240
3-15-85 JS(3)

Energy

**F
O
S
S
I
L**

DR- 0867-2

DOE/BC/10215-30
(DE85000118)

NUMERICAL MODEL OF MASSIVE HYDRAULIC FRACTURE
Final Report

By
I. D. Palmer
H. R. Craig
C. T. Luiskutty

March 1985
Date Published

Work Performed Under Contract No. AS19-80BC10215

Oral Roberts University
Tulsa, Oklahoma

MASTER

Technical Information Center
Office of Scientific and Technical Information
United States Department of Energy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

NUMERICAL MODEL OF MASSIVE HYDRAULIC FRACTURE
FINAL REPORT

I. D. Palmer, Principal Investigator
H. R. Craig
C. T. Luiskutty

Oral Roberts University
7777 South Lewis
Tulsa, Oklahoma 74171

Work Performed for the Department of Energy
Under Contract No. DE-AS19-80BC10215

Keith Westhusing, Technical Project Officer
Bartlesville Project Office
Department of Energy
Bartlesville, Oklahoma 74005

Date Published--March 1985

ABSTRACT

This project has involved development of a hydraulic fracture simulator which calculates fracture height as a function of distance from the wellbore in a situation in which a payzone is bounded by two zones in which the minimum in-situ stress is higher (the fracture is vertical). The fracture must be highly elongated (length/height ratio $\gtrsim 4$) and variations in elastic modulus across zones are ignored. First, we describe the leakoff and spurt loss calculations employed in the modeling. Second, we discuss a revised version of the vertically symmetric simulator (bounding zone stresses equal). The addition of non-Newtonian flow and leakoff (including spurt loss) are described in detail. An illustrative result is given. Third, we describe in detail the vertically asymmetric simulator (bounding zone stresses not equal). To illustrate the last results, we present design calculations for a 30,000 gallon fracture, which was the first stimulation in the Multi-Well Experiment. The 80 ft fracture interval in the Paludal zone has at its upper edge a 520 psi stress contrast, and at its lower edge a 1195 psi contrast. Computed fracture height growth above and below the perforated interval, bottomhole pressure, and width profiles in vertical sections are displayed. Comparison is made with diagnostic measurements of fracture length, height, and bottomhole pressure. The appropriate computer codes are included in this report.

Significance of Project

The result of this project is the first part of a variable height fracture simulator. For a vertical fracture initiated in a pay zone bounded by zones of higher in situ stress, the fracture's growth in the vertical direction is restricted. If the fracture is highly elongated, the enclosed computer program calculates fracture height, width, bottomhole pressure, and time, for a particular fracture length.

Other features of the fracture calculation are: (i) the upwards fracture growth can be different from the downwards growth, (ii) non-Newtonian (power law) flow of fracturing fluid, (iii) leakoff and spurt loss are included. Limitations are: (i) no time variations in pump rate or injected fluid viscosity, (ii) no proppant transport. The shape and size of the fracture calculated by our model are in basic agreement with the variable height models of Amoco and Terra-Tek. However, our model takes less than 5 minutes of CP time on a Data-General Eclipse mini-computer.

The model can be used to predict the growth (height, length, and width) of a vertical fracture as a function of time. The bottomhole pressure variation with time can be predicted too. This is a very cheap way to predict the height and pressure of a hydraulic fracture, given the pumping rate, fluid viscosity, and appropriate formation parameters, including stress contrast between pay and bounding zones.

TABLE OF CONTENTS

	<u>Page</u>
Introduction	1
Construction of fracture simulators	3
Improved version of vertically symmetric 3-layer simulator	6
New vertically asymmetric 3-layer simulator	7
Conclusion	12
Acknowledgment	16
Nomenclature	17
References	18
Appendix A--Width calculations for symmetrical fracture	23
Appendix B--Non-Newtonian flow in pressure gradient equation	25
Appendix C--Revised leakoff and spurt calculation . . .	26
Appendix D--Width calculations for asymmetrical fracture	31
Appendix E--Fracture expansion criterion for asymmetrical fracture	38
Appendix F--Fortran code and user instructions for vertically symmetric simulator	40
Appendix G--Fortran code and user instructions for vertically asymmetric simulator	86

LIST OF TABLES

<u>Table I</u>	Parameters used to illustrate vertically symmetric simulator
<u>Table II</u>	Predictions of symmetric fracture simulator for parameters in Table I
<u>Table III</u>	Parameters used to illustrate vertically asymmetric simulator--first MWX stimulation
<u>Table IV</u>	Predictions of asymmetric fracture simulator, and comparison with field measurements, in first MWX stimulation

LIST OF FIGURES

- Figure 1 Geometry and parameters of the model.
- Figure 2 Fracture shape computed by the vertically symmetric simulator for the parameters in Table I.
- Figure 3 Bottomhole pressure as a function of time for the parameters of Table I.
- Figure 4 Fracture widths in a vertical section at the wellbore for the parameters of Table I.
- Figure 5 Fracture shape computed by the vertically asymmetric simulator for the first MWX stimulation (Table II).
- Figure 6 Pressure profile predicted for the first MWX stimulation.
- Figure 7 Prediction of fracture width profile at wellbore for first MWX stimulation.
- Figure 8 Predicted width profile halfway between wellbore and tip for first MWX stimulation.
- Figure 9 Geometry and parameters for vertical section of asymmetric fracture, as used in calculating fracture widths.
- Figure 10 Net pressure corresponding to Figure 9, but with lithostatic or hydrostatic gradients superimposed.
- Figure 11 Net pressure and gradients when crack center exits payzone.

INTRODUCTION

At depths of a few thousand feet or more, induced hydraulic fractures will normally be vertical. Height growth containment is important so that the fracture will reach farther along the payzone, and so that the chance of vertical penetration into, for example, a water-bearing zone will be reduced. Although many factors influence height growth, the most important one appears to be the stress contrast between pay and bounding zones,¹ where by stress we mean minimum in-situ stress.

Here we study fracture height growth by developing a model for an expanding hydraulic fracture applicable when the fracture is highly elongated, with length along the payzone much greater than height. Vertical variations in elastic parameters are not considered. The fracture shape in this paper is self-determined, in contrast to that in which an elliptical shape is chosen⁵. A variable-height fracture model has been intensively studied by Cleary and co-workers.^{2,3} This so-called "pseudo-3D" model treats the fluid flow as a dominant 1D flow along the payzone, plus an auxiliary 1D flow in the vertical direction. Although the models of Nolte⁴ and Palmer and Carroll^{5,6} take the vertical flow to be zero, thus simplifying the problem considerably, the general formulations are similar enough to Cleary's to be included under the rubric "pseudo-3D." In all these models, the fracture width is approximated by dividing the fracture into a number of vertical sections, and applying 2D elasticity theory to each vertical "line" crack. Thus the fracture is assumed to be highly elongated with length/height ratio ≥ 4 .

Fully 3D modeling, with proper 2D fluid flow, is under development,⁷⁻⁹ but the problem is formidable and the computer run time enormous. The first complete results have just appeared.⁷ Meanwhile we can learn much about

fracture behavior from pseudo-3D models. Furthermore, recent comparisons indicate¹⁰ that the the ORU pseudo-3D models discussed below are a good first approximation to the fully 3-D model.⁷

In this report we consider a 3-layer scenario in which a continuous payzone is bounded by zones of higher minimum in-situ stress. If the bounding zone stresses are equal, the fracture will be vertically symmetric (if lithostatic and hydrostatic gradients are neglected). Since we have presented a fracture simulator appropriate to this case before^{5,6}, here we briefly describe a revised version which contains (i) non-Newtonian flow, (ii) leakoff, including spurt loss. A discussion of the modeling of the latter is given in some detail because it has not been presented before.

In general, the bounding layer stresses will not be equal, leading to a fracture which is vertically asymmetric, and furthermore both the minimum in-situ stress and the fluid pressure will vary with depth (i.e., both will have vertical gradients). In the asymmetric model of this paper, calculation of fracture width is the most difficult task. We give most of the details here. Theoretical calculations of asymmetric fracture shapes have been reported by Settari and Cleary,² but they appear to emphasize low stress contrasts ($\lesssim$ couple hundred psi) which may not satisfy our prerequisite for a highly elongated fracture. Nolte⁴ gives one asymmetric width profile in a vertical section, but no method of calculation, nor any overall fracture shapes, were given.

Finally, to illustrate the results of the asymmetric model, we use the model to predict fracture height, pressure, and width for the first stimulation of the Multi-Well Experiment (MWX) carried out in December 1983. This prediction is compared with available fracture diagnostic measurements.

II. CONSTRUCTION OF FRACTURE SIMULATORS

The geometry and parameters of a fracture initiated in a continuous sand bounded by zones of higher minimum in-situ stress are shown in Figure 1. At this stage we describe the modeling in general terms, so as to include both vertically symmetric and asymmetric fractures. There are 4 equations which govern the expansion of a hydraulic fracture.⁶ To obtain the first 3 equations, we divide the fracture into a series of vertical sections. If the fracture length/height ratio is large enough (≥ 4), we can treat each vertical section as independent of its neighbor, to a good approximation.

(a) Width equation. As described previously,⁶ each vertical section is regarded as a line crack in plane strain, and the width is obtained as a function of net pressure and total height:

$$w(x,y) = f_1[p(x), h(x)] \quad (1)$$

(b) Pressure gradient equation. The fluid pressure drop across a vertical section is a function of the width profile, the height of the section, and the flow rate through the section:

$$\frac{dp(x)}{dx} = f_2 [w(x,y), h(x), q(x)] \quad (2)$$

Non-Newtonian flow is included.

(c) The fracture expansion criterion. We use a result from linear elastic fracture mechanics, that $K_i = K_c$ where K_i is stress intensity factor and K_c is fracture toughness.¹¹ For relatively slow vertical expansion (e.g., a highly elongated fracture in which the vertical flow component is ignored) the result is a direct relation between net pressure and fracture height.

$$p(x) = f_3[h_u(x), h_l(x)] = f_4[h_u(x), h_l(x)] \quad (3)$$

where, in allowing for a vertically asymmetric fracture, $h_u(x) + h_l(x) = h(x)$ is total height.

(d) The equation of continuity. The flow rate change across a vertical section equals the rate of fluid leakoff, per unit fracture length, plus the "storage" term as the section expands in volume.¹²

$$\frac{-dq}{dx}(x,t) = \lambda(x,t) + \frac{dA}{dt}(x,t) \quad (4)$$

In addition to the regular filtration loss $\lambda(x,t)$, spurt loss can be included.

The 4 equations (1)-(4) above have to be solved simultaneously to study fracture growth with time. Since time only occurs in equation (4) our approach is to solve (1)-(3) first, then satisfy (4) by iteration. Equations (1)-(3) can in principle be combined⁶ into an ordinary, first-order, first-degree, non-linear differential equation in $h(x)$ or $p(x)$. Then if a reasonable assumption is made for $q(x)$, this equation can be integrated from the fracture tip at $x = a$, where the boundary condition is specified, back to the wellbore to find $h(x)$ and $p(x)$.

In practice, a value is chosen for p at $x = a$ (the boundary condition), then (3) is used to find the fracture height. The $p(a)$ value is chosen such that the total height of the section at $x = a$ is just slightly greater than H , the payzone height. As discussed previously,⁶ the wellbore height and pressure that result are not sensitive to the actual height or pressure at the fracture tip. The width profile $w(x,y)$ is then calculated at 10 points per vertical section using (1). Then (2) is integrated by the standard fourth order Runge-Kutta technique to find a new pressure at $x = a - dx$. Using this new pressure,

(3) is solved to find a new height at $x = a - dx$. The process then repeats until we reach the wellbore at $x = 0$.

The usual resolution for the numerical integration is determined by dividing the fracture half-length into 10 segments, and each segment into 11 divisions, so that $dx = a/110$. However, in the segment nearest the tip at $x = a$, we increase the resolution to $dx = a/550$ because the pressure can change much more rapidly in this region. This resolution is usually sufficient--we check it by doubling the resolution to verify that the results do not change.

The way we handle the channel shape factor $\phi(x)$ in the fluid flow equation (2) (see Appendix B) is to assume $\phi = 3\pi/16$ in the segment closest to $x = a$. This value corresponds to that for a vertical width profile which is elliptical,^{12,13} and is a good approximation when the fracture has not penetrated far into the bounding zones. After integrating $p(x)$ across the first segment, using this ϕ value, a new ϕ value is calculated from the vertical width profile at the end of the segment. This adjusted ϕ value is then used to integrate $p(x)$ across the next segment, and so on. This is an adequate way to do it because $\phi(x)$ changes only slowly with x .

The solution for $h(x)$ and $p(x)$ obtained so far may be regarded as a first approximation for computing wellbore height and pressure--first because the flow rate $q(x)$ has been assumed without satisfying equation (4) (unless by coincidence). However, the results are not strongly dependent on $q(x)$, so long as it is chosen judiciously.

To calculate the time corresponding to a given fracture length requires that equation (4) be properly satisfied, and this is considerably more complicated. Although the details are given in Appendix C, in brief the leakoff term $\lambda(x,t)$ requires a knowledge of the fracture length expansion rate $a(t)$ --we assume this. The storage term $dA(x,t)/dt$ in (4) comes from solving for $h(x)$,

$p(x)$, and $w(x,y)$ in a second fracture incremented in length to $a + da$. Subtraction gives da . The time increment dt comes from the assumed expansion rate $a(t)$. Then (4) can be integrated to obtain $q(x)$, which is compared with the $q(x)$ assumed for the simultaneous solution of equations (1)-(3) described earlier. If they do not agree, new estimates for $q(x)$ and $a(t)$ are tried until the iteration converges. This becomes the full solution with leakoff included and continuity satisfied. Finally, the time corresponding to a given fracture length, a , comes from the converged length expansion rate $a(t)$. This full solution is valid for any leakoff coefficient.

Now we expand on this description. First, we present recent improvements to the vertically symmetric model, which was published previously^{6,21}. Second, we present the details of the modeling involved in the asymmetric simulator.

III. IMPROVED VERSION OF VERTICALLY SYMMETRIC 3-LAYER SIMULATOR

(a) Improvements to the model

The main improvements over previous model design^{6,21} are three:

(i) Analytic expressions for fracture width

Previously^{6,21} some widths had to be calculated by numerical integration. Appendix A gives the new results, which we have been able to express in closed form. These have been checked against the previous calculations, and they agree. The main advantage is a savings in computer time.

(ii) Incorporation of non-Newtonian flow

Whereas previously^{6,21} only Newtonian flow was included, now we have incorporated flow of power law fluids. The equations are given in Appendix B.

(iii) Leakoff analysis (including spurt loss) has been improved

This is the most difficult part of the modeling because it involves iterating on both $a(t)$, the fracture expansion rate along the x-axis, and on $q(x)$, the flow rate at each station (and there are 3 stations a_1 , a_2 , and a_3). The details are given in Appendix C.

(b) Illustrative result

The computer code SYMFRAC1 and running instructions are given in Appendix F.

For the parameters given in Table I, we obtain the following results:

- Values of wellbore height, pressure, and width-see Table II
- Fracture shape $h(x)$ -see Figure 2
- Bottomhole pressure, $p(t)$, as a function of time-see Figure 3
- Wellbore width in a vertical section, $w(y)$ -see Figure 4

IV. NEW VERTICALLY ASYMMETRIC 3-LAYER SIMULATOR

(a) Details of the modeling

(i) Fracture width

This is the most complicated part of the theory. However, all component integrals in the width formulation can be expressed in closed

form, which reduces the computer run time. The calculation is contained in Appendix D.

(ii) Pressure gradient equation

We use exactly the same equation as for the 3-layer symmetric simulator-see Appendix B.

(iii) Fracture expansion criterion

For a vertically symmetric simulator the fracture expansion criterion $K_i = K_c$, where K_i is stress-intensity factor and K_c is fracture toughness, is relatively easy to formulate, and the result has been given previously^{6,21}:

$$p = \sqrt{\frac{2}{\pi h}} K_c + \frac{2}{\pi} (S_2 - S_1) \cos^{-1} \left(\frac{H}{h} \right) \quad (5)$$

Here we assumed, for a highly elongated fracture and relatively slow vertical expansion, that there were no pressure gradients due to fluid flow in the vertical direction. We also assumed each vertical section to be a 2D or line crack.

For a vertically asymmetric fracture, the formulation is more difficult because $K_i = K_c$ must be satisfied at both the upper and lower tips of the line crack, and this requires an iterative procedure. The details of the calculation and the results are given in Appendix E.

(iv) Leakoff and solution of the continuity equation

The approach follows that for the symmetrical simulator (see Appendix C) up to the point where the double iteration on $q(x)$ and $a(t)$ is performed by calculating 3 fractures of lengths a_1 , a_2 , and a_3 simultaneously, and using a bilinear power law (see equation C3) for $a(t)$, the length

expansion rate. Since the double iteration on $q(x)$ and $a(t)$ makes the solution rather tedious, we have developed a short-cut as follows.

An "input" flow rate, $q(x)$, is guessed, and then we obtain two fracture solutions for lengths a and $a + da$. Now we assume a single power law

$$x = mt^n \quad \text{with } a_1 = mt_1^n \quad (6)$$

over the whole range of time. But instead of allowing m and n to vary independently, we tie their values together such that $\log m = n$. For a chosen value of n , equation (C7) can be used to find the total volume leaked off, and when this is added to the actual volume of the fracture computed from the vertical cross section areas, it leads to a time via the inlet flow rate. If this time does not agree with that from (6), a new value of n is chosen until the time converges.

Now the "output" flow rate is calculated via (C9) and (C10), using the volumes of the 2 fractures, and their time difference, dt . If "input" and "output" flow rates agree, we have a solution. If not, we iterate on $q(x)$ until we do.

From tests against the solution for a bilinear $a(t)$ described in Appendix C, we have found this short-cut solution to be a good approximation for the time corresponding to a given fracture length, a , (although we have been unable to fully justify it theoretically). However, $a(t)$ is not properly determined for times less than that corresponding to the current position of the fracture. To obtain this $a(t)$ would require simultaneous solutions for at least two different fracture lengths (which can certainly be done).

(b) Application to first MWX stimulation

The computer code ASYMFRACT and running instructions are given in Appendix G.

The Multi-Well Experiment (MWX) is a major field test being conducted by the U.S. Department of Energy in northwestern Colorado. It is designed to test the latest technology and determine whether recovery techniques are viable in the tight gas sand reservoirs characterizing this area. Three wells in a triangular geometry (100-200 ft apart) have been drilled to around 8000 ft in the Mesaverde section of the Piceance Basin.

The Paludal zone, at 6600-7450 ft, contains lenticular, distributary channel sandstones formed in a lower delta plain environment. Although many stimulations are planned for the MWX, the first one, in December 1983, was in an 80 ft fracture interval at 7100 ft in the Paludal zone. Minimum in-situ stresses were measured¹⁴ using mini-fracs. Surprisingly high stress contrasts were found at the top of the fracture interval at 7070 ft (520 psi) and at the bottom at 7150 ft (1195 psi).

The fracture design was based on a 30,646 gal stimulation with 60 lbs gel per 1000 gal fluid, pumped at a rate of 10 bpm. Other parameters are given in Table III. Young's modulus varied slightly between the fractured interval and the bounding zone below (4.8 to 5.0×10^6 psi), but such variations are ignored here. Also ignored are lithostatic gradients of in-situ stress and hydrostatic gradients of fluid pressure. The power-law flow parameter k is taken to be independent of distance x away from the wellbore (i.e., we ignore fluid degradation due to temperature and time). Finally, the leakoff is determined from the wall-building properties of the fluid.

The predictions of our fracture simulator are summarized in Table IV. The half length would be 440 ft. The total fracture height was predicted to

be 127 ft at the wellbore, and the bottomhole pressure 388 psi above the closure pressure (= minimum in-situ stress S_1). The variations of height and pressure with distance along the fracture are displayed in Figures 5 and 6. There was concern that the fracture might break out of the upper confining zone at 7015 ft into another zone in which the stress is 60 psi below the perforated interval. However, the prediction was that, although the fracture penetrated much farther into the upper bounding zone (42 ft) than into the lower zone (6 ft), there was still another 13 ft to get to the low stress zone.

Finally, the maximum fracture width at the wellbore is predicted to be 0.44 cm (total width). The corresponding width profile in the vertical section is shown in Figure 7. The most conspicuous feature is the "pinched-in" width due to the higher-stress bounding zone above the top of the perforated interval (= payzone). Half-way between the wellbore and the square fracture tip, the effect is much less conspicuous as shown by Figure 8. That is, the width profile is more like an ellipse, as would be the case in a uniform stress field, because there is less penetration into the high stress zone above.

The MWX has a battery of diagnostic measurements, chief among which are a bottomhole pressure gauge, a temperature-versus-height profile at the wellbore to infer fracture height, and a borehole seismic system to delineate the fracture size and shape. Table IV also contains the best estimates of height, pressure, and half-length inferred from the measurements.¹⁵ It is also important to note that bottomhole pressure continued to rise during the experiment--there was no levelling off.¹⁵

The outstanding discrepancy in Table IV is the bottomhole pressure. This immediately suggests that the stress contrast in the upper bounding zone is much greater than 520 psi. In fact it would need to be ≥ 1200 psi to give such a high bottomhole pressure. But if this were the case, the penetration above

7070 ft would be less than the 41.7 ft we predicted, that is, the total fracture height should be less than our predicted 127 ft, not more! Clearly, more in-situ stress measurements need to be made above the perforated interval.

Another possibility is that the effective height of the low-stress region in the perforated interval (S_1 in Figure 1) is less than 80 ft. The net pressure increases as the effective height decreases, as shown by Nolte⁴ (see his Figure 9). Along the same lines, if a high-stress stringer lies somewhere in the middle of the fractured interval, once again Nolte⁴ has shown that this increases the bottomhole pressure dramatically, with an accompanying height increase.

There may be other explanations too: for example, if one wing of the fracture were stopped from expanding, that would cause an increase in pressure.

It is clear that fairly close-spaced mini-frac measurements of in-situ stress could help discriminate between the different explanations above. There is substantial variability within the Paludal zone,¹⁵ e.g., layers of silt, coal, sand, shale, and mudstone, varying from inches to feet in thickness, and the stresses may be very different in these layers.

V. CONCLUSION

The basic model of fracture propagation in a three-layer situation, which has been described previously,^{6,21} has been extended in this report by inclusion of the following:

1. Non-Newtonian fluid flow (power law) has been incorporated.
2. Spurt loss has been added to the leakoff. The leakoff calculation, based on simultaneous fracture solutions at three stations, has been improved in the symmetrical simulator. In the asymmetrical simulator a short-cut

approximation has been employed based on one station only (i.e., just one fracture length). It gives a good approximation for the time corresponding to a given fracture length, and, of course, it can be extended to include other stations simultaneously.

3. Minimum in-situ stresses in the bounding zones do not have to be equal, leading to a vertically asymmetric fracture. The width calculation in each vertical section is considerably more complicated. Width is calculated exactly, not approximated.

The limitations of the modeling are:

1. The length/height aspect ratio of the fracture has to be large (≥ 4), meaning the rate of vertical height growth is much less than the rate of length growth (e.g., the stress contrasts are large, $\geq$ few hundred psi).
2. Vertical variations in elastic modulus are ignored.
3. Time variations in pump rate (q) or fluid viscous properties (n, k) have not been considered.

The fracture height at the square tip (see Figure 1) is assumed to be slightly greater than the payzone height. As shown elsewhere,⁶ if the fracture is highly elongated the height and pressure at the tip have minimal effect on the solution for height and pressure well away from the tip, including the wellbore.

Our fracture expansion criterion is $K_i = K_c$ with $K_c \neq 0$. While the effect of finite K_c is small in many cases, it cannot always be ignored, especially when stress contrasts are lower.

The asymmetric simulator has the ability to treat vertical gradients in fluid pressure (hydrostatic) and minimum in-situ stress (lithostatic), but has not yet been debugged.

One advantage of the simulators described herein is that they do not have to be marched forward in time, as most of the simulators do. That is, instead of starting with a tiny fracture and then computing its growth in time, we start with a given fracture length and from that determine the time. The only sense in which they are marched forward is in the case of the vertically symmetric simulator, where the solutions for 3 increasing fracture lengths, or stations, are found simultaneously. A consequence of this is very low CP times--about 2 min on a Data General Eclipse minicomputer for the symmetrical simulator, and about 5 min for the asymmetrical simulator.

We have recently made¹⁰ a detailed comparison between the ORU simulators described above and the AMOCO, MIT, and TERRA-TEK simulators (i.e., all simulators which predict a variable height). Although there exist discrepancies, we have been able to reconcile some of these through detailed analysis; others we have not been able to reconcile. A conclusion of the work was that for a highly elongated fracture (length/height ratio ≥ 4), the ORU simulator provides a fairly inexpensive way to obtain a good approximation to more sophisticated simulators^{3,4}, including the fully-3D formulation.⁷

An illustrative prediction of fracture shape, width, and bottomhole pressure has been given in this report for the symmetrical simulator.

To illustrate the results in the case of the asymmetrical simulator, we have predicted the asymmetric growth of a 30,000 gal stimulation--the first in the Multi-Well Experiment. The first suite of mini-frac measurements suggested a stress contrast between the fractured interval and the upper zone of 520 psi, and 1195 psi for the lower zone. However, the observed net bottomhole pressure rose to a much higher value (~1200 psi) than predicted by the simulator (388 psi). In addition, the fracture height inferred from the measurements (~150 ft) was somewhat greater than that predicted from the model (127 ft). Some suggestions

have been made to account for the discrepancies. It appears that more close-spaced in-situ stress measurements will have to be made to see whether there exist other high-stress zones in the vicinity of the fracture that could be responsible for the elevated bottomhole pressure.

ACKNOWLEDGMENT

Many thanks to Lian Soh for her conscientious programming efforts. Special thanks to Dale Williams for his constant support. This work was partly supported by the U.S. Dept. of Energy and partly by Oral Roberts University. We acknowledge a discussion with N.R. Warpinski, who suggested some of the ideas for reconciling our prediction with the MWX field measurements.

NOMENCLATURE

A = area of vertical section of fracture
a = fracture half-length
C = leakoff coefficient
E = Young's modulus
f(y) = even function component of net pressure
g(y) = odd function component of net pressure
h = total fracture height
 h_u = fracture height from center of payzone to top
 h_l = fracture height from center of payzone to bottom
 H^l = payzone height
 K_i = stress intensity factor
 K^i = fracture toughness
 λ^c = rate of fluid leakoff per unit fracture length
 μ = fluid viscosity
k = amplitude parameter for power law fluid
n = exponent parameter for power law fluid
 ν = Poisson's ratio
 ℓ = half-height of 2D line crack
 p_f = fluid pressure
 $p = p_f - S$ = net pressure
 p_o = net bottomhole pressure
 p_{o1}, p_{o2}, p_{o3} = vertical net pressure gradients in three zones
 ϕ = channel shape factor
 $q(x)$ = flow rate (one wing)
 S_1 = minimum in-situ stress in payzone
 S_2, S_3 = minimum in-situ stresses in bounding layers
 S = spurt loss
 t^p = time
 $\tau(x)$ = time at which fracture reaches x and leakoff begins
 V = volume lost in spurt loss
 $w^s(y)$ = width of 2D crack in vertical section
w = maximum width of crack in vertical section
 x^o = horizontal coordinate (along payzone)
y = vertical coordinate

REFERENCES

1. Warpinski, N. R., R. A. Schmidt, and D. A. Northrop: "In-situ stresses: The predominant influence on hydraulic fracture containment," J. Pet. Tech., 653-663, March 1982.
2. Settari, A. and M. P. Cleary, "Development and testing of a pseudo-three-dimensional model of hydraulic fracture geometry (P3DH)," SPE 10505 Sixth SPE Symp. Reservoir Simulation, New Orleans, Jan. 1982.
3. Cleary, M. P., R. G. Keck and M. E. Mear: "Microcomputer models for the design of hydraulic fractures," SPE/DOE 11628, Symp. Low Permeability, Denver, March 1983.
4. Nolte, K. G.: "Fracture design considerations based on pressure analysis," SPE 10911, Cotton Valley Symp. of SPE, Tyler, TX, May 1982.
5. Palmer, I. D. and H. B. Carroll, Jr.: "3D hydraulic fracture propagation in the presence of stress variations," Soc. Pet. Eng. J., p. 870-878, Dec. 1983.
6. Palmer, I. D. and H. B. Carroll, Jr.: "Numerical solution for height of elongated hydraulic fractures," SPE/DOE 11627, Symp. Low Permeability, Denver, March 1983.
7. Abou-Sayed, A. S., R. J. Clifton, R. L. Dougherty, and R. H. Morales, "Evaluation of the influence of in-situ reservoir conditions on the geometry of hydraulic fractures using a 3D simulator: Part 2-Case studies," SPE/DOE/GRI 12878, Proc. Unconventional Gas Recovery Symp., p. 441, Pittsburgh, 1984.
8. Cleary, M. P., M. Kavvadas, and K. Y. Lam: "Development of a fully three-dimensional simulator for analysis and design of hydraulic fracturing," SPE/DOE 11631, Symp. Low Permeability, Denver, March 1983.

9. Barree, R. D.: "A practical numerical simulator for three-dimensional fracture propagation in heterogeneous media," SPE 12273, Reservoir Simulation Symp., San Francisco, Nov. 1983.
10. Palmer, I. D. and C. T. Luiskutty: "Comparison of hydraulic fracture simulators for elongated fractures of variable height," Oral Roberts University preprint, 1984.
11. Simonson, E. R., A. S. Abou-Sayed, and R. J. Clifton: "Containment of massive hydraulic fractures," Soc. Pet. Eng. J., 18, 27-32, Feb. 1978.
12. Nordgren, R. P., "Propagation of a vertical hydraulic fracture," Soc. Pet. Engrs. J., 12, 306, 1972.
13. Nolte, K. G.: "Determination of fracture parameters from fracturing pressure decline," SPE 8341, 54th Ann. Fall Tech. Conf. and Exhib. of SPE, Las Vegas, 1979.
14. Warpinski, N. R.: "In-situ stress measurements at DOE's Multi-Well Experiment," Western Gas Sands Subprogram Review Technical Proceedings, DOE/METC/84-3, 160, 1983
15. Warpinski, N. R.: "Summary of results of MWX Paludal zone phase I stimulation," Sandia National Laboratories Memo, Jan. 1984
16. England, A. H. and A. E. Green: "Some two-dimensional punch and crack problems in classical elasticity," Proc. Cambridge Phil. Soc., 59, 489, 1963.
17. Daneshy, A. A., "Opening of a pressurized fracture in an elastic medium," paper 7616 pres. at 22nd Ann. Tech. Mtg. of Pet. Soc of CIM, Banff, June 1971.
18. Rice, J. R., "Mathematical analysis in the mechanics of fracture," in Fracture - an Advanced Treatise (ed. H. Liebowitz), Academic Press, New York, Vol. II, 1968.

19. Perkins, T. K. and L. R. Kern: "Widths of hydraulic fractures," J. Pet. Tech., 937-949, Sept. 1961.
20. Geertsma, J. and R. Haafkens: "A comparison of the theories for predicting width and extent of vertical hydraulically induced fractures," J. Energy Resources Technology, 101, 8-19, 1979.
21. Palmer, I. D. and H. B. Carroll, Jr., "Numerical model of massive hydraulic fracture," Second Ann. Rept. DOE/BC/10215-27, 1983.

Table I

Parameters used to illustrate
vertically symmetric simulator

Fracture length (1 wing)	3200 ft
Payzone height	75 ft
Young's modulus	5.76×10^6 psi
Poisson's ratio	0.2
Fracture toughness	0
Stress contrast	1200 psi
Pump rate (1 wing)	15 bpm
Fluid viscosity	100 cp
(constant with x)	
Leakoff coefficient	$.001 \text{ ft/min}^{\frac{1}{2}}$
Spurt loss coefficient	0

Table II

Predictions of symmetric fracture
simulator for parameters in Table I

Volume pumped, total	253,000 gal
Pump time	201 min
Total wellbore height	272 ft
Bottomhole pressure (above S_1)	987 psi
Wellbore maximum width	1.14 cm

Table III

Parameters used to illustrate
vertically asymmetric
simulator-first MWX stimulation

Perforated height	80 ft
Young's modulus	4.8×10^6 psi
Poisson's ratio	.22
Fracture toughness	1200 psi in ^{1/2}
Stress contrast @ 7070 ft	520 psi
Stress contrast @ 7150 ft	1195 psi
Pump rate (1 wing)	5 bpm
Gel concentration	60lb/1000gal
n (200°F)	.46
k (200°F) (constant with x)	.02 lb _f · sec ⁿ /ft ²
Leakoff coefficient (wall-building)	.00142 ft/min ^{1/2}
Spurt loss coefficient	.0453 gal/ft ²

Table IV

Predictions of asymmetric fracture
simulator, and comparison with field
measurements, in first MWX stimulation

	Predicted	Measured
Volume pumped	30,646 gal	34,500 gal
Pump time	73.2 min	
Half-length	440 ft	~400 ft
Penetration above 7070 ft	41.7 ft	~50 ft
Penetration below 7150 ft	5.6 ft	~20ft
Total wellbore height	127 ft	~150 ft
Bottomhole pressure (above closure = 5805 psi)	388 psi	~1200 psi
Wellbore maximum width	0.44 cm	

APPENDIX A

Width Equation for 2D (line) Crack - Symmetrical Fracture

For an elongated fracture which originates in a payzone bounded by zones of higher, but equal, in situ stress, the full width in each vertical section is given by¹⁶

$$w(f_\ell) = \frac{8(1-\nu^2)\ell}{\pi E} \int_{f_\ell}^1 \frac{f_2 df_2}{\sqrt{f_2^2 - f_\ell^2}} \int_0^{f_2} \frac{p(f_1) df_1}{\sqrt{f_2^2 - f_1^2}} \quad (A1)$$

where f_1 , f_2 and $f_\ell = (y/\ell)$ are fractions of vertical half height ℓ of the crack in the section. Setting $f_{\ell'} = (H/2)/\ell = H/h$ where $h = 2\ell$ and H is payzone height, the above equation can be rewritten for (i) points within the payzone ($|y| < \frac{H}{2}$), and (ii) for points outside the payzone ($|y| > \frac{H}{2}$). The net pressure profile is

$$\begin{aligned} p(y) &= p_f - S_1 & |y| < \frac{H}{2} \\ &= p_f - S_2 & |y| > \frac{H}{2} \end{aligned}$$

(i) Points within the payzone ($f_\ell < f_{\ell'}$)

Equation (B1) is rearranged as

$$\frac{\pi E w(f_\ell)}{8(1-\nu^2)\ell} = \int_{f_\ell}^{f_{\ell'}} \frac{f_2 df_2}{\sqrt{f_2^2 - f_\ell^2}} \int_0^{f_2} \frac{p df_1}{\sqrt{f_2^2 - f_1^2}} + \int_{f_{\ell'}}^1 \frac{f_2 df_2}{\sqrt{f_2^2 - f_\ell^2}} \int_0^{f_2} \frac{p df_1}{\sqrt{f_2^2 - f_1^2}} \quad (A2)$$

In the first term on RHS $f_1 < f_{\ell'}$, and

$$\int_0^{f_2} \frac{p(f_1) df_1}{\sqrt{f_2^2 - f_1^2}} = \frac{\pi}{2} (p_f - S_1)$$

In the second term we may write

$$\int_0^{f_2} = \int_0^{f_{\ell'}} + \int_{f_{\ell'}}^{f_2}$$

Substituting these in (B2) the width equation yields

$$\begin{aligned} \frac{\pi E w(f_\ell)}{8(1-v^2)\ell} &= \int_{f_\ell}^{f_{\ell'}} \frac{f_2 df_2}{\sqrt{f_2^2 - f_1^2}} \cdot \frac{\pi}{2} (p_f - S_1) \\ &+ \int_{f_{\ell'}}^1 \frac{f_2 df_2}{\sqrt{f_2^2 - f_1^2}} \left\{ \frac{\pi}{2} (p_f - S_1) \left[\left(1 - \frac{2}{\pi}\right) \frac{S_2 - S_1}{p_f - S_1} \cos^{-1} (f_{\ell}'/f_2) \right] \right\} \quad (A3) \end{aligned}$$

This can be evaluated analytically, giving

$$\begin{aligned} \frac{\pi E w(f_\ell)}{8(1-v^2)\ell} &= \sqrt{1-f_\ell^2} \left[\frac{\pi}{2} (p_f - S_1) - (S_2 - S_1) \cos^{-1} f_{\ell}' \right] \\ &+ (S_2 - S_1) \left[f_{\ell}' \ln \left(\frac{\sqrt{1-f_\ell^2} + \sqrt{1-f_{\ell'}^2}}{\sqrt{f_{\ell'}^2 - f_\ell^2}} \right) - f_\ell \ln \left(\frac{f_{\ell}' \sqrt{1-f_\ell^2} + f_\ell \sqrt{1-f_{\ell'}^2}}{\sqrt{f_{\ell'}^2 - f_\ell^2}} \right) \right] \quad (A4) \end{aligned}$$

(ii) Points outside the payzone ($f_\ell > f_{\ell}'$)

The second integral in (B1) is written as a sum of two integrals, i.e.

$$\frac{\pi E w(f_\ell)}{8(1-v^2)\ell} = \int_{f_\ell}^1 \frac{f_2 df_2}{\sqrt{f_2^2 - f_1^2}} \left[\int_0^{f_{\ell}'} \frac{p(f_1) df_1}{\sqrt{f_2^2 - f_1^2}} + \int_{f_{\ell}'}^{f_2} \frac{p(f_1) df_1}{\sqrt{f_2^2 - f_1^2}} \right]$$

And after the evaluation of the second integral, one gets

$$\frac{\pi E w(f_\ell)}{8(1-v^2)\ell} = \int_{f_\ell}^1 \frac{f_2 df_2}{\sqrt{f_2^2 - f_\ell^2}} \frac{\pi}{2} (p_f - S_1) \left[1 - \frac{2}{\pi} \left(\frac{S_2 - S_1}{p_f - S_1} \right) \cos^{-1} (f_{\ell}'/f_2) \right]$$

The integral is the same as the second one in (A3) except that the lower limit

is f_ℓ , not f_{ℓ}' . When integrated this becomes

$$\begin{aligned} \frac{\pi E w(f_\ell)}{8(1-v^2)\ell} &= \sqrt{1-f_\ell^2} \left[\frac{\pi}{2} (p_f - S_1) - (S_2 - S_1) \cos^{-1} f_{\ell}' \right] \\ &+ (S_2 - S_1) \left[f_{\ell}' \ln \left(\frac{\sqrt{1-f_\ell^2} + \sqrt{1-f_{\ell'}^2}}{\sqrt{f_{\ell'}^2 - f_\ell^2}} \right) \right. \\ &\quad \left. - f_\ell \ln \left(\frac{f_{\ell}' \sqrt{1-f_\ell^2} + f_\ell \sqrt{1-f_{\ell'}^2}}{\sqrt{f_{\ell'}^2 - f_\ell^2}} \right) \right] \quad (A5) \end{aligned}$$

APPENDIX B

The Pressure Gradient Equation in Non-Newtonian Flow

The pressure drop across each vertical section for the 1D flow along the payzone direction (x-axis) of a power law fluid is given by^{13,19}

$$\frac{dp}{dx} = - \frac{Mk(2q)^n}{h^n w_o^{2n+1}} \quad (B1)$$

where q is the integrated flow rate down one wing, h is the total fracture height at that point,

$$M = 2 \left[\frac{m}{\Phi(n)} \right]^n, \quad m = \frac{2n+1}{n}$$

and n is the component in shear stress = k (shear rate) ^{n} . In the above equation $\Phi(n)$ is defined as

$$\Phi(n) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\frac{w}{w_o} \right)^m d\left(\frac{y}{H} \right)$$

where w is the full width at each height and w_o the maximum width in that vertical section. With the appropriate substitutions, (B1) becomes

$$\frac{dp}{dx} = -2^{n+1} \left[\frac{(2n+1) q(x)}{n\Phi(n) h(x)} \right]^n \frac{k(x)}{[w_o(x)]^{2n+1}} \quad (B2)$$

when both $q(x)$ and $k(x)$ can in general vary with x . For Newtonian flow, $n=1$ and $k=\mu$ is the viscosity and the above equation reduces to

$$\frac{dp}{dx} = - \frac{12 \mu(x) q(x)}{\Phi(1) h(x) w_o^3(x)} \quad (B3)$$

If we substitute the limiting value of $\Phi(1) = \frac{3\pi}{16}$ for an elliptical cross section, the pressure drop equation reduces to

$$\frac{dp}{dx} = - \frac{64 \mu(x) q(x)}{\pi h(x) w_o^3(x)}$$

in agreement with Lamb (see Nordgren¹²)

APPENDIX C

Inclusion of leakoff and spurt loss in solution of the continuity equation

The classical leakoff rate per unit fracture length is given by¹²

$$\lambda(x,t) = 2HC/\sqrt{t - \tau(x)} \quad (C1)$$

where C is leakoff coefficient, t the current time, and $\tau(x)$ the time at which the fracture reached the point x . Hence to calculate $\lambda(x,t)$, a rate of fracture length expansion $a(t)=x(\tau)$ needs to be assumed ($\tau(x)$ is the inverse of $a(t)$). The rate of spurt loss, when new fracture surface area is formed, can be given by²⁰

$$\frac{dV_s}{dt} = S_p \frac{dA}{dt}$$

where S_p is spurt loss coefficient, having dimensions of volume/area.

Our goal is to solve for $q(x)$ by integrating equation (4) in the text. This we do by utilizing the 10 segments into which the fracture is divided lengthwise. The leakoff rate in each segment is obtained by integrating (C1) over the length of the segment. The spurt loss, which is an additional leakoff rate in the segment closest to the tip at $x \approx a$, is given by

$$\frac{dV_s}{dt} = 2S_p \frac{dA}{dt} = 2S_p H \frac{da}{dt} \quad (C2)$$

where H is the height of the permeable section (the factor of 2 accounts for leakoff through both fracture faces).

The storage term, dA/dt , in (4) corresponds to the rate of fluid "lost" to volume expansion in a segment. The volume expansion comes by subtraction of the solutions for $h(x)$, $p(x)$, and $w(x,y)$ for two fractures of lengths a and $a + da$, while the time increment dt corresponding to this length increment da comes from the assumed length expansion rate $a(t)$.

Equation (4) is integrated by starting with the inlet flow rate at the wellbore, and subtracting the leakoff rate and storage term in the first

segment to give the flow rate entering the second segment, and so on. The result is a flow rate profile $q(x)$. If this $q(x)$ agrees with that assumed in order to solve equations (1)-(3), then we have a solution. If not, we iterate on $q(x)$ and $a(t)$ until $q(x)$ converges. Once we have convergence, to an acceptable accuracy, then we have a solution for the time corresponding to the given fracture length, a , from the length expansion rate $a(t)$.

The double iteration on $q(x)$ and $a(t)$ has been performed in the following way. We solve for 3 different fracture lengths simultaneously (e.g., a_1 , a_2 , a_3). We assume $a(t)$ to be a bilinear power law

$$a(t) = x(\tau) = \begin{cases} m_1 t^{n_1} & a \leq a_2 \\ m_2 t^{n_2} & a_2 < a < a_3 \end{cases} \quad (C3)$$

which is continuous at $a = a_2$. Values for m_1 , m_2 and n_1 , n_2 are initially guessed, and then $a(t)$ is used to compute total volume of fluid leaked off from each fracture. This volume, when added to the actual fracture volume, gives via the inlet pump rate the times t_1 , t_2 , t_3 corresponding to each of the 3 fracture lengths a_1 , a_2 , a_3 . These three values of $a(t)$ are then compared with those assumed in (C3), and revised values computed. At the same time, flow rates $q(x)$ for each of the three fractures are being converged. The end result is convergence of $a(t)$ and $q(x)$ to an acceptable degree of accuracy.

(a) Iteration on $a(t)$

The total volume leaked off is found by integrating (C1) over both time and fracture length (for one wing)

$$V_{\ell 1} = 2HC \int_x \int_t \frac{dt dx}{\sqrt{t-\tau(x)}} \quad (C4)$$

Since for a given point x , leakoff only begins when the fracture arrives there, at time $\tau(x)$, we have

$$V_{\ell 1} = 2HC \int_x \left[\int_{\tau}^{t_1} \frac{dt}{\sqrt{t-\tau(x)}} \right] dx$$

where t_1 = time corresponding to a final fracture length, a_1 ,

$$V_{\ell 1} = 4HC \int_0^{a_1} \sqrt{t_1 - \tau(x)} \quad dx$$

where the integration is over the entire fracture length.

Now suppose for a moment that the length expansion rate is a single power law $a = mt^n$ or $x = m\tau^n$, with $a_1 = mt_1^n$. Then

$$V_{\ell 1} = (4HCmn) \int_0^{t_1} \tau^{n-1} \sqrt{t_1 - \tau} \quad d\tau$$

Substituting $y = \tau/t_1$ gives

$$\begin{aligned} V_{\ell 1} &= 4HC \, mn \, t_1^{n+\frac{1}{2}} \int_0^1 y^{n-1} \sqrt{1-y} \, dy \\ &= 4HCn \, a_1 \sqrt{t_1} \, f_1 \end{aligned} \quad (C5)$$

$$\text{where } f_1 = \int_0^1 y^{n-1} \sqrt{1-y} \, dy$$

Since spurt loss has units of volume per area of fracture surface, the total spurt loss over one wing is

$$V_s = 2 S_p H a_1 \quad (C6)$$

where the 2 accounts for front and back fracture surfaces.

Altogether, the total fluid loss per wing is the sum of (C5) and (C6)

$$V_{\ell} = 2S_p H a_1 + 4HC \, n \, a_1 \sqrt{t_1} \, f_1 \quad (C7)$$

Now this is added to the actual volume of the fracture (1 wing), computed from the cross-section areas corresponding to each vertical segment (the areas come from properly calculated widths). The total volume is then divided by the inlet flow rate to find the time t'_1 corresponding to the fracture length a_1 . If t'_1 is not equal to t_1 (assumed in the choice of $a = mt^n$ and $a_1 = mt_1^n$), the value of m, n are adjusted to give a time between t'_1 and t_1 .

In practice, simultaneous calculations are made for fracture lengths a_1 , a_2 , and a_3 , and then adjustments are made to m_1 , n_1 , and m_2 , n_2 in (C3). First, (C7) needs to be applied to a_1 , a_2 , and a_3 :

$$\begin{aligned} V_{\ell} &= 2S_p H a_1 + 4HC n_1 a_1 \sqrt{t_1} f_1 & \text{for } a &= a_1 \\ &= 2S_p H a_2 + 4HC n_1 a_2 \sqrt{t_2} f_2 & \text{for } a &= a_2 \\ &= 2S_p H a_3 + 4HC \left[n_1 a_2 \sqrt{t_3} \frac{t_3}{t_2} n_1 f_2 + n_2 a_3 \sqrt{t_3} f_3 \right] & \text{for } a &= a_3 \end{aligned} \quad (C8)$$

where $f_2 = \int_0^{t_2/t_4} y^{n_1-1} \sqrt{1-y} dy$, $f_3 = \int_{t_2/t_4}^1 y^{n_2-1} \sqrt{1-y} dy$

(b) Iteration on $q(x)$

The flow rate at a point $x=x'$ comes from integration of the continuity equation (4) from the wellbore to x' :

$$\begin{aligned} - \int_{q(0)}^{q(x')} dq &= \int_0^{x'} \left[\lambda(x) + \frac{dA(x)}{dt} \right] dx \\ q(0) - q(x') &= \int_0^{x'} \frac{2HC}{\sqrt{t_1 - \tau(x)}} dx + \frac{1}{dt} \int_0^{x'} [A_2(x) - A_1(x)] dx \end{aligned}$$

where for the moment we again assume a single power law $x = m\tau^n$, $a_1 = mt_1^n$.

The second integral may be written $V_2(x') - V_1(x')$, or the volume of fracture $(a + da)$ minus the volume of fracture (a) , both measured from wellbore to $x = x'$. dt is the time difference between the two fractures. For the first integral

$$\int_0^{x'} \frac{dx}{\sqrt{t_1 - \tau(x)}} = mn \int_0^{\tau'} \frac{\tau^{n-1} d\tau}{\sqrt{t_1 - \tau}}$$

substituting $y = \tau/t_1$ gives

$$\int_0^{x'} = mnt_1^{n-\frac{1}{2}} \int_0^{\tau'/t_1} \frac{y^{n-1} dy}{\sqrt{1-y}} = \frac{na_1}{\sqrt{t_1}} \int_0^{\tau'/t_1} \frac{y^{n-1} dy}{\sqrt{1-y}}$$

Altogether,

$$q(x') = q(0) - \frac{V_2(x') - V_1(x')}{dt} - \frac{2HCna_1}{\sqrt{t_1}} g(\tau'/t_1) \quad (C9)$$

For example, consider the flow rate at the fracture tip, $x = a_1$. It is non-zero because the premise of our calculations is that a fracture expands from a length, a_1 , to $a_1 + da_1$ in a time dt . Firstly though, spurt loss, which only occurs in the tip segment, needs to be added. In (C2), da/dt at a_1 comes from $x = m\tau^n$:

$$\left. \frac{da}{dt} \right|_{a_1} = \left. mnt^{n-1} \right|_{x=a_1} = mnt_1^{n-1} = \frac{na_1}{t_1}$$

Hence

$$q(a_1) = q(0) - \left[\frac{V_2(a_1) - V_1(a_1)}{dt} \right] - \frac{2HCna_1}{\sqrt{t_1}} g_1 - 2S_p^{Hn_1} \frac{a_1}{t_1} \quad (C10)$$

$$\text{where } g_1 = \int_0^1 \frac{y^{n-1} dy}{\sqrt{1-y}}$$

As before with total volume leaked off, the result needs to be modified when applied to the 3 fracture lengths: a_1, a_2, a_3 :

$$\begin{aligned} q(a_1) &= q(0) - \left[\frac{V_2(a_1) - V_1(a_1)}{dt} \right] - \frac{2HCn_1 a_1 g_1}{\sqrt{t_1}} - 2S_p^{Hn_1} \frac{a_1}{t_1} \quad \text{for } a = a_1 \\ &= q(0) - \left[\frac{V_2(a_2) - V_1(a_2)}{dt} \right] - \frac{2HCn_1 a_2 g_1}{\sqrt{t_2}} - 2S_p^{Hn_1} \frac{a_2}{t_2} \quad \text{for } a = a_2 \\ &= q(0) - \left[\frac{V_2(a_3) - V_1(a_3)}{dt} \right] - 2HC \left[\frac{n_1 a_2 g_2}{\sqrt{t_3}} \left(\frac{t_3}{t_2} \right)^{n_1} + \frac{n_2 a_3 g_3}{\sqrt{t_3}} \right] \\ &\quad - 2S_p^{Hn_2} \frac{a_3}{t_3} \quad \text{for } a = a_3 \quad (C11) \end{aligned}$$

where

$$g_2 = \int_0^{t_2/t_4} \frac{y^{n_1-1} dy}{\sqrt{1-y}}, \quad g_3 = \int_{t_2/t_4}^1 \frac{y^{n_2-1} dy}{\sqrt{1-y}}$$

APPENDIX D

Width calculations for elongated fracture in three layers with asymmetric stresses and vertical gradients of stress and fluid pressure

As described earlier, the fracture is divided into a number of vertical sections, and each section is assumed to be independent of neighboring sections. That is, each vertical section is regarded as a 2D line crack which may penetrate into regions of higher minimum in-situ stress. The stresses are assumed to change discontinuously from one layer to another. Figure 9 is a sketch of the situation for which lithostatic changes of stress and hydrostatic changes of pressure are neglected, and $S_1 < S_2 < S_3$. The coordinate system is shifted so that its origin is at the center of the crack and the y direction is positive downwards.

When lithostatic gradients in stress and hydrostatic gradients in fluid pressure are present, as well as discontinuous stress changes at the interfaces between payzone and bounding zones, the net pressure (fluid pressure minus minimum in-situ stress) may be written as

$$p(y) = \begin{cases} p_2 - p_{02} y & -\ell < y < y_3 \\ p_1 - p_{01} y & y_3 < y < y_4 \\ p_3 - p_{03} y & y_4 < y < \ell \end{cases} \quad (D1)$$

The gradient terms p_{01} , p_{02} , and p_{03} are found by combining the stress and fluid pressure gradients in each layer (the stress gradients can be different in each layer). The fluid pressure, which is obtained by integrating the fluid pressure drop, dp/dx , from the tip of the elongated fracture in the payzone back towards the wellbore, is assumed to be the pressure in the center of the payzone. This defines p_1 , since the stresses are known in each layer. p_2 is found from the two values in (D1) for $p(y)$ at the upper interface $y = y_3$, such

that the difference between them equals the given stress contrast at the interface. p_3 is computed in similar fashion from the two values of $p(y)$ at the lower interface.

Figure 10 illustrates the net pressures and their gradients within the vertical crack section of Figure 9 corresponding to (D1).

The results of England and Green¹⁶ (see also Daneshy¹⁷) may be used to give the full width $w(y)$ of the crack at any height y :

$$w(y) = \frac{-16(1-\nu^2)}{E} \int_{|y|}^{\ell} \frac{F(t) + y G(t)}{\sqrt{t^2 - y^2}} dt \quad (D2)$$

Here ℓ is crack half-length. The functions $F(t)$ and $G(t)$ are obtained from the net pressure in (D1) by expressing it, in the new coordinate system, by the sum of an even and an odd function of y :

$$p(y) = f(y) + g(y) \quad -\ell < y < \ell \quad (D3)$$

Then F and G are given by:

$$F(t) = \frac{-t}{2\pi} \int_0^t \frac{f(y) dy}{\sqrt{t^2 - y^2}} \quad (D4)$$

$$G(t) = \frac{-1}{2\pi t} \int_0^t \frac{y g(y) dy}{\sqrt{t^2 - y^2}} \quad (D5)$$

In general, we may write (see also Figure 5):

$$\begin{aligned} f(y) &= \begin{cases} f_1 + f_{01} y & 0 < y < -y_3 \\ f_2 + f_{02} y & -y_3 < y < y_4 \\ f_3 + f_{03} y & y_4 < y < \ell \end{cases} \\ g(y) &= \begin{cases} g_1 + g_{01} y & 0 < y < -y_3 \\ g_2 + g_{02} y & -y_3 < y < y_4 \\ g_3 + g_{03} y & y_4 < y < \ell \end{cases} \end{aligned} \quad (D6)$$

where the constants f_1 , f_{01} , etc. are determined as shown later in this appendix. Since $f(y)$ and $g(y)$ are even and odd functions, respectively, we only have to define them for positive y .

Now $F(t)$ and $G(t)$ can be evaluated from (D4) and (D5):

$$\frac{-2\pi F(t)}{t} = \frac{\pi f_1}{2} + f_{01}t \quad \text{for } 0 < t < -y_3 \quad (D7a)$$

$$\begin{aligned} \text{or } \frac{\pi f_1}{2} + f_{01}t + (f_2 - f_1) \cos^{-1} \left(\frac{-y_3}{t} \right) \\ + (f_{02} - f_{01}) \sqrt{t^2 - y_3^2} \quad \text{for } -y_3 < t < y_4 \end{aligned} \quad (D7b)$$

$$\begin{aligned} \text{or } \frac{\pi f_1}{2} + f_{01}t + (f_2 - f_1) \cos^{-1} \left(\frac{-y_3}{t} \right) \\ + (f_3 - f_2) \cos^{-1} \left(\frac{y_4}{t} \right) \\ + (f_{02} - f_{01}) \sqrt{t^2 - y_3^2} + (f_{03} - f_{02}) \sqrt{t^2 - y_4^2} \quad \text{for } y_4 < t < \ell \end{aligned} \quad (D7c)$$

$$-2\pi t G(t) = g_1 t + \frac{\pi}{4} g_{01} t^2 \quad \text{for } 0 < t < -y_3 \quad (D8a)$$

$$\begin{aligned} \text{or } g_1 t + \frac{\pi}{4} g_{02} t^2 + (g_2 - g_1) \sqrt{t^2 - y_3^2} \\ + (g_{02} - g_{01}) \left[\frac{-y_3}{2} \sqrt{t^2 - y_3^2} - \frac{t^2}{2} \sin^{-1} \left(\frac{-y_3}{t} \right) \right] \quad \text{for } -y_3 < t < y_4 \end{aligned} \quad (D8b)$$

$$\begin{aligned} \text{or } g_1 t + \frac{\pi}{4} g_{03} t^2 + (g_2 - g_1) \sqrt{t^2 - y_3^2} + (g_3 - g_2) \sqrt{t^2 - y_4^2} \\ + (g_{02} - g_{01}) \left[\frac{-y_3}{2} \sqrt{t^2 - y_3^2} - \frac{t^2}{2} \sin^{-1} \left(\frac{-y_3}{t} \right) \right] \\ + (g_{03} - g_{02}) \left[\frac{y_4}{2} \sqrt{t^2 - y_4^2} - \frac{t^2}{2} \sin^{-1} \left(\frac{y_4}{t} \right) \right] \quad \text{for } y_4 < t < \ell \end{aligned} \quad (D8c)$$

Rewriting (D2) as

$$\begin{aligned} \frac{\pi E w(y)}{8(1-v^2)} &= \int_{|y|}^{\ell} \frac{t(-2\pi F(t)/t) dt}{\sqrt{t^2 - y^2}} + y \int_{|y|}^{\ell} \frac{-2\pi t G(t) dt}{t \sqrt{t^2 - y^2}} \\ &= w_1(y) + y w_2(y) \end{aligned}$$

allows $w_1(y)$ and $w_2(y)$ to be evaluated from (D7) and (D8). The evaluation depends on the position of y , with the integrands chosen appropriately from (D7) and (D8).

$$\text{Case 1: } 0 < |y| < -y_3 \int_{|y|}^{\ell} = \int_{|y|}^{-y_3} + \int_{-y_3}^{y_4} + \int_{y_4}^{\ell}$$

$$w_1(y) = WCONA1 + WCONA2 + WCONA3$$

$$= \int_{|y|}^{-y_3} \frac{(A8a)t dt}{\sqrt{t^2 - y^2}} + \int_{-y_3}^{y_4} \frac{(A8b)t dt}{\sqrt{t^2 - y^2}} + \int_{y_4}^{\ell} \frac{(A8c)t dt}{\sqrt{t^2 - y^2}}$$

$$w_2(y) = WCONB1 + WCONB2 + WCONB3$$

$$= \int_{|y|}^{-y_3} \frac{(A9a)dt}{t\sqrt{t^2 - y^2}} + \int_{-y_3}^{y_4} \frac{(A9b)dt}{t\sqrt{t^2 - y^2}} + \int_{y_4}^{\ell} \frac{(A9c)dt}{t\sqrt{t^2 - y^2}} \quad (D9)$$

$$\text{Case 2: } -y_3 < |y| < y_4 \int_{|y|}^{\ell} = \int_{|y|}^{y_4} + \int_{y_4}^{\ell}$$

$$w_1(y) = WCONA2 + WCONA3$$

$$= \int_{|y|}^{y_4} \frac{(A8b)t dt}{\sqrt{t^2 - y^2}} + \int_{y_4}^{\ell} \frac{(A8c)t dt}{\sqrt{t^2 - y^2}}$$

$$w_2(y) = WCONB2 + WCONB3$$

$$= \int_{|y|}^{y_4} \frac{(A9b)dt}{t\sqrt{t^2 - y^2}} + \int_{y_4}^{\ell} \frac{(A9c)dt}{t\sqrt{t^2 - y^2}} \quad (D10)$$

$$\text{Case 3: } y_4 < |y| < \ell$$

$$w_1(y) = WCONA3 = \int_{|y|}^{\ell} \frac{(A8c)t dt}{\sqrt{t^2 - y^2}}$$

$$w_2(y) = WCONB3 = \int_{|y|}^{\ell} \frac{(A9c)dt}{t\sqrt{t^2 - y^2}} \quad (D11)$$

It may be shown that all these integrations can be expressed in terms of six separate integrals, all of which can be obtained in closed form:

$$\text{FUN 1: } \int \frac{dt}{\sqrt{t^2-y^2}} = \ln(t + \sqrt{t^2-y^2})$$

$$\text{FUN 2: } \int \frac{t dt}{\sqrt{t^2-y^2}} = \sqrt{t^2-y^2}$$

$$\text{FUN 3: } \int \frac{t^2 dt}{\sqrt{t^2-y^2}} = \frac{t\sqrt{t^2-y^2}}{2} + \frac{y^2}{2} \ln(t + \sqrt{t^2-y^2})$$

$$\text{FUN 4: } \int \frac{t \cos^{-1}(d/t) dt}{\sqrt{t^2-y^2}} = \sqrt{t^2-y^2} \cos^{-1}(d/t)$$

$$\begin{aligned} & -d \ln(\sqrt{t^2-y^2} + \sqrt{t^2-d^2}) \\ & + \frac{y^2 d}{|y d|} \ln \left(\frac{\sqrt{\frac{t^2-y^2}{y^2}} + \sqrt{\frac{t^2-d^2}{d^2}}}{t} \right) \end{aligned}$$

$$\begin{aligned} \text{FUN 5: } \int \frac{t \sqrt{t^2-d^2} dt}{\sqrt{t^2-y^2}} &= \frac{1}{2} \sqrt{(t^2-d^2)(t^2-y^2)} \\ &+ \left(\frac{y^2-d^2}{2} \right) \ln(\sqrt{t^2-d^2} + \sqrt{t^2-y^2}) \end{aligned}$$

$$\begin{aligned} \text{FUN 6: } \int \frac{\sqrt{t^2-d^2} dt}{t \sqrt{t^2-y^2}} &= \ln(\sqrt{t^2-d^2} + \sqrt{t^2-y^2}) \\ &- \frac{d^2}{|y d|} \ln \left(\frac{\sqrt{\frac{t^2-d^2}{d^2}} + \sqrt{\frac{t^2-y^2}{y^2}}}{t} \right) \end{aligned}$$

To illustrate the width formulations in terms of these integrals, we include just a couple of them here:

$$\text{Case 1: } WCONA1 = 2\pi[(f_1/4) \text{ FUN 2 } (-y_3, |y|) \\ + (f_{01}/2\pi) \text{ FUN 3 } (-y_3, |y|)]$$

Here $-y_3$ and $|y|$ are the limits on the integral.

$$WCONB2 = g_1 \text{ FUN 1 } (y_4, -y_3) + \pi/4 g_{02} \\ \cdot \text{FUN 2 } (y_4, -y_3) + (g_2 - g_1) \text{ FUN 6 } (y_4, -y_3) \\ + (g_{02} - g_{01})(-y_3/2) \text{ FUN 6 } (y_4, -y_3) + (g_{01} - g_{02})/2 \\ \cdot [\pi/2 \text{ FUN 2 } (y_4, -y_3) - \text{FUN 4 } (y_4, -y_3)]$$

Next, we consider the derivation of the constants f_1 , f_{01} , etc. in equation (D6). To resolve $p(y)$ into an even function $f(y)$ and an odd function $g(y)$, divide the y -axis into pairs, 3 and 4, 2 and 5, 1 and 6, as shown in Figure 10. Then operate on each pair — we use 1 and 6 to illustrate — as follows:

(a) reflect 1 in $y = 0$ giving

$$\begin{array}{ll} p(y) = p_2 + p_{02}y & \text{1 reflected in 6} \\ \text{and } p_3 - p_{03}y & \text{already in 6} \end{array}$$

(b) take the difference and divide by 2, giving

$$\frac{p_2 - p_3}{2} + \left(\frac{p_{02} + p_{03}}{2} \right) y$$

(c) add this to $p(y)$ in 6, giving $f(y)$:

$$f(y) = \frac{p_2 + p_3}{2} + \left(\frac{p_{02} - p_{03}}{2} \right) y \quad y_4 < y < \ell$$

which is an even function in regions 1 and 6.

(d) $g(y)$ in 6 is the negative of (b):

$$g(y) = \frac{p_3 - p_2}{2} - \left(\frac{p_{02} + p_{03}}{2} \right) y \quad y_4 < y < \ell$$

Repeating this procedure for regions 2 and 5, and 3 and 4 leads to the coefficients in equation (D6) in terms of the net pressure parameters:

$$f_1, f_2, f_3 = p_1, \frac{p_1 + p_2}{2}, \frac{p_2 + p_3}{2}$$

$$f_{01}, f_{02}, f_{03} = 0, \frac{p_{02} - p_{01}}{2}, \frac{p_{02} - p_{03}}{2}$$

$$g_1, g_2, g_3 = 0, \frac{p_1 - p_2}{2}, \frac{p_3 - p_2}{2}$$

(D15)

$$g_{01}, g_{02}, g_{03} = -p_{01}, -\left(\frac{p_{01} + p_{02}}{2}\right), -\left(\frac{p_{02} + p_{03}}{2}\right)$$

These coefficients are used in the computation of $w(y)$.

Finally, the discussion so far has assumed the center of the vertical section lies within the payzone. If the fracture extends upwards far enough (see Figure 9), the center of the crack will lie in the layer above the payzone, as shown in Figure 11 y_3 becomes positive, and $f(y)$ and $g(y)$ will still be defined as in (D6), but over three regions which are different from before: $0 \rightarrow y_3$, $y_3 \rightarrow y_4$, $y_4 \rightarrow \ell$. The results are:

$$\begin{aligned} f_1, f_2, f_3 &= p_2, \frac{p_1 + p_2}{2}, \frac{p_2 + p_3}{2} \\ f_{01}, f_{02}, f_{03} &= 0, -\left(\frac{p_{01} - p_{02}}{2}\right), -\left(\frac{p_{03} - p_{02}}{2}\right) \\ g_1, g_2, g_3 &= 0, \frac{p_1 - p_2}{2}, \frac{p_3 - p_2}{2} \\ g_{01}, g_{02}, g_{03} &= -p_{02}, -\left(\frac{p_{01} + p_{02}}{2}\right), -\left(\frac{p_{02} + p_{03}}{2}\right) \end{aligned} \quad (D16)$$

Lastly, the equations (D7)-(D11) can still be used provided we replace the old limit $-y_3$ by the new limit y_3 , since y_3 is now positive.

APPENDIX E

Fracture expansion criteria for three-layer case with asymmetric stresses and vertical gradients of stress and fluid pressure

The fracture expansion criterion we use is $K_i = K_c$, where K_i is stress-intensity factor and K_c is fracture toughness. For a highly elongated fracture and relatively slow vertical expansion, we assume no pressure gradients due to flow in the vertical direction. We assume each vertical section of the elongated fracture acts as a 2D line crack, in which case K_i is given by¹⁸

$$K_i = \frac{1}{\sqrt{\pi\ell}} \int_{-\ell}^{+\ell} p(y) \left(\frac{\ell + y}{\ell - y} \right)^{\frac{1}{2}} dy \quad (E1)$$

At the tip c of the crack (see Figure 9) K_{ic} can be written as

$$\sqrt{\pi\ell} K_{ic} = \int_{-\ell}^{y_3} + \int_{y_3}^{y_4} + \int_{y_4}^{\ell} p(y) \left(\frac{\ell + y}{\ell - y} \right)^{\frac{1}{2}} dy \quad (E2)$$

with the net pressure, $p(y)$, given by equation (D1) of Appendix D. The form of each integral is

$$\int_{u_1}^{u_2} (q - ry) \left(\frac{\ell + y}{\ell - y} \right)^{\frac{1}{2}} dy \quad (E3)$$

which reduces to

$$\begin{aligned} & \left(q\ell - \frac{r\ell^2}{2} \right) \left[\sin^{-1} \left(\frac{u_2}{\ell} \right) - \sin^{-1} \left(\frac{u_1}{\ell} \right) \right] \\ & + \left(q\ell - \ell^2 r - \frac{u_1 r \ell}{2} \right) \sqrt{\frac{1 - u_1^2}{\ell^2}} - \left(q\ell - \ell^2 r - \frac{u_2 r \ell}{2} \right) \sqrt{\frac{1 - u_2^2}{\ell^2}} \end{aligned} \quad (E4)$$

and this is encoded to compute K_{ic} for given ℓ , $p(y)$. To calculate K_{ib} , repeat the approach above:

$$\sqrt{\pi\ell} K_{ib} = - \int_{\ell}^{y_4} - \int_{y_4}^{y_3} - \int_{y_3}^{-\ell} p(y) \left(\frac{\ell - y}{\ell + y} \right)^{\frac{1}{2}} dy \quad (E5)$$

with $p(y)$ still given by (D1). This is encoded in the manner of K_{ic} above.

In practice we know the net pressure $p(x)$ at the center of the payzone at each step of the integration from the square tip back to the wellbore. The job is to set K_{ic} , K_{ib} equal to K_c , and to solve (E2) and (E5) for $-\ell$, $+\ell$, or alternatively, for b, c in Figure 9 (we equated b, c to $h_u(x)$, $h_\ell(x)$ in the main text). The solution is found by iteration.

APPENDIX F

Fortran code and user instructions for vertically symmetric simulator

1. Program description

SYMFRAC1, a fracture simulator in which fracture height is self-determined is an extension of FL5, published previously.²¹ It is appropriate if the vertical fracture is highly elongated (length/height ratio $\gtrsim 4$), and if the minimum stresses in the two bounding zones are higher than in the payzone, and equal. Vertical lithostatic or hydrostatic gradients are not included. The fluid flow is assumed to be 1D along the payzone direction. Variations in elastic modulus from one zone to another are not included, nor are time variations in pumping rate or viscosity at the wellbore.

The program solves simultaneously the four equations in the text: (1)-(4), subject to the boundary condition that the fracture height at the tip of the long axis is approximately equal to the payzone height.

In this way it solves for 2 closely-spaced fractures at each of 3 stations (or half-lengths), called in the program A1, A2, and A4.

2. Program listing

Main program
SYMFRAC1

Subroutines
WIDTH
RK2
VLTQ
QTEST

Functions
FUN
DFX
FFX11
ASIN
ACOS

3. Operating mode

Batch

4. Language

Fortran IV

5. Program developed on

Data General Eclipse minicomputer

6. Total CP time

About 2 min

7. Effect of step size on Runge-Kutta integration

The resolution in all 10 segments along the x-axis is determined by NDIV (normally 110). The resolution in the tip segment is enhanced by the factor MFAC (e.g., MFAC = 5 means steps will be 5 times closer together). If the steps are not close enough in the first segment, the pressures along the fracture, given by PXA values, will be too similar and clearly erroneous. In this case, reduce the step size by increasing MFAC by a factor of 2. As a final check on the results, double NDIV, which reduces the step spacing along the entire x-axis. The results for wellbore half-height, b, and wellbore pressure (maximum PXA value) should not change significantly.

8. Too rapid height growth near the wellbore

For certain combinations of input parameters the program may abort because height grows too fast as the integration proceeds from the tip at $x=a$ back to the wellbore. When this happens, heights and pressures will be printed out from the tip to some point between tip and wellbore, followed by a RUNTIME error. If the height is growing rapidly, this is almost certainly the reason. To check it, double the stress contrast S_2-S_1 and a successful simulation should result.

Parameters which can give this error include:

(i) too small S_2-S_1	(ii) too small H	(iii) too high E
(iv) too large μ	(iv) too large Q	

9. Sample input data file

```

2_SELECTS CALCULATIONS, ONE=TIME, TWO=TIME & Q, THREE=NEITHER
1_IWRITE - FLAG USED TO SELECT OUTPUT
0.0      KCB = FRACTURE TOUGHNESS (DYNES/CM**ONE POINT FIVE)
75.0     HZ(FT) - PAYZONE HEIGHT
15.0     QIN(BPM) - INPUT FLOW RATE (ONE WING)
1.0      FLN = NPRIME = INDEX OF POWER LAW FLUID FLOW
1.0      FLOWK = KPRIME=AMPLITUDE OF FLUID FLOW(DYNES-SEC**NPRIME/CM**TWO)

5760000.0 EMOD(Psi) - YOUNG'S MODULUS
0.2      RNU - POISSON'S RATIO

1200.0    S2MS1(Psi) - STRESS CONTRAST - SAME FOR BOTH BARRIERS

500.0     A1(FT) - THESE ARE THREE FRACTURE HALF_LENGTH STATIONS USUALLY
1200.0     A2(FT)
3200.0     A4(FT) - A4 IS DESIRED FINAL FRAC LENGTH
0.00394    LEAKC - LEAKOFF COEFFICIENT (CM/SEC**POINT FIVE)
5.0      DFT(%) - PERCENT DIFFERENCE IN L(T) FIRST ESTIMATE (USUALLY FIVE)

8_IRUNZ - IRUNZ=EIGHT(USUALLY) => ONE ITERATION WITH TWO LINE SEGMENTS

1.0      Q1 - QX AT X/A = ZERO Q1 Q2 Q3 ARE USED TO CALC INPUT FLOW RATE
0.2      Q2 - QX AT X/A = NINE TENTHS
0.1      Q3 - QX AT XA = ONE

1_SELECTS OPTION TO RECOMPUTE AND LIST ALL KI'S FOR CHECKING PURPOSES

5.0      PERCENTAGE FOR QX CONVERGENCE (USUALLY FIVE)
110_NDIV - # OF DIVISION TO SPLIT FRACTURE INTO(LENGTHWISE)
0.999    XAZ - XA POSITION AT WHICH NUMERICAL INTEGRATION BEGINS
30.0     YZF(Psi) - PRESSURE AT FRACTURE TIP (DUMMY VALUE NOT USED IN CALCN)
11_N - # OF SEGMENTS FRACTURE WING IS SPLIT INTO(LENGTHWISE)

1_NRES
5_MFAC_MULT FACTOR USED TO INCREASE RESOLUTION OF TIP SEG (USUALLY FIVE)

40_NODIV - # OF STEPS FOR NUMERICAL INTEGRATIONS (NOT USED NOW)
10_NUMWID - # OF WIDTH CALCULATIONS IN EACH VERTICAL FRACTURE SECTION

1.01     HSTART/HZ = FRAC HEIGHT / PAYZONE HEIGHT AT FRAC TIP X=A
0.0      SP- SPURT LOSS COEFF. IN CM CUBED / CM SQUARED
0-IMU IS 0 MEANS KPRIME IS CONSTANT WITH X OTHERWISE ITS PROP TO(ONE-XA)

```

10. Sample output data file

Solutions for 3 fracture lengths are obtained simultaneously--A1, A2, and A4. Since there are several iterations, the last section that contains results for A1=500 ft, A2=1200 ft, and A4=3200 ft represents the most converged results.

Essentially, there are two fractures at each station--in this case 500 and 550 ft; 1200 and 1320 ft, and 3200 and 3520 ft. For each of these 6 fractures are given net pressure above payzone stress in psi (see 11 or 12 PXA values), half-height at the wellbore in ft(b), fracture full height in units of payzone height (see 11 or 12 H/HZ values), and width profile in the vertical section through the wellbore, in cm (see 11 WIDTH values).

After each pair of fractures, we calculate flow rate profile (see OUTPUT QX), and this can be compared with the flow rate assumed to solve for the fractures (see INPUT QX). Fracture volume (1 wing, cm^3) is given by VTOT1, leakoff volume (1 wing, cm^3) by TOTLEAK1, and total volume pumped by V/2 PUMPED FRAC1 (1 wing, cm^3)--all of these corresponding to the first frac at each station. The time corresponding to the first frac is given by T1CHECK.

Finally, the latest iteration on length expansion rate is given by ADJUSTED L(T) at the very end of the output file. It gives the slope and intercept for 2 line segments for a plot of $\log L$ (in ft) versus $\log t$ (in sec). That is, $TCEPT = \log_{10}(L_0)$ where L_0 is length at $t = 1$ sec. Using these values, one can plot a bilinear frac expansion rate on log-log graph paper.

OUTPUT DATA FILE

FOR SYMFRAC 1

TY FLK11.OUT
FLAMLEAK11 OUTPUT

INPUT VALUES:

ITQ= 2 IWRITE= 1
KCB= 0.000D 0 HZ= 75.0 QIN= 15.0 NPRIME = 1.000 KPRIME= 1.00000
YOUNG'S MODULUS= 0.576D 7 POISS.RATIO= 0.20
S2-S1= 1200.0
A1= 500. A2= 1200. A4= 3200.
LEAKC= 0.00394 DFT= 5.
IRUNZ= 8
Q1= 1.000Q2= 0.200Q3= 0.100
SLOPE = 1.000 1.000 0.889 0.889 0.889 0.889 0.889 0.889 0.889 0.889 0.889 0
.889
CEPT = 1.100 1.100 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.
000
ICOMP= 1 IF ICOMP=1, THEN RECOMPUTE, LIST ALL KI'S IN RANGE 0-1.
Q(X) WILL CONVERGE TO 5. PERCENT
NDIV= 110 XAZ= 0.999000 YZF= 30.000 N= 11
NRES= 1 MFAC= 5
NO. OF STEPS FOR NUM. INTEGRN.= 40 (USUALLY 40)
NO. OF WIDTH CALCNS. IN EACH VERT. SECTION= 10 (USUALLY 10)
HSTART/HZ = 1.0100
SPURT LOSS COEFF. IN CGS = 0.0000
IMU= 0 ; IMU=0 MEANS MU IS CONSTANT

CALC. ONE L(T) VALUE FOR A=A1 AND C=0.

PHI= 0.4179D 0 DELTAY= 0.1000D 0 H= 0.3716D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.621D 0 0.615D 0 0.596D 0 0.564D 0 0.514D 0 0.441D 0
0.318D 0 0.155D 0 0.739D -1 0.238D -1 0.000D 0
I = 11 , AREA(I) = 1341.9306
RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0
RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 500.0 B= 61.0

FXA:
107.59 243.72 339.06 409.68 467.32 516.54
559.75 598.23 632.98 664.71 693.77
ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 61.0A = 500.0
B= 61.0 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.0531 1.1073 1.1633 1.2215 1.2821
1.3452 1.4109 1.4794 1.5510 1.6254

PHI= 0.4179D 0 DELTAY= 0.1000D 0 H= 0.3716D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.621D 0 0.615D 0 0.596D 0 0.564D 0 0.514D 0 0.441D 0
0.318D 0 0.155D 0 0.739D -1 0.238D -1 0.000D 0
I = 11 , AREA(I) = 1341.9306
AREAS(X) AS X=A TO 0 :
.0000D 00.3450D 30.4974D 30.6204D 30.7345D 30.8366D 30.9408D 3

.1042D 40.1137D 40.1239D 40.1342D 4
VOLUME(X) AS X=A TO 0 : (CGS)

45

.2629D 60.6419D 60.8517D 60.1032D 70.1197D 70.1354D 70.1511D 7

.1661D 70.1811D 70.1967D 70.0000D 0

CALC. ONE L(T) VALUE FOR A=A4 AND C=0.

PHI= 0.2509D 0 DELTAY= 0.1000D 0 H= 0.7409D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.107D 1 0.104D 1 0.947D 0 0.726D 0 0.437D 0 0.291D 0

0.189D 0 0.113D 0 0.576D -1 0.192D -1 0.000D 0
I = 11 , AREA(I) = 3231.4427

RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0

RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????

A= 3200.0 B= 121.5

FXA:
107.59 375.51 517.63 617.23 694.89 758.15
811.28 856.84 895.98 930.39 960.38
ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 121.5A = 3200.0

B= 121.5 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.1343 1.2835 1.4472 1.6285 1.8293
2.0527 2.3029 2.5804 2.8932 3.2410

PHI= 0.2509D 0 DELTAY= 0.1000D 0 H= 0.7409D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.107D 1 0.104D 1 0.947D 0 0.726D 0 0.437D 0 0.291D 0

0.189D 0 0.113D 0 0.576D -1 0.192D -1 0.000D 0
I = 11 , AREA(I) = 3231.4427
AREAS(X) AS X=A TO 0 :

.0000D 00.5579D 30.8391D 30.1091D 40.1346D 40.1601D 40.1869D 4

.2171D 40.2479D 40.2839D 40.3231D 4
VOLUME(X) AS X=A TO 0 : (CGS)

.2721D 70.6813D 70.9414D 70.1189D 80.1437D 80.1692D 80.1970D 8

.2268D 80.2593D 80.2960D 80.0000D 0

NOW USE THESE RESULTS TO CALC. 1ST APPROX'N FOR L(T) [1 SEGMENT].

A1,SAVT1,A4,SAVT2,ALLOG,AU,TSLOPE,TCEPT =
0.5000D 3 0.3104D 3 0.3200D 4 0.4042D 4 0.2401D 1 0.4592D 5 0.5768D 0 0.
1201D 1

NOW USE 1ST APPROX'N FOR L(T) AND A=A1 RESULTS TO ADJUST L(T) VALUE AT A=A1.

VOLUME(X) AS X=A TO 0 : (CGS)

.2629D 60.6419D 60.8517D 60.1032D 70.1197D 70.1354D 70.1511D 7

.1661D 70.1811D 70.1967D 70.0000D 0

PHI= 0.4101D 0 DELTAY= 0.1000D 0 H= 0.3803D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.638D 0 0.632D 0 0.612D 0 0.577D 0 0.524D 0 0.445D 0

0.300D 0 0.151D 0 0.725D -1 0.234D -1 0.000D 0
I = 12 , AREA(I) = 1389.9717

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????

A= 550.0 B= 62.4

PXA:
107.59 238.54 331.84 401.00 457.56 505.91
548.44 586.36 620.63 651.99 680.76 707.31

ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 62.4A = 550.0

B= 62.4 PXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.0508 1.1024 1.1556 1.2108 1.2680
1.3277 1.3896 1.4540 1.5212 1.5910 1.6635

PHI= 0.4101D 0 DELTAY= 0.1000D 0 H= 0.3803D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.638D 0 0.632D 0 0.612D 0 0.577D 0 0.524D 0 0.445D 0

0.300D 0 0.151D 0 0.725D -1 0.234D -1 0.000D 0
I = 12, AREA(I) = 1389.9717
AREAS(X) AS X=A TO 0 :

.0000D 00.3370D 30.4857D 30.6041D 30.7146D 30.8131D 30.9124D 3

.1010D 40.1101D 40.1197D 40.1295D 4
VOLUME(X) AS X=A TO 0 : (CGS)

.2568D 60.6269D 60.8304D 60.1005D 70.1164D 70.1315D 70.1465D 7

.1609D 70.1751D 70.1899D 70.2046D 7

TAU(I) AS I=1 TO N+1
0.0000D 0 0.7292D 1 0.2425D 2 0.4899D 2 0.8067D 2 0.1188D 3
0.1629D 3 0.2128D 3 0.2683D 3 0.3291D 3 0.3950D 3 0.4660D 3

T(1)= 0.3950D 3, T(2)= 0.4660D 3
T1CHECK:(SEC,MIN) 0.5229D 3 0.8715D 1 T2CHECK:(SEC,MIN) 0.6066D 3 0.1011D 2
THESE TIMES ARE TOTAL PUMPING TIMES.
VTOT1 = 0.1229D 8 VTOT2 = 0.1397D 8
TOTLEAK1 = 0.8416D 7, TOTLEAK2 = 0.1006D 8
V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.2071D 8 0.1307D 3 0.5471D 4
V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.2402D 8 0.1517D 3 0.6347D 4

Z DIFF. BETWEEN T1CHECK AND T(1) = 24.450865
Z DIFF. BETWEEN T2CHECK AND T(2) = 23.182094

DT(T2CHECK-T1CHECK)= 0.8376D 2 SEC., T(2)-T(1)= 0.7098D 2 SEC.
LEAKRATE(11), QX(11) : 0.2297D 5-0.3353D 3

OUTPUT QX(I) AS X=0 TO A:
0.1500D 2 0.1412D 2 0.1318D 2 0.1222D 2 0.1121D 2 0.1010D 2
0.8917D 1 0.7596D 1 0.6056D 1 0.4101D 1 -0.1270D 0

ERROR.....ERROR.....ERROR.....
ERROR.....ERROR.....ERROR.....
FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO : 0.00

INPUT QX, AS X=0 TO A:
0.1500D 2 0.1350D 2 0.1200D 2 0.1050D 2 0.9000D 1 0.7500D 1
0.6000D 1 0.4500D 1 0.3000D 1 0.1500D 1 0.3331D-14

A= 0.500D 3 INPUT T= 0.395D 3 OUTPUT T= 0.523D 3 DIFT1= 0.324D 2 ADJ
USTED T ESTIMATE = 0.491D 3

NOW USE 1ST APPROX'N FOR L(T) AND A=A4 RESULTS TO ADJUST L(T) VALUE AT A=A4.

VOLUME(X) AS X=A TO 0 : (CGS)

.2721D 70.6813D 70.9414D 70.1189D 80.1437D 80.1692D 80.1970D 8

.2268D 80.2593D 80.2960D 80.0000D 0

PHT= 0.2377D 0 DELTAY= 0.1000D 0 H= 0.7838D 4

```

WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.111D 1 0.107D 1 0.964D 0 0.674D 0 0.429D 0 0.288D 0
47
0.187D 0 0.112D 0 0.572D -1 0.191D -1 0.000D 0
I = 12 , AREA(I) = 3416.6900
RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0
RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????
*****
A= 3520.0 B= 128.6
*****
FXA:
107.59 367.70 507.37 605.48 682.13 744.86
797.78 843.11 882.39 916.79 947.17 973.91
ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 128.6A = 3520.0
B= 128.6 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.1282 1.2699 1.4245 1.5945 1.7820
1.9900 2.2204 2.4760 2.7603 3.0775 3.4288
PHI= 0.2377D 0 DELTAY= 0.1000D 0 H= 0.7838D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.111D 1 0.107D 1 0.964D 0 0.674D 0 0.429D 0 0.288D 0
0.187D 0 0.112D 0 0.572D -1 0.191D -1 0.000D 0
I = 12 , AREA(I) = 3416.6900
AREAS(X) AS X=A TO 0 :
.0000D 00.5443D 30.8162D 30.1061D 40.1300D 40.1540D 40.1800D 4
.2072D 40.2371D 40.2684D 40.3051D 4
VOLUME(X) AS X=A TO 0 : (CGS)
.2654D 70.6635D 70.9154D 70.1151D 80.1385D 80.1629D 80.1888D 8
.2167D 80.2465D 80.2797D 80.3154D 8
TAU(I) AS I=1 TO N+1
0.0000D 0 0.1822D 3 0.6061D 3 0.1224D 4 0.2016D 4 0.2968D 4
0.4071D 4 0.5319D 4 0.6704D 4 0.8223D 4 0.9871D 4 0.1164D 5
T(1)= 0.9871D 4, T(2)= 0.1164D 5
T1CHECK:(SEC,MIN) 0.1080D 5 0.1799D 3 T2CHECK:(SEC,MIN) 0.1273D 5 0.2122D 3
THESE TIMES ARE TOTAL PUMPING TIMES.
VTOT1 = 0.1601D 9 VTOT2 = 0.1848D 9
TOTLEAK1 = 0.2674D 9, TOTLEAK2 = 0.3194D 9
V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.4275D 9 0.2699D 4 0.1129D 6
V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.5042D 9 0.3183D 4 0.1332D 6
% DIFF. BETWEEN T1CHECK AND T(1) = 8.562351
% DIFF. BETWEEN T2CHECK AND T(2) = 8.545949
DT(T2CHECK-T1CHECK)= 0.1937D 4 SEC., T(2)-T(1)= 0.1774D 4 SEC.
LEAKRATE(11), QX(11) : 0.2929D 5-0.1101D 4
OUTPUT QX(I) AS X=0 TO A:
0.1500D 2 0.1395D 2 0.1287D 2 0.1178D 2 0.1066D 2 0.9502D 1
0.8290D 1 0.6976D 1 0.5481D 1 0.3612D 1 -0.4172D 0
ERROR.....ERROR.....ERROR.....
ERROR.....ERROR.....ERROR.....
FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO : 0.00
INPUT QX, AS X=0 TO A:
0.1500D 2 0.1350D 2 0.1200D 2 0.1050D 2 0.9000D 1 0.7500D 1
0.6000D 1 0.4500D 1 0.3000D 1 0.1500D 1 0.3331D-14
A= 0.320D 4 INPUT T= 0.987D 4 OUTPUT T= 0.108D 5 DIFT4= 0.936D 1 ADJ
USTED T ESTIMATE = 0.106D 5
-----
NOW ADJUST L(T) [1 SEGMENT].
-----

```

ADJUSTED TSLOPE,TCEPT = 0.605D 0 0.107D 1
VOLUME(X) AS X=A TO 0 : (CGS)

48

.2629D 60.6419D 60.8517D 60.1032D 70.1197D 70.1354D 70.1511D 7

.1661D 70.1811D 70.1967D 70.0000D 0

PHI= 0.4101D 0 DELTAY= 0.1000D 0 H= 0.3803D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.638D 0 0.632D 0 0.612D 0 0.577D 0 0.524D 0 0.445D 0

0.300D 0 0.151D 0 0.725D -1 0.234D -1 0.000D 0
I = 12 , AREA(I) = 1389.9717

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 550.0 B= 62.4

FXA:
107.59 238.54 331.84 401.00 457.56 505.91
548.44 586.36 620.63 651.99 680.76 707.31

ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 62.4A = 550.0

B= 62.4 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.0508 1.1024 1.1556 1.2108 1.2680
1.3277 1.3896 1.4540 1.5212 1.5910 1.6635

PHI= 0.4101D 0 DELTAY= 0.1000D 0 H= 0.3803D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.638D 0 0.632D 0 0.612D 0 0.577D 0 0.524D 0 0.445D 0

0.300D 0 0.151D 0 0.725D -1 0.234D -1 0.000D 0
I = 12 , AREA(I) = 1389.9717
AREAS(X) AS X=A TO 0 :

.0000D 00.3370D 30.4857D 30.6041D 30.7146D 30.8131D 30.9124D 3

.1010D 40.1101D 40.1197D 40.1295D 4
VOLUME(X) AS X=A TO 0 : (CGS)

.2568D 60.6269D 60.8304D 60.1005D 70.1164D 70.1315D 70.1465D 7

.1609D 70.1751D 70.1899D 70.2046D 7

TAU(I) AS I=1 TO N+1
0.0000D 0 0.1091D 2 0.3431D 2 0.6707D 2 0.1079D 3 0.1561D 3
0.2110D 3 0.2722D 3 0.3395D 3 0.4124D 3 0.4909D 3 0.5747D 3

T(1)= 0.4909D 3, T(2)= 0.5747D 3
T1CHECK:(SEC,MIN) 0.5447D 3 0.9078D 1 T2CHECK:(SEC,MIN) 0.6316D 3 0.1053D 2
THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.1229D 8 VTOT2 = 0.1397D 8
TOTLEAK1 = 0.9279D 7, TOTLEAK2 = 0.1104D 8
V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.2157D 8 0.1362D 3 0.5698D 4
V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.2501D 8 0.1579D 3 0.6608D 4

% DIFF. BETWEEN T1CHECK AND T(1) = 9.867158
% DIFF. BETWEEN T2CHECK AND T(2) = 9.007958

DT(T2CHECK-T1CHECK)= 0.8693D 2 SEC., T(2)-T(1)= 0.8378D 2 SEC.

LEAKRATE(11), QX(11) : 0.2092D 5 0.2331D 4

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1418D 2 0.1332D 2 0.1243D 2 0.1148D 2 0.1045D 2

0.9344D 1 0.8109D 1 0.6667D 1 0.4836D 1 0.8830D 0

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1350D 2 0.1200D 2 0.1050D 2 0.9000D 1 0.7500D 1

0.6000D 1 0.4500D 1 0.3000D 1 0.1500D 1 0.3331D-14

A= 0.500D 7 INPUT T= 0.481D 7 OUTPUT T= 0.545D 7 DIST1= 0.100D 2 ADJ

USTED T ESTIMATE = 0.531D 3

40

NDW USE 1ST APPROX'N FOR L(T) AND A=A4 RESULTS TO ADJUST L(T) VALUE AT A=A4.

VOLUME(X) AS X=A TO 0 : (CGS)

.2721D 70.6813D 70.9414D 70.1189D 80.1437D 80.1692D 80.1970D 8

.2268D 80.2593D 80.2960D 80.0000D 0

PHI= 0.2377D 0 DELTAY= 0.1000D 0 H= 0.7838D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.111D 1 0.107D 1 0.964D 0 0.674D 0 0.429D 0 0.288D 0

0.187D 0 0.112D 0 0.572D -1 0.191D -1 0.000D 0

I = 12, AREA(I) = 3416.6900

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????

A= 3520.0 B= 128.6

PXA:

107.59 367.70 507.37 605.48 682.13 744.86

797.78 843.11 882.39 916.79 947.17 973.91

ITERATION SUCCESSFUL ON KI

Q= 15.0 RMU= 0.0H= 0.0 B= 128.6A = 3520.0

B= 128.6 PXA(1) = 7417307.9

SHAPE OF FRAC H/HZ FROM XA=1 TO 0

1.0100 1.1282 1.2699 1.4245 1.5945 1.7820

1.9900 2.2204 2.4760 2.7603 3.0775 3.4288

PHI= 0.2377D 0 DELTAY= 0.1000D 0 H= 0.7838D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.111D 1 0.107D 1 0.964D 0 0.674D 0 0.429D 0 0.288D 0

0.187D 0 0.112D 0 0.572D -1 0.191D -1 0.000D 0

I = 12, AREA(I) = 3416.6900

AREAS(X) AS X=A TO 0 :

.0000D 00.5443D 30.8162D 30.1061D 40.1300D 40.1540D 40.1800D 4

.2072D 40.2371D 40.2684D 40.3051D 4

VOLUME(X) AS X=A TO 0 : (CGS)

.2654D 70.6635D 70.9154D 70.1151D 80.1385D 80.1629D 80.1888D 8

.2167D 80.2465D 80.2797D 80.3154D 8

TAU(I) AS I=1 TO N+1

0.0000D 0 0.2347D 3 0.7383D 3 0.1443D 4 0.2322D 4 0.3359D 4

0.4540D 4 0.5858D 4 0.7305D 4 0.8875D 4 0.1056D 5 0.1237D 5

T(1)= 0.1056D 5, T(2)= 0.1237D 5

T1CHECK:(SEC,MIN) 0.1096D 5 0.1826D 3 T2CHECK:(SEC,MIN) 0.1289D 5 0.2149D 3

THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.1601D 9 VTOT2 = 0.1848D 9

TOTLEAK1 = 0.2738D 9, TOTLEAK2 = 0.3257D 9

V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.4338D 9 0.2739D 4 0.1146D 6

V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.5105D 9 0.3223D 4 0.1349D 6

% DIFF. BETWEEN T1CHECK AND T(1) = 3.569260

% DIFF. BETWEEN T2CHECK AND T(2) = 4.073207

DT(T2CHECK-T1CHECK)= 0.1937D 4 SEC., T(2)-T(1)= 0.1803D 4 SEC.

LEAKRATE(11), QX(11) : 0.2876D 5-0.5689D 3

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1397D 2 0.1291D 2 0.1184D 2 0.1074D 2 0.9599D 1

0.8404D 1 0.7106D 1 0.5627D 1 0.3777D 1 -0.2155D 0

ERROR.....ERROR.....ERROR.....

ERROR.....ERROR.....ERROR.....

FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO : 0.00

50

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1350D 2 0.1200D 2 0.1050D 2 0.9000D 1 0.7500D 1

0.6000D 1 0.4500D 1 0.3000D 1 0.1500D 1 0.3331D-14

A= 0.320D 4 INPUT T= 0.106D 5 OUTPUT T= 0.110D 5 DIFT4= 0.370D 1 ADJ
USTED T ESTIMATE = 0.109D 5

NOW ADJUST L(T) [1 SEGMENT].

ADJUSTED TSLOPE, TCEPT = 0.615D 0 0.102D 1
VOLUME(X) AS X=A TO 0 : (CGS)

.2629D 60.6419D 60.8517D 60.1032D 70.1197D 70.1354D 70.1511D 7

.1661D 70.1811D 70.1967D 70.0000D 0

PHI= 0.4101D 0 DELTAY= 0.1000D 0 H= 0.3803D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.638D 0 0.632D 0 0.612D 0 0.577D 0 0.524D 0 0.445D 0

0.300D 0 0.151D 0 0.725D -1 0.234D -1 0.000D 0
I = 12, AREA(I) = 1389.9717

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 550.0 B= 62.4

FXA:

107.59 238.54 331.84 401.00 457.56 505.91

548.44 586.36 620.63 651.99 680.76 707.31

ITERATION SUCCESSFUL ON KI

Q= 15.0 RMU= 0.0H= 0.0 B= 62.4A = 550.0

B= 62.4 FXA(1) = 7417307.9

SHAPE OF FRAC H/HZ FROM XA=1 TO 0

1.0100 1.0508 1.1024 1.1556 1.2108 1.2680

1.3277 1.3896 1.4540 1.5212 1.5910 1.6635

PHI= 0.4101D 0 DELTAY= 0.1000D 0 H= 0.3803D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.638D 0 0.632D 0 0.612D 0 0.577D 0 0.524D 0 0.445D 0

0.300D 0 0.151D 0 0.725D -1 0.234D -1 0.000D 0
I = 12, AREA(I) = 1389.9717

AREAS(X) AS X=A TO 0 :

.0000D 00.3370D 30.4857D 30.6041D 30.7146D 30.8131D 30.9124D 3

.1010D 40.1101D 40.1197D 40.1295D 4

VOLUME(X) AS X=A TO 0 : (CGS)

.2568D 60.6269D 60.8304D 60.1005D 70.1164D 70.1315D 70.1465D 7

.1609D 70.1751D 70.1899D 70.2046D 7

TAU(I) AS I=1 TO N+1

0.0000D 0 0.1258D 2 0.3882D 2 0.7505D 2 0.1198D 3 0.1722D 3
0.2316D 3 0.2975D 3 0.3696D 3 0.4476D 3 0.5312D 3 0.6202D 3

T(1)= 0.5312D 3, T(2)= 0.6202D 3

T1CHECK:(SEC,MIN) 0.5531D 3 0.9218D 1 T2CHECK:(SEC,MIN) 0.6413D 3 0.1069D 2
THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.1229D 8 VTOT2 = 0.1397D 8

TOTLEAK1 = 0.9613D 7, TOTLEAK2 = 0.1143D 8

V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.2190D 8 0.1383D 3 0.5787D 4

V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.2539D 8 0.1603D 3 0.6709D 4

% DIFF. BETWEEN T1CHECK AND T(1) = 3.955177

% DIFF. BETWEEN T2CHECK AND T(2) =

DT(T2CHECK-T1CHECK)= 0.8816D 2 SEC., T(2)-T(1)= 0.8902D 2 SEC. 51
 LEAKRATE(11), QX(11) : 0.2022D 5 0.3256D 4
 OUTPUT QX(I) AS X=0 TO A:
 0.1500D 2 0.1421D 2 0.1336D 2 0.1250D 2 0.1158D 2 0.1057D 2
 0.9492D 1 0.8286D 1 0.6879D 1 0.5091D 1 0.1234D 1
 INPUT QX, AS X=0 TO A:
 0.1500D 2 0.1350D 2 0.1200D 2 0.1050D 2 0.9000D 1 0.7500D 1
 0.6000D 1 0.4500D 1 0.3000D 1 0.1500D 1 0.3331D-14
 A= 0.500D 3 INPUT T= 0.531D 3 OUTPUT T= 0.553D 3 DIFT1= 0.412D 1 ADJ
 USTED T ESTIMATE = 0.548D 3

 NOW USE 1ST APPROX'N FOR L(T) AND A=A4 RESULTS TO ADJUST L(T) VALUE AT A=A4.

 VOLUME(X) AS X=A TO 0 : (CGS)

.2721D 70.6813D 70.9414D 70.1189D 80.1437D 80.1692D 80.1970D 8
 .2268D 80.2593D 80.2960D 80.0000D 0
 PHI= 0.2377D 0 DELTAY= 0.1000D 0 H= 0.7838D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.111D 1 0.107D 1 0.964D 0 0.674D 0 0.429D 0 0.288D 0
 0.187D 0 0.112D 0 0.572D -1 0.191D -1 0.000D 0
 I = 12 , AREA(I) = 3416.6900
 RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0
 RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????

 A= 3520.0 B= 128.6

FXA:
 107.59 367.70 507.37 605.48 682.13 744.86
 797.78 843.11 882.39 916.79 947.17 973.91
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 R= 128.6A = 3520.0
 B= 128.6 FXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.1282 1.2699 1.4245 1.5945 1.7820
 1.9900 2.2204 2.4760 2.7603 3.0775 3.4288

PHI= 0.2377D 0 DELTAY= 0.1000D 0 H= 0.7838D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.111D 1 0.107D 1 0.964D 0 0.674D 0 0.429D 0 0.288D 0
 0.187D 0 0.112D 0 0.572D -1 0.191D -1 0.000D 0
 I = 12 , AREA(I) = 3416.6900
 AREAS(X) AS X=A TO 0 :
 .0000D 00.5443D 30.8162D 30.1061D 40.1300D 40.1540D 40.1800D 4
 .2072D 40.2371D 40.2684D 40.3051D 4
 VOLUME(X) AS X=A TO 0 : (CGS)
 .2654D 70.6635D 70.9154D 70.1151D 80.1385D 80.1629D 80.1888D 8
 .2167D 80.2465D 80.2797D 80.3154D 8

TAU(I) AS I=1 TO N+1
 0.0000D 0 0.2572D 3 0.7935D 3 0.1534D 4 0.2448D 4 0.3519D 4
 0.4733D 4 0.6080D 4 0.7554D 4 0.9149D 4 0.1086D 5 0.1268D 5
 T(1)= 0.1086D 5, T(2)= 0.1268D 5
 T1CHECK:(SEC,MIN) 0.1102D 5 0.1837D 3 T2CHECK:(SEC,MIN) 0.1296D 5 0.2160D 3
 THESE TIMES ARE TOTAL PUMPING TIMES.
 VTOT1 = 0.1601D 9 VTOT2 = 0.1848D 9
 TOTLEAK1 = 0.2765D 9, TOTLEAK2 = 0.3285D 9
 U/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.4365D 9 0.2756D 4 0.1153D 6
 U/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.5177D 9 0.3241D 4 0.1554D 6

```

% DIFF. BETWEEN T1CHECK AND T(1) = 1.507949
% DIFF. BETWEEN T2CHECK AND T(2) = 2.205986
52
DT(T2CHECK-T1CHECK)= 0.1939D 4 SEC., T(2)-T(1)= 0.1819D 4 SEC.
LEAKRATE(11), QX(11) : 0.2853D 5-0.3246D 3
OUTPUT QX(I) AS X=0 TO A:
0.1500D 2 0.1398D 2 0.1293D 2 0.1186D 2 0.1077D 2 0.9640D 1
0.8453D 1 0.7162D 1 0.5692D 1 0.3850D 1 -0.1229D 0
ERROR.....ERROR.....ERROR.....
ERROR.....ERROR.....ERROR.....
FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO : 0.00
INPUT QX, AS X=0 TO A:
0.1500D 2 0.1350D 2 0.1200D 2 0.1050D 2 0.9000D 1 0.7500D 1
0.6000D 1 0.4500D 1 0.3000D 1 0.1500D 1 0.3331D-14
A= 0.320D 4 INPUT T= 0.109D 5 OUTPUT T= 0.110D 5 DIFT4= 0.153D 1 ADJ
USTED T ESTIMATE = 0.110D 5
-----
NOW ADJUST L(T) [1 SEGMENT].
-----
ADJUSTED TSLOPE,TCEPT = 0.619D 0 0.100D 1
ITERATION IN IRUN=3 STOPPED BECAUSE TIME DIFFERENCES DIFT1= 4.1 OR DIFT
2= 1.5
ARE BOTH LESS THAN 5.0 PERCENT, OR ELSE NO. OF ITERATIONS 3 EXCEEDS 4
-----
USE THIS ADJUSTED L(T) [1 SEGMENT] TO DO FULL LEAKOFF CALCN
FOR A1 THEN A2 THEN A4. FROM THE TIME DISCREPANCY WE COMPUTE
AN ADJUSTED L(T) VALUE (=TA).
-----
PHI= 0.3999D 0 DELTAY= 0.1000D 0 H= 0.3926D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.661D 0 0.654D 0 0.633D 0 0.595D 0 0.536D 0 0.448D 0
0.273D 0 0.146D 0 0.0708D -1 0.229D -1 0.000D 0
I = 11 , AREA(I) = 1455.9300
RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0
RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????
*****
A= 500.0 B= 64.4
*****
FXA:
107.59 324.23 420.08 484.93 535.73 578.01
614.43 646.61 675.37 701.39 725.14
ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 64.4A = 500.0
B= 64.4 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.0974 1.1729 1.2419 1.3089 1.3752
1.4417 1.5090 1.5772 1.6466 1.7174
PHI= 0.3999D 0 DELTAY= 0.1000D 0 H= 0.3926D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.661D 0 0.654D 0 0.633D 0 0.595D 0 0.536D 0 0.448D 0
0.273D 0 0.146D 0 0.0708D -1 0.229D -1 0.000D 0
I = 11 , AREA(I) = 1455.9300
AREAS(X) AS X=A TO 0 :
.0000D 00.4733D 30.6402D 30.7706D 30.8813D 30.9882D 30.1084D 4
.1180D 40.1276D 40.1369D 40.1456D 4
VOLUME(X) AS X=A TO 0 : (CGS)
.3606D 60.8485D 60.1075D 70.1259D 70.1425D 70.1579D 70.1725D 7
.1871D 70.2016D 70.2153D 70.0000D 0

```

PHI= 0.3928D 0 DELTAY= 0.1000D 0 H= 0.4029D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.680D 0 0.672D 0 0.649D 0 0.608D 0 0.546D 0 0.449D 0

53

0.260D 0 0.142D 0 0.695D -1 0.226D -1 0.000D 0
I = 12 , AREA(I) = 1514.2385

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????

A= 550.0 B= 66.1

PXA:
107.59 319.92 415.32 479.57 529.81 571.64
607.64 639.46 667.95 693.72 717.28 739.07

ITERATION SUCCESSFUL ON KI

Q= 15.0 RMU= 0.0H= 0.0 B= 66.1A = 550.0

B= 66.1 PXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.0946 1.1685 1.2356 1.3004 1.3645
1.4286 1.4933 1.5589 1.6253 1.6930 1.7624

PHI= 0.3928D 0 DELTAY= 0.1000D 0 H= 0.4029D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.680D 0 0.672D 0 0.649D 0 0.608D 0 0.546D 0 0.449D 0

0.260D 0 0.142D 0 0.695D -1 0.226D -1 0.000D 0
I = 12 , AREA(I) = 1514.2385
AREAS(X) AS X=A TO 0 :

.0000D 00.4662D 30.6311D 30.7597D 30.8673D 30.9715D 30.1066D 4

.1157D 40.1251D 40.1342D 40.1425D 4
VOLUME(X) AS X=A TO 0 : (CGS)

.3553D 60.8362D 60.1060D 70.1240D 70.1401D 70.1553D 70.1694D 7

.1835D 70.1975D 70.2108D 70.2240D 7

TAU(I) AS I=1 TO N+1

0.0000D 0 0.1328D 2 0.4069D 2 0.7832D 2 0.1247D 3 0.1787D 3
0.2400D 3 0.3078D 3 0.3819D 3 0.4619D 3 0.5476D 3 0.6388D 3

T(1)= 0.5476D 3, T(2)= 0.6388D 3
T1CHECK:(SEC,MIN) 0.6075D 3 0.1012D 2 T2CHECK:(SEC,MIN) 0.7039D 3 0.1173D 2
THESE TIMES ARE TOTAL PUMPING TIMES.
VTOT1 = 0.1431D 8 VTOT2 = 0.1630D 8
TOTLEAK1 = 0.9745D 7, TOTLEAK2 = 0.1158D 8
V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.2406D 8 0.1519D 3 0.6356D 4
V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.2787D 8 0.1760D 3 0.7364D 4

% DIFF. BETWEEN T1CHECK AND T(1) = 9.852548
% DIFF. BETWEEN T2CHECK AND T(2) = 9.250641

DT(T2CHECK-T1CHECK)= 0.9641D 2 SEC., T(2)-T(1)= 0.9115D 2 SEC.

LEAKRATE(11), QX(11): 0.1996D 5 0.2733D 4

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1421D 2 0.1339D 2 0.1252D 2 0.1159D 2 0.1061D 2

0.9542D 1 0.8351D 1 0.6965D 1 0.5169D 1 0.1035D 1

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1403D 2 0.1302D 2 0.1200D 2 0.1093D 2 0.9804D 1

0.8619D 1 0.7339D 1 0.5909D 1 0.4193D 1 0.9251D 0

A= 500. OLD TA= 547.6 NEW TA= 607.5 ADJUSTED TA= 592.5

PHI= 0.3292D 0 DELTAY= 0.1000D 0 H= 0.5130D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.843D 0 0.829D 0 0.788D 0 0.712D 0 0.582D 0 0.343D 0

0.212D 0 0.124D 0 0.622D -1 0.206D -1 0.000D 0
I = 11 , AREA(I) = 2100.5757

RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0

RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2 54
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

 A= 1200.0 B= 84.2

FXA:
 107.59 379.15 497.19 574.97 634.58 683.50
 724.94 761.10 792.93 821.39 847.18
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 84.2A = 1200.0

B= 84.2 FXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.1372 1.2569 1.3701 1.4828 1.5981
 1.7166 1.8402 1.9685 2.1028 2.2442

PHI= 0.3292D 0 DELTAY= 0.1000D 0 H= 0.5130D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.843D 0 0.829D 0 0.788D 0 0.712D 0 0.582D 0 0.343D 0

0.212D 0 0.124D 0 0.622D -1 0.206D -1 0.000D 0
 I = 11, AREA(I) = 2100.5757
 AREAS(X) AS X=A TO 0 :

.0000D 00.5644D 30.7946D 30.9802D 30.1142D 40.1305D 40.1455D 4

.1615D 40.1775D 40.1929D 40.2101D 4
 VOLUME(X) AS X=A TO 0 : (CGS)

.1032D 70.2485D 70.3246D 70.3881D 70.4475D 70.5047D 70.5615D 7

.6201D 70.6774D 70.7369D 70.0000D 0

PHI= 0.3204D 0 DELTAY= 0.1000D 0 H= 0.5333D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.868D 0 0.853D 0 0.808D 0 0.724D 0 0.575D 0 0.333D 0

0.208D 0 0.122D 0 0.615D -1 0.203D -1 0.000D 0
 I = 12, AREA(I) = 2207.1339

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

 A= 1320.0 B= 87.5

FXA:
 107.59 373.07 490.95 568.13 627.26 675.78
 716.87 752.75 784.37 812.62 838.28 861.54
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 87.5A = 1320.0

B= 87.5 FXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.1324 1.2492 1.3587 1.4675 1.5783
 1.6918 1.8097 1.9320 2.0592 2.1930 2.3327

PHI= 0.3204D 0 DELTAY= 0.1000D 0 H= 0.5333D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.868D 0 0.853D 0 0.808D 0 0.724D 0 0.575D 0 0.333D 0

0.208D 0 0.122D 0 0.615D -1 0.203D -1 0.000D 0
 I = 12, AREA(I) = 2207.1339
 AREAS(X) AS X=A TO 0 :

.0000D 00.5537D 30.7825D 30.9624D 30.1120D 40.1278D 40.1423D 4

.1576D 40.1731D 40.1877D 40.2038D 4
 VOLUME(X) AS X=A TO 0 : (CGS)

.1013D 70.2444D 70.3191D 70.3808D 70.4385D 70.4940D 70.5485D 7

55

.6048D 70.6598D 70.7160D 70.7764D 7

TAU(I) AS I=1 TO N+1

0.0000D 0 0.5462D 2 0.1673D 3 0.3221D 3 0.5127D 3 0.7352D 3
0.9869D 3 0.1266D 4 0.1571D 4 0.1900D 4 0.2252D 4 0.2627D 4

T(1)= 0.2252D 4, T(2)= 0.2627D 4

T1CHECK:(SEC,MIN) 0.2363D 4 0.3938D 2 T2CHECK:(SEC,MIN) 0.2757D 4 0.4595D 2

THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.4613D 8 VTOT2 = 0.5284D 8

TOTLEAK1 = 0.4743D 8, TOTLEAK2 = 0.5635D 8

V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.9356D 8 0.5906D 3 0.2472D 5

V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.1092D 9 0.6893D 3 0.2885D 5

% DIFF. BETWEEN T1CHECK AND T(1) = 4.666236

% DIFF. BETWEEN T2CHECK AND T(2) = 4.715688

DT(T2CHECK-T1CHECK)= 0.3946D 3 SEC., T(2)-T(1)= 0.3749D 3 SEC.

LEAKRATE(11), QX(11) : 0.2362D 5 0.1546D 4

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1409D 2 0.1318D 2 0.1224D 2 0.1124D 2 0.1020D 2

0.9087D 1 0.7856D 1 0.6440D 1 0.4621D 1 0.5855D 0

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1394D 2 0.1286D 2 0.1176D 2 0.1063D 2 0.9454D 1

0.8229D 1 0.6918D 1 0.5464D 1 0.3728D 1 0.4626D 0

A= 1200. OLD TA= 2252.3 NEW TA= 2362.6 ADJUSTED TA= 2335.0

PHI= 0.2317D 0 DELTAY= 0.1000D 0 H= 0.8116D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.113D 1 0.109D 1 0.973D 0 0.655D 0 0.425D 0 0.286D 0

0.186D 0 0.112D 0 0.570D -1 0.190D -1 0.000D 0

I = 11, AREA(I) = 3546.3136

RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0

RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 3200.0 B= 133.1

FXA:

107.59 447.28 594.79 687.55 756.41 811.21

856.65 894.88 928.03 956.71 981.86

ITERATION SUCCESSFUL ON KI

Q= 15.0 RMU= 0.0H= 0.0 B= 133.1A = 3200.0

B= 133.1 FXA(1) = 7417307.9

SHAPE OF FRAC H/HZ FROM XA=1 TO 0

1.0100 1.1998 1.4047 1.6087 1.8229 2.0524

2.3017 2.5715 2.8692 3.1938 3.5501

PHI= 0.2317D 0 DELTAY= 0.1000D 0 H= 0.8116D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.113D 1 0.109D 1 0.973D 0 0.655D 0 0.425D 0 0.286D 0

0.186D 0 0.112D 0 0.570D -1 0.190D -1 0.000D 0

I = 11, AREA(I) = 3546.3136

AREAS(X) AS X=A TO 0:

.0000D 00.6938D 30.1033D 40.1320D 40.1593D 40.1869D 40.2170D 4

.2469D 40.2811D 40.3181D 40.3546D 4

VOLUME(X) AS X=A TO 0: (CGS)

.3384D 70.8420D 70.1147D 80.1420D 80.1688D 80.1970D 80.2262D 8

.2575D 80.2922D 80.3281D 80.0000D 0

PHI= 0.2223D 0 DELTAY= 0.1000D 0 H= 0.8641D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.117D 1 0.113D 1 0.985D 0 0.631D 0 0.418D 0 0.283D 0

0.184D 0 0.111D 0 0.567D -1 0.189D -1 0.000D 0

I = 12, AREA(I) = 7000.2140

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0⁵⁶
 RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

 A= 3520.0 B= 141.7

 PXA:
 107.59 438.29 586.51 678.99 747.59 802.24
 847.50 885.71 918.76 947.56 972.78 995.45
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 141.7A = 3520.0
 B= 141.7 PXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.1906 1.3899 1.5864 1.7915 2.0102
 2.2461 2.5006 2.7787 3.0820 3.4123 3.7798
 PHI= 0.2223D 0 DELTAY= 0.1000D 0 H= 0.8641D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.117D 1 0.113D 1 0.985D 0 0.631D 0 0.418D 0 0.283D 0
 0.184D 0 0.111D 0 0.567D -1 0.189D -1 0.000D 0
 I = 12, AREA(I) = 3800.2169
 AREAS(X) AS X=A TO 0 :
 .0000D 00.6758D 30.1011D 40.1289D 40.1552D 40.1820D 40.2103D 4
 .2395D 40.2706D 40.3056D 40.3400D 4
 VOLUME(X) AS X=A TO 0 : (CGS)
 .3296D 70.8225D 70.1122D 80.1386D 80.1645D 80.1913D 80.2194D 8
 .2488D 80.2810D 80.3148D 80.3511D 8
 TAU(I) AS I=1 TO N+1
 0.0000D 0 0.2663D 3 0.8159D 3 0.1571D 4 0.2500D 4 0.3585D 4
 0.4812D 4 0.6173D 4 0.7659D 4 0.9264D 4 0.1098D 5 0.1281D 5
 T(1)= 0.1098D 5, T(2)= 0.1281D 5
 T1CHECK:(SEC,MIN) 0.1167D 5 0.1945D 3 T2CHECK:(SEC,MIN) 0.1372D 5 0.2287D 3
 THESE TIMES ARE TOTAL PUMPING TIMES.
 VTOT1 = 0.1845D 9 VTOT2 = 0.2137D 9
 TOTLEAK1 = 0.2777D 9, TOTLEAK2 = 0.3298D 9
 V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.4621D 9 0.2917D 4 0.1221D 6
 V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.5434D 9 0.3431D 4 0.1436D 6
 % DIFF. BETWEEN T1CHECK AND T(1) = 5.892809
 % DIFF. BETWEEN T2CHECK AND T(2) = 6.655798
 DT(T2CHECK-T1CHECK)= 0.2054D 4 SEC., T(2)-T(1)= 0.1828D 4 SEC.
 LEAKRATE(11), QX(11): 0.2842D 5-0.1450D 4
 OUTPUT QX(I) AS X=0 TO A:
 0.1500D 2 0.1394D 2 0.1287D 2 0.1176D 2 0.1064D 2 0.9479D 1
 0.8254D 1 0.6936D 1 0.5439D 1 0.3544D 1 -0.5491D 0
 ERROR.....ERROR.....ERROR.....
 ERROR.....ERROR.....ERROR.....
 FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO : 0.00
 INPUT QX, AS X=0 TO A:
 0.1500D 2 0.1386D 2 0.1269D 2 0.1152D 2 0.1033D 2 0.9105D 1
 0.7840D 1 0.6497D 1 0.5019D 1 0.3263D 1 0.6661D-14
 A= 3200. OLD TA= 10982.2 NEW TA= 11669.9 ADJUSTED TA= 11498.0

 NOW CALC. ADJUSTED L(T) FOR 2 LINE-SEGMENTS.

 FROM 500. TO 1200. TSLOPE= 0.638 TCEPT= 0.929
 FROM 1200. TO 3200. TSLOPE= 0.615 TCEPT= 1.007

 NOW DO FULL LEAKOFF CALC. BASED ON 2 LINE-SEGMENTS OF L(T).

FOR A=A1, THEN A2, THEN A4. THIS IS ITERATED TIL IRUN=IRUNZ.
E.G. IRUNZ=8 GIVES 2(NOTE IT IS TWO!!!) OF THESE ITERATIONS.

57

PHI= 0.3962D 0 DELTAY= 0.1000D 0 H= 0.3979D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.671D 0 0.664D 0 0.641D 0 0.602D 0 0.541D 0 0.448D 0
0.266D 0 0.144D 0 0.701D -1 0.227D -1 0.000D 0
I = 11 , AREA(I) = 1486.0174
RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0

RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 500.0 B= 65.3

FXA:
107.59 335.58 434.00 498.97 549.21 590.57
625.96 657.05 684.73 709.68 732.49
ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 65.3A = 500.0
B= 65.3 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.1049 1.1864 1.2592 1.3289 1.3971
1.4648 1.5329 1.6013 1.6704 1.7407

PHI= 0.3962D 0 DELTAY= 0.1000D 0 H= 0.3979D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.671D 0 0.664D 0 0.641D 0 0.602D 0 0.541D 0 0.448D 0
0.266D 0 0.144D 0 0.701D -1 0.227D -1 0.000D 0
I = 11 , AREA(I) = 1486.0174
AREAS(X) AS X=A TO 0 :

.0000D 00.4918D 30.6674D 30.7983D 30.9143D 30.1022D 40.1116D 4
.1214D 40.1309D 40.1397D 40.1486D 4
VOLUME(X) AS X=A TO 0 : (CGS)
.3747D 60.8833D 60.1117D 70.1305D 70.1475D 70.1629D 70.1775D 7
.1923D 70.2063D 70.2197D 70.0000D 0

PHI= 0.3891D 0 DELTAY= 0.1000D 0 H= 0.4086D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.689D 0 0.682D 0 0.658D 0 0.616D 0 0.550D 0 0.448D 0

0.255D 0 0.141D 0 0.689D -1 0.224D -1 0.000D 0
I = 12 , AREA(I) = 1546.7364
RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0
RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 550.0 B= 67.0

FXA:
107.59 330.98 429.35 493.94 543.79 584.82
619.92 650.76 678.22 702.98 725.58 746.39
ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 67.0A = 550.0
B= 67.0 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.1018 1.1818 1.2529 1.3207 1.3869
1.4526 1.5184 1.5844 1.6511 1.7186 1.7873

PHI= 0.3891D 0 DELTAY= 0.1000D 0 H= 0.4086D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS

```

WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.689D 0 0.682D 0 0.658D 0 0.616D 0 0.550D 0 0.448D 0
0.255D 0 0.141D 0 0.689D -1 0.224D -1 0.000D 0
I = 12 , AREA(I) = 1546.7364
AREAS(X) AS X=A TO 0 :
.0000D 00.4843D 30.6582D 30.7881D 30.9009D 30.1006D 40.1099D 4
.1193D 40.1286D 40.1375D 40.1457D 4
VOLUME(X) AS X=A TO 0 : (CGS)
.3690D 60.8706D 60.1102D 70.1287D 70.1453D 70.1604D 70.1746D 7
.1889D 70.2028D 70.2158D 70.2289D 7
TAU(I) AS I=1 TO N+1
0.0000D 0 0.1608D 2 0.4762D 2 0.8987D 2 0.1410D 3 0.2001D 3
0.2662D 3 0.3389D 3 0.4177D 3 0.5024D 3 0.5925D 3 0.6879D 3
T(1)= 0.5925D 3, T(2)= 0.6879D 3
T1CHECK:(SEC,MIN) 0.6263D 3 0.1044D 2 T2CHECK:(SEC,MIN) 0.7253D 3 0.1209D 2
THESE TIMES ARE TOTAL PUMPING TIMES.
VTOT1 = 0.1474D 8 VTOT2 = 0.1680D 8
TOTLEAK1 = 0.1006D 8, TOTLEAK2 = 0.1192D 8
V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.2480D 8 0.1566D 3 0.6553D 4
V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.2872D 8 0.1813D 3 0.7588D 4
% DIFF. BETWEEN T1CHECK AND T(1) = 5.396328
% DIFF. BETWEEN T2CHECK AND T(2) = 5.151597
DT(T2CHECK-T1CHECK)= 0.9897D 2 SEC., T(2)-T(1)= 0.9541D 2 SEC.
LEAKRATE(11), QX(11) : 0.1938D 5 0.3179D 4
OUTPUT QX(I) AS X=0 TO A:
0.1500D 2 0.1422D 2 0.1341D 2 0.1255D 2 0.1164D 2 0.1068D 2
0.9641D 1 0.8459D 1 0.7088D 1 0.5310D 1 0.1204D 1
INPUT QX, AS X=0 TO A:
0.1500D 2 0.1417D 2 0.1330D 2 0.1239D 2 0.1143D 2 0.1041D 2
0.9312D 1 0.8098D 1 0.6701D 1 0.4925D 1 0.1008D 1
A= 500. OLD TA= 592.5 NEW TA= 626.3 ADJUSTED TA= 617.9
PHI= 0.3249D 0 DELTAY= 0.1000D 0 H= 0.5229D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.855D 0 0.841D 0 0.798D 0 0.718D 0 0.579D 0 0.338D 0
0.210D 0 0.123D 0 0.619D -1 0.205D -1 0.000D 0
I = 11 , AREA(I) = 2152.6773
RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0
RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????
*****
A= 1200.0 B= 85.8
*****
PXA:
107.59 393.85 513.77 591.09 649.60 697.15
737.23 771.98 802.47 829.77 854.32
ITERATION SUCCESSFUL ON KI
Q= 15.0 RMU= 0.0H= 0.0 B= 85.8A = 1200.0
B= 85.8 PXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.1494 1.2784 1.3980 1.5157 1.6347
1.7562 1.8818 2.0113 2.1465 2.2873
PHI= 0.3249D 0 DELTAY= 0.1000D 0 H= 0.5229D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.855D 0 0.841D 0 0.798D 0 0.718D 0 0.579D 0 0.338D 0
0.210D 0 0.123D 0 0.619D -1 0.205D -1 0.000D 0
I = 11 , AREA(I) = 2152.6773
AREAS(X) AS X=A TO 0 :
.0000D 00.5909D 30.8304D 30.1023D 40.1189D 40.1354D 40.1506D 4
.1669D 40.1822D 40.1982D 40.2153D 4
VOLUME(X) AS X=A TO 0 : (CGS)

```

.1081D 70.2599D 70.3389D 70.4046D 70.4651D 70.5231D 70.5806D 7 50

.6383D 70.6955D 70.7561D 70.0000D 0

PHI= 0.3157D 0 DELTAY= 0.1000D 0 H= 0.5439D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.880D 0 0.865D 0 0.818D 0 0.730D 0 0.568D 0 0.329D 0

0.206D 0 0.121D 0 0.612D -1 0.203D -1 0.000D 0
I = 12 , AREA(I) = 2262.0591

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 1320.0 B= 89.2

FXA:
107.59 387.60 507.78 584.80 642.95 690.23
730.04 764.60 794.94 822.01 846.48 868.62

ITERATION SUCCESSFUL ON KI

Q= 15.0 RMU= 0.0H= 0.0 B= 89.2A = 1320.0

B= 89.2 FXA(1) = 7417307.9
SHAPE OF FRAC H/HZ FROM XA=1 TO 0
1.0100 1.1441 1.2705 1.3869 1.5009 1.6159
1.7328 1.8533 1.9773 2.1060 2.2401 2.3792

PHI= 0.3157D 0 DELTAY= 0.1000D 0 H= 0.5439D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS
0.880D 0 0.865D 0 0.818D 0 0.730D 0 0.568D 0 0.329D 0

0.206D 0 0.121D 0 0.612D -1 0.203D -1 0.000D 0
I = 12 , AREA(I) = 2262.0591
AREAS(X) AS X=A TO 0 :

.0000D 00.5796D 30.8171D 30.1006D 40.1168D 40.1329D 40.1476D 4

.1632D 40.1786D 40.1933D 40.2096D 4
VOLUME(X) AS X=A TO 0 : (CGS)

.1060D 70.2554D 70.3334D 70.3976D 70.4567D 70.5130D 70.5684D 7

.6251D 70.6800D 70.7367D 70.7969D 7

TAU(I) AS I=1 TO N+1
0.0000D 0 0.6336D 2 0.1877D 3 0.3542D 3 0.5558D 3 0.7884D 3
0.1049D 4 0.1335D 4 0.1646D 4 0.1980D 4 0.2335D 4 0.2711D 4

T(1)= 0.2335D 4, T(2)= 0.2711D 4
T1CHECK:(SEC,MIN) 0.2415D 4 0.4025D 2 T2CHECK:(SEC,MIN) 0.2816D 4 0.4693D 2
THESE TIMES ARE TOTAL PUMPING TIMES.
VTOT1 = 0.4770D 8 VTOT2 = 0.5469D 8
TOTLEAK1 = 0.4793D 8, TOTLEAK2 = 0.5681D 8
V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.9564D 8 0.6038D 3 0.2527D 5
V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.1115D 9 0.7039D 3 0.2946D 5

% DIFF. BETWEEN T1CHECK AND T(1) = 3.313088
% DIFF. BETWEEN T2CHECK AND T(2) = 3.720190

DT(T2CHECK-T1CHECK)= 0.4007D 3 SEC., T(2)-T(1)= 0.3760D 3 SEC.

LEAKRATE(11), QX(11) : 0.2343D 5 0.1374D 4

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1409D 2 0.1317D 2 0.1223D 2 0.1123D 2 0.1019D 2

0.9075D 1 0.7840D 1 0.6415D 1 0.4583D 1 0.5204D 0

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1405D 2 0.1310D 2 0.1212D 2 0.1109D 2 0.1002D 2

0.8873D 1 0.7621D 1 0.6196D 1 0.4397D 1 0.5547D 0

A= 1200. OLD TA= 2335.0 NEW TA= 2415.0 ADJUSTED TA= 2395.0

PHI= 0.2295D 0 DELTAY= 0.1000D 0 H= 0.8230D 4
WIDTH, BROKEN DOWN IN 10 INCREMENTS

WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.114D 1 0.110D 1 0.976D 0 0.649D 0 0.423D 0 0.285D 0
 60
 0.186D 0 0.112D 0 0.569D -1 0.190D -1 0.000D 0
 I = 11 , AREA(I) = 3601.1444
 RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0
 RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

 A= 3200.0 B= 135.0

PXA:
 107.59 453.51 602.48 695.48 763.90 817.97
 862.63 900.07 932.48 960.44 984.98
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 135.0A = 3200.0

B= 135.0 PXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.2064 1.4188 1.6301 1.8506 2.0855
 2.3397 2.6137 2.9149 3.2418 3.6003

PHI= 0.2295D 0 DELTAY= 0.1000D 0 H= 0.8230D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.114D 1 0.110D 1 0.976D 0 0.649D 0 0.423D 0 0.285D 0

0.186D 0 0.112D 0 0.569D -1 0.190D -1 0.000D 0
 I = 11 , AREA(I) = 3601.1444
 AREAS(X) AS X=A TO 0 :
 .0000D 00.7064D 30.1053D 40.1348D 40.1629D 40.1908D 40.2216D 4

.2516D 40.2864D 40.3232D 40.3601D 4
 VOLUME(X) AS X=A TO 0 : (CGS)
 .3445D 70.8581D 70.1171D 80.1452D 80.1725D 80.2011D 80.2308D 8

.2624D 80.2973D 80.3333D 80.0000D 0
 PHI= 0.2203D 0 DELTAY= 0.1000D 0 H= 0.8771D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.118D 1 0.113D 1 0.988D 0 0.627D 0 0.417D 0 0.282D 0

0.184D 0 0.111D 0 0.566D -1 0.189D -1 0.000D 0
 I = 12 , AREA(I) = 3864.0766

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

 A= 3520.0 B= 143.9

PXA:
 107.59 444.42 594.15 687.03 755.32 809.33
 853.95 891.40 923.81 951.89 976.48 998.57
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 143.9A = 3520.0

B= 143.9 PXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.1969 1.4035 1.6074 1.8189 2.0434
 2.2849 2.5441 2.8272 3.1339 3.4670 3.8368

PHI= 0.2203D 0 DELTAY= 0.1000D 0 H= 0.8771D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.118D 1 0.113D 1 0.988D 0 0.627D 0 0.417D 0 0.282D 0

0.184D 0 0.111D 0 0.566D -1 0.189D -1 0.000D 0
 I = 12 , AREA(I) = 3864.0766
 AREAS(X) AS X=A TO 0 :

.0000D 00.6881D 30.1031D 40.1318D 40.1588D 40.1858D 40.2150D 4

61

.2439D 40.2762D 40.3114D 40.3457D 4
VOLUME(X) AS X=A TO 0 : (CGS)

.3356D 70.8384D 70.1145D 80.1417D 80.1680D 80.1955D 80.2238D 8

.2537D 80.2866D 80.3205D 80.3570D 8

TAU(I) AS I=1 TO N+1

0.0000D 0 0.2945D 3 0.8723D 3 0.1646D 4 0.2593D 4 0.3727D 4
0.5012D 4 0.6440D 4 0.8000D 4 0.9688D 4 0.1150D 5 0.1342D 5

T(1)= 0.1150D 5, T(2)= 0.1342D 5

T1CHECK:(SEC,MIN) 0.1193D 5 0.1988D 3 T2CHECK:(SEC,MIN) 0.1403D 5 0.2339D 3

THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.1880D 9 VTOT2 = 0.2179D 9

TOTLEAK1 = 0.2843D 9, TOTLEAK2 = 0.3378D 9

V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.4723D 9 0.2982D 4 0.1248D 6

V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.5557D 9 0.3508D 4 0.1468D 6

% DIFF. BETWEEN T1CHECK AND T(1) = 3.589355

% DIFF. BETWEEN T2CHECK AND T(2) = 4.336098

DT(T2CHECK-T1CHECK)= 0.2107D 4 SEC., T(2)-T(1)= 0.1926D 4 SEC.

LEAKRATE(11), QX(11) : 0.2772D 5-0.7175D 3

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1395D 2 0.1289D 2 0.1180D 2 0.1070D 2 0.9563D 1

0.8359D 1 0.7067D 1 0.5594D 1 0.3733D 1 -0.2718D 0

ERROR.....ERROR.....ERROR.....

ERROR.....ERROR.....ERROR.....

FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO : 0.00

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1392D 2 0.1282D 2 0.1170D 2 0.1056D 2 0.9386D 1

0.8150D 1 0.6826D 1 0.5334D 1 0.3474D 1 0.6661D-14

A= 3200. OLD TA= 11498.0 NEW TA= 11926.0 ADJUSTED TA= 11819.0

ADJUSTED L(T) =

FROM 500. TO 1200. TSLOPE= 0.646 TCEPT= 0.896
FROM 1200. TO 3200. TSLOPE= 0.614 TCEPT= 1.003

PHI= 0.3949D 0 DELTAY= 0.1000D 0 H= 0.3998D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.674D 0 0.667D 0 0.644D 0 0.604D 0 0.543D 0 0.449D 0

0.264D 0 0.143D 0 0.699D -1 0.227D -1 0.000D 0

I = 11, AREA(I) = 1496.8444

RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0

RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 500.0 B= 65.6

PXA:

107.59 341.13 439.56 504.25 554.13 595.09

630.07 660.74 688.01 712.59 735.04

ITERATION SUCCESSFUL ON KI

Q= 15.0 RMU= 0.0H= 0.0 B= 65.6A = 500.0

B= 65.6 PXA(1) = 7417307.9

SHAPE OF FRAC H/HZ FROM XA=1 TO 0

1.0100 1.1087 1.1919 1.2659 1.3364 1.4052

1.4733 1.5415 1.6100 1.6790 1.7490

PHI= 0.3949D 0 DELTAY= 0.1000D 0 H= 0.3998D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.674D 0 0.667D 0 0.644D 0 0.604D 0 0.543D 0 0.449D 0

0.264D 0 0.143D 0 0.699D -1 0.227D -1 0.000D 0

I = 11, AREA(I) = 1496.8444

AREAS(Y) AS Y=A TO 0

0.0000D 00.5007D 30.6784D 30.8095D 30.9266D 30.1034D 40.1128D 4

62

.1226D 40.1321D 40.1407D 40.1497D 4
VOLUME(X) AS X=A TO 0 : (CGS)

.3815D 60.8984D 60.1134D 70.1323D 70.1494D 70.1647D 70.1794D 7

.1941D 70.2079D 70.2213D 70.0000D 0

PHI= 0.3878D 0 DELTAY= 0.1000D 0 H= 0.4106D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.693D 0 0.685D 0 0.661D 0 0.618D 0 0.552D 0 0.448D 0

0.253D 0 0.140D 0 0.687D -1 0.224D -1 0.000D 0

I = 12 , AREA(I) = 1558.4006

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 550.0 B= 67.4

PXA:

107.59 336.64 435.05 499.41 548.95 589.61

624.32 654.76 681.83 706.23 728.49 748.96

ITERATION SUCCESSFUL ON KI

R= 15.0 RMU= 0.0H= 0.0 B= 67.4A = 550.0

B= 67.4 PXA(1) = 7417307.9

SHAPE OF FRAC H/HZ FROM XA=1 TO 0

1.0100 1.1056 1.1874 1.2597 1.3285 1.3954

1.4615 1.5275 1.5937 1.6604 1.7278 1.7963

PHI= 0.3878D 0 DELTAY= 0.1000D 0 H= 0.4106D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.693D 0 0.685D 0 0.661D 0 0.618D 0 0.552D 0 0.448D 0

0.253D 0 0.140D 0 0.687D -1 0.224D -1 0.000D 0

I = 12 , AREA(I) = 1558.4006

AREAS(X) AS X=A TO 0 :

.0000D 00.4935D 30.6694D 30.7992D 30.9136D 30.1019D 40.1111D 4

.1206D 40.1299D 40.1386D 40.1469D 4

VOLUME(X) AS X=A TO 0 : (CGS)

.3760D 60.8861D 60.1119D 70.1305D 70.1473D 70.1623D 70.1766D 7

.1909D 70.2046D 70.2176D 70.2307D 7

TAU(I) AS I=1 TO N+1

0.0000D 0 0.1751D 2 0.5119D 2 0.9587D 2 0.1496D 3 0.2114D 3
0.2803D 3 0.3558D 3 0.4374D 3 0.5249D 3 0.6179D 3 0.7161D 3

T(1)= 0.6179D 3, T(2)= 0.7161D 3

T1CHECK:(SEC,MIN) 0.6350D 3 0.1058D 2 T2CHECK:(SEC,MIN) 0.7353D 3 0.1225D 2

THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.1491D 8 VTOT2 = 0.1699D 8

TOTLEAK1 = 0.1024D 8, TOTLEAK2 = 0.1213D 8

V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.2515D 8 0.1588D 3 0.6644D 4

V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.2912D 8 0.1838D 3 0.7693D 4

% DIFF. BETWEEN T1CHECK AND T(1) = 2.704449

% DIFF. BETWEEN T2CHECK AND T(2) = 2.609727

DT(T2CHECK-T1CHECK)= 0.1002D 3 SEC., T(2)-T(1)= 0.9820D 2 SEC.

LEAKRATE(11), QX(11) : 0.1905D 5 0.3523D 4

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1422D 2 0.1343D 2 0.1258D 2 0.1168D 2 0.1073D 2

0.9703D 1 0.8533D 1 0.7176D 1 0.5416D 1 0.1334D 1

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1420D 2 0.1338D 2 0.1251D 2 0.1159D 2 0.1062D 2

0.9559D 1 0.8369D 1 0.6992D 1 0.5214D 1 0.1155D 1
 A= 500. OLD TA= 617.9 NEW TA= 635.0 ADJUSTED TA= 630.8 63
 PHI= 0.3239D 0 DELTAY= 0.1000D 0 H= 0.5250D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.858D 0 0.844D 0 0.800D 0 0.720D 0 0.578D 0 0.337D 0
 0.209D 0 0.123D 0 0.618D -1 0.204D -1 0.000D 0
 I = 11 , AREA(I) = 2164.1627
 RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0
 RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

 A= 1200.0 B= 86.1

PXA:
 107.59 395.89 516.76 594.22 652.64 699.98
 739.82 774.30 804.52 831.57 855.87
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 86.1A = 1200.0

B= 86.1 PXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.1511 1.2824 1.4036 1.5227 1.6426
 1.7649 1.8909 2.0207 2.1561 2.2968

PHI= 0.3239D 0 DELTAY= 0.1000D 0 H= 0.5250D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.858D 0 0.844D 0 0.800D 0 0.720D 0 0.578D 0 0.337D 0
 0.209D 0 0.123D 0 0.618D -1 0.204D -1 0.000D 0
 I = 11 , AREA(I) = 2164.1627
 AREAS(X) AS X=A TO 0 :

.0000D 00.5947D 30.8371D 30.1031D 40.1199D 40.1364D 40.1518D 4

.1680D 40.1832D 40.1993D 40.2164D 4
 VOLUME(X) AS X=A TO 0 : (CGS)

.1088D 70.2618D 70.3417D 70.4079D 70.4688D 70.5270D 70.5848D 7

.6423D 70.6996D 70.7603D 70.0000D 0

PHI= 0.3147D 0 DELTAY= 0.1000D 0 H= 0.5463D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.883D 0 0.868D 0 0.820D 0 0.731D 0 0.566D 0 0.328D 0

0.206D 0 0.121D 0 0.611D -1 0.202D -1 0.000D 0
 I = 12 , AREA(I) = 2274.1502

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

 A= 1320.0 B= 89.6

PXA:
 107.59 389.49 510.75 587.97 646.09 693.19
 732.79 767.10 797.18 824.03 848.25 870.15
 ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 89.6A = 1320.0

B= 89.6 PXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.1457 1.2744 1.3925 1.5079 1.6239
 1.7417 1.8628 1.9873 2.1163 2.2505 2.3896

PHI= 0.3147D 0 DELTAY= 0.1000D 0 H= 0.5463D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS

WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.883D 0 0.868D 0 0.820D 0 0.731D 0 0.566D 0 0.328D 0
 0.206D 0 0.121D 0 0.611D -1 0.202D -1 0.000D 0
 I = 12, AREA(I) = 2274.1502
 AREAS(X) AS X=A TO 0 :

64

.0000D 00.5830D 30.8237D 30.1015D 40.1178D 40.1340D 40.1487D 4

.1644D 40.1797D 40.1945D 40.2108D 4
 VOLUME(X) AS X=A TO 0 : (CGS)

.1066D 70.2572D 70.3362D 70.4010D 70.4605D 70.5170D 70.5727D 7

.6293D 70.6843D 70.7413D 70.8015D 7

TAU(I) AS I=1 TO N+1
 0.0000D 0 0.6788D 2 0.1984D 3 0.3716D 3 0.5800D 3 0.8193D 3
 0.1086D 4 0.1379D 4 0.1696D 4 0.2035D 4 0.2395D 4 0.2776D 4

T(1)= 0.2395D 4, T(2)= 0.2776D 4
 T1CHECK:(SEC,MIN) 0.2435D 4 0.4058D 2 T2CHECK:(SEC,MIN) 0.2838D 4 0.4730D 2
 THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.4803D 8 VTOT2 = 0.5508D 8
 TOTLEAK1 = 0.4840D 8, TOTLEAK2 = 0.5731D 8
 V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.9643D 8 0.6088D 3 0.2548D 5
 V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.1124D 9 0.7095D 3 0.2969D 5

% DIFF. BETWEEN T1CHECK AND T(1) = 1.644388
 % DIFF. BETWEEN T2CHECK AND T(2) = 2.199622

DT(T2CHECK-T1CHECK)= 0.4030D 3 SEC., T(2)-T(1)= 0.3807D 3 SEC.
 LEAKRATE(11), QX(11) : 0.2323D 5 0.1536D 4

OUTPUT QX(I) AS X=0 TO A:
 0.1500D 2 0.1410D 2 0.1318D 2 0.1224D 2 0.1126D 2 0.1022D 2

0.9106D 1 0.7876D 1 0.6455D 1 0.4628D 1 0.5818D 0
 INPUT QX, AS X=0 TO A:
 0.1500D 2 0.1408D 2 0.1316D 2 0.1220D 2 0.1120D 2 0.1015D 2

0.9025D 1 0.7785D 1 0.6360D 1 0.4536D 1 0.5290D 0

A= 1200. OLD TA= 2395.0 NEW TA= 2435.1 ADJUSTED TA= 2425.1

PHI= 0.2283D 0 DELTAY= 0.1000D 0 H= 0.8294D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS
 0.114D 1 0.110D 1 0.978D 0 0.646D 0 0.422D 0 0.285D 0

0.185D 0 0.112D 0 0.569D -1 0.190D -1 0.000D 0
 I = 11, AREA(I) = 3631.6102

RXA(I): 0.999D 0 0.899D 0 0.799D 0 0.699D 0 0.599D 0 0.499D 0

RXA(I): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????

 A= 3200.0 B= 136.0

FXA:
 107.59 458.97 608.27 700.79 768.56 821.97

866.03 902.94 934.89 962.44 986.67

ITERATION SUCCESSFUL ON KI
 Q= 15.0 RMU= 0.0H= 0.0 B= 136.0A = 3200.0

B= 136.0 PXA(1) = 7417307.9
 SHAPE OF FRAC H/HZ FROM XA=1 TO 0
 1.0100 1.2123 1.4298 1.6449 1.8684 2.1057
 2.3620 2.6377 2.9403 3.2683 3.6280

PHI= 0.2283D 0 DELTAY= 0.1000D 0 H= 0.8294D 4
 WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.114D 1 0.110D 1 0.978D 0 0.646D 0 0.422D 0 0.285D 0

0.185D 0 0.112D 0 0.569D -1 0.190D -1 0.000D 0
 I = 11, AREA(I) = 3631.6102
 AREAS(X) AS X=A TO 0 :

.0000D 00.7175D 30.1068D 40.1367D 40.1652D 40.1932D 40.2242D 4

.2543D 40.2893D 40.3260D 40.3632D 4
 VOLUME(X) AS X=A TO 0 : (CGS)

VOLUME(X) AS X=A TO 0 : (CGS)

.3499D 70.8705D 70.1187D 80.1472D 80.1748D 80.2036D 80.2334D 8

65

.2651D 80.3001D 80.3361D 80.0000D 0

PHI= 0.2192D 0 DELTAY= 0 1000D 0 H= 0.8841D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.118D 1 0.114D 1 0.989D 0 0.625D 0 0.416D 0 0.282D 0

0.184D 0 0.111D 0 0.566D -1 0.189D -1 0.000D 0

I = 12 , AREA(I) = 3898.5708

RXA(I): 0.999D 0 0.908D 0 0.817D 0 0.726D 0 0.635D 0 0.544D 0

RXA(I): 0.454D 0 0.363D 0 0.272D 0 0.181D 0 0.899D -1 -0.100D -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

A= 3520.0 B= 145.0

FXA:

107.59 449.80 600.10 692.53 760.21 813.59

857.64 894.56 926.53 954.19 978.43 1000.21

ITERATION SUCCESSFUL ON KI

Q= 15.0 RMU= 0.0H= 0.0 B= 145.0A = 3520.0

B= 145.0 PXA(1) = 7417307.9

SHAPE OF FRAC H/HZ FROM XA=1 TO 0

1.0100 1.2025 1.4144 1.6221 1.8368 2.0639

2.3079 2.5690 2.8541 3.1621 3.4967 3.8676

PHI= 0.2192D 0 DELTAY= 0.1000D 0 H= 0.8841D 4

WIDTH, BROKEN DOWN IN 10 INCREMENTS

0.118D 1 0.114D 1 0.989D 0 0.625D 0 0.416D 0 0.282D 0

0.184D 0 0.111D 0 0.566D -1 0.189D -1 0.000D 0

I = 12 , AREA(I) = 3898.5708

AREAS(X) AS X=A TO 0 :

.0000D 00.6989D 30.1047D 40.1337D 40.1611D 40.1882D 40.2177D 4

.2466D 40.2793D 40.3146D 40.3489D 4

VOLUME(X) AS X=A TO 0 : (CGS)

.3408D 70.8514D 70.1163D 80.1438D 80.1704D 80.1980D 80.2265D 8

.2565D 80.2896D 80.3235D 80.3603D 8

TAU(I) AS I=1 TO N+1

0.0000D 0 0.3097D 3 0.9053D 3 0.1696D 4 0.2660D 4 0.3825D 4

0.5147D 4 0.6614D 4 0.8220D 4 0.9957D 4 0.1182D 5 0.1380D 5

T(1)= 0.1182D 5, T(2)= 0.1380D 5

T1CHECK:(SEC,MIN) 0.1208D 5 0.2013D 3 T2CHECK:(SEC,MIN) 0.1422D 5 0.2369D 3

THESE TIMES ARE TOTAL PUMPING TIMES.

VTOT1 = 0.1901D 9 VTOT2 = 0.2204D 9

TOTLEAK1 = 0.2882D 9, TOTLEAK2 = 0.3426D 9

V/2 PUMPED(FRAC1) (CGS,BRL,GAL): 0.4783D 9 0.3020D 4 0.1264D 6

V/2 PUMPED(FRAC2) (CGS,BRL,GAL): 0.5630D 9 0.3554D 4 0.1487D 6

% DIFF. BETWEEN T1CHECK AND T(1) = 2.154216

% DIFF. BETWEEN T2CHECK AND T(2) = 2.915611

DT(T2CHECK-T1CHECK)= 0.2137D 4 SEC., T(2)-T(1)= 0.1983D 4 SEC.

LEAKRATE(11), QX(11) : 0.2733D 5-0.3160D 3

OUTPUT QX(I) AS X=0 TO A:

0.1500D 2 0.1394D 2 0.1291D 2 0.1183D 2 0.1074D 2 0.9613D 1

0.8423D 1 0.7144D 1 0.5684D 1 0.3840D 1 -0.1197D 0

ERROR.....ERROR.....ERROR.....

ERROR.....ERROR.....ERROR.....

FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO : 0.00

INPUT QX, AS X=0 TO A:

0.1500D 2 0.1394D 2 0.1288D 2 0.1178D 2 0.1067D 2 0.9519D 1

0.8307D 1 0.7007D 1 0.5528D 1 0.3448D 1 0.9982D -14

0.000000 A 0.000000 A 0.000000 A 0.000000 A 0.000000 A

A= 3200. OLD TA= 11819.0 NEW TA= 12079.2 ADJUSTED TA= 12014.2 66

ADJUSTED L(T) =

FROM	500. TO	1200. TSLOPE=	0.650 TCEPT=	0.879
FROM	1200. TO	3200. TSLOPE=	0.613 TCEPT=	1.005

)


```

WRITE(2,762) EMOD,RNU
762  FORMAT('  YOUNG'S MODULUS= ',E12.3,' POISS.RATIO= ',F10.2)
READ(1) S2MS1
WRITE(2,604) S2MS1
604  FORMAT(' S2-S1= ',F10.1)
READ(1) A1,A2,A4
WRITE(2,642) A1,A2,A4
642  FORMAT(' A1= ',F10.0,' A2= ',F10.0,' A4= ',F10.0)
READ(1) LEANC,DFT
WRITE(2,644) LEANC,DFT
644  FORMAT(' LEANC= ',F10.5,' DFT= ',F10.0)
READ(1)IRUNZ
WRITE(2,650)IRUNZ
650  FORMAT(' IRUNZ= ',I3)
READ(1)Q1,Q2,Q3
XA1=0.
XA2=.9
XA3=1.0

C  INITIALIZE SLOPE,CEPT VALUES
SLOP1 = (Q1 - Q2)/(XA2 - XA1)
CEP1 = Q1 + SLOP1*XA1
SLOP2 = (Q2 - Q3)/(XA3 - XA2)
CEP2 = Q2 + SLOP2*XA2
DO 561 J=1,2
SLOPE(J) = SLOP2
CEPT(J) = CEP2
561 DO 562 J=3-11
SLOPE(J) = SLOP1
CEPT(J) = CEP1
562 C  SLOPE(1),CEPT(1) CORRESPOND TO THE SEGMENT AT X=A
WRITE(2,460) Q1,Q2,Q3,(SLOPE(J),J=1,11), (CEPT(J),J=1,11)
460 C  FORMAT(' Q1=',F6.3,' Q2=',F6.3,' Q3=',F6.3/
C  ' SLOPE = ',11(F6.3,1X)/' CEPT = ',11(F6.3,1X))

C  ?????????????????????????????
C  THE ABOVE IS NOT PROGRAMMED FOR N=21, ONLY N=11

READ(1) ICOMP
WRITE(2,570)ICOMP
570  FORMAT(' ICOMP= ',I3,' IF ICOMP=1, THEN RECOMPUTE,LIST ALL KI'S
C  IN RANGE 0-1.')
READ(1)QPER
WRITE(2,572)QPER
572  FORMAT(' Q(X) WILL CONVERGE TO ',F10.0,' PERCENT ')

C  CONVERT UNITS
S2MS1 = 68940. * S2MS1
HZ= 30.48 * HZ
RF=HZ/2.
EMOD=EMOD*68940.
G=EMOD/(2.*(1.+RNU))
Q = QIN * 2640.
A=L*30.48
FLNP1 = FLN + 1.
FLNP = 2.* FLN + 1.
C1=(1.-RNU) / G
C4= FLNP * Q/ FLN
C5= -( 2.*FLNP1)*FLOWK/(C1**FLNP)
C2=S2MS1*2./PI
IFRAC = 1
ITERQ = 0
ICYCLE = 0

C  CONTINUE
IYZ=0
READ(1) NDIU,XAZ,YZF,N
613  WRITE(2,613) NDIU,XAZ,YZF,N
FORMAT(' NDIU= ',I5,' XAZ= ',F10.6,' YZF= ',F10.3,' N= ',I3)
CONTINUE
C  BEGIN NEW ITERATION

```

```

C
9      ICYCLE=ICYCLE+1
      READ(1) NRES,MFAC
      WRITE(2,592) NRES,MFAC
592    FORMAT(' NRES = ',I4,' MFAC = ',I4)
      READ(1) NODIV
      WRITE(2,61) NODIV
61     FORMAT(' NO. OF STEPS FOR NUM. INTEGRN.= ',I4,' (USUALLY 40)')
      READ(1) NUMWID
      WRITE(2,63) NUMWID
63     FORMAT(' NO. OF WIDTH CALONS. IN EACH VERT. SECTION= ',I4,
C      ' (USUALLY 10)')
      READ(1) HSTART
      WRITE(2,6232) HSTART
6232   FORMAT(' HSTART/HZ = ',F10.4)
      READ(1) SP
      WRITE(2,6233) SP
6233   FORMAT(' SPURT LOSS COEFF. IN CGS = ',F10.4)
      READ(1) IMU
      WRITE(2,6234) IMU
6234   FORMAT(' IMU= ',I2,' ; IMU=0 MEANS MU IS CONSTANT ')

C
C      -----
C      CONTROL PROGRAM TO AUTOMATIZE LEAKOFF
C      CONTINUE
C      CALCULATE L(T) FOR A=A1 AND C=0
101    WRITE(2,101)
      FORMAT(/' -----
C      '
C      ' CALC. ONE L(T) VALUE FOR A=A1 AND C=0.'/
C      '
C      ' -----')
      ISLOPE=1
      IRUN=1
      A=A1*30.48
      IFRAC=1
      ITQ=1
C      SET Q(X) TO Q(0) (1-X/A)
      DO 2501 J=1,N
2501   SLOPE(J)= 1.
      CEPT(J)= 1.
      GO TO 350
C      SAVE IFRAC=1 RESULTS FOR IRUN=3 CALONS
251    SAVT1 = TIME1
      S1AF1 = AF1
      S1VTOT1 = VTOT1
      DO 2511 J=1,N
2511   SVX1(1,J) = VX(1,J)
      SHF1(J)=HF(J)
      SPXA1(J)=PXA(J)
C
C      -----
C      CALCULATE L(T) FOR A=A4 AND C=0
102    WRITE(2,102)
      FORMAT(/' -----
C      '
C      ' CALC. ONE L(T) VALUE FOR A=A4 AND C=0.'/
C      '
C      ' -----')
      IRUN=2
      A=A4*30.48
      IFRAC=1
      ITQ=1
C      Q(X) SAME AS IRUN=1,SAME SLOPE,CEPT VALUES)
      GO TO 350
C      SAVE IFRAC=1 RESULTS FOR IRUN=3 CALONS
252    SAVT2 = TIME1
      S2AF1 = AF1
      S2VTOT1 = VTOT1
      DO 2522 J=1,N
2522   SVX2(1,J) = VX(1,J)
      SHF2(J)=HF(J)
      SPXA2(J)=PXA(J)

```

```

C      NOW CALC. 2 END POINTS OF 1ST APPROX OF L(T) WITH C = 0
C      (1) LOWER END POINT IS EXTRAPOLATION OF L(T) FROM L=A1,A4
C      TO T=2 MIN (BASED ON BERKELEY PLOT)
103    WRITE(2,103)
      FORMAT(/'-----'
C      ' NOW USE THESE RESULTS TO CALC. 1ST APPROX'N FOR L(T) [1 SEGMENT].'/
C      '-----')
      TL=120.
      ALLOG = DLOG10(A1) - (DLOG10(A4) - DLOG10(A1))*
C      (DLOG10(SAVT1) - DLOG10(120.))/(DLOG10(SAVT2) - DLOG10(SAVT1))
      (2) UPPER END POINT COMES FROM NORDGREN'S SOLUTION FOR L(T)
C      FOR T, C=INFINITY (WE TAKE T=10**6 SEC)
      TU = 1.D6
      AU = Q*DSQRT(TU)/(PI*LEAKC*HZ)
      AU = AU/30.48
C      NOW CALC PARAMS FOR LEAKOFF CALCN
      TSLOPE = (DLOG10(AU) - ALLOG)/(DLOG10(TU) - DLOG10(TL))
C      TSLOPE = SLOPE OF LOG L(FT) VS LOG T(SEC) PLOT
      TCEPT = ALLOG - TSLOPE * DLOG10(TL)
C      TCEPT = INTERCEPT OF PLOT AT LOG T=0.
      WRITE(2,2521) A1,SAVT1,A4,SAVT2,ALLOG,AU,TSLOPE,TCEPT
2521    FORMAT(' A1,SAVT1,A4,SAVT2,ALLOG,AU,TSLOPE,TCEPT = ',1X,8E11.4)
C      NOW USE THIS 1ST ESTIMATE OF TSLOPE & TCEPT PLUS H,P RESULTS
C      FOR A=A1 (WHEN C=0) TO ADJUST L(T) AT A=A1
C      WE ASSUME Q(X)=Q(0)*(1-X/A) FOR NOW. CAN'T ALLOW Q(X) TO
C      CONVERGE COZ H,P FOR IFRAC=1 ARE THOSE DERIVED FROM
C      Q(X)=Q(0)*(1-X/A)
      WRITE(2,104)
104    FORMAT(/'-----'
C      ' NOW USE 1ST APPROX'N FOR L(T) AND A=A1 RESULTS TO ADJUST L(T)
C      ' VALUE AT A=A1.'/'
C      '-----')
      ITRUN3=0
      TA4LOG = DLOG10(TU)-(DLOG10(AU)-DLOG10(A4))/TSLOPE
      TA(4)= 10.**TA4LOG
      TA1LOG = DLOG10(TU) - (DLOG10(AU) - DLOG10(A1))/TSLOPE
      TA(1)= 10.**TA1LOG
2525    A=A1*30.48
      IRUN = 3
      ITQ = 2
      IFRAC = 1
C      RESET Q(X) TO Q(0)*(1-X/A)
      DO 2529 J=1,N
        SLOPE(J)= 1.
2529      CEPT(J)= 1.
        DO 2523 J=1,N
          HP(J) = SHP1(J)
2523      FXA(J) = SPXA1(J)
          GO TO 729
        RTA(1)= (1.*TA(1) + 3.*T1CHECK)/4.
        DIFT1 = DABS(TA(1)- T1CHECK) / TA(1) *100.
        WRITE(2,2531)A1,TA(1),T1CHECK,DIFT1,RTA(1)
2531      FORMAT(/' A= ',E11.3,' INPUT T= ',E11.3,' OUTPUT T= ',E11.3,
C      ' DIFT1= ',E11.3,' ADJUSTED T ESTIMATE = ',E11.3)
C      SAVE ADJUSTED CEPT, SLOPE VALUES
      CALL QTEST(N,QPER,ITERQ)
      DO 2528 J=1,N
        SAVS(1,J)=SLOPE(J)
2528      SAVC(1,J)=CEPT(J)
C      REPEAT TO ADJUST L(T) AT A=A4
      WRITE(2,105)
105    FORMAT(/'-----'
C      ' NOW USE 1ST APPROX'N FOR L(T) AND A=A4 RESULTS TO ADJUST L(T)
C      ' VALUE AT A=A4.'/'

```

```

C *-----*
C *-----*)
A = A4*30.48
IRUN = 4
ITQ = 2
IFRAC = 1
C RESET Q(X) TO Q(0)*(1-X/A)
DO 2539 J=1,N
SLOPE(J)=1.
2539 CEPT(J)=1.
DO 2533 J=1,N
HP(J) = SHF2(J)
2533 PXA(J) = SPXA2(J)
GO TO 729
254 RTA(4) = (1.*TA(4) + 3.*T1CHECK)/4
DIFT4 = DABS(TA(4) - T1CHECK) / TA(4) *100.
WRITE(2,2541) A4,TA(4),T1CHECK,DIFT4,RTA(4)
2541 FORMAT(/' A= ',E11.3,' INPUT T= ',E11.3,' OUTPUT T= ',E11.3,
C * DIFT4= ',E11.3,' ADJUSTED T ESTIMATE = ',E11.3)
C SAVE ADJUSTED CEPT,SLOPE VALUES
CALL QTEST(N,QPER,ITERQ)
DO 2537 J=1,N
SAVS(3,J)=SLOPE(J)
2537 SAVC(3,J)=CEPT(J)
C *-----*
C NOW CALC. ADJUSTED TSLOPE,TCEPT
WRITE(2,106)
106 FORMAT(/' *-----*
C * NOW ADJUST L(T) [1 SEGMENT].*/
C *-----*
C *-----*)
TSLOPE = (DLOG10(A4)-DLOG10(A1))/(DLOG10(RTA(4))-DLOG10(RTA(1)))
TCEPT = DLOG10(A1) - TSLOPE * DLOG10(RTA(1))
WRITE(2,2542) TSLOPE,TCEPT
2542 FORMAT(/' ADJUSTED TSLOPE,TCEPT = ',2E11.3)
C NOW REPEAT TSLOPE,TCEPT ADJUSTMENT TIL TIME DIFFERENCES < DFT
TA(1) = RTA(1)
TA(4) = RTA(4)
ITRUN3 = ITRUN3 + 1
IF(ITRUN3.GT.4) GO TO 2526
IF(DIFT1.LT.DFT.AND.DIFT4.LT.DFT) GO TO 2526
GO TO 2525
2526 WRITE(2,2527) DIFT1,DIFT4,DFT,ITRUN3
2527 FORMAT(/' ITERATION IN IRUN=3 STOPPED BECAUSE TIME DIFFERENCES
C DIFT1= ',F10.1,' OR DIFT2= ',F10.1/
C * ARE BOTH LESS THAN ',F10.1,' PERCENT, OR ELSE NO. OF ITERATIONS
C ,I3,' EXCEEDS 4 ')
C *-----*
C CALC. T1CHECK FOR A1,A2,A3 AND ADJUSTED TA1,TA2,TA3(WITH LEAKOFF
C AND ONE Q(X) ITERATION.) ASSUMES L(T)=ONE LINE SEGMENT ONLY.
WRITE(2,107)
107 FORMAT(/' *-----*
C * USE THIS ADJUSTED L(T) [1 SEGMENT] TO DO FULL LEAKOFF CALCN */
C * FOR A1 THEN A2 THEN A4. FROM THE TIME DISCREPANCY WE COMPUTE */
C * AN ADJUSTED L(T) VALUE(=TA).*/
C *-----*
C *-----*)
TA2LOG = DLOG10(RTA(4)) - (DLOG10(A4) - DLOG10(A2))/TSLOPE
TA(2) = 10.** TA2LOG
C RESET INDEX 4 TO 3 FOR A = A4
TA(3) = TA(4)
C SET SAVED SLOPE, CEPT FOR A2 TO AV OF A1,A4
DO 2558 J=1,N
SAVS(2,J)=(SAVS(1,J) + SAVS(3,J))/2.
2558 SAVC(2,J)=(SAVC(1,J) + SAVC(3,J))/2.
DO 2556 K=1,3
IF(K.EQ.1) A=A1*30.48
IF(K.EQ.2) A=A2*30.48
IF(K.EQ.3) A=A4*30.48

```

```

      IRUN=5
      ITERQ=0
C     RESET Q(X) TO SAVED VALUES
      DO 2554 J=1,N
      SLOPE(J)=SAVS(K,J)
2554    CEPT(J)=SAVC(K,J)
2553    ITQ=2
      IFRAC=1
      GO TO 350
255    CALL QTEST(N,QPER,ITERQ)
C     WHEN THIS IS WORKING, MIGHT BUMP ITERATION BELOW UP TO 2
      IF(ITERQ.EQ.1) GO TO 2552
      GO TO 2553
C     SAVE ITERATED Q(X) FOR NEXT TIME
2552    DO 2557 J=1,N
      SAVS(K,J)=SLOPE(J)
2557    SAVC(K,J)=CEPT(J)
      RTA(K)=(1.*TA(K) + 3.*T1CHECK)/4.
      ATP = A/30.48
2556    WRITE(2,2555) ATP,TA(K),T1CHECK,RTA(K)
2555    FORMAT(/" A= ",F10.0," OLD TA= ",F10.1," NEW TA= ",F10.1,
      " ADJUSTED TA= ",F10.1)
C
C     NOW CALC ADJUSTED TSLOPE,TCEPT FOR REGIONS TA1->TA2, TA2->TA4
C     ASSUMES L(T) = 2 LINE SEGMENTS
      WRITE(2,108)
108    FORMAT(/" -----
C     "
C     " NOW CALC. ADJUSTED L(T) FOR 2 LINE-SEGMENTS."
C     "
C     " -----")
      TSL1=(DLOG10(A2)-DLOG10(A1))/(DLOG10(RTA(2))-DLOG10(RTA(1)))
      TSL2=(DLOG10(A4)-DLOG10(A2))/(DLOG10(RTA(3))-DLOG10(RTA(2)))
      TCE1=DLOG10(A1) - TSL1 * DLOG10(RTA(1))
      TCE2=DLOG10(A2) - TSL2 * DLOG10(RTA(2))
      WRITE(2,2575) A1,A2,TSL1,TCE1,A2,A4,TSL2,TCE2
2575    FORMAT(/" FROM ",F10.0," TO ",F10.0," TSLOPE= ",F10.3,
      " TCEPT= ",F10.3/" FROM ",F10.0," TO ",F10.0,
      " TSLOPE= ",F10.3," TCEPT= ",F10.3)
C
C     ISLOPE=2
C     CALC T1CHECK FOR A1,A2,A4 WITH LEAKOFF AND ONE Q(X) ITERATION
C     (START WITH SAVED Q(X))
      WRITE(2,109)
109    FORMAT(/" -----
C     "
C     " NOW DO FULL LEAKOFF CALCN. BASED ON 2 LINE-SEGMENTS OF L(T)."
C     " FOR A=A1, THEN A2, THEN A4. THIS IS ITERATED TIL IRUN=IRUNZ."
C     " E.G. IRUNZ=8 GIVES 2(NOTE IT IS TWO!!!) OF THESE ITERATIONS."
C     " -----")
C
      IRUN=6
2601    TA(1)=RTA(1)
      TA(2)=RTA(2)
      TA(3)=RTA(3)
      DO 2566 K=1,3
      IF(K.EQ.1) A=A1*30.48
      IF(K.EQ.2) A=A2*30.48
      IF(K.EQ.3) A=A4*30.48
      ITERQ=0
      DO 2564 J=1,N
C     RESET Q(X) TO SAVED Q(X)
      SLOPE(J)=SAVS(K,J)
2564    CEPT(J)=SAVC(K,J)
C     SELECT TSLOPE, TCEPT BELOW & ABOVE JOIN AT A2
      IF(K.EQ.3) GO TO 2568
      TSLOPE=TSL1
      TCEPT=TCE1
      GO TO 2563
2568    TSLOPE=TSL2
      TCEPT=TCE2
2563    ITQ=2

```



```

IFRAC=1
GO TO 350
256 CALL QTEST(N,QPER,ITERQ)
C NO ITERATION ON Q(X) WITH 1 BELOW
IF(ITERQ.EQ.1) GO TO 2562
GO TO 2563
C SAVE ITERATED Q(X) FOR NEXT TIME
2562 DO 2569 J=1,N
SAVS(K,J)=SLOPE(J)
2569 SAVC(K,J)=CEPT(J)
RTA(K)=(1.*TA(K) + 3.*T1CHECK)/4.
ATP= A/30.48
2566 WRITE(2,2565) ATP,TA(K),T1CHECK,RTA(K)
2565 FORMAT(/' A= ',F10.0,' OLD TA= ',F10.1,' NEW TA= ',F10.1,
' ADJUSTED TA= ',F10.1)
C CALC. ADJUSTED L(T) ..... TWO LINE SEGMENTS
TSL1=(DLOG10(A2)-DLOG10(A1))/(DLOG10(RTA(2))-DLOG10(RTA(1)))
TSL2=(DLOG10(A4)-DLOG10(A2))/(DLOG10(RTA(3))-DLOG10(RTA(2)))
TCE1=DLOG10(A1) - TSL1 * DLOG10(RTA(1))
TCE2=DLOG10(A2) - TSL2 * DLOG10(RTA(2))
110 WRITE(2,110)
FORMAT(/' -----
C ' ADJUSTED L(T) = ' /
C ' -----
C ' )
C WRITE(2,2575) A1,A2,TSL1,TCE1,A2,A4,TSL2,TCE2
IRUN=IRUN+1
C IF WANT ONE MORE ITERATION ON L(T), CHANGE 8 TO 9 BELOW
IF(IRUN.EQ.IRUNZ) GO TO 2600
GO TO 2601
2600 CONTINUE
STOP
C -----
350 CONTINUE
C -----
C BEGIN NEW FRAC HEIGHT CALCULATION.
C
C FLAG IS SET UP SO THAT WIDTHS ARE CALCULATED ONCE USING
C PHI = 3.*PI/16., THEN SWITCHED OVER TO THE DP/DX
C CORRECTION, IN EVERY RUN. THE CORRECTION IS APPLIED
C IN SR DFUNAM6 FOR C4.
93 FLAGPHI = 1
PHI = 3.*PI/16.
C
C IHST=0
HOLD= HSTART
HF(1) = HSTART
AFAC = 1. + 1./(N-1)
IF(IFRAC.EQ.1) NTEMP=N
NUMSEG=N-1
IF(IFRAC.EQ.2) NTEMP=N+1
IF(IFRAC.EQ.2) NUMSEG=N
XAINC=1./FLOAT(NUMSEG)
STEP=1./FLOAT(NDIV)
C CONVERT UNITS
YZ = YZF*68940.
C
C CALCULATE SOLN OF D.E. ARISING FROM W(P) AND P(W) EQNS, USING RUNGE-
TTA
C
KDIV= NDIV/NUMSEG
ISTART=1
H2= STEP/2.
STEP=STEP/FLOAT(MFAC)
H2=H2/FLOAT(MFAC)
KDIV=KDIV*MFAC
C THIS UPS THE NO OF STEPS IN 1ST SEGMENT. BY FACTOR MFAC
C FOR MORE ACCURATE INTEGRATION NEAR XA=1.

```

```

CONTINUE
C          WRITE(2,92)
C92        FORMAT(' IF YOU WANT TO CHANGE HSTART, HIT 1 ')
C          READ(1) IHST
C          IF(IHST.EQ.1) GO TO 93
C          ISEG = 2

C16        CONTINUE
C          CALL RK2( A,STEP,XAZ,YZ,NDIV,N,PXA,RXA,NRES,MFAC,IFRAC,H2,KDIV,ISEG)
C          IF(IWRITE.EQ.1) WRITE(2,615) C1,C2,C4,STEP
C615       FORMAT(' C1,C2,C4 = ',3E12.3,' STEP=',E12.3/)

C          CALC. PRESSURE FROM KI=KC
C          HH=HP(2)* HZ
C          PXA(2)=KCB*DSQRT(TOP/HH) + TOP*S2MS1*ACOS(1./HP(2))
C          IF(HP(2).LT.1.) PXA(2)=DSQRT(TOP/HH) * KCB
C          WRITE(2,19) HP(2),PXA(2),RXA(2)
C19        FORMAT(' HP(2)/HZ= ',E12.4,' PXA(2) = ',E12.4,' RXA(2) = ',E12.4)
C          TO CALCULATE NEW PHI
C          I = ISEG
C          H = HP(I)*HZ
C          FLP = HZ/H
C          L = H/2.

C          CALL WIDTH

C          IF (FLAGPHI.NE.1)GO TO 442
C          FLAGPHI = 0
C          GO TO 16

C          NOW RESET STEP, KDIV
C442       CONTINUE
C          STEP=STEP*FLOAT(MFAC)
C          H2=H2*FLOAT(MFAC)
C          KDIV=KDIV/MFAC

C          DO 336 ISEG=3,NTEMP
C          CALL RK2(A,STEP,XAZ,YZ,NDIV,N,PXA,RXA,NRES,MFAC,IFRAC,H2,KDIV,ISEG)
C          WRITE(2,17) ISEG
C17        FORMAT(' ISEG = ',I4)

C          CALC. PRESSURE FROM KI=KC
C          HH=HP(ISEG)* HZ
C          PXA(ISEG)=KCB*DSQRT(TOP/HH) + TOP*S2MS1*ACOS(1./HP(ISEG))
C          IF(HP(ISEG).LT.1.) PXA(ISEG)=DSQRT(TOP/HH) * KCB
C          WRITE(2,19) HP(ISEG),PXA(ISEG),RXA(ISEG)

C          TO CALCULATE NEW PHI FOR EACH SEGMENT
C          I = ISEG
C          H = HP(I)*HZ
C          FLP = HZ/H
C          L = H/2.

C          CALL WIDTH

C          IF(ISEG.EQ.NTEMP) GO TO 334
336       CONTINUE
334       CONTINUE
C          B=HP(NTEMP) * HZ/2.
C          IF(IWRITE.EQ.1)WRITE(2,620)(PXA(I), I=1,NTEMP)
C620       FORMAT(1X,' PXA(I):',6E12.3/)
C          IF(IWRITE.EQ.1)WRITE(2,630)(RXA(I), I=1,NTEMP)
C630       FORMAT(1X,' RXA(I):',6E12.3/)
C          IF(IWRITE.EQ.1) WRITE (2,634)
C634       FORMAT(' ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????'
C          CONTINUE
C          CONTINUE
C          CONTINUE

```

```

AFT=A/30.48
HH = HP(1)* HZ
FXA(1)= KCB*DSQRT(TOP/HH) + TOP*S2MS1*ACOS(1./HP(1))
IF(HP(1).LT.1.) FXA(1)=KCB*DSQRT(TOP/HH)
BFT=B/30.48
DO 1111 I=1,NTEMP
1111 FXAP(I)=FXA(I)/68940.
500 WRITE(2,500)AFT,BFT
C   FORMAT(/" *****"/" A= ",F10.1," B= ",
      F10.1/" *****"/)
501 WRITE(2,501)(FXAP(II),II=1,NTEMP)
      FORMAT(" FXA: ",(6F10.2/))

CONTINUE
QB=Q/2640.
SRMU=RMU/.01
WRITE(2,627) QB,SRMU,HFT,BFT,AFT
627  FORMAT(" ITERATION SUCCESSFUL ON KI
C     " Q= ",F10.1," RMU= ",F10.1," H= ",F10.1," B= ",F10.1,
C     " A= ",F10.1)
WRITE(2,607)BFT,FXA(1)
607  FORMAT(1X/," B= ",F10.1," FXA(1) = ",F10.1)
      WRITE(2,337) (HP(I),I=1,NTEMP)
337  FORMAT(" SHAPE OF FRAC H/HZ FROM XA=1 TO 0 ",(6F12.4/))

C   NOW COMPUTE ALL KI'S FOR CHECKING PURPOSES.
      IF(ICOMP.NE.1) GO TO 729
      DO 281 J=1,NTEMP
      MCON=1
      IF(HP(J).LE.1.) MCON=0
      KICHOICE(J,2)= FXA(J) * DSQRT(PI*HP(J)*HZ /2.)
      *(1.-2./PI * (S2MS1/PXA(J))*ACOS(MCON/HP(J))) *MCON)
C   WRITE(2,282) (KICHOICE(J,2),J=1,NTEMP)
C282  FORMAT(" K(I) = ",(6E12.4))

729  CONTINUE

C   CHECK WHICH CALCULATIONS ARE TO BE DONE - TIME, Q, NEITHER?
C   IF (ITQ-2) 770,750,9
C   CALCULATE TIME & Q
750  CONTINUE

C   DIVIDE FRAC INTO N SEGMENTS AND CALC VOLUME OF EA SEGMENT.
C   CALL THIS FRAC1 THEN ITERATE FOR FRAC2 WHICH IS TO BE ONE
C   SEGMENT LONGER.

      IF (IFRAC.EQ.2) GO TO 770
      CALL ULTQ
      ICYCLE = 0
      GO TO 93

C   NOW CALC THE VOLUME OF EA OF THE N+1 SEGMENTS FOR FRAC2. FROM
C   SEGMENT VOL'S FOR FRACS 1&2 CALC FLOW RATES DOWN THE FRAC.
C   770 CONTINUE
      CALL ULTQ
      IF (IRUN.EQ.1) GO TO 251
      IF (IRUN.EQ.2) GO TO 252
      IF (IRUN.EQ.3) GO TO 253
      IF (IRUN.EQ.4) GO TO 254
      IF (IRUN.EQ.5) GO TO 255
      IF (IRUN.EQ.6) GO TO 256
      IF (IRUN.LT.IRUNZ) GO TO 256
END

```

```

      COMPILER NOSTACK
      COMPILER DOUBLE PRECISION
C*****
C SUBROUTINE CLOSEWID, CREATED FROM MODIFICATION OF OLD DWIDTH
C THIS SUBROUTINE CALCULATES THE CLOSED FORM INTEGRAL FOR THE WIDTH AND
C FROM THE WIDTH CALCULATES THE AREA
C
C      CLOSEWID11 IS FOR NONNEWT.FLUID FLOW; PRINTS OUT WELLBORE WIDTHS
C*****
      SUBROUTINE WIDTH
      DIMENSION VX(2,22),T(2),AREAS(22),W(51),PXA(22),HP(22)
      REAL L
      INTEGER FFLAG,WFLAG
      COMMON /SUBVTQ/ IFRAC,VX,T,AFAC,ITQ,Q,A,PI,S2MS1,WFLAG,PXA,N
      C      ,EMOD,RNU,NODIV,A1,A2,A1
      COMMON /ABLOC/B,C2,HZ,C4,CONST,E,XAC,TOF,NFLOW,DELTA,HNEW,HOLD,
      C      XAOLD,XAINC,HP,PHI
      COMMON /SUBWIDTH/ W,AREAS,NUMWID,I,L,H
      COMMON /FNCTN/ FLP,FL,FFLAG
      COMMON /NONNEWT/FLN,FLOWK,C5,FLNP1,FLNP
      COMMON /EBLOC/NTEMP

C INITIALIZE VARIABLES BEFORE ENTERING W LOOP
      FFMS1= PXA(I)
      WINC = 1./NUMWID
      IENDNUMWID = NUMWID

C CALCULATE W FOR 'NUMWID' INTERVALS OF THE HALF-HEIGHT
      DO 200 K = 1,IENDNUMWID
      FL = WINC*(K-1)

      WP = (PI/2.*(FFMS1)-S2MS1*ACOS(FLP))*DSQRT(1.-FL**2)
C TEST H, HZ RELATIONSHIP
C      NOTE THAT IN THESE CALCNS.,H IS ALWAYS > HZ.
      IF (H.GT.HZ)GO TO 19
      AREA2=0.0
      GO TO 20

19      IF (FL-FLP) 10,20,30
10      CONTINUE

C      CASE I: BELOW THE EDGE OF THE PAYZONE
C      FL.LT.FLP
C
      GFL = DSQRT(FLP**2-FL**2)
      GO TO 40
20      CONTINUE

C      IF FL=FLP, THEN SET FL=FLP+.001
C      OTHERWISE WE WILL GET A ZERO IN THE DENOM.
C
      FL = FL + .001
30      CONTINUE

C      CASE II: ABOVE THE EDGE OF THE PAYZONE
C      FL.GE.FLP
C
      GFL = DSQRT(FL**2-FLP**2)

40      WA = FLP* DLOG((DSQRT(1.-FL**2)+DSQRT(1.-FLP**2))/GFL)
      WB = FL * DLOG((FLP*DSQRT(1.-FL**2)+FL*DSQRT(1.-FLP**2))/GFL)

      W(K) = (WP+S2MS1*(WA-WB))*B.*(1.-RNU**2)*L/(PI*EMOD)

200      CONTINUE

C      DEFINE TOP WIDTH TO BE 0.0
C
      KK = NUMWID + 1
      W(KK) = 0.0
C

```

```

C      CALCULATE THE TOTAL AREA.
C
      WTOT = 0.0
      DO 250 K=1, NUMWID
        KK = K + 1
        WTOT = WTOT + (W(K) + W(KK))/2.
250    CONTINUE
      AREAS(I) = 2. * WINC * WTOT * L

C      CALCULATE PHI
      FLM=(2.*FLN+1.)/FLN
      DELTAY = 1./NUMWID
      WNOT = W(1)
      NUMWPF1=NUMWID+1
      SUM=0.5
      DO 275 K=2, NUMWPF1
275    SUM = SUM + (W(K)/WNOT)**FLM
      PHI = DELTAY*SUM

C
C      TO PRINT WELLBORE WIDTHS ONLY
C
      IF(I.NE.NTEMP)GO TO 401
      WRITE(2,285) PHI,DELTAY,H
285    FORMAT(/' PHI=',E12.4,' DELTAY=',E12.4,' H=',E12.4)

C      CHANGE IENDNUMWID SO ALL NUMWID + 1 WIDTHS ARE PRINTED OUT
C
      IENDNUMWID = NUMWID + 1

      WRITE(2,300)NUMWID
300    FORMAT(1X,' WIDTH, BROKEN DOWN IN ',I4,' INCREMENTS')
      WRITE(2,350)(W(J),J=1,IENDNUMWID)
350    FORMAT(1X,(6E10.3,1X)/)
      WRITE(2,400)I,AREAS(I)
400    FORMAT(1X,' I = ',I2,' , AREA(I) = ',F10.4) -
401    CONTINUE
      RETURN
      END

      COMPILER NOSTACK
      COMPILER DOUBLE PRECISION

C
C
C
C      *****
C      THIS SUBROUTINE PERFORMS RUNGE-KUTTA SOLUTION OF
C      DIFFERENTIAL EQN ARISING FROM W(P) AND P(W) EQNS.
C      *****
      SUBROUTINE RK2(A,STEP,XAZ,YZ,NDIV,N,FXA,RXA,NRES,MFAC,IFRAC,H2,
C
C      KDIV,IDUM)
      COMMON/ABLOC/B,C2,HZ,C4,CONST,E,XAC,TOF,NFLOW,DELTA,HNEW,HOLD,
C      XAOLD,XAINC,HP,PHI
      COMMON/KVAL/KCB, HSTART, ISEG
      REAL KCB
      DIMENSION FXA(22),RXA(22),HP(22)
      IF(ISEG.GE.3) GO TO 20
      IFR=0
      RXA(1)=XAZ
      Y=HSTART* HZ
      XA=XAZ

```

```

DO 11 J=1,KDIV
  T1=-STEP*FUN(XA,Y)
  T2=-STEP*FUN(XA-H2,Y+T1/2.)
  T3=-STEP*FUN(XA-H2,Y+T2/2.)
  T4=-STEP*FUN(XA-STEP,Y+T3)
  ALL STEPS ARE -VE COZ WE ARE INTEGRATING BACKWARDS ALONG
  THE X AXIS (DY=-DX * F(X,Y) )
  Y=Y+ A* (T1+2.*T2 + 2.*T3 + T4)/6.
  THIS *A IS DONE HERE INSTEAD OF IN FUN, COZ ALL FUN( )
  HAVE TO BE MULTIPLIED BY A.
  IF(IPR.GT. 1) GO TO 31
  WRITE(2,55) STEP,XA,T1,T2,T3,T4,Y
  FORMAT(' STEP= ',E12.3,' XA= ',E12.3,' T1,T2,T3,T4= ',
    55 4E12.3,' Y= ',E12.3)
  IPR=IPR+1
  31 CONTINUE
  11 XA=XA-STEP
  HP(2) = Y/ HZ
  RXA(2) = XA
  RETURN
  20 CONTINUE
  ITEMP=ISEG-1
  XA=RXA(ITEMP)
  Y=HP(ITEMP)* HZ
  DO 1 J=1,KDIV
    T1=-STEP*FUN(XA,Y)
    T2=-STEP*FUN(XA-H2,Y+T1/2.)
    T3=-STEP*FUN(XA-H2,Y+T2/2.)
    T4=-STEP*FUN(XA-STEP,Y+T3)
    ALL STEPS ARE -VE COZ WE ARE INTEGRATING BACKWARDS ALONG
    THE X AXIS (DY=-DX * F(X,Y) )
    Y=Y+ A* (T1+2.*T2 + 2.*T3 + T4)/6.
    THIS *A IS DONE HERE INSTEAD OF IN FUN, COZ ALL FUN( )
    HAVE TO BE MULTIPLIED BY A.
    IF(IPR.GT. 1) GO TO 3
    WRITE(2,55) STEP,XA,T1,T2,T3,T4,Y
    IPR=IPR+1
    3 CONTINUE
    1 XA=XA-STEP
    HP(ISEG)= Y/ HZ
    2 RXA(ISEG)= XA
  RETURN
END

```

```

C C C C C
  COMPILER DOUBLE PRECISION
  COMPILER NOSTACK
  *****
  SUBROUTINE VLTQ - CALC VOLUME, LEAKOFF, TIME, AND Q
  -- DVLTO11B + MODIF. TOTLEAK AND CORRECT ISLOPE,TCEPT VALUES
  FOR ISLOPE=1,2 + RESET ANY -VE QX OUTPUT TO DU/DT.(071884)
  *****
  SUBROUTINE VLTQ
  DIMENSION VX(2,22),T(2),QX(22),AREAS(22),W(51),FXA(22),HP(22)
  DIMENSION TAU(21),SVX1(1,11),SVX2(1,11),TOTLEAK(2),AVTAU(2)
  INTEGER FFLAG,WFLAG,DD
  REAL INQX(22),L,LEAKC,NEWQX,LEAKRATE
  COMMON /SUBVTQ/ IFRAC,VX,T,AFAC,ITQ,Q,A,PI,S2MS1,WFLAG,FXA,N
  ,EMOD,RNU,MODIV,A1,A2,A4
  C COMMON /ABLOC/B, C2,HZ,C4,CONST,E,XAC,TOP,NFLOW,DELTA,HNEW,HOLD,
  C XAOLD,XAINC,HP,PHI

```

```

COMMON /SUBWIDTH/ W,AREAS,NUMWID,I,L,H
COMMON /FNCTN/ FLP,FL,FFLAG
COMMON /CBLOC/ CEPT(22),SLOPE(22)
COMMON /SUBLEAK/ VTOT1,TIME1,LEAKC,TSLOPE,TCEPT,T1CHECK,ISLOPE,
C AF1,IRUN,SUX1,SUX2,S1AF1,S1VTOT1,S2AF1,S2VTOT1
COMMON/QUAL/ QX,INQX
COMMON/TC/TSL1,TCE1,TSL2,TCE2
COMMON/DBLOC/SP

DD = 0
IF (IFRAC.EQ.2) NUMSEG = N
IF (IFRAC.EQ.2) GO TO 3
IF (IRUN.EQ.3.OR.IRUN.EQ.4) GO TO 360
C IFRAC = 1
C NUMSEG = N-1
C
C CALCULATE AREA(FLP)
C
C 3 XA = 1.0
XAINCR=1./NUMSEG
IF (IFRAC.EQ.1) AFAC = 1. + XAINCR
C
C LOOP OVER THE VERTICAL SECTIONS CALCULATING AREAS
C 'AREAS(I)' IS THE AREA OF THE I' TH VERTICAL SECTION IN CGS
C
AREAS(1) = 0.0
IEND = NUMSEG + 1
DO 100 I=2,IEND
H = HF(I) * HZ
FLP= HZ/H
L = H/2.
C
C CALL WIDTH SUBROUTINE TO CALCULATE WIDTH AND AREA FOR THIS SEGMENT
C
CALL WIDTH
C WRITE(2,2222) FLP,HZ,H,B,N,I,IFRAC
C2222 FORMAT(/' FLP,HZ,H,B,N,I,IFRAC: ',4E11.4,3I5)
100 CONTINUE
WRITE(2,1500)
1500 FORMAT(' AREAS(X) AS X=A TO 0 :')
WRITE(2,2000) (AREAS(I),I=1,N)
C
C NOW, CALCULATE VOLUME BY INTEGRATING THE AREA OVER X
C V = 2 * INTEGRAL(A(X)DX) FROM X=0 TO X=A
C
VTOT = 0.0
DO 300 I = 1, NUMSEG
II= I + 1
VX(IFRAC,I) = XAINCR*A * (AREAS(I)+AREAS(II))/2.
300 VTOT = VTOT + VX(IFRAC,I)
C
C SAVE IFRAC = 1 VALUES FOR LATER USE IN LEAKOFF CALCULATIONS
IF (IFRAC.NE.1) GOTO 350
AF1 = A/30.48
VTOT1 = VTOT
TIME1 = VTOT1/Q
GOTO 360
350 VTOT2=VTOT
360 CONTINUE
C RESET IFRAC=1 RESULTS WHEN IRUN=3 OR 4
IF (IRUN.EQ.3.AND.IFRAC.EQ.1) GO TO 531
GO TO 532
531 AF1 = S1AF1
VTOT1 = S1VTOT1
DO 533 J=1,N
533 VX(1,J) = SUX1(1,J)
532 CONTINUE
IF (IRUN.EQ.4.AND.IFRAC.EQ.1) GO TO 534
GO TO 535
534 AF1 = S2AF1
VTOT1 = S2VTOT1
DO 536 J=1,N
536 VX(1,J) = SUX2(1,J)
535 CONTINUE

```

```

C                                     Q IS THE FLOW RATE DOWN A 1-SIDED FRAC &
C                                     VTOT IS THE VOLUME OF A 1-SIDED FRAC
1900 WRITE(2,1900)
    FORMAT(' VOLUME(X) AS X=A TO 0 : (CGS)')
2000 WRITE(2,2000) (VX(IFRAC,I),I=1,N)
    FORMAT(7E10.4/)
C
C CHECK IF Q CALC'NS ARE TO BE PERFORMED (ITQ=2)
C IF (ITQ.NE.2) GO TO 999
C
C IF FRAC2 HAS NOT BEEN ITERATED FOR THEN ITERATE FOR FRAC2
C IF (IFRAC.EQ.2) GO TO 750
    A = AFAC * A
    IFRAC = IFRAC + 1
    GO TO 999
750 CONTINUE
C
C NOW CALC LEAKOFF RATE AND VOLUME FOR EACH SEGMENT
C
C COMPUTE NEEDED VALUES FROM INPUT VALUES
C IF (ISLOPE.EQ.2) GO TO 10
    XSL = TSLOPE
    XCE = TCEPT
    TSL1 = XSL
    TCE1 = XCE
    TSL2 = XSL
    TCE2 = XCE
    GO TO 30
10 IF (AF1.GT.A2) GO TO 20
    XSL = TSL1
    XCE = TCE1
    GO TO 30
20 XSL = TSL2 ; IF (AF1=A4) THERE ARE 2 SLOPES:
    XCE = TCE2 ; (TSL1,TCE1),(TSL2,TCE2)
30 TEXP = 1./XSL
    TDIV = 10.**XCE
C CALC TIMES FROM L/T GRAPH
    NM1 = N - 1
    AF2 = A/30.48
    T(1) = (AF1/TDIV)**TEXP
    T(2) = (AF2/TDIV)**TEXP
C AT JUNCTION POINT A2, WE ASSUME T(2) AS WELL AS T(1)
C ARE DETERMINED FROM SLOPE, CEPT BELOW JUNCTION
C (THE LAST TAU VALUE SHOULD BE AT THE JUNCTION).
C CALC TAU TIMES; TAU(1)=0, TAU(2)=TIME FRAC ARRIVES AT END OF 1ST SEG
    DD = IFIX(AF1 - A4)
    DO 75 I=1,N
        HL = AF1 * FLOAT(I-1)/(FLOAT(N)-1.)
        TAU(I) = (HL/TDIV)**TEXP
    IF (DD.EQ.0.AND.HL.LE.A2) TAU(I)=(HL/10.**TCE1)**(1./TSL1)
75 CONTINUE
    TAU(12) = T(2)
    WRITE(2,76) (TAU(I),I=1,12)
76 FORMAT('/' TAU(I) AS I=1 TO N+1'/(6(E11.4,2X)))
C CALC LEAKOFF VOLUME WITH SPURT LOSS
    C1 = AF1 * 30.48 / (N-1) ; C1=LENGTH OF EACH SEGMENT
    CC = C1*LEAKC*HZ
    A2C = A2*30.48
    A4C = A4*30.48
    DO 86 J=1,2
        AFX = AF1*(1.+(J-1)/10.)*30.48
        SPT = 2.*SP*HZ*AFX
        TC = 4.*HZ*LEAKC
        IF (DD.EQ.0) GO TO 25
        TOTLEAK(J)=TC*XSL*AFX*FFX11(0.,1.,NODIV,XSL,1)*SQRT(T(J))
        TOTLEAK(J)=TOTLEAK(J) + SPT
    GOTO 86

```



```

85      B1=(A2C/AFX)**TEXP
      FX1=FFX11(0.,B1,NODIV,TSL1,1)
      TERM1 = TSL1*A2C*SQRT(T(J))*(AFX/A2C)**(TSL1/TSL2)*FX1
      TERM2 = TSL2*AFX*SQRT(T(J))*FFX11(B1,1.,NODIV,TSL2,1)
      TOTLEAK(J) = (TERM1 + TERM2 + SF*AFX/(2.*LEAKC))*TC
86      CONTINUE

C CALC TOTAL VOLUME,T1CHECK,T2CHECK
      TOTVOL1 = VTOT1 + TOTLEAK(1)
      T1CHECK = TOTVOL1/Q
      TOTVOL2 = VTOT2 + TOTLEAK(2)
      T2CHECK = TOTVOL2/Q
      VBRL1 = TOTVOL1/(2640.*60.)
      VBRL2 = TOTVOL2/(2640.*60.)
      VGAL1 = TOTVOL1/3785.
      VGAL2 = TOTVOL2/3785.
      T1MIN = T1CHECK/60.
      T2MIN = T2CHECK/60.
      T1PER = 100. * (T1CHECK - T(1))/T1CHECK
      T2PER = 100. * (T2CHECK - T(2))/T2CHECK
      WRITE(2,2500) T(1),T(2),T1CHECK,T1MIN,T2CHECK,T2MIN,VTOT1,VTOT2,
2500    TOTLEAK(1),TOTLEAK(2),TOTVOL1,VBRL1,VBRL2,VGAL1,TOTVOL2,VBRL2,VGAL2
      FORMAT(/' T(1)= ',E11.4,' , T(2)= ',E11.4/' T1CHECK:(SEC,MIN)',
      2E11.4,' T2CHECK:(SEC,MIN)',2E11.4/' THESE TIMES ARE TOTAL'
      ' PUMPING TIMES.'/' VTOT1 = ',E11.4,3X,' VTOT2 = ',E11.4/
      ' TOTLEAK1 = ',E11.4,' , TOTLEAK2 = ',E11.4/
      ' V/2 PUMPED(FRAC1) (CGS,BRL,GAL): ',3E11.4/
      ' V/2 PUMPED(FRAC2) (CGS,BRL,GAL): ',3E11.4/
      WRITE(2,2550) T1PER,T2PER
2550    FORMAT(/' % DIFF. BETWEEN T1CHECK AND T(1) = ',F14.6/
      ' % DIFF. BETWEEN T2CHECK AND T(2) = ',F14.6)

C      NOW CALC FLOW RATE DOWN EACH SEGMENT:OUTPUT Q(X) AS X=0 TO A
C
      DT = T2CHECK-T1CHECK
      T2MT1 = T(2)-T(1)
      WRITE(2,2750) DT,T2MT1
2750    FORMAT(/' DT(T2CHECK-T1CHECK)=',E11.4' SEC., T(2)-T(1)=
      ' ',E11.4,' SEC.')
      QX(1) = Q
      IV1 = N-1
      IV2 = N
      DO 755 I=2,NM1
          R1 = DSQRT(T(1)-TAU(I-1))
          R2 = DSQRT(T(1)-TAU(I))
          QC = CC*(1./R1 + 1./R2)
          QX(I) = QX(I-1)-(VX(2,IV2)-VX(1,IV1))/DT - QC
          IV1 = IV1 - 1
          IV2 = IV2 - 1
755
C NOW CALC LAST FLOW RATE QX(A)
      CL = 2.*SF*HZ/T(1)
      IF(DD.EQ.0) GOTO 756
      LEAKRATE = 20.*CC*TSL1*FFX11(0.,1.,NODIV,TSL1,2)/SQRT(T(1))
      LEAKRATE = LEAKRATE + CL*TSL1*AF1*30.48
      GOTO 757
756    B2 = (A2/A4)**TEXP
      FX1 = FFX11(0.,B2,NODIV,TSL1,2)
      TERM1 = TSL1*A2C*FX1/SQRT(T(1))*(A4/A2)**(TSL1/TSL2)
      FX1 = FFX11(B2,1.,NODIV,TSL2,2)
      TERM2 = TSL2*A4C*FX1/SQRT(T(1))
      LEAKRATE = (TERM1+TERM2)*2.*HZ*LEAKC + CL*TSL2*A4C
757    QX(11) = QX(1) - LEAKRATE - (VTOT2-VX(2,1)-VTOT1)/DT
      WRITE(2,783) LEAKRATE,QX(11)
783    FORMAT(' LEAKRATE(11), QX(11) :',2E11.4)

C RESET ANY -VE QX OUTPUT TO VTIF(FRAC2)/DT
      DO 786 I=1,N
786    QX(I) = QX(I)/2640.
      WRITE(2,785) (QX(I),I=1,N)
785    FORMAT(' OUTPUT QX(I) AS X=0 TO A:',6(E11.4,2X)/)
      QXLAST = 0.0

```

```

C      B3 = T(1)/T(2)
CC     F7 = FFX11(B3,1.0,NODIV,XSL,1)
CC     E = 4.4*HZ*LEAKD*XSL*AF1*30.48*SQRT(T(2))*F7
CC     F = 0.2*SF*AF1*30.48*HZ
CC     UTIF2 = VX(2,1) + E + F
C      QXLAST = ((UTIF2)/DT)/2640.
      NEG = 0
      DO 784 I=1,N
        IF(QX(I).GT.0.) GOTO 784
        NEG = 1
        QX(I) = QXLAST
784     CONTINUE
      IF(NEG.EQ.1) GOTO 13
C      WRITE(2,14) QXLAST
C14     FORMAT(' ALTERNATE QX(11) CALCULATED FROM FRAC2 LAST SEGMENT'
C      C      ' (BPM) = ',E11.4/)
      GOTO 15
13     WRITE(2,11) QXLAST
11     FORMAT(/' ERROR.....ERROR.....ERROR.....'/
C      C      ' ERROR.....ERROR.....ERROR.....'/
C      C      ' FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO :',F5.2/)
C REFORM AND WRITE OUT INPUT QX
15     INQX(1) = Q/2640.
      QNOT = INQX(1)
      XA = 0.0
      XAINCR = 1./(N-1)
      DO 3100 I=2,N
        XA=XA+XAINCR
        ISE = 13 - I
        INQX(I) = QNOT*(-SLOPE(ISE)*XA + CEPT(ISE))
3100    CONTINUE
      WRITE(2,3200)(INQX(I),I=1,N)
3200    FORMAT(' INPUT QX, AS X=0 TO A: ',/(6(E11.4,2X)/))
      A = A/AFAC
C      RESTORES A TO FRAC1 VALUE - NEED IF ITERATE ON Q(X).
999    RETURN
      END

```

```

C      COMPILER DOUBLE PRECISION
CC     *****
CC     THIS FUNCTION SUBPROGRAM IS CALLED BY RK2 IN RUNGE-KUTTA
CC     SOLUTION OF DIFF.EQN. ARISING FROM W(P) AND P(W) EQNS.
CC     IN FLAMOCO
CC     *****
C      FUNCTION FUN(XA,Y)
      COMMON/KVAL/KCB,HSTART,ISEG
      COMMON/SUBVTQ/IFRAC,VX,TIME,AFAC,ITQ,Q,A,PI,S2MS1,WFLAG,PXA,N,
C      EMOD,RNU,NODIV,A1,A2,A4
      INTEGER WFLAG
      REAL KCB
      COMMON /CBLOC/ CEPT(22),SLOPE(22)
      COMMON/ABLOC/ B,C2,HZ,C4,CONST,E,XAC,TOF,NFLOW,DELTA,HNEW,HOLD,
C      XAOLD,XAINC,HP,FHI
      COMMON/ NONNEWT/ FLN,FLOWK,C5,FLNF1,FLNF
      COMMON/FBLOC/IMU
      DIMENSION HP(22),QX(22),VX(2,22),TIME(2),PXA(22)
      IF(IFRAC.EQ.1) M=ISEG
      IF(IFRAC.EQ.2) M=IF1X(12.-XA*10.)
      IF(M.GT.11) M=11

```

```

C      THIS REMOVES PROBLEM WITH SMALL -VE XA
C      THIS IS NOT PROGRAMMED FOR N=21 , ONLY N=11 !!!!!!!!!!!!!!!!!!!!!
C      C=1./E-(1./E-TOP) *XA**5
C      C = 1.
C      THIS IS APPROXIMATION TO VARIABLE C(X).
C      IF(Y.LT.HZ) GO TO 10
C      BELOW IS CASE WHEN H > HZ
C      WE ASSUME FLOW THRU ENTIRE VERT. SECTION (ELLIPSE).
C      DP/DX PROP. 64/(PI*H) (ONLY THE FIRST TIME IN EACH RUN;
C      THEN PHI CALCULATED IN DWIDTH IS USED FOR DP/DX.)
C      FP =HZ/Y
C      HS1=DSQRT(1. - FP*FP)
C      F1A=KCB/DSQRT(2.*PI*Y) - TOP*S2MS1*FP/HS1
C      F1B=KCB*DSQRT(TOP*Y) + TOP*S2MS1*HZ*DLG((1.+HS1)/FP)
C      F1=F1A*F1B**3
C      ASSUME CLASSICAL FLUID FLOW ACROSS ENTIRE(ELLIPTICAL)
C      VERTICAL SECTION
C      F2 =-((C4/PHI)*(CEPT(M) - SLOPE(M) *XA))**FLN
C      GO TO 20
C      BELOW IS CASE WHEN H < HZ.
10      F1=TOP/PI * KCB**4 * Y
20      F2 =-((C4/PHI)*(CEPT(M) - SLOPE(M) *XA))**FLN
C      CONTINUE
C      THE (1.-XA)**(1+DELTA) TERM IS X VARIATION FOR BOTH RMU AND Q.
C      THE (1.-XA) IS FOR FLOWK VARIATION ( FOLLOWING NOLTE'S)
C      FLNM1 = FLN-1.
C      DUM=0 MEANS FLOWK IS CONSTANT; OTHERWISE VARYING WITH XA
C      DUM=XA
C      IF (IMU.EQ.0) DUM=0.0
C      FUN = F2 *(1.-DUM)*C5 / (F1A * F1B**FLNP)/Y**FLNM1
C      WRITE(2,100)XA,Y,HZ,KCB,S2MS1,F1,ISEG,CEPT(ISEG),SLOPE(ISEG),
C100 C      F2,FUN,C4
C      FORMAT(' XA,Y,HZ,KCB,S2MS1,F1,ISEG,CEPT,SLOPE,F2,FUN,C4= '/
C      1X,6E13.4 ,I3,/1X,5E13.4 )
C      THIS FUN HAS TO BE MULTIPLIED BY A - SEE Q22 - BUT WE
C      DO IT IN RK2 INSTEAD - IT'S EASIER.
C      RETURN
C      END

```

COMPILER DOUBLE PRECISION

```

C
C      *****
C      SUBROUTINE QTEST(N,QPER,ITERQ)
C      *****
C      THIS SUBROUTINE COMPARES INPUT QX AND OUTPUT QX, AND
C      AVERAGES THEM POINT BY POINT. THE RESULT IS FED BACK
C      AS A NEW INPUT QX AND THE FRACTURE RE-COMPUTED. CYCLE
C      IS REPEATED TIL INPUT AND OUTPUT QX CONVERGES.
C
C      COMMON/QVAL/QX,INQX
C      COMMON/CBLOC/CEPT(22),SLOPE(22)
C      DIMENSION QX(22)
C      REAL INQX(22)
C      DO 10 J=1,N
C      DIFQ =ABS(QX(J) - INQX(J)) / INQX(J) *100.
10      IF(DIFQ.GT.QPER) GO TO 20
C      CONTINUE
50      WRITE(2,50)QPER,ITERQ
C      FORMAT(/ ' FLOW RATE CONVERGES TO WITHIN ',F10.0,' PERCENT AFTER ',
C      I4,' ITERATIONS ')
C      STOP

```

```

20      DO 30 J=1,N
30      QX(J) = (3.*QX(J) + 1.*INQX(J))/4.
C      THIS NEW QX IS WEIGHTED AVERAGE OF INPUT AND OUTPUT QX.
      DO 80 J=2,N
      I = 13-J
      SLOPE(J) = (QX(I-1) - QX(I)) / (0.1 * QX(1))
C      80      CEPT(J) = QX(I-1)/QX(1) + SLOPE(J)*(1.0 - (J-1)*.1)
C      DIVIDING BY QX(1) NORMALIZES THE QX'S TO 1 AT THE WELLBORE

      SLOPE(1) = SLOPE(2)
      CEPT(1) = CEPT(2)
C      WRITE(2,100) (SLOPE(J),J=1,N),(CEPT(J),J=1,N)
C100     FORMAT(/' SLOPE = '/11(E9.3,1X)/
C      ' CEPT = '/11(E9.3,1X))
C      ITERQ = ITERQ + 1
C      IF(ITERQ.GT.5) GO TO 60
      GO TO 70
      60      WRITE(2,90) ITERQ
      90      FORMAT(' QX ITERATIONS EXCEED ',I4,' SO PROGRAM STOPPED ')
      STOP
      70      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION DFX( FLVAR,I )
C*****
C      DFX, RNUMINT INTEGRATES OVER THIS FUNCTION (FOR FLK10)
C*****
      COMMON /DSLOP/ TSLOPE

      IF(I.EQ.1) DFX = FLVAR**(TSLOPE-1.) * SQRT(1.-FLVAR)
      IF(I.EQ.2) DFX = FLVAR**(TSLOPE-1.) / SQRT(1.-FLVAR)
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FFX11(A,B,NODIV,SLOPE,K)
C*****
C      PERFORM THE NUMERICAL INTEGRATION OF WIDTH EQN USING THE
C      TRAPEZOIDAL METHOD TO INTEGRATE OVER F2 FROM A TO B (FOR FLK11)
C*****
C      PERFORM NUMERICAL INTEGRATION

      COMMON/DSLOP/XSL

      XSL = SLOPE
      AREA = 0.0
      WIDTH = (B-A) / NODIV

```

```

DO 30 J = 1,NODIV
  XL = A + (J-1) * WIDTH
  IF(XL.EQ.0.0) XL=WIDTH/10.
  XR = A + J * WIDTH
  IF(K.EQ.1) GOTO 2
  IF(B.EQ.1.0.AND.J.EQ.NODIV) XR=1.-WIDTH/10.
  SMALLA = (WIDTH/2.) * ( DFX(XL,K) + DFX(XR,K) )
  AREA = AREA + SMALLA
  FFX11 = AREA
  WRITE(2,1) XSL,FFX11
C1  FORMAT(' FFX(',E11.4,')= ',E13.6)
  RETURN
END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION ASIN(YY)
C     THIS IS USED TO COMPUTE ARCSIN FROM THE DATA GIVEN
C     IF YY IS POSITIVE, RETURNS 0-90
C     IF YY IS NEGATIVE, RETURNS 270-360
      XX=SQRT(1.-YY**2)
      ASIN=ATAN2(YY,XX)
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION ACOS( XX )
C     THIS IS USED TO COMPUTE ARCCOS FROM THE DATA GIVEN
C     IF XX IS POSITIVE, RETURNS 0-90
C     IF XX IS NEGATIVE, RETURNS 90-180
      YY=SQRT(1.-XX**2)
      ACOS=ATAN2(YY,XX)
      RETURN
      END

```

APPENDIX G

Fortran code and user instructions for vertically asymmetric simulator

1. Program description

ASYMFRAC1, a fracture simulator in which fracture height is self-determined, is appropriate if the vertical fracture is highly elongated (length/height ratio $\gtrsim 4$), and if the minimum stresses in the two bounding zones are higher than in the payzone, but not equal. Vertical lithostatic or hydrostatic gradients have been encoded but not debugged or tested. The fluid flow is assumed to be 1D along the payzone direction. Variations in elastic modulus from one zone to another are not included, nor are time variations in pumping rate or viscosity at the wellbore.

The program solves simultaneously the four equations in the text, (1) - (4), subject to the boundary condition that the fracture height at the tip of the long axis is approximately equal to the payzone height.

The program solves for 2 closely-spaced fractures at just one station. If other stations are to be solved simultaneously, refer to subsection 10. below.

2.	<u>Main program</u> ASYMFRAC1	<u>Subroutines</u> VLTQ RK2 KCONT WCOZ7 CLPHI	<u>Functions</u> FUN CINT RKB RKC FUN1 FUN2 FUN3 FUN4 FN57 FN58 DFX FFX ASIN ACOS
----	----------------------------------	--	---

3. Operating mode

Batch

4. Language

Fortran IV

5. Program developed on

Data General Eclipse minicomputer

6. Total CP time

About 5 min

7. Effect of step size on Runge-Kutta integration

The resolution in all 10 segments along the x-axis is determined by NDIV (normally 110). The resolution in the tip segment is enhanced by the factor MFAC (e.g., MFAC = 5 means steps will be 5 times closer together). If the steps are not close enough in the first segment, the pressures along the fracture, given by PXA values, will be too similar and clearly erroneous. In this case, reduce the step size by increasing MFAC by a factor of 2. As a final check on the results, double NDIV, which reduces the step spacing along the entire x-axis. The results for wellbore half heights, b and c, and wellbore pressure (maximum PXA value) should not change significantly.

8. Too rapid height growth near the wellbore

For certain combinations of input parameters the program may abort because height grows too fast as the integration proceeds from the tip at $x=a$ back to the wellbore. When this happens, heights and pressures will be printed out from the tip to some point between tip and wellbore, followed by a RUNTIME error. If the height is growing rapidly, this is almost certainly the reason. To check it, double the stress contrast S_2-S_1 and a successful simulation should result.

Parameters which can give this error include:

- (i) too small S_2-S_1
- (ii) too small H
- (iii) too high E
- (iv) too large μ
- (iv) too large Q

9. Sample input data file

```

0-----SELECT DEBUG OPTION ZERO=NONE, ONE=SOME, TWO=ALL
2-----SELECTS CALCULATIONS, ONE=TIME, TWO=TIME & Q, THREE=NEITHER
1-----IWRITE - FLAG USED TO SELECT OUTPUT
132000000.0 KCB - FRACTURE TOUGHNESS (DYNES/CM**ONE POINT FIVE)
80.0    HZ(FT) - PAYZONE HEIGHT
5.0     QIN(BPM) - INPUT FLOW RATE (ONE WING)
0.46    NPRIME - INDEX OF POWER LAW FLUID FLOW
9.57    KPRIME - AMPLITUDE OF POWER LAW FLUID FLOW(DYNES-SEC**NPRIME/CM**TWO)
4800000.0 YOUNG'S MODULUS (PSI)
0.22    RNU - POISSON'S RATIO
520.0    S2MS1(PSI) - STRESS CONTRAST AT UPPER BARRIER
1195.0    S3MS1(PSI) - STRESS CONTRAST AT LOWER BARRIER(MUST BE > S2MS1)
400.0    A1(FT) - FRACTURE HALF-LENGTH
0--IMU IF IMU=ZERO KPRIME IS CONSTANT; IMU<>ZERO, KPRIME IS PROP TO (ONE-XA)
0.0      F01 NET GRADIENT
0.0      F02
0.0      F03
0.3      XA1
0.9      XA2
0.85     Q1 - VALUE OF QX INPUT AT X/A = XA ONE
0.444    Q2 - X/A = XA TWO
0.082    Q3 - X/A = ONE
11-----IVER - NUMBER OF WIDTH CALCNS IN EACH VERTICAL SECTION
1-----SELECTS OPTION TO RECOMPUTE AND LIST ALL KI'S FOR CHECKING PURPOSES
110-----NDIV - # OF DIVISIONS TO SPLIT FRACTURE INTO(LENGTHWISE)
0.999    XAZ - X/A POSITION AT WHICH NUMERICAL INTEGRATION BEGINS
100.0    YZF(PSI) - PRESSURE AT FRACTURE TIP X/A=ONE CHOSEN SO THAT H JUST > HZ
11-----N - # OF SEGMENTS FRACTURE WING IS SPLIT INTO(LENGTHWISE)
0.1      DIFFAC - PERCENTAGE ITERATION ON KIB=KIC, KIC=KCB, USUALLY=ONE
1-NRES
5-MFAC - MULT FACTOR USED TO INCREASE RESOLUTION IN TIP SEGMENT(USUALLY FIVE)
0.5-----TSLOPE
0.5-----TCEPT
0.00559-----LEAKC - LEAKOFF COEFFICIENT (CM/SEC**POINT FIVE)
0.1846-----SPURT LOSS COEFF. IN CM CUBED / CM SQUARED
0.01-----% CONVERGENCE FOR TIMES

```

)

10. Sample output data file

The calculation performs two iterations:

- (i) iteration on flow rate profile, $q(x)$
- (ii) iteration on fracture expansion rate, $a(t)$ or $x(t)$. As explained in the text, a short-cut is used for (ii), by locking $\log m = n$ in $x = mt^n$.

In practice, we guess a flow rate profile $q(x)$, and this is input into the program via Q1, Q2, and Q3. The program solves for the first fracture, outputting the height profile: at the top of the fracture in units of half-payzone height (see SHAPE OF FRAC HB), and at the bottom of the fracture (see SHAPE OF FRAC HC). Next is given the net pressure above payzone stress in psi (see PRESSURE FROM TIP TO WELLBORE). Then we have width profiles--11 values in each vertical section of the fracture (in cm).

For the short-cut method used here, values for $n = \text{TSLOPE}$ and $\log_{10} m = \text{TCEPT}$ are locked together. Iteration (ii) above proceeds by varying n until the percentage difference between T1CHECK and T(1) and that between T2CHECK and T(2) are as low as possible, and also as equal as possible, in magnitude, although opposite in sign.

Having done this, the output flow rate profile is computed in bpm (see OUTPUT QX), and compared with the input used in solving for the two fractures (see INPUT QX). If these do not agree to within 5%, a weighted average (3 parts OUTPUT, 1 part INPUT) can be calculated manually, converted to Q1, Q2, Q3, and a new run can be made. This is iteration (i) above.

When this is done, fracture volume is given by VTOT1 (1 wing, cm^3), leakoff and spurt volume by TOTLEAK1 (1 wing, cm^3), and total volume

pumped by TOTAL VOLUME (FRAC 1) (1 wing, cm³)--all these corresponding to the first frac at the station. The corresponding time is given by T1CHECK.

Finally, the above run does not give an accurate calculation of fracture expansion rate, $a(t)$ or $x = m\tau^n$. To improve this requires running the program on 2 separate CRT's using a different frac length or station on each. Initially, each is run, as described above, to obtain a first approximation for the time corresponding to each fracture length. Suppose we find $a_1(t_1)$ and $a_2(t_2)$. Then these points are plotted on a log-log graph and joined by a straight line to get improved values for m, n which can now be input into each run as revised values for TSLOPE, TCEPT that are no longer locked together. This will give improved wellbore heights and pressures, as well as a better length expansion rate $a(t)$ or $x(\tau) = m\tau^n$.

According to the listing of ASYMFrac1 there are two output files: FLMCOZ7·OUT and FLMCOZ7·INT. The one below is FLMCOZ7·OUT. The other file has additional details of the calculations. To avoid writing to this file, insert in the CALL statement early in the main program "@ NULL" in place of FLMCOZ7·INT.

XA1,XA2,XA3: 0.300 0.900 1.000 Q1,Q2,Q3: 0.8500 0.4440 0.0820
 DELTA = 0.46
 XA1,CEPT,SLOPE: 0.30 1.0530 0.6767
 XA2,CEPT2,SLOPE2: 0.90 3.7020 3.6200
 FLAMCOZ7 OUTPUT 92

INPUT VALUES: HZ,Q,NPRIME,KPRIME,KI2,EMOD,RNU,S2-S1,S3-S1,L,DELTA,LEAKC,SP
 80.000 5.000 0.460 0.9570 1 0.1320 9 0.4800 7
 0.22 520.00 1195.00 400.00 0.46 0.55900 -2 0.16460 0

B= 76.6C= 45.1 YZ = 100.0
 SHAPE OF FRAC HB/(HZ/2) FROM TIP TO WELLBORE
 1.0222 1.0900 1.1646 1.2436 1.3246 1.4090

SHAPE OF FRAC HC/(HZ/2) =
 1.4981 1.5925 1.6933 1.8007 1.9157
 1.0042 1.0164 1.0292 1.0420 1.0543 1.0664
 1.0785 1.0905 1.1025 1.1145 1.1265

PRESSURE FROM TIP TO WELLBORE (PSI):
 100.00 167.38 212.76 245.07 271.19 293.48
 313.10 330.68 346.67 361.26 374.71
 IFRAC= 1 N= 11

MAX. VALUE OF WIDTH IN EACH VERTICAL SECTION (CM)
 0.00000 0 0.16570 0 0.21400 0 0.25010 0 0.28040 0 0.30710 0
 0.33180 0 0.35590 0 0.37870 0 0.40040 0 0.42120 0

AREAS OF VERT. SECTIONS XA = 1 TO 0 ARE (CM**2)
 0.00000 0 0.31840 3 0.42170 3 0.49960 3 0.57160 3 0.64100 3
 0.70620 3 0.76610 3 0.83160 3 0.89800 3 0.95950 3

I = 1 WZ(I,J) =
 0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0
 0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0

I = 2 WZ(I,J) =
 0.00000 0 0.88790 -1 0.12760 0 0.14950 0 0.16140 0 0.16570 0
 0.16280 0 0.15250 0 0.13290 0 0.98750 -1 0.00000 0

I = 3 WZ(I,J) =
 0.00000 0 0.98700 -1 0.15850 0 0.19010 0 0.20740 0 0.21400 0
 0.21090 0 0.19780 0 0.17220 0 0.12700 0 0.00000 0

I = 4 WZ(I,J) =
 0.00000 0 0.78850 -1 0.17600 0 0.21790 0 0.24070 0 0.25010 0
 0.24730 0 0.23220 0 0.20210 0 0.14790 0 0.00000 0

I = 5 WZ(I,J) =
 0.00000 0 0.65600 -1 0.18470 0 0.23890 0 0.26790 0 0.28040 0
 0.27830 0 0.26180 0 0.22780 0 0.16550 0 0.00000 0

I = 6 WZ(I,J) =
 0.00000 0 0.59230 -1 0.18470 0 0.25480 0 0.29090 0 0.30710 0
 0.30610 0 0.28860 0 0.25110 0 0.18120 0 0.00000 0

I = 7 WZ(I,J) =
 0.00000 0 0.55160 -1 0.17100 0 0.26610 0 0.31070 0 0.33120 0
 0.33180 0 0.31370 0 0.27290 0 0.19570 0 0.00000 0

I = 8 WZ(I,J) =
 0.00000 0 0.52290 -1 0.14000 0 0.27240 0 0.32750 0 0.35320 0
 0.35590 0 0.33750 0 0.29380 0 0.20940 0 0.00000 0

I = 9 WZ(I,J) =
 0.00000 0 0.50130 -1 0.12600 0 0.27270 0 0.34140 0 0.37350 0
 0.37870 0 0.36040 0 0.31410 0 0.22270 0 0.00000 0

I = 10 WZ(I,J) =
 0.00000 0 0.48450 -1 0.11670 0 0.26440 0 0.35220 0 0.39190 0
 0.40040 0 0.38270 0 0.33400 0 0.23570 0 0.00000 0

I = 11 WZ(I,J) =
 0.00000 0 0.47130 -1 0.10980 0 0.23420 0 0.35930 0 0.40850 0
 0.41130 0 0.39400 0 0.35750 0 0.24470 0 0.00000 0

0.42120 0 0.40440 0 0.33300 0 0.24870 0 0.00000 0
VOLUME(X) AS X=A TO 0 : (CGS)

.19410 60.45120 60.56160 60.65300 60.73920 60.82120 60.89750 6

93

.97390 60.10540 70.11320 7

B= 80.7C= 45.5 YZ = 100.0
SHAPE OF FRAC HB/(HZ/2) FROM TIP TO WELLBORE

1.0222 1.0882 1.1618 1.2401 1.3199 1.4028

1.4899 1.5820 1.6802 1.7849 1.8969 2.0174

SHAPE OF FRAC HC/(HZ/2) =

1.0042 1.0161 1.0288 1.0414 1.0536 1.0656

1.0774 1.0892 1.1010 1.1128 1.1246 1.1364

PRESSURE FROM TIP TO WELLBORE (PSI):

100.00 166.21 211.78 244.03 270.00 292.12

311.57 328.98 344.84 359.36 372.74 385.17

IFRAC= 2 N= 11

MAX. VALUE OF WIDTH IN EACH VERTICAL SECTION (CM)

0.00000 0 0.18450 0 0.21290 0 0.24890 0 0.27900 0 0.30540 0
0.32970 0 0.35340 0 0.37600 0 0.39750 0 0.41810 0 0.43780 0

AREAS OF VERT. SECTIONS XA= 1 TO 0 ARE (CM**2)

0.00000 0 0.31590 3 0.41920 3 0.49700 3 0.56800 3 0.63650 3
0.70110 3 0.75950 3 0.82360 3 0.88690 3 0.95170 3 0.10120 4

I = 1 WZ(I,J) =
0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0

0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0

I = 2 WZ(I,J) =
0.00000 0 0.88400 -1 0.12680 0 0.14840 0 0.16030 0 0.16450 0

0.16160 0 0.15130 0 0.13190 0 0.98040 -1 0.00000 0

I = 3 WZ(I,J) =
0.00000 0 0.98950 -1 0.15800 0 0.18920 0 0.20640 0 0.21290 0

0.20980 0 0.19670 0 0.17130 0 0.12640 0 0.00000 0

I = 4 WZ(I,J) =
0.00000 0 0.80080 -1 0.17560 0 0.21700 0 0.23960 0 0.24890 0

0.24610 0 0.23110 0 0.20110 0 0.14720 0 0.00000 0

I = 5 WZ(I,J) =
0.00000 0 0.66220 -1 0.18460 0 0.23800 0 0.26660 0 0.27900 0

0.27680 0 0.26040 0 0.22650 0 0.16460 0 0.00000 0

I = 6 WZ(I,J) =
0.00000 0 0.59710 -1 0.18520 0 0.25400 0 0.28950 0 0.30540 0

0.30430 0 0.28690 0 0.24960 0 0.18020 0 0.00000 0

I = 7 WZ(I,J) =
0.00000 0 0.55570 -1 0.17340 0 0.26540 0 0.30920 0 0.32930 0

0.32970 0 0.31160 0 0.27110 0 0.19450 0 0.00000 0

I = 8 WZ(I,J) =
0.00000 0 0.52650 -1 0.14210 0 0.27210 0 0.32590 0 0.35110 0

0.35340 0 0.33510 0 0.29170 0 0.20810 0 0.00000 0

I = 9 WZ(I,J) =
0.00000 0 0.50460 -1 0.12760 0 0.27320 0 0.34000 0 0.37120 0

0.37600 0 0.35770 0 0.31170 0 0.22120 0 0.00000 0

I = 10 WZ(I,J) =
0.00000 0 0.48760 -1 0.11800 0 0.26650 0 0.35100 0 0.38950 0

0.39750 0 0.37970 0 0.33130 0 0.23400 0 0.00000 0

I = 11 WZ(I,J) =
0.00000 0 0.47410 -1 0.11090 0 0.24390 0 0.35660 0 0.40610 0

0.41810 0 0.43780 0 0.45690 0 0.47550 0 0.49360 0

```

0.41810 0 0.40110 0 0.33080 0 0.24870 0 0.00000 0
I = 12 NZ(I,J) =
0.00000 0 0.46320 -1 0.10530 0 0.20810 0 0.36190 0 0.42090 0 94
0.43780 0 0.42230 0 0.37000 0 0.25950 0 0.00000 0
VOLUME(X) AS X=A TO 0 : (CGS)
.19260 60.44810 60.55850 60.64920 60.73420 60.81540 60.89040 6
.96510 60.10440 70.11220 70.11970 7
TSLOPE,TEXPONENT = 0.56863 1.75860 , TCEPT,TDIV = 0.5686 3.7037
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(I) AS I=1 TO N+1
0.00000 0 0.65670 2 0.22220 3 0.45340 3 0.75190 3 0.11130 4
0.15340 4 0.20120 4 0.25440 4 0.31300 4 0.37670 4 0.44540 4

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.42540 8 VTOT1(=FRAC VOL)= 0.74780 7
TOTLEAK2(=LEAKOFF + SPINT VOL)= 0.49630 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.37670 4, T(2)= 0.44540 4
(SEC,MIN) : T1CHECK = 0.37890 4 0.63160 2 , T2CHECK = 0.44280 4 0.73800 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.50020 8 0.31580 3 0.13220 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58450 8 0.36900 3 0.15440 5

X DIFF. BETWEEN T1CHECK AND T(1) = 0.594340
X DIFF. BETWEEN T2CHECK AND T(2) = -0.595907
DT (T2CHECK-T1CHECK)= 0.63850 3 SEC., T(2)-T(1)= 0.68740 3 SEC.

FRAC 1
LEAKRATE(X) AS X=0 TO A :
0.54390 3 0.55230 3 0.56780 3 0.59140 3 0.62530 3 0.67430 3
0.74840 3 0.87190 3 0.11340 4 0.48130 4

FRAC 2
LEAKRATE(X) AS X=0 TO A :
0.49990 3 0.50630 3 0.51820 3 0.53590 3 0.56060 3 0.59500 3
0.64380 3 0.71650 3 0.83690 3 0.10900 4 0.25880 4

OUTPUT Q(X) AS X=A TO 0:
0.50000 1 0.47560 1 0.45060 1 0.42490 1 0.39850 1 0.37080 1
0.34070 1 0.30750 1 0.26930 1 0.22000 1 0.41590 0
INPUT QX, AS X=A TO 0:
0.50000 1 0.47660 1 0.45170 1 0.42500 1 0.39120 1 0.35730 1
0.32350 1 0.28970 1 0.25580 1 0.22200 1 0.41000 0
XA1 = 0.3000, XA2 = 0.9000
QX AT XA=XA1 USING DELT A CALC. 0.42500 1
QX AT XA = XA2, USING SLOPE AND CEPT 0.22200 1
MAX.(QX OUTPUT, XDIFF) = 0.30750 1 0.61660 1

```

DEST=IDP.FAC USER=IDP.FAC QUEUE=LPT DEVICE=aLPB
SEQ=494 UPRI=128 LPP=88 CPL=80 COPIES=1 LIMIT=33

```

      CREATED: 27-SEP-84 22:02:30
      ENQUEUED: 27-SEP-84 22:20:20
      PRINTING: 27-SEP-84 22:20:20

```

PATH=:AOS1:PHYSICS:IDP.FAC:REACH:FLMCO27.INT

[illegible]

ORG ACADEMIC/RESEARCH COMPUTING

AUS REV 04.38
AOS XLPT REV 04.21

```

IDEBUG = 2 0
IIR = 2
IWRITE = 1
KCB = 0.1320 9 CGS
HZ = 80.0 FT
G/2 = 5.0 BPM
NPRIME = 0.460
KPRIME = 9.570 CGS
YOUNG'S MODULUS = 0.4800 7 PSI
POISS. RATIO = 0.22
S2-S1 = 520.0 PSI
S3-S1 = 1195.0 PSI
A = 400.0 FT
IMU = 0 ; IMU = 0 MEANS MU(X) IS CONST,
      IMU = 1 MEANS MU(X) IS PROPORTIONAL TO 1 - X/A

PO1,PO2,PO3 = 0.0000 0.0000 0.0000 PSI/FT
XA1,XA2,XA3: 0.300 0.900 1.000 W1,W2,W3: 0.8500 0.4440 0.0820
DELTA = 0.46
XA1,CEPT,SLOPE: 0.30 1.0530 0.6767
XA2,CEPT2,SLOPE2: 0.90 3.7020 3.8200
IVER = 11 ; NO OF WIDTH CALC PER VERTICAL SECTION
ICOMP = 1
NDIV = 110 ; NO OF DIVISIONS WE BREAK FRAC HALF LENGTH INTO
XAZ = 0.999000 ; STARTING X/A POSITION
YZP = 100.000 PSI ; STARTING PRESSURE AT X=A
N = 11 ; NO. OF SEGMENTS HALF LENGTH IS DIVIDED INTO
ALLOWANCE FOR 0.10% ERROR IN ITERATING UNTIL KIC = KC, ETC.
NRES = 1
MFAC = 5 ; EXTRA RESOLUTION IN FIRST SEGMENT
LEAKC = 0.55900 -2 ; LEAKOFF COEFF. (IN CGS)
SP = 0.18460 0 ; SPURT LOSS COEFF
INITIAL VALUES OF TSLOPE,TCEPT = 0.50 0.50
% CONVERGENCE FOR TIME = 0.0100
PHI = 0.52360 0
ISEG = 2 F1 = 0.9980 21 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9990 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3230 0 FUN = -
ISEG = 2 F1 = 0.9980 21 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9980 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3280 0 FUN = -
ISEG = 2 F1 = 0.9980 21 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9980 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3280 0 FUN = -
ISEG = 2 F1 = 0.9960 21 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9970 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3340 0 FUN = -
STEP = 0.1820 -2 XA = 0.9990 0 T1,T2,T3,T4 = 0.7600 1 0.7730 1 0.7
ISEG = 2 F1 = 0.1020 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9970 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3340 0 FUN = -
ISEG = 2 F1 = 0.1020 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9960 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3390 0 FUN = -
ISEG = 2 F1 = 0.1020 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9960 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3390 0 FUN = -
ISEG = 2 F1 = 0.1020 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9950 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3450 0 FUN = -
STEP = 0.1820 -2 XA = 0.9970 0 T1,T2,T3,T4 = 0.7650 1 0.7780 1 0.7
ISEG = 2 F1 = 0.1050 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9950 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3450 0 FUN = -
ISEG = 2 F1 = 0.1050 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9940 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3500 0 FUN = -
ISEG = 2 F1 = 0.1050 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9940 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3500 0 FUN = -
ISEG = 2 F1 = 0.1050 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9940 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3550 0 FUN = -
STEP = 0.1820 -2 XA = 0.9950 0 T1,T2,T3,T4 = 0.7690 1 0.7810 1 0.7
ISEG = 2 F1 = 0.1080 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9940 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3550 0 FUN = -
ISEG = 2 F1 = 0.1080 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9930 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3600 0 FUN = -
ISEG = 2 F1 = 0.1080 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9930 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3600 0 FUN = -
ISEG = 2 F1 = 0.1080 22 XA2 = 0.9000 0 C4 = 0.0000 0
XA = 0.9920 0 CEPT2 = 0.3700 ISLOPE2 = 0.3620 1 H = 0.2470 4 F2 = 0.3650 0 FUN = -
STEP = 0.1820 -2 XA = 0.9940 0 T1,T2,T3,T4 = 0.7720 1 0.7830 1 0.7
STEP = 0.1820 -2 XA = 0.9920 0 T1,T2,T3,T4 = 0.7730 1 0.7840 1 0.7
STEP = 0.1820 -2 XA = 0.9900 0 T1,T2,T3,T4 = 0.7740 1 0.7830 1 0.7
PHI = 0.51070 0
C1,C2,C4 = 0.5750-11 0.2280 8 0.0000 USTEP = 0.1820 -2

PHI = 0.49840 0
PHI = 0.48610 0
PHI = 0.47660 0
PHI = 0.46770 0
PHI = 0.45540 0
PHI = 0.43810 0
PHI = 0.42340 0
PHI = 0.40840 0
PHI = 0.39200 0

RXA(1): 0.9990 0 0.8990 0 0.7990 0 0.6990 0 0.5990 0 0.4990 0

```


RXA(1): 0.399D 0 0.299D 0 0.199D 0 0.990D -1 -0.100D -2
 ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

27

A= 400.0FT, B= 76.6FT C= 45.1FT AT THE WELLBORE

PRESSURE FROM TIP TO WELLBORE (PSI):
 100.00 167.38 212.78 245.07 271.19 293.42

SHAPE OF FRAC HB/(HZ/2) FROM TIP TO WELLBORE
 313.10 330.68 346.67 361.26 374.71
 1.0222 1.0900 1.1646 1.2436 1.3246 1.4090

SHAPE OF FRAC HC/(HZ/2) = 1.4981 1.5925 1.6933 1.8007 1.9157
 1.0042 1.0164 1.0292 1.0420 1.0543 1.0664

IFRAC= 1.0785 1.0905 1.1025 1.1145 1.1265
 1 N= 11

MAX. VALUE OF WIDTH IN EACH VERTICAL SECTION (CM)
 0.0000D 0 0.1657D 0 0.2140D 0 0.2501D 0 0.2804D 0 0.3071D 0
 0.3318D 0 0.3559D 0 0.3787D 0 0.4004D 0 0.4212D 0

AREAS OF VERT. SECTIONS XA = 1 TO 0 ARE (CM**2)
 0.0000D 0 0.3184D 3 0.4217D 3 0.4996D 3 0.5716D 3 0.6410D 3
 0.7062D 3 0.7661D 3 0.8316D 3 0.8980D 3 0.9595D 3

RATIO1 = FRACTIONAL DISTANCE FROM BOTTOM OF CROSS-SECTION TO CENTER OF PZ, RELATIVE TO TOTAL HEIGHT OF SECTION - USE TO SCALE TOTAL SECTION TO GRAPH PAGE

RATIO2 = FRACTIONAL DISTANCE FROM CENTER OF PZ TO TOP OF PZ, REL TO DISTANCE FROM CENTER OF PZ TO TOP OF CROSS SECTION - USE TO LOCATE TOP OF PAYZONE

RATIO3 = FRACTIONAL DISTANCE FROM CENTER OF PZ TO BOTTOM OF PZ, REL TO DISTANCE FROM CENTER OF PZ TO BOTTOM OF CROSS-SECTION - USE TO LOCATE BOTTOM OF PAYZONE

I = 1 WZ(I,J) =
 0.0000D 0 0.0000D 0 0.0000D 0 0.0000D 0 0.0000D 0 0.0000D 0
 0.0000D 0 0.0000D 0 0.0000D 0 0.0000D 0 0.0000D 0

RATIO1 = 0.4955D 0
 RATIO2 = 0.9783D 0
 RATIO3 = 0.9958D 0

I = 2 WZ(I,J) =
 0.0000D 0 0.8879D -1 0.1276D 0 0.1495D 0 0.1614D 0 0.1657D 0
 0.1628D 0 0.1525D 0 0.1329D 0 0.9875D -1 0.0000D 0

RATIO1 = 0.4825D 0
 RATIO2 = 0.9174D 0
 RATIO3 = 0.9838D 0

I = 3 WZ(I,J) =
 0.0000D 0 0.9870D -1 0.1585D 0 0.1901D 0 0.2074D 0 0.2140D 0
 0.2109D 0 0.1978D 0 0.1722D 0 0.1270D 0 0.0000D 0

RATIO1 = 0.4691D 0
 RATIO2 = 0.8587D 0
 RATIO3 = 0.9716D 0

I = 4 WZ(I,J) =
 0.0000D 0 0.7885D -1 0.1760D 0 0.2179D 0 0.2407D 0 0.2501D 0
 0.2473D 0 0.2322D 0 0.2021D 0 0.1479D 0 0.0000D 0

RATIO1 = 0.4559D 0
 RATIO2 = 0.8041D 0
 RATIO3 = 0.9597D 0

I = 5 WZ(I,J) =
 0.0000D 0 0.6560D -1 0.1847D 0 0.2389D 0 0.2679D 0 0.2804D 0
 0.2783D 0 0.2618D 0 0.2278D 0 0.1655D 0 0.0000D 0

RATIO1 = 0.4432D 0
 RATIO2 = 0.7550D 0
 RATIO3 = 0.9465D 0

I = 6 WZ(I,J) =
 0.0000D 0 0.5235D -1 0.1877D 0 0.2500D 0 0.2804D 0 0.2804D 0
 0.2773D 0 0.2618D 0 0.2278D 0 0.1655D 0 0.0000D 0

0.00000 0 0.39230 -1 0.18470 0 0.23480 0 0.23050 0 0.30710 0
 0.30610 0 0.28860 0 0.25110 0 0.18120 0 0.00000 0
 RATIO1 = 0.43080 0
 RATIO2 = 0.70970 0
 RATIO3 = 0.93770 0

98

I = 7 WZ(I,J) =
 0.00000 0 0.55160 -1 0.17100 0 0.26610 0 0.31070 0 0.33120 0
 0.33180 0 0.31370 0 0.27290 0 0.19570 0 0.00000 0

RATIO1 = 0.41860 0
 RATIO2 = 0.68750 0
 RATIO3 = 0.92720 0

I = 8 WZ(I,J) =
 0.00000 0 0.52290 -1 0.14000 0 0.27240 0 0.32750 0 0.35320 0
 0.35590 0 0.33750 0 0.29380 0 0.20940 0 0.00000 0

RATIO1 = 0.40640 0
 RATIO2 = 0.62790 0
 RATIO3 = 0.91700 0

I = 9 WZ(I,J) =
 0.00000 0 0.50130 -1 0.12600 0 0.27270 0 0.34140 0 0.37350 0
 0.37870 0 0.36040 0 0.31410 0 0.22270 0 0.00000 0

RATIO1 = 0.39430 0
 RATIO2 = 0.59060 0
 RATIO3 = 0.90700 0

I = 10 WZ(I,J) =
 0.00000 0 0.48450 -1 0.11670 0 0.26440 0 0.35220 0 0.39190 0
 0.40040 0 0.38270 0 0.33400 0 0.23570 0 0.00000 0

RATIO1 = 0.38230 0
 RATIO2 = 0.55540 0
 RATIO3 = 0.89730 0

I = 11 WZ(I,J) =
 0.00000 0 0.47130 -1 0.10980 0 0.23420 0 0.35930 0 0.40850 0
 0.42120 0 0.40440 0 0.35360 0 0.24870 0 0.00000 0

RATIO1 = 0.37030 0
 RATIO2 = 0.52200 0
 RATIO3 = 0.88770 0

FRAC1 WITHOUT LEAKOFF: VOL = 0.74780 7 CGS

TIME = 566.5 SEC

VOLUME(X) AS X=A TO 0 : (CGS)

.19410 60.45120 60.56160 60.65300 60.73920 60.82120 60.89750 6

.97390 60.10540 70.11320 7

NRES = 1
 MFAC = 5 ; EXTRA RESOLUTION IN FIRST SEGMENT
 LEAKC = 0.55900 -2 ; LEAKOFF COEFF. (IN CGS)
 SP = 0.18460 0 ; SPURT LOSS COEFF
 INITIAL VALUES OF TSLUPE, TCEPT = 0.50 0.50
 % CONVERGENCE FOR TIME = 0.0100

PHI = 0.52360 0
 STEP= 0.1820 -2 XA= 0.9990 0 T1,T2,T3,T4= 0.7600 1 0.7730 1 0.7
 STEP= 0.1820 -2 XA= 0.9970 0 T1,T2,T3,T4= 0.7630 1 0.7760 1 0.7
 STEP= 0.1820 -2 XA= 0.9950 0 T1,T2,T3,T4= 0.7650 1 0.7770 1 0.7
 STEP= 0.1820 -2 XA= 0.9940 0 T1,T2,T3,T4= 0.7660 1 0.7770 1 0.7
 STEP= 0.1820 -2 XA= 0.9920 0 T1,T2,T3,T4= 0.7660 1 0.7760 1 0.7
 STEP= 0.1820 -2 XA= 0.9900 0 T1,T2,T3,T4= 0.7640 1 0.7740 1 0.7

PHI = 0.51100 0
 C1,C2,C4 = 0.5750-11 0.2280 8 0.0000 0STEP= 0.1820 -2

PHI = 0.49880 0
 PHI = 0.48660 0
 PHI = 0.47710 0
 PHI = 0.46830 0
 PHI = 0.45710 0
 PHI = 0.43980 0
 PHI = 0.42520 0
 PHI = 0.41060 0
 PHI = 0.39510 0
 PHI = 0.38140 0

KXA(I): 0.9990 0 0.9080 0 0.8170 0 0.7260 0 0.6350 0 0.5440 0

RXA(1): 0.4540 0 0.3630 0 0.2720 0 0.1810 0 0.8990 -1 -0.1000 -2

ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ????????

99

A= 440.0FT, B= 80.7FT C= 45.5FT AT THE WELLBORE

PRESSURE FROM TIP TO WELLBORE (PSI):

100.00 166.21 211.78 244.03 270.00 292.12

311.57 328.98 344.84 359.36 372.74 385.17

THE 2ND PRESSURE VALUE IN FRAC2 SHOULD BE
LESS THAN THE 2ND VALUE IN FRAC1. THE LAST
VALUE IN FRAC2 SHOULD BE GREATER THAN THE
LAST VALUE IN FRAC1. ARE THEY ??????????

SHAPE OF FRAC HB/(HZ/2) FROM TIP TO WELLBORE

1.0222 1.0882 1.1618 1.2401 1.3199 1.4028

1.4899 1.5820 1.6802 1.7849 1.8969 2.0174

SHAPE OF FRAC HC/(HZ/2) =

1.0042 1.0161 1.0288 1.0414 1.0536 1.0656

1.0774 1.0892 1.1010 1.1128 1.1246 1.1364

IFRAC= 2 N= 11

MAX. VALUE OF WIDTH IN EACH VERTICAL SECTION (CM)

0.00000 0 0.16450 0 0.21290 0 0.24890 0 0.27900 0 0.30540 0
0.32970 0 0.35340 0 0.37600 0 0.39750 0 0.41810 0 0.43780 0

AREAS OF VERT. SECTIONS XA = 1 TO 10 ARE (CM**2)

0.00000 0 0.31590 3 0.41920 3 0.49700 3 0.56800 3 0.63650 3
0.70110 3 0.75950 3 0.82360 3 0.88690 3 0.95170 3 0.10120 4

RATIO1 = FRACTIONAL DISTANCE FROM BOTTOM OF CROSS-
SECTION TO CENTER OF PZ, RELATIVE TO TOTAL
HEIGHT OF SECTION - USE TO SCALE TOTAL
SECTION TO GRAPH PAGE

RATIO2 = FRACTIONAL DISTANCE FROM CENTER OF PZ TO
TOP OF PZ, REL TO DISTANCE FROM CENTER OF
PZ TO TOP OF CROSS SECTION - USE TO LOCATE
TOP OF PAYZONE

RATIO3 = FRACTIONAL DISTANCE FROM CENTER OF PZ TO
BOTTOM OF PZ, REL TO DISTANCE FROM CENTER
OF PZ TO BOTTOM OF CROSS-SECTION - USE TO
LOCATE BOTTOM OF PAYZONE

I = 1 WZ(I,J) =

0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0

0.00000 0 0.00000 0 0.00000 0 0.00000 0 0.00000 0

RATIO1 = 0.49550 0

RATIO2 = 0.97830 0

RATIO3 = 0.99580 0

I = 2 WZ(I,J) =

0.00000 0 0.88400 -1 0.12680 0 0.14840 0 0.16030 0 0.16450 0

0.16160 0 0.15130 0 0.13190 0 0.98040 -1 0.00000 0

RATIO1 = 0.48290 0

RATIO2 = 0.91890 0

RATIO3 = 0.98410 0

I = 3 WZ(I,J) =

0.00000 0 0.98950 -1 0.15800 0 0.18920 0 0.20640 0 0.21290 0

0.20980 0 0.19670 0 0.17130 0 0.12640 0 0.00000 0

RATIO1 = 0.46960 0

RATIO2 = 0.86070 0

RATIO3 = 0.97200 0

I = 4 WZ(I,J) =

0.00000 0 0.80000 -1 0.17500 0 0.21700 0 0.23960 0 0.24890 0

0.24610 0 0.23110 0 0.20110 0 0.14720 0 0.00000 0

RATIO1 = 0.45650 0

RATIO2 = 0.80640 0

RATIO3 = 0.95020 0

I = 5 WZ(I,J) =

0.00000 0 0.45330 -1 0.18040 0 0.23800 0 0.26540 0 0.27900 0

```

0.000000 0 0.000000 -1 0.104000 0 0.250000 0 0.200000 0 0.270000 0
0.270000 0 0.260400 0 0.220500 0 0.164000 0 0.000000 0 100
RATIO1 = 0.44390 0
RATIO2 = 0.75770 0
RATIO3 = 0.94910 0
I = 6 WZ(I,J) =
0.000000 0 0.59710 -1 0.18520 0 0.25400 0 0.26950 0 0.30540 0
0.30430 0 0.26690 0 0.24960 0 0.18020 0 0.00000 0
RATIO1 = 0.43170 0
RATIO2 = 0.71290 0
RATIO3 = 0.93850 0
I = 7 WZ(I,J) =
0.000000 0 0.55570 -1 0.17340 0 0.26540 0 0.30920 0 0.32930 0
0.32970 0 0.31100 0 0.27110 0 0.19450 0 0.00000 0
RATIO1 = 0.41970 0
RATIO2 = 0.67120 0
RATIO3 = 0.92820 0
I = 8 WZ(I,J) =
0.000000 0 0.52050 -1 0.14210 0 0.27210 0 0.32590 0 0.35110 0
0.35340 0 0.33510 0 0.29170 0 0.20810 0 0.00000 0
RATIO1 = 0.40780 0
RATIO2 = 0.63210 0
RATIO3 = 0.91810 0
I = 9 WZ(I,J) =
0.000000 0 0.50460 -1 0.12700 0 0.27320 0 0.34000 0 0.37120 0
0.37600 0 0.35770 0 0.31170 0 0.22120 0 0.00000 0
RATIO1 = 0.39590 0
RATIO2 = 0.59520 0
RATIO3 = 0.90830 0
I = 10 WZ(I,J) =
0.000000 0 0.48700 -1 0.11800 0 0.26650 0 0.35100 0 0.38950 0
0.39750 0 0.37970 0 0.33130 0 0.23400 0 0.00000 0
RATIO1 = 0.38400 0
RATIO2 = 0.56030 0
RATIO3 = 0.89870 0
I = 11 WZ(I,J) =
0.000000 0 0.47410 -1 0.11090 0 0.24390 0 0.35860 0 0.40610 0
0.41810 0 0.40110 0 0.35000 0 0.24670 0 0.00000 0
RATIO1 = 0.37220 0
RATIO2 = 0.52720 0
RATIO3 = 0.88920 0
I = 12 WZ(I,J) =
0.000000 0 0.46320 -1 0.10530 0 0.20810 0 0.36190 0 0.42090 0
0.43780 0 0.42230 0 0.37000 0 0.25950 0 0.00000 0
RATIO1 = 0.36030 0
RATIO2 = 0.49570 0
RATIO3 = 0.88000 0
VOLUME(X) AS X=A TO 0 : (CGS)
.19200 60.44810 60.55850 60.64920 60.73420 60.81540 60.89040 6
.96510 60.10440 70.11220 70.11970 7
ISLOPE,TEXPONENT = 0.50000 2.00000 , TCEPT,TDIV = 0.5000 3.162
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(I) AS I=1 TO N+1
0.000000 0 0.16000 3 0.64000 3 0.14400 4 0.25600 4 0.40000 4
0.57600 4 0.78400 4 0.10240 5 0.12960 5 0.16000 5 0.19200 5
FX(1) = 0.1589170 1
FX(2) = 0.3179590 1
TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.77790 8 VTOT1(=FRAC VOL)= 0.74780 7

```

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.75720 0 VTOT1(=FRAC VOL)= 0.08170 7
T(1)= 0.16000 5, T(2)= 0.19360 5
(SEC,MIN) : T1CHECK = 0.64600 4 0.10770 3 , T2CHECK = 0.78920 4 0.12820 5
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.85270 8 0.53830 3 0.22530 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.10150 9 0.64100 3 0.26820 5 101

% DIFF. BETWEEN T1CHECK AND T(1) = -147.695506
% DIFF. BETWEEN T2CHECK AND T(2) = -151.695450

DT (T2CHECK-T1CHECK)= 0.12320 4 SEC., T(2)-T(1)= 0.33600 4 SEC.

TSLOPE,TEXPONENT = 0.75000 1.33333 , TCEPT,TDIV = 0.7500 5.6234
LEAKC = 0.005590 SP = 0.184600 IN CGS

TAU(1) AS I=1 TO N+1
0.00000 0 0.13680 2 0.34470 2 0.59190 2 0.86860 2 0.11700 3
0.14910 3 0.18320 3 0.21890 3 0.25610 3 0.29470 3 0.33470 3

FX(1) = 0.9624950 0
FX(2) = 0.2420060 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.19210 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.21730 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.29470 3, T(2)= 0.33470 3
(SEC,MIN) : T1CHECK = 0.20220 4 0.33700 2 , T2CHECK = 0.22990 4 0.36320 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.26690 8 0.16850 3 0.70520 4
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.30350 8 0.19160 3 0.80180 4

% DIFF. BETWEEN T1CHECK AND T(1) = 85.425244
% DIFF. BETWEEN T2CHECK AND T(2) = 85.443039

DT (T2CHECK-T1CHECK)= 0.27680 3 SEC., T(2)-T(1)= 0.39940 2 SEC.

TSLOPE,TEXPONENT = 0.62500 1.60000 , TCEPT,TDIV = 0.6250 4.2170
LEAKC = 0.005590 SP = 0.184600 IN CGS

TAU(1) AS I=1 TO N+1
0.00000 0 0.36580 2 0.11090 3 0.21220 3 0.33620 3 0.48040 3
0.64320 3 0.82310 3 0.10190 4 0.12310 4 0.14560 4 0.16960 4

FX(1) = 0.1210720 1
FX(2) = 0.2730400 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.30170 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.34860 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.14560 4, T(2)= 0.16960 4
(SEC,MIN) : T1CHECK = 0.28520 4 0.47540 2 , T2CHECK = 0.32940 4 0.54900 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.37650 8 0.23770 3 0.59470 4
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.43480 8 0.27450 3 0.11490 5

% DIFF. BETWEEN T1CHECK AND T(1) = 48.938483
% DIFF. BETWEEN T2CHECK AND T(2) = 48.499228

DT (T2CHECK-T1CHECK)= 0.44160 3 SEC., T(2)-T(1)= 0.23990 3 SEC.

TSLOPE,TEXPONENT = 0.56250 1.77778 , TCEPT,TDIV = 0.5625 3.6517
LEAKC = 0.005590 SP = 0.184600 IN CGS

TAU(1) AS I=1 TO N+1
0.00000 0 0.70490 2 0.24170 3 0.49700 3 0.82880 3 0.12320 4
0.17040 4 0.22410 4 0.28420 4 0.35040 4 0.42260 4 0.50060 4

FX(1) = 0.1378810 1
FX(2) = 0.2932660 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.44490 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.52200 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.42260 4, T(2)= 0.50060 4
(SEC,MIN) : T1CHECK = 0.39370 4 0.65620 2 , T2CHECK = 0.46070 4 0.76790 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.51970 8 0.32810 3 0.13730 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.60810 8 0.38390 3 0.16070 5

% DIFF. BETWEEN T1CHECK AND T(1) = -7.330433
% DIFF. BETWEEN T2CHECK AND T(2) = -8.652261

DT (T2CHECK-T1CHECK)= 0.67020 3 SEC., T(2)-T(1)= 0.78020 3 SEC.

TSLOPE,TEXPONENT = 0.59375 1.68421 , TCEPT,TDIV = 0.5937 3.9242
LEAKC = 0.005590 SP = 0.184600 IN CGS

TAU(1) AS I=1 TO N+1
0.00000 0 0.49910 2 0.16040 3 0.31750 3 0.51550 3 0.75060 3
0.10200 4 0.13230 4 0.16570 4 0.20200 4 0.24120 4 0.28120 4

FX(1) = 0.1290210 1
FX(2) = 0.2826690 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.35990 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.44490 8 VTOT2(=FRAC VOL)= 0.86170 7

```

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.41000 8 VTOT1(=FRAC VOL)= 0.86170 7
T(1)= 0.24120 4, T(2)= 0.28320 4
(SEC,MIN) : T1CHECK = 0.32930 4 0.54880 2 , T2CHECK = 0.38260 4 0.63760 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.43470 8 0.27440 3 0.11480 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.50500 8 0.31880 3 0.13340 5 102

% DIFF. BETWEEN T1CHECK AND T(1) = 26.742719
% DIFF. BETWEEN T2CHECK AND T(2) = 25.970596

DT (T2CHECK-T1CHECK)= 0.53300 3 SEC., T(2)-T(1)= 0.42000 3 SEC.

TSLOPE,TEXPONENT = 0.57812 1.72973 , TCEPT,TDIV = 0.5781 3.7855
LEAKC = 0.003590 SP = 0.184600 IN CGS
TAU(1) AS I=1 TO N+1
0.00000 0 0.59040 2 0.19580 3 0.39480 3 0.64940 3 0.95530 3
0.13100 4 0.17100 4 0.21540 4 0.26410 4 0.31690 4 0.37360 4

FX(1) = 0.1333290 1
FX(2) = 0.2878380 1

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.39820 8 VTOT1(=FRAC VOL)= 0.74780 7
TOTLEAK2(=LEAKOFF + SPINT VOL)= 0.46530 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.31690 4, T(2)= 0.37360 4
(SEC,MIN) : T1CHECK = 0.35830 4 0.59720 2 , T2CHECK = 0.41770 4 0.69620 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.47300 8 0.29860 3 0.12500 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.55140 8 0.34810 3 0.14570 5

% DIFF. BETWEEN T1CHECK AND T(1) = 11.568966
% DIFF. BETWEEN T2CHECK AND T(2) = 10.557279

DT (T2CHECK-T1CHECK)= 0.59440 3 SEC., T(2)-T(1)= 0.56790 3 SEC.

TSLOPE,TEXPONENT = 0.57031 1.75342 , TCEPT,TDIV = 0.5703 3.7180
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(1) AS I=1 TO N+1
0.00000 0 0.64430 2 0.21720 3 0.44230 3 0.73240 3 0.10630 4
0.14910 4 0.19540 4 0.24690 4 0.30360 4 0.36520 4 0.43160 4

FX(1) = 0.1355730 1
FX(2) = 0.2905180 1

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.42040 8 VTOT1(=FRAC VOL)= 0.74780 7
TOTLEAK2(=LEAKOFF + SPINT VOL)= 0.49220 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.36520 4, T(2)= 0.43160 4
(SEC,MIN) : T1CHECK = 0.37510 4 0.62520 2 , T2CHECK = 0.43810 4 0.73020 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.49510 8 0.31260 3 0.13080 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.57830 8 0.36510 3 0.15280 5

% DIFF. BETWEEN T1CHECK AND T(1) = 2.644265
% DIFF. BETWEEN T2CHECK AND T(2) = 1.487696

DT (T2CHECK-T1CHECK)= 0.63030 3 SEC., T(2)-T(1)= 0.66430 3 SEC.

TSLOPE,TEXPONENT = 0.56641 1.76552 , TCEPT,TDIV = 0.5664 3.6847
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(1) AS I=1 TO N+1
0.00000 0 0.67370 2 0.22910 3 0.46860 3 0.77880 3 0.11550 4
0.15930 4 0.20920 4 0.26480 4 0.32600 4 0.39260 4 0.46460 4

FX(1) = 0.1367190 1
FX(2) = 0.2918830 1

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.43230 8 VTOT1(=FRAC VOL)= 0.74780 7
TOTLEAK2(=LEAKOFF + SPINT VOL)= 0.50670 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.39260 4, T(2)= 0.46460 4
(SEC,MIN) : T1CHECK = 0.38420 4 0.64030 2 , T2CHECK = 0.44910 4 0.74860 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.50710 8 0.32010 3 0.13400 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.59290 8 0.37430 3 0.15660 5

% DIFF. BETWEEN T1CHECK AND T(1) = -2.203484
% DIFF. BETWEEN T2CHECK AND T(2) = -3.439935

DT (T2CHECK-T1CHECK)= 0.64970 3 SEC., T(2)-T(1)= 0.71950 3 SEC.

TSLOPE,TEXPONENT = 0.56836 1.75945 , TCEPT,TDIV = 0.5684 3.7015
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(1) AS I=1 TO N+1
0.00000 0 0.65880 2 0.22300 3 0.45520 3 0.75520 3 0.11180 4
0.15410 4 0.20210 4 0.25570 4 0.31450 4 0.37860 4 0.44770 4

FX(1) = 0.1361440 1
FX(2) = 0.2911990 1

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.42630 8 VTOT1(=FRAC VOL)= 0.74780 7
TOTLEAK2(=LEAKOFF + SPINT VOL)= 0.48830 8 VTOT2(=FRAC VOL)= 0.86170 7

```

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.47750 8 VTOT1(=FRAC VOL)= 0.86170 7
T(1)= 0.37800 4, T(2)= 0.44770 4
(SEC,MIN) : T1CHECK = 0.37900 4 0.63200 2 , T2CHECK = 0.44300 4 0.73930 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.50100 8 0.31630 3 0.13240 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58550 8 0.36900 3 0.15470 5 103

% DIFF. BETWEEN T1CHECK AND T(1) = 0.254200
% DIFF. BETWEEN T2CHECK AND T(2) = -0.941050

DT (T2CHECK-T1CHECK)= 0.63980 3 SEC., T(2)-T(1)= 0.69120 3 SEC.

TSLOPE,TEXPONENT = 0.56934 1.75643 , TCEPT,TDIV = 0.5693 3.7097
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(I) AS I=1 TO N+1
0.00000 0 0.65150 2 0.22010 3 0.44870 3 0.74370 3 0.11010 4
0.15160 4 0.19870 4 0.25130 4 0.30900 4 0.37180 4 0.43960 4
FX(1) = 0.1358580 1
FX(2) = 0.2908580 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.42330 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.49570 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.37180 4, T(2)= 0.43960 4
(SEC,MIN) : T1CHECK = 0.37730 4 0.62890 2 , T2CHECK = 0.44080 4 0.73470 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.49810 8 0.31440 3 0.13160 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58190 8 0.36740 3 0.15370 5

% DIFF. BETWEEN T1CHECK AND T(1) = 1.457553
% DIFF. BETWEEN T2CHECK AND T(2) = 0.281505

DT (T2CHECK-T1CHECK)= 0.63500 3 SEC., T(2)-T(1)= 0.67760 3 SEC.

TSLOPE,TEXPONENT = 0.56885 1.75794 , TCEPT,TDIV = 0.5688 3.7055
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(I) AS I=1 TO N+1
0.00000 0 0.65510 2 0.22160 3 0.45190 3 0.74940 3 0.11090 4
0.15290 4 0.20040 4 0.25350 4 0.31180 4 0.37520 4 0.44360 4
FX(1) = 0.1360010 1
FX(2) = 0.2910280 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.42480 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.49750 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.37520 4, T(2)= 0.44360 4
(SEC,MIN) : T1CHECK = 0.37840 4 0.63070 2 , T2CHECK = 0.44220 4 0.73700 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.49960 8 0.31540 3 0.13200 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58370 8 0.36850 3 0.15420 5

% DIFF. BETWEEN T1CHECK AND T(1) = 0.857973
% DIFF. BETWEEN T2CHECK AND T(2) = -0.327935

DT (T2CHECK-T1CHECK)= 0.63740 3 SEC., T(2)-T(1)= 0.66440 3 SEC.

TSLOPE,TEXPONENT = 0.56860 1.75869 , TCEPT,TDIV = 0.5686 3.7034
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(I) AS I=1 TO N+1
0.00000 0 0.65700 2 0.22230 3 0.45360 3 0.75230 3 0.11140 4
0.15350 4 0.20130 4 0.25460 4 0.31320 4 0.37690 4 0.44570 4
FX(1) = 0.1360730 1
FX(2) = 0.2911140 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.42550 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.49840 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.37690 4, T(2)= 0.44570 4
(SEC,MIN) : T1CHECK = 0.37900 4 0.63170 2 , T2CHECK = 0.44290 4 0.73810 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.50030 8 0.31580 3 0.13220 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58460 8 0.36910 3 0.15450 5

% DIFF. BETWEEN T1CHECK AND T(1) = 0.556613
% DIFF. BETWEEN T2CHECK AND T(2) = -0.634256

DT (T2CHECK-T1CHECK)= 0.63860 3 SEC., T(2)-T(1)= 0.68780 3 SEC.

TSLOPE,TEXPONENT = 0.56873 1.75832 , TCEPT,TDIV = 0.5687 3.7041
LEAKC = 0.005590 SP = 0.184600 IN CGS
TAU(I) AS I=1 TO N+1
0.00000 0 0.65600 2 0.22190 3 0.45280 3 0.75080 3 0.11120 4
0.15320 4 0.20090 4 0.25400 4 0.31250 4 0.37610 4 0.44470 4
FX(1) = 0.1360370 1
FX(2) = 0.2910710 1

TITLEAK1(=LEAKOFF + SPINT VOL)= 0.42510 8 VTOT1(=FRAC VOL)= 0.74780 7
TITLEAK2(=LEAKOFF + SPINT VOL)= 0.49800 8 VTOT2(=FRAC VOL)= 0.86170 7

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.47000 8 VTOT1(=FRAC VOL)= 0.66170 7
T(1)= 0.37610 4, T(2)= 0.44470 4
(SEC,MIN) : T1CHECK = 0.37870 4 0.63120 2 , T2CHECK = 0.44250 4 0.73760 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.49990 8 0.31560 3 0.13210 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58410 8 0.36880 3 0.15430 5 104

% DIFF. BETWEEN T1CHECK AND T(1) = 0.707424
% DIFF. BETWEEN T2CHECK AND T(2) = -0.480962

DT (T2CHECK-T1CHECK)= 0.63800 3 SEC., T(2)-T(1)= 0.68610 3 SEC.

TSLOPE,TEXPONENT = 0.56860 1.75851 , TCEPT,TDIV = 0.5687 3.7039
LEAKC = 0.005590 SP = 0.184600 IN CGS

TAU(I) AS I=1 TO N+1
0.00000 0 0.65650 2 0.22210 3 0.45320 3 0.75160 3 0.11130 4

0.15330 4 0.20110 4 0.25430 4 0.31280 4 0.37650 4 0.44520 4

FX(1) = 0.1360550 1
FX(2) = 0.2910920 1

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.42530 8 VTOT1(=FRAC VOL)= 0.74780 7
TOTLEAK2(=LEAKOFF + SPINT VOL)= 0.49820 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.37650 4, T(2)= 0.44520 4
(SEC,MIN) : T1CHECK = 0.37890 4 0.63150 2 , T2CHECK = 0.44270 4 0.73780 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.50010 8 0.31570 3 0.13210 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58440 8 0.36890 3 0.15440 5

% DIFF. BETWEEN T1CHECK AND T(1) = 0.632051
% DIFF. BETWEEN T2CHECK AND T(2) = -0.557575

DT (T2CHECK-T1CHECK)= 0.63830 3 SEC., T(2)-T(1)= 0.68690 3 SEC.

TSLOPE,TEXPONENT = 0.56863 1.75860 , TCEPT,TDIV = 0.5686 3.7037
LEAKC = 0.005590 SP = 0.184600 IN CGS

TAU(I) AS I=1 TO N+1
0.00000 0 0.65670 2 0.22220 3 0.45340 3 0.75190 3 0.11130 4

0.15340 4 0.20120 4 0.25440 4 0.31300 4 0.37670 4 0.44540 4

FX(1) = 0.1360640 1
FX(2) = 0.2911030 1

TOTLEAK1(=LEAKOFF + SPINT VOL)= 0.42540 8 VTOT1(=FRAC VOL)= 0.74780 7
TOTLEAK2(=LEAKOFF + SPINT VOL)= 0.49830 8 VTOT2(=FRAC VOL)= 0.86170 7
T(1)= 0.37670 4, T(2)= 0.44540 4
(SEC,MIN) : T1CHECK = 0.37890 4 0.63160 2 , T2CHECK = 0.44280 4 0.73800 2
TOTAL VOLUME (FRAC1): (CGS,BRL,GAL) 0.50020 8 0.31580 3 0.13220 5
TOTAL VOLUME (FRAC2): (CGS,BRL,GAL) 0.58450 8 0.36900 3 0.15440 5

% DIFF. BETWEEN T1CHECK AND T(1) = 0.594340
% DIFF. BETWEEN T2CHECK AND T(2) = -0.595907

DT (T2CHECK-T1CHECK)= 0.63850 3 SEC., T(2)-T(1)= 0.68740 3 SEC.

FRAC 1
LEAKRATE(X) AS X=0 TO A :
0.54390 3 0.55230 3 0.56780 3 0.59140 3 0.62530 3 0.67430 3
0.74840 3 0.87190 3 0.11340 4 0.48130 4

FRAC 2
LEAKRATE(X) AS X=0 TO A :
0.49990 3 0.50630 3 0.51820 3 0.53590 3 0.56060 3 0.59500 3
0.64380 3 0.71650 3 0.83690 3 0.10900 4 0.25880 4

OUTPUT Q(X) AS X=A TO 0:
0.50000 1 0.47560 1 0.45060 1 0.42490 1 0.39850 1 0.37080 1

0.34070 1 0.30750 1 0.26930 1 0.22000 1 0.41590 0

INPUT QX, AS X=A TO 0:
0.50000 1 0.47660 1 0.45170 1 0.42500 1 0.39120 1 0.35730 1

0.32350 1 0.28970 1 0.25580 1 0.22200 1 0.41000 0

XA1 = 0.3000, XA2 = 0.9000
QX AT XA=XA1 USING DELT A CALC. 0.42500 1
QX AT XA = XA2, USING SLOPE AND CEPT 0.22200 1
MAX.(QX OUTPUT, XQIF) = 0.30750 1 0.61660 1


```

        WRITE(2,602) IWRITE, KCB,HZ,Q,FLN,FLOWK
602      C      FORMAT(' IWRITE= ',I3,' KCB= ',E11.3,' CGS= ',F10.1,
        C      ' FT= ',F10.1,' BPM= ',F10.3,' NPRIME = ',F10.3,' KPRIME = ',
        C      ' F10.3,' CGS')
        WRITE(2,762) EMOD,RNU
762      C      FORMAT(' YOUNG'S MODULUS= ',E12.3,' PSI'
        C      /' POISS.RATIO= ',F10.2)
        WRITE(2,603) S2MS1,S3MS1,L
603      C      FORMAT(' S2-S1= ',F10.1,' PSI'/' S3-S1= ',F10.1,' PSI'
        C      /' A= ',F10.1,' FT')
        WRITE(2,604) IMU
604      C      FORMAT(' IMU = ',I2,' ; IMU = 0 MEANS MU(X) IS CONST,
        C      /' IMU = 1 MEANS MU(X) IS PROPORTIONAL TO 1 - X/A'/' )
        WRITE(2,882) P01,P02,P03
882      C      FORMAT(' P01,P02,P03= ',3F12.4,' PSI/FT')

C      WRITE(2,62) NODIV
C      WRITE(3,62) NODIV
C62     C      FORMAT(1X,I6,' STEPS FOR NUMERICAL INTEGRATION'/' )

C
C      CALCULATE QX FROM PREVIOUS QX VALUES
C
C      XA3 = 1.
C CALC DELTA
        DELTA = 9.99
        IF(XA1.NE.0.0) DELTA = DLOG(Q1) / DLOG(1.- XA1)
C CALC SLOPE,CEPT,SLOPE2,CEPT2
        SLOPE = (Q1 - Q2) / (XA2-XA1)
        CEPT = Q1 + SLOPE*XA1
        SLOPE2 = (Q2 - Q3) / (XA3-XA2)
        CEPT2 = Q2 + SLOPE2*XA2
C DERUG *****
        WRITE(2,460)XA1,XA2,XA3,Q1,Q2,Q3
        WRITE(3,460)XA1,XA2,XA3,Q1,Q2,Q3
460      C      FORMAT(' XA1,XA2,XA3: ',3(F6.3,2X),' Q1,Q2,Q3: ',3(F7.4,2X))
        WRITE(2,530)DELTA,XA1,CEPT,SLOPE,XA2,CEPT2,SLOPE2
        WRITE(3,530)DELTA,XA1,CEPT,SLOPE,XA2,CEPT2,SLOPE2
530      C      FORMAT(' DELTA = ',F11.2/' XA1,CEPT,SLOPE: ',F11.2,2X,
        C      ' 2(F12.4,2X)'/' XA2,CEPT2,SLOPE2: ',F11.2,2X,2(F12.4,2X))
        WRITE(2,540)IVER
540      C      FORMAT(1X,' IVER = ',I4,' ;NO OF WIDTH CALCN PER ',
        C      ' VERTICAL SECTION')
        WRITE(3,6661)HZ,Q,FLN,FLOWK,KCB,EMOD,RNU,S2MS1,S3MS1,L,DELTA,
        C      LEAKC,SP
6661      C      FORMAT(' FLAMCOZ7 OUTPUT'/' INPUT VALUES: HZ,Q,NPRIME,KPRIME,'
        C      ' K12,EMOD,RNU,S2-S1,S3-S1,L,DELTA,LEAKC,SP'/' 3F10.3,3(E10.3,2X),
        C      /5F10.2,3X,E10.4,3X,E10.4)

        WRITE(2,550)ICOMP
550      C      FORMAT(' ICOMP = ',I4)

C      CONVERT UNITS
        S2MS1 = 68940. * S2MS1
        S3MS1 = 68940. * S3MS1
        P01 = P01*68940./30.48
        P02 = P02*68940./30.48
        P03 = P03*68940./30.48
        HZ = 30.48 * HZ
        RF=HZ/2.
        HZH=HZ/2.
        EMOD=EMOD*68940.
        G=EMOD/(2.*(1.+RNU))
        Q=Q*2640.
        A=L*30.48
        C1=(1.-RNU) / G
        WRITE(2,147)C5,PFI
        C147      C      FORMAT(2X,' C5 = ',E12.4,' PFI = ',E12.4)
        C      IA C4 USED ELSEWHERE ??????????
        C      C4=-64.*Q*RMU / PI
        C      C4=C4/C1**3
        C      C2=S2MS1*2./PI

```

```

IFRAC = 1
ICYCLE = 0
IWF = 0
ISC = 0
C
CONTINUE
IYZ = 0
WRITE(2,613) NDIW
613 C FORMAT('NDIW = ',I5,';NO OF DIVISIONS WE BREAK ',
C 'FRAC HALF LENGTH INTO')
WRITE(2,140) XAZ
140 C FORMAT('XAZ = ',F10.6,';STARTING X/A POSITION')
WRITE(2,141) YZF
141 C FORMAT('YZF = ',F10.3,' PSI;STARTING PRESSURE AT X=A')
WRITE(2,142) N
142 C FORMAT('N = ',I3,';NO. OF SEGMENTS HALF ',
C 'LENGTH IS DIVIDED INTO')
WRITE(2,652) DIFFAC
652 C FORMAT('ALLOWANCE FOR ',F10.2,'% ERROR IN ITERATING UNTIL ',
C 'KIC = KC, ETC.')
199 C CONTINUE
C BEGIN NEW ITERATION
C
9 ICYCLE=ICYCLE+1
ISEG = 0
WRITE(2,592) NRES
592 C FORMAT('NRES = ',I4)
WRITE(2,593) MFAC
593 C FORMAT('MFAC = ',I4,';EXTRA RESOLUTION IN FIRST '
C 'SEGMENT')
WRITE(2,594) LEAKC,SP
594 C FORMAT('LEAKC = ',E10.4,';LEAKOFF COEFF.(IN CGS)'/
C 'SP = ',E10.4,';SPURT LOSS COEFF')
WRITE(2,595) TSLOPE,TCEPT,TIMP
595 C FORMAT('INITIAL VALUES OF TSLOPE,TCEPT = ',2F5.2/
C '% CONVERGENCE FOR TIME = ',F8.4)
IHST=0
IF (FLN.GE.0.7) GOTO 520
MM = .16
BB = .45
GOTO 521
520 CONTINUE
MM = .08333
BB = .506
521 CONTINUE
PHI = MM*FLN + BB
FLNP = 2.*FLN + 1.
C5 = -1. * 2.*(FLN + 1.) * (FLNP/FLN * 1./PHI)**FLN
C * (PI/(4.*C1))**FLNP * FLOWK * Q**FLN
WRITE(2,147) PHI
147 C FORMAT('PHI = ',E12.4)
AFAC = 1. + 1./(N-1)
NTEMP=N
NUMSEG=N-1
IF(IFRAC.EQ.2) NTEMP=N+1
IF(IFRAC.EQ.2) NUMSEG=N
XAINC=1./FLOAT(NUMSEG)
STEP=1./FLOAT(NDIW)
L CONVERT UNITS
YZ = YZF*68940.
C
C CALCULATE SOLN OF D.E.ARISING FROM W(P) AND P(W) EQNS USING RUNGE-KUTTA
C
NDIW= NDIW/NUMSEG
ISTART=1
H2= STEP/2.
STEP=STEP/FLOAT(MFAC)
H2=H2/FLOAT(MFAC)
NDIW=NDIW*MFAC
C THIS UPS THE NO OF STEPS IN 1ST SEGMENT BY FACTOR MFAC
C FOR MORE ACCURATE INTEGRATION NEAR XA=1.

```

```

XAOLD=XAZ
ISEG=2
CALL RK2( A,STEP,XAZ,YZ,NDIV,N,PXA,RXA,NRES,MFAC,IFRAC,H2,KDIV,
C          ISEG,KCB,NTEMP)
CALL CLPHI(PHI,WZ,WCON,FLN,ISEG,IVER,PI,EMOD,RNU)
C5 = -1. * 2.**(FLN + 1.) * (FLNP/FLN * 1./PHI)**FLN
C          * (PI/(4.*C1))**FLNP * FLOWK * Q**FLN
C          WRITE(2,147)C5,PHI
C          WRITE(2,147)PHI
615      IF(IWRITE.EQ.1) WRITE(2,615) C1,C2,C4,STEP
        FORMAT(' C1,C2,C4 = ',3E12.3,'STEP=',E12.3/)
        HPR(ISEG) = HBM
        HPC(ISEG) = HCM
C          WRITE(2,19) HPR(ISEG),HPC(ISEG),PXA(2),RXA(2),ISEG
C19      FORMAT(' IN MAIN, HB ',F10.3,'HC=',F10.3,' PXA(I) = ',E12.4,' RXA(I) =
        ',E12.4,'
C          C          ISEG= ',I3)
C          C
C          NOW RESET STEP, KDIV
        STEP=STEP*FLOAT(MFAC)
        H2=H2*FLOAT(MFAC)
        KDIV=KDIV/MFAC
        XAOLD=RXA(2)
        DO 336 ISEG=3,NTEMP
C          CALL RK2( A,STEP,XAZ,YZ,NDIV,N,PXA,RXA,NRES,MFAC,IFRAC,H2,KDIV,
C          ISEG,KCB,NTEMP)
C          CALL CLPHI(PHI,WZ,WCON,FLN,ISEG,IVER,PI,EMOD,RNU)
C          C5 = -1. * 2.**(FLN + 1.) * (FLNP/FLN * 1./PHI)**FLN
C          * (PI/(4.*C1))**FLNP * FLOWK * Q**FLN
C          WRITE(2,147)C5,PHI
C          WRITE(2,147)PHI
C          WRITE(2,353) WCON
C          WRITE(3,353) WCON
C353      FORMAT(/' WCON= ',E13.5/)
        HPR(ISEG)=HBM
        HPC(ISEG)=HCM
C          WRITE(2,19) HPR(ISEG),HPC(ISEG),PXA(ISEG),RXA(ISEG),ISEG
        XAOLD=RXA(ISEG)
        IF(ISEG.EQ.NTEMP) GO TO 334
336      CONTINUE
334      CONTINUE
C          IF(IWRITE.EQ.1)WRITE(2,620)(PXA(I), I=1,NTEMP)
C620      FORMAT(1X,' PXA(I):',6E12.3/)
        IF(IWRITE.EQ.1)WRITE(2,630)(RXA(I), I=1,NTEMP)
630      FORMAT(1X,' RXA(I):',6E12.3/)
        IF(IWRITE.EQ.1) WRITE(2,634)
634      FORMAT(' ONE OF THE RXA'S MUST BE REAL CLOSE TO 0. IS IT ?????????')
        CONTINUE
        CONTINUE
        CONTINUE

AFT=A/30.48
DO 1111 I=1,NTEMP
1111      PXAF(I)=PXA(I)/68940.
        HNB = (HPR(NTEMP) * HZH)/30.48
        HNC = (HPC(NTEMP) * HZH)/30.48
        WRITE(2,500)AFT,HNB,HNC
500      FORMAT(/' A= ',F10.1,'FT, B= ',F10.1,'FT C= ',F10.1,'FT',
C          ' AT THE WELLBORE')
        WRITE(2,501)(PXAF(II),II=1,NTEMP)
501      FORMAT(/' PRESSURE FROM TIP TO WELLBORE (PSI): ',(6F10.2/))
        IF(IFRAC.EQ.2) WRITE(2,5011)
5011     FORMAT(/' THE 2ND PRESSURE VALUE IN FRAC2 SHOULD BE '/
C          ' LESS THAN THE 2ND VALUE IN FRAC1. THE LAST '/
C          ' VALUE IN FRAC2 SHOULD BE GREATER THAN THE '/
C          ' LAST VALUE IN FRAC1. ARE THEY ?????????? ')
        CONTINUE
        HNEWB = HPR(NTEMP) * (HZ/30.48/2.)
        HNEWC = HPC(NTEMP) * (HZ/30.48/2.)
        ALPHA=A/F
        QB=Q/2840.
        GRMU=RMU/.01

```

```

C      WRITE(2,627) QR,SRMU,HNEWB,HNEWC,AFT
C627  C      FORMAT('  ITERATION SUCCESSFUL ON KI',F10.1,' /'
C      C      ' Q= ',F10.1,' R MU= ',F10.1,' R= ',F10.1,' C= ',F10.1,
C      C      ' A = ',F10.1)
C      WRITE(3,607)HNEWB,HNEWC,YZP
607    C      FORMAT(1X/,' R= ',F10.1,' C= ',F10.1,' YZ = ',F10.1)
C      WRITE(2,337) (HFB(I),I=1,NTEMP)
337    C      FORMAT('  SHAPE OF FRAC HB/(HZ/2) FROM TIP TO WELLBORE '
C      C      '/(6F12.4/)')
C      WRITE(3,337) (HFB(I),I=1,NTEMP)
C      WRITE(2,338) (HFC(I),I=1,NTEMP)
338    C      FORMAT('  SHAPE OF FRAC HC/(HZ/2) = ',F10.1,'/(6F12.4/)')
C      WRITE(3,338) (HFC(I),I=1,NTEMP)
C      WRITE(3,501) (FXAP(II),II=1,NTEMP)
C      CONTINUE

C
C      CHECK WHICH CALCULATIONS ARE TO BE DONE - TIME, Q, NEITHER?
C
C      IF (ITQ-2) 770,750,9
C740  C      ONLY CALCULATE TIME
C      CALL VLTQ
C      IFRAC = 0
C      GO TO 9
C      CALCULATE TIME & Q
750  C      CONTINUE
C
C      DIVIDE FRAC INTO N SEGMENTS AND CALC VOLUME OF EA SEGMENT.
C      CALL THIS FRAC1 THEN ITERATE FOR FRAC2 WHICH IS TO BE ONE
C      SEGMENT LONGER.
C
C      IF (IFRAC.EQ.2) GO TO 770
C      CALL VLTQ
C      ICYCLE = 0
C      GO TO 199
C
C      NOW CALC THE VOLUME OF EA OF THE N+1 SEGMENTS FOR FRAC2. FROM
C      SEGMENT VOL'S FOR FRACS 1&2 CALC FLOW RATES DOWN THE FRAC.
770  C      CONTINUE
C      CALL VLTQ
C      STOP
C      END

C
C      COMPILER NOSTACK
C      COMPILER DOUBLE PRECISION
C      *****
C      SUBROUTINE DVCOZ7 - CALC VOLUME, LEAKOFF, TIME, AND Q
C      -- DVCOZ7B + BISECTION METHOD OF CONVERGENCE FOR TSLOPE,TCEPT
C      TO DIF % (092684)
C      *****
C      SUBROUTINE VLTQ
C      DIMENSION VX(2,22),T(2),QX(22),AREAS(22),WZ(22,22),WZM(22,2)
C      DIMENSION FXA(22),HFB(22),HFC(22)
C      DIMENSION TAU(21),TTOTLEAK(2),AVTAU(2),FX(2)
C      INTEGER FFLAG,WFLAG
C      REAL INOX(21),L,LEAKC,LEAKRATE(2,21),LEAKVOLUME(2,21)
C      COMMON /SUBRTQ/IFRAC,VX,T,AFAC,ITQ,Q,A,PI,S2MS1,S3MS1,WFLAG,
C      FXA,N,EMOD,RNU,NODIV
C      COMMON /ABLOC/P,C2,HZ,C4,CONST,E,XAC,TOP,NFLOW,DELTA,HNEWB,
C      HNEWC,HOLDB,HOLDC,XAOLD,XAINC,HFB,HFC

```

```

COMMON /SUBWIDTH/WZ,WZM,AREAS,NUMWID,I,L,H
COMMON /FNCTN/FLP,FL,FFLAG
COMMON /CBLOC/XA1,CEPT,SLOPE,XA2,CEPT2,SLOPE2
COMMON /SUBLEAK/A1,VTOT1
COMMON /EBLOC/HZH,P01,P02,P03
COMMON /GELOC/C1
COMMON /WCBLOC/EL,Y3,Y4,F1,WCON,IVER
COMMON /FBLOC/IWR,ISC,IDEBUG
COMMON /NONNEWT/FLN,FLOWK,C5,FLNP1,FLNP
COMMON /TCONV/TTSLOPE,TCEPT,LEAKC,SP,TIMP

C=====DEBUG=====
      IF (IDEBUG.EQ.1) WRITE(2,20)
20      FORMAT(' IN DVCOZ7 ')
C=====END DEBUG=====

      IF (IFRAC.EQ.2) NUMSEG = N
      IF (IFRAC.EQ.2) GO TO 3
C      IFRAC = 1
C      NUMSEG = N-1
C
C      CALCULATE AREA(FLP)
C
3      XA = .1.0
      XAINCR = 1./ NUMSEG
      IF (IFRAC.EQ.1) AFAC = 1. + XAINCR
C
C      LOOP OVER THE VERTICAL SECTIONS CALCULATING AREAS
C      *AREAS(I)* IS THE AREA OF THE I'TH VERTICAL SECTION IN CGS
C
      AREAS(1) = 0.0
      IEND = NUMSEG + 1
      DO 100 I=2,IEND
C      PROPER CALCULATION OF AREA OF VERTICAL SECTIONS
          HBC= (HPB(I) + HPC(I)) *HZH
          EL=(HPB(I) + HPC(I))*HZH/2.
          ELMB=EL - HPB(I)*HZH
          IF(HPB(I).EQ.HPC(I)) ELMB=0.
          Y3= -(ELMB + HZH)
          Y4= HZH - ELMB
          F1= FXA(I) - ELMB*P01
          IAREA=2
          CALL WCOZ7(I,WZ,WZM,AREAS,HPB,HPC,C1,IAREA)
          WMAX = WZM(I,1)
          CALL CLPHI(PHI,WZ,WZM,FLN,IEND,IVER,PI,EMOD,RNU)
          TYPE ' IN DVCOZ7, PHI = ',PHI
C      THIS CALCS AREA OF A VERTL SECTION FROM THE WIDTH PROFILE
C      -- ITS DONE PROPERLY!!
100      CONTINUE
          WRITE(2,1500) IFRAC,N
          WRITE(3,1500) IFRAC,N
1500      FORMAT(' IFRAC= ',I3,' N= ',I3)
C      WRITE(2,1501) (HPB(I),I=1,IEND)
C      WRITE(3,1501) (HPB(I),I=1,IEND)
C1501      FORMAT(/ ' HPB(I)= ',(6E12.4))
C      WRITE(2,1502) (HPC(I),I=1,IEND)
C      WRITE(3,1502) (HPC(I),I=1,IEND)
C1502      FORMAT(/ ' HPC(I)= ',(6E12.4))
          WRITE(2,1503) (WZM(I,1),I=1,IEND)
          WRITE(3,1503) (WZM(I,1),I=1,IEND)
1503      FORMAT(/ ' MAX. VALUE OF WIDTH IN EACH VERTICAL SECTION (CM) */
          (6E12.4))
C      WRITE(2,1504) (AREAS(I),I=1,IEND)
          WRITE(3,1504) (AREAS(I),I=1,IEND)
1504      FORMAT(/ ' AREAS OF VERT. SECTIONS XA = 1 TO 0 ARE (CM**2) '
          (6E12.4))
C      WRITE(2,15045)
15045      FORMAT(/ ' RATIO1 = FRACTIONAL DISTANCE FROM BOTTOM OF CROSS-',
          / ' SECTION TO CENTER OF PZ, RELATIVE TO TOTAL',
          / ' HEIGHT OF SECTION - USE TO SCALE TOTAL ',
          / ' SECTION TO GRAPH PAGE',
          / ' RATIO2 = FRACTIONAL DISTANCE FROM CENTER OF PZ TO ',

```



```

C ITERATION FOR CONVERGENCE OF TIMES BEGINS HERE
72   TEXPONENT = 1./TSLOPE ;=1/N FOR A(T)=M*T**N
    TDIV = 10.**TCEPT      ;=M

    WRITE(2,4010)TSLOPE,TEXPONENT,TCEPT,TDIV,LEAKC,SP
4010  FORMAT(/' TSLOPE,TEXPONENT = ',2(F10.5,2X),', TCEPT,TDIV = ',
    C 2(F10.4,2X)/' LEAKC = ',F10.6,' SP = ',F10.6,' IN CGS')

C CALC TIMES FROM L/T GRAPH
    NM1 = N - 1
    T(1) = (A1/TDIV)**TEXPONENT
    HL = A/30.48
C DEBUG *****
C   WRITE(2,4035)HL
C4035  FORMAT(' AT T(2), HL = ',E11.4)
    T(2) = (HL/TDIV)**TEXPONENT
    CALC TAU TIMES. TAU = TIME (IN SEC) WHEN FRAC HITS MIDDLE
    C OF SEGMENT
    DO 75 I=1,N
        HL = A1 * FLOAT(I-1)/(FLOAT(N)-1.)
    C   HL = POSITION OF MIDDLE OF SEGMENT (IN FEET)
C DEBUG
C   IF (IDEBUG.EQ.1) WRITE(2,4030)HL
        TAU(I) = (HL/TDIV)**TEXPONENT
C   11 VALUES OF TAU(I) FOR 10 SEGMENTS (IFRAC=1)
75   CONTINUE
C DEBUG *****
C4030  FORMAT(' HL = ',E11.4)
    TAU(12) = T(2)
    NP1 = N + 1
    WRITE(2,4040)(TAU(I),I=1,NP1)
4040  FORMAT(' TAU(I) AS I=1 TO N+1 ',/(6(E11.4,2X)/))
C END DEBUG *****
C CALC LEAKOFF RATE, LEAKOFF VOLUME
    C1 = A1 * 30.48 / (N-1)
    CC = C1 * LEAKC * HZ
    SPL = 2.* SP * HZ
    TC = 4.* HZ * LEAKC * TSLOPE
    DO 86 J=1,2
        TOTLEAK=0.
        NX=NM1/(J-1)
        TX=T(J)
        AVTAU(J)=(TAU(I)**(1.-TSLOPE) + TAU(I+1)**(1.-TSLOPE))/2.
        SPT = 2.* SP * HZ * TSLOPE * TDIV / AVTAU(J)
        DO 95 I=1,NX
            R1 = DSORT(TX-TAU(I))
            R2 = DSORT(TX-TAU(I+1))
            LEAKRATE(J,I) = CC * (1./R1 + 1./R2)
            IF(I.EQ.NX) LEAKRATE(J,I)=TC*TDIV*R1/AVTAU(J)*30.48 + SPT
            LEAKVOLUME(J,I) = 2.* CC * (R1 + R2) + SPL*C1
            TOTLEAK = TOTLEAK + LEAKVOLUME(J,I)
        C 95   CONTINUE
        TTOTLEAK(J)=TOTLEAK + SPL*A1*(1.+(J-1)/10.)*30.48
    C 86   CONTINUE

C IMPROVE TOTLEAK AS ON PAGE Q21
    DO 111 J=1,2
        FX(J) = FFX(0.0,1.0,40,J)
C   FFX COMES FROM NUMERICAL INTEGRATION
        WRITE(2,112) J,FX(J)
112   FORMAT(' FX(',I1,') = ',E13.6)
111   TTOTLEAK(J) = (TC*SQRT(T(J))*FX(1)+SPL)*A1*(1.+(J-1)/10.)*30.48
        WRITE(2,4050) TTOTLEAK(1),VTOT1,TTOTLEAK(2),VTOT2
4050  FORMAT(/' TOTLEAK1(=LEAKOFF + SPINT VOL)=',E11.4,' VTOT1'
    C   '(=FRAC VOL)=',E11.4/' TOTLEAK2(=LEAKOFF + SPINT VOL)=',E11.4,
    C   ' VTOT2(=FRAC VOL)=',E11.4)
C CALC TOTAL VOLUME,T1CHECK,T2CHECK
    TOTVOL1 = VTOT1 + TTOTLEAK(1)
    T1CHECK = TOTVOL1/Q
    TOTVOL2 = VTOT2 + TTOTLEAK(2)
    T2CHECK = TOTVOL2/Q
    VERL1 = TOTVOL1/(2640.*60.)

```



```

VBRL2 = TOTVOL2/(2640.*60.)
VGAL1 = TOTVOL1/3785.
VGAL2 = TOTVOL2/3785.
T1MIN = T1CHECK/60.
T2MIN = T2CHECK/60.
T1PER = 100. * (T1CHECK - T(1))/T1CHECK
T2PER = 100. * (T2CHECK - T(2))/T2CHECK
WRITE(2,2500) T(1),T(2),T1CHECK,T1MIN,T2CHECK,T2MIN,
2500 C TOTVOL1,VBRL1,VGAL1,TOTVOL2,VBRL2,VGAL2
C FORMAT(' T(1)=',E11.4,' T(2)=',E11.4/' (SEC,MIN) :')
C ' T1CHECK=',2E11.4,2X,' T2CHECK=',2E11.4/
C ' TOTAL VOLUME (FRAC1): (CGS,BRL,GAL)',3E11.4/
C ' TOTAL VOLUME (FRAC2): (CGS,BRL,GAL)',3E11.4/)
2550 C WRITE(2,2550) T1PER,T2PER
C FORMAT(' % DIFF. BETWEEN T1CHECK AND T(1) =',F14.6/
C ' % DIFF. BETWEEN T2CHECK AND T(2) =',F14.6)
DT = T2CHECK - T1CHECK
T2MT1 = T(2)-T(1)
754 C WRITE(2,754) DT,T2MT1
C FORMAT(/' DT (T2CHECK-T1CHECK)=' ,E11.4,' SEC., T(2)-T(1)='
, E11.4,' SEC.')
```

```

DIFF = T1PER + T2PER
DIFA = ABS(T1PER) + ABS(T2PER)
IF(T1PER.GT.0.) GOTO 61
IF(ABS(DIFF).LT.DIFA) GOTO 71
74 C XL = XM
C GOTO 62
61 C IF(ABS(DIFF).LT.DIFA) GOTO 71
77 C XR = XM
62 C XM = (XR + XL)/2.
C TSLOPE = XM
C TCEPT = TSLOPE
71 C GOTO 72
C CONTINUE
C CONVERGE T1PER AND T2PER BY CONVERGING DIFF TO 0
IF(ABS(DIFF).LE.TIMP) GOTO 78
IF(DIFF.LT.ABS(DIFF)) GOTO 74
GOTO 77

78 C TSLOPE = XM
C TCEPT = TSLOPE
C WRITE(3,4010)TSLOPE,TEXPONENT,TCEPT,TDIV,LEAKC,SP
C WRITE(3,4035)HL
C IF (IDEBUS.EQ.1) WRITE(3,4030)HL
C WRITE(3,4040)(TAU(I),I=1,NP1)
C WRITE(3,4050) ITOTLEAK(1),VTOT1,ITOTLEAK(2),VTOT2
C WRITE(3,2500) T(1),T(2),T1CHECK,T1MIN,T2CHECK,T2MIN,
C TOTVOL1,VBRL1,VGAL1,TOTVOL2,VBRL2,VGAL2
C WRITE(3,2550) T1PER,T2PER
C WRITE(3,754) DT,T2MT1
DO 88 J=1,2
  NX=NM1+(J-1)
  WRITE(2,2590) J
  WRITE(3,2590) J
2590 C FORMAT(/' FRAC ',I2)
C WRITE(2,2600) (LEAKRATE(J,I),I=1,NX)
C WRITE(3,2600) (LEAKRATE(J,I),I=1,NX)
2600 C FORMAT(' LEAKRATE(X) AS X=0 TO A :'/ (6(E11.4,2X)))
88 C CONTINUE
C
C NOW CALC FLOW RATE DOWN EACH SEGMENT: Q(X) AS X=A TO 0
C
QX(1) = Q
IV1 = N-1
IV2 = N
DO 755 I=2,NM1
  M = I-1
  QX(1) = QX(M)-(VX(2,IV2)-VX(1,IV1))/DT-LEAKRATE(1,M)
  IV1 = IV1 - 1
  IV2 = IV2 - 1
755 C

```

```

C IMPROVE Q(A) AS ON PAGE Q21
  QC1 = TC/2.*A1*30.48*FX(2)/SQRT(T(1))
  QC2 = SPL*TSLOPE*A1*30.48/T(1)
  QX(11) = QX(1) - (VTOT2-VX(2,1)-VTOT1)/DT - QC1 - QC2
DO 756 I=1,N
756   QX(I) = QX(I)/2640.
  WRITE(2,3000) (QX(I),I=1,N)
  WRITE(3,3000) (QX(I),I=1,N)
3000  FORMAT(/'  OUTPUT Q(X) AS X=A TO 0:'/(6(E11.4,2X)/))

C RESET ANY -VE QX TO QXLAST
  QXLAST = 0.0
  NEG = 0
  DO 12 I=1,N
    IF(QX(I).GT.0.) GOTO 12
    NEG = 1
    QX(I) = QXLAST
12    CONTINUE
  IF(NEG.EQ.1) WRITE(2,13) QXLAST
13    FORMAT(/'  ERROR ..... ERROR ..... ERROR .....'/
C      '  ERROR ..... ERROR ..... ERROR .....'/
C      '  FOR COMPUTING NEW INPUT QX, ANY -VE QX IS RESET TO :',F5.3/)
  IF(NEG.EQ.1) WRITE(3,13) QXLAST

C REFORM AND WRITE OUT INPUT QX
  INQX(1) = Q/2640.
  QNOT = INQX(1)
  XA = 0.0
  XAINCR = 1./(N-1)

C DEBUG
  IF (IDEBUG.EQ.1) WRITE(2,4200)SLOPE,SLOPE2
4200  FORMAT('  SLOPE = ',F8.5,'  SLOPE2 = ',F8.5)
  DO 3100 I=2,N
    XA = XA + XAINCR
    IF(XA.LT.XA1)INQX(I) = QNOT*(1.-XA)**DELTA
    IF(XA.GE.XA1.AND.XA.LT.XA2)INQX(I) = (-SLOPE*XA + CEPT)*QNOT
    IF(XA.GE.XA2)INQX(I) = (-SLOPE2*XA + CEPT2)*QNOT

C DEBUG
  IF (IDEBUG.EQ.1) WRITE(2,4300)XA
3100  CONTINUE
4300  FORMAT('  XA = ',E11.4)
  WRITE(2,3200)(INQX(I),I=1,N)
  WRITE(3,3200)(INQX(I),I=1,N)
3200  FORMAT('  INPUT QX, AS X=A TO 0:'/(6(E11.4,2X)/))
  XMAT1 = QNOT*(1.-XA1)**DELTA
  XMAT2 = (-SLOPE*XA2 + CEPT)*QNOT
  WRITE(2,4400)XA1,XA2,XMAT1,XMAT2
  WRITE(3,4400)XA1,XA2,XMAT1,XMAT2

C PRINT MAX-QX OUTPUT AND ITS %DIF FROM INQX
  QMAX = 0.
  DO 758 I=1,N
    QXMQI = QX(I)-INQX(I)
    IF(QXMQI.LE.QMAX) GOTO 758
    QMAX = QXMQI
    KMAX = I
758  CONTINUE
  QMP = (QMAX / INQX(KMAX)) * 100.
  WRITE(2,757) QX(KMAX),QMP
  WRITE(3,757) QX(KMAX),QMP
757  FORMAT('  MAX.(QX OUTPUT, %DIF) = ',2(E11.4,2X))
4400  FORMAT('  XA1 = ',F7.4,'  XA2 = ',F7.4/'  QX AT XA=XA1 USING DELT
C      A CALC.'E11.4/'  QX AT XA = XA2, USING SLOPE AND CEPT ',E11.4)
999  RETURN
END

```

COMPILER DOUBLE PRECISION

```

C *****
C THIS SUBROUTINE PERFORMS RUNGE-KUTTA SOLUTION OF
C DIFFERENTIAL EQN ARISING FROM W(P) AND F(W) EQNS.
C WHERE W(P) IS CORRECTED 2D EQN. AND F(W) IS 1D
C FLUID FLOW EQN.
C *****
C SUBROUTINE RK2(A,STEP,XAZ,YZ,NDIV,N,PXA,RXA,NRES,MFAC,IFRAC,H2,
C
C      KDIV,ISEG,KCB,NTEMP)
COMMON/EBLOC/ HZH,P01,P02,P03,Y
COMMON /FBLOC/ IWR,ISC,IDEBUG
DIMENSION PXA(22),RXA(22)
REAL KCB

C=====DEBUG=====
IF (IDEBUG.EQ.1) WRITE(2,40)
40  FORMAT(' IN DRK2COZ7 ')
C=====END DEBUG=====

IF(ISEG.GE.3) GO TO 20
IPR=0
PXA(1)=YZ
RXA(1)=XAZ
Y=YZ
XA=XAZ
IWFLAG = 0 ;USED TO REDUCE NUMBER OF CALLS TO WCOZ7
DO 11 J=1,KDIV
  CALL KCONT(KCB,ISEG,NTEMP,IPR)
  T1=-STEP*FUN(XA,Y,ISEG,IWFLAG)
  IF (IDEBUG.EQ.1) WRITE(2,100)Y
100  FORMAT(1X,' IN RK2, Y = ',E20.10)
  IWFLAG = 1 ;FUN WON'T CALL WCOZ7 WHEN IWFLAG = 1
  T2=-STEP*FUN(XA-H2,Y+T1/2.,ISEG,IWFLAG)
  T3=-STEP*FUN(XA-H2,Y+T2/2.,ISEG,IWFLAG)
  T4=-STEP*FUN(XA-STEP,Y+T3,ISEG,IWFLAG)
  IWFLAG = 0 ;WCOZ7 WILL BE CALLED ONCE NEXT TIME THRU
C  ALL STEPS ARE -VE COZ WE ARE INTEGRATING BACKWARDS ALONG
C  THE X AXIS (DY=-DX * F(X,Y) )
C  Y=Y+ A* (T1+2.*T2 + 2.*T3 + T4)/6.
C  THIS *A IS DONE HERE INSTEAD OF IN FUN, COZ ALL FUN( )
C  HAVE TO BE MULTIPLIED BY A.
  IF(IPR.GT. 5) GO TO 31
  WRITE(2,55) STEP,XA,T1,T2,T3,T4,Y
55  FORMAT(' STEP= ',E12.3,' XA= ',E12.3,' T1,T2,T3,T4= '
C      4E12.3,' Y= ',E12.3)
  IPR=IPR+1
31  CONTINUE
11  XA=XA-STEP
  PXA(2) = Y
  RXA(2) = XA
20  RETURN
CONTINUE
ITEMP=ISEG-1
XA=RXA(ITEMP)
Y=PXA(ITEMP)
IWFLAG = 0 ;USED TO REDUCE NUMBER OF CALLS TO WCOZ7
DO 1 J=1,KDIV
  CALL KCONT(KCB,ISEG,NTEMP,IPR)
  IF (IDEBUG.EQ.1) WRITE(2,100)Y
  T1=-STEP*FUN(XA,Y,ISEG,IWFLAG)
  IWFLAG = 1 ;FUN WON'T CALL WCOZ7 WHEN IWFLAG = 1
  T2=-STEP*FUN(XA-H2,Y+T1/2.,ISEG,IWFLAG)
  T3=-STEP*FUN(XA-H2,Y+T2/2.,ISEG,IWFLAG)
  T4=-STEP*FUN(XA-STEP,Y+T3,ISEG,IWFLAG)
  IWFLAG = 0 ;WCOZ7 WILL BE CALLED ONCE NEXT TIME THRU
C  ALL STEPS ARE -VE COZ WE ARE INTEGRATING BACKWARDS ALONG
C  THE X AXIS (DY=-DX * F(X,Y) )
C  Y=Y+ A* (T1+2.*T2 + 2.*T3 + T4)/6.

```

```

C          THIS *A IS DONE HERE INSTEAD OF IN FUN, COZ ALL FUN( )
C          HAVE TO BE MULTIPLIED BY A.
          IF(IPR.GT. 5) GO TO 3
          WRITE(2,55) STEP,XA,T1,T2,T3,T4,Y
          IPR=IPR+1
3         CONTINUE
1         XA=XA-STEP
          PXA(ISEG)= Y
2         RXA(ISEG)= XA
          RETURN
          END

```

COMPILER DOUBLE PRECISION

```

C *****
C DFUNCOZ7 02/08/84
C THIS FUNCTION SUBPROGRAM IS CALLED BY RK2 IN RUNGE-KUTTA
C SOLUTION OF DIFF.EQN. ARISING FROM W(P) AND P(W) EQNS.
C *****
C FUNCTION FUN(XA,Y,ISEG,IWFLAG)
C DIMENSION HFB(22),HPC(22),QX(22)
C DIMENSION VX(2,22),TIME(2),PXA(22),WZ(22,22),WZM(22,2),AREAS(22)
C COMMON /EBLOC/ HZH, P01, P02, P03
C COMMON /CBLOC/ XA1,CEPT,SLOPE,XA2,CEPT2,SLOPE2
C COMMON/ABLOC/ B,C2,HZ,C4,CONST,E,XAC,TOF,NFLOW,DELTA,HNEWB,
C HNEWC,HOLDB,HOLDC,XAOLD,XAINC,HFB,HPC
C COMMON/SUBVTQ/IFRAC,VX,TIME,AFAC,ITQ,Q,A,PI,S2MS1,S3MS1,WFLAG,
C PXA,N,EMOD,RNU,NODIV
C COMMON /SUBWIDTH/ WZ,WZM,AREAS,NUMWID,I,L,H
C COMMON/WCBLOC/ EL,Y3,Y4,P1,WCON,IVER
C COMMON/FBLOC/ IWR,ISC,IDERUG
C COMMON/GBLOC/C1
C COMMON/KCNBLOC/ HBM,HCM
C COMMON/NONNEWT/ FLN,FLOWK,C5,FLNF1,FLNF
C COMMON/BBLOC/ IMU
C INTEGER WFLAG

```

```

C=====DEBUG=====
20 IF (IDERUG.EQ.1) WRITE(2,20)
   FORMAT(' IN DFUNCOZ7 ')
C=====END DEBUG=====

```

```

C SUM=HBM + HCM
  H= SUM * HZH
  THIS IS APPROXIMATION TO VARIABLE C(X).
  EL=(HBM + HCM)*HZH/2.
  ELMB=EL - HBM*HZH
  IF(HBM.EQ.HCM)ELMB=0.
  Y3= -(ELMB + HZH)
  Y4=HZH - ELMB
  P1=Y - ELMB * P01

```

```

C NEED BELOW THESE TEMPORARY PARAMETERS TO COMPUTE IN BETWEEN
C MAIN VERTICAL SECTIONS
  HFB(ISEG) = HBM
  HPC(ISEG) = HCM

```

```

IAREA=1
IF(IWFLAG.EQ.1) GOTO 200
CALL WCDZ7(ISEG,WZ,WZH,AREAS,HFB,HFC,C1,IAREA)
C      WRITE(2,30)WZH(ISEG,1)
C30    FORMAT(1X," WCON = ",E20.10)
C200   WCON = WZH(ISEG,1)
C      THIS WCON IS NOW THE MAX WIDTH DETD FROM THE
C      RESOLUTION FOR WIDTH CALCN IN A VERT SECTION (USUALLY 10)
C      F1 = H**FLN * WCON**(FLNF)
C      BELOW ASSUMES FLUID FLOW ACROSS ENTIRE VERTICAL SECTION OF
C      FRACTURE.
      DUM = XA
      IF(IMU.EQ.0) DUM = 0.0
      IF(XA.LT.XA1) F2=(1.- DUM) * ((1.-XA) **DELTA) **FLN
      IF(XA.GE.XA1.AND.XA.LT.XA2) F2=(1.- DUM) * (CEPT - SLOPE * XA)
C      **FLN
      IF(XA.GE.XA2) F2 = (1.- DUM) * (CEPT2-SLOPE2*XA) **FLN
C      THE (1.-XA)**(1+DELTA) TERM IS X VARIATION FOR BOTH KPRIMEAND Q.
      FUN = C5 * F2/F1

C      WE HAVE NOT DISTINGUISHED FRAC 1 FROM FRAC 2 HERE - THIS
C      MEANS, AS A FUNCTION OF XA, THE FLOW RATE IS THE SAME FOR
C      BOTH. THE ONLY DIFFERENCE IS THAT RXA(2) = .9 FOR FRAC 1,
C      = .908 FOR FRAC2. THE PRESSURE AT RXA(2) WILL BE LESS IN
C      FRAC 2 THAN FRAC 1 BECAUSE Q(X) IS LESS AT RXA(2). NOTE THAT
C      THE ABSOLUTE LENGTH OF THE FIRST SEGMENT IS THE SAME IN BOTH
C      FRAC 1 AND 2 ( = A/10).

      IF(ISC.LT.5) WRITE(2,50) ISEG,F1,XA2,C4,XA,CEPT2,SLOPE2,H,F2,FUN
C50    FORMAT(" ISEG=",I3," F1=",E10.3," XA2=",E10.3,
C      " C4=",E10.3," XA=",E10.3," CEPT2=",E10.3," SLOPE2=",E10.3,
C      " H=",E10.3," F2=",E10.3," FUN=",E10.3)
C      THIS FUN HAS TO BE MULTIPLIED BY A - SEE Q22 - BUT WE
C      DO IT IN RK2 INSTEAD - IT'S EASIER.
      RETURN
      END

```

```

C      COMPILER DOUBLE PRECISION
C      CINT7
C      *****
C      *****
C      THIS FUNCTION SUBPROGRAM IS TO CALCULATE THE INTEGRAL FORM WHICH IS
C      REPEATED IN THE KIB, KIC CALCULATION -- SEE PAGE (C4)
C      *****
C      FUNCTION CINT(F1,F2,Q,R,ELL)

      X1=P1/ELL
      X2=P2/ELL
      F1= ASIN(X1)
      F2= ASIN(X2)
      S1= DSQRT(1.- X1**2)
      S2= DSQRT(1.- X2**2)
      C1= Q*ELL*(F2 - F1 - S2 + S1)
      C2= R* ELL*ELL*(S1-S2)
      C3= R* ELL/2. *(P2*S2 - F1*S1)
      C4= R* ELL*ELL/2. *(F2 - F1)
      CINT= C1 - C2 + C3 - C4
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      SUBROUTINE KCONT(KCB,ISEG,NTEMP,IPR)
C=====
C
C      THIS IS SUBROUTINE TO CONTROL KIB=KCB, KIC=KCB ITERATION.
C=====
      DIMENSION C(3),D(9),PXA(22),RXA(22),PXA(22),VX(2,22),TIME(2)
      DIMENSION PSEC(22),HPB(22),HPC(22),QX(22)
      INTEGER WFLAG,FFLAG,KIFLAG
      REAL K,KSQ,KPSQ,KIA,KIB,KI1,KI2,KDIF,M,L,KCA,KCB
      COMMON/SUBVTQ/IFRAC,VX,TIME,AFAC,ITQ,Q,A,PI,S2MS1,S3MS1,WFLAG,
C      PXA,N,EMOD,RNU,NODIV
      COMMON/ABLOC/ B,C2,HZ,C4,CONST,E,XAC,TOF,NFLOW,DELTA,HNEWB,
C      HNEWC,HOLDB,HOLDC,XAOLD,XAINC,HPB,HPC
      COMMON/DBLOC/ DIFB,DIFC,DIFFAC
      COMMON/EBLOC/ HZH, P01,P02,P03
      COMMON/KCNBLOC/ HBM,HCM
      COMMON/FBLOC/ IWR,ISC,IDEBUG
      IF (IPR.NE.0) GO TO 500
      HBL = 1.001
      HBR = 1.2
      HCL = HBL
      HCR = HBR
      HCM = HCL
      GOTO 510
500    HBL = HBM
      HER = 1.1 * HBM
      HCL = HCM
      HCR = 1.1 * HCM
510    CONTINUE
      DO 400 J = 1,10
C      METHOD OF FALSE POSITION SOLUTION FOR HBM
      DO 100 I = 1,100
        T1 = RKB(HER,HCM,KCB,ISEG) - KCB; SUBTRACTING KCB MAKES
        T2 = RKB(HBL,HCM,KCB,ISEG) - KCB; SELECTION LOGIC CORRECT.
53      CONTINUE
        IF (T2)63,73,73
63      HBL=HBL - 0.01*HBL ;THIS MAKES SURE THAT
        T2 = RKB(HBL,HCM,KCB,ISEG)-KCB;RKB WILL ALWAYS BE>=KCB
73      GOTO 53
        CONTINUE
        HBMOLD = HBM
        HRM = (HBL*T1 - HBR*T2)/(T1-T2)
        IF (IDEBUG.EQ.1) WRITE(2,10)I,HBL,HBM,HER,HBMOLD,HCM
10      C      FORMAT(' I= ',I3,' HBL,HBM,HER= ',3F10.3,' HBMOLD=',F10.3,
          ' HCM = ',F10.3)
        T5 = RKB(HBM,HCM,KCB,ISEG) - KCB
        T6 = RKB(HBL,HCM,KCB,ISEG) - KCB
        IF (T5 * T6)20,30,40
20      HBR = HBM
        GOTO 50
30      HBR = HBM
40      HBL = HBM
50      CONTINUE
        DIFB = T5/KCB * 100.
      IF (IDEBUG.EQ.1)TYPE= ' T5 =',T5,' DIFB = ',DIFB,' DIFFAC=',DIFFAC
      IF (ABS(DIFB).LT.DIFFAC)GOTO 150
100    CONTINUE
      WRITE(2,125)
125    FORMAT(' NO SOLUTION TO RKB AFTER 100 ITERATIONS ')
      STOP

```

```

150      CONTINUE
C      METHOD OF FALSE POSITION SOLUTION FOR HCM
      DO 300 I = 1,100
        IF (HCR.LT.HBM) GOTO 301
        IF (IDEBUG.EQ.1) WRITE(2,302)HCR,HBM
302      C      FORMAT(' HCR = ',F10.3,' GREATER THAN HBM = ',F10.3,
        ' SET HCR = HBM .')
        HCR = HBM
301      T3 = RKC(HBM,HCR,KCB,ISEG) - KCB
82      CONTINUE
        IF (T3)102,92,92
92      HCR = HCR + 0.01*HCR ;THIS MAKES SURE THAT
        T3 = RKC(HBM,HCR,KCB,ISEG) - KCB;T3 WILL ALWAYS BE -VE
        GOTO 82
102     CONTINUE
83      T4 = RKC(HBM,HCL,KCB,ISEG) - KCB
        CONTINUE
93      IF (T4)93,103,103
        HCL=HCL - 0.01*HCL ;THIS MAKES SURE THAT
        T4 = RKC(HCL,HCM,KCB,ISEG)-KCB;RKC WILL ALWAYS BE>=KCB
        GOTO 83
103     CONTINUE
C      HCL < HBM ALWAYS I THINK - IS IT ??????????????????????
        HCMOLD = HCM
        HCM= (HCL*T3 - HCR*T4)/(T3-T4)
        IF (IDEBUG.EQ.1) WRITE(2,210)I,HBM,HCL,HCM,HCR,HCMOLD
210     C      FORMAT(' I = ',I3,' HBM = ',F10.3,
        ' HCL,HCM,HCR = ',3F10.3,' HCMOLD = ',F10.3)
        T7 = RKC(HBM,HCM,KCB,ISEG) - KCB
        T8 = RKC(HBM,HCL,KCB,ISEG) - KCB
        IF (T7 * T8)220,230,240
220     HCR = HCM
230     HCR = HCM
240     HCL = HCM
250     CONTINUE
        DIFC = T7/KCB * 100.
        IF (ABS(DIFC).LT.DIFFAC) GOTO 350
300     CONTINUE
        WRITE(2,325)
325     FORMAT(' NO SOLUTION TO RKC AFTER 100 ITERATIONS')
        STOP
350     CONTINUE
C      IF (((ABS(HBM - HBMOLD)/HBM*100.).LT.DIFFAC).AND.
        ((ABS(HCM - HCMOLD)/HCM*100.).LT.DIFFAC)) GOTO 700
400     CONTINUE
425     WRITE(2,425)
        FORMAT(' NO SOLUTION TO KIB=KIC=KCB AFTER 10 ITERATIONS')
        STOP
700     CONTINUE
C 700    WRITE(2,710) DIFFAC,HBM,HCM,HBMOLD,HCMOLD
        IF(S2MS1.EQ.S3MS1)HCM=HBM; FIX FOR SYMMETRIC CASE
710     C      FORMAT(' KIB = KIC = KCB TO WITHIN ',F10.1,'% WHEN HBM,HCM = ',
        2F10.3,' HBMOLD,HCMOLD = ',2F10.3)
        IF(IPR.NE.0)GOTO 800
        HPB(1) = HBM
        HPC(1) = HCM
800     RETURN
        END

```

```

      COMPILER DOUBLE PRECISION
C*****
C*****
C      THIS CALCULATES THE VALUE OF KIB AT TOP
C      OF FRAC AT DIFFERENT POINTS ALONG THE PAYZONE.
C      THE STRESSES CAN BE ASYMMETRIC.
C*****
C      FUNCTION RKB(HNEWB,HNEWC,KCB,ISEG)
      DIMENSION TIME(2),VX(2,22),PXA(22),HPB(22),HPC(22)
      INTEGER WFLAG
      REAL KCB
      COMMON/ABLOC/ B,C2,HZ,C4,CONST,E,XAC, TOP,NFLOW,DELTA,HJUNK,
C      HSTUF,HOLDB,HOLDC,XAOLD,XAINC,HPB,HPC
      COMMON/DBLOC/ DIFB,DIFC,DIFFAC
      COMMON/SUBVTQ/ IFRAC,VX,TIME,AFAC,ITQ,Q,A,PI,S2MS1,S3MS1,WFLAG,
C      PXA,N,EMOD,RNU,NODIV
      COMMON/FBLOC/ IWR,ISC,IDEBUG
      COMMON/EBLOC/ HZH,P01,P02,P03,Y

C      NOW CALC KIB
      EL = (HNEWB + HNEWC) * HZH / 2.
      ELMB = EL - HNEWB * HZH
      IF (HNEWB.EQ.HNEWC) ELMB = 0.
      Y3 = -(HZH - ELMB)
      Y4 = ELMB + HZH
C      BELOW P01,P02,P03 ARE NEGATIVE OF VALUES USED IN K1C CALCULATIONS
C      SO WE JUST REVERSE THEIR SIGNS IN THE FOLLOWING EQNS.
      P1 = Y - ELMB * P01
      P2 = P1 + (P01 - P02) * Y4 - S2MS1
      P3 = P1 + (P01 - P03) * Y3 - S3MS1
      IF (IDEBUG.EQ.1) WRITE(2,100) EL,ELMB,Y3,Y4,P1,P2,P3
100  C      FORMAT(' EL = ',E12.4,' ELMB= ',E12.4,' Y3= ',E12.4,
      ' Y4= ',E12.4/' P1= ',E12.4,' P2= ',E12.4,' P3= ',E12.4)

      IF (HNEWB.LT. 1..AND.HNEWC.GE. 1.) GO TO 50
      IF (HNEWB.GE. 1..AND.HNEWC.LT. 1.) GO TO 60
      VAL4 = CINT(-EL,Y3,P3,-P03,EL)
      VAL5 = CINT(Y3,Y4,P1,-P01,EL)
      VAL6 = CINT(Y4,EL,P2,-P02,EL)
      GOTO 70
50  VAL4 = CINT(-EL,Y3,P3,-P03,EL)
      VAL5 = CINT(Y3,EL,P1,-P01,EL)
      VAL6 = 0.
      GOTO 70
60  VAL4 = 0.
      VAL5 = CINT(-EL,Y4,P1,-P01,EL)
      VAL6 = CINT(Y4,EL,P2,-P02,EL)
70  IF (IDEBUG.EQ.1) WRITE(2,111) VAL4,VAL5,VAL6
111  FORMAT('/ VAL4= ',E12.4,' VAL5= ',E12.4,' VAL6= ',E12.4)
      RKB = (VAL4 + VAL5 + VAL6) / DSQRT(PI * EL)
      RETURN
      END

```



```

C*****COMPILER DOUBLE PRECISION*****
C*****
C THIS CALCULATES THE VALUE OF KIC AT BOTTOM
C OF FRAC AT DIFFERENT POINTS ALONG THE PAYZONE
C THE STRESSES CAN BE ASYMMETRIC.
C *****
C *****
C FUNCTION RKC(HNEWB,HNEWC,KCB,ISEG)
C INTEGER WFLAG
C REAL KCB
C COMMON/ABLOC/B,C2,HZ,C4,CONST,E,XAC,TOF,NFLOW,DELTA,HJUNK,
C HSTUF,HOLDB,HOLDC,XAOLD,XAINC,HFB,HPC
C COMMON/DBLOC/ DIFB,DIFC,DIFFAC
C DIMENSION TIME(2),VX(2,22),FXA(22),HFB(22)
C ,HPC(22)
C COMMON/SUBVTQ/ IFRAC,VX,TIME,AFAC,ITQ,Q,A,PI,S2MS1,S3MS1,WFLAG,
C PXA,N,EMOD,RNU,NODIV
C COMMON/FBLOC/ IWR, ISC,IDEBUG
C COMMON/EBLOC/ HZH,P01,P02,P03,Y

C NOW CALC. KIC
EL=(HNEWB + HNEWC) *HZH /2.
ELMB=EL - HNEWB *HZH
IF(HNEWB.EQ.HNEWC) ELMB=0.
Y3= -(ELMB + HZH)
Y4= HZH - ELMB
P1= Y - ELMB*P01
P2= P1 + (P02 - P01)*Y3 - S2MS1
P3= P1 + (P03 - P01)*Y4 - S3MS1
IF(IDEBUG.EQ.1) WRITE(2,100) EL,ELMC,Y3,Y4,P1,P2,P3
100 C FORMAT(' EL= ',E12.4,' ELMC= ',E12.4,' Y3= ',E12.4,
' Y4= ',E12.4/' P1= ',E12.4,' P2= ',E12.4,' P3= ',E12.4)
IWR=IWR+1

IF(HNEWB.LT. 1..AND.HNEWC.GE. 1.) GOTO 10
IF(HNEWB .GE. 1..AND.HNEWC.LT. 1.) GOTO 20
VAL1= CINT(-EL,Y3,P2,P02,EL)
VAL2= CINT(Y3,Y4,P1,P01,EL)
VAL3= CINT(Y4,EL,P3,P03,EL)
GOTO 30
10 VAL1=0.
VAL2= CINT(-EL,Y4,P1,P01,EL)
VAL3= CINT(Y4,EL,P3,P03,EL)
C HERE Y3 IS INCORRECT, BUT NO MATTER COZ P2 NOT USED!
GOTO 30
20 VAL1= CINT(-EL,Y3,P2,P02,EL)
VAL2= CINT(Y3,EL,P1,P01,EL)
VAL3=0.
C HERE Y4 IS INCORRECT, BUT NO MATTER COZ P3 IS NOT USED!
30 IF(IDEBUG.EQ.1) WRITE(2,110) VAL1,VAL2,VAL3
110 C FORMAT('/ VAL1= ',E12.4,' VAL2= ',E12.4,' VAL3= ',E12.4)
RKC = (VAL1 + VAL2 + VAL3)/DSQRT(PI*EL)
RETURN
END

```

```

C *****
C      COMPILER DOUBLE PRECISION
C      SUBROUTINE WCOZ7(I,WZ,WZM,AREAS,HFB,HFC,C1,IAREA)
C*****
C      WCOZ7 - USED TO DO WIDTH CALC. ANALYTICALLY
C      USES DFUN1 THRU DFUN4, DFN57 AND DFN58
C      CREATED 11/15/82
C*****
C      CHANGE ALL Y3 TO -Y3 IN FUNCTIONS,CHANGE CONSTANT F,G CALCNS
C      ON 9/13/83
C*****

      INTEGER WFLAG,LASTI
      DIMENSION WZ(22,22),WZM(22,2),AREAS(22),HFB(22),HPC(22),UX(2,22)
      DIMENSION TIME(2),PXA(22)
      COMMON/SUBVTQ/IFRAC,UX,TIME,AFAC,ITR,Q,A,PI,S2MS1,S3MS1,WFLAG,
C          PXA,N,EMOD,RNU,NODIV
      COMMON/EBLOC/HZH,P01,P02,P03
      COMMON/FBLOC/IWR,ISC,IDEBUG
      COMMON/WCBLOC/EL,Y3,Y4,F1,WCON,IVER
      COMMON/INWBLC/LASTI

      DISPLAY WIDTHS WHEN I CHANGES FROM 2 TO 3, 3 TO 4, 4 TO 5, ETC
      IF (LASTI.EQ.I) GOTO 55
      IF (LASTI.LE.0) LASTI = 1
      WRITE(2,50) LASTI,(WZ(LASTI,J),J=1,IVER)
      FORMAT(/' I = ',I4,' WZ(I,J) = '(/6E12.4))
      LASTI = I
      CONTINUE

C *****DEBUG*****
      IF (IDEBUG.NE.2) GOTO 350
      TYPE ' IN WCOZ7, I = ',I
      TYPE ' HFB(I) = ',HFB(I)
      TYPE ' HFC(I) = ',HFC(I)
      TYPE ' Y3 = ',Y3
      TYPE ' Y4 = ',Y4
      TYPE ' HZH = ',HZH
      TYPE ' EL = ',EL

      GO TO 350

C *****
C      CASE 1 - FOR YMOD<-Y3 AND L>Y4 (WHICH IMPLIES -L<Y3 SINCE B>C)
C*****

      100 CONTINUE

C DEBUG
      105 IF(IDEBUG.EQ.2) WRITE(2,105)
C DEBUG
      FORMAT(' IN CASE ONE CALCULATIONS IN WCOZ7')

      WCONA1 = 2*PI*F1/4.*FUN2(-Y3,YMOD,Y)
C          + F01*FUN3(-Y3,YMOD,Y)

      WCONB1 = G1*FUN1(-Y3,YMOD,Y) + PI/4.*G01*FUN2(-Y3,YMOD,Y)

      WCONA2 = PI/2.*F1*FUN2(Y4,-Y3,Y) + F01*FUN3(Y4,-Y3,Y)
C          + (F02 - F01)*FN57(Y4,-Y3,-Y3,Y)
C          + (F2 - F1)*FUN4(Y4,-Y3,-Y3,Y)

      WCONB2 = G1*FUN1(Y4,-Y3,Y) + PI/4.*G02*FUN2(Y4,-Y3,Y)
C          + (G2 - G1)*FN58(Y4,-Y3,-Y3,Y)
C          + (G02 - G01)*(-Y3)/2.*FN58(Y4,-Y3,-Y3,Y)
C          + (G02 - G01)/(-2.)
C          * (PI/2.*FUN2(Y4,-Y3,Y) - FUN4(Y4,-Y3,-Y3,Y))

```

```

C      WCONA3 = PI/2.*F1*FUN2(EL,Y4,Y) + F01*FUN3(EL,Y4,Y)
C      + (F2 - F1)*FUN4(EL,Y4,-Y3,Y)
C      + (F3 - F2)*FUN4(EL,Y4,Y4,Y)
C      + (F02 - F01)*FN57(EL,Y4,-Y3,Y)
C      + (F03 - F02)*FN57(EL,Y4,Y4,Y)

C      WCONB3 = G1*FUN1(EL,Y4,Y) + PI/4.*G03*FUN2(EL,Y4,Y)
C      + ((G2-G1) + (G02-G01)*(-Y3)/2.)*FN58(EL,Y4,-Y3,Y)
C      + ((G3-G2) + (G03-G02)*Y4/2.)*FN58(EL,Y4,Y4,Y)
C      + (G02 - G01)/(-2.)
C      * (PI/2.*FUN2(EL,Y4,Y) - FUN4(EL,Y4,-Y3,Y))
C      + (G03 - G02)/(-2.)
C      * (PI/2.*FUN2(EL,Y4,Y) - FUN4(EL,Y4,Y4,Y))

C      WCON = WCONA1 + WCONA2 + WCONA3
C      + Y*(WCONB1 + WCONB2 + WCONB3)
C      IF (IDEBUG.EQ.2)WRITE(2,1000)WCONA1,WCONB1,WCONA2,WCONB2,WCONA3,
1000      WCONB3
C      FORMAT(1X," WCONA1,WCONB1 ",2E12.4," WCONA2,WCONB2",2E12.4,
C      /,"WCONA3,WCONB3",2E12.4)
C      GOTO 400

C      *****
C      CASE 2 - WHERE -Y3<|Y|<Y4      NOTE: |Y| = YMOD
C      *****

200      CONTINUE

C DEBUG
205      IF(IDEBUG.EQ.2) WRITE(2,205)
C DEBUG      FORMAT(" IN CASE 2 OF WCOZ7")

C      WCONA2 = PI/2.*F1*FUN2(Y4,YMOD,Y) + F01*FUN3(Y4,YMOD,Y)
C      + (F2 - F1)*FUN4(Y4,YMOD,-Y3,Y)
C      + (F02 - F01)*FN57(Y4,YMOD,-Y3,Y)

C      WCONB2 = G1*FUN1(Y4,YMOD,Y) + PI/4.*G02*FUN2(Y4,YMOD,Y)
C      + (G2 - G1)*FN58(Y4,YMOD,-Y3,Y)
C      + (G02 - G01)*(-Y3)/2.*FN58(Y4,YMOD,-Y3,Y)
C      + (G02 - G01)/(-2.)
C      * (PI/2.*FUN2(Y4,YMOD,Y) - FUN4(Y4,YMOD,-Y3,Y))

C      WCONA3 = PI/2.*F1*FUN2(EL,Y4,Y) + F01*FUN3(EL,Y4,Y)
C      + (F2 - F1)*FUN4(EL,Y4,-Y3,Y)
C      + (F3 - F2)*FUN4(EL,Y4,Y4,Y)
C      + (F02 - F01)*FN57(EL,Y4,-Y3,Y)
C      + (F03 - F02)*FN57(EL,Y4,Y4,Y)

C      WCONB3 = G1*FUN1(EL,Y4,Y) + PI/4.*G03*FUN2(EL,Y4,Y)
C      + ((G2-G1) + (G02-G01)*(-Y3)/2.)*FN58(EL,Y4,-Y3,Y)
C      + ((G3-G2) + (G03-G02)*Y4/2.)*FN58(EL,Y4,Y4,Y)
C      + (G02 - G01)/(-2.)
C      * (PI/2.*FUN2(EL,Y4,Y) - FUN4(EL,Y4,-Y3,Y))
C      + (G03 - G02)/(-2.)
C      * (PI/2.*FUN2(EL,Y4,Y) - FUN4(EL,Y4,Y4,Y))

C      WCON = WCONA2 + WCONA3 + Y*(WCONB2 + WCONB3)
2000      IF (IDEBUG.EQ.2)WRITE(2,2000)WCONA2,WCONB2,WCONA3,WCONB3
C      FORMAT(1X," WCONA2,WCONB2 ",2E12.4," WCONA3,WCONB3",2E12.4)
C      GOTO 400

C      *****
C      CASE 3 - WHERE Y4<|Y|<L NOTE: |Y| = YMOD
C      *****

300      CONTINUE

C DEBUG
305      IF(IDEBUG.EQ.2) WRITE(2,305)
C DEBUG      FORMAT(" IN CASE 3 CALCULATIONS OF WCOZ7")

```

```

C      WCONA3 = PI/2.*F1*FUN2(EL,YMOD,Y) + F01*FUN3(EL,YMOD,Y)
C      + (F2 - F1)*FUN4(EL,YMOD,-Y3,Y)
C      + (F3 - F2)*FUN4(EL,YMOD,Y4,Y)
C      + (F02 - F01)*FN57(EL,YMOD,-Y3,Y)
C      + (F03 - F02)*FN57(EL,YMOD,Y4,Y)

C      WCONB3 = G1*FUN1(EL,YMOD,Y) + PI/4.*G03*FUN2(EL,YMOD,Y)
C      + ((G2-G1)+(G02-G01)*(-Y3)/2.)*FN58(EL,YMOD,-Y3,Y)
C      + ((G3-G2) + (G03-G02)*Y4/2.)*FN58(EL,YMOD,Y4,Y)
C      + (G02 - G01)/(-2.)
C      * (PI/2.*FUN2(EL,YMOD,Y) - FUN4(EL,YMOD,-Y3,Y))
C      + (G03 - G02)/(-2.)
C      * (PI/2.*FUN2(EL,YMOD,Y) - FUN4(EL,YMOD,Y4,Y))

WCON = WCONA3 + Y*WCONB3
IF (IDEBUG.EQ.2)WRITE(2,3000)WCONA3,WCONB3
3000  FORMAT(1X," WCONA3,WCONB3",2E12.4)
      GOTO 400
      GOTO 400

350    CONTINUE

IF (IDEBUG.EQ.2) WRITE(2,380) I,IVER,C1
380    FORMAT(" I = ",I4," IVER= ",I4," C1= ",E12.4)

C      CALCULATE CONSTANTS , F SERIES , G SERIES
      F2=F1 + (P02 - P01)*Y3 - S2MS1
      F3=F1 + (P03 - P01)*Y4 - S3MS1
      F1=P1
      F01=0.0
      F2=(P1 + P2)/2.
      F02=(P02 - P01)/2.
      F3=(P2 + P3)/2.
      F03=(P02 - P03)/2.
      G1=0.0
      G01=-P01
      G2=(P1 - P2)/2.
      G02=-(P01 + P02)/2.
      G3=(P3 - P2)/2.
      G03=-(P03 + P02)/2.
C      ADJUST FOR NEW ORIGIN BELOW PAYZONE
      IF(Y3.GT.0.) GO TO 70
      GO TO 80
70      F1=F2
      IY4=99
      Y3=-Y3
80      ;<== DO WE REALLY NEED THIS ??????????
      CONTINUE

C      DO WIDTH CALCULATIONS, CALLING APPROPRIATE WCON CALCULATIONS.
      IF (IDEBUG.EQ.2)TYPE " Y3 = ",Y3," Y4 = ",Y4

      TOTL = (HPB(I) + HPC(I))*HZH
      ELL = TOTL/2.
C      TYPE "ELL= ",ELL
C      HINC = TOTL/(IVER - 1)
C      Y = -ELL + HINC
C      Y IS POSITION MEASURED FROM CENTER OF FRACTURE
C      SELECT CASE AND CALC. WIDTHS OF A CROSS-SECTION
      WZ(I,1) = 0.0
      WZ(I,IVER) = 0.0
      ITEMP = IVER-1
      ZEROFLAG = 0.0
      DO 500 J = 2,ITEMP
        YMOD = DABS(Y)
        IF(YMOD.GE.0.1) GO TO 41
        WHICH ARE DICY IN FUN1, ETC. CALCULATIONS
      ZEROFLAG = 1.0
      YMOD = 10.0
      Y = 10.0
41      CONTINUE

```

```

IF(IDEBUG.EQ.2)TYPE * Y = ',Y
IF(0.0.LT.YMOD.AND.YMOD.LT.-Y3) GOTO 100
IF(-Y3.LE.YMOD.AND.YMOD.LT.Y4) GOTO 200
IF(Y4.LE.YMOD.AND.YMOD.LT.EL) GOTO 300
400 CONTINUE
WZ(I,J) = WCON
IF(IAREA.EQ.2) WZ(I,J) = WCON*4.*C1/PI
IF(IDEBUG.EQ.2) WRITE(2,390)Y,WZ(I,J),J
IF(IDEBUG.EQ.2) WRITE(3,390)Y,WZ(I,J),J
C 390 IF(I.GE.3) WRITE(2,390)Y,WZ(I,J),J ; ***DEBUG***
FORMAT(/' HEIGHT/HZ/2= ',F10.3,' WIDTH = ',E12.4,' J = ',I4)
IF(ZEROFLAG.NE.1.0) GOTO 42 ;FOR READJUSTING Y AFTER Y=0 CASE
ZEROFLAG = 0.0
YMOD = 0.0
Y = 0.0
42 CONTINUE
Y = Y + HINC
500 CONTINUE

C CALC. MAXIMUM WIDTH
WMAX = WZ(I,1)
JTEMP = 1
DO 550 J = 2,IVER
IF(WZ(I,J).LE.WMAX) GOTO 520
WMAX = WZ(I,J)
JTEMP = J
520 CONTINUE
550 CONTINUE
WZM(I,1) = WMAX
WZM(I,2) = -HFC(I) + (JTEMP - 1)*HINC/HZH
C WZM IS AN ARRAY THAT CONTAINS MAXIMUM WIDTHS, WZM(I,1) IS THE
C MAXIMUM WIDTH FOR THE ITH SECTION AND WZM(I,2) IS THE HEIGHT
C OF THAT MAXIMUM WIDTH IN HZ UNITS ABOVE OR BELOW THE PAYZONE
IF(IDEBUG.EQ.2) WRITE(2,600)WZM(I,1),WZM(I,2)
600 FORMAT(/' MAXIMUM WIDTH = ',E12.4,' HEIGHT/HZH = ',F10.3)
IF(IAREA.EQ.1) GO TO 51

C CALC. AREA OF CROSS SECTION USING WIDTHS (FOR BVCOZ4 ONLY).
ATOT = 0.0
DO 650 J = 2,IVER
ATEMP = (WZ(I,J) + WZ(I,J-1))/2.*HINC
650 ATOT = ATOT + ATEMP
C ASSIGN ATOT TO AREAS(I)
AREAS(I) = ATOT
C WRITE(2,700)AREAS(I),I
700 FORMAT(/' AREAS(I) = ',E12.4,' IN CGS, I = ',I4)
51 CONTINUE

Y3N=-Y3
IF(IDEBUG.EQ.2) WRITE(2,60) Y3N,Y4,EL,F1,F01,F2,F02,F3,F03,
60 C F1,F01,F2,F02,F3,F03,G1,G01,G2,G02,G3,G03,WZM(I,1)
FORMAT(' -Y3= ',E12.4,' Y4= ',E12.4,' EL= ',E12.4/
C ' F1= ',E12.3,' F01= ',E12.3,' F2= ',E12.3,' F02= ',E12.3,
C ' F3= ',E12.3,' F03= ',E12.3/' F1= ',E12.3,' F01= ',E12.3,
C ' F2= ',E12.3,' F02= ',E12.3,' F3= ',E12.3,' F03= ',E12.3/
C /' G1= ',E12.3,' G01= ',E12.3,
C ' G2= ',E12.3,' G02= ',E12.3,' G3= ',E12.3,' G03= ',E12.3/
C ' WZM(I,1) = ',E12.3)
ISC=ISC+1
IF(IY4.EQ.99) Y3=-Y3
IY4=0
C THIS RESTORES Y4 TO ITS RIGHT VALUE
999 RETURN
END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FUN1(I1,INOT,Y)
C*****
C      DFUN1 - USED IN WCOZ5,DFUN6, ANALYTICAL SOLUTION OF INTEGRAL
C      CREATED 11/8/82
C*****
      REAL I1,INOT
      FUN1 = DLOG(I1 + DSQRT(I1*I1 - Y*Y))
C      - DLOG(INOT + DSQRT(INOT*INOT - Y*Y))
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FUN2(I1,INOT,Y)
C*****
C      DFUN2 - USED IN WCOZ5 ANALYTICAL SOLUTION OF INTEGRAL
C      CREATED 11/8/82
C*****
      REAL I1,INOT
      FUN2 = DSQRT(I1*I1 - Y*Y) - DSQRT(INOT*INOT - Y*Y)
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FUN3(I1,INOT,Y)
C*****
C      DFUN3 - USED IN WCOZ5 ANALYTICAL SOLUTION OF INTEGRAL
C      CREATED 11/8/82
C*****
      REAL I1,INOT
      S1 = DSQRT(I1*I1 - Y*Y)
      S2 = DSQRT(INOT*INOT - Y*Y)
      FUN3 = I1/2.*S1 + Y*Y/2.*DLOG(I1 + S1)
C      -(INOT/2.*S2 + Y*Y/2.*DLOG(INOT + S2))
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FUN4(I1,INOT,C2,Y)
C*****
C      DFUN4 - USED IN WCOZ4 ANALYTICAL SOLUTION OF INTEGRAL
C      CREATED 10/1/83
C*****
      REAL I1,INOT
      YS = Y*Y
      CS = C2*C2
      SR1 = DSQRT(I1*I1 - YS)
      SR2 = DSQRT(INOT*INOT - YS)
      SR3 = DSQRT(I1*I1 - CS)
      SR4 = DSQRT(INOT*INOT - CS)

      FUN4 = SR1*ACOS(C2/I1) - C2*DLOG(SR1 + SR3)
C      + C2*DABS(Y/C2)*DLOG((SR1/DABS(Y) + SR3/DABS(C2))/I1)
C      - (SR2*ACOS(C2/INOT) - C2*DLOG(SR2 + SR4)
C      + C2*DABS(Y/C2)*DLOG((SR2/DABS(Y) + SR4/DABS(C2))/INOT))
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FN57(I1,INOT,C2,Y)
C*****
C      DFN57 - USED IN WCOZ4 SOLUTION OF INTEGRAL
C      CREATED 07/25/83
C*****
      REAL I1,INOT
      CS = C2*C2
      YS = Y*Y
      SR1 = DSQRT(I1*I1 - CS)
      SR2 = DSQRT(INOT*INOT - CS)
      SR3 = DSQRT(I1*I1 - YS)
      SR4 = DSQRT(INOT*INOT - YS)

C      FN57 = .5*SR1*SR3 + ((CS - YS)/2. - CS)*DLOG(SR1 + SR3)
C      - (.5*SR2*SR4 + ((CS - YS)/2. - CS)*DLOG(SR2 + SR4))
      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FN58(I1,INOT,C2,Y)
C*****
C      IFN58 - USED IN WCOZ4 SOLUTION OF INTEGRAL
C      CREATED 07/25/83
C*****
      REAL I1,INOT
      CS = C2*C2
      YS = Y*Y
      SR1 = DSQRT(I1*I1 - CS)
      SR2 = DSQRT(INOT*INOT - CS)
      SR3 = DSQRT(I1*I1 - YS)
      SR4 = DSQRT(INOT*INOT - YS)

      FN58 = DLOG(SR1 + SR3) -
C      DABS((C2/Y))*DLOG(((1./(DABS(C2)))*SR1 +
C      (1./(DABS(Y)))*SR3)/I1)
C      -(DLOG(SR2 + SR4) -
C      DABS((C2/Y))*DLOG(((1./(DABS(C2)))*SR2 +
C      (1./(DABS(Y)))*SR4)/INOT))

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C      SEE NOTE PAGE A 20 FOR REASON FOR DABS(C2) AND DABS(Y).
C      THIS CHANGE WAS MADE 9/5/83. HRC
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C      SEE NOTE PAGE A 21 FOR REASON FOR DABS(C2/Y).
C      THIS CHANGE WAS MADE 9/13/83. HRC
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

      RETURN
      END

```

```

      COMPILER DOUBLE PRECISION
      FUNCTION FFX(A,B,NODIV,F)
C*****
C      PERFORM THE NUMERICAL INTEGRATION OF WIDTH EQN USING THE
C      TRAPEZOIDAL METHOD TO INTEGRATE OVER F2 FROM A TO B (FOR FLK10)
C*****
C      PERFORM NUMERICAL INTEGRATION
C
      INTEGER F
      AREA = 0.0
      WIDTH = (B-A) / NODIV

      DO 30 J = 1,NODIV
        XL = A + (J-1) * WIDTH
        IF(XL.EQ.0.0) XL=WIDTH/10.
        XR = A + J * WIDTH
        IF(F.EQ.1) GOTO 2
        IF(B.EQ.1.0.AND.J.EQ.NODIV) XR=1.- WIDTH/10.
        SMALLA = (WIDTH/2.) * ( DFX(XL,F) + DFX(XR,F) )
        AREA = AREA + SMALLA
      2  FFX = AREA
C      WRITE(2,1) F,FFX
C1     FORMAT(' AREA(',I1,')= ',E13.6)
      RETURN
      END

```



```

      COMPILER DOUBLE PRECISION
      FUNCTION DFX( FLVAR,I )
C*****
C DFX, RNUMINT INTEGRATES OVER THIS FUNCTION (FOR FLK10)
C*****
      REAL LEAKC
      COMMON /TCONV/ TSLOPE,TCEPT,LEAKC,SP,TIMP

      IF(I.EQ.1) DFX = FLVAR**(TSLOPE-1.) * SQRT(1.-FLVAR)
      IF(I.EQ.2) DFX = FLVAR**(TSLOPE-1.) / SQRT(1.-FLVAR)
      RETURN
      END

```

```

C
C
C
C
      COMPILER DOUBLE PRECISION
      *****
      THIS SUBROUTINE CALCULATE PHI, USING FRACTURE CROSS SECTION
      WIDTH AND N.
      *****
      SUBROUTINE CLPHI(PHI,WZ,WMAX,FLN,ISEG,IVER,PI,EMOD,RNU)

      DIMENSION WZ(22,22),WC(22)
      REAL M

      M = (2.*FLN + 1.)/FLN
      SUM = 0.0

C *** CONVERT WIDTHS ***
      CONVERTCONST = 8. * (1. - RNU*RNU)/(PI*EMOD)
      DO 100 J=1,IVER
         WC(J) = 0.0
         WC(J) = WZ(ISEG,J)*CONVERTCONST
      100 CONTINUE
C *** CALCULATE PHI FROM WIDTHS ***
C *** CONVERT WMAX ***
      WMAXC = WMAX * CONVERTCONST
      IX = IVER - 1
      DO 200 J=2,IX
         TEMP1 = (WC(J)/WMAXC)**M
         SUM = SUM + TEMP1
      200 CONTINUE
      PHI = SUM/FLOAT(IVER - 1)
      RETURN
      END

```

```
      COMPILER DOUBLE PRECISION
      FUNCTION ASIN(YY)
C     THIS IS USED TO COMPUTE ARCSIN FROM THE DATA GIVEN
C     IF YY IS POSITIVE, RETURNS 0-90
C     IF YY IS NEGATIVE, RETURNS 270-360
      XX=SQRT(1.-YY**2)
      ASIN=ATAN2(YY,XX)
      RETURN
      END
```

```
      COMPILER DOUBLE PRECISION
      FUNCTION ACOS( XX )
C     THIS IS USED TO COMPUTE ARCCOS FROM THE DATA GIVEN
C     IF XX IS POSITIVE, RETURNS 0-90
C     IF XX IS NEGATIVE, RETURNS 90-180
      YY=SQRT(1.-XX**2)
      ACOS=ATAN2(YY,XX)
      RETURN
      END
```

FIGURE 1

SHAPE OF 3200 FRAC

FIGURE 2

FIGURE 3

FIGURE 4

FIGURE 5

FIGURE 6

FIGURE 7

FIGURE 8

FIGURE 9

FIGURE 10

FIGURE 11