

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/

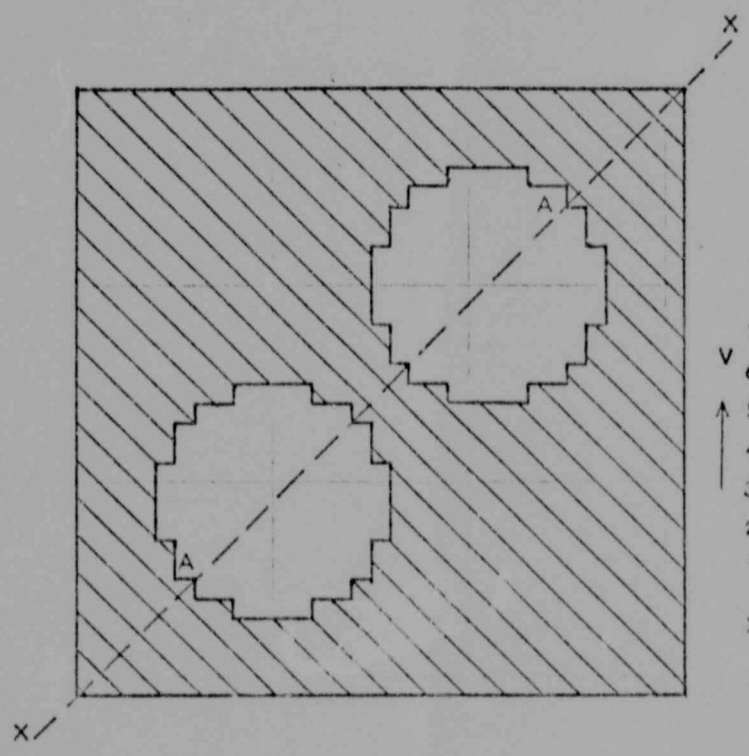

Figure 3.12 Two Circular Excavations

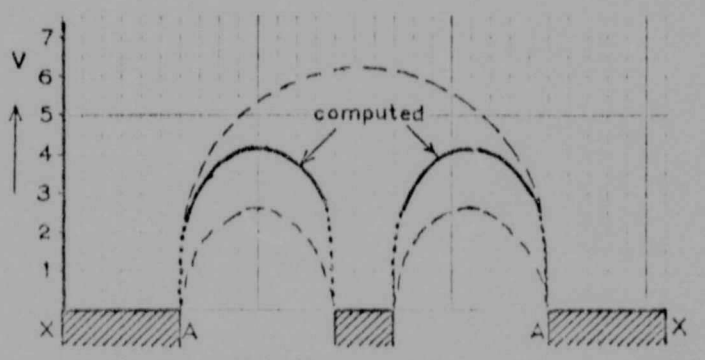

Figure 3.13 Graph of Voltage Distributions in Two Circular Excavations

3.3.4 General Comment

These results, especially those with theoretical backing (§3.3.2) show that, provided there is a sufficiently large rim of intact area around the excavations, the calculated results are very accurate and the iterative method is therefore satisfactory.

3.4 Iterative Convergence Rates

As mentioned in the previous section, the 21 x 21 problem of a circular excavation required 75 iterations, taking a total of about 3 hours to converge to a solution.

A full size problem (60 x 60) would take 70 times as long. Clearly this is impracticable. As a result, the values assumed by V and ΔE at each iteration were reviewed. The values of one such element is tabulated in Appendix VI and is shown graphically in Figure 3.14.

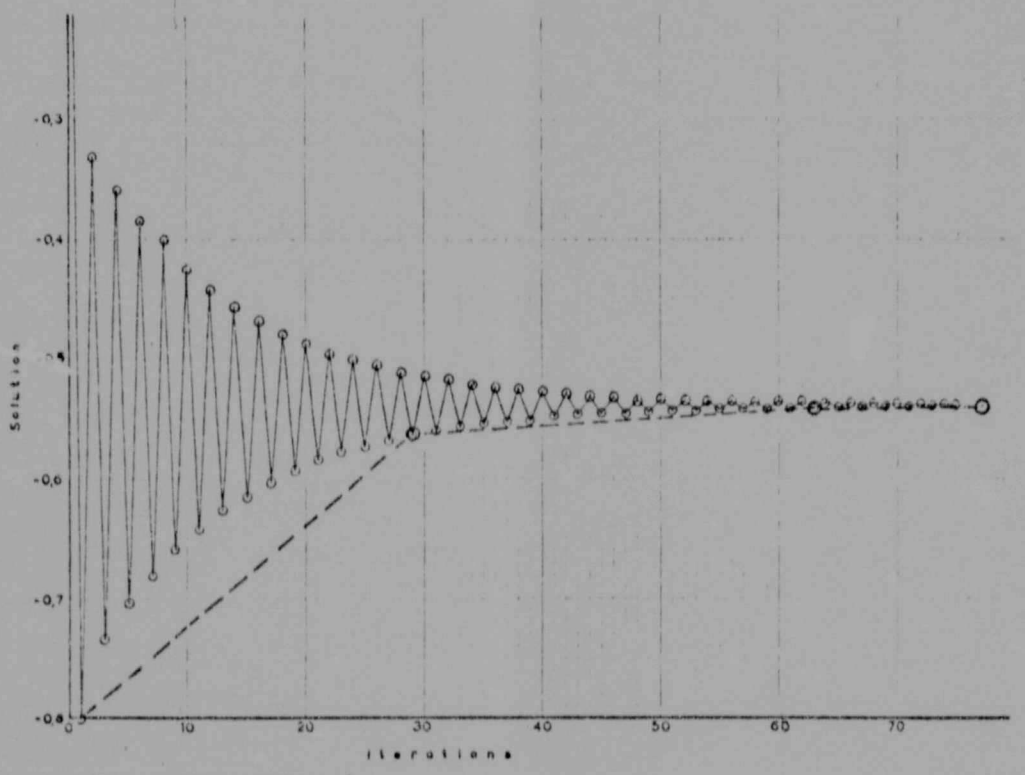

Figure 3.14 Oscillation of a Solution Value

If, for example, Figure 3.14, were the response of one point, in not an 11 x 11 plan, but in a 1 x 2 plan of one mined and one intact point then, having imposed an incorrect starting stress of 0,4, the elastic system over-reacts to this and pulls the result through the final solution to a value of -0,798, much like a pendulum. This value is too far the other way and so the solution is pulled back to its third value of -0,332, and so on in a damped oscillation, where the damping may be seen from the envelope of the iterative values in Figure 3.14. Now this exhibits a fairly underdamped situation so that, knowing two successive values assumed, an estimate of the final solution can be made. If there were virtually no damping at all, the estimate would be the average.

Consider the 2nd and 3rd values, -0,798 and -0,332, which have an average of -0,656. This is close to the value of iteration 29. Averaging this with value 30 gives 0,5405, which jumps to

approximately iteration number 63. Averaging once more gives 0,5388, which is better than at iteration number 75. Thus only 4 iterations would be needed.

This averaging was applied to the program and the solution emerged after 5 iterations. This shows that although the above reasoning will not, in general, apply to the actual situation where there are more than 2 points to consider, i.e. where the values will interact with one another, some sort of averaging is necessary, and that the problem should be examined further.

In general, the iterative scheme of equations (3.4) can be modified to the following form :-

$$V_{i+1} = \frac{\beta_i}{1 + \beta_i} V_i + \frac{1}{1 + \beta_i} (V_b - K_1 E_i) \quad \text{.....(3.8a)}$$

$$E_{i+1} = \frac{\gamma_i}{1 + \gamma_i} E_i + \frac{1}{1 + \gamma_i} (E_b - K_2 V_{i+1}) \quad \text{.....(3.8b)}$$

Note that if $\beta_i = 0$ and γ_i = constant = 1, where i is the iteration count, this is the averaging as described above. If $\gamma_i = 0$ also, then the equations revert to equations (3.4). The scheme of equations (3.8), with $\beta_i = \gamma_i$ = constant, as a result of the two - cyclic nature of the procedure, reduces to the Gauss-Seidel method, with an over-relaxation factor $w = \frac{1}{1 + \gamma}$. This suggests a dependence on the eigenvalues of the system, which is dealt with more fully in Appendix VII.

3.4.1 Some Experimental Results

In order to ascertain how sensitive the system was to β or γ, some experiments were performed. In order to simplify matters as mentioned in Appendix VII, let $\beta = 0$.

On a maximum plan size of 31 x 31, the problems as illustrated in Figures 3.12 and 3.15 are solved.

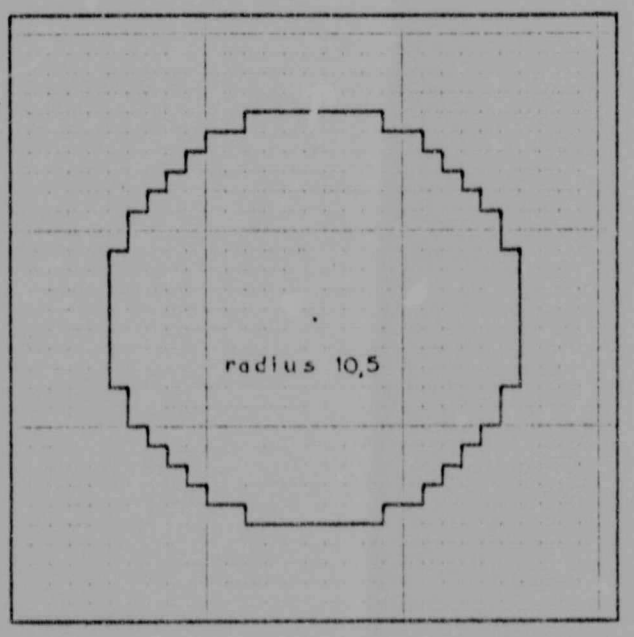

Figure 3.15 Large Circular Excavation

The results of the number of iterations vs. γ , to reach a solution having an error criterion of < 1% change, are tabulated in Table 3.3, and a graph is plotted in Figure 3.16.

Table 3.3(a) Circular Excavation Results

γ	No. of Iterations
0,9	5
1,0	5
1,1	4
1,3	5
1,5	6

Table 3.3(b) 2 - Circular Excavation Results

γ	No. of Iterations
0,7	11
0,9	8
1,0	7
1,025	6
1,050	6
1,075	6
1,1	7
1,2	7
1,3	8
1,5	8

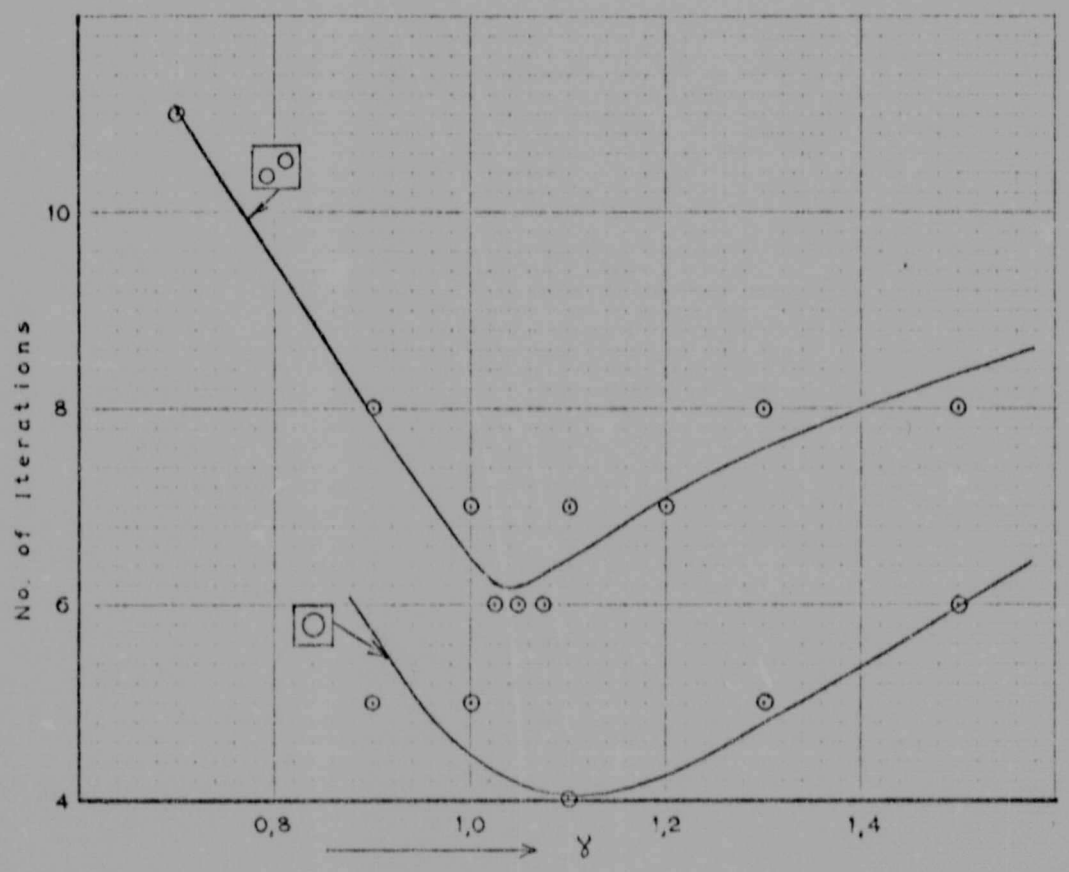

Figure 3.16 Stability vs. γ

The slope of the curves for values of $\gamma<\gamma$ opt. is steeper than if γ is too big. Thus it would appear better to estimate a larger value of γ than one would guess for the optimum.

3.5 Conclusions

The close agreement between these results and theory, especially for the circular excavation, shows that the iterative method is accurate. However, the solution times are ridiculous, especially when a full size 60 x 60 plan is envisaged.

Therefore, one must concentrate on methods of speeding up the solution, without too much sacrifice in accuracy. Also some automatic choice of the over-relaxation factor should be made so as to minimise intervention by the user.

CHAPTER 4

MINSIM AND ITS "SPEED-UP" TECHNIQUES

4.1 Introduction

Minsim is the name coined for the full working version of the system, It stems from the first syllables of MINing SIMulation.

The description of the system will proceed in two parts. Since the last chapter stressed the need for programming techniques to speed up a solution, this chapter will deal with these aspects of MINSIM. Chapter 5 will analyse the various functions and features of the system.

4.2 Reducing the Number of Effect Calculations per Iteration

4.2.1 Grouping of Points

Since for a configuration of side n units, there are, on average, $\frac{n^2}{2}$ mined points affecting $\frac{n^2}{2}$ intact points and vice versa, there are $2 \times \frac{n^2}{2} \times \frac{n^2}{2} = \frac{n^4}{2}$ effect calculations per iteration, i.e. the number of calculations is proportional to n^4. It would be desirable to reduce this figure by perhaps trading accuracy for running time.

Consider the form of the influence kernels. They reduce as $\frac{1}{r}$ and $\frac{1}{r^3}$ with distance, r. Consider the kernel which reduces slowest with distance, $\frac{1}{r}$, as seen in Figure 4.1 or, more importantly, consider its slope $(-\frac{1}{r^2})$.

The latter tends to zero very quickly, having at 10 units distance only 1% of its value at 1 unit distance. Thus the kernel $\propto \frac{1}{r}$, at this distance, is not varying by more than 1% per unit distance. It is fairly linear as the slope is not changing rapidly, the first derivative being 0,001. If points at distances of 9 and 10 units were grouped together and given an average value at an average distance of 9,5 units, the errors induced would be small.

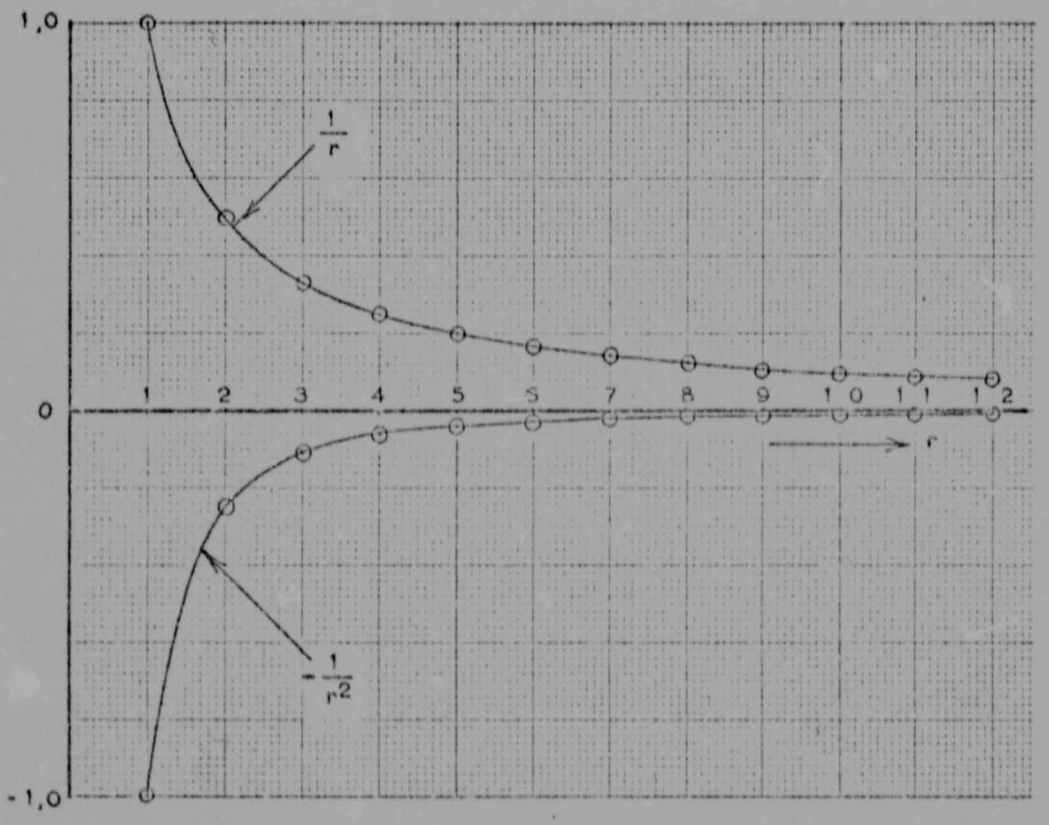

Figure 4.1 Plot of $\frac{1}{r}$ and $-\frac{1}{r^2}$ vs. r

Consider the situation of Figure 4.2. The points 1, 2, 3, 4 are beyond the distance 8. Some or all of them may be valid 'from-points', i.e. mined if ◎ is intact. Considering unit kernel amplitudes on these points, they can have one of the configurations shown in Table 4.1. Below each configuration is tabulated the total effect at point ◎, and the corresponding effect evaluating all from-points at an average distance of 9,5 units. Below this are grouped those configurations which are equally likely to occur based on a randomly-shaped border cutting across the 4 from-points, and the % error of the approximation.

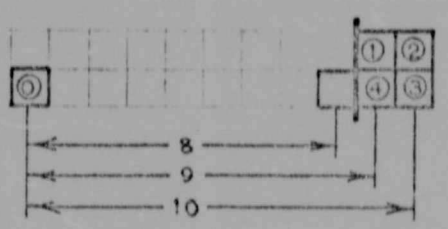

Figure 4.2 Grouped Influence Points

Table 4.1 Grouped Average Errors

Configuration	1 1 1 1	10 01 or 01 10	1 0 1 0	0 1 0 1	1 1 1 0	1 1 0 1
$\frac{1}{1}$ effect	0,4222	0,2111	0,2222	0,2000	0,3222	0,3111
$\frac{1}{(\text{average r})}$ effect	0,4210	0,2105	0,2105	0,2105	0,3158	0,3158
$\frac{1}{r}$ grouped average			0,2111		0,3167	
$\frac{1}{\text{average r}}$ " "			0,2105		0,3158	
error	0,28%	0,28%	0,28%		0,27%	

These % errors are relative to accurately taken values at distances of 9 and 10 units. The % errors become still smaller when taken relative to the effect calculations closer to point ◎, reducing by a factor of 10 in the limit. The influence range, the distance beyond which this type of averaging can take place, was chosen as 8. Tests were performed to justify the conclusion that this influence gives sufficient accuracy (Appendix IX).

The mechanism is as follows :-

The calculations are performed at the finest scale up till a range of about 8 (the exact influence boundary will be discussed in the next section), bounded by ABC in Figure 4.3; this is performed for all the to-points; thereafter, calculations of exactly the same nature are performed in another table, a lower level table, level '1', which considers to-points of the type ODEF, i.e. a single to-point representing 4 to-points at the next higher level, level '0'. The effect calculations are then performed in the region ABC to GHI. This is done for all the to-points of this type, and so on, up to a maximum level of 3.

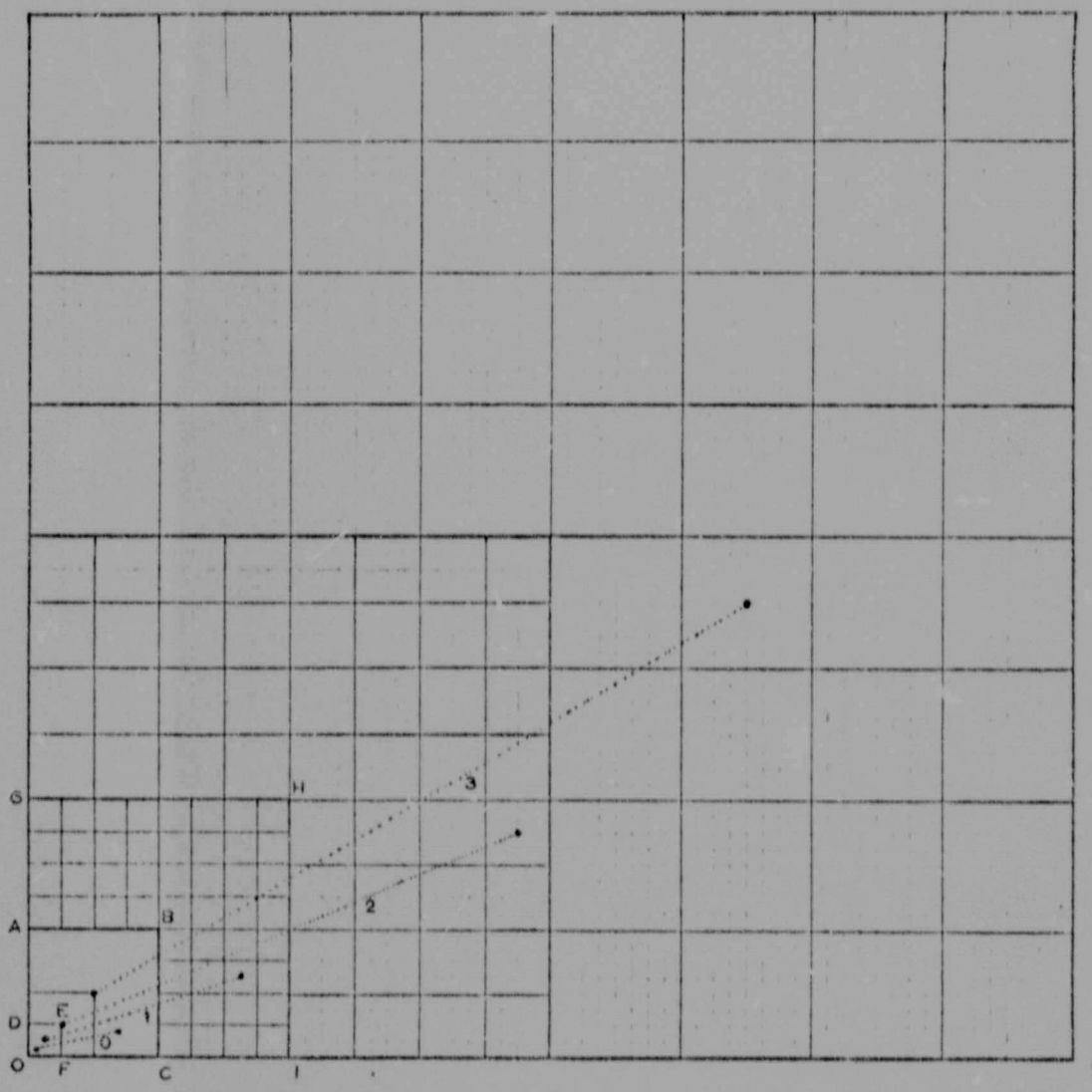

Figure 4.3 Influence Points at all Levels

When this is complete, the effects are rippled up the tables from the lowest level to the highest, each to-point averaging into the group of 4 to-points of the next higher stage as illustrated in Figure 4.4.

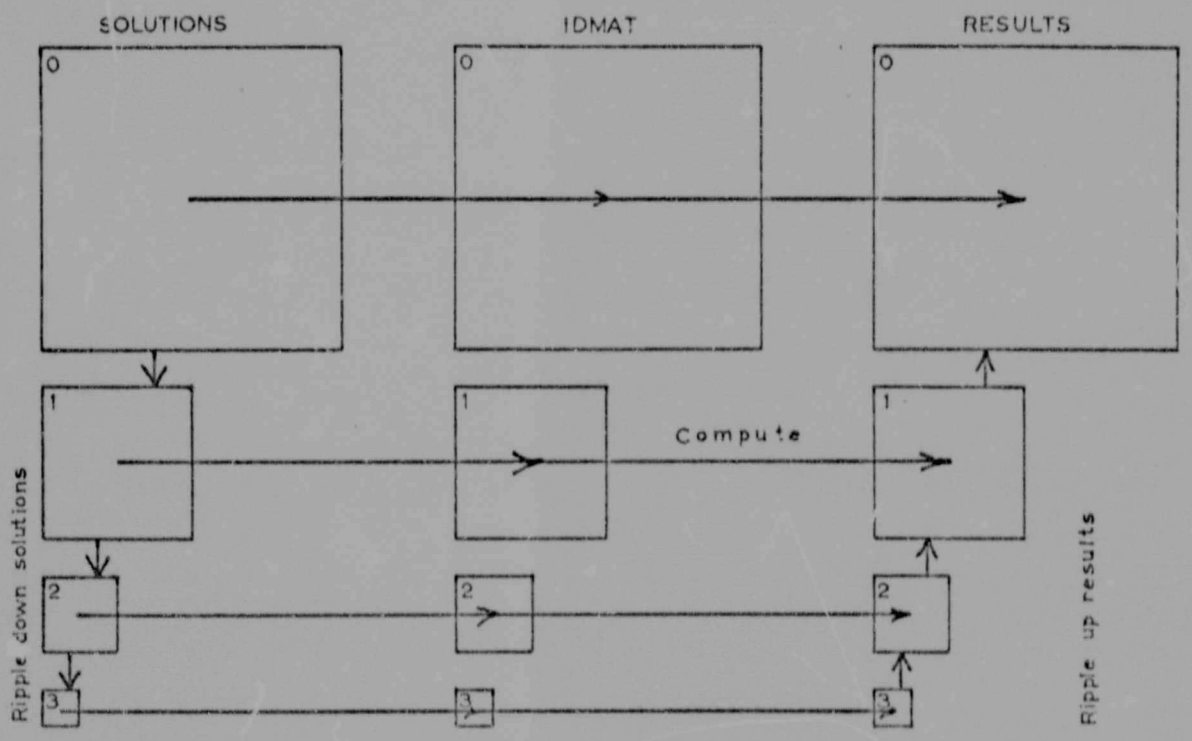

Figure 4.4 Use of Levels in Solution Phase

4.2.2 Table Level Structure

Since 4 levels, each having squares with twice the linear dimension of the next higher level, were chosen, level 3 had a side corresponding to 8 units at the finest scale. The maximum tanksize[1] must be a multiple of 8 (or 2^{ℓ} (where ℓ = maximum level)). A maximum size of 64 units was chosen in order to give an accuracy as good as, or better than, the electrolytic tank analogue.

In order for the table boundaries at the different levels to coincide, the following scheme had to be developed :-

Since 4 points on the finest scale correspond to one point in the next level, consider the 4 points 1, 2, 3, 4 in Figure 4.5(a) as a group, having a common range of influence, a. To ensure a proper termination, point 1 must have a range of 'a' up and to the left, and (a + 1) down and to the right. In order to end on an even boundary, 'a' must be even. As mentioned earlier, a = 8 was found most suitable.

Up to a range of a or (a + 1) as indicated, the effects are treated on the finest scale as if no grouping occurred. Beyond this

---

[1] The memory structure will often be referred to as the 'tank' because of its analogy with the electrolytic tank which has been constructed by Rand Mines Limited.

boundary, the table at the next level is used, e.g. level 1. Points 1, 2, 3, 4 on level 0 are represented by Point 1 in Figure 4.5(b) on level 1. Since the shaded area has already been taken into account

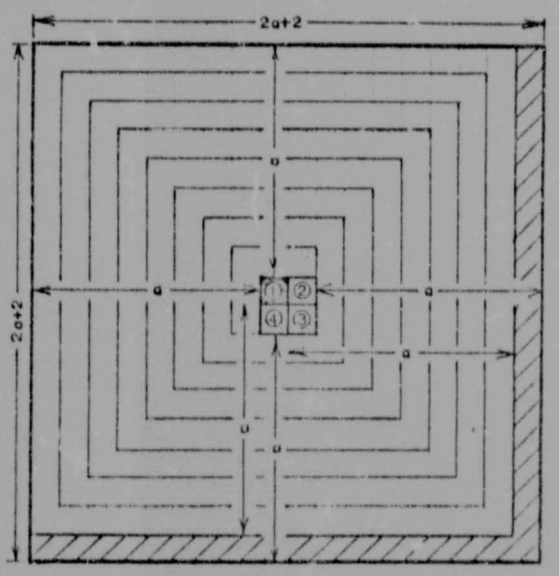

Figure 4.5(a) Level 0 Influence Area

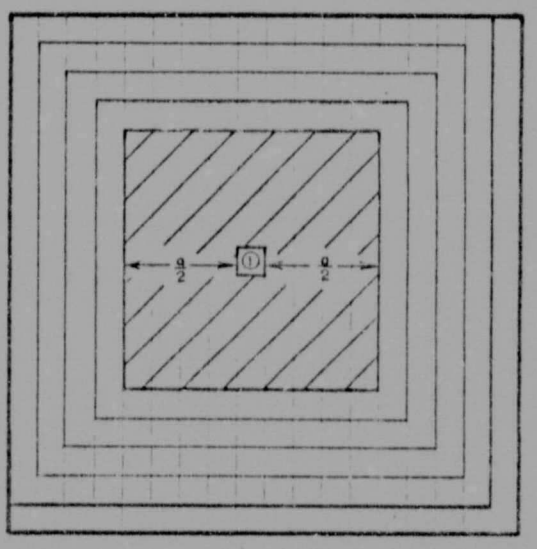

Figure 4.5(b) Level 1 Influence Area

by calculations at level 0, computation at level 1 begins beyond $\frac{a}{2}$ to the a or a + 1 boundary as before. The process continues up to level 3 and, while a more detailed description will be given in the discussion of the solution phase in Chapter 5, here it suffices to indicate the reduction in the number of effect calculations.

Consider as a typical example, the 'to-point' as shown in Figure 4.6. There are on average $\frac{323}{2}$ 'from-points' of the correct type for the finest level, $\frac{243}{2}$ for the next level and $\frac{115}{2}$ at the 3rd level. For the finest level there are on average $\frac{n^2}{2}$ 'to-points', where n is the length of a side of the configuration. For the first level the total is $\frac{1}{2}(\frac{n}{2})^2 = \frac{1}{2} \cdot \frac{n^2}{4}$, and, for the third, $\frac{1}{2}(\frac{n}{4})^2 = \frac{1}{2} \frac{n^2}{16}$. For a complete iteration there is a mined-to-intact and intact-to-mined iteration. Hence the total number of calculations is approximately $\frac{1}{2}$ (323 n^2 + 243 $\frac{n^2}{4}$ + 115 $\frac{n^2}{16}$ + a negligible amount). This simplifies to approximately 200 n^2 computations, compared with $\frac{n^4}{2}$ calculated earlier. For n = 64 the comparison is 820 x 10^3 calculations as

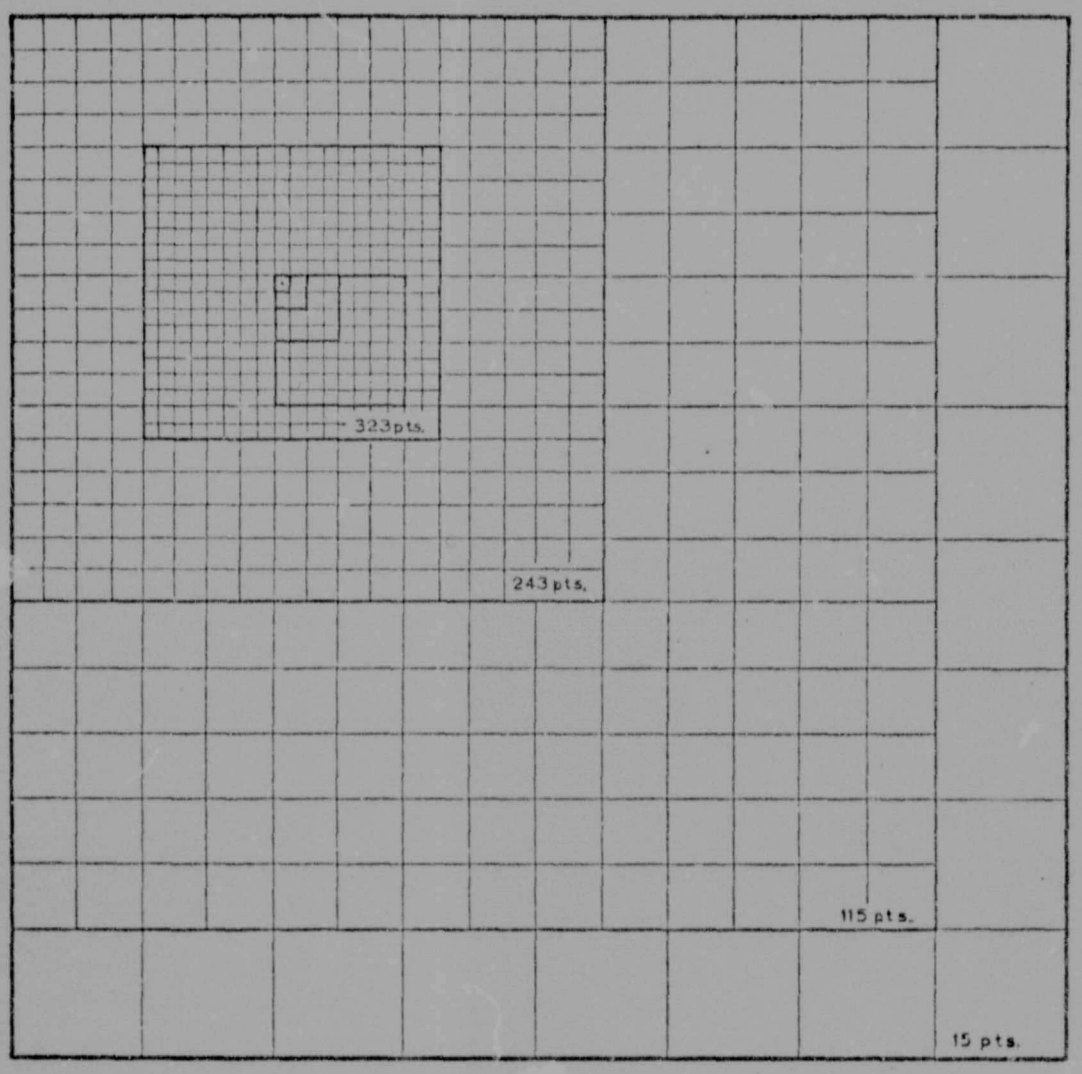

Figure 4.6 'From'-points at all Levels (64 x 64 grid)

against $16,8 \times 10^6$ calculations without grouping of points. Therefore, one can expect the improvement factor due to this method to be of the order of 20 for a 64 x 64 mesh solution.

4.2.3 Solution Windows

If the regions of mining are small in comparison with the intact area, or if a small mining update or change is made to an existing solution then, because of the nature of the fast decay of the kernels with distance, it is obvious that changes to the solution

will be largest close to these regions of mining. A method was evolved to minimise the number of computations by initially only forming a solution in a rectangular window enclosing the mined regions or, in the case of updating, enclosing only the updated region, as shown in an example in Figure 4.7.

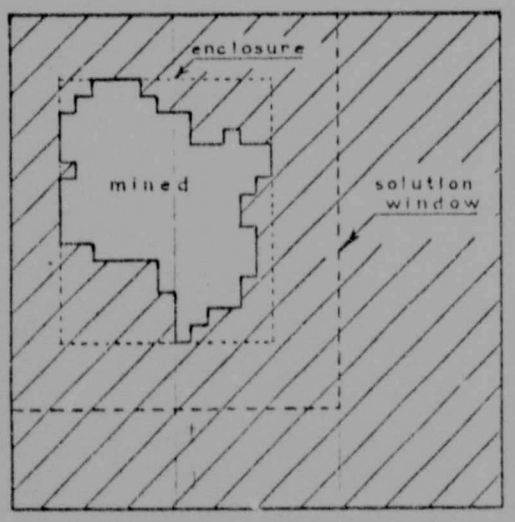

Figure 4.7(a) Initial Solution Windows

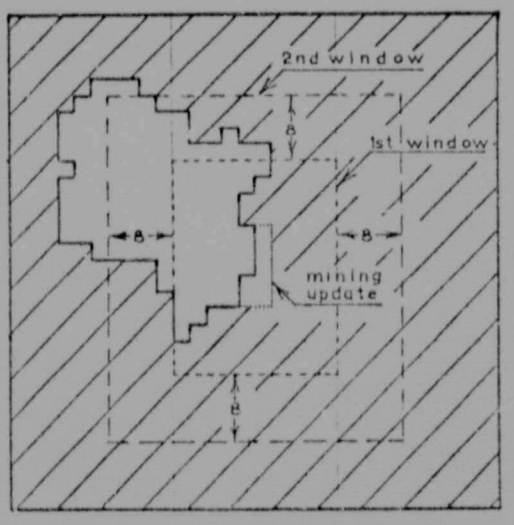

Figure 4.7(b) Update Solution Windows

Obviously each iteration is now considerably faster as both the number of to-points and the number of from-points have been reduced. Once a solution has been obtained according to the convergence criterion for this window, the solution on its perimeter is compared with the old solution and, if the convergence criterion is satisfied, it is considered that the effect inside the window will not progress to regions outside it. If the criterion is not met, however, the window is increased by an influence range, e.g. 8 units, and the system re-enters the solution phase with this new window and perimeter. This algorithm is repeated until the maximum solution window is the specified size of the problem, e.g. 64 x 64. In this way the computing times especially for updates were considerably reduced.

4.3 Reducing the Number of Influence Tests

Consider the region of influence around a to-point as shown in Figure 4.5(a). For each iteration, each from-point must be tested for type, e.g. if the to-point is mined, from-points must be intact for internal solutions. This would require, for the finest level in the structure, 323 tests per to-point per iteration. Note that a test requires computing the effect in the identity table (IDMAT), which keeps the mined / intact information of the whole area, followed by the type-test.

The method used in MINSIM to reduce the amount of testing is via a 'pattern' matrix which is constructed once per solution and contains information about each point as described below.

Imagine the influence area around a to-point to be divided into regions as shown in Figure 4.8 for a to-point of type 1, (the significance of which may be seen in Figure 4.5(a)).

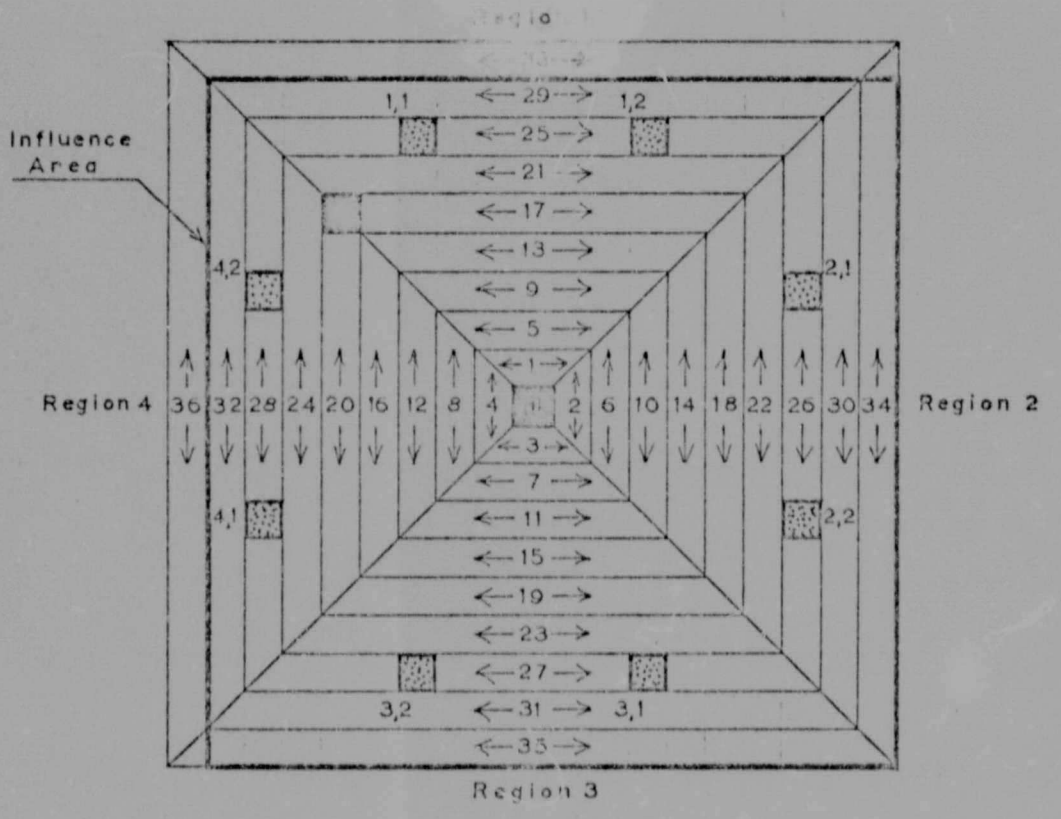

Figure 4.8 Regions for Pattern-Matrix Generation

In the pattern matrix the pattern of one to-point is a string of bits. Each bit position represents information about a row in each region, as illustrated by the numbers and arrows in Figure 4.8. If a particular bit is ON, (i.e. a 1), it means there is a valid from-point, i.e. of the correct type, somewhere in the row represented by that position. In the example shown in Figure 4.8, bit positions 33 and 36 are OFF, (i.e. 0), since for a to-point of type 1 these rows are outside the influence area, shown by thick lines; for an influence range of 8, as shown, one needs 36 bits, i.e. more than a full word (32 bits on IBM 360). Thus double words (64 bits), were used in the pattern matrix table. It can be shown that the maximum influence range that could be handled is 14.

Considering an influence range of 8, there will be 36 row-tests. In the quadrants there will be a total of 4 x (3 + 5 + 7 + 9 + 11 + 13 + 15 + 17) + (2 x 19) i.e. 358, (more than before, since from-points bordering regions, e.g. darkened square shown between regions 1 and 4 are tested twice). However, since on average the row-tests will exclude half of these, there are about 179 of these tests, i.e. slightly more than half the number obtained without using patterns. (The 36 row tests are neglected here, since the time taken in a pattern test is far less than for a from-point test; this is because a pattern test involves a pattern shift and a test, whereas a from-point test in addition requires a more time-consuming table offset calculation.

4.4 Reducing the Number of Computations for an Effect Calculation

As was noted in Chapter 3, § 3.12, since the kernels are functions of $r = \sqrt{x^2 + y^2}$, i.e. the distance between the from-points and the to-points, they exhibit 8-way symmetry with respect to x and y. For example, the points (1,1; 1,2; 2,1; 2,2; 3,1; 3,2; 4,1; 4,2) in Figure 4.8 are all at the same distance, r, from the to-point. Thus, instead of an average of 4 calculations of references to the kernel table and 4 multiplications and 4 additions, which would be needed if the symmetries were not taken into account, one can arrange for the

4 additions of the from-point values, one kernel-offset calculation and one multiplication. On the IBM 360/50, a floating point multiply takes 29 μs, and a floating point addition takes 16 μs. Making use of symmetries can thus be seen to have considerable advantages.

In Chapter 3, §3.12, the kernel table offset computation involved the calculation of the form $j(j + 1)/2 + i$, (equation 3.7), which was estimated to take approximately 100 μs. Because of the ordered method of scanning through from-points as described in this section, such a computation is not necessary as scanning of the kernel table is ordered in the same fashion. This involves merely fixed point additions having a computation time of approximately 12 μs.

4.5 Table Addressing

The effect computation is illustrated in Figure 4.9.

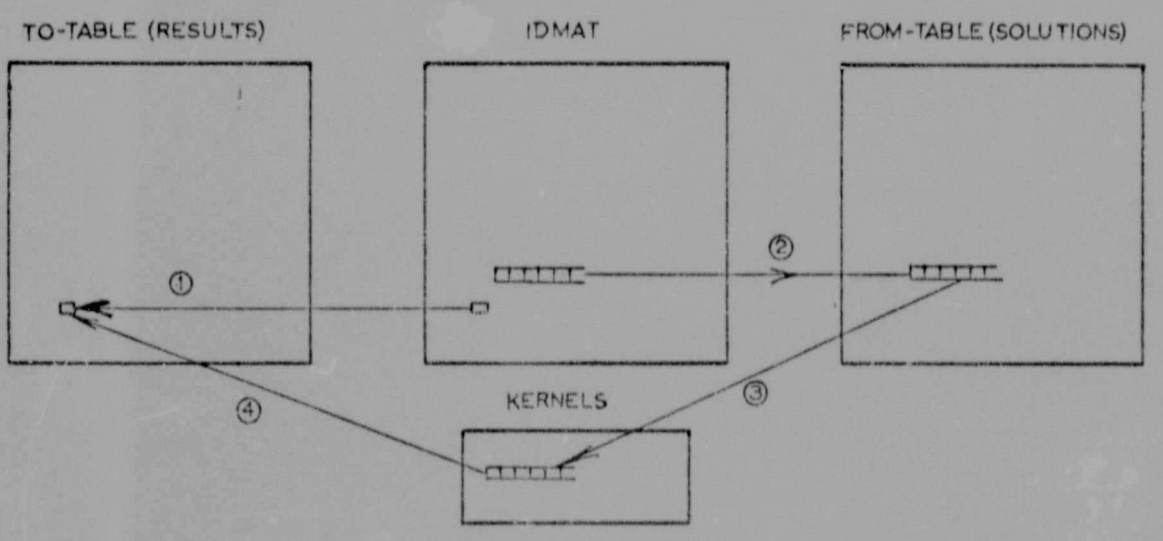

Figure 4.9 Order of Table Addressing - Effect Computation

The table, IDMAT, carries the information of whether a point considered is intact or mined. It is used to select a to-point. As described in the previous section, from-points are selected via IDMAT, and, in combination with the kernels, a result is computed.

When this has been performed at all levels, the old solutions at to-points, as selected by IDMAT, are combined with the results and boundary conditions to form the new solutions, as illustrated in Figure 4.10.

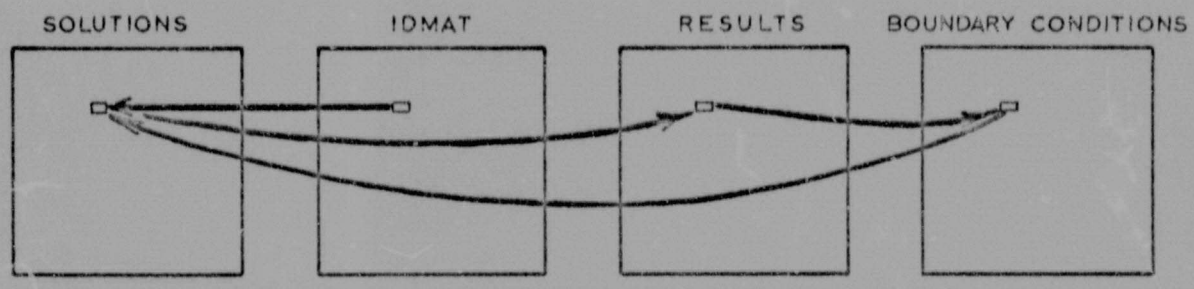

Figure 4.10 Order of Table Addressing - Combination

The entries in the IDMAT table do not carry much information. Only the information MINED or INTACT and, as shall be seen later, totally closed, need to be stored. Thus, in principle, only 2 bits are required. However, if the data were compressed in this fashion, a separate computation would be necessary to calculate the offsets in order to reference the IDMAT entries. Since the other tables (except the kernel table, which is referenced differently, and the pattern table, which is referenced only once per to-point), are of the same size and may be referenced by the same offset in an index register, it becomes obvious that the same register should be used to reference IDMAT, i.e. IDMAT should also be a table of full-words (32 bits). The consequent waste of CORE must thus be sacrificed to save computing time.

4.6 Overlays

For a full size problem of 64 x 64 elements, the program and all the tables in core require 350 K. This was not considered extravagant, since the computer available had a maximum of 850 K of user space with a practical multiprogramming maximum partition size of 450 K; it was thus not considered worthwhile to attempt to overlay

program segments and tables in order to save memory space at the expense of a time overhead.

4.7 Conclusions

The techniques mentioned in this chapter allowed the solutions of full size problems, (64 x 64 elements, of which the user specified the problem in a central 56 x 56 window), to a reasonable accuracy, (4% relative convergence criterion), in a reasonable computing time. For instance, a typically complex problem needing 10 to 14 iterations takes from 30 to 45 minutes of 360/50 LCS time, thus achieving the desired reduction in computing time.

CHAPTER 5

MINSIM I - FEATURES AND FUNCTIONS

5.1 Introduction

As mentioned in Chapter 4, the purpose of MINSIM is to be a useful mining simulator incorporating the model tested by the feasibility programs and reducing its computing time.

The system can cope with moderately inclined deep tabular excavations in one plane. This chapter will deal with the adjustments to the theory in order to take small inclinations into account as well as the additional theory needed in connection with other features.

The introductory chapter gives as the primary object of the system, the avoidance of the cumbersome transfer of convergence data from the tank or resistance analogue to a digital computer for the calculation of off-reef plane quantities, especially after each change of mining configuration. This has been achieved and will be described here.

In order to achieve man/machine communication similar to that of the analogues, a computer terminal on-line operation mode can be used where the whole configuration and changes to and interrogations of the system can be an interactive process. This enables the user to make "on-the-spot" decisions and changes to the configuration.

If no on-line communication is desired, the whole problem including planned updates to the configuration can be performed off-line from cards.

In order to facilitate changes and interrogations of a solution, provision was made to store computer configurations which can be recalled at any time.

The scaling phase allows one to model detailed small scale mining activities, while still taking into account the effect of the whole environment.

The overall system and its particular sub-systems will now be dealt with individually.

5.2 System Organisation

The general organisation of the computer system is shown in Figure 5.1. In general, the parts of the program being strongly compute-limited, were written in IBM/360 ASSEMBLER, whereas those parts involved in input/output communications were written in IBM/360 FULL FORTRAN IV, using the FORTRAN G or FORTRAN H compilers. (FORTRAN H was used where there was some input/output and a moderate amount of computing, e.g. for the off-reef Integrator, since it has a speed advantage of ∿3 over the machine-code generated by the FORTRAN G compiler.

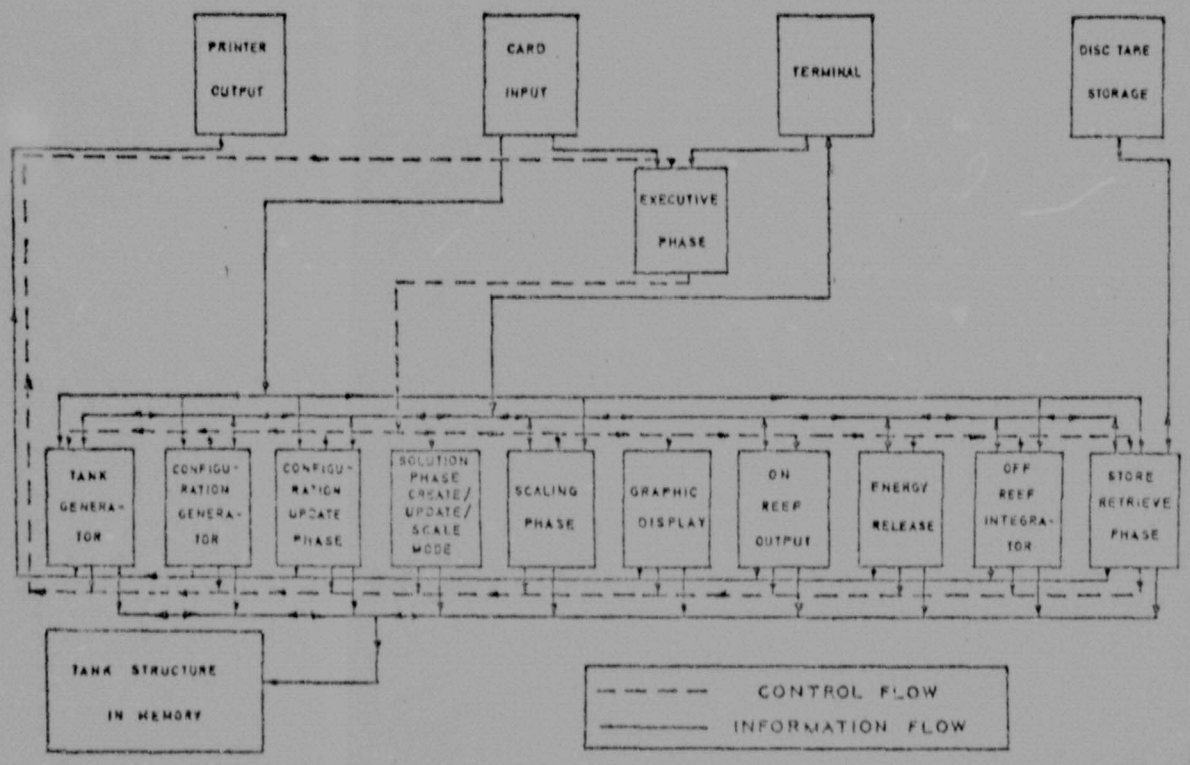

Figure 5.1 Organisation Diagram of Computer System

5.2.1 The Executive Phase

This is handled by the main program - TANKSYS, the tank-system main despatcher, and an input/output handling routine - CONTRL - which handles the verbs or commands of the user and traps invalid requests. (For example, when starting up the system, all commands except CREATE and RETRIEVE are invalid). On completion of any one

of the functional phases, control is returned to the executive phase.

5.2.2 The Functional Phase

The various functional requests of the user are entered via the CONTRL routine. The commands or verbs are in mnemonic form, being the first two characters of the possibilities listed alphabetically below :

CReate	- the new problems are entered- all constants relating to the new problem are entered together with the configuration or mine plan. With reference to Figure 5.1 this involves the TANK GENERATOR and the CONFIGURATION GENERATOR.
DElete	- destroying one of the stored solutions
DIsplacements and Stresses	- the OFF-REEF INTEGRATOR
ENergy	- release
EXit	- closing down the system
FInish	- solution after some parameter change in a PA.
OUtput	- the ON-REEF OUTPUT
PAtch	- changing of certain parameters of the system
REtrieve	- one of the stored solutions
SCaling	- by a factor of 2
SOlution	- after CR, UP or SC
(SNap	- a memory dump used in 'debugging' the system)
SToring	- a solution under a given name
UPdating	- mining configuration
VErifying	- a GRAPHICAL DISPLAY of the mine plan
WAit	- except when verifying, one must wait until a solution in progress is finished before further commands are issued.

Each of these functions will be dealt with in turn in §5.3.

5.2.3 General Program Organisation

The main program - TANKSYS - communicates with the called sub-routines in two ways :-

(i) Parameter lists are used in communicating with the Fortran sub-routines, and

(ii) A common area, the DSECT MEMCONTL is used in communications between most Assembler programs. The main program issues a GETMAIN instruction, requesting memory space dependent on the size of the "tank", as entered in create mode. The address of MEMCONTL is kept permanently in one of the computer's general registers, (Register 12).

The DSECT MEMCONTL consists of two parts :-

(a) A parameter area, where the constants of the system, the data read in at create time, etc., and the lengths and start-addresses of the various tables, are stored.

(b) A table work area set up by the routine TABLEGEN.

Since the flow-chart of TANKSYS closely resembles Figure 5.1, this will not be given, but a listing of TANKSYS, MEMCONTL and CONTRL are given in Appendix VII, out of interest.

The ATTACH macro is used in the solution phase in order that this may be multi-processed at a lower priority than the main program. This allows a function like VErify to be performed simultaneously since, when using the terminal mode of operation, the verify process is very time-consuming, taking 20 minutes for a full tank display, mostly for output typing which does not interfere with the main solution.

5.3 MINSIM Operation

5.3.1 General

As illustrated in Figure 5.1, after execution of any of the functional phases, control always returns to the executive phase and,

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.