrieving a solution, this user data is first read and a GETMAIN, for the size stored, issued. The base-address the computer GETMAIN has obtained for MEMCONTL is, in general, different from that used when the solution was stored. The difference in the old and new base-addresses is used to adjust all the relocatable addresses of tables in MEMCONTL. This is all performed by DMPRSTOR in REtrieve mode.

DMPRSTOR can also DElete members from the data-set.

5.4 The System in Practice

5.4.1 General Discussion

So far, little mention has been made of rempte 2741 terminal operation. In the initial studies and development this mode of operation was used exclusively. It is also useful in class demonstrations to students in the Department of Mining at the University. This department has a terminal with a 12 character/inch horizontal spacing, so that the graphic display which has spacers between mine-points would be 6 units per inch horizontally, and, as mentioned before, 6 vertically, thus printing an undeformed square display. Latterly, however, most users have favoured off-line mode via cards.

It was found that the number of iterations for a practical problem varies from 5 to 25 depending on the size and complexity of the mining outline and the extent of the total closure, which needs more iterations as the boundaries of the zone must be adjusted until the total normal field passes through zero at its boundary, i.e. the problem becomes non-linear. In practice, a full-size solution, hence, takes anything from 20 to 40 minutes. Mining updates, i.e. changes in configuration, may require a solution time of only a few minutes or as much as a basic solution. Each scale up takes about the same time as a basic solution, which amounts to a cost of R60 to R150. On-reef quantities can be calculated at about R0,30, and off-

Author Cohen B
Name of thesis Digital computer solution of special electrostatic and elastostatic problems with applications to the mining of tabular deposits 1975

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.