Computer programs for the two regime flow towards wells.

Author:
Huyakorn, Pongsarl

Publication Date:
1974

DOI:
https://doi.org/10.26190/unsworks/5309

COMPUTER PROGRAMS FOR SOLVING TWO-REGIME FLOW TOWARDS WELLS

P. HUYAKORN

Name/contact details	Date consulted	Signature

COMPUTER PROGRAMS FOR SOLVING TWO-REGIME FLOW TOWARDS WELLS

by

Pongsarl Huyakorn, B.E. (Hons.)

A volume supplementary to the Ph.D. thesis "Finite Element Solution of Two-regime Flow towards Wells"

February, 1974.

To

COL DUDGEON and DOUG FOSTER

for their help, patience and understanding
during the course of my Ph.D. work.

ACKNOWLEDGEMENTS

The author is indebted to Mr. C.R. Dudgeon for his supervision and guidance on the preparation of this supplementary volume of the thesis. The author would also like to express his gratitude for the financial support provided by the Australian Water Resources Council through its funding of Project 71/25, "Drilling and Development in Unconsolidated Sediments".

Particular thanks are also given to Mr. C. Doney of the University of New South Wales Computer Centre, who assisted in obtaining the program listings and made available the large number of computer runs necessary for this work.

Finally, thanks are due to Mrs. P.M. Auld and Mrs. S.E. Dion for their effort in tracing the figures and typing the manuscript.

SUMMARY

This report contains descriptions and listings of eight finite element computer programs which are written in FORTRAN IV language to simulate either Darcy or two-regime flow of groundwater towards a pumped well. Included are instructions for assembling the input data deck, sample input data for test problems and complete printouts of computed results.

A wide range of different aquifer conditions and well constructions has been considered (e.g. single-layer or multi-layer aquifers, confined or water table aquifers, fully or partially screened wells with or without gravel packs and with one or more screened intervals). The flow can be either transient or steady state. Each program has been designed to handle specific flow cases encountered in practice with a minimum of input data and to operate with any consistent set of units.

The programs have practical application in both aquifer test evaluation and hydraulic design of pumped wells. Although written specifically to handle groundwater flow problems, these programs can readily be adapted to deal with flow of other types of incompressible fluid towards a pumped well.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd.)

1. INTRODUCTION

Well-flow problems of considerable complexity can now be analysed with the aid of modern numerical techniques and digital computers. Partial penetration, partial screening, gravel packing and the occurrence of non-Darcy flow can all be taken into account in steady or transient flow towards wells in single or multilayer, confined or unconfined aquifers. Whether such an analysis is an economic proposition depends largely on the savings which can be made by refining well design, the cost of computing and the complexity of the problem.

During the course of a research project into the extraction of water from unconsolidated sediments funded by the Australian Water Resources Council, eight computer programs were developed to allow the effects of various well and aquifer parameters on flow near wells to be studied. These programs, written in FORTRAN IV language, use the finite element method described in Reference 3 to solve a number of flow cases of interest to those engaged in well design or evaluation of local aquifer characteristics. The presence of non-Darcy flow near the well can be taken into account since the Forchheimer equation is used instead of the Darcy equation for the velocity-hydraulic gradient relationship when Darcy's law is violated near the well. The term "two-regime flow" (Reference 3) is used to denote flow which includes both Darcy and non-Darcy regimes.

The programs included in this report have been designed and documented in such a way as to allow their use by personnel with a limited knowledge of numerical methods and computer programming. Documentation of the programs has followed as far as possible the recommendations of the A.S.C.E. Subcommittee on Program Documentation of the Committee on Computer Applications of the Soil Mechanics and Foundations Divisions (Reference 6).

Source decks can be made available if required. Comments on any problems met in use or suggestions for improvements will be welcomed by the author. Updating of the programs will be carried out in the light of such comments and suggestions. To date the programs have been run successfully on IBM 360/50 and CDC 6600 computers for a wide range of values of well and aquifer variables.

It is hoped eventually to develop an Integrated Well-Aquifer Solution System (IWASS) by integrating the single well programs, which perform detailed analysis of flow towards individual wells, into a number of additional programs which employ the finite element method to analyse regional groundwater flow.

2. THEORETICAL BACKGROUND

2.1 DESCRIPTION OF FLOW PROBLEMS

The problems solved by the programs are those involving one-dimensional and two-dimensional, axi-symmetric flow towards a pumped well constructed in unconsolidated materials. The Forchheimer non-linear velocity-hydraulic gradient relationship is used to describe non-Darcy flow which may exist near the well screen whilst Darcy's law is used to describe flow outside the non-Darcy zone. Derivation of generalised field equations for three-dimensional flow are presented in Reference 3, Chapter 2. The formation materials are assumed to be isotropic (Modification of the programs to deal with anisotropy is outlined in Section 5). Both single and multi-layered formations are considered. For a confined aquifer, the hydraulic parameters necessary to define a transient, two-regime flow problem are the Forchheimer coefficients (a, and b), the critical velocity (V_{cr}), coefficient of hydraulic conductivity (K) and the coefficient of specific storage (S_s). For a water table aquifer, additional parameters required are the coefficient of specific yield (S_y) and Boulton's reciprocal of the delayed yield index (α). For an aquitard, where only Darcy flow occurs, the hydraulic parameters are S_s' , S_y' , K' and α' . However, if one is concerned only with steady flow, the coefficients S_s , S_y , α , S_s' , S_y' and α' may be omitted.

Apart from the material data, the following data are also necessary for complete definition of the problem.

(i) Well characteristics which include radius of well screen(s), length and position of each screen, geometry of gravel pack and its material properties if the well is gravel packed.

(ii) Formation geometry which includes thickness of each layer (m) and an external radius (r_o) for all layer(s).

(iii) Boundary conditions which include the pumping condition of the well (constant head (h_w) or constant discharge (Q)), the flow condition at the external radius (barrier boundary or recharge boundary condition), and the flow condition at the top boundary of the system (leaky or non-leaky boundary condition).

(iv) Initial condition which is the initial height of the water table above the datum (h_o).

2.2 FINITE ELEMENT METHOD OF SOLUTION

2.2.1 GENERAL DESCRIPTION OF THE METHOD

The programs employ the variational approach and finite element method presented in Reference 3 to obtain numerical solutions of the flow problems. The procedures adopted consist of first replacing the initial-boundary value problem, described by the field equations, initial and boundary conditions, by an equivalent variational problem which is that of finding a hydraulic head function that minimises a certain functional. Secondly, an approximate solution is then obtained as follows:

(i) The continuous region of the flow system is discretised into a finite number of closed sub-regions termed "finite elements". The finite elements are assumed to be interconnected at a discrete number of nodal points situated on their boundaries.

(ii) A piecewise function is chosen for each element. The function defines uniquely the hydraulic head distribution within the element in terms of its nodal parameters.

(iii) The functional over the entire flow region is assumed to be contributed by each element and the process of minimisation is accomplished by evaluating the elemental contributions, adding all such contributions, differentiating the resulting functional with respect to the nodal parameters and equating the differentials to zero. This gives rise to a system of simultaneous algebraic equations which can be readily solved by direct elimination or iterative methods.

For steady flow problems, numerical solutions are obtained in terms of the nodal values of hydraulic head, element velocities (if required) and the total discharge into the well. For transient flow problems, solutions are obtained for each time step in terms of the nodal values of hydraulic head, dimensionless nodal drawdowns and, if required, element velocities.

2.2.2 AUTOMATIC MESH GENERATION SCHEMES

A number of schemes for automatic discretisation of one-dimensional and two-dimensional flow regions are incorporated in the programs to avoid the tedious preparation and checking of the mesh data (nodal co-ordinates and node connections of elements) by the user. Input data required is thus reduced to the general problem data, described previously in Section 2.1, and a few discretisation parameters. The schemes are now described as follows:

(i) Discretisation of one-dimensional region

The region is divided into a number of line segments, each of which is further subdivided into a number of 3-node quadratic elements. The length of the first line segment and the number of elements for each segment are to be specified by the user. The lengths of the remaining line segments are generated from

$\Delta r_i = f \times \Delta r_{i-1}$, where f is a scale factor.

Nodal co-ordinates and element nodal connections can be readily established once all the lengths have been computed.

(ii) Discretisation of two-dimensional regions

Figure 1 shows the discretisation pattern generated for a single aquifer with a fully penetrating well. The region is divided into a number of vertical blocks, each of which is further subdivided into a number of triangular elements. The horizontal width of the first block Δr_1 , the number of vertical sub-divisions of this block and the number of repeated regular blocks before grading of the subdivision takes place are to be specified by the program user. The widths of the remaining blocks are generated from $\Delta r_i = \Delta r_{i-1} \times f$, and the number of vertical subdivisions in the next repeated blocks is established by reducing the number in the previous blocks by half or by one. Node numbering traverses vertically across the aquifer from bottom to top. For each vertical block, nodal co-ordinates and node connections of elements in the block are also established.

Figure 2 shows the discretisation pattern for a single aquifer with a partially penetrating well. The zone immediately surrounding the well is specially discretised whilst the rest of the flow region is descretised in the manner described above.

Figure 3 shows the discretisation pattern for a multi-layer system with a fully penetrating well. The entire region is divided into a number of vertical blocks, each of which is further split up into sub-blocks which belong to separate layers. The sub-block in each layer is then subdivided into rectangular or triangular elements. The latter are used when grading is required.

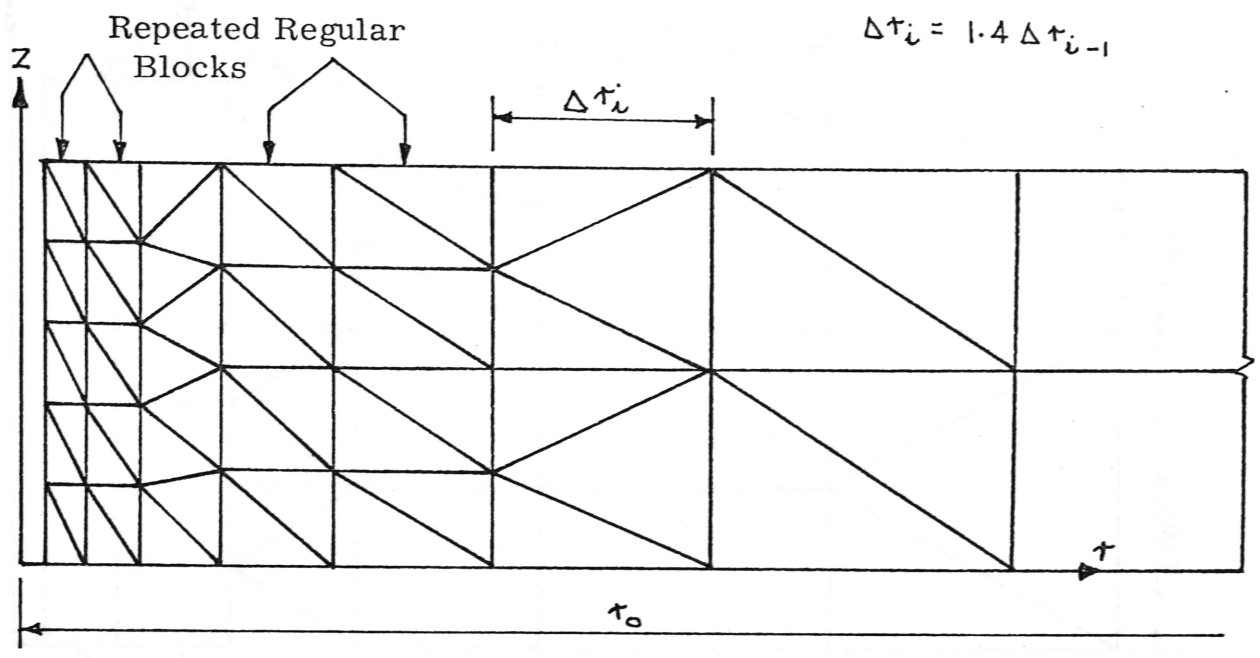


Fig. 1: Discretisation pattern for a single aquifer with a fully penetrating well.

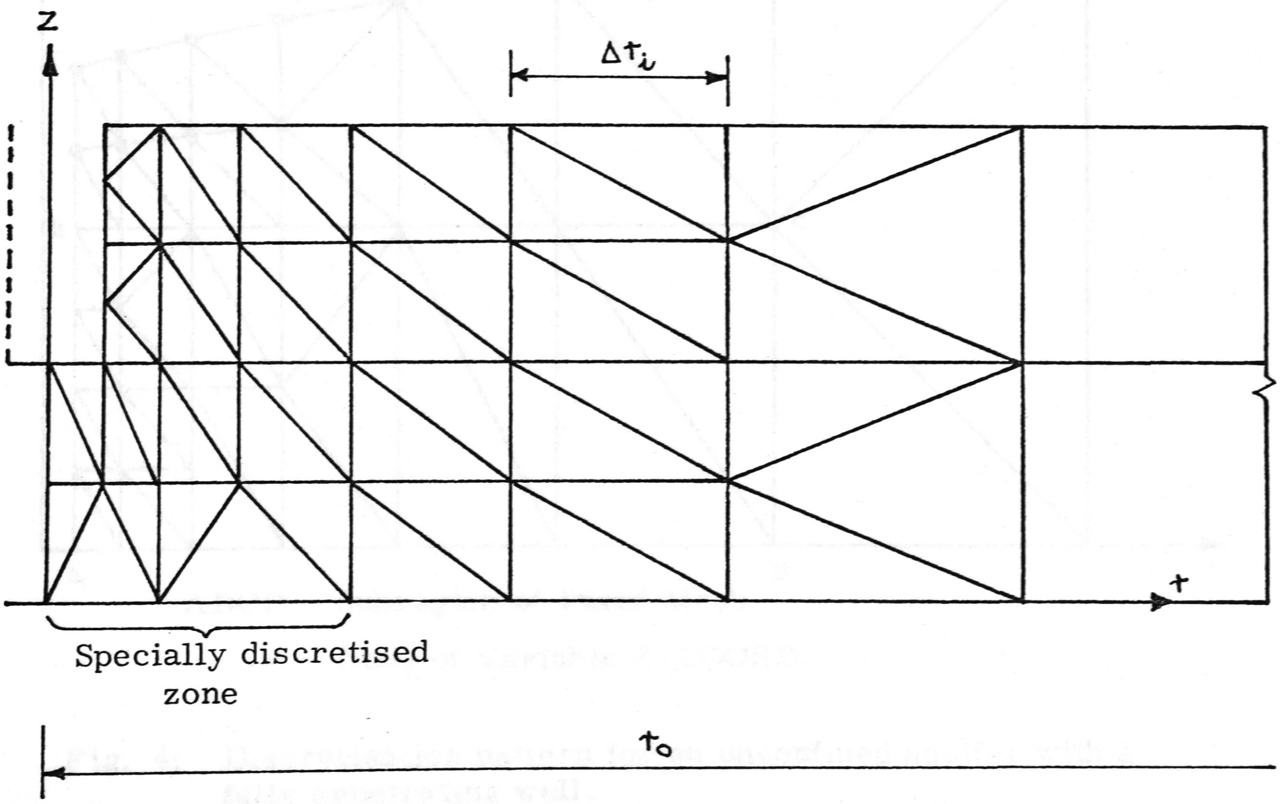


Fig. 2: Discretisation pattern for a single aquifer with a partially penetrating well.

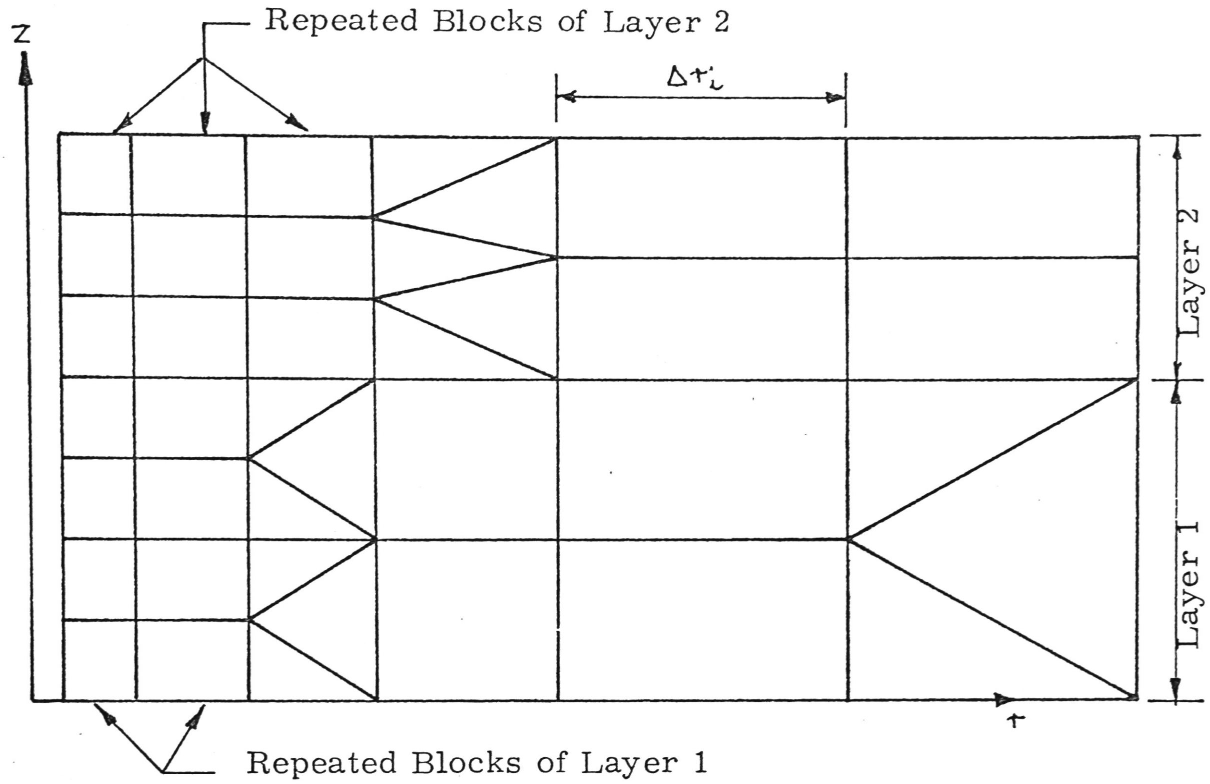


Fig. 3: Discretisation pattern for a multi-layer system with a fully penetrating well.

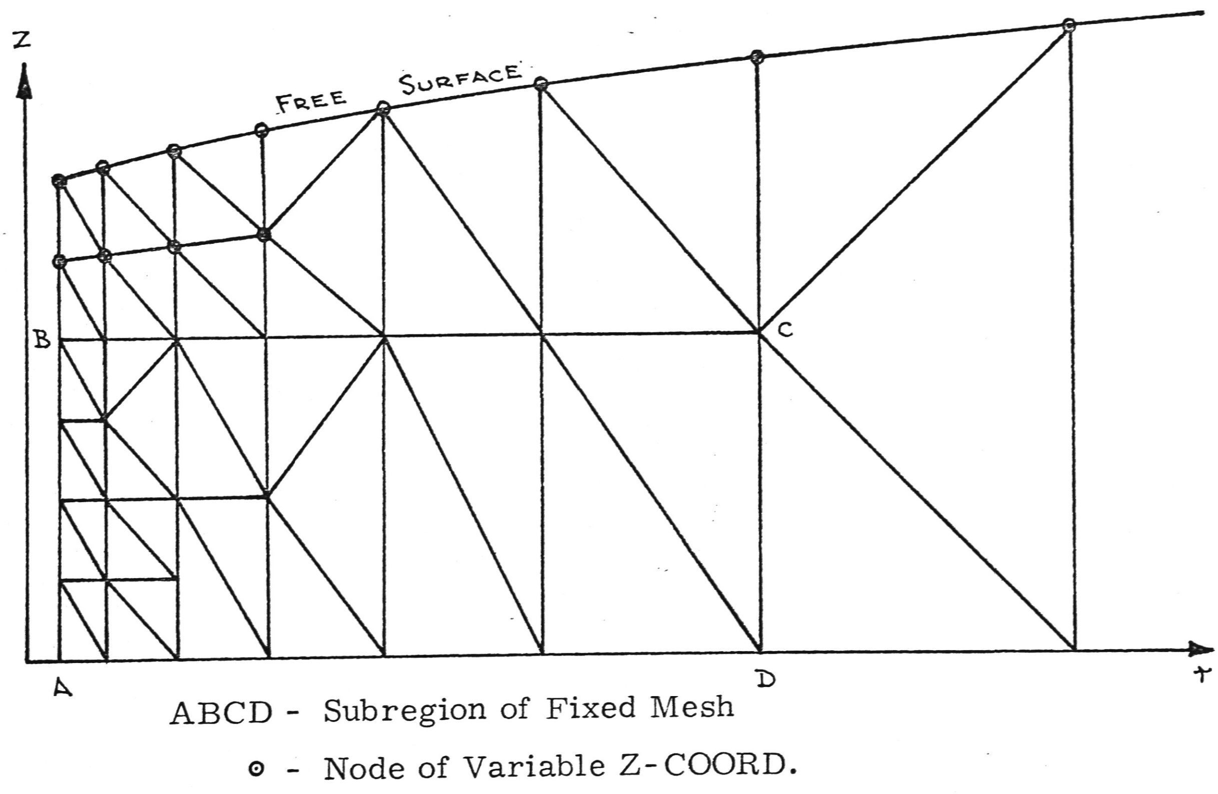


Fig. 4: Discretisation pattern for an unconfined aquifer with a fully penetrating well.

Figure 4 shows the discretisation pattern for an unconfined aquifer with a fully penetrating well. To accommodate the movement of the free surface, the saturated flow region is divided into 2 zones, one where the mesh is fixed and another where the mesh is allowed to contract or expand. Each zone is then discretised into triangular elements in the manner described previously. The two zones are merged into one at a radial distance where the numbers of their vertical subdivisions are equal to 2.

3. PROGRAM DETAILS AND SAMPLE RUNS

3.1 DEFINITION OF INPUT AND OUTPUT VARIABLES

A list of the programs and their functions is given in Table 0. All programs are written in FORTRAN IV and are currently operational on an IBM 360/50 machine with a G and H level compiler and on a CDC 6600 machine. The core storage required by each of the one-dimensional flow programs (STCON1 and TRCON1) is approximately 120K. The core storage required by each of the remaining six two-dimensional flow programs is approximately 150K.

For each program, more than one problem may be solved and any consistent set of units may be used. (The units adopted in the sample runs are Ft.-Min.). Optional printout of the numerical output is also included. Also, if required, the user may insert his plotting subroutines to obtain graphical outputs. Plotting subroutines written by the author are for the CALCOM plotter which is currently used at the University of New South Wales.

3.1.1 INPUT VARIABLES

A list of the input variables is given in Table 1. The variables are classified into nine groups and those appearing in each program are indicated by an x-sign. In the program listings presented in Section 4, a brief definition of all variables used is given in order of their presence in the FORMAT statements. Wherever possible, recommended values are also included. Full description of the variables are given below.

(i) Group 1: Variables common to all programs

NPROB denotes the number of problems to be solved by the programs.

IVEL is an index used to indicate whether the printout of computed velocities is required. IVEL takes the value of 0 or 1. A value of IVEL = 0 will result in deletion of the printout.

IDISCR is an index used to indicate whether the printout of generated mesh data is required. IDISCR takes the value of 0 or 1. A value of IDISCR = 0 will result in deletion of the printout.

TABLE 0: LIST OF PROGRAMS

Program Name	Function
STCON1	Solves steady, one-dimensional flow by employing 3-node line elements
TRCON1	Solves transient, one-dimensional flow by employing 3-node line elements.
STCON3	Solves steady, two-dimensional flow by employing triangular elements
TRCON3	Solves transient, two-dimensional flow by employing triangular elements.
STCOND	Solves steady, two-dimensional flow by employing rectangular and trianglar elements.
TRCOND	Solves transient, two-dimensional flow by employing rectangular and triangular elements.
STFREE	Solves steady, two-dimensional, free surface flow by employing triangular elements.
TRFREE	Solves transient, two-dimensional, free surface flow by employing triangular elements.

Table 1 : List of Input Variables for all Programs

Input Variables		Program							
		STCON1	TRCON1	STCON3	TRCON3	STCOND	TRCOND	STFREE	TRFREE
Group 1	NPROB	x	x	x	x	x	x	x	x
	IVEL	x	x	x	x	x	x	x	x
	IDISCR	x	x	x	x	x	x	x	x
	ORELAX	x	x	x	x	x	x	x	x
	RW	x	x	x	x	x	x	x	x
	RO	x	x	x	x	x	x	x	x
	HO	x	x	x	x	x	x	x	x
	HTOL	x	x	x	x	x	x	x	x
Group 2	HW	x		x		x		x	
	QFIX	x	x		x		x		x
	RCSNG		x		x		x		x
	QRTOL		x		x		x		x
Group 3	IAQTA	x	x	x		x			
	IGP	x	x	x	x	x	x	x	x
	IBOUND		x		x		x		x
	IWBC	x	x		x		x		x
	IKMAX				x	x	x		
	IWAT		x		x		x		
	NSTEP							x	x
Group 4	AK	x	x	x				x	x
	BK	x	x	x				x	x
	VCR	x	x	x				x	x
	TH	x	x	x				x	x
	SS		x						x
	SY		x		x		x		x
	DINDEX		x		x		x		x
	THA	x	x	x		x			
	PA	x	x	x		x			

Input Variables		Program							
		STCON1	TRCON1	STCON3	TRCON3	STCOND	TRCOND	STFREE	TRFREE
Group 5	NLAYR				x	x	x		
	AKL(I)				x	x	x		
	BKL(I)				x	x	x		
	VCRL(I)				x	x	x		
	SSL(I)				x		x		
	THL(I)				x		x		
Group 6	AGP	x	x	x	x	x	x	x	x
	BGP	x	x	x	x	x	x	x	x
	VGP	x	x	x	x	x	x	x	x
	SSP		x						
	THGP	x	x	x	x	x	x	x	x
	BTGP			x	x	x	x	x	x
Group 7	NELF	x	x						
	FRLEN	x	x	x	x	x	x	x	x
	SCFAC	x	x	x	x	x	x	x	x
	XLMAX			x	x	x	x	x	x
	IREG			x				x	x
	NMIN			x				x	x
	IREGL(I)				x	x	x		
	NMINL(I)				x	x	x		
	NFRL(I)				x	x	x		
Group 8	NDSC			x	x	x	x	x	x
	NSCREN			x	x	x	x	x	x
	XSCR(I)			x	x	x	x	x	x
	HSCR(I)			x	x	x	x	x	x
	IPENTR			x					
Group 9	NTICR		x		x		x		x
	ITST				x		x		x
	TFACTR		x		x		x		x
	TMUL		x		x		x		x
	DTMUL		x		x		x		x

<u>ORELAX</u> is an over-relaxation factor to be used in the iterative solution of a system of non-linear algebraic equations. For fast convergence, a value of ORELAX between 1.50 and 1.85 is recommended.

<u>RW</u> is the radius of the well screen, which may or may not be the same as the radius of well casing.

<u>RO</u> denotes the external radius of the flow system. At this radius, it is assumed in all steady flow programs (STCON1, STCON3, STCOND and STFREE) that the system is intercepted by a recharge source, and that the hydraulic head of this source is unaffected by pumping.

<u>HO</u> denotes the hydraulic head at the external radius RO. In transient flow programs, HO also denotes the initial head in the flow system.

<u>HTOL</u> is a tolerance for successive iterations on head values. A head tolerance of 0.10 ft. or a few percent of well drawdown usually gives satisfactory results within 4 or 5 iterations. As the solution scheme employed in the programs is an efficient combination of the over-relaxation and Gauss elimination schemes, a more refined value of HTOL usually does not involve a significant increase in the number of iterations.

(ii) Group 2: Variables relating to boundary conditions at the well

In all the transient flow programs, the prescribed discharge boundary condition at the well is simulated. The effect of well storage may be taken into account or neglected. In all the steady flow programs,except STCON1, the prescribed head condition is simulated. In the program STCON1, either the prescribed discharge or prescribed head condition may be simulated.

HW denotes the prescribed hydraulic head at the well. HW does not include screen losses and other losses due to flow into or inside the well.

QFIX denotes the prescribed discharge into the well.

RCSNG denotes the radius of well casing.

QRTOL is the prescribed discharge tolerance to be used in the iteration for correct well discharge. Due to the fact that the hydraulic head or well drawdown corresponds to a given value of discharge is unknown a priori, the condition of prescribed well discharge has to be satisfied by trial and error.

(iii) Group 3: Index variables

IAQTA is an index used to indicate whether the pumped aquifer is overlain by an aquitard. IAQTA takes the value of either 0 or 1. A value of IAQTA = 1 is used if the aquifer is overlain by an aquitard whose hydraulic head at the top boundary is unaffected by pumping. (i.e. Hantush-Jacob aquifer-aquitard system). A value of IAQTA = 0 is used for a single aquifer confined by impermeable strata. When IAQTA = 1, solution for the aquifer-aquitard system is obtained on the assumption that flow in the aquitard layer is vertically downward towards the top of the aquifer.

IGP is an index used to indicate whether the well is gravel packed. IGP = 1 for gravel packed wells. IGP = 0 for non-gravel packed wells.

IBOUND is an index used to indicate whether an impermeable barrier boundary or a recharge boundary is present at the external radius. IBOUND is used only in transient flow programs. Its value is 1 for a recharge boundary and 0 for a barrier boundary.

IWBC — is an index used in transient programs and steady flow programs STCON1 to indicate the type of boundary condition prevailing at the well. When used in the transient flow programs, IWBC = 1 indicates the prescribed discharge condition with well storage whilst IWBC = 0 also indicates the prescribed discharge condition but without well storage. When used in STCON1, IWBC = 1 indicates the prescribed discharge condition whilst IWBC = 0 indicates prescribed head.

IKMAX — is an index which appears in programs TRCON3, STCOND and TRCOND, which are capable of solving more complex problems involving flow in multi-layer systems. IKMAX is used to indicate the layer of maximum permeability. Its value corresponds to the layer number of the most permeable layer. (Each layer is numbered starting from bottom to top of the flow system).

IWAT — is an index used in programs TRCON3 and TRCOND to indicate whether the top layer is confined or unconfined. IWAT = 1 indicates an unconfined flow system whilst IWAT = 0 indicates a confined flow system. When IWAT = 1, an assumption of constant saturated thickness for the top layer is made in obtaining the numerical solution.

NSTEP — is an index used only in the free surface flow programs (STFREE and TRFREE) to indicate whether a seepage face exists at the well boundary. (The seepage face will exist if the water level in the well lies below the top of the well screen.) NSTEP = 2 indicates that the seepage face is present and a two-step iterative procedure is to be employed to obtain the correct free surface. NSTEP = 1 indicates that the seepage face does not exist and a one-step iterative procedure is to be employed. Detail on the one-step and two-step iteration procedures for correct free surface is given in Reference 3.

(iv) Group 4: Input Data on the Hydraulic Properties of an Aquifer and Aquitard

AK corresponds to the Forchheimer linear coefficient, a

BK corresponds to the Forchheimer non-linear coefficient, b If wholly Darcy flow is to be simulated the value of BK should be set to zero and AK should be equal to 1/PM, where PM denotes the hydraulic conductivity of the aquifer.

VCR corresponds to V_{cr}, which denotes a critical velocity where non-Darcy flow commences.

TH corresponds to m, which denotes the initial saturated thickness of a confined or an unconfined aquifer.

SS corresponds to S_s, which denotes the coefficient of specific storage of a confined or an unconfined aquifer.

SY corresponds to S_y, which denotes the coefficient of specific yield of an unconfined aquifer or the top layer of a multi-layer flow system.

DINDEX corresponds to α, which denotes the reciprocal of Boulton's delayed yield index of an unconfined aquifer or the top layer of a multi-layer flow system.

THA corresponds to m', which denotes the thickness of an overlying aquitard of the Hantush-Jacob flow system.

PA corresponds to K', which denotes the coefficient of hydraulic conductivity of the overlying aquitard.

(v) Group 5: Input Data on Hydraulic Properties of a Multi-layer flow System

NLAYR — is the number of layers. The value of NLAYR is not to exceed 3 in the current version of programs TRCON3, STCOND and TRCOND.

AKL(I) — is the linear Forchheimer coefficient of layer number I.

BKL(I) — is the non-linear Forchheimer coefficient of layer number I. If wholly Darcy flow is to be simulated, BKL(I) should be set to zero, and AKL(I) = 1/PML(I), where PML(I) denotes the hydraulic conductivity of layer I.

VCRL(I) — is the critical flow velocity of layer number I.

SSL(I) — is the specific storage of layer number I.

THL(I) — is the thickness of layer number I.

(vi) Group 6: Input Data on Hydraulic Properties and Geometry of Gravel Pack

AGP — is the linear Forchheimer coefficient of gravel pack material.

BGP — is the non-linear Forchheimer coefficient of gravel pack material.

VGP — is the critical flow velocity of gravel pack material.

SSP — is the specific storage of gravel pack material.

THGP — is the thickness of gravel pack annulus.

BTGP is the height of the base of gravel pack above the bottom of the flow system.

(vii) Group 7: Input Data required for Mesh Generation

Figure 5 shows diagrammatic sketches of the discretised, one-dimensional and two-dimensional flow regions for a single layer system. The subdivision of the regions into finite elements is also illustrated. Figure 6 shows a diagrammatic sketch of the discretised region for a multi-layer system. All of the input variables in group 7 are depicted in both figures.

These variables are defined as follows:

NELF is the number of 3-node,one-dimensional elements in the first subregion.

FRLEN is the length of the first subregion or first vertical block.

SCFAC is the scale factor used in generating the lengths of the remaining subregions.

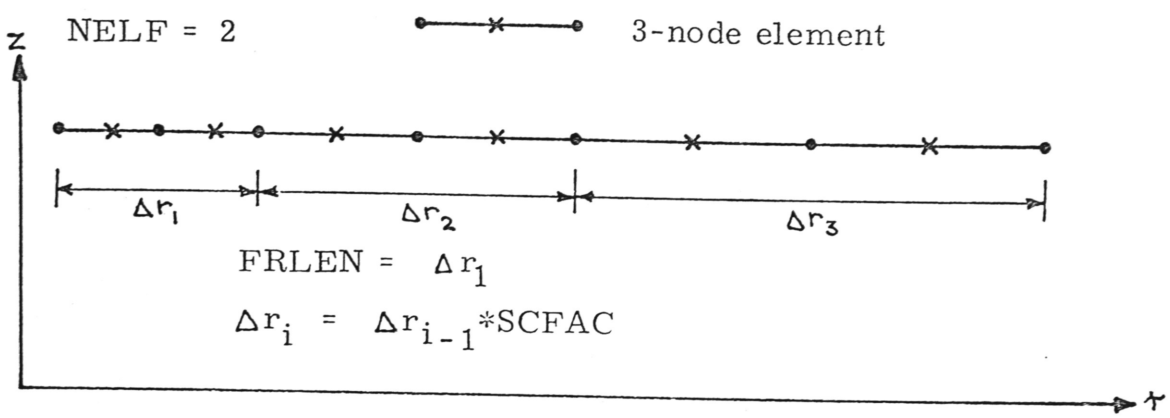
XLMAX is the maximum horizontal width of the block. XLMAX is specified to avoid ill-conditioned triangular or rectangular elements.

IREG-1 is the number of repeated regular blocks with the same number of nodes on its left and right vertical lines.

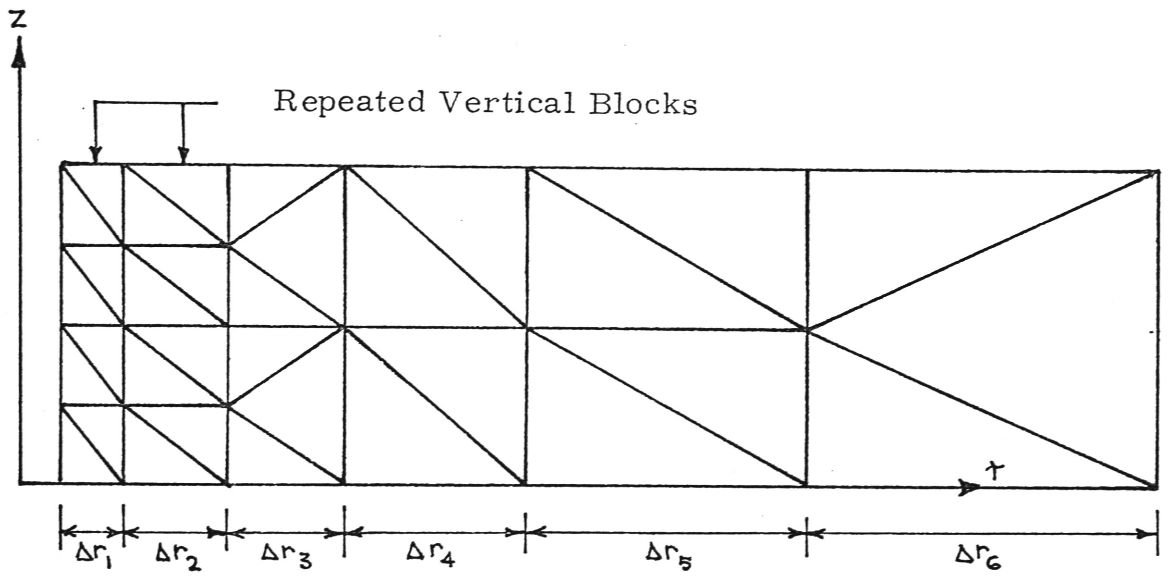
NMIN is the minimum number of nodes on a vertical line. (NMIN $\geq$ 2)

IREGL(I)-1 is the number of repeated regular blocks with the same number of nodes on the left and right vertical lines across layer I.

NMINL(I) is the minimum number of nodes along a vertical line across layer I.



(i) Subdivision of one-dimensional region into subregions.



(ii) Subdivision of region into subregions

FRLEN = Δr_1

$\Delta r_i = \Delta r_{i-1}$ * SCFAC

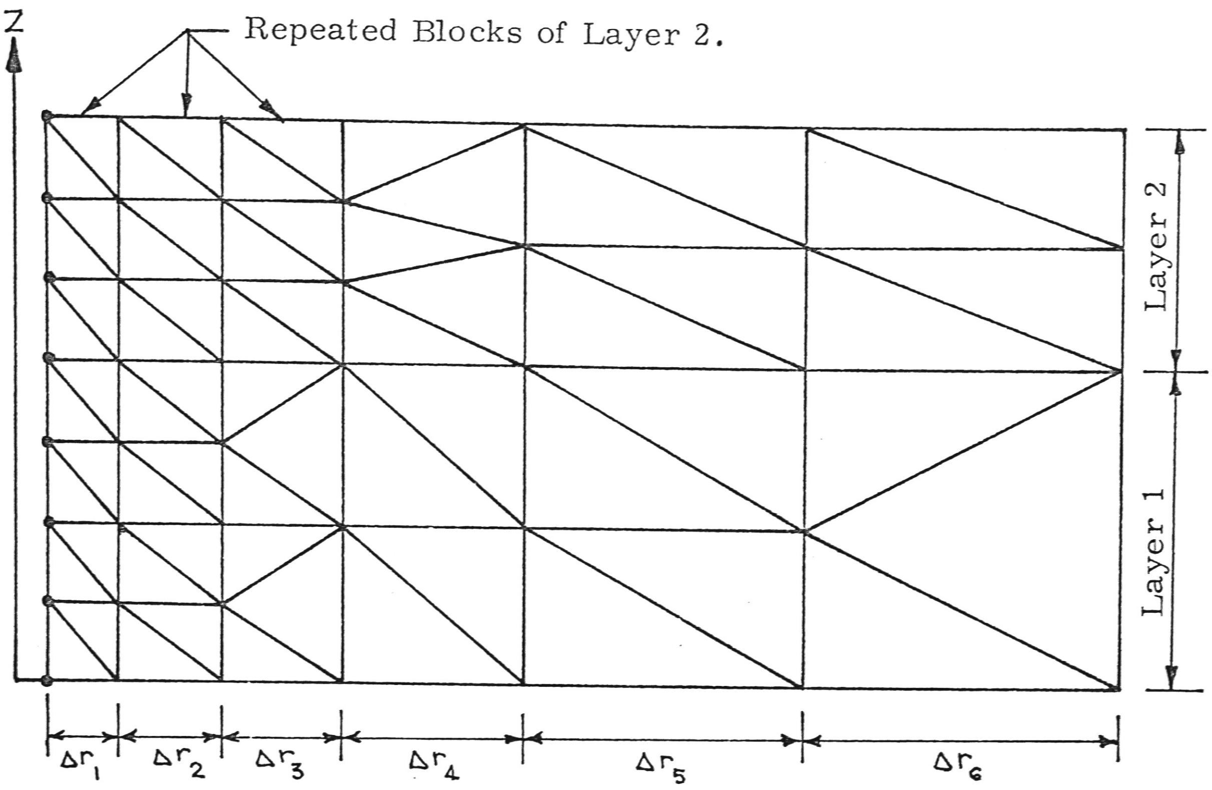
$\Delta r_6 \leqslant$ XLMAX

(iii) Subdivision of subregion into triangular elements

IREG-1 = 2

NMIN = 2

Fig. 5: Discretised One- and Two-dimensional flow regions for a single layer system.



(i) Subdivision of region into vertical blocks

$$\text{FRLEN} = \Delta r_1$$

$$\Delta r_i = \Delta r_{i-1} * \text{SCFAC}$$

$$\Delta r_6 \leqslant \text{XLMAX}$$

(ii) Subdivision of each block into elements

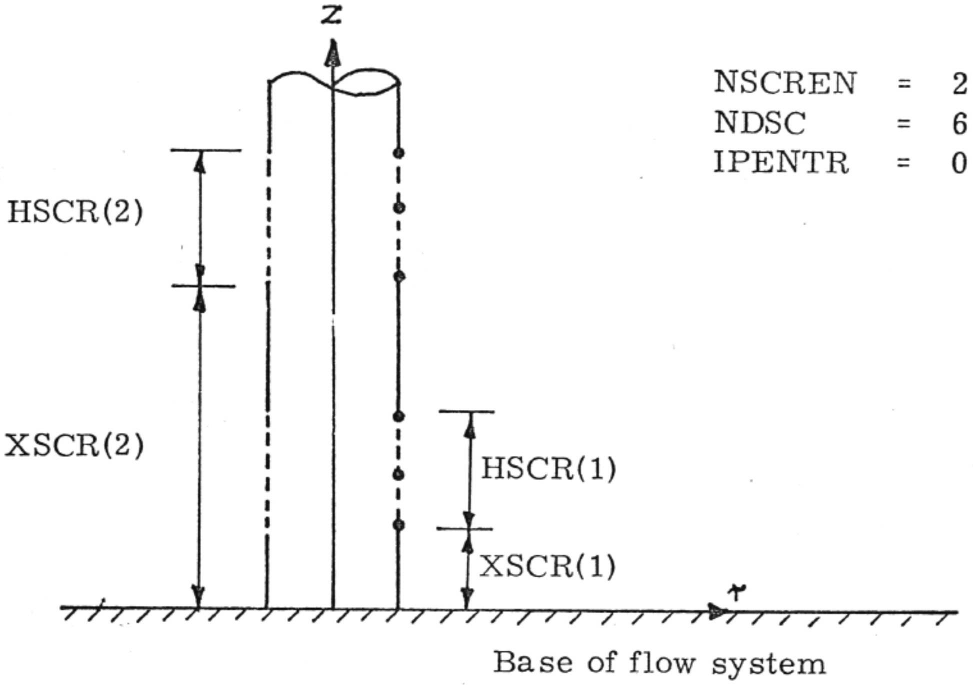
<u>Layer 1</u>

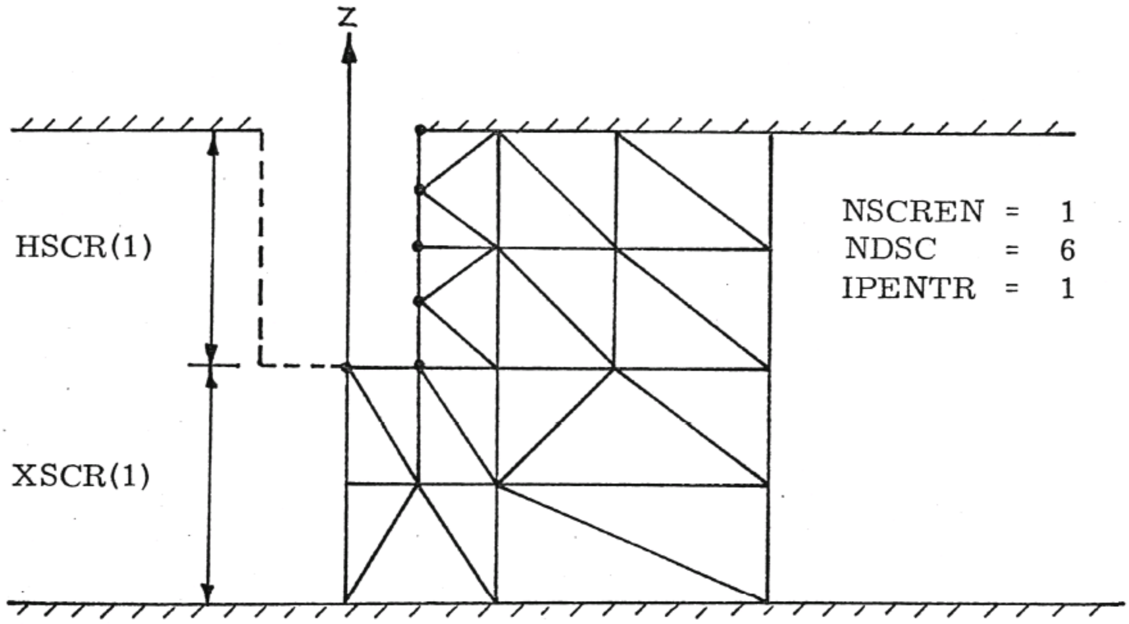
IREGL (1)-1 = 2
NMINL(1) = 2
NFRL (1) = 5

<u>Layer 2</u>

IREGL(2)-1 = 3
NMINL(2) = 3
NFRL (2) = 4

Fig. 6: Discretised two-dimensional flow regions for a multi-layer system.





(For partially penetrating case NDSC must be an even number.)

Fig. 7: Boundaries of fully and partially penetrating wells showing screen details.

NFRL(I) is the number of nodes on the portion of well boundary which penetrates layer I.

(viii) Group 8: Input Data on Well Screens

Figure 7 shows a diagrammatic sketch of a fully penetrating but partially screened well and a sketch of a partially penetrating well. All input variables of this group are also depicted in the figure.

The variables are fully described as follows:

NDSC is the total number of nodes on well screen(s).

NSCREN is the number of screened intervals.

XSCR(I) is the height of screen number I above the bottom of the flow system.

HSCR(I) is the length of screen number I.

IPENTR is the penetration index. For a fully penetrating well, IPENTR = 0. For a partially penetrating well IPENTR = 1.

(ix) Group 9: Input Data on Time Discretisation

NTICR is the number of time steps.

ITST is the starting time step number where the solution starts. If the solution starts from the initial time (when the hydraulic head in the system is equal to HO) ITST is equal to 1. If the solution does not start from this initial time, ITST should be set equal to an appropriate time step number IT, which is greater than or equal to 2. In such a case, additional input data on nodal values of the hydraulic head and flux at the time corresponding to that step number is also required. (For detailed information, see Section 5.3).

TFACTR is the dimensionless time of the first time step. TFACTR is computed from the real time value as follows: Let Δt be the real time value of the starting time step.

$$\text{TFACTR} = 4\Delta t K/(r^2 S_s)$$

where K and S_s are the hydraulic conductivity and specific storage of the single aquifer or the most permeable layer of a multi-layer system, and r is equal to $r_w + 2$.

TMUL is the time multiplier to be used in generating the remaining time steps. A value of TMUL between 1.40 and 2.00 is recommended.

DTMUL is the increment of the time multiplier. A value of DTMUL between 0 and 0.02 is recommended.

3.1.2 OUTPUT VARIABLES

Table 2 lists the variables which are printed out for each flow problem solved. These variables are classified into 5 different groups. The printout of the variables which belong to groups 1 and 4 may be omitted by specifying IDISCR and IVEL as zero respectively.

(i) Group 1: Generated discretisation data

NNODE is the total number of nodes in the finite element network. For the one-dimensional programs STCON1 and TRCON1, NNODE must not exceed 200. For the remaining two-dimensional programs, NNODE must not exceed 300. To cater for larger values of NNODE, the COMMON blocks have to be extended as described in Section 5.

NELEM is the total number of elements in the network. For the one-dimensional programs, NELEM must not exceed 150. For the remaining two-dimensional programs, NELEM must not exceed 500 in programs STCON3, TRCON3, STFREE and TRFREE and 200 in programs STCOND and TRCOND.

Table 2 : Listing of Output Variables

Output Variables		Program							
		STCON1	TRCON1	STCON3	TRCON3	STCOND	TRCOND	STFREE	TRFREE
Group 1	NNODE	x	x	x	x	x	x	x	x
	NELEM	x	x	x	x	x	x	x	x
	X(I,J)	x	x	x	x	x	x	x	x
	NOD(K,L)			x	x	x	x	x	x
	NREP(K)			x	x	x	x	x	x
	IPROP(K)					x	x		x
	LEN			x	x	x	x	x	x
	JBD(M)			x		x		x	
	DISP(M)			x		x		x	
Group 2	IT		x		x		x		x
	TM		x		x		x		x
	TMM		x		x		x		x
Group 3	H(I)	x	x	x	x	x	x	x	x
	TLESS(I)	x	x	x	x	x	x	x	x
	HLESS(I)	x	x	x	x	x	x	x	x
	III	x	x	x	x	x	x	x	x
	EMAX	x	x	x	x	x	x	x	x
	PARAM	x		x		x		x	
	RVEC(N)			x	x	x	x	x	x
	HF(N)			x	x	x	x	x	x
Group 4	VEL(K)			x	x	x	x	x	x
	VCOMP1			x	x	x	x	x	x
	VCOMP2			x	x	x	x	x	x
	VMEAN(I,J)	x	x						
	IDARCY(K)			x	x	x	x	x	x
Group 5	IQ		x		x		x		x
	SW		x		x		x		x
	QAQFR		x		x		x		x
	QSTRGE		x		x		x		x
	QCALL		x		x		x		x
	QRDIF		x		x		x		x
	QWB(J)			x					
	QSUM	x		x				x	

<u>X(I,J)</u> is a matrix containing the nodal co-ordinates. Its first subscript I ranges from 1 to NNODE. Its second subscript J ranges from 1 to 2.

<u>NOD(K,L)</u> is a matrix containing information about the nodal connections of all elements in the network. Its first subscript K ranges from 1 to NELEM. Its second subscript L ranges from 1 to the number of nodes belonging to each element.

<u>NREP(K)</u> is a vector containing information about the repetition of element geometry (shape and size). Its subscript ranges from 1 to NELEM. The information contained in NREP is useful in generating element matrices as significant computational effort can be saved by generating only the matrices for elements of different geometry.

<u>IPROP(K)</u> is a vector containing information about the type of material for each element. (e.g. if an element K lies in layer 1, IPROP(K) is equal to 1. If the element lies in the gravel pack zone IPROP(K) = 0). The subscript K of IPROP ranges from 1 to NELEM.

<u>LEN</u> is the length of the gross vector which contains the coefficients in each row of the gross matrix, formed by assembling all elements in the network. The maximum value of LEN must not exceed 2000. To cater for larger values require an extension of certain COMMON blocks in the programs.

<u>JBD(M)</u> is a vector containing the node numbers of certain nodes in the network at which the head values are known or prescribed. Subscript M ranges from 1 to the total number of such nodes.

<u>DISP(M)</u> is a vector containing the prescribed values.

(ii) Group 2: Generated time step data

IT is the time step number.

TMM is the real time value at the mid-point of time step IT.

TM is the real time value at the end of time step IT.

(iii) Group 3: Nodal head values and related Solution Parameters

H(I) is a vector containing the values of hydraulic head at various nodes in the network. Its subscript I ranges from 1 to NNODE.

TLESS(I) is a vector containing the values of dimensionless radius or dimensionless time for various nodes in the network. For steady flow programs, TLESS(I) corresponds to $1/u = r/r_o$. For transient flow programs, TLESS(I) corresponds to $1/u = 4Kt/r^2 S_s$

HLESS(I) is a vector containing the values of dimensionless drawdown for various nodes in the network. For steady flow programs, HLESS(I) corresponds to $W(u) = 2\pi Kms/Q$ For transient flow programs, HLESS(I) corresponds to $W(u) = 4\pi Kms/Q$

III is the number of iterations required for final solution of the system of non-linear algebraic equations.

EMAX is the maximum error in head values.

PARAM is a non-linear parameter for the type curves obtained by plotting HLESS(I) versus TLESS(I). PARAM corresponds to λ , which is calculated from $\lambda = bQK/2\pi m r_o$

RVEC(N) is a vector containing the radial co-ordinates of the nodes situated on the top of the flow system. Subscript N ranges from 1 to the total number of such nodes.

HF(N) is the Z-co-ordinates of the top boundary nodes.

(iv) Group 4: Calculated element and nodal velocities

VEL(K) is a vector containing values of absolute velocities at the centroids of the elements. Subscript K of VEL ranges from 1 to NELEM.

VCOMP1 is a radial component of an element velocity.

VCOMP2 is a vertical component of an element velocity.

VMEAN(I,J) is a matrix containing values of nodal velocities. Its subscript I ranges from 1 to NNODE and subscript J ranges from 1 to 2. J = 1 refers to the radial component while J = 2 refers to the vertical component.

IDARCY(K) is an index vector used to indicate the type of flow (Darcy or non-Darcy) occuring within a particular element. IDARCY(K) = 1 indicates that non-Darcy flow occurs within element K whilst IDARCY(K) = 0 indicates that Darcy flow occurs.

Group 5: Discharge variables and related parameters

IQ is the discharge iteration number.

SW is the drawdown at the well.

QAQFR is the discharge from aquifer(s) into the well.

QSTRGE is the discharge from well storage. QSTRGE may be significant for large diameter wells, particularly at early times.

QCALL is the calculated well discharge, which is the sum of QAQFR and QSTRGE.

QRDIF is the absolute value of the difference between QCALL and the prescribed well discharge, QFIX.

QWB(J) is a vector containing values of the nodal discharges from aquifer into well. Subscript J ranges from 1 to the total number of nodes on well screen(s).

QSUM is the steady discharge from aquifer(s) into well. It is computed from

$$QSUM = \sum_{J} QWB(J)$$

3.2 STCON1 - PROGRAM FOR SOLVING STEADY, ONE-DIMENSIONAL FLOW

The program STCON1 is used to analyse steady, one-dimensional Darcy or two-regime flow. It can handle the situations involving radial flow in a single confined aquifer or a two-layer system consisting of a confined aquifer overlain by a water table aquitard. For this latter system, the assumptions of vertical flow in the aquitard and constant saturated thickness are made.

To save labour and prevent errors in the preparation of input data, a form for direct keypunching of data cards is presented in Figure 8. The input sequence and data formats are clearly specified. The first card is the specification card which specifies the number of problems to be solved and whether optional printout is required. The remaining cards are data cards for each problem.

Completed form for a sample run is shown in Figure 9. Two problems were solved by the program. The first problem involves one-dimensional two-regime flow in a confined aquifer. The second involves flow in an aquifer-aquitard system. For both problems, the well is fully screened through the aquifer and non-gravel packed and HO is taken as the datum in measuring the hydraulic head (i.e. HO = 0.0). Numerical output is shown on pages 35 to 38. The output consists of a reformatted playback of the general input data and three tables of results which are described in the following manner.

The first table lists the nodal co-ordinates, head values and nodal values of Rho- and Tzi-co-ordinates which are defined as $\rho = r/2m$ and $\xi = s/s_w$, where s denotes the nodal drawdowns ($s = h_0 - h$) and s_w denotes the drawdown at the well.

The second table lists the node numbers and nodal velocities which are obtained by averaging the element velocities. The discharge into the well is also given in the table.

The last table lists the node numbers, radial co-ordinates and nodal values of the well function, $W(u) = 2\pi Kms/Q$, and its argument, $1/u = r/r_0$. From this table, a type curve for steady, one-dimensional flow may be obtained by plotting $W(u)$ versus $1/u$. The non-linear parameter of the tupe curve, λ , is also given in the table. λ was calculated from $\lambda = bQK/2\pi m r_0$.

FIGURE 8: INPUT DATA FORM FOR STCON1

COMPUTER DATA SHEET	PROGRAM STCON1 Steady, one-dimensional flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL

(5F10.2)

AK	BK	VCR

(3F10.3)

IGP	IAQTA	IWBC

(3I10)

AGP	BGP	VGP	THGP

(Omit this card if IGP = 0)

(4F10.3)

PA	THA

(Omit this card if IAQTA = 0)

(F10.4, F10.2)

HW

(Omit this card if IWBC = 0)

(F10.2)

QFIX

(Omit this card if IWBC = 1)

(F10.2)

NELF	SCFAC	FRLEN

(I10, 2F10.2)

FIGURE 9: COMPLETED FORM FOR 2 SAMPLE PROBLEMS

COMPUTER DATA SHEET	PROGRAM STCON1 Steady, one-dimensional flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
2	1	0	1.50

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL
1.00	1000.00	0.00	50.00	0.20

(5F10.2)

AK	BK	VCR
10.000	20.000	0.060

(3F10.3)

IGP	IAQTA	IWBC
0	0	0

(3I10)

AGP	BGP	VGP	THGP

(Omit this card if IGP = 0)

(4F10.3)

PA	THA

(Omit this card if IAQTA = 0)

(F10.4, F10.2)

HW

(F10.2)

(Omit this card if IWBC = 0)

QFIX
100.00

(F10.2)

(Omit this card if IWBC = 1)

NELF	SCFAC	FRLEN
2	2.00	0.50

(I10, 2F10.2)

COMPUTER DATA SHEET	PROGRAM STCON1 Steady, one-dimensional flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL
0.50	1000.00	0.00	50.00	0.20

(5F10.2)

AK	BK	VCR
10.000	20.000	0.060

(3F10.3)

IGP	IAQTA	IWBC
0	1	0

(3I10)

AGP	BGP	VGP	THGP

(Omit this card if IGP = 0)

(4F10.3)

PA	THA
0.0010	50.00

(F10.4, F10.2)

(Omit this card if IAQTA = 0)

HW

(F10.2)

(Omit this card if IWBC = 0)

QFIX
100.00

(F10.2)

(Omit this card if IWBC = 1)

NELF	SCFAC	FRLEN
2	2.00	0.50

(I10, 2F10.2)

```
*************************************************
*                                               *
*         FINITE ELEMENT ANALYSIS OF            *
*                                               *
*  STEADY STATE,ONE-DIMENSIONAL,CONFINED FLOW.  *
*                                               *
*************************************************
```

```
*************************************************
*              PROBLEM NUMBER  #      1         *
*************************************************
```

GENERAL INPUT DATA

AQUIFER AND WELL CHARACTERISTICS

RADIUS OF WELL # 1.000

RADIUS OF INFLUENCE # 1000.000

HEIGHT OF WATER TABLE # 0.0

THICKNESS OF AQUIFER # 50.000

TOLERANCE FOR SUCCESSIVE HEAD VALUES # 0.200

AQUIFER PROPERTIES

COEFFICIENT A # 10.0000

COEFFICIENT B # 20.0000

CRITICAL VELOCITY # 0.0600

COEFFICIENT K # 0.0893

OVER-RELAXATION FACTOR # 1.5000

TOTAL NUMBER OF NODES # 29

TOTAL NUMBER OF ELEMENTS # 14

```
***************************
*     FINAL SOLUTION      *
***************************
```

NO. OF ITERATIONS REQUIRED # 3

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.122

HEAD VERSUS RADIAL DISTANCE

NODE	R-COORDINATE	HEAD VALUE	RHO-COORD	TZI-COORD
1	1.0000	-25.6291	0.0100	1.0000
2	1.1250	-25.0266	0.0112	0.9765
3	1.2500	-24.5060	0.0125	0.9563
4	1.3750	-24.0546	0.0137	0.9386
5	1.5000	-23.6523	0.0150	0.9229
6	1.7500	-22.9649	0.0175	0.8960
7	2.0000	-22.3916	0.0200	0.8737
8	2.2500	-21.6669	0.0225	0.8540
9	2.5000	-21.4631	0.0250	0.8374
10	3.0000	-20.7503	0.0300	0.8095
11	3.5000	-20.1637	0.0350	0.7765
12	4.0000	-19.6670	0.0400	0.7674
13	4.5000	-19.2360	0.0450	0.7506
14	6.5000	-17.9295	0.0650	0.6996
15	8.5000	-16.9714	0.0850	0.6622
16	12.5000	-15.6019	0.1250	0.6083
17	16.5000	-14.6100	0.1650	0.5703
18	24.5000	-13.2065	0.2450	0.5153
19	32.5000	-12.1960	0.3250	0.4759
20	48.5000	-10.7757	0.4850	0.4204
21	64.5000	-9.7571	0.6450	0.3807
22	96.5000	-8.3269	0.9650	0.3249
23	128.5000	-7.3036	1.2850	0.2759
24	192.5000	-5.9620	1.9250	0.2220
25	256.5000	-4.6433	2.5650	0.1966
26	384.5000	-3.4065	3.8450	0.1329
27	512.5000	-2.3796	5.1250	0.0928
28	756.2500	-0.9821	7.5625	0.0330
29	1000.0000	-0.0000	10.0000	0.0000

NODAL VELOCITY PRINT OUT

NODE	COMMON ELEMENTS	VELOCITY
1	1.0	-0.31595
2	1.0	-0.28546
3	2.0	-0.25359
4	1.0	-0.23333
5	2.0	-0.21077
6	1.0	-0.18424
7	2.0	-0.16063
8	1.0	-0.14414
9	2.0	-0.12369
10	1.0	-0.10703
11	2.0	-0.08933
12	1.0	-0.07938
13	2.0	-0.06798
14	1.0	-0.05055
15	2.0	-0.03489
16	1.0	-0.02636
17	2.0	-0.01789
18	1.0	-0.01347
19	2.0	-0.00906
20	1.0	-0.00681
21	2.0	-0.00456
22	1.0	-0.00342
23	2.0	-0.00229
24	1.0	-0.00172
25	2.0	-0.00115
26	1.0	-0.00086
27	2.0	-0.00057
28	1.0	-0.00044
29	1.0	-0.00030

DISCHARGE INTO WELL # 99.9991

```
*****************************************************
*              STEADY STATE TYPE CURVE              *
*****************************************************
```

NON-LINEAR FACTOR # 0.0006

NODE NUMBER	R-COORDINATE	FUNCTION W(U)	ARGUMENT U
1	1.00	0.7192E 01	0.1000E-02
2	1.13	0.7023E 01	0.1125E-02
3	1.25	0.6877E 01	0.1250E-02
4	1.38	0.6750E 01	0.1375E-02
5	1.50	0.6637E 01	0.1500E-02
6	1.75	0.6444E 01	0.1750E-02
7	2.00	0.6283E 01	0.2000E-02
8	2.25	0.6142E 01	0.2250E-02
9	2.50	0.6023E 01	0.2500E-02
10	3.00	0.5823E 01	0.3000E-02
11	3.50	0.5658E 01	0.3500E-02
12	4.00	0.5510E 01	0.4000E-02
13	4.50	0.5398E 01	0.4500E-02
14	6.50	0.5031E 01	0.6500E-02
15	8.50	0.4762E 01	0.8500E-02
16	12.50	0.4378E 01	0.1250E-01
17	16.50	0.4100E 01	0.1650E-01
18	24.50	0.3706E 01	0.2450E-01
19	32.50	0.3423E 01	0.3250E-01
20	48.50	0.3024E 01	0.4850E-01
21	64.50	0.2738E 01	0.6450E-01
22	96.50	0.2337E 01	0.9650E-01
23	128.50	0.2050E 01	0.1285E 00
24	192.50	0.1647E 01	0.1925E 00
25	256.50	0.1359E 01	0.2565E 00
26	384.50	0.9559E 00	0.3845E 00
27	512.50	0.6676E 00	0.5125E 00
28	756.25	0.2801E 00	0.7562E 00
29	1000.00	0.6929E-21	0.1000E 01

```
******************************************************
*            PROBLEM NUMBER    #     2               *
******************************************************
```

GENERAL INPUT DATA

AQUIFER AND WELL CHARACTERISTICS

RADIUS OF WELL # 0.500

RADIUS OF INFLUENCE # 1000.000

HEIGHT OF WATER TABLE # 0.0

THICKNESS OF AQUIFER # 50.000

TOLERANCE FOR SUCCESSIVE HEAD VALUES # 0.200

AQUIFER PROPERTIES

COEFFICIENT A # 10.0000

COEFFICIENT B # 20.0000

CRITICAL VELOCITY # 0.0600

COEFFICIENT K # 0.0893

AQUITARD PROPERTIES

HYDRAULIC CONDUCTIVITY # 0.00100

THICKNESS # 50.00

OVER-RELAXATION FACTOR # 1.5000

TOTAL NUMBER OF NODES # 29

TOTAL NUMBER OF ELEMENTS # 14


```
**************************
*     FINAL SOLUTION     *
**************************
```

NO. OF ITERATIONS REQUIRED # 3

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.114

HEAD VERSUS RADIAL DISTANCE

NODE	R-COORDINATE	HEAD VALUE	RHO-COORD	TZI-COORD
1	0.5000	-27.4126	0.0050	1.0000
2	0.6250	-25.9207	0.0062	0.9456
3	0.7500	-24.8052	0.0075	0.9049
4	0.8750	-23.9300	0.0087	0.8732
5	1.0000	-23.2132	0.0100	0.8460
6	1.2500	-22.0944	0.0125	0.8069
7	1.5000	-21.2372	0.0150	0.7747
8	1.7500	-20.5499	0.0175	0.7497
9	2.0000	-19.9576	0.0200	0.7280
10	2.5000	-19.0505	0.0250	0.6950
11	3.0000	-18.3363	0.0300	0.6682
12	3.5000	-17.7505	0.0350	0.6475
13	4.0000	-17.2534	0.0400	0.6294
14	6.0000	-15.8144	0.0600	0.5749
15	8.0000	-14.7867	0.0800	0.5394
16	12.0000	-13.3486	0.1200	0.4870
17	16.0000	-12.3223	0.1600	0.4495
18	24.0000	-10.9976	0.2400	0.3972
19	32.0000	-9.7656	0.3200	0.3593
20	48.0000	-8.4417	0.4800	0.3079
21	64.0000	-7.4329	0.6400	0.2711
22	96.0000	-6.0417	0.9600	0.2204
23	128.0000	-5.0716	1.2800	0.1850
24	192.0000	-3.7703	1.9200	0.1375
25	256.0000	-2.6997	2.5600	0.1053
26	384.0000	-1.9082	3.8400	0.0669
27	512.0000	-1.1413	5.1200	0.0416
28	756.0000	-0.4231	7.5600	0.0154
29	1000.0000	-0.0000	10.0000	0.0000

NODAL VELOCITY PRINT OUT

NODE	COMMON ELEMENTS	VELOCITY
1	1.0	-0.60704
2	1.0	-0.51417
3	2.0	-0.41244
4	1.0	-0.36716
5	2.0	-0.31125
6	1.0	-0.26000
7	2.0	-0.20674
8	1.0	-0.18641
9	2.0	-0.15381
10	1.0	-0.12390
11	2.0	-0.10342
12	1.0	-0.09153
13	2.0	-0.07585
14	1.0	-0.05506
15	2.0	-0.03669
16	1.0	-0.02750
17	2.0	-0.01831
18	1.0	-0.01371
19	2.0	-0.00910
20	1.0	-0.00679
21	2.0	-0.00447
22	1.0	-0.00329
23	2.0	-0.00212
24	1.0	-0.00151
25	2.0	-0.00091
26	1.0	-0.00061
27	2.0	-0.00032
28	1.0	-0.00021
29	1.0	-0.00010

DISCHARGE INTO WELL # 99.9982

****
* STEADY STATE TYPE CURVE *
****

NON-LINEAR FACTOR # 0.6012

NODE NUMBER	R-COORDINATE	FUNCTION W(U)	ARGUMENT U
1	0.50	0.7692E 01	0.1053E-02
2	0.63	0.7274E 01	0.1323E-02
3	0.75	0.6961E 01	0.1537E-02
4	0.88	0.6715E 01	0.1852E-02
5	1.00	0.6514E 01	0.2117E-02
6	1.25	0.6200E 01	0.2646E-02
7	1.50	0.5960E 01	0.3175E-02
8	1.75	0.5757E 01	0.3704E-02
9	2.00	0.5600E 01	0.4233E-02
10	2.50	0.5346E 01	0.5292E-02
11	3.00	0.5145E 01	0.6350E-02
12	3.50	0.4981E 01	0.7409E-02
13	4.00	0.4842E 01	0.8466E-02
14	6.00	0.4438E 01	0.1270E-01
15	8.00	0.4149E 01	0.1693E-01
16	12.00	0.3746E 01	0.2540E-01
17	16.00	0.3458E 01	0.3387E-01
18	24.00	0.3055E 01	0.5080E-01
19	32.00	0.2765E 01	0.6773E-01
20	48.00	0.2369E 01	0.1016E 00
21	64.00	0.2086E 01	0.1355E 00
22	96.00	0.1695E 01	0.2032E 00
23	128.00	0.1423E 01	0.2709E 00
24	192.00	0.1053E 01	0.4064E 00
25	256.00	0.8137E 00	0.5419E 00
26	384.00	0.5074E 00	0.8128E 00
27	512.00	0.3203E 00	0.1084E 01
28	756.00	0.1187E 00	0.1600E 01
29	1000.00	0.3613E-21	0.2117E 01

3.3 TRCON1 - PROGRAM FOR SOLVING TRANSIENT, ONE-DIMENSIONAL FLOW

The program TRCON1 is used to analyse transient, one-dimensional flow in a single confined or unconfined aquifer or in an aquifer-aquitard system referred to in Section 3.2. When the single aquifer is unconfined, only radial flow components are considered and Boulton's concept of delayed yield index and assumption of constant saturated thickness are applied.

Figures 10 and 11 show the input data form and completed form for a sample run respectively. One problem was solved by the program, with IDISCR = 0 and IVEL = 1. The problem involves transient, two-regime flow in a confined aquifer. Numerical solutions were obtained for five time steps. A major part of the printed output is shown on pages 42 to 43. It consists of a reformatted playback of the input data, the discretisation data generated by the program, and two tables of results for time values at the mid-point and the end of the fifth time step. Each table contains the listing of node numbers, radial co-ordinates, head values, nodal values of the dimensionless time ($1/u = 4Kt/r^2S_s$)and nodal values of the dimensionless drawdown ($W(u) = 4\pi K m s/Q$).

A number of punched cards were also produced by the program. The information contained in these cards is nodal values of the hydraulic head and flux at the end of the final time step. By adding the punched cards to the input data deck, and changing the values of the starting time step and the number of time steps to ITST = 6 and NTICR = 20, the program may be rerun to obtain solutions for later times, starting from step no. 6 onwards.

From the printed output, the type curve of $W(u)$ versus $1/u$ for a particular nodal point may be obtained by plotting the nodal values at various times. It may also be noted that all head values are measured from the initial position of the water table, as HO is set equal to 0.0 ft.

FIGURE 10: INPUT DATA FORM FOR TRCON1

COMPUTER DATA SHEET	PROGRAM TRCON1 Transient, one-dimensional flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX	
				(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL	
					(5F10.2)

QFIX	
	(F10.2)

IWAT	IAQTA	IGP	IBOUND	IWBC	
					(5I10)

RCSNG	QRTOL	
		(2F10.2)

(Omit this card if IWBC = 0)

AK	BK	VCR	SS	
				(3F10.3, E10.2)

AGP	BGP	VGP	SSP	THGP	
					(3F10.3, E10.2, F10.2)

(Omit this card if IGP = 0)

SY	DINDEX	
		(2E10.2)

(Omit this card if IWAT = 0)

PA	THA	
		(F10.4, F10.2)

(Omit this card if IAQTA = 0)

NTICR	TFACTR	TMUL	DTMUL	
				(I10, 3F10.3)

NELF	SCFAC	FRLEN	
			(I10, 2F10.2)

FIGURE 11: COMPLETED FORM FOR SAMPLE PROBLEM

COMPUTER DATA SHEET	PROGRAM TRCON1 Transient, one-dimensional flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
1	0	1	1.50

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL
1.00	1000.00	0.00	20.00	0.20

(5F10.2)

QFIX
280.00

(F10.2)

IWAT	IAQTA	IGP	IBOUND	IWBC
0	0	0	0	0

(5I10)

RCSNG	QRTOL

(2F10.2)

(Omit this card if IWBC = 0)

AK	BK	VCR	SS
10.000	20.000	0.060	0.10E-04

(3F10.3, E10.2)

AGP	BGP	VGP	SSP	THGP

(3F10.3, E10.2, F10.2)

(Omit this card if IGP = 0)

SY	DINDEX

(2E10.2)

(Omit this card if IWAT = 0)

PA	THA

(F10.4, F10.2)

(Omit this card if IAQTA = 0)

NTICR	TFACTR	TMUL	DTMUL
5	50.000	1.500	0.000

(I10, 3F10.3)

NELF	SCFAC	FRLEN
2	2.00	0.50

(I10, 2F10.2)

```
**********************************************
*                                            *
*    FINITE ELEMENT ANALYSIS OF WELL PROBLEMS *
*                                            *
* TRANSIENT,TWO REGIME FLOW TOWARD A SINGLE WELL *
*                                            *
**********************************************
```

```
************************************************
*              PROBLEM NUMBER  #     1          *
************************************************
```

AQUIFER AND WELL CHARACTERISTICS

RADIUS OF WELL # 1.000

RADIUS OF INFLUENCE # 1000.000

HEIGHT OF WATER TABLE # 0.0

THICKNESS OF AQUIFER # 20.000

TOLERANCE FOR SUCCESSIVE HEAD VALUES # 0.200

PRESCRIBED WELL DISCHARGE # 280.000

AQUIFER PROPERTIES

COEFFICIENT A # 10.0000

COEFFICIENT B # 20.0000

CRITICAL VELOCITY # 0.0600

COEFFICIENT K # 0.0893

SPECIFIC STORAGE # 0.10E-04

GENERAL TIME DATA

NUMBER OF TIME INCREMENTS # 5

TIME FACTOR # 50.0000

TIME MULTIPLIER # 1.5000

INCREMENT OF MULTIPLIER # 0.0

DISCRETISATION DATA

NUMBER OF ELEMENTS IN FIRST SUBREGION # 2

SCALE FACTOR FOR ELEMENT LENGTH # 2.000

LENGTH OF FIRST SUBREGION # 0.500

SUBREGION	NO. OF ELEMENTS	LENGTH OF SUBREGION
1	2	0.500
2	2	1.000
3	2	2.000
4	1	4.000
5	1	8.000
6	1	16.000
7	1	32.000
8	1	64.000
9	1	128.000
10	1	256.000
11	1	487.500

TOTAL NO. OF NODES # 29

TOTAL NO. OF ELEMENTS # 14

MESH STABILITY FACTOR # 0.1092E-05

```
*************************************
*    TIME STEP NUMBER  #    5       *
*************************************
```

```
****************************************
*         TIME  #     0.528E-01        *
****************************************
```

ESTIMATED RADIUS OF INFLUENCE # 256.50

CORRESPONDING NO. OF NODES # 25

CORRESPONDING NO. OF ELEMENTS # 12

NO. OF ITERATIONS REQUIRED FOR FINAL SOLUTION # 5

MAXIMUM ABSOLUTE ERROR IN HEAD # 0.160

```
*********************************************
*                                           *
*         FINAL RESULTS OF ANALYSIS         *
*                                           *
*********************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	HEAD	1/U	W(U)
1	1.00	-169.0469	0.1886E 04	13.5533
2	1.13	-155.4840	0.1490E 04	12.4659
3	1.25	-144.3705	0.1207E 04	11.5749
4	1.38	-135.0735	0.9974E 03	10.8295
5	1.50	-127.1537	0.8381E 03	10.1945
6	1.75	-114.3392	0.6158E 03	9.1671
7	2.00	-104.3089	0.4714E 03	8.3629
8	2.25	-96.2197	0.3725E 03	7.7144
9	2.50	-89.5006	0.3017E 03	7.1757
10	3.00	-78.9294	0.2095E 03	6.3281
11	3.50	-70.8482	0.1539E 03	5.6802
12	4.00	-64.4315	0.1179E 03	5.1658
13	4.50	-59.1500	0.9313E 02	4.7423
14	6.50	-44.7660	0.4463E 02	3.5891
15	8.50	-35.6278	0.2610E 02	2.8565
16	12.50	-24.5280	0.1207E 02	1.9665
17	16.50	-17.6704	0.6927E 01	1.4167
18	24.50	-9.8832	0.3142E 01	0.7924
19	32.50	-5.6613	0.1785E 01	0.4539
20	48.50	-1.7956	0.8017E 00	0.1440
21	64.50	-0.4864	0.4533E 00	0.0390
22	96.50	-0.0319	0.2025E 00	0.0026
23	128.50	-0.0062	0.1142E 00	0.0005
24	192.50	0.0003	0.5089E-01	-0.0000
25	256.50	-0.0000	0.2866E-01	0.0000

```
****************************************
*         TIME  #     0.634E-01        *
****************************************
```

NO. OF ITERATIONS REQUIRED FOR FINAL SOLUTION # 5

MAXIMUM ABSOLUTE ERROR IN HEAD # 0.172

```
*********************************************
*                                           *
*         FINAL RESULTS OF ANALYSIS         *
*                                           *
*********************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	HEAD	1/U	W(U)
1	1.00	-171.8602	0.2263E 04	13.7789
2	1.13	-158.3152	0.1788E 04	12.6929
3	1.25	-147.2058	0.1448E 04	11.6022
4	1.38	-137.9157	0.1197E 04	11.0574
5	1.50	-129.9563	0.1006E 04	10.4224
6	1.75	-117.1809	0.7387E 03	9.3950
7	2.00	-107.1464	0.5657E 03	8.5904
8	2.25	-99.0504	0.4470E 03	7.9413
9	2.50	-92.3238	0.3621E 03	7.4020
10	3.00	-81.7361	0.2514E 03	6.5532
11	3.50	-73.6370	0.1847E 03	5.9033
12	4.00	-67.2011	0.1414E 03	5.3878
13	4.50	-61.8996	0.1118E 03	4.9623
14	6.50	-47.4277	0.5356E 02	3.8025
15	8.50	-38.1939	0.3132E 02	3.0622
16	12.50	-26.9044	0.1443E 02	2.1571
17	16.50	-19.8114	0.8312E 01	1.5984
18	24.50	-11.5821	0.3770E 01	0.9296
19	32.50	-6.9600	0.2142E 01	0.5530
20	48.50	-2.4602	0.9620E 00	0.1972
21	64.50	-0.7621	0.5439E 00	0.0611
22	96.50	-0.0672	0.2430E 00	0.0054
23	128.50	-0.0126	0.1370E 00	0.0010
24	192.50	0.0006	0.6107E-01	-0.0000
25	256.50	-0.0000	0.3440E-01	0.0000

3.4 STCON3 - PROGRAM FOR SOLVING STEADY, TWO-DIMENSIONAL FLOW USING TRIANGULAR ELEMENTS

The program STCON3 is used to analyse steady two-dimensional flow in a confined aquifer or in an aquifer-aquitard system, referred to in Section 3.2. It can handle the situation involving a partially or fully penetrating well with or without a gravel pack. The well may be partly or fully screened and up to 5 screened intervals may be used.

Figures 12 and 13 show the input data form and completed form for a sample run. One problem was solved by the program, with IDISCR = 1 and IVEL = 1. The problem involves steady, two-regime flow towards a fully penetrating but partially screened well. Results were obtained for a value of HW equal to -50.ft. All values of the hydraulic head are measured from the position of the water table at the external radius. The printed output is shown on pages 48 to 57. It consists of the following:

(i) a reformatted playback of the input data;

(ii) the generated discretisation data, namely node connections of elements, nodal co-ordinates, prescribed nodes and values and the length of the gross vector;

(iii) the first table of results for HW = -50.ft., which lists the node numbers, radial and vertical co-ordinates, head values, Rho- and Tzi- co-ordinates which are defined as $\rho = r/2m$ and $\xi = z/m$ and values of the dimensionless drawdown s/s_w ;

(iv) the second table of results, which lists the element numbers, radial and vertical components of element velocities and values of the flow type index, IDARCY;

(v) the third table of results, which lists node numbers of the nodes on well screen, corresponding values of discharge flux, and the total discharge into the well;

(vi) the final table of results, which lists the non-linear factor $\lambda = bQK/2\pi m r_0$, the node numbers, radial co-ordinates, values of $W(u) = 2\pi K m/Q$ and values of $1/u = r/r_0$.

When using STCON3, the user should at first familiarise himself with the discretisation data generated by the program. A plot of the mesh for the sample problem is shown in Figure 14. Only a major part of this mesh is included together with the node numbers.

FIGURE 12: INPUT DATA FORM FOR STCON3

COMPUTER DATA SHEET	PROGRAM STCON3 Steady, two-dimensional, confined flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL

(5F10.2)

IGP	IAQTA

(2I10)

AK	BK	VCR

(3F10.3)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

PA	THA

(F10.4, F10.2)

(Omit this card if IAQTA = 0)

HW

(F10.2)

FRLEN	SCFAC	XLMAX	IREG	NMIN

(3F10.2, 2I10)

IPENTR	NDSC	NSCREN

(3I10)

XSCR	HSCR

(2F10.2)

(No. of cards = NSCREN)

FIGURE 13: COMPLETED FORM FOR SAMPLE PROBLEM

COMPUTER DATA SHEET	PROGRAM STCON3 Steady, two-dimensional, confined flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
1	1	1	1.50

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL
1.00	100.00	0.00	20.00	0.20

(5F10.2)

IGP	IAQTA
0	0

(2I10)

AK	BK	VCR
10.000	20.000	0.060

(3F10.3)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

PA	THA

(Omit this card if IAQTA = 0)

(F10.4, F10.2)

HW
-50.00

(F10.2)

FRLEN	SCFAC	XLMAX	IREG	NMIN
0.50	1.50	50.00	4	2

(3F10.2, 2I10)

IPENTR	NDSC	NSCREN
0	5	1

(3I10)

XSCR	HSCR
10.00	5.00

(2F10.2)

(No. of cards = NSCREN)

```
************************************************
*                                              *
*        FINITE ELEMENT SOLUTION OF            *
*                                              *
*   STEADY STATE,TWO-DIMENSIONAL CONFINED FLOW *
*                                              *
************************************************
```

```
*******************************************************
*                PROBLEM NUMBER   #     1             *
*******************************************************
```

GENERAL INPUT DATA

OVER-RELAXATION FACTOR # 1.50

GRAVEL PACK INDEX # 0

AQUITARD INDEX # 0

AQUIFER AND WELL CHARACTERISTICS

PERM. CONSTANT A # 10.000

PERM. CONSTANT B # 20.000

RADIUS OF WELL # 1.000

RADIUS OF INFLUENCE # 100.000

HEIGHT OF WATER TABLE # 0.0

THICKNESS OF AQUIFER # 20.000

TOLERANCE FOR SUCCESSIVE HEAD VALUES # 0.200

SCREEN NO.	BASE HEIGHT	LENGTH
1	10.00	5.00

PENETRATION INDEX # 0

TOTAL NUMBER OF NODES ON WELL SCREEN(S) # 5

DISCRETIZATION DATA

NUMBER OF NODES AT WELL BOUNDARY # 17

TOTAL NUMBER OF REGIONS # 12

IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS

ELEM NO	NODE1	NODE2	NODE3	REPETITION NUMBER
1	1	18	2	1
2	2	19	3	1
3	3	20	4	1
4	4	21	5	1
5	5	22	6	1
6	6	23	7	1
7	7	24	8	1
8	8	25	9	1
9	9	26	10	1
10	10	27	11	1
11	11	28	12	1
12	12	29	13	1
13	13	30	14	1
14	14	31	15	1
15	15	32	16	1
16	16	33	17	1
17	2	18	19	2
18	3	19	20	2

19	4	20	21	2
20	5	21	22	2
21	6	22	23	2
22	7	23	24	2
23	8	24	25	2
24	9	25	26	2
25	10	26	27	2
26	11	27	28	2
27	12	28	29	2
28	13	29	30	2
29	14	30	31	2
30	15	31	32	2
31	16	32	33	2
32	17	33	34	2
33	18	35	19	3
34	19	36	20	3
35	20	37	21	3
36	21	38	22	3
37	22	39	23	3
38	23	40	24	3
39	24	41	25	3
40	25	42	26	3
41	26	43	27	3
42	27	44	28	3
43	28	45	29	3
44	29	46	30	3
45	30	47	31	3
46	31	48	32	3
47	32	49	33	3
48	33	50	34	3
49	19	35	36	4
50	20	36	37	4
51	21	37	38	4
52	22	38	39	4
53	23	39	40	4
54	24	40	41	4
55	25	41	42	4
56	26	42	43	4
57	27	43	44	4
58	28	44	45	4
59	29	45	46	4
60	30	46	47	4
61	31	47	48	4
62	32	48	49	4
63	33	49	50	4
64	34	50	51	4
65	35	52	36	5
66	36	53	37	5
67	37	54	38	5
68	38	55	39	5
69	39	56	40	5
70	40	57	41	5
71	41	58	42	5
72	42	59	43	5
73	43	60	44	5
74	44	61	45	5
75	45	62	46	5
76	46	63	47	5
77	47	64	48	5
78	48	65	49	5
79	49	66	50	5
80	50	67	51	5
81	36	52	53	6
82	37	53	54	6
83	38	54	55	6
84	39	55	56	6
85	40	56	57	6
86	41	57	58	6
87	42	58	59	6
88	43	59	60	6
89	44	60	61	6
90	45	61	62	6
91	46	62	63	6
92	47	63	64	6
93	48	64	65	6
94	49	65	66	6
95	50	66	67	6
96	51	67	68	6
97	52	69	53	7
98	54	70	55	7
99	56	71	57	7
100	58	72	59	7
101	60	73	61	7
102	62	74	63	7
103	64	75	65	7
104	66	76	67	7
105	53	69	70	8
106	55	70	71	8

107	57	71	72	8
108	59	72	73	8
109	61	73	74	8
110	63	74	75	8
111	65	75	76	8
112	67	76	77	8
113	53	70	54	9
114	55	71	56	9
115	57	72	58	9
116	59	73	60	9
117	61	74	62	9
118	63	75	64	9
119	65	76	66	9
120	67	77	68	9
121	69	78	70	10
122	70	79	71	10
123	71	80	72	10
124	72	81	73	10
125	73	82	74	10
126	74	83	75	10
127	75	84	76	10
128	76	85	77	10
129	70	78	79	11
130	71	79	80	11
131	72	80	81	11
132	73	81	82	11
133	74	82	83	11
134	75	83	84	11
135	76	84	85	11
136	77	85	86	11
137	78	87	79	12
138	79	88	80	12
139	80	89	81	12
140	81	90	82	12
141	82	91	83	12
142	83	92	84	12
143	84	93	85	12
144	85	94	86	12
145	79	87	88	13
146	80	88	89	13
147	81	89	90	13
148	82	90	91	13
149	83	91	92	13
150	84	92	93	13
151	85	93	94	13
152	86	94	95	13
153	87	96	88	14
154	88	97	89	14
155	89	98	90	14
156	90	99	91	14
157	91	100	92	14
158	92	101	93	14
159	93	102	94	14
160	94	103	95	14
161	88	96	97	15
162	89	97	98	15
163	90	98	99	15
164	91	99	100	15
165	92	100	101	15
166	93	101	102	15
167	94	102	103	15
168	95	103	104	15
169	96	105	97	16
170	98	106	99	16
171	100	107	101	16
172	102	108	103	16
173	97	105	105	17
174	99	106	107	17
175	101	107	108	17
176	103	108	109	17
177	97	106	98	18
178	99	107	100	18
179	101	108	102	18
180	103	109	104	18
181	105	110	106	19
182	106	111	107	19
183	107	112	108	19
184	108	113	109	19
185	106	110	111	20
186	107	111	112	20
187	108	112	113	20
188	109	113	114	20
189	110	115	111	21
190	111	116	112	21
191	112	117	113	21
192	113	118	114	21
193	111	115	116	22
194	112	116	117	22
195	113	117	118	22
196	114	118	119	22
197	115	120	116	23
198	116	121	117	23
199	117	122	118	23
200	118	123	119	23
201	116	120	121	24
202	117	121	122	24
203	118	122	123	24
204	119	123	124	24
205	120	125	121	25
206	122	126	123	25
207	121	125	126	26
208	123	126	127	26
209	121	126	122	27
210	123	127	124	27

NODE DATA

NODE	R-COORD	Z-COORD
1	1.00	0.0
2	1.00	1.25
3	1.00	2.50
4	1.00	3.75
5	1.00	5.00
6	1.00	6.25
7	1.00	7.50
8	1.00	8.75
9	1.00	10.00
10	1.00	11.25
11	1.00	12.50
12	1.00	13.75
13	1.00	15.00
14	1.00	16.25
15	1.00	17.50
16	1.00	18.75
17	1.00	20.00
18	1.50	0.0
19	1.50	1.25
20	1.50	2.50
21	1.50	3.75
22	1.50	5.00
23	1.50	6.25
24	1.50	7.50
25	1.50	8.75
26	1.50	10.00
27	1.50	11.25
28	1.50	12.50
29	1.50	13.75
30	1.50	15.00
31	1.50	16.25
32	1.50	17.50
33	1.50	18.75
34	1.50	20.00
35	2.25	0.0
36	2.25	1.25
37	2.25	2.50
38	2.25	3.75
39	2.25	5.00
40	2.25	6.25
41	2.25	7.50
42	2.25	8.75
43	2.25	10.00
44	2.25	11.25
45	2.25	12.50
46	2.25	13.75
47	2.25	15.00
48	2.25	16.25
49	2.25	17.50
50	2.25	18.75
51	2.25	20.00
52	3.38	0.0
53	3.38	1.25
54	3.38	2.50
55	3.38	3.75
56	3.38	5.00
57	3.38	6.25
58	3.38	7.50
59	3.38	8.75
60	3.38	10.00
61	3.38	11.25
62	3.38	12.50
63	3.38	13.75
64	3.38	15.00
65	3.38	16.25
66	3.38	17.50
67	3.38	18.75
68	3.38	20.00
69	5.06	0.0
70	5.06	2.50
71	5.06	5.00
72	5.06	7.50
73	5.06	10.00
74	5.06	12.50
75	5.06	15.00
76	5.06	17.50
77	5.06	20.00
78	7.59	0.0
79	7.59	2.50
80	7.59	5.00
81	7.59	7.50
82	7.59	10.00
83	7.59	12.50
84	7.59	15.00
85	7.59	17.50
86	7.59	20.00

87	11.39	0.0
88	11.39	2.50
89	11.39	5.00
90	11.39	7.50
91	11.39	10.00
92	11.39	12.50
93	11.39	15.00
94	11.39	17.50
95	11.39	20.00
96	17.09	0.0
97	17.09	2.50
98	17.09	5.00
99	17.09	7.50
100	17.09	10.00
101	17.09	12.50
102	17.09	15.00
103	17.09	17.50
104	17.09	20.00
105	25.63	0.0
106	25.63	5.00
107	25.63	10.00
108	25.63	15.00
109	25.63	20.00
110	38.44	0.0
111	38.44	5.00
112	38.44	10.00
113	38.44	15.00
114	38.44	20.00
115	51.40	0.0
116	51.40	5.00
117	51.40	10.00
118	51.40	15.00
119	51.40	20.00
120	70.84	0.0
121	70.84	5.00
122	70.84	10.00
123	70.84	15.00
124	70.84	20.00
125	100.00	0.0
126	100.00	5.00
127	100.00	10.00
128	100.00	15.00
129	100.00	20.00

TOP BOUNDARY NODES AND RADIAL COORDINATES.

NODE NUMBER	R-COORDINATE
17	1.00
34	1.50
51	2.25
68	3.38
77	5.06
86	7.59
95	11.39
104	17.09
109	25.63
114	38.44
119	51.40
124	70.84
129	100.00

PRESCRIBED NODES AND VALUES

NODES	PRESCRIBED VALUES
9	-50.000
10	-50.000
11	-50.000
12	-50.000
13	-50.000
125	0.0
126	0.0
127	0.0
128	0.0
129	0.0

LENGTH OF GROSS VECTOR # 1626

```
***************************************
*   WATER LEVEL IN THE WELL # -50.00   *
***************************************
```

NO. OF ITERATIONS REQUIRED # 6

MAXIMUM ERROR IN HEAD # 0.102

FINAL SOLUTION

NODE	R-COORD	Z-COORD	HEAD	RHO-COORD	TZI-COORD	GRAD
1	1.00	0.0	-11.9660	0.0250	0.0	0.23
2	1.00	1.25	-12.0421	0.0250	0.0625	0.24
3	1.00	2.50	-12.2839	0.0250	0.1250	0.245
4	1.00	3.75	-12.7394	0.0250	0.1875	0.254
5	1.00	5.00	-13.4645	0.0250	0.2500	0.26
6	1.00	6.25	-14.6513	0.0250	0.3125	0.29
7	1.00	7.50	-16.6120	0.0250	0.3750	0.336
8	1.00	8.75	-21.9315	0.0250	0.4375	0.43
9	1.00	10.00	-50.0000	0.0250	0.5000	1.00
10	1.00	11.25	-50.0000	0.0250	0.5625	1.0
11	1.00	12.50	-50.0000	0.0250	0.6250	1.00
12	1.00	13.75	-50.0000	0.0250	0.6875	1.0
13	1.00	15.00	-50.0000	0.0250	0.7500	1.0
14	1.00	16.25	-22.3495	0.0250	0.8125	0.447
15	1.00	17.50	-18.2349	0.0250	0.8750	0.36
16	1.00	18.75	-16.7798	0.0250	0.9375	0.335
17	1.00	20.00	-16.3757	0.0250	1.0000	0.327
18	1.50	0.0	-11.9538	0.0375	0.0	0.23
19	1.50	1.25	-12.0294	0.0375	0.0625	0.24
20	1.50	2.50	-12.2737	0.0375	0.1250	0.245
21	1.50	3.75	-12.7139	0.0375	0.1875	0.254
22	1.50	5.00	-13.4323	0.0375	0.2500	0.25
23	1.50	6.25	-14.5900	0.0375	0.3125	0.291
24	1.50	7.50	-16.6423	0.0375	0.3750	0.332
25	1.50	8.75	-20.9202	0.0375	0.4375	0.41
26	1.50	10.00	-33.3390	0.0375	0.5000	0.66
27	1.50	11.25	-36.3811	0.0375	0.5625	0.72
28	1.50	12.50	-37.0749	0.0375	0.6250	0.741
29	1.50	13.75	-36.5103	0.0375	0.6875	0.730
30	1.50	15.00	-33.4662	0.0375	0.7500	0.66
31	1.50	16.25	-21.6589	0.0375	0.8125	0.433
32	1.50	17.50	-18.0820	0.0375	0.8750	0.361
33	1.50	18.75	-16.7064	0.0375	0.9375	0.334
34	1.50	20.00	-16.3192	0.0375	1.0000	0.326
35	2.25	0.0	-11.9006	0.0562	0.0	0.23
36	2.25	1.25	-11.9739	0.0562	0.0625	0.239
37	2.25	2.50	-12.2075	0.0562	0.1250	0.244
38	2.25	3.75	-12.6393	0.0562	0.1875	0.252
39	2.25	5.00	-13.2973	0.0562	0.2500	0.265
40	2.25	6.25	-14.3378	0.0562	0.3125	0.286
41	2.25	7.50	-16.0493	0.0562	0.3750	0.321
42	2.25	8.75	-18.9544	0.0562	0.4375	0.379
43	2.25	10.00	-24.1160	0.0562	0.5000	0.462
44	2.25	11.25	-26.5703	0.0562	0.5625	0.531
45	2.25	12.50	-27.3741	0.0562	0.6250	0.54
46	2.25	13.75	-26.7731	0.0562	0.6875	0.535
47	2.25	15.00	-24.4462	0.0562	0.7500	0.48
48	2.25	16.25	-19.9040	0.0562	0.8125	0.39
49	2.25	17.50	-17.5131	0.0562	0.8750	0.35
50	2.25	18.75	-16.4037	0.0562	0.9375	0.32
51	2.25	20.00	-16.0787	0.0562	1.0000	0.321
52	3.38	0.0	-11.7576	0.0844	0.0	0.235
53	3.38	1.25	-11.8271	0.0844	0.0625	0.23
54	3.38	2.50	-12.0335	0.0844	0.1250	0.24
55	3.38	3.75	-12.4026	0.0844	0.1875	0.24
56	3.38	5.00	-12.9641	0.0844	0.2500	0.25
57	3.38	6.25	-13.7674	0.0844	0.3125	0.275
58	3.38	7.50	-14.9696	0.0844	0.3750	0.29
59	3.38	8.75	-16.5327	0.0844	0.4375	0.33
60	3.38	10.00	-18.6721	0.0844	0.5000	0.373
61	3.38	11.25	-19.8666	0.0844	0.5625	0.39
62	3.33	12.50	-20.5107	0.0844	0.6250	0.41
63	3.38	13.75	-20.1016	0.0844	0.6875	0.402
64	3.38	15.00	-19.1690	0.0844	0.7500	0.38
65	3.38	15.25	-17.5178	0.0844	0.8125	0.35
66	3.33	17.50	-16.3954	0.0844	0.8750	0.32
67	3.38	18.75	-15.6928	0.0844	0.9375	0.31
68	3.38	20.00	-15.4912	0.0844	1.0000	0.30
69	5.06	0.0	-11.4240	0.1266	0.0	0.22
70	5.06	2.50	-11.6239	0.1266	0.1250	0.232
71	5.06	5.00	-12.2912	0.1266	0.2500	0.24
72	5.06	7.50	-13.4670	0.1266	0.3750	0.25
73	5.06	10.00	-14.9950	0.1266	0.5000	0.29
74	5.06	12.50	-15.8694	0.1266	0.6250	0.317
75	5.06	15.00	-15.5438	0.1266	0.7500	0.31
76	5.06	17.50	-14.6640	0.1266	0.8750	0.293

77	5.06	20.00	-14.2564	0.1266	1.0000	[illegible]
78	7.59	0.0	-10.7774	0.1898	0.0	0.215
79	7.59	2.50	-10.8954	0.1898	0.1250	0.217
80	7.59	5.00	-11.2539	0.1898	0.2500	[illegible]
81	7.59	7.50	-11.8153	0.1898	0.3750	0.236
82	7.59	10.00	-12.4225	0.1898	0.5000	[illegible]
83	7.59	12.50	-12.9051	0.1898	0.6250	0.255
84	7.59	15.00	-12.8318	0.1898	0.7500	0.256
85	7.59	17.50	-12.6659	0.1898	0.8750	0.253
86	7.59	20.00	-12.5604	0.1898	1.0000	0.251
87	11.39	0.0	-9.7195	0.2848	0.0	[illegible]
88	11.39	2.50	-9.7685	0.2848	0.1250	[illegible]
89	11.39	5.00	-9.9135	0.2848	0.2500	[illegible]
90	11.39	7.50	-10.1215	0.2848	0.3750	[illegible]
91	11.39	10.00	-10.3395	0.2848	0.5000	[illegible]
92	11.39	12.50	-10.5013	0.2848	0.6250	0.210
93	11.39	15.00	-10.5779	0.2848	0.7500	0.211
94	11.39	17.50	-10.5910	0.2848	0.8750	0.211
95	11.39	20.00	-10.5837	0.2848	1.0000	0.211
96	17.09	0.0	-8.2733	0.4271	0.0	0.165
97	17.09	2.50	-8.2905	0.4271	0.1250	0.165
98	17.09	5.00	-8.3392	0.4271	0.2500	0.166
99	17.09	7.50	-8.3913	0.4271	0.3750	0.167
100	17.09	10.00	-8.4611	0.4271	0.5000	[illegible]
101	17.09	12.50	-8.5091	0.4271	0.6250	0.170
102	17.09	15.00	-8.5560	0.4271	0.7500	0.171
103	17.09	17.50	-8.5725	0.4271	0.8750	0.171
104	17.09	20.00	-8.5922	0.4271	1.0000	0.171
105	25.63	0.0	-6.6224	0.6407	0.0	0.132
106	25.63	5.00	-6.6236	0.6407	0.2500	0.132
107	25.63	10.00	-6.6413	0.6407	0.5000	0.132
108	25.63	15.00	-6.6570	0.6407	0.7500	0.133
109	25.63	20.00	-6.6561	0.6407	1.0000	0.133
110	38.44	0.0	-4.8301	0.9611	0.0	0.096
111	38.44	5.00	-4.8327	0.9611	0.2500	0.096
112	38.44	10.00	-4.8367	0.9611	0.5000	0.096
113	38.44	15.00	-4.8407	0.9611	0.7500	0.096
114	38.44	20.00	-4.8432	0.9611	1.0000	0.096
115	51.40	0.0	-3.5315	1.2851	0.0	0.070
116	51.40	5.00	-3.5326	1.2851	0.2500	0.070
117	51.40	10.00	-3.5358	1.2851	0.5000	0.070
118	51.40	15.00	-3.5386	1.2851	0.7500	0.070
119	51.40	20.00	-3.5403	1.2851	1.0000	0.070
120	70.84	0.0	-2.0666	1.7710	0.0	0.041
121	70.84	5.00	-2.0760	1.7710	0.2500	0.041
122	70.84	10.00	-2.1062	1.7710	0.5000	0.042
123	70.84	15.00	-2.1144	1.7710	0.7500	0.042
124	70.84	20.00	-2.1465	1.7710	1.0000	0.042
125	100.00	0.0	0.0	2.5000	0.0	0.0
126	100.00	5.00	0.0	2.5000	0.2500	0.0
127	100.00	10.00	0.0	2.5000	0.5000	0.0
128	100.00	15.00	0.0	2.5000	0.7500	0.0
129	100.00	20.00	0.0	2.5000	1.0000	0.0

```
**************************************************
*              ELEMENT VELOCITIES                *
**************************************************
```

ELEM NO.	RADIAL VEL	VERTIC VEL	IDARCY
1	-0.002	0.005	0
2	-0.002	0.018	0
3	-0.003	0.032	0
4	-0.004	0.052	0
5	-0.005	0.082	1
6	-0.010	0.136	1
7	-0.022	0.267	1
8	-0.075	0.837	1
9	-1.065	0.0	1
10	-0.943	0.0	1
11	-0.914	0.0	1
12	-0.938	0.0	1
13	-0.983	-0.657	1
14	-0.093	-0.222	1
15	-0.025	-0.097	1
16	-0.013	-0.029	0
17	-0.002	0.005	0
18	-0.003	0.017	0
19	-0.004	0.032	0
20	-0.006	0.051	0
21	-0.011	0.030	1
22	-0.027	0.130	1
23	-0.133	0.225	1
24	-1.047	0.312	1
25	-0.942	0.084	1
26	-0.914	0.020	1
27	-0.938	-0.016	1
28	-1.059	-0.078	1
29	-0.070	-0.430	1
30	-0.022	-0.203	1
31	-0.012	-0.093	1
32	-0.010	-0.028	0
33	-0.006	0.005	0
34	-0.007	0.017	0
35	-0.008	0.032	0
36	-0.011	0.051	0
37	-0.016	0.080	1
38	-0.027	0.130	1
39	-0.054	0.232	1
40	-0.130	0.492	1
41	-0.569	0.113	1
42	-0.596	0.025	1
43	-0.592	-0.021	1
44	-0.590	-0.111	1
45	-0.519	-0.407	1
46	-0.157	-0.191	1
47	-0.062	-0.090	1
48	-0.036	-0.028	0
49	-0.007	0.005	0

50	-0.008	0.017	0
51	-0.011	0.030	0
52	-0.016	0.048	0
53	-0.029	0.072	1
54	-0.063	0.109	1
55	-0.178	0.158	1
56	-0.562	0.139	1
57	-0.594	0.089	1
58	-0.592	0.029	1
59	-0.593	-0.022	1
60	-0.552	-0.087	1
61	-0.150	-0.234	1
62	-0.058	-0.146	1
63	-0.035	-0.076	1
64	-0.029	-0.023	0
65	-0.011	0.005	0
66	-0.012	0.017	0
67	-0.014	0.030	0
68	-0.018	0.048	0
69	-0.026	0.072	1
70	-0.041	0.111	1
71	-0.070	0.170	1
72	-0.136	0.260	1
73	-0.295	0.120	1
74	-0.350	0.038	1
75	-0.356	-0.028	1
76	-0.344	-0.108	1
77	-0.276	-0.214	1
78	-0.151	-0.136	1
79	-0.082	-0.073	1
80	-0.056	-0.023	1
81	-0.012	0.005	0
82	-0.014	0.015	0
83	-0.018	0.026	0
84	-0.026	0.040	0
85	-0.044	0.056	1
86	-0.079	0.079	1
87	-0.158	0.092	1
88	-0.297	0.105	1
89	-0.349	0.056	1
90	-0.356	0.030	1
91	-0.349	-0.019	1
92	-0.294	-0.047	1
93	-0.155	-0.097	1
94	-0.081	-0.074	1
95	-0.055	-0.049	1
96	-0.047	-0.014	0
97	-0.018	0.005	0
98	-0.022	0.026	0
99	-0.035	0.057	1
100	-0.071	0.100	1
101	-0.161	0.071	1
102	-0.197	-0.023	1
103	-0.157	-0.097	1
104	-0.086	-0.047	1
105	-0.016	0.007	0
106	-0.024	0.024	0
107	-0.047	0.042	1
108	-0.110	0.049	1
109	-0.190	0.025	1
110	-0.189	-0.009	1
111	-0.116	-0.028	1
112	-0.065	-0.014	1
113	-0.022	0.015	0
114	-0.036	0.040	0
115	-0.073	0.079	1
116	-0.156	0.123	1
117	-0.196	0.037	1
118	-0.160	-0.056	1
119	-0.084	-0.073	1
120	-0.065	-0.014	1
121	-0.023	0.007	0
122	-0.026	0.024	0
123	-0.036	0.042	0
124	-0.057	0.053	1
125	-0.086	0.030	1
126	-0.101	-0.011	1
127	-0.090	-0.030	1
128	-0.069	-0.014	1
129	-0.026	0.004	0
130	-0.036	0.013	0
131	-0.058	0.020	1
132	-0.086	0.021	1
133	-0.101	0.013	1
134	-0.091	0.001	1
135	-0.069	-0.006	1
136	-0.060	-0.003	0
137	-0.025	0.004	0
138	-0.026	0.013	0
139	-0.032	0.020	0
140	-0.040	0.022	0
141	-0.049	0.014	0
142	-0.054	0.001	0
143	-0.053	-0.006	0
144	-0.049	-0.003	0
145	-0.026	0.002	0
146	-0.032	0.005	0
147	-0.040	0.007	0
148	-0.049	0.008	0
149	-0.054	0.006	0
150	-0.053	0.003	0

151	-0.049	0.000	0
152	-0.047	-0.000	0
153	-0.023	0.002	0
154	-0.023	0.005	0
155	-0.025	0.007	0
156	-0.027	0.008	0
157	-0.029	0.006	0
158	-0.031	0.003	0
159	-0.032	0.000	0
160	-0.032	-0.000	0
161	-0.023	0.000	0
162	-0.025	0.002	0
163	-0.027	0.002	0
164	-0.029	0.002	0
165	-0.031	0.002	0
166	-0.032	0.002	0
167	-0.032	0.001	0
168	-0.031	0.001	0
169	-0.017	0.000	0
170	-0.018	0.002	0
171	-0.019	0.002	0
172	-0.020	0.001	0
173	-0.017	0.000	0
174	-0.018	0.000	0
175	-0.019	0.000	0
176	-0.020	-0.000	0
177	-0.018	0.002	0
178	-0.019	0.002	0
179	-0.020	0.002	0
180	-0.020	0.001	0
181	-0.012	0.000	0
182	-0.012	0.000	0
183	-0.013	0.000	0
184	-0.013	-0.000	0
185	-0.012	0.000	0
186	-0.013	0.000	0
187	-0.013	0.000	0
188	-0.013	0.000	0
189	-0.009	0.000	0
190	-0.009	0.000	0
191	-0.009	0.000	0
192	-0.009	0.000	0
193	-0.009	0.000	0
194	-0.009	0.000	0
195	-0.009	0.000	0
196	-0.009	0.000	0
197	-0.007	0.000	0
198	-0.007	0.000	0
199	-0.007	0.000	0
200	-0.007	0.000	0
201	-0.007	0.000	0
202	-0.007	0.001	0
203	-0.007	0.000	0
204	-0.006	0.001	0
205	-0.006	0.000	0
206	-0.006	0.000	0
207	-0.006	0.0	0
208	-0.006	0.0	0
209	-0.006	0.001	0
210	-0.006	0.001	0

NODAL FLUXES AT WELL BOUNDARY

NODE	DISCHARGE FLUX
9	11.8963
10	9.2582
11	8.9767
12	9.2124
13	11.2508

TOTAL DISCHARGE INTO THE WELL # 50.5943

```
************************************************
*              STEADY STATE TYPE CURVE
************************************************
```

NON-LINEAR FACTOR # 0.0072

NODE NUMBER	R-COORDINATE	FUNCTION W(U)	ARGUMENT U
1	1.00	0.2655E 01	0.1000E-01
2	1.00	0.2672E 01	0.1000E-01
3	1.00	0.2726E 01	0.1000E-01
4	1.00	0.2826E 01	0.1000E-01
5	1.00	0.2987E 01	0.1000E-01
6	1.00	0.3250E 01	0.1000E-01
7	1.00	0.3730E 01	0.1000E-01
8	1.00	0.4866E 01	0.1000E-01
9	1.00	0.1109E 02	0.1000E-01
10	1.00	0.1109E 02	0.1000E-01
11	1.00	0.1109E 02	0.1000E-01
12	1.00	0.1109E 02	0.1000E-01
13	1.00	0.1109E 02	0.1000E-01
14	1.00	0.4958E 01	0.1000E-01
15	1.00	0.4045E 01	0.1000E-01

16	1.00	0.3723E 01	0.1000E-01
17	1.00	0.3633E 01	0.1000E-01
18	1.50	0.2652E 01	0.1500E-01
19	1.50	0.26[illegible]E 01	0.1500E-01
20	1.50	0.2723E 01	0.1500E-01
21	1.50	0.2822E 01	0.1500E-01
22	1.50	0.29[illegible]E 01	0.1500E-01
23	1.50	0.3237E 01	0.1500E-01
24	1.50	0.3692E 01	0.1500E-01
25	1.50	0.4641E 01	0.1500E-01
26	1.50	0.7394E 01	0.1500E-01
27	1.50	0.8071E 01	0.1500E-01
28	1.50	0.8225E 01	0.1500E-01
29	1.50	0.8100E 01	0.1500E-01
30	1.50	0.7425E 01	0.1500E-01
31	1.50	0.4805E 01	0.1500E-01
32	1.50	0.4012E 01	0.1500E-01
33	1.50	0.3706E 01	0.1500E-01
34	1.50	0.3620E 01	0.1500E-01
35	2.25	0.2640E 01	0.2250E-01
36	2.25	0.2656E 01	0.2250E-01
37	2.25	0.2708E 01	0.2250E-01
38	2.25	0.2602E 01	0.2250E-01
39	2.25	0.2950E 01	0.2250E-01
40	2.25	0.3131E 01	0.2250E-01
41	2.25	0.3561E 01	0.2250E-01
42	2.25	0.4205E 01	0.2250E-01
43	2.25	0.5350E 01	0.2250E-01
44	2.25	0.5895E 01	0.2250E-01
45	2.25	0.6073E 01	0.2250E-01
46	2.25	0.5740E 01	0.2250E-01
47	2.25	0.5423E 01	0.2250E-01
48	2.25	0.4416E 01	0.2250E-01
49	2.25	0.3685E 01	0.2250E-01
50	2.25	0.3639E 01	0.2250E-01
51	2.25	0.3567E 01	0.2250E-01
52	3.38	0.2609E 01	0.3375E-01
53	3.38	0.2624E 01	0.3375E-01
54	3.38	0.2670E 01	0.3375E-01
55	3.38	0.2752E 01	0.3375E-01
56	3.38	0.2876E 01	0.3375E-01
57	3.38	0.3054E 01	0.3375E-01
58	3.38	0.3321E 01	0.3375E-01
59	3.38	0.3668E 01	0.3375E-01
60	3.38	0.4142E 01	0.3375E-01
61	3.38	0.4407E 01	0.3375E-01
62	3.38	0.4550E 01	0.3375E-01
63	3.38	0.4460E 01	0.3375E-01
64	3.38	0.4253E 01	0.3375E-01
65	3.38	0.3886E 01	0.3375E-01
66	3.38	0.3637E 01	0.3375E-01
67	3.38	0.3462E 01	0.3375E-01
68	3.38	0.3437E 01	0.3375E-01
69	5.06	0.2535E 01	0.5062E-01
70	5.06	0.2579E 01	0.5062E-01
71	5.06	0.2727E 01	0.5062E-01
72	5.06	0.2988E 01	0.5062E-01
73	5.06	0.3327E 01	0.5062E-01
74	5.06	0.3521E 01	0.5062E-01
75	5.06	0.3449E 01	0.5062E-01
76	5.06	0.3253E 01	0.5062E-01
77	5.06	0.3163E 01	0.5062E-01
78	7.59	0.2391E 01	0.7594E-01
79	7.59	0.2417E 01	0.7594E-01
80	7.59	0.2498E 01	0.7594E-01
81	7.59	0.2621E 01	0.7594E-01
82	7.59	0.2756E 01	0.7594E-01
83	7.59	0.2841E 01	0.7594E-01
84	7.59	0.2847E 01	0.7594E-01
85	7.59	0.2810E 01	0.7594E-01
86	7.59	0.2789E 01	0.7594E-01
	11.39	0.2156E 01	0.1139E 00
88	11.39	0.2167E 01	0.1139E 00
89	11.39	0.2199E 01	0.1139E 00
90	11.39	0.2245E 01	0.1139E 00
91	11.39	0.2294E 01	0.1139E 00
92	11.39	0.2330E 01	0.1139E 00
93	11.39	0.2347E 01	0.1139E 00
94	11.39	0.2350E 01	0.1139E 00
95	11.39	0.2349E 01	0.1139E 00
96	17.09	0.1837E 01	0.1709E 00
97	17.09	0.1839E 01	0.1709E 00
98	17.09	0.1850E 01	0.1709E 00
99	17.09	0.1862E 01	0.1709E 00
100	17.09	0.1877E 01	0.1709E 00
101	17.09	0.1883E 01	0.1709E 00
102	17.09	0.1895E 01	0.1709E 00
103	17.09	0.1902E 01	0.1709E 00
104	17.09	0.1906E 01	0.1709E 00
105	25.63	0.1469E 01	0.2563E 00
106	25.63	0.1463E 01	0.2563E 00
107	25.63	0.1473E 01	0.2563E 00
108	25.63	0.1477E 01	0.2563E 00
109	25.63	0.1477E 01	0.2563E 00
110	38.44	0.1072E 01	0.3844E 00
111	38.44	0.1072E 01	0.3844E 00
112	38.44	0.1073E 01	0.3844E 00
113	38.44	0.1074E 01	0.3844E 00
114	38.44	0.1074E 01	0.3844E 00
115	51.40	0.7835E 00	0.5140E 00
116	51.40	0.7337E 00	0.5140E 00
117	51.40	0.7344E 00	0.5140E 00
118	51.40	0.7350E 00	0.5140E 00
119	51.40	0.7354E 00	0.5140E 00
120	70.84	0.4585E 00	0.7084E 00
121	70.84	0.4606E 00	0.7084E 00
122	70.84	0.4673E 00	0.7084E 00
123	70.84	0.4691E 00	0.7084E 00
124	70.84	0.4762E 00	0.7084E 00
125	100.00	0.0	0.1000E 01
126	100.00	0.0	0.1000E 01
127	100.00	0.0	0.1000E 01
128	100.00	0.0	0.1000E 01
129	100.00	0.0	0.1000E 01

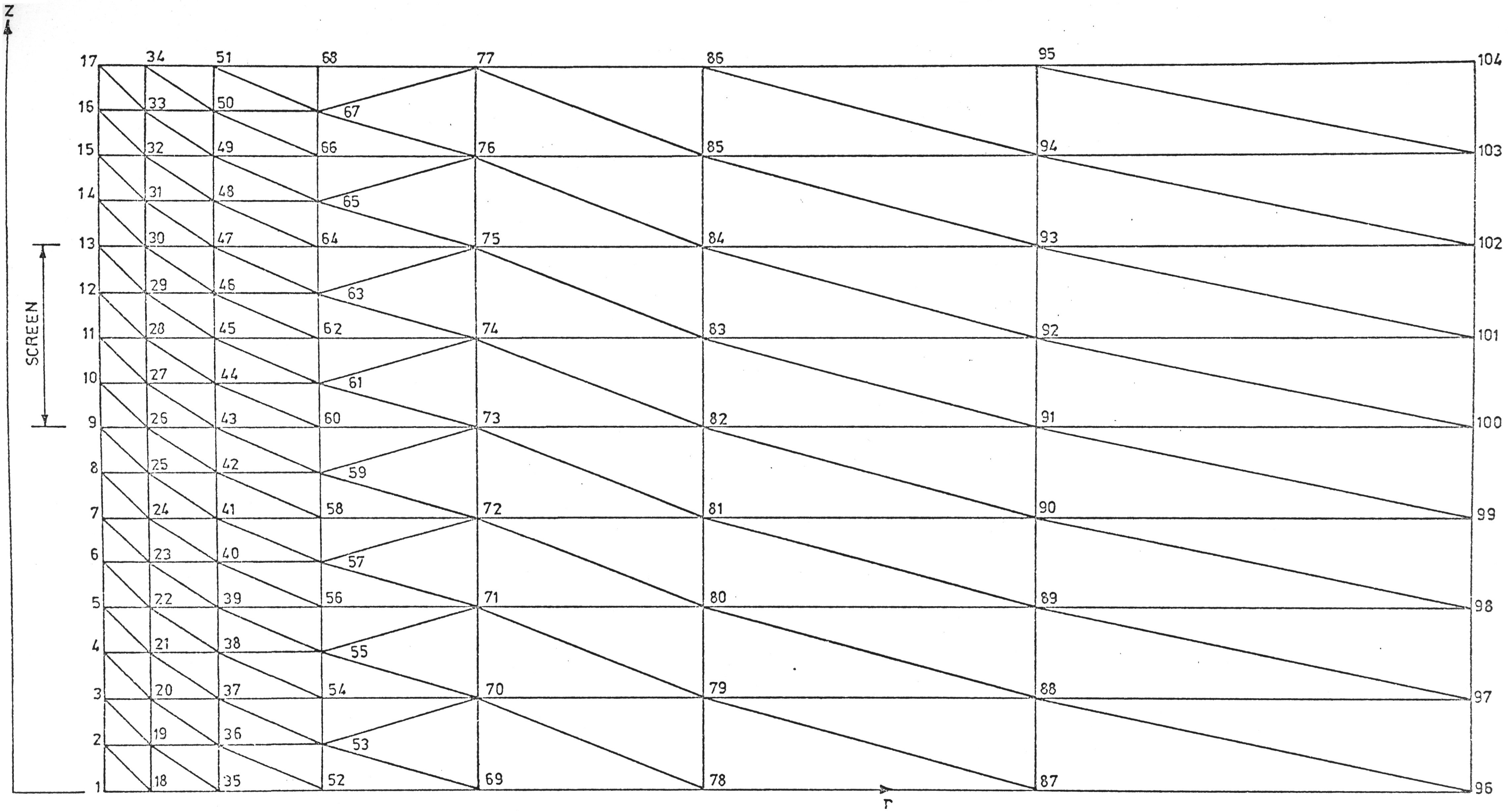


FIGURE 14: MESH GENERATED FOR SAMPLE PROBLEM

3.5 TRCON3 - PROGRAM FOR SOLVING TRANSIENT, TWO-DIMENSIONAL FLOW USING TRIANGULAR ELEMENTS

The program TRCON3 is used to analyse transient, two-dimensional flow in a single confined or unconfined aquifer or in a multi-layer system consisting of a number of aquifers and/or aquitards. Up to 3 layers of the multi-layer system can be handled in the current version of the program. Modification to handle a greater number of layers is given in Section 5. When the single aquifer or the top layer of the multi-layer system is unconfined, Boulton's concept of delayed yield and his assumption of constant saturated thickness are applied. Fully penetrating wells with partial or complete screening and with or without gravel packs are considered. Up to 5 screened intervals may be used.

Figures 15 and 16 show the input data form and completed form for a sample run. One problem was solved by the program, with IDISCR = 1 and IVEL = 0. The problem involves transient, two-regime flow towards a partially screened well in a confined aquifer. Numerical solutions were obtained for ten time steps. A portion of the printed output is shown on pages 63 to 68. It consists of the following:

(i) a reformatted playback of input data;

(ii) the generated discretisation data, namely node connections of elements, nodal co-ordinates, node numbers of top boundary nodes, the radial co-ordinates of such nodes and the length of the gross vector;

(iii) two tables of results for the time values at the mid-point and the end of time step number 10. Each table lists the node numbers, radial co-ordinates, nodal values of the hydraulic head (measured from the initial position of the water table), nodal values of $1/u = 4Kt/r^2 S_s$ and nodal values of $W(u) = 4\pi K m s/Q$.

A number of punched cards were also produced by the program. The information contained in these cards is nodal values of the hydraulic head and flux at the end of the final time step. As in TRCON1, the punched cards may be added to the input data deck to rerun the program from step no. 10 onwards.

From the printed output, a family of type curve , $W(u)$ Vs. $1/u$, may be obtained by plotting the nodal values at various times for selected nodal points in the mesh.

FIGURE 15: INPUT DATA FORM FOR TRCON3

COMPUTER DATA SHEET	PROGRAM TRCON3 Transient, two-dimensional, confined or unconfined flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	HTOL	NLAYR

(4F10.2, I10)

QFIX	RCSNG	QRTOL

(3F10.2)

IGP	IBOUND	IWBC	IKMAX	IWAT

(5I10)

THL	AKL	BKL	VCRL	SSL

(5E10.2)

(No. of cards = NLAYR)

AGP	BGP	VGP	THGP	BTGP

(5F10.3)

(Omit this card if IGP = 0)

SY	DINDEX

(2E10.2)

NTICR	ITST	TFACTR	TMUL	DTMUL

(2I10, 3F10.2)

FRLEN	SCFAC	XLMAX

(3F10.2)

NDSC	NSCREN

(2I10)

IREGL	NMINL	NFRL

(3I10)

XSCR	HSCR

(2F10.2)

FIGURE 16: COMPLETED FORM FOR SAMPLE PROBLEM

COMPUTER DATA SHEET	PROGRAM TRCON3 Transient, two-dimensional, confined or unconfined flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
1	0	1	1.30

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	HTOL	NLAYR
1.00	1000.00	0.00	0.50	1

(4F10.2, I10)

QFIX	RCSNG	QRTOL
100.00	1.00	0.01

(3F10.2)

IGP	IBOUND	IWBC	IKMAX	IWAT
0	1	0	1	0

(5I10)

THL	AKL	BKL	VCRL	SSL
0.20E02	0.10E02	0.20E02	0.60E-01	0.10E-04

(5E10.2)

(No. of cards = NLAYR)

AGP	BGP	VGP	THGP	BTGP

(5F10.3)

(Omit this card if IGP = 0)

SY	DINDEX

(2E10.2)

NTICR	ITST	TFACTR	TMUL	DTMUL
10	1	50.00	2.00	0.00

(2I10, 3F10.2)

FRLEN	SCFAC	XLMAX
0.50	1.50	100.00

(3F10.2)

NDSC	NSCREN
3	1

(2I10)

IREGL	NMINL	NFRL
4	3	5

(3I10)

XSCR	HSCR
10.00	10.00

(2F10.2)

```
*************************************************
*    FINITE ELEMENT ANALYSIS OF WELL PROBLEMS    *
*                                               *
* TRANSIENT,TWO REGIME FLOW TOWARD A SINGLE WELL *
*                                               *
*************************************************
```

```
**************************************************
*            PROBLEM NUMBER   #      1           *
**************************************************
```

GENERAL DATA

RADIUS OF WELL # 1.00

RADIUS OF INFLUENCE # 1000.00

HEIGHT OF WATER TABLE # 0.0

RADIUS OF WELL CASING # 1.00

DISCHARGE INTO WELL # 100.00

DISCHARGE TOLERANCE RATIO # 0.0100

OVER RELAXATION FACTOR # 1.3000

HEAD TOLERANCE # 0.5000

GRAVEL PACK INDEX # 0

BOUNDARY INDEX # 1

WELL B.C. INDEX # 0

LAYER OF MAX PERMEABILITY # 1

WATER TABLE AQUIFER INDEX # 0

FORMATION PROPERTIES

LAYER NO.	THICKNESS	COEFF-A	COEFF-B	CRIT. VELOCITY	SPECIFIC STORAGE
1	20.00	0.10E 02	0.20E 02	0.60E-01	0.10E-04

SCREEN NO.	BASE HEIGHT	LENGTH
1	10.00	10.00

DISCRETISATION DATA

NUMBER OF LAYERS # 1

NUMBER OF REGIONS # 22

IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS

ELEM NO	NODE1	NODE2	NODE3	REPETITION NUMBER
1	1	6	2	1
2	2	7	3	1
3	3	8	4	1
4	4	9	5	1
5	2	6	7	2
6	3	7	8	2
7	4	8	9	2
8	5	9	10	2
9	6	11	7	3
10	7	12	8	3
11	8	13	9	3
12	9	14	10	3
13	7	11	12	4
14	8	12	13	4
15	9	13	14	4
16	10	14	15	4
17	11	16	12	5
18	12	17	13	5
19	13	18	14	5
20	14	19	15	5

21	12	16	17	6
22	13	17	18	6
23	14	18	19	6
24	15	19	20	6
25	16	21	17	7
26	18	22	19	7
27	17	21	22	8
28	19	22	23	8
29	17	22	18	9
30	19	23	20	9
31	21	24	22	10
32	22	25	23	10
33	22	24	25	11
34	23	25	26	11
35	24	27	25	12
36	25	28	26	12
37	25	27	28	13
38	26	28	29	13
39	27	30	28	14
40	28	31	29	14
41	28	30	31	15
42	29	31	32	15
43	30	33	31	16
44	31	34	32	16
45	31	33	34	17
46	32	34	35	17
47	33	36	34	18
48	34	37	35	18
49	34	36	37	19
50	35	37	38	19
51	36	39	37	20
52	37	40	38	20
53	37	39	40	21
54	38	40	41	21
55	39	42	40	22
56	40	43	41	22
57	40	42	43	23
58	41	43	44	23
59	42	45	43	24
60	43	46	44	24
61	43	45	46	25
62	44	46	47	25
63	45	48	46	26
64	46	49	47	26
65	46	48	49	27
66	47	49	50	27
67	48	51	49	28
68	49	52	50	28
69	49	51	52	29
70	50	52	53	29
71	51	54	52	30
72	52	55	53	30
73	52	54	55	31
74	53	55	56	31
75	54	57	55	32
76	55	58	56	32
77	55	57	58	33
78	56	58	59	33
79	57	60	58	34
80	58	61	59	34
81	58	60	61	35
82	59	61	62	35
83	60	63	61	36
84	61	64	62	36
85	61	63	64	37
86	62	64	65	37
87	63	66	64	38
88	64	67	65	38
89	64	66	67	39
90	65	67	68	39
91	66	69	67	40
92	67	70	68	40
93	67	69	70	41
94	68	70	71	41
95	69	72	70	42
96	70	73	71	42
97	70	72	73	43
98	71	73	74	43
99	72	75	73	44
100	73	76	74	44
101	73	75	76	45
102	74	76	77	45

NODAL COORDINATES

NODE	R-COORD	Z-COORD
1	1.00	0.0
2	1.00	5.00
3	1.00	10.00
4	1.00	15.00
5	1.00	20.00
6	1.50	0.0
7	1.50	5.00
8	1.50	10.00
9	1.50	15.00
10	1.50	20.00
11	2.25	0.0
12	2.25	5.00
13	2.25	10.00
14	2.25	15.00
15	2.25	20.00
16	3.38	0.0
17	3.38	5.00
18	3.38	10.00
19	3.38	15.00
20	3.38	20.00

21	5.06	0.0
22	5.06	10.00
23	5.06	20.00
24	7.59	0.0
25	7.59	10.00
26	7.59	20.00
27	11.39	0.0
28	11.39	10.00
29	11.39	20.00
30	17.09	0.0
31	17.09	10.00
32	17.09	20.00
33	25.63	0.0
34	25.63	10.00
35	25.63	20.00
36	38.44	0.0
37	38.44	10.00
38	38.44	20.00
39	57.67	0.0
40	57.67	10.00
41	57.67	20.00
42	86.50	0.0
43	86.50	10.00
44	86.50	20.00
45	129.75	0.0
46	129.75	10.00
47	129.75	20.00
48	194.62	0.0
49	194.62	10.00
50	194.62	20.00
51	291.93	0.0
52	291.93	10.00
53	291.93	20.00
54	391.93	0.0
55	391.93	10.00
56	391.93	20.00
57	491.93	0.0
58	491.93	10.00
59	491.93	20.00
60	591.93	0.0
61	591.93	10.00
62	591.93	20.00
63	691.93	0.0
64	691.93	10.00
65	691.93	20.00
66	791.93	0.0
67	791.93	10.00
68	791.93	20.00
69	835.73	0.0
70	835.73	10.00
71	835.73	20.00
72	901.44	0.0
73	901.44	10.00
74	901.44	20.00
75	1000.00	0.0
76	1000.00	10.00
77	1000.00	20.00

TOP BOUNDARY NODES AND RADIAL COORDINATES

NODE NUMBER	R-COORDINATE
5	1.00
10	1.50
15	2.25
20	3.38
23	5.06
26	7.59
29	11.39
32	17.09
35	25.63
38	38.44
41	57.67
44	86.50
47	129.75
50	194.62
53	291.93
56	391.93
59	491.93
62	591.93
65	691.93
68	791.93
71	835.73
74	901.44
77	1000.00

LENGTH OF GROSS VECTOR # 338

```
*********************************
*    TIME STEP NUMBER  #   10   *
*********************************
```

```
*****************************************
*        TIME  #     0.481E 01          *
*****************************************
```

ESTIMATED RADIUS OF INFLUENCE # 1000.00

CORRESPONDING NO. OF NODES # 77

CORRESPONDING NO. OF ELEMENTS # 102

CORRESPONDING COMPONENT OF VECTOR NDVEC # 23

\- -

DISCHARGE ITERATION NUMBER # 1

DRAWDOWN INCREMENT # -1.793

DRAWDOWN VALUE # -80.770

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.0009

DISCHARGE FROM AQUIFER INTO WELL # 100.0951

DISCHARGE FROM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 100.095

RESIDUAL DISCHARGE # 0.0951

```
*******************************************
*                                         *
*       FINAL RESULTS OF ANALYSIS         *
*                                         *
*******************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	HEAD	1/U	W(U)
1	1.00	-32.4561	0.1716E 06	7.2861
2	1.00	-35.9894	0.1716E 06	8.0793
3	1.00	-80.7703	0.1716E 06	18.1321
4	1.00	-80.7703	0.1716E 06	18.1321
5	1.00	-80.7703	0.1716E 06	18.1321
6	1.50	-32.4208	0.7629E 05	7.2781
7	1.50	-35.8532	0.7629E 05	8.0498
8	1.50	-64.5709	0.7629E 05	14.4955
9	1.50	-66.2339	0.7629E 05	14.8683
10	1.50	-67.2370	0.7629E 05	15.0940
11	2.25	-32.2689	0.3391E 05	7.2440
12	2.25	-35.4906	0.3391E 05	7.9673
13	2.25	-52.8173	0.3391E 05	11.8569
14	2.25	-54.9210	0.3391E 05	12.3292
15	2.25	-56.6892	0.3391E 05	12.7261
16	3.38	-31.8716	0.1507E 05	7.1548
17	3.38	-34.8019	0.1507E 05	7.8127
18	3.38	-44.1988	0.1507E 05	9.9222
19	3.38	-45.7593	0.1507E 05	10.2726
20	3.38	-48.2462	0.1507E 05	10.8308
21	5.06	-30.9693	0.6698E 04	6.9523
22	5.06	-36.9654	0.6698E 04	8.2983
23	5.06	-40.1956	0.6698E 04	9.0235
24	7.59	-29.4525	0.2977E 04	6.6118
25	7.59	-32.5598	0.2977E 04	7.3093
26	7.59	-34.4041	0.2977E 04	7.7234
27	11.39	-27.3260	0.1323E 04	6.1344
28	11.39	-28.6373	0.1323E 04	6.4288
29	11.39	-29.5165	0.1323E 04	6.6266
30	17.09	-24.5712	0.5680E 03	5.5160

31	17.09	-25.0078	0.5880E 03	5.6140
32	17.09	-25.3400	0.5880E 03	5.6886
33	25.63	-21.3417	0.2613E 03	4.7910
34	25.63	-21.4358	0.2613E 03	4.8128
35	25.63	-21.5243	0.2613E 03	4.8320
36	38.44	-17.8951	0.1161E 03	4.0150
37	38.44	-17.8976	0.1161E 03	4.0178
38	38.44	-17.9003	0.1161E 03	4.0206
39	57.67	-14.3888	0.5162E 02	3.2301
40	57.67	-14.3892	0.5162E 02	3.2302
41	57.67	-14.3897	0.5162E 02	3.2303
42	86.50	-10.9523	0.2294E 02	2.4587
43	86.50	-10.9519	0.2294E 02	2.4586
44	86.50	-10.9515	0.2294E 02	2.4585
45	129.75	-7.6686	0.1020E 02	1.7215
46	129.75	-7.6683	0.1020E 02	1.7214
47	129.75	-7.6679	0.1020E 02	1.7214
48	194.62	-4.7008	0.4532E 01	1.0553
49	194.62	-4.7006	0.4532E 01	1.0552
50	194.62	-4.7004	0.4532E 01	1.0552
51	291.93	-2.3176	0.2014E 01	0.5203
52	291.93	-2.3179	0.2014E 01	0.5203
53	291.93	-2.3182	0.2014E 01	0.5204
54	391.93	-1.1024	0.1117E 01	0.2475
55	391.93	-1.1026	0.1117E 01	0.2475
56	391.93	-1.1028	0.1117E 01	0.2476
57	491.93	-0.5099	0.7093E 00	0.1145
58	491.93	-0.5100	0.7093E 00	0.1145
59	491.93	-0.5101	0.7093E 00	0.1145
60	591.93	-0.2293	0.4899E 00	0.0515
61	591.93	-0.2293	0.4899E 00	0.0515
62	591.93	-0.2294	0.4899E 00	0.0515
63	691.93	-0.1004	0.3585E 00	0.0225
64	691.93	-0.1004	0.3585E 00	0.0225
65	691.93	-0.1005	0.3585E 00	0.0226
66	791.93	-0.0424	0.2737E 00	0.0095
67	791.93	-0.0425	0.2737E 00	0.0095
68	791.93	-0.0425	0.2737E 00	0.0095
69	835.73	-0.0283	0.2458E 00	0.0063
70	835.73	-0.0283	0.2458E 00	0.0063
71	835.73	-0.0283	0.2458E 00	0.0063
72	901.44	-0.0138	0.2112E 00	0.0031
73	901.44	-0.0138	0.2112E 00	0.0031
74	901.44	-0.0138	0.2112E 00	0.0031
75	1000.00	0.0	0.1716E 00	0.0
76	1000.00	0.0	0.1716E 00	0.0
77	1000.00	0.0	0.1716E 00	0.0

```
****************************************
*          TIME  #    0.641E 01         *
****************************************

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER  #          1
DRAWDOWN INCREMENT   #    -1.793
DRAWDOWN VALUE   #   -82.563

NUMBER OF ITERATIONS REQUIRED  #   2

TOLERANCE COUNTER FOR HEAD   #  0
ABSOLUTE MAXIMUM ERROR IN HEAD   #       0.0349

DISCHARGE FROM AQUIFER INTO WELL   #    100.5935
DISCHARGE FROM WELL STORAGE #        0.0

TOTAL CALCULATED DISCHARGE   #      100.593
RESIDUAL DISCHARGE          #   0.5935
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER  #          2
DRAWDOWN INCREMENT   #    -1.306
DRAWDOWN VALUE   #   -82.076

NUMBER OF ITERATIONS REQUIRED  #   2

TOLERANCE COUNTER FOR HEAD   #  0
ABSOLUTE MAXIMUM ERROR IN HEAD   #       0.0013
```

DISCHARGE FROM AQUIFER INTO WELL # 100.0796

DISCHARGE FROM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 100.080

RESIDUAL DISCHARGE # 0.0796

```
*********************************************
*                                           *
*       FINAL RESULTS OF ANALYSIS           *
*                                           *
*********************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	HEAD	1/U	W(U)
1	1.00	-33.7709	0.2289E 06	7.5812
2	1.00	-37.3038	0.2289E 06	8.3743
3	1.00	-82.0758	0.2289E 06	18.4252
4	1.00	-82.0758	0.2289E 06	18.4252
5	1.00	-82.0758	0.2289E 06	18.4252
6	1.50	-33.7356	0.1017E 06	7.5733
7	1.50	-37.1726	0.1017E 06	8.3449
8	1.50	-65.8798	0.1017E 06	14.7893
9	1.50	-67.5425	0.1017E 06	15.1626
10	1.50	-68.5454	0.1017E 06	15.3877
11	2.25	-33.5838	0.4521E 05	7.5392
12	2.25	-36.8050	0.4521E 05	8.2623
13	2.25	-54.1285	0.4521E 05	12.1513
14	2.25	-56.2318	0.4521E 05	12.6235
15	2.25	-57.9998	0.4521E 05	13.0203
16	3.38	-33.1866	0.2009E 05	7.4500
17	3.38	-36.1165	0.2009E 05	8.1078
18	3.38	-45.5117	0.2009E 05	10.2169
19	3.38	-47.0724	0.2009E 05	10.5673
20	3.38	-49.5584	0.2009E 05	11.1253
21	5.06	-32.2842	0.8930E 04	7.2475
22	5.06	-38.2795	0.8930E 04	8.5934
23	5.06	-41.5092	0.8930E 04	9.3184
24	7.59	-30.7674	0.3969E 04	6.9070
25	7.59	-33.8743	0.3969E 04	7.6044
26	7.59	-35.7184	0.3969E 04	8.0184
27	11.39	-28.6405	0.1764E 04	6.4295
28	11.39	-29.9518	0.1764E 04	6.7239
29	11.39	-30.8330	0.1764E 04	6.9217
30	17.09	-25.8847	0.7840E 03	5.8108
31	17.09	-26.3213	0.7840E 03	5.9089
32	17.09	-26.6536	0.7840E 03	5.9835
33	25.63	-22.6526	0.3484E 03	5.0853
34	25.63	-22.7498	0.3484E 03	5.1071
35	25.63	-22.8355	0.3484E 03	5.1263
36	38.44	-19.1897	0.1549E 03	4.3079
37	38.44	-19.2023	0.1549E 03	4.3107
38	38.44	-19.2147	0.1549E 03	4.3135
39	57.67	-15.6789	0.6883E 02	3.5198
40	57.67	-15.6795	0.6883E 02	3.5199
41	57.67	-15.6800	0.6883E 02	3.5200
42	86.50	-12.2099	0.3059E 02	2.7410
43	86.50	-12.2096	0.3059E 02	2.7409
44	86.50	-12.2093	0.3059E 02	2.7409
45	129.75	-8.8564	0.1360E 02	1.9882
46	129.75	-8.8561	0.1360E 02	1.9881
47	129.75	-8.8558	0.1360E 02	1.9880
48	194.62	-5.7479	0.6042E 01	1.2903
49	194.62	-5.7476	0.6042E 01	1.2903
50	194.62	-5.7474	0.6042E 01	1.2902
51	291.93	-3.1142	0.2686E 01	0.6991
52	291.93	-3.1145	0.2686E 01	0.6992
53	291.93	-3.1148	0.2686E 01	0.6992
54	391.93	-1.6513	0.1490E 01	0.3707
55	391.93	-1.6515	0.1490E 01	0.3707
56	391.93	-1.6517	0.1490E 01	0.3708
57	491.93	-0.8585	0.9458E 00	0.1927
58	491.93	-0.8586	0.9458E 00	0.1927
59	491.93	-0.8587	0.9458E 00	0.1928
60	591.93	-0.4355	0.6532E 00	0.0978
61	591.93	-0.4356	0.6532E 00	0.0978
62	591.93	-0.4357	0.6532E 00	0.0978
63	691.93	-0.2147	0.4780E 00	0.0482
64	691.93	-0.2147	0.4780E 00	0.0482
65	691.93	-0.2148	0.4780E 00	0.0482
66	791.93	-0.1009	0.3649E 00	0.0226
67	791.93	-0.1009	0.3649E 00	0.0227
68	791.93	-0.1010	0.3649E 00	0.0227
69	835.73	-0.0698	0.3277E 00	0.0157
70	835.73	-0.0699	0.3277E 00	0.0157
71	835.73	-0.0698	0.3277E 00	0.0157
72	901.44	-0.0358	0.2816E 00	0.0080
73	901.44	-0.0358	0.2816E 00	0.0080
74	901.44	-0.0358	0.2816E 00	0.0080
75	1000.00	0.0	0.2289E 00	0.0
76	1000.00	0.0	0.2289E 00	0.0
77	1000.00	0.0	0.2289E 00	0.0

TOTAL NUMBER OF PUNCHED CARDS # 14

3.6 STCOND - PROGRAM FOR SOLVING STEADY, TWO-DIMENSIONAL FLOW USING RECTANGULAR AND TRIANGULAR ELEMENTS

The program STCOND is used to analyse steady, two-dimensional flow in a confined aquifer or in a multi-layer system. Unlike STCON3, a combination of rectangular and triangular elements is employed, and only fully penetrating wells with or without gravel packing are considered. (i.e. for a partially penetrating well, this program neglects the portion of the formation which is directly below the base of the well.) For the multi-layer system, the maximum number of formations that can be handled in the current version of the program is 3, and up to 5 screened intervals may be used.

Figures 17 and 18 show the input data form and completed form for a sample run. One problem was solved, with IVEL = 1 and IDISCR = 1. The problem is almost identical to that solved by STCON3 except for the length of the well screen, which in this case is 10 ft. The printed output is shown on pages 72 to 75. The content of this output is identical to that given by STCON3. Figure 19 shows the plot of the discretisation data. The node numbers and well screen are also depicted in the figure.

FIGURE 17: INPUT DATA FORM FOR STCOND

COMPUTER DATA SHEET	PROGRAM STCOND Steady, two-dimensional flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	HTOL	NLAYR

(4F10.2, I10)

IGP	IKMAX	IAQTA

(3I10)

THL	AKL	BKL	VCRL

(No. of cards = NLAYR)

(4F10.3)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

PA	THA

(2F10.3)

(Omit this card if IAQTA = 0)

HW

(F10.2)

FRLEN	SCFAC	XLMAX

(3F10.2)

NDSC	NSCREN

(2I10)

IREGL	NMINL	NFRL

(No. of cards = NLAYR)

(3I10)

XSCR	HSCR

(2F10.2)

(No. of cards = NSCREN)

FIGURE 18: COMPLETED FORM FOR SAMPLE PROBLEMS

COMPUTER DATA SHEET	PROGRAM STCOND Steady, two-dimensional flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
1	1	1	1.50

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	HTOL	NLAYR
1.00	100.00	0.00	0.20	1

(4F10.2, I10)

IGP	IKMAX	IAQTA
0	1	0

(3I10)

THL	AKL	BKL	VCRL
0.20E02	0.10E02	0.20E02	0.50E-01

(No. of cards = NLAYR)

(4F10.3)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

PA	THA

(2F10.3)

(Omit this card if IAQTA = 0)

HW
-50.00

(F10.2)

FRLEN	SCFAC	XLMAX
0.50	1.50	50.00

(3F10.2)

NDSC	NSCREN
3	1

(2I10)

IREGL	NMINL	NFRL
4	2	5

(3I10)

(No. of cards = NLAYR)

XSCR	HSCR
10.00	10.00

(2F10.2)

(No. of cards = NSCREN)

```
*********************************************
*                                           *
*   FINITE ELEMENT ANALYSIS OF WELL PROBLEMS    *
*                                           *
*   STEADY,TWO REGIME FLOW TOWARD A SINGLE WELL *
*                                           *
*********************************************
```

```
*********************************************
*          PROBLEM NUMBER   #    1          *
*********************************************
```

GENERAL INPUT DATA

RADIUS OF WELL # 1.00

RADIUS OF INFLUENCE # 100.00

HEIGHT OF WATER TABLE # 0.0

OVER-RELAXATION FACTOR # 1.500

PRESCRIBED HEAD TOLERANCE # 0.200

GRAVEL PACK INDEX # 0

AQUITARD INDEX # 0

LAYER OF MAX PERMEABILITY # 1

FORMATION PROPERTIES

LAYER NO.	THICKNESS	COEFF.-A	COEFF.-B	CRIT. VELOCITY
1	0.20E 02	0.10E 02	0.20E 02	0.50E-01

SCREEN NO.	BASE HEIGHT	LENGTH
1	10.00	10.00

DISCRETISATION DATA

NUMBER OF LAYERS # 1

NUMBER OF SUBREGIONS # 12

IDENTIFICATION OF ELEMENT-NODE CONNECTIONS

ELEM NO	ITYPE	NREP	IPROP	NODE1	NODE2	NODE3	NODE4
1	4	1	1	1	6	7	2
2	4	1	1	2	7	8	3
3	4	1	1	3	8	9	4
4	4	1	1	4	9	10	5
5	4	2	1	6	11	12	7
6	4	2	1	7	12	13	8
7	4	2	1	8	13	14	9
8	4	2	1	9	14	15	10
9	4	3	1	11	16	17	12
10	4	3	1	12	17	18	13
11	4	3	1	13	18	19	14
12	4	3	1	14	19	20	15
13	3	4	1	16	21	17	
14	3	4	1	18	22	19	
15	3	5	1	17	21	22	
16	3	5	1	19	22	23	
17	3	6	1	17	22	13	
18	3	6	1	19	23	20	
19	4	7	1	21	24	25	22
20	4	7	1	22	25	26	23
21	4	8	1	24	27	28	25
22	4	8	1	25	28	29	26
23	4	9	1	27	30	31	28
24	4	9	1	28	31	32	29
25	3	10	1	30	33	31	
26	3	11	1	31	33	34	
27	3	12	1	31	34	32	
28	4	13	1	33	35	36	34
29	4	14	1	35	37	38	36
30	4	15	1	37	39	40	38
31	4	16	1	39	41	42	40

NODAL COORDINATES

NODE	R-COORD	Z-COORD
1	1.00	0.0
2	1.00	5.00
3	1.00	10.00
4	1.00	15.00
5	1.00	20.00
6	1.50	0.0
7	1.50	5.00
8	1.50	10.00
9	1.50	15.00
10	1.50	20.00
11	2.25	0.0
12	2.25	5.00
13	2.25	10.00
14	2.25	15.00
15	2.25	20.00
16	3.38	0.0
17	3.38	5.00
18	3.38	10.00
19	3.38	15.00
20	3.38	20.00
21	5.06	0.0
22	5.06	10.00
23	5.06	20.00
24	7.59	0.0
25	7.59	10.00
26	7.59	20.00
27	11.39	0.0
28	11.39	10.00
29	11.39	20.00
30	17.09	0.0
31	17.09	10.00
32	17.09	20.00
33	25.63	0.0
34	25.63	20.00
35	38.44	0.0
36	38.44	20.00
37	51.40	0.0
38	51.40	20.00
39	70.84	0.0
40	70.84	20.00
41	100.00	0.0
42	100.00	20.00

TOP BOUNDARY NODES AND RADIAL COORDINATES

NODE NUMBER	R-COORDINATE
5	1.00
10	1.50
15	2.25
20	3.38
23	5.06
26	7.59
29	11.39
32	17.09
34	25.63
36	38.44
38	51.40
40	70.84
42	100.00

PRESCRIBED NODE DATA

NODE	PRESCRIBED HEAD VALUE
3	-50.000
4	-50.000
5	-50.000
41	0.0
42	0.0

LENGTH OF GROSS VECTOR # 212

```
**************************************
*  PRESCRIBED WELL DRAWDOWN  # -50.00*
**************************************
```

NO. OF ITERATIONS REQUIRED # 5

MAXIMUM ERROR IN HEAD # 0.194

FINAL SOLUTION

NODE	R-COORD	Z-COORD	HEAD	RHO-COORD	TZI-COORD
1	1.00	0.0	-16.9599	0.0250	0.0
2	1.00	5.00	-13.8578	0.0250	0.2500
3	1.00	10.00	-50.0000	0.0250	0.5000
4	1.00	15.00	-50.0000	0.0250	0.7500
5	1.00	20.00	-50.0000	0.0250	1.0000
6	1.50	0.0	-15.3954	0.0375	0.0
7	1.50	5.00	-16.0473	0.0375	0.2500
8	1.50	10.00	-38.9957	0.0375	0.5000
9	1.50	15.00	-40.5912	0.0375	0.7500
10	1.50	20.00	-41.1598	0.0375	1.0000
11	2.25	0.0	-15.2777	0.0562	0.0
12	2.25	5.00	-17.3115	0.0562	0.2500
13	2.25	10.00	-30.5633	0.0562	0.5000
14	2.25	15.00	-33.0127	0.0562	0.7500
15	2.25	20.00	-34.1420	0.0562	1.0000
16	3.38	0.0	-14.8122	0.0844	0.0
17	3.38	5.00	-17.8132	0.0844	0.2500
18	3.38	10.00	-24.3942	0.0844	0.5000
19	3.38	15.00	-26.5669	0.0844	0.7500
20	3.38	20.00	-28.4394	0.0844	1.0000
21	5.06	0.0	-14.1655	0.1266	0.0
22	5.06	10.00	-19.9603	0.1266	0.5000
23	5.06	20.00	-23.2106	0.1266	1.0000
24	7.59	0.0	-14.0262	0.1898	0.0
25	7.59	10.00	-16.6714	0.1898	0.5000
26	7.59	20.00	-18.5375	0.1898	1.0000
27	11.39	0.0	-12.9192	0.2848	0.0
28	11.39	10.00	-13.8542	0.2848	0.5000
29	11.39	20.00	-14.6650	0.2848	1.0000
30	17.09	0.0	-11.0862	0.4271	0.0
31	17.09	10.00	-11.1957	0.4271	0.5000
32	17.09	20.00	-11.5203	0.4271	1.0000
33	25.63	0.0	-8.6139	0.6407	0.0
34	25.63	20.00	-8.7306	0.6407	1.0000
35	38.44	0.0	-6.0970	0.9611	0.0
36	38.44	20.00	-6.1003	0.9611	1.0000
37	51.40	0.0	-4.2425	1.2851	0.0
38	51.40	20.00	-4.2426	1.2851	1.0000
39	70.84	0.0	-2.1963	1.7710	0.0
40	70.84	20.00	-2.1963	1.7710	1.0000
41	100.00	0.0	0.0	2.5000	0.0
42	100.00	20.00	0.0	2.5000	1.0000

```
**************************************
*        ELEMENT VELOCITIES          *
**************************************
```

ELEM NO.	RADIAL VEL	VERTIC VEL	IDARCY
1	0.094	-0.025	1
2	-0.436	0.288	1
3	-0.793	0.007	1
4	-0.737	0.002	1
5	0.039	0.020	0
6	-0.277	0.213	1
7	-0.520	0.020	1
8	-0.491	0.009	1
9	0.002	0.046	0
10	-0.174	0.137	1
11	-0.335	0.028	1
12	-0.326	0.018	1
13	-0.034	0.053	1
14	-0.189	0.031	1
15	-0.039	0.051	1
16	-0.208	0.023	1
17	-0.185	0.093	1
18	-0.216	0.026	1
19	-0.060	0.037	1
20	-0.126	0.020	1
21	-0.047	0.016	0
22	-0.076	0.012	1
23	-0.036	0.005	0
24	-0.046	0.005	0
25	-0.026	0.001	0
26	-0.027	0.001	0
27	-0.030	0.003	0
28	-0.018	0.000	0
29	-0.013	0.000	0
30	-0.010	0.000	0
31	-0.007	-0.000	0

NODAL FLUXES AT WELL BOUNDARY

NODE	DISCHARGE FLUX
3	29.4061
4	29.7683
5	14.3252

TOTAL DISCHARGE INTO THE WELL # 73.4996

```
************************************************
*           STEADY STATE TYPE CURVE            *
************************************************
```

NON-LINEAR FACTOR # 0.0106

NODE NUMBER	R-COORDINATE	FUNCTION W(U)	ARGUMENT U
1	1.00	0.2637E 01	0.1000E-01
2	1.00	0.2155E 01	0.1000E-01
3	1.00	0.7775E 01	0.1000E-01
4	1.00	0.7775E 01	0.1000E-01
5	1.00	0.7775E 01	0.1000E-01
6	1.50	0.2472E 01	0.1500E-01
7	1.50	0.2495E 01	0.1500E-01
8	1.50	0.6048E 01	0.1500E-01
9	1.50	0.6312E 01	0.1500E-01
10	1.50	0.6400E 01	0.1500E-01
11	2.25	0.2376E 01	0.2250E-01
12	2.25	0.2692E 01	0.2250E-01
13	2.25	0.4752E 01	0.2250E-01
14	2.25	0.5133E 01	0.2250E-01
15	2.25	0.5309E 01	0.2250E-01
16	3.38	0.2303E 01	0.3375E-01
17	3.38	0.2771E 01	0.3375E-01
18	3.38	0.3793E 01	0.3375E-01
19	3.38	0.4131E 01	0.3375E-01
20	3.38	0.4422E 01	0.3375E-01
21	5.06	0.2203E 01	0.5062E-01
22	5.06	0.3105E 01	0.5062E-01
23	5.06	0.3609E 01	0.5062E-01
24	7.59	0.2181E 01	0.7594E-01
25	7.59	0.2592E 01	0.7594E-01
26	7.59	0.2382E 01	0.7594E-01
27	11.39	0.2099E 01	0.1139E 00
28	11.39	0.2154E 01	0.1139E 00
29	11.39	0.2280E 01	0.1139E 00
30	17.09	0.1724E 01	0.1709E 00
31	17.09	0.1741E 01	0.1709E 00
32	17.09	0.1791E 01	0.1709E 00
33	25.63	0.1339E 01	0.2563E 00
34	25.63	0.1358E 01	0.2563E 00
35	38.44	0.9480E 00	0.3844E 00
36	38.44	0.9485E 00	0.3844E 00
37	51.40	0.6597E 00	0.5140E 00
38	51.40	0.6597E 00	0.5140E 00
39	70.84	0.3415E 00	0.7084E 00
40	70.84	0.3415E 00	0.7084E 00
41	100.00	0.0	0.1000E 01
42	100.00	0.0	0.1000E 01

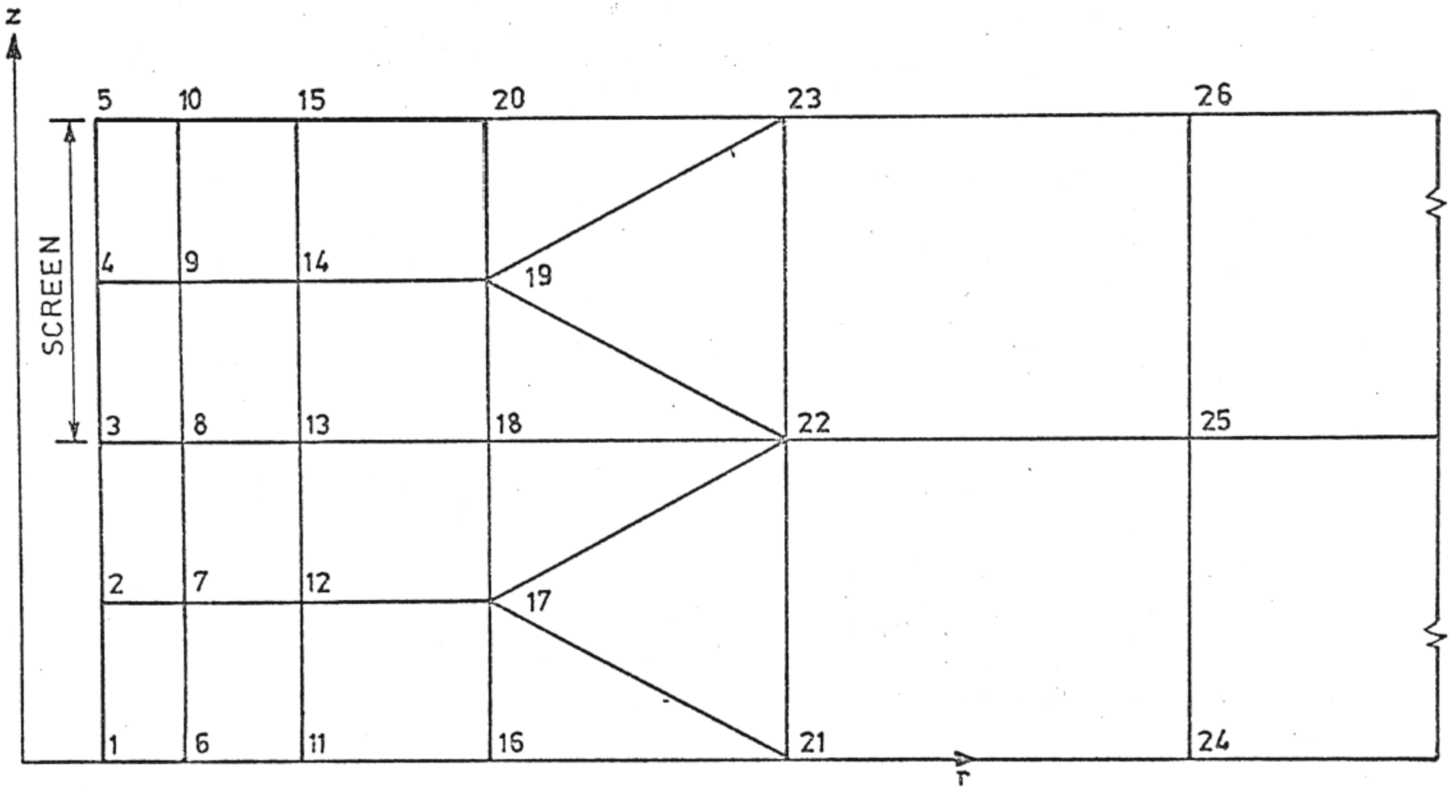


FIGURE 19: MESH GENERATED FOR SAMPLE PROBLEM

3.7 TRCOND - PROGRAM FOR SOLVING TRANSIENT, TWO-DIMENSIONAL FLOW USING RECTANGULAR AND TRIANGULAR ELEMENTS

The program TRCOND is used to analyse transient, two-dimensional flow in a single confined or unconfined aquifer or in a multi-layer system consisting of a number of aquifers and/or aquitards. It has the same capability as TRCON3 except that a combination of rectangular and triangular elements is used instead of using triangular elements alone.

Figures 20 and 21 show the input data form and completed form for a sample run. One problem was solved, with IVEL = 0 and IDISCR = 1. The problem solved was identical to that solved by TRCON3. Input data used was also identical. The printed output is shown on pages 80 to 88. The content of this output is as previously described in Section 3.6.

FIGURE 20: INPUT DATA FORM FOR TRCOND

COMPUTER DATA SHEET	PROGRAM TRCOND Transient, two-dimensional flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	HTOL	NLAYR

(4F10.2, I10)

QFIX	RCSNG	QRTOL

(3F10.2)

IGP	IBOUND	IWBC	IKMAX	IWAT

(5I10)

THL	AKL	BKL	VCRL	SSL

(No. of cards = NLAYR)

(5E10.2)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

SY	DINDEX

(2E10.2)

(Omit this card if IWAT = 0)

NTICR	ITST	TFACTR	TMUL	DTMUL

(2I10, 3F10.2)

FRLEN	SCFAC	XLMAX

(3F1C.2)

NDSC	NSCREN

(2I10)

IREGL	NMINL	NFRL

(No. of cards = NLAYR)

(3I10)

XSCR	HSCR

(No. of cards = NSCREN)

(2F10.2)

FIGURE 21: COMPLETED FORM FOR SAMPLE PROBLEM

COMPUTER DATA SHEET	PROGRAM TRCOND Transient, two-dimensional flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
1	0	1	1.30

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	HTOL	NLAYR
1.00	1000.00	0.00	0.50	1

(4F10.2, I10)

QFIX	RCSNG	QRTOL
100.00	1.00	0.01

(3F10.2)

IGP	IBOUND	IWBC	IKMAX	IWAT
0	0	0	1	0

(5I10)

THL	AKL	BKL	VCRL	SSL
0.20E02	0.10E02	0.20E02	0.60E-01	0.10E-04

(No. of cards = NLAYR)

(5E10.2)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

SY	DINDEX

(2E10.2)

(Omit this card if IWAT = 0)

NTICR	ITST	TFACTR	TMUL	DTMUL
10	1	50.00	2.00	0.00

(2I10, 3F10.2)

FRLEN	SCFAC	XLMAX
0.50	1.50	100.00

(3F1C.2)

NDSC	NSCREN
3	1

(2I10)

IREGL	NMINL	NFRL
4	3	5

(3I10)

(No. of cards = NLAYR)

XSCR	HSCR
10.00	10.00

(2F10.2)

(No. of cards = NSCREN)

```
**************************************************
*                                                *
*    FINITE ELEMENT ANALYSIS OF WELL PROBLEMS    *
*                                                *
* TRANSIENT,TWO REGIME FLOW TOWARD A SINGLE WELL *
*                                                *
**************************************************
```

```
**************************************************************
*                  PROBLEM NUMBER  #     1                   *
**************************************************************
```

GENERAL DATA

RADIUS OF WELL # 1.00

RADIUS OF INFLUENCE # 1000.00

HEIGHT OF WATER TABLE # 0.0

RADIUS OF WELL CASING # 1.00

DISCHARGE INTO WELL # 100.00

DISCHARGE TOLERANCE RATIO # 0.0100

OVER RELAXATION FACTOR # 1.3000

HEAD TOLERANCE # 0.5000

GRAVEL PACK INDEX # 0

BOUNDARY INDEX # 0

WELL B.C. INDEX # 0

LAYER OF MAX PERMEABILITY # 1

FORMATION PROPERTIES

LAYER NO.	THICKNESS	COEFF-A	COEF-B	CRIT. VELOCITY	SPECIFIC STORAGE
1	20.00	0.10E 02	0.20E 02	0.60E-01	0.10E-04

SCREEN NO.	BASE HEIGHT	LENGTH
1	10.00	10.00

DISCRETISATION DATA

NUMBER OF LAYERS # 1

NUMBER OF SUBREGIONS # 22

IDENTIFICATION OF ELEMENT-NODE CONNECTIONS

ELEM NO	ITYPE	NREP	IPROP	NODE1	NODE2	NODE3	NODE4
1	4	1	1	1	6	7	2
2	4	1	1	2	7	8	3
3	4	1	1	3	8	9	4
4	4	1	1	4	9	10	5
5	4	2	1	6	11	12	7
6	4	2	1	7	12	13	8
7	4	2	1	8	13	14	9
8	4	2	1	9	14	15	10
9	4	3	1	11	16	17	12
10	4	3	1	12	17	18	13
11	4	3	1	13	18	19	14
12	4	3	1	14	19	20	15
13	3	4	1	16	21	17	
14	3	4	1	18	22	19	
15	3	5	1	17	21	22	
16	3	5	1	18	22	23	
17	3	6	1	17	22	18	
18	3	6	1	18	23	20	
19	4	7	1	21	24	25	22
20	4	7	1	22	25	26	23
21	4	8	1	24	27	28	25
22	4	8	1	25	28	29	26
23	4	9	1	27	30	31	28
24	4	9	1	28	31	32	29
25	4	10	1	30	33	34	31

26	4	10	1	31	34	35	32
27	4	11	1	33	36	37	34
28	4	11	1	34	37	38	35
29	4	12	1	36	39	40	37
30	4	12	1	37	40	41	38
31	4	13	1	39	42	43	40
32	4	13	1	40	43	44	41
33	4	14	1	42	45	46	43
34	4	14	1	43	46	47	44
35	4	15	1	45	48	49	46
36	4	15	1	46	49	50	47
37	4	16	1	48	51	52	49
38	4	16	1	49	52	53	50
39	4	17	1	51	54	55	52
40	4	17	1	52	55	56	53
41	4	18	1	54	57	58	55
42	4	18	1	55	58	59	56
43	4	19	1	57	60	61	58
44	4	19	1	58	61	62	59
45	4	20	1	60	63	64	61
46	4	20	1	61	64	65	62
47	4	21	1	63	66	67	64
48	4	21	1	64	67	68	65
49	4	22	1	66	69	70	67
50	4	22	1	67	70	71	68
51	4	23	1	69	72	73	70
52	4	23	1	70	73	74	71
53	4	24	1	72	75	76	73
54	4	24	1	73	76	77	74

NODAL COORDINATES

NODE	R-COORD	Z-COORD
1	1.00	0.0
2	1.00	5.00
3	1.00	10.00
4	1.00	15.00
5	1.00	20.00
6	1.50	0.0
7	1.50	5.00
8	1.50	10.00
9	1.50	15.00
10	1.50	20.00
11	2.25	0.0
12	2.25	5.00
13	2.25	10.00
14	2.25	15.00
15	2.25	20.00
16	3.38	0.0
17	3.38	5.00
18	3.38	10.00
19	3.38	15.00
20	3.38	20.00
21	5.06	0.0
22	5.06	10.00
23	5.06	20.00
24	7.59	0.0
25	7.59	10.00
26	7.59	20.00
27	11.39	0.0
28	11.39	10.00
29	11.39	20.00
30	17.09	0.0
31	17.09	10.00
32	17.09	20.00
33	25.63	0.0
34	25.63	10.00
35	25.63	20.00
36	38.44	0.0
37	38.44	10.00
38	38.44	20.00
39	57.67	0.0
40	57.67	10.00
41	57.67	20.00
42	86.50	0.0
43	86.50	10.00
44	86.50	20.00
45	129.75	0.0
46	129.75	10.00
47	129.75	20.00
48	194.62	0.0
49	194.62	10.00
50	194.62	20.00
51	291.93	0.0
52	291.93	10.00
53	291.93	20.00
54	391.93	0.0
55	391.93	10.00
56	391.93	20.00
57	491.93	0.0
58	491.93	10.00
59	491.93	20.00
60	591.93	0.0
61	591.93	10.00
62	591.93	20.00
63	691.93	0.0
64	691.93	10.00
65	691.93	20.00
66	791.93	0.0
67	791.93	10.00
68	791.93	20.00
69	835.73	0.0
70	835.73	10.00
71	835.73	20.00
72	901.44	0.0
73	901.44	10.00
74	901.44	20.00
75	1000.00	0.0
76	1000.00	10.00
77	1000.00	20.00

TOP BOUNDARY NODES AND RADIAL COORDINATES

NODE NUMBER	R-COORDINATE
5	1.00
10	1.50
15	2.25
20	3.38
23	5.06
26	7.59
29	11.39
32	17.09
35	25.63
38	33.44
41	57.67
44	86.50
47	129.75
50	194.62
53	291.93
56	391.93
59	491.93
62	591.93
65	691.93
68	791.93
71	835.73
74	901.44
77	1000.00

LENGTH OF GROSS VECTOR # 386

```
************************************
*   TIME STEP NUMBER  #    1       *
************************************

******************************************
*        TIME  #     0.626E-02           *
******************************************
```

ESTIMATED RADIUS OF INFLUENCE # 1000.00

CORRESPONDING NO. OF NODES # 77

CORRESPONDING NO. OF ELEMENTS # 54

CORRESPONDING COMPONENT OF VECTOR NDVEC # 23

- -

DISCHARGE ITERATION NUMBER # 1

DRAWDOWN INCREMENT # -29.247

DRAWDOWN VALUE # -29.247

NUMBER OF ITERATIONS REQUIRED # 3

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.3892

DISCHARGE FROM AQUIFER INTO WELL # 69.1831

DISCHARGE FROMM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 69.183

RESIDUAL DISCHARGE # 30.8169

- -

DISCHARGE ITERATION NUMBER # 2

DRAWDOWN INCREMENT # -42.275

DRAWDOWN VALUE # -42.275

NUMBER OF ITERATIONS REQUIRED # 4

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.1600

DISCHARGE FROM AQUIFER INTO WELL # 90.7438

DISCHARGE FROMM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 90.744

RESIDUAL DISCHARGE -- # 9.2562

- -

DISCHARGE ITERATION NUMBER # 3

DRAWDOWN INCREMENT # -47.868

DRAWDOWN VALUE # -47.868

NUMBER OF ITERATIONS REQUIRED # 4

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.2063

DISCHARGE FROM AQUIFER INTO WELL # 99.1322

DISCHARGE FROMM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 99.132

RESIDUAL DISCHARGE # 0.8678

- -

DISCHARGE ITERATION NUMBER # 4

DRAWDOWN INCREMENT # -48.447

DRAWDOWN VALUE # -48.447

NUMBER OF ITERATIONS REQUIRED # 4

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.2112

DISCHARGE FROM AQUIFER INTO WELL # 99.9740

DISCHARGE FROMM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 99.974

RESIDUAL DISCHARGE # 0.0260

```
*************************************************
*                                               *
*          FINAL RESULTS OF ANALYSIS            *
*                                               *
*************************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	HEAD	1/U	W(U)
1	1.00	-4.1867	0.2235E 03	0.9399
2	1.00	1.6679	0.2235E 03	-0.3744
3	1.00	-48.4465	0.2235E 03	10.8757
4	1.00	-48.4465	0.2235E 03	10.8757
5	1.00	-48.4465	0.2235E 03	10.6757
6	1.50	-2.6096	0.9933E 02	0.5858
7	1.50	-1.6306	0.9933E 02	0.3661
8	1.50	-31.5261	0.9933E 02	7.0773
9	1.50	-33.2066	0.9933E 02	7.4545
10	1.50	-33.8055	0.9933E 02	7.5891
11	2.25	-1.8003	0.4415E 02	0.4042
12	2.25	-3.4981	0.4415E 02	0.7853
13	2.25	-19.6552	0.4415E 02	4.4124
14	2.25	-21.9374	0.4415E 02	4.9247
15	2.25	-23.0980	0.4415E 02	5.1853
16	3.38	-1.3627	0.1962E 02	0.3059
17	3.38	-4.3587	0.1962E 02	0.9785
18	3.38	-11.6641	0.1962E 02	2.6185
19	3.38	-13.4634	0.1962E 02	3.0224
20	3.38	-15.4451	0.1962E 02	3.4673
21	5.06	-0.7398	0.8721E 01	0.2119
22	5.06	-6.6221	0.8721E 01	1.4866
23	5.06	-9.3421	0.8721E 01	2.0988
24	7.59	-1.3261	0.3876E 01	0.2977
25	7.59	-3.6111	0.3876E 01	0.8106
26	7.59	-4.9902	0.3876E 01	1.1203
27	11.39	-1.0037	0.1723E 01	0.2253
28	11.39	-1.7009	0.1723E 01	0.3818
29	11.39	-2.2520	0.1723E 01	0.5055
30	17.09	-0.4864	0.7656E 00	0.1092

31	17.09	-0.6157	0.7656E 00	0.1382
32	17.09	-0.7396	0.7656E 00	0.1660
33	25.63	-0.1338	0.3403E 00	0.0300
34	25.63	-0.1428	0.3403E 00	0.0320
35	25.63	-0.1521	0.3403E 00	0.0341
36	38.44	-0.0143	0.1512E 00	0.0032
37	38.44	-0.0141	0.1512E 00	0.0032
38	38.44	-0.0138	0.1512E 00	0.0031
39	57.67	0.0002	0.6721E-01	-0.0000
40	57.67	0.0002	0.6721E-01	-0.0000
41	57.67	0.0001	0.6721E-01	-0.0000
42	86.50	-0.0000	0.2987E-01	0.0000
43	86.50	-0.0000	0.2987E-01	0.0000
44	86.50	-0.0000	0.2987E-01	0.0000
45	129.75	0.0000	0.1328E-01	-0.0000
46	129.75	0.0000	0.1328E-01	-0.0000
47	129.75	0.0000	0.1328E-01	-0.0000
48	194.62	-0.0000	0.5901E-02	0.0000
49	194.62	-0.0000	0.5901E-02	0.0000
50	194.62	-0.0000	0.5901E-02	0.0000
51	291.93	0.0000	0.2623E-02	-0.0000
52	291.93	0.0000	0.2623E-02	-0.0000
53	291.93	0.0000	0.2623E-02	-0.0000
54	391.93	-0.0000	0.1455E-02	0.0000
55	391.93	-0.0000	0.1455E-02	0.0000
56	391.93	-0.0000	0.1455E-02	0.0000
57	491.93	0.0000	0.9236E-03	-0.0000
58	491.93	0.0000	0.9236E-03	-0.0000
59	491.93	0.0000	0.9236E-03	-0.0000
60	591.93	-0.0000	0.6379E-03	0.0000
61	591.93	-0.0000	0.6379E-03	0.0000
62	591.93	-0.0000	0.6379E-03	0.0000
63	691.93	0.0000	0.4668E-03	-0.0000
64	691.93	0.0000	0.4668E-03	-0.0000
65	691.93	0.0000	0.4668E-03	-0.0000
66	791.93	-0.0000	0.3564E-03	0.0000
67	791.93	-0.0000	0.3564E-03	0.0000
68	791.93	-0.0000	0.3564E-03	0.0000
69	835.73	0.0000	0.3200E-03	-0.0000
70	835.73	0.0000	0.3200E-03	-0.0000
71	835.73	0.0000	0.3200E-03	-0.0000
72	901.44	-0.0000	0.2750E-03	0.0000
73	901.44	-0.0000	0.2750E-03	0.0000
74	901.44	-0.0000	0.2750E-03	0.0000
75	1000.00	0.0000	0.2235E-03	-0.0000
76	1000.00	0.0000	0.2235E-03	-0.0000
77	1000.00	0.0000	0.2235E-03	-0.0000

```
****************************************
*         TIME   #     0.125E-01       *
****************************************

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER    #         1
DRAWDOWN INCREMENT    #    -48.447
DRAWDOWN VALUE    #    -96.893

NUMBER OF ITERATIONS REQUIRED  #    3

TOLERANCE COUNTER FOR HEAD    #   0
ABSOLUTE MAXIMUM ERROR IN HEAD    #       0.3402

DISCHARGE FROM AQUIFER INTO WELL    #      150.2560
DISCHARGE FROM WELL STORAGE   #          0.0
TOTAL CALCULATED DISCHARGE    #        150.256
RESIDUAL DISCHARGE            #     50.2560
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER    #         2
DRAWDOWN INCREMENT    #    -16.039
DRAWDOWN VALUE    #    -64.455

NUMBER OF ITERATIONS REQUIRED  #    2

TOLERANCE COUNTER FOR HEAD    #   0
ABSOLUTE MAXIMUM ERROR IN HEAD    #       0.4017
DISCHARGE FROM AQUIFER INTO WELL    #      111.7153
DISCHARGE FROM WELL STORAGE   #          0.0
TOTAL CALCULATED DISCHARGE    #        111.715
RESIDUAL DISCHARGE            #     11.7153
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

DISCHARGE ITERATION NUMBER # 3

DRAWDOWN INCREMENT # -6.168

DRAWDOWN VALUE # -54.634

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.2795

DISCHARGE FROM AQUIFER INTO WELL # 97.9016

DISCHARGE FROMM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 97.902

RESIDUAL DISCHARGE # 2.0984

\- -

DISCHARGE ITERATION NUMBER # 4

DRAWDOWN INCREMENT # -7.684

DRAWDOWN VALUE # -56.131

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.2114

DISCHARGE FROM AQUIFER INTO WELL # 100.0617

DISCHARGE FROMM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 100.062

RESIDUAL DISCHARGE # 0.0617

```
**************************************************
*                                                *
*          FINAL RESULTS OF ANALYSIS             *
*                                                *
**************************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	HEAD	1/U	W(U)
1	1.00	-6.4753	0.4470E 03	1.4536
2	1.00	-1.1183	0.4470E 03	0.2511
3	1.00	-56.1307	0.4470E 03	12.6008
4	1.00	-56.1307	0.4470E 03	12.6008
5	1.00	-56.1307	0.4470E 03	12.6008
6	1.50	-4.3486	0.1987E 03	1.0885
7	1.50	-4.5013	0.1987E 03	1.0105
8	1.50	-38.2253	0.1987E 03	8.5812
9	1.50	-40.4876	0.1987E 03	9.0890
10	1.50	-41.2680	0.1987E 03	9.2642
11	2.25	-3.9749	0.8830E 02	0.8923
12	2.25	-6.4296	0.8830E 02	1.4434
13	2.25	-25.3586	0.8830E 02	5.6928
14	2.25	-28.5035	0.8830E 02	6.3987
15	2.25	-30.0262	0.8830E 02	6.7406
16	3.33	-3.4206	0.3924E 02	0.7679
17	3.33	-7.2895	0.3924E 02	1.6364
18	3.33	-16.3146	0.3924E 02	3.6625
19	3.33	-18.9704	0.3924E 02	4.2587
20	3.33	-21.5067	0.3924E 02	4.8280
21	5.06	-2.7677	0.1744E 02	0.6213
22	5.06	-10.2146	0.1744E 02	2.2931
23	5.06	-14.2012	0.1744E 02	3.1880
24	7.59	-3.0293	0.7752E 01	0.6800
25	7.59	-6.2524	0.7752E 01	1.4036
26	7.59	-5.3950	0.7752E 01	1.8846
27	11.39	-2.3342	0.3445E 01	0.5251
28	11.39	-3.4306	0.3445E 01	0.7701
29	11.39	-4.3271	0.3445E 01	0.9714
30	17.09	-1.2746	0.1531E 01	0.2861
31	17.09	-1.5203	0.1531E 01	0.3414
32	17.09	-1.7572	0.1531E 01	0.3945
33	25.63	-0.4339	0.6805E 00	0.0974
34	25.63	-0.4581	0.6805E 00	0.1023
35	25.63	-0.4828	0.6805E 00	0.1084
36	38.44	-0.0676	0.3025E 00	0.0152
37	38.44	-0.0672	0.3025E 00	0.0151
38	38.44	-0.0668	0.3025E 09	0.0150
39	57.67	-0.0008	0.1344E 00	0.0002

40	57.67	-0.0008	0.1344E 00	0.0002
41	57.67	-0.0009	0.1344E 00	0.0002
42	86.50	0.0001	0.5975E-01	-0.0000
43	86.50	0.0001	0.5975E-01	-0.0000
44	86.50	0.0001	0.5975E-01	-0.0000
45	129.75	-0.0000	0.2655E-01	0.0000
46	129.75	-0.0000	0.2655E-01	0.0000
47	129.75	-0.0000	0.2655E-01	0.0000
48	194.62	0.0000	0.1180E-01	-0.0000
49	194.62	0.0000	0.1180E-01	-0.0000
50	194.62	0.0000	0.1180E-01	-0.0000
51	291.93	-0.0000	0.5245E-02	0.0000
52	291.93	-0.0000	0.5245E-02	0.0000
53	291.93	-0.0000	0.5245E-02	0.0000
54	391.93	0.0000	0.2910E-02	-0.0000
55	391.93	0.0000	0.2910E-02	-0.0000
56	391.93	0.0000	0.2910E-02	-0.0000
57	491.93	-0.0000	0.1847E-02	0.0000
58	491.93	-0.0000	0.1847E-02	0.0000
59	491.93	-0.0000	0.1847E-02	0.0000
60	591.93	0.0000	0.1276E-02	-0.0000
61	591.93	0.0000	0.1276E-02	-0.0000
62	591.93	0.0000	0.1276E-02	-0.0000
63	691.93	-0.0000	0.9337E-03	0.0000
64	691.93	-0.0000	0.9337E-03	0.0000
65	691.93	-0.0000	0.9337E-03	0.0000
66	791.93	0.0000	0.7128E-03	-0.0000
67	791.93	0.0000	0.7128E-03	-0.0000
68	791.93	0.0000	0.7128E-03	-0.0000
69	835.73	-0.0000	0.6400E-03	0.0000
70	835.73	-0.0000	0.6400E-03	0.0000
71	835.73	-0.0000	0.6400E-03	0.0000
72	901.44	0.0000	0.5501E-03	-0.0000
73	901.44	0.0000	0.5501E-03	-0.0000
74	901.44	0.0000	0.5501E-03	-0.0000
75	1000.00	-0.0000	0.4470E-03	0.0000
76	1000.00	-0.0000	0.4470E-03	0.0000
77	1000.00	-0.0000	0.4470E-03	0.0000

```
*********************************
*   TIME STEP NUMBER   #   10   *
*********************************

*******************************************
*        TIME  #     0.481E 01           *
*******************************************
```

ESTIMATED RADIUS OF INFLUENCE # 1000.00

CORRESPONDING NO. OF NODES # 77

CORRESPONDING NO. OF ELEMENTS # 54

CORRESPONDING COMPONENT OF VECTOR NDVEC # 23

- -

DISCHARGE ITERATION NUMBER # 1

DRAWDOWN INCREMENT # -1.764

DRAWDOWN VALUE # -84.324

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.0011

DISCHARGE FROM AQUIFER INTO WELL # 100.0801

DISCHARGE FROM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 100.080

RESIDUAL DISCHARGE # 0.0801

```
**********************************************
*                                            *
*        FINAL RESULTS OF ANALYSIS           *
*                                            *
**********************************************
```

NODE	R-COORD	HEAD	1/U	W(U)
1	1.00	-33.1075	0.1716E 06	7.4323
2	1.00	-28.0929	0.1716E 06	6.3066
3	1.00	-84.8244	0.1716E 06	19.0422
4	1.00	-84.8244	0.1716E 06	19.0422
5	1.00	-84.8244	0.1716E 06	19.0422
6	1.50	-31.4675	0.7629E 05	7.0641
7	1.50	-31.4783	0.7629E 05	7.0654
8	1.50	-66.6605	0.7629E 05	14.9691
9	1.50	-69.2736	0.7629E 05	15.5517
10	1.50	-70.1705	0.7629E 05	15.7526
11	2.25	-30.5482	0.3391E 05	6.8577
12	2.25	-33.3503	0.3391E 05	7.4935
13	2.25	-53.4712	0.3391E 05	12.0037
14	2.25	-57.1272	0.3391E 05	12.3245
15	2.25	-53.8584	0.3391E 05	13.2131
16	3.38	-29.3698	0.1507E 05	6.7055
17	3.38	-34.1735	0.1507E 05	7.6716
18	3.38	-43.9716	0.1507E 05	9.8712
19	3.38	-47.1593	0.1507E 05	10.5890
20	3.38	-50.0018	0.1507E 05	11.2242
21	5.06	-28.9549	0.6698E 04	6.5001
22	5.06	-37.3045	0.6698E 04	8.3745
23	5.06	-42.9996	0.6698E 04	9.4509
24	7.59	-28.7903	0.2977E 04	6.4532
25	7.59	-32.5595	0.2977E 04	7.3093
26	7.59	-35.2581	0.2977E 04	7.9151
27	11.39	-27.2374	0.1323E 04	6.1150
28	11.39	-28.5310	0.1323E 04	6.4161
29	11.39	-29.7285	0.1323E 04	6.6737
30	17.09	-24.6109	0.5880E 03	5.5249
31	17.09	-24.9529	0.5880E 03	5.6039
32	17.09	-25.3041	0.5880E 03	5.6805
33	25.63	-21.3549	0.2613E 03	4.7940
34	25.63	-21.4051	0.2613E 03	4.8052
35	25.63	-21.4557	0.2613E 03	4.8166
36	38.44	-17.8635	0.1161E 03	4.0102
37	38.44	-17.8643	0.1161E 03	4.0105
38	38.44	-17.8660	0.1161E 03	4.0107
39	57.67	-14.3571	0.5162E 02	3.2230
40	57.67	-14.3570	0.5162E 02	3.2230
41	57.67	-14.3569	0.5162E 02	3.2230
42	36.50	-10.9211	0.2294E 02	2.4517
43	86.50	-10.9211	0.2294E 02	2.4517
44	36.50	-10.9211	0.2294E 02	2.4517
45	127.75	-7.6402	0.1020E 02	1.7151
46	129.75	-7.6402	0.1020E 02	1.7151
47	129.75	-7.6402	0.1020E 02	1.7151
48	194.62	-4.6777	0.4532E 01	1.0501
49	194.62	-4.6777	0.4532E 01	1.0501
50	194.62	-4.6777	0.4532E 01	1.0501
51	291.93	-2.3049	0.2014E 01	0.5174
52	291.93	-2.3049	0.2014E 01	0.5174
53	291.93	-2.3049	0.2014E 01	0.5174
54	391.93	-1.0958	0.1117E 01	0.2460
55	391.93	-1.0958	0.1117E 01	0.2460
56	391.93	-1.0958	0.1117E 01	0.2460
57	491.93	-0.5069	0.7093E 00	0.1138
58	491.93	-0.5069	0.7093E 00	0.1138
59	491.93	-0.5069	0.7093E 00	0.1138
60	591.93	-0.2284	0.4699E 00	0.0513
61	591.93	-0.2284	0.4699E 00	0.0513
62	591.93	-0.2284	0.4699E 00	0.0513
63	691.93	-0.1012	0.3585E 00	0.0227
64	691.93	-0.1012	0.3585E 00	0.0227
65	691.93	-0.1012	0.3585E 00	0.0227
66	791.93	-0.0454	0.2737E 00	0.0102
67	791.93	-0.0454	0.2737E 00	0.0102
68	791.93	-0.0454	0.2737E 00	0.0102
69	835.73	-0.0325	0.2458E 00	0.0073
70	835.73	-0.0325	0.2458E 00	0.0073
71	835.73	-0.0325	0.2458E 00	0.0073
72	901.44	-0.0211	0.2112E 00	0.0047
73	901.44	-0.0211	0.2112E 00	0.0047
74	901.44	-0.0211	0.2112E 00	0.0047
75	1000.00	-0.0158	0.1716E 00	0.0035
76	1000.00	-0.0158	0.1716E 00	0.0035
77	1000.00	-0.0158	0.1716E 00	0.0035

```
******************************************
*        TIME  #      0.541E 01          *
******************************************

- - - - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER     #          1

DRAWDOWN INCREMENT     #    -1.764

DRAWDOWN VALUE    #    -86.589

NUMBER OF ITERATIONS REQUIRED  #      2

TOLERANCE COUNTER FOR HEAD    #   0

ABSOLUTE MAXIMUM ERROR IN HEAD     #        0.0322

DISCHARGE FROM AQUIFER INTO WELL     #      100.5216

DISCHARGE FROM WELL STORAGE   #           0.0

TOTAL CALCULATED DISCHARGE     #       100.522

RESIDUAL DISCHARGE        #      0.5216 [illegible]
```

DISCHARGE ITERATION NUMBER # 2

DRAWDOWN INCREMENT # -1.315

DRAWDOWN VALUE # -86.139

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.0004

DISCHARGE FROM AQUIFER INTO WELL # 100.0764

DISCHARGE FROM WELL STORAGE # 0.0

TOTAL CALCULATED DISCHARGE # 100.076

RESIDUAL DISCHARGE # 0.0764

```
*********************************************************
*                                                       *
*          FINAL RESULTS OF ANALYSIS                    *
*                                                       *
*********************************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD.	HEAD	1/U	W(U)
1	1.00	-34.4257	0.2289E 06	7.7282
2	1.00	-29.4114	0.2289E 05	6.6026
3	1.00	-86.1395	0.2289E 06	19.3374
4	1.00	-36.1395	0.2289E 06	19.3374
5	1.00	-86.1395	0.2289E 06	19.3374
6	1.50	-32.7858	0.1017E 06	7.3601
7	1.50	-32.7916	0.1017E 06	7.3614
8	1.50	-67.9968	0.1017E 06	15.2646
9	1.50	-70.5399	0.1017E 06	15.8467
10	1.50	-71.4367	0.1017E 06	16.0430
11	2.25	-31.8665	0.4521E 05	7.1537
12	2.25	-34.6985	0.4521E 05	7.7895
13	2.25	-54.7884	0.4521E 05	12.2994
14	2.25	-58.4443	0.4521E 05	13.1201
15	2.25	-60.1753	0.4521E 05	13.5087
16	3.38	-31.1880	0.2009E 05	7.0014
17	3.38	-35.4916	0.2009E 05	7.9675
18	3.38	-45.2893	0.2009E 05	10.1670
19	3.38	-48.4869	0.2009E 05	10.6848
20	3.38	-51.3192	0.2009E 05	11.5206
21	5.06	-30.2730	0.8930E 04	6.7960
22	5.06	-38.6224	0.8930E 04	8.6703
23	5.06	-43.4173	0.8930E 04	9.7467
24	7.59	-30.1086	0.3969E 04	6.7591
25	7.59	-33.8772	0.3969E 04	7.6051
26	7.59	-36.5758	0.3969E 04	8.2109
27	11.39	-29.5556	0.1764E 04	6.4107
28	11.39	-29.8733	0.1764E 04	6.7118
29	11.39	-31.0457	0.1764E 04	6.9694
30	17.09	-25.9269	0.7840E 03	5.8203
31	17.09	-26.2788	0.7840E 03	5.8953
32	17.09	-26.6201	0.7840E 03	5.9759
33	25.63	-22.6580	0.3494E 03	5.0887
34	25.63	-22.7152	0.3484E 03	5.1000
35	25.63	-22.7638	0.3494E 03	5.1114
36	38.44	-19.1700	0.1542E 03	4.3035
37	38.44	-19.1712	0.1542E 03	4.3037
38	38.44	-19.1725	0.1542E 03	4.3040
39	57.67	-15.6487	0.6883E 02	3.5130
40	57.67	-15.6496	0.6883E 02	3.5127
41	57.67	-15.6505	0.6883E 02	3.5129
42	86.50	-12.1753	0.3059E 02	2.7342
43	86.50	-12.1759	0.3059E 02	2.7343
44	86.50	-12.1760	0.3059E 02	2.7343
45	129.75	-8.6206	0.1360E 02	1.9819
46	129.75	-8.6206	0.1360E 02	1.9819
47	129.75	-8.6206	0.1360E 02	1.9817
48	194.62	-5.7246	0.6042E 01	1.2851
49	194.62	-5.7246	0.6042E 01	1.2851
50	194.62	-5.7246	0.6042E 01	1.2851
51	291.93	-3.1004	0.2686E 01	0.6960
52	291.93	-3.1004	0.2686E 01	0.6960
53	291.93	-3.1004	0.2686E 01	0.6960
54	391.93	-1.6437	0.1420E 01	0.3690
55	391.93	-1.6437	0.1420E 01	0.3690
56	391.93	-1.6437	0.1420E 01	0.3690
57	491.93	-0.8553	0.9459E 00	0.1920
58	491.93	-0.8553	0.9459E 00	0.1920
59	491.93	-0.8553	0.9459E 00	0.1920
60	591.93	-0.4362	0.6532E 00	0.0979
61	591.93	-0.4362	0.6532E 00	0.0979
62	591.93	-0.4362	0.6532E 00	0.0979
63	691.93	-0.2197	0.4700E 00	0.0493
64	691.93	-0.2197	0.4700E 00	0.0493
65	691.93	-0.2197	0.4700E 00	0.0493
66	791.93	-0.1126	0.3649E 00	0.0253
67	791.93	-0.1126	0.3649E 00	0.0253
68	791.93	-0.1126	0.3649E 00	0.0253
69	891.93	-0.0559	0.3277E 00	0.0183
70	891.93	-0.0559	0.3277E 00	0.0183
71	891.93	-0.0559	0.3277E 00	0.0183
72	991.44	-0.0516	0.2816E 00	0.0137
73	991.44	-0.0610	0.2816E 00	0.0137
74	991.44	-0.0610	0.2816E 00	0.0137
75	1000.00	-0.0490	0.2289E 00	0.0110
76	1000.00	-0.0490	0.2289E 00	0.0110
77	1000.00	-0.0490	0.2289E 00	0.0110

TOTAL NUMBER OF PUNCHED CARDS # 14

3.8 STFREE - PROGRAM FOR SOLVING STEADY, FREE SURFACE FLOW USING TRIANGULAR ELEMENTS

The program STFREE is used to analyse steady, two-dimensional flow in an unconfined aquifer. The presence of a free surface and/or a seepage face at the well is taken into account. Only fully penetrating wells with or without gravel packing are considered. For a partially penetrating well, the program neglects the portion of aquifer which lies directly below the bottom of the well. Up to 5 screened intervals can be handled in the current version.

Figures 22 and 23 show the input data form and completed form for a sample run. Two problems were solved, with IVEL = 1 and IDISCR = 1. The first problem involves steady two-regime flow towards a gravel packed well which is fully screened through the saturated thickness of the aquifer. The second problem involves steady, two-regime flow towards a partially screened well. For both problems, results were obtained for a value of HW equal to 10 ft. above the base of aquifer. (In this program, all values of the hydraulic head are measured from the aquifer base.) The printed output is shown on pages 94 to 104. It consists of the following:

(i) a reformatted playback of the input data;

(ii) the generated discretisation data, which consists of node connections of elements, nodal co-ordinates, top boundary nodes and their radial co-ordinates, a list of nodes of variable z-co-ordinates and elements of variable shape, a list of prescribed nodes and head values and the length of the gross vector;

(iii) the first table of results for HW = 10 ft., which lists the node numbers, radial and vertical co-ordinates, head values, Rho- and Tzi-co-ordinates which are defined as $\rho = r/r_0$, $\xi = z/m$, and dimensionless drawdowns, s/s_w (not shown on pages 97 and 102);

(iv) the second table of results, which lists radial co-ordinates and heights of the free surface nodes;

(v) the third table of results, which lists all the element numbers, radial and vertical components of element velocities and values of the flow type index IDARCY;

(vi) the final table of results, which lists all the node numbers, radial co-ordinates, values of the dimensionless well function $W(u) = 2\pi h_0(h_0 - h)K/Q$ and values of $1/u = r/r_0$. The value of the non-linear flow parameter, $\lambda = bQK/2\pi r_0 h_0$ is also included in the table.

To ensure that no error was made in the preparation of input data, a check was made on the discretisation data by plotting the nodal co-ordinates and element connections. The plot is shown in Figure 24. The well screen depicted in the figure is for problem 2.

FIGURE 22: INPUT DATA FORM FOR STFREE

COMPUTER DATA SHEET	PROGRAM STFREE Steady, free surface flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL

(5F10.2)

IGP	NSTEP

(2I10)

AK	BK	VCR

(3F10.3)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

HW

(F10.2)

FRLEN	SCFAC	XLMAX	IREG	NMIN

(3F10.2, 2I10)

NDSC	NSCREN

(2I10)

XSCR	HSCR

(2F1C.2)

No. of cards = NSCREN

FIGURE 23: COMPLETED FORM FOR 2 SAMPLE PROBLEMS

COMPUTER DATA SHEET	PROGRAM STFREE Steady, free surface flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
2	1	1	1.50

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL
0.50	1000.00	50.00	50.00	0.20

(5F10.2)

IGP	NSTEP
1	2

(2I10)

AK	BK	VCR
10.000	20.000	0.0125

(3F10.3)

AGP	BGP	VGP	THGP	BTGP
1.000	2.000	0.125	0.500	0.000

(5F10.3)

(Omit this card if IGP = 0)

HW
10.00

(F10.2)

FRLEN	SCFAC	XLMAX	IREG	NMIN
0.10	1.50	150.00	2	2

(3F10.2, 2I10)

NDSC	NSCREN
7	1

(2I10)

XSCR	HSCR
0.00	50.00

(2F10.2)

No. of cards = NSCREN

COMPUTER DATA SHEET	PROGRAM STFREE Steady, free surface flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL
1.00	1000.00	50.00	50.00	0.20

(5F10.2)

IGP	NSTEP
0	2

(2I10)

AK	BK	VCR
1.000	2.000	0.125

(3F10.3)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

HW
10.00

(F10.2)

FRLEN	SCFAC	XLMAX	IREG	NMIN
0.30	1.50	150.00	2	2

(3F10.2, 2I10)

NDSC	NSCREN
7	1

(2I10)

XSCR	HSCR
20.00	30.00

(2F10.2)

No. of cards = NSCREN

```
*************************************************************
*              PROBLEM NUMBER   #           1               *
*************************************************************
```

GENERAL INPUT DATA

RADIUS OF WELL # 0.50

EXTERNAL RADIUS # 1000.00

HEIGHT OF WATER TABLE AT EXTERNAL RADIUS # 50.00

INITIAL SATURATED THICKNESS OF AQUIFER . # 50.00

OVER-RELAXATION FACTOR # 1.50

PRESCRIBED HEAD TOLERANCE # 0.20

GRAVEL PACK INDEX # 1

APPLY 2 STEP ITERATIVE SCHEME

AQUIFER PROPERTIES

FORCHHEIMER COEFF. A # 10.0000

FORCHHEIMER COEFF. B # 20.0000

CRITICAL FLOW VELOCITY # 0.0125

GRAVEL PACK PROPERTIES

COEFFICIENT A # 1.0000

COEFFICIENT B # 2.0000

CRITICAL VELOCITY # 0.1250

COEFFICIENT K # 0.8000

THICKNESS OF PACK # 0.50

RADIUS OF PACK # 1.00

SCREEN NO.	BASE HEIGHT	LENGTH
1	0.0	50.00

DISCRETISATION DATA

IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS

ELEM NO	NODE1	NODE2	NODE3	REPETITION NUMBER
1	1	10	2	1
2	3	11	4	1
3	5	12	6	1
4	2	10	11	2
5	4	11	12	2
6	6	12	13	2
7	2	11	3	3
8	4	12	5	3
9	6	13	7	3
10	10	16	11	4
11	11	17	12	4
12	12	18	13	4
13	11	16	17	5
14	12	17	18	5
15	13	18	19	5
16	16	21	17	6
17	17	22	18	7
18	18	23	19	8
19	17	21	22	9
20	18	22	23	10
21	21	25	22	11
22	22	26	23	11
23	22	25	26	12
24	23	26	27	12
25	25	29	26	13

26	26	29	30	14
27	26	30	27	15
28	29	32	30	16
29	30	32	33	17
30	32	35	33	18
31	33	36	34	19
32	33	35	36	20
33	35	37	36	21
34	36	37	38	22
35	37	39	39	23
36	38	39	40	24
37	39	41	40	25
38	40	41	42	26
39	41	43	42	27
40	42	43	44	28
41	43	45	44	29
42	44	45	46	30
43	45	47	46	31
44	46	47	48	32
45	47	49	48	33
46	48	49	50	34
47	49	51	50	35
48	50	51	52	36
49	51	53	52	37
50	52	53	54	38
51	53	55	54	39
52	54	55	56	40
53	55	57	56	41
54	56	57	58	42
55	57	59	58	43
56	58	59	60	44
57	59	61	60	45
58	60	61	62	46
59	61	63	62	47
60	62	63	64	48
61	63	65	64	49
62	64	65	66	50
63	65	67	66	51
64	66	67	68	52
65	7	13	8	53
66	8	14	9	54
67	8	13	14	55
68	9	14	15	56
69	13	19	14	57
70	14	20	15	58
71	14	19	20	59
72	19	23	20	60
73	20	23	24	61
74	23	27	24	62
75	24	27	28	63
76	27	30	28	64
77	28	30	31	65
78	30	33	31	66
79	31	33	34	67

NODE DATA

NODE	R-COORD	Z-COORD
1	0.50	0.0
2	0.50	3.47
3	0.50	6.94
4	0.50	10.40
5	0.50	13.87
6	0.50	17.34
7	0.50	20.81
8	0.50	31.21
9	0.50	41.61
10	0.60	0.0
11	0.60	6.94
12	0.60	13.87
13	0.60	20.81
14	0.60	31.21
15	0.60	41.62
16	0.75	0.0
17	0.75	6.94
18	0.75	13.87
19	0.75	20.81
20	0.75	41.63
21	0.97	0.0
22	0.97	10.40
23	0.97	20.81
24	0.97	41.63
25	1.31	0.0
26	1.31	10.40
27	1.31	20.81
28	1.31	41.64
29	1.82	0.0
30	1.82	20.81
31	1.82	41.65
32	2.58	0.0
33	2.58	20.81
34	2.58	41.66
35	3.72	0.0

36	3.72	41.69
37	5.43	0.0
38	5.43	41.72
39	7.99	0.0
40	7.99	41.77
41	11.83	0.0
42	11.83	41.84
43	17.60	0.0
44	17.60	41.95
45	26.25	0.0
46	26.25	42.11
47	39.22	0.0
48	39.22	42.34
49	58.69	0.0
50	58.69	42.66
51	87.88	0.0
52	87.88	43.13
53	131.67	0.0
54	131.67	43.77
55	197.35	0.0
56	197.35	44.65
57	295.88	0.0
58	295.88	45.79
59	443.67	0.0
60	443.67	47.20
61	593.67	0.0
62	593.67	48.31
63	679.21	0.0
64	679.21	48.82
65	807.53	0.0
66	807.53	49.42
67	1000.00	0.0
68	1000.00	50.00

TOP BOUNDARY NODES AND RADIAL COORDINATES.

NODE NUMBER	R-COORDINATE	SURFACE HEIGHT
9	0.50	41.61
15	0.60	41.62
20	0.75	41.63
24	0.97	41.63
28	1.31	41.64
31	1.82	41.65
34	2.58	41.66
36	3.72	41.69
38	5.43	41.72
40	7.99	41.77
42	11.83	41.84
44	17.60	41.95
46	26.25	42.11
48	39.22	42.34
50	58.69	42.66
52	87.88	43.13
54	131.67	43.77
56	197.35	44.65
58	295.88	45.79
60	443.67	47.20
62	593.67	48.31
64	679.21	48.82
66	807.53	49.42
68	1000.00	50.00

NODES OF FLEXIBLE Z-COORDINATES

8	9	14	15	20	24	28	31	34	36
38	40	42	44	46	48	50			

ELEMENTS OF VARIABLE SHAPE ARE NUMBER 30 TO 79

PRESCRIBED NODES AND HEAD VALUES

NODE	PRESCRIBED VALUES
1	10.000
2	10.000
3	10.000
4	10.403
5	13.970
6	17.338
7	20.805
8	31.208
67	50.000
68	50.000

LENGTH OF GROSS VECTOR # 304

```
*********************************************
*   WATER LEVEL IN THE WELL # 10.00         *
*********************************************
```

NUMBER OF ITERATIONS REQUIRED # 6

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.0789

NODE	R-COORDINATE	Z-COORDINATE	HEAD VALUE	RHO-COORD	TZI-COORD
1	0.50	0.0	10.0000	0.0005	0.0
2	0.50	3.47	10.0000	0.0005	0.0694
3	0.50	6.94	10.0000	0.0005	0.1387
4	0.50	10.40	10.4075	0.0005	0.2081
5	0.50	13.67	13.8783	0.0005	0.2774
6	0.50	17.34	17.3472	0.0005	0.3469
7	0.50	20.81	20.8161	0.0005	0.4161
8	0.50	24.70	24.7099	0.0005	0.4940
9	0.50	28.60	28.5954	0.0005	0.5719
10	0.60	0.0	10.0548	0.0006	0.0
11	0.60	6.94	10.4042	0.0006	0.1387
12	0.60	13.67	14.1868	0.0006	0.2774
13	0.60	20.81	20.9116	0.0006	0.4161
14	0.60	24.70	24.7693	0.0006	0.4941
15	0.60	28.60	28.6022	0.0006	0.5720
16	0.75	0.0	10.2051	0.0007	0.0
17	0.75	6.94	11.0477	0.0007	0.1387
18	0.75	13.87	14.7566	0.0007	0.2774
19	0.75	20.81	21.0599	0.0007	0.4161
20	0.75	28.64	28.6450	0.0007	0.5729
21	0.97	0.0	10.3636	0.0010	0.0
22	0.97	10.40	12.8724	0.0010	0.2081
23	0.97	20.81	21.1924	0.0010	0.4161
24	0.97	28.86	28.8611	0.0010	0.5772
25	1.31	0.0	14.1575	0.0013	0.0
26	1.31	10.40	17.4765	0.0013	0.2081
27	1.31	20.81	23.1429	0.0013	0.4161
28	1.31	29.90	29.6985	0.0013	0.5980
29	1.82	0.0	17.3053	0.0018	0.0
30	1.82	20.81	25.1100	0.0018	0.4161
31	1.82	30.60	30.5988	0.0018	0.6120
32	2.58	0.0	21.6254	0.0026	0.0
33	2.58	20.81	27.4375	0.0026	0.4161
34	2.58	31.10	31.0971	0.0026	0.6219
35	3.72	0.0	25.1460	0.0037	0.0
36	3.72	31.72	31.7187	0.0037	0.6344
37	5.43	0.0	28.0424	0.0054	0.0
38	5.43	33.26	33.2618	0.0054	0.6652
39	7.99	0.0	30.6904	0.0080	0.0
40	7.99	34.62	34.6167	0.0080	0.6923
41	11.83	0.0	33.1203	0.0118	0.0
42	11.83	35.87	35.8701	0.0118	0.7174
43	17.60	0.0	35.3405	0.0176	0.0
44	17.60	37.09	37.0867	0.0176	0.7417
45	26.25	0.0	37.3474	0.0262	0.0
46	26.25	38.32	38.3172	0.0262	0.7663
47	39.22	0.0	39.1418	0.0392	0.0
48	39.22	39.60	39.5992	0.0392	0.7920
49	58.69	0.0	40.7599	0.0587	0.0
50	58.69	40.94	40.9357	0.0587	0.8187
51	87.88	0.0	42.2432	0.0879	0.0
52	87.88	42.30	42.2999	0.0879	0.8460
53	131.67	0.0	43.6391	0.1317	0.0
54	131.67	43.66	43.6572	0.1317	0.8731
55	197.35	0.0	44.9797	0.1974	0.0
56	197.35	44.99	44.9863	0.1974	0.8997
57	295.88	0.0	46.2784	0.2959	0.0
58	295.88	46.28	46.2811	0.2959	0.9256
59	443.67	0.0	47.5414	0.4437	0.0
60	443.67	47.54	47.5418	0.4437	0.9508
61	593.67	0.0	48.4348	0.5937	0.0
62	593.67	48.43	48.4343	0.5937	0.9687
63	679.21	0.0	48.8436	0.6792	0.0
64	679.21	48.84	48.8444	0.6792	0.9769
65	807.53	0.0	49.3646	0.8075	0.0
66	807.53	49.37	49.3651	0.8075	0.9873
67	1000.00	0.0	50.0000	1.0000	0.0
68	1000.00	50.00	50.0000	1.0000	1.0000

FREE SURFACE POSITON

RADIUS	SURFACE HEIGHT
0.50	28.5954
0.60	28.6022
0.75	28.6450
0.97	28.8611
1.31	29.8985
1.82	30.5988
2.58	31.0971
3.72	31.7187
5.43	33.2618
7.99	34.6167
11.83	35.8701
17.60	37.0867
26.25	38.3172
39.22	39.5992
58.69	40.9357
87.88	42.2999
131.67	43.6572
197.35	44.9863
295.88	46.2811
443.67	47.5418
593.67	48.4343
679.21	48.8444
807.53	49.3651
1000.00	50.0000

SURFACE TOLERANCE COUNTER # 0

MAXIMUM ERROR OF SURFACE HEIGHT # 0.0789

```
*****************************************
*          ELEMENT VELOCITIES           *
*****************************************
```

ELEM NO.	RADIAL VEL	VERTIC VEL	IDARCY
1	-0.050	0.0	1
2	-0.264	-0.003	1
3	-0.213	-0.069	1
4	-0.171	-0.004	1
5	-0.753	-0.022	1
6	-0.151	-0.073	1
7	-0.264	0.0	1
8	-0.213	-0.069	1
9	-0.078	-0.032	1
10	-0.086	-0.004	1
11	-0.276	-0.035	1
12	-0.250	-0.064	1
13	-0.276	-0.008	1
14	-0.252	-0.035	1
15	-0.077	-0.075	1
16	-0.063	-0.011	1
17	0.012	-0.049	1
18	-0.053	-0.077	1
19	-0.281	-0.015	1
20	-0.259	-0.052	1
21	-0.543	-0.012	1
22	-0.613	-0.036	1
23	-0.613	-0.014	1
24	-0.342	-0.032	1
25	-0.398	-0.018	1
26	-0.425	-0.019	1
27	-0.256	-0.036	1
28	-0.310	-0.022	1
29	-0.214	-0.020	1
30	-0.216	-0.019	1
31	-0.032	-0.033	1
32	-0.139	-0.016	1
33	-0.134	-0.016	1
34	-0.067	-0.014	1
35	-0.088	-0.013	1
36	-0.043	-0.010	1
37	-0.057	-0.010	1
38	-0.028	-0.007	1
39	-0.036	-0.007	1
40	-0.019	-0.005	1
41	-0.022	-0.005	1
42	-0.013	-0.002	1
43	-0.013	-0.002	1
44	-0.010	-0.001	0
45	-0.008	-0.001	0
46	-0.007	-0.000	0
47	-0.005	-0.000	0
48	-0.005	-0.000	0
49	-0.003	-0.000	0
50	-0.003	-0.000	0
51	-0.002	-0.000	0
52	-0.002	-0.000	0
53	-0.001	-0.000	0
54	-0.001	-0.000	0
55	-0.001	-0.000	0
56	-0.001	-0.000	0
57	-0.001	-0.000	0
58	-0.001	0.000	0
59	-0.000	0.000	0
60	-0.000	-0.000	0
61	-0.000	-0.000	0
62	-0.000	-0.000	0
63	-0.000	-0.000	0
64	-0.000	0.0	0
65	-0.078	-0.031	1
66	-0.047	-0.033	1
67	-0.047	-0.033	1
68	-0.000	-0.034	1
69	-0.076	-0.080	1
70	-0.005	-0.064	1
71	-0.035	-0.082	1
72	-0.053	-0.081	1
73	-0.019	-0.031	1
74	-0.342	-0.056	1
75	-0.030	-0.032	1
76	-0.256	-0.049	1
77	-0.053	-0.049	1
78	-0.214	-0.039	1
79	-0.033	-0.033	1

TOTAL DISCHARGE INTO THE WELL = 90.8703

****
*   STEADY STATE TYPE CURVE   *
****

LAMBDA # 0.0006

NODE NUMBER	R-COORDINATE	FUNCTION W(U)	ARGUMENT U
1	0.50	0.8098E 01	0.5000E-03
2	0.50	0.8098E 01	0.5000E-03
3	0.50	0.8098E 01	0.5000E-03
4	0.50	0.8070E 01	0.5000E-03
5	0.50	0.7786E 01	0.5000E-03
6	0.50	0.7420E 01	0.5000E-03
7	0.50	0.6974E 01	0.5000E-03
8	0.50	0.6375E 01	0.5000E-03
9	0.50	0.5677E 01	0.5000E-03
10	0.60	0.8095E 01	0.6000E-03
11	0.60	0.8070E 01	0.6000E-03
12	0.60	0.7757E 01	0.6000E-03
13	0.60	0.6960E 01	0.6000E-03
14	0.60	0.6365E 01	0.6000E-03
15	0.60	0.5675E 01	0.6000E-03
16	0.75	0.8084E 01	0.7500E-03
17	0.75	0.8024E 01	0.7500E-03
18	0.75	0.7701E 01	0.7500E-03
19	0.75	0.6940E 01	0.7500E-03
20	0.75	0.5667E 01	0.7500E-03
21	0.97	0.8073E 01	0.9750E-03
22	0.97	0.7877E 01	0.9750E-03
23	0.97	0.6920E 01	0.9750E-03
24	0.97	0.5625E 01	0.9750E-03
25	1.31	0.7756E 01	0.1312E-02
26	1.31	0.7405E 01	0.1312E-02
27	1.31	0.6623E 01	0.1312E-02
28	1.31	0.5419E 01	0.1312E-02
29	1.82	0.7366E 01	0.1819E-02
30	1.82	0.6308E 01	0.1819E-02
31	1.82	0.5276E 01	0.1819E-02
32	2.58	0.6358E 01	0.2578E-02
33	2.58	0.5895E 01	0.2578E-02
34	2.58	0.5173E 01	0.2578E-02
35	3.72	0.6302E 01	0.3717E-02
36	3.72	0.5041E 01	0.3717E-02
37	5.43	0.5782E 01	0.5426E-02
38	5.43	0.4703E 01	0.5426E-02
39	7.99	0.5257E 01	0.7999E-02
40	7.99	0.4392E 01	0.7999E-02
41	11.83	0.4734E 01	0.1183E-01
42	11.83	0.4094E 01	0.1183E-01
43	17.60	0.4221E 01	0.1760E-01
44	17.60	0.3795E 01	0.1760E-01
45	26.25	0.3729E 01	0.2625E-01
46	26.25	0.3482E 01	0.2625E-01
47	39.22	0.3266E 01	0.3922E-01
48	39.22	0.3144E 01	0.3922E-01
49	58.69	0.2830E 01	0.5869E-01
50	58.69	0.2781E 01	0.5869E-01
51	87.88	0.2414E 01	0.8788E-01
52	87.88	0.2398E 01	0.8788E-01
53	131.67	0.2010E 01	0.1317E 00
54	131.67	0.2004E 01	0.1317E 00
55	197.35	0.1509E 01	0.1974E 00
56	197.35	0.1507E 01	0.1974E 00
57	295.88	0.1209E 01	0.2959E 00
58	295.88	0.1208E 01	0.2959E 00
59	443.67	0.8092E 00	0.4437E 00
60	443.67	0.8091E 00	0.4437E 00
61	593.67	0.5199E 00	0.5937E 00
62	593.67	0.5200E 00	0.5937E 00
63	679.21	0.3957E 00	0.6792E 00
64	679.21	0.3854E 00	0.6792E 00
65	807.53	0.2130E 00	0.8075E 00
66	807.53	0.2129E 00	0.8075E 00
67	1000.00	0.0	0.1000E 01
68	1000.00	0.0	0.1000E 01

```
*********************************************************
*                PROBLEM NUMBER   #       2             *
*********************************************************
```

GENERAL INPUT DATA

RADIUS OF WELL # 1.00

EXTERNAL RADIUS # 1000.00

HEIGHT OF WATER TABLE AT EXTERNAL RADIUS # 50.00

INITIAL SATURATED THICKNESS OF AQUIFER # 50.00

OVER-RELAXATION FACTOR # 1.50

PRESCRIBED HEAD TOLERANCE # 0.20

GRAVEL PACK INDEX # 0

APPLY 2 STEP ITERATIVE SCHEME

AQUIFER PROPERTIES

FORCHHEIMER COEFF. A # 1.0000

FORCHHEIMER COEFF. B # 2.0000

CRITICAL FLOW VELOCITY # 0.1250

SCREEN NO.	BASE HEIGHT	LENGTH
1	20.00	30.00

DISCRETISATION DATA

IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS

ELEM NO	NODE1	NODE2	NODE3	REPETITION NUMBER
1	1	10	2	1
2	3	11	4	1
3	5	12	6	1
4	2	10	11	2
5	4	11	12	2
6	6	12	13	2
7	2	11	3	3
8	4	12	5	3
9	6	13	7	3
10	10	16	11	4
11	11	17	12	4
12	12	18	13	4
13	11	16	17	5
14	12	17	18	5
15	13	18	19	5
16	16	21	17	6
17	17	22	18	7
18	18	23	19	8
19	17	21	22	9
20	18	22	23	10
21	21	25	22	11
22	22	26	23	11
23	22	25	26	12
24	23	26	27	12
25	25	29	26	13
26	26	29	30	14
27	26	30	27	15
28	29	32	30	16
29	30	32	33	17
30	32	35	33	18
31	33	36	34	19
32	33	35	36	20
33	35	37	36	21
34	36	37	38	22
35	37	39	38	23
36	38	39	40	24
37	39	41	40	25
38	40	41	42	26
39	41	43	42	27
40	42	43	44	28
41	43	45	44	29
42	44	45	46	30
43	45	47	46	31
44	46	47	48	32
45	47	49	48	33
46	48	49	50	34
47	49	51	50	35
48	50	51	52	36
49	51	53	52	37
50	52	53	54	38
51	53	55	54	39
52	54	55	56	40
53	55	57	56	41
54	56	57	58	42
55	57	59	58	43
56	58	59	60	44
57	59	61	60	45
58	60	61	62	46
59	61	63	62	47
60	62	63	64	48
61	7	13	8	49
62	8	14	9	50
63	8	13	14	51
64	9	14	15	52
65	13	19	14	53
66	14	20	15	54
67	14	19	20	55
68	19	23	20	56
69	20	23	24	57
70	23	27	24	58
71	24	27	28	59
72	27	30	28	60
73	28	30	31	61
74	30	33	31	62

NODE DATA

NODE	R-COORD	Z-COORD
1	1.00	0.0
2	1.00	3.37
3	1.00	6.73
4	1.00	10.10
5	1.00	13.47
6	1.00	16.84
7	1.00	20.20
8	1.00	30.30
9	1.00	40.41
10	1.30	0.0
11	1.30	6.73
12	1.30	13.47
13	1.30	20.20
14	1.30	30.32
15	1.30	40.44
16	1.75	0.0
17	1.75	6.73
18	1.75	13.47
19	1.75	20.20
20	1.75	40.45
21	2.42	0.0
22	2.42	10.10
23	2.42	20.20
24	2.42	40.46
25	3.44	0.0
26	3.44	10.10
27	3.44	20.20
28	3.44	40.49
29	4.96	0.0
30	4.96	20.20
31	4.96	40.52
32	7.23	0.0
33	7.23	20.20
34	7.23	40.57
35	10.65	0.0
36	10.65	40.65
37	15.78	0.0
38	15.78	40.76
39	23.47	0.0
40	23.47	40.92
41	35.00	0.0
42	35.00	41.15
43	52.30	0.0
44	52.30	41.49
45	78.25	0.0
46	78.25	41.97
47	117.17	0.0
48	117.17	42.64
49	175.56	0.0
50	175.56	43.56
51	263.14	0.0
52	263.14	44.78
53	394.50	0.0
54	394.50	46.31
55	544.50	0.0
56	544.50	47.70
57	694.50	0.0
58	694.50	48.75
59	758.82	0.0
60	758.82	49.11
61	855.29	0.0
62	855.29	49.56
63	1000.00	0.0
64	1000.00	50.00

TOP BOUNDARY NODES AND RADIAL COORDINATES.

NODE NUMBER	R-COORDINATE	SURFACE HEIGHT
9	1.00	40.41
15	1.30	40.44
20	1.75	40.45
24	2.42	40.46
28	3.44	40.49
31	4.96	40.52
34	7.23	40.57
36	10.65	40.65
38	15.78	40.76
40	23.47	40.92
42	35.00	41.15
44	52.30	41.49
46	78.25	41.97
48	117.17	42.64
50	175.56	43.56
52	263.14	44.78
54	394.50	46.31
56	544.50	47.70
58	694.50	48.75
60	758.82	49.11
62	855.29	49.56
64	1000.00	50.00

NODES OF FLEXIBLE Z-COORDINATES

8	9	14	15	20	24	28	31	34	36
38	40	42	44						

ELEMENTS OF VARIABLE SHAPE ARE NUMBER 30 TO 75

PRESCRIBED NODES AND HEAD VALUES

NODE	PRESCRIBED VALUES
7	20.203
8	30.305
63	50.000
64	50.000

LENGTH OF GROSS VECTOR # 292

```
****************************************
*   WATER LEVEL IN THE WELL #  10.00   *
****************************************
```

NUMBER OF ITERATIONS REQUIRED # 10

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.1062

NODE	R-COORDINATE	Z-COORDINATE	HEAD VALUE	RHO-COORD	TZI-COORD
1	1.00	0.0	43.5719	0.0010	0.0
2	1.00	3.37	43.2851	0.0010	0.0673
3	1.00	6.73	42.9936	0.0010	0.1347
4	1.00	10.10	41.9834	0.0010	0.2020
5	1.00	13.47	40.9584	0.0010	0.2694
6	1.00	16.84	35.0067	0.0010	0.3367
7	1.00	20.20	20.2845	0.0010	0.4041
8	1.00	30.22	30.2608	0.0010	0.6044
9	1.00	40.24	40.2376	0.0010	0.8048
10	1.30	0.0	43.5742	0.0013	0.0
11	1.30	6.73	42.9962	0.0013	0.1347
12	1.30	13.47	40.9707	0.0013	0.2694
13	1.30	20.20	29.0314	0.0013	0.4041
14	1.30	30.22	33.9845	0.0013	0.6045
15	1.30	40.24	40.2440	0.0013	0.8049
16	1.75	0.0	43.5795	0.0017	0.0
17	1.75	6.73	43.0017	0.0017	0.1347
18	1.75	13.47	40.9963	0.0017	0.2694
19	1.75	20.20	33.7204	0.0017	0.4041
20	1.75	40.30	40.3043	0.0017	0.8061
21	2.42	0.0	43.5943	0.0024	0.0
22	2.42	10.10	42.5614	0.0024	0.2020
23	2.42	20.20	37.5582	0.0024	0.4041
24	2.42	41.30	41.3044	0.0024	0.8261
25	3.44	0.0	43.5832	0.0034	0.0
26	3.44	10.10	42.6496	0.0034	0.2020
27	3.44	20.20	37.8709	0.0034	0.4041
28	3.44	42.07	42.0747	0.0034	0.8415
29	4.96	0.0	43.6036	0.0050	0.0
30	4.96	20.20	41.8672	0.0050	0.4041
31	4.96	42.67	42.6655	0.0050	0.8533
32	7.23	0.0	43.6978	0.0072	0.0
33	7.23	20.20	42.9284	0.0072	0.4041
34	7.23	43.16	43.1763	0.0072	0.8635
35	10.65	0.0	43.8251	0.0107	0.0
36	10.65	43.72	43.7152	0.0107	0.8743
37	15.78	0.0	44.3670	0.0158	0.0
38	15.78	44.32	44.3168	0.0158	0.8863
39	23.47	0.0	44.9191	0.0235	0.0
40	23.47	44.91	44.9064	0.0235	0.3981
41	35.00	0.0	45.4781	0.0350	0.0
42	35.00	45.48	45.4835	0.0350	0.9097
43	52.30	0.0	46.0399	0.0523	0.0
44	52.30	46.05	46.0494	0.0523	0.9210
45	78.25	0.0	46.6001	0.0782	0.0
46	78.25	46.61	46.6071	0.0782	0.9321
47	117.17	0.0	47.1552	0.1172	0.0
48	117.17	47.16	47.1597	0.1172	0.9432
49	175.56	0.0	47.7038	0.1756	0.0
50	175.56	47.71	47.7054	0.1756	0.9541
51	263.14	0.0	48.2457	0.2631	0.0
52	263.14	48.25	48.2464	0.2631	0.9649
53	394.50	0.0	48.7817	0.3945	0.0
54	394.50	48.76	48.7817	0.3945	0.9756
55	544.50	0.0	49.2053	0.5445	0.0
56	544.50	49.21	49.2053	0.5445	0.9841
57	694.50	0.0	49.5249	0.6945	0.0
58	694.50	49.52	49.5245	0.6945	0.9905
59	758.82	0.0	49.6407	0.7588	0.0
60	758.82	49.64	49.6408	0.7588	0.9928
61	855.29	0.0	49.7969	0.8553	0.0
62	855.29	49.80	49.7970	0.8553	0.9959
63	1000.00	0.0	50.0000	1.0000	0.0
64	1000.00	50.00	50.0000	1.0000	1.0000

FREE-SURFACE-POSITON-

RADIUS	SURFACE HEIGHT
1.00	40.2376
1.30	40.2440
1.75	40.3043
2.42	41.3044
3.44	42.0747
4.96	42.6655
7.23	43.1763
10.65	43.7152
15.78	44.3160
23.47	44.9064
35.00	45.4635
52.30	46.0494
78.25	46.6071
117.17	47.1537
175.56	47.7054
263.14	48.2464
394.50	48.7817
544.50	49.2058
694.50	49.5245
758.32	49.6408
855.29	49.7970
1000.00	50.0000

SURFACE TOLERANCE COUNTER # 0

MAXIMUM ERROR OF SURFACE HEIGHT # 0.1062

```
*************************************
*         ELEMENT VELOCITIES        *
*************************************
```

ELEM NO.	RADIAL VEL	VERTIC VEL	IDARCY
1	-0.006	0.068	0
2	-0.006	0.211	1
3	-0.017	0.723	1
4	-0.000	0.069	0
5	-0.000	0.211	1
6	0.008	0.724	1
7	-0.007	0.069	0
8	-0.029	0.213	1
9	-3.553	0.534	1
10	-0.009	0.069	0
11	-0.009	0.211	1
12	-0.023	0.724	1
13	-0.010	0.069	0
14	-0.040	0.203	1
15	-2.041	0.212	1
16	-0.018	0.069	0
17	-0.434	0.155	1
18	-1.444	0.274	1
19	0.112	0.030	1
20	0.093	0.303	1
21	0.009	0.032	0
22	-0.054	0.306	1
23	-0.070	0.074	0
24	-0.845	0.102	1
25	-0.011	0.074	0
26	-0.045	0.069	0
27	-0.594	0.124	1
28	-0.033	0.069	0
29	-0.293	0.024	1
30	-0.030	0.030	0
31	-0.125	0.009	1
32	-0.182	0.002	1
33	-0.085	0.002	0
34	-0.094	0.001	0
35	-0.057	0.001	0
36	-0.061	0.000	0
37	-0.039	0.000	0
38	-0.040	-0.000	0
39	-0.026	-0.000	0
40	-0.026	-0.000	0
41	-0.017	-0.000	0
42	-0.017	-0.000	0
43	-0.011	-0.000	0
44	-0.011	-0.000	0
45	-0.008	-0.000	0
46	-0.007	-0.000	0
47	-0.005	-0.000	0
48	-0.005	-0.000	0
49	-0.003	-0.000	0
50	-0.003	0.000	0
51	-0.002	0.000	0
52	-0.002	0.000	0
53	-0.002	0.000	0
54	-0.002	0.000	0
55	-0.001	0.000	0
56	-0.001	-0.000	0
57	-0.001	-0.000	0
58	-0.001	-0.000	0
59	-0.001	-0.000	0
60	-0.001	-0.0	0
61	-3.575	-0.123	1
62	-2.249	-0.181	1
63	-2.252	-0.090	1
64	-0.004	-0.363	1
65	-2.045	-0.097	1
66	-0.023	-0.363	1
67	-1.597	-0.079	1
68	-1.453	-0.084	1
69	-0.567	-0.093	1
70	-0.947	-0.066	1
71	-0.386	-0.057	1
72	-0.378	-0.046	1
73	-0.250	-0.024	1
74	-0.293	-0.022	1
75	-0.166	-0.008	1

TOTAL DISCHARGE INTO THE WELL # 327.1267

```
**************************************************
*          STEADY STATE TYPE CURVE               *
**************************************************
```

LAMBDA # 0.0017

NODE NUMBER	R-COORDINATE	FUNCTION W(U)	ARGUMENT U
1	1.00	0.4623E 01	0.1000E-02
2	1.00	0.4814E 01	0.1000E-02
3	1.00	0.5008E 01	0.1000E-02
4	1.00	0.5668E 01	0.1000E-02
5	1.00	0.6321E 01	0.1000E-02
6	1.00	0.9726E 01	0.1000E-02
7	1.00	0.1605E 02	0.1000E-02
8	1.00	0.1218E 02	0.1000E-02
9	1.00	0.6771E 01	0.1000E-02
10	1.30	0.4622E 01	0.1300E-02
11	1.30	0.5006E 01	0.1300E-02
12	1.30	0.6313E 01	0.1300E-02
13	1.30	0.1274E 02	0.1300E-02
14	1.30	0.1034E 02	0.1300E-02
15	1.30	0.6767E 01	0.1300E-02
16	1.75	0.4618E 01	0.1750E-02
17	1.75	0.5002E 01	0.1750E-02
18	1.75	0.6297E 01	0.1750E-02
19	1.75	0.1048E 02	0.1750E-02
20	1.75	0.6730E 01	0.1750E-02
21	2.42	0.4608E 01	0.2425E-02
22	2.42	0.5292E 01	0.2425E-02
23	2.42	0.6373E 01	0.2425E-02
24	2.42	0.6102E 01	0.2425E-02
25	3.44	0.4615E 01	0.3437E-02
26	3.44	0.5234E 01	0.3437E-02
27	3.44	0.6997E 01	0.3437E-02
28	3.44	0.5609E 01	0.3437E-02
29	4.96	0.4602E 01	0.4956E-02
30	4.96	0.5742E 01	0.4956E-02
31	4.96	0.5224E 01	0.4956E-02
32	7.23	0.4539E 01	0.7234E-02
33	7.23	0.5051E 01	0.7234E-02
34	7.23	0.4887E 01	0.7234E-02
35	10.65	0.4453E 01	0.1065E-01
36	10.65	0.4527E 01	0.1065E-01
37	15.78	0.4066E 01	0.1578E-01
38	15.78	0.4120E 01	0.1578E-01
39	23.47	0.3707E 01	0.2347E-01
40	23.47	0.3716E 01	0.2347E-01
41	35.00	0.3318E 01	0.3500E-01
42	35.00	0.3315E 01	0.3500E-01
43	52.30	0.2923E 01	0.5230E-01
44	52.30	0.2916E 01	0.5230E-01
45	78.25	0.2524E 01	0.7825E-01
46	78.25	0.2519E 01	0.7825E-01
47	117.17	0.2124E 01	0.1172E 00
48	117.17	0.2122E 01	0.1172E 00
49	175.56	0.1724E 01	0.1756E 00
50	175.56	0.1723E 01	0.1756E 00
51	263.14	0.1325E 01	0.2631E 00
52	263.14	0.1324E 01	0.2631E 00
53	394.50	0.9250E 00	0.3945E 00
54	394.50	0.9250E 00	0.3945E 00
55	544.50	0.6055E 00	0.5445E 00
56	544.50	0.6056E 00	0.5445E 00
57	694.50	0.3634E 00	0.6945E 00
58	694.50	0.3637E 00	0.6945E 00
59	758.82	0.2752E 00	0.7588E 00
60	758.82	0.2751E 00	0.7588E 00
61	855.29	0.1559E 00	0.8553E 00
62	855.29	0.1557E 00	0.8553E 00
63	1000.00	0.0	0.1000E 01
64	1000.00	0.0	0.1000E 01

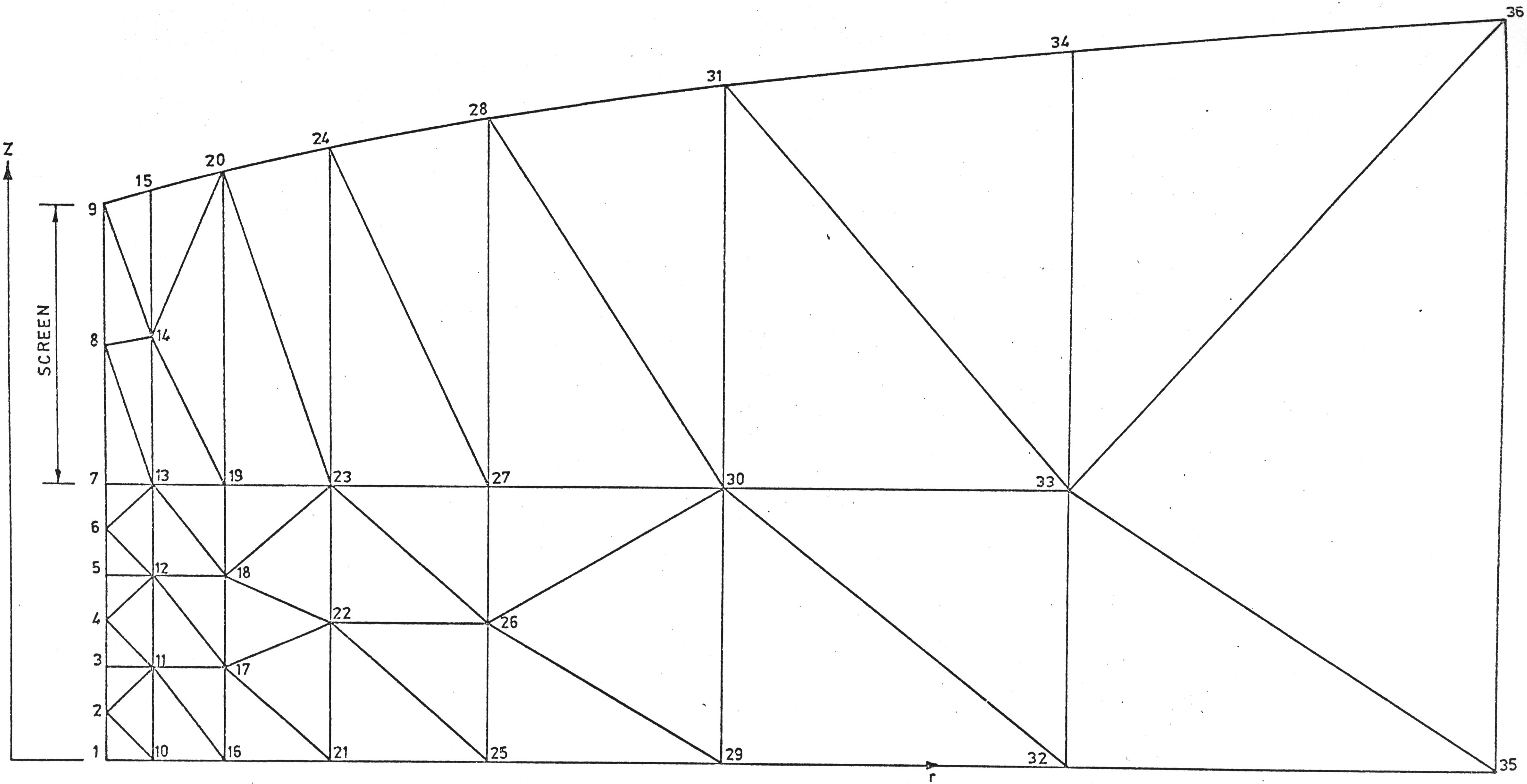


FIGURE 24: MESH GENERATED FOR SAMPLE PROBLEMS

3.9 TRFREE - PROGRAM FOR SOLVING TRANSIENT, FREE SURFACE FLOW USING TRIANGULAR ELEMENTS

The program TRFREE is used to analyse transient, free surface flow in an unconfined aquifer. Unlike TRCON1, which considers only radial flow components and assumes constant saturated thickness of the unconfined aquifer, this program performs a two-dimensional analysis and considers the presence of the free surface and/or the seepage face. It can handle fully and partially penetrating wells with or without gravel packs. For a partially penetrating well, the portion of aquifer directly below the base of the well is neglected. Up to 5 screened intervals may be used.

Figures 25 and 26 show the input data form and completed form for a sample run. One problem was solved, with IVEL = 0 and IDISCR = 1. The problem involves transient, two-regime flow towards a well which is fully screened through the initial saturated thickness of the aquifer. Numerical solutions were obtained for 10 time steps. All values of the hydraulic head are measured from the base of the aquifer. A portion of the printed output is shown on pages 110 to 120. It consists of the following:

(i) a reformatted playback of input data;

(ii) the generated discretisation data, which consists of connections of elements, nodal co-ordinates, a list of top boundary nodes and their radial co-ordinates and elements of variable shape and the length of the gross vector;

(iii) tables of results for time values at the mid-point and the end of the first time step; (Included in these tables are the result of iteration for correct well discharge, radial co-ordinates and heights of the free surface nodes, a list of all the node numbers, radial and vertical co-ordinates, values of the dimensionless time $1/u = 4Kt/r^2S_s$ and values of the well function $W(u) = 4\pi K h_o s/Q$, where $s = h_o - h$)

(iv) tables of results for time values at the mid-point and the end of time step number 10.

A number of punched cards were also produced by the program. The information contained in these cards is nodal values of the hydraulic head and flux at the end of the final time step. As in TRCON1 and TRCON3, the punched cards may be added to the input data deck to rerun the program from time step number 10 onwards.

FIGURE 25: INPUT DATA FORM FOR TRFREE

COMPUTER DATA SHEET	PROGRAM TRFREE Transient, free surface flow	By .... Date ....

Specification Card

NPROB	IVEL	IDISCR	ORELAX

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL

(5F10.2)

QFIX	RCSNG	QRTOL

(3F10.2)

AK	BK	VCR	SS	SY	DINDEX

(6E10.2)

IGP	IBOUND	IWBC	NSTEP

(4I10)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

NTICR	ITST	TFACTR	TMUL	DTMUL

(2I10, 3F10.2)

FRLEN	SCFAC	XLMAX	IREG	NMIN

(3F10.2, 2I10)

NDSC	NSCREN

(2I10)

XSCR	HSCR

(2F10.2)

FIGURE 26: COMPLETED FORM FOR SAMPLE PROBLEM

COMPUTER DATA SHEET	PROGRAM TRFREE Transient, free surface flow	By P. Huyakorn Date 14/12/73

Specification Card

NPROB	IVEL	IDISCR	ORELAX
1	0	1	1.00

(3I10, F10.2)

Problem Data Cards

RW	RO	HO	TH	HTOL
0.50	1000.00	50.00	50.00	0.20

(5F10.2)

QFIX	RCSNG	QRTOL
100.00	0.50	0.01

(3F10.2)

AK	BK	VCR	SS	SY	DINDEX
0.10E02	0.20E02	0.60E-01	0.10E-04	0.20E00	0.00E00

(6E10.2)

IGP	IBOUND	IWBC	NSTEP
0	0	0	2

(4I10)

AGP	BGP	VGP	THGP	BTGP

(Omit this card if IGP = 0)

(5F10.3)

NTICR	ITST	TFACTR	TMUL	DTMUL
10	1	100.00	2.00	0.00

(2I10, 3F10.2)

FRLEN	SCFAC	XLMAX	IREG	NMIN
0.30	1.50	150.00	2	2

(3F10.2, 2I10)

NDSC	NSCREN
11	1

(2I10)

XSCR	HSCR
0.00	50.00

(2F10.2)

****
* FINITE ELEMENT SOLUTION OF *
* TRANSIENT,TWO-REGIME,FREE SURFACE FLOW. *
****

****
* PROBLEM NUMBER # 1 *
****

GENERAL INPUT DATA

RADIUS OF WELL # 0.50
EXTERNAL RADIUS # 1000.00
INITIAL HEIGHT OF WATER TABLE # 50.00
INITIAL SATURATED THICKNESS OF AQUIFER # 50.00
OVER-RELAXATION FACTOR # 1.00
PRESCRIBED HEAD TOLERANCE # 0.20
PRESCRIBED WELL DISCHARGE # 100.00
DISCHARGE TOLERANCE RATIO # 0.010

AQUIFER PROPERTIES

FORCHHEIMER COEFF. A # 10.0000
FORCHHEIMER COEFF. B # 20.0000
CRITICAL FLOW VELOCITY # 0.0600
SPECIFIC STORAGE # 0.100E-04
SPECIFIC YIELD # 0.200E 00
RECIPROCAL OF DELAYED INDEX # 0.0
GRAVEL PACK INDEX # 0
BOUNDARY INDEX # 0
WELL B.C. INDEX # 1
ADOPT 2 STEP FREE SURFACE ITERATIVE SCHEME

SCREEN NO.	BASE HEIGHT	LENGTH
1	0.0	50.00

DISCRETISATION DATA

IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS

ELEM NO	NODE1	NODE2	NODE3	REPETITION NUMBER
1	1	16	2	1
2	3	17	4	1
3	5	18	6	1
4	7	19	8	1
5	9	20	10	1
6	2	16	17	2
7	4	17	18	2
8	6	18	19	2
9	8	19	20	2
10	10	20	21	2
11	2	17	3	3
12	4	18	5	3
13	6	19	7	3
14	8	20	9	3
15	10	21	11	3
16	16	26	17	4
17	17	27	18	4
18	18	28	19	4
19	19	29	20	4
20	20	30	21	4
21	17	26	27	5
22	18	27	28	5
23	19	28	29	5
24	20	29	30	5
25	21	30	31	5
26	26	35	27	6
27	27	36	28	7
28	28	37	29	8
29	29	38	30	9
30	30	39	31	10
31	27	35	36	11

32	28	36	37	12
33	29	37	38	13
34	30	38	39	14
35	35	43	36	15
36	36	44	37	15
37	37	45	38	15
38	38	46	39	15
39	36	43	44	16
40	37	44	45	16
41	38	45	46	16
42	39	46	47	16
43	43	50	44	17
44	45	51	46	17
45	44	50	51	18
46	46	51	52	18
47	44	51	45	19
48	46	52	47	19
49	50	55	51	20
50	51	56	52	20
51	51	55	56	21
52	52	56	57	21
53	55	59	56	22
54	56	59	60	23
55	56	60	57	24
56	59	62	60	25
57	60	62	63	26
58	62	65	63	27
59	63	66	64	28
60	63	65	66	29
61	65	67	66	30
62	66	67	68	31
63	67	69	68	32
64	68	69	70	33
65	69	71	70	34
66	70	71	72	35
67	71	73	72	36
68	72	73	74	37
69	73	75	74	38
70	74	75	76	39
71	75	77	76	40
72	76	77	78	41
73	77	79	78	42
74	78	79	80	43
75	79	81	80	44
76	80	81	82	45
77	81	83	82	46
78	82	83	84	47
79	83	85	84	48
80	84	85	86	49
81	85	87	86	50
82	86	87	88	51
83	87	89	88	52
84	88	89	90	53
85	11	21	12	54
86	12	22	13	55
87	13	23	14	56
88	14	24	15	57
89	12	21	22	58
90	13	22	23	59
91	14	23	24	60
92	15	24	25	61
93	21	31	22	62
94	22	32	23	63
95	23	33	24	64
96	24	34	25	65
97	22	31	32	66
98	23	32	33	67
99	24	33	34	68
100	31	39	32	69
101	32	40	33	70
102	33	41	34	71
103	32	39	40	72
104	33	40	41	73
105	34	41	42	74
106	39	47	40	75
107	40	48	41	76
108	41	49	42	77
109	40	47	48	78
110	41	48	49	79
111	47	52	48	80
112	48	53	49	81
113	48	52	53	82
114	49	53	54	83
115	52	57	53	84
116	53	58	54	85
117	53	57	58	86
118	57	60	58	87
119	58	60	61	88
120	60	63	61	89
121	61	63	64	90

NODAL COORDINATES

NODE	R-COORD	Z-COORD
1	0.50	0.0
2	0.50	2.00
3	0.50	4.00
4	0.50	6.00
5	0.50	8.00
6	0.50	10.00
7	0.50	12.00
8	0.50	14.00
9	0.50	16.00
10	0.50	18.00
11	0.50	20.00
12	0.50	27.49
13	0.50	34.99
14	0.50	42.49
15	0.50	49.99
16	0.80	0.0
17	0.80	4.00
18	0.80	8.00
19	0.80	12.00
20	0.80	16.00
21	0.80	20.00
22	0.80	27.50
23	0.80	35.00
24	0.80	42.50
25	0.80	50.00
26	1.25	0.0
27	1.25	4.00
28	1.25	8.00
29	1.25	12.00
30	1.25	16.00
31	1.25	20.00
32	1.25	30.00
33	1.25	40.00
34	1.25	50.00
35	1.92	0.0
36	1.92	5.00
37	1.92	10.00
38	1.92	15.00
39	1.92	20.00
40	1.92	30.00
41	1.92	40.00
42	1.92	50.00
43	2.94	0.0
44	2.94	5.00
45	2.94	10.00
46	2.94	15.00
47	2.94	20.00
48	2.94	35.00
49	2.94	50.00
50	4.46	0.0
51	4.46	10.00
52	4.46	20.00
53	4.46	35.00
54	4.46	50.00
55	6.73	0.0
56	6.73	10.00
57	6.73	20.00
58	6.73	50.00
59	10.15	0.0
60	10.15	20.00
61	10.15	50.00
62	15.28	0.0
63	15.28	20.00
64	15.28	50.00
65	22.97	0.0
66	22.97	50.00
67	34.50	0.0
68	34.50	50.00
69	51.80	0.0
70	51.80	50.00
71	77.75	0.0
72	77.75	50.00
73	116.67	0.0
74	116.67	50.00
75	175.06	0.0
76	175.06	50.00
77	262.64	0.0
78	262.64	50.00
79	394.00	0.0
80	394.00	50.00
81	544.00	0.0
82	544.00	50.00
83	694.00	0.0
84	694.00	50.00
85	758.42	0.0
86	758.42	50.00
87	855.05	0.0
88	855.05	50.00
89	1000.00	0.0
90	1000.00	50.00

TOP BOUNDARY NODES AND RADIAL COORDINATES.

NODE NUMBER	R-COORDINATE	SURFACE HEIGHT
15	0.50	49.99
25	0.80	50.00
34	1.25	50.00
42	1.92	50.00
49	2.94	50.00
54	4.46	50.00
58	6.73	50.00
61	10.15	50.00
64	15.28	50.00
66	22.97	50.00
68	34.50	50.00
70	51.80	50.00
72	77.75	50.00
74	116.67	50.00
76	175.06	50.00
78	262.64	50.00
80	394.00	50.00
82	544.00	50.00
84	694.00	50.00
86	758.42	50.00
88	855.05	50.00
90	1000.00	50.00

NODES OF FLEXIBLE Z-COORDINATES

12	13	14	15	22	23	24	25	32	33
34	40	41	42	48	49	53	54	58	61
64	66	68	70						

ELEMENTS OF VARIABLE SHAPE ARE NUMBER 58 TO 121

LENGTH OF GROSS VECTOR # 654

```
**********************************
*   TIME STEP NUMBER  #    1     *
**********************************


****************************************
*         TIME  #     0.314E 00        *
****************************************
```

ESTIMATED RADIUS OF INFLUENCE # 1000.00

CORRESPONDING NO. OF NODES # 90

CORRESPONDING NO. OF ELEMENTS # 121

CORRESPONDING COMPONENT OF VECTOR NDVEC # 22

- -

DISCHARGE ITERATION NUMBER # 1

HEAD INCREMENT # -39.240

HEAD VALUE # 10.760

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.0326

DISCHARGE FROM AQUIFER INTO WELL # 136.3077

DISCHARGE FROM WELL STORAGE # 98.100

TOTAL CALCULATED DISCHARGE # 234.408

RESIDUAL DISCHARGE # 134.4076

- -

DISCHARGE ITERATION NUMBER # 2

HEAD INCREMENT # -11.492

HEAD VALUE # 38.508

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.0629

DISCHARGE FROM AQUIFER INTO WELL # 70.8378

DISCHARGE FROM WELL STORAGE # 28.730

TOTAL CALCULATED DISCHARGE # 99.568

RESIDUAL DISCHARGE # 0.4325

FREE SURFACE POSITON

RADIUS	SURFACE HEIGHT
0.50	49.7747
0.80	49.7562
1.25	49.6002
1.92	49.8202
2.94	49.7673
4.46	49.9532
6.73	49.8456
10.15	49.9867
15.28	49.9958
22.97	49.9780
34.50	49.9999
51.80	49.9986
77.75	49.9996
116.67	49.9999
175.06	50.0001
262.64	49.9998
394.00	50.0000
544.00	50.0000
694.00	50.0000
758.42	49.9999
855.05	50.0000
1000.00	50.0000

SURFACE TOLERANCE COUNTER # 0

MAXIMUM ERROR OF SURFACE HEIGHT # 0.0629

```
**********************************************
*                                            *
*          FINAL RESULTS OF ANALYSIS         *
*                                            *
**********************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	Z-COORD	HEAD	1/U	W(U)
1	0.50	0.0	38.5081	0.4490E 05	6.4495
2	0.50	2.00	38.5081	0.4490E 05	6.4495
3	0.50	4.00	38.5081	0.4490E 05	6.4495
4	0.50	6.00	38.5081	0.4490E 05	6.4495
5	0.50	8.00	38.5081	0.4490E 05	6.4495
6	0.50	10.00	38.5081	0.4490E 05	6.4495
7	0.50	12.00	38.5081	0.4490E 05	6.4495
8	0.50	14.00	38.5081	0.4490E 05	6.4495
9	0.50	16.00	38.5081	0.4490E 05	6.4495
10	0.50	18.00	38.5081	0.4490E 05	6.4495
11	0.50	20.00	38.5081	0.4490E 05	6.4495
12	0.50	27.44	38.5081	0.4490E 05	6.4495
13	0.50	34.89	38.5081	0.4490E 05	6.4495
14	0.50	42.33	42.3109	0.4490E 05	4.3153
15	0.50	49.77	49.7747	0.4490E 05	0.1265
16	0.80	0.0	39.9252	0.1754E 05	5.6542
17	0.80	4.00	40.1303	0.1754E 05	5.5391
18	0.80	8.00	40.1276	0.1754E 05	5.5406
19	0.80	12.00	40.1516	0.1754E 05	5.5272
20	0.80	16.00	40.1883	0.1754E 05	5.5065
21	0.80	20.00	40.1218	0.1754E 05	5.5439
22	0.80	27.44	40.4061	0.1754E 05	5.3843
23	0.80	34.88	40.9639	0.1754E 05	5.0713
24	0.80	42.32	44.0891	0.1754E 05	3.3173
25	0.80	49.76	49.7562	0.1754E 05	0.1368
26	1.25	0.0	41.3020	0.7184E 04	4.8815
27	1.25	4.00	41.4919	0.7184E 04	4.7749
28	1.25	8.00	41.5108	0.7184E 04	4.7644
29	1.25	12.00	41.5530	0.7184E 04	4.7406
30	1.25	16.00	41.6084	0.7184E 04	4.7096
31	1.25	20.00	41.4543	0.7184E 04	4.7961
32	1.25	29.86	42.1408	0.7184E 04	4.4108
33	1.25	39.73	44.2839	0.7184E 04	3.2080
34	1.25	49.60	49.6002	0.7184E 04	0.2244
35	1.92	0.0	42.5122	0.3029E 04	4.2023
36	1.92	5.00	42.7073	0.3029E 04	4.0928
37	1.92	10.00	42.7208	0.3029E 04	4.0853
38	1.92	15.00	42.8522	0.3029E 04	4.0115
39	1.92	20.00	42.7725	0.3029E 04	4.0562
40	1.92	29.94	43.7014	0.3029E 04	3.5349
41	1.92	39.88	45.7894	0.3029E 04	2.3631
42	1.92	49.82	49.8202	0.3029E 04	0.1009
43	2.94	0.0	43.6362	0.1301E 04	3.5715
44	2.94	5.00	43.8243	0.1301E 04	3.4660
45	2.94	10.00	43.8362	0.1301E 04	3.4593
46	2.94	15.00	44.0082	0.1301E 04	3.3628
47	2.94	20.00	43.9060	0.1301E 04	3.4201
48	2.94	34.88	45.7562	0.1301E 04	2.3817
49	2.94	49.77	49.7673	0.1301E 04	0.1306
50	4.46	0.0	44.6578	0.5652E 03	2.9982
51	4.46	10.00	44.9528	0.5652E 03	2.8326
52	4.46	20.00	45.1039	0.5652E 03	2.7478
53	4.46	34.97	46.9660	0.5652E 03	1.7028
54	4.46	49.95	49.9532	0.5652E 03	0.0262
55	6.73	0.0	45.6871	0.2475E 03	2.4205
56	6.73	10.00	45.9857	0.2475E 03	2.2529
57	6.73	20.00	46.1582	0.2475E 03	2.1561
58	6.73	49.85	49.8456	0.2475E 03	0.0867
59	10.15	0.0	46.6458	0.1089E 03	1.8824
60	10.15	20.00	47.2423	0.1089E 03	1.5477
61	10.15	49.99	49.9867	0.1089E 03	0.0075
62	15.28	0.0	47.5932	0.4809E 02	1.3508
63	15.28	20.00	48.2005	0.4809E 02	1.0099
64	15.28	50.00	49.9958	0.4809E 02	0.0023
65	22.97	0.0	48.3712	0.2128E 02	0.9141
66	22.97	49.98	49.9780	0.2128E 02	0.0124
67	34.50	0.0	49.0781	0.9431E 01	0.5174
68	34.50	50.00	49.9999	0.9431E 01	0.0000
69	51.80	0.0	49.5751	0.4183E 01	0.2385
70	51.80	50.00	49.9986	0.4183E 01	0.0008

71	77.75	0.0	49.8553	0.1857E 01	0.0812
72	77.75	50.00	49.9996	0.1857E 01	0.0002
73	116.67	0.0	49.9682	0.8246E 00	0.0179
74	116.67	50.00	49.9999	0.8246E 00	0.0001
75	175.06	0.0	49.9962	0.3663E 00	0.0021
76	175.06	50.00	50.0001	0.3663E 00	-0.0001
77	262.64	0.0	49.9996	0.1627E 00	0.0002
78	262.64	50.00	49.9998	0.1627E 00	0.0001
79	394.00	0.0	50.0000	0.7230E-01	0.0000
80	394.00	50.00	50.0000	0.7230E-01	0.0000
81	544.00	0.0	49.9999	0.3793E-01	0.0000
82	544.00	50.00	50.0000	0.3793E-01	0.0000
83	694.00	0.0	49.9999	0.2330E-01	0.0001
84	694.00	50.00	50.0000	0.2330E-01	0.0000
85	758.42	0.0	49.9999	0.1951E-01	0.0000
86	758.42	50.00	49.9999	0.1951E-01	0.0000
87	855.05	0.0	50.0000	0.1535E-01	0.0
88	855.05	50.00	50.0000	0.1535E-01	0.0000
89	1000.00	0.0	50.0000	0.1122E-01	0.0
90	1000.00	50.00	50.0000	0.1122E-01	0.0000

```
****************************************
*          TIME   #     0.629E 00      *
****************************************
```

```
DISCHARGE ITERATION NUMBER   #          1
    HEAD INCREMENT    #    -11.665
    HEAD VALUE    #    26.843

NUMBER OF ITERATIONS REQUIRED #         2

TOLERANCE COUNTER FOR HEAD    #  0
ABSOLUTE MAXIMUM ERROR IN HEAD    #       0.0354

DISCHARGE FROM AQUIFER INTO WELL   #    107.1969
DISCHARGE FROM WELL STORAGE #        29.162
TOTAL CALCULATED DISCHARGE    #      136.359
RESIDUAL DISCHARGE          #    36.3591
- - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER   #          2
    HEAD INCREMENT    #    -3.939
    HEAD VALUE    #    34.569

NUMBER OF ITERATIONS REQUIRED #         2

TOLERANCE COUNTER FOR HEAD    #  0
ABSOLUTE MAXIMUM ERROR IN HEAD    #       0.0178

DISCHARGE FROM AQUIFER INTO WELL   #     84.4069
DISCHARGE FROM WELL STORAGE #         9.847
TOTAL CALCULATED DISCHARGE    #       94.254
RESIDUAL DISCHARGE          #     5.7459
- - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER   #          3
    HEAD INCREMENT    #    -4.993
    HEAD VALUE    #    33.515

NUMBER OF ITERATIONS REQUIRED #         2

TOLERANCE COUNTER FOR HEAD    #  0
ABSOLUTE MAXIMUM ERROR IN HEAD    #       0.0165

DISCHARGE FROM AQUIFER INTO WELL   #     87.7618
DISCHARGE FROM WELL STORAGE #        12.483
TOTAL CALCULATED DISCHARGE    #      100.245
RESIDUAL DISCHARGE          #     0.2448
```

FREE SURFACE POSITON

RADIUS	SURFACE HEIGHT
0.50	49.5781
0.80	49.5432
1.25	49.3728
1.92	49.6142
2.94	49.5901
4.46	49.8507
6.73	49.7017
10.15	49.9534
15.28	49.9888
22.97	49.9490
34.50	49.9985
51.80	49.9966
77.75	49.9990
116.67	49.9996
175.06	50.0002
262.64	49.9996
394.00	49.9999
544.00	49.9999
694.00	49.9999
758.42	49.9999
855.05	49.9999
1000.00	50.0000

SURFACE TOLERANCE COUNTER # 0

MAXIMUM ERROR OF SURFACE HEIGHT # 0.0165

FINAL RESULTS OF ANALYSIS

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	Z-COORD	HEAD	1/U	W(U)
1	0.50	0.0	33.5149	0.8980E 05	9.2518
2	0.50	2.00	33.5149	0.8980E 05	9.2518
3	0.50	4.00	33.5149	0.8980E 05	9.2518
4	0.50	6.00	33.5149	0.8980E 05	9.2518
5	0.50	8.00	33.5149	0.8980E 05	9.2518
6	0.50	10.00	33.5149	0.8980E 05	9.2518
7	0.50	12.00	33.5149	0.8980E 05	9.2518
8	0.50	14.00	33.5149	0.8980E 05	9.2518
9	0.50	16.00	33.5149	0.8980E 05	9.2518
10	0.50	18.00	33.5149	0.8980E 05	9.2518
11	0.50	20.00	33.5149	0.8980E 05	9.2518
12	0.50	27.39	33.5149	0.8980E 05	9.2518
13	0.50	34.79	34.7937	0.8980E 05	8.5341
14	0.50	42.18	42.1864	0.8980E 05	4.3852
15	0.50	49.58	49.5781	0.8980E 05	0.2368
16	0.80	0.0	35.7847	0.3508E 05	7.9780
17	0.80	4.00	36.1219	0.3508E 05	7.7887
18	0.80	8.00	36.1243	0.3508E 05	7.7874
19	0.80	12.00	36.1710	0.3508E 05	7.7612
20	0.80	16.00	36.2411	0.3508E 05	7.7218
21	0.80	20.00	36.1539	0.3508E 05	7.7707
22	0.80	27.38	36.6715	0.3508E 05	7.4803
23	0.80	34.77	38.3645	0.3508E 05	6.5301
24	0.80	42.16	43.7656	0.3508E 05	3.4989
25	0.80	49.54	49.5432	0.3508E 05	0.2564
26	1.25	0.0	37.8752	0.1437E 05	6.8047
27	1.25	4.00	38.1725	0.1437E 05	6.6379
28	1.25	8.00	38.2109	0.1437E 05	6.6163
29	1.25	12.00	38.2905	0.1437E 05	6.5716
30	1.25	16.00	38.3970	0.1437E 05	6.5119
31	1.25	20.00	38.1819	0.1437E 05	6.6326
32	1.25	29.79	39.4656	0.1437E 05	5.9121
33	1.25	39.58	43.3894	0.1437E 05	3.7100
34	1.25	49.37	49.3728	0.1437E 05	0.3520
35	1.92	0.0	39.6308	0.6058E 04	5.8195
36	1.92	5.00	39.9263	0.6058E 04	5.6536
37	1.92	10.00	39.9632	0.6058E 04	5.6329
38	1.92	15.00	40.1916	0.6058E 04	5.5047
39	1.92	20.00	40.1244	0.6058E 04	5.5424
40	1.92	29.87	41.6602	0.6058E 04	4.6805
41	1.92	39.74	44.9823	0.6058E 04	2.8161
42	1.92	49.61	49.6142	0.6058E 04	0.2165
43	2.94	0.0	41.2147	0.2602E 04	4.9305
44	2.94	5.00	41.4947	0.2602E 04	4.7734
45	2.94	10.00	41.5296	0.2602E 04	4.7538
46	2.94	15.00	41.8215	0.2602E 04	4.5900
47	2.94	20.00	41.7368	0.2602E 04	4.6375
48	2.94	34.79	44.5901	0.2602E 04	3.0362
49	2.94	49.59	49.5901	0.2602E 04	0.2300
50	4.46	0.0	42.6273	0.1130E 04	4.1377
51	4.46	10.00	43.0732	0.1130E 04	3.8875
52	4.46	20.00	43.3964	0.1130E 04	3.7061
53	4.46	34.92	46.0367	0.1130E 04	2.2243
54	4.46	49.85	49.8507	0.1130E 04	0.0838
55	6.73	0.0	44.0112	0.4950E 03	3.3611
56	6.73	10.00	44.4628	0.4950E 03	3.1076
57	6.73	20.00	44.8163	0.4950E 03	2.9092
58	6.73	49.70	49.7017	0.4950E 03	0.1674
59	10.15	0.0	45.3023	0.2178E 03	2.6365
60	10.15	20.00	46.2165	0.2178E 03	2.1234

61	10.15	49.95	49.9534	0.2178E 03	0.0262
62	15.28	0.0	46.5869	0.9618E 02	1.9155
63	15.28	20.00	47.4797	0.9618E 02	1.4144
64	15.28	49.99	49.9888	0.9618E 02	0.0063
65	22.97	0.0	47.6562	0.4256E 02	1.3154
66	22.97	49.95	49.9490	0.4256E 02	0.0286
67	34.50	0.0	48.6342	0.1886E 02	0.7665
68	34.50	50.00	49.9985	0.1886E 02	0.0008
69	51.80	0.0	49.3433	0.8367E 01	0.3686
70	51.80	50.00	49.9966	0.8367E 01	0.0019
71	77.75	0.0	49.7625	0.3714E 01	0.1333
72	77.75	50.00	49.9990	0.3714E 01	0.0006
73	116.67	0.0	49.9430	0.1649E 01	0.0320
74	116.67	50.00	49.9996	0.1649E 01	0.0002
75	175.06	0.0	49.9923	0.7325E 00	0.0043
76	175.06	50.00	50.0002	0.7325E 00	-0.0001
77	262.64	0.0	49.9992	0.3255E 00	0.0005
78	262.64	50.00	49.9996	0.3255E 00	0.0002
79	394.00	0.0	49.9999	0.1446E 00	0.0000
80	394.00	50.00	49.9999	0.1446E 00	0.0000
81	544.00	0.0	49.9999	0.7586E-01	0.0001
82	544.00	50.00	49.9999	0.7586E-01	0.0000
83	694.00	0.0	49.9999	0.4661E-01	0.0001
84	694.00	50.00	49.9999	0.4661E-01	0.0001
85	758.42	0.0	49.9999	0.3903E-01	0.0001
86	758.42	50.00	49.9999	0.3903E-01	0.0000
87	855.05	0.0	49.9999	0.3070E-01	0.0000
88	855.05	50.00	49.9999	0.3070E-01	0.0001
89	1000.00	0.0	50.0000	0.2245E-01	0.0
90	1000.00	50.00	50.0000	0.2245E-01	0.0

```
***********************************
*    TIME STEP NUMBER  #    10     *
***********************************


*****************************************
*          TIME  #     0.241E 03        *
*****************************************

ESTIMATED RADIUS OF INFLUENCE  #   1000.00
CORRESPONDING NO. OF NODES  #   90
CORRESPONDING NO. OF ELEMENTS   #  121
CORRESPONDING COMPONENT OF VECTOR NDVEC   #   22
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER   #          1
    HEAD INCREMENT   #    -1.922
    HEAD VALUE   #    14.836

NUMBER OF ITERATIONS REQUIRED #          2

TOLERANCE COUNTER FOR HEAD   #  0
ABSOLUTE MAXIMUM ERROR IN HEAD   #       0.0632

DISCHARGE FROM AQUIFER INTO WELL   #     98.8546
DISCHARGE FROM WELL STORAGE #         0.019
TOTAL CALCULATED DISCHARGE   #        98.873
RESIDUAL DISCHARGE          #    1.1266
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

DISCHARGE ITERATION NUMBER   #          2
    HEAD INCREMENT   #    -2.825
    HEAD VALUE   #    13.933

NUMBER OF ITERATIONS REQUIRED #          2

TOLERANCE COUNTER FOR HEAD   #  0
ABSOLUTE MAXIMUM ERROR IN HEAD   #       0.1431

DISCHARGE FROM AQUIFER INTO WELL   #     99.6348
DISCHARGE FROM WELL STORAGE #         0.028
TOTAL CALCULATED DISCHARGE   #        99.662
RESIDUAL DISCHARGE          #    0.3376
```

FREE SURFACE POSITON

RADIUS	SURFACE HEIGHT
0.50	40.1313
0.80	40.0242
1.25	40.0407
1.92	40.3750
2.94	40.8233
4.46	41.3799
6.73	42.0860
10.15	43.0126
15.28	43.9414
22.97	45.0465
34.50	46.2963
51.80	47.4841
77.75	48.5370
116.67	49.3396
175.06	49.8051
262.64	49.9728
394.00	49.9986
544.00	49.9990
694.00	49.9988
758.42	49.9990
855.05	49.9995
1000.00	50.0000

SURFACE TOLERANCE COUNTER # 0

MAXIMUM ERROR OF SURFACE HEIGHT # 0.1431

```
*******************************************
*                                         *
*        FINAL RESULTS OF ANALYSIS        *
*                                         *
*******************************************
```

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	Z-COORD	HEAD	1/U	W(U)
1	0.50	0.0	13.9327	0.3448E 08	20.2419
2	0.50	2.00	13.9327	0.3448E 08	20.2419
3	0.50	4.00	13.9327	0.3448E 08	20.2419
4	0.50	6.00	13.9327	0.3448E 08	20.2419
5	0.50	8.00	13.9327	0.3448E 08	20.2419
6	0.50	10.00	13.9327	0.3448E 08	20.2419
7	0.50	12.00	13.9327	0.3448E 08	20.2419
8	0.50	14.00	14.0268	0.3448E 08	20.1890
9	0.50	16.00	16.0397	0.3448E 08	19.0593
10	0.50	18.00	18.0436	0.3448E 08	17.9347
11	0.50	20.00	20.0474	0.3448E 08	16.8101
12	0.50	25.03	25.0596	0.3448E 08	13.9971
13	0.50	30.06	30.0758	0.3448E 08	11.1820
14	0.50	35.10	35.0991	0.3448E 08	8.3627
15	0.50	40.13	40.1313	0.3448E 08	5.5386
16	0.80	0.0	19.5971	0.1347E 08	17.0628
17	0.80	4.00	20.4337	0.1347E 08	16.9933
18	0.80	8.00	20.4439	0.1347E 08	16.5876
19	0.80	12.00	20.5849	0.1347E 08	16.5085
20	0.80	16.00	22.3202	0.1347E 08	15.5346
21	0.80	20.00	25.1301	0.1347E 08	13.9576
22	0.80	25.00	28.8324	0.1347E 08	11.8798
23	0.80	30.01	32.6452	0.1347E 08	9.7399
24	0.80	35.02	36.4397	0.1347E 08	7.6104
25	0.80	40.02	40.0242	0.1347E 08	5.5987
26	1.25	0.0	24.4734	0.5517E 07	14.3261
27	1.25	4.00	25.1600	0.5517E 07	13.9408
28	1.25	8.00	25.2556	0.5517E 07	13.8871
29	1.25	12.00	25.5959	0.5517E 07	13.6962
30	1.25	16.00	27.0260	0.5517E 07	12.8936
31	1.25	20.00	28.8189	0.5517E 07	11.8873
32	1.25	26.68	32.6908	0.5517E 07	9.7143
33	1.25	33.36	36.5994	0.5517E 07	7.5207
34	1.25	40.04	40.0407	0.5517E 07	5.5894
35	1.92	0.0	28.2958	0.2326E 07	12.1809
36	1.92	5.00	28.9403	0.2326E 07	11.8192
37	1.92	10.00	29.0594	0.2326E 07	11.7524
38	1.92	15.00	30.3140	0.2326E 07	11.0483
39	1.92	20.00	32.0025	0.2326E 07	10.1007
40	1.92	26.79	34.8978	0.2326E 07	8.4757
41	1.92	33.58	37.9318	0.2326E 07	6.7730
42	1.92	40.37	40.3750	0.2326E 07	5.4018
43	2.94	0.0	31.5630	0.9990E 06	10.3473
44	2.94	5.00	32.1562	0.9990E 06	10.0144
45	2.94	10.00	32.2700	0.9990E 06	9.9505
46	2.94	15.00	33.4842	0.9990E 06	9.2690
47	2.94	20.00	34.5539	0.9990E 06	8.6687
48	2.94	30.41	37.9532	0.9990E 06	6.7610
49	2.94	40.82	40.8233	0.9990E 06	5.1502
50	4.46	0.0	34.3489	0.4341E 06	8.7938
51	4.46	10.00	35.2781	0.4341E 06	8.2623
52	4.46	20.00	36.8698	0.4341E 06	7.3690
53	4.46	30.69	39.3277	0.4341E 06	5.9835
54	4.46	41.38	41.3799	0.4341E 06	4.8379
55	6.73	0.0	36.9729	0.1901E 06	7.3111
56	6.73	10.00	37.8179	0.1901E 06	6.8369
57	6.73	20.00	38.9483	0.1901E 06	6.2025
58	6.73	42.09	42.0860	0.1901E 06	4.4415
59	10.15	0.0	39.2779	0.8365E 05	6.0175
60	10.15	20.00	40.7644	0.8365E 05	5.1833

61	10.15	43.01	43.0126	0.8365E 05	3.9215
62	15.28	0.0	41.5314	0.3693E 05	4.7528
63	15.28	20.00	42.5743	0.3693E 05	4.1675
64	15.28	43.94	43.9414	0.3693E 05	3.4002
65	22.97	0.0	43.5463	0.1634E 05	3.6220
66	22.97	45.05	45.0465	0.1634E 05	2.7800
67	34.50	0.0	45.3307	0.7243E 04	2.6205
68	34.50	46.30	46.2963	0.7243E 04	2.0786
69	51.80	0.0	46.8986	0.3213E 04	1.7406
70	51.80	47.48	47.4841	0.3213E 04	1.4120
71	77.75	0.0	48.1953	0.1426E 04	1.0129
72	77.75	48.54	48.5370	0.1426E 04	0.8211
73	116.67	0.0	49.1569	0.6333E 03	0.4732
74	116.67	49.34	49.3396	0.6333E 03	0.3706
75	175.06	0.0	49.7306	0.2813E 03	0.1512
76	175.06	49.81	49.8051	0.2813E 03	0.1094
77	262.64	0.0	49.9553	0.1250E 03	0.0251
78	262.64	49.97	49.9728	0.1250E 03	0.0152
79	394.00	0.0	49.9966	0.5553E 02	0.0019
80	394.00	50.00	49.9986	0.5553E 02	0.0008
81	544.00	0.0	49.9989	0.2913E 02	0.0006
82	544.00	50.00	49.9990	0.2913E 02	0.0006
83	694.00	0.0	49.9988	0.1790E 02	0.0007
84	694.00	50.00	49.9988	0.1790E 02	0.0007
85	758.42	0.0	49.9990	0.1499E 02	0.0006
86	758.42	50.00	49.9990	0.1499E 02	0.0006
87	855.05	0.0	49.9996	0.1179E 02	0.0002
88	855.05	50.00	49.9995	0.1179E 02	0.0003
89	1000.00	0.0	50.0000	0.8620E 01	0.0
90	1000.00	50.00	50.0000	0.8620E 01	0.0

```
****************************************
*         TIME  #     0.322E 03        *
****************************************
```

- -

DISCHARGE ITERATION NUMBER # 1

HEAD INCREMENT # -1.922

HEAD VALUE # 12.010

NUMBER OF ITERATIONS REQUIRED # 2

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERRCR IN HEAD # 0.0554

DISCHARGE FROM AQUIFER INTO WELL # 99.4254

DISCHARGE FROM WELL STORAGE # 0.019

TOTAL CALCULATED DISCHARGE # 99.444

RESIDUAL DISCHARGE # 0.5558

- -

DISCHARGE ITERATION NUMBER # 2

HEAD INCREMENT # -2.483

HEAD VALUE # 11.449

NUMBER OF ITERATIONS REQUIRED # 3

TOLERANCE COUNTER FOR HEAD # 0

ABSOLUTE MAXIMUM ERROR IN HEAD # 0.0411

DISCHARGE FROM AQUIFER INTO WELL # 100.1219

DISCHARGE FROM WELL STORAGE # 0.024

TOTAL CALCULATED DISCHARGE # 100.146

RESIDUAL DISCHARGE # 0.1462

FREE SURFACE POSITON

RADIUS	SURFACE HEIGHT
0.50	39.5186
0.80	39.4118
1.25	39.4423
1.92	39.7702
2.94	40.2160
4.46	40.7682
6.73	41.4775
10.15	42.4025
15.28	43.3408
22.97	44.4739
34.50	45.7616
51.80	47.0092
77.75	48.1541
116.67	49.0834
175.06	49.6844
262.64	49.9438
394.00	49.9971
544.00	49.9990
694.00	49.9988
758.42	49.9990
855.05	49.9996
1000.00	50.0000

SURFACE TOLERANCE COUNTER # 0

MAXIMUM ERROR OF SURFACE HEIGHT # 0.0411

****
FINAL RESULTS OF ANALYSIS
****

HEAD .VS. RADIUS AND 1/U .VS. W(U)

NODE	R-COORD	Z-COORD	HEAD	1/U	W(U)
1	0.50	0.0	11.4492	0.4598E 08	21.6356
2	0.50	2.00	11.4492	0.4598E 08	21.6356
3	0.50	4.00	11.4492	0.4598E 08	21.6356
4	0.50	6.00	11.4492	0.4598E 08	21.6356
5	0.50	8.00	11.4492	0.4598E 08	21.6356
6	0.50	10.00	11.4492	0.4598E 08	21.6356
7	0.50	12.00	11.9868	0.4598E 08	21.3339
8	0.50	14.00	13.9832	0.4598E 08	20.2135
9	0.50	16.00	15.9798	0.4598E 08	19.0929
10	0.50	18.00	17.9780	0.4598E 08	17.9715
11	0.50	20.00	19.9763	0.4598E 08	16.8500
12	0.50	24.88	24.8595	0.4598E 08	14.1094
13	0.50	29.76	29.7427	0.4598E 08	11.3689
14	0.50	34.64	34.6277	0.4598E 08	8.6273
15	0.50	39.52	39.5186	0.4598E 08	5.8824
16	0.80	0.0	17.6217	0.1796E 08	18.1715
17	0.80	4.00	18.5432	0.1796E 08	17.6543
18	0.80	8.00	18.5433	0.1796E 08	17.6543
19	0.80	12.00	19.3183	0.1796E 08	17.2193
20	0.80	16.00	22.1100	0.1796E 08	15.6525
21	0.80	20.00	24.8753	0.1796E 08	14.1006
22	0.80	24.85	28.5039	0.1796E 08	12.0641
23	0.80	29.70	32.2285	0.1796E 08	9.9738
24	0.80	34.56	35.9286	0.1796E 08	7.8972
25	0.80	39.41	39.4118	0.1796E 08	5.9423
26	1.25	0.0	22.9018	0.7356E 07	15.2082
27	1.25	4.00	23.6584	0.7356E 07	14.7835
28	1.25	8.00	23.7953	0.7356E 07	14.7067
29	1.25	12.00	24.5589	0.7356E 07	14.2781
30	1.25	16.00	26.6211	0.7356E 07	13.1208
31	1.25	20.00	28.4525	0.7356E 07	12.0930
32	1.25	26.48	32.2620	0.7356E 07	9.9550
33	1.25	32.96	36.0900	0.7356E 07	7.8066
34	1.25	39.44	39.4423	0.7356E 07	5.9252
35	1.92	0.0	27.0080	0.3102E 07	12.9037
36	1.92	5.00	27.7238	0.3102E 07	12.5019
37	1.92	10.00	28.0013	0.3102E 07	12.3462
38	1.92	15.00	29.7120	0.3102E 07	11.3861
39	1.92	20.00	31.5553	0.3102E 07	10.3516
40	1.92	26.59	34.4114	0.3102E 07	8.7487
41	1.92	33.18	37.3889	0.3102E 07	7.0777
42	1.92	39.77	39.7702	0.3102E 07	5.7412
43	2.94	0.0	30.4937	0.1332E 07	10.9474
44	2.94	5.00	31.1613	0.1332E 07	10.5727
45	2.94	10.00	31.3774	0.1332E 07	10.4514
46	2.94	15.00	32.8486	0.1332E 07	9.6258
47	2.94	20.00	34.0479	0.1332E 07	8.9527
48	2.94	30.11	37.4092	0.1332E 07	7.0662
49	2.94	40.22	40.2160	0.1332E 07	5.4910
50	4.46	0.0	33.4433	0.5788E 06	9.2920
51	4.46	10.00	34.5184	0.5788E 06	8.6856
52	4.46	20.00	36.3233	0.5768E 06	7.6757
53	4.46	30.38	38.7560	0.5788E 06	6.3104
54	4.46	40.77	40.7682	0.5788E 06	5.1811
55	6.73	0.0	36.2068	0.2534E 06	7.7411
56	6.73	10.00	37.1400	0.2534E 06	7.2173
57	6.73	20.00	38.3818	0.2534E 06	6.5204
58	6.73	41.48	41.4775	0.2534E 06	4.7830
59	10.15	0.0	33.6169	0.1115E 06	6.3084
60	10.15	20.00	40.1926	0.1115E 06	5.5042
61	10.15	42.40	42.4025	0.1115E 06	4.2639
62	15.28	0.0	40.9444	0.4925E 05	5.0822
63	15.28	20.00	42.0183	0.4925E 05	4.4795
64	15.28	43.34	43.3408	0.4925E 05	3.7373
65	22.97	0.0	43.0262	0.2179E 05	3.9139
66	22.97	44.47	44.4739	0.2179E 05	3.1014
67	34.50	0.0	44.8557	0.9657E 04	2.8871
68	34.50	45.76	45.7616	0.9657E 04	2.3787
69	51.80	0.0	46.4769	0.4284E 04	1.9772
70	51.80	47.01	47.0092	0.4284E 04	1.6785
71	77.75	0.0	47.8466	0.1901E 04	1.2085
72	77.75	48.15	48.1541	0.1901E 04	1.0359
73	116.67	0.0	48.9105	0.8444E 03	0.6114
74	116.67	49.03	49.0834	0.8444E 03	0.5144
75	175.06	0.0	49.6041	0.3751E 03	0.2222
76	175.06	49.68	49.6844	0.3751E 03	0.1771
77	262.64	0.0	49.9205	0.1666E 03	0.0446
78	262.64	49.94	49.9438	0.1666E 03	0.0315
79	394.00	0.0	49.9936	0.7404E 02	0.0036
80	394.00	50.00	49.9971	0.7404E 02	0.0016
81	544.00	0.0	49.9987	0.3884E 02	0.0007
82	544.00	50.00	49.9990	0.3884E 02	0.0006
83	694.00	0.0	49.9988	0.2386E 02	0.0007
84	694.00	50.00	49.9988	0.2386E 02	0.0007
85	758.42	0.0	49.9991	0.1998E 02	0.0005
86	758.42	50.00	49.9990	0.1998E 02	0.0005
87	855.05	0.0	49.9996	0.1572E 02	0.0002
88	855.05	50.00	49.9996	0.1572E 02	0.0002
89	1000.00	0.0	50.0000	0.1149E 02	0.0
90	1000.00	50.00	50.0000	0.1149E 02	0.0

TOTAL NUMBER OF PUNCHED CARDS # 16

4. LISTINGS OF THE PROGRAMS

4.1 LISTING OF STCON1

```
C
C     INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C     DEVELOPED BY P.S. HUYAKORN.
C
C     STCON1,PROGRAM FOR ANALYSING STEADY,ONE-DIMENSIONAL DARCY OR TWO-REGIME
C
C     FLOW TOWARDS A PUMPED WELL.
C
C     VERSION DATED OCTOBER,1973.
C
C      FOR FURTHER INFORMATION,CONTACT
C
C        P.S. HUYAKORN OR C.R. DUDGEON
C
C        WATER RESEARCH LABORATORY
C
C        KING ST.,MANLY VALE
C
C        SYDNEY,N.S.W. 2093,AUSTRALIA.
C
C
C------------------------------
C     LIST OF INPUT VARIABLES
C------------------------------
C
C------------------ALL READ STATEMENTS ARE LOCATED IN THE MAIN-------------------
C------------------PROGRAM AND INDICATED BY  $1 OR $1,SK  SIGN-------------------
C
C***  PROBLEM VARIABLES  ***
C
C     NPROB    =  NUMBER OF PROBLEMS TO BE SOLVED
C     IVEL     =  VELOCITY PRINT-OUT INDEX.FEED IN IVEL=0 IF VELOCITY PRINT-OUT
C                 IS NOT REQUIRED OTHERWISE FEED IN IVEL=1
C     IDISCR   =  DISCRETISATION DATA PRINT-OUT INDEX
C                 FEED IN IDISCR=0 IF DISCRETISATION PRINT-OUT IS NOT REQUIRED
C                 OTHERWISE FEED IN IDISCR=1
C     ORELAX   =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                 SUGGESTED VALUE LIES BETWEEN 1.50 TO 1.85
C     RW       =  RADIUS OF WELL SCREEN
C     RO       =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C     HO       =  HEIGHT OR DRAWDOWN OF WATER TABLE AT THE EXTERNAL RADIUS
C     TH       =  THICKNESS OF AQUIFER
C     HTOL     =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                 SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C     AK       =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                 FEED IN AK=1./PM IF ONLY DARCY FLOW SOLUTION IS REQUIRED
C                 WHERE PM = COEFFICIENT OF PERMEABILITY OF AQUIFER
C     BK       =  NON-LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                 IF ONLY DARCY FLOW SOLUTION IS REQUIRED SET BK=0.
C     VCR      =  CRITICAL FLOW VELOCITY WHERE NON-DARCY FLOW COMMENCES
C     IGP      =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                 IGP=0 FOR NON-GRAVEL PACKED WELL
C     IAQTA    =  AQUITARD INDEX,IAQTA=1 IF THERE IS AN OVERLYING AQUITARD
C                 IAQTA=0 IF THE MAIN AQUIFER IS CONFINED BY IMPERMEABLE STRATA
C     IWBC     =  WELL BOUNDARY CONDITION INDEX,IWBC=0 IF WELL DISCHARGE
C                 IS PRESCRIBED,IWBC=1 IF WELL DRAWDOWN OR HEAD IS PRESCRIBED
C     AGP      =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     BGP      =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     VGP      =  CRITICAL FLOW VELOCITY FOR GRAVEL PACK MATERIAL
C     THGP     =  THICKNESS OF GRAVEL PACK
C     PA       =  COEFFICIENT OF PERMEABILITY OF OVERLYING AQUITARD
C     THA      =  THICKNESS OF OVERLYING AQUITARD
C     HW       =  HEIGHT OF WATER LEVEL IN THE WELL OR WELL DRAWDOWN
C     QFIX     =  PRESCRIBED WELL DISCHARGE
C
C***  DISCRETISATION PARAMETERS  ***
C
C     NELF     =  NUMBER OF 3-NODE LINE ELEMENTS IN FIRST SUBREGION
C                 SUGGESTED VALUE IS NELF=2
C     SCFAC    =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                 SUBREGIONS.SUGGESTED VALUE IS SCFAC=2.0
C     FRLEN    =  LENGTH OF FIRST SUBREGION
C                 SUGGESTED VALUE IS FRLEN=4.*RW
C                 FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
C------------------------------
C     LIST OF OUTPUT VARIABLES
C------------------------------
C     NNODE    =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C     NELEM    =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C     X        =  NODAL RADIAL DISTANCES FROM CENTRE LINE OF PUMPED WELL
C     H        =  NODAL HEAD OR DRAWDOWN VALUES
C     TLESS    =  NODAL VALUES OF DIMENSIONLESS RADIUS,1/U
C     HLESS    =  NODAL VALUES OF WELL FUNCTION FOR STEADY FLOW,W(U)
C     QW       =  DISCHARGE FROM AQUIFER INTO WELL AT DRAWDOWN SW
C     NOD      =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C     VEL      =  ELEMENT VELOCITIES
C     VMEAN    =  AVERAGE NODAL VELOCITIES
C     IDARCY   =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                 OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                 IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
```

```
C
C
C
C
C
      DIMENSION X(200,1),NOD(150,6),NBAND(200),ID(200)
      DIMENSION M(40),XLEN(40),NNE(40),JBD(2),DISP(2)
      DIMENSION XEN(6),HE(6),DELK(6,6)
      DIMENSION VK(1000),CK(200,1),H(200),VEL(150,6),VMEAN(200,1)
      DIMENSION SELK(6,6),ZEN(6),DSHF(6)
      DIMENSION IPROP(150),IDARCY(150),HDAR(200)
      DIMENSION WK(10),VD(1000)
C
      DATA NREAD,NPRINT/1,3/
C
C     PRINT INITIAL HEADINGS
C
      WRITE(NPRINT,1003)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1023)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1033)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1003)
  993 FORMAT('1',4X,51H ************************************************
     1**)
 1003 FORMAT(5X,51H **************************************************)
 1013 FORMAT(5X,51H *                                                *)
 1023 FORMAT(5X,51H *           FINITE ELEMENT ANALYSIS OF           *)
 1033 FORMAT(6X,'*   STEADY STATE,ONE-DIMENSIONAL,CONFINED FLOW.  *')
      ITS=1
      ITND=20
      READ(NREAD,1011)NPROB,IVEL,IDISCR,ORELAX                           $1
 1011 FORMAT(3I10,F10.2)
      DO 9005 JPRO=1,NPROB
      WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1          I6,12X,'*',/,20X,50('*'))
C
C     READ AND PRINT GENERAL DATA
C
 0001 READ (NREAD,2003) RW,RO,HO,TH,HTOL                                 $1
 2003 FORMAT(5F10.2)
 2004 WRITE (NPRINT,23) RW,RO,HO,TH,HTOL
   23 FORMAT(//,15X,18HGENERAL INPUT DATA,//,
     1          10X,32HAQUIFER AND WELL CHARACTERISTICS,//,
     310X,19HRADIUS OF WELL    = ,F10.3,//,
     410X,22HRADIUS OF INFLUENCE = ,F10.3,//,
     510X,24HHEIGHT OF WATER TABLE = ,F10.3,//,
     610X,24HTHICKNESS OF AQUIFER   = ,F10.3,//,
     710X,39HTOLERANCE FOR SUCCESSIVE HEAD VALUES = ,F10.3)
C
C     READ AQUIFER AND GRAVEL PACK DATA
C
      READ(NREAD,2013)AK, BK, VCR                                        $1
 2013 FORMAT(3F10.3)
      GCR=AK*VCR+BK*VCR**2
      PM=1./AK
      IF(GCR.NE.0) PM=VCR/GCR
      WRITE(NPRINT,53) AK,BK,VCR,PM
   53 FORMAT(///,20X,'AQUIFER PROPERTIES'///
     1          10X,'COEFFICIENT A =',F10.4//
     2          10X,'COEFFICIENT B = ',F10.4//
     3          10X,'CRITICAL VELOCITY = ',F10.4//
     4          10X,'COEFFICIENT K = ',F10.4//)
      READ(NREAD,111)IGP, IAQTA, IWBC                                    $1
  111 FORMAT(3I10)
      RGP=RW
      IF(IGP.EQ.0) GO TO 29
      READ(NREAD,331)AGP,BGP,VGP,THGP                                    $1
  331 FORMAT(4F10.3)
      GRGP=AGP*VGP+BGP*VGP**2
      PMGP=1./AGP
      IF(GRGP.NE.0) PMGP=VGP/GRGP
      WRITE(NPRINT,153)AGP,BGP,VGP,PMGP
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1          10X,'COEFFICIENT A =',F10.4//
     2          10X,'COEFFICIENT B = ',F10.4//
     3          10X,'CRITICAL VELOCITY = ',F10.4//
     4          10X,'COEFFICIENT K =',F10.4,/)
      RGP=RW+THGP
      WRITE(NPRINT,163)THGP,RGP
  163 FORMAT(10X,'THICKNESS OF PACK = ',F10.2//
     1          10X,'RADIUS OF PACK = ',F10.2//)
   29 CONTINUE
C
C     READ AQUITARD DATA.
C
      IF(IAQTA.EQ.0) GO TO 39                                            $1,SK
      READ(NREAD,71)PA,THA
```

```
   71 FORMAT(F10.4,F10.2)
      BFAC=PA/THA
      WRITE(NPRINT,273)PA,THA
  273 FORMAT(///,20X,'AQUITARD PROPERTIES',///,
     1           10X,'HYDRAULIC CONDUCTIVITY    =',F10.5//
     2           10X,'THICKNESS    =',F10.2//)
   39 CONTINUE
C
C
C     READ PRESCRIBED VALUE OF WELL DRAWDOWN OR DISCHARGE
C
      IF(IWBC.EQ.0) GO TO 59
      READ(NREAD,2023)HW                                                $1,SK
 2023 FORMAT(F10.2)
      GO TO 69
   59 CONTINUE
      READ(NREAD,2023)QFIX                                              $1,SK
      TWPI=44./7.
      QFL=QFIX/(TWPI*TH)
   69 CONTINUE
C
C     GENERATE AND PRINT DISCRETIZATION PARAMETER.
C
      READ(NREAD,391)NELF,SCFAC,FRLEN                                   $1
  391 FORMAT(I10,2F10.2)
      WRITE(NPRINT,1773) ORELAX
 1773 FORMAT(10X,'OVER-RELAXATION FACTOR  =',F10.4)
      CALL DCRGEN(RO,RW,NELF,SCFAC,FRLEN,NRR,NNE,M,XLEN)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,33)NELF,SCFAC,FRLEN
   33 FORMAT(2(/),15X,'DISCRETISATION DATA',///,
     1        10X,'NUMBER OF ELEMENTS IN FIRST SUBREGION  =',I10,///,
     2          10X,'SCALE FACTOR FOR ELEMENT LENGTH    =',F10.3,///,
     3          10X,'LENGTH OF FIRST SUBREGION    =',F10.3,///,
     4'   SUBREGION  ',10X,'NO. OF ELEMENTS',15X,'LENGTH OF SUBREGION',
     5    //)
C
   31 FORMAT(I3)
      DO 55 I=1,NRR
   41 FORMAT(2I3,F8.2)
      WRITE(NPRINT,43) I,M(I),XLEN(I)
   43 FORMAT(5X,I3,20X,I3,21X,F9.2)
   55 CONTINUE
   94 CONTINUE
      RI=RW
      NFNOD=1
      NNODE=0
      NELEM=0
      NSEL=1
C
C
      DO 65 K=1,NRR
      NNODE=NNODE+(NNE(K)-1)*M(K)
      NELEM=NELEM+M(K)
   65 CONTINUE
      NNODE=NNODE+1
C
C     PRINT OUT NO OF NODES AND NO OF ELEMS
C
      WRITE(NPRINT,73) NNODE,NELEM
   73 FORMAT(3(/),12X,'TOTAL NUMBER OF NODES =',I5,//,12X,
     1'TOTAL NUMBER OF ELEMENTS =',I5)
      DO 75 II=1,NRR
      MM=M(II)
      NN=NNE(II)
      XL=XLEN(II)
C
C      GENERATE THE NODE COORDINATES
C
      CALL CORDC1(XL,MM,NN,X,RI,NFNOD)
C
C     DISCRETIZE THE REGION INTO ELEMS
C
      CALL CONC1(NSEL,NOD,NFNOD,MM,NN)
      NFNOD=NFNOD+(NNE(II)-1)*M(II)
   75 CONTINUE
      CALL HYPROP(IPROP,NELEM,RGP,X,NOD)
C
C
C     FIND BANDWIDTH ETC...
      NN=3
      CALL EBFIND(NBAND,ID,LEN,NOD,NELEM,NN,NNODE)
      IF(IAQTA.EQ.0) GO TO 200
C
C     FORM GROSS VECTOR,VD
C
      ZEN(1)=-1.0
      ZEN(2)=1.0
      ZEN(3)=0.0
      DO 130 I=1,LEN
```

```
      VD(I)=0.0
  130 CONTINUE
      DO 135 IE=1,NELEM
      DO 145 J=1,NN
      NJ=NOD(IE,J)
      XEN(J)=X(NJ,1)
  145 CONTINUE
      SC=1.
      CALL ELGND(XEN,NN,DELK,IE,SC,0)
      CALL MERBS(VD,NBAND,ID,DELK,NOD,NN,IE)
  135 CONTINUE
  200 CONTINUE
C
C     SET IDARCY
C
      IF(ITS.EQ.2) GO TO 599
      DO 99 I=1,NELEM
      IDARCY(I)=0
   99 CONTINUE
  599 CONTINUE
      IF(ITS.EQ.1) GO TO 82
      DO 182 I=1,NNODE
      H(I)=HDAR(I)
  182 CONTINUE
   82 CONTINUE
      ZEN(1)=-1.0
      ZEN(2)=1.0
      ZEN(3)=0.0
C
C     SET UP LOOP FOR ITERATION ON HEAD
      DO 999 III=ITS,ITND
C
C     ZERO GROSS STIFFNESS MATRIX
C
      DO 300 I=1,LEN
      VK(I)=0.0
  300 CONTINUE
      DO 305 I=1,NNODE
      CK(I,1)=0.0
  305 CONTINUE
      IF(IWBC.EQ.0) CK(1,1)=-QFL
C
C     COMPUTE STIFFNESS MATRIX FOR EACH ELEM AND MERGE
C
      NN=3
      DO 350 IE=1,NELEM
      DO 355 J=1,NN
      NJ=NOD(IE,J)
      XEN(J)=X(NJ,1)
      IF(III.EQ.1) GO TO 355
      HE(J)=H(NJ)
  355 CONTINUE
      IF(IPROP(IE).EQ.0) GO TO 755
      AKK=AK
      BKK=BK
      PKK=PM
      GO TO 365
  755 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
  365 CONTINUE
      CALL ELGEN1(XEN,NN,SELK,AKK,BKK,HE,PKK,IDARCY,IE,0)
      CALL MERBS(VK,NBAND,ID,SELK,NOD,NN,IE)
  350 CONTINUE
C
C     ADD LEAKAGE TERM CONTRIBUTED BY AQUITARD.
C
      IF(IAQTA.EQ.0) GO TO 86
      FTERM=BFAC/TH
      DO 106 I=1,LEN
  106 VK(I)=VK(I)+VD(I)*FTERM
   86 CONTINUE
C
C     SOLVE EQNS BY BANDED ELIMINATION SCHEME
C
      NBAN=NBAND(1)
      DO 105 I=1,NBAN
      WK(I)=VK(I)
  105 CONTINUE
      NBD=1
      JBD(1)=NNODE
      DISP(1)=HO
      CALL PBC(NBD,JBD,CK,DISP,ID,VK)
C
C     FIX WELL DRAWDOWN.
C
      IF(IWBC.EQ.0) GO TO 496
      JBD(1)=1
      DISP(1)=HW
```

```
      CALL PBC(NBD,JBD,CK,DISP,ID,VK)
  496 CONTINUE
      NLL= 1
      CALL SOLVBS(CK,VK,NBAND,NNODE,NLL)
C
C     PRINT OUT SOLUTION FOR FUNCTION
C     ONLY THE DARCY AND FINAL SOLUTIONS
C
      IF(III.EQ.1) GO TO 900
      IF(III.NE.2)  GO TO 410
      GO TO 415
  410 CONTINUE
  415 CONTINUE
      NCOUNT=0
      EMAX=0.
      DO 450 I=1,NNODE
      EPSI=CK(I,1)-H(I)
      EPSIS=EPSI
      IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.HTOL)  GO TO 460
      NCOUNT=NCOUNT+1
      H(I)=H(I)+ORELAX*EPSIS
      GO TO 450
  460 CONTINUE
      H(I)=CK(I,1)
  450 CONTINUE
      IF(NCOUNT.EQ.0) GO TO 1000
      GO TO 1999
  900 CONTINUE
      DO 950 I=1,NNODE
      H(I)=CK(I,1)
      HDAR(I)=H(I)
  950 CONTINUE
 1999 CONTINUE
C
C     REGENERATE IDARCY
C
      DO 199 I=1,NELEM
      CALL SETARG(AK,BK,PM,AGP,BGP,PMGP,AKK,BKK,PKK,IPROP,I)
      CALL VCHECK(NOD,H,X,ZEN,IDARCY,VEL,NN,AKK,BKK,PKK,VCR,I)
  199 CONTINUE
  999 CONTINUE
 1000 CONTINUE
      WRITE(NPRINT,543)
  543 FORMAT(///,20X,26('*'),/,20X,'*',5X,'FINAL SOLUTION',5X,
     1            '*',/,20X,26('*'))
      WRITE(NPRINT,413)III,EMAX
  413 FORMAT(////,10X,'NO. OF ITERATIONS REQUIRED  =',
     1          I5,//,10X,'ABSOLUTE MAXIMUM ERROR IN HEAD =',F10.3)
      IF(NCOUNT.NE.0)  GO TO 5000
      WRITE(NPRINT,433)
  433 FORMAT(//,20X,'HEAD VERSUS RADIAL DISTANCE',//,
     1            10X,'NODE',8X,'R-COORDINATE',8X,'HEAD VALUE',8X,
     2            'RHO-COORD',8X,'TZI-COORD',//)
      HW=H(1)
      DO 470 I=1,NNODE
      RHO=0.5*X(I,1)/TH
      TZI=(HO-H(I))/(HO-HW)
      WRITE(NPRINT,1953)I,X(I,1),H(I),RHO,TZI
  470 CONTINUE
 1953 FORMAT(10X,I3,4(8X,F10.4))
C
C     EVALUATE NODAL VELOCITY FOR EACH ELEMENT
C
      IF(IVEL.EQ.0) GO TO 7000
      DO 3000 I=1,NELEM
      CALL SETARG(AK,BK,PM,AGP,BGP,PMGP,AKK,BKK,PKK,IPROP,I)
      CALL VCHECK(NOD,H,X,ZEN,IDARCY,VEL,NN,AKK,BKK,PKK,VCR,I)
 3000 CONTINUE
C
C     AVERAGE THE VELOCITY AT EACH NODE
C
      NN=3
      CALL VAVEG1(VMEAN,VEL,NOD,NNODE,NELEM,NN)
 7000 CONTINUE
C
      TWPI=44.0/7.0
      QW=0.0
      DO 206 I=1,NBAN
      QW=QW+WK(I)*H(I)
  206 CONTINUE
      QW=(-(TWPI*TH*QW))
      WRITE (NPRINT,2001) QW
 2001 FORMAT(///,10X,'DISCHARGE INTO WELL  =',F12.4)
      BLEAK=RO
      IF(IAQTA.EQ.0) GO TO 4505
      BLEAK=PM*TH*THA/PA
      BLEAK=SQRT(BLEAK)
 4505 CONTINUE
      CALL TCURV1(H,HO,BLEAK,X,NNODE,QW,AK,BK,PM,TH,1)
```

```
 5000 CONTINUE
C-----------------------------------------------------------------------C
C                                                                       C
C          IF SO REQUIRED,USER MAY CALL HIS PLOTTING SUBROUTINES        C
C                                                                       C
C          AT THIS LOCATION.                                            C
C                                                                       C
C-----------------------------------------------------------------------C
C
C     THE FOLLOWING PLOTS MAY BE OBTAINED -
C          (1) DRAWDOWN VERSUS RADIUS CURVE,OBTAINED BY USING VRAIABLES H,
C              X,NNODE
C          (2) DIMENSIONLESS TYPE CURVES,OBTAINED BY USING VARIABLES HLESS,
C              TLESS,NNODE
C          (3) NODAL VELOCITY VERSUS RADIUS,OBTAINED BY USING VARIABLES
C              VMEAN,X,NNODE
C
 9005 CONTINUE
 9000 WRITE (NPRINT,9100)
 9100 FORMAT (1H1)
      STOP
      END
      SUBROUTINE DCRGEN(RO,RW,NELF,SCFAC,FRLEN,NRR,NNE,M,XLEN)
C
C      GENERATES DISCRETISATION PARAMETERS.
C
      DIMENSION NNE(1),M(1),XLEN(1)
      MAXNR=40
      NNE(1)=3
      XLEN(1)=FRLEN
      M(1)=NELF
      RLEN=RO-RW
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      SUM=SUM+XLEN(I)
      M(I)=NELF
      IF(NELF.GT.1) M(I)=NELF-1
      IF(I.LE.3) M(I)=M(1)
      NNE(I)=NNE(1)
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I-1
C
C     SUBDIVIDE THE REMAINING LENGTH.
C
      ELEN=XLEN(NRR)/M(NRR)
      IF(XREM.LE.ELEN) GO TO 30
      NRR=NRR+1
      M(NRR)=XREM/ELEN
      XLEN(NRR)=XREM
      GO TO 40
   30 CONTINUE
      XREM=XREM+XLEN(NRR)+XLEN(NRR-1)
      XLEN(NRR-1)=XREM/(1.+SCFAC)
      XLEN(NRR)=XLEN(NRR-1)*SCFAC
   40 CONTINUE
      RETURN
      END
      SUBROUTINE CORDC1(XL,MM,NN,X,RI,NFNOD)
C
C     GENERATES NODAL COORDINATES.
C
      DIMENSION X(200,1)
      DXL=XL/(MM*(NN-1))
      NLNOD=NFNOD+(NN-1)*MM
      DO 10 J=NFNOD,NLNOD
      X(J,1)=RI+(J-NFNOD)*DXL
   10 CONTINUE
      RI=X(NLNOD,1)
      RETURN
      END
      SUBROUTINE CONC1(NSEL,NOD,NFNOD,MM,NN)
C
C     DISCRETISES THE REGION INTO 3-NODE LINE ELEMENTS.
C
      DIMENSION NDINT(6),NOD(150,6)
C
      NDINT(1)=NFNOD
      NDINT(2)=NFNOD+2
      NDINT(3)=NFNOD+1
      NLST=NSEL+MM-1
      DO 10 L=NSEL,NLST
      DO 20 K=1,NN
      NOD(L,K)=NDINT(K)+(L-NSEL)*(NN-1)
   20 CONTINUE
   10 CONTINUE
      NSEL=NLST+1
```

```
      RETURN
      END
      SUBROUTINE ELGND(XEN,NN,DELK,IE,SC,IVERT)
C
C     GENERATES ELEMENT MATRIX DELK FOR 3-NODE ELEMENTS.
C
      DIMENSION XEN(1),DELK(6,6),ZGSP(6),WC(6),SHF(6)
      DIMENSION DSHF(6)
C
      NGP=3
      ZGSP(1)=0.7745966692415
      ZGSP(2)=-ZGSP(1)
      ZGSP(3)=0.0
      WC(1)=5.0/9.0
      WC(2)=WC(1)
      WC(3)=8.0/9.0
      DO 15 I=1,NN
      DO 15 J=1,NN
      DELK(I,J)=0.0
   15 CONTINUE
      DO 10 I=1,NGP
      ZE=ZGSP(I)
      CALL SHFUN1(XEN,I,ZE,NN,XE,DSHF,TJ,SHF)
      XC=XE
      IF(IVERT.NE.0) XC=1.0
      DO 60 K=1,NN
      DO 50 L=1,NN
      IF(L-K) 50,40,40
   40 DELK(K,L)=DELK(K,L)+SHF(K)*SHF(L)*XC*WC(I)*SC*TJ
   50 CONTINUE
   60 CONTINUE
   10 CONTINUE
      DO 70 K=1,NN
      DO 80 L=1,NN
      IF(L-K) 90,80,80
   90 DELK(K,L)=DELK(L,K)
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE ELGEN1(XEN,NN,SELK,AK,BK,HE,PKK,IDARCY,IE,IVERT)
C
C     GENERATES ELEMENT MATRIX SELK FOR 3-NODE ELEMENTS.
C
      DIMENSION XEN(1),SELK(6,6),HE(1)
      DIMENSION ZGSP(6),WC(6),DSHF(6)
      DIMENSION IDARCY(1),SHF(6)
C
      NGP=3
      ZGSP(1)=0.7745966692415
      ZGSP(2)=-ZGSP(1)
      ZGSP(3)=0.0
      WC(1)=5.0/9.0
      WC(2)=WC(1)
      WC(3)=8.0/9.0
      DO 15 I=1,NN
      DO 15 J=1,NN
      SELK(I,J)=0.0
   15 CONTINUE
      DO 10 I=1,NGP
      ZE=ZGSP(I)
      CALL SHFUN1(XEN,I,ZE,NN,XE,DSHF,TJ,SHF)
      IF(IVERT.NE.0) XE=1.
      IF(IDARCY(IE).EQ.0) GO TO 25
      HS=0.0
      DO 30 JJ=1,NN
      HS=HS+DSHF(JJ)*HE(JJ)
   30 CONTINUE
      HS=ABS(HS/TJ)
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      CONS=1./TEMP
      GO TO 35
   25 CONTINUE
      CONS=PKK
   35 CONTINUE
      DO 60 K=1,NN
      DO 50 L=1,NN
      IF(L-K) 50,40,40
   40 SELK(K,L)=SELK(K,L)+(DSHF(K)*DSHF(L)*XE*WC(I)*CONS)/TJ
   50 CONTINUE
   60 CONTINUE
   10 CONTINUE
      DO 70 K=1,NN
      DO 80 L=1,NN
      IF(L-K) 90,80,80
   90 SELK(K,L)=SELK(L,K)
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SHFUN1(XEN,I,ZE,NN,XE,DSHF,TJ,SHF)
C
C     GENERATES SHAPE FUNCTION MATRICES,SHF AND DSHF,FOR 3-NODE ELEMENTS.
C
      DIMENSION SHF(6),DSHF(6),XEN(1)
C
      ZSQ=ZE**2
      SHF(1)=0.5*(-ZE+ZSQ)
      SHF(2)=0.5*(ZE+ZSQ)
      SHF(3)=1.0-ZSQ
      DSHF(1)=-.5+ZE
      DSHF(2)=0.5+ZE
      DSHF(3)=-2.0*ZE
      TJ=0.0
      XE=0.0
      DO 10 J=1,NN
      TJ=TJ+DSHF(J)*XEN(J)
      XE=XE+SHF(J)*XEN(J)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE HYPROP(IPROP,NELEM,RGP,X,NOD)
C
C     GENERATES PROPERTY INDEX FOR ALL ELEMENTS
C
      DIMENSION IPROP(1),NOD(150,3),X(200,1)
      DO 10 I=1,NELEM
      NJ=NOD(I,3)
      IPROP(I)=1
      IF(X(NJ,1).LT.RGP) IPROP(I)=0
   10 CONTINUE
      RETURN
      END
      SUBROUTINE EBFIND(NBAND,ID,LEN,LOC,NM,NN,LL)
C
C     COMPUTES BANDWIDTH FOR EACH ROW OF THE GROSS MATRIX.
C
      DIMENSION NBAND(1),ID(1),LOC(150,6),LV(6)
C
      DO 20 I = 1,LL
   20 NBAND(I) = 1
C
C     SCAN THROUGH THE LOCATION VECTOR FOR ELEM TO FIND THE POSITION
C     OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW.
C
      DO 25 I=1,NM
      NF2= NN
      DO 30 J=1,NF2
      LV(J) = LOC(I,J)
   30 CONTINUE
      DO 45 J =1,NF2
      IF (LV(J).EQ.0) GO TO 45
      DO 40 K=1,NF2
      IF (LV(J).GT.LV(K)) GO TO 40
      NW = LV(K)-LV(J)+1
      NR = LV(J)
      IF (NW.GT.NBAND(NR)) NBAND(NR)= NW
   40 CONTINUE
   45 CONTINUE
   25 CONTINUE
C     SET UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN = NBAND(1)
      DO 50 I = 2,LL
      IF (NBAND(I)- NBAND(I-1).LT.0) NBAND(I)= NBAND(I-1)-1
      LEN = LEN+NBAND(I)
   50 CONTINUE
      ID(1) =1
      DO 60 I=2,LL
   60 ID(I) = ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE MERBS(V,NBAND,ID,R,LOC,N1,M1)
C
C     MERGES ELEMENT MATRIX INTO GROSS VECTOR.
C
      DIMENSION V(1), R(6,6),LOC(150,6),NBAND(1),ID(1)
C
      DO 10 I=1,N1
      IK = LOC(M1,I)
      IF(IK.EQ.0) GO TO 10
      DO 20 J=1,N1
      JK = LOC(M1,J)
      IF (IK.GT.JK) GO TO 20
      IPOS = ID(IK)+JK-IK
      V(IPOS) = V(IPOS)+R(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
```

```
      END
      SUBROUTINE PBC(NBD,JBD,CK,DISP,ID,VK)
C
C     INTRODUCES PRESCRIBED BOUNDARY CONDITIONS.
C
      DIMENSION JBD(1),DISP(1),CK(200,1),ID(1),VK(1)
      DO 400 I=1,NBD
      NR=JBD(I)
      CK(NR,1)=1.D+20*DISP(I)
      IDT=ID(NR)
      VK(IDT)=1.D+20
  400 CONTINUE
      RETURN
      END
      SUBROUTINE SOLVBS(C,V,NB,LL,NLL)
C
C     SUBROUTINE FOR THE SOLUTION OF BANDED SYMMETRIC SYSTEM
C
C     OF LINEAR EQUATIONS. THE  COEFFICIENTS ARE STORED IN
C     VECTOR FORM IN V, THE LENGTH OF EACH ROW OF COEFFICIENTS
C     IN V IS GIVEN BY THE VECTOR N
C
      DIMENSION V(1),NB(1),C(LL,NLL)
      DOUBLE PRECISION TEMP,TEMP2
C
      ID=1
      DO 10 I=1,LL
      TEMP = V(ID)
      NEB = ID+NB(I)-1
C
C     NORMALISE ROW I
C
      DO 20 J= ID,NEB
   20 V(J) = V(J)/TEMP
      DO 25 L = 1,NLL
      C(I,L) = C(I,L)/TEMP
   25 CONTINUE
C
C     ELIMINATION
C
      ID1 = ID+1
      IDJ = ID
      IF(ID1.GT.NEB) GO TO 35
      DO 30 J= ID1,NEB
      JI = J-ID1+I
      IDJ = IDJ + NB(JI)
      IF (V(J)) 50,30,50
   50 TEMP2 = TEMP*V(J)
      IDP = J
      DO 40 K = IDP,NEB
      KJ = IDJ+K-J
   40 V(KJ) = V(KJ)-V(K)*TEMP2
      NJ = I+J-ID
      DO 32 L = 1,NLL
      C(NJ,L) = C(NJ,L)-C(I,L)*TEMP2
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID = ID+NB(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID = ID-1
      LL1 = LL-1
      DO 70 IB = 1,LL1
      I = LL1-IB+1
      ID = ID-NB(I)
      IS = I+1
      IN = I+NB(I)-1
      DO 80 J=IS,IN
      NJ = ID+J-I
C
      DO 75 L= 1,NLL
      C(I,L) = C(I,L) - C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE SETARG(AK,BK,PM,AGP,BGP,PMGP,AKK,BKK,PKK,IPROP,I)
C
C     IDENTIFIES THE PROPERTIES OF EACH ELEMENT.
C
      DIMENSION IPROP(1)
      IF(IPROP(I).EQ.0) GO TO 20
      AKK=AK
      BKK=BK
      PKK=PM
      GO TO 10
```

```
   20 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
   10 CONTINUE
      RETURN
      END
      SUBROUTINE VCHECK(NOD,H,X,ZEN,IDARCY,VEL,NN,AK,BK,PK,VCR,I)
C
C     COMPUTES ELEMENT VELOCITIES AND CHECKS IF A PARTICULAR ELEMENT BELONGS
C     TO DARCY OR NON-DARCY FLOW ZONE.
C
      DIMENSION NOD(150,3),X(200,1),ZEN(6),IDARCY(1),VEL(150,6)
      DIMENSION XEN(6),DSHF(6),SHF(6)
      DIMENSION H(200),HE(6)
      DO 3100 J=1,NN
      NJ=NOD(I,J)
      HE(J)=H(NJ)
      XEN(J)=X(NJ,1)
 3100 CONTINUE
      DO 3200 J=1,NN
      ZE=ZEN(J)
      CALL SHFUN1(XEN,J,ZE,NN,XE,DSHF,TJ,SHF)
      GRAD=0.0
      DO 3300 K=1,NN
      GRAD=GRAD+DSHF(K)*HE(K)/TJ
 3300 CONTINUE
      HS=ABS(GRAD)
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(I,J)=-HS/TEMP
      IF(IDARCY(I).EQ.0) VEL(I,J)=-HS*PK
 3200 CONTINUE
      IDARCY(I)=0
      IF(ABS(VEL(I,3)).GT.VCR) IDARCY(I)=1
      RETURN
      END
      SUBROUTINE VAVEG1(VMEAN,VEL,NOD,NNODE,NELEM,NN)
C
C     COMPUTES AVERAGE NODAL VELOCITIES.
C
      DIMENSION NOD(150,6),VMEAN(200,1),VEL(150,6),ICOUNT(200)
      DATA NREAD,NPRINT/1,3/
C
      WRITE(NPRINT,43)
   43 FORMAT('1',//,10X,'NODAL VELOCITY PRINT OUT',///,
     110X,'NODE',14X,'COMMON ELEMENTS',14X,'VELOCITY',///)
      DO 10 J=1,NNODE
      VMEAN(J,1)=0.0
      ICOUNT(J)=0
   10 CONTINUE
      DO 20 K=1,NELEM
      DO 30 J=1,NN
      NJ=NOD(K,J)
      VMEAN(NJ,1)=VMEAN(NJ,1)+VEL(K,J)
      ICOUNT(NJ)=ICOUNT(NJ)+1
   30 CONTINUE
   20 CONTINUE
      DO 40 J=1,NNODE
      DIV=ICOUNT(J)
      VMEAN(J,1)=VMEAN(J,1)/DIV
      WRITE(NPRINT,13) J,DIV,VMEAN(J,1)
   13 FORMAT(10X,I3,20X,F3.1,17X,F10.5)
   40 CONTINUE
      RETURN
      END
      SUBROUTINE DISCH1(TH,VMEAN,X,NSELCT,NFNOD,II)
C
C     COMPUTES DISCHARGE AT SELECTED VERTICAL SECTIONS.
C
      DIMENSION VMEAN(200,1),X(200,1),QAV(1),Q(200)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,43)
   43 FORMAT('1',//,5X,'NODE NO',20X,'DISCHARGE',//)
      CONST=6.2832
      NLNOD=NFNOD+NSELCT-1
      QAV(II)=0.0
      DO 10 J=NFNOD,NLNOD
      Q(J)=CONST*X(J,1)*TH*ABS(VMEAN(J,1))
      QAV(II)=QAV(II)+Q(J)
      WRITE(NPRINT,13)J,Q(J)
   13 FORMAT('0',10X,I3,20X,F10.4)
   10 CONTINUE
      QAV(II)=QAV(II)/NSELCT
      WRITE(NPRINT,23)II,QAV(II)
   23 FORMAT(////,10X,'MEAN DISCHARGE FOR THE REGION NO',I3,' =',F10.4)
      CONTINUE
      RETURN
      END
      SUBROUTINE TCURV1(H,HO,B,X,NNODE,QFIX,AK,BK,PMK,TH,J)
C
```

```
C     COMPUTES TYPE CURVE FOR STEADY STATE FLOW.
C
      DIMENSION H(1),X(200,1)
      TWPI=44./7.
      TMIS=PMK*TH
      PARAM=BK*QFIX*TMIS/(TH**2*B*TWPI)
   13 FORMAT(10X,I5,20X,F10.2,2(10X,E12.4))
      DENOM=QFIX/(TWPI*TMIS)
      WRITE(3,3)PARAM
    3 FORMAT(///,10X,50('*'),/,10X,'*',13X,'STEADY STATE TYPE CURVE',
     1 12X,'*',/,10X,50('*'),///,15X,'NON-LINEAR FACTOR    =',F12.4)
      IF(J.EQ.2) GO TO 5
      WRITE(3,23)
      GO TO 15
    5 CONTINUE
      WRITE(3,33)
   15 CONTINUE
   23 FORMAT(//,10X,'NODE NUMBER',15X,'R-COORDINATE',10X,'FUNCTION W(U)'
     1            ,10X,'ARGUMENT U',//)
   33 FORMAT(//,10X,'NODE NUMBER',15X,'Z-COORDINATE',10X,'FUNCTION W(U)'
     1            ,10X,'ARGUMENT U',//)
      DO 10 I=1,NNODE
      SDRAW=HO-H(I)
      SLESS=SDRAW/DENOM
      RLESS=X(I,1)/B
      WRITE(3,13)I,X(I,J),SLESS,RLESS
   10 CONTINUE
      RETURN
      END
```

4.2 LISTING OF TRCON1

```
C
C     INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C     DEVELOPED BY P.S. HUYAKORN.
C
C     TRCON1,PROGRAM FOR ANALYSING TRANSIENT,ONE-DIMENSIONAL,DARCY OR
C
C     TWO-REGIME FLOW TOWARDS A PUMPED WELL.
C
C     VERSION DATED OCTOBER,1973.
C
C      FOR FURTHER INFORMATION,CONTACT
C
C        P.S. HUYAKORN OR C.R. DUDGEON
C
C        WATER RESEARCH LABORATORY
C
C        KING ST.,MANLY VALE
C
C        SYDNEY,N.S.W. 2093,AUSTRALIA.
C
C-----------------------------
C     LIST OF INPUT VARIABLES
C-----------------------------
C
C***  PROBLEM VARIABLES  ***
C
C     NPROB   =  NUMBER OF PROBLEMS TO BE SOLVED
C     IVEL    =  VELOCITY PRINT-OUT INDEX.FEED IN IVEL=0 IF VELOCITY PRINT-OUT
C                IS NOT REQUIRED OTHERWISE FEED IN IVEL=1
C     IDISCR  =  DISCRETISATION DATA PRINT-OUT INDEX
C                FEED IN IDISCR=0 IF DISCRETISATION PRINT-OUT IS NOT REQUIRED
C                OTHERWISE FEED IN IDISCR=1
C     ORELAX  =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                SUGGESTED VALUE LIES BETWEEN 1.50 TO 1.85
C     RW      =  RADIUS OF WELL SCREEN
C     RO      =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C     HO      =  INITIAL HEIGHT OR DRAWDOWN OF WATER TABLE
C     TH      =  THICKNESS OF AQUIFER
C     HTOL    =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C     QFIX    =  PRESCRIBED WELL DISCHARGE
C     IWAT    =  WATER TABLE AQUIFER INDEX,IWAT=1 FOR WATER TABLE AQUIFER,
C                IWAT=0 FOR CONFINED AQUIFER
C     IAQTA   =  AQUITARD INDEX,IAQTA=1 IF THERE IS AN OVERLYING AQUITARD
C                IAQTA=0 IF THE MAIN AQUIFER IS CONFINED BY IMPERMEABLE STRATA
C     IGP     =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                IGP=0 FOR NON-GRAVEL PACKED WELL
C     IBOUND  =  EXTERNAL RADIUS BOUNDARY INDEX,IBOUND=1 FOR RECHARGE
C                BOUNDARY,IBOUND=0 FOR BARRIER BOUNDARY
C     IWBC    =  WELL DISCHARGE BOUNDARY CONDITION INDEX,IWBC=1 IF WELL STORAGE
C                IS TO BE TAKEN INTO ACCOUNT OTHERWISE IWBC=0
C     RCSNG   =  RADIUS OF WELL CASING
C     QRTOL   =  RATIO OF PRESCRIBED DISCHARGE TOLERANCE TO PRESCRIBED DISCHARGE
C                SUGGESTED VALUE LIES BETWEEN 0.01 TO 0.02
C     AK      =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                FEED IN AK=1./PM IF ONLY DARCY FLOW SOLUTION IS REQUIRED
C                WHERE PM = COEFFICIENT OF PERMEABILITY OF AQUIFER
C     BK      =  NON-LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                IF ONLY DARCY FLOW SOLUTION IS REQUIRED SET BK=0.
C     VCR     =  CRITICAL FLOW VELOCITY WHERE NON-DARCY FLOW COMMENCES
C     SS      =  COEFFICIENT OF SPECIFIC STORAGE OF MAIN AQUIFER
C     THGP    =  THICKNESS OF GRAVEL PACK
C     AGP     =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     BGP     =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     VGP     =  CRITICAL FLOW VELOCITY OF GRAVEL PACK MATERIAL
C     SSP     =  COEFFICIENT OF SPECIFIC STORAGE OF GRAVEL PACK MATERIAL
C     SY      =  COEFFICIENT OF SPECIFIC YIELD OF WATER TABLE AQUIFER
C     DINDEX  =  RECIPROCAL OF DELAYED YIELD INDEX OF WATER TABLE AQUIFER
C     PA      =  COEFFICIENT OF PERMEABILITY OF OVERLYING AQUITARD
C     THA     =  THICKNESS OF OVERLYING AQUITARD
C
C***  DISCRETISATION PARAMETERS  ***
C
C     NTICR   =  TOTAL NUMBER OF TIME STEPS
C     TFACTR  =  VALUE OF THE FIRST TIME STEP,EXPRESSED IN DIMENSIONLESS FORM
C     TMUL    =  TIME MULTIPLIER,SUGGESTED VALUE VARIES FROM 1.50 TO 2.00
C     DTMUL   =  INCREMENT OF TIME MULTIPLIER
C                SUGGESTED VALUE IS DTMUL=0. OR 0.02
C     NELF    =  NUMBER OF 3-NODE LINE ELEMENTS IN FIRST SUBREGION
C                SUGGESTED VALUE IS NELF=2
C     FRLEN   =  LENGTH OF FIRST SUBREGION
C                SUGGESTED VALUE IS FRLEN=4.*RW
C                FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
C     SCFAC   =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                SUBREGIONS.SUGGESTED VALUE IS SCFAC=2.0
C
C
C-----------------------------
C     LIST OF OUTPUT VARIABLES
C-----------------------------
```

```
C     NNODE    =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C     NELEM    =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C     IT       =  TIME STEP NUMBER
C     THM      =  REAL TIME VALUE AT THE MID-POINT OF TIME STEP IT
C     TM       =  REAL TIME VALUE AT THE END OF TIME STEP IT
C     X        =  NODAL RADIAL DISTANCES FROM CENTRE LINE OF PUMPED WELL
C     H        =  NODAL HEAD OR DRAWDOWN VALUES
C     TLESS    =  VALUES OF DIMENSIONLESS TIME AT NODAL POINTS,TLESS CORRESPONDS
C                 TO THE ARGUMENT 1/U OF WELL FUNCTION W(U)
C     HLESS    =  VALUES OF WELL FUNCTION AT NODAL POINTS
C     IQ       =  DISCHARGE ITERATION NUMBER
C     SW       =  DRAWDOWN VALUE AT CURRENT TIME
C     QAQFR    =  DISCHARGE FROM AQUIFER INTO WELL AT DRAWDOWN SW
C     QSTRGE   =  DISCHARGE FROM WELL STORAGE
C     QCALL    =  TOTAL CALCULATED DISCHARGE,QCALL=DISCHARGE FROM WELL STORAGE
C                 +QAQFR
C     QRDIF    =  RESIDUAL DISCHARGE,QRDIF = THE ABSOLUTE DIFFERENCE BETWEEN
C                 PRESCRIBED DISCHARGE AND CALCULATED DISCHARGE
C     NOD      =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C     VEL      =  ELEMENT VELOCITIES
C     VMEAN    =  AVERAGE NODAL VELOCITIES
C     IDARCY   =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                 OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                 IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      COMMON /BDISC/M(50),XLEN(50),NNE(50),JBD(2),DISP(2)
      COMMON /AELEM/DELK(3,3),SELK(3,3),HE(3),ZEN(3)
     1,DSHF(3),XEN(3),SHF(3)
      COMMON /ASOLV/ISTART(100),NBAND(100),ID(100)
      COMMON /BSOLV/CK(100,1),VK(300)
      COMMON /CSOLV/VD(300),HINT(100,1),GK(100,1),CTEMP(100,1)
      COMMON /APARA/TLESS(100),HLESS(100),TIME(60),SW(60)
      COMMON /BPARA/DSW(3),QCALC(3),VS(10),QEL(6),GELK(6)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      COMMON /AFLXX/FLEAK(100),QLEAK(100),HCORE(900),HCOLD(100)
C
      DOUBLE PRECISION ARGU,TERM,DEXP,EN,U,HWF
      DATA NREAD,NPRINT/1,3/
C
C     PRINT INITIAL HEADINGS
C
      WRITE(NPRINT,1773)
 1773 FORMAT('1')
      WRITE(NPRINT,1003)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1023)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1033)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1003)
  993 FORMAT('1',4X,51H **************************************************
     1**)
 1003 FORMAT(5X,51H **************************************************)
 1013 FORMAT(5X,51H *                                                *)
 1023 FORMAT(5X,51H *     FINITE ELEMENT ANALYSIS OF WELL PROBLEMS     *)
 1033 FORMAT(6X,'* TRANSIENT,TWO REGIME FLOW TOWARD A SINGLE WELL *')
```

```
      READ(NREAD,101)NPROB,IVEL,IDISCR,ORELAX                           $1
  101 FORMAT(3I10,F10.2)
      DO 3555 JPRO=1,NPROB
      WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1              I6,12X,'*',/,20X,50('*'))
C
C     READ AND PRINT GENERAL DATA
C
      READ(NREAD,2003)RW,RO,HO,TH,HTOL                                  $1
 2003 FORMAT(5F10.2)
      WRITE(NPRINT,23) RW,RO,HO,TH,HTOL
   23 FORMAT(///,20X,32HAQUIFER AND WELL CHARACTERISTICS,///,
     310X,19HRADIUS OF WELL   = ,F10.3,//,
     410X,22HRADIUS OF INFLUENCE = ,F10.3,//,
     510X,24HHEIGHT OF WATER TABLE = ,F10.3,//,
     610X,24HTHICKNESS OF AQUIFER  = ,F10.3,//,
     710X,39HTOLERANCE FOR SUCCESSIVE HEAD VALUES = ,F10.3,//)
      READ(NREAD,51)QFIX                                                $1
   51 FORMAT(F10.2)
      WRITE(NPRINT,2773)QFIX
 2773 FORMAT(10X,'PRESCRIBED WELL DISCHARGE  =',F12.3)
C
C     READ AQUIFER AND GRAVEL PACK DATA
C
      READ(NREAD,111)IWAT,IAQTA,IGP,IBOUND,IWBC                         $1
  111 FORMAT(5I10)
      IQUASI=0
      IF(IWBC.EQ.0) QRTOL=0.0
      IF(IWBC.EQ.0) GO TO 309
      READ(NREAD,191)RCSNG,QRTOL                                        $1,SK
  191 FORMAT(2F10.3)
      QRTOL=QRTOL*QFIX
      WRITE(NPRINT,193)RCSNG,QRTOL
  193 FORMAT(///,10X,'RADIUS OF CASING   =',F10.3,//,
     1          10X,'DISCHARGE TOLERANCE           =',F10.3,//)
  309 CONTINUE
      READ(NREAD,2013)AK,BK,VCR,SS                                      $1
 2013 FORMAT(3F10.3,E10.2)
      GCR=AK*VCR+BK*VCR**2
      PM=1./AK
      IF(VCR.GT.0.000001) PM=VCR/GCR
      WRITE(NPRINT,53)AK,BK,VCR,PM,SS
      SS=SS*1000.
   53 FORMAT(///,20X,'AQUIFER PROPERTIES'///
     1          10X,'COEFFICIENT A =',F10.4//
     2          10X,'COEFFICIENT B = ',F10.4//
     3          10X,'CRITICAL VELOCITY = ',F10.4//
     4          10X,'COEFFICIENT K = ',F10.4//
     5          10X,'SPECIFIC STORAGE  =',E10.2)
      RGP=RW
      IF(IGP.EQ.0) GO TO 29
      READ(NREAD,331)AGP,BGP,VGP,SSP,THGP                               $1,SK
  331 FORMAT(3F10.3,E10.2,F10.2)
      GRGP=AGP*VGP+BGP*VGP**2
      PMGP=1./AGP
      IF(VCR.GT.0.000001) PMGP=VGP/GRGP
      WRITE(NPRINT,153)AGP,BGP,VGP,PMGP,SSP
      SSP=SSP*10.0**3
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1          10X,'COEFFICIENT A =',F10.4//
     2          10X,'COEFFICIENT B = ',F10.4//
     3          10X,'CRITICAL VELOCITY = ',F10.4//
     4          10X,'COEFFICIENT K = ',F10.4//
     5          10X,'SPECIFIC STORAGE  =',E10.2)
      RGP=RW+THGP
      WRITE(NPRINT,163)THGP,RGP
  163 FORMAT(10X,'THICKNESS OF PACK = ',F10.2//
     1          10X,'RADIUS OF PACK = ',F10.2//)
   29 CONTINUE
C
C     READ BOULTON SPECIFIC YIELD DATA FOR UNCONFINED AQUIFER.
C
      IF(IWAT.EQ.0) GO TO 96
      READ(NREAD,171)SY,DINDEX                                          $1,SK
  171 FORMAT(2E10.2)
      SFAC=DINDEX*SY/(6.*TH)
      WRITE(NPRINT,173)SY,DINDEX,SFAC
      DINDEX=DINDEX/1000.
  173 FORMAT(///,20X,'BOULTON DELAY YIELD DATA',///,
     1          10X,'SPECIFIC YIELD    =',E20.3,//,
     2          10X,'DELAY INDEX     =',E20.3,//,
     3          10X,'DELAY FACTOR    =',E20.3,//)
   96 CONTINUE
C
C     READ AND PRINT AQUITARD DATA.
C
      IF((IAQTA.EQ.0).AND.(IQUASI.EQ.0)) GO TO 39
      READ(NREAD,71)PA,THA                                              $1,SK
   71 FORMAT(F10.4,F10.2)
```

```
      SA=0.
      WRITE(NPRINT,273)PA,SA,THA
  273 FORMAT(///,20X,'AQUITARD DATA',///,
     1           10X,'HYDRAULIC CONDUCTIVITY    =',F20.4//
     2           10X,'SPECIFIC STORAGE     =',E20.4//
     3           10X,'AQUITARD THICKNESS    =',F10.3)
      SA=SA*1000.
      BFAC=PA/THA
   39 CONTINUE
C
C
C     GENERATE AND PRINT DISCRETIZATION PARAMETER.
C
      READ(NREAD,121)NTICR,TFACTR,TMUL,DTMUL                          $1
  121 FORMAT(I10,3F10.3)
      ITST=1
      NNODS=1
      WRITE(NPRINT,703)NTICR,TFACTR,TMUL,DTMUL
  703 FORMAT(///,20X,'GENERAL TIME DATA',//,
     1           10X,'NUMBER OF TIME INCREMENTS  =',I10,//,
     1          10X,'TIME FACTOR                =',F10.4,//,
     2          10X,'TIME MULTIPLIER            =',F10.4,//,
     3          10X,'INCREMENT OF MULTIPLIER    =',F10.4)
C
C     GENERATE ELEMENTS OF VECTOR TIME.
C
      CALL TIGEN(NTICR,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX)
      READ(NREAD,391)NELF,SCFAC,FRLEN                                 $1
  391 FORMAT(I10,2F10.2)
      CALL DCRGEN(RO,RW,NELF,SCFAC,FRLEN,NRR)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,33)NELF,SCFAC,FRLEN
   33 FORMAT(3(/),20X,'DISCRETISATION DATA',///,
     1         10X,'NUMBER OF ELEMENTS IN FIRST SUBREGION  =',I10,///,
     2           10X,'SCALE FACTOR FOR ELEMENT LENGTH    =',F10.3,///,
     3           10X,'LENGTH OF FIRST SUBREGION    =',F10.3,///,2X,
     4'    SUBREGION  ',10X,'NO. OF ELEMENTS',15X,'LENGTH OF SUBREGION',
     5    //)
C
   31 FORMAT(I3)
      DO 55 I=1,NRR
   41 FORMAT(2I3,F8.2)
      WRITE(NPRINT,43) I,M(I),XLEN(I)
   43 FORMAT(5X,I3,20X,I3,16X,F14.3)
   55 CONTINUE
   94 CONTINUE
      RI=RW
      NFNOD=1
      NNODE=0
      NELEM=0
      NSEL=1
C
C
      DO 65 K=1,NRR
      NNODE=NNODE+(NNE(K)-1)*M(K)
      NELEM=NELEM+M(K)
   65 CONTINUE
      NNODE=NNODE+1
C
C     PRINT OUT NO OF NODES AND NO OF ELEMS
C
      IF(IDISCR.EQ.0) GO TO 104
      WRITE(NPRINT,73) NNODE,NELEM
   73 FORMAT(3(/),12X,'TOTAL NO. OF NODES = ',I5,//,12X,
     1'TOTAL NO. OF ELEMENTS = ',I5)
  104 CONTINUE
      DO 75 II=1,NRR
      MM=M(II)
      NN=NNE(II)
      XL=XLEN(II)
C
C      GENERATE THE NODE COORDINATES
C
      CALL CORDC1(XL,MM,NN,RI,NFNOD)
C
C     DISCRETIZE THE REGION INTO ELEMS
C
      CALL CONC1(NSEL,NFNOD,MM,NN)
      NFNOD=NFNOD+(NNE(II)-1)*M(II)
   75 CONTINUE
      CALL HYPROP(NELEM,RGP)
      IF(IDISCR.EQ.0) GO TO 114
      DXO=(X(2,1)-X(1,1))**2
      ALPHA=SS*DXO**2/(2.0*PM*TIME(1))
      WRITE(NPRINT,7773) ALPHA
 7773 FORMAT( //,12X,'MESH STABILITY FACTOR    =',E20.4)
  114 CONTINUE
      ROMAX=RO
      TWPI=44./7.
      QFL=QFIX/(TWPI*TH)
```

```
C
C     FORM GROSS VECTOR VD
C
      NN=3
      CALL EBFIND(LEN,NELEM,NN,NNODE)
C
      ZEN(1)=-1.0
      ZEN(2)=1.0
      ZEN(3)=0.0
      DO 130 I=1,LEN
      VD(I)=0.0
  130 CONTINUE
      DO 135 IE=1,NELEM
      DO 145 J=1,NN
      NJ=NOD(IE,J)
      XEN(J)=X(NJ,1)
  145 CONTINUE
      SC=SS
      IF(IPROP(IE).EQ.0) SC=SSP
      CALL ELGND(NN,IE,SC,0)
      CALL MERBD(NN,IE)
  135 CONTINUE
C
C     CALCULATE HEAD AT VARIOUS TIME VALUES
C
      TMIS=TH*PM
      DIFFUS=PM/SS
      NST=2
      DO 642 I=1,NNODE
      HCOLD(I)=HO
      FLEAK(I)=0.0
      QLEAK(I)=0.0
      H(I)=HO
  642 CONTINUE
      IF(ITST.EQ.1) GO TO 1245
      CALL HREAD(NNODS,1,1)
      CALL HPUNCH(NNODS,3,1)
      IF(IWAT.GT.0) CALL HREAD(NNODS,1,2)
      IF(IWAT.GT.0) CALL HPUNCH(NNODS,3,2)
      IST=NNODS+1
      DO 1345 I=IST,NNODE
 1345 H(I)=HO
      GO TO 1445
 1245 CONTINUE
      DO 3545 I=1,NNODE
 3545 H(I)=HO
 1445 CONTINUE
      DO 245 I=1,NNODE
      HINT(I,1)=HO
  245 CONTINUE
      DELT=TIME(1)
      NDTO=NNODE
      ITMIN=NTICR
      QAQFR=2.*QRTOL
      SWOLD=0.0
      IST=1
C
C     SET IDARCY=0
C
      DO 99 I=1,NELEM
      IDARCY(I)=0
   99 CONTINUE
C
C     LOOPING WITH LOOP PARAMETER IT=1,NTICR
C
      DO 7007 IT=ITST,NTICR
      WRITE(NPRINT,683)IT
  683 FORMAT(///,10X,35('*'),/,10X,'*',4X,
     1             'TIME STEP NUMBER  =',I5,5X,'*',/,10X,35('*'))
      ITCUR=IT
      IF(IT.GT.1) DELT=TIME(IT)-TIME(IT-1)
      TM=TIME(IT)
      CALL ROEST(DIFFUS,TM,NDTO,QFIX,TMIS,NST,NN,RO,NELEM,NNODE,CUNST)
      TMM=TM-DELT*0.5
      MBAND=NBAND(NNODE)
      TMIL=TMM/1000.
      WRITE(NPRINT,333)TMIL
  333 FORMAT(///,10X,41('*'),/,10X,'*',9X,'TIME  =',
     1             E14.3,9X,'*',/,10X,41('*'),//)
      WRITE(NPRINT,343)RO,NNODE,NELEM
  343 FORMAT(/,10X,'ESTIMATED RADIUS OF INFLUENCE  =',F10.2,
     1//,10X,'CORRESPONDING NO. OF NODES  =',I5,//,
     210X,'CORRESPONDING NO. OF ELEMENTS  =',I5,//)
C
C     COMPUTE DIMENSIONLESS TIME.
C
      CONS=4.0*DIFFUS*TMM
      DO 709 I=1,NNODE
      TLESS(I)=CONS/X(I,1)**2
  709 CONTINUE
```

```
C
C     FOR UNCONFINED AQUIFER,COMPUTE BOULTON DELAYED YIELD.
C
      IF(IWAT.EQ.0)GO TO 777
      ARGU=-DINDEX*DELT*0.5
      TERM=DEXP(ARGU)
      ES=SY*(1.0-TERM)
      ES=ES/TH
      WRITE(NPRINT,903)ES
      ES=ES*1000.
      EN=ES/SS
  903 FORMAT(///,'  EFFECTIVE SPECIFIC YIELD  =',E14.4,5X,'/FT.')
  777 CONTINUE
C
C     INITIALISE HEAD VALUES.
C
      IF(IT.EQ.1) GO TO 165
  677 CONTINUE
      DO 155 I=1,NNODE
      HINT(I,1)=H(I)
  155 CONTINUE
  165 CONTINUE
      LEN=NBAND(1)
      NBAND(NNODE)=1
      DO 169 I=2,NNODE
      LEN=LEN+NBAND(I)
  169 CONTINUE
      CALL VECMUL(NNODE,LEN)
C
C     FOR UNCONFINED AQUIFER,ADD BOULTON DELAYED YIELD
C
      IF(IWAT.EQ.0) GO TO 278
      DO 178 I=1,NNODE
      GK(I,1)=GK(I,1)*(1.+EN)
  178 CONTINUE
      IF(IT.EQ.1) GO TO 278
      FTERM=0.5*DELT*SFAC/SS
      DO 1178 I=1,NNODE
 1178 CK(I,1)=QLEAK(I)
      CALL VMUL(NNODE,LEN)
      DO 1278 I=1,NNODE
 1278 GK(I,1)=GK(I,1)+CTEMP(I,1)*FTERM
  278 CONTINUE
C
C     STORE FIRST ELEMENT OF MATRIX GK FOR DISCHARGE COMPUTATION.
C
      GP=GK(1,1)
      IF(IWBC.EQ.0) GK(1,1)=GK(1,1)-DELT*QFL*0.5
C
C     SET UP LOOP FOR WELL DISCHARGE ITERATION.
C
      NQITER=5
      DO 998 IQ=1,NQITER
      IF(IWBC.EQ.0) GO TO 98
C
C     ADJUST VALUE OF WELL DRAWDOWN.
C
      WRITE(NPRINT,5003)
 5003 FORMAT(/,5X,30('- '))
      CALL SWMOD(IQ,IT,RCSNG,DELT,TIME,QFIX,QAQFR,QCALC,
     1SW,DSW,SWOLD)
      WRITE(NPRINT,833)IQ,DSW(IQ),SW(IT)
  833 FORMAT(///,10X,'DISCHARGE ITERATION NUMBER   =',I10,//,
     1           10X,'DRAWDOWN INCREMENT    =',F10.3,//,
     2           10X,'DRAWDOWN VALUE    =',F10.3,//)
   98 CONTINUE
C
C     SET UP LOOP FOR ITERATION ON HEAD
C
      NITER=10
      IF(BK.LE.0.0) NITER=1
      DO 999 III=IST,NITER
C
C     ZERO GROSS STIFFNESS MATRIX
C
      DO 300 I=1,LEN
      VK(I)=0.0
  300 CONTINUE
      DO 305 I=1,NNODE
      CK(I,1)=GK(I,1)
  305 CONTINUE
C
C     COMPUTE STIFFNESS MATRIX FOR EACH ELEM AND MERGE
C
      NN=3
      DO 350 IE=1,NELEM
      DO 355 J=1,NN
      NJ=NOD(IE,J)
      XEN(J)=X(NJ,1)
      IF(III.EQ.1) GO TO 355
```

```
      HE(J)=H(NJ)
  355 CONTINUE
      IF(IPROP(IE).EQ.0) GO TO 755
      AKK=AK
      BKK=BK
      PKK=PM
      GO TO 365
  755 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
  365 CONTINUE
      CALL ELGEN1(NN,AKK,BKK,PKK,IE,0)
      CALL MERB3(NN,IE)
  350 CONTINUE
      DO 530 I=1,LEN
      VK(I)=VK(I)*DELT*0.5+VD(I)
  530 CONTINUE
C
C     ADD LEAKAGE TERMS CONTRIBUTED BY AQUITARD.
C
      IF(IAQTA.EQ.0) GO TO 86
      FTERM=0.5*DELT*BFAC/(SS*TH)
      DO 106 I=1,LEN
      VK(I)=VK(I)+VD(I)*FTERM
  106 CONTINUE
   86 CONTINUE
C
C     FOR UNCONFINED AQUIFER,ADD BOULTON DELAYED YIELD.
C
      IF(IWAT.EQ.0) GO TO 196
      DO 606 I=1,LEN
      VK(I)=VK(I)+VD(I)*EN
  606 CONTINUE
  196 CONTINUE
      IF((RO.EQ.ROMAX).AND.(IBOUND.EQ.0))GO TO 296
      NBD=1
      JBD(1)=NNODE
      DISP(1)=HO
      CALL PBC(NBD)
  296 CONTINUE
C
C     FIX THE WELL DRAWDOWN.
C
      NLEN=NBAND(1)
      DO 378 I=1,NLEN
      VS(I)=VK(I)
  378 CONTINUE
      IF(IWBC.EQ.0) GO TO 496
      JBD(1)=1
      DISP(1)=SW(IT)+HO
      CALL PBC(NBD)
  496 CONTINUE
C
C     SOLVE EQNS BY BANDED ELIMINATION SCHEME
C
      NLL= 1
      CALL SOLVBS(NNODE,NLL)
C
C     PRINT OUT SOLUTION FOR FUNCTION
C     ONLY THE DARCY AND FINAL SOLUTIONS
C
      IF(III.EQ.1) GO TO 900
  415 CONTINUE
      EMAX=0.0
      NCOUNT=0
      DO 450 I=1,NNODE
      EPSI=CK(I,1)-H(I)
      EPSIS=EPSI
      IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.HTOL)  GO TO 460
      NCOUNT=NCOUNT+1
      IF((IST.EQ.2).AND.(III.EQ.2)) GO TO 460
      H(I)=H(I)+ORELAX*EPSIS
      GO TO 450
  460 CONTINUE
      H(I)=CK(I,1)
  450 CONTINUE
      IF(NCOUNT.EQ.0) GO TO 1000
      GO TO 1999
  900 CONTINUE
      DO 950 I=1,NNODE
      H(I)=CK(I,1)
      HLESS(I)=(HO-H(I))/CUNST
  950 CONTINUE
 1999 CONTINUE
      IF(NITER.EQ.1) GO TO 999
C
C     REGENERATE IDARCY
C
```

```
      DO 199 I=1,NELEM
      CALL SETARG(AK,BK,PM,AGP,BGP,PMGP,AKK,BKK,PKK,I)
      CALL VCHECK(NN,AKK,BKK,PKK,VCR,I)
  199 CONTINUE
      IF(IDARCY(1).EQ.0) GO TO 1000
  999 CONTINUE
 1000 CONTINUE
      WRITE(NPRINT,413)III,EMAX
  413 FORMAT(//,10X,'NO. OF ITERATIONS REQUIRED FOR FINAL SOLUTION =',I5
     1         ,//,10X,'MAXIMUM ABSOLUTE ERROR IN HEAD =',F10.3)
C
C     CHECK FOR ACCURACY OF DISCHARGE RATIO.
C
      IF(IWBC.EQ.0) GO TO 2102
      DWD=DSW(IQ)
      CALL AQDIS(GP,NLEN,QAQFR,QCALC,QRDIF,DELT,TH,RCSNG,DSW,IQ,QFIX)
C
C
C     PRINT FINAL DISCHARGE VALUES.
C
      QCALL=QCALC(IQ)
      QSTRGE=QCALL-QAQFR
      WRITE(NPRINT,1203)QAQFR,QSTRGE,QCALL,QRDIF
 1203 FORMAT(///,10X,'DISCHARGE FROM AQUIFER INTO WELL   =',F12.4,//,
     1         10X,'DISCHARGE FROM WELL STORAGE =',F14.3,//,
     1          10X,'TOTAL CALCULATED DISCHARGE    =',F14.3,//,
     210X,'RESIDUAL DISCHARGE          =',F10.4)
      IF(QRDIF.LE.QRTOL) GO TO 1102
  998 CONTINUE
 1102 CONTINUE
      SWOLD=SW(IT)
 2102 CONTINUE
      DO 470 I=1,NNODE
      HLESS(I)=(HO-H(I))/CUNST
      IF((ITCUR.GT.IT).AND.(IQ.EQ.1)) TLESS(I)=TLESS(I)*TM/TMM
  470 CONTINUE
      CALL ROUT(NNODE)
 2010 CONTINUE
      NBAND(NNODE)=MBAND
      IF(IT.GT.ITMIN) GO TO 577
      IF(ITCUR.GT.IT) GO TO 7000
      TMIL=TM/1000.
      WRITE(NPRINT,333)TMIL
      ITCUR=IT+1
      GO TO 677
  577 CONTINUE
      TMIL=TM/1000.
      WRITE(NPRINT,333)TMIL
C
C     OBTAIN HEAD VALUES AT THE END OF TIME INTERVAL BY EXTRAPOLATION.
C
      DO 477 I=1,NNODE
      TLESS(I)=TLESS(I)*TM/TMM
      H(I)=2.0*H(I)-HINT(I,1)
      HLESS(I)=(HO-H(I))/CUNST
  477 CONTINUE
      IF(IWBC.EQ.0) GO TO 7009
      SWTEMP=SWOLD
      SWOLD=H(1)-HO
      QFR=QAQFR
      DWTEMP=SWOLD-SWTEMP
      QAQFR=QFR*SWOLD/SWTEMP
      ACSNG=22.*RCSNG**2/7.
      TRM=ABS(ACSNG*DWTEMP*2./DELT)
      QCALL=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALL)
      SW(IT)=SWOLD
      WRITE(NPRINT,1203)QAQFR,QCALL,QRDIF
      QWSTR=QCALL-QAQFR
      IF(ABS(QWSTR).LT.QRTOL) IWBC=0
 7009 CONTINUE
      CALL ROUT(NNODE)
C
C     EVALUATE NODAL VELOCITIES
C
      IF(IVEL.EQ.0) GO TO 7000
      DO 3000 I=1,NELEM
      CALL SETARG(AK,BK,PM,AGP,BGP,PMGP,AKK,BKK,PKK,I)
      CALL VCHECK(NN,AKK,BKK,PKK,VCR,I)
 3000 CONTINUE
      NN=3
      CALL VAVEG1(NNODE,NELEM,NN)
 7000 CONTINUE
C
C     FOR WATER TABLE AQUIFER,COMPUTE FIRST PORTION OF BOULTON'S INTEGRAL.
C
      IF(IWAT.EQ.0) GO TO 7007
      IF(IT.EQ.NTICR) GO TO 7007
      CALL BSIMP(NNODE,TH,DINDEX,SY,DELT,IT,ITMIN)
      DO 8007 I=1,NNODE
```

```
      HCOLD(I)=H(I)
 8007 CONTINUE
 7007 CONTINUE
C
C     PUNCH OUT SOLUTION AT FINAL TIME.
C
      CALL HPUNCH(NNODE,2,1)
      IF(IWAT.GT.0) CALL HPUNCH(NNODE,2,2)
 5000 CONTINUE
 3555 CONTINUE
      STOP
      END
      SUBROUTINE ELGEN1(NN,AK,BK,PKK,IE,IVERT)
C
C     GENERATES ELEMENT MATRIX SELK FOR 3-NODE ELEMENTS.
C
      DIMENSION ZGSP(3),WC(3)
      COMMON /AELEM/DELK(3,3),SELK(3,3),HE(3),ZEN(3)
     1,DSHF(3),XEN(3),SHF(3)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
C
      NGP=3
      ZGSP(1)=0.7745966692415
      ZGSP(2)=-ZGSP(1)
      ZGSP(3)=0.0
      WC(1)=5.0/9.0
      WC(2)=WC(1)
      WC(3)=8.0/9.0
      DO 15 I=1,NN
      DO 15 J=1,NN
      SELK(I,J)=0.0
   15 CONTINUE
      DO 10 I=1,NGP
      ZE=ZGSP(I)
      CALL SHFUN1(I,ZE,NN,XE,TJ)
      XC=XE
      IF(IVERT.NE.0) XC=1.0
      IF(IDARCY(IE).EQ.0) GO TO 25
      IF(IVERT.NE.0) GO TO 55
      HS=0.0
      DO 30 JJ=1,NN
      HS=HS+DSHF(JJ)*HE(JJ)
   30 CONTINUE
      HS=ABS(HS/TJ)
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      CONS=1./TEMP
      GO TO 35
   55 CONTINUE
      CONS=0.0
      DO 65 JJ=1,NN
   65 CONS=CONS+HE(JJ)*SHF(JJ)
      GO TO 35
   25 CONTINUE
      CONS=PKK
   35 CONTINUE
      DO 60 K=1,NN
      DO 50 L=1,NN
      IF(L-K) 50,40,40
   40 SELK(K,L)=SELK(K,L)+(DSHF(K)*DSHF(L)*XC*WC(I)*CONS)/TJ
   50 CONTINUE
   60 CONTINUE
   10 CONTINUE
      DO 70 K=1,NN
      DO 80 L=1,NN
      IF(L-K) 90,80,80
   90 SELK(K,L)=SELK(L,K)
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE SHFUN1(I,ZE,NN,XE,TJ)
C
C     GENERATES SHAPE FUNCTION MATRICES,SHF AND DSHF,FOR 3-NODE ELEMENTS.
C
      COMMON /AELEM/DELK(3,3),SELK(3,3),HE(3),ZEN(3)
     1,DSHF(3),XEN(3),SHF(3)
C
      ZSQ=ZE**2
      SHF(1)=0.5*(-ZE+ZSQ)
      SHF(2)=0.5*(ZE+ZSQ)
      SHF(3)=1.0-ZSQ
      DSHF(1)=-.5+ZE
      DSHF(2)=0.5+ZE
      DSHF(3)=-2.0*ZE
      TJ=0.0
      XE=0.0
      DO 10 J=1,NN
      TJ=TJ+DSHF(J)*XEN(J)
      XE=XE+SHF(J)*XEN(J)
   10 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE CORDC1(XL,MM,NN,RI,NFNOD)
C
C     GENERATES NODAL COORDINATES.
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      DXL=XL/(MM*(NN-1))
      NLNOD=NFNOD+(NN-1)*MM
      DO 10  J=NFNOD,NLNOD
      X(J,1)=RI+(J-NFNOD)*DXL
   10 CONTINUE
      RI=X(NLNOD,1)
      RETURN
      END
      SUBROUTINE CONC1(NSEL,NFNOD,MM,NN)
C
C     DISCRETISES THE REGION INTO 3-NODE LINE ELEMENTS.
C
      DIMENSION NDINT(3)
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
C
      NDINT(1)=NFNOD
      NDINT(2)=NFNOD+2
      NDINT(3)=NFNOD+1
      NLST=NSEL+MM-1
      DO 10  L=NSEL,NLST
      DO 20  K=1,NN
      NOD(L,K)=NDINT(K)+(L-NSEL)*(NN-1)
   20 CONTINUE
   10 CONTINUE
      NSEL=NLST+1
      RETURN
      END
      SUBROUTINE MERBD(N,M)
C
C     MERGES ELEMENT MATRIX DELK INTO GROSS VECTOR VD.
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      COMMON /AELEM/DELK(3,3),SELK(3,3),HE(3),ZEN(3)
     1,DSHF(3),XEN(3),SHF(3)
      COMMON /ASOLV/ISTART(100),NBAND(100),ID(100)
      COMMON /CSOLV/VD(300),HINT(100,1),GK(100,1),CTEMP(100,1)
      DO 10  I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20  J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VD(IPOS)=VD(IPOS)+DELK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MERB3(N,M)
C
C     MERGES ELEMENT MATRIX SELK INTO GROSS VECTOR VK.
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      COMMON /AELEM/DELK(3,3),SELK(3,3),HE(3),ZEN(3)
     1,DSHF(3),XEN(3),SHF(3)
      COMMON /ASOLV/ISTART(100),NBAND(100),ID(100)
      COMMON /BSOLV/CK(100,1),VK(300)
      DO 10  I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20  J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VK(IPOS)=VK(IPOS)+SELK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE EBFIND(LEN,NM,NN,LL)
C
C     COMPUTES BANDWIDTH FOR EACH ROW OF THE GROSS MATRIX.
C
      DIMENSION LV(3)
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      COMMON /ASOLV/ISTART(100),NBAND(100),ID(100)
      DO 20 I=1,LL
      NBAND(I)=1
      ISTART(I)=I
   20 CONTINUE
C
C     SCAN THROUGH THE LOCATION VECTOR FOR EACH MEMBER TO FIND
C     THE POSITION OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW
```

```
C
      NF2=3
      DO 25 I=1,NM
      DO 30 J=1,NF2
      LV(J)=NOD(I,J)
   30 CONTINUE
      DO 45 J=1,NF2
      IF(LV(J).EQ.0)GO TO 45
      DO 40 K=1,NF2
      IF(LV(J).GT.LV(K)) GO TO 40
      NW=LV(K)-LV(J)+1
      NR=LV(J)
      IF(NW.GT.NBAND(NR))  NBAND(NR)=NW
   40 CONTINUE
   45 CONTINUE
C       SEARCH FOR THE FURTHEREST OFF-LEFT TERM
C
      DO 55 J=1,NF2
      IF(LV(J).EQ.0) GO TO 55
      DO 65 K=1,NF2
      IF(LV(J).LT.LV(K)) GO TO 65
      NW=LV(K)
      NR=LV(J)
      IF(NW.LT.ISTART(NR)) ISTART(NR)=NW
   65 CONTINUE
   55 CONTINUE
   25 CONTINUE
C
C     SET UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN=NBAND(1)
      DO 50 I=2,LL
      IF(NBAND(I)-NBAND(I-1).LT.0)  NBAND(I)=NBAND(I-1)-1
      LEN=LEN+NBAND(I)
   50 CONTINUE
      ID(1)=1
      DO 60 I=2,LL
   60 ID(I)=ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE SOLVBS(LL,NLL)
C
C       SUBROUTINE FOR THE SOLUTION OF BANDED SYMMETRIC EQUATIONS.
C
      COMMON /ASOLV/ISTART(100),N(100),IDUM(100)
      COMMON /BSOLV/C(100,1),V(300)
      DOUBLE PRECISION TEMP,TEMP2
      ID=1
      DO 10 I=1,LL
      TEMP=V(ID)
      NEB=ID+N(I)-1
C
C     NORMALISE ROW I
C
      DO 20 J=ID,NEB
   20 V(J)=V(J)/TEMP
      DO 25 L=1,NLL
      C(I,L)=C(I,L)/TEMP
   25 CONTINUE
C
C     ELIMINATION
C
      ID1=ID+1
      IDJ=ID
      IF(ID1.GT.NEB)  GO TO 35
      DO 30 J=ID1,NEB
      JI=J-ID1+I
      IDJ=IDJ+N(JI)
      IF(V(J))50,30,50
   50 TEMP2=TEMP*V(J)
      IDP=J
      DO 40 K=IDP,NEB
      KJ=IDJ+K-J
   40 V(KJ)=V(KJ)-V(K)*TEMP2
      NJ=I+J-ID
      DO 32 L=1,NLL
      C(NJ,L)=C(NJ,L)-C(I,L)*TEMP2
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID=ID+N(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID=ID-1
      LL1=LL-1
      DO 70 IB=1,LL1
      I=LL1-IB+1
```

```
      ID=ID-N(I)
      IS=I+1
      IN=I+N(I)-1
      DO 80 J=IS,IN
      NJ=ID+J-I
      DO 75 L=1,NLL
      C(I,L)=C(I,L)-C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE PBC(NBD)
C
C     INTRODUCES PRESCRIBED BOUNDARY CONDITIONS.
C
      COMMON /ASOLV/ISTART(100),NBAND(100),ID(100)
      COMMON /BSOLV/CK(100,1),VK(300)
      COMMON /BDISC/M(50),XLEN(50),NNE(50),JBD(2),DISP(2)
      DO 400 I=1,NBD
      NR=JBD(I)
      CK(NR,1)=1.D+20*DISP(I)
      IDT=ID(NR)
      VK(IDT)=1.D+20
  400 CONTINUE
      RETURN
      END
      SUBROUTINE HYPROP(NELEM,RGP)
C
C     GENERATES PROPERTY INDEX FOR ALL ELEMENTS
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      DO 10 I=1,NELEM
      NJ=NOD(I,3)
      IPROP(I)=1
      IF(X(NJ,1).LT.RGP) IPROP(I)=0
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SETARG(AK,BK,PM,AGP,BGP,PMGP,AKK,BKK,PKK,I)
C
C     IDENTIFIES THE PROPERTIES OF EACH ELEMENT.
C
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      IF(IPROP(I).EQ.0) GO TO 20
      AKK=AK
      BKK=BK
      PKK=PM
      GO TO 10
   20 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
   10 CONTINUE
      RETURN
      END
      SUBROUTINE VCHECK(NN,AK,BK,PK,VCR,I)
C
C     COMPUTES ELEMENT VELOCITIES AND CHECKS IF A PARTICULAR ELEMENT BELONGS
C     TO DARCY OR NON-DARCY FLOW ZONE.
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      COMMON /AELEM/DELK(3,3),SELK(3,3),HE(3),ZEN(3)
     1,DSHF(3),XEN(3),SHF(3)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      DO 3100 J=1,NN
      NJ=NOD(I,J)
      HE(J)=H(NJ)
      XEN(J)=X(NJ,1)
 3100 CONTINUE
      DO 3200 J=1,NN
      ZE=ZEN(J)
      CALL SHFUN1(J,ZE,NN,XE,TJ)
      GRAD=0.0
      DO 3300 K=1,NN
      GRAD=GRAD+DSHF(K)*HE(K)/TJ
 3300 CONTINUE
      HS=ABS(GRAD)
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(I,J)=-HS/TEMP
      IF(IDARCY(I).EQ.0) VEL(I,J)=-HS*PK
 3200 CONTINUE
      IDARCY(I)=0
      IF(ABS(VEL(I,3)).GT.VCR) IDARCY(I)=1
      RETURN
      END
      SUBROUTINE ROEST(DIFFUS,TM,NDTO,QFIX,THIS,NST,NN,RO,NELEM,NNODE,
     1CONST)
C
```

```
C     ESTIMATES RADIUS OF INFLUENCE AT TIME TM.
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      FPI=88./7.
      CONST=QFIX/(FPI*TMIS)
      DO 10 I=NST,NDTO
      U=X(I,1)**2/(4.0*DIFFUS*TM)
      WW=W(U)
      SDRAW=CONST*WW
      IF(SDRAW.GT.0.005) GO TO 10
      NNODE=I
      GO TO 20
   10 CONTINUE
      NNODE=NDTO
      NELEM=(NNODE-1)/(NN-1)
      GO TO 30
   20 CONTINUE
      NELEM=(NNODE-1)/(NN-1)+1
      IF(NELEM.LT.3)NELEM=3
      NNODE=NELEM*(NN-1)+1
   30 CONTINUE
      RO=X(NNODE,1)
      IF(NNODE.GT.NDTO) NNODE=NDTO
      RETURN
      END
      FUNCTION W(U)
C
C     COMPUTES THEIS WELL FUNCTION.
C
      WC=-0.5772-ALOG(U)
      W=WC
      TERM=1
      J=1
      DO 10 I=1,30
      TERM=(TERM*U)/I
      IF(J.EQ.0) GO TO 20
      W=W+TERM/I
      J=0
      GO TO 30
   20 CONTINUE
      W=W-TERM/I
      J=1
   30 CONTINUE
      EPSI=ABS((W-WC)/W)
      IF(EPSI.LE.0.01) GO TO 40
      WC=W
   10 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE ELGND(NN,IE,SC,IVERT)
C
C     GENERATES ELEMENT MATRIX DELK FOR 3-NODE ELEMENTS.
C
      DIMENSION WC(3),ZGSP(3)
      COMMON /AELEM/DELK(3,3),SELK(3,3),HE(3),ZEN(3)
     1,DSHF(3),XEN(3),SHF(3)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
C
      SV=SC
      NGP=3
      ZGSP(1)=0.7745966692415
      ZGSP(2)=-ZGSP(1)
      ZGSP(3)=0.0
      WC(1)=5.0/9.0
      WC(2)=WC(1)
      WC(3)=8.0/9.0
      DO 15 I=1,NN
      DO 15 J=1,NN
      DELK(I,J)=0.0
   15 CONTINUE
      DO 10 I=1,NGP
      ZE=ZGSP(I)
      CALL SHFUN1(I,ZE,NN,XE,TJ)
      XC=XE
      IF(IVERT.NE.0) XC=1.0
      IF(IVERT.EQ.0) GO TO 25
      IF(IDARCY(IE).EQ.0) GO TO 25
      SV=0.0
      DO 65 JJ=1,NN
   65 SV=SV+HE(JJ)*SHF(JJ)
   25 CONTINUE
      DO 60 K=1,NN
      DO 50 L=1,NN
      IF(L-K) 50,40,40
   40 DELK(K,L)=DELK(K,L)+SHF(K)*SHF(L)*XC*WC(I)*SC*TJ
   50 CONTINUE
   60 CONTINUE
   10 CONTINUE
      DO 70 K=1,NN
```

```
      DO 80 L=1,NN
      IF(L-K) 90,80,80
   90 DELK(K,L)=DELK(L,K)
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE VECMUL(NNODE,LEN)
C
C     PERFORMS VECTOR*MATRIX MULTIPLICATION.
C     VKD = VK*D
C
      COMMON /ASOLV/ISTART(100),NBAND(100),ID(100)
      COMMON /CSOLV/VK(300),D(100,1),VKD(100,1),CTEMP(100,1)
      DO 10 I=1,NNODE
      IS=ID(I)
      IL=ID(I)+NBAND(I)-1
      L=1
      VKD(I,L)=0.0
      DO 20 J=IS,IL
      K=I+J-IS
      VKD(I,L)=VKD(I,L)+VK(J)*D(K,L)
   20 CONTINUE
      IF(I.EQ.1) GO TO 40
      I1=I-1
      IST=ISTART(I)
      DO 30 J=IST,I1
      K=I1-J+1
      IP=ID(J)+K
   30 VKD(I,L)=VKD(I,L)+VK(IP)*D(J,L)
   40 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE VMUL(NNODE,LEN)
C
C     PREFORM VECTOR*MATRIX MULTIPLICATION.
C     CTEMP=VD*CK
C
      COMMON /ASOLV/ISTART(100),NBAND(100),ID(100)
      COMMON /BSOLV/CK(100,1),VK(300)
      COMMON /CSOLV/VD(300),HINT(100,1),GK(100,1),CTEMP(100,1)
      DO 10 I=1,NNODE
      IS=ID(I)
      IL=ID(I)+NBAND(I)-1
      L=1
      CTEMP(I,1)=0.0
      DO 20 J=IS,IL
      K=I+J-IS
      CTEMP(I,1)=CTEMP(I,1)+VD(J)*CK(K,1)
   20 CONTINUE
      IF(I.EQ.1) GO TO 40
      I1=I-1
      IST=ISTART(I)
      DO 30 J=IST,I1
      K=I1-J+1
      IP=ID(J)+K
   30 CTEMP(I,1)=CTEMP(I,1)+VD(IP)*CK(J,1)
   40 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE TIGEN(NTICR,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX)
C
C     GENERATES DISCRETE TIME VECTOR,TIME.
C
      COMMON /APARA/TLESS(100),HLESS(100),TIME(60),SW(60)
      SWST=0.8
      CUN=22.*RW**2/(7.*QFIX*TFACTR)
      TIME(1)=TFACTR
      DO 10 I=2,NTICR
      TIME(I)=TIME(I-1)*TMUL
      TMUL=TMUL+DTMUL
   10 CONTINUE
      RR=RW+1.99
      CONST=RR**2*SS/(4.0*PM)
      IF(IWBC.NE.0)CONST=1000.*CUN*SWST
      DO 20 I=1,NTICR
      TIME(I)=TIME(I)*CONST
   20 CONTINUE
      RETURN
      END
      SUBROUTINE HPUNCH(NNODE,L,INDEX)
C
C     PUNCHES OUT HEAD VALUES AT FINAL TIME.
C
      COMMON /WORKA/VWORK(900)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      COMMON /AFLXX/FLEAK(100),QLEAK(100),HCORE(900),HCOLD(100)
      NPOINT=NNODE
```

```
      DO 15 I=1,NPOINT
      IF(INDEX.EQ.1) VWORK(I)=H(I)
      IF(INDEX.EQ.2)VWORK(I)=QLEAK(I)
   15 CONTINUE
      NCARD=NPOINT/6
      NCARDT=NCARD+2
      WRITE(3,23)NCARDT
   23 FORMAT(///,10X,'TOTAL NUMBER OF PUNCHED CARDS    =',I10)
      WRITE(L,3)NNODE
    3 FORMAT(I10)
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      WRITE(L,13)(VWORK(I),I=IST,IEND)
   13 FORMAT(6E13.5)
      IST=IEND+1
   10 CONTINUE
      NREM=NPOINT-NCARD*6
      IEND=IST+NREM-1
      WRITE(L,13)(VWORK(I),I=IST,IEND)
      RETURN
      END
      SUBROUTINE HREAD(NNODE,L,INDEX)
C
C     READS IN HEAD VALUES AT INITIAL TIME.
C
      COMMON /WORKA/VWORK(900)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      COMMON /AFLXX/FLEAK(100),QLEAK(100),HCORE(900),HCOLD(100)
      READ(L,3)NNODE
      NPOINT=NNODE
    3 FORMAT(I10)
      NCARD=NPOINT/6
      NCARDT=NCARD+1
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      READ(L,13)(VWORK(I),I=IST,IEND)
   13 FORMAT(6E13.5)
      IST=IEND+1
   10 CONTINUE
      NREM=NPOINT-NCARD*6
      IEND=IST+NREM-1
      READ(L,13)(VWORK(I),I=IST,IEND)
      DO 15 I=1,NPOINT
      IF(INDEX.EQ.1) H(I)=VWORK(I)
      IF(INDEX.EQ.2)QLEAK(I)=VWORK(I)
   15 CONTINUE
      RETURN
      END
      SUBROUTINE DCRGEN(RO,RW,NELF,SCFAC,FRLEN,NRR)
C
C      GENERATES DISCRETISATION PARAMETERS.
C
      COMMON /BDISC/M(50),XLEN(50),NNE(50),JBD(2),DISP(2)
      MAXNR=40
      NNE(1)=3
      XLEN(1)=FRLEN
      M(1)=NELF
      RLEN=RO-RW
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      SUM=SUM+XLEN(I)
      M(I)=NELF
      IF(NELF.GT.1) M(I)=NELF-1
      IF(I.LE.3) M(I)=M(1)
      NNE(I)=NNE(1)
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I-1
C
C     SUBDIVIDE THE REMAINING LENGTH.
C
      ELEN=XLEN(NRR)/M(NRR)
      IF(XREM.GT.ELEN) GO TO 30
      M(NRR)=M(NRR)+1
      XLEN(NRR)=XLEN(NRR)+XREM
      GO TO 40
   30 CONTINUE
      NRR=NRR+1
      M(NRR)=XREM/(SCFAC*ELEN)+1
      XLEN(NRR)=XREM
   40 CONTINUE
      RETURN
      END
      SUBROUTINE SWMOD(IQ,IT,RCSNG,DLTA,QFIX,QAQFR,SWOLD)
C
```

```
C     ADJUSTS THE VALUE OF WELL DRAWDOWN.
C
      COMMON /APARA/TLESS(100),HLESS(100),TIME(60),SW(60)
      COMMON /BPARA/DSW(3),QCALC(3),VS(10),QEL(6),GELK(6)
      DELT=DLTA/10.**3
      ACSNG=22.0*RCSNG**2/7.0
      IF(IQ.NE.1) GO TO 15
      QRATIO=QAQFR/QFIX
      IF(ABS(QRATIO).GE.0.8) GO TO 10
      TMM=TIME(IT)-0.5*DLTA
      FACTR=0.5*DELT/ACSNG
      QDEL=QFIX-QAQFR*0.95*TIME(IT)/TMM
      IF(ABS(QRATIO).GE.0.5) QDEL=QFIX-QAQFR
      DSW(IQ)=FACTR*QDEL
      DSW(IQ)=-DSW(IQ)
      GO TO 20
   10 CONTINUE
      TOLD=TIME(IT-2)
      ARGL1=(TIME(IT)-0.5*DLTA)/TOLD
      ARGL2=TIME(IT-1)/TOLD
      TLOG=ALOG(ARGL1)/ALOG(ARGL2)
      DSW(IQ)=(SW(IT-1)-SW(IT-2))*(1.-TLOG)
      DSW(IQ)=-DSW(IQ)
      IF(DSW(IQ).GT.0.) DSW(IQ)=0.0
      GO TO 20
   15 CONTINUE
      IF(IQ.GT.2) GO TO 25
      DSW(IQ)=DSW(IQ-1)*QFIX/QCALC(IQ-1)
      GO TO 20
   25 CONTINUE
      DDSW=DSW(IQ-1)-DSW(IQ-2)
      TERM1=QFIX-QCALC(IQ-1)
      TERM2=QCALC(IQ-1)-QCALC(IQ-2)
      DQR=TERM1/TERM2
      DSW(IQ)=DSW(IQ-1)+DDSW*DQR
   20 CONTINUE
      SW(IT)=DSW(IQ)+SWOLD
      RETURN
      END
      SUBROUTINE AQDIS(GP,NLEN,QAQFR,QRDIF,DELT,TH,RCSNG,DSW,IQ,QFIX)
C
C     COMPUTES TOTAL DISCHARGE INTO THE WELL.
C
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      COMMON /BPARA/DUM(3),QCALC(3),VS(10),QEL(6),GELK(6)
      SUM=0.0
      DO 10 I=1,NLEN
      SUM=SUM+VS(I)*H(I)
   10 CONTINUE
      SUM=SUM-GP
      TWPI=44./7.
      SUM=-SUM*TWPI*TH*2./DELT
      QAQFR=SUM
      ACSNG=0.5*TWPI*RCSNG**2
      TRM=ABS(ACSNG*DSW*2./DELT)
      QCALX=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALX)
      QCALC(IQ)=QCALX
      RETURN
      END
      SUBROUTINE ROUT(NNODE)
C
C     PRINTS OUT HAED VALUES AT FINAL TIME.
C
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      COMMON /APARA/TLESS(100),HLESS(100),TIME(60),SW(60)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,53)
   53 FORMAT(///)
      WRITE(NPRINT,3)
      WRITE(NPRINT,13)
      WRITE(NPRINT,23)
      WRITE(NPRINT,13)
      WRITE(NPRINT,3)
    3 FORMAT(5X,50('*'))
   13 FORMAT(5X,'*',48X,'*')
   23 FORMAT(5X,'*',11X,'FINAL RESULTS OF ANALYSIS',12X,'*')
      WRITE(NPRINT,33)
      DO 10 I=1,NNODE
      WRITE(NPRINT,43)I,X(I,1),H(I),TLESS(I),HLESS(I)
   10 CONTINUE
   33 FORMAT(///,20X,'HEAD .VS. RADIUS AND 1/U .VS. W(U)',///,
     1             10X,'NODE',10X,'R-COORD',10X,'HEAD',14X,'1/U',19X,
     1'W(U)',//)
   43 FORMAT(10X,I3,6X,F10.2,8X,F10.4,7X,E11.4,10X,F10.4)
      RETURN
      END
      SUBROUTINE BSIMP(NNODE,TH,DINDEX,SY,DELT,IT,ITMIN)
C
```

```
C     EVALUATES BOULTON'S CONVOLUTIONAL INTEGRAL BY SIMPSON'S 1/3 RULE.
C
      DIMENSION FN(3)
      COMMON /APARA/TLESS(100),HLESS(100),TIME(60),SW(60)
      COMMON /CSOLV/VD(300),HINT(100,1),GK(100,1),CTEMP(100,1)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      COMMON /AFLXX/FLEAK(100),QLEAK(100),HCORE(900),HCOLD(100)
C
      DOUBLE PRECISION DEXP,ARGU,FN,QCUNT
C
      WRITE(3,3)
    3 FORMAT(///,20X,'LEAKAGE NODAL FLUX',///,10X,'NODE NO.',
     1              20X,'FLUX VALUE',//)
      ITM=IT-1
      IF(ITM.EQ.0) GO TO 5
      TM=TIME(ITM)
      TOLD=(TIME(IT)+TIME(ITM))*0.5
      GO TO 15
    5 TM=0.0
      TOLD=0.5*TIME(IT)
   15 TMM=(TIME(IT)+TIME(IT+1))*0.5
C
      DO 20 II=1,3
      TC=TM+DELT*(II-1)*0.5
      ARGU=-(TMM-TC)*DINDEX
      FN(II)=DEXP(ARGU)
   20 CONTINUE
      ARGU=-DINDEX*(TMM-TOLD)
      NNOD1=NNODE-1
      QLEAK(NNODE)=0.0
      DO 30 I=1,NNOD1
      IF(IT.GT.ITMIN) GO TO 40
      T1=2.*FN(1)*(HINT(I,1)-HCOLD(I))
      T2=4.*FN(2)*(H(I)-HCOLD(I))
      T3=2.*FN(3)*(H(I)-HINT(I,1))
      DQ    =T1+T2+T3
      GO TO 50
   40 CONTINUE
      DQ    =(H(I)-HINT(I,1))*(FN(1)+4.*FN(2)+FN(3))
   50 CONTINUE
      DQ=-DQ
      QCUNT=QLEAK(I)*DEXP(ARGU)
      IF(DQ.LT.0.0) DQ=0.0
      QLEAK(I)=QCUNT+DQ
      WRITE(3,13)I,QLEAK(I)
   13 FORMAT(10X,I5,20X,E13.3)
   30 CONTINUE
      WRITE(3,13) NNODE,QLEAK(NNODE)
      RETURN
      END
      SUBROUTINE VAVEG1(NNODE,NELEM,NN)
C
C     COMPUTES AVERAGE NODAL VELOCITIES.
C
      DIMENSION ICOUNT(100)
      COMMON /VCOM/IPROP(50),IDARCY(50),H(100),VEL(50,3),VMEAN(100,1)
      COMMON /ADISC/X(100,1),NOD(50,3),VCORE(300)
      DATA NREAD,NPRINT/1,3/
C
      WRITE(NPRINT,43)
   43 FORMAT('1',//,10X,'NODAL VELOCITY PRINT OUT',///,
     110X,'NODE',14X,'COMMON ELEMENTS',14X,'VELOCITY',///)
      DO 10 J=1,NNODE
      VMEAN(J,1)=0.0
      ICOUNT(J)=0
   10 CONTINUE
      DO 20 K=1,NELEM
      DO 30 J=1,NN
      NJ=NOD(K,J)
      VMEAN(NJ,1)=VMEAN(NJ,1)+VEL(K,J)
      ICOUNT(NJ)=ICOUNT(NJ)+1
   30 CONTINUE
   20 CONTINUE
      DO 40 J=1,NNODE
      DIV=ICOUNT(J)
      VMEAN(J,1)=VMEAN(J,1)/DIV
      WRITE(NPRINT,13) J,DIV,VMEAN(J,1)
   13 FORMAT('0',10X,I3,20X,F3.1,20X,F10.5)
   40 CONTINUE
      RETURN
      END
```

4.3 LISTING OF STCON3

```
C
C      INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C      DEVELOPED BY P.S. HUYAKORN.
C
C      STCON3,PROGRAM FOR SOLVING STEADY,TWO-DIMENSIONAL,DARCY OR TWO-REGIME
C
C      FLOW USING TRIANGULAR ELEMENTS.
C
C      VERSION DATED OCTOBER,1973.
C
C       FOR FURTHER INFORMATION,CONTACT
C
C         P.S. HUYAKORN OR C.R. DUDGEON
C
C         WATER RESEARCH LABORATORY
C
C         KING ST.,MANLY VALE
C
C         SYDNEY,N.S.W. 2093,AUSTRALIA.
C
C---------------ALL READ STATEMENTS ARE LOCATED IN THE MAIN------------------
C
C---------------PROGRAM AND INDICATED BY  $1 OR $1,SK  SIGN------------------
C
C------------------------------
C      LIST OF INPUT VARIABLES
C------------------------------
C
C***   PROBLEM VARIABLES  ***
C
C      NPROB   =  NUMBER OF PROBLEMS TO BE SOLVED
C      IVEL    =  VELOCITY PRINT-OUT INDEX
C                 FEED IN IVEL=0 IF VELOCITY PRINT-OUT IS NOT REQUIRED OTHERWISE
C                 FEED IN IVEL=1
C      IDISCR  =  DISCRETISATION DATA PRINT-OUT INDEX
C                 FEED IN IDISCR=0 IF DISCRETISATION PRINT-OUT IS NOT REQUIRED
C                 OTHERWISE FEED IN IDISCR=1
C      ORELAX  =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                 SUGGESTED VALUE LIES BETWEEN 1.50 AND 1.85
C      RW      =  RADIUS OF WELL SCREEN
C      RO      =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C      HO      =  INITIAL HEIGHT OR DRAWDOWN OF WATER TABLE AT EXTERNAL RADIUS
C      TH      =  THICKNESS OF AQUIFER
C      HTOL    =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                 SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C      IGP     =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                 IGP=0 FOR NON-GRAVEL PACKED WELL
C      IAQTA   =  AQUITARD INDEX,IAQTA=1 IF THERE IS AN OVERLYING AQUITARD
C                 IAQTA=0 IF THE MAIN AQUIFER IS CONFINED BY IMPERMEABLE STRATA
C      AK      =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                 FEED IN AK = 1./PM IF ONLY DARCY FLOW SOLUTION IS REQUIRED
C                 WHERE PM = COEFFICIENT OF PERMEABILITY OF AQUIFER
C      BK      =  NON-LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                 IF ONLY DARCY FLOW SOLUTION IS REQUIRED SET BK=0.
C      VCR     =  CRITICAL FLOW VELOCITY WHERE NON-DARCY FLOW COMMENCES
C      AGP     =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C      BGP     =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C      VGP     =  CRITICAL FLOW VELOCITY OF GRAVEL PACK MATERIAL
C      THGP    =  THICKNESS OF GRAVEL PACK
C      BTGP    =  HEIGHT OF BASE OF GRAVEL PACK ABOVE BASE OF AQUIFER
C      PA      =  COEFFICIENT OF PERMEABILITY OF OVERLYING AQUITARD
C      THA     =  THICKNESS OF OVERLYING AQUITARD
C      HW      =  HEIGHT OF WATER LEVEL IN THE WELL OR WELL DRAWDOWN
C
C***   DISCRETISATION PARAMETERS  ***
C
C      FRLEN   =  LENGTH OF FIRST SUBREGION
C                 SUGGESTED VALUE IS FRLEN=RW .
C                 FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
C      SCFAC   =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                 SUBREGIONS.SUGGESTED VALUE IS SCFAC=1.50
C      XLMAX   =  PRESCRIBED MAXIMUM BLOCK LENGTH TO AVOID ILL-CONDITIONED
C                 ELEMENTS.MAXIMUM VALUE OF XLMAX SHOULD NOT EXCEED 25.*TH
C      IREG - 1 = NUMBER OF REPEATED REGULAR BLOCKS WITH THE SAME NUMBER OF NODES
C                 ON THE LEFT AND RIGHT VERTICAL LINES
C                 SUGGESTED VALUE IS IREG=2
C      NMIN    =  MINIMUM NUMBER OF NODES ALONG A VERTICAL LINE
C                 TO MINIMISE THE TOTAL NUMBER OF NODES,SUGGEST NMIN=2
C      IPENTR  =  PENETRATION INDEX TO INDICATE WHETHER THE WELL PARTIALLY OR
C                 FULLY PENETRATES THE AQUIFER
C                 IPENTR=1 IF THE WELL PARTIALLY PENETRATES THE AQUIFER AND
C                 BOTTOM ENTRY THROUGH SCREEN IS TO BE TAKEN INTO ACCOUNT.
C                 OTHERWISE IPENTR=0
C      NDSC    =  TOTAL NUMBER OF NODES ON WELL SCREEN(S)
C                 NDSC IS TO BE GREATER THAN OR EQUAL TO 2
C      NSCREN  =  NUMBER OF SCREENED INTERVALS
C      XSCR(I) =  Z-COORDINATE OF BASE OF SCREEN I ABOVE DATUM
C      HSCR(I) =  LENGTH OF WELL SCREEN NUMBER I
C
```

```
C-----------------------------
C     LIST OF OUTPUT VARIABLES
C-----------------------------
C
C     NNODE   =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C     X       =  RADIAL AND VERTICAL NODAL COORDINATES
C     H       =  NODAL HEAD OR DRAWDOWN VALUES
C     TLESS   =  NODAL VALUES OF DIMENSIONLESS RADIUS,1/U
C     HLESS   =  NODAL VALUES OF WELL FUNCTION FOR STEADY FLOW,W(U)
C     QSUM    =  CALCULATED DISCHARGE FROM AQUIFER INTO WELL
C     NELEM   =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C     NOD     =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C     VEL     =  ABSOLUTE ELEMENT VELOCITIES
C     VCOMP1  =  RADIAL COMPONENT OF ELEMENT VELOCITY
C     VCOMP2  =  VERTICAL COMPONENT OF ELEMENT VELOCITY
C     IDARCY  =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
C
C
C
      COMMON /WSCREN/XSCR(5),HSCR(5)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BQCAL/VKQ(600),QWB(40)
      DATA NREAD,NPRINT/1,3/
C
C     **************
C     *            *
C     *  BLOCK 1   *
C     *            *
C     **************
C
C     PRINT INITIAL HEADINGS
C
      WRITE(NPRINT,993)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1023)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1033)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1003)
  993 FORMAT(1H1,4X,51H *************************************************
     1**)
 1003 FORMAT(5X,51H **************************************************)
 1013 FORMAT(5X,51H *                                                *)
 1023 FORMAT(5X,51H *              FINITE ELEMENT SOLUTION OF        *)
 1033 FORMAT(5X,51H *    STEADY STATE,TWO-DIMENSIONAL CONFINED FLOW  *)
C
C     READ AND PRINT GENERAL DATA
C
      READ(NREAD,1011)NPROB,IVEL,IDISCR,ORELAX                          $1
 1011 FORMAT(3I10,F10.2)
      DO 4800 JPRO=1,NPROB
      WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1              I6,12X,'*',/,20X,50('*'))
      READ(NREAD,2003)RW,RO,HO,TH,HTOL                                  $1
 2003 FORMAT(5F10.2)
      WRITE(NPRINT,2193)ORELAX
 2193 FORMAT(//,20X,'GENERAL INPUT DATA',//,
     1              10X,'OVER-RELAXATION FACTOR  =',F10.2)
      READ(NREAD,111)IGP,IAQTA                                           $1
  111 FORMAT(2I10)
      WRITE(NPRINT,193)IGP,IAQTA
  193 FORMAT(/,10X,'GRAVEL PACK INDEX  =',I5,//,
     1         10X,'AQUITARD INDEX   =',I5)
      READ(NREAD,2013) AK,BK,VCR                                        $1
 2013 FORMAT(3F10.3)
      PMK=1./AK
      IF(VCR.GT.0.00001) PMK=1./(AK+BK*VCR)
      RGP=RW
      WRITE(NPRINT,23)AK,BK,RW,RO,HO,TH,HTOL
   23 FORMAT(2(/),10X,32HAQUIFER AND WELL CHARACTERISTICS,///,
     110X,19HPERM. CONSTANT A = ,F10.3,//,
     210X,19HPERM. CONSTANT B = ,F10.3,//,
     310X,19HRADIUS OF WELL   = ,F10.3,//,
     410X,22HRADIUS OF INFLUENCE = ,F10.3,//,
     510X,24HHEIGHT OF WATER TABLE = ,F10.3,//,
     610X,24HTHICKNESS OF AQUIFER  = ,F10.3,//,
     710X,39HTOLERANCE FOR SUCCESSIVE HEAD VALUES = ,F10.3)
      IF(IGP.EQ.0) GO TO 29
      READ(NREAD,331)AGP,BGP,VGP,THGP,BTGP                              $1,SK
  331 FORMAT(5F10.3)
      GRGP=AGP*VGP+BGP*VGP**2
      PMGP=1./AGP
      IF(GRGP.NE.0) PMGP=VGP/GRGP
      WRITE(NPRINT,153)AGP,BGP,VGP,PMGP
```

```
      RGP = RW + THGP
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1           10X,'COEFFICIENT A =',F10.4//
     2           10X,'COEFFICIENT B = ',F10.4//
     3           10X,'CRITICAL VELOCITY = ',F10.4//
     4           10X,'COEFFICIENT K = ',F10.4//)
      WRITE(NPRINT,163)THGP,RGP
  163 FORMAT(10X,' THICKNESS OF PACK = ',F10.2//
     1           10X,'RADIUS OF PACK = ',F10.2//)
   29 CONTINUE
C
C     READ AQUITARD DATA.
C
      IF(IAQTA.EQ.0) GO TO 39
      READ(NREAD,71)PA,THA                                          $1,SK
   71 FORMAT(F10.4,F10.2)
      BFAC=PA/THA
      WRITE(NPRINT,273)PA,THA
  273 FORMAT(///,20X,'AQUITARD PROPERTIES',///,
     1           10X,'HYDRAULIC CONDUCTIVITY   =',F10.5//
     2           10X,'THICKNESS  =',F10.2//)
   39 CONTINUE
      READ(NREAD,2023)HW                                            $1
 2023 FORMAT(F10.2)
C
C     **************
C     *            *
C     *  BLOCK 2   *
C     *            *
C     **************
C
C     READ AND PRINT DISCRETIZATION PARAMETERS.
C
      READ(NREAD,901)FRLEN,SCFAC,XLMAX,IREG,NMIN                    $1
  901 FORMAT(3F10.2,2I10)
      READ(NREAD,801)IPENTR,NDSC,NSCREN                             $1
  801 FORMAT(3I10)
C
C     READ WELL SCREEN DATA.
C
      SCLEN=0.
      WRITE(NPRINT,553)
  553 FORMAT(/,10X,'SCREEN NO.',10X,'BASE HEIGHT',14X,'LENGTH',//)
      DO 702 I=1,NSCREN                                             *NSCREN*
      READ(NREAD,601)XSCR(I),HSCR(I)                                $1
  601 FORMAT(2F10.2)
      WRITE(NPRINT,563)I,XSCR(I),HSCR(I)
  563 FORMAT(10X,I6,14X,F11.2,10X,F10.2)
  702 SCLEN=SCLEN+HSCR(I)
      WRITE(NPRINT,139)IPENTR,NDSC                                  /*
      NDSC= NDSC- NSCREN + 1
  139 FORMAT(/,10X,'PENETRATION INDEX  =',I5,//,
     1        10X,'TOTAL NUMBER OF NODES ON WELL SCREEN(S)  =',I5)
C
C     GENERATE AND PRINT DISCRETIZATION DATA.
C
      CALL GXNOD(IPENTR,NDSC,SCLEN,FRLEN,SCFAC,
     1XLMAX,IREG,NMIN,RW,RO,TH,NNODE,NELEM,LVEC,IDISCR)
C
C     GENERATE IPROP.
C
      CALL AQPROP(NELEM,RGP,BTGP,TH,IGP)
C
C     **************
C     *            *
C     *  BLOCK 3   *
C     *            *
C     **************
C
C     PRESCRIBE HEAD VALUES AT BOUNDARIES AND FIND BANDWIDTHS.
C
      HWDRAW=HW-HO
      HODRAW=0.0
      NDW=NDVEC(1)
      NDRO=NDVEC(LVEC)-NDVEC(LVEC-1)
      CALL BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,RW,NSCREN)
      WRITE(NPRINT,203)
  203 FORMAT(////,10X,'PRESCRIBED NODES AND VALUES',//,
     1        10X,'NODES',6X,'PRESCRIBED VALUES',//)
      DO 200 I=1,NBD
      WRITE(NPRINT,213)JBD(I),DISP(I)
  213 FORMAT(8X,I5,8X,F10.3)
  200 CONTINUE
      IF(IAQTA.EQ.0) GO TO 166
      CALL VDFB(LVEC)
      DO 176 I=1,NNODE
      HINT(I,1)=0.0
  176 CONTINUE
  166 CONTINUE
C
C     FIND BANDWIDTHS ETC.
```

```
C
      CALL EBFIN3(LEN,NELEM,3,NNODE)
      WRITE(NPRINT,233)LEN
  233 FORMAT(//////,10X,25HLENGTH OF GROSS VECTOR = ,I8,//)
      DO 130 I=1,LEN
      VD(I)=0.0
  130 CONTINUE
      NT=0
      NN=3
      SC=1.
      DO 135 IE=1,NELEM
      CALL ELGND3(NT,SC,IE)
      CALL MERBD(NN,IE)
  135 CONTINUE
C
C     SET IDARCY
C
      DO 99 I=1,NELEM
      IDARCY(I)=0
   99 CONTINUE
      NBSTOR=NBW
      IF(XSCR(1).GT.0.) NBW=0
      NBP=NBW+1
      IF(NBP.EQ.1) GO TO 94
      DO 117 I=1,NBW
      CK(I,1)=HW-HO
  117 CONTINUE
   94 CONTINUE
C
C
C      SET UP LOOP FOR ITERATION ON H
C
      WRITE(NPRINT,1583)HW
 1583 FORMAT('1',///,10X,'*****************************************',/,
     1                  10X,'*   WATER LEVEL IN THE WELL =',F7.2,3X,'*',/,
     2                  10X,'*****************************************')
      VCOUNT=0
      NCOUNT=NNODE
      IF(BK.LT.0.000001) NCOUNT=0
      DO 999 III=1,10
C
C     **************
C     *            *
C     *  BLOCK 4   *
C     *            *
C     **************
C
C     ZERO GROSS STIFFNESS MATRIX AND LOAD MATRIX
C
      DO 300 I=1,LEN
      VK(I)=0.0
  300 CONTINUE
      DO 305 I=NBP,NNODE
      CK(I,1)=0.0
  305 CONTINUE
C
C      COMPUTE STIFFNESS MATRIX FOR EACH ELEMENT AND MERGE
C
C      FOR III=1  ONLY PERFORM SOLUTION FOR DARCY CASE
C
      DO 350 I=1,NELEM
      IF( IPROP(I).EQ.0) GO TO 755
      AKK=AK
      BKK=BK
      PKK=PMK
      GO TO 365
  755 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
  365 CONTINUE
      NT=NREP(I)
      CALL ELGNCT(I,III,AKK,BKK,NT,VCOUNT,PKK)
      CALL MERB3(NN,I)
  350 CONTINUE
C
C     ADD LEAKAGE TERM CONTRIBUTED BY AQUITARD.
C
      IF( IAQTA.EQ.0) GO TO 86
      BFL=BFAC
      CALL GVMOD(LVEC,BFL,0)
   86 CONTINUE
      I=NOVEC(1)+1
      LNN=ID(I)-1
      DO 122 I=1,LNN
      VKQ(I)=VK(I)
  122 CONTINUE
C
C     **************
C     *            *
```

```
C      *  BLOCK 5    *
C      *             *
C      ***************
C
C      SOLVE EQUATIONS BY EMPLOYING BANDED ELIMINATION SCHEME.
C
       NLL=1
       CALL QFLUX(NBW,NNODE)
       CALL SYMSOL(NNODE,NLL,NBW)
C
C      PRINT OUT SOLUTION FOR FUNCTION
C      ONLY THE DARCY AND FINAL SOLUTIONS
C
       IF(III.EQ.1) GO TO 900
       NCOUNT=0
       EMAX=0.0
       DO 450 I=1,NNODE
       EPSI=CK(I,1)-H(I)
       IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
       IF(ABS(EPSI).LE.HTOL)  GO TO 460
       NCOUNT=NCOUNT+1
       H(I)=H(I)+ORELAX*EPSI
       GO TO 450
  460  CONTINUE
       H(I)=CK(I,1)
  450  CONTINUE
       IF(NCOUNT.EQ.0) GO TO 1000
       GO TO 1999
  900  CONTINUE
       DO 950 I=1,NNODE
       H(I)=CK(I,1)
       RHO=0.5*X(I,1)/TH
       TZI=X(I,2)/TH
       SDLESS=(HO-H(I))/(HO-HW)
  950  CONTINUE
       IF(NCOUNT.EQ.0) GO TO 1000
 1999  CONTINUE
C
C      REGENERATE IDARCY
C
       DO 199 I=1,NELEM
                  CALL HYPROP(I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,
                  AL ,BL,PML,VCL)
       J1=NOD(I,1)
       J2=NOD(I,2)
       J3=NOD(I,3)
       NT=NREP(I)
       CALL VCHEC3(I,AL,BL,J1,J2,J3,NT,PML,VCL,HRRX,HRRY)
  199  CONTINUE
  999  CONTINUE
 1000  CONTINUE
       IF(NCOUNT.NE.0)  GO TO 5000
       WRITE(NPRINT,413)III,EMAX
  413  FORMAT(//,10X,'NO. OF ITERATIONS REQUIRED =',I5,//,
      1             10X,'MAXIMUM ERROR IN HEAD =',F10.3)
       WRITE(NPRINT,433)
  433  FORMAT('0',20X,'FINAL SOLUTION',///,
      1             5X,'NODE',10X,'R-COORD',10X,'Z-COORD',10X,'HEAD',
      2            10X,'RHO-COORD',10X,'TZI-COORD',10X,'DRAWDOWN RATIO',/)
       DO 470 I=1,NNODE
       RHO=0.5*X(I,1)/TH
       TZI=X(I,2)/TH
       SDLESS=(HO-H(I))/(HO-HW)
       WRITE(NPRINT,953)I,X(I,1),X(I,2),H(I),RHO,TZI,SDLESS
  953  FORMAT(5X,I3,2(6X,F10.2),4(8X,F10.4))
  470  CONTINUE
C
C      ***************
C      *             *
C      *  BLOCK 6    *
C      *             *
C      ***************
C
C      EVALUATE VELOCITY AND GRADIENT USING FINAL HEADS
C
  799  CONTINUE
       VCOUNT=1
C
C      COMPUTE AND PRINT ELEMENT VELOCITIES
C
       IF(IVEL.EQ.0) GO TO 7000
       WRITE(NPRINT,1203)
 1203  FORMAT('1',///,20X,'*******************************************'/
      1                 20X,'*           ELEMENT VELOCITIES            *'/
      2                 20X,'*******************************************'///
      310X,'ELEM NO.',10X,'RADIAL VEL',10X,'VERTIC VEL',10X,'IDARCY',///)
       DO 3000 I=1,NELEM
       J1=NOD(I,1)
       J2=NOD(I,2)
       J3=NOD(I,3)
                  CALL HYPROP(I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,
                  AL,BL,PML,VCL)
       NT=NREP(I)
       CALL VCHEC3(I,AL,BL,J1,J2,J3,NT,PML,VCL,HRRX,HRRY)
```

```
      VCOMP1=-VEL(I)*HRRX
      VCOMP2=-VEL(I)*HRRY
      WRITE(NPRINT,1103)I,VCOMP1,VCOMP2,IDARCY(I)
 1103 FORMAT(10X,I5,2(10X,F10.3),10X,I5)
 3000 CONTINUE
 7000 CONTINUE
C
      NBW=NBSTOR
      CALL QCALC(NBW,QSUM)
      BLEAK=RO
      IF(IAQTA.EQ.0) GO TO 4505
      BLEAK=PMK*TH*THA/PA
      BLEAK=SQRT(BLEAK)
 4505 CONTINUE
      J=1
      CALL TCURV3(HO,BLEAK,NNODE,QSUM,AK,BK,PMK,TH,J)
 5000 CONTINUE
C------------------------------------------------------------------C
C                                                                  C
C         IF SO REQUIRED,USER MAY CALL HIS PLOTTING SUBROUTINES    C
C                                                                  C
C         AT THIS LOCATION.                                        C
C                                                                  C
C------------------------------------------------------------------C
C
C     THE FOLLOWING PLOTS MAY BE OBTAINED:-
C           (1) DISCRETISATION PATTERN,OBTAINED BY USING VARIABLES NOD,X,
C               NELEM,NNODE
C           (2) CONTOURS OF HYDRAULIC HEADS,OBTAINED BY USING VARIABLES
C               H,X,NNODE,NOD,NELEM
C           (3) DIMENSIONLESS TYPE CURVES,OBTAINED BY USING VARIABLES HLESS,
C               TLESS,NNODE
C           (4) VELOCITY FIELD,OBTAINED BY USING VARIABLES VEL,X,NOD,VCOMP1,
C               VCOMP2
C           (5) LOCATION OF NON-DARCY FLOW ZONE,OBTAINED BY USING VARIABLES
C               IDARCY,NELEM
C
 4800 CONTINUE
      STOP
      END
      SUBROUTINE GXNOD(IPENTR,NDSC,SCLEN,FRLEN,
     1SCFAC,XLMAX,IREG,NMIN,RW,RO,TH,NNODE,NELEM,LVEC,IDISCR)
C
C     GENERATES AND PRINTS ALL DISCRETIZATION DATA.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      DATA NREAD,NPRINT/1,3/
      NESC=NDSC
      IF(IPENTR.NE.0) NESC=NDSC-1
      XEL=SCLEN/(NESC-1)
      THCSNG=TH-SCLEN+.01
      NCSNG=THCSNG/XEL
      XREM=THCSNG-XEL*NCSNG
      IF(XREM.GE.XEL) NCSNG=NCSNG+1
      NDWB(1)=NCSNG+NDSC
      TH=SCLEN+XEL*NCSNG
      CALL DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
      IF(IPENTR.NE.0) GO TO 50
      NRST=1
      NFR=NDWB(1)
      NSTFR=1
      GO TO 55
   50 CONTINUE
      XEL=2.0*SCLEN/(NESC-1)
      THCSNG=TH-SCLEN+.01
      NCSNG=THCSNG/XEL
      XREM=THCSNG-XEL*NCSNG
      XEL=0.5*XEL
      IF(XREM.GE.XEL) NCSNG=NCSNG+1
      NDWB(1)=NCSNG+NDSC
      NRST=4
      NHALF=NDSC/2
      NFR=NDWB(1)-NHALF
      NSTFR=NDWB(1)+NFR+NHALF+2
   55 CONTINUE
      CALL NCRGEN(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,33) NDWB(1),NRR
   33 FORMAT('1',20X,'DISCRETIZATION DATA',///,
     1             10X,'NUMBER OF NODES AT WELL BOUNDARY     =',I10,///,
     2             10X,'TOTAL NUMBER OF REGIONS     =',I10)
   94 CONTINUE
C
C     DISCRETIZE ENTIRE REGION INTO FINITE ELEMENTS.
C
      KCREP=1
      NTSEL=1
```

```
      IF(NRST.EQ.1) GO TO 655
      CALL DCRFST(IPENTR,NDSC,KCREP,NTSEL)
  655 CONTINUE
      DO 65 I=NRST,NRR
      N1=NH1(1,I)
      NFND1=NDB1(1,I)
      NFND2=NDB2(1,I)
      NPAT=NTRAN(1,I)
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
   65 CONTINUE
C
C     PRINT OUT ELEMENT DATA
C
      NELEM=NTSEL-1
      IF(IDISCR.EQ.0) GO TO 104
      WRITE(NPRINT,83)
   83 FORMAT(///,10X,46H IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS,/
     1//,12X,7HELEM NO,10X,5HNODE1,10X,5HNODE2,10X,5HNODE3,10X,17HREPETI
     2TION NUMBER,//)
      DO 80 I=1,NELEM
      WRITE(NPRINT,93)I,(NOD(I,K),K=1,3),NREP(I)
   93 FORMAT(10X,I5,3(10X,I5),12X,I5)
   80 CONTINUE
  104 CONTINUE
C
C     GENERATE NODAL COORDINATES.
C
      NS=1
      RI=RW
      IF(NRST.EQ.1) GO TO 755
      CALL CORFST(IPENTR,NDSC,SCFAC,RI,TH,SCLEN,RW,NS)
  755 CONTINUE
      ZI=0.0
      MM=1
      DO 75 I=NRST,NRR
      NN=NH1(1,I)
      DXG=XLEN(I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   75 CONTINUE
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
C
C
      NNODE = NS-1
      IF(IDISCR.EQ.0) GO TO 114
      WRITE(NPRINT,103)
  103 FORMAT(1H1,10X,10H NODE DATA,//,10X,27H NODE     R-COORD     Z-COORD
     1,//)
C
      DO 85 I=1,NNODE
      WRITE(NPRINT,113)I,(X(I,K),K=1,2)
  113 FORMAT(10X,I4,2F11.2)
   85 CONTINUE
  114 CONTINUE
C
C     GENERATE TOP BOUNDARY COORDINATE VECTOR.
C
      NDTO=NS-1
      CALL TOPVEC(TH,LVEC,NDTO)
C
C     PRINT OUT BOUNDARY COORDINATE VECTORS.
C
      WRITE(NPRINT,223)
  223 FORMAT(///,20X,'TOP BOUNDARY NODES AND RADIAL COORDINATES.',///,
     1            10X,'NODE NUMBER',20X,'R-COORDINATE',//)
      DO 335 I=1,LVEC
      WRITE(NPRINT,333)NDVEC(I),RVEC(I)
  335 CONTINUE
  333 FORMAT(10X,I7,25X,F10.2)
      RETURN
      END
      SUBROUTINE DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NRR,XLEN
C
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      MAXNR=89
      XLEN(1)=FRLEN
      RLEN=RO-RW
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      IF(XLEN(I).GT.XLMAX) XLEN(I)=XLMAX
      SUM=XLEN(I)+SUM
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I
      DENOM=1.+SCFAC+SCFAC**2
      XLEN(I-2)=(XREM+XLEN(I-1)+XLEN(I-2))/DENOM
```

```
      XLEN(I-1)=XLEN(I-2)*SCFAC
      XLEN(I)=XLEN(I-1)*SCFAC
      RETURN
      END
      SUBROUTINE NCRGEN(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NTRAN,NH1,NH2,NDB1,NDB2
C     ONLY FOR REGIONS 2 TO NRR,REGION 1 IS SPECIALLY TREATED.
C
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      NC=NFR
      NCOUNT=0
      DO 10 I=NRST,NRR
      IRGG=IREG
      IF((I.EQ.NRST).AND.(IPENTR.NE.0)) IRGG=0
      IF(I.EQ.1) IRGG=2
      NCOUNT=NCOUNT+1
      IF(NCOUNT.LT.IRGG) GO TO 20
      NCOUNT=0
      NHALF=NC/2
      NREM=NC-2*NHALF
      IF(NREM.GT.0) GO TO 15
      NC1=NC-1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(1,I)=2
      NC=NC1
      NH2(1,I)=NC
      NH1(1,I)=NC+1
      GO TO 10
   15 CONTINUE
      NC1=NHALF+1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(1,I)=1
      NC=NC1
      NH2(1,I)=NC
      NH1(1,I)=2*NH2(1,I)-1
      GO TO 10
   20 CONTINUE
      NH2(1,I)=NC
      NTRAN(1,I)=0
      NH1(1,I)=NH2(1,I)
   10 CONTINUE
C
C     GENERATE NDB1,NDB2
C
      J=NRST
      NDB1(1,J)=NSTFR
      NDB2(1,J)=NSTFR+NH1(1,J)
      J=J+1
      DO 50 I=J,NRR
      NDB1(1,I)=NDB1(1,I-1)+NH1(1,I-1)
   50 CONTINUE
      NR1=NRR-1
      DO 60 I=J,NR1
      NDB2(1,I)=NDB1(1,I+1)
   60 CONTINUE
      NDB2(1,NRR)=NDB1(1,NRR)+NH1(1,NRR)
      RETURN
      END
      SUBROUTINE DCRFST(IPENTR,NDSC,KCREP,NTSEL)
C
C     GENERATES DISCRETIZATION DATA FOR COMPOSITE REGIONS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      KCREP=1
      NSEL=1
      IF(IPENTR.EQ.0) GO TO 60
      NUPEN=NDWB(1)-NDSC +1
      N1=NUPEN
      N2=2
      N3=N1
      NFND1=1
      NFND2=N1+1
      NFND3=NDWB(1)+2
      NTSEL=1
      NCPAT=3
      CALL CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
      N1=NDWB(1)-NUPEN
      NPAT=1
      NFND1=NFND2+1
      NFND2=NFND1+NDWB(1)-1
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
      NCPAT=3
      N2=N1/2+1
      N1=NDWB(1)-N2
      N3=N1
      NFND1=NDWB(1)+2
```

```
      NFND2=NFND1+N1
      NFND3=NFND2+N2
      CALL CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
      GO TO 400
   60 CONTINUE
      NTSEL=1
      NUPEN=NDSC
      NCPAT=-3
      N1=NDWB(1)
      N2=NUPEN/2+1
      N3=N1-N2+1
      IF(N2.EQ.NUPEN) N3=N1
      NFND1=1
      NFND2=NFND1+N1
      NFND3=NFND2+N2
      CALL CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
  400 CONTINUE
      RETURN
      END
      SUBROUTINE CORFST(IPENTR,NDSC,SCFAC,RI,TH,SCLEN,RW,NS)
C
C     GENERATES NODAL COORDINATES FOR FIRST COMPOSITE REGION.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      NS=1
      IF(IPENTR.EQ.0) GO TO 60
      PENLEN=SCLEN
      XL1=RW
      XL2=XLEN(1)
      NUPEN=NDWB(1)-NDSC+1
      N1=NUPEN
      N2=2
      N3=N1
      TH1=TH-PENLEN
      TH2=TH1/(N1-1)
      RI=0.0
      NPAT=3
      CALL COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
      RI=RW
      NS=NS-1
      MM=1
      NN=NDSC-1
      DXG=XLEN(1)
      ZI=X(NS,2)
      TH1=PENLEN
      CALL COORDC(NS,MM,NN,DXG,TH1,RI,ZI)
C
      XL1=XLEN(2)
      XL2=XLEN(3)
      N1=NDWB(1)-NDSC/2
      N2=NDSC/2
      N3=N1
      TH1=TH
      TH2=PENLEN
      CALL COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
      GO TO 400
   60 CONTINUE
      NPAT=-3
      RI=RW
      TH1=TH
      TH2=SCLEN
      N1=NDWB(1)
      N2=NDSC/2+1
      N3=N1-N2+1
      XL1=XLEN(1)/(1.+SCFAC)
      XL2=SCFAC*XL1
      CALL COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
  400 CONTINUE
      RETURN
      END
      SUBROUTINE UBLOCK(NFND1,NFND2,NFND3,NPAT,NTSEL,KCREP)
C
C     GENERATES ELEMENT CONNECTIVITIES IN GRADED UNIT BLOCK.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      K=0
      IF(NPAT.NE.3) K=1
      ICOUNT=0
      NST=NTSEL
      NLST=NST+2
      DO 10 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      ICOUNT=ICOUNT+1
      IF(ICOUNT.EQ.2) GO TO 20
      IF(ICOUNT.EQ.3) GO TO 30
      NOD(J1,1)=NFND1
```

```
      NOD(J1,2)=NFND2
      NOD(J1,3)=NFND1+1
      GO TO 10
   20 CONTINUE
      NOD(J1,1)=NFND1+K
      NOD(J1,2)=NFND2
      NOD(J1,3)=NFND3+K
      GO TO 10
   30 CONTINUE
      NOD(J1,1)=NFND2
      NOD(J1,2)=NFND3
      NOD(J1,3)=NFND3+1
   10 CONTINUE
      NTSEL=NLST+1
      RETURN
      END
      SUBROUTINE BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC BLOCKS.
C
      DIMENSION ND(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND1+1
      ND(2,1)=ND(1,3)
      ND(2,2)=ND(1,2)
      ND(2,3)=ND(1,2)+1
      IF(NPAT.EQ.1) GO TO 200
      IF(NPAT.EQ.0) GO TO 265
      NTEMP=N1-2
      DO 90 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      DO 50 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      NTEMP=NTEMP-1
      NTSEL=NLST+1
   90 CONTINUE
      GO TO 400
  200 CONTINUE
      ND(3,1)=ND(2,1)
      ND(3,2)=ND(2,2)+1
      ND(3,3)=ND(3,1)+1
      NTEMP=(N1-1)/2-1
      L1=2
      L2=1
      DO 250 JJ=1,3
      L3=2
      IF(JJ.EQ.2) L3=1
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      DO 270 K=1,3
      IF(K.EQ.1) LL=L1
      IF(K.EQ.2) LL=L2
      IF(K.EQ.3) LL=L3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)*LL
  270 CONTINUE
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  250 CONTINUE
      GO TO 400
  265 CONTINUE
      NTEMP=N1-2
      DO 190 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 140 J1=NST,NLST
      NREP(J1)=KCREP
      DO 150 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
  150 CONTINUE
  140 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  190 CONTINUE
  400 CONTINUE
      RETURN
      END
      SUBROUTINE CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
C
C     GENERATES ELEMENT CONNECTIVITIES IN COMPOSITE BLOCKS.
```

```
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
C
C     BLOCK1 :- CONNECTS VERTICAL LINE 1 TO LINE 2.
C
      NDF=N1-N3
      NPAT=0
      IF(NDF.NE.0) NPAT=1
      NF2=NFND2
      IF(NPAT.EQ.1) GO TO 20
      NF1=NFND1+(N1-N2)
      IF(NCPAT.NE.3) NF1=NFND1
      NN2=N2
      NN1=N2
      GO TO 30
   20 CONTINUE
      NF1=NFND1+(N1-2*N2+1)
      IF(NCPAT.NE.3) NF1=NFND1
      NN2=N2
      NN1=2*N2-1
   30 CONTINUE
      CALL BBLOCK(KCREP,NN1,NF1,NF2,NPAT,NTSEL)
C
C     BLOCK 2 :- CONNECTS VERTICAL LINE 2 TO LINE 3.
C
      NS1=NF1
      NS2=NF2
      NPAT=0
      NN1=NN2
      NF1=NF2
      NF2=NFND3+N3-N2
      IF(NCPAT.NE.3) NF2=NFND3
      CALL BBLOCK(KCREP,NN1,NF1,NF2,NPAT,NTSEL)
C
C     BLOCK 3 :- UNIT BLOCK.
C
      IF(NCPAT.NE.3) GO TO 60
      NF1=NS1-1
      NF3=NF2-1
      NF2=NS2
      GO TO 70
   60 CONTINUE
      NF1=NFND1+N1-1-(N3-N2)
      NF2=NFND2+N2-1
      NF3=NFND3+N2-1
      NS1=NF1
   70 CONTINUE
      CALL UBLOCK(NF1,NF2,NF3,NCPAT,NTSEL,KCREP)
C
C     BLOCK 4 :- CONNECT LINE 1 TO LINE 3.
C
      NF1=NFND1
      NF2=NFND3
      IF(NCPAT.NE.3) NF1=NS1+1
      IF(NCPAT.NE.3) NF2=NF3+1
      NN1=N3-N2
      IF(NN1.EQ.1) GO TO 85
      NPAT=0
      CALL BBLOCK(KCREP,NN1,NF1,NF2,NPAT,NTSEL)
   85 CONTINUE
      RETURN
      END
      SUBROUTINE COORDC(NS,MM,NN,DXG,TH,RI,ZI)
C
C     GENERATES NODAL COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /ASCLV/XX(5,90),XY(5,90)
      DY=TH/(NN-1)
      DO 20 L=1,NN
   20 XX(1,L)=RI
      DO 50 K=1,MM
   50 XY(K,1)=ZI
      DO 60 J=2,NN
      JM1=J-1
      DO 70 K=1,MM
      XY(K,J)=XY(K,JM1)+DY
   70 CONTINUE
   60 CONTINUE
      DO 80 J=1,MM
      DO 90 K=1,NN
      L=NN*(J-1)+K+NS-1
      X(L,1)=XX(J,K)
      X(L,2)=XY(J,K)
   90 CONTINUE
   80 CONTINUE
      NEN=NS+(MM*NN)-1
      NS=NEN+1
      RI=X(NEN,1)+DXG
      RETURN
```

```
      END
      SUBROUTINE COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
C
C     GNERATES NODAL COORDINATES FOR COMPOSITE BLOCK.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      JJ=0
      MM=1
      IF(NPAT.NE.3) GO TO 20
      NN=N1
      DXG=XL1
      ZI=0.
      TH=TH1
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   15 CONTINUE
      ZI=TH1-TH2
      NN=N2
      TH=TH2
      DXG=XL2
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      GO TO 25
   20 CONTINUE
      NN=N1-N3+N2
      DXG=0.
      ZI=0.
      TH=TH2
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH+(TH1-TH2)/(N3-N2)
      NN=N1-NN
      DXG=XL1
      TH=TH1-ZI
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      DXG=XL2
      TH=TH2
      NN=N2
      ZI=0.0
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   25 CONTINUE
      RETURN
      END
      SUBROUTINE TOPVEC(TH,LVEC,NDTO)
C
C     COMPUTES RADIAL COORDINATES FOR NODES ON TOP BOUNDARY.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      J=0
      DO 10 I=1,NDTO
      ZDIF=ABS(TH-X(I,2))
      IF(ZDIF.GT.0.001) GO TO 10
      J=J+1
      NDVEC(J)=I
      RVEC(J)=X(I,1)
   10 CONTINUE
      LVEC=J
      RETURN
      END
      SUBROUTINE BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,RW,NSCREN)
C
C     LOCATES NODES WHERE HEAD VALUES ARE PRESCRIBED.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      K=0
      NFND=1
      DO 30 I=1,NSCREN
      XST=XSCR(I)-0.01
      XEND=XST+HSCR(I)+0.02
      DO 40 J=1,NNODE
      L=NFND+J-1
      IF(L.EQ.1) GO TO 222
      IF(X(L,1).GT.RW) GO TO 30
  222 CONTINUE
      IF((X(L,2).LT.XST).OR.(X(L,2).GT.XEND)) GO TO 70
   22 K=K+1
      JBD(K)=L
      DISP(K)=HW
      GO TO 40
   70 IF(X(L,2).GT.XEND) GO TO 30
   40 CONTINUE
      NFND=L+1
   30 CONTINUE
      J=K
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
```

```
   25 CONTINUE
      NBD=J
      RETURN
      END
      SUBROUTINE CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
C
C     COMPUTES NON-LINEAR COEFFICIENT FOR TRIANGULAR ELEMENTS.
C
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      ACONST=AREA(NT)*ORX(M)/TEMP
  500 CONTINUE
      RETURN
      END
      SUBROUTINE ELGENC(M,III,AK,BK,NT,VCOUNT,PMK)
C
C     GENERATES ELEMENT MATRIX,EK,FOR TRIANGULAR ELEMENTS.
C
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      IF(III.NE.1)  GO TO 10
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
   10 CONTINUE
      IF(NREP(M).EQ.NT)   GO TO 50
      NT=NREP(M)
      IF(III.NE.1)  GO TO 50
      XJ=X(J2,1)-X(J1,1)
      XM=X(J3,1)-X(J1,1)
      YJ=X(J2,2)-X(J1,2)
      YM=X(J3,2)-X(J1,2)
      AREA(NT)=0.5*(XJ*YM-XM*YJ)
      AFUN=2.*AREA(NT)
      B(NT,1)=(YJ-YM)/AFUN
      B(NT,2)=   (YM)/AFUN
      B(NT,3)=  (-YJ)/AFUN
      C(NT,1)=(XM-XJ)/AFUN
      C(NT,2)=  (-XM)/AFUN
      C(NT,3)=   (XJ)/AFUN
      DO 100 I=1,3
      DO 100 J=1,3
      IF(J-I) 105,110,110
  110 ELK(I,J,NT)=B(NT,I)*B(NT,J)+C(NT,I)*C(NT,J)
      GO TO 100
  105 ELK(I,J,NT)=ELK(J,I,NT)
  100 CONTINUE
   50 CONTINUE
      IF(IDARCY(M).EQ.0) GO TO 70
      IF(III.EQ.1)  GO TO 70
  500 CONTINUE
      CALL CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
      GO TO 80
   70 CONTINUE
      PX=PMK
      ACONST=PX*ORX(M)*AREA(NT)
   80 CONTINUE
C
C                    CALCULATE STIFFNESS MATRIX
C     NORMAL STIFFNESS MATRIX MODIFIED BY THE FACTOR  OACONSTO
C
      DO 200 I=1,3
      DO 200 J=1,3
      EK(I,J)=ACONST*ELK(I,J,NT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE AQPROP(NELEM,RGP,BTGP,TH,IGP)
C
C     GENERATES ELEMENT PROPERTIES INDEX,IPROP.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      DO 10 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      ORX=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      ORY=(X(J1,2)+X(J2,2)+X(J3,2))/3.
      IPROP(I)=1
      IF(IGP.EQ.0) GO TO 10
      IF((ORX.LT.RGP).AND.(ORY.GT.BTGP)) IPROP(I)=0
```

```
   10 CONTINUE
      RETURN
      END
      SUBROUTINE ELGND3(NT,SC,M)
C
C     GENERATES ELEMENT MATRIX,D,FOR TRIANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      IF(NREP(M).EQ.NT) GO TO 50
      NT=NREP(M)
      XJ=X(J2,1)-X(J1,1)
      XM=X(J3,1)-X(J1,1)
      YJ=X(J2,2)-X(J1,2)
      YM=X(J3,2)-X(J1,2)
      AREA(NT)=0.5*(XJ*YM-XM*YJ)
      AFUN=2.*AREA(NT)
      B(NT,1)=(YJ-YM)/AFUN
      B(NT,2)=   (YM)/AFUN
      B(NT,3)=  (-YJ)/AFUN
      C(NT,1)=(XM-XJ)/AFUN
      C(NT,2)=  (-XM)/AFUN
      C(NT,3)=   (XJ)/AFUN
      DO 100 I=1,3
      DO 100 J=1,3
      IF(J-I) 105,110,110
  110 ELK(I,J,NT)=B(NT,I)*B(NT,J)+C(NT,I)*C(NT,J)
      GO TO 100
  105 ELK(I,J,NT)=ELK(J,I,NT)
  100 CONTINUE
C
C     FORM ELEMENT MATRIX:- D
C
      D(1,1)=0.5*ORX(M)*AREA(NT)*SC/3.
      D(2,2)=D(1,1)
      D(3,3)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      D(1,3)=D(1,2)
      D(3,1)=D(1,3)
      D(2,3)=D(1,2)
      D(3,2)=D(2,3)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE ELGNCT(M,III,AK,BK,NT,VCOUNT,PMK)
C
C     GENERATES ELEMENT MATRIX,EK,FOR TRIANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      IF(IDARCY(M).EQ.0) GO TO 70
      IF(III.EQ.1)  GO TO 70
  500 CONTINUE
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      CALL CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
      GO TO 80
   70 CONTINUE
      PX=PMK
      ACONST=PX*ORX(M)*AREA(NT)
   80 CONTINUE
C
C                   CALCULATE STIFFNESS MATRIX
C     NORMAL STIFFNESS MATRIX MODIFIED BY THE FACTOR  OACONSTO
C
      DO 200 I=1,3
      DO 200 J=1,3
      EK(I,J)=ACONST*ELK(I,J,NT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE MERB3(N,M)
C
C     MERGES ELEMENT MATRIX EK INTO GROSS VECTOR VK.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
```

```
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VK(IPOS)=VK(IPOS)+EK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MERBD(N,M)
C
C     MERGES ELEMENT MATRIX D INTO GROSS VECTOR VD.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VD(IPOS)=VD(IPOS)+D(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE QMUL(NBW,QSUM)
C
C     COMPUTES NODAL FLUXES AT WELL BOUNDARY.
C
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DATA NREAD,NPRINT/1,3/
      DO 10 L=1,NBW
      I=JBD(L)
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
      QWB(L)=0.0
      DO 20 J = IS, IL
      K=I+J-IS
      QWB(L)=QWB(L)+VKQ(J)*H(K)
   20 CONTINUE
      IF(I.EQ.1) GO TO 15
      I1 = I-1
      IST= ISTART(I)
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
      QWB(L)=QWB(L)+VKQ(IP)*H(J)
   30 CONTINUE
   15 CONTINUE
      QSUM=QSUM+QWB(L)
      QWTEMP=-2.*3.1416*QWB(L)
      WRITE(NPRINT,4403)I,QWTEMP
 4403 FORMAT(/,10X,I5,15X,F12.4)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE QCALC(NBW,QSUM)
C
C     COMPUTES TOTAL DISCHARGE INTO THE WELL..
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BQCAL/VKQ(6C0),QWB(40)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,4103)
 4103 FORMAT(1H1,//,20X,29HNODAL FLUXES AT WELL BOUNDARY,///,
     1        10X,4HNODE,16X,14HDISCHARGE FLUX,/)
      QSUM=0.0
      CALL QMUL(NBW,QSUM)
      QSUM=ABS(2.*3.1416*QSUM)
      WRITE(NPRINT,4203)QSUM
 4203 FORMAT(////,10X,'TOTAL DISCHARGE INTO THE WELL  =',F10.4)
      RETURN
      END
      SUBROUTINE EBFIN3(LEN,NM,NN,LL)
C
C     SUBROUTINE TO GENERATE THE BANDWIDTHS OF THE BANDED SYMMETRIC
C     MATRIX. NBAND CONTAINS THE BANDWIDTHS, ID THE POSITION OF THE
C     TERM ON THE DIAGONALS OF THE ORIGINAL MATRIX, LEN IS THE LENGTH
```

```
C     THE VECTOR
C
      DIMENSION LV(3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      DO 20 I=1,LL
      NBAND(I)=1
      ISTART(I)=I
   20 CONTINUE
C
C     SCAN THROUGH THE LOCATION VECTOR FOR EACH MEMBER TO FIND
C     THE POSITION OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW
C
      NF2=3
      DO 25 I=1,NM
      DO 30 J=1,NF2
      LV(J)=NOD(I,J)
   30 CONTINUE
      DO 45 J=1,NF2
      IF(LV(J).EQ.0)GO TO 45
      DO 40 K=1,NF2
      IF(LV(J).GT.LV(K)) GO TO 40
      NW=LV(K)-LV(J)+1
      NR=LV(J)
      IF(NW.GT.NBAND(NR))  NBAND(NR)=NW
   40 CONTINUE
   45 CONTINUE
C      SEARCH FOR THE FURTHEREST OFF-LEFT TERM
C
      DO 55 J=1,NF2
      IF(LV(J).EQ.0) GO TO 55
      DO 65 K=1,NF2
      IF(LV(J).LT.LV(K)) GO TO 65
      NW=LV(K)
      NR=LV(J)
      IF(NW.LT.ISTART(NR)) ISTART(NR)=NW
   65 CONTINUE
   55 CONTINUE
   25 CONTINUE
C
C     SET-UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN=NBAND(1)
      DO 50 I=2,LL
      IF(NBAND(I)-NBAND(I-1).LT.0)  NBAND(I)=NBAND(I-1)-1
      LEN=LEN+NBAND(I)
   50 CONTINUE
      ID(1)=1
      DO 60 I=2,LL
   60 ID(I)=ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE QFLUX(NBM,NNODE)
C
C     MODIFIES MATRIX CK BY ADDING THE FLUX TERMS FOR NODES ON WELL BOUNDARY.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      DATA NREAD,NPRINT/1,3/
      IF(NBM.EQ.0) GO TO 20
      NBP=NBM+1
      DO 10 I=NBP,NNODE
      IST=ISTART(I)
      IF(IST.GT.NBM) GO TO 10
      I1=I-1
      DO 30 J=IST,NBM
      K=I1-J+1
      IP=ID(J)+K
      CK(I,1)=CK(I,1)-VK(IP)*CK(J,1)
   30 CONTINUE
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SYMSOL(LL,NLL,NBM)
C
C     AN IN CORE BAND SOLVER.
C     USE IN CONJUNCTION WITH TOPOLOGICAL SUBROUTINE FOR LARGE BANDWIDTH.
C
      DIMENSION VTEMP(90)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),N(300),IDUM(300)
      COMMON /BSOLV/C(300,1),V(2000)
      DOUBLE PRECISION TEMP
      JBOUN=NBM+1
      NBP=JBOUN
      ID=1
      IF(NBM.EQ.0) GO TO 150
      DO 100 I=1,NBM
```

```
      ID=ID+N(I)
  100 CONTINUE
  150 CONTINUE
      DO 10 I=NBP,LL
      TEMP=V(ID)
      NEB=ID+N(I)-1
      ID1=ID+1
      IF(I.EQ.JBD(JBOUN)) GO TO 16
C
C     NORMALISE ROW I
C
      KK=0
      V(ID)=1.0
      IF(ID1.GT.NEB) GO TO 29
      DO 20 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
   20 V(J)=V(J)/TEMP
   29 CONTINUE
      DO 25 L=1,NLL
      C(I,L)=C(I,L)/TEMP
   25 CONTINUE
      GO TO 46
   16 CONTINUE
      IF(ID1.GT.NEB) GO TO 39
      KK=0
      DO 120 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
  120 V(J)=0.0
   39 CONTINUE
      DO 125 L=1,NLL
  125 C(I,L)=DISP(JBOUN)
      JBOUN=JBOUN+1
C
C     ELIMINATION
C
   46 CONTINUE
      IDJ=ID
      IF(ID1.GT.NEB)  GO TO 35
      KK=0
      DO 30 J=ID1,NEB
      JI=J-ID1+I
      IDJ=IDJ+N(JI)
      KK=KK+1
      IF(VTEMP(KK)) 50,30,50
   50 CONTINUE
      IF(JBOUN.EQ.1) GO TO 240
      JBM1=JBOUN-1
      IF(I.EQ.JBD(JBM1)) GO TO 140
  240 IDP=J
      DO 40 K=IDP,NEB
      KJ=IDJ+K-J
   40 V(KJ)=V(KJ)-V(K)*VTEMP(KK)
  140 CONTINUE
      NJ=I+J-ID
      DO 32 L=1,NLL
      C(NJ,L)=C(NJ,L)-C(I,L)*VTEMP(KK)
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID=ID+N(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID=ID-1
      LLM1=LL-1
      LL1=LL-NBP
      DO 70 IB=1,LL1
      I=LLM1-IB+1
      ID=ID-N(I)
      IS=I+1
      IN=I+N(I)-1
      DO 80 J=IS,IN
      NJ=ID+J-I
      DO 75 L=1,NLL
      C(I,L)=C(I,L)-C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE VCHEC3(M,AK,BK,J1,J2,J3,NT,PMK,VCR,GRRX,GRRY)
C
C     COMPUTES ELEMENT VELOCITIES AND CHECK IF A PARTICULAR ELEMENT BELONGS TO
C     DARCY OR NON-DARCY FLOW ZONE.
C
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
```

```
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      GRRX=HX/HS
      GRRY=HY/HS
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE VDFB(LVEC)
C
C     GENERATES VECTOR VDTOP.
C
      DIMENSION D(2,2)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      NELTOP=LVEC-1
      NN=2
      ID(1)=1
      DO 10 I=2,LVEC
   10 ID(I)=ID(I-1)+2
      LENT=2*LVEC-1
      DO 108 I=1,LENT
      VDTOP(I)=0.0
  108 CONTINUE
      DO 208 IE=1,NELTOP
      IN=NDVEC(IE)
      IP=NDVEC(IE+1)
      RAVE=0.5*(X(IP,1)+X(IN,1))
      RDIF=X(IP,1)-X(IN,1)
      D(1,1)=RAVE*RDIF/3.0
      D(2,2)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      DO 308 I=1,2
      IK=IE+(I-1)
      DO 408 J=1,2
      JK=IE+(J-1)
      IF(IK.GT.JK) GO TO 408
      IPOS=ID(IK)+JK-IK
      VDTOP(IPOS)=VDTOP(IPOS)+D(I,J)
  408 CONTINUE
  308 CONTINUE
  208 CONTINUE
      RETURN
      END
      SUBROUTINE GVMOD(LACT,SY,IGK)
C
C     MODIFIES VECTORS GK AND VK TO TAKE ACCOUNT OF LEAKAGE FLUX ACROSS TOP
C     BOUNDARY.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DATA NREAD,NPRINT/1,3/
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      J2=J1
      JM=J1
      IF(L.LT.LACT) J2=NDVEC(L+1)
      IF(L.GT.1) JM=NDVEC(L-1)
      SUM=VF1*HINT(J1,1)+VF2*HINT(J2,1)+VFM*HINT(JM,1)
      IF(IGK.GT.0) GK(J1,1)=GK(J1,1)+SUM
      IS=ID(I)
      VK(IS)=VK(IS)+VF1
      IL=IS+NBAND(I)-1
      VK(IL)=VK(IL)+VF2
 1078 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TCURV3(HO,B,NNODE,QFIX,AK,BK,PMK,TH,J)
C
C     COMPUTES TYPE CURVE FOR STEADY STATE FLOW.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      TWPI=44./7.
      TMIS=PMK*TH
      PARAM=BK*QFIX*TMIS/(TH**2*B*TWPI)
      DENOM=QFIX/(TWPI*TMIS)
      WRITE(3,3)PARAM
    3 FORMAT(///,10X,50('*'),/,10X,'*',13X,'STEADY STATE TYPE CURVE',
     1      12X,/,10X,50('*'),///,15X,' NON-LINEAR FACTOR    =',F12.4)
      IF(J.EQ.2) GO TO 5
      WRITE(3,23)
      GO TO 15
    5 CONTINUE
      WRITE(3,33)
   15 CONTINUE
   23 FORMAT(//,10X,'NODE NUMBER',15X,'R-COORDINATE',10X,'FUNCTION W(U)'
     1           ,10X,'ARGUMENT U',//)
   33 FORMAT(//,10X,'NODE NUMBER',15X,'Z-COORDINATE',10X,'FUNCTION W(U)'
     1           ,10X,'ARGUMENT U',//)
      DO 10 I=1,NNODE
      SDRAW=HO-H(I)
      SLESS=SDRAW/DENOM
      RLESS=X(I,1)/B
      WRITE(3,13)I,X(I,J),SLESS,RLESS
   10 CONTINUE
   13 FORMAT(10X,I5,20X,F10.2,2(10X,E12.4))
      RETURN
      END
      SUBROUTINE EBOUND(NELEM,AK,BK,PMK,HW,HO,QSUM,TH)
C
C     COMPUTES UPPER BOUND OF TOTAL DISSIPATION ENERGY.
C
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      CONST=PMK*TH*(HO-HW)**2
      ENERGY=0.0
      DO 10 I=1,NELEM
      NT=NREP(I)
      VSQ=VEL(I)**2
      IF(IDARCY(I).EQ.0) GO TO 15
      VCU=VSQ*VEL(I)
      TERM=0.5*AK*VSQ+2.*BK*VCU/3.
      TERM=TERM*AREA(NT)*ORX(I)
      GO TO 20
   15 CONTINUE
      TERM=0.5*VSQ/PMK
   20 CONTINUE
      ENERGY=ENERGY+TERM
   10 CONTINUE
      ENERGY=ENERGY/CONST
      WRITE(3,13)CONST,ENERGY
   13 FORMAT(///,20X,'RESULT OF DISSIPATION ENERGY CALCULATION.',//,
     1           10X,'CONSTANT FACTOR    =',E12.4,3X,'FT.**4/MIN.',//,
     2           10X,'DIMENSIONLESS DISSIPATION ENERGY    =',E12.4)
      RETURN
      END
      SUBROUTINE HYPROP(I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,AL,BL,PML,VCL)
C
C     IDENTIFIES HYDRAULICS PROPERTIES OF EACH ELEMENT.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      IF(IPROP(I).EQ.0) GO TO 10
      AL=AK
      BL=BK
      PML=PMK
      VCL=VCR
      GO TO 20
   10 CONTINUE
      AL=AGP
      BL=BGP
      PML=PMGP
      VCL=VGP
   20 CONTINUE
      RETURN
      END
```

4.4 LISTING OF TRCON3

```
C
C      INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C      DEVELOPED BY P.S. HUYAKORN.
C
C      TRCON3,PROGRAM FOR SOLVING TRANSIENT,TWO-DIMENSIONAL,DARCY OR TWO-REGIME
C
C      FLOW USING TRIANGULAR ELEMENTS.
C
C      VERSION DATED OCTOBER,1973.
C
C       FOR FURTHER INFORMATION,CONTACT
C
C         P.S. HUYAKORN OR C.R. DUDGEON
C
C         WATER RESEARCH LABORATORY
C
C         KING ST.,MANLY VALE
C
C         SYDNEY,N.S.W. 2093,AUSTRALIA.
C
C----------------------------
C      LIST OF INPUT VARIABLES
C----------------------------
C
C--------------ALL READ STATEMENTS ARE LOCATED IN THE MAIN------------------
C
C---------------PROGRAM AND INDICATED BY  $1 OR $1,SK  SIGN-----------------
C
C***   PROBLEM VARIABLES  ***
C
C      NPROB    =  NUMBER OF PROBLEMS TO BE SOLVED
C      IVEL     =  VELOCITY PRINT-OUT INDEX
C                  FEED IN IVEL=0 IF VELOCITY PRINT-OUT IS NOT REQUIRED OTHERWISE
C                  FEED IN IVEL=1
C      IDISCR   =  DISCRETISATION DATA PRINT-OUT INDEX
C                  FEED IN IDISCR=0 IF DISCRETISATION PRINT-OUT IS NOT REQUIRED
C                  OTHERWISE FEED IN IDISCR=1
C      ORELAX   =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                  SUGGESTED VALUE LIES BETWEEN 1.50 TO 1.85
C      RW       =  RADIUS OF WELL SCREEN
C      RO       =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C      HO       =  INITIAL HEIGHT OR DRAWDOWN OF WATER TABLE
C      HTOL     =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                  SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C      NLAYR    =  NUMBER OF LAYERS OF WATER BEARING FORMATIONS
C      QFIX     =  PRESCRIBED WELL DISCHARGE
C      RCSNG    =  RADIUS OF WELL CASING
C      QRTOL    =  RATIO OF DISCHARGE TOLERANCE TO PRESCRIBED WELL DISCHARGE
C                  SUGGESTED VALUE IS 0.01 OR 0.02
C      IGP      =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                  IGP=0 FOR NON-GRAVEL PACKED WELL
C      IBOUND   =  EXTERNAL BOUNDARY INDEX
C                  IBOUND=0 FOR BARRIER BOUNDARY OTHERWISE IBOUND =1
C      IWBC     =  WELL BOUNDARY CONDITION INDEX
C                  IWBC=0 IF EFFECT OF WELL STORAGE IS TO BE NEGLECTED OTHERWISE
C                  IWBC=1
C      IKMAX    =  LAYER OF MAXIMUM PERMEABILITY OR MINIMUM VALUE OF AKL(I)
C      IWAT     =  INDEX USED TO INDICATE WHETHER THE TOP LAYER IS CONFINED OR
C                  UNCONFINED
C      THL(I)   =  THICKNESS OF LAYER NUMBER I
C      AKL(I)   =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER OR AQUITARD
C                  LAYER I
C                  FOR AQUITARD I,FEED IN AKL(I)=1./PML(I) WHERE PML(I) IS ITS
C                  HYDRAULIC CONDUCTIVITY
C      BKL(I)   =  FORCHHEIMER NON-LINEAR COEFFICIENT OF AQUIFER LAYER I
C                  FOR AQUITARD I,FEED IN BKL(I)=0.0
C      VCRL(I)  =  CRITICAL VELOCITY OF AQUIFER LAYER I
C                  FOR AQUITARD I FEED IN VCRL(I)=0.0
C      SSL(I)   =  SPECIFIC STORAGE OF AQUIFER OR AQUITARD I
C      AGP      =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C      BGP      =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C      VGP      =  CRITICAL FLOW VELOCITY FOR GRAVEL PACK MATERIAL
C      THGP     =  THICKNESS OF GRAVEL PACK
C      BTGP     =  HEIGHT OF THE BASE OF GRAVEL PACK
C      SY       =  COEFFICIENT OF SPECIFIC YIELD OF TOP LAYER WATER TABLE AQUIFER
C                  OR AQUITARD
C      DINDEX   =  RECIPROCAL OF DELAYED YIELD INDEX FOR WATER TABLE AQUIFER OR
C                  AQUITARD
C
C***   DISCRETISATION PARAMETERS  ***
C
C      NTICR    =  TOTAL NUMBER OF TIME STEPS
C      ITST     =  STARTING TIME STEP NUMBER
C      TFACTR   =  VALUE OF THE FIRST TIME STEP,EXPRESSED IN DIMENSIONLESS FORM
C      TMUL     =  TIME MULTIPLIER,SUGGESTED VALUE VARIES FROM 1.50 TO 2.00
C      DTMUL    =  INCREMENT OF TIME MULTIPLIER
C                  SUGGESTED VALUE LIES BETWEEN 0.01 TO 0.02
C      FRLEN    =  LENGTH OF FIRST SUBREGION
C                  FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
```

```
C      SCFAC    =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                  SUBREGIONS.SUGGESTED VALUE IS SCFAC=1.50
C      XLMAX    =  MAXIMUM LENGTH OF A BLOCK,PRESCRIBED TO AVOID ILL-CONDITIONED
C                  ELEMENTS.MAXIMUM VALUE OF XLMAX SHOULD NOT EXCEED 25.*TH
C      NDSC     =  TOTAL NUMBER OF NODES ON WELL SCREEN(S)
C                  NDSC IS TO BE GREATER THAN OR EQUAL TO 2
C      NSCREN   =  NUMBER OF SCREENED INTERVALS
C      IREGL(I)=   NUMBER OF REPEATED REGULAR BLOCKS WITH THE SAME NUMBER OF NODES
C         -1       ON THE LEFT AND RIGHT VERTICAL LINES ACROSS LAYER I
C                  SUGGESTED VALUE IS IREG=2
C      NMINL(I)=   MINIMUM NUMBER OF NODES ALONG A VERTICAL LINE ACROSS LAYER I
C                  TO MINIMISE THE TOTAL NUMBER OF NODES,SUGGEST NMIN=2
C      NFRL(I)  =  NUMBER OF NODES ON PORTION OF WELL BOUNDARY PENETRATING AQUIFER
C                  OR AQUITARD LAYER I
C      XSCR(I)  =  Z-COORDINATE OF BASE OF SCREEN I ABOVE DATUM
C      HSCR(I)  =  LENGTH OF WELL SCREEN NUMBER I
C-----------------------------
C      LIST OF OUTPUT VARIABLES
C-----------------------------
C      NNODE    =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C      NELEM    =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C      IT       =  TIME STEP NUMBER
C      TMM      =  REAL TIME VALUE AT THE MID-POINT OF TIME STEP IT
C      TM       =  REAL TIME VALUE AT THE END OF TIME STEP IT
C      X        =  RADIAL AND VERTICAL NODAL COORDINATES
C      H        =  NODAL HEAD OR DRAWDOWN VALUES
C      TLESS    =  NODAL VALUES OF DIMENSIONLESS TIME,1/U
C      HLESS    =  NODAL VALUES OF WELL FUCTION FOR TRANSIENT FLOW,W(U)
C      IQ       =  DISCHARGE ITERATION NUMBER
C      SW       =  DRAWDOWN VALUE AT CURRENT TIME
C      QAQFR    =  DISCHARGE FROM AQUIFER INTO WELL AT DRAWDOWN SW
C      QSTRGE   =  DISCHARGE FROM WELL STORAGE
C      QCALL    =  TOTAL CALCULATED DISCHARGE
C                  QCALL = DISCHARGE FROM WELL STORAGE + QAQFR
C      QRDIF    =  RESIDUAL DISCHARGE
C                  QRDIF IS THE ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN
C                  PRESCRIBED DISCHARGE AND CALCULATED DISCHARGE
C      NOD      =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C      VEL      =  ABSOLUTE ELEMENT VELOCITIES
C      VCOMP1   =  RADIAL COMPONENT OF ELEMENT VELOCITY
C      VCOMP2   =  VERTICAL COMPONENT OF ELEMENT VELOCITY
C      IDARCY   =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                  OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                  IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
      DIMENSION JDARCY(500)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      COMMON /ALAYR/AKL(3),BKL(3),VCRL(3),SSL(3),THL(3),IREGL(3),
     1NMINL(3),NFRL(3),PML(3)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BQCAL/VS(600),QWB(40)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ACORE/VCORE(2000)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DOUBLE PRECISION DEXP,ARGU
      DATA NREAD,NPRINT/1,3/
C
C      **************
C      *            *
C      *  BLOCK 1   *
C      *            *
```

```
C      **************
C
C      PRINT INITIAL HEADINGS
C
       WRITE(NPRINT,1773)
 1773 FORMAT('1')
       WRITE(NPRINT,1003)
       WRITE(NPRINT,1013)
       WRITE(NPRINT,1023)
       WRITE(NPRINT,1013)
       WRITE(NPRINT,1033)
       WRITE(NPRINT,1013)
       WRITE(NPRINT,1003)
  993 FORMAT('1',4X,51H *************************************************
     1**)
 1003 FORMAT(5X,51H ***************************************************)
 1013 FORMAT(5X,51H *                                                 *)
 1023 FORMAT(5X,51H *     FINITE ELEMENT ANALYSIS OF WELL PROBLEMS    *)
 1033 FORMAT(6X,'* TRANSIENT,TWO REGIME FLOW TOWARD A SINGLE WELL *')
       READ(NREAD,1011)NPROB,IVEL,IDISCR,ORELAX                        $1
 1011 FORMAT(3I10,F10.2)
       DO 4800 JPRO=1,NPROB
       WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1           I6,12X,'*',/,20X,50('*'))
C
C      READ AND PRINT GENERAL DATA.
C
       READ(NREAD,2001)RW,RO,HO,HTOL,NLAYR                             $1
 2001 FORMAT(4F10.2,I10)
       READ(NREAD,201)QFIX,RCSNG,QRTOL                                 $1
  201 FORMAT(3F10.2)
       READ(NREAD,111)IGP,IBOUND,IWBC,IKMAX,IWAT                       $1
  111 FORMAT(5I10)
       IAQTA=IWAT
       WRITE(NPRINT,23)RW,RO,HO,RCSNG
       WRITE(NPRINT,2193)QFIX,QRTOL,ORELAX,HTOL
       WRITE(NPRINT,193)IGP,IBOUND,IWBC,IKMAX,IWAT
   23 FORMAT(///,20X,'GENERAL DATA',///,
     1           10X,'RADIUS OF WELL   =',F10.2,//,
     2           10X,'RADIUS OF INFLUENCE   =',F10.2,//,
     3           10X,'HEIGHT OF WATER TABLE   =',F10.2,//,
     4           10X,'RADIUS OF WELL CASING   =',F10.2,//)
 2193 FORMAT(10X,'DISCHARGE INTO WELL   =',F10.2,//,
     1           10X,'DISCHARGE TOLERANCE RATIO   =',F12.4,//,
     2           10X,'OVER RELAXATION FACTOR   =',F12.4,//,
     3           10X,'HEAD TOLERANCE          =',F12.4,//)
  193 FORMAT(10X,'GRAVEL PACK INDEX   =',I5,//,
     1           10X,'BOUNDARY INDEX   =',I5,//,
     2           10X,'WELL B.C. INDEX   =',I5,//,
     3           10X,'LAYER OF MAX PERMEABILITY   =',I5,//,
     4        10X,'WATER TABLE AQUIFER INDEX  =',I5)
       QRTOL=QRTOL*QFIX/2.0
       IF(IWBC.EQ.0) RCSNG=0.0
C
C      READ MATERIAL DATA.
C
       WRITE(NPRINT,6013)
 6013 FORMAT(///,10X,'FORMATION PROPERTIES',//,
     1           10X,'LAYER NO.',5X,'THICKNESS',5X,'COEFF-A',
     2  5X,'COEF-B',5X,'CRIT. VELOCITY',5X,'SPECIFIC STORAGE',//)
       DO 17 I=1,NLAYR                                                 *NLAYR**
       READ(NREAD,3013)THL(I),AKL(I),BKL(I),VCRL(I),SSL(I)             $1
 3013 FORMAT(5E10.2)
       GCR=AKL(I)*VCRL(I)+BKL(I)*VCRL(I)**2
       PML(I)=1./AKL(I)
       IF(GCR.GT.0.) PML(I)=VCRL(I)/GCR
       WRITE(NPRINT,7013)I,THL(I),AKL(I),BKL(I),VCRL(I),SSL(I)
 7013 FORMAT(10X,I5,9X,F9.2,3(5X,E9.2),9X,E9.2)
       SSL(I)=SSL(I)*1000.                                             /*
   17 CONTINUE
       RGP=RW
       IF(IGP.EQ.0) GO TO 29
       READ(NREAD,331)AGP,BGP,VGP,THGP,BTGP                            $1,SK
  331 FORMAT(5F10.3)
       RGP=THGP+RW
       GRGP=AGP*VGP+BGP*VGP**2
       PMGP=1./AGP
       IF(GRGP.NE.0) PMGP=VGP/GRGP
       WRITE(NPRINT,153)AGP,BGP,VGP,PMGP
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1           10X,'COEFFICIENT A =',F10.4//
     2           10X,'COEFFICIENT B = ',F10.4//
     3           10X,'CRITICAL VELOCITY = ',F10.4//
     4           10X,'COEFFICIENT K = ',F10.4//)
       WRITE(NPRINT,163)THGP,RGP
  163 FORMAT(10X,'THICKNESS OF PACK = ',F10.2//
     1           10X,'RADIUS OF PACK = ',F10.2//)
   29 CONTINUE
C
```

```
C     FOR WATER TABLE AQUITARD,READ DELAYED YIELD DATA.
C
      IF(IAQTA.EQ.0) GO TO 200
      READ(NREAD,213)SY,DINDEX                                          1$,SK
  213 FORMAT(2E10.2)
      SFAC=DINDEX*SY/6.
      DINDEX=DINDEX/1000.
  200 CONTINUE
      READ(NREAD,121) NTICR,ITST,TFACTR,TMUL,DTMUL                     $1
  121 FORMAT(2I10,3F10.2)
      NNODS=1
C
C     GENERATE ELEMENTS OF VECTOR TIME.
C
      TH=THL(IKMAX)
      PM=PML(IKMAX)
      BK=BKL(IKMAX)
      SS=SSL(IKMAX)
      BKMAX=BK
      TPAT=1
      NTICP=NTICR+1
      CALL TIGEN(NTICP,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX,TPAT)
C
C     READ AND PRINT DISCRETIZATION DATA.
C
      READ(NREAD,901)FRLEN,SCFAC,XLMAX                                 $1
  901 FORMAT(3F10.2)
      READ(NREAD,801)NDSC,NSCREN                                       $1
  801 FORMAT(2I10)
      IPENTR= 0
      DO 602 I=1,NLAYR                                                 *NLAYR**
      READ(NREAD,1901)IREGL(I),NMINL(I),NFRL(I)                        $1
 1901 FORMAT(3I10)
  602 CONTINUE                                                         /*
C
C     READ WELL SCREEN DATA.
C
      SCLEN=0.
      WRITE(NPRINT,553)
  553 FORMAT(/,10X,'SCREEN NO.',10X,'BASE HEIGHT',14X,'LENGTH',//)
      DO 702 I=1,NSCREN                                                *NSCREN*
      READ(NREAD,601)XSCR(I),HSCR(I)                                   $1
  601 FORMAT(2F10.2)
      WRITE(NPRINT,563)I,XSCR(I),HSCR(I)
  563 FORMAT(10X,I6,14X,F11.2,10X,F10.2)
      SCLEN=SCLEN+HSCR(I)
  702 CONTINUE                                                         /*
C
C     GENERATE AND PRINT DISCRETIZATION DATA.
C
      CALL GXNODR(FRLEN,SCFAC,XLMAX,RW,RO,NNODE,NELEM,LVEC,NLAYR,IPENTR,
     1IDISCR)
C
C     GENERATE IPROP
C
      DO 1544 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      XCEN=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      IF(XCEN.LT.RGP) IPROP(I)=0
 1544 CONTINUE
      ROMAX=RO
C
C     FORM GROSS VECTOR VDTOP.
C
      IF(IAQTA.EQ.0) GO TO 166
      CALL VDFB(LVEC)
      DO 176 I=1,NNODE
      HINT(I,1)=0.0
  176 CONTINUE
  166 CONTINUE
      NN=3
      CALL EBFIN3(LEN,NELEM,NN,NNODE)
      WRITE(NPRINT,3233)LEN
 3233 FORMAT(////,10X,'LENGTH OF GROSS VECTOR =',I8,//)
      DO 130 I=1,LEN
      VD(I)=0.0
  130 CONTINUE
      IELST=1
      DO 632 M=1,NELEM
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
  632 CONTINUE
C
C     CALCULATE HEAD AT VARIOUS TIME VALUES
C
      TMIS=TH*PM
```

```
      FPI=88./7.
      CUNST=QFIX/(FPI*TMIS)
      DIFFUS=PM/SS
      LVST=2
      NST=NDVEC(1)
      LSTEL=NDVEC(1)
      NELTO=NELEM
      DO 642 I=1,LVEC
      HCOLD(I)=0.
      HF(I)=0.
      FINT(I,1)=0.
  642 QLEAK(I)=0.
      IF(ITST.EQ.1) GO TO 1245
      CALL HREAD(NNODS,1,1)
      CALL HPUNCH(NNODS,3,1)
      CALL HREAD(LVEC,1,2)
      CALL HPUNCH(LVEC,3,2)
      IST=NNODS+1
      DO 1345 I=IST,NNODE
      H(I)=HO
 1345 CONTINUE
      GO TO 1445
 1245 CONTINUE
      DO 3545 I=1,NNODE
      H(I)=HO
 3545 CONTINUE
 1445 CONTINUE
      DO 245 I=1,NNODE
      HINT(I,1)=H(I)
  245 CONTINUE
      DELT=TIME(1)
      NDTO=NNODE
      ITMIN=NTICR
      QAQFR=2.*QRTOL
      SWOLD=0.0
C
C     ESTIMATE WELL DRAWDOWN FOR FIRST TIME STEP.
C
      CALL DSWINT(DSOLD,BK,QFIX,TMIS,TH,RW,SCLEN,DIFFUS,CUNST)
      DSOLD=0.6*DSOLD
      IF(NLAYR.EQ.1) DSOLD=1.2*DSOLD
      IST=1
      FQ=0.0
      NQITER=5
C
C     SET JDARCY.
C
      DO 299 I=1,NELEM
      JDARCY(I)=0
  299 CONTINUE
C
C     LOOPING WITH LOOP PARAMETER IT=1,NTICR
C
      DO 7007 IT=ITST,NTICR
      ITCUR=IT
      IF(IT.GT.1) DELT=TIME(IT)-TIME(IT-1)
      TM=TIME(IT)
      LVST=LVEC
      NST=NNODE
      TMM=TM-DELT*0.5
      WRITE(NPRINT,683)IT
  683 FORMAT(///,10X,35('*'),/,10X,'*',4X,
     1            'TIME STEP NUMBER  =',I5,5X,'*',/,10X,35('*'))
      TMIL=TMM/1000.
      WRITE(NPRINT,333)TMIL
  333 FORMAT(///,10X,41('*'),/,10X,'*',9X,'TIME  =',
     1            E14.3,9X,'*',/,10X,41('*'),//)
      WRITE(NPRINT,343)RO,NNODE,NELEM,LVST
  343 FORMAT(/,10X,'ESTIMATED RADIUS OF INFLUENCE  =',F10.2,
     1//,10X,'CORRESPONDING NO. OF NODES  =',I5,//,
     210X,'CORRESPONDING NO. OF ELEMENTS   =',I5,//,
     310X,'CORRESPONDING COMPONENT OF VECTOR NDVEC   =',I5,//)
C
C     COMPUTE DIMENSIONLESS TIME.
C
      CONS=4.0*DIFFUS*TMM
      DO 709 I=1,NNODE
      TLESS(I)=CONS/X(I,1)**2
  709 CONTINUE
C
C     FOR WATER TABLE AQUITARD COMPUTE BOUTON'S INCREMENTAL DELAYED YIELD.
C
      IF(IAQTA.EQ.0) GO TO 777
      ARGU=-DINDEX*DELT*0.5
      TERM=DEXP(ARGU)
      ES=SY*(1.-TERM)
      WRITE(NPRINT,903)ES
      ES=ES*1000.
  903 FORMAT(///,1X,'EFFECTIVE SPECIFIC YIELD  =',E14.4)
  777 CONTINUE
```

```
C
C     INITIALISE HEAD VALUES.
C
      IF(IT.EQ.1) GO TO 165
  677 CONTINUE
      DO 155 I=1,NNODE
      HINT(I,1)=H(I)
  155 CONTINUE
  165 CONTINUE
      CALL EBFIN3(LEN,NELEM,NN,NNODE)
C
C     FORM GROSS VECTOR VD.
C
      IF(IELST.GT.NELEM) GO TO 1235
      NT=0
      DO 135 IE=IELST,NELEM
      IF(IPROP(IE).NE.0) L=IPROP(IE)
      SC=SSL(L)
      CALL ELGND3(NT,SC,IE)
      CALL MERBD(NN,IE)
  135 CONTINUE
      IELST=NELEM+1
 1235 CONTINUE
      DO 2722 I=LVST,LVEC
      IF(NST.GT.NDVEC(I)) GO TO 2722
      JJ=NDVEC(I-1)
      JF=I
      GO TO 2822
 2722 CONTINUE
 2822 CONTINUE
      LMB=NDVEC(JF)-JJ
C
      CALL VECMUL(NNODE,LEN)
C
C     FOR WATER TABLE AQUITARD,COMPUTE BOULTON'S DELAYED YIELD.
C
      IF(IAQTA.EQ.0) GO TO 278
      CALL GVMOD(LVST,ES,1)
      IF(IT.EQ.1) GO TO 278
      FTERM=0.5*DELT*SFAC
      CALL GKMOD(LVST,FTERM)
  278 CONTINUE
      HW=HO
      NDW=NDVEC(1)
      NDRO=LMB
      CALL BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,RW,NSCREN)
C
C     STORE ELEMENTS OF MATRIX GK FOR DISCHARGE COMPUTATION.
C
      DO 1378 I=1,NBW
      JJ=JBD(I)
 1378 GP(I)=GK(JJ,1)
C
C     SET UP LOOP FOR WELL DISCHARGE ITERATION.
C
      DO 998 IQ=1,NQITER
C
C     ADJUST VALUE OF WELL DRAWDOWN.
C
C
C     SET IDARCY=JDARCY.
C
      DO 99 I=1,NELEM
      IDARCY(I)=JDARCY(I)
   99 CONTINUE
C
C     RESET HEAD VALUES.
C
      DO 1559 I=1,NNODE
      H(I)=HINT(I,1)
 1559 CONTINUE
      WRITE(NPRINT,5003)
 5003 FORMAT(/,5X,30('- '))
      CALL FQSET(IT,ITCUR,DELT,FQ,TPAT)
      CALL SWMOD(IQ,IT,RCSNG,DELT,QFIX,QAQFR,SWOLD,DSOLD,TPAT,FQ)
      WRITE(NPRINT,833)IQ,DSW(IQ),SW(IT)
  833 FORMAT(///,10X,'DISCHARGE ITERATION NUMBER   =',I10,//,
     1              10X,'DRAWDOWN INCREMENT   =',F10.3,//,
     2              10X,'DRAWDOWN VALUE    =',F10.3,//)
      NBIS=NBW
      IF(XSCR(1).GT.0.) NBIS=0
      NBP=NBIS+1
      IF(NBP.EQ.1) GO TO 94
      DO 117 I=1,NBW
  117 CK(I,1)=SW(IT)+HO
   94 CONTINUE
      HW=HO+SW(IT)
      CALL BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,RW,NSCREN)
C
C     SET UP LOOP FOR ITERATION ON HEAD
```

```
C
      NITER=5
      IF(BKMAX.LE.0.0) NITER=1
      VCOUNT=0
      DO 999 III=1,NITER
C
C     ZERO GROSS STIFFNESS MATRIX AND LOAD MATRIX
C
      DO 300 I=1,LEN
      VCORE(I)=0.0
  300 CONTINUE
      DO 305 I=NBP,NNODE
      CK(I,1)=GK(I,1)
  305 CONTINUE
C
C      COMPUTE STIFFNESS MATRIX FOR EACH ELEMENT AND MERGE
C
C      FOR III=1  ONLY PERFORM SOLUTION FOR DARCY CASE
C
      NN=3
      DO 350 I=1,NELEM
      NT=NREP(I)
      IF(IPROP(I).EQ.0) GO TO 755
      L=IPROP(I)
      AKK=AKL(L)
      BKK=BKL(L)
      PKK=PML(L)
      GO TO 365
  755 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
  365 CONTINUE
      CALL ELGNCT(I,III,AKK,BKK,NT,VCOUNT,PKK)
      CALL MERB3(NN,I)
  350 CONTINUE
      DO 978 I=1,LEN
      VK(I)=0.0
  978 CONTINUE
      IF(IAQTA.EQ.0) GO TO 555
      CALL GVMOD(LVST,ES,0)
  555 CONTINUE
      DO 530 I=1,LEN
      VK(I)=VCORE(I)*DELT*0.5+VD(I)+VK(I)
  530 CONTINUE
C
C     FIX THE WELL DRAWDOWN.
C
      NLEN=0
      JJ=JBD(NBW)
      DO 98 I=1,JJ
      NLEN=NLEN+NBAND(I)
   98 CONTINUE
      DO 378 I=1,NLEN
  378 VS(I)=VK(I)
C
C     SOLVE EQNS BY BANDED ELIMINATION SCHEME
C
      NLL=1
      NBM=NBP-1
      CALL FQFLUX(NBM,NNODE)
      NBX=NBW
      IF(IBOUND.EQ.1) NBX=NBD
      CALL SYMSOL(NNODE,NLL,NBM,NBX)
C
C     PRINT OUT SOLUTION FOR FUNCTION
C
      IF(III.EQ.1) GO TO 900
  415 CONTINUE
      NCOUNT=0
      EMAX=0.0
      DO 450 I=1,NNODE
      EPSI=CK(I,1)-H(I)
      EPSIS=EPSI
      IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.HTOL)  GO TO 460
      NCOUNT=NCOUNT+1
      H(I)=H(I)+ORELAX*EPSIS
      GO TO 450
  460 CONTINUE
      H(I)=CK(I,1)
  450 CONTINUE
      IF(NCOUNT.EQ.0) GO TO 1000
      GO TO 1999
  900 CONTINUE
      DO 950 I=1,NNODE
      H(I)=CK(I,1)
      HLESS(I)=(HO-H(I))/CUNST
      IF((ITCUR.GT.IT).AND.(IQ.EQ.1)) TLESS(I)=TLESS(I)*TM/THM
  950 CONTINUE
```

```
 1999 CONTINUE
      IF(NITER.EQ.1) GO TO 999
C
C     REGENERATE IDARCY
C
       DO 199 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      NT=NREP(I)
      IF(IPROP(I).NE.0) L=IPROP(I)
      AK=AKL(L)
      BK=BKL(L)
      PMK=PML(L)
      VCR=VCRL(L)
      CALL SETARG(AGP,BGP,PMGP,VGP,AK,BK,PMK,VCR,I)
      CALL VCHEC3(I,AK,BK,J1,J2,J3,NT,PMK,VCR,HRRX,HRRY)
  199 CONTINUE
  999 CONTINUE
 1000 CONTINUE
      WRITE(NPRINT,413)III
  413 FORMAT(//,10X,'NUMBER OF ITERATIONS REQUIRED  =',I5)
      WRITE(NPRINT,473)NCOUNT,EMAX
  473 FORMAT(//,10X,'TOLERANCE COUNTER FOR HEAD    =',I3,//,
     1            10X,'ABSOLUTE MAXIMUM ERROR IN HEAD    =',F12.4)
C
C     CHECK FOR ACCURACY OF DISCHARGE RATIO.
C
      DWD=DSW(IQ)
      CALL AQDIS(NBW,QAQFR,QRDIF,DELT,TH,RCSNG,IQ,QFIX,DWD)
C
C
C     PRINT FINAL DISCHARGE VALUES.
C
      QCALL=QCALC(IQ)
      QSTRGE=QCALL-QAQFR
      WRITE(NPRINT,1203)QAQFR,QSTRGE,QCALL,QRDIF
 1203 FORMAT(///,10X,'DISCHARGE FROM AQUIFER INTO WELL   =',F12.4,//,
     1          10X,'DISCHARGE FROM WELL STORAGE =',F14.3,//,
     1           10X,'TOTAL CALCULATED DISCHARGE   =',F14.3,//,
     210X,'RESIDUAL DISCHARGE          =',F10.4)
      IF(QRDIF.LE.QRTOL) GO TO 1102
  998 CONTINUE
 1102 CONTINUE
C
C     RESET IDARCY.
C
      DO 2599 I=1,NELEM
      JDARCY(I)=IDARCY(I)
 2599 CONTINUE
      SWOLD=SW(IT)
      DO 470 I=1,NNODE
      HLESS(I)=(HO-H(I))/CUNST
  470 CONTINUE
      CALL ROUT(NNODE)
C
C     COMPUTE AND PRINT ELEMENT VELOCITIES
C
      IF(IVEL.EQ.0) GO TO 557
      WRITE(NPRINT,4203)
 4203 FORMAT('1',///,20X,'*******************************************'/
     1                20X,'*           ELEMENT VELOCITIES              *'/
     2                20X,'*******************************************'///
     310X,'ELEM NO.',10X,'RADIAL VEL',10X,'VERTIC VEL',10X,'IDARCY',///)
      DO 3000 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      NT=NREP(I)
      IF(IPROP(I).NE.0) L=IPROP(I)
      AK=AKL(L)
      BK=BKL(L)
      PMK=PML(L)
      VCR=VCRL(L)
      CALL SETARG(AGP,BGP,PMGP,VGP,AK,BK,PMK,VCR,I)
      CALL VCHEC3(I,AK,BK,J1,J2,J3,NT,PMK,VCR,HRRX,HRRY)
      VCOMP1=-VEL(I)*HRRX
      VCOMP2=-VEL(I)*HRRY
      WRITE(NPRINT,1103)I,VCOMP1,VCOMP2,IDARCY(I)
 1103 FORMAT(10X,I5,2(10X,F10.3),10X,I5)
 3000 CONTINUE
  557 CONTINUE
      IF(IT.GT.ITMIN) GO TO 577
      IF(ITCUR.GT.IT) GO TO 7000
      TMIL=TM/1000.
      WRITE(NPRINT,333)TMIL
      ITCUR=IT+1
      GO TO 677
  577 CONTINUE
      TMIL=TM/1000.
```

```
      WRITE(NPRINT,333)TMIL
C
C     OBTAIN HEAD VALUES AT THE END OF TIME INTERVAL BY EXTRAPOLATION.
C
      DO 477 I=1,NNODE
      TLESS(I)=TLESS(I)*TM/TMM
      H(I)=2.0*H(I)-HINT(I,1)
      HLESS(I)=(HO-H(I))/CUNST
  477 CONTINUE
      SWTEMP=SWOLD
      SWOLD=H(1)-HO
      QFR=QAQFR
      DWTEMP=SWOLD-SWTEMP
      QAQFR=QFR*SWOLD/SWTEMP
      ACSNG=22.*RCSNG**2/7.
      TRM=ABS(ACSNG*DWTEMP*2./DELT)
      QCALL=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALL)
      SW(IT)=SWOLD
      QWSTR=QCALL-QAQFR
      WRITE(NPRINT,1203)QAQFR,QWSTR,QCALL,QRDIF
 7009 CONTINUE
      CALL ROUT(NNODE)
 7000 CONTINUE
      IF(IAQTA.EQ.0) GO TO 6007
      DO 9007 I=1,LVST
      L=NDVEC(I)
      HF(I)=H(L)
      FINT(I,1)=HINT(L,1)
 9007 CONTINUE
      CALL BSIMP(LVST,TH,DINDEX,SY,DELT,IT,0)
      DO 8007 I=1,LVST
      L=NDVEC(I)
      HCOLD(I)=H(L)
 8007 CONTINUE
 6007 CONTINUE
 7007 CONTINUE
C
C     PUNCH OUT SOLUTION AT FINAL TIME.
C
      CALL HPUNCH(NNODE,2,1)
      CALL HPUNCH(NNODE,2,2)
 5000 CONTINUE
 4800 CONTINUE
      STOP
      END
      SUBROUTINE GXNODR(FRLEN,SCFAC,XLMAX,RW,RO,NNODE,NELEM,LVEC,NLAYR,
     1IPENTR,IDISCR)
C
C     GENERATES DISCRETIZATION DATA.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      COMMON /ALAYR/AKL(3),BKL(3),VCRL(3),SSL(3),THL(3),IREGL(3),
     1NMINL(3),NFRL(3),PML(3)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      DATA NPRINT/3/
      CALL DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
      NRST=1
      THTO=0.
      NSTFR=1
      NDWB(1)=0
      DO 10 I=1,NLAYR
      THTO=THTO+THL(I)
      NFR=NFRL(I)
      NDWB(1)=NDWB(1)+NFR
      IREG=IREGL(I)
      NMIN=NMINL(I)
      CALL NCRGR1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,I)
   10 CONTINUE
      CALL NCRGR2(NRST,NRR,NLAYR,NSTFR)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,33)NLAYR,NRR
   33 FORMAT('1',20X,'DISCRETISATION DATA',//,
     1          10X,'NUMBER OF LAYERS  =',I5,//,
     1            10X,'NUMBER OF REGIONS    =',I5,//)
   94 CONTINUE
C
C     DICRETISE ENTIRE REGION INTO TRIANGULAR ELEMENTS.
C
      KCREP=1
      NSTOR=1
      NTSEL=1
      DO 65 I=NRST,NRR
      DO 691 L=1,NLAYR
      N1=NH1(L,I)
      N2=NH2(L,I)
      NFND1=NDB1(L,I)
```

```
      NFND2=NDB2(L,I)
      NPAT=NTRAN(L,I)
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
      NELEM=NTSEL-1
      DO 68 IE=NSTOR,NELEM
      IPROP(IE)=L
   68 CONTINUE
      NSTOR=NTSEL
  691 CONTINUE
   65 CONTINUE
C
C     PRINT OUT ELEMENT DATA.
C
      NELEM=NTSEL-1
      IF(IDISCR.EQ.0) GO TO 104
      WRITE(NPRINT,83)
   83 FORMAT(///,10X,46H IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS,/
     1//,12X,7HELEM NO,10X,5HNODE1,10X,5HNODE2,10X,5HNODE3,10X,17HREPETI
     2TION NUMBER,//)
      DO 80 I=1,NELEM
      WRITE(NPRINT,93)I,(NOD(I,K),K=1,3),NREP(I)
   93 FORMAT(10X,I5,3(10X,I5),12X,I9)
   80 CONTINUE
  104 CONTINUE
C
C     GNERATE NODE COORDINATES.
C
      RI=RW
      NS=1
      MM=1
      DO 75 I=NRST,NRR
      DXG=XLEN(I)
      ZI=0.0
      DO 87 L=1,NLAYR
      TH=THL(L)
      NN=NH1(L,I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH
      NS=NS-1
      RI=RI-DXG
   87 CONTINUE
      RI=RI+DXG
      NS=NS+1
   75 CONTINUE
      ZI=0.0
      DO 877 L=1,NLAYR
      TH=THL(L)
      NN=NH2(L,NRR)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH
      NS=NS-1
      RI=RI-DXG
  877 CONTINUE
C
C
      NNODE = NS
      IF(IDISCR.EQ.0) GO TO 114
      WRITE(NPRINT,103)
  103 FORMAT('1',10X,'NODAL COORDINATES',//,10X,'NODE',5X,
     1'R-COORD',5X,'Z-COORD',//)
C
      DO 85 I=1,NNODE
      WRITE(NPRINT,113)I,X(I,1),X(I,2)
  113 FORMAT(10X,I4,2F11.2)
   85 CONTINUE
  114 CONTINUE
C
C     GENERATE TOP BOUNDARY COORDINATE VECTORS.
C
      NDTO=NS
      CALL TOPVEC(THTO,LVEC,NDTO)
C
C     PRINT OUT BOUNDARY COORDINATE VECTORS.
      WRITE(NPRINT,223)
  223 FORMAT(///,20X,'TOP BOUNDARY NODES AND RADIAL COORDINATES',///,
     110X,'NODE NUMBER',20X,'R-COORDINATE',//)
      DO 335 I=1,LVEC
      WRITE(NPRINT,333)NDVEC(I),RVEC(I)
  335 CONTINUE
  333 FORMAT(10X,I7,25X,F10.2)
      RETURN
      END
      SUBROUTINE DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NRR,XLEN
C
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      MAXNR=89
      XLEN(1)=FRLEN
      RLEN=RO-RW
```

```
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      IF(XLEN(I).GT.XLMAX) XLEN(I)=XLMAX
      SUM=XLEN(I)+SUM
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I
      DENOM=1.+SCFAC+SCFAC**2
      XLEN(I-2)=(XREM+XLEN(I-1)+XLEN(I-2))/DENOM
      XLEN(I-1)=XLEN(I-2)*SCFAC
      XLEN(I)=XLEN(I-1)*SCFAC
      RETURN
      END
      SUBROUTINE NCRGR1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,L)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NTRAN,NH1,NH2,NDB1,NDB2
C
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      NC=NFR
      NCOUNT=0
      DO 10 I=NRST,NRR
      IRGG=IREG
      IF((I.EQ.NRST).AND.(IPENTR.NE.0)) IRGG=0
      NCOUNT=NCOUNT+1
      IF(NCOUNT.LT.IRGG) GO TO 20
      NCOUNT=0
      NHALF=NC/2
      NREM=NC-2*NHALF
      IF(NREM.GT.0) GO TO 15
      NC1=NC-1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=2
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=NC+1
      GO TO 10
   15 CONTINUE
      NC1=NHALF+1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=1
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=2*NH2(L,I)-1
      GO TO 10
   20 CONTINUE
      NH2(L,I)=NC
      NTRAN(L,I)=0
      NH1(L,I)=NH2(L,I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE NCRGR2(NRST,NRR,NLAYR,NSTFR)
C
C     MODIFIES THE VALUES OF NDB1 AND NDB2.
C
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      NDBC=NSTFR
      DO 50 J=NRST,NRR
      NDB1(1,J)=NDBC
      IF(NLAYR.LT.2) GO TO 56
      DO 55 L=2,NLAYR
      NDB1(L,J)=NDB1(L-1,J)+NH1(L-1,J)-1
   55 CONTINUE
   56 CONTINUE
      L=NLAYR
      NDBC=NDB1(L,J)+NH1(L,J)
   50 CONTINUE
      NR1=NRR-1
      DO 60 J=NRST,NR1
      DO 65 L=1,NLAYR
      NDB2(L,J)=NDB1(L,J+1)
   65 CONTINUE
   60 CONTINUE
      NDB2(1,NRR)=NDBC
      DO 75 L=2,NLAYR
      NDB2(L,NRR)=NDB2(L-1,NRR)+NH2(L-1,NRR)-1
   75 CONTINUE
      RETURN
      END
      SUBROUTINE COORDC(NS,MM,NN,DXG,TH,RI,ZI)
C
C     GENERATES NODAL COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /ASOLV/XX(5,90),XY(5,90)
```

```
      DY=TH/(NN-1)
      DO 20 L=1,NN
   20 XX(1,L)=RI
      DO 50 K=1,MM
   50 XY(K,1)=ZI
      DO 60 J=2,NN
      JM1=J-1
      DO 70 K=1,MM
      XY(K,J)=XY(K,JM1)+DY
   70 CONTINUE
   60 CONTINUE
      DO 80 J=1,MM
      DO 90 K=1,NN
      L=NN*(J-1)+K+NS-1
      X(L,1)=XX(J,K)
      X(L,2)=XY(J,K)
   90 CONTINUE
   80 CONTINUE
      NEN=NS+(MM*NN)-1
      NS=NEN+1
      RI=X(NEN,1)+DXG
      RETURN
      END
      SUBROUTINE BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC BLOCKS.
C
      DIMENSION ND(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND1+1
      ND(2,1)=ND(1,3)
      ND(2,2)=ND(1,2)
      ND(2,3)=ND(1,2)+1
      IF(NPAT.EQ.1) GO TO 200
      IF(NPAT.EQ.0) GO TO 265
      NTEMP=N1-2
      DO 90 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      DO 50 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      NTEMP=NTEMP-1
      NTSEL=NLST+1
   90 CONTINUE
      GO TO 400
  200 CONTINUE
      ND(3,1)=ND(2,1)
      ND(3,2)=ND(2,2)+1
      ND(3,3)=ND(3,1)+1
      NTEMP=(N1-1)/2-1
      L1=2
      L2=1
      DO 250 JJ=1,3
      L3=2
      IF(JJ.EQ.2) L3=1
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      DO 270 K=1,3
      IF(K.EQ.1) LL=L1
      IF(K.EQ.2) LL=L2
      IF(K.EQ.3) LL=L3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)*LL
  270 CONTINUE
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  250 CONTINUE
      GO TO 400
  265 CONTINUE
      NTEMP=N1-2
      DO 190 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 140 J1=NST,NLST
      NREP(J1)=KCREP
      DO 150 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
  150 CONTINUE
  140 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
```

```
  190 CONTINUE
  400 CONTINUE
      RETURN
      END
      SUBROUTINE AQPROP(NELEM,RGP,BTGP,TH)
C
C     COMPUTES ELEMENT PROPERTIES INDEX,IPROP.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      DO 10 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      ORX=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      IPROP(I)=1
      IF((ORX.LT.RGP).AND.(ORY.GT.BTGP)) IPROP(I)=0
   10 CONTINUE
      RETURN
      END
      SUBROUTINE TOPVEC(TH,LVEC,NDTO)
C
C     COMPUTES RADIAL COORDINATES FOR NODES SITUATED ON TOP BOUNDARY.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      J=0
      DO 10 I=1,NDTO
      ZDIF=ABS(TH-X(I,2))
      IF(ZDIF.GT.0.001) GO TO 10
      J=J+1
      NDVEC(J)=I
      RVEC(J)=X(I,1)
   10 CONTINUE
      LVEC=J
      RETURN
      END
      SUBROUTINE ELGND3(NT,SC,M)
C
C     GENERATES ELEMENT MATRIX,D, FOR TRIANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      IF(NREP(M).EQ.NT) GO TO 50
      NT=NREP(M)
      XJ=X(J2,1)-X(J1,1)
      XM=X(J3,1)-X(J1,1)
      YJ=X(J2,2)-X(J1,2)
      YM=X(J3,2)-X(J1,2)
      AREA(NT)=0.5*(XJ*YM-XM*YJ)
      AFUN=2.*AREA(NT)
      B(NT,1)=(YJ-YM)/AFUN
      B(NT,2)=   (YM)/AFUN
      B(NT,3)=  (-YJ)/AFUN
      C(NT,1)=(XM-XJ)/AFUN
      C(NT,2)=  (-XM)/AFUN
      C(NT,3)=   (XJ)/AFUN
      DO 100 I=1,3
      DO 100 J=1,3
      IF(J-I) 105,110,110
  110 ELK(I,J,NT)=B(NT,I)*B(NT,J)+C(NT,I)*C(NT,J)
      GO TO 100
  105 ELK(I,J,NT)=ELK(J,I,NT)
  100 CONTINUE
C
C     FORM ELEMENT MATRIX:- D
C
      D(1,1)=0.5*ORX(M)*AREA(NT)*SC/3.
      D(2,2)=D(1,1)
      D(3,3)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      D(1,3)=D(1,2)
      D(3,1)=D(1,3)
      D(2,3)=D(1,2)
      D(3,2)=D(2,3)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE ELGNCT(M,III,AK,BK,NT,VCOUNT,PMK)
C
C     GENERATES ELEMENT MATRIX,EK,FOR TRIANGULAR ELEMENTS.
C
```

```
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      IF(IDARCY(M).EQ.0) GO TO 70
  500 CONTINUE
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      CALL CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
      GO TO 80
   70 CONTINUE
      PX=PMK
      ACONST=PX*ORX(M)*AREA(NT)
   80 CONTINUE
C
C                 CALCULATE ELEMENT MATRIX
C
      DO 200 I=1,3
      DO 200 J=1,3
      EK(I,J)=ACONST*ELK(I,J,NT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
C
C     COMPUTES NON-LINEAR COEFFICIENT FOR TRIANGULAR ELEMENTS.
C
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      ACONST=AREA(NT)*ORX(M)/TEMP
  500 CONTINUE
      RETURN
      END
      SUBROUTINE EBFIN3(LEN,NM,NN,LL)
C
C     SUBROUTINE TO GENERATE THE BANDWIDTHS OF THE BANDED SYM STIFFNESS
C     MATRIX. NBAND CONTAINS THE BANDWIDTHS, ID THE POSITION OF THE
C     TERM ON THE DIAGONALS OF THE ORIGINAL MATRIX, LEN IS THE LENGTH
C     THE VECTOR.
C
      DIMENSION LV(3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      DO 20 I=1,LL
      NBAND(I)=1
      ISTART(I)=I
   20 CONTINUE
C
C     SCAN THROUGH THE LOCATION VECTOR FOR EACH MEMBER TO FIND
C     THE POSITION OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW
C
      NF2=NN
      DO 25 I=1,NM
      DO 30 J=1,NF2
      LV(J)=NOD(I,J)
   30 CONTINUE
      DO 45 J=1,NF2
      IF(LV(J).EQ.0)GO TO 45
      DO 40 K=1,NF2
      IF(LV(J).GT.LV(K)) GO TO 40
      NW=LV(K)-LV(J)+1
      NR=LV(J)
      IF(NW.GT.NBAND(NR))  NBAND(NR)=NW
   40 CONTINUE
   45 CONTINUE
C      SEARCH FOR THE FURTHEREST OFF-LEFT TERM
C
      DO 55 J=1,NF2
      IF(LV(J).EQ.0) GO TO 55
      DO 65 K=1,NF2
      IF(LV(J).LT.LV(K)) GO TO 65
      NW=LV(K)
      NR=LV(J)
      IF(NW.LT.ISTART(NR)) ISTART(NR)=NW
   65 CONTINUE
   55 CONTINUE
   25 CONTINUE
C
C     SET UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN=NBAND(1)
      DO 50 I=2,LL
      IF(NBAND(I)-NBAND(I-1).LT.0)  NBAND(I)=NBAND(I-1)-1
```

```
      LEN=LEN+NBAND(I)
   50 CONTINUE
      ID(1)=1
      DO 60 I=2,LL
   60 ID(I)=ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE MERBD(N,M)
C
C     MERGES ELEMENT MATRIX D INTO GROSS VECTOR VD.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VD(IPOS)=VD(IPOS)+D(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MERB3(N,M)
C
C     MERGES ELEMENT MATRIX EK INTO GROSS VECTOR VCORE.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ACORE/VCORE(2000)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS= ID(IK)+JK-IK
      VCORE(IPOS)=VCORE(IPOS)+EK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SYMSOL(LL,NLL,NBM,NBX)
C
C     AN IN CORE BAND SOLVER.
C     USE IN CONJUNCTION WITH TOPOLOGICAL SUBROUTINE FOR LARGE BANDWIDTH.
C
      DIMENSION VTEMP(90)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),N(300),IDUM(300)
      COMMON /BSOLV/C(300,1),V(2000)
      DOUBLE PRECISION TEMP
      JBOUN=NBM+1
      NBP=JBOUN
      ID=1
      IF(NBM.EQ.0) GO TO 150
      DO 100 I=1,NBM
  100 ID=ID+N(I)
  150 CONTINUE
      DO 10 I=NBP,LL
      IF(JBOUN.GT.NBX) JBOUN=NBX
C
      TEMP=V(ID)
      NEB=ID+N(I)-1
      ID1=ID+1
      IF(I.EQ.JBD(JBOUN)) GO TO 16
C
C     NORMALISE ROW I
C
      KK=0
      V(ID)=1.0
      IF(ID1.GT.NEB) GO TO 29
      DO 20 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
   20 V(J)=V(J)/TEMP
   29 CONTINUE
      DO 25 L=1,NLL
      C(I,L)=C(I,L)/TEMP
   25 CONTINUE
      GO TO 46
   16 CONTINUE
      IF(ID1.GT.NEB) GO TO 39
      KK=0
      DO 120 J=ID1,NEB
```

```
      KK=KK+1
      VTEMP(KK)=V(J)
  120 V(J)=0.0
   39 CONTINUE
      DO 125 L=1,NLL
  125 C(I,L)=DISP(JBOUN)
      JBOUN=JBOUN+1
C
C     ELIMINATION
C
   46 CONTINUE
      IDJ=ID
      IF(ID1.GT.NEB)  GO TO 35
      KK=0
      DO 30 J=ID1,NEB
      JI=J-ID1+I
      IDJ=IDJ+N(JI)
      KK=KK+1
      IF(VTEMP(KK)) 50,30,50
   50 CONTINUE
      IF(JBOUN.EQ.1) GO TO 240
      JBM1=JBOUN-1
      IF(I.EQ.JBD(JBM1)) GO TO 140
  240 IDP=J
      DO 40 K=IDP,NEB
      KJ=IDJ+K-J
   40 V(KJ)=V(KJ)-V(K)*VTEMP(KK)
  140 CONTINUE
      NJ=I+J-ID
      DO 32 L=1,NLL
      C(NJ,L)=C(NJ,L)-C(I,L)*VTEMP(KK)
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID=ID+N(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID=ID-1
      LLM1=LL-1
      LL1=LL-NBP
      DO 70 IB=1,LL1
      I=LLM1-IB+1
      ID=ID-N(I)
      IS=I+1
      IN=I+N(I)-1
      DO 80 J=IS,IN
      NJ=ID+J-I
      DO 75 L=1,NLL
      C(I,L)=C(I,L)-C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,RW,NSCREN)
C
C     LOCATES NODES WHERE HEAD VALUES ARE PRESCRIBED.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      K=0
      NFND=1
      DO 30 I=1,NSCREN
      XST=XSCR(I)-0.01
      XEND=XST+HSCR(I)+0.02
      DO 40 J=1,NNODE
      L=NFND+J-1
      IF(L.EQ.1) GO TO 222
      IF(X(L,1).GT.RW) GO TO 30
  222 CONTINUE
      IF((X(L,2).LT.XST).OR.(X(L,2).GT.XEND)) GO TO 70
   22 K=K+1
      JBD(K)=L
      DISP(K)=HW
      GO TO 40
   70 IF(X(L,2).GT.XEND) GO TO 30
   40 CONTINUE
      NFND=L+1
   30 CONTINUE
      J=K
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
```

```
      NBD=J
      RETURN
      END
      SUBROUTINE VCHEC3(M,AK,BK,J1,J2,J3,NT,PMK,VCR,GRRX,GRRY)
C
C     COMPUTES ELEMENT VELOCITIES FOR TRIANGULAR ELEMENTS AND CHECK IF
C     A PARTICULAR ELEMENT BELONGS TO DARCY OR NON-DARCY FLOW ZONE.
C
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      GRRX=HX/HS
      GRRY=HY/HS
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SETARG(AGP,BGP,PMGP,VGP,AKK,BKK,PKK,VCR,I)
C
C     IDENTIFIES THE ELEMENT PROPERTIES.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      IF(IPROP(I).EQ.0) GO TO 20
      GO TO 10
   20 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
      VCR=VGP
   10 CONTINUE
      RETURN
      END
      SUBROUTINE VDFB(LVEC)
C
C     GENERATES VECTOR VDTOP.
C
      DIMENSION D(2,2)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      NELTOP=LVEC-1
      NN=2
      ID(1)=1
      DO 10 I=2,LVEC
   10 ID(I)=ID(I-1)+2
      LENT=2*LVEC-1
      DO 108 I=1,LENT
      VDTOP(I)=0.0
  108 CONTINUE
      DO 208 IE=1,NELTOP
      IN=NDVEC(IE)
      IP=NDVEC(IE+1)
      RAVE=0.5*(X(IP,1)+X(IN,1))
      RDIF=X(IP,1)-X(IN,1)
      D(1,1)=RAVE*RDIF/3.0
      D(2,2)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      DO 308 I=1,2
      IK=IE+(I-1)
      DO 408 J=1,2
      JK=IE+(J-1)
      IF(IK.GT.JK) GO TO 408
      IPOS=ID(IK)+JK-IK
      VDTOP(IPOS)=VDTOP(IPOS)+D(I,J)
  408 CONTINUE
  308 CONTINUE
  208 CONTINUE
      RETURN
      END
      SUBROUTINE GVMOD(LACT,SY,IGK)
C
C     MODIFIES GROSS VECTOR VK AND GK TO TAKE ACCOUNT OF LEAKAGE FLUX ACROSS TOP
C     BOUNDARY.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
```

```
C
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      J2=J1
      JM=J1
      IF(L.LT.LACT) J2=NDVEC(L+1)
      IF(L.GT.1) JM=NDVEC(L-1)
      SUM=VF1*HINT(J1,1)+VF2*HINT(J2,1)+VFM*HINT(JM,1)
      IF(IGK.GT.0) GK(J1,1)=GK(J1,1)+SUM
      IF(IGK.GT.0) GO TO 1078
      IS=ID(I)
      VK(IS)=VK(IS)+VF1
      IL=IS+NBAND(I)-1
      VK(IL)=VK(IL)+VF2
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE GKMOD(LACT,SY)
C
C     MODIFIES GK TO TAKE ACCOUNT OF LEAKAGE FLUX ACROSS TOP BOUNDARY.
C
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      LM=L
      LP=L
      IF(L.LT.LACT) LP=L+1
      IF(L.GT.1) LM=L-1
      SUM=VF1*QLEAK(L)+VF2*QLEAK(LP)+VFM*QLEAK(LM)
      GK(J1,1)=GK(J1,1)+SUM
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE ROEST(DIFFUS,TM,LVEC,QFIX,TMIS,RO,NELEM,NNODE,NELTO,
     1   LST,LSTEL,LVST)
C
C     ESTIMATES RADIUS OF INFLUENCE AT TIME TM.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      FPI=88./7.
      CONST=QFIX/(FPI*TMIS)
      DO 10 I=LVST,LVEC
      U=RVEC(I)**2/(4.*DIFFUS*TM)
      WW=W(U)
      SDRAW=CONST*WW
      IF(SDRAW.GT.0.001) GO TO 10
      NNODE=NDVEC(I)
      RO=RVEC(I)
      GO TO 20
   10 CONTINUE
      NNODE=NDVEC(LVEC)
      RO=RVEC(LVEC)
   20 CONTINUE
      LST=NNODE
      LVST=I
      DO 30 I=LSTEL,NELTO
      IF(ORX(I).LT.RO) GO TO 30
      NELEM=I-1
      GO TO 40
   30 CONTINUE
      NELEM=NELTO
   40 CONTINUE
      LSTEL=NELEM
      RETURN
      END
      SUBROUTINE TIGEN(NTICR,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX,TPAT)
C
C     GENERATES DISCRETE TIME VECTOR,TIME.
C
```

```
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      SWST=0.8
      CUN=22.*RW**2/(7.*QFIX)
      TIME(1)=TFACTR
      IF(TPAT.EQ.2) GO TO 15
      DO 10 I=2,NTICR
      TIME(I)=TIME(I-1)*TMUL
      TMUL=TMUL+DTMUL
   10 CONTINUE
      GO TO 17
   15 CONTINUE
      DO 20 I=2,NTICR,2
      TIME(I)=TIME(I-1)
      TIME(I+1)=2.*TIME(I)
   20 CONTINUE
      DO 25 I=2,NTICR
      TIME(I)=TIME(I)+TIME(I-1)
   25 CONTINUE
   17 CONTINUE
      RR=RW+1.99
      CONST=RR**2*SS/(4.0*PM)
      IF(IWBC.NE.0)CONST= 1000.*CUN*SWST
      DO 30 I=1,NTICR
      TIME(I)=TIME(I)*CONST
   30 CONTINUE
      RETURN
      END
      FUNCTION W(U)
C
C     EVALUATES THEIS WELL FUNCTION.
C
      WC=-0.5772-ALOG(U)
      W=WC
      TERM=1
      J=1
      DO 10 I=1,30
      TERM=(TERM*U)/I
      IF(J.EQ.0) GO TO 20
      W=W+TERM/I
      J=0
      GO TO 30
   20 CONTINUE
      W=W-TERM/I
      J=1
   30 CONTINUE
      EPSI=ABS(W-WC)
      IF(EPSI.LE.0.01) GO TO 40
      WC=W
   10 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE VECMUL(NNODE,LEN)
C
C     PERFORMS MATRIX MULTIPLICATION.
C        VKD  =  VK*D
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VK(2000),D(300,1),IPROP(500),DUM(3,3)
      COMMON /ADPARA/TLESS(300),HLESS(300),VKD(300,1),VDTOP(200)
      L=1
      DO 10 I=1,NNODE
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
C
      VKD(I,1) = 0.0
      DO 20 J = IS, IL
      K=I+J-IS
      VKD(I,1)=VKD(I,1)+VK(J)*D(K,L)
   20 CONTINUE
C
      IF(I.EQ.1) GO TO 40
      I1 = I-1
      IST= ISTART(I)
C
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =    ID(J) + K
      VKD(I,1)=VKD(I,1)+VK(IP)*D(J,L)
   30 CONTINUE
   40 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE AQDIS(NLEN,QAQFR,QRDIF,DELT,TH,RCSNG,IQ,QFIX,DWD)
C
C     COMPUTES TOTAL DISCHARGE INTO THE WELL.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
```

```
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BQCAL/VS(600),QWB(40)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      QSUM=0.0
      TWPI=44./7.
      CALL QMULT(NLEN,QSUM)
      QSUM=ABS(TWPI*QSUM)/(0.5*DELT)
      QAQFR=QSUM
      ACSNG=0.5*TWPI*RCSNG**2
      TRM=ABS(ACSNG*DWD*2./DELT)
      QCALX=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALX)
      QCALC(IQ)=QCALX
      RETURN
      END
      SUBROUTINE QMULT(NBW,QSUM)
C
C     COMPUTES NODAL FLUXES AT WELL BOUNDARY.
C
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DO 10 L=1,NBW
      I=JBD(L)
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
      QWB(L)=0.0
      DO 20 J = IS, IL
      K=I+J-IS
      QWB(L)=QWB(L)+VKQ(J)*H(K)
   20 CONTINUE
      IF(I.EQ.1) GO TO 15
      I1 = I-1
      IST= ISTART(I)
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
      QWB(L)=QWB(L)+VKQ(IP)*H(J)
   30 CONTINUE
   15 CONTINUE
      QWB(L)=QWB(L)-GP(L)
      QSUM=QSUM+QWB(L)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE FQFLUX(NBM,NNODE)
C
C     MODIFIES MATRIX CK BY ADDING THE FLUX TERMS AT WELL BOUNDARY.
C
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
C
C     MODIFY MATRIX CK BY ADDING THE FLUX AT THE WELL BOUNDARY
C
      IF(NBM.EQ.0) GO TO 20
      NBP=NBM+1
      DO 10 I=NBP,NNODE
      IST=ISTART(I)
      IF(IST.GT.NBM) GO TO 10
      I1=I-1
      DO 30 J=IST,NBM
      K=I1-J+1
      IP=ID(J)+K
      CK(I,1)=CK(I,1)-VK(IP)*CK(J,1)
   30 CONTINUE
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SWMOD(IQ,IT,RCSNG,DLTA,QFIX,QAQFR,SWOLD,DSOLD,TPAT,FQ)
C
C     ADJUSTS THE VALUE OF WELL DRAWDOWN.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      DELT=DLTA/10.**3
      ACSNG=22.0*RCSNG**2/7.0
      IF(IQ.NE.1) GO TO 15
      IF(ACSNG.GT.0.0) GO TO 35
      IF(IT.GT.5) GO TO 10
      DSW(IQ)=-DSOLD
      GO TO 20
   35 CONTINUE
      IF(IT.GT.5) GO TO 10
      QRATIO=QAQFR/QFIX
      IF(ABS(QRATIO).GE.0.8) GO TO 10
      TMM=TIME(IT)-0.5*DLTA
      FACTR=0.5*DELT/ACSNG
```

```
      QDEL=QFIX-QAQFR*0.95*TIME(IT)/TMM
      IF(ABS(QRATIO).GE.0.5) QDEL=QFIX-QAQFR
      DSW(IQ)=FACTR*QDEL
      DSW(IQ)=-DSW(IQ)
      GO TO 20
   10 CONTINUE
      TOLD=TIME(IT-2)
      ARGL1=(TIME(IT)-0.5*DLTA)/TOLD
      ARGL2=TIME(IT-1)/TOLD
      TLOG=ALOG(ARGL1)/ALOG(ARGL2)
      DSW(IQ)=(SW(IT-1)-SW(IT-2))*(1.-TLOG)
      DSW(IQ)=-DSW(IQ)
      IF(DSW(IQ).GT.0.) DSW(IQ)=0.0
      GO TO 20
   15 CONTINUE
      IF(IQ.GT.2) GO TO 25
      IF(TPAT.EQ.2) GO TO 62
      DSW(IQ)=DSW(IQ-1)*QFIX/QCALC(IQ-1)
      IF(ACSNG.LE.0.) DSW(IQ)=SW(IT)*QFIX/QCALC(IQ-1)-SWOLD
      GO TO 20
   62 CONTINUE
      IF((ACSNG.LE.0.).AND.(FQ.LT.0.)) GO TO 40
      DSW(IQ)=DSW(1)*QFIX/QCALC(1)
      GO TO 20
   40 CONTINUE
      SWCOR=SW(IT)-FQ*(QCALC(1)-QFIX)/QFIX
      DSW(IQ)=SWCOR-SWOLD
      GO TO 20
   25 CONTINUE
      DDSW=DSW(IQ-1)-DSW(IQ-2)
      TERM1=QFIX-QCALC(IQ-1)
      TERM2=QCALC(IQ-1)-QCALC(IQ-2)
      DQR=TERM1/TERM2
      DSW(IQ)=DSW(IQ-1)+DDSW*DQR
   20 CONTINUE
      SW(IT)=DSW(IQ)+SWOLD
      DSOLD=-DSW(IQ)
      RETURN
      END
      SUBROUTINE RCUT(NNODE)
C
C     PRINTS OUT NODAL VALUES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /ADPARA/TLESS(300),HLESS(300),VKD(300,1),VDTOP(200)
      DATA NPRINT/3/
      WRITE(NPRINT,53)
   53 FORMAT(///)
      WRITE(NPRINT,3)
      WRITE(NPRINT,13)
      WRITE(NPRINT,23)
      WRITE(NPRINT,13)
      WRITE(NPRINT,3)
    3 FORMAT(5X,50('*'))
   13 FORMAT(5X,'*',48X,'*')
   23 FORMAT(5X,'*',11X,'FINAL RESULTS OF ANALYSIS',12X,'*')
      WRITE(NPRINT,33)
      DO 10 I=1,NNODE
      WRITE(NPRINT,43)I,X(I,1),H(I),TLESS(I),HLESS(I)
   10 CONTINUE
   33 FORMAT(///,20X,'HEAD .VS. RADIUS AND 1/U .VS. W(U)',///,
     1          10X,'NODE',10X,'R-COORD',10X,'HEAD',14X,'1/U',19X,
     1'W(U)',//)
   43 FORMAT(10X,I3,6X,F10.2,8X,F10.4,7X,E11.4,10X,F10.4)
      RETURN
      END
      SUBROUTINE HPUNCH(NNODE,L,INDEX)
C
C     PUNCHES OUT NODAL VALUES AT FINAL TIME.
C
      COMMON /WORKA/VWORK(500)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DO 15 I=1,NNODE
      IF(INDEX.EQ.1) VWORK(I)=H(I)
      IF(INDEX.EQ.2) VWORK(I)=QLEAK(I)
   15 CONTINUE
      NCARD=NNODE/6
      NCARDT=NCARD+2
      WRITE(3,23)NCARDT
   23 FORMAT(///,10X,'TOTAL NUMBER OF PUNCHED CARDS   =',I10)
      WRITE(L,3)NNODE
    3 FORMAT(I10)
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      WRITE(L,13)(VWORK(I),I=IST,IEND)
   13 FORMAT(6E13.4)
      IST=IEND+1
```

```
   10 CONTINUE
      NREM=NNODE-NCARD*6
      IEND=IST+NREM-1
      WRITE(L,13)(VWORK(I),I=IST,IEND)
      RETURN
      END
      SUBROUTINE BSIMP(NNODE,TH,DINDEX,SY,DELT,IT,ITMIN)
C
C     EVALUATES BOULTON'S CONVOLUTIONAL INTEGRAL BY SIMPSON'S 1/3 RULE.
C
      DIMENSION FN(3)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /VCOM/ORX(500),VEL(500),HDUM(500),IDARCY(500)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),H(50),HINT(50,1),GP(40)
C
      DOUBLE PRECISION DEXP,ARGU,FN,QCUNT
C
      WRITE(3,3)
    3 FORMAT(///,20X,'LEAKAGE NODAL FLUX',///,10X,'NODE NO.',
     1            20X,'FLUX VALUE',//)
      ITM=IT-1
      IF(ITM.EQ.0) GO TO 5
      TM=TIME(ITM)
      TOLD=(TIME(IT)+TIME(ITM))*0.5
      GO TO 15
    5 TM=0.0
      TOLD=0.5*TIME(IT)
   15 TMM=(TIME(IT)+TIME(IT+1))*0.5
C
      DO 20 II=1,3
      TC=TM+DELT*(II-1)*0.5
      ARGU=-(TMM-TC)*DINDEX
      FN(II)=DEXP(ARGU)
   20 CONTINUE
      ARGU=-DINDEX*(TMM-TOLD)
      DO 30 I=1,NNODE
      IF(IT.GT.ITMIN) GO TO 40
      T1=2.*FN(1)*(HINT(I,1)-HCOLD(I))
      T2=4.*FN(2)*(H(I)-HCOLD(I))
      T3=2.*FN(3)*(H(I)-HINT(I,1))
      DQ   =T1+T2+T3
      GO TO 50
   40 CONTINUE
      DQ   =(H(I)-HINT(I,1))*(FN(1)+4.*FN(2)+FN(3))
   50 CONTINUE
      DQ=-DQ
      QCUNT=QLEAK(I)*DEXP(ARGU)
      QLEAK(I)=QCUNT+DQ
      IF(DINDEX.LE.0.) QLEAK(I)=DQ
      WRITE(3,13)I,QLEAK(I)
   13 FORMAT(10X,I5,20X,E13.3)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE FQSET(IT,ITCUR,DELT,FQ,TPAT)
C
C     SETS DRAWDOWN FUNCTION FOR DISCHARGE ITERATION.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      IF(TPAT.EQ.1) GO TO 20
      IF(IT.GE.3) GO TO 10
      IF((IT.EQ.1).AND.(ITCUR.GT.IT)) FQ=SW(IT)
      GO TO 20
   10 CONTINUE
      DO 15 I=1,IT
      HDLTA=0.499*DELT
      IF(HDLTA.GT.TIME(I)) GO TO 15
      FQ=SW(I)
      GO TO 20
   15 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE HREAD(NNODE,L,INDEX)
C
C     READS IN NODAL VALUES AT STARTING TIME.
C
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /WORKA/VWORK(500)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      READ(L,3)NNODE
    3 FORMAT(I10)
      NCARD=NNODE/6
      NCARDT=NCARD+1
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      READ(L,13)(VWORK(I),I=IST,IEND)
   13 FORMAT(6E13.4)
      IST=IEND+1
```

```
   10 CONTINUE
      NREM=NNODE-NCARD*6
      IEND=IST+NREM-1
      READ(L,13)(VWORK(I),I=IST,IEND)
      DO 15 I=1,NNODE
      IF(INDEX.EQ.1) H(I)=VWORK(I)
      IF(INDEX.EQ.2) QLEAK(I)=VWORK(I)
   15 CONTINUE
      RETURN
      END
      FUNCTION HPW(U,RM,SCLEN,TH)
C
C     COMPUTES HANTUSH WELL FUNCTION FOR PARTIALLY PENETRATING WELLS.
C
      DOUBLE PRECISION U,HWF,DSUM,SUM
      DIF=TH-SCLEN
      PI=22./7.
      PISQ=PI**2
      FACTR=2.*TH*TH/(PISQ*SCLEN**2)
      SUM=0.0
      DO 30 I=1,50
      APSI=3.14*I*RM
      ARGU=I*3.14*DIF/TH
      TERM=SIN(ARGU)**2/I**2
      DSUM=TERM*HWF(U,APSI)
      SUM=SUM+DSUM
      IF((I.GT.15).AND.(SUM.LT.1.D-10)) GO TO 40
      IF(SUM.LT.1.D-20) GO TO 30
      SR=DSUM/SUM
      STOL=ABS(SR)
      IF(STOL.LT.0.0001) GO TO 40
   30 CONTINUE
   40 CONTINUE
      UX=U
      SUM=SUM*FACTR+W(UX)
      HPW=SUM
      RETURN
      END
      FUNCTION HJW(U,RB,Y)
C
C     EVALUATES INTEGRAND EXPRESSION OF HANTUSH WELL FUNCTION.
C
      DOUBLE PRECISION U,F,DEXP,HJW,TERM
      TERM=-Y-RB**2/(4.*Y)
      F=DEXP(TERM)/Y
      HJW=F
      RETURN
      END
      FUNCTION HWF(U,APSI)
C
C     EVALUATES HANTUSH FUNCTION BY SIMSON'S 3-POINT FORMULA.
C
      DOUBLE PRECISION HDW,U,HWF,DSUM,SUM,DW1,DW2,DW3,DABS,DY,HJW
      TMUL=1.4
      Y1=U
      SUM=0.0
      DO 30 I=1,50
      Y3=Y1*TMUL
      Y2=(Y1+Y3)/2.
      DY=Y3-Y1
      DW1=HJW(U,APSI,Y1)
      DW2=HJW(U,APSI,Y2)
      DW3=HJW(U,APSI,Y3)
      DSUM=DY*(DW1+4.*DW2+DW3)/6.
      SUM=SUM+DSUM
      Y1=Y3
      IF((I.GT.15).AND.(SUM.LT.1.D-10)) GO TO 40
      IF(SUM.LT.1.D-20) GO TO 30
      SR=DSUM/SUM
      STOL=ABS(SR)
      IF(STOL.LT.0.0001) GO TO 40
   30 CONTINUE
   40 CONTINUE
      HWF=SUM
      RETURN
      END
      SUBROUTINE DSWINT(DSOLD,BK,QFIX,TMIS,TH,RW,SCLEN,DIFFUS,CUNST)
C
C     ESTIMATES WELL DRAWDOWN FOR FIRST TIME STEP.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(60),QCALC(5)
      DOUBLE PRECISION UW
      DLAMDA=BK*QFIX*TMIS/(TH*TH*RW)
      DLAMDA=7.*DLAMDA/44.
      RM=RW/TH
      GAMMA=SCLEN/TH+0.01
      CONS=4.0*DIFFUS*0.5*TIME(1)
      UW=CONS/RW**2
      V=UW
      UW=1./UW
```

```
U=UW
TERM=0.25*DLAMDA*(ALOG(V)-3.0)
WW=W(U)
IF(GAMMA.LT.1.) WW=HPW(UW,RM,SCLEN,TH)
DSOLD=WW+1.5*DLAMDA+TERM
IF(U.GT.0.1E-01) DSOLD=WW
DSOLD=DSOLD*CUNST
RETURN
END
```

4.5 LISTING OF STCOND

```
C
C       INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C       DEVELOPED BY P.S. HUYAKORN.
C
C       STCOND,PROGRAM FOR SOLVING STEADY,TWO-DIMENSIONAL,DARCY OR TWO-REGIME FLOW
C
C       USING RECTANGULAR ELEMENTS OR A COMBINATION OF RECTANGULAR AND TRIANGULAR
C
C       ELEMENTS.
C
C       VERSION DATED OCTOBER,1973.
C
C        FOR FURTHER INFORMATION,CONTACT
C
C          P.S. HUYAKORN OR C.R. DUDGEON
C
C          WATER RESEARCH LABORATORY
C
C          KING ST.,MANLY VALE
C
C          SYDNEY,N.S.W. 2093,AUSTRALIA.
C
C-----------------------------
C       LIST OF INPUT VARIABLES
C-----------------------------
C
C----------------ALL READ STATEMENTS ARE LOCATED IN THE MAIN------------------
C
C----------------PROGRAM AND INDICATED BY  $1 OR $1,SK  SIGN------------------
C
C***    PROBLEM VARIABLES  ***
C
C       NPROB    =  NUMBER OF PROBLEMS TO BE SOLVED
C       IVEL     =  VELOCITY PRINT-OUT INDEX
C                   FEED IN IVEL=0 IF VELOCITY PRINT-OUT IS NOT REQUIRED OTHERWISE
C                   FEED IN IVEL=1
C       IDISCR   =  DISCRETISATION DATA PRINT-OUT INDEX
C                   FEED IN IDISCR=0 IF DISCRETISATION PRINT-OUT IS NOT REQUIRED
C                   OTHERWISE FEED IN IDISCR=1
C       ORELAX   =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                   SUGGESTED VALUE LIES BETWEEN 1.5 AND 1.85
C       RW       =  RADIUS OF WELL SCREEN.
C       RO       =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C       HO       =  HEIGHT OR DRAWDOWN OF WATER TABLE AT THE EXTERNAL RADIUS
C       HTOL     =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                   SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C       NLAYR    =  NUMBER OF LAYERS OF WATER BEARING FORMATIONS
C                   FEED IN NLAYR=1 WHEN DEALING WITH AQUIFER-AQUITARD SYSTEM WITH
C                   TOP AQUITARD UNCONFINED(HANTUSH-JACOB SYSTEM)
C       IGP      =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                   IGP=0 FOR NON-GRAVEL PACKED WELL
C       IKMAX    =  LAYER OF MAXIMUM PERMEABILITY OR MINIMUM VALUE OF AKL(I)
C       IAQTA    =  AQUITARD INDEX. FEED IN IAQTA = 1 WHEN DEALING WITH AN
C                   AQUIFER-AQUITARD SYSTEM WITH OVERLYING AQUITARD UNCONFINED
C                   OTHERWISE FEED IN IAQTA = 0
C       THL(I)   =  THICKNESS OF LAYER NUMBER I
C       AKL(I)   =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER OR AQUITARD
C                   LAYER I
C                   FOR AQUITARD I,FEED IN AKL(I)=1./PML(I) WHERE PML(I) IS ITS
C                   HYDRAULIC CONDUCTIVITY
C       BKL(I)   =  FORCHHEIMER NON-LINEAR COEFFICIENT OF AQUIFER LAYER I
C                   FOR AQUITARD I,FEED IN BKL(I)=0.0
C       VCRL(I)  =  CRITICAL VELOCITY OF AQUIFER LAYER I
C                   FOR AQUITARD I,FEED IN VCRL(I)=0.0
C       AGP      =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C       BGP      =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C       VGP      =  CRITICAL FLOW VELOCITY OF GRAVEL PACK MATERIAL
C       THGP     =  THICKNESS OF GRAVEL PACK
C       BTGP     =  HEIGHT OF THE BASE OF GRAVEL PACK
C       PA       =  COEFFICIENT OF PERMEABILITY OF OVERLYING AQUITARD
C       THA      =  THICKNESS OF OVERLYING AQUITARD
C       HW       =  HEIGHT OF WATER LEVEL IN THE WELL OR WELL DRAWDOWN
C
C***    DISCRETISATION PARAMETERS  ***
C
C       FRLEN    =  LENGTH OF FIRST SUBREGION
C                   SUGGESTED VALUE IS FRLEN=RW
C                   FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
C       SCFAC    =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                   SUBREGIONS. SUGGESTED VALUE IS SCFAC=1.50
C       XLMAX    =  MAXIMUM BLOCK LENGTH,PRESCRIBED TO AVOID ILL-CONDITIONED
C                   ELEMENTS
C       NDSC     =  TOTAL NUMBER OF NODES ON WELL SCREEN(S)
C                   NDSC IS TO BE GREATER THAN OR EQUAL TO 2
C       NSCREN   =  NUMBER OF SCREENED INTERVALS
C       IREGL(I)=   NUMBER OF REPEATED REGULAR BLOCKS WITH THE SAME NUMBER OF NODES
C         -1        ON THE LEFT AND RIGHT VERTICAL LINES ACROSS LAYER I
C                   SUGGESTED VALUE IS IREG=2
C       NMINL(I)=   MINIMUM NUMBER OF NODES ALONG A VERTICAL LINE ACROSS LAYER I.
```

```
C                   TO MINIMISE THE TOTAL NUMBER OF NODES,SUGGEST NMIN=2
C      NFRL(I) =   NUMBER OF NODES ON PORTION OF WELL BOUNDARY PENETRATING AQUIFER
C                   OR AQUITARD LAYER I
C
C      XSCR(I) =   Z-COORDINATE OF BASE OF SCREEN I ABOVE DATUM
C      HSCR(I) =   LENGTH OF SCREENED INTERVAL I
C---------------------------------
C      LIST OF OUTPUT VARIABLES
C---------------------------------
C
C      NNODE    =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C      NELEM    =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C      X        =  RADIAL AND VERTICAL NODAL COORDINATES
C      H        =  NODAL HEAD OR DRAWDOWN VALUES
C      TLESS    =  NODAL VALUES OF DIMENSIONLESS RADIUS,1/U
C      HLESS    =  NODAL VALUES OF WELL FUNCTION FOR STEDY FLOW,W(U)
C      QWB      =  NODAL FLUXES AT WELL BOUNDARY
C      QSUM     =  TOTAL WELL DISCHARGE
C      NOD      =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C      VEL      =  ABSOLUTE ELEMENT VELOCITIES
C      VCOMP1   =  RADIAL COMPONENT OF ELEMENT VELOCITY
C      VCOMP2   =  VERTICAL COMPONENT OF ELEMENT VELOCITY
C      IDARCY   =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                  OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                  IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
C
C
C
C
C
C
C
C
C
C
C
C
C
C
      COMMON /WSCREN/XSCR(5),HSCR(5)
      COMMON /ALAYR/AKL(3),BKL(3),VCRL(3),SSL(3),THL(3),IREGL(3),
     1NMINL(3),NFRL(3),PML(3)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /BELEM/E(6,6,3),F(6,6,3),DSHFZ(2,6)
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /BQCAL/VS(600),QWB(40)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DOUBLE PRECISION DEXP,ARGU
      DATA NREAD,NPRINT/1,3/
C
C      PRINT INITIAL HEADINGS
C
      WRITE(NPRINT,1773)
 1773 FORMAT('1')
      WRITE(NPRINT,1003)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1023)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1033)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1003)
 1003 FORMAT(5X,51H ***************************************************)
 1013 FORMAT(5X,51H *                                                 *)
 1023 FORMAT(5X,51H *    FINITE ELEMENT ANALYSIS OF WELL PROBLEMS     *)
 1033 FORMAT(6X,'*    STEADY,TWO REGIME FLOW TOWARD A SINGLE WELL *')
      READ(NREAD,1011)NPROB,IVEL,IDISCR,ORELAX                              $1
 1011 FORMAT(3I10,F10.2)
      DO 4800 JPRO=1,NPROB
      WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1          I6,12X,'*',/,20X,50('*'))
C
C      READ AND PRINT GENERAL DATA.
C
      READ(NREAD,2001)RW,RO,HO,HTOL,NLAYR                                   $1
 2001 FORMAT(4F10.2,I10)
      READ(NREAD,111)IGP,IKMAX,IAQTA                                        $1
  111 FORMAT(3I10)
      IF(IAQTA.GT.0) NLAYR=1
      WRITE(NPRINT,23)RW,RO,HO,ORELAX,HTOL
      WRITE(NPRINT,193)IGP,IAQTA,IKMAX
   23 FORMAT(///,20X,'GENERAL INPUT DATA',//,
     1           10X,'RADIUS OF WELL  =',F10.2,//,
```

```
     2            10X,'RADIUS OF INFLUENCE  =',F10.2,//,
     3            10X,'HEIGHT OF WATER TABLE  =',F10.2,//,
     4            10X,'OVER-RELAXATION FACTOR  =',F10.3,//,
     5            10X,'PRESCRIBED HEAD TOLERANCE  =',F10.3)
  193 FORMAT(//,10X,'GRAVEL PACK INDEX  =',I5,//,
     1              10X,'AQUITARD INDEX    =',I5,//,
     3              10X,'LAYER OF MAX PERMEABILITY    =',I5)
C
C     READ MATERIAL DATA.
C
      WRITE(NPRINT,6013)
 6013 FORMAT(///,10X,'FORMATION PROPERTIES',//,
     1            10X,'LAYER NO.',5X,'THICKNESS',5X,'COEFF.-A',
     2             5X,'COEFF.-B',5X,'CRIT. VELOCITY',//)
      DO 17 I=1,NLAYR                                                   *NLAYR**
      READ(NREAD,3013)THL(I),AKL(I),BKL(I),VCRL(I)                      $1
 3013 FORMAT(4F10.3)
      GCR=AKL(I)*VCRL(I)+BKL(I)*VCRL(I)**2
      PML(I)=1./AKL(I)
      IF(GCR.GT.0.) PML(I)=VCRL(I)/GCR
      WRITE(NPRINT,7013) I,THL(I),AKL(I),BKL(I),VCRL(I)
 7013 FORMAT(10X,I5,9X,E9.2,3(5X,E9.2))
   17 CONTINUE                                                          /*
      RGP=RW
      TH=THL(IKMAX)
      PM=PML(IKMAX)
      BK=BKL(IKMAX)
      BKMAX=BK
      IF(IGP.EQ.0) GO TO 29
      READ(NREAD,331)AGP,BGP,VGP,THGP,BTGP                              $1,SK
  331 FORMAT(5F10.3)
      RGP=THGP+RW
      GRGP=AGP*VGP+BGP*VGP**2
      PMGP=1./AGP
      IF(GRGP.NE.0) PMGP=VGP/GRGP
      WRITE(NPRINT,153)AGP,BGP,VGP,PMGP
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1              10X,'COEFFICIENT A =',F10.4//
     2              10X,'COEFFICIENT B = ',F10.4//
     3              10X,'CRITICAL VELOCITY = ',F10.4//
     4              10X,'COEFFICIENT K = ',F10.4)
      WRITE(NPRINT,163)RGP,BTGP
  163 FORMAT(10X,'RADIUS OF GRAVEL PACK  =',F10.2,//,
     1            10X,'HEIGHT OF GRAVEL PACK BASE  =',F10.2,//)
   29 CONTINUE
C
C     READ AQUITARD DATA.
C
      IF(IAQTA.EQ.0) GO TO 39
      READ(NREAD,71)PA,THA                                              $1,SK
   71 FORMAT(2F10.3)
      BFAC=PA/THA
      WRITE(NPRINT,273)PA,THA
  273 FORMAT(///,20X,'AQUITARD PROPERTIES',///,
     1              10X,'HYDRAULIC CONDUCTIVITY    =',F10.5//
     2              10X,'THICKNESS    =',F10.2)
   39 CONTINUE
      READ(NREAD,2021)HW                                                $1
 2021 FORMAT(F10.2)
C
C     READ AND PRINT DISCRETIZATION DATA.
C
      READ(NREAD,901)FRLEN,SCFAC,XLMAX                                  $1
  901 FORMAT(3F10.2)
      READ(NREAD,801)NDSC,NSCREN                                        $1
  801 FORMAT(2I10)
      IPENTR=0
      DO 602 I=1,NLAYR                                                  *NLAYR**
      READ(NREAD,1901)IREGL(I),NMINL(I),NFRL(I)                         $1
 1901 FORMAT(3I10)
  602 CONTINUE                                                          /*
C
C     READ WELL SCREEN DATA
C
      WRITE(NPRINT,553)
  553 FORMAT(//,10X,'SCREEN NO.',10X,'BASE HEIGHT',14X,'LENGTH',//)
      SCLEN=0.0                                                         *NSCREN*
      DO 702 I=1,NSCREN                                                 $1
      READ(NREAD,601)XSCR(I),HSCR(I)
  601 FORMAT(2F10.2)
      SCLEN=SCLEN+HSCR(I)
      WRITE(NPRINT,563)I,XSCR(I),HSCR(I)
  563 FORMAT(12X,I6,13X,F11.2,8X,F10.2)                                /*
  702 CONTINUE
C
C     GENERATE AND PRINT DISCRETIZATION DATA.
C
      CALL GXNODR(FRLEN,SCFAC,XLMAX,RW,RO,NNODE,NELEM,LVEC,NLAYR,IPENTR,
     1  IDISCR)
C
```

```
C      MODIFY IPROP.
C
       DO 1544 I=1,NELEM
       J1=NOD(I,1)
       J2=NOD(I,2)
       XCEN=(X(J1,1)+X(J2,1))*0.5
       IF(XCEN.GT.RGP) GO TO 1545
       YCEN = (X(J1,2) + X(J2,2))*0.5
       IF(YCEN.GT.BTGP) IPROP(I) = 0
 1544  CONTINUE
 1545 CONTINUE
C
       HWDRAW=HW-HO
       HODRAW=0.0
       NDW=NDVEC(1)
       NDRO=NDVEC(LVEC)-NDVEC(LVEC-1)
       CALL BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,NSCREN,RW)
       WRITE(NPRINT,203)
  203 FORMAT(///,10X,'PRESCRIBED NODE DATA',//,
     1            10X,'NODE',6X,'PRESCRIBED HEAD VALUE',//)
       DO 200 I=1,NBD
       WRITE(NPRINT,213)JBD(I),DISP(I)
  213 FORMAT(8X,I5,8X,F10.3)
  200 CONTINUE
       IF(IAQTA.EQ.0) GO TO 166
       CALL VDFB(LVEC)
       DO 176 I=1,NNODE
       HINT(I,1)=0.0
  176 CONTINUE
  166 CONTINUE
C
C      FIND BANDWIDTHS ETC.
C
       CALL EBFINR(LEN,NELEM,NN,NNODE)
       WRITE(NPRINT,233)LEN
  233 FORMAT(///,10X,'LENGTH OF GROSS VECTOR  =',I8)
       DO 130 I=1,LEN
       VD(I)=0.0
  130 CONTINUE
       NNP=0
       NT=0
       SC=1.
       DO 135 IE=1,NELEM
       NN=ITYPE(IE)
       IF(IPROP(IE).NE.0) L=IPROP(IE)
       IF(NN.EQ.3) GO TO 235
       CALL ELGNDR(NT,SC,IE,DET,NNP)
       GO TO 335
  235 CONTINUE
       CALL ELGND3(NT,SC,IE)
  335 CONTINUE
       CALL MERBD(NN,IE)
  135 CONTINUE
C
C      SET IDARCY
C
       DO 99 I=1,NELEM
       IDARCY(I)=0
   99 CONTINUE
C
C      SET UP LOOP FOR ITERATION ON HEAD
C
       WRITE(NPRINT,1583) HW
 1583 FORMAT(///,10X,40('*'),/,
     1            10X,'*',3X,'PRESCRIBED WELL DRWADOWN  =',F8.2,'*',/,
     2            10X,40('*'))
       NITER=10
       IF(BKMAX.LE.0.0) NITER=1
       NCOUNT=0
       EMAX=0.0
       DO 999 III=1,NITER
C
C      INITIALISE VECTOR CK.
C
       DO 305 I=1,NNODE
       CK(I,1)=0.0
  305 CONTINUE
C
C      FORM VECTOR VK
C
       DO 300 I=1,LEN
       VK(I)=0.0
  300 CONTINUE
       NT=0
       NDIFF=2
       CALL INFLUC(NN,NDIFF)
       DO 350 I=1,NELEM
       NN=ITYPE(I)
       IF(IPROP(I).EQ.0) GO TO 755
       L=IPROP(I)
       AKK=AKL(L)
       BKK=BKL(L)
```

```
      PKK=PML(L)
      GO TO 365
  755 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
  365 CONTINUE
      IF(NN.EQ.3) GO TO 465
      CALL ELGNCR(I,III,AKK,BKK,NT,PKK)
      GO TO 565
  465 CONTINUE
      NT=NREP(I)
      CALL ELGNCT(I,III,AKK,BKK,NT,VCOUNT,PKK)
  565 CONTINUE
      CALL MERB3(NN,I)
  350 CONTINUE
C
C     ADD LEAKAGE TERM CONTRIBUTED BY OVERLYING AQUITARD
C
      IF(IAQTA.EQ.0) GO TO 86
      BFL=BFAC
      CALL GVMOD(LVEC,BFL,0)
   86 CONTINUE
      NLEN=0
      JJ=JBD(NBW)
      DO 98 I=1,JJ
      NLEN=NLEN+NBAND(I)
   98 CONTINUE
      DO 378 I=1,NLEN
  378 VS(I)=VK(I)
C
C     SOLVE EQNS BY BANDED ELIMINATION SCHEME
C
      NLL=1
      CALL SYMSOL(NNODE,NLL)
C
      IF(III.EQ.1) GO TO 900
      NCOUNT=0
      EMAX=0.0
      DO 450 I=1,NNODE
      EPSI=CK(I,1)-H(I)
      IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.HTOL)  GO TO 460
      NCOUNT=NCOUNT+1
      H(I)=H(I)+ORELAX*EPSI
      GO TO 450
  460 CONTINUE
      H(I)=CK(I,1)
  450 CONTINUE
      IF(NCOUNT.EQ.0) GO TO 1000
      GO TO 1999
  900 CONTINUE
      DO 950 I=1,NNODE
      H(I)=CK(I,1)
  950 CONTINUE
 1999 CONTINUE
      IF(NITER.EQ.1) GO TO 999
      DO  199 I=1,NELEM
      NT=NREP(I)
      NN=ITYPE(I)
      IF(IPROP(I).EQ.0) L=1
      IF(IPROP(I).NE.0) L=IPROP(I)
      AK=AKL(L)
      BK=BKL(L)
      PMK=PML(L)
      VCR=VCRL(L)
      CALL SETARG(AGP,BGP,PMGP,VGP,AK,BK,PMK,VCR,I)
      IF(NN.EQ.3) GO TO 399
      CALL VCHECR(I,AK,BK,NT,PMK,VCR,GRRX,GRRY,NN)
      GO TO 499
  399 CONTINUE
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      CALL VCHEC3(I,AK,BK,J1,J2,J3,NT,PMK,VCR,HRRX,HRRY)
  499 CONTINUE
  199 CONTINUE
  999 CONTINUE
 1000 CONTINUE
      IF(NCOUNT.NE.0)  GO TO 5000
      WRITE(NPRINT,413)III,EMAX
  413 FORMAT(//,10X,'NO. OF ITERATIONS REQUIRED =',I5,//,
     1            10X,'MAXIMUM ERROR IN HEAD =',F10.3)
      WRITE(NPRINT,433)
  433 FORMAT('0',20X,'FINAL SOLUTION',///,
     1            5X,'NODE',10X,'R-COORD',10X,'Z-COORD',10X,'HEAD',
     2           10X,'RHO-COORD',10X,'TZI-COORD',10X,'DRAWDOWN RATIO',/)
      DO 470 I=1,NNODE
      RHO=0.5*X(I,1)/TH
      TZI=X(I,2)/TH
```

```
      SDLESS=(HO-H(I))/(HO-HW)
      WRITE(NPRINT,953)I,X(I,1),X(I,2),H(I),RHO,TZI,SDLESS
  953 FORMAT(5X,I3,2(6X,F10.2),4(8X,F10.4))
  470 CONTINUE
C
C
C     EVALUATE VELOCITY AND GRADIENT USING FINAL HEADS
C
  799 CONTINUE
      VCOUNT=1
C
C     COMPUTE AND PRINT ELEMENT VELOCITIES
C
      IF(IVEL.EQ.0) GO TO 7000
      WRITE(NPRINT,1203)
 1203 FORMAT(///,20X,'*******************************************'/
     1                 20X,'*              ELEMENT VELOCITIES         *'/
     2                 20X,'*******************************************'///
     310X,'ELEM NO.',10X,'RADIAL VEL',10X,'VERTIC VEL',10X,'IDARCY',///)
      DO 3000 I=1,NELEM
      NT=NREP(I)
      NN=ITYPE(I)
      IF(IPROP(I).EQ.0) L=1
      IF(IPROP(I).NE.0) L=IPROP(I)
      AK=AKL(L)
      BK=BKL(L)
      PMK=PML(L)
      VCR=VCRL(L)
      CALL SETARG(AGP,BGP,PMGP,VGP,AK,BK,PMK,VCR,I)
      IF(NN.EQ.3) GO TO 2399
      CALL VCHECR(I,AK,BK,NT,PMK,VCR,HRRX,HRRY,NN)
      GO TO 2499
 2399 CONTINUE
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      CALL VCHEC3(I,AK,BK,J1,J2,J3,NT,PMK,VCR,HRRX,HRRY)
 2499 CONTINUE
      VCOMP1=-VEL(I)*HRRX
      VCOMP2=-VEL(I)*HRRY
      WRITE(NPRINT,1103)I,VCOMP1,VCOMP2,IDARCY(I)
 1103 FORMAT(10X,I5,2(10X,F10.3),10X,I5)
 3000 CONTINUE
 7000 CONTINUE
C
      CALL QCALC(NBW,QSUM)
      BLEAK=RO
      IF(IAQTA.EQ.0) GO TO 4505
      BLEAK=PMK*TH*THA/PA
      BLEAK=SQRT(BLEAK)
 4505 CONTINUE
      J=1
      AK=AKL(IKMAX)
      BK=BKL(IKMAX)
      PMK=PML(IKMAX)
      TH=THL(IKMAX)
      CALL TCURV3(HO,BLEAK,NNODE,QSUM,AK,BK,PMK,TH,J)
 5000 CONTINUE
C---------------------------------------------------------------------C
C                                                                     C
C          IF SO REQUIRED,USER MAY CALL HIS PLOTTING SUBROUTINES      C
C                                                                     C
C          AT THIS LOCATION.                                          C
C                                                                     C
C---------------------------------------------------------------------C
C
C     THE FOLLOWING PLOTS MAY BE OBTAINED -
C          (1) DISCRETISATION PATTERN,OBTAINED BY USING VARIABLES NOD,X,
C              NELEM,NNODE
C          (2) CONTOURS OF HYDRAULIC HEADS,OBTAINED BY USING VARIABLES
C              H,X,NNODE,NOD,NELEM
C          (3) DIMENSIONLESS TYPE CURVES,OBTAINED BY USING VARIABLES HLESS,
C              TLESS,NNODE
C          (4) VELOCITY FIELD,OBTAINED BY USING VARIABLES VEL,X,NOD,VCOMP1,
C              VCOMP2
C          (5) LOCATION OF NON-DARCY FLOW ZONE,OBTAINED BY USING VARIABLES
C              IDARCY,NELEM
C
 4800 CONTINUE
      STOP
      END
      SUBROUTINE GXNODR(FRLEN,SCFAC,XLMAX,RW,RO,NNODE,NELEM,LVEC,NLAYR,
     1IPENTR,IDISCR)
C
C     GENERATES DISCRETIZATION DATA.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      COMMON /ALAYR/AKL(3),BKL(3),VCRL(3),SSL(3),THL(3),IREGL(3),
```

```
     1NMINL(3),NFRL(3),PML(3)
      COMMON /AELEM/NTRAN(4,100),NH1(4,100),NH2(4,100),NDB1(4,100),NDB2(
     14,100),NDWB(22)
      DATA NPRINT/3/
      CALL DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
      NRST=1
      THTO=0.
      NSTFR=1
      NDWB(1)=0
      DO 10 I=1,NLAYR
      THTO=THTO+THL(I)
      NFR=NFRL(I)
      NDWB(1)=NDWB(1)+NFR
      IREG=IREGL(I)
      NMIN=NMINL(I)
      CALL NCRGR1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,I)
   10 CONTINUE
      CALL NCRGR2(NRST,NRR,NLAYR,NSTFR)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,33)NLAYR,NRR
   33 FORMAT('1',20X,'DISCRETISATION DATA',//,
     1         10X,'NUMBER OF LAYERS  =',I5,//,
     2         10X,'NUMBER OF SUBREGIONS  =',I5)
   94 CONTINUE
C
C     DISCRETISE ENTIRE REGION INTO RECTANGULAR ELEMENTS.
C
      KCREP=1
      NSTOR=1
      NTSEL=1
      DO 65 I=NRST,NRR
      DO 691 L=1,NLAYR
      N1=NH1(L,I)
      N2=NH2(L,I)
      NFND1=NDB1(L,I)
      NFND2=NDB2(L,I)
      NPAT=NTRAN(L,I)
      IF(NPAT.NE.0) GO TO 791
      CALL RBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
      GO TO 891
  791 CONTINUE
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
  891 CONTINUE
      NELEM=NTSEL-1
      DO 68 IE=NSTOR,NELEM
      IPROP(IE)=L
   68 CONTINUE
      NSTOR=NTSEL
  691 CONTINUE
   65 CONTINUE
C
C     PRINT OUT ELEMENT DATA.
C
      NELEM=NTSEL-1
      IF(IDISCR.EQ.0) GO TO 104
      WRITE(NPRINT,83)
   83 FORMAT(///,10X,'IDENTIFICATION OF ELEMENT-NODE CONNECTIONS',///,
     112X,'ELEM NO',5X,'ITYPE',5X,'NREP',5X,'IPROP',5X,'NODE1',5X,
     2'NODE2',5X,'NODE3',5X,'NODE4',//)
      DO 80 I=1,NELEM
      NN=ITYPE(I)
      WRITE(NPRINT,93)I,ITYPE(I),NREP(I),IPROP(I),(NOD(I,K),K=1,NN)
   93 FORMAT(12X,9(I5,5X))
   80 CONTINUE
  104 CONTINUE
C
C     GNERATE NODE COORDINATES.
C
      RI=RW
      NS=1
      MM=1
      DO 75 I=NRST,NRR
      DXG=XLEN(I)
      ZI=0.0
      DO 87 L=1,NLAYR
      TH=THL(L)
      NN=NH1(L,I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH
      NS=NS-1
      RI=RI-DXG
   87 CONTINUE
      RI=RI+DXG
      NS=NS+1
   75 CONTINUE
      ZI=0.0
      DO 877 L=1,NLAYR
      TH=THL(L)
      NN=NH2(L,NRR)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
```

```
      ZI=TH
      NS=NS-1
      RI=RI-DXG
  877 CONTINUE
C
C
      NNODE = NS
      IF(IDISCR.EQ.0) GO TO 114
      WRITE(NPRINT,103)
  103 FORMAT('1',10X,'NODAL COORDINATES',//,10X,'NODE',5X,
     1'R-COORD',5X,'Z-COORD',//)
C
      DO 85 I=1,NNODE
      WRITE(NPRINT,113)I,X(I,1),X(I,2)
  113 FORMAT(10X,I4,2F11.2)
   85 CONTINUE
  114 CONTINUE
C
C     GENERATE TOP BOUNDARY COORDINATE VECTORS.
C
      NDTO=NS
      CALL TOPVEC(THTO,LVEC,NDTO)
C
C     PRINT OUT BOUNDARY COORDINATE VECTORS.
      WRITE(NPRINT,223)
  223 FORMAT(///,10X,'TOP BOUNDARY NODES AND RADIAL COORDINATES',///,
     110X,'NODE NUMBER',20X,'R-COORDINATE',//)
      DO 335 I=1,LVEC
      WRITE(NPRINT,333)NDVEC(I),RVEC(I)
  335 CONTINUE
  333 FORMAT(10X,I7,25X,F10.2)
      RETURN
      END
      SUBROUTINE NCRGR1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,L)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NTRAN,NH1,NH2,NDB1,NDB2
C
      COMMON /AELEM/NTRAN(4,100),NH1(4,100),NH2(4,100),NDB1(4,100),
     1NDB2(4,100),NDWB(22)
      NC=NFR
      NCOUNT=0
      DO 10 I=NRST,NRR
      IRGG=IREG
      IF((I.EQ.NRST).AND.(IPENTR.NE.0)) IRGG=0
      NCOUNT=NCOUNT+1
      IF(NCOUNT.LT.IRGG) GO TO 20
      NCOUNT=0
      NHALF=NC/2
      NREM=NC-2*NHALF
      IF(NREM.GT.0) GO TO 15
      NC1=NC-1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=2
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=NC+1
      GO TO 10
   15 CONTINUE
      NC1=NHALF+1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=1
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=2*NH2(L,I)-1
      GO TO 10
   20 CONTINUE
      NH2(L,I)=NC
      NTRAN(L,I)=0
      NH1(L,I)=NH2(L,I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE NCRGR2(NRST,NRR,NLAYR,NSTFR)
C
C     MODIFIES THE VALUES OF NDB1 AND NDB2.
C
      COMMON /AELEM/NTRAN(4,100),NH1(4,100),NH2(4,100),NDB1(4,100),
     1NDB2(4,100),NDWB(22)
      NDBC=NSTFR
      DO 50 J=NRST,NRR
      NDB1(1,J)=NDBC
      IF(NLAYR.LT.2) GO TO 56
      DO 55 L=2,NLAYR
      NDB1(L,J)=NDB1(L-1,J)+NH1(L-1,J)-1
   55 CONTINUE
   56 CONTINUE
      L=NLAYR
      NDBC=NDB1(L,J)+NH1(L,J)
   50 CONTINUE
      NR1=NRR-1
```

```
      DO 60 J=NRST,NR1
      DO 65 L=1,NLAYR
      NDB2(L,J)=NDB1(L,J+1)
   65 CONTINUE
   60 CONTINUE
      NDB2(1,NRR)=NDBC
      DO 75 L=2,NLAYR
      NDB2(L,NRR)=NDB2(L-1,NRR)+NH2(L-1,NRR)-1
   75 CONTINUE
      RETURN
      END
      SUBROUTINE DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NRR,XLEN
C
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      MAXNR=89
      XLEN(1)=FRLEN
      RLEN=RO-RW
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      IF(XLEN(I).GT.XLMAX) XLEN(I)=XLMAX
      SUM=XLEN(I)+SUM
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I
      DENOM=1.+SCFAC+SCFAC**2
      XLEN(I-2)=(XREM+XLEN(I-1)+XLEN(I-2))/DENOM
      XLEN(I-1)=XLEN(I-2)*SCFAC
      XLEN(I)=XLEN(I-1)*SCFAC
      RETURN
      END
      SUBROUTINE RBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC RECTANGULAR BLOCKS.
C
      DIMENSION ND(1,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND2+1
      ND(1,4)=NFND1+1
      IF(NPAT.EQ.1) GO TO 200
      NTEMP=N1-2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=4
      DO 50 K=1,4
      NOD(J1,K)=ND(1,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
      GO TO 400
  200 CONTINUE
      ND(1,5)=ND(1,4)
      ND(1,4)=ND(1,5)+1
      N2=N1/2+1
      NTEMP=N2-2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=5
      NI=J1-NST
      NOD(J1,1)=ND(1,1)+2*NI
      NOD(J1,4)=ND(1,4)+2*NI
      NOD(J1,5)=ND(1,5)+2*NI
      NOD(J1,2)=ND(1,2)+NI
      NOD(J1,3)=ND(1,3)+NI
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  400 CONTINUE
      RETURN
      END
      SUBROUTINE COORDC(NS,MM,NN,DXG,TH,RI,ZI)
C
C     GENERATES NODAL COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /ASOLV/XX(5,90),XY(5,90)
      DY=TH/(NN-1)
      DO 20 L=1,NN
   20 XX(1,L)=RI
```

```
      DO 50 K=1,MM
   50 XY(K,1)=ZI
      DO 60 J=2,NN
      JM1=J-1
      DO 70 K=1,MM
      XY(K,J)=XY(K,JM1)+DY
   70 CONTINUE
   60 CONTINUE
      DO 80 J=1,MM
      DO 90 K=1,NN
      L=NN*(J-1)+K+NS-1
      X(L,1)=XX(J,K)
      X(L,2)=XY(J,K)
   90 CONTINUE
   80 CONTINUE
      NEN=NS+(MM*NN)-1
      NS=NEN+1
      RI=X(NEN,1)+DXG
      RETURN
      END
      SUBROUTINE TOPVEC(TH,LVEC,NDTO)
C
C     COMPUTES RADIAL COORDINATES FOR NODES ALONG THE TOP BOUNDARY.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      J=0
      DO 10 I=1,NDTO
      ZDIF=ABS(TH-X(I,2))
      IF(ZDIF.GT.0.001) GO TO 10
      J=J+1
      NDVEC(J)=I
      RVEC(J)=X(I,1)
   10 CONTINUE
      LVEC=J
      RETURN
      END
      SUBROUTINE EBFINR(LEN,NM,NN,LL)
C
C     SUBROUTINE TO GENERATE THE BANDWIDTHS OF THE BANDED SYMMETRIC
C     MATRIX. NBAND CONTAINS THE BANDWIDTHS, ID THE POSITION OF THE
C     TERM ON THE DIAGONALS OF THE ORIGINAL MATRIX, LEN IS THE LENGTH
C     THE VECTOR
C
      DIMENSION LV(6)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
C
      DO 20 I=1,LL
      NBAND(I)=1
      ISTART(I)=I
   20 CONTINUE
C
C     SCAN THROUGH THE LOCATION VECTOR FOR EACH MEMBER TO FIND
C     THE POSITION OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW
C
      DO 25 I=1,NM
      NN=ITYPE(I)
      NF2=NN
      DO 30 J=1,NF2
      LV(J)=NOD(I,J)
   30 CONTINUE
      DO 45 J=1,NF2
      IF(LV(J).EQ.0)GO TO 45
      DO 40 K=1,NF2
      IF(LV(J).GT.LV(K)) GO TO 40
      NW=LV(K)-LV(J)+1
      NR=LV(J)
      IF(NW.GT.NBAND(NR))  NBAND(NR)=NW
   40 CONTINUE
   45 CONTINUE
C      SEARCH FOR THE FURTHEREST OFF-LEFT TERM
C
      DO 55 J=1,NF2
      IF(LV(J).EQ.0) GO TO 55
      DO 65 K=1,NF2
      IF(LV(J).LT.LV(K)) GO TO 65
      NW=LV(K)
      NR=LV(J)
      IF(NW.LT.ISTART(NR)) ISTART(NR)=NW
   65 CONTINUE
   55 CONTINUE
   25 CONTINUE
C
C     SET UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN=NBAND(1)
      DO 50 I=2,LL
      IF(NBAND(I)-NBAND(I-1).LT.0)  NBAND(I)=NBAND(I-1)-1
```

```
      LEN=LEN+NBAND(I)
   50 CONTINUE
      ID(1)=1
      DO 60 I=2,LL
   60 ID(I)=ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE ELGNCR(M,III,AK,BK,NT,PMK)
C
C     GENERATES ELEMENT MATRIX,SELK,FOR RECTANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /BELEM/E(6,6,3),F(6,6,3),DSHFZ(2,6)
      NN=ITYPE(M)
      KK=NN-3
      IF(NREP(M).EQ.NT) GO TO 30
      NT=NREP(M)
      DO 10 J=1,NN
      DO 20 K=1,NN
      IF(J.GT.K) GO TO 20
      EK(J,K)=BA(NT)*E(J,K,KK)+F(J,K,KK)/BA(NT)
      EK(J,K)=EK(J,K)*ORX(M)
   20 CONTINUE
   10 CONTINUE
   30 CONTINUE
      IF( IDARCY(M).EQ.0) GO TO 70
  500 CONTINUE
      CALL CONSTR(M,AK,BK,ACONST,NT,NN)
      GO TO 80
   70 CONTINUE
      ACONST=PMK
   80 CONTINUE
      DO 40 J=1,NN
      DO 50 K=1,NN
      IF(J.GT.K) GO TO 60
      SELK(J,K)=EK(J,K)*ACONST
      GO TO 50
   60 SELK(J,K)=SELK(K,J)
   50 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE INFLUC(NN,NDIFF)
C
C     GENERATES FIRST SET OF INFLUENCE COEFFICIENTS.
C
      COMMON /BELEM/E(6,6,3),F(6,6,3),DSHFZ(2,6)
      DO 200 I=1,NDIFF
      NN=I+3
      J=I
      GO TO (10,30),J
C
C     4-NODES ELEM
C
   10 CONTINUE
      DO 150 K=1,4
      E(K,K,J)=1./3.
  150 F(K,K,J)=E(K,K,J)
      E(1,2,J)=-1./3.
      E(1,3,J)=-1./6.
      E(1,4,J) = 1./6.
      E(2,3,J)=1./6.
      E(2,4,J)=-1./6.
      E(3,4,J)=-1./3.
      F(1,2,J)=1./6.
      F(1,3,J)=-1./6.
      F(1,4,J)=-1./3.
      F(2,3,J)=-1./3.
      F(2,4,J)=-1./6.
      F(3,4,J)=1./6.
      GO TO 500
C
C     5 NODES LEFT INCLUDED
C
   30 CONTINUE
      E(1,1,J) =  2./15.
      E(1,2,J) =  -1./6.
      E(1,3,J) = 0.0
      E(1,4,J)=   -1./30.
      E(1,5,J) =  1./15.
      E(2,2,J) =  1./3.
      E(2,3,J) =  1./6.
      E(2,4,J) =  0.0
      E(2,5,J) =  -1./3.
      E(3,3,J) =  1./3.
      E(3,4,J) =  -1./6.
      E(3,5,J) = -1./3.
      E(4,4,J) = 2./15.
```

```
      E(4,5,J) = 1./15.
      E(5,5,J) =  8./15.
      F(1,1,J) =  7./9.
      F(1,2,J) =  1./6.
      F(1,3,J) =  -1./6.
      F(1,4,J)=   1./9.
      F(1,5,J) =  -8./9.
      F(2,2,J) =  1./3.
      F(2,3,J) =  -1./3.
      F(2,4,J) =  -1./6.
      F(2,5,J) =  0.0
      F(3,3,J) = 1./3.
      F(3,4,J) =  1./6.
      F(3,5,J) =  0.0
      F(4,4,J) =    7./9.
      F(4,5,J) =  -8./9.
      F(5,5,J) =  16./9.
  500 CONTINUE
  200 CONTINUE
      RETURN
      END
      SUBROUTINE INFLUD(NN)
C
C     GENERATES SECOND SET OF INFLUENCE COEFFICIENTS.
C
      DIMENSION ZEN(6,2),ZE(1,2)
      COMMON /BELEM/E(6,6,3),G(6,6,3),DSHFZ(2,6)
      JP=1
      K=NN-3
      ZEN(1,1)=-1.
      ZEN(2,1)=1.
      ZEN(3,1)=1.
      ZEN(4,1)=-1.
      ZEN(1,2)=-1.
      ZEN(2,2)=-1.
      ZEN(3,2)=1.
      ZEN(4,2)=1.
      ZEN(5,1)=0.
      ZEN(5,2)=-1.
      ZE(1,1)=0.
      ZE(1,2)=0.
      IF(K.EQ.2) GO TO 50
C
C     4-NODE ELEMENT.
C
      DO 2 I=1,NN
      T1=1.+ZEN(I,1)*ZE(JP,1)
      T2=1.+ZEN(I,2)*ZE(JP,2)
      DSHFZ(1,I)=.25*T2*ZEN(I,1)
      DSHFZ(2,I)=.25*T1*ZEN(I,2)
      G(I,I,K)=4./9.
      IF(I.LE.3) G(I,I+1,K)=2./9.
      IF(I.LE.2) G(I,I+2,K)=1./9.
      IF(I.EQ.1) G(I,4,K)=2./9.
    2 CONTINUE
      GO TO 100
   50 CONTINUE
C
C
C     5-NODE ELEMENT WITH MIDDLE NODE ON SIDE 1-4
C
      Z1=ZE(JP,1)
      Z2=ZE(JP,2)
      P1=1.0+Z1
      S1=1.0-Z1
      P2=1.0+Z2
      S2=1.0-Z2
      DSHFZ(1,1)=0.25*Z2*S2
      DSHFZ(1,4)=-0.25*Z2*P2
      DSHFZ(2,1)=-0.25*S1*(S2-Z2)
      DSHFZ(2,4)=0.25*S1*(P2+Z2)
C
      DSHFZ(1,2)=0.25*S2
      DSHFZ(1,3)=0.25*P2
      DSHFZ(2,2)=-0.25*P1
      DSHFZ(2,3)=-DSHFZ(2,2)
C
      DSHFZ(1,5)=-0.5*S2*P2
      DSHFZ(2,5)=-Z2*S1
C
      G(1,1,K)=0.177777
      G(1,2,K)=0.111111
      G(1,3,K)=0.00000
      G(1,4,K)=-0.444444E-01
      G(1,5,K)=0.888886E-01
      G(2,2,K)=0.444444
      G(2,3,K)=0.222222
      G(2,4,K)=0.372529E-08
      G(2,5,K)=0.222222
      G(3,3,K)=0.444444
```

```
      G(3,4,K)=0.111111
      G(3,5,K)=0.222222
      G(4,4,K)=0.177777
      G(4,5,K)=0.888886E-01
      G(5,5,K)=0.711111
  100 CONTINUE
      RETURN
      END
      SUBROUTINE ELGNDR(NT,SC,M,DET,NNP)
C
C     GENERATES ELEMENT MATRIX,D,FOR RECTANGULAR ELEMENTS.
C
      DIMENSION TJ(2,2)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /BELEM/E(6,6,3),G(6,6,3),DSHFZ(2,6)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      J4=NOD(M,4)
      ORX(M)=(X(J1,1)+X(J2,1))*0.5
      IF(NREP(M).EQ.NT) GO TO 50
      NT=NREP(M)
      NN=ITYPE(M)
      K=NN-3
      BB=ABS(X(J4,2)-X(J1,2))
      AA=ABS(X(J3,1)-X(J1,1))
      BA(NT)=BB/AA
      DET= AA*BB/4.
      TJ(1,1)=2./AA
      TJ(2,2)=2./BB
      IF(NNP.EQ.NN) GO TO 15
      CALL INFLUD(NN)
   15 CONTINUE
      NNP=NN
      DO 10 I=1,NN
      SLX(NT,I)=TJ(1,1)*DSHFZ(1,I)
      SLY(NT,I)=TJ(2,2)*DSHFZ(2,I)
   10 CONTINUE
C
C     FORM ELEMENT MATRIX:-D
C
      DO 100 I=1,NN
      DO 100 J=1,NN
      IF(J-I) 105,110,110
  110 D(I,J)=ORX(M)*SC*DET*G(I,J,K)
      GO TO 100
  105 D(I,J)=D(J,I)
  100 CONTINUE
   50 CONTINUE
      RETURN
      END
      SUBROUTINE CONSTR(M,AK,BK,ACONST,NT,NN)
C
C     COMPUTES ELEMENT VELOCITIES FOR RECTANGULAR ELEMENTS AND CHECK IF A
C     PARTICULAR ELEMENT BELONGS TO DARCY OR NON-DARCY FLOW ZONE.
C
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      HX=0.0
      HY=0.0
      DO 10 JN=1,NN
      J=NOD(M,JN)
      HX=HX+SLX(NT,JN)*H(J)
      HY=HY+SLY(NT,JN)*H(J)
   10 CONTINUE
      HS=ABS(SQRT(HX**2+HY**2))
C
C     EVALUATES ELEMENT VELOCITY.
C
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      ACONST=1./TEMP
      RETURN
      END
      SUBROUTINE VCHECR(M,AK,BK,NT,PMK,VCR,GRRX,GRRY,NN)
C
C     COMPUTES ELEMENT VELOCITIES FOR RECTANGULAR ELEMENTS.
C
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      HX=0.0
      HY=0.0
      DO 10 JN=1,NN
      J=NOD(M,JN)
      HX=HX+SLX(NT,JN)*H(J)
      HY=HY+SLY(NT,JN)*H(J)
```

```
   10 CONTINUE
      HS=ABS(SQRT(HX**2+HY**2))
      GRRX=HX/HS
      GRRY=HY/HS
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE BNDFIX(IPENTR,ZB,HW,HO,NNODE,NBD,NDW,NDRO,NBW)
C
C     LOCATES NODES WHERE HEAD VALUES ARE FIXED.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      ZTOL=0.01
      J=0
      DO 10 I=1,NDW
      ZDIF=X(I,2)-ZB
      IF(IPENTR.NE.0) ZDIF=-ZDIF
      IF(ZDIF.GT.ZTOL) GO TO 10
      J=J+1
      JBD(J)=I
      DISP(J)=HW
   10 CONTINUE
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
      NBD=J
      RETURN
      END
      SUBROUTINE MERB3(N,M)
C
C     MERGES ELEMENT MATRIX SELK INTO GROSS VECTOR VK.
C
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VK(IPOS)=VK(IPOS)+SELK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MERBD(N,M)
C
C     MERGES ELEMENT MATRIX D INTO GROSS VECTOR VD.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VD(IPOS)=VD(IPOS)+D(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SYMSOL(LL,NLL)
C
C     AN IN CORE BAND SOLVER.
C     USE IN CONJUNCTION WITH TOPOLOGICAL SUBROUTINE FOR LARGE BANDWIDTH.
C
      DIMENSION VTEMP(90)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),N(300),IDUM(300)
      COMMON /BSOLV/C(300,1),V(2000)
```

```
      DOUBLE PRECISION TEMP,VTEMP
      JBOUN=1
      ID=1
      DO 10 I=1,LL
      TEMP=V(ID)
      NEB=ID+N(I)-1
      ID1=ID+1
      IF(I.EQ.JBD(JBOUN)) GO TO 16
C
C     NORMALISE ROW I
C
      KK=0
      V(ID)=1.0
      IF(ID1.GT.NEB) GO TO 29
      DO 20 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
   20 V(J)=V(J)/TEMP
   29 CONTINUE
      DO 25 L=1,NLL
      C(I,L)=C(I,L)/TEMP
   25 CONTINUE
      GO TO 46
   16 CONTINUE
      IF(ID1.GT.NEB) GO TO 39
      KK=0
      DO 120 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
  120 V(J)=0.0
   39 CONTINUE
      DO 125 L=1,NLL
  125 C(I,L)=DISP(JBOUN)
      JBOUN=JBOUN+1
C
C     ELIMINATION
C
   46 CONTINUE
      IDJ=ID
      IF(ID1.GT.NEB)  GO TO 35
      KK=0
      DO 30 J=ID1,NEB
      JI=J-ID1+I
      IDJ=IDJ+N(JI)
      KK=KK+1
      IF(VTEMP(KK)) 50,30,50
   50 CONTINUE
      IF(JBOUN.EQ.1) GO TO 240
      JBM1=JBOUN-1
      IF(I.EQ.JBD(JBM1)) GO TO 140
  240 IDP=J
      DO 40 K=IDP,NEB
      KJ=IDJ+K-J
   40 V(KJ)=V(KJ)-V(K)*VTEMP(KK)
  140 CONTINUE
      NJ=I+J-ID
      DO 32 L=1,NLL
      C(NJ,L)=C(NJ,L)-C(I,L)*VTEMP(KK)
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID=ID+N(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID=ID-1
      LL1=LL-1
      DO 70 IB=1,LL1
      I=LL1-IB+1
      ID=ID-N(I)
      IS=I+1
      IN=I+N(I)-1
      DO 80 J=IS,IN
      NJ=ID+J-I
      DO 75 L=1,NLL
      C(I,L)=C(I,L)-C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE VDFB(LVEC)
C
C     GENERATES VECTOR VDTOP.
C
      DIMENSION D(2,2)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
```

```
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      NELTOP=LVEC-1
      NN=2
      ID(1)=1
      DO 10 I=2,LVEC
   10 ID(I)=ID(I-1)+2
      LENT=2*LVEC-1
      DO 108 I=1,LENT
      VDTOP(I)=0.0
  108 CONTINUE
      DO 208 IE=1,NELTOP
      IN=NDVEC(IE)
      IP=NDVEC(IE+1)
      RAVE=0.5*(X(IP,1)+X(IN,1))
      RDIF=X(IP,1)-X(IN,1)
      D(1,1)=RAVE*RDIF/3.0
      D(2,2)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      DO 308 I=1,2
      IK=IE+(I-1)
      DO 408 J=1,2
      JK=IE+(J-1)
      IF(IK.GT.JK) GO TO 408
      IPOS=ID(IK)+JK-IK
      VDTOP(IPOS)=VDTOP(IPOS)+D(I,J)
  408 CONTINUE
  308 CONTINUE
  208 CONTINUE
      RETURN
      END
      SUBROUTINE GVMOD(LACT,SY,IGK)
C
C     MODIFIES VECTORS VK AND GK TO TAKE ACCOUNT OF LEAKAGE FLUX ACROSS
C     TOP BOUNDARY.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      J2=J1
      JM=J1
      IF(L.LT.LACT) J2=NDVEC(L+1)
      IF(L.GT.1) JM=NDVEC(L-1)
      SUM=VF1*HINT(J1,1)+VF2*HINT(J2,1)+VFM*HINT(JM,1)
      IF(IGK.GT.0) GK(J1,1)=GK(J1,1)+SUM
      IF(IGK.GT.0) GO TO 1078
      IS=ID(I)
      VK(IS)=VK(IS)+VF1
      IL=IS+NBAND(I)-1
      VK(IL)=VK(IL)+VF2
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE GKMOD(LACT,SY)
C
C     MODIFIES GK TO TAKE ACCOUNT OF LEAKAGE FLUX ACROSS TOP BOUNDARY.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      LM=L
      LP=L
      IF(L.LT.LACT) LP=L+1
      IF(L.GT.1) LM=L-1
      SUM=VF1*QLEAK(L)+VF2*QLEAK(LP)+VFM*QLEAK(LM)
      GK(J1,1)=GK(J1,1)+SUM
```

```
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC BLOCKS.
C
      DIMENSION ND(3,3)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND1+1
      ND(2,1)=ND(1,3)
      ND(2,2)=ND(1,2)
      ND(2,3)=ND(1,2)+1
      IF(NPAT.EQ.1) GO TO 200
      IF(NPAT.EQ.0) GO TO 265
      NTEMP=N1-2
      DO 90 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      DO 50 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      NTEMP=NTEMP-1
      NTSEL=NLST+1
   90 CONTINUE
      GO TO 400
  200 CONTINUE
      ND(3,1)=ND(2,1)
      ND(3,2)=ND(2,2)+1
      ND(3,3)=ND(3,1)+1
      NTEMP=(N1-1)/2-1
      L1=2
      L2=1
      DO 250 JJ=1,3
      L3=2
      IF(JJ.EQ.2) L3=1
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=3
      DO 270 K=1,3
      IF(K.EQ.1) LL=L1
      IF(K.EQ.2) LL=L2
      IF(K.EQ.3) LL=L3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)*LL
  270 CONTINUE
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  250 CONTINUE
      GO TO 400
  265 CONTINUE
      NTEMP=N1-2
      DO 190 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 140 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=3
      DO 150 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
  150 CONTINUE
  140 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  190 CONTINUE
  400 CONTINUE
      RETURN
      END
      SUBROUTINE ELGND3(NT,SC,M)
C
C     GENERATES ELEMENT MATRIX,D,FOR TRIANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      IF(NREP(M).EQ.NT) GO TO 50
      NT=NREP(M)
      XJ=X(J2,1)-X(J1,1)
```

```
      XM=X(J3,1)-X(J1,1)
      YJ=X(J2,2)-X(J1,2)
      YM=X(J3,2)-X(J1,2)
      AREA(NT)=0.5*(XJ*YM-XM*YJ)
      AFUN=2.*AREA(NT)
      B(NT,1)=(YJ-YM)/AFUN
      B(NT,2)=   (YM)/AFUN
      B(NT,3)=  (-YJ)/AFUN
      C(NT,1)=(XM-XJ)/AFUN
      C(NT,2)=  (-XM)/AFUN
      C(NT,3)=   (XJ)/AFUN
      DO 100 I=1,3
      DO 100 J=1,3
      IF(J-I) 105,110,110
  110 ELK(I,J,NT)=B(NT,I)*B(NT,J)+C(NT,I)*C(NT,J)
      GO TO 100
  105 ELK(I,J,NT)=ELK(J,I,NT)
  100 CONTINUE
C
C     FORM ELEMENT MATRIX:- D
C
      D(1,1)=0.5*ORX(M)*AREA(NT)*SC/3.
      D(2,2)=D(1,1)
      D(3,3)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      D(1,3)=D(1,2)
      D(3,1)=D(1,3)
      D(2,3)=D(1,2)
      D(3,2)=D(2,3)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE ELGNCT(M,III,AK,BK,NT,VCOUNT,PMK)
C
C     GENERATES ELEMENT MATRIX,SELK,FOR TRIANGULAR ELEMENTS.
C
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      IF(IDARCY(M).EQ.0) GO TO 70
  500 CONTINUE
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      CALL CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
      GO TO 80
   70 CONTINUE
      PX=PMK
      ACONST=PX*ORX(M)*AREA(NT)
   80 CONTINUE
C
C                 CALCULATE ELEMENT MATRIX
C
      DO 200 I=1,3
      DO 200 J=1,3
      SELK(I,J)=ACONST*ELK(I,J,NT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
C
C     COMPUTES NON-LINEAR COEFFICIENT FOR TRIANGULAR ELEMENTS.
C
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      ACONST=AREA(NT)*ORX(M)/TEMP
  500 CONTINUE
      RETURN
      END
      SUBROUTINE VCHEC3(M,AK,BK,J1,J2,J3,NT,PMK,VCR,GRRX,GRRY)
C
C     COMPUTES ELEMENT VELOCITIES FOR TRIANGULAR ELEMENTS.
C
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      GRRX=HX/HS
      GRRY=HY/HS
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
```

```
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SETARG(AGP,BGP,PMGP,VGP,AKK,BKK,PKK,VCR,I)
C
C     IDENTIFIES ELEMENTS IN THE GRAVEL PACKED ZONE.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      IF(IPROP(I).EQ.0) GO TO 20
      GO TO 10
   20 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
      VCR=VGP
   10 CONTINUE
      RETURN
      END
      SUBROUTINE TCURV3(HO,B,NNODE,QFIX,AK,BK,PMK,TH,J)
C
C     COMPUTES TYPE CURVE FOR STEADY STATE FLOW.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      TWPI=44./7.
      TMIS=PMK*TH
      PARAM=BK*QFIX*TMIS/(TH**2*B*TWPI)
      DENOM=QFIX/(TWPI*TMIS)
      WRITE(3,3)PARAM
    3 FORMAT(///,10X,50('*'),/,10X,'*',13X,'STEADY STATE TYPE CURVE',
     1 12X,'*',/,10X,50('*'),///,15X,' NON-LINEAR FACTOR    =',F12.4)
      IF(J.EQ.2) GO TO 5
      WRITE(3,23)
      GO TO 15
    5 CONTINUE
      WRITE(3,33)
   15 CONTINUE
   23 FORMAT(//,10X,'NODE NUMBER',15X,'R-COORDINATE',10X,'FUNCTION W(U)'
     1          ,10X,'ARGUMENT U',//)
   33 FORMAT(//,10X,'NODE NUMBER',15X,'Z-COORDINATE',10X,'FUNCTION W(U)'
     1          ,10X,'ARGUMENT U',//)
      DO 10 I=1,NNODE
      SDRAW=HO-H(I)
      SLESS=SDRAW/DENOM
      RLESS=X(I,1)/B
      WRITE(3,13)I,X(I,J),SLESS,RLESS
   10 CONTINUE
   13 FORMAT(10X,I5,20X,F10.2,2(10X,E12.4))
      RETURN
      END
      SUBROUTINE QMUL(NBW,QSUM)
C
C     COMPUTES NODAL FLUXES AT WELL BOUNDARY.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      DATA NREAD,NPRINT/1,3/
      DO 10 L=1,NBW
      I=JBD(L)
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
      QWB(L)=0.0
      DO 20 J = IS, IL
      K=I+J-IS
      QWB(L)=QWB(L)+VKQ(J)*H(K)
   20 CONTINUE
      IF(I.EQ.1) GO TO 15
      I1 = I-1
      IST= ISTART(I)
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
      QWB(L)=QWB(L)+VKQ(IP)*H(J)
   30 CONTINUE
   15 CONTINUE
      QSUM=QSUM+QWB(L)
      QTEMP=-QWB(L)*2.*3.1416
      WRITE(NPRINT,4403)I,QTEMP
 4403 FORMAT(/,10X,I5,15X,F12.4)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE QCALC(NBW,QSUM)
```

```
C
C     COMPUTES TOTAL DISCHARGE INTO WELL.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,4103)
 4103 FORMAT(1H1,//,20X,29HNODAL FLUXES AT WELL BOUNDARY,///,
     1        10X,4HNODE,16X,14HDISCHARGE FLUX,/)
      QSUM=0.0
      CALL QMUL(NBW,QSUM)
      QSUM=ABS(2.*3.1416*QSUM)
      WRITE(NPRINT,4203)QSUM
 4203 FORMAT(////,10X,33HTOTAL DISCHARGE INTO THE WELL   = ,F10.4)
      RETURN
      END
      SUBROUTINE BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,NSCREN,RW)
C
C     IDENTIFIES NODES WHERE HEAD VALUES ARE PRESCRIBED.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      K=0
      NFND=1
      DO 30 I=1,NSCREN
      XST=XSCR(I)-0.01
      XEND=XST+HSCR(I)+0.02
      DO 40 J=1,NNODE
      L=NFND+J-1
      IF(L.EQ.1) GO TO 222
      IF(X(L,1).GT.RW) GO TO 30
  222 CONTINUE
      IF((X(L,2).LT.XST).OR.(X(L,2).GT.XEND)) GO TO 70
   22 K=K+1
      JBD(K)=L
      DISP(K)=HW
      GO TO 40
   70 IF(X(L,2).GT.XEND) GO TO 30
   40 CONTINUE
      NFND=L+1
   30 CONTINUE
      J=K
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
      NBD=J
      RETURN
      END
```

4.6 LISTING OF TRCOND

```
C
C     INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C     DEVELOPED BY P.S. HUYAKORN.
C
C     TRCOND,PROGRAM FOR SOLVING TRANSIENT,TWO-DIMENSIONAL,DARCY OR TWO-REGIME
C
C     FLOW USING RECTANGULAR ELEMENTS OR A COMBINATION OF RECTANGULAR AND
C
C     TRIANGULAR ELEMENTS.
C
C     VERSION DATED OCTOBER,1973.
C
C      FOR FURTHER INFORMATION,CONTACT
C
C        P.S. HUYAKORN OR C.R. DUDGEON
C
C        WATER RESEARCH LABORATORY
C
C        KING ST.,MANLY VALE
C
C        SYDNEY,N.S.W. 2093,AUSTRALIA.
C
C-----------------------------
C     LIST OF INPUT VARIABLES
C-----------------------------
C
C---------------ALL READ STATEMENTS ARE LOCATED IN THE MAIN------------------
C
C---------------PROGRAM AND INDICATED BY  $1 OR $1,SK  SIGN------------------
C
C***  PROBLEM VARIABLES  ***
C
C     NPROB     =  NUMBER OF PROBLEMS TO BE SOLVED
C     IVEL      =  VELOCITY PRINT-OUT INDEX
C                  FEED IN IVEL=0 IF VELOCITY PRINT-OUT IS NOT REQUIRED OTHERWISE
C                  FEED IN IVEL=1
C     IDISCR    =  DISCRETISATION DATA PRINT-OUT INDEX
C                  FEED IN IDISCR=0 IF DISCRETISATION PRINT-OUT IS NOT REQUIRED
C                  OTHERWISE FEED IN IDISCR=1
C     ORELAX    =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                  SUGGESTED VALUE LIES BETWEEN 1.50 TO 1.85
C     RW        =  RADIUS OF WELL SCREEN
C     RO        =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C     HO        =  INITIAL HEIGHT OR DRAWDOWN OF WATER TABLE
C     HTOL      =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                  SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C     NLAYR     =  NUMBER OF LAYERS OF WATER BEARING FORMATIONS
C     QFIX      =  PRESCRIBED WELL DISCHARGE
C     RCSNG     =  RADIUS OF WELL CASING
C     QRTOL     =  RATIO OF DISCHARGE TOLERANCE TO PRESCRIBED WELL DISCHARGE
C                  SUGGESTED VALUE LIES BETWEEN 0.01 AND 0.02
C     IGP       =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                  IGP=0 FOR NON-GRAVEL PACKED WELL
C     IBOUND    =  EXTERNAL BOUNDARY INDEX
C                  IBOUND=0 FOR BARRIER BOUNDARY OTHERWISE IBOUND =1
C     IWBC      =  WELL BOUNDARY CONDITION INDEX
C                  IWBC=0 IF EFFECT OF WELL STORAGE IS TO BE NEGLECTED OTHERWISE
C                  IWBC=1
C     IKMAX     =  LAYER OF MAXIMUM PERMEABILITY OR MINIMUM VALUE OF AKL(I)
C     IWAT      =  INDEX USED TO INDICATE WHETHER THE TOP LAYER IS CONFINED OR
C                  UNCONFINED
C     THL(I)    =  THICKNESS OF LAYER NUMBER I
C     AKL(I)    =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER OR AQUITARD
C                  LAYER I
C                  FOR AQUITARD I,FEED IN AKL(I)=1./PML(I) WHERE PML(I) IS ITS
C                  HYDRAULIC CONDUCTIVITY
C     BKL(I)    =  FORCHHEIMER NON-LINEAR COEFFICIENT OF AQUIFER LAYER I
C                  FOR AQUITARD I,FEED IN BKL(I)=0.0
C     VCRL(I)   =  CRITICAL VELOCITY OF AQUIFER LAYER I
C                  FOR AQUITARD I FEED IN VCRL(I)=0.0
C     SSL(I)    =  SPECIFIC STORAGE OF AQUIFER OR AQUITARD I
C     AGP       =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     BGP       =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     VGP       =  CRITICAL FLOW VELOCITY FOR GRAVEL PACK MATERIAL
C     THGP      =  THICKNESS OF GRAVEL PACK
C     BTGP      =  HEIGHT OF THE BASE OF GRAVEL PACK
C     SY        =  COEFFICIENT OF SPECIFIC YIELD OF TOP LAYER WATER TABLE AQUIFER
C                  OR AQUITARD
C     DINDEX    =  RECIPROCAL OF DELAYED YIELD INDEX FOR WATER TABLE AQUIFER OR
C                  AQUITARD
C
C***  DISCRETISATION PARAMETERS  ***
C
C     NTICR     =  TOTAL NUMBER OF TIME STEPS
C     ITST      =  STARTING TIME STEP NUMBER
C     TFACTR    =  VALUE OF THE FIRST TIME STEP,EXPRESSED IN DIMENSIONLESS FORM
C     TMUL      =  TIME MULTIPLIER,SUGGESTED VALUE VARIES FROM 1.50 TO 2.00
C     DTMUL     =  INCREMENT OF TIME MULTIPLIER
C                  SUGGESTED VALUE LIES BETWEEN 0. TO 0.02
C
```

```
C      FRLEN    =  LENGTH OF FIRST SUBREGION
C                  SUGGESTED VALUE IS FRLEN=RW
C                  FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
C      SCFAC    =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                  SUBREGIONS.SUGGESTED VALUE IS SCFAC=1.50
C      XLMAX    =  MAXIMUM LENGTH OF A BLOCK,PRESCRIBED TO AVOID ILL-CONDITIONED
C                  ELEMENTS.MAXIMUM VALUE OF XLMAX SHOULD NOT EXCEED 25.*TH
C      NDSC     =  TOTAL NUMBER OF NODES ON WELL SCREEN(S)
C                  NDSC IS TO BE GREATER OR EQUAL TO 2
C      NSCREN   =  NUMBER OF SCREENED INTERVAL
C      IREGL(I)=   NUMBER OF REPEATED REGULAR BLOCKS WITH THE SAME NUMBER OF NODES
C        -1        ON THE LEFT AND RIGHT VERTICAL LINES ACROSS LAYER I
C                  SUGGESTED VALUE IS IREG=2
C      NMINL(I)=   MINIMUM NUMBER OF NODES ALONG A VERTICAL LINE ACROSS LAYER I
C                  TO MINIMISE THE TOTAL NUMBER OF NODES,SUGGEST NMIN=2
C      NFRL(I) =   NUMBER OF NODES ON PORTION OF WELL BOUNDARY PENETRATING AQUIFER
C                  OR AQUITARD LAYER I
C      XSCR(I) =   Z-COORDINATE OF BASE OF SCREEN I ABOVE DATUM
C      HSCR(I) =   LENGTH OF SCREENED INTERVAL I
C-----------------------------
C      LIST OF OUTPUT VARIABLES
C-----------------------------
C      NNODE    =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C      NELEM    =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C      IT       =  TIME STEP NUMBER
C      TMM      =  REAL TIME VALUE AT THE MID-POINT OF TIME STEP IT
C      TM       =  REAL TIME VALUE AT THE END OF TIME STEP IT
C      X        =  RADIAL AND VERTICAL NODAL COORDINATES
C      H        =  NODAL HEAD OR DRAWDOWN VALUES
C      TLESS    =  NODAL VALUES OF DIMENSIONLESS TIME,1/U
C      HLESS    =  NODAL VALUES OF WELL FUNCTION FOR TRANSIENT FLOW,W(U)
C      IQ       =  DISCHARGE ITERATION NUMBER
C      SW       =  DRAWDOWN VALUE AT CURRENT TIME
C      QAQFR    =  DISCHARGE FROM AQUIFER INTO WELL AT DRAWDOWN SW
C      QSTRGE   =  DISCHARGE FROM WELL STORAGE
C      QCALL    =  TOTAL CALCULATED DISCHARGE
C                  QCALL = DISCHARGE FROM WELL STORAGE + QAQFR
C      QRDIF    =  RESIDUAL DISCHARGE
C                  QRDIF = THE ABSOLUTE DIFFERENCE BETWEEN PRESCRIBED AND
C                  DISCHARGE AND CALCULATED DISCHARGE
C      NOD      =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C      VEL      =  ABSOLUTE ELEMENT VELOCITIES
C      VCOMP1   =  RADIAL COMPONENT OF ELEMENT VELOCITY
C      VCOMP2   =  VERTICAL COMPONENT OF ELEMENT VELOCITY
C      IDARCY   =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                  OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                  IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
      DIMENSION JDARCY(500)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      COMMON /ACORE/VCORE(2000)
      COMMON /ALAYR/AKL(3),BKL(3),VCRL(3),SSL(3),THL(3),IREGL(3),
     1NMINL(3),NFRL(3),PML(3)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
```

```
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /BELEM/E(6,6,3),F(6,6,3),DSHFZ(2,6)
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /BQCAL/VS(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DOUBLE PRECISION DEXP,ARGU
      DATA NREAD,NPRINT/1,3/
C
C     **************
C     *            *
C     *  BLOCK 1   *
C     *            *
C     **************
C
C     PRINT INITIAL HEADINGS
C
      WRITE(NPRINT,1773)
 1773 FORMAT('1')
      WRITE(NPRINT,1003)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1023)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1033)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1003)
  993 FORMAT('1',4X,51H ************************************************
     1**)
 1003 FORMAT(5X,51H **************************************************)
 1013 FORMAT(5X,51H *                                                *)
 1023 FORMAT(5X,51H *    FINITE ELEMENT ANALYSIS OF WELL PROBLEMS    *)
 1033 FORMAT(6X,'* TRANSIENT,TWO REGIME FLOW TOWARD A SINGLE WELL *')
      READ(NREAD,1011)NPROB,IVEL,IDISCR,ORELAX                          $1
 1011 FORMAT(3I10,F10.2)
      DO 4800 JPRO=1,NPROB
      WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1              I6,12X,'*',/,20X,50('*'))
C
C     READ AND PRINT GENERAL DATA.
C
      READ(NREAD,2001)RW,RO,HO,HTOL,NLAYR                               $1
 2001 FORMAT(4F10.2,I10)
      READ(NREAD,201)QFIX,RCSNG,QRTOL                                   $1
  201 FORMAT(3F10.2)
      READ(NREAD,111)IGP,IBOUND,IWBC,IKMAX,IWAT                         $1
  111 FORMAT(5I10)
      IAQTA=IWAT
      WRITE(NPRINT,23)RW,RO,HO,RCSNG
      WRITE(NPRINT,2193)QFIX,QRTOL,ORELAX,HTOL
      WRITE(NPRINT,193)IGP,IBOUND,IWBC,IKMAX
      QRTOL=QRTOL*QFIX/2.0
   23 FORMAT(///,20X,'GENERAL DATA',///,
     1              10X,'RADIUS OF WELL    =',F10.2,//,
     2              10X,'RADIUS OF INFLUENCE    =',F10.2,//,
     3              10X,'HEIGHT OF WATER TABLE    =',F10.2,//,
     4              10X,'RADIUS OF WELL CASING    =',F10.2,//)
 2193 FORMAT(10X,'DISCHARGE INTO WELL    =',F10.2,//,
     1              10X,'DISCHARGE TOLERANCE RATIO    =',F12.4,//,
     2              10X,'OVER RELAXATION FACTOR    =',F12.4,//,
     3              10X,'HEAD TOLERANCE            =',F12.4,//)
  193 FORMAT(10X,'GRAVEL PACK INDEX    =',I5,//,
     1              10X,'BOUNDARY INDEX    =',I5,//,
     2              10X,'WELL B.C. INDEX    =',I5,//,
     3              10X,'LAYER OF MAX PERMEABILITY    =',I5)
C
C     READ AND PRINT MATERIAL DATA
C
      WRITE(NPRINT,6013)
 6013 FORMAT(///,10X,'FORMATION PROPERTIES',//,
     1              10X,'LAYER NO.',5X,'THICKNESS',5X,'COEFF-A',
     2  5X,'COEF-B',5X,'CRIT. VELOCITY',5X,'SPECIFIC STORAGE',//)
      DO 17 I=1,NLAYR                                                   *NLAYR**
      READ(NREAD,3013)THL(I),AKL(I),BKL(I),VCRL(I),SSL(I)               $1
 3013 FORMAT(5E10.2)
      GCR=AKL(I)*VCRL(I)+BKL(I)*VCRL(I)**2
      PML(I)=1./AKL(I)
      IF(GCR.GT.0.) PML(I)=VCRL(I)/GCR
      WRITE(NPRINT,7013)I,THL(I),AKL(I),BKL(I),VCRL(I),SSL(I)
 7013 FORMAT(10X,I5,9X,E9.2,3(5X,E9.2),9X,E9.2)
      SSL(I)=SSL(I)*1000.
   17 CONTINUE                                                          /*
      RGP=RW
      IF(IGP.EQ.0) GO TO 29
```

```
      READ(NREAD,331)AGP,BGP,VGP,THGP,BTGP                               $1,SK
  331 FORMAT(5F10.3)
      RGP=THGP+RW
      GRGP=AGP*VGP+BGP*VGP**2
      PMGP=1./AGP
      IF(GRGP.NE.0) PMGP=VGP/GRGP
      WRITE(NPRINT,153)AGP,BGP,VGP,PMGP
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1           10X,'COEFFICIENT A =',F10.4//
     2           10X,'COEFFICIENT B = ',F10.4//
     3           10X,'CRITICAL VELOCITY = ',F10.4//
     4           10X,'COEFFICIENT K = ',F10.4)
      WRITE(NPRINT,163)THGP,RGP
  163 FORMAT(10X,'THICKNESS OF PACK = ',F10.2//
     1           10X,'RADIUS OF PACK = ',F10.2)
   29 CONTINUE
C
C     FOR WATER TABLE AQUITARD READ DELAYED YIELD DATA.
C
      IF(IAQTA.EQ.0) GO TO 117
      READ(NREAD,173)SY,DINDEX                                           $1,SK
  173 FORMAT(2E10.2)
      SFAC=DINDEX*SY/6.
      DINDEX=DINDEX/1000.
  117 CONTINUE
C
C     READ AND PRINT DISCRETIZATION DATA.
C
      READ(NREAD,121)NTICR,ITST,TFACTR,TMUL,DTMUL                        $1
  121 FORMAT(2I10,3F10.2)
      NNODS=1
C
C     GENERATE ELEMENTS OF VECTOR TIME.
C
      TH=THL(IKMAX)
      PM=PML(IKMAX)
      SS=SSL(IKMAX)
      BK=BKL(IKMAX)
      BKMAX=BK
      TPAT=1
      NTICP=NTICR+1
      CALL TIGEN(NTICP,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX,TPAT)
C
C     READ AND PRINT DISCRETIZATION DATA.
C
      READ(NREAD,901)FRLEN,SCFAC,XLMAX                                   $1
  901 FORMAT(3F10.2)
      READ(NREAD,801)NDSC,NSCREN                                         $1
  801 FORMAT(2I10)
      IPENTR=0
      DO 602 I=1,NLAYR                                                   *NLAYR**
      READ(NREAD,1901)IREGL(I),NMINL(I),NFRL(I)                          $1
 1901 FORMAT(3I10)
  602 CONTINUE                                                           /*
C
C     READ WELL SCREEN DATA
C
      WRITE(NPRINT,553)
  553 FORMAT(///,5X,'SCREEN NO.',10X,'BASE HEIGHT',14X,'LENGTH',//)
      SCLEN=0.0
      DO 702 I=1,NSCREN                                                  *NSCREN*
      READ(NREAD,601)XSCR(I),HSCR(I)                                     $1
  601 FORMAT(2F10.2)
      SCLEN=SCLEN+HSCR(I)
      WRITE(NPRINT,563)I,XSCR(I),HSCR(I)
  563 FORMAT(7X,I6,13X,F11.2,8X,F10.2)
  702 CONTINUE                                                           /*
C
C     GENERATE AND PRINT DISCRETIZATION DATA.
C
      CALL GXNODR(FRLEN,SCFAC,XLMAX,RW,RO,NNODE,NELEM,LVEC,NLAYR,IPENTR,
     1  IDISCR)
C
C     MODIFY IPROP.
C
      DO 1544 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      XCEN=(X(J1,1)+X(J2,1))*0.5
      IF(XCEN.GT.RGP) GO TO 1545
 1544 CONTINUE
 1545 CONTINUE
      IF(IAQTA.EQ.0) GO TO 166
      CALL VDFB(LVEC)
      DO 176 I=1,NNODE
      HINT(I,1)=0.0
  176 CONTINUE
  166 CONTINUE
C
C     FIND BANDWIDTHS ETC.
```

```
C
      CALL EBFINR(LEN,NELEM,NN,NNODE)
      WRITE(NPRINT,233)LEN
  233 FORMAT(//////,10X,25HLENGTH OF GROSS VECTOR = ,I8,//)
      DO 130 I=1,LEN
      VD(I)=0.0
  130 CONTINUE
      NNP=0
      NT=0
      DO 135 IE=1,NELEM
      NN=ITYPE(IE)
      IF(IPROP(IE).NE.0) L=IPROP(IE)
      SC=SSL(L)
      IF(NN.EQ.3) GO TO 235
      CALL ELGNDR(NT,SC,IE,DET,NNP)
      GO TO 335
  235 CONTINUE
      CALL ELGND3(NT,SC,IE)
  335 CONTINUE
      CALL MERBD(NN,IE)
  135 CONTINUE
C
C     CALCULATE HEAD AT VARIOUS TIME VALUES
C
      TMIS=TH*PM
      FPI=88./7.
      CONST=QFIX/(FPI*TMIS)
      DIFFUS=PM/SS
      LVST=2
      NST=NDVEC(1)
      LSTEL=NDVEC(1)
      NELTO=NELEM
      DO 642 I=1,LVEC
      HCOLD(I)=0.
      HF(I)=0.
      FINT(I,1)=0.
  642 QLEAK(I)=0.
      IF(ITST.EQ.1) GO TO 1245
      CALL HREAD(NNODS,1,1)
      CALL HPUNCH(NNODS,3,1)
      CALL HREAD(NNODS,1,2)
      CALL HPUNCH(LVEC,3,2)
      IST=NNODS+1
      DO 1345 I=IST,NNODE
      H(I)=HO
 1345 CONTINUE
      GO TO 1445
 1245 CONTINUE
      DO 3545 I=1,NNODE
      H(I)=HO
 3545 CONTINUE
 1445 CONTINUE
      DO 245 I=1,NNODE
      HINT(I,1)=H(I)
  245 CONTINUE
      DELT=TIME(1)
      NDTO=NNODE
      ITMIN=NTICR
      QAQFR=2.*QRTOL
      SWOLD=0.0
C
C     ESTIMATE WELL DRAWDOWN FOR FIRST TIME STEP.
C
      CALL DSWINT(DSOLD,BK,QFIX,TMIS,TH,RW,SCLEN,DIFFUS,CUNST)
C
      DSOLD=0.6*DSOLD
      IST=1
      FQ=0.0
      NQITER=5
      IF(IWBC.EQ.0) RCSNG = 0.
C     SET JDARCY.
C
      DO 299 I=1,NELEM
      JDARCY(I)=0
  299 CONTINUE
C
C     LOOPING WITH LOOP PARAMETER IT=1,NTICR
C
      DO 7007 IT=ITST,NTICR
      ITCUR=IT
      IF(IT.GT.1) DELT=TIME(IT)-TIME(IT-1)
      TM=TIME(IT)
      LVST=LVEC
      NST=NNODE
      TMM=TM-DELT*0.5
      WRITE(NPRINT,683)IT
  683 FORMAT(///,10X,35('*'),/,10X,'*',4X,
     1           'TIME STEP NUMBER  =',I5,5X,'*',/,10X,35('*'))
      TMIL=TMM/1000.
      WRITE(NPRINT,333)TMIL
```

```
  333 FORMAT(///,10X,41('*'),/,10X,'*',9X,'TIME  =',
     1              E14.3,9X,'*',/,10X,41('*'),//)
      WRITE(NPRINT,343)RO,NNODE,NELEM,LVST
  343 FORMAT(/,10X,'ESTIMATED RADIUS OF INFLUENCE  =',F10.2,
     1//,10X,'CORRESPONDING NO. OF NODES   =',I5,//,
     210X,'CORRESPONDING NO. OF ELEMENTS   =',I5,//,
     310X,'CORRESPONDING COMPONENT OF VECTOR NDVEC   =',I5,//)
C
C     COMPUTE DIMENSIONLESS TIME.
C
      CONS=4.0*DIFFUS*TMM
      DO 709 I=1,NNODE
      TLESS(I)=CONS/X(I,1)**2
  709 CONTINUE
C
C     FOR WATER TABLE AQUITARD,COMPUTE BOULTON'S INCREMENTAL DELAYED YEILD.
C
      IF(IAQTA.EQ.0) GO TO 777
      ARGU=-DINDEX*DELT*0.5
      TERM=DEXP(ARGU)
      ES=SY*(1.0-TERM)
      WRITE(NPRINT,903) ES
      ES=ES*1000.
  903 FORMAT(///,'EFFECTIVE SPECIFIC YIELD   =',E14.4)
  777 CONTINUE
C
C     INITIALISE HEAD VALUES.
C
      IF(IT.EQ.1) GO TO 165
  677 CONTINUE
      DO 155 I=1,NNODE
      HINT(I,1)=H(I)
  155 CONTINUE
  165 CONTINUE
      CALL EBFINR(LEN,NELEM,NN,NNODE)
      DO 2722 I=LVST,LVEC
      IF(NST.GT.NDVEC(I)) GO TO 2722
      JJ=NDVEC(I-1)
      JF=I
      GO TO 2822
 2722 CONTINUE
 2822 CONTINUE
      LMB=NDVEC(JF)-JJ
C
      CALL VECMUL(NNODE,LEN)
C
C     FOR WATER TABLE AQUITARD,ADD BOULTON'S DELAYED YIELD.
C
      IF(IAQTA.EQ.0) GO TO 278
      CALL GVMOD(LVST,ES,1)
      IF(IT.EQ.1) GO TO 278
      FTERM=0.5*DELT*SFAC
      CALL GKMOD(LVST,FTERM)
  278 CONTINUE
      HW=HO
      NDW=NDVEC(1)
      NDRO=LMB
      CALL BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,NSCREN,RW)
      IF(IBOUND.EQ.0) JBD(NBW+1)=0
C
C     STORE ELEMENTS OF MATRIX GK FOR DISCHARGE COMPUTATION.
C
      DO 1378 I=1,NBW
      JJ=JBD(I)
 1378 GP(I)=GK(JJ,1)
C
C     SET UP LOOP FOR WELL DISCHARGE ITERATION.
C
      DO 998 IQ=1,NQITER
      DO 1559 I=1,NNODE
      H(I)=HINT(I,1)
 1559 CONTINUE
C
C     SET IDARCY
C
      DO 99 I=1,NELEM
      IDARCY(I)=JDARCY(I)
   99 CONTINUE
C
C     ADJUST VALUE OF WELL DRAWDOWN.
C
      WRITE(NPRINT,5003)
 5003 FORMAT(/,5X,30('- '))
      CALL FQSET(IT,ITCUR,DELT,FQ,TPAT)
      CALL SWMOD(IQ,IT,RCSNG,DELT,QFIX,QAQFR,SWOLD,DSOLD,TPAT,FQ)
      WRITE(NPRINT,833)IQ,DSW(IQ),SW(IT)
  833 FORMAT(///,10X,'DISCHARGE ITERATION NUMBER   =',I10,//,
     1              10X,'DRAWDOWN INCREMENT    =',F10.3,//,
     2            10X,'DRAWDOWN VALUE    =',F10.3)
C
```

```
C      SET UP LOOP FOR ITERATION ON HEAD
C
       DO 1177 I=1,NBW
       DISP(I)=SH(IT)+HO
 1177  CONTINUE
       NITER=5
       IF(BKMAX.LE.0.0) NITER=1
       VCOUNT=0
       DO 999 III=1,NITER
C
C      INITIALISE VECTOR CK.
C
       DO 305 I=1,NNODE
       CK(I,1)=GK(I,1)
  305  CONTINUE
C
C      FORM VECTOR VCORE.
C
       DO 300 I=1,LEN
       VCORE(I)=0.0
  300  CONTINUE
       NT=0
       NDIFF=2
       CALL INFLUC(NN,NDIFF)
       DO 350 I=1,NELEM
       NN=ITYPE(I)
       IF(IPROP(I).EQ.0) GO TO 755
       L=IPROP(I)
       AKK=AKL(L)
       BKK=BKL(L)
       PKK=PML(L)
       GO TO 365
  755  CONTINUE
       AKK=AGP
       BKK=BGP
       PKK=PMGP
  365  CONTINUE
       IF(NN.EQ.3) GO TO 465
       CALL ELGNCR(I,III,AKK,BKK,NT,PKK)
       GO TO 565
  465  CONTINUE
       NT=NREP(I)
       CALL ELGNCT(I,III,AKK,BKK,NT,VCOUNT,PKK)
  565  CONTINUE
       CALL MERB3(NN,I)
  350  CONTINUE
C
C      INITIALISE VECTOR VK.
C
       DO 978 I=1,LEN
       VK(I)=0.0
  978  CONTINUE
       IF(IAQTA.EQ.0) GO TO 555
       CALL GVMOD(LVST,ES,0)
  555  CONTINUE
       DO 530 I=1,LEN
       VK(I)=VCORE(I)*DELT*0.5+VD(I)+VK(I)
  530  CONTINUE
       NLEN=0
       JJ=JBD(NBW)
       DO 98 I=1,JJ
       NLEN=NLEN+NBAND(I)
   98  CONTINUE
       DO 378 I=1,NLEN
  378  VS(I)=VK(I)
C
C      SOLVE EQNS BY BANDED ELIMINATION SCHEME
C
       NLL=1
       CALL SYMSOL(NNODE,NLL)
C
       IF(III.EQ.1) GO TO 900
  415  CONTINUE
       NCOUNT=0
       EMAX=0.0
       DO 450 I=1,NNODE
       EPSI=CK(I,1)-H(I)
       IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
       IF(ABS(EPSI).LE.HTOL)  GO TO 460
       NCOUNT=NCOUNT+1
       H(I)=H(I)+ORELAX*EPSI
       GO TO 450
  460  CONTINUE
       H(I)=CK(I,1)
  450  CONTINUE
       IF(NCOUNT.EQ.0) GO TO 1000
       GO TO 1999
  900  CONTINUE
       DO 950 I=1,NNODE
       H(I)=CK(I,1)
```

```
  950 CONTINUE
 1999 CONTINUE
      IF(NITER.EQ.1) GO TO 999
      DO  199 I=1,NELEM
      NT=NREP(I)
      NN=ITYPE(I)
      IF(IPROP(I).EQ.0) L=1
      IF(IPROP(I).NE.0) L=IPROP(I)
      AK=AKL(L)
      BK=BKL(L)
      PMK=PML(L)
      VCR=VCRL(L)
      CALL SETARG(AGP,BGP,PMGP,VGP,AK,BK,PMK,VCR,I)
      IF(NN.EQ.3) GO TO 399
      CALL VCHECR(I,AK,BK,NT,PMK,VCR,HRRX,HRRY,NN)
      GO TO 499
  399 CONTINUE
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      CALL VCHEC3(I,AK,BK,J1,J2,J3,NT,PMK,VCR,HRRX,HRRY)
  499 CONTINUE
  199 CONTINUE
  999 CONTINUE
 1000 CONTINUE
      WRITE(NPRINT,413) III
  413 FORMAT(///,10X,'NUMBER OF ITERATIONS REQUIRED  =',I5)
      WRITE(NPRINT,473)NCOUNT,EMAX
  473 FORMAT(//,10X,'TOLERANCE COUNTER FOR HEAD   =',I3,//,
     1            10X,'ABSOLUTE MAXIMUM ERROR IN HEAD   =',F12.4)
      DO 470 I=1,NNODE
      H(I)=CK(I,1)
      IF((ITCUR.GT.IT).AND.(IQ.EQ.1)) TLESS(I)=TLESS(I)*TM/TMM
      HLESS(I)=(HO-H(I))/CUNST
  470 CONTINUE
C
C     CHECK FOR ACCURACY OF DISCHARGE RATIO.
C
      DWD=DSW(IQ)
      CALL AQDIS(NBW,QAQFR,QRDIF,DELT,TH,RCSNG,IQ,QFIX,DWD)
C
C
C     PRINT FINAL DISCHARGE VALUES.
C
      QCALL=QCALC(IQ)
      QSTRGE=QCALL-QAQFR
      WRITE(NPRINT,1203)QAQFR,QSTRGE,QCALL,QRDIF
 1203 FORMAT(///,10X,'DISCHARGE FROM AQUIFER INTO WELL    =',F12.4,//,
     1          10X,'DISCHARGE FROMM WELL STORAGE   =',F12.3,//,
     1           10X,'TOTAL CALCULATED DISCHARGE    =',F14.3,//,
     210X,'RESIDUAL DISCHARGE              =',F10.4)
      IF(QRDIF.LE.QRTOL) GO TO 1102
  998 CONTINUE
 1102 CONTINUE
C
C     RESET JDARCY.
C
      DO 3359 I=1,NELEM
      JDARCY(I)=IDARCY(I)
 3359 CONTINUE
      SWOLD=SW(IT)
      CALL ROUT(NNODE)
C
C      COMPUTE AND PRINT ELEMENT VELOCITIES
C
      IF(IVEL.EQ.0) GO TO 557
      WRITE(NPRINT,4203)
 4203 FORMAT(///,20X,'*****************************************'/
     1               20X,'*            ELEMENT VELOCITIES          *'/
     2               20X,'*****************************************'///
     310X,'ELEM NO.',10X,'RADIAL VEL',10X,'VERTIC VEL',10X,'IDARCY',///)
      DO 3000 I=1,NELEM
      NT=NREP(I)
      NN=ITYPE(I)
      IF(IPROP(I).EQ.0) L=1
      IF(IPROP(I).NE.0) L=IPROP(I)
      AK=AKL(L)
      BK=BKL(L)
      PMK=PML(L)
      VCR=VCRL(L)
      CALL SETARG(AGP,BGP,PMGP,VGP,AK,BK,PMK,VCR,I)
      IF(NN.EQ.3) GO TO 3999
      CALL VCHECR(I,AK,BK,NT,PMK,VCR,HRRX,HRRY,NN)
      GO TO 4999
 3999 CONTINUE
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      CALL VCHEC3(I,AK,BK,J1,J2,J3,NT,PMK,VCR,HRRX,HRRY)
 4999 CONTINUE
```

```
      VCOMP1=-VEL(I)*HRRX
      VCOMP2=-VEL(I)*HRRY
      WRITE(NPRINT,1103)I,VCOMP1,VCOMP2,IDARCY(I)
 1103 FORMAT(10X,I5,2(10X,F10.3),10X,I5)
 3000 CONTINUE
  557 CONTINUE
      IF(IT.GT.ITMIN) GO TO 577
C
C     TEMPORARY BLOCK.
C
      IF(ITCUR.GT.IT) GO TO 7000
      TMIL=TM/1000.
      WRITE(NPRINT,333)TMIL
      ITCUR=IT+1
      GO TO 677
  577 CONTINUE
C
C     OBTAIN HEAD VALUES AT THE END OF TIME INTERVAL BY EXTRAPOLATION.
C
      TMIL=TM/1000.
      WRITE(NPRINT,333)TMIL
      DO 477 I=1,NNODE
      TLESS(I)=TLESS(I)*TM/TMM
      H(I)=2.0*H(I)-HINT(I,1)
      HLESS(I)=(HO-H(I))/CONST
  477 CONTINUE
      SWTEMP=SWOLD
      SWOLD=H(1)-HO
      QFR=QAQFR
      DWTEMP=SWOLD-SWTEMP
      QAQFR=QFR*SWOLD/SWTEMP
      ACSNG=22.*RCSNG**2/7.
      TRM=ABS(ACSNG*DWTEMP*2./DELT)
      QCALL=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALL)
      SW(IT)=SWOLD
      QSTRGE=QWSTR
      WRITE(NPRINT,1203)QAQFR,QSTRGE,QCALL,QRDIF
 7009 CONTINUE
      CALL ROUT(NNODE)
 7000 CONTINUE
C
C     FOR WATER TABLE AQUITARD,COMPUTE FIRST PORTION OF BOULTON'S INTEGRAL.
C
      IF(IAQTA.EQ.0) GO TO 6007
      DO 9007 I=1,LVST
      L=NDVEC(I)
      HF(I)=H(L)
      FINT(I,1)=HINT(L,1)
 9007 CONTINUE
      CALL BSIMP(LVST,TH,DINDEX,SY,DELT,IT,0)
      DO 8007 I=1,LVST
      L=NDVEC(I)
      HCOLD(I)=H(L)
 8007 CONTINUE
 6007 CONTINUE
 7007 CONTINUE
C
C     PUNCH OUT SOLUTION AT FINAL TIME.
C
      CALL HPUNCH(NNODE,2,1)
      CALL HPUNCH(LVEC,2,2)
 5000 CONTINUE
 4800 CONTINUE
      STOP
      END
      SUBROUTINE GXNODR(FRLEN,SCFAC,XLMAX,RW,RO,NNODE,NELEM,LVEC,NLAYR,
     1IPENTR,IDISCR)
C
C     GENERATES DISCRETIZATION DATA.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      COMMON /ALAYR/AKL(3),BKL(3),VCRL(3),SSL(3),THL(3),IREGL(3),
     1NMINL(3),NFRL(3),PML(3)
      COMMON /AELEM/NTRAN(4,100),NH1(4,100),NH2(4,100),NDB1(4,100),NDB2(
     14,100),NDWB(22)
      DATA NPRINT/3/
      CALL DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
      NRST=1
      THTO=0.
      NSTFR=1
      NDWB(1)=0
      DO 10 I=1,NLAYR
      THTO=THTO+THL(I)
      NFR=NFRL(I)
      NDWB(1)=NDWB(1)+NFR
      IREG=IREGL(I)
      NMIN=NMINL(I)
```

```
      CALL NCRGR1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,I)
   10 CONTINUE
      CALL NCRGR2(NRST,NRR,NLAYR,NSTFR)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,33)NLAYR,NRR
   33 FORMAT('1',20X,'DISCRETISATION DATA',//,
     1          10X,'NUMBER OF LAYERS    =',I5,//,
     2          10X,'NUMBER OF SUBREGIONS   =',I5)
   94 CONTINUE
C
C     DISCRETISE ENTIRE REGION INTO RECTANGULAR ELEMENTS.
C
      KCREP=1
      NSTOR=1
      NTSEL=1
      DO 65 I=NRST,NRR
      DO 691 L=1,NLAYR
      N1=NH1(L,I)
      N2=NH2(L,I)
      NFND1=NDB1(L,I)
      NFND2=NDB2(L,I)
      NPAT=NTRAN(L,I)
      IF(NPAT.NE.0) GO TO 791
      CALL RBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
      GO TO 891
  791 CONTINUE
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
  891 CONTINUE
      NELEM=NTSEL-1
      DO 68 IE=NSTOR,NELEM
      IPROP(IE)=L
   68 CONTINUE
      NSTOR=NTSEL
  691 CONTINUE
   65 CONTINUE
C
C     PRINT OUT ELEMENT DATA.
C
      NELEM=NTSEL-1
      IF(IDISCR.EQ.0) GO TO 104
      WRITE(NPRINT,83)
   83 FORMAT(///,10X,'IDENTIFICATION OF ELEMENT-NODE CONNECTIONS',///,
     112X,'ELEM NO',5X,'ITYPE',5X,'NREP',5X,'IPROP',5X,'NODE1',5X,
     1'NODE2',5X,'NODE3',5X,'NODE4',//)
      DO 80 I=1,NELEM
      NN=ITYPE(I)
      WRITE(NPRINT,93)I,ITYPE(I),NREP(I),IPROP(I),(NOD(I,K),K=1,NN)
   93 FORMAT(12X,9(I5,5X))
   80 CONTINUE
  104 CONTINUE
C
C     GNERATE NODE COORDINATES.
C
      RI=RW
      NS=1
      MM=1
      DO 75 I=NRST,NRR
      DXG=XLEN(I)
      ZI=0.0
      DO 87 L=1,NLAYR
      TH=THL(L)
      NN=NH1(L,I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH
      NS=NS-1
      RI=RI-DXG
   87 CONTINUE
      RI=RI+DXG
      NS=NS+1
   75 CONTINUE
      ZI=0.0
      DO 877 L=1,NLAYR
      TH=THL(L)
      NN=NH2(L,NRR)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH
      NS=NS-1
      RI=RI-DXG
  877 CONTINUE
C
C
      NNODE = NS
      IF(IDISCR.EQ.0) GO TO 114
      WRITE(NPRINT,103)
  103 FORMAT(///,10X,'NODAL COORDINATES',//,10X,'NODE',5X,
     1'R-COORD',5X,'Z-COORD',//)
C
      DO 85 I=1,NNODE
      WRITE(NPRINT,113)I,X(I,1),X(I,2)
  113 FORMAT(10X,I4,2F11.2)
```

```
   85 CONTINUE
  114 CONTINUE
C
C      GENERATE TOP BOUNDARY COORDINATE VECTORS.
C
       NDTO=NS
       CALL TOPVEC(THTO,LVEC,NDTO)
C
C      PRINT OUT BOUNDARY COORDINATE VECTORS.
       WRITE(NPRINT,223)
  223 FORMAT(///,20X,'TOP BOUNDARY NODES AND RADIAL COORDINATES',///,
     110X,'NODE NUMBER',20X,'R-COORDINATE',//)
       DO 335 I=1,LVEC
       WRITE(NPRINT,333)NDVEC(I),RVEC(I)
  335 CONTINUE
  333 FORMAT(10X,I7,25X,F10.2)
       RETURN
       END
       SUBROUTINE NCRGR1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,L)
C
C      GENERATES DISCRETIZATION PARAMETERS:- NTRAN,NH1,NH2,NDB1,NDB2
C
       COMMON /AELEM/NTRAN(4,100),NH1(4,100),NH2(4,100),NDB1(4,100),
     1NDB2(4,100),NDWB(22)
       NC=NFR
       NCOUNT=0
       DO 10 I=NRST,NRR
       IRGG=IREG
       IF((I.EQ.NRST).AND.(IPENTR.NE.0)) IRGG=0
       NCOUNT=NCOUNT+1
       IF(NCOUNT.LT.IRGG) GO TO 20
       NCOUNT=0
       NHALF=NC/2
       NREM=NC-2*NHALF
       IF(NREM.GT.0) GO TO 15
       NC1=NC-1
       IF(NC1.LT.NMIN) GO TO 20
       NTRAN(L,I)=2
       NC=NC1
       NH2(L,I)=NC
       NH1(L,I)=NC+1
       GO TO 10
   15 CONTINUE
       NC1=NHALF+1
       IF(NC1.LT.NMIN) GO TO 20
       NTRAN(L,I)=1
       NC=NC1
       NH2(L,I)=NC
       NH1(L,I)=2*NH2(L,I)-1
       GO TO 10
   20 CONTINUE
       NH2(L,I)=NC
       NTRAN(L,I)=0
       NH1(L,I)=NH2(L,I)
   10 CONTINUE
       RETURN
       END
       SUBROUTINE NCRGR2(NRST,NRR,NLAYR,NSTFR)
C
C      MODIFIES THE VALUES OF NDB1 AND NDB2.
C
       COMMON /AELEM/NTRAN(4,100),NH1(4,100),NH2(4,100),NDB1(4,100),
     1NDB2(4,100),NDWB(22)
       NDBC=NSTFR
       DO 50 J=NRST,NRR
       NDB1(1,J)=NDBC
       IF(NLAYR.LT.2) GO TO 56
       DO 55 L=2,NLAYR
       NDB1(L,J)=NDB1(L-1,J)+NH1(L-1,J)-1
   55 CONTINUE
   56 CONTINUE
       L=NLAYR
       NDBC=NDB1(L,J)+NH1(L,J)
   50 CONTINUE
       NR1=NRR-1
       DO 60 J=NRST,NR1
       DO 65 L=1,NLAYR
       NDB2(L,J)=NDB1(L,J+1)
   65 CONTINUE
   60 CONTINUE
       NDB2(1,NRR)=NDBC
       DO 75 L=2,NLAYR
       NDB2(L,NRR)=NDB2(L-1,NRR)+NH2(L-1,NRR)-1
   75 CONTINUE
       RETURN
       END
       SUBROUTINE DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
C
C      GENERATES DISCRETIZATION PARAMETERS:- NRR,XLEN
C
```

```
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      MAXNR=89
      XLEN(1)=FRLEN
      RLEN=RO-RW
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      IF(XLEN(I).GT.XLMAX) XLEN(I)=XLMAX
      SUM=XLEN(I)+SUM
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I
      DENOM=1.+SCFAC+SCFAC**2
      XLEN(I-2)=(XREM+XLEN(I-1)+XLEN(I-2))/DENOM
      XLEN(I-1)=XLEN(I-2)*SCFAC
      XLEN(I)=XLEN(I-1)*SCFAC
      RETURN
      END
      SUBROUTINE RBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC RECTANGULAR BLOCKS.
C
      DIMENSION ND(1,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND2+1
      ND(1,4)=NFND1+1
      IF(NPAT.EQ.1) GO TO 200
      NTEMP=N1-2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=4
      DO 50 K=1,4
      NOD(J1,K)=ND(1,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
      GO TO 400
  200 CONTINUE
      ND(1,5)=ND(1,4)
      ND(1,4)=ND(1,5)+1
      N2=N1/2+1
      NTEMP=N2-2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=5
      NI=J1-NST
      NOD(J1,1)=ND(1,1)+2*NI
      NOD(J1,4)=ND(1,4)+2*NI
      NOD(J1,5)=ND(1,5)+2*NI
      NOD(J1,2)=ND(1,2)+NI
      NOD(J1,3)=ND(1,3)+NI
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  400 CONTINUE
      RETURN
      END
      SUBROUTINE COORDC(NS,MM,NN,DXG,TH,RI,ZI)
C
C     GENERATES NODAL COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /ASOLV/XX(5,90),XY(5,90)
      DY=TH/(NN-1)
      DO 20 L=1,NN
   20 XX(1,L)=RI
      DO 50 K=1,MM
   50 XY(K,1)=ZI
      DO 60 J=2,NN
      JM1=J-1
      DO 70 K=1,MM
      XY(K,J)=XY(K,JM1)+DY
   70 CONTINUE
   60 CONTINUE
      DO 80 J=1,MM
      DO 90 K=1,NN
      L=NN*(J-1)+K+NS-1
      X(L,1)=XX(J,K)
      X(L,2)=XY(J,K)
   90 CONTINUE
   80 CONTINUE
```

```
      NEN=NS+(MM*NN)-1
      NS=NEN+1
      RI=X(NEN,1)+DXG
      RETURN
      END
      SUBROUTINE TOPVEC(TH,LVEC,NDTO)
C
C     COMPUTES RADIAL COORDINATES FOR NODES ALONG THE TOP BOUNDARY.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(100)
      J=0
      DO 10 I=1,NDTO
      ZDIF=ABS(TH-X(I,2))
      IF(ZDIF.GT.0.001) GO TO 10
      J=J+1
      NDVEC(J)=I
      RVEC(J)=X(I,1)
   10 CONTINUE
      LVEC=J
      RETURN
      END
      SUBROUTINE EBFINR(LEN,NM,NN,LL)
C
C     SUBROUTINE TO COMPUTE BANDWIDTHS OF THE BANDED SYMMETRIC GROSS
C     MATRIX. NBAND CONTAINS THE BANDWIDTHS, ID THE POSITION OF THE
C     TERM ON THE DIAGONALS OF THE ORIGINAL MATRIX, LEN IS THE LENGTH
C     THE VECTOR
C
      DIMENSION LV(6)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
C
      DO 20 I=1,LL
      NBAND(I)=1
      ISTART(I)=I
   20 CONTINUE
C
C     SCAN THROUGH THE LOCATION VECTOR FOR EACH MEMBER TO FIND
C     THE POSITION OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW
C
      DO 25 I=1,NM
      NN=ITYPE(I)
      NF2=NN
      DO 30 J=1,NF2
      LV(J)=NOD(I,J)
   30 CONTINUE
      DO 45 J=1,NF2
      IF(LV(J).EQ.0)GO TO 45
      DO 40 K=1,NF2
      IF(LV(J).GT.LV(K)) GO TO 40
      NW=LV(K)-LV(J)+1
      NR=LV(J)
      IF(NW.GT.NBAND(NR))  NBAND(NR)=NW
   40 CONTINUE
   45 CONTINUE
C      SEARCH FOR THE FURTHEREST OFF-LEFT TERM
C
      DO 55 J=1,NF2
      IF(LV(J).EQ.0) GO TO 55
      DO 65 K=1,NF2
      IF(LV(J).LT.LV(K)) GO TO 65
      NW=LV(K)
      NR=LV(J)
      IF(NW.LT.ISTART(NR)) ISTART(NR)=NW
   65 CONTINUE
   55 CONTINUE
   25 CONTINUE
C
C     SET UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN=NBAND(1)
      DO 50 I=2,LL
      IF(NBAND(I)-NBAND(I-1).LT.0)  NBAND(I)=NBAND(I-1)-1
      LEN=LEN+NBAND(I)
   50 CONTINUE
      ID(1)=1
      DO 60 I=2,LL
   60 ID(I)=ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE ELGNCR(M,III,AK,BK,NT,PMK)
C
C     GENERATES ELEMENT MATRIX SELK FOR RECTANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /BELEM/E(6,6,3),F(6,6,3),DSHFZ(2,6)
```

```
      NN=ITYPE(M)
      KK=NN-3
      IF(NREP(M).EQ.NT) GO TO 30
      NT=NREP(M)
      DO 10 J=1,NN
      DO 20 K=1,NN
      IF(J.GT.K) GO TO 20
      EK(J,K)=BA(NT)*E(J,K,KK)+F(J,K,KK)/BA(NT)
      EK(J,K)=EK(J,K)*ORX(M)
   20 CONTINUE
   10 CONTINUE
   30 CONTINUE
      IF(IDARCY(M).EQ.0) GO TO 70
  500 CONTINUE
      CALL CONSTR(M,AK,BK,ACONST,NT,NN)
      GO TO 80
   70 CONTINUE
      ACONST=PMK
   80 CONTINUE
      DO 40 J=1,NN
      DO 50 K=1,NN
      IF(J.GT.K) GO TO 60
      SELK(J,K)=EK(J,K)*ACONST
      GO TO 50
   60 SELK(J,K)=SELK(K,J)
   50 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE INFLUC(NN,NDIFF)
C
C     GENERATES FIRST SET OF INFLUENCE COEFFICIENTS FOR RECTANGULAR ELEMENTS.
C
      COMMON /BELEM/E(6,6,3),F(6,6,3),DSHFZ(2,6)
      DO 200 I=1,NDIFF
      NN=I+3
      J=I
      GO TO (10,30),J
C
C     4-NODES ELEM
C
   10 CONTINUE
      DO 150 K=1,4
      E(K,K,J)=1./3.
  150 F(K,K,J)=E(K,K,J)
      E(1,2,J)=-1./3.
      E(1,3,J)=-1./6.
      E(1,4,J) = 1./6.
      E(2,3,J)=1./6.
      E(2,4,J)=-1./6.
      E(3,4,J)=-1./3.
      F(1,2,J)=1./6.
      F(1,3,J)=-1./6.
      F(1,4,J)=-1./3.
      F(2,3,J)=-1./3.
      F(2,4,J)=-1./6.
      F(3,4,J)=1./6.
      GO TO 500
C
C     5 NODES LEFT INCLUDED
C
   30 CONTINUE
      E(1,1,J) =  2./15.
      E(1,2,J) =  -1./6.
      E(1,3,J) = 0.0
      E(1,4,J)=   -1./30.
      E(1,5,J) =  1./15.
      E(2,2,J) =  1./3.
      E(2,3,J) =  1./6.
      E(2,4,J) =  0.0
      E(2,5,J) =  -1./3.
      E(3,3,J) =  1./3.
      E(3,4,J) =  -1./6.
      E(3,5,J) = -1./3.
      E(4,4,J) = 2./15.
      E(4,5,J) = 1./15.
      E(5,5,J) =  8./15.
      F(1,1,J) =  7./9.
      F(1,2,J) =  1./6.
      F(1,3,J) =  -1./6.
      F(1,4,J)=   1./9.
      F(1,5,J) =  -8./9.
      F(2,2,J) =  1./3.
      F(2,3,J) =  -1./3.
      F(2,4,J) =  -1./6.
      F(2,5,J) =  0.0
      F(3,3,J) = 1./3.
      F(3,4,J) =  1./6.
      F(3,5,J) =  0.0
      F(4,4,J) =    7./9.
```

```
      F(4,5,J) =  -8./9.
      F(5,5,J) =  16./9.
  500 CONTINUE
  200 CONTINUE
      RETURN
      END
      SUBROUTINE INFLUD(NN)
C
C     GENERATES THE SECOND SET OF INFLUENCE COEFFICIENTS FOR RECTANGULAR
C     ELEMENTS.
C
      DIMENSION ZEN(6,2),ZE(1,2)
      COMMON /BELEM/E(6,6,3),G(6,6,3),DSHFZ(2,6)
      JP=1
      K=NN-3
      ZEN(1,1)=-1.
      ZEN(2,1)=1.
      ZEN(3,1)=1.
      ZEN(4,1)=-1.
      ZEN(1,2)=-1.
      ZEN(2,2)=-1.
      ZEN(3,2)=1.
      ZEN(4,2)=1.
      ZEN(5,1)=0.
      ZEN(5,2)=-1.
      ZE(1,1)=0.
      ZE(1,2)=0.
      IF(K.EQ.2) GO TO 50
C
C     4-NODE ELEMENT.
C
      DO 2 I=1,NN
      T1=1.+ZEN(I,1)*ZE(JP,1)
      T2=1.+ZEN(I,2)*ZE(JP,2)
      DSHFZ(1,I)=.25*T2*ZEN(I,1)
      DSHFZ(2,I)=.25*T1*ZEN(I,2)
      G(I,I,K)=4./9.
      IF(I.LE.3) G(I,I+1,K)=2./9.
      IF(I.LE.2) G(I,I+2,K)=1./9.
      IF(I.EQ.1) G(I,4,K)=2./9.
    2 CONTINUE
      GO TO 100
   50 CONTINUE
C
C
C     5-NODE ELEMENT WITH MIDDLE NODE ON SIDE 1-4
C
      Z1=ZE(JP,1)
      Z2=ZE(JP,2)
      P1=1.0+Z1
      S1=1.0-Z1
      P2=1.0+Z2
      S2=1.0-Z2
      DSHFZ(1,1)=0.25*Z2*S2
      DSHFZ(1,4)=-0.25*Z2*P2
      DSHFZ(2,1)=-0.25*S1*(S2-Z2)
      DSHFZ(2,4)=0.25*S1*(P2+Z2)
C
      DSHFZ(1,2)=0.25*S2
      DSHFZ(1,3)=0.25*P2
      DSHFZ(2,2)=-0.25*P1
      DSHFZ(2,3)=-DSHFZ(2,2)
C
      DSHFZ(1,5)=-0.5*S2*P2
      DSHFZ(2,5)=-Z2*S1
C
      G(1,1,K)=0.177777
      G(1,2,K)=0.111111
      G(1,3,K)=0.00000
      G(1,4,K)=-0.444444E-01
      G(1,5,K)=0.888886E-01
      G(2,2,K)=0.444444
      G(2,3,K)=0.222222
      G(2,4,K)=0.372529E-08
      G(2,5,K)=0.222222
      G(3,3,K)=0.444444
      G(3,4,K)=0.111111
      G(3,5,K)=0.222222
      G(4,4,K)=0.177777
      G(4,5,K)=0.888886E-01
      G(5,5,K)=0.711111
  100 CONTINUE
      RETURN
      END
      SUBROUTINE FLGNDR(NT,SC,M,DET,NNP)
C
C     GENERATES ELEMENT MATRIX D FOR RECTANGULAR ELEMENTS.
C
      DIMENSION TJ(2,2)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
```

```
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /BELEM/F(6,6,3),G(6,6,3),DSHFZ(2,6)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      J4=NOD(M,4)
      ORX(M)=(X(J1,1)+X(J2,1))*0.5
      IF(NREP(M).EQ.NT) GO TO 50
      NT=NREP(M)
      NN=ITYPE(M)
      K=NN-3
      BB=ABS(X(J4,2)-X(J1,2))
      AA=ABS(X(J3,1)-X(J1,1))
      BA(NT)=BB/AA
      DET= AA*BB/4.
      TJ(1,1)=2./AA
      TJ(2,2)=2./BB
      IF(NNP.EQ.NN) GO TO 15
      CALL INFLUD(NN)
   15 CONTINUE
      NNP=NN
      DO 10 I=1,NN
      SLX(NT,I)=TJ(1,1)*DSHFZ(1,I)
      SLY(NT,I)=TJ(2,2)*DSHFZ(2,I)
   10 CONTINUE
C
C     FORM ELEMENT MATRIX:-D
C
      DO 100 I=1,NN
      DO 100 J=1,NN
      IF(J-I) 105,110,110
  110 D(I,J)=ORX(M)*SC*DET*G(I,J,K)
      GO TO 100
  105 D(I,J)=D(J,I)
  100 CONTINUE
   50 CONTINUE
      RETURN
      END
      SUBROUTINE CONSTR(M,AK,BK,ACONST,NT,NN)
C
C     COMPUTES THE VALUE OF NON-LINEAR COEFFICIENT FOR RECTANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      HX=0.0
      HY=0.0
      DO 10 JN=1,NN
      J=NOD(M,JN)
      HX=HX+SLX(NT,JN)*H(J)
      HY=HY+SLY(NT,JN)*H(J)
   10 CONTINUE
      HS=ABS(SQRT(HX**2+HY**2))
C
C     EVALUATES ELEMENT VELOCITY.
C
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      ACONST=1./TEMP
      RETURN
      END
      SUBROUTINE VCHECR(M,AK,BK,NT,PMK,VCR,GRRX,GRRY,NN)
C
C     COMPUTES ELEMENT VELOCITIES AND CHEK IF A PARTICULAR ELEMENT BELONGS TO
C      DARCY OR NON-DARCY FLOW ZONE.
C
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      HX=0.0
      HY=0.0
      DO 10 JN=1,NN
      J=NOD(M,JN)
      HX=HX+SLX(NT,JN)*H(J)
      HY=HY+SLY(NT,JN)*H(J)
   10 CONTINUE
      HS=ABS(SQRT(HX**2+HY**2))
      GRRX=HX/HS
      GRRY=HY/HS
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
```

```
      END
      SUBROUTINE BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,NSCREN,RW)
C
C     IDENTIFIES NODES WHERE HEAD VALUES ARE PRESCRIBED.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      K=0
      NFND=1
      DO 30 I=1,NSCREN
      XST=XSCR(I)-0.01
      XEND=XST+HSCR(I)+0.02
      DO 40 J=1,NNODE
      L=NFND+J-1
      IF(L.EQ.1) GO TO 222
      IF(X(L,1).GT.RW) GO TO 30
  222 CONTINUE
      IF((X(L,2).LT.XST).OR.(X(L,2).GT.XEND)) GO TO 70
   22 K=K+1
      JBD(K)=L
      DISP(K)=HW
      GO TO 40
   70 IF(X(L,2).GT.XEND) GO TO 30
   40 CONTINUE
      NFND=L+1
   30 CONTINUE
      J=K
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
      NBD=J
      RETURN
      END
      SUBROUTINE BNDFIX(IPENTR,ZB,HW,HO,NNODE,NBD,NDW,NDRO,NBW)
C
C     LOCATES NODES WHERE HEAD VALUES ARE FIXED.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      ZTOL=0.01
      J=0
      DO 10 I=1,NDW
      ZDIF=X(I,2)-ZB
      IF(IPENTR.NE.0) ZDIF=-ZDIF
      IF(ZDIF.GT.ZTOL) GO TO 10
      J=J+1
      JBD(J)=I
      DISP(J)=HW
   10 CONTINUE
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
      NBD=J
      RETURN
      END
      SUBROUTINE MERB3(N,M)
C
C     MERGES ELEMENT MATRIX SELK INTO GROSS VECTOR VCORE.
C
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ACORE/VCORE(2000)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VCORE(IPOS)=VCORE(IPOS)+SELK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MERBD(N,M)
C
C     MERGES ELEMENT MATRIX D INTO GROSS VECTOR VD.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
```

```
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VD(IPOS)=VD(IPOS)+D(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SYMSOL(LL,NLL)
C
C     SOLVES A LINEAR SYMMETRIC SYSTEM OF LINEAR EQUATIONS BY BANDED GAUSSIAN
C     ELIMINATION SCHEME.
C
      DIMENSION VTEMP(90)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),N(300),IDUM(300)
      COMMON /BSOLV/C(300,1),V(2000)
      DOUBLE PRECISION TEMP,VTEMP
      JBOUN=1
      ID=1
      DO 10 I=1,LL
      TEMP=V(ID)
      NEB=ID+N(I)-1
      ID1=ID+1
      IF(I.EQ.JBD(JBOUN)) GO TO 16
C
C     NORMALISE ROW I
C
      KK=0
      V(ID)=1.0
      IF(ID1.GT.NEB) GO TO 29
      DO 20 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
   20 V(J)=V(J)/TEMP
   29 CONTINUE
      DO 25 L=1,NLL
      C(I,L)=C(I,L)/TEMP
   25 CONTINUE
      GO TO 46
   16 CONTINUE
      IF(ID1.GT.NEB) GO TO 39
      KK=0
      DO 120 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
  120 V(J)=0.0
   39 CONTINUE
      DO 125 L=1,NLL
  125 C(I,L)=DISP(JBOUN)
      JBOUN=JBOUN+1
C
C     ELIMINATION
C
   46 CONTINUE
      IDJ=ID
      IF(ID1.GT.NEB)  GO TO 35
      KK=0
      DO 30 J=ID1,NEB
      JI=J-ID1+I
      IDJ=IDJ+N(JI)
      KK=KK+1
      IF(VTEMP(KK)) 50,30,50
   50 CONTINUE
      IF(JBOUN.EQ.1) GO TO 240
      JBM1=JBOUN-1
      IF(I.EQ.JBD(JBM1)) GO TO 140
  240 IDP=J
      DO 40 K=IDP,NEB
      KJ=IDJ+K-J
   40 V(KJ)=V(KJ)-V(K)*VTEMP(KK)
  140 CONTINUE
      NJ=I+J-ID
      DO 32 L=1,NLL
      C(NJ,L)=C(NJ,L)-C(I,L)*VTEMP(KK)
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID=ID+N(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID=ID-1
```

```
      LL1=LL-1
      DO 70 IB=1,LL1
      I=LL1-IB+1
      ID=ID-N(I)
      IS=I+1
      IN=I+N(I)-1
      DO 80 J=IS,IN
      NJ=ID+J-I
      DO 75 L=1,NLL
      C(I,L)=C(I,L)-C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE VDFB(LVEC)
C
C     GENERATES VECTOR VDTOP.
C
      DIMENSION D(2,2)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      NELTOP=LVEC-1
      NN=2
      ID(1)=1
      DO 10 I=2,LVEC
   10 ID(I)=ID(I-1)+2
      LENT=2*LVEC-1
      DO 108 I=1,LENT
      VDTOP(I)=0.0
  108 CONTINUE
      DO 208 IE=1,NELTOP
      IN=NDVEC(IE)
      IP=NDVEC(IE+1)
      RAVE=0.5*(X(IP,1)+X(IN,1))
      RDIF=X(IP,1)-X(IN,1)
      D(1,1)=RAVE*RDIF/3.0
      D(2,2)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      DO 308 I=1,2
      IK=IE+(I-1)
      DO 408 J=1,2
      JK=IE+(J-1)
      IF(IK.GT.JK) GO TO 408
      IPOS=ID(IK)+JK-IK
      VDTOP(IPOS)=VDTOP(IPOS)+D(I,J)
  408 CONTINUE
  308 CONTINUE
  208 CONTINUE
      RETURN
      END
      SUBROUTINE GVMOD(LACT,SY,IGK)
C
C     MODIFIES VECTORS VK AND GK TO ACCOUNT FOR LEAKAGE FLUX ACROSS TOP
C     BOUNDARY.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      J2=J1
      JM=J1
      IF(L.LT.LACT) J2=NDVEC(L+1)
      IF(L.GT.1) JM=NDVEC(L-1)
      SUM=VF1*HINT(J1,1)+VF2*HINT(J2,1)+VFM*HINT(JM,1)
      IF(IGK.GT.0) GK(J1,1)=GK(J1,1)+SUM
      IF(IGK.GT.0) GO TO 1078
      IS=ID(I)
      VK(IS)=VK(IS)+VF1
      IL=IS+NBAND(I)-1
      VK(IL)=VK(IL)+VF2
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE GKMOD(LACT,SY)
```

```
C
C     MODIFIES MATRIX GK TO TAKE INTO ACCOUNT LEAKAGE FLUXES.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      LM=L
      LP=L
      IF(L.LT.LACT) LP=L+1
      IF(L.GT.1) LM=L-1
      SUM=VF1*QLEAK(L)+VF2*QLEAK(LP)+VFM*QLEAK(LM)
      GK(J1,1)=GK(J1,1)+SUM
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC BLOCKS.
C
      DIMENSION ND(3,3)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND1+1
      ND(2,1)=ND(1,3)
      ND(2,2)=ND(1,2)
      ND(2,3)=ND(1,2)+1
      IF(NPAT.EQ.1) GO TO 200
      IF(NPAT.EQ.0) GO TO 265
      NTEMP=N1-2
      DO 90 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      DO 50 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      NTEMP=NTEMP-1
      NTSEL=NLST+1
   90 CONTINUE
      GO TO 400
  200 CONTINUE
      ND(3,1)=ND(2,1)
      ND(3,2)=ND(2,2)+1
      ND(3,3)=ND(3,1)+1
      NTEMP=(N1-1)/2-1
      L1=2
      L2=1
      DO 250 JJ=1,3
      L3=2
      IF(JJ.EQ.2) L3=1
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=3
      DO 270 K=1,3
      IF(K.EQ.1) LL=L1
      IF(K.EQ.2) LL=L2
      IF(K.EQ.3) LL=L3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)*LL
  270 CONTINUE
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  250 CONTINUE
      GO TO 400
  265 CONTINUE
      NTEMP=N1-2
      DO 190 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 140 J1=NST,NLST
      NREP(J1)=KCREP
      ITYPE(J1)=3
      DO 150 K=1,3
```

```
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
  150 CONTINUE
  140 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  190 CONTINUE
  400 CONTINUE
      RETURN
      END
      SUBROUTINE HREAD(NNODE,L,INDEX)
C
C     READS IN INITIAL NODAL VALUES.
C
      COMMON /WORKA/VWORK(500)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      READ(L,3)NNODE
      NPOINT=NNODE
    3 FORMAT(I10)
      NCARD=NPOINT/6
      NCARDT=NCARD+1
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      READ(L,13)(VWORK(I),I=IST,IEND)
   13 FORMAT(6E13.5)
      IST=IEND+1
   10 CONTINUE
      NREM=NPOINT-NCARD*6
      IEND=IST+NREM-1
      READ(L,13)(VWORK(I),I=IST,IEND)
      DO 15 I=1,NPOINT
      IF(INDEX.EQ.1) H(I)=VWORK(I)
      IF(INDEX.EQ.2)QLEAK(I)=VWORK(I)
   15 CONTINUE
      RETURN
      END
      SUBROUTINE ELGND3(NT,SC,M)
C
C     GENERATES MATRIX D FOR TRIANGULAR ELEMENTS.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      IF(NREP(M).EQ.NT) GO TO 50
      NT=NREP(M)
      XJ=X(J2,1)-X(J1,1)
      XM=X(J3,1)-X(J1,1)
      YJ=X(J2,2)-X(J1,2)
      YM=X(J3,2)-X(J1,2)
      AREA(NT)=0.5*(XJ*YM-XM*YJ)
      AFUN=2.*AREA(NT)
      B(NT,1)=(YJ-YM)/AFUN
      B(NT,2)=   (YM)/AFUN
      B(NT,3)=  (-YJ)/AFUN
      C(NT,1)=(XM-XJ)/AFUN
      C(NT,2)=  (-XM)/AFUN
      C(NT,3)=   (XJ)/AFUN
      DO 100 I=1,3
      DO 100 J=1,3
      IF(J-I) 105,110,110
  110 ELK(I,J,NT)=B(NT,I)*B(NT,J)+C(NT,I)*C(NT,J)
      GO TO 100
  105 ELK(I,J,NT)=ELK(J,I,NT)
  100 CONTINUE
C
C     FORM ELEMENT MATRIX:- D
C
      D(1,1)=0.5*ORX(M)*AREA(NT)*SC/3.
      D(2,2)=D(1,1)
      D(3,3)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      D(1,3)=D(1,2)
      D(3,1)=D(1,3)
      D(2,3)=D(1,2)
      D(3,2)=D(2,3)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE ELGNCT(M,III,AK,BK,NT,VCOUNT,PMK)
C
C     GENERATES MATRIX SELK FOR TRIANGULAR ELEMENTS.
C
      COMMON /AELEM/BA(150),SLX(150,6),SLY(150,6),SELK(6,6),EK(6,6)
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
```

```
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      IF(IDARCY(M).EQ.0) GO TO 70
  500 CONTINUE
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      CALL CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
      GO TO 80
   70 CONTINUE
      PX=PMK
      ACONST=PX*ORX(M)*AREA(NT)
   80 CONTINUE
C
C                 CALCULATE ELEMENT MATRIX
C
      DO 200 I=1,3
      DO 200 J=1,3
      SELK(I,J)=ACONST*ELK(I,J,NT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
C
C     COMPUTES NON-LINEAR COEFFICIENT FOR TRIANGULAR ELEMENTS.
C
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      TEMP=0.5*AK+ SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      ACONST=AREA(NT)*ORX(M)/TEMP
  500 CONTINUE
      RETURN
      END
      SUBROUTINE VCHEC3(M,AK,BK,J1,J2,J3,NT,PMK,VCR,GRRX,GRRY)
C
C     COMPUTES ELEMENT VELOCITIES FOR TRIANGULAR ELEMENTS.
C
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /CELEM/ELK(3,3,150),B(150,3),C(150,3),AREA(150)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      GRRX=HX/HS
      GRRY=HY/HS
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SETARG(AGP,BGP,PMGP,VGP,AKK,BKK,PKK,VCR,I)
C
C     IDENTIFIES ELEMENTS IN THE GRAVEL PACKED ZONE.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(200)
      IF(IPROP(I).EQ.0) GO TO 20
      GO TO 10
   20 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
      VCR=VGP
   10 CONTINUE
      RETURN
      END
      SUBROUTINE AQDIS(NLEN,QAQFR,QRDIF,DELT,TH,RCSNG,IQ,QFIX,DWD)
C
C     COMPUTES TOTAL DISCHARGE INTO THE WELL.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /BQCAL/VS(600),QWB(40)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      QSUM=0.0
      TWPI=44./7.
      CALL QMULT(NLEN,QSUM)
      QSUM=ABS(TWPI*QSUM)/(0.5*DELT)
      QAQFR=QSUM
      ACSNG=0.5*TWPI*RCSNG**2
      TRM=ABS(ACSNG*DWD*2./DELT)
```

```
      QCALX=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALX)
      QCALC(IQ)=QCALX
      RETURN
      END
      SUBROUTINE QMULT(NBW,QSUM)
C
C     COMPUTES NODAL FLUXES AT WELL BOUNDARY.
C
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      DO 10 L=1,NBW
      I=JBD(L)
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
      QWB(L)=0.0
      DO 20 J = IS, IL
      K=I+J-IS
      QWB(L)=QWB(L)+VKQ(J)*H(K)
   20 CONTINUE
      IF(I.EQ.1) GO TO 15
      I1 = I-1
      IST= ISTART(I)
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
      QWB(L)=QWB(L)+VKQ(IP)*H(J)
   30 CONTINUE
   15 CONTINUE
      QWB(L)=QWB(L)-GP(L)
      QSUM=QSUM+QWB(L)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE ROUT(NNODE)
C
C     PRINTS OUT HEAD VALUES.
C
      COMMON /ADISC/X(300,2),NOD(200,6),NREP(200),ITYPE(200)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /ADPARA/TLESS(300),HLESS(300),VKD(300,1),VDTOP(200)
      DATA NPRINT/3/
      WRITE(NPRINT,53)
   53 FORMAT(///)
      WRITE(NPRINT,3)
      WRITE(NPRINT,13)
      WRITE(NPRINT,23)
      WRITE(NPRINT,13)
      WRITE(NPRINT,3)
    3 FORMAT(5X,50('*'))
   13 FORMAT(5X,'*',48X,'*')
   23 FORMAT(5X,'*',11X,'FINAL RESULTS OF ANALYSIS',12X,'*')
      WRITE(NPRINT,33)
      DO 10 I=1,NNODE
      WRITE(NPRINT,43)I,X(I,1),H(I),TLESS(I),HLESS(I)
   10 CONTINUE
   33 FORMAT(///,20X,'HEAD .VS. RADIUS AND 1/U .VS. W(U)',///,
     1          10X,'NODE',10X,'R-COORD',10X,'HEAD',14X,'1/U',19X,
     1'W(U)',//)
   43 FORMAT(10X,I3,6X,F10.2,8X,F10.4,7X,E11.4,10X,F10.4)
      RETURN
      END
      SUBROUTINE ROEST(DIFFUS,TM,LVEC,QFIX,TMIS,RO,NELEM,NNODE,NELTO,
     1   LST,LSTEL,LVST)
C
C     ESTIMATES RADIUS OF INFLUENCE AT TIME TM.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(50),DISP(50)
      COMMON /VCOM/ORX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      FPI=88./7.
      CONST=QFIX/(FPI*TMIS)
      DO 10 I=LVST,LVEC
      U=RVEC(I)**2/(4.*DIFFUS*TM)
      WW=W(U)
      SDRAW=CONST*WW
      IF(SDRAW.GT.0.001) GO TO 10
      NNODE=NDVEC(I)
      RO=RVEC(I)
      GO TO 20
   10 CONTINUE
      NNODE=NDVEC(LVEC)
      RO=RVEC(LVEC)
   20 CONTINUE
      LST=NNODE
      LVST=I
      DO 30 I=LSTEL,NELTO
      IF(ORX(I).LT.RO) GO TO 30
```

```
      NELEM=I-1
      GO TO 40
   30 CONTINUE
      NELEM=NELTO
   40 CONTINUE
      LSTEL=NELEM
      RETURN
      END
      SUBROUTINE TIGEN(NTICR,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX,TPAT)
C
C     GENERATES DISCRETE TIME VECTOR,TIME.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      SWST=0.8
      CUN=22.*RW**2/(7.*QFIX)
      TIME(1)=TFACTR
      IF(TPAT.EQ.2) GO TO 15
      DO 10 I=2,NTICR
      TIME(I)=TIME(I-1)*TMUL
      TMUL=TMUL+DTMUL
   10 CONTINUE
      GO TO 17
   15 CONTINUE
      DO 20 I=2,NTICR,2
      TIME(I)=TIME(I-1)
      TIME(I+1)=2.*TIME(I)
   20 CONTINUE
      DO 25 I=2,NTICR
      TIME(I)=TIME(I)+TIME(I-1)
   25 CONTINUE
   17 CONTINUE
      RR=RW+1.99
      CONST=RR**2*SS/(4.0*PM)
      IF(IWBC.NE.0)CONST=1000.*CUN*SWST
      DO 30 I=1,NTICR
      TIME(I)=TIME(I)*CONST
   30 CONTINUE
      RETURN
      END
      FUNCTION W(U)
C
C     EVALUATES THEIS WELL FUNCTION.
C
      WC=-0.5772-ALOG(U)
      W=WC
      TERM=1
      J=1
      DO 10 I=1,30
      TERM=(TERM*U)/I
      IF(J.EQ.0) GO TO 20
      W=W+TERM/I
      J=0
      GO TO 30
   20 CONTINUE
      W=W-TERM/I
      J=1
   30 CONTINUE
      EPSI=ABS(W-WC)
      IF(EPSI.LE.0.01) GO TO 40
      WC=W
   10 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE VECMUL(NNODE,LEN)
C
C     PERFORMS MATRIX MULTIPLICATION.
C       VKD  =  VK*D
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VK(2000),D(300,1),IPROP(200)
      COMMON /ADPARA/TLESS(300),HLESS(300),VKD(300,1),VDTOP(200)
      L=1
      DO 10 I=1,NNODE
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
C
      VKD(I,1) = 0.0
      DO 20 J = IS, IL
      K=I+J-IS
      VKD(I,1)=VKD(I,1)+VK(J)*D(K,L)
   20 CONTINUE
C
      IF(I.EQ.1) GO TO 40
      I1 = I-1
      IST= ISTART(I)
C
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
```

```
      VKD(I,1)=VKD(I,1)+VK(IP)*D(J,L)
   30 CONTINUE
   40 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE FQSET(IT,ITCUR,DELT,FQ,TPAT)
C
C
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      IF(TPAT.EQ.1) GO TO 20
      IF(IT.GE.3) GO TO 10
      IF((IT.EQ.1).AND.(ITCUR.GT.IT)) FQ=SW(IT)
      GO TO 20
   10 CONTINUE
      DO 15 I=1,IT
      HDLTA=0.499*DELT
      IF(HDLTA.GT.TIME(I)) GO TO 15
      FQ=SW(I)
      GO TO 20
   15 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SWMOD(IQ,IT,RCSNG,DLTA,QFIX,QAQFR,SWOLD,DSOLD,TPAT,FQ)
C
C     ADJUSTS THE VALUE OF WELL DRAWDOWN.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      DELT=DLTA/10.**3
      ACSNG=22.0*RCSNG**2/7.0
      IF(IQ.NE.1) GO TO 15
      IF(ACSNG.GT.0.0) GO TO 35
      IF(IT.GT.5) GO TO 10
      DSW(IQ)=-DSOLD
      GO TO 20
   35 CONTINUE
      IF(IT.GT.5) GO TO 10
      QRATIO=QAQFR/QFIX
      IF(ABS(QRATIO).GE.0.8) GO TO 10
      TMM=TIME(IT)-0.5*DLTA
      FACTR=0.5*DELT/ACSNG
      QDEL=QFIX-QAQFR*0.95*TIME(IT)/TMM
      IF(ABS(QRATIO).GE.0.5) QDEL=QFIX-QAQFR
      DSW(IQ)=FACTR*QDEL
      DSW(IQ)=-DSW(IQ)
      GO TO 20
   10 CONTINUE
      TOLD=TIME(IT-2)
      ARGL1=(TIME(IT)-0.5*DLTA)/TOLD
      ARGL2=TIME(IT-1)/TOLD
      TLOG=ALOG(ARGL1)/ALOG(ARGL2)
      DSW(IQ)=(SW(IT-1)-SW(IT-2))*(1.-TLOG)
      DSW(IQ)=-DSW(IQ)
      IF(DSW(IQ).GT.0.) DSW(IQ)=0.0
      GO TO 20
   15 CONTINUE
      IF(IQ.GT.2) GO TO 25
      IF(TPAT.EQ.2) GO TO 62
      DSW(IQ)=DSW(IQ-1)*QFIX/QCALC(IQ-1)
      IF(ACSNG.LE.0.) DSW(IQ)=SW(IT)*QFIX/QCALC(IQ-1)-SWOLD
      GO TO 20
   62 CONTINUE
      IF((ACSNG.LE.0.).AND.(FQ.LT.0.)) GO TO 40
      DSW(IQ)=DSW(1)*QFIX/QCALC(1)
      GO TO 20
   40 CONTINUE
      SWCOR=SW(IT)-FQ*(QCALC(1)-QFIX)/QFIX
      DSW(IQ)=SWCOR-SWOLD
      GO TO 20
   25 CONTINUE
      DDSW=DSW(IQ-1)-DSW(IQ-2)
      TERM1=QFIX-QCALC(IQ-1)
      TERM2=QCALC(IQ-1)-QCALC(IQ-2)
      DQR=TERM1/TERM2
      DSW(IQ)=DSW(IQ-1)+DDSW*DQR
   20 CONTINUE
      SW(IT)=DSW(IQ)+SWOLD
      DSOLD=-DSW(IQ)
      RETURN
      END
      SUBROUTINE HPUNCH(NNODE,L,INDEX)
C
C     PUNCHES OUT HEAD VALUES AT FINAL TIME.
C
      COMMON /WORKA/VWORK(500)
      COMMON /VCOM/DRX(200),VEL(200),H(300),IDARCY(200),D(6,6)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      NPOINT=NNODE
```

```
      DO 15 I=1,NPOINT
      IF(INDEX.EQ.1) VWORK(I)=H(I)
      IF(INDEX.EQ.2) VWORK(I)=QLEAK(I)
   15 CONTINUE
      NCARD=NNODE/6
      NCARDT=NCARD+2
      WRITE(3,23)NCARDT
   23 FORMAT(///,10X,'TOTAL NUMBER OF PUNCHED CARDS   =',I10)
      WRITE(L,3)NNODE
    3 FORMAT(I10)
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      WRITE(L,13)(VWORK(I),I=IST,IEND)
   13 FORMAT(6E13.4)
      IST=IEND+1
   10 CONTINUE
      NREM=NNODE-NCARD*6
      IEND=IST+NREM-1
      WRITE(L,13)(VWORK(I),I=IST,IEND)
      RETURN
      END
      SUBROUTINE BSIMP(NNODE,TH,DINDEX,SY,DELT,IT,ITMIN)
C
C     EVALUATES BOULTON'S CONVOLUTIONAL INTEGRAL BY SIMPSON'S 1/3 RULE.
C
      DIMENSION FN(3)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),H(50),HINT(50,1),GP(40)
C
      DOUBLE PRECISION DEXP,ARGU,FN,QCUNT
C
      WRITE(3,3)
    3 FORMAT(///,20X,'LEAKAGE NODAL FLUX',///,10X,'NODE NO.',
     1             20X,'FLUX VALUE',//)
      ITM=IT-1
      IF(ITM.EQ.0) GO TO 5
      TM=TIME(ITM)
      TOLD=(TIME(IT)+TIME(ITM))*0.5
      GO TO 15
    5 TM=0.0
      TOLD=0.5*TIME(IT)
   15 TMM=(TIME(IT)+TIME(IT+1))*0.5
C
      DO 20 II=1,3
      TC=TM+DELT*(II-1)*0.5
      ARGU=-(TMM-TC)*DINDEX
      FN(II)=DEXP(ARGU)
   20 CONTINUE
      ARGU=-DINDEX*(TMM-TOLD)
      NNOD1=NNODE-1
      QLEAK(NNODE)=0.0
      DO 30 I=1,NNOD1
      IF(IT.GT.ITMIN) GO TO 40
      T1=2.*FN(1)*(HINT(I,1)-HCOLD(I))
      T2=4.*FN(2)*(H(I)-HCOLD(I))
      T3=2.*FN(3)*(H(I)-HINT(I,1))
      DQ   =T1+T2+T3
      GO TO 50
   40 CONTINUE
      DQ   =(H(I)-HINT(I,1))*(FN(1)+4.*FN(2)+FN(3))
   50 CONTINUE
      DQ=-DQ
      QCUNT=QLEAK(I)*DEXP(ARGU)
      QLEAK(I)=QCUNT+DQ
      WRITE(3,13)I,QLEAK(I)
   13 FORMAT(10X,I5,20X,E13.3)
   30 CONTINUE
      WRITE(3,13) NNODE,QLEAK(NNODE)
      RETURN
      END
      FUNCTION HPW(U,RM,SCLEN,TH)
C
C     EVALUATES HANTUSH'S WELL FUNCTION FOR PARTIALLY PENETRATING WELLS.
C
      DOUBLE PRECISION U,HWF,DSUM,SUM
      DIF=TH-SCLEN
      PI=22./7.
      PISQ=PI**2
      FACTR=2.*TH*TH/(PISQ*SCLEN**2)
      SUM=0.0
      DO 30 I=1,50
      APSI=3.14*I*RM
      ARGU=I*3.14*DIF/TH
      TERM=SIN(ARGU)**2/I**2
      DSUM=TERM*HWF(U,APSI)
      SUM=SUM+DSUM
      IF((I.GT.15).AND.(SUM.LT.1.D-10)) GO TO 40
      IF(SUM.LT.1.D-20) GO TO 30
      SR=DSUM/SUM
```

```
      STOL=ABS(SR)
      IF(STCL.LT.0.0001) GO TO 40
   30 CONTINUE
   40 CONTINUE
      UX=U
      SUM=SUM*FACTR+W(UX)
      HPW=SUM
      RETURN
      END
      FUNCTION HJW(U,RB,Y)
C
C     EVALUATES INTEGRAND EXPRESSION OF HANTUSH'S WELL FUNCTION.
C
      DOUBLE PRECISION U,F,DEXP,HJW,TERM
      TERM=-Y-RB**2/(4.*Y)
      F=DEXP(TERM)/Y
      HJW=F
      RETURN
      END
      FUNCTION HWF(U,APSI)
C
C     EVALUATES HANTUSH FUNCTION BY SIMSON'S 3-POINT FORMULA.
C
      DOUBLE PRECISION HDW,U,HWF,DSUM,SUM,DW1,DW2,DW3,DABS,DY,HJW
      TMUL=1.4
      Y1=U
      SUM=0.0
      DO 30 I=1,50
      Y3=Y1*TMUL
      Y2=(Y1+Y3)/2.
      DY=Y3-Y1
      DW1=HJW(U,APSI,Y1)
      DW2=HJW(U,APSI,Y2)
      DW3=HJW(U,APSI,Y3)
      DSUM=DY*(DW1+4.*DW2+DW3)/6.
      SUM=SUM+DSUM
      Y1=Y3
      IF((I.GT.15).AND.(SUM.LT.1.D-10)) GO TO 40
      IF(SUM.LT.1.D-20) GO TO 30
      SR=DSUM/SUM
      STOL=ABS(SR)
      IF(STOL.LT.0.0001) GO TO 40
   30 CONTINUE
   40 CONTINUE
      HWF=SUM
      RETURN
      END
      SUBROUTINE DSWINT(DSOLD,BK,QFIX,TMIS,TH,RW,SCLEN,DIFFUS,CUNST)
C
C     ESTIMATES WELL DRAWDOWN FOR FIRST TIME STEP.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(60),QCALC(5)
      DOUBLE PRECISION UW
      DLAMDA=BK*QFIX*TMIS/(TH*TH*RW)
      DLAMDA=7.*DLAMDA/44.
      RM=RW/TH
      GAMMA=SCLEN/TH+0.01
      CONS=4.0*DIFFUS*0.5*TIME(1)
      UW=CONS/RW**2
      V=UW
      UW=1./UW
      U=UW
      TERM=0.25*DLAMDA*(ALOG(V)-3.0)
      WW=W(U)
      IF(GAMMA.LT.1.) WW=HPW(UW,RM,SCLEN,TH)
      DSOLD=WW+1.5*DLAMDA+TERM
      IF(U.GT.0.1E-01) DSOLD=WW
      DSOLD=DSOLD*CUNST
      RETURN
      END
```

4.7 LISTING OF STFREE

```
C
C
C      INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C      DEVELOPED BY P.S. HUYAKORN.
C
C      STFREE,PROGRAM FOR SOLVING STEADY STATE,DARCY OR TWO-REGIME,FREE SURFACE
C
C      FLOW USING TRIANGULAR ELEMENTS.
C
C      VERSION DATED OCTOBER,1973.
C
C       FOR FURTHER INFORMATION,CONTACT
C
C         P.S. HUYAKORN OR C.R. DUDGEON
C
C         WATER RESEARCH LABORATORY
C
C         KING ST.,MANLY VALE
C
C         SYDNEY,N.S.W.,AUSTRALIA.
C
C-----------------------------
C      LIST OF INPUT VARIABLES
C-----------------------------
C
C---------------ALL READ STATEMENTS ARE LOCATED IN THE MAIN------------------
C
C---------------PROGRAM AND INDICATED BY  $1 OR $1,SK  SIGN------------------
C
C***   PROBLEM VARIABLES  ***
C
C      NPROB    =  NUMBER OF PROBLEMS TO BE SOLVED
C      IVEL     =  VELOCITY PRINT-OUT INDEX
C                  FEED IN IVEL=0 IF VELOCITY PRINT-OUT IS NOT REQUIRED OTHERWISE
C                  FEED IN IVEL=1
C      IDISCR   =  DISCRETISATION PRINT-OUT INDEX.FEED IN IDISCR=0 IF PRINT-OUT
C                  OF DISCRETISATION DATA IS NOT REQUIRED
C                  OTHERWISE FEED IN IDISCR=1
C      ORELAX   =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                  SUGGESTED VALUE LIES BETWEEN 1.00 AND 1.50
C      RW       =  RADIUS OF WELL SCREEN
C      RO       =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C      HO       =  INITIAL HEIGHT OF WATER TABLE ABOVE THE BASE OF AQUIFER
C                  IN THIS PARTICULAR PROGRAM,HO MUST NOT BE INTERPRETED AS THE
C                  INITIAL DRAWDOWN OF THE WATER TABLE
C      TH       =  THICKNESS OF AQUIFER
C      HTOL     =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                  SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C      IGP      =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                  IGP=0 FOR NON-GRAVEL PACKED WELL
C      NSTEP    =  A PARAMETER USED TO INDICATE WHETHER ONE- OR TWO- STEP
C                  FREE SURFACE ITERATION IS REQUIRED
C                  FEED IN NSTEP=2 FOR FREE SURFACE FLOW WITH SEEPAGE FACE
C                  OTHERWISE FEED IN NSTEP=1
C      AK       =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C      BK       =  NON-LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                  IF ONLY DARCY FLOW SOLUTION IS REQUIRED SET BK=0.
C      VCR      =  CRITICAL FLOW VELOCITY WHERE NON-DARCY FLOW COMMENCES
C      AGP      =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C      BGP      =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C      VGP      =  CRITICAL FLOW VELOCITY FOR GRAVEL PACK MATERIAL
C      THGP     =  THICKNESS OF GRAVEL PACK
C      BTGP     =  HEIGHT OF BASE OF GRAVEL PACK ABOVE BASE OF AQUIFER
C      HW       =  HEIGHT OF WATER LEVEL IN THE WELL ABOVE BASE OF AQUIFER
C                  IN THIS PARTICULAR PROGRAM HW MUST NOT BE INTERPRETED AS
C                  WELL DRAWDOWN
C
C***   DISCRETISATION PARAMETERS  ***
C
C      FRLEN    =  LENGTH OF FIRST SUBREGION
C                  SUGGESTED VALUE IS FRLEN=RW
C                  FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
C      SCFAC    =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                  SUBREGIONS.SUGGESTD VALUE IS SCFAC=1.50
C      XLMAX    =  PRESCRIBED MAXIMUM BLOCK LENGTH TO AVOID ILL-CONDITIONED
C                  ELEMENTS.MAXIMUM VALUE OF XLMAX SHOULD NOT EXCEED 25.*TH
C      IREG - 1 =  NUMBER OF REPEATED REGULAR BLOCKS WITH THE SAME NUMBER OF NODES
C                  ON THE LEFT AND RIGHT VERTICAL LINES
C                  SUGGESTED VALUE IS IREG=2
C      NMIN     =  MINIMUM NUMBER OF NODES ALONG A VERTICAL LINE
C                  TO MINIMISE THE TOTAL NUMBER OF NODES,SUGGEST NMIN=2
C      NDSC     =  TOTAL NUMBER OF NODES ON WELL SCREEN(S)
C                  NDSC IS TO BE GREATER THAN OR EQUAL TO 2
C      NSCREN   =  NUMBER OF SCREENED INTERVALS
C      XSCR(I)  =  Z-COORDINATE OF BASE OF SCREEN I ABOVE DATUM
C      HSCR(I)  =  LENGTH OF SCREENED INTERVAL I
C
C-----------------------------
C      LIST OF OUTPUT VARIABLES
```

```
C-----------------------------
C       NNODE    =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C       X        =  RADIAL AND VERTICAL NODAL COORDINATES
C       H        =  NODAL HEAD OR DRAWDOWN VALUES
C       RVEC     =  RADIAL COORDINATES OF FREE SURFACE NODES
C       HF       =  HEIGHTS OF FREE SURFACE NODES
C       TLESS    =  NODAL VALUES OF DIMENSIONLESS RAIDUS,1/U
C       HLESS    =  NODAL VALUES OF WELL FUNCTION FOR STEADY FLOW,W(U)
C       QSUM     =  CALCULATED DISCHARGE FROM AQUIFER INTO WELL
C       NELEM    =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C       NOD      =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C       VEL      =  ABSOLUTE ELEMENT VELOCITIES
C       VCOMP1   =  RADIAL COMPONENT OF ELEMENT VELOCITY
C       VCOMP2   =  VERTICAL COMPONENT OF ELEMENT VELOCITY
C       IDARCY   =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                   OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                   IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
C
C
C
C
C
C
C
C
C
C
      DIMENSION JSEEP(20),DSEEP(20)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      COMMON /BQCAL/VKQ(600),QHB(40)
C
C
      DATA NREAD,NPRINT/1,3/
C
C     **************
C     *            *
C     *  BLOCK 1   *
C     *            *
C     **************
C
C     PRINT INITIAL HEADINGS
C
      WRITE(NPRINT,993)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1023)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1033)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1003)
  993 FORMAT(1H1,4X,51H *************************************************
     1**)
 1003 FORMAT(5X,51H ***************************************************)
 1013 FORMAT(5X,51H *                                                 *)
 1023 FORMAT(5X,51H *          FINITE ELEMENT SOLUTION OF             *)
 1033 FORMAT(5X,51H *     STEADY,TWO-REGIME,UNCONFINED FLOW           *)
C
C     READ AND PRINT GENERAL DATA
C
      READ(NREAD,1011)NPROB,IVEL,IDISCR,ORELAX                         $1
 1011 FORMAT(3I10,F10.2)
      DO 4800 JPRO=1,NPROB
      WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1          I6,12X,'*',/,20X,50('*'))
      READ(NREAD,2003) RW,RO,HO,TH,HTOL                                $1
 2003 FORMAT(5F10.2)
      WRITE(NPRINT,2193)RW,RO,HO,TH,ORELAX,HTOL
 2193 FORMAT(///,20X,'GENERAL INPUT DATA',//,
     1             10X,'RADIUS OF WELL  =',F10.2,//,
     2             10X,'EXTERNAL RADIUS  =',F10.2,//,
     3         10X,'HEIGHT OF WATER TABLE AT EXTERNAL RADIUS  =',F10.2,//,
     4           10X,'INITIAL SATURATED THICKNESS OF AQUIFER  =',F10.2,//,
     5             10X,'OVER-RELAXATION FACTOR  =',F10.2,//,
     6             10X,'PRESCRIBED HEAD TOLERANCE  =',F10.2)
      READ(NREAD,201)IGP,NSTEP                                          $1
  201 FORMAT(2I10)
      WRITE(NPRINT,193)IGP,NSTEP
  193 FORMAT(/,10X,'GRAVEL PACK INDEX  =',I5,//,
     1          10X,'APPLY',I3,' STEP ITERATIVE SCHEME')
C
C     READ IN MATERIAL DATA
C
      READ(NREAD,2013) AK,BK,VCR                                        $1
 2013 FORMAT(3F10.3)
      PMK=1./AK
```

```
      IF(VCR.GT.0.00001) PMK=1./(AK+BK*VCR)
      WRITE(NPRINT,23) AK,BK,VCR
   23 FORMAT(//,20X,'AQUIFER PROPERTIES ',//,
     1              10X,'FORCHHEIMER COEFF. A  =',F10.4,//,
     2              10X,'FORCHHEIMER COEFF. B  =',F10.4,//,
     3              10X,'CRITICAL FLOW VELOCITY  =',F10.4)
      RGP=RW
      IF(IGP.EQ.0) GO TO 29
      READ(NREAD,331)AGP,BGP,VGP,THGP,BTGP                              $1,SK
  331 FORMAT(5F10.3)
      RGP=THGP+RW
      GRGP=AGP*VGP+BGP*VGP**2
      PMGP=1./AGP
      IF(GRGP.GT.0.) PMGP=VGP/GRGP
      WRITE(NPRINT,153)AGP,BGP,VGP,PMGP
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1              10X,'COEFFICIENT A =',F10.4//
     2              10X,'COEFFICIENT B = ',F10.4//
     3              10X,'CRITICAL VELOCITY = ',F10.4//
     4              10X,'COEFFICIENT K = ',F10.4//)
      WRITE(NPRINT,163)THGP,RGP
  163 FORMAT(10X,' THICKNESS OF PACK =',F10.2//
     1              10X,'RADIUS OF PACK =',F10.2//)
   29 CONTINUE
C
C     READ ON WELL DRAWDOWN OR HYDRAULIC HEAD
C
      READ(NREAD,2023)HW                                                $1
 2023 FORMAT(F10.2)
C
C     **************
C     *            *
C     *  BLOCK 2   *
C     *            *
C     **************
C
C     READ AND PRINT DISCRETIZATION PARAMETERS.
C
      READ(NREAD,901)FRLEN,SCFAC,XLMAX,IREG,NMIN                        $1
  901 FORMAT(3F10.2,2I10)
      READ(NREAD,801)NDSC,NSCREN                                        $1
  801 FORMAT(2I10)
      IPENTR=0
C
C     READ WELL SCREEN DATA
C
      WRITE(NPRINT,553)
  553 FORMAT(//,5X,'SCREEN NO.',10X,'BASE HEIGHT',14X,'LENGTH',//)
      SCLEN=0.0
      DO 702 I=1,NSCREN                                                 *NSCREN*
      READ(NREAD,601)XSCR(I),HSCR(I)                                    $1
  601 FORMAT(2F10.2)
      SCLEN=SCLEN+HSCR(I)
      WRITE(NPRINT,563)I,XSCR(I),HSCR(I)
  563 FORMAT(7X,I6,13X,F11.2,8X,F10.2)
  702 CONTINUE                                                          /*
C
C     GENERATE AND PRINT DISCRETIZATION DATA.
C
      CALL GXNODU(IPENTR,NDSC,SCLEN,FRLEN,SCFAC,XLMAX,IREG,NMIN,RW,RO,
     1       TH,NNODE,NELEM,LVEC,NLM,NVLST,HEIGHT,NDFLEX,HO,HW,IDISCR)
C
C     GENERATE IPROP.
C
      CALL AQPROP(NELEM,RGP,BTGP,TH,IGP)
C
C     **************
C     *            *
C     *  BLOCK 3   *
C     *            *
C     **************
C
C     FIX VALUES AT BOUNDARIES THAT HAVE VALUES FIXED BUT NOT EQUAL 0.0
C
      HWDRAW=HW-HO
      HODRAW=0.0
      ZB=TH-SCLEN
      IF(IPENTR.EQ.0) ZB=SCLEN
      NDW=NDVEC(1)
      NDRO=NDVEC(LVEC)-NDVEC(LVEC-1)
      CALL BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,NSCREN,RW)
      NBQ=NBW+1
C
C     MODIFY VECTORS JBD + DISP
C
      CALL BNDMOD(NBW,HW,NBD,1,NBDTO)
      WRITE(NPRINT,203)
  203 FORMAT(////,10X,'PRESCRIBED NODES AND HEAD VALUES',//,
     1          10X,'NODE',6X,'PRESCRIBED VALUES',//)
      DO 200 I=1,NBD
```

```
      WRITE(NPRINT,213)JBD(I),DISP(I)
  213 FORMAT(8X,I5,8X,F10.3)
  200 CONTINUE
      IF(NSTEP.EQ.1) GO TO 706
C
C     FOR TWO STEP ITERATION,ALSO PRESCRIBE FREE SURFACE POINTS.
C
      IST=NBD+1
      IEND=NBD+LVEC-1
      J=0
      DO 806 I=IST,IEND
      J=J+1
      JBD(I)=NDVEC(J)
      DISP(I)=HF(J)
  806 CONTINUE
      NBDTO=IEND
  706 CONTINUE
C
C     FIND BANDWIDTHS ETC.
C
      CALL EBFIN3(LEN,NELEM,3,NNODE)
      WRITE(NPRINT,233)LEN
  233 FORMAT(//////,10X,25HLENGTH OF GROSS VECTOR = ,I8,//)
C
C     FORM MATRIX ELK
C
      NT=0
      NN=3
      DO 135 IE=1,NELEM
      CALL ELKGEN(IE,NT)
  135 CONTINUE
C
C     SET IDARCY
C
      DO 99 I=1,NELEM
      IDARCY(I)=0
   99 CONTINUE
      NBIS=NBW
      IF(XSCR(1).GT.0.) NBIS=0
C
C
C      SET UP LOOP FOR ITERATION ON H
C
      WRITE(NPRINT,1583)HW
 1583 FORMAT('1',///,10X,'*******************************************',/,
     1                 10X,'*   WATER LEVEL IN THE WELL =',F7.2,3X,'*',/,
     2                 10X,'*******************************************')
      VCOUNT=0
      SFTOL=HTOL
      NCOUNT=NNODE
C
C     ONE OR TWO STEP ITERATIVE LOOP.
C
      DO 999 III=1,10
C
      NHW=0
      NBP=NBW+1
      CALL BNDMOD(NBW,HW,NBD,1,NBDTO)
      IF(NBIS.EQ.0) GO TO 94
      DO 117 I=1,NBW
      CK(I,1)=DISP(I)
      IF(CK(I,1).LE.HW) NHW=NHW+1
  117 CONTINUE
   94 CONTINUE
      DO 899 ISTEP=1,NSTEP
      IF(ISTEP.EQ.2) NBP=NHW+1
      IF(NBIS.EQ.0) NBP=1
      DO 305 I=NBP,NNODE
      CK(I,1)=0.0
  305 CONTINUE
      IF(ISTEP.EQ.2) GO TO 509
C
C     **************
C     *            *
C     *  BLOCK 4   *
C     *            *
C     **************
C
C     ZERO GROSS STIFFNESS MATRIX AND LOAD MATRIX
C
      DO 300 I=1,LEN
      VK(I)=0.0
  300 CONTINUE
C
C     MODIFY MATRIX ELK
C
      IF(III.EQ.1) GO TO 307
      NT=0
      DO 207 IE=NLM,NVLST
      CALL ELKGEN(IE,NT)
```

```
  207 CONTINUE
C
C     MODIFY VECTORS JBD + DISP.
C
      NBX=NBW
      CALL BNDMOD(NBX,HW,NBD,NSTEP,NBDTO)
  307 CONTINUE
C
C      COMPUTE STIFFNESS MATRIX FOR EACH ELEMENT AND MERGE
C
C      FOR III=1  ONLY PERFORM SOLUTION FOR DARCY CASE
C
      DO 350 I=1,NELEM
      NT=NREP(I)
      IF(IPROP(I).EQ.0) GO TO 755
      AKK=AK
      BKK=BK
      PKK=PMK
      GO TO 365
  755 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
  365 CONTINUE
      CALL ELGENU(I,III,AKK,BKK,NT,VCOUNT,PKK)
      CALL MERB3(NN,I)
  350 CONTINUE
      I=NDVEC(1)+1
      LNN=ID(I)-1
      DO 122 I=1,LNN
      VKQ(I)=VK(I)
  122 CONTINUE
      DO 1202 I=1,LEN
      VD(I)=VK(I)
 1202 CONTINUE
C
      NBF=NBD
      IF(NSTEP.EQ.2) NBF=NBDTO
      GO TO 809
  509 CONTINUE
C
C     SHUFFLE MATRICES JBD + DISP.
C
      J=0
      DO 709 I=1,NBW
      HWTEMP=HW+0.01
      IF(DISP(I).LE.HWTEMP) GO TO 709
      J=J+1
      JSEEP(J)=JBD(I)
      JBD(I)=JBD(NBW+J)
      DSEEP(J)=DISP(I)
      DISP(I)=DISP(NBW+J)
  709 CONTINUE
      NBF=NBD-J
      IF(J.EQ.0) L=NBW
      IF(J.EQ.0) GO TO 7199
C
      DO 719 I=1,J
      IPOS=JSEEP(I)
      L=NBW-J+I
      CK(IPOS,1)=QWB(L)
  719 CONTINUE
 7199 CONTINUE
      IP=IPOS+1
      CK(IP,1)=QWB(L+1)
C
C     REGENERATE MATRIX VK.
C
      DO 1302 I=1,LEN
      VK(I)=VD(I)
 1302 CONTINUE
  809 CONTINUE
C
C     **************
C     *            *
C     *  BLOCK 5   *
C     *            *
C     **************
C
C     SOLVE EQUATIONS BY BANDED SOLUTION SCHEME
C
      NLL=1
      NBM=NBP-1
      CALL QFLUX(NBM,NNODE)
      IF(NBF.LE.NBD) GO TO 482
      CALL SHUF(1,NBD,NBW,NBDTO)
      CALL SYMSOL(NNODE,NLL,NBM)
      CALL SHUF(2,NBD,NBW,NBDTO)
      GO TO 582
  482 CONTINUE
```

```
      CALL SYMSOL(NNODE,NLL,NBM)
  582 CONTINUE
      IF(NSTEP.EQ.1) GO TO 959
      IF(ISTEP.EQ.2) GO TO 959
      DO 949 I=1,NNODE
      H(I)=CK(I,1)
  949 CONTINUE
      NBQ=NBW+1
      QSUM=0.0
      CALL QMUL (NBQ,QSUM)
  959 CONTINUE
      IF(ISTEP.EQ.1) GO TO 899
C
C     RESHUFFLE MATRICES JBD + DISP.
C
      IF(J.EQ.0) GO TO 899
      DO 729 I=1,J
      IPOS=NBW-J+I
      JBD(IPOS)=JSEEP(I)
      DISP(IPOS)=DSEEP(I)
  729 CONTINUE
      IF(ISTEP.LT.NSTEP) NBP=NBW+1-J
  899 CONTINUE
      IF(III.EQ.1) GO TO 900
  415 CONTINUE
      NCOUNT=0
      EMAX=0.0
      DO 450 I=1,NNODE
      EPSI=CK(I,1)-H(I)
      IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.HTOL)  GO TO 460
      NCOUNT=NCOUNT+1
      H(I)=H(I)+ORELAX*EPSI
      GO TO 450
  460 CONTINUE
      H(I)=CK(I,1)
  450 CONTINUE
  473 FORMAT(//,10X,'TOLERANCE COUNTER FOR HEAD   =',I3,//,
     1             10X,'ABSOLUTE MAXIMUM ERROR IN HEAD   =',F12.4)
      GO TO 1999
  900 CONTINUE
      DO 950 I=1,NNODE
      H(I)=CK(I,1)
  950 CONTINUE
 1999 CONTINUE
C
C     REGENERATE IDARCY
C
      DO 199 I=1,NELEM
      CALL HYPROP(I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,
      AL,BL,PML,VCL)
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      NT=NREP(I)
      CALL VCHEC3(I,AL,BL,J1,J2,J3,NT,PML,VCL,HRRX,HRRY)
  199 CONTINUE
C
C     MODIFY THE VARIABLE MESH.
C
      CALL MOMESH(KSURF,ESMAX,LVEC,NDFLEX,SFTOL,HEIGHT,ORELAX)
C
C     PRINT OUT ADJUSTED FREE SURFACE POSITON.
C
      IF(NCOUNT.EQ.0) GO TO 1000
  999 CONTINUE
 1000 CONTINUE
      WRITE(NPRINT,413) III
  413 FORMAT(//,10X,'NUMBER OF ITERATIONS REQUIRED =',I10)
      WRITE(NPRINT,473)NCOUNT,EMAX
      CALL HOUT(NNODE,HO,HW,RO)
      CALL SURDUT(KSURF,ESMAX,LVEC,NDFLEX)
      IF(NCOUNT.NE.0)  GO TO 5000
  433 FORMAT('1',20X,'FINAL SOLUTION',///,
     1             5X,'NODE',10X,'R-COORD',10X,'Z-COORD',10X,'HEAD',
     2            10X,'RHO-COORD',10X,'TZI-COORD',10X,'DRAWDOWN RATIO',/)
      DO 470 I=1,NNODE
      RHO=0.5*X(I,1)/TH
      TZI=X(I,2)/TH
      SDLESS=(HO-H(I))/(HO-HW)
  470 CONTINUE
C
C     **************
C     *            *
C     *  BLOCK 6   *
C     *            *
C     **************
C
C     EVALUATE VELOCITY AND GRADIENT USING FINAL HEADS
C
  799 CONTINUE
      VCOUNT=1
```

```
C
C     COMPUTE AND PRINT ELEMENT VELOCITIES
C
      IF(IVEL.EQ.0) GO TO 3509
      WRITE(NPRINT,1203)
 1203 FORMAT('1',///,20X,'*****************************************'/
     1               20X,'*          ELEMENT VELOCITIES            *'/
     2               20X,'*****************************************'///
     310X,'ELEM NO.',10X,'RADIAL VEL',10X,'VERTIC VEL',10X,'IDARCY',///)
      DO 3000 I=1,NELEM
                                  CALL HYPROP( I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,
                                  AL,BL,PML,VCL)
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      NT=NREP(I)
      CALL VCHEC3(I,AL,BL,J1,J2,J3,NT,PML,VCL,HRRX,HRRY)
      VCOMP1=-VEL(I)*HRRX
      VCOMP2=-VEL(I)*HRRY
      WRITE(NPRINT,1103)I,VCOMP1,VCOMP2,IDARCY(I)
 1103 FORMAT(10X,I5,2(10X,F10.3),10X,I5)
 3000 CONTINUE
 3509 CONTINUE
C
C     COMPUTE DISCHARGE INTO WELL.
C
      CALL QCALC(NBQ,QSUM)
      BLEAK=RO
      J=1
      CALL TCURV3(HO,BLEAK,NNODE,QSUM,AK,BK,PMK,TH,J)
C
C---------------------------------------------------------------------C
C                                                                     C
C            IF SO REQUIRED,USER MAY CALL HIS PLOTTING SUBROUTINES    C
C                                                                     C
C            AT THIS LOCATION.                                        C
C                                                                     C
C---------------------------------------------------------------------C
C
C     THE FOLLOWING PLOTS MAY BE OBTAINED -
C
C            (1) DISCRETISATION PATTERN,OBTAINED BY USING VARIABLES NOD,X,
C                NELEM,NNODE
C            (2) CONTOURS OF HYDRAULIC HEADS,OBTAINED BY USING VARIABLES
C                H,X,NNODE,NOD,NELEM
C            (3) DIMENSIONLESS TYPE CURVES,OBTAINED BY USING VARIABLES HLESS,
C                TLESS,NNODE
C            (4) VELOCITY FIELD,OBTAINED BY USING VARIABLES VEL,X,NOD,VCOMP1,
C                VCOMP2
C            (5) LOCATION OF NON-DARCY FLOW ZONE,OBTAINED BY USING VARIABLES
C                IDARCY,NELEM
C            (6) FREE SURFACE CURVE,OBTAINED BY USING VARIABLES HF,RVEC
C
 5000 CONTINUE
 4800 CONTINUE
      STOP
      END
      SUBROUTINE GXNODU(IPENTR,NDSC,SCLEN,FRLEN,SCFAC,XLMAX,IREG,
     1NMIN,RW,RO,TH,NNODE,NELEM,LVEC,NLM,NVLST,HEIGHT,NDFLEX,HO,HW,
     2  IDISCR)
C
C     GENERATES NODAL COORDINATES AND ELEMENT CONNECTIONS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      SCLEN = TH
      DATA NREAD,NPRINT/1,3/
      NESC=NDSC
      CALL DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
      CALL HFREE(RO,HW,HSEEP,HO,RW,NRR)
      TH=0.5*HF(1)
      IF(TH.LT.SCLEN) NDWB(1)=NDSC
      IF(SCLEN.GT.HF(1)) SCLEN=HF(1)-0.02
      NRST=2
      NFR=NDWB(1)-NDSC/2
      NSTFR=NDWB(1)+NDSC/2+2
      IF(NDWB(1).EQ.NDSC) NSTFR=1
      IF(NDWB(1).EQ.NDSC) NFR=NDWB(1)
      IF(NDWB(1).EQ.NDSC) NRST=1
      NTOP=0.4*NFR+1
      IF(NRST.GT.1) NTOP=NTOP+1
      NSTOR=NRST
      NDWB(1)=NDSC+NTOP-1
      DO 10 L=1,2
      CALL NCRGU1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,L)
      NRST=1
      NSTFR=NFR
      NFR=NTOP
   10 CONTINUE
```

```
      NRST=NSTOR
      NSTFR=1
      IF(NRST.GT.1) NSTFR=NDWB(1)-NTOP+1
      CALL NCRGU2(NSTFR,NRR,NRST,NDSC)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,303)
  303 FORMAT('1',20X,'DISCRETISATION DATA')
   94 CONTINUE
C
C     DISCRETISE THE ENTIRE REGION INTO FINITE ELEMENTS.
C
      NRST=NSTOR
      KCREP=1
      NTSEL=1
      DO 691 L=1,2
      IF(L.EQ.2) NRST=1
      IF(NRST.EQ.1) GO TO 655
      CALL DCRFST(IPENTR,NDSC,KCREP,NTSEL)
  655 CONTINUE
      DO 65 I=NRST,NRR
      N1=NH1(L,I)
      N2=NH2(L,I)
      N3=NH2(2,I)
      IF((N1.EQ.3).AND.(N3.EQ.0)) NLMX=NTSEL
      IF(N2.EQ.0) GO TO 89
      NFND1=NDB1(L,I)
      NFND2=NDB2(L,I)
      NPAT=NTRAN(L,I)
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
   65 CONTINUE
      IF(L.EQ.1) NLM=NTSEL
   89 CONTINUE
      NH1(1,I)=2
  691 CONTINUE
      IF(N2.NE.0) NLMX=NLM
      NELEM=NTSEL-1
      DO 77 I=NLMX,NELEM
      NREP(I)=NREP(I-1)+1
   77 CONTINUE
C
C     PRINT OUT ELEMENT DATA.
C
C
      NELEM=NTSEL-1
      IF(IDISCR.EQ.0) GO TO 104
      WRITE(NPRINT,83)
   83 FORMAT(///,10X,46H IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS,/
     1//,12X,7HELEM NO,10X,5HNODE1,10X,5HNODE2,10X,5HNODE3,10X,17HREPETI
     2TION NUMBER,//)
      DO 80 I=1,NELEM
      WRITE(NPRINT,93)I,(NOD(I,K),K=1,3),NREP(I)
   93 FORMAT(10X,I5,3(10X,I5),12X,I5)
   80 CONTINUE
  104 CONTINUE
C
C     GENERATE NODAL COORDINATES.
C
      NRST=NSTOR
      NS=1
      RI=RW
      MM=1
      HEIGHT=TH
      DO 75 I=1,NRR
      DXG=XLEN(I)
      ZI=0.0
      TH=HEIGHT
      NN2=NH1(2,I)
      IF(NN2.EQ.0) GO TO 537
      DO 87 L=1,2
      IF(I.LT. NRST) GO TO 97
      NN=NH1(L, I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH
      NS=NS-1
      RI=RI-DXG
      TH=HF(I)-HEIGHT
      GO TO 87
   97 CONTINUE
      CALL CORFST(IPENTR,NDSC,SCFAC,RI,TH,SCLEN,RW,NS)
   87 CONTINUE
      RI=RI+DXG
      NS=NS+1
      GO TO 75
  537 CONTINUE
      TH=HF(I)
      NN=NH1(1,I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   75 CONTINUE
      IF(NH2(2,NRR).NE.0) GO TO 999
      NN=NH1(1,NRR)
```

```
      TH=HF(NRR+1)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      GO TO 909
  999 CONTINUE
      ZI=0.0
      TH=HEIGHT
      NN=NH1(1,NRR)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      RI=RI-DXG
      ZI=TH
      NS=NS-1
      TH=HF(I)-HEIGHT
      NN=NH1(2,NRR)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
  909 CONTINUE
C
C     PRINT OUT NODAL COORDINATES.
C
      NNODE=NS-1
      IF(IDISCR.EQ.0) GO TO 114
      WRITE(NPRINT,103)
  103 FORMAT('1',10X,' NODE DATA',//,10X,' NODE',5X,'R-COORD',5X,'Z-COOR
     1D',//)
      DO 85 I=1,NNODE
      WRITE(NPRINT,113)I,(X(I,K),K=1,2)
  113 FORMAT(10X,I4,2F11.2)
   85 CONTINUE
  114 CONTINUE
C
C     GENERATE TOP FREE SURFACE NODE VECTORS.
C
C
C     DETERMINE RADIAL EXTENT OF VARIABLE MESH.
C
      NDTO=NNODE
      CALL FRVEC(HEIGHT,LVEC,NDTO,NLM,NVLST,NDFLEX,NELEM,IV)
C
C     PRINT OUT FREE SURFACE NODE VECTORS.
C
      IF(IDISCR.EQ.0) GO TO 124
      WRITE(NPRINT,223)
  223 FORMAT(///,20X,'TOP BOUNDARY NODES AND RADIAL COORDINATES.',///,
     1            10X,'NODE NUMBER',20X,'R-COORDINATE',20X,'SURFACE HEIGH
     2T',//)
      DO 335 I=1,LVEC
      WRITE(NPRINT,333)NDVEC(I),RVEC(I),HF(I)
  335 CONTINUE
  333 FORMAT(10X,I7,2(25X,F10.2))
C
C     PRINT OUT NODES OF FLEXIBLE Z-COORDINATES.
C
      WRITE(NPRINT,433) (NVMESH(I),I=1,NDFLEX)
      NLM=NLMX
      WRITE(NPRINT,533) NLM,NVLST
  433 FORMAT(///,20X,'NODES OF FLEXIBLE Z-COORDINATES',//,10(5X,I5))
  533 FORMAT(///,20X,'ELEMENTS OF VARIABLE SHAPE ARE NUMBER',
     1            I6,2X,'TO',I6)
  124 CONTINUE
      RETURN
      END
      SUBROUTINE NCRGU1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,L)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NTRAN,NH1,NH2,NDB1,NDB2
C     ONLY FOR REGIONS 2 TO NRR,REGION 1 IS SPECIALLY TREATED.
C
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      NC=NFR
      NCOUNT=0
      DO 10 I=NRST,NRR
      IRGG=IREG
      IF((I.EQ.NRST).AND.(IPENTR.NE.0)) IRGG=0
      IF((L.EQ.1).AND.(I.EQ.1)) IRGG=0
      IF((L.EQ.2).AND.(I.LE.5)) IRGG=2
      NCOUNT=NCOUNT+1
      IF(NCOUNT.LT.IRGG) GO TO 20
      NCOUNT=0
      NHALF=NC/2
      NREM=NC-2*NHALF
      IF(L.EQ.2) NREM=0
      IF(NREM.GT.0) GO TO 15
      NC1=NC-1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=2
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=NC+1
      GO TO 10
   15 CONTINUE
      NC1=NHALF+1
```

```
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=1
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=2*NH2(L,I)-1
      GO TO 10
   20 CONTINUE
      NH2(L,I)=NC
      NTRAN(L,I)=0
      NH1(L,I)=NH2(L,I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE NCRGU2(NSTFR,NRR,NRST,NDSC)
C
C     MODIFIES NH1,NH2 AND COMPUTES NDB1 AND NDB2.
C
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      IPOS=NRR+2
      IDUM=IPOS-1
      DO 10 I=1,NRR
      N1=NH1(1,I)
      N2=NH1(2,I)
      IF(N1.NE.N2) GO TO 10
      IF(N1.EQ.2) GO TO 15
   10 CONTINUE
      GO TO 25
   15 CONTINUE
      IP1=I+1
      IF(IP1.GT.NRR) GO TO 28
      NH1(1,I+1)=3
      NH2(2,I+1)=0
      IPOS=I+2
      IDUM=I+1
      NTRAN(1,I+1)=2
      NRCOM=I+2
      DO 35 I=NRCOM,NRR
      NH1(2,I)=0
   35 NH2(2,I)=0
   28 CONTINUE
   25 CONTINUE
      IF(NRST.GT.1) GO TO 29
C
      J=NRST
      NDB1(1,J)=NSTFR
      NDB2(1,J)=NSTFR+NH1(1,J)+NH1(2,J)-1
      NDB1(2,J)=NDB1(1,J)+NH1(1,J)-1
      NDB2(2,J)=NDB2(1,J)+NH2(1,J)-1
      J=J+1
      GO TO 39
   29 CONTINUE
      NDB1(2,1)=NSTFR
      NDB2(2,1)=NSTFR+NH1(1,2)+NDSC/2+NTOP
   39 CONTINUE
      DO 50 I=J,NRR
      NDB1(1,I)=NDB1(1,I-1)+NH1(1,I-1)+NH1(2,I-1)-1
      IF(I.EQ.IPOS) NDB1(1,I)=NDB1(1,I)-1
      IF(I.GT.IPOS) NDB1(1,I)=NDB1(1,I)+1
      NDB1(2,I)=NDB1(1,I)+NH1(1,I)-1
      IF(I.EQ.IDUM) NDB1(2,I)=NDB1(2,I)-1
   50 CONTINUE
      NR1=NRR-1
      DO 60 I=J,NR1
      NDB2(1,I)=NDB1(1,I+1)
      NDB2(2,I)=NDB1(2,I+1)
   60 CONTINUE
      IF(IPOS.GT.NRR) GO TO 48
      NDB2(1,NRR)=NDB1(1,NRR)+NH1(1,NRR)+NH2(1,NRR)-2
      NDB2(2,NRR)=NDB2(1,NRR)+NH2(1,NRR)-1
      GO TO 58
   48 CONTINUE
      NDB2(1,NRR)=NDB1(1,NRR)+NH1(1,NRR)+NH1(2,NRR)-1
      NDB2(2,NRR)=NDB2(1,NRR)+NH2(1,NRR)-1
   58 CONTINUE
      RETURN
      END
      SUBROUTINE HFREE(RO,HW,HSEEP,HO,RW,NRR)
C
C     COMPUTES THE HEIGHT OF FREE SURFACE BY EMPLOYING HALL'S FORMULA.
C
      COMMON /BCISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      FACA=1.0-(HW/HO)**2.4
      FACB=1.0+(5.0*RW)/HO
      RATIO=ALOG(RO/RW)
      FACC=1.0+0.02*RATIO
      HSEEP=(HO-HW)*FACA/(FACB*FACC)+HW
C
C     COMPUTE VECTOR HF
```

```
C
      R=RW
      HF(1)=HSEEP
      DO 10 KK=1,NRR
      DX=XLEN(KK)
      R=R+DX
      FACD=2.5*R/(RO-RW)
      FACE=1.5*(R/RO)**1.5
      K1=KK+1
      HF(K1)=HSEEP+(HO-HSEEP)*(FACD-FACE)
      IF(HF(K1).GT.HO) HF(K1)=HO
   10 CONTINUE
C
C     FOR POSITIVE CORRECTION,LOWER ESTIMATED FREE SURFACE.
C
      HSEEP=HSEEP-(HO-HSEEP)*0.10
      NPR=NRR+1
      DO 20 KK=1,NPR
      HF(KK)=HF(KK)-(HO-HF(KK))*0.10
   20 CONTINUE
      RETURN
      END
      SUBROUTINE DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NRR,XLEN
C
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      MAXNR=89
      XLEN(1)=FRLEN
      RLEN=RO-RW
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      IF(XLEN(I).GT.XLMAX) XLEN(I)=XLMAX
      SUM=XLEN(I)+SUM
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I
      DENOM=1.+SCFAC+SCFAC**2
      XLEN(I-2)=(XREM+XLEN(I-1)+XLEN(I-2))/DENOM
      XLEN(I-1)=XLEN(I-2)*SCFAC
      XLEN(I)=XLEN(I-1)*SCFAC
      RETURN
      END
      SUBROUTINE DCRFST(IPENTR,NDSC,KCREP,NTSEL)
C
C     GENERATES DISCRETIZATION DATA FOR COMPOSITE REGIONS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      KCREP=1
      NSEL=1
      IF(IPENTR.EQ.0) GO TO 60
      NUPEN=NDWB(1)-NDSC +1
      N1=NUPEN
      N2=2
      N3=N1
      NFND1=1
      NFND2=N1+1
      NFND3=NDWB(1)+2
      NTSEL=1
      NCPAT=3
      CALL CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
      N1=NDWB(1)-NUPEN
      NPAT=1
      NFND1=NFND2+1
      NFND2=NFND1+NDWB(1)-1
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
      NCPAT=3
      N2=N1/2+1
      N1=NDWB(1)-N2
      N3=N1
      NFND1=NDWB(1)+2
      NFND2=NFND1+N1
      NFND3=NFND2+N2
      CALL CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
      GO TO 400
   60 CONTINUE
      NTSEL=1
      NUPEN=NDSC
      NCPAT=-3
      N1=NDWB(1)
      N2=NUPEN/2+1
      N3=N1-N2+1
      IF(N2.EQ.NUPEN) N3=N1
      NFND1=1
      NFND2=NFND1+N1
```

```
      NFND3=NFND2+N2
      CALL CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
  400 CONTINUE
      RETURN
      END
      SUBROUTINE CORFST(IPENTR,NDSC,SCFAC,RI,TH,SCLEN,RW,NS)
C
C     GENERATES NODAL COORDINATES FOR FIRST COMPOSITE REGIONS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NOB1(3,90),
     1NDB2(3,90),NDWB(9)
      NS=1
      IF(IPENTR.EQ.0) GO TO 60
      PENLEN=SCLEN
      XL1=RW
      XL2=XLEN(1)
      NUPEN=NDWB(1)-NDSC+1
      N1=NUPEN
      N2=2
      N3=N1
      TH1=TH-PENLEN
      TH2=TH1/(N1-1)
      RI=0.0
      NPAT=3
      CALL COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
      RI=RW
      NS=NS-1
      MM=1
      NN=NDSC-1
      DXG=XLEN(1)
      ZI=X(NS,2)
      TH1=PENLEN
      CALL COORDC(NS,MM,NN,DXG,TH1,RI,ZI)
C
      XL1=XLEN(2)
      XL2=XLEN(3)
      N1=NDWB(1)-NDSC/2
      N2=NDSC/2
      N3=N1
      TH1=TH
      TH2=PENLEN
      CALL COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
      GO TO 400
   60 CONTINUE
      NPAT=-3
      RI=RW
      TH1=TH
      TH2=SCLEN
      N1=NDWB(1)
      N2=NDSC/2+1
      N3=N1-N2+1
      XL1=XLEN(1)/(1.+SCFAC)
      XL2=SCFAC*XL1
      CALL COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
  400 CONTINUE
      RETURN
      END
      SUBROUTINE UBLOCK(NFND1,NFND2,NFND3,NPAT,NTSEL,KCREP)
C
C     GENERATES ELEMENT CONNECTIVITIES IN GRADED UNIT BLOCK.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      K=0
      IF(NPAT.NE.3) K=1
      ICOUNT=0
      NST=NTSEL
      NLST=NST+2
      DO 10 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      ICOUNT=ICOUNT+1
      IF(ICOUNT.EQ.2) GO TO 20
      IF(ICOUNT.EQ.3) GO TO 30
      NOD(J1,1)=NFND1
      NOD(J1,2)=NFND2
      NOD(J1,3)=NFND1+1
      GO TO 10
   20 CONTINUE
      NOD(J1,1)=NFND1+K
      NOD(J1,2)=NFND2
      NOD(J1,3)=NFND3+K
      GO TO 10
   30 CONTINUE
      NOD(J1,1)=NFND2
      NOD(J1,2)=NFND3
      NOD(J1,3)=NFND3+1
   10 CONTINUE
      NTSEL=NLST+1
```

```
      RETURN
      END
      SUBROUTINE BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC BLOCKS.
C
      DIMENSION ND(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND1+1
      ND(2,1)=ND(1,3)
      ND(2,2)=ND(1,2)
      ND(2,3)=ND(1,2)+1
      IF(NPAT.EQ.1) GO TO 200
      IF(NPAT.EQ.0) GO TO 265
      NTEMP=N1-2
      DO 90 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      DO 50 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      NTEMP=NTEMP-1
      NTSEL=NLST+1
   90 CONTINUE
      GO TO 400
  200 CONTINUE
      ND(3,1)=ND(2,1)
      ND(3,2)=ND(2,2)+1
      ND(3,3)=ND(3,1)+1
      NTEMP=(N1-1)/2-1
      L1=2
      L2=1
      DO 250 JJ=1,3
      L3=2
      IF(JJ.EQ.2) L3=1
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      DO 270 K=1,3
      IF(K.EQ.1) LL=L1
      IF(K.EQ.2) LL=L2
      IF(K.EQ.3) LL=L3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)*LL
  270 CONTINUE
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  250 CONTINUE
      GO TO 400
  265 CONTINUE
      NTEMP=N1-2
      DO 190 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 140 J1=NST,NLST
      NREP(J1)=KCREP
      DO 150 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
  150 CONTINUE
  140 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  190 CONTINUE
  400 CONTINUE
      RETURN
      END
      SUBROUTINE CBLOCK(KCREP,N1,N2,N3,NFND1,NFND2,NFND3,NTSEL,NCPAT)
C
C     GENERATES ELEMENT CONNECTIVITIES IN COMPOSITE BLOCKS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
C
C     BLOCK1 :- CONNECTS VERTICAL LINE 1 TO LINE 2.
C
      NDF=N1-N3
      NPAT=0
      IF(NDF.NE.0) NPAT=1
      NF2=NFND2
      IF(NPAT.EQ.1) GO TO 20
      NF1=NFND1+(N1-N2)
      IF(NCPAT.NE.3) NF1=NFND1
      NN2=N2
      NN1=N2
```

```
      GO TO 30
   20 CONTINUE
      NF1=NFND1+(N1-2*N2+1)
      IF(NCPAT.NE.3) NF1=NFND1
      NN2=N2
      NN1=2*N2-1
   30 CONTINUE
      CALL BBLOCK(KCREP,NN1,NF1,NF2,NPAT,NTSEL)
C
C     BLOCK 2 :- CONNECTS VERTICAL LINE 2 TO LINE 3.
C
      NS1=NF1
      NS2=NF2
      NPAT=0
      NN1=NN2
      NF1=NF2
      NF2=NFND3+N3-N2
      IF(NCPAT.NE.3) NF2=NFND3
      CALL BBLOCK(KCREP,NN1,NF1,NF2,NPAT,NTSEL)
C
C     BLOCK 3 :- UNIT BLOCK.
C
      IF(NCPAT.NE.3) GO TO 60
      NF1=NS1-1
      NF3=NF2-1
      NF2=NS2
      GO TO 70
   60 CONTINUE
      NF1=NFND1+N1-1-(N3-N2)
      NF2=NFND2+N2-1
      NF3=NFND3+N2-1
      NS1=NF1
   70 CONTINUE
      CALL UBLOCK(NF1,NF2,NF3,NCPAT,NTSEL,KCREP)
C
C     BLOCK 4 :- CONNECT LINE 1 TO LINE 3.
C
      NF1=NFND1
      NF2=NFND3
      IF(NCPAT.NE.3) NF1=NS1+1
      IF(NCPAT.NE.3) NF2=NF3+1
      NN1=N3-N2
      IF(NN1.EQ.1) GO TO 85
      NPAT=0
      CALL BBLOCK(KCREP,NN1,NF1,NF2,NPAT,NTSEL)
   85 CONTINUE
      RETURN
      END
      SUBROUTINE COORDC(NS,MM,NN,DXG,TH,RI,ZI)
C
C     GENERATES NODAL COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /ASOLV/XX(5,90),XY(5,90)
      DY=TH/(NN-1)
      DO 20 L=1,NN
   20 XX(1,L)=RI
      DO 50 K=1,MM
   50 XY(K,1)=ZI
      DO 60 J=2,NN
      JM1=J-1
      DO 70 K=1,MM
      XY(K,J)=XY(K,JM1)+DY
   70 CONTINUE
   60 CONTINUE
      DO 80 J=1,MM
      DO 90 K=1,NN
      L=NN*(J-1)+K+NS-1
      X(L,1)=XX(J,K)
      X(L,2)=XY(J,K)
   90 CONTINUE
   80 CONTINUE
      NEN=NS+(MM*NN)-1
      NS=NEN+1
      RI=X(NEN,1)+DXG
      RETURN
      END
      SUBROUTINE COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
C
C     GENERATES NODAL COORDINATES FOR COMPOSITE BLOCKS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      JJ=0
      MM=1
      IF(NPAT.NE.3) GO TO 20
      NN=N1
      DXG=XL1
      ZI=0.
      TH=TH1
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
```

```
   15 CONTINUE
      ZI=TH1-TH2
      NN=N2
      TH=TH2
      DXG=XL2
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      GO TO 25
   20 CONTINUE
      NN=N1-N3+N2
      DXG=0.
      ZI=0.
      TH=TH2
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH+(TH1-TH2)/(N3-N2)
      NN=N1-NN
      ZI=TH+TH/(NN-1)
      DXG=XL1
      TH=TH1-ZI
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      DXG=XL2
      TH=TH2
      NN=N2
      ZI=0.0
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   25 CONTINUE
      RETURN
      END
      SUBROUTINE FRVEC(TH,LVEC,NDTO,NLM,NVLST,NDFLEX,NELEM,IV)
C
C     GENERATES TOP FREE SURFACE NODE VECTORS AND VARIABLE NODE VECTORS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      J=0
      HW=2.*TH
      DO 10 I=1,NDTO
      JP=J+1
      ZH=HF(JP)
      ZDIF=ABS(ZH-X(I,2))
      IF(ZDIF.GT.0.001) GO TO 10
      J=J+1
      NDVEC(J)=I
      RVEC(J)=X(I,1)
   10 CONTINUE
      LVEC=J
      DO 132 I=1,LVEC
      IF(HW.GT.RVEC(I)) GO TO 132
      IV=I
      GO TO 135
  132 CONTINUE
  135 CONTINUE
C
C     GENERATE VARIABLE NODE VECTORS.
      RLIM=RVEC(IV)
      J=0
      DO 20 I=1,NDTO
      IF(X(I,1).GT.RLIM) GO TO 30
      ZDIF=X(I,2)-TH-0.01
      IF(ZDIF.LT.0.) GO TO 20
      J=J+1
      NVMESH(J)=I
   20 CONTINUE
   30 CONTINUE
      NDFLEX=J
      NVLST=NELEM
      RETURN
      END
      SUBROUTINE BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,NSCREN,RW)
C
C     IDENTIFIES NODES WHERE HEAD VALUES ARE PRESCRIBED.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      K=0
      NFND=1
      DO 30 I=1,NSCREN
      XST=XSCR(I)-0.01
      XEND=XST+HSCR(I)+0.02
      DO 40 J=1,NNODE
      L=NFND+J-1
      IF(L.EQ.1) GO TO 222
      IF(L.GE.NDW) GO TO 30
  222 CONTINUE
      IF((X(L,2).LT.XST).OR.(X(L,2).GT.XEND)) GO TO 70
   22 K=K+1
      JBD(K)=L
      DISP(K)=HW
```

```
      GO TO 40
   70 IF(X(L,2).GT.XEND) GO TO 30
   40 CONTINUE
      NEND=L+1
   30 CONTINUE
      J=K
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
      NBD=J
      RETURN
      END
      SUBROUTINE BNDMOD(NBW,HW,NBD,NSTEP,NBDTO)
C
C     MODIFIES VECTORS JBD + DISP.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      DO 10 I=1,NBW
      J=JBD(I)
      ZJ=X(J,2)
      IF(ZJ.GT.HW) DISP(I)=ZJ
   10 CONTINUE
      IF(NSTEP.EQ.1) GO TO 20
      IP=NBD+1
      DO 15 I=IP,NBDTO
      J=JBD(I)
      DISP(I)=X(J,2)
   15 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE AQPROP(NELEM,RGP,BTGP,TH,IGP)
C
C      COMPUTES VECTOR IPROP.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      DO 10 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      ORX=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      ORY=(X(J1,2)+X(J2,2)+X(J3,2))/3.
      IPROP(I)=1
      IF(IGP.EQ.0) GO TO 10
      IF((ORX.LT.RGP).AND.(ORY.GT.BTGP)) IPROP(I)=0
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MOMESH(KSURF,ESMAX,LVEC,NDFLEX,SFTOL,HEIGHT,ORLXX)
C
C     ADJUSTS FREE SURFACE POSITION AND MODIFY VARIABLE MESH.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      ORELAX=ORLXX
      NST=1
      KSURF=0
      ESMAX=0.
      DO 10 I=1,LVEC
      RR=RVEC(I)
      J=NDVEC(I)
      HSTOR=HF(I)
      EPSI=H(J)-X(J,2)
      IF(ESMAX.LT.ABS(EPSI)) ESMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.SFTOL) GO TO 25
      HF(I)=HF(I)+EPSI/ORELAX
      X(J,2)=HF(I)
      KSURF=KSURF+1
      GO TO 26
   25 CONTINUE
      HF(I)=H(J)
      X(J,2)=HF(I)
   26 CONTINUE
C
C     ADJUSTS Z-COORDINATES OF VARIABLE NODES.
C
      NDT=NVMESH(NST)
      XCC=X(NDT,1)+0.01
      IF(RR.GT.XCC) GO TO 10
      ZDIF=X(NDT,2)-X(NDT-1,2)
      DXRTIO=(HF(I)-HEIGHT)/(HSTOR-HEIGHT)
```

```
      IF(NST.EQ.NDFLEX) GO TO 10
      DO 19 K=NST,NDFLEX
      NK=NVMESH(K)
      XK=X(NK,1)-0.01
      IF(XK.GT.RR) GO TO 29
      ZSTOR=X(NK,2)
      X(NK,2)=X(NK-1,2)+DXRTIO*ZDIF
   19 CONTINUE
   29 CONTINUE
      NST=K
      IF(NST.GT.NDFLEX) NST=NDFLEX
      X(J,2)=HF(I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
C
C     COMPUTES NON-LINEAR COEFFICIENT FOR TRIANGULAR ELEMENTS.
C
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
C
C     EVALUATE VELOCITY AND ITS DIRECTION AT EACH ELEMENT
C
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
C
C     EVALOATION OF THE CONSTANT MULTIPLIER OACONSTO
C
      ACONST=AREA(NT)*ORX(M)/TEMP
  500 CONTINUE
      RETURN
      END
      SUBROUTINE ELKGEN(M,NT)
C
C     GENERATES GEOMETRY MATRIX,ELK,FOR TRIANGULAR ELEMENTS.
C
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
   10 CONTINUE
      IF(NREP(M).EQ.NT)   GO TO 50
      NT=NREP(M)
      XJ=X(J2,1)-X(J1,1)
      XM=X(J3,1)-X(J1,1)
      YJ=X(J2,2)-X(J1,2)
      YM=X(J3,2)-X(J1,2)
      AREA(NT)=0.5*(XJ*YM-XM*YJ)
      AFUN=2.*AREA(NT)
      B(NT,1)=(YJ-YM)/AFUN
      B(NT,2)=   (YM)/AFUN
      B(NT,3)=  (-YJ)/AFUN
      C(NT,1)=(XM-XJ)/AFUN
      C(NT,2)=  (-XM)/AFUN
      C(NT,3)=   (XJ)/AFUN
      DO 100 I=1,3
      DO 100 J=1,3
      IF(J-I) 105,110,110
  110 ELK(I,J,NT)=B(NT,I)*B(NT,J)+C(NT,I)*C(NT,J)
      GO TO 100
  105 ELK(I,J,NT)=ELK(J,I,NT)
  100 CONTINUE
   50 CONTINUE
      RETURN
      END
      SUBROUTINE ELGENU(M,III,AK,BK,NT,VCOUNT,PMK)
C
C     GENERATES ELEMENT MATRIX,EK,FOR TRIANGULAR ELEMENTS.
C
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      IF(IDARCY(M).EQ.0) GO TO 70
      IF(III.EQ.1)  GO TO 70
  500 CONTINUE
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      CALL CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
      GO TO 80
   70 CONTINUE
```

```
      PX=PMK
      ACONST=PX*ORX(M)*AREA(NT)
   80 CONTINUE
      DO 200  I=1,3
      DO 200  J=1,3
      EK(I,J)=ACONST*ELK(I,J,NT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE MERB3(N,M)
C
C     MERGES ELEMENT MATRIX EK INTO GROSS VECTOR VK.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      DO 10  I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20  J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS= ID(IK)+JK-IK
      VK(IPOS)=VK(IPOS)+EK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE EBFIN3(LEN,NM,NN,LL)
C
C     SUBROUTINE TO GENERATE BANDWIDTHS OF BANDED SYMMETRIC GROSS
C     MATRIX. NBAND CONTAINS THE BANDWIDTHS, ID THE POSITION OF THE
C     TERM ON THE DIAGONALS OF THE ORIGINAL MATRIX, LEN IS THE LENGTH
C     THE VECTOR
C
      DIMENSION LV(3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      DO 20  I=1,LL
      NBAND(I)=1
      ISTART(I)=I
   20 CONTINUE
C
C     SCAN THROUGH THE LOCATION VECTOR FOR EACH MEMBER TO FIND
C     THE POSITION OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW
C
      NF2=3
      DO 25  I=1,NM
      DO 30  J=1,NF2
      LV(J)=NOD(I,J)
   30 CONTINUE
      DO 45  J=1,NF2
      IF(LV(J).EQ.0)GO TO 45
      DO 40  K=1,NF2
      IF(LV(J).GT.LV(K))  GO TO 40
      NW=LV(K)-LV(J)+1
      NR=LV(J)
      IF(NW.GT.NBAND(NR))  NBAND(NR)=NW
   40 CONTINUE
   45 CONTINUE
C      SEARCH FOR THE FURTHEREST OFF-LEFT TERM
C
      DO 55  J=1,NF2
      IF(LV(J).EQ.0) GO TO 55
      DO 65  K=1,NF2
      IF(LV(J).LT.LV(K))  GO TO 65
      NW=LV(K)
      NR=LV(J)
      IF(NW.LT.ISTART(NR))  ISTART(NR)=NW
   65 CONTINUE
   55 CONTINUE
   25 CONTINUE
C
C     SET UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN=NBAND(1)
      DO 50  I=2,LL
      IF(NBAND(I)-NBAND(I-1).LT.0)  NBAND(I)=NBAND(I-1)-1
      LEN=LEN+NBAND(I)
   50 CONTINUE
      ID(1)=1
      DO 60  I=2,LL
   60 ID(I)=ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE QFLUX(NBM,NNODE)
C
C     MODIFIES MATRIX CK BY ADDING THE FLUX TERMS WHICH EXIST AT WELL BOUNDARY.
```

```
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      DATA NREAD,NPRINT/1,3/
      IF(NBM.EQ.0) GO TO 20
      NBP=NBM+1
      DO 10 I=NBP,NNODE
      IST=ISTART(I)
      IF(IST.GT.NBM) GO TO 10
      I1=I-1
      DO 30 J=IST,NBM
      K=I1-J+1
      IP=ID(J)+K
      CK(I,1)=CK(I,1)-VK(IP)*CK(J,1)
   30 CONTINUE
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SYMSOL(LL,NLL,NBM)
C
C     AN IN CORE BAND SOLVER.
C     USE IN CONJUNCTION WITH TOPOLOGICAL SUBROUTINE FOR LARGE BANDWIDTH.
C
      DIMENSION VTEMP(90)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /ASOLV/ISTART(300),N(300),IDUM(300)
      COMMON /BSOLV/C(300,1),V(2000)
      DOUBLE PRECISION TEMP
      JBOUN=NBM+1
      NBP=JBOUN
      ID=1
      IF(NBM.EQ.0) GO TO 150
      DO 100 I=1,NBM
      ID=ID+N(I)
  100 CONTINUE
  150 CONTINUE
      DO 10 I=NBP,LL
      TEMP=V(ID)
      NEB=ID+N(I)-1
      ID1=ID+1
      IF(I.EQ.JBD(JBOUN)) GO TO 16
C     NORMALISE ROW I
C
      KK=0
      V(ID)=1.0
      IF(ID1.GT.NEB) GO TO 29
      DO 20 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
   20 V(J)=V(J)/TEMP
   29 CONTINUE
      DO 25 L=1,NLL
      C(I,L)=C(I,L)/TEMP
   25 CONTINUE
      GO TO 46
   16 CONTINUE
      IF(ID1.GT.NEB) GO TO 39
      KK=0
      DO 120 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
  120 V(J)=0.0
   39 CONTINUE
      DO 125 L=1,NLL
  125 C(I,L)=DISP(JBOUN)
      JBOUN=JBOUN+1
C
C     ELIMINATION
C
   46 CONTINUE
      IDJ=ID
      IF(ID1.GT.NEB)  GO TO 35
      KK=0
      DO 30 J=ID1,NEB
      JI=J-ID1+I
      IDJ=IDJ+N(JI)
      KK=KK+1
      IF(VTEMP(KK)) 50,30,50
   50 CONTINUE
      IF(JBOUN.EQ.1) GO TO 240
      JBM1=JBOUN-1
      IF(I.EQ.JBD(JBM1)) GO TO 140
  240 IDP=J
      DO 40 K=IDP,NEB
      KJ=IDJ+K-J
   40 V(KJ)=V(KJ)-V(K)*VTEMP(KK)
  140 CONTINUE
      NJ=I+J-ID
      DO 32 L=1,NLL
```

```
      C(NJ,L)=C(NJ,L)-C(I,L)*VTEMP(KK)
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID=ID+N(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID=ID-1
      LLM1=LL-1
      LL1=LL-NBP
      DO 70 IB=1,LL1
      I=LLM1-IB+1
      ID=ID-N(I)
      IS=I+1
      IN=I+N(I)-1
      DO 80 J=IS,IN
      NJ=ID+J-I
      DO 75 L=1,NLL
      C(I,L)=C(I,L)-C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE SUROUT(KSURF,ESMAX,LVEC,NDFLEX)
C
C     PRINTS OUT FREE SURFACE COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,3)
    3 FORMAT(///,10X,'FREE SURFACE POSITON',//,
     1            15X,'RADIUS',10X,' SURFACE HEIGHT',//)
      DO 10 I=1,LVEC
      WRITE(NPRINT,13)RVEC(I),HF(I)
   10 CONTINUE
   13 FORMAT(10X,F10.2,10X,F10.4)
      WRITE(NPRINT,23) KSURF,ESMAX
   23 FORMAT(///,10X,'SURFACE TOLERANCE COUNTER    =',I10,///,
     1            10X,'MAXIMUM ERROR OF SURFACE HEIGHT    =',F12.4)
C
C     PRINT Z-COORDINTE AND HEAD VALUE OF FLEXIBLE NODES.
C
C***  WRITE(NPRINT,33)
      DO 30 I=1,NDFLEX
      J=NVMESH(I)
      XV=X(J,1)
      ZV=X(J,2)
      HV=H(J)
C***  WRITE(NPRINT,43)J,XV,ZV,HV
   43 FORMAT(///,10X,I5,15X,F12.4,2(10X,F12.4))
   33 FORMAT(///,20X,'HEIGHT AND HEAD OF FLEXIBLE NODES',///,
     1            10X,'NODE NUMBER',10X,'R-COORDINATE',10X,'Z-COORDINATE'
     2            ,10X,'HEAD VALUE',//)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE VCHEC3(M,AK,BK,J1,J2,J3,NT,PMK,VCR,GRRX,GRRY)
C
C     COMPUTES ELEMENT VELOCITIES AND CHECK IF A PARTICULAR ELEMENT BELONGS
C     TO DARCY OR NON-DARCY FLOW ZONE.
C
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      GRRX=HX/HS
      GRRY=HY/HS
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE QMUL(NBW,QSUM)
C
C     COMPUTES NODAL FLUXES AT WELL BOUNDARY.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
```

```
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DO 10 L=1,NBW
      I=JBD(L)
      IF(L.EQ.NBW) I=L
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
      QWB(L)=0.0
      DO 20 J = IS, IL
      K=I+J-IS
      QWB(L)=QWB(L)+VKQ(J)*H(K)
   20 CONTINUE
      IF(I.EQ.1) GO TO 15
      I1 = I-1
      IST= ISTART(I)
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
      QWB(L)=QWB(L)+VKQ(IP)*H(J)
   30 CONTINUE
   15 CONTINUE
      QSUM=QSUM+QWB(L)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE QCALC(NBW,QSUM)
C
C     COMPUTES TOTAL DISCHARGE INTO THE WELL.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DATA NREAD,NPRINT/1,3/
      QSUM=0.0
      CALL QMUL(NBW,QSUM)
      QSUM=ABS(2.*3.1416*QSUM)
      WRITE(NPRINT,4203)QSUM
 4203 FORMAT(////,10X,'TOTAL DISCHARGE INTO THE WELL  =',F10.4)
      RETURN
      END
      SUBROUTINE TCURV3(HO,B,NNODE,QFIX,AK,BK,PMK,TH,J)
C
C     COMPUTES TYPE CURVE FOR STEADY STATE FLOW.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      PARAM=0.0
      TWPI=44./7.
      IF(BK.GT.0) PARAM=BK*QFIX*PMK/(TWPI*HO*B)
      DENOM= 2.*QFIX/(TWPI*PMK)
      WRITE(3,3)PARAM
    3 FORMAT(///,10X,50('*'),/,10X,'*',13X,'STEADY STATE TYPE CURVE',
     1 12X,'*',/,10X,50('*'),///,15X,'LAMBDA      =',F10.4)
      IF(J.EQ.2) GO TO 5
      WRITE(3,23)
      GO TO 15
    5 CONTINUE
      WRITE(3,33)
   15 CONTINUE
   23 FORMAT(//,10X,'NODE NUMBER',19X,'R-COORDINATE',10X,'FUNCTION W(U)'
     1              ,10X,'ARGUMENT U',//)
   33 FORMAT(//,10X,'NODE NUMBER',15X,'Z-COORDINATE',10X,'FUNCTION W(U)'
     1              ,10X,'ARGUMENT U',//)
      DO 10 I=1,NNODE
      SDRAW=HO*HO-H(I)**2
      SLESS=SDRAW/DENOM
      RLESS=X(I,1)/B
      WRITE(3,13)I,X(I,J),SLESS,RLESS
   10 CONTINUE
   13 FORMAT(10X,I5,25X,F10.2,2(10X,E12.4))
      RETURN
      END
      SUBROUTINE HOUT(NNODE,HO,HW,RO)
C
C     PRINTS OUT NODAL VALUES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DATA NREAD,NPRINT/1,3/
C
      WRITE(NPRINT,423)
      DO 10 I=1,NNODE
      RHO=X(I,1)/RO
      TZI=X(I,2)/HO
      SDLESS=(HO-H(I))/(HO-HW)
      WRITE(NPRINT,953)I,X(I,1),X(I,2),H(I),RHO,TZI,SDLESS
   10 CONTINUE
  953 FORMAT(5X,I3,2(6X,F12.2),4(8X,F10.4))
```

```
  423 FORMAT(///,5X,'NODE',5X,'R-COORDINATE',5X,'Z-COORDINATE',9X,
     1'HEAD VALUE',9X,'RHO-COORD',9X,'TZI-COORD',5X,'DIM. DRAWDOWN',//)
      RETURN
      END
      SUBROUTINE SHUF(JF,NBD,NBW,NBDTO)
C
C     SHUFFLES VECTORS JBD AND DISP.
C
      DIMENSION JSEEP(50),DSEEP(50)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      NWID=NBD-NBW
      JS=0
      JST=NBD+1
      JEND=NBDTO
      IF(JF.EQ.2) JST=NBDTO-NWID+1
      DO 10 J=JST,JEND
      JS=JS+1
      JSEEP(JS)=JBD(J)
      DSEEP(JS)=DISP(J)
   10 CONTINUE
      JST=NBW+1
      JEND=NBD
      IF(JF.EQ.2) JEND=NBDTO-NWID
      DO 15 J=JST,JEND
      JS=JS+1
      JSEEP(JS)=JBD(J)
      DSEEP(JS)=DISP(J)
   15 CONTINUE
      JS=0
      NST=NBW+1
      DO 20 J=NST,NBDTO
      JS=JS+1
      JBD(J)=JSEEP(JS)
      DISP(J)=DSEEP(JS)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE HYPROP(I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,AL,BL,PML,VCL)
C
C     IDENTIFIES HYDRAULICS PROPERTIES OF EACH ELEMENT.
C
      COMMON /CSOLV/VE(2000),HINT(300,1),IPROP(500),D(3,3)
      IF(IPROP(I).EQ.0) GO TO 10
      AL=AK
      BL=BK
      PML=PMK
      VCL=VCR
      GO TO 20
   10 CONTINUE
      AL=AGP
      BL=BGP
      PML=PMGP
      VCL=VGP
   20 CONTINUE
      RETURN
      END
```

4.8 LISTING OF TRFREE

```
C
C     INTEGRATED WELL-AQUIFER SOLUTION SYSTEM.
C
C     DEVELOPED BY P.S. HUYAKORN.
C
C     TRFREE,PROGRAM FOR SOLVING TRANSIENT,FREE SURFACE,DARCY OR TWO-REGIME
C
C     FLOW USING TRIANGULAR ELEMENTS.
C
C     VERSION DATED OCTOBER,1973.
C
C      FOR FURTHER INFORMATION,CONTACT
C
C        P.S. HUYAKORN OR C.R. DUDGEON
C
C        WATER RESEARCH LABORATORY
C
C        KING ST.,MANLY VALE
C
C        SYDNEY,N.S.W. 2093,AUSTRALIA.
C
C---------------------------
C     LIST OF INPUT VARIABLES
C---------------------------
C
C---------------ALL READ STATEMENTS ARE LOCATED IN THE MAIN-----------------
C
C---------------PROGRAM AND INDICATED BY  $1 OR $1,SK  SIGN-----------------
C
C***  PROBLEM VARIABLES  ***
C
C     NPROB    =  NUMBER OF PROBLEMS TO BE SOLVED
C     IVEL     =  VELOCITY PRINT-OUT INDEX
C                 FEED IN IVEL=0 IF VELOCITY PRINT-OUT IS NOT REQUIRED OTHERWISE
C                 FEED IN IVEL=1
C     IDISCR   =  DISCRETISATION DATA PRINT-OUT INDEX
C                 FEED IN IDISCR=0 IF DISCRETISATION PRINT-OUT IS NOT REQUIRED
C                 OTHERWISE FEED IN IDISCR=1
C     ORELAX   =  OVER-RELAXATION FACTOR FOR NON-LINEAR HEAD ITERATION
C                 SUGGESTED VALUE LIES BETWEEN 1.00 AND 1.50
C     RW       =  RADIUS OF WELL SCREEN
C     RO       =  EXTERNAL RADIUS OR RADIUS OF INFLUENCE
C     HO       =  INITIAL HEIGHT OF WATER TABLE ABOVE THE BASE OF AQUIFER
C                 IN THIS PARTICULAR PROGRAM,HO MUST NOT BE INTERPRETED AS THE
C                 INITIAL DRAWDOWN OF THE WATER TABLE
C     TH       =  THICKNESS OF AQUIFER
C     HTOL     =  HEAD TOLERANCE FOR NON-LINEAR ITERATION ON HEAD VALUES
C                 SUGGESTED VALUE IS 0.10 OR A FEW PERCENT OF HO-HW
C     QFIX     =  PRESCRIBED WELL DISCHARGE
C     RCSNG    =  RADIUS OF WELL CASING
C     QRTOL    =  RATIO OF DISCHARGE TOLERANCE TO PRESCRIBED WELL DISCHARGE
C                 SUGGESTED VALUE IS 0.01 OR 0.02
C     AK       =  FORCHHEIMER LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                 FEED IN AK = 1./PM IF ONLY DARCY FLOW SOLUTION IS REQUIRED
C                 WHERE PM IS THE COEFFICIENT OF HYDRAULIC CONDUCTIVITY OF
C                 THE AQUIFER
C     BK       =  NON-LINEAR HYDRAULIC COEFFICIENT OF AQUIFER
C                 IF ONLY DARCY FLOW SOLUTION IS REQUIRED FEED IN BK=0.
C     VCR      =  CRITICAL FLOW VELOCITY WHERE NON-DARCY FLOW COMMENCES
C     SS       =  SPECIFIC STORAGE COEFFICIENT OF AQUIFER
C                 FEED IN SS=0. IF SPECIFIC STORAGE EFFECT IS TO BE NEGLECTED
C      SY      =  COEFFICIENT OF SPECIFIC YIELD OF AQUIFER
C     DINDEX   =  RECIPROCAL OF DELAYED YIELD INDEX
C                 FEED IN DINDEX=0. FOR COARSE AQUIFER MATERIAL IN WHICH DELAYED
C                 YIELD EFFECT MAY BE NEGLECTED
C     IGP      =  GRAVEL PACK INDEX,IGP=1 FOR GRAVEL PACKED WELL,
C                 IGP=0 FOR NON-GRAVEL PACKED WELL
C     IBOUND   =  EXTERNAL BOUNDARY INDEX
C                 IBOUND=0 FOR BARRIER BOUNDARY OTHERWISE IBOUND =1
C     IWBC     =  WELL BOUNDARY CONDITION INDEX
C                 IWBC=0 IF EFFECT OF WELL STORAGE IS TO BE NEGLECTED OTHERWISE
C                 IWBC=1
C     NSTEP    =  A PARAMETER USED TO INDICATE WHETHER ONE- OR TWO- STEP
C                 FREE SURFACE ITERATION IS REQUIRED
C                 FEED IN NSTEP=2 FOR FREE SURFACE FLOW WITH SEEPAGE FACE
C                 OTHERWISE FEED IN NSTEP=1
C     AGP      =  LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     BGP      =  NON-LINEAR HYDRAULIC COEFFICIENT OF GRAVEL PACK MATERIAL
C     VGP      =  CRITICAL FLOW VELOCITY FOR GRAVEL PACK MATERIAL
C     THGP     =  THICKNESS OF GRAVEL PACK
C     BTGP     =  HEIGHT OF THE BASE OF GRAVEL PACK ABOVE THE BOTTOM BOUNDARY
C                 OF THE AQUIFER
C
C***  DISCRETISATION PARAMETERS  ***
C
C     NTICR    =  TOTAL NUMBER OF TIME STEPS
C     ITST     =  STARTING TIME STEP NUMBER
C     TFACTR   =  VALUE OF THE FIRST TIME STEP,EXPRESSED IN DIMENSIONLESS FORM
C     TMUL     =  TIME MULTIPLIER,SUGGESTED VALUE VARIES FROM 1.50 TO 2.00
C     DTMUL    =  INCREMENT OF TIME MULTIPLIER
```

```
C                 SUGGESTED VALUE LIES BETWEEN O. TO 0.02
C      FRLEN   =  LENGTH OF FIRST SUBREGION
C      SUGGESTED VALUE IS FRLEN=RW
C                 FOR GRAVEL PACKED WELL,FRLEN MUST NOT EXCEED THICKNESS OF PACK
C      SCFAC   =  SCALE FACTOR TO BE USED IN COMPUTING THE LENGTHS OF REMAINING
C                 SUBREGIONS.SUGGESTED VALUE IS SCFAC=1.50
C      XLMAX   =  PRESCRIBED MAXIMUM BLOCK LENGTH TO AVOID ILL-CONDITIONED
C                 ELEMENTS.MAXIMUM VALUE OF XLMAX SHOULD NOT EXCEED 25.*TH
C      IREG - 1 = NUMBER OF REPEATED REGULAR BLOCKS WITH THE SAME NUMBER OF NODES
C                 ON THE LEFT AND RIGHT VERTICAL LINES
C                 SUGGESTED VALUE IS IREG=2
C      NMIN    =  MINIMUM NUMBER OF NODES ALONG A VERTICAL LINE
C                 TO MINIMISE THE TOTAL NUMBER OF NODES,SUGGEST NMIN=2
C      NDSC    =  TOTAL NUMBER OF NODES ON WELL SCREEN(S)
C                 NDSC IS TO BE GREATER THAN OR EQUAL TO 2
C      NSCREN  =  NUMBER OF SCREENED INTERVALS
C      XSCR(I) =  Z-COORDINATE OF BASE OF SCREEN I ABOVE DATUM
C      HSCR(I) =  LENGTH OF SCREENED INTERVAL I
C
C---------------------------------
C      LIST OF OUTPUT VARIABLES
C---------------------------------
C      NNODE   =  TOTAL NUMBER OF NODES IN THE FINITE ELEMENT NETWORK
C      NELEM   =  TOTAL NUMBER OF ELEMENTS IN THE NETWORK
C      IT      =  TIME STEP NUMBER
C      TMM     =  REAL TIME VALUE AT THE MID-POINT OF TIME STEP IT
C      TM      =  REAL TIME VALUE AT THE END OF TIME STEP IT
C      X       =  RADIAL AND VERTICAL NODAL COORDINATES
C      H       =  NODAL HEAD OR DRAWDOWN VALUES
C      RVEC    =  RADIAL COORDINATES OF FREE SURFACE NODES
C      HF      =  HEIGHTS OF FREE SURFACE NODES
C      TLESS   =  NODAL VALUES OF DIMENSIONLESS TIME,1/U
C      HLESS   =  NODAL VALUES OF WELL FUNCTION FOR TRANSIENT FLOW,W(U)
C      IQ      =  DISCHARGE ITERATION NUMBER
C      SW      =  DRAWDOWN VALUE AT CURRENT TIME
C      QAQFR   =  DISCHARGE FROM AQUIFER INTO WELL AT DRAWDOWN SW
C      QSTRGE  =  DISCHARGE FROM WELL STORAGE
C      QCALL   =  TOTAL CALCULATED DISCHARGE
C                 QCALL = DISCHARGE FROM WELL STORAGE + QAQFR
C      QRDIF   =  RESIDUAL DISCHARGE
C                 QRDIF = THE ABSOLUTE DIFFERENCE BETWEEN PRESCRIBED AND
C                 DISCHARGE AND CALCULATED DISCHARGE
C      NOD     =  NODE CONNECTIONS OF ELEMENTS IN THE FINITE ELEMENT NETWORK
C      VEL     =  ABSOLUTE ELEMENT VELOCITIES
C      VCOMP1  =  RADIAL COMPONENT OF ELEMENT VELOCITY
C      VCOMP2  =  VERTICAL COMPONENT OF ELEMENT VELOCITY
C      IDARCY  =  INDEX TO INDICATE IF A PARTICULAR ELEMENT BELONGS TO DARCY
C                 OR NON-DARCY ZONE,IDARCY=0 FOR ELEMENTS IN THE DARCY ZONE,
C                 IDARCY=1 FOR ELEMENTS IN THE NON-DARCY ZONE
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
       DIMENSION JSEEP(20),DSEEP(20),JDARCY(500)
       COMMON /HSCREN/XSCR(5),HSCR(5)
       COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
       COMMON /BDISC/RVEC(50),NOVEC(50),JBD(90),DISP(90)
       COMMON /AELEM/ELK(3,3,150),EK(3,3)
       COMMON /BELEM/B(150,3),C(150,3),AREA(150)
       COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
```

```
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /ACORE/VCORE(2000),VKORE(2000)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),ZF(50),FINT(50,1),GP(40)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DOUBLE PRECISION DEXP,ARGU,VSHIT
      DATA NREAD,NPRINT/1,3/
C
C     **************
C     *            *
C     *  BLOCK 1   *
C     *            *
C     **************
C
C     PRINT INITIAL HEADINGS
C
      WRITE(NPRINT,993)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1023)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1033)
      WRITE(NPRINT,1013)
      WRITE(NPRINT,1003)
  993 FORMAT(1H1,4X,51H *************************************************
     1**)
 1003 FORMAT(5X,51H **************************************************)
 1013 FORMAT(5X,51H *                                                *)
 1023 FORMAT(5X,51H *         FINITE ELEMENT SOLUTION OF             *)
 1033 FORMAT(5X,51H *    TRANSIENT,TWO-REGIME,FREE SURFACE FLOW.     *)
      READ(NREAD,1011)NPROB,IVEL,IDISCR,ORELAX                          $1
 1011 FORMAT(3I10,F10.2)
C
C     READ AND PRINT GENERAL DATA
C
      DO 4800 JPRO=1,NPROB
      WRITE(NPRINT,9003)JPRO
 9003 FORMAT(///,20X,50('*'),/,20X,'*',13X,'PROBLEM NUMBER  =',
     1          I6,12X,'*',/,20X,50('*'))
C
C     READ AND PRINT GENERAL DATA.
C
      READ(NREAD,2001)RW,RO,HO,TH,HTOL                                  $1
 2001 FORMAT(5F10.2)
      READ(NREAD,201)QFIX,RCSNG,QRTOL                                   $1
  201 FORMAT(3F10.2)
      WRITE(NPRINT,2193)RW,RO,HO,TH,ORELAX,HTOL,QFIX,QRTOL
 2193 FORMAT(///,20X,'GENERAL INPUT DATA',//,
     1          10X,'RADIUS OF WELL  =',F10.2,//,
     2          10X,'EXTERNAL RADIUS  =',F10.2,//,
     3       10X,'INITIAL HEIGHT OF WATER TABLE  =',F10.2,//,
     4         10X,'INITIAL SATURATED THICKNESS OF AQUIFER  =',F10.2,//,
     5          10X,'OVER-RELAXATION FACTOR  =',F10.2,//,
     6          10X,'PRESCRIBED HEAD TOLERANCE  =',F10.2,//,
     7          10X,'PRESCRIBED WELL DISCHARGE  =',F10.2,//,
     8          10X,'DISCHARGE TOLERANCE RATIO  =',F10.3)
      QRTOL=QRTOL*QFIX/2.0
      READ(NREAD,3013)AK,BK,VCR,SS,SY,DINDEX                            $1
 3013 FORMAT(6E10.2)
      SFAC=DINDEX*SY/6.
      DINDEX=DINDEX/1000.
      SS=SS*1000.
      PMK=1./AK
      IF(VCR.GT.0.00001) PMK=1./(AK+BK*VCR)
      PM=PMK
      WRITE(NPRINT,23) AK,BK,VCR,SS,SY,DINDEX
   23 FORMAT(//,20X,'AQUIFER PROPERTIES ',//,
     1          10X,'FORCHHEIMER COEFF. A  =',F10.4,//,
     2          10X,'FORCHHEIMER COEFF. B  =',F10.4,//,
     3          10X,'CRITICAL FLOW VELOCITY  =',F10.4,//,
     4          10X,'SPECIFIC STORAGE  =',E10.3,//,
     5          10X,'SPECIFIC YIELD    =',E10.3,//,
     6       10X,'RECIPROCAL OF DELAYED INDEX  =',E10.3)
      READ(NREAD,111)IGP,IBOUND,IWBC,NSTEP                              $1
  111 FORMAT(4I10)
      WRITE(NPRINT,193)IGP,IBOUND,IWBC,NSTEP
  193 FORMAT(10X,'GRAVEL PACK INDEX   =',I5,//,
     1          10X,'BOUNDARY INDEX   =',I5,//,
     2          10X,'WELL B.C. INDEX   =',I5,//,
     3        10X,'ADOPT',I3,' STEP FREE SURFACE ITERATIVE SCHEME')
      RGP=RW
      IF(IGP.EQ.0) GO TO 29
      READ(NREAD,331)AGP,BGP,VGP,THGP,BTGP                              $1,SK
  331 FORMAT(5F10.3)
      RGP=RW+THGP
      GRGP=AGP*VGP+BGP*VGP**2
      PMGP=1./AGP
```

```
      IF(GRGP.GT.0.) PMGP=VGP/GRGP
      WRITE(NPRINT,153)AGP,BGP,VGP,PMGP
  153 FORMAT(///,20X,'GRAVEL PACK PROPERTIES'///
     1             10X,'COEFFICIENT A =',F10.4//
     2             10X,'COEFFICIENT B = ',F10.4//
     3             10X,'CRITICAL VELOCITY = ',F10.4//
     4             10X,'COEFFICIENT K = ',F10.4)
      WRITE(NPRINT,163)THGP,RGP
  163 FORMAT(10X,'THICKNESS OF PACK = ',F10.2//
     1             10X,'RADIUS OF PACK = ',F10.2//)
   29 CONTINUE
C
C     **************
C     *            *
C     *  BLOCK 2   *
C     *            *
C     **************
C
C     READ AND PRINT TIME DATA.
C
      READ(NREAD,121)NTICR,ITST,TFACTR,TMUL,DTMUL                        $1
  121 FORMAT(2I10,3F10.2)
C
C     READ AND PRINT DISCRETISATION DATA.
C
      READ(NREAD,901)FRLEN,SCFAC,XLMAX,IREG,NMIN                         $1
  901 FORMAT(3F10.2,2I10)
      READ(NREAD,801)NDSC,NSCREN                                         $1
  801 FORMAT(2I10)
      IPENTR=0
C
C     READ WELL SCREEN DATA
C
      WRITE(NPRINT,553)
  553 FORMAT(//,5X,'SCREEN NO.',10X,'BASE HEIGHT',14X,'LENGTH',//)
      SCLEN=0.0
      DO 702 I=1,NSCREN                                                  *NSCREN*
      READ(NREAD,601)XSCR(I),HSCR(I)                                     $1
  601 FORMAT(2F10.2)
      SCLEN=SCLEN+HSCR(I)
      WRITE(NPRINT,563)I,XSCR(I),HSCR(I)
  563 FORMAT(7X,I6,13X,F11.2,8X,F10.2)
  702 CONTINUE                                                           /*
C
C     GENERATE AND PRINT DISCRETIZATION DATA.
C
      CALL GXNODU(IPENTR,NDSC,SCLEN,FRLEN,SCFAC,XLMAX,IREG,NMIN,RW,RO,
     1      TH,NNODE,NELEM,LVEC,NLM,NVLST,HEIGHT,NDFLEX,HO,HW,IDISCR)
C
C     GENERATE IPROP.
C
      CALL AQPROP(NELEM,RGP,BTGP,TH,IGP)
C
C     **************
C     *            *
C     *  BLOCK 3   *
C     *            *
C     **************
C
      TPAT=1
      CALL TIGEN(NTICR,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX,TPAT)
      CALL VOFB(LVEC)
C
C     FIND BANDWIDTHS ETC.
C
      CALL EBFIN3(LEN,NELEM,3,NNODE)
      WRITE(NPRINT,233)LEN
  233 FORMAT(///,10X,'LENGTH OF GROSS VECTOR  =',I8)
C
C     FORM MATRIX ELK
C
      NT=0
      NN=3
      DO 435 IE=1,NELEM
      CALL ELKGEN(IE,NT)
  435 CONTINUE
C
C     CALCULATE HEAD AT VARIOUS TIME VALUES
C
      ROMAX=RO
      TMIS=TH*PM
      FPI=88./7.
      CUNST=QFIX/(FPI*TMIS)
      IF(SS.GT.0.) DIFFUS=PM/SS
      LVST=2
      NST=NDVEC(1)
      LSTEL=NDVEC(1)
      NELTO=NELEM
      DO 642 I=1,LVEC
      HCOLD(I)=HO
```

```
      ZF(I)=HO
      FINT(I,1)=HO
  642 QLEAK(I)=0.
      IF(ITST.EQ.1) GO TO 1245
      CALL HREAD(NNODS,1,1)
      CALL HPUNCH(NNODS,3,1)
      CALL HREAD(LVEC,1,2)
      CALL HPUNCH(LVEC,3,2)
      IST=NNODS+1
      DO 1345 I=IST,NNODE
      H(I)=HO
 1345 CONTINUE
      GO TO 1445
 1245 CONTINUE
      DO 3545 I=1,NNODE
      H(I)=HO
 3545 CONTINUE
 1445 CONTINUE
      DO 245 I=1,NNODE
      HINT(I,1)=H(I)
      GK(I,1)=0.0
  245 CONTINUE
      DELT=TIME(1)
      NDTO=NNODE
      ITMIN=NTICR
      QAQFR=2.*QRTOL
      SWOLD=HO
      DSOLD=0.25*HO
      IF(IWBC.EQ.0) RCSNG=0.0
      IST=1
      FQ=0.0
      NQITER=5
C
C     SET JDARCY.
C
      DO 299 I=1,NELEM
      JDARCY(I)=0
  299 CONTINUE
C
C     LOOPING WITH LOOP PARAMETER IT=1,NTICR
C
      DO 7007 IT=ITST,NTICR
      ITCUR=IT
      IF(IT.GT.1) DELT=TIME(IT)-TIME(IT-1)
      TM=TIME(IT)
      LVST=LVEC
      NST=NNODE
      TMM=TM-DELT*0.5
      WRITE(NPRINT,683)IT
  683 FORMAT(///,10X,35('*'),/,10X,'*',4X,
     1           'TIME STEP NUMBER  =',I5,5X,'*',/,10X,35('*'))
      TMIL=TMM/1000.
      WRITE(NPRINT,333)TMIL
  333 FORMAT(///,10X,41('*'),/,10X,'*',9X,'TIME  =',
     1           E14.3,9X,'*',/,10X,41('*'),//)
      WRITE(NPRINT,343)RO,NNODE,NELEM,LVST
  343 FORMAT(/,10X,'ESTIMATED RADIUS OF INFLUENCE  =',F10.2,
     1//,10X,'CORRESPONDING NO. OF NODES  =',I5,//,
     210X,'CORRESPONDING NO. OF ELEMENTS   =',I5,//,
     310X,'CORRESPONDING COMPONENT OF VECTOR NDVEC   =',I5)
C
C     COMPUTE DIMENSIONLESS TIME.
C
      CONS=TMM
      IF(SS.GT.0.) CONS=4.0*DIFFUS*TMM
      DO 909 I=1,NNODE
      TLESS(I)=CONS/X(I,1)**2
  909 CONTINUE
C
C     FOR WATER TABLE AQUITARD COMPUTE BOUTON'S INCREMENTAL DELAYED YIELD.
C
      ARGU=-DINDEX*DELT*0.5
      TERM=DEXP(ARGU)
      ES=SY*(1.-TERM)
      IF(DINDEX.LE.0.0) ES=SY
      ES=ES*1000.
C
C     INITIALISE HEAD VALUES.
C
      IF(IT.EQ.1) GO TO 165
  677 CONTINUE
      DO 155 I=1,NNODE
      HINT(I,1)=H(I)
  155 CONTINUE
  165 CONTINUE
C
C     SET UP LOOP FOR WELL DISCHARGE ITERATION.
C
      DO 998 IQ=1,NQITER
C
```

```
C     SET IDARCY=JDARCY.
C
      DO 1559 I=1,NNODE
      H(I)=HINT(I,1)
 1559 CONTINUE
      DO 99 I=1,NELEM
      IDARCY(I)=JDARCY(I)
   99 CONTINUE
      WRITE(NPRINT,5003)
 5003 FORMAT(/,5X,30('- '))
      CALL FQSET(IT,ITCUR,DELT,FQ,TPAT)
C
C     ADJUST VALUE OF WELL DRAWDOWN.
C
      CALL SWMOD(IQ,IT,RCSNG,DELT,QFIX,QAQFR,SWOLD,DSOLD,TPAT,FQ,HO)
      WRITE(NPRINT,833)IQ,DSW(IQ),SW(IT)
  833 FORMAT(///,10X,'DISCHARGE ITERATION NUMBER   =',I10,//,
     1           10X,'    HEAD INCREMENT    =',F10.3,//,
     2           10X,'    HEAD VALUE    =',F10.3,/)
      HW=SW(IT)
      IF(SCLEN.GT.HF(1)) SCLEN=HF(1)-0.02
      ZB=TH-SCLEN
      IF(IPENTR.EQ.0) ZB= SCLEN
      NDW=NDVEC(1)
      NDRO=NDVEC(LVEC)-NDVEC(LVEC-1)
      CALL BNDFIX(IPENTR,ZB,HW,HO,NNODE,NBD,NDW,NDRO,NBW)
      NBQ=NBW+1
C
C     MODIFY VECTORS JBD + DISP
C
      CALL BNDMOD(NBW,HW,NBD,1,NBDTO)
      IF(NSTEP.EQ.1) GO TO 706
C
C     FOR TWO STEP ITERATION,ALSO PRESCRIBE FREE SURFACE POINTS.
C
      IST=NBD+1
      IEND=NBD+LVEC-1
      J=0
      DO 806 I=IST,IEND
      J=J+1
      JBD(I)=NDVEC(J)
      DISP(I)=HF(J)
  806 CONTINUE
      NBDTO=IEND
  706 CONTINUE
C
C
C     SET UP LOOP FOR ITERATION ON H
C
      VCOUNT=0
      SFTOL=HTOL
      NCOUNT=NNODE
C
C     ONE OR TWO STEP ITERATIVE LOOP.
C
      DO 999 III=1,10
C
      NHW=0
      NBP=NBW+1
      CALL BNDMOD(NBW,HW,NBD,1,NBDTO)
      DO 117 I=1,NBW
      CK(I,1)=DISP(I)
      IF(CK(I,1).LE.HW) NHW=NHW+1
  117 CONTINUE
      DO 899 ISTEP=1,NSTEP
      IF(ISTEP.EQ.2) NBP=NHW+1
      DO 305 I=NBP,NNODE
      CK(I,1)=GK(I,1)
  305 CONTINUE
      IF(ISTEP.EQ.2) GO TO 509
C
C     **************
C     *            *
C     *  BLOCK 4   *
C     *            *
C     **************
C
C     ZERO GROSS STIFFNESS MATRIX AND LOAD MATRIX
C
      DO 300 I=1,LEN
      VCORE(I)=0.0
      VD(I)=0.0
  300 CONTINUE
C
C     MODIFY MATRIX ELK
C
      IF(III.EQ.1) GO TO 307
      NT=0
      DO 207 IE=NLM,NVLST
      CALL ELKGEN(IE,NT)
```

```
  207 CONTINUE
  307 CONTINUE
C
C     MODIFY VECTORS JBD + DISP.
C
      NBX=NBW
      CALL BNDMOD(NBX,HW,NBD,NSTEP,NBDTO)
C
C      COMPUTE STIFFNESS MATRIX FOR EACH ELEMENT AND MERGE
C
C      FOR III=1  ONLY PERFORM SOLUTION FOR DARCY CASE
C
C
C     FORM GROSS VECTOR VD.
C
      NT=0
      SC=SS
      DO 135 IE=1,NELEM
      CALL ELGND3(NT,SC,IE)
      CALL MERBD(NN,IE)
  135 CONTINUE
      CALL VECMUL(NNODE,LEN)
C
C     COMPUTE BOULTON'S DELAYED YEILD.
C
      CALL GVMOD(LVST,ES,1)
      IF(IT.EQ.1) GO TO 278
      FTERM=0.5*DELT*SFAC
      CALL GKMOD(LVST,FTERM)
  278 CONTINUE
      DO 1378 I=1,NBW
      JJ=JBD(I)
 1378 GP(I)=GK(JJ,1)
      GP(NBW+1)=GK(NBW+1,1)
C
C     SET VECTOR CK.
C
      DO 4305 I=NBP,NNODE
 4305 CK(I,1)=GK(I,1)
      DO 350 I=1,NELEM
      IF(IPROP(I).EQ.0) GO TO 755
      NT=NREP(I)
      AKK=AK
      BKK=BK
      PKK=PMK
      GO TO 365
  755 CONTINUE
      AKK=AGP
      BKK=BGP
      PKK=PMGP
  365 CONTINUE
      CALL ELGENU(I,III,AKK,BKK,NT,VCOUNT,PKK)
      CALL MERB3(NN,I)
  350 CONTINUE
      DO 978 I=1,LEN
      VK(I)=0.0
  978 CONTINUE
      CALL GVMOD(LVST,ES,0)
      DO 530 I=1,LEN
      VSHIT=VCORE(I)*DELT*0.5+VD(I)+VK(I)
      VK(I)=VSHIT
      VKORE(I)=VK(I)
  530 CONTINUE
      I=NDVEC(1)+1
      LNN=ID(I)-1
      DO 122 I=1,LNN
      VKQ(I)=VK(I)
  122 CONTINUE
C
      NBF=NBD
      IF(NSTEP.EQ.2) NBF=NBDTO
      GO TO 809
  509 CONTINUE
C
C     SHUFFLE MATRICES JBD + DISP.
C
      J=0
      DO 709 I=1,NBW
      HWTEMP=HW+0.01
      IF(DISP(I).LE.HWTEMP) GO TO 709
      J=J+1
      JSEEP(J)=JBD(I)
      JBD(I)=JBD(NBW+J)
      DSEEP(J)=DISP(I)
      DISP(I)=DISP(NBW+J)
  709 CONTINUE
C
      NBF=NBD-J
      IF(J.EQ.0) L=NBW
      IF(J.EQ.0) GO TO 7199
```

```
      DO 719 I=1,J
      IPOS=JSEEP(I)
      L=NBW-J+I
      CK(IPOS,1)=QWB(L)+GP(L)
  719 CONTINUE
 7199 CONTINUE
      IP=IPOS+1
      CK(IP,1)=QWB(L+1)+GK(IP,1)
C
C     REGENERATE MATRIX VK.
C
      DO 1302 I=1,LEN
      VK(I)=VKORE(I)
 1302 CONTINUE
  809 CONTINUE
C
C     **************
C     *            *
C     *  BLOCK 5   *
C     *            *
C     **************
C
C     SOLVE EQUATIONS BY BANDED SOLUTION SCHEME
C
      NLL=1
      NBM=NBP-1
      CALL QFLUX(NBM,NNODE)
      IF(NBF.LE.NBD) GO TO 482
      CALL SHUF(1,NBD,NBW,NBDTO)
      CALL SYMSOL(NNODE,NLL,NBM)
      CALL SHUF(2,NBD,NBW,NBDTO)
      GO TO 582
  482 CONTINUE
      CALL SYMSOL(NNODE,NLL,NBM)
  582 CONTINUE
      IF(NSTEP.EQ.1) GO TO 959
      IF(ISTEP.EQ.2) GO TO 959
      DO 949 I=1,NNODE
      H(I)=CK(I,1)
  949 CONTINUE
      NBQ=NBW+1
      CALL QMULT(NBQ,QSUM)
  959 CONTINUE
      IF(ISTEP.EQ.1) GO TO 899
C
C     RESHUFFLE MATRICES JBD + DISP.
C
      IF(J.EQ.0) GO TO 899
      DO 729 I=1,J
      IPOS=NBW-J+I
      JBD(IPOS)=JSEEP(I)
      DISP(IPOS)=DSEEP(I)
  729 CONTINUE
      IF(ISTEP.LT.NSTEP) NBP=NBW+1-J
  899 CONTINUE
C
C     PRINT OUT SOLUTION FOR FUNCTION
C     ONLY THE DARCY AND FINAL SOLUTIONS
C
      IF(III.EQ.1) GO TO 900
  415 CONTINUE
      NCOUNT=0
      EMAX=0.0
      DO 450 I=1,NNODE
      EPSI=CK(I,1)-H(I)
      IF(ABS(EPSI).GT.EMAX) EMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.HTOL)  GO TO 460
      NCOUNT=NCOUNT+1
      H(I)=H(I)+ORELAX*EPSI
      GO TO 450
  460 CONTINUE
      H(I)=CK(I,1)
  450 CONTINUE
      GO TO 1999
  900 CONTINUE
      DO 950 I=1,NNODE
      H(I)=CK(I,1)
  950 CONTINUE
 1999 CONTINUE
C
C     REGENERATE IDARCY
C
      DO 199 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      NT=NREP(I)
      CALL VCHEC3(I,AL,BL,J1,J2,J3,NT,PML,VCL,HRRX,HRRY)
  199 CONTINUE
C
```

CALL HYPROP (I, AK, BK, PMK, VCR, AGP, BGP, PMGP, VGP, AL, BL, PML, VCL)

```
C       MODIFY THE VARIABLE MESH.
C
        CALL MOMESH(KSURF,ESMAX,LVEC,NDFLEX,SFTOL,HEIGHT,ORELAX)
C
C       PRINT OUT ADJUSTED FREE SURFACE POSITON.
C
        IF(NCOUNT.EQ.0) GO TO 1000
  999 CONTINUE
 1000 CONTINUE
      WRITE(NPRINT,413) III
  413 FORMAT(//,10X,'NUMBER OF ITERATIONS REQUIRED =',I10)
      WRITE(NPRINT,473)NCOUNT,EMAX
  473 FORMAT(//,10X,'TOLERANCE COUNTER FOR HEAD   =',I3,//,
     1             10X,'ABSOLUTE MAXIMUM ERROR IN HEAD   =',F12.4)
C
C       CHECK FOR ACCURACY OF DISCHARGE RATIO.
C
      DWD=DSW(IQ)
      CALL AQDIS(NBQ,QAQFR,QRDIF,DELT,TH,RCSNG,IQ,QFIX,DWD)
C
C
C       PRINT FINAL DISCHARGE VALUES.
C
      QCALL=QCALC(IQ)
      QSTRGE=QCALL-QAQFR
      WRITE(NPRINT,1203)QAQFR,QSTRGE,QCALL,QRDIF
 1203 FORMAT(///,10X,'DISCHARGE FROM AQUIFER INTO WELL    =',F12.4,//,
     1          10X,'DISCHARGE FROM WELL STORAGE =',F14.3,//,
     1           10X,'TOTAL CALCULATED DISCHARGE    =',F14.3,//,
     210X,'RESIDUAL DISCHARGE             =',F10.4)
      IF(QRDIF.LE.QRTOL) GO TO 1102
C
C       RESET TRIAL FREE SURFACE POSITION AND ADJUST Z-COORDINATE.
C
      CALL FOMESH(LVEC,NDFLEX,HEIGHT)
  998 CONTINUE
 1102 CONTINUE
      CALL SUROUT(KSURF,ESMAX,LVEC,NDFLEX)
C
C       RESET JDARCY.
C
      DO 3599 I=1,NELEM
      JDARCY(I)=IDARCY(I)
 3599 CONTINUE
      SWOLD=SW(IT)
      DO 470 I=1,NNODE
      IF(ITCUR.GT.IT) TLESS(I)=TLESS(I)*TM/TMM
      HLESS(I)=(HO-H(I))/CUNST
  470 CONTINUE
      CALL ROUT(NNODE)
C
C       COMPUTE AND PRINT ELEMENT VELOCITIES
C
      IF(IVEL.EQ.0) GO TO 557
      WRITE(NPRINT,4203)
 4203 FORMAT(///,20X,'*******************************************'/
     1             20X,'*            ELEMENT VELOCITIES            *'/
     2             20X,'*******************************************'///
     310X,'ELEM NO.',10X,'RADIAL VEL',10X,'VERTIC VEL',10X,'IDARCY',///)
      DO 3000 I=1,NELEM
            CALL HYPROP(I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,
            AL,BL,PML,VCL)
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      NT=NREP(I)
      CALL VCHEC3(I,AL,BL,J1,J2,J3,NT,PML,VCL,HRRX,HRRY)
      VCOMP1=-VEL(I)*HRRX
      VCOMP2=-VEL(I)*HRRY
      WRITE(NPRINT,1103)I,VCOMP1,VCOMP2,IDARCY(I)
 1103 FORMAT(10X,I5,2(10X,F10.3),10X,I5)
 3000 CONTINUE
  557 CONTINUE
      IF(IT.GT.ITMIN) GO TO 577
      IF(ITCUR.GT.IT) GO TO 7000
      TMIL=TM/1000.
      WRITE(NPRINT,333) TMIL
      ITCUR=IT+1
      GO TO 677
  577 CONTINUE
      TMIL=TM/1000.
      WRITE(NPRINT,333) TMIL
C
C       OBTAIN HEAD VALUES AT THE END OF TIME INTERVAL BY EXTRAPOLATION.
C
      DO 477 I=1,NNODE
      TLESS(I)=TLESS(I)*TM/TMM
      H(I)=2.0*H(I)-HINT(I,1)
      HLESS(I)=(HO-H(I))/CUNST
  477 CONTINUE
      SWTEMP=SWOLD
      SWOLD=H(1)-HO
```

```
      QFR=QAQFR
      DWTEMP=SWOLD-SWTEMP
      QAQFR=QFR*SWOLD/SWTEMP
      ACSNG=22.*RCSNG**2/7.
      TRM=ABS(ACSNG*DWTEMP*2./DELT)
      QCALL=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALL)
      SW(IT)=SWOLD
      QWSTR=QCALL-QAQFR
      WRITE(NPRINT,1203)QAQFR,QWSTR,QCALL,QRDIF
 7009 CONTINUE
      CALL ROUT(NNODE)
 7000 CONTINUE
C
C     COMPUTE FIRST PORTION OF BOULTON'S EXPONENTIAL INTEGRAL.
C
      IF(IT.EQ.NTICR) GO TO 7007
      DO 9007 I=1,LVST
      L=NDVEC(I)
      ZF(I)=H(L)
      FINT(I,1)=HINT(L,1)
 9007 CONTINUE
      CALL BSIMP(LVST,TH,DINDEX,SY,DELT,IT,0)
      DO 8007 I=1,LVST
      L=NDVEC(I)
 8007 HCOLD(I)=H(L)
 7007 CONTINUE
C
C     PUNCH OUT SOLUTION AT FINAL TIME.
C
      CALL HPUNCH(NNODE,2,1)
      CALL HPUNCH(NNODE,2,2)
 4800 CONTINUE
 5000 CONTINUE
      STOP
      END
      SUBROUTINE GXNODU(IPENTR,NDSC,SCLEN,FRLEN,SCFAC,XLMAX,IREG,
     1NMIN,RW,RO,TH,NNODE,NELEM,LVEC,NLM,NVLST,HEIGHT,NDFLEX,HO,HW,
     2    IDISCR)
C
C     GENERATES DISCRETISATION DATA.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
C
      DATA NREAD,NPRINT/1,3/
      NESC=NDSC
      CALL DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
      CALL HOFREE(HO,HSEEP,NRR)
      HF(1)=HF(1)-0.01
      THMAX=TH
      TH=0.4*HF(1)
      IF(TH.LT.SCLEN) NDWB(1)=NDSC
      IF(SCLEN.GT.HF(1)) SCLEN=HF(1)-0.02
      NRST=2
      NFR=NDWB(1)-NDSC/2
      NSTFR=NDWB(1)+NDSC/2+2
      IF(NDWB(1).EQ.NDSC) NSTFR=1
      IF(NDWB(1).EQ.NDSC) NFR=NDWB(1)
      IF(NDWB(1).EQ.NDSC) NRST=1
      NTOP=0.4*NFR+1
      IF(NRST.GT.1) NTOP=NTOP+1
      NSTOR=NRST
      NDWB(1)=NDSC+NTOP-1
      DO 10 L=1,2
      CALL NCRGU1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,L)
      NRST=1
      NSTFR=NFR
      NFR=NTOP
   10 CONTINUE
      NRST=NSTOR
      NSTFR=1
      IF(NRST.GT.1) NSTFR=NDWB(1)-NTOP+1
      CALL NCRGU2(NSTFR,NRR,NRST,NDSC)
      IF(IDISCR.EQ.0) GO TO 94
      WRITE(NPRINT,303)
  303 FORMAT('1',20X,'DISCRETISATION DATA')
   94 CONTINUE
C
C     DISCRETISE THE ENTIRE REGION INTO FINITE ELEMENTS.
C
      NRST=NSTOR
      KCREP=1
      NTSEL=1
      DO 691 L=1,2
      IF(L.EQ.2) NRST=1
      DO 65 I=NRST,NRR
```

```
      N1=NH1(L,I)
      N2=NH2(L,I)
      N3=NH2(2,I)
      IF((N1.EQ.3).AND.(N3.EQ.0)) NLMX=NTSEL
      IF(N2.EQ.0) GO TO 89
      NFND1=NDB1(L,I)
      NFND2=NDB2(L,I)
      NPAT=NTRAN(L,I)
      CALL BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
   65 CONTINUE
      IF(L.EQ.1) NLM=NTSEL
   89 CONTINUE
      NH1(1,I)=2
  691 CONTINUE
      IF(N2.NE.0) NLMX=NLM
      NELEM=NTSEL-1
      DO 77 I=NLMX,NELEM
      NREP(I)=NREP(I-1)+1
   77 CONTINUE
C
C     PRINT OUT ELEMENT DATA.
C
C
      NELEM=NTSEL-1
      IF(IDISCR.EQ.0) GO TO 104
      WRITE(NPRINT,83)
   83 FORMAT(///,10X,46H IDENTIFICATION OF ELEMENTS - NODE CONNECTIONS,/
     1//,12X,7HELEM NO,10X,5HNODE1,10X,5HNODE2,10X,5HNODE3,10X,17HREPETI
     2TION NUMBER,//)
      DO 80 I=1,NELEM
      WRITE(NPRINT,93)I,(NOD(I,K),K=1,3),NREP(I)
   93 FORMAT(10X,I5,3(10X,I5),12X,I5)
   80 CONTINUE
  104 CONTINUE
C
C     GENERATE NODAL COORDINATES.
C
      NRST=NSTOR
      NS=1
      RI=RW
      MM=1
      HEIGHT=TH
      DO 75 I=1,NRR
      DXG=XLEN(I)
      ZI=0.0
      TH=HEIGHT
      NN2=NH1(2,I)
      IF(NN2.EQ.0) GO TO 537
      DO 87 L=1,2
      NN=NH1(L,I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH
      NS=NS-1
      RI=RI-DXG
      TH=HF(I)-HEIGHT
   87 CONTINUE
      RI=RI+DXG
      NS=NS+1
      GO TO 75
  537 CONTINUE
      TH=HF(I)
      NN=NH1(1,I)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   75 CONTINUE
      IF(NH2(2,NRR).NE.0) GO TO 999
      NN=NH1(1,NRR)
      TH=HF(NRR+1)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      GO TO 909
  999 CONTINUE
      ZI=0.0
      TH=HEIGHT
      NN=NH1(1,NRR)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      RI=RI-DXG
      ZI=TH
      NS=NS-1
      TH=HF(I)-HEIGHT
      NN=NH1(2,NRR)
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
  909 CONTINUE
C
C     PRINT OUT NODAL COORDINATES.
C
      NNODE=NS-1
      IF(IDISCR.EQ.0) GO TO 114
      WRITE(NPRINT,103)
  103 FORMAT('1',10X,'NODAL COORDINATES',//,10X,'NODE',5X,
     1'R-COORD',5X,'Z-COORD',//)
      DO 85 I=1,NNODE
```

```
      WRITE(NPRINT,113)I,X(I,1),X(I,2)
  113 FORMAT(10X,I4,2F11.2)
   85 CONTINUE
  114 CONTINUE
C
C     GENERATE TOP FREE SURFACE NODE VECTORS.
C
C
C     DETERMINE RADIAL EXTENT OF VARIABLE MESH.
C
      NDTO=NNODE
      CALL FRVEC(HEIGHT,LVEC,NDTO,NLM,NVLST,NDFLEX,NELEM,IV)
C
C     PRINT OUT FREE SURFACE NODE VECTORS.
C
      IF(IDISCR.EQ.0) GO TO 124
      WRITE(NPRINT,223)
  223 FORMAT(///,20X,'TOP BOUNDARY NODES AND RADIAL COORDINATES.',///,
     1             10X,'NODE NUMBER',20X,'R-COORDINATE',20X,'SURFACE HEIGH
     2T',//)
      DO 335 I=1,LVEC
      WRITE(NPRINT,333)NDVEC(I),RVEC(I),HF(I)
  335 CONTINUE
  333 FORMAT(10X,I7,2(25X,F10.2))
C
C     PRINT OUT NODES OF FLEXIBLE Z-COORDINATES.
C
      WRITE(NPRINT,433) (NVMESH(I),I=1,NDFLEX)
      NLM=NLMX
      WRITE(NPRINT,533) NLM,NVLST
  433 FORMAT(///,20X,'NODES OF FLEXIBLE Z-COORDINATES',//,10(5X,I5))
  533 FORMAT(///,20X,'ELEMENTS OF VARIABLE SHAPE ARE NUMBER',
     1             I6,2X,'TO',I6)
  124 CONTINUE
      TH=THMAX
      RETURN
      END
      SUBROUTINE NCRGU1(NFR,IREG,NRR,NMIN,NSTFR,NRST,IPENTR,L)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NTRAN,NH1,NH2,NDB1,NDB2
C     ONLY FOR REGIONS 2 TO NRR,REGION 1 IS SPECIALLY TREATED.
C
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
      NC=NFR
      NCOUNT=0
      DO 10 I=NRST,NRR
      IRGG=IREG
      IF((I.EQ.NRST).AND.(IPENTR.NE.0)) IRGG=0
      IF((L.EQ.1).AND.(I.EQ.1)) IRGG=0
      IF((L.EQ.2).AND.(I.LE.5)) IRGG=2
      NCOUNT=NCOUNT+1
      IF(NCOUNT.LT.IRGG) GO TO 20
      NCOUNT=0
      NHALF=NC/2
      NREM=NC-2*NHALF
      IF(L.EQ.2) NREM=0
      IF(NREM.GT.0) GO TO 15
      NC1=NC-1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=2
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=NC+1
      GO TO 10
   15 CONTINUE
      NC1=NHALF+1
      IF(NC1.LT.NMIN) GO TO 20
      NTRAN(L,I)=1
      NC=NC1
      NH2(L,I)=NC
      NH1(L,I)=2*NH2(L,I)-1
      GO TO 10
   20 CONTINUE
      NH2(L,I)=NC
      NTRAN(L,I)=0
      NH1(L,I)=NH2(L,I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE NCRGU2(NSTFR,NRR,NRST,NDSC)
      COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
     1NDB2(3,90),NDWB(9)
C
C     MODIFIES NH1,NH2 AND GENERATES NDB1 AND NDB2.
C
      IPOS=NRR+2
      IDUM=IPOS-1
      DO 10 I=1,NRR
      N1=NH1(1,I)
```

```
      N2=NH1(2,I)
      IF(N1.NE.N2) GO TO 10
      IF(N1.EQ.2) GO TO 15
   10 CONTINUE
      GO TO 25
   15 CONTINUE
      IP1=I+1
      IF(IP1.GT.NRR) GO TO 28
      NH1(1,I+1)=3
      NH2(2,I+1)=0
      IPOS=I+2
      IDUM=I+1
      NTRAN(1,I+1)=2
      NRCOM=I+2
      DO 35 I=NRCOM,NRR
      NH1(2,I)=0
   35 NH2(2,I)=0
   28 CONTINUE
   25 CONTINUE
      IF(NRST.GT.1) GO TO 29
C
      J=NRST
      NDB1(1,J)=NSTFR
      NDB2(1,J)=NSTFR+NH1(1,J)+NH1(2,J)-1
      NDB1(2,J)=NDB1(1,J)+NH1(1,J)-1
      NDB2(2,J)=NDB2(1,J)+NH2(1,J)-1
      J=J+1
      GO TO 39
   29 CONTINUE
      NDB1(2,1)=NSTFR
      NDB2(2,1)=NSTFR+NH1(1,2)+NDSC/2+NTOP
   39 CONTINUE
      DO 50 I=J,NRR
      NDB1(1,I)=NDB1(1,I-1)+NH1(1,I-1)+NH1(2,I-1)-1
      IF(I.EQ.IPOS) NDB1(1,I)=NDB1(1,I)-1
      IF(I.GT.IPOS) NDB1(1,I)=NDB1(1,I)+1
      NDB1(2,I)=NDB1(1,I)+NH1(1,I)-1
      IF(I.EQ.IDUM) NDB1(2,I)=NDB1(2,I)-1
   50 CONTINUE
      NR1=NRR-1
      DO 60 I=J,NR1
      NDB2(1,I)=NDB1(1,I+1)
      NDB2(2,I)=NDB1(2,I+1)
   60 CONTINUE
      IF(IPOS.GT.NRR) GO TO 48
      NDB2(1,NRR)=NDB1(1,NRR)+NH1(1,NRR)+NH2(1,NRR)-2
      NDB2(2,NRR)=NDB2(1,NRR)+NH2(1,NRR)-1
      GO TO 58
   48 CONTINUE
      NDB2(1,NRR)=NDB1(1,NRR)+NH1(1,NRR)+NH1(2,NRR)-1
      NDB2(2,NRR)=NDB2(1,NRR)+NH2(1,NRR)-1
   58 CONTINUE
      RETURN
      END
      SUBROUTINE DCRGN3(RO,RW,SCFAC,FRLEN,NRR,XLMAX)
C
C     GENERATES DISCRETIZATION PARAMETERS:- NRR,XLEN
C
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      MAXNR=89
      XLEN(1)=FRLEN
      RLEN=RO-RW
      SUM=XLEN(1)
      DO 10 I=2,MAXNR
      XLEN(I)=XLEN(I-1)*SCFAC
      IF(XLEN(I).GT.XLMAX) XLEN(I)=XLMAX
      SUM=XLEN(I)+SUM
      IF(SUM.GT.RLEN) GO TO 20
   10 CONTINUE
   20 CONTINUE
      XREM=RLEN+XLEN(I)-SUM
      NRR=I
      DENOM=1.+SCFAC+SCFAC**2
      XLEN(I-2)=(XREM+XLEN(I-1)+XLEN(I-2))/DENOM
      XLEN(I-1)=XLEN(I-2)*SCFAC
      XLEN(I)=XLEN(I-1)*SCFAC
      RETURN
      END
      SUBROUTINE BBLOCK(KCREP,N1,NFND1,NFND2,NPAT,NTSEL)
C
C     GENERATES ELEMENT CONNECTIVITIES IN BASIC BLOCKS.
C
      DIMENSION ND(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      ND(1,1)=NFND1
      ND(1,2)=NFND2
      ND(1,3)=NFND1+1
      ND(2,1)=ND(1,3)
      ND(2,2)=ND(1,2)
      ND(2,3)=ND(1,2)+1
```

```
      IF(NPAT.EQ.1) GO TO 200
      IF(NPAT.EQ.0) GO TO 265
      NTEMP=N1-2
      DO 90 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 40 J1=NST,NLST
      NREP(J1)=KCREP
      KCREP=KCREP+1
      DO 50 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
   50 CONTINUE
   40 CONTINUE
      NTEMP=NTEMP-1
      NTSEL=NLST+1
   90 CONTINUE
      GO TO 400
  200 CONTINUE
      ND(3,1)=ND(2,1)
      ND(3,2)=ND(2,2)+1
      ND(3,3)=ND(3,1)+1
      NTEMP=(N1-1)/2-1
      L1=2
      L2=1
      DO 250 JJ=1,3
      L3=2
      IF(JJ.EQ.2) L3=1
      NST=NTSEL
      NLST=NST+NTEMP
      DO 260 J1=NST,NLST
      NREP(J1)=KCREP
      DO 270 K=1,3
      IF(K.EQ.1) LL=L1
      IF(K.EQ.2) LL=L2
      IF(K.EQ.3) LL=L3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)*LL
  270 CONTINUE
  260 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  250 CONTINUE
      GO TO 400
  265 CONTINUE
      NTEMP=N1-2
      DO 190 JJ=1,2
      NST=NTSEL
      NLST=NST+NTEMP
      DO 140 J1=NST,NLST
      NREP(J1)=KCREP
      DO 150 K=1,3
      NOD(J1,K)=ND(JJ,K)+(J1-NST)
  150 CONTINUE
  140 CONTINUE
      KCREP=KCREP+1
      NTSEL=NLST+1
  190 CONTINUE
  400 CONTINUE
      RETURN
      END
      SUBROUTINE COORDC(NS,MM,NN,DXG,TH,RI,ZI)
C
C     GENERATES NODAL COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /ASOLV/XX(5,90),XY(5,90)
      DY=TH/(NN-1)
      DO 20 L=1,NN
   20 XX(1,L)=RI
      DO 50 K=1,MM
   50 XY(K,1)=ZI
      DO 60 J=2,NN
      JM1=J-1
      DO 70 K=1,MM
      XY(K,J)=XY(K,JM1)+DY
   70 CONTINUE
   60 CONTINUE
      DO 80 J=1,MM
      DO 90 K=1,NN
      L=NN*(J-1)+K+NS-1
      X(L,1)=XX(J,K)
      X(L,2)=XY(J,K)
   90 CONTINUE
   80 CONTINUE
      NEN=NS+(MM*NN)-1
      NS=NEN+1
      RI=X(NEN,1)+DXG
      RETURN
      END
      SUBROUTINE COORD1(NS,N1,N2,N3,XL1,XL2,TH1,TH2,RI,NPAT)
C
```

```
C     GNERATES NODAL COORDINATES FOR COMPOSITE BLOCK.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      JJ=0
      MM=1
      IF(NPAT.NE.3) GO TO 20
      NN=N1
      DXG=XL1
      ZI=0.
      TH=TH1
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   15 CONTINUE
      ZI=TH1-TH2
      NN=N2
      TH=TH2
      DXG=XL2
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      GO TO 25
   20 CONTINUE
      NN=N1-N3+N2
      DXG=0.
      ZI=0.
      TH=TH2
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      ZI=TH+(TH1-TH2)/(N3-N2)
      NN=N1-NN
      ZI=TH+TH/(NN-1)
      DXG=XL1
      TH=TH1-ZI
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
      DXG=XL2
      TH=TH2
      NN=N2
      ZI=0.0
      CALL COORDC(NS,MM,NN,DXG,TH,RI,ZI)
   25 CONTINUE
      RETURN
      END
      SUBROUTINE FRVEC(TH,LVEC,NDTO,NLM,NVLST,NDFLEX,NELEM,IV)
C
C     GENERATES TOP FREE SURFACE NODE VECTORS AND VARIABLE NODE VECTORS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      J=0
      HW=2.*TH
      DO 10 I=1,NDTO
      JP=J+1
      ZH=HF(JP)
      ZDIF=ABS(ZH-X(I,2))
      IF(ZDIF.GT.0.001) GO TO 10
      J=J+1
      NDVEC(J)=I
      RVEC(J)=X(I,1)
   10 CONTINUE
      LVEC=J
      DO 132 I=1,LVEC
      IF(HW.GT.RVEC(I)) GO TO 132
      IV=I
      GO TO 135
  132 CONTINUE
  135 CONTINUE
C
C     GENERATE VARIABLE NODE VECTORS.
      RLIM=RVEC(IV)
      J=0
      DO 20 I=1,NDTO
      IF(X(I,1).GT.RLIM) GO TO 30
      ZDIF=X(I,2)-TH-0.01
      IF(ZDIF.LT.0.) GO TO 20
      J=J+1
      NVMESH(J)=I
   20 CONTINUE
   30 CONTINUE
      NDFLEX=J
      NVLST=NELEM
      RETURN
      END
      SUBROUTINE BNDFIX(IPENTR,ZB,HW,HO,NNODE,NBD,NDW,NDRO,NBW)
C
C     LOCATES NODES WHERE HEAD VALUES ARE FIXED.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      ZTOL=0.01
      J=0
      DO 10 I=1,NDW
      ZDIF=X(I,2)-ZB
```

```
      IF(IPENTR.NE.0) ZDIF=-ZDIF
      IF(ZDIF.GT.ZTOL) GO TO 10
      J=J+1
      JBD(J)=I
      DISP(J)=HW
   10 CONTINUE
      NBW=J
      NST=NNODE-NDRO+1
      DO 25 I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
      NBD=J
      RETURN
      END
      SUBROUTINE BNDMOD(NBW,HW,NBD,NSTEP,NBDTO)
C
C     MODIFIES VECTORS JBD + DISP.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      DO 10 I=1,NBW
      J=JBD(I)
      ZJ=X(J,2)
      IF(ZJ.GT.HW) DISP(I)=ZJ
   10 CONTINUE
      IF(NSTEP.EQ.1) GO TO 20
      IP=NBD+1
      DO 15 I=IP,NBDTO
      J=JBD(I)
      DISP(I)=X(J,2)
   15 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE AQPROP(NELEM,RGP,BTGP,TH,IGP)
C
C     GENERATES ELEMENT PROPERTIES INDEX,IPROP.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      DO 10 I=1,NELEM
      J1=NOD(I,1)
      J2=NOD(I,2)
      J3=NOD(I,3)
      ORX=(X(J1,1)+X(J2,1)+X(J3,1))/3.
      ORY=(X(J1,2)+X(J2,2)+X(J3,2))/3.
      IPROP(I)=1
      IF(IGP.EQ.0) GO TO 10
      IF((ORX.LT.RGP).AND.(ORY.GT.BTGP)) IPROP(I)=0
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MOMESH(KSURF,ESMAX,LVEC,NDFLEX,SFTOL,HEIGHT,ORLXX)
C
C     ADJUSTS FREE SURFACE POSITION AND MODIFY VARIABLE MESH.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      ORELAX=ORLXX
      NST=1
      KSURF=0
      ESMAX=0.
      DO 10 I=1,LVEC
      RR=RVEC(I)
      J=NDVEC(I)
      HSTOR=HF(I)
      EPSI=H(J)-X(J,2)
      IF(ESMAX.LT.ABS(EPSI)) ESMAX=ABS(EPSI)
      IF(ABS(EPSI).LE.SFTOL) GO TO 25
      HF(I)=HF(I)+EPSI/ORELAX
      X(J,2)=HF(I)
      KSURF=KSURF+1
      GO TO 26
   25 CONTINUE
      HF(I)=H(J)
      X(J,2)=HF(I)
   26 CONTINUE
C
C     ADJUSTS Z-COORDINATES OF VARIABLE NODES.
C
      NDT=NVMESH(NST)
      XCC=X(NDT,1)+0.01
      IF(RR.GT.XCC) GO TO 10
      ZDIF=X(NDT,2)-X(NDT-1,2)
      DXRTIO=(HF(I)-HEIGHT)/(HSTOR-HEIGHT)
```

```
      IF(NST.EQ.NOFLEX) GO TO 10
      DO 19 K=NST,NOFLEX
      NK=NVMESH(K)
      XK=X(NK,1)-0.01
      IF(XK.GT.RR) GO TO 29
      ZSTOR=X(NK,2)
      X(NK,2)=X(NK-1,2)+DXRTIO*ZDIF
   19 CONTINUE
   29 CONTINUE
      NST=K
      IF(NST.GT.NOFLEX) NST=NOFLEX
      X(J,2)=HF(I)
   10 CONTINUE
      JDUM=0
      IF(JDUM.EQ.0) GO TO 59
C
C     READJUST POSITION OF TOP SEEPAGE POINT.
C
      RR=RVEC(1)
      HSTOR=HF(1)
      HF(1)=2.*HF(2)-HF(3)
      I=1
      NST=1
      NDT=NVMESH(NST)
      ZDIF=X(NDT,2)-X(NDT-1,2)
      DXRTIO=(HF(I)-HEIGHT)/(HSTOR-HEIGHT)
      DO 49 K=NST,NOFLEX
      NK=NVMESH(K)
      XK=X(NK,1)-0.01
      IF(XK.GT.RR) GO TO 59
      ZSTOR=X(NK,2)
      X(NK,2)=X(NK-1,2)+DXRTIO*ZDIF
   49 CONTINUE
   59 CONTINUE
      RETURN
      END
      SUBROUTINE CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
C
C     COMPUTES NON-LINEAR COEFFICIENT FOR TRIANGULAR ELEMENTS.
C
C
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
C
C     EVALUATE VELOCITY AND ITS DIRECTION AT EACH ELEMENT
C
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
C
C
      ACONST=AREA(NT)*ORX(M)/TEMP
  500 CONTINUE
      RETURN
      END
      SUBROUTINE ELKGEN(M,NT)
C
C     GENERATES ELEMENT GEOMETRY MATRIX,ELK.
C
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      ORX(M)=(X(J1,1)+X(J2,1)+X(J3,1))/3.
   10 CONTINUE
      IF(NREP(M).EQ.NT)   GO TO 50
      NT=NREP(M)
      XJ=X(J2,1)-X(J1,1)
      XM=X(J3,1)-X(J1,1)
      YJ=X(J2,2)-X(J1,2)
      YM=X(J3,2)-X(J1,2)
      AREA(NT)=0.5*(XJ*YM-XM*YJ)
      AFUN=2.*AREA(NT)
      B(NT,1)=(YJ-YM)/AFUN
      B(NT,2)=   (YM)/AFUN
      B(NT,3)=  (-YJ)/AFUN
      C(NT,1)=(XM-XJ)/AFUN
      C(NT,2)=  (-XM)/AFUN
      C(NT,3)=   (XJ)/AFUN
      DO 100 I=1,3
      DO 100 J=1,3
      IF(J-I) 105,110,110
  110 ELK(I,J,NT)=B(NT,I)*B(NT,J)+C(NT,I)*C(NT,J)
      GO TO 100
  105 ELK(I,J,NT)=ELK(J,I,NT)
```

```
  100 CONTINUE
   50 CONTINUE
      RETURN
      END
      SUBROUTINE ELGENU(M,III,AK,BK,NT,VCOUNT,PMK)
C
C     GENERATES ELEMENT MATRIX EK FOR TRIANGULAR ELEMENTS.
C
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      IF(IDARCY(M).EQ.0) GO TO 70
  500 CONTINUE
      J1=NOD(M,1)
      J2=NOD(M,2)
      J3=NOD(M,3)
      CALL CONSTT(M,AK,BK,ACONST,J1,J2,J3,NT)
      GO TO 80
   70 CONTINUE
      PX=PMK
      ACONST=PX*ORX(M)*AREA(NT)
   80 CONTINUE
C
C                 CALCULATE STIFFNESS MATRIX
C     NORMAL STIFFNESS MATRIX MODIFIED BY THE FACTOR  OACONSTO
C
      DO 200 I=1,3
      DO 200 J=1,3
      EK(I,J)=ACONST*ELK(I,J,NT)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE MERB3(N,M)
C
C     MERGES ELEMENT MATRIX EK INTO GROSS VECTOR VCORE.
C
      COMMON /ACORE/VCORE(2000),VKORE(2000)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK--IK
      VCORE(IPOS)=VCORE(IPOS)+EK(I,J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE EBFIN3(LEN,NM,NN,LL)
C
C     SUBROUTINE TO GENERATE THE BANDWIDTHS OF THE BANDED SYM STIFFNESS
C     MATRIX. NBAND CONTAINS THE BANDWIDTHS, ID THE POSITION OF THE
C     TERM ON THE DIAGONALS OF THE ORIGINAL MATRIX, LEN IS THE LENGTH
C     THE VECTOR
C
      DIMENSION LV(3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      DO 20 I=1,LL
      NBAND(I)=1
      ISTART(I)=I
   20 CONTINUE
C
C     SCAN THROUGH THE LOCATION VECTOR FOR EACH MEMBER TO FIND
C     THE POSITION OF THE TERM FURTHEREST FROM THE DIAGONAL IN EACH ROW
C
      NF2=3
      DO 25 I=1,NM
      DO 30 J=1,NF2
      LV(J)=NOD(I,J)
   30 CONTINUE
      DO 45 J=1,NF2
      IF(LV(J).EQ.0)GO TO 45
      DO 40 K=1,NF2
      IF(LV(J).GT.LV(K)) GO TO 40
      NW=LV(K)-LV(J)+1
      NR=LV(J)
      IF(NW.GT.NBAND(NR))  NBAND(NR)=NW
   40 CONTINUE
   45 CONTINUE
C      SEARCH FOR THE FURTHEREST OFF-LEFT TERM
C
      DO 55 J=1,NF2
      IF(LV(J).EQ.0) GO TO 55
      DO 65 K=1,NF2
```

```
      IF(LV(J).LT.LV(K)) GO TO 65
      NW=LV(K)
      NR=LV(J)
      IF(NW.LT.ISTART(NR)) ISTART(NR)=NW
   65 CONTINUE
   55 CONTINUE
   25 CONTINUE
C
C     SET UP ID VECTOR AND COMPUTE LENGTH OF STIFFNESS VECTOR
C     ALSO CHECK THAT NBAND DOES NOT DECREASE BY MORE THAN 1 AT A TIME
C
      LEN=NBAND(1)
      DO 50 I=2,LL
      IF(NBAND(I)-NBAND(I-1).LT.0)  NBAND(I)=NBAND(I-1)-1
      LEN=LEN+NBAND(I)
   50 CONTINUE
      ID(1)=1
      DO 60 I=2,LL
   60 ID(I)=ID(I-1)+NBAND(I-1)
      RETURN
      END
      SUBROUTINE QFLUX(NBM,NNODE)
C
C     MODIFIES MATRIX CK BY ADDING THE FLUX TERMS FOR NODES ON WELL BOUNDARY.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      DATA NREAD,NPRINT/1,3/
      IF(NBM.EQ.0) GO TO 20
      NBP=NBM+1
      DO 10 I=NBP,NNODE
      IST=ISTART(I)
      IF(IST.GT.NBM) GO TO 10
      I1=I-1
      DO 30 J=IST,NBM
      K=I1-J+1
      IP=ID(J)+K
      CK(I,1)=CK(I,1)-VK(IP)*CK(J,1)
   30 CONTINUE
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SYMSOL(LL,NLL,NBM)
C
C     AN IN CORE BAND SOLVER.
C     USE IN CONJUNCTION WITH TOPOLOGICAL SUBROUTINE FOR LARGE BANDWIDTH.
C
      DIMENSION VTEMP(90)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /ASOLV/ISTART(300),N(300),IDUM(300)
      COMMON /BSOLV/C(300,1),V(2000)
      DOUBLE PRECISION TEMP
      JBOUN=NBM+1
      NBP=JBOUN
      ID=1
      IF(NBM.EQ.0) GO TO 150
      DO 100 I=1,NBM
      ID=ID+N(I)
  100 CONTINUE
  150 CONTINUE
      DO 10 I=NBP,LL
      TEMP=V(ID)
      NEB=ID+N(I)-1
      ID1=ID+1
      IF(I.EQ.JBD(JBOUN)) GO TO 16
C     NORMALISE ROW I
C
      KK=0
      V(ID)=1.0
      IF(ID1.GT.NEB) GO TO 29
      DO 20 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
   20 V(J)=V(J)/TEMP
   29 CONTINUE
      DO 25 L=1,NLL
      C(I,L)=C(I,L)/TEMP
   25 CONTINUE
      GO TO 46
   16 CONTINUE
      IF(ID1.GT.NEB) GO TO 39
      KK=0
      DO 120 J=ID1,NEB
      KK=KK+1
      VTEMP(KK)=V(J)
  120 V(J)=0.0
   39 CONTINUE
      DO 125 L=1,NLL
  125 C(I,L)=DISP(JBOUN)
```

```
      JBOUN=JBOUN+1
C
C     ELIMINATION
C
   46 CONTINUE
      IDJ=ID
      IF(ID1.GT.NEB)  GO TO 35
      KK=0
      DO 30 J=ID1,NEB
      JI=J-ID1+I
      IDJ=IDJ+N(JI)
      KK=KK+1
      IF(VTEMP(KK)) 50,30,50
   50 CONTINUE
      IF(JBOUN.EQ.1) GO TO 240
      JBM1=JBOUN-1
      IF(I.EQ.JBD(JBM1)) GO TO 140
  240 IDP=J
      DO 40 K=IDP,NEB
      KJ=IDJ+K-J
   40 V(KJ)=V(KJ)-V(K)*VTEMP(KK)
  140 CONTINUE
      NJ=I+J-ID
      DO 32 L=1,NLL
      C(NJ,L)=C(NJ,L)-C(I,L)*VTEMP(KK)
   32 CONTINUE
   30 CONTINUE
   35 CONTINUE
      ID=ID+N(I)
   10 CONTINUE
C
C     BACK SUBSTITUTION
C
      ID=ID-1
      LLM1=LL-1
      LL1=LL-NBP
      DO 70 IB=1,LL1
      I=LLM1-IB+1
      ID=ID-N(I)
      IS=I+1
      IN=I+N(I)-1
      DO 80 J=IS,IN
      NJ=ID+J-I
      DO 75 L=1,NLL
      C(I,L)=C(I,L)-C(J,L)*V(NJ)
   75 CONTINUE
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE SUROUT(KSURF,ESMAX,LVEC,NDFLEX)
C
C     PRINTS OUT FREE SURFACE COORDINATES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,3)
    3 FORMAT(///,10X,'FREE SURFACE POSITON',//,
     1              15X,'RADIUS',10X,' SURFACE HEIGHT',//)
      DO 10 I=1,LVEC
      WRITE(NPRINT,13)RVEC(I),HF(I)
   10 CONTINUE
   13 FORMAT(10X,F10.2,10X,F10.4)
      WRITE(NPRINT,23) KSURF,ESMAX
   23 FORMAT(///,10X,'SURFACE TOLERANCE COUNTER    =',I10,///,
     1            10X,'MAXIMUM ERROR OF SURFACE HEIGHT   =',F12.4)
C
C     PRINT Z-COORDINTE AND HEAD VALUE OF FLEXIBLE NODES.
C
C***  WRITE(NPRINT,33)
      DO 30 I=1,NDFLEX
      J=NVMESH(I)
      XV=X(J,1)
      ZV=X(J,2)
      HV=H(J)
C***  WRITE(NPRINT,43)J,XV,ZV,HV
   43 FORMAT(///,10X,I5,15X,F12.4,2(10X,F12.4))
   33 FORMAT(///,20X,'HEIGHT AND HEAD OF FLEXIBLE NODES',///,
     1            10X,'NODE NUMBER',10X,'R-COORDINATE',10X,'Z-COORDINATE'
     2            ,10X,'HEAD VALUE',//)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE VCHEC3(M,AK,BK,J1,J2,J3,NT,PMK,VCR,GRRX,GRRY)
C
C     COMPUTES ELEMENT VELOCITIES FOR TRIANGULAR ELEMENTS.
C
```

```
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      HX=B(NT,1)*H(J1)+B(NT,2)*H(J2)+B(NT,3)*H(J3)
      HY=C(NT,1)*H(J1)+C(NT,2)*H(J2)+C(NT,3)*H(J3)
      HS= ABS( SQRT(HX**2+HY**2))
      IF(IDARCY(M).EQ.0) GO TO 20
      TEMP=0.5*AK+SQRT(0.25*AK**2+BK*HS)
      VEL(M)=HS/TEMP
      IF(ABS(VEL(M)).LE.VCR) IDARCY(M)=0
      GO TO 30
   20 CONTINUE
      VEL(M)=PMK*HS
      IF(ABS(VEL(M)).GT.VCR) IDARCY(M)=1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SHUF(JF,NBD,NBW,NBDTO)
C
C     SHUFFLES VECTOR JBD AND DISP.
C
      DIMENSION JSEEP(50),DSEEP(50)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      NWID=NBD-NBW
      JS=0
      JST=NBD+1
      JEND=NBDTO
      IF(JF.EQ.2) JST=NBDTO-NWID+1
      DO 10 J=JST,JEND
      JS=JS+1
      JSEEP(JS)=JBD(J)
      DSEEP(JS)=DISP(J)
   10 CONTINUE
      JST=NBW+1
      JEND=NBD
      IF(JF.EQ.2) JEND=NBDTO-NWID
      DO 15 J=JST,JEND
      JS=JS+1
      JSEEP(JS)=JBD(J)
      DSEEP(JS)=DISP(J)
   15 CONTINUE
      JS=0
      NST=NBW+1
      DO 20 J=NST,NBDTO
      JS=JS+1
      JBD(J)=JSEEP(JS)
      DISP(J)=DSEEP(JS)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE ELGND3(NT,SC,M)
C
C     GENERATES ELEMENT MATRIX D FOR TRIANGULAR ELMENTS.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /BELEM/B(150,3),C(150,3),AREA(150)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      IF(NREP(M).EQ.NT) GO TO 50
      NT=NREP(M)
C
C     FORM ELEMENT MATRIX:- D
C
      D(1,1)=0.5*ORX(M)*AREA(NT)*SC/3.
      D(2,2)=D(1,1)
      D(3,3)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      D(1,3)=D(1,2)
      D(3,1)=D(1,3)
      D(2,3)=D(1,2)
      D(3,2)=D(2,3)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE MERBD(N,M)
C
C     MERGES ELEMENT MATRIX D INTO GROSS VECTOR VD.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /AELEM/ELK(3,3,150),EK(3,3)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      DO 10 I=1,N
      IK=NOD(M,I)
      IF(IK.EQ.0)  GO TO 10
      DO 20 J=1,N
      JK=NOD(M,J)
      IF(IK.GT.JK)  GO TO 20
      IPOS=ID(IK)+JK-IK
      VD(IPOS)=VD(IPOS)+D(I,J)
```

```
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE VDFB(LVEC)
C
C     GENERATES VECTOR VDTOP.
C
      DIMENSION D(2,2)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      NELTOP=LVEC-1
      NN=2
      ID(1)=1
      DO 10 I=2,LVEC
   10 ID(I)=ID(I-1)+2
      LENT=2*LVEC-1
      DO 108 I=1,LENT
      VDTOP(I)=0.0
  108 CONTINUE
      DO 208 IE=1,NELTOP
      IN=NDVEC(IE)
      IP=NDVEC(IE+1)
      RAVE=0.5*(X(IP,1)+X(IN,1))
      RDIF=X(IP,1)-X(IN,1)
      D(1,1)=RAVE*RDIF/3.0
      D(2,2)=D(1,1)
      D(1,2)=D(1,1)*0.5
      D(2,1)=D(1,2)
      DO 308 I=1,2
      IK=IE+(I-1)
      DO 408 J=1,2
      JK=IE+(J-1)
      IF(IK.GT.JK) GO TO 408
      IPOS=ID(IK)+JK-IK
      VDTOP(IPOS)=VDTOP(IPOS)+D(I,J)
  408 CONTINUE
  308 CONTINUE
  208 CONTINUE
      RETURN
      END
      SUBROUTINE GVMOD(LACT,SY,IGK)
C
C     MODIFIES GROSS VECTORS VK AND GK TO ACCOUNT FOR LEAKAGE FLUX ACROSS
C     TOP BOUNDARY.
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BSOLV/CK(300,1),VK(2000)
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
C
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      J2=J1
      JM=J1
      IF(L.LT.LACT) J2=NDVEC(L+1)
      IF(L.GT.1) JM=NDVEC(L-1)
      SUM=VF1*HINT(J1,1)+VF2*HINT(J2,1)+VFM*HINT(JM,1)
      IF(IGK.GT.0) GK(J1,1)=GK(J1,1)+SUM
      IF(IGK.GT.0) GO TO 1078
      IS=ID(I)
      VK(IS)=VK(IS)+VF1
      IL=IS+NBAND(I)-1
      VK(IL)=VK(IL)+VF2
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE GKMOD(LACT,SY)
C
C     MODIFIES MATRIX GK TO TAKE ACCOUNT OF LEAKAGE FLUX ACROSS TOP BOUNDARY.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ADPARA/TLESS(300),HLESS(300),GK(300,1),VDTOP(200)
      DO 1078 L=1,LACT
      IP1=(L-1)*2+1
      IP2=IP1+1
      IPM=IP1-1
```

```
      VF1=VDTOP(IP1)*SY
      VF2=0.0
      VFM=0.0
      IF(L.GT.1) VFM=VDTOP(IPM)*SY
      IF(L.LT.LACT) VF2=VDTOP(IP2)*SY
      I=NDVEC(L)
      J1=I
      LM=L
      LP=L
      IF(L.LT.LACT) LP=L+1
      IF(L.GT.1) LM=L-1
      SUM=VF1*QLEAK(L)+VF2*QLEAK(LP)+VFM*QLEAK(LM)
      GK(J1,1)=GK(J1,1)+SUM
 1078 CONTINUE
      RETURN
      END
      SUBROUTINE TIGEN(NTICR,TFACTR,TMUL,DTMUL,RW,PM,SS,IWBC,QFIX,TPAT)
C
C     GENERATES DISCRETE TIME VECTOR,TIME.
C
      COMMON /ATIME/TIME(60),SH(60),DSW(5),QCALC(5)
      SWST=0.8
      CUN=22.*RW**2/(7.*QFIX)
      TIME(1)=TFACTR
      IF(TPAT.EQ.2) GO TO 15
      DO 10 I=2,NTICR
      TIME(I)=TIME(I-1)*TMUL
      TMUL=TMUL+DTMUL
   10 CONTINUE
      GO TO 17
   15 CONTINUE
      DO 20 I=2,NTICR,2
      TIME(I)=TIME(I-1)
      TIME(I+1)=2.*TIME(I)
   20 CONTINUE
      DO 25 I=2,NTICR
      TIME(I)=TIME(I)+TIME(I-1)
   25 CONTINUE
   17 CONTINUE
      RR=RW+1.99
      CONST=RR**2*SS/(4.0*PM)
      IF(SS.LE.0.)CONST=1.0
      IF(IWBC.NE.0)CONST=1000.*CUN*SWST
      DO 30 I=1,NTICR
      TIME(I)=TIME(I)*CONST
   30 CONTINUE
      RETURN
      END
      SUBROUTINE VECMUL(NNODE,LEN)
C
C     MULTIPLIES GROSS VECTOR AND COLUMN MATRIX.
C        VKD  =  VK*D
C
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /CSOLV/VK(2000),D(300,1),IPROP(500),DUM(3,3)
      COMMON /ADPARA/TLESS(300),HLESS(300),VKD(300,1),VDTOP(200)
      DATA NREAD,NPRINT/1,3/
      L=1
      DO 10 I=1,NNODE
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
C
      VKD(I,1) = 0.0
      DO 20 J = IS, IL
      K=I+J-IS
      VKD(I,1)=VKD(I,1)+VK(J)*D(K,L)
   20 CONTINUE
C
      IF(I.EQ.1) GO TO 40
      I1 = I-1
      IST= ISTART(I)
C
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
      VKD(I,1)=VKD(I,1)+VK(IP)*D(J,L)
   30 CONTINUE
   40 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE AQDIS(NLEN,QAQFR,QRDIF,DELT,TH,RCSNG,IQ,QFIX,DWD)
C
C     COMPUTES TOTAL DISCHARGE INTO THE WELL.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
```

```
      QSUM=0.0
      TWPI=44./7.
      CALL QMULT(NLEN,QSUM)
      QSUM=ABS(TWPI*QSUM)/(0.5*DELT)
      QAQFR=QSUM
      ACSNG=0.5*TWPI*RCSNG**2
      TRM=ABS(ACSNG*DWD*2./DELT)
      QCALX=QAQFR+TRM*10.**3
      QRDIF=ABS(QFIX-QCALX)
      QCALC(IQ)=QCALX
      RETURN
      END
      SUBROUTINE QMULT(NBW,QSUM)
C
C     COMPUTES NODAL FLUXES AT WELL BOUNDARY.
C
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      COMMON /ASOLV/ISTART(300),NBAND(300),ID(300)
      COMMON /BQCAL/VKQ(600),QWB(40)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DATA NREAD,NPRINT/1,3/
      DO 10 L=1,NBW
      I=JBD(L)
      IF(L.EQ.NBW) I=L
      IS = ID(I)
      IL = ID(I)+NBAND(I)-1
      QWB(L)=0.0
      DO 20 J = IS, IL
      K=I+J-IS
      QWB(L)=QWB(L)+VKQ(J)*H(K)
   20 CONTINUE
      IF(I.EQ.1) GO TO 15
      I1 = I-1
      IST= ISTART(I)
      DO 30 J= IST,I1
      K  = I1- J+1
      IP =   ID(J) + K
      QWB(L)=QWB(L)+VKQ(IP)*H(J)
   30 CONTINUE
   15 CONTINUE
      QWB(L)=QWB(L)-GP(L)
      QSUM=QSUM+QWB(L)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE FQSET(IT,ITCUR,DELT,FQ,TPAT)
C
C     SETS DRAWDOWN FUNCTION FOR WELL DISCHARGE ITERATION.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      IF(TPAT.EQ.1) GO TO 20
      IF(IT.GE.3) GO TO 10
      IF((IT.EQ.1).AND.(ITCUR.GT.IT)) FQ=SW(IT)
      GO TO 20
   10 CONTINUE
      DO 15 I=1,IT
      HDLTA=0.499*DELT
      IF(HDLTA.GT.TIME(I)) GO TO 15
      FQ=SW(I)
      GO TO 20
   15 CONTINUE
   20 CONTINUE
      RETURN
      END
      SUBROUTINE HREAD(NNODE,L,INDEX)
C
C     READS IN NODAL VALUES AT STARTING TIME.
C
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /WORKA/VWORK(500)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      NCARD=NNODE/6
      NCARDT=NCARD+1
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      READ(L,13)(VWORK(I),I=IST,IEND)
   13 FORMAT(6E13.4)
      IST=IEND+1
   10 CONTINUE
      NREM=NNODE-NCARD*6
      IEND=IST+NREM-1
      READ(L,13)(VWORK(I),I=IST,IEND)
      DO 15 I=1,NNODE
      IF(INDEX.EQ.1) H(I)=VWORK(I)
      IF(INDEX.EQ.2) QLEAK(I)=VWORK(I)
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HOFREE(HO,HSEEP,NRR)
C
C     COMPUTES INITIAL FREE SURFACE.
C
      COMMON /BDISC/RVEC(50),NDVEC(50),XLEN(180)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      HSEEP=HO
      HF(1)=HSEEP
      DO 10 KK=1,NRR
      K1=KK+1
      HF(K1)=HSEEP
   10 CONTINUE
      RETURN
      END
      SUBROUTINE ROUT(NNODE)
C
C     PRINTS OUT NODAL VALUES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /ADPARA/TLESS(300),HLESS(300),VKD(300,1),VDTOP(200)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,53)
   53 FORMAT(///)
      WRITE(NPRINT,3)
      WRITE(NPRINT,13)
      WRITE(NPRINT,23)
      WRITE(NPRINT,13)
      WRITE(NPRINT,3)
    3 FORMAT(5X,50('*'))
   13 FORMAT(5X,'*',48X,'*')
   23 FORMAT(5X,'*',11X,'FINAL RESULTS OF ANALYSIS',12X,'*')
      WRITE(NPRINT,33)
      DO 10 I=1,NNODE
      WRITE(NPRINT,43)I,X(I,1),X(I,2),H(I),TLESS(I),HLESS(I)
   10 CONTINUE
   33 FORMAT(///,20X,'HEAD .VS. RADIUS AND 1/U .VS. W(U)',///,
     1          10X,'NODE',10X,'R-COORD',10X,'Z-COORD',10X,'HEAD',14X,
     2          '1/U',19X,'W(U)',//)
   43 FORMAT(10X,I3,2(6X,F10.2),8X,F10.4,7X,E11.4,10X,F10.4)
      RETURN
      END
      SUBROUTINE HPUNCH(NNODE,L)
C
C     PUNCHES OUT NODAL VALUES AT FINAL TIME.
C
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      NCARD=NNODE/6
      NCARDT=NCARD+1
      WRITE(3,23)NCARDT
   23 FORMAT(///,10X,'TOTAL NUMBER OF PUNCHED CARDS    =',I10)
      IST=1
      DO 10 J=1,NCARD
      IEND=IST+5
      WRITE(L,13)(H(I),I=IST,IEND)
   13 FORMAT(6E13.4)
      IST=IEND+1
   10 CONTINUE
      NREM=NNODE-NCARD*6
      IEND=IST+NREM-1
      WRITE(L,13)(H(I),I=IST,IEND)
      RETURN
      END
      SUBROUTINE HMEANV(LVEC)
C
C     AVERAGES NODAL HEAD VALUES OVER VERTICAL SECTIONS.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      WRITE(NPRINT,3)
    3 FORMAT(///,20X,'MEAN HEAD VS. RADIAL DISTANCE',//,
     1          10X,'RADIAL DISTANCE',20X,'MEAN HYDRAULIC HEAD',//)
      NBOT=1
      DO 10 I=1,LVEC
      NTOP=NDVEC(I)
      SUM=0.0
      DO 20 J=NBOT,NTOP
      JM=J-1
      JP=J+1
      IF(JM.LT.NBOT) JM=NBOT
      IF(JP.GT.NTOP) JP=NTOP
      XD1=(X(JP,2)-X(J,2))*0.5
      XD2=(X(J,2)-X(JM,2))*0.5
      SUM=SUM+H(J)*(XD1+XD2)
   20 CONTINUE
      HF(I)=SUM/(X(NTOP,2)-X(NBOT,2))
      WRITE(NPRINT,13)RVEC(I),HF(I)
   13 FORMAT(10X,F15.2,20X,F15.4)
```

```
      NBOT=NTOP+1
   10 CONTINUE
      RETURN
      END
      SUBROUTINE BSIMP(NNODE,TH,DINDEX,SY,DELT,IT,ITMIN)
C
C     EVALUATES BOULTON'S CONVOLUTIONAL INTEGRAL BY SIMPSON'S 1/3 RULE.
C
      DIMENSION FN(3)
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /VCOM/ORX(500),VEL(500),HDUM(500),IDARCY(500)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),H(50),HINT(50,1),GP(40)
C
      DOUBLE PRECISION DEXP,ARGU,FN,QCUNT
C
      ITM=IT-1
      IF(ITM.EQ.0) GO TO 5
      TM=TIME(ITM)
      TOLD=(TIME(IT)+TIME(ITM))*0.5
      GO TO 15
    5 TM=0.0
      TOLD=0.5*TIME(IT)
   15 TMM=(TIME(IT)+TIME(IT+1))*0.5
C
      DO 20 II=1,3
      TC=TM+DELT*(II-1)*0.5
      ARGU=-(TMM-TC)*DINDEX
      FN(II)=DEXP(ARGU)
   20 CONTINUE
      ARGU=-DINDEX*(TMM-TOLD)
      DO 30 I=1,NNODE
      IF(IT.GT.ITMIN) GO TO 40
      T1=2.*FN(1)*(HINT(I,1)-HCOLD(I))
      T2=4.*FN(2)*(H(I)-HCOLD(I))
      T3=2.*FN(3)*(H(I)-HINT(I,1))
      DQ   =T1+T2+T3
      GO TO 50
   40 CONTINUE
      DQ   =(H(I)-HINT(I,1))*(FN(1)+4.*FN(2)+FN(3))
   50 CONTINUE
      DQ=-DQ
      QCUNT=QLEAK(I)*DEXP(ARGU)
      QLEAK(I)=QCUNT+DQ
      IF(DINDEX.LE.0.) QLEAK(I)=DQ
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SWMOD(IQ,IT,RCSNG,DLTA,QFIX,QAQFR,SWOLD,DSOLD,TPAT,FQ,
     1HO)
C
C     ADJUSTS THE VALUE OF WELL DRAWDOWN.
C
      COMMON /ATIME/TIME(60),SW(60),DSW(5),QCALC(5)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),HF(50),FINT(50,1),GP(40)
      DELT=DLTA/10.**3
      ACSNG=22.0*RCSNG**2/7.0
      IF(IQ.NE.1) GO TO 15
      IF(ACSNG.GT.0.0) GO TO 35
C***  IF(IT.GT.5) GO TO 10
      DSW(IQ)=-DSOLD
      GO TO 20
   35 CONTINUE
      IF(IT.GT.5) GO TO 10
      QRATIO=QAQFR/QFIX
      IF(ABS(QRATIO).GE.0.8) GO TO 10
      TMM=TIME(IT)-0.5*DLTA
      FACTR=0.5*DELT/ACSNG
      QDEL=QFIX-QAQFR*0.95*TIME(IT)/TMM
      IF(ABS(QRATIO).GE.0.5) QDEL=QFIX-QAQFR
      DSW(IQ)=FACTR*QDEL
      DSW(IQ)=-DSW(IQ)
      GO TO 20
   10 CONTINUE
      TOLD=TIME(IT-2)
      ARGL1=(TIME(IT)-0.5*DLTA)/TOLD
      ARGL2=TIME(IT-1)/TOLD
      TLOG=ALOG(ARGL1)/ALOG(ARGL2)
      DSW(IQ)=(SW(IT-1)-SW(IT-2))*(1.-TLOG)
      DSW(IQ)=-DSW(IQ)
      IF(DSW(IQ).GT.0.) DSW(IQ)=0.0
      GO TO 20
   15 CONTINUE
      IF(IQ.GT.2) GO TO 25
      IF(TPAT.EQ.2) GO TO 62
      TERM1=(1.-QFIX/QCALC(IQ-1))*HO**2
      TERM2=QFIX/QCALC(IQ-1)*SW(IT)**2
      FT=SQRT(TERM1+TERM2)
      DSW(IQ)=FT-SWOLD
      GO TO 20
   62 CONTINUE
```

```
      IF((ACSNG.LE.0.).AND.(FQ.LT.0.)) GO TO 40
      DSW(IQ)=DSW(1)*QFIX/QCALC(1)
      GO TO 20
   40 CONTINUE
      SWCOR=SW(IT)-FQ*(QCALC(1)-QFIX)/QFIX
      DSW(IQ)=SWCOR-SWOLD
      GO TO 20
   25 CONTINUE
      DDSW=DSW(IQ-1)-DSW(IQ-2)
      TERM1=QFIX-QCALC(IQ-1)
      TERM2=QCALC(IQ-1)-QCALC(IQ-2)
      DQR=TERM1/TERM2
      DSW(IQ)=DSW(IQ-1)+DDSW*DQR
   20 CONTINUE
      SW(IT)=DSW(IQ)+SWOLD
      DSOLD=-DSW(IQ)
      RETURN
      END
      SUBROUTINE HOUT(NNODE,HO,HW,RO)
C
C     PRINTS OUT NODAL VALUES.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /VCOM/ORX(500),VEL(500),H(500),IDARCY(500)
      DATA NREAD,NPRINT/1,3/
      WRITE(NPRINT,53)
   53 FORMAT(///)
C
      WRITE(NPRINT,423)
      DO 10 I=1,NNODE
      RHO=X(I,1)/RO
      TZI=X(I,2)/HO
      SDLESS=(HO-H(I))/(HO-HW)
      WRITE(NPRINT,953)I,X(I,1),X(I,2),H(I),RHO,TZI,SDLESS
   10 CONTINUE
  953 FORMAT(5X,I3,2(6X,F12.2),4(8X,F10.4))
  423 FORMAT(///,5X,'NODE',5X,'R-COORDINATE',5X,'Z-COORDINATE',9X,
     1'HEAD VALUE',9X,'RHO-COORD',9X,'TZI-COORD',5X,'DIM. DRAWDOWN',//)
      RETURN
      END
      SUBROUTINE FOMESH(LVEC,NDFLEX,HEIGHT)
C
C     RESETS TRIAL POSITION OF THE FREE SURFACE.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /AFREE/HF(50),NVMESH(120),NVELM(150)
      COMMON /ALEAK/QLEAK(50),HCOLD(50),ZF(50),FINT(50,1),GP(40)
      NST=1
      DO 10 I=1,LVEC
      RR=RVEC(I)
      J=NDVEC(I)
      HSTOR=HF(I)
      HF(I)=ZF(I)
      X(J,2)=HF(I)
C
C     ADJUSTS Z-COORDINATES OF VARIABLE NODES.
C
      NDT=NVMESH(NST)
      XCC=X(NDT,1)+0.01
      IF(RR.GT.XCC) GO TO 10
      ZDIF=X(NDT,2)-X(NDT-1,2)
      DXRTIO=(HF(I)-HEIGHT)/(HSTOR-HEIGHT)
      IF(NST.EQ.NDFLEX) GO TO 10
      DO 19 K=NST,NDFLEX
      NK=NVMESH(K)
      XK=X(NK,1)-0.01
      IF(XK.GT.RR) GO TO 29
      ZSTOR=X(NK,2)
      X(NK,2)=X(NK-1,2)+DXRTIO*ZDIF
   19 CONTINUE
   29 CONTINUE
      NST=K
      IF(NST.GT.NDFLEX) NST=NDFLEX
      X(J,2)=HF(I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE BMDFIX(HW,HO,NNODE,NBD,NDW,NDRO,NBW,NSCREN,RW)
C
C     LOCATE NODES WHERE NODAL HEAD VALUES ARE PRESCRIBED.
C
      COMMON /ADISC/X(300,2),NOD(500,3),NREP(500)
      COMMON /BDISC/RVEC(50),NDVEC(50),JBD(90),DISP(90)
      COMMON /WSCREN/XSCR(5),HSCR(5)
      K=0
      NFND=1
      DO 30 I=1,NSCREN
      XST=XSCR(I)-0.01
      XEND=XST+HSCR(I)+0.02
```

```
      DO 40  J=1,NNODE
      L=NFND+J-1
      IF(L.EQ.1) GO TO 222
      IF(X(L,1).GT.RW) GO TO 30
  222 CONTINUE
      IF((X(L,2).LT.XST).OR.(X(L,2).GT.XEND)) GO TO 70
   22 K=K+1
      JBD(K)=L
      DISP(K)=HW
      GO TO 40
   70 IF(X(L,2).GT.XEND) GO TO 30
   40 CONTINUE
      NFND=L+1
   30 CONTINUE
      J=K
      NBW=J
      NST=NNODE-NDRO+1
      DO 25  I=NST,NNODE
      J=J+1
      JBD(J)=I
      DISP(J)=HO
   25 CONTINUE
      NBD=J
      RETURN
      END
      SUBROUTINE HYPROP(I,AK,BK,PMK,VCR,AGP,BGP,PMGP,VGP,AL,BL,PML,VCL)
C
C     IDENTIFIES HYDRAULICS PROPERTIES OF EACH ELEMENT.
C
      COMMON /CSOLV/VD(2000),HINT(300,1),IPROP(500),D(3,3)
      IF(IPROP(I).EQ.0) GO TO 10
      AL=AK
      BL=BK
      PML=PMK
      VCL=VCR
      GO TO 20
   10 CONTINUE
      AL=AGP
      BL=BGP
      PML=PMGP
      VCL=VGP
   20 CONTINUE
      RETURN
      END
```

5. MODIFICATIONS

5.1 EXTENSION OF COMMON BLOCKS TO SOLVE LARGER PROBLEMS

In using the two-dimensional flow programs, a situation may arise where it becomes necessary to increase the sizes of some COMMON blocks in the main programs and subroutines to cater for larger values of the following variables:

(i) NLAYR, which denotes the number of layers in the multi-layer system (Maximum value allowable in the current listings is 3)

(ii) NNODE, which denotes the number of nodes in the entire finite element network (Maximum value allowable in the current listings is 300)

(iii) NELEM, which denotes the total number of elements in the network (Maximum value allowable in the current listings is 500 in programs STCON3, TRCON3, STFREE and TRFREE, and 200 in STCOND and TRCOND)

(iv) NT, which denotes the number of elements having different geometry (Maximum value allowable in the current listings is 150)

(v) LEN, which denotes the length of gross vectors formed by assembling all elements in the network (Maximum value allowable in the current listings is 2000)

For convenience in modifying the programs, the COMMON statements which must be altered in their main programs are written in the following manner:

(i) M/PROG. OF STCON3

```
COMMON /ADISC/X(NNODE,2), NOD(NELEM,3),NREP(NELEM)
COMMON /ASOLV/ISTART(NNODE),NBAND(NNODE),ID(NNODE)
COMMON /CSOLV/VD(LEN),HINT(NNODE,1),IPROP(NELEM),D(3,3)
COMMON /BSOLV/CK(NNODE,1),VK(LEN)
```

```
COMMON /ADPARA/TLESS(NNODE),HLESS(NNODE),GK(NNODE,1),VDTOP(200)
COMMON /AELEM/ELK(3,3,NT),EK(3,3)
COMMON /BELEM/B(NT,3),C(NT,3),AREA(NT)
COMMON /VCOM/ORX(NELEM),VEL(NELEM),H(NELEM),IDARCY(NELEM)
```

(ii) <u>M/PROG. OF TRCON3</u>

```
DIMENSION JDARCY(NELEM)
COMMON /ALAYR/AKL(NLAYR),BKL(NLAYR),VCRL(NLAYR),SSL(NLAYR),
THL(NLAYR),IREGL(NLAYR),NMINL(NLAYR),NFRL(NLAYR),PML(NLAYR)
COMMON /AELEM/ELK(3,3,NT),EK(3,3)
COMMON /BELEM/B(NT,3),C(NT,3),AREA(NT)
COMMON /BSOLV/CK(NNODE,1),VK(LEN)
COMMON /ASOLV/ISTART(NNODE),NBAND(NNODE),ID(NNODE)
COMMON /CSOLV/VD(LEN),HINT(NNODE,1),IPROP(NELEM),D(3,3)
COMMON /VCOM/ORX(NELEM),VEL(NELEM),H(NELEM),IDARCY(NELEM)
COMMON /ADISC/X(NNODE,2),NOD(NELEM,3),NREP(NELEM)
COMMON /ACORE/VCORE(LEN)
COMMON /ADPARA/TLESS(NNODE),HLESS(NNODE),GK(NNODE,1)
VDTOP(200)
```

(iii) <u>M/PROG. OF STCOND</u>

```
COMMON /ALAYR/AKL(NLAYR),BKL(NLAYR),VCRL(NLAYR),SSL(NLAYR),
THL(NLAYR),IREGL(NLAYR),NMINL(NLAYR),NFRL(NLAYR),PML(NLAYR)
COMMON /ADISC/X(NNODE,2),NOD(NELEM,6),NREP(NELEM),
ITYPE(NELEM)
COMMON /AELEM/BA(NT),SLX(NT,6),SLY(NT,6),SELK(6,6),EK(6,6)
COMMON /CELEM/ELK(3,3,NT),B(NT,3),C(NT,3),AREA(NT)
COMMON /ASOLV/ISTART(NNODE),NBAND(NNODE),ID(NNODE)
COMMON /BSOLV/CK(NNODE,1),VK(LEN)
COMMON /CSOLV/VD(LEN),HINT(NNODE,1),IPROP(NELEM)
COMMON /VCOM/ORX(NELEM),VEL(NELEM),H(NNODE),IDARCY(NELEM),
D(6,6)
COMMON /ADPARA/TLESS(NNODE),HLESS(NNODE),GK(NNODE,1),
VDTOP(200)
```

(iv) M/PROG. OF TRCOND

```
COMMON /ACORE/VCORE(LEN)
COMMON /ALAYR/AKL(NLAYR),BKL(NLAYR),VCRL(NLAYR),SSL(NLAYR),
THL(NLAYR),IREGL(NLAYR),NMINL(NLAYR),NFRL(NLAYR),PML(NLAYR)
COMMON /ADISC/X(NNODE,2),NOD(NELEM,6),NREP(NELEM),
ITYPE(NELEM)
COMMON /AELEM/BA(NT),SLX(NT,6),SLY(NT,6),SELK(6,6),EK(6,6)
COMMON /CELEM/ELK(3,3,NT),B(NT,3),C(NT,3),AREA(NT)
COMMON /ASOLV/ISTART(NNODE),NBAND(NNODE),ID(NNODE)
COMMON /BSOLV/CK(NNODE,1),VK(LEN)
COMMON /CSOLV/VD(LEN),HINT(300,1),IPROP(NELEM)
COMMON /VCOM/ORX(NELEM),VEL(NELEM),H(NNODE),IDARCY(NNODE),
D(6,6)
COMMON /ADPARA/TLESS(NNODE),HLESS(NNODE),GK(NNODE,1),
VDTOP(200)
```

(v) M/PROG. OF STFREE

```
COMMON /ADISC/X(NNODE,2),NOD(NELEM,3),NREP(NELEM)
COMMON /AELEM/ELK(3,3,NT),EK(3,3)
COMMON /BELEM/B(NT,3)C(NT,3),AREA(NT)
COMMON /ASOLV/ISTART(NNODE),NBAND(NNODE),ID(NNODE)
COMMON /BSOLV/CK(NNODE,1), VK(LEN)
COMMON /CSOLV/VD(LEN),HINT(NNODE,1)IPROP(NELEM),
D(3,3)
COMMON /VCOM/ORX(NELEM),VEL(NELEM),H(NELEM),IDARCY(NELEM)
```

(vi) M/PROG. OF TRFREE

```
COMMON /ADISC/X(NNODE,2),NOD(NELEM,3),NREP(NELEM)
COMMON /AELEM/ELK(3,3,NT),EK(3,3)
COMMON /BELEM/B(NT,3),C(NT,3),AREA(NT)
COMMON /ASOLV/ISTART(NNODE),NBAND(NNODE),ID(NNODE)
COMMON /BSOLV/CK(NNODE,1)VK(LEN)
COMMON /CSOLV/VD(LEN),HINT(NNODE,1),IPROP(NELEM),D(3,3)
COMMON /VCOM/ORX(NELEM),VEL(NELEM),H(NELEM),IDARCY(NELEM)
COMMON /ACORE/VCORE(LEN),VKORE(LEN)
COMMON /ADPARA/TLESS(NNODE),HLESS(NNODE),GK(NNODE,1),
VDTOP(200)
```

Modification of a particular main program is accomplished by inserting appropriate values of the variables NNODE, NELEM, LEN and NT in the above COMMON statements. Modification of the subroutines containing any of the COMMON blocks is accomplished in the same manner if the array variables in the blocks correspond to those listed in the main program, under the same COMMON block names. If the variables do not correspond to those in the main program, the COMMON blocks in the subroutines should be extended in such a way that the total length of each block is equal to the length of the corresponding block (under the same COMMON name).

For example in subroutine GXNOD of Program STCON3. Suppose the statement

```
COMMON /AELEM/ELK(3,3,150),EK(3,3)
```

is modified to

```
COMMON /AELEM/ELK(3,3,300),EK(3,3)
```

The corresponding COMMON statement in the subroutine GXNOD should be altered

<u>from</u>

```
COMMON /AELEM/NTRAN(3,90),NH1(3,90),NH2(3,90),NDB1(3,90),
NDB2(3,90),NDWB(9)
```

<u>to</u>

```
COMMON /AELEM/NTRAN(5,90),NH1(5,90),NH2(5,90),NDB1(5,90),
NDB2(5,90),NDWB(459)
```

It may be noted the total length of this COMMON block is increased from the previous block length of 3 x 3 x 150 + 9 = 909 to the new block length of 3 x 3 x 300 + 9 = 5(5 x 90) + 459 = 2709.

The remaining task is left to the user.

5.2 PROBLEMS INVOLVING FLOW THROUGH ANISOTROPIC MATERIALS

The current versions of the two-dimensional programs are being modified to handle problems involving flow through anisotropic materials. Only anisotropic Darcy flow can be analysed at this stage as there is still a lack of information on the non-linear constitutive relations for anisotropic materials. It is assumed that the directions of the two principal coefficients of permeability coincide with the horizontal and vertical axes.

Modification of the programs involves

(i) Alteration of the READ statements for input of material data

(ii) Alteration of a number of statements in the subroutines which generate element matrices and the subroutines which compute the element velocities

(iii) Alteration of the CALL statements in the main programs which invoke the modified subroutines.

5.3 MODIFICATION OF PROGRAMS TRCON3, TRCOND AND TRFREE TO START THE SOLUTION AT ITST ≥ 2

The programs TRCON3, TRCOND and TRFREE can also be used, with slight modification, to solve

(i) the transient flow problems starting from the second or later time step at which the flow system is not at rest. (i.e. the hydraulic heads at all points in each layer are not equal to HO)

(ii) problems in which the initial heads in separate layers differ.

For case (i) the appropriate value greater than or equal to 2 should be specified for ITST, and for case (ii) ITST should be set equal to 2.

The modification is accomplished by replacing the boxed statements on pages 176, 222 and 273 by the following statements:

```
     READ(NREAD,6343)SWOLD,DSOLD,QAQFR
6343 FORMAT(3F10.3)
```

where in Programs TRCON3 and TRCOND

SW = absolute value of the well drawdown at the beginning of the starting time step

DSOLD = absolute value of the drawdown increment at the beginning of the starting time step

QAQFR = value of the total discharge from aquifer(s) into well at the beginning of the starting time step and where in program TRFREE

SWOLD = height of the water level in well at the beginning of the starting time step

DSOLD = increment of SWOLD

QAQFR = value of the discharge from the unconfined aquifer at the beginning of the starting time step.

In running the programs, input data additional to the problem data listed in the input data forms (see pages 61, 78 and 108) must be supplied by the user. The additional data cards should be inserted in the following order after the problem data cards:

(i) Nodal values of the hydraulic head at the beginning of the starting time step. The data cards are to be punched according to the following formats:

NNODE (I10)

H(1) H(2) H(3) H(4) H(5) H(6)

H(7) H(8) H(9) H(10) H(11) H(12)

-- H(NNODE)

(6E10.4)

(ii) Values of the flux for the top boundary nodes. The data cards are to be punched according to the following formats:

LVEC

(I10)

QLEAK(1)	QLEAK(2)	QLEAK(3)	QLEAK(4)	QLEAK(5)	QLEAK(6)
QLEAK(7)	QLEAK(8)	QLEAK(9)	QLEAK(10)	QLEAK(11)	QLEAK(12)
----------	----------	----------	----------	QLEAK(LVEC)	

6(E10.4)

(iii) Values of the variables SWOLD, DSOLD and QAQFR. The data card is to be punched according to

SWOLD	DSOLD	QAQFR

(3F10.3)

6. INPUT DATA FOR ADDITIONAL FLOW PROBLEMS

3	1	0	1.50		Specification Card
1.00	1000.00	0.00	50.00	0.20	Problem 1
10.000	20.000	0.060			
0	0	0			
100.00					
2	2.00	0.50			
0.50	1000.00	0.00	50.00	0.20	Problem 2
10.000	20.000	0.060			
0	1	0			
0.0010	50.00				
100.00					
2	2.00	0.50			
0.50	1000.00	0.00	50.00	0.20	Problem 3
10.000	20.000	0.060			
1	1	1			
0.060	0.040	0.010	0.500		
0.0010	50.00				
-50.00					
2	2.00	0.50			

Figure 27: Input data for STCON1

3	0	1	1.50		Specification Card
0.50	1000.00	0.00	20.00	0.20	Problem 1
280.00					
0	0	0	0	0	
10.000	20.000	0.060	0.10E-04		
10	500.000	1.500	0.000		
3	2.00	0.50			
0.50	1000.00	0.00	20.00	0.20	Problem 2
280.00					
0	0	1	0	1	
1.00	0.04				
10.000	20.000	0.060	0.10E-04		
0.060	0.050	0.010	0.10E-04	0.50	
10	50.000	1.500	0.000		
2	2.00	0.50			
0.25	1000.00	0.00	20.00	0.20	Problem 3
100.00					
1	0	0	1	0	
10.000	20.000	0.060	0.10E-04		
0.10E 00	0.10E-01				
10	100.000	1.500	0.000		
2	2.00	0.25			

Figure 28: Input data for TRCON1

3	1	1	1.50		Specification Card
0.50	100.00	0.00	20.00	0.20	Problem 1
0	0				
10.000	20.000	0.060			
-50.00					
0.50	1.50	200.00	2	2	
1	6	1			
10.00	10.00				
0.50	1000.00	0.00	20.00	0.20	Problem 2
1	0				
10.000	20.000	0.060			
0.060	0.050	0.010	0.500	10.000	
-50.00					
0.50	1.50	200.00	2	2	
0	5	1			
10.00	10.00				
0.50	1000.00	0.00	40.00	0.20	Problem 3
0	1				
10.000	20.000	0.060			
0.0010	20.00				
-50.00					
0.50	1.50	200.00	2	2	
0	11	2			
0.00	10.00				
20.00	10.00				

Fig. 29: Input data for STCON3

2	0	0	1.50		Specification Card
0.50	5000.00	0.00	0.20	2	
100.00	1.00	0.01			
0	1	0	1	0	
0.40E 02	0.10E 02	0.60E 02	0.60E-01	0.10E-04	
0.20E 02	0.10E 03	0.00E 00	0.00E 00	0.10E-03	Problem 1
10	1	1000.00	1.50	0.00	
0.50	1.50	500.00			
3	1				
2	2	5			
4	3	5			
0.00	20.00				
1.00	5000.00	0.00	0.20	3	
100.00	1.00	0.01			
0	0	0	1	1	
0.20E 02	0.10E 02	0.20E 02	0.60E-01	0.10E-04	
0.20E 02	0.10E 04	0.00E 00	0.00E 00	0.10E-03	
0.20E 02	0.10E 02	0.20E 02	0.60E-01	0.10E-04	
0.10E 00	0.10E-01				Problem 2
10	1	4000.00	1.50	0.00	
0.50	1.50	500.00			
6	2				
2	2	3			
5	3	5			
2	2	3			
0.00	20.00				
40.00	20.00				

Figure 30: Input data for TRCON3

3	1	1	1.50		Specification Card
0.50	100.00	0.00	0.20	2	
0	1	0			
20.000	10.000	20.000	0.060		
20.000	1000.000	0.000	0.000		
-50.00					Problem 1
0.50	1.50	20.00			
5	1				
2	2	5			
2	3	5			
0.00	20.00				
0.50	1000.00	0.00	0.20	1	
0	1	1			
20.000	10.000	20.000	0.060		
0.001	40.000				Problem 2
-50.00					
0.50	1.50	200.00			
5	1				
2	2	9			
5.00	10.00				
0.50	1000.00	0.00	0.20	2	
1	2	0			
20.000	100.000	0.000	0.000		
20.000	10.000	20.000	0.060		
0.060	0.050	0.010	0.500	20.000	Problem 3
-50.00					
0.50	1.50	200.00			
5	1				
2	3	5			
2	2	5			
20.00	20.00				

Figure 31: Input data for STCOND

2	0	0	1.50		Specification Card
0.50	5000.00	0.00	0.20	2	
100.00	1.00	0.01			
1	1	0	2	1	
0.40E 02	0.10E 02	0.20E 02	0.60E-01	0.10E-04	
0.20E 02	0.10E 03	0.00E 00	0.00E 00	0.10E-03	
0.100	0.100	0.010	0.500	20.000	Problem 1
0.20E 00	0.10E-01				
10	1	1000.00	1.50	0.00	
0.50	1.50	500.00			
3	1				
6	3	5			
5	2	3			
20.00	20.00				
0.50	5000.00	0.00	0.20	1	
100.00	1.00	0.01			
0	0	1	1	0	
0.40E 02	0.10E 02	0.20E 02	0.60E-01	0.10E-04	Problem 2
100.000	50.000	0.010	0.500	20.000	
20	1	50.00	1.50	0.00	
0.50	1.50	20.00			
5	1				
2	3	2			
20.00	20.00				

Figure 32: Input data for TRCOND

3	1	1	1.50		Specification Card
0.50	1000.00	50.00	50.00	0.20	
1	2				
10.000	20.000	0.060			
0.060	0.040	0.010	0.500	0.000	Problem 1
20.00					
0.50	1.50	200.00	2	2	
11	1				
0.00	50.00				
0.50	1000.00	50.00	50.00	0.20	
1	1				
10.000	20.000	0.060			
50.000	100.000	0.100	2.000	0.000	Problem 2
25.00					
0.45	1.50	200.00	2	2	
5	-1				
0.00	20.00				
0.50	1000.00	50.00	50.00	0.20	
0	2				
10.000	20.000	0.060			
20.00					Problem 3
0.45	1.50	200.00	2	2	
5	1				
15.00	20.00				

Figure 33: Input data for STFREE

2	0	0	1.50			Specification Card
0.50	1000.00	50.00	50.00	0.20		
100.00	1.00	0.01				
0.10E 02	0.20E 02	0.60E-01	0.10E-04	0.20E 00	0.10E-01	
1	1	1	2			
0.060	0.050	0.010	0.500	0.000		Problem 1
10	1	50.00	2.00	0.00		
0.50	1.50	200.00	2	2		
11	1					
0.00	50.00					
0.50	1000.00	50.00	50.00	0.20		
100.00	0.50	0.01				
0.10E 02	0.20E 02	0.60E-01	0.00E 00	0.20E 00	0.00E 00	
1	1	0	1			Problem 2
0.060	0.050	0.010	0.500	0.000		
10	1	500.00	2.00	0.00		
0.50	1.50	500.00	2	2		
5	1					
0.00	20.00					

Figure 34: Input data for TRFREE

LIST OF REFERENCES

1. Dudgeon, C.R. Huyakorn, P.S. and Swan, W.H.C. — Hydraulics of flow near wells in unconsolidated sediments, vol. 1 and 2, University of New South Wales, Water Research Laboratory Report No. 126, (1972).

2. Huyakorn, P.S. and Dudgeon, C.R. — Groundwater and well hydraulics;- An annotated bibliography, University of New South Wales, Water Research Laboratory Report No. 121, (1972).

3. Huyakorn, P.S. — Finite element solution of two-regime flow towards wells, Thesis submitted to the University of New South Wales, Australia, in partial fulfilment of the requirements for the degree of Doctor of Philosophy, (1973).

4. Pilgrim, D.H. — Digital models in groundwater studies , Proc. of Groundwater Symposium, Aug. 28-29, 1969, edited by Watson, K.K., University of New South Wales, Water Research Laboratory Report No. 113, (1970).

5. Prickett, T.A. and Lonnquist, C.G. — Selected digital computer techniques for groundwater resource evaluation, Illinois State Water Survey, Urbane, Illinois, Bull. 55, (1971).

6. A.S.C.E. - Committee on Computer Application of the Soil Mechanics and Foundation Division - Subcommittee on Program Documentation. — Engineering computer program documentation standards, A.S.C.E. Jour., Vol. 99, No. SM3, March (1973).

LIST OF FIGURES

LIST OF FIGURES (Cont'd.)